

Práctica de laboratorio 2: Aplicaciones de redes neuronales con Texto

Castro Elvira D. ²⁰²²⁷¹⁰⁰³⁹, Pineda Rugerio N. ²⁰²²⁷¹⁰²⁴⁰, Castañón Hernández V. J. ²⁰²²⁷¹⁰⁰²⁰,
Galicia Cocoltzi N. ²⁰²²⁷¹⁰²³⁴, Sánchez Zanjuampa M. A. ²⁰²²⁷¹⁰⁰²⁹, y Nava Mendez E. U. ²⁰²¹⁷¹⁰¹⁴⁴.

Profesor: Lauro Reyes Cocoltzi

Resumen—Las redes neuronales forman parte de las técnicas de aprendizaje automático para el procesamiento de datos. Su entrenamiento implica la selección adecuada de hiperparámetros como el tamaño del lote y el número de épocas, los cuales juegan un papel fundamental en el logro de un aprendizaje óptimo. Para ajustar estos hiperparámetros, se realizan pruebas combinando diferentes valores con el fin de analizar su comportamiento y determinar cuáles son los más adecuados para el entrenamiento de la red neuronal. Este proceso de ajuste de hiperparámetros es crucial para mejorar el rendimiento del modelo y garantizar una generalización efectiva a nuevos datos.

Palabras clave—Redes neuronales, épocas, Batch size, Loss, validation, overfitting, underfitting, procesamiento de texto

I. INTRODUCCIÓN

I-A. Marco teórico

I-A1. ¿Qué es una red neuronal?:

Una red neuronal es un método de la inteligencia artificial que enseña a las computadoras a procesar datos de una manera que está inspirada en la forma en que lo hace el cerebro humano. Se trata de un tipo de proceso de machine learning llamado aprendizaje profundo, que utiliza los nodos o las neuronas interconectados en una estructura de capas que se parece al cerebro humano. Crea un sistema adaptable que las computadoras utilizan para aprender de sus errores y mejorar continuamente. De esta forma, las redes neuronales artificiales intentan resolver problemas complicados, como la realización de resúmenes de documentos o el reconocimiento de rostros, con mayor precisión.

I-A2. Estructura de una red neuronal:

Una red neuronal básica tiene neuronas artificiales interconectadas en tres capas:

- Capa de entrada. La información del mundo exterior entra en la red neuronal artificial desde la capa de entrada. Los nodos de entrada procesan los datos, los analizan o los clasifican y los pasan a la siguiente capa. Capa oculta
- Capa oculta. Las capas ocultas toman su entrada de la capa de entrada o de otras capas ocultas. Las redes neuronales artificiales pueden tener una gran cantidad de capas ocultas. Cada capa oculta analiza la salida de la capa anterior, la procesa aún más y la pasa a la siguiente capa. Capa oculta
- Capa de salida. La capa de salida proporciona el resultado final de todo el procesamiento de datos que realiza la red

neuronal artificial. Puede tener uno o varios nodos. Por ejemplo, si tenemos un problema de clasificación binaria (sí/no), la capa de salida tendrá un nodo de salida que dará como resultado 1 o 0. Sin embargo, si tenemos un problema de clasificación multiclase, la capa de salida puede estar formada por más de un nodo de salida.

I-A3. Embeddings:

Son representaciones numéricas de objetos del mundo real que los sistemas de machine learning (ML) e inteligencia artificial (IA) utilizan para comprender dominios de conocimiento complejos, como lo hacen los humanos. Permiten que los modelos de aprendizaje profundo entiendan los dominios de datos del mundo real de manera más eficaz. Simplifican la forma en que se representan los datos del mundo real y, al mismo tiempo, conservan las relaciones semánticas y sintácticas.

I-A4. GlobalAveragePooling1D:

se trata de una capa para el procesamiento de datos unidimensionales, como secuencias temporales o series de tiempo. Esta capa toma como entrada un tensor tridimensional de forma (batch_size, steps, features) y calcula la media a lo largo del eje de pasos (steps), produciendo un tensor bidimensional de forma (batch_size, features).

I-A5. Binary_crossentropy:

Se trata de una función de pérdida utilizada comúnmente en problemas de clasificación binaria en redes neuronales. Es especialmente útil cuando se está realizando una tarea de clasificación donde cada instancia puede pertenecer a una de dos clases mutuamente excluyentes, como "positivo" o "negativo", "sí" o "no", "1" o "0".

