Using Deep Learning to Discriminate Between Quark and Gluon Jets

Tyler Evans, Florida State University

Supervisor: Yan Wang

September 5, 2018

Abstract

To deal with the enormous amounts of data produced in colliders, it is useful to automate processes for storing or analyzing data. Machine learning is used to automate some processes in physics already, e.g., detector triggering, object reconstruction, detector simulation. In this paper I go over how I used deep learning to allow a computer to learn to discriminate between heavy-quark jets and gluon jets. Automating jet classification in this way is useful for Higgs physics and Beyond the Standard Model physics. After analyzing my results, I see that the deep neural network performs well for jets coming from collisions with similar center of mass energy to the jets the system was trained on. I also find that jets in higher transverse momentum regions are easier for the computer to classify correctly.

Contents

1 Introduction 3

2 Background 3
2.1 Jet Physics . . . 3
2.2 Deep Learning . . . 3

3 Experimental Techniques 4
3.1 Event Generation . . . 4
3.2 Pre-processing Data . . . 6
3.3 Network Topology . . . 6
3.4 Training and Testing Procedure . . . 8

4 Results 8

5 Discussion 16
5.1 Future Research . . . 16

6 Conclusion 17

1 Introduction

Colliders in high energy physics typically create large amounts of data. The Large Hadron Collider (LHC) produces about 30 petabytes of data annually. Any means of automating the process of storing or analyzing such enormous amounts of data is useful. Deep learning, a machine learning method, is one way of automatically analyzing data. Machine learning is used in physics data processing already. It used for things such as detector triggers, object reconstruction, and detector simulation [1]. In my case, it is used for analysis of raw data from a detector. Deep learning has been shown to be successful at classification tasks, e.g. classifying hand written digits. Previous studies show that it can be used for classifying jets too [2]. This jet classification is important for Higgs physics and Beyond the Standard Model (BSM) physics [3]. In my experiment, I study whether a neural network can effectively discriminate between heavy-quark and gluon jets. The goal of my experiment is to see whether deep learning can be applied at a future electron-positron collider like the International Linear Collider (ILC).

2 Background

2.1 Jet Physics

Jets are narrow cones of hadrons and other particles produced by the hadronization of a quark or gluon in a particle physics experiment. It is important to know the source of a jet so that one can understand what occurred in a collision [3]. Quark jets and gluon jets have a different color charge. The color charge for quarks (C_F) is $\frac{4}{3}$ and the color charge for gluons (C_A) is 3. See [3] for more detail on the differences between quark and gluon jet substructure.

2.2 Deep Learning

Machine learning is the technique of programming computers so they can learn from data. At the core of machine learning is artificial neural networks (ANNs). ANNs are a tool in machine learning inspired by the brain's architecture. ANNs give computers the ability to learn without being explicitly programmed. With a neural network and lots of data, a computer can "learn" and become better at a given task. Discriminating between heavy-quark and gluon jets is a classification task. A dense layer is a layer that is fully connected to the next layer, meaning that every node in one layer connects to every node in the other. Deep learning is a machine learning method where the neural network learns features itself. It is useful for raw data like pictures or text. One type of deep neural network is a convolutional neural network (CNN). CNNs are based off of the brain's visual cortex. It is useful for tasks that involve visual perception. Since my experiment uses jet pictures as input, I use CNNs.

An introduction to machine learning methods and definitions can be found in [4].

3 Experimental Techniques

In this section, I'll describe how I simulated collision events using an event generator and how I pre-processed the data from these events for use on the neural network. Then I'll show the architecture of my neural network and explain how I trained and tested it.

3.1 Event Generation

To generate events, I used ILCsoft. The Feynman diagrams for the simulated processes are shown in Figure 1.

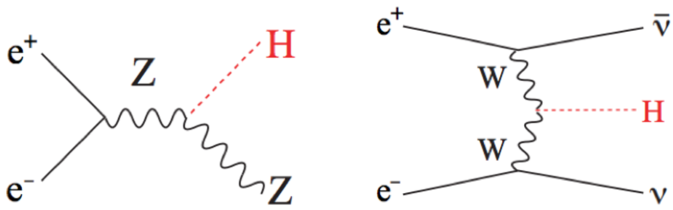


Figure 1: On the left, is the main production line. An electron-positron pair annihilate, producing a Z boson which becomes a Z boson and Higgs boson. The Higgs will then decay into either a pair of charm quarks, bottom quarks, or gluons. This is the main production line. On the right, is the other production line.

