Deep Learning
Fall 2024
Homework 1
Name: Anna Kompan (安娜)
ID: 313540009

1. Regression

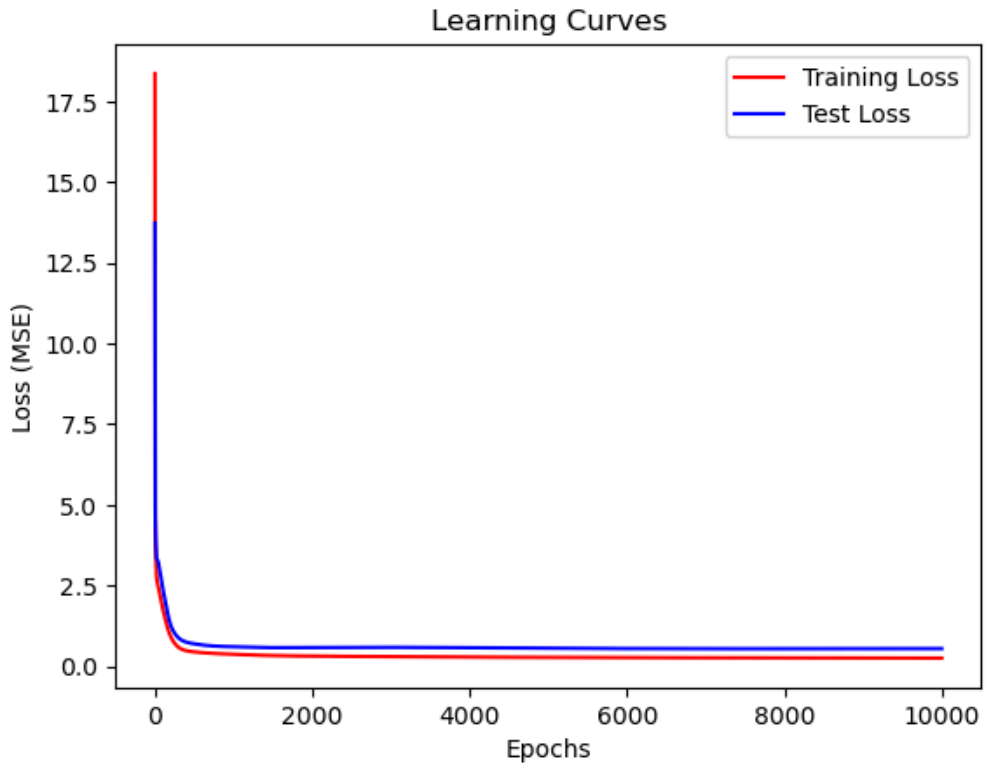
(1) Network architecture:

Input features (in_features = X_train.shape[1]) #16
Output features(out_features = 16)
1 Hidden Layer

```
Model([
    layers.Linear(in_features, out_features),
    layers.Sigmoid(),
    layers.Linear(out_features, 4),
    layers.Sigmoid(),
    layers.Linear(4, 1),
])
```

