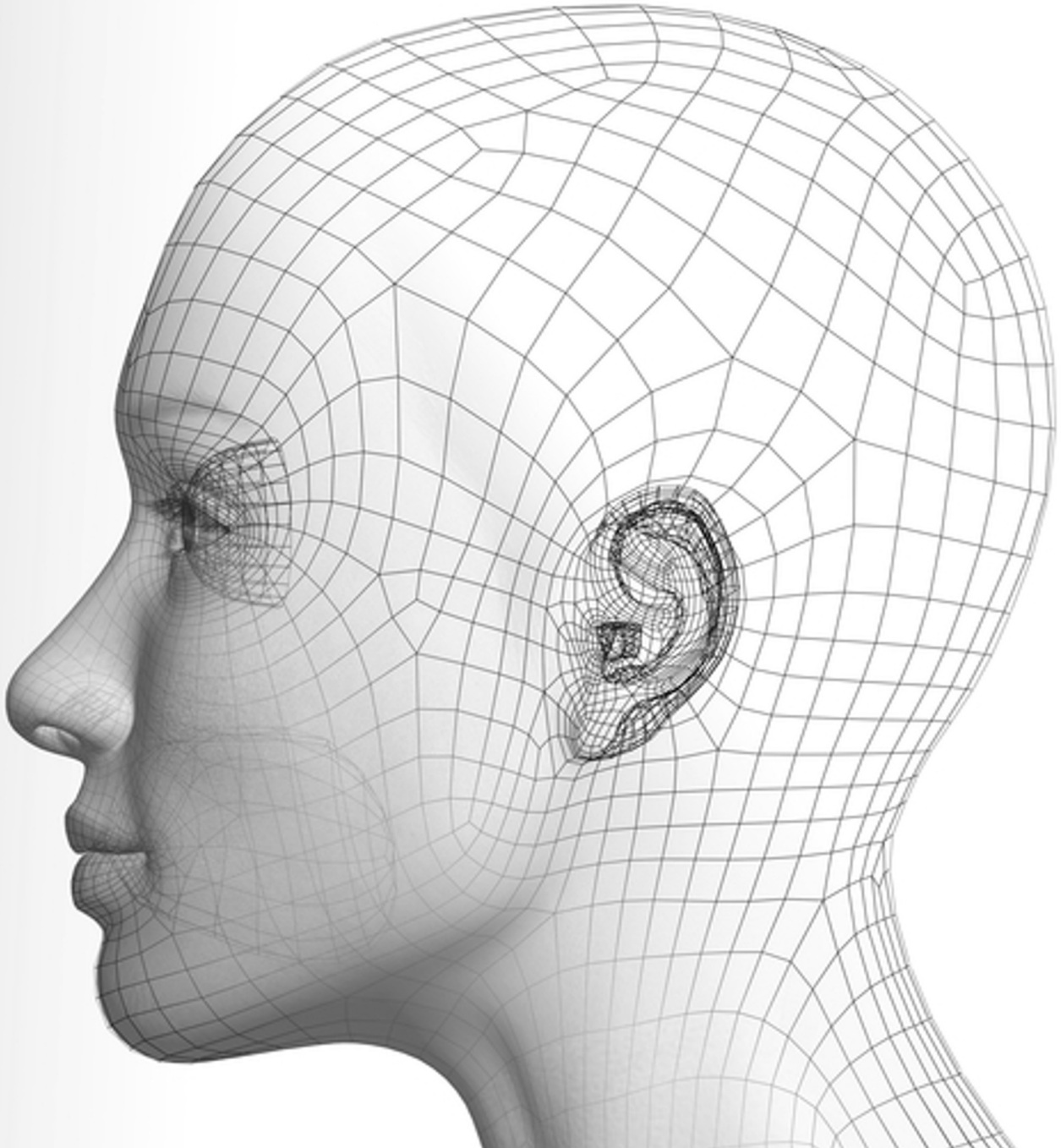
Tutorial 8
Recurrent Neural Networks

Xingang Pan
潘新钢



Question 1

1. A recurrent neural network (RNN) receives sequences of 3-dimensional inputs and has two hidden neurons and one output neuron. The weight matrix $\boldsymbol{U}$ connecting the input to the hidden layer, the weight matrix $\boldsymbol{V}$ connecting the hidden output to the output layer, the hidden layer bias $\boldsymbol{b}$ and the output layer bias $\boldsymbol{c}$ are given by

$$\boldsymbol{U} = \begin{pmatrix} -1.0 & 0.5 \\ 0.5 & 0.1 \\ 0.2 & -2.0 \end{pmatrix}, \boldsymbol{V} = \begin{pmatrix} 2.0 \\ -1.5 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}, \text{ and } \boldsymbol{c} = 0.5.$$

