

Wenance

Introducción a Tensorflow y Google Cloud ML Engine

Silvio Musolino, Bernabé Panarello

Table of contents

1. Explotación de datos - Conceptos
2. Aprendizaje Supervisado
3. Ejemplo práctico - Regresión Logística
4. Tensorflow

Explotación de datos - Conceptos

Explotación de datos - Conceptos

Data Mining

«es el proceso de análisis secundario de grandes bases de datos con el objetivo de encontrar relaciones insospechadas que sean de interés o valor» (Hand, 1998)



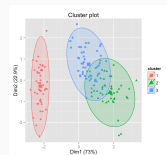
Machine learning

«es una rama de la IA que se ocupa del diseño y la aplicación de algoritmos de aprendizaje» (Mena, 1999)



Statistical Analysis

«metodología para extraer información de los datos y expresar la cantidad de incertidumbre en las decisiones que tomamos» (C. R. Rao, 1989)



Clasificación de algoritmos de Machine Learning

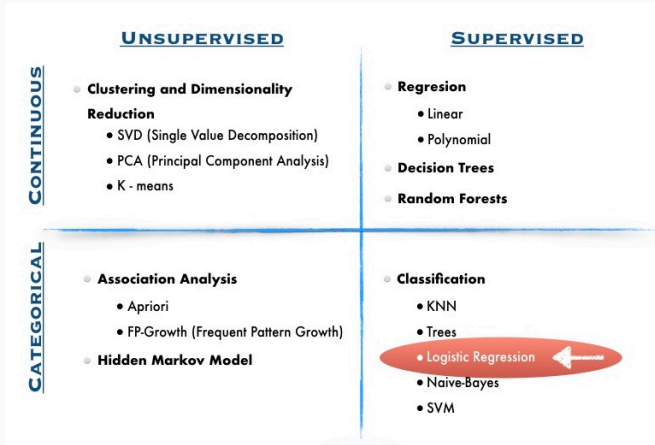
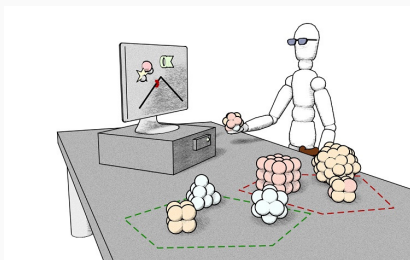


Figure 1: Machine Learning Algorithms

Aprendizaje Supervisado

Aprendizaje Supervisado

El *Aprendizaje Supervisado* es una técnica de machine learning que tiene el fin de deducir una función a partir de datos de entrenamiento.



Los datos de entrenamiento consisten de pares de objetos (normalmente vectores): una componente del par son los datos de entrada y el otro, los resultados deseados

Ejemplos - Taxi Fare Estimator

Dados los datos de a un viaje en taxi (origen, destino, hora, etc), predecir el costo del viaje

Yellow cab
TAXI SERVICE

Fare estimator

Results

Estimated Fare:
\$19.92

Per minute: \$1.00 (min)
\$20.00 (est. 20 min)

Call with Taxi:
\$12.00 (est. 12 min)
\$19.92 (est. 19 min)

Map:
How much does a taxi cost from Honolulu International Airport (HNL) to Waikiki in Honolulu, HI?

Services:
• Add \$1.00 for large items
• Add \$1.00 for each major surface
• See fare rules
• Cancellations (15 min)
• Tolls & surcharges may apply

Call Us:
1-800-555-1234

En este caso, X es un vector de n posiciones mientras que Y es un número real representando el valor del viaje

Ejemplos - Client Scoring

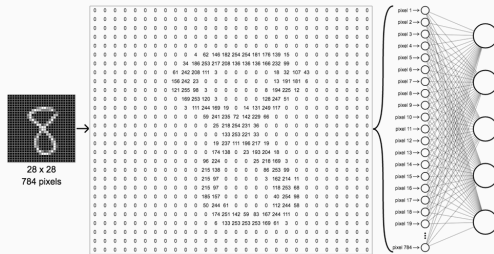
Dados los datos de un potencial cliente y un crédito, predecir si el cliente entrará en mora en los próximos 60 días



En este caso, X es un vector de n posiciones con los datos del cliente y el crédito mientras que y es un número real entre 0 y 1.

