Natural Language Processing
Midterm Exam
Oct. 6, 2022
Parker Smith

1. Which of the following are correct for the given regular expression [\w.-]+@[\w.-]+
   (a) al.ice@yahoo.com
   (b) bob.alice-com
   (c) 1.Abc@kennesaw.edu1
   (d) @ksu.edu
   (e) All of the above

2. Which of the following are correct for the given regular expression ^[^A-Z0-6]+
   (a) ^abc 10
   (b) abc9
   (c) 1 “Hello”
   (d) ^abc9
   (e) None of the above

3. Write down the differences of different activation functions, sigmoid, tanh, relu. Which one (activation function) will you use in your logistic regression classifier and why?

   Sigmoid: The Sigmoid activation function is $\frac{1}{1+e^{-z}}$. It will always output a decimal value from 0 to 1 inclusive. It is most often used to determine probabilities as the values will always be within a 0% to 100% probability chance.

   Tanh: The Tanh activation function is $\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$. It will always output a decimal value from -1 to 1 inclusive and is centered around 0. The derivative of the Tanh function is non-monotonic and is therefore sometimes impossible to use in back-propagation.

   Relu: The Relu activation function is linear unless the value is negative where it is immediately converted to 0. The function has no Vanishing Gradient problem and is extraordinarily fast compared to the speed of other activation functions.

   If I were to create a logistic regression classifier, I would prefer to use the sigmoid function because it will always return a percentage value that will explicitly tell me how probable the chosen class is to being correct.

4. What is sequence labeling? How would you build your parts of speech baseline model?

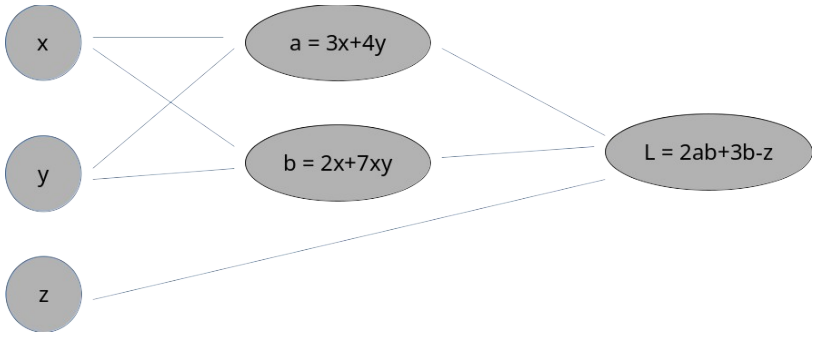
   Sequence labeling is taking an input sequence and applying a label to each token within the sequence to achieve an output sequence that is of the same length as the input sequence. If I were to build a parts of speech model, I would use a Hidden Markov Model as it neatly provides an output for every input token and can quickly be pruned to improve

efficiency. By deleting all vertices and edges which have a probability of 0, the model can be heavily compressed and computational times can be drastically reduced.

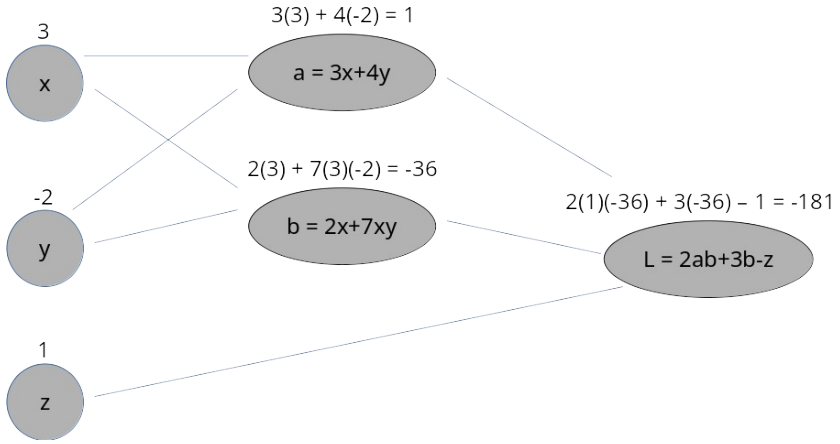
5. What are name entities? What is entity recognition task? Give an NER example. Why is Name Entity Recognition (NER) a difficult task? Please describe with example, how window based NER classification can be performed to detect Person Name.

