machine learning_week4

Created:	18/11/11 06:04
Author:	2440027575@qq.com

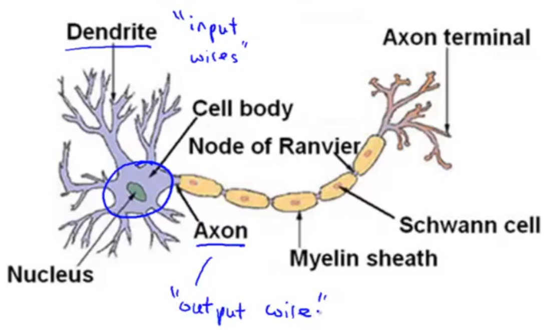
Neural Network

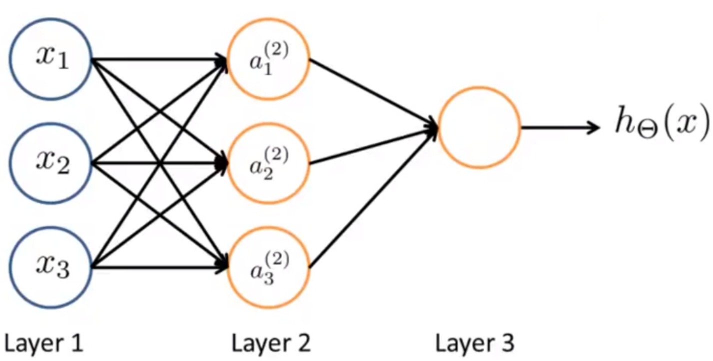
preface

why do we have to talk about neural network?
Technically speaking, beacause when facing non-linear situation with many attrubutes, neither linear regression nor logistic regression has a satisfied behaviour.
However, more importantly neural network is related to a dream: **understand and even imitate human's brain. **This may be the way led to real artificial intelligence.**

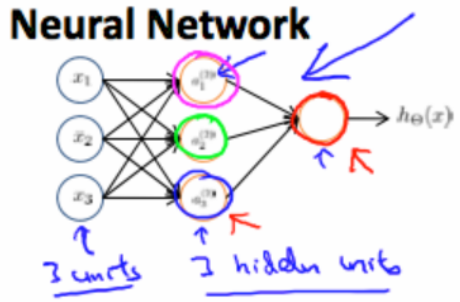
> example from video:
> \>1. the auditory cortex will learn.
> \>2. BrainPort undergoing FDA trials now to help blind people see.
> \>3. the Haptic Belt

Model Representation

- natural neuron & logical simplified neuron
  - 

  - 

(neural network with one layer)
- terminology
  - **hidden layer:**we consider layer 1 as input and the last layer as output (in above pic is layer 3), Then the hidden layer are those layer between input and output.
    - **activation units:**nodes in hidden layer
  - **weights:**"theta" parameters.
  - **bias unit**: X_0 =1usually doesn't present in pic.
  - sigmoid (logistic) **activation** function
  - **forward propagation**
- denotation:
  - see an example



$\rightarrow a_i^{(j)} =$ "activation" of unit i in layer j

$\rightarrow \Theta^{(j)} =$ matrix of weights controlling function mapping from layer j to layer $j+1$

$\Theta^{(1)} \in \mathbb{R}^{3\times4}$ $\quad h_\Theta(x)$

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})$$

$\Theta^{(2)}$

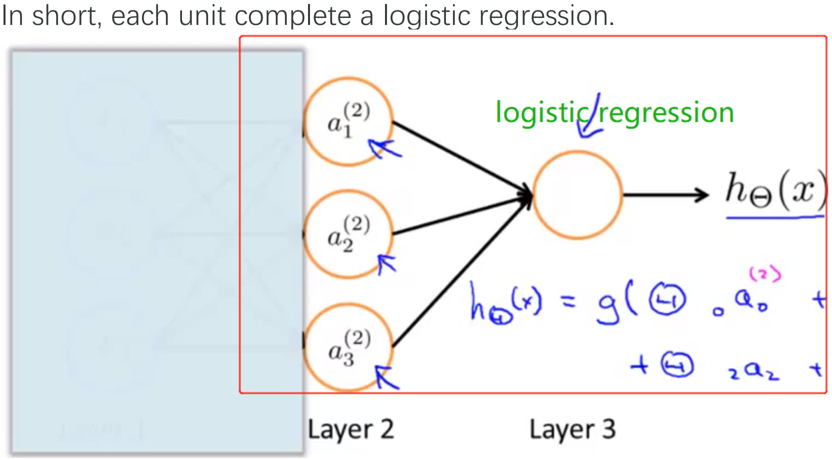
  - subscript *i* denote the *i th* unit while superscript *j* denote the *j th* layer.
- dimension of theta
  - Each layer gets its own matrix of weights, $\Theta^{(j)}$.

