CS 270 Homework

Submit your homework through learning suite as a PDF (or JPEG) file. You may scan a copy of a handwritten solution, but it should be neat and legible. Specific problems are found in the homework file linked to at the top of the schedule. On occasion we will update the homework, so check the file before doing each 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 instance and just do one epoch
- Training set 1 0 1 -> 0
  1 .5 0 -> 0
  1 -.4 1 -> 1
  0 1 .5 -> 1

Data	Target	Weight Vector	Net	Output	ΔW
		1 1 1 1			

Error Values Homework

- Given the following data set, what is the L1, SSE (L2), MSE, and RMSE error?

Instance	x	y	Output	Target	Data Set
1	-1	-1	.6	1.0	
2	-1	1	-.3	0	
3	1	-1	1.2	.5	
4	1	1	0	-.2	
L1			?		?
SSE			?		?
MSE			?		?
RMSE			?		?

Quadric Machine Homework

- Assume a 2-input perceptron expanded to be a quadric (2nd order) perceptron, with 5 input weights (x, y, $x*y$, x^2, y^2) and the bias weight
  - Assume it outputs 1 if net > 0, else 0
- Assume a learning rate c of .5 and initial weights all 0
  - $\Delta w_i = c(t - z)\, x_i$
- Show all weights after each pattern for one epoch with the following training set

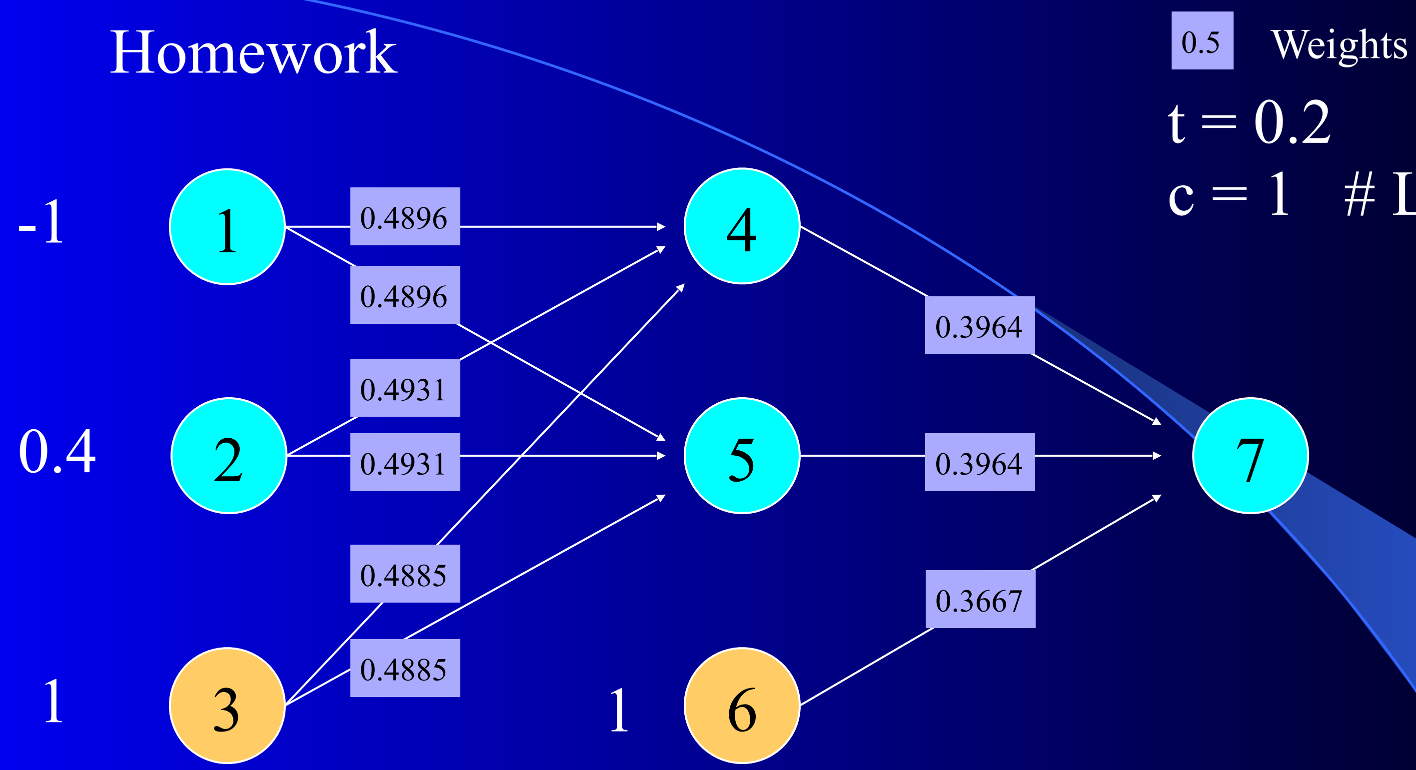
x	y	Target
0	.4	0
-.1	1.2	1
.5	.8	0

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
- How does it then generalize for the novel input (1, .5)?

