Lastname: Riojas S.S. #: 766641137

Firstname: Alejandro

ISC4223 COMP METH DISCR PROB

Mid Term Examination

Fall 2020

Instructions for this test:

1. Put your name and SS at the top of each page, especially this page.
2. Read each problem carefully and be sure to provide the answer requested.
3. Use the provided pages only. Do *not* use any extra paper.
4. SHOW ALL WORK.
5. Do not cheat.

Upon completion of the test please sign the following statement:

I have neither given nor received aid from any unauthorized source during this exam.

Problem 1:

(a) What is the goal of dimensionality reduction techniques, such as PCA-for what are they used in practice?

(b) What does it mean if an attribute is irrelevant for a classification problem?

(c) Give an example of using PCA.

a) The goal of dimensionality reduction is to find the most statistically relevant attributes of a dataset and ~~find the~~ transform the dataset into a subspace of those attributes. In practice, they are used to lower the computational cost of ~~the~~ working with the dataset.

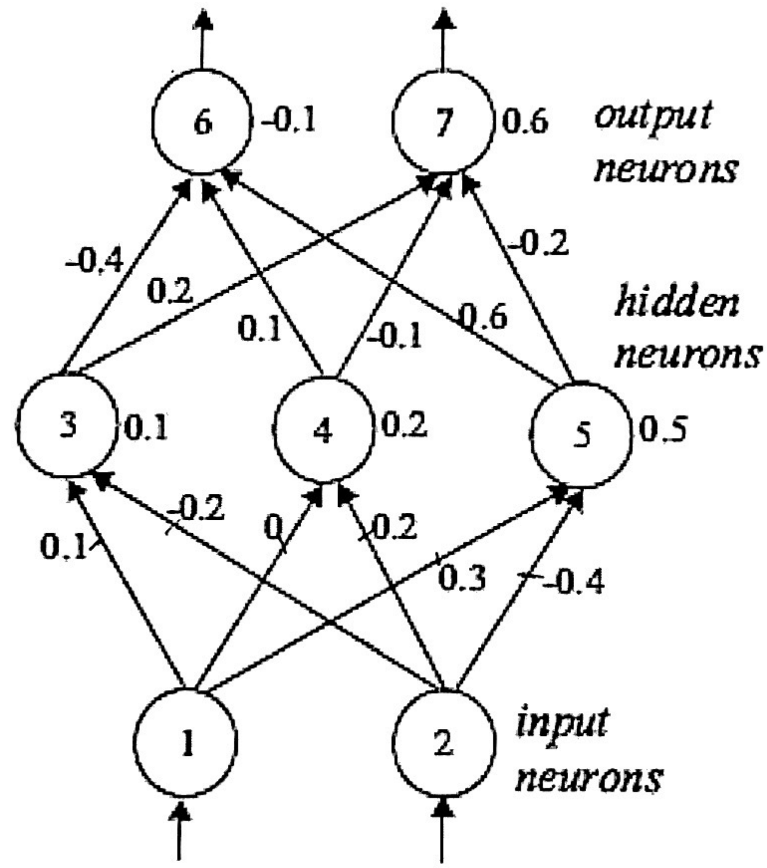
b) It does not do a good job in determining which objects belong to which class

c) Trying to classify faces to a training set of faces by computing the covariance matrix of each subjects training set photos, computing the eigenvectors, and sorting the principal components. Find the distance for each testing face and that's which face it classifies it as.

by computing the change of basis matrix PX where P is the principle components, and X is the testing data.

Problem 2:

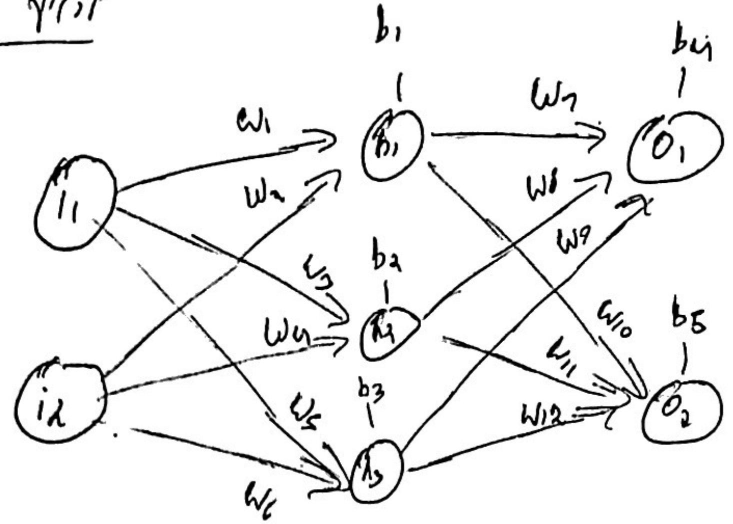
Given the following neural network with initialized weights as in the picture, explain the network architecture knowing that we are trying to distinguish between nails and screws and an example of training tuples is as follows: T1(0.6, 0.1, nail), T2(0.2, 0.3, screw).



