Redes Neuronales – Proyecto 3

Perceptrón Multicapa
(*MLP – MultiLayer Perceptron*)

- Leer previamente el documento «*Preprocesamiento de datos*»

Objetivo 1:

Implementar el MLP en Python usando NumPy » ***Valor: 10 puntos***

- Se facilita más si lo implementan como una clase, pero no es obligatorio hacerlo.
- Capas y neuronas ocultas: Cualquier valor > 0.
- Algoritmo de retropropagación del error: Ecuaciones para la función sigmoide.
- Codificar las funciones de ***Scikit-Learn***: `fit()` y `predict()`.
- Codificar también las funciones: `forward()`, `sigmoid()` y `cost()`.
- Usar la tasa de aprendizaje: $\alpha \in (0, 1)$.
- Criterio de paro: Número de épocas.
- Inicializar pesos con valores aleatorios entre -0.01 y +0.01 (inicializar la semilla del generador de números aleatorios al valor ***42***).
- La función `fit()` deberá recibir como parámetros los conjuntos de entrenamiento y validación (y sus etiquetas): `X_train`, `y_train`, `X_valid` y `y_valid`.
- En la función `fit()`, además, deberán guardar y presentar en pantalla el valor del error cuadrático promedio *global* (***MSE*** – *Mean Squared Error* -la función de costo-) que el MLP comete en cada época, tanto para el conjunto de entrenamiento como el de validación (los van a graficar cuando el entrenamiento termine).

Objetivo 2:

Probar la implementación con el conjunto de datos Iris. ***Valor: 5 puntos***

- Deberán utilizar la codificación *OHE* (*One-Hot Encoding*) para las etiquetas.
- La capa de salida del MLP tendrá 3 neuronas para que, con la función de activación sigmoide, simulen la función *Softmax*, esto es, la neurona con el valor más alto determinará el valor 1 del OHE que el MLP está prediciendo.
- Presentar, con Matplotlib y en una sola gráfica, la evolución del error de entrenamiento y la del de validación (las épocas en el eje X vs. el MSE en el eje Y).
- Volver a clasificar los 150 patrones del conjunto de datos Iris con el MLP entrenado.
- Mostrar los resultados DE ESTA CLASIFICACIÓN graficando (de forma separada):
  - Longitud del sépalo (eje x) vs. longitud del pétalo.
  - Ancho del sépalo (eje x) vs. ancho del pétalo.
- Por último, indicar la exactitud global de la clasificación.

Objetivo 3:

Probar la implementación con el conjunto de datos ***Boston Housing***. ***Valor: 5 puntos***

- Mostrar los resultados graficando con Matplotlib:
  - En una sola gráfica las épocas (eje x) vs. MSE de la evolución del error de entrenamiento y la del de validación.

Puntos extras:

- Implementar el aprendizaje por gradiente estocástico con minibatches: ***3 puntos***.
- Implementar la verificación del gradiente (*Gradient Checking* o *Grad Check*): ***2 puntos***.

Notas:

- Para obtener los puntos del MLP deben probarlo con el conjunto Iris, el Housing o ambos.
- Trabajo en equipo de hasta 3 miembros.
- Quien obtenga el MSE mínimo establecerá la escala de calificación.
- En el reporte del trabajo deberán incluir la captura de pantalla (o la salida de texto) de la ejecución de su MLP con los conjuntos con los que lo prueben.

Límite de entrega

Miércoles 31 de octubre de 2018, 23:59:59.99 horas