RYERSON UNIVERSITY

Faculty of Engineering, Architecture, and Science

Department of Electrical and Computer Engineering

Course Number	EE8603
Course Title	Neural Networks and Deep Learning
Semester/Year	Summer/2018

Instructor	Dr. Kandasamy Illanko

Assignment No.	**3**

Assignment Title	Convex Optimization & Multi-layer NNs

Submission Date	12^{th} June 2018
Due Date	12^{th} June 2018

Student Name	Muhammad Obaidullah
Student ID.	500671408
Signature*	

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: www.ryerson.ca/senate/current/pol60.pdf.

1 EXERSICE 5.2.4

1.1 MATLAB CODE

```
N = 100;
s = 0.1; % Step size
d = 0.01; % Stopping condition distance
x(1) = 1; % Initial point x coordinate
y(1) = 1; % Initial point y coordinate
k = 1;
for i = 1:N
  xinc = -s*(4*(x(i)-1)^3-1); % Partial derivative of f with respect to x
  yinc = -s*2*(y(i)-2); % Partial derivative of f with respect to x
  if(xinc^2+yinc^2<d^2)
    break % Stopping condition reached, come out of loop
  else
    x(i+1) = x(i) + xinc; % Step towards steep
    y(i+1) = y(i) + yinc; % Step towards steep
    k = k + 1; % Iteration count increment
  end
end
figure
scatter(x,y,3,'b','filled');
hold on
i = 1:k;
plot(i, (x-1).^4-x+(y-2).^2);
fprintf('Iterations: %i, Final distance: %.4f\n', k, xinc^2+yinc^2);
fprintf('x_min: %.2f, y_min: %.2f\n', x(length(x)), y(length(y)));
fprintf('Minimum Value: %.2f\n', (x(length(x))-1).^4-x(length(x))+(y(length(y))-2).^2);
```

1.2 FIGURES

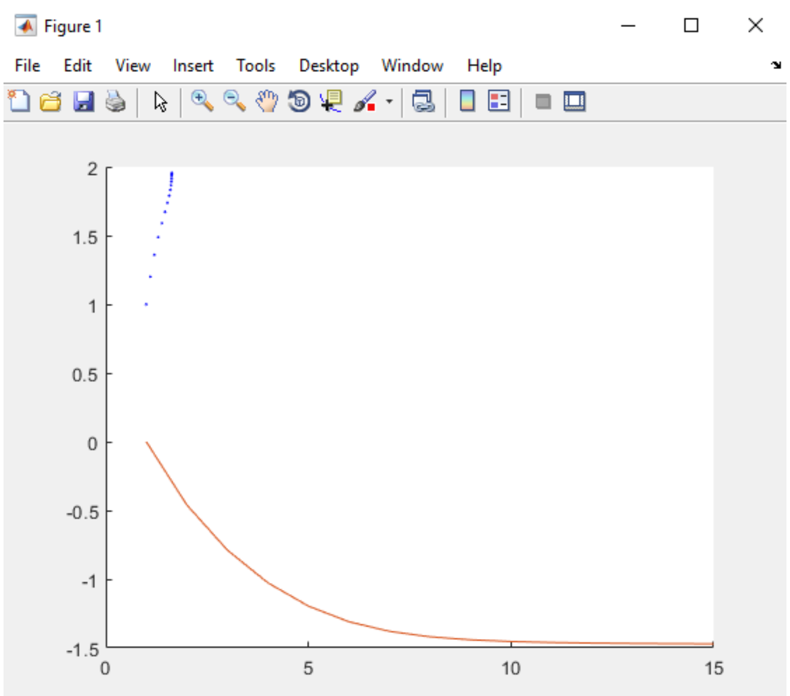


Figure 1.1: Progress is shown in red line while the x and y progress is shown by blue dots.

