

Desafío 4

Juan está interesado en el mercado de la venta de ropa en línea. Su revolucionaria idea implica usar una inteligencia artificial que recomiende ropa que se parezca. Para poder realizar esto, Juan deberá primero clasificar los tipos de ropa. Para ello, se les ha pedido a los alumnos que implementen una inteligencia artificial que haga el trabajo.

Objetivos:

- Procesar la información otorgada para mejor trabajo posterior.
- Utilizar los conocimientos del módulo para la solución de un problema.

Actividad:

Se adjunta un [dataset](#) con **imágenes en escala de grises** que representan distintos tipos de ropas, junto con su etiqueta correspondiente. Para este trabajo se deberá:

1. Leer el Dataset.
2. **Implementar** (sin uso de librerías que lo hagan) y entrenar un perceptrón multicapa para la tarea de clasificación. Para esto se debe dividir los datos en entrenamiento y pruebas (70/30 sugerido).
3. Con los datos de prueba, calcule la precisión (accuracy) de la predicción.

Bonus:

1. Implemente una red neuronal convolucional para realizar el proceso de clasificación de [este dataset](#). Pueden revisar una implementación [aquí](#).

Información adicional:

A. "label" : etiqueta de la imagen. tiene 10 clases:

- 0 - Polera
- 1 - Pantalón
- 2 - sweater
- 3 - Vestido
- 4 - Saco
- 5 - Sandalia
- 6 - Camisa
- 7 - Zapatilla
- 8 - Bolso/cartera
- 9 - Bota

B. "pixel[1...784]": Pixel n de la imagen, el valor es entre 0 y 1, siendo 0 completamente negro y 1 completamente blanco.

Repositorio de git:

- Deben crear un repositorio git para trabajar en el desafío.
- El repositorio debe incluir un archivo README.md en el cual se indican claramente las instrucciones para ejecutar el programa.

Video explicativo (25% de la nota):

Además del código, en la entrega se debe adjuntar un link a un video en el cual el equipo explique la aplicación realizada. En 5-7 minutos, el vídeo debería incluir:

- Una breve descripción del problema
- Una explicación de los parámetros utilizados (learning rate, c/ neuronas y neuronas por capa, optimizador utilizado, etc)
- Un **pequeño análisis** explicando los resultados obtenidos
- Conclusiones

Coevaluación:

En el repositorio Git deben incluir un archivo llamado COEVALUACION.md. En este archivo deben:

- Ponderar el desempeño de cada integrante para el cálculo de la nota individual. Por ejemplo, Integrante A: 25% (trabajó menos), integrante B: 35%, integrante Z: 40% (trabajó más). Pueden usar [estos criterios](#).
- Indicar un aspecto positivo y otro a mejorar por cada uno de los integrantes.

Restricciones:

- La entrega se realiza a través del aula virtual.
- Deben enviar un link a un **repositorio git**.
- El repositorio debe incluir un archivo **README.md** indicando las instrucciones para instalar y ejecutar el código. Además se debe incluir una clara explicación del algoritmo propuesto.
- El repositorio debe incluir un archivo **COEVALUACION.md** indicando ponderación para cada integrante.
- Deben adjuntar **link** a vídeo explicativo de **no más de 7 minutos de duración**.
- La fecha de entrega es el día **domingo 1 de noviembre** a las 23:59 hrs.