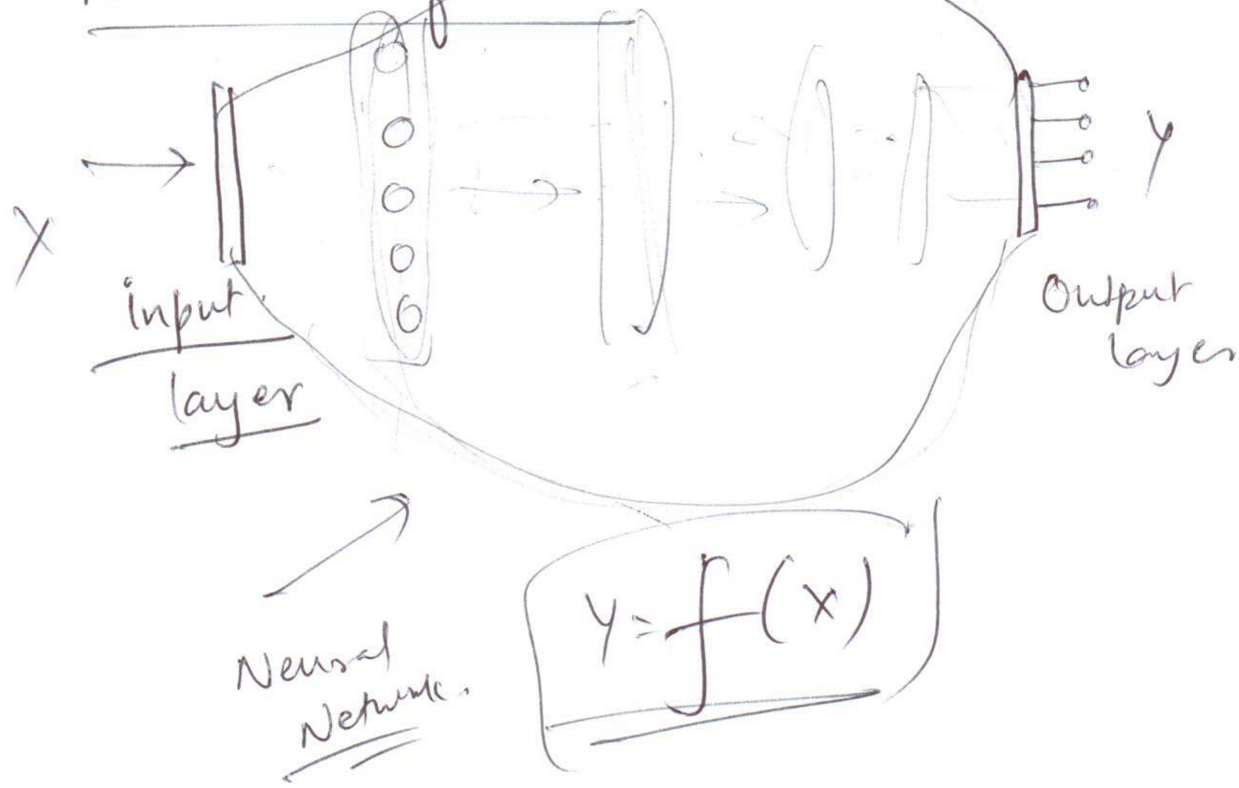
Neural Networks · 03/01

Units.

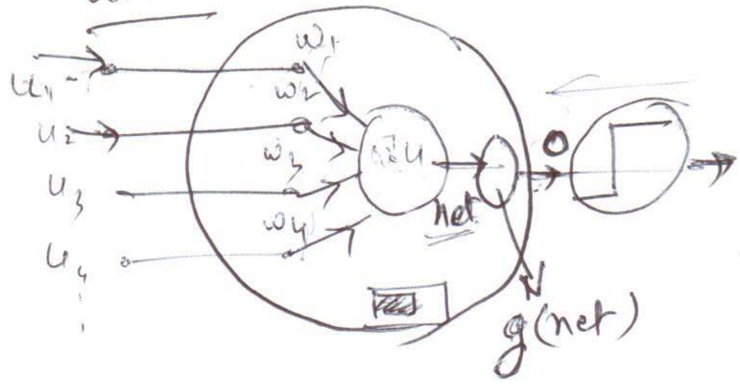
Network of units.

layered architecture.



