

Laboratorio 02

Objetivo de la red neuronal

Nuestro objetivo es hacer una regresión de la calificación promedio de una película (average_rating). Luego de un pequeño análisis exploratorio se tomaron en cuenta las siguientes variables para el desarrollo de la Red Neuronal:

- prod_budget
- domestic_gross
- worldwide_gross
- director
- runtime_minutes
- approval_index
- num_votes

Esta decisión puede justificarse al observar la matriz de correlación obtenida (Imagen 1).

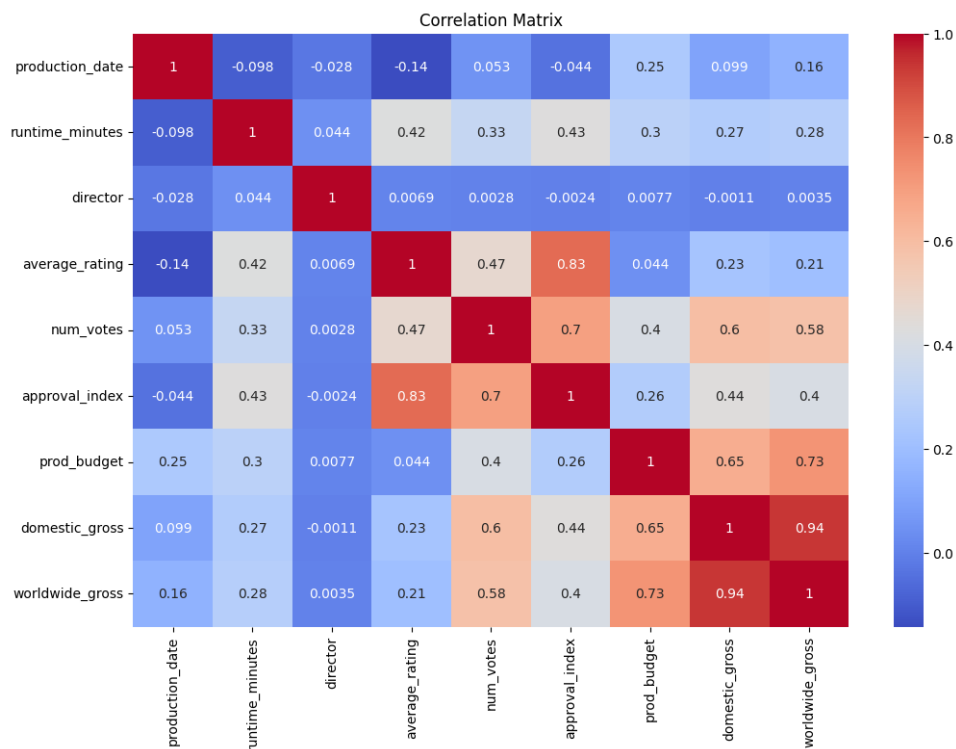


Imagen 1 - Matriz de correlación

Composición y resultados obtenidos

Red Neuronal 1:

Topología:

- Input Layer: 7 neuronas.
- Hidden Layer: 16 neuronas. Activación relu.
- Hidden Layer: 128 neuronas. Activación relu.
- Hidden Layer: 64 neuronas. Activación relu.
- Hidden Layer: 64 neuronas. Activación tanh.
- Output Layer: 1 neurona.

Técnicas de regularización:

- Batch Normalization.

