NetID _____ Name _____

ANSWERS

QUESTION 1.

a) 0.2

b) $\sqrt{6}$

c) D1 and D2: $\frac{1}{3}$

D3 and D2: $\frac{1}{3}$

D4 and D2: $\frac{3}{\sqrt{15}}$

Hence D4 is most similar to D2.

d) Jaccard: [0, 1]
Cosine: [0, 1] (for vectors with non-negative values)
Euclidean: [0, Infinity)

QUESTION 2.

a) $P(London|U) = \frac{2}{4} = \frac{1}{2}$, $P(London|S) = \frac{1}{4}$, $P(Paris|U) = \frac{1}{4}$, $P(Paris|S) = \frac{1}{4}$
or
$P(London|U) = 1$, $P(London|S) = \frac{1}{2}$ $P(Paris|U) = \frac{1}{2}$, $P(Paris|S) = \frac{1}{2}$

b) The system first computes class-specific scores:

$$Score(U) \propto P(London|U) * P(Paris|U) = \frac{1}{8} \; or \; \frac{1}{4}$$

$$Score(S) \propto P(London|S) * P(Paris|S) = \frac{1}{16} \; or \; \frac{1}{8}$$

The system then predicts the class with the highest score, here: U.

c) The Naive Bayes classification requires us to multiply probabilities, which are numbers between 0 and 1. If a document contains many words, this can lead to underflow. Converting probabilities to log probabilities avoids this problem.

QUESTION 3.

A. Recursive neural networks can be thought of as generalizations of recurrent NNs.
B. You'd use LSTMs or GRUs to try and fix the vanishing gradient problem.
C. You'd want to use a recursive neural network.
D. You'd want to use a recurrent neural network.

QUESTION 4.

a)

NetID Name

$$h_1 = ReLU(W_1 x + b_1) = ReLU\left(\begin{bmatrix} -2 \\ 9 \\ 3 \\ -1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 9 \\ 3 \\ 0 \end{bmatrix}$$

b)

$$p = softmax(ReLU(W_2 h_1 + b_2)) = softmax\left(ReLU\left(\begin{bmatrix} -3 \\ 3 \end{bmatrix}\right)\right) = softmax\left(\begin{bmatrix} 0 \\ 3 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 1/(1+e^3) \\ e^3/(1+e^3) \end{bmatrix} = \begin{bmatrix} 0.0474 \\ 0.9526 \end{bmatrix}$$

QUESTION 5.
1-5: ACACB
6-10: (BD)BCCA

QUESTION 6.

S1:
Viterbi algorithm: given a sequence of symbols/observations and a model, what is the most likely sequence of states that produced the sequence.
Forward algorithm: what is the probability that a particular sequence of symbols is produced by a particular model?
This was discussed in the class slides.
See also:
https://en.wikipedia.org/wiki/Viterbi_algorithm#Pseudocode
https://en.wikipedia.org/wiki/Forward_algorithm#Pseudocode

S2:

$$g'_{logistic}(z) = \frac{\partial}{\partial z}\left(\frac{1}{1+e^{-z}}\right)$$

$= \frac{e^{-z}}{(1+e^{-z})^2}$(chain rule) Note: (-1) * (-1) = 1 (the two cancel out)

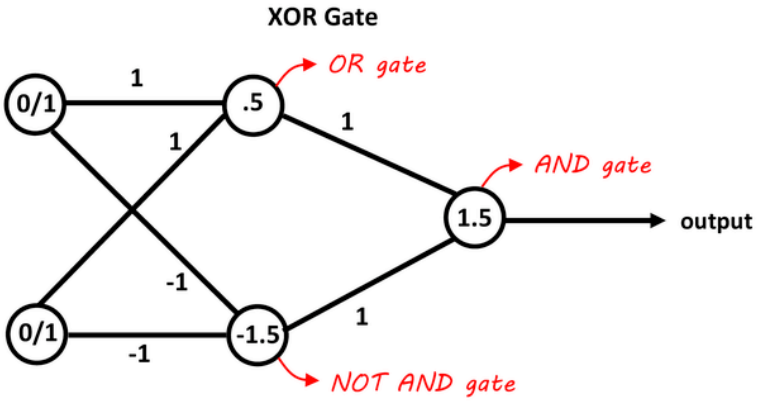
$$= \frac{1+e^{-z}-1}{(1+e^{-z})^2}$$

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

$$= g_{logistic}(z) - g_{logistic}(z)^2$$

$$= g_{logistic}(z)(1 - g_{logistic}(z))$$

S3: There are many possible answers. Here is one of them.



Reference: https://blog.abhranil.net/2015/03/03/training-neural-networks-with-genetic-algorithms/

S4: Distributional Similarity

S5: 2*1+3*1 = 5; 2*1 = 2

S6: some examples include: (1) coordination test "Janet and the kids from school went to the park", (2) pronoun substitution test "Janet went to the park", "She went to the park", (3) topicalization "Janet went to the park", "To the park went Janet", (4) the intuition test (Do you intuitively feel that this is a syntactic constituent". More in the class notes

S7: gate, deer

S8: choose the one with the probability $P(w|c) \cdot P(c)$ the largest, which is "dear"

S9: 6000/2=3000, 6000/3=2000, 6000/4=1500, 6000/5=1200

S10: For all x, if x is y's mother, then x is y's parent

Question 7

a. LCS (lowest common subsumer) the most specific concept which is an ancestor of both A and B. A mention of concepts, wordnet or an example is required for full credit.

b. confusion matrix for classification is a table which specifies predicted values vs actual values for binary classification and is useful for calculating statistics such as recall and precision; shows true/false positives and negatives
c. the principle of semantic compositionality – meaning of a whole is a function of the meanings of simpler parts and the way those parts are put together.
d. negative sampling (for word embeddings) a way of approximating the softmax by using a small number of randomly selected contexts for the denominator for efficiency in training word embeddings.
e. backpropagation over time for RNN – a method used in gradient descent for recurrent networks. It begins by unfolding the network over time and then applying backprogagation to find the gradient of the cost with respect to all parameters