

LABORATORIO DE DATOS

Primer Cuatrimestre 2024

Práctica N° 9: Descenso por gradiente y Redes Neuronales

Para realizar esta guía de ejercicios, descargar de la página de la materia el archivo `tf_regressor.py` para poder correr el siguiente comando

```
from tf_regressor import Regressor, train_test_split_scale_center
```

1. En este ejercicio utilizaremos el dataset `casos_coronavirus`.

- (a) Cargar el dataset y añadirle la columna `dias_transcurridos` con el índice de cada observación
- (b) Plotear `dias_transcurridos` vs. `confirmados_Nuevos`
- (c) Queremos armar un modelo de regresión que permita explicar la evolución de casos de coronavirus (Y) en función de los días transcurridos (X). Para esto se proponen los siguientes modelos:
 - i. $Y = b + w_0 X + w_1 X^2$
 - ii. $Y = b + w_0 X^{w_1}$
 - iii. $Y = b + w e^X$
 - iv. $Y = b + w_0 e^{w_1 X}$

Dividir el conjunto de datos en entrenamiento y testeo y decidir qué modelo resulta más adecuado. Utilizar `scikit-learn` para los modelos lineales y `Regressor` para los no lineales. En este último caso, se pueden usar de guía los modelos lineales para establecer valores iniciales de los pesos. Probar con distintas cantidades de épocas y valores iniciales para los pesos y el bias.

Obs: para el modelo iv) escribir `f` utilizando `np.e**(w[1]*x)` para $e^{w_1 x}$

2. En este ejercicio trabajaremos con el dataset `titanic` de `seaborn`

```
titanic = sns.load_dataset('titanic')
```

- (a) Limpiando el dataset:
 - i. contar cuantos `NaN` tiene cada columna, y en base a eso decidir qué columna del dataset descartar antes de ejecutar `.dropna()`
 - ii. graficar un boxplot de `fare` (precio del boleto), ¿qué se observa?
 - iii. explorar el método de pandas `quantile` para calcular el cuantil 0.99 de la columna `fare` y utilizarlo para eliminar las observaciones con outliers en esa columna.
- (b) Realizar regresión logística para predecir la variable binaria de supervivencia (`survived`) a partir del precio del boleto (`fare`). ¿Qué porcentaje de casos clasifica correctamente?
- (c) Repetir el item anterior, considerando la interacción de la suma de `fare` y `age` con `adult_male`. ¿Cuánto mejoró la precisión de la clasificación? ¿Qué se puede concluir a partir de la mejora en la precisión y del análisis de los pesos que el modelo otorga a cada variable?

- (d) Proponer un método que permita obtener una clasificación más precisa mediante regresión logística. Las demás columnas del DataFrame son:
- `pcclass` : clase en la que viajaba
 - `sibsp` : si viajaba con hermanos/as o cónyuges
 - `parch` : cantidad de hijos o padres con los que viajaba
 - `embarked` : donde se embarcó
 - `class` : nombre de la clase en la que viajaba (dato de `pcclass` en string)
 - `embark_town` : nombre del lugar donde embarcó
 - `alive` : si sobrevivió (mismo valor que `survived` pero booleano, por lo tanto no usar para predecir)
 - `alone` : si viajaba solo/a (es `True` si `sibsp = 0` y `parch = 0`)
3. Utilizando el dataset `diabetes.csv`, se quiere desarrollar un modelo para predecir si una persona tiene o no diabetes en base a las características descritas en el resto de las columnas.
- (a) Entrenar un perceptrón simple que permita categorizar a una persona como diabética o no diabética utilizando la sigmoidea como función de activación. ¿Cuántos falsos negativos hay? ¿Cómo se interpretan los pesos de la red entrenada?
- (b) Diseñar una red multicapa para la clasificación, utilizando la sigmoidea como función de activación. ¿Cómo se desempeña este modelo con respecto al anterior?
4. Probar que un perceptrón simple con la identidad como función de activación es equivalente a un modelo de Regresión Lineal.