Función de pérdida: MSE

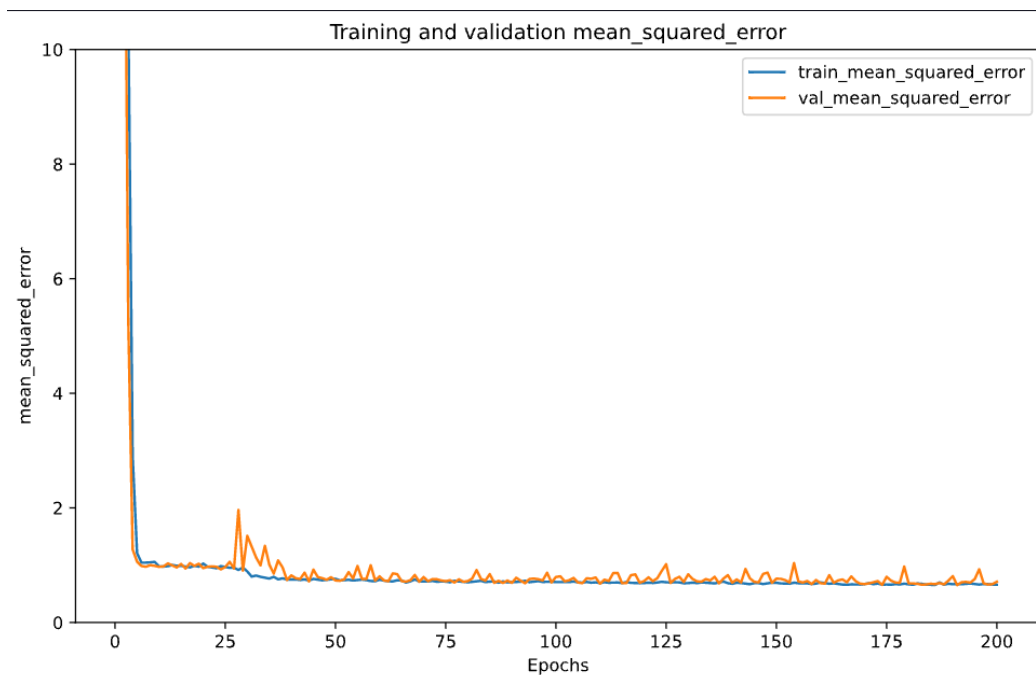


Imagen 2 - Training vs Validation MSE

Validation loss: 0.71

Test loss: 0.63

Red Neuronal 2:

Topología:

- Input Layer: 7 neuronas.
- Hidden Layer: 16 neuronas. Activación relu.
- Hidden Layer: 128 neuronas. Activación relu
- Hidden Layer: 64 neuronas. Activación relu
- Hidden Layer: 32 neuronas. Activación tanh.
- Output Layer: 1 neurona.

Técnicas de regularización:

- Batch Normalization.
- Early stopping con una paciencia de 20 epochs.

Función de pérdida: Mean Absolute Error

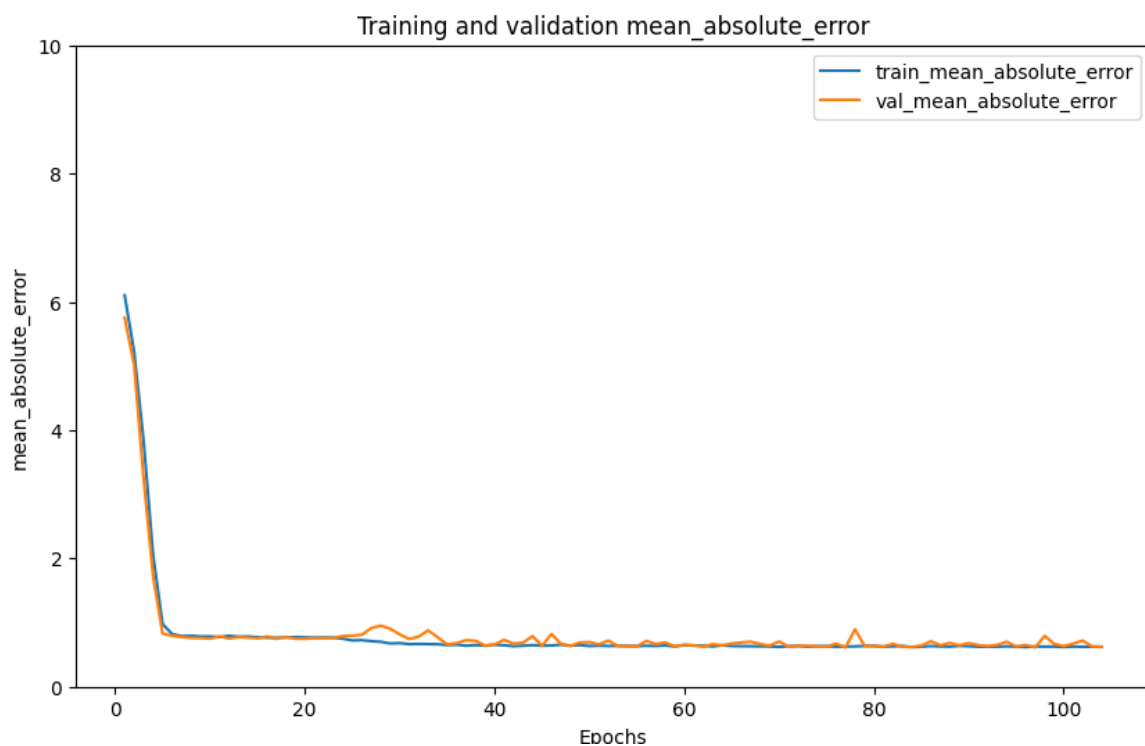


Imagen 3 - Training vs validation mean absolute error

Validation loss: 0.62

Test loss: 0.60

Red Neuronal 3:

Topología:

- Input Layer: 7 neuronas.
- Hidden Layer: 16 neuronas. Activación relu.
- Hidden Layer: 128 neuronas. Activación relu
- Hidden Layer: 64 neuronas. Activación relu
- Hidden Layer: 128 neuronas. Activación tanh.
- Output Layer: 1 neurona.

