

TAREA 8: PROPUESTA Y ENTRENAMIENTO DE RED NEURONAL

Marcos Rafael Roman Salgado¹

¹UACH, Texcoco de Mora – <marcosrafael2000@hotmail.com>

RESUMEN

Este reporte presenta una comparación entre dos arquitecturas de redes neuronales convolucionales vistas en clase para la clasificación de imágenes en los conjuntos de datos CIFAR-10 y MNIST. Adicionalmente, se propone una nueva arquitectura optimizada basada en las observaciones y resultados obtenidos. Los resultados experimentales muestran que la red propuesta mejora significativamente la precisión en ambos conjuntos de datos.

Palabras-clave: Arquitecturas de Redes Neuronales, Clasificación de Imágenes, Aprendizaje Profundo, CIFAR-10, MNIST, Optimización, Dropout, Mejora de Precisión

ABSTRACT

This report presents a comparison between two Convolutional Neural Network (CNN) architectures studied in class for image classification on the CIFAR-10 and MNIST datasets. Additionally, a new optimized architecture based on the observations and results obtained is proposed. Experimental results show that the proposed network significantly improves accuracy on both datasets.

Keywords: Neural Network Architectures, Image Classification, Deep Learning, CIFAR-10, MNIST, Optimization, Dropout, Accuracy Improvement

1. INTRODUCCIÓN

Las redes neuronales convolucionales (CNN) han demostrado ser altamente efectivas en tareas de visión por computadora, particularmente en la clasificación de imágenes. En este trabajo, se comparan dos arquitecturas de CNN y se propone una nueva arquitectura mejorada. Los conjuntos de datos utilizados son CIFAR-10 y MNIST, los cuales son benchmarks comunes en el campo del aprendizaje profundo.

Atencion

Es crucial comprender las diferencias y mejoras potenciales en las arquitecturas de redes neuronales para optimizar el rendimiento en tareas de clasificación de imágenes. La experimentación y el análisis de múltiples arquitecturas permiten identificar configuraciones más eficientes y precisas.

2. MATERIALES Y MÉTODOS

2.1. Conjuntos de Datos

- CIFAR-10:** Conjunto de datos que consta de 60,000 imágenes de 32x32 píxeles en 10 clases, con 6,000 imágenes por clase.
- MNIST:** Base de datos de dígitos escritos a mano, que contiene 70,000 imágenes en escala de grises de 28x28 píxeles, distribuidas en 10 clases.

2.2. Arquitecturas de Redes Neuronales

2.2.1. Modelo 1: Arquitectura Vista en Clase

Código 1. Modelo 1

```
1 from keras import layers
2 from keras import models
3
4 model = models.Sequential()
5 model.add(layers.Conv2D(32, (3,3),
6     activation='relu', padding='same', input_shape=(32, 32, 3)))
7 model.add(layers.MaxPooling2D(
8     pool_size=(2,2)))
9
10 # Segunda capa
11 model.add(layers.Conv2D(64, (3, 3),
12     activation='relu', padding='same'))
13 model.add(layers.MaxPooling2D(
14     pool_size=(2, 2)))
15
16 # Aplanado y calificación
```

```

13 model.add(layers.Flatten())
14 model.add(layers.Dense(512,
    activation='relu'))
15 model.add(layers.Dense(10,
    activation='softmax'))

```

2.2.2. Modelo 2: Otra Arquitectura Vista en Clase

Código 2. Modelo 2

```

1 model2 = models.Sequential()
2 model2.add(layers.Conv2D(32, (3, 3)
    , activation='relu', padding='
    same', input_shape=(32, 32, 3)))
3 model2.add(layers.MaxPooling2D(
    pool_size=(2, 2)))
4 model2.add(layers.Dropout(0.3))
5 model2.add(layers.Conv2D(64, (3, 3)
    , activation='relu', padding='
    same'))
6 model2.add(layers.MaxPooling2D(
    pool_size=(2, 2)))
7 model2.add(layers.Dropout(0.3))
8 model2.add(layers.Flatten())
9 model2.add(layers.Dense(512,
    activation='relu'))
10 model2.add(layers.Dropout(0.3))
11 model2.add(layers.Dense(10,
    activation='softmax'))