Find the output sequence for an input sequence of $(\boldsymbol{x}(t))_{t=1}^{4}$ where

$$\boldsymbol{x}(1) = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \boldsymbol{x}(2) = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix}, \boldsymbol{x}(3) = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}, \text{ and } \boldsymbol{x}(4) = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} \text{if}$$

(a) The RNN is of hidden-recurrence (Elman) type with the recurrence weight matrix $\boldsymbol{W}$ connecting the previous hidden output to the current hidden layer input is given by

$$\boldsymbol{W} = \begin{pmatrix} 2.0 & 1.3 \\ 1.5 & 0.0 \end{pmatrix}.$$

Assume that the hidden activations are initialized to zero.

(b) The RNN is of top-down recurrence (Jordan) type with the weight matrix $\boldsymbol{W}$ connecting the previous output to the current state of hidden layer is given by

$$\boldsymbol{W} = (2.0 \quad 1.3).$$

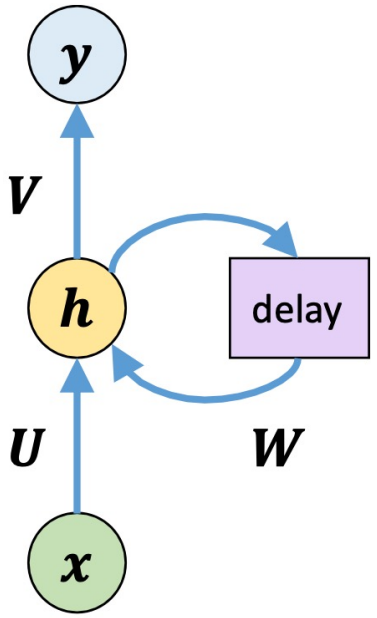
Assume that the output activations are initialized to zero.

t8q1.ipynb

Question 1

Hidden Recurrence:

$$\boldsymbol{U} = \begin{pmatrix} -1.0 & 0.5 \\ 0.5 & 0.1 \\ 0.2 & -2.0 \end{pmatrix}, \boldsymbol{V} = \begin{pmatrix} 2.0 \\ -1.5 \end{pmatrix} \text{ and } \boldsymbol{W} = \begin{pmatrix} 2.0 & 1.3 \\ 1.5 & 0.0 \end{pmatrix}.$$



$$\boldsymbol{b} = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}, \text{ and } \boldsymbol{c} = 0.5$$

Three input neurons, two hidden neurons, and one output neuron.

$$\boldsymbol{h}(t) = \phi\left(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}(t) + \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{h}(t-1) + \boldsymbol{b}\right)$$
$$\boldsymbol{y}(t) = \sigma\left(\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{h}(t) + \boldsymbol{c}\right)$$

$$\phi(u) = \tanh(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}$$
$$\sigma(u) = \text{sigmoid}(u) = \frac{1}{1 + e^{-u}}.$$

Assume $\boldsymbol{h}(0) = (0 \quad 0)^{T}$.

Question 1

At t=1, $\boldsymbol{x}(1) = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$:

$$\boldsymbol{h}(1) = \phi\left(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}(1) + \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{h}(0) + \boldsymbol{b}\right)$$

$$= \tanh\left(\begin{pmatrix} -1.0 & 0.5 & 0.2 \\ 0.5 & 0.1 & -2.0 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} + \begin{pmatrix} 2.0 & 1.5 \\ 1.3 & 0.0 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}\right) = \begin{pmatrix} 0.0 \\ 0.99 \end{pmatrix}$$

$$\boldsymbol{y}(1) = \sigma\left(\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{h}(1) + \boldsymbol{c}\right) = \text{sigmoid}\left((2.0 \quad -1.5)\begin{pmatrix} 0.0 \\ 0.99 \end{pmatrix} + (0.5)\right) = (0.27)$$

Question 1

At t=2, $\boldsymbol{x}(2) = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix}$:

$$\boldsymbol{h}(2) = \phi\left(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}(2) + \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{h}(1) + \boldsymbol{b}\right)$$

$$= \tanh\left(\begin{pmatrix} -1.0 & 0.5 & 0.2 \\ 0.5 & 0.1 & -2.0 \end{pmatrix}\begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix} + \begin{pmatrix} 2.0 & 1.5 \\ 1.3 & 0.0 \end{pmatrix}\begin{pmatrix} 0.0 \\ 0.99 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}\right) = \begin{pmatrix} 0.99 \\ 1.0 \end{pmatrix}$$

$$\boldsymbol{y}(2) = \sigma\left(\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{h}(2) + \boldsymbol{c}\right) = \text{sigmoid}\left((2.0 \quad -1.5)\begin{pmatrix} 0.99 \\ 1.0 \end{pmatrix} + (0.5)\right) = (0.73)$$