1.3 Output

```
>> exersice_5_2_4
Iterations: 15, Final distance: 0.0001
x_min: 1.63, y_min: 1.96
Minimum Value: -1.47
```

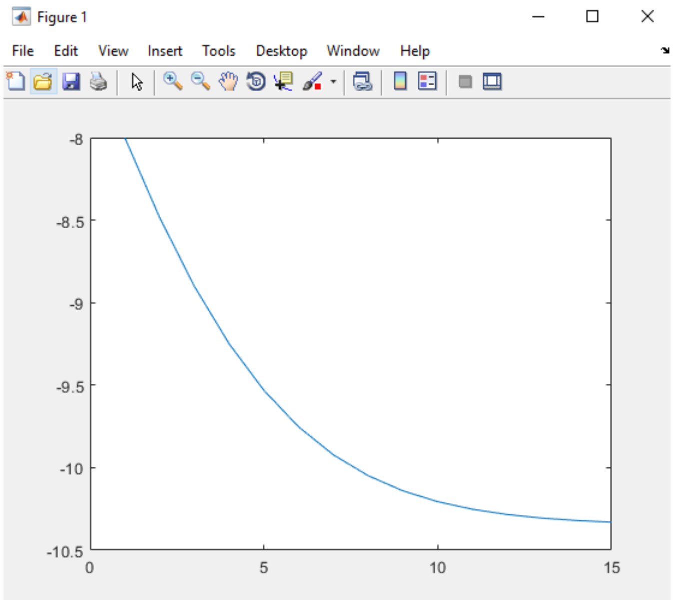
Figure 1.2: Program output showing final x value as 1.63 and y value as 1.96.

2 Exersice 5.2.5

2.1 Solution to part a

$$f(x,y) = x^2(x-2)(x+3) + y^2(y-2)(y+3)$$
$$= (x^3 - 2x^2)(x+3) + (y^3 - 2y^2)(y+3)$$
$$= x^3(x+3) - 2x^2(x+3) + y^3(y+3) - 2y^2(y+3)$$
$$= x^4 + 3x^3 - 2x^3 - 6x^2 + y^4 + 3y^3 - 2y^3 - 6y^2$$
$$= x^4 + x^3 - 6x^2 + y^4 + y^3 - 6y^2$$
$$\frac{\partial f}{\partial x} = 4x^3 + 3x^2 - 12x$$
$$\frac{\partial f}{\partial y} = 4y^3 + 3y^2 - 12y$$

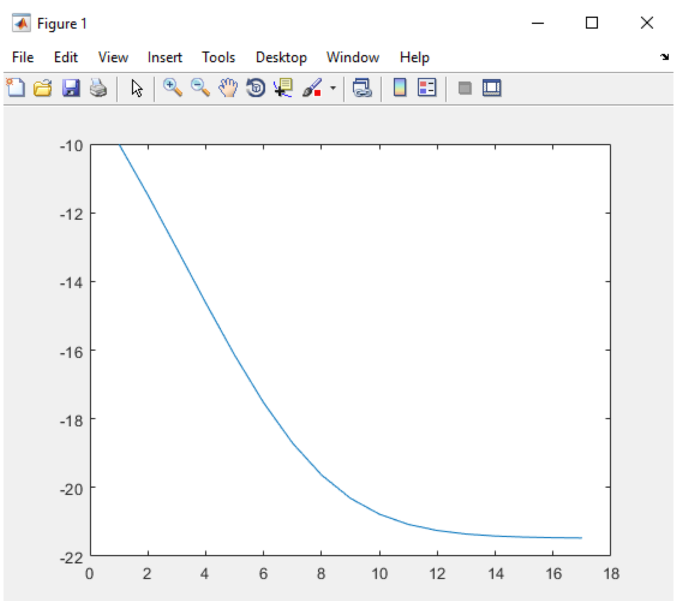
```
N = 100;
s = 0.01; % Step size
d = 0.01; % Stopping condition distance
x(1) = 1; % Initial point x coordinate
y(1) = 1; % Initial point y coordinate
k = 1;
for i = 1:N
  xinc = -s*(4*x(i)^3+3*x(i)^2-12*x(i)); % Partial derivative of f with respect to x
  yinc = -s*(4*y(i)^3+3*y(i)^2-12*y(i)); % Partial derivative of f with respect to y
  if(xinc^2+yinc^2<d^2)
    break % Stopping condition reached, come out of loop
  else
    x(i+1) = x(i) + xinc; % Step towards steep
    y(i+1) = y(i) + yinc; % Step towards steep
    k = k + 1; % Iteration count increment
  end
end
figure
%scatter(x,y,3,'b','filled');
%hold on
i = 1:k;
plot(i, x.^2.*(x-2).*(x+3)+y.^2.*(y-2).*(y+3));
fprintf('Iterations: %i, Final distance: %.4f\n', k, xinc^2+yinc^2);
fprintf('x_min: %.2f, y_min: %.2f\n', x(length(x)), y(length(y)));
fprintf('Minimum Value: %.2f\n', (x(length(x))).^2*(x(length(x))-2)*(x(length(x))+3)+y(length(y))
    .^2*(y(length(y))-2)*(y(length(y))+3));
```




```
>> exersice_5_2_5
Iterations: 15, Final distance: 0.0001
x_min: 1.37, y_min: 1.37
Minimum Value: -10.33
```

