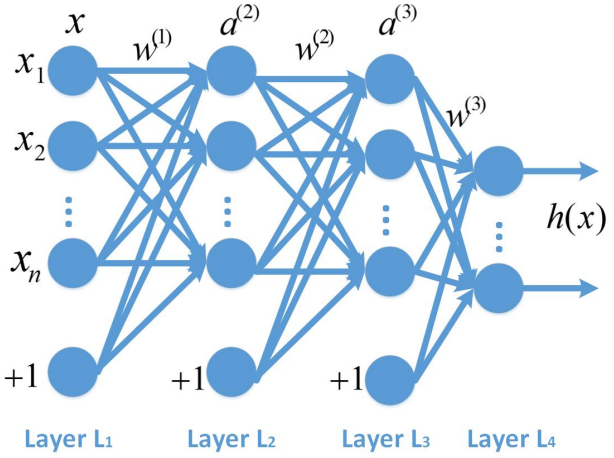
Neural Network:

Q)
In the following feed-forward neural networks, denote

$a_i^{(k)}$: the output of the ith neuron in the kth layer,
$w_{ij}^{(k)}$: the weight from the ith neuron in the kth layer to the jth neuron in the (k+1)th layer,
$b_j^{(k)}$: the weight from the bias neuron in the kth layer to the jth neuron in the (k+1)th layer,
f: the activation function on each neuron,
x,y : the training data and label (both are vectors).

a) (Forward pass) Compute the output (layer 4) $a^{(4)}$ using w, b and x.
b) (Backward pass) Suppose we use the square loss function as the cost function, which is $L(h(x), y) = \frac{1}{2}||h(x) - y||_2^2$. Compute $\frac{\partial L}{\partial w^{(2)}}$ and $\frac{\partial L}{\partial b^{(2)}}$.



A):
Denote .* as the element wise product between two equal-length vectors, i.e. z=x.*y iff $z_i = x_i \times y_i$, all i.

a). $a^{(4)} = f(w^{(3)}f(w^{(2)}f(w^{(1)}x + b^{(1)}) + b^{(2)}) + b^{(3)})$

b). $\frac{\partial L}{\partial z^{(4)}} = (a^{(4)} - y).* f'(w^{(3)}a^{(3)} + b^{(3)})$

$\frac{\partial L}{\partial z^{(3)}} = ((w^{(3)})^T \frac{\partial L}{\partial z^{(4)}}).* f'(w^{(2)}a^{(2)} + b^{(2)})$

$\frac{\partial L}{\partial w^{(2)}} = \frac{\partial L}{\partial z^{(3)}}(a^{(2)})^T, \frac{\partial L}{\partial b^{(2)}} = \frac{\partial L}{\partial z^{(3)}}.$

IMPORTANT Note: A similar question will show up in the exam. Please make sure you fully understand the algorithm and can derive the results by yourself. *In the lecture on May 11th our lecturer Fangqiu will teach again how to solve this question.*

Word Embedding:

Q1)

a) Please briefly describe the skig-gram model and write down its objective function.
b) An important reason for the success of the skip-gram model is that its training is very efficient and therefore it can be trained on huge text corpora. Negative sampling is an important technique to speed up the training of the skig-gram model, please briefly describe the idea.

A1)

a) The basic idea of the skig-gram model is, if two words w_1 and w_2 co-occr frequently in text corpora, the model should learn that when it sees one word w_1, the probability of seeing the word w_2 , $p(w_2|w_1)$ should be high. Its objective is to maximize the following function:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} \sum_{-c \le j \le c, j \neq 0} \log p(w_{t+j}|w_t)$$

Where T is the total length of the training text corpus. So it processes one local window of size c a time. For each window, it tries to maximize the sum of the probabilities of each context word given the central word.