Universal Approximators.

Unit



~~Perpec~~

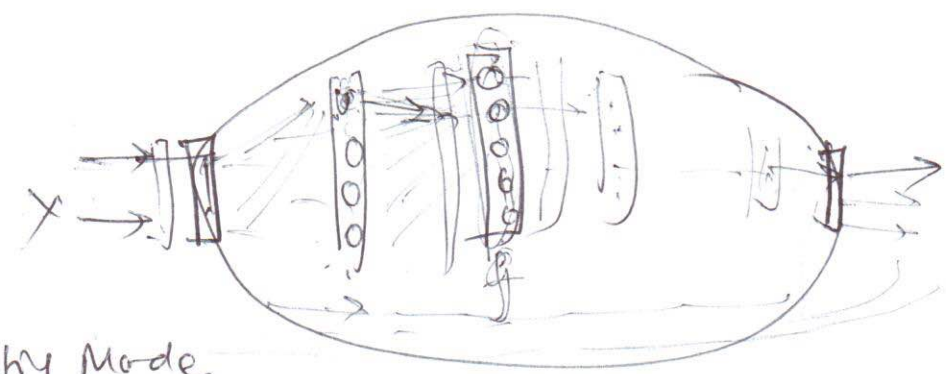
Perceptron $g \to 1(\ )$

Sigmoid $g \to \sigma(\ ) = \frac{1}{1+\exp(-)}$

tanh =

RELU = ( )

Thresholded unit



Testing Mode

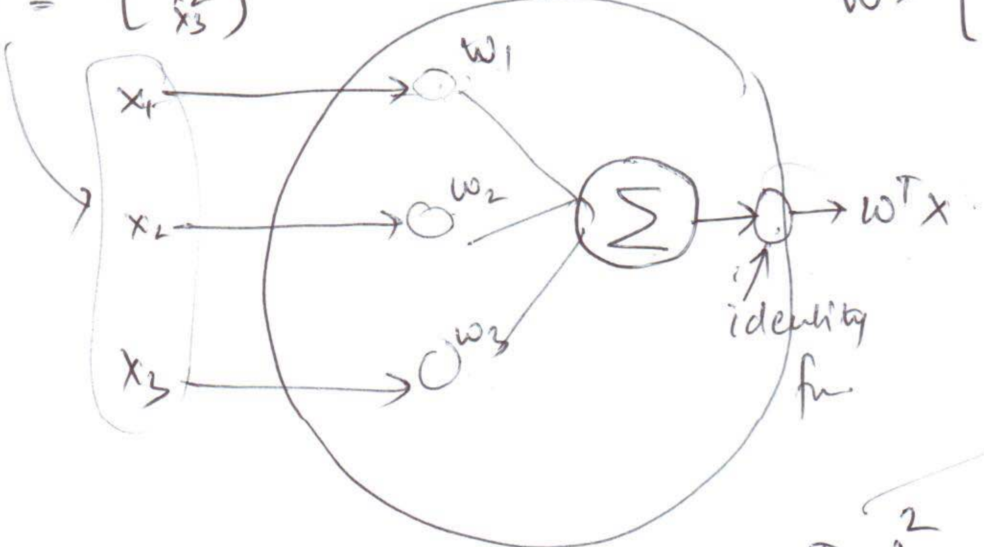
NN

Feed forward Mode

Training Mode

Back Propogation

$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ $\qquad W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$



$$\Sigma = w^T x = w_1 x_1 + w_2 x_2 + w_3 x_3$$

y

$$L(w) = \frac{1}{2}\left(y - w^T x\right)^2$$

We have only one data point in training

$$\frac{1}{2}\sum (y_i - w^T x_i)^2$$

$$= \frac{1}{2}\left(y - w_1 x_1 - w_2 x_2 - w_3 x_3\right)^2$$

$$\frac{\partial L}{\partial w_1} = \left(y - w_1 x_1 - w_2 x_2 - w_3 x_3\right)(-x_1)$$

$$= -(y - w^T x)\, x_1$$

$$\frac{\partial L}{\partial w_2} = -(y - w^T x)\, x_2$$

$$\frac{\partial L}{\partial w_3} = -(y - w^T x)\, x_3$$

$$\nabla = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \frac{\partial L}{\partial w_3} \end{bmatrix}$$