Let the learning rate η be 0.1 and the weights be as indicated in the figure above. Do the forward propagation of the signals in the network using T1 as input, then perform the back propagation of the error. Show the changes of the weights.

Here we use a 2-3-2 ~~ANN to~~ multilayer perceptron network to classify the data tuples as nails or screws. An output of $\begin{bmatrix}1\\0\end{bmatrix}$ is a nail and an output of $\begin{bmatrix}0\\1\end{bmatrix}$ is a screw.

Forward Prop



$i_1 = .6 \quad i_2 = .1 \quad target = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$W_1 = .1 \quad W_2 = -.2 \quad W_3 = 0 \quad W_4 = .2$

$W_5 = .3 \quad W_6 = -.4$

$W_7 = -.4 \quad W_8 = .1 \quad W_9 = .6$

$W_{10} = .2 \quad W_{11} = -.1 \quad W_{12} = -.2$

$b_1 = .1 \quad b_2 = .2 \quad b_3 = .5 \quad b_4 = -.1 \quad b_5 = .6$

$Out\,h_1 = \sigma(i_1 \times w_1 + i_2 \times w_2 + b_1) = \sigma(.6 \times .1 + .1 \times -.2 + .1) = .5349$

$net\,h_1 = .14$

$out\,h_2 = \sigma(i_1 \times w_3 + i_2 w_4 + b_2) = \sigma(.6 \times 0 + .1 \times .2 + .2) = .5548$

$net\,h_2 = .22$

$Out\,h_3 = \sigma(i_1 \times w_5 + i_2 w_6 + b_3) = \sigma(.6 \times .3 + .1 \times -.4 + .5) = .6548$

$net\,h_3 = .64$

$Out\,O_1 = \sigma(h_1 \times w_7 + h_2 w_8 + h_3 w_9 + b_4) = \sigma(.5349 \times -.4 + .5548 \times .1 + .6548 \times .6 + b$

$= .5335 \qquad net\,O_1 = .1344$

$Out\,O_2 = \sigma(h_1 \times W_{10} + h_2 \times W_{11} + h_3 W_{12}) = \sigma(.5349 \times .2 + .5548 \times -.1 + .6548 \times -.2 + b$

$= .6273 \qquad net\,O_1 = .5205$

$Error\,O_1 = \frac{1}{2}(1 - .5335)^2 = .1088 \qquad Error\,O_2 = \frac{1}{2}(0 - .6273)^2 = .1968$

$E_{total} = Error\,O_1 + Error\,O_2 = .3056$

Back Prop – output layer

$$w_7 \quad \frac{\partial E_{total}}{\partial w_7} = \underbrace{\frac{\partial E_{total}}{\partial out_{O_1}}}_{①} \times \underbrace{\frac{\partial out_{O_1}}{\partial net_{O_1}}}_{②} \times \underbrace{\frac{\partial net_{O_1}}{\partial w_5}}_{③} = \nabla E(w_7)$$

① $E_{total} = \frac{1}{2}(1 - out_{O_1})^2 + \frac{1}{2}(0 - out_{O_2})^2 \Rightarrow \frac{\partial E_{total}}{\partial out_{O_1}} = 1(1 - out_{O_1})^{2-1} \cdot -1 + 0 = \boxed{out_{O_1} - 1} = \boxed{-.4665}$

② $out_{O_1} = \frac{1}{e^{-net_{O_1}} + 1} \Rightarrow \frac{\partial out_{O_1}}{\partial net_{O_1}} = \sigma'(out_{O_1}) = \sigma(out_{O_1})(1 - out_{O_1}) = \boxed{.2489}$

