

Universidad del Valle De Guatemala

Facultad de Ingeniería

Deep Learning y Sistemas inteligentes



Laboratorio 2

Javier Mombiela 20067

Roberto Ríos 20979

Guatemala, 6 de agosto 2023

Objetivo de la red

El objetivo de la red es de regresión y es poder predecir en base a las variables seleccionadas el índice de aprobación (approval_index) que tendrá una película.

¿Por qué la implementación de una red neuronal puede obtener el resultado deseado?

Estos modelos de redes neuronales pueden proporcionar el resultado deseado ya que aprenden automáticamente características relevantes a partir de los datos proporcionados. También podemos aplicar un enfoque end-to-end y experimentar con los componentes de la red para ver qué aprendizaje es el más indicado. Como funciones de activación, se pretende utilizar ReLu, ya que suele tener gran rendimiento y es selectiva para apagar y encender neuronas que contribuyen al aprendizaje. También se pretende utilizar sigmoid que es muy común para estos casos, por último también se pretende usar tangente hiperbolica, aunque se debe tomar en cuenta que hay casos en los que esta funcion deja apagadas las neuronas

Subconjunto de variables a utilizar:

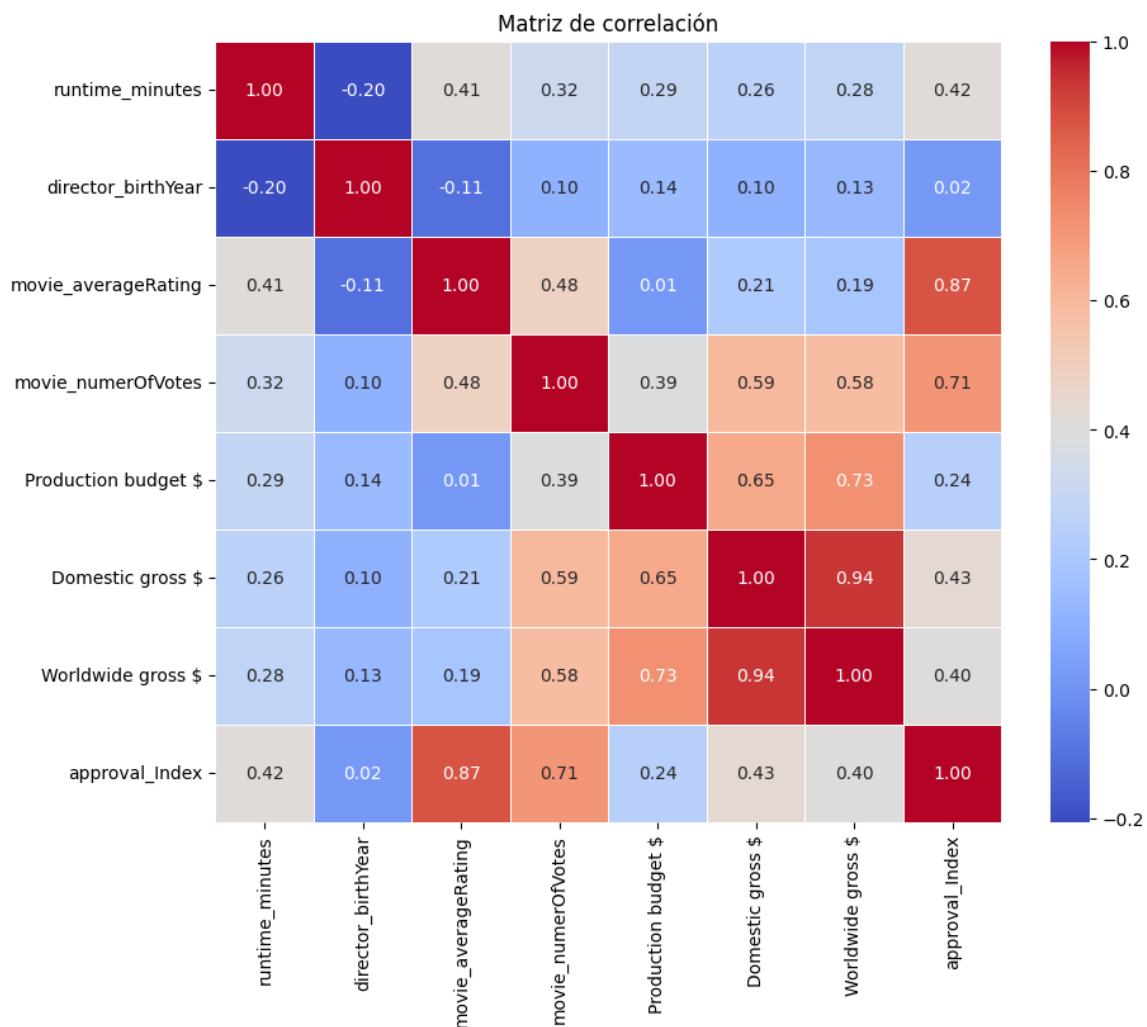


Figura 1: matriz de correlación para la selección de variables.

Para seleccionar las variables a utilizar, se observó que `movie_averageRating` y `movie_numOfVotes` son las que presentan mayor correlación con el `aproval_index`, por lo tanto, estas fueron seleccionadas para hacer las predicciones.

Resultados y composición

- Primer Modelo

Para el primer modelo se utilizaron 6 capas. Como función de activación se implementó `relu` y se optimizó con Batch normalization. Como función de pérdida se empleo MSE y para ir iterando 75 epochs.

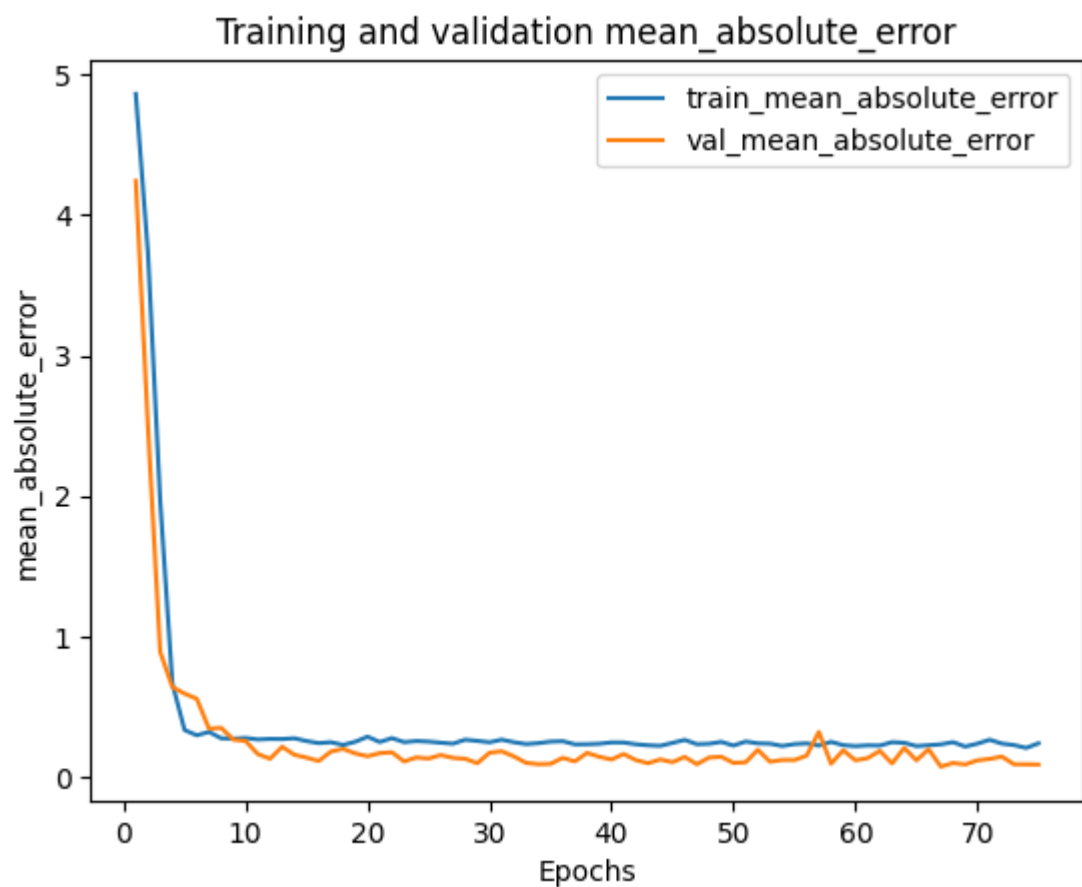


Figura 2: MAE vs epochs del modelo 1.

