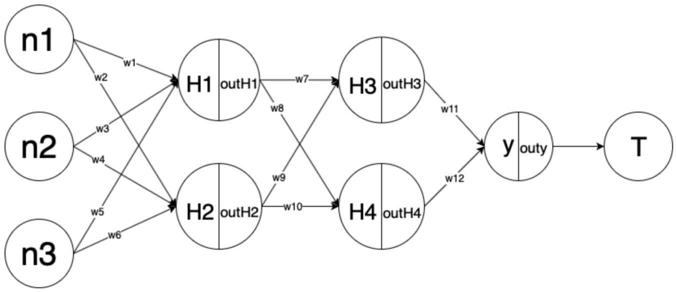
MATH7502 Deep learing

Xiaoqi Zhuang, Ziqing Yan, Zhipei Tao, Dabang Sheng

Deep Neural Networks is a discriminant model that can be trained using backpropagation algorithms. The weight update can be solved by the stochastic gradient descent method. In this project, we bulid two hidden layers. Both the hidden layer has 2 stages.



Backpropagation algorithm is one of two ways to compute the derivatives dF/dx. Backpropagation is the process taking the error and feeding backward to the error though the net work.

Mathematics of gradient descent tells us how to take an error to nudge weight then we calculated the error coming out of the hidden layer and keep going back and that is back propagation and how the hidden errors are calculated.

We assume that $\hat{y}=\beta_0+\beta_1 x$, and the loss function is $L(\beta)=\sum_{j=1}^{N} \frac{1}{N}\left(\beta_0+\beta_1 x_j-y_j\right)^2$.

$$\nabla L=\left(\frac{\partial L}{\partial \beta_0}, \frac{\partial L}{\partial \beta_1}\right)=\left(\frac{2}{N} \sum_{j=1}^{N}\left(\beta_0+\beta_1 x_j-y_j\right), \frac{2}{N} \sum_{j=1}^{N} x_j\left(\beta_0+\beta_1 x_j-y_j\right)\right)$$

This is just a linear function, but not more complicated neural network. So we introduce a new method named SGD. The only different is that it just use one random (x,y) once.

To avoid point oscillating by the random, $m_t=\beta_1 \cdot m_{t-1}+\left(1-\beta_1\right) \cdot g_t$, $\beta_1=0.9, m_0=0$

To let it metric history update frequency, $V_t=\beta_2 * V_{t-1}+\left(1-\beta_2\right) g_t^2$, $\beta_2=0.99, v_0=0$

New iterative formula: $\beta_{n+1}=\beta_n-\frac{\alpha}{\sqrt{v_t}+\epsilon} * m_t$,$\epsilon=10e-8$, which just do not expect divis by 0.

We call this method ADAM and it is very popular in the maching learning.

As we can see from the content above, deep neural networks involves in a large number of calculation. This number could increase rapidly with adding more layers or neural to our network. To solve this issue and reduce the computation, we use convolutional neural network to extract features from the original pictures in order to compress image files, or say our input. Edge detecting and Pooling are the two basic and commonly use convolutional strategies. They could significantly reduce our input size and simultaneously remain the crucial features.