# DS-GA 1008 HW 3

Long Chen
Center for Data Science, New York University
lc3424@nyu.edu

Fall 2021

## 1 Theory

### 1.1 Convolutional Neural Network

**a)**

For width:

$$W_{output} = \frac{W - K + 2P}{S} + 1 = \frac{10 - 3}{2} + 1 = 4.5$$

Thus we round it down to get $W_{output} = 4$.

For hight:

$$H_{output} = \frac{H - K + 2P}{S} + 1 = \frac{11 - 3}{2} + 1 = 5$$

Therefore, the output dimension will be $4 \times 5$.

**b)**

For height:

$$H_{output} = \frac{H - D \times (K - 1) + 2P - 1}{S} + 1$$

For width:

$$W_{output} = \frac{W - D \times (K - 1) + 2P - 1}{S} + 1$$

Therefore the output dimension is $F \times H_{output} \times W_{output}$. If we encounter non-integer results in the calculation, round it down.

**c)**

**i)** Dimension of the output is:

$$f_w(x) \in \mathbb{R}^{1\times 2},$$

where the first dimension indicates the number of filters, second dimension indicates the number of channels, and third dimension indicates the size of output for each convolution.

Let's further define $W \in \mathbb{R}^{1\times 3}$, where the first dimension indicates the number of filters, second dimension indicates the number of channels, and third dimension indicates the kernel size.

Intuitively, the value of $f_w(x)[i,k]$ is the sum over channels of the dot products of $x' = x[j, 2k : 2k+3]$ (i.e. the slice of x in $jth$ channel with index equal to $2k, 2k+1, 2k+2$, where $k$ is 0-indexed), and $W' = W[i,j,:]$ (i.e. the weight for filter i in channel j):

$$f_w(x)[i,j,k] = (W')^T \times X' \in \mathbb{R}.$$

**ii)** Dimension of

$$\frac{\partial f_W(x)}{\partial W}$$

is $1 \times 2 \times 3$, where the first dimension indicates the output size of convolution for each channel, and the second dimension indicates the kernel size.

Let $[f_w(x)]_{jk}$ the partial output of convolution, where $j$ indicates $j$-th channel and $k$ indicates $k$-th output from convolution.

Let $W_j$ be the weight of $j$-th channel and $W_{jp}$ be the weight of $p$-th kernel index in the $j$-th channel.

Let $x_{jk}$ be the slice vector of $x$ in $j$-th channel such that index $i \in [2k, 2k+3] \cap \mathbb{Z}$.

Note that we are ignoring filter dimension because we only have 1 filter, though adding the additional dimension is trivial. With such setting:

$$\begin{aligned}[f_w(x)]_{jk} &= (W_j)^T \times x_{jk} \\ &= W_{j0} \times x_{jk}[0] + W_{j1} \times x_{jk}[1] + W_{j2} \times x_{jk}[2],\end{aligned}$$

where $x_{jk}[i]$ indicates the $i$-th element in the $x_{jk}$ vector. This is essentially the dot product between corresponding weight and data. Therefore, easily can we find its partial derivative $w.r.t\ W_j$:

$$\begin{aligned}\frac{\partial [f_w(x)]_{jk}}{\partial w_j} &= \begin{pmatrix} \frac{\partial [f_w(x)]_{jk}}{\partial w_{j0}} & \frac{\partial [f_w(x)]_{jk}}{\partial w_{j1}} & \frac{\partial [f_w(x)]_{jk}}{\partial w_{j2}} \end{pmatrix} \\ &= \begin{pmatrix} x_{jk}[0] & x_{jk}[1] & x_{jk}[2] \end{pmatrix}\end{aligned}$$

Therefore we could find all $\frac{\partial [f_w(x)]_{jk}}{\partial w_j}$ for $j = 1, \ldots, C$ and $k = 1, \ldots, O$, where $O$ indicates the output size of convolution, that form $\frac{\partial f_w(x)}{\partial w}$.

