Homework 5 - Machine Learning.
Name - Anushree Jumade
(axj174430)

Q.1. For the given data ~~and~~ → x is attribute & y is class variable

a). Test sample $x = 4.2$
1-NN would be $x = 4$ & hence it would be classified as 'B'.

b). Test sample $x = 4.2$.
3-NN would make use of $x=3$, $x=4$, $x=5$ having corresponding labels - {A, B, A}
Majority is 'A' so $x=4.2$ would be labelled as 'A'.

c). Leave-one-out cross validation of 1-NN :-
For each datapt in X, if the nearest neighbour has a different label, then X would be misclassified.
For the given dataset :-
$x=3$ label - 'A' chooses $x=4$ as 1-NN whose label is 'B'.
① so $x=3$ would get classified as 'A'
② $x=4$ has $x=3$ & $x=5$ as 1NN. Choosing either would classify $x=4$ as 'A' so error.
③ $x=9$ chooses $x=08$ as 1-NN & gets labelled as ~~'A'~~.
④ $x=13$ chooses either of $x=12$ or $x=14$ as its 1NN & gets misclassified as $x=$ 'B'.

so in the given dataset of 18 pts, at 4 instances data pts could be misclassified for 1NN.
⑤ $x=5$ chooses $x=4$ as its 1NN & gets misclassified
~~Ans~~ :- ~~4/18~~ ⑥ $x=8$ chooses $x=9$ as its 1NN & misclassified
⑦ $x=12$ chooses $x=13$ & misclassified
⑧ $x=14$ chooses $x=13$ as its 1NN & gets classified as 'A'.

of the given 18 data pts → 8 cases where the misclassification happens.

Ans → $\underline{8/18}$.

(2)

	X1 = Acid durability	X_2 = Strength	Y = classification
	7	7	Bad
	7	4	Bad
	3	4	Good
	1	4	Good
Test	3	7	

To find 3-nearest neighbours, we find euclidean distance b/w given pt & the data pts.

$(x_1, x_2) \equiv (3, 7) \longrightarrow$ Test.

$d_{T1} = (7-3)^2 + (7-7)^2 = 4^2 \qquad \sqrt{d_{T1}} = 4$

$d_{T2} = (7-3)^2 + (7-4)^2 = 4^2 + 3^2 \qquad \sqrt{d_{T2}} = 5$

$d_{T3} = (3-3)^2 + (4-7)^2 = 3^2 \qquad ✓ \sqrt{d_{T3}} = 3$

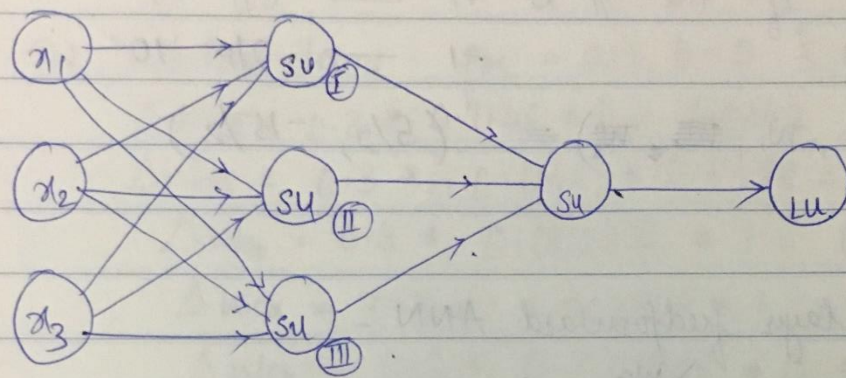
$d_{T4} = (1-3)^2 + (4-7)^2 = 2^2 + 3^2 \qquad ✓ \sqrt{d_{T4}} = 3.605$

so the three nearest neighbours are 1, 3 & 4 labelled as Bad, Good & Good respectively.

so the given new tissue would be classified as "<u>Good</u>"

③. Represent $f(x_1, x_2, x_3)$ using NN having linear units ~~or~~ and sign units

One solution is :- 2 hidden layes & 1 o/p node.



Function implemented at 'I' $\Rightarrow$ $\bar{x}_1 \wedge \bar{x}_2 \wedge x_3$.

If True $\rightarrow$ O/P = 1.
If False $\rightarrow$ O/P = 0.

Weights (w_0, w_1, w_2, w_3) could be $(-0.5, -1, -1 \,\&\, 1)$.

Function implemented at 'II' $\Rightarrow$ $\bar{x}_1 \wedge x_2 \wedge \bar{x}_3$ if True $\rightarrow$ O/P 1.
False $\rightarrow$ O/P 0.

$(w_0, w_1, w_2, w_3) \equiv (-0.5, -1, 1, -1)$.

Function implemented at 'III' $\rightarrow$ $x_1 \wedge \bar{x}_2 \wedge \bar{x}_3$ if True $\rightarrow$ O/P 1.
if False $\rightarrow$ O/P 0.

$(w_0, w_1, w_2, w_3) \equiv (-0.5, 1, -1, -1)$.

