

Universidad Nacional de Asunción  
Facultad de Ingeniería  
Programa de Postgrado en Ingeniería Civil



TÉCNICAS DE APRENDIZAJE  
AUTOMÁTICO PARA ANÁLISIS A  
FLEXIÓN EN VIGAS RECTANGULARES DE  
HORMIGÓN ARMADO

Fredy Gabriel Ramírez Villanueva

San Lorenzo - Paraguay  
2019

**TÉCNICAS DE APRENDIZAJE  
AUTOMÁTICO PARA ANÁLISIS A  
FLEXIÓN EN VIGAS RECTANGULARES DE  
HORMIGÓN ARMADO**

Tesis presentada al Programa de Postgrado en Ingeniería Civil de la Universidad Nacional de Asunción, como requisito para la obtención del Grado de Máster en Ciencias de la Ingeniería Civil, énfasis en Estructuras.

Orientador:

Prof. MSc. Sergio Daniel Cardozo López

San Lorenzo - Paraguay

Octubre - 2019

RAMÍREZ VILLANUEVA, Fredy Gabriel

Técnicas de aprendizaje automático para análisis a flexión en vigas rectangulares de hormigón armado/ Fredy Gabriel Ramírez Villanueva. – San Lorenzo: PPGIC / UNA, 2019.

148 p.

Tesis de Maestría, Programa de Postgrado en Ingeniería Civil de la Universidad Nacional de Asunción; Maestría en Ciencias de la Ingeniería Civil. Orientador: Sergio Daniel Cardozo López

1. Hormigón armado 2. Aprendizaje automático  
3. Redes neuronales 4. Vector soporte 5. Vigas

---

**FREDY GABRIEL RAMÍREZ VILLANUEVA**

**TÉCNICAS DE APRENDIZAJE  
AUTOMÁTICO PARA ANÁLISIS A  
FLEXIÓN EN VIGAS RECTANGULARES DE  
HORMIGÓN ARMADO**

Esta tesis fue juzgada adecuada para la obtención del Grado de MÁSTER EN CIENCIAS DE LA INGENIERÍA CIVIL, y aprobada en su forma final por los profesores orientadores y por el Programa de Postgrado en Ingeniería Civil de la Universidad Nacional de Asunción.

---

Prof. Sergio Daniel Cardozo López

MSc. por la Universidade Federal do Rio Grande do Sul  
Orientador

---

Prof. Fulgencio Antonio Aquino Duarte

Coordinador del PPGIC/UNA

**BANCA EXAMINADORA**

---

Prof. Miembro 1 Universidad...

PhD por \_\_\_\_\_

---

Prof. Miembro 2 Universidad...

PhD por \_\_\_\_\_

---

Prof. Miembro 3 Universidad...

MSc por \_\_\_\_\_

*A mi mamá*

*“El salto de 0 a 1 es mayor  
que el de 1 a 100”  
Peter Thiel*

## Agradecimientos

A mi familia pequeña, mi esposa Mirian y mis hijos Lucas y Sofía quienes tuvieron que renunciar a muchas cosas para acompañarme, literalmente, en todo momento.

A mi papá y mis hermanos, que aunque ya no vivamos juntos, siempre estamos conectados y sabemos que estamos ahí para lo que sea. Les tengo siempre conmigo.

A mi orientador Prof. Sergio Cardozo, que me tuvo paciencia durante todo este tiempo, y supo apuntar este trabajo hacia el rumbo correcto.

A la Facultad de Ingeniería, Universidad Nacional de Asunción, por organizar este tipo de maestrías, alineada con su tradición de excelencia en la formación de ingenieros en el Paraguay.

Al CONACYT por el apoyo económico indispensable para la realización de la maestría y por la disponibilización de la biblioteca CICCO, herramienta fundamental para la investigación.

A las autoridades y mis compañeros de trabajo de la Facultad de Ciencias y Tecnologías Universidad Nacional de Caaguazú, que comprenden muy bien la importancia de la capacitación y me cubrieron en momentos clave.

A toda la comunidad de software libre y todas las personas que están ahí afuera, ayudando con sus conocimientos en los foros de discusión, cursos, videos, etc. Todos ellos hacen que internet sea una bendición sin el cual este tipo de investigaciones sería absolutamente imposible.

---

## Resumen

Se utilizaron cuatro técnicas de aprendizaje automático, a saber, Regresión Lineal (LR), Vector Soporte Automático (SVM), Redes Neuronales Artificiales (ANN) y Aprendizaje de Máquina Extremo (ELM) para el análisis de la aplicabilidad de estas técnicas al estudio de vigas rectangulares de hormigón armado sometidas a flexión. Previamente se desarrollaron dos algoritmos, el primero para el cálculo del momento último resistente de una sección rectangular de hormigón armado y el segundo, para el cálculo de flechas utilizando el diagrama de momento - curvatura, ambos por el método de los estados límite de conformidad a los lineamientos del Eurocódigo N° 2. Con los citados algoritmos se generó una base de datos para el análisis de secciones, que considera la resistencia característica del acero  $f_{yk}$ , la del hormigón  $f_{ck}$ , la base  $b$  y la altura  $h$  de la sección transversal, el armado longitudinal de tracción  $A_{st}$  y el de compresión  $A_{sc}$ , así como también el momento último resistente de la sección  $M_u$ . Esta base de datos se sometió al análisis de componentes principales o PCA, con el cual se determinó el desacoplamiento entre variables que definen la geometría de los elementos de la sección transversal y las variables que definen la resistencia de los materiales, llegándose a un gráfico que permite la verificación de secciones transversales de vigas de hormigón armado. Posteriormente se utilizaron las técnicas de aprendizaje automático, citadas más arriba, para la predicción del momento último resistente de una sección dada. Se siguió el mismo procedimiento para el análisis de flechas, generándose una base de datos con sendos valores de  $f_{yk}$ ,  $f_{ck}$ ,  $b$ ,  $h$ , longitud  $L$ ,  $A_{st}$ ,  $A_{sc}$ , carga uniformemente distribuida  $q$  en toda la longitud de la viga y la flecha  $f$ , esta última calculada con el diagrama momento - curvatura, suponiendo la viga simplemente apoyada. Luego de los estudios, se rechazó el uso de LR por las predicciones poco fiables en el presente caso, encontrándose que las demás técnicas predicen bastante bien los resultados esperados, en especial el método de ANN. Se resalta el uso de ELM por su simplicidad matemática, eficiencia en las predicciones y bajo costo computacional. Finalmente se utilizó ANN para predecir la resistencia característica del hormigón a partir de la flecha. Se concluye que las técnicas de aprendizaje automático son potentes herramientas para el análisis de vigas y podrían utilizarse en estructuras en general.

---

## Abstract

Four machine learning techniques were used, namely, Linear Regression (LR), Support Vector Machine (SVM), Artificial Neural Networks (ANN) and Extreme Learning Machine (ELM) for the applicability analysis of these techniques to the study of rectangular reinforced concrete beams subjected to bending moments. Two algorithms were previously developed, the first one in order to obtain the ultimate bending moment of a reinforced concrete beam with rectangular cross-section, and the second one, to obtain beam deflections using the moment-curvature relationship, both by limit state method following the guidelines of the Eurocode No. 2. Using the aforementioned algorithms, a database for beam sectional analysis was generated, which considers the characteristic strength of steel  $f_{yk}$ , and concrete  $f_{ck}$ , base  $b$  and height  $h$  of the cross section, the area of longitudinal tensile reinforcement  $A_{st}$ , the compression reinforcement  $A_{sc}$ , as well as the ultimate bending moment  $M_u$ . This database was subjected to the Principal Components Analysis or PCA, in which the decoupling between variables that define the geometry of elements of the beam cross section and the variables that define the resistance of materials was determined, arriving at a graph that allows the verification of reinforced concrete beams cross sections. Subsequently, machine learning techniques, mentioned above, were used for predictions of ultimate moment bendings of given sections. The same procedure was performed for the analysis of deflections, generating a database with values of  $f_{yk}$ ,  $f_{ck}$ ,  $b$ ,  $h$ , length  $L$ ,  $A_{st}$ ,  $A_{sc}$ , uniformly distributed load  $q$  over the entire length of the beam and deflection  $f$ , the latter calculated using the moment - curvature relationship assuming a simply supported beam. After the studies, the use of LR was rejected due to the unreliable predictions in the present case, finding that the other techniques predict the expected results quite well, especially the ANN method. The use of ELM is highlighted by its mathematical simplicity, efficiency in predictions and low computational cost. Finally ANN was used to predict the characteristic strength of concrete from the beam's deflection. It is concluded that machine learning techniques are powerful tools for beam analysis and could be used in structures in general.

# Índice general

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>1</b>
1.1	Alcance del proyecto . . . . .	2
1.2	Justificación . . . . .	3
1.3	Objetivos . . . . .	3
1.3.1	Objetivo general . . . . .	3
1.3.2	Objetivos específicos . . . . .	4
1.4	Antecedentes . . . . .	4
1.4.1	Algoritmos de cálculo de hormigón armado . . . . .	4
1.4.2	Aprendizaje automático a estructuras de hormigón . . . . .	6
1.5	Metodología del trabajo . . . . .	7
1.5.1	Creación de base de datos y análisis . . . . .	7
1.5.2	Lenguaje de programación adoptado . . . . .	8
1.5.3	Normativa adoptada . . . . .	9
<b>2</b>	<b>HORMIGÓN ARMADO</b>	<b>10</b>
2.1	Consideraciones generales . . . . .	11
2.1.1	Hipótesis generales . . . . .	11
2.1.2	Definiciones importantes . . . . .	11
2.1.3	Verificación en estados límite último . . . . .	14
2.1.4	Verificación en estados límite de servicio . . . . .	14
2.1.5	Coeficientes parciales de seguridad de los materiales . . . . .	15
2.2	Tensiones normales . . . . .	15
2.2.1	Modelo constitutivo del Hormigón . . . . .	15
2.2.2	Diagrama adoptado . . . . .	21
2.2.3	Acero . . . . .	22
2.2.4	Cálculo en rotura . . . . .	23
2.2.5	Diagrama de interacción . . . . .	28
2.3	Estado límite de deformación . . . . .	29
2.3.1	Relación momento-curvatura . . . . .	30

<b>3</b>	<b>APRENDIZAJE AUTOMÁTICO</b>	<b>33</b>
3.1	Conceptos iniciales . . . . .	33
3.1.1	Definición . . . . .	33
3.1.2	Paradigma y justificación . . . . .	33
3.1.3	Tipos de sistemas de aprendizaje automático . . . . .	34
3.1.4	Ventajas y desventajas de los métodos adoptados . . . . .	35
3.2	Análisis de componentes principales . . . . .	38
3.2.1	Definiciones importantes . . . . .	38
3.2.2	El método PCA . . . . .	40
3.2.3	Propiedades de las componentes principales . . . . .	42
3.3	Modelos de entrenamiento . . . . .	42
3.3.1	Regresión lineal . . . . .	43
3.3.2	Vector soporte automático . . . . .	46
3.3.3	Redes neuronales artificiales . . . . .	56
3.3.4	Aprendizaje automático extremo . . . . .	63
<b>4</b>	<b>BASE DE DATOS</b>	<b>67</b>
4.1	Programas de computadora . . . . .	68
4.2	Comprobación de secciones . . . . .	69
4.2.1	Caracterización de los materiales . . . . .	69
4.2.2	Momento último . . . . .	71
4.3	Deformada de la viga . . . . .	72
4.3.1	Diagrama de momento-curvatura . . . . .	73
4.3.2	Algoritmo del diagrama momento-curvatura . . . . .	74
4.3.3	Cálculo de flechas . . . . .	76
4.3.4	Deformada con diagrama de momento-curvatura . . . . .	78
4.4	Contrastación de resultados . . . . .	81
4.4.1	Primera comparación . . . . .	82
4.4.2	Segunda comparación . . . . .	83
4.5	Generación de base de datos . . . . .	84
4.5.1	Base de datos para análisis de secciones . . . . .	84
4.5.2	Base de datos para análisis de deformaciones . . . . .	86
<b>5</b>	<b>ANÁLISIS DE SECCIONES</b>	<b>89</b>
5.1	Visión general de la base de datos . . . . .	89
5.2	Gráficas . . . . .	90
5.3	Análisis de componentes principales . . . . .	91
5.3.1	Estandarización de variables . . . . .	91

5.3.2	Matriz de correlaciones . . . . .	91
5.3.3	Valores característicos . . . . .	93
5.3.4	Vectores característicos . . . . .	94
5.3.5	Componentes principales . . . . .	94
5.3.6	Correlaciones entre componentes principales y variables originales . . . . .	95
5.3.7	Notas . . . . .	95
5.4	Decisiones sobre la base de datos . . . . .	96
5.4.1	Selección de datos de entrada y salida . . . . .	97
5.4.2	Normalización de los datos de entrada . . . . .	97
5.4.3	División en conjuntos de entrenamiento y verificación . . .	98
5.5	Entrenamiento . . . . .	98
5.5.1	Regresión lineal . . . . .	98
5.5.2	Vector soporte automático . . . . .	99
5.5.3	Redes neuronales artificiales . . . . .	99
5.5.4	Aprendizaje automático extremo . . . . .	99
5.6	Verificación . . . . .	100
5.6.1	Regresión lineal . . . . .	100
5.6.2	Vector soporte automático . . . . .	100
5.6.3	Redes neuronales . . . . .	101
5.6.4	Aprendizaje automático extremo . . . . .	102
5.7	Comprobación de casos particulares . . . . .	103
<b>6</b>	<b>ANÁLISIS DE DEFORMACIONES</b>	<b>104</b>
6.1	Visión general de la base de datos . . . . .	104
6.2	Gráficas . . . . .	105
6.3	Análisis de componentes principales . . . . .	106
6.3.1	Estandarización de las variables de entrada . . . . .	106
6.3.2	Matriz de correlaciones . . . . .	107
6.3.3	Valores característicos . . . . .	107
6.3.4	Vectores característicos . . . . .	108
6.3.5	Componentes principales . . . . .	109
6.3.6	Correlaciones entre componentes principales y variables originales . . . . .	109
6.4	Decisiones sobre la base de datos . . . . .	110
6.4.1	Normalización de los datos de entrada . . . . .	110
6.4.2	División en conjuntos de entrenamiento y verificación . . .	110
6.5	Entrenamiento . . . . .	111

6.5.1	Regresión lineal . . . . .	111
6.5.2	Vector soporte automático . . . . .	111
6.5.3	Redes neuronales . . . . .	111
6.5.4	Aprendizaje automático extremo . . . . .	112
6.6	Verificación . . . . .	112
6.6.1	Regresión lineal . . . . .	112
6.6.2	Vector soporte automático . . . . .	113
6.6.3	Redes neuronales . . . . .	113
6.6.4	Aprendizaje automático extremo . . . . .	114
6.7	Comprobación de casos particulares . . . . .	115
6.8	Selección del método y cálculos adicionales . . . . .	115
6.8.1	Predicción de la resistencia del hormigón a partir de la flecha	116
<b>7</b>	<b>CONCLUSIONES</b>	<b>118</b>
7.1	Análisis de componentes principales . . . . .	118
7.2	Predicciones de momento último . . . . .	119
7.2.1	Regresión lineal . . . . .	119
7.2.2	Vector soporte automático . . . . .	119
7.2.3	Redes neuronales . . . . .	120
7.2.4	Aprendizaje automático extremo . . . . .	120
7.3	Predicciones de flecha . . . . .	120
7.3.1	Vector soporte automático . . . . .	120
7.3.2	Redes neuronales artificiales . . . . .	121
7.3.3	Aprendizaje automático extremo . . . . .	121
7.4	Conclusiones finales . . . . .	121
7.5	Recomendaciones y trabajos futuros . . . . .	122
	<b>APÉNDICES</b>	<b>124</b>
<b>A</b>	<b>Módulos para hormigón armado</b>	<b>125</b>
A.1	Manual de uso: materiales.py . . . . .	125
A.1.1	Clase: Hormigón . . . . .	125
A.1.2	Clase: Acero . . . . .	127
A.2	Manual de uso: vigas.py . . . . .	129
A.2.1	Clase: Viga . . . . .	129
<b>B</b>	<b>Módulos para aprendizaje automático</b>	<b>134</b>
B.1	Generación de base de datos . . . . .	134
B.1.1	Comprobación de secciones . . . . .	134

---

B.1.2	Flechas de vigas simplemente apoyadas . . . . .	136
B.2	Módulo: mielm.py . . . . .	140
B.2.1	Atributos . . . . .	140
B.2.2	Propiedades derivadas . . . . .	141
<b>REFERENCIAS BIBLIOGRÁFICAS</b>		<b>143</b>

# Índice de figuras

2.1	Curva típica de tensión-deformación del hormigón a compresión. . .	16
2.2	Diagramas tensión-deformación del hormigón . . . . .	20
2.3	Diagramas de tracción de Wang & Hsu. . . . .	21
2.4	Diagrama de hormigón adoptado . . . . .	22
2.5	Diagrama Tensión-Deformación del acero . . . . .	23
2.6	Comportamiento de una sección de hormigón armado. . . . .	24
2.7	Diagrama de pivotes . . . . .	26
2.8	Ejemplo de diagrama de interacción . . . . .	29
2.9	Diagrama de momento-curvatura típico . . . . .	31
3.1	Datos con tendencia lineal . . . . .	43
3.2	Predicciones con el modelo de regresión lineal . . . . .	45
3.3	Datos linealmente separables . . . . .	46
3.4	Criterios del SVM . . . . .	49
3.5	Arquitectura básica de un PERCEPTRÓN. . . . .	57
3.6	Arquitectura del PERCEPTRÓN multicapa. . . . .	58
3.7	Arquitectura del SLFN . . . . .	64
4.1	Gráfico tensión-deformación de cálculo del hormigón . . . . .	70
4.2	Diag. de interacción de una sección de hormigón armado . . . . .	72
4.3	Elemento de viga con 4 grados de libertad . . . . .	73
4.4	Algoritmo del diagrama momento-curvatura . . . . .	75
4.5	Diagrama momento-curvatura de una sección de hormigón con di- ferentes armados . . . . .	76
4.6	Comparación de flechas . . . . .	78
4.7	Viga de ejemplo para cálculo de su deformada . . . . .	79
4.8	Diagramas de momento flector, de curvaturas y deformada de una viga . . . . .	81
4.9	Viga tipo para primera comparación . . . . .	82
4.10	Primera comparación de resultados . . . . .	82
4.11	Viga tipo para segunda comparación . . . . .	83

4.12	Segunda comparación de resultados . . . . .	83
5.1	Relación momento último - áreas de armaduras . . . . .	91
5.2	Relación entre algunas variables . . . . .	92
5.3	Momentos últimos en función de componentes principales . . . . .	96
5.4	Bondad de ajuste con el método de regresión lineal . . . . .	100
5.5	Bondad de ajuste con el método vector soporte automático . . . . .	101
5.6	Ajuste con método ANN - capacidad de secciones . . . . .	101
5.7	Distribución conjunta de momentos últimos por ELM . . . . .	102
6.1	Nube de puntos, base de datos de flechas . . . . .	106
6.2	Bondad de ajuste - método de regresión lineal para flechas . . . . .	112
6.3	Ajuste con método SVR para cálculo de flechas . . . . .	113
6.4	Ajuste con método ANN para cálculo de flechas . . . . .	114
6.5	Ajuste con método ELM para cálculo de flechas . . . . .	114
6.6	Estimación de $f_{ck}$ a partir de la flecha . . . . .	117

# Lista de tablas

2.1	Coeficientes parciales para mayoración de acciones. . . . .	13
2.2	Coeficientes de seguridad en ELU . . . . .	15
2.3	Relaciones tensión-deformación del hormigón para análisis no lineal	17
3.1	Funciones kernel más utilizadas . . . . .	53
4.1	Muestra de base de datos para análisis de secciones . . . . .	86
4.2	Muestra base de datos para análisis de deformaciones . . . . .	88
5.1	Estadísticos de variables de secciones . . . . .	90
5.2	Porcentaje explicado por cada componente principal . . . . .	94
5.3	Correlaciones entre variables . . . . .	95
5.4	Cambio de capacidad de sección ante cambios de variables . . . . .	96
5.5	Base de datos de secciones con $f_{yk} = 400$ MPa . . . . .	97
5.6	Base de datos de secciones con $f_{yk} = 500$ MPa . . . . .	97
5.7	Predicciones de momento último por diferentes métodos . . . . .	103
6.1	Estadísticos de variables, análisis flechas . . . . .	105
6.2	Porcentaje explicado - flechas . . . . .	108
6.3	Correlaciones entre variables para cálculo de flechas . . . . .	109
6.4	Comprobación de flecha en casos particulares . . . . .	115

# Lista de símbolos

## SIGLAS

- AI *Artificial Intelligence*. Inteligencia artificial
- ANN *Artificial Neural Network*. Redes neuronales artificiales
- ELM *Extreme Learning Machine*. Aprendizaje automático extremo
- ELS Estado Límite de Servicio
- ELU Estado Límite Último
- HA Hormigón Armado
- LN Línea neutra
- LR *Linear Regression*. Regresión lineal
- LS-SVM *Least Square - Support Vector Machine*. Mínimos cuadrados de vector soporte automático
- ML *Machine Learning*. Aprendizaje automático
- MSE *Minimum Square Error*. Error medio cuadrático
- PCA *Principal Component Analysis*. Análisis de componentes principales
- RMSE *Root Mean Square Error*. Raíz del error medio cuadrático
- SI Sistema Internacional de Unidades
- SLFN *Single-hidden Layer Feedforward Network*. Red de avance de una sola capa oculta
- SVM *Support Vector Machine*. Vector soporte automático

**HORMIGÓN ARMADO**

$\alpha_{cc}$	Factor de cansancio del hormigón a compresión
$\alpha_{ct}$	Factor de cansancio del hormigón a tracción
$\bar{\kappa}$	Curvatura promedio
$\bar{M}$	Momento flector promedio
$\eta$	Utilizado en la ecuación de tensión-deformación del hormigón
$\gamma_f$	Coefficiente parcial de mayoración de una acción
$\gamma_c$	Coefficiente de seguridad para el hormigón
$\gamma_s$	Coefficiente de seguridad del acero
$\kappa$	Curvatura
$\psi_0$	Coefficiente de combinación de una acción variable
$\psi_1$	Coefficiente para valor frecuente de una acción variable
$\psi_2$	Coefficiente para valor cuasi-permanente de una acción variable
$\rho_s$	Densidad del acero
$\sigma_c$	Tensión en hormigón
$\sigma_s$	Tensión en acero
$\sigma_t$	Tensión de tracción
$\varepsilon$	Deformación unitaria
$\varepsilon_{cr}$	Deformación unitaria crítica a tracción del hormigón
$\varepsilon_{cu1}$	Deformación unitaria última del hormigón
$\varepsilon_{cu}$	Deformación unitaria última del hormigón
$\varepsilon_c$	Deformación unitaria del hormigón
$\varepsilon_{su}$	Deformación unitaria máxima permitida al acero en el interior del hormigón
$\varepsilon_t$	Deformación unitaria a tracción del hormigón

---

$\varepsilon_{ud}$	Deformación unitaria última de diseño del acero
$\varepsilon_{uk}$	Deformación unitaria característica del acero en fuerza máxima
$\varepsilon_y$	Deformación unitaria del acero, correspondiente al límite elástico
$A$	Acción accidental
$A_s$	Área de armaduras
$b$	Base de sección rectangular
$C$	Resultante de fuerzas en hormigón
$d$	Posición de armadura respecto a fibra superior de una sección
$d_1$	Canto útil, distancia desde la fibra superior hasta el centro de áreas de las armaduras de tracción
$E$	Efecto de la acción
$E_{cm}$	Módulo de elasticidad (secante) medio del hormigón
$E_s$	Módulo de elasticidad del acero
$F$	Acción sobre una estructura
$F_k$	Valor característico de una acción
$f_y$	Tensión del acero en su límite elástico
$f_{cd}$	Resistencia de cálculo del hormigón
$f_{ck}$	Resistencia característica del hormigón
$f_{cm}$	Resistencia media a la compresión del hormigón a los 28 días
$f_{ctm}$	Resistencia media del hormigón a tracción uniaxial
$F_d$	Valor de cálculo de una acción
$f_{yd}$	Resistencia de cálculo del acero
$f_{yk}$	Resistencia característica del acero
$f_{ym}$	Resistencia media del acero
$G$	Acción permanente

---

$h$	Altura de sección rectangular
$k$	Factor de la ecuación de tensión-deformación del hormigón EC2
$k$	Relación entre la tensión máxima de rotura y el límite elástico del acero
$k_t$	Coefficiente de corrección de los factores de cansancio del hormigón
$M$	Momento flector real
$M$	Momento flector
$m$	Momento flector virtual
$M_u$	Momento flector último resistente de una sección de HA
$N_u$	Fuerza normal última resistente de una sección de HA
$Q$	Acción variable
$T$	Resultante de fuerzas en acero
$tol$	Tolerancia
$v$	Deflexión vertical
$x$	Coordenada de un punto del eje de una viga
$x$	Profundidad de línea neutra
$x_{lim}$	Profundidad máxima de la línea neutra en dominio 3
$y$	Posición de una fibra cualquiera de una sección respecto a la línea neutra

### APRENDIZAJE AUTOMÁTICO

$\alpha$	Factor de escala para el cálculo del número de neuronas
$\alpha_i$	Multiplicadores de Lagrange
$\bar{x}$	Media muestral
$\mathbf{1}_v$	Vector de unos
$\beta$	Matriz de pesos de salida
$\epsilon$	Vector error

---

$\Lambda$	Matriz diagonal de valores característicos
$\hat{a}$	VARIABLES DE ENTRADA NORMALIZADAS
$\hat{y}$	Estimador de $y$
$\mathbf{A}$	Matriz principal de un sistema de ecuaciones lineales
$E[X]$	Esperanza matemática de la variable aleatoria $X$
$\mathbf{H}$	Matriz de salida de la capa oculta
$\mathbf{H}^+$	Pseudoinversa de Moore-Penrose
$\mathbf{Q}$	Matriz ortogonal para descomposición espectral de la matriz de covarianzas
$\mathbf{R}$	Matriz de correlaciones
$\mathbf{S}$	Matriz de covarianzas
$\mathbf{X}$	Vector de variables aleatorias
$\mathbf{Y}$	Matriz de valores de salida
$\mathbf{Z}$	Vector de componentes principales
$\mu$	Media de la población
$\rho(X, Y)$	Coefficiente de correlación entre las variables aleatorias $X$ e $Y$
$\text{sgn}(x)$	Función signo de $x$
$\sigma_X$	Desviación estándar de la variable aleatoria $X$
$b$	Bias
$c$	Índice asignado a la capa de salida
$C(A)$	Espacio columna de $A$
$d_{\mathbf{x} \rightarrow r}$	Distancia del punto $\mathbf{x}$ a la recta $r$
$G(\mathbf{x})$	Función de decisión
$i, j, k, d$	Índices
$L$	Número de neuronas en la capa oculta (método ELM)

---

$L_D$	Función dual de Wolfe
$L_P$	Función primal de Lagrange
$m$	Número de variables de salida
$N$	Número de puntos. Tamaño del conjunto de entrenamiento
$n$	Número de variables de entrada
$N(A)$	Espacio nulo de A
$N(A^T)$	Espacio nulo izquierdo de A
$n_0$	Número de neuronas en la capa de entrada
$n_h$	Número de neuronas en la capa oculta
$n_k$	Número de neuronas de la capa $k$
$p_X(x)$	Probabilidad de que la variable aleatoria $X$ asuma el valor $x$
$s$	Desviación estándar muestral
$s^2$	Varianza muestral
$w_i$	Coefficientes de combinación lineal
$X$	Variable aleatoria
$x$	Valor que puede asumir una variable aleatoria $X$
$x_j$	Variables independientes
$y$	Variable dependiente
NE	Tamaño del conjunto de entrenamiento
NV	Tamaño del conjunto de verificación

# Capítulo 1

## INTRODUCCIÓN

Los métodos de aprendizaje automático han demostrado ser muy eficientes en muchas ramas de la ciencia, llamó la atención del autor el hecho que aún están siendo poco explorados en ingeniería civil. Son muy interesantes los avances que se están logrando utilizando estas técnicas. Con los teléfonos inteligentes, por ejemplo, se convirtieron en temas de uso cotidiano, ya que están incorporados en varios programas de uso frecuente, como ser: reconocimiento de voz, sugerencias de palabras, traducciones a otros idiomas, recomendaciones de productos, la mejor ruta para llegar a un lugar, por citar nada más algunos usos. En todos ellos, las novedades se han encontrado, no por resolver precisamente un problema nuevo, (la mayoría de estos problemas ya estaban resueltos, aunque de una manera poco práctica), sino enfrentando el mismo problema con nuevas técnicas que avizoran nuevos caminos y puertas que abrir en el desarrollo del conocimiento humano y, por supuesto, descubriendo mejores maneras de hacer las cosas.

Por lo dicho precedentemente, no se pretende encontrar una teoría original para el análisis, sino utilizar las técnicas de aprendizaje automático, de manera a mirar desde otro ángulo el problema, en este caso, flexión en vigas rectangulares de hormigón armado, con lo cual, se espera abrir nuevos horizontes en el área de la ingeniería estructural.

## 1.1. Alcance del proyecto

Se realiza el análisis de vigas de hormigón armado con sección rectangular sometidas a flexión, utilizando técnicas de aprendizaje automático. El alcance del trabajo se expresa en los siguientes puntos:

En relación al objeto de estudio:

1. Se estudian vigas sometidas a flexión únicamente, se descartan otros tipos de esfuerzos;
2. El elemento de análisis es la viga de hormigón armado con dos grados de libertad por nudo (desplazamiento vertical y giro);
3. Se consideran diferentes secciones transversales, longitudes, armados y resistencias de sus materiales constitutivos, en todos los casos buscando mantener los valores en los rangos usuales del mercado de la construcción en Paraguay;
4. La sección transversal es rectangular y constante a lo largo de toda la viga;
5. Para el análisis de flechas, se consideran solamente vigas simplemente apoyadas con cargas uniformemente distribuidas en toda su longitud;
6. Se asume que el armado de la viga es constante en toda su longitud y tiene solo dos capas de armado, una a tracción y otra a compresión<sup>1</sup>.

En relación al aprendizaje automático:

1. Los enfoques utilizados son del tipo supervisado;
2. El análisis de la base de datos se realiza con el análisis de componentes principales, en adelante PCA.
3. Las técnicas de aprendizaje automático a ser utilizadas son:
  - Regresión lineal, en adelante LR;
  - Vector soporte automático, en adelante SVM;
  - Redes neuronales artificiales, en adelante ANN;
  - Aprendizaje automático extremo, en adelante ELM.

---

<sup>1</sup>Debidos a la flexión.

## 1.2. Justificación

El aprendizaje automático o, en general, la inteligencia artificial (AI) está modificando la manera de trabajar de las empresas, y está transformando la forma en que las personas interactúan con el mundo, sus inicios se remontan hace bastante tiempo pero es en la actualidad donde se refleja su verdadero potencial y utilidad. Las empresas de construcción, y la ingeniería estructural en particular, no deben estar ajenas a estas nuevas maneras de abordar los problemas, que son aplicables a muchas áreas de la ciencia y la ingeniería.

Se ha notado aún pocos desarrollos con la utilización de estas técnicas en el análisis de estructuras, lo cual es la motivación principal para la realización de este trabajo. La mecánica computacional ha logrado enormes avances en el análisis de estructuras de todo tipo, el autor considera que el presente es un paso natural, esto es, el uso de métodos de aprendizaje automático en el estudio de vigas de hormigón armado para posteriormente ampliar a casos de estructuras en general.

La diferencia respecto a otros métodos es que, en realidad, no se pretende resolver el problema, en este caso, de dimensionamiento o verificación de vigas de hormigón armado, el mismo ya está resuelto, sin embargo, se hace notar que el núcleo del aprendizaje automático, está justamente en la palabra *aprendizaje*, la cual se refiere a analizar una gran cantidad de información o datos y “aprender” de los mismos, con lo cual, adicionalmente al hecho de resolver un problema utilizando otro enfoque se puede: brindar luz adicional en el fenómeno estudiado, resolver el problema de una forma más eficiente y responder preguntas aún no resueltas e incluso aún no planteadas.

## 1.3. Objetivos

### 1.3.1. Objetivo general

Aplicar técnicas de aprendizaje automático para el análisis a flexión simple de vigas rectangulares de hormigón armado.

### 1.3.2. Objetivos específicos

1. Desarrollar un algoritmo para el dimensionamiento y verificación a flexión simple aplicable a vigas de hormigón armado;
2. Obtener una base de datos de secciones transversales de vigas rectangulares de hormigón armado, que relacione sus parámetros definatorios con su capacidad resistente a flexión;
3. Obtener una base de datos de flechas en diferentes situaciones para vigas rectangulares de hormigón armado simplemente apoyadas y sometidas a cargas uniformemente distribuidas en toda su longitud;
4. Utilizar métodos de aprendizaje automático para el análisis de los conjuntos de datos anteriores;
5. Seleccionar la técnica de aprendizaje automático más adecuada y de menor gasto computacional;
6. Aplicar el aprendizaje automático para el análisis directo e inverso en flexión simple de vigas rectangulares de hormigón armado.

## 1.4. Antecedentes

El presente trabajo involucra dos áreas bien definidas: Algoritmos para cálculo de vigas de hormigón armado y aprendizaje automático. Se comenta a continuación algunos trabajos que sirven de antecedente y que serían los más próximos al presente, con un breve comentario explicativo del alcance de cada uno de ellos.

### 1.4.1. Algoritmos de cálculo de hormigón armado

1. **Soares (2001)** presenta su tesis “Un estudio sobre modelos mecánicos – probabilistas para pórticos de hormigón armado” en el cual el proyecto estructural considera la seguridad a través de métodos probabilistas, lo que permite estimar la probabilidad de falla de pórticos de hormigón armado para diversos estados límites. Indica brevemente un camino a seguir para lograr la optimización acoplada al problema mecánico – probabilista.

2. **Velame Branco (2002)** en su “Análisis no lineal de pórticos planos, considerando los efectos de cizallamiento en los cálculos de esfuerzos y desplazamientos” desarrolla un algoritmo y lo implementa en FORTRAN, basado en el método de los elementos finitos, considerando el análisis no lineal físico y geométrico de pórticos planos de hormigón armado. Agrega el efecto de corte en la matriz de rigidez, es decir adopta el modelo de Timoshenko (1878-1972). Es interesante también su aporte en el modelo físico adoptado, basado en el modelo de daño de Mazars (1984).
3. **Nogueira (2005)**, en su disertación de maestría propone “Un modelo de confiabilidad y optimización aplicado a estructuras de barras de hormigón armado” en el cual aporta un método para acoplar el problema de confiabilidad al de optimización, esto es, la confiabilidad es considerada como una restricción más en el proceso de optimización, visto como minimización de los costos de la estructura. El cálculo de esfuerzos y desplazamientos se basa Branco (2002).
4. **Payá Zaforteza (2009)** estudia la “Optimización heurística de pórticos de edificación de hormigón armado” en donde presenta el diseño de algoritmos que permitan automatizar el diseño óptimo de pórticos de hormigón armado y extraer conclusiones generales sobre las estructuras optimizadas”. Realiza un muy buen trabajo de comparación de varios métodos de optimización aplicado sobre un pórtico de dos vanos y cuatro plantas sometido a acciones verticales y horizontales, como: Saltos Múltiples Aleatorios (RW), Gradiente First Best (FB), Cristalización Simulada (SA), Aceptación por Umbrales (TA) y Algoritmos Genéticos (GA), con el cual adopta el SA, por obtener con ese método mejores resultados, para posteriormente utilizarlo en otros pórticos de características diferentes. Además demuestra que no es posible encontrar un diseño de optimización simultánea del costo de la estructura, facilidad constructiva, sostenibilidad ambiental y seguridad.
5. **Nina, 2006**, presenta su disertación titulada “Optimización de secciones transversales de hormigón armado: aplicación a pórticos planos”, en el cual desarrolla una rutina computacional para optimizar el predimensionamiento de edificios de hormigón armado tratados simplificadaamente como pórticos planos. No toma en cuenta los esfuerzos de corte. Utiliza la Norma Brasileira y el método de programación cuadrática sucesiva para resolver el problema general de programación no lineal.

6. **Cardozo López (2009)**, presenta su trabajo “Optimización de placas y cáscaras de materiales compuestos, utilizando algoritmos genéticos, redes neuronales y elementos finitos” buscando la implementación de una herramienta computacional capaz de optimizar estructuras complejas fabricadas de esos materiales, a un bajo costo computacional.
7. **Nogueira (2010)** generaliza su trabajo del (2005) con el “Desarrollo de modelos mecánicos, de confiabilidad y de optimización para aplicación en estructuras de hormigón armado”. En ambos trabajos solo son considerados los costos iniciales de la estructura.
8. **Gomes Pedrosa (2015)** aborda la optimización de sistemas estructurales simples, considerando el equilibrio entre modos de falla concurrentes. El estudio muestra cómo los diferentes modos de falla, asociados a sus respectivos costos de falla, llevan a diferentes proyectos óptimos. Los resultados muestran que proyectos óptimos suficientemente diferentes son obtenidos cuando el equilibrio entre modos de falla concurrentes es alterado.

