Análisis de variación de hiperparametros en una red neuronal

Adolfo Banchio*
Facultad de Matemática, Astronomía, Física y Computación, Universidad Nacional de Córdoba
(Dated: 26 de noviembre de 2024)

En este trabajo se estudiara el comportamiento de una red neuronal feed-forward en un problema de clasificación de imágenes mediante variación de los parámetros que definen al modelo. Se entrenaron las diferentes configuraciones dentro del mismo sistema para poder analizar métricas del aprendizaje de la red como el valor promedio de la función de costo y la precisión de la red. En pos de poder aprender y observar el gran impacto que tiene la elección de hiperparametros según la arquitectura elegida y el problema a solucionar.

I. INTRODUCCIÓN

Las redes neuronales feed-forward es una de las tantas arquitecturas de redes neuronales existentes y quizás una de las mas simples en cuanto a su estructura[1]. Sin embargo, en su simpleza, encontramos una gran capacidad para poder resolver problemas de clasificación cuando los datos no son linealmente separables y sus tamaños son fijos. Ademas es una arquitectura que resulta mas fácil de configurar y su entrenamiento requiere de menos recursos computacionales que otros tipos de arquitectas. Frente a esta situación, en este trabajo se buscara analizar y responder como influye el parámetro de velocidad de aprendizaje (i.e. learning rate) en la estabilidad y la velocidad con la que la que nuestra red aprende un problema de clasificación de imágenes.

II. TEORÍA

El problema de clasificación que buscaremos resolver uno de los problemas por el MNIST (Modified National Institute of Standards and Technology database) que consiste en crear una red neuronal capaz de clasificar correctamente imagines de 28x28 pixeles de prendas de ropa en 10 categorías diferentes. El instituto provee una base de datos de 70,000 imágenes correctamente etiquetadas, donde 60,000 corresponden al conjunto de entrenamiento y las restantes se utilizan para verificar el comportamiento de la red. Mas información al respecto se puede encontrar en el repositorio oficial del instituto para este problema.[2]

Dentro de una red neuronal FNN (*feedforward neural network*) la información fluye desde la capa de entrada hacia la salida, pasando por todas las neuronas de cada capa. Atravesando conexiones ligadas a diferentes **pesos** entre las neuronas. Durante el entrenamiento se elige un método de optimización, que sera el encargado de calcular el paso siguiente para modificar los pesos, y una función de costo. El algoritmo mas utilizado para estas redes es el de back-propagation, que consiste en luego de una época de entrenamiento, ajustar los pesos bajo el valor de alguna función de costo y un método optimizador. Para este caso se decidió que durante una época el conjunto de datos de entrenamiento sera dividido en mini-batches de 100 elementos de modo que los pesos se actualicen luego de cada mini-batcht. Para este caso utilizaremos la *entropía cruzada* como función de costo y principal mente el *método del descenso por el gradiente estocástico* como optimizador. [3]

La red utilizada esta conformada por una capa de entrada de 784 neuronas, capa oculta n1 de 128 neuronas y capa n2 de 64 neuronas, finalmente la salida tendrá solo 10 neuronas. Las funciones de activación de las capas ocultas es la función ReLU y las métricas utilizadas para medir la estabilidad y rendimiento del modelo serán el promedio de la función costo y la precisión para el conjunto de entrenamiento, tanto durante una época de entrenamiento como después de dicha época, y el conjunto de validación luego de cada época de entrenamiento.

El parámetro tomado como principal objeto de estudio sera el learning rate. Este controla la magnitud de los cambios en los pesos de las sinapsis durante el entrenamiento. Buscamos ver que sucede entre razones de aprendizajes altas y bajas. En la teoría las primeras podrían llevar a una convergencia mas rápida, a una oscilación al rededor de un mínimo e incluso a una divergencia en el aprendizaje, por el otro lado una baja magnitud de crecimiento lleva a una convergencia mas lenta pero mas estable. El valor correcto de este hiperparametro es quizás el mas importante de todos ya que condiciona completamente la capacidad de aprendizaje de el modelo.

III. RESULTADOS

En esta sección se presentaran los datos obtenidos para cada configuración del modelo utilizada variando el learning rate y en algunos casos extendiendo las épocas para analizar como evoluciona el aprendizaje en casos específicos. Para

IV. CASOS PRINCIPALES

1. *Learning rate = 0.001*

En la figura 1 podemos ver que tanto el costo como la precisión son estables y tienen curvas suaves. Siguiendo el resultado esperado de modo que la precisión es mayor luego del entrenamiento, seguido de los datos de validación y finalmente durante el entrenamiento la precisión del modelo es menor. Análogamente sucede lo esperado para el costo en cada caso. Los valores finales de precisión obtenidos son 80.2 %, 79.33 % y 77.81 % respectivamente.

