

Introducción a la Inteligencia Artificial

13 de diciembre de 2022

Aprendizaje Automático



UNIVERSIDAD
DE GRANADA



Presentación

Juan Gómez Romero

Profesor Titular de Universidad
Departamento de Ciencias de la Computación e Inteligencia Artificial

E-mail: jgomez@decsai.ugr.es
Web: <http://decsai.ugr.es/~jgomez>



Miguel Molina Solana

Profesor Contratado Doctor
Departamento de Ciencias de la Computación e Inteligencia Artificial

E-mail: miguelmolina@ugr.es
Web: <https://miguelmolina.me>



Bibliografía

Libros

Aprendizaje no supervisado

J. Han, M. Kamber, J. Pei (2011) Data Mining: Concepts and Techniques. Morgan Kaufmann.

H.I. Rhys (2020) Machine Learning with R, the tidyverse, and mlr. Manning.

G. Gan, C. Ma, J. Wu. (2007) Data Clustering: Theory, Algorithms and Applications. SIAM.

Aprendizaje supervisado

T. Hastie , R. Tibshirani , J. Friedman (2009) The Elements of Statistical Learning. Springer.

S. García, J. Luengo, F. Herrera (2015) Data Preprocessing in Data Mining. Springer.

Aprendizaje profundo

F. Chollet, J. Allaire (2018) Deep Learning with R. Manning.

F. Berzal (2018) Redes Neuronales & Deep Learning.

I. Goodfellow, Y. Bengio, A. Courville (2016) Deep Learning. MIT Press.

F. Chollet (2022) Deep Learning with Python (2nd edition). Manning.

Índice

Bloque I 1.5 horas

Aprendizaje no supervisado

Reglas de asociación

Agrupación

Aprendizaje supervisado

Árboles de clasificación

Random forest

Boosting

SVM

Bloque II 1.5 horas

Aprendizaje profundo

Fundamentos de redes neuronales

Redes convolutivas

Redes recurrentes

Redes para grafos

Índice

Aprendizaje no supervisado

Reglas de asociación

Agrupación

Aprendizaje automático

Ejemplo

Datos: Pasajeros del Titanic



titanic.Rmd

Variables o características (9):

- survival = {0, 1}
- pclass = {1st, 2nd, 3rd}
- sex: sexo
- age: edad
- sibsp: número de parientes (hermano/a, hermanastro/a) / cónyuge (esposo/a) a bordo
- parch: número de padres (madre/padre) / hijos a bordo (hijo/a, hijastro/a)
- ticket: número de ticket
- fare: precio del ticket
- cabin: número del camarote
- embarked: puerto de embarque

Aprendizaje automático

Ejemplo

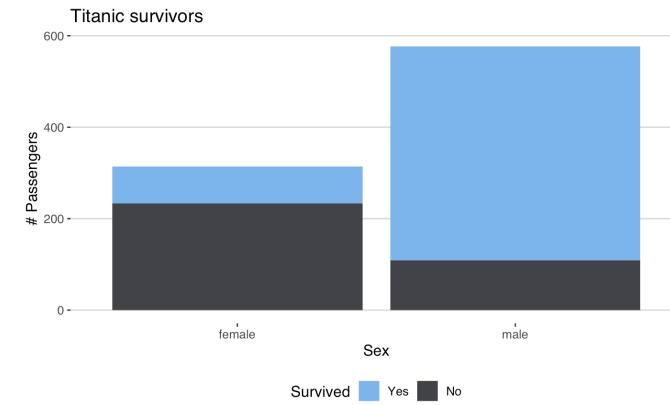
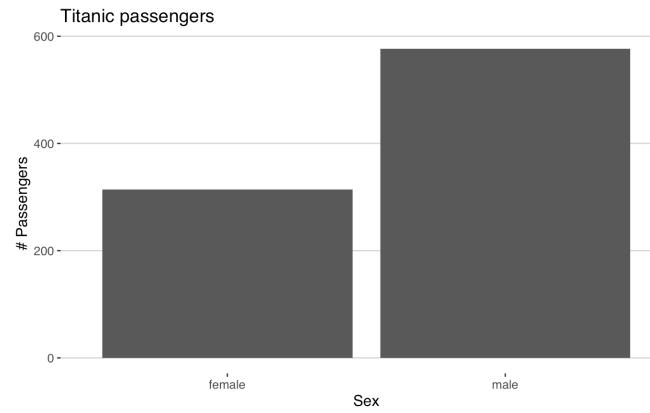
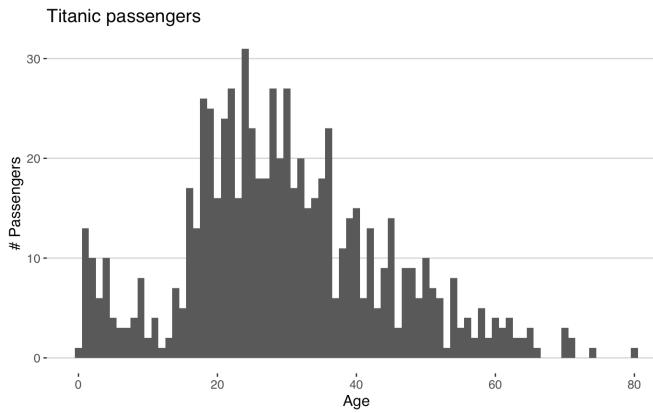
Datos: Pasajeros del Titanic

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275		S
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125		Q
18	1	2	Williams, Mr. Charles Eugene	male		0	0	244373	13		S
19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31	1	0	345763	18		S
20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7.225		C

Aprendizaje automático

Ejemplo

Datos: Pasajeros del Titanic



Aprendizaje automático

Introducción

Definición

Capacidad de aprender a resolver problemas para los que [los ordenadores] no han sido programados (A. Samuel)

Encontrar de patrones en conjuntos de datos que se pueden utilizar para deducir información no conocida explícitamente

Ejemplo: Robot al que le presentamos fotos de perros y gatos, indicándole qué animal hay en cada una de ellas. El robot puede aprender a distinguir perros y gatos y, ante una foto desconocida, inferir de qué se trata [← aprendizaje supervisado + clasificación]

Aprendizaje automático moderno: las características que componen estos patrones no se indican a priori

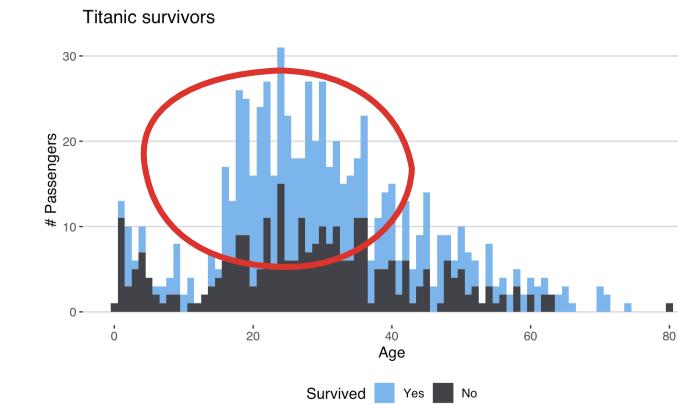
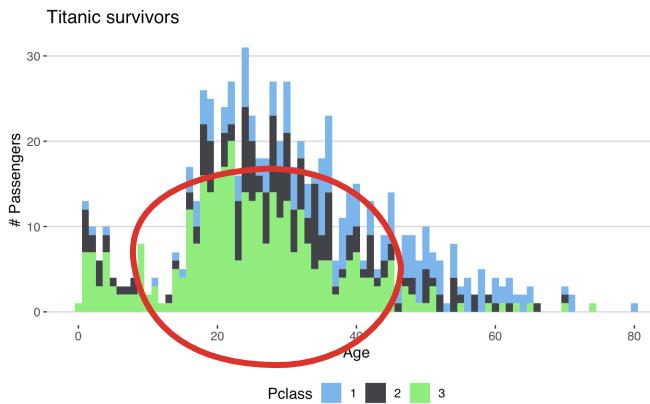
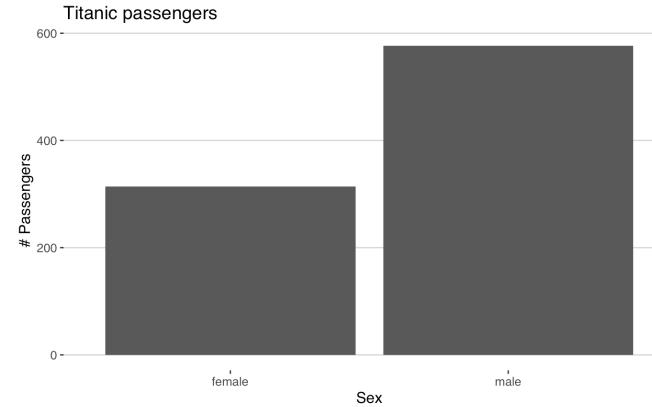
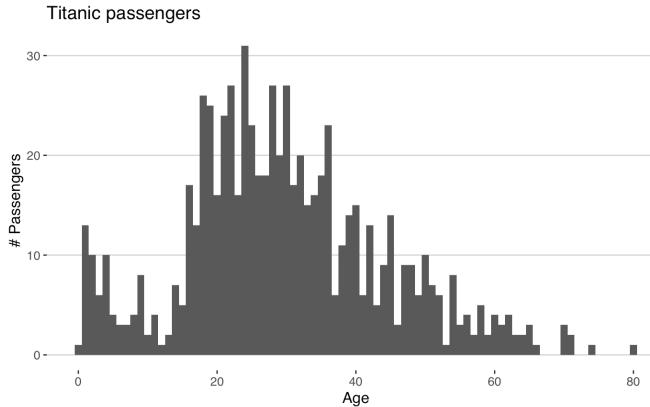
Aprendizaje automático

Ejemplo

Datos: Pasajeros del Titanic



titanic.Rmd



Aprendizaje automático

Introducción

Aprendizaje no-supervisado

Se construyen descripciones, hipótesis o teorías a partir de un conjunto de hechos u observaciones, sin que exista información adicional acerca de cómo deberían clasificarse los ejemplos del conjunto de entrenamiento.

Será el método de aprendizaje no supervisado el que decida cómo han de agruparse los datos del conjunto de entrenamiento o qué tipo de patrones son más interesantes dentro del conjunto de entrenamiento.

Aprendizaje supervisado

Los ejemplos de entrenamiento van acompañados de la salida correcta que el sistema debería ser capaz de reproducir.

Una vez entrenado el modelo, lo verdaderamente importante es que sea capaz de generalizar correctamente [...] > el modelo proporcione salidas adecuadas para datos de entrada diferentes a los datos utilizados durante su entrenamiento.

Índice

Aprendizaje no supervisado

Reglas de asociación

Agrupación

Aprendizaje no supervisado

Reglas de asociación

Búsqueda de dependencias en datos

Representan patrones de comportamiento en los datos en términos de aparición conjunta de valores de manera frecuente

Es una técnica de aprendizaje no supervisado

No se dispone de clases predefinidas ni de ejemplo etiquetados; es decir, no se conocen las asociaciones para ningún subconjunto de individuos

Suele realizarle en las primeras etapas del proceso de análisis de datos

Las reglas sirven para resumir los datos, de forma que se pueden utilizar las reglas como representación colectiva de los individuos

Aprendizaje no supervisado

Reglas de asociación

si A entonces B $A \Rightarrow B$

A : antecedente

B : consecuente

Formalmente

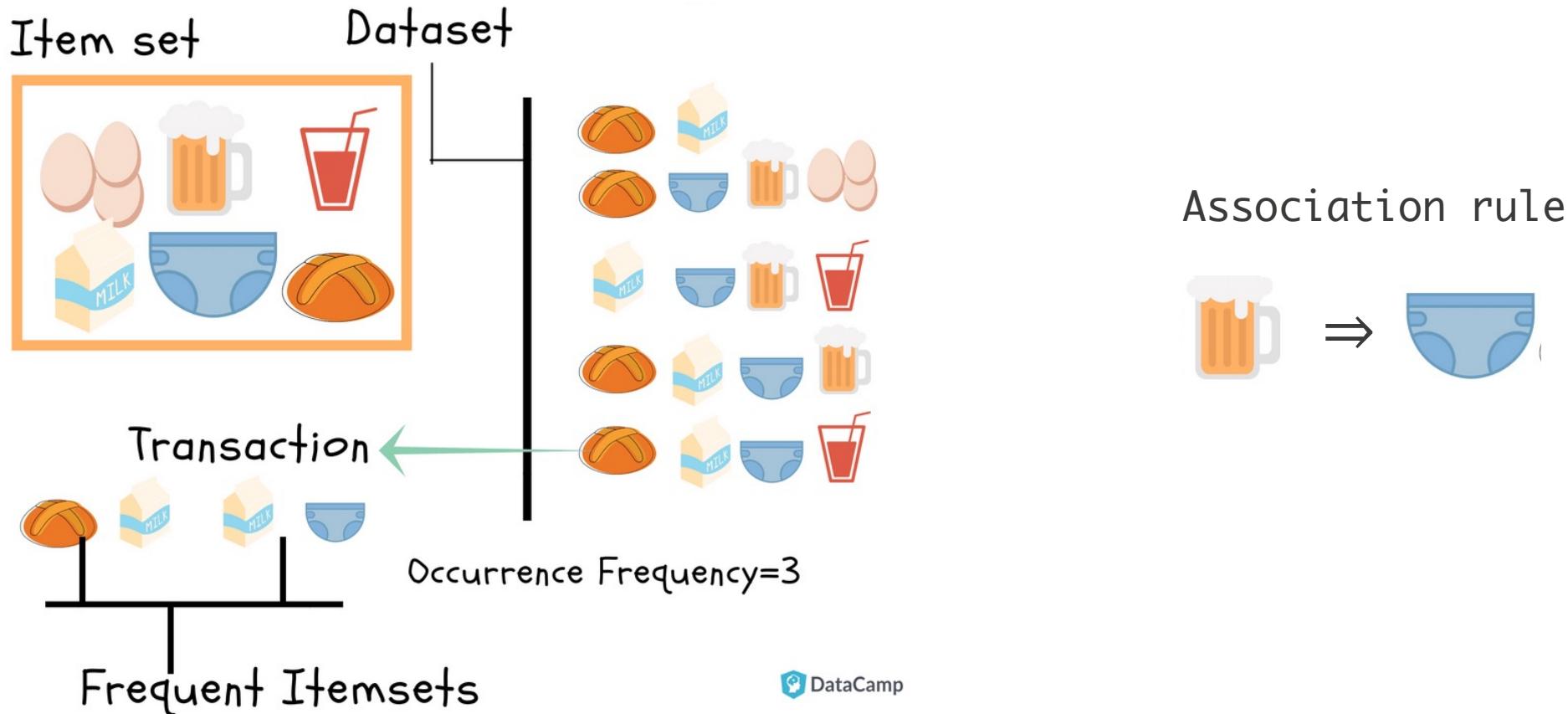
I un conjunto de ítems

T un conjunto de transacciones con ítems en I

La regla de asociación $A \Rightarrow B$ con $A, B \subseteq I$, $A, B \neq \emptyset$ y $A \cap B = \emptyset$ significa que cada transacción de T que contiene a A también contiene a B

Aprendizaje no supervisado

Reglas de asociación



Aprendizaje no supervisado

Reglas de asociación

Soporte: porcentaje de asociaciones que la regla predice correctamente [probabilidad]

Sobre un conjunto de ítems $X \subseteq I$: frecuencia con la que ocurre en la base de datos

$$\text{Soporte}(X) = \frac{\text{Nº de ocurrencias de } X}{\text{Nº total de transacciones en la BD}}$$

Considerando una regla: número de transacciones que contienen la unión de los ítems de la regla

$$\text{Soporte}(A \Rightarrow C) = \text{Soporte}(A \cup C) = \frac{\text{Nº de ocurrencias de } A \cup C}{\text{Nº total de transacciones en la BD}}$$

Confianza: porcentaje de veces que la regla se cumple cuando se puede aplicar [probabilidad condicionada]

$$\text{Confianza}(A \Rightarrow C) = \frac{\text{Soporte}(A \cup C)}{\text{Soporte}(A)}$$

Aprendizaje no supervisado

Reglas de asociación

Algoritmo Apriori

Fuerza bruta 

1. Obtener todas las combinaciones de ítems y crear reglas
2. Calcular el soporte y la confianza de cada regla
3. Eliminar las reglas que no superan los umbrales de soporte y confianza

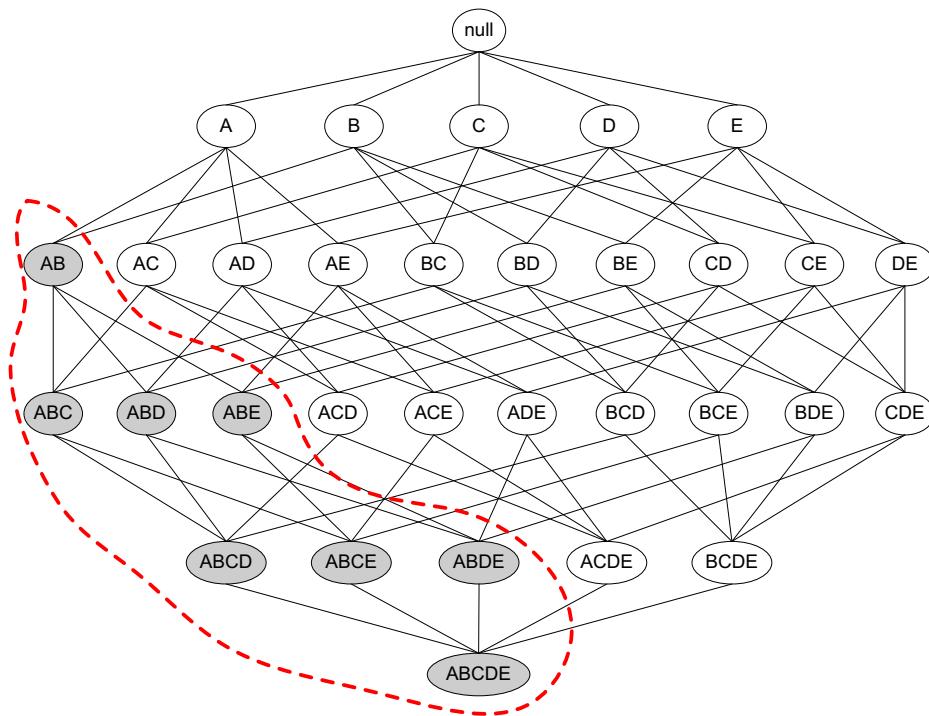
Solución en dos etapas

1. Generar itemsets frecuentes (por encima del umbral de soporte)
2. Generar reglas de asociación con los itemsets del paso 1 (por encima del umbral de confianza)

