

# Clasificación de Fashion MNIST

Alejandro Matias Toledo\*

Facultad de Matemática, Astronomía, Física y Computación, Universidad Nacional de Córdoba

(Dated: 18 de noviembre de 2024)

En este proyecto, se evaluó el desempeño de una red neuronal feedforward para la clasificación de imágenes del conjunto de datos Fashion MNIST. La arquitectura del modelo consta de una capa de entrada de 784 unidades, dos capas ocultas con 128 y 64 neuronas, respectivamente, activación ReLU, *dropout* con tasa de 0.02, y una capa de salida con 10 neuronas. El modelo fue entrenado durante 60 épocas utilizando los optimizadores SGD y Adam, explorando el impacto de diferentes tamaños de lote (16, 64 y 100) en la precisión de entrenamiento y prueba.

Los resultados mostraron que el tamaño de los lotes influye significativamente en la estabilidad y el rendimiento del modelo. Los tamaños de lote pequeños introducen mayor inestabilidad, pero permiten una mejor precisión final, especialmente con SGD, donde se alcanzó una precisión máxima de 89.97% en el conjunto de entrenamiento y 87.67% en el conjunto de prueba. Por otro lado, el optimizador Adam fue más eficiente en las primeras épocas, pero su desempeño se vio afectado negativamente por tamaños de lote pequeños, mostrando mayor inestabilidad y valores máximos más bajos.

En general, la métrica de precisión en el conjunto de entrenamiento (*Prec. Train*) fue consistentemente superior, lo que sugiere posibles casos de overfitting al final del entrenamiento. Este análisis destaca la relevancia del tamaño de lote y del optimizador en la estabilidad del entrenamiento y la capacidad del modelo para adaptarse a datos desconocidos. Estos resultados aportan claridad sobre cómo ajustar estos parámetros para optimizar el desempeño en tareas de clasificación.

## I. INTRODUCCIÓN

Las redes neuronales feedforward son una de las arquitecturas más simples y fundamentales dentro del aprendizaje automático. Estas redes están compuestas por una capa de entrada, una o más capas ocultas y una capa de salida. Cada nodo o neurona de una capa se conecta con las neuronas de la capa siguiente a través de pesos, los cuales se ajustan iterativamente durante el proceso de entrenamiento. [2].

En este trabajo, se utiliza una red neuronal feedforward para abordar la tarea de clasificación de imágenes en el conjunto de datos Fashion MNIST. Este conjunto contiene un total de 70,000 imágenes de prendas de ropa, divididas en 60,000 muestras para entrenamiento y 10,000 para evaluación. La arquitectura de la red consta de 784 unidades de entrada, dos capas ocultas con 128 y 64 neuronas, respectivamente, cada una con función de activación ReLU y un mecanismo de *dropout* con una tasa de 0.02, y una capa de salida con 10 neuronas correspondientes a las categorías de las prendas.

El modelo será entrenado durante 60 épocas utilizando diferentes métodos de optimización, como SGD (*Stochastic Gradient Descent*) y Adam (*Adaptive Moment Estimation*, en este proyecto se utilizará con los parámetros predeterminados). Se busca analizar el impacto del tamaño de los lotes y el método de optimización en la precisión final del modelo.

## II. TEORÍA

Las redes neuronales feedforward, también conocidas como redes prealimentadas, son arquitecturas donde la información fluye en una única dirección, desde la capa de entrada hasta la capa de salida, pasando por una o más capas ocultas. Cada conexión entre neuronas está asociada a un peso, que se ajusta durante el entrenamiento para minimizar una función de pérdida.

El entrenamiento de una red se realiza en iteraciones llamadas épocas, que implican recorrer el conjunto de datos de entrenamiento completo. En cada iteración, los pesos se actualizan utilizando algoritmos de retropropagación, los cuales ajustan los parámetros del modelo según los gradientes calculados a partir de la función de pérdida.

En este proyecto, la precisión también se utiliza como métrica para evaluar el desempeño del modelo. Esta métrica mide la proporción de predicciones correctas sobre el total de ejemplos evaluados, permitiendo comparar el impacto de diferentes configuraciones de entrenamiento y métodos de optimización.

El conjunto de datos Fashion MNIST se compone de imágenes de 28x28 píxeles en escala de grises, cada una asociada a una etiqueta que representa una categoría de prenda. Este conjunto se utiliza ampliamente para evaluar modelos de clasificación debido a su simplicidad y naturaleza bien definida.[1]

### III. RESULTADOS

En esta sección se presentan los resultados obtenidos al entrenar el modelo. Se analizarán tres métricas principales:

- **Prec. Test:** Precisión evaluada con el conjunto de prueba, calculada al final de cada época.
- **Prec. Train:** Precisión evaluada con el conjunto de entrenamiento, calculada al final de cada época.
- **Prec. During:** Precisión durante el entrenamiento del modelo, registrada mientras el modelo procesa los lotes de datos.

