The results of the research on the best network type

2.1:

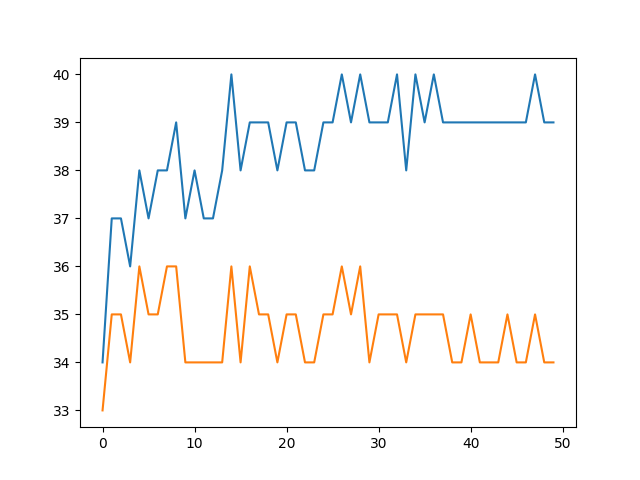


Figure 1: Lazynet with logistic activation accuracies plot

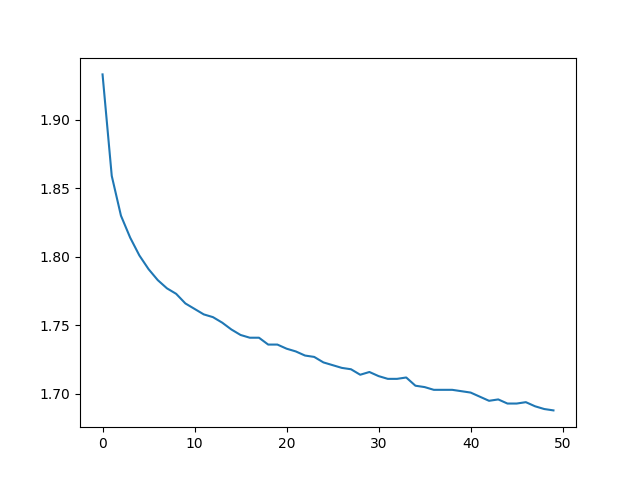


Figure 2: Lazynet with logistic activation loss function plot

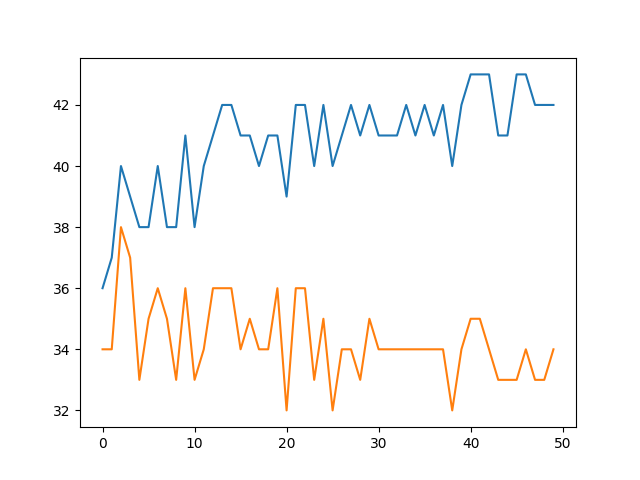


Figure 3: Lazynet with RELU activation accuracues plot

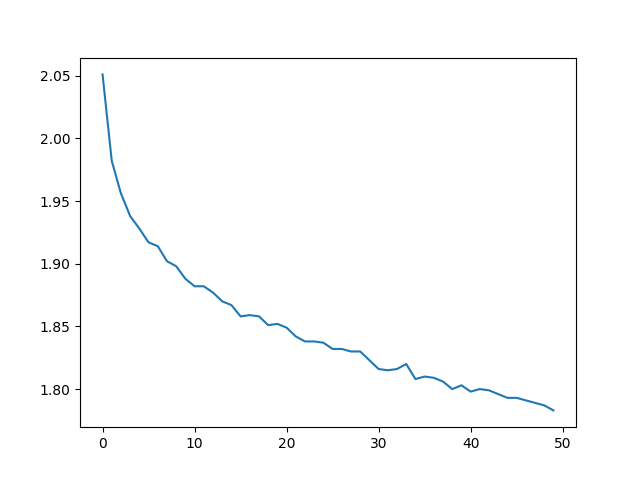


Figure 4: Lazynet with RELU activation loss function plot

Both functions act similar to each other, however the logistic activation acts a little better. The accuracy plots are spiky which means that the learning rate may be too big for them.

