



类神经网络训练不起来怎么办

(四) 损失函数Loss也可能有影响

分类。

分类 (classification) 可以看作回归 (Regression)

Regression

$$\begin{array}{c} \text{label} \\ \hat{y} \end{array} \longleftrightarrow \begin{array}{c} y \\ y \end{array} = \begin{array}{c} b \\ b \end{array} + \begin{array}{c} c^T \\ c^T \end{array} \sigma \left(\begin{array}{c} b \\ b \end{array} + \begin{array}{c} W \\ W \end{array} \begin{array}{c} x \\ x \end{array} \right)$$

feature

Classification

$$\begin{array}{c} y \\ y \end{array} = \begin{array}{c} b' \\ b' \end{array} + \begin{array}{c} W' \\ W' \end{array} \sigma \left(\begin{array}{c} b \\ b \end{array} + \begin{array}{c} W \\ W \end{array} \begin{array}{c} x \\ x \end{array} \right)$$

feature

$$\begin{array}{c} \text{label} \\ \hat{y} \end{array} \longleftrightarrow \begin{array}{c} y' \\ y' \end{array} = \text{softmax} \left(\begin{array}{c} y \\ y \end{array} \right)$$

0 or 1 Make all values between 0 and 1 Can have any value

6

在这里，softmax可以理解为归一化。

下面是softmax的运作示意图：

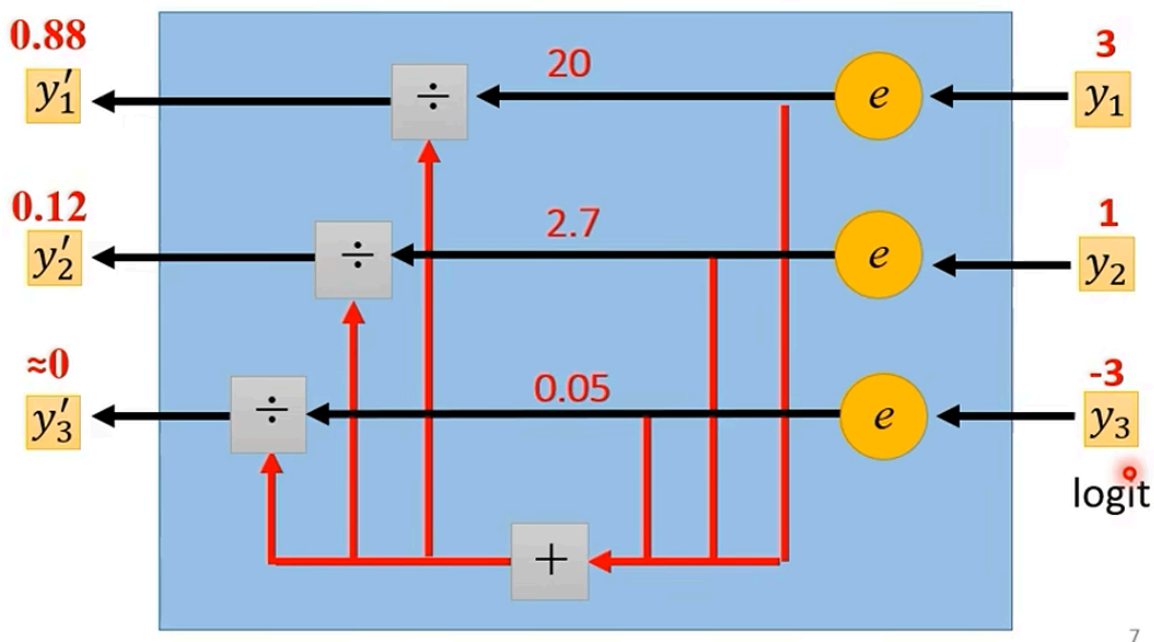
