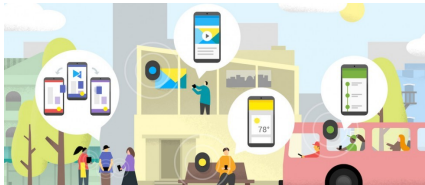


Transfer Learning and Data augmentation

Deep Learning summary



Marco Teran



**UNIVERSIDAD
SERGIO ARBOLEDA**

Noviembre 2020- Bogotá

¿Por qué usar redes neuronales?

- Disponibilidad de datos para entrenar redes.
- Capacidad computacional a menor costo y con mayor disponibilidad.
- Mejores algoritmos de entrenamiento.
- Gran comunidad que constantemente comparte desarrollos, nuevas arquitecturas o nuevas aplicaciones.
- APIs flexibles, simples y gratuitos.

Frameworks de Deep Learning

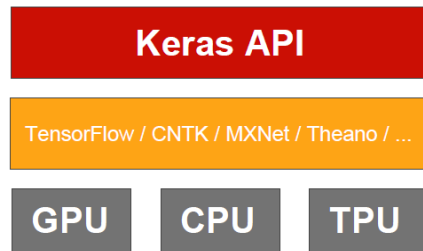
Frameworks de Bajo Nivel



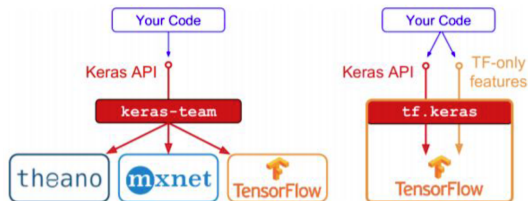
Frameworks de Alto Nivel



Keras API



Keras



Comparativo Frameworks

Framework/ Initial Release	Creadores	Escrito en qué Lenguaje?	Open Source	Soporta CUDA?	Desarrollo activo?	Modelos Preentrenados?
TensorFlow / 2015	Google Brain	C++, Python	Si	Si	Si	Si
Keras / 2015	Francois Chollet	Python	Si	Si	Si	Si
PyTorch / 2016	Facebook	Python, C	Si	Si	Si	Si
Caffe / 2016	Berkeley AI Research	C++	Si	Si	Si	Si
Theano/2007	Université de Montréal	C++/ Open CL	Si	Si	No	No
MATLAB+ Deep Learning Toolbox / 2016	MathWorks	MATLAB	No	Si	Si	Si

The Sequential API

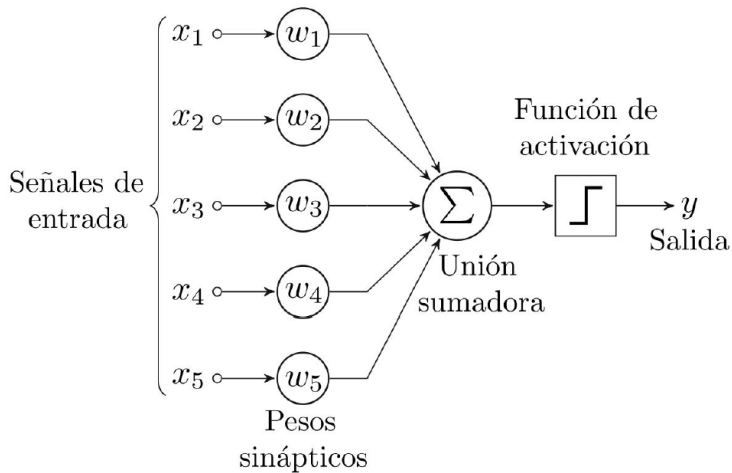
```
import keras
from keras import layers

model = keras.Sequential()
model.add(layers.Dense(20, activation='relu', input_shape=(10,)))
model.add(layers.Dense(20, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))

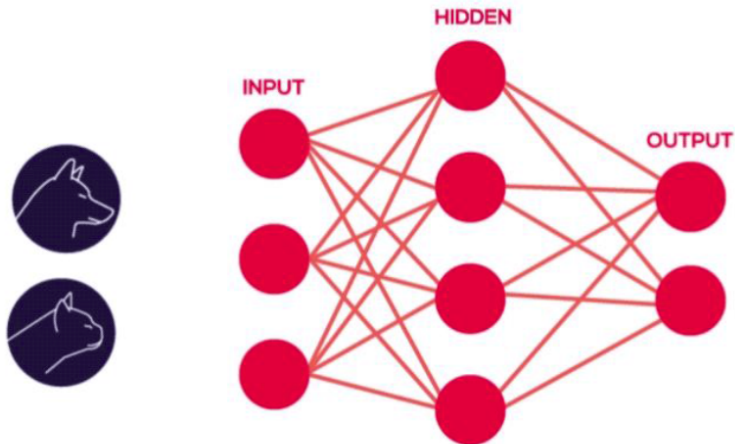
model.fit(x, y, epochs=10, batch_size=32)
```

