THU-70250043-0, Pattern Recognition (Spring 2021) **Homework: 9**

Deep Learning 2

Lecturer: Changshui Zhang zcs@mail.tsinghua.edu.cn

Hong Zhao vzhao@tsinghua.edu.cn

Student: Jingxuan Yang yangjx20@mails.tsinghua.edu.cn

Recurrent Neural Network (RNN)

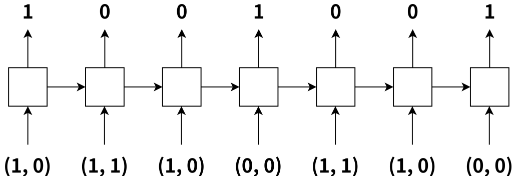
1. In this problem, you will implement a recurrent neural network which implements binary addition. The inputs are given as binary sequences, starting with the least significant binary digit. (It is easier to start from the least significant bit, just like how you did addition in grade school.) The sequences will be padded with at least one zero on the end. For instance, the problem

$$110111 + 10010 = 1001001 \tag{1}$$

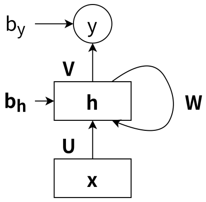
would be represented as:

- **Input 1**: 1, 1, 1, 0, 1, 1, 0
- **Input 2**: 0, 1, 0, 0, 1, 0, 0
- **Correct output**: 1, 0, 0, 1, 0, 0, 1

There are two input units corresponding to the two inputs, and one output unit. Therefore, the pattern of inputs and outputs for this example would be:



Design the weights and biases for an RNN which has two input units, three hidden units, and one output unit, which implements binary addition. All the units use the hard threshold activation function. In particular,



specify the values of weight matrices $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{W}$, bias vector $\boldsymbol{b_h}$, and scalar bias b_y. The details of the architecture and the computation are as follows:

$$\begin{aligned} \boldsymbol{h}^{(t)} &= \mathrm{hard}(\boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{b_h}) \\ y^{(t)} &= \mathrm{hard}(\boldsymbol{V}\boldsymbol{h}^{(t)} + b_y) \end{aligned} \tag{2}$$

$$\mathrm{hard}(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geqslant 0 \end{cases} \tag{3}$$

If $\boldsymbol{x}$ is a vector, $\mathrm{hard}(\boldsymbol{x})$ represents element-wise operation. And we initialize $\boldsymbol{h}^{(0)}$ as $\mathbf{0}$.

Hint: One simple implementation of the first layer is that you just add up the values of each unit, including the carry. Activate the first one of your hidden units if the sum is at least 1, the second one if it is at least 2, and the third one if it is 3. Obviously, the answer is not unique, and you are encouraged to find more interesting implementation, but one is enough for this homework.

解: 根据提示, 可设计 RNN 如下.

$$\boldsymbol{U} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{V} = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}, \quad \boldsymbol{W} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{b_h} = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix}, \quad b_y = -\frac{1}{2} \tag{4}$$

初始化 $\boldsymbol{h}^{(0)} = \mathbf{0}$, 对于示例题目可计算如下. 输入为

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5}$$

正确输出为

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

第 1 步计算:

$$\boldsymbol{h}^{(1)} = \mathrm{hard}\left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix}\right) = \mathrm{hard}\left(\begin{bmatrix} 0 \\ -1 \\ -2 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

$$y^{(1)} = \text{hard}\left(\begin{bmatrix}1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} - \frac{1}{2}\right) = \text{hard}\left(\frac{1}{2}\right) = 1 \tag{8}$$

第 2 步计算:

$$\boldsymbol{h}^{(2)} = \text{hard}\left(\begin{bmatrix}1 & 1\\1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} + \begin{bmatrix}-1\\-2\\-3\end{bmatrix}\right) = \text{hard}\left(\begin{bmatrix}3/2\\1/2\\-1/2\end{bmatrix}\right) = \begin{bmatrix}1\\1\\0\end{bmatrix} \tag{9}$$

$$y^{(2)} = \text{hard}\left(\begin{bmatrix}1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} - \frac{1}{2}\right) = \text{hard}\left(-\frac{1}{2}\right) = 0 \tag{10}$$

第 3 步计算:

$$\boldsymbol{h}^{(3)} = \text{hard}\left(\begin{bmatrix}1 & 1\\1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} + \begin{bmatrix}-1\\-2\\-3\end{bmatrix}\right) = \text{hard}\left(\begin{bmatrix}1\\0\\-1\end{bmatrix}\right) = \begin{bmatrix}1\\1\\0\end{bmatrix} \tag{11}$$

$$y^{(3)} = \text{hard}\left(\begin{bmatrix}1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} - \frac{1}{2}\right) = \text{hard}\left(-\frac{1}{2}\right) = 0 \tag{12}$$

第 4 步计算:

$$\boldsymbol{h}^{(4)} = \text{hard}\left(\begin{bmatrix}1 & 1\\1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} + \begin{bmatrix}-1\\-2\\-3\end{bmatrix}\right) = \text{hard}\left(\begin{bmatrix}0\\-1\\-2\end{bmatrix}\right) = \begin{bmatrix}1\\0\\0\end{bmatrix} \tag{13}$$

$$y^{(4)} = \text{hard}\left(\begin{bmatrix}1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} - \frac{1}{2}\right) = \text{hard}\left(\frac{1}{2}\right) = 1 \tag{14}$$

