Thomas J. Watson School of Engineering & Applied Science
Department of Electrical & Computer Engineering

Neural Network & Deep Learning

(EECE680C)

Homework_2

Solutions

Submitted By

Alem Haddush Fitwi
afitwi1@binghamton.edu
Graduate Student

Submitted To

Dr. Xiaohua Li
Associate Professor
xli@binghamton.edu

Submission Date

01 March 2018

24 February 2018

Binghamton, New York

Solutions to problems 1-4

(1) Single Neuron classification problem

1.1 Given variables & parameters

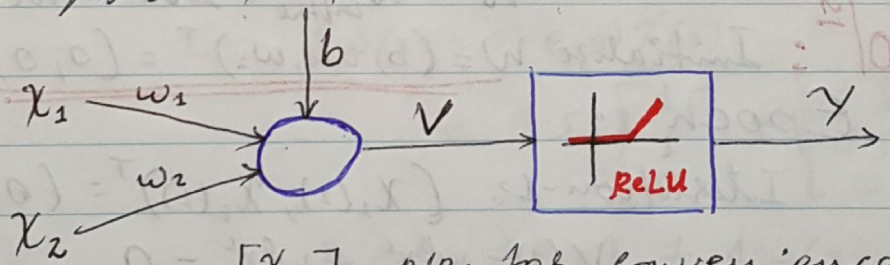
Input X		Class
x_1	x_2	
0	0	C_1
0	1	C_1
1	0	C_2
1	1	C_2

& $\eta = 0.5$

NB: Assumptions made for C_1 & C_2

If $W^T X(n) \leq 0$, $X(n)$ is in class C_1
If $W^T X(n) > 0$, $X(n)$ is in class C_2

1.2 Architecture



$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

NB: for convenience, the bias is usually viewed as being the weight associated with an input x_0 that is permanently set to 1. $\Rightarrow$ $x_0 = 1$

$$W = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$$

ReLu $\rightarrow$ $\phi(v) = \max(0, v) = \begin{cases} v, & \text{if } v > 0 \\ 0, & \text{if } v \leq 0 \end{cases}$

1.3 The Activity Rule

(a) $V = \sum_{i}^{n} w_i x_i$

(b) $y = \phi(v) = \max(0, v) = \begin{cases} v, & \text{if } v > 0 \\ 0, & \text{if } v \leq 0 \end{cases}$

1.4 The Learning Rule

(a) Learning error calculation

$e = t - y \rightarrow e(n) = t(n) - y(n)$

where t = target value

$t = -1$ for class C_1 } In my case!

$t = 1$ for class C_2 }

y = the actual value

(b) Weight update (or change in weight)

$\Delta w = \eta e x \rightarrow \Delta w(n) = \eta e(n) x(n)$

$w = w_{initial} + \Delta w \rightarrow w(n) = w(n-1) + \Delta w(n)$

Solⁿ: Initialize $w = (b, w_1, w_2)^T = (0, 0, 0)$

Epoch-1:

Iteration-1: $(x_1(1), x_2(1))^T = (0, 0)^T \rightarrow C_1$

$v(1) = w_1^0 x_1^1 + w_2^0 x_2^1 = 0$

$y(1) = \phi(v(1) + b(0)) = 0$

$e(1) = t(1) - y(1) = -1 - 0 = -1$

$\Delta w(1) = \eta e(1) x(1)$

$= (0.5) \times (-1) \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0 \\ 0 \end{bmatrix}$

$w(1) = w(0) + \Delta w(1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0 \\ 0 \end{bmatrix}$

Iteration-2: $(x_1(2), x_2(2))^T = (0,1)^T \rightarrow C_1$

$$v(2) = w_1(1)x_1(2) + w_2(1)x_2(2) = 0$$

$$y(2) = \phi(v(2) + b(1)) = 0$$

$$e(2) = t(2) - y(2) = -1 - 0 = -1 //$$

$$\Delta w(2) = \eta e(2) x(2)$$

$$= -0.5 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0 \\ -0.5 \end{bmatrix}$$

$$w(2) = w(1) + \Delta w(2)$$

$$= \begin{bmatrix} -0.5 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -0.5 \end{bmatrix}$$

Iteration-3: $(x_1(3), x_2(3))^T = (1,0)^T \rightarrow C_2$

$$v(3) = w_1(2)x_1(3) + w_2(2)x_2(3) = 0$$

$$y(3) = \phi(v(3) + b(2)) = 0$$

$$e(3) = t(3) - y(3) = 1 - 0 = 1 //$$

$$\Delta w(3) = \eta e(3) x(3)$$

$$= 0.5 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}$$

$$w(3) = w(2) + \Delta w(3)$$

$$= \begin{bmatrix} -1 \\ 0 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

Iteration-4: $(x_1(4), x_2(4))^T = (1,1)^T \rightarrow C_2$

$$v(4) = w_1(3)x_1(4) + w_2(3)x_2(4) = 0 //$$

$$y(4) = \phi(v(4) + b(3)) = 0 //$$

$$e(4) = t(4) - y(4) = 1 - 0 = 1 //$$

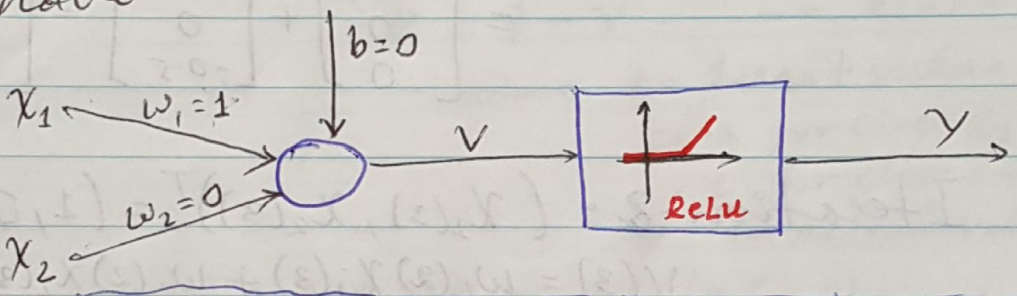
$$\Delta w(4) = \eta\, e(4)\, X(4)$$

$$= 0.5 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

$$w(4) = w(3) + \Delta w(4)$$

$$= \begin{bmatrix} -0.5 \\ 0.5 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Hence, at the end of Epoch-1, we have



X		Given Class	$w_i x_i + b$	Result	
x_1	x_2				
0	0	C_1	0	≤ 0	C_1
0	1	C_1	0	≤ 0	C_1
1	0	C_2	1	>0	C_2
1	1	C_2	1	>0	C_2

* No need to go for more epoches !!

(2) Regression problem

2.1 Given variables or parameters or values

- Input (or Regressor)

$$X = \{-0.5, -0.2, -0.1, 0.3, 0.4, 0.5, 0.7\}$$

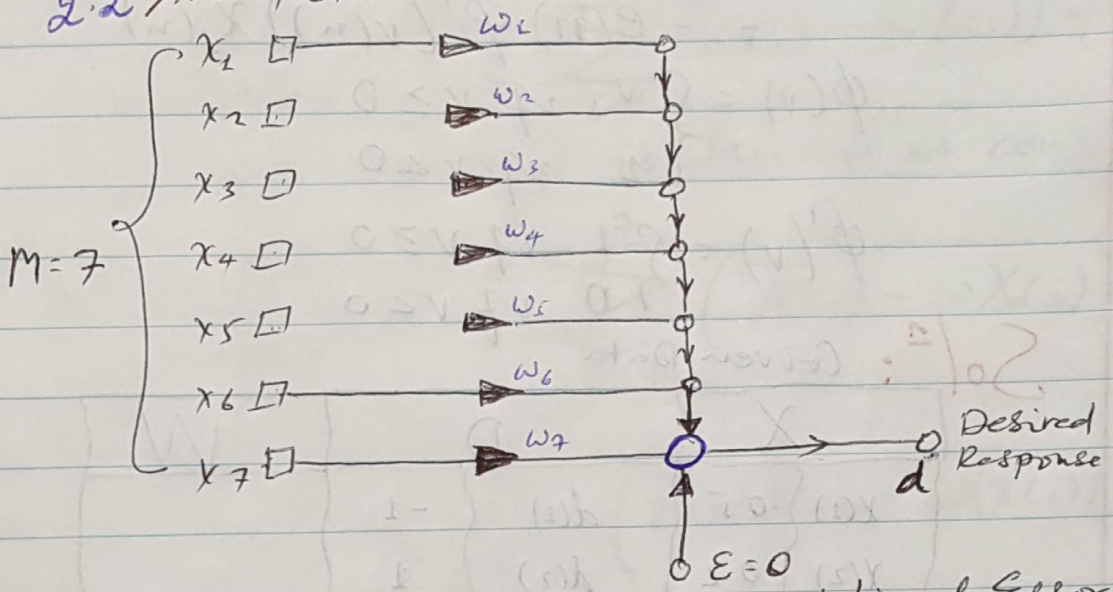
- Desired output (or Response)