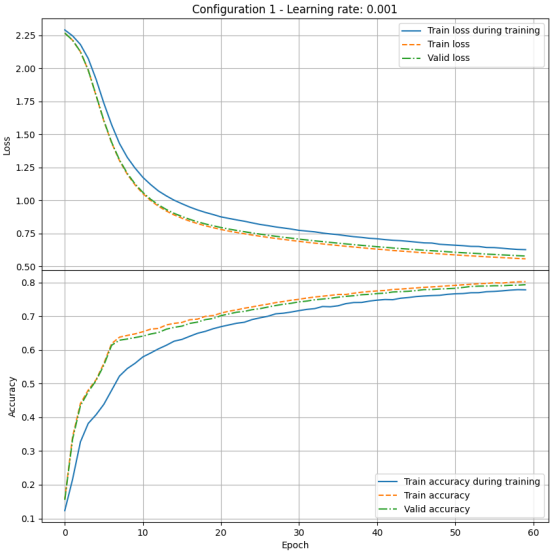

Fig. 1. Gráfico de costo y precisión para el entrenamiento y validación de la red neuronal con learning rate de 0.001.

2. *Learning rate = 0.01*

En la figura 2 nuevamente el modelo aprende de manera estable y sin oscilaciones pero mas rápido que el caso presentado anteriormente. Podemos ver las presciones en la décima época y notamos que las 3 curvas se encuentran por encima del 80 % de precisión. Obteniendo finalmente los valores de 89.4 % posterior al entrenamiento, 88.1 % durante y 87.3 % para el conjunto de validación.

3. *Learning rate = 0.1*

En la figura 3 a simple vista vemos que el aprendizaje en este caso es menos estable y presenta muchos picos

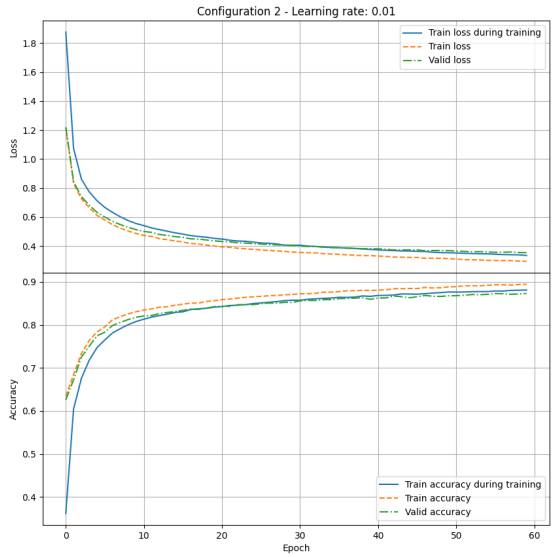

Fig. 2. Gráfico de costo y precisión para el entrenamiento y validación de la red neuronal con learning rate de 0.01.

tanto en el costo como en la precisión. Los valores obtenidos fueron de una precisión de 93.6 % luego de entrenar, 92.2 % durante y 88.79 % para el conjunto de validación.

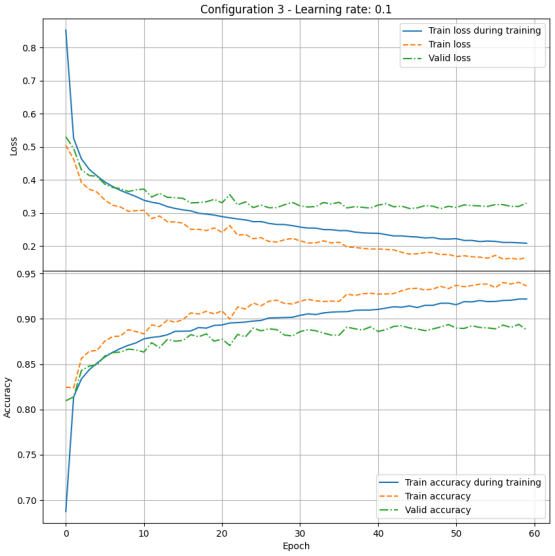

Fig. 3. Gráfico de costo y precisión para el entrenamiento y validación de la red neuronal con learning rate de 0.1.

A. Casos extra

Para analizar e intentar visibilizar la importancia y variabilidad de la razón de aprendizaje en una red neuronal, se hicieron dos casos extra cambiando el optimizador por Adam (*Adaptive Moment Estimation*, con parámetros predeterminados[4]).

El primer caso se puede ver en la figura 4 donde utilizamos el learning rate con mejores resultados de los 3 casos anteriores, 0.01, y vemos como ahora el aprendizaje de la red ya no es el mas optimo para este optimizador.