第 5 步计算:

$$\boldsymbol{h}^{(5)} = \text{hard}\left(\begin{bmatrix}1 & 1\\1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} + \begin{bmatrix}-1\\-2\\-3\end{bmatrix}\right) = \text{hard}\left(\begin{bmatrix}3/2\\1/2\\-1/2\end{bmatrix}\right) = \begin{bmatrix}1\\1\\0\end{bmatrix} \tag{15}$$

$$y^{(5)} = \text{hard}\left(\begin{bmatrix}1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} - \frac{1}{2}\right) = \text{hard}\left(-\frac{1}{2}\right) = 0 \tag{16}$$

第 6 步计算:

$$\boldsymbol{h}^{(6)} = \text{hard}\left(\begin{bmatrix}1 & 1\\1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} + \begin{bmatrix}-1\\-2\\-3\end{bmatrix}\right) = \text{hard}\left(\begin{bmatrix}1\\0\\-1\end{bmatrix}\right) = \begin{bmatrix}1\\1\\0\end{bmatrix} \tag{17}$$

$$y^{(6)} = \text{hard}\left(\begin{bmatrix}1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} - \frac{1}{2}\right) = \text{hard}\left(-\frac{1}{2}\right) = 0 \tag{18}$$

第 7 步计算:

$$\boldsymbol{h}^{(7)} = \text{hard}\left(\begin{bmatrix}1 & 1\\1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} + \begin{bmatrix}-1\\-2\\-3\end{bmatrix}\right) = \text{hard}\left(\begin{bmatrix}0\\-1\\-2\end{bmatrix}\right) = \begin{bmatrix}1\\0\\0\end{bmatrix} \tag{19}$$

$$y^{(7)} = \text{hard}\left(\begin{bmatrix}1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} - \frac{1}{2}\right) = \text{hard}\left(\frac{1}{2}\right) = 1 \tag{20}$$

由于

$$\boldsymbol{y} = \begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}y^{(1)} & y^{(2)} & y^{(3)} & y^{(4)} & y^{(5)} & y^{(6)} & y^{(7)}\end{bmatrix} \tag{21}$$

则上述设计的 RNN 可以实现二进制加法.

Long-Term Short-Term Memory (LSTM)

2. Here, you'll derive the Backprop Through Time (BPTT) equations for the univariate version of the Long-Term Short-Term Memory (LSTM) architecture.

Note: This question is an important context for understanding LSTMs, but it is just ordinary BPTT, so you have enough knowledge to do parts (a), (b). As for parts (c), you may find it helpful to read more materials about LSTM.

For reference, here are the computations it performs for inputs $x^{(t)}$, $t = 1, 2, \ldots, T$:

$$\begin{aligned}
i^{(t)} &= \sigma\left(w_{ix}x^{(t)} + w_{ih}h^{(t-1)}\right) \\
f^{(t)} &= \sigma\left(w_{fx}x^{(t)} + w_{fh}h^{(t-1)}\right) \\
o^{(t)} &= \sigma\left(w_{ox}x^{(t)} + w_{oh}h^{(t-1)}\right) \\
g^{(t)} &= \tanh\left(w_{gx}x^{(t)} + w_{gh}h^{(t-1)}\right) \\
c^{(t)} &= f^{(t)}c^{(t-1)} + i^{(t)}g^{(t)} \\
h^{(t)} &= o^{(t)}\tanh\left(c^{(t)}\right)
\end{aligned} \tag{22}$$

And the loss function $\mathcal{L}$ with label y is:

$$\mathcal{L} = \frac{1}{2}\left(y - h^{(T)}\right)^2 \tag{23}$$

A slightly more convenient notation:

- σ is the activation function, we use Sigmoid function here.
- Use $\bar{y}$ to denote the derivative $\partial\mathcal{L}/\partial y$, sometimes called the error signal, where $\mathcal{L}$ is the loss function, y can be any intermediate variable. This emphasizes that the error signals are just values our program is computing rather than a mathematical operation.
- As an example, we compute the loss:

$$\begin{aligned}
z &= wx + b \\
y &= \sigma(z) \\
\mathcal{L} &= \frac{1}{2}(y - y_{\text{pred}})^2
\end{aligned} \tag{24}$$

  Then we could compute the derivatives:

$$\bar{y} = y - y_{\text{pred}}, \quad \bar{z} = \bar{y}\sigma'(z), \quad \bar{w} = \bar{z}x, \quad \bar{b} = \bar{z} \tag{25}$$

- You should use this notation in the following questions.

Suppose that $T = 2$, and the initial values of $h^{(0)}$ and $c^{(0)}$ are known.

(a) Derive the Backprop Through Time equations for the activations and the gates for $t = 1, 2$.

$$\overline{h^{(t)}}, \quad \overline{c^{(t)}}, \quad \overline{o^{(t)}}, \quad \overline{g^{(t)}}, \quad \overline{i^{(t)}}, \quad \overline{f^{(t)}}. \tag{26}$$

解: 由于本题是单变量 (univariate) 版本 LSTM, 即全部变量为实数, 故将 $\odot$ 省略.