#### 1.4.2. Aprendizaje automático a estructuras de hormigón

1. **Aggarwal (2007)** realiza análisis de aprendizaje automático para predecir el armado de una viga de hormigón, armada en una sola capa, utiliza Redes Neuronales Artificiales (ANN), Regresión lineal (LR), Vectores de Soporte Automático (SVM) y Árboles Modales (MT). Las variables consideradas fueron: las dimensiones de la viga ( $b, h, L$ ), la carga  $q = cte$ , (considerada como uniformemente distribuida en toda la longitud de la viga), el momento flector  $M_f$  y la cuantía de armado  $\rho$ . Su base de datos contaba con 130 casos. Encontró los mejores desempeños en los modelos con MT y SVM.
2. **Preetham S y M (2015)** utilizan ANN y SVM para predecir la deflexión del punto medio de una viga simplemente apoyada bajo la acción de dos cargas concentradas. Las variables consideradas fueron:  $b, h, L, f_{ck}, f_{yk}, P_{x=a}, P_{x=b}, \delta$ . Su base datos estaba conformada por 18 casos de estudios experimentales. Encontraron el mejor desempeño en SVM.
3. **Imam y Kazmi (2017)** configuran modelos de regresión modificada y ANN para predecir la capacidad de carga de vigas de hormigón armado corroídas, con el objetivo de encontrar un método que mejore substancial-

mente a predicción de esfuerzos residuales. Su base de datos contaba con 48 casos. Encontraron un buen desempeño con redes neuronales (ANN).

4. **Rafiei y col. (2016)** utilizan técnicas de inteligencia artificial para estimar propiedades del hormigón, como resistencia, adherencia, fluencia, retracción y serviciabilidad con datos previamente colectados. Presentan el estado del arte en este aspecto, con revisiones de modelos estadísticos, reconocimiento de patrones, aprendizaje automático, algoritmos evolutivos, y aproximaciones híbridas.
5. **Cevik y col. (2014)** revisan el estado del arte de la aplicación de la técnica Vector Soporte Automático a la ingeniería estructural e investigan la usabilidad de este enfoque basado en el aprendizaje automático aplicado a tres estudios de caso que se centran en problemas de ingeniería estructural. Sus resultados muestran que esta técnica es superior a varias otras considerando la capacidad de generalización que produce el modelo.

## 1.5. Metodología del trabajo

### 1.5.1. Creación de base de datos y análisis

Para aplicar las técnicas de aprendizaje automático supervisado al estudio de vigas de hormigón armado (o a cualquier otro tema) es necesario un conjunto de datos suficientemente amplio de manera a ser utilizado en la etapa de entrenamiento y luego de verificación.

Como se ha visto en la sección 1.4, la base de datos disponible en los diferentes estudios es muy limitada. El propósito de este trabajo es considerar la mayor cantidad de datos posibles, por lo que se ha optado en generar una base de datos propia para este estudio, utilizando las teorías y recomendaciones normativas para el dimensionamiento y verificación de estructuras de hormigón armado, en particular adoptamos el Eurocódigo 2 (UNE-EN:1992, 2004) para este efecto. Esto, sin embargo, implica un desafío no menor, ya que incluso utilizando softwares comerciales para la generación de los datos buscados, es decir, para el relacionamiento de las variables de entrada con las de salida, tomará bastante tiempo, por lo tanto, se adopta la siguiente metodología:

1. **Desarrollo de un algoritmo para el cálculo del momento último de una sección transversal dada:** De esta forma se relacionarán las variables de base, altura y armado con la capacidad resistente de la sección.
2. **Desarrollo de un algoritmo para el cálculo de flechas:** Conforme al cuál se tendrá la base de datos de flechas para diferentes longitudes, cargas, secciones y armado de vigas<sup>2</sup>;
3. **Entrenamiento de los diferentes métodos de aprendizaje automático** utilizando las bases de datos generadas en los dos puntos anteriores;
4. **Análisis de los resultados:** para determinar el método de aprendizaje automático más efectivo y aplicable al estudio de vigas rectangulares de hormigón armado;
5. **Cálculos adicionales:** con los que se pretende responder cuestiones de interés práctico.

### 1.5.2. Lenguaje de programación adoptado

Los algoritmos o programas de computadora se desarrollaron en `python 3` con sus paquetes científicos.

Python es language de programación de propósito general que puede ser utilizado efectivamente para construir casi cualquier clase de programas que no necesita acceso directo al hardware de la computadora (Guttag, 2013).

Probablemente Python es el lenguaje de programación más sencillo de aprender y fácil de usar. El código Python es claro tanto de leerlo como de escribirlo, y es preciso sin ser complicado. Python es un lenguaje muy expresivo, lo que significa que, por regla general, se puede escribir menos líneas de código Python que si se estuviese escribiendo la misma aplicación en, por decir, C++ o Java. Uno de los puntos fuertes de `python` es que viene junto a una librería estándar muy completa, además de miles de librerías de terceros que proporcionan utilidades más potentes y sofisticadas, como la librería numérica `Numpy` y la librería simbólica `Simpy` (Summerfield, 2010).

---

<sup>2</sup>Se tomarán en cuenta solo vigas simplemente apoyadas con cargas uniformemente distribuidas en toda su longitud.

A parte de ser software libre, es el lenguaje de programación de mayor crecimiento en los últimos años, según el índice TIOBE (2019).

A la pregunta de ¿por qué Python? Raschka (2015) responde:

Es potente y a la vez muy accesible. Python se ha convertido en el más popular lenguaje de programación para ciencia de datos porque permite olvidarnos de la parte tediosa de la programación y nos ofrece un ambiente donde podemos, rápidamente, poner nuestras ideas y conceptos en acción.

### 1.5.3. Normativa adoptada

Se opta por el conjunto de normas colectivamente denominadas *Eurocódigos*, los cuales fueron elaborados con el objeto de eliminar las disparidades entre los estados miembros de la Unión Europea, y lograr niveles uniformes de seguridad en la construcción. (CEN, 2019).

Los Eurocódigos EN que se utilizan en este trabajo son:

- Bases del diseño estructural (UNE-EN:1990, 2002);
- Acciones en las estructuras (UNE-EN:1991, 2002);
- Proyecto de estructuras de hormigón (UNE-EN:1992, 2004).

## Capítulo 2

# HORMIGÓN ARMADO

El hormigón armado (HA) es un material compuesto muy utilizado en obras de ingeniería, tanto que pareciera los ingenieros ya tienen resuelto el problema de comprender la respuesta del hormigón armado a los diferentes requerimientos estructurales. La realidad es que el conocimiento no es acabado, con lo cual aún es tema de muchas investigaciones, sin embargo suficiente como para utilizarlo en la mayoría de las obras que son actualmente construidas. El hormigón es un material muy complejo, un abordaje completo del mismo es muy difícil de llevar adelante, los diferentes fenómenos que aparecen ante las cargas aplicadas, las condiciones ambientales, las condiciones de confinamiento, etc. son muchos, a veces simples pero no pocas veces bastante difíciles de reproducir con un modelo matemático. Esto recuerda la frase de Feynman (1964) referida a la mecánica cuántica “*Si usted piensa que entiende a la mecánica cuántica, entonces usted no entiende a la mecánica cuántica*”, es probable que se pueda decir lo mismo del hormigón armado.

Por lo tanto es algo forzoso tener que tomar en cuenta solo algunos fenómenos presentes en el hormigón armado y dejar de considerar otros. A pesar de lo dicho en el párrafo anterior, reiteramos que el conocimiento actual del hormigón armado es suficientemente bueno para realizar maravillosas obras.

## 2.1. Consideraciones generales

### 2.1.1. Hipótesis generales

Las consideraciones de carácter general asumidas en UNE-EN:1990 (2002) son:

- Que la elección del sistema estructural y del procedimiento de cálculo de la estructura ha sido hecha por un personal debidamente cualificado y experimentado;
- Que la ejecución se lleva a cabo por personal que tenga la debida habilidad y experiencia;
- Que durante la ejecución de la obra se dispone de adecuada supervisión y control de calidad, es decir, en oficinas de proyecto, fábricas, plantas y en la obra;
- Que los materiales y productos de construcción se usan según se indican en las normas o las especificaciones de materiales o productos de referencia;
- Que la estructura se va a mantener adecuadamente;
- Que la estructura será utilizada de acuerdo con las consideraciones de cálculo asumidas.

### 2.1.2. Definiciones importantes

Estas definiciones son dadas en UNE-EN:1990 (2002) y son importantes en este estudio:

- **Estados límite:** Estados más allá de los cuales la estructura deja de satisfacer los criterios de proyecto correspondientes;
- **Estados límites últimos (ELU):** Estados asociados al colapso u otras formas de fallo estructural;
- **Estados límite de servicio (ELS):** Estados que corresponden a condiciones más allá de las cuales no se cumplen los requisitos de aptitud al servicio especificados para una estructura o elemento estructural.

- **Capacidad portante:** Capacidad de un elemento, o componente, o de una sección transversal de un elemento o componente de una estructura de soportar acciones sin fallos mecánicos, por ejemplo, resistencia a la flexión, resistencia al pandeo, resistencia a la tracción;
- **Resistencia:** Propiedad mecánica de un material que indica su habilidad de resistir acciones, normalmente dadas en unidades de tensión;
- **Acción ( $F$ ):** Conjunto de fuerzas (cargas) aplicadas a la estructura;
- **Efecto de la acción ( $E$ ):** Efecto de las acciones (o efecto de la acción) sobre los elementos estructurales (por ejemplo, fuerza interna, momento, tensión, deformación) o sobre la totalidad de la estructura (por ejemplo, flexión, rotación);
- **Acción permanente ( $G$ ):** Acción que previsiblemente va a actuar durante un período de referencia dado y cuya variación de magnitud en el tiempo es despreciable, o cuya variación es siempre en la misma dirección (monotónica) hasta que la acción alcance un cierto valor límite;
- **Acción variable ( $Q$ ):** Acción cuya variación de magnitud en el tiempo no es despreciable ni monotónica;
- **Acción accidental ( $A$ ):** Acción, normalmente de corta duración pero de magnitud significativa, que no es previsible que ocurra en una estructura dada durante su vida útil de cálculo;
- **Valor característico de una acción ( $F_k$ ):** Principal valor representativo de una acción;
- **Periodo de referencia:** Periodo de tiempo elegido que se usa como base para valorar estadísticamente acciones variables y, posiblemente, acciones accidentales.
- **Valor de combinación de una acción variable ( $\psi_0 Q$ ):** *Valor escogido* -en tanto en cuanto pueda fijarse sobre bases estadísticas- de forma que la probabilidad de los efectos causados por la combinación vayan a ser excedidos sea aproximadamente la misma que los causados por el valor característico de una acción concreta. Puede expresarse como una parte determinada del valor característico empleando un factor  $\psi_0 \leq 1$ ;

- **Valor frecuente de una acción variable** ( $\psi_1 Q$ ): *Valor determinado* - en tanto en cuanto pueda fijarse sobre bases estadísticas- de forma que bien el tiempo total, dentro del período de frecuencia durante el cual esté superado, sea solo una parte pequeña dada del periodo de referencia, o bien que la frecuencia con la que se exceda esté limitada a un valor dado. Puede expresarse como una parte determinada del valor característico empleando un factor  $\psi_1 \leq 1$ ;
- **Valor cuasi-permanente de una acción variable** ( $\psi_2 Q$ ): Valor determinado de manera que el periodo total de tiempo durante el cual vaya a ser superado sea una fracción grande del período de referencia. Puede expresarse como una parte determinada del valor característico empleando un factor  $\psi_2 \leq 1$ ;
- **Valor representativo de una acción** ( $F_{rep}$ ): Valor utilizado para la verificación de un estado límite. Un valor representativo puede ser un valor característico ( $F_k$ ) o un valor asociado ( $\psi F_k$ )<sup>1</sup>;
- **Valor de cálculo de una acción** ( $F_d$ ): Valor obtenido al multiplicar el valor representativo por el coeficiente parcial  $\gamma_f$  ;

Los valores de  $\gamma_f$  que se utilizan, de conformidad a UNE-EN:1990, 2002, Tabla A.1.2(B) son:

Tipo de acción	$\gamma_f$
Permanente	1.35
Variable	1.50

**Tabla 2.1** – Coeficientes parciales para mayoración de acciones.

- **Análisis estructural**: procedimiento o algoritmo para la determinación de los efectos de las acciones en cada punto de la estructura;
- **Análisis no lineal de primer orden**: Análisis estructural, llevado a cabo sobre la geometría inicial, que tiene en cuenta las propiedades de deformación no lineal de los materiales.

---

<sup>1</sup>Los valores de los coeficientes  $\psi_0$ ,  $\psi_1$  y  $\psi_2$  se dan en UNE-EN:1990, 2002, Tabla A.1.1.

### 2.1.3. Verificación en estados límite último

Al considerar un estado límite de rotura o deformación excesiva, se debe verificar:

$$E_d \leq R_d \quad (2.1)$$

En donde:

$E_d$  valor de cálculo de los efectos de las acciones;

$R_d$  valor de cálculo de la resistencia correspondiente.

El valor de cálculo de los efectos de las acciones que se considera en este trabajo puede expresarse como<sup>2</sup>:

$$E_d = \sum_{j \geq 1} \gamma_{G,j} G_{k,j} + \sum_{i \geq 1} \gamma_{Q,i} \psi_{0,i} Q_{k,i} \quad (2.2)$$

donde

+ significa “a combinarse con”;

$\sum$  significa “el efecto combinado de”

### 2.1.4. Verificación en estados límite de servicio

Se debe verificar que:

$$E_d \leq C_d \quad (2.3)$$

En donde:

$E_c$  Es el valor de cálculo de los efectos de las acciones especificadas en el criterio de aptitud al servicio, determinado sobre la base de la combinación correspondiente;

$C_d$  Es el valor de cálculo límite del criterio correspondiente de aptitud al servicio.

El valor de cálculo de los efectos de las acciones, que se utiliza aquí, puede expresarse como<sup>3</sup>:

$$E_d = \sum_{j \geq 1} G_{k,j} + \sum_{i \geq 1} \psi_{0,i} Q_{k,i} \quad (2.4)$$

---

<sup>2</sup>No se consideran las fuerzas de pretensado.

<sup>3</sup>*supra.*

- + significa “a combinarse con”;
- $\sum$  significa “el efecto combinado de”

Notar que, en este caso, los coeficientes  $\gamma_f$  son todos iguales a 1.

### 2.1.5. Coeficientes parciales de seguridad de los materiales

El Eurocódigo 2, en la Tabla 2.1N, recomienda los siguientes valores para el hormigón y el acero en estado límite último:

Situaciones de diseño	$\gamma_c$ para el hormigón	$\gamma_s$ para el acero
Persistentes y transitorias	1,5	1,15
Accidentales	1,2	1,0

**Tabla 2.2** – Coeficientes parciales de seguridad de los materiales en estado límite último.

En estado límite de servicio el valor recomendado, para situaciones no cubiertas por cláusulas especiales, es 1,0 (UNE-EN:1992, 2004, Nota 2.4.2.4(2)).

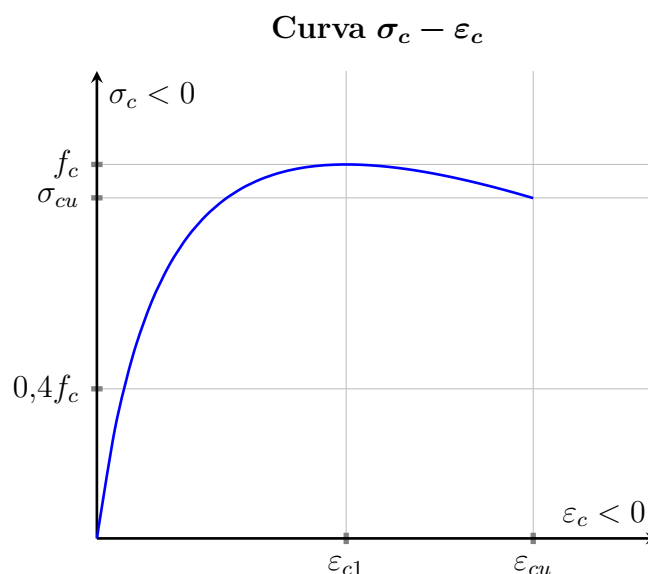
## 2.2. Tensiones normales

Adoptamos el EN 1992-1-1:2004 (Eurocódigo N<sup>o</sup> 2) “Proyecto de Estructuras de Hormigón” como normativa que regirá este trabajo en cuanto a las consideraciones para este material.

### 2.2.1. Modelo constitutivo del Hormigón

#### 2.2.1.1. Curvas de tensión-deformación a compresión

Una curva típica de tensión-deformación del hormigón a compresión es la mostrada en la figura 2.1:



**Figura 2.1** – Curva típica de tensión-deformación del hormigón a compresión.

Existen varias propuestas de curvas tensión - deformación del hormigón para el análisis no lineal, algunas de las cuales se muestran en la tabla 2.3 (Kmiecik y Kaminski, 2011):

### 2.2.1.2. Curvas propuestas por el EC2

Como se ha adoptado la norma Eurocódigo N<sup>o</sup> 2 (UNE-EN:1992, 2004), se analiza a continuación las curvas ahí establecidas. Para los gráficos de este capítulo<sup>4</sup> se utiliza, la clase C25/30, la cual se refiere a la resistencia característica en probeta cilíndrica/cúbica<sup>5</sup>. La clase C25/30 tiene un  $f_{ck} = 25MPa$  en probeta cilíndrica<sup>6</sup>.

El UNE-EN:1992 (2004) propone tres diagramas de tensión-deformación, a saber:

#### 1. Diagrama para el análisis estructural

<sup>4</sup>En lo que sigue, y en general en todo este trabajo, adoptamos el Sistema Internacional de Unidades. Las fórmulas serán presentadas tal que no requieran factores de conversión (a menos que se indique explícitamente las unidades), es decir, estarán en unidades coherentes del SI, sin múltiplos ni submúltiplos, por lo tanto, en los casos de fórmulas empíricas, también los datos de entrada deberán introducirse sin múltiplos ni submúltiplos.

<sup>5</sup>En Paraguay se utiliza probeta cilíndrica.

<sup>6</sup>EL UNE-EN:1992, 2004 define la resistencia característica a compresión del hormigón  $f_{ck}$  como: *el valor de la resistencia que el 5% de todos los posibles resultados de ensayos sobre un determinado hormigón no se llega a alcanzar.*

Nombre / Fuente	Fórmula	Variables
Parábola de Madrid	$\sigma_c = E_c \varepsilon_c \left[ 1 - \frac{1}{2} \left( \frac{\varepsilon_c}{\varepsilon_{c1}} \right) \right]$	$\sigma_c = f(E_c, \varepsilon_{c1})$
Desay & Krishnan	$\sigma_c = \frac{E_c \varepsilon_c}{1 + \left( \frac{\varepsilon_c}{\varepsilon_{c1}} \right)^2}$	$\sigma_c = f(E_c, \varepsilon_{c1})$
EN 1992-1-1	$\sigma_c = f_{cm} \frac{k\eta - \eta^2}{1 + (k-2)\eta}$ $k = 1,1 E_{cm} \frac{\varepsilon_{c1}}{f_{cm}}; \quad \eta = \frac{\varepsilon_c}{\varepsilon_{c1}}$	$\sigma_c = f(E_{cm}, f_{cm}, \varepsilon_{c1})$
Majewski	$\sigma_c = E_c \varepsilon_c \text{ si } \sigma_c \leq e_{lim} f_{cm}$ $\left. \begin{aligned} \sigma_c = f_{cm} \frac{(e_{lim} - 2)^2}{4(e_{lim} - 1)} \left( \frac{\varepsilon_c}{\varepsilon_{c1}} \right)^2 + \\ - f_{cm} \frac{(e_{lim} - 2)^2}{2(e_{lim} - 1)} + \\ f_{cm} \frac{e_{lim}^2}{4(e_{lim} - 1)} \end{aligned} \right\} E_c = \frac{f_{cm}}{\varepsilon_c} (2 - e_{lim}) \text{ si } \sigma_c > e_{lim} f_{cm}$ $e_{lim} = 1 - \exp\left(-\frac{f_c}{80}\right)$	$\sigma_c = f(E_c, f_{cm}, \varepsilon_{c1})$
Wang & Hsu	$\sigma_c = \zeta f_{cm} \left[ 2 \left( \frac{\varepsilon_c}{\zeta \varepsilon_{c1}} \right) - \left( \frac{\varepsilon_c}{\varepsilon_{c1}} \right)^2 \right] \text{ si } \frac{\varepsilon_c}{\zeta \varepsilon_{c1}} \leq 1$ $\sigma_c = \zeta f_{cm} \left[ 1 - \left( \frac{\varepsilon_c / \zeta \varepsilon_{c1} - 1}{2/\zeta - 1} \right)^2 \right] \text{ si } \frac{\varepsilon_c}{\zeta \varepsilon_{c1}} \geq 1$	$\sigma_c = f(f_{cm}, \varepsilon_{c1})$
Fórmula de Sáenz	$\sigma_c = \frac{\varepsilon_c}{A + B\varepsilon_c + C\varepsilon_c^2 + D\varepsilon_c^3}$	$\sigma_c = f(E_c, f_{cm}, f_{cu}, \varepsilon_{c1}, \varepsilon_{cu1})$

**Tabla 2.3** – Relaciones de tensión-deformación del hormigón para análisis no lineal. Fuente: Kmieciak y Kaminski (2011).

2. Diagrama parábola-rectángulo
3. Diagrama bilineal

**Diagrama  $\sigma_c - \varepsilon_c$  para el análisis estructural**

$$\sigma_c = \begin{cases} f_{cm} \frac{k\eta - \eta^2}{1 + (k-2)\eta} & \text{si } 0 \geq \varepsilon_c > \varepsilon_{cu1} \\ 0 & \text{de otra manera} \end{cases}$$

$$k = 1,05 E_{cm} \frac{|\varepsilon_{c1}|}{f_{cm}}; \quad \eta = \frac{\varepsilon_c}{\varepsilon_{c1}}; \quad f_{cm} = f_{ck} + 8 \text{ (MPa)} \quad (2.5)$$

$$E_{cm} = \underbrace{1,2}_{\text{basalto}} \times 22 \left[ \frac{f_{cm}}{10} \right]^{0,3} \text{ (GPa)} \quad (f_{cm} \text{ en MPa})$$

$$\varepsilon_{c1} = 0,7 f_{cm}^{0,31} \leq 2,8 \text{ (\%)}_0$$

Donde:

- $\sigma_c$ : tensión de compresión del hormigón.
- $\varepsilon_c \in [\varepsilon_{cu1}, 0]$ : deformación unitaria del concreto a compresión, ( $\varepsilon_c \leq 0$ ).
- $\eta = \frac{\varepsilon_c}{\varepsilon_{c1}}$  (siendo  $\varepsilon_c$  y  $\varepsilon_{c1}$  ambas menores que 0)
- $\varepsilon_{c1}$ : deformación unitaria correspondiente a la tensión máxima  $f_{cm}$ , en las fórmulas entra con valor negativo.
- $\varepsilon_{cu1} = -3,5 \text{ ‰}$ : deformación unitaria última a compresión del hormigón.
- $f_{cm}$ : resistencia media a la compresión a los 28 días.
- $E_{cm}$ : módulo de elasticidad secante, el cual está definido por  $\sigma_c = 0$  y  $\sigma_c = 0,4 f_{cm}$ . El factor 1,2 que aparece en su fórmula es debido a que en Paraguay el agregado grueso más utilizado es el basáltico<sup>7</sup>.

Derivando la ecuación (2.5) tenemos el módulo de elasticidad tangente del hormigón en la zona de compresión:

$$\frac{d\sigma_c}{d\varepsilon_c} = E_c = \frac{f_{cm}}{\varepsilon_{c1}} \left( \frac{k - (k^2 - 4k + 2)\eta + 3(k-2)\eta^2}{[1 + (k-2)\eta]^2} \right) \quad (2.6)$$

que para  $\varepsilon_c = 0 \Rightarrow \eta = 0$  se convierte en:

$$E_{c0} = 1,05 E_{cm} \quad (2.7)$$

<sup>7</sup>UNE-EN:1992, 2004 3.1.3 (2)

**Diagrama parábola-rectángulo** Para<sup>8</sup>  $f_{cd} = \alpha_{cc} \frac{f_{ck}}{\gamma_c}$

$$\sigma_c = \begin{cases} f_{cd} \left[ 1 - \left( 1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^n \right] & \text{si } 0 \leq |\varepsilon_c| \leq |\varepsilon_{c2}| \text{ (parábola)} \\ f_{cd} & \text{si } |\varepsilon_{c2}| \leq |\varepsilon_c| \leq |\varepsilon_{cu2}| \text{ (rectángulo)} \end{cases} \quad (2.8)$$

Donde:

- $\varepsilon_{cu2} = -3,5 \text{ ‰}$ : deformación unitaria última a compresión del hormigón<sup>9</sup>
- $\varepsilon_{c2} = -2 \text{ ‰}$ : deformación unitaria en el vértice de la parábola<sup>10</sup>.
- $\alpha_{cc}$ : coeficiente de 'cansancio' a compresión<sup>11</sup>, generalmente  $\alpha_{cc} = 1,00$ , sin embargo este valor (y el correspondiente  $\alpha_{ct}$  para tracción) debería ser reducido por un factor  $k_t = 0,85$  para edades del hormigón  $t > 28$  días<sup>12</sup>.
- $\gamma_c = 1,5$ : coeficiente parcial de seguridad del hormigón<sup>13</sup>.
- $n = 2$ : grado de la parábola<sup>14</sup>.

**Diagrama bilineal**

$$\sigma_c = \begin{cases} \frac{f_{cd}}{\varepsilon_{c3}} \varepsilon_c & \text{si } |\varepsilon_c| \leq |\varepsilon_{c3}| \\ f_{cd} & \text{si } |\varepsilon_c| > |\varepsilon_{c3}| \end{cases} \quad (2.9)$$

Puestos en un gráfico, la representación de estas ecuaciones es:

<sup>8</sup>UNE-EN:1992, 2004 eq. (3.15)

<sup>9</sup>Para todas las clases de Hormigón con  $f_{ck} \leq 50 \text{ MPa}$ .

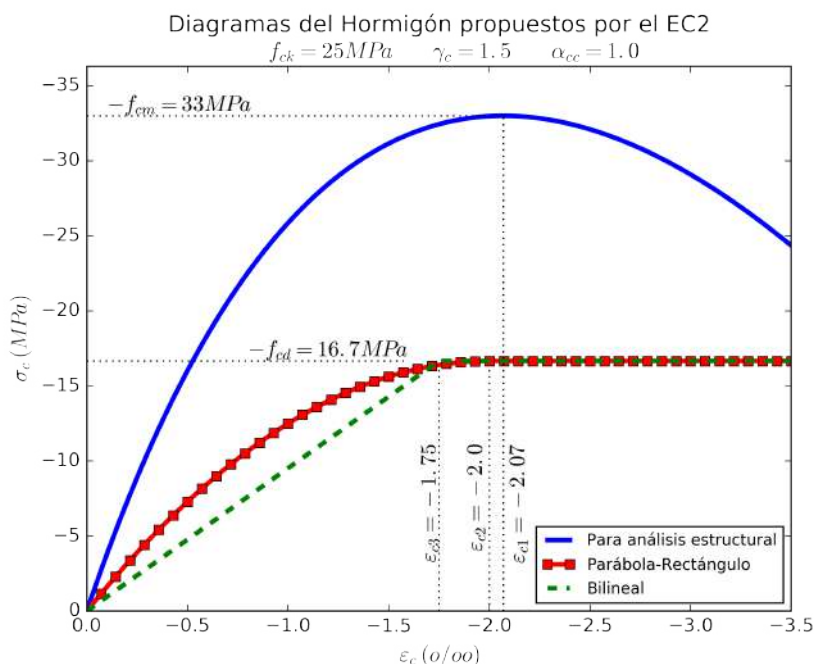
<sup>10</sup>Ibídem.

<sup>11</sup>UNE-EN:1992, 2004 3.1.6 (1)

<sup>12</sup>UNE-EN:1992, 2004 3.1.2 (4)

<sup>13</sup>UNE-EN:1992, 2004 Tabla 2.1N

<sup>14</sup>UNE-EN:1992, 2004 Tabla 3.1



**Figura 2.2** – Diagramas tensión-deformación del hormigón: 1. Para el análisis estructural (adoptado), 2. Parábola-rectángulo y 3. Bilineal, propuestos en el Eurocódigo N° 2 (nótese que el diagrama para el análisis estructural no está afectado por el coeficiente de cansancio  $\alpha$ , por lo que es considerado para cargas de 'corta duración'). Fuente: elaboración propia.

### 2.2.1.3. Curvas de tracción

Bastante más difícil, experimentalmente hablando, es la obtención del diagrama tensión-deformación del hormigón a tracción, por las dificultades que envuelve su ejecución.

El EC2 define a la resistencia a tracción del hormigón como la *máxima tensión que el hormigón puede soportar cuando está sometido a tracción uniaxial*, y propone la siguiente ecuación:

$$f_{ctm} = 0,30 f_{ck}^{2/3} \quad [\text{MPa}] \quad (2.10)$$

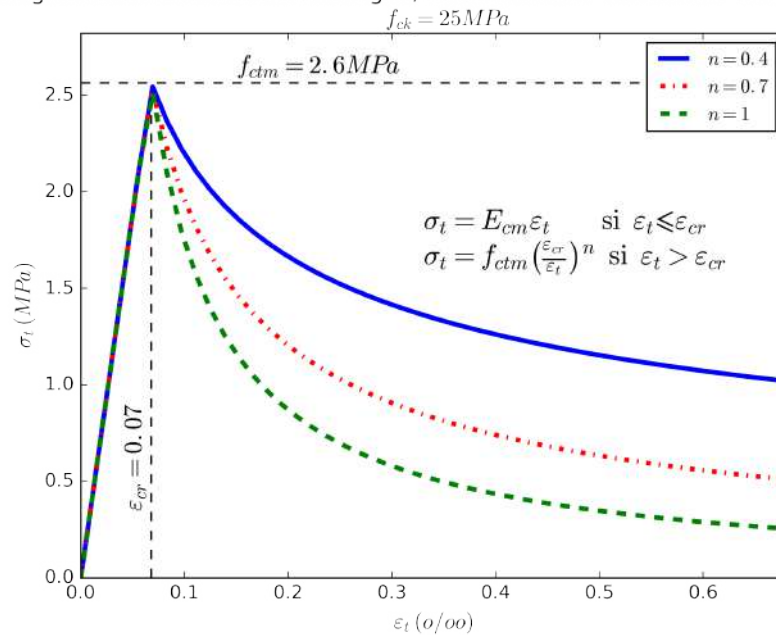
La ecuación  $\sigma_t - \epsilon_t$  del hormigón, es decir en tracción uniaxial, adoptada, es la propuesta por Wang y Hsu, 2001:

$$\sigma_t = \begin{cases} E_{cm} \epsilon_t & \text{si } 0 \leq \epsilon_t \leq \epsilon_{cr} \\ f_{ctm} \left( \frac{\epsilon_{cr}}{\epsilon_t} \right)^{0,4} & \text{si } \epsilon_t > \epsilon_{cr} \end{cases} \quad (2.11)$$

Esta ecuación toma en cuenta el fenómeno de *tension-stiffening*, es decir, la colaboración del hormigón a tracción entre fisuras. La deformación crítica a tracción  $\varepsilon_{cr}$  se tiene cuando  $\sigma_t = f_{ctm}$ , consecuentemente  $f_{ctm} = \sigma_{cr} = E_{cm}\varepsilon_{cr}$ , el cual combinado con (2.5) y (2.7) nos permite averiguar que:

$$\varepsilon_{cr} = \frac{0,30f_{ck}^{2/3}}{1,2 \times 22(f_{ck} + 8)^{0,3}} \approx 0,068 \text{‰} \quad (f_{ck} = 25 \text{MPa}) \quad (2.12)$$

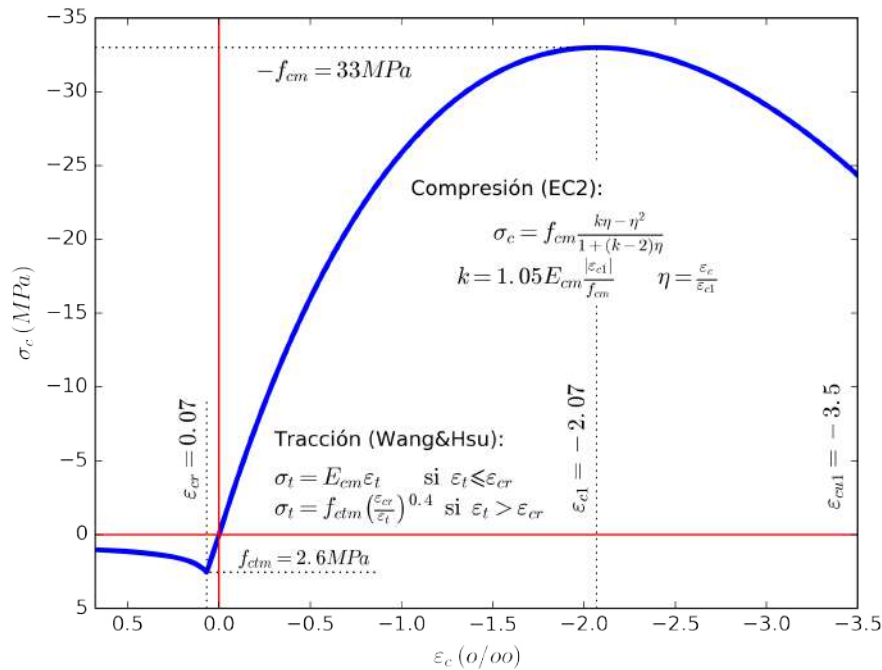
Diagramas de tracción del Hormigón, modificando la fórmula de Wang & Hsu



**Figura 2.3** – Diagramas de tracción de Wang y Hsu (2001). Se adopta el diagrama original con  $n = 0,4$ . Fuente: elaboración propia.

### 2.2.2. Diagrama adoptado

Conforme a lo visto precedentemente, se asume el siguiente diagrama para el hormigón:



**Figura 2.4** – Diagrama de hormigón adoptado para este trabajo. Rama de compresión según el UNE-EN:1992 (2004) y la rama de tracción según Wang y Hsu (2001). *Fuente:* elaboración propia.

### 2.2.3. Acero

Para el acero, se adopta directamente el diagrama propuesto en el UNE-EN:1992 (2004). A continuación se describen las variables utilizadas y sus correspondientes valores para la realización de la figura 2.5:

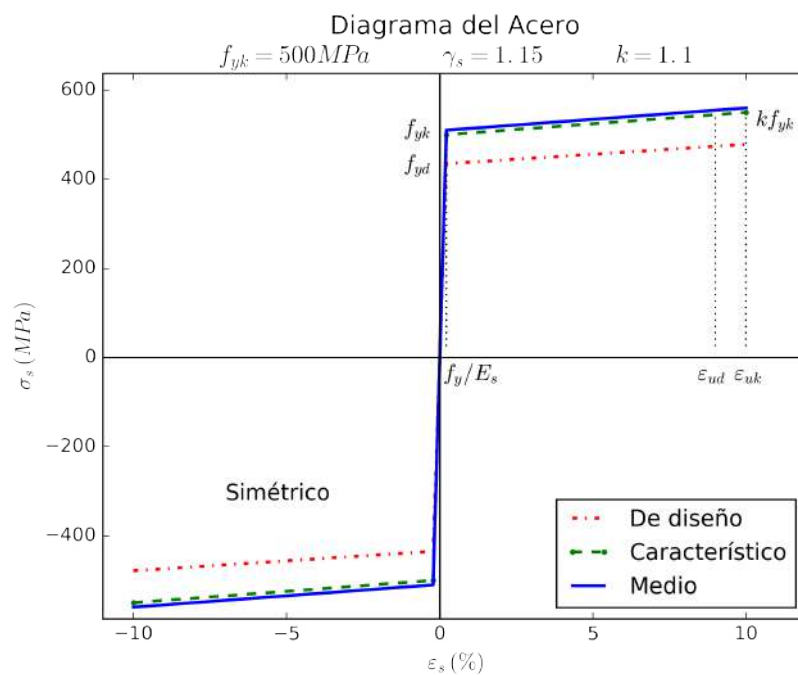
- $f_{yk} = 500 \text{ MPa}$  Resistencia característica del acero.
- $f_{ym}$ : Resistencia media (2.13).
- $\gamma_s = 1,15$ : Coeficiente de seguridad.
- $f_{yd} = \frac{f_{yk}}{\gamma_s}$ : Resistencia de cálculo.
- $E_s = 200 \text{ GPa}$ : Módulo de elasticidad de diseño.
- $\varepsilon_{uk} = 10 \%$ : Deformación unitaria característica en fuerza máxima<sup>15</sup>.

<sup>15</sup>Es igual a la deformación unitaria media a fuerza máxima. UNE-EN:1992, 2004 Annex C C.1 (3).

- $\varepsilon_{ud} = 0,9 \varepsilon_{uk}$ : Deformación unitaria última de diseño
- $k = (f_t/f_y)_k = 1,10$ : Valor característico<sup>16</sup> de la relación entre la tensión máxima de rotura y el límite elástico.
- $\rho_s = 7850 \text{ kg/m}^3$ : Densidad.

