Done so far :

Baseline Model \*

Replicating Baseline Model \*

Google cloud \*

My instance \*

The simplest way to prevent overfitting is to reduce the size of the model: the number of learnable parameters in the mode

To recap, these are the most common ways to prevent overfitting in neural networks:

ϒ Get more training data.  
ϒ Reduce the capacity of the network. ϒ Add weight regularization.  
ϒ Add dropout.

Always monitor the training loss and validation loss, as well as the training and valida- tion values for any metrics you care about. When you see that the model’s performance on the validation data begins to degrade, you’ve achieved overfitting. The next stage is to start regularizing and tuning the model, to get as close as pos- sible to the ideal model that neither underfits nor overfits.

Typical features of the learning curve of a good fit model

1. Training loss and Validation loss are close to each other with validation loss being slightly greater than the training loss.
2. Initially decreasing training and validation loss and a pretty flat training and validation loss after some point till the end.

An underfit model is one that is demonstrated to perform well on the training dataset and poor on the test dataset.

This can be diagnosed from a plot where the training loss is lower than the validation loss, and the validation loss has a trend that suggests further improvements are possible.

 performance may be improved by increasing the number of training epochs.

Alternately, a model may be underfit if performance on the training set is better than the validation set and performance has leveled off.

GOOD fit = This can be diagnosed from a plot where the train and validation loss decrease and stabilize around the same point.

OVERFitting = An overfit model is one where performance on the train set is good and continues to improve, whereas performance on the validation set improves to a point and then begins to degrade.

## Overfitting

In general, if you're seeing much higher validation loss than training loss, then it's a sign that your model is overfitting - it learns "superstitions" i.e. patterns that accidentally happened to be true in your training data but don't have a basis in reality, and thus aren't true in your validation data.

our model is completely overfitting. The training loss is constantly decreasing but the validation loss isn't. This means that the your current model is complex enough to 'memorize' the patterns in the training data. In such situations, you need to regularize your model.

95

Are you using dropout in your model? If so, that could explain the difference since dropout is enabled during training (leading to higher losses) whereas it is not enabled during validation/test.

Relu , gradient slope never smaller than one unless 0, so the more gradients we multiply it will not get smaller and smaller

Final dense 128, 0.01 dropout

Chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

Overfitting or something ? 0.05 dropout , 128 dense

Graphical user interface, application

Description automatically generatedChart, line chart, histogram

Description automatically generated