TriForce Updates and Classification with New Data

Matt Zhang | Apr. 13, 2018

TriForce Updates

Added ability to split data into train-validation-test. If validation split percentage is not specified, the training set will also be the validation set.

TriForce code significantly refactored. Training histories for all tools now stored internally in a single data structure.

Accuracies split by signal/background, and training history stored by batch/epoch are now stored for all tools.

Plotting scripts now part of default analysis tool.

Additional protections added when reading in options file. TriForce warns when options are missing or mis-set. Certain unspecified options are now given default values.

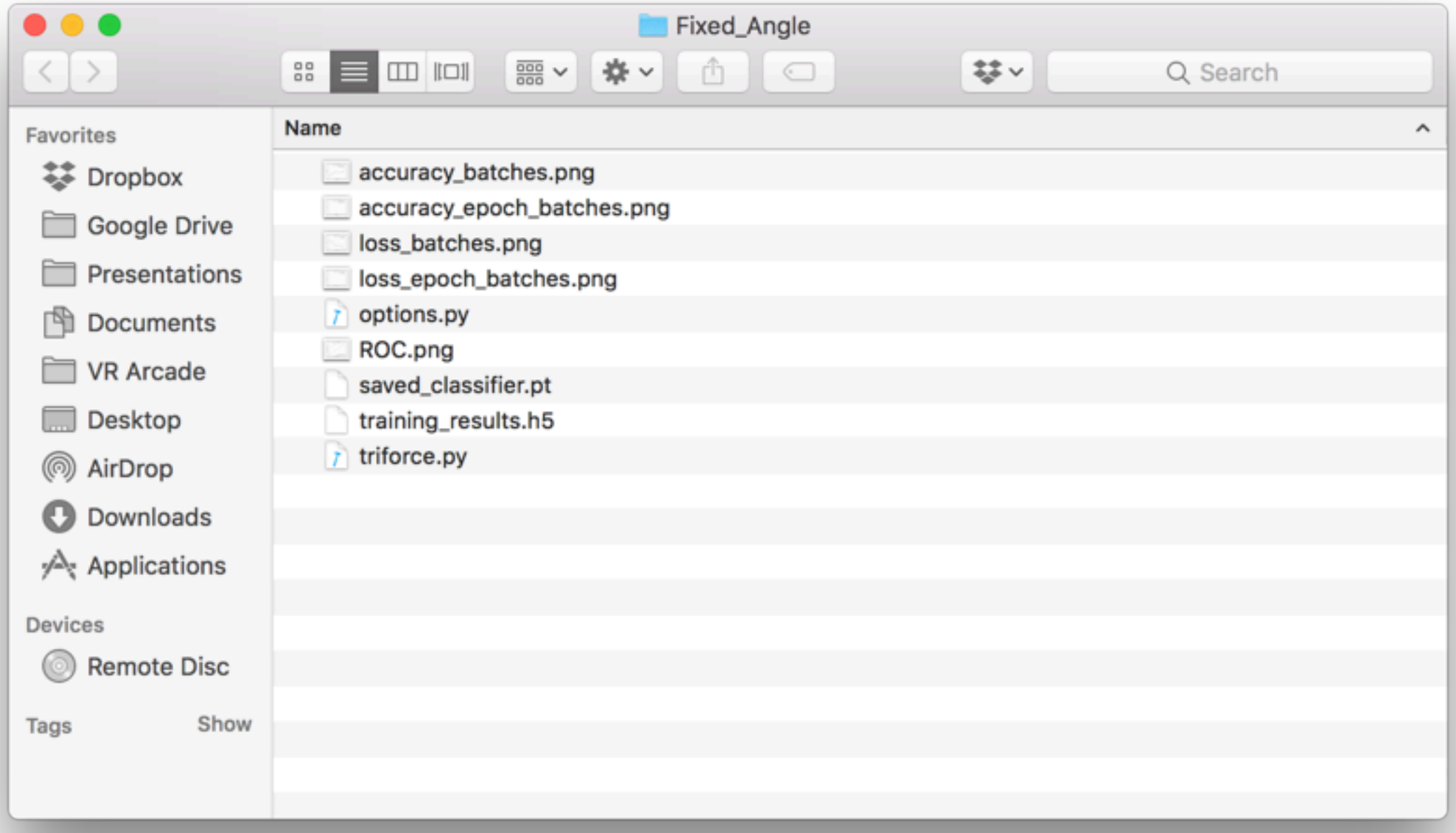
Required Options

samplePath - folders to find samples in
classPdgID - pdg IDs of sample classes
trainRatio - what percent of the data files to use for the training set
nEpochs - end training after this many epochs
relativeDeltaLossThreshold - end training early if delta test loss over an epoch falls below this threshold
relativeDeltaLossNumber - …or if it falls below relativeDeltaLossThreshold for this many sets of batches
batchSize - number of samples in a batch
saveModelEveryNEpochs - save trained model every N epochs (set to 0 to only save at end of training)
outPath - folder to save outputs in