③ $net_{O_1} = out_{h_1} w_7 + out_{h_2} w_8 + b_4 \Rightarrow \frac{\partial net_{O_1}}{\partial w_7} = out_{h_1} = \boxed{.5349}$

therefore $\frac{\partial E_{total}}{\partial w_7} = -.4665 \times .2489 \times .6349 = \boxed{-.0621}$

$w_7\ new = w_7 - .1 \times (-.0621) = -.3938$

$\nabla E(w_8) = -.4665 \times .2489 \times .5548 = -.0644$

$w_8\ new = w_8 - .1 \times (-.0644) = .1064$

$\nabla E(w_9) = -.4665 \times .2489 \times .6548 = -.0760$

$w_9\ new = w_9 - .1 \times (-.0760) = .6076$

$\nabla E(w_{10}) = out_{O_2} - 0 \times \cancel{\sigma(out_{O_2})} \; out_{O_2}(1 - out_{O_2}) \times out_{h_1} =$
$.6273 \times .2338 \times .5349 = .0784$

$w_{10}new = w_{10} - .1 \times .0784 = .1922$

$\nabla E(w_{11}) = .6273 \times .2338 \times .5546 = .0814$

$w_{11}\ new = -.1 - (.1 \times .0814) = -.1081$

$\nabla E(w_{12}) = .6273 \times .2338 \times .6548 = .0960$

$w_{12}\ new = -.2 - (.1 \times .0960) = -.2096$

Back Prop – hidden layer

w_1

$$\frac{\partial E_{total}}{\partial w_2} = \overset{①}{\frac{\partial E_{total}}{\partial out_{h1}}} \times \overset{②}{\frac{\partial out_{h1}}{\partial net_{h1}}} \times \overset{③}{\frac{\partial net_{h1}}{\partial w_1}} = \nabla E(w_1)$$

$$\frac{\partial E_{O1}}{\partial net_{O1}} = \frac{\partial E_{O1}}{\partial out_{O1}} \times \frac{\partial out_{O1}}{\partial net_{O1}}$$

$$= (out_{O1} - 1) \times (out_{O1}) \times (1 - out_{O1})$$

$$= -.4665 \times .2489 = -.1161$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{O1}}{\partial out_{h1}} + \frac{\partial E_{O2}}{\partial out_{h1}}$$

$$\frac{\partial E_{O1}}{\partial out_{h1}} = \frac{\partial E_{O1}}{\partial net_{O1}} \times \frac{\partial net_{O1}}{\partial out_{h1}} = .0464$$

$net_{O1} = w_7 h_1 + \dots$

$= w_7 = -.4$

$$\frac{\partial E_{O2}}{\partial out_{h1}} = \frac{\partial E_{O2}}{\partial net_{O2}} \times \frac{\partial net_{O2}}{\partial out_{h1}}$$

$net_{O2} = h_1 w_{10} + \dots$

$= w_{10} = .2$

$$\frac{\partial E_{O2}}{\partial net_{O2}} = \frac{\partial E_{O2}}{\partial out_{O2}} \times \frac{\partial out_{O2}}{\partial net_{O2}} = (out_{O2} - 0) \times (out_{O2}) \times (1 - out_{O2}) = .1467$$

$= .1467 \times .2 = .0293$ therefore, $\frac{\partial E_{total}}{\partial out_{h1}} = .0464 + .0293 = .0757$ ①

$$\frac{\partial out_{h1}}{\partial net_{h1}} = \sigma'(net_{h1}) = out_{h1} \cdot (1 - out_{h1}) = ~~.2989~~\ .2488 \quad ②$$

$$\frac{\partial net_{h1}}{\partial w_1} = \text{if } net_{h1} = w_1 i_1 + w_2 i_2 + b_1 \text{ then } \frac{\partial net_{h1}}{\partial w_1} = i_1 = .6 \quad ③$$

$$\nabla E(w_1) = .0757 \times .2488 \times .6 = .0113$$

$$w_1 \text{ new} = w_1 - (.1 \times .0113) = .0989$$

$$\nabla E(w_2) = .0757 \times .2488 \times .1 = .0019$$

w_2 new $=$ ~~.0757 × .2488 ×~~ $w_2 - (.1 \times .0019) = -.2002$

Back Prop Hidden layer (cont.).

