CSE - 5526

Homework 5

Submitted by

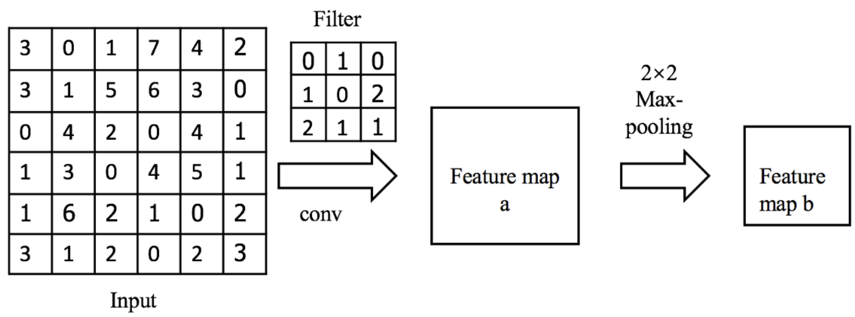
Utkarsh Pratap Singh Jadon (500711257)
jadon.1@osu.edu

Contents

1 Question 1

As shown below, a 2-D input with dimensions 6×6 is convolved with a 3×3 filter with stride 1 (i.e. shift by one "pixel") to obtain feature map a, which then undergoes 2×2 max pooling with stride 2 to get feature map b.

1. Give the values of feature maps a and b.
2. How many "pixels" do we need to pad around the borders of the input in order to maintain the size of the input image after convolutoin? With such zero padding, what are the values of the resulting feature maps a and b?



Solution

1. Here, the dimension of input, (m) is 6, dimension of the filter, (n) is 3, and stride (s) is 1. Therefore, dimension of the output of convolution, Feature Map a, is given by $\frac{m-n}{s} + 1 = \frac{6-3}{1} + 1 = 4$.

In the 4x4 Feature Map, the $(0, 0)^{th}$ entry is given by:
$(3 * 0) + (0 * 1) + (1 * 0) + (3 * 1) + (1 * 0) + (5 * 2) + (0 * 2) + (4 * 1) + (2 * 1) = 19$

Similarly, other values can be calculated, which generates Feature Map a as follows:

$$\begin{bmatrix} 19 & 24 & 26 & 15 \\ 10 & 19 & 25 & 19 \\ 15 & 28 & 15 & 14 \\ 17 & 12 & 12 & 15 \end{bmatrix}$$

Now, 2x2 Max-pooling is performed on Feature Map a to get Feature Map b. Similarly, dimensions of Feature Map b is given by $\frac{4-2}{2} + 1 = 2$.

Feature Map b is given as:

$$\begin{bmatrix} 24 & 26 \\ 28 & 15 \end{bmatrix}$$

2. To maintain the size of input after convolution, we need to zero-pad input with 2 rows and 2 columns, each consisting of 6 pixel values, thereby making input as 8 x 8. Therefore, total number of pixels to be added is: $(4 * 6) + 4 = 28$.

Now, dimension of the output of convolution, Feature Map a, is given by $\frac{m-n}{s} + 1$ $= \frac{8-3}{1} + 1 = 6$.

In the 6x6 Feature Map, the $(0, 0)^{th}$ entry is given by:
$(0*0) + (0*1) + (0*0) + (0*1) + (3*0) + (0*2) + (0*2) + (3*1) + (1*1) = 4$

Similarly, other values can be calculated, which generates Feature Map a as follows:

$$\begin{bmatrix} 4 & 17 & 27 & 28 & 26 & 10 \\ 9 & 19 & 24 & 26 & 15 & 14 \\ 15 & 10 & 19 & 25 & 19 & 15 \\ 13 & 15 & 28 & 15 & 14 & 8 \\ 17 & 17 & 12 & 12 & 15 & 8 \\ 3 & 13 & 3 & 7 & 6 & 4 \end{bmatrix}$$

Now, 2x2 Max-pooling is performed on Feature Map a to get Feature Map b. Similarly, dimensions of Feature Map b is given by $\frac{6-2}{2} + 1 = 3$.
Feature Map b is given as:

$$\begin{bmatrix} 19 & 28 & 26 \\ 15 & 28 & 19 \\ 17 & 12 & 15 \end{bmatrix}$$

2 Question 2

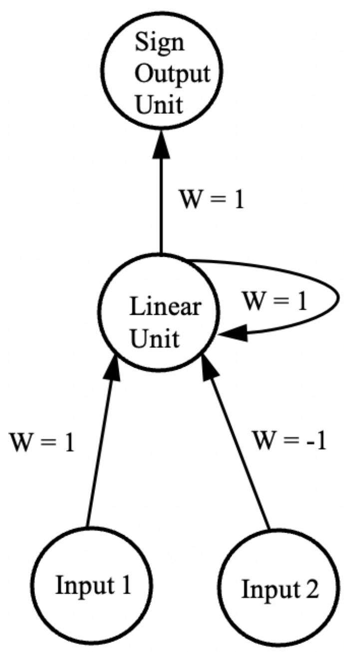
An RNN and its corresponding inputs at three time steps are given below. Recall that, for a linear unit, the output is the same as the input, and the sign activation function is

$$Sign(x) = \begin{cases} 1 & \text{if x} \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Answer the following questions:

1. Unfold the network and show the values of the hidden unit and the output unit at $t = 1, 2$, and 3.

2. Describe the computational task this RNN performs, i.e., what is this RNN intuitively doing. Justify your answer.



	$t = 1$	$t = 2$	$t = 3$
Input 1	2	1	0
Input 2	-2	5	1

Solution

1. The network shown in Figure 1 can be defined as:

$$a^{(t)} = Wa^{(t-1)} + Ux^{(t)}$$
$$h^{(t)} = sign(a^{(t)}$$
$$o^{(t)} = Vh^{(t)}$$

where W is the weight between two hidden layers, a is the output of linear unit, U is the weight between input and linear unit, x is the input, $sign$ is the activation function, h is the output of hidden unit, V is the weight between activation and output unit.

Here, $W = 1$, $U = [1, -1]$, $V = 1$

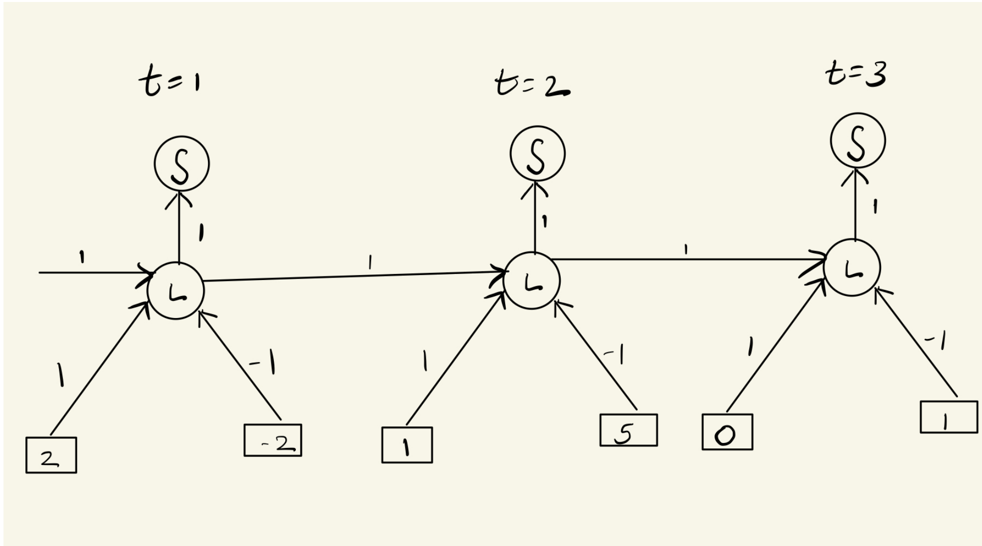


Figure 1: Unfolded Recurrent Neural Network

t = 1

Assuming $h^0 = 0$

$a^1 = Wh^0 + Ux^1 = 1 * 0 + [1, -1] * [2, -2]^T = 0 + 2 + 2 = 4$

$h^1 = sign(a^1) = sign(4) = 1$

$o^1 = Vh^1 = 1 * 1 = 1$

t = 2

$a^2 = Wa^1 + Ux^2 = 1 * 4 + [1, -1] * [1, 5]^T = 4 + 1 - 5 = 0$

$h^2 = sign(a^2) = sign(0) = 1$

$o^2 = Vh^2 = 1 * 1 = 1$

t = 3

$a^3 = Wa^2 + Ux^3 = 1 * 0 + [1, -1] * [0, 1]^T = 0 + 0 - 1 = -1$

$h^3 = sign(a^3) = sign(-1) = -1$

$o^3 = Vh^3 = 1 * -1 = -1$

2. The RNN network appears to be carrying out a classification task. According to the sign activation function that has been applied to the output of the linear unit, it specifically categorizes the inputs at each time step as either positive or negative. The RNN network processes inputs using a linear unit with a sign activation function at three different time steps. Depending on the input, the linear unit outputs a value which is fed to the activation function. Then, the sign activation function converts this output to either 1 or -1, which,

respectively, represents positive or negative.

Overall, the RNN network is intuitively performing a simple classification task. This task can be useful in a variety of applications, such as sentiment analysis in natural language processing or anomaly detection in time series data.