    The dimensions of these matrices of weights is determined as follows:

    If network has s_j units in layer j and s_{j+1} units in layer $j+1$, then $\Theta^{(j)}$ will be of dimension $\boxed{s_{j+1} \times (s_j + 1)}$.

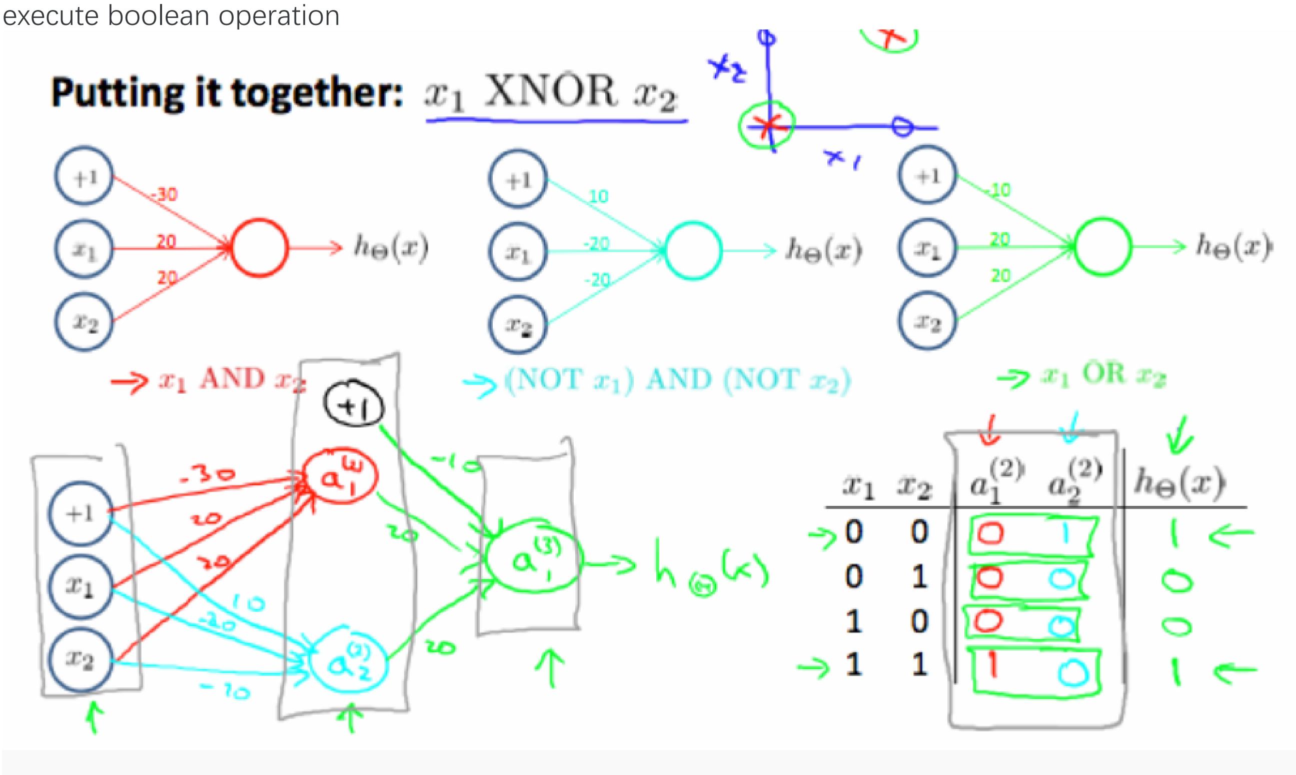
- **vectorlize**
  - still use the example above (see the pic)
  - make $a^{(j)}$ as a vector which denote $a^{(j)} = g(z^{(j)})$ e.g, $a_1^{(2)} = g(z_1^{(2)})$, $a_2^{(2)} = g(z_2^{(2)})$, $a_3^{(2)} = g(z_3^{(2)})$
  - g() means the sigmoid activation function.
  - to compute $z^{(j)}$, we vectorlize it in this way:
    - $$z^{(j)} = \Theta^{(j-1)}a^{(j-1)}$$
    - example :
    - In other words, for layer j=2 and node k, the variable z will be:

      $$z_k^{(2)} = \Theta_{k,0}^{(1)}x_0 + \Theta_{k,1}^{(1)}x_1 + \cdots + \Theta_{k,n}^{(1)}x_n$$

- how it work:
  - In short, each unit complete a logistic regression.
  - 

    - it's really amazing how the neural network can imitate human's decision-making way. It wonderfully abstract human judgement-process (comparing each factors and finally get a conclusion) by computing units in network.

Application:

- **intuition:**
    - execute boolean operation
    - 

- **multi-class classification**: use OvA (one versus all) strategy, set more output units.
        - see here
        -