I-A6. Dataset:

Se trata de un conjunto de datos para la clasificación binaria de sentimientos que contiene muchos más datos que los anteriores conjuntos de datos de referencia. Contiene un conjunto de 25.000 críticas de películas altamente polares para el entrenamiento y 25.000 para las pruebas. Contiene dos características, la reseña y si es positiva o negativa

I-B. Objetivos

- Comprender el proceso de tokenización, limitación del tamaño del vocabulario y padding de secuencias para preparar los datos de texto para su uso en modelos de aprendizaje automático.
- Adquirir experiencia en la construcción de modelos de redes neuronales para tareas de clasificación binaria de sentimientos, utilizando capas de Embedding, GlobalAveragePooling1D, capas densas y capas de salida con activaciones adecuadas.
- Experimentar con técnicas como regularización, dropout, ajuste de hiperparámetros y el uso de modelos pre-entrenados para optimizar y mejorar el rendimiento del modelo de clasificación de sentimientos.

I-C. Aporte

- La práctica permite adquirir experiencia práctica en la manipulación de datos, preprocesamiento, construcción de modelos de redes neuronales y optimización de modelos para tareas específicas como la clasificación de sentimientos.
- Se aplica y refuerza el conocimiento teórico adquirido en machine learning en un contexto práctico y realista, lo que ayuda a consolidar la comprensión de conceptos como embeddings, capas densas, funciones de activación y optimizadores.
- Se experimenta con diferentes estrategias para mejorar el rendimiento del modelo, como regularización, dropout, ajuste de hiperparámetros y el uso de modelos pre-entrenados, lo que amplía la comprensión sobre cómo optimizar y ajustar modelos de aprendizaje automático.

II. DESARROLLO

1. Carga y preprocesamiento del conjunto de datos IMDB

- Se carga el conjunto de datos IMDB proporcionado por Keras.
- Se realiza un preprocesamiento que incluye la tokenización de palabras, limitación del tamaño del vocabulario, padding de secuencias para tener la misma longitud y etiquetado de sentimientos (positivo o negativo).

2. Creación de un diccionario de mapeo de palabras

- Se crea un diccionario que mapea enteros a palabras y viceversa, lo que permite decodificar las secuencias de enteros de vuelta a texto humano legible

3. Modelo de clasificación de reseñas

- Se define un modelo de red neuronal para la clasificación de sentimientos.
- El modelo consta de una capa de Embedding, una capa de GlobalAveragePooling1D, una capa densa con activación ReLU y una capa de salida con activación sigmoide.

- El modelo se compila utilizando el optimizador Adam, la función de pérdida binary_crossentropy y se monitorea la precisión como métrica.

III. RESULTADOS

Se realizaron varias pruebas para diferentes valores en las épocas(Epochs) y tamaño de muestra (Batch Size). De las cuales obtuvimos los siguientes resultados para la precisión y y perdida.

Tabla I
 PRECISIÓN Y PERDIDA

Prueba	Epochs	Batch Size	Accuracy	Loss
1	20	50	0.9852	0.0539
2	20	512	0.9174	0.2177
3	40	512	0.9628	0.1173
4	40	1024	0.9351	0.1863
5	80	100	0.9997	0.0017
6	80	246	0.9985	0.0135
7	100	100	0.9999	0.0036
8	100	246	0.9997	0.0040
9	100	512	0.9969	0.0236
10	100	1000	0.9836	0.0664
11	512	512	1	0.00006

La siguiente tabla muestra los valores obtenidos para las validaciones de precisión y perdida.