$D = \{-1, 1, 2, 3.2, 3.5, 5, 6\}$

✓ Bias (ε) is set to 0

$\varepsilon = 0$

2.2 Architecture



2.3 Linear Regressor

$$d = \sum_{j=1}^{M} w_j x_j + \varepsilon \quad \text{or} \quad \Rightarrow \boxed{d = w^* x + b}$$

(weight, bias)

$$\Rightarrow \varepsilon = 0$$

$$d = W^T X + \varepsilon \Rightarrow \text{In vector form}$$

2.4 The cost function

$$W^* = \arg\min_{w} \frac{1}{2} \sum_{i=1}^{N} \left(d(i) - W^T x(i)\right)^2 \quad \text{OR}$$

$$E(w) = \frac{1}{2} \sum_{i=1}^{N} (d(i) - w x(i))^2$$

2.4 The Gradient (for Linear Regression)

$$\nabla E = \frac{dE(w)}{dw} = -\sum_{i=1}^{N} (d(i) - w x(i)) x(i)$$

2.5 The Gradient for non-linear Regression

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial v}\frac{\partial v}{\partial w_j}$$

$$= -[d(n) - y(n)]\,\phi'(v(n))\,x_j(n)$$

$$= -e(n)\,\varphi'(v(n))\,x_j(n)$$

$$\phi(v) = \begin{cases} v & \text{if } v \geq 0 \\ 1 & \text{if } v < 0 \end{cases}$$

$$\phi'(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

Sol^{n}: Given Data

X		D		W
x(1)	-0.5	d(1)	-1	
x(2)	-0.2	d(2)	1	
x(3)	-0.1	d(3)	2	w = 2
x(4)	0.3	d(4)	3.2	
x(5)	0.4	d(5)	3.5	
x(6)	0.5	d(6)	5	
x(7)	0.7	d(7)	6	

2(a). using linear activation Function

i	x(i)	d(i)	x(i) d(i)	$(x(i))^2$	wx(i)
1	-0.5	-1	0.5	0.25	-0.5w
2	-0.2	1	-0.2	0.04	-0.2w
3	-0.1	2	-0.2	0.01	-0.1w
4	0.3	3.2	0.96	0.09	0.3w
5	0.4	3.5	1.4	0.16	0.4w
6	0.5	5	2.5	0.25	0.5w
7	0.7	6	4.2	0.49	0.7w
Σ	1.1	19.7	9.16	1.29	1.1w

Then, the gradient is computed as ensues

$$E = \frac{1}{2}\sum_{i=1}^{7}\left(d(i) - wX(i)\right)^2$$

- partial derivative wrt w

$$\frac{dE}{dw} = 2 * \frac{1}{2}\sum_{i=1}^{7}\left(d(i) - wX(i)\right) * \left(-X(i)\right)$$

→ $d(i)$ is treated as constant

$$\frac{dE}{dw} = -\sum_{i=1}^{7}\left(d(i) - wX(i)\right) * X(i)$$

$$= -\sum_{i=1}^{7}\left(d(i)X(i) - w(X(i))^2\right)$$

$$= -\Big[(0.5 - 0.25w) + (-0.2 - 0.04w) + (-0.2 - 0.01w) + (0.96 - 0.09w) + (1.4 - 0.16w) + (2.5 - 0.25w) + (4.2 - 0.49w)\Big]$$

$$= -\left[9.16 - 1.29w\right]$$

$$\frac{dE}{dw} = \nabla E = 1.29w - 9.16$$

$\nabla E = 1.29w - 9.16$

At $w = 2$

$\nabla E = 1.29 * 2 - 9.16$

$= -6.58$

2(b) using ReLU non-linear function

The Gradient expression or equation is

$$E = \frac{1}{2}[d(i) - y(i)]^2$$

$$\nabla E = \frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial v}\frac{\partial v}{\partial w_j}$$

$$= -[d(i) - y(i)]\,\varphi'(v(i))\,x(i)$$

At $w = 2$

i	x(i)	d(i)	v(i) = wx(i)	y = φ(v(i))	φ'(v(i))
1	-0.5	-1	-0.5w	0	-0.5
2	-0.2	1	-0.2w	0	-0.2
3	-0.1	2	-0.1w	0	-0.1
4	0.3	3.2	0.3w	0.6	0.3
5	0.4	3.5	0.4w	0.8	0.4
6	0.5	5	0.5w	1.0	0.5
7	0.7	6	0.7w	1.4	0.7
Σ	1.1	19.7			

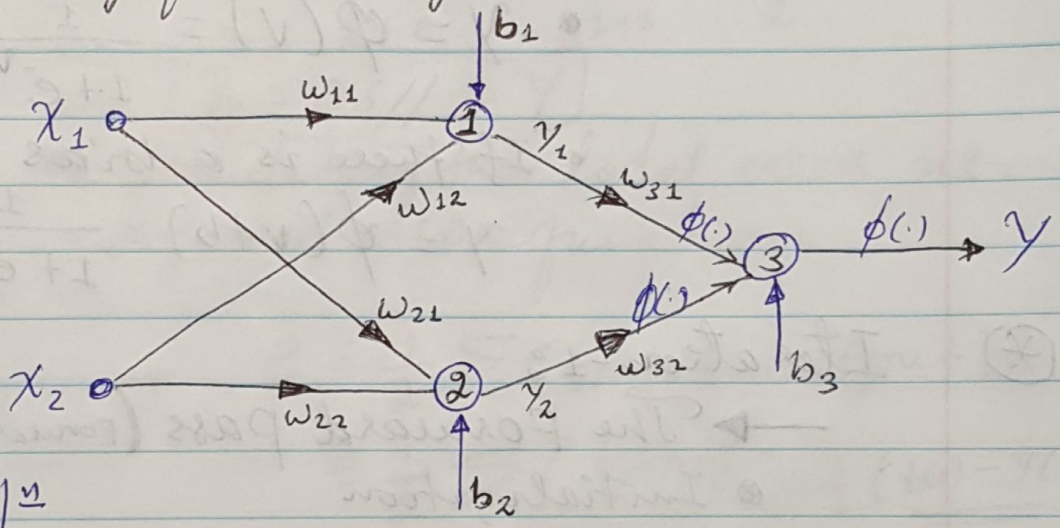
$$\nabla E = -[(-1)*0.25 + (1)*0.04 + (2)*0.01 + (3.2)*0.09 + (3.5)*0.16 + (5)*0.25 + (6)*0.49]$$

$$= -[-0.25 + 0.04 + 0.02 + 0.288 + 0.56 + 1.25 + 2.94]$$

$$= -[4.848]$$

$$= -4.848 //$$

(3) Use the backpropagation algorithm for computing a set of synaptic weights & bias levels for a neural network structured in figure 4.8 page 221 problem 4.2 of the text book to solve the XOR problem. Assume the use of a logistic function for the non-linearity. Compute just two iterations of the BPA updating beginning from all zero initial conditions.



Sol/:

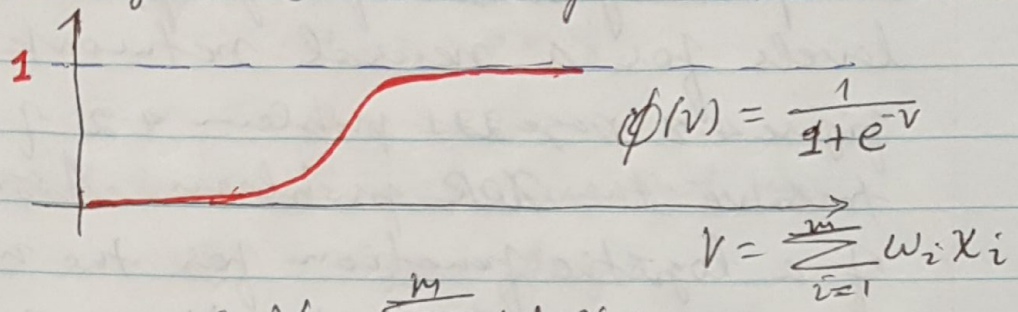
Truth Table of XOR

x_1	x_2	Target $= t = x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

$$x^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \& \quad t = [0\ 1\ 1\ 0]$$

✓ Activation Function

- Logistic (a.k.a Sigmoid) Function



- $v = \sum_{i=1}^{m} w_i x_i$
- $y = \phi(v) = \frac{1}{1+e^{-v}}$
- If there is a bias

$$y = \phi(v+b) = \frac{1}{1+e^{-(v+b)}}$$

⊛ Iteration-1:

→ The Forward pass (Forward propagation)

- Initialization

$$W_{11} = W_{12} = W_{21} = W_{22} = b_1 = b_2 = 0 \text{ //}$$
$$W_{31} = W_{32} = b_3 = 0 \text{ //}$$

