

# Deep Learning

El '**DL**' es un método de '**Aprendizaje Automático Reforzado**' que busca simular al ser humano y sus distintas capacidades en la realización de tareas.

Se define un estado de inicio mínimo y aleatorio del modelo para que al inicializar aprenda por cuenta propia reconociendo patrones mediante el uso de muchas capas de procesamiento he iteraciones.

Al ser '**Reinforcement Learning**', el modelo aumenta su rendimiento tomando como base el resultado o la recompensa que se genera por cada interacción previa realizada.

## Objetivos

### Aprendizaje Supervisado

'Predecir' y/o 'Clasificar'

ANN

CNN

RNN

### Aprendizaje No Supervisado

Comprender he identificar  
'Patrones' subyacentes y  
'Relaciones' complejas.

SOM

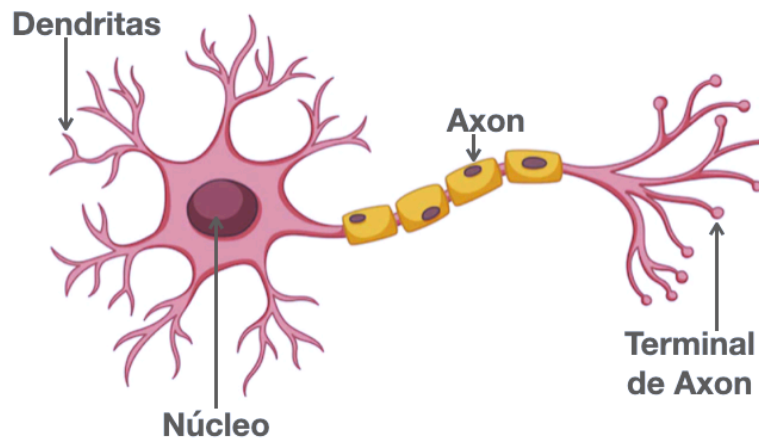
+BM

AEN

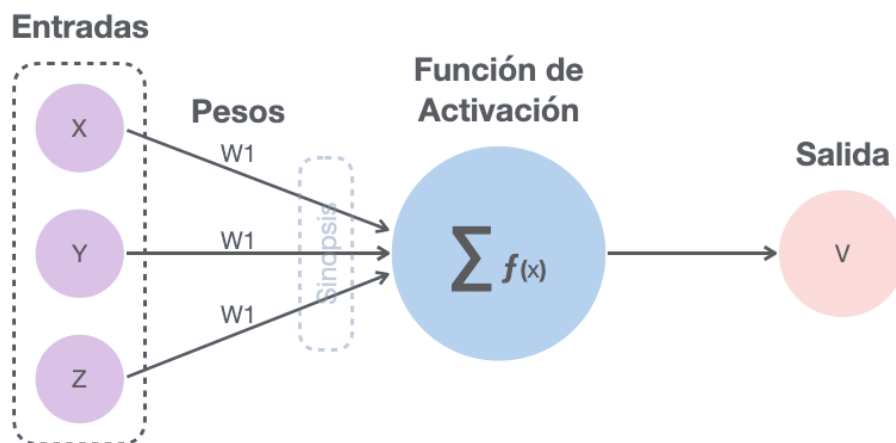
Un modelo de '**Deep Learning**' esta conformado por varias variables

- **Neurona  $\neq$  Perceptron** (modelo matemático de una neurona biológica)

