Project 1.1: Software 1.0 Versus Software 2.0

Srivenkata Krishnan Sesharamanujam
CSE574 Introduction to Machine Learning
UB Person no: 50288730
srivenka@buffalo.edu

1 Introduction

The following report is for the fizz buzz program implemented using the machine learning approach. The report describes the performance of fizz buzz program on how the choice of hyper-parameters affects performance, how the model works with different network settings such as different number of layers in the network, different number of nodes in each layer, different optimization methods, different dropout rates, different activation functions and accuracy measures. The rest of the report is organized as follows: Section 2 outlines the accuracy measures while we discuss what hyper parameter is and how does it affect the model in section 3. In the penultimate section, we compare the graph outputs of different measures discussed in the previous sections and conclude in section 4.

2 Accuracy Measures

Evaluating the machine learning algorithm is an essential part of any project. The model may give you satisfying results when evaluated using a metric say accuracy but may give poor results when evaluated against other metrics such as logarithmic loss or any other such metric. Most of the times we use classification accuracy to measure the performance of our model, however it is not enough to truly judge our model. The different types of common metrics used are

- Classification Accuracy.
- Logarithmic Loss.
- Area Under ROC Curve.
- Confusion Matrix.
- Mean Absolute Error.
- Mean Squared Error.

2.1 Classification Accuracy

Classification accuracy is ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{NoofCorrectPredictions}{NoofTotalPredictions} \tag{1}$$

2.2 Logarithmic Loss

Logarithmic loss (or logloss) is a performance metric for evaluating the predictions of probabilities of membership to a given class.

Suppose, there are N samples belonging to M classes, then the Log Loss is calculated as below :

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * log(p_{ij}) \tag{2}$$

where,

y-ij == whether sample i belongs to class j or not.

p-ij == the probability of sample i belonging to class j.

2.3 Area under ROC Curve

Area under ROC Curve is a performance metric for binary classification problems.The AUC represents a models ability to discriminate between positive and negative classes. An area of 1.0 represents a model that made all predictions perfectly. An area of 0.5 represents a model as good as random. ROC can be broken down into sensitivity and specificity.

- Sensitivity is the true positive rate also called the recall. It is the number instances from the positive (first) class that actually predicted correctly.
- Specificity is also called the true negative rate. Is the number of instances from the negative class (second) class that were actually predicted correctly.

2.4 Confusion Matrix

The confusion matrix is a handy presentation of the accuracy of a model with two or more classes. The table presents predictions on the x-axis and accuracy outcomes on the y-axis. The cells of the table are the number of predictions made by a machine learning algorithm.

2.5 Mean Absolute error

Mean Absolute Error is the average of the difference between the Original Values and the Predicted Values. It gives us the measure of how far the predictions were from the actual output.

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1} N|y_j - y'_j| \tag{3}$$

2.6 Mean Squared Error

Mean Squared Error(MSE) is quite similar to Mean Absolute Error, the only difference being that MSE takes the average of the square of the difference between the original values and the predicted values. The advantage of MSE being that it is easier to compute the gradient, whereas Mean Absolute Error requires complicated linear programming tools to compute the gradient.

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1} N(y_j - y'_j)^2 \tag{4}$$

3 Hyper parameters

Hyper parameters are the variables which determines the network structure(Example: Number of Hidden Units) and the variables which determine how the network is trained(Example: Learning Rate). Hyper parameters are tuning parameters of a machine learning algorithm. In any machine learning algorithm, these parameters need to be initialized before training a model. Hyper parameters can be classified as,

- Related to Network structure.
- Related to training algorithm.

3.1 Related to Network Structure

The hyper parameters that fall into these are

- No of hidden layers/units.
- Dropout.
- Network Weight Initialization.
- Activation Function.

3.1.1 No of Hidden Layers/Units

Hidden layers are the layers between input layer and output layer.

