Machine Learning Lab3

Neural Networks Learning

handwritten digits recognition

Prelab

Know and understand the THE MNIST DATABASE of handwritten digits

http://yann.lecun.com/exdb/mnist/

read the material or watch the video on cousera of neural networks by Andrew Ng

https://www.coursera.org/learn/machine-learning/resources/EcbzQ

Learn the basic idea of back propagation and try to derive the conclusion

https://hawktom.github.io/BP/

A good website to explain what neural network and deep learning actually is and what they actually work

http://neuralnetworksanddeeplearning.com/

Prelab

1.Give a brief description of neural network including basic mathematic knowledge, important branches, algorithms and models.

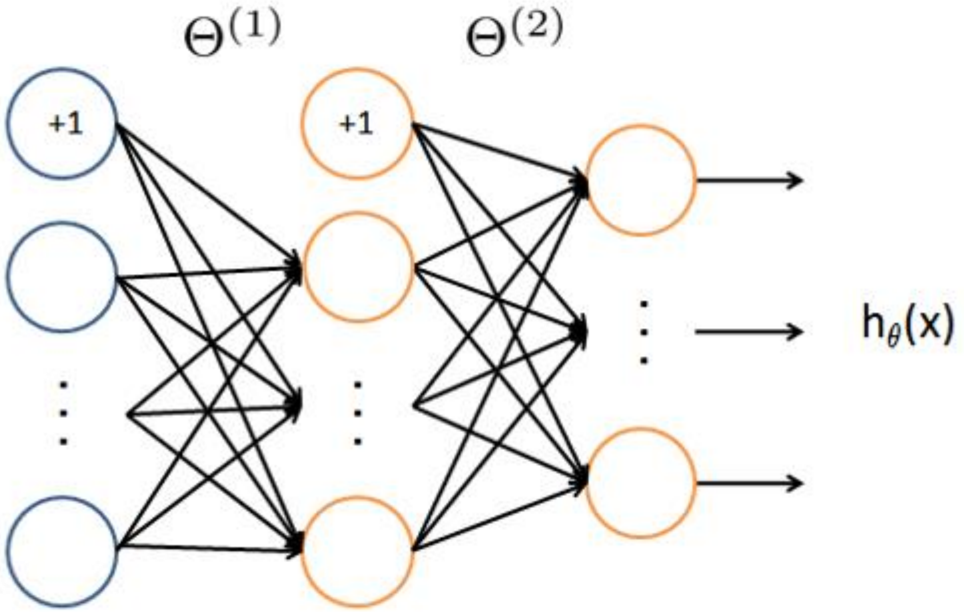
2.Try to explain clearly what BP (back propagation) is and give a conclusion of the derivation process.

Model Interpretation

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log((h_\theta(x^{(i)}))_k) - (1-y_k^{(i)})\log(1-(h_\theta(x^{(i)}))_k)\right] +$$
$$\frac{\lambda}{2m}\left[\sum_{j=1}^{25}\sum_{k=1}^{400}(\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{10}\sum_{k=1}^{25}(\Theta_{j,k}^{(2)})^2\right].$$

So, how we use this cost function

$$\frac{\partial J(\Theta)}{\partial \Theta_{i,j}^{(l)}} \qquad + \lambda\Theta_{i,j}^{(l)}$$



$a^{(1)} = x$ (add $a_0^{(1)}$)

$z^{(2)} = \Theta^{(1)}a^{(1)}$
$a^{(2)} = g(z^{(2)})$ (add $a_0^{(2)}$)

$z^{(3)} = \Theta^{(2)}a^{(2)}$
$a^{(3)} = g(z^{(3)}) = h_\theta(x)$

BP Algorithm

$$\frac{\partial J(\theta)}{\partial \theta^{(L-1)}} = \frac{\partial J(\theta)}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial z^{(L)}}{\partial \theta^{(L-1)}}$$

$$\delta^{(L)} = \frac{\partial J(\theta)}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \qquad \delta^{(L)} = a^{(L)} - y$$

$$\delta^{(L-1)} = \frac{\partial J(\theta)}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \qquad \delta^{(L-1)} = \delta^{(L)} \theta^{(L-1)} a^{(L-1)} (1 - a^{(L-1)})$$

$$\frac{\partial J(\theta)}{\partial \theta^{(L-1)}} = \delta^{(L)} \frac{\partial z^{(L)}}{\partial \theta^{(L-1)}}$$

$$\frac{\partial J(\theta)}{\partial \theta^{(L-1)}} = (a^{(L)} - y)(a^{(L-1)})$$

$$\frac{\partial J(\theta)}{\partial \theta^{(L-2)}} = \delta^{(L-1)} \frac{\partial z^{(L-1)}}{\partial \theta^{(L-2)}}$$

$$\frac{\partial J(\theta)}{\partial \theta^{(L-2)}} = (\delta^{(L)} \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}})(a^{(L-2)})$$

$$\frac{\partial J(\theta)}{\partial \theta^{(L-2)}} = ((a^{(L)} - y)(\theta^{(L-1)})(a^{(L-1)}(1 - a^{(L-1)})))(a^{(L-2)})$$

Training a Neural Network

1. Randomly initialize the weights
2. Implement forward propagation to get $h_\theta(x^{(i)})$
3. Implement the cost function
4. Implement backpropagation to compute partial derivatives
5. Use gradient checking to confirm that your backpropagation works. Then disable gradient checking.
6. Use gradient descent or a built-in optimization function to minimize the cost function with the weights in theta.

Dataset

- The data files train.csv and test.csv contain gray-scale images of hand-drawn digits, from zero through nine.
- Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. This pixel-value is an integer between 0 and 255, inclusive.
- The training data set, (train.csv), has 785 columns. The first column, called "label", is the digit that was drawn by the user. The rest of the columns contain the pixel-values of the associated image
- For more details:

https://www.kaggle.com/c/digit-recognizer

Lab requirement

- Using the supplied dataset to design a 3-layer neural network(one input layer, one hidden layer, one output layer). It’s allowable to use some tools like scipy to process the optimization function, but it’s not allowed to use a framework to get the result directly.

* To optimize the classifier with more data on MNIST

* Try to train a networks using tools such as tensorflow.

Submit

- Prelab
- Python code of the network
- A text document of clear explanation

Sakai until 11.6   22:00