## Neurona



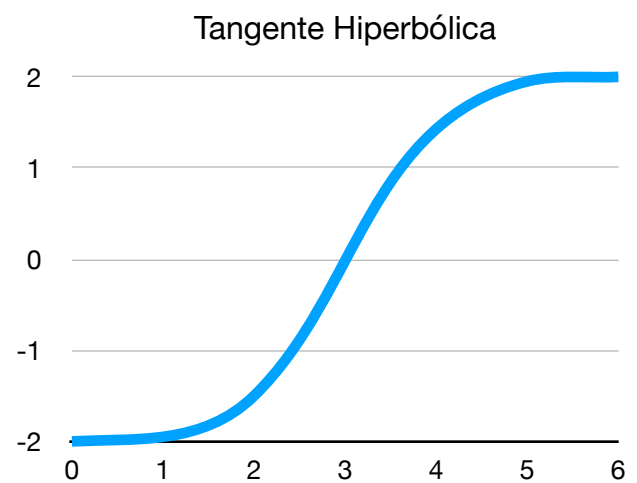
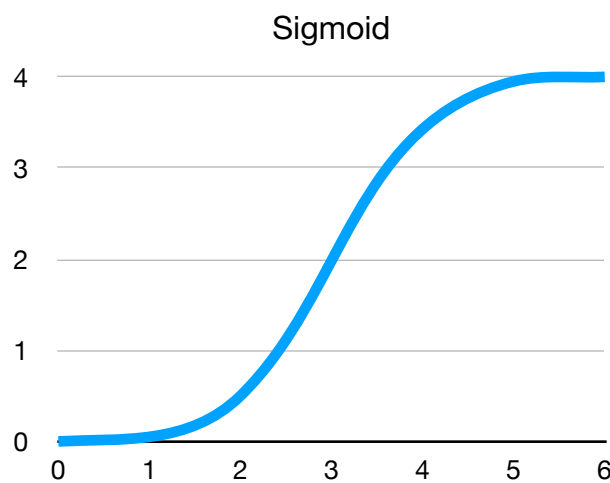
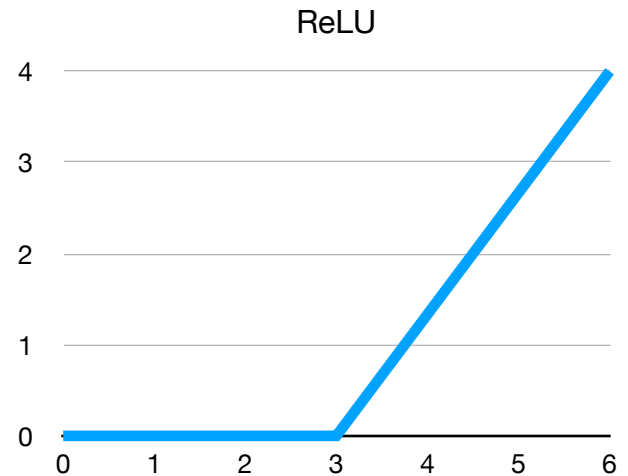
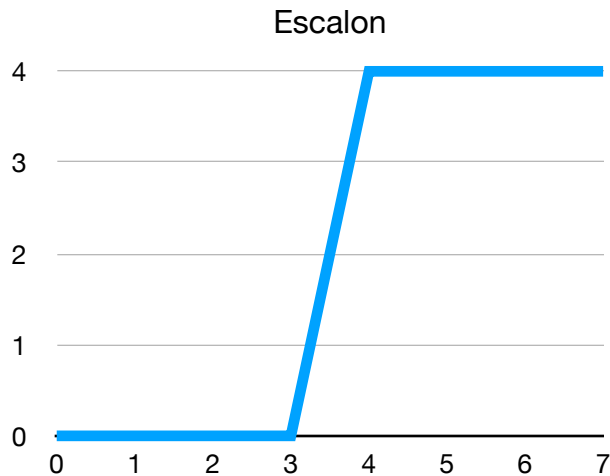
## Perceptron



- **Entradas:** valores numéricos comúnmente producto (salida) de la actividad de otras neuronas previas y/o del entorno (ambiente).
- **Pesos:** coeficientes (estadísticos) dinámicos que determinan la intensidad (pesos) de los valores de las '**Entradas**'.
- **Función de Activación:** formula matemática que procesa los valores de '**Entradas**' y da como producto un valor de '**Salida**'.
- **Salida:** valor de resultado de la red.

## - Función de Activación

- **Filtro y/o Umbral** que procesa los valores de las '**Entradas**' entregando un valor de '**Salida**' resultante.



## - Gradiente Descendente y Gradiente Descendente Estocástico

- Algoritmo matemática que permite encontrar el '**mínimo valor de pendiente**' de una función. Esta es la manera de hallar/ajustar los '**pesos**' óptimos en una **RNA** (Red Neuronal Artificial).
- Ambos algoritmo de **Gradiente** usan la '**Función de Coste**'.

# Función de Coste

## Función de Perdida (loss)

Evalúa los datos modelados para aprender a reducir el **'error'** en la predicción.

