

Lab 5 Exercise - A little Linear Regression

Luke McClure
29573904

1 AN INITIAL ATTEMPT

As this is a regression problem MSE was used as the loss function for all models.

1.1 A simple CNN baseline

When considering the connections between the convolutional and fully connected layers, and therefore the learnable parameters within the fully connected layers (9830786 to be exact), it is no surprise that the model starts to overfit to the data relatively quickly as seen in Figure 1.

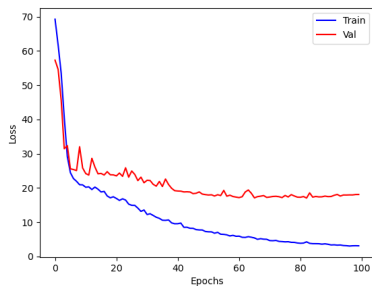


Figure 1: CNN Baseline

The validation and training losses start to diverge at around 25, with the final validation loss at close to 20 while final training loss is closer to 5.

2 AN SECOND ATTEMPT

2.1 A simple CNN with global pooling

By using a global max pooling layer between the convolutional and fully connected layers we can cut down on the number of learnable parameters, while also abstracting out detail from the image. This vastly reduces the problem of the model overfitting to data, as is shown in Figure 2. Both training and validation losses drop to around 12 with the global max pooling, with the two staying fairly coupled even through bumps of the validation loss.

3 SOMETHING THAT ACTUALLY WORKS?

3.1 Let's regress

This model performs much better than previous, avoiding the overfitting issues and with losses dropping to close to 5 as seen in Figure 3.

The third CNN is an architecture known as CoordConv, the other input channels are designed to let the network know where it is in Cartesian space. By providing a grid of the x coordinates in one

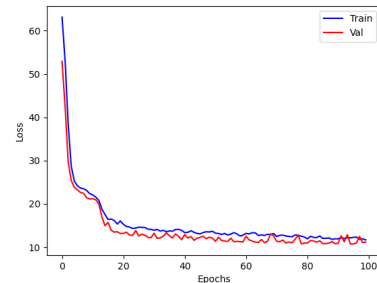


Figure 2: CNN with global pooling

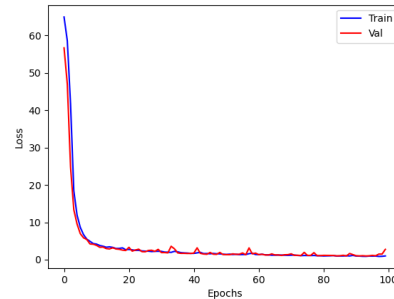


Figure 3: CoordConv

input channel and a grid of the y coordinates in the other hardcoded input channel the filter is able to pinpoint exactly where the points of the line are.

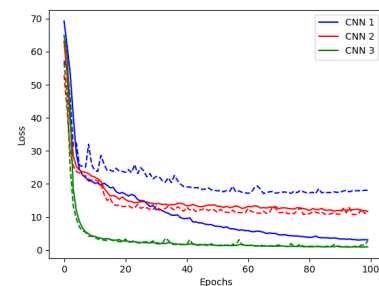


Figure 4: All CNN's compared (- train, - validation)

CNN	Test Loss
CNN Baseline	19.5
CNN with global pooling	14.1
CoordConv	3.21