$$w \leftarrow w^{old} - \eta \nabla$$

$$w_1 = w_1^{old} - \eta \frac{\partial L}{\partial w_1}$$

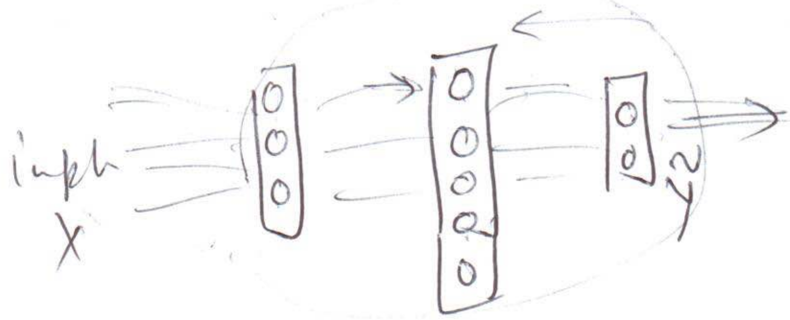
$$= w_1^{old} + \eta (y - w^T x) x_1$$

$$w_2 = w_2^{old} + \eta (y - w^T x) x_2$$

$$w_3 = w_3^{old} + \eta (y - w^T x) x_3$$

→ Error between true label (y) and ($w^T x$) → prediction.

→ learning rate (eta) → step-size

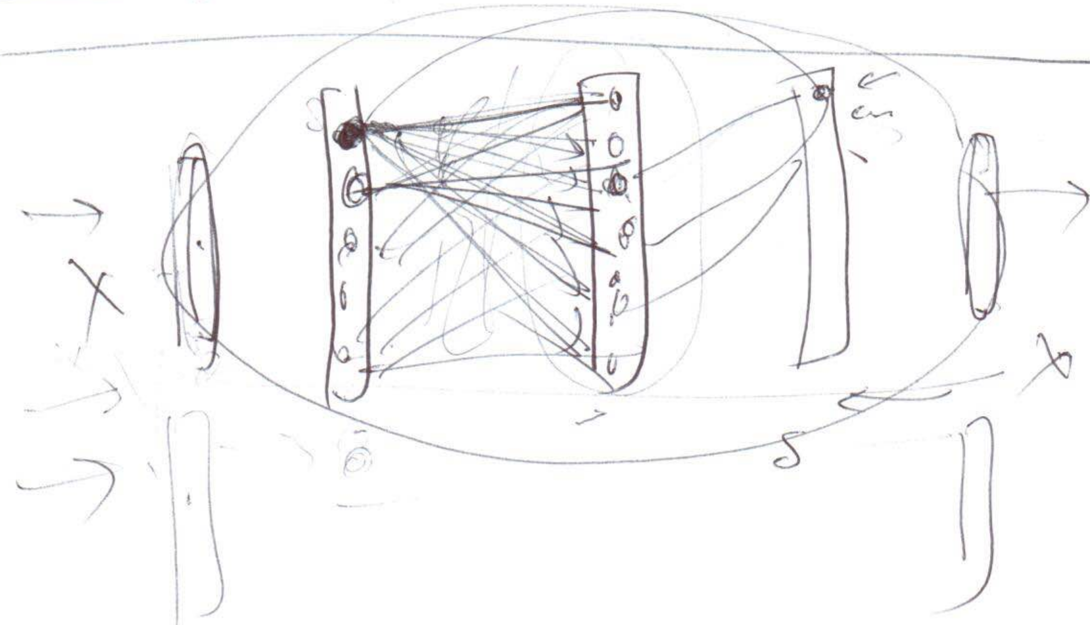


y

$(y - \tilde{y})$

error

Backpropogation

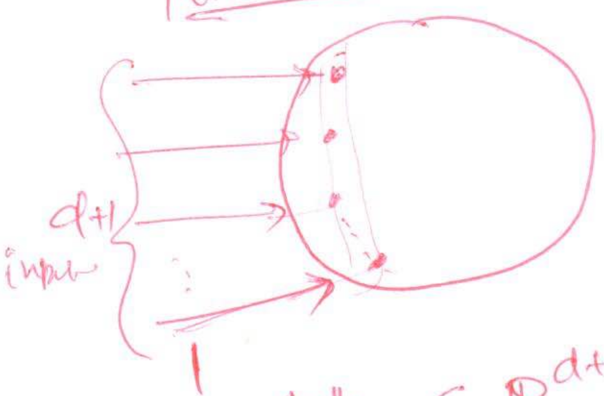


Neural Networks.

