CS 478 Homework

Perceptron Homework

- Assume a 3 input perceptron plus bias (it outputs 1 if net > 0, else 0)
- Assume a learning rate c of 1 and initial weights all 1: $\Delta w_i = c(t - z)\, x_i$
- Show weights after each pattern for just one epoch
- Training set 1 0 1 -> 0
  1 1 0 -> 0
  1 0 1 -> 1
  0 1 1 -> 1

Pattern	Target	Weight Vector	Net	Output	ΔW
		1 1 1 1			

SSE Homework

- Given the following data set, what is the L1 ($\Sigma|t_i - z_i|$), SSE/L2 ($\Sigma(t_i - z_i)^2$), MSE, and RMSE error for the entire data set? Fill in cells that have an x.

x	y	Output1	Target1	Output2	Target 2	Data Set
-1	-1	0	1	.6	1.0	
-1	1	1	1	-.3	0	
1	-1	1	0	1.2	.5	
1	1	0	0	0	-.2	
L1		x		x		x
SSE		x		x		x
MSE		x		x		x
RMSE		x		x		x

Quadric Machine Homework

- Assume a 2 input perceptron expanded to be a quadric perceptron (it outputs 1 if net > 0, else 0). Note that with binary inputs of -1, 1, that x^2 and y^2 would always be 1 and thus do not add info and are not needed (they would just act like two more bias weights)
- Assume a learning rate *c* of .4 and initial weights all 0: $\Delta w_i = c(t - z)\, x_i$
- Show weights after each pattern for one epoch with the following non-linearly separable training set (XOR).
- Has it learned to solve the problem after just one epoch?
- Which of the quadric features are actually needed to solve this training set?

x	*y*	*Target*
-1	-1	0
-1	1	1
1	-1	1
1	1	0

Linear Regression Homework

- Assume we start with all weights as 0 (don't forget the bias)
- What are the new weights after one iteration through the following training set using the delta rule with a learning rate of .2

x	*y*	*Target*
.3	.8	.7
-.3	1.6	-.1
.9	0	1.3

Logistic Regression Homework

- You don't actually have to come up with the weights for this one, though you could quickly by using the closed form linear regression approach
- Sketch each step you would need to learn the weights for the following data set using logistic regression
- Sketch how you would generalize the probability of a heart attack given a new input heart rate of 60

Heart Rate	Heart Attack
50	Y
50	N
50	N
50	N
70	N
70	Y
90	Y
90	Y
90	N
90	Y
90	Y

Backpropagation Homework

Network Equations

$$Output: O_j = f(net_j) = \frac{1}{1+e^{-net_j}}$$

$$f'(net_j) = \frac{\partial O_j}{\partial net_j} = O_j(1 - O_j)$$

$$\Delta w_{ij}\ (general\ node):\ C\ O_i\ \delta_j$$

$$\Delta w_{ij}\ (output\ node):$$

$$\delta_j = (t_j - O_j)\ f'(net_j)$$

$$\Delta w_{ij} = C\ O_i\ \delta_j = C\ O_i\ (t_j - O_j)\ f'(net_j)$$

$$\Delta w_{ij}\ (hidden\ node)$$

$$\delta_j = \sum_k (\delta_k \cdot w_{jk})\ \ f'(net_j)$$

$$\Delta w_{ij} = C\ O_i\ \delta_j = C\ O_i\ \left(\sum_k (\delta_k \cdot w_{jk})\right) f'(net_j)$$

BP-1) A 2-2-1 backpropagation model has initial weights as shown. Work through one cycle of learning for the following pattern(s). Assume 0 momentum and a learning constant of 1. Round calculations to 3 significant digits to the right of the decimal. Give values for all nodes and links for activation, output, error signal, weight delta, and final weights. Nodes 4, 5, 6, and 7 are just input nodes and do not have a sigmoidal output.

For each node calculate the following (show necessary equation for each). Hint: Calculate bottom-top-bottom.