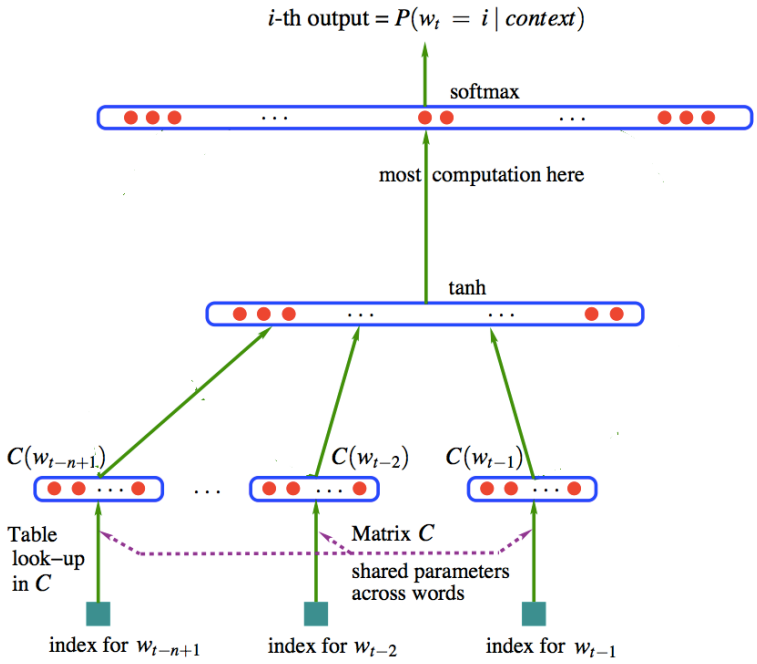
(Optional) Skip-gram associates two vectors, a "center" vector and a "context" vector, with each word. The center vector is used when the word is the center of a window, while the context word is used when the word is in the context of another word. To model the probability of seeing a context word given the central word ($p(w_{t+j}|w_t)$ in the objective function), skip-gram employs the softmax function:

$$p(w_{t+j} \mid w_t) = \frac{\exp(v'^{T}_{t+j} v_t)}{\sum_{k=1}^{|V|} \exp(v'^{T}_{k} v_t)}$$

where v_i and v_i' are the center vector and context vector of the word i, respectively, and $|V|$ is the vocabulary size.

b) The main computation cost of skip-gram is to compute the normalization term of the softmax function. Negative sampling is a technique to approximate the normalization term in an efficient way. To do so, it randomly samples a few negative words from an assumed negative word distribution and contrast just these a few words with the correct word, instead of comparing the correct word with all of the words in the vocabulary.

Q2)
a) Below is the architecture of the Neural Language Model. Pleas briefly describe it from bottom to up.
b) Why the Neural Language Model is slow? Specify the bottleneck and briefly explain why.



A2)
a) Each training sample is a local window of n words. The model takes the first n-1 words as input ("context"), and tries to predict the nth word.

Input layer: The context words, represented as one-hot vectors
Projection layer: Each context word is projected into a vector via the lookup table. The word vectors are then concatenated into a single vector, which serves as the input to the next layer.
Non-linear hidden layer: It's just a normal non-linear layer.
Output layer: The output layer first computes an un-normalized probability for each word in the vocabulary, then compute the normalized probabilities using the *softmax* function.

b) The bottleneck is the output layer. More specifically, it's the *softmax* function. To compute the normalized probabilities, we need to loop over every word in the vocabulary, whose size typically ranges from thousands to even millions.

Question Answering:

Q)
a) What is question answering (QA) and why it's important? Briefly describe.
b) A key component of QA systems is the knowledge sources, i.e., where to acquire the necessary knowledge for answering questions. From the early time of QA to recent days, the mainstream knowledge source of QA systems have undergone several significant changes. Please *describe these changes*, and analyze *why* these changes happen and *how* QA systems benefit from them.
c) (Open question) What do you think QA systems will be like in 10 or 20 years?