   Name entities are any entities which are classified using proper nouns. Often, the first letter of each word of a name entity is capitalized. Abbreviations of proper nouns are also name entities. Entity recognition is a task where name entities within a sentence must be identified and classified.
   An example of this can be found in the question of “Give an NER example” itself. “NER” is a name entity representing the proper noun “Name Entity Recognition.” A task for an NER model would be to determine that “NER” from the list of tokens is a named entity.
   NER is a difficult task because it can often be hard to differentiate between multiple entities if they interlap, or even a single entity if a subset of the entity could also be considered a named entity.
   Window based NER classification can detect a Person Name by using neighboring words to determine the context of the name entity, therefore improving the chance of classification. An example of this is the sentence “Austin grabbed the toy.” Since neighboring words are also being classified, the NER classification model can easily understand that “Austin” is a person and not the city, as a city cannot perform actions such as grabbing. Therefore, the entity in the sentence must be a person.

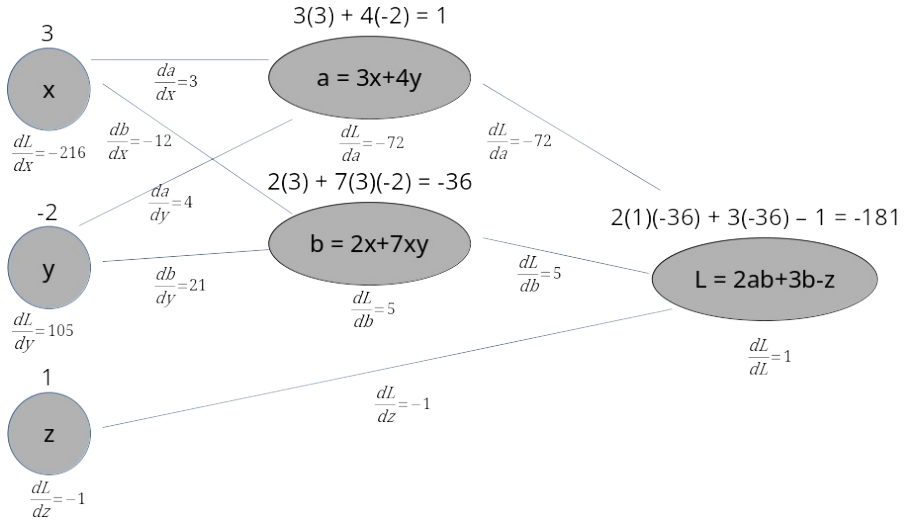
6. Given the following equations:
   a = 3x + 4y b = 2x + 7yx L = 2ab + 3b − z
   (a) Please draw computation graph (circuit diagram) for the given equation above.



   (b) Show forward pass values on the diagram, for the given values of x = 3, y = -2 and z = 1.



   (c) Show a complete back-propagation circuit diagram with corresponding gradient values.



7. Please build your character-gram (char-gram) language models for the given training set. Please assume that your experiment will only have the following characters – [a, b, c, d, f, h] that exists in the training set.

Training set:
a a b c h d b f
b c h h a d f f a h b
b b a a h h c c h d d f
a b f f h c c d f f h
h h f c f c a a c c d d d

Test set:
a b c d f
a b b c c d

Task:
(a) Build char-unigram language model

P(<s>) = 5/65, P(a) = 9/65, P(b) = 7/65, P(c) = 10/65, P(d) = 8/65, P(f) = 10/65, P(h) = 11/65, P(</s>) = 5/65