Figure 2.1: Function f plotted against number of iterations with initial point (1,1) and smooth step size of 0.01.

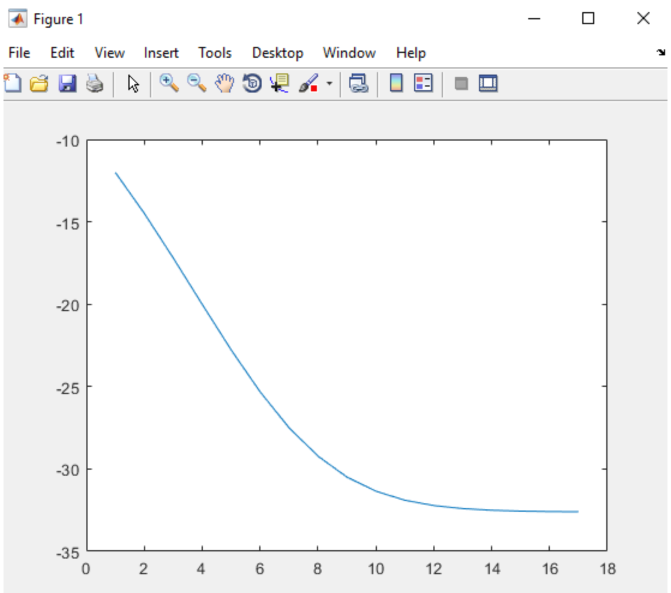
2.2 SOLUTION TO PART B




```
>> exersice_5_2_5
Iterations: 17, Final distance: 0.0001
x_min: 1.38, y_min: -2.13
Minimum Value: -21.47
```

Figure 2.2: Function f plotted against number of iterations with initial point (1,-1).

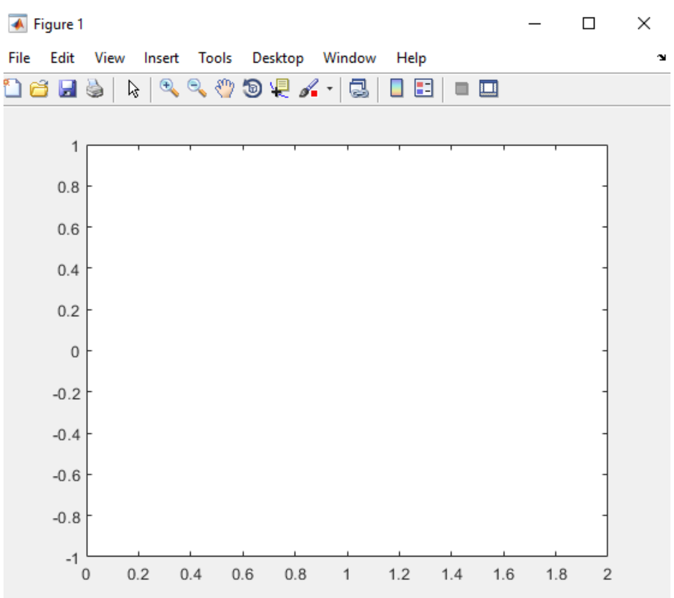
2.3 SOLUTION TO PART C




```
>> exersice_5_2_5
Iterations: 17, Final distance: 0.0001
x_min: -2.13, y_min: -2.13
Minimum Value: -32.60
```

Figure 2.3: Function f plotted against number of iterations with initial point (-1,-1).