epochs = 10000
learning_rate = 0.01
batch_size = 32

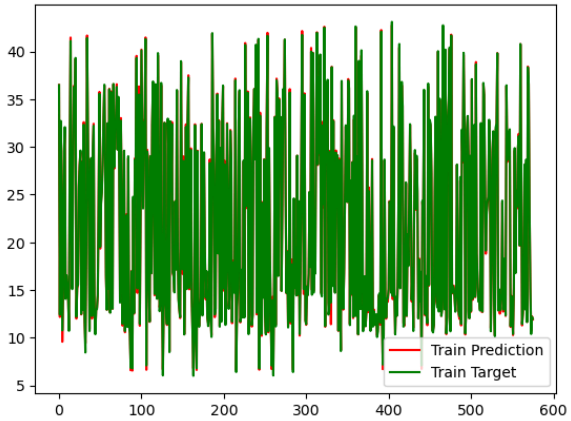
(2) Learning curve

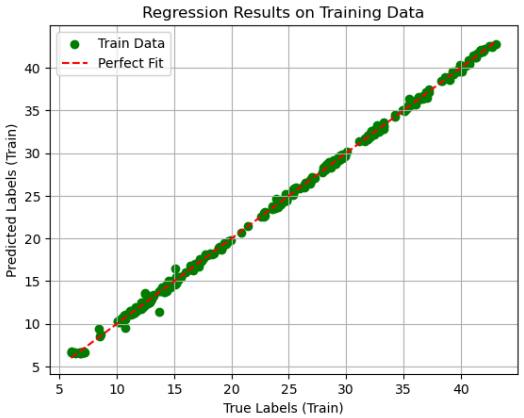


(3/4)Training and Test RMS Error

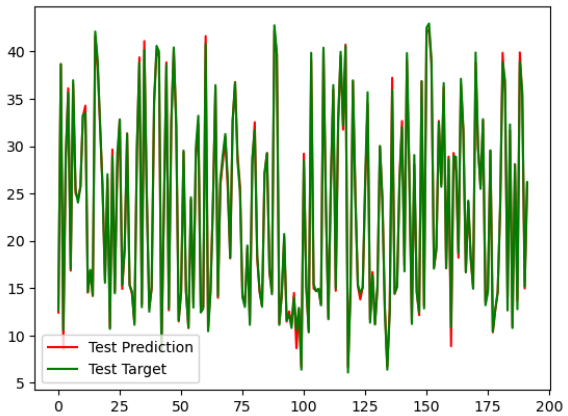
```
Epoch 999: train loss 0.36831727424571253, test loss 0.6030615553529559
Epoch 1999: train loss 0.3113178562656298, test loss 0.5779713110662483
Epoch 2999: train loss 0.2956680231461228, test loss 0.5842916816191894
Epoch 3999: train loss 0.28315732537568433, test loss 0.5723027246853564
Epoch 4999: train loss 0.27275310643480577, test loss 0.5566507966609838
Epoch 5999: train loss 0.2651228386682787, test loss 0.5459885926985769
Epoch 6999: train loss 0.25976466508861984, test loss 0.5410401440774965
Epoch 7999: train loss 0.25588260279529157, test loss 0.5402953338656176
Epoch 8999: train loss 0.2529166407818461, test loss 0.5418024198606921
Epoch 9999: train loss 0.250541355263316, test loss 0.5441922500370032
```

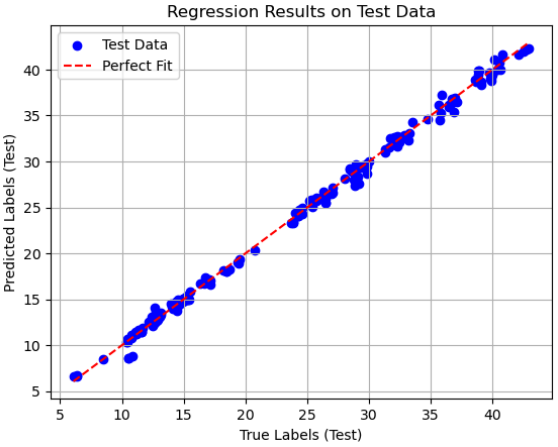
(5)Regression results with training labels (Prediction)





(6)Regression results with test labels (Prediction)





(c)Failure selection procedure to find out which input features influence the energy load significantly.

Firstly set a function that will set zeroes for all features except of one feature at one loop.

Secondly map feature to its index and calculate loss (RMS) for each.

Sort features based on lowest RMS, to see what features are more important.

Result is: “Feature importance: Roof Area is more important than Overall Height is more important than Surface Area is more important than Wall Area is more

important than Glazing Area Distribution is more important than Orientation is more important than Glazing Area is more important than Relative Compactness”

2. Classification

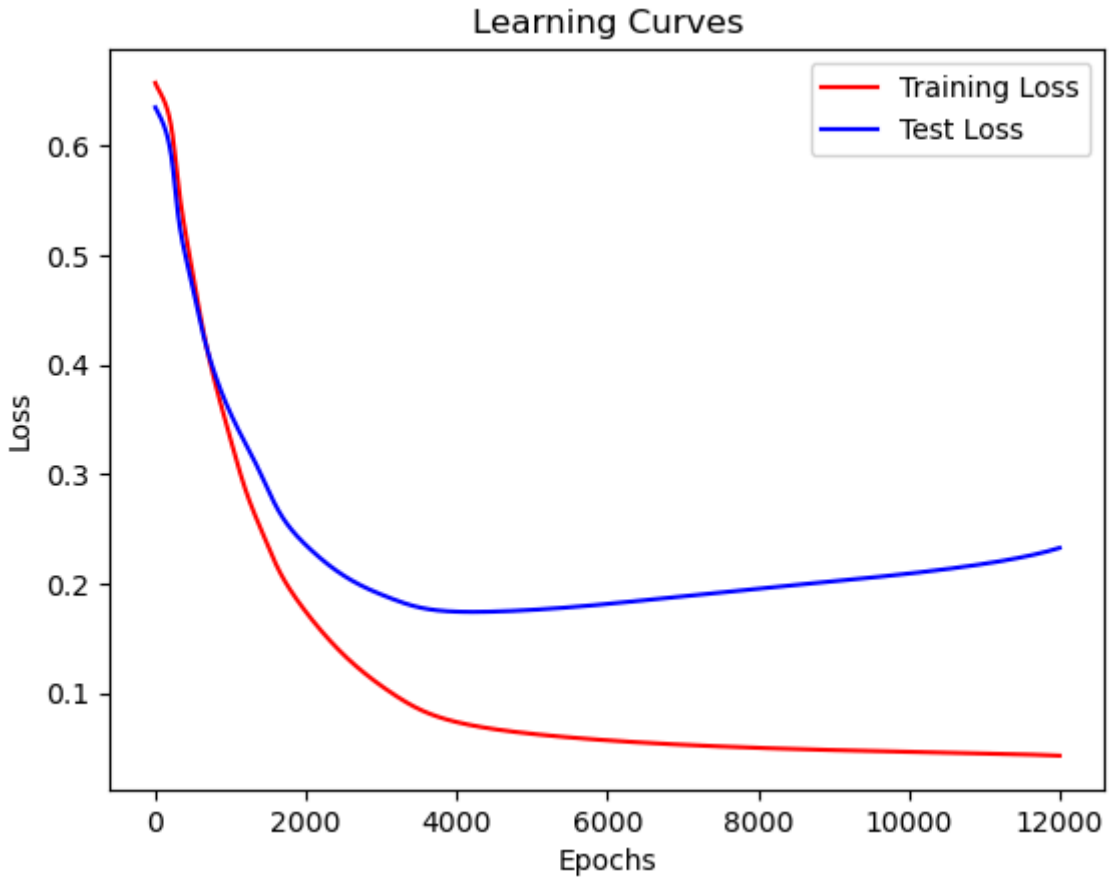
(1) Network architecture:

Input features (in_features = X_train.shape[1]) #34
Output features(out_features = 8)
1 Hidden Layer

```
model = Model([
    layers.Linear(in_features, out_features),
    layers.Sigmoid(),
    layers.Linear(out_features, 2),
    layers.Sigmoid(),
    layers.Linear(2, 1),
    layers.Sigmoid(),
])
```

epochs = 12000
batch_size = 64
learning_rate = 0.03

(2) Learning curve:



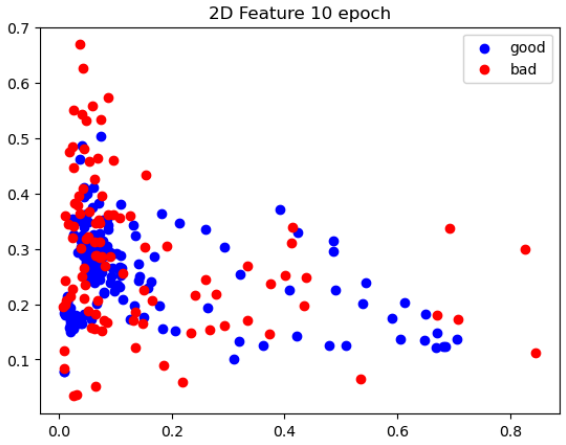
(3/4)Training and Test Error Rate

```
Epoch 999:
train loss 0.24747658783843046, test loss 0.26233223512606607
Epoch 1999:
train loss 0.10143390952395978, test loss 0.13771541612196447
Epoch 2999:
train loss 0.061810592244374796, test loss 0.12051535562926512
Epoch 3999:
train loss 0.04598188064500199, test loss 0.12385128484030916
Epoch 4999:
train loss 0.03723064424750221, test loss 0.13390044926743735
Epoch 5999:
train loss 0.03166793749905687, test loss 0.14541243444797375
Epoch 6999:
train loss 0.027801684107039742, test loss 0.15631020891796196
Epoch 7999:
train loss 0.024848133396530022, test loss 0.16626786983905006
Epoch 8999:
train loss 0.02238960934854244, test loss 0.1754503955617927
Epoch 9999:
train loss 0.020198971793111826, test loss 0.18406327829762156
Epoch 10999:
train loss 0.018146214703761123, test loss 0.19223450291786556
```

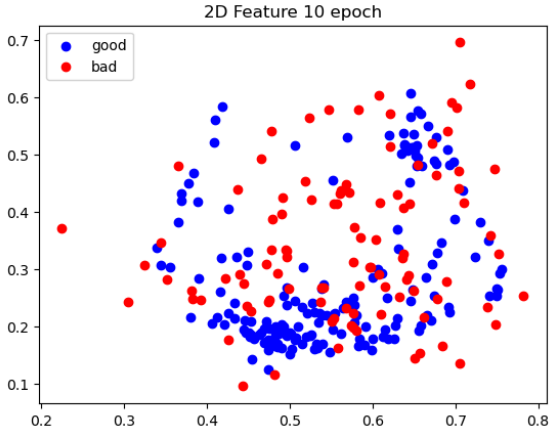
(c) Distribution of latent features at different training stages.

1) 2D distribution of latent features after 10 epoch

2 Neurons

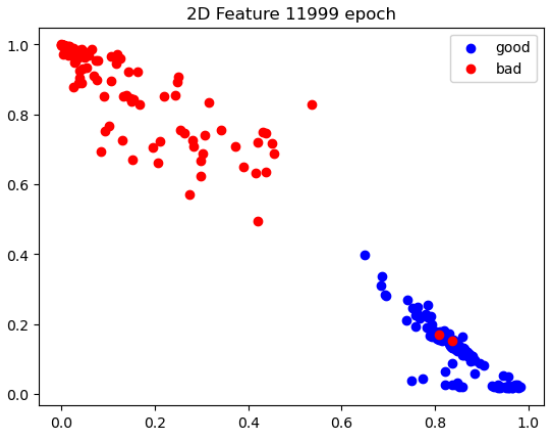


3 Neurons



2) 2D distribution of latent features after 11999 epoch

2 Neurons



3 Neurons