Machine Learning INF2008

Lecture 03: The Neural Network

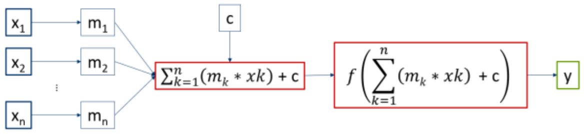
Donny Soh

Singapore Institute of Technology

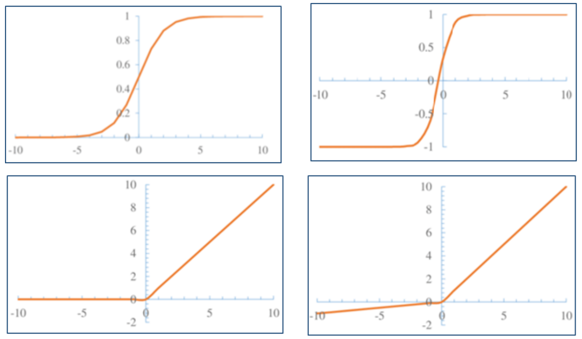
What are the steps in training a single layer perceptron?

How does learning take place in a NN?

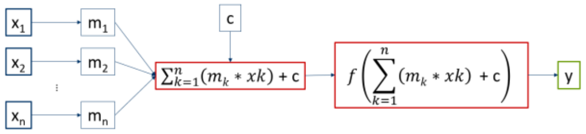
Forward



Activation Functions



Loss Calculation



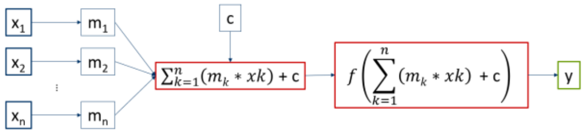
MSE

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2$$

Cross Entropy Loss

$$L_{CE} = -\sum_{i=0}^{n} t_i * log(p_i)$$

Backpropogation



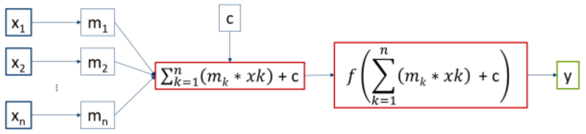
$$Z = A.W^T + \iota.b^T$$

$$\frac{\partial L}{\partial A} = \left(\frac{\partial L}{\partial Z}\right)\left(\frac{\partial Z}{\partial A}\right)^T$$

$$\frac{\partial L}{\partial W} = \left(\frac{\partial L}{\partial Z}\right)^T \cdot \left(\frac{\partial Z}{\partial W}\right)$$

$$\frac{\partial L}{\partial b} = \left(\frac{\partial L}{\partial Z}\right)^T \cdot \left(\frac{\partial Z}{\partial b}\right)$$

Gradient Descent



SGD Equation (Without Momentum)

$$W = W - \lambda\frac{\partial L}{\partial W}$$

$$b = b - \lambda\frac{\partial L}{\partial b}$$

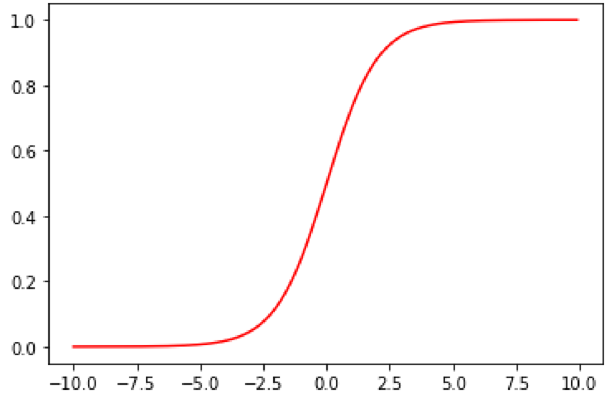
SGD Equation (With Momentum)

$$v_w = \mu.v_w + \frac{\partial L}{\partial W}$$

$$v_b = \mu.v_b + \frac{\partial L}{\partial b}$$

Activation Functions

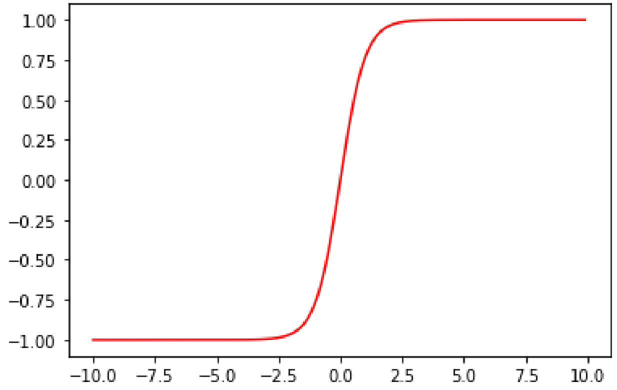
Sigmoid



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

- Sigmoid / tanh: binary classification problems / regression.
- Relu / Leaky Relu: NN internal layers.
- Softmax: multiclass classification problems.

