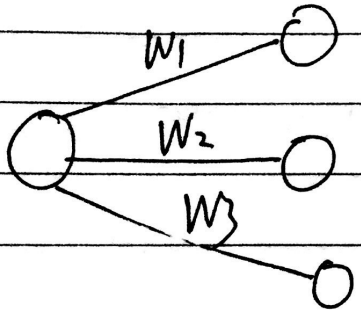
Date

神经网络分类模型小结

1. 首先要需要设计好整体的网络架构。输入层为特征数，输出层为label的个数。重要是hidden-layer的层数以及每一层unit的个数。

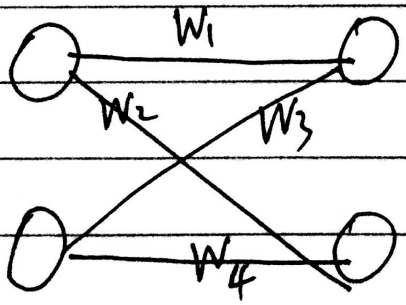
2. 随机initializer各层参数矩阵$\Theta^{(l)}$。为什么不能全部初始为0呢？因为如果$\Theta^{(l)}$全为0，则每个hidden-layer的activation的值全部一样，最后output units全一样。虽然通过BP能更新这些$\Theta_{ij}^{(l)}$，但对于每个neuron的输出边



$$\frac{\partial J(\theta)}{\partial w_1} = \frac{\partial J(\theta)}{\partial w_2} = \frac{\partial J(\theta)}{\partial w_3}$$

这样使每次更新之后 $W_1 = W_2 = W_3$

当再进行 Forward Prop时，由于每个neuron



$W_1 = W_2$，$W_3 = W_4$

导到下一层每个neuron的输入值相同

进而输出值相同。

这样一个neural network模型无论输出全是一致的。

No.

Date.

无法取一个最大值将其划分为一个具体的 class

一般 random initilization 的经验

$\Theta_{ij}^{(l)} \in [-\varepsilon, +\varepsilon)$ $\varepsilon = \frac{\sqrt{6}}{\sqrt{\text{input\_unit}} \times \sqrt{\text{output\_unit}}}$

input_unit：第 l 层 neuron 个数

output_unit：第 (l+1) 层 neuron 个数

3. Forward Prop 计算当前 example 预测值。

第 l 层输出 * $\Theta^{(l)}$ ⇒ 第 (l+1) 层输入

一层一层往后计算，最后计算得出 output = $h_\theta(x)_K$

4. 计算 cost function。将每个 example 的 $h_\theta(x)$ 计算最后累加起来，除上 m。考虑正则化时，加上所有 $\Theta_{ij}^{(l)2}$ 注意，每个 bias 对应的参数不参与正则化。即忽略掉每个 $\Theta^{(l)}$ 所在的第 1 列。

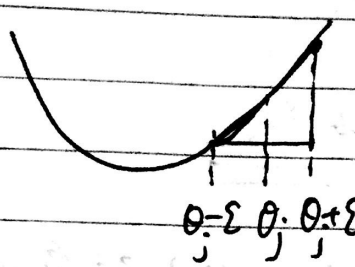
5. BP 来计算 $\frac{\partial J(\theta)}{\partial \Theta_{ij}^{(l)}}$ partial derivitive.

最后一层误差 $\delta^{(L)} = (y - \text{target})$ target 要转换为 binary vector。然后向前计算上一层 $\delta^{(L-1)} = (\Theta^{(L-1)})^T \cdot \delta^{(L)} .* \text{sigmoid}'(z^{(L-1)})$

计算完每一层的 $\delta^{(l)}$，最后用公式

$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} = \delta_i^{(l+1)} a_j^{(l)}$ 得到偏导数。将m个example的偏导加和再除上m得到最终偏导数。注意，考虑正则化时需要加上正则项的单独偏导，加上到总偏导中。

6. gradient checking。要保证通过公式计算的偏导与数值偏导基本相同。



将所 $\Theta^{(l)}$ unroll 为 vector 将 $J(\Theta)$ 看成 Θ 的函数，然后其它 Θ 不变，Θ_j 改变 $\pm\varepsilon$ 得

需要从从相等于 $\Leftarrow \frac{f(\Theta_j+\varepsilon) - f(\Theta_j-\varepsilon)}{2\varepsilon}$

$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}}$ 的值

7. ~~已经使用 Forward~~ 优化函数去优化 $J(\Theta)$

已经计算出3个东西 func的值以及对每个参数的偏导，直接用 fmincg 来优化 cost function 使其值越来越小。最后得到最优 Θ 的值

8. 用上述 Θ 作预测。Forward Prop 然后取最大的值合为该类