

**Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el desempeño del  
modelo. (Portafolio Análisis)**

Por: Fernando Bustos Monsiváis – A00829931

Escuela de Ingeniería y Ciencias, Instituto Tecnológico y de Estudios Superiores de Monterrey

TC3006C.102: Inteligencia artificial avanzada para la ciencia de datos I

Profesor: MSc Jesús Adrián Rodríguez Rocha

Sábado 07 de septiembre de 2024

## Índice

1. Introducción .....	3
2. Análisis .....	3
2.1. Separación y Evaluación del Modelo con Train/Test/Validation .....	3
2.2. Diagnóstico y explicación del grado de sesgo o bias.....	4
2.3. Diagnóstico y explicación del grado de varianza (Training Loss - Test Loss).....	4
2.4. Diagnóstico y explicación del nivel de ajuste del modelo .....	4
3. Mejora del desempeño con técnicas de regularización.....	5
4. Conclusión .....	5
Anexo.....	6

## 1. Introducción

En este reporte, se analiza el desempeño de un modelo de red neuronal convolucional (CNN) entrenado para clasificar imágenes de dígitos escritos a mano usando el dataset MNIST. Se evaluarán varios aspectos del modelo, incluyendo su capacidad para generalizar, su grado de sesgo y varianza, y el impacto de las técnicas de regularización. Además, se realizarán comparaciones entre el modelo con y sin regularización mediante gráficos de aprendizaje que respalden los diagnósticos.

## 2. Análisis

El análisis incluye una evaluación de las diferentes métricas que permiten diagnosticar la calidad del modelo, como la precisión (accuracy), la pérdida (loss), el grado de sesgo (bias) y la varianza (variance). También se aborda el rendimiento del modelo en términos de ajuste, para identificar si el modelo presenta problemas de subajuste (underfitting) o sobreajuste (overfitting).

### 2.1. Separación y Evaluación del Modelo con Train/Test/Validation

El dataset MNIST se dividió en tres conjuntos:

- **Training set (entrenamiento).** Originalmente, el dataset contiene 60 000 imágenes para entrenamiento. Como se usa un ‘validation\_split=0.2’, esto significa que el 20% de las imágenes del conjunto de entrenamiento se usarán para validación, dejando el 80% para entrenamiento. El 80% de 60 000 imágenes son 48 000 imágenes.
- **Validation set (validación).** El 20% de 60 000 imágenes son 12 000 imágenes.
- **Test set (prueba).** Un conjunto independiente para evaluar la capacidad de generalización del modelo una vez finalizado el entrenamiento. El Test set tiene 10 000 imágenes.

Se utilizaron dos modelos, uno sin técnicas de regularización y otro con Dropout, Batch Normalization y Callbacks: Early Stopping. En la *Tabla 1*, se pueden apreciar las métricas arrojadas por cada uno de los modelos:

Tabla 1. *Evaluación del Modelo sin regularización vs. Modelo regularizado*

	Modelo sin regularización	Modelo regularizado
<b>Test Accuracy (%)</b>	98.96	99.38
<b>Test Loss</b>	0.0344	0.019
<b>Validation Accuracy (%)</b>	98.83	99.18
<b>Validation Loss</b>	0.0454	0.0293
<b>Training Accuracy (%)</b>	99.60	98.89
<b>Training Loss</b>	0.0119	0.0319

Se aprecia que ambos modelos arrojaron muy buenos resultados, aunque el ganador fue el Modelo regularizado por muy ligeras diferencias.

## 2.2. Diagnóstico y explicación del grado de sesgo o bias

Cuando hay mucho sesgo o bias, es porque el modelo es muy simple y no es capaz de capturar los patrones y relaciones relevantes en los datos. El performance del modelo es deficiente en todos los datasets (training, validation y test), es decir, hay underfitting. Como se pudo ver en la *Tabla 1*, esto no ocurre en ninguno de los dos modelos:

- **Modelo sin regularización:** El Test Accuracy es muy alto (98.96%), mismo caso en el Training Accuracy y Validation Accuracy (99.60% y 98.83%, respectivamente). El performance de este modelo es muy alto en todos los datasets.
- **Modelo regularizado:** Este modelo es un poco mejor que el Modelo sin regularización en accuracy. Con 99.38% en el Test Accuracy, 98.89% en el Training Accuracy y 99.18% en el Validation Accuracy.