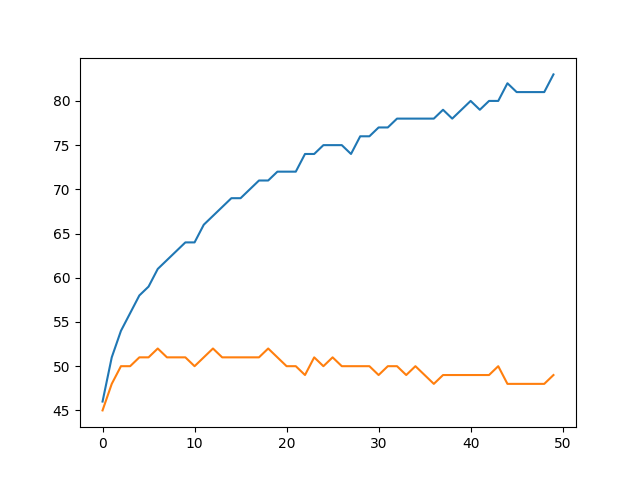
2.2:

Figure 5: Boringnet with linear activation (last layer) accuracies plot

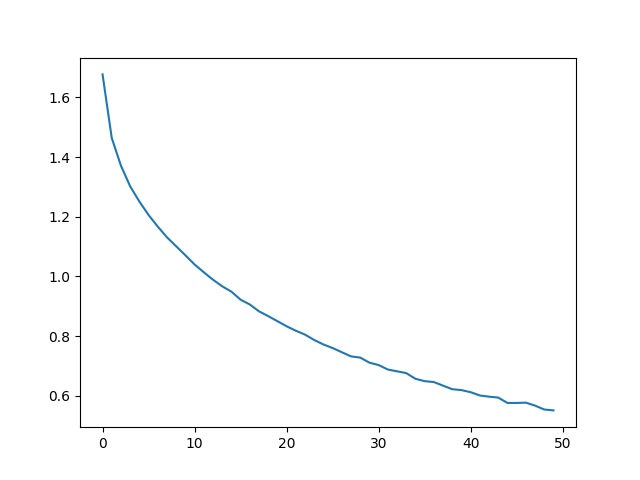


Figure 6: Boringnet with linear activation (last layer) loss function plot

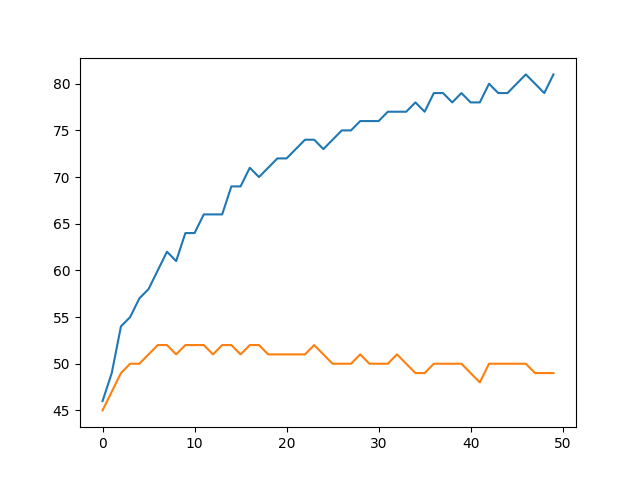


Figure 7: Boringnet with logistic activation (last layer) accuracies plot

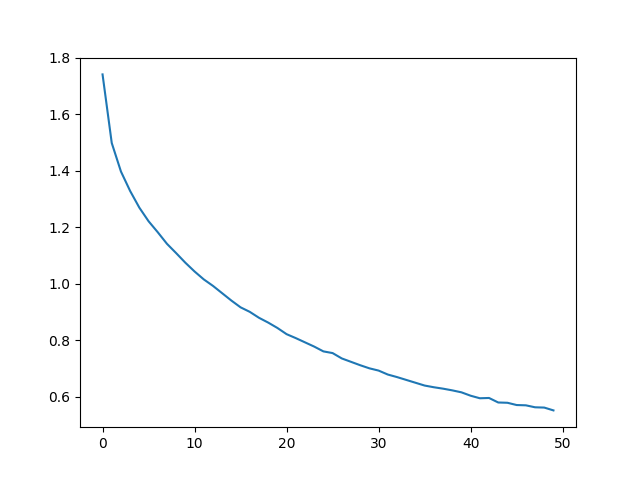


Figure 8: Boringnet with logistic activation (last layer) loss function plot

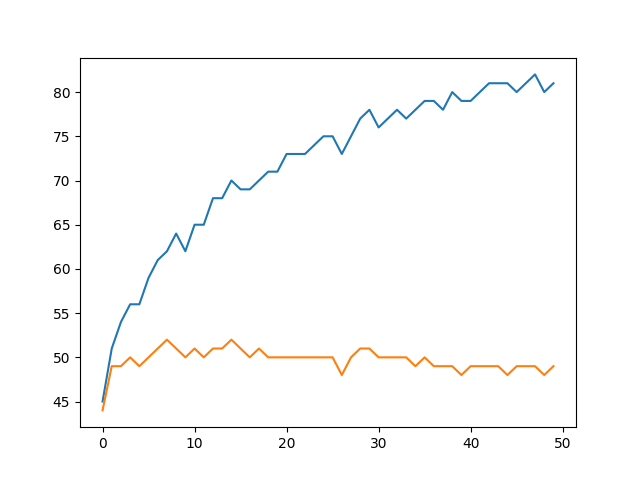