Tanh

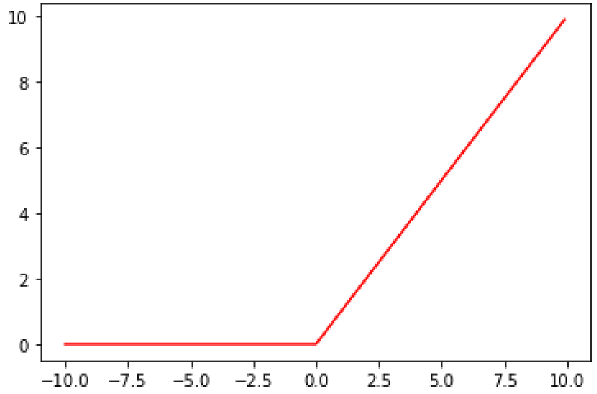


$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

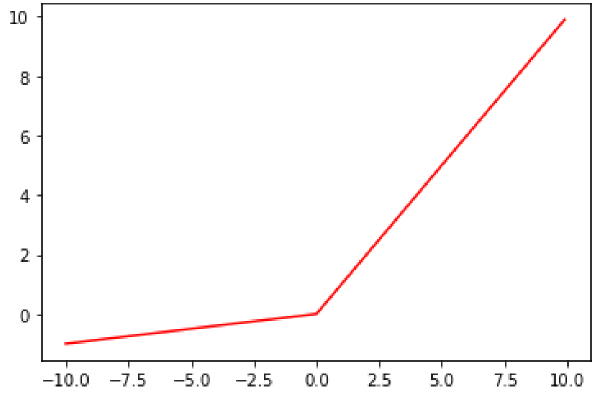
$$f(x) = \frac{\sinh(x)}{\cosh(x)}$$

$$f(x) = \tanh(x)$$

Relu / Leaky Relu

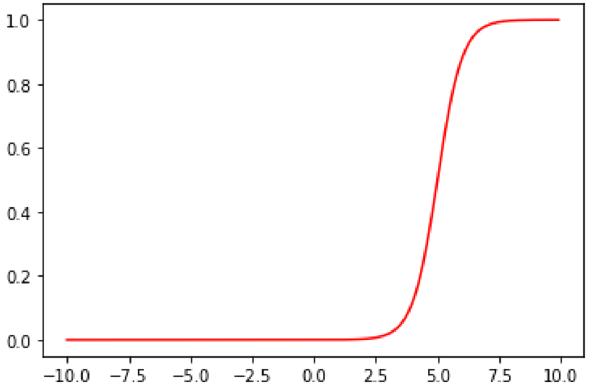


$$f(x) = \begin{cases} x & otherwise \\ 0 & for\ x < 0 \end{cases}$$



$$f(x) = \begin{cases} x & otherwise \\ \alpha * x & for\ x < 0 \end{cases}$$

Softmax



The graph is in the event for two outputs.

$$f(x_i) = \frac{e^{x_i}}{\sum e^{x_j}}$$

$$f(x_i) = \frac{e^{x_1}}{e^{x_1} + e^{x_2}}$$

$$f(x_i) = \frac{1}{1 + e^{x_2 - x_1}}$$

Loss Contribution

Cross Entropy Loss (classification)

$$L_{CE} = -\sum_{i=0}^{n} t_i * log(p_i)$$

- t_i is the truth label for the i^{th} class.
- p_i is the softmax probability for the i^{th} class.

x_i	t_i	p_i	$log(p_i)$	$t_i * log(p_i)$
3	0	6.14207644e-06	-1.73128421e+01	0
15	1	9.99652370e-01	-5.01611506e-04	-5.01611506e-04
7	0	3.35346011e-04	-1.15420619e+01	0
3	0	6.14207644e-06	-1.73128421e+01	0

MSE

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2$$

Backpropogation

$Z = A.W^T + \iota.b^T$

The derivative of Z with respect to A is W^T.
The derivative of Z with respect to W^T is A.
The derivative of Z with respect to W is A^T.

