The Second Week Report

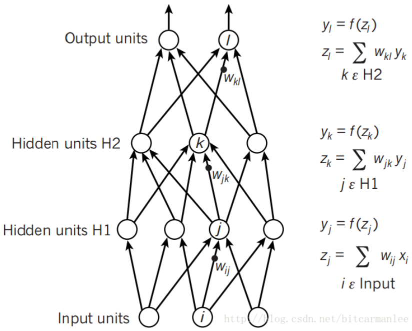
Lu Guorui

2018.6.3

Contents

1 Deep Neural network

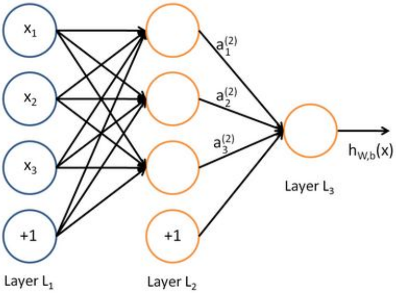
1.1 Forward propagation


Output units
Hidden units H2
Hidden units H1
Input units
$y_l = f(z_l)$
$z_l = \sum_{k \in H2} w_{kl} y_k$
$y_k = f(z_k)$
$z_k = \sum_{j \in H1} w_{jk} y_j$
$y_j = f(z_j)$
$z_j = \sum_{i \in Input} w_{ij} x_i$
w_{kl}
w_{jk}
w_{ij}
http://blog.csdn.net/bitcarmanlee

The forward propagation steps can be written as follows:

1. $z^{[l]} = w^{[l]} \cdot a^{[l-1]} + b^{[l]}$
2. $a^{[l]} = g^{[l]}\left(z^{[l]}\right)$
3. vectorize the formulas above:
   (a) $Z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}$
   (b) $A^{[l]} = g^{[l]}(Z^{[l]})$

1.2 Backward propagation



propagation.jpg

The backward propagation steps can be written as follows:

1. $dz^{[l]} = da^{[l]} * g^{[l]'}(z^{[l]})$
2. $dw^{[l]} = dz^{[l]} \cdot a^{[l-1]}$
3. $db^{[l]} = dz^{[l]}$
4. $da^{[l-1]} = w^{[l]T} \cdot dz^{[l]}$
5. $dz^{[l]} = w^{[l+1]T} dz^{[l+1]} \cdot g^{[l]'}(z^{[l]})$
6. vectorize the formulas above:
   (a) $dZ^{[l]} = dA^{[l]} * g^{[l]'}\left(Z^{[l]}\right)$
   (b) $dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$
   (c) $db^{[l]} = \frac{1}{m} np.sum(dz^{[l]}, axis = 1, keepdims = True)$
   (d) $dA^{[l-1]} = W^{[l]T}.dZ^{[l]}$

2 Practical aspects of DeepLearning

2.1 Bias and Variance

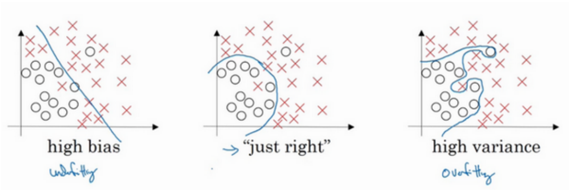
- Bias&Underfitting

  If our logistic regression can't fit the data well, which is the condition of high bias, we call it underfitting.

- Variance&Overfitting

  If out logistic regression fit too much data, which is the condition of high variance, we call it overfitting.

- Just right: If our logistic regression's fitting degree is between the underfitting and overfitting, we call it ”just right”.