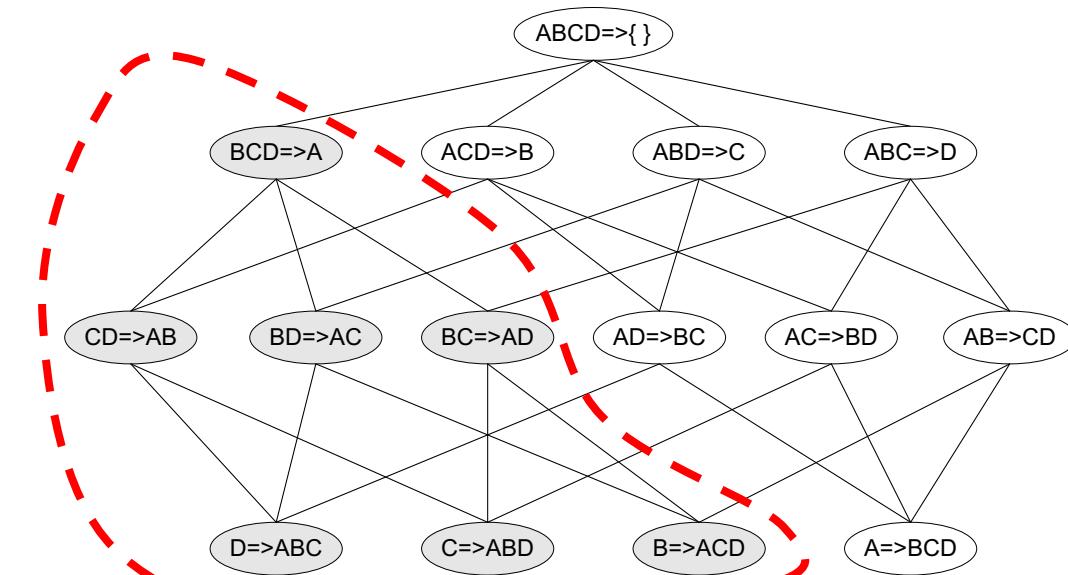
Aprendizaje no supervisado

Reglas de asociación

Algoritmo Apriori



Cálculo de itemsets con alto soporte

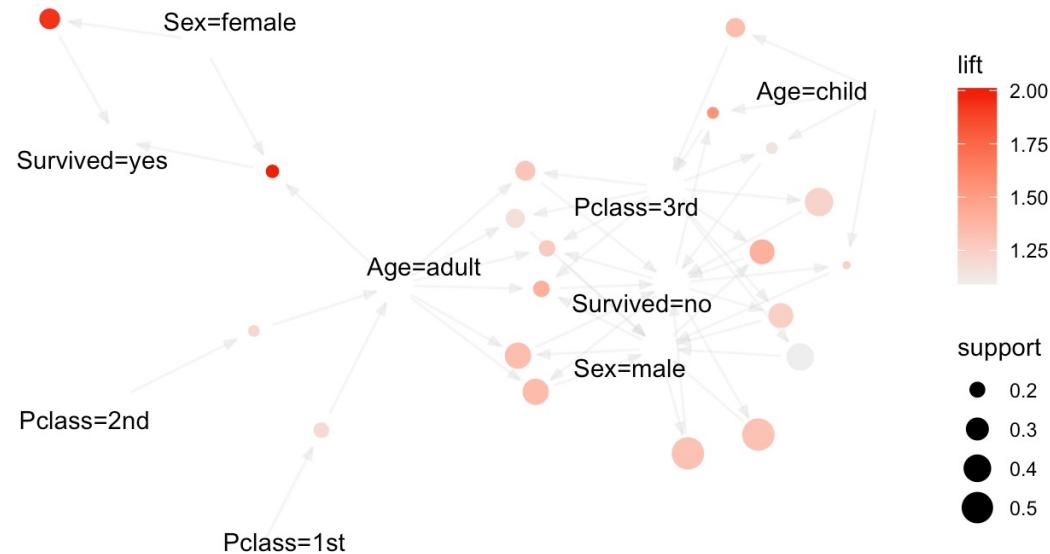


Generación de reglas con alta confianza

Aprendizaje no supervisado

Reglas de asociación

Algoritmo Apriori



```
minSup <- 0.01
minConf <- 0.70

reglas <- apriori(datost,
                    parameter = list(support = minSup,
                                      confidence = minConf,
                                      target = "rules",
                                      minlen = 2))

plot(head(reglas, n=20, by="support"), method = "graph")
```

Índice

Aprendizaje no supervisado

Reglas de asociación

Agrupación

Aprendizaje no supervisado

Agrupación

Búsqueda de agrupaciones en datos

Proceso de agrupar un conjunto de objetos descritos mediante propiedades en grupos (o clústeres), de forma que un clúster contiene objetos similares entre sí y diferentes a los de otros clústeres

Es una técnica de aprendizaje no supervisado

No se dispone de clases predefinidas ni de ejemplo etiquetados; es decir, no se conocen las agrupaciones para ningún subconjunto de individuos

Suele realizarle en las primeras etapas del proceso de análisis de datos

Los clústeres sirven para resumir los datos, de forma que se pueden utilizar las agrupaciones como representación colectiva de los individuos

Aprendizaje no supervisado

Agrupación

Elemento fundamental, ya que cuantifica si dos individuos son similares o no

Aplicación en la generación de los *clústeres* o en la evaluación de la calidad de los *clústeres*

Distancia Euclídea

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

Distancia de Minkowski

$$d(x, y) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{\frac{1}{r}}$$

Correlación de Pearson

$$\rho_{X,Y} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X) \text{Var}(Y)}}$$

Distancia de Mahalanobis

$$d(x, y) = \sqrt{(x - y) \boldsymbol{\sigma}^{-1} (x - y)^T}$$

Distancia coseno

$$\cos(x, y) = \frac{(x \bullet y)}{\|x\| \|y\|}$$

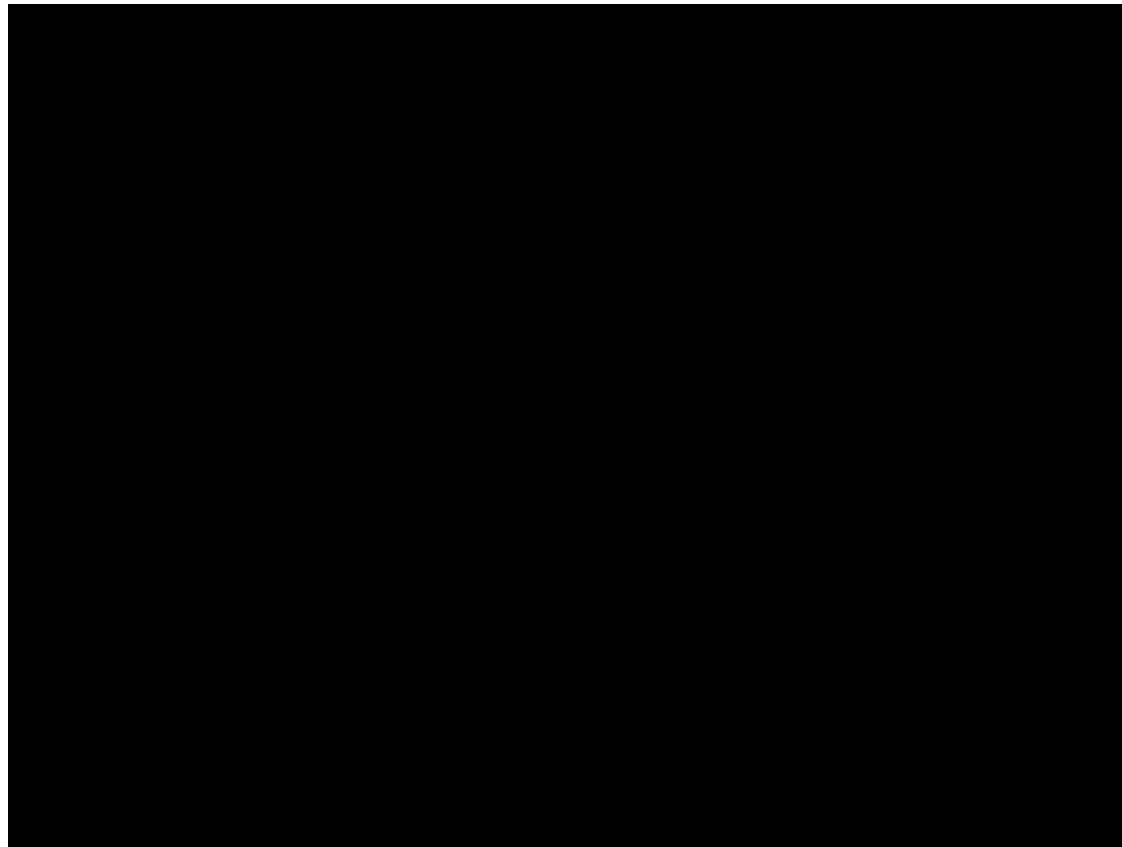
Distancia de Jaccard

$$d_J(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}.$$

Aprendizaje no supervisado

Agrupación

Algoritmo k-means



Aproximación básica (algoritmo de Lloyd)

Entrada: k (número de clústeres), n objetos

Procedimiento:

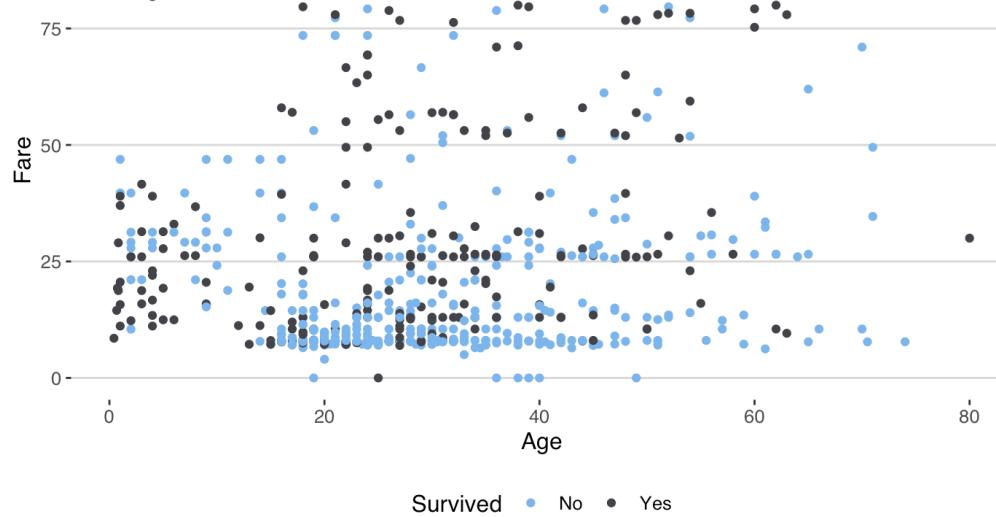
1. elegir aleatoriamente los centros de los clústeres
2. repetir mientras haya cambios:
 - 2.1 (re)asignar cada objeto al clúster con centro más cercano
 - 2.2. recalcular los centros como el punto medio de cada clúster

<https://youtu.be/5I3Ei69I40s>

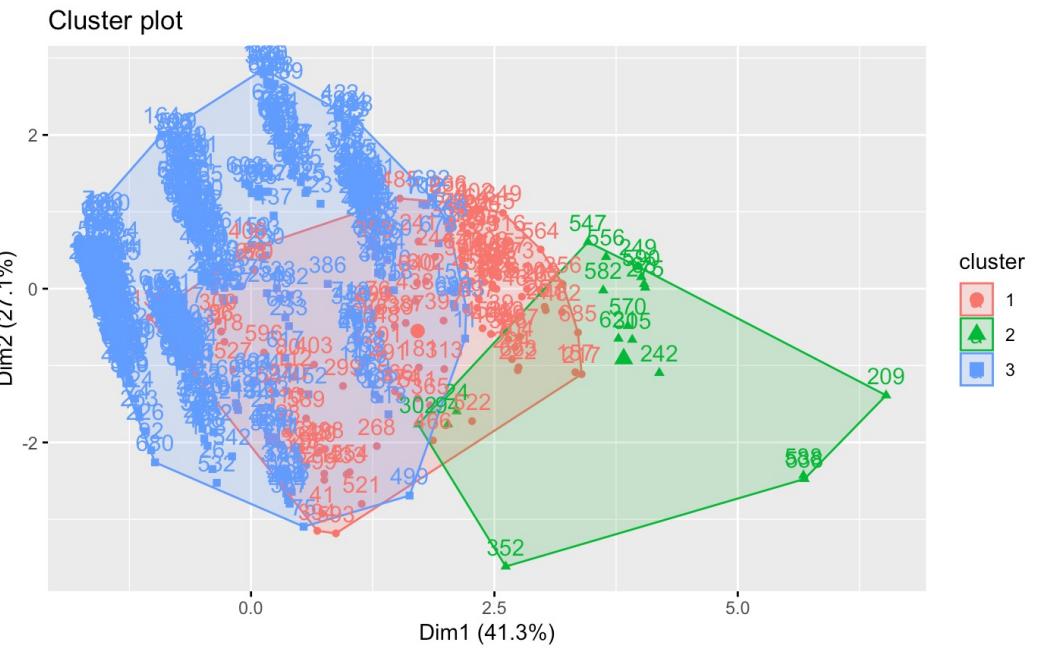
Aprendizaje no supervisado

Agrupación

Algoritmo k-means



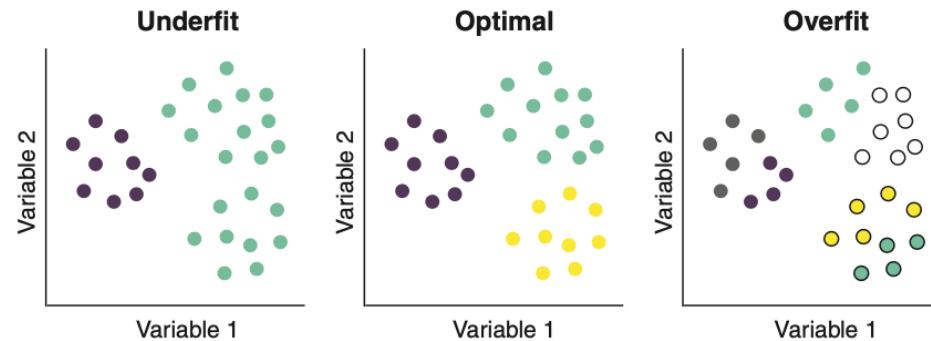
```
clustering <- kmeans(clusterdata, centers = 3,  
                      algorithm = "Lloyd",  
                      iter.max = 100)  
  
library(factoextra)  
fviz_cluster(clustering, data = clusterdata)
```



Aprendizaje no supervisado

Agrupación

Para seleccionar k necesitamos discriminar entre agrupaciones buenas y no tan buenas



H.I. Rhys. Machine Learning with R, the tidyverse, and mlr. Manning, 2020.

Medidas de calidad del clustering

- Comparar con un *ground truth*, donde se conozcan las agrupaciones ideales [métodos *extrínsecos* > aprendizaje supervisado]
- Número de elementos, densidad de elementos, distancia entre las agrupaciones, etc. [métodos *intrínsecos*]
 - Silueta: cómo de lejano es un objeto de un clúster a los objetos de los otros clústeres
 - Davies-Boulding: "separabilidad media" de un clúster frente a la del clúster más cercano
 - Dunn: ratio entre la distancia mínima entre puntos de diferentes clústeres y el diámetro máximo de los clústeres

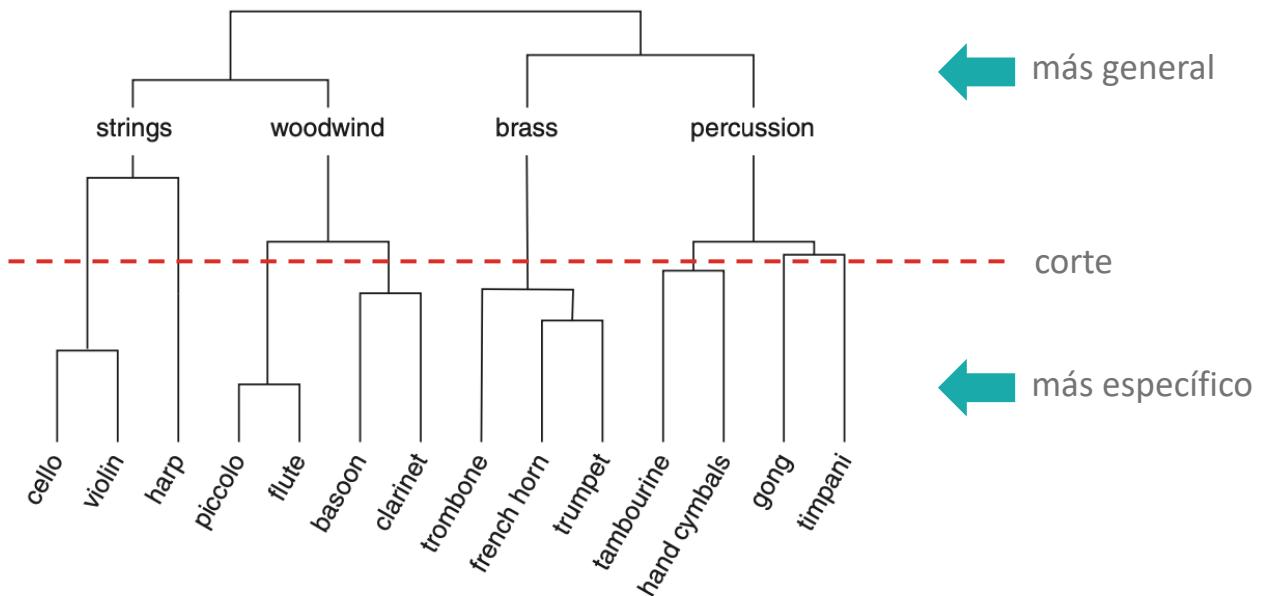
Aprendizaje no supervisado

Agrupación

En muchos problemas no es apropiado agrupar los objetos en divisiones “planas”: ítems de una tienda virtual, clientes de un supermercado, instrumentos de una banda, etc. <<clases y subclases