Figure 9: Boringnet with RELU activation (last layer) accuracies plot

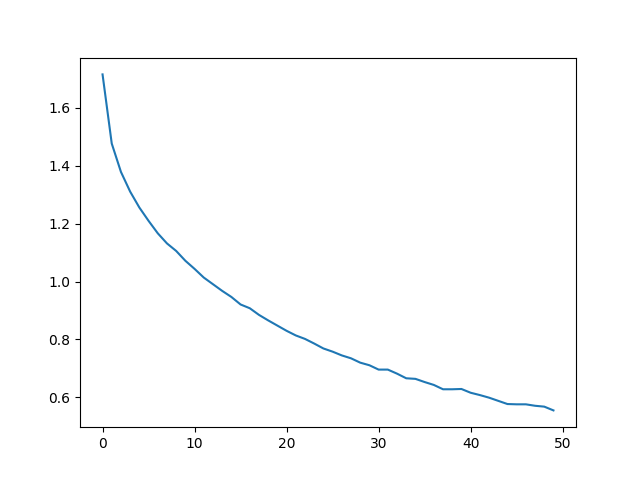


Figure 10: Boringnet with RELU activation (last layer) loss function plot

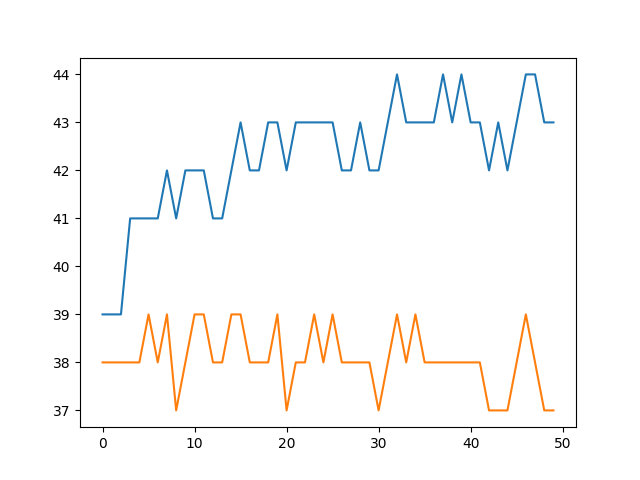


Figure 11: Boringnet with no activation (all layers) accuracies plot

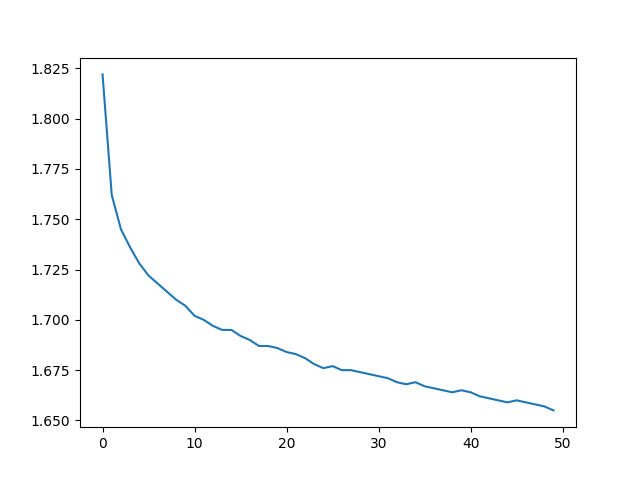


Figure 12: Boringnet with no activation (all layers) loss function plot

The no-activation nn model is acting quite bad on the dataset, probably because there is no linearization on the nn weights. All other models act quite the same.

2.3:

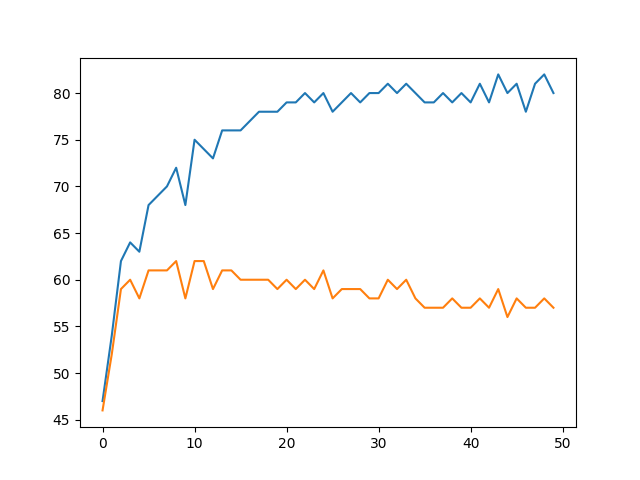


Figure 13: Coolnet, stock, accruacies plot

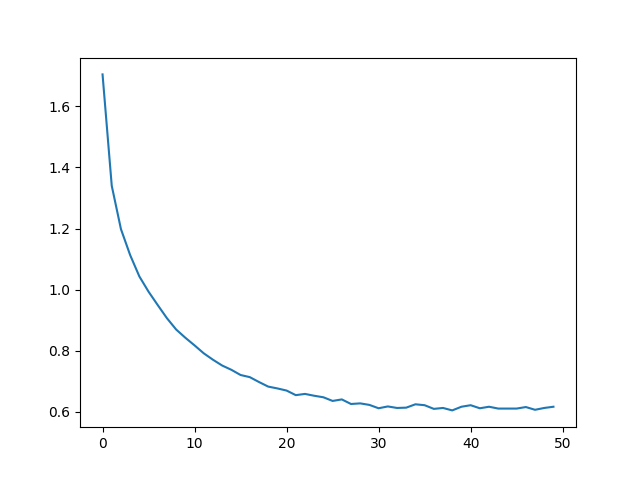


Figure 14: Coolnet, stock, loss function plot

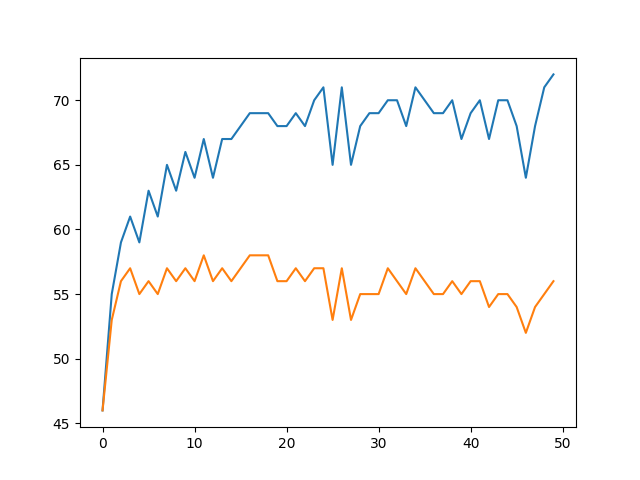


Figure 15: Coolnet, batchsize=2, accuracies plot

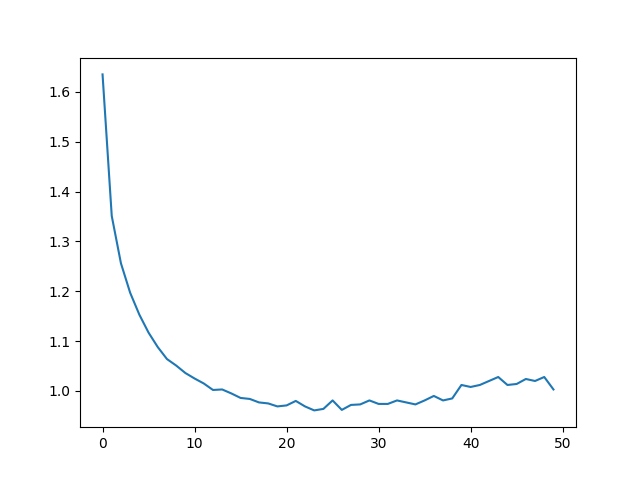


Figure 16: Coolnet, batchsize=2, loss function plot

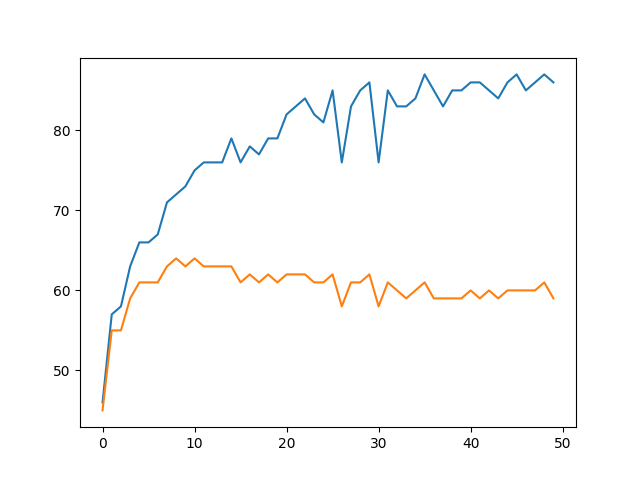


Figure 17: Coolnet, batchsize=6, accuracies plot

