Homework 2

Guilherme Albertini

October 11, 2022

Theory

Problem 1.1: Convolutional Neural Networks

1. Given an input image of dimension 21×12, what will be output dimension after applying a convolution with 4×5 kernel, stride of 4, and no padding?

   5×2

2. 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 $K \times K$, padding P, stride S, dilation D, and F filters. Assume that H $\geq$ K, W $\geq$ K.

Define Padding along height on top P_{H1}

Define Padding along height on bottom P_{H2}

Define Padding along width on left P_{W1}

Define Padding along width on right P_{W2}

Define Kernel width K_H

Define Kernel height K_W

Define Stride horizontal S_W

Define Stride vertical S_H

Define Batch Count B

Note that for Dilated kernel:

$$K' = K + (K-1)(D-1) = K + KD - K - D + 1 = D(K-1) + 1$$

.

Effect of adding padding and applying kernel to dimensions:

$$\begin{aligned}
H_P &= P_{H1} + P_{H2} + H \\
W_P &= P_{W1} + P_{W2} + W \\
H_{PK} &= H_1 - [D_H(K_H - 1) + 1] \\
&= P_{H1} + P_{H2} + H - [D_H(K_H - 1) + 1] \\
W_{PK} &= W_1 - [D_W(K_W - 1) + 1] \\
&= P_{W1} + P_{W2} + W - [D_W(K_W - 1) + 1]
\end{aligned}$$

Considering stride to dimensions:

$$\begin{aligned}
H_{PKS} &= \left\lfloor \frac{H_P - [D_H(K_H - 1) + 1] + S_H}{S_H} \right\rfloor \\
&= \left\lfloor \frac{P_{H1} + P_{H2} + H - [D_H(K_H - 1) + 1]}{S_H} \right\rfloor + 1 \\
W_{PKS} &= \left\lfloor \frac{W_P - [D_W(K_W - 1) + 1] + S_W}{S_W} \right\rfloor \\
&= \left\lfloor \frac{P_{W1} + P_{W2} + W - [D_W(K_W - 1) + 1]}{S_W} \right\rfloor + 1
\end{aligned}$$

We can make simplifcations that I think are implied here:

$$\begin{aligned}
S &= S_W = S_H \\
D &= D_W = D_H \\
K &= K_W = K_H \\
B &= 1 \\
P &= P_{W1} + P_{W2} = P_{H1} + P_{H2}
\end{aligned}$$

Thus the output dimension is:

$$\begin{aligned}
F \times & \left(\left\lfloor \frac{2P + H - [D(K-1) + 1]}{S} \right\rfloor + 1 \right) \\
\times & \left(\left\lfloor \frac{2P + W - [D(K-1) + 1]}{S} \right\rfloor + 1 \right)
\end{aligned}$$

3. 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 $W, W \in \mathbb{R}^{1 \times 5 \times 3}$ and there is no bias and no non-linearity.

   (a) 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: $f_W(x) \in \mathbb{R}^{42 \times 42 \times 42}$, $f_W(x)[i, j, k] = 42$.

The general recurrence equation (which is first-order, non-homogeneous, with variable coefficients): $r_{l-1} = s_l r_l - (s_l - k_l) = s_l(r_l - 1) + k_l$

$$f_W(x) \in \mathbb{R}^3$$
$$f_W(x)[r] = \sum_{c=1}^{5}\sum_{k=1}^{3} x[k + 2(r-1), c]W_{1,c,k}$$

For $r = \{i : i \in \mathbb{N}, i \in [1, \dim(f_W)]\}$

(b) What is the dimension of $\frac{\partial f_W(x)}{\partial W}$? What are its values?

Note: There are a few ways one could interpret the transpose of the tensor (W) depending on which dimensions are to be transposed in numerator format. Using a chosen transpose with numerator layout format.

$$\frac{\partial f_W(x)}{\partial W} \in \mathbb{R}^{3\times(3\times5\times1)}$$
$$\frac{\partial f_W(x)}{\partial W}[r, c, k] = x[k + 2(r-1), c]$$

(c) What is the dimension of $\frac{\partial f_W(x)}{\partial x}$? What are its values?

