1. (2 points) Explain with your own words, using a short paragraph for each, what are:

Phonetics: Scientific study of the phonic elements of language and the processes of spoken communication.  
  
Phonology: Set of principles or rules determining the sound systems in a given natural language.

Morphology: Part of grammar that studies the problems related to the formation of words and the variations of their inflections.  
  
Syntax: A set of rules that are characteristic of a particular language.

Semantics: Study of the meaning of linguistic units and their combinations.   
  
Pragmatics: Which is susceptible of practical application, which has a practical value.

1. (1 point) What is the difference between stemming and lemmatization?

How do they both work?

Stemming and lemmatization are methods used in machine learning to analyze the meaning behind a word. Stemming uses the stem of the word, while lemmatization uses the context in which the word is being used. We'll later go into more detailed explanations and examples.

What are the pros and cons of both methods?

Stemming algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. This indiscriminate cutting can be successful in some occasions, but not always, and that is why we affirm that this approach presents some limitations.  
  
Lemmatization on the other hand, takes into consideration the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma.

1. (1 point) On logistic regression:

How does stochastic gradient descent work?

Stochastic gradient descent is an iterative method for optimizing an objective function. The purpose is to find the model parameters that correspond to the best fit between predicted and actual outputs. A gradient descent means descending a slope to reach the lowest point on a surface. Stochastic gradient descent chooses randomly one data point from the whole data set at each iteration to reduce the computations enormously.

What is the role of the learning rate?

The learging rate is the amount of change to effectively done to parameters of the model during each step of the SGD. It is the most important hyperparameter to tune the neural network in order to achieve good performance, having a very low learning rate can result in an excessively long duration of computation, whereas a to big learning rate will result in a model that does not converge and gives false predictions.

Will it always find the global minimum?

No, it will finds the global minimum only if the function is convex, which is not the case most of the time.  
However a local minimum is always found.

1. (1 point) What problems does TF-iDF try to solve?

TF\_iDF stands for Term Frequency–Inverse Document Frequency. It is a numerical statistic that reflect the importance of a word to a document in a collection or corpus.

What the is the TF part for?

The Term Frequency is related to the number of times a term occurs in a document. I t is a weight that corresponds to the frequency of a word in a text.

What is the iDF part for?

The Inverse Document Frequency measures the meaning of a term according to its distribution and use in all documents, not in a particualr document.   
  
We multiply both results for a word to have its total TF-iDF weight for a document. It allows to solve the problem of the weights of common words in a corpus.  
Because some words like "a" are common in a text, the term frequency will tend to incorrectly emphasize documents which happen to use those words more frequently, without giving enough weight to more meaningful terms. Therefore 'a' is not a good keyword to distinguish relevant and non-relevant documents and terms. Hence, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

1. (2 point) Summarize how the skip-gram method of Word2Vec works using a couple of paragraphs.

First of all, we need a set of data telling us which words are occurring close to a certain word. In order to do this, it uses a context window, all the word in the context window of a given word are considered as context words. We now have pair of words and context words.

Then we pass each pair to the neural network in order to train it and after this if we put any target word into the neural network it will give a vector output which reporesents the words which have a high probability to be near the given word.

How does it uses the fact that two words appearing in similar contexts are likely to have similar meanings?

If two different words have very similar contexts, then the model needs to output very similar results for these two words. And one way for the network to output similar context predictions for these two words is if the word vectors are similar. So, if two words have similar contexts, then the network is will learn similar word vectors for these two words.

1. (1 point) What are the differences between an RNN and an LSTM?

RNN stands for Recurrent Neural Network, it is a neural network algorithm that can remember previous inputs in memory and LSTM that stands Long Short-Term Memory for are a special type of RNN in fact they extend the memory of the RNN.

What problem is an LSTM trying to solve compared to a basic RNN?

LSTM tries to improve the method of back propagating the error.

1. (1 point) What would you expect if we use one of our classifiers trained on IMDB on Twitter data, and why?

In twitter messages we have a colloquial speech rather than a formal one in IMDB reviews. The length of messages changes as well, in IMDB we have long messages with long sentences and shorter ones in twitter. That’s why we think our models wouldn’t work so much on Twitter data