3.1.2 Dropout

Dropout is regularization technique to avoid over fitting, thus increasing the generalizing power.

3.1.3 Network Weight Initialization

Based on the activation function, use different weights.

3.1.4 Activation Function

Activation functions are used to introduce non linearity to models, which allows deep learning models to learn nonlinear prediction boundaries.

Most Popular types of activation functions are,

- Sigmoid or Logistic
- tanh - Hyperbolic Tangent
- ReLu - Rectified Linear Units

Sigmoid Activation Function It is defined as,

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

Hyperbolic tangent - tanh It is defined as,

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{2x}} \tag{6}$$

ReLu - Rectified Linear Units

$$R(x) = max(0, x) \tag{7}$$

if x less than 0, R(x) = 0,

x greater than 0, R(x) = x

3.2 Related to Training Algorithm

- Learning Rate.
- Momentum.
- Number of epochs.
- Batch size.

3.2.1 Learning Rate

The learning rate defines how quickly a network updates its parameters. Low learning rate slows down the learning process but converges smoothly. Larger learning rate speeds up the learning but may not converge.

3.2.2 Momentum

Momentum helps to know the direction of the next step with the knowledge of the previous steps. It helps to prevent oscillations. A typical choice of momentum is between 0.5 to 0.9.

3.2.3 Number of epochs

Number of epochs is the number of times the whole training data is shown to the network while training.

3.2.4 Batch size

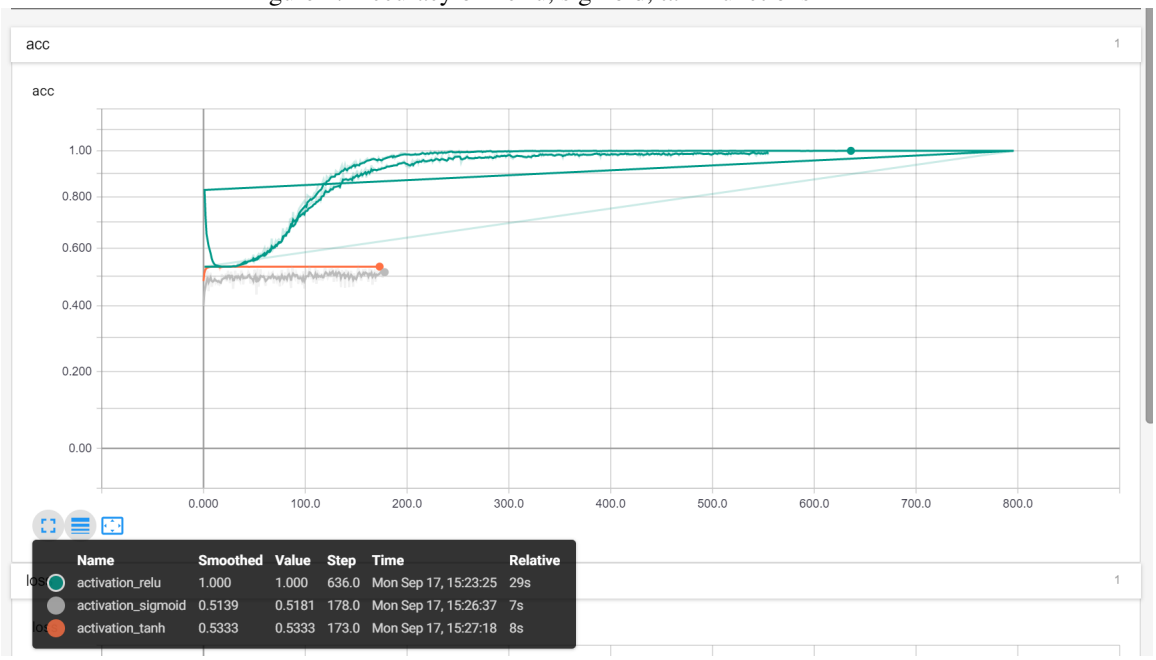
Mini batch size is the number of sub samples given to the network after which parameter update happens.A good default for batch size might be 32.