@ Hidden Node 1

$$v_1 = w_{11}x_1 + w_{12}x_2 = 0 \text{ //}$$
$$y_1 = \phi(v_1 + b_1) = \phi(0)$$
$$= \frac{1}{1+e^{-0}} = \frac{1}{1+1} = \frac{1}{2}$$

$$\boxed{y_1 = 0.5}$$

@ Hidden Node 2

$$v_2 = w_{21}x_1 + w_{22}x_2 = 0 \text{ //}$$

$$Y_2 = \phi(V_2 + b_2) = 0$$

$$= \frac{1}{1+e^{-0}} = \frac{1}{1+1}$$

$$= 0.5 //$$

@ Output Node 3

$$V_3 = W_{31}Y_1 + W_{32}Y_2 = 0$$

$$Y_3 = \phi(V_3 + b_3) = \phi(0)$$

$$= \frac{1}{1+e^{-0}} = \frac{1}{1+1} = \frac{1}{2}$$

$$Y_3 = 0.5 // = Y$$

- Calculating the total error using the squared error function

$$e = \frac{1}{2}\sum(\text{target} - \text{output})^2$$

$$= \frac{1}{2}\sum(t-y)^2 = \frac{1}{2}(t(1) - Y(1))^2$$

- There is only one output node, ∴

$$e = \frac{1}{2}(t-y)^2 = \frac{1}{2}(t(1) - Y(1))^2$$

$$= \frac{1}{2}(0-0.5)^2$$

$$e = \frac{1}{8}$$

$$\boxed{e = \frac{1}{8}}$$

The Backward pass (Backpropagation)

The prime goal of Backpropagation is

to update the weights in the network so that they cause the output to be closer to the target output.

✓ @ Output Layer (node 3) ⇒ w_{31} & w_{32}

We want to know how much a change in w_{31} & w_{32} affects the overall error

overall error, ⇒ $\frac{\partial e}{\partial w}$, that's

- Gradient w.r.t w_{31}

$$= \frac{\partial e}{\partial w_{31}} = \frac{\partial e}{\partial y} \frac{\partial y}{\partial y_1} \frac{\partial y_1}{\partial w_{31}}$$

- Gradient w.r.t w_{32}

$$= \frac{\partial e}{\partial w_{32}} = \frac{\partial e}{\partial y} \cdot \frac{\partial y}{\partial y_2} \cdot \frac{\partial y_2}{\partial w_{32}}$$

- As computed in the forward pass

$$e = \frac{1}{2}(t-y)^2$$

$$\frac{\partial e}{\partial y} = 2 \times \frac{1}{2}(t-y) \cdot (-1)$$

$$\frac{\partial e}{\partial y} = -(t-y)$$

$$\frac{\partial e}{\partial y} = -(0-0.5)$$

$$= 0.5 //$$

- How much does the output, y, change w.r.t its total net input

$$y = \frac{1}{1+e^{-(v_3+b_3)}}$$

$$y = \frac{1}{1+e^{-v}}, \quad v = v_3 + b_3$$

$$\frac{\partial Y}{\partial V_3} = Y(1-Y) \rightarrow \text{(derivative of the logistic function)}$$

$$\phi(v) = \frac{1}{1+e^{-v}} = (1+e^{-v})^{-1}$$

$$\phi'(v) = -(1+e^{-x})^{-2}(-e^{-x})$$

$$= \frac{-1}{1+e^{-x}} \cdot \frac{-1}{1+e^{-x}} \cdot e^{-x}$$

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

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

$$= (\phi(v))(1-\phi(v))$$

$$\phi'(v) = \phi(v)(1-\phi(v))$$

$$\frac{\partial Y}{\partial V_3} = Y(1-Y) = 0.5(1-0.5)$$

$$= \frac{1}{4} = 0.25 //$$

- How much does the total input of Y change w.r.to W_{31}?

$$V_3 = W_{31} * Y_1 + W_{32} * Y_2 + b_3 * 1$$

$$\frac{\partial V_3}{\partial W_{31}} = 1 * Y_1 * W_{31}^{(1-1)} + 0 + 0$$

$$= Y_1$$

$$= 0.5 //$$

* putting it all together

$$\frac{\partial e}{\partial w_{31}} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} * \frac{\partial v_3}{\partial w_{31}} = \Delta w_{31}$$

$$= 0.5 * 0.25 * 0.5$$

$$= \frac{1}{2} * \frac{1}{4} * \frac{1}{2}$$

$$= \frac{1}{16}$$

$\Delta w_{31} = 0.0625$ //

* Often this is combined in the form of the delta rule

$$\frac{\partial e}{\partial w_{31}} = -(\text{target} - \text{out}) * \text{out}(1 - \text{out}) * y_1$$

(Delta Rule)

$$= -(t - y) * y(1 - y) * y_1$$

$$= -(0 - 0.5) * 0.5(1 - 0.5) * 0.5$$

$$= -(-0.5) * (0.5)(0.5)(0.5)$$

The same?

$\Delta w_{31} = \frac{1}{16}$ //

* OR

$$\delta = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} = \frac{\partial e}{\partial v_3}$$

$$\delta = -(t - y) * y(1 - y)$$

Hence, $\frac{\partial e}{\partial w_{31}} = \delta * y_1 = \Delta w_{31}$

$$\Delta w_{32} = \frac{\partial e}{\partial w_{32}} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} * \frac{\partial v_3}{\partial w_{32}}$$

$$\frac{\partial v_3}{\partial w_{32}} = \frac{\partial (w_{31} y_2 + w_{32} x_2 + b_2 * 1)}{\partial w_{32}}$$

$$= 0 + y_2 + 0 = y_2 = 0.5$$ //

$$\Delta w_{32} = \frac{\partial e}{\partial w_{32}} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} * \frac{\partial v_3}{\partial w_{32}}$$

$$= 0.5 * 0.25 * 0.5$$

$$= \frac{1}{16} = 0.0625 //$$

@ $\frac{\partial e}{\partial b_3} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} * \frac{\partial v_3}{\partial b_3} = 0.125 //$

↳ Updating Output layer synaptic weights

Let the learning rate, $\eta = 0.5$

$$w_{31}^{+} = w_{31} - \eta \Delta w_{31}$$
$$= 0 - 0.5 * 0.0625$$
$$= -0.03125 //$$

$$w_{32}^{+} = w_{32} - \eta \Delta w_{32}$$
$$= 0 - 0.5 * 0.0625$$
$$= -0.03125 //$$

$$\Delta b_3 = \frac{\partial e}{\partial b_3}$$
$$= \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} * \frac{\partial v_3}{\partial b_3}$$
$$= 0.5 * 0.25 * 1$$
$$= 0.125$$

$$b_3^{+} = b_3 - \eta \Delta b_3$$
$$= -0.0625 //$$

@ Hidden Layer Node 1 → w_{11} & w_{12}

$$\begin{cases} \Delta w_{11} = \frac{\partial e}{\partial w_{11}} = \frac{\partial e}{y_1} * \frac{\partial y_1}{\partial v_1} * \frac{\partial v_1}{\partial w_{11}} \\ \Delta w_{12} = \frac{\partial e}{\partial w_{12}} = \frac{\partial e}{\partial y_1} * \frac{\partial y_1}{\partial v_1} * \frac{\partial v_1}{\partial w_{12}} \end{cases}$$

↳ computing Δw_{11}

$$\frac{\partial e}{\partial y_1} = \frac{\partial e}{\partial v_3} * \frac{\partial v_3}{\partial y_1}$$

• $\frac{\partial e}{\partial v_3} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} = 0.5 * 0.25 = 0.125 //$

• $v_3 = w_{31} y_1 + w_{32} y_2 + b_3$

$$\frac{\partial v_3}{\partial y_1} = w_{31} + 0 + 0 = w_{31}$$

$$\frac{\partial v_3}{\partial y_1} = 0 //$$

$$\frac{\partial e}{\partial y_1} = \frac{\partial e}{\partial v_3} * \frac{\partial v_3}{\partial y_1} = 0.125 * 0 = 0 //$$

- $\frac{\partial y_1}{\partial v_1} = y_1(1-y_1) = 0.5(1-0.5)$

$= 0.25 //$

- $\frac{\partial v_1}{\partial w_{11}} = \frac{\partial}{\partial w_{11}}(w_{11}x_1 + w_{12}x_2 + b_1 * 1)$

$= x_1$

$= 0 //$

- putting it all together

$$\Delta w_{11} = \frac{\partial e}{\partial w_{11}} = \frac{\partial e}{\partial y_1} * \frac{\partial y_1}{\partial v_1} * \frac{\partial v_1}{\partial w_{11}}$$