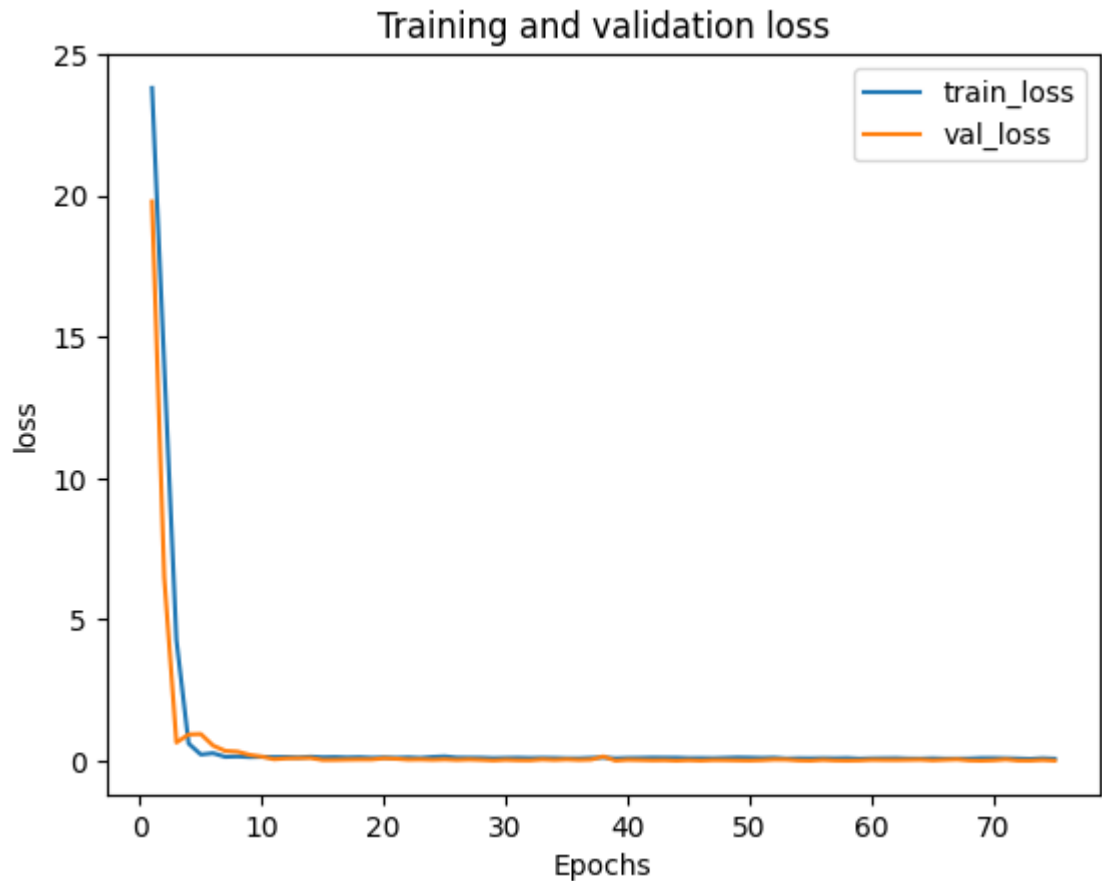


Figura 3: Pérdida vs Epochs del modelo 1.

Podemos observar que durante las 75 iteraciones el error desciende de la forma esperada resultando en las siguientes métricas:

MAE modelo 1: 0.09275183330045365

MSE modelo 1: 0.015734319116595275

R2 modelo 1: 0.9897108754615339

- Segundo Modelo

Para el segundo modelo se utilizaron 3 capas. Como función de activación se implementó sigmoide y se optimizó con Dropout. Como función de pérdida se empleó MSE y para ir iterando 75 epochs.