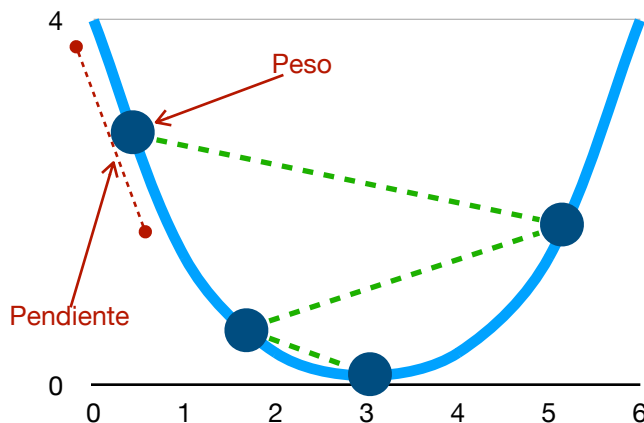


## Pendiente

La **'Derivada'** (inclinación) de la curva en un punto de la gráfica determinado.

Desde un **'peso'** (•) se puede identificar la **'dirección de corrección'** y un **'rango de ajuste'**. Este proceso **'itera'** hasta que el **'peso'** se aproxima bastante a 0 (eje Y).

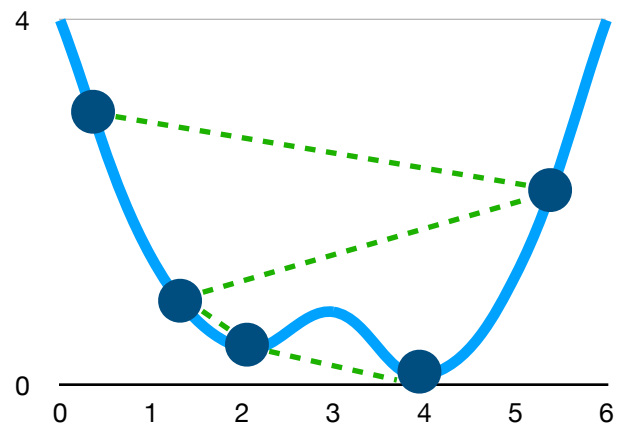
Gradiente Descendente



G.D. en **'lotes'** (por bloques): se calcula el **'Gradiente'** usando todas las muestras como un solo bloque por cada **'iteración'**.

**Problemas de estancamiento.**

Gradiente Descendente Estocástico



G.D. **'estocástico'** (aleatorio): se calcula el **'Gradiente'** al introducir cada una de las muestras a lo largo de cada **'iteración'**.

