Our Approach

Deep neural network with two kinds of predictions:

- Binary classification: muon vs. everything else
- Multi-class classification: muon, pion, kaon, proton, unmatched

Data Preparation & Network Input

Data Preparation

- Taking data from ROOT trees and converted it to pandas DataFrames
- Evenly sampled among classes from different MC generated processes
  - Each process produces kinematically different muons with different origins
- One-hot encoded the classes for categorical output
- Normalized data

Network Inputs

- Minimal model: variables that can ID a good muon (cut-based benchmark)
- Complex model: energy, position, track information (MVA benchmark)
- Custom: combination of the two
- Evaluate performance with accuracy and loss values and efficiency and ROC plots

Network Architecture & Training Stats

Architecture

- 4 hidden layers
- 128 neurons/layer
- ReLU activation
- Softmax activation on last layer
- Adam optimizer
- Categorical cross entropy loss

Training

- 35/65 test/training split
- 10/90 validation/training split
- 100 epochs
- 256 batch size

Results: Binary Classifier

Network statistics

Training

- Accuracy: 0.9593
- Loss: 0.1060

Validation

- Accuracy: 0.9299
- Loss: 0.2374

Test

- Accuracy: 0.9244
- Loss: 0.2728

Results: Binary Classifier

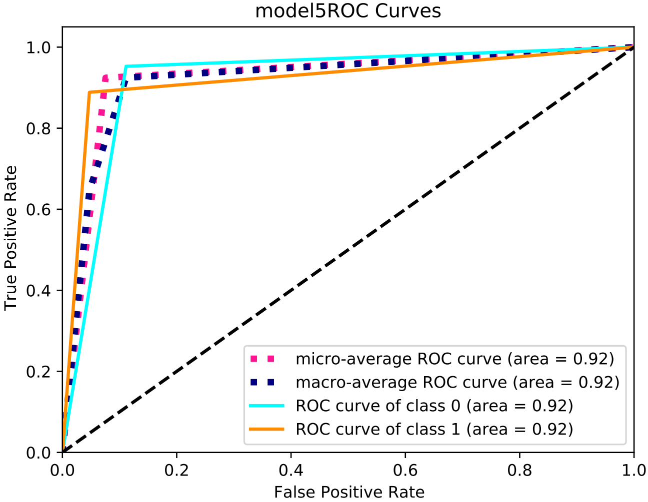
Muon Classification

- Efficiency: 95.274%
- Purity: 91.540%

Efficiency $= \frac{TP}{TP+FN}$

Purity $= \frac{TP}{TP+FP}$

Results: Binary Classifier



Results: Multiclass Classifier

Network statistics

Training

- Accuracy: 0.6854
- Loss: 0.7739

Validation

- Accuracy: 0.5323
- Loss: 1.2388

Test

- Accuracy: 0.5330
- Loss: 1.2864

Results: Multiclass Classifier

Muon Classification

- Efficiency: 84.032%
- Purity: 87.322%

Pion Classification

- Efficiency: 17.401%
- Purity: 41.398%

Kaon Classification

- Efficiency: 21.601%
- Purity: 33.932%

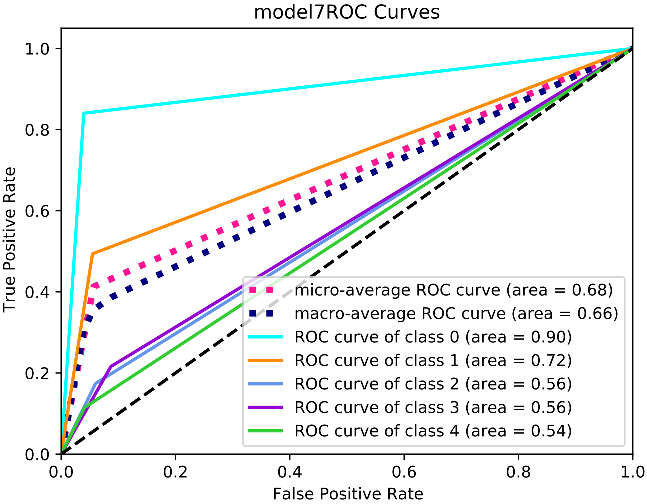
Proton Classification

- Efficiency: 11.496%
- Purity: 4.088%

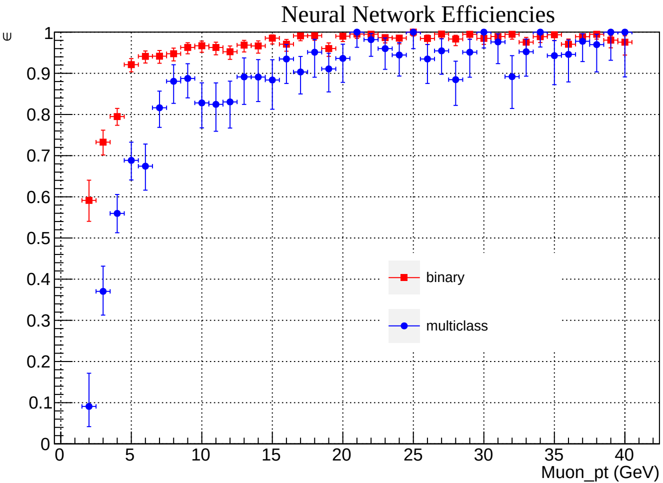
$$\text{Efficiency} = \frac{TP}{TP+FN}$$