w_3

$$\frac{\partial E_{total}}{\partial w_3} = \left(\overset{①}{\frac{\partial E O_1}{\partial out_{h2}} + \frac{\partial E O_2}{\partial out_{h2}}}\right)\left(\overset{②}{\frac{\partial out_{h2}}{\partial net_{h2}}}\right)\left(\overset{③}{\frac{\partial net_{h2}}{\partial w_3}}\right)$$

$\cancel{out_{O2} \times out_{O2}(1-out_{O1})}$

$out_{O2} - 0 \times out_{O2}(1 - out_{O2})$

w_{11}

$$\frac{\partial E O_1}{\partial out_{h2}} = \frac{\partial E O_1}{\partial net_{O1}} \times \frac{\partial net_{O1}}{\partial out_{h2}} = -.1161 \times .1 = -.0116$$

w_8

$$\frac{\partial E O_2}{\partial out_{h2}} = \frac{\partial E O_2}{\partial net_{O2}} \times \frac{\partial net_{O2}}{\partial out_{h2}} = .1467 \times -.1 = -.0147$$

① $= -.0263$ ② $= out_{h2}(1 - out_{h2}) = .2470$

③ $= i_1$

$\nabla E(w_3) = -.0263 \times .2470 \times .6 = -.0039$

$w_3\ new = w_3 - (.1 \times -.0039) = .00039$

$\nabla E(w_4) = -.0263 \times .2470 \times .1 = -.00064961$

$w_4\ new = w_4 - (.1 \times -.00064961) = .2001$

w_5

① $= -.1161 \times w_9$ ② $= out_{h3}(1 - out_{h3})$ ③ $= i_1$

$\nabla E(w_5) = -.0697 \times .2260 \times .6 = -.0095$

$w_5 new = w_5 - (.1 \times -.0095) = .3009$

$\nabla E(w_6) = -.0697 \times .2260 \times .1 = -.0016$

$w_6\ new = -.4 - (.1 \times -.0016) = -.4002$

Answer ↓

$w_1\ new = .0989$
$w_2\ new = -.2002$
$w_3\ new = .00039$
$w_4\ new = .2001$
$w_5\ new = .3009$
$w_6\ new = -.4002$
$w_7\ new = -.3938$
$w_8\ new = .1064$
$w_9\ new = .6076$
$w_{10}\ new = .1922$
$w_{11}\ new = -.1081$
$w_{12}\ new = -.2026$

Done!

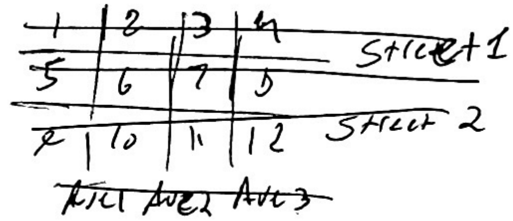
Problem 3:

(a) Choose a classification problem from an area of interest to you and describe the steps which are necessary to solve this specific problem based on a K-means classifier.

(b) Write a pseudo-code for the k-means clustering algorithm in the computer language of your choice!

(c) The error back-propagation training algorithm has been extensively used in multilayer feedforward networks. What is the deriving criterion for this algorithm? Write down (and explain) possible methods that can help to speedup or correct the training procedure.

~~a) You're trying to classify neighborhoods as Red or Blue based on their voting records. Step 1 is to turn the map of neighborhoods into an x, y plane~~ like so.



~~The neighborhoods are then weighted by the proportion of~~

~~a) You are trying to classify galaxies as types {ellipticals, spirals, irregulars} and by taking measurements of mass, apparent magnitude~~

See Next pages.

3a) You are trying to classify ~~objects~~ ~~fans~~ People as Taylor Swift fans and rolling stones fans based on the attributes age and gender.

The first step would be to turn gender into a quantitative measure by turning male = 1 and female = 0. Now we can use KNN to classify the fans by using the euclidean distance measure.

3b) Knn

% classify $(\vec{X}, \vec{y}, x)$ X = training data y = class labels
x = unknown sample

for i = 1 to m do

compute distance $d(X_i, x)$

end for

Compute set I containing indecies for the K smallest distances $d(X_i, x)$

return majority label for Y_i.

3C) The goal of back-prop is to adjust the weights and biases of the neuron by computing the negative gradient of the error function, and ~~ste~~ moving the weights in that direction to find a local minima of the error function.