由 $T = 2$ 可知损失函数 $\mathcal{L}$ 为

$$\mathcal{L} = \frac{1}{2}\left(y - h^{(2)}\right)^2 \tag{27}$$

所以, 损失函数 $\mathcal{L}$ 对 $h^{(2)}$ 的偏导数为

$$\overline{h^{(2)}} = \frac{\partial \mathcal{L}}{\partial h^{(2)}} = h^{(2)} - y \tag{28}$$

损失函数 $\mathcal{L}$ 对 $c^{(2)}$ 的偏导数为

$$\overline{c^{(2)}} = \frac{\partial \mathcal{L}}{\partial c^{(2)}} = \frac{\partial \mathcal{L}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial c^{(2)}} = \overline{h^{(2)}} o^{(2)} \left[1 - \tanh^2\left(c^{(2)}\right)\right] \tag{29}$$

损失函数 $\mathcal{L}$ 对 $o^{(2)}$ 的偏导数为

$$\overline{o^{(2)}} = \frac{\partial \mathcal{L}}{\partial o^{(2)}} = \frac{\partial \mathcal{L}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial o^{(2)}} = \overline{h^{(2)}} \tanh\left(c^{(2)}\right) \tag{30}$$

损失函数 $\mathcal{L}$ 对 $g^{(2)}$ 的偏导数为

$$\overline{g^{(2)}} = \frac{\partial \mathcal{L}}{\partial g^{(2)}} = \frac{\partial \mathcal{L}}{\partial c^{(2)}} \frac{\partial c^{(2)}}{\partial g^{(2)}} = \overline{c^{(2)}} i^{(2)} \tag{31}$$

损失函数 $\mathcal{L}$ 对 $i^{(2)}$ 的偏导数为

$$\overline{i^{(2)}} = \frac{\partial \mathcal{L}}{\partial i^{(2)}} = \frac{\partial \mathcal{L}}{\partial c^{(2)}} \frac{\partial c^{(2)}}{\partial i^{(2)}} = \overline{c^{(2)}} g^{(2)} \tag{32}$$

损失函数 $\mathcal{L}$ 对 $f^{(2)}$ 的偏导数为

$$\overline{f^{(2)}} = \frac{\partial \mathcal{L}}{\partial f^{(2)}} = \frac{\partial \mathcal{L}}{\partial c^{(2)}} \frac{\partial c^{(2)}}{\partial f^{(2)}} = \overline{c^{(2)}} c^{(1)} \tag{33}$$

损失函数 $\mathcal{L}$ 对 $h^{(1)}$ 的偏导数为

$$\begin{aligned}
\overline{h^{(1)}} &= \frac{\partial \mathcal{L}}{\partial h^{(1)}} \\
&= \frac{\partial \mathcal{L}}{\partial i^{(2)}} \frac{\partial i^{(2)}}{\partial h^{(1)}} + \frac{\partial \mathcal{L}}{\partial f^{(2)}} \frac{\partial f^{(2)}}{\partial h^{(1)}} + \frac{\partial \mathcal{L}}{\partial o^{(2)}} \frac{\partial o^{(2)}}{\partial h^{(1)}} + \frac{\partial \mathcal{L}}{\partial g^{(2)}} \frac{\partial g^{(2)}}{\partial h^{(1)}} \\
&= \overline{i^{(2)}} i^{(2)}(1 - i^{(2)}) w_{ih} + \overline{f^{(2)}} f^{(2)}(1 - f^{(2)}) w_{fh} + \overline{o^{(2)}} o^{(2)}(1 - o^{(2)}) w_{oh} + \overline{g^{(2)}} [1 - (g^{(2)})^2] w_{gh}
\end{aligned} \tag{34}$$

损失函数 $\mathcal{L}$ 对 $c^{(1)}$ 的偏导数为

$$\begin{aligned}
\overline{c^{(1)}} &= \frac{\partial \mathcal{L}}{\partial c^{(1)}} \\
&= \frac{\partial \mathcal{L}}{\partial c^{(2)}} \frac{\partial c^{(2)}}{\partial c^{(1)}} + \frac{\partial \mathcal{L}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial c^{(1)}} \\
&= \overline{c^{(2)}} f^{(2)} + \overline{h^{(1)}} o^{(1)} \left[1 - \tanh^2\left(c^{(1)}\right)\right]
\end{aligned} \tag{35}$$

损失函数 $\mathcal{L}$ 对 $o^{(1)}$ 的偏导数为

$$\overline{o^{(1)}} = \frac{\partial \mathcal{L}}{\partial o^{(1)}} = \frac{\partial \mathcal{L}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial o^{(1)}} = \overline{h^{(1)}} \tanh\left(c^{(1)}\right) \tag{36}$$

损失函数 $\mathcal{L}$ 对 $g^{(1)}$ 的偏导数为

$$\overline{g^{(1)}} = \frac{\partial \mathcal{L}}{\partial g^{(1)}} = \frac{\partial \mathcal{L}}{\partial c^{(1)}} \frac{\partial c^{(1)}}{\partial g^{(1)}} = \overline{c^{(1)}} i^{(1)} \tag{37}$$

损失函数 $\mathcal{L}$ 对 $i^{(1)}$ 的偏导数为

$$\overline{i^{(1)}} = \frac{\partial \mathcal{L}}{\partial i^{(1)}} = \frac{\partial \mathcal{L}}{\partial c^{(1)}} \frac{\partial c^{(1)}}{\partial i^{(1)}} = \overline{c^{(1)}} g^{(1)} \tag{38}$$

损失函数 $\mathcal{L}$ 对 $f^{(1)}$ 的偏导数为

$$\overline{f^{(1)}} = \frac{\partial \mathcal{L}}{\partial f^{(1)}} = \frac{\partial \mathcal{L}}{\partial c^{(1)}} \frac{\partial c^{(1)}}{\partial f^{(1)}} = \overline{c^{(1)}} c^{(0)} \tag{39}$$

(b) Derive the Backprop Through Time equation for the weight w_{ix}. (The other weight matrices are basically the same, so we won’t make you write those out.)