Representación mediante **dendrograma**

- Woodwinds
 - Reeded
 - Clarinet
 - Bassoon
 - Non-reeded
 - Flute
 - Piccolo
- Strings
 - Plucked
 - Harp
 - Bowed
 - Violin
 - Cello



Aprendizaje no supervisado

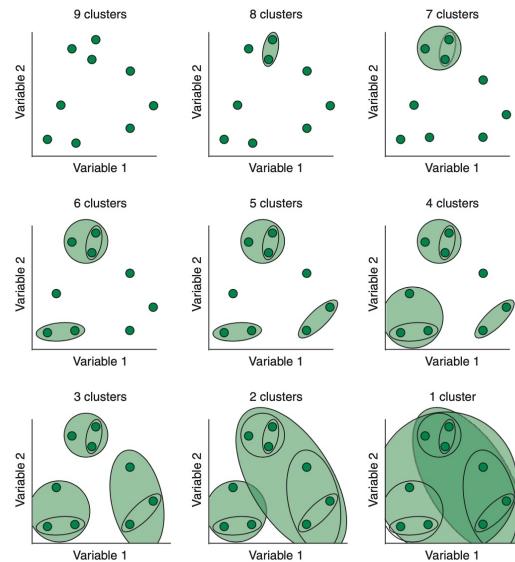
Agrupación

Agglomerative clustering

Comienza con clústeres muy pequeños y va fusionando hasta llegar a una división adecuada

Cálculo: iterativamente, combina los dos clústeres más cercanos en uno solo

El criterio de parada puede establecerse de diferentes formas (1 solo clúster)



Agglomerative clustering

Agglomerative: Single linkage

1. Initialize each point to be its own cluster

Machine Learning: Clustering & Retrieval
Universidad de Washington
4.7 (2244 calificaciones) | 86 mil estudiantes inscritos
Curso 4 de 4 en Aprendizaje Automático Programa Especializado

Inscríbete gratis

Índice

Aprendizaje supervisado

Árboles de clasificación

Random forest

SVM

Aprendizaje supervisado

Proceso

Conjunto de entrenamiento

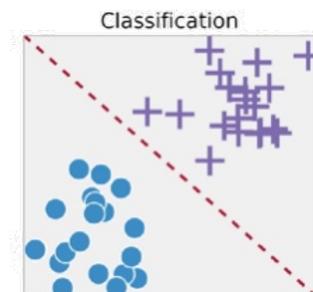
Survived	pclass	sex	...

técnica de clasificación
automática

Conjunto de *test*

pclass	sex	...

Modelo de clasificación



Survived

?

Aprendizaje supervisado

Proceso

Salidas disponibles

Clase	V ₁	...	V _n

Clase	V ₁	...	V _n

Error de entrenamiento
Modelo es capaz de separar las clases

Salidas no disponibles

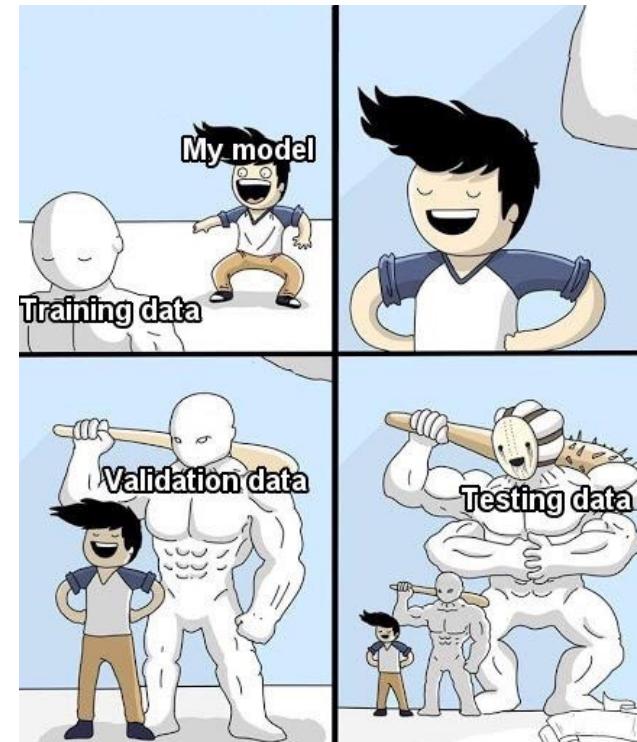
V ₁	...	V _n

Error de test

El modelo funciona con datos no conocidos
No se conoce la clase o se necesita un trabajo extra para calcular una métrica

Clase	V ₁	...	V _n

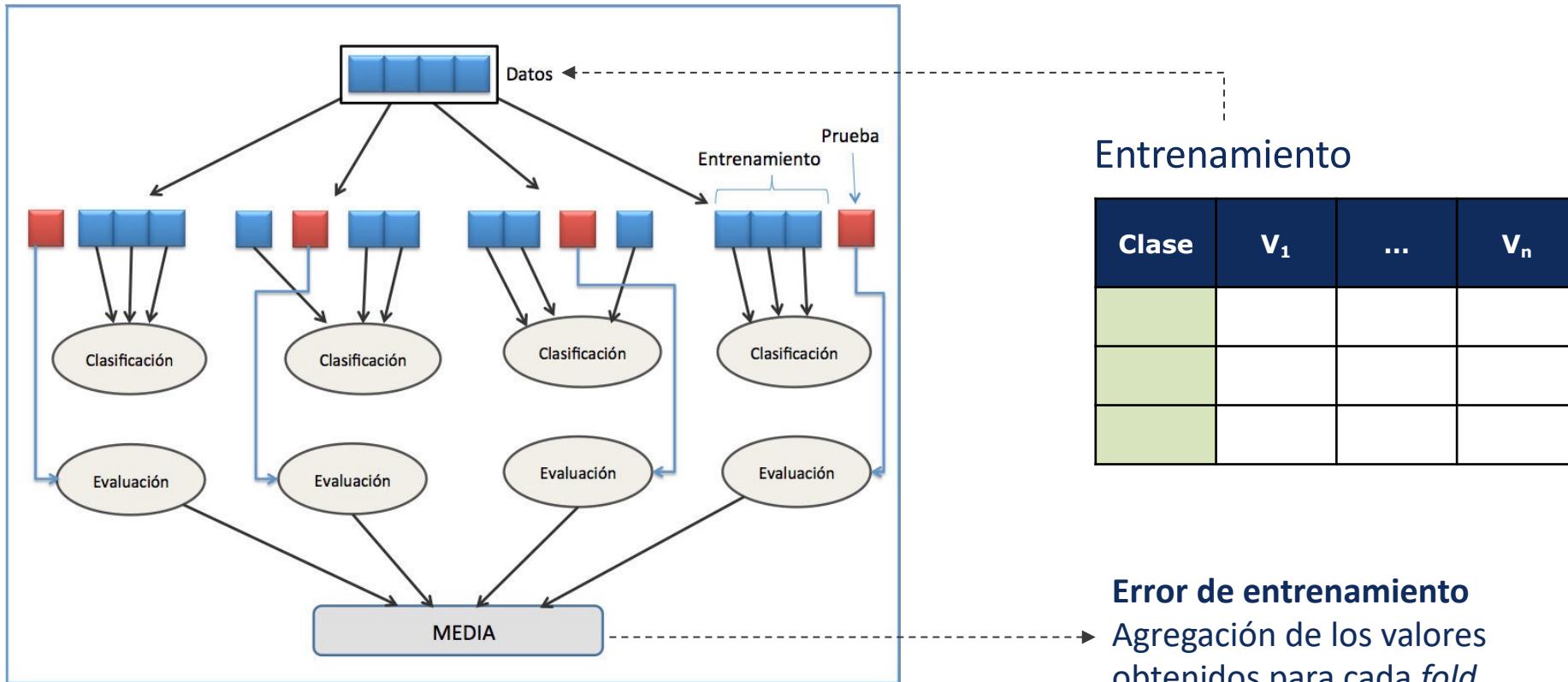
Error de validación
El modelo es capaz de generalizar



Aprendizaje supervisado

Proceso

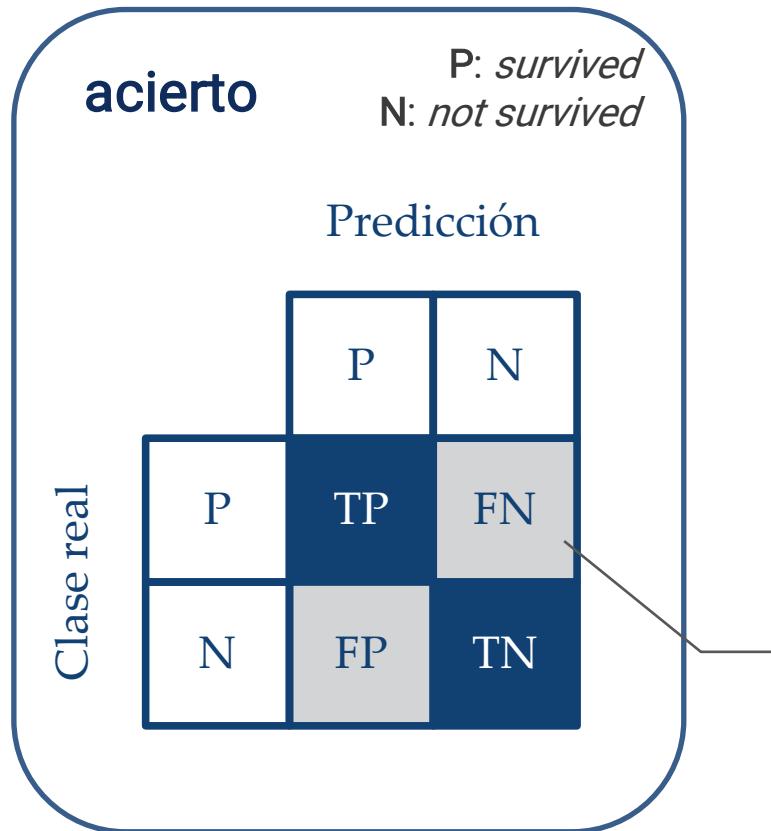
Validación cruzada (*cross validation*)



De Joan.domenech91 - Trabajo propio, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=17617674>

Aprendizaje supervisado

Métricas de clasificación



$$accuracy = \text{acíerto} = \frac{TP + TN}{n} = \frac{\text{aciertos}}{n}$$

$$error = \frac{FP + FN}{n} = \frac{\text{errores}}{n}$$

Matriz de contingencia

FN: *false negative*
Predicción: not survived
Realidad: survived

Aprendizaje supervisado

Métricas de clasificación

sensitivity

Predicción		
Clase real	P	N
	P	TP
N	FP	TN

$$TPR = \text{sensitivity} = \text{recall} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

sensitivity

sensibilidad
% de aciertos considerando solo la clase real P

↑ => “el modelo está capturando los P ”]

Un clasificador que prediga 100% de P tendrá $\text{sensitivity} = 1$

specificity

especificidad
% de aciertos considerando solo la clase real N

↑ => “el modelo está capturando los N ”]

Un clasificador que dé 100% de N tendrá $\text{specificity} = 1$

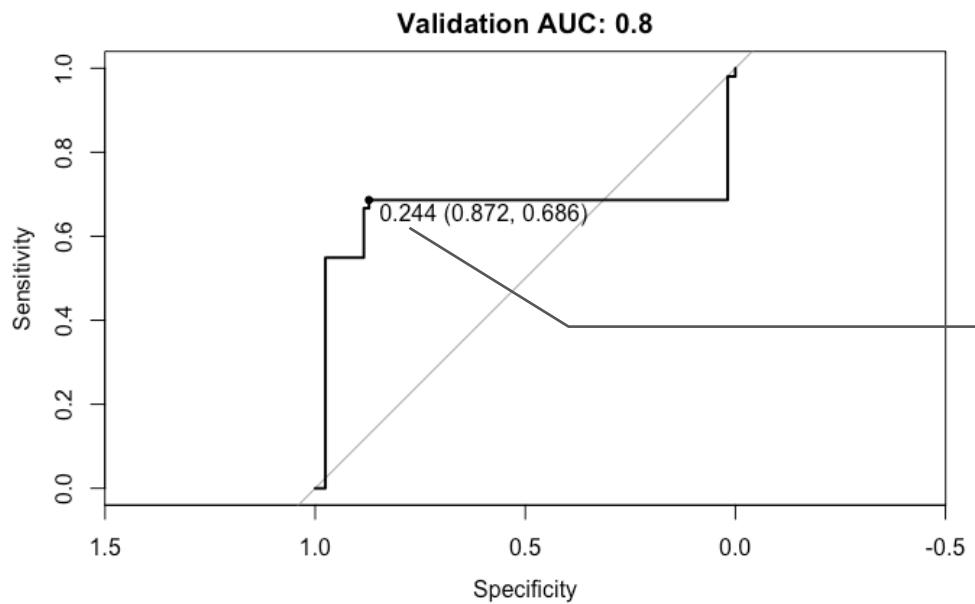
Aprendizaje supervisado

Métricas de clasificación

Curva ROC

sensitivity vs specificity : Representa el compromiso entre el % de aciertos en la clase positiva y el % de aciertos en la clase negativa de un modelo para todos los valores de umbral de clasificación

$$\frac{TP}{TP + FN} = \frac{TP}{P}$$



prob > 0.244 : clasificado como positivo
specificity : 0.872
sensitivity : 0.686

$$\frac{TN}{TN + FP} = \frac{TN}{N}$$

Aprendizaje supervisado

Bibliotecas

R



caret



tidymodels



mlr

Python



scikit-learn

Índice

Aprendizaje supervisado

Árboles de clasificación

Random forest

Boosting

SVM

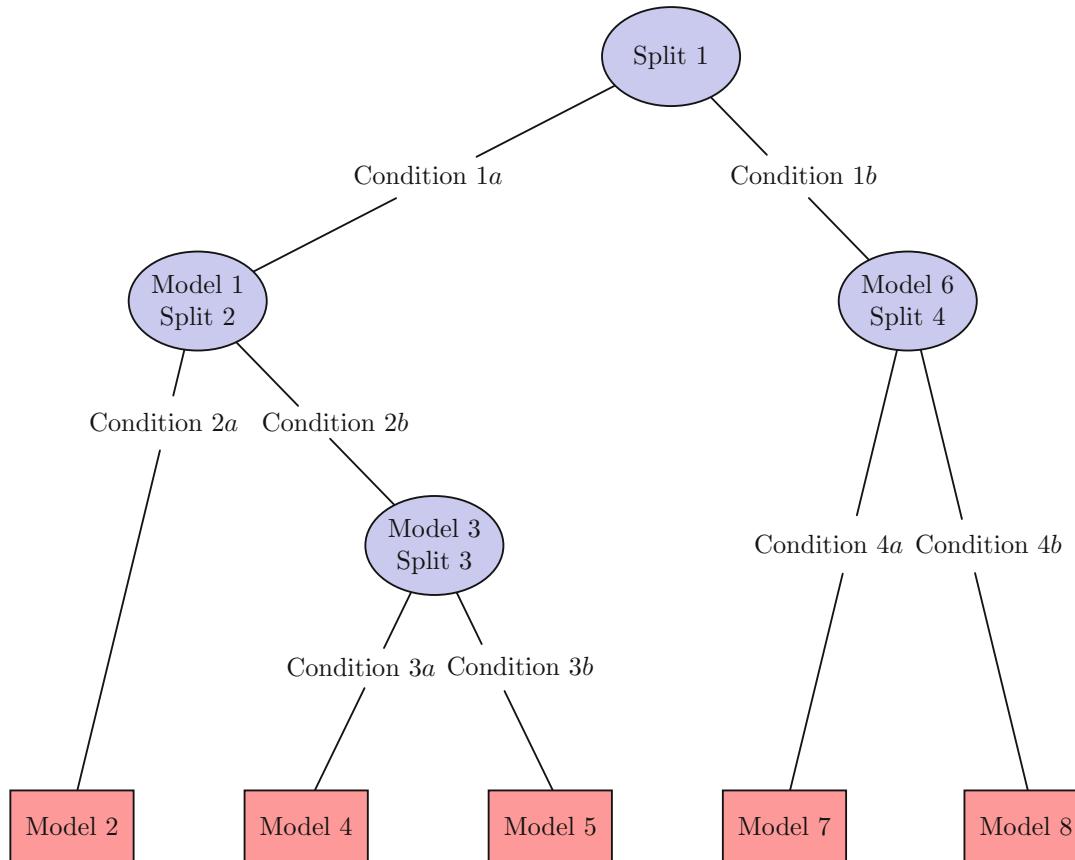
Aprendizaje supervisado

Árboles de decisión

CART

`method = "rpart"`

- `.cp` : Complexity parameter



Aprendizaje supervisado

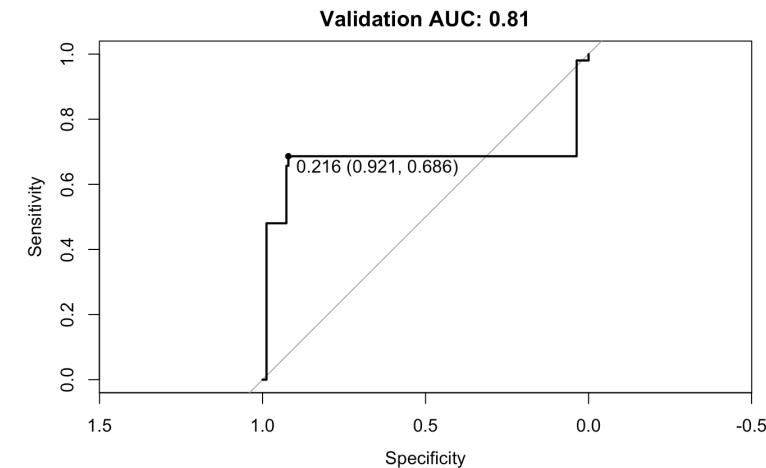
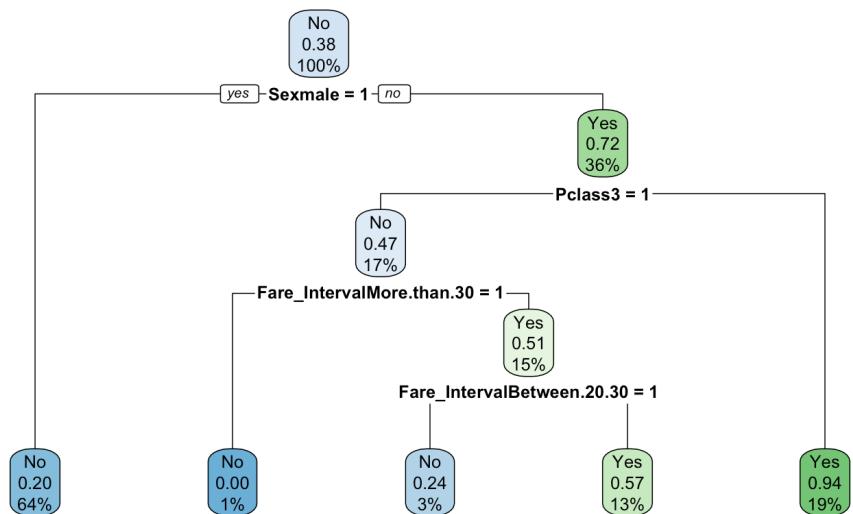
Árboles de decisión