Como conclusión parcial, ambos modelos presentan muy poco sesgo. Por lo tanto, ningún modelo presenta underfitting.

## 2.3. Diagnóstico y explicación del grado de varianza (Training Loss - Test Loss)

Cuando hay mucha varianza, el modelo es demasiado complejo y captura el ruido y outliers en los datos de entrenamiento. Se desempeña muy bien en el training set; sin embargo, no es capaz de generalizar. El performance del modelo es deficiente en el dataset de test (proporciona demasiadas predicciones incorrectas), es decir, hay overfitting. Como se pudo ver en la *Tabla 1*, ninguno de los dos modelos tiene alta varianza:

- **Modelo sin regularización:** La diferencia de error entre el Training set y el Test set es de 0.0225, la cual es muy pequeña.
- **Modelo regularizado:** La diferencia de error entre el Training set y el Test set es de 0.0129, la cual es ligeramente aún más pequeña. Aunque la diferencia de varianza es mínima entre el Modelo sin regularización y el Modelo regularizado, esta diferencia se puede apreciar un poco más echando un vistazo a la *Figura 1, 2, 3, 4, 5 y 6*, que se encuentran en el *Anexo* de este documento.

Como conclusión parcial, ambos modelos generalizan muy bien. El Modelo sin regularización tiene muy poca varianza y el Modelo regularizado, ligeramente aún menos.

## 2.4. Diagnóstico y explicación del nivel de ajuste del modelo

Como ya se analizó en las subsecciones anteriores, ni el Modelo sin regularización ni el Modelo regularizado presentan underfitting u overfitting. Ambos modelos tienen un ajuste adecuado; sin embargo, el Modelo regularizado tiene un fitting ligeramente mejor.

### 3. Mejora del desempeño con técnicas de regularización

Las técnicas de regularización sirven para reducir la complejidad de los modelos de machine learning durante el entrenamiento con el objetivo de mejorar la generalización del modelo y prevenir el overfitting. Las técnicas de regularización implementadas en este proyecto son:

- **Dropout.** Ayuda a reducir el sobreajuste al apagar aleatoriamente algunas neuronas durante el entrenamiento.
- **Batch Normalization.** Normaliza la salida de una capa para acelerar el entrenamiento y mejorar la estabilidad.
- **Callbacks: Early Stopping.** Detiene el entrenamiento cuando el rendimiento en el conjunto de validación deja de mejorar. En el programa, el entrenamiento se detiene si no mejora después de 3 epochs consecutivas.

El uso de estas técnicas ayudó a que el Modelo regularizado tenga 0.42% más de precisión (accuracy) en el Test set y que esté ligeramente menos sobreajustado. En las *Figuras 1, 2, 3, 4, 5 y 6*, se puede ver que el Modelo regularizado presenta una curva de aprendizaje más estable y menor diferencia entre las curvas de entrenamiento y validación. Lo único negativo, por decirlo de alguna manera, es que la técnica de Dropout y Callbacks: Early Stopping hacen que el número de epochs varíe (con un tope de 20 epochs), mientras que el Modelo sin regularización tiene una constante de 10 epochs. El hecho de que el Modelo regularizado alcance el máximo de 20 epochs sólo generaría un ligero incremento en el tiempo de entrenamiento en comparación con el Modelo sin regularización (3 min (si alcanza los 20 epochs) vs. 1 min (Modelo sin regularizar)).

### 4. Conclusión

El análisis del modelo de red neuronal convolucional (CNN) para el dataset MNIST revela que ambos modelos, tanto sin regularización como con regularización, tienen un rendimiento excelente. No obstante, el uso de técnicas de regularización mejora ligeramente la capacidad del modelo para generalizar, estabiliza la curva de aprendizaje y reduce la diferencia entre las curvas de entrenamiento y validación.

