

Calo ML Updates

Matt Zhang | May 11, 2018



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN



Sample Events

The following samples are incomplete (less than 10k events) and will crash TriForce. Remove them before running:

Pi0Escan_5_MERGED/Pi0Escan_5_5.h5
Pi0Escan_5_MERGED/Pi0Escan_5_3.h5
Pi0Escan_5_MERGED/Pi0Escan_5_10.h5
Pi0Escan_7_MERGED/Pi0Escan_7_1.h5
Pi0Escan_6_MERGED/Pi0Escan_6_4.h5

GammaEscan_1_MERGED/GammaEscan_1_4.h5
GammaEscan_7_MERGED/GammaEscan_7_6.h5

Hyperparameter Scan

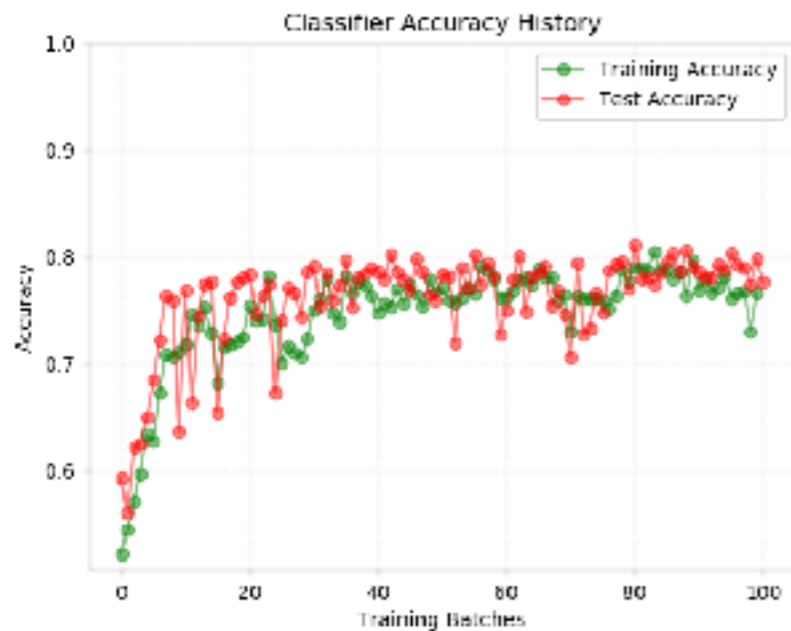
Number of hidden layers = {4, 5, 6}
Number of neurons per hidden layer = {256, 512, 1024}
Learning rate = {0.001, 0.005, 0.01}

Using the following classification net:

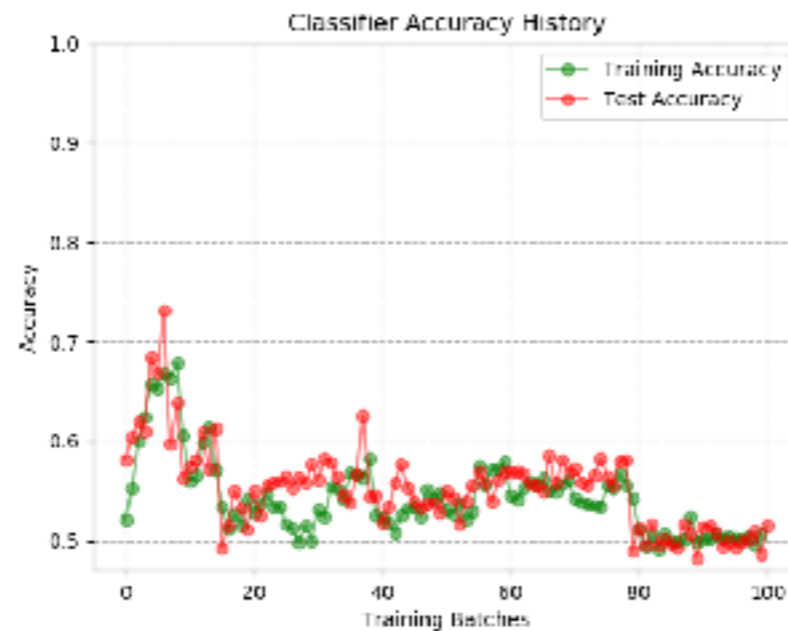
```
class Classifier_Net(nn.Module):
    def __init__(self, hiddenLayerNeurons, nHiddenLayers, dropoutProb):
        super().__init__()
        self.input = nn.Linear(51 * 51 * 25, hiddenLayerNeurons)
        self.input = nn.DataParallel(self.input) # multi-GPU
        self.hidden = nn.Linear(hiddenLayerNeurons, hiddenLayerNeurons)
        self.hidden = nn.DataParallel(self.hidden) # multi-GPU
        self.nHiddenLayers = nHiddenLayers
        self.dropout = nn.Dropout(p = dropoutProb)
        self.output = nn.Linear(hiddenLayerNeurons, 2)
        self.output = nn.DataParallel(self.output) # multi-GPU
    def forward(self, x, _):
        x = x.view(-1, 51 * 51 * 25)
        x = self.input(x)
        for i in range(self.nHiddenLayers-1):
            x = F.relu(self.hidden(x))
            x = self.dropout(x)
        x = F.softmax(self.output(x), dim=1)
        return x
```