4 Performance of model used in the Fizz buzz program

4.1 Activation Functions

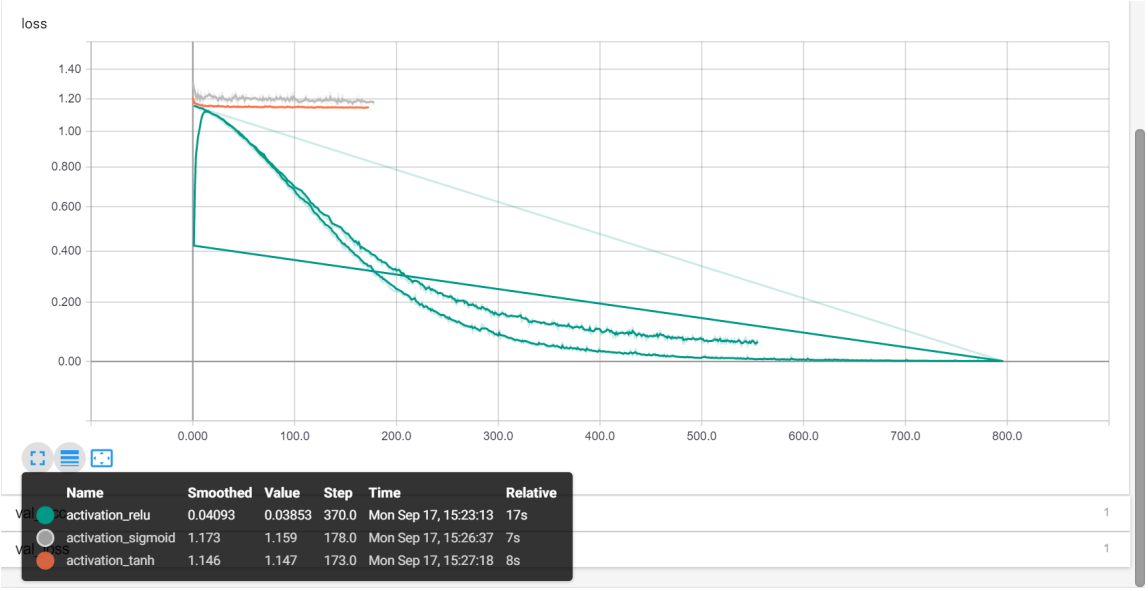
The idea here is to find which activation function is the best for the fizz buzz program.

Figure 1: Accuracy of ReLu, sigmoid, tanh functions



From the figures 1, 2, we can clearly understand that both sigmoid and tanh functions cannot be used as activation to train the model as it suffers drastically. In both sigmoid and tanh functions, the disadvantage is it has a vanishing gradient problem. ReLu rectifies this vanishing gradient problem. Hence it is preferred over others.