解: 损失函数 $\mathcal{L}$ 对 w_{ix} 的偏导数为

$$\begin{aligned} \overline{w_{ix}} &= \frac{\partial \mathcal{L}}{\partial w_{ix}} \\ &= \frac{\partial \mathcal{L}}{\partial i^{(2)}} \frac{\partial i^{(2)}}{\partial w_{ix}} + \frac{\partial \mathcal{L}}{\partial i^{(1)}} \frac{\partial i^{(1)}}{\partial w_{ix}} \\ &= \overline{i^{(2)}} i^{(2)} (1 - i^{(2)}) x^{(2)} + \overline{i^{(1)}} i^{(1)} (1 - i^{(1)}) x^{(1)} \end{aligned} \tag{40}$$

(c) (*Optional*) Based on your answers above, can you explain why the gradient doesn’t explode if the values of the forget gates ($f^{(t)}$) are very close to 1 and the values of the input and output gates ($i^{(t)}$ and $o^{(t)}$) are very close to 0? (Your answer may involve both $\overline{h^{(t)}}$ and $\overline{c^{(t)}}$.)

解: 由 $f^{(t)} \approx 1$, $o^{(t)} \approx 0$, $\forall\, t = 1, 2, \ldots, T$ 可知

$$\overline{c^{(t)}} = \overline{c^{(t+1)}} f^{(t+1)} + \overline{h^{(t)}} o^{(t)} \left[1 - \tanh^2\left(c^{(t)}\right)\right] \approx \overline{c^{(t+1)}}, \quad \forall\, t = 1, 2, \ldots, T-1 \tag{41}$$

所以

$$\overline{c^{(t)}} \approx \overline{c^{(T)}}, \quad \forall\, t = 1, 2, \ldots, T-1 \tag{42}$$

损失函数 $\mathcal{L}$ 对 w_{ix} 的偏导数为

$$\begin{aligned} \overline{w_{ix}} &= \sum_{t=1}^{T} \overline{i^{(t)}} i^{(t)} (1 - i^{(t)}) x^{(t)} \\ &= \sum_{t=1}^{T} \overline{c^{(t)}} g^{(t)} i^{(t)} (1 - i^{(t)}) x^{(t)} \\ &\approx \sum_{t=1}^{T} \overline{c^{(T)}} g^{(t)} i^{(t)} (1 - i^{(t)}) x^{(t)} \\ &= \sum_{t=1}^{T} \overline{h^{(T)}} \left[1 - \tanh^2\left(c^{(T)}\right)\right] g^{(t)} i^{(t)} (1 - i^{(t)}) x^{(t)} \\ &= \sum_{t=1}^{T} (h^{(T)} - y) \left[1 - \tanh^2\left(c^{(T)}\right)\right] g^{(t)} i^{(t)} (1 - i^{(t)}) x^{(t)} \end{aligned} \tag{43}$$

注意到 $i^{(t)} \approx 0, \forall\, t = 1, 2, \ldots, T$, 对 $|\overline{w_{ix}}|$ 估计其上界得

$$\begin{aligned}
|\overline{w_{ix}}| &= |h^{(T)} - y| \left[1 - \tanh^2\left(c^{(T)}\right)\right] \left|\sum_{t=1}^{T} g^{(t)} i^{(t)} (1 - i^{(t)}) x^{(t)}\right| \\
&\leqslant |h^{(T)} - y| \left[1 - \tanh^2\left(c^{(T)}\right)\right] \sum_{t=1}^{T} g^{(t)} i^{(t)} (1 - i^{(t)}) |x^{(t)}| \\
&\leqslant (1 + |y|) \sum_{t=1}^{T} i^{(t)} (1 - i^{(t)}) \max_{1 \leqslant t \leqslant T} |x^{(t)}| \\
&= (1 + |y|) \max_{1 \leqslant t \leqslant T} |x^{(t)}| \sum_{t=1}^{T} i^{(t)} (1 - i^{(t)}) \\
&\approx 0
\end{aligned} \tag{44}$$

因此梯度 $\overline{w_{ix}}$ 并不会爆炸.

Attention

3. Recall that attention can be viewed as an operation on a query $\boldsymbol{q} \in \mathbb{R}^d$, a set of value vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$, $\boldsymbol{v}_i \in \mathbb{R}^d$, and a set of key vectors $\{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n\}$, $\boldsymbol{k}_i \in \mathbb{R}^d$, specified as follows:

$$\boldsymbol{c} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i \tag{45}$$

$$\alpha_i = \frac{\exp(\boldsymbol{k}_i^\top \boldsymbol{q})}{\sum_{j=1}^{n} \exp(\boldsymbol{k}_j^\top \boldsymbol{q})} \tag{46}$$

where α_i are frequently called the "attention weights", and the output $\boldsymbol{c} \in \mathbb{R}^d$ is a correspondingly weighted average over the value vectors.

(a) **Copying via attention:** We'll first show that it's particularly simple for attention to "copy" a value vector to the output $\boldsymbol{c} \in \mathbb{R}^d$. Describe (in one sentence) what properties of the inputs to the attention operation would result in the output $\boldsymbol{c}$ being *approximately* equal to $\boldsymbol{v}_j$ for some $j \in \{1, 2, \ldots, n\}$. Specifically, what must be true about the query $\boldsymbol{q}$, the values $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ and/or the keys $\{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n\}$?

Hint: When does softmax result approach a one-hot vector?