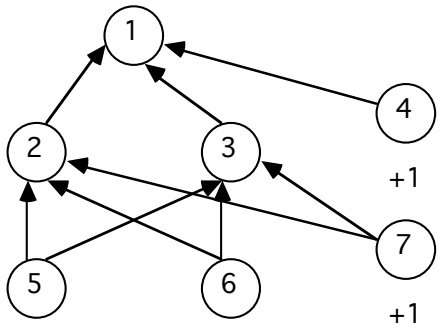
a =

o =

δ =

Δw =

w =



a) All weights initially 1.0

Training Patterns

1) 0 0 -> 1

2) 0 1 -> 0

BP-1)

$net2 = \Sigma w_i x_i = (1*0 + 1*0 + 1*1) = 1$

$net3 = 1$

$o2 = 1/(1+e^{-net}) = 1/(1+e^{-1}) = 1/(1+.368) = .731$

$o3 = .731$

$o4 = 1$

$net1 = (1*.731 + 1*.731 + 1) = 2.462$

$o1 = 1/(1+e^{-2.462}) = .921$

$\delta 1 = (t1 - o1)\, o1\, (1 - o1) = (1 - .921)\ .921\ (1 - .921) = .00575$

$\Delta w21 = \eta\, \delta_j\, o_i = \eta\, \delta 1\, o2 = 1 * .00575 * .731 = .00420$

$\Delta w31 = 1 * .00575 * .731 = .00420$

$\Delta w41 = 1 * .00575 * 1 = .00575$

$\delta 2 = o_j (1 - o_j) \Sigma \delta_k w_{jk} = o2\ (1 - o2)\ \delta 1\ w21 =$

$.731 \qquad (1 - .731)\ (.00575 * 1) = .00113$

$\delta 3 = .00113$

$\Delta w52 = \eta\, \delta_j\, o_i = \eta\, \delta 2\, o5 = 1 * .00113 * 0 = 0$

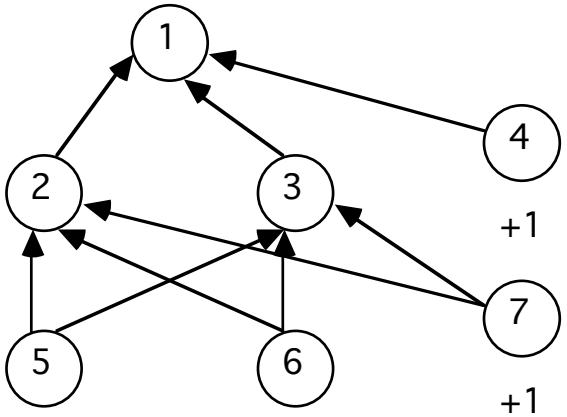
$\Delta w62 = 0$

$\Delta w72 = 1 * .00113 * 1 = .00113$

$\Delta w53 = 0$

$\Delta w63 = 0$

$\Delta w73 = 1 * .00113 * 1 = .00113$



PCA Homework

Terms		
m	5	Number of instances in data set
n	2	Number of input features
p	1	Final number of principal components chosen

Original Data		
	x	*y*
p1	.2	-.3
p2	-1.1	2
p3	1	-2.2
p4	.5	-1
p5	-.6	1
mean	0	-.1

- Use PCA on the given data set to get a transformed data set with just one feature (the first principal component (PC)). Show your work along the way.
- Show what % of the total information is contained in the 1st PC.
- Do not use a PCA package to do it. You need to go through the steps yourself, or program it yourself.
- You may use a spreadsheet, Matlab, etc. to do the arithmetic for you.
- You may use any web tool or Matlab to calculate the eigenvectors from the covariance matrix.