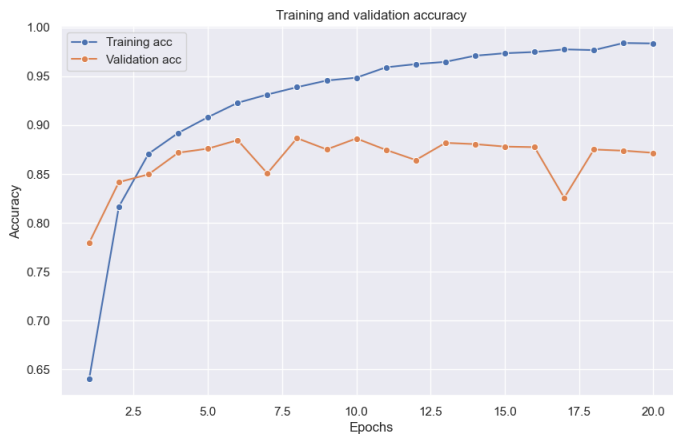
Tabla II
 VALIDACIÓN DE PRECISIÓN Y PERDIDA

Prueba	Validation Accuracy	Validation Loss
1	0.8716	0.4173
2	0.8843	0.2889
3	0.8847	0.2970
4	0.8847	0.2867
5	0.8569	0.9539
6	0.8651	0.6083
7	0.8567	0.9302
8	0.8579	0.7864
9	0.8694	0.4905
10	0.8785	0.3524
11	0.8563	1.3011

III-A. Prueba 1: 20 épocas, 50 Batch Size

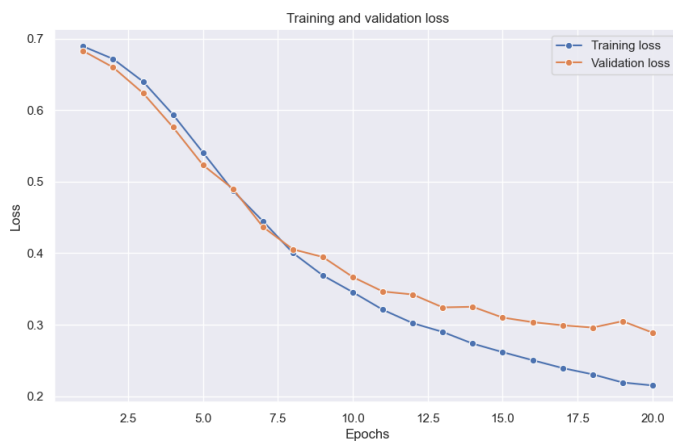


En la prueba 1, observamos que el modelo por la parte de training loss decae de manera constante sin tener grandes picos de incremento o decremento, pareciera que se entrenó correctamente; sin embargo, cuando se verifica el dataset de validación, observamos como va en decremento y en un punto empieza a incrementar, esto nos quiere decir que entrenamos por mucho tiempo (épocas) el modelo y tiene un overfitting. Para abordar el overfitting, se pueden considerar varias estrategias, como la regularización (por ejemplo, L2 regularization), el uso de dropout, la reducción del tamaño del modelo, la recopilación de más datos de entrenamiento o la aplicación de técnicas de aumento de datos. Estas estrategias pueden ayudar a mejorar la generalización del modelo y reducir el efecto del sobreajuste.



En la parte de la validación, vemos que se comporta de manera adecuada el modelo, no es ideal porque como anteriormente observamos en la gráfica de pérdida que tiene un overfitting, por lo que no es muy fiable el modelo, para mayor exactitud se tiene que hacer un análisis de la matriz de confusión para obtener mayor información

III-B. Prueba 2: 20 épocas, 512 Batch Size



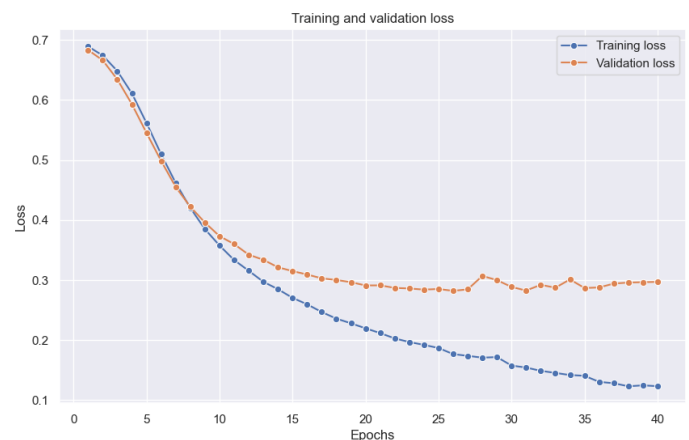
Analizando la grafica, tiene un buen comportamiento de las dos partes analizadas (pérdida de entrenamiento y validación),

observamos como decrecen de manera constante y el gap entre las dos gráficas es pequeño