Similarly,

at t=3, $\boldsymbol{x}(3) = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}$; $\boldsymbol{h}(3) = \begin{pmatrix} 1.0 \\ -0.21 \end{pmatrix}$ and $\boldsymbol{y}(3) = (0.94)$

at t=4, $\boldsymbol{x}(4) = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$; $\boldsymbol{h}(4) = \begin{pmatrix} -0.54 \\ 0.98 \end{pmatrix}$ and $\boldsymbol{y}(4) = (0.11)$

Question 1

The input sequence $\left(\boldsymbol{x}(1), \boldsymbol{x}(2), \boldsymbol{x}(3), \boldsymbol{x}(4)\right)$:

$$\left(\begin{pmatrix}1\\2\\-1\end{pmatrix} \quad \begin{pmatrix}-1\\1\\-2\end{pmatrix} \quad \begin{pmatrix}0\\3\\1\end{pmatrix} \quad \begin{pmatrix}2\\-1\\0\end{pmatrix}\right)$$

The output sequence $\left(\boldsymbol{y}(1), \boldsymbol{y}(2), \boldsymbol{y}(3), \boldsymbol{y}(4)\right)$:

$$((0.27) \quad (0.73) \quad (0.94) \quad (0.11))$$

Question 1

1. A recurrent neural network (RNN) receives sequences of 3-dimensional inputs and has two hidden neurons and one output neuron. The weight matrix $\boldsymbol{U}$ connecting the input to the hidden layer, the weight matrix $\boldsymbol{V}$ connecting the hidden output to the output layer, the hidden layer bias $\boldsymbol{b}$ and the output layer bias $\boldsymbol{c}$ are given by

$$\boldsymbol{U} = \begin{pmatrix} -1.0 & 0.5 \\ 0.5 & 0.1 \\ 0.2 & -2.0 \end{pmatrix}, \boldsymbol{V} = \begin{pmatrix} 2.0 \\ -1.5 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}, \text{ and } \boldsymbol{c} = 0.5.$$

Find the output sequence for an input sequence of $(\boldsymbol{x}(t))_{t=1}^{4}$ where

$$\boldsymbol{x}(1) = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \boldsymbol{x}(2) = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix}, \boldsymbol{x}(3) = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}, \text{ and } \boldsymbol{x}(4) = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} \text{if}$$

(a) The RNN is of hidden-recurrence (Elman) type with the recurrence weight matrix $\boldsymbol{W}$ connecting the previous hidden output to the current hidden layer input is given by

$$\boldsymbol{W} = \begin{pmatrix} 2.0 & 1.3 \\ 1.5 & 0.0 \end{pmatrix}.$$

Assume that the hidden activations are initialized to zero.

(b) The RNN is of top-down recurrence (Jordan) type with the weight matrix $\boldsymbol{W}$ connecting the previous output to the current state of hidden layer is given by

$$\boldsymbol{W} = (2.0 \quad 1.3).$$

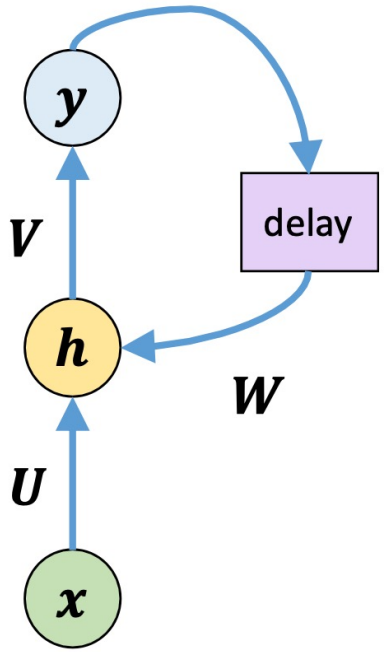
Assume that the output activations are initialized to zero.

t8q1.ipynb

Question 1

Top-down Recurrence:

$$\boldsymbol{U} = \begin{pmatrix} -1.0 & 0.5 \\ 0.5 & 0.1 \\ 0.2 & -2.0 \end{pmatrix}, \boldsymbol{V} = \begin{pmatrix} 2.0 \\ -1.5 \end{pmatrix}, \boldsymbol{W} = (2.0 \quad 1.3)$$



$$\boldsymbol{b} = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}, \text{ and } \boldsymbol{c} = 0.5$$