3c cont.

Speeding up the learning process: Stochastic gradient descent. It takes computers an extremely long time to add up the individual changes to the weights in back-prop for every training example. SGD has you shuffling your training examples and putting them into smaller, "mini-batches" then you compute a step according to the mini batch.

every gradient descent step

If you were to plot the trajectory of your network's weights onto the relevant error function surface, it would be more like a drunk man stumbling down a n-d hill than a careful precise man carefully choosing each step.

Correcting learning process: it is possible for the back-prop to train to precisely to the training examples, therefore not generalizing well to the test examples. You can use a learning parameter $\eta \in [0,1]$ to only take a fraction of the -gradient for each weight to update it, thereby preventing overfitting.

Problem 4:

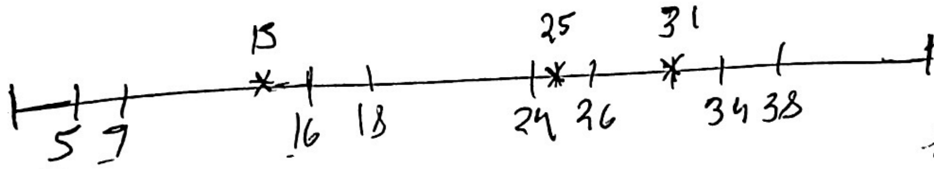
Consider the following set of one-dimensional points: $\{5; 7; 16; 18; 24; 26; 34; 38\}$.

(a) Suppose we apply k-means clustering to obtain three clusters, A, B, and C. If the three initial centroids are located at (15, 25, 31), respectively, show the clustering results after assigning each point to their closest centroid.

(b) Based on your answer in part (a), recompute the new locations of the centroids for A, B, and C. Compute also their overall SSE.

(c) What are the locations of the cluster centroids when the algorithm converges? Compute also their overall SSE.

a)



$X_1 \to C_1$
$X_2 \to C_1$
$X_3 \to C_1$
$X_4 \to C_1$
$X_5 \to C_2$
$X_6 \to C_2$
$X_7 \to C_3$
$X_8 \to C_3$

$d(X_1, C_1) = 10$ $\quad d(X_1, C_2) = 20$ $\quad d(X_1, C_3) = 26$

$d(X_2, C_1) = 8$ $\quad d(X_2, C_2) = 18$ $\quad d(X_2, C_3) = 24$

$d(X_3, C_1) = 1$ $\quad d(X_3, C_2) = 9$ $\quad d(X_3, C_3) = 15$

$d(X_4, C_1) = 3$ $\quad d(X_4, C_2) = 7$ $\quad d(X_4, C_3) = 13$

$d(X_5, C_1) = 9$ $\quad d(X_5, C_2) = 1$ $\quad d(X_5, C_3) = 7$

$d(X_6, C_1) = 11$ $\quad d(X_6, C_2) = 1$ $\quad d(X_6, C_3) = 5$

$d(X_7, C_1) = 19$ $\quad d(X_7, C_2) = 9$ $\quad d(X_7, C_3) = 3$

$d(X_8, C_1) = 23$ $\quad d(X_8, C_2) = 13$ $\quad d(X_8, C_3) = 7$

$C_1 = \{X_1, X_2, X_3, X_4\}$ $\quad C_2 = \{X_5, X_6\}$ $\quad C_3 = \{X_7, X_8\}$

4b)

$C_{1new} = \frac{5+7+16+18}{4} = 11.5$ $C_{2new} = \frac{24+26}{2} = 25$ $C_{3new} = \frac{34+38}{2} = 36$

4c)