A partir de estas definiciones, las métricas serán referenciadas como *Prec. Test*, *Prec. Train* y *Prec. During* a lo largo del análisis.

#### A. SGD

##### 1. Tamaño de lote 100

En la figura 1, se observa que las métricas *Prec. Train* y *Prec. Test* coinciden durante las primeras dos épocas. A partir de la tercera época, las métricas se estabilizan siguiendo el orden esperado: *Prec. Train*, seguida de *Prec. Test*, y finalmente *Prec. During*. Los valores finales representan los máximos alcanzados en esta configuración, con *Prec. Train* logrando un 83.56 % y *Prec. Test* un 82.13 %.

##### 2. Tamaño de lote 64

Como se observa en el gráfico 2, las métricas *Prec. Train* y *Prec. Test* también coinciden durante las primeras dos épocas, comportándose de manera similar al caso anterior. Posteriormente, el orden de las métricas se mantiene: *Prec. Train*, *Prec. Test* y, por último, *Prec. During*. Sin embargo, en esta configuración se alcanzan mayores valores máximos: *Prec. Train* logra un 85.30 %, mientras que *Prec. Test* llega a 83.84 %, superando a los obtenidos con lotes de tamaño 100.

##### 3. Tamaño de lote 16

En la figura 3, se observa un cambio respecto a los casos anteriores. A partir de la época 19, el orden entre *Prec. Test* y *Prec. During* se invierte, siendo *Prec. During* consistentemente mayor que *Prec. Test*, aunque *Prec. Train* sigue manteniéndose como la métrica más alta. En esta configuración, los valores máximos se alcanzan en la época 60, con *Prec. Train* logrando el mayor valor entre los tres casos de SGD (89.97 %) y *Prec. Test* mostrando una mejora significativa con 87.67 %.

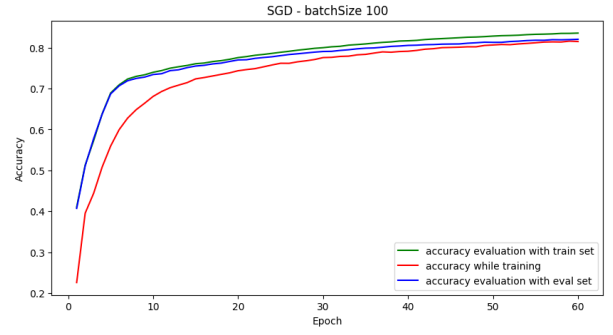


Fig. 1. Gráfico de la precisión en función de las épocas utilizando el optimizador SGD con un tamaño de lote de 100. La línea roja representa a *Prec. During*, la azul corresponde a *Prec. Train*, y la verde muestra la *Prec. Test*.

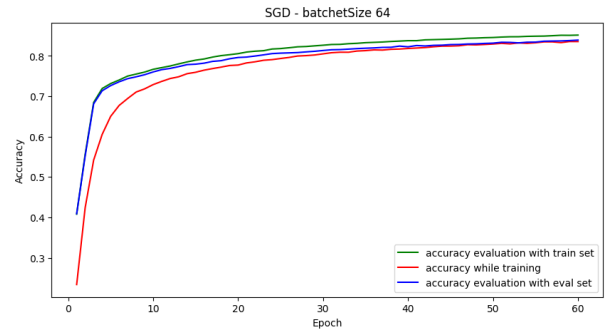


Fig. 2. Gráfico de la precisión en función de las épocas utilizando el optimizador SGD con un tamaño de lote de 64. La línea roja representa a *Prec. During*, la azul corresponde a *Prec. Train*, y la verde muestra la *Prec. Test*.

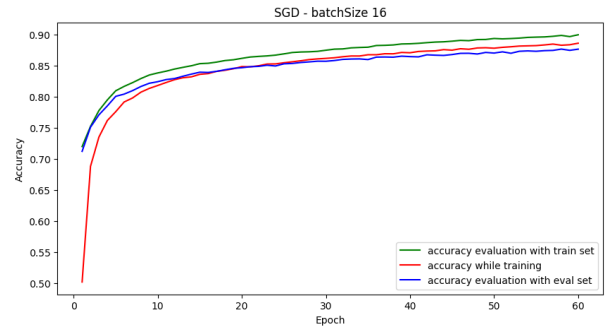


Fig. 3. Gráfico de la precisión en función de las épocas utilizando el optimizador SGD con un tamaño de lote de 16. La línea roja representa a *Prec. During*, la azul corresponde a *Prec. Train*, y la verde muestra la *Prec. Test*.