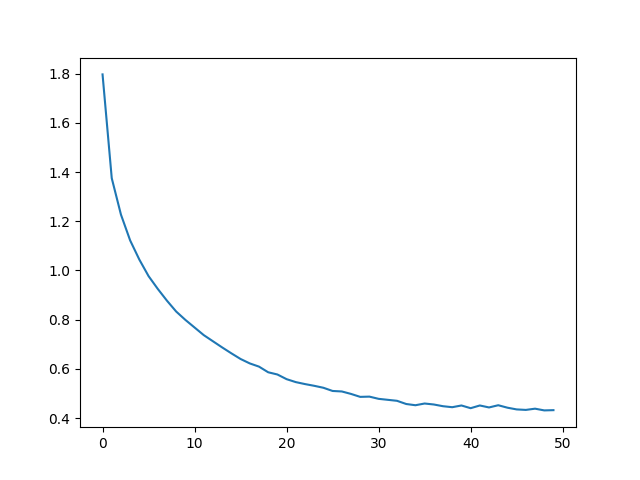


Figure 18: Coolnet, batchsize=6, loss function plot

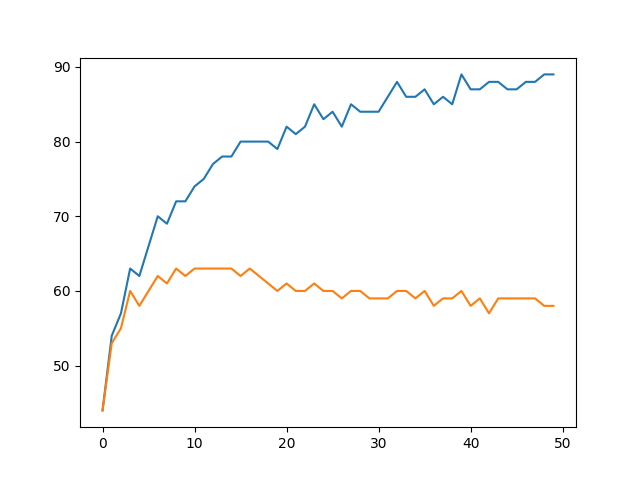


Figure 19: Coolnet, batchsize=8, accuracies plot

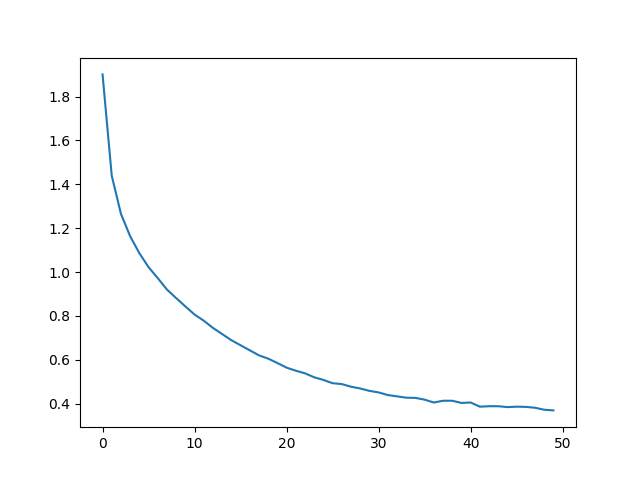


Figure 20: Coolnet, batchsize=8, loss function plot

The batchsize for the stock coolnet model is 4. Thus, increasing the batchsize leads to increasing smoothiness of the plots and generally improves the accuracy of the model.