Hyperparameter Scan

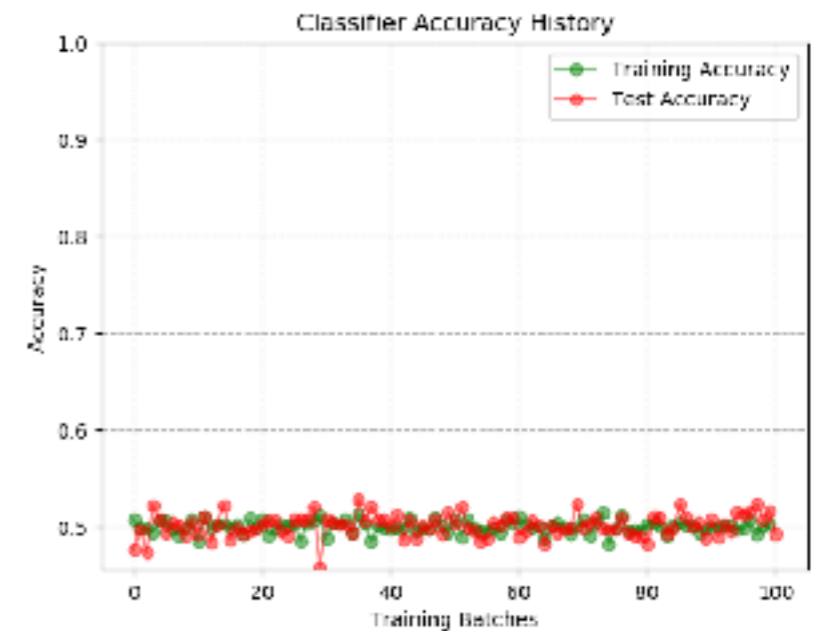
Learning rate had the greatest effect. Everything with a learning rate > 0.001 ended up with a final accuracy of 0.5.