La resistencia media del acero es UNE-EN:1992, 2004:

$$f_{ym} = f_{yk} + 10 \text{ (MPa)} \quad (2.13)$$



**Figura 2.5** – Diagrama Tensión-Deformación del acero. *Fuente:* elaboración propia.

#### 2.2.4. Cálculo en rotura

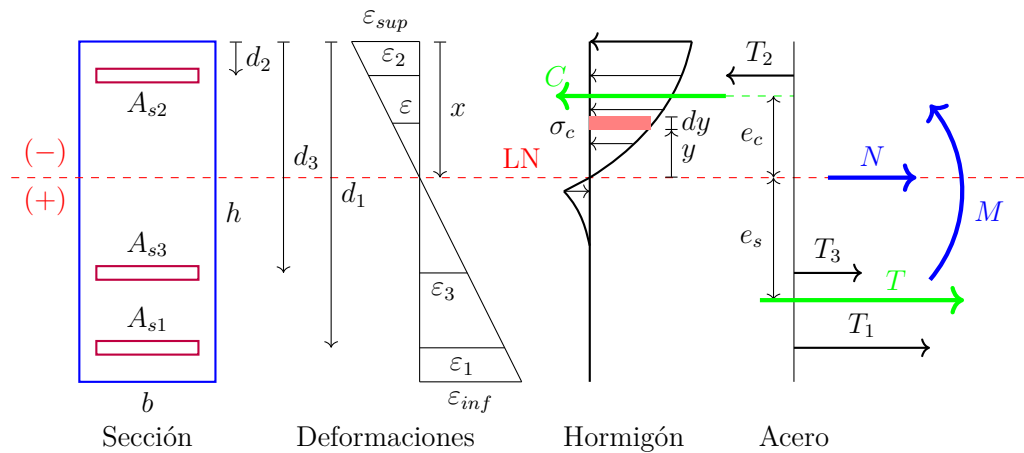
Para determinar la capacidad última de una sección dada de hormigón armado, se utiliza el método de los estados límites considerados por los Eurocódigos.

<sup>16</sup>Es igual a su valor medio UNE-EN:1992, 2004

### 2.2.4.1. Hipótesis básicas

Las siguientes son hipótesis asumidas para el análisis:

- Secciones planas y perpendiculares al eje de la viga antes de la deformación permanecen planas y perpendiculares al eje, después de la deformación (Viga de Euler - Bernoulli);
- No hay deslizamiento relativo entre hormigón y acero de armado (adherencia perfecta);
- El hormigón no está confinado;
- Se considera que los desplazamientos son muy pequeños en relación a las dimensiones de la viga, por lo que no será necesario considerar la no-linealidad geométrica;
- Se limita el análisis a sección rectangular;
- No se consideran efectos de fluencia y retracción.



**Figura 2.6** – Comportamiento esquemático de una sección de hormigón armado.  
Fuente: elaboración propia.

En base a la figura 2.6 escribimos las ecuaciones siguientes<sup>17</sup>:

$$\begin{aligned}
 \text{Compatibilidad: } \quad \kappa &= \frac{-\varepsilon_{sup}}{x} = \frac{\varepsilon_{inf}}{h-x} & \kappa &= \frac{-\varepsilon}{y} = \frac{\varepsilon_i}{d_i-x} \\
 \text{Acero:} \quad T &= \sum T_i = \sum \sigma_{s_i} A_{s_i} & e_s &= \frac{\sum T_i d_i}{T} - x \\
 \text{Hormigón:} \quad C &= \int_{x-h}^x b \sigma_c dy & e_c &= \frac{\int_{x-h}^x b \sigma_c y dy}{C} \\
 \text{Equilibrio:} \quad T + C &= N & M &= T e_s + C e_c
 \end{aligned} \tag{2.14}$$

En flexión simple  $N = 0$ , entonces  $T + C = 0$ . Como esta condición será revisada en una rutina computacional, es mejor escribirla de la siguiente manera:

$$|T + C| \leq tol \tag{2.15}$$

Donde  $tol$  es la tolerancia adoptada. Se ha verificado que la definida a continuación es suficiente para los propósitos establecidos:

$$tol := \frac{bh f_{cm}}{4000} \tag{2.16}$$

Se aclara la notación utilizada:

$\kappa$ :	curvatura;
$b, h$ :	Dimensiones de la sección transversal;
$A_{s_i}$ :	Armaduras longitudinales <sup>18</sup> ;
$d_i$ :	Posiciones de las armaduras respecto a la fibra superior de la sección;
$\varepsilon_{inf}, \varepsilon_{sup}$ :	Deformaciones unitarias de las fibras inferior y superior respectivamente;
$\varepsilon_i$ :	Deformaciones unitarias de las armaduras longitudinales;
$\varepsilon$ :	Deformación unitaria genérica;
LN:	Línea neutra;
$x$ :	Profundidad de la línea neutra;
$\sigma_c$ :	Tensión en el hormigón;

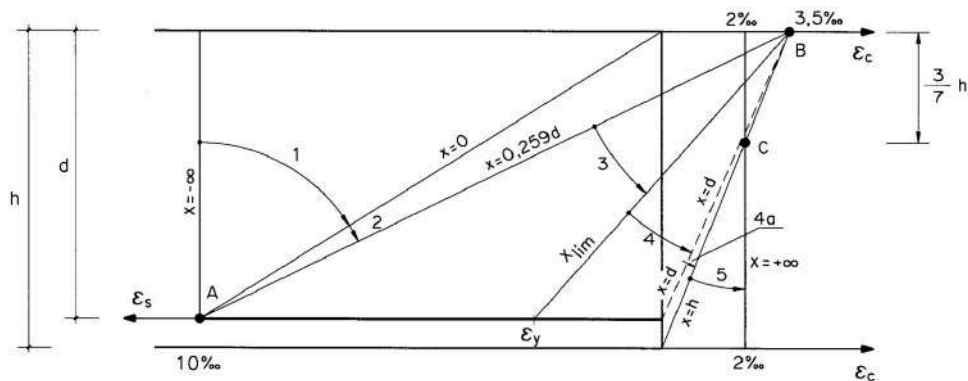
---

<sup>17</sup>Nótese que estas son válidas para cualquier sección transversal simétrica respecto a un eje vertical que pasa por su centro de área.

- $\sigma_s$ : Tensión en el acero;  
 $y$ : Posición de una fibra cualquiera respecto a la línea neutra;  
 $C$ : Resultante de las fuerzas en el hormigón;  
 $e_c$ : Posición de la resultante de las fuerzas del hormigón respecto a la línea neutra;  
 $T_i$ : Fuerzas en las armaduras longitudinales;  
 $T$ : Resultante de las fuerzas en el acero;  
 $e_s$ : Posición de la resultante de las fuerzas del acero respecto a la LN;  
 $M$ : Momento flector externo.

### 2.2.4.2. Dominios de deformación. Diagrama de pivotes

Tal como se cita en Jiménez Montoya y col. (2009): *Las deformaciones límite de las secciones, según la naturaleza de la solicitación, conducen a admitir varios dominios de deformación*, estos dominios se muestran en la figura 2.7:



**Figura 2.7** – Diagrama de pivotes. Fuente: Jiménez Montoya y col. (2009).

La capacidad última de la sección es la suma de los aportes del hormigón y del acero, por lo que, en base a las figuras 2.6, el equilibrio de fuerzas y momentos<sup>19</sup> requiere:

$$\begin{aligned}
 N_u &= \sum (\sigma_{s_i} A_{s_i}) + \int_{x-h}^x b \sigma_c dy \\
 M_u &= \sum \left[ \sigma_{s_i} A_{s_i} \left( d_i - \frac{h}{2} \right) \right] + \int_{x-h}^x b \sigma_c \left( \frac{h}{2} - x + y \right) dy
 \end{aligned}
 \tag{2.17}$$

<sup>19</sup>Respecto al centro de gravedad de la sección.

donde  $N_u$  y  $M_u$  son la fuerza y el momento últimos resistentes de la sección respectivamente. Las ecuaciones de compatibilidad son las indicadas en (2.14), con las consideraciones adecuadas a cada plano de rotura definido en cada dominio. (Calavera, 1985):

- **Dominio 1. Tracción simple o compuesta**

Profundidad de LN:  $-\infty < x \leq 0$

El plano de deformación giran alrededor del punto  $A$ , la armadura inferior, en estas condiciones, experimenta una deformación constante  $\varepsilon_{su}$ <sup>20</sup>. Toda la sección está traccionada. En el límite, cuando  $x \rightarrow -\infty$ , tenemos tracción simple.

- **Dominio 2. Flexión simple o compuesta**

Profundidad de LN:  $0 < \frac{x}{d_1} \leq \frac{\varepsilon_{cu1}}{\varepsilon_{cu1} + \varepsilon_{su}}$

El pivote sigue siendo  $A$ , pero la sección comienza a soportar tensiones de compresión, hasta la rotura del hormigón, en el que llega a la deformación  $\varepsilon_{cu1}$ . En el límite de este dominio y el 3, se tienen al hormigón y al acero, ambos a su capacidad máxima.

- **Dominio 3. Flexión simple o compuesta**

Profundidad de LN:  $\frac{\varepsilon_{cu1}}{\varepsilon_{cu1} + \varepsilon_{su}} < \frac{x}{d_1} \leq \frac{\varepsilon_{cu1}}{\varepsilon_{cu1} + \varepsilon_y}$

La profundidad máxima de la LN en este dominio es:

$$x_{lim} := \frac{\varepsilon_{cu1}}{\varepsilon_{cu1} + \varepsilon_y} d_1 \quad (2.18)$$

El plano de deformación, esta vez gira alrededor de  $B$ , con lo que la fibra superior del hormigón se encuentra a una deformación unitaria constante  $\varepsilon_{cu1}$  y la del acero decrece desde  $\varepsilon_{su}$  hasta  $\varepsilon_y = f_y/E_s$  que es la correspondiente al límite elástico.

- **Dominio 4. Flexión simple o compuesta**

Profundidad de LN:  $\frac{\varepsilon_{cu1}}{\varepsilon_{cu1} + \varepsilon_y} < \frac{x}{d_1} \leq 1$

Los planos de deformación siguen girando alrededor de  $B$  hasta que la profundidad de la LN alcance el valor  $x = d_1$  en la que la armadura inferior no trabaja.

---

<sup>20</sup> $\varepsilon_{su}$  es la deformación unitaria máxima permitida al acero en el interior del hormigón, generalmente  $\varepsilon_{su} = 1\%$  Jiménez Montoya y col. (2009).

- **Dominio 4a. Flexión compuesta**

Profundidad de LN:  $d_1 < x \leq h$

Los planos de deformación siguen girando alrededor de  $B$  hasta que la profundidad de la LN alcance el valor  $x = h$ , en este instante toda la sección se encuentra a compresión.

- **Dominio 5. Compresión simple o compuesta**

Profundidad de LN:  $h < x < +\infty$

Los planos de deformación giran alrededor de  $C$  y se tiene compresión compuesta. En el límite  $x \rightarrow +\infty$  se tiene compresión simple con  $\varepsilon_c = \varepsilon_{c1}$ ,

La profundidad de  $C$  es:

$$x_C = \frac{\varepsilon_{cu} - \varepsilon_{c1}}{\varepsilon_{cu}} h \quad (2.19)$$

$d_1$ : canto útil, distancia desde la fibra superior hasta el centro de gravedad de las armaduras de tracción;

$\varepsilon_{cu1}$ : deformación unitaria última del hormigón (usualmente 0,35 %);

En donde:  $\varepsilon_{su}$ : deformación unitaria última del acero (usualmente 1 %)

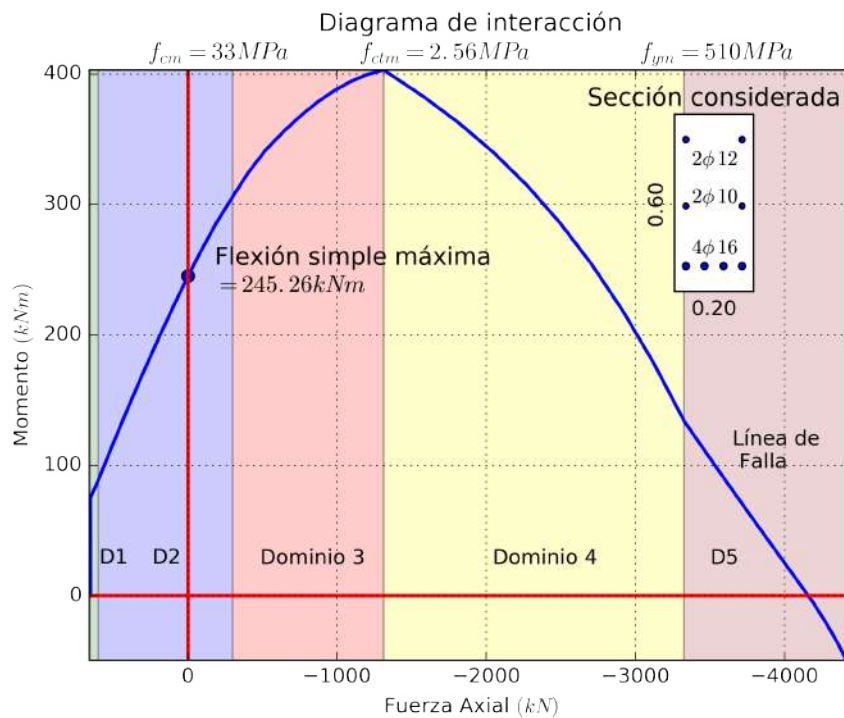
$\varepsilon_y$ : deformación unitaria en el límite elástico del acero;

$x_{lim}$ : profundidad máxima de la línea neutra en dominio 3;

$f_y$ : tensión del acero en su límite elástico;

### 2.2.5. Diagrama de interacción

Tomando en cuenta la sección anterior 2.2.4, y en cumplimiento con uno de los objetivos de este trabajo, se ha desarrollado un programa en **python** que realiza el diagrama de interacción de cualquier sección rectangular de hormigón armado. El manual de uso del mismo se encuentra en el Anexo A.2. Se muestra el diagrama siguiente como ejemplo elaborado con el citado algoritmo:



**Figura 2.8** – Ejemplo de diagrama de interacción de una sección de hormigón armado realizado con el método `diag_interaccion()` del módulo `Viga.py` (Anexo A.2). La curva indicada representa la línea de rotura para la sección de hormigón considerada. En este ejemplo, está confeccionada con los valores medios de resistencia y no con los característicos. *Fuente:* elaboración propia.

## 2.3. Estado límite de deformación

En UNE-EN:1990, 2002 se señala que la verificación de los estados límite de servicio (o de aptitud al servicio) debería basarse en criterios referentes a los siguientes aspectos:

- Deformaciones;
- Vibraciones;
- Daños que puedan afectar adversamente a la apariencia, durabilidad o funcionamiento de la estructura.

Dado el alcance de este trabajo, sección 1.1, adoptamos el estado límite de deformación como criterio a considerar entre los estados límite de servicio.

### 2.3.1. Relación momento-curvatura

Para la verificación de los estados límite de servicio, en particular el referente a los desplazamientos en las vigas de hormigón armado, existe, en primera aproximación, la teoría elemental de la mecánica de materiales, y desarrollos matemáticos más elaborados para considerar los múltiples fenómenos que presenta el hormigón armado, el cual se desvela cada vez más complejo a medida que avanzan los estudios en este campo. Sin embargo, en este trabajo, se pretende llegar con aproximación razonable a la situación real, considerando la no linealidad material del hormigón armado, reuniendo la información de su naturaleza compuesta en el diagrama momento - curvatura.

Existe una variedad de métodos utilizados para calcular la flecha de una viga de hormigón armado, la que utilizaremos aquí es el método del **diagrama de momento - curvatura**, por la razones siguientes:

El más riguroso método para calcular deflexiones es computando las curvaturas en secciones frecuentes a lo largo del elemento y luego calcular la deflexión por integración numérica (UNE-EN:1992, 2004, 7.4.3(7)).

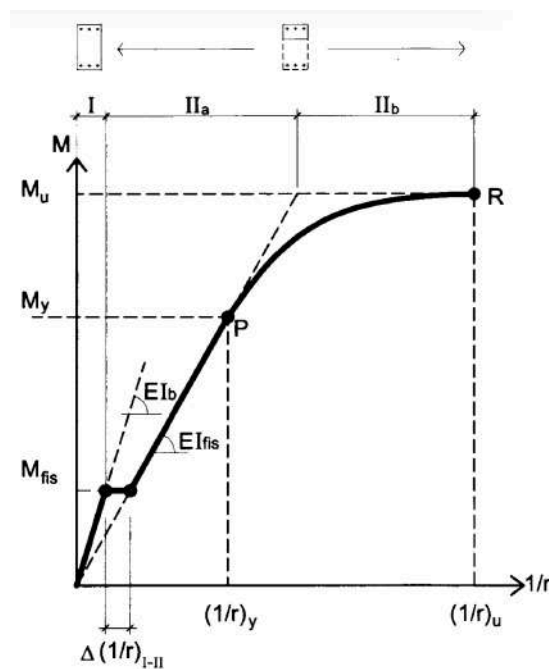
La mejor forma de obtener una flecha creíble para cualquier estado de cargas, es el método general del Eurocódigo que requiere calcular las flechas mediante integración de las curvaturas. (Arroyo, 2016).

El diagrama de momento - curvatura, es un recurso que ha mostrado ser razonablemente preciso para estimar deformaciones, el problema central proviene de la fisuración del hormigón, la cual ocasiona una disminución de la sección útil del elemento estructural. Existen algunas aproximaciones que se enfrentan al problema, por ejemplo la presentada por Branson y Metz (1963).

En la figura 2.9 se muestra un diagrama típico con sus estadios de deformación, a saber:

- **Estadio I:** Estadio elástico sin fisurar. Tanto el hormigón como el acero permanecen en su tramo elástico, el hormigón resiste aún las tracciones;

- **Estadio IIa:** Estadio fisurado. El hormigón pierde capacidad a tracción y se fisura, todas las tracciones pasan a ser absorbidas por el acero;
- **Estadio IIb:** Estadio de no linealidad. El hormigón o el acero o ambos están en etapa no lineal, esto es, superados uno de los límites de elasticidad o ambos;
- **Estadio III:** Estadio de prerrotura. El elemento falla cuando el hormigón o el acero alcanza su capacidad última, esto es, 1 % de deformación unitaria en el acero o 0,35 % en el hormigón.



**Figura 2.9** – Diagrama de momento-curvatura típico. *Fuente:* Jiménez Montoya y col. (2009).

La idea central, de relacionar el momento con la curvatura, proviene de la conocida ecuación de la flexión:

$$\frac{d^2v}{dx^2} = \kappa = \frac{M}{EI} \quad (2.20)$$

En donde:

- $v$  Deflexión vertical;
- $x$  Coordenada de un punto del eje de la viga;

- $\kappa$  Curvatura en cualquier punto  $x$  de la viga;  
 $M$  Momento flector;  
 $EI$  Módulo de elasticidad  $\times$  inercia, producto considerado variable a lo largo de la viga.

El objetivo es relacionar el momento flector con la curvatura sin pasar antes por la determinación del factor  $EI$ , de la forma.

$$\kappa = f(M) \tag{2.21}$$

Esto es posible lograr. En este trabajo consideramos las ecuaciones (2.14) para la obtención de la citada relación (2.21). El algoritmo es explicado y verificado en la sección 4.3. Finalmente la deformada de la viga y, en particular, la flecha se obtiene, por ejemplo, por doble integración de la ecuación (2.20).

# Capítulo 3

## APRENDIZAJE AUTOMÁTICO

### 3.1. Conceptos iniciales

#### 3.1.1. Definición

*Machine Learning* o, en español, *Aprendizaje Automático* (traducción que se ha adoptado para este trabajo) es la ciencia de programar computadoras tal que ellas puedan ‘aprender’ a partir de datos (Géron, 2017). Es el área de estudios que proporciona a las computadoras la habilidad de aprender a realizar tareas sin haberse explícitamente programado (Samuel, 1959).

Una definición más rigurosa es la siguiente (Mitchell y col., 1997):

Se dice que un programa de computadora aprende de la experiencia  $E$  respecto a alguna tarea  $T$  y alguna medida de desempeño  $P$ , si su desempeño en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ .

#### 3.1.2. Paradigma y justificación

El **paradigma** básico del aprendizaje automático es el siguiente: (Grimson, 2016):

1. Observar un conjunto de datos: Conjunto de entrenamiento (*training set*)

2. Estudiar procesos con los que se pueda inferir la obtención de esos datos;
3. Utilizar esas inferencias para realizar predicciones respecto a datos previamente no considerados. Conjunto de prueba (*test set*.)

La **justificación** para utilizar el aprendizaje automático está en (Abu-Mostafa, 2012):

- La existencia de un patrón en el comportamiento de las variables;
- La dificultad de expresarlo matemáticamente;
- La existencia de datos que puedan ser analizados.

### 3.1.3. Tipos de sistemas de aprendizaje automático

Se puede categorizar los sistemas de aprendizaje automático de la siguiente manera (Géron, 2017):

- Si son o no entrenados bajo supervisión humana.
  - **Supervisado:** Cuando en la base de datos de entrenamiento, se cuenta con las soluciones deseadas. Esto es, los datos están “etiquetados”.
  - **No supervisado:** Cuando no se cuentan con las soluciones, los datos de entrenamiento no están etiquetados y el sistema trata de aprender sin un “profesor”.
  - **Semisupervisado:** Cuando parte de los datos están etiquetados y parte no.
  - **Aprendizaje reforzado:** En este caso, el sistema es llamado *agente* y debe aprender por sí mismo adoptando una estrategia, conocida en este contexto como *política*. Una política define qué acciones debería de tomar ante las diferentes situaciones.
- Si pueden o no aprender incrementalmente en el proceso.
  - **En línea:** Cuando el sistema entrena incrementalmente, alimentándose secuencialmente. Es ideal para sistemas que reciben un flujo continuo de información y necesita adaptarse rápidamente y de manera autónoma.

- **Aprendizaje por lotes:** El sistema también entrena incrementalmente pero con todos los datos disponibles. Una vez entrenado trabaja sin nuevo aprendizaje.
- Si trabajan por simple comparación de nuevos datos con datos conocidos, o en su defecto detectan patrones en la base de datos de entrenamiento y construyen modelos predictivos, como haría un científico.
  - **Aprendizaje basado en instancias:** el sistema aprende con ejemplos, luego los generaliza utilizando casos similares.
  - **Aprendizaje basado en modelos:** Se trata de construir modelos basados en un conjunto de ejemplos y luego utilizar esos modelos para realizar predicciones.

*Para los casos tratados aquí, el aprendizaje automático es del tipo supervisado.*

### 3.1.4. Ventajas y desventajas de los métodos adoptados

A continuación se comenta brevemente sobre los métodos de aprendizaje automático a ser utilizados:

#### 3.1.4.1. Regresión lineal

Es el método más simple y directo de aproximación y predicción de resultados. Como se verá el problema que se enfrenta en este trabajo no es lineal, por lo que su utilización se limitará a pocos casos, sin embargo es importante considerarlo ya que da una visión muy útil de las relaciones entre las diferentes variables que intervienen en el objeto de estudio. Se desarrolla este método en la sección 3.3.1

#### 3.1.4.2. Vector soporte automático

Esta técnica se desarrolla en la sección 3.3.2. Es un potente método de aprendizaje automático que ha mostrado excelentes resultados en otros casos, algunos

de los cuales se mencionan en la sección 1.4.2. Sus ventajas y desventajas son las siguientes (Cevik y col., 2014):

■ **Ventajas**

- Los casos irregulares y el “ruido” pueden ser minimizados con la correcta elección de parámetros;
- Independientemente de la distribución de datos (no monótono o no lineal) esta técnica exhibe un buen desempeño para problemas de clasificación y puede dar estimaciones muy precisas;
- Las condiciones ruidosas pueden tratarse mediante identificación automática e incorporación de vectores soporte, ya que el modelo minimiza el efecto de la falta de vector soporte en el entrenamiento;
- Gracias a algunos vectores clave en el modelo de entrenamiento, las predicciones futuras se pueden mejorar rastreando en el modelado histórico.

■ **Desventajas**

- El principal problema está en el proceso de selección de la función kernel adecuada (ver ecuación 3.61) y los parámetros en función del proceso de prueba y error, que consume tiempo y gasto computacional;
- La interpretación del modelo de regresión no lineal puede ser bastante difícil debido a que el mapeo de entradas no lineales en el espacio de alta dimensión implica una alta complejidad. De este modo, el proceso de entrenamiento lleva más tiempo en comparación con los modelos lineales;
- Como el modelo depende de datos pasados, la precisión de extrapolación puede ser insatisfactoria en caso que los datos pasados sean inconsistentes.

### 3.1.4.3. Redes neuronales artificiales

La metodología inspirada en las neuronas del ser humano es esta. Es desarrollada en la sección 3.3.3

■ **Ventajas**

- Aprendizaje adaptativo. Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial;
- Tolerancia a fallos. La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño;
- Operación en tiempo real. Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

#### ■ Desventajas

- Es preciso conocer bien el problema que se desea modelar;
- El efecto caja negra “black box”. Los datos entran en la caja negra y se obtienen las predicciones, pero no se revela normalmente la naturaleza de las relaciones entre las variables independientes y dependientes;
- Los procesos pueden ser muy largos;
- Requieren la definición de muchos parámetros antes de poder aplicar la metodología.

#### 3.1.4.4. Aprendizaje de máquina extremo

Un método muy interesante que aprovecha el teorema de aproximación universal, mostrado en la página 64, con lo cual se logra aproximar cualquier función con una red neuronal de una sola capa oculta. La matemática en este caso se torna más simple con resultados excelentes.

Este método se desarrolla en 3.3.4. Sus ventajas y desventajas son:

#### ■ Ventajas (Guang-Bin Huang, 2006)

- La matemática es simple, el algoritmo solo tiene 3 pasos;
- La velocidad de aprendizaje es rápida;
- Es diferente a los algoritmos de descenso del gradiente que solo trabajan con funciones de activación diferenciables;
- Suele llegar a la solución con bastante facilidad a diferencia de los demás métodos que pueden presentar problemas de diversas índoles;

- El algoritmo es mucho más simple que los demás métodos.
- **Desventajas**
  - Para problemas complejos puede requerir demasiadas neuronas en la capa oculta;
  - No queda claro como encontrar la solución óptima;
  - No está exento de problemas de mínimo local y sobre entrenamiento.

## 3.2. Análisis de componentes principales

Las formulaciones se basan en varias variables, cada una con relativa importancia, cuanto más de ellas están en juego, más potencia computacional será requerida, por lo que sería interesante, si es posible, reducir esa cantidad en términos de sus “componentes principales”, o determinar cuáles de ellas aportan más a las estimaciones y cuáles podrían ser descartadas. A este efecto se presenta el análisis de componentes principales o PCA por sus siglas en inglés.

### 3.2.1. Definiciones importantes

Algunas definiciones importantes, tomadas de Bertsekas y Tsitsiklis (2002):

- Esperanza matemática o valor esperado:

$$\mathbf{E}[X] := \sum_x xp_X(x) \quad (3.1)$$

Donde:

- $X$  variable aleatoria;
- $x$  valor que asume la variable aleatoria  $X$ ;
- $p_X(x)$  probabilidad que la variable aleatoria  $X$  asuma el valor  $x$ .

Si  $p_X(x)$  es la misma para todos los valores  $x$ , entonces la esperanza matemática es igual a la media:

$$\mathbf{E}[X] = \frac{1}{N} \sum_{d=1}^N x_d = \mu \quad (3.2)$$

Para muestras, la media se denota con  $\bar{x}$

- Varianza:

$$\text{var}(X) := \mathbf{E} [(X - \mathbf{E}[X])^2] \quad (3.3)$$

En los casos en que se cumple (3.2), se tiene:

$$\text{var}(X) = \frac{1}{N} \sum_{d=1}^N (x_d - \bar{x})^2 = \frac{1}{N} \|\mathbf{x} - \mu \mathbf{1}_v\|^2 = \frac{1}{N} (\mathbf{x} - \mu \mathbf{1}_v)^T (\mathbf{x} - \mu \mathbf{1}_v) \quad (3.4)$$

En donde  $\mathbf{1}_v$  es un vector en  $\mathbb{R}^N$  de unos:

$$\mathbf{1}_v := \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N \times 1} \quad (3.5)$$

Para muestras, la varianza es:

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (3.6)$$

- Desviación estándar:

$$\sigma_X := \sqrt{\text{var}(X)} \quad (3.7)$$

La desviación estándar muestral es:

$$s := \sqrt{s^2} \quad (3.8)$$

- Covarianza:

$$\text{cov}(X, Y) := \mathbf{E} [(X - \mathbf{E}[X])(Y - \mathbf{E}[Y])] = \mathbf{E}[XY] - \mathbf{E}[X]\mathbf{E}[Y] \quad (3.9)$$

Notar que si  $\mathbf{E}[X] = \mathbf{E}[Y] = 0 \quad \therefore$

$$\text{cov}(X, Y) = \mathbf{E}[XY] = \frac{1}{N} \mathbf{x}^T \mathbf{y} \quad (3.10)$$

Si se tienen  $X_i$  ( $i = 1, \dots, n$ ) variables aleatorias, la siguiente es la matriz

de covarianzas:

$$\mathbf{S} = \begin{bmatrix} \text{cov}(X_1, X_1) & \cdots & \text{cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \cdots & \text{cov}(X_n, X_n) \end{bmatrix} \quad (3.11)$$

Se observa que si  $\mathbf{E}[X_i] = 0$ , de (3.9):

$$\mathbf{S} = \begin{bmatrix} \mathbf{E}[X_1 X_1] & \cdots & \mathbf{E}[X_1 X_n] \\ \vdots & \ddots & \vdots \\ \mathbf{E}[X_n X_1] & \cdots & \mathbf{E}[X_n X_n] \end{bmatrix} = \frac{1}{N} \mathbf{X} \mathbf{X}^T \quad (3.12)$$

siendo  $\mathbf{X} = [X_1, \dots, X_n]$  el vector de variables aleatorias.

- Coeficiente de correlación:

$$\rho(X, Y) := \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\text{var}(Y)}} \quad (3.13)$$

Para varias variables, la siguiente es la matriz de correlaciones:

$$\mathbf{R} = \begin{bmatrix} \rho(X_1, X_1) & \cdots & \rho(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \rho(X_n, X_1) & \cdots & \rho(X_n, X_n) \end{bmatrix} \quad (3.14)$$

- Si las variables aleatorias, por ejemplo  $Y_i$ , tienen varianzas unitarias, lo cual se logra con la transformación de estandarización (Bertsekas y Tsitsiklis, 2002):

$$Y_i = \frac{X_i - \mu_i}{\sigma_i} \quad (3.15)$$

entonces:

$$\mathbf{S} = \mathbf{R} \quad (3.16)$$

### 3.2.2. El método PCA

El método de análisis de componentes principales o PCA por sus siglas en inglés, se presenta a continuación.

Si se tienen  $N$  observaciones de las variables  $X_1, \dots, X_n$ , la matriz de datos

muestrales es:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nn} \end{bmatrix}_{N \times n} \quad (3.17)$$

En adelante se supone que  $\mathbf{X}$  es una matriz centrada, esto es, los promedios de cada variable es igual a cero, si no lo fuera, la transformación  $\mathbf{HX}$ , donde:

$$\mathbf{H} = \mathbf{I} - \frac{1}{N} \mathbf{1}_v \mathbf{1}_v^T \quad (3.18)$$

logra el centrado deseado (Grané Chávez, 2010).

De esta forma, si  $\mathbf{X} = [X_1, \dots, X_n]^T$  es el vector de variables aleatorias, con medias iguales a cero. Se desea encontrar una matriz ortogonal  $\mathbf{Q}$  tal que  $\mathbf{Z} = \mathbf{Q}^T \mathbf{X}$  (Maiti, 2014) tenga una matriz de covarianzas diagonal, entonces:

$$\begin{aligned} \text{cov}(\mathbf{Q}^T \mathbf{X}) &= \mathbf{E}[(\mathbf{Q}^T \mathbf{X})(\mathbf{Q}^T \mathbf{X})^T] \\ &= \mathbf{E}[\mathbf{Q}^T \mathbf{X} \mathbf{X}^T \mathbf{Q}] \\ &= \mathbf{Q}^T \mathbf{E}[\mathbf{X} \mathbf{X}^T] \mathbf{Q} \\ &= \mathbf{Q}^T \text{cov}(\mathbf{X}) \mathbf{Q} \\ &= \mathbf{Q}^T \mathbf{S} \mathbf{Q} \\ &= \mathbf{\Lambda} \end{aligned} \quad (3.19)$$

Como  $\mathbf{S}$  es simétrica real, con valores no negativos<sup>1</sup>, esto es  $\mathbf{S}$  es una matriz semidefinida positiva, con lo cual,  $\mathbf{Q}^T = \mathbf{Q}^{-1}$  es una matriz ortogonal y  $\mathbf{\Lambda}$  es una matriz diagonal (Strang, 2006).

Por lo tanto:

$$\mathbf{S} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \quad (3.20)$$

es la *descomposición espectral* de la matriz de covarianzas. En las columnas de  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_n]$  están los vectores característicos de  $\mathbf{S}$  y en la diagonal de  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$  sus valores característicos.

De esta forma, las *componentes principales* son:

$$\mathbf{Z} = \mathbf{Q}^T \mathbf{X} \quad (3.21)$$

---

<sup>1</sup>Ver la definición de  $\mathbf{S}$  en (3.11) según la definición de covarianzas de (3.9).

o, para cada componente principal:

$$Z_i = \mathbf{q}_i^T \mathbf{X} \quad (3.22)$$

### 3.2.3. Propiedades de las componentes principales

Las siguientes son algunas propiedades importantes de las componentes principales (Grané Chávez, 2010):

- Tienen **varianza decreciente**:

$$\begin{aligned} \text{var}(Z_i) &= \text{var}(\mathbf{q}_i^T \mathbf{X}) = \frac{1}{N} (\mathbf{q}_i^T \mathbf{X}) (\mathbf{q}_i^T \mathbf{X})^T = \frac{1}{N} \mathbf{q}_i^T \mathbf{X} \mathbf{X}^T \mathbf{q}_i \\ &= \mathbf{q}_i^T \mathbf{S} \mathbf{q}_i = \mathbf{q}_i^T \lambda_i \mathbf{q}_i = \lambda_i \end{aligned} \quad (3.23)$$

con  $\lambda_1 > \lambda_2 > \dots > \lambda_n$ .

- Están **incorrelacionadas**:

$$\text{cov}(Z_i, Z_j) = \text{cov}(\mathbf{q}_i^T \mathbf{X}, \mathbf{q}_j^T \mathbf{X}) = \mathbf{q}_i^T \mathbf{S} \mathbf{q}_j = \lambda_j \mathbf{q}_i^T \mathbf{q}_j = 0 \quad \forall i \neq j \quad (3.24)$$

- Las covarianzas entre cada componente principal  $Z$  y las variables originales  $X$  son:

$$\text{cov}(\mathbf{Z}, \mathbf{X}) = \frac{1}{n} \mathbf{Z} \mathbf{X}^T = \frac{1}{n} \mathbf{Q}^T \mathbf{X} \mathbf{X}^T = \mathbf{Q}^T \mathbf{S} = \mathbf{\Lambda} \mathbf{Q}^T \quad (3.25)$$

de donde,

$$\text{cov}(Z_i, \mathbf{X}) = \lambda_i \mathbf{q}_i^T \quad (3.26)$$

- La correlación entre  $Z_j$  y la variable original  $X_i$  es:

$$\rho(Z_j, X_i) = \frac{\text{cov}(Z_j, X_i)}{\sqrt{\text{var}(Z_j) \text{var}(X_i)}} = \frac{\lambda_j q_{ij}}{\sqrt{\lambda_j s_{ii}}} = q_{ij} \sqrt{\frac{\lambda_j}{s_{ii}}} \quad (3.27)$$

## 3.3. Modelos de entrenamiento

Los modelos a ser utilizados en este trabajo son: Regresión Lineal (LR: *Linear Regression*), Vector Soporte Automático (SVM: *Support Vector Machine*),

Redes neuronales artificiales (ANN: *Artificial Neural Networks*) y Aprendizaje de Automático Extremo (ELM: *Extreme Learning Machine*).

Para lo que sigue se considera un conjunto  $\mathbf{X}$  de  $N$  datos:

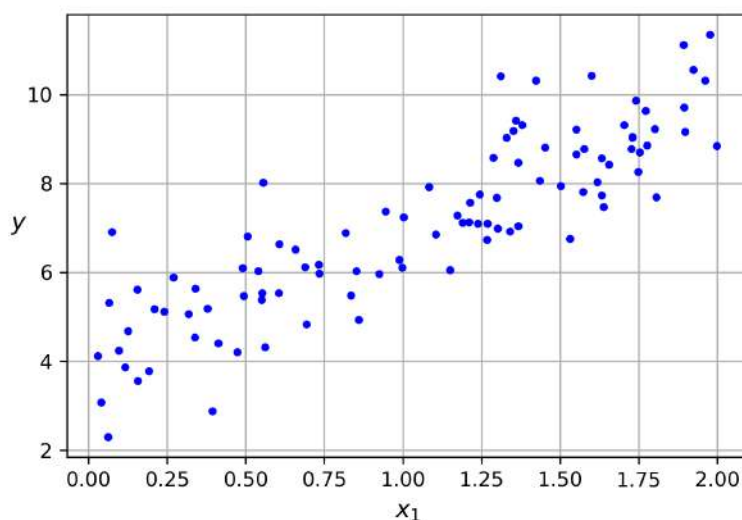
$$\mathbf{X} = \{(\mathbf{x}_d, \mathbf{y}_d) \in \mathbb{R}^n \times \mathbb{R}^m \mid d = 1, \dots, N\} \quad (3.28)$$

$\mathbf{x}_d$  es el vector de variables de entrada, dimensión  $n$  e  $\mathbf{y}_d$  es el vector de variables de salida, dimensión  $m$ .

### 3.3.1. Regresión lineal

#### 3.3.1.1. Fundamentos teóricos

Cada una serie de datos que siguen una tendencia de línea recta, como en la figura:



**Figura 3.1** – Datos con tendencia lineal. Se busca una recta tal que  $\hat{y} = w_0 + w_1x_1$ .  
Fuente: elaboración propia.