The derivative of Z with respect to b^T is ι.

$$\frac{\partial L}{\partial A} = \left(\frac{\partial L}{\partial Z}\right)\left(\frac{\partial Z}{\partial A}\right)^T \qquad \frac{\partial L}{\partial W} = \left(\frac{\partial L}{\partial Z}\right)^T . \left(\frac{\partial Z}{\partial W}\right) \qquad \frac{\partial L}{\partial b} = \left(\frac{\partial L}{\partial Z}\right)^T . \left(\frac{\partial Z}{\partial b}\right)$$

Stochastic Gradient Descent

$$Z = A.W^T + \iota.b^T$$

SGD Equation (Without Momentum)

$$W = W - \lambda \frac{\partial L}{\partial W}$$

L*: Loss (at a layer)*
λ: Learning Rate

$$b = b - \lambda \frac{\partial L}{\partial b}$$

SGD Equation (With Momentum)

$$v_w = \mu . v_w + \frac{\partial L}{\partial W}$$

$$W = W - \lambda v_w$$

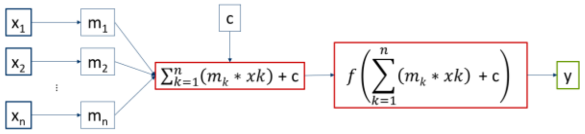
$$v_b = \mu . v_b + \frac{\partial L}{\partial b}$$

$$b = b - \lambda v_b$$

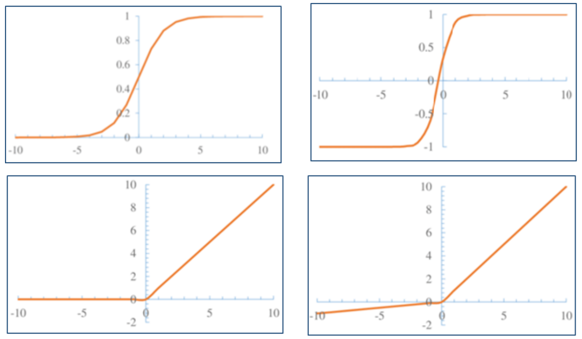
μ: 0.9
λ: 0.01 / 0.001

Putting it all together

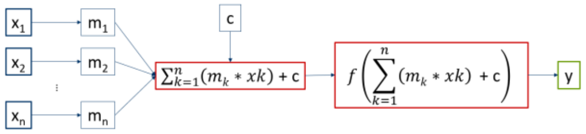
Forward



Activation Functions



Loss Calculation



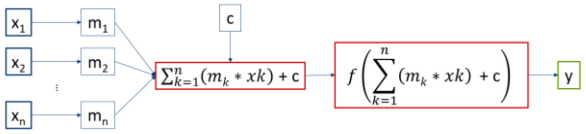
MSE

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2$$

Cross Entropy Loss

$$L_{CE} = -\sum_{i=0}^{n} t_i * log(p_i)$$

Backpropogation



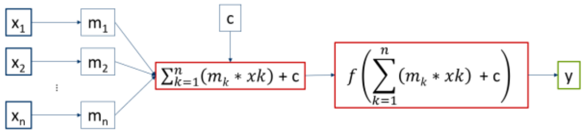
$$Z = A.W^T + \iota.b^T$$

$$\frac{\partial L}{\partial A} = \left(\frac{\partial L}{\partial Z}\right)\left(\frac{\partial Z}{\partial A}\right)^T$$

$$\frac{\partial L}{\partial W} = \left(\frac{\partial L}{\partial Z}\right)^T \cdot \left(\frac{\partial Z}{\partial W}\right)$$

$$\frac{\partial L}{\partial b} = \left(\frac{\partial L}{\partial Z}\right)^T \cdot \left(\frac{\partial Z}{\partial b}\right)$$

Gradient Descent



SGD Equation (Without Momentum)

$$W = W - \lambda\frac{\partial L}{\partial W}$$

$$b = b - \lambda\frac{\partial L}{\partial b}$$

SGD Equation (With Momentum)

$$v_w = \mu.v_w + \frac{\partial L}{\partial W}$$

$$v_b = \mu.v_b + \frac{\partial L}{\partial b}$$