Homework Assignment 4

CSE 190: Neural Networks

Fall 2015

Instructions

Due on Friday, Dec 4^{th} 11:59pm PST

1. You are encouraged to form a group of 2-3 people to finish this homework assignment.
2. You are encouraged to make a LaTeX report for each assignment. You are free to choose the template you want.
3. You can submit the homework electronically to **both** TAs' Emails (pawang@ucsd.com; liam.fedus@gmail.com), or submit a hard-copy to the TA's office (EBU3B 4154).
4. You may use a language of your choice (Python and MATLAB are recommended) but all source code used in the assignment must be attached in the appendix. Please keep the code clean with explanatory comments, as they may be reused in the future.
5. Late policy: You will get deduction from the total score for: a) 0-1 day:10%; b) 1-2 days: 20%; c) 2-3 days: 50%; d) more than 3 days: 100%;
6. Any form of cheating, lying, or plagiarism will not be tolerated. Discussions on course materials and homework solutions are allowed, but you should write the final solutions alone. Books, notes, and Internet resources can be consulted, but not copied from. Working with people outside your team on homeworks must follow the (spirit of) Gilligan's Island Rule (Dymond, 1986): No notes can be made during a discussion, and you must watch one hour of Gilligan's Island or equally mindless TV before writing anything down. Suspected cheating will be reported to the Dean.

Recurrent Neural Networks

In this assignment, you will design and implement a character-level Recurrent Neural Network (RNN).

1. **Find and Load Training Text.**
   (a) Choose the text you want your neural network to learn, but keep in mind that your data set must be quite large in order to learn the structure! RNNs have been trained on highly diverse texts (novels, Eminem lyrics, Linux Kernel, etc.) with success, so you can get creative. As one easy option, Gutenberg Books is a source of free books where you may download full novels in a .txt format.
   (b) We will use a *character-level* representation for this model. To do this, you may use extended ASCII with 256 characters. As you read your chosen training set, you will read in the *characters* one at a time into a one-hot-encoding, that is, each character will map to a vector of ones and zeros, where the one indicates which of the characters is present,

$$\text{char} \mapsto [0, 0, \cdots, 1, \cdots, 0, 0] \tag{1}$$

   Your RNN will read in these length-256 binary vectors as input.

2. **Implement an RNN.** You should be able to leverage your model developed in the feed-forward neural network in Homework Assignment 2.

   (a) **Back-propagation Through Time:** In an RNN with a single hidden layer you should have three set of weights: W_{xh} (from the input layer to the hidden layer), W_{hh} (the recurrent connection in the hidden layer), and W_{ho} (from the hidden layer to the output layer). Suppose you use *Softmax* units for the output layer and *Tanh* units for the hidden layer, show:

      i. Write the equation for the activation at time step t in the hidden layer and the equation for the output layer in the **forward propagation** step.

      ii. Write the equation for the **weight update rule** at time step t for W_{xh}, W_{hh}, and W_{ho} in a vectorized notation. Suppose the backpropagation goes back to time step k $(0 < k < t)$.

   (b) **Network Training:** Train your recurrent neural network using the dataset you created in 1(b). You are free to choose learning parameters (sequence length, learning rate, stopping criteria, etc.). Complete the following task:

      i. Report your training procedure. Plot the training loss vs. the number of training epochs.

      ii. During training, choose 5 breaking points (for example, you train the network for 100 epochs and you choose the end of epoch 20,40,60,80,100) and show how well your network learns through time. You can do it by feeding in the network a chunk of your training text and show what is the output of the network. Report your result.

      iii. To add randomness into the outputted sequence, we can either change the beginning character, we can randomly initialize the hidden state or we can introduce randomness via the Softmax function. In this case, if we have already seen sequence $x = (x_1, \cdots, x_t)$ of characters, then the probability that the output will be character j at time-step $t+1$, i.e. $y_{t+1} = j$, is given by the Softmax function,

$$p(y_{t+1} = j|x) = \frac{e^{x^T w_j}}{\sum_k e^{x^T w_k}} \mapsto \frac{e^{(x^T w_j)/\tau}}{\sum_k e^{(x^T w_k)/\tau}} \tag{2}$$

      where $\tau \in (0, \infty)$ is the temperature. Intuitively, at higher temperatures $\tau \to \infty$, all characters will be chosen equally likely and at low temperatures $\tau \to 0$, only the most probable character will be chosen with nearly probability 1.0. So in order to generate the next character of the sequence, simply sample from this probability. Please report your result of 5-different generated texts of length-100 once your model is fully trained. Try this for three different temperatures τ. What dynamics do you expect and what do you see?

   (c) **Experiment with Network Structure:** As before, we want to explore how the network learns when we change parameters:

      i. **Number of hidden units.** Try doubling and halving your number of hidden units. Like in 2(b), plot the training loss vs. the number of training epochs and show your text sampling results. Discuss your findings.

      ii. **Sequence length.** Try doubling and halving your length of sequence that feeds into the network. Like in 2(b), plot the training loss vs. the number of training epochs and show your text sampling results. Discuss your findings.

Extra Credit. (+20% for Homework Assignment 4, maximum of 120%)
As an extra credit, you may generalize your RNN to utilize LSTM units. This is a non-trivial generalization, but you may find the equations of Chapter 4 of Alex Graves' thesis to be very useful. http://www.cs.toronto.edu/ graves/preprint.pdf. As an additional resource for RNNs, the blog Awesome-RNN is extremely helpful. Show your result in the format of 2(b).