Nuevamente, observamos que se tiene un buen comportamiento de la gráfica, observamos que el gap entre las gráficas es pequeño, y en general se comportó de manera más confiable el modelo

III-C. Prueba 3: 40 épocas, 512 Batch Size

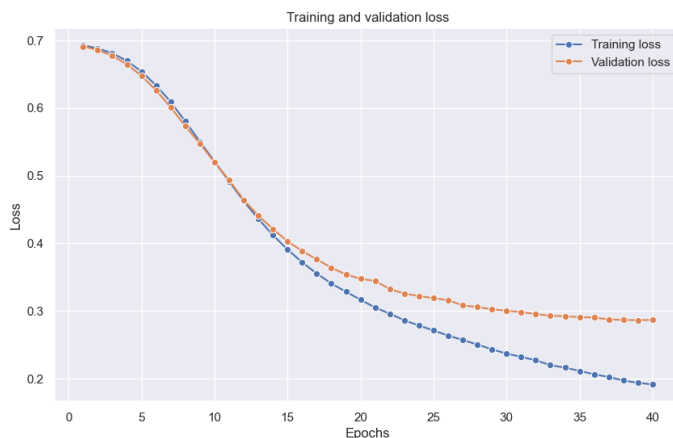


Aquí observamos que el modelo tuvo buenos resultados entrenando, pero llegando la validación no le fue tan bien, podemos decir que el modelo no está generalizando de manera correcta, es un mal modelo.



En el análisis de la precisión observamos que el gap generado entre las dos gráficas es notable, ha este modelo se puede explicar su comportamiento por la falta de muestras del dataset de entrenamiento, o existe una diferencia entre el dataset de entrenamiento y el dataset de pruebas

III-D. Prueba 4: 40 épocas, 1024 Batch Size

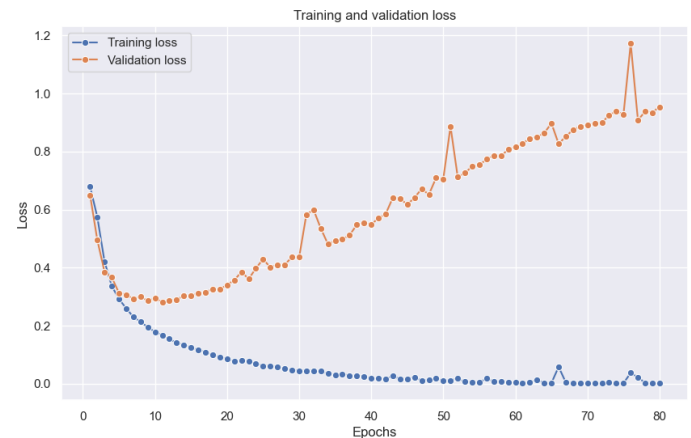


Tenemos un comportamiento parecido al caso anterior, tenemos un gap entre las dos pérdidas; sin embargo, observamos que es más constante la perdida, por lo que el modelo requiere algunos ajustes, pero no es tan grave.



En la parte de la precisión observamos que se alcanza un buen número, como se mencionó, el modelo en general tiene un buen comportamiento, hace falta obtener mas amétricas como la matriz de confusión para afirmar que el modelo tiene un comportamiento bueno