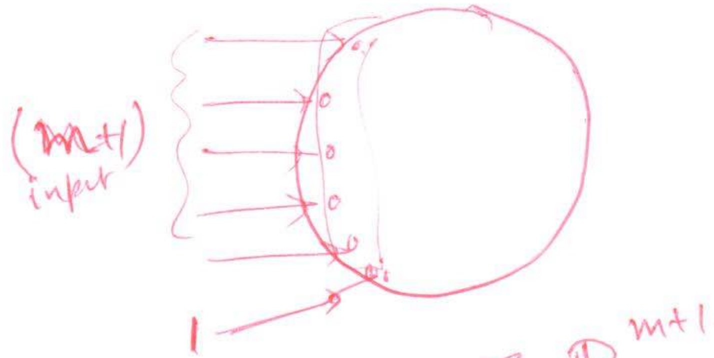
1 hidden layer $\rightarrow$ m units.

1 output layer $\rightarrow$ k units (K classes)

Hidden unit



Output unit



$w_j \in \mathbb{R}^{d+1}$

weight vector for j^{th} hidden unit.

$W_\ell \in \mathbb{R}^{m+1}$

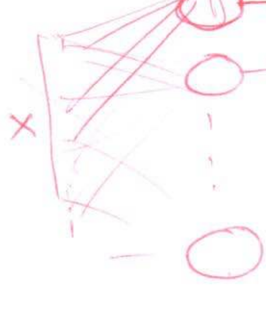
weight vector for the ℓ^{th} output unit.

$1 \leq j \leq m$ total # hidden units.

$1 \leq \ell \leq k$ total # classes

Let x be the input. $x \in \mathbb{R}^d$ (we will add a bias term to this)

