Homework #3 – Convolutional and Recurrent Neural Networks
CAP 5619, Deep & Reinforcement Learning (Spring 2020), Department of Computer Science, Florida State University

---

Points: 85
Due: Beginning of the class (11:00am) on Thursday, March 12th, 2020

Submission: You need to submit electronically via Canvas by uploading a) a pdf file (named "**hw3-Firstname-Lastname.pdf**") for your answers to the questions, and b) the program(s) you have created (named as "**hw3-prog-Firstname-Lastname.???**"); if there are multiple program files, please zip them as a single archive. Here replace "Firstname" using your first name and replace "Lastname" using your last name in the file names

The main purpose of this assignment is to gain a deeper understanding of convolutional neural networks and recurrent neural networks.

Problem 1 (20 points) In the deep learning framework you have established, train a convolutional neural network or obtain a pretrained model on MNIST. You can use any program available to you as long as there are at least three convolutional layers and the accuracy on the training set is at least 95%. Answer the following questions:

(1) Visualize the filters (as images) in the first three convolution layers.
(2) Visualize the feature map of the third convolutional layer (i.e., the output from the third convolutional layer) for a digit '0' and a digit '8' (which are classified correctly). By comparing the two feature maps, could you tell the important discriminant features between the two digits (i.e., the features that can be used to distinguish them)?
(3) Shift a digit '1' (which is classified correctly initially) to the left three pixels, and to the right three pixels, will the prediction by the network change? Then classify them using the convolutional neural network you have trained or obtained. Here fill the missing values using the closest valid ones (i.e., using the clamp border padding method).

Problem 2 (20 points) Here we empirically test the robustness of the model you have for Problem 1. Imagine that we have a 8x8 black patch and we move it from left to right, from top to down with a stride of 1 to occlude part of the image. Using a digit '6' (which is classified correctly initially) as an example, answer the following questions.

(1) Create three maps (i.e., images) in the following way. For each position of the black patch, store the probability of '6' of the partially covered image in map 1, the highest probability (among the 10 classes) in map 2, and classified label ('0' to '9') in map 3. Display the maps. Make sure that they are clearly legible by scaling values.
(2) By analyzing the maps, explain which parts of the '6' are important for recognition.
(3) Based on your result, would you be able to create "adversarial" images (i.e., to be classified as another digit) by covering some parts of '6' using patches from the images of the other digits? (See "The Elephant in the Room," available from https://arxiv.org/pdf/1808.03305.pdf for examples on other datasets).

Problem 3 (30 points) The main purpose of this problem is to gain a deeper understanding of the back-propagation through time algorithm for a recurrent neural network via an example. We will use the recurrent neural network defined by equations (10.8) to (10.11) (in the Deep Learning textbook) with a customized loss function:

$$L\left(\{x^{(1)}, \cdots, x^{(\tau)}\}\right) = (\hat{y}_1^{(\tau)} - 0.5)^2 - \log(\hat{y}_2^{(\tau)}) \, .$$

and the following parameter values:

$$b = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}, \quad W = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}, \quad U = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix}, \quad V = \begin{bmatrix} -2 & 1 \\ -1 & 0 \end{bmatrix}.$$

for the following sequence:

$$x^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad x^{(2)} = \begin{bmatrix} 0.50 \\ 0.25 \end{bmatrix}, \quad x^{(3)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

(1) Write a program that computes the outputs and loss using the given sequence and parameter values. Give $\hat{y}^{(t)}$ for t=1:3 and the customized loss.
(2) Estimate the gradient of the loss function with respect to b_1 and b_2 using the central difference method using ε=0.0001.
(3) Compute the gradient of the loss function with respect to b_1 and b_2 by unfolding the network through time. You need to show the intermediate results.
(4) Fixing other parameters, perform one step of gradient descent optimization on b_1 and b_2 using a learning rate of 0.002.
(5) Use your program to compute the loss using the new values for b (with other parameters as given) for the original sequence.

Problem 4 (**15 points**) For the same model defined by equations (10.8) to (10.11) (in the Deep Learning textbook) as in the previous question, now we use the network to learn how to classify long protein sequences consisting of thousands of inputs.

(1) Which of the parameters are most difficult to learn? Explain briefly.
(2) Suppose that we treat the network as an echo state network, which weights should be fixed and which weights should be learned?
(3) Explain why using an LSTM cell could help overcome some of the difficulties.

Extra Credit Problem

Problem 5 (8 points) Implement an LSTM cell and use it to replace the recurrent connection in Problem 3. Use the same U and W matrices as in Problem 3 but choose biases properly. Then apply your LSTM on the same sequence as in Problem 3, compute the outputs, the customized loss, and the gradient of the customized loss (by unfolding the network through time) with respect to the biases of the forget gate. For gradient calculation, you need to show the intermediate results.