4 hidden layers
256 hidden neurons
0.001 learning rate



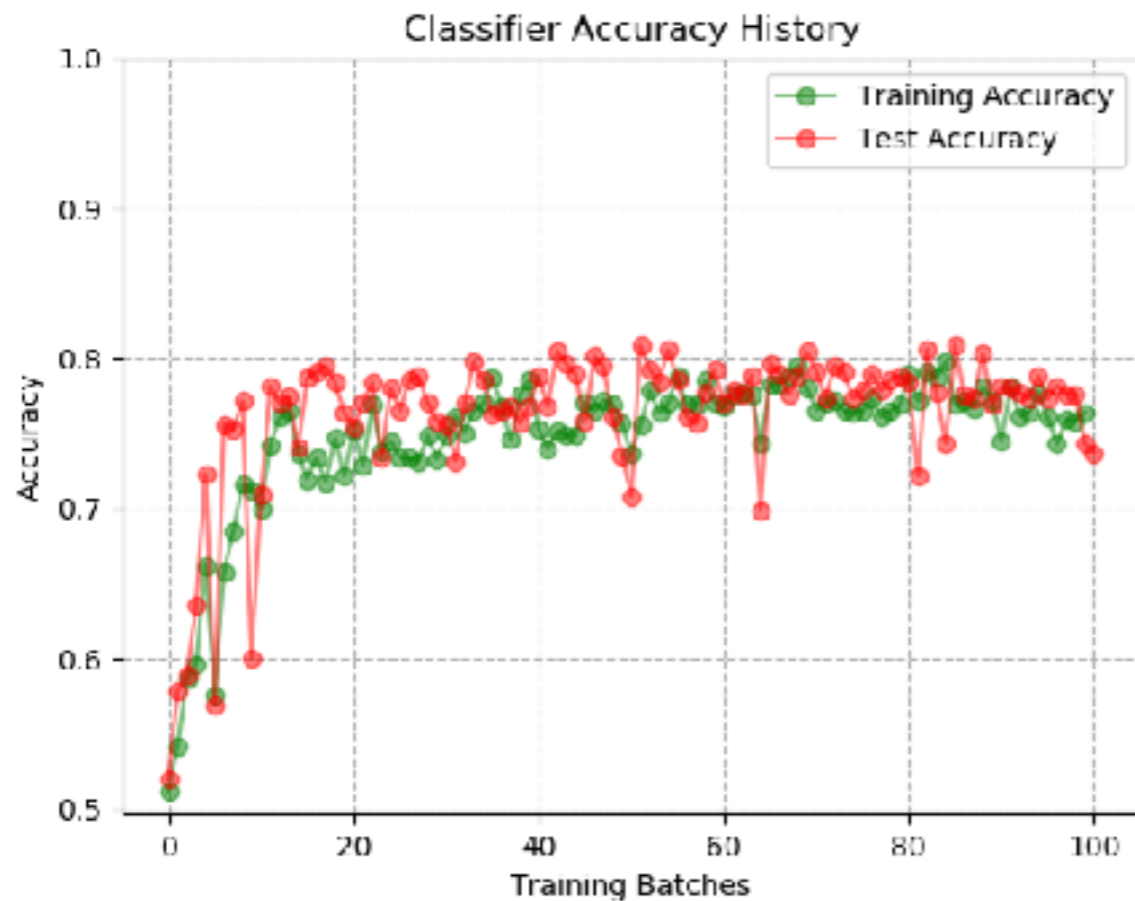
4 hidden layers
256 hidden neurons
0.005 learning rate