3.0

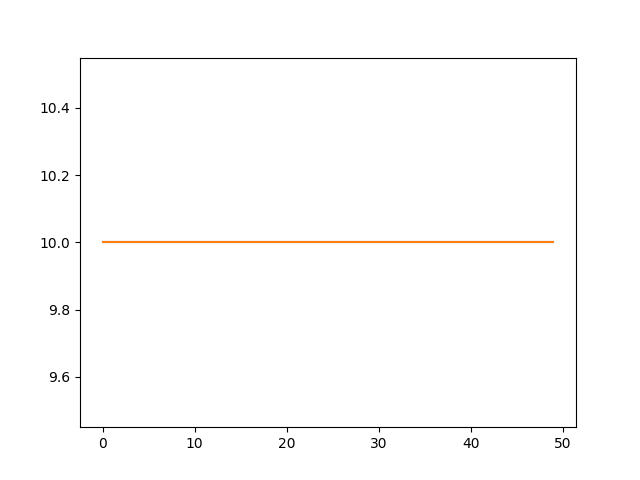


Figure 21: Coolnet, lr=10, accuracies plot

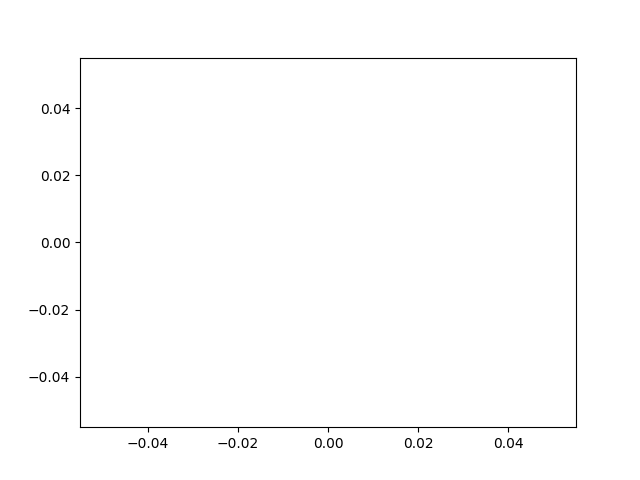


Figure 22: Coolnet, lr=10, loss function plot

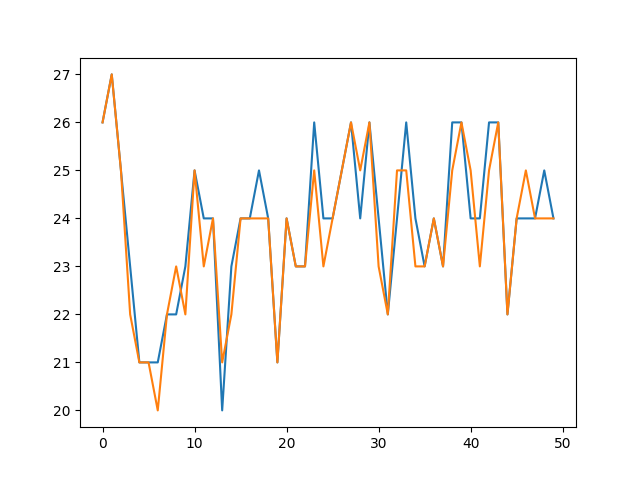


Figure 23: Coolnet, lr=0.1, accuracies plot

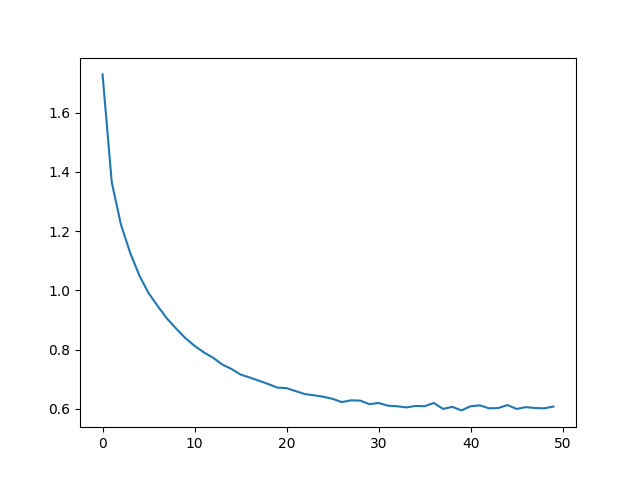


Figure 24: Coolnet, lr=0.1, loss function plot

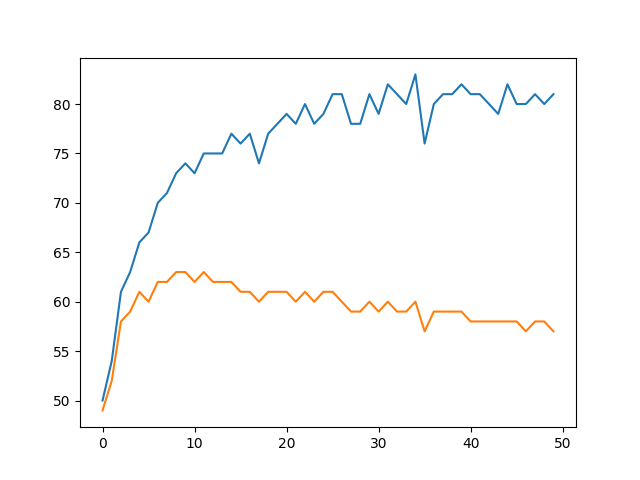


Figure 25: Coolnet, lr=0.01, accuracies plot

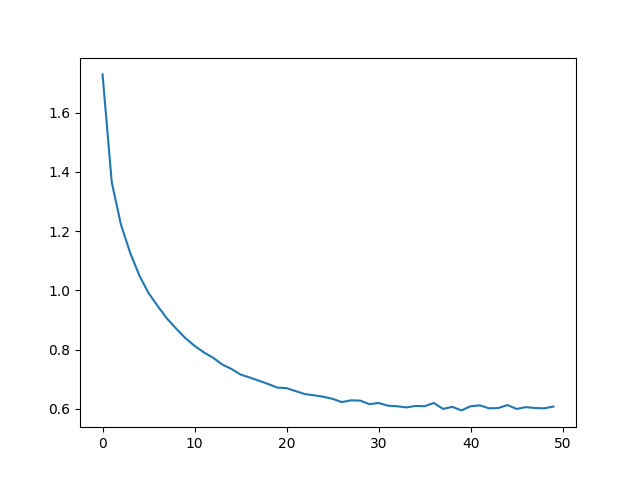


Figure 26: Coolnet, lr=0.01, loss plot

