Taller de machine Learning

Semana 02

Freddy Rospigliosi Cohaila

¿Qué son los datos de entrenamiento y prueba?

- **Datos de entrenamiento (train set)**
- Son los datos que usas para **enseñar al modelo**. Aquí el modelo:
- Aprende las relaciones entre las variables (por ejemplo, “más horas de estudio → mejor nota”).
- Ajusta sus **pesos internos** si es una red neuronal.
  - Ejemplo: Si tienes 100 estudiantes con sus horas de estudio y notas, puedes usar 75 para entrenar.

Datos de prueba (test set)

- Son los datos que **NO se usan para entrenar**. Sirven para:
  - Ver si el modelo **realmente ha aprendido** y no solo **memorizado**.
  - Evaluar si el modelo puede generalizar con **nuevos datos no vistos**.
  - Siguiendo el ejemplo: Los otros 25 estudiantes se usan para probar si el modelo predice bien sus notas, aunque nunca los haya visto antes.

¿Por qué es tan importante separarlos?

- Evita el sobreajuste (overfitting)
- Si solo usas los datos de entrenamiento, el modelo puede memorizar los ejemplos, pero fallar con nuevos datos.
  - Es como estudiar solo las respuestas del examen del año pasado. Puedes sacar 20 en ese examen, pero fallar en uno nuevo.
- **Permite medir el rendimiento real**
- Los datos de prueba **simulan la vida real**: nuevos casos que el modelo debe resolver bien, aunque no los haya visto.

Muestra completa

Aplico el modelo a datos que no he usado para entrenarlo

¿Qué es normalizar?

- Es transformar los datos numéricos para que estén en la misma escala.
- Ejemplo:

Edad (años)	Ingresos ($/mes)	Horas de estudio	¿Aprobó?
22	900	2	No (0)
35	4500	3	No (0)
28	1200	5	Sí (1)
40	6000	6	Sí (1)
25	1000	4	Sí (1)

¿Dónde está el problema?

- Observa los **rangos** de las variables:
  - Edad: entre **22 y 40**
  - Ingresos: entre **900 y 6000**
  - Horas de estudio: entre **2 y 6**
- Los ingresos tienen un rango mucho mayor. Si alimentas estos datos sin normalizar a un modelo como una red neuronal, el modelo le dará más importancia a los ingresos, aunque no sea la variable más relevante.

¿Qué pasa si no normalizas?

- El modelo puede:
  - **Aprender patrones erróneos** (darle más peso a ingresos por sus números grandes).
  - **Tener dificultad para converger** (en el caso de redes neuronales).
  - **Tener menor precisión** o comportamientos impredecibles.

- Datos normalizados: **Normalización Min-Max**
  - Escala los valores al rango [0, 1] usando:

Desnormalizado

Edad (años)	Ingresos ($/mes)	Horas de estudio	¿Aprobó?
22	900	2	No (0)
35	4500	3	No (0)
28	1200	5	Sí (1)
40	6000	6	Sí (1)
25	1000	4	Sí (1)

Normalizado

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Edad (Min-Max)	Ingresos (Min-Max)	Horas (Min-Max)	¿Aprobó?
0.00	0.000	0.00	0
0.72	0.698	0.20	0
0.24	0.058	0.60	1
1.00	1.000	1.00	1
0.12	0.020	0.40	1

Capa Dense (Densa o Totalmente Conectada)

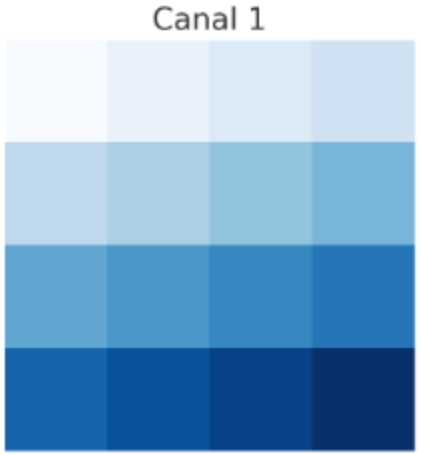
- ¿Qué es?
- Una capa Dense (también llamada fully connected) es aquella donde cada neurona está conectada con todas las neuronas de la capa anterior. Es una de las capas más comunes y fundamentales en redes neuronales.
- Se usa en redes clásicas (MLP) y como capa final en redes convolucionales.
- **¿Cuándo se usa?**
  - Para **clasificación o regresión**.
  - Como capa final en CNNs (redes convolucionales).
  - Cuando el modelo necesita aprender relaciones complejas entre características.

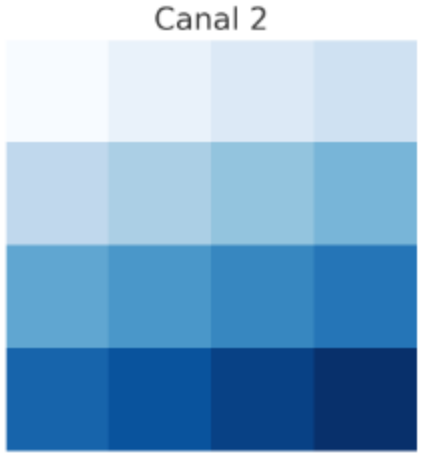
Capa Flatten (Aplanar)