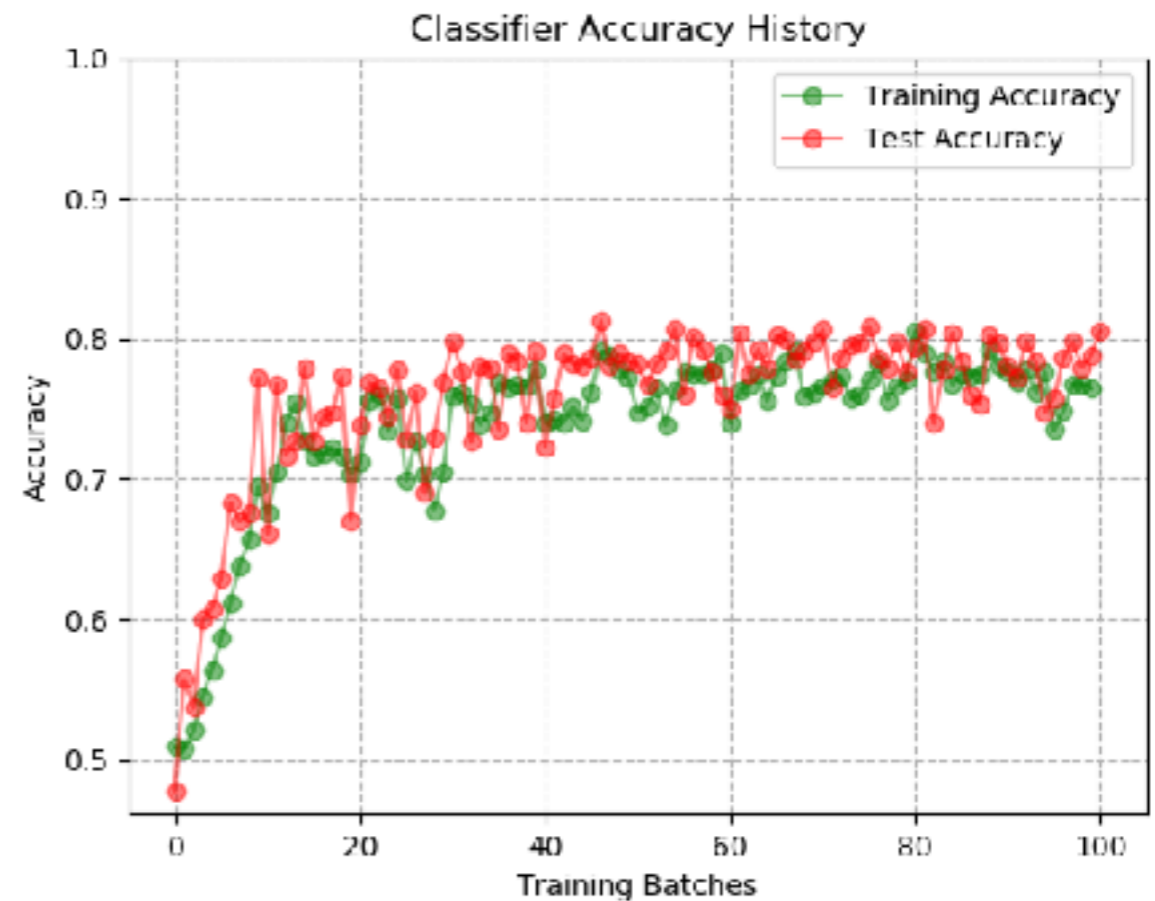
4 hidden layers
256 hidden neurons
0.01 learning rate

Hyperparameter Scan

Aside from that, it wasn't clear whether the number of hidden layers or neurons had any effect. All