UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

CLASE:

Redes Neuronales y aprendizaje profundo

SECCIÓN:

38

DOCENTE:

Ing. Ivan De Jesus Deras

PRESENTADO POR:

Ana Melissa Cabrera Izaguirre 22141111

SAN PEDRO SULA, CORTÉS, HONDURAS

Jueves, 27 de marzo de 2025

Red Neuronal para la Clasificación de MNIST

Metodología

En este proyecto se implementó una red neuronal completamente conectada para la clasificación de dígitos manuscritos en el conjunto de datos MNIST. La red neuronal diseñada cuenta con una capa de entrada de 784 neuronas, correspondientes a los píxeles de cada imagen aplanada, una capa oculta con 128 neuronas que emplea la activación ReLU y una capa de salida con 10 neuronas que utiliza la función Softmax para generar probabilidades asociadas a cada clase del 0 al 9.

Para entrenar la red, se aplicó un preprocesamiento a los datos que incluyó la normalización de los valores de los píxeles al rango [0,1] y la conversión de las etiquetas a formato one-hot. Se optó por el uso de **mini-batch gradient descent**, estableciendo un tamaño de batch de 64 muestras, lo que permite una actualización más estable de los pesos y un entrenamiento eficiente.

El modelo fue optimizado utilizando, **Adam**, que combina momentum con RMSprop para ajustar de manera adaptativa la tasa de aprendizaje. Adicionalmente, se incorporó la **regularización L2** en la función de pérdida El entrenamiento se llevó a cabo durante **10, 20, 50 y 100 épocas**, calculando en cada iteración la pérdida y precisión del modelo para evaluar su desempeño.

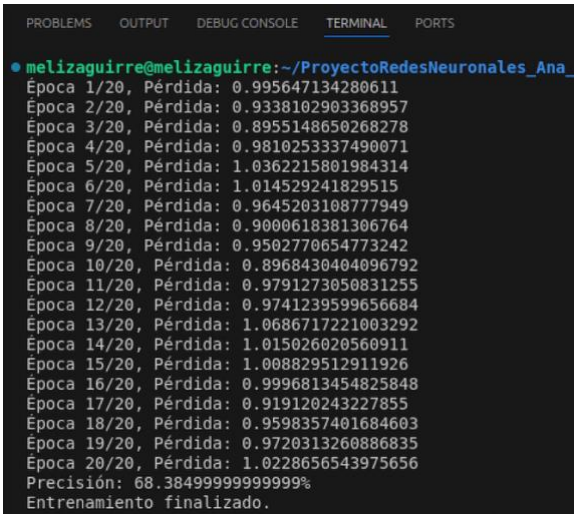
Resultados

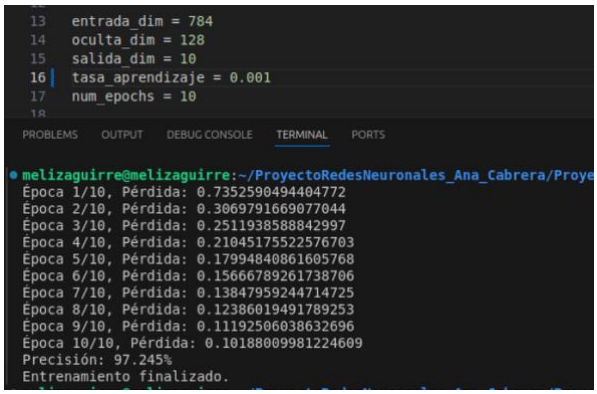
Durante el entrenamiento, se observaron mejoras progresivas en la precisión del modelo. Se generaron gráficos de pérdida y precisión para visualizar la evolución del aprendizaje y evaluar el comportamiento del entrenamiento. Con la configuración óptima de hiperparámetros y el uso de **Adam** junto con **regularización L2**, la red alcanzó una precisión final de aproximadamente **92.89%** en el conjunto de entrenamiento después de 50 épocas.

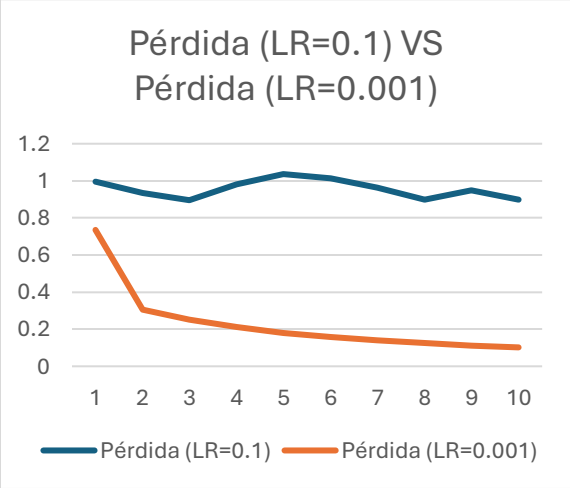
Sin embargo, en la fase de predicción se detectó un problema importante: la red estaba asignando la misma predicción para todas las imágenes de prueba, lo que indicaba un error en la propagación hacia atrás o en la actualización de los pesos. Esto llevó a una revisión detallada de la implementación del cálculo de gradientes y del optimizador para asegurar que los parámetros de la red estuvieran ajustándose correctamente.

