1. Problem

Logistic Regression:

① Implement the logistic regression algorithm for two-class discrimination.

② Use non-linear combinations of inputs to increase the capacity of the classifier.

③ Implement the logistic regression algorithm for K-class discrimination.

④ Evaluate the performance of the algorithms you implemented.

Multilayer perceptron:

① Assume a two layer feedforward MLP with a single output where all the elements use sigmoid activation. Derive the backpropagation update equations for the weights of the output by minimizing the following error function:

$$\frac{1}{2}\sum_{i=1}^{m}(y^{(i)}-\hat{y}^{(i)})^2$$

Compare the result you obtain to that obtained in class using maximum likelihood estimation.

② Implement a two layer feedforward MLP for three class-classification. Evaluate the performance of the algorithm

2. Proposed Solution

Logistic Regression

① Find a 2 class dataset using Logistic regression to calculate the result and evaluate it

② Find a dataset that is not linear combinations of inputs to get the result by logistic regression and evaluate it.

③ Find a K-class dataset of inputs to get the result by logistic regression and evaluate it.

Multilayer Perception

Separate calculation to several layers and every layer is sigmoid. Derive the equation by

$$\frac{1}{2}\sum_{i=1}^{m}(y^{(i)}-\hat{y}^{(i)})^2$$

3. Implementation

Logistic Regression Equation derive process as follows:

$$h_\theta(x)=\frac{1}{1+\exp(-\theta^T x)}$$

$$\theta \leftarrow \theta-\eta\sum_{i=1}^{n}(h_\theta(x^{(i)})-y^{(i)})x^{(i)}$$

$$h_{\theta_j}(x)=\frac{\exp(\vartheta_j^T x)}{\sum_{i=1}^{k}\exp(\theta_i^T x)}$$

$$\theta_j \leftarrow \theta_j-\eta\sum_{i=1}^{n}(h_{\theta_j}(x^{(i)})-1(y^{(i)}-j))x^{(i)}$$

We suppose

$$\{x^{(i)}, y^{(i)}\}_{i=1}^{m} \qquad \begin{matrix} x^{(i)} \in R^n \\ y^{(i)} \in [j....k] \end{matrix}$$

$$\log\left(\frac{p(y=j \mid x)}{p(y=k \mid x)}\right) = \theta_j^T x \rightarrow p(y=j \mid x) = \frac{\exp(\theta_j^T x)}{1 + \sum_{i=1}^{k-1} \exp(\theta_i^T x)}$$

Then we have

$$h_{\theta j}(x) = p(y=j \mid x) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^{k} \exp(\theta_i^T x)}$$

Gradient decent

$$\theta_{j+1} \leftarrow \theta_j - \eta \sum_{i=1}^{m} (h_{\theta j}(x^{(i)}) - 1(y^{(i)} = j)) x^{(i)}$$

Using Logistic Regression to train data to get the parameters Θ and calculate the result of test data. Compare the expect result and the real result.

Multilayer Regression

My update equations for the weights of the output by minimizing the following error function:

$$\frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \hat{y}^{(i)})^2$$

The neural network model has unknown parameters, often called weights, and we seek values for them. We denote the complete set of weights by Θ which consists of

$\{\alpha_{0m}, \alpha_m; m = 1,2,...M\}$ M(P+1) weights

$\{\beta_{0k}, \beta_k; k = 1,2,...,k\}$ K(M+1) weights

Depends

$$R(\theta) = \sum_{i=1}^{N} R_i = \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2$$

$$\frac{\partial R_i}{\partial \beta_{km}} = -2(y_{ik} - f_k(x_i)) g_k^{'}(\beta_k^T z_i) z_{mi}$$