$$\text{Purity} = \frac{TP}{TP+FP}$$

Results: Multiclass Classifier

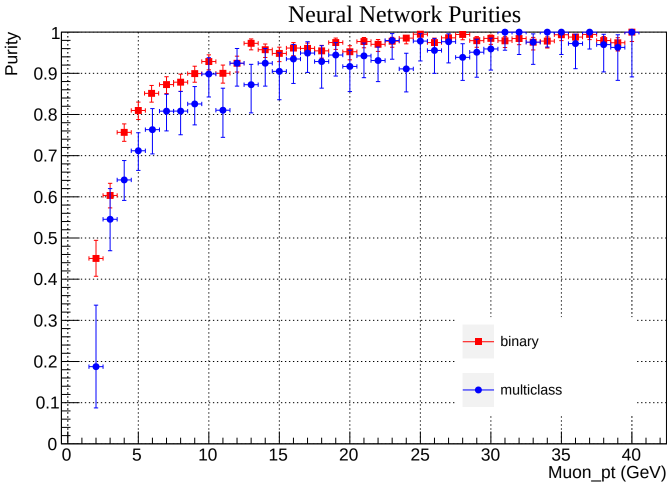


Results: Efficiencies



$\epsilon = \#$ true muons that pass ID$/\#$ true muons

Results: Purities



Purity = # true muons that pass ID/ #reconstructed muons

Benchmark Model

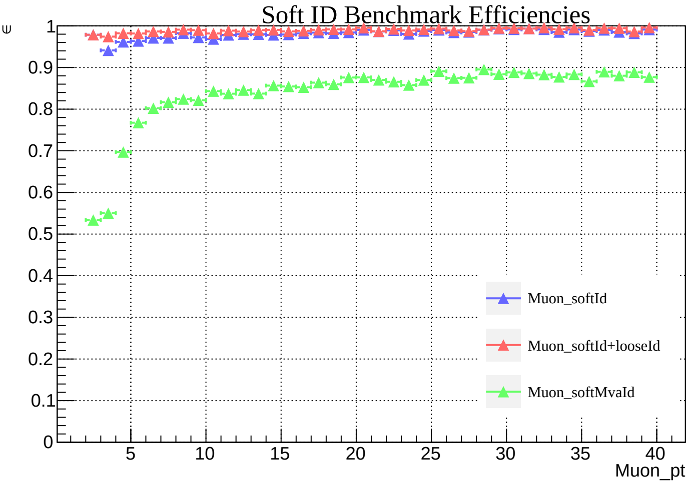
Current techniques to identify low momentum muons are a cut-based ID and a multi-variate analysis (MVA) that uses a gradient boosted regression forest

- Cut-based ID - uses cuts on a few key variables
- Soft MVA - gradient boosted regression forest

Evaluation Statistics

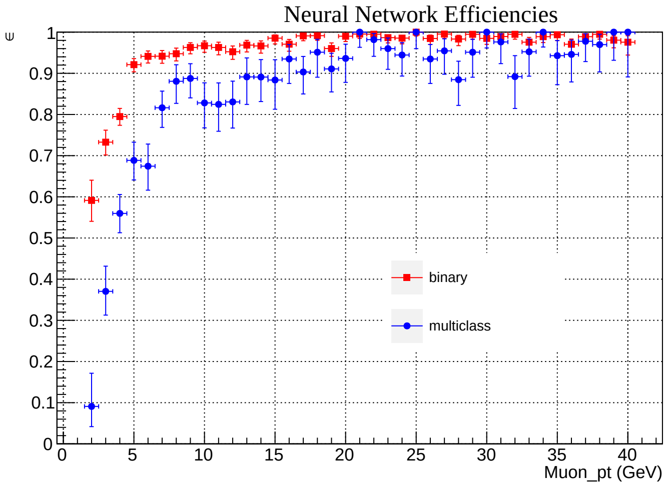
- Efficiency $\epsilon = \#$ true muons that pass ID/$\#$ true muons
- Purity $= \#$ true muons that pass ID/ $\#$reconstructed muons

