Kieran Cosgrove Student #: 20226842

CISC/CMPE 452 /COGS 400

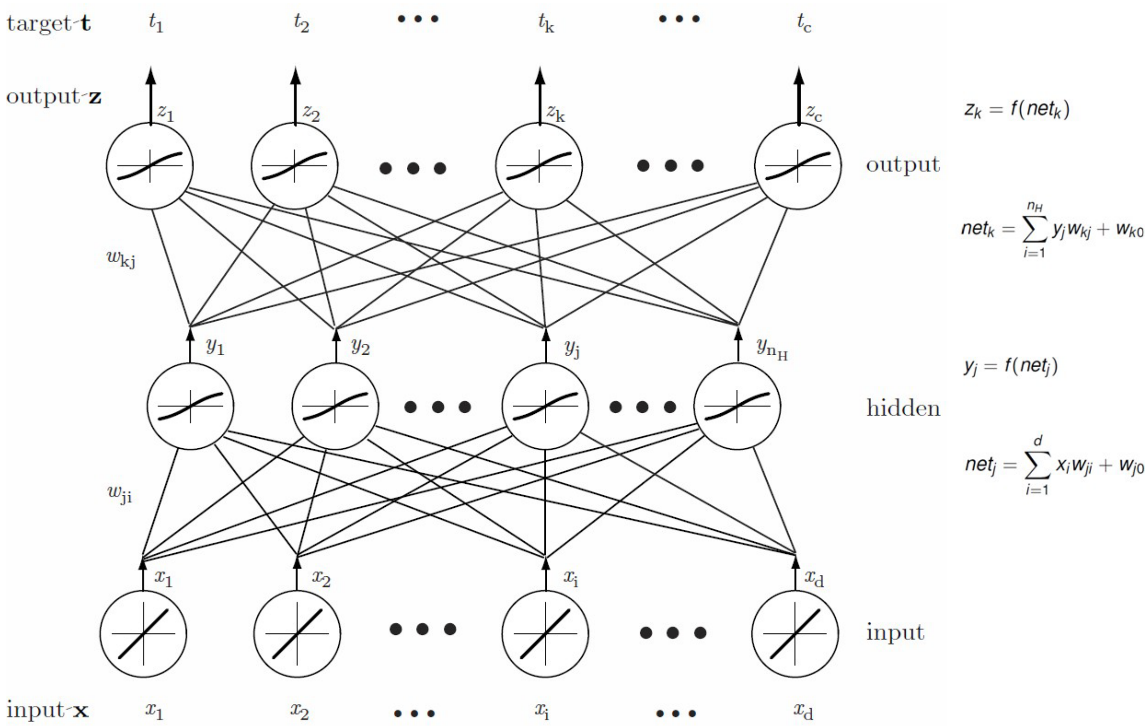
Assignment #2 – Theoretical Part

1. Given a neural network with two input neurons, x_1 and x_2, three hidden neurons, h_1, h_2 and h_3, and two output neurons, y_1 and y_2. The network acts as a classifier (2 classes). The following equations govern the network operation:

$$h_1 = \sigma(x_1 + 1) \qquad (1)$$
$$h_2 = \sigma(x_2 + 1) \qquad (2)$$
$$h_3 = \sigma(1 - x_1 - 2x_2) \qquad (3)$$
$$y_1 = \sigma(2.5 - h_1 - h_2 - h_3) \qquad (4)$$
$$y_2 = \sigma(h_1 + h_2 + h_3 - 2.5) \qquad (5)$$

where $\sigma(x) = 1$ if $x \geq 0$ and else $\sigma(x) = 0$ otherwise.
Draw the decision region for each class and the decision boundary.

2. Consider a three layer neural network whose structure is shown in Figure 1. You are required to calculate the sensitivity $\delta_k = -\frac{\partial J}{\partial \mathrm{net}_k}$ at the output node k, where J is the objective function to be minimized and net_k is the net activation of the output node k. We consider two cases, where the objective function J and the nonlinear activation function at the output layer are chosen differently.
   a. In the first case, J is chosen as the squared error $J(W) = \frac{1}{2}\left\|t - z\right\|^2$ where z_k is the prediction at the output node k and t_k is the corresponding target value. In the classification problem, only one t_k equals to 1 (corresponding the ground truth class) and all the other t_k's are all zeros. Sigmoid $\mathrm{f}(\mathrm{net}_k) = 1/(1 + \mathrm{e}^{-\mathrm{net}_k})$ is chosen as the activation function at the output layer. Calculate the sensitivity δ_k in terms t_k, z_k and net_k. Show that all the δ_k could be close to zero even if the prediction error is large and explain why this is bad.
   b. In the second case, the objective function is chosen as cross entropy, $J(W) = -\sum_{k=1}^{cC} t_k \, log(z_k)$ and the nonlinear activation function at the output layer is chosen as softmax $f(net_k) = e^{net_k} / \sum_{j=1}^{C} e^{net_j}$. Calculate the sensitivity $\delta_k$tk again. Prove that if the prediction error is large, at least one of the δ_k will be large.