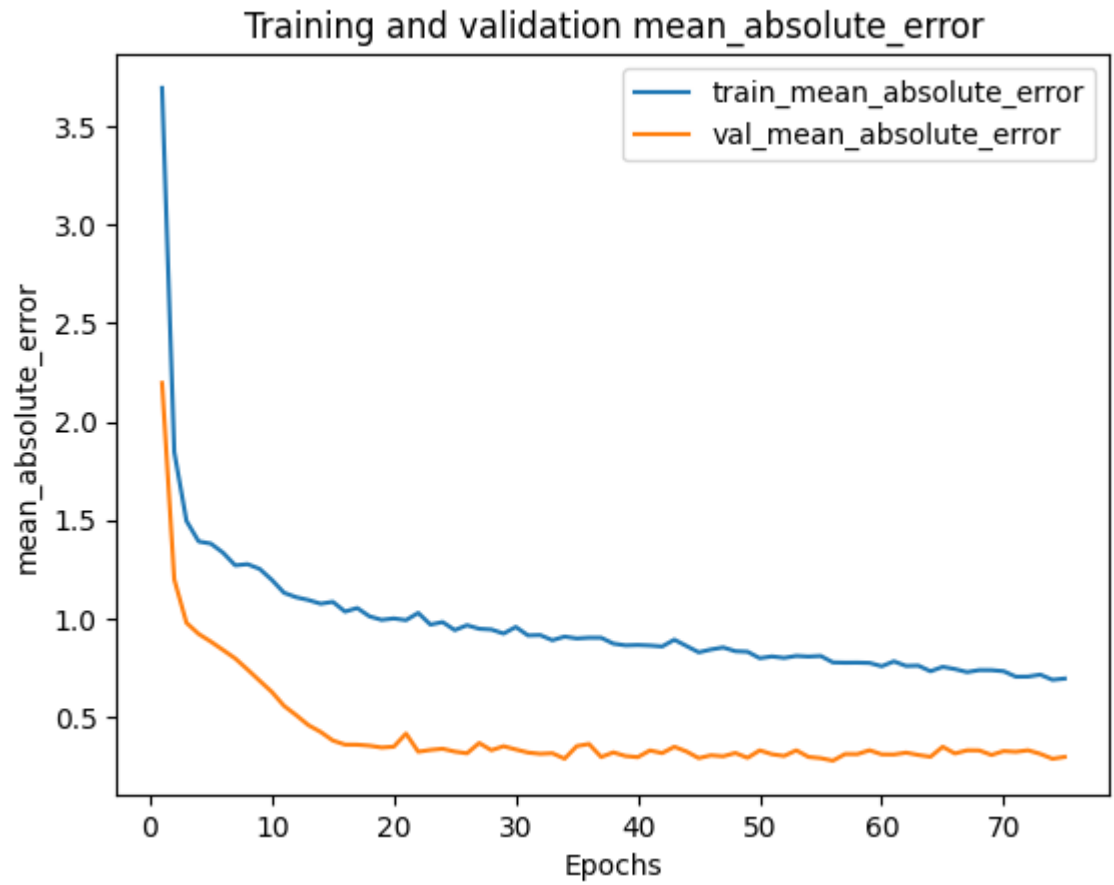


Figura 4: MAE vs Epochs del modelo 2.