other hyperparameter points had final test accuracies of $77 \pm 3.5\%$, but with no real pattern.



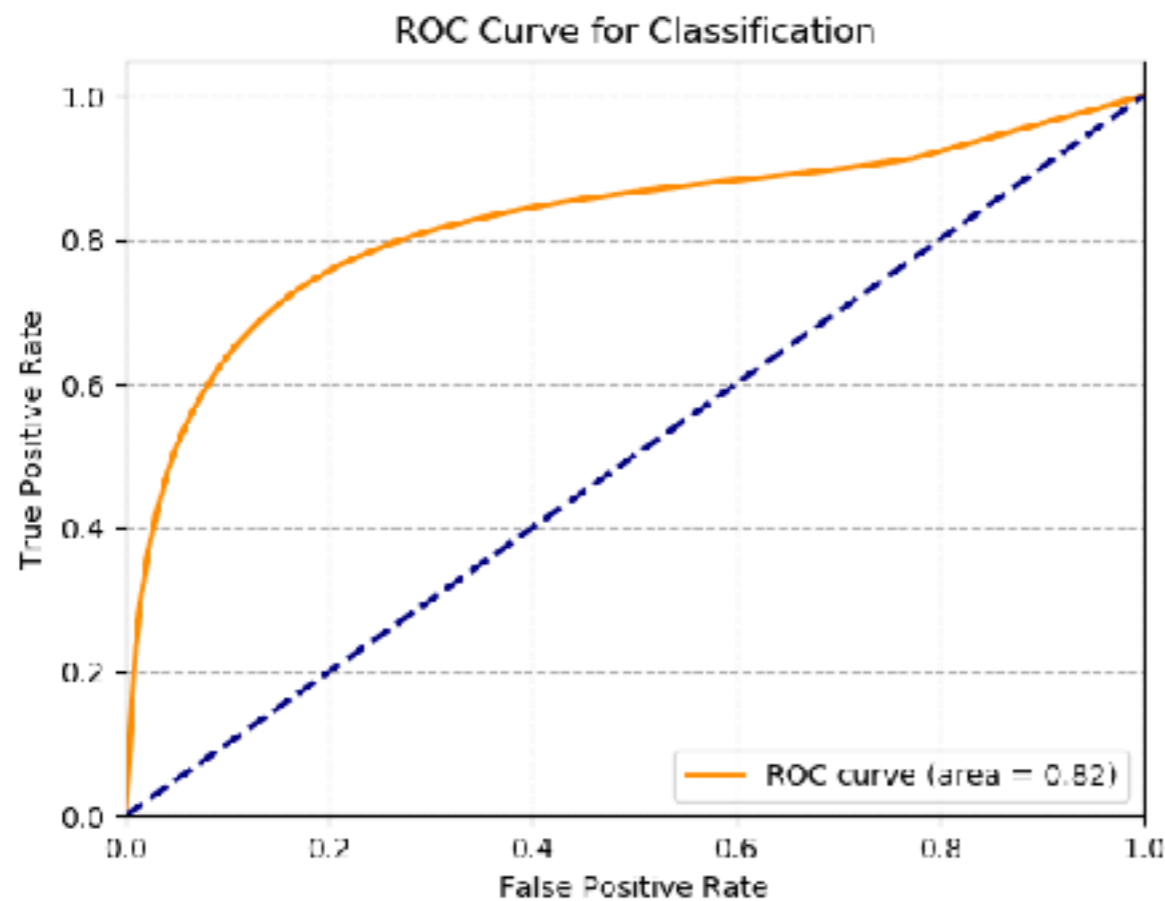
5 hidden layers
512 hidden neurons
test accuracy - 0.7346
(lowest)