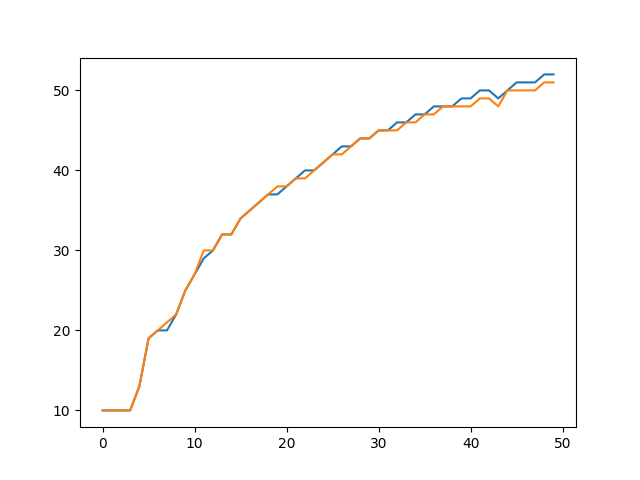


Figure 27: Coolnet, lr=0.0001, accuracies plot

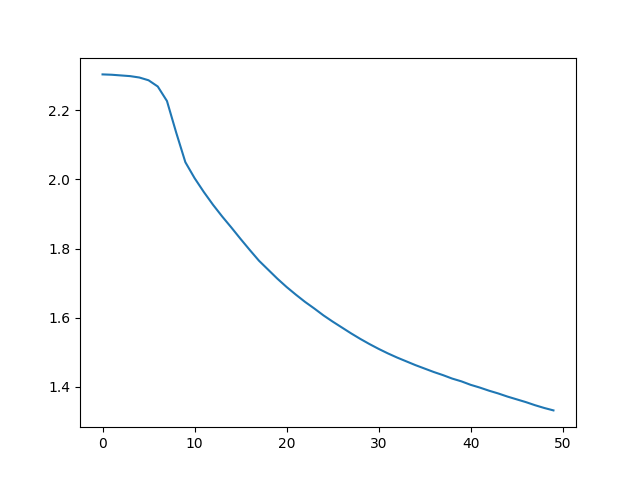


Figure 28: Coolnet, lr=0.0001, loss function plot

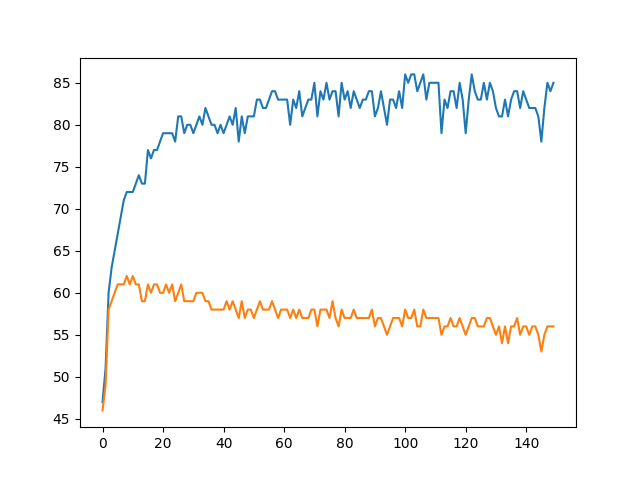


Figure 29: Coolnet, decreasing lr, accuracies plot

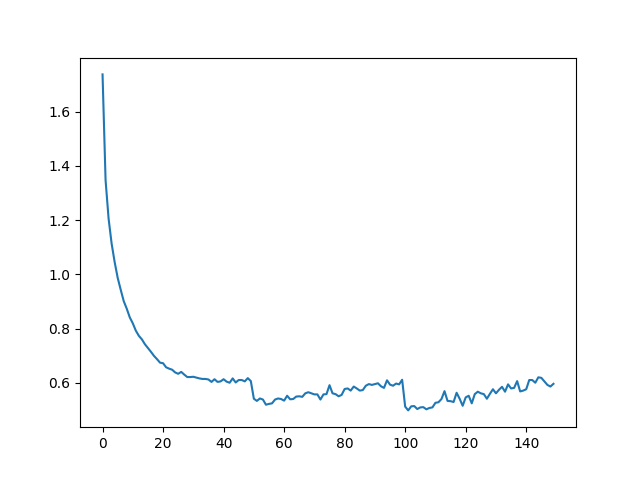


Figure 30: Coolnet, decreasing lr, loss function plot

Here lr=10 does not learn at all, lr=0.1 is too spiky, decreasing lr up to 0.01 improves the plots. The model with decreasing lr did not give much better results.