CART

method = “rpart”

```
rpartModel <- train(Survived ~ .,
                      data = train,
                      method = "rpart",
                      metric = "Accuracy",
                      trControl = rpartCtrl,
                      tuneGrid = rpartParametersGrid)
```

```
predictionValidationProb <- predict(rpartModel, val, type = "prob")
auc1 <- roc(val$Survived,
            predictionValidationProb[["Yes"]])
roc_validation1 <- plot.roc(auc1,
                            ylim=c(0,1),
                            type = "S",
                            print.thres = TRUE,
                            main=paste('Validation AUC:', round(auc1$auc[[1]], 2)))
```



Aprendizaje supervisado

Random forest

Random forest

method = “rf”

- `.mtry` : Number of variables randomly sampled as candidates at each split
- `ntree` : Max number of trees (en train)

Model	method Value	Type	Libraries	Tuning Parameters
Conditional Inference Random Forest	cforest	Classification, Regression	party	mtry
Ensembles of Generalized Linear Models	randomGLM	Classification, Regression	randomGLM	maxInteractionOrder
Oblique Random Forest	ORFlog	Classification	obliqueRF	mtry
Oblique Random Forest	ORFpls	Classification	obliqueRF	mtry
Oblique Random Forest	ORFridge	Classification	obliqueRF	mtry
Oblique Random Forest	ORFsvm	Classification	obliqueRF	mtry
Parallel Random Forest	parRF	Classification, Regression	e1071, randomForest, foreach, import	mtry
Quantile Random Forest	qrf	Regression	quantregForest	mtry
Random Ferns	rFerns	Classification	rFerns	depth
Random Forest	ranger	Classification, Regression	e1071, ranger, dplyr	mtry, splitrule
Random Forest	Rborist	Classification, Regression	Rborist	predFixed
Random Forest	rf	Classification, Regression	randomForest	mtry
Random Forest by Randomization	extraTrees	Classification, Regression	extraTrees	mtry, numRandomCuts
Random Forest Rule-Based Model	rfRules	Classification, Regression	randomForest, inTrees, plyr	mtry, maxdepth
Regularized Random Forest	RRF	Classification, Regression	randomForest, RRF	mtry, coefReg, coefImp
Regularized Random Forest	RRFglobal	Classification, Regression	RRF	mtry, coefReg
Weighted Subspace Random Forest	wsrf	Classification	wsrf	mtry

Aprendizaje supervisado

Boosting

Boosting

`method = "xgbTree"`

XGBoost (eXtreme Gradient Boosting)

<http://topepo.github.io/caret/train-models-by-tag.html#boosting>

La construcción del árbol se hace de forma que se minimizan iterativamente los errores de clasificación (*residuals*)

General

<http://xgboost.readthedocs.io/en/latest/>

Fundamentos teóricos

<https://xgboost.readthedocs.io/en/latest/tutorials/model.html>

Ejemplos

<https://github.com/dmlc/xgboost/tree/master/demo>

Parámetros

<https://xgboost.readthedocs.io/en/latest/parameter.html>

Análisis predictivo

Clasificación automática

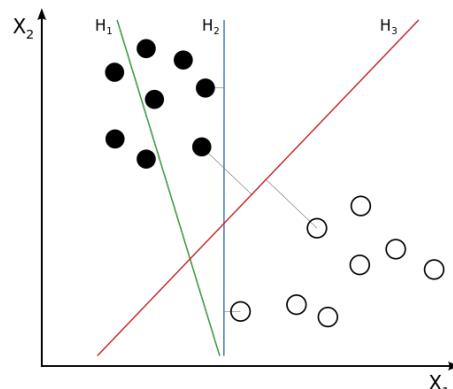
SVM

`method = "svmRadial"`

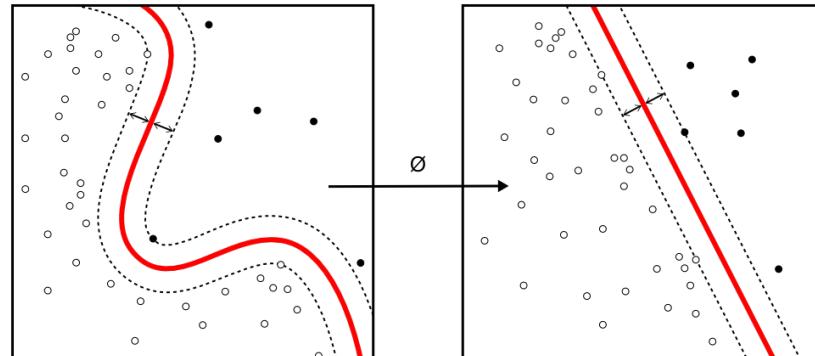
Hiperplano de máximo margen

Kernels para proyectar instancias a un espacio de mayor dimensionalidad

> Teoría de Lagrange para optimización



By User:ZackWeinberg, based on PNG version
by User:Cyc - This file was derived from: Svm
separating hyperplanes.png, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=22877598>



$$\text{radial basis function} = \exp(-\sigma \|\mathbf{x} - \mathbf{u}\|^2)$$

- `.sigma` : Kernel parameter
- `.C` : Cost coefficient

$$\text{Cost} \sum_{i=1}^n L_\epsilon(y_i - \hat{y}_i)$$

Índice

Aprendizaje profundo

Fundamentos de redes neuronales

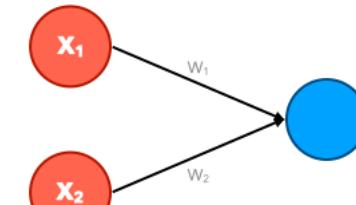
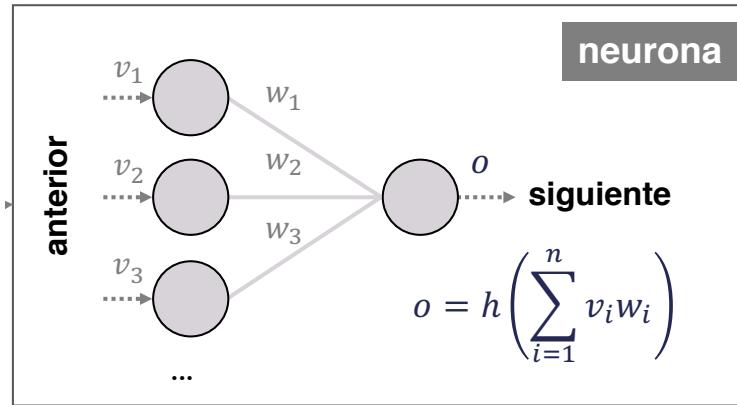
Redes convolutivas

Redes recurrentes

Redes para grafos

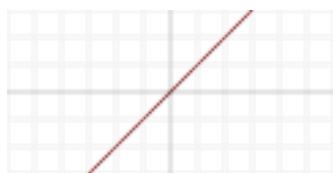
Aprendizaje profundo

Fundamentos de redes neuronales

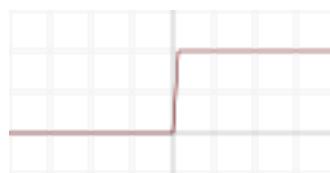


<https://towardsdatascience.com/an-illustrated-guide-to-artificial-neural-networks-f149a549ba74>

Función de activación h



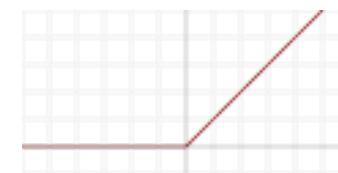
lineal



salto



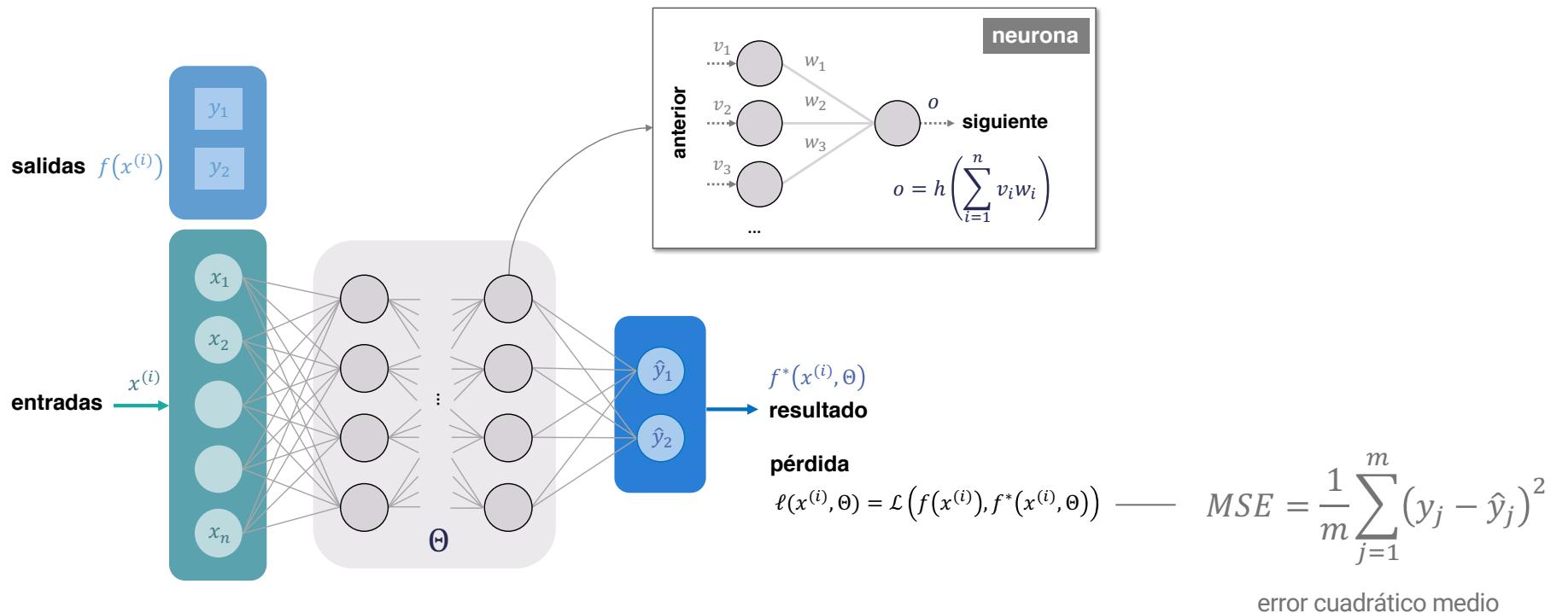
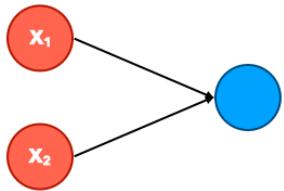
sinusoidal / sigmoide



rectificador lineal (ReLU)

Aprendizaje profundo

Fundamentos de redes neuronales

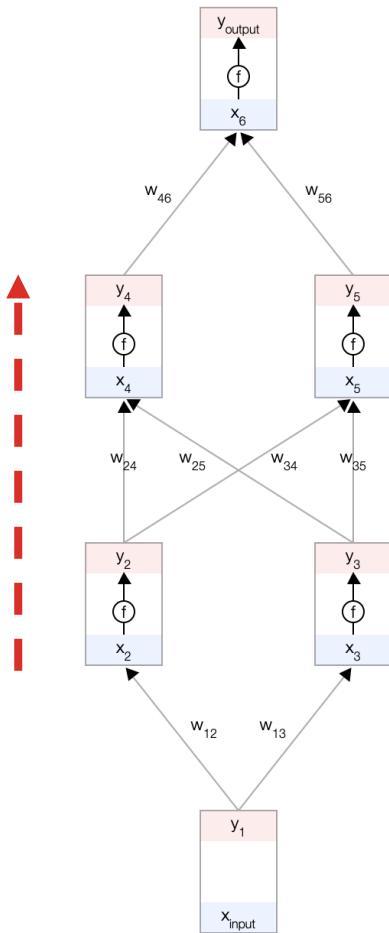


Aprendizaje profundo

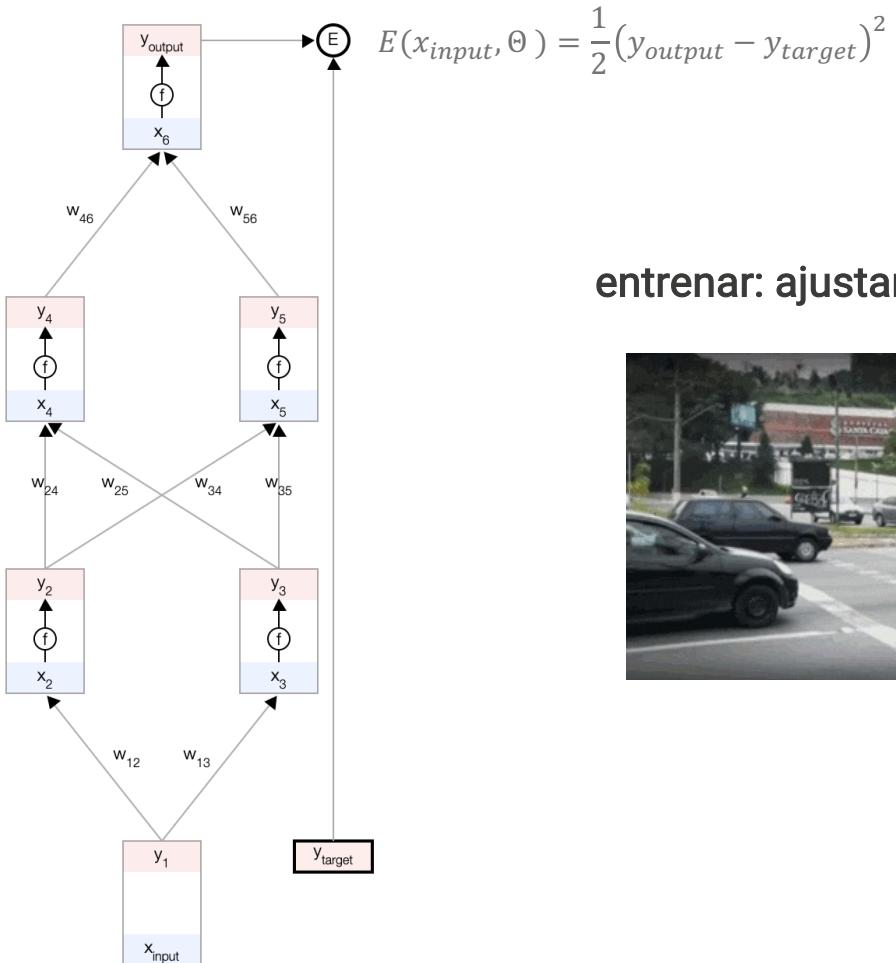
Fundamentos de redes neuronales



Introducción al aprendizaje automático
http://sl.ugr.es/0atf_v2



cálculo de la salida (*forward pass*)