$d(X_1, C_1, C_2, C_3) = 6.5 \quad 20 \quad 31$

$d(X_2, C_1, C_2, C_3) = 4.5 \quad 18 \quad 29$

$d(X_3, C_1, C_2, C_3) = 4.5 \quad 9 \quad 20$

$d(X_4, C_1, C_2, C_3) = 6.5 \quad 7 \quad 18$

$d(X_5, C_1, C_2, C_3) = 12.5 \quad 1 \quad 12$

$d(X_6, C_1, C_2, C_3) = 14.5 \quad 1 \quad 10$

$d(X_7, C_1, C_2, C_3) = 22.5 \quad 9 \quad 2$

$d(X_8, C_1, C_2, C_3) = 26.5 \quad 13 \quad 2$

$C_1 = \{X_1, X_2, X_3\}$

$C_2 = \{X_4, X_5, X_6\}$

$C_3 = \{X_7, X_8\}$

$C_{1new} = \frac{5+7+16}{3} = 9.3333$ $C_2 = \frac{18+24+26}{3} = 22.666$

$C_3 = 36$

$[4.33, 17.666, 31] = d(X_1, \vec{C})$

$[2.33, 15.66, 29] = d(X_2, \vec{C})$

$[6.66, 6.66, 20] = d(X_3, \vec{C})$

$[8.66 \quad 4.66 \quad 18] = d(X_4, \vec{C})$

$[14.66 \quad 1.33 \quad 12] = d(X_5, \vec{C})$

$[16.66 \quad 3.33 \quad 10] = d(X_6, \vec{C})$

$[24.66 \quad 11.33 \quad 2] = d(X_7, \vec{C})$

$[28.66 \quad 15.33 \quad 2] = d(X_8, \vec{C})$

Clusters stay the same therefore these are the final centroids

$SSE = (5-9.33)^2 + (7-9.33)^2 + (16-9.33)^2 + (18-22.66)^2 + (24-22.66)^2 + (26-22.66)^2 + (34-36)^2 + (38-36)^2 =$

$\boxed{= 111.333\ldots}$

Problem 5:

A known activation function is the logistic function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

whose limiting values are 0 and +1.

(a) Show that the derivative of $f(x)$ with respect to x is given by:

$$\frac{df}{dx} = f(x)(1-f(x)) \tag{2}$$

(b) What is the value of the derivative at the origin?

(c) Does the logistic function qualify as a cumulative distribution function? Justify your answer!

a) $f(x) = \frac{1}{1+e^{-x}}$ $\qquad \frac{(1+e^{x})(0) - (1)(0 - e^{-x})}{(1+e^{x})^2} = f'(x)$

$$\frac{+e^{-x}}{(1+e^{x})^2} \stackrel{?}{=} \left(\frac{1}{1+e^{-x}}\right)\left(1 - \frac{1}{1+e^{-x}}\right)$$

$$\frac{+e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2}$$

$$\frac{+e^{-x}}{(1+e^{x})^2} = \frac{\cancel{1}+e^{-x}\cancel{-1}}{(1+e^{x})^2}$$

$$\boxed{\frac{e^{-x}}{(1+e^{x})^2} = \frac{e^{-x}}{(1+e^{-x})^2} \text{ true!}}$$

b) $\frac{e^0}{(1+e^0)^2} = \frac{1}{4}$

c) yes, limiting values are 0 and 1 and limit as $x \to \infty = 1$ and limit as $x \to -\infty = 0$. the function is differentiable ~~for all~~ over all real numbers.

Problem 6:

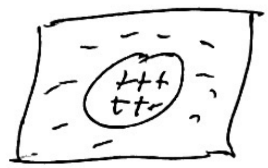
(a) Describe the types of decision regions that can be specified using the following NN architectures: single layer NN, FFNN with one hidden node and FFNN with two hidden nodes.

(b) Describe the FFNN neuron model and explain why the step function cannot be used as activation function in the backprop learning algorithm. (FFNN = feedforward neural network)

a) Single layer NN: linear decision regions

FFNN: 1 HN: Any continuous function on compact compact subsets of R^n.

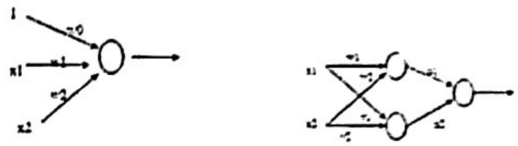
FFNN: 2 HN: Closed form decision boundaries like



b) Part of ~~Back~~ computing the gradient of the cost function is using the derivative of the activation function, however the step function has $\varphi'(\bullet>0)=0$ and $\varphi'(\bullet<0)=0$ and $\varphi'(\bullet=0)=\infty$. Where $\bullet$ is the weighted sum of the input neurons. This will cause the gradient to be zero for all ~~neuro~~ weights.