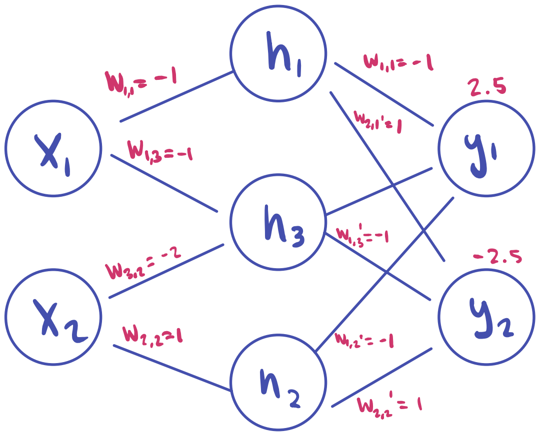
target t
t_1
t_2
t_k
t_c
output z
z_1
z_2
z_k
z_c
$z_k = f(net_k)$
output
w_{kj}
$net_k = \sum_{i=1}^{n_H} y_j w_{kj} + w_{k0}$
y_1
y_2
y_j
y_{n_H}
$y_j = f(net_j)$
hidden
w_{ji}
$net_j = \sum_{i=1}^{d} x_i w_{ji} + w_{j0}$
x_1
x_2
x_i
x_d
input
input x
x_1
x_2
x_i
x_d

Figure 1

Kieran Cosgrove
Student #: 20226841

Assignment 2 Theory

Question 1:



$\Theta = w_1 x_1 + w_2 x_2$

$x_1 = -\frac{w_2}{w_1} x_2 + \frac{\Theta}{w_1}$

$x_2 = -\frac{w_2}{w_1} x_1 + \frac{\Theta}{w_2}$

$h_1: x_1 \geq -1 \implies x_1 = -1$

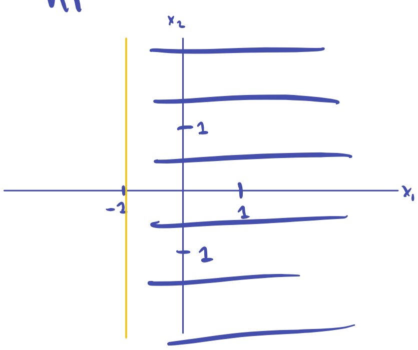
$h_3: 1 - x_1 - 2x = -\frac{1}{2}x_1 + \frac{1}{2}$

$h_2: x_2 \geq -1 \implies x_2 = -1$

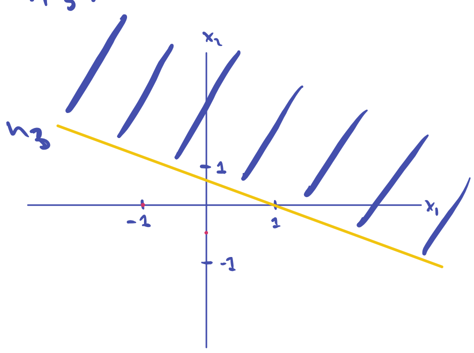
$\sigma(x) = 1$ when $x \geq 0$

$\sigma(x) = 0$ when $x < 0$

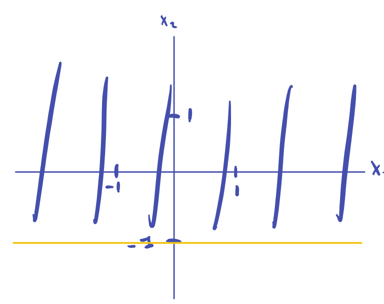
h_1:



h_3:

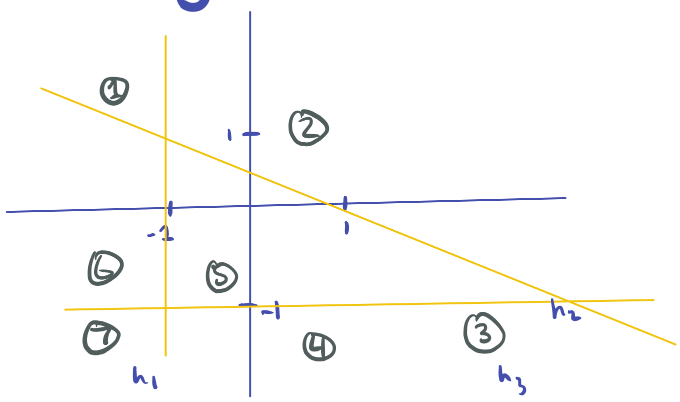


h_2:



Next page ...

Boundary regions:



There are 7 distinct boundaries.