Es posible encontrar la recta o, en general un hiperplano de mínimos cuadrados, si se trata de varias variables, de la siguiente manera:

Se supone que la variable  $y$  depende de las variables  $x_j$ ,  $j = 1, \dots, n$ , esto es

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n \text{ según:}$$

$$y = f(x_1, \dots, x_n) = f(\mathbf{x}) \quad (3.29)$$

En regresión lineal se asume que cada valor  $y_i \in \mathbb{R}$  ( $i = 1, \dots, N$ ) puede relacionarse aproximadamente con sus correspondientes  $\mathbf{x}_i \in \mathbb{R}^n$  utilizando ecuaciones lineales de la forma:

$$\hat{y}_i = w_0 + w_1 x_{i1} + \dots + w_n x_{in} \quad (3.30)$$

siendo  $\hat{y}_i$  el valor predicho de  $y_i$ . A  $w_0 = b$  se lo conoce como umbral o *bias*. Para cada  $i$  la forma matricial de la ecuación (3.30) es:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Nn} \end{bmatrix}_{N \times (n+1)} \begin{bmatrix} w_0 \\ \vdots \\ w_n \end{bmatrix}_{(n+1) \times 1} = \mathbf{A}\mathbf{w} \quad (3.31)$$

Con esta aproximación, el error  $\boldsymbol{\epsilon} \in \mathbb{R}^N$  es:

$$\boldsymbol{\epsilon} := \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{A}\mathbf{w} \quad (3.32)$$

Desde el punto de vista del Álgebra Lineal (Strang, 2006), ese error existe porque el vector  $\mathbf{y}$  no se encuentra en el espacio columna de  $\mathbf{A}$ , esto es  $\mathbf{y} \notin C(\mathbf{A})$ . La norma  $\|\boldsymbol{\epsilon}\|$  será mínima si  $\boldsymbol{\epsilon}$  es ortogonal a  $C(\mathbf{A})$ . Ahora bien según el teorema fundamental del álgebra lineal el complemento ortogonal del espacio columna es el espacio nulo izquierdo  $N(\mathbf{A}^T)$ , es decir:  $N(\mathbf{A}^T) = (C(\mathbf{A}))^\perp$ , por tanto  $\boldsymbol{\epsilon} \in N(\mathbf{A}^T)$  con lo cual el error será mínimo. Considerando estos hechos, se debe cumplir:

$$\mathbf{A}^T \boldsymbol{\epsilon} = \mathbf{0} \quad (3.33)$$

Tomando en cuenta (3.32):

$$\mathbf{A}^T \mathbf{A} \hat{\mathbf{w}} = \mathbf{A}^T \mathbf{y} \quad (3.34)$$

De donde:

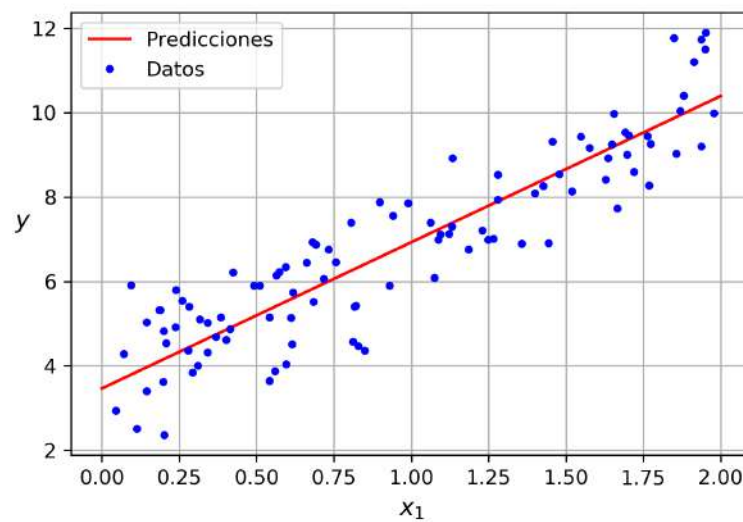
$$\hat{\mathbf{w}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \quad (3.35)$$

El error cuadrático medio MSE resulta por tanto:

$$\text{MSE} = \frac{1}{n} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{w}}\|^2 \quad (3.36)$$

La raíz cuadrada del MSE se denota por RMSE.

Aplicar estos conceptos al caso de la figura 3.1 devuelve la regresión lineal deseada:



**Figura 3.2** – Predicciones con el modelo de regresión lineal. *Fuente:* elaboración propia.

A pesar de que los gráficos muestran un caso de dos dimensiones, las ecuaciones presentadas acá son aplicables a varias variables, para casos de regresión lineal múltiple (Miller, Freund y Johnson, 1992).

### 3.3.1.2. Algoritmo de regresión lineal

Dados los datos  $y_i = f(x_{i1}, \dots, x_{in})$   $i = 1, \dots, N$  que conforman las componentes del vector  $\mathbf{y} \in \mathbb{R}^N$ :

1. Construir la matriz  $\mathbf{A} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Nn} \end{bmatrix}$  dada en (3.31)

2. Calcular  $\hat{\mathbf{w}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}$  según se mostró en (3.35).

3. La predicción de  $y$  para algún  $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$  se obtiene con  $\hat{y} = \hat{\mathbf{w}}^T \mathbf{x}'$ , siendo

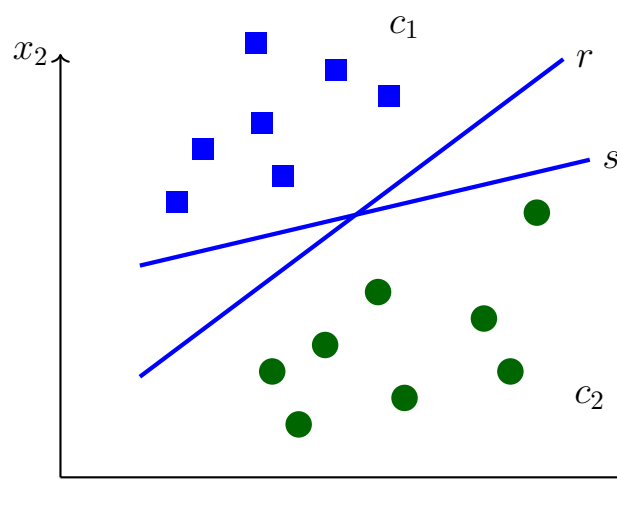
$$\mathbf{x}' = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

### 3.3.2. Vector soporte automático

*Support vector machine* o vector soporte automático (traducción que se ha adoptado) es una técnica muy versátil para elaborar superficies de separación de manera a realizar clasificaciones, la misma se puede generalizar para análisis de regresión. Se inicia con el caso más simple de separación en dos grupos, hasta llegar al análisis de regresión general utilizando esta técnica.

#### 3.3.2.1. Perfectamente separables linealmente

Se trata de separar dos grupos de datos, en la figura siguiente se muestran dos alternativas utilizando líneas rectas:



**Figura 3.3** – Si se desea separar dos grupos de datos con rectas,  $r$  y  $s$  son dos alternativas. En el SVM se busca la mejor de ellas. *Fuente:* elaboración propia.

Se toma, para aclarar conceptos, la recta  $r$ . Se define la siguiente función lineal (Biswas, 2014):

$$g(\mathbf{x}) := \mathbf{w}^T \mathbf{x} + b \quad (3.37)$$

en donde  $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$ ;  $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$  y  $g(\mathbf{x}) = 0$  representa la recta (o el hiperplano)  $r$ , por lo tanto, el vector  $\mathbf{w}$  es perpendicular a ella. Si un punto  $\mathbf{x} \notin r$ , es posible calcular su distancia a la línea con la ecuación:

$$d_{\mathbf{x} \rightarrow r} = \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} \quad (3.38)$$

Esta distancia será positiva,  $\text{sgn}(d_{\mathbf{x} \rightarrow r}) = +1$ , si el punto  $\mathbf{x}$  y el origen están situados a diferentes lados de la recta y será negativa,  $\text{sgn}(d_{\mathbf{x} \rightarrow r}) = -1$ , en caso contrario (Kletenik, 1979), matemáticamente:

$$\begin{cases} \mathbf{w}^T \mathbf{x} + b > 0 & \text{para los puntos } c_1 \text{ de la figura 3.3;} \\ \mathbf{w}^T \mathbf{x} + b = 0 & \text{sobre la recta } r \\ \mathbf{w}^T \mathbf{x} + b < 0 & \text{para los puntos } c_2. \end{cases}$$

Por lo tanto, se puede determinar si un punto está a uno u otro lado de la recta  $r$  de acuerdo al signo de  $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ , de esta manera la función de decisión es:

$$G(\mathbf{x}) := \text{sgn}[g(\mathbf{x})] = \text{sgn}(\mathbf{w}^T \mathbf{x} + b) \quad (3.39)$$

conforme al cual se define una variable  $y_i$ ,  $i = 1, \dots, N$  para cada punto  $\mathbf{x}_i$  tal que:

$$y_i = \begin{cases} +1 & \text{si } \mathbf{w}^T \mathbf{x}_i + b > 0 \\ -1 & \text{si } \mathbf{w}^T \mathbf{x}_i + b < 0 \end{cases} \quad (3.40)$$

por lo tanto, en cualquiera de los dos casos anteriores, se tendrá:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) > 0 \quad (3.41)$$

Ahora bien, de forma que la línea esté en la mejor posición, entre los grupos de puntos, es decir, aproximadamente “en medio” de ambos grupos, con el mayor ancho de margen posible, se debe conseguir que la distancia  $\nu$  de separación entre la recta y los puntos más próximos a ella sea el máximo posible, de esta forma

para todos los puntos se debe cumplir:

$$d_{\mathbf{x} \rightarrow r} \geq \nu \quad (3.42)$$

De acuerdo a (3.38), esto es:

$$\frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} \geq \nu \quad \implies \quad \mathbf{w}^T \mathbf{x} + b \geq \|\mathbf{w}\| \nu \quad (3.43)$$

Se elige la escala de medición tal que:

$$\|\mathbf{w}\| \nu = 1 \quad (3.44)$$

con lo cual:

$$\begin{cases} \mathbf{w}^T \mathbf{x} + b \geq 1 & \text{para los puntos } c_1 \text{ de la figura 3.3;} \\ \mathbf{w}^T \mathbf{x} + b \leq -1 & \text{para los puntos } c_2. \end{cases}$$

y para ambos casos, según (3.41):

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad (3.45)$$

De la ecuación (3.44), según la convención de escala adoptada, la distancia  $\nu$  de la recta  $r$  a los puntos más cercanos a ella debe ser:

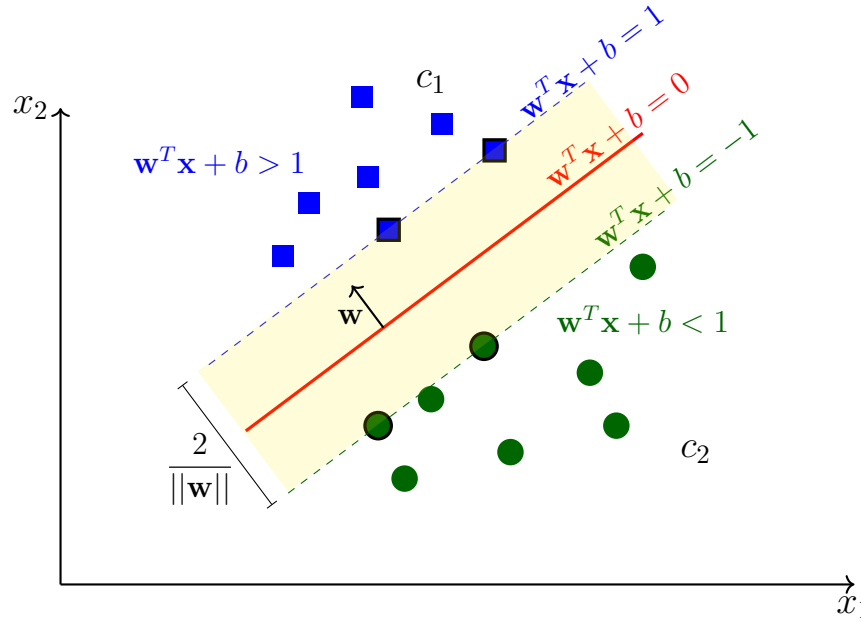
$$\nu = \frac{1}{\|\mathbf{w}\|} \quad (3.46)$$

El ancho de la “franja de dominio”, es consecuentemente  $\frac{2}{\|\mathbf{w}\|}$ .

Los vectores soporte automático son aquellos que están en los límites de la franja de dominio, se tiene entonces:

$$\begin{cases} |\mathbf{w}^T \mathbf{x} + b| = 1 & \text{para los vectores soporte;} \\ |\mathbf{w}^T \mathbf{x} + b| > 1 & \text{para los demás puntos;} \\ |\mathbf{w}^T \mathbf{x} + b| < 1 & \text{en el interior de la franja;} \\ \mathbf{w}^T \mathbf{x} + b = 0 & \text{para la recta de separación.} \end{cases} \quad (3.47)$$

Estos hechos se muestran en la figura 3.4:



**Figura 3.4** – Ancho de la “franja de dominio” de la recta de separación. Los vectores de soporte están en los límites de la franja. *Fuente:* elaboración propia.

Se ha presentado la idea general pero hasta ahora no se ha determinado la ecuación de la línea separadora, la línea ideal es la que maximiza la distancia  $\nu$  que, conforme a (3.43), implica minimizar  $\|\mathbf{w}\|$  y maximizar  $b$ . Por una conveniencia matemática, en vez de buscar minimizar  $\|\mathbf{w}\|$  se toma su equivalente, minimizar  $\frac{1}{2}\|\mathbf{w}\|^2$ , de acuerdo a esto, el objetivo es:

$$\begin{aligned} \underset{\mathbf{w}, b}{\text{minimizar:}} & \quad \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{sujeto a:} & \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i \end{aligned} \quad (3.48)$$

Este es un problema de optimización convexa (criterio cuadrático con restricciones de desigualdades lineales), con el cual, la forma *primal* de la función de Lagrange a ser minimizada es (Hastie y col., 2005):

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1] \quad (3.49)$$

donde los  $\alpha_i$  son los multiplicadores de Lagrange.

Si  $(\mathbf{w}, b, \boldsymbol{\alpha})$  es un punto silla de (3.49), entonces  $(\mathbf{w}, b)$  es un óptimo de (3.48)  $\forall \alpha_i \geq 0$  (Lauritzen, 2013).

La forma expandida de la ecuación (3.49) es:

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^N \alpha_i y_i b + \sum_{i=1}^N \alpha_i$$

Tomando derivadas parciales respecto de  $\mathbf{w}$  y  $b$ , se tiene:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^N \alpha_i y_i = 0$$

Con lo cual:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (3.50)$$

Sustituyendo estas relaciones en (3.49) se obtiene la llamada forma dual de Wolfe, que depende solo de los multiplicadores de Lagrange:

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (3.51)$$

El problema dual es:

$$\begin{aligned} & \underset{\alpha}{\text{maximizar:}} && L_D \\ & \text{sujeto a:} && \alpha_i \geq 0 \end{aligned} \quad (3.52)$$

Esto es, encontrar un máximo de  $L_D$  para valores positivos de  $\alpha_i$ , el cual es un problema de optimización más fácil de resolver. Se remarca que, además, deben cumplirse las condiciones de Karush-Kuhn-Tucker (Fox, 1971) que son las ecuaciones (3.50), las restricciones de (3.52) y:

$$\alpha_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1] = 0 \quad (3.53)$$

Notar de (3.53) que:

- Si  $\alpha_i > 0$ , entonces  $(\mathbf{w}^T \mathbf{x}_i + b) = 1$  y el punto  $\mathbf{x}_i$  está sobre los límites de la franja de dominio;
- Si  $(\mathbf{w}^T \mathbf{x}_i + b) > 1$ ,  $\mathbf{x}_i$  está fuera de la franja de dominio y  $\alpha_i = 0$

### 3.3.2.2. No perfecta ni linealmente separables

El caso mostrado hasta acá es válido cuando ninguno de ellos están dentro de la franja de dominio, para considerar posibles “errores”, es decir, la posibilidad de que alguno o algunos de los puntos caigan en la franja de dominio o en el lado incorrecto de clasificación, se definen las variables que consideran esa posibilidad como  $\xi_i$ ,  $i = 1, \dots, n$ , así, si se limita la suma  $\sum \xi_i$  a algún valor positivo  $K$ , se estará limitando la cantidad de puntos que son admitidos en el lado equivocado del hiperplano de separación. Bajo esta consideración, las ecuaciones de optimización (3.48) se transforman en:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimizar:}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{sujeto a:} \quad & \begin{cases} y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i; \\ \xi_i \geq 0 \\ \sum_{i=1}^N \xi_i \leq K \end{cases} \end{aligned} \quad (3.54)$$

Computacionalmente, es mejor reescribir (3.54) de la forma equivalente:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimizar:}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{sujeto a:} \quad & \begin{cases} y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i; \\ \xi_i \geq 0 \end{cases} \end{aligned} \quad (3.55)$$

donde  $C$  es el parámetro de costo que le da un peso a las clasificaciones equivocadas.

La función primal lagrangiana resulta en este caso:

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - (1 - \xi_i)] - \sum_{i=1}^N \mu_i \xi_i \quad (3.56)$$

Tomando derivadas parciales respecto de  $\mathbf{w}$ ,  $b$  y  $\xi_i$  se tiene:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \\ \sum_{i=1}^N \alpha_i y_i = 0 \\ \alpha_i = C - \mu_i \quad \forall i \end{cases} \quad (3.57)$$

Llevando estas expresiones a (3.56) el problema de optimización se convierte en:

$$\begin{aligned} \text{maximizar: } & L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{sujeto a: } & \begin{cases} 0 \leq \alpha_i \leq C \\ \alpha_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - (1 - \xi_i)] = 0 \\ \mu_i \xi_i = 0 \\ y_i (\mathbf{w}^T \mathbf{x}_i + b) - (1 - \xi_i) \geq 0 \end{cases} \end{aligned} \quad (3.58)$$

### 3.3.2.3. Separación no lineal. Kernels

Cuando se trata de realizar la clasificación en grupos que no son linealmente separables, es decir, cuando no es posible dividir los grupos utilizando hiperplanos, se propone utilizar superficies. En este caso, la idea central reside en buscar unas funciones de transformación no lineal, con las cuales se transforman los vectores originales a otro espacio vectorial en el que sean separables los puntos dados, conforme a las ecuaciones presentadas precedentemente, se buscan unas funciones base  $h_k(\mathbf{x})$ ,  $k = 1, \dots, r$ , esto es,  $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^r$  ( $r > n$ ) tal que el vector  $\mathbf{h}(\mathbf{x}_i) = [h_1(\mathbf{x}_i), \dots, h_r(\mathbf{x}_i)]$ ,  $i = 1, \dots, N$  es utilizado en reemplazo de  $\mathbf{x}$  en la función  $g(\mathbf{x})$  que se transforma por tanto en  $g(\mathbf{h}(\mathbf{x})) = \mathbf{w}^T \mathbf{h}(\mathbf{x}) + b$  y el clasificador resulta de esta manera ser:

$$\hat{G}(\mathbf{x}) := \text{sgn} [g(\mathbf{h}(\mathbf{x}))] = \text{sgn} (\mathbf{w}^T \mathbf{h}(\mathbf{x}) + b) \quad (3.59)$$

como en (3.39).



o, matricialmente:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_r(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_r(\mathbf{x}_N) \end{bmatrix}_{N \times r} \begin{bmatrix} w_1 \\ \vdots \\ w_r \end{bmatrix}_{r \times 1} = \mathbf{H}\mathbf{w} \quad (3.63)$$

el error es:

$$\boldsymbol{\epsilon} := \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{H}\mathbf{w} \quad (3.64)$$

el cuadrado de la norma euclídea es:

$$\|\boldsymbol{\epsilon}\|^2 = (\mathbf{y} - \mathbf{H}\mathbf{w})^T (\mathbf{y} - \mathbf{H}\mathbf{w}) \quad (3.65)$$

Para estimar los valores de los coeficientes  $w_i$  se escribe el criterio de mínimos cuadrados penalizados:

$$H(\mathbf{w}) = (\mathbf{y} - \mathbf{H}\mathbf{w})^T (\mathbf{y} - \mathbf{H}\mathbf{w}) + \lambda \|\mathbf{w}\|^2 \quad (3.66)$$

tomando derivadas respecto de  $\mathbf{w}$ :

$$\frac{\partial H(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{H}^T (\mathbf{y} - \mathbf{H}\hat{\mathbf{w}}) + 2\lambda\hat{\mathbf{w}} = \mathbf{0}$$

pre-multiplicando por  $\mathbf{H}$  y ordenando términos se tiene:

$$\hat{\mathbf{y}} = \mathbf{H}\hat{\mathbf{w}} = (\mathbf{H}\mathbf{H}^T + \lambda\mathbf{I}) \mathbf{H}\mathbf{H}^T \mathbf{y} \quad (3.67)$$

Por la definición de kernel dada en (3.61) y de  $\mathbf{H}$  dada en (3.63), se define la matriz simétrica de Gram:

$$\mathbf{G} := \mathbf{H}\mathbf{H}^T = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_N, \mathbf{x}_1) & \cdots & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \quad (3.68)$$

En conclusión, no es necesario el cálculo de las transformaciones  $h_k(\mathbf{x}_i)$ , ya que la aproximación (3.67) se consigue a partir de la matriz Kernel dada en (3.68).

### 3.3.2.5. El método LS-SVM

Se presenta el método LS-SVM: *Least Square Support Vector Machine*, o, mínimos cuadrados de vectores soporte automático.

Se considera el siguiente modelo de regresión no lineal (Caicedo y col., 2019):

$$\hat{f}(\mathbf{x}) = \mathbf{w}^T \mathbf{h}(\mathbf{x}) + b \quad (3.69)$$

Según las definiciones ya presentadas de cada uno de los términos.

El modelo LS-SVM se desarrolla según Suykens (2002, pág. 98-100). El truco está en modificar levemente el planteamiento de optimización dado en (3.55) como sigue:

$$\underset{\mathbf{w}, b, \xi}{\text{minimizar:}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \gamma \frac{1}{2} \sum_{i=1}^n \xi_i^2 \quad (3.70)$$

$$\text{sujeto a:} \quad \left\{ y_i = \mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b + \xi_i \quad \forall i; \right.$$

El langrangiano del problema es, consecuentemente:

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 + \gamma \frac{1}{2} \sum_{i=1}^N \xi_i^2 - \sum_{i=1}^N \alpha_i [\mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b + \xi_i - y_i] \quad (3.71)$$

Tomando derivadas, para encontrar el punto silla de (3.71), se tiene:

$$\left\{ \begin{array}{l} \frac{\partial L_P}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i \mathbf{h}(\mathbf{x}_i) \\ \frac{\partial L_P}{\partial b} = 0 \rightarrow \sum_{i=1}^n \alpha_i = 0 \\ \frac{\partial L_P}{\partial \xi_i} = 0 \rightarrow \alpha_i = \gamma \xi_i \quad \forall i \\ \frac{\partial L_P}{\partial \alpha_i} = 0 \rightarrow \mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b + \xi_i - y_i = 0 \quad \forall i \end{array} \right. \quad (3.72)$$

A continuación se definen los siguientes vectores en  $\mathbb{R}^N$ :

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}; \quad \mathbf{1}_v = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}; \quad \boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_N \end{bmatrix}; \quad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} \quad (3.73)$$

Eliminando  $\mathbf{w}$  y  $\boldsymbol{\xi}$  de las ec. (3.72) se obtiene el siguiente sistema de ecuaciones

lineales en  $\boldsymbol{\alpha}$  y  $b$ :

$$\left[ \begin{array}{c|c} 0 & \mathbf{1}_v^T \\ \hline \mathbf{1}_v & \mathbf{G} + \mathbf{I}/\gamma \end{array} \right]_{(N+1) \times (N+1)} \left[ \begin{array}{c} b \\ \boldsymbol{\alpha} \end{array} \right]_{(N+1) \times 1} = \left[ \begin{array}{c} 0 \\ \mathbf{y} \end{array} \right]_{(N+1) \times 1} \quad (3.74)$$

En donde  $\mathbf{G}$  es la matriz de Gram que fue presentada en (3.68). El sistema es cuadrado y de solución única.

Finalmente el modelo de LS-SVM para estimación de funciones  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  es:

$$\hat{f}(\mathbf{x}) = b + \sum_{i=1}^N \alpha_i K(\mathbf{x}, \mathbf{x}_i) \quad (3.75)$$

en donde  $b$  y  $\alpha_i$  son soluciones del sistema (3.74).

### 3.3.2.6. Algoritmo LS-SVM

Dado un conjunto de datos  $\mathbf{X} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$  en donde  $(\mathbf{x}_d, y_d) \in \mathbb{R}^n \times \mathbb{R}$

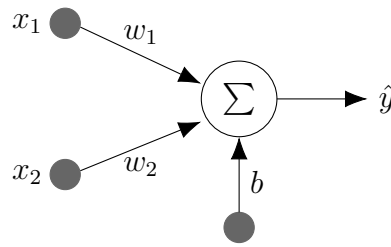
1. Calcular la matriz de Gram dada en (3.68) para la función kernel adoptada;
2. Resolver el sistema (3.74) con un valor adecuado de  $\gamma$  para obtener  $b$  y  $\alpha_i$ ;
3. Estimar la función  $f(\mathbf{x})$  con (3.75).

### 3.3.3. Redes neuronales artificiales

Las redes neuronales artificiales o *Artificial Neural Networks* están inspiradas en las neuronas o células nerviosas de los seres vivos. Básicamente una neurona tiene como misión recibir información de otras neuronas o de receptores, procesa esta información en un código de activación interno y luego transmite nuevos impulsos a las neuronas siguientes u otras células.

#### 3.3.3.1. Perceptrón

Una neurona artificial opera de manera similar, recibe unos valores de entrada ponderados, los procesa y emite una salida.



**Figura 3.5** – Arquitectura básica de un PERCEPTRÓN. *Fuente:* elaboración propia.

Conforme a la figura 3.5 los valores de entrada son  $x_i$ , los pesos  $w_i$ , la salida es  $y$  y existe un parámetro adicional  $b$  que es el umbral o *bias*, ya mencionado anteriormente. A continuación se puede escribir:

$$a := \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^n w_i x_i + b \quad (3.76)$$

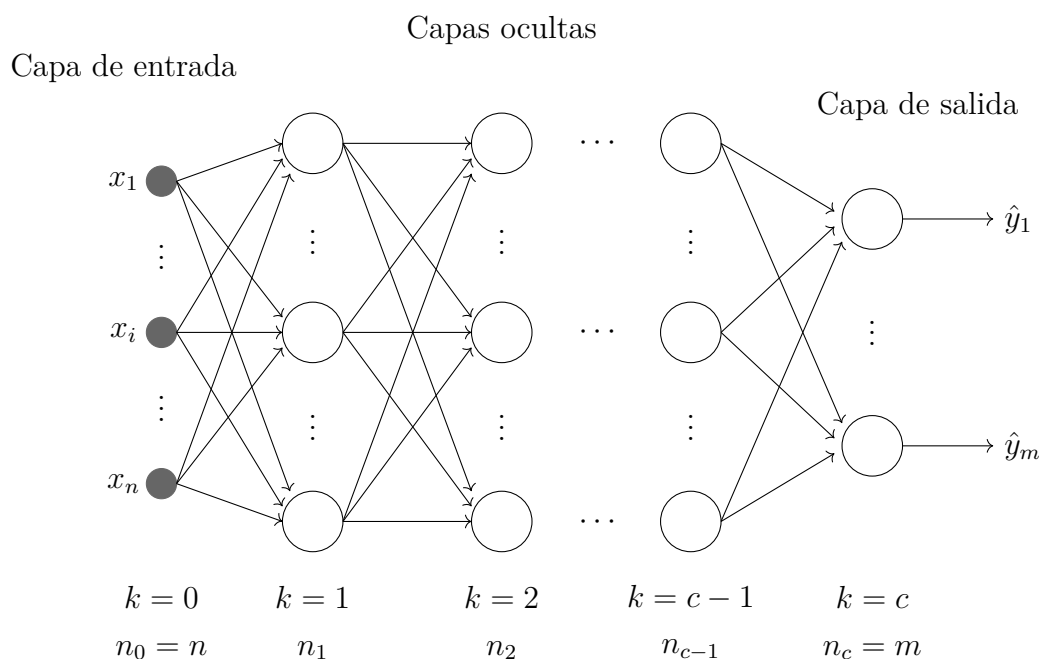
En donde:  $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$  son los valores de entrada y  $\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$  el vector de pesos.

La función:

$$g(a) = \hat{y} \quad (3.77)$$

es llamada *función de activación*.

## 3.3.3.2. Perceptrón multicapa



**Figura 3.6** – Arquitectura del PERCEPTRÓN multicapa. *Fuente:* elaboración propia.

Observando la figura 3.6:

- $x_i$ : valores de entrada  $i = 1, \dots, n$ ;
- $n_k$ : número de neuronas, capa  $k = 0, \dots, c$ ;
- $c + 1$ : número total de capas, la de entrada (capa 0), las ocultas (capas  $k = 1, \dots, c - 1$ ) y la de salida (capa  $c$ ); el número de capas ocultas es  $c - 1$ .
- $\hat{y}_j$ : valores de salida  $j = 1, \dots, m$ .

Además, se utiliza la siguiente nomenclatura:

- $w_{ij}^k, k > 0$ : peso de conexión entre el nudo  $i$  y el nudo  $j$  de las capas  $k - 1$  y  $k$  respectivamente;
- $b_i^k$ : bias del nudo  $i$ , capa  $k > 0$ ;
- $g$ : función de activación.

Formalmente, se tiene lo siguiente (McGonagle y col., 2019) :

- **Conjunto de entrenamiento:** Es un conjunto de pares de vectores, de entrada  $\mathbf{x}_d$ , y de salida  $\mathbf{y}_d$ , de tamaño  $N$  denotado por:

$$\mathbf{X} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\} \quad (3.78)$$

- **Red neuronal:** como la figura 3.6, cuyos parámetros son denotados por  $\theta$  de manera colectiva. Tiene pesos  $w_{ij}^k$  de conexión entre el nudo  $i$  de la capa  $k - 1$  al nudo  $j$  de la capa  $k$ , y  $b_j^k$  el bias de este último nudo<sup>2</sup>. No hay conexiones entre neuronas de la misma capa, sin embargo todas las de una capa están conectadas a todas las de la capa siguiente<sup>3</sup>.
- **Función de error:**  $E(\mathbf{X}, \theta)$  representa el error promedio entre la salida deseada  $\mathbf{y}$ , respecto a la calculada  $\hat{\mathbf{y}}^t$  en la iteración  $t$ . Es promedio de los errores de cada par  $(\mathbf{x}_d, \mathbf{y}_d)$  del conjunto de entrenamiento  $\mathbf{X}$  para un particular conjunto de valores  $\theta$ .

Los parámetros  $\theta$  se actualizan en cada iteración utilizando la técnica descenso estocástico del gradiente:

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial E(\mathbf{X}, \theta)}{\partial \theta} \quad (3.79)$$

Donde  $\theta^t$  es el valor del parámetro  $\theta$  en la iteración  $t$  y  $\alpha$  es la llamada *razón de aprendizaje*.

Este proceso se realiza con el objetivo de minimizar el error hasta un valor aceptable. El error adoptado es el de mínimos cuadrados definido como sigue:

$$\epsilon_d := \frac{1}{2} \|\mathbf{y}_d - \hat{\mathbf{y}}_d\|^2 \quad (3.80)$$

el cual corresponde al par  $(\mathbf{x}_d, \mathbf{y}_d)$ . El error promedio para todo el conjunto de entrenamiento es:

$$E(\mathbf{X}, \theta) := \frac{1}{N} \sum_{d=1}^N \epsilon_d = \frac{1}{2N} \sum_{d=1}^N \|\mathbf{y}_d - \hat{\mathbf{y}}_d\|^2 \quad (3.81)$$

<sup>2</sup>Notar que los parámetros  $\theta$  son precisamente  $w_{ij}^k$  y  $b_j^k$ .

<sup>3</sup>Red neuronal completamente conectada.

Su derivada respecto a los pesos  $w_{ij}^k$  es:

$$\frac{\partial E(\mathbf{X}, \theta)}{\partial w_{ij}^k} = \frac{1}{N} \sum_{d=1}^N \frac{\partial \epsilon_d}{\partial w_{ij}^k} \quad (3.82)$$

Conforme se vio en (3.76) es posible escribir:

$$a_j^k := \sum_{i=1}^{n_{k-1}} w_{ij}^k o_i^{k-1} + b_j^k \quad (3.83)$$

Para simplificar la matemática, si  $b_j^k = w_{0j}^k$  y  $o_0^{k-1} = 1$  para el nudo  $0^4$  de la capa  $k - 1$ , entonces la ecuación anterior (3.83) pasa a ser:

$$a_j^k := \sum_{i=0}^{n_{k-1}} w_{ij}^k o_i^{k-1} \quad (3.84)$$

con los cuales la salida del nudo  $i$  de la capa  $k$ , según se presentó en (3.77), es:

$$o_i^k := g(a_i^k) \quad (3.85)$$

Notar que para la capa de entrada  $k = 0$ :

$$o_i^0 = x_i \quad (3.86)$$

La derivada del error  $\epsilon$  es:

$$\frac{\partial \epsilon}{\partial w_{ij}^k} = \frac{\partial \epsilon}{\partial a_j^k} \frac{\partial a_j^k}{\partial w_{ij}^k} \quad (3.87)$$

Se adopta la notación:

$$\delta_j^k := \frac{\partial \epsilon}{\partial a_j^k} \quad (3.88)$$

además se observa que:

$$\frac{\partial a_j^k}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k} \left( \sum_{l=0}^{n_{k-1}} w_{lj}^k o_l^{k-1} \right) = o_i^{k-1} \quad (3.89)$$

con los cuales (3.87) se puede escribir:

$$\frac{\partial \epsilon}{\partial w_{ij}^k} = \delta_j^k o_i^{k-1} \quad (3.90)$$

---

<sup>4</sup>Nudo del bias, no representado en la figura 3.6.

Así, la derivada del error  $\epsilon$  para una muestra dada respecto a  $w_{ij}^k$ , es el producto del término  $\delta_j^k$  de la neurona  $j$  de la capa  $k$  multiplicado por la salida  $o_i^{k-1}$  de la neurona  $i$  de la capa anterior  $k-1$ , lo cual tiene sentido considerando que  $w_{ij}^k$  es el peso de conexión entre la neurona  $i$  (de salida) y la neurona  $j$  (de llegada) de las capas  $k-1$  y  $k$  respectivamente.

### Capa de salida

Ahora bien, en la capa de salida ( $k=c$ ), de acuerdo a (3.80):

$$\epsilon = \frac{1}{2} \sum_{i=1}^m (y_i - \hat{y}_i)^2 = \frac{1}{2} \sum_{i=1}^m (y_i - g(a_i^c))^2 \quad (3.91)$$

en base al cual se puede aplicar la ecuación (3.88) para obtener:

$$\delta_j^c = \frac{\partial \epsilon}{\partial a_j^c} = - \sum_{i=1}^m (y_i - g(a_i^c)) g'(a_i^c) \frac{\partial a_i^c}{\partial a_j^c} \quad (3.92)$$

como

$$\frac{\partial a_i^c}{\partial a_j^c} = \begin{cases} 1 & \text{si } i = j; \\ 0 & \text{si } i \neq j \end{cases} \quad (3.93)$$

$$\delta_j^c = -g'(a_j^c) (y_j - g(a_j^c)) \quad (3.94)$$

Por lo tanto, de (3.90) y (3.94) la derivada del error  $\epsilon$  es:

$$\frac{\partial \epsilon}{\partial w_{ij}^c} = -g'(a_j^c) o_i^{c-1} (y_j - g(a_j^c)) \quad (3.95)$$

### Capas ocultas

Nuevamente, conforme a (3.88), para  $1 \leq k < c$  es posible escribir:

$$\delta_j^k = \frac{\partial \epsilon}{\partial a_j^k} = \sum_{l=1}^{n_{k+1}} \frac{\partial \epsilon}{\partial a_l^{k+1}} \frac{\partial a_l^{k+1}}{\partial a_j^k} = \sum_{l=1}^{n_{k+1}} \delta_l^{k+1} \frac{\partial a_l^{k+1}}{\partial a_j^k} \quad (3.96)$$

observando la definición dada en (3.84) se obtiene la derivada siguiente:

$$\frac{\partial a_l^{k+1}}{\partial a_j^k} = w_{jl}^{k+1} g'(a_j^k) \quad (3.97)$$

con la cual se tiene la fórmula que da el nombre de *retropropagación* a este método:

$$\delta_j^k = g'(a_j^k) \sum_{l=1}^{n_{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (3.98)$$

ya que el valor  $\delta_j^k$  de la capa  $k$  depende de los valores  $\delta_l^{k+1}$  de la capa siguiente.

Finalmente, la derivada del error  $\epsilon$ , de acuerdo a (3.90) es:

$$\frac{\partial \epsilon_d}{\partial w_{ij}^k} = g'(a_j^k) o_i^{k-1} \sum_{l=1}^{n_{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (3.99)$$

### 3.3.3.3. Estructura de la red neuronal

Al momento de llevar adelante un entrenamiento con redes neuronales artificiales, se debe decidir, entre otras cosas, el número de capas ocultas y el número de neuronas en cada capa.