See note above.

$$\frac{\partial f_W(x)}{\partial x} \in \mathbb{R}^{3\times(7\times5)}$$
$$\frac{\partial f_W(x)}{\partial x}[r, c, k] = \begin{cases} W_{1,c,k-2(r-1)} & \text{if } k - 2(r-1) \in [1, 3] \\ 0 & \text{otherwise} \end{cases}$$

(d) Now, suppose you are given the gradient of the loss ℓ with respect to 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 its expression. Explain the similarities and differences of this and expression in (a).

$$
\begin{aligned}
\frac{\partial \ell}{\partial W} &= \frac{\partial \ell}{\partial f_W}\frac{\partial f_W}{\partial W} \\
\frac{\partial \ell}{\partial W} &\in \mathbb{R}^{3\times 5\times 1}
\end{aligned}
$$

$$
\left(\frac{\partial \ell}{\partial W}\right)[1,c,k] = \sum_{r=1}^{3}\left(\frac{\partial \ell}{\partial f_W(x)}\right)[r]x[k+2(r-1),c]
$$

Both the backward and forward pass of the convolutional layer apply a convolution but the stride dilates in the backward pass; we can consider dilation factor D as the gradient of the loss with respect to the output of the convolutional layer.

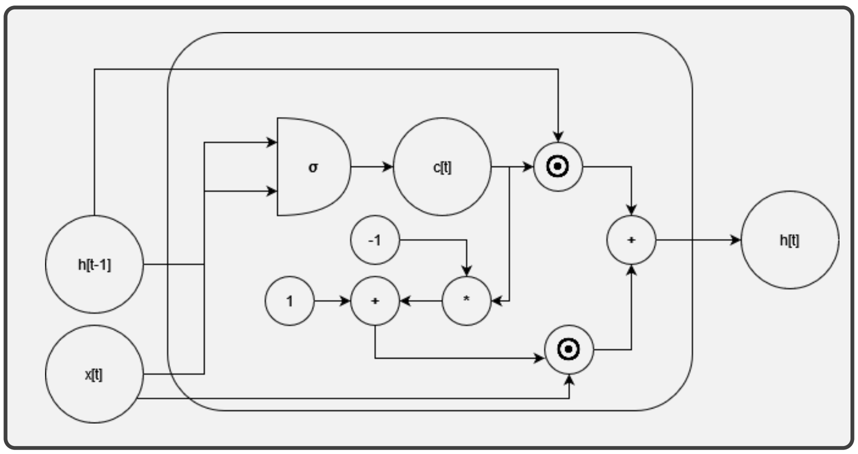
Problem 1.2: Recurrent Neural Networks

In this section consider simple recurrent neural network defined by:

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

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

1. Draw a diagram for this RNN.



2. What is the dimension of $c[t]$?

$$c[t] \in \mathbb{R}^m$$

3. 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 the dimension of an expression for values of $\frac{\partial \ell}{\partial W_x}$. What are the similarities and differences between backward and forward pass of RNN?

$$\frac{\partial \ell}{\partial W_x} = \sum_{t=1}^{K} \frac{\partial \ell}{\partial h[t]} \frac{\partial h[t]}{\partial W_x}$$

Note that the first term of $\frac{\partial h[t]}{\partial W_x}$ has dependence on the prior term recursively so need chain rule: $\frac{\partial h[t]}{\partial h[t-1]} \frac{h[t-1]}{W_x}$

$$= \sum_{t=1}^{K} \frac{\partial \ell}{\partial h[t]} \left(\frac{\partial([1-c[t]] \odot W_x x[t])}{\partial W_x} + \sum_{i=1}^{t-1} \left(\frac{\partial h[i]}{\partial h[i-1]} \frac{\partial h[i-1]}{\partial W_x} \right) \right)$$
$$= \sum_{t=1}^{K} \frac{\partial \ell}{\partial h[t]} \left(\frac{\partial([1-c[t]] \odot W_x x[t])}{\partial W_x} + \sum_{i=1}^{t-1} \left(\frac{\partial h[i+1]}{\partial h[i]} \frac{\partial h[i]}{\partial W_x} \right) \right)$$

Where the last step was done to account for the adjustment at undefined behavior for partials of $h[0]$ and keep at most $K-1$ recursive sums. Note:

$$\begin{aligned} \frac{\partial([1-c[t]] \odot W_x x[t])}{W_x} &= \frac{diag(1-c[t])\partial(W_x x[t]) + diag(W_x x[t])\partial(1-c[t])}{\partial W_x} \\ &= diag(1-c[t]) \frac{\partial W_x x[t]}{\partial W_x} \\ &= (1-c[t]) \odot X \end{aligned}$$