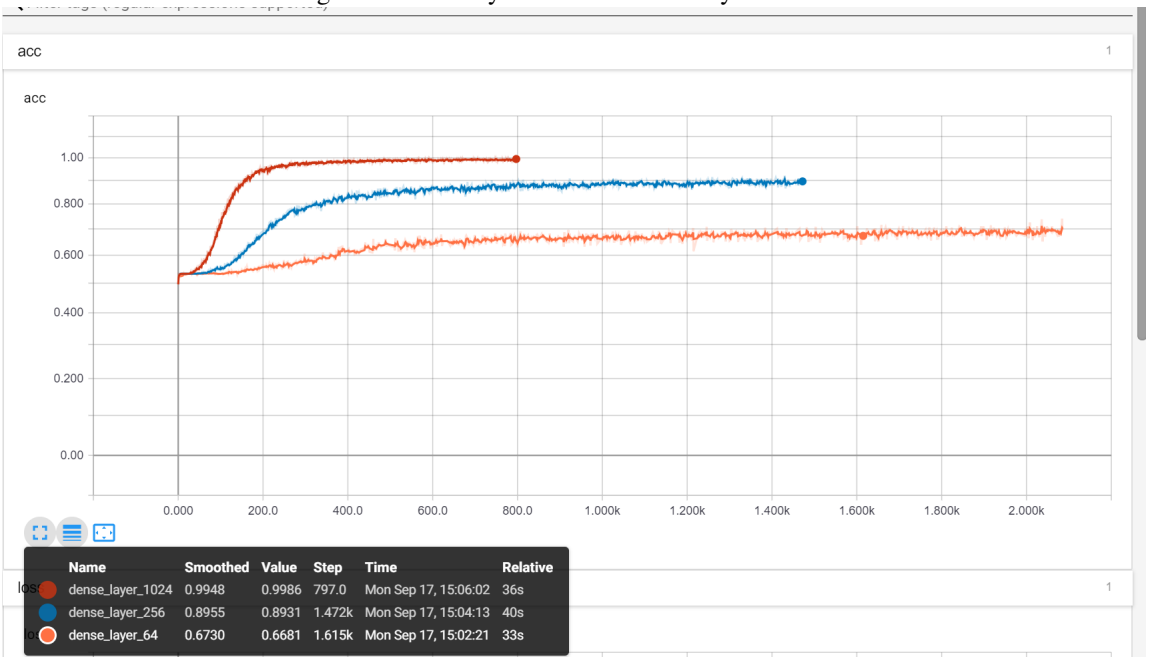
Figure 2: Loss of ReLu, sigmoid, tanh functions



4.2 Dense Layer Nodes

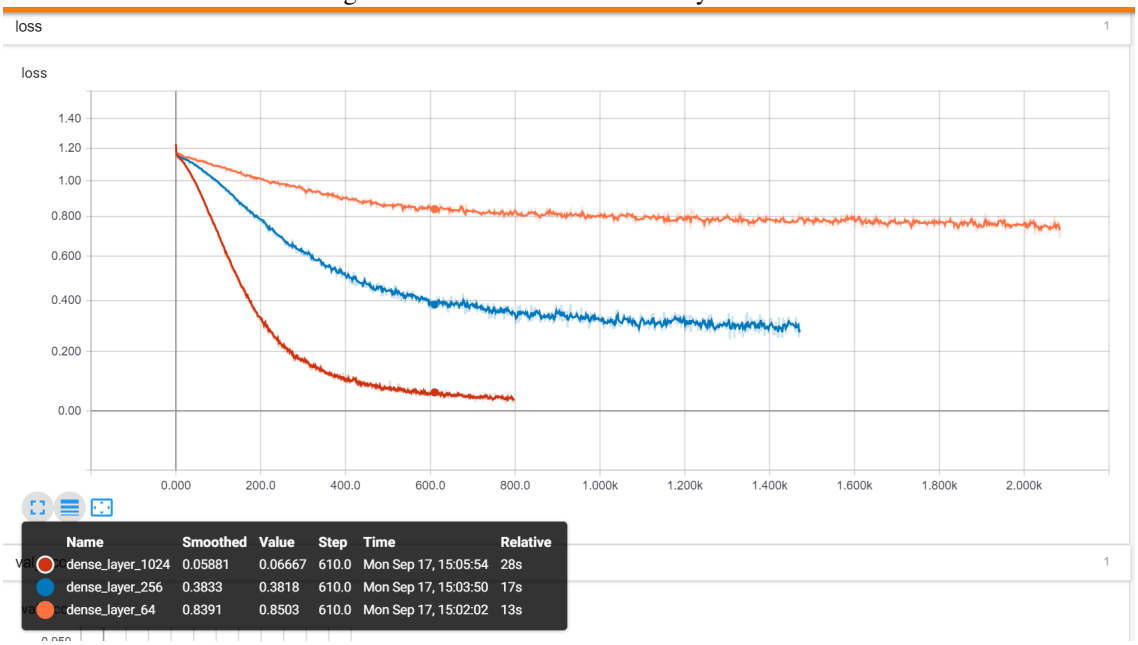
In this section, we will see how the no of dense layer nodes determine the accuracy and loss.

