TAAC/MECD/FEUP 2023/2024

Assignment 2

To be solved in groups of at most four elements

Submit by December 2, 2023, 11:59 PM at Moodle. Only a member of the group submits the work. The other member(s) of the group only submit a txt file stating "joint submission with [Colleague Name]".

1. **Convolutional Neural Networks (50%)**

   (a) (15%) Describe, in your own words, the expected behaviour in a convolutional layer, in terms of output and input dimensions, if we:

   i. (5%) Change the padding from 0 to 1.

   ii. (5%) Change the kernel size from 3 to 4.

   iii. (5%) Change the stride from 1 to 2

   (b) (15%) Frequently, after a convolutional layer and respective activation, it is possible to find a pooling layer. Describe this layer and comment the following statement: "Pooling layers introduce a trade-off between information loss and dimensionality reduction".

   (c) (20%)

   i. (10%) Build a custom CNN composed of 5 convolutional layers with ReLU activation between all the five layers and a fully-connected output layer. From the second to the third convolutional layer add a pooling layer. Train this CNN on PyTorch's Fashion-MNIST dataset (PyTorch Class), using Adam, a learning rate of 0.01 and Cross-Entropy. Feel free to define the hyperparameters of your convolutional and pooling layers. Train as many epochs as you find necessary. Report your accuracy, loss and number of epochs to convergence (if the network converges).

   ii. (5%) Redo the previous exercise but use a ResNet-18 without pre-trained weights. Train as many epochs as you find necessary. Report your accuracy, loss and number of epochs to convergence (if the network converges)

   iii. (5%) Repeat the previous exercise, but using a pre-trained ResNet-18. Train as many epochs as you find necessary. Report your accuracy, loss and number of epochs to convergence (if the network converges)

2. **Natural language processing (50%)**

   (a) (15%) You are in the process of training a machine learning model and you anticipate the occurrence of numerous out-of-vocabulary words during test time. Given this consideration, what type of tokenization would you opt for and why?

   (b) (10%) The following vocabulary has been constructed from a collection of 100 documents denoted as C:

word	"and"	"apple"	"banana"	"eat"	"hate"	"I"	"pie"	"strawberry"	"the"	"they"
no. docs.	90	30	15	40	10	60	20	5	85	30

Table 1: Vocabulary and frequency of appearance in collection C.

   Considering the sentence "You and I eat the strawberry pie":

   i. (5%) Compute its bag-of-words representation.

   ii. (5%) Compute its TF-IDF representation.

(c) (25%) Write the code for a PyTorch `nn.Module` defining an LTSM- or GRU-based language model. In addition to the standard `__init__` and `forward` methods, you should implement a `generate` method. This method should take an input token (`x`), the initial state of the RNN (`h0`), and the desired number of tokens (`no`) to be generated, and return the decoded text as a sequence of token indices (integers). Implement the generation using the greedy decoding algorithm. Training the model or providing the code for the training routine is not required, and no extra credit will be awarded for doing so. Please ensure that your code is well-documented with comments and docstrings. The latter should clearly explain the purpose, type, and shape (if applicable) of all the inputs and outputs for each method.

DPC, PDNC