解: 当 $\boldsymbol{k}_j^\top \boldsymbol{q} \gg \boldsymbol{k}_i^\top \boldsymbol{q}, \forall\, i \neq j$ 时, $\boldsymbol{k}_j$ 对应的注意力权重为

$$\alpha_j = \frac{\exp(\boldsymbol{k}_j^\top \boldsymbol{q})}{\sum_{j=1}^{n} \exp(\boldsymbol{k}_j^\top \boldsymbol{q})} \approx 1 \tag{47}$$

则有

$$\boldsymbol{c} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i \approx \boldsymbol{v}_j \tag{48}$$

(b) **An average of two value vectors via attention:** Consider a set of key vectors $\{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n\}$ where all key vectors are perpendicular, that is $\boldsymbol{k}_i \perp \boldsymbol{k}_j$ for all $i \neq j$. Let $\|\boldsymbol{k}_i\| = 1$ for all i. Let $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ be a set of arbitrary value vectors. Let $\boldsymbol{v}_a, \boldsymbol{v}_b \in \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ be two of the value vectors. Design a query vector $\boldsymbol{q}$ such that the output $\boldsymbol{c}$ is *approximately* equal to the average of $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$, that is, $\frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b)$. Note that you can reference the corresponding key vector of $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$ as $\boldsymbol{k}_a$ and $\boldsymbol{k}_b$.

Hint: While the softmax function will never exactly average the two vectors, you can get close by using a large scalar multiple in the expression.

解: 令 $\boldsymbol{q} = M(\boldsymbol{k}_a + \boldsymbol{k}_b)$, 其中 $M \in \mathbb{R}$ 是一个大数, 则 $\boldsymbol{k}_a$ 注意力权重为

$$\alpha_a = \frac{\exp(\boldsymbol{k}_a^\top \boldsymbol{q})}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q})} = \frac{\exp(M)}{2\exp(M) + n - 2} \approx \frac{1}{2} \tag{49}$$

同理可得, $\boldsymbol{k}_b$ 注意力权重为

$$\alpha_b = \frac{\exp(\boldsymbol{k}_b^\top \boldsymbol{q})}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q})} = \frac{\exp(M)}{2\exp(M) + n - 2} \approx \frac{1}{2} \tag{50}$$

所以

$$\boldsymbol{c} = \sum_{i=1}^n \alpha_i \boldsymbol{v}_i \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b) \tag{51}$$

(c) **Drawbacks of single-headed attention:** In the previous part, we saw how it was possible for a single-headed attention to focus equally on two values. The same concept could easily be extended to any subset of values. In this question we'll see why it's not a practical solution. Consider a set of key vectors $\{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n\}$ that are now randomly sampled, $\boldsymbol{k}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where the means $\boldsymbol{\mu}_i$ are known to you, but the covariances $\boldsymbol{\Sigma}_i$ are unknown. Further, assume that the means $\boldsymbol{\mu}_i$ are all perpendicular; $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j = 0$ if $i \neq j$, and unit norm, $\|\boldsymbol{\mu}_i\| = 1$.

i. Assume that the covariance matrices are $\boldsymbol{\Sigma}_i = \alpha \boldsymbol{I}$, for vanishingly small α. Design a query vector $\boldsymbol{q}$ in terms of the $\boldsymbol{\mu}_i$ such that as before, $\boldsymbol{c} \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b)$, and provide a brief argument as to why it works.

解: 令 $\boldsymbol{q} = M(\boldsymbol{\mu}_a + \boldsymbol{\mu}_b)$, 其中 $M \in \mathbb{R}$ 是一个大数. 由于协方差矩阵 $\boldsymbol{\Sigma}_i = \alpha \boldsymbol{I}$ 为对角线元素非常小的对角矩阵, 则 $\boldsymbol{k}_i \approx \boldsymbol{\mu}_i$, 又 $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j = 0, \forall\ i \neq j$, 则有 $\boldsymbol{k}_i^\top \boldsymbol{q} \approx 0, \forall\ i \neq a, b$. 则 $\boldsymbol{k}_a$ 注意力权重为

$$\alpha_a = \frac{\exp(\boldsymbol{k}_a^\top \boldsymbol{q})}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q})} \approx \frac{\exp(M)}{2\exp(M) + n - 2} \approx \frac{1}{2} \tag{52}$$

同理可得, $\boldsymbol{k}_b$ 注意力权重为

$$\alpha_b = \frac{\exp(\boldsymbol{k}_b^\top \boldsymbol{q})}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q})} \approx \frac{\exp(M)}{2\exp(M) + n - 2} \approx \frac{1}{2} \tag{53}$$

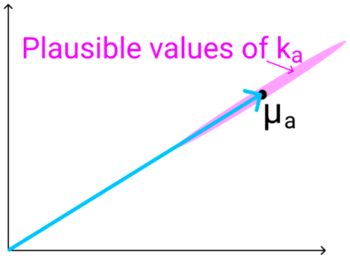


图 1: The vector $\boldsymbol{\mu}_a$ (shown here in 2D as an example), with the range of possible values of $\boldsymbol{k}_a$ shown in red. As mentioned previously, $\boldsymbol{k}_a$ points in roughly the same direction as $\boldsymbol{\mu}_a$, but may have larger or smaller magnitude.