## Anexo

Figura 1. *Accuracy en Entrenamiento y Validación (Sin Regularización)*

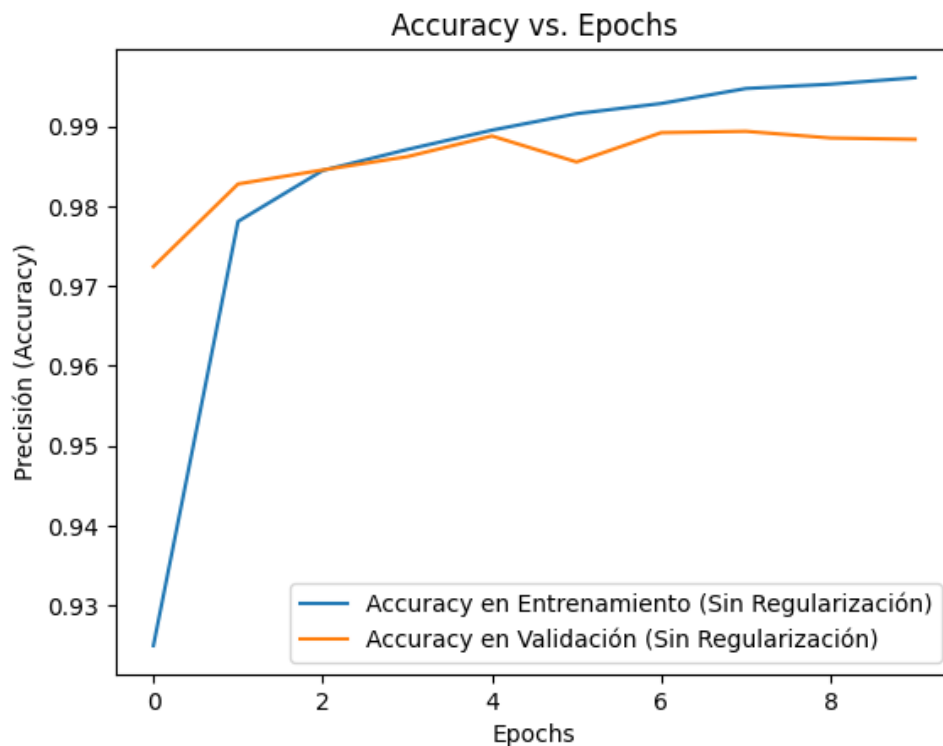


Figura 2. *Loss en Entrenamiento y Validación (Sin Regularización)*

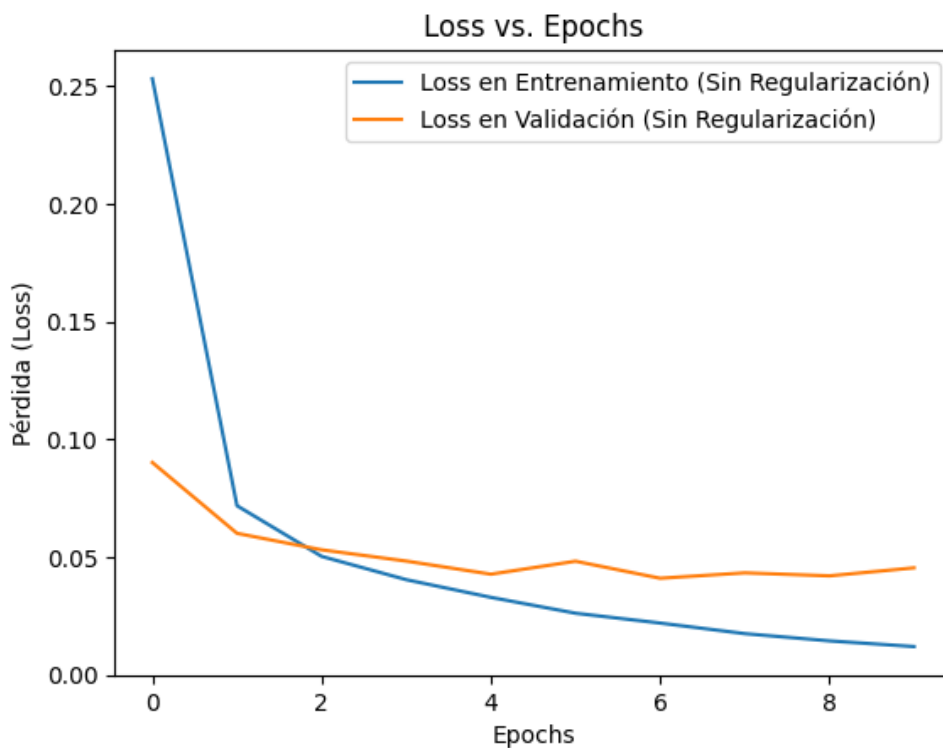