Optional Options

importGPU (False) - whether to use the GPU packages on the CalTech clusters
eventsPerFile (10,000) - how many events are in each file
nTrainMax (-1) - maximum number of files in the training set (-1 for no max)
nValidationMax (-1) - maximum number of files in the validation set (-1 for no max)
nTestMax (-1) - maximum number of files in the test set (-1 for no max)
validationRatio (-1) - what percent of the data files to use for the validation set (-1 if validation set = test set)
nWorkers (0) - number of workers to use in the data loader
calculate_loss_per_n_batches (20) - calculate loss every N batches
test_loss_eval_max_n_batches (10) - the maximum number of test batches to use when calculating test loss
earlyStopping (True) - set to false to turn off early stopping during training

Outputs


Fixed_Angle
Search
Favorites
Dropbox
Google Drive
Presentations
Documents
VR Arcade
Desktop
AirDrop
Downloads
Applications
Devices
Remote Disc
Tags
Show
Name
accuracy_batches.png
accuracy_epoch_batches.png
loss_batches.png
loss_epoch_batches.png
options.py
ROC.png
saved_classifier.pt
training_results.h5
triforce.py

Running on New Data

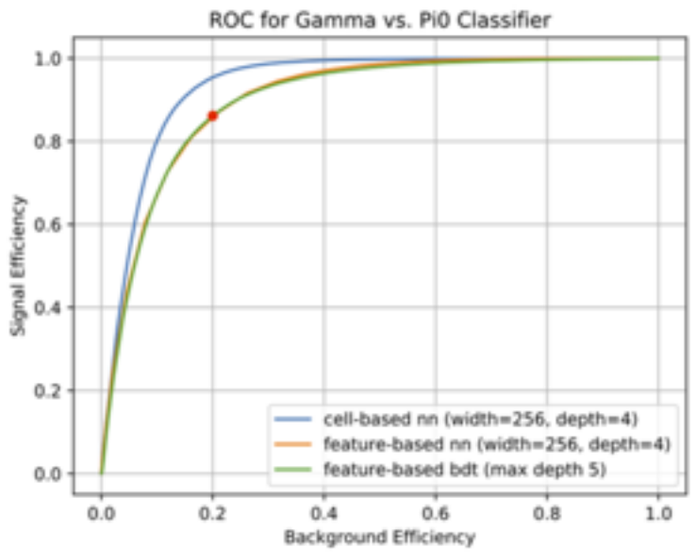
Only ECAL used.

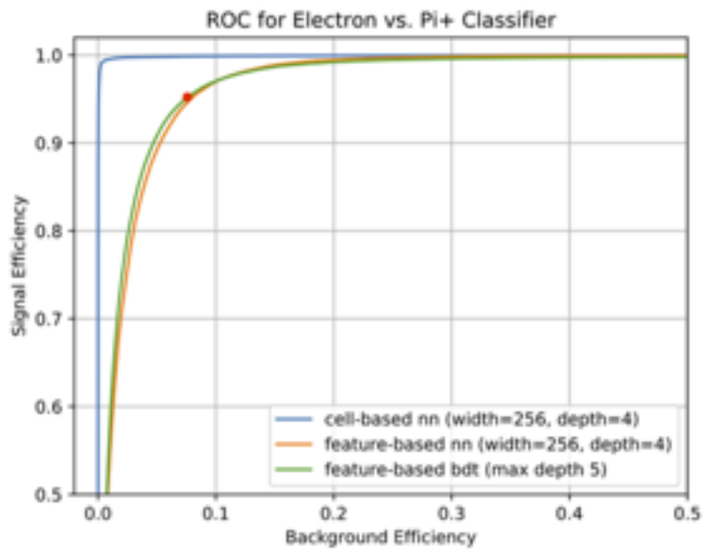
Size set to 51x51x51.

NIPS architecture used (aside from side of input layer) for comparison.

Running on gamma vs. pi0 for both fixed-angle and variable-angle data (using the same architecture).