Ejemplos - Hand written digits

Dada una imagen de un dígito manuscrito, inferir de que dígito se trata



En este caso, si la imagen de entrada está en escala de grises de tamaño (width × height), representamos X como una matriz donde cada posición representa el valor de brillo de un pixel, y como salida se define un vector Y de dimensión k para la cantidad de clases a clasificar.

Ejemplo práctico - Regresión Logística

Modelo

- Consideremos el siguiente dataset:

Edad (x_1)	Sueldo(x_2)	Mora (0=No, 1=Si)
25	30000	1
24	40000	0
52	41000	1
23	50000	0
...

- Queremos aproximar $f(X) = y$, donde, aplicado a una sola muestra, X e y se interpretan como:
 - $X = [x_1, x_2] \in \mathbb{R}^2$
 - $y \in \{0, 1\}$

Clasificación Binaria - Regresión logística con modelo lineal

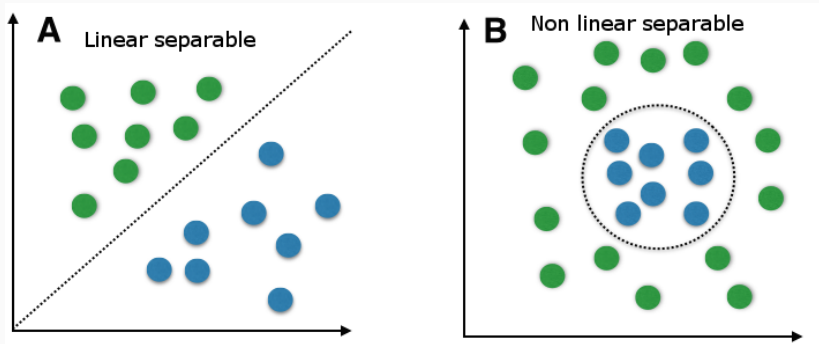


Figure 2: Linealmente separable (A) vs no linealmente separable (B)

Aprendizaje Supervisado - Modelo Lineal

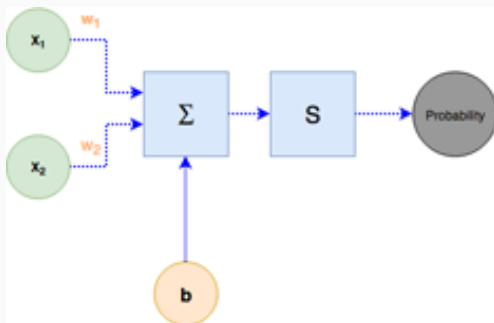


Figure 3: Grafo de cómputo regresión logística lineal (perceptrón simple).

- $X = [x_1, x_2]$, $y \in \{0, 1\}$
- $g_W(X) = \text{sigmoid}(w_1x_1 + w_2x_2 + b)$. Donde $W = \{w_1, w_2\}$ y b el término independiente .

Función *sigmoidea* o logística (función de activación)

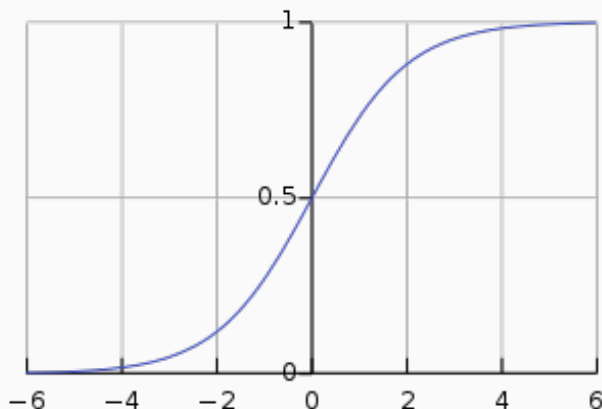


Figure 4: Gráfica de función logística $g(z) = \frac{1}{1+\exp(-z)}$.

