
Proyecto Final Statistical Learning II 2022

Anonymous Author(s)

Affiliation

Address

email

Abstract

En el siguiente artículo se presentan 3 aplicaciones con distintos algoritmos de deep learning en los cuales se comparan sus resultados y niveles de precisión, todo esto enfocado en redes neuronales y teniendo como principales objetos de estudio las redes neuronales multicapa perceptron (MLP), redes neuronales convolucionales (CNN) y redes neuronales recurrentes usando datasets como MNIST y conjuntos de datos de tweets.

1 Introducción

Deep Learning es un derivado de Machine learning además de ser parte de la inteligencia artificial, en donde hay distintos tipos de algoritmos y en estos algoritmos destacan las redes neuronales las cuales funcionan como el cerebro de un humano y pueden ser usadas para aprender diferentes cosas y empleados en muchos campos.

Las redes neuronales en un principio fueron creadas para simular los sistemas nerviosos tanto de humanos como de animales y funcionando como nodos conectados unos con otros y a medida que la tecnología fue avanzando se convirtieron en modelos computacionales que procesan información imitando el funcionamiento de las neuronas humanas y su objetivo es ayudar a que los sistemas computacionales puedan funcionar como máquinas de aprendizaje y pensamiento y es a partir de esto que nace la idea de "Inteligencia Artificial".

Y es entonces donde nacen también diferentes algoritmos que se pueden aplicar para diferentes tareas como lo son las redes neuronales perceptrón multicapa (MLP), las redes neuronales convolucionales (CNN) o las redes neuronales recurrentes (RNN) que veremos en este artículo.

Please read the instructions below carefully and follow them faithfully.

2 Metodología

Para aplicar cualquiera de estos métodos primero tendremos que optar por un dataset, el cual se tiene que adaptar para un algoritmo como lo podría ser un dataset MNIST que se adapta de excelente forma a una red MLP y a una red CNN, además de un dataset de tweets que lo podemos usar perfectamente en una RNN.

2.1 Red Neuronal Perceptron Multicapa (MLP)

Ya que se tienen un dataset que en este caso es un MNIST de números dibujados a mano alzada, se procede a hacer la transformación de la data, ya que, esta data viene en un formato de tabla de 3 dimensiones de 70,000 x 28 x 28.

31 Para empear se tiene que partir el dataset en 2 para hacer un dataset de entrenamiento y uno de test,
 32 luego se realizan las funciones para una matriz de confusión que luego se utilizará en conjunto con
 33 una funcion de errores para ver el desempeño de los algoritmos utilizados.

34 Para este dataset se realizaron 2 ejercicios, una regresión logística y una red neuronal perceptron
 35 multicapa.

36 El primero fue una regresión logística la cual se utilizó como una red neuronal de 10 neuronas
 37 en donde se utilizó una función de activación softmax, transformando el dataset en 2 dimensiones
 38 de 60000 de los datos de entrenamiento por 784 de los pixeles y las variables y que se usaron de
 39 forma categorica llevando acabo una transformación por medio de One Hot Encoding para volver las
 40 variables numericas a categoricas, además de agregar 30 epocas al modelo y usando el dataset de
 41 validación.

42 Seguido de esto se realizó una red neuronal perceptrón multicapa usando la misma información
 43 transformada con una capa de activación relu de 100 neuronas, una capa oculta de 50 neuronas, una
 44 capa oculta de 25 neuronas y una capa de salida con 10 neuronas y activación softmax.

45 2.2 Red Neuronal Convolucional (CNN)

46 En este caso se utilizó el mismo dataset de MNIST de imágenes de números dibujados a mano
 47 alzada y se hicieron transformaciones diferentes en las que se usaron diferentes dimensiones para los
 48 metodos que se efectuaron. y se identificó que la data estaba equilibrada para todos los números por
 49 lo que se llegó a la conclusión de que era factible aplicar redes neuronales.

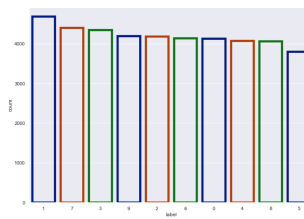


Figure 1: Data equilibrada

50 Para este dataset se realizaron 2 ejercicios, una red neuronal normal artificial y una red convolucional.

51 Entonces, en primera instancia se realizó una red neuronal artifical (ANN) con el objetivo de comparar
 52 los resultados con la red convolucional, para esta ANN se aplico una capa de activación relu con una
 53 capa de batch normalization, seguido de una capa oculta de 24 neuronas y otra capa oculta de 12
 54 neuronas para finalizar con una capa de activación sigmoid.