Paper Results

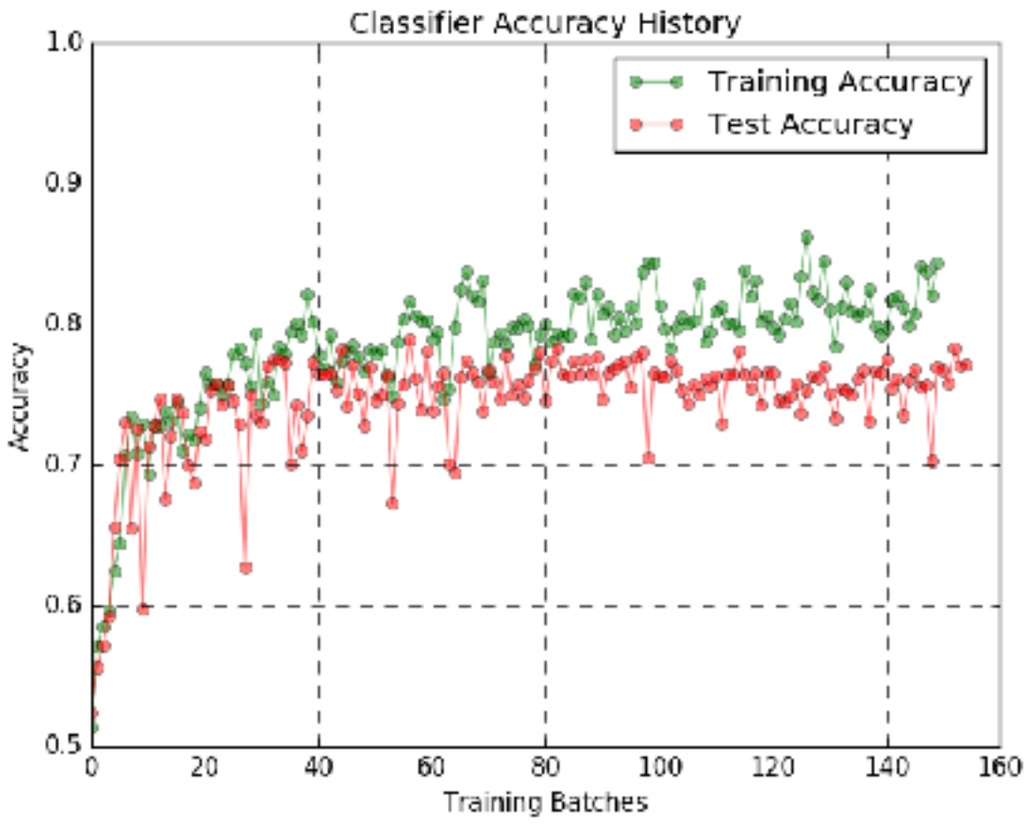


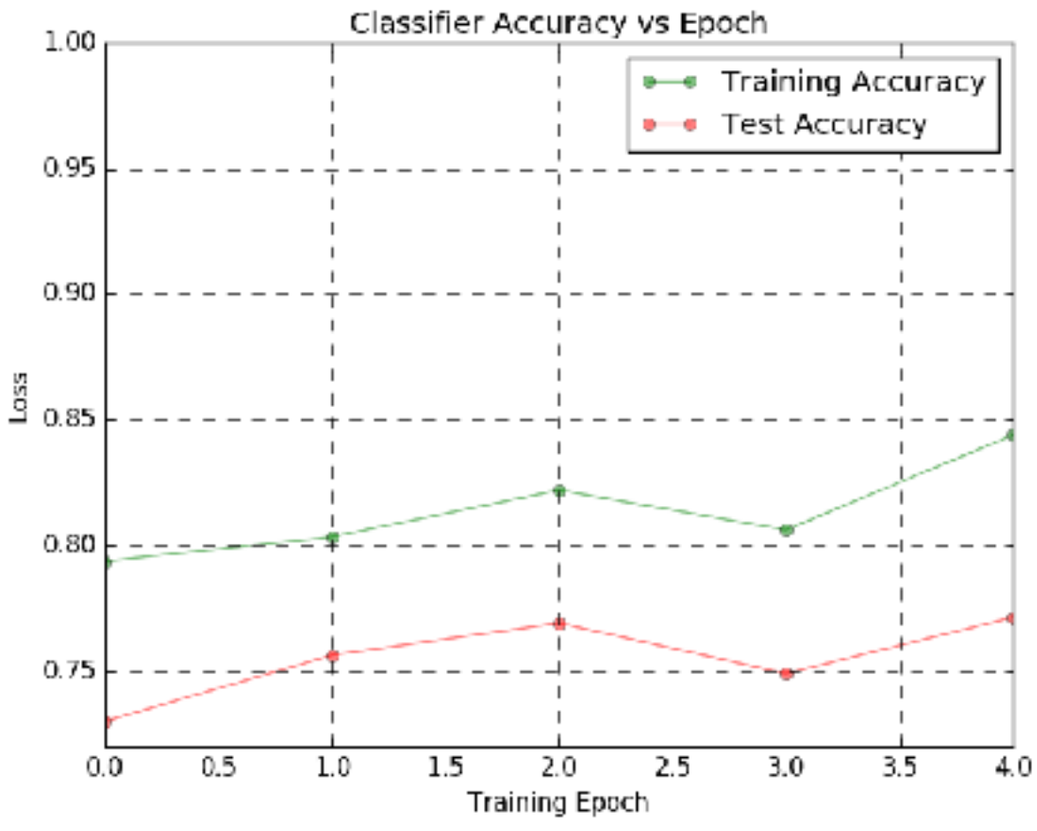


	γ **vs.** π^0				e **vs.** π			
Model	**acc.**	**AUC**	$\Delta\epsilon_{sig}$	ΔR_{bkg}	**acc.**	**AUC**	$\Delta\epsilon_{sig}$	ΔR_{bkg}
BDT	83.1%	89.8%	-	-	93.8%	98.0%	-	-
DNN (features)	82.8%	90.2%	0.9%	0.95	93.6%	98.0%	-0.1%	0.95
DNN (cells)	87.2%	93.5%	9.4%	1.63	99.4%	99.9%	4.9%	151