From generated events, I receive the identity, charge, pseudorapidity (η), ϕ angle, and transverse momentum (pT) for each particle in an event. Using FastJet with kt algorithm and specifying two jets, I then retrieve each jet's constituents, η, ϕ, and pT in the detector.
In order for the data to be useful input for a neural network, it needs to be of consistent size. So I turn each jet with varying amounts of particles into a picture of consistent size. I do this by creating 2D histograms with η and ϕ as the axes. Then multiple histograms are filled, each representing a different "color". In my project, I used the same color scheme as [2]. Red represents the transverse momenta of neutral particles. Blue represents the transverse momenta of charged particles. Green represents the charged particle multiplicity.

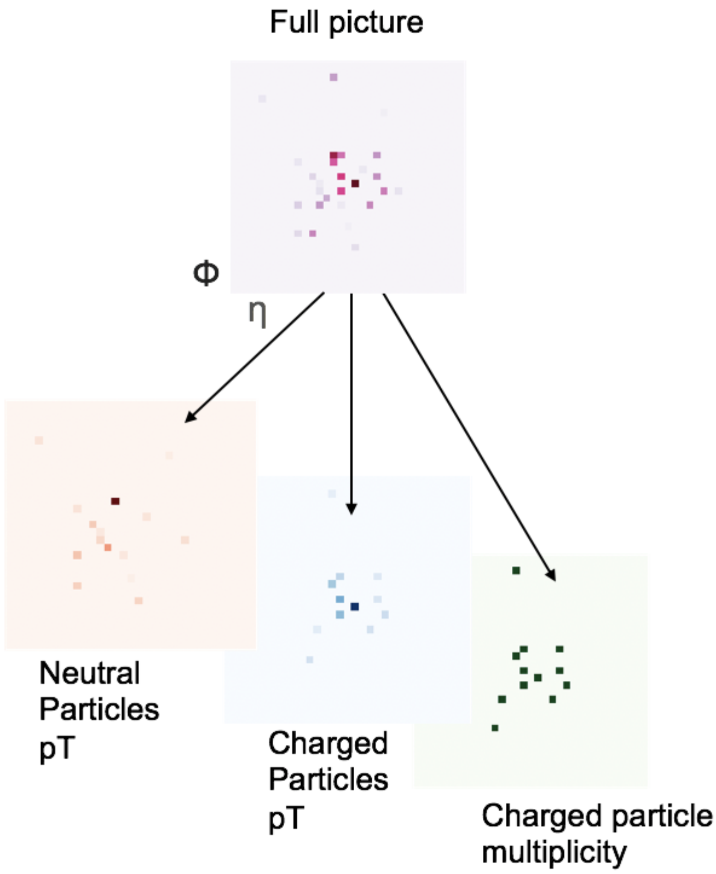


Figure 2: This is an example of a jet picture. Below the full picture is the three different colors that make up the picture. The three different colors represent neutral particle transverse momenta, charged particle transverse momenta, and charged particle multiplicity.

3.2 Pre-processing Data

In [2], they used the following steps for pre-processing: center the jet image, crop the image to a 33 x 33 pixel region, normalize the pixel intensities, zero-center (subtract the mean so that the mean is 0), and standardize (divide by the standard deviation so that the standard deviation is 1). I used these same steps for pre-processing.
Last, I separated the jets into pT ranges:

- 0-30 GeV/c
- 40-50 GeV/c
- 60-70 GeV/c
- 800-100 GeV/c
- 110-160 GeV/c

3.3 Network Topology

I used the network topology from [2], shown in Figure 3. It consists of an input layer for the jet image, three convolutional/max-pooling layer sets, a dense layer, and an output layer. In addition, there are dropout layers after each convolutional/max-pooling layer set and after the dense layer. I initialized the model weights with He-uniform initialization.