$= 0 * 0.25 * 0$

$= 0 //$

- computing Δw_{12}

$$\Delta w_{12} = \frac{\partial e}{\partial w_{12}} = \frac{\partial e}{\partial y_1} * \frac{\partial y_1}{\partial v_1} * \frac{\partial v_1}{\partial w_{12}}$$

$\longrightarrow$

$$\frac{\partial v_1}{\partial w_{12}} = \frac{\partial}{\partial w_{12}}(w_{11}x_1 + w_{12}x_2 + b_1 * 1)$$

$= 0 + x_2 + 0$

$= x_2$

$= 0 //$

- updating w_{11} & w_{12}

$w_{11}^+ = w_{11} - \eta \Delta w_{11}$

$= 0 - 0.5 * 0$

$= 0 //$

$$w_{12}^{+} = w_{12} - \eta \Delta w_{12}$$
$$= 0 - 0.5 * 0$$
$$= 0 //$$

$$\Delta b_1 = \frac{\partial e}{\partial b_1} = \frac{\partial e}{\partial y_1} \cdot \frac{\partial y_1}{\partial v_1} \cdot \frac{\partial v_1}{\partial b_1}$$
$$= 0 * 0.25 * 1$$
$$= 0 //$$
$$b_1^{+} = 0 - 0$$
$$= 0 //$$

@ Hidden Layer Node 2 → w_{21} & w_{22}

$$\Delta w_{21} = \frac{\partial e}{\partial w_{21}} = \frac{\partial e}{\partial y_2} * \frac{\partial y_2}{\partial v_2} * \frac{\partial v_2}{\partial w_{21}}$$

$$\rightarrow \frac{\partial e}{\partial y_2} = \frac{\partial e}{\partial v_3} * \frac{\partial v_3}{\partial y_2}$$

$$\rightarrow \frac{\partial e}{\partial v_3} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} = 0.5 * 0.25$$
$$= 0.125 //$$

$$\rightarrow \frac{\partial v_3}{\partial y_2} = \frac{\partial}{\partial y_2}(w_{31} * y_1 + w_{32} * y_2 + b_3 * 1)$$
$$= w_{32} //$$
$$= 0 //$$

$$\rightarrow \frac{\partial y_2}{\partial v_2} = y_2(1 - y_2) = 0.5(1 - 0.5) = 0.25 //$$

$$\rightarrow \frac{\partial v_2}{\partial w_{21}} = \frac{\partial}{\partial w_{21}}(w_{21} x_1 + w_{22} x_2 + b_2 * 1)$$
$$= x_1$$
$$= 0 //$$

$$\therefore \Delta w_{21} = \frac{\partial e}{\partial w_{21}} = 0 * 0.25 * 0$$
$$= 0 //$$

$$\Delta w_{22} = \frac{\partial e}{\partial w_{22}} = \frac{\partial e}{\partial y_2} * \frac{\partial y_2}{\partial v_2} * \frac{\partial v_2}{\partial w_{22}}$$

$$\rightarrow \frac{\partial e}{\partial y_2} = \frac{\partial e}{\partial v_3} * \frac{\partial v_3}{\partial y_2} = 0.125 * 0 = 0 //$$

$$\rightarrow \frac{\partial e}{\partial v_3} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} = 0.5 * 0.25$$

$$= 0.125 //$$

$$\rightarrow \frac{\partial v_3}{\partial y_2} = 0$$

$$\rightarrow \frac{\partial y_2}{\partial v_2} = y_2(1-y_2) = 0.5(1-0.5)$$

$$= 0.25 //$$

$$\rightarrow \frac{\partial v_2}{\partial w_{22}} = \frac{\partial (w_{21}x_1 + w_{22}x_2 + b_2 * 1)}{\partial w_{22}}$$

$$= x_2$$

$$= 0 //$$

$$\therefore \Delta w_{22} = \frac{\partial e}{\partial w_{22}} = 0 * 0.25 * 0 = 0 //$$

- Updating w_{21} & w_{22}

$$w_{21}^+ = w_{21} - \eta \Delta w_{21}$$
$$= 0 - 0.5 * 0$$
$$= 0 //$$

$$w_{22}^+ = w_{22} - \eta \Delta w_{22}$$
$$= 0 - 0.5 * 0$$
$$= 0 //$$

$$\Delta b_2 = \frac{\partial e}{\partial b_2}$$
$$= \frac{\partial e}{\partial y_2} * \frac{\partial y_2}{\partial v_2} * \frac{\partial v_2}{\partial b_2}$$
$$= 0 * 0.25 * 1$$
$$= 0 //$$

$$b_2^+ = b_2 - \eta \Delta b_2$$
$$= 0 - 0$$
$$= 0 //$$

(*) Iteration - 2:

→ Forward pass $(x_1, x_2) = (0, 1)$

↳ New Values of synaptic weights

$$\begin{cases} w_{11} = w_{12} = b_1 = 0 \\ w_{21} = w_{22} = b_2 = 0 \\ w_{31} = w_{32} = -0.03125 \\ b_3 = -0.0625 \end{cases}$$

(a) Hidden Layer, Node 1

$$v_1 = w_{11}x_1 + w_{12}x_2 + b_1 * 1$$
$$= 0$$
$$y_1 = \phi(v_1) = \frac{1}{1+e^{-v}}$$
$$= \frac{1}{1+1}$$
$$= 0.5 //$$

(a) Hidden Layer, Node 2

$$v_2 = w_{21}x_1 + w_{22}x_2 + b_2 * 1$$
$$= 0$$
$$y_2 = \phi(v_2) = \frac{1}{1+e^{-v}}$$
$$= \frac{1}{1+1} = 1/2$$
$$= 0.5 //$$

(a) Output Layer, Node 3

$$v_3 = w_{31}y_1 + w_{32}y_2 + b_3 * 1$$
$$= -0.03125 * 0.5 - 0.03125 * 0.5 + (-0.0625) \times 1$$
$$= -0.09375$$
$$y_3 = y = \phi(v_3) = \frac{1}{1+e^{+0.09375}}$$
$$y_3 = y \simeq 0.4766 //$$

(a) Error calculation

$$e = \frac{1}{2}(t(2) - y(2))^2 //$$

$$\frac{\partial e}{\partial y} = 2 * \frac{1}{2}(t(2) - y(2)) * (-y(2))'$$

$$= -(t(2) - y(2))\, y(2)'$$

$$= -(t(2) - y(2))$$

Backward Pass

@ Output Layer, Node 3

$$\Delta w_{31} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} * \frac{\partial v_3}{\partial w_{31}}$$

$$\rightarrow \frac{\partial e}{\partial y} = -(t(2) - y(2))$$

$$= -(1 - 0.4766)$$

$$= -0.5234 //$$

$$\rightarrow \frac{\partial y}{\partial v_3} = y(1-y) = 0.4766(1 - 0.4766)$$

$$\approx 0.2495 //$$

$$\rightarrow \frac{\partial v_3}{\partial w_{31}} = \frac{\partial}{\partial w_{31}}(w_{31} y_1 + w_{32} y_2 + b_3 * 1)$$

$$= y_1 + 0 + 0$$

$$= y_1$$

$$= 0.5 //$$

$$\therefore \Delta w_{31} = \frac{\partial e}{\partial w_{31}} = -0.5234 * 0.2495 * 0.5$$

$$\approx -0.0653 //$$

$$\Delta w_{32} = \frac{\partial e}{\partial w_{32}} = \frac{\partial e}{\partial y} \frac{\partial y}{\partial v_3} \frac{\partial v_3}{\partial w_{32}}$$

$$\rightarrow \frac{\partial v_3}{\partial w_{32}} = y_2 = 0.5 //$$

$\therefore \Delta w_{32} = \frac{\partial e}{\partial w_{32}} = -0.5234 * 0.2495 * 0.5$

≈ -0.0653 //

↳ Updating w_{31}, w_{32}, & b_3

$w_{31}^{+} = w_{31} - \eta \Delta w_{31}$

$= -0.03125 - 0.5 * (-0.0653)$

$= 0.0014$ //

$w_{32}^{+} = w_{32} - \eta \Delta w_{32}$

$= -0.03125 - 0.5(-0.0653)$

$= 0.0014$

$b_3^{+} = b_3 - \eta \Delta b_3$

$\Delta b_3 = \frac{\partial e}{\partial b_3} = \frac{\partial e}{\partial y} * \frac{\partial y}{\partial v_3} * \frac{\partial v_3}{\partial b_3}$

$\rightarrow \frac{\partial v_3}{\partial b_3} = \frac{\partial}{\partial b_3}(w_{31} y_1 + w_{32} y_2 + b_3 * 1)$

$= 0 + 0 + (b_3 * 1)'$

$= 1$ //

$\therefore \Delta b_3 = (-0.5234)(0.2495)(1)$

$= -0.1306$ //

$\therefore b_3^{+} = b_3 - \eta \Delta b_3$

$= -0.0625 + 0.5 * 0.1306$

$= 0.0028$ //

@ Hidden Layer, Node 1 → w_{11}, w_{12}, & b_1

↳ $\Delta w_{11} = \frac{\partial e}{\partial w_{11}} = \frac{\partial e}{\partial y_1} \cdot \frac{\partial y_1}{\partial v_1} * \frac{\partial v_1}{\partial w_{11}}$