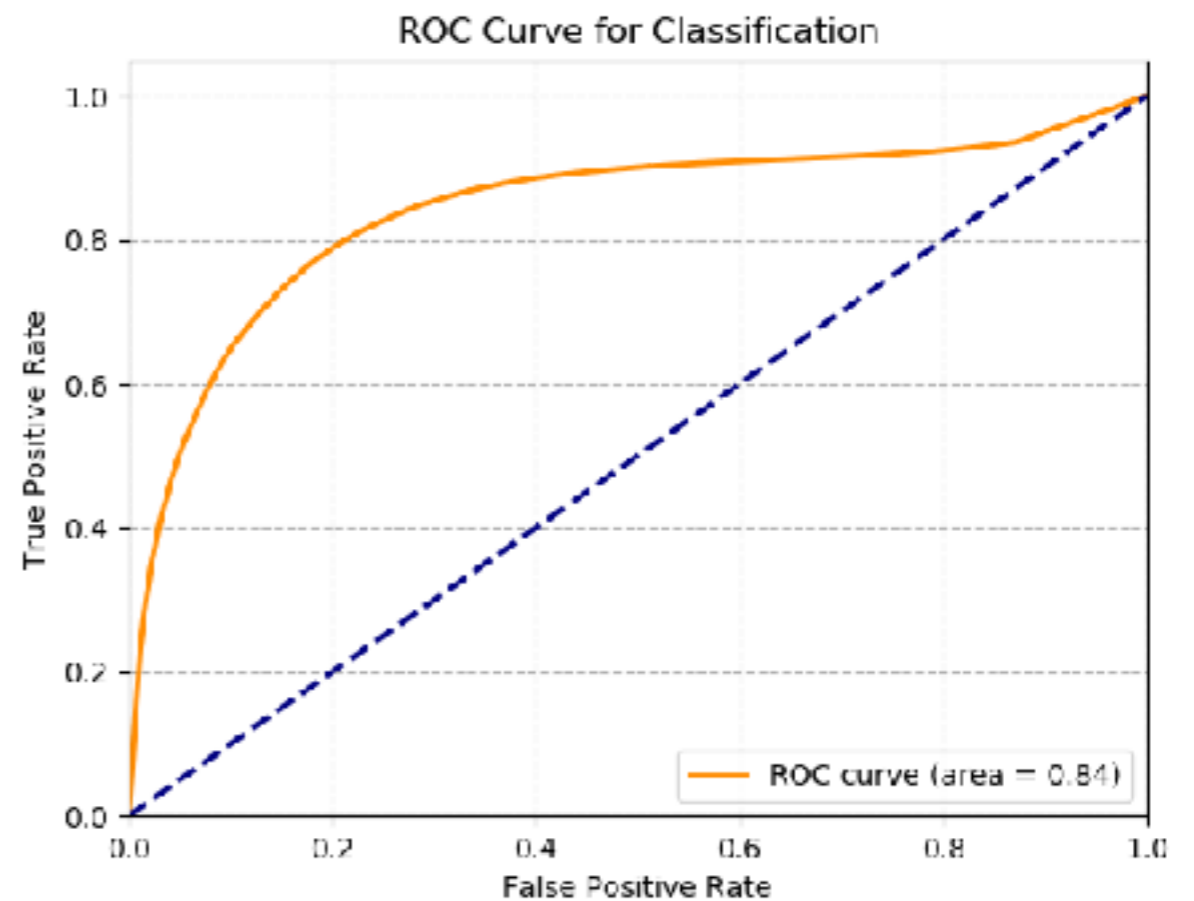
6 hidden layers
256 hidden neurons
test accuracy - 0.7937
(highest)