**iii)** The dimension is $2 \times 2 \times 5$. Using settings from ii), we could easily find that:

$$\frac{\partial [f_w(x)]_{jk}}{\partial x_j} = \left( \frac{\partial [f_w(x)]_{jk}}{\partial x_j[0]} \quad \frac{\partial [f_w(x)]_{jk}}{\partial x_j[1]} \quad \frac{\partial [f_w(x)]_{jk}}{\partial x_j[2]} \quad \frac{\partial [f_w(x)]_{jk}}{\partial x_j[3]} \quad \frac{\partial [f_w(x)]_{jk}}{\partial x_j[4]} \right)$$

Therefore we could find all $\frac{\partial [f_w(x)]_{jk}}{\partial x_{jk}}$ for $j = 1, \ldots, C$ and $k = 1, \ldots, O$, where $O$ indicates the output size of convolution, that form $\frac{\partial f_w(x)}{\partial x}$.

**iv)** The dimension of $\frac{\partial \ell}{\partial f_w(x)}$ is $2 \times 2$. The dimension of $\frac{\partial \ell}{\partial W}$ is $2 \times 3$, where the first dimension indicates channel, and the second dimension indicates kernel size. For $j = 1, \ldots, C$, $p = 1, \ldots, K$ ($K$ is the kernel size), and $k = 1, \ldots, O$ ($O$ is the output size of convolution). We thus have:

$$\frac{\partial \ell}{\partial W_{jp}} = \sum_{k=0}^{1} \frac{\partial \ell}{\partial [f_w(x)]_{jk}} \frac{\partial [f_w(x)]_{jk}}{W_{jp}}$$

## 1.2 Recurrent Neural Networks

### a) Diagram

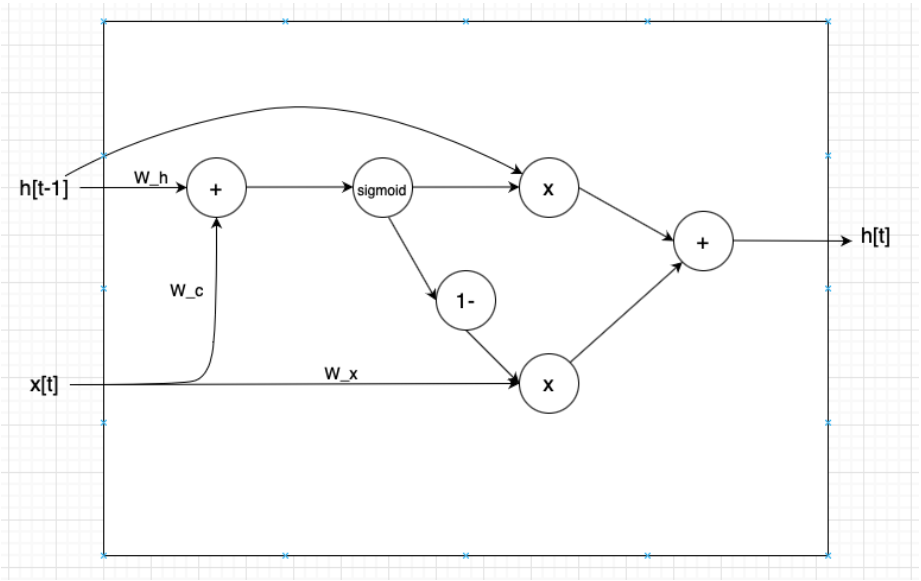


Figure 1: RNN visualization

## b)

$c[t] \in \mathbb{R}^m$, since $W_c x[t] \in \mathbb{R}^{m \times 1}$, $W_h h[t] \in \mathbb{R}^{m \times 1}$, and element-wise sigmoid does not change dimensions.

## c)

$\frac{\partial \ell}{\partial w_x} \in \mathbb{R}^{m \times n}$.

$$\frac{\partial \ell}{\partial w_x} = \sum_{t=1}^{k} \left[ \frac{\partial \ell}{\partial h[t]} \frac{\partial h[t]}{\partial w_x} \right]$$

Similarities: Need all timestamp $t$ for forward and backward.

## d)

Yes. Gradient at timestamp depends on all previous gradients and does not have long-term memories. Therefore, gradient could still vanish or explode.