Given the formulas:

$y_1 = 2.5 - h_1 - h_2 - h_3 \geq 0$ $\quad$ $y_2 = h_1 + h_2 + h_3 - 2.5 \geq 0$

$2.5 \geq h_1 + h_2 + h_3$ $\quad$ $h_1 + h_2 + h_3 \geq 2.5$

Here are the possible outcomes:

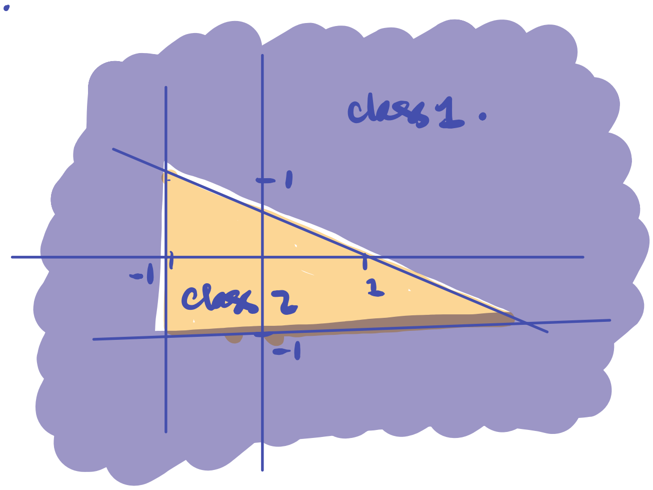
h_1	h_2	h_3	y_1	y_2	Quadrant
0	0	0	1	0	—
0	0	1	1	0	7
0	1	0	1	0	1
0	1	1	1	0	6
1	0	0	1	0	3
1	0	1	1	0	4
1	1	0	1	0	2
1	1	1	0	1	5

Thus there are two main classes:

class 1: $(y_1, y_2) = (1, 0)$

class 2: $(y_1, y_2) = (0, 1)$

Visualization:



Question 2:

(a.)

calculating sensitivity δ_k in terms of t_k, z_k, and net_k

- $\delta_k = -\frac{\partial J}{\partial net_k}$
- $J(w) = \frac{1}{2}||t-z||^2 = \frac{1}{2}\sum(t_k - z_k)^2$
- z_k = prediction at output node k
- t is the target value

$$J(w) = \frac{1}{2}\sum(t_k - f(net_k))^2$$
$$= \frac{1}{2}\sum\left(t_k - \frac{1}{1+e^{-net_k}}\right)^2$$
$$\delta_k = -\frac{1}{2}\frac{\partial}{\partial net_k}\left[\sum\left(t_k - \frac{1}{1+e^{-net_k}}\right)^2\right]$$
$$\delta_k = -\frac{1}{\cancel{2}}\sum \cancel{2}(t_k - z_k)\left(\frac{e^{-net_k}}{(1+e^{-net_k})^2}\right)$$
$$\delta_k = -(t_k - z_k)\left(\frac{e^{-net_k}}{(1+e^{-net_k})^2}\right)$$

$$\lim_{net_k \to \infty}(\delta_k) = 0 \quad \text{and} \quad \lim_{net_k \to -\infty}(\delta_k) = 0$$

Therefore, when the net_k value increases towards positive ∞, the sensitivity will approach zero. This is a problem because if the error is very large, the sensitivity will be close to zero. This can lead to stalled learning, preventing the network from effectively updating its weights and converging to a good solution.

(b)

$$J(w) = -\sum_{k=1}^{c} t_k \log(z_k)$$

softmax $f(net_k) = e^{net_k} / \sum_{j=1}^{c} e^{net_j}$

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k} \frac{\partial z_k}{\partial net_k}$$

$$\frac{\partial J}{\partial z_k} = \frac{\partial}{\partial z_k}\left(-\sum_{k=1}^{c} t_k \log(z_k)\right)$$

$$= -\frac{t_k}{z_k}$$

$$\frac{\partial z_k}{\partial net_k} = \frac{\partial}{\partial net_k}\left(\frac{e^{net_k}}{\sum_{j=1}^{c} e^{net_j}}\right)$$

$$= \frac{e^{net_k}}{e^{net_k} + \sum_{j=1, j\neq k}^{c} e^{net_j}} - \frac{e^{2net_k}}{\left(e^{net_k} + \sum_{j=1, j\neq k}^{c} e^{net_j}\right)^2}$$

$$= \frac{e^{net_k}}{\sum_{j=1}^{c} e^{net_j}}\left[1 - \frac{e^{net_k}}{\sum_{j=1}^{c} e^{net_j}}\right]$$

$$= z_k[1 - z_k]$$

Thus $\delta_k = t_k(1 - z_k)$

If the training error is large, then z_k approaches zero

$$\delta_k = 1(1 - z_k \to 0)$$

$$= 1 - z_k \to 0$$

Therefore, there is a k such that the prediction error will be large.