所以

$$\boldsymbol{c} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b) \tag{54}$$

ii. Though single-headed attention is resistant to small perturbations in the keys, some types of larger perturbations may pose a bigger issue. Specifically, in some cases, one key vector $\boldsymbol{k}_a$ may be larger or smaller in norm than the others, while still pointing in the same direction as $\boldsymbol{\mu}_a$. As an example, let us consider a covariance for item a as $\boldsymbol{\Sigma}_a = \alpha \boldsymbol{I} + \frac{1}{2}\boldsymbol{\mu}_a\boldsymbol{\mu}_a^\top$ for vanishingly small α (as shown in Figure 1). Further, let $\boldsymbol{\Sigma}_i = \alpha\boldsymbol{I}$ for all $i \neq a$. When you sample $\{\boldsymbol{k}_1, \boldsymbol{k}_2, \dots, \boldsymbol{k}_n\}$ multiple times, and use the $\boldsymbol{q}$ vector that you designed in part i., what qualitatively do you expect the vector $\boldsymbol{c}$ will look like for different samples?

解: 定性来说, 不同采样得到的 $\boldsymbol{c}$ 有很大的差别. 由 $\boldsymbol{\Sigma}_a = \alpha \boldsymbol{I} + \frac{1}{2}\boldsymbol{\mu}_a\boldsymbol{\mu}_a^\top$ 及图 1 可知 $\boldsymbol{k}_a$ 的方向近似为 $\boldsymbol{\mu}_a$ 的方向, $\boldsymbol{k}_a$ 的长度大多数情况位于以下区间

$$\|\boldsymbol{k}_a\| \in [1-\beta,\ 1+\beta], \quad \beta \in (0,1) \tag{55}$$

由 i. 可知 $\boldsymbol{q} = M(\boldsymbol{\mu}_a + \boldsymbol{\mu}_b)$, 则 $\boldsymbol{k}_a$ 注意力权重为

$$\begin{aligned}\alpha_a &= \frac{\exp(\boldsymbol{k}_a^\top \boldsymbol{q})}{\sum_{j=1}^{n}\exp(\boldsymbol{k}_j^\top \boldsymbol{q})} \\ &\in \left[\frac{\exp((1-\beta)M)}{\exp((1-\beta)M)+\exp(M)+n-2}, \frac{\exp((1+\beta)M)}{\exp((1+\beta)M)+\exp(M)+n-2}\right] \\ &\approx [0,1]\end{aligned} \tag{56}$$

$\boldsymbol{k}_b$ 注意力权重为

$$\begin{aligned}\alpha_b &= \frac{\exp(\boldsymbol{k}_b^\top \boldsymbol{q})}{\sum_{j=1}^{n}\exp(\boldsymbol{k}_j^\top \boldsymbol{q})} \\ &\in \left[\frac{\exp(M)}{\exp((1+\beta)M)+\exp(M)+n-2}, \frac{\exp(M)}{\exp((1-\beta)M)+\exp(M)+n-2}\right] \\ &\approx [0,1]\end{aligned} \tag{57}$$

则由

$$\boldsymbol{c} \approx \alpha_a \boldsymbol{v}_a + \alpha_b \boldsymbol{v}_b \tag{58}$$

可知 $\boldsymbol{c}$ 与 $\boldsymbol{k}_a$ 的长度大小有非常强的关联, 当 $\boldsymbol{k}_a$ 的长度发生变化时, $\boldsymbol{c}$ 也会发生很大的变化.

(d) **Benefits of multi-headed attention:** Now we'll see some power of multi-headed attention. We'll consider a simple version of multi-headed attention which is identical to single-headed self-attention as we've presented it in this homework, except two query vectors ($\boldsymbol{q}_1$ and $\boldsymbol{q}_2$) are defined, which leads to a pair of vectors ($\boldsymbol{c}_1$ and $\boldsymbol{c}_2$), each the output of single-headed attention given its respective query vector. The final output of the multi-headed attention is their average, $\frac{1}{2}(\boldsymbol{c}_1 + \boldsymbol{c}_2)$. As in question (c), consider a set of key vectors $\{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n\}$ that are randomly sampled $\boldsymbol{k}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where the means $\boldsymbol{\mu}_i$ are known to you, but the covariances $\boldsymbol{\Sigma}_i$ are unknown. Also as before, assume that the means $\boldsymbol{\mu}_i$ are mutually orthogonal; $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j = 0$ if $i \neq j$, and unit norm, $\|\boldsymbol{\mu}_i\| = 1$.

i. Assume that the covariance matrices are $\boldsymbol{\Sigma}_i = \alpha \boldsymbol{I}$, for vanishingly small α. Design query vectors $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ in terms of the $\boldsymbol{\mu}_i$ such that $\boldsymbol{c}$ is *approximately* equal to $\frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b)$.

Hint: For the convenience of further analysis, you'd better recall the copy operation in question (a).

解: 令 $\boldsymbol{q}_1 = M\boldsymbol{\mu}_a$, $\boldsymbol{q}_2 = M\boldsymbol{\mu}_b$, 由协方差矩阵 $\boldsymbol{\Sigma}_i = \alpha \boldsymbol{I}$ 可知 $\boldsymbol{k}_i \approx \boldsymbol{\mu}_i$, 则 $\boldsymbol{k}_a$ 相对 $\boldsymbol{q}_1$ 注意力权重为

$$\alpha_a = \frac{\exp(\boldsymbol{k}_a^\top \boldsymbol{q}_1)}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q}_1)} \approx 1 \tag{59}$$

则对 $\boldsymbol{q}_1$ 输出

$$\boldsymbol{c}_1 = \sum_{i=1}^n \alpha_i \boldsymbol{v}_i \approx \boldsymbol{v}_a \tag{60}$$

$\boldsymbol{k}_b$ 相对 $\boldsymbol{q}_2$ 注意力权重为

$$\alpha_b = \frac{\exp(\boldsymbol{k}_b^\top \boldsymbol{q}_2)}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q}_2)} \approx 1 \tag{61}$$