(b) Build char-bigram language model

P(<s>|<s>) = 0/5	P(a|<s>) = 2/5	P(b|<s>) = 2/5	P(c|<s>) = 0/5	P(d|<s>) = 0/5	P(f|<s>) = 0/5	P(h|<s>) = 1/5	P(</s>|<s>) = 0/5
P(<s>|a) = 0/9	P(a|a) = 3/9	P(b|a) = 2/9	P(c|a) = 1/9	P(d|a) = 1/9	P(f|a) = 0/9	P(h|a) = 2/9	P(</s>|a) = 0/9
P(<s>|b) = 0/7	P(a|b) = 1/7	P(b|b) = 1/7	P(c|b) = 2/7	P(d|b) = 0/7	P(f|b) = 2/7	P(h|b) = 0/7	P(</s>|b) = 1/7
P(<s>|c) = 0/10	P(a|c) = 1/10	P(b|c) = 0/10	P(c|c) = 3/10	P(d|c) = 2/10	P(f|c) = 1/10	P(h|c) = 3/10	P(</s>|c) = 0/10
P(<s>|d) = 0/8	P(a|d) = 0/8	P(b|d) = 1/8	P(c|d) = 0/8	P(d|d) = 2/8	P(f|d) = 3/8	P(h|d) = 0/8	P(</s>|d) = 1/8
P(<s>|f) = 0/10	P(a|f) = 1/10	P(b|f) = 0/10	P(c|f) = 2/10	P(d|f) = 0/10	P(f|f) = 3/10	P(h|f) = 2/10	P(</s>|f) = 2/10
P(<s>|h) = 0/11	P(a|h) = 1/11	P(b|h) = 1/11	P(c|h) = 2/11	P(d|h) = 2/11	P(f|h) = 1/11	P(h|h) = 3/11	P(</s>|h) = 1/11
P(<s>|</s>) = 4/5	P(a|</s>) = 0/5	P(b|</s>) = 0/5	P(c|</s>) = 0/5	P(d|</s>) = 0/5	P(f|</s>) = 0/5	P(h|</s>) = 0/5	P(</s>|</s>) = 0/5

(c) Compute joint probability for the given test set using char-unigram model.

P(<s>) = .0769, P(a) = .1385, P(b) = .1077, P(c) = .1538, P(d) = .1231, P(f) = .1538, P(</s>) = .0769
Test set: "<s> a b c d f </s> <s> a b b c c d </s>"
Joint probability: 7.1055E-15

(d) Compute perplexity of your models (char-unigram, char-bigram) and compare which model is better.

Perplexity algorithm = $\sqrt[N]{\frac{1}{P(w_1 w_2 ... w_n)}}$ where N = 15 and $P(w_1 w_2 ... w_n)$ is the joint probability for the test set for each model.

Char-unigram perplexity = 8.7746
Char-bigram perplexity = 3.4574

The char-bigram model is better because it has a lower complexity by a factor of 2.5.

8. Assume that we are in an alien world and their languages are different and only contain the following vocabulary [delta, gamma, alpha, beta, sigma, derivative, summation]. Their parts-of-speech tags are given as [A, B, C, D]. You are given a task to assign tags using Hidden Markov Model for the given test sentence.
Given the following sentences as training examples

Sentence1: delta gamma sigma summation
Tags: [A, B, C, A]
Sentence 2: alpha, sigma, beta derivative
Tags: [A, C, D, A]
Sentence 3: derivative gamma delta beta
Tags: [A, B, B, D]
Sentence 4: sigma summation beta alpha
Tags: [C, B, C, D]
Sentence 5: alpha beta sigma derivative
Tags: [A, B, C, A]

Test Sentence: alpha gamma beta sigma

(a) Please calculate transition probabilities

Transition Probability	A	B	C	D	<E>	Total
<S>	4/5	0	1/5	0	0	5
A	0	3/5	1/5	0	3/5	7
B	0	1/5	3/5	1/5	0	5
C	2/5	1/5	0	2/5	0	5
D	1/5	0	0	0	2/5	3