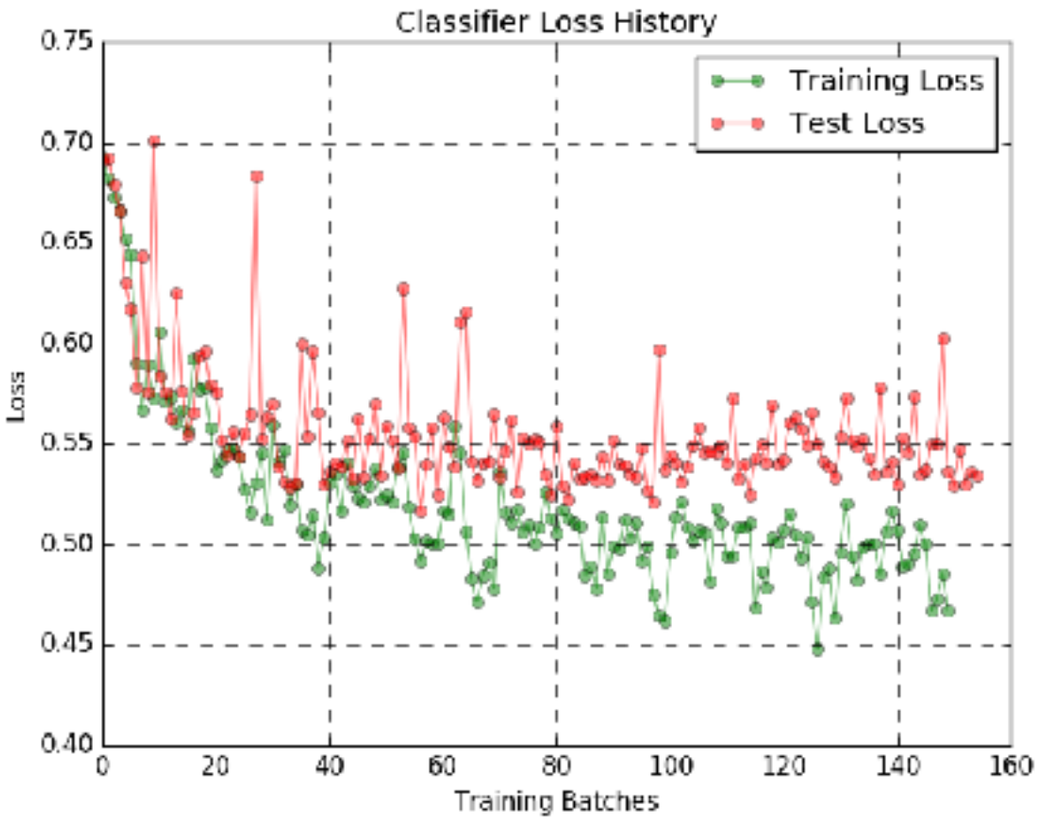
Fixed Angle

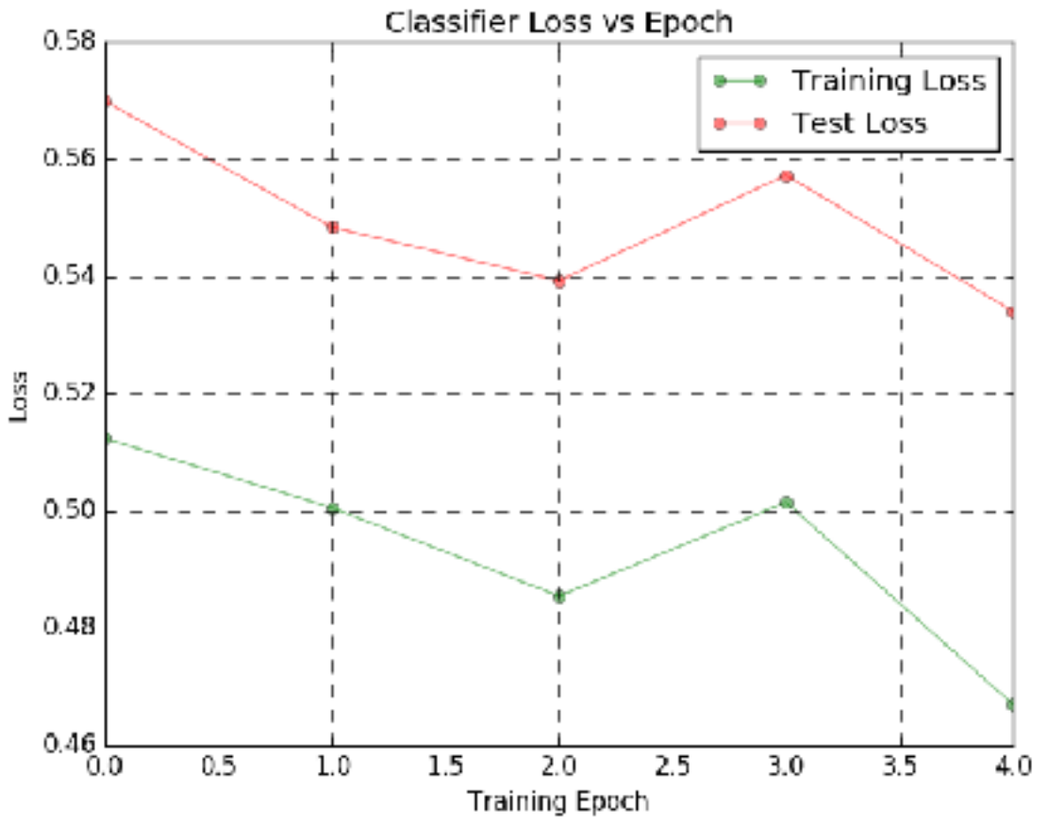
Accuracy

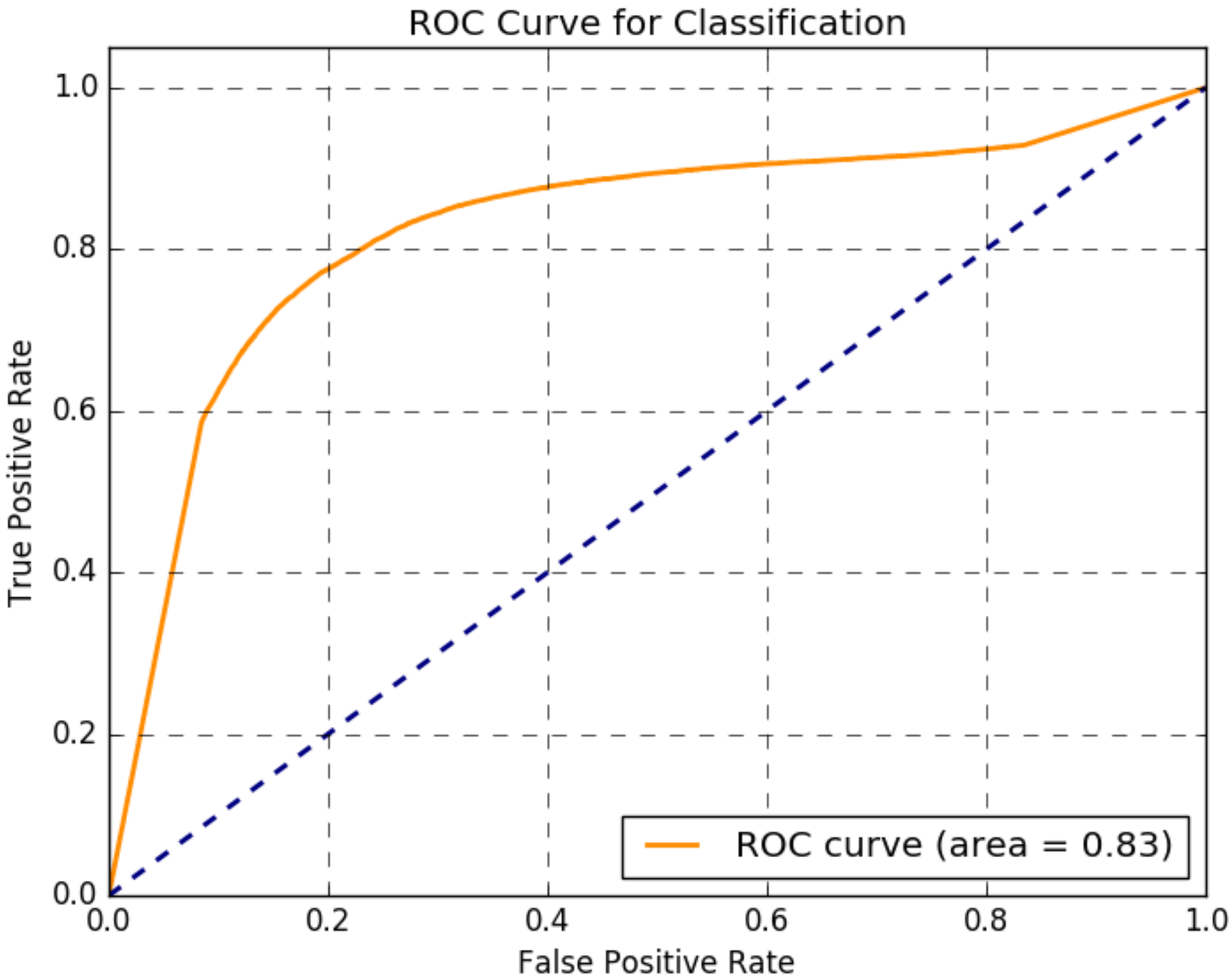

Classifier Accuracy History
Training Accuracy
Test Accuracy
Accuracy
Training Batches
1.0
0.9
0.8
0.7
0.6
0.5
0
20
40
60
80
100
120
140
160


Classifier Accuracy vs Epoch
Training Accuracy
Test Accuracy
Loss
Training Epoch
1.00
0.95
0.90
0.85
0.80
0.75
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Loss


Classifier Loss History
Training Loss
Test Loss
Loss
Training Batches
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0
20
40
60
80
100
120
140
160


Classifier Loss vs Epoch
Training Loss
Test Loss
Loss
Training Epoch
0.58
0.56
0.54
0.52
0.50
0.48
0.46
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


ROC Curve for Classification
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.83)

Takeaway

Net has a harder time training with the larger data window.

