AMCS 215
Mathematical Foundations of Machine Learning
Assignment 10 and mini-project plan
Due Nov 19, 2023

Reading material:

- Chapter 15 from Reference 1 (RNNs), and/or Chapters 9-10 of `https://d2l.ai`
- Section 15.1 and 15.2 from `https://d2l.ai` (word2vec) or alternatively Section 20.2 from Ref 1

Announcements:

- please submit a short description of your mini-project (no longer than one-page) separately form the rest of the assignment. There is a separate submission area on blackboard. Please upload one submission per group.

1. RNN for classification.
At the link bellow, you will find a notebook that builds a character RNN for identifying the language of an input name
`https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html`
As written the code builds the RNN "from scratch" using dense layers and manually concatenates the hidden state and input vector.

- rewrite the RNN using the `RNNCell` pytorch building block, which keeps the hidden states and inputs logically separate. Your forward method should loop through the input tokens to generate its output.
- rewrite the RNN using the `RNN` pytorch building block.
- Do you think using an LSTM instead of the vanilla RNN will improve the accuracy that the trained network can reach? Justify very briefly.

2. RNN for generation.
At the link bellow, you will find a notebook that builds a character RNN for *generating* names in a given target language.
`https://pytorch.org/tutorials/intermediate/char_rnn_generation_tutorial.html`

- sketch the RNN network using an RNN Cell as the building block. Write the equations that define a forward pass through the network.
- describe in a short sentence how the network is trained.
- rewrite the RNN using either the `RNNCell` or the `RNN` pytorch block.
- suppose you were given the task of evaluating the performance of the network after training. How might you come up with an evaluation metric for this task. (*Note.* this is a somewhat open-ended question. Think about what kind of test data you may need to obtain and how you might use the data.)
- do you think you will obtain better results with a bigger network? How about with an LSTM that replaces the vanilla RNN?

3. Understanding ord2vec.
`word2vec` is an algorithm for producing word embeddings given a corpus of text. Sections 15.1.3 and 15.2.1 from `https://d2l.ai` describe a version of the algorithm called skip-gram. You are to read these sections and briefly answer the questions below.

In skip-gram `word2vec`, we loop over windows of the text, where each window has a center word (index c) and neighboring or outside context words (we will use o to refer to their indices). Our goal is to learn the probability distribution $P(w_o \mid w_c)$ from the text. This is the probability that word o is an outside word for c (i.e., falls inside the window centered at c). The probability is modeled by Eq 15.1.4. For

each word, we learn two vectors: a vector v when the word is a center vector and vector u when the word is an outside word. The optimization parameters are the u and v vectors for all words. The final word embeddings use the average of u and v for every word, or may simply use v.

- explain how the softmax training loss of Eq 15.1.6 is the cross entropy between the predicted and the true distribution.
- (re-)derive the gradient expression of Eq 15.1.8, the gradient of the loss with respect to v_c
- the gradient expression above is the difference between two terms. Provide an interpretation of how driving it towards zero improves the learned vector v_c. (*Hint.* "observed - expected")
- what is the computational complexity for calculating each gradient? What could be the issue if the vocabulary size size is huge?
- in order to reduce the computational complexity, an alternative loss function, called negative sampling loss is introduced as an alternative to the softmax loss. The expression of the negative sampling loss ($-\log P(w_o \mid w_c)$) is described by Eq 15.2.6. Explain this expression in a few words, and why it is a reasonable loss to use?
- derive an expression of the gradient of the negative sampling loss with respect to v_c. Does the computational complexity of evaluating this gradient increase if the vocabulary size increases. Comment.

4. Mini-project plan.
In the remainder of this term, you will be working on an appropriate ML problem of your choice. For this submission, you are asked to develop your basic idea into a one-page description. Your description should be a clear plan for execution. Please identify the task, the data you need (and where you will get it from), and your plan for evaluating the quality of your solution.

You will be asked to give a 5-minute presentation of your project during the last week of class (Dec 3).