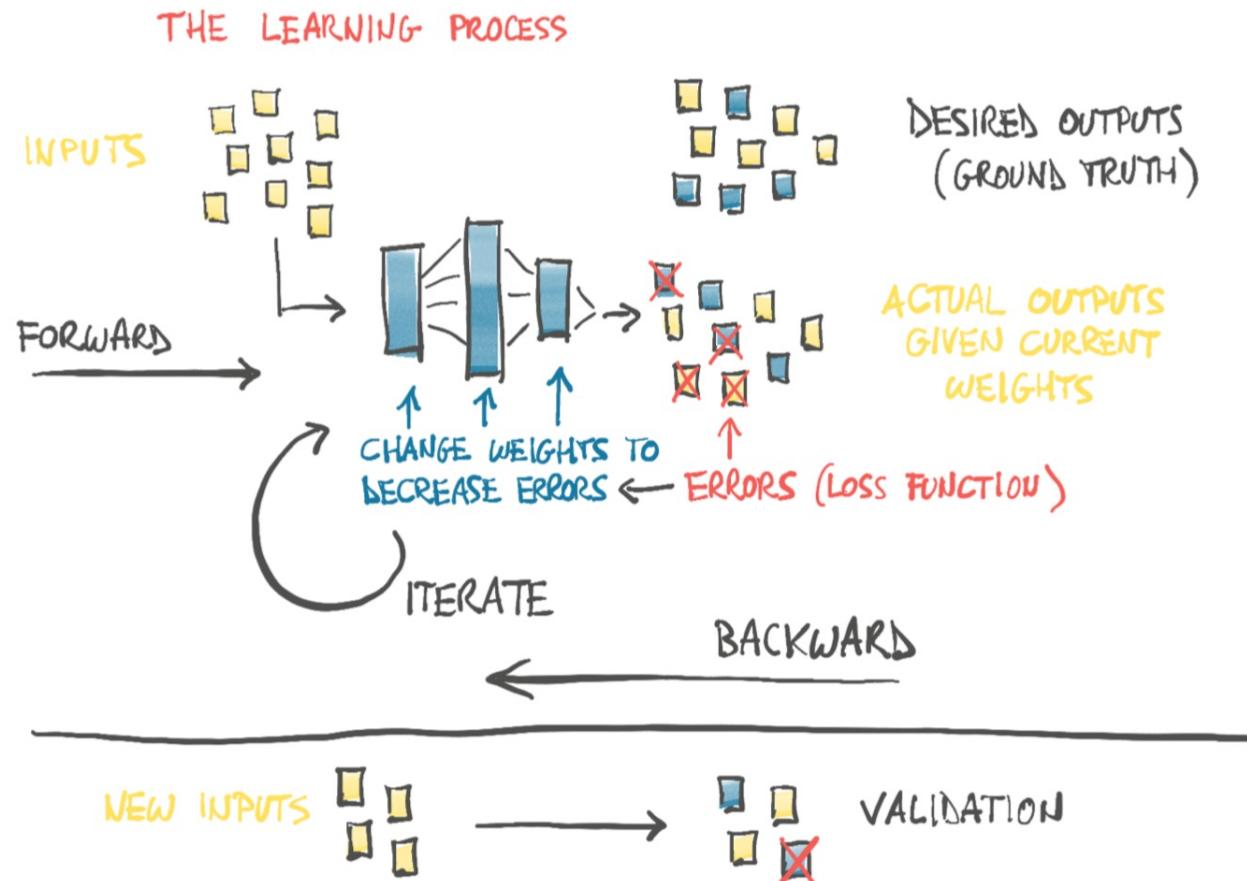
cálculo del error

entrenar: ajustar Θ para minimizar E



Aprendizaje profundo

Fundamentos de redes neuronales



Aprendizaje profundo

Fundamentos de redes neuronales

datos históricos

	x_1	...	x_n	y_1	...	y_m
$x^{(1)}$						
$x^{(2)}$						
...						
$x^{(K)}$						

random

datos entrenamiento [80%]

	x_1	...	x_n	y_1	...	y_m
$x^{(i)}$						
...						
$x^{(K')}$						

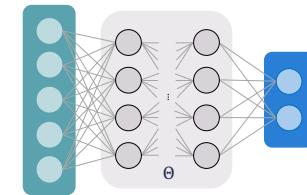
datos validación [20%]

	x_1	...	x_n	y_1	...	y_m
$x^{(i)}$						
...						
$x^{(K'')}$						

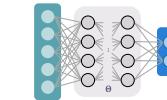


$$\Theta = \arg \min_{\Theta} \frac{1}{K'} \sum_{i=1}^{K'} \ell(x^{(i)}, \Theta)$$

optimización



evaluación



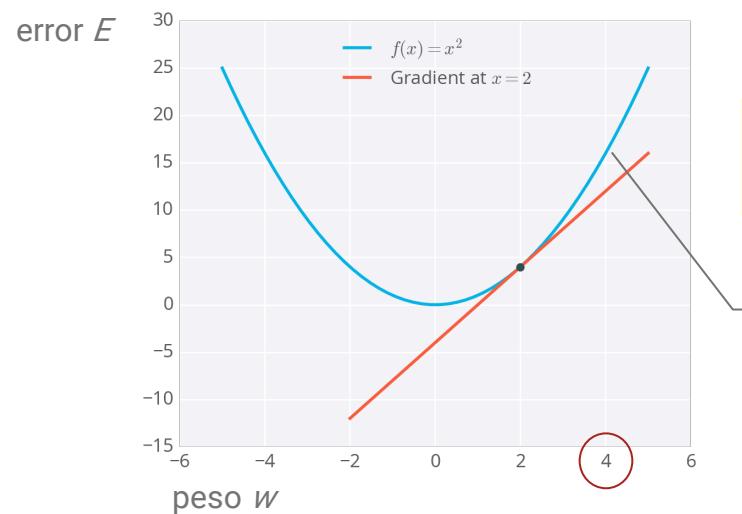
*accuracy
sensitivity – specificity
...*

Aprendizaje profundo

Fundamentos de redes neuronales ► Entrenamiento

Gradiente descendente

La actualización depende del gradiente de la función de error



$$\nabla f = \frac{\partial f(x)}{\partial x}$$

$$\begin{aligned}f(4) &= 16 \\ \nabla f &= 2x \\ \nabla f(4) &= 8\end{aligned}$$

Inicialmente, $w = 4$, $E(w) = 16$
Para buscar el w que minimiza E
Mover w una **cantidad proporcional** a -8

B. Rohrer (2020) How optimization works. https://brohrer.github.io/how_optimization_works_1.html

Aprendizaje profundo

Fundamentos de redes neuronales ► Entrenamiento

Backpropagation

Actualiza los pesos de la red propagando los errores desde la capa de salida hasta la capa inicial

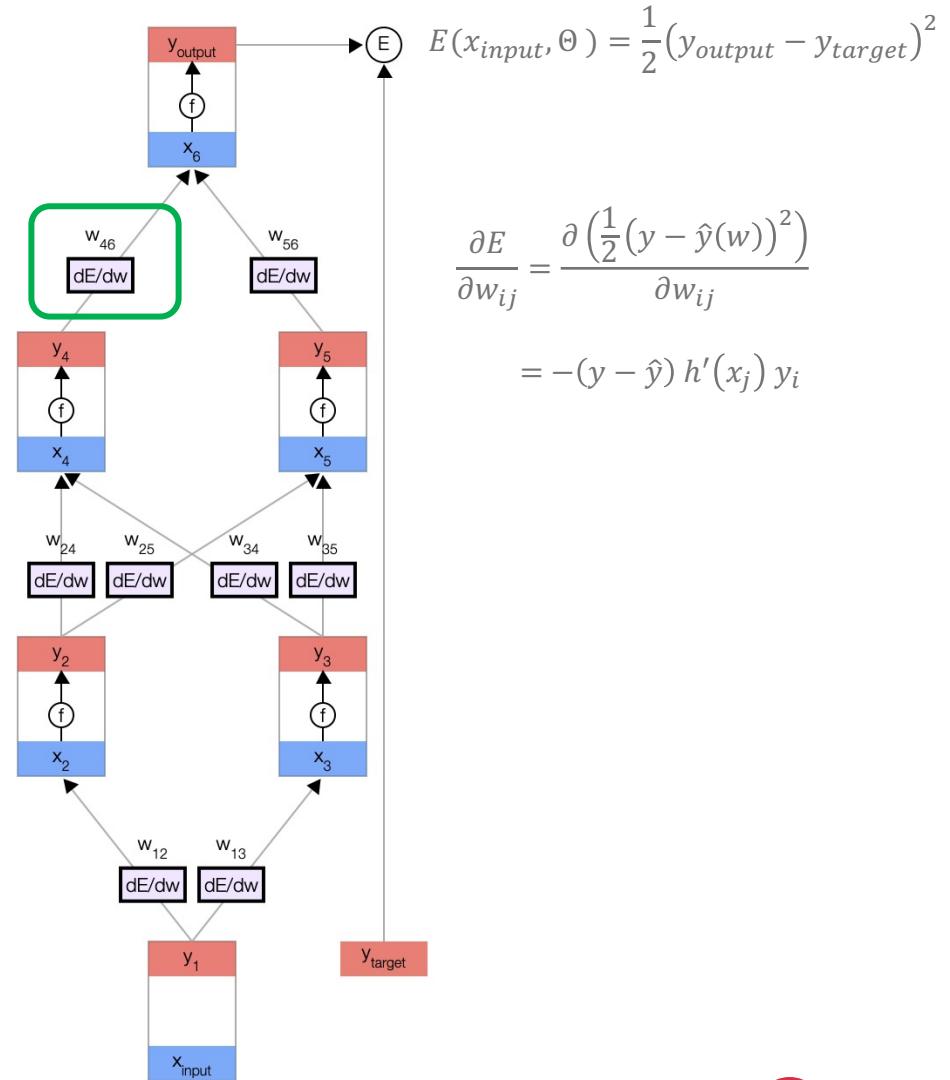
$$\nabla E = \frac{\partial f(w)}{\partial w}$$

$$\Delta w_{ij} \propto -\frac{\partial E}{\partial w_{ij}}$$

$$\boxed{\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}}$$



Introducción al aprendizaje automático
http://sl.ugr.es/0atf_v2



Aprendizaje profundo

Fundamentos de redes neuronales ► Entrenamiento

Algoritmos de optimización

En cada iteración...

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

Gradiente descendente por lotes

calcula gradientes para todo el conjunto de entrenamiento

Gradiente descendente estocástico

calcula gradientes para una sola instancia del conjunto de entrenamiento

Gradiente descendente (estocástico) con mini-lotes

calcula gradientes para un subconjunto de instancias del conjunto de entrenamiento

... y actualiza pesos

Tasas de aprendizaje adaptativas:

Adagrad, Adadelta, RMSProp, Adam, Nadam

Aprendizaje profundo

Fundamentos ► Desarrollo

Tensorflow + Keras [Google]

<https://www.tensorflow.org> , <https://keras.io>

PyTorch [Facebook]

<https://pytorch.org>

CNTK [Microsoft]

<https://www.microsoft.com/en-us/cognitive-toolkit>

MXNET [Apache]

<https://mxnet.incubator.apache.org>

Theano [MILA]

<http://deeplearning.net/software/theano/>

Aprendizaje profundo

Fundamentos ► Desarrollo

<https://colab.to/r>



[titanic-dl.ipynb](#)

Creacion de la red

```
model <- keras_model_sequential()
model <- model %>%
  layer_dense(units = 32, activation = "relu", input_shape = c(ncol(data) - 1)) %>%
  layer_dense(units = 16, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

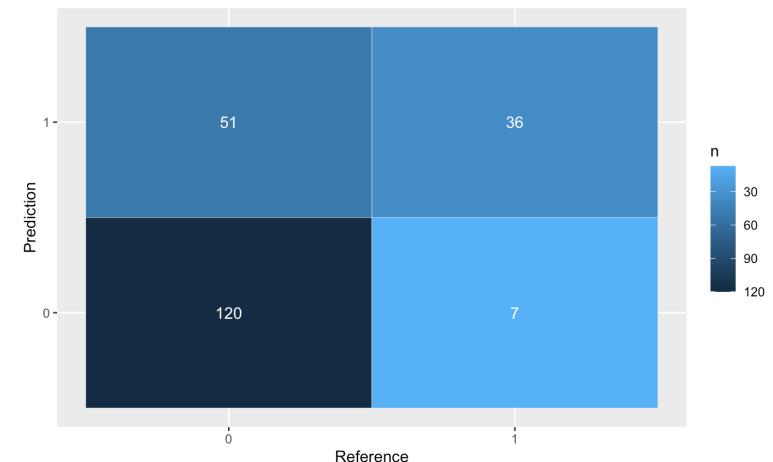
Compilar modelo

```
model %>% compile(
  loss = 'binary_crossentropy',
  metrics = c('accuracy'),
  optimizer = optimizer_adam()
)
```

Preparar entrenamiento y validación

Entrenar modelo

```
history <- model %>%
  fit(
    x_train, y_train,
    epochs = 20,
    batch_size = 100,
    validation_split = 0.2
)
```



Índice

Aprendizaje profundo

Fundamentos de redes neuronales

Redes convolutivas

Redes recurrentes

Redes para grafos

Aprendizaje profundo

Redes convolutivas

Aproximaciones a la clasificación de imágenes

Codificación directa



MNIST

28

28



784

Multi
clasificador