Estructura de redes neuronales

Perceptrón



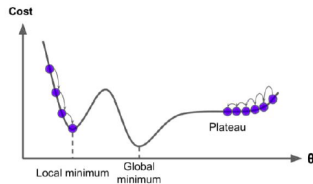
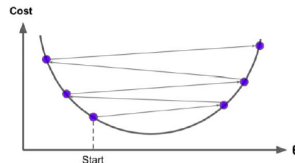
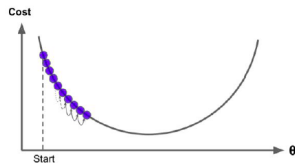
Multi-Layer Perceptron



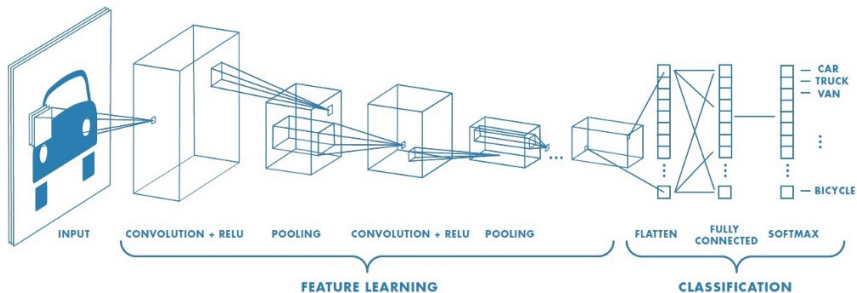
Comparación de funciones de pérdidas

Función de Pérdidas	Operación Residual	Robusto frente a Outliers
MAE (Mean Absolute Error)	Valor Absoluto	Si
MSE (Mean Squared Error)	Cuadrado	No
MAPE (Mean Absolute Percentage Error)	Valor Absoluto	Si

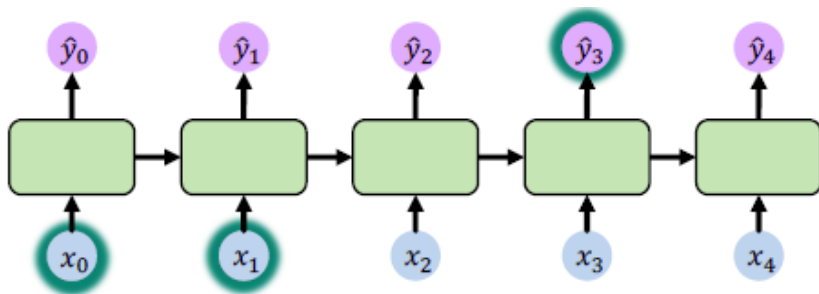
Gradiente descendiente



CNN

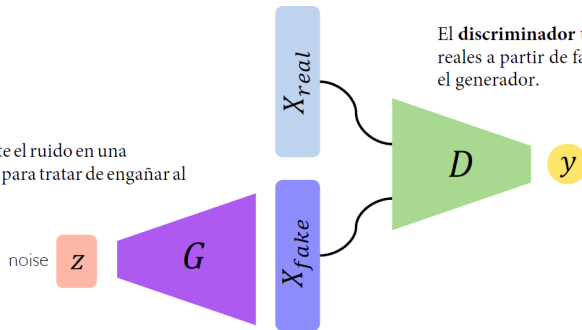


RNN



GAN

El **generador** convierte el ruido en una imitación de los datos para tratar de engañar al discriminador.