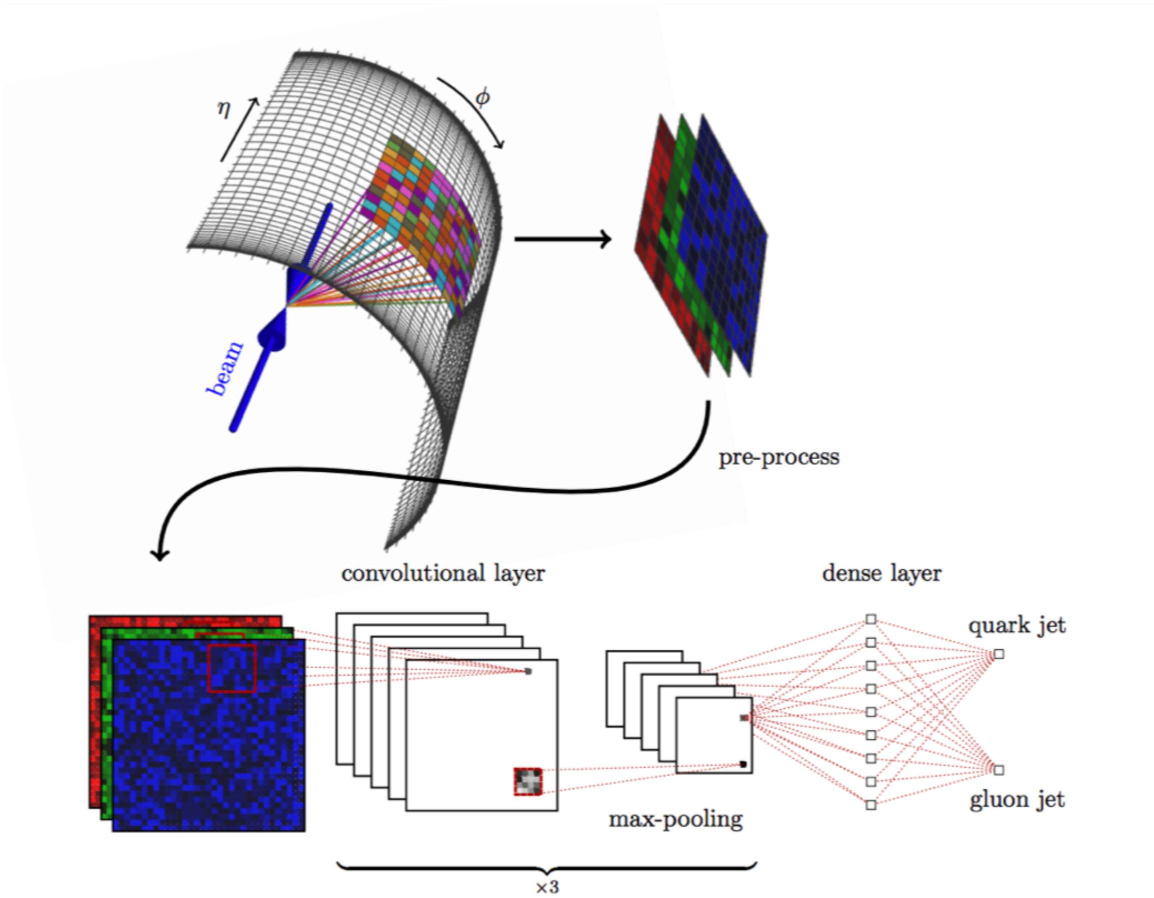


Figure 3: [2] This depicts the process for creating data and using it for the neural network. I make "colored" pictures using the reconstruction data from the simulated collision and detector hits. These are then sent through three convolutional/max-pooling layer sets, and then into a dense layer which connects to the output.

Each CNN has 64 filters and a kernel size of (8,8) for the first one, (4,4) for the second and third. Each ma-pooling layer has a pool size of (2,2), a stride of (2,2), and 'same' padding. The first dense layer has 128 nodes and the output layer has one node. I also apply spatial dropout of .5 after every convolutional layer and dropout of .5 after the first dense layer. Every layer has a relu activation function except for the output layer which has a sigmoid activation function. I created this network using the Keras API for Tensorflow.

3.4 Training and Testing Procedure

I trained the network using the Adam algorithm and the binary cross-entropy loss function with a batch size of 256 over a maximum of 100 epochs and an early stopping patience of 10 epochs. The learning rate was set to 0.001. The amount of data used for training and testing depended on the pT region. The amount of training data, validation data, and testing data used in each pT region is given in Table 1 in the results section. I trained the network on two Tesla K20x GPUs on the maxwell cluster at DESY.
After training, I tested the neural network on the test set to obtain accuracy and to generate a ROC plot. For the 40-50 GeV/c pT region, I also tested the neural network on the 250 GeV and 500 GeV center of mass energy collision events with the same pT region.

4 Results

pT region (GeV/c)	**train events**	**validation events**	**test events**
0-30	20,840	2,316	2,573
40-50	18,951	2,106	2,340
60-70	14,625	1,625	1,806
80-100	15,180	1,687	1,874
110-160	9,763	1,085	1,205

Table 1: The amount of events used for training, validation, and testing for each pT region. Of the total amounts of data, 90% was used for training, 10% for testing. And 10% of the training data was designated for validation.

pT region (GeV/c)	**training time** (s)	**epochs**	**AUC**	**Accuracy**
0-30	328.36	41	0.62672	0.59456
40-50	505.99	71	0.68486	0.63818
60-70	270.27	47	0.70142	0.64330
80-100	285.2	49	0.70765	0.65205
110-160	229.64	62	0.76422	0.68664

Table 2: The time it took to train each neural network, the amount of epochs reached before early stopping or max epoch, the AUC score, and the accuracy score for each pT region. One can see that the networks never reached the 100th epoch (max epoch). This is because of early stopping which stopped the neural network from training anymore after it was unable to lower the loss on the validation data for 10 consecutive epochs. The average training time was 324 seconds (¿5 minutes). The average amount of epochs is 54. There is a positive correlation between pT region and AUC score. This shows that jets with higher pT are easier to classify correctly. This positive correlation occurs between pT region and accuracy too.

