CC5114 - T1

Nombre: Ariel Suil A.
Repositorio: asuil/CC5114-Neural
Profesor: Alexandre Bergel
Auxiliar: Ignacio Slater M.
Entrega: 12/10/20

1. Datos

Los datos utilizados corresponden a señales recibidas por radar al medir electrones libres en la ionosfera, y su clasificación binaria "good" o "bad" especificando el éxito de la lectura.
Fuente: machinelearningmastery

1.1. Estructura

Los datos cuentan con 17 lecturas de dos parámetros cada una (34 columnas numéricas) y 1 columna de un carácter ("g" o "b") completando así 35 columnas. Por otro lado se cuenta con 351 mediciones clasificadas, es decir 351 filas.

1.2. Limpieza

La segunda columna del dataset era exactamente igual a 0 en todas las mediciones, por esto para evitar ruido en el modelo se decidió eliminar esta columna (al no ser testeado no sabemos qué hará el modelo si recibe un valor diferente a 0, además siendo siempre 0 en la data de entrenamiento no ayuda a diferenciar las clasificaciones).

Algunas columnas numéricas tenían sus datos entre -1 y 1, mientras que otras entre 0 y 1, por lo que se normalizó todas las columnas para estar en el rango estándar [0, 1].

Se utilizó One Hot Encoding en la output del experimento para convertir una salida "g" o "b" en dos columnas que representan con un 0 o 1 las outputs previamente mencionadas.

Los datos no tenían valores faltantes por lo que no fue necesario rellenar/eliminar filas. Dado que se trabajó con floats no se revisó la existencia de filas repetidas, esto podría afectar el bias del modelo. El data set venía ordenado intercalando entre clasificación "g" y "b" por lo que al hacer nuestra partición de training nos aseguramos de tener la misma cantidad de ambas y así disminuir el bias. El dataset contaba con más clasificaciones "g" que "b" pero las restantes se encontraban al final por lo que solo se utilizaron para testing.

2. Red Neuronal

El modelo entrenado es el proporcionado por el profesor, excepto por un cambio para pasar una variable m como parámetro de las funciones calculate_cost y backward_prop debido a un error de scope al importar; y una modificación en predict para retornar múltiples valores, ya que se trabajó con dos outputs (one hot encoding).

3. Entrenamiento

El entrenamiento se realizó con el la función entregada por el profesor, en un comienzo se realizó con una división 80/20 en training y testing pero debido a los buenos resultados se decidió disminuir la data de entrenamiento dejando una división 50/50 para priorizar un mejor análisis del modelo.

4. Resultados

Al realizar los experimentos se fueron guardando las outputs para luego ser comparadas con los valores esperados. Las outputs originales eran un par de floats, pero utilizando un threshold de 0.5 se discretizaron a 0 o 1 en la función predict. Dado que las outputs son independientes entre sí, es posible obtener resultados [1, 1] o [0, 0] donde la red neuronal no presenta ninguna preferencia sobre la clasificación, estos escenarios se mencionan como "no determinados".

4.1. Cálculos

Los cálculos realizados fueron un conteo de TP, FP, TN y FN para la Matriz de Confusión, además de las métricas de precisión y recall utilizando los conteos mencionados. Al realizar los cálculos de recall se consideraron los valores totales en vez de la suma de "g" y "b", es decir, se consideraron los valores no determinados como resultados erróneos.

4.2. Presentación

Para la presentación de resultados solo se configuró un print debido a que no había mucho lugar para un gráfico más elaborado, a continuación se adjunta el resultado esperado para la seed ingresada.

```
           | pred 'g'   | pred 'b'  | total
actual 'g' | 128        | 8         | 137
actual 'b' | 3          | 34        | 39
not determined: 3

Results for 'g' class
precision: 0.9770   recall: 0.9343
Results for 'b' class
precision: 0.8095   recall: 0.8717
```

Figura 1: Resultado esperado