

Sequence-to-sequence autoencoder for classifier

Dataset: IMDB tensor flow dataset- It contains text and labels indicating sentiments(1 or 0)

Model Details:

The text data needs to be embedded into vectors. Google embedding “nnlm-en-dim50” this takes the text as input and embed it into a vector of 50 dimensions.

The output of the embedding is fed into a Convolution 1D layer followed by a MaxPool1d Layer and then to a Dense layer. These three form the encoder.

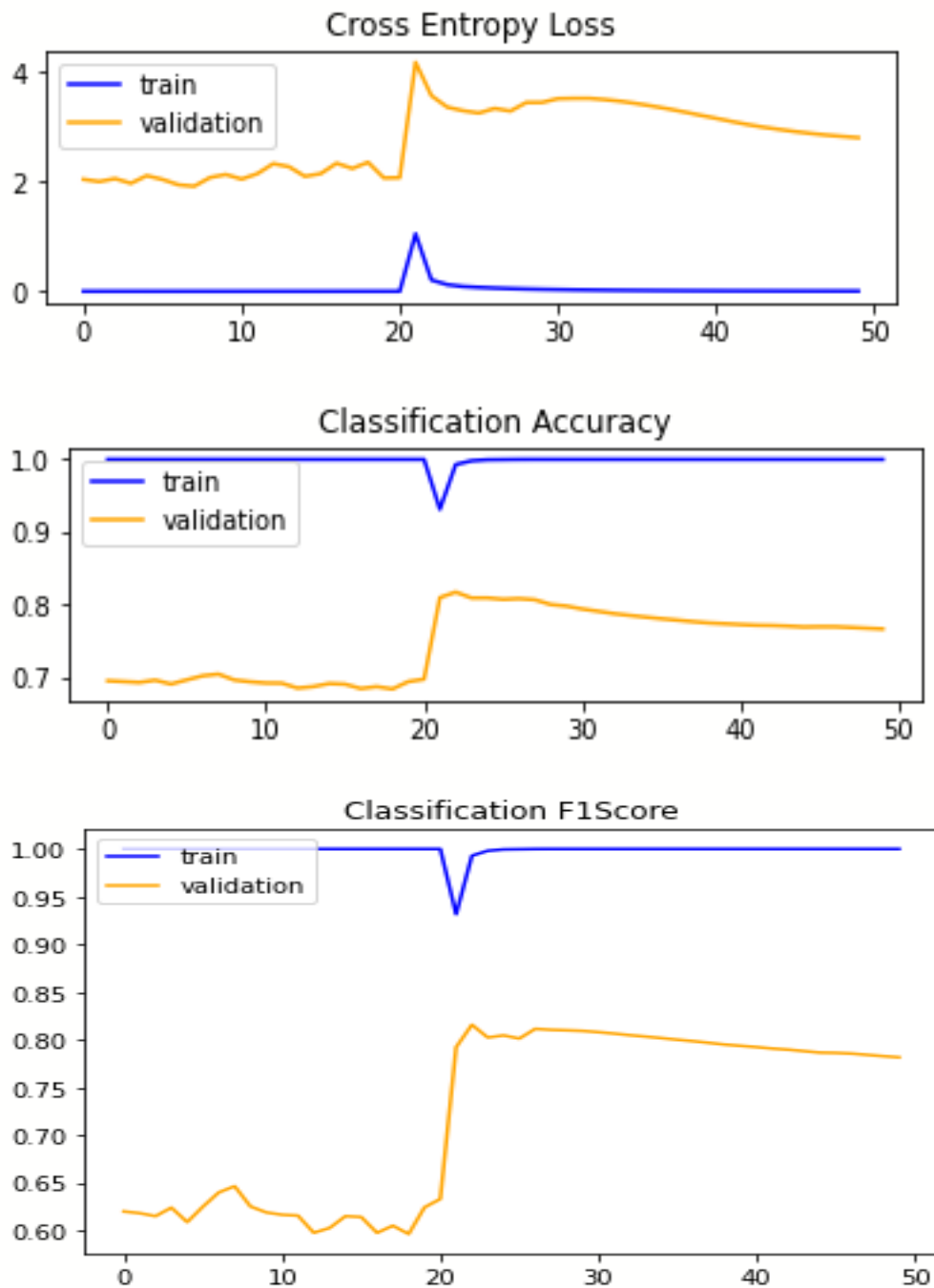
Output of the encoding is passed by convolution layer and then a max pooling layer. It's then flattened and applied to 2 dense layers.

Optimizer: ADAM

Loss function: Binary Cross Entropy

Layer (type)	Output Shape	Param #
keras_layer (KerasLayer)	(None, 50)	48190600
reshape (Reshape)	(None, 50, 1)	0
encoder_conv11 (Conv1D)	(None, 50, 32)	160
encoder_pool1 (MaxPooling1D)	(None, 50, 32)	0
dense (Dense)	(None, 50, 50)	1650
fc-conv1 (Conv1D)	(None, 47, 16)	3216
fc-pool1 (MaxPooling1D)	(None, 11, 16)	0
flatten (Flatten)	(None, 176)	0
dense_1 (Dense)	(None, 16)	2832
output (Dense)	(None, 1)	17
Total params: 48,198,475		
Trainable params: 48,198,475		
Non-trainable params: 0		

Result:



The loss function, accuracy and F1 Score follows a horizontal path till epoch 20 and then jerks a bit and then follows the same for training set but improves for test set.

The observation is that the model is over fitting the training data set.