$\sqrt{\mathbf{s}}$ (GeV)	**pT region** (GeV/c)	**AUC**	**Accuracy**
250	60-70	0.52540	0.40881
500	60-70	0.50871	0.34130

Table 3: The AUC score and accuracy score for the 60-70 GeV/c pT region for center of mass energies of 250 GeV and 500 GeV. We see that, even though these jets are in the same pT region, the network classifies them poorly. This means that a trained neural network will only work well for events with the same center of mass energy as the events is was trained on. The below 50% accuracy occurs because there is not an even amount of quark jets as gluon jets in the test data.

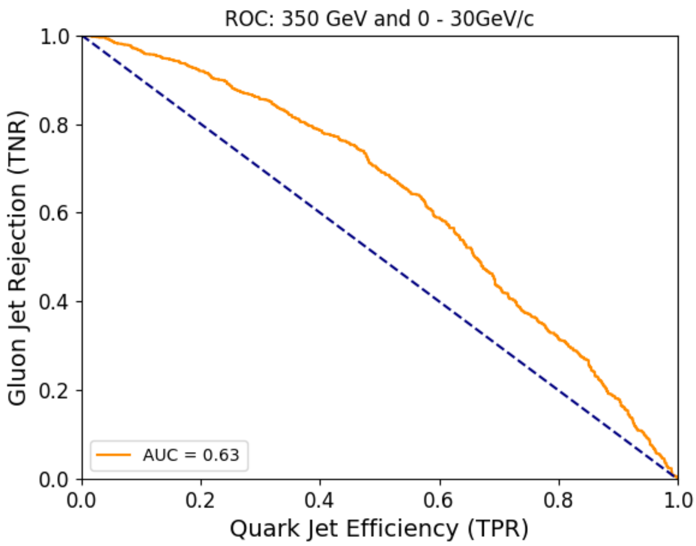


Figure 4: ROC curve for classifying jets with pT between 0 and 30 GeV/c, and a center of mass energy of 350 GeV. On the vertical axis is the gluon jet rejection (true negative rate). This displays the ability of the neural network to correctly classify a gluon jet as gluon jet. The horizontal axis is the quark jet efficiency (true positive rate). This represents the neural networks ability to correctly classify quark jets as quark jets. A higher quark jet efficiency and gluon jet rejection means that the neural network is performing better. Thus, the better the neural network, the closer the orange curve is to the upper right corner. This ROC curve is the farthest away from the upper right corner. This means that the neural network for the lowest pT region performed the worst.

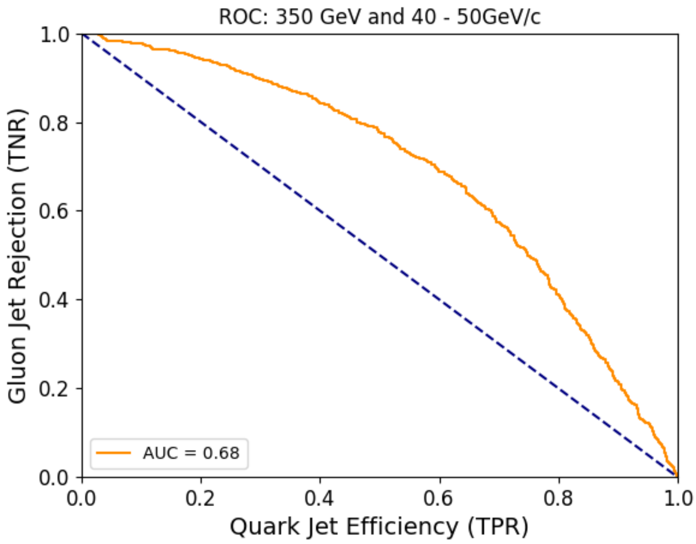


Figure 5: ROC curve for classifying jets with pT between 40 and 50 GeV/c, and a center of mass energy of 350 GeV.

