Deep learning HW 2

wbg231

September 2023

1 theroy

1.1 Convolutional Neural Netoworks

(a) (1pt) Given an input image of dimension 11 × 19, what will be output dimension after applying a convolution with 5×4 kernel, stride of 4, and no padding

- we can use the formula use the formula

$$out = \frac{H - D * (k-1) + 2(p) - 1}{s} + 1$$

- so in this case our output would be

$$H_output = \frac{11 - 1 * (5-1) + 2(0) - 1}{4} + 1 = \frac{11-5}{4} + 1 = 10/4$$

  and we round down to 2

- then we can get

$$W_out = \frac{19 - 1 * (4-1) + 2(0) - 1}{4} + 1 = 4.75$$

  and we round down to 4.

- so our output would be (1,2,4)

(b) Given an input of dimension C × H ×W, what will be the dimension of the output of a convolutional layer with kernel of size K × K, padding P, stride S, dilation D, and F filters. Assume that H ≥ K, W ≥ K

- our formula for the height and with would be

$$H_{out} = floor(\frac{H + 2 * P - D * (K-1) - 1}{S} + 1)$$

  and

$$w_{out} = floor(\frac{W + 2 * P - D * (K-1) - 1}{S} + 1)$$

- then we would have to do that for each of the F filters yielding an output of (F,Hout,Wout)

(c) Let's consider an input x[n] $\in \mathbb{R}^5$

, with $1 \leq n \leq 7$, e.g. it is a length 7 sequence with 5 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, $W \in \mathbb{R}^{1\times5\times3}$, there's no bias and no non-linearity.

i. (1pt)What is the dimension of the output $f_W(x)$? Provide an expression for the value of elements of the convolutional layer output $f_W(x)$

- we can use the same formula to determine that

$$f_W(x) \in \mathbb{R}^{1\times1\times3}$$

- further we can write

$$f_w(x)_{i,j,k} = sn(i) = \sum_{i=1}^{3} x[2(n-1)+i]^T W[1,:,i]$$

ii. what is the dimension of $\frac{\partial \mathbf{f_W(x)}}{\partial \mathbf{W}}$? Provide an expression for the value of elements of it.

- sorry that looks kind of wierd i typed this question in another document and only had the pdf so i just included screenshots from it. the cut off name is wbg231 (my net id) wbg231

September 2023

1 question 1c.2

  - ok so we want to find $\frac{\partial \mathbf{f_w(x)}}{\partial \mathbf{W}}$
  - we can write this as $\frac{\partial \mathbf{f_w(x)}}{\partial \mathbf{W}} = \begin{pmatrix} \frac{\partial f_W(x)1}{\partial \mathbf{W}} \\ \frac{\partial f_W(x)2}{\partial \mathbf{W}} \\ \frac{\partial f_W(x)3}{\partial \mathbf{W}} \end{pmatrix}$
  - now we want to think about what $\frac{\partial f_w(x)_n}{\partial \mathbf{W}}$ is for $n \in [1,3]$
  - well we know that $f_w(x)_(n) = \sum_{i=1}^{3} x[2(n-1)+i]^t W[1,:,i] = x[2(n-1)+1]^t W[1,:,1] + x[2(n-1)+2]^t W[1,:,2] + x[2(n-1)+3]^t W[1,:,3]$
  - thus we can see that $\frac{\partial f_w(x)_n}{\partial \mathbf{W}} = \left(x[2(n-1)+1]^t, x[2(n-1)+2]^t, x[2(n-1)+3]^t\right)$
  - as we know each $x[i]^t$ is a row vector in $\mathbb{R}^{1\times 5}$ and we know that $n \in [1,3]$ we can see that $\frac{\partial f_w(x)_n}{\partial \mathbf{W}} \in \mathbb{R}^{3\times 3\times 5}$
  - and finally we can write $\frac{\partial \mathbf{f_w(x)}}{\partial \mathbf{W}}_{n,i,j} = x[2(n-1)+i]_k^T$
-

iii. what is the dimension of $\frac{\partial \mathbf{f_W(x)}}{\partial \mathbf{x}}$? Provide an expression for the value of elements of it.