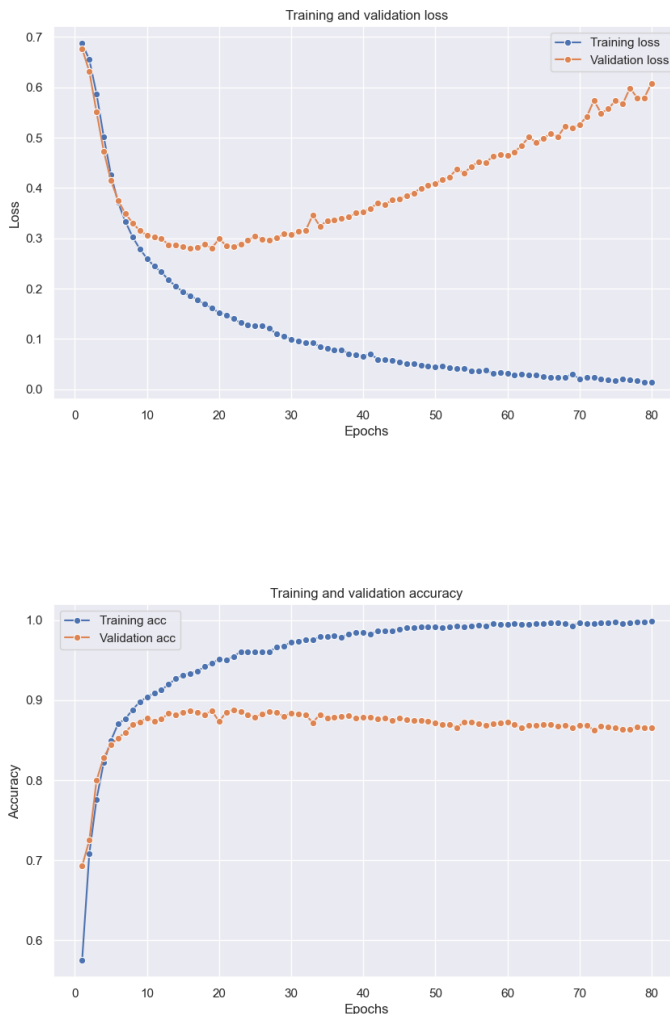
III-E. Prueba 5: 80 épocas, 100 Batch Size



Observamos un comportamiento parecido a la prueba 1; sin embargo, aquí es más notable y es más grave, como se mencionó anteriormente, este comportamiento se puede explicar por

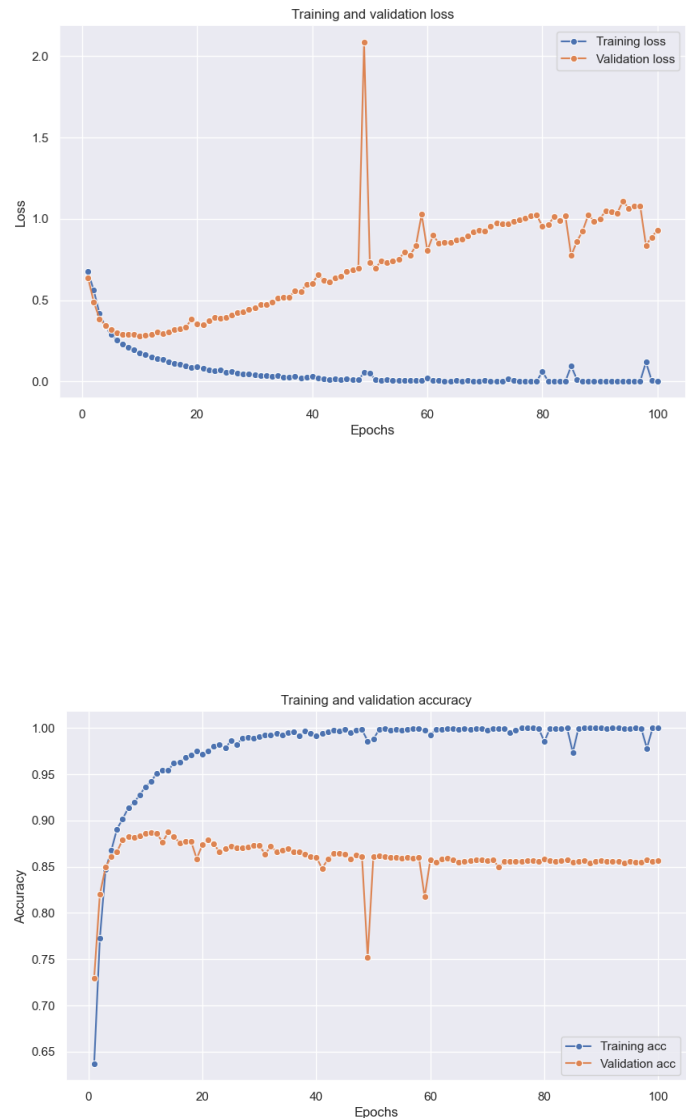
el tiempo de entrenamiento, donde podemos aplicar técnicas como regularización (por ejemplo, L2 regularization), el uso de dropout, la reducción del tamaño del modelo, la recopilación de más datos de entrenamiento o la aplicación de técnicas de aumento de datos. Además, tenemos saltos abruptos que pueden ser explicados por el mismo sobre ajuste que se tiene y está memorizando los datos de entrenamiento. El modelo puede sobre ajustar durante algunas épocas antes de corregirse y volver a generalizar mejor.