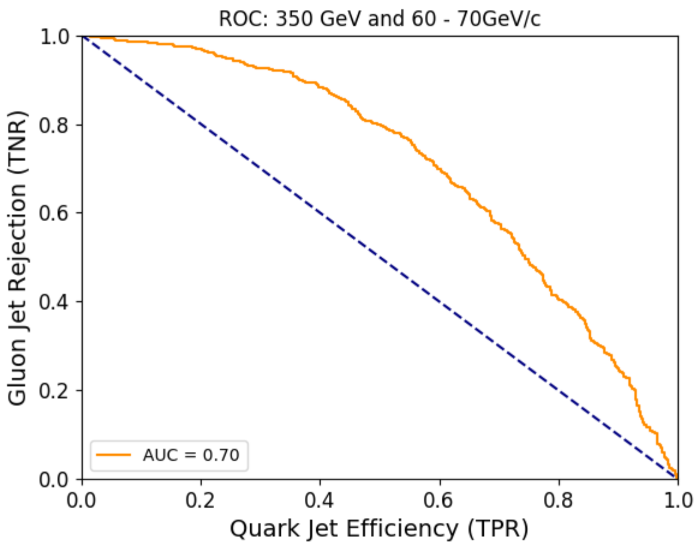


Figure 6: ROC curve for classifying jets with pT between 60 and 70 GeV/c, and a center of mass energy of 350 GeV.

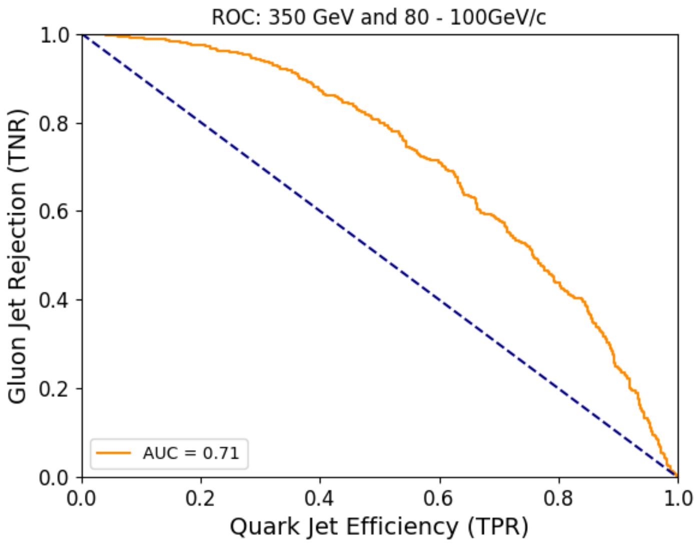


Figure 7: ROC curve for classifying jets with pT between 80 and 100 GeV/c, and a center of mass energy of 350 GeV.

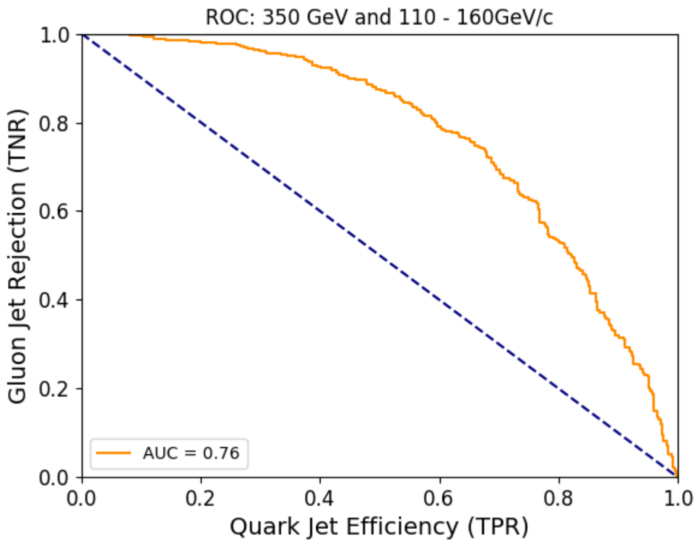


Figure 8: ROC curve for classifying jets with pT between 110 and 1600 GeV/c, and a center of mass energy of 350 GeV. With an AUC of 0.76, this neural network performed the best. This is also the highest pT region, which emphasizes the positive correlation between a jets pT and it's ability to classified correctly.

