

ML Assignment 3

Name : Utkarsh Aditya

Roll No : S20180010182

Softmax Regression

Overview

Softmax Regression is a generalization of logistic regression that we can use for multi-class classification. In *Softmax Regression* (SMR), we replace the sigmoid logistic function by the so-called *softmax* function. We use the *Logistic Regression* model in binary classification tasks

$$P(y = j | z^{(i)}) = \phi_{softmax}(z^{(i)}) = \frac{e^{z_j^{(i)}}}{\sum_{k=0}^K e^{z_k^{(i)}}},$$

where we define the net input z as

$$z = w_1 x_1 + \dots + w_m x_m + b = \sum_{l=1}^m w_l x_l + b = \mathbf{w}^T \mathbf{x} + b.$$

Understanding

The softmax function computes the probability of the training sample $\mathbf{x}^{(i)}$ belonging to class j given the weight and net input $z^{(i)}$. So, we compute the probability $p(y=j|\mathbf{x}^{(i)}; \mathbf{w}(j))$ for each class label in $j=1, \dots, K$. The normalization term in the denominator which causes these class probabilities to sum up to one.

Formulas Used

1. Update Rule for weight vector

$$\mathbf{w}_j := \mathbf{w}_j - \eta \nabla_{\mathbf{w}_j} J(\mathbf{W}; \mathbf{b})$$

where η is the learning rate, \mathbf{w}_j is the weight matrix and, the cost derivative is

$$\nabla_{\mathbf{w}_j} J(\mathbf{W}; \mathbf{b}) = \frac{1}{n} \sum_{i=0}^n [\mathbf{x}^{(i)} (O_i - T_i)]$$

where O_i is the output vector and T_i is the target vector.

and for updating bias vector, the following formula is user

$$\mathbf{b}_j := \mathbf{b}_j - \eta \left[\frac{1}{n} \sum_{i=0}^n (O_i - T_i) \right].$$

2. Cross Entropy function

$$H(T_i, O_i) = - \sum_m T_i \cdot \log(O_i).$$

Here the T stands for "target" (i.e., the *true* class labels) and the O stands for output -- the computed *probability* via softmax; **not** the predicted class label.

Procedure

Libraries Used : Numpy and csv

1. **Reading the data :** The train.csv is read and the data is split into feature vectors array and respective class labels array.
2. **Training :** The theta matrix is started off as a zero matrix. Using the update formula the theta matrix is kept on updating until a certain number of epochs are reached.
3. **Prediction :** Given the input feature vector, select the class which has the maximum probability among the probabilities vector returned by the softmax function.
4. **Testing(Accuracy) :** If the prediction matches with the class of the feature vector in the test set, the prediction is correct. The performance of the classifier is tested by taking the number of correct predictions and dividing them by the total number of predictions. This is critical in understanding the veracity of the model.

Results

Stopping Criterion : Number of epochs

1. **Training Set Accuracy :** 0.94
2. **Test Set Accuracy :** 0.91

Feed Forward Neural Network

Overview

A Feed Forward Neural Network is an artificial neural network in which the connections between nodes does not form a cycle. The feed forward model is the simplest form of neural network as information is only processed in one direction.

Understanding

The feedforward model is so called because information flows through the function being evaluated from \mathbf{x} , through the intermediate computations used to define f , and finally to the output y . There are no feedback connections in which outputs of the model are fed back into itself.

Formulas Used

1. Cost Error

$$J(W) = \frac{1}{2} \sum_{r=1}^c (t_r - z_r)^2$$

Backpropagation implementation using chain rule

$$\begin{aligned}\frac{\partial J}{\partial w_{ji}} &= \underbrace{\frac{\partial J}{\partial net_j}}_{-\delta_j} \underbrace{\frac{\partial net_j}{\partial w_{ji}}}_{x_i} = \left(\sum_{r=1}^c \underbrace{\frac{\partial J}{\partial net_r}}_{-\delta_r} \underbrace{\frac{\partial net_r}{\partial y_j}}_{w_{rj}} \underbrace{\frac{\partial y_j}{\partial net_j}}_{f'(net_j)} \right) x_i \\ &= \underbrace{\sum_{r=1}^c -\delta_r w_{rj} f'(net_j)}_{\frac{\partial J}{\partial net_j} = -\delta_j} x_i \\ \Delta w_{ji} &= \eta x_i \delta_j\end{aligned}$$

2. Activation Function

$$f(t) = \frac{1}{1 + e^{-t}}$$

3. Hypothesis Function

$$\begin{aligned}h &= \sigma(W_1 x + b_1) \\ y &= \sigma(W_2 h + b_2)\end{aligned}$$

Weights

Activation functions

Procedure

Libraries Used : Numpy and panda

1. **Reading the data :** The train.csv is read and the data is split into feature vectors array and respective class labels array.
2. **Training :** First the input is multiplied with weights and then they are passed to the activation function and then the output is feed forwarded to the next layer. Then according to the error the weights are balanced using the backpropagation algorithm.
3. **Prediction :** Given the input feature vector, the class cell with the highest score in the output layer is selected.
4. **Testing(Accuracy) :** If the prediction matches with the class of the feature vector in the test set, the prediction is correct. The performance of the classifier is tested by taking the number of correct predictions and dividing them by the total number of predictions. This is critical in understanding the veracity of the model.

Results

Stopping Criterion : Number of epochs

1. **Training Set Accuracy :** 87.91 %
2. **Test Set Accuracy :** 87.84 %

Comparison

Softmax Regression

Performance :-> Training accuracy : 0.94 Testing Accuracy : 0.91

Stopping Criterion :-> Number Of Epochs = 500 Learning Rate = 0.1

Feed Forward Neural Network

Performance :-> Training accuracy : 0.94 Testing Accuracy : 0.91

Stopping Criterion :-> Number Of Epochs = 280 Learning Rate = 0.45

For the given dataset softmax regression classifier is performing better than the feed forward neural network. The Number of epochs for Softmax is more so it fits better to the given data compared to the Feed forward neural network, but we know in general FFNN should perform better. Better performance of FFNN can be achieved if number of epochs is increased, but it will increase the training time.