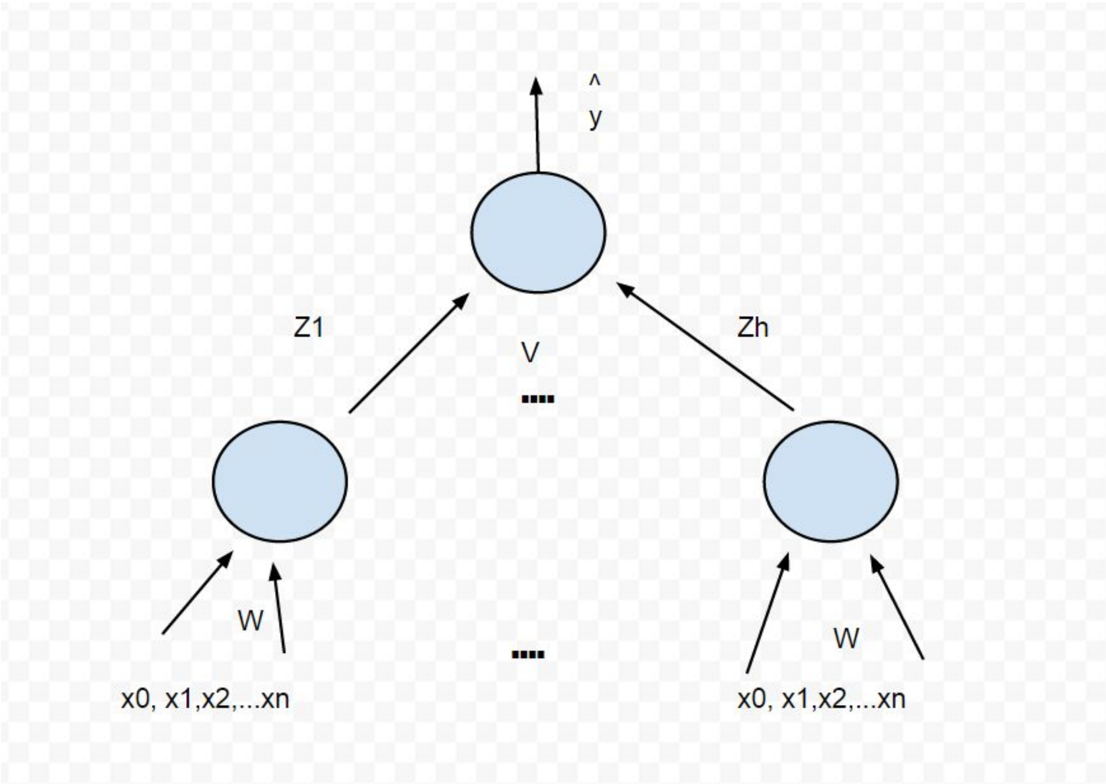
$$\frac{\partial R_i}{\partial \alpha_{ml}} = -\sum_{k=1}^{k} 2(y_{ik} - f_k(x_i)) g_k^{'}(\beta_k^T z_i) \beta_{km} \sigma^{'}(\alpha_m^T x_i) x_{il}$$

Gradient Decent:

$$\beta_{km}^{(r+1)} = \beta_{km}^{(r)} - \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{(r)}}$$

$$\alpha_{ml}^{(r+1)} = \alpha_{ml}^{(r)} - \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{ml}^{(r)}}$$

The Maximum likelihood estimation



Objective:

$$E(\{w_i\}, V) = \frac{1}{2} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2$$

Update parameters:

$$v_{i+1} \leftarrow v_i - \eta \frac{\partial E}{\partial V}$$

$$w_{j+1} \leftarrow w_j - \eta \frac{\partial E}{\partial w_j}$$

Using chain rule:

$$\frac{\partial E}{\partial V} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial V}$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_j} \frac{\partial z_j}{\partial w_j}$$

We get

$$v \leftarrow v - \eta \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) z^{(i)}$$

$$w_j \leftarrow w_j - \eta \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) v_j z_j^{(i)} (1 - z_j^{(i)}) x^{(i)}$$

4. Result and Discussion

Logistic Regression

	Test Accuracy(90%Train, 10%Test)	Average 5-fold Cross Validation
Two-class	1	0.37
Non-Linear Combination	0.269	0.24
K-class(Image)	0.895	0.91

Analyze:

For Two-class: Test accuracy is 100%, because the number of example data is too small (30)and it just have two classes. In 5-fold cross validation,the accuracy is very low. We can see logistic regression is not very good at this dataset.

Non-Linear Combination:Although we have a lot of training data(4177), the inputs is not linear, because the logistic regression doesn't works very well. The accuracy is very low.

K-class:The number of example is 1593. It is linear input. We can see the accuracy is very high.

Conclusion: For logistic regression, it is not good at non-linear inputs. For a good result , we need a lot of training data to train the algorithm.

5. Reference

The Elements of Statistical Learning ,Second Edition ,Trevor Hastie

http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

http://en.wikipedia.org/wiki/Multilayer_perceptron

http://deeplearning.net/tutorial/mlp.html