Deep Learning HW1 Report

Alfons Hwu

Student ID: 0416324

alfons.cs04@g2.nctu.edu.tw

Dept of Comuter Science

Writing with LaTeXon Overleaf

Deep Learning HW1 Report
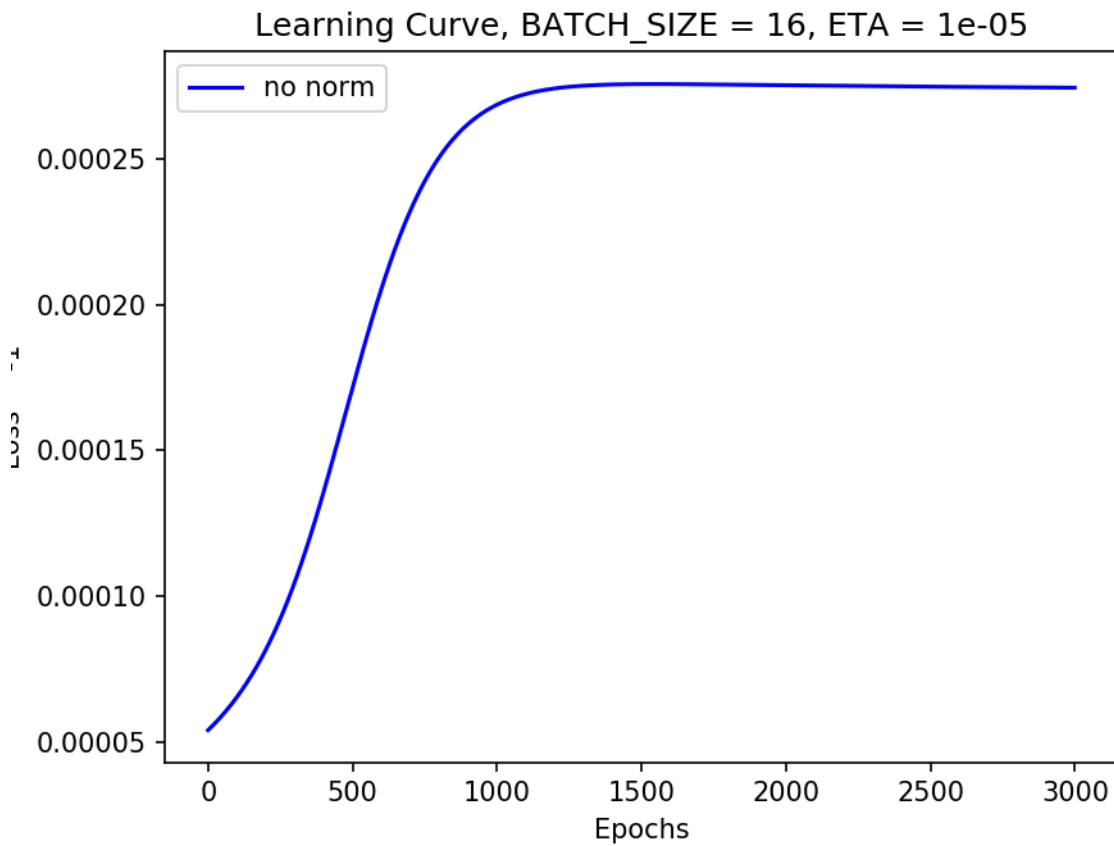
April 6, 2019

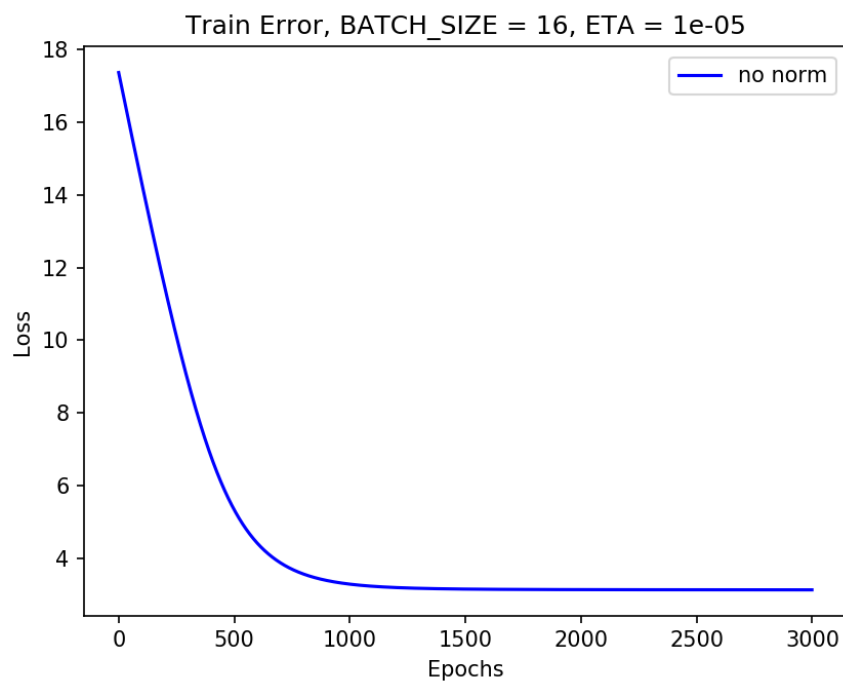## 1 Self-designed DNN for binary classification