Hyperparameter Scan

Aside from that, it wasn't clear whether the number of hidden layers or neurons had any effect. All other hyperparameter points had final test accuracies of $77 \pm 3.5\%$, but with no real pattern.



5 hidden layers
512 hidden neurons
test accuracy - 0.7346
(lowest)



6 hidden layers
256 hidden neurons
test accuracy - 0.7937
(highest)

Suggested Further Scans

Learning rate = {0.0001, 0.0005, 0.001}

Hidden layers = {4, 8, 16}

Neurons per hidden layer = 256

Dropout rate = {0.1, 0.3, 0.5}

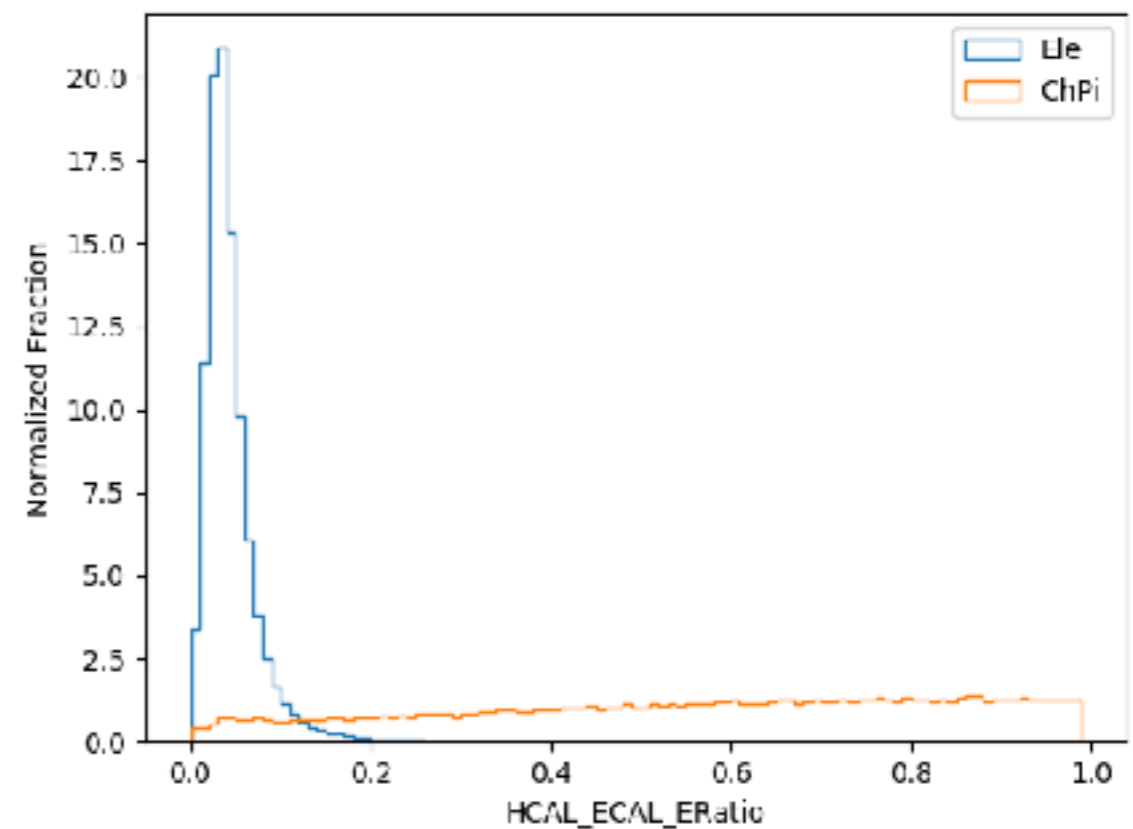
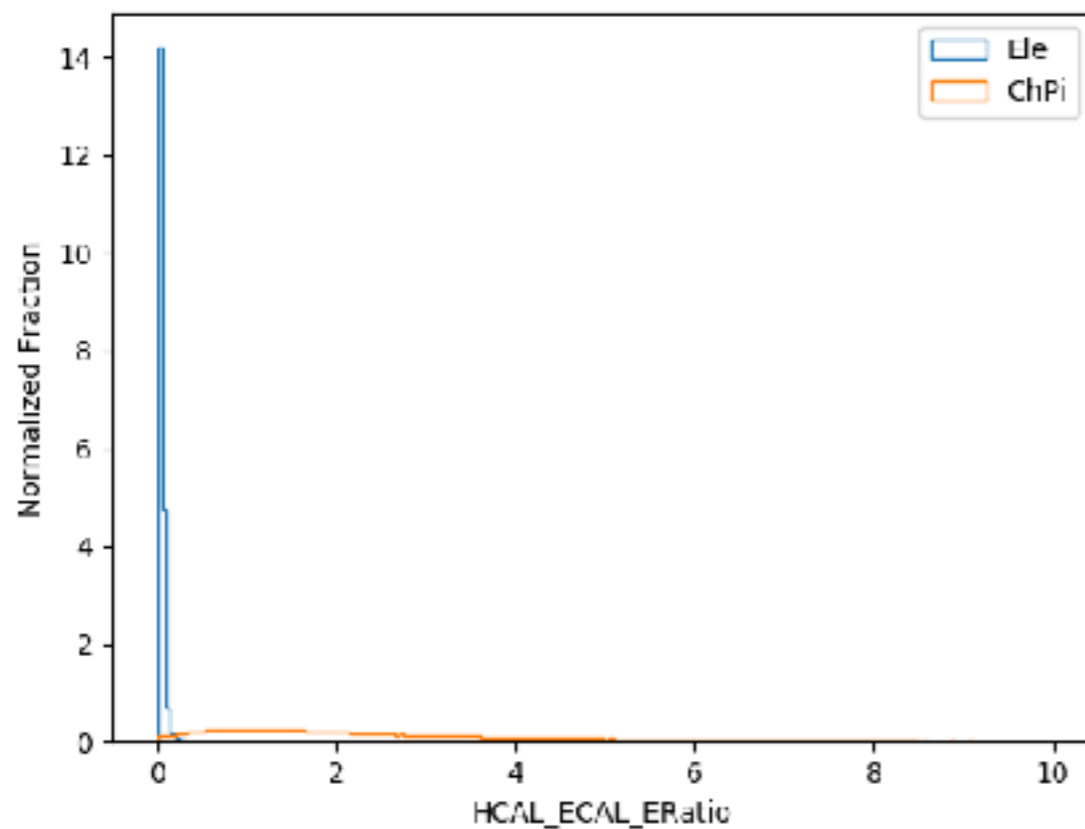
Dropout was at 0.5 for the previous scan.

Also perform a scan on GoogLeNet.