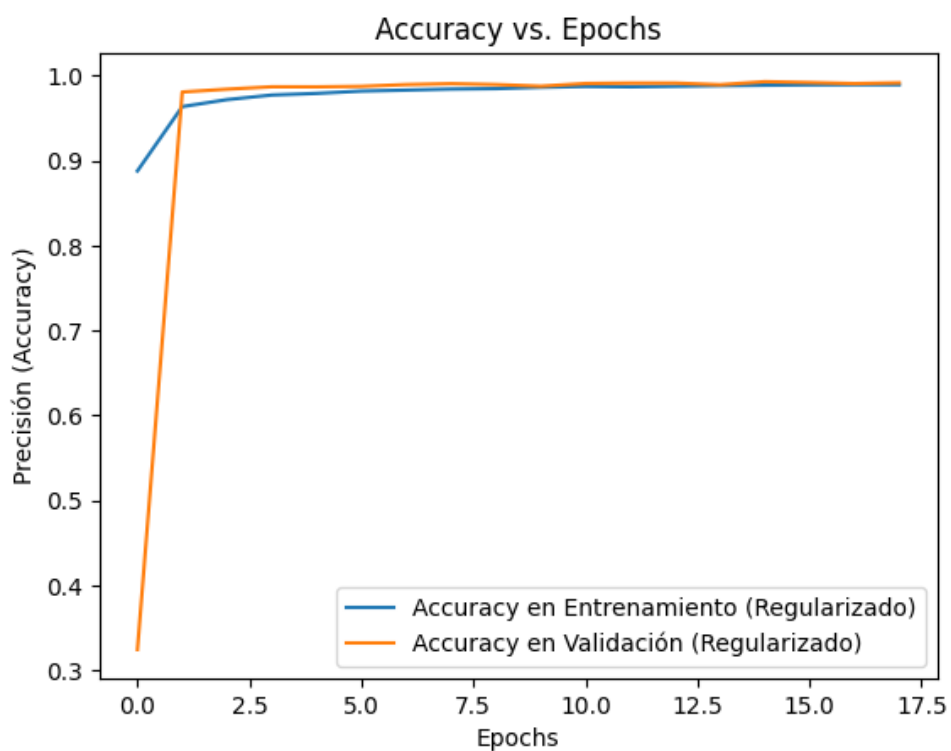
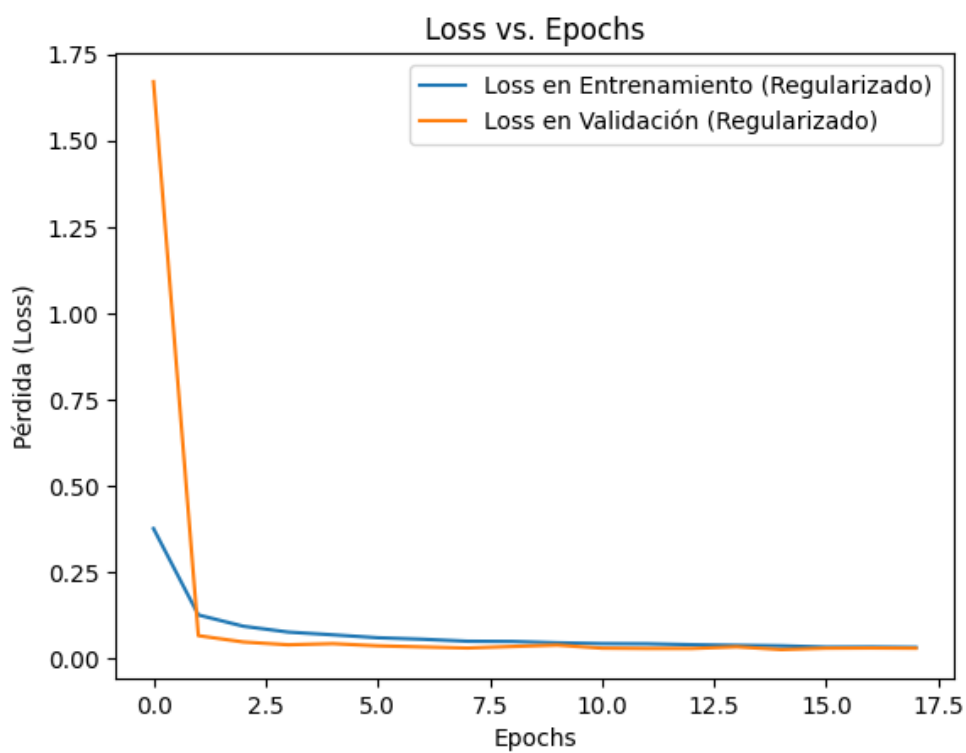
Figura 3. *Accuracy en Entrenamiento y Validación (Regularizado)*Figura 4. *Loss en Entrenamiento y Validación (Regularizado)*