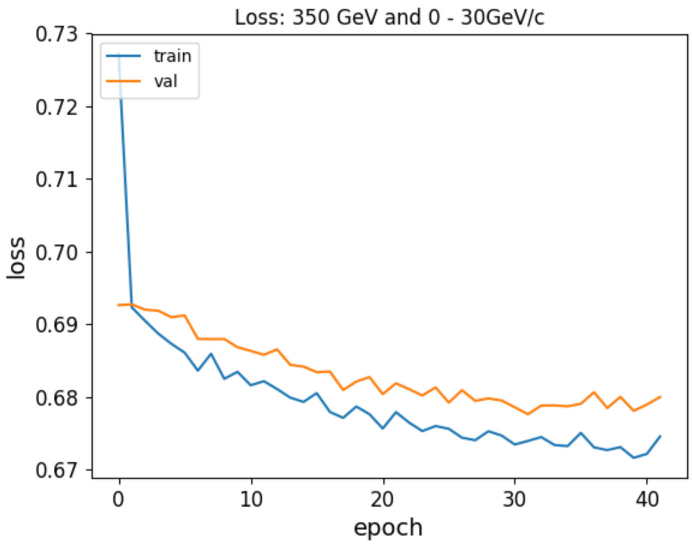


Figure 9: Loss calculated on the training and validation set at each epoch for the network learning on jets with pT between 0 and 30 GeV/c, and a center of mass energy of 350 GeV. Notice that the tick marks on the vertical axis only differ by .01. This means that the plateau we see is very flat. This means that the neural network learned very slowly after the first epoch.

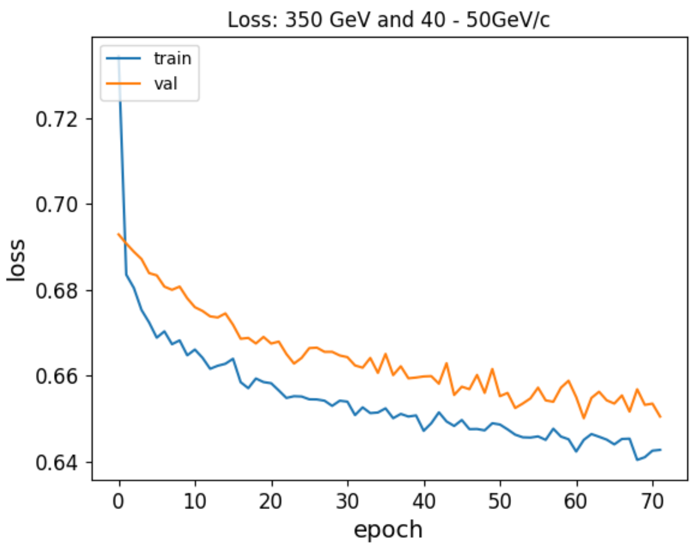


Figure 10: Loss calculated on the training and validation set at each epoch for the network learning on jets with pT between 40 and 50 GeV/c, and a center of mass energy of 350 GeV. There is a plateau after the first epoch.

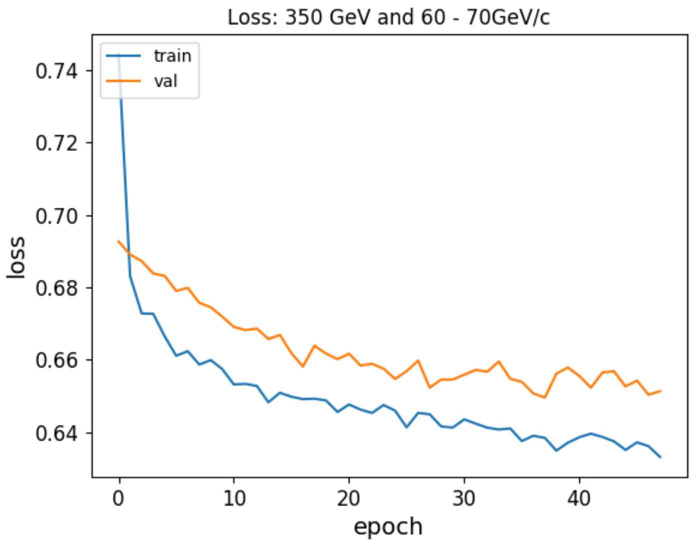


Figure 11: Loss calculated on the training and validation set at each epoch for the network learning on jets with pT between 60 and 70 GeV/c, and a center of mass energy of 350 GeV. There is a plateau after the first couple of epochs.