- **Número de capas ocultas:** El *Teorema de Aproximación Universal*, cuya primera versión para funciones sigmoide fue probada por Cybenko (1989), establece que una red neuronal con una sola capa oculta y un número finito de neuronas, puede aproximar a cualquier función continua en un subconjunto compacto de  $\mathbb{R}^n$ , por lo que, es probable, una sola capa oculta sea suficiente dependiendo de la complejidad del problema. Si puede subdividirse en partes menos complejas, cada subdivisión puede ser representada por una capa oculta adicional, aunque según Heaton (2008) es muy raro encontrar problemas que requieran dos capas ocultas;
- **Número de neuronas:** Para prevenir el sobre-entrenamiento, en Frontline-Systems (2019) se propone la siguiente fórmula:

$$n_h = \frac{N}{\alpha(n + m)} \quad (3.100)$$

En donde:

- $n_h$ : número de neuronas en la capa oculta;
- $N$ : número de muestras en la base de datos de entrenamiento;
- $n$ : número de neuronas en la capa de entrada;
- $m$ : número de neuronas en la capa de salida;
- $\alpha$  factor de escala arbitrario, usualmente entre 2 y 10.

### 3.3.3.4. Algoritmo de retropropagación

Según sugiere McGonagle y col. (2019).

Dados un conjunto de valores aleatorios iniciales  $w_{ij}^k$  y una tasa de aprendizaje adecuada  $\alpha$ :

1. **Fase de cálculo hacia adelante:** para cada par  $(\mathbf{x}_d, \mathbf{y}_d)$  del conjunto de entrenamiento  $\mathbf{X}$  realizar los cálculos con valores iniciales aleatorios de  $w_{ij}^k$  para obtener  $\hat{\mathbf{y}}_d$ . Guardar los valores  $a_j^k$ , y  $o_j^k$  de cada nudo  $j$ , desde la capa  $k = 0$  hasta la capa  $k = c$ ;
2. **Fase de retropropagación:**
  - a) Evaluar el error  $\delta_j^c$  de la última capa, según (3.94);
  - b) Retropropagar los errores de las capas ocultas, esto es calcular  $\delta_j^k$  dado en (3.98);
  - c) Evaluar las derivadas de  $\epsilon_d$  según (3.90).
3. **Cálculo del gradiente total:**  $\frac{\partial E(\mathbf{X}, \theta)}{\partial w_{ij}^k}$  conforme a (3.82) para todo el conjunto de entrenamiento  $\mathbf{X} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ ;
4. **Actualización de los pesos  $w_{ij}^k$**  agregándoles el término:

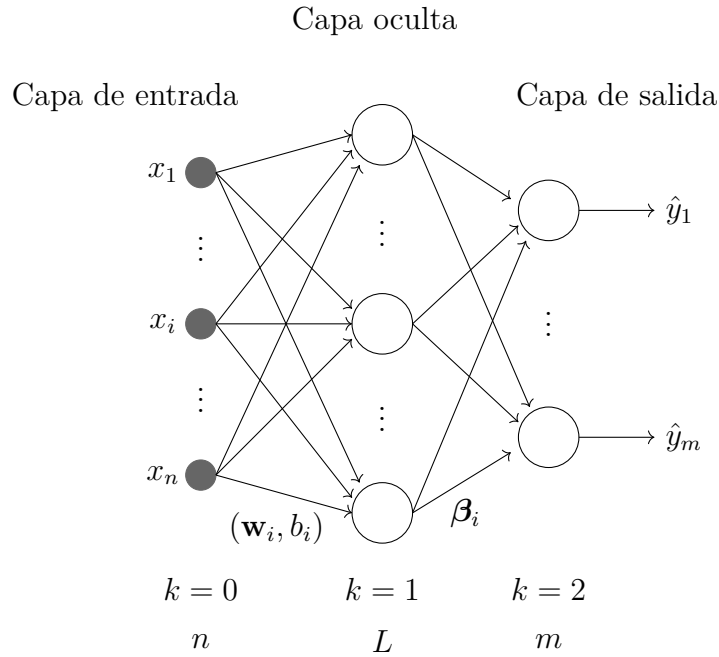
$$\Delta w_{ij}^k = -\alpha \frac{E(\mathbf{X}, \theta)}{\partial w_{ij}^k} \quad (3.101)$$

según se indica en la ecuación (3.79).

5. **Iteración**, repetir el proceso hasta llegar a minimizar el error a niveles aceptables.

### 3.3.4. Aprendizaje automático extremo

El método de Aprendizaje automático extremo o *Extreme Learning Machine (ELM)*, es una red neuronal artificial con una sola capa oculta *Single-hidden Layer Feedforward Network (SLFN)*, de ahí el nombre de “extremo”, la red neuronal se reduce al extremo de tener una sola capa oculta (Guan-Bin Huang, 2013).



**Figura 3.7** – Arquitectura del SLFN. *Fuente:* elaboración propia.

Su bondad está en tener *capacidad de aproximador universal*, conforme al siguiente teorema probado por Guang-Bin Huang, Lei y Siew (2006):

**Teorema 1.** *Dada cualquier función no-constante seccionalmente continua como función de activación, si ajustando los parámetros de las neuronas se puede hacer que las redes neuronales de una sola capa oculta aproximen cualquier función  $f(\mathbf{x})$ , entonces la secuencia  $\{h_i(\mathbf{x})\}_{i=1}^L$  puede ser generada aleatoriamente conforme a cualquier distribución continua de probabilidad, y  $\lim_{L \rightarrow \infty} \|\sum_{i=1}^L \beta_i h_i(\mathbf{x}) - f(\mathbf{x})\|$  se cumple con probabilidad uno, con la apropiada selección de los pesos  $\beta$ .*

### 3.3.4.1. Teoría básica

Conforme al teorema 1, se puede establecer lo siguiente: Para un conjunto de entrenamiento  $\mathbf{X} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$  de tamaño  $N$ , en donde  $(\mathbf{x}_d, \mathbf{y}_d) \in \mathbb{R}^n \times \mathbb{R}^m$ , la red SLFN con  $L$  neuronas en la capa oculta y función de activación  $g$  es matemáticamente modelada como:

$$\sum_{i=1}^L \beta_i g(\mathbf{w}_i, b_i, \mathbf{x}_d) = \mathbf{y}_d \quad (3.102)$$

o, matricialmente:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} \quad (3.103)$$

siendo:

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1, b_1, \mathbf{x}_1) & \cdots & g(\mathbf{w}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1, b_1, \mathbf{x}_N) & \cdots & g(\mathbf{w}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L} \quad (3.104)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1m} \\ \vdots & \ddots & \vdots \\ \beta_{L1} & \cdots & \beta_{Lm} \end{bmatrix}_{L \times m} \quad \mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{N1} & \cdots & y_{Nm} \end{bmatrix}_{N \times m} \quad (3.105)$$

En donde:

- $\mathbf{H}$ : matriz de salida de la capa oculta;
- $\mathbf{w}_i \in \mathbb{R}^n$ : vector de pesos para los valores de entrada que llegan a la neurona  $i = 1, \dots, L$  de la capa oculta;
- $b_i$ : bias de la neurona  $i$  de la capa oculta;
- $\mathbf{x}_d$ : vector de valores de entrada de la muestra  $d = 1, \dots, N$ ;
- $\beta_{ij}$ : peso del nudo  $i$  de la capa oculta al nudo  $j = 1, \dots, m$ , capa de salida;
- $y_{dj}$ : valor de salida de la muestra  $d$  en el nudo  $j$ , capa de salida.

De acuerdo a (3.103), los parámetros  $\beta_{ij}$  se obtienen de:

$$\boldsymbol{\beta} = \mathbf{H}^+ \mathbf{Y} \quad (3.106)$$

En donde  $\mathbf{H}^+$  es la pseudoinversa de Moore-Penrose, que puede calcularse de la siguiente manera:

$$\mathbf{H}^+ = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \quad (3.107)$$

análogo a lo visto en (3.35).

### 3.3.4.2. Algoritmo de aprendizaje automático extremo

Modelo de tres pasos (Guang-Bin Huang, 2006):

Dado un conjunto de  $N$  datos de entrenamiento

$\mathbf{X} = \{(\mathbf{x}_d, \mathbf{y}_d) \mid \mathbf{x}_d \in \mathbb{R}^n, \mathbf{y}_d \in \mathbb{R}^m, d = 1, \dots, N\}$ , función de activación  $g$  y número de neuronas  $L$  en la capa oculta,

1. Asignar aleatoriamente los pesos  $\mathbf{w}_i \in \mathbb{R}^n$  y bias  $b_i$  con  $i = 1, \dots, L$ ;
2. Calcular la matriz de salida de la capa oculta  $\mathbf{H}$  según la ec. (3.104);
3. Calcular los pesos de salida  $\boldsymbol{\beta}$  de acuerdo a la ec. (3.106).

El programa correspondiente se encuentra en el anexo B.2.

# Capítulo 4

## BASE DE DATOS

A continuación se presentan los programas en computadora que se han desarrollado a fin de generar las bases de datos necesaria para el análisis. Como se ha explicado en el apartado 1.5, es necesario generar una gran cantidad de datos de manera a cubrir un amplio espectro de casos usuales en el mercado de la construcción.

Todos estos programas fueron realizados en `Python 3` (Van Rossum y Drake Jr, 1995) siguiendo las guías de estilo para códigos en `Python` (Rossum, Warsaw y Coghlan, 2013), con los siguientes paquetes científicos y entornos de programación para este lenguaje<sup>1</sup>:

- `numpy`, paquete fundamental para el cálculo numérico en `Python` (T. E. Oliphant, 2006);
- `sympy`, librería para matemática simbólica (Meurer y col., 2017);
- `pandas`, herramientas de estructuración y análisis de datos en `Python` (McKinney y col., 2010);
- `scipy`, librería para computación científica y técnica (Jones, T. Oliphant, Peterson y col., 2001).
- `scikit-learn`, algoritmos de aprendizaje automático (Pedregosa y col., 2011);
- `matplotlib`, librería para gráficos en 2D (Hunter, 2007);

---

<sup>1</sup>Python, las librerías y los entornos citados son libres y de código abierto.

- **seaborn**, librería para visualización de datos estadísticos (Waskom y col., 2017);
- **IPython**, entorno interactivo de Python (Pérez y Granger, 2007);
- **Jupyter Notebook**, aplicación web que permite crear y compartir códigos, ecuaciones, textos y otros elementos web (Kluyver y col., 2016);
- **Spyder**, entorno de desarrollo científico para python (Raybaut, 2009);

Todos los datos de entrada y salida se encuentran en unidades coherentes del Sistema Internacional de Unidades (BIPM, 1960), sin múltiplos ni submúltiplos para la generación de la base de datos y, luego, con los múltiplos y submúltiplos adecuados para el análisis de aprendizaje automático.

## 4.1. Programas de computadora

El análisis de las secciones transversales se realiza con la determinación de la capacidad resistente de la sección, conforme a sus dimensiones geométricas, el armado dispuesto en la viga y las características de los materiales. Se realiza con los siguientes módulos:

- **materiales.py**: con el cual se definen las características de interés de los materiales a ser utilizados, es decir, el hormigón y el acero, con sus respectivos diagramas de tensión-deformación, en función a sus resistencias características y demás elementos que los define inequívocamente (apéndice A.1);
- **vigas.py**: en conjunción con el módulo anterior, aquí se programan todos los cálculos necesarios, para la obtención de diagramas de interacción, capacidad última de la sección, diagramas de momento-curvatura, cálculo de flechas y otros (apéndice A.2);

## 4.2. Comprobación de secciones

### 4.2.1. Caracterización de los materiales

Las definiciones de los materiales se realiza en el módulo `materiales.py` (ver apéndice A.1).

#### 4.2.1.1. Material: Hormigón

Como ejemplo, es posible “crear” un material hormigón, denominado H con resistencia característica  $f_{ck} = 20$  MPa para análisis en Estado Límite Último<sup>2</sup> y sin capacidad a tracción:

---

```
import materiales as mt
H = mt.Hnl(fk=20e6, est_lim="ELU", traccion=False)
```

---

Con lo cual es posible obtener cualquier información de interés respecto a H, por decir, la tensión del hormigón para una deformación unitaria de 0,0025:

---

```
print(H.sigma_cc(-0.0025))
-13108176.56945103
```

---

Es decir: 13,1 MPa a compresión<sup>3</sup>

Su módulo de elasticidad de cálculo es:

---

```
print(H.Ecd)
29961951054.64031
```

---

Es decir: 29,96 GPa.

También se puede generar el gráfico tensión-deformación correspondiente a H con el siguiente código:

---

```
import matplotlib.pyplot as plt
X = H.abcisas_cc() #Abcisas: Deformaciones unitarias
Y = H.ordenadas_cc(X) #Ordenadas: Tensiones
```

---

<sup>2</sup>Con lo cual se consideran los coeficientes de seguridad adecuados.

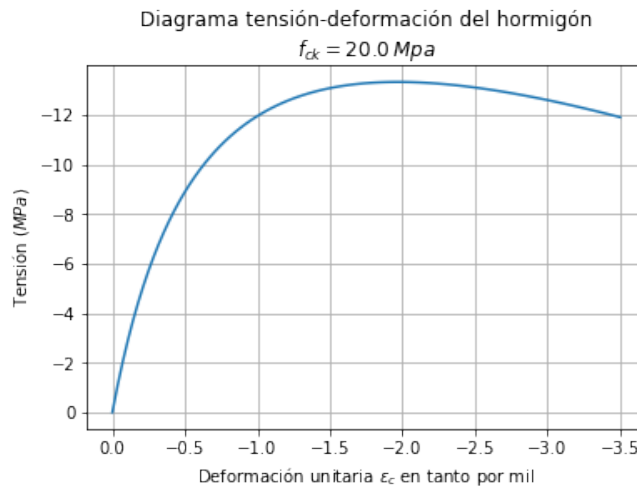
<sup>3</sup>Se recuerda que las entradas y las salidas están en el SI sin múltiplos ni submúltiplos.

```

plt.plot(X*1e3, Y*1e-6) #Curva de tensión-deformación
plt.gca().invert_xaxis() #Invierte el eje x
plt.gca().invert_yaxis() #Invierte el eje y
plt.title("Diagrama tensión-deformación del hormigón\n\n
$f_{{ck}} = {} \, ,Mpa$".format(H.fk*1e-6)) #Título
plt.xlabel(r"Deformación unitaria $\varepsilon_c$ en \
tanto por mil") #Título del eje x
plt.ylabel(r"Tensión $(MPa)$") #Título del eje y
plt.grid() #Dibuja una grilla
plt.show()

```

El cual devuelve la figura siguiente:



**Figura 4.1** – Gráfico tensión-deformación de cálculo del hormigón generado con el módulo materiales.py. Fuente elaboración propia.

#### 4.2.1.2. Material: Acero

De igual manera se puede crear un material acero denominado, por ejemplo, A, con resistencia característica  $f_{yk} = 500$  MPa, e imprimir, por decir, su densidad y resistencia de cálculo  $f_{yd}$ :

```

A = mt.Acero(fk=500e6)
print("Densidad= {} kg/m3; fcd= {:.2f} MPa".format(A.densidad,
A.fyd*1e-6))
Densidad= 7850 kg/m3; fyd= 434.78 MPa

```

También es posible obtener el gráfico de tensión-deformación del acero y otros valores importantes.

### 4.2.2. Momento último

El diagrama de interacción, el momento último de la sección y varios otros cálculos se realiza con el módulo `vigas.py` (ver apéndice A.2).

#### 4.2.2.1. Diagrama de interacción

Se ve el caso de obtener el diagrama de interacción de una viga de sección rectangular de  $20\text{cm} \times 50\text{cm}$  para los materiales H y A definidos más arriba, y con una armadura inferior de  $2\phi 16 \rightarrow A_{st} = 4,02\text{cm}^2$  y superior de  $2\phi 10 \rightarrow A_{st} = 1,57\text{cm}^2$  (por ahora no es importante la longitud de la viga).

---

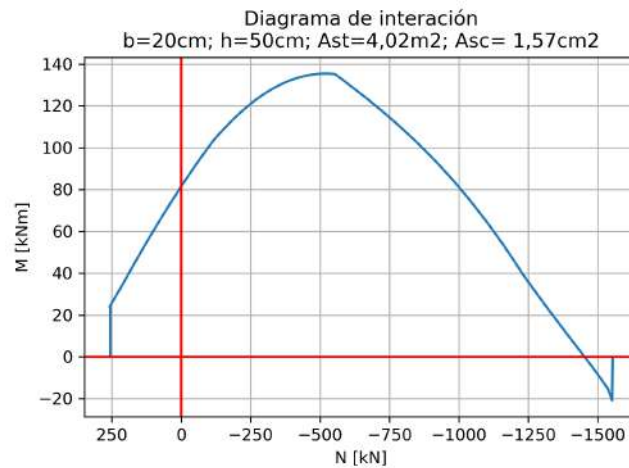
```
#Importación del módulo vigas.py
import vigas as vg

#Creación de una viga denominada V
V = vg.Viga(b=.2, h=.5, hormigon=H, acero=A, Ast=4.02e-4,
Asc=1.57e-4)

#Abcisas y ordenadas del diag. de interacción
N, M = V.diag_interaccion()

#Generación del diagrama
plt.plot(N*1e-3, M*1e-3) #Genera la curva con ejes en kN y kNm
plt.gca().invert_xaxis() #Invierte el eje x
plt.title("Diagrama de interacción\n\
b=20cm; h=50cm; Ast=4,02m2; Asc= 1,57cm2") #Título del gráfico
plt.xlabel("N [kN]") #Rótulo del eje x
plt.ylabel("M [kNm]") #Rótulo del eje y
plt.axhline(color='red') #Eje x
plt.axvline(color='red') #Eje y
plt.grid() #Dibuja una grilla
```

---



**Figura 4.2** – Diagrama de interacción de una sección de hormigón armado obtenido con el módulo `vigas.py`. Fuente: elaboración propia.

#### 4.2.2.2. Momento último

El momento último de la sección se obtiene fácilmente haciendo:

---

```
print(V.Mu())
81494.71050479135
```

---

Es decir:  $M_u = 81,49$  kNm que corresponde al punto del diagrama de interacción en donde la curva corta al eje de ordenadas (fuerza normal igual a cero).

### 4.3. Deformada de la viga

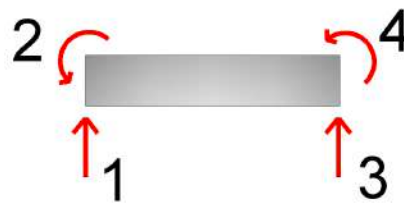
Se desarrolla un algoritmo para la obtención del diagrama momento - curvatura que se basa, primordialmente, en la hipótesis de secciones planas antes y después de la deformación. Para ser completamente coherentes con la formulación presentada en el capítulo 2, se adopta la hipótesis de Euler-Bernoulli, las secciones planas y perpendiculares al eje de la viga, mantienen esa condición luego de la deformación. En este algoritmo, básicamente, se supone el giro de las secciones buscando la profundidad de la línea neutra en la cual se anula la resultante de las fuerzas axiales, de manera a llegar a la situación de flexión simple que luego es relacionada con la curvatura del eje de la viga en la sección considerada. Para modelar el hormigón y el acero, se adopta el eurocódigo 2 (UNE-EN:1992, 2004).

Una vez obtenido el diagrama de momento - curvatura se lo aplica al cálculo de desplazamientos, en el cual se utiliza un modelo de elemento finito de 4 grados de libertad, como verificación, se revisan los cálculos con el método de la carga unitaria, basado en la teorema de los trabajos virtuales. En el proceso de cálculo se parte del diagrama de momento flector de la viga considerada, luego este es relacionado con la curvatura, según el diagrama de momento - curvatura, con lo cual es obtenido el factor  $EI$  que es utilizado siguiendo el proceso normal de cálculo de elementos finitos lineales y/o trabajo virtual lineal, pero nótese que en cada situación varía el valor de  $EI$ , por lo que, de esta forma, se está considerando la no linealidad del material. El análisis por tanto cae en la categoría de *Análisis no lineal de primer orden* definido en el apartado 2.1.2.

Para corroborar la exactitud de los resultados, se contrasta con dos estudios anteriores, el primero, realizado con un software de elementos finitos (Santos y col., 2014) y el segundo con resultados de ensayos experimentales (Kaczmarek y Szymańska, 2016)<sup>4</sup>.

#### 4.3.1. Diagrama de momento-curvatura

Se selecciona el conocido elemento de viga con 4 grados de libertad del método de elementos finitos:



**Figura 4.3** – Elemento de viga con cuatro grados de libertad. *Fuente:* elaboración propia.

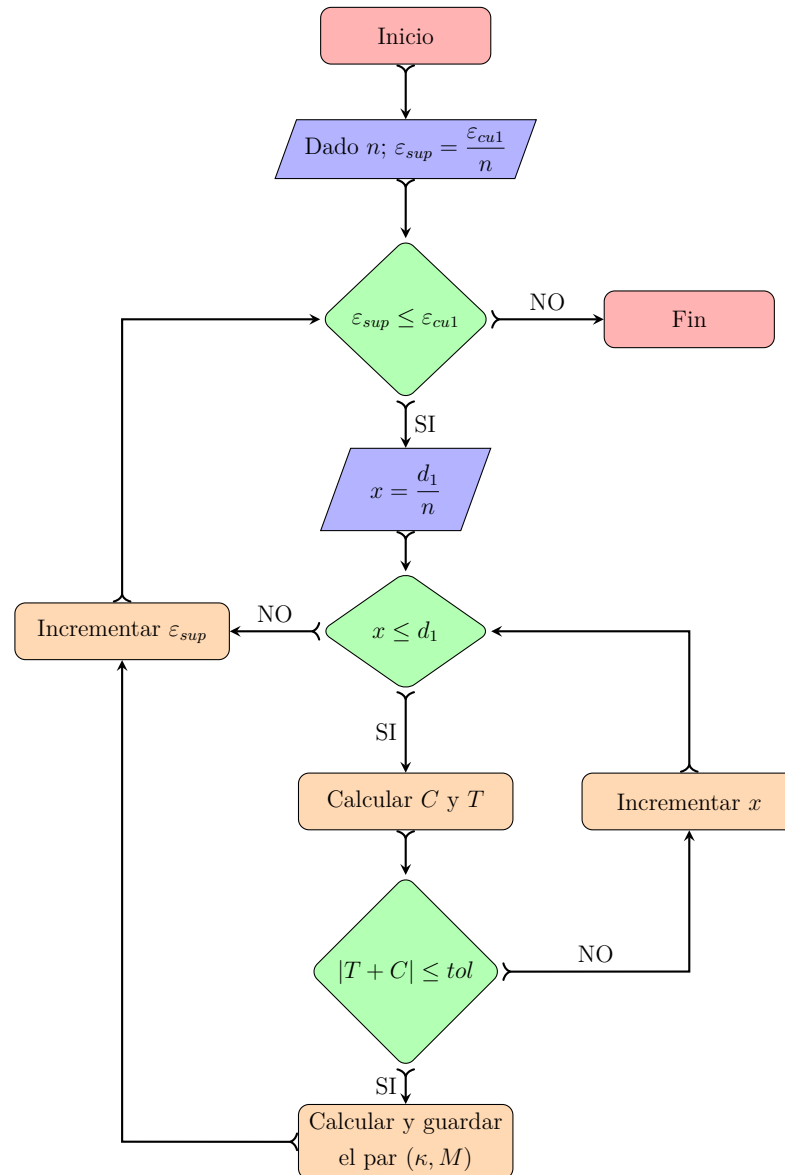
Se adopta un modelo de elemento finito aplicable a materiales homogéneos e isotrópicos, sin embargo, la naturaleza compuesta del hormigón armado es abordada de la siguiente manera:

La matriz de rigidez  $\mathbf{k}_e$  del elemento puede ser escrita de la forma

<sup>4</sup>Ver apartado 4.4.

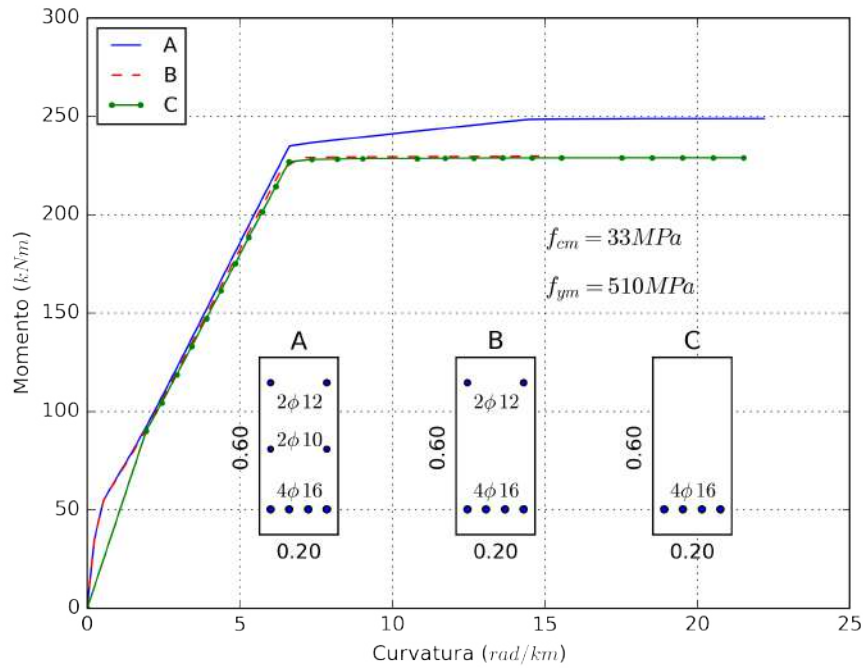


decir se verifica que la suma de las fuerzas del acero  $T$  y del hormigón  $C$  sea cero o menor a una tolerancia prefijada, llegados a este punto se calcula la curvatura y el momento flector. Las ecuaciones utilizadas están indicadas en (2.14).



**Figura 4.4** – Algoritmo utilizado para calcular el diagrama de momento-curvatura. Programado en el método `dmc()` del módulo `vigas.py`, Apéndice A.2. Fuente: elaboración propia.

Aplicando el algoritmo de la figura 4.4 es posible hacer varios análisis, por ejemplo, determinar la influencia de las armaduras de piel y compresión en el diagrama Momento-Curvatura en una sección de hormigón armado. Se toma una de  $0,20m \times 0,60m$ , se adopta un armado dado y se obtiene la figura 4.5:



**Figura 4.5** – Diagrama momento-curvatura de una sección de hormigón con diferentes armados. Se ve la influencia de las armaduras de piel y compresión. Fueron utilizados valores medios para las resistencias del hormigón y el acero. *Fuente:* elaboración propia.

### 4.3.3. Cálculo de flechas

Al calcular las deformaciones por el método de **elementos finitos**, utilizando como funciones de interpolación los polinomios de Hermite, se encuentra la matriz de rigidez  $\mathbf{k}_e$  del elemento (Kassimali, 2012):

$$\mathbf{k}_e = \frac{EI}{l_e^3} \begin{bmatrix} 12 & 6l_e & -12 & 6l_e \\ 6l_e & 4l_e^2 & -6l_e & 2l_e^2 \\ -12 & -6l_e & 12 & -6l_e \\ 6l_e & 2l_e^2 & -6l_e & 4l_e^2 \end{bmatrix} \quad (4.3)$$

en el cual se es posible insertar el factor  $EI$  tomado del diagrama que se está abordando.

Un método bastante adecuado para ser tratado con el diagrama momento - curvatura es el de la **carga unitaria**, en donde, la deflexión se calcula con

(Hibbeler, 2012):

$$v = \int_L \frac{Mm}{EI} dx \quad (4.4)$$

En donde<sup>6</sup>:

- $v$  deflexión vertical;
- $L$  longitud de la viga;
- $m$  momento virtual debido a una carga unitaria en la posición y dirección de la deflexión que se desea calcular;

Incorporando (4.2) en (4.4), se tiene:

$$v = \int_L \kappa m dx \quad (4.5)$$

Es decir, se puede *calcular la deflexión vertical de un punto dado del eje de la viga, integrando en toda su longitud el producto de la curvatura por el momento flector debido a una carga unitaria vertical en el punto considerado.*

#### Procedimiento:

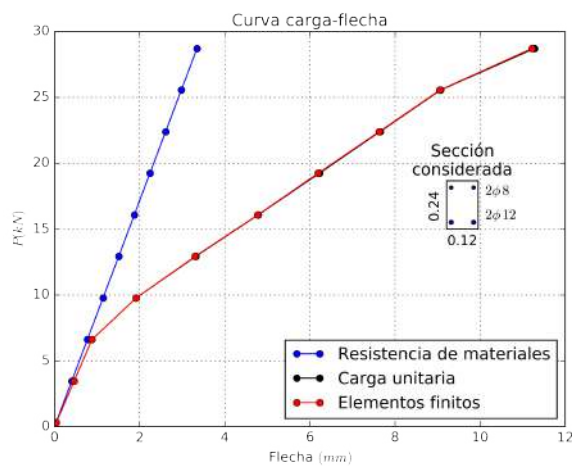
1. Obtener el diagrama de momento flector en vigas estáticamente determinadas;
2. Realizar el diagrama de curvatura de la viga, utilizando la relación momento - curvatura;
3. En el caso del método de elemento finito:
  - a) Promediar el momento flector de sus extremos ( $\bar{M}$ ), el cual es utilizado para la determinación de la curvatura promedio de este elemento  $\bar{\kappa}$ , así el cociente de estos valores determina el factor  $EI = \frac{\bar{M}}{\bar{\kappa}}$  que será utilizado en la matriz de rigidez del elemento considerado.
  - b) Ensamblar la matriz de rigidez global de la estructura y seguir el procedimiento usual para el cálculo de deformaciones, con lo cual se obtiene la deformada de la viga y en particular la flecha.

---

<sup>6</sup>A excepción de  $L$ , todas las variables de la ec. (4.4) dependen de  $x$ , incluso el producto  $EI$ .

4. En el caso del método de la carga unitaria: realizar la integral indicada en (4.4).

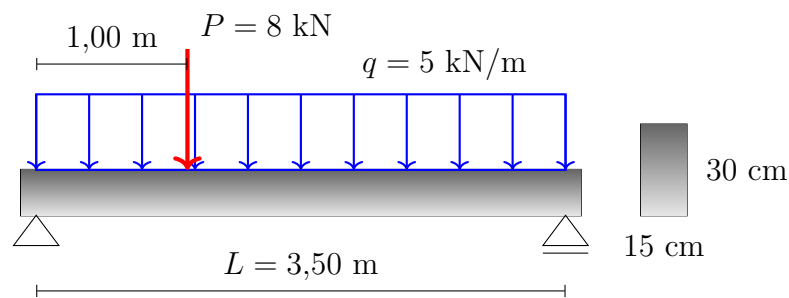
Como es de esperarse, estos dos métodos devuelven exactamente el mismo resultado. Se presenta la figura 4.6 que corresponde a la viga de la figura 4.9. a manera de comparación y se incluyen los resultados obtenidos con la teoría elemental de la Resistencia de Materiales.



**Figura 4.6** – Comparación de las flechas predichas por la teoría elemental de la mecánica de materiales y el modelo propuesto. El método de carga unitaria y elementos finitos tienen curvas superpuestas. *Fuente:* elaboración propia.

#### 4.3.4. Deformada con diagrama de momento-curvatura

A continuación se da un ejemplo de obtención de la deformada de una viga, y por lo tanto de la flecha correspondiente. Se considera la viga simplemente apoyada de la figura 4.7 con resistencia característica del hormigón  $f_{ck} = 25$  MPa; resistencia característica del acero  $f_{yk} = 420$  MPa; armado de tracción  $2\phi 12$ ; armado de compresión  $2\phi 8$ , recubrimiento  $rec = 2$  cm.



**Figura 4.7** – Viga de ejemplo para cálculo de su deformada. *Fuente:* elaboración propia.

Se obtiene la deformada de la viga, utilizando el procedimiento señalado más arriba, con los módulos ya presentados, de la siguiente manera:

```
#Librerías
import materiales as mt
import vigas as vg
import numpy as np

#Estado límite
eLim = "ELS" #Estado límite de servicio

#Materiales
H = mt.Hnl(fk=25e6, est_lim=eLim) #hormigón
A = mt.Acero(fk=420e6, est_lim=eLim) #acero

#Armado de la viga
Ast = 2*(12e-3)**2*np.pi/4 #tracción
Asc = 2*(8e-3)**2*np.pi/4 #compresión

#Instancia de viga
V1 = vg.Viga(b=.15, h=.30, L=3.5, hormigon=H, acero=A,
Ast=Ast, Asc=Asc, rec=0.02, est_lim=eLim)

#Cargas
P = V1.carga_puntual(2e3, 1.50)
q = V1.carga_unif_dist(5e3)
cargas = [P, q]
```

---

```
#Diagramas
k1, M1 = V1.dmc() #momento-curvatura
X_dmf, Y_dmf = V1.DMF(cargas) #momento flector
Xk, Yk = V1.diag_curvatura(cargas) #diagrama de curvaturas
X_lin, Y_lin = V1.elastica_lin(cargas) #elástica lineal
X_dmc, Y_dmc = V1.elastica_dmc(cargas) #elástica no lineal
```

---

Los diagramas resultantes<sup>7</sup> se muestran en la figura 4.8.

Un código breve, que devuelve directamente la flecha utilizando el diagrama de momento-curvatura, con los datos recientemente dados, es:

---

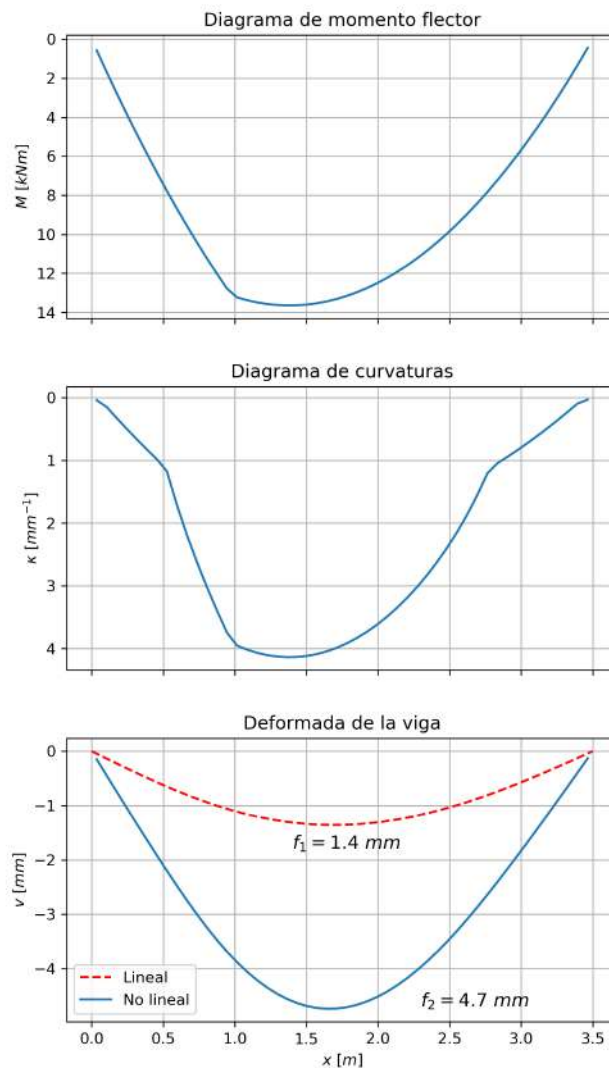
```
V1.flecha_dmc(cargas)
(0.004739661510335414, 1.645)
```

---

El cual significa una flecha de 4,7 mm en la posición  $x = 1,645$  m del apoyo izquierdo.

---

<sup>7</sup>El código específico de `matplotlib` (Hunter, 2007) para realizar los diagramas lo se ha omitido por brevedad.



**Figura 4.8** – Diagramas de momento flector, de curvaturas y deformada por dos métodos, correspondientes a la viga de la figura 4.7. Notar la diferencia de las flechas obtenidas por la teoría lineal y el obtenido por integración de las curvaturas. *Fuente:* elaboración propia.

## 4.4. Contrastación de resultados

De manera a verificar la validez de los resultados obtenidos con los módulos presentados hasta aquí<sup>8</sup>, se realizan comparaciones con resultados obtenidos en otros estudios.

<sup>8</sup>Los módulos referidos son: `materiales.py`, `vigas.py` y `utiles.py`.

#### 4.4.1. Primera comparación

Se presenta como primer ejemplo las vigas VRA, VRB y VRC<sup>9</sup> modeladas por Santos y col. (2014) con el paquete computacional de elementos finitos DIANA, utilizando una viga bi-apoyada con cargas puntuales en los tercios del vano, como se indica en la figura 4.9.

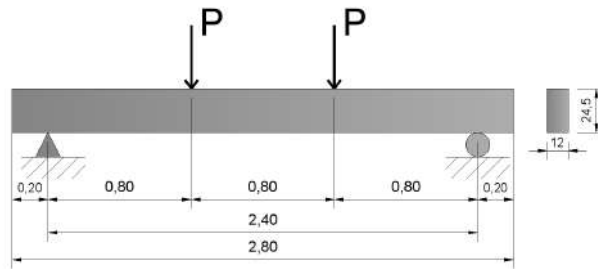


Figura 4.9 – Viga tipo, modelada por Santos y col. (2014)

Para mayores detalles se debe recurrir a la referencia citada. La comparación de resultados se refleja en la figura 4.10.