2.4 Solution to part D




```
>> exersice_5_2_5
Iterations: 1, Final distance: 0.0000
x_min: 0.00, y_min: 0.00
Minimum Value: -0.00
```

Figure 2.4: Function f plotted against number of iterations with initial point (0,0).

2.5 Solution to part E

```
x = -3.2:0.1:2.2;
y = -3.2:0.1:2.2;
[X, Y] = meshgrid(x,y);
Z = X.^2.*(X-2).*(X+3)+Y.^2.*(Y-2).*(Y+3);
surf(X,Y,Z);
```

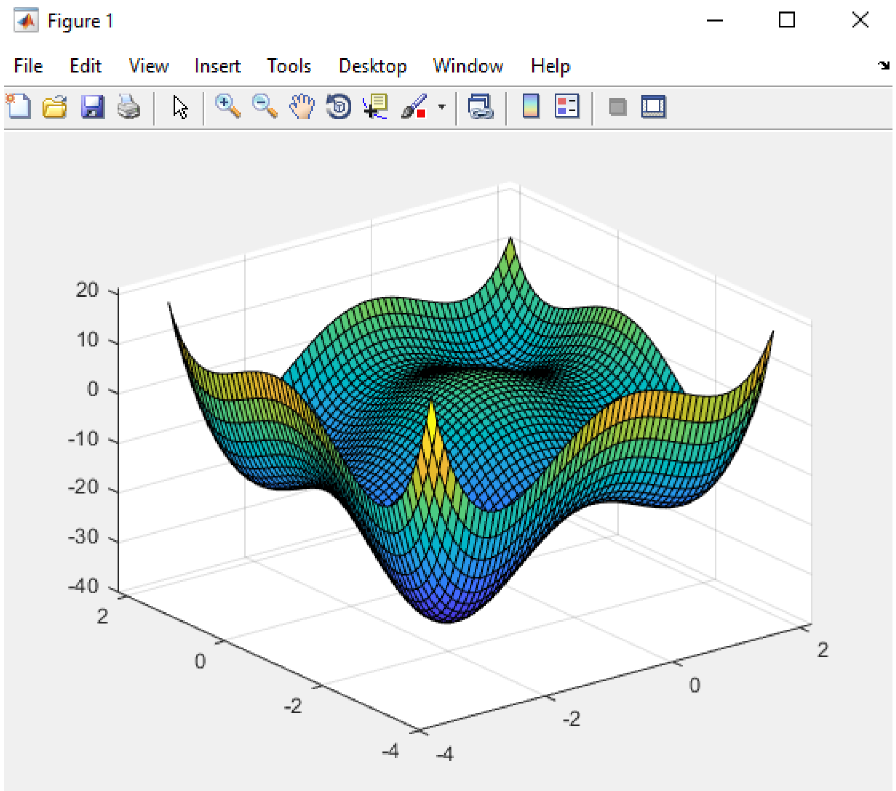


Figure 2.5: 3D surf plot showing function f.

3 EXERSICE 6.4.1

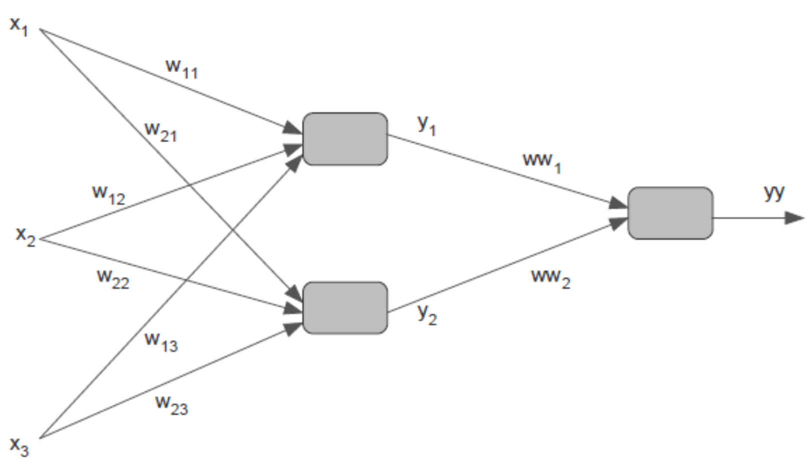


Figure 3.1: A 3/2-1 Neural Network.