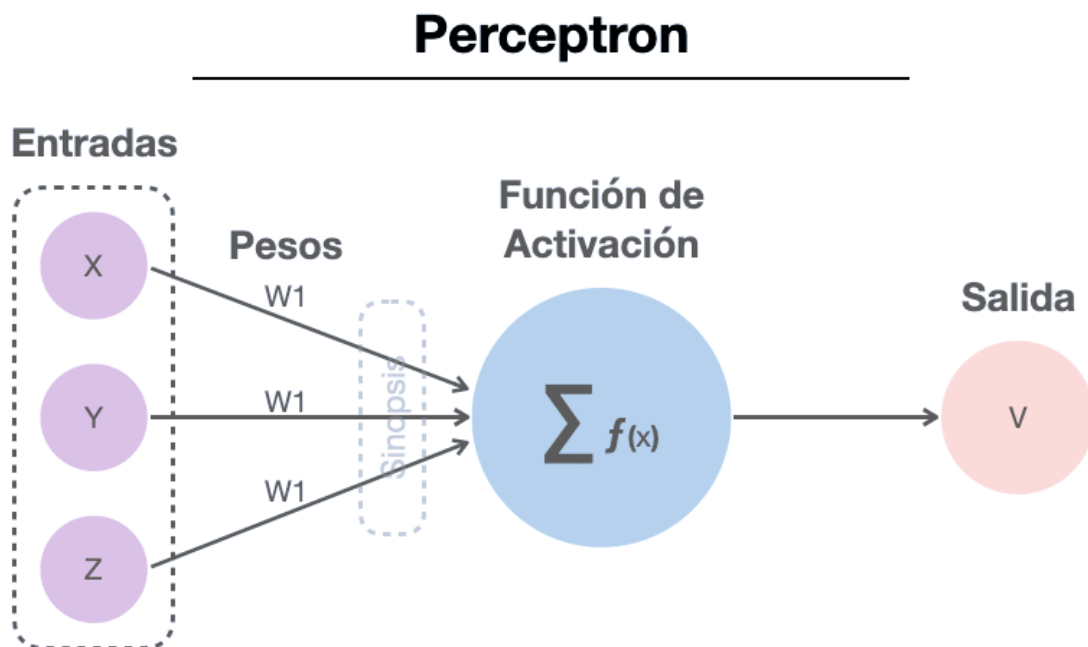
**Es mas lento y costoso.**

## - Back Propagation

- Algoritmo matemático que hace fluir la '**Información de Costos**' desde la '**Capa de Salida**' hacia las capas de atrás (**Capas Ocultas y Capa de Entrada**) para determinar el '**grado de error**' de los '**pesos**' de la **RNA (ANN)**.
- De acuerdo al '**grado de error**', se calcula un '**rango**' y '**dirección**' de *ajuste* para cada '**peso**' que permite ir re-ajustando los '**pesos**' de la **RNA** adecuadamente.
- El proceso de re-ajuste de '**pesos**' de la **RNA** se efectúa en cada iteración del proceso de '**training**' buscando lograr que todos los '**pesos**' (valores) se optimicen buscando que las '**pendientes**' se aproximen lo más posible a 0.
- Una vez que los '**pesos**' de la **RNA** hallan sido ajustados (optimizados), se somete a la **RNA** al proceso de '**testing**' para visualizar los estadísticos que revelen el grado (porcentaje) de efectividad y eficacia de nuestro modelo (**RNA**).

## - Perceptron

- Es un modelo matemático de una neurona biológica.
- Es '**UNA**' unidad (neurona) de una **Red Neuronal Artificial (RNA)**.
- Puede tener una '**Capa de Entrada**' con múltiples neuronas ingesting múltiples '**pesos**', pero solo una '**Función de Activación**' y una sola '**Salida**'.



## Artificial Neural Network (ANN)

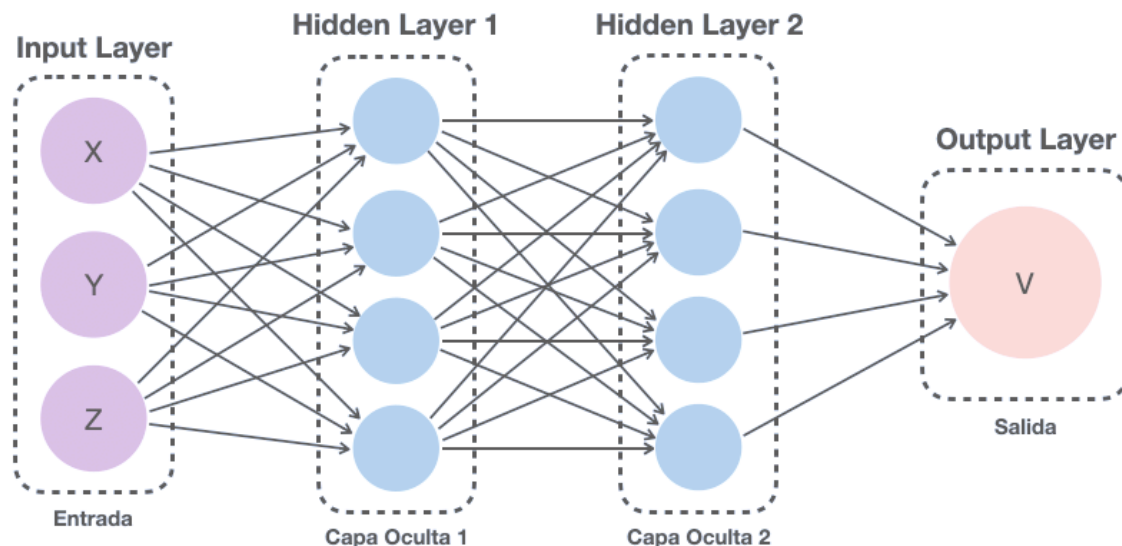
Método de **Artificial Intelligence** para hacer que una **ANN** (RNA) aprenda de una manera inspirada en cómo lo hacen los '**Humanos**'.