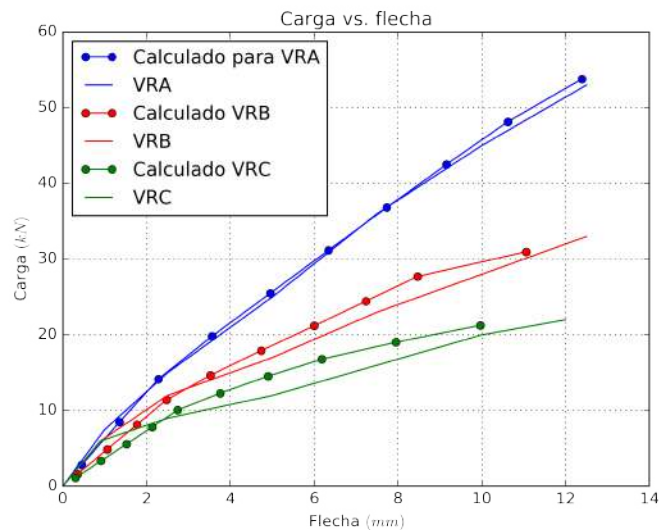
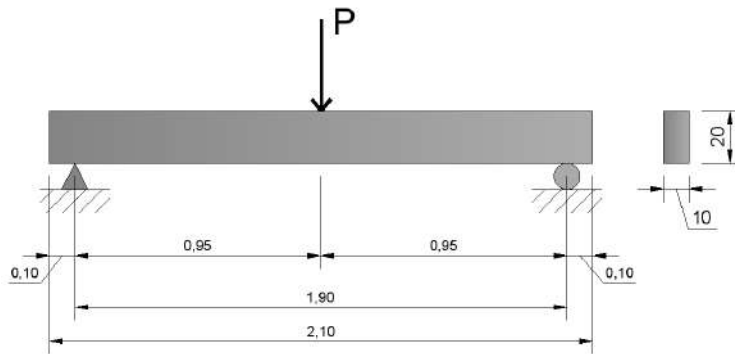


Figura 4.10 – Comparación de resultados entre el modelo propuesto y los presentados por Santos y col. (2014). Las líneas punteadas corresponden al modelo propuesto y las otras a las curvas de comparación.

<sup>9</sup>Estas denominaciones son las dadas en el estudio de referencia.

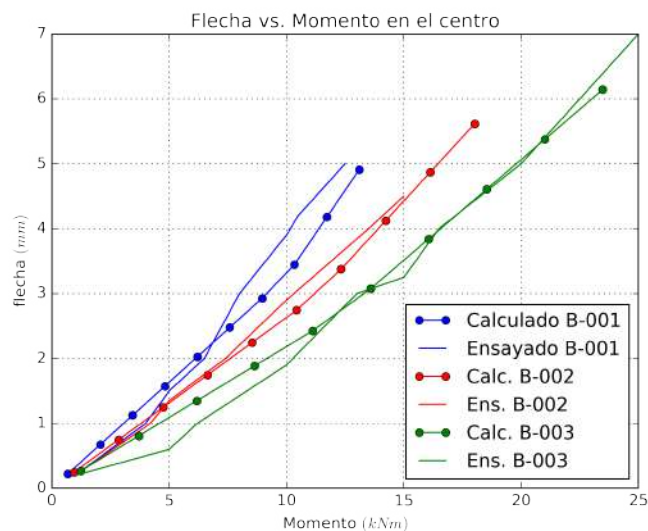
#### 4.4.2. Segunda comparación

Ahora se consideran las vigas ensayadas experimentalmente por Kaczmarek y Szymańska (2016), las cuales tienen la configuración mostrada en la figura 4.11.



**Figura 4.11** – Viga tipo, ensayada experimentalmente por Kaczmarek y Szymańska (2016)

Igual que el caso anterior, se obvian los detalles referentes a los datos precisos de las vigas, ya que pueden corroborarse en la referencia citada. La comparación de resultados para este caso, se muestra en la figura 4.12. Las denominaciones dadas por estos investigadores a las vigas ensayadas son: B-001, B-002 y B-003.



**Figura 4.12** – Comparación de resultados entre el modelo propuesto y los presentados por Kaczmarek y Szymańska (2016). Las líneas punteadas corresponden al modelo propuesto y las otras a las curvas de comparación.

Se considera que estas aproximaciones son suficientes, por lo tanto, de esta

manera se asume como válidos los códigos relacionados con el hormigón armado y se pasa a generar una base de datos para el análisis con las técnicas de aprendizaje automático.

## 4.5. Generación de base de datos

En vista del objetivo de realizar análisis de vigas rectangulares de hormigón armado con aprendizaje automático, se precisa de una base de datos de entrenamiento y otra de verificación o comprobación. Se divide esto en dos partes, por un lado la comprobación de secciones (en ELU), que se puede realizar calculando la capacidad última resistente de una serie de secciones transversales, en nuestro caso, a flexión simple y, por otro lado, la verificación de la flecha (ELS).

Por lo tanto se pasa a calcular la capacidad resistente de una serie de secciones con diferentes geometrías, resistencias características y armados y, posteriormente, se calcula la flecha, para diferentes secciones transversales, longitudes y situaciones de carga.

### 4.5.1. Base de datos para análisis de secciones

La comprobación de las secciones requiere de las siguientes variables:

$X_1 = f_{yk}$ : Resistencia característica del acero;

$X_2 = f_{ck}$ : Resistencia característica del hormigón;

$X_3 = b$ : base de la sección transversal;

$X_4 = h$ : altura de la sección transversal;

$X_5 = A_{st}$ : área del armado a tracción;

$X_6 = A_{sc}$ : área del armado a compresión;

$X_7 = M_u$ : momento último resistente de la sección

Definidas las primeras 6 variables es posible determinar la séptima<sup>10</sup>.

---

<sup>10</sup>Se verá luego que, con el aprendizaje automático, es posible determinar cualquier variable dadas las otras 6. Incluso en algunos casos es posible determinar 2 variables dadas las otras 5.

Para generar la base de datos se adoptan los siguientes criterios:

- **Recubrimiento**, en general, para todos los casos:

$$rec = 2,5 \text{ cm}$$

- **Resistencias características del acero:**  $f_{yk} = \{400, 500\}$  MPa;
- **Resistencias características del hormigón<sup>11</sup>:**

$$f_{ck} = \{12, 16, 20, 25, 30, 35\} \text{ MPa}$$

- **Bases:**  $b = \{5, 10, 15, \dots, 40\}$  cm;
- **Alturas:**  $h$ : desde  $h_{\text{mín}} = \text{máx}(15\text{cm}, b)$  hasta  $h_{\text{máx}} = \text{mín}(1,00 \text{ m}, 4b)$  con saltos de 5 cm;
- **Áreas de armaduras de tracción:**  $A_{st}$

- Armadura mínima  $A_{st,\text{mín}}$

- Criterio de armado mínimo:

$$A_{\text{mín}}^{(1)} = \begin{cases} 2\phi 8 & \text{si } b < 20 \text{ cm} \\ 2\phi 10 & \text{si } 20 \text{ cm} \leq b < 25 \text{ cm} \\ 3\phi 10 & \text{en los demás casos} \end{cases}$$

- Armadura mínima según Eurocódigo 2, 9.1.1.1 (9.1N)

$$A_{\text{mín}}^{(2)} = \text{máx} \left( 0,26 \frac{f_{ctm}}{f_{yk}} bd; 0,0013bd \right)$$

$$\text{Armadura mínima adoptada: } A_{st,\text{mín}} = \text{máx} \left( A_{\text{mín}}^{(1)}; A_{\text{mín}}^{(2)} \right)$$

- Armadura de tracción máxima:  $A_{st,\text{máx}} = 0,04bh$
- Paso o saltos de valor entre áreas de armado:  $A_{\text{mín}}^{(1)}$

- **Áreas de armaduras de compresión:**  $A_{sc}$

$$A_{sc} = \text{máx} \left( A_{\text{mín}}^{(1)}; A_{\text{mín}}^{(2)}; 0,30A_{st} \right)$$

---

<sup>11</sup>Son las clases de hormigón dadas en el Eurocódigo 2 Tabla 3.1, no se incluyen los hormigones de alta resistencia.

El cálculo se realiza en Estado Límite Último conforme se estuvo presentando<sup>12</sup>, en base al cual, y con los criterios mostrados precedentemente, se ejecuta el código del apéndice B.1.1, con el que se obtiene una base de datos de 33.037 registros, 5 de los cuales se muestran a continuación, en la tabla 4.1<sup>13</sup>:

	<b>fyk</b>	<b>fck</b>	<b>b</b>	<b>h</b>	<b>Ast</b>	<b>Asc</b>	<b>Mu</b>
23818	500.0	20.0	35.0	65.0	75.885779	22.765734	1747.672062
24967	500.0	25.0	15.0	45.0	10.053096	3.015929	177.905187
19778	500.0	16.0	20.0	65.0	9.478982	2.843694	255.126550
29900	500.0	30.0	40.0	70.0	8.779030	4.066641	297.180633
14330	400.0	35.0	20.0	70.0	29.520280	8.856084	678.858934

**Tabla 4.1** – Muestra de 5 registros de un total de 33.037, correspondientes a secciones transversales de vigas.  $f_{yk}$  y  $f_{ck}$  están en MPa,  $b$  y  $h$  en cm,  $A_{st}$  y  $A_{sc}$  en  $\text{cm}^2$  y  $M_u$  en kNm.

En la base de datos de la tabla 4.1 se puede ver, por ejemplo (en el 3er. registro), que una viga de  $20 \times 65 \text{ cm}^2$ , con  $f_{ck} = 16 \text{ MPa}$ ,  $f_{yk} = 500 \text{ MPa}$ , armadura de tracción  $A_{st} = 9,48 \text{ cm}^2$ , armadura de compresión  $A_{sc} = 2,84 \text{ cm}^2$ , tiene un momento resistente en ELU  $M_u = 255 \text{ kNm}$ , conforme a las directrices de la norma UNE-EN:1992 (2004).

#### 4.5.2. Base de datos para análisis de deformaciones

La deformación considerada es la flecha, la cual se refiere a la deflexión máxima vertical de la viga.

Como se había adelantado en la sección 1.1, la carga considerada es uniformemente distribuida en toda la longitud de la viga, la cual se supone simplemente apoyada.

Las variables son las siguientes:

$X_1 = f_{yk}$ : Resistencia característica del acero;

$X_2 = f_{ck}$ : Resistencia característica del hormigón;

<sup>12</sup>En los cálculos realizados, se tomaron en cuenta el aporte del hormigón a tracción, así como el aporte de la armadura a compresión.

<sup>13</sup>En formato Pandas Data Frame (McKinney y col., 2010).

$X_3 = b$ : base de la sección transversal;

$X_4 = h$ : altura de la sección transversal;

$X_5 = L$ : Longitud de la viga;

$X_6 = A_{st}$ : área del armado a tracción;

$X_7 = q$ : carga uniformemente distribuida en toda la longitud de la viga;

$X_8 = f$ : flecha;

Los valores adoptados para la generación de la base de datos son:

- **Recubrimiento**, en general, para todos los casos:

$$rec = 2,5 \text{ cm}$$

- **Resistencias características del acero**:  $f_{yk} = \{400, 500\}$  MPa;
- **Resistencias características del hormigón**:  $f_{ck} = \{16, 20, 25\}$  MPa;
- **Bases**:  $b = \{20, 25, 30\}$  cm;
- **Longitudes**:  $L = \{3, 4, 5, 6, 7, 8\}$  m;
- **Alturas**:  $h = \left\{\frac{L}{10}, \frac{L}{12}, \frac{L}{14}\right\}$  m, redondeados por exceso en múltiplos de 5 cm;
- **Áreas armaduras de tracción**:  $A_{st}$  se obtuvieron tomando 4 valores intermedios, considerando el momento mínimo y máximo que podría soportar la sección, con armado mínimo y máximo respectivamente.
- **Áreas armaduras de compresión**:  $0,25A_{st}$ ;
- **Cargas externas**:  $q = \frac{8M}{L^2} - pp$ , es decir la obtenida considerando el momento máximo del diagrama de momento flector para una viga simplemente apoyada con carga uniformemente distribuida en toda su longitud, a la que se le descuenta el peso propio de la viga<sup>14</sup>.
- **Limitación de flecha**:  $f \leq \frac{L}{250}$  (UNE-EN:1992, 2004, 7.4.1(4)).

<sup>14</sup>La carga total incluye el peso propio de la viga, que es agregado luego en el proceso de cálculo.

El cálculo se realiza en Estado Límite de Servicio, en el cual todos los coeficientes de seguridad son iguales a la unidad. Conforme a estos criterios se obtuvo una base de datos con 2.628 registros, 5 de los cuales se muestran en la tabla 4.2.

La base de datos fue generada con el código presentado en el Apéndice B.1.2.

	<b>fyk</b>	<b>fck</b>	<b>b</b>	<b>h</b>	<b>L</b>	<b>Ast</b>	<b>q</b>	<b>f</b>
1299	500.0	20.0	30.0	40.0	5.0	9.972399	32.237941	15.010447
2296	400.0	25.0	20.0	80.0	8.0	25.091958	52.769663	16.676050
2511	400.0	20.0	25.0	70.0	8.0	53.302891	90.193343	23.398768
2248	500.0	25.0	30.0	50.0	7.0	12.310155	25.382408	22.822195
2451	400.0	20.0	20.0	70.0	8.0	11.399918	30.607555	26.116156

**Tabla 4.2** – Muestra de 5 registros de un total de 2.628, correspondiente a cálculos de flechas en vigas simplemente apoyadas con cargas uniformemente distribuidas en toda su longitud.  $f_{yk}$  y  $f_{ck}$  en MPa,  $b$  y  $h$  en cm,  $L$  en m,  $A_{st}$  en  $\text{cm}^2$ ,  $q$  en kN y  $f$  en mm.

Los análisis de estas bases de datos, a saber, capacidad de secciones (tabla 4.1) y deformaciones (tabla 4.2) se realizan en el capítulo 5 y 6 respectivamente.

# Capítulo 5

## ANÁLISIS DE SECCIONES

En este capítulo se realiza el análisis de la base de datos de secciones transversales de vigas rectangulares de hormigón armado generada conforme se indicó en el apartado 4.5.1.

Primeramente se realiza el análisis de las componentes principales PCA y luego se aborda el problema de predecir el momento último resistente de una sección a partir de sus parámetros constitutivos, utilizando cuatro métodos de aprendizaje automático:

1. Regresión Lineal (LR);
2. Vector Soporte Automático (SVM);
3. Redes Neuronales Artificiales (ANN); y
4. Aprendizaje Automático Extremo (ELM).

### 5.1. Visión general de la base de datos

Conforme se presentó en la tabla 4.1, en una muestra de la base de datos, las variables de entrada son:

$$X_1 = f_{yk}, X_2 = f_{ck}, X_3 = b, X_4 = h, X_5 = A_{st}, X_6 = A_{sc} \quad (5.1)$$

y la de salida:

$$Y = M_u \quad (5.2)$$

La matriz datos de entrada tiene la forma:

$$\mathbf{X} = \begin{bmatrix} f_{yk}^{(1)} & f_{ck}^{(1)} & b^{(1)} & h^{(1)} & A_{st}^{(1)} & A_{sc}^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{yk}^{(N)} & f_{ck}^{(N)} & b^{(N)} & h^{(N)} & A_{st}^{(N)} & A_{sc}^{(N)} \end{bmatrix} \quad (5.3)$$

Los valores que puede tomar cada una de estas variables, se presentaron en 4.5.1. El vector de salida es:

$$\mathbf{y} = \begin{bmatrix} M_u^{(1)} \\ \vdots \\ M_u^{(N)} \end{bmatrix} \quad (5.4)$$

Estas variables fueron definidas en la página 84. En este caso  $N = 33.037$ .

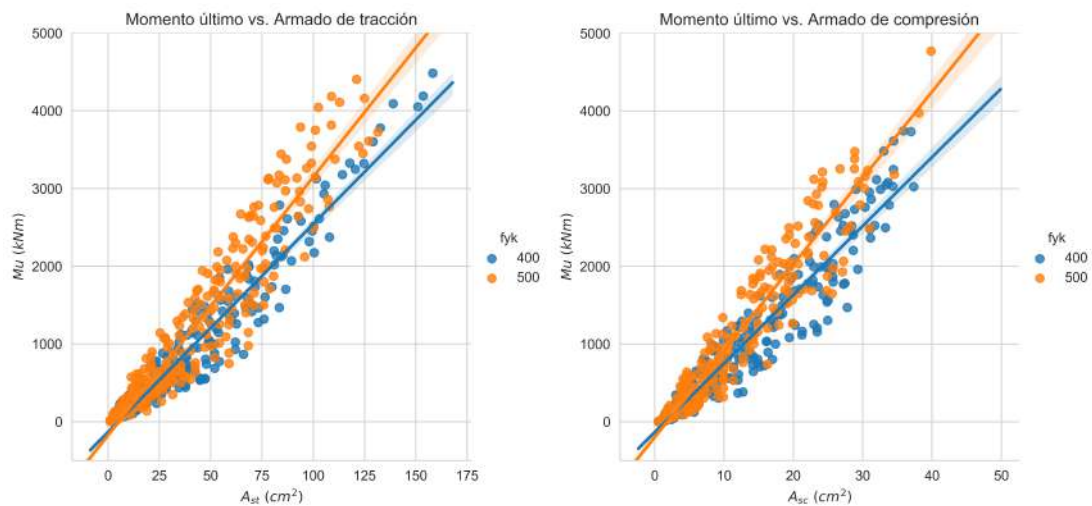
Los valores medios  $\bar{x}$ , las desviaciones estándar  $s$  y los coeficientes de variación  $s/\bar{x}$  son:

Variable	$\bar{x}$	$s$	Unidad	$s/\bar{x}$
$f_{yk}$	450	50,0	MPa	0,11
$f_{ck}$	23	7,9	MPa	0,34
$b$	29,1	8,8	cm	0,30
$h$	67,6	21,6	cm	0,32
$A_{st}$	42,0	32,5	cm <sup>2</sup>	0,78
$A_{sc}$	9,6	9,62	cm <sup>2</sup>	0,76
$M_u$	1101	1023	kNm	0,93

**Tabla 5.1** – Estadísticos de las variables para el análisis de secciones

## 5.2. Gráficas

Como ejemplo de la información contenida en la base de datos de secciones, se muestra la figura 5.1:



(a) Arm. de tracción vs. Momento último      (b) Arm. de compresión vs. Momento último

**Figura 5.1** – Variación del momento último resistente de una sección de hormigón armado en función a las áreas de armaduras de tracción (a) y compresión (b).  
Fuente: elaboración propia.

En la figura 5.2 se reflejan las relaciones de varias de las variables respecto a las demás.

## 5.3. Análisis de componentes principales

La secuencia de análisis es la siguiente:

### 5.3.1. Estandarización de variables

La estandarización de las variables de entrada se logra restando a cada valor la media correspondiente y dividiendo este resultado entre su desviación estándar (3.15), para obtener, a partir de  $\mathbf{X}$  la matriz con datos estandarizados  $\tilde{\mathbf{X}}$ , cada columna de  $\tilde{\mathbf{X}}$  tiene media igual a cero y desviación estándar igual a uno.

### 5.3.2. Matriz de correlaciones

Obtención de la **matriz de correlaciones**, conforme a la ecuación (3.14). Dado que los valores están estandarizados, la matriz de covarianzas es igual a la



Figura 5.2 – Relación gráfica entre algunas variables. Fuente: elaboración propia.

matriz de correlaciones (3.16):

$$\mathbf{R}_1 = \begin{bmatrix}
 f_{yk} & f_{ck} & b & h & A_{st} & A_{sc} \\
 \left[ \begin{array}{cccccc}
 1 & 0,001 & 0,001 & 0,001 & -0,003 & -0,004 \\
 0,001 & 1 & -0,001 & -0,001 & 0,006 & 0,008 \\
 0,001 & -0,001 & 1 & 0,433 & 0,499 & 0,510 \\
 0,001 & -0,001 & 0,006 & 1 & 0,529 & 0,539 \\
 -0,003 & 0,006 & 0,499 & 0,529 & 1 & 0,999 \\
 -0,004 & 0,008 & 0,510 & 0,539 & 0,999 & 1
 \end{array} \right] & \begin{array}{l}
 f_{yk} \\
 f_{ck} \\
 b \\
 h \\
 A_{st} \\
 A_{sc}
 \end{array}
 \end{bmatrix} \quad (5.5)$$

A partir de  $\mathbf{R}_1$ , podemos hacer las siguientes observaciones:

- $f_{yk}$  prácticamente no está correlacionada con ninguna de las otras variables;
- $f_{ck}$  tampoco tiene ninguna correlación con las demás variables;
- $b$  está medianamente correlacionada con la altura  $h$ , debido a la relación establecida en el apartado 4.5.1. También está medianamente correlacionada con las áreas de armadura  $A_{st}$  y  $A_{sc}$ , debido a los criterios de armado mínimo y máximo que dependen de  $b$ ;
- $h$  está medianamente correlacionada con  $b$ ,  $A_{st}$  y  $A_{sc}$ , de acuerdo a las mismas observaciones hechas para  $b$ ;
- $A_{st}$  y  $A_{sc}$  están muy, correlacionadas, hay una dependencia fuerte entre ambas variables, también debido a los criterios establecidos en 4.5.1. Conforme a esto, **en los análisis de regresión podemos obviar  $A_{sc}$ .**

La matriz  $\mathbf{R}$  adoptada es la siguiente:

$$\mathbf{R} = \begin{bmatrix} f_{yk} & f_{ck} & b & h & A_{st} \\ 1 & 0,001 & 0,001 & 0,001 & -0,003 \\ 0,001 & 1 & -0,001 & -0,001 & 0,006 \\ 0,001 & -0,001 & 1 & 0,433 & 0,499 \\ 0,001 & -0,001 & 0,006 & 1 & 0,529 \\ -0,003 & 0,006 & 0,499 & 0,529 & 1 \end{bmatrix} \begin{matrix} f_{yk} \\ f_{ck} \\ b \\ h \\ A_{st} \end{matrix} \quad (5.6)$$

En donde hemos eliminado la fila y la columna correspondientes a  $A_{sc}$  en (5.5).

### 5.3.3. Valores característicos

La matriz de valores característicos de  $\mathbf{R}$  es:

$$\mathbf{\Lambda} = \begin{bmatrix} 1,976 & 0 & 0 & 0 & 0 \\ 0 & 1,001 & 0 & 0 & 0 \\ 0 & 0 & 0,999 & 0 & 0 \\ 0 & 0 & 0 & 0,569 & 0 \\ 0 & 0 & 0 & 0 & 0,455 \end{bmatrix} \quad (5.7)$$

El **porcentaje explicado** por cada componente es:

Componente	Autovalor	% Explicado	% Acumulado
$Z_1$	1,976	39,51 %	39,51 %
$Z_2$	1,001	20,03 %	59,54 %
$Z_3$	0,999	19,97 %	79,51 %
$Z_4$	0,569	11,33 %	90,90 %
$Z_5$	0,455	9,10 %	100,00 %

**Tabla 5.2** – Porcentaje explicado por cada componente principal

Con las dos primeras componentes principales, el porcentaje de variabilidad explicado es 59,24 %.

### 5.3.4. Vectores característicos

Los **vectores característicos** están en las columnas de  $\mathbf{Q}$ , el cual es:

$$\mathbf{Q} = \begin{bmatrix} -0,0008 & 0,7016 & -0,7126 & 0,0009 & 0,0051 \\ 0,0018 & 0,7126 & 0,7015 & -0,0017 & -0,0101 \\ 0,5604 & -0,0020 & -0,0057 & -0,7676 & -0,311 \\ 0,5731 & -0,0020 & -0,0055 & 0,6305 & -0,5234 \\ 0,5979 & 0,0025 & 0,0076 & 0,1151 & 0,7933 \end{bmatrix} \quad (5.8)$$

### 5.3.5. Componentes principales

Atendiendo a la ecuación (3.21), las dos primeras **componentes principales** son<sup>1</sup>:

$$\begin{cases} Z_1 = -0,0008\tilde{f}_{yk} + 0,0018\tilde{f}_{ck} + \mathbf{0,5604}\tilde{b} + \mathbf{0,5731}\tilde{h} + \mathbf{0,5979}\tilde{A}_{st} \\ Z_2 = \mathbf{0,7016}\tilde{f}_{yk} + \mathbf{0,7126}\tilde{f}_{ck} - 0,0020\tilde{b} - 0,0020\tilde{h} + 0,0025\tilde{A}_{st} \end{cases} \quad (5.9)$$

Las tildes puestas en las variables están para recordar que se tratan de variables estandarizadas. Observar, en negritas, los términos preponderantes de cada componente principal.

<sup>1</sup>Según el criterio de Kaiser (Grané Chávez, 2010) es suficiente con las componentes principales cuyos autovalores sean mayores a 1.

### 5.3.6. Correlaciones entre componentes principales y variables originales

Las correlaciones entre las componentes principales y las variables originales son:

	$Z_1$	$Z_2$
$\tilde{f}_{yk}$	-0,0011	<b>0,7020</b>
$\tilde{f}_{ck}$	0,0025	<b>0,7131</b>
$\tilde{b}$	<b>0,7877</b>	-0,0020
$\tilde{h}$	<b>0,8056</b>	-0,0020
$\tilde{A}_{st}$	<b>0,8404</b>	0,0025

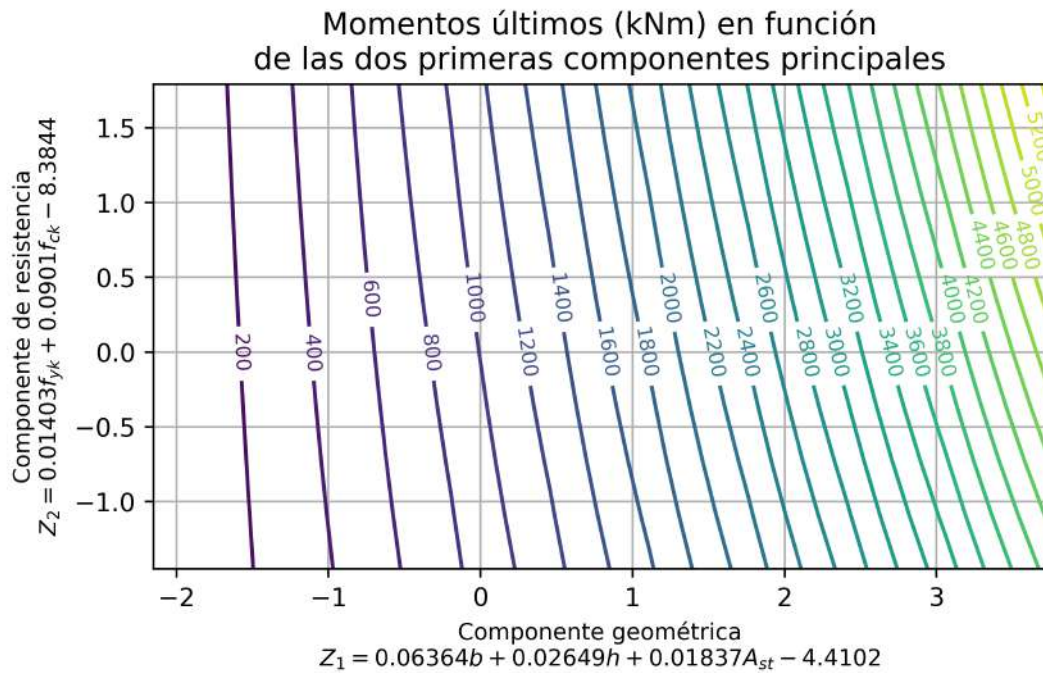
**Tabla 5.3** – Correlaciones entre las dos primeras componentes principales y las variables originales. Se observa la influencia de los datos geométricos en  $Z_1$  y de los valores de resistencia en  $Z_2$ .

### 5.3.7. Notas

- La componente principal que aporta mayor información hemos denominado como  $Z_1$ , tiene mayor porcentaje de variación explicada (tabla 5.2) y está muy correlacionada con las variables geométricas ( $b, h$ ) de la sección y el área ( $A_{st}$ ) de armaduras (tabla 5.3), por lo que podemos denominarla *componente geométrica*;
- $Z_2$  está muy correlacionada con la resistencia de los materiales (tabla 5.3), por lo que podemos denominarla *componente de resistencia*;
- Los datos geométricos tienen mayor preponderancia que los datos de resistencia de materiales;
- Observando (5.9) podemos despreciar el efecto de  $\tilde{f}_{yk}$  y  $\tilde{f}_{ck}$  en  $Z_1$  y de  $\tilde{b}$ ,  $\tilde{h}$  y  $\tilde{A}_{st}$  en  $Z_2$ , y luego “desnormalizar” las variables de acuerdo a (3.15) para obtener:

$$\begin{cases} Z_1 = 0,06364b + 0,02649h + 0,01837A_{st} - 4,4102 \\ Z_2 = 0,01403f_{yk} + 0,0901f_{ck} - 8,3844 \end{cases} \quad (5.10)$$

Con los cuales se han calculado los momentos últimos ilustrados en la figura 5.3



**Figura 5.3** – Momentos últimos (capacidad resistente en kNm) de una sección rectangular de hormigón armado en función de las componentes principales  $Z_1$  y  $Z_2$ ;  $f_{yk}$  y  $f_{ck}$  en MPa;  $b$  y  $h$  en cm;  $A_{st}$  en  $\text{cm}^2$ .

- Disminución del momento último al disminuir un 10 % cada variable:

Como información adicional, estudiamos la variación de la capacidad resistente de la sección al disminuir un porcentaje dado cada variable. Estos valores fueron calculados en relación a la capacidad resistente de una sección adoptando los valores medios  $\bar{x}$  de la base de datos (tabla 5.1).

	$f_{yk}$	$f_{ck}$	$b$	$h$	$A_{st}$	$A_{sc}$
Disminución del $M_u$	8,8 %	0,71 %	1,25 %	11,7 %	8,3 %	1,9 %

**Tabla 5.4** – Porcentaje de disminución del momento último cuando cada variable (independientemente) disminuye un 10%.

## 5.4. Decisiones sobre la base de datos

Para los análisis posteriores, se hacen las siguientes consideraciones:

Como los valores posibles de las resistencias características del acero son dos, se recuerda, 400 MPa y 500 MPa, se divide el conjunto de datos en dos partes,

precisamente en función de este par de valores.

	<b>fyk</b>	<b>fck</b>	<b>b</b>	<b>h</b>	<b>Ast</b>	<b>Asc</b>	<b>Mu</b>
11619	400	30	20	75	13.725496	4.117649	358.499672
670	400	12	20	85	8.428185	2.528456	248.346426
4326	400	16	30	100	17.939667	5.381900	627.462671
12026	400	30	25	90	6.474610	4.118416	241.634478
183	400	12	15	45	2.010619	1.005310	32.021795

**Tabla 5.5** – Muestra, base de datos de secciones con  $f_{yk} = 400$  MPa. Número de registros: 16.490.  $f_{ck}$  en MPa,  $b$  y  $h$  en cm,  $A_{st}$  y  $A_{sc}$  en  $\text{cm}^2$  y  $M_u$  en kNm.

	<b>fyk</b>	<b>fck</b>	<b>b</b>	<b>h</b>	<b>Ast</b>	<b>Asc</b>	<b>Mu</b>
23811	500	20	35	65	59.392418	17.817725	1426.643529
24730	500	20	40	100	52.193890	15.658167	2142.947066
21062	500	16	35	65	90.022946	27.006884	1876.282710
23774	500	20	35	60	54.452529	16.335759	1202.648173
27144	500	25	40	70	8.313598	3.601209	277.106242

**Tabla 5.6** – Muestra, base de datos de secciones con  $f_{yk} = 500$  MPa. Número de registros: 16.547.  $f_{ck}$  en MPa,  $b$  y  $h$  en cm,  $A_{st}$  y  $A_{sc}$  en  $\text{cm}^2$  y  $M_u$  en kNm.

### 5.4.1. Selección de datos de entrada y salida

Conforme se ha explicado en la sección 5.3.2, se adoptan las siguientes variables de entrada:  $\{X_1 = f_{ck}, X_2 = b, X_3 = h, X_4 = A_{st}\}$  (ya que se ha dividido la base de datos conforme a las dos resistencias características de acero estudiadas). La variable de salida es:  $\{Y = M_u\}$

### 5.4.2. Normalización de los datos de entrada

En este punto, es importante notar que las variables están en diferentes unidades y, consecuentemente, con diferentes órdenes de magnitud, de manera que, para prevenir que esta situación afecte a los resultados, se debe “normalizar”

los datos, esto es, para cada variable, hay que realizar la siguiente operación:

$$x = \frac{X - X_{\text{mín}}}{X_{\text{máx}} - X_{\text{mín}}} \quad (5.11)$$

de esta manera el dominio de todas las variables normalizadas es:  $0 \leq x \leq 1$ .

Para el caso de Aprendizaje Automático Extremo (ELM), se utiliza la normalización en el dominio  $-1 \leq x \leq 1$ , que se logra con la siguiente transformación:

$$x = 2 \left( \frac{X - X_{\text{mín}}}{X_{\text{máx}} - X_{\text{mín}}} \right) - 1 \quad (5.12)$$

tal como lo recomienda Guan-Bin Huang (2013).

### 5.4.3. División en conjuntos de entrenamiento y verificación

- **Conjunto de entrenamiento:** selección, al azar, de 10.000 registros;
- **Conjunto de verificación:** de los registros restantes, se selecciona al azar 400 para el conjunto de verificación.

## 5.5. Entrenamiento

### 5.5.1. Regresión lineal

Se sigue el algoritmo presentado en 3.3.1.2 con el cual se llega a las siguientes ecuaciones de hiperplanos de mínimos cuadrados para los dos valores utilizados como resistencias características del acero:

$$\begin{cases} \hat{M}_u = 4,41f_{ck} - 5,99b + 10,88h + 24,06A_{st} - 673,27 & \text{para: } f_{yk} = 400 \text{ MPa} \\ \hat{M}_u = 6,84f_{ck} - 6,77b + 13,38h + 28,57A_{st} - 858,73 & \text{para: } f_{yk} = 500 \text{ MPa} \end{cases} \quad (5.13)$$

$\hat{M}_u$  en kNm,  $f_{ck}$  en MPa,  $b, h$  en cm,  $A_{st}$  en cm<sup>2</sup>.

### 5.5.2. Vector soporte automático

Se utiliza el método `sklearn.svm.SVR` (Pedregosa y col., 2011) basado a en Chang y Lin (2003).

Los mejores resultados se han obtenido con las adopciones siguientes:

- **Parámetros:**  $C = 200$ ;  $\xi_i = 0,01$  (ec. (3.55), página 51);
- **Kernel:** Función de base radial con  $\gamma = 0,2$  (tabla 3.1, página 53)

### 5.5.3. Redes neuronales artificiales

Se utiliza el método `sklearn.neural_network.MLPRegressor` (Pedregosa y col., 2011), el cual es un regresor de perceptrón multi-capas (apartado 3.3.3.2) que minimiza el error cuadrático usando el descenso estocástico del gradiente<sup>2</sup> y otros métodos.

La siguiente elección de parámetros consigue buenos resultados:

- Cantidad de capas ocultas: 1 (figura 3.6)
- Número de neuronas en la capa oculta: aplicando (3.100) con  $\alpha = 10$ , se obtiene: 200
- Función de activación: sigmoide logística:  $g(z) = \frac{1}{1 + e^{-z}}$
- Método de optimización: 'lbfgs', algoritmo de Broyden – Fletcher – Goldfarb – Shanno de memoria limitada (Nocedal y Wright, 2006).
- Razón de aprendizaje:  $\alpha = 0,001$  (ec. (3.79))

### 5.5.4. Aprendizaje automático extremo

Se realiza con el código presentado en el anexo B.2 creado por el autor, con los siguientes parámetros:

---

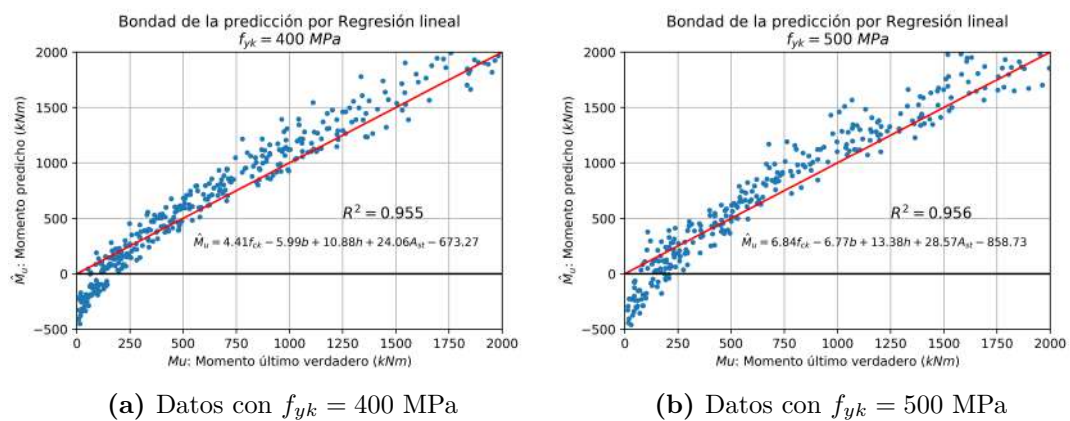
<sup>2</sup>Ec. (3.79) de la página 59.

- Número de neuronas en la capa oculta: 200
- Función de activación:  $g(z) = \sin(z)$

## 5.6. Verificación

### 5.6.1. Regresión lineal

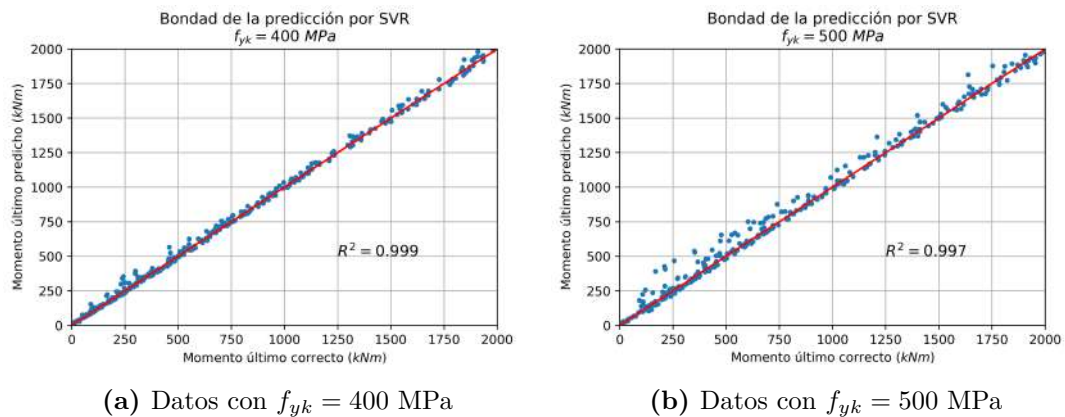
Se comprueban las ecuaciones (5.13) con los datos de verificación, los resultados son reflejados en la figura 5.4:



**Figura 5.4** – Momento último predicho vs. Momento último verdadero, el ajuste sería excelente si los puntos estuvieran sobre la recta a  $45^\circ$ .