$\rightarrow \frac{\partial e}{\partial y_1} = \frac{\partial e}{\partial v_3} * \frac{\partial v_3}{\partial y_1}$

$$\rightarrow \frac{\partial e}{\partial v_3} = \frac{\partial e}{\partial y} \cdot \frac{\partial y}{\partial v_3} = (-0.5234 * 0.2495)$$

$$= -0.13059 //$$

$$\rightarrow \frac{\partial v_3}{\partial y_1} = \frac{\partial}{\partial y_1}(w_{31}y_1 + w_{32}y_2 + b_3 * 1)$$

$$= w_{31} + 0 + 0$$

$$= w_{31}$$

$$= -0.03125 //$$

$$\rightarrow \frac{\partial y_1}{\partial v_1} = y_1(1-y_1) = 0.5 * 0.5 = 0.25 //$$

$$\rightarrow \frac{\partial v_1}{\partial w_{11}} = \frac{\partial}{\partial w_{11}}(w_{11}x_1 + w_{12}x_2 + b_1 * 1)$$

$$= x_1 + 0 + 0$$

$$= x_1$$

$$= 0 //$$

$$\therefore \Delta w_{11} = \frac{\partial e}{\partial w_{11}} = 0 //$$

$$\therefore \Delta w_{12} = \frac{\partial e}{\partial w_{12}} = \frac{\partial e}{\partial y_1} \cdot \frac{\partial y_1}{\partial v_1} \cdot \frac{\partial v_1}{\partial w_{12}}$$

$$\rightarrow \frac{\partial v_1}{\partial w_{12}} = \frac{\partial}{\partial w_{12}}(w_{11}x_1 + w_{12}x_2 + b_1 * 0)$$

$$= 0 + x_2 + 0$$

$$= x_2$$

$$= 1 //$$

$$\Delta w_{12} = (-0.13059) * (-0.03125) * (0.25)(1)$$

$$= 0.00102 //$$

$$\therefore \Delta b_1 = \frac{\partial e}{\partial b_1} = \frac{\partial e}{\partial y_1} * \frac{\partial y_1}{\partial v_1} * \frac{\partial v_1}{\partial b_1}$$

$$\rightarrow \frac{\partial v_1}{\partial b_1} = \frac{\partial}{\partial b_1}(w_{11}x_1 + w_{12}x_2 + b_1 \times 1)$$

$$= 0 + 0 + 1$$

$$= 1 //$$

$$\therefore \Delta b_1 = (-0.5234 * 0.2495 * (-0.03125))(0.5 * 0.5)(2)$$

$$= 0.00102 //$$

- Updating w_{11}, w_{12}, & b_1

$$w_{11}^{+} = w_{11} - \eta \Delta w_{11}$$

$$= 0 - 0.5 * 0$$

$$= 0 //$$

$$w_{12}^{+} = w_{11} - \eta \Delta w_{11}$$

$$= 0 - 0.5 * 0.00102$$

$$= -0.00051 //$$

$$b_1^{+} = b_1 - \eta \Delta b_1$$

$$= 0 - 0.5 * 0.00102$$

$$= -0.00051 //$$

@ Hidden Layer, Node 2 → w_{21}, w_{22}, & b_2

- $\Delta w_{21} = \frac{\partial e}{\partial w_{21}} = \frac{\partial e}{\partial y_2} \times \frac{\partial y_2}{\partial v_2} \times \frac{\partial v_2}{\partial w_{21}}$

$$\rightarrow \frac{\partial v_2}{\partial w_{21}} = \frac{\partial}{\partial w_{21}}(w_{21}x_1 + w_{22}x_2 + b_2 \times 1)$$

$$= x_1$$

$$= 0 //$$

$$\therefore \Delta w_{21} = 0 //$$

- $\Delta w_{22} = \frac{\partial e}{\partial w_{22}} = \frac{\partial e}{\partial y_2} \times \frac{\partial y_2}{\partial v_2} \times \frac{\partial v_2}{\partial w_{22}}$

$$\rightarrow \frac{\partial e}{\partial Y_2} = \frac{\partial e}{\partial V_3} * \frac{\partial V_3}{\partial Y_2}$$

$$\rightarrow \frac{\partial e}{\partial V_3} = \frac{\partial e}{\partial Y} * \frac{\partial Y}{\partial V_3} = (-0.5234 * 0.2495)$$

$$\rightarrow \frac{\partial V_3}{\partial Y_2} = \frac{\partial}{\partial Y_2}(W_{31}Y_1 + W_{32}Y_2 + b_3 * 1)$$

$$= W_{32}$$

$$= -0.03125 //$$

$$\rightarrow \frac{\partial V_2}{\partial W_{22}} = \frac{\partial}{\partial W_{22}}(W_{21}X_1 + W_{22}X_2 + b_2 * 1)$$

$$= X_2$$

$$= 1$$

$$\therefore \Delta W_{22} = (-0.5234 * 0.2495 * -0.03125) * 1 * Y_2(1-Y_2)$$

$$= (0.00408)(0.5 * 0.5)(1)$$

$$= 0.00102 //$$

$$\therefore \Delta b_2 = \frac{\partial e}{\partial b_2} = \frac{\partial e}{\partial Y_2} * \frac{\partial Y_2}{\partial V_2} * \frac{\partial V_2}{\partial b_2}$$

$$\rightarrow \frac{\partial V_2}{\partial b_2} = \frac{\partial}{\partial b_2}(W_{21}X_1 + W_{22}X_2 + b_2 * 1)$$

$$= 1 //$$

$$\therefore \Delta b_2 = (0.00408)(0.5 * 0.5)(1)$$

$$= 0.00102 //$$

∴ Updating W_{21}, W_{22}, & b_2

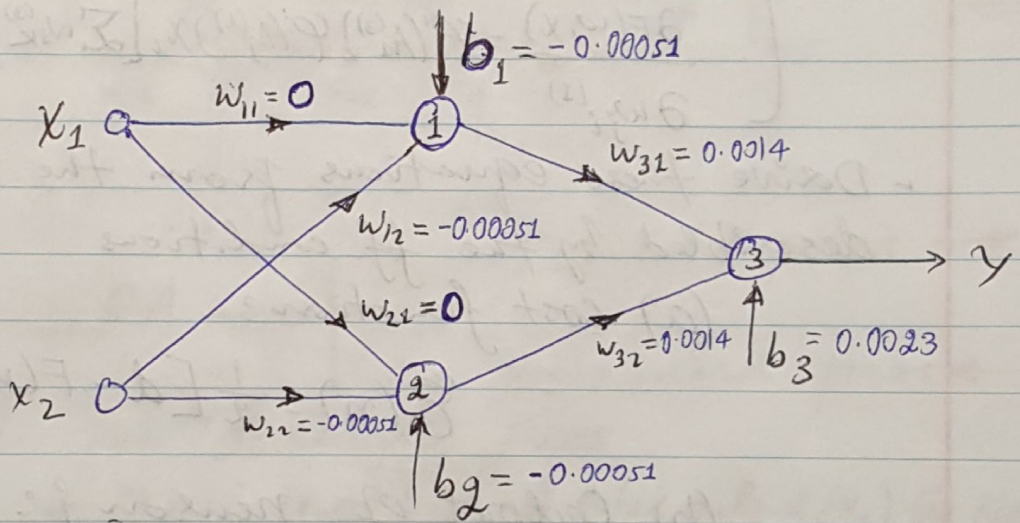
$$W_{21}^{+} = W_{21} - \eta \Delta W_{21}$$

$$= 0 - 0.5 * 0$$

$$= 0 //$$

$$W_{22}^{+} = W_{22} - \eta \Delta W_{22}$$
$$= 0 - 0.5 * 0.00102$$
$$= -0.00051 //$$
$$b_2^{+} = b_2 - \eta \Delta b_2$$
$$= 0 - 0.5 * 0.00102$$
$$= -0.00051 //$$

At the end of the 2nd iteration of the the first epoch, we have the ff:



That is,

- $w_{11} = w_{21} = 0$
- $w_{12} = w_{22} = -0.00051$
- $b_1 = b_2 = -0.00051$
- $w_{31} = w_{32} = 0.0014$
- $b_3 = 0.0023$

