TensorBoard Observations

Both the model architectures incorporate two different callbacks ***ModelCheckpoint*** and ***EarlyStopping***.

ModelCheckpoint is used to save the best model during training at the end of each epoch where validation loss is minimum.

EarlyStopping callback restores the best weights only. Using this callback prevents overfitting.

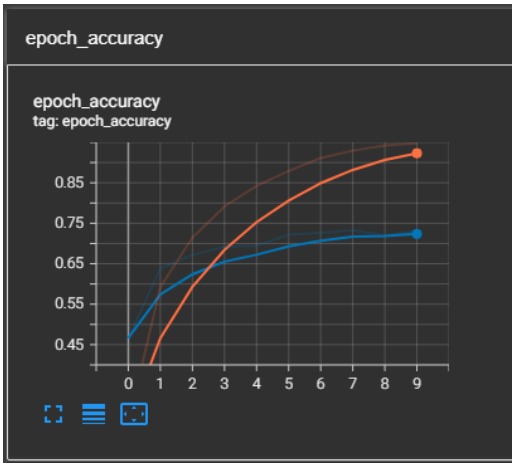
Model-1

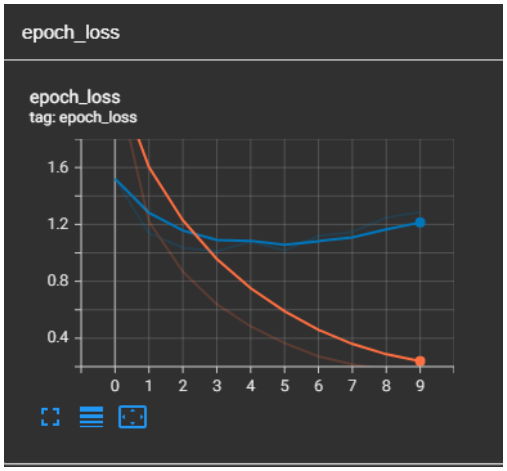
My model is set to train for 50 epochs, using early stopping with the patience of 2, and the best weights occur in epoch 10, then the best weights are restored from epoch 10.

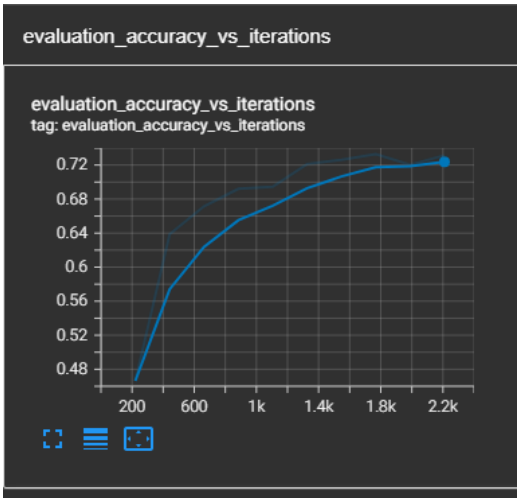
When we evaluate the model on the test data, the accuracy is 73%.

Note: Red is the train curve and blue is the validation curve.

1. The train epoch accuracy is 0.9478 and the validation epoch accuracy is 0.7315.
2. The train epoch loss is 0.1672 and the validation epoch loss is 1.286.
3. The final evaluation accuracy increases over iterations which is 0.7315.





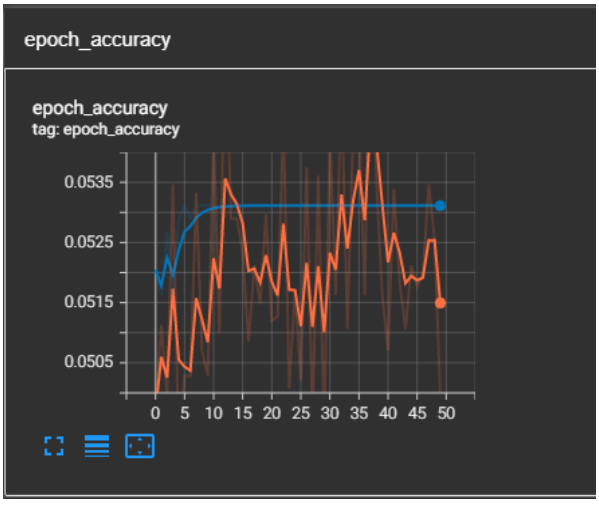


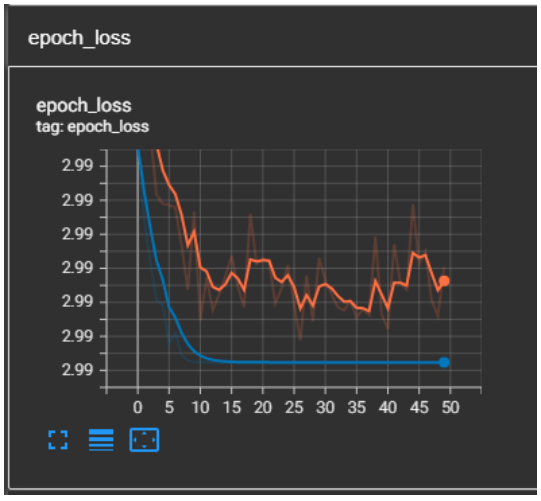
Model-2

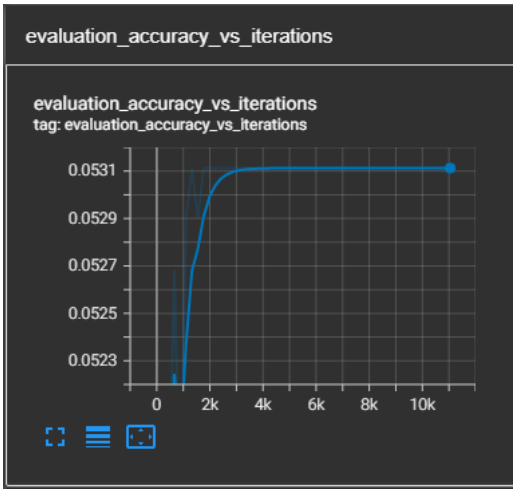
Reference: https://keras.io/api/callbacks/reduce_lr_on_plateau/
This model uses an additional callback which is ***ReduceLROnPlateau*** which reduces the Learning rate when a metric has stopped improving. I did not incorporate the EarlyStopping callback to increase the accuracy of the model and trained the model for 50 epochs.
When we evaluate the model on the test data, the accuracy is 5.31%.

Note: Red is the train curve and blue is the validation curve.

1. The train epoch accuracy is quite erratic throughout the training process which is 0.0499 and the validation epoch accuracy is 0.05311 and it is quite stable.
2. The train epoch loss is decreasing haphazardly and the validation epoch loss is decreasing smoothly over 50 epochs.
3. The evaluation accuracy increases over iterations.






evaluation_accuracy_vs_iterations
evaluation_accuracy_vs_iterations
tag: evaluation_accuracy_vs_iterations
0.0531
0.0529
0.0527
0.0525
0.0523
0
2k
4k
6k
8k
10k