### 5.6.2. Vector soporte automático

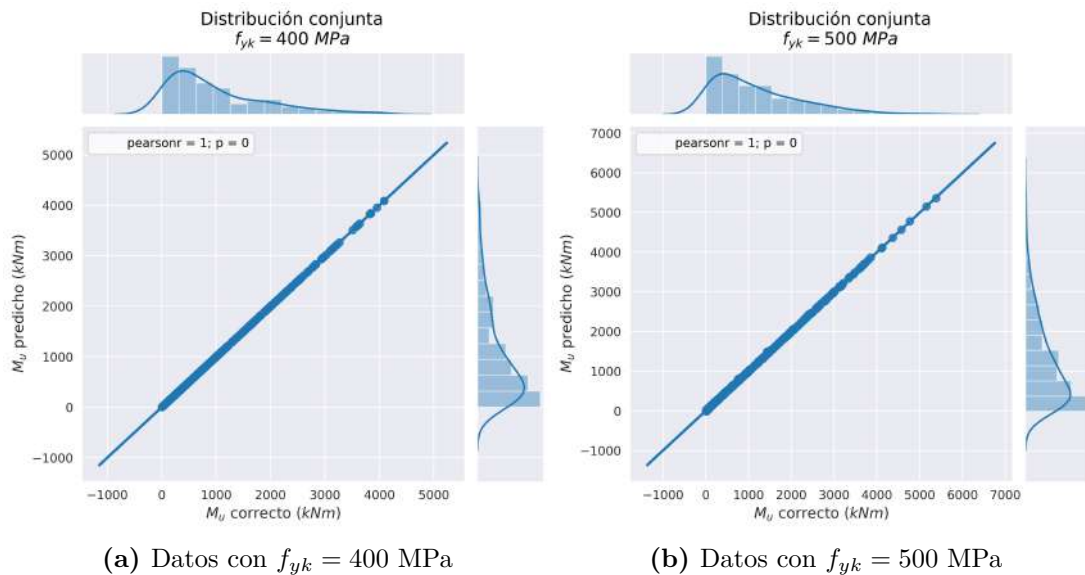
La comparación entre la predicción y los resultados correctos se da a continuación:



**Figura 5.5** – Momento último predicho por el método de Vector Soporte Automático vs. Momento último correcto. Cuanto más cerca los puntos de la recta a  $45^\circ$ , mejor el ajuste.

### 5.6.3. Redes neuronales

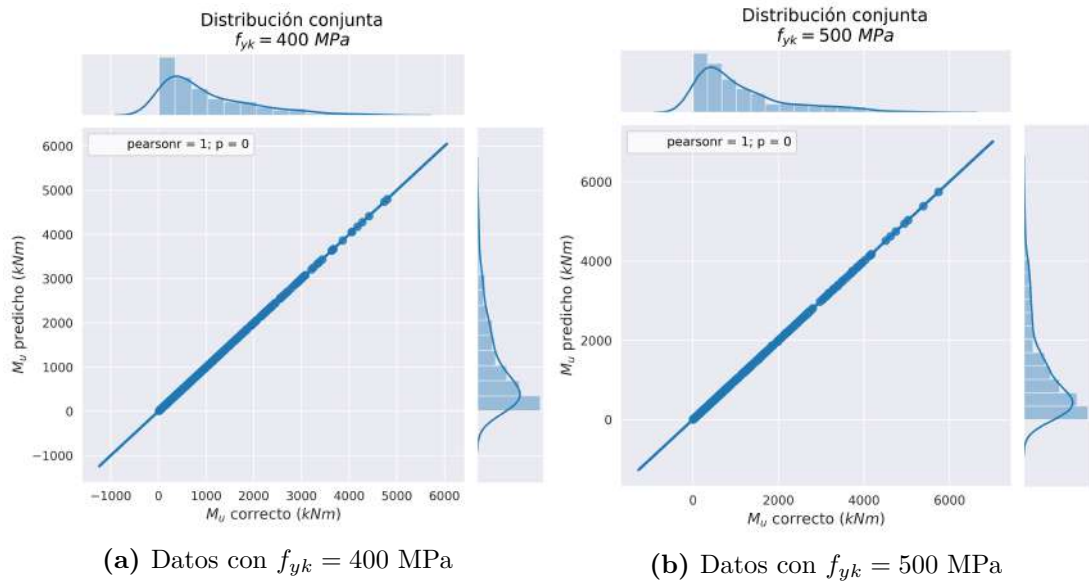
Se muestra la distribución conjunta de los valores correctos y los valores predichos para  $f_{ck} = 400$  MPa en la figura 5.6a y para  $f_{ck} = 500$  MPa en la figura 5.6b.



**Figura 5.6** – Distribución conjunta de momentos últimos correctos y predichos para (a)  $f_{yk} = 400$  MPa y (b)  $f_{yk} = 500$  MPa, por el método de redes neuronales artificiales. *Fuente:* elaboración propia.

### 5.6.4. Aprendizaje automático extremo

Se muestra la distribución conjunta de valores correctos y valores predichos para  $f_{yk} = 400$  MPa y  $f_{yk} = 500$  MPa en la figura 5.7.



**Figura 5.7** – Distribución conjunta de momentos últimos correctos y predichos para (a)  $f_{yk} = 400$  MPa y (b)  $f_{yk} = 500$  MPa, por el método de aprendizaje automático extremo. *Fuente:* elaboración propia.

## 5.7. Comprobación de casos particulares

Datos de entrada	Valor correcto (kNm)	Predicciones (kNm)			
		LR	SVM	ANN	ELM
$f_{yk} = 400$ MPa $f_{ck} = 18$ MPa $b = 15$ cm $h = 35$ cm $A_{st} = 2,26$ cm <sup>2</sup>	28,32	-195,75 (-791,20 %)	30,03 (6,05 %)	28,81 (1,75 %)	26,65 (-5,90 %)
$f_{yk} = 400$ MPa $f_{ck} = 20$ MPa $b = 20$ cm $h = 60$ cm $A_{st} = 8,04$ cm <sup>2</sup>	168,97	141,19 (-16,44 %)	169,44 (0,28 %)	167,32 (-0,97 %)	166,63 (-1,38 %)
$f_{yk} = 500$ MPa $f_{ck} = 30$ MPa $b = 40$ cm $h = 90$ cm $A_{st} = 68,72$ cm <sup>2</sup>	2473,02	2243,10 (-9,30 %)	2482,03 (0,36 %)	2500,17 (1,10 %)	2511,72 (1,56 %)

**Tabla 5.7** – Comparación de las predicciones de momento último, dadas por los métodos estudiados en tres casos particulares de vigas pequeña, mediana y grande. Entre paréntesis los porcentajes de error. Recubrimiento de 2,5 cm en todos los casos.

# Capítulo 6

## ANÁLISIS DE DEFORMACIONES

A continuación se presenta el análisis de la base de datos de flechas<sup>1</sup>, generada según los criterios de la sección 4.5.2. La estructura del análisis es similar a la del capítulo 5.

Se utiliza el método PCA en primer término y posteriormente se aborda el problema de predecir la flecha en una viga de hormigón armado con sección rectangular constante, simplemente apoyada, armadura uniforme y sometida a carga uniformemente distribuida en toda su longitud, utilizando las técnicas de aprendizaje automático, LR, SVR, ANN y ELM, ya presentadas en capítulos anteriores.

### 6.1. Visión general de la base de datos

Conforme se presentó en el tabla 4.2, en una muestra de la base de datos, las variables de entrada son:

$$X_1 = f_{yk}, X_2 = f_{ck}, X_3 = b, X_4 = h, X_5 = L, X_6 = A_{st}, X_7 = q \quad (6.1)$$

---

<sup>1</sup>Deformación máxima vertical.

y la de salida:

$$Y = f \quad (6.2)$$

La matriz de datos de entrada tiene la forma:

$$\mathbf{X} = \begin{bmatrix} f_{yk}^{(1)} & f_{ck}^{(1)} & b^{(1)} & h^{(1)} & L^{(1)} & A_{st}^{(1)} & q^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ f_{yk}^{(N)} & f_{ck}^{(N)} & b^{(N)} & h^{(N)} & L^{(N)} & A_{st}^{(N)} & q^{(N)} \end{bmatrix} \quad (6.3)$$

Los valores que pueden tomar cada una de estas variables, se definieron en 4.5.2. El vector de salida es:

$$\mathbf{y} = \begin{bmatrix} f^{(1)} \\ \vdots \\ f^{(N)} \end{bmatrix} \quad (6.4)$$

Estas variables fueron definidas en la página 86. En este caso  $N = 2.628$ .

Los valores medios  $\bar{x}$ , las desviaciones estándar  $s$  y los coeficientes de variación  $s/\bar{x}$  son:

Variable	$\bar{x}$	$s$	Unidad	$s/\bar{x}$
$f_{yk}$	443,5	49,6	MPa	0,11
$f_{ck}$	20,6	3,7	MPa	0,18
$b$	24,8	4,0	cm	0,16
$h$	49,9	15,6	cm	0,31
$L$	5,63	1,67	m	0,30
$A_{st}$	21,9	13,5	cm <sup>2</sup>	0,62
$q$	48,7	31,9	kN	0,66
$f$	12,8	6,7	mm	0,52

**Tabla 6.1** – Estadísticos de las variables para el análisis de flechas.  $N = 2.628$ .

## 6.2. Gráficas

En la figura 6.1 se observan nubes de puntos que relacionan las variables  $h$ ,  $L$ ,  $A_{st}$ ,  $q$  y  $f$  entre sí, para tener una apreciación general de la base de datos.



**Figura 6.1** – Nube de puntos de la base de datos de flechas. Se muestran solamente las variables  $h$ ,  $L$ ,  $Ast$ ,  $q$  y  $f$  y sus relaciones entre sí. *Fuente:* elaboración propia.

## 6.3. Análisis de componentes principales

### 6.3.1. Estandarización de las variables de entrada

Las variables de entrada se definieron en 6.1, se procede de manera análoga al apartado 5.3.1 estandarizándolas, esto es, haciendo la transformación  $Y_i = \frac{X_i - \mu_i}{\sigma_i}$ , ec. (3.15), con lo cual, las mismas tienen medias iguales a uno y desviaciones estándares iguales a cero.

### 6.3.2. Matriz de correlaciones

Una vez expresadas las variables en unidades estandarizadas, la matriz de correlación es:

$$\mathbf{R} = \begin{array}{cccccccc} & f_{yk} & f_{ck} & b & h & L & A_{st} & q \\ \left[ \begin{array}{cccccccc} & 1 & 0,021 & 0,002 & 0,016 & 0,007 & -0,039 & 0,024 \\ & 0,021 & 1 & 0,000 & -0,012 & -0,009 & 0,060 & 0,118 \\ & 0,002 & 0,000 & 1 & 0,069 & 0,068 & 0,301 & 0,265 \\ & 0,016 & -0,012 & 0,069 & 1 & 0,922 & 0,527 & 0,190 \\ & 0,007 & -0,009 & 0,068 & 0,922 & 1 & 0,483 & 0,018 \\ & -0,039 & 0,060 & 0,301 & 0,527 & 0,018 & 1 & 0,598 \\ & 0,024 & 0,118 & 0,265 & 0,190 & 0,018 & 0,598 & 1 \end{array} \right] & \begin{array}{l} f_{yk} \\ f_{ck} \\ b \\ h \\ L \\ A_{st} \\ q \end{array} \end{array} \quad (6.5)$$

La mayor correlación se observada entre  $h$  y  $L$  debido a la relación luz/canto que se ha establecido en la página 87. Se decide mantener todas variables, es decir, no descartar ninguna de las presentadas.

### 6.3.3. Valores característicos

Los valores característicos de la matriz de correlaciones (6.5) son:

$$\mathbf{\Lambda} = \begin{array}{cccccccc} & \left[ \begin{array}{cccccccc} 2,512 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1,384 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1,023 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,979 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,762 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,285 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,057 \end{array} \right] & \end{array} \quad (6.6)$$

Con los cuales se calculan los porcentajes explicados por cada componente:

Componente	Autovalor	% Explicado	% Acumulado
$Z_1$	2,512	35,88 %	39,51 %
$Z_2$	1,384	19,76 %	55,64 %
$Z_3$	1,023	14,60 %	70,24 %
$Z_4$	0,979	13,98 %	84,22 %
$Z_5$	0,762	10,89 %	95,11 %
$Z_6$	0,285	4,07 %	99,18 %
$Z_7$	0,057	0,82 %	100,00 %

**Tabla 6.2** – Porcentaje explicado, por cada componente principal, para el cálculo de flechas. *Fuente:* elaboración propia.

Siguiendo el criterio de Kaiser (Grané Chávez, 2010), que selecciones los autovalores mayores a uno, debemos adoptar las componentes  $Z_1$ ,  $Z_2$  y  $Z_3$ , con los cuales el porcentaje de variabilidad explicado es de 70,25 %.

#### 6.3.4. Vectores característicos

La matriz de vectores característicos  $\mathbf{Q}$  en la descomposición espectral (3.20)  $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$  es:

$$\mathbf{Q} = \begin{bmatrix} 0,0008 & -0,0026 & -0,7508 & -0,6565 & 0,0272 & 0,0677 & -0,0026 \\ 0,0389 & -0,2414 & -0,6102 & 0,6877 & -0,3071 & 0,0143 & -0,0193 \\ 0,2105 & -0,4580 & 0,2338 & -0,3065 & -0,767 & -0,0917 & -0,0246 \\ 0,5498 & 0,3599 & -0,0514 & 0,0251 & -0,0281 & -0,3341 & -0,6727 \\ 0,5172 & 0,4451 & -0,0454 & 0,034 & -0,1658 & -0,068 & 0,7064 \\ 0,5284 & -0,2517 & 0,0640 & 0,0174 & 0,2188 & 0,7715 & -0,0994 \\ 0,3242 & -0,5839 & -0,0216 & -0,0071 & 0,4904 & -0,5247 & 0,1941 \end{bmatrix} \quad (6.7)$$

### 6.3.5. Componentes principales

Utilizando la ecuación (3.21)  $\mathbf{Z} = \mathbf{Q}^T \mathbf{X}$  y tomando las primeras tres componentes, se tiene:

$$\begin{cases} Z_1 = 0,001\tilde{f}_{yk} + 0,039\tilde{f}_{ck} + 0,211\tilde{b} + 0,550\tilde{h} + 0,517\tilde{L} + 0,528\tilde{A}_{st} + 0,324\tilde{q} \\ Z_2 = -0,003\tilde{f}_{yk} - 0,241\tilde{f}_{ck} - 0,458\tilde{b} + 0,360\tilde{h} + 0,445\tilde{L} - 0,252\tilde{A}_{st} - 0,584\tilde{q} \\ Z_3 = -0,751\tilde{f}_{yk} - 0,610\tilde{f}_{ck} + 0,234\tilde{b} - 0,051\tilde{h} - 0,045\tilde{L} + 0,064\tilde{A}_{st} - 0,022\tilde{q} \end{cases} \quad (6.8)$$

Se utilizan tildes sobre las variables para resaltar que se encuentran estandarizadas.

### 6.3.6. Correlaciones entre componentes principales y variables originales

De acuerdo con la ecuación (3.27) se obtienen las correlaciones entre las tres componentes principales  $Z$  y las variables originales  $X$ .

	$Z_1$	$Z_2$	$Z_3$
$\tilde{f}_{yk}$	-0,0101	0,2875	0,8684
$\tilde{f}_{ck}$	0,1450	0,6102	0,1602
$\tilde{b}$	0,5330	0,1785	-0,2456
$\tilde{h}$	0,5122	-0,6558	0,3760
$\tilde{L}$	0,9082	-0,1408	0,0306
$\tilde{A}_{st}$	0,7474	0,3787	-0,1391

**Tabla 6.3** – Correlaciones entre las tres primeras componentes principales y las variables originales para el cálculo de flechas. No se observa un desacoplamiento claro entre los diferentes tipos de variables.

A diferencia del tabla 5.3, en este caso, no es claro el desacoplamiento entre geometría y resistencias u otras características, las variables se encuentran más interrelacionadas.

## 6.4. Decisiones sobre la base de datos

El tamaño de la base de datos para el cálculo de flechas es  $N = 2.628$  observaciones. Se decide dividir el conjunto según los dos valores considerados de la resistencia característica del acero, con lo cual, para  $f_{yk} = 400\text{MPa}$  se tienen 1485 registros y para  $f_{yk} = 500\text{MPa}$  se tienen 1.143 registros.

Además, se adoptan todas las variables de entrada propuestas inicialmente en 6.1.

### 6.4.1. Normalización de los datos de entrada

La normalización se realiza como en los casos precedentes, esto es:

- Para LR, SVR y ANN:

$$x = \frac{X - X_{\text{mín}}}{X_{\text{máx}} - X_{\text{mín}}}$$

con lo cual:  $0 \leq x \leq 1$ .

- Para ELM:

$$x = 2 \left( \frac{X - X_{\text{mín}}}{X_{\text{máx}} - X_{\text{mín}}} \right) - 1$$

con lo cual:  $-1 \leq x \leq 1$

### 6.4.2. División en conjuntos de entrenamiento y verificación

De acuerdo a los datos disponibles, se dividen los conjuntos de entrenamiento y verificación, para cada valor de  $f_{yk}$ , de la siguiente manera:

- **Conjunto de entrenamiento:** se seleccionan al azar,  $NE = 1.000$  registros.
- **Conjunto de verificación:** de los registros restantes, se seleccionan al azar  $NV = 100$  de ellos para el conjunto de verificación.

## 6.5. Entrenamiento

### 6.5.1. Regresión lineal

Nuevamente, con el algoritmo de la sección 3.3.1.2, se encuentran las siguientes ecuaciones de mínimos cuadrados:

$$\left\{ \begin{array}{l} \text{Para } f_{yk} = 400 \text{ MPa :} \\ \hat{f} = -0,163f_{ck} - 0,187b - 0,451h + 7,127L - 0,267A_{st} + 0,202q - 1,036 \\ \\ \text{Para } f_{yk} = 500 \text{ MPa :} \\ \hat{f} = -0,185f_{ck} - 0,156b - 0,433h + 7,218L - 0,293A_{st} + 0,204q - 1,963 \end{array} \right. \quad (6.9)$$

$\hat{f}$  en mm,  $f_{ck}$  en MPa,  $b, h$  en cm,  $L$  en m,  $A_{st}$  en  $\text{cm}^2$  y  $q$  en  $\text{kN/m}$ .

### 6.5.2. Vector soporte automático

Análogamente a la sección 5.5.2, los mejores resultados se han obtenido con las adopciones siguientes:

- **Parámetros:**  $C = 200$ ;  $\xi_i = 0,01$  (ec. (3.55), página 51);
- **Kernel:** Función de base radial con  $\gamma = 0,2$  (tabla 3.1, página 53)

### 6.5.3. Redes neuronales

Se ha utilizado el mismo procedimiento indicado en 5.5.3, con la siguiente elección de parámetros:

- Cantidad de capas ocultas: 1 (figura 3.6)
- Número de neuronas en la capa oculta: aplicando (3.100) con  $\alpha = 5$ , se obtiene: 28,6; se adopta 30.
- Función de activación: tangente hiperbólica:  $g(z) = \tanh(z)$

- Método de optimización: 'lbfgs', algoritmo de Broyden – Fletcher – Goldfarb – Shanno de memoria limitada (Nocedal y Wright, 2006).
- Razón de aprendizaje:  $\alpha = 0,001$  (ec. (3.79))

#### 6.5.4. Aprendizaje automático extremo

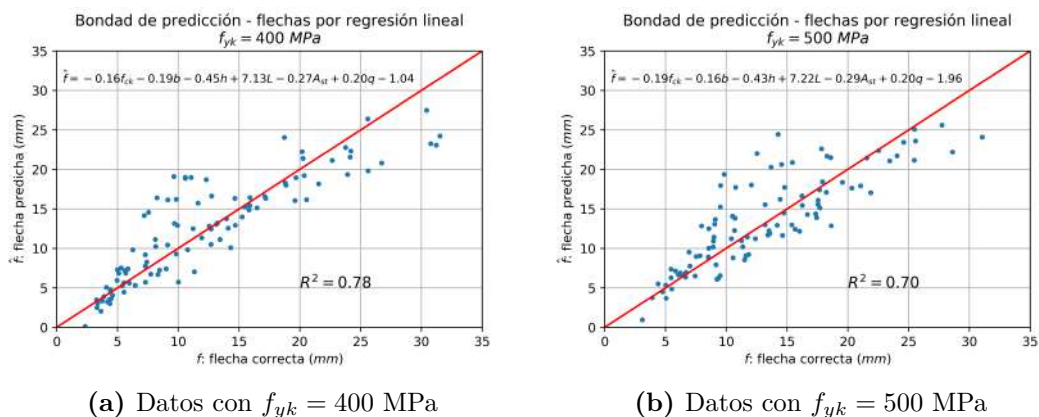
Los parámetros adoptados son:

- Número de neuronas de la capa oculta:  $L = 200$
- Función de activación: sigmoide ponderado  $g(z) = \frac{z}{1 - e^{-z}}$

### 6.6. Verificación

Los resultados de la fase de entrenamiento se ponen a prueba en los siguientes gráficos:

#### 6.6.1. Regresión lineal

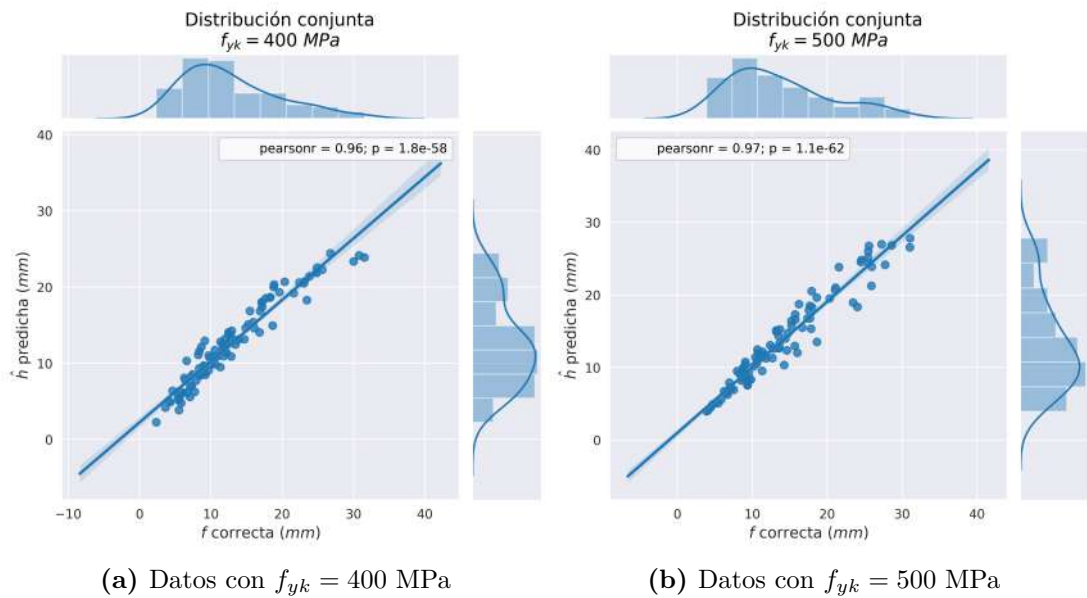


**Figura 6.2** – flecha predicha vs. flecha correcta, y su bondad de ajuste por el método de regresión lineal.

La bondad de ajuste es bastante pobre por lo que **se descarta este método para predicción de flechas.**

### 6.6.2. Vector soporte automático

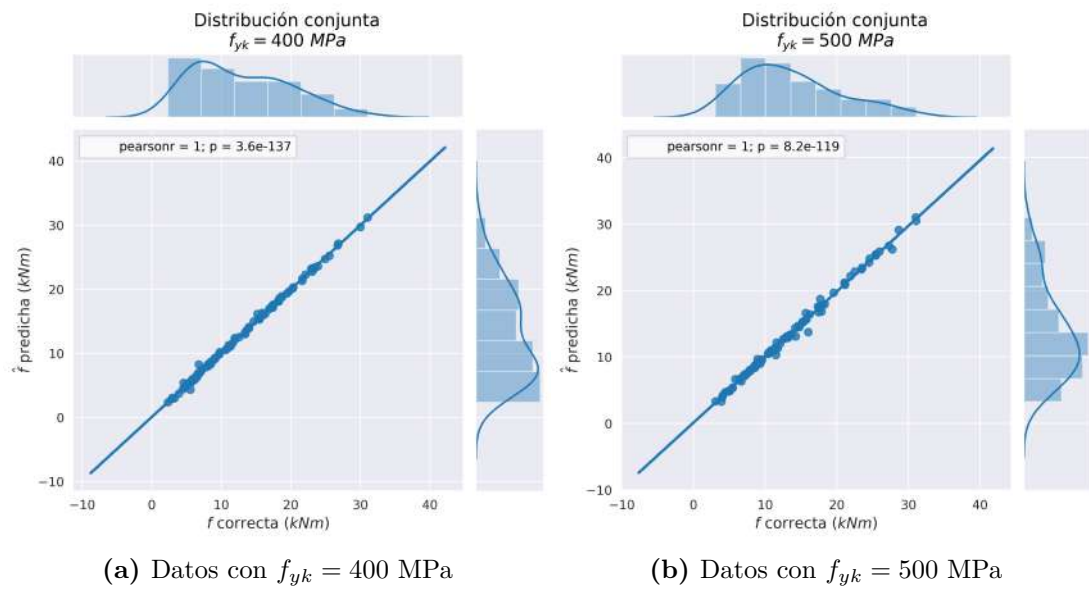
La comparación entre la predicción y los resultados correctos se dan a continuación:



**Figura 6.3** – Flecha predicha por el método Vector Soporte Automático vs. flecha correcta.

### 6.6.3. Redes neuronales

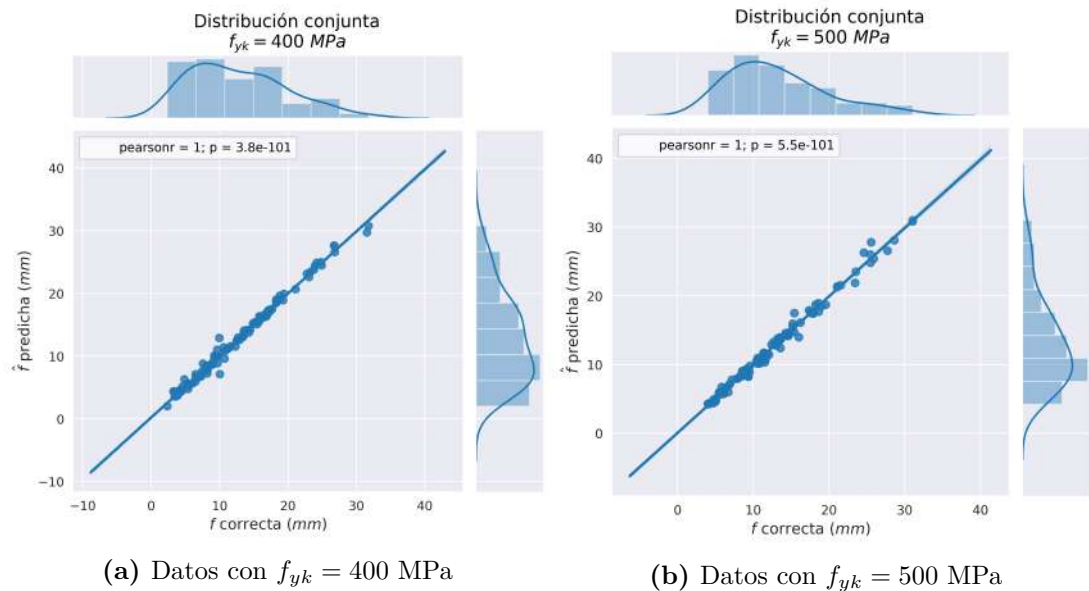
Las predicciones sobre la base de datos de verificación, para los dos valores de resistencia de acero considerados, se muestran a continuación en la figura 5.6:



**Figura 6.4** – Flecha predicha por el método de Redes Neuronales Artificiales vs. flecha correcta.

#### 6.6.4. Aprendizaje automático extremo

Se muestra el contraste entre las predicciones y los valores adoptados como correctos para el aprendizaje supervisado:



**Figura 6.5** – Flecha predicha por el método ELM vs. flecha correcta.

## 6.7. Comprobación de casos particulares

Datos de entrada	Correcto (mm)	Predicciones (mm)		
		SVM	ANN	ELM
$f_{yk} = 400$ MPa $f_{ck} = 16$ MPa $b = 20$ cm $h = 30$ cm $L = 4,0$ m $A_{st} = 13,63$ cm <sup>2</sup> $q = 18,81$ kN/m	6,3	7,5 (19,0%)	6,6 (4,8%)	6,5 (3,2%)
$f_{yk} = 400$ MPa $f_{ck} = 20$ MPa $b = 25$ cm $h = 50$ cm $L = 6,0$ m $A_{st} = 28,67$ cm <sup>2</sup> $q = 62,55$ kN/m	17,4	17,9 (2,9%)	17,3 (-0,6%)	17,5 (0,6%)
$f_{yk} = 500$ MPa $f_{ck} = 25$ MPa $b = 30$ cm $h = 80$ cm $L = 8,0$ m $A_{st} = 73,76$ cm <sup>2</sup> $q = 180,0$ kN/m	24,6	24,3 (-1,2%)	24,2 (-1,6%)	24,6 (0,0%)

**Tabla 6.4** – Comprobación de predicción de flechas, por vector soporte automático, redes neuronales y aprendizaje automático extremo, en casos particulares, considerando vigas pequeña, mediana y grande. Se muestran los errores entre paréntesis. Recubrimiento de 2,5 cm en todos los casos.

## 6.8. Selección del método y cálculos adicionales

Los métodos que se muestran más robustos y mejor consistencia en el análisis de regresión son ANN y ELM.

### 6.8.1. Predicción de la resistencia del hormigón a partir de la flecha

Se supone que se conocen los siguientes datos:

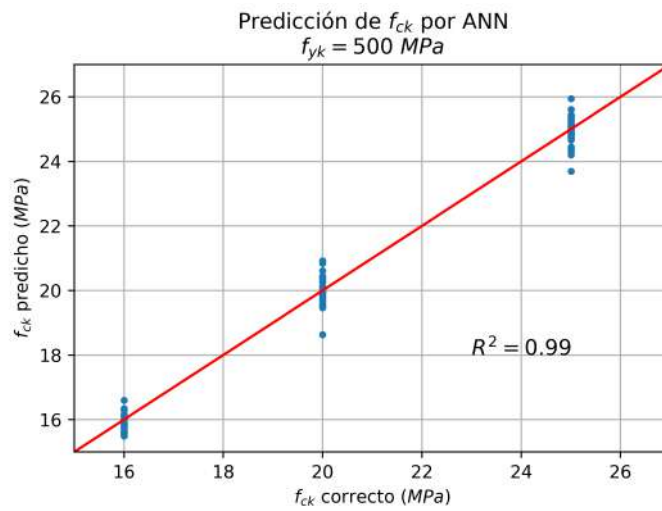
- Resistencia característica del acero:  $f_{yk}$ ;
- Base, altura y longitud de la viga:  $b$ ,  $h$  y  $L$ ;
- Armados de tracción y compresión:  $A_{st}$  y  $A_{sc}$ ;
- Carga externa:  $q$ ;
- Recubrimiento:  $rec$ .

y se desea averiguar la resistencia característica del hormigón  $f_{ck}$ .

Se entrena una red neuronal con las siguientes características:

- Cantidad de capas ocultas: 2;
- Número de neuronas: 100 y 50 respectivamente en cada capa oculta;
- Función de activación: tangente hiperbólica:  $g(z) = \tanh(z)$ ;
- Método de optimización: 'lbfgs', algoritmo de Broyden – Fletcher – Goldfarb – Shanno de memoria limitada;
- Razón de aprendizaje:  $\alpha = 0,001$ ;

Los resultados se reflejan en la figura 6.6:

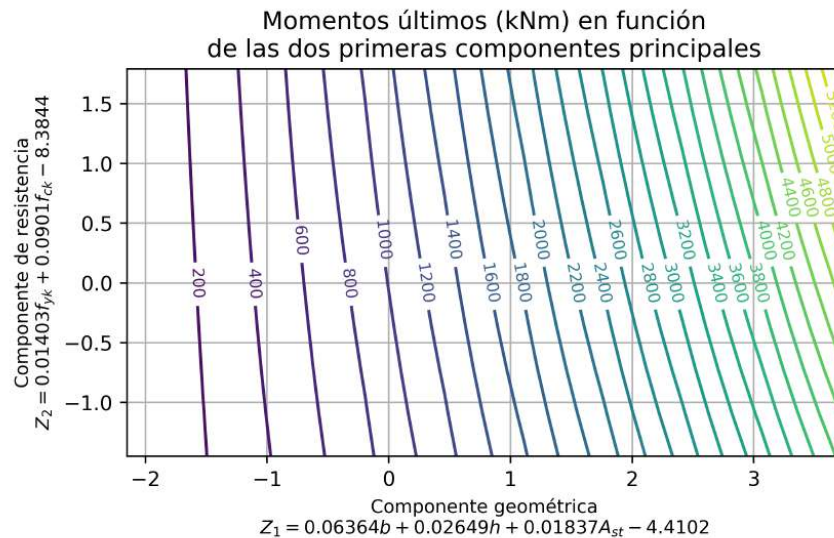


# Capítulo 7

## CONCLUSIONES

### 7.1. Análisis de componentes principales

El análisis de componente principales, aplicado al estudio de las variables que determinan el momento último de una sección, arroja un hermoso desacoplamiento entre variables geométricas y de resistencia. Este hecho que se refleja en el gráfico 5.3, el cual se repite a continuación:



**Figura 5.3** – Momentos últimos (capacidad resistente en kNm) de una sección rectangular de hormigón armado en función de las componentes principales  $Z_1$  y  $Z_2$ ;  $f_{yk}$  y  $f_{ck}$ , resistencias características del acero y el hormigón, respectivamente, en MPa;  $b$  y  $h$ , base y altura de la sección, en cm;  $A_{st}$ , área armadura de tracción, en  $\text{cm}^2$  (repetido de la página 96).

## 7.2. Predicciones de momento último

### 7.2.1. Regresión lineal

- Para momentos pequeños ( $< 500$  MPa), en general, la predicción es inferior al valor correcto (sobre dimensionamiento), sin embargo, también puede ocurrir lo inverso, los momentos predichos podrían ser mayores a los correctos (sub-dimensionamiento);
- Para vigas pequeñas, hay casos con momentos predichos negativos, lo cual, en el contexto de análisis, es absurdo;
- Con ambos valores de  $f_{yk}$  se encuentran elevados valores de  $R^2$  (0,95+), los cuales dan una impresión de buen ajuste, pero se debe considerar lo mencionado en el primer punto, como razón de este hecho;
- En la verificación de casos particulares los valores predichos no están lo suficientemente cerca de los valores correctos;
- Concluimos que **el método de regresión lineal no es adecuado** para realizar predicciones de momento último.

### 7.2.2. Vector soporte automático

- La predicción es bastante mejor en relación al método de regresión lineal;
- Se tiene un coeficiente  $R^2$  muy cercano a la unidad, lo que indica un excelente ajuste;;
- Se puede notar que la mayoría de los puntos por fuera de la recta a  $45^\circ$ , en la figura 5.5 están por encima de esta, lo que indica un momento predicho mayor al correcto, esto significaría un ligero sub-dimensionamiento de la sección.
- La comprobación de casos particulares, tabla 5.7, arroja un error relativamente importante en la viga de pequeña capacidad, para los demás casos, el error es despreciable.
- La técnica SVM muestra ser una poderosa herramienta para predicción de momentos últimos.

### 7.2.3. Redes neuronales

- La predicción mejora en todos los casos respecto a las técnicas anteriores;
- El ajuste tiene un mejor coeficiente de Pearson y, por lo tanto, un mejor coeficiente  $R^2$ , para ambos valores de resistencias de acero;
- En el caso de vigas “pequeñas” se observa una interesante mejoría respecto a las técnicas LR y SVM;
- En general, para todos los casos verificados, este método presenta una buena consistencia en los resultados, mostrando ser una herramienta robusta en la predicción de momentos últimos.

### 7.2.4. Aprendizaje automático extremo

- El ajuste es bastante razonable considerando la simplicidad relativa de este método;
- Los resultados se consiguen con un costo computacional menor en relación a los demás métodos analizados;
- Los coeficientes de bondad de ajuste indican excelente correlación entre los valores correctos y los predichos;
- En los casos particulares considerados, mayor diferencia se observa en vigas pequeñas;

## 7.3. Predicciones de flecha

### 7.3.1. Vector soporte automático

- La mayor discrepancia se observa en vigas pequeñas, pero se debe señalar que el caso de prueba está fuera de los rangos considerados en la base de datos;
- En la mayoría de los casos se ven aproximaciones razonables.