También emular al '**Cerebro Humano**' al procesar la información y efectuar acciones (actividades).

**Dificultad para identificar las formulas matemáticas adecuadas y estadísticos apropiados que el modelo necesita.**

**Alto poder predictivo, captura de relaciones complejas he identificación de patrones imposibles para los humanos.**

## Artificial Neural Network



- **Resolver** problemáticas con las **IT** (tecnologías de la información) de los negocios y **Automatizar** actividades.
- **Predecir** y **Clasificar** (segmentar) '**comportamientos**' y '**patrones**' en los datos (información).

# Convolutional Neural Network (CNN)

Método de **Artificial Intelligence** orientado al '**Reconocimiento de Patrones**' en '**imágenes**'.

**Convolution** se refiere a operaciones matemáticas de '**multiplicación**' con '**matrices**' (pueden ser de dimensiones diferentes) que *reducen* las dimensiones y *resaltan* los '**rasgos**' de las imágenes.

Pueden llegar a producirse demasiados parámetros, lo que dificulta en eficacia, el tiempo y costo.

Las conexiones internas (locales) de la CNN pueden reducirse, esto significa que se puede reducir la cantidad de parámetros.

## Secuencia de Funcionamiento

### Ingesta de Imagen

- Primero, se efectúa la **Convolución** (multiplicación de '**matrices**') a la imagen de entrada dando como resultado un set de la misma imagen con '**filtros**' diferentes que revelan todo tipo de '**rasgos**'.

### Primera Capa

- Luego, se aplican **Convoluciones** (multiplicación de '**matrices**') al set de imágenes con '**filtros**' reduciendo las dimensiones y generando un set de '**matrices**' (**Mapa de Características**) que conforman la primera capa llamada '**Convolutional Layer**'.

### Segunda Capa

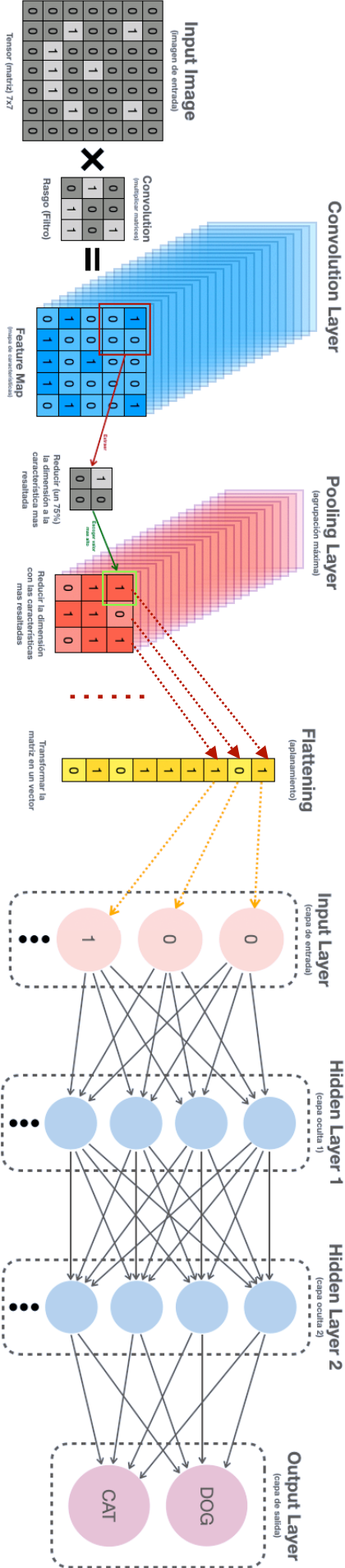
- Después, se efectúan **Funciones de Activación** (ReLU, etc.) y operaciones de **Max Pooling** (agrupar píxeles y preservar el máximo) a la capa '**Convolutional Layer**' reduciendo aun mas las dimensiones de las '**matrices**' del '**Mapa de Características**' y logrando resaltar los '**rasgos**' (características) en cada una de ellas conformando una siguiente capa llamada '**Pooling Layer**'.

