Homework #3 –Convolutional and Recurrent Neural Networks - Solutions

CAP 5619, Deep & Reinforcement Learning (Spring 2020), Department of Computer Science, Florida State University

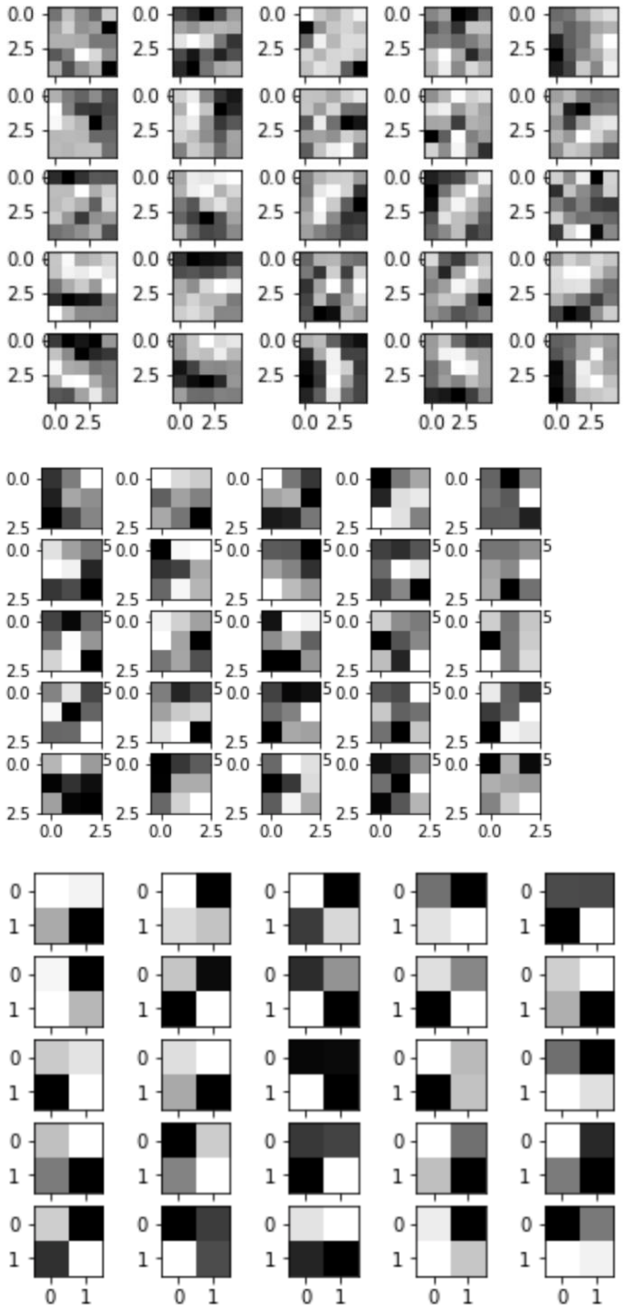
---

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

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:

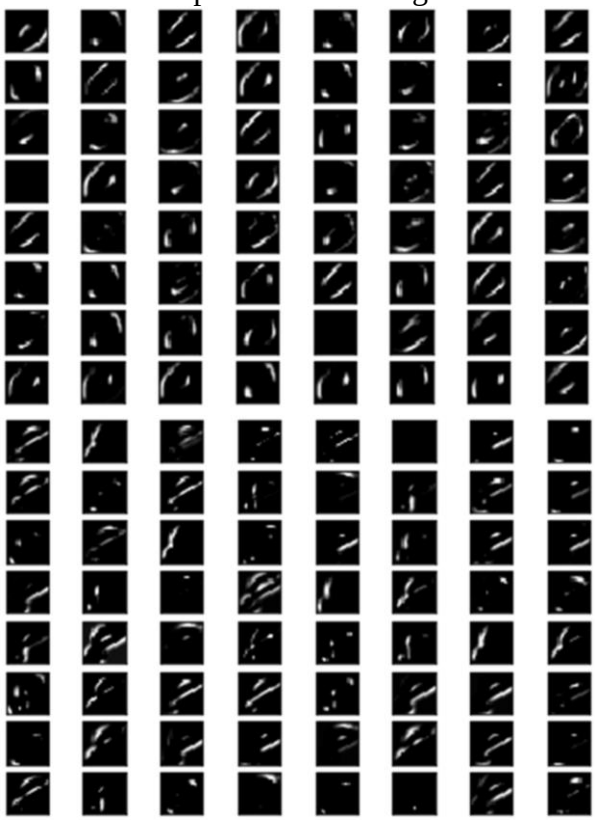
(1) Visualize the filters (as images) in the first three convolution layers.

It depends on the design of your network. The visualization may look like the following. Note that here visualization of some filters from first three convolution layers are given as an example:



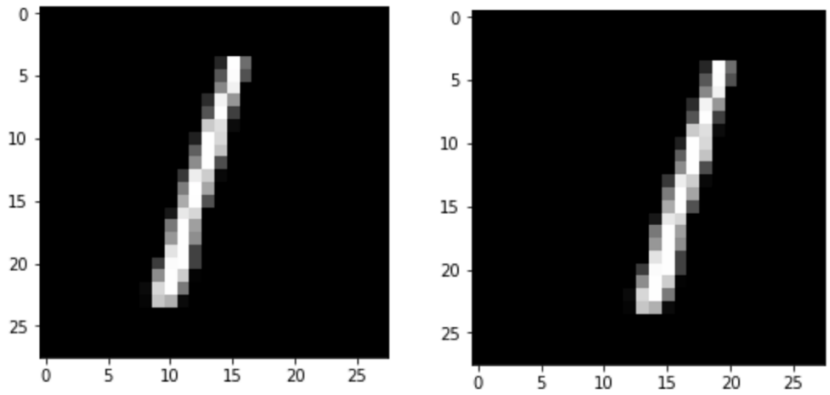
(2) Visualize the feature map of the third convolutional layer (i.e., the output from the third convolutional layer) for a digit ‘0’ and a digit ‘8’ (which are classified correctly). By comparing the two feature maps, could you tell the important discriminant features between the two digits (i.e., the features that can be used to distinguish them)?

The feature map for 0 and 8 are given following (Thanks to Yunzheng for the solution):



Here, for digit 0, we get more curves, whereas it is less visible for digit 8.

(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).

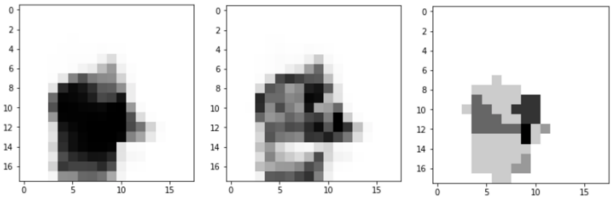


The prediction may or may not get changed depending on the network, parameters and the image you have chosen.

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.

Depends on the implementation, it may look like the following:



(2) By analyzing the maps, explain which parts of the '6' are important for recognition.

The main curves comprising the digit '6' are most important. In addition, the round portion of the digit is also important part 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).

It is possible to create adversarial images like what the authors did in the paper 'Elephant in the Room'. We can take one image of 6 from the intermediate image pool of 6 and cover some part of it with another digit's patched image to create adversarial examples.

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.

The output and loss using the given sequence and parameter values are given bellow:

output y1 : [0.94921601 0.05078399]
output y2 : [0.95221995 0.04778005]
output y3 : [0.94001124 0.05998876]

loss: 3.007207986556634

(2) Estimate the gradient of the loss function with respect to b_1 and b_2 using the central difference method using $\varepsilon = 0.0001$.

$\partial L/\partial b_1 = (\text{loss}(b1+ \varepsilon)- \text{loss}(b1- \varepsilon))/2\varepsilon = -0.009961388396373394$
$\partial L/\partial b_2 = (\text{loss}(b2+ \varepsilon)- \text{loss}(b2- \varepsilon))/2\varepsilon = 0.43400995503928286$

(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.

$\partial L/\partial b_1 = -0.009961388396373394$
$\partial L/\partial b_2 = 0.43400995503928286$

The intermediate detailed calculation and results are given below:

Problem 3-3 solution:

From the textbook, we have the equations (10.8 → 10.11)
they are:

$$a^{(t)} = b + W h^{(t-1)} + U x^{(t)}$$

$$h^{(t)} = \tanh(a^{(t)})$$

$$o^{(t)} = c + V h^{(t)}$$

$$\hat{y}^{(t)} = \text{softmax}(o^{(t)})$$

By unfolding the network through time →
we get that,

$$a_1^{(t)} = b_1 + W_{11} h_1^{(t-1)} + W_{12} h_2^{(t-1)} + U_{11} x_1^{(t)} + U_{12} x_2^{(t)}$$

$$a_2^{(t)} = b_2 + W_{21} h_1^{(t-1)} + W_{22} h_2^{(t-1)} + U_{21} x_1^{(t)} + U_{22} x_2^{(t)}$$

$$h_1^{(t)} = \tanh(a_1^{(t)})$$

$$h_2^{(t)} = \tanh(a_2^{(t)})$$

$$o_1^{(t)} = c_1 + V_{11} h_1^{(t)} + V_{22} h_2^{(t)}$$

$$o_2^{(t)} = c_2 + V_{21} h_1^{(t)} + V_{22} h_2^{(t)}$$

$$\hat{y}_1^{(t)} = \frac{e^{o_1^{(t)}}}{e^{o_1^{(t)}} + e^{o_2^{(t)}}}$$

$$\hat{y}_2^{(t)} = \frac{e^{o_2^{(t)}}}{e^{o_2^{(t)}} + e^{o_2^{(t)}}}$$

Given, loss,

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

We want to find,

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial b_1} + \frac{\partial L}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial b_1} = ? \quad \text{— (1)}$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial b_2} + \frac{\partial L}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial b_2} = ? \quad \text{— (2)}$$

We can get,

$$\frac{\partial L}{\partial a_1^{(3)}} = \left[\frac{\partial L}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial h_1^{(3)}} + \frac{\partial L}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial h_1^{(3)}} \right] \cdot \frac{\partial h_1^{(3)}}{\partial a_1^{(3)}}$$

$$= \left\{ \left[2\left(\hat{y}_1^{(3)} - 0.5\right)\left(\hat{y}_1^{(3)} - \left(\hat{y}_1^{(3)}\right)^2\right) + y_1^{(3)} \right] \cdot v_{11} - \left[2\left(\hat{y}_1^{(3)} - 0.5\right) \hat{y}_1^{(3)} \hat{y}_2^{(3)} + \frac{1}{\hat{y}_2^{(3)}}\left(\hat{y}_2^{(3)} - \left(\hat{y}_2^{(3)}\right)^2\right) \right] \cdot v_{21} \right\} \cdot \left[1 - \left(h_1^{(3)}\right)^2 \right]$$

$$= -0.00991676$$

By similar calculation,

$$\frac{\partial L}{\partial a_2^{(3)}} = 0.422889$$

Again,

$$\frac{\partial a_2^{(3)}}{\partial b_1} = w_{21} \frac{\partial h_1^{(2)}}{\partial b_1} + w_{22} \cdot \frac{\partial h_2^{(2)}}{\partial b_1}$$

$$\frac{\partial a_2^{(2)}}{\partial b_1} = w_{21} \cdot \frac{\partial h_1^{(1)}}{\partial b_1} + w_{22} \cdot \frac{\partial h_2^{(1)}}{\partial b_1}$$

$$\frac{\partial a_1^{(3)}}{\partial b_1} = 1 + w_{11} \frac{\partial h_1^{(2)}}{\partial b_1} + w_{12} \frac{\partial h_2^{(2)}}{\partial b_1}$$

$$\frac{\partial a_1^{(2)}}{\partial b_1} = 1 + w_{11} \frac{\partial h_1^{(1)}}{\partial b_1} + w_{12} \frac{\partial h_2^{(1)}}{\partial b_1}$$

$$\frac{\partial h_1^{(2)}}{\partial b_1} = \frac{\partial h_1^{(2)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial b_1} = \left(1 - h_1^{(2)}\right)^2 \cdot \frac{\partial a_1^{(2)}}{\partial b_1} = 0.004500106$$

$$\frac{\partial h_2^{(2)}}{\partial b_1} = \frac{\partial h_2^{(2)}}{\partial a_2^{(2)}} \cdot \frac{\partial a_2^{(2)}}{\partial b_1} = 0$$

$$\frac{\partial h_1^{(1)}}{\partial b_1} = \frac{\partial h_1^{(1)}}{\partial a_1^{(1)}} \cdot \frac{\partial a_1^{(1)}}{\partial b_1} = \left(1 - \left(h_1^{(1)}\right)^2\right) \cdot 1 = 0.07065$$

$$\frac{\partial h_2^{(1)}}{\partial b_1} = \frac{\partial h_2^{(1)}}{\partial a_2^{(1)}} \cdot \frac{\partial a_2^{(1)}}{\partial b_1} = 0$$

