

Tarea 1

Aprendizaje con Redes Neuronales

Alumno: Gonzalo Uribe
Profesor: Alexandre Bergel Profesor 2
Auxiliar: Juan-Pablo Silva
Ayudantes: Alonso Reyes Feris
Gabriel Chandía G.

Fecha de entrega: 9 de noviembre de 2018
Santiago, Chile

1. Dataset

El set de datos con el que se trabajó fue obtenido del Machine Learning Repository de la Universidad de California¹. Este consiste en los resultados del análisis químico de vinos cultivados en la misma región de Italia, pero en tres diferentes viñedos. El análisis calculó la cantidad de 13 diferentes constituyentes que se encontraron en cada uno de los tres tipos de vino.

Estos constituyentes son:

- Alcohol
- Ácido Málico
- Ceniza
- Alcalinidad de la ceniza
- Magnesio
- Fenoles Totales
- Flavonoides
- Fenoles no flavonoides
- Proantocianidinas
- Intensidad del color
- Matiz del color
- OD280/OD315 de los vinos diluidos
- Prolina

Cabe mencionar que el autor del dataset señaló que este se “comporta bien” al intentar clasificarlo, por lo que es una buena set de prueba para redes neuronales.

2. Instrucciones de Ejecución

El programa que se utilizó para el estudio del dataset fue hecho en Python 3, su única dependencia externa es la librería matplotlib que se utilizó para generar los gráficos con los resultados.

El repositorio de Github donde se encuentra el proyecto está en <https://github.com/Mortup/RedesNeuronales>

Para probar el programa en Ubuntu se pueden utilizar los siguientes comandos:

¹ <https://archive.ics.uci.edu/ml/datasets/Wine>

```
git clone git@github.com:Mortup/RedesNeuronales.git
cd RedesNeuronales
python3 -m venv venv
source venv/activate/bin
pip install matplotlib
cd tarea1
python main.py 3000
```

Al ejecutar el programa se debe pasar como argumento el número de epochs que quieren realizarse.

Para ejecutar un set de unit tests simplemente se debe llamar python y el nombre del test. Por ejemplo: `python test_neurons.py`

3. Resultados

Se realizó un primer entrenamiento para tener como referencia con una red con dos capas ocultas y un learning rate de 0.01. Como se puede ver en la figura 1 se puede considerar que la red aprendió a clasificar los diferentes tipos de vino al rededor de la epoch 1500.

Para realizar todas las pruebas que se presentarán a continuación se tuvo la precaución de usar la misma red neuronal y usar la misma semilla para el generador de numeros aleatorios, de esta manera se evita que hayan factores externos alterando los resultados.