Figure 3: Accuracy of different dense layers



From the graphs 3, 4, we can infer that more the no of dense layers, the learning takes place at a quick rate with respect to this model

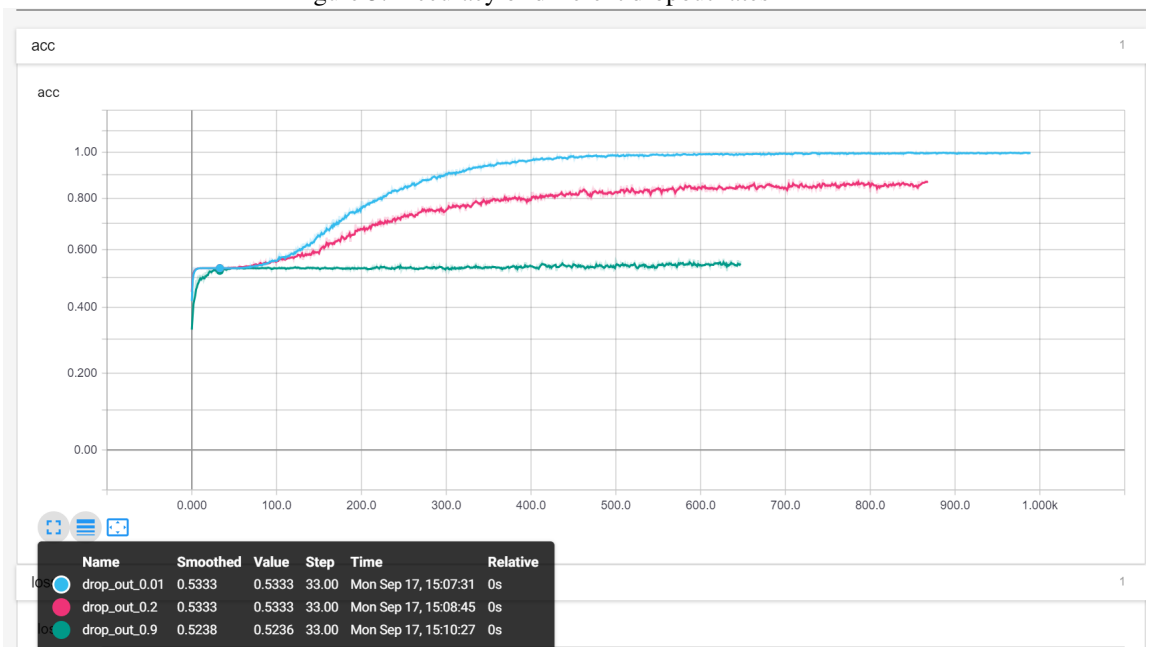
Figure 4: Loss of different dense layers



4.3 Dropout Rates

In this section, we will see how different dropout rates can change the way the model is being trained

Figure 5: Accuracy of different dropout rates



From the plots 5, 6, we can infer that if the dropout is low, accuracy is quite high while the other way around if it is high. So, using a low dropout rate is ideal.

Figure 6: Loss of different dropout rates

4.4 No of Epochs

Number of epochs is the number of times the whole training data is shown to the network while training.We should ideally increase the number of epochs until the validation accuracy starts decreasing even when training accuracy is increasing (overfitting).

4.5 Batch Size

The following section will describe the performance of accuracy and loss against the change of batch size

From the plots 7, 8, we can infer that a good default batch size will be 32.

