



Universidad Simón Bolívar

EC5724 - Redes Neuronales y Aprendizaje Profundo

Enero - Marzo 2020

Actividad 4 (10 %)

1. Descripción

Para la Actividad 4, deben desarrollar redes neuronales utilizando PyTorch junto a la clase **`torch.nn.Module`**. Además, deben desarrollar cada paso del algoritmo de entrenamiento explicado en clases. Solo se permitirá utilizar un único ciclo (`for`, `while`, etc.), y se utilizará para iterar en el número de épocas del entrenamiento.

Esta actividad se realizará en parejas (para estudiantes de postgrado, se hará de manera individual).

1.1. Actividad 4

Deben entrenar a una red neuronal para que aprenda a diferenciar entre una mina y una roca. Para esta actividad se les proporcionará un archivo junto al enunciado, llamado `sonar.csv`, que contendrá el *dataset* que utilizarán. El *dataset* cuenta con 208 muestras tomadas con un sonar; cada una contiene 60 datos con valores entre 0 y 1 que indican la energía en diferentes bandas frecuenciales. Las etiquetas de cada muestra son: R para roca, M para mina.

Este dataset se encuentra ordenado (primero las rocas y luego las minas). Deben reorganizar aleatoriamente el orden de este dataset (deben guardarlo en un nuevo archivo `.csv`).

Luego, utilizando las clases `Dataset` y `DataLoader` de PyTorch, deben dividirlo en los tres conjuntos de entrenamiento estudiados: conjunto de entrenamiento (60 %), conjunto de validación (20 %) y conjunto de prueba (20 %).

Deben entrenar utilizando *Mini-batch Gradient Descent* (el `DataLoader` les va a facilitar esta tarea). Durante el entrenamiento, deben imprimir el número de la época y el valor de la función de costo para el conjunto de entrenamiento y el de validación (el costo de cada conjunto completo, no de cada mini-lote), junto con la métrica que usted considere más apropiada

(también para ambos conjuntos). Al final del entrenamiento, deben graficar la función de costo a lo largo de las épocas de entrenamiento.

Debe guardar los parámetros de su red neuronal en un archivo (utilizar `torch.save`) cada vez que se logre un máximo en la métrica deseada en el conjunto de validación.

Debe realizar este entrenamiento utilizando dos casos: sin *dropout* y con *dropout*.

Para el informe, deben colocar la matriz de confusión para los tres conjuntos de datos y los resultados de la métrica que usted considere más importante para este problema (indique el motivo). Deben colocar el *learning rate* utilizado y el número de épocas con la que entrenó a su red. Deben colocar la arquitectura de red neuronal utilizada. Deben anexar la gráfica de la función de costo para los conjuntos de entrenamiento y validación. Deben realizar todo esto para los dos casos mencionados anteriormente.

2. Entregables

Creen una carpeta con el nombre **carnet1_carnet2**. Deben subir un archivo **.zip**, con el nombre **carnet1_carnet2.zip** (para postgrado, solo un carnet) que contenga los siguientes archivos:

1. **actividad_4.py**
2. **shuffled_sonar.csv** (*dataset* reordenado)

Si van a utilizar archivos `.py` adicionales, subánlos sin ningún formato en específico. Cada uno de los archivos listados debe contener la solución de su actividad correspondiente, es decir, si ejecuto un archivo, voy a ver únicamente la solución de esa actividad.

Además del archivo **.zip**, debe anexar un informe con el nombre **carnet1_carnet2.pdf**, que contenga las especificaciones mencionadas anteriormente en la descripción de cada actividad. Cualquier incumplimiento del formato de entrega será penalizado con un cuarto de los puntos de esta evaluación. Cualquier indicio de plagio será llevado a coordinación.

Deben entregar esta actividad antes del viernes 28/02/2020, hora 11:59 P.M.