Where $X \in \mathbb{R}^{m \times (m \times n)}$ and column j is not zero: $X_j = [0 \times (j-1), x[t], 0, \ldots]$

$$\frac{\partial \ell}{\partial W_x} = \sum_{t=1}^{K} \frac{\partial \ell}{\partial h[t]} \left((1-c[t]) \odot X + \sum_{i=1}^{t-1} \left(\prod_{j=i}^{t-2} \frac{\partial h[i+1]}{\partial h[i]} \right) [(1-c[t]) \odot X] \right)$$

$$\frac{\partial h[i+1]}{h[i]} = diag(c[t]) + diag(h[t-1]) \frac{\partial c[t]}{\partial h[t-1]} - diag(W_x x[t]) \frac{\partial c}{\partial h[t-1]}$$

$$\frac{\partial c[t]}{\partial h[t-1]} = W_h diag(\sigma(W_c x[t] + W_h h[t-1])) \odot (1 - \sigma(W_c x[t] + W_h h[t-1]))$$

4. Can this network be subject to vanishing or exploding gradients?

The vector $h[t]$ is not being multiplied by matrices throughout timesteps so will not have exploding gradients. It can vanishing gradients as the element-wise multiplication of values of $h[t]$ and $c[t]$ are between 0 and 1.

Problem 1.3: AttentionRNN(2)

Now define AttentionRNN(2) as:

$$q_0[t], q_1[t], q_2[t] = Q_0 x[t], Q_1 h[t-1], Q_2 h[t-2] \tag{3}$$

$$k_0[t], k_1[t], k_2[t] = K_0 x[t], K_1 h[t-1], K_2 h[t-2] \tag{4}$$

$$v_0[t], v_1[t], v_2[t] = V_0 x[t], V_1 h[t-1], V_2 h[t-2] \tag{5}$$

$$w_i[t] = q_i[t]^T k_i[t] \tag{6}$$

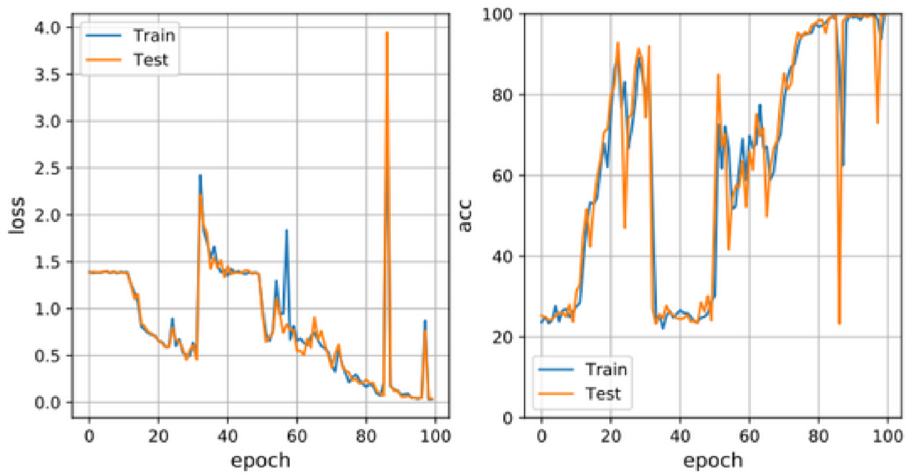
$$a[t] = \text{softargmax}([w_0[t], w_1[t], w_2[t]]) \tag{7}$$

$$h[t] = \sum_{i=0}^{2} a_i[t] v_i[t] \tag{8}$$

where $x_i[t], h[t] \in \mathbb{R}^n$ and $Q_i, K_i, V_i \in \mathbb{R}^{n \times n}$. We define $h[t] = 0$ for $t < 1$. You may safely ignore base cases in the following.

1. Draw a diagram for this RNN.
2. What is the dimension of $a[t]$?
3. Extend this to AttentionRNN(k), a netowrk that uses the last k state vectors h. Write out a system of equations that defines it.
4. Modify the above netork to produce AttentionRNN(∞), a network that uses every past state vector. Write out a system of equations that defines it.
5. Suppose the loss ℓ is computed, and we know the derivative $\frac{\partial \ell}{\partial h[i]}$ for all $i \geq t$. Write down expression for $\frac{\partial h[t]}{\partial h[t-1]}$ for AttentionRNN(2).
6. Suppose we know $\frac{\partial h[t]}{\partial h[T]}$ and $\frac{\partial \ell}{\partial h[t]} \forall t > T$. Write down expression for $\frac{\partial \ell}{\partial h[T]}$ for AttentionRNN(k).

Problem 1.4: Debugging Loss Curves



1. What causes the spikes on the left?
2. How can they be higher than the initial value of the loss?
3. What are some ways to fix them?
4. Explain why the loss and accuracy are at these set values before training starts. You may need to check the task definition in the notebook,