# CS4851/6851 IDL Homework 6 April 18, 2022

Note: All coding problems to be submited with Github Link. Do not Upload the files/folder. Use git commands only.

Note: this is the distribution of questions:

1. Question 1 to Question 2: Required for everyone.
2. Question 3 and Question 4: Required by Graduate Students and Bonus for Undergrads
3. Question 5 to Question 6: Bonus question for both Graduate Students and Undergraduate Students

# Problem 1 (10 points)

We can represent the words in a vocabulary with binary vectors that have dimension of the number of words in the vocabulary and all values set to zero except the one value that corresponds to the index of the given word in the sorted version of this vocabulary. This is the so called one-in-K or one-hot encoding.

(a) Describe a representation of a document with a vector. (Think of a representation that is based on the one-hot encoding of the words in that document and has the same dimension as a single word (size of the vocabulary).)

We can a vector of dimension nx1 with n being the length of the dictionary do represent each word in the document. The representation will be a vector of ones and zeros. Say, for example, we have 3 words in our vocabulary: word1, word2, and word3.

Each of these word representations will be 3x1 vector. Word1 will be [1,0,0], word2 [0,1,0], and word3 will be [0,0,1].

(b) Explain why this representation is problematic:

We have a problem when repeating words within the document. Of course, we are about to represent unique words, but we are not able to represent the words as they appear in sentences. Suppose, for example, if our entire document is “The man in black said to the man in white this.” Our entire vocabulary will be of length 7… although our document has more words.

1. Simple sentence or two
2. Examples of the problem(s)
3. Provide at least two more options to fix this problem.

A solution is to create a nxm matrix where n is the length of the document and m the length of the dictionary. This will allow us to represent the document in a way that is meaningful to the document. Another solution could be to have a multidimensional matrix that would store the encoding and their positions in the document. TFIDF is also a solution that addresses this issue with repeating words.

# Problem 2 (30 points)

A recurrent network in Figure 3 takes a sequence of integers as an input and at the end of the sequence, on the last element produces a number between 0 and 1. What does a 0 mean? What does a 1 mean? Describe which function this network is computing (what is the meaning of this function). Assume all biases are 0, and make sure the hidden state is initialized to 0 as well. Note, the inputs, the weights, and the hidden state are just scalars in this RNN.

The figure below shows a recurrent neural network with a sigmoid activation function at the output layer. The hidden state of this RNN is just a linear layer. [0,1] are the probability of the outcome; that is the closer to zero the less likely it is, and the closer to 1; the more likely it is. This network basically computes the probability that the integer input is the output.

Figure 1: The RNN for Problem 2

Bonus for undergraduates beyond this line.

# Problem 3 (20 points)

In this problem, you will implement a recurrent neural network which implements binary addition. The inputs are given as binary sequences, starting with the least significant binary digit. (It is easier to start from the least significant bit, just like how you did addition in grade school.) The sequences will be padded with at least one zero on the end. For instance, the problem

100111+110010=1011001 (1)

would be represented as:

1. Input 1: 1, 1, 1, 0, 0, 1, 0
2. Input 2: 0, 1, 0, 0, 1, 1, 0
3. Correct output: 1, 0, 0, 1, 1, 0, 1

There are two input units corresponding to the two inputs, and one output unit. Therefore, the pattern of inputs and outputs for this example would be:

Design the weights and biases for an RNN which has two input units, three hidden units, and one output unit, which implements binary addition. All of the units use the hard threshold activation function. In particular, specify weight

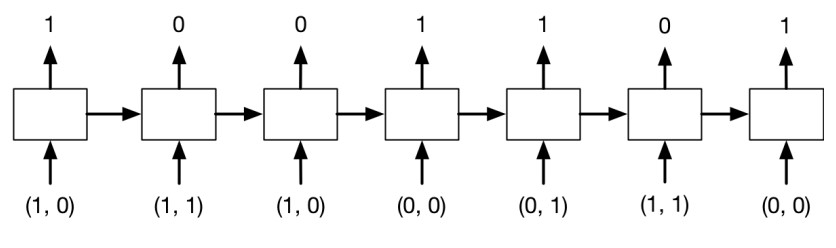


Figure 2: The RNN Binary for Problem 3

matrices U, V, and W, bias vector bh, and scalar bias by for the following architecture:

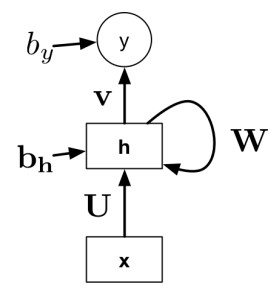


Figure 3: The RNN Architecture for Problem 3

Hint: In the grade school algorithm, you add up the values in each column, including the carry. Have one of your hidden units activate if the sum is at least 1, the second one if it is at least 2, and the third one if it is 3.

**U =**open square brackets table row 1 row 1 end table close square brackets**, W =**open square brackets table row 0 row 0 end table close square brackets**, V =** open square brackets table row 1 row 0 end table close square brackets**.** If the sum is 1, we multiply x by V to get the output if adding up the values in the column and the carry gives 3. To find the carry, we’ll need to multiply x by **W.** If the sum is 2, we multiply the input x by **W** to get the output and x by **U** to get the carry. Finally, if the sum is 1, we multiply the input vector **U** to get the output and to find its carry we multiply it by x by **W.** The bias vector bh is 0 in all cases. h is scalar quantity holding the carry from previous state.

# Problem 4 (20 points)

We have learned about regularization in image processing. How does regularization help in the context of Recurrent Neural Networks?

In recurrent neural networks, regularization can help prevent the vanishing and exploding gradients. Ref (Razvan Pascalu, Tomas Mikolov, Yoshua Bengio, http://proceedings.mlr.press/v28/pascanu13.pdf).

Bonus for both undergraduates and graduates beyond this line.

# Problem 5 (40 points)

How is teacher forcing ratio more accurate than the model output for a sequence of inputs? How can we use teacher process to parallelize the computation?

Teacher forcing feeds the ground to the next cycle after computing the loss when the model gets it wrong in the training phase. It is useful since the model needs a correct prior prediction to make a correct current prediction. Teacher forcing forces the model to always have a correct input even if the output is wrong in the training phase.

# Problem 6 (40 points)

Write a report on one of the following topics:

1. Attention Is All You Need {https://arxiv.org/pdf/1706.03762.pdf}
2. Transformers: {https://arxiv.org/pdf/1910.03771v5.pdf}

Transformers:

A tranformers is an open-source library designed to facilitate users to access large scale pretrained models, to build and experiment on top of them, and to deploy them in downstream tasks with state-of-the-art performance.

Suppose that we are interested in detecting doing some prediction given a medical text. We can use a transformer model that’s already been pretrained on medical corpora. The paper argues that using pretrained model increases the accuracy of the model for downstream tasks related to NLP.

Each transformer model is composed of a tokenizer which converts raw texts to sparse index encodings, a transformer which transforms sparse indices to contextual embeddings, and a head which uses contextual embeddings to make context specific predictions. There exists a variety of each components to choose from in the *transformers* library.

There exists Model Hub which makes it easy to access a transformer model and use it on their data. Users can also upload their own model to the hub for people to use.