Three input neurons, two hidden neurons, and one output neuron.

$$\boldsymbol{h}(t) = \phi\left(\boldsymbol{U}^T\boldsymbol{x}(t) + \boldsymbol{W}^T\boldsymbol{y}(t-1) + \boldsymbol{b}\right)$$
$$\boldsymbol{y}(t) = \sigma\left(\boldsymbol{V}^T\boldsymbol{h}(t) + \boldsymbol{c}\right)$$

Initially,

$$\boldsymbol{Y}(0) = (0.0)$$

Question 1

At t=1, $\boldsymbol{x}(1) = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$:

$$\boldsymbol{h}(1) = \phi\left(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}(1) + \boldsymbol{W}^{\boldsymbol{T}}y(0) + \boldsymbol{b}\right)$$

$$= \tanh\left(\begin{pmatrix} -1.0 & 0.5 & 0.2 \\ 0.5 & 0.1 & -2.0 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} + \begin{pmatrix} 2.0 \\ 1.3 \end{pmatrix}(0) + \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}\right) = \begin{pmatrix} 0.0 \\ 0.99 \end{pmatrix}$$

$$\boldsymbol{y}(1) = \sigma\left(\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{h}(1) + \boldsymbol{c}\right) = \text{sigmoid}\left((2.0 \quad -1.5 \quad 0.2)\begin{pmatrix} 0.0 \\ 0.99 \end{pmatrix} + (0.5)\right) = (0.27)$$

Question 1

At t=2, $\boldsymbol{x}(2) = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix}$:

$$\boldsymbol{h}(2) = \phi\left(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}(2) + \boldsymbol{W}^{\boldsymbol{T}}y(1) + \boldsymbol{b}\right)$$

$$= \tanh\left(\begin{pmatrix} -1.0 & 0.5 & 0.2 \\ 0.5 & 0.1 & -2.0 \end{pmatrix}\begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix} + \begin{pmatrix} 2.0 \\ 1.3 \end{pmatrix}(0.27) + \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}\right) = \begin{pmatrix} 0.95 \\ 1.0 \end{pmatrix}$$

$$\boldsymbol{y}(2) = \sigma\left(\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{h}(2) + \boldsymbol{c}\right) = \text{sigmoid}\left((2.0 \quad -1.5 \quad 0.2)\begin{pmatrix} 0.95 \\ 1.0 \end{pmatrix} + (0.5)\right) = (0.71)$$

Similarly,

at t=3, $\boldsymbol{x}(3) = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}$; $\boldsymbol{h}(3) = \begin{pmatrix} 1.0 \\ -0.52 \end{pmatrix}$ and $\boldsymbol{y}(3) = (0.96)$

at t=4, $\boldsymbol{x}(4) = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$; $\boldsymbol{h}(4) = \begin{pmatrix} -0.36 \\ 0.98 \end{pmatrix}$ and $\boldsymbol{y}(4) = (0.16)$

Question 1

The input sequence $\left(\boldsymbol{x}(1), \boldsymbol{x}(2), \boldsymbol{x}(3), \boldsymbol{x}(4)\right)$:

$$\left(\begin{pmatrix}1\\2\\-1\end{pmatrix} \quad \begin{pmatrix}-1\\1\\-2\end{pmatrix} \quad \begin{pmatrix}0\\3\\1\end{pmatrix} \quad \begin{pmatrix}2\\-1\\0\end{pmatrix}\right)$$

The output sequence $\left(\boldsymbol{y}(1), \boldsymbol{y}(2), \boldsymbol{y}(3), \boldsymbol{y}(4)\right)$:

$$((0.27) \quad (0.71) \quad (0.96) \quad (0.16))$$

Question 2

2. An RNN receives 3-dimensional input sequence and produces 2-dimensional output sequence. It has 5 hidden neurons and receives sequences of 100 time steps.

   - Generate 8 training input sequences by drawing values uniformly between 0 and 1.0.
   - If the sequence $\left(\boldsymbol{y}(t)\right)_{t=1}^{100}$ where $\boldsymbol{y}(t) = \left(y_k(t)\right)_{k=1}^{2} \in \boldsymbol{R}^2$ denotes the output sequence for an input sequence $\left(\boldsymbol{x}(t)\right)_{t=1}^{100}$ where $\boldsymbol{x}(t) = \left(x