Loss is defined by

$$E(w) = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(X_n, w)$$
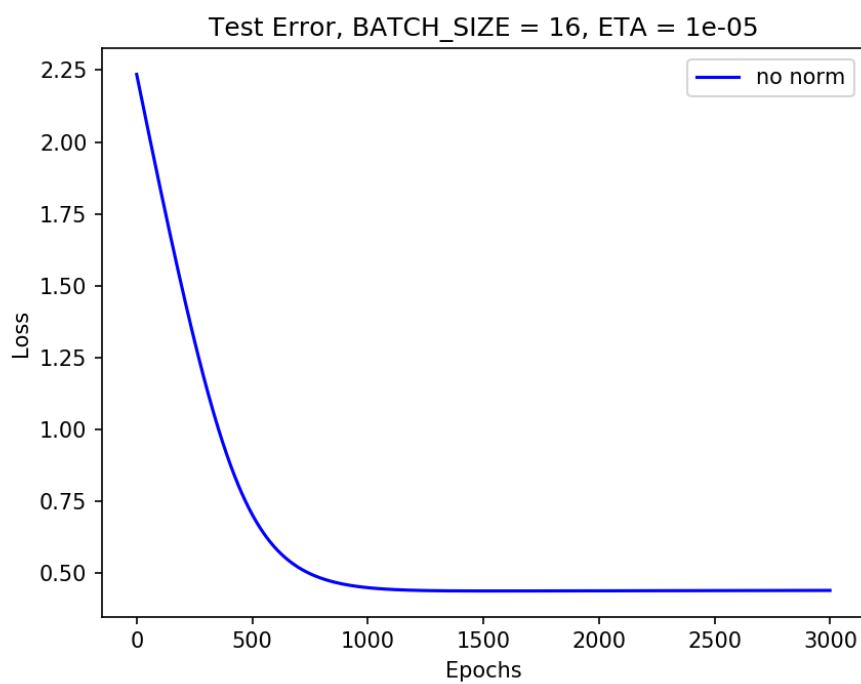
1.1 Learning Curve

Learning Curve, BATCH_SIZE = 16, ETA = 1e-05

The Y-axis represents 1/Loss, telling the learning curve(loss is dropping and finally converged)

1.2 Training Loss

Train Error, BATCH_SIZE = 16, ETA = 1e-05

The Y-axis represents loss, telling the learning curve(loss is dropping and finally converged)

1.3 Testing Loss

Test Error, BATCH_SIZE = 16, ETA = 1e-05

The Y-axis represents loss, telling the learning curve(loss is dropping and finally converged)

1.4 Explanation of my neural network architecture

- Layer: Input 6unit (input including 6 features, which is 6 dimensions).

  Hidden 4unit, since in my opinion, what influences the result most lies in the learning rate, the neuron set 4 will be optimal for computation (too many neurons merely increases the matrix computation time)

  Click for ref

  Output: 1unit for alive or death.

  For the following two items, I select them based on the combination with shell script, and comparing the optimal results using the **same random seed** for benchmarking the learning curve vs batch size and learning rate. (Choose the normal learning rate first (such as 0.5) and slightly decrease by empirical method (automated training with shell script and comparing with pyplot figure).

  According to this article the **high learning rate will cause the non-convergent of learning curve.**

  With the aforementioned site, I start up with learning rate 0.5 and empirically dropped down to about **0.00001 +- 0.000005**

  ```
  if [ $sel -eq 1 ];
  then
      batch_size=$2
      for learning_rate in 0.000001 0.000003 0.000005 0.000008 0.00001 0.000
      do
          echo $learning_rate
          python3 dnn_1.py $1\_$batch_size\_$learning_rate $batch_size $lea
      done
  else
      python3 dnn_1.py $1_8_0.000005 8 0.000005
  fi

  mkdir -p $1_1
  mv $1*\.png $1_1/
  ```

- Batch size: 16 for optimal, (too large will consume too much memory resource, and too small will cause the unstable learning curve. Although more randomness, less chance to converge)
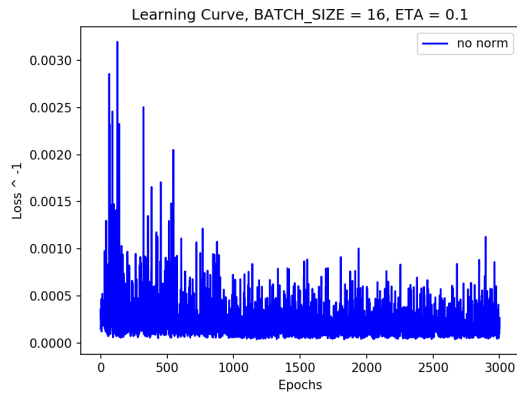
  Ref link

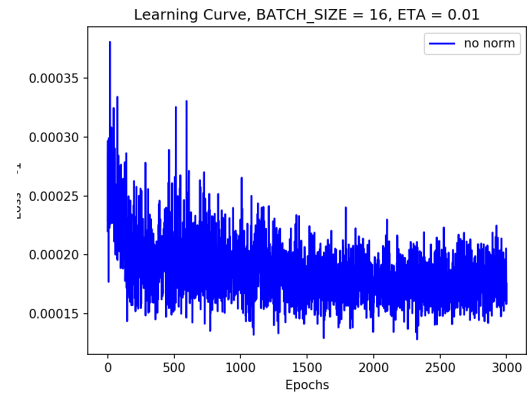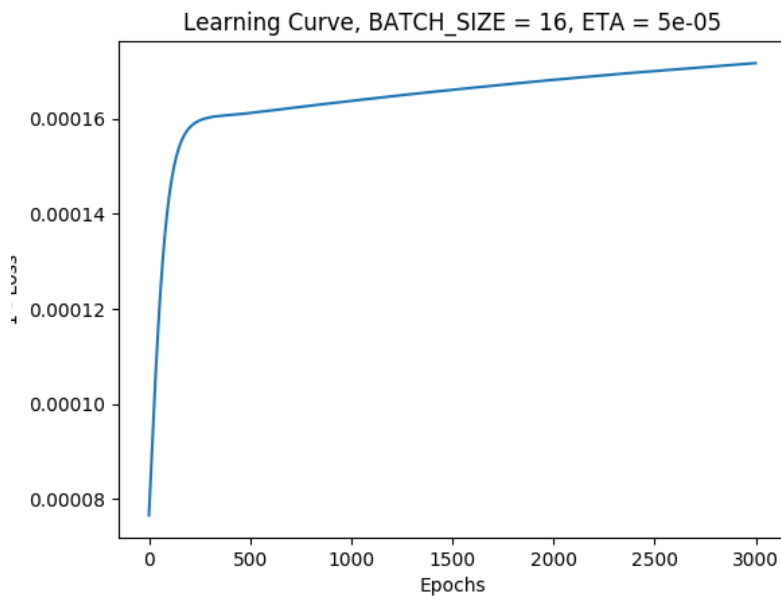Figure 1: Learning rate = 0.1
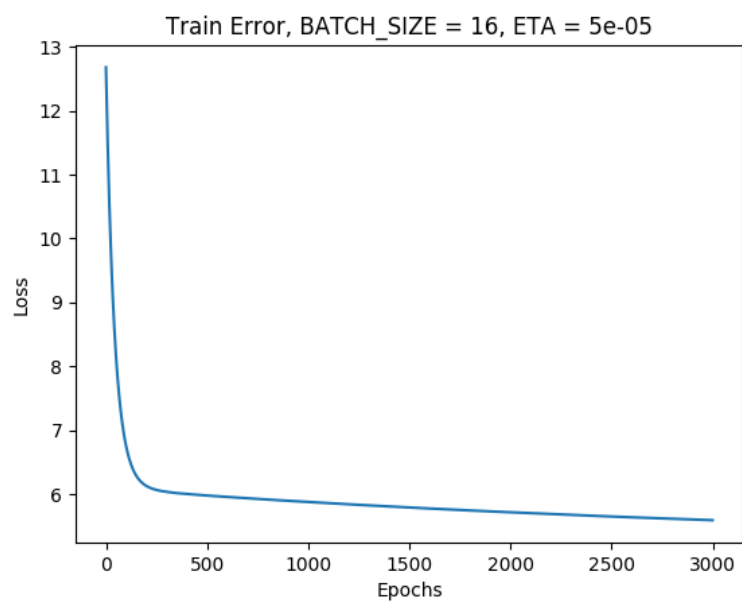

Figure 2: Learning rate = 0.01

- Learning rate: 0.00001 (1e-5) will be optimal (The above figures shows some high learning rate causing non-convergent of learning curve)
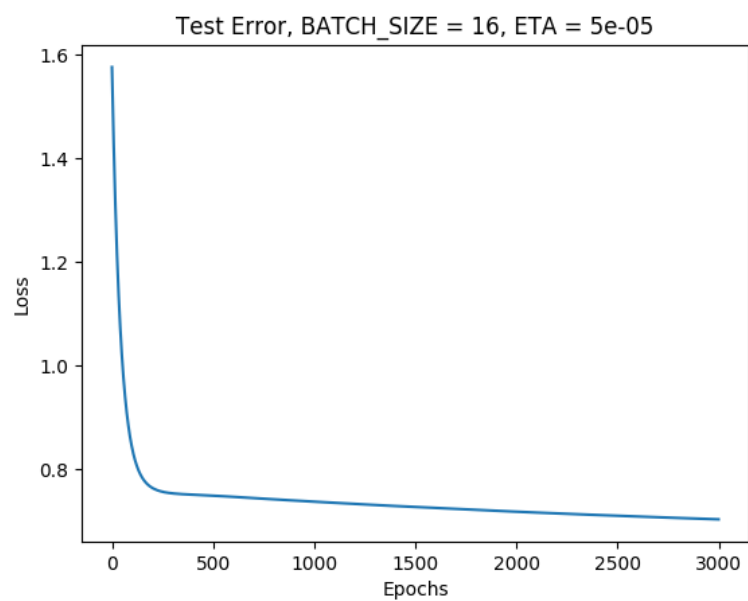
2 TA-designed DNN

The selection of learning rate and training batch size is the same as above
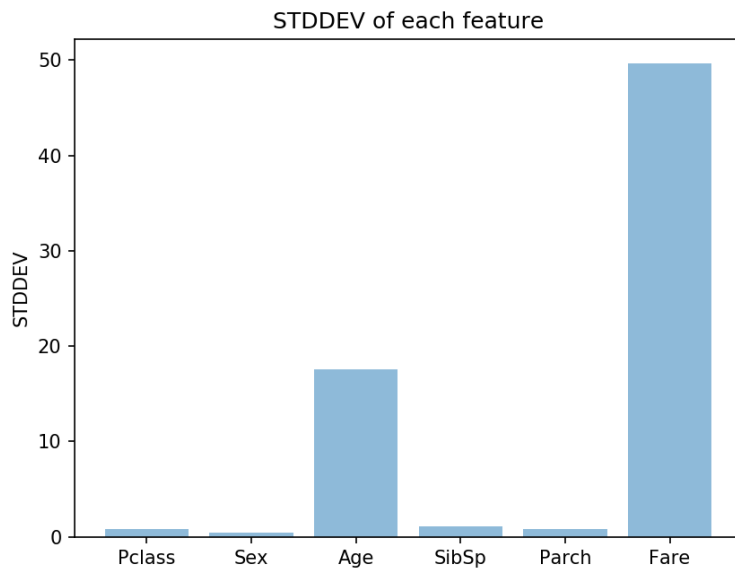

Learning curve

4

Training error

Testing error

3   The normalization of features

Why normalize the data?

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization. It is required only when features have different ranges.

The following figure is the result of normalized fare and normalized all the features, which shows the better performance under normalization (use the same random seed, same batch size and same learning rate for fairness), the normalized results converged better and quicker . Note: Same cross entropy formula is used.
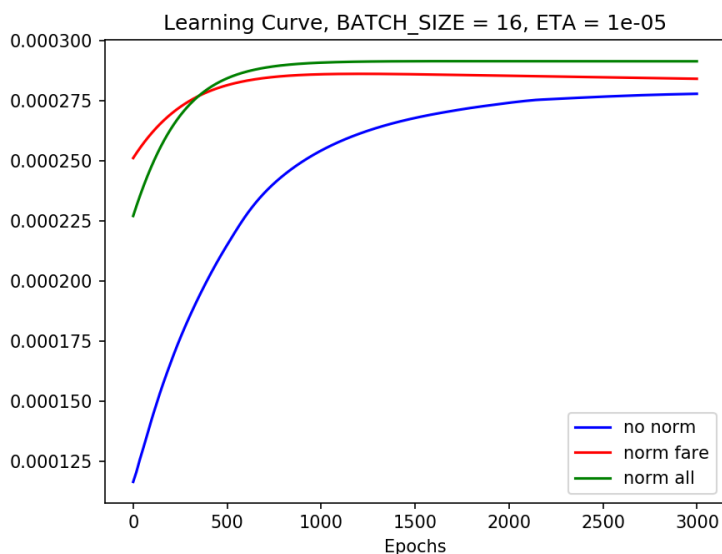
4   Feature that affects the prediction performance most

By calculating the correlation coefficient for the feature vs output (alive or death), the following figure shows the **fare feature** affects the most (the higher the fare is, the more chance a passenger will survive).

CORRCOEF of features vs alive