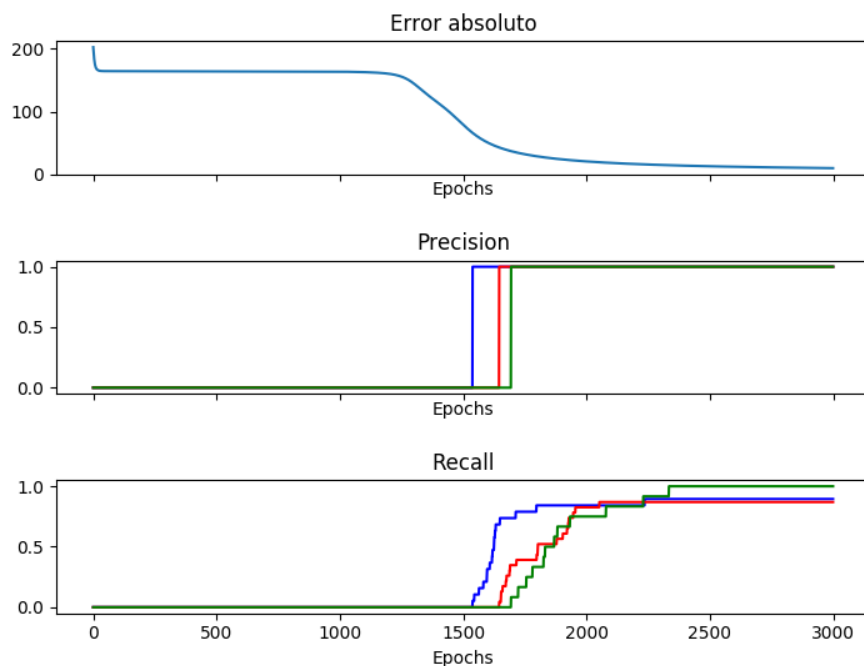


Figura 1: Aprendizaje con 2 capas ocultas y learning rate = 0.01. Las diferentes líneas en precisión y recall representan la precisión y recall de cada una de las tres categorías de vinos.

3.1. Impacto del número de capas ocultas

Para evaluar los cambios producidos por el número de capas ocultas se compararon tres diferentes redes neuronales.

Como se puede ver en la figura 2 para este dataset en particular las capas ocultas solamente retrasan el aprendizaje de la red. De esto se puede concluir que es necesario limitar el número de capas ocultas dependiendo de la complejidad del problema que se intenta resolver, como en este caso el dataset es muy fácil de clasificar conviene simplemente no usar capas ocultas.

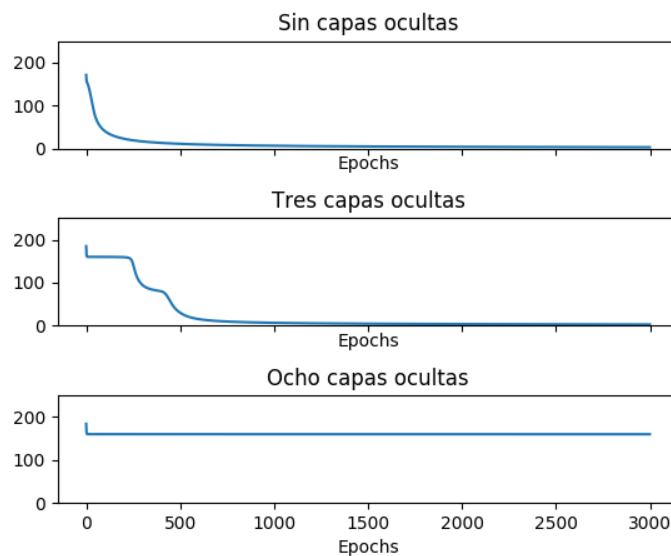


Figura 2: Aprendizaje con diferentes cantidades de capas ocultas.

3.2. Variaciones de Learning Rate

Para evaluar el efecto del learning rate en la velocidad de aprendizaje de la red primero se hizo una comparación con tres learning rate diferentes: 0.01, 0.1 y 0.5. Como se puede ver en la figura 3, no hay espigas ni cambios súbitos en el error.

Por esto se consideró una buena metrica el revisar cuanto tarda cada learning rate en obtener un error menor que 25 por primera vez.

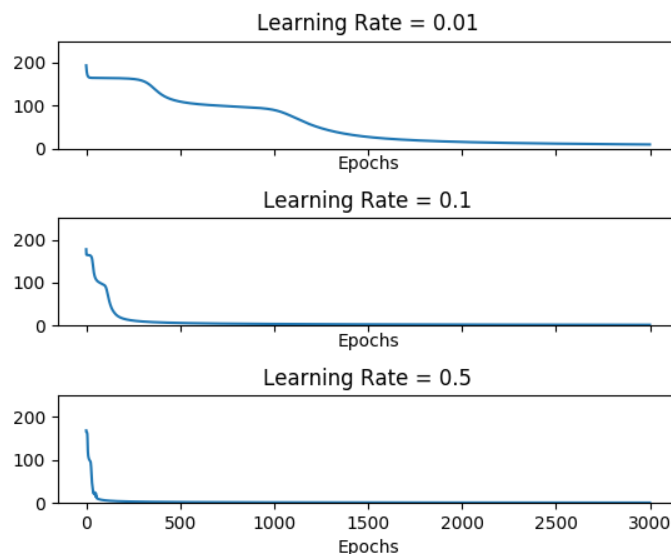


Figura 3: Aprendizaje con diferentes learning rates.

