Chapter 1 Neural Network: Representation

1.1 Neural Network's Motivation

Neural Network has appeared for a long time and become state-of-the-art technique recently.

1.1.1 Non-linear Hypotheses

In non-linear classification, there could be a LOT of parameters or features.

For example, a image classifier get input in a matrix form. We need to allocate some certain pixels to a space, then classify the sample space. If a image has a size of 50×50 then the quadratic features could be $(2500)^2/2 \approx 3 \times 10^6$. So, this kind of classifier could have realy lots of features.

1.1.2 Neurons and the Brain

We will get to know some background of NNs, and have a sense of what they do.

The NN algorithms are aimed to mimic the brain's work. Though it could do many many things like math, writings, only "the one learning algorithm" is needed.

1.1.3 Model Representation

We will learn how to represent hypothesis in NN model.

Features are input, the hypothesis is output, and we need something between them. Input sometimes gets an extra **bias**. The previous θ is called parameters.

The NN is just putting every features strong together. Normally, the first layer is called input layer, the last is called output layer and the else are called the hidden layers.

Let's define some symbols:

- $a_i^{(j)}$: activation of unit i in layer j
- $\Theta^{(j)}$: matrix of weights controlling functions mapping from layer j to layer $j+1$

For example, if layer 1 gets 2 input and the layer 2 gets 4 input, the $\Theta^{(1)}$ will have a size of 4×3. Then let's vectorize the representation. That is[1]:

$$X^{(j+1)} = g\left(\Theta^{(i)} \times \begin{bmatrix} bias^j \\ X^j \end{bmatrix}\right)$$

The architectures are how the neurons are connected.

[1] by default, bias is set to 1

1.2 Applications of NNs

1.2.1 XOR or XNOR with NNs

Word

resurgence

cortex

dendrite