Interpretación

- "Mapea" la salida de la función lineal en el intervalo $(0, 1)$
- Se puede interpretar como la probabilidad de que X sea de clase 1 (dado un W)
- Si la salida de $g(X) > 0.5 \rightarrow$ Predecir clase 1
- Si la salida de $g(X) < 0.5 \rightarrow$ Predecir clase 0.

Utilizando Tensores

¿Qué son? El nombre TensorFlow deriva de las operaciones que las redes neuronales realizan sobre arrays multidimensionales de datos. Estos arrays multidimensionales son referidos como "tensores"

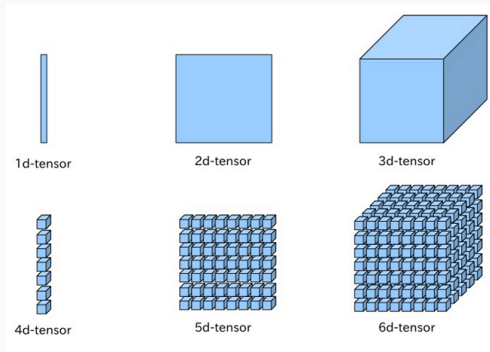


Figure 5: Representación gráfica de tensores de distintas dimensiones.

Continuando con el ejemplo ...

Generalización a n muestras

- Podemos generalizar nuestro modelo de ejemplo a notación de tensores, de manera que se pueda aplicar a un *batch* de n muestras.

$$X \in \mathbb{R}^{n \times 2} = \begin{bmatrix} x_1^1 & x_2^1 \\ x_1^2 & x_2^2 \\ \vdots & \vdots \\ x_1^n & x_2^n \end{bmatrix}, y \in \{0, 1\}^{n \times 1} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix}$$
$$(W^t \times X) \oplus b \in \mathbb{R}^{1 \times n}$$

- \oplus : Suma element-wise sobre dimensión principal (batch).

Error Cuadrático Medio (RMSE)

- dado el tensor $X \in \mathbb{R}^{n \times 2}$ y el tensor $Y \in \mathbb{R}^{n \times 1}$
- Se define el $RMSE_{\langle W, b \rangle}(X, y) = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(g_{\langle W, b \rangle}(X^i) - y^i \right)^2}$
- Mide cuanto el modelo (utilizando el conjunto de parámetros $(\{W, b\})$ "se equivocó" al predecir respecto de un batch del dataset.
- ¿A que recta van a corresponder los parámetros que minimicen el error?.
- Clase 0 $\rightarrow x_1 w_1 + x_2 w_2 + b < 0$.
- Clase 1 $\rightarrow x_1 w_1 + x_2 w_2 + b > 0$.

Minimización de la función de Error

- Es un problema de optimización.
- Queremos ajustar los parámetros W y b de manera que el error respecto al dataset sea mínimo.
- Para esto primero dividimos el dataset en 3 partes:
 - **Training**: Datos que se utilizan para ajustar los parámetros
 - **Validation**: Nos sirven como guía para ajustar hiperparámetros y comparar distintos modelos.
 - **Testing**: Utilizados para testear el modelo final y ver si generaliza bien.

