

Week 1

Task: Reproduce the LTH results:

Baseline code:

https://github.com/facebookresearch/open_lth

Command to run:

```
python open_lth.py lottery --default_hparams=mnist_lenet_300_100 --levels=3 --dataset_name mnist  
--model_name mnist_lenet_300_100 --batchnorm_init uniform
```

Display save location:

```
python open_lth.py lottery --default_hparams=mnist_lenet_300_100 --levels=3 --dataset_name mnist  
--model_name mnist_lenet_300_100 --batchnorm_init uniform --display_output_location
```

With 3 levels of pruning, weights went to 50%, while accuracy remained around 98%.

My code:

https://github.com/f2010126/LTH_Master

Implemented the LeNet5 architecture and trained with MNIST.

Pruning done using torch.nn.utils.prune modules to generate and apply masks w/o use of extra hooks.

Issues:

- Levels of pruning
- Unsure about the pruning rate equation used as it stagnates badly after the first epoch.
- Combining masks, AND/OR.

But the pruned network did show good performance even at 50% pruned weights. At 80% pruning, the performance went below the baseline initially(95-96%) but was able to train and outdo the baseline. As pruning went above 90%, performance declined.