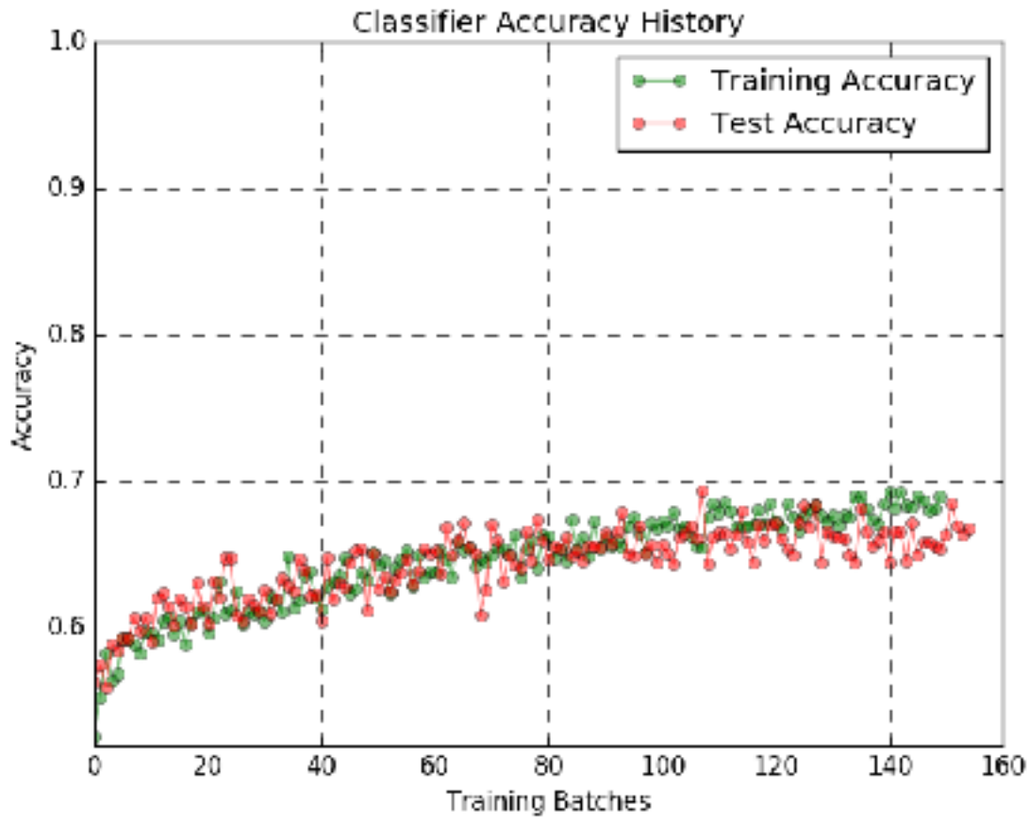
Results are not too bad, but number of neurons in hidden layers should probably be increased.

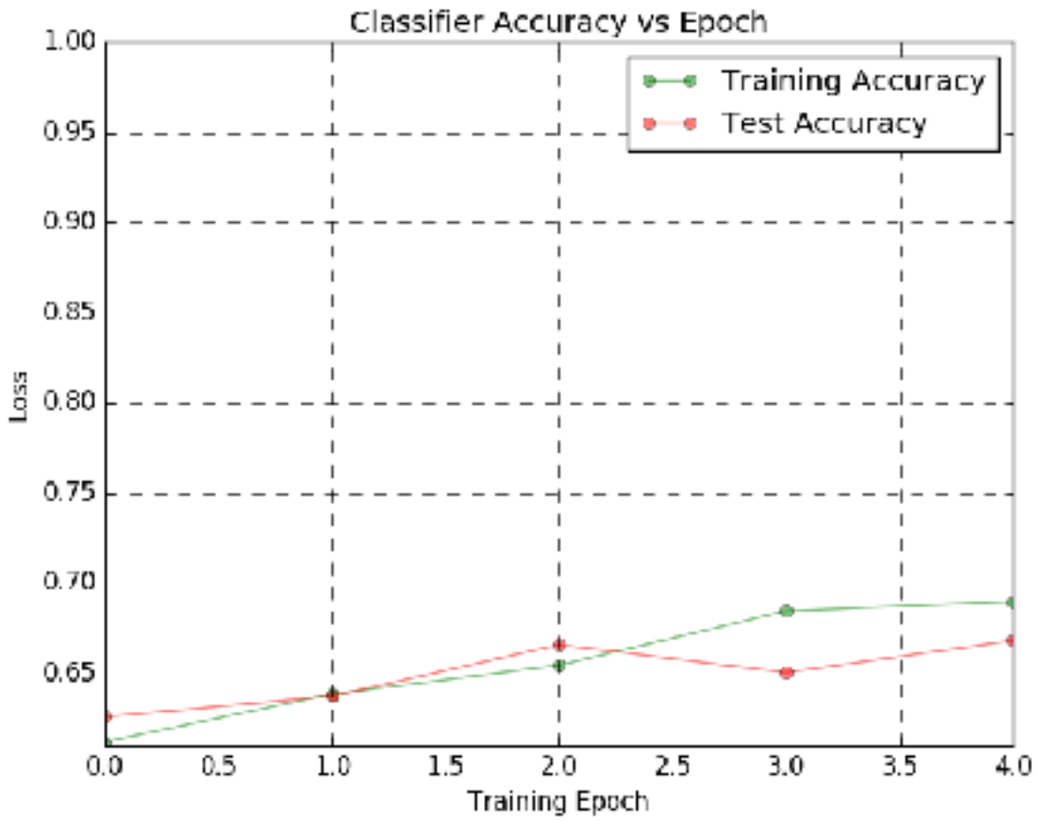
Will run hyperparameter scan on Blue Waters.

Will also train using GoogLeNet.