### 7.3.2. Redes neuronales artificiales

- La predicción se muestra consistente en todos los casos, con lo cual el método demuestra ser bastante robusto;
- La predicción puede aumentar de precisión al modificar los parámetros, por ejemplo aumentando la cantidad de neuronas, sin embargo con los valores adoptados se considera que los resultados están dentro del margen de precisión esperado.

### 7.3.3. Aprendizaje automático extremo

- El ajuste es muy bueno, considerando los coeficientes de correlación y los casos particulares revisados;
- La cantidad de neuronas necesarias es bastante superior al caso de ANN, sin embargo no implica una gran diferencia en cuanto al costo computacional;
- La función de activación que arrojó mejores resultados es el sigmoide ponderado, el cual no se había utilizado aún en los demás casos estudiados;

## 7.4. Conclusiones finales

Los métodos analizados fueron Regresión Lineal (LR), Vector Soporte Automático (SVM), Redes Neuronales Artificiales (ANN) y Aprendizaje Automático Extremo (ELM). Previamente se realizó el Análisis de Componentes Principales (PCA), sobre dos bases de datos, la primera de momentos últimos en secciones transversales y la segunda de flechas, todas referidas a vigas de hormigón armado.

1. El análisis de componentes principales PCA, permitió desacoplar las variables geométricas de las de resistencia, con lo cual se pudo determinar el momento flector último de una sección a partir de un solo gráfico 5.3;
2. La altura es la variable más sensible en la determinación de la capacidad resistente de la viga, seguida de la capacidad mecánica del acero;
3. La resistencia característica del hormigón es la variable que menos influye en la capacidad resistente de la estructura;

4. Con el método PCA se pueden confeccionar otros gráficos para dimensionamiento y verificación de vigas de hormigón armado u otros elementos estructurales que facilitarían el cálculo;
5. El método LR, no es adecuado para predicciones en estructuras de hormigón armado;
6. SVM, ANN y ELM son métodos que arrojaron predicciones bastantes razonables, de estas, la técnica ANN ha demostrado ser la más robusta y precisa y la técnica ELM de menor costo computacional.
7. Se ha visto en 6.8.1 que es posible determinar la resistencia característica del hormigón a partir de la flecha y las demás variables con muy buena precisión, se ha encontrado una estimación expresada en la ec. (6.10):

$$f_{ck} = \hat{f}_{ck} \pm 1 \text{ MPa}$$

8. Con lo anterior se demuestra que es totalmente posible predecir, con precisiones razonables, prácticamente cualquier variable desconocida, en función de las demás, por ejemplo, otra evaluación de interés para casos prácticos, podría ser determinar la cuantía de armado, esto se puede realizar fácilmente a partir de estas técnicas.
9. La ciencia de datos, ciencias de la computación y afines, pueden brindar importantes herramientas para la ingeniería de estructuras.

## 7.5. Recomendaciones y trabajos futuros

1. Utilizar análisis de componentes principales para confeccionar gráficos o nuevos métodos de dimensionamiento y verificación de estructuras;
2. Considerar seriamente las técnicas de aprendizaje automático para el cálculo de estructuras;
3. Fomentar el uso de las ciencias de la computación en la ingeniería estructural;
4. Investigar sobre el uso de técnicas de procesamiento de imágenes para determinar flechas en vigas de hormigón armado, con lo cual será posible estimar otros parámetros desconocidos de la misma;

5. Investigar la coloración de las mezclas de hormigón para estimación de su resistencia.
6. Ampliar el estudio aquí presentado a otras situaciones y otros elementos estructurales, considerando mayor variabilidad de los parámetros, en computadoras más potentes que puedan analizar mayor cantidad de información.

# APÉNDICES

# Apéndice A

## Módulos para hormigón armado

### A.1. Manual de uso: materiales.py

En el módulo `materiales.py` se definen los materiales hormigón y acero con sus propiedades principales.

*Todas las unidades son del Sistema Internacional sin múltiplos ni submúltiplos.*

#### A.1.1. Clase: Hormigón

##### A.1.1.1. Atributos

Un material de tipo **hormigón** para análisis no lineal se define en la clase `Hn1`. Sus atributos son:

`Hn1.traccion`

Considerar tracción. Tipo: booleano. Por defecto: `True`

`Hn1.n`

Exponente del diagrama de tracción Wang & Tsu. Tipo: número real. Por defecto: 0,4.

`Hn1.alpha_cc`

Coefficiente de cansancio a compresión. Tipo: número real. Por defecto: 1.0.

`Hn1.alpha_ct`

Coefficiente de cansancio a tracción. Tipo: número real. Por defecto: 1.0.

**Hnl.agregado**

Tipología de agregado. Tipo: cadena. Por defecto: 'basalto'. Valores posibles: 'arenisca', 'basalto', 'caliza', 'cuarcita'

**Hnl.nu**

Módulo de Poisson. Tipo: número real. Por defecto: 0.2

**Hnl.densidad**

Densidad del hormigón. Tipo: número real. Por defecto: 2400. Unidad:  $\text{kg}/\text{m}^3$

**Hnl.est\_lim**

Estado límite. Tipo: cadena. Por defecto: 'VMS'. Valores posibles: 'ELU', 'ELS', 'VMS', estado límite último, estado límite de servicio y valores medios respectivamente.

**Hnl.fk**

Resistencia característica. Tipo: número real. Por defecto:  $25 \times 10^6$ . Unidad:  $\text{N}/\text{m}^2 = \text{Pa}$ .

**A.1.1.2. Propiedades derivadas**

Una vez definidos los valores anteriores, los siguientes pueden ser obtenidos:

**Hnl.fc**

Resistencia de cálculo a compresión.

**Hnl.fct**

Resistencia de cálculo a tracción.

**Hnl.Ec**

Módulo de elasticidad

**Hnl.epsilon\_cr**

Deformación crítica a tracción en tanto por uno.

**Hnl.sigma\_c(epsilon\_c)**

Devuelve la tensión en el hormigón dada la deformación unitaria `epsilon_c`

**Hnl.area\_c(a,b)**

Devuelve el área bajo la curva tensión-deformación del hormigón en valores medios, desde `a` hasta `b` con `a < b`.

`Hnl.momento_estatico_c(a, b)`

Devuelve el momento estático respecto al origen en valores medios de la curva del hormigón comprendida entre 'a' y 'b'.

`Hnl.ordenadas_c(Epsilon)`

Devuelve las ordenadas del diagrama de tensión - deformación del hormigón, dado un `ndarray`<sup>1</sup> de valores deformación unitaria `Epsilon`.

## A.1.2. Clase: Acero

### A.1.2.1. Atributos

Un material de tipo **Acero** se define en la clase **Acero**. Sus atributos son:

`Acero.fk`

Resistencia característica. Tipo: número real. Por defecto:  $500 \times 10^6$ . Unidad:  $\text{N/m}^2 = \text{Pa}$ .

`Acero.epsilon_uk`

Deformación unitaria última. Tipo: número real. Por defecto: 0,10

`Acero.epsilon_su`

Deformación unitaria última permitida en el interior del hormigón. Tipo: número real. Por defecto: 0,01

`Acero.k`

Carga de rotura / límite elástico. Tipo: número real. Por defecto: 1,1

`Acero.Es`

Módulo de elasticidad. Tipo: número real. Por defecto:  $200 \times 10^9$ , Unidad:  $\text{N/m}^2 = \text{Pa}$

`Hnl.est_lim`

Estado límite. Tipo: cadena. Por defecto: 'VMS'. Valores posibles: 'ELU', 'ELS', 'VMS', estado límite último, estado límite de servicio y valores medios respectivamente.

`Acero.densidad`

Densidad del acero. Tipo: número real. Por defecto: 7850. Unidad:  $\text{kg/m}^3$

---

<sup>1</sup>Tipo base de `numpy`

**A.1.2.2. Propiedades derivadas**

`Acero.gamma_s`

Coficiente parcial de seguridad.

`Acero.fym`

Resistencia media.

`Acero.fyd`

Resistencia de cálculo

`Acero.epsilon_y`

Deformación unitaria en el límite elástico, en tanto por uno.

`Acero.epsilon_ud`

Deformación unitaria última de cálculo.

`Acero.fyud`

Tensión máxima de cálculo en rotura.

`Acero.area(d)`

Área de la sección transversal de una varilla de diámetro  $d$ .

`Acero.sigma_s(epsilon_s)`

Devuelve la tensión de cálculo del acero en Pa, a partir de la deformación unitaria introducida en tanto por uno.

## A.2. Manual de uso: vigas.py

En el módulo `vigas.py` se realizan todos los cálculos relacionados con las vigas, tales como momento último de una sección, diagrama de interacción, flecha, y muchos otros que se describen a continuación:

### A.2.1. Clase: Viga

#### A.2.1.1. Atributos

Una estructura de tipo **viga** se define en la clase **Viga** con lo siguientes atributos:

`Viga.b`

Base de la sección transversal. Tipo: número real. Por defecto: 0,20. Unidad: m.

`Viga.h`

Altura de la sección transversal. Tipo: número real. Por defecto: 0,50. Unidad: m.

`Viga.L`

Longitud. Tipo: número real. Por defecto: 5,0. Unidad: m.

`Viga.hormigon`

Definido en el módulo `materiales.py`

`Viga.acero`

Definido en el módulo `materiales.py`

`Viga.rec`

Recubrimiento. Tipo: número real. Por defecto: 0,025. Unidad: m.

`Viga.est_lim`

Estado límite. Tipo: cadena. Por defecto: 'VMS'. Valores posibles: 'ELU', 'ELS', 'VMS', estado límite último, estado límite de servicio y valores medios respectivamente.

`Viga.Ast`

Área de la armadura de tracción. Tipo: número real. Por defecto:  $2 \times 10^{-4}$ . Unidad: m<sup>2</sup>.

Viga.Asc

Área de armadura de compresión. Tipo: número real. Por defecto:  $1 \times 10^{-4}$ . Unidad:  $m^2$ .

Viga.NEF

Número de elementos finitos. Tipo: número entero. Por defecto: 50.

### A.2.1.2. Propiedades derivadas

Viga.d

Canto útil

Viga.Ac

Área de la sección transversal bruta  $b \times h$ .

Viga.Vol\_Ho

Volumen de hormigón.

Viga.fck

Resistencia característica del hormigón.

Viga.fyk

Resistencia característica del acero.

Viga.E\_Ho

Módulo de elasticidad del hormigón.

Viga.E\_Ao

Módulo de elasticidad del acero.

Viga.inercia

Inercia de la sección transversal.

Viga.canto\_util

Canto útil de la viga.

Viga.Kse

Factor de sistema estructural para consideraciones respecto a la flecha.

Viga.lim\_luz\_canto

Límite de la relación luz/canto con el cual puede omitirse el cálculo de flecha.

Viga.As\_min

Área mínima a tracción EC2 9.1.1.1 (9.1N)

`Viga.As_max`

Área máxima a tracción o compresión.

`Vigas.areas_varillas`

ndarray con las áreas de las secciones transversales de las varillas longitudinales de acero [m<sup>2</sup>].

`Viga.rho_0`

Cuantía de referencia. EC2 7.4.2 (2).

`Viga.rho_t`

Cuantía armadura de tracción.

`Viga.rho_c`

Cuantía armadura de compresión.

`Viga.NN`

Número de nudos del método de elementos finitos.

`Viga.gdl`

Grados de libertad

`Viga.le`

Longitud de cada elemento finito.

`Viga.x_centros`

Coordenadas de los centros de los elementos finitos.

`Viga.x_nudos`

Coordenadas de los nudos de los elementos finitos.

`Viga.rigidez_elemento`

Matriz de rigidez del elemento finito considerando 2 grados de libertad por nudo.

`Viga.rigidez_global`

Matriz de rigidez global de la viga considerando 2 grados de libertad por nudo.

`Viga.rigidez_modif`

Matriz de rigidez global modificada según el enfoque de penalización.

`Viga.pp`

Peso propio por unidad de longitud. Sin considerar coeficientes de seguridad.

**Viga.pp\_nodales**

Cargas nodales en cada elemento finito debidas al peso propio.

**Viga.carga\_unif\_dist(q)**

Cargas nodales debidas a una carga uniformemente distribuida  $q$  en toda la longitud de la viga.

**Viga.carga\_puntual(P, pos)**

Cargas nodales debidas a una carga puntual  $P$  en una posición dada  $pos$ .

**Viga.cargas\_nodales\_totales(cargas\_nodales\_ext)**

Cargas nodales totales en los elementos de la viga. Devuelve la suma de las cargas nodales externas y las debidas al peso propio.

**Viga.desp\_nodales(cargas\_nodales\_ext)**

Desplazamientos nodales: deflexiones verticales y giros.

**Viga.elastica\_lin(cargas\_nodales\_ext)**

Curva elástica obtenida considerando la teoría lineal, dadas las cargas nodales externas como `ndarray`.

**Viga.flecha\_lin(cargas\_nodales\_ext)**

Deflexión máxima calculada con la teoría lineal, dadas las cargas nodales externas como `ndarray`.

**Viga.DMF(cargas\_nodales)**

Devuelva las abcisas y ordenadas del diagrama de momento flector de la viga, dadas las cargas nodales externas.

**Viga.Md**

Momento máximo de cálculo.

**Viga.dmc()**

Devuelve `ndarray` de valores de curvatura y sus correspondientes valores de momentos flectores, para el gráfico del diagrama momento - curvatura.

**Viga.diag\_interaccion()**

Devuelve `ndarray` de valores de fuerzas normales y sus correspondientes momentos flectores para el diagrama de interacción.

**Viga.Mu**

Momento flector último que puede soportar una sección.

**Viga.diag\_curvatura(cargas\_nodales\_ext)**

Devuelve abcisas y ordenadas para el diagrama de curvaturas, dadas las cargas nodales externas.

`Viga.elastica_dmc(cargas_nodales_ext)`

Devuelve las abcisas y ordenadas para la curva elástica de la viga utilizando el diagrama de momento curvatura, dadas las cargas nodales externas.

`Viga.flecha_dmc(cargas_nodales_ext)`

Deflexión máxima calculada con el diagrama de momento curvatura.

`Viga.verif_ELU(cargas_nodales_ext)`

Devuelve una variable booleana indicando si verifica o no la viga, en estado límite último, para las cargas nodales externas dadas.

`Viga.verif_ELS(cargas_nodales_ext)`

Devuelve una variable booleana indicando si verifica o no la viga, en estado límite de servicio, para las cargas nodales externas dadas.

# Apéndice B

## Módulos para aprendizaje automático

### B.1. Generación de base de datos

#### B.1.1. Comprobación de secciones

La base de datos para comprobación de secciones se generó con el siguiente código:

---

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Base de datos para comprobación de secciones

Created on Tue Sep 10 19:34:08 2019

@author: Fredy Gabriel Ramírez Villanueva
"""

#Librerías
import numpy as np
import pandas as pd

#Mis módulos
import vigas as vg
import materiales as mt
```

```

#Materiales
aceros = np.array([400, 500])*1e6
hormigones = np.array([12, 16, 20, 25, 30, 35])*1e6

#Geometría
#Redondeo en dimensiones de vigas
step = 0.05

#Lista de valores de la base
b_min = .05
b_max = .40
bases = np.arange(b_min, b_max + step, step)

#Dataframe para almacenar los datos
df = pd.DataFrame(columns=['fyk', 'fck', 'b', 'h', 'Ast',
                          'Asc', 'Mu'])

def base_de_datos(aceros, hormigones, bases):
    fila = 0 #contador de filas de datos
    for fyk in aceros:
        A = mt.Acero(fk=fyk)
        for fck in hormigones:
            H = mt.Hnl(fk=fck)
            for base in bases:
                #Igual a base y no menor a 0,15m
                h_min = max(0.15, base)

                #Igual a 4xbase y no mayor a 1,00
                h_max = min(1.00, 4*base)

                alturas = np.arange(h_min, h_max + step, step)
                for altura in alturas:
                    #Viga provisoria p/ verif. de cuantías
                    v1 = vg.Viga(b=base, h=altura,
                                hormigon=H, acero=A)

                    #ARMADURA MÍNIMA
                    #Un criterio de armadura de montaje mínima
                    if base < 0.10:
                        Amin1 = (8/1000)**2*np.pi/4
                    elif base < 0.20:
                        Amin1 = 2*(8/1000)**2*np.pi/4
                    elif base < 0.25:
                        Amin1 = 2*(10/1000)**2*np.pi/4
                    else:

```

---

```

    Amin1 = 3*(10/1000)**2*np.pi/4

    #Armadura mínima según EC2 (9.1N)
    Amin2 = v1.As_min

    #Armadura de tracción mínima adoptada
    Ast_min_adopt = max(Amin1, Amin2)

    #ARMADURA MÁXIMA
    Asmax = v1.As_max

    paso_arm = Amin1 #Otro criterio
    areas_traccion = np.arange(Ast_min_adopt,
                                Asmax, paso_arm)

    for area in areas_traccion:
        #Armadura de comp. definitiva adoptada
        Asc_adopt = max(Amin1, Amin2,
                        0.30*area)
        v2 = vg.Viga(b=base, h=altura,
                    hormigon=H, acero=A, Ast=area,
                    Asc=Asc_adopt)
        momento = v2.Mu()

        df.loc[filas] = [fyk, fck, base, altura,
                        area, Asc_adopt, momento]

        filas += 1 #siguiente fila de datos
    if filas%100 == 0:
        print("Fila: {}".format(filas))

```

---

### B.1.2. Flechas de vigas simplemente apoyadas

La base de datos de flechas se generó con el siguiente código:

---

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Sep 23 23:19:19 2019

@author: Fredy Gabriel Ramírez Villanueva
"""
import numpy as np
import pandas as pd
import materiales as mt

```

```

import vigas as vg
import utiles as ut

#Base de datos de secciones
dset = pd.read_csv('dset_vigas.csv', index_col=0)

#Criterios
bases = [20, 25, 30] #en cm
longitudes = [3, 4, 5, 6, 7, 8] #en m
hormigones = [16, 20, 25] # en MPa
aceros = [400, 500] #en MPa

def BD_def(longitudes, bases, hormigones, aceros, Nam=4):
    '''
    Devuelve Base de Datos de deformaciones para una viga
    simplemente apoyada con carga uniformemente distribuida
    en toda su longitud.
    Nam: int, mín: 4
        Número Áreas–Momentos. Se tomarán Nam valores del
        gráfico Ast vs Mu para unos parámetros fijos de b, h,
        fyk, fck.
    '''
    df = pd.DataFrame({'fyk':[], 'fck':[], 'b':[], 'h':[],
                      'L':[], 'Ast':[], 'q':[], 'f':[]})
    for L in longitudes:
        alturas = np.array([L/10, L/12, L/14])
        for hprov in alturas:
            #redondeo de las alturas al múltiplo de 0,05 más
            #cercano hacia arriba
            h = int(ut.dea5cm_ex(hprov)*100)
            for b in bases:
                for fck in hormigones:
                    Ho = mt.Hnl(fk=fck*1e6, est_lim='ELS')
                    for fyk in aceros:
                        Ao = mt.Acero(fk=fyk*1e6, est_lim='ELS')

                    #Base de datos de áreas y momentos
                    AM = dset.loc[(dset['b']==b)&
                                  (dset['h']==h)&(dset['fyk']==fyk)&
                                  (dset['fck']==fck)]

                    #Base de datos
                    momentos = np.array(AM['Mu'])
                    if len(momentos) > 0:

```

```

if len(momentos) > 1:
    #Momentos
    Mmax = AM['Mu'].max()
    Mmin = AM['Mu'].min()
    #Valores de momentos para el
    #cálculo de las cargas.
    #Se descarta el 1ro. y último
    moms = np.linspace(Mmin, Mmax,
                        Nam+2)[1:-1]

    areas = np.array(AM['Ast'])

for M in moms:
    #Determ. de Ast para M
    As = ut.yLista(momentos,
                    areas, M)

    #Creación de la viga
    V1 = vg.Viga(b/100, h/100,
                 L, Ho, Ao, est_lim='ELS',
                 Ast=As/10000,
                 Asc=.25*As/10000)

    #Máx. q externa
    qmax = 8*M/L**2*1e3 - V1.pp

    #Desde pp hasta qmax+pp
    cargas=np.linspace(0, qmax,
                       Nam)[1:]

    for q in cargas:
        q_nod =\
        V1.carga_unif_dist(q)
        f =V1.flecha_dmc(
            [q_nod])[0]

        #EC2 7.4.1(4)
        if f <= L/250:
            df= df.append({
                'b':b, 'h':h, 'L':L,
                'Ast':As, 'fck':fck,
                'fyk':fyk, 'q':q/1e3,
                'f':f*1000},
                ignore_index=True)
            print("{} , {} , {} , \
                {:.2f} , {} , {} , \

```

```

        {:.2f},{:.5f}"\
        .format(b,h,L,As,fck,\
        fyk,q/1000,f*1000))
else:
    M = AM['Mu'][0]
    As = AM['Ast'][0]
    #Creación de la viga
    V1 = vg.Viga(b/100,h/100,L,Ho,
    Ao, est_lim='ELS', Ast=As/1e4,
        Asc=.25*As/1e4)

    #q_ext
    qmax = 8*M/L**2*1e3 - V1.pp

    #Desde pp hasta qmax+pp
    cargas=np.linspace(0,qmax,
        Nam)[1:]

    for q in cargas:
        q_nod=V1.carga_unif_dist(q)
        f = V1.flecha_dmc(
            [q_nod])[0]
        if f <= L/250:#EC2 7.4.1(4)
            df= df.append({'b':b,
                'h':h,'L':L,'Ast':As,
                'fck':fck,'fyk':fyk,
                'q':q/1000,'f':f*1000},
                ignore_index=True)
            print("{} , {}, {}, {:.2f}\
            , {}, {}, {:.2f}, {:.5f}"\
            .format(b,h,L,As,fck,\
                fyk,q/1000,f*1000))

df.to_csv('bdd.csv')

BD_def(longitudes, bases, hormigones, aceros, Nam=4)

```

## B.2. Módulo: mielm.py

Implementación del algoritmo de Extreme Learning Machine tal como se indica en el paper “Extreme learning machine: Theory and applications” de Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew (2006). ELM de tipo regresión. Admite  $n$  datos de entrada y  $m$  datos de salida.

Requiere: `numpy` y `pandas`

### B.2.1. Atributos

Para el entrenamiento y verificación por el método de aprendizaje automático extremo se define la clase `ELM` con los siguientes atributos:

`ELM.training_set`

Conjunto de entrenamiento. Tipo: `pandas dataframe`. No requiere normalización previa.

`ELM.testing_set`

Conjunto de verificación. Tipo: `pandas dataframe`. No requiere normalización previa.

`ELM.entradas`

Variables de entrada. Tipo: lista de cadenas (corresponden a columnas de los conjuntos de entrenamiento y verificación)

`ELM.salidas`

Variables de salida. Tipo: lista de cadenas (corresponden a columnas de los conjuntos de entrenamiento y verificación)

`ELM.NN`

Número de neuronas en la capa oculta. Tipo: número entero.

`ELM.semilla`

Semilla para control de las variables aleatorias. Tipo: número entero. Por defecto: 0.

`ELM.tipo_fact`

Tipo de función de activación. Tipo: cadena. Valores posibles: `'sig'` sigmoide; `'sen'` seno, `'gauss'` gaussiana, `'tanh'` tangente hiperbólica, `'arctan'` arctangente, `'isru'` raíz cuadrada inversa, `'silu'` sigmoide ponderado.

`ELM.tol`

Tolerancia para considerar un valor como “exacto” Tipo: número real. Por defecto: 0,05

### B.2.2. Propiedades derivadas

ELM.NE

Tamaño del conjunto de entrenamiento.

ELM.NV

Tamaño del conjunto de verificación.

ELM.NVE

Número de variables de entrada.

ELM.NVS

Número de variables de salida.

ELM.sal\_e

Valores de salida para la fase de entrenamiento.

ELM.sal\_v

Valores de salida para la fase de verificación.

ELM.W

Matriz de pesos con valores aleatorios de -1 a 1.

ELM.b

Vector de bias con valores aleatorios de 0 a 1.

ELM.ent\_min

Mínimos de los valores de entrada.

ELM.ent\_max

Máximos de los valores de entrada.

ELM.ent\_en

Valores de entrada normalizados para la fase de entrenamiento.

ELM.ent\_vn

Valores de entrada normalizados para la fase de verificación.

ELM.H(x)

Matriz de salida de la capa oculta para los valores de entrada  $\mathbf{x}$  (pandas dataframe).

ELM.beta

Matriz de pesos beta.

ELM.exactitud(B, x, t)

Comprobación de exactitud. Devuelve raíz del error medio cuadrático, error medio absoluto y porcentaje de aciertos según la tolerancia adoptada. B: matriz beta, x: valores de entrada, t: valores de salida.

ELM.exactitud\_e(B)

Exactitud en la fase de entrenamiento, dada la matriz beta.

ELM.exactitud\_v(B)

Exactitud en la fase de verificación, dada la matriz beta.

ELM.pred(xp, beta)

Predicción de casos particulares. Devuelve ndarray, dado el caso particular xp: Pandas dataframe; y la matriz beta B.

# REFERENCIAS BIBLIOGRÁFICAS

## Libros

- Bertsekas, Dimitri P y John N Tsitsiklis (2002). *Introduction to probability*. Vol. 1. Athena Scientific Belmont, MA.
- Branson, Dan Earle y Gene Alan Metz (1963). *Instantaneous and time-dependent deflections of simple and continuous reinforced concrete beams*. Department of Civil Engineering y Auburn Research Foundation.
- Calavera, José (1985). *Proyecto y cálculo de estructuras de hormigón armado para edificios: p.: il., gráf.* 2da. Ed. Intemac.
- Feynman, Richard Phillips (1964). *Probability & Uncertainty: The Quantum Mechanical View of Nature*. Education Development Center.
- Fox, Richard L (1971). *Optimization methods for engineering design*. Ed. por Howard W. Emmons y Bernard Burdiansky. Addison-Wesley Pub. Co.
- Géron, Aurélien (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc.
- Guttag, John V (2013). *Introduction to computation and programming using Python*. Mit Press.
- Heaton, Jeff (2008). *Introduction to neural networks with Java*. Heaton Research, Inc.
- Hibbeler, Russel C. (2012). *Análisis estructural*. 8va. Pearson Educación.
- Jiménez Montoya, Pedro y col. (2009). *Hormigón armado*. 15ta. Gustavo Gili.
- Kassimali, Aslam (2012). *Matrix Analysis of Structures*. Cengage Learning.
- Kletenik, D (1979). *Problemas de geometría analítica*. MIR.
- Lauritzen, Niels (2013). *Undergraduate Convexity: From Fourier And Motzkin To Kuhn And Tucker*. [ World Scientific. URL: <http://search.ebscohost.com/>

- `login.aspx?direct=true&db=nlebk&AN=575386&authtype=sso&custid=ns021105&lang=es&site=eds-live&custid=ns021105&authtype=ip,sso.`
- Miller, Irwin R., John E. Freund y Richard Johnson (1992). *Probabilidad y estadística para ingenieros*. 4ta. Prentice Hall.
- Nocedal, Jorge y Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media.
- Oliphant, Travis E (2006). *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.
- Raschka, Sebastian (2015). *Python Machine Learning*. Community Experience Distilled. Packt Publishing. ISBN: 9781783555130. URL: `http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1071004&authtype=sso&custid=ns021105&lang=es&site=eds-live&custid=ns021105&authtype=ip,sso.`
- Strang, Gilbert (2006). *Linear Algebra and Its Application*. Thomson Corporation.
- Summerfield, Mark (2010). *Programación: Python 3*. Ed. por Ediciones ANAYA Multimedia. Addison-Wesley Professional.
- Suykens, Johan A. K. (2002). *Least Squares Support Vector Machines*. World Scientific. ISBN: 9789812381514. URL: `http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=210682&authtype=sso&custid=ns021105&lang=es&site=eds-live&custid=ns021105&authtype=ip,sso.`
- Van Rossum, Guido y Fred L Drake Jr (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

## Artículos

- Aggarwal, Yogesh (2007). «Modeling of reinforcement in concrete beams using machine learning tools». En: *World Academy of Science, Engineering and Technology* 32, págs. 130-131.
- Caicedo, Alexander y col. (2019). «Functional form estimation using oblique projection matrices for LS-SVM regression models.» En: *PLoS ONE* 14.6, págs. 1-21. ISSN: 19326203. URL: `http://search.ebscohost.com/login.aspx?direct=true&db=fsr&AN=136872638&authtype=sso&custid=ns021105&lang=es&site=eds-live&custid=ns021105&authtype=ip,sso.`
- Cevik, Abdulkadir y col. (2014). «Support vector machines in structural engineering: a review.» En: *JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT* 21.3, págs. 261-281. ISSN: 13923730. URL: `http://search.ebscohost.com/login.aspx?direct=true&db=edswsc&AN=000350032200001&authtype=sso&custid=ns021105&lang=es&site=eds-live&custid=ns021105&authtype=ip,sso.`

- Cybenko, George (1989). «Approximation by superpositions of a sigmoidal function». En: *Mathematics of control, signals and systems* 2.4, págs. 303-314.
- Grané Chávez, Aurea (2010). «Análisis de Componentes Principales». En: *Publicaciones Material Docente - Universidad Carlos III de Madrid*.
- Hastie, Trevor y col. (2005). «The elements of statistical learning: data mining, inference and prediction». En: *The Mathematical Intelligencer* 27.2, págs. 83-85.
- Huang, Guang-Bin (nov. de 2006). «Hands-on Workshop on Machine Learning for BioMedical Informatics». En:
- Huang, Guang-Bin, Chen Lei y Chee-Kheong Siew (jul. de 2006). «Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes». En: *IEEE TRANSACTIONS ON NEURAL NETWORKS* 17.4, págs. 879-892.
- Hunter, J. D. (2007). «Matplotlib: A 2D graphics environment». En: *Computing in Science & Engineering* 9.3, págs. 90-95. DOI: 10.1109/MCSE.2007.55.
- Imam, Ashhad y Zaman Abbas Kazmi (2017). «Modified regression and ANN model for load carrying capacity of corroded reinforced concrete beam.» En: *AIMS Materials Science* 4.5, pág. 1140. ISSN: 23720484. URL: <http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=127033037&authtype=sso&custid=ns021105&lang=es&site=eds-live&custid=ns021105&authtype=ip,sso>.
- Kaczmarek, Mateusz y Agnieszka Szymańska (2016). «Application of artificial neural networks to predict the deflections of reinforced concrete beams». En:
- Kluyver, Thomas y col. (2016). «Jupyter Notebooks – a publishing format for reproducible computational workflows». En: ed. por F. Loizides y B. Schmidt, págs. 87-90.
- Kmieciak, P. y M. Kaminski (2011). «Modelling of reinforced concrete structures and composite structures with concrete strength degradation taken into consideration». En: *Archives of civil and mechanical engineering* XI.3, págs. 623-636.
- Kwak, Hyo-Gyoung y Sun-Pil Kim (2002). «Nonlinear analysis of RC beams based on moment–curvature relation». En: *Computers & structures* 80.7, págs. 615-628.
- McKinney, Wes y col. (2010). «Data structures for statistical computing in python». En: 445, págs. 51-56.
- Meurer, Aaron y col. (2017). «SymPy: symbolic computing in Python». En: *PeerJ Computer Science* 3, e103.
- Mitchell, Tom M y col. (1997). «Machine learning». En: *Burr Ridge, IL: McGraw Hill* 45.37, págs. 870-877.
- Pedregosa, F. y col. (2011). «Scikit-learn: Machine Learning in Python ». En: *Journal of Machine Learning Research* 12, págs. 2825-2830.

- Pérez, Fernando y Brian E. Granger (mayo de 2007). «IPython: a System for Interactive Scientific Computing». En: *Computing in Science and Engineering* 9.3, págs. 21-29. ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.53. URL: <https://ipython.org>.
- Preetham S Ravi Kumar H, Prema Kumar W P y Shivaraj M (2015). «Prediction of deflection of reinforced concrete beams using machine learning tools». En: *International Journal of Engineering Research & Technology* 04, págs. 883-886.
- Rafiei, Mohammad H. y col. (2016). «Neural Network, Machine Learning, and Evolutionary Approaches for Concrete Material Characterization.» En: *ACI Materials Journal* 113.6, pág. 781. ISSN: 0889325X. URL: <http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=120163748&authtype=sso&custid=ns021105&lang=es&site=eds-live&custid=ns021105&authtype=ip,sso>.
- Samuel, Arthur (1959). «Some Studies in Machine Learning Using the Game of Checkers.» En: *IBM Journal of Research and Development* 3.3, págs. 210-29.
- Santos, Ana y col. (2014). «Análise não linear de vigas de concreto armado considerando a fissuração». Portugués. En:
- Wang, Taijun y Thomas TC Hsu (2001). «Nonlinear finite element analysis of concrete structures using new constitutive models». En: *Computers & Structures* 79.32, págs. 2781-2791.

## Normas

- UNE-EN:1992 (abr. de 2004). *Eurocode 2: Design of concrete structures - Part 1-1: General rules and rules for buildings*. Standard. Brussels: European Committee for Standardization.
- UNE-EN:1990 (abr. de 2002). *Eurocódigos. Bases de cálculo de estructuras*. Standard. Bruxelles: Comité Europeo de Normalización.
- UNE-EN:1991 (ago. de 2002). *Eurocódigos. Acciones en estructuras*. Standard. Comité Europeo de Normalización.

## Tesis

- Cardozo López, Sergio Daniel (2009). «Otimização de placas e cascas de materiais compósitos, utilizando algoritmos genéticos, redes neurais e elementos finitos».
- Gomes Pedrosa, Thais (2015). «Otimização estrutural sob incertezas considerando consecuencias de falha». Portugués. Escola de Engenharia de São Carlos-Universidade de São Paulo.

- Nina, Thais Coimbra (2006). «Otimização de seções transversais de concreto armado: aplicação a pórticos planos». Português. Escola de Engenharia de São Carlos-Universidade de São Paulo.
- Nogueira, Caio Gorla (2005). «Un modelo de confiabilidade e de otimização aplicado as estruturas de barras de concreto armado». Português.
- Nogueira, Caio Gorla (2010). «Desenvolvimento de modelos mecánicos, de confiabilidade e de otimização para aplicação em estruturas de concreto armado». Português.
- Payá Zaforteza, Ignacio Javier (2009). «Optimización heurística de pórticos de edificación de hormigón armado».
- Soares, Rodrigo de Carvalho (2001). «Um estudo sobre modelos mecânico-probabilísticos para pórticos de concreto armado». Português. Escola de Engenharia de São Carlos-Universidade de São Paulo.
- Velame Branco, André Luís Lima (2002). «Análise não linear de pórticos planos, considerando os efeitos do cisalhamento no cálculo de esforços e deslocamentos». Português. Escola de Engenharia de São Carlos-Universidade de São Paulo.

## Internet

- Abu-Mostafa, Yaser (abr. de 2012). *Lecture 01 - The Learning Problem*. URL: <https://www.youtube.com/watch?v=mbyG85GZ0PI> (visitado ).
- Arroyo, Juan Carlos (2016). *¿El cálculo de flechas es de fiar?* Estructurando. URL: <http://estructurando.net/2016/12/12/el-calculo-de-flechas-es-de-fiar/> (visitado ).
- BIPM (1960). *Système International d'Unités*. Bureau International des Poids et Mesures. URL: <https://www.bipm.org/en/measurement-units/> (visitado 20-09-2019).
- Biswas, P.K. (2014). *Mod-01 Lec-29 Support Vector Machine*. URL: <https://www.youtube.com/watch?v=SRVswRH5Q7E> (visitado ).
- CEN (2019). *Eurocodes. Building the future*. European Commission. URL: <https://eurocodes.jrc.ec.europa.eu> (visitado ).
- Chang, Chih-Chung y Chih-Jen Lin (2003). *LIBSVM - A Library for Support Vector Machines*. National Taiwan University. URL: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/> (visitado ).
- Frontline-Systems (2019). *Training an Artificial Neural Network - Intro*. URL: <https://www.solver.com/training-artificial-neural-network-intro> (visitado 21-09-2019).

- Grimson, Eric (2016). *Introduction to Machine Learning*. URL: <https://www.youtube.com/watch?v=h0e2HAPTGF4> (visitado ).
- Huang, Guan-Bin (2013). *Extreme Learning Machines (ELM): Filling the Gap between Frank Rosenblatt's Dream and John von Neumann's Puzzle*. URL: <https://www.ntu.edu.sg/home/egbhuang/> (visitado 25-08-2019).
- Jones, Eric, Travis Oliphant, Pearu Peterson y col. (2001). *SciPy: Open source scientific tools for Python*. [Online]. URL: <http://www.scipy.org/> (visitado 13-09-2019).
- Maiti, J. (2014). *Mod-01 Lec-30 Principal Component Analysis (PCA)*. IIT Kharagpur India. URL: <https://www.youtube.com/watch?v=hkCT-6KJAK0&frags=pl%2Cwn> (visitado ).
- McGonagle, John y col. (2019). *Backpropagation*. URL: <https://brilliant.org/wiki/backpropagation/> (visitado ).
- Raybaut, Pierre (2009). *Scientific Python Development Environment (SPYDER)*. URL: <https://www.spyder-ide.org> (visitado 20-09-2019).
- Rossum, Guido van, Barry Warsaw y Nick Coghlan (ago. de 2013). *PEP 8 – Style Guide for Python Code*. Python. URL: <https://www.python.org/dev/peps/pep-0008/> (visitado 02-09-2019).
- TIOBE (2019). *TIOBE Index for August 2019*. TIOBE. The software quality company. URL: <https://www.tiobe.com/tiobe-index/> (visitado 31-08-2019).
- Waskom, Michael y col. (sep. de 2017). *mwaskom/seaborn: v0.8.1 (September 2017)*. URL: <https://doi.org/10.5281/zenodo.883859> (visitado 20-09-2019).