neural network

Neural network is an algorithm that learns and generate its own features from the input and use those features to generate more features until it reaches the end and output the conclusion. The first layer is called the input layer and the last layer is called the output layer.

forward propagation

Forward propagation is like the hypothesis function. It is used to generate a prediction given a set of parameters to optimize the parameters.

{

Th is algorithm works by calculating the activation values from the input layer to output layer using the activation values from the previous activation values, parameter, and the activation function g(z), which is the sigmoid function if we use the neural network to solve classification problem.

cost function

First, we will define the cost function for each output node. Since we will be talking about a classification problem, we will be using cross-entropy, or log loss.

Then, we will define the cost for a training example by summing all the costs of the output nodes.

After that, we will define the cost for a training set by taking the mean.

Lastly, add the regularization term for all parameters.

backpropagation

Let’s define the error of each node, delta.

For the last layer:

For other layers:

To find the partial derivatives of all parameters for a training example:

Derivative of sigmoid function:

Backpropagation algorithm:

random initialization

In neural networks, if we initialize all parameters as zero, the whole network will not learn as all nodes will update to the same value when we backpropagate. Therefore, we will randomly assign a number for each parameter.

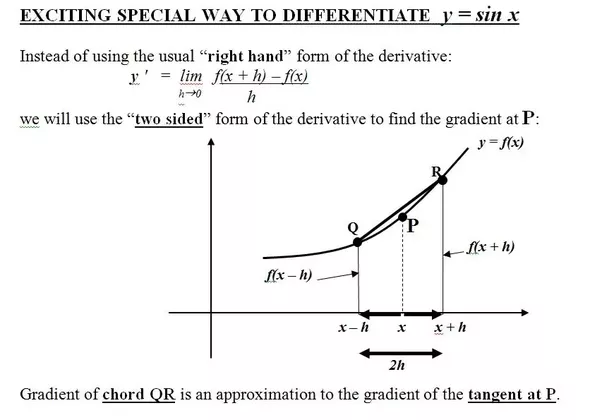
for i = 1 : L - 1

theta(i) = rand(n(i), n(i + 1)) .\* 200 .- 100;

endfor

% rand(n, m) creates a matrix of size n by m with random values from 0 to 1.

gradient checking



If there were only 2 layers, we can approximate the derivative of the cost function with respect to the parameter matrix using the following method.

Otherwise, we can compute it like this:

final algorithm

1. random initialization

2. forward propagation

3. use backpropagation to calculate the partial derivatives

4. use gradient checking to check if the backpropagation code is bug-free

5. gradient descent by repeatedly doing forward propagation and backpropagation

6. classify new examples