NB: Values are rounded off during calculation! //

4 Define the partial derivatives of the approximation function $F(w,x)$ realized by the multilayer perceptron in Fig. 4.14 of the text book

$$\frac{\partial F(w,x)}{\partial w_{1k}^{(3)}} = \varphi'(A_1^{(3)})\,\varphi(A_k^{(2)}) \quad (4.51)$$

$$\frac{\partial F(w,x)}{\partial w_{kj}^{(2)}} = \varphi'(A_1^{(3)})\,\varphi'(A_k^{(2)})\,(A_j^{(1)})\,w_{1k}^{(3)} \quad (4.52)$$

$$\frac{\partial F(w,x)}{\partial w_{ji}^{(1)}} = \varphi'(A_1^{(3)})\,\varphi'(A_j^{(1)})\,x_i\left[\sum_k w_{1k}^{(3)}\,\varphi'(A_k^{(2)}\,w_{kj}^{(2)}\right] \quad (4.53)$$

- Derive these equations from the scenario described by the ff conditions

(a) cost function:

$$\mathcal{E}(n) = \frac{1}{2}[d - F(w,x)]^2$$

(b) Output of neuron j:

$$Y_j = \varphi\left(\sum_i w_{ji}\,Y_i\right)$$

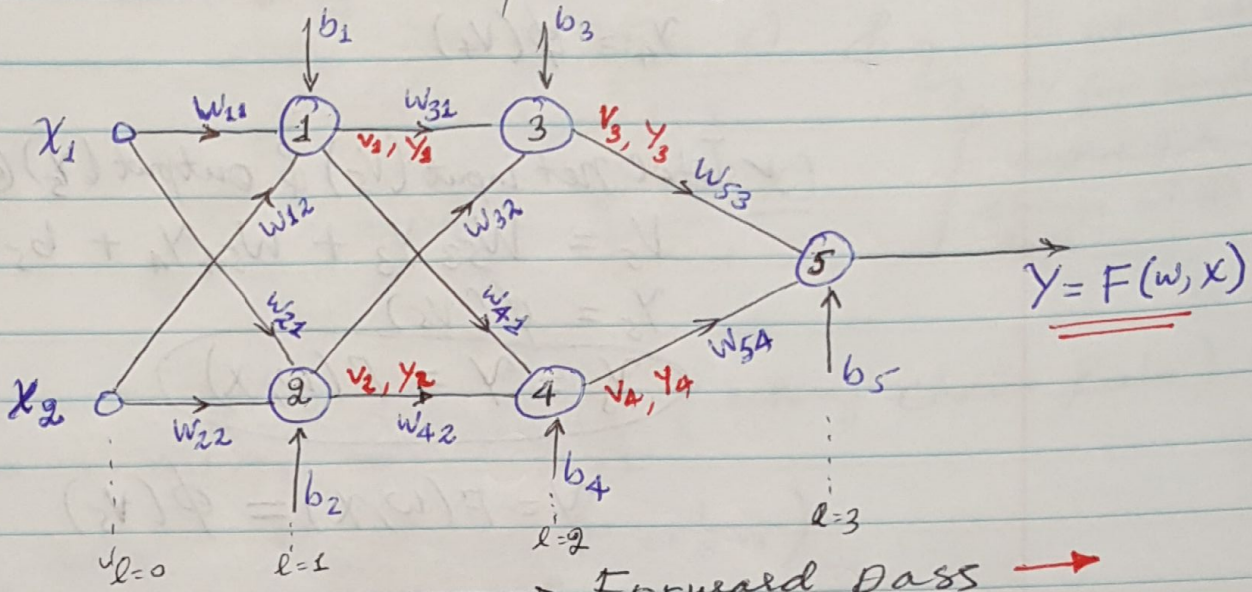
w_{ji} = Synaptic weight from neuron i to n.j.

(c) Nonlinearity

$$\phi(v) = \frac{1}{1 + e^{(-v)}}$$

Solution

For simplicity & better visualization, I have opted to use a simplified version of the multilayer perceptron network given in Fig. 4.14 of the textbook. The simplified one is hereafter portrayed.



- BPA → Forward pass →
  Backward pass ←

I. The Forward pass

- Total input (v_1) & total output (y_1) @ Node h_1

$$v_1 = w_{11}x_1 + w_{12}x_2 + b_1 * 1$$
$$y_1 = \phi(v_1)$$

- Total net input (v_2) & output (y_2) @ Node h_2

$$v_2 = w_{21}x_1 + w_{22}x_2 + b_2 * 1$$
$$y_2 = \phi(v_2)$$

- Total net input (v_3) & output (y_3) @ Node h_3

$$V_3 = W_{31} Y_1 + W_{32} Y_2 + b_3 * 1$$
$$Y_3 = \phi(V_3)$$

✓ Total net input (V_4) & output (Y_4) @ Node 4

$$V_4 = W_{41} Y_1 + W_{42} Y_2 + b_4 * 1$$
$$Y_4 = \phi(V_4)$$

✓ Total net input (V_5) & output (Y_5) @ Node 5

$$V_5 = W_{53} Y_3 + W_{54} Y_4 + b_5 * 1$$
$$Y_5 = \phi(V_5)$$
$$Y_5 = Y = F(w, x)$$

$$Y = F(w, x) = \phi(V_5)$$

✓ Computing the total error

$$E = \sum \frac{1}{2} [\text{target} - \text{output}]^2$$

where

target $= d$ &

output $= Y = F(w, x)$

$$E = \frac{1}{2} \sum [d - Y]^2$$

$$= \frac{1}{2} \sum [d - F(w, x)]^2$$

$$\frac{\partial E}{\partial Y} = - \sum [d - Y] Y$$

II. The Backward Pass

@ Output Layer ($l=3$), Node 5 → $w_{53}, w_{54}, \phi b_5$

✓ What is required here (as stated in the problem) is the derivation of $\frac{\partial Y}{\partial w_{ks}} = \frac{\partial F(w,x)}{\partial w_{ks}}$

Hence, using the simplified diagram:

$$\frac{\partial Y}{\partial w_{53}} = \frac{\partial F(w,x)}{\partial w_{53}} \quad \& \quad \frac{\partial Y}{\partial w_{54}} = \frac{\partial F(w,x)}{\partial w_{54}}$$

I will use Y for simple representation & change it @ last!

✓ $\frac{\partial Y}{\partial w_{53}} = \frac{\partial Y}{\partial v_5} * \frac{\partial v_5}{\partial w_{53}}$ (using chain Rule)

⊛ $\frac{\partial Y}{\partial v_5} = \frac{\partial}{\partial v_5}\left(\frac{1}{1+e^{-v_3}}\right)$

⊛⊛ Derivative of logistic function

$$\phi(v) = \frac{1}{1+e^{-v}} = (1+e^{-v})^{-1}$$

$$\phi'(v) = (1+e^{-v})^{-2} * e^{-v}$$

$$= \left(\frac{1}{1+e^{-v}}\right) * \left(\frac{e^{-v}}{1+e^{-v}}\right)$$

$$= \left(\frac{1}{1+e^{-v}}\right) * \left(1 - \frac{1}{1+e^{-v}}\right)$$

$$= \phi(v)\,(1-\phi(v))$$

$\therefore \frac{\partial Y}{\partial v_5} = \phi(v_5)(1-\phi(v_5)) = Y(1-Y)$

$$\frac{\partial V_5}{\partial w_{53}} = \frac{\partial}{\partial w_{53}}\left(w_{53}y_3 + w_{54}y_4 + b_5 * 1\right)$$

$$= y_3 + 0 + 0$$

$$= y_3$$

$$\therefore \frac{\partial y}{\partial w_{53}} = \frac{\partial y}{\partial V_5} * \frac{\partial V_5}{\partial w_{53}}$$

$$= y(1-y) * y_3$$

$$\frac{\partial y}{\partial w_{53}} = y(1-y)\,y_3$$

$\downarrow$ $y = F(w, x)$

$$\frac{\partial F(w,x)}{\partial w_{53}} = F(w,x)(1-F(w,x))\,y_3$$

$\downarrow$ Generalizing it using the text book's notation

$$\frac{\partial F(w,x)}{\partial w_{k1}} = F(w,x)(1-F(w,x))\,y_k^{(2)} \quad (1)$$

where y_k = output of k^{th} neuron!

@ Hidden Layer ($l=2$), Node 3 $\rightarrow w_{31}, w_{32}$, & b_3

↳ Here, compute (derive)

$\rightarrow \frac{\partial y}{\partial w_{31}}$, $\frac{\partial y}{\partial w_{32}}$, & $\frac{\partial y}{\partial b_3}$

$\rightarrow$ But one of them is enough!