```
$loss  
[1] 0.1518216
```

```
$acc  
[1] 0.9803
```

Aprendizaje profundo

Redes convolutivas

Aproximaciones a la clasificación de imágenes

Pre-procesamiento y cálculo de características

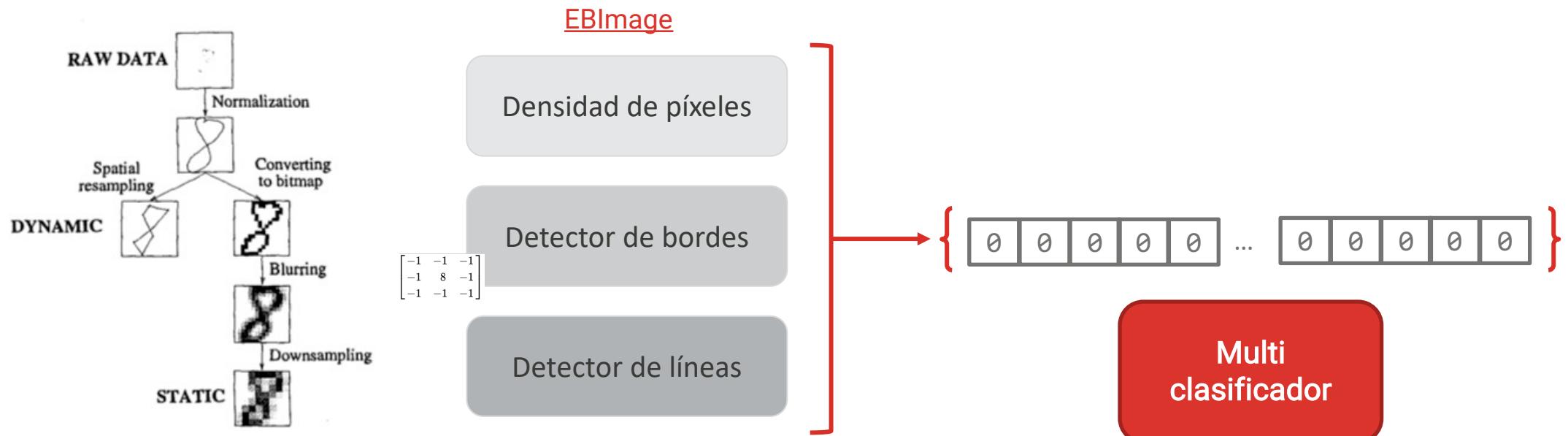


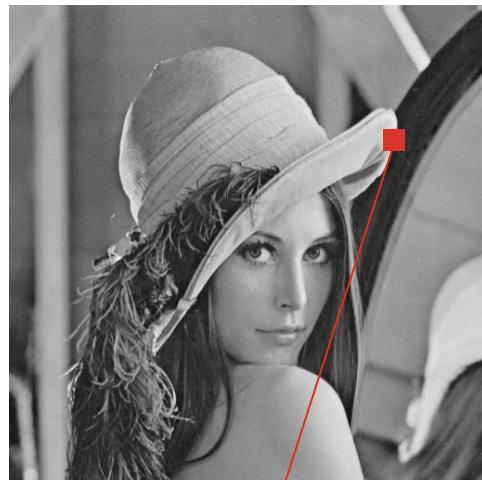
Figure 1: The stages of preprocessing.

F. Alimoglu, E. Alpaydin (1997) Combining multiple representations and classifiers for pen-based handwritten digit recognition. Proceedings of the 4th International Conference on Document Analysis and Recognition.

Aprendizaje profundo

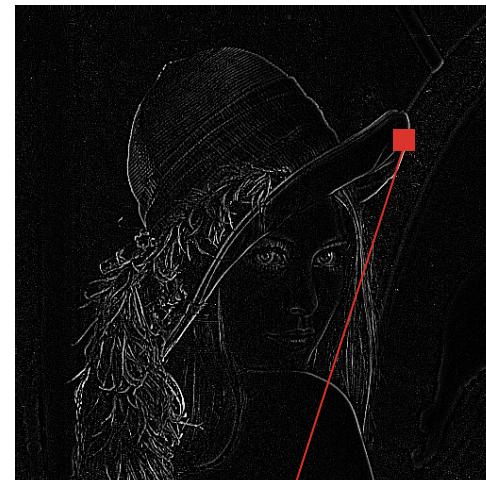
Redes convolutivas

Pre-procesamiento y extracción de características



filtro kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



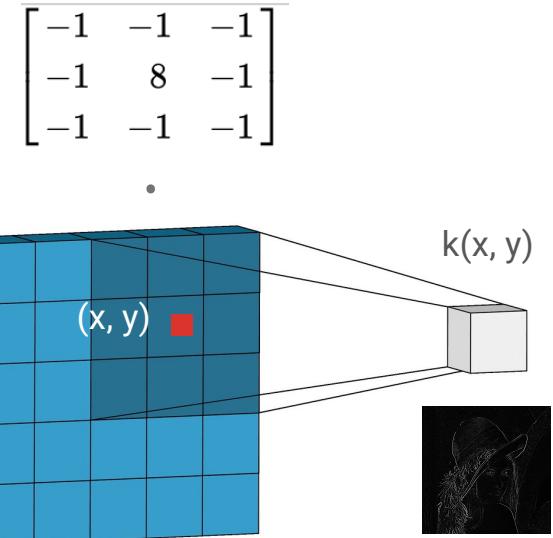
(414, 149)

$$\begin{array}{ccc} 221^{-1} & 217^{-1} & 218^{-1} \\ 213^{-1} & \textcolor{red}{203}^{+8} & 182^{-1} \\ 140^{-1} & 105^{-1} & 66^{-1} \end{array}$$

(414, 149)

30 44 132
207 255 232
79 0 0

$$k(414, 149) = 8 \times 203 - 221 - 217 - 218 - 213 - 182 - 140 - 105 - 66$$



Aprendizaje profundo

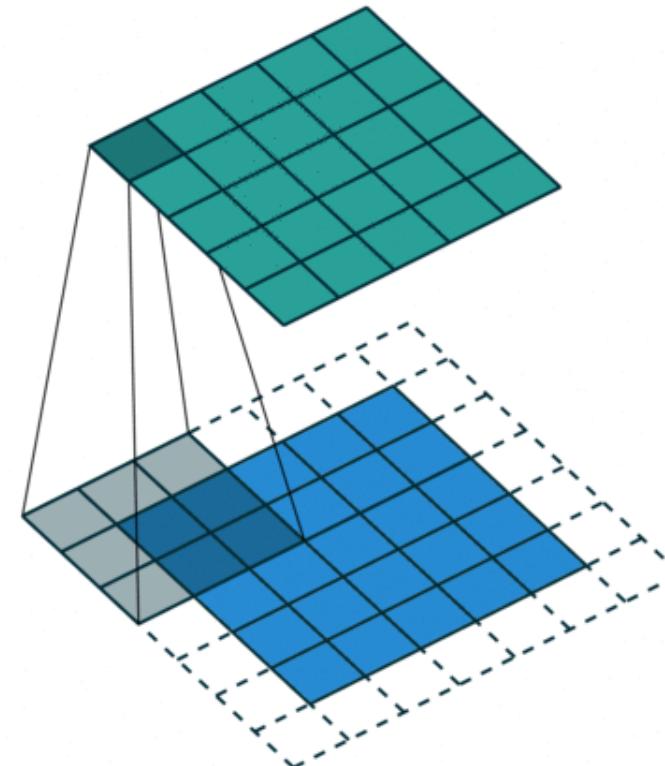
Redes convolutivas

Pre-procesamiento y extracción de características

3 ₀	3 ₁	2 ₂	1	0
0 ₂	0 ₂	1 ₀	3	1
3 ₀	1 ₁	2 ₂	2	3
2	0	0	2	2
2	0	0	0	1

0	1	2
2	2	0
0	1	2

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0



I. Shafkat (2018) Intuitively Understanding Convolutions for Deep Learning [[link](#)]

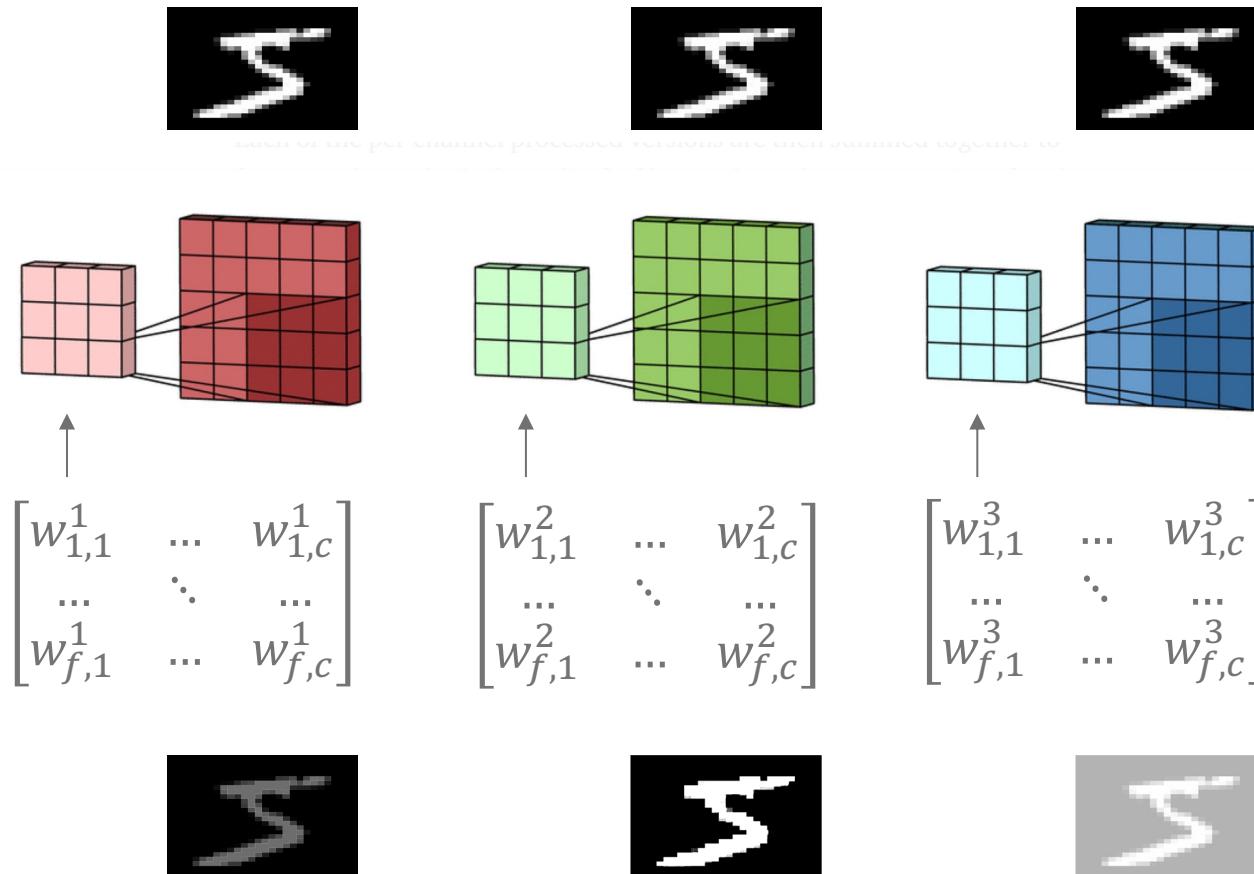
V. Dumoulin, F. Visin (2018) A guide to convolution arithmetic for deep learning [[link](#)]

Aprendizaje profundo

Redes convolutivas

Capas de convolución

Cada capa consta de varios filtros



Aprendizaje profundo

Redes convolutivas

Pooling

Se reduce el tamaño de una imagen

max-pooling 2x2, stride = 2

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters
and stride 2

6	8
3	4

max-pooling 3x3, stride = 1

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

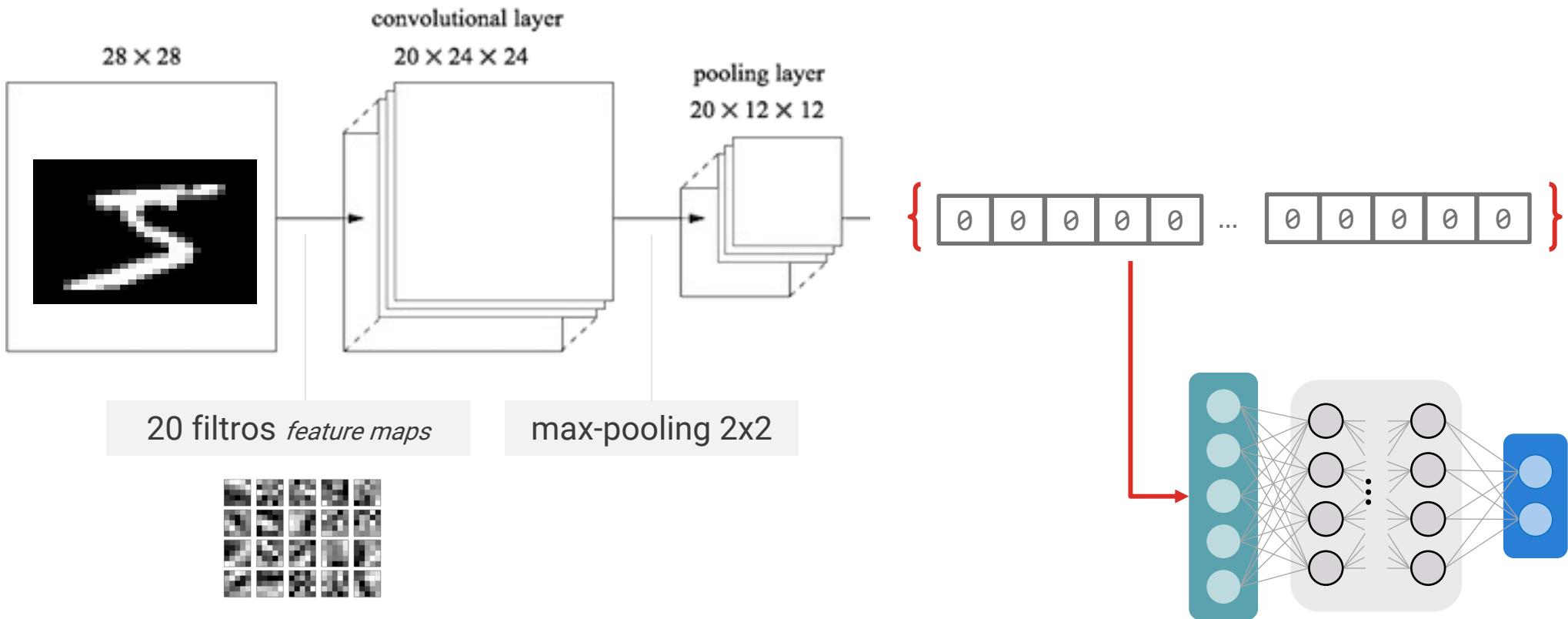
3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

Aprendizaje profundo

Redes convolutivas

Red completa

Capas de convolución + capa totalmente conectada (perceptrón multicapa)

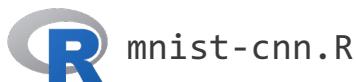
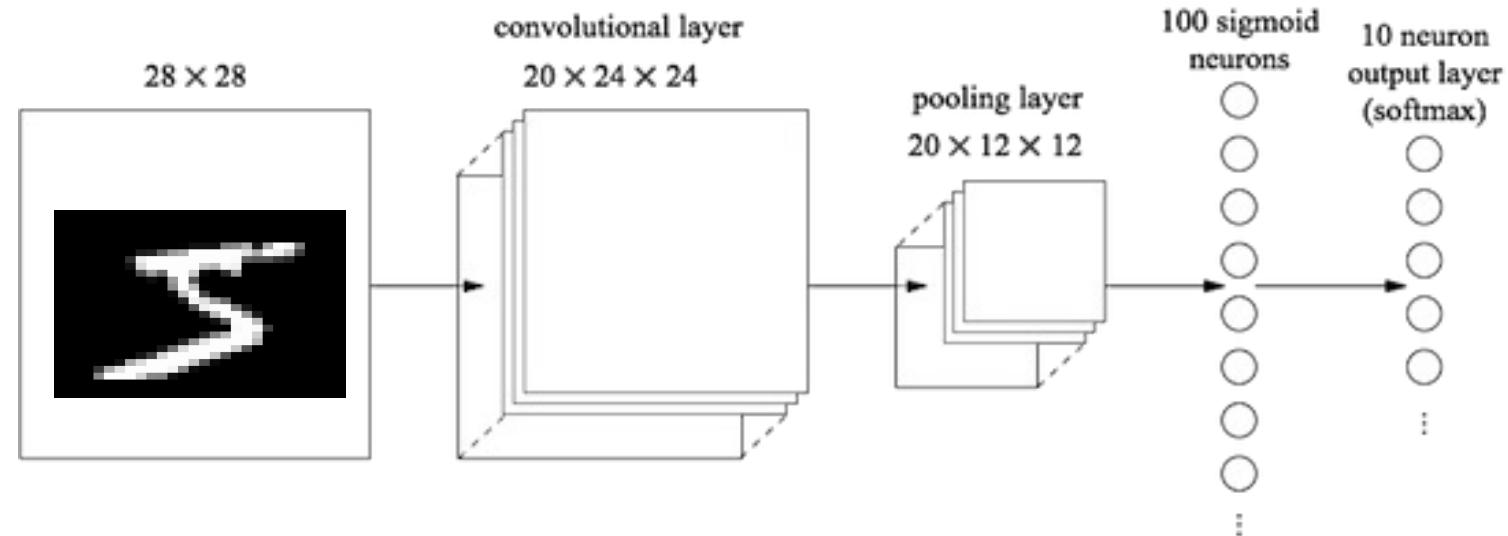


Aprendizaje profundo

Redes convolutivas

Red completa

Capas de convolución + capa totalmente conectada (perceptrón multicapa)



```
model <- keras_model_sequential()
model %>%
  layer_conv_2d(filters = 20, kernel_size = c(5, 5), activation = "relu", input_shape = c(28, 28, 1)) %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_flatten() %>%
  layer_dense(units = 100, activation = "sigmoid") %>%
  layer_dense(units = 10, activation = "softmax")
```