2.2 Regularization

Regularization is a method which we used to prevent overfitting. In function $J(w, b)$, we only regularize the parameter m by adding a regularization term $\frac{\lambda}{2m}\sum_{j=1}^{n_x} ||w||_2^2$. And in backward propagation, we also add the term behind the $dw^{[l]}$. So we get the result:

1.

$$J(w, b) = \frac{1}{2m}\sum_{i=1}^{m} L(\hat{y}^{[i]}, y^{[i]}) + \frac{\lambda}{2m}||w||_2^2$$

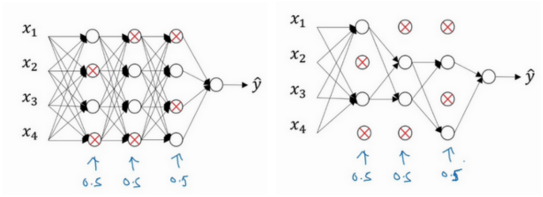
2.

$$dw^{[l]} = (frombackprop) + \frac{\lambda}{2m}w^{[l]}$$

$$\begin{aligned} w^{[l]} &= w^{[l]} - \alpha dw^{[l]} \\ &= w^{[l]} - \frac{\alpha\lambda}{m} - \alpha(frombackprop) \end{aligned}$$

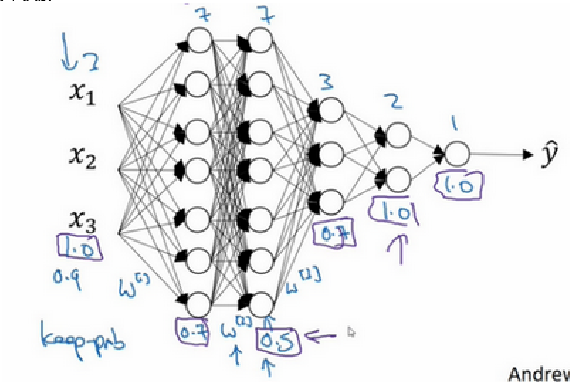
2.3 Dropout Regularization

We use dropout regularization to remove nodes in every layer randomly.



2.3.1 Invert dropout

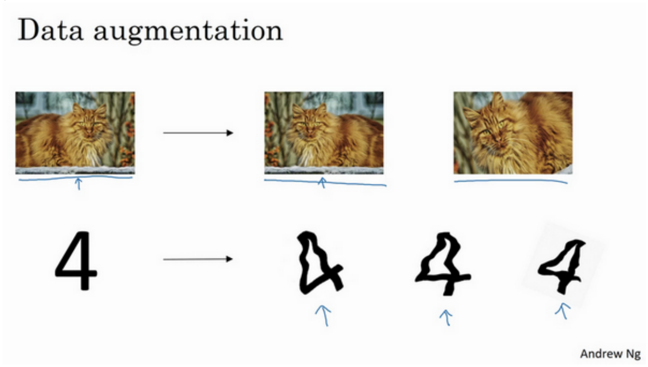
We generate a random matrix $d^{[l]}$, and set the keep-prob value to a certain number which is less than 1.Then we multiply the $d^{[l]}$ and $a^{[l]}$. Because $d^{[l]}$ is a booleans matrix which only contains 1 and 0, the process can set some nodes to 0, which means that this nodes are removed.



2.4 Other regularization methods

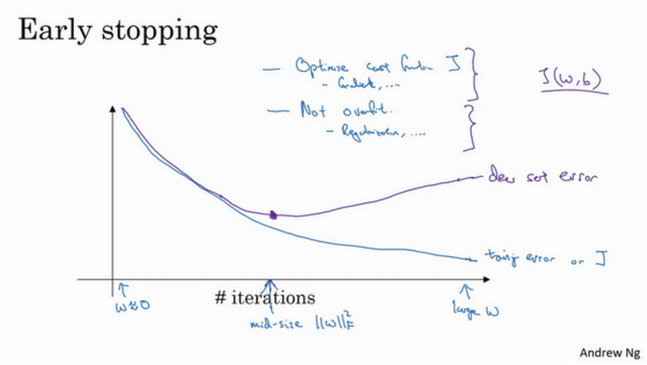
2.4.1 Data Augmentation

By synthesizing examples, we can get more data almost for free. For example, we can flop the photo horizontally, take random crops of the image and imposing rotations and distortions to it so that to gain much more data.



2.4.2 Early Stopping

Early stopping means stop training your neural network early. By doing so we can prevent overfitting to some extent, but when we stop the gradient descent, we stop to optimize the cost function J.



3 Learning Summary

This week I finished the first course and learned some lessons in the second course. The most difficult pari for me is the Regularization for I don't figure out why we are supposed to add a term like this and how does it works on earth. It still remains a lot for me to study.