Learning - Algoritmo Gradient Descent

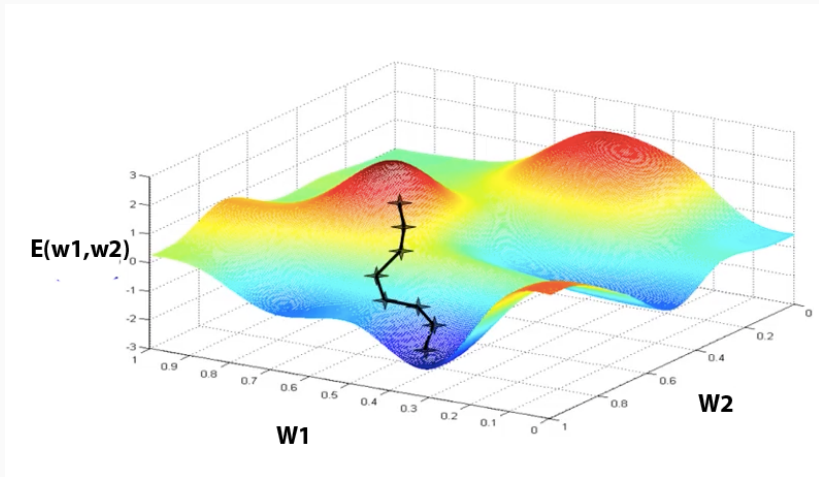


Figure 6: Superficie de error y descenso en dirección del gradiente (no se muestra la dimensión del término de *bias* b).

Learning - Algoritmo Gradient Descent (v. minibatch)

Entrada: $X \in \mathbb{R}^{n \times 2}$, $y \in \{0, 1\}^{n \times 1}$: Training set, donde n es el tamaño del conjunto.

1. Inicializar W y b (random)
2. Repetir hasta que el error sea aceptablemente pequeño (o algún otro criterio de parada)
 - 2.1 Repetir hasta que se acaben los datos de entrenamiento
 - 2.1.1 $(X^*, y^*) \leftarrow \text{nextBatch}(X, y)$ // Próximo batch de training data
 - 2.1.2 $W, b \leftarrow W, b - \alpha \cdot \nabla_{\text{error}<W,b>}(X^*, y^*)$

donde:

- $\nabla_{\text{error}<W,b>}(X, y)$: Es el *gradiente* del error respecto de de cada parámetro w_i y b . Es decir, cuanto "crece" el error respecto a cada parámetro ajustable.
- α : Learning rate.

Hay otros modelos mas complejos

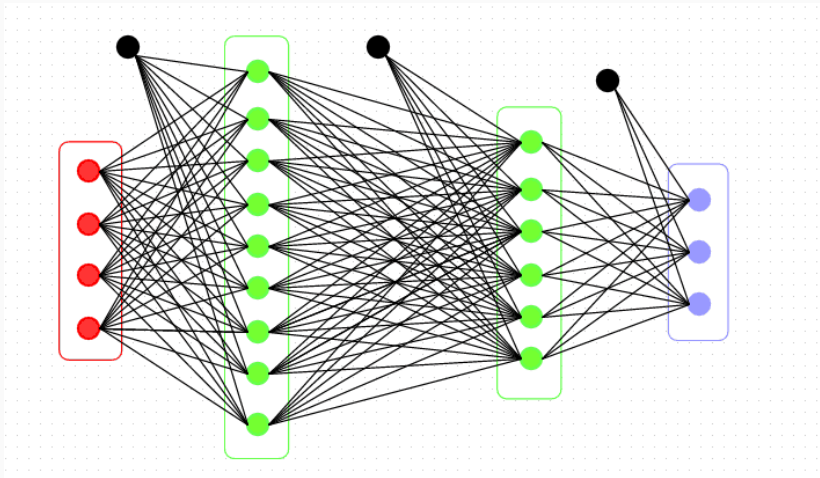


Figure 7: Perceptrón multicapa. Cada eje corresponde a un parámetro ajustable.

mas complejo todavía...

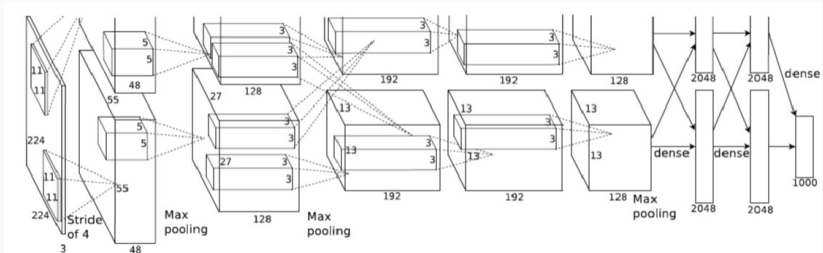


Figure 8: Arquitectura AlexNet, cerca de 62M de parámetros ajustables. (ver <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>)

Vanishing gradient problem

- Durante el entrenamiento el gradiente tiende a cero deteniendo el progreso del entrenamiento.