Problem 7:

(a) What are the limitations of single-layer neural networks?
(b) Derive the Delta rule for the following two networks:



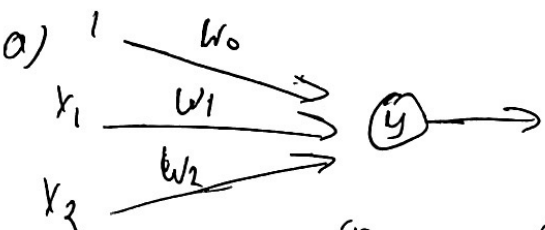
(a) (b)

Figure 1: (a) Network 1. (b) Network 2.

(c) Describe the types of decision regions that can be specified using the following NN architectures: single layer NN, FFNN with one hidden node and FFNN with two hidden nodes.

a) Single layer NN can only learn linear problems.
~~meaning only linearly seperable data can be classified.~~

b) a)



$$\frac{\partial E_{total}}{\partial w_n} = \overset{①}{\frac{\partial E_{total}}{\partial out\,y}} \times \overset{②}{\frac{\partial out\,y}{\partial net\,y}} \times \overset{③}{\frac{\partial net\,y}{\partial w_n}}$$

$$E_{total} = \frac{1}{2}(target - out\,y)^2 \Rightarrow \frac{\partial E_{total}}{\partial out\,y} = \frac{2}{2}(target - out\,y)^{2-1} \times -1 = (out\,y - target)$$

$$\frac{\partial out\,y}{\partial net\,y} = \frac{1}{1+e^{-net\,y}} = \sigma'(net\,y) = out\,y \times (1 - out\,y)$$

$$\frac{\partial net\,y}{\partial w_n} \Rightarrow net\,y = 1 \times w_0 \times x_1 w_1 \times x_2 w_2 = \begin{cases} 1 & \text{if } w_n = w_0 \\ x_1 & \text{if } w_n = w_1 \\ x_2 & \text{if } w_n = w_2 \end{cases}$$

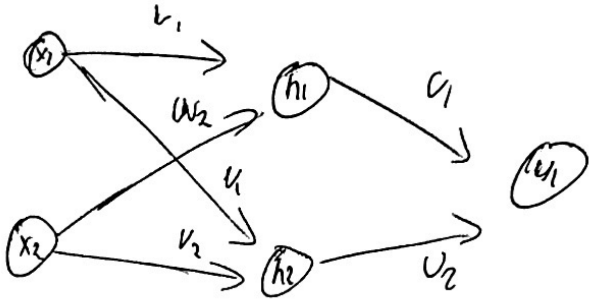
Delta Rule for w_n $= (out\,y - target)(out\,y) \times (1 - out\,y) \times x_i$

15

~~$x_i = 0$~~ $x_0 = 1$

7b Cont

See 6b for answer to 7c.



$$\frac{\partial E_{total}}{V_n} = \frac{\partial E_{total}}{\partial out_y} \times \frac{\partial out_y}{\partial net_y} \times \frac{\partial net_y}{\partial V_n}$$

① = (out y − target) ② = out y × (1 − out y)

③ = h_1 for V_1 h_2 for V_2 (same process as a)

$$\boxed{\text{Delta Rule } V_1, V_2 = (out_y - target) \times out_y \times (1 - out_y) \times h_1 \text{ or } h_2}$$

$$\frac{\partial E_{total}}{\partial W_n} = \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out_{h_1}}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial W_n}$$

$$\frac{\partial E_{total}}{\partial out_{h_1}} = \frac{\partial E_{total}}{\partial out_y} \times \frac{\partial out_y}{\partial net_y} \times \frac{\partial net_y}{\partial out_{h_1}}$$

$(out_y - target)\, out_y (1 - out_y) \times V_1$

$$\frac{\partial out_{h_1}}{\partial net_{h_1}} = out_{h_1} \times (1 - out_{h_2}) \times X_1 \text{ or } X_2$$

Delta Rule W_1, W_2 = $(out_y - target)(out_y)(1 - out_y) \times V_1 \times out_{h_1} \times (1 - out_{h_2}) \times X_1$ or X_2

X_1 for w_1
X_2 for w_2

Delta Rule V_1, V_2 = $(out_y - target)(out_y)(1 - out_y) \times V_2 \times out_{h_2} \times (1 - out_{h_2}) \times X_1$ or X_2

X_1 for V_1
X_2 for V_2,