III-F. Prueba 6: 80 épocas, 246 Batch Size



Tenemos un comportamiento muy similar al caso anterior, las recomendaciones son las mismas para corregir los errores presentados, aquí es un crecimiento constante y sin tantos saltos abruptos

III-G. Prueba 7: 100 épocas, 100 Batch Size



Obtenemos un comportamiento similar al caso anterior, pero aquí los saltos son muy importantes, esto también tiene una explicación a que el tamaño del lote utilizado durante el entrenamiento es grande, es posible que el modelo no esté generalizando bien en cada paso de la época y esté "saltando" a través de los gradientes. Esto puede hacer que el validation loss tenga saltos abruptos, ya que la evaluación se realiza en un conjunto de datos diferente al de entrenamiento y podría revelar problemas de generalización.

III-H. Prueba 8: 100 épocas, 246 Batch Size



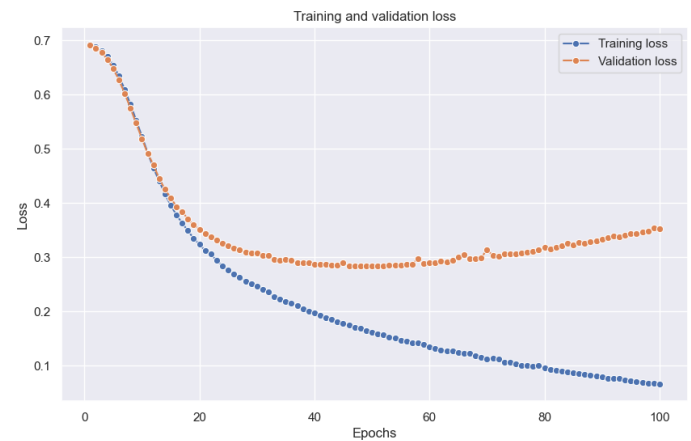
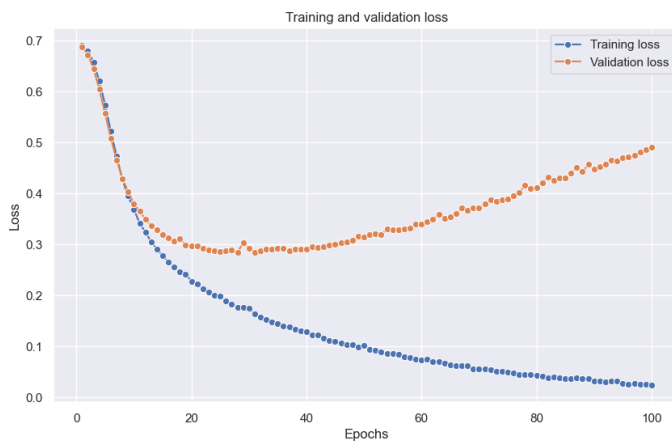
Las gráficas de la prueba 9 muestra un comportamiento igual al de la prueba anterior, solo que en este caso se uso un tamaño de muestras el doble de grande.

III-J. Prueba 10: 100 épocas, 1000 Batch Size



Nuevamente, tenemos un comportamiento similar, observamos que un número de épocas grande afecta de manera negativa al rendimiento del modelo, a pesar de que se cambie el número de pruebas, no se obtiene un mejor rendimiento que los casos anteriores