```

2.2.3. Modelo Propuesto

Código 3. Modelo Propuesto

```

1 from keras import layers
2 from keras import models
3
4 model3 = models.Sequential()
5 model3.add(layers.Conv2D(32, (3, 3)
    , activation='relu', padding='
    same', input_shape=(32, 32, 3)))
6 model3.add(layers.Conv2D(32, (3, 3)
    , activation='relu', padding='
    same'))
7 model3.add(layers.MaxPooling2D(
    pool_size=(2, 2)))
8 model3.add(layers.Dropout(0.25))
9 model3.add(layers.Conv2D(64, (3, 3)
    , activation='relu', padding='
    same'))
10 model3.add(layers.Conv2D(64, (3, 3)
    , activation='relu', padding='

```

```

    same'))
11 model3.add(layers.MaxPooling2D(
    pool_size=(2, 2)))
12 model3.add(layers.Dropout(0.25))
13 model3.add(layers.Conv2D(128, (3,
    3), activation='relu', padding='
    same'))
14 model3.add(layers.MaxPooling2D(
    pool_size=(2, 2)))
15 model3.add(layers.Dropout(0.5))
16 model3.add(layers.Flatten())
17 model3.add(layers.Dense(512,
    activation='relu'))
18 model3.add(layers.Dropout(0.5))
19 model3.add(layers.Dense(10,
    activation='softmax'))

```

3. RESULTADOS

3.1. CIFAR-10

3.1.1. Red Neuronal Vista en Clase 1

Exactitud en el conjunto de prueba: 66.6 %

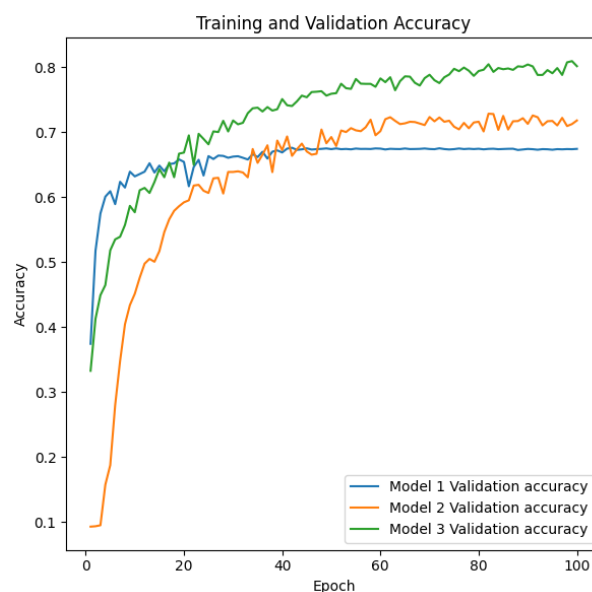
3.1.2. Red Neuronal Vista en Clase 2

Exactitud en el conjunto de prueba: 71.5

3.1.3. Red Neuronal Propuesta

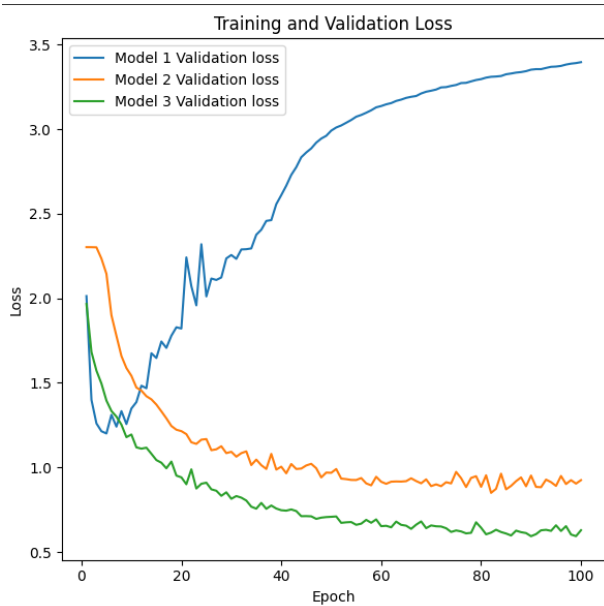
Exactitud en el conjunto de prueba: 80.11 %

Figura 1. Resultados de exactitud de los modelos.(CIFAR-10)