Benchmark Model: Efficiency



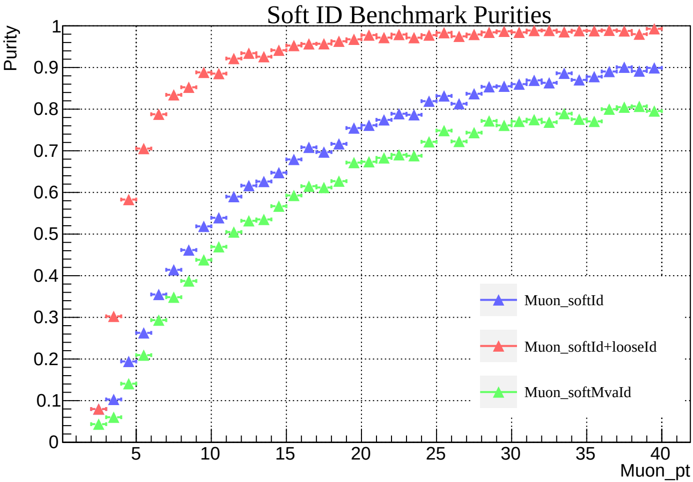
$\epsilon = \#$ true muons that pass ID/$\#$ true muons

Results: Efficiencies



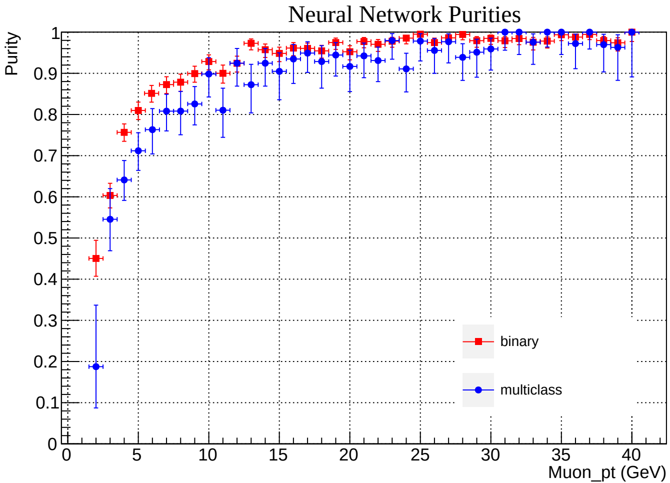
$\epsilon = \#$ true muons that pass ID$/\#$ true muons

Benchmark Model: Purity



Purity = # true muons that pass ID/ #reconstructed muons

Results: Purities



Purity = # true muons that pass ID/ #reconstructed muons

Summary

Goal: correctly classify low momentum muons from proton-proton collisions at the LHC with a neural network

Results

- kicked ass
- we did it lads
- binary classifier more successful at classifying electrons than multiclass output
  - higher efficiency and purity
- both models competitive with current classification techniques in efficiency
- both models have similar or higher purity than current classification techniques

Future work: integrate classifier into searches for Supersymmetry to find Dark Matter candidates

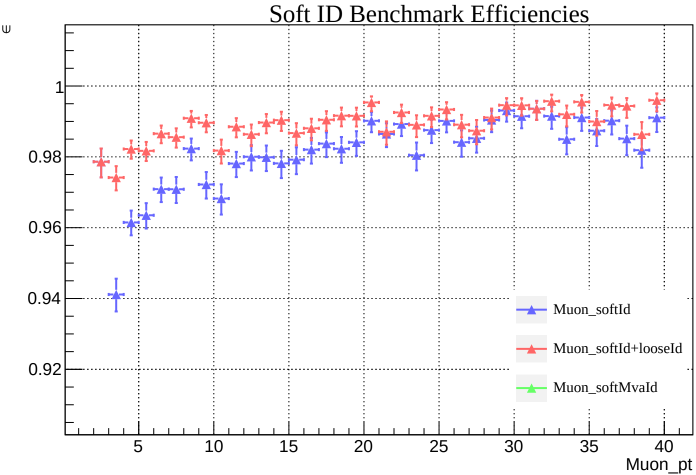
Project Repository: `https://github.com/Jphsx/KUSoftMVA`

Thank You!

Questions?

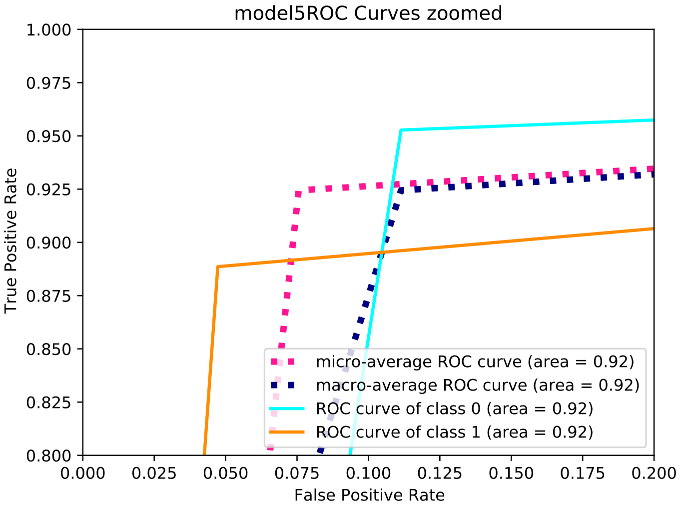
Backup

Benchmark Model Performances



$\epsilon = \#$ true muons that pass ID$/\#$ true muons

Results: Multiclass Classifier



Results: Multiclass Classifier