**Assignment 4 – Report**

See the README.txt file for details on changing hyperparameters, package installations, and how to run the files.

**Hyperparameters**

These hyperparameters were used for all models:

Number of target classes = 101

Path to food dataset = “./data/food/”

Batch size = 25

Number of workers = 4

Number of GPUs = 1

Maximum epochs = 8

Learning rate = 1e-3

**Basic CNN**

*Chosen Architecture:*

The following architecture similar to AlexNet was used:

1. Convolutional layer with 3 input channels, 8 output channels, and kernel size 11
2. ReLU activation function
3. 2D Max pooling with kernel size 2
4. Convolutional layer with 8 input channels, 32 output channels, and kernel size 5
5. ReLU activation function
6. 2D Max pooling with kernel size 3
7. Convolutional layer with 32 input channels, 128 output channels, and kernel size 3
8. ReLU activation function
9. 2D Max pooling with kernel size 2
10. Fully connected layer using 32758 input features and 1152 output features
11. ReLU activation function
12. Fully connected layer using 1152 input features and 576 output features
13. ReLU activation function
14. Fully connected layer using 576 input features and 256 output features
15. ReLU activation function
16. Fully connected layer using 256 input features and 101 output features

Text

Description automatically generated

*Training Loss:*

*Chart, line chart

Description automatically generated*

*Validation Accuracy:*

*Chart, line chart

Description automatically generated*

*Validation Loss:*

*Chart, line chart

Description automatically generated*

*Final Test Accuracy:*

Graphical user interface, application

Description automatically generated

**Chart, scatter chart

Description automatically generated**

**All Convolutional Net**

*Chosen Architecture:*

The following architecture similar to AlexNet was used:

1. Convolutional layer with 3 input channels, 8 output channels, and kernel size 3
2. ReLU activation function
3. Convolutional layer with 8 input channels, 32 output channels, kernel size 3, and strides 2
4. ReLU activation function
5. Convolutional layer with 32 input channels, 64 output channels, kernel size 5, and strides 2
6. ReLU activation function
7. Convolutional layer with 64 input channels, 128 output channels, kernel size 3
8. ReLU activation function
9. Convolutional layer with 128 input channels, 101 output channels, kernel size 2

Text

Description automatically generated

Here we see the total number of parameters in the all convolutional model is 179,000 parameters. In the basic CNN used in the previous section, the total number of parameters was 38.6 million parameters.

*Training Loss:*

*Chart

Description automatically generated*

*Validation Accuracy:*

*Chart, line chart

Description automatically generated*

*Validation Loss:*

*Chart, line chart

Description automatically generated*

*Final Test Accuracy:*

*Graphical user interface, text, application

Description automatically generated*

*Chart, scatter chart

Description automatically generated*

**Regularization**

*Chosen Model:*

BasicCNN model

*Additions:*

Added two dropouts after the 2nd and 3rd fully connected layer

*New Architecture:*

1. Convolutional layer with 3 input channels, 8 output channels, and kernel size 11
2. ReLU activation function
3. 2D Max pooling with kernel size 2
4. Convolutional layer with 8 input channels, 32 output channels, and kernel size 5
5. ReLU activation function
6. 2D Max pooling with kernel size 3
7. Convolutional layer with 32 input channels, 128 output channels, and kernel size 3
8. ReLU activation function
9. 2D Max pooling with kernel size 2
10. Fully connected layer using 32758 input features and 1152 output features
11. ReLU activation function
12. Fully connected layer using 1152 input features and 576 output features
13. ReLU activation function
14. Dropout
15. Fully connected layer using 576 input features and 256 output features
16. ReLU activation function
17. Dropout
18. Fully connected layer using 256 input features and 101 output features

*Text

Description automatically generated*

*Training Loss:*

*Chart, background pattern

Description automatically generated*

*Validation Accuracy:*

*Chart, line chart

Description automatically generated*

*Validation Loss:*

*Chart, line chart

Description automatically generated*

*Final Test Accuracy:*

*Graphical user interface, text, application

Description automatically generated*

*Chart, scatter chart

Description automatically generated*

**Transfer Learning**

*Pre-trained model used:*

GoogLeNet which is 22 layers deep, 27 layers when pooling layers are included.

Image of full architecture can be found [here](https://dillhoffaj.utasites.cloud/ox-hugo/2022-04-14_15-07-51_screenshot.png).

*Changes:*

Excluded the last layer. Added linear layer with the number of filters the layer before the last layer as the input features and the target classes (101) as the output feature.

Text

Description automatically generated

*Training Loss:*

*Chart, line chart

Description automatically generated*

*Validation Accuracy:*

*Chart, line chart

Description automatically generated*

*Validation Loss:*

*Graphical user interface, chart, line chart

Description automatically generated*

*Final Test Accuracy:*

**Graphical user interface, text, application

Description automatically generated**

**Chart, scatter chart

Description automatically generated**