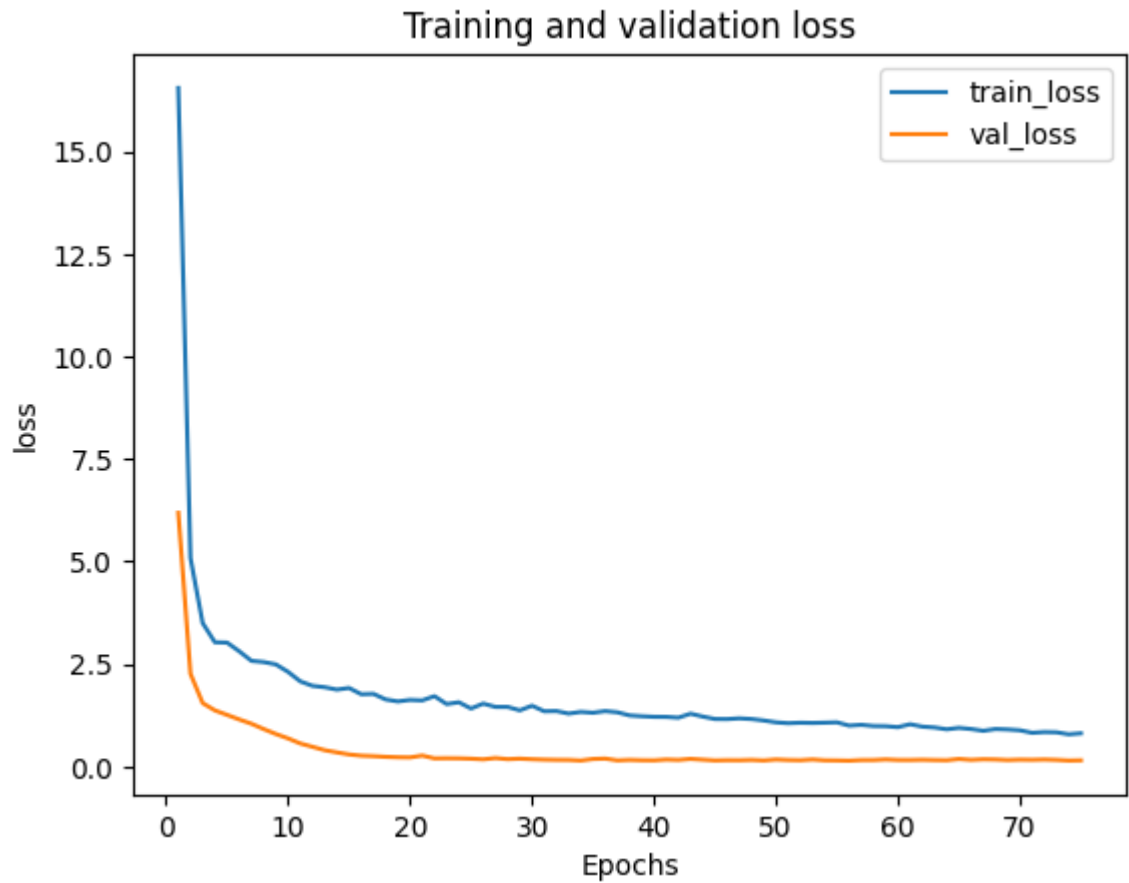


Figura 5: Pérdida vs Epochs del modelo 2.

El modelo 2 tiene las siguientes métricas de rendimiento:

MAE modelo 2: 0.3042071260888859

MSE modelo 2: 0.1811666541818259

R2 modelo 2: 0.864984744497896

- Tercer Modelo

Para el tercer modelo se utilizaron 9 capas. Como función de activación se implementó tangente hiperbólico, se optimizó con L2 como método de regularización. Como función de pérdida se empleó MSE y para ir iterando 75 epochs.

