

Consigna TP 1

Problema de clasificación:

<https://www.kaggle.com/t/5f2641ef201d4e8aa500d4876fb2b64e>

Se utilizará el dataset de Fashion MNIST, que se encuentra en la competencia de Kaggle ([link](#)) que quedará abierta hasta el final de la cursada y otorgará puntos extras a los grupos ganadores. La competencia puede hacerse por grupos (la cantidad de integrantes que se deseen pero se reparten los puntos al final).

La entrega del TP es individual.

La entrega del TP deberá ser estructurada de la siguiente forma:

- Jupyter notebooks con el informe/gráficos/código relevante/análisis, etc.
- Archivos helper con código irrelevante
- Todo esto subido a un repositorio personal de GitHub.

1- EDA: Visualizar algunas instancias de cada clase. Hacer histogramas de la distribución de intensidades para cada clase, cualquier otra propuesta es bienvenida.

2- Modelos: Entrenar un modelo de clasificación en las 10 clases de Fashion MNIST. Regresión softmax y MLP. En el caso que corresponda probar y comparar: distintas funciones de activación, distintos optimizadores, distintas funciones de costo, distintos tamaños de red (cantidad de capas y tamaño de cada capa), learning rate, batch_size, dropout rates, batch_normalization (si/no), inicializaciones de pesos (glorot_uniform, glorot_normal, normal(0,1), normal(0,0.001)).

Métricas a reportar para cada entrenamiento de interés: Accuracy como métrica principal. Como métricas secundarias: curva ROC, área bajo la curva ROC, F1-score, Precisión y Recall. Enunciar para el F1-score y el área bajo la curva ROC cómo se calculan los promedios macro y micro. Justificar cuál de los dos es el que corresponde reportar en este problema.

Para un buen modelo obtenido en el punto anterior (así si lo mejoran no tienen que repetir este punto), variar los hiperparámetros de a uno y graficar:

a- Accuracy vs [LEARNING_RATE, BATCH_SIZE, OPTIMIZADORES, ACTIVACIONES, DROPOUT_RATE, BATCH_NORMALIZATION, INICIALIZACIONES DE PESOS]

b- Iteraciones de entrenamiento vs [LEARNING_RATE, BATCH_SIZE, OPTIMIZADORES, ACTIVACIONES, DROPOUT_RATE, BATCH_NORMALIZATION, INICIALIZACIONES DE PESOS]

Intente justificar los gráficos obtenidos.

Valores mínimos esperados para cada modelo: SoftmaxReg: .70, MLP: .83

3. El mejor modelo debe ser reentrenado agregando una capa de tamaño 2 antes de la salida. Mapear los datos de entrada a la salida de la capa agregada y graficarla con los labels correspondientes. Superponer en el mismo gráfico los umbrales de decisión. Para obtener los umbrales de decisión se debe armar una grilla de puntos que debe darse como input a la última capa de la red neuronal (softmax) y tomar el argumento máximo como clase correspondiente.

Se alienta a intentar aumentar los datos de entrenamiento haciendo espejamiento de las imágenes en las situaciones que tenga sentido hacerlo.

Una vez entregado el TP se pueden seguir subiendo predicciones de modelos nuevos a Kaggle hasta el final de la cursada.

4- Para un buen conjunto de hiperparámetros de un MLP obtenido en 2, entrenar de la siguiente manera:

- Salvar la red inicializada con pesos aleatorios (antes de entrenar). La llamaremos red 1.
- Entrenar la red. La llamaremos red 2.
- Comparar los pesos obtenidos en red 2 con los pesos de inicialización (red 1) y marcar a aquel 50% que más haya variado.
- Sobre la red 1 eliminar los pesos que no hayan sido marcados en el punto anterior. La llamaremos red 3.
- Sin entrenar, medir el accuracy de red 3.
- Entrenar la red 3 (de 1/2 del tamaño original) y comparar la métrica con la métrica obtenida con red 2.
- Discutiremos los resultados en clase.

Problema de regresión:

Se trabajará con el dataset MNIST.

Se utilizará información adicional obtenida de <https://github.com/dccastro/Morpho-MNIST> (original MNIST morphometrics) la cual indica para cada imagen distintas características como ser largo del trazo, inclinación del número, ancho del trazo, etc.

Para acceder a los datos es necesario tener instalado tensorflow, descargados los archivos con los metadatos (disponibles en campus) y ejecutar:

```
from tensorflow.keras.datasets import mnist
import pandas as pd
(x_train, y_train), (x_test, y_test) = mnist.load_data()
train_morpho = pd.read_csv("train-morpho.csv")
test_morpho = pd.read_csv("t10k-morpho.csv")
```

En este problema intentaremos predecir la inclinación del número en función de:

- a. La imagen
- b. La imagen y el label al que pertenece (Utilizando embeddings de tamaño 2).
- c. La imagen y una variable continua más a elección (a partir de lo visto en el EDA)

EDA: Histogramas de inclinación para cada clase. Scatters entre la inclinación y los otros descriptores morfológicos).

Para b. graficar los embeddings obtenidos, intentar explicar lo obtenido.

Modelos: MLP con todas las opciones del punto 2.

Métricas a reportar: MAE

Fecha de entrega (15/5)