### Capa de Entrada (Input Layer) de la ANN

- Finalmente, a cada matriz de la capa '**Pooling Layer**' se le efectúa el '**Flattening**' (aplanamiento) para convertir la '**matriz**' en un '**vector**' (lista/array) que se ingesta a la capa '**Input Layer**' de la **ANN** para que procese los valores de entrada de acuerdo a los pesos de la red.

# Convolutional Neural Network

## Convolutional Process





## - Imagen de Entrada

- **Matriz** : tabla bi-dimensional que puede sumarse y/o multiplicarse con otra tabla.
- **Convolución** : multiplicación de matrices (pueden ser de dimensiones diferentes).
- **Grafos** : conjunto de nodos (vertices) unidos por aristas.
  - También: se refiere a aplicar operaciones y funciones a los tensores.
- **Tensor** : matrices de N dimensiones.
- **Aumentar** dimensión : aumentar la dimensión de espacio de nuestro modelo, ejemplo: pasar de 2D a 3D ...
  - A mayor dimensión, mayor espacio para cubrir con los datos.
- **Reducir dimensión** : bajar de dimensión de espacio de nuestro modelo, ejemplo: de 3D a 2D ...
  - Al reducir las dimensiones también se reduce el espacio donde situar los datos.
- **Filtro** : Se trata de métodos para resaltar o suprimir, de forma selectiva, información contenida en una **imagen** a diferentes escalas espaciales, para destacar algunos elementos de la **imagen**, o también para ocultar valores anómalos.
- **Rasgo** : línea o forma trazada que es el valor mínimo de representación de dato que puede ser aprovechado como información. (un semi círculo, un contorno, etc.)

## - Convolution

- **Mapa de Característica** : set de imágenes que contienen diferentes características y patrones de una sola imagen original.
  - Se obtiene después de procesar con filtros a la imagen de origen y efectuar operaciones matemáticas (convoluciones y Funciones de Activación) para extraer todos los patrones y rasgos que la ANN pudo reconocer.

- **Funciones de Activación** : funciones matemáticas que procesan un dato de entrada y propagan el valor de salida hacia la siguiente capa.

#### - **Max Pooling**

- A una imagen se le efectúan convoluciones (multiplicación de matrices) para extraer un conjunto de pixeles en una matriz y **'preservar el máximo'** (el objetivo es generar otra matriz de la imagen solo con los pixeles mas altos de valor), esto permite eliminar el Overfitting.

#### - **Flattening**

- Convertir todas las **'matrices'** bi-dimensionales resultante en un solo **'vector'** (lista, array) que alimenta a la **ANN**.
- Esta capa es donde el **Modelo Convolutivo** se conecta con la **Red Neuronal**.

#### - **Softmax Function**

- Algoritmo matemático que evalúa (calcula) la probabilidad de eficacia del modelo de CNN normalizando los datos en un rango de  $[0, 1]$ .
- Da como resultado el porcentaje de eficacia de nuestro modelo al ser sometido al conjunto de **Testing**.

#### - **Cross Entropy**

- Algoritmo matemático que evalúa (calcula) la eficacia de nuestro modelo CNN solo evaluando las predicciones correctas.
- Evalúa : ¿ qué tan probable es que nuestra red neuronal se equivoque al hacer las predicciones ?.

## Métodos para calcular la eficacia de nuestro modelo CNN

Estos 3 métodos calculan de manera sutilmente diferente el porcentaje de eficacia de nuestro modelo al haber sido sometido al conjunto de **Testing**.

**CNN 1** ← ----- → **CNN 2**

0,33	<b>Error de Clasificación</b>	0,33
0,25	<b>Error Cuadrado Medio</b> (modelo de predicción)	0,71
0,38	<b>Entropía Cruzada</b> (modelo de clasificación)	1,06

A **Menor** los valores, **Menor** es el error.

Estos Métodos matemáticos evalúan (calculan) el error del modelo de CNN, y de acuerdo al ejemplo del curso, el método de Entropía Cruzada (Cross Entropy) fue el mejor en evaluar las dos CNN comparándolas.

La problemática trataba sobre un sistema de clasificación de imágenes (perro o gato), por lo cual se requería un modelo de validación adecuado para un modelo de clasificación,