

## Tarea 3

### PyTorch

El propósito de esta tarea es aprender a utilizar una librería especializada para redes neurales para generalizar y escalar lo realizado en la tarea 2.

Desarrolle un programa con PyTorch que permita crear redes neuronales para clasificar el conjunto de imágenes Fashion MNIST y cree con él cuatro redes neuronales.

### Requerimientos Funcionales

Las redes neuronales deben tener 784 neuronas de entrada, cada una correspondiendo a un píxel de una imagen de  $28 \times 28$ , y 10 neuronas de salida, cada una correspondiendo a una de las clases Fashion MNIST:

- |                          |                               |  |
|--------------------------|-------------------------------|--|
| 1. T-shirt/top (franela) | 6. Sandal (sandalia)          | 10. Ankle boot (bota hasta los tobillos) |
| 2. Trouser (pantalón)    | 7. Shirt (camisa)             |  |
| 3. Pullover (sweater)    | 8. Sneaker (zapato deportivo) |  |
| 4. Dress (vestido)       |                               |  |
| 5. Coat (chaqueta)       | 9. Bag (mochila)              |  |

Al igual que la tarea 2, el programa debe permitir crear una nueva red o cargarla de un archivo. Al crearla, el usuario tendrá la opción de especificar

- El número de capas ocultas (por defecto: 2)
- Si es convolucional o rectangular (por defecto: rectangular)
  - En caso de ser rectangular, el número de neuronas por capa (por defecto 512)
  - En caso de ser convolucional, deberá calcular el número de neuronas por capa usando interpolación lineal. (Con los valores por defecto, la capa de entrada tiene 784 neuronas, la capa de salida tiene 10, y hay 2 capas ocultas por lo que la primera capa oculta deberá llevar el número de neuronas un tercio del camino entre 784 y 10 – es decir 526 – y la segunda capa oculta deberá llevarla el tercio restante – es decir, 258 – logrando así que cada capa sucesiva disminuya en la misma cantidad de neuronas; véase la figura 1.)

Una vez creada, el programa debe permitir

- Entrenar la red por un número de épocas
- Probarla contra el conjunto de prueba
- Guardar la red en un archivo \*.pth

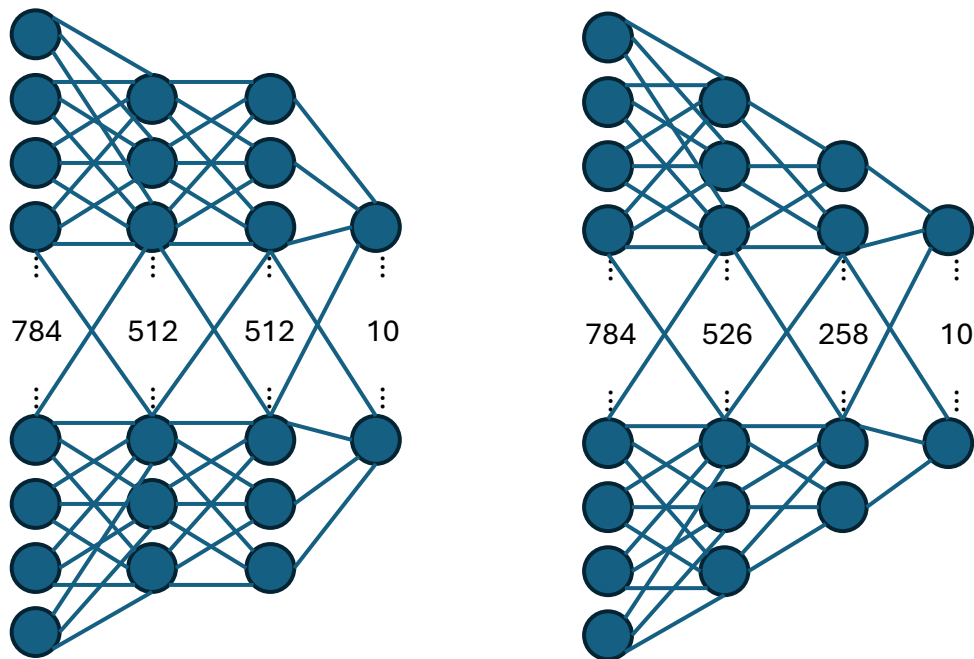


Figura 1. Arquitecturas de la red neuronal con dos capas ocultas, la cual puede ser rectangular (izq) ó convolucional (der). La conexión entre capas se muestra simplificada para efectos ilustrativos.

Luego de la primera opción, debe mostrarse el gráfico de la pérdida (*loss*) en cada época en un gráfico de línea. Luego de la segunda opción, debe mostrarse la matriz de confusión y las métricas de *precisión*, *accuracy* y *recall*.

Este menú debe volver a presentarse luego de terminar cualquiera de estas acciones, permitiendo seguir entrenando.

## Requerimientos Técnicos

Deberá entregar el programa y cuatro redes: una convolucional y una rectangular sobre la configuración básica del conjunto de datos Fashion MNIST, y una sobre una clasificación basada en cuatro clases: Top, bottom, footwear, bag, donde las clases se componen de las siguientes clases de Fashion MNIST:

- Top
  - T-shirt (franela)
  - Pullover (sweater)
  - Coat (chaqueta)
  - Shirt (camisa)
- Footwear (calzado)
  - Sandal (sandalia)
  - Sneaker (zapato deportivo)
  - Ankle boot (bota)
- Bottom
  - Trouser (pantalón)
  - Dress (vestido)
- Bag (mochila)

Nótese que para lograr este cometido, deberá variar el número de capas ocultas (y, en el caso de la red rectangular, nodos por capa) hasta encontrar una cantidad que produzca una buena precisión sobre el conjunto de prueba sin tener demasiadas neuronas ni requerir demasiadas épocas. Nótese que puede ser necesario introducir otras optimizaciones. Estas deben poderse habilitar como opciones en el menú.

## Rúbrica

- La red rectangular entregada clasifica correctamente el conjunto en las clases originales el conjunto de prueba: 25%
  - La gran mayoría de las veces (más de tres cuartas partes): 25
  - La mayoría de las veces (más de la mitad): 15
  - Mejor que al azar (más de la décima parte): 10
  - Peor que al azar (menos de la décima parte): 5
  - No ejecuta esta red: 0
- La red convolucional entregada clasifica correctamente en las clases originales el conjunto de prueba: 25%
  - La gran mayoría de las veces (más de tres cuartas partes): 25
  - La mayoría de las veces (más de la mitad): 15
  - Mejor que al azar (más de la décima parte): 10
  - Peor que al azar (menos de la décima parte): 5
  - No ejecuta esta red: 0
- La red rectangular entregada para las clases simplificadas clasifica correctamente el conjunto de prueba: 25%
  - La gran mayoría de las veces (más de cuatro quintos): 25
  - La mayoría de las veces (más de la mitad): 15
  - Mejor que al azar (más de la cuarta parte): 10
  - Peor que al azar (menos de la cuarta parte): 5
  - No ejecuta esta red: 0
- La red convolucional para las clases simplificadas clasifica correctamente el conjunto de prueba: 25%
  - La gran mayoría de las veces (más de cuatro quintos): 25
  - La mayoría de las veces (más de la mitad): 15
  - Mejor que al azar (más de la cuarta parte): 10
  - Peor que al azar (menos de la cuarta parte): 5
  - No ejecuta esta red: 0