Overfitting

- El modelo se sobre ajusta a los datos de entrenamiento perdiendo capacidad en la generalización.

Exploding gradient problem

- Durante el entrenamiento, el gradiente diverge (NaN).

Overfitting

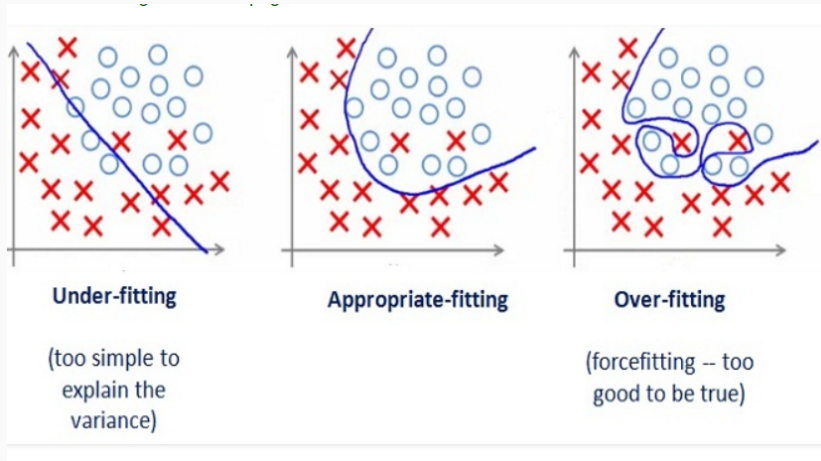


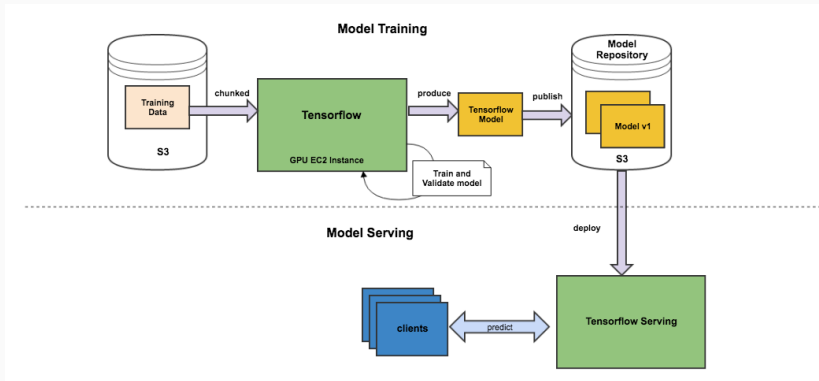
Figure 9: Underfitting y Overfitting.

Problemas de complejidad computacional

- A medida que aumentan la cantidad de nodos (neuronas), aumenta muchísimo ($O(n^2)$) la cantidad de conexiones (parámetros a ajustar).
- El cálculo del gradiente para modelos multicapa requiere la ejecución del algoritmo de back-propagation sobre un gran número de parámetros, lo cual es computacionalmente intensivo.
- Por ejemplo, entrenar una AlexNet con una nVidia Tesla k40c lleva alrededor de 5 días.

