Deep Learning:
Homework 2: CNN and RNN

Due on 03/11/2021

Xinyi Zhao

Problem 1

1.1 Convolutional Neural Netoworks

(a) **Answer:**

$$\frac{10-(3-2)}{2} = 4.5 \approx 4$$
$$\frac{11-(3-2)}{2} = 5$$

So the output dimension is 4×5.

(b) **Answer:**
Assume the dimension of output is $C_1 \times H_1 \times W_1$.

$$C_1 = F$$
$$H_1 = \lfloor \frac{H + 2P - (K + (K-1)(D-1))}{S} + 1 \rfloor$$
$$W_1 = \lfloor \frac{W + 2P - (K + (K-1)(D-1))}{S} + 1 \rfloor$$

(c) **Answer:**

(i) The dimensions: $x \in \mathbb{R}^{5\times 2}$, $x[n] \in \mathbb{R}^2$, $W \in \mathbb{R}^{1\times 2\times 3}$.

$$\begin{aligned} s[n] = (x * k)[n] &= \sum_{m=-M}^{M} x[n+m] \cdot k[m] \\ &= \sum_{m=-M}^{M} x[n+m]^\top k[m] \\ &= \sum_{m=-M}^{M} \sum_{i=1}^{c} x[n+m][c]k[m][c] \end{aligned}$$

where $k[m] = W[1]^\top[m+2]$, $c = 2$.

Therefore, the dimension of the output $fw(x) \in \mathbb{R}^{2\times 1}$.

$$fw(x)[i,j] = \sum_{m=-1}^{1} \sum_{c=1}^{2} x[2i+m][c]W[1]^\top[m+2][c]$$
$$i \in \{1,2\}, j = 1$$

(ii)

$$\frac{\partial fw(x)}{\partial W} \in \mathbb{R}^{(2\times1)\times(1\times2\times3)}$$

$$\frac{\partial fw(x)}{\partial W}[i,j,k,l,m] = x[m+2(i-1)][l]$$

$$i \in \{1,2\}, j = 1, k = 1, l \in \{1,2\}, m \in \{1,2,3\}$$

(iii)

$$\frac{\partial fw(x)}{\partial x} \in \mathbb{R}^{(2\times1)\times(5\times2)}$$

$$\frac{\partial fw(x)}{\partial x}[i,j,k,l] = W[1]^\top[k-2(i-1)][l]$$

$$i \in \{1,2\}, j = 1, k \in \{1,2,3,4,5\}, l \in \{1,2\}$$

(iv)

$$\frac{\partial l}{\partial W} \in \mathbb{R}^{1\times2\times3}$$

$$\frac{\partial l}{\partial W}[k,l,m] = \frac{\partial l}{\partial fw(x)}[1][1]\, x[m][l] + \frac{\partial l}{\partial fw(x)}[1][2]\, x[m+2][l]$$

$$k = 1, l \in \{1,2\}, m \in \{1,2,3\}$$

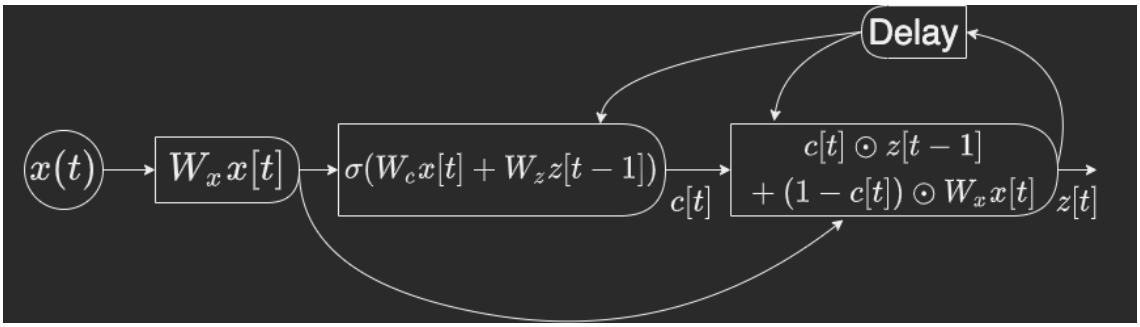
- Similarities: During the forward pass, we slide (convolve) the filter across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. The backward pass for a convolution operation (for the weights) is also a convolution of the input x.
- Differences: The backward pass is a convolution of the input x but with spatially-flipped filters.

Problem 2

1.2 Recurrent Neural Networks

(a) **Answer:**

$$c[t] = \sigma(W_c x[t] + W_z z[t-1])$$
$$z[t] = c[t] \odot z[t-1] + (1 - c[t]) W_x x[t]$$



(b) **Answer:**

The dimensions of $c[t]$: $c[t]$ has the same dimension as $z[t]$, so $c[t] \in \mathbb{R}^m$.

(c) **Answer:**

$$\frac{\partial l}{\partial W_x} \in \mathbb{R}^{n \times m}$$

$$\frac{\partial l}{\partial W_x} = \frac{\partial l}{\partial z[t]} [\sum_{k=1}^{t} (\prod_{i=k+1}^{t} \frac{\partial z[i]}{\partial z[i-1]}) \frac{\partial z[k]}{\partial W_x}]$$

The similarities of backward pass and forward pass in this RNN: These two passes both keep memory of previous states. The current state of time t is based on the previous states from 1 to $t-1$ in both passes.

(d) **Answer:**

$x[t]$ is the input vector, $z[t]$ is the output vector, $c[t]$ is the update gate vector, and W_x, W_c, W_z are learnable parameters.

$c[t]$ is a gating vector that determines how much of the past information should be passed along to the future. It applies a sigmoid function to the sum of two linear layers over the input $x[t]$ and the previous state $z[t-1]$. $c[t]$ contains coefficients between 0 and 1 as a result of applying sigmoid. The final output state $z[t]$ is a convex

combination of $z[t-1]$ and $W_x x[t]$. In a long sequence, the gradients get multiplied by the weight matrix at every time step. If we have a large weight matrix and the non-linearity in the recurrent layer is not saturating, $c[t]$ can be close to 1, the current unit output is just a copy of the previous state and ignores the input. This could avoid exploding gradients. But if there are small values in the weight matrix, the norm of gradients get smaller and smaller exponentially.

Therefore, this network cannot be subject to exploding gradients but it can be subject to vanishing gradients.