Assignment-2 Nvidia Report

Deep Neural Networks: GPU Task 1

Epochs	Batch Size	Solver Type	Base Learning Rate	Network	Accuracy	Loss	Time Taken	Model Type
2	Network Default	SGD	0.01	Alexnet	48.91	51.09	13 sec	New Model
100	Network Default	SGD	0.01	Alexnet	100	0.004	2min 47secs	Pre-Trained
25	Network Default	SGD	0.01	Alexnet	100	0.5	32sec	Pre-Trained
50	Network Default	SGD	0.01	LeNEt	100	0	2min	Pre-Trained
50	Network Default	SGD	0.01	LeNEt	48.04	0.6	2min 3sec	New Model
30	Network Default	SGD	0.01	GoogleNet	73.22	0.7	51sec	New Model
60	Network Default	SGD	0.01	GoogleNet	100	0.25	1min 1sec	Pre-Trained

AlexNet - 5 CNN & 2 fully connected layer (ANN) plus a softmax layer
Each CNN -> convolution filters and a non linear activation (relu) - 3 layers have max pooling
input size to be fixed (224,224,3)

GoogleNet has 22 layers

In this experiment we can determine that the more we train the model, the more it learns. When the number of epochs was very small the model couldn't predict correctly. Increasing the epochs to a certain level, increased the accuracy.
And also when a pretained model is trained again, it is able to shape its weights and biases accordingly so that it gives a better accuracy and lower loss.

Deep Neural Networks: GPU Task 2

Epochs	Batch Size	Solver Type	Base Learning Rate	Network	Accuracy	Train Loss	Val Loss	Time Taken	Model Type
5	**Network Default**	**SGD**	**0.01**	**Alexnet**	**81.37**	**0.42**	**0.407**	**3min 50sec**	**New**
10	**Network Default**	**SGD**	**0.01**	**Alexnet**	**89.15**	**0.29**	**0.25**	**7min 28sec**	**New**
5	**Network Default**	**SGD**	**0.01**	**LeNet**	**50**	**0.69**	**0.6**	**6min 5sec**	**New**
5	**Network Default**	**SGD**	**0.01**	**GoogleNet**	**76.29**	**0.44**	**0.49**	**16min 18sec**	**New**

In this experiment we created our new dataset and also alloted 75% for training and 25% for validation

After that we created several new models and trained with the data and examined its validation accuracy

When tested across new unseen images by the model, its accuracy was close to 90% in several models

Deep Neural Networks: GPU Task 3

The Architecture and Weights of models are stored in the job directory.

Caffe model with GPU is set up. A classifer object is created

Input preprocessing is performed on the image loaded

Image is first resized and later normalized

Forward Propogation is performed on the model

Postprocessing the image to generate useful input

Deploy them finally as a python package or application

Deep Neural Networks: GPU Task 4

Epochs	Batch Size	Solver Type	Base Learning Rate	Network	Accuracy	Train Loss	Val Loss	Time Taken	Policy
7	Network Defaults	SGD	0.0001	Pretrained	82.69	0.421	0.38	4min 55sec	Fixed
4	Network Defaults	SGD	0.001	Pretrained	82.382	0.31	0.38	2min 54sec	Exponential Decay

- The accuracy increases from where the pretrained model accuracy was last left
- As the number of epochs increases the accuracy also increases
- The rate of accuracy increase gets gradually small, so this decides to change other parameters

The main 4 catergories are:

1. **Data**
2. **Hyperparameters**
3. **Training time**
4. **Network Architecture**

- wget can be used to download data from web directly to server working without pulling it to local machine
- Passing the weights and architecture we can create a model and predict the test images
- Preprocess the image
- predict the class

Deep Neural Networks: GPU Task 5

There are 3 approaches for object detection:

1. sliding window
2. modify neural architecture
3. DetectNet

Sliding window:

This approach runs classifier on each 256X256 segment, we saw it identified the dog present in the picture to some extent but not completely.

Modify the Alexnet's architecture by replacing a convolution layer with some other convolutional layer.

Require labelled data. Labelling data is curating of input output mappings. In this case our input is image of any size and output is location of our object.

Assessment Model:

Topic: Model to identify that the image has a whale face or not

Model configurations used:

Epochs = 7

Learning Rate= 0.0001

Activation Function: Adam

Model accuracy = 95

successfully deployed the model