```
$loss
[1] 0.04407667

$acc
[1] 0.9854
```

Índice

Aprendizaje profundo

Fundamentos de redes neuronales

Redes convolutivas

Redes recurrentes

Redes para grafos

Aprendizaje profundo

Redes recurrentes

Series de datos

Secuencias de datos con relación temporal entre ellos

Univariate

Una variable a la entrada (una a la salida)

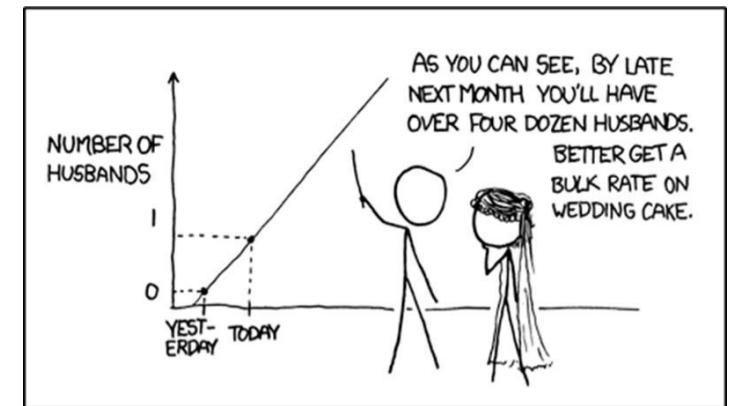
Multivariate

Varias variables a la entrada (una o varias a la salida)

Características del aprendizaje

Se busca predecir un valor (o valores) de la serie en el futuro

La predicción se realiza a partir de los valores $t, t-1, t-2, \dots$ de la(s) secuencia(s)



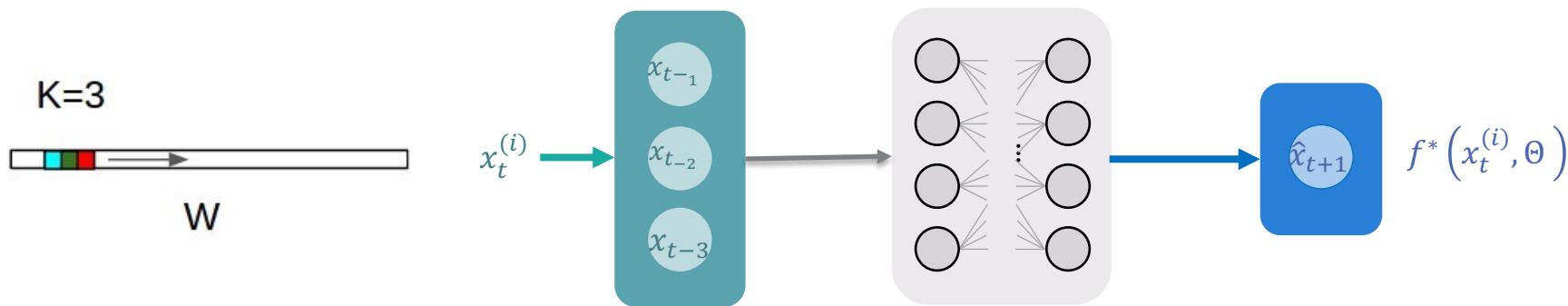
Extrapolating <https://xkcd.com/605/>

Aprendizaje profundo

Redes recurrentes

MLP “temporal”

La secuencia se segmenta en intervalos



Convolución temporal

Se aplica una convolución sobre la secuencia para agregar los valores



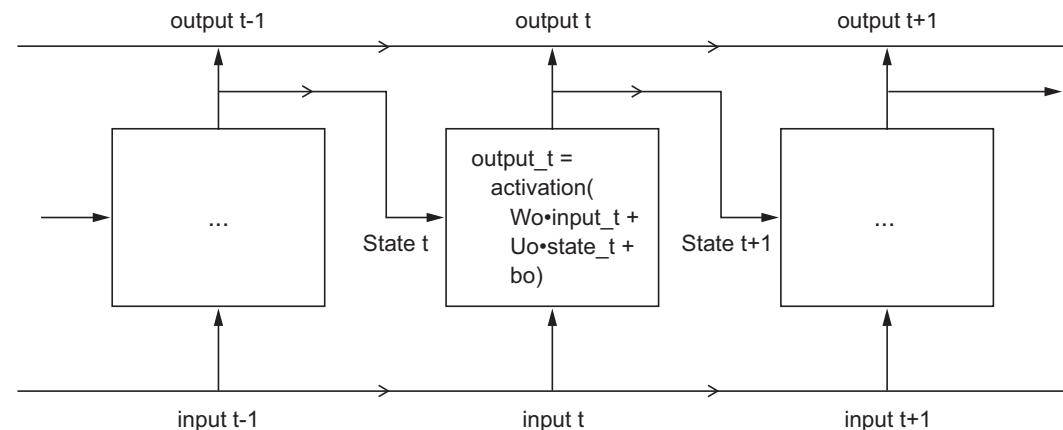
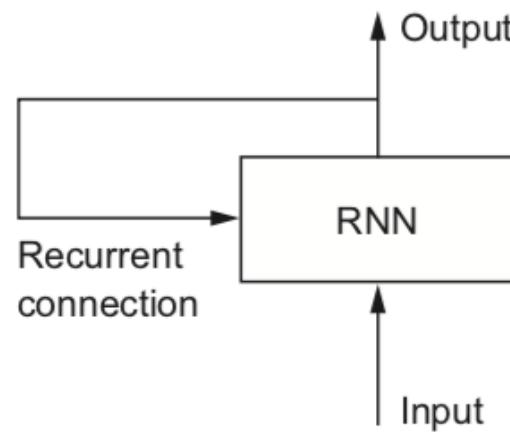
<https://stackoverflow.com/questions/42883547/intuitive-understanding-of-1d-2d-and-3d-convolutions-in-convolutional-neural-n>

Aprendizaje profundo

Redes recurrentes

Redes neuronales recurrentes

La salida depende de los valores de entrada y de la salida previa de la red



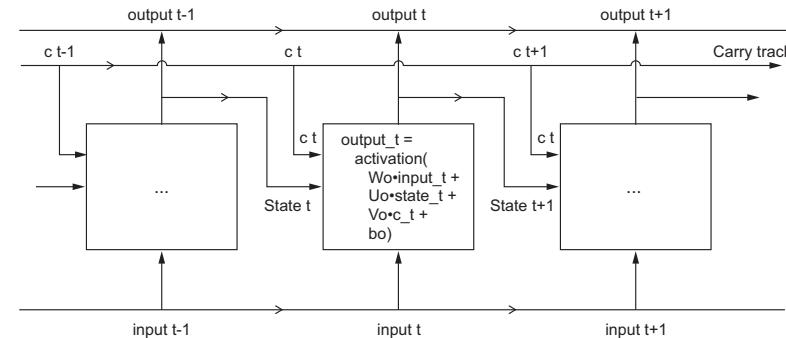
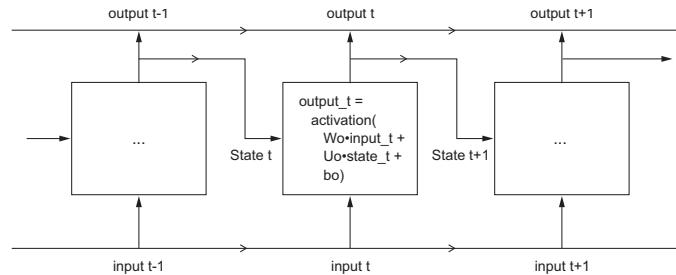
layer_simple_rnn

Aprendizaje profundo

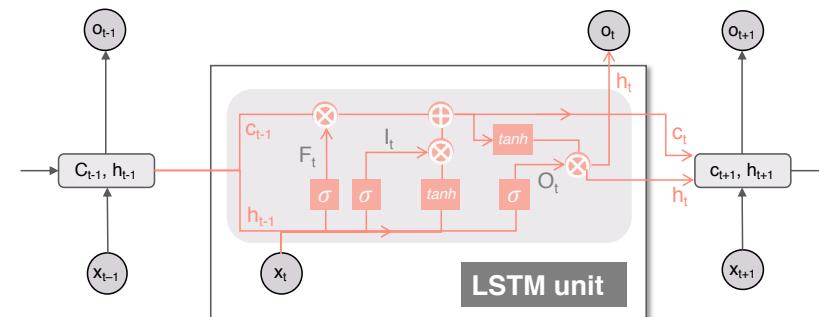
Redes recurrentes

Redes neuronales recurrentes avanzadas

Arquitecturas más complejas para conseguir recordar a más largo plazo



LSTM: Long-short therm memory
layer_lstm



Índice

Aprendizaje profundo

Fundamentos de redes neuronales

Redes convolutivas

Redes recurrentes

Redes para grafos

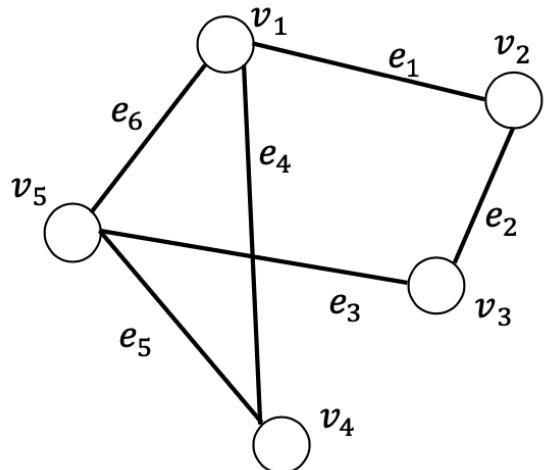
Aprendizaje profundo

Redes para grafos

Definición

Un grafo es un par $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, donde $\mathcal{V} = \{v_1, \dots, v_n\}$ es un conjunto de $N = |\mathcal{V}|$ **nodos** (o vértices) y $\mathcal{E} = \{e_1, \dots, e_m\}$ es un conjunto de $M = |\mathcal{E}|$ **arcos** (o aristas)

Representación



Matriz de adyacencia

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Aprendizaje profundo

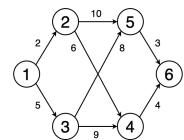
Redes para grafos

Algoritmos

Características estructurales (por ej.)

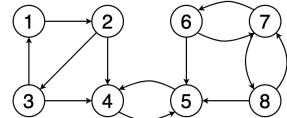
Caminos más cortos

Algoritmo de Dijkstra



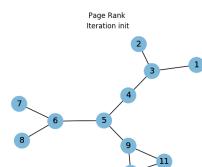
Comunidades

Algoritmo de Tarjan



Elementos relevantes

PageRank



Visualización

ForceAtlas2



Predictión sobre el grafo

Clasificación de grafos

Predicción de enlaces

Predicción de propiedades de nodos

...

Aprendizaje Automático

Aprendizaje profundo

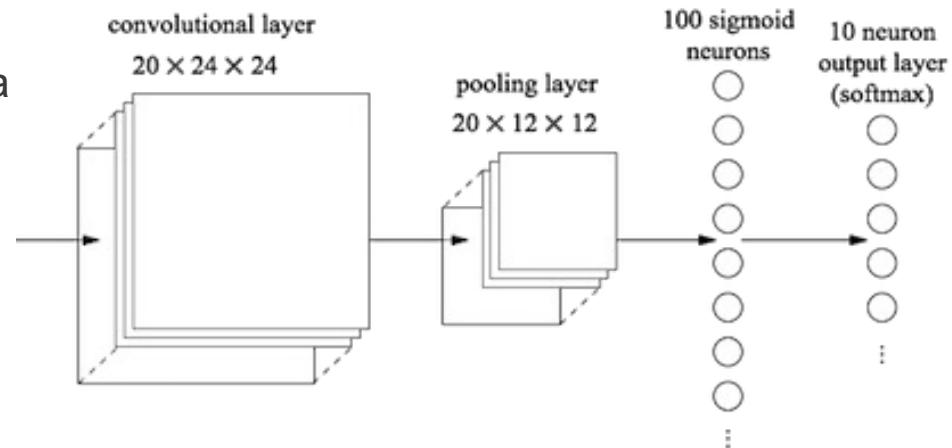
Redes para grafos

Una red neuronal puede procesar directamente la representación matricial del grafo...

Redes neuronales convolutivas

Matriz de adyacencia

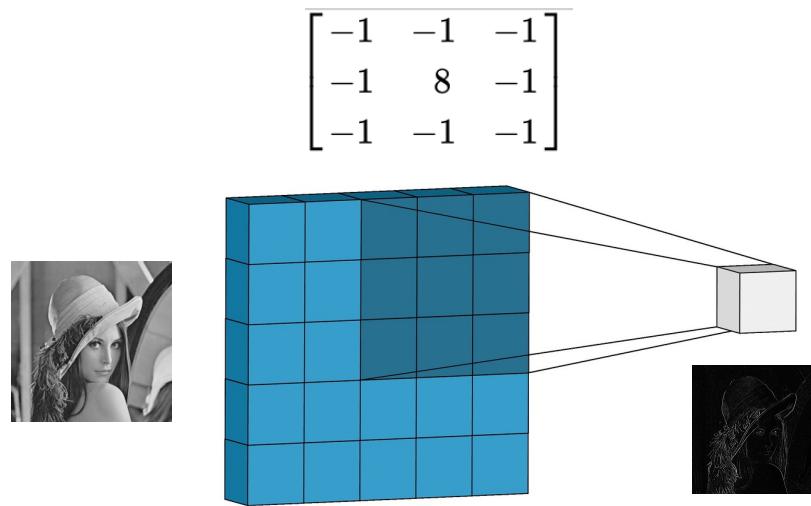
$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$



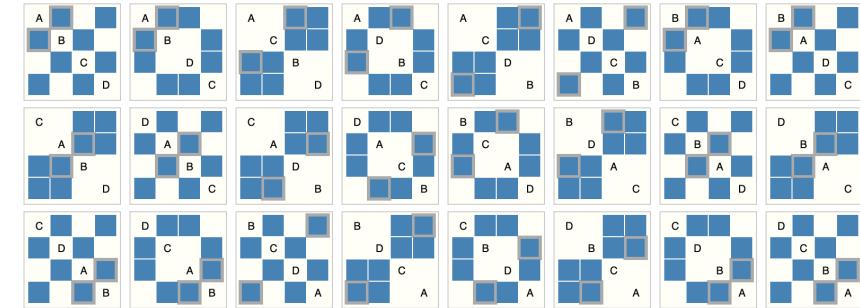
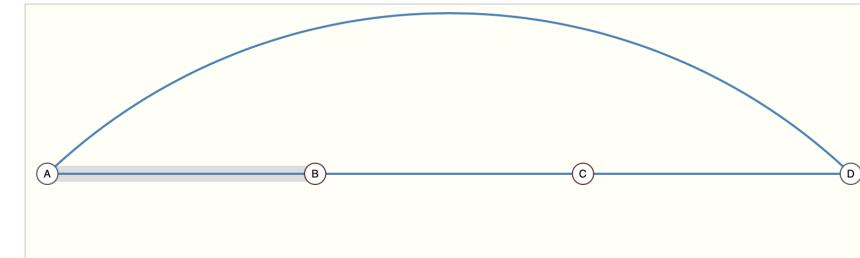
Aprendizaje profundo

Redes para grafos

...pero la representación debería ser independiente de cambios en el orden de los nodos



las convoluciones sobre imágenes son invariantes



B. Sanchez-Lengeling et al. A Gentle Introduction to Graph Neural Networks. Distill, 2021

las convoluciones sobre la representación matricial no son invariantes

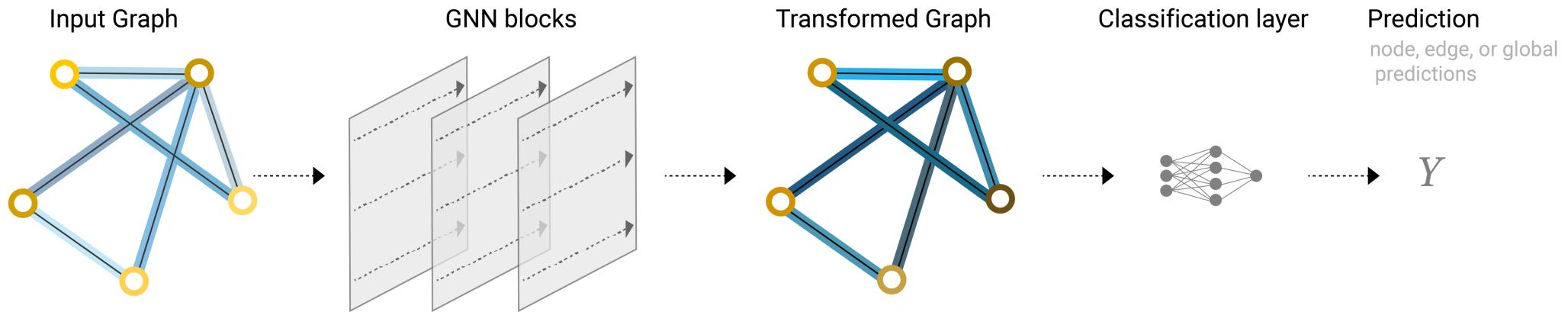
Aprendizaje profundo

Redes para grafos

Graph neural networks (GNNs)

Entrada: grafo

Salida: predicción sobre nodos, aristas o el grafo completo



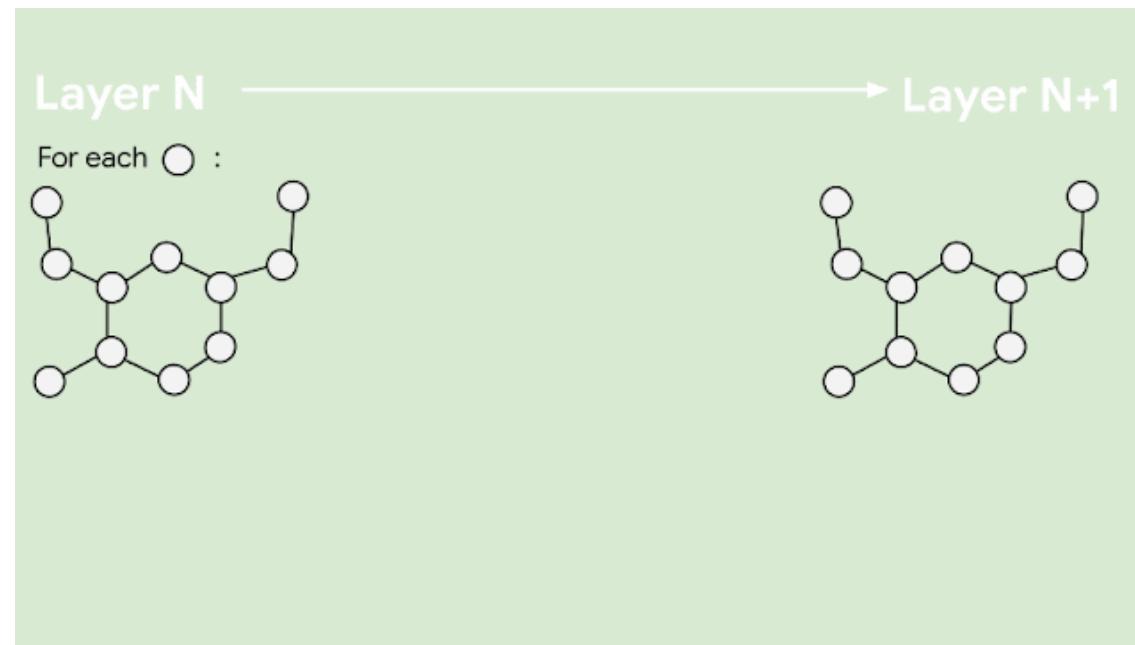
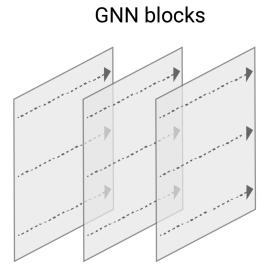
B. Sanchez-Lengeling et al. A Gentle Introduction to Graph Neural Networks. Distill, 2021

Aprendizaje profundo

Redes para grafos

Graph neural networks (GNNs)

Agregación de valores en los bloques GNN



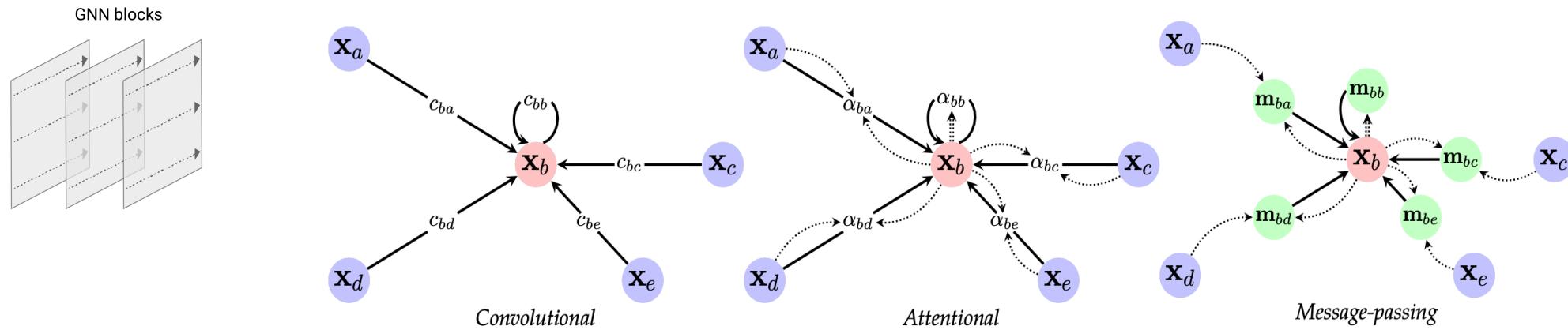
A.B. Wiltschko et al. Learning to Smell: Using Deep Learning to Predict the Olfactory Properties of Molecules. 2021

Aprendizaje profundo

Redes para grafos

Graph neural networks (GNNs)

Agregación de valores en los bloques GNN



diferenciables

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} c_{uv} \psi(\mathbf{x}_v) \right)$$

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} a(\mathbf{x}_u, \mathbf{x}_v) \psi(\mathbf{x}_v) \right)$$

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v) \right)$$

M. M. Bronstein, J. Bruna, T. Cohen, P. Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. 2021

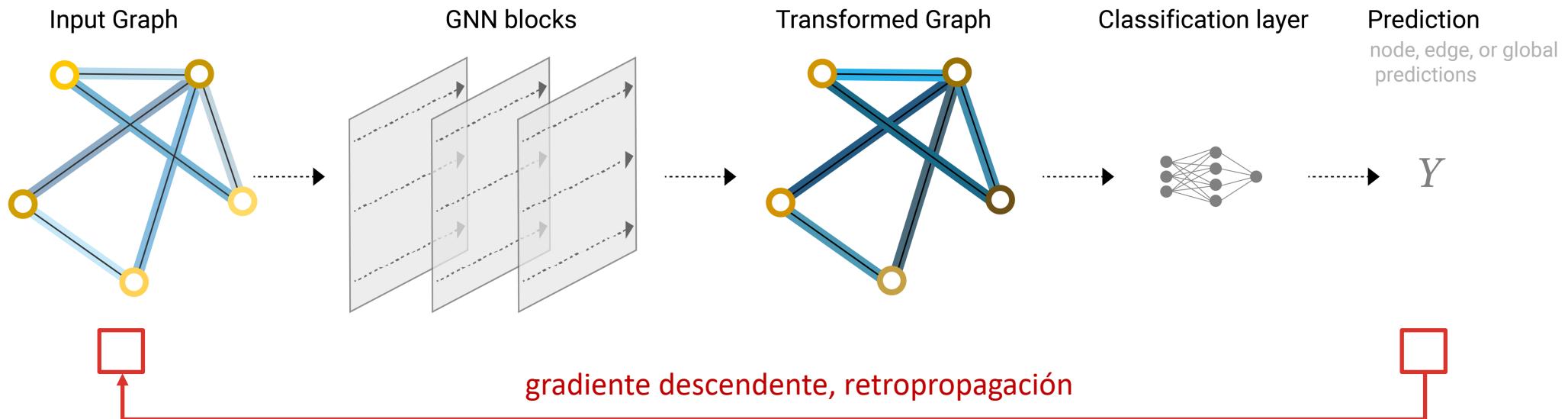
Aprendizaje profundo

Redes para grafos

Graph neural networks (GNNs)

Entrada: grafo

Salida: predicción sobre nodos, aristas o el grafo completo

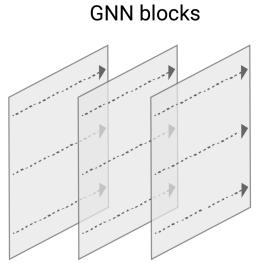


Aprendizaje profundo

Redes para grafos

Graph neural networks (GNNs)

Agregación de valores en los bloques GNN



Graph Convolutional Network (GCN)

Kipf & Welling, 2016 <https://arxiv.org/abs/1609.02907>

Graph Attention Networks (GAT)

Veličković et al., 2017 <https://arxiv.org/abs/1710.10903>

GraphSAGE

Hamilton, Yin & Leskovec, 2019 <https://arxiv.org/abs/1706.02216>

Graph Isomorphism Network (GIN)

Xu et al., 2019 <https://arxiv.org/abs/1810.00826>

Aprendizaje profundo

Redes para grafos

Graph neural networks (GNNs)

Bibliotecas



PyTorch Geometric

E/S de grafos

Datasets & benchmarks

Definición de capas GNN

Definición de capas de *pooling*

Gestión del proceso de entrenamiento



Spektral