$$y_1 = b_1 + w_{11}x_1 + w_{12}x_2 + w_{13}x_3$$
$$y_2 = b_2 + w_{21}x_1 + w_{22}x_2 + w_{23}x_3$$
$$yy = bb + ww_1y_1 + ww_2y_2$$
$$f_o = \begin{cases} 0 & if \quad yy < 0 \\ 1 & if \quad yy \geq 0 \end{cases}$$
$$f_e = (t - f_o)^2$$

3.1 Update for w_{12}

$$w_{12} := w_{12} - \alpha \frac{\partial f_e}{\partial w_{12}}$$

To calculate the partial derivative:-

$$\frac{\partial f_e}{\partial w_{12}} = \frac{\partial f_e}{\partial f_o} \frac{\partial f_o}{\partial yy} \frac{\partial yy}{\partial y_1} \frac{\partial y_1}{\partial w_{12}}$$

$$= -2(t - f_o)(1)(ww_1)(x_2)$$

$$= -2eww_1x_2$$

$$w_{12} := w_{12} + \alpha eww_1x_2$$

3.2 Update for w_{23}

$$w_{23} := w_{23} - \alpha \frac{\partial f_e}{\partial w_{23}}$$

To calculate the partial derivative:-

$$\frac{\partial f_e}{\partial w_{23}} = \frac{\partial f_e}{\partial f_o} \frac{\partial f_o}{\partial yy} \frac{\partial yy}{\partial y_2} \frac{\partial y_2}{\partial w_{23}}$$

$$= -2(t - f_o)(1)(ww_2)(x_3)$$

$$= -2eww_2x_3$$

$$w_{23} := w_{23} + \alpha eww_2x_3$$

4 Exersice 6.4.2

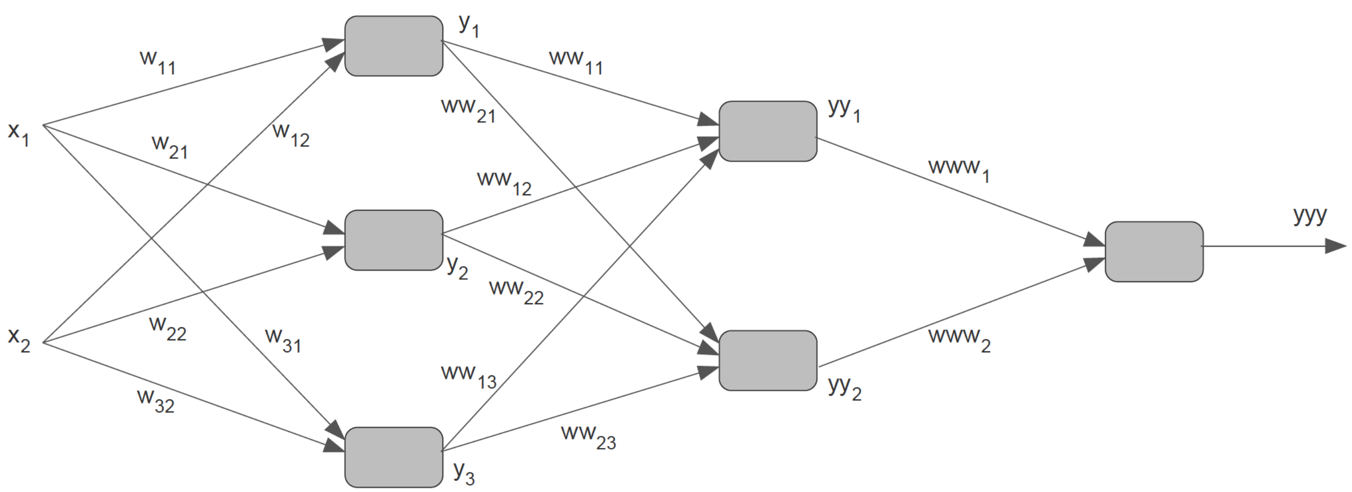


Figure 4.1: A 2/3-2-1 Neural Network.