则对 $\boldsymbol{q}_2$ 输出

$$\boldsymbol{c}_2 = \sum_{i=1}^n \alpha_i \boldsymbol{v}_i \approx \boldsymbol{v}_b \tag{62}$$

所以

$$\boldsymbol{c} = \frac{1}{2}(\boldsymbol{c}_1 + \boldsymbol{c}_2) \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b) \tag{63}$$

ii. Assume that the covariance matrices are $\boldsymbol{\Sigma}_a = \alpha \boldsymbol{I} + \frac{1}{2}\left(\boldsymbol{\mu}_a \boldsymbol{\mu}_a^\top\right)$ for vanishingly small α, and $\boldsymbol{\Sigma}_i = \alpha \boldsymbol{I}$ for all $i \neq a$. Take the query vectors $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ that you designed in part i. What, qualitatively, do you expect the vector $\boldsymbol{c}$ will look like for different samples? Please briefly explain why. You can ignore cases in which $\boldsymbol{k}_a^\top \boldsymbol{q}_i < 0$.

解: 定性来说, 不同采样得到的 $\boldsymbol{c}$ 几乎相同. 由 $\boldsymbol{\Sigma}_a = \alpha \boldsymbol{I} + \frac{1}{2}\boldsymbol{\mu}_a\boldsymbol{\mu}_a^\top$ 及图 1 可知 $\boldsymbol{k}_a$ 的方向近似为 $\boldsymbol{\mu}_a$ 的方向, $\boldsymbol{k}_a$ 的长度多数情况位于以下区间

$$\|\boldsymbol{k}_a\| \in [1-\beta,\ 1+\beta], \quad \beta \in (0,1) \tag{64}$$

忽略 $\boldsymbol{k}_a^\top \boldsymbol{q}_1 < 0$ 的情况, 由 $\boldsymbol{q}_1 = M\boldsymbol{\mu}_a$ 可知 $\boldsymbol{k}_a$ 相对 $\boldsymbol{q}_1$ 注意力权重为

$$\alpha_a = \frac{\exp(\boldsymbol{k}_a^\top \boldsymbol{q}_1)}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q}_1)} \approx \frac{\exp(\|\boldsymbol{k}_a\| M)}{\exp(\|\boldsymbol{k}_a\| M) + n - 1} \approx 1 \tag{65}$$

$\boldsymbol{k}_b$ 相对 $\boldsymbol{q}_2$ 注意力权重没有变化, 仍然为

$$\alpha_b = \frac{\exp(\boldsymbol{k}_b^\top \boldsymbol{q}_2)}{\sum_{j=1}^n \exp(\boldsymbol{k}_j^\top \boldsymbol{q}_2)} \approx 1 \tag{66}$$

注意到 $\alpha_a \approx 1$ 与 $\boldsymbol{k}_a$ 的长度几乎无关, 则不同的采样仍然都会得到

$$\boldsymbol{c} \approx \frac{1}{2}(\boldsymbol{v}_a + \boldsymbol{v}_b) \tag{67}$$

(e) **Visualization:** Use the computer to sample 2-dimension key vectors multiple times and visualize the distributions of the output $\boldsymbol{c}$ of query $\boldsymbol{q}$ designed in (c) and (d) under different covariance matrices conditions. More specifically, we set $n = d = 2$, $\boldsymbol{v}_1 = \boldsymbol{\mu}_1 = (0,1)^\top$ and $\boldsymbol{v}_2 = \boldsymbol{\mu}_2 = (1,0)^\top$. You may have to carefully choose α to produce the desired phenomenon, we recommend $\alpha = 1 \times 10^{-10}$. Does this small experiment verify your findings in the above questions?

解: 这个小的数值实验可以验证上述题目的分析结果. 对键向量 $\{\boldsymbol{k}_1, \boldsymbol{k}_2\}$ 随机采样 10 次, 得到 (c) 题的可视化结果如图 2 和 3 所示.