```
#####
# Fit model
#####
@tf.function(input_signature=loader_tr.tf_signature(), experimental_relax_shapes=True)
def train_step(inputs, target):
    with tf.GradientTape() as tape:
        predictions = model(inputs, training=True)
        loss = loss_fn(target, predictions) + sum(model.losses)
        gradients = tape.gradient(loss, model.trainable_variables)
        optimizer.apply_gradients(zip(gradients, model.trainable_variables))
        acc = tf.reduce_mean(categorical_accuracy(target, predictions))

    return loss, acc

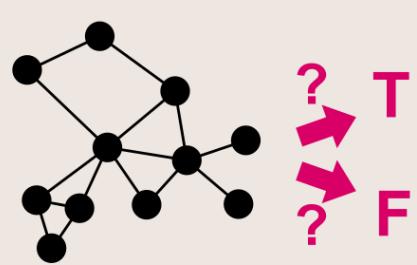
def evaluate(loader):
    output = []
    step = 0
    while step < loader.steps_per_epoch:
        step += 1
        inputs, target = loader.__next__()
        pred = model(inputs, training=False)
        outs = (
            loss_fn(target, pred),
            tf.reduce_mean(categorical_accuracy(target, pred)),
            len(target), # Keep track of batch size
        )
        output.append(outs)
    if step == loader.steps_per_epoch:
        output = np.array(output)
    return np.average(output[:, :-1], 0, weights=output[:, -1])
```



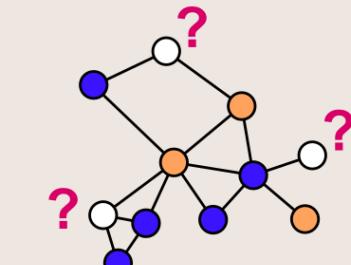
Aprendizaje profundo

Redes para grafos

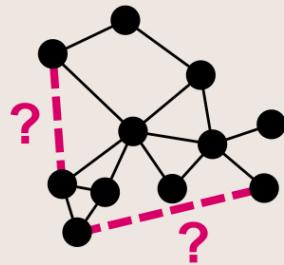
Graph Classification



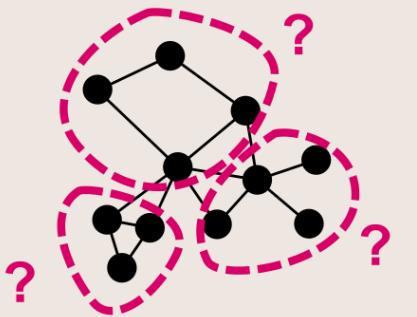
Node Classification



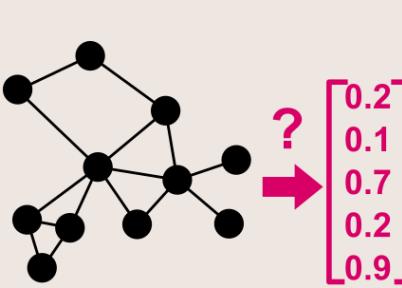
Link Prediction



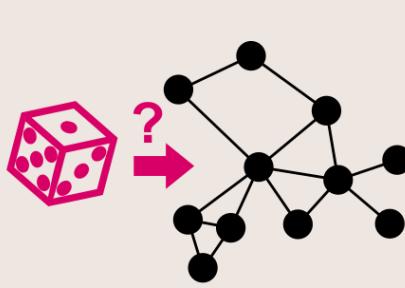
Community Detection



Graph Embedding



Graph Generation

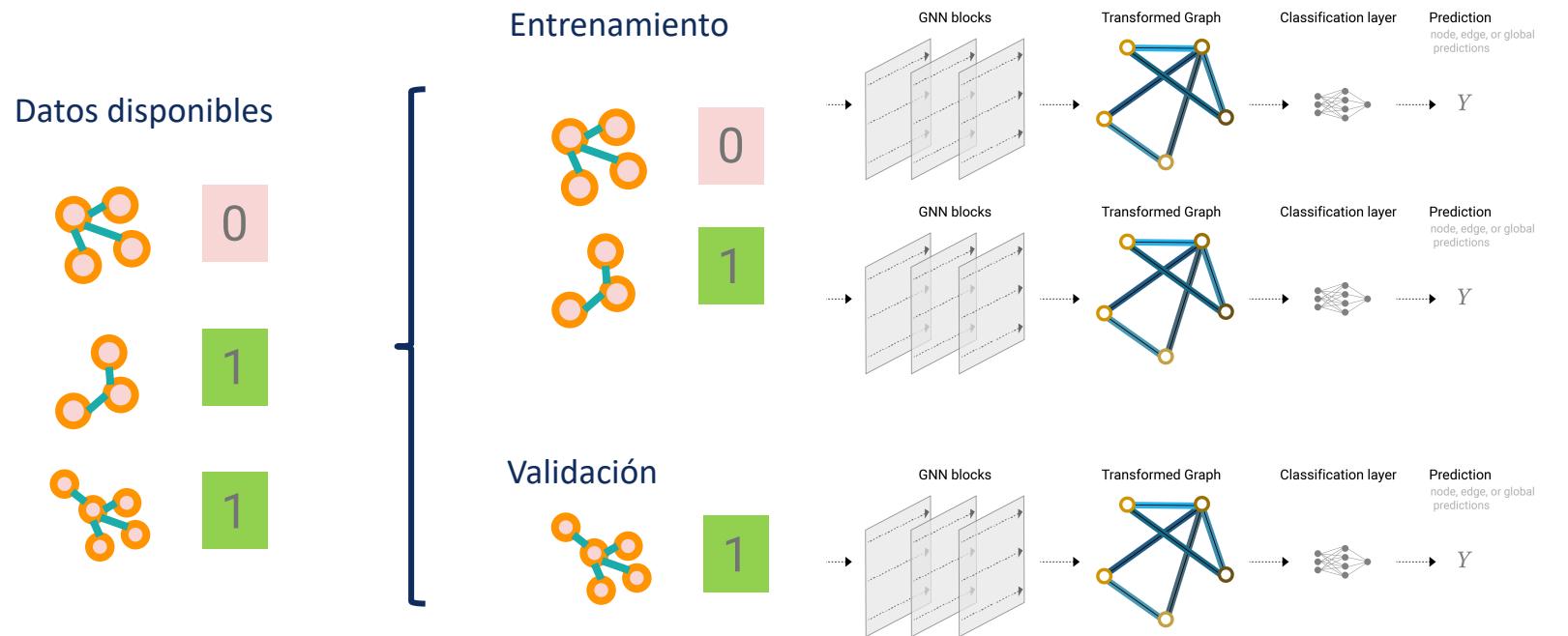
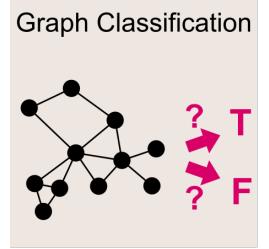


Formular el problema a resolver como una tarea de predicción sobre elementos del grafo

Aprendizaje profundo

Redes para grafos

Clasificación de grafos



Error de entrenamiento

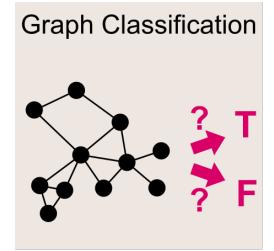
El modelo es capaz de separar las clases

Error de validación

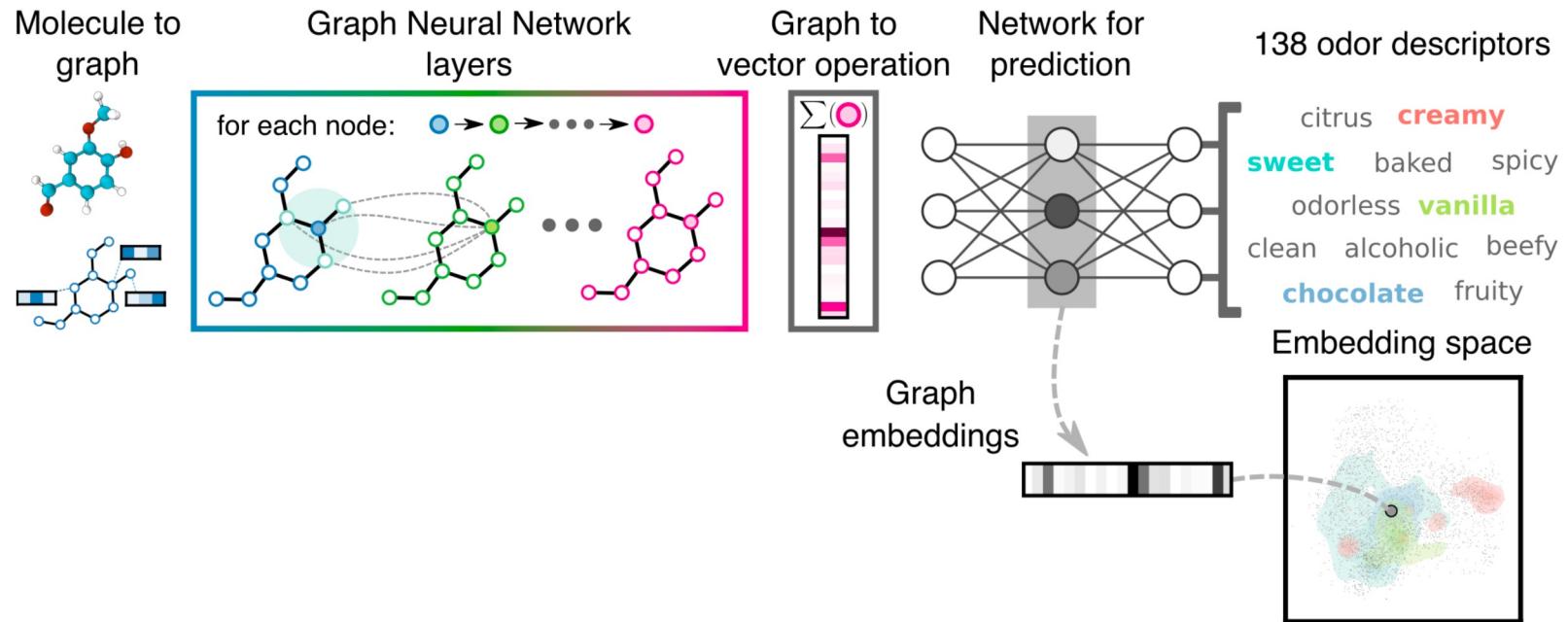
El modelo es capaz de generalizar a ejemplos no vistos

Aprendizaje profundo

Redes para grafos



Predicción de olores de moléculas

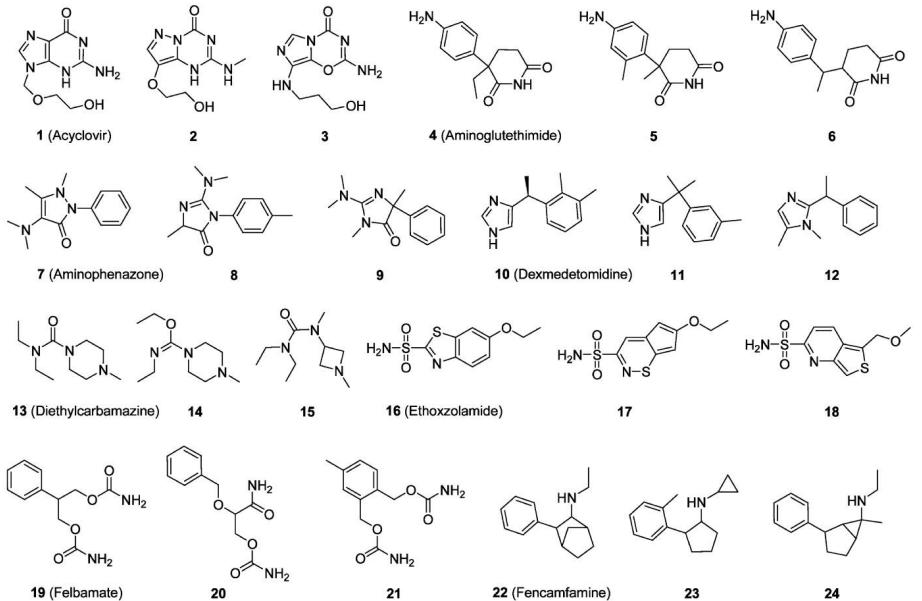


A.B. Wiltschko et al. Learning to Smell: Using Deep Learning to Predict the Olfactory Properties of Molecules. 2021

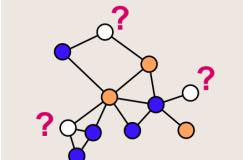
Aprendizaje profundo

Redes para grafos

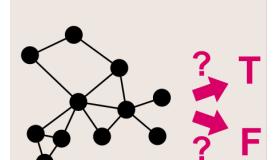
Estimación de propiedades de moléculas QM9



Node Classification



Graph Classification



QM9

[source]

```
spektral.datasets.qm9.QM9(amount=None, n_jobs=1)
```

The QM9 chemical data set of small molecules.

In this dataset, nodes represent atoms and edges represent chemical bonds. There are 5 possible atom types (H, C, N, O, F) and 4 bond types (single, double, triple, aromatic).

Node features represent the chemical properties of each atom and include:

- The atomic number, one-hot encoded;
- The atom's position in the X, Y, and Z dimensions;
- The atomic charge;
- The mass difference from the monoisotope;

The edge features represent the type of chemical bond between two atoms, one-hot encoded.

Each graph has an 19-dimensional label for regression.

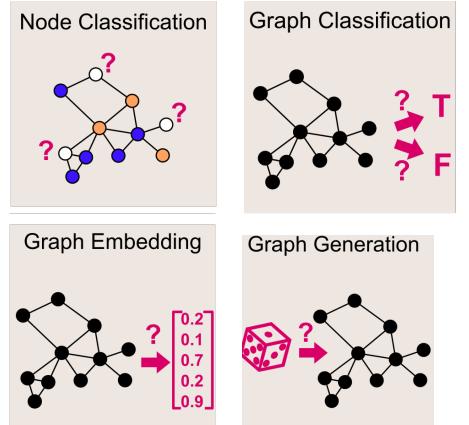
Arguments

- `amount` : int, load this many molecules instead of the full dataset (useful for debugging).
- `n_jobs` : number of CPU cores to use for reading the data (-1, to use all available cores).

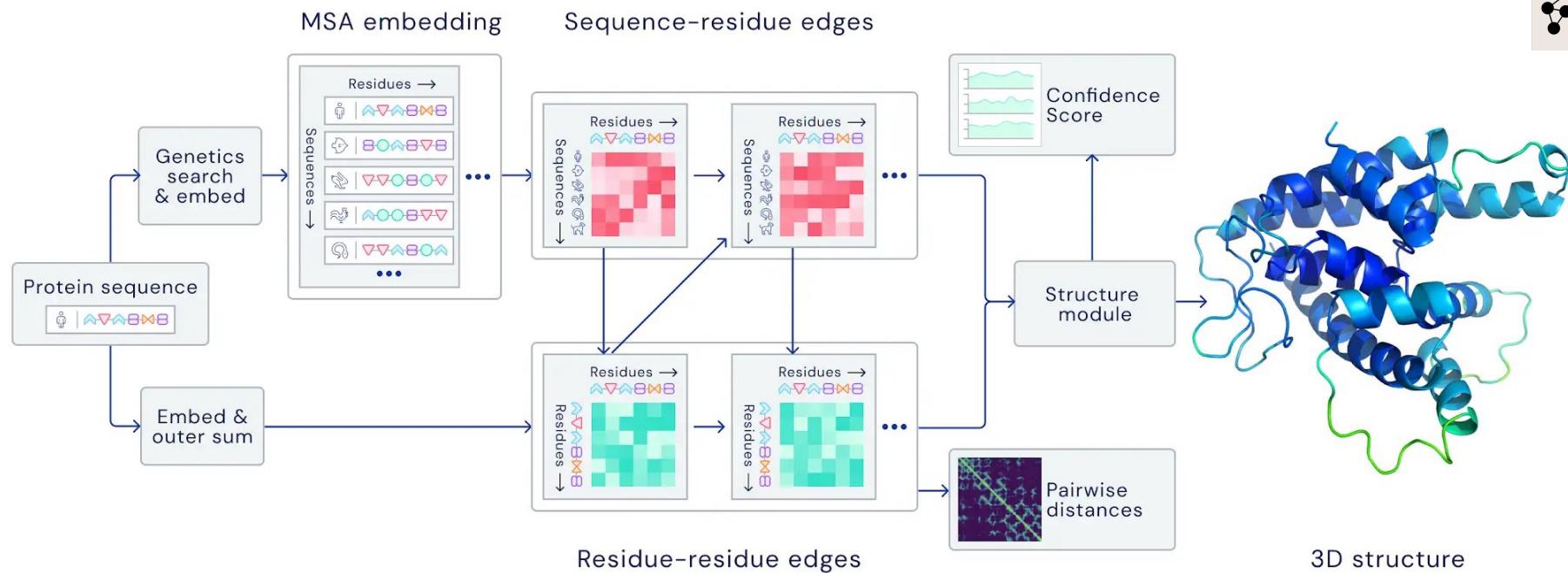
https://github.com/danielegrattarola/spektral/blob/master/examples/graph_prediction/qm9_ecc.py

Aprendizaje profundo

Redes para grafos



Plegamiento de proteínas con AlphaFold



Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021)

Limitaciones

- Disponibilidad de datos
- Sobreaprendizaje
- Incorporación de conocimiento sobre el dominio
- Transferencia de aprendizaje
- Explicabilidad
- Sesgos en el aprendizaje