Meat N,Y	Crust D,S,T	Veg N,Y	Quality B,G,Gr
Y	Thin	N	Great
N	Deep	N	Bad
N	Stuffed	Y	Good
Y	Stuffed	Y	Great
Y	Deep	N	Good
Y	Deep	Y	Great
N	Thin	Y	Good
Y	Deep	N	Good
N	Thin	N	Bad

Decision Tree Homework

$$Info(S) = -\sum_{i=1}^{|C|} p_i \log_2(p_i)$$

$$Info_A(S) = \sum_{i=j}^{|A|} \frac{|S_j|}{|S|} Info(S_j) = -\sum_{j=1}^{|A|} \frac{|S_j|}{|S|} \cdot \sum_{i=1}^{|C|} p_i \log_2(p_i)$$

- Info(S) = - 2/9·$\log_2$2/9 - 4/9·$\log_2$4/9 -3/9·$\log_2$3/9 = 1.53
  - Not necessary unless you want to calculate information gain
- Starting with all instances, calculate gain for each attribute
- Let's do Meat:
- $Info_{Meat}(S)$ = 4/9·(-2/4$\log_2$2/4 - 2/4·$\log_2$2/4 - 0·$\log_2$0/4) +
  5/9·(-0/5·$\log_2$0/5 - 2/5·$\log_2$2/5 - 3/5·$\log_2$3/5) = .98
  - Information Gain is 1.53 - .98 = .55
- Finish this level, find best attribute and split, and then find the best attribute for at least the left most node at the next level
  - Assume sub-nodes are sorted alphabetically left to right by attribute

k-Nearest Neighbor Homework

- Assume the following training set.
- Assume a new point (.5, .2)
  - For nearest neighbor distance use Manhattan distance
  - What would the output be for 3-nn with no distance weighting? Show work and vote.
  - What would the output be for 3-nn with distance weighting? Show work and vote.

x	*y*	*Target*
.3	.8	A
-.3	1.6	B
.9	0	B
1	1	A

RBF Homework

- Assume you have an RBF with
  - Two inputs
  - Three output classes A, B, and C (linear units)
  - Three prototype nodes at (0,0), (.5,1) and (1,.5)
  - The radial basis function of the prototype nodes is
    - max(0, 1 – Manhattan distance between the prototype node and the instance)
  - Assume no bias and initial weights of .6 into output node A, -.4 into output node B, and 0 into output node C
  - Assume top layer training is the delta rule with LR = .1
- Assume we input the single instance .6 .8
  - Which class would be the winner?
  - What would the weights be updated to if it were a training instance of .6 .8 with target class B? (thus B has target 1 and A has target 0)

Naïve Bayes Homework

Size (B, S)	Color (R,G,B)	Output (P,N)
B	R	P
S	B	P
S	B	N
B	R	N
B	B	P
B	G	N
S	B	P

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$$

For the given training set:

1. Create a table of the statistics needed to do Naïve Bayes
2. What would be the output for a new instance which is Small and Blue?
3. What is the Naïve Bayes value and the normalized probability for each output class (P or N) for this case of Small and Blue?

HAC Homework

- For the data set below show all iterations (from 5 clusters until 1 cluster remaining) for HAC single link. Show work. Use Manhattan distance. In case of ties go with the cluster containing the least alphabetical instance. Show the dendrogram for the HAC case, including properly labeled distances on the vertical-axis of the dendrogram.

Pattern	*x*	*y*
a	.8	.7
b	-.1	.2
c	.9	.8
d	0	.2
e	.2	.1

k-means Homework

- For the data below, show the centroid values and which instances are closest to each centroid *after* centroid calculation for two iterations of *k*-means using Manhattan distance
- By 2 iterations I mean 2 centroid changes after the initial centroids
- Assume $k = 2$ and that the first two instances are the initial centroids

Pattern	*x*	*y*
a	.9	.8
b	.2	.2
c	.7	.6
d	-.1	-.6
e	.5	.5

Q-Learning Homework

- Assume the deterministic 4 state world below (each cell is a state) where the immediate reward is 0 for entering all states, except the rightmost state, for which the reward is 10, and which is an absorbing state. The only actions are move right and move left (only one of which is available from the border cells). Assuming a discount factor of .8, give the final optimal Q values for each action in each state and describe an optimal policy.