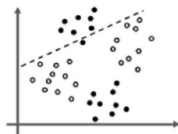
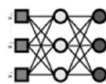


El **discriminador** trata de identificar datos reales a partir de falsificaciones creadas por el generador.

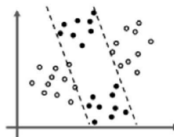
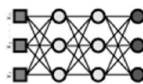
Ajuste de redes neuronales: hiper parámetros

Número Capas y Neuronas por Capa

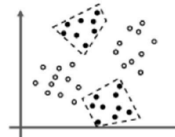
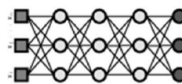
single layer



two layer



three layer

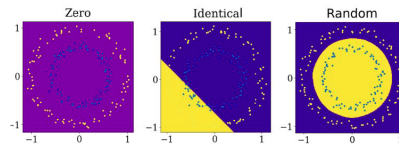
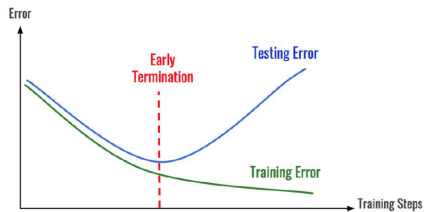


Cantidad Neuronas
por Capa =

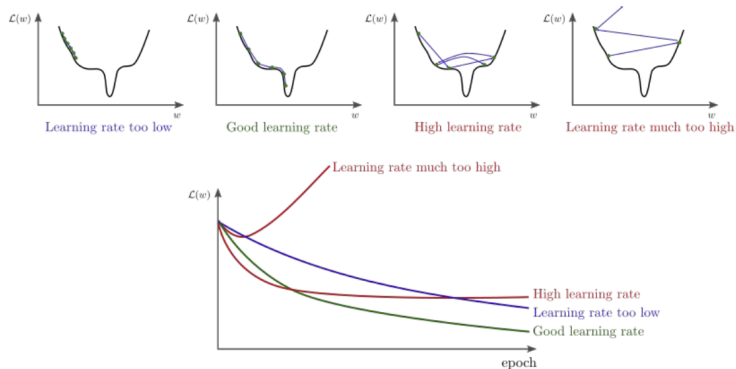
Potencia de 2 más cercana a 2^*
(Número de entradas)

Para 30 entradas $\mapsto 2 * 30 = 60 \mapsto 2^6 = \mathbf{64}$

Épocas e Inicializadores



Tasa de Aprendizaje

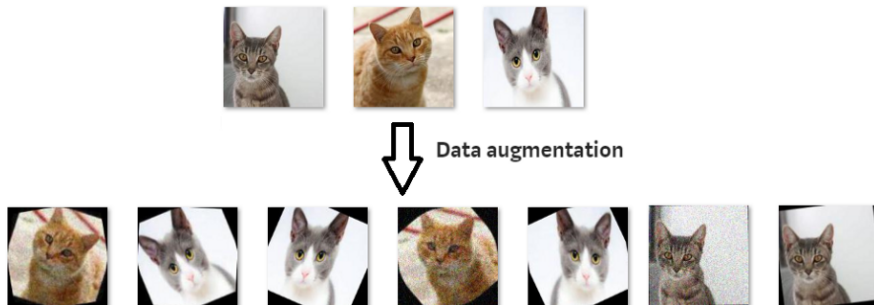


Función de activación y pérdidas

Tipo de Problema	Tipo de salida	Función de activación final	Función de pérdidas
Regresión	Valor Numérico	Linear	Mean Squared Error (MSE)
Clasificación	Salida Binaria	Sigmoid	Binary Cross Entropy
Clasificación	Única etiqueta, multiples clases	Softmax	Cross Entropy
Clasificación	Multiples etiquetas, multiples clases	Sigmoid	Binary Cross Entropy

Transferencia de aprendizaje y aumentación de datos

Aumentación de datos



Transferencia de aprendizaje