$$y_1 = b_1 + w_{11}x_1 + w_{12}x_2$$

$$y_2 = b_2 + w_{21}x_1 + w_{22}x_2$$

$$y_3 = b_3 + w_{31}x_1 + w_{32}x_2$$

$$yy_1 = bb_1 + ww_{11}y_1 + ww_{12}y_2 + ww_{13}y_3$$

$$yy_2 = bb_2 + ww_{21}y_1 + ww_{22}y_2 + ww_{23}y_3$$

$$yyy = bbb + www_1yy_1 + www_2yy_2$$

4.1 UPDATE FOR www_2

$$www_2 := www_2 - \alpha \frac{\partial f_e}{\partial www_2}$$

To calculate the partial derivative:-

$$\frac{\partial f_e}{\partial www_2} = \frac{\partial f_e}{\partial f_o} \frac{\partial f_o}{\partial yyy} \frac{\partial yyy}{\partial www_2}$$
$$= -2(t - f_o)(1)(yy_2)$$
$$= -2eyy_2$$
$$www_2 := www_2 + \alpha eyy_2$$

4.2 UPDATE FOR ww_{21}

$$ww_{21} := ww_{21} - \alpha \frac{\partial f_e}{\partial ww_{21}}$$

To calculate the partial derivative:-

$$\frac{\partial f_e}{\partial ww_{21}} = \frac{\partial f_e}{\partial f_o} \frac{\partial f_o}{\partial yyy} \frac{\partial yyy}{\partial yy_2} \frac{\partial yy_2}{\partial ww_{21}}$$
$$= -2(t - f_o)(1)(www_2)(y_1)$$
$$= -2ewww_2 y_1$$
$$www_2 := www_2 + \alpha ey_1$$

4.3 UPDATE FOR w_{12}

$$w_{12} := w_{12} - \alpha \frac{\partial f_e}{\partial w_{12}}$$

To calculate the partial derivative:-

$$\frac{\partial f_e}{\partial w_{12}} = \frac{\partial f_e}{\partial f_o} \frac{\partial f_o}{\partial yyy} \Big(\frac{\partial yyy}{\partial yy_1} \frac{\partial yy_1}{\partial y_1} \frac{\partial y_1}{\partial w_{12}} + \frac{\partial yyy}{\partial yy_2} \frac{\partial yy_2}{\partial y_1} \frac{\partial y_1}{\partial w_{12}} \Big)$$
$$= -2(t - f_o)(1)(www_1 ww_{11} x_2 + www_2 ww_{21} x_2)$$
$$= -2e(www_1 ww_{11} x_2 + www_2 ww_{21} x_2)$$
$$www_2 := www_2 + \alpha e(www_1 ww_{11} x_2 + www_2 ww_{21} x_2)$$

5 EXERSICE 6.5.1

$$y_1 = b_1 + w_{11}x_1 + w_{12}x_2 + w_{13}x_3$$
$$y_2 = b_2 + w_{21}x_1 + w_{22}x_2 + w_{23}x_2$$
$$y_3 = b_3 + w_{31}x_1 + w_{32}x_2 + w_{33}x_3$$
$$xx_1 = f(y_1)$$
$$xx_2 = f(y_2)$$
$$xx_3 = f(y_3)$$
$$yy_1 = bb_1 + ww_{11}xx_1 + ww_{12}xx_2 + ww_{13}xx_3$$
$$yy_2 = bb_2 + ww_{21}xx_1 + ww_{22}xx_2 + ww_{23}xx_3$$
$$xxx_1 = f(yy_1)$$
$$xxx_2 = f(yy_2)$$
$$yyy = www_1 xxx_1 + www_2 xxx_2 + bbb$$

