Individual Report – Aaron Skipper

Description of my individual portion to the project was initiating the group report.

Introduction

Classification problems are types of problems that present itself in many professional industries. Experts and professionals within their perspective fields use classification techniques to better operate and understand problems that their businesses are faced with, and these types issues are general enough to explain several problems across a general working industry. We are motivated with the idea of continuing to enhance our learning and understanding of classification problems, and we selected the Kaggle produced natural image dataset to experiment with.
We will attempt to train a convolution neural network (CNN) to correctly classify these images according their classes. We utilized the PyTorch framework to express our experimenting within the CNN. For image analysis, CNN works well as they specialize in making sense of patterns. We will experiment with the analysis of the problem statements of how accurately we can build a neural network model that classifies the 8 classes within the dataset and identify which parameters within the model influence the neural network's accuracy and time to perform.

Description of the deep learning network and training algorithm

We choose to use to use CNN to address this classification problem on the nature dataset. Our baseline model consists of 2 convolutional layers using the framework PyTorch. We used batch size of 128 images, with each image having a dimension of 224x224. Each layer of the network has parameter settings that will influence the layer interactions along with the performance of the network. The first convolution layer has input nodes set to 16, and output nodes set at 32. The convolutional filters (kernels) are set to be 3x3. The batch normalization setting that follow the

convolutions layer, is used to address the network phenomena of internal covariate shift. Mean and std of that layer is considered and can be used as tuning parameters for the layer, with batch normalization, but they were not used in our experimenting. Covariate shift is when inputs of a layer consistently change during training, due to the other layers' input activations changes that occur within those layers. ReLU, non-linear activation functions are then used following the convolution setting. The max pooling method of convolution is applied to each layer, which acts to reduce computational load as out dimensions are decreased after applied. In the final layers of the network, the fully connected layer is necessary to complete the pattern recognition process and perform the classifications. This ensures that all the nodes at the end of the network will be used for classification. The SoftMax transfer function was used as the output layer, which serves well for pattern recognition networks with multiple neurons.

Slides for presentation were spun up.

Basic 2 layer CNN model and accuracy done. Since the 2 CNN network was the basis for all of the other models, it served as a contribution to all of the codes done by my counterpart.

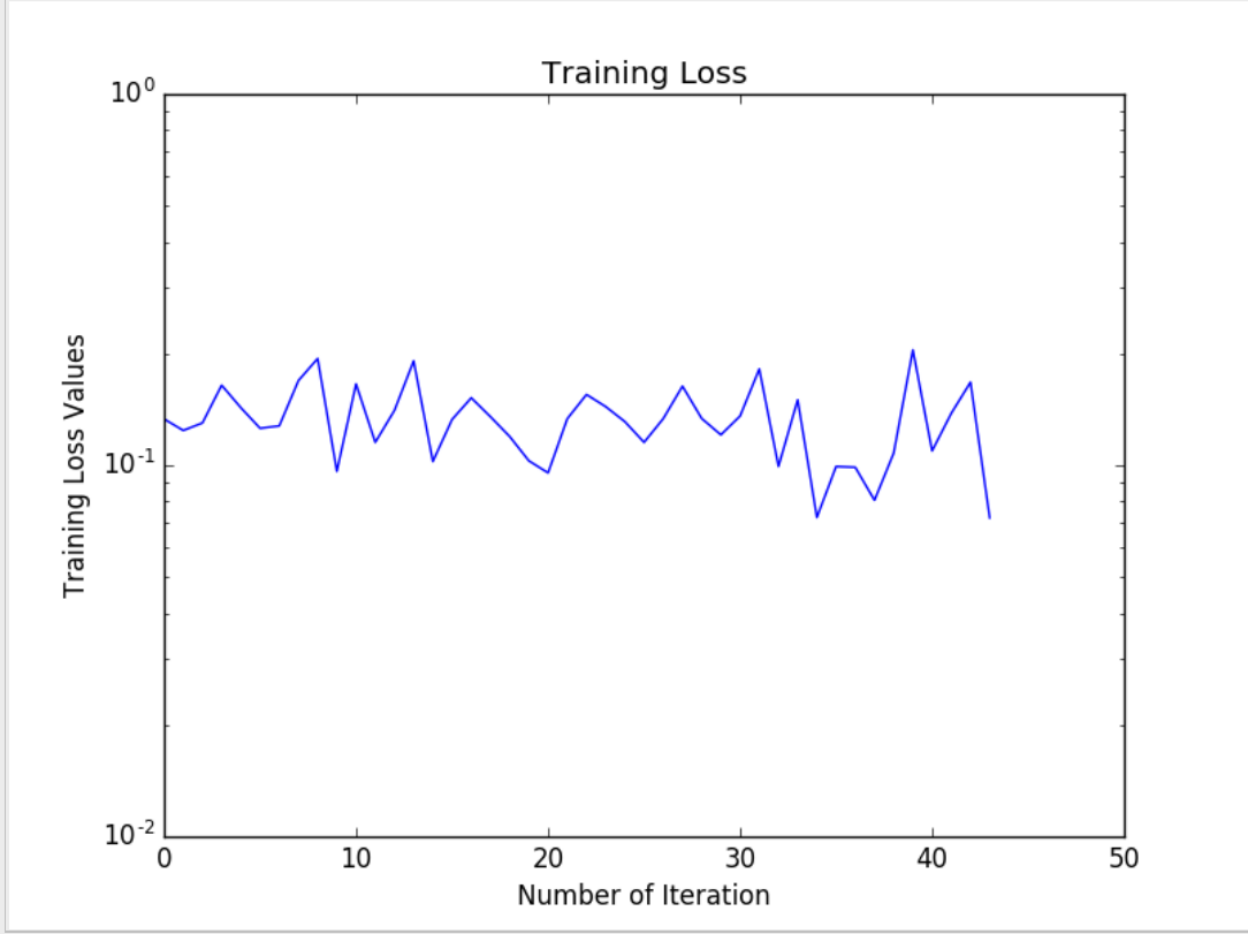
The confusion matrix for the nat_img2.py shows that

1 Misclassification for airplane as flower, 1 misclass of cat as a dog, Dog miscalled twice 1 as cat and 1 as car.

```
classes                                                    ...
airplane    727.0   93.155433   19.365537   54.0  ...  2
car         968.0  100.000000    0.000000  100.0  ...  1
cat         885.0  303.732203   95.744543   73.0  ...  2
dog         702.0  311.933048   99.654237   50.0  ...  2
flower      843.0  328.167260  189.871326   59.0  ...  2
fruit      1000.0  100.000000    0.000000  100.0  ...  1
motorbike   788.0  109.114213   12.083767   66.0  ...  1
person      986.0  255.981744    0.330622  250.0  ...  2

[8 rows x 16 columns]
5519 690 690
torch.Size([128, 3, 224, 224]) torch.Size([128])
Test Accuracy of the model on the 690 test images: 90 %
Predicted  0  1  2  3   4   5  6  7  All
Actual
0          4  0  0  0   0   0  0  0    4
1          0  6  0  1   0   0  0  0    7
2          0  0  2  1   0   0  0  0    3
3          0  0  1  4   0   0  0  0    5
4          0  0  0  0  11   0  0  0   11
5          0  0  0  0   0  10  0  0   10
6          0  1  0  0   0   0  2  0    3
7          0  0  0  0   0   0  0  7    7
All        4  7  3  6  11  10  2  7   50
```

Training Loss for basic model:



GITHUB Repo setup