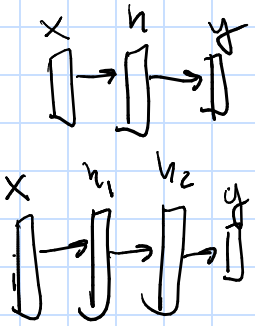
Neural Nets III (continued)

$$w(t+1) = w(t) + \Delta w(t)$$

$$\begin{pmatrix}\text{weight correction for } w_{ji}(n)\end{pmatrix} = \eta \begin{pmatrix}\text{local gradient } \delta_j(n)\end{pmatrix} \cdot \begin{pmatrix}\text{Input signal } y_i(n)\end{pmatrix}$$

$\eta > 0$



Local gradients

Output Node *

$$\delta_j(n) = e_j(n)\,\varphi_j'(v_j(n))$$

Hidden Node

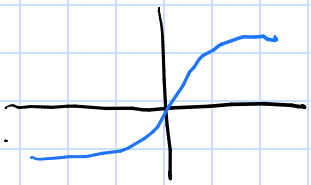
$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n)\, w_{kj}(n)$$

previous

Activation functions

sigmoid

$$\varphi(x) = \frac{1}{1-e^{-x}}$$



tan sigmoid

relu

leaky relu

$$\varphi'(x) = \varphi(x)(1-\varphi(x)) \leq \frac{1}{4}$$

The error signal can vanish as we propagate the error through a deep net.

Softmax → multiclass

Regression → MSE loss

Classification → MC → categorical crossentropy

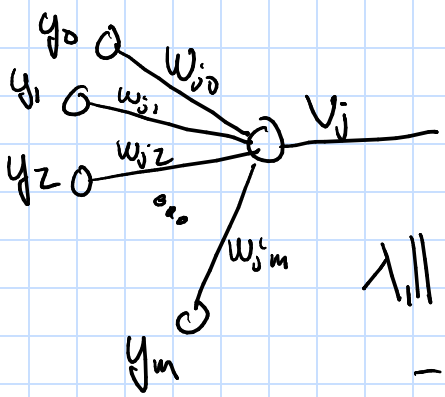
→ BC → binary cross entropy

- Pytorch
- Tensorflow
- Keras

Regularization

① L_1 $\quad \|W\|_1$

② L_2 $\quad \|W\|_2^2$



$\lambda_1 \|W_{j:}\|_1$

— or —

$\lambda_2 \|W_{j:}\|_2^2$

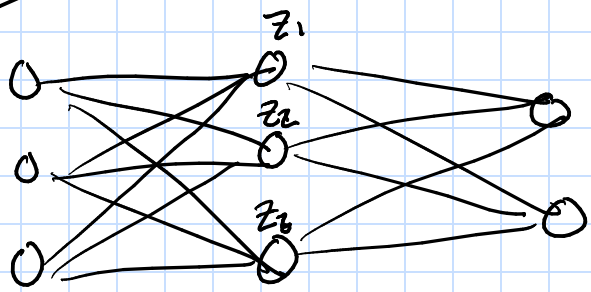
$$\min \; E(\theta) + \lambda \Omega(\theta)$$

↳ L_1

↳ L_2

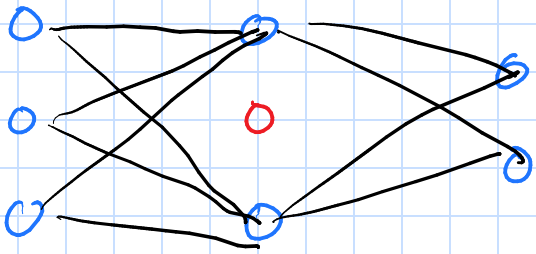
$$\frac{\partial}{\partial \theta}\{E(\theta) + \lambda \Omega(\theta)\} = \boxed{\frac{\partial}{\partial \theta} E(\theta)} + \lambda \frac{\partial}{\partial \theta} \Omega(\theta)$$

Drop out

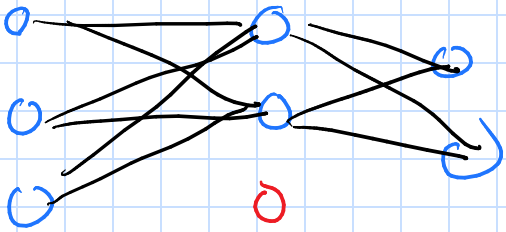


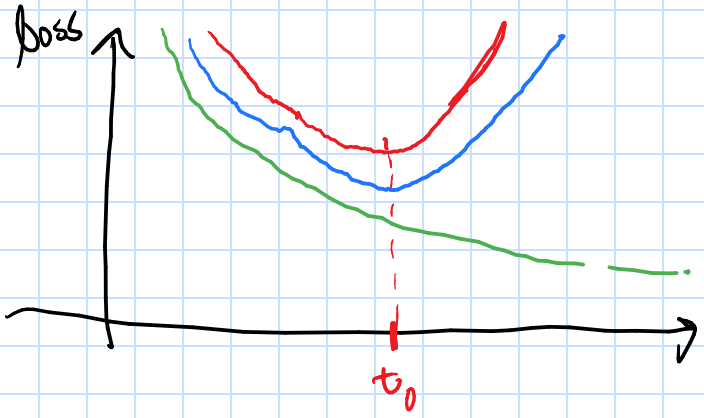
$z \sim \text{Bern}(p)$

$z_1 = 1, \; z_2 = 0, \; z_3 = 1$



$z_1 = 1, \; z_2 = 1, \; z_3 = 0$





training
testing
validation