main

February 15, 2025

###Mount Google Drive

[ ]:
```
# Cell 1
from google.colab import drive
drive.mount('/content/drive')
```

###Part1: Implementing CNN from Scratch

[ ]:
```
# Cell 2
%cd /content/drive/MyDrive/hw2/student_version/part1-convnet/
```

Load CIFAR-10 data

[ ]:
```
# Cell 3
%cd data
!sh get_data.sh
%cd ..
```

Cross-check your implementation

[ ]:
```
# Cell 3
# If you get an error saying test not found, add an __init__.py file in the
# tests directory
!python -m unittest tests.test_vanilla_cnn
```

Train your ConvNet

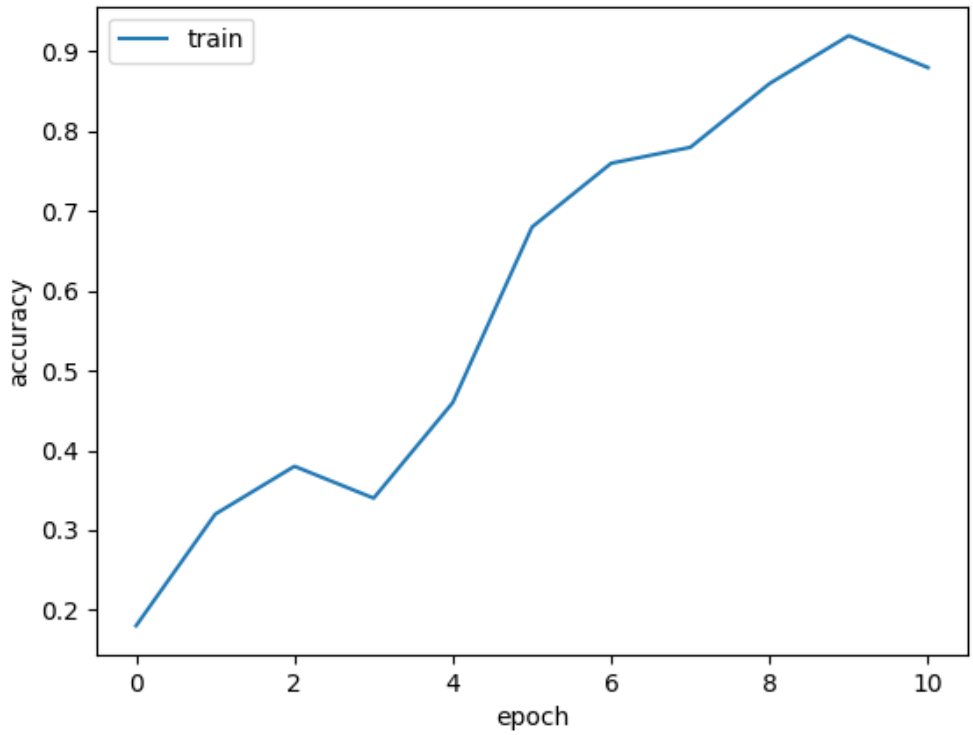
[ ]:
```
# Cell 4
!python train.py
```

```
Finished epoch 0 / 10: cost 2.306153, train: 0.180000, lr 1.000000e-04
Finished epoch 1 / 10: cost 2.251948, train: 0.320000, lr 9.500000e-05
Finished epoch 2 / 10: cost 2.159310, train: 0.380000, lr 9.025000e-05
Finished epoch 3 / 10: cost 1.805048, train: 0.340000, lr 8.573750e-05
Finished epoch 4 / 10: cost 1.545141, train: 0.460000, lr 8.145062e-05
Finished epoch 5 / 10: cost 1.097030, train: 0.680000, lr 7.737809e-05
Finished epoch 6 / 10: cost 0.666996, train: 0.760000, lr 7.350919e-05
Finished epoch 7 / 10: cost 1.318808, train: 0.780000, lr 6.983373e-05
Finished epoch 8 / 10: cost 0.306557, train: 0.860000, lr 6.634204e-05
```

```
Finished epoch 9 / 10: cost 0.097204, train: 0.920000, lr 6.302494e-05
Finished epoch 10 / 10: cost 0.614703, train: 0.880000, lr 5.987369e-05
```

Visualize Part-1 training curve

[ ]:
```
# Cell 5
from IPython.display import display, Image
display(Image(filename='./train.png', width=500))
```



0.1 Zip your Part-1 submission

[ ]:
```
# Cell 6
!python3 collect_submission.py
```

###Part2: PyTorch

[ ]:
```
# Cell 7
%cd /content/drive/MyDrive/hw2/student_version/part2-pytorch/
```

Load CIFAR-10 data

[ ]:
```
# Cell 8
%cd data
!sh get_data.sh
%cd ..
```

Train your Two-Layer Net

[ ]:
```
# Cell 9
!python main.py --config configs/config_twolayer.yaml
```

Train your Vanilla ConvNet

[ ]:
```
# Cell 10
!python main.py --config configs/config_vanilla_cnn.yaml
```

Train your own model

[ ]:
```
# Cell 11
!python main.py --config configs/config_mymodel.yaml
```

0.2 Zip your Part-2 submission

[ ]:
```
# Cell 12
# If you get an error saying test not found, add an __init__.py file in the
# tests directory
!python3 collect_submission.py
```

1 Assignment 2 Writeup

- Name: Adrian Kukla
- GT Email: adrian.kukla@gatech.edu
- GT ID: 904056948

How does the training curve in Part-1 look like?

Answer: The curve spans 10 epochs. The first epoch the accuracy is just under 20%, increasing to just under 90% at epoch 10. As the model gradually learns from the data the accuracy increases. In the initial stages the model is slowly learning as it starts adjusting its weights followed by a big improvement in accuracy at the midpoint. The highest improvement in accuracy is for epoch 4 and 5 (as seen by the steeper slope of the line). The accuracy score decreases between epoch 2 and 3 and epoch 9 to 10. The slight dip towards the end could be a potential sign of overfitting due to the model memorizing the data. Actual graph is shown earlier in the output.

What are the accuracies with these networks?

- Two-Layer Network: 0.3312
- Vanilla ConvNet: 0.4334
- My ConvNet model: 0.6957

[ ]: