**VAN-HUNG, LE (黎文雄)**

**Student ID: 0860831**

**DEEP LEANING – REPORT HOMEWORK 1**

1. **Regression**

In this task, we need to minimizing the sum-of-squares-error function:



Evaluate the performance by root-mean-square error (RMS)



1. **Network Architecture**

|  |  |
| --- | --- |
| Network Architecture | 15-10-10-1 |
| Activation function | [sigmoid, sigmoid, sigmoid] |
| Training | 0.06099833 |
| Testing | 0.06167703 |
| Epochs | 10000 |
| Learning rate | 0.002 |
| Batch Size | 32 |
| Training/Testing Size | 75%/25% |

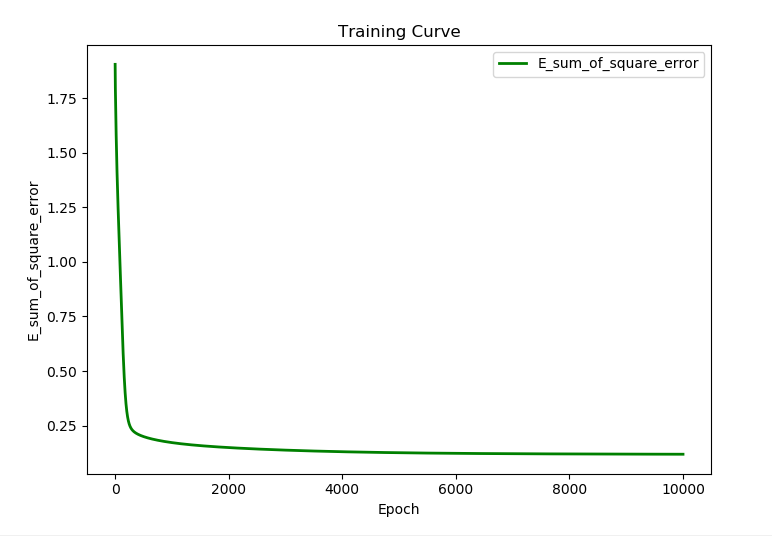
**Data processing:**

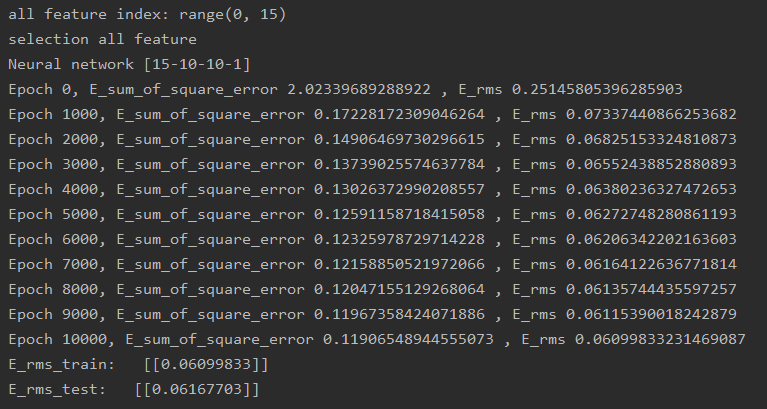
The categorical features (orientation and glazing area distribution) need to encode them into one-hot vectors.

The other features and output data are normalized to be from 0 to 1

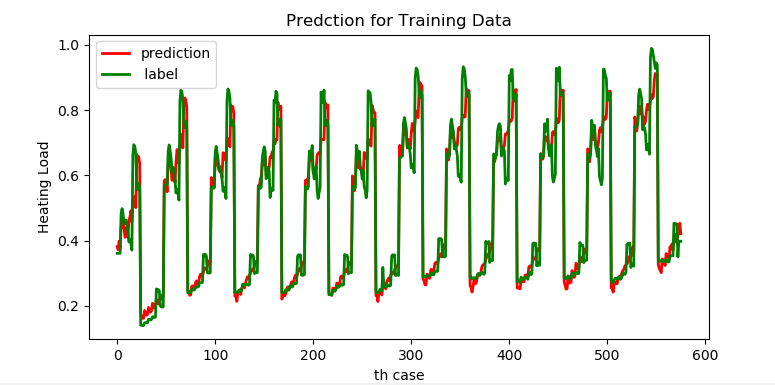
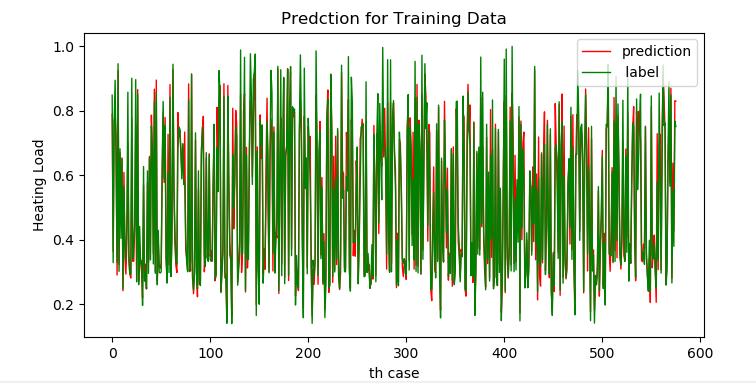
1. **Learning Curve**

Plot graph  of each bath-size, it will be reduced by each epoch



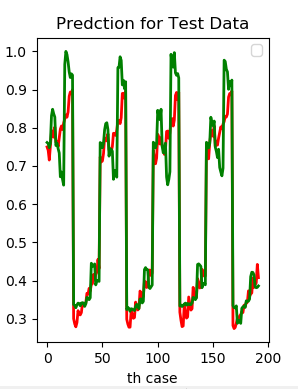


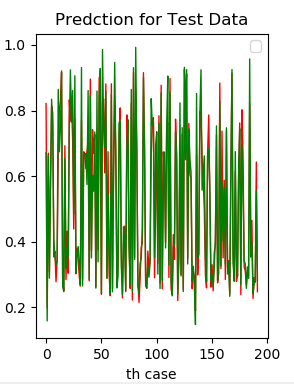
1. **Regression result with training label**

* **Shuffle = FALSE**
* **Shuffle = TRUE**
* 

1. **Regression result with testing label**

* **Shuffle = FALSE**



* **Shuffle = TRUE**
* 

1. **Design a feature selection procedure**

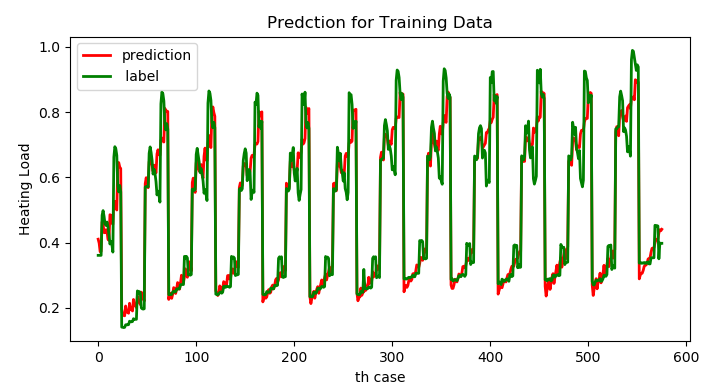
- In this task, I use **variance threshold** method to find out which input features influence the energy load significantly and which features have very little effect on energy load.

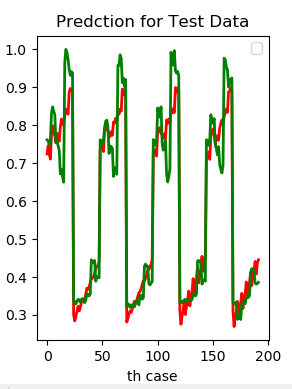
- **The idea in here is when a feature doesn’t vary much within itself, it generally has very little predictive power.**

- Intuitively, we can see that the feature which has the value variance that smaller than a threshold (or the others) will not affect to the result. Similarly, the one with bigger value variance influence result significantly. Now I will remove features to see on it.

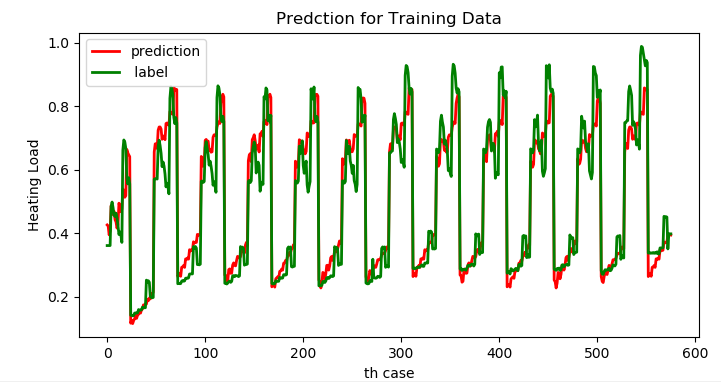
**Remove 1 feature:**

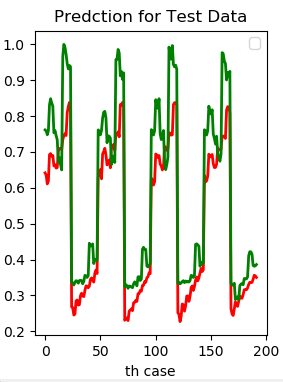
- **Remove feature with minimum value variance** (feature\_index = 3).The result is **NO** different than when using all feature. So I can say that this feature is not important with power prediction.





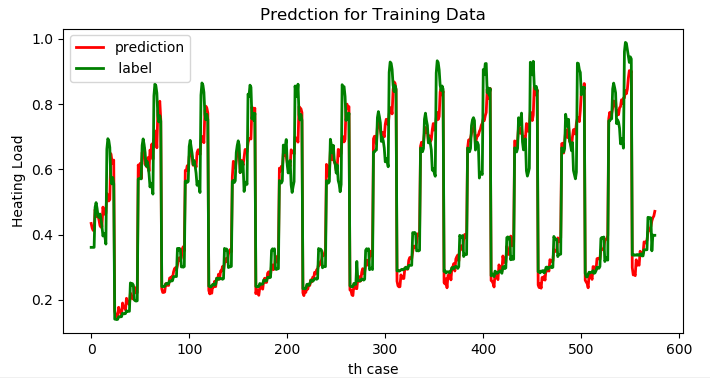
**- Remove feature with maximum value variance** (feature\_index = 6). The result is different than when using all feature. So, I can say that this feature is important with power prediction.

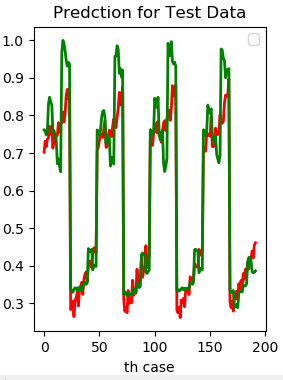




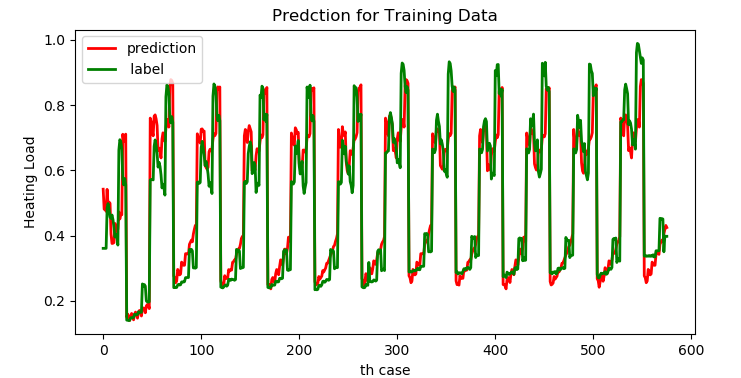
**Remove 2 features:**

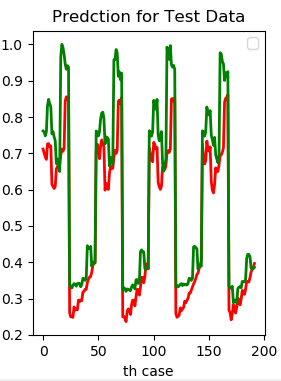
**Remove 2 features with minimum value variance** (feature\_index = [0,3]).The result is verry little different than when using all feature.





**Remove 2 features with maximum value variance** (feature\_index = [4,5]). The result is different than when using all feature.



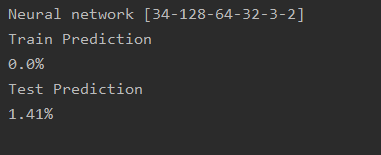


**The error per data comparing between training data and testing data:**

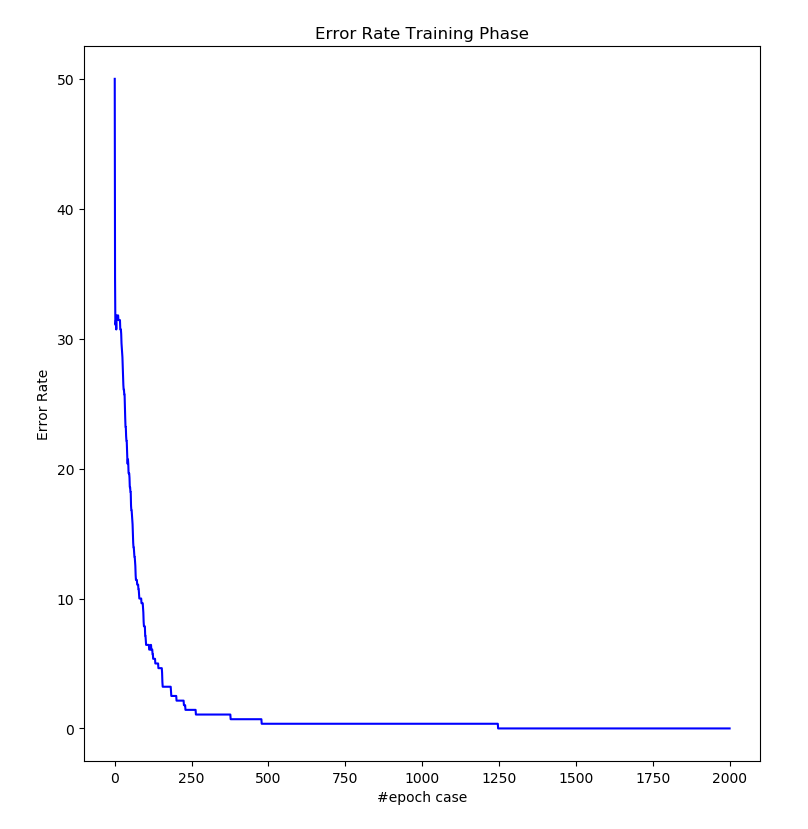
|  |  |  |
| --- | --- | --- |
|  | Error Training | Error Testing |
| Remove 1 feature (minimum value variance) | 0.06160198 | 0.06363681 |
| Remove 2 feature (minimum value variance) | 0.06262843 | 0.07005903 |
| Remove 1 feature (maximum value variance) | 0.07054658 | 0.10302955 |
| Remove 2 feature (maximum value variance) | 0.07904408 | 0.10479482 |

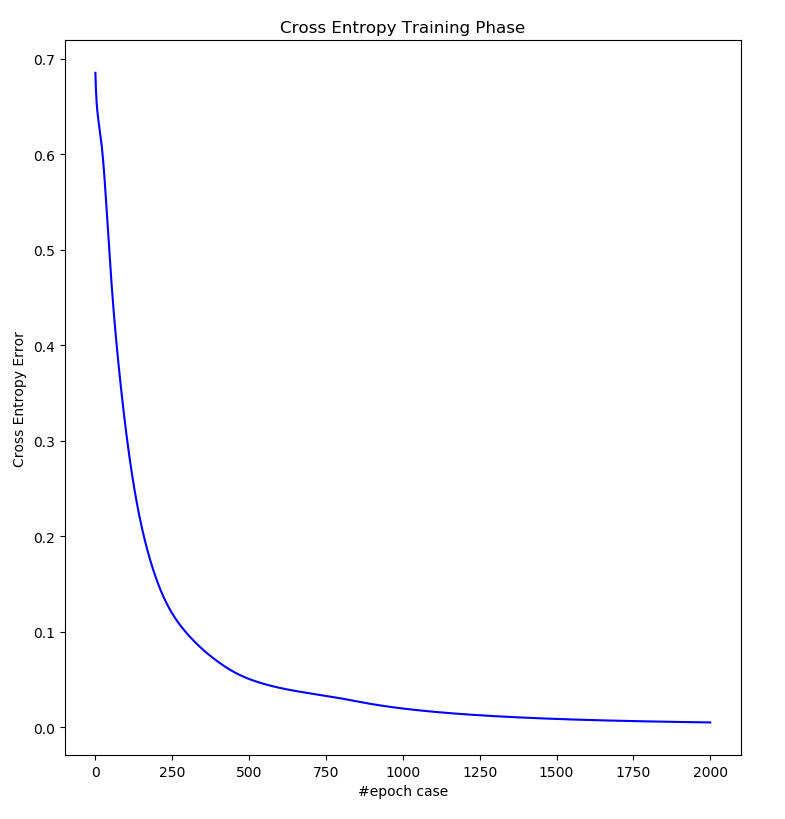
1. **Classification**
2. Network Architecture

|  |  |
| --- | --- |
| Network Architecture | 34-128-64-32-3-2 |
| Error Rate Training | 0.0% |
| Error Rate Testing | 1.41% |
| Epoches | 2000 |
| Learning Rate | 0.0007 |
| Batch Size | 32 |
| Training Size / Testing Size | 80% / 20% |



1. **Learning Curve**





1. **Plotting latent features at different training state**

By changing the hidden layer before output layer, using 2 or 3 nodes, we then

compare the result by visualize (2D or 3D) the distribution of this hidden layer at

diffident training stage.