55 Seguido de esto se realizó la red neuronal convolucional en la cual se aplico una capa de convolucion
 56 con un kernel de 3x3 y una activacion relu seguido de una capa de maxpooling de 2x2 y una capa
 57 adicional de convolucion de 3x3, seguido de esto se aplico otra capa de maxpooling de 2x2 y por
 58 consiguiente un flatten y por ultimo una capa de salida softmax.

```
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d_8 (conv2D)	(None, 26, 26, 32)	320
max_pooling2d_8 (MaxPooling2	(None, 13, 13, 32)	0
conv2d_9 (conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_9 (MaxPooling2	(None, 5, 5, 64)	0
flatten_4 (Flatten)	(None, 1680)	0
dropout_7 (Dropout)	(None, 1680)	0
dense_8 (Dense)	(None, 10)	16910
Total params: 34,826		
Trainable params: 34,826		
Non-trainable params: 0		

Figure 2: Red Neuronal Convolucional

2.3 Red Neuronal Recurrente (RNN)

Para esta instancia se utilizó un dataset de tweets en los que se aplicaron 3 algoritmos, una red neuronal recurrente, una red neuronal artificial de memoria a corto plazo LSTM y una red neuronal cerrada GRU.

Para esto se utilizaron diferentes algoritmos de preprocesamiento de data como lo fue, el removido de stop words, lematización y tokenización, además se hizo el split de data de entrenamiento y data de test, para luego aplicar la embedding matrix que ayuda para aplicar los modelos recurrentes.

En el RNN simple, se aplico la matriz embedding, se aplico una capa rnn y una capa de activación softmax, para la LSTM se aplico la matriz embedding, se aplico una capa lstm y una capa de activación softmax y para la GRU se aplico la matriz embedding, se aplico una capa gru y una capa de activación softmax,

3 Resultados

Para los resultados se aplico un algoritmo implementado en python para el histórico de modelos ejecutados el cual utiliza ModelCheckpoint y earlystopping para poder guardar los resultados.

3.1 Red Neuronal Perceptron Multicapa (MLP)

Debido a que se aplicaron dos modelos para hacer una comparación se pudo constatar que la regresión logística cumple muy bien su trabajo sin embargo la red neuronal perceptrón multicapa arroja mejores resultados siendo estos mejores para el aprendizaje de diferentes campos. Además, para el MLP se utilizaron las capas de activación Relu, Tahn, Sigmoid y Selu, teniendo resultados parecidos pero con dos modelos que funcionaron de una mejor forma, relu y selu.