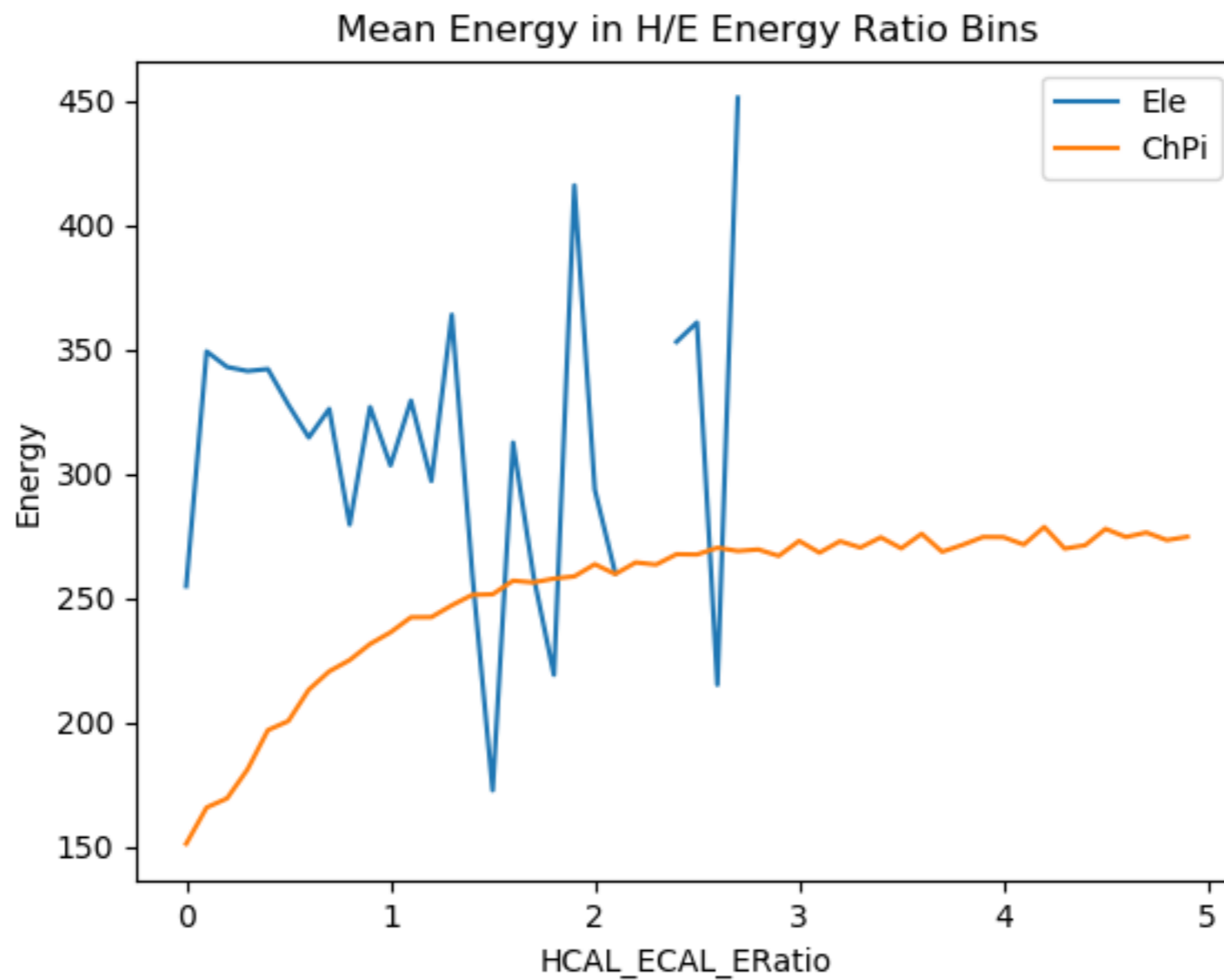
H/E Energy Ratio Cut

Need a better H/E energy ratio filter for ChPi and Electron task.

Below 0.1 seems like a good choice, though the big difference in H/E ratio shapes between electrons and charged pions may be a bit of a problem. About 6% ChPi events in this range.



H/E Energy Ratio Cut



ChPi Passing $H/E < 0.1$ Cut