Figura 5. Comparación de Accuracy: Sin Regularización vs Regularizado

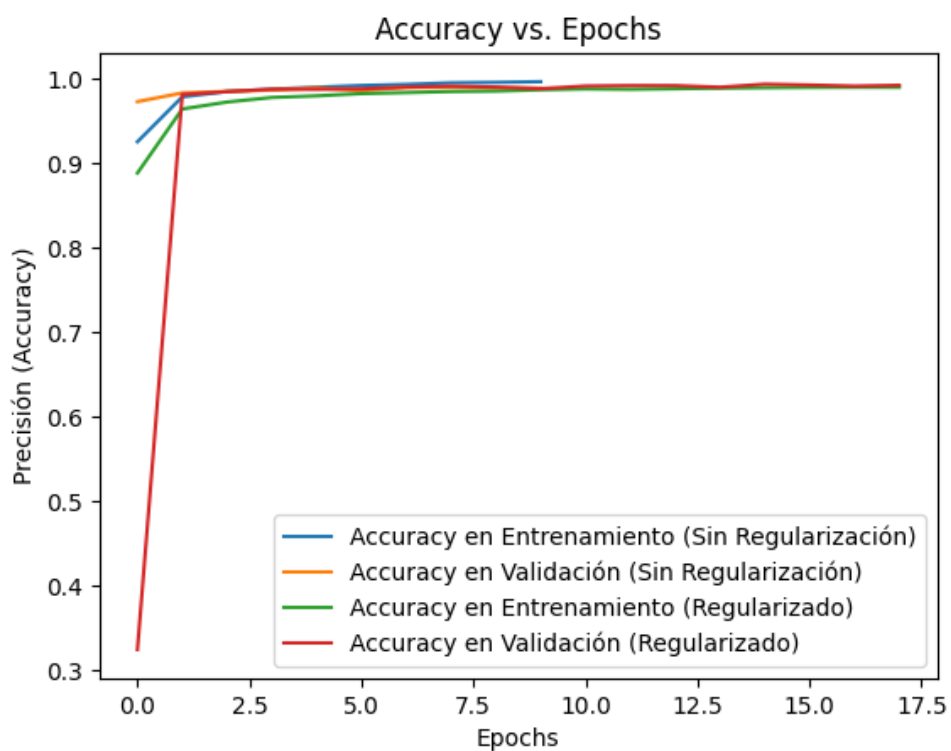


Figura 6. Comparación de Loss: Sin Regularización vs Regularizado