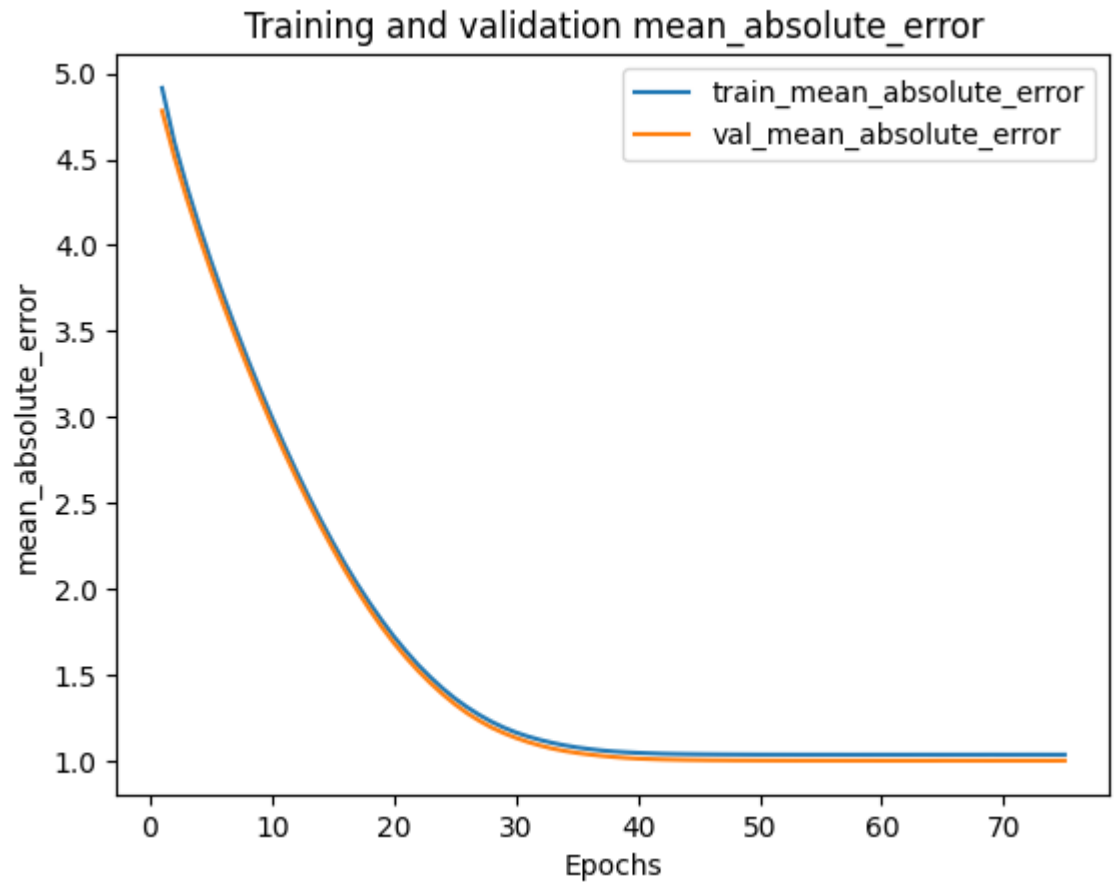


Figura 6: MAE vs Epochs del modelo 3.

