Results are discussed in this file.

We started the exercise by learning how tokenization of text is done. We did this by converting words of sentence sequence to tokens by using the Tokenizer object.

We also learned how to convert sentences into data, by forming padded arrays for all sentence sequence.

After learning, basic pre-processing of text, we applied the same techniques on a dataset of news headlines and try to classify them as sarcastic or not sarcastic.

We divided the dataset into two parts, training dataset and test dataset.

The classifier uses the word-vector model and the classifier is trained using the training dataset over 30 epochs.

The fully trained model is then fed with the test dataset.

The accuracy of the classifier over 30 epochs on test data is 0.8106 which is approximately 81%.

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

We see that val\_loss(test)  starts increasing, val\_acc(test) starts decreasing. This means model is cramming values not learning.

There are following situations that can be encountered given a classifier:

1. val\_loss starts increasing, val\_acc starts decreasing. This means model is cramming values not learning
2. val\_loss starts increasing, val\_acc also increases. This could be case of overfitting or diverse probability values in cases where softmax is being used in output layer
3. val\_loss starts decreasing, val\_acc starts increasing. This is also fine as that means model built is learning and working fine.

Here, we encounter the first scenario.