- we can solve for the partial as

$$\frac{\partial \mathbf{f_W(x)}}{\partial \mathbf{x}} = ([\frac{\partial f_W(x)_{1,1,1}}{\partial \mathbf{x}}, \frac{\partial f_W(x)_{1,1,2}}{\partial \mathbf{x}}, \frac{\partial f_W(x)_{1,1,3}}{\partial \mathbf{x}}]) \in \mathbb{R}^{1\times 1\times 3}$$

- then we just need to check all of those
- so we can write

$$\frac{\partial f_W(x)_{1,1,n}}{\partial \mathbf{x}} = \frac{\partial}{\mathbf{X}} \sum_{i=1}^{3} x[2(n-1)+i]^T W[1,:,i] = (0, 0..0, W[1,:,1], W[1,:,2], W[1,:,3], 0..,0) \in \mathbb{R}^1$$

(that is a vector with 0 vectors everywhere and W in hte $(2(n-1)+1, 2(n-1)+2$ and $2(n-1)+3))$ spaces

- so its total dimensionility is

$$\frac{\partial \mathbf{F_w(x)}}{\partial \mathbf{x}} \in \mathbb{R}^{(3)\times(7\times 5)}$$

- and

$$\frac{\partial \mathbf{F_w(x)}}{\partial \mathbf{x}}_{n,i,:} = \begin{cases} W[1,1,:] & \text{if } i = 2(n-1)+1 \\ W[1,2,:] & \text{if } i = 2(n-1)+2 \\ W[1,3,:] & \text{if } i = 2(n-1)+3 \\ 0 & \text{otherwise} \end{cases}$$

iv. if we are given $\frac{\partial \ell}{\partial \mathbf{f_W(x)}}$ what is $\frac{\partial \ell}{\partial \mathbf{W}}$, what is its dimensionility.

2 question 1c.IV

- we can see that $\frac{\partial \ell}{\partial \mathbf{f_w(x)}} \in \mathbb{R}^{1\times 1\times 3}$
- and we know from part 2 that $\frac{\partial \mathbf{f_w(x)}}{\partial \mathbf{W}} \in \mathbb{R}^{3\times 3\times 5}$
- thus we can see that $\frac{\partial \ell}{\partial \mathbf{W}} = \frac{\partial \ell}{\partial \mathbf{f_w(x)}} \frac{\partial \mathbf{f_W(x)}}{\partial \mathbf{W}} \in \mathbb{R}^{3\times 1\times 5}$

-

- we can write this out as

$$\frac{\partial \ell}{\partial \mathbf{W}} = \frac{\partial \ell}{\partial \mathbf{f_w(x)}} \frac{\partial \mathbf{f_W(x)}}{\partial \mathbf{W}} = \begin{pmatrix} \begin{pmatrix} \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,1} \\ \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,2} \\ \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,3} \end{pmatrix} \end{pmatrix} \begin{pmatrix} \begin{pmatrix} X[1]^T \\ X[2]^T \\ X[3]^T \end{pmatrix} \\ \begin{pmatrix} X[3]^T \\ X[4]^T \\ X[5]^T \end{pmatrix} \\ \begin{pmatrix} X[5]^T \\ X[6]^T \\ X[7]^T \end{pmatrix} \end{pmatrix}$$

$$= \begin{pmatrix} \left(\frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,1} X[1]^T + \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,2} X[2]^T + \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,3} X[3]^T \right) \\ \left(\frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,1} X[3]^T + \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,2} X[4]^T + \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,3} X[5]^T \right) \\ \left(\frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,1} X[5]^T + \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,2} X[6]^T + \frac{\partial \ell}{\partial \mathbf{f_w(x)}}_{1,1,3} X[7]^T \right) \end{pmatrix}$$

$$= x * \frac{\partial \ell}{\partial \mathbf{f_w(x)}}$$

- this is similar to what we saw in part 1 of this question as this is a convolution of stride 2 with kernel size 3, with X
- however in this is a convolution of stride 2 with kernel size 3, with $\frac{\partial \ell}{\partial \mathbf{f_w(x)}}$ instead of the weight matrix W

-

1.2 Recurant Neural Netoworks

1.2.1 part 1

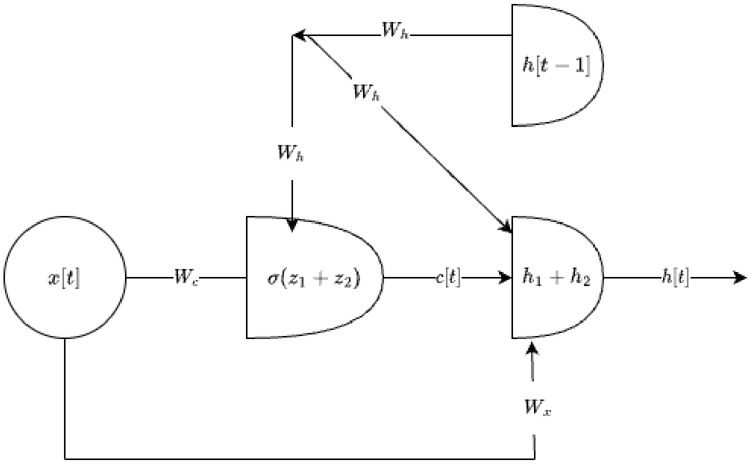
In this section we consider a simple recurrent neural network defined as follows:

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

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

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) Draw a diagram for this recurrent neural network, similar to the diagram of RNN we had in class



$$z_1 = W_c x[t]$$

$$z_2 = W_h h[t-1]$$

$$h_1 = c[t] \odot h[t-1]$$

$$h_2 = (1 - c[t]) \odot W_x x[t]$$

-

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

- we can see that $(W_c X[t]) \in \mathbb{R}^{m\times 1}$
- and $W_h h[t-1] \in \mathbb{R}^{m\times 1}$
- and we know that vector addition and the element wise sigmoid preserve dimensionility thus $c[t] \in \mathbb{R}^{m\times 1}$

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

- we can see that $\ell \in \mathbb{R}, h[t] \in \mathbb{R}^{m \times 1} \Rightarrow \frac{\partial \ell}{\partial \mathbf{h[t]}} \in \mathbb{R}^{1 \times m}$
- furhter we see earlier that $\frac{\partial \mathbf{h[t]}}{\partial \mathbf{W_x}} \in \mathbb{R}^{m \times m \times m}$
- so we know that our $\frac{\partial \ell}{\partial \mathbf{W_x}} = \frac{\partial \ell}{\partial \mathbf{h[t]}} \frac{\partial \mathbf{h[t]}}{\partial \mathbf{W_x}} \in \mathbb{R}^{m \times m}$
-
- so now we can solve $\frac{\partial \mathbf{c[t]}}{\mathbf{W_x}} = \frac{\partial}{\partial \mathbf{W_x}}(\sigma(W_c X[t] + W_h h[t-1]))$ we can call $z = W_c X[t] + W_h h[t-1]$
- and then we get $\frac{\partial \mathbf{c[t]}}{\mathbf{W_x}} = \frac{\partial}{\partial \mathbf{W_x}}(\sigma(z(W_x))) = diag(\sigma'(z))h[t]\frac{\partial \mathbf{h[t-1]}}{\partial \mathbf{W^x}} = d_c \in \mathbb{R}^{m \times m \times m}$
- let us also call $\frac{\partial \mathbf{h[t-1]}}{\partial \mathbf{w_x}} = d_{h[t-1]}$
- and $\frac{\partial \mathbf{W_x x[t]}}{\partial \mathbf{W_x}}_{i,j,k} = \begin{cases} x[t]_i & \text{if } i = j = k \\ 0 & \text{otherwise} \end{cases} \in \mathbb{R}^{M \times M \times M} = d_{x[t]}$
- now we can write $\frac{\partial \mathbf{h[t]}}{\partial \mathbf{W_x}} = \frac{\partial}{\partial \mathbf{W_x}}(c[t] \odot h[t-1] + (1 - c[t]) \odot W_x x[t])$

  $$= h[t-1]^t d_c + c[t]^t d_{h[t-1]} - (W_x x[t])^T d_c + (1 - c[t])^T d_{x[t]}$$

- so finally we have

  $$\frac{\partial \ell}{\partial \mathbf{W_x}} = \frac{\partial \ell}{\partial \mathbf{h[t]}} \frac{\partial \mathbf{h[t]}}{\partial \mathbf{W_x}} = \frac{\partial \ell}{\partial W_x}(h[t-1]^t d_c + c[t]^t d_{h[t-1]} - (W_x x[t])^T d_c + (1-c[t])^T d_{x[t]})$$

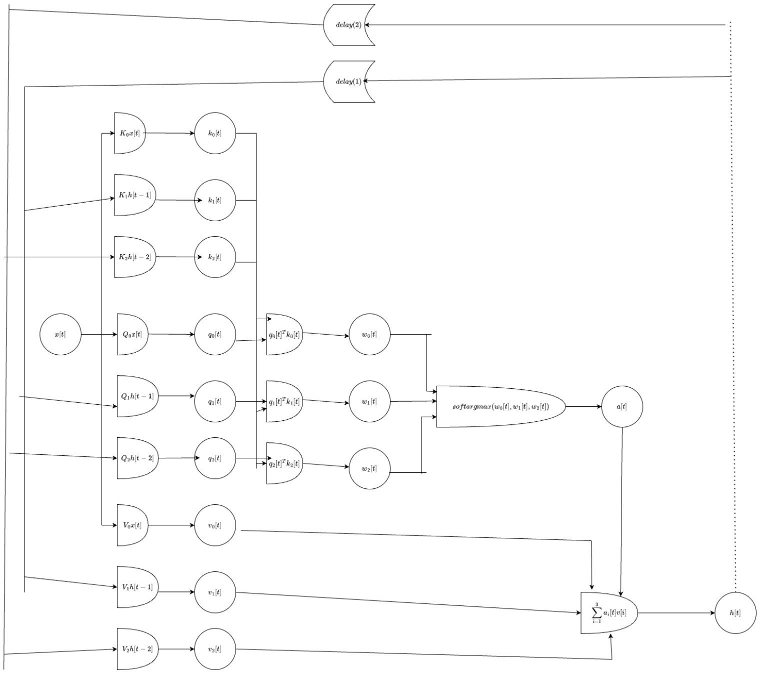
- this is similar to the forward pass in that there is still a recurrent relationship
- and we are also integrating both $c[t], h[t]$ and $h[t-1]$ into our predictions

(d) Can our gradient vanish or explode in this network?

- yes this network can be subject to exploding gradients as our sigmoid activation function will constantly scale our inputs to be between 0 and 1 so after many itterations our gradinets could get quite small and vanish

1.2.2 part 2

(a) draw the network diagram

- 

(b) what is the dimension of $a[t]$?

- $a[t] \in \mathbb{R}^3$

(c) 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.

- $\forall i \in [0, K]$
- $\begin{cases} q_i[t] = Q_0x[t] \text{if i=0} \\ q_i[t] = Q_ih[t-i] \text{if } i > 0 \end{cases}$
- $\begin{cases} k_i[t] = K_0x[t] \text{if i=0} \\ k_i[t] = K_ih[t-i] \text{if } i > 0 \end{cases}$
- $\begin{cases} v_i[t] = V_0x[t] \text{if i=0} \\ v_i[t] = V_ih[t-i] \text{if } i > 0 \end{cases}$
- $w_t[t] = q_i^T[t]k_[i][t] \quad \forall i \in [0, k]$
- $a[t] = \text{softmax}(\{w_i[t]\}_{i=1}^{k})$
- $h[t] = \sum_{i=1}^{t} a_i[t]v_i[t]$

(d) 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. HINT: We can do this by tying together some set of parameters, e.g. weight sharing.

- $q_0[t], q_1[t] \cdots q_{m-1}[t] = Q_0 x[t], Q_1 h[t-1] \cdots Q_m h[0]$
- $k_0[t], k_1[t] \cdots k_{m-1}[t] = K_0 x[t], K_1 h[t-1] \cdots K_m h[0]$
- $v_0[t], v_1[t] \cdots v_{m-1}[t] = V_0 x[t], V_1 h[t-1] \cdots V_m h[0]$
- $w_i[t] = q_i[t]^T k_i[t]$
- a[t]=softmax($\{w_i[t]\}_{i=1}^m$)
- h[t]=$\sum_{i=0}^m a_i[t] v_i[t]$

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

- so we know that $\frac{\partial \mathbf{h[t]}}{\partial \mathbf{h[t-1]}} = \frac{\partial}{\partial h[t-1]}(\sum_{i=1}^2 a_i[t] v_i[t])$
- so now we need $\frac{\partial \mathbf{a_i[t] v_i[t]}}{\partial \mathbf{h[t-1]}} = a_i[t] \frac{\partial \mathbf{v_i[t]}}{\partial \mathbf{h[t-1]}} + \frac{\partial a_i[t]}{\partial \mathbf{h[t-1]}} v_i[t]$
- ok so now we can check $\frac{\partial a_i[t]}{\partial \mathbf{h_i[t]}} = softargmax(w_0, w_1, w_2) = \frac{\partial softargmax}{\partial w} \odot (\frac{\partial w_0}{\partial \mathbf{h[t-1]}} + \frac{\partial w_1}{\partial \mathbf{h[t-1]}} + \frac{\partial w_2}{\partial \mathbf{h[t-1]}})$
- we can see $d_0 = \frac{\partial w_0}{\partial h[t-1]} = \frac{\partial}{\partial h[t-1]}(Q_0 x[t])^T K_0 x[t] = 0$
- so we can see that $d_1 = \frac{\partial w_1}{\partial \mathbf{h[t-1]}} = (h[t-1]^T K_1 Q_1^T + h[t-1]^T Q_1 K_1)$
- further wen see that $d_2 = \frac{\partial w_2}{\partial \mathbf{h[t-1]}} = \frac{\partial}{\partial \mathbf{h[t-1]}}(q_2[t]^t h_2[t]) = h[t-2]^t K_2 \frac{\partial h[t-2]}{\partial h[t-1]}^t Q_2^t + h[t-2]^t Q_2 (\frac{\partial h[t-2]}{\partial h[t-1]})^t K_2^t$
- thus we have

$$v = \frac{\partial a_i[t]}{\partial \mathbf{h_i[t]}} = softargmax(w_0, w_1, w_2) = \frac{\partial softargmax}{\partial w} \odot (\frac{\partial w_0}{\partial \mathbf{h[t-1]}} + \frac{\partial w_1}{\partial \mathbf{h[t-1]}} + \frac{\partial w_2}{\partial \mathbf{h[t-1]}})$$

$$= \frac{\partial softargmax(w)}{\partial w} \odot (d_1 + d_2)$$

- we can also see that $e_0 = \frac{\partial \mathbf{v_0}}{\partial \mathbf{h[t-1]}} = 0$
- $e_1 = \frac{\partial \mathbf{v_1}}{\partial \mathbf{h[t-1]}} = V_1^T$
- $e_2 = \frac{\partial \mathbf{v_2}}{\partial \mathbf{h[t-1]}} = \frac{\partial \mathbf{h[t-2]}}{\partial \mathbf{h[t-2]}} V_2^T$
- and thus we finally have $e = \frac{\partial v_0}{\partial \mathbf{h[t-1]}} + \frac{\partial v_1}{\partial \mathbf{h[t-1]}} + \frac{\partial v_0}{\partial \mathbf{h[t-1]}} = e_1 + e_2$
- so in total we have

$$\frac{\partial \mathbf{h[t]}}{\partial \mathbf{h[t-1]}} =$$

$$\frac{\partial}{\partial h[t-1]}(\sum_{i=1}^2 a_i[t] v_i[t]) = \sum_{i=1}^2 a_i[t] \frac{\partial \mathbf{v_i[t]}}{\partial \mathbf{h[t-1]}} + \frac{\partial a_i[t]}{\partial \mathbf{h[t-1]}} v_i[t] = \sum_{i=1}^2 a_i[t] e_i[t] + v_i[t] d_i[t]$$

(f) what is the formula for $\frac{\partial \ell}{h[T]}$ with AttentionRNN(K)

- in AttentionRNN(k) we can write $\frac{\partial h[t]}{\partial h[t-1]} = \sum_{i=1}^k a_i[t] e_[i][t] (\frac{\partial \mathbf{h(t-i)}}{\partial \mathbf{h[t-1]}}) + v_i[t] d_i[t] (\frac{\partial \mathbf{h(t-i)}}{\partial \mathbf{h[t-1]}})$
- thus we can see that $\frac{\partial \ell}{\partial \mathbf{h[T]}} = \sum_{i \in [k+1,t]} \frac{\partial \ell}{\partial \mathbf{h[i]}} + \frac{\partial \mathbf{h[T+1]}}{\partial \mathbf{H[T]}} = \sum_{i \in [k+1,t]} \frac{\partial \ell}{\partial \mathbf{h[i]}} + \sum_{i=1}^K a_i[t] e_i[t] (\frac{\mathbf{h[T-i]}}{\partial \mathbf{h[T]}}) + v_i[t] d_i[t] \frac{\partial \mathbf{h[T-i]}}{\partial \mathbf{h[T]}}$

1.3 Debugging loss curves

(a) what is the cause of the spikes on the left graph

-
- the left spikes are areas where despite increasing epochs there are large spikes in model error for an rnn model. This is likely due to the exploding gradient problem where the gradients are so large that they cause the model to diverge and thus the loss to increase. that is if our gradient is really high using GD even with a small learing rate we will move very far away forom our prevous optima

(b) how can they be higher than the inital value

- effectivly this is becuase the weights were intialized to be very small and random so the that erorr is pretty much just guassing evenly, here we are following the exploding gradinet in a way that we are doing worse than guessing

(c) what are some ways to fix them

- we can use gradient clipping to prevent the gradients from getting too large
- we can also use a smaller learning rate to prevent the gradients from getting too large
- we can use an LSTM model which is less prone to exploding gradients

(d) Explain why the loss and accuracy are at these values before training starts. You may need to check the task definition in the notebook.

- there are 4 possibilities Q,R,S,U so we are guessing rand omly with $\frac{1}{4}$ then taking the natural log of that we get$log(4) = 1.3$