BUAN 6382 Applied Deep Learning HW5 Report

Megan Yiu

mny170000

Q2 - Experiments

The first few experiments focused on trying to overfit a small dataset of the training dataset. All experiments used 10 epochs. Afterwards, the next experiments utilized all the training data and configured different optimizers and

**Experiment 1 - Overfitting small dataset (change architecture)**

Kept the same hyperparameters as in question 1. The only changes made to this model run is the architecture.

The architecture for this run was: Conv -> Relu -> BatchNorm -> Conv -> Relu -> BatchNorm

Test accuracy for the small dataset was 74%, so this setup does not overfit the training data quite yet, but there is improvement from the model in question 1.

**Experiment 2 - Overfitting small dataset (increase filters on last conv layer)**

Next, I tried increasing the number of filters on the last convolutional layer to 64 instead of 16. This change seemed to improve the fitting of the data, increasing accuracy to 80%. However, it is still not fully overfitting to the small subset.

**Experiment 3 - Overfitting small dataset (decrease filter size on first conv layer)**

Next, I tried decreasing the size of the filter on the first convolutional layer from 5x5 to 3x3 and decreasing the padding size to 1 to match the second layer. This change led to an increase to 90% accuracy with the training data.

**Experiment 4 - Overfitting small dataset (change architecture)**

This run, Max Pooling was added to the network architecture. With max pooling, had to add a stride of 1 to the convolution network.

This change did not help the accuracy of the small dataset. It decreased by 8% to 82%.

**Experiment 5 - Overfitting small dataset (learning rate)**

Tried decreasing the learning rate to 1e-3 instead of 1e-2 to see if this helps improve.

It did not help to improve the model. Accuracy is 74%.

**Experiment 6 - Overfitting small dataset (change architecture)**

Changed architecture back without max pooling or strides. This time, accuracy is even worse at 70%. Will change learning rate to a higher number next run.

**Experiment 7 - Overfitting small dataset (learning rate)**

Tried changing learning rate to 0.1, but this change worsened the model to an accuracy of 14% on the training data.

**Experiment 8 - Overfitting small dataset (change architecture and learning rate)**

Reverted back to the old learning rate of 1e-2. For the architecture, the max pooling and strides were added back in for the first block. Additionally, a second block of Conv2d with 128 outputs -> ReLU -> Conv2d with 128 outputs -> ReLU -> BatchNorm -> MaxPool was added.

Accuracy was 78%.

**Notes: After these experiments, kept the architecture and configuration from Experiment 3, which had the best training results to test and improve on the full dataset.**

**Experiment 9 - Full training**

Initial run on the full training dataset with the configuration of Experiment 3.

This run resulted in a validation accuracy of 52.96%.

**Experiment 10 - Full Training (change optimizer)**

Using the same basic configurations as experiment 9 except the optimizer is changed from SGD to Adam.

This resulted in a validation accuracy of 9.73%.

**Experiment 11 - Full training (change optimizer)**

This time, changed optimizer to adagrad. Resulted in 10.73% validation accuracy, which is slightly better than Adam but not better than SGD.

**Experiment 12 - Full training (change optimizer)**

Changed optimizer to RMSprop. Resulted in 9.73% training accuracy.

**Experiment 13 - Full training (change architecture)**

Tried adding more fully connected dense layers into the mix as well as more convolution layers.

Resulted in ~70% validation accuracy within the 5 epochs that were able to run.

https://wandb.ai/ymegan/DL\_Course\_HW5/runs/23k2dhoh?workspace=user-ymegan