↳ $\frac{\partial y}{\partial w_{31}} = \frac{\partial y}{\partial V_5} * \frac{\partial V_5}{\partial y_3} * \frac{\partial y_3}{\partial V_3} * \frac{\partial V_3}{\partial w_{31}}$ → (chain Rule)

* $\frac{\partial Y}{\partial V_5} = Y(1-Y)$ // → computed at output layer

* $\frac{\partial V_5}{\partial Y_3} = \frac{\partial}{\partial Y_3}(W_{53}Y_3 + W_{54}Y_4 + b_5 * 1)$

$= W_{53} + 0 + 0$

$= W_{53}$ //

* $\frac{\partial Y_3}{\partial V_3} = \frac{\partial}{\partial V_3}\left(\frac{1}{1+e^{-V_3}}\right) = Y_3(1-Y_3)$ //

* $\frac{\partial V_3}{\partial W_{31}} = \frac{\partial}{\partial W_{31}}(W_{31}Y_1 + W_{32}Y_2 + b_3 * 1)$

$= Y_1 + 0 + 0$

$= Y_1$ //

$\therefore \frac{\partial Y}{\partial W_{31}} = \frac{\partial Y}{\partial V_5} * \frac{\partial V_5}{\partial Y_3} * \frac{\partial Y_3}{\partial V_3} * \frac{\partial V_3}{\partial W_{31}}$

$= Y(1-Y) * W_{53} * Y_3(1-Y_3) * Y_1$

$= Y(1-Y) * Y_3(1-Y_3) * Y_1 * W_{53}$

$= F(w,x)(1-F(w,x)) * Y_3(1-Y_3) * Y_1 * W_{53}$

↓ Generalizing it using the text book's notation & multilayer perceptron

$$\frac{\partial F(w,x)}{\partial w_{kj}^{(2)}} = F(w,x)(1-F(w,x)) * y_k^{(2)}(1-y_k^{(2)}) * y_j^{(1)} * w_{1k}^{(3)}$$

where $y_k^{(2)}$ = layer 2 Neuron o/put, $y_j^{(1)}$ = Layer 1 neuron o/p (2)

(a) Hidden Layer ($l=1$), Node 1 → w_{11}, w_{12}, & b_1

- Here, compute (derive) → $\frac{\partial Y}{\partial w_{11}}$, $\frac{\partial Y}{\partial w_{12}}$, & $\frac{\partial Y}{\partial b_1}$

* But one of them's enough!

$$\checkmark \frac{\partial Y}{\partial w_{11}} = \frac{\partial Y}{\partial V_5} * \frac{\partial V_5}{\partial Y_3} * \frac{\partial Y_3}{\partial V_3} * \frac{\partial V_3}{\partial Y_1} * \frac{\partial Y_1}{\partial V_1} * \frac{\partial V_1}{\partial w_{11}}$$

$$\circledast \frac{\partial Y}{\partial V_5} = Y(1-Y) //$$

$$\circledast \frac{\partial V_5}{\partial Y_3} = W_{53} //$$

$$\circledast \frac{\partial Y_3}{\partial V_3} = Y_3(1-Y_3) //$$

$$\circledast \frac{\partial V_3}{\partial Y_1} = \frac{\partial}{\partial Y_1}(w_{31}Y_1 + w_{32}Y_2 + b_3 * 1)$$

$$= w_{31} + 0 + 0$$

$$= w_{31} //$$

$$\circledast \frac{\partial V_1}{\partial w_{11}} = \frac{\partial}{\partial w_{11}}(w_{11}x_1 + w_{12}x_2 + b_1 * 1)$$

$$= x_1 + 0 + 0$$

$$= x_1 //$$

$$\circledast \frac{\partial Y_1}{\partial V_1} = \frac{\partial}{\partial V_1}\left(\frac{1}{1+e^{-V_1}}\right) = Y_1(1-Y_1)$$

$$= Y_1(1-Y_1) //$$

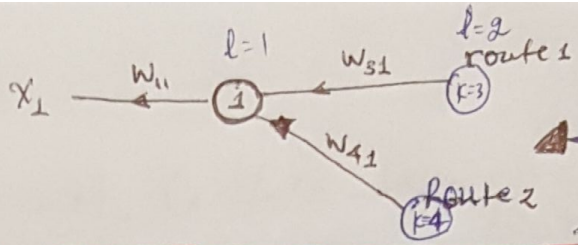
$$\therefore \frac{\partial Y}{\partial w_{11}} = Y(1-Y) * W_{53} * Y_3(1-Y_3) * (w_{31}) * Y_1(1-Y_1) * x_1$$

$$= Y(1-Y)\, Y_1(1-Y_1) * x_1 * W_{53} * Y_3(1-Y_3) * w_{31}$$

↓ Generalizing it using the text-book's notation (Do the same for 2nd route)

$$\frac{\partial F(w,x)}{\partial w_{11}} = F(w,x)(1-F(w,x))\, Y_1(1-Y_1) * x_1 * W_{53} * Y_3(1-Y_3) * w_{31}$$

↓ Making it General.



$$\frac{\partial F(w,x)}{\partial w_{ji}^{(1)}} = F(w,x)(1-F(w,x))\, y_j^{(1)}(1-y_j^{(1)})\, x_i \left[\sum_k w_{1k}^{(3)}\, y_k^{(2)}(1-y_k^{(2)})\, w_{kj}^{(2)}\right] \quad (3)$$

(→ Do the same for the 2nd route)

Therefore,

Layer-2

① $$\frac{\partial F(w,x)}{\partial w_{1k}^{(3)}} = F(w,x)\left(1-F(w,x)\right) y_k^{(2)} \quad \cdots (1)$$

Layer 1 → $y_j^{(1)}$; Layer 3 → $w_{1k}^{(3)}$

② $$\frac{\partial F(w,x)}{\partial w_{kj}^{(2)}} = F(w,x)\left(1-F(w,x)\right) y_k^{(2)}(1-y_k^{(2)})\, y_j^{(1)}\, w_{1k}^{(3)} \quad \cdots (2)$$

③ $$\frac{\partial F(w,x)}{\partial w_{ji}^{(1)}} = F(w,x)(1-F(w,x))\, y_j^{(1)}(1-y_j^{(1)})\, x_i \left[\sum_k w_{1k}^{(3)}\, y_k^{(2)}(1-y_k^{(2)})\, w_{kj}^{(2)}\right] \quad \cdots (3)$$

$$\frac{\partial F(w,x)}{\partial w_{1k}^{(3)}} = F(w,x)(1-F(w,x))\, y_k^{(2)} \quad \text{for } l=3,\ l=\text{layer}$$

①

$$\frac{\partial F(w,x)}{\partial w_{kj}^{(2)}} = F(w,x)(1-F(w,x))\, y_k^{(2)}(1-y_k^{(2)})\, y_j^{(1)}\, w_{1k}^{(3)} \quad \text{for } l=2,\ l=\text{layer}$$

②

$$\frac{\partial F(w,x)}{\partial w_{ji}^{(1)}} = F(w,x)(1-F(w,x))\, y_j^{(1)}(1-y_j^{(1)})\, x_i \left[\sum_k w_{1k}^{(3)}\, y_k^{(2)}(1-y_k^{(2)})\, w_{kj}^{(2)}\right] \quad \text{for } l=3,\ l=\text{layer}$$

③

End of

8.2. IRIS DATA Classification

```
---------Iris-ANN-Model Training Result Statistics----------
************************************************************
----------Training the ANN Iris_model... running now ------
Epoch 50 | Loss: 23.5968
Epoch 100 | Loss: 16.2516
Epoch 150 | Loss: 13.1641
Epoch 200 | Loss: 9.87121
Epoch 250 | Loss: 7.69699
Epoch 300 | Loss: 6.21547
Epoch 350 | Loss: 5.40875
Epoch 400 | Loss: 4.86537
Epoch 450 | Loss: 4.46044
Epoch 500 | Loss: 4.15025

---------------------End of Training----------------------
************************************************************

-------------------**********----------------------------
-----------Iris-ANN-Model Prediction Statistics-------------
************************************************************

Actual Values                Predicted Values
-------------                ----------------
[ 1.  0.  0.]                [[ 1.  0.  0.]]
[ 1.  0.  0.]                [[ 1.  0.  0.]]
[ 1.  0.  0.]                [[ 1.  0.  0.]]
[ 0.  0.  1.]                [[ 0.  0.  1.]]
[ 0.  0.  1.]                [[ 0.  0.  1.]]
[ 0.  1.  0.]                [[ 0.  1.  0.]]
[ 1.  0.  0.]                [[ 1.  0.  0.]]
[ 1.  0.  0.]                [[ 1.  0.  0.]]
[ 0.  1.  0.]                [[ 0.  1.  0.]]
[ 1.  0.  0.]                [[ 1.  0.  0.]]

NB: The Test dataset size used is 10!
---------------End of Prediction Section------------------
************************************************************
```