Soft-max

$$y'_i = \frac{\exp(y_i)}{\sum_j \exp(y_i)}$$

- $1 > y'_i > 0$
- $\sum_i y'_i = 1$

Softmax

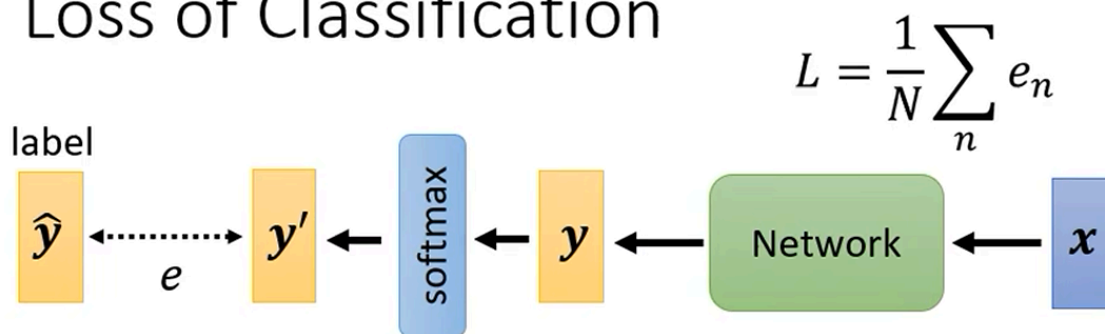
How about **binary classification**? 😊



可以看到，是每一个数求exp然后求百分比的一个结果。这里的输入叫logit。

可以看到上图是三个分类，特殊的对于俩个分类，可以用sigmoid，二者没区别。

Loss of Classification



Mean Square Error (MSE) $e = \sum_i (\hat{y}_i - y'_i)^2$

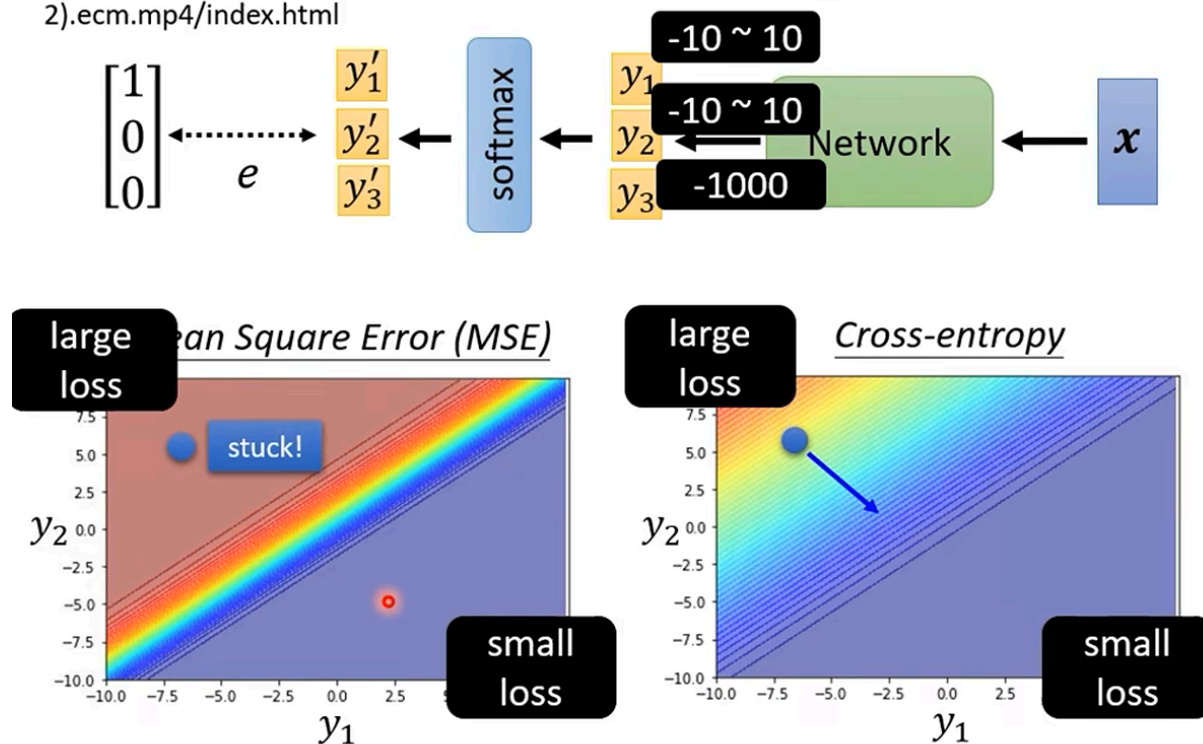
Cross-entropy $e = - \sum_i \hat{y}_i \ln y'_i$

Minimizing cross-entropy is equivalent to maximizing likelihood.

这里的损失函数可以用MSE，也可以用Cross-entropy（交叉熵）。但是交叉熵更多的用在分类中。Cross-entropy和softmax绑定的，当使用Cross-entropy，pytorch自动将softmax加到NetWork的最后一层。

Cross-entropy其实是和maximizing likelihood是一样的。

[http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/Deep%20More%20\(v2\).ecm.mp4/index.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/Deep%20More%20(v2).ecm.mp4/index.html)



Changing the loss function can change the difficulty of optimization.

上图解释了为什么在分类中更多的使用交叉熵。可以看到，在左上角loss较大的部分，交叉熵是有梯度的，可以进行梯度下降，然而MSE在loss较大的部分，是很平坦的，梯度不变，很难进行优化，会卡住。