Lab 5-A little Linear Regression

By: Abhash Shrestha (35464984) as14n23@soton.ac.uk

1.1 A simple CNN baseline

As Instructed, I have used the CNN as:
Convolution2D , channels=48, size=3x3, stride=1, padding=1 ReLU
Linear , 128 outputs
ReLU
Linear , 2 outputs

This was implemented with the code snippet:

```
class CNNModel(nn.Module):

    def __init__(self):
        super(CNNModel, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(1, 48, kernel_size=3, stride=1, padding=1),
            nn.ReLU()
        )
        self.fc = nn.Sequential(
            nn.Linear(48 * 40 * 40, 128),
            nn.ReLU(),
            nn.Linear(128, 2)
        )
```

1st epoch, training loss = 63.243856048583986
10th epoch, training loss = 14.679002332687379
100th epoch, training loss = 0.1304
There is continuous and rapid reduction in the training loss and the model converges. MSE loss was used.
Test Loss = 11.372702598571777
Validation Loss = 8.4850
This is a simple CNN model which is not very deep and fails to learn the relationships efficiently.
The high values of Validation loss indicate that the model has an overfitting problem.

2.1 A simple CNN with global pooling

For this CNN the last training loss (Epoch 100/100)was 15.196919416427612, compared to the previous CNN this is higher, it got worse. The Validation loss was 16.716641187667847 which is higher than the first model. Overall, due to its lower complexity the second model performed worse than the first one.

3.1 Modified model using Adam

Training Loss : 1.0438045848727226
Test Loss : 1.356826052069664
Validation Loss : 1.0082564296722412

In comparison to the first and second one the model converges much faster as there is a more rapid reduction in the training loss. The curve is also much smoother.

It's Test and Validation loss is lower than both the the 1st Model and the 2nd model. And it's training loss is lower than the second model, but only a little greater than the1st model. This leads to the conclusion that it is the best generalizer among the 3 models.

With 3 channels the model can accept input images with multiple channels. This modification can help accommodate RBG inputs which allows the model to utilize color information present in the image and increase its ability to capture the relevant patterns and features. MSE loss was used.

Adding the grid of indices(idxx,idxy,idx) can provide the model with additional spatial information or context. This can help it understand the structure of the input better. In this context, for predicting the parameters of best fit line for a scatter plot, having info about the spatial orientation of the points can help the model learn meaningful patterns and relations, this ultimately improves the predictive ability as shown by loss.
Adam is a fast converging algorithm suitable for deep learning and it efficient as it adapts the learning rate for each parameter based on the first and second moments of the gradients. It gives efficient parameter updates during training, leading to a faster convergence as seen in the plot. Training for 100 epochs allows the model to capture complex relations and perform better on test and validation sets and shuffling the training data and dividing it into batches of 128 items helps to introduce randomness and diversity into the training process.

Fig 1:Train Loss and Validation for Model 1

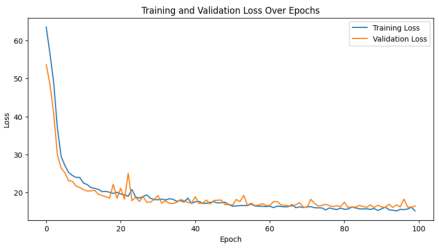

Fig 2: Train Loss for Model 2 (Worse)

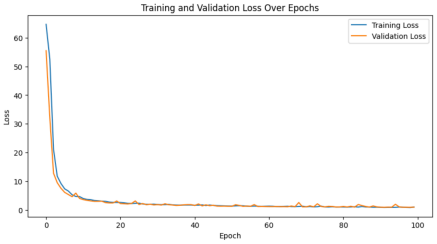

Fig 3: Train Loss for Model 3