VGG Implementation Project

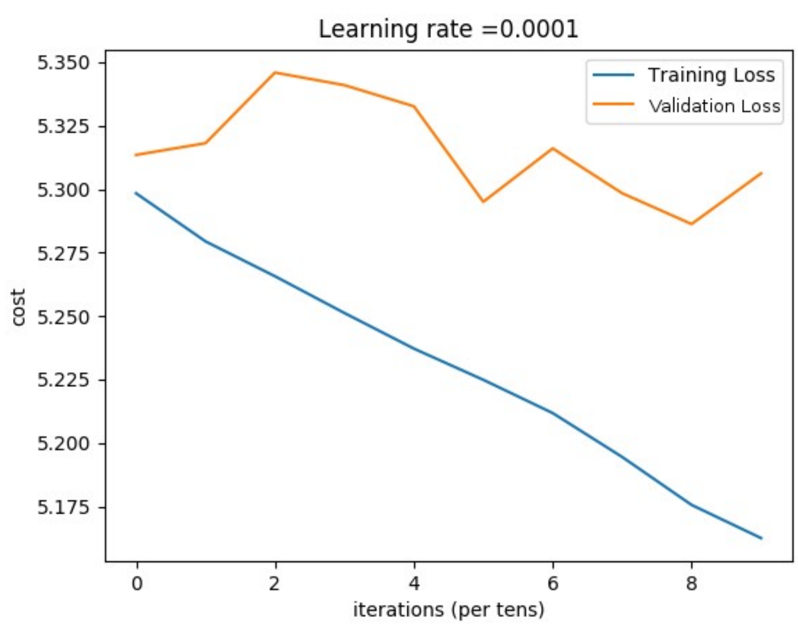
Implementing VGG from scratch was a great learning experience. Due to timing and computation constraints, an exact replication of the ground-breaking VGG16 model on the full ImageNet dataset was not feasible, however, initial results from my work suggest that the simpler implemented model (VGG11) is indeed learning to classify images from the smaller TinyImageNet dataset.

Notable deviations from the original experiment are listed below:

Deviation	In Publication	Implemented
Model	VGG 16	VGG 11
Dataset	Augmented ImageNet • 1.3 Million original images • 1,000 Classes • Various sizes • Augmented via cropping, scaling, etc.	TinyImageNet • 100,000 Images • 200 Classes • All 64x64 • Not Augmented
Epochs Run	74 Epochs	20 Epochs
Training Time	2-3 Weeks on 4x Nvidia Titan Black GPUs	6 Hours on 1 Nvidia Tesla K80

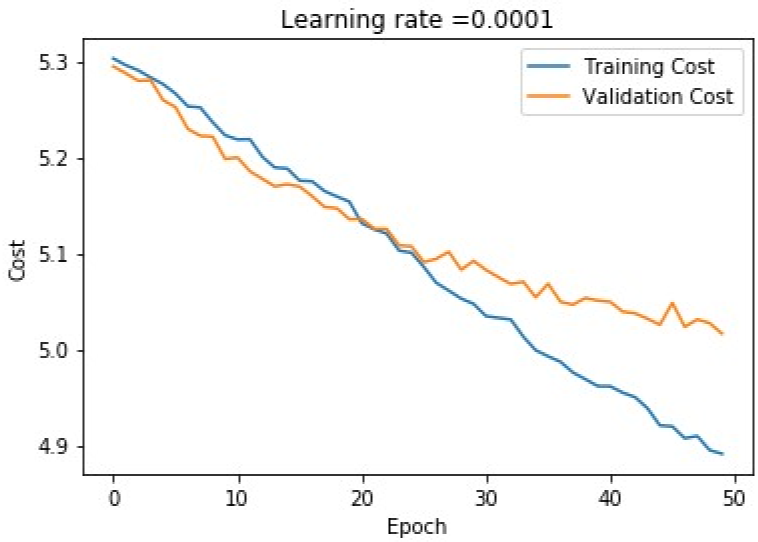
Simple Model

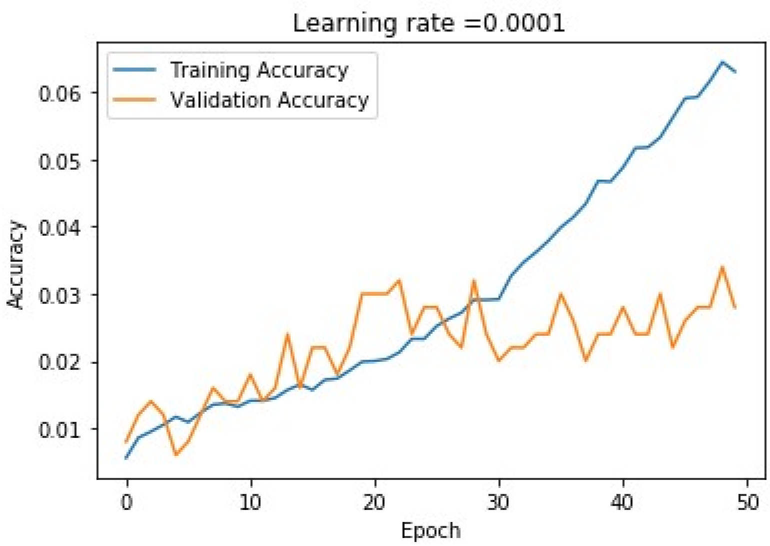
A simple model was built to test that images were being imported properly, memory managed gracefully, and that models were learning as expected. A small subset of the data was used as well to speed up processing. Initial loss metrics on the simple model matched that of random guessing on 200 classes (initial loss approx. 5.3). The simple model then proceeded to show improved loss on the training set with less improvement shown in the validation set. A graph of the loss for the training and validation sets are presented below:



The training error continues to decrease, but a corresponding decrease in validation error is not observed. This suggests that the model does not generalize well. This is not unexpected given the small size of the dataset and simple architecture used.

A second model trained on the small network is visualized below. This model used a larger portion of the data and was run for more epochs. The cost and accuracies are visualized below:





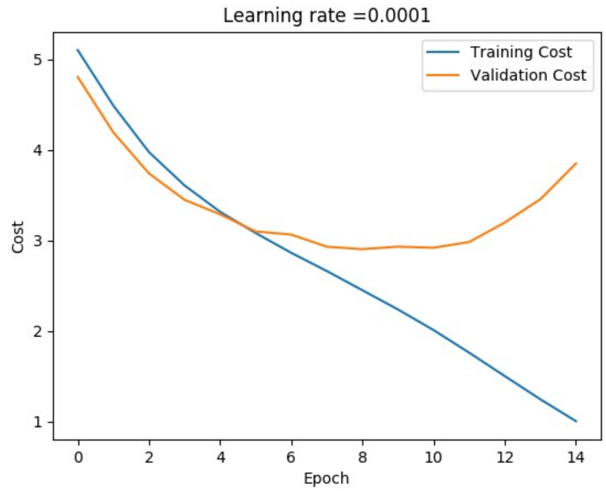
This model has become saturated with respect to improving accuracy on the validation set however the loss on the validation set continues to decrease. This signifies that the model is still making improvements in its prediction probabilities on each epoch despite not increasing its accuracy.

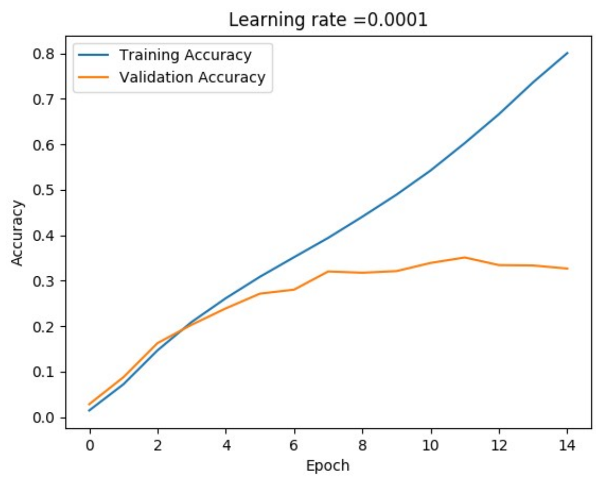
VGG11

2 runs of the full VGG11 architecture (with differing learning rates) were run on the Floydhub Platform.

Model 1 (Larger Learning Rate)

A run of 16 epochs with the full TinyImageNet dataset was run on Floydhub. A graphical representation of the cost and accuracy curves are presented below.





The above model shows clear signs of overfitting the training data. We see that the train cost is very low but the validation/test loss is very high. In fact, the loss on the validation set even reaches levels worse than at the time of random initialization. This means that the network is becoming really confident about some of its incorrect predictions! Reduction of overfitting can be achieved through decreasing the 'keep probability' of the dropout applied to the fully connected layers and by applying a regularization term to the cost function. These hyperparameters could be tuned as time allows.

Model 2 (Smaller Learning Rate)

Due to time constraints and other externalities, this model's run was unexpectedly terminated. As a result, graphical representations of the loss and accuracy curves are not available. Data from that run is presented below:

Epoch 0:

```
2017-11-16 00:26:22,367 INFO - Epoch 0:
2017-11-16 00:26:22,370 INFO - Train cost: 5.16432000935
2017-11-16 00:26:22,370 INFO - Test cost: 4.92807418346
2017-11-16 00:26:22,370 INFO - Train Accuracy: 0.0137399996929
2017-11-16 00:26:22,370 INFO - Test Accuracy: 0.0279999993742
```

Epoch 34:

```
2017-11-16 07:25:29,010 INFO - Epoch 34:
2017-11-16 07:25:29,013 INFO - Train cost: 2.19332818516
2017-11-16 07:25:29,013 INFO - Test cost: 2.93322078347
2017-11-16 07:25:29,013 INFO - Train Accuracy: 0.561669981882
2017-11-16 07:25:29,014 INFO - Test Accuracy: 0.333799988553
```

Model 2 does not exhibit the overfitting demonstrated by model 1. Costs on both the training and validation/test set were still decreasing at the time the job was interrupted. It is likely that were this model to have continued running, the costs would continue to decrease up to a point.

The final results presented by model 2 are not nearly as good as the results presented in the research publication. The VGG11 Model presented therein achieved a top-1 validation error of 29. 6%). The results of model 2 are achieving a 33.3% accuracy or a 66.6% error rate. While this is not nearly as impressive as the paper's results, it does show substantial improvement over a baseline model of random guesses. With more time and resources, it is expected that this number would continue to improve.