Homework 2: Convolutional Neural Networks and Recurrent Neural Networks

CSCI-GA 2572 Deep Learning

Spring 2025

The goal of homework 2 is to get you to work with convolutional neural networks and recurrent neural networks.

In the theoretical part, you will work on figuring out how backpropagation works in these networks. In part 2, we will implement and train them.

In part 1, you should submit all your answers in a pdf file. As before, we recommend using LaTeX.

For part 2, you will implement some neural networks by adding your code to the provided ipynb file.

As before, please use numerator layout.

The due date of homework 2 is 11:59pm 02/23. Submit the following files in a zip file `your_net_id.zip` to CampusWire:

- `hw2_theory.pdf`
- `hw2_cnn.ipynb`
- `hw2_rnn.ipynb`
- `seq_classification.ipynb`

The following behaviors will result in penalty of your final score:

1. 10% penalty (1.5/60) for submitting your file without using the correct naming format (including naming the zip file, PDF file or python file wrong, adding extra files in the zip folder, like the testing scripts in your zip file).
2. 10% penalty (1.5/60) for late submission every 24 hours past the deadline. We will not accept any late submission after 4 days (96 hours) of lateness.
3. 20% penalty (3/60) for code submission that cannot be executed following the steps we mentioned.

1 Theory (50pt)

1.1 Convolutional Neural Netoworks (15 pts)

(a) (1 pts) Given an input image of dimension 19×26, what will be output dimension after applying a convolution with 4×4 kernel, stride of 2, and padding of 1?

(b) (2 pts) Given an input of dimension $C \times H \times W$, what will be the dimension of the output of a convolutional layer with kernel of size $H_K \times W_K$, padding P, stride S, dilation D, and F filters. Assume that $H \geq H_K$, $W \geq W_K$.

(c) (12 pts) In this section, we are going to work with 1-dimensional convolutions. Discrete convolution of 1-dimensional input $x[n]$ and kernel $k[n]$ is defined as follows:

$$s[n] = (x * k)[n] = \sum_m x[n-m]k[m]$$

However, in machine learning convolution is usually implemented as cross-correlation, which is defined as follows:

$$s[n] = (x * k)[n] = \sum_m x[n+m]k[m]$$

Note the difference in signs, which will get the network to learn an "flipped" kernel. In general it doesn't change much, but it's important to keep it in mind. In convolutional neural networks, the kernel $k[n]$ is usually 0 everywhere, except a few values near 0: $\forall_{|n|>M} k[n] = 0$. Then, the formula becomes:

$$s[n] = (x * k)[n] = \sum_{m=-M}^{M} x[n+m]k[m]$$

Let's consider an input $x[n] \in \mathbb{R}^7$, with $1 \leq n \leq 11$, e.g. it is a length 11 sequence with 7 channels. We consider the convolutional layer f_W with one filter, with kernel size 3, stride of 2, no dilation, and no padding. The only parameters of the convolutional layer is the weight W, $W \in \mathbb{R}^{1\times 7\times 3}$, there's no bias and no non-linearity.

(i) (1 pts) What is the dimension of the output $f_W(x)$? Provide an expression for the value of elements of the convolutional layer output $f_W(x)$. Example answer format here and in the following sub-problems: $f_W(x) \in \mathbb{R}^{42\times 42\times 42}$, $f_W(x)[i,j,k] = 42$.

(ii) (4 pts) What is the dimension of $\frac{\partial f_W(x)}{\partial W}$? Provide an expression for the values of the derivative $\frac{\partial f_W(x)}{\partial W}$.

(iii) (4 pts) What is the dimension of $\frac{\partial f_W(x)}{\partial x}$? Provide an expression for the values of the derivative $\frac{\partial f_W(x)}{\partial x}$.

(iv) (5 pts) Now, suppose you are given the gradient of the loss ℓ w.r.t. the output of the convolutional layer $f_W(x)$, *i.e.* $\frac{\partial \ell}{\partial f_W(x)}$. What is the dimension of $\frac{\partial \ell}{\partial W}$? Provide an expression for $\frac{\partial \ell}{\partial W}$. Explain similarities and differences of this expression and expression in (i).

1.2 Recurrent Neural Networks (30 pts)

1.2.1 Part 1

In this section we consider a simple recurrent neural network defined as follows:

$$c[t] = \sigma(W_c x[t] + W_h h[t-1]) \tag{1}$$
$$h[t] = c[t] \odot h[t-1] + (1 - c[t]) \odot W_x x[t] \tag{2}$$

where σ is element-wise sigmoid, $x[t] \in \mathbb{R}^n$, $h[t] \in \mathbb{R}^m$, $W_c \in \mathbb{R}^{m \times n}$, $W_h \in \mathbb{R}^{m \times m}$, $W_x \in \mathbb{R}^{m \times n}$, $\odot$ is Hadamard product, $h[0] \doteq 0$.

(a) (4 pts) Draw a diagram for this recurrent neural network, similar to the diagram of RNN we had in class. We suggest using diagrams.net.

(b) (1pts) What is the dimension of $c[t]$?

(c) (3 pts) Derive expressions for $\frac{\partial h[t]}{\partial h[t-1]}$ and $\frac{\partial c[t]}{\partial h[t-1]}$

(d) (4 pts) Suppose that we run the RNN to get a sequence of $h[t]$ for t from 1 to K. Assuming we know the derivative $\frac{\partial \ell}{\partial h[t]}$, provide dimension of and an expression for the value of $\frac{\partial \ell}{\partial W_x}$ in terms of ℓ, h, W_x.
What are the similarities of backward pass and forward pass in this RNN?

(e) (2pts) Can this network be subject to vanishing or exploding gradients? Why?

1.2.2 Part 2

We define an AttentionRNN(2) as

$$q_0[t], q_1[t], q_2[t] = Q_0 x[t], Q_1 h[t-1], Q_2 h[t-2] \tag{3}$$
$$k_0[t], k_1[t], k_2[t] = K_0 x[t], K_1 h[t-1], K_2 h[t-2] \tag{4}$$
$$v_0[t], v_1[t], v_2[t] = V_0 x[t], V_1 h[t-1], V_2 h[t-2] \tag{5}$$
$$w_i[t] = q_i[t]^\top k_i[t] \tag{6}$$
$$a[t] = \text{softargmax}([w_0[t], w_1[t], w_2[t]]) \tag{7}$$
$$h[t] = \sum_{i=0}^{2} a_i[t] v_i[t] \tag{8}$$

Where $x[t], h[t] \in \mathbb{R}^n$, and $Q_i, K_i, V_i \in \mathbb{R}^{n \times n}$. We define $h[t] = \mathbf{0}$ for $t < 1$. You may safely ignore these bases cases in the following questions.

(a) (1 pt) What is the dimension of $a[t]$?

(b) (3 pts) Extend this to, AttentionRNN(k), a network that uses the last k state vectors h. Write out the system of equations that defines it. You may use set notation or ellipses (...) in your definition.

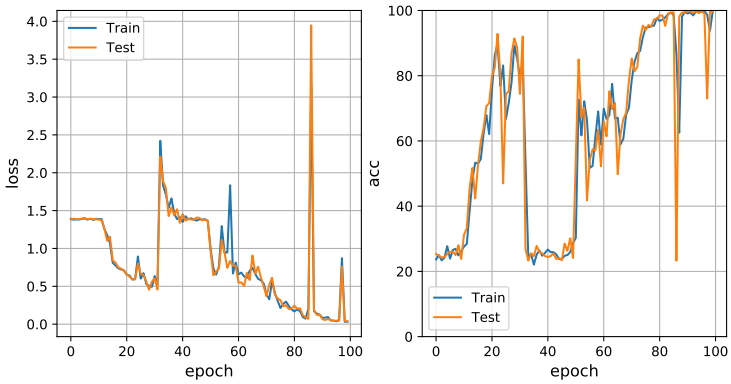
(c) (3 pts) Modify the above network to produce AttentionRNN(∞), a network that uses every past state vector. Write out the system of equations that defines it. You may use set notation or ellipses (...) in your definition. How many weight matrices are required for this model? Is there a way to limit this number? HINT: We can do this by tying together some set of parameters, e.g. weight sharing.

(d) (5 pts) Suppose the loss ℓ is computed. Please write down the expression for $\frac{\partial h[t]}{\partial h[t-1]}$ for AttentionRNN(2).

(e) (2 pts) Suppose we know the derivative $\frac{\partial h[t]}{\partial h[T]}$, and $\frac{\partial \ell}{\partial h[t]}$ for all $t > T$. Please write down the expression for $\frac{\partial \ell}{\partial h[T]}$ for AttentionRNN(k).

1.3 Debugging loss curves (5pts)

When we run the notebook *seq_classification.ipynb* (included in hw, and we also went through this code in class) we saw RNN training curves. In Section 8 "Visualize LSTM", we observed some "kinks" in the loss curve.



(a) (2pts) What caused the spikes on the left?

(b) (1pts) How can they be higher than the initial value of the loss?

(c) (1pts) What are some ways to fix them?

(d) (1pts) Explain why the loss and accuracy starts at that certain number in the start of training. You may need to check the task definition in the notebook.

2 Implementation (50pts)

There are three notebooks in the practical part:

- (25pts) Convolutional Neural Networks notebook: *hw2_cnn.ipynb*
- (20pts) Recurrent Neural Networks notebook: *hw2_rnn.ipynb*
- (5pts) : This builds on Section 1.3 of the theoretical part.
  - Change the model training procedure of Section 8 in *seq_classification.ipynb* to make the training curves have no spikes. You should only change the training of the model, and not the model itself or the random seed.

We encourage you to upload the notebooks to Google Colab to work on it.

First two notebooks contain parts marked as `TODO`, where you should put your code. After you finish, you should download your solutions, and then include them as a part of your submission zip.