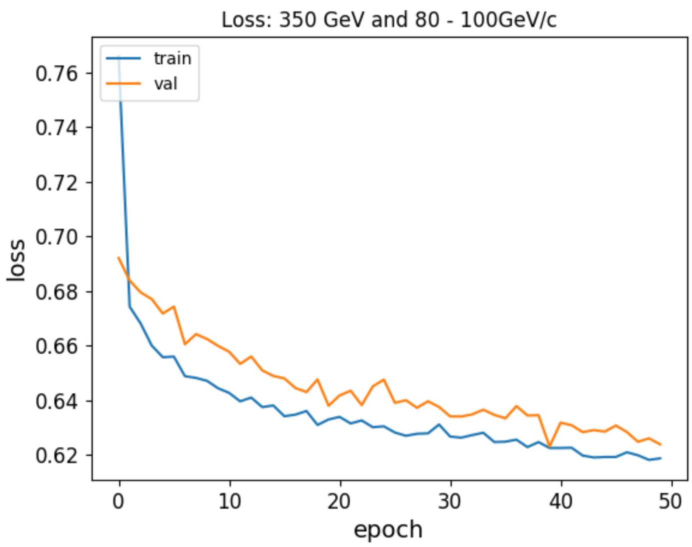


Figure 12: Loss calculated on the training and validation set at each epoch for the network learning on jets with pT between 80 and 100 GeV/c, and a center of mass energy of 350 GeV. There is a plateau after the first couple of epochs.

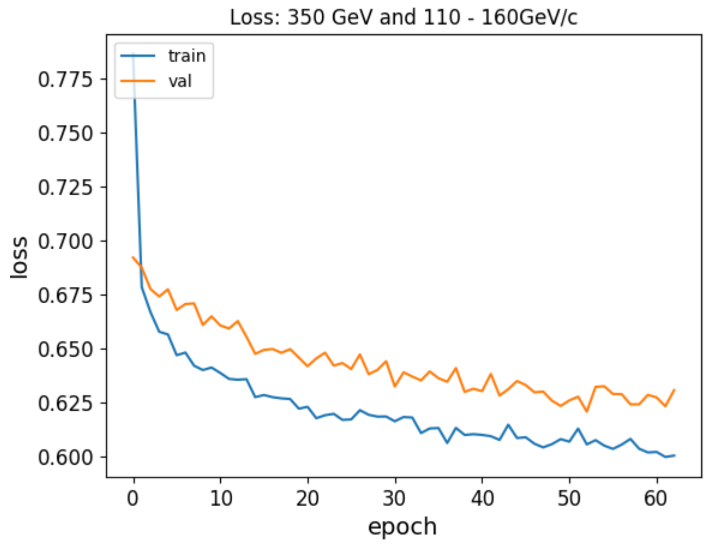
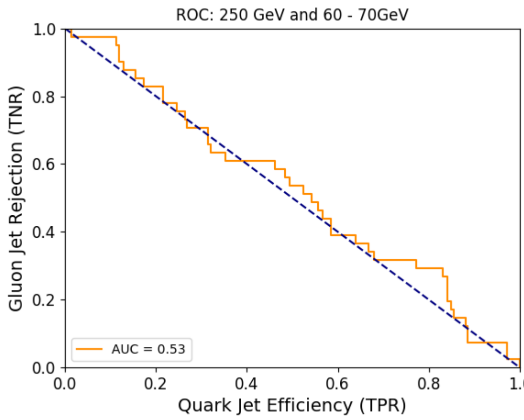
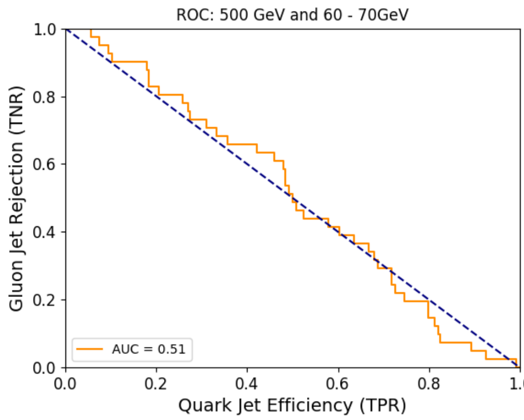


Figure 13: Loss calculated on the training and validation set at each epoch for the network learning on jets with pT between 110 and 160 GeV/c, and a center of mass energy of 350 GeV. Again, we see this flattening out in the plot.



(a) Center of mass energy: 250 GeV



(b) Center of mass energy: 500 GeV

Figure 14: ROC curves obtained by testing the network trained on jets with pT between 60 and 70 GeV/c from collisions with center of mass energy at 350 GeV on jets from the same pT region but from collisions with center of mass energy at 250 GeV or 500 GeV. We see that the orange curve hugs the baseline (blue dashed line). This means the neural network has not learned to classify these jets.

5 Discussion

As shown by Table 2 in the results section, my system can allow a computer to learn to discriminate between heavy-quark and gluon jets. Table 2 shows that there is a positive correlation between jet pT and the jets ability to be classified correctly. One problem is that this neural network does poorly to classify jets in the same pT region but from collisions of different center of mass energy, shown in Figure 14. Thus, one would need to train a whole new network on the jets from the certain center of mass energy.

