CS8803DL: Assignment 3

March 14, 2016

Environment setup

Setup your environment following the instructions: Piazza post.
In this assignment we will be using the Flickr Style dataset for classifying the given image into 20 different classes. The dataset and models are uploaded at: model.zip: `https://www.dropbox.com/s/ykor3d0d0uxmawp/models.zip?dl=0` dataset.zip: `https://www.dropbox.com/s/gw6ac20l5utv0bx/flickr_style.zip?dl=0` Download the zip file and unzip it into the same folder as the starter code provided. Note: this assignment will require installing a package called the loadcaffe. A new docker image "ashshenoi8:v2" has been released with these changes.

Part 1: Building your own CNN [30 pts]

Implementing the CNN

For this part you will have to edit the given *model.lua* file. There are various TODO sections that you will need to code. Basic architecture of should be:

Input → conv → Activation → Pooling → conv → Activation → Pooling → Linear → Activation → Linear → SoftMax

Decide the parameters of theses based on the constraints that your input of the network will be a 224x224x3 image and the output of the network should be 20 classes.

Once you have the basic architecture setup, experiment with:

1. Different activation Tanh vs ReLU
2. Dropout: add a dropout just before the first fully connected layer

In your report, include the graphs of training and testing accuracies. Also mention the calculations for coming up with the various parameters.

Command line:

```
qlua run.lua --model CNN -b 10
qlua run.lua --model CNN_TANH -b 10
```

Data Augmentation

Data augmentation helps in increasing the training samples which can help the model from over-fitting. For this part of the assignment you will edit the function *data_augmentation()* in *train.lua*. The training samples contain images of size 256x256, various data augmentation you could perform are cropping and mirroring. Use the image package for this exercise. **Command line:**

```
qlua run.lua --model CNN -b 10 --dataAugment
```

Filter Visualization

Often while training your network, it is important to visualize the weights of the convolutional layers to see if they are learning features that make sense. Use the saved nets in the results folder to visualize the weights of the first two convolutional layers. Use the *filter_visualize.lua* for this exercise.

Include the filter visualization in your report.

Command line:

```
qlua filter_visualize.lua
```

NOTE: For part 1, it is sufficient to run the training for around 50 epochs.

Part 2: Fine-tuning [40 pts]

In this exercise, we will load a pre-trained network (VGG_CNN_M) and retrain for our task. Read more about the model here: `https://gist.github.com/ksimonyan/f194575702fae63b2829`

Modifying a pre-trained network

In order to use the pre-trained model, you will need to first replace the last fully connected layer to match the number of outputs that we desire(20). Edit the *model.lua* under the option for "CNN_FINETUNE". Use functions provided in the Sequential module for this purpose.

Training

When you fine tune a model, you can either freeze all the layers except for the last or tune the final layer with a learning rate higher than the rest of the layers. Edit the train.lua to do this The train.lua has partial code for training with different leanring rates. You may choose to do the training with the layers frozen.

Some reference for freezing:
`https://gist.github.com/soumith/6cd0f9b8462d0507a91b`
Include the training and testing accuracy graphs in your report.
Command Line

```
qlua run.lua --model CNN_FINETUNE -b 10 --dataAugment
```

NOTE: For part 2, the training is slow on CPU. So in interest of time it is sufficient to run the training for a few epochs(3-4).

Part 3: Class model Visualization [30 pts]

In this part, we will visualize class model learned by a model. You need to edit the *ClassModelVisualize.lua* file for this exercise.

In the skeleton code, we load the VGG_CNN_M model. You need use back-propagation to optimize the image input to generate the class model image. It is also recommended to remove the last Softmax layer before you do this.

In your write up include the few class models that your code generates. You can refer to

Command Line

```
qlua ClassModelVisualize.lua
```

Required submission

Please, submit a folder named with your first name and last name such as $firstname\_lastname$. You should upload the attachment as a zip file with the same naming format, $firstname\_lastname.zip$. The zip file should contain the write up in PDF format and your code.

Some things to keep in mind: You will be penalized if your code doesn't run on our setup. Ensure that you test your code on Docker. Late Submission will be penalized 10% of the total for each day after the deadline.