After adding bias term $\rightarrow$ $x \in \mathbb{R}^{d+1}$

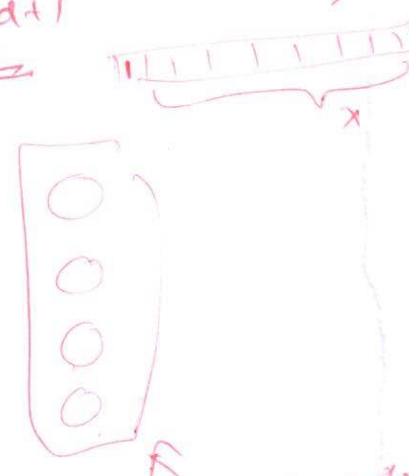


$z_1 = \sigma(w_1^T x)$

$z_2 = \sigma(w_2^T x)$

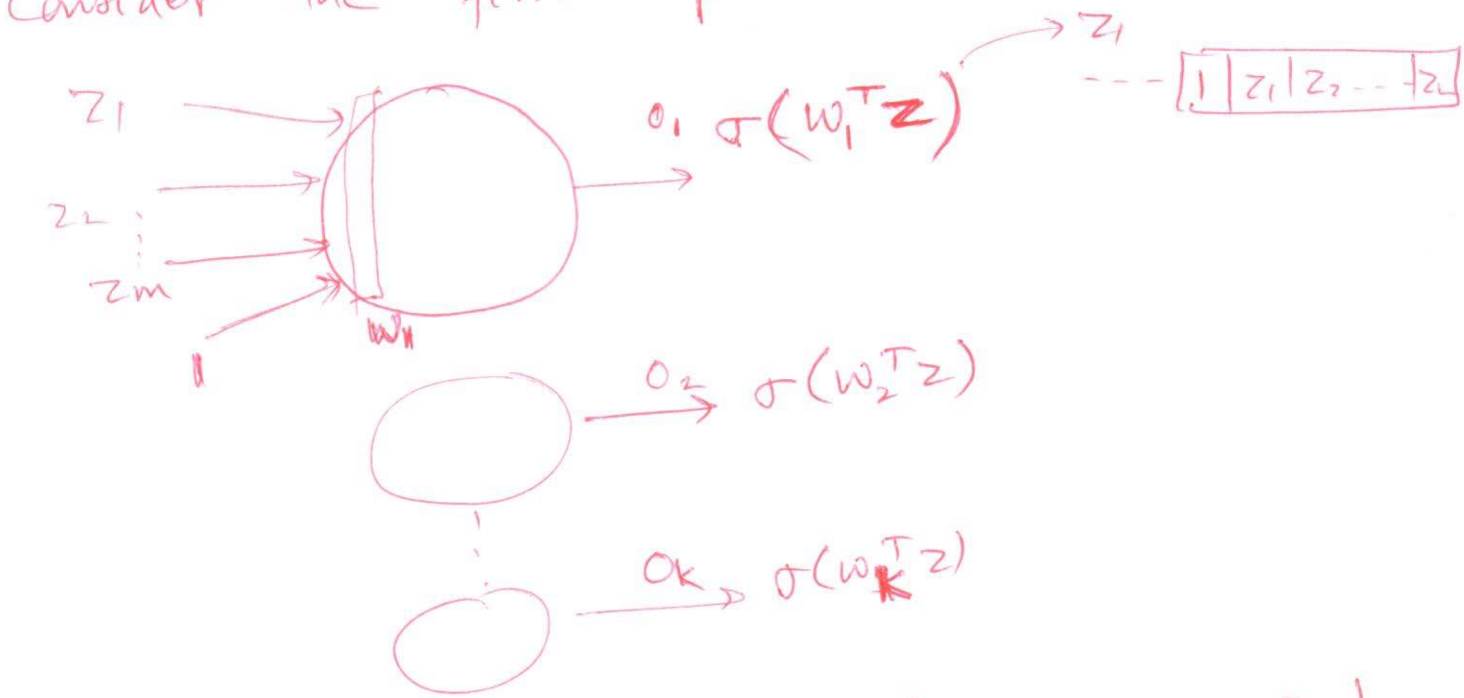
$z_m = \sigma(w_m^T x)$

activation



output nodes

Consider the first output node.



Predicted Class $\rightarrow$ $\underset{k}{\operatorname{argmax}}\,(o_1, o_2, \cdots o_k)$