#### B. Adam

##### 1. Tamaño de lote 100

En la figura 4, se observa que al finalizar cada época, *Prec. Train* se mantiene consistentemente por encima de

las demás métricas. Sin embargo, en torno a la época 5, *Prec. Test* supera temporalmente a *Prec. During*. A partir de ese punto, el orden de las métricas se estabiliza: primero *Prec. Train*, luego *Prec. During*, y por último *Prec. Test*. Cabe destacar que *Prec. Test* no supera el 90 %, mientras que *Prec. Train* alcanza su máximo valor de 95.1 %.

### 2. Tamaño de lote 64

En la figura 5, nuevamente *Prec. Train* supera a las otras métricas al final de cada época. De manera similar al caso anterior, alrededor de la época 5, *Prec. Test* supera temporalmente a *Prec. During*. El orden entre las métricas se mantiene consistente: primero *Prec. Train*, luego *Prec. During*, y finalmente *Prec. Test*. No obstante, el máximo alcanzado por *Prec. Train* en esta configuración es de 94.4 % (en la época 56), un valor menor al obtenido con el tamaño de lote de 100. En cuanto a *Prec. Test*, su máxima precisión se registra en la época 33, con un 89.01 %, sin llegar al 90 %.

### 3. Tamaño de lote 16

En la figura 6, al igual que en los casos anteriores, *Prec. Train* sigue siendo superior a las otras métricas al final de cada época. Sin embargo, aquí la mayor precisión de *Prec. Test* sobre *Prec. During* ocurre en la época 2. Posteriormente, el orden entre las métricas se mantiene sin cambios. Es importante señalar que en este caso, *Prec. Train* no supera el 94 %, y *Prec. Test* no llega al 89 % de precisión, un valor inferior al alcanzado en los otros casos. Este comportamiento resalta la alta inestabilidad de las métricas cuando se utiliza un tamaño de lote pequeño, lo cual afecta negativamente el rendimiento del modelo.

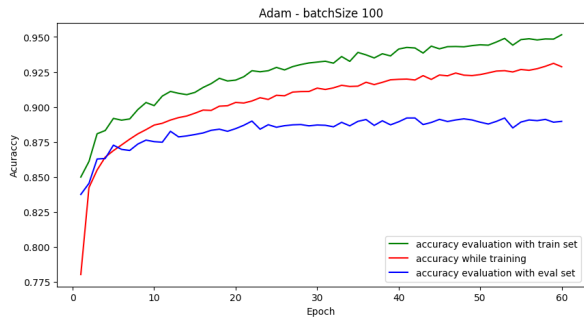


Fig. 4. Gráfico de la precisión en función de las épocas utilizando el optimizador Adam con un tamaño de lote de 100. La línea roja representa a *Prec. During*, la azul corresponde a *Prec. Train*, y la verde muestra la *Prec. Test*.

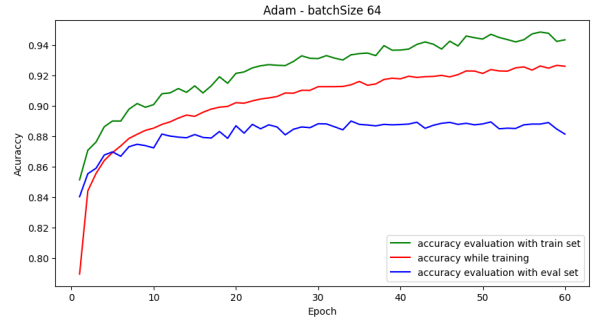


Fig. 5. Gráfico de la precisión en función de las épocas utilizando el optimizador Adam con un tamaño de lote de 64. La línea roja representa a *Prec. During*, la azul corresponde a *Prec. Train*, y la verde muestra la *Prec. Test*.

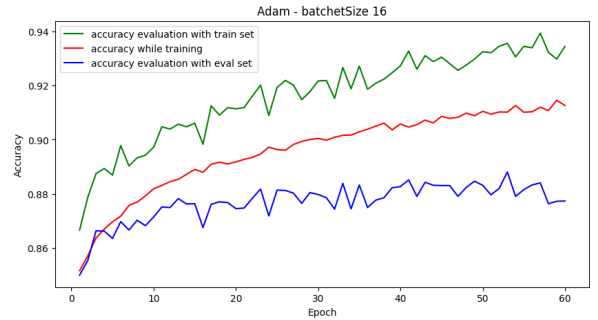


Fig. 6. Gráfico de la precisión en función de las épocas utilizando el optimizador Adam con un tamaño de lote de 16. La línea roja representa a *Prec. During*, la azul corresponde a *Prec. Train*, y la verde muestra la *Prec. Test*.

