



Tecnológico de Monterrey

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE
MONTERREY CAMPUS ESTADO DE MÉXICO

Inteligencia artificial avanzada para la ciencia de datos I.

Grp. 101.

Alumno:

Leonardo Cossio Dinorin.

Profesor:

Jorge Adolfo Ramírez Uresti.

Módulo:

Aprendizaje máquina

**Implementación de una técnica de aprendizaje máquina sin el uso de un
Framework.**

Fecha de entrega:

01 de septiembre del 2024.

Evaluación del modelo creado:

Descripción del modelo:

Se desarrolló una red neuronal capaz de clasificar imágenes que contienen dígitos del 0 al 9 escritos a mano sin utilizar un Framework.

Para esto se programó una serie de funciones en el archivo ***NN_Functions.py*** y se mandaron a llamar desde el archivo principal: ***EvidenciaLeoC.py***.

Para este ejemplo, se inicializó una red neuronal cuyas dimensiones son:

- 🚦 **Capa de entrada:** 784 (Ya que se están clasificando imágenes de 28x28 píxeles)
- 🚦 **Capa oculta 1:** 32 neuronas
- 🚦 **Capa oculta 2:** 16 neuronas
- 🚦 **Capa de salida:** 10 (Ya que se clasifican 10 clases diferentes)

```
input_dim = 784
hidden1_dim = 32 # Neuronas de la primera capa oculta
hidden2_dim = 16 # Neuronas de la segunda capa oculta
output_dim = 10
```

Declaración de las dimensiones de la red neuronal

El *dataset* y reto utilizado se obtuvo de: <https://www.kaggle.com/c/digit-recognizer>

Entrenamiento de la red neuronal:

Utilizando la técnica de *backward propagation* y descenso de la gradiente para la actualización de pesos y sesgos, se calculó la *accuracy* del modelo utilizando el set de entrenamiento y validación cada 50 iteraciones.

```
Iteration: 0
Training Accuracy: 0.10504201680672269
Validation Accuracy: 0.12429122249943297
Iteration: 50
Training Accuracy: 0.6280112044817927
Validation Accuracy: 0.6543433885234747
Iteration: 100
Training Accuracy: 0.769827931172469
Validation Accuracy: 0.7636652302109321
```

```
Iteration: 150
Training Accuracy: 0.8305322128851541
Validation Accuracy: 0.8249036062599229
Iteration: 200
Training Accuracy: 0.8591836734693877
Validation Accuracy: 0.8541619414833296
Iteration: 250
Training Accuracy: 0.8760704281712685
Validation Accuracy: 0.8745747334996598
```

Accuracy durante el entrenamiento de la red neuronal con 300 iteraciones

Evaluación del modelo (Métricas):

Para evaluar el modelo creado, se utilizaron 5 métricas aplicadas al set de validación:

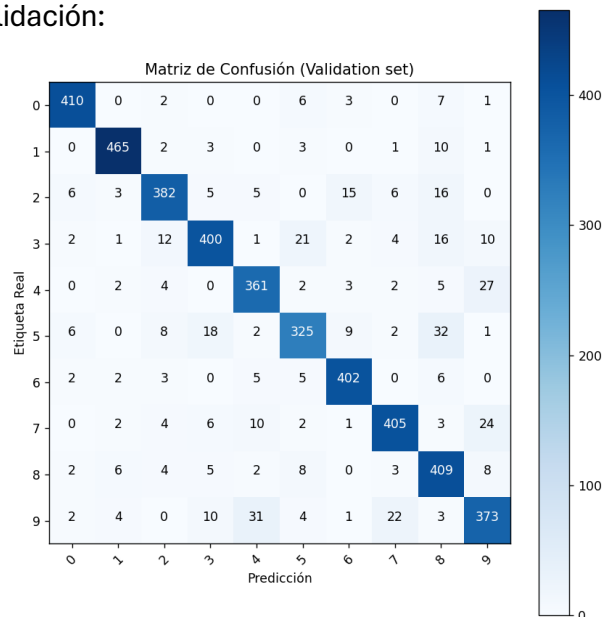
- 📊 **Accuracy:** Precisión del modelo comparando las predicciones con las etiquetas reales.
- 📊 **Precision:** Proporción de verdaderos positivos sobre el total de predicciones positivas.
- 📊 **Recall:** Proporción de verdaderos positivos sobre el total de casos que deberían haber sido positivos.
- 📊 **Specificity:** Proporción de verdaderos negativos sobre el total de casos que deberían haber sido negativos.
- 📊 **F1 score:** Nos da una idea del balance entre la precisión y el recall.

```
*****
Validation Set Accuracy: 0.8918122023134497
Validation Set Precision: 0.9587628865979382
Validation Set Recall: 1.0
Validation Set Specificity: 1.0
Validation Set F1 Score: 0.9789473684210526
*****
```

Como podemos observar, en el set de validación los resultados son bastante altos, pues el modelo muestra un 0.89 de accuracy, un 0.95 de precisión, recall y specificity de 1.0 y finalmente un F1 score de 0.97.

Esto nos revela un comportamiento prometedor para nuestra red neuronal, pues según estas métricas, los verdaderos positivos y negativos son identificados de manera perfecta.

Sin embargo, para estar más seguros de eso, también se obtiene la matriz de confusión utilizando el set de validación:



En esta matriz puede observarse como la red neuronal creada puede clasificar correctamente la mayor parte del tiempo, pues los falsos positivos son mínimos a comparación de los verdaderos.

