Foundations of Deep Learning – HW 1

Part 1

SVM with linear kernel achieved train accuracy of 0.9894 and test accuracy of 0.323.

SVM with RBF kernel achieved train accuracy of 0.7134 and test accuracy of 0.433.

Part 2

1. Values over which the grid search was preformed:

Momentum – [0.6, 0.7, 0.8, 0.9], learning rate – [0.002, 0.003, 0.004, 0.005], std – [0.05, 0.1, 0.5].

The best hyperparameters found were momentum=0.9, learning rate=0.005, std=0.05, and achieved test accuracy of 0.431 and train accuracy of 0.4531.

1. The model converged a lot quicker with the ADAM optimizer, but achieved worse results – test accuracy at ~0.3, train accuracy at a maximum close to 0.5 compared to ~0.4 and maximum of 0.8 with SGD, respectively. The train and test losses also converged quicker, but to higher values – above 1.6 and 2.0, compared to 0.8 and under 2.0 after 50 epochs with SGD. The difference in results is probably due to the fact that the grid search over hyper-parameters was done with SGD optimization. If we repeated the grid search with ADAM optimization, we would probably find hyper-parameters more suited for the different optimization and achieve higher results compared to SGD.

Results of SGD:

Chart

Description automatically generated Chart, line chart

Description automatically generated

Results of ADAM:

A picture containing chart

Description automatically generated Chart

Description automatically generated

When we ran optimization with ADAM’s default learning rate (0.001) the model performed much better, getting results similar to those achieved with SGD:

A picture containing graphical user interface

Description automatically generated Chart, line chart

Description automatically generated

1. The Xavier initialization didn’t have a strong effect on either convergence time or results. The test accuracy was slightly higher, but the difference does not seem significant.

A picture containing chart

Description automatically generated Chart, line chart

Description automatically generated

1. Weight decay with a very small value (0.001) did not seem to have much effect on convergence or performance. Decay value of 0.01 seemed to slightly improve test accuracy and lower train accuracy, but also didn’t affect convergence time. Decay value of 0.1 seemed to really hurt the model – train loss converged very quickly, but to a value higher than test accuracy and both accuracies fell considerably.

Chart

Description automatically generated Chart, line chart

Description automatically generated

A picture containing text

Description automatically generated A picture containing chart

Description automatically generated

A picture containing shape

Description automatically generatedA picture containing company name

Description automatically generated

Dropout: the higher the value tested, test prformance was better (although not by a very significant margin) and train performance was worse. This makes sense as dropout makes infering in train time hard (changing some weights to 0) and does not affect test inference directly.

Chart

Description automatically generated with low confidence Chart, line chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated Chart, line chart

Description automatically generated

A picture containing text

Description automatically generated Chart, line chart

Description automatically generated

A picture containing text

Description automatically generated Chart

Description automatically generated

Chart

Description automatically generated Chart

Description automatically generated

1. PCA did not have a strong effect on convergence, but improved train performance by a significant margin – nearing 0 loss and 1.0 accuracy. Test prformance was slightly better at the first ~20 epochs, but got worse over time. This probably indicates the model overfitted the data, which was easier to do because of the reduced dimension. It is also worth mentioning the losses as a function of epoch were smoother.

Chart

Description automatically generated with medium confidence Chart, line chart

Description automatically generated

1. Widening the network mainly improved train perforamnce, with the last network reaching an almost perfect fit of the train data. convergence and test perforamnce were not gratly affected.

A picture containing chart

Description automatically generated Chart, line chart

Description automatically generated

Chart

Description automatically generated with medium confidence Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

1. Icreasing model depth improved test perforamnce slightly, but hurt train performance and did not affect convergence. The deepest model achieved very bad results, comparable to random guess. This may be due to vanishing or exploding gradients.

Chart

Description automatically generated Chart

Description automatically generated

A picture containing text

Description automatically generated Chart, line chart

Description automatically generated

Histogram

Description automatically generated A picture containing diagram

Description automatically generated