5 Best Performance and Conclusion

With all the performance based parameters considered, we can find that the activation function which works best is ReLu, the no of dense layers being 1024, drop out rate being 0.2 and batch size 32. The following plots 9 and 10 depict for best accuracy and loss.

Figure 7: Accuracy of different Batch sizes

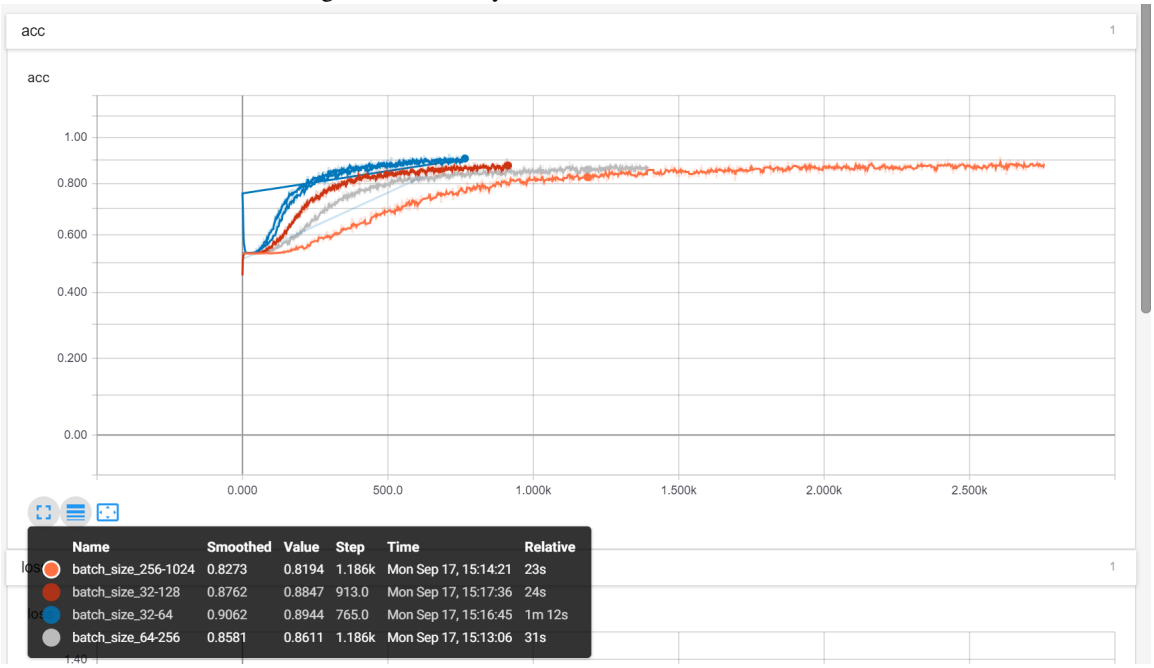


Figure 8: Loss of different Batch sizes

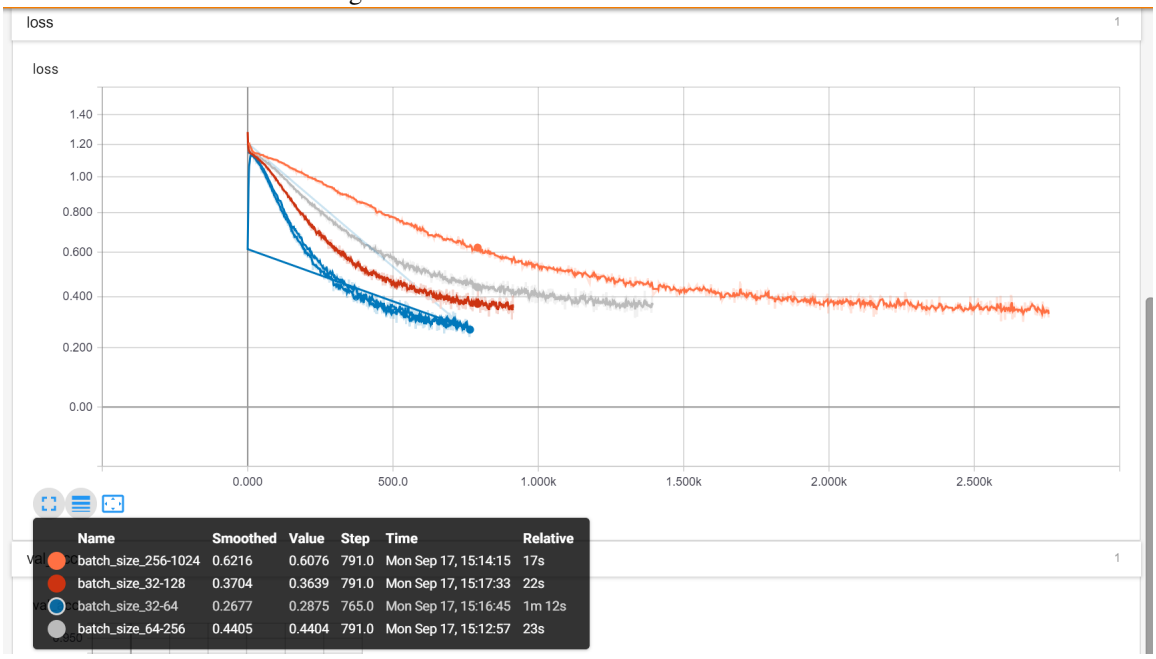


Figure 9: Accuracy of Best Performance Model

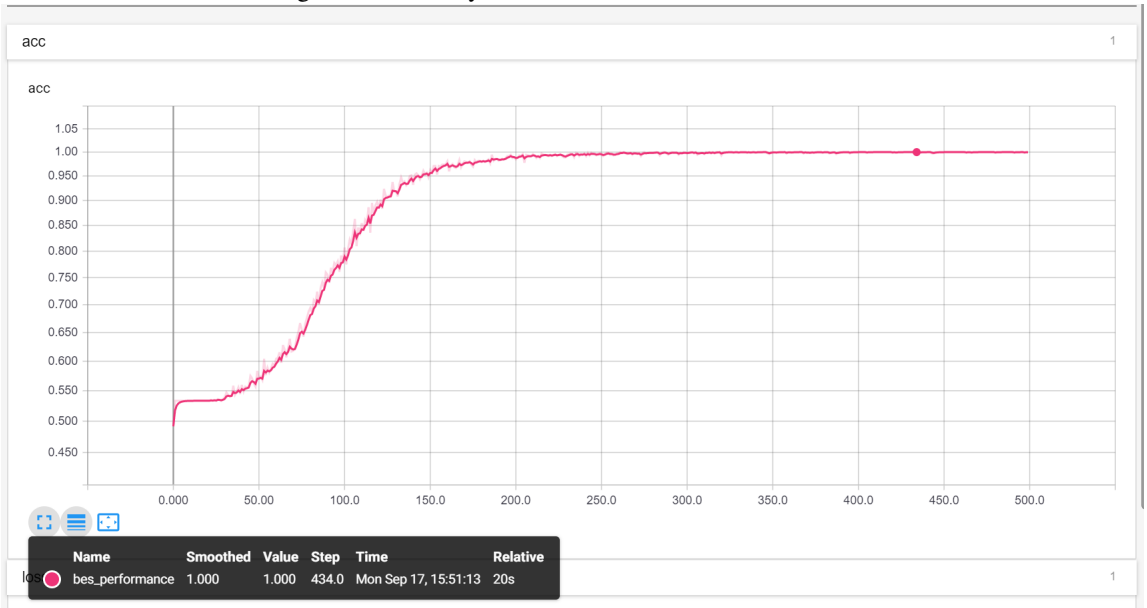


Figure 10: Loss of Best Performance Model