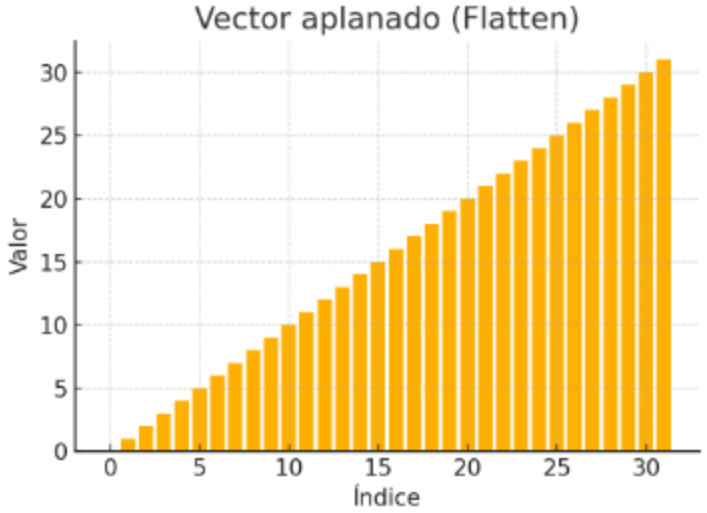
- Una capa Flatten convierte una entrada multidimensional en un vector plano (unidimensional).Es decir, “aplana” una estructura como una imagen 2D o una salida de convolución 3D

Ejemplo

- A la izquierda y centro tienes 2 canales (como salidas de una red convolucional con 2 filtros). Cada uno es una matriz 4×4.A la derecha, esa estructura se aplana en un único vector de 32 elementos (4×4×2), que luego puede conectarse a una capa Dense.







- Este paso es fundamental en modelos de imágenes, donde se extraen características visuales (convoluciones) y luego se usan para tomar decisiones (clasificación, por ejemplo).

Perceptrón MultiCapa – MultiLayer Perceptron (MLP)

- En su esencia, un perceptrón es el **modelo más simple de una neurona biológica** (de ahí que se le llame también "neurona artificial"). Su objetivo principal es realizar **clasificaciones binarias**, es decir, decidir entre dos categorías (por ejemplo, "sí" o "no", "perro" o "gato", "spam" o "no spam").
- Cómo funcionan: Son el tipo más básico de red neuronal. La información fluye en una sola dirección, desde la capa de entrada, a través de una o más capas ocultas, hasta la capa de salida. No hay bucles ni conexiones que regresen. Cada neurona en una capa está conectada a todas las neuronas de la siguiente capa.

Redes convolucionales- Convolutional Neuronal Network (CNN)

- Es un tipo de red neuronal diseñada para **extraer automáticamente patrones espaciales** (formas, texturas, bordes, etc.) de datos estructurados, como imágenes.
- A diferencia de las redes Dense, que conectan todo con todo, las CNN usan filtros que se mueven por la imagen para encontrar características locales.

Componentes principales de una CNN

- Capas convolucionales: Son como "detectives de características". En lugar de mirar la imagen completa, estas capas buscan patrones pequeños, como bordes, esquinas o texturas. Imagina que un detective busca primero la nariz, luego las orejas, y así sucesivamente, en lugar de intentar reconocer al animal de una vez.
- Capas de "pooling" (submuestreo): Después de que los "detectives" encuentran algunas características, estas capas reducen la cantidad de información sin perder lo importante. Es como hacer un resumen de lo que se encontró para no tener que procesar todos los detalles.
- Reconocimiento de patrones complejos: Al combinar estas capas, las RNC pueden aprender a reconocer patrones cada vez más complejos. Primero detectan líneas, luego formas, y finalmente objetos completos.
- Compartir pesos: A diferencia de otras redes, las RNC usan los mismos "detectives" (filtros) para buscar patrones en diferentes partes de la imagen. Esto las hace muy eficientes y buenas para encontrar la misma característica aunque esté en un lugar diferente de la imagen.

¿Dónde se usan las CNN?

- Clasificación de imágenes (gato, perro, etc.)
- Detección de objetos (auto, peatón, semáforo)
- Reconocimiento facial
- Diagnóstico médico con imágenes
- Lectura automática de matrículas (OCR)
- Video análisis y seguridad

Matriz de confusión

- Una **matriz de confusión** es una **herramienta fundamental para evaluar modelos de clasificación**. Nos permite ver en detalle **cómo de bien (o mal)** está prediciendo un modelo, comparando los valores **reales vs. los predichos**.
- Imagina que estás enseñando a una computadora a clasificar algo, por ejemplo, a reconocer si una foto es de un **perro** o de un **gato**. Una vez que la computadora ha intentado clasificar muchas fotos, necesitas saber qué tan bien lo hizo. Ahí es donde entra la **matriz de confusión**.
- Piensa en la matriz de confusión como una **tabla especial que resume los resultados de las predicciones de tu modelo de inteligencia artificial (IA)**, comparándolas con la realidad. Es una forma sencilla de ver cuántas veces tu modelo acertó y cuántas se equivocó, y lo más importante, ¡en qué se equivocó!

- **Verdaderos Positivos (VP):** Son las veces que tu modelo predijo correctamente que era un **perro**, y en la realidad, ¡realmente era un **perro**! (Acertó).
    - Ejemplo: La IA dice "perro" y la foto era de un perro.
- **Verdaderos Negativos (VN):** Son las veces que tu modelo predijo correctamente que **NO era un perro** (es decir, era un gato), y en la realidad, ¡realmente **NO era un perro**! (Acertó).
    - Ejemplo: La IA dice "gato" y la foto era de un gato.
- **Falsos Positivos (FP):** Son las veces que tu modelo predijo que era un **perro**, pero en la realidad, ¡era un **gato**! (Se equivocó, es un "falso alarma").
    - Ejemplo: La IA dice "perro" pero la foto era de un gato.
- **Falsos Negativos (FN):** Son las veces que tu modelo predijo que **NO era un perro** (es decir, era un gato), pero en la realidad, ¡sí era un **perro**! (Se equivocó, "se le pasó" un perro).
    - Ejemplo: La IA dice "gato" pero la foto era de un perro.

Video de redes convolucionales

- https://www.youtube.com/watch?v=4sWhhQwHqug