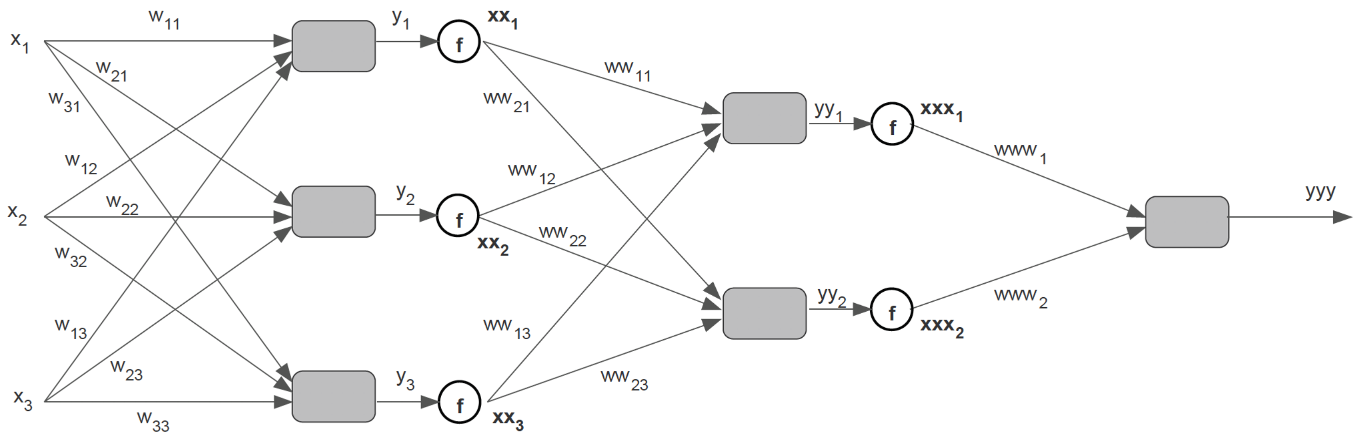


Figure 5.1: A 3/3-2-1 Neural Network.

5.1 UPDATE FOR ww_{21}

$$ww_{21} := ww_{21} - \alpha \frac{\partial f_e}{\partial ww_{21}}$$

To calculate the partial derivative:-

$$\frac{\partial f_e}{\partial ww_{21}} = \frac{\partial f_e}{\partial f_o} \frac{\partial f_o}{\partial yyy} \frac{\partial yyy}{\partial xxx_2} \frac{\partial xxx_2}{\partial yy_2} \frac{\partial yy_2}{\partial ww_{21}}$$

Denoting $\frac{\partial xxx_2}{yy_2} = dxxx_2$:

$$= -2(t - f_o)(1)(www_1 dxxx_2 xx_2)$$

$$= -2e(www_1 dxxx_2 xx_2)$$

$$www_2 := www_2 + \alpha e(www_1 dxxx_2 xx_2)$$

5.2 UPDATE FOR w_{13}

$$w_{13} := w_{13} - \alpha \frac{\partial f_e}{\partial w_{13}}$$

To calculate the partial derivative:-

$$\frac{\partial f_e}{\partial w_{13}} = \frac{\partial f_e}{\partial f_o} \frac{\partial f_o}{\partial yyy} \Big(\frac{\partial yyy}{\partial xxx_1} \frac{\partial xxx_1}{\partial yy_1} \frac{\partial yy_1}{\partial xx_1} \frac{\partial xx_1}{\partial y_1} \frac{\partial y_1}{\partial w_{13}} + \frac{\partial yyy}{\partial xxx_2} \frac{\partial xxx_2}{\partial yy_2} \frac{\partial yy_2}{\partial xx_1} \frac{\partial xx_1}{\partial y_1} \frac{\partial y_1}{\partial w_{13}} \Big)$$

Denoting $\frac{\partial xxx_1}{yy_1} = dxxx_1$, $\frac{\partial xxx_2}{yy_2} = dxxx_2$, $\frac{\partial xx_1}{y_1} = dxx_1$, $\frac{\partial xx_2}{yy_1} = dxx_2$:

$$= -2(t - f_o)(1)(www_1 dxxx_1 ww_{11} dxx_1 x_3 + www_2 dxxx_2 ww_{21} dxx_1 x_3)$$

$$= -2e(www_1 dxxx_1 ww_{11} dxx_1 x_3 + www_2 dxxx_2 ww_{21} dxx_1 x_3)$$

$$www_2 := www_2 + \alpha e(www_1 dxxx_1 ww_{11} dxx_1 x_3 + www_2 dxxx_2 ww_{21} dxx_1 x_3)$$