$$\tilde{o}_1 = \frac{o_1}{\sum o_\ell}$$

---

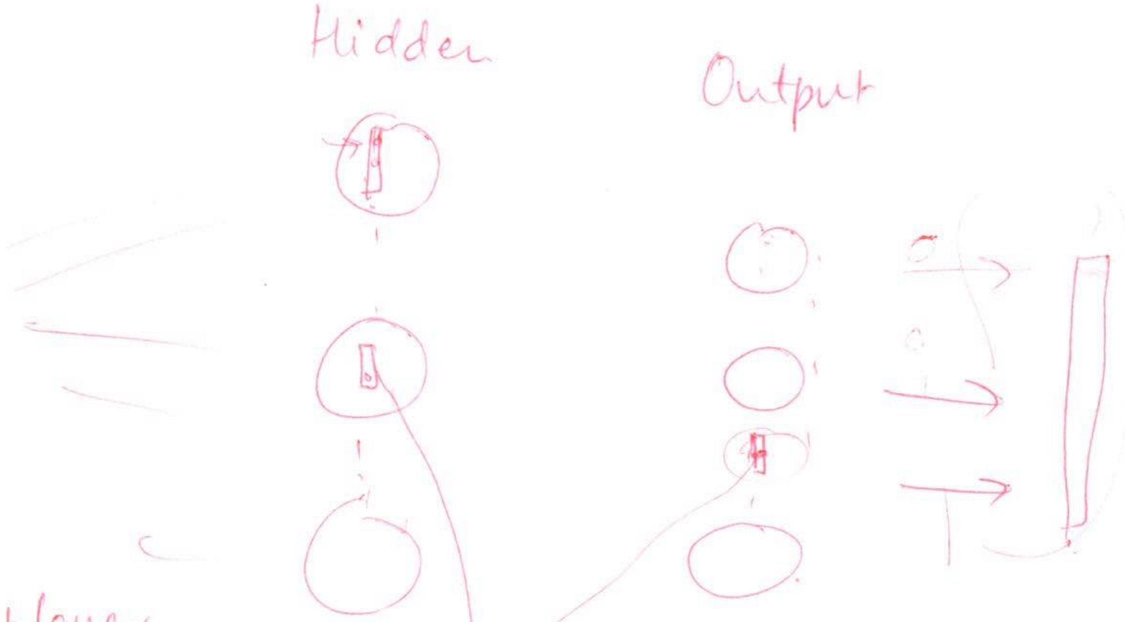
loss function for LinRegression

$$J(w) = \frac{1}{2}\sum\left(y_i - w^T x_i\right)^2$$

Grad Descent

$$W^{(t)} \leftarrow w^{(t-1)} - \eta\, \nabla J(w)$$

$$\boxed{w_i^{(t)} \leftarrow w_i^{(t-1)} - \eta\, \frac{\partial J}{\partial w_i}} \rightarrow$$



At output layer

$$\frac{\partial J}{\partial w_{kj}}$$

At ~~inp~~ hidden layer

$$\frac{\partial J}{\partial w_{ji}}$$

??

At the Output Layer:

$(o_1 - y_1)^2$

$(o_2 - y_2)^2$

$\vdots$

$(o_k - y_k)^2$

$y \in \{1, \dots, k\}$

1 of k Encoding (Dummy).

$\bar{Y} \in R^k$

Vector with k entries

For example, if k = 10.

Let y = 4

Neural Networks

Squared loss

$$J = \sum_{i=1}^{N} J_i \qquad \frac{\partial J}{\partial w_{ij}} = \sum_{i=1}^{N} \frac{\partial J_i}{\partial w}$$

$$= \sum_{i=1}^{N}\sum_{l=1}^{K} (y_{il} - O_{il})^2$$

True label for i^{th} training point.

Output of the NN at the l^{th} output unit for the i^{th} training point.

For any y_i ($\in \{1, \cdots, k\}$)

0	0	0	0	0	1	0	0	0

k length vector

e.g. $y_1 = 4$

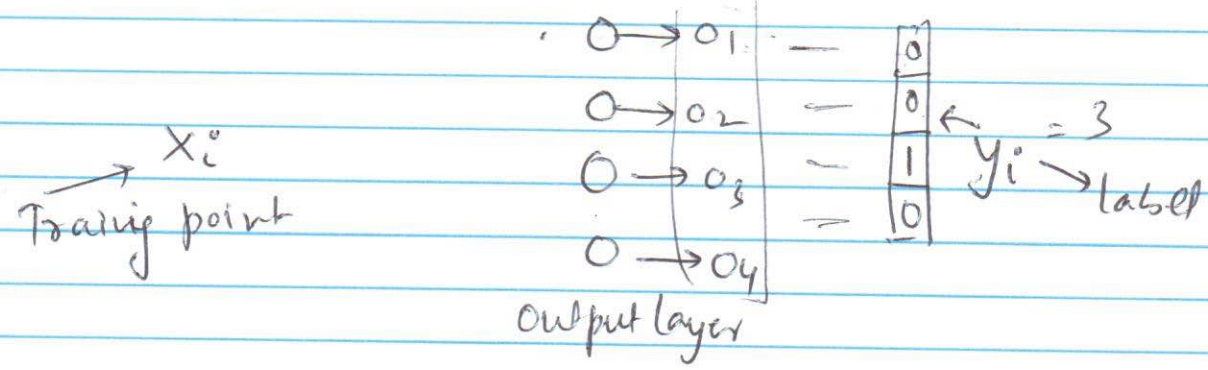
0	0	0	1	0	0	0	0	0	0

1 of k encoding.

Such that $y_{il} = 1$ if $l = y_i$
$y_{il} = 0$ else

---

e.g ~~k = 10~~ $k = 4$



i – tracks training examples $\quad i \in 1, \ldots, N$

$x_i \rightarrow i^{th}$ training example.

$p \rightarrow$ track features $\quad p \in 1, \ldots, D$

$x_{ip} \rightarrow p^{th}$ feature of x_i

~~l – tra~~

---

j – track hidden units $\quad j \in 1, \ldots, M$

net_j – Summation for j^{th} hidden unit.

$$net_j = w_j^T x$$

$$z_j = \sigma(net_j) \rightarrow \text{output of the } j^{th} \text{ hidden unit.}$$

w_j – weight vector for j^{th} hidden unit.

