1. BERT is an unsupervised deep network for representing text that learns text representations from unstructured text and because it is trained with unlabeled data, it uses unsupervised deep learning representation. BERT learns how representation words behave in different contexts. As a result, BERT can learn richer semantic representations that receive different meanings of words depending on their context. In fact, BERT is optimized with a special type of supervised -self task that does not require manually annotated data.

BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The paper's results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. In the paper, the researchers detail a novel technique named Masked LM (MLM) which allows bidirectional training in models in which it was previously impossible.

The encoder maps the input sentence to a series of embedding vectors (one for each word in the sentence). These vectors pass through a SoftMax layer that calculates the probability of the entire vocabulary so that the most likely words have a greater chance of being selected. In other words, it is a sentence-filling task where BET aims to reconstruct the corrupted input signal from the partially existing context.

A set of trained transformer model encoders is the basis for the BERT model and a large number of encoder layers are included in both BERT models. BERT BASE model consists of 12 encoder layers (Transformer Block) and an expanded model called the BERTLARGE model. The BERTBASE model is composed of 12 encoder layers (Transformer Blocks). It consists of 24 encoder layers; The basic model has 110 million parameters and the large model has 345 million parameters. The number of layers is 12 in the former and 16 in the latter.

The first input token enters the model with a special CLS, and just like the encoder part of the transformer model discussed in the previous section, the BERT model receives a sequence of words as input and moves along the existing encoder layers. This is similar to the encoder part of the transformer model discussed in the previous section.

A single encoder layer is a self-contained one, an attention-self layer and a feed-forward network layer that provides inputs to pass through the next encoder level and then enter the next encoder layer. For example, in this model, the divine hidden size is 768. In other words, in the BERTABASE model, it is 768.

3.A. our test accuracy is 0.099 and it demonstrate that based on don't have enough labels, our model overfit on train data.

C. We tested on the following weights and achieved classification accuracy = 10% and rotation accuracy = 25% in all of them. This shows that our models have not learned anything at all, and one of the reasons that could cause this is that we have used the MobileNet model, which is a robust model, and based on not having enough labels to classify, our model is not able to do it for learning.

(classification, rotation) = (1,1), (1,5), (5,1), (1,50), (50,1), (1,200), (200,1)

4.A.

```
conv_blocks: 3
Dropout rate in conv: 0.3
dense size: 128
Dropout rate in dense: 0.3
lr: 0.0007450811834797452
```

B. As can be seen, the suitable hyperparameters of 3 black convolutions are Dropout probability with a value of 0.3, the number of penultimate Dense layer neurons of 128, and learning rate with a value of 0.0007. More convolutional blocks make the model learn more meaningful features, less dropout reduces the probability of disabling neurons, and learning increases with more neurons, the number of dense layer neurons is increased enough. To increase the learning, but not to the extent that the model overfits, the learning rate is also reduced to reduce the exploring model.

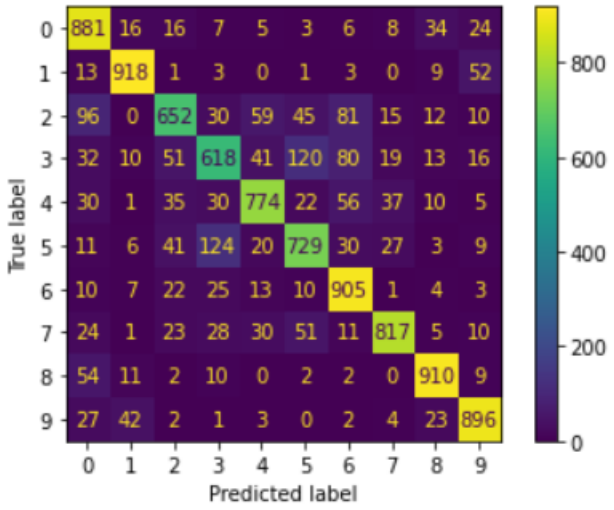
C. accuracy: 0.8003 - val_loss: 0.5663 - val_accuracy: 0.8100

F1 Score: 0.81 Precision Score: 0.81 Recall Score: 0.81

According to the results of the metrics, we can see that the performance of the model was appropriate. The main reason for using these metrics is to make sure that the model correctly identifies duplicates and avoids false positives.

F1 balances precision and recall of the positive class, while accuracy looks at correctly classified positive and negative observations. So, in this dataset, we don't need these metrics and accuracy is enough to check the performance of the model.

D. Our matrix shows that in class 2 and 3 we have lowest accuracy compared to other class in predicting correct label.



True positive = 737 (Our model predict correct and real label was +)

False positive = 16 (Our model predict negative and real label was +)

False negative = 4 (Our model predict negative and real label was -)

True negative = 875 (Our model predict correct and real label was -)

REF:

https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270