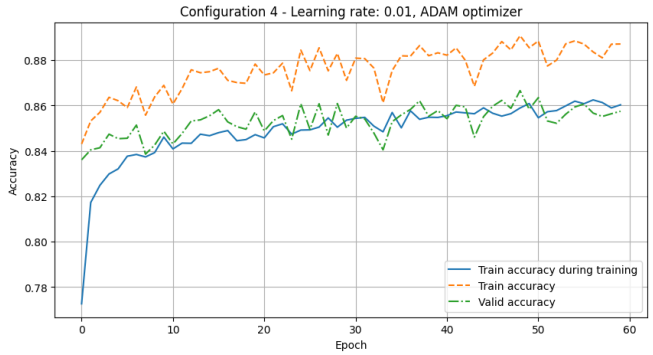

Fig. 4. Gráfico de precisión para el entrenamiento y validación de la red neuronal con learning rate de 0.01 y optimizador Adam

El segundo caso extra se utilizo nuevamente un learning rate de 0.001 y un optimizador Adam con valores predeterminados. Esta vez el gráfico es algo mas estable pero no presenta mejorías con los anteriores.

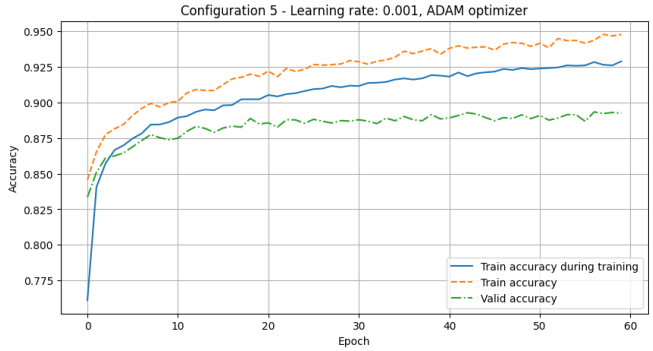

Fig. 5. Gráfico de precisión para el entrenamiento y validación de la red neuronal con learning rate de 0.001 y optimizador Adam

V. DISCUSIÓN

Los experimentos realizados permiten poder analizar como puede influir abruptamente el cambio de este hiperparametro. Para el primer caso notamos que al ser pequeño, podemos notar en la curva de precisión como el aprendizaje se hace cada vez aun mas lento pero mantiene una buena estabilidad y las 3 curvas son cercanas.

A medida que elegimos una razón de aprendizaje mayor, cuando lr = 0,01, podemos ver como en las primeras épocas de entrenamiento este aumento en la magnitud de los pasos se hace notar y el aprendizaje crece rápidamente como también los costos disminuyen de gran manera. Pero nuevamente el crecimiento se hace lento pero muy estable que es algo que nos interesa.

Por ultimo a mayor learning rate vemos curvas menos estables y con mayor cantidad de máximos y mínimos locales, esto refleja los saltos mas grandes que se da en los gradientes calculados. Si bien la precisión alcanzada en los datos de entrenamiento luego de corregir los pesos es muy alta, podemos ver que la precisión del conjunto de validación entra en una tendencia mas bajista. Esto nos dice que nuestra red esta realizando un **sobre ajuste**, lo que significa que esta perdiendo la capacidad de generalizar a costa de aprender de mejor manera las imágenes del conjunto de entrenamiento. Esto evidentemente no es bueno para nuestra red por lo que un learning rate tan alto es descartado.

Finalmente se decide mostrar como un valor de este hiperparametro puede ser bueno bajo una condición muy especifica donde nuestra red se eligió para utilizar el método del descenso por el gradiente, pero apenas se modifica el método de optimización, toda nuestra búsqueda de un learning rate que nos sea favorable pierde valor. En las imágenes 4 y 5 vemos como los dos valores que nos habían dado resultados beneficiosos ahora generaban un sobre ajuste en nuestro modelo.

VI. CONCLUSIONES

Pudimos notar que para nuestra estructura y arquitectura elegida un learning rate al rededor del 0.01 puede brindarnos buenos resultados. Pero también se observo y se logro representar la sensibilidad en el resultado que se presenta cuando hacemos cambios en nuestra arquitectura. Otros parámetros que podrían haber sido modificados para contra restar las imperfecciones es el tamaño de los mini-batch para poder introducir mas ruido en el calculo de los gradientes o al revés, buscar hacer cálculos del gradiente mas estables y por ende equilibrar las grandes variaciones generadas por un learning rate alto.

Los hiperparametros y configuraciones de modelos de redes neuronales son elementos muy delicados a la hora de querer generar modelos precisos. Es por ello que es de gran importancia conocer y analizar el set de datos previo al diseño de la red en pos de poder elegir primeros hiperparametros adecuados y así acercarnos hacia el mejor resultado posible.

VII. AGRADECIMIENTOS

A la cátedra de Redes Neuronales, en particular a Benjamin Marcolongo que ha sido de gran apoyo durante la cursada de la materia, también agradezco a FAMAFyC por brindar un espacio de aprendizaje.

* adolfo.banchio@mi.unc.edu.ar

[1] Wikipedia contributors, Feedforward neural network (2024).
[2] MNIST, Fasgion-mnist (2024).
[3] Wikipedia contributors, Back propagation (2024).
[4] P. Documentation, Adam optimizer (2024).