5.1 Future Research

Here, I will discuss possible steps to take next in this area of research. One step is to improve recursive neural networks (RecNNs). In a study by T. Cheng [5], they used a RecNN to discriminate between light-quark jets and gluon jets at the same energies as in [2].

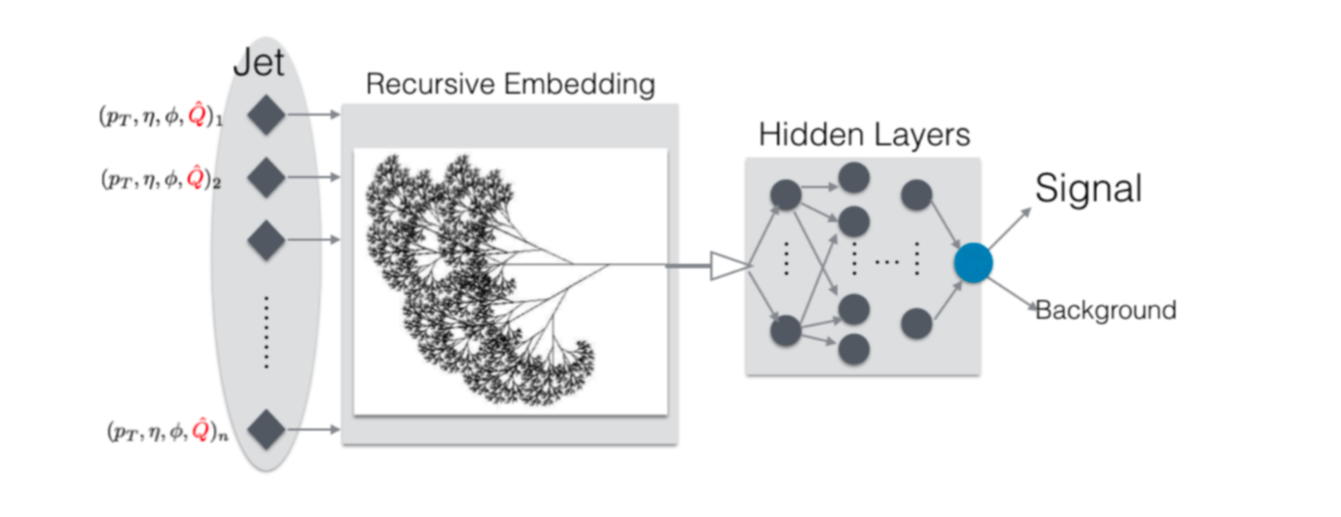


Figure 15: [5] The recursive neural network is able to take all of the information about particles in a jet by putting it through a recursive embedding before sending it through a regular deep neural network made of many hidden, fully-connected layers.

Even though the recursive network uses all of the jet information so that there is no information loss, it did not score better than the network in [2] where information is lost by making pictures. In machine learning, the quality of data is important. Turning the data into pictures degrades the data quality. So a system that doesn't degrade the data in such a way should be able to out perform the one with worse data. Thus, RecNNs need to be improved so that they can compete with CNNs and, in the case of jet classification, outperform CNNs because of better data quality.
Another step is to train and test my neural network using light-quark jets and gluon jets with 350 GeV center of mass energy and the same pT ranges. Then I can compare

these results to the results from this experiment where I used heavy-quark jets instead of light-quark jets.
Finally, a future step is to search for better neural network parameters and functions. For example, one could try applying "ReduceLROnPlateau" which reduces the learning rate when a plateau in loss is detected. Or one could try different optimizers or initializers.

6 Conclusion

During my stay at DESY, I created a deep learning system that learns how to discriminate between heavy quark jets and gluon jets. It works well for jets coming from collisions with similar center of mass energy to the jets the system was trained on. And jets in higher transverse momentum regions are easier for the computer to classify correctly. This jet discrimination is important for Higgs physics and Beyond the Standard Model physics [3]. Jet discrimination should also be useful for a future electron-positron collider like the ILC.

References

[1] K. Albertsson et al., *Machine Learning in High Energy Physics Community White Paper*, 2018, [arXiv:1807.02876v1]

[2] P. Komiske et al., *Deep learning in color: towards automated quark/gluon jet discrimination*, 2017 *JINST* **01** 110 [arXiv:1612.01551v2]

[3] J. Gallicchio et al., *Quark and Gluon Jet Substructure*, 2013, [arXiv:1211.7038v2]

[4] Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st Edition, O'Reilly Media, 2017.

[5] T. Cheng, *Recursive Neural Networks in Quark/Gluon Tagging*, 2018, [arXiv:1711.02633v2]