## IV. DISCUSIÓN

En los experimentos realizados, se observa que *Prec. Train* es consistentemente superior a las demás métricas en todos los casos estudiados, lo que refleja que el modelo logra ajustarse de manera más eficiente al conjunto de entrenamiento.

Con el optimizador SGD, se notó una relación directa entre el tamaño de los lotes y la estabilidad del modelo. Lotes pequeños introducen mayor fluctuación en las métricas, pero también logran un ajuste más fino en *Prec. Train* y *Prec. Test*, particularmente con tamaños de lote de 16. Este fenómeno puede explicarse por el ruido añadido en cada iteración, que favorece una exploración más profunda del espacio de parámetros. Sin embargo, esta inestabilidad puede comprometer la generalización del modelo, como se observa en el cruce más notable entre *Prec. Test* y *Prec. During* en este caso 3.

Por otro lado, con el optimizador Adam, se observa un comportamiento distinto. Aunque inicialmente parece más robusto (como lo demuestra el rápido crecimiento de las métricas en las primeras épocas), los valores máximos alcanzados disminuyen conforme el tamaño de los lotes

se reduce. Además, la inestabilidad se incrementa significativamente, lo que sugiere que Adam es más sensible a variaciones en el tamaño de lote, afectando al modelo negativamente.

En general, es notable que *Prec. Test* tiende a "estancarse" al acercarse a las últimas épocas en ambos optimizadores, mientras que *Prec. Train* y *Prec. During* mantienen cierto crecimiento, y aumenta cada vez más la diferencia entre *Prec. Train* y *Prec. Test*. Esto podría indicar un posible caso de overfitting (sobreadaptación) en el conjunto de entrenamiento, especialmente en las configuraciones con tamaños de lote más pequeños.

## V. CONCLUSIONES

Frente a esta investigación se obtuvo conclusiones en diferentes enfoques:

**Relación entre tamaño de lote y estabilidad:** Los tamaños de lote grandes proporcionan mayor estabilidad durante el entrenamiento, permitiendo alcanzar valores máximos más altos en las métricas de evaluación. Sin embargo, lotes pequeños, aunque más inestables, logran mejores ajustes finales en configuraciones específicas (modelo SGD), lo que lo destaca para optimizar la precisión en entornos donde la inestabilidad es controlable.

### Diferencias entre optimizadores:

**SGD:** Presenta un comportamiento más estable y predecible, incluso con tamaños de lote pequeños. Su rendimiento mejora significativamente en configuraciones con lotes más pequeños, lo que sugiere que su sensibilidad al ruido favorece la exploración del espacio de parámetros encontrando "mejores errores mínimos" y entrenando el modelo con mas precisión.

**Adam:** Aunque más eficiente en las primeras épocas, su desempeño se ve afectado negativamente por tamaños de

lote pequeños, mostrando menor capacidad de alcanzar valores máximos y mayor inestabilidad general.

**Patrones comunes:** En todos los casos estudiados, se observa un crecimiento logarítmico en las tres métricas (*Prec. Train*, *Prec. Test* y *Prec. During*), especialmente en las primeras épocas. Y la *Prec. Train* se mantiene por encima de las otras métricas.

En resumen, el tamaño de los lotes y la elección del optimizador tienen un impacto significativo en la estabilidad y en la precisión. SGD con tamaños de lote pequeños parece ser más efectivo en configuraciones donde se busca maximizar la precisión, mientras que Adam podría ser más adecuado para entrenamientos rápidos y menos sensibles a fluctuaciones iniciales.

## VI. AGRADECIMIENTOS

A la cátedra de Redes Neuronales, en particular a Benjamin Marcolongo que ha sido de gran apoyo durante la cursada de la materia, agradezco a FAMAFyC por brindarme un espacio de aprendizaje. También agradecer a MNIST por proveer un dataset libre de uso y por ultimo agradecer a Nicolas Wolowick por brindarme acceso a una computadora virtual para poder entrenar los modelos.

---

\* amtoledo2002@mi.unc.edu.ar

- [1] Dr. Juan Ignacio Perotti Dr. Francisco Tamarit. Notas de clases de redes neuronales, 2024.
- [2] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination press, 2016. [Enlace consultado el 18 de noviembre de 2024].