Neural Network and Deep Learning

Qingyun Li

July 7, 2018

1 Binary Classification

We have a example of a binary classification problem, an image is an input and we output a label to recognize this image as being a cat, in which case we output 1, or not-cat which case you output 0, we're going to use y to denote the output label.

2 Logistic Regression

This is a learning algorithm that we use when the output labels y in a supervised learning problem are all either zero or one, so far binary classification problems.

Given x, we want

$$\hat{y} = \sigma(w^T x + b) \tag{1}$$

where $w \in R^{n_x}$, $b \in R$, $\sigma(z) = \frac{1}{1+e^{-z}}$.

When we implement logistic regression, our job is to try to learn parameters w and b, so that $\hat{y}$ becomes a good estimate.

3 Logistic Regression cost function

First, we defined a loss function to measure how good our output $\hat{y}$ is when the true label is y

$$L(\hat{y}, y) = \frac{(\hat{y} - y)^2}{2} \tag{2}$$

But it makes gradient descent not work well, so we define a new loss function as follows:

$$L(\hat{y}, y) = -(ylog\hat{y} + (1-y)log(1-\hat{y})) \tag{3}$$

The loss function was defined with respect to a single training example, it measures how well you're doing on a single training example.

The cost function measures how well you're doing an entire training set:

$$J(w, b) = \frac{1}{m}\sum_{i-1}^{m} L(\hat{y}^{(i)}, y^{(i)}) \tag{4}$$

The cost function is the cost of our parameters, so in training logistic regression model, we're going to try to find parameters w and b that minimize the overall cost function J written at the bottom.