4.0

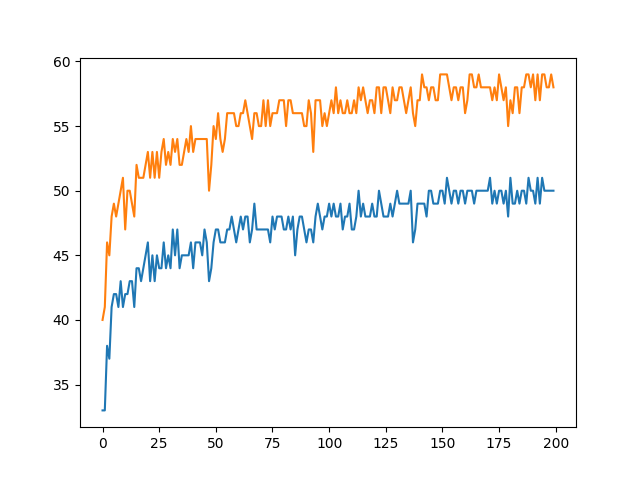


Figure 31: Coolnet, data augmentations, accuracies plot

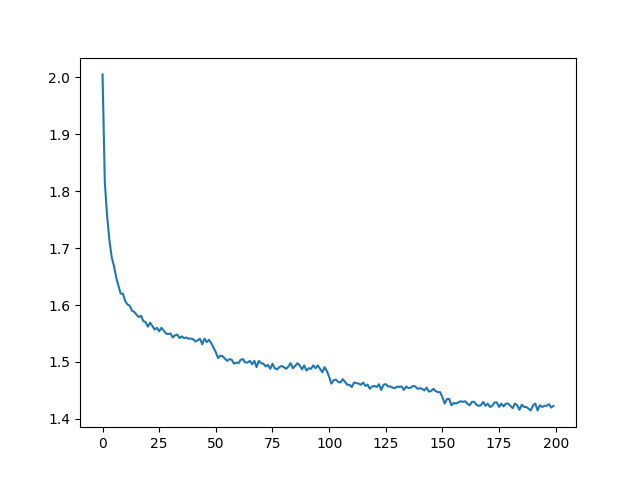


Figure 32: Coolnet, data augmentations, loss function plot

The data augmentations help with overfitting, but the general learning rate decreases.

5.0:

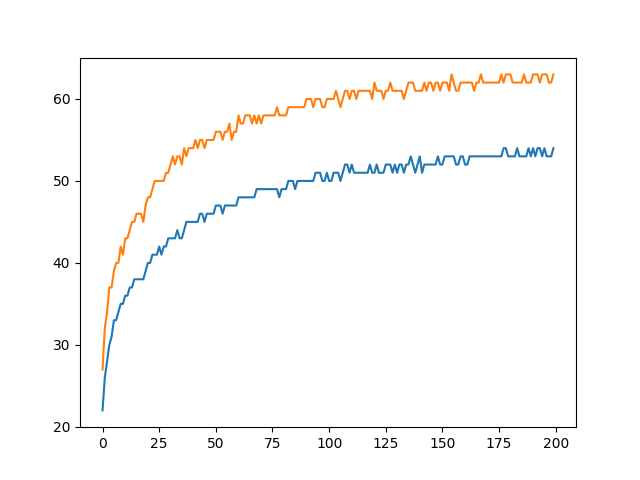


Figure 33: Coolnet, MSE, accuracies plot

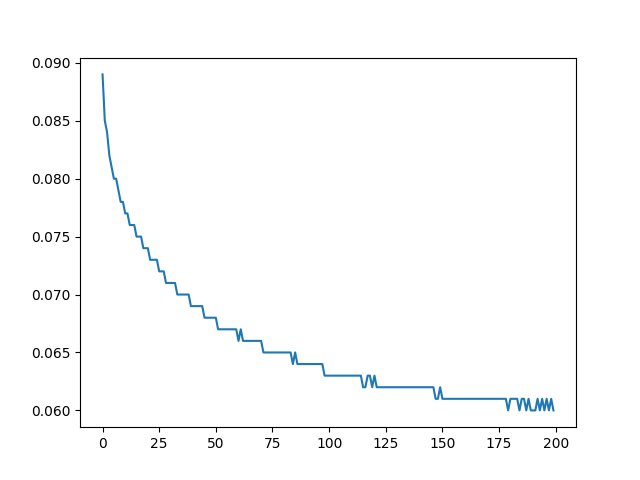


Figure 34: Coolnet, MSE, loss function plot

The MSE loss seems to learn quite slower (the plots show that 200 epochs are quite enough for it) whereas the cross-enthropy gets similar results much faster.