Como se puede ver en la figura 4, donde se muestran las iteraciones necesarias para converger en función de los diferentes learning rates, este disminuye exponencialmente la cantidad de epochs necesarias. Se puede ver también que es difícil bajar de las 25 iteraciones, pues se puede observar un comportamiento asintótico en la función de la figura.

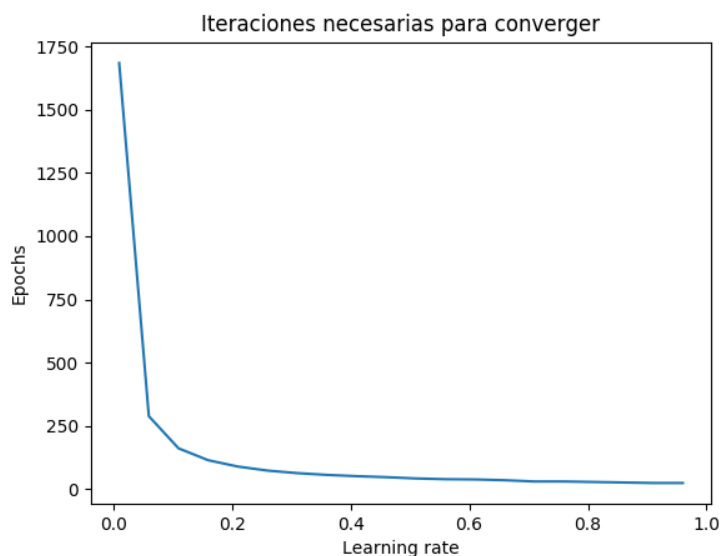


Figura 4: Epochs necesarias para converger.

Hay que tomar en cuenta que el dataset con el que se está haciendo el estudio se “comporta bien” y que con uno mas complicado es probable que al seguir aumentando el learning rate el error sea más inestable y la métrica utilizada en esta medición no sea representativa.

3.3. Otras consideraciones

A pesar de que el dataset contenía solo 200 entradas para el entrenamiento de la red, esta toma 39.6 segundos realizar 3000 epoch en un computador personal. Esto se debe a que no fue optimizada en ningún momento y fue escrita solo con fines de aprendizaje.

También cabe mencionar que el orden de los datos no afecta en absoluto los resultados obtenidos, se hizo una prueba con los datos ordenados por resultado y otra con los datos ordenados aleatoriamente y se puede ver en la figura 5 que el resultado es exactamente el mismo en ambos casos.

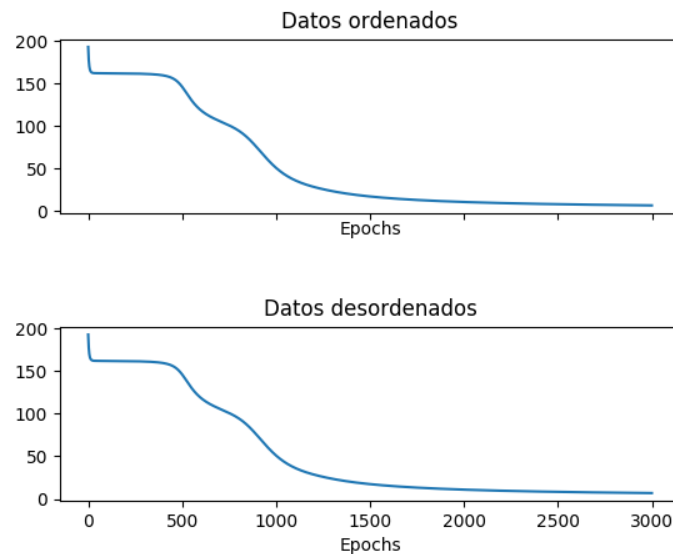


Figura 5: Error con datos ordenados y desordenados.