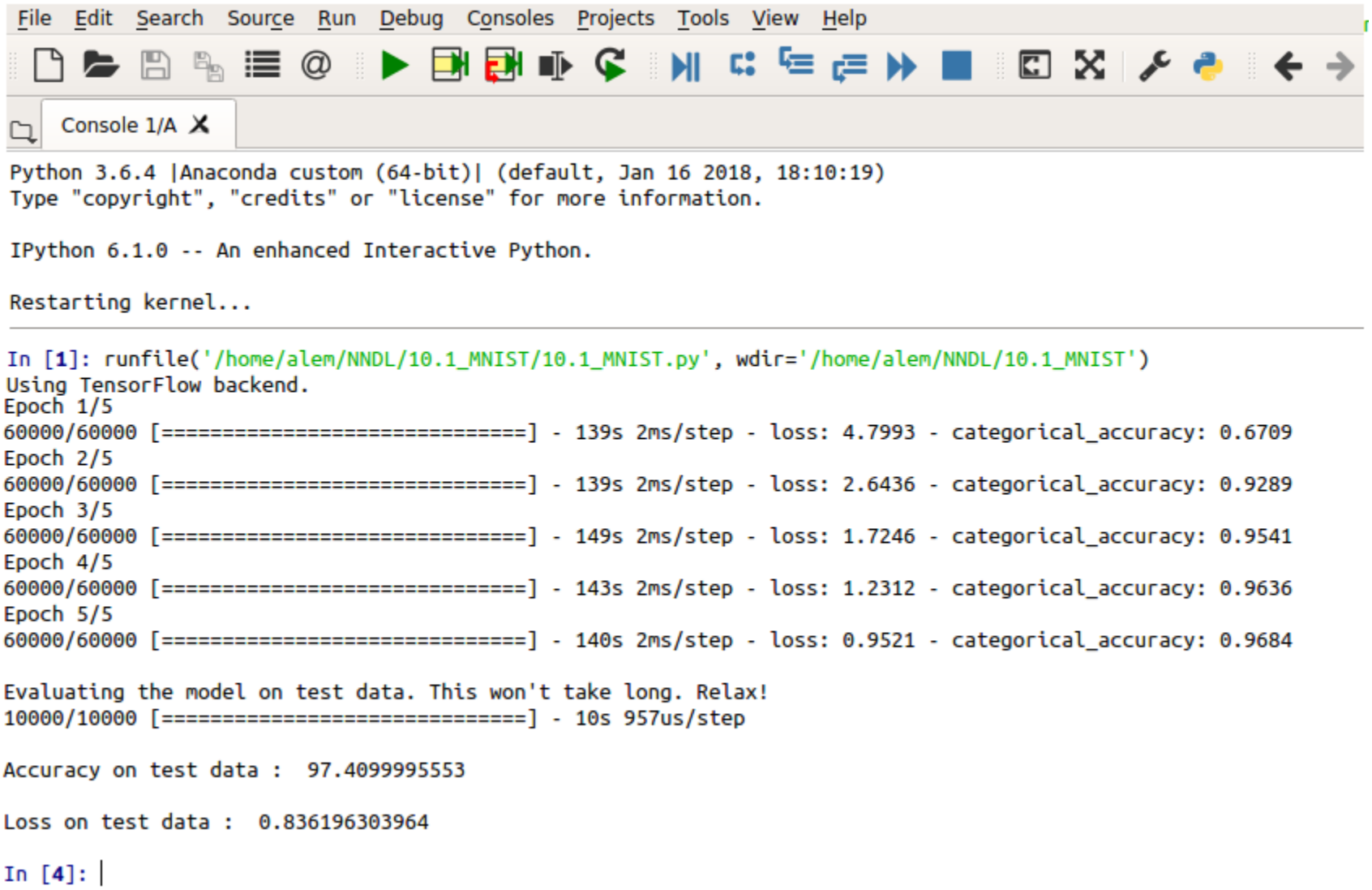
8.3. Cancer Tumor Classification

```
In [3]: runfile('/home/alem/NNDL/8.3_BreastCancer/8.3_BreastCancer.py', wdir='/home/alem/NNDL/8.3_BreastCancer')
INFO:tensorflow:Using default config.

        ..............................
INFO:tensorflow:global_step/sec: 361.032
INFO:tensorflow:loss = 0.0442343, step = 1001 (0.277 sec)
INFO:tensorflow:global_step/sec: 272.655
INFO:tensorflow:loss = 0.0411754, step = 1101 (0.367 sec)
INFO:tensorflow:global_step/sec: 290.464
INFO:tensorflow:loss = 0.0386572, step = 1201 (0.344 sec)
INFO:tensorflow:global_step/sec: 268.689
INFO:tensorflow:loss = 0.0354444, step = 1301 (0.372 sec)
INFO:tensorflow:global_step/sec: 307.387
INFO:tensorflow:loss = 0.0329505, step = 1401 (0.325 sec)
INFO:tensorflow:global_step/sec: 351
INFO:tensorflow:loss = 0.0294613, step = 1501 (0.285 sec)
INFO:tensorflow:global_step/sec: 368.23
INFO:tensorflow:loss = 0.0249967, step = 1601 (0.272 sec)
INFO:tensorflow:global_step/sec: 342.294
INFO:tensorflow:loss = 0.0216202, step = 1701 (0.292 sec)
INFO:tensorflow:global_step/sec: 270.278
INFO:tensorflow:loss = 0.0191258, step = 1801 (0.370 sec)
INFO:tensorflow:global_step/sec: 358.784
INFO:tensorflow:loss = 0.0174194, step = 1901 (0.279 sec)
INFO:tensorflow:Saving checkpoints for 2000 into /tmp/cancer_model/model.ckpt.
INFO:tensorflow:Loss for final step: 0.0156143.
```

Test Dataset

```
------------------**********----------------------
-------Five New Cancer Tumor Prediction Statistics---------  [[5, 10, 8, 4, 7, 4, 8, 11, 2],
************************************************************   [2, 1, 2, 1, 2, 1, 2, 1, 1],
New Samples, Class Predictions:    [1, 0, 1, 1, 0]             [8, 4, 5, 1, 2, 4, 7, 2, 1],
                                                               [8, 10, 10, 8, 3, 10, 9, 7, 1],
---------------End of Prediction Section------------------    [5, 1, 1, 1, 1, 1, 1, 1, 2]], dtype=np.float32)
************************************************************

In [4]:
```

10.1 MNIST Classification



10.2. EMNIST Classification

File Edit Search Source Run Debug Consoles Projects Tools View Help

/home/alem/NNDL/10

Console 1/A

```
Python 3.6.4 |Anaconda custom (64-bit)| (default, Jan 16 2018, 18:10:19)
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

Restarting kernel...

In [1]: runfile('/home/alem/NNDL/10.2_EMNIST/10.2_EMNIST.py', wdir='/home/alem/NNDL/10.2_EMNIST')
Using TensorFlow backend.
Train on 124800 samples, validate on 20800 samples
Epoch 1/20
124800/124800 [==============================] - 680s 5ms/step - loss: 0.8799 - acc: 0.7365 - val_loss: 0.3624 - val_acc: 0.8875
Epoch 2/20
124800/124800 [==============================] - 669s 5ms/step - loss: 0.4941 - acc: 0.8473 - val_loss: 0.2914 - val_acc: 0.9098
Epoch 3/20
124800/124800 [==============================] - 646s 5ms/step - loss: 0.4104 - acc: 0.8717 - val_loss: 0.2642 - val_acc: 0.9168
Epoch 4/20
124800/124800 [==============================] - 675s 5ms/step - loss: 0.3717 - acc: 0.8839 - val_loss: 0.2504 - val_acc: 0.9217
Epoch 5/20
```

.....

Test Image used after the 20th epoch

10.3. CIFAR-10 Image Classification

Name	Type	Size	Value
acc	float64	1	74.530000000000001
batch_xs	float64	(16, 3072)	array([[0.1254902 , 0.12941176, …
batch_ys	float64	(16, 10)	array([[0., 0., 0., ..., 1., … [0., 0., 0., ...
class_name	str	1	(9) truck
class_numbers	list	10	[' (0)', ' (1)', ' (2)', ' (3)', '…
cm	int64	(10, 10)	array([[729, 21, 44, ..., 9, … [8, 852, 3, ...
correct	bool	(10000,)	ndarray object of numpy module
correct_numbers	int64	1	7453

Help | Variable explorer | File explorer

IPython console

Console 1/A

```
Accuracy on Test-Set: 74.53% (7453 / 10000)
[729  21  44  47  22   9   9   9  74  36] (0) airplane
[  8 852   3  17   2   7   9   1  29  72] (1) automobile
[ 43   0 638  81  82  52  54  27  12  11] (2) bird
[ 20  10  53 599  82 117  65  34   6  14] (3) cat
[ 13   1  43  56 730  30  45  66  13   3] (4) deer
[  9   7  60 173  61 588  41  45   8   8] (5) dog
[  1   1  39  69  35  16 827   3   6   3] (6) frog
[  9   0  32  36  56  42   8 804   1  12] (7) horse
[ 33  17  12  27   7   3  10   3 863  25] (8) ship
[ 20  75   7  36   6   9   3   7  14 823] (9) truck
 (0) (1) (2) (3) (4) (5) (6) (7) (8) (9)

In [3]:
```