The I/P to the second hidden layer is the O/P of all the hidden nodes in the first hidden layer.. It should represent the following logic function :-

If at least one of the inputs is a +1, the unit O/Ps at +1 otherwise it O/Ps a -1. It thus represents an OR node

(w_0, w_1, w_2, w_3) here take the $(-0.5, 1, 1, 1)$. Values

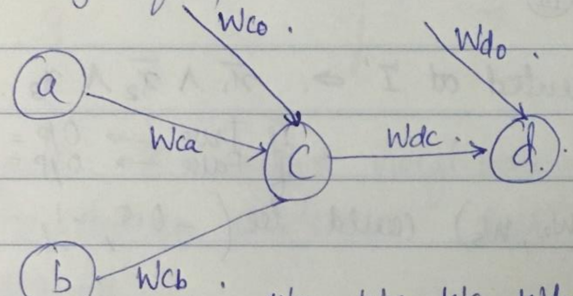
The I/P to the O/P node is the O/P of the node in the second hidden layer. It represents the logic function :-

If the I/P is +1 → O/P '-5'

-1 → O/P '10'

$(W_0, W_1, \cancel{W_2}, \cancel{W_3}) = (5/2, -15/2)$.

Q.4. Two-layer feedforward ANN :-



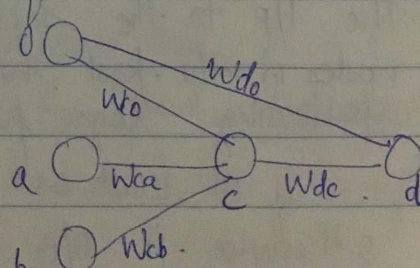
Initial values $\begin{array}{ccccc} W_{ca} & W_{cb} & W_{co} & W_{dc} & W_{do} \\ (0.1, & 0.1, & 0.1, & 0.1, & 0.1) \end{array}$

$\eta = 0.3$ & $\alpha = 0.9$. Training set =

a	b	d
1	0	1
0	1	0

Soln The network and the sigmoid activation function are as follows :-

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



Training example 1:-

$a = 1$ & $b = 0$.

$O_c = \sigma(0.1 \times 1 + 0.1 \times 0 + 0.1 \times 1) = \sigma(0.2) = 0.5498$

$O_d = \sigma(0.1 \times 0.5498 + 0.1 \times 1) = \sigma(0.15498) = 0.53867$.

The error terms ~~of~~ for the two neurons, noting $d=1$ for training sample '1'.

$$\partial d = 0.53867 * (1-0.53867) * (1-0.53867) = 0.1146.$$
$$\partial c = 0.5498 * (1-0.5498) * 0.1 * 0.1146 = 0.002836.$$

The correction terms for $= a=1,\ b=0\ \&\ \eta=0.3$:-

$$\Delta W_{do} = 0.3 * 0.1146 * 1 = 0.0342.$$
$$\Delta W_{dc} = 0.3 * 0.1146 * 0.5498 = 0.0189.$$
$$\Delta W_{co} = 0.3 * 0.002836 * 1 = 0.000849.$$
$$\Delta W_{ca} = 0.3 * 0.002836 * 1 = 0.000849.$$
$$\Delta W_{cb} = 0.3 * 0.002836 * 0 = 0$$

and the new weights become :-

$$W_{do} = 0.1 + 0.0342 = 0.1342.$$
$$W_{dc} = 0.1 + 0.0189 = 0.1189.$$
$$W_{co} = 0.1 + 0.000849 = 0.100849$$
$$W_{ca} = 0.1 + 0.000849 = 0.100849$$
$$W_{cb} = 0.1 + 0 = 0.1$$

Training example 2' :-

The O/P for $a=0\ \&\ b=1$ is :-

$$O_c = \sigma(0.100849 \times 0 + 0.1 \times 1 + 0.100849 \times 1)$$
$$= \sigma(0.200849) = 0.55.$$

$$O_d = \sigma(0.1189 * 0.55 + 0.1342 \times 1) = \sigma(0.1996)$$
$$= 0.5497.$$

$d = 0 \rightarrow$ Error term

$$\partial d = 0.5497 * (1-0.5497) * (0-0.5497)$$
$$= -0.1361$$

$$\partial c = 0.55 * (1-0.55) * 0.1189 * (-0.1361) = -0.004$$

Computing the correction terms as follows,

$a = 0$; $b = 1$; $\eta = 0.3$ & $\alpha = 0.9$

$\Delta W_{do} = 0.3 * (-0.1361) * 1 + 0.9 * 0.0342 = -0.01$

$\Delta W_{dc} = 0.3 \times (-0.1361) * 0.55 + 0.9 * 0.0189$
$= -0.0055$.

$\Delta W_{co} = 0.3 * (-0.004) * 1 + 0.9 * 0.000849$
$= -0.004$.

$\Delta W_{ca} = 0.3 \times (-0.004) * 0 + 0.9 * 0.000849$
$= 0.00086$

$\Delta W_{cb} = 0.3 * (-0.004) * 1 + 0.9 * 0 = -0.0012$.

and the new weights become :.

$W_{do} = 0.1342 - 0.01 = 0.1242$.

$W_{dc} = 0.1189 - 0.0055 = 0.1134$.

$W_{co} = 0.100849 - 0.0004 = 0.100809$.

$W_{ca} = 0.100849 + 0.00086 = 0.1016$.

$W_{cb} = 0.1 - 0.0012 = 0.0988$

5]. Original (as in Table 4.2)

$\delta_h \leftarrow O_h(1 - O_h) \sum W_{kh} \delta_k$.

$\delta_k \leftarrow O_k(1 - O_k)(t_k - O_k)$.

Assuming → $O = \tanh(\overline{w} * \overline{x})$.

$\tanh'(x) = 1 - \tanh^2(x)$

so the new updation rule becomes :.

$\partial_k \rightarrow (1 - O_k^2)(t_k - O_k)$.

$\partial_h \rightarrow (1 - O_h^2) \sum W_{kh} \partial_k$