So,

$$\frac{\partial a_1^{(1)}}{\partial b_1} = 1$$

$$\frac{\partial a_2^{(1)}}{\partial b_1} = 0$$

$$\frac{\partial a_1^{(2)}}{\partial b_1} = 1.07065$$

$$\frac{\partial a_2^{(2)}}{\partial b_1} = 0$$

$$\frac{\partial a_1^{(3)}}{\partial b_1} = 1.004500106$$

$$\frac{\partial a_2^{(3)}}{\partial b_1} = 0$$

By putting these values in (1),

$$\frac{\partial L}{\partial b_1} = -0.00996\,1388 \quad \text{—— (3)}$$

we perform a similar calculation with respect to b_2 and put the values at (2),

$$\frac{\partial h_1^{(1)}}{\partial b_2} = \frac{\partial h_1^{(1)}}{\partial a_1^{(1)}} \cdot \frac{\partial a_1^{(1)}}{\partial b_2} = 0$$

$$\frac{\partial h_2^{(1)}}{\partial b_2} = \frac{\partial h_2^{(1)}}{\partial a_2^{(1)}} \cdot \frac{\partial a_2^{(1)}}{\partial b_2} = \left(1 - (h_2^{(1)})^2\right) \cdot 1$$

$$= 0.0706508$$

$$\frac{\partial a_1^{(2)}}{\partial b_2} = w_{11} \frac{\partial h_1^{(1)}}{\partial b_2} + w_{12} \frac{\partial h_2^{(1)}}{\partial b_2} = -0.07065$$

$$\frac{\partial a_2^{(2)}}{\partial b_2} = 1 + w_{21} \frac{\partial h_1^{(1)}}{\partial b_2} = 1.1413016$$

$$\frac{\partial h_1^{(2)}}{\partial b_2} = \frac{\partial h_1^{(2)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial b_2} = -0.00029695$$

$$\frac{\partial h_2^{(2)}}{\partial b_2} = \frac{\partial h_1^{(2)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial b_2} = 0.012997$$

$$\frac{\partial a_1^{(3)}}{\partial b_2} = w_{11} \frac{\partial h_1^{(2)}}{\partial b_2} + w_{12} \cdot \frac{\partial h_2^{(2)}}{\partial b_2} = -0.0132897$$

$$\frac{\partial a_2^{(3)}}{\partial b_2} = 1 + w_{21} \cdot \frac{\partial h_1^{(2)}}{\partial b_2} + w_{22} \cdot \frac{\partial h_2^{(2)}}{\partial b_2}$$

$$= 1.025985517$$

So, $$\frac{\partial L}{\partial b_2} = 0.43400\,9953 \quad \text{—— (4)}$$

(4) Fixing other parameters, perform one step of gradient descent optimization on b1 and b2 using a learning rate of 0.002.

$b_1 = b_1 - lr*\partial L/\partial b_1 = 0.99998008$

$b_2 = b_2 - lr*\partial L/\partial b_2 = 0.99913184$

(5) Use your program to compute the loss using the new values for b (with other parameters as given) for the original sequence.

The loss with the new b = 3.006830735098699

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.

The parameters most difficult to learn are: the recurrent weights mapping from $h^{(t-1)}$ to $h^{(t)}$, the input weights mapping from $x^{(t)}$ to $h^{(t)}$. RNNs get difficulties in learning long-term dependencies (dependencies between steps that are far apart) because of the occurred vanishing and exploding gradient problem.

(2) Suppose that we treat the network as an echo state network, which weights should be fixed and which weights should be learned?

In echo state network, early layers are made random and fixed. Then it just learns the last layer which is a linear model that uses the transformed inputs to predict the target outputs. So, regarding echo state network: a) fix the input to hidden connections and hidden to hidden connections at random values, b) learn the hidden to output connections

(3) Explain why using an LSTM cell could help overcome some of the difficulties.

There are two difficulties RNNs face – exploding gradient, vanishing gradient. Exploding gradient occurs when the algorithm assigns a stupidly high importance to the weights. However, this problem can be solved by truncate or squash of the gradients. On the other hand, vanishing gradient occurs when the values of a gradients are too small. As a result, the model stops learning or takes long period of time. Vanishing gradient is much harder problem than the exploding gradient.

The vanishing gradient problem can be solved by Long Short-Term Memory (LSTM) networks. They are actually an extension for RNNs. LSTM can mainly extend their memory which enables RNNs to remember their inputs over a long period of time via cell states implemented using self loops. LSTMs contain their information in a memory. Moreover, it can read, write and delete information from its memory. By exploiting these features, it learns over time which information is important and which is not. The problem of vanishing gradients is solved by LSTM as it keeps the gradients steep enough as well as keeps the training time comparatively short and the accuracy high.

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.