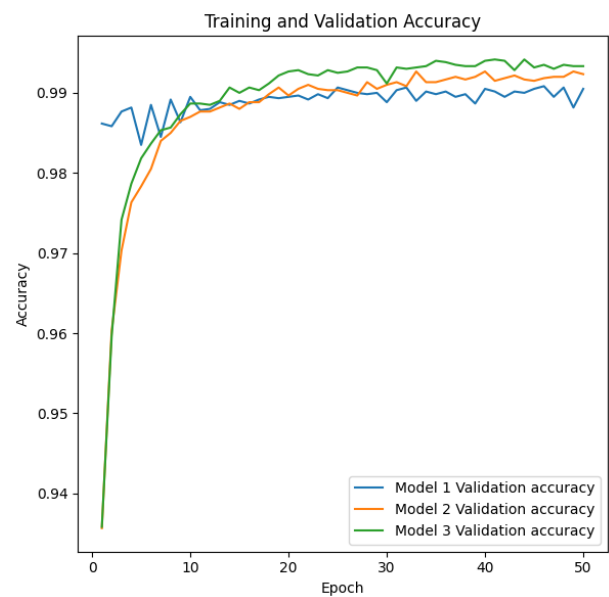
Fuente: autoría propia.

Figura 2. Resultados de valores Loss de los modelos.(CIFAR-10)



Fuente: autoría propia.

Figura 3. Resultados de exactitud de los modelos. (MNIST)



Fuente: autoría propia.

3.2. MNIST

3.2.1. Red Neuronal Vista en Clase 1

Exactitud en el conjunto de prueba: 99.08 %

3.2.2. Red Neuronal Vista en Clase 2

Exactitud en el conjunto de prueba: 99.33 %

3.2.3. Red Neuronal Propuesta

Exactitud en el conjunto de prueba: 99.55 %

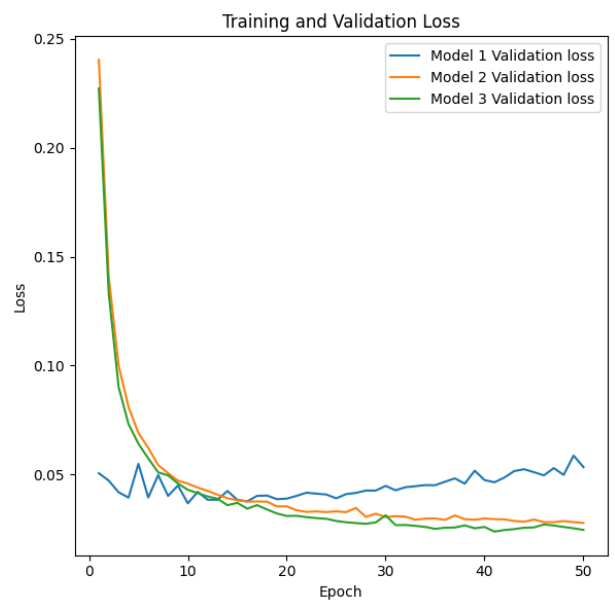
4. DISCUSIÓN

Los resultados experimentales muestran que la red neuronal convolucional propuesta supera significativamente a las redes neuronales simples y a las vistas en clase tanto en CIFAR-10 como en MNIST. La adición de más capas convolucionales y de regularización a través de dropout ha permitido una mejor generalización y precisión.

5. CONCLUSIONES

En este trabajo, se han comparado dos arquitecturas de redes neuronales vistas en clase y se ha propuesto una nueva arquitectura optimizada. La red propuesta ha demostrado ser más efectiva para la clasificación de imágenes en los conjuntos de datos CIFAR-10 y MNIST, alcanzando mayores niveles de precisión.

Figura 4. Resultados de valores Loss de los modelos. (MNIST)



Fuente: autoría propia.

APÉNDICES

Repositorio

En este artículo, el código relacionado se encuentra en el siguiente repositorio: https://github.com/MarcosRoms/Deep_learning_colab.

■ Contacto

Para cualquier duda favor de contactar:

✉ marcosrafael2000@hotmail.com

☎ +52 5537149673

■ Referencias

Repositorio del código en GitHub.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. Communications of the ACM, 60(6), 84–90. Recuperado de <<https://dl.acm.org/doi/10.1145/2207676.2207686>>

Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Proceedings of the International Conference on Learning Representations (ICLR). Recuperado de <<https://arxiv.org/abs/1409.1556>>

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. Recuperado de <<https://arxiv.org/abs/1512.03385>>

Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*. Technical Report. Recuperado de <<https://www.cs.toronto.edu/~kri/learning-features-2009-TR.pdf>>

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). *Gradient-Based Learning Applied to Document Recognition*. Proceedings of the IEEE, 86(11), 2278–2324. Recuperado de <<https://ieeexplore.ieee.org/document/726791>>

LeCun, Y., & Bengio, Y. (1995). *Convolutional Networks for Images, Speech, and Time-Series*. The MIT Press. Recuperado de <https://www.cs.nyu.edu/~yann/talks/lecun_springer.pdf>