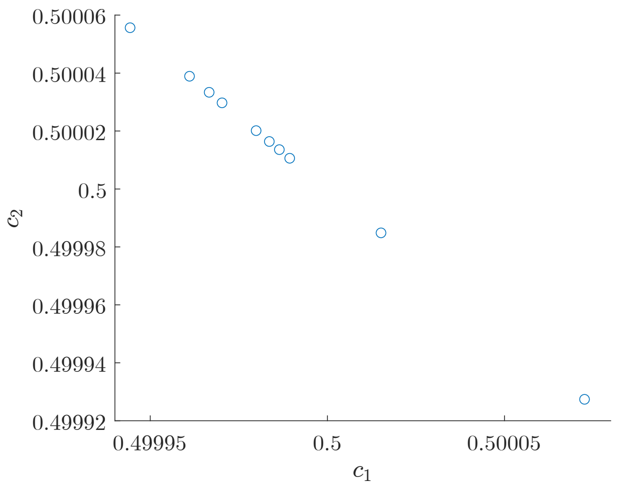


图 2: (c) i. 注意力机制输出结果 $\boldsymbol{c}$

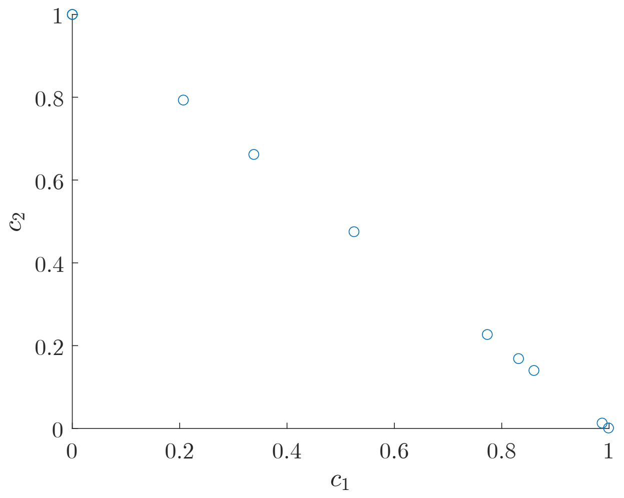


图 3: (c) ii. 注意力机制输出结果 $\boldsymbol{c}$

由图 2 可知, 对单向 (single-headed) 注意力机制而言, 当 $\boldsymbol{\Sigma}_i = \alpha\boldsymbol{I}$ 时, 输出的结果都是 $\boldsymbol{c} \approx \frac{1}{2}(\boldsymbol{v}_1 + \boldsymbol{v}_2)$; 由图 3 可知, 当 $\boldsymbol{\Sigma}_1 = \alpha\boldsymbol{I} + \frac{1}{2}\boldsymbol{\mu}_1\boldsymbol{\mu}_1^\top$ 时, 输出结果 $\boldsymbol{c}$ 是 $\boldsymbol{v}_1$ 与 $\boldsymbol{v}_2$ 的任意凸组合.

(d) 题的可视化结果如图 4 和 5 所示.

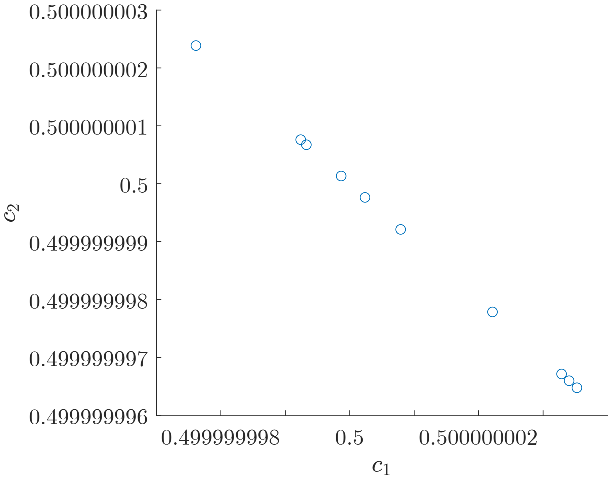


图 4: (d) i. 注意力机制输出结果 $\boldsymbol{c}$

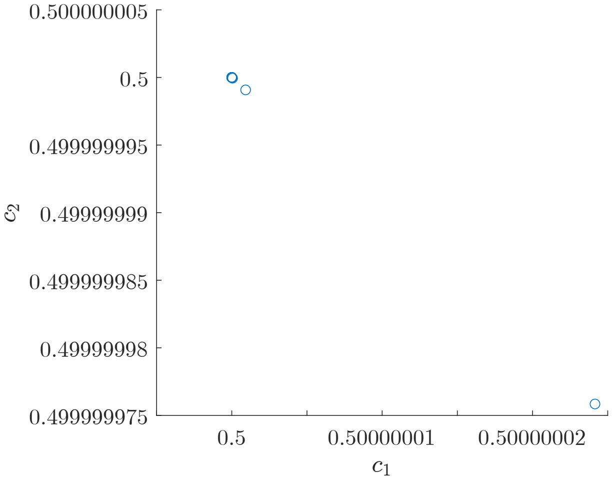


图 5: (d) ii. 注意力机制输出结果 $\boldsymbol{c}$

由图 4 可知, 对多向 (multi-headed) 注意力机制而言, 当 $\boldsymbol{\Sigma}_i = \alpha \boldsymbol{I}$ 时, 输出的结果都是 $\boldsymbol{c} \approx \frac{1}{2}(\boldsymbol{v}_1 + \boldsymbol{v}_2)$; 由图 5 可知, 当 $\boldsymbol{\Sigma}_1 = \alpha \boldsymbol{I} + \frac{1}{2}\boldsymbol{\mu}_1\boldsymbol{\mu}_1^\top$ 时, 忽略 $\boldsymbol{k}_1^\top\boldsymbol{\mu}_1 < 0$ 的情况, 输出的结果也是 $\boldsymbol{c} \approx \frac{1}{2}(\boldsymbol{v}_1 + \boldsymbol{v}_2)$, 不受协方差矩阵 $\boldsymbol{\Sigma}_1$ 变化的影响, 即若 $\boldsymbol{k}_1$ 只是大小发生变化而方向几乎不变则对输出 $\boldsymbol{c}$ 几乎没有影响.

Programming: Natural Language Processing (NLP)

4. Please install PyTorch, Jupyter Notebook and run the NLP tutorial. If you want to use TensorFlow or other deep learning frameworks, please find corresponding language translation tutorial for that framework and run it. The tutorial is the third example for NLP From Scratch in Pytorch tutorials, where we write our own classes and functions to preprocess the data to do NLP modeling tasks. We hope after you complete this tutorial that you'll proceed to learn how torchtext can handle much of this preprocessing for you in the three tutorials immediately following this one. In this project we will be teaching a neural network to translate from French to English. The code is given as a jupyter notebook file.

All you need to do is to read, run and think. You need not write any code or any report in this programming.

参考文献

[1] Olah, Chris. Understanding LSTM Networks, 2015.
`http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.