$w_j \rightarrow$ length D [assume D includes the bias]

$w_{jp} \rightarrow p^{th}$ entry of the weight vector of j^{th} hidden unit.

---

l – track output units $\quad l \in 1, \ldots, k$

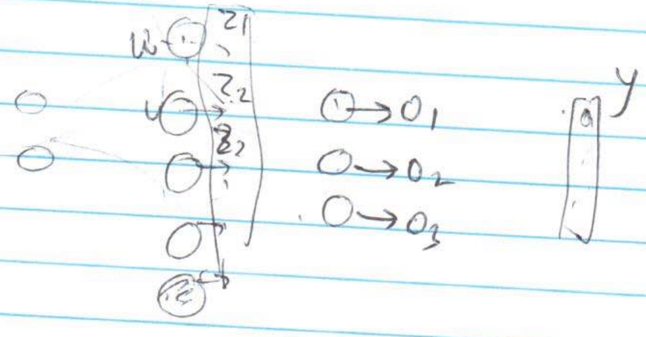
~~net~~ w_l – weight vector for l^{th} output unit

$\rightarrow$ length $(M+1)$

$w_{lj} \rightarrow j^{th}$ entry of the weight vector of the l^{th} output unit

$$net_l = w_l^T \mathbf{z} \qquad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \\ 1 \end{bmatrix}$$

$$O_l = \sigma(net_l) \rightarrow \text{output at } l^{th} \text{ output unit.}$$



r th unit.

$$\frac{\partial J}{\partial w_{r1}} = \frac{\partial J}{\partial net_r} \frac{\partial net_r}{\partial w_{r1}}$$

$net_r = u_{r1} w_{r1} + u_{r2} w_{r2} + \cdots$

$$\boxed{\frac{\partial net_r}{\partial w_{r1}} = u_{r1}}$$

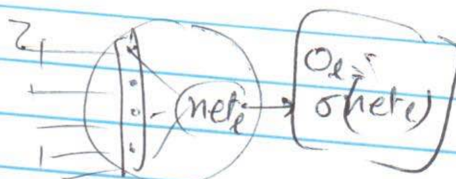
$$\boxed{\frac{\partial J}{\partial w_{rq}} = \frac{\partial J}{\partial net_r} u_{rq}}$$

Output unit.

$$\boxed{\frac{\partial J}{\partial w_{jp}} = \frac{\partial J}{\partial net_j} x_p} \qquad \boxed{\frac{\partial J}{\partial w_{\ell j}} = \frac{\partial J}{\partial net_\ell} z_j}$$

At the output node (ℓ)

$$\frac{\partial J}{\partial net_\ell} = \frac{\partial J}{\partial o_\ell} \frac{\partial o_\ell}{\partial net_\ell}$$



What is $\frac{\partial o_\ell}{\partial net_\ell} = \frac{\partial}{\partial net_\ell} \sigma(net_\ell)$

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$

$$\frac{d}{da}\sigma(a) = \frac{-\exp(-a)}{(1+\exp(-a))^2}$$

$$= \left(\frac{1}{1+\exp(-a)}\right)\left(1 - \frac{1}{1+\exp(-a)}\right)$$

$$\boxed{\frac{d\sigma(a)}{da} = \sigma(a)(1-\sigma(a))}$$

$$\boxed{\frac{\partial o_\ell}{\partial net_\ell} = o_\ell(1-o_\ell)}$$

$$\frac{\partial J}{\partial net_\ell} = \frac{\partial J}{\partial o_\ell}\frac{\partial o_\ell}{\partial net_\ell} = o_\ell(1-o_\ell)\frac{\partial J}{\partial o_\ell}$$

What is $\frac{\partial J}{\partial o_\ell}$?

$$J = \frac{1}{2}(y_\ell - o_\ell)^2$$

$$\frac{\partial J}{\partial o_\ell} = -(y_\ell - o_\ell)$$

$$\boxed{\frac{\partial J}{\partial net_\ell} = -o_\ell(1-o_\ell)(y_\ell - o_\ell)}$$

$$\boxed{\frac{\partial J}{\partial w_{\ell j}} = -o_\ell(1-o_\ell)(y_\ell - o_\ell)z_j}$$