A)
a) Question answering (QA) aims to use machines to *automatically* find answer to *natural language questions*. It is important because humans are built-in with natural language communication capabilities. QA systems provide a natural communication interface from humans to machines, so that humans can effortlessly instruct machines to find information for them, which can greatly improve efficiency.

b)

1) In the early time of QA (1960 to ~2000), QA systems mainly seek knowledge from *tiny closed-domain knowledge bases* (KBs) and can only answer questions within a specific domain.
2) Beginning from ~2000, because of the birth and blossom of *the Internet*, which provides a huge amount of open-domain knowledge, QA systems start to use the Internet as their knowledge source. The most significant benefit is that now QA systems go *from closed domains to the open domain*, which means they get the capability of answering questions from any domain, as long as the related knowledge exists on the Internet. The coverage of QA systems is therefore significantly improved. However, since the knowledge on the web is expressed as *plain, unstructured text*, the precision of QA systems is not very good.
3) Beginning from ~2007, because of the emergence of the *Linked Data* as well as some *large-scale open-domain KBs*, QA systems start to seek knowledge from these *structured* knowledge sources. The most significant benefit, comparing to using the Internet, is that QA systems can potentially get *more accurate* than before, because the knowledge is now represented in a clear and meaningful way. Finding the exact answer becomes easier.

c) Open question.

NLP:

Q1)
Explain the difference between stemming and lemmatization. List one task which is more suitable for stemming and one task which is more suitable for lemmatization.

A1)
Stemming is to reduce word-forms to their stems, which can be linguistically invalid. Lemmatization tries to transform a word-form into a linguistically valid form (a lemma). Stemming is usually independent of POS tags, while lemmatization usually needs to consider them.

Web search is a task more suitable for stemming, because it makes it easier to match different word-forms with the same or similar meaning. For example, *adjustment* and *adjusting* will both be matched with *adjust*.

Machine translation is a task more suitable for lemmatization, since we expect linguistically valid words as output.

Q2)
Each NLP task below takes a sentence as input. State which method is better for this task: (1) a sequence tagging model, such as a Maximum Entropy Markov Model (MEMM), or (2) an ordinary classifier (nonsequential), such as a logistic regression classifier or SVM. Briefly justify your answer.
a) Verb phrase chunking
b) Determining whether the sentence is in Spanish or English.
c) Identifying person names
d) Identifying the hashtags in a tweet (e.g., #ilovenlp)

A2)
Different answers with reasonable explanations will be given (partial) credits.
a) Sequence tagging, because identifying verb phrase boundaries often depends on sequences of POS tags, e.g., passive voice and verb tenses.
b) Ordinary classifier, because an unordered set of words should be sufficient. The sequence of the words doesn't matter much.
c) Sequence tagging, because identifying person name boundaries often involves recognizing sequences of words based on person titles (e.g., Mr., Mrs.), job titles (e.g., CEO), and common first names.
d) Ordinary classifier, because identifying hashtags depends only on the intra-word characters (e.g., finding a hashtag in the first position of the string). Other words in the sentence do not matter.

Knowledge Base:

Q)
Answer each question below with a short (1-2 sentence) explanation.
(a) Suppose you want to build an NLP system to identify the names of sports teams. Would this task be more well-suited for named entity recognition (NER) techniques or semi-supervised bootstrapping (i.e. seed expansion)?
(b) Suppose you need to train a relation extraction system to identify relations for the company that you work for. Your boss gives you two choices: (1) The company is willing to purchase a database so you can train with distant supervision using the relation pairs in the database, or (2) The company will pay several people to manually label instances of the relations in a text corpus. Let's say that the database contains relation pairs that occur in N sentences (in total). If you choose option (2) then the company is willing to pay to obtain N manually labeled sentences. So the number of sentences containing relation instances from both choices will be the same. Which one will work better, or you think they will work equally well?
(c) Given the book *Pride and Prejudice*, design a system to extract a knowledge graph from it.

A)
Different answers with reasonable explanations will be given (partial) credits.
a) NER. Sports teams are often named after animals (e.g., Eagles, Bears, Hawks, Bees) and other things (e.g., Jazz, Utes), which would make bootstrapped learning difficult, both for acquiring unambiguous seeds and for the bootstrapping algorithm to avoid semantic drift. NER techniques, however, can utilize contextual features around each instance to determine which instances are in a sports context and which ones are not.
b) Manually annotated data. The premise of the question indicates that you will have exactly the same amount of data either way. Distant supervision almost always provides noisy labeled data. Manually annotated data should have almost no noise (only some inevitable human errors) since humans label each context by hand.
c) Hint: extract entities and relations which together form a large graph.

Paraphrasing:

Q) (Open question) Propose an algorithm to determine whether two paragraphs (not sentences) paraphrase each other.