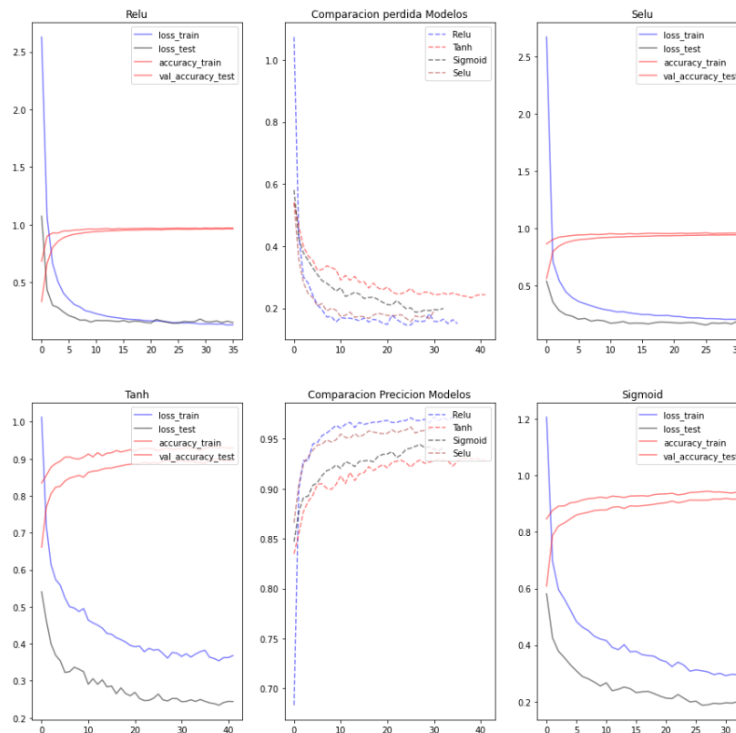


Figure 3: Resultados MLP

3.2 Red Neuronal Convolucional (CNN)

Para este problema se hicieron dos modelos teniendo resultados sorprendentemente parecidos ya que ANN tuvo un accuracy de 0.98 respecto de los 0.99 de CNN aunque la red neuronal artificial simple no tuvo el mismo resultado que la CNN, da un buen indicio de que esta red neuronal es util para clasificación.

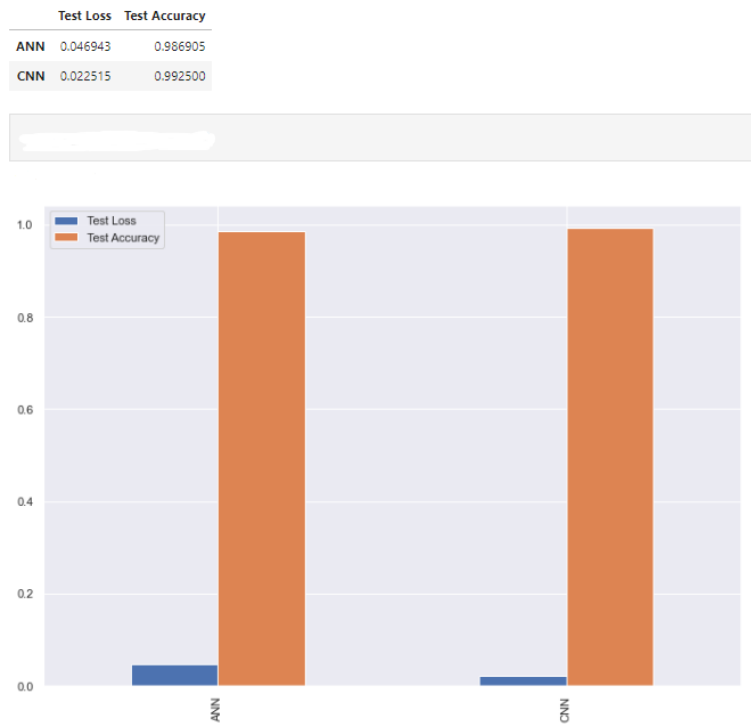


Figure 4: Resultados CNN

Además se pueden observar los cambios que se obtuvo en la data conforme se aplicaban los filtros.

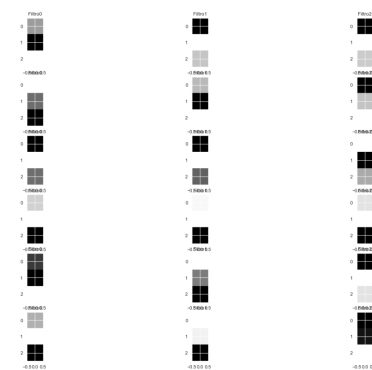


Figure 5: Filtros CNN

3.3 Red Neuronal Recurrente (RNN)

Para la red neuronal recurrente se tuvieron buenos resultados en general ya que todos los modelos actuaron de una buena forma y tienen buenas puntuaciones aunque LSTM y GRU obtuvieron mejor accuracy llegando a un 0.77 los dos con respecto del 0.74 de la red neuronal recurrente simple.

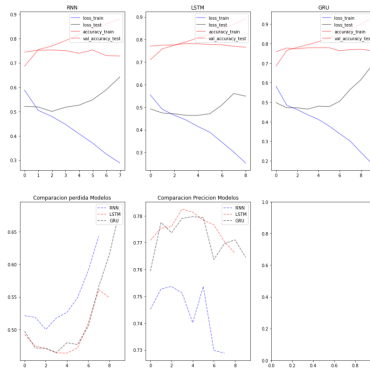


Figure 6: Resultados RNN

4 Conclusiones

Las redes neuronales son de gran ayuda al querer resolver problemas en los que no se conoce una variable y se requiere predecir, además de ser más óptimos en ciertos casos pero también evidenciando que algunos algoritmos de machine learning pueden funcionar de cierta forma para algunos problemas como en MNIST, si bien es cierto solo pueden ver los números de una misma forma, pueden funcionar para problemas más simples.

Las redes MLP funcionan muy bien a la hora de hacer clasificaciones al igual que las CNN con la diferencia que las CNN se pueden aplicar para filtros y detección de objetos y las RNN funcionan de una excelente forma en el análisis de sentimientos.

References

[1] Juan, I.B. & Big Data, M.C. (2019) Matriz de confusión y sus metricas. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

A Appendix

Articulo realizado por Huber Torres para el Instituto de investigación de operaciones de la Universidad Galileo, Guatemala, 2022