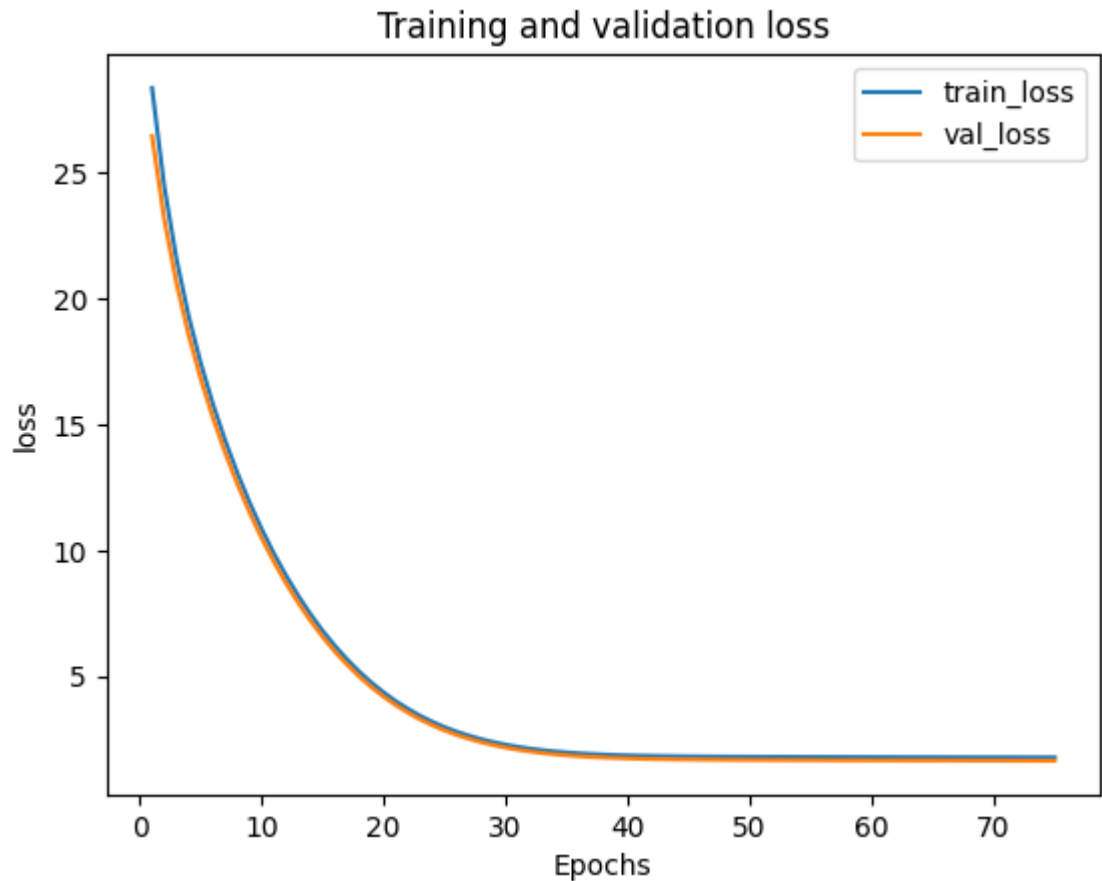


Figura 7: Pérdida vs Epochs del modelo 3.

A este modelo corresponden las siguientes métricas:

MAE modelo 3: 1.0328872582184678

MSE modelo 3: 1.7213675179903856

R2 modelo 3: -7570654406824.367

Discusión

En general, podemos observar en las gráficas que los aprendizajes durante las iteraciones progresan de manera efectiva. Para las métricas de evaluación, se buscó que el MAE y el MSE sean lo más bajos posible, ya que esto indicaría que las predicciones del modelo son cercanas a sus valores esperados. Además, un R2 cercano a 1 indica que el modelo explica bien la variabilidad de los datos.

- Rendimiento

Con respecto al primer modelo tiene un MAE muy bajo (0.0927) y un MSE aún más bajo (0.0157), lo que sugiere que este modelo tiene un buen rendimiento y está muy cerca de los

valores reales en las predicciones. Además, el R2 es alto (0.9897), lo que indica que el modelo explica bien la variabilidad de los datos.

El segundo Modelo tiene un MAE (0.3042) y un MSE (0.1812) más altos en comparación con el primer modelo, lo que indica que sus predicciones están menos cerca de los valores reales. El R2 (0.8650) es también más bajo que el primer modelo, lo que sugiere que el modelo no está explicando tan bien la variabilidad de los datos.

El tercer modelo tiene el MAE (1.0329) y MSE (1.7214) más altos de los tres modelos, lo que indica que sus predicciones están bastante lejos de los valores reales. Además, el R2 es negativo y extremadamente bajo (-7570654406824.367), lo que indica que el modelo es muy pobre y no explica bien la variabilidad de los datos. Este modelo puede estar sobre ajustado.

- Composición

Basándose en estas métricas, el primer modelo parece ser el mejor de los tres, ya que tiene el MAE y el MSE más bajos y un alto R2. La diferencia en el rendimiento puede deberse a varios componentes, como el número de capas (9), las función de activación (Rectified Linear Unit) y las técnicas utilizadas en cada modelo (Batch Normalization, Dropout y L2). El primer modelo está mejor diseñado y tiene una arquitectura más adecuada para el problema en cuestión, lo que le permite obtener un mejor rendimiento en las predicciones.

Se concluye que el primer modelo tiene el mejor rendimiento ya que su función de activación ayuda a mitigar el problema del desvanecimiento del gradiente, un problema común en el entrenamiento de redes neuronales profundas, esto le permite un flujo más fácil de gradientes en comparación con funciones como la función sigmoide.

Al usar la batch normalization, permite la aceleración del entrenamiento, pues permite utilizar tasas de aprendizaje mucho más altas sin el riesgo de divergencia. La normalización por lotes reduce el cambio interno de covariables, lo que significa que la distribución de las entradas de cada capa no cambia tanto durante el entrenamiento. A su vez, la regularización elimina la necesidad de Dropout, una técnica comúnmente utilizada para prevenir el sobreajuste pero que en este laboratorio específicamente no ayudó porque solo empeoró los resultados en los modelos que se implementó.

Conclusión

El primer modelo (9 capas, función de activación ReLu y normalización por lotes) es el más adecuado para este problema ya que presenta las mejores métricas y presenta los resultados más óptimos al predecir los índices de aprobación de las películas.

Referencias

Ioffe, S. & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv, 1. <https://doi.org/10.48550/arXiv.1502.03167>

Repositorio utilizado: <https://github.com/javim7/Deep-Learning>