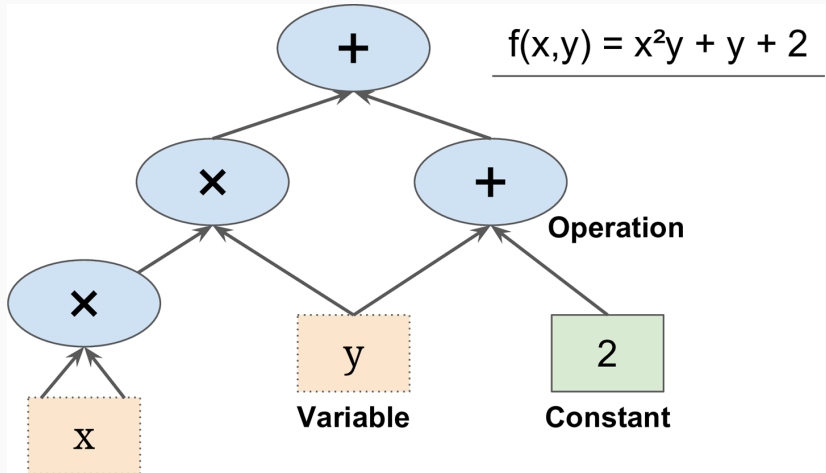
Tensorflow

Tensorflow - Modelo



Tensoflow - Grafos de cómputo

- Es una librería que permite definir grafos de cómputo sobre tensores y luego ejecutarlos.



Tensoflow - Grafos de cómputo (cont.)

- Los grafos de cómputo pueden ejecutarse fácilmente en paralelo utilizando procesadores diversos.

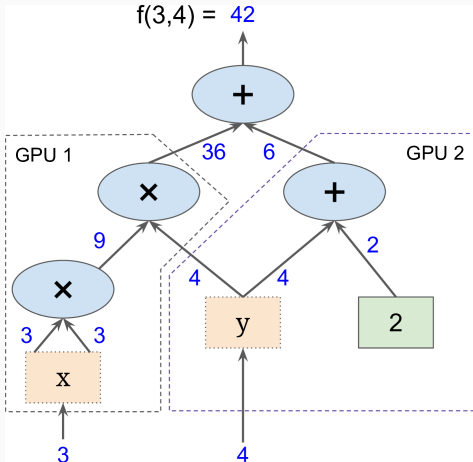


Figure 11: Ejecución en dos GPU.

Tensoflow - Operaciones sobre tensores

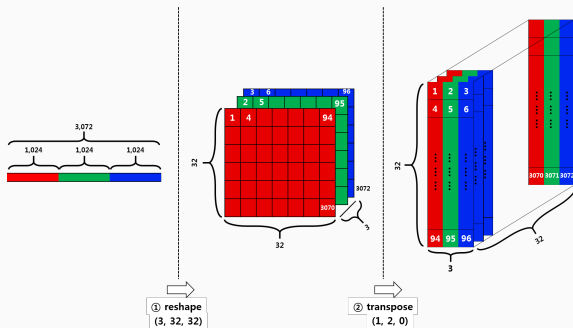


Figure 12: Operaciones reshape y transpose

MNIST - Modelo Deep Convolutional Neural Network

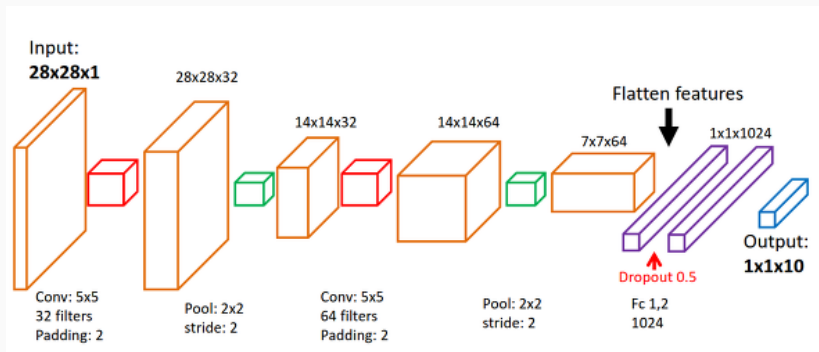


Figure 13: Arquitectura convolucional profunda para reconocimiento de dígitos manuscritos (MNIST)

¿ Preguntas ?