6 EXERSICE 6.6.1

7 EXERSICE 6.6.2

8 EXERSICE 6.6.3

9 EXERSICE 6.6.4

10 QUESTION 10

In this problem you will prove the 2 by 2 version of an important theorem that we will be using in Chapter 8 to derive the matrix version of the back propagation algorithm. The idea is to write the derivatives of the output of a neuron with respect to the weights of a previous layer as a matrix. This matrix of the derivatives is to have the exact same dimensions as the weights written as a matrix.

Lets call the output by yy and the weight matrix $W = [w_{ij}]$. The derivative matrix of yy with respect to the entries of W is denoted by $D_{W_{yy}}$. In this matrix $D_{W_{yy}}$, the derivative $\frac{\partial yy}{\partial w_{ij}}$ will be written at the same position as the entry w_{ij} in the matrix W, that is, at row i, column j. Suppose $A = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$, $W = \begin{bmatrix} ww_{11} & ww_{12} \\ ww_{21} & ww_{22} \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and $yy = AWx$

You will now prove $D_{W_{yy}} = [xA]^T$ by following the steps below:

(a) Write out AWx to express yy as a function of a_i's, w_ij's and x_i's

(b) Find all four derivatives $\frac{\partial yy}{\partial w_{ij}}$

(c) Arrange the derivatives in the appropriate order to obtain $D_{W_{yy}}$. For example, $\frac{\partial yy}{\partial w_{11}}$ should be written at the position row one column one, $\frac{\partial yy}{\partial w_{12}}$ should be written at position row one column two and so on.

(d) Express $[xA]^T$ as a function of a_i's, and x_i's and conclude that $D_{W_{yy}} = [xA]^T$

10.1 SOLUTION TO PART A

$$yy = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} ww_{11} & ww_{12} \\ ww_{21} & ww_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} a_1 ww_{11} + a_2 ww_{21} & a_1 ww_{12} + a_2 ww_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \Big(a_1 ww_{11} + a_2 ww_{21}\Big) x_1 + \Big(a_1 ww_{12} + a_2 ww_{22}\Big) x_2$$

$$= a_1 ww_{11} x_1 + a_2 ww_{21} x_1 + a_1 ww_{12} x_2 + a_2 ww_{22} x_2$$

10.2 SOLUTION TO PART B

$$\frac{\partial yy}{\partial w_{11}} = a_1 x_1$$

$$\frac{\partial yy}{\partial w_{12}} = a_1 x_2$$

$$\frac{\partial yy}{\partial w_{21}} = a_2 x_1$$

$$\frac{\partial yy}{\partial w_{22}} = a_2 x_2$$

10.3 SOLUTION TO PART C

$$\begin{bmatrix} \frac{\partial yy}{\partial w_{11}} & \frac{\partial yy}{\partial w_{12}} \\ \frac{\partial yy}{\partial w_{21}} & \frac{\partial yy}{\partial w_{22}} \end{bmatrix} = \begin{bmatrix} a_1 x_1 & a_1 x_2 \\ a_2 x_1 & a_2 x_2 \end{bmatrix}$$

$$\therefore [D_{W_{yy}}] = \begin{bmatrix} a_1 x_1 & a_1 x_2 \\ a_2 x_1 & a_2 x_2 \end{bmatrix}$$

10.4 Solution to part D

$$[xA]^T = \left[\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} a_1 & a_2 \end{bmatrix} \right]^T$$

$$= \begin{bmatrix} a_1x_1 & a_2x_1 \\ a_1x_2 & a_2x_2 \end{bmatrix}^T$$

$$= \begin{bmatrix} a_1x_1 & a_1x_2 \\ a_2x_1 & a_2x_2 \end{bmatrix}$$

$$\therefore D_{W_{yy}} = [xA]^T$$