Técnicas de regularización:

- Batch Normalization.
- Dropout con una probabilidad de 20%.

Función de pérdida: Huber Loss

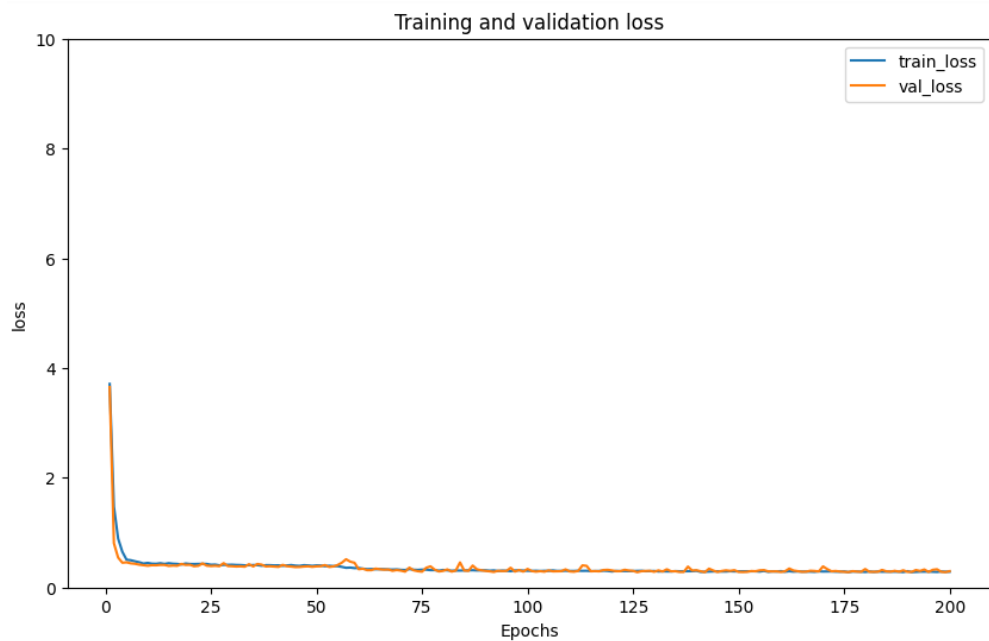


Imagen 4 - Training vs validation huber loss

Validation loss: 0.29

Test loss: 0.27

Red	Validation Loss	Test Loss
1	0.71	0.73
2	0.62	0.60
3	0.29	0.27

Cuadro 1 - Comparando resultados de cara red

Discusión

Las tres redes neuronales demostraron generalizar la información de una manera aceptable. Asimismo, las técnicas de regularización permitieron evitar el *overfitting* y hacer que el modelo fuese entrenado eficientemente. Se comprobó que al emplear técnicas de regularización se puede disminuir el *overfitting* de un modelo y que la combinación de técnicas de regularización junto con la técnica *Batch Normalization* resulta en una menor pérdida. Sin embargo, al usar exclusivamente la técnica de *Batch Normalization* se notó una pérdida considerable (71% - Validation Loss y 73% Test Loss). La técnica de *Batch Normalization* resultó indispensable debido a que trabajamos con valores que varían enormemente. Por ejemplo, las cifras de dinero alcanzan las millones de unidades, mientras que los índices de aprobación únicamente están entre cero y diez. En cuanto al tiempo de ejecución, la segunda red (que implementa *EarlyStopping*) demostró superar a las demás, dando un tiempo de entrenamiento menor a los 5 segundos mientras que las redes uno y tres tenían un tiempo mayor a los 13 segundos. Sin embargo, al agregar la posibilidad de las neuronas de “apagarse” de manera aleatoria, la última red neuronal produjo una pérdida mucho menor a las anteriores (Tabla 1).

Red neuronal óptima

La red neuronal óptima es la número tres, demostrando tener una menor pérdida para el set de validación y el set de prueba. El uso de *Batch Normalization* y *Dropout* fueron la mejor combinación. El tiempo de ejecución podría reducirse utilizando *Early Stopping*, pero la librería *pytorch* no permite utilizar esta funcionalidad en conjunto con la función de pérdida escogida (Huber Loss).