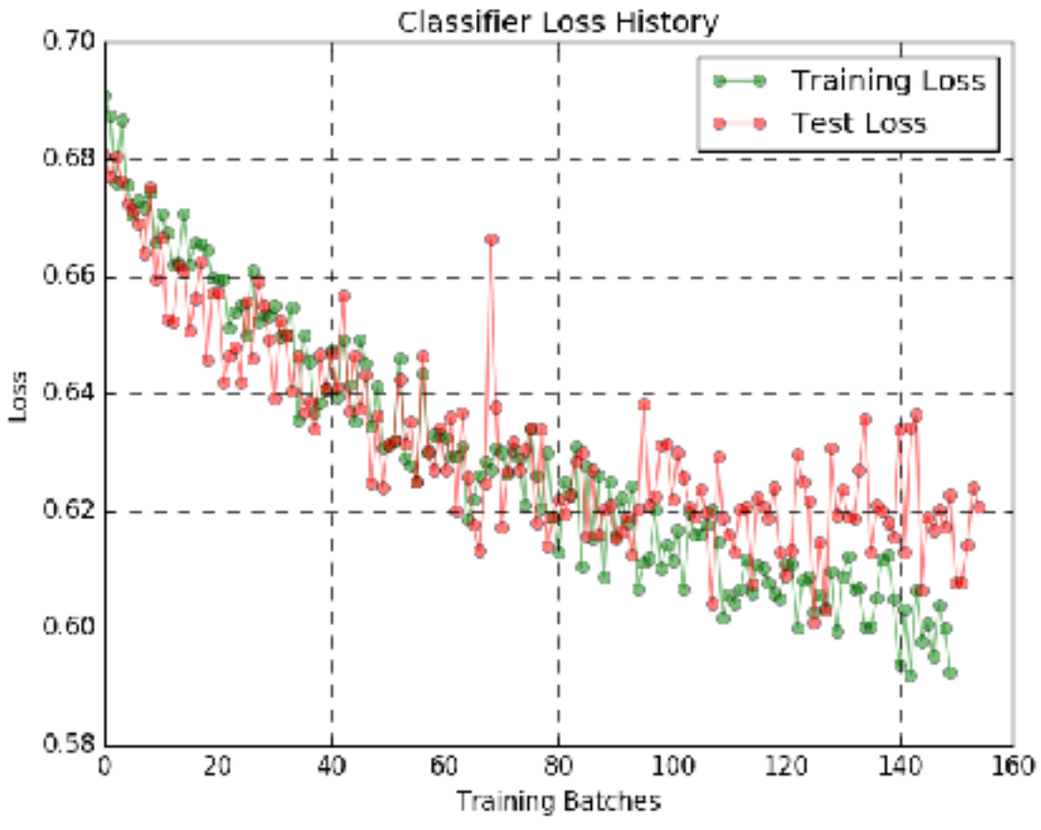
Variable Angle

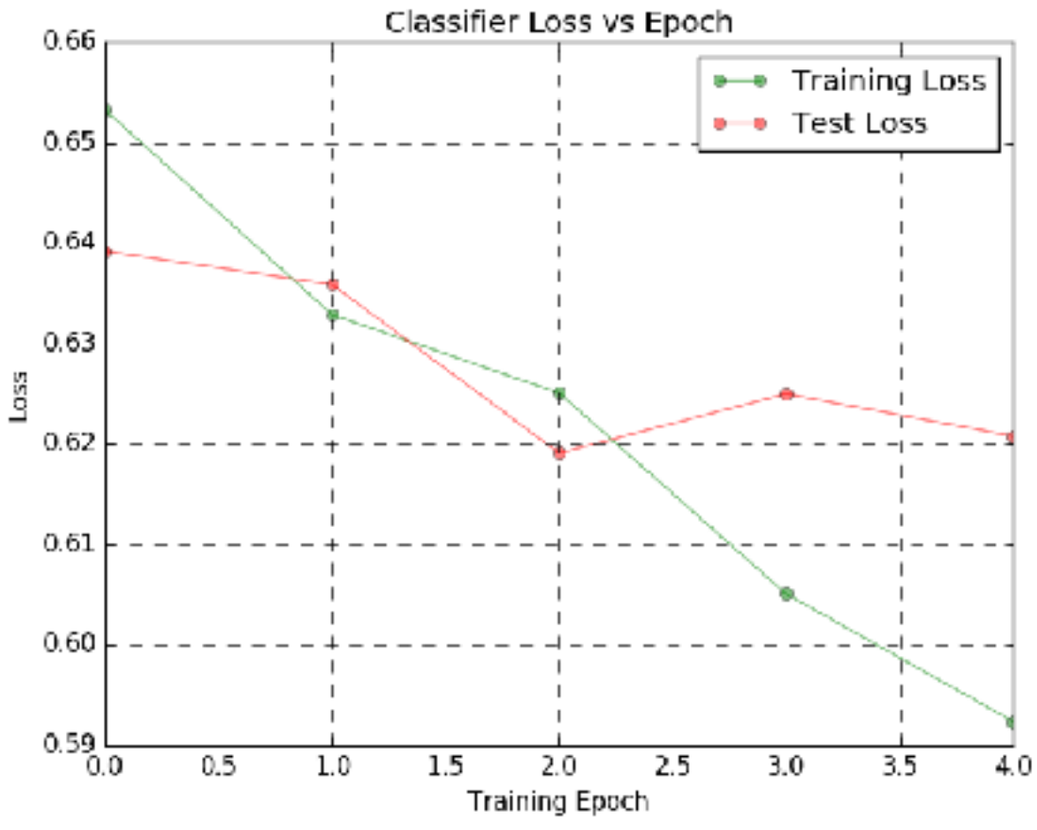
Accuracy

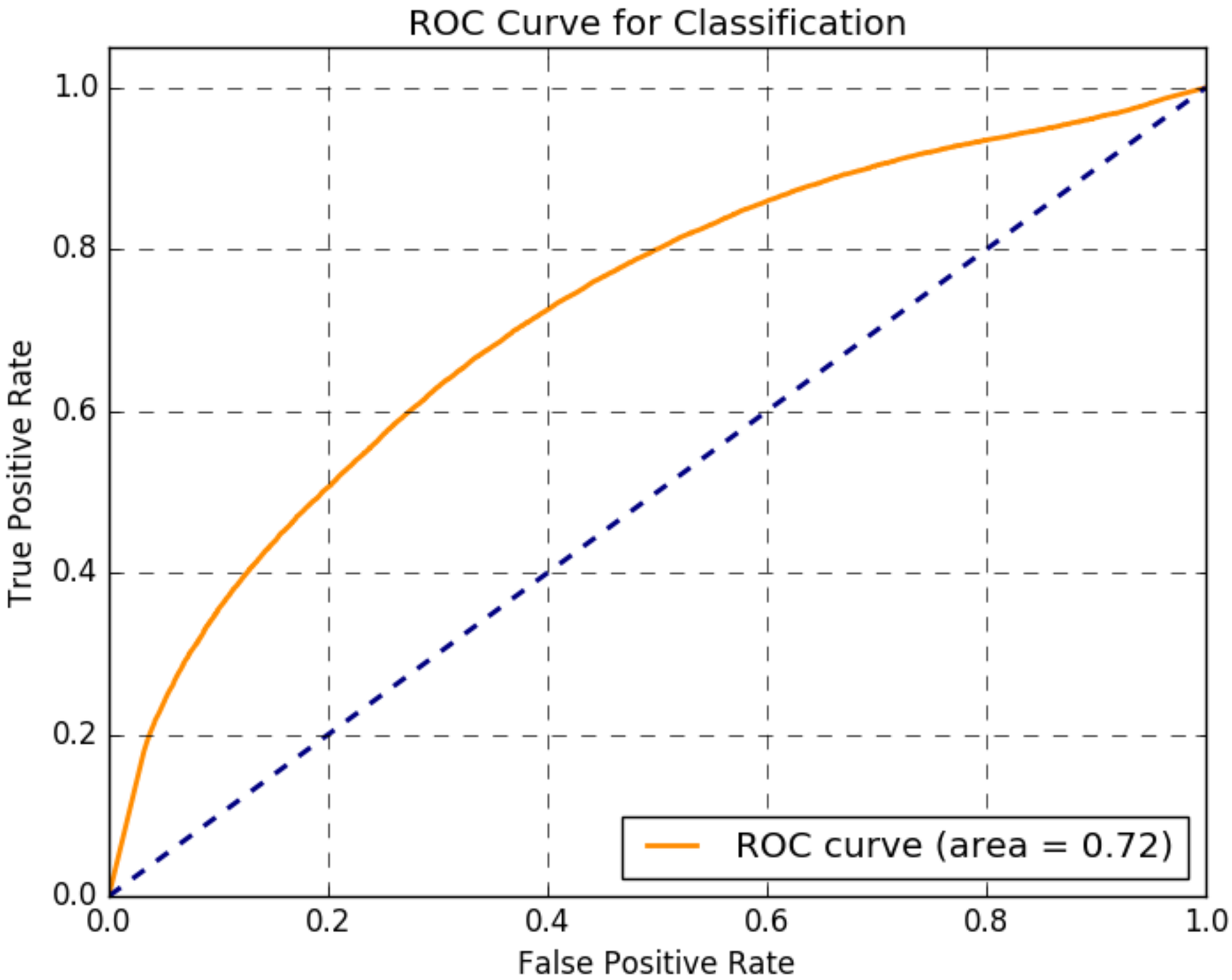

Classifier Accuracy History
Training Accuracy
Test Accuracy
Accuracy
Training Batches
1.0
0.9
0.8
0.7
0.6
0
20
40
60
80
100
120
140
160


Classifier Accuracy vs Epoch
Training Accuracy
Test Accuracy
Loss
Training Epoch
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Loss


Classifier Loss History
Training Loss
Test Loss
Loss
Training Batches
0.70
0.68
0.66
0.64
0.62
0.60
0.58
0
20
40
60
80
100
120
140
160


Classifier Loss vs Epoch
Training Loss
Test Loss
Loss
Training Epoch
0.66
0.65
0.64
0.63
0.62
0.61
0.60
0.59
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


ROC Curve for Classification
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.72)

Takeaway

I am incredibly surprised that we were able to do this well on the variable-angle data.

Simply adding a single extra input for the particle eta may be enough to push up the accuracy. Will try this.