III-I. Prueba 9: 100 épocas, 512 Batch Size

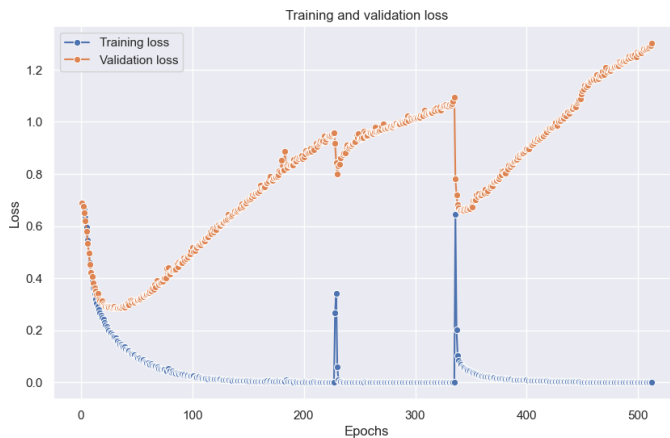


La curva de trainin loss en perdida tiene una caída constante a lo largo de las 100 épocas. Sin embargo la correspondiente a validation loss empieza a divergir a partir de las 40 épocas e incluso comienza a levantarse y comportarse como algunas pruebas anteriores.



En cuanto a la precisión, los valores son en general buenos, aunque es debido a un sobre entrenamiento, pues el tamaño de la muestra fue casi tan grande como el doble del que se planteaba inicialmente.

III-K. Prueba 11: 512 épocas, 512 Batch Size



Para esta prueba podemos observar que la gráfica de training loss cae de manera constante y parece estabilizarse en las 100 épocas. Sin embargo, a partir de ahí existen un par de aumentos puntuales, llegando el máximo hasta un valor de más 0.6. Esto es debido a un sobre-ajuste que se da en algunas épocas; sin embargo, el modelo lo vuelve a corregir casi de inmediato. En cuanto a la validación de la pérdida, la gráfica comienza a divergir en aproximadamente las 40 épocas y las gráficas se van alejando más entres sí. Al igual que ocurrió con la curva de training loss, hay algunas épocas donde ocurre una caída brusca; sin embargo, al modelo le toma más épocas estabilizarse y seguir con su comportamiento.



En cuanto a la precisión del modelo, el entrenamiento alcanza un valor de 1; sin embargo, esto es debido a un sobre ajuste, lo que indica que no es un modelo adecuado. Además, la curva de validación está muy alejada de la curva de training acc, aunque la diferencia parece ser constante. Cabe resaltar que esta prueba se realizó con un exceso de épocas.

IV. CONCLUSIONES

La selección adecuada de hiperparámetros, como el número de épocas y el tamaño del lote (número de muestras), es esencial para el entrenamiento efectivo de una red neuronal de clasificación binaria. Un número limitado de épocas puede resultar en un modelo subentrenado que no ha convergido completamente, lo que genera que no pueda capturar debidamente patrones complejos en los datos. Por otro lado, un número excesivo de épocas puede conducir al sobreajuste, donde el modelo se adapta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos.

La selección de la cantidad adecuada de muestras para el entrenamiento es indispensable, ya que una elección insuficiente puede llevar al modelo a no aprender de forma óptima, mientras que una elección demasiado grande puede llevar a requerir un mayor tiempo de entrenamiento.

Con base en lo anterior, la selección de los valores óptimos para estos hiperparámetros implica realizar pruebas exhaustivas mediante la exploración de diferentes combinaciones de valores. Esto puede implicar ajustar el número de épocas en función del rendimiento del modelo en el conjunto de validación, observando la convergencia de la función de pérdida y monitoreando la métrica de precisión en los datos de validación, lo cual se combina con el tamaño del lote.

Retomando lo anterior, se realizaron una serie de pruebas en donde obtenemos un patrón repetitivo, que indica que si se aplican más de 80 épocas al modelo se obtienen resultados similares, pero si se aplicaba una cantidad mayor se generaba un sobreajuste notable y las precisiones también se comportaban de manera similar, manteniéndose alrededor del 86 %. Por otro lado, cuando se tenía un número menor a 20 épocas, el modelo no era confiable.

REFERENCIAS

- [1] ¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS <https://aws.amazon.com/es/what-is/neural-network/>. Recuperado el 17 de marzo de 2024
- [2] Global Average Pooling <https://paperswithcode.com/method/global-average-pooling>. Recuperado el 17 de marzo 2024
- [3] What is embeddings in machine learning? <https://aws.amazon.com/es/what-is/embeddings-in-machine-learning/#:~:text=Embeddings%20are%20numerical%20representations%20of,knowledge%20domains%20like%20humans%20do..> Recuperado el 17 de marzo 2024
- [4] What is overfitting? <https://aws.amazon.com/es/what-is/overfitting/>. Recuperado el 17 de marzo 2024