Intro Deep Learning: Homework 6

Daniel Novikov

May 2nd, 2022

Problem 1

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).)
(b) Explain why this representation is problematic:
(i) Simple sentence or two (ii) Examples of the problem(s)
(c) Provide atleast two more options to fix this problem.

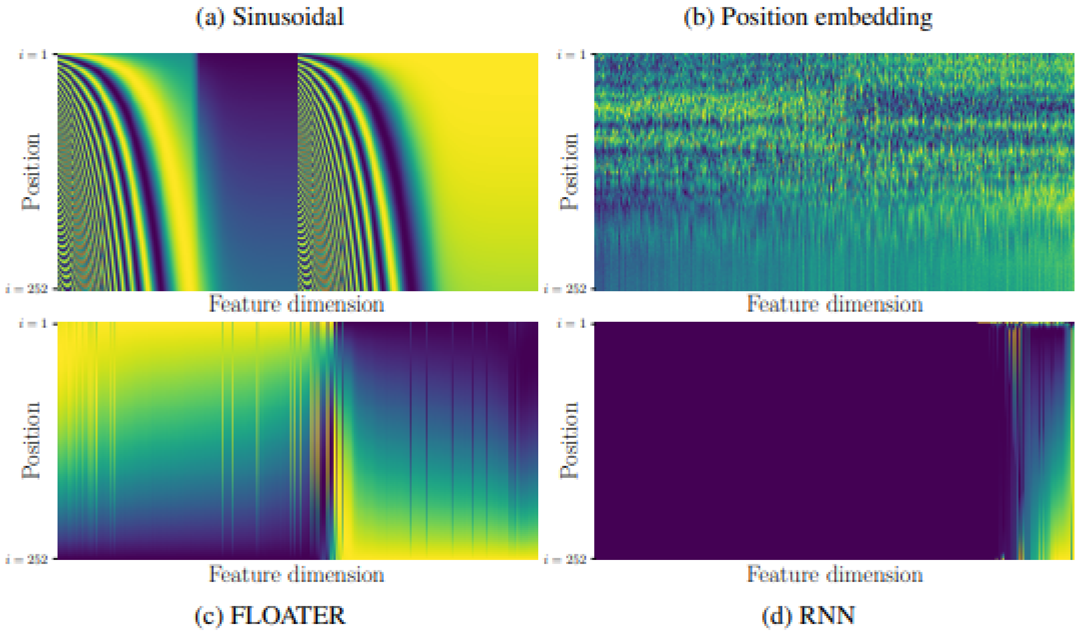
Answer:

(a). To represent a document, we can add up the one-hot embedding vectors that represent words present in the document. This will give a vocabulary length vector that is 0 everywhere except for words that appear in the document, whose value is the number of times that word appears in the document.
(b). This representation is problematic because it does not capture meaning encoded in the order of words, and it also does not create a space where similar words are close together.

- ”This cat looks like a dog”
- ”This dog looks like a cat”

Both of these documents will have the same document embedding under the scheme described above, but have different meanings, which are encoded in the word order.
(c) Solutions to this are:

- (1) N-gram embeddings, where each token is a tuple of the n words that precede it in the sequence;



(a) Sinusoidal

(b) Position embedding

(c) FLOATER

(d) RNN

Figure 1: Position embeddings

- and (2) positional coding, where a position-specific vector is added to the input token for the model to learn the position of the token, as shown in figure 1

Problem 2

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.

Answer:

The resulting unfolded network is just a bunch of linear transformations and a sigmoid non-linearity. In other words, we are fitting a sigmoid curve to separate two classes. The output is a number between 0 and 1, where 1 represents certainty for one class, and 0 represents certainty for the other class.

Problem 3

Specify weight matrices U, V, W, bias vector bh, and scalar bias by for a recurrent neural network that takes two inputs as binary numbers and produces their sum in binary.

Answer:

I got the simulation to work when the first operand is 0. But when it gets to 1 it crashes. I attached my work in a separate file in the github. Here's my solution:

$$\begin{array}{cccc} \overset{3\times 3}{U} & \overset{3\times 3}{W} & \overset{3\times 1}{V} & \vec{b}_h \\ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 2 \\ -2 \\ 2 \end{bmatrix} & \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \end{array}$$

Problem 4

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

Answer:

In the context of recurrent neural networks, regularization helps reduce overfitting and improve generalization. A common regularization technique used in RNNs is dropout, in which random values from output of previous layer are zeroed out. This forces each individual neuron in the hidden layers to be more robust and can improve the networks performance when anomalies are present in the input. In 2015, Zaremba et. al. showed that it is better to use dropout within a recurrent cell, that is, between layers of the architecture, but not between unfolding cells. All hidden information carries through to the next unfolded cell without dropout, but within the cell architecture layers there is still dropout. They found this is a good way to regularize recurrent networks.

Problem 5 (EC)

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

Answer:

In Seq2Seq tasks, one recurrent network architecture is to use output-to-input connections between cells. Teacher forcing is the random input of ground truth to the next cell instead of the output of the previous cell. Doing this has two benefits. Firstly, a model can recover from mistakes as it produces the output during training. Secondly, the cells with ground truth as input can produce their output tokens in parallel, as, they do not rely on the previous tokens having already been generated. Specifically, if the teacher forcing ratio is 25%, you can choose 25% of the tokens and begin by computing them immediately and in parallel. Then for the rest of the sequence, as your network unfolds from the beginning and reaches the precomputed outputs, it can use the already stored result rather than recomputing.