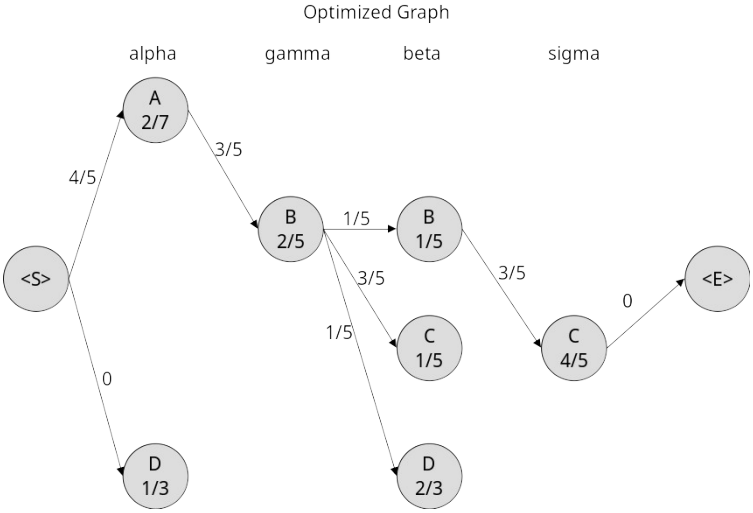
(b) Calculate emission probabilities

Emission Probability	A	B	C	D
delta	1/7	1/5	0	0
gamma	0	2/5	0	0
alpha	2/7	0	0	1/3
beta	0	1/5	1/5	2/3
sigma	0	0	4/5	0
derivative	3/7	0	0	0
summation	1/7	1/5	0	0
Total	7	5	5	3

(c) How many possible tag sequences (paths) can be generated for the given test sentence?

The possible tag sequences can be determined by the formula $SentenceLength^{NumberOfSpeechParts}$. Therefore, the possible tag sequences that can be generated for the test sentence is 4^4, or 256.

(d) Assign possible tags for the given test sentence following maximum likelihood calculation. Please show details likelihood calculation for all the optimized tag sequences.



To the right is the optimized graph for this model. All vertices with a probability of zero were removed, and all edges with a probability of zero (ignoring those from <S> and to <E>) were also removed. This leaves a single chain from <S> to <E>. The likelihood for this chain (ignoring the probability of 0 from sigma to <E>) is

$$\frac{4}{5}*\frac{2}{7}*\frac{3}{5}*\frac{2}{5}*\frac{1}{5}*\frac{1}{5}*\frac{3}{5}*\frac{4}{5}=0.0011$$

Therefore, the maximum possible tag sequence for this text is A → B → B → C

9. A restaurant chain wants to see whether customers like their foods or not. They hire you as an NLP scientist to do this task. You crawled their website, collect and annotate all the reviews. Your task is to conduct the experiment for this text data.

   (a) How many classes would you identify for this food review and why? Please list down the classes name as your preferences.

   For this food review I would make two classes: Like and Dislike. I specifically picked these two classes because the client specifically requested that they would like to know whether their customers Like or Dislike their food. They do not ask for a middle ground or a range, therefore just a binary classifier should be fine.

   (b) You train two classifiers – 1: logistic regression/soft-max classifier, and 2: a two-layer feed forward neural network. Please mention differences between these two classifiers.

   The logistic regression/soft-max classifier is simple. The features are placed in a regression formula and an activation function is used to determine which class the features belong to. Then, soft-max and back-propagation train the classifier until minimum loss is achieved.

   The two-layer feed-forward neural network is much more complex. The input layer holds all the features to start. Then, the features move on to the first layer after being multiplied by (at first) randomly determined weights. Next, the features move to the second layer where they are multiplied by more randomly determined weights. Finally, the output layer applies an activation function to each of the weighted features to output a percentage that determines which class the set of input features belongs to. From this output percentage, back-propagation trains the neural network until minimum loss is achieved.

   (c) Now you are deciding to pick only one classifier for the deployment environment. How would you design your experimental setup to evaluate these two classifiers? How would you decide to pick which classifier for the deployment environment? Please mention the details reasons behind your decision.

   I would primarily evaluate the two classifiers by running them on identical testing and training sets and comparing both the time it took each classifier to run and the resulting accuracy from each classifier. Typically, I would select the classifier that has the highest testing accuracy as the one to deploy, however if the classifier with the second highest accuracy was not significantly under-performing and ran much faster, then I would deploy the classifier with the second highest testing accuracy and fastest run time.