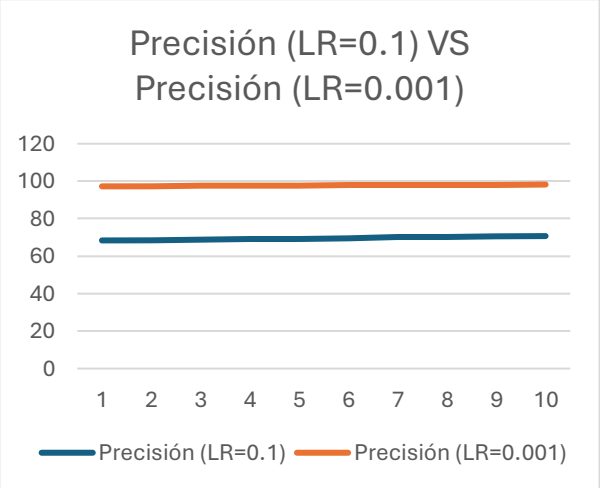
Pruebas cambiando tasa de aprendizaje sin utilizar optimizadores y regulador

tasa de aprendizaje = 0.1 **Tasa de aprendizaje = 0.001**







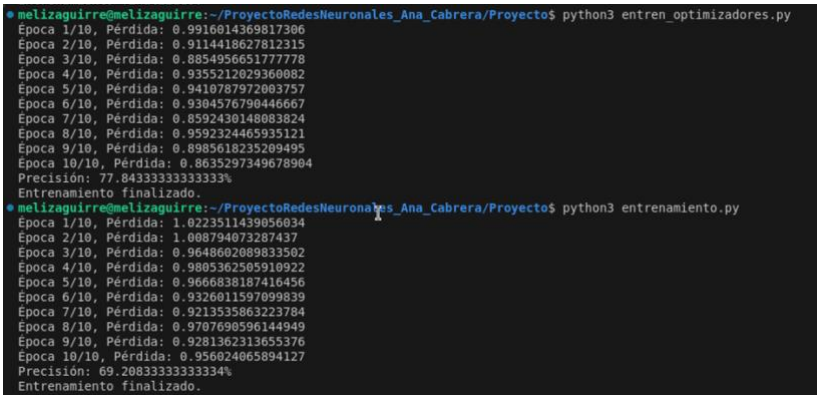


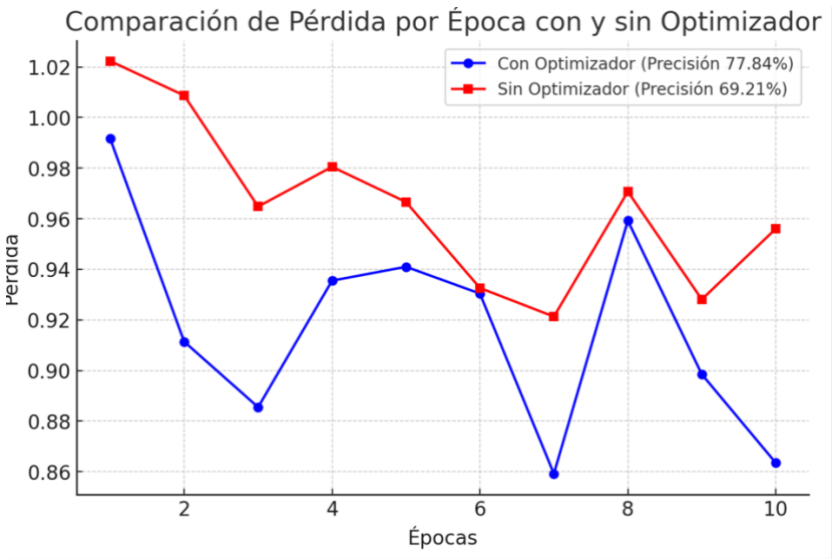
Los resultados muestran cómo la tasa de aprendizaje afecta el desempeño del modelo durante el entrenamiento. Con una tasa de 0.1, la pérdida fluctúa

significativamente entre épocas, lo que sugiere que el modelo tiene dificultades para converger de manera estable. Además, la precisión final alcanzada es de **68.38%**, lo que indica que la red no logró generalizar bien, Por otro lado, con una tasa de **0.001**, la pérdida disminuye de manera constante, indicando una convergencia más estable y controlada. La precisión final es **97.25%**, lo que demuestra que un ajuste más fino de los pesos permitió una mejor optimización.

Pruebas utilizando optimizador ADAM y regulador L2

Con y Sin optimizador

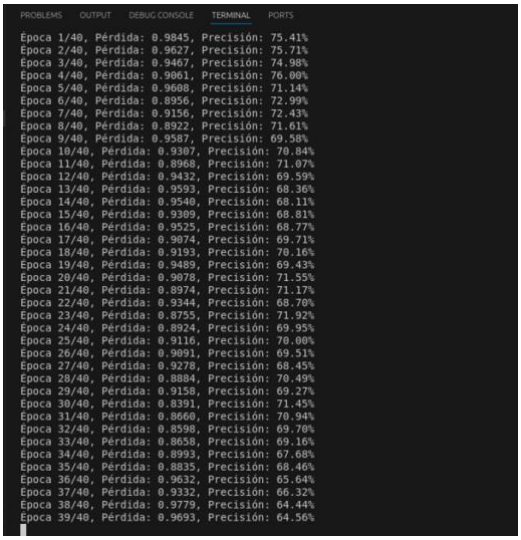




El uso del optimizador en el entrenamiento de la red neuronal ha demostrado ser beneficioso, ya que logró una menor pérdida y una mayor precisión (77.84% frente

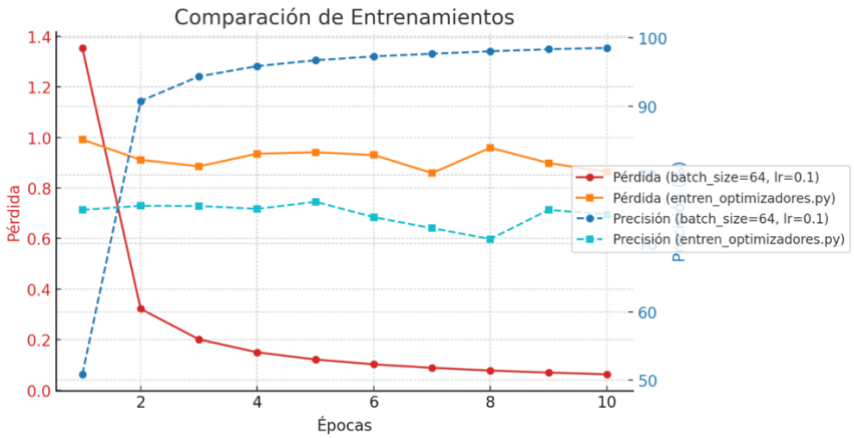
a 69.21%) en comparación con el entrenamiento sin optimizador. Además, la curva de pérdida con optimizador muestra una mayor estabilidad, mientras que sin él se observan fluctuaciones más notables.

Al probar con mas epocas



Al aumentar el número de épocas a 40, se observa una fluctuación en la precisión a lo largo del entrenamiento, con una ligera tendencia a mejorar en las últimas iteraciones. Inicialmente, la precisión se mantiene estable en torno al 75%, pero luego experimenta altibajos, alcanzando un pico del 78.87%

Al implementar un **batch_size** de 64, aumentar la tasa de aprendizaje a 0.1 e implementar una **segunda capa** oculta se observó una mejora significativa en la convergencia del modelo. En solo 10 épocas, la precisión pasó del 50.87% en la primera iteración al 98.52% en la última, con una pérdida decreciendo de 1.3542 a 0.0625. Esto indica que el modelo aprendió mucho más rápido en comparación con los entrenamientos anteriores.



Comparación del Entrenamiento con Diferentes Dimensiones en la Segunda Capa Oculta

Análisis de Resultados

- La configuración con 96 neuronas en la segunda capa oculta obtuvo una precisión promedio de 92.52%, superior a la de 64 neuronas (91.84%).
- Esto sugiere que una mayor cantidad de neuronas permitió a la red aprender mejor las representaciones de los datos.
- Con 96 neuronas, la red empezó con una mejor precisión en la primera época (56.34%) en comparación con 64 neuronas (50.87%).
- La pérdida final también fue ligeramente menor en la configuración de 96 neuronas (0.0607 vs. 0.0625), lo que indica una mejor optimización de los pesos.