Deep Learning (Fall 2024)

Homework 1

Deadline 10/18 (Friday) 23:59

In this homework, you are asked to build a deep neural network model by using the backpropagation and stochastic gradient descent algorithm. You may design the network architecture by yourself, including the number of hidden layers, the number of hidden units, learning rate, the number of epochs and mini-batch size. Do **not** use available machine learning or deep learning packages. It means that you cannot use function like "tf.nn.dense", " tf.nn.sigmoid cross entropy with logits" which have been already written. Put your **source code** and the **report** with results into a com- pressed file (**hw1 StudentID.zip**) and submit through e3.

1) Regression

Implement the neural network for regression by using the energy efficiency dataset. There are 2 simulation energy loads and 8 different features in this dataset. Shuffle the dataset then use 75% of data samples for training and 25% for testing. Note that for the categorical features (orientation, glazing area distribution), you need to encode them into one-hotvectors.

Attribute information: Heating load, Cooling Load, Relative compactness, Surface area, Wall area, Roof area, Overall height, Orientation (north, south, east or west), Glazing area, Glazing area distribution (uniform, north, south, east or west).

(a) Please try to predict the **heating load** of buildings by minimizing the sum-of-squares error function.

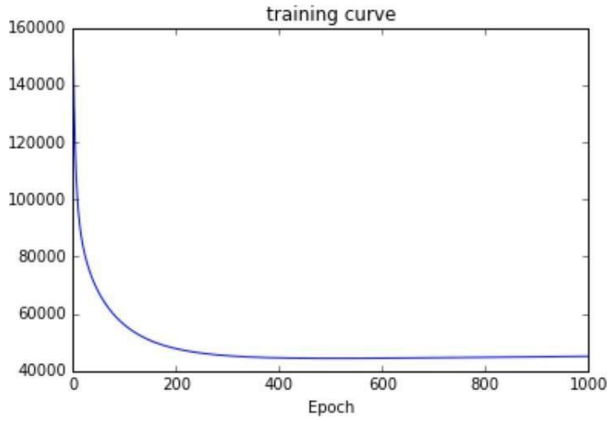
$$E(\mathbf{w}) = \sum_{n=1}^{N} (t_n - y(\mathbf{x}_n; \mathbf{w}))^2$$

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

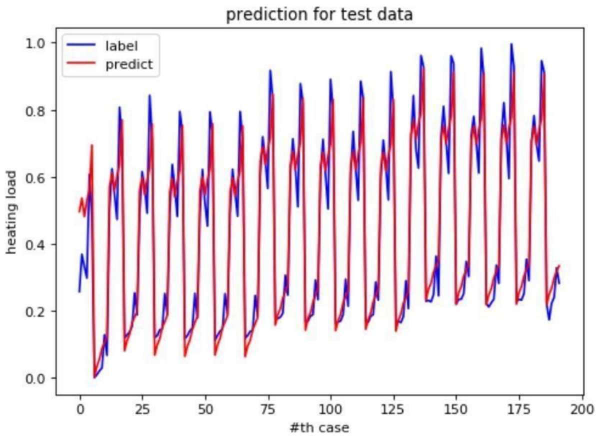
$$E_{RMS}(\mathbf{w}) = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (t_n - y(x_n; \mathbf{w}))^2}$$

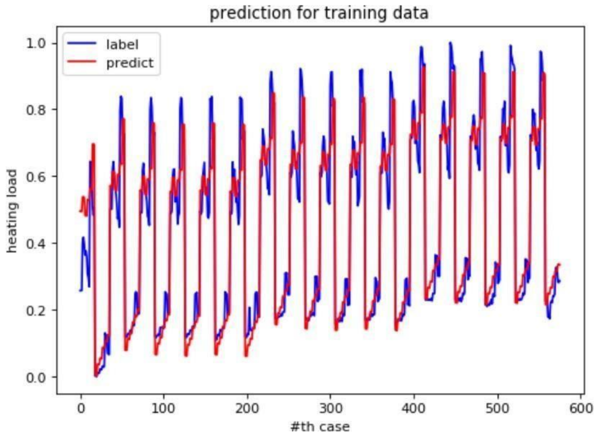
(b) Show your (1) network architecture (number of hidden layers and neurons), (2) learning curve, (3) training RMS error, (4) test RMS error, (5) regression result with training labels and (6) regression result with test labels in the report.

(c) Design a feature selection procedure to find out which input features influence the energy load significantly and explain why it works. You may compare the performance of choosing different features.



Network architecture	$15 - 10 - 10 - 1$
Selected features	$[0, \ldots, 7]$
Training E_{RMS}	5.94988
Test E_{RMS}	5.99459





2) Classification

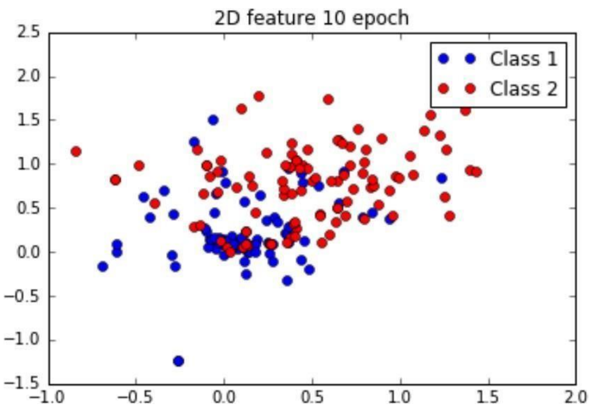
Implement the neural network for binary classification by using the Ionosphere dataset. There are 34 different features and 2 classes. The last column represents their corresponding labels:"g" for good and "b" for bad. Use 80% of data samples for training and 20% for testing.

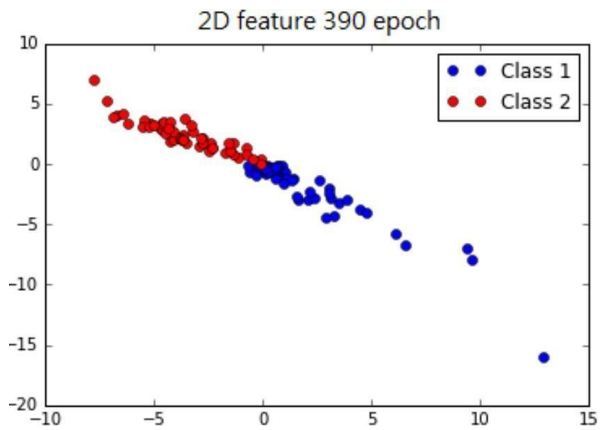
(a) Please try to classify the Ionosphere data by minimizing the cross-entropy error function.

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_k(\mathbf{x}_n, \mathbf{w})$$

(b) Show your (1) network architecture (number of hidden layers and neurons), (2) learning curve, (3) training error rate, (4) test error rate in the report.

(c) Compare the results of choosing different numbers of nodes in the layer before the output layer by plotting the distribution of latent features at different training stage.





Recommended library:

- **NumPy** (dot, multiply...)
- **Matplotlib**
- **Pandas** for reading csv file