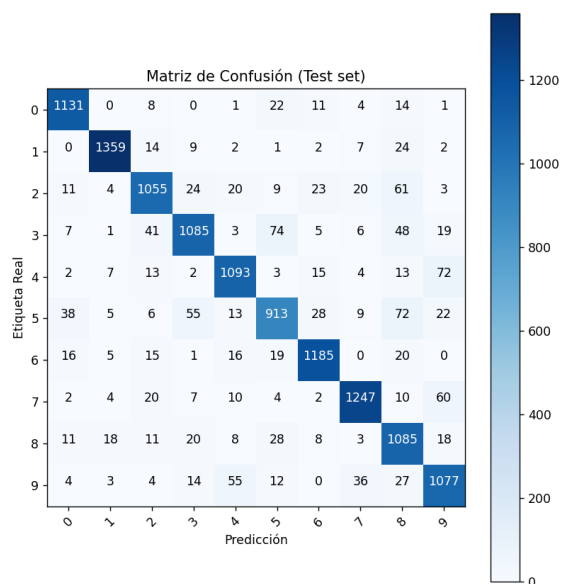
Esta misma prueba se aplicó para el set de prueba (Test) para comprobar que nuestra red neuronal no tuviera un overfitting y asegurarnos de que pueda funcionar para datos nunca antes vistos.

```

*****
Test Set Accuracy: 0.8911991111816523
Test Set Precision: 0.9665718349928877
Test Set Recall: 1.0
Test Set Specificity: 1.0
Test Set F1 Score: 0.983001808318264
*****

```




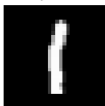
















Se pueden observar resultados muy parecidos a los obtenidos utilizando el set de validación, incluso con un F1 score ligeramente más alto.



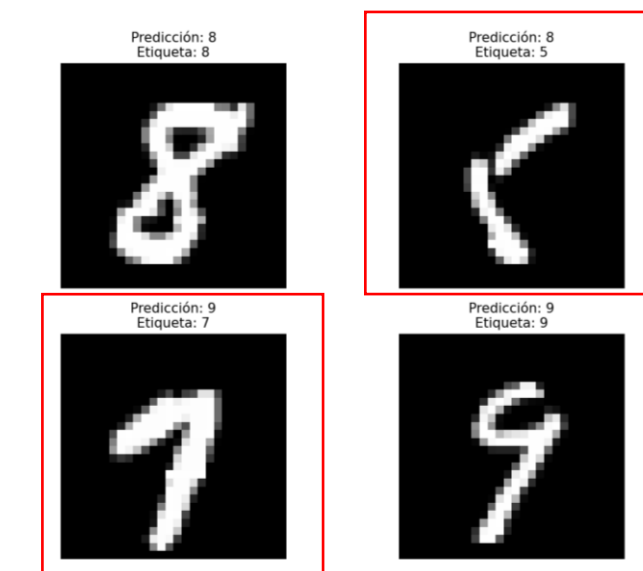
En la matriz de confusión se pueden observar proporciones similares entre verdaderos y falsos positivos, pues siendo un set de datos mucho más grande que el de validación, sigue identificando correctamente la mayoría de los objetos.

Visualización de los resultados:

Por último, se diseñó una manera de observar la imagen con la que se prueba al modelo, así como la etiqueta y la predicción de la misma, de esta manera, el usuario puede observar el funcionamiento de la red de una manera mucho más gráfica.

Predicción: 2 Etiqueta: 2	Predicción: 1 Etiqueta: 1	Predicción: 6 Etiqueta: 6	Predicción: 1 Etiqueta: 1	Predicción: 7 Etiqueta: 7
				
Predicción: 6 Etiqueta: 6	Predicción: 7 Etiqueta: 7	Predicción: 2 Etiqueta: 2	Predicción: 7 Etiqueta: 7	Predicción: 9 Etiqueta: 9
				
Predicción: 0 Etiqueta: 0	Predicción: 8 Etiqueta: 8	Predicción: 3 Etiqueta: 3	Predicción: 9 Etiqueta: 9	Predicción: 5 Etiqueta: 5
				
Predicción: 3 Etiqueta: 3	Predicción: 1 Etiqueta: 1	Predicción: 1 Etiqueta: 1	Predicción: 1 Etiqueta: 1	Predicción: 9 Etiqueta: 9
				

En esta prueba se seleccionaron 20 imágenes al azar del set de prueba y el modelo predijo correctamente todas ellas, sin embargo, como pudimos observar en la matriz de confusión, esta red neuronal no es perfecta, por lo que en una segunda iteración, se obtuvieron ejemplos de predicciones erróneas.



Podemos ver como existen casos donde claramente cuesta diferenciar los números escritos y la red neuronal termina confundiéndose.

Conclusión:

Logró implementarse una red neuronal capaz de clasificar imágenes con dígitos escritos a mano desde cero y sin utilizar un Framework especializado o una librería de estadística avanzada. Las únicas librerías utilizadas para este proyecto fueron numpy, pandas y matplotlib, todo lo demás fue hecho a partir de la modificación de un dataset obtenido de kaggle y técnicas de machine learning clásico como lo es la retropropagación para la actualización de pesos y sesgos utilizando el descenso de gradiente. También es importante mencionar que se logró probar la eficacia de la red neuronal utilizando métricas vistas en clase y software capaz de mostrar una interfaz gráfica como lo es matplotlib, de esta manera podemos observar el funcionamiento de la red neuronal e incluso explicarle el mismo a personas que no están familiarizadas con este tipo de temas.