x_1	x_2	*Target*
.3	.8	.7
-.3	1.6	-.1
.9	0	1.3

Homework



t = 0.2
c = 1 # Learning Rate

Evaluate the network using the new weights on the new input

Using those values, back propagate the error and calculate new weights for the network

Nodes 1 and 2 (input nodes) and 3 and 6 (bias inputs) are just placeholder nodes and **do not pass their values through a sigmoid**.

Backprop Homework

- For your homework, update the weights for a second instance -1 .4 ==> .2. Continue using the updated weights shown on the previous slide. Show your work like we did on the previous slide.

- Then go to the link below: Neural Network Playground and play around with the BP simulation. Try different training sets, layers, inputs, etc. and get a feel for what the nodes are doing. **You do not have to hand anything in for this part.**

- http://playground.tensorflow.org/

PCA Homework

Original Data				
ID	*x*	*y*	*x'*	*y'*
1	.2	-.3	-	-
2	-1.1	2	-	-
3	1	-2.2	-	-
4	.5	-1	-	-
5	-.6	1	-	-
mean			0	0

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

- Use PCA on the given data set to get a transformed data set with just one feature (the first principal component). Show your work along the way.
- Show what % of the total information is contained in the 1st principal component.
- 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, Python, etc. to do the arithmetic for you.
- You may use Python or any web tool to calculate the eigenvectors/eigenvalues for the covariance matrix.
- Optional: After, use any PCA solver (e.g. sklearn) and use it to solve the problem and check your answers.

Decision Tree Homework

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

$$\text{Info}_F(S) = \sum_{j=1}^{|F|} \frac{|S_j|}{|S|} \cdot \text{Info}(S_j)$$

$$\text{Gain}(\text{S}, F) = \text{Info}(S) - \text{Info}_F(S)$$

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

- Finish the first level, find the best attribute and split
- Then find the best attribute for the left most node at the second level and split the node accordingly
  - Assume sub-nodes are sorted alphabetically left to right by attribute
  - Label any leaf nodes with their majority class
  - You could/should continue with the other nodes to get more practice

k-Nearest Neighbor Homework

- Assume the following training set
- Assume a new point (.5, .2)
  - For all below, use Manhattan distance, if required, and show work
  - What would the output class for 3-nn be with no distance weighting?
  - What would the output class for 3-nn be with squared inverse distance weighting?
  - What would the 3-nn regression value be for the point if we used the regression values rather than class labels? Show results for *both* no distance weighting and squared inverse distance weighting.

x	*y*	*Class Label*	*Regression Label*
.3	.8	A	.6
-.3	1.6	B	-.3
.9	0	B	.8
1	1	A	1.2

Naïve Bayes Homework

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

For the given training set:

1. Create a table of the statistics needed to do Naïve Bayes
2. What would be the best output for a new instance which is Small and Blue? (e.g. the class which wins the argmax)
3. What is the true probability for each output class (P or N) for 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

Silhouette Homework

- Assume a clustering with {a,b} in cluster 1 and {c,d,e} in cluster 2. What would the Silhouette score be for a) each instance, b) each cluster, and c) the entire clustering. d) Sketch the Silhouette visualization for this clustering. Use Manhattan distance for your distance calculations.

Pattern	*x*	*y*
a	.8	.7
b	.9	.8
c	.6	.6
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 leftmost 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). Assume a discount factor of .8, and all initial Q-values of 0. Give the final optimal Q values for each action in each state and describe an optimal policy.

Reward: 10			

CNN Homework

- Assume a traditional CNN with an initial input image of 16x16, followed by a convolutional layer with 8 feature maps using 5x5 receptor fields, followed by a max pooling layer with 2x2 receptive fields, followed by a convolution layer with 10 feature maps using 3x3 receptor fields. Those outputs go straight into (no additional pooling layer) a fully connected MLP with 20 hidden nodes followed by 3 output nodes for 3 possible output classes. Assume no zero-padding and stride=1 for convolution layers, no overlap and no trainable weights for the one pooling layer, and convolutional maps connect to all maps in the previous layer. Sketch the network. For each layer state a) What is the size of the maps in the layer (e.g. the input layer is 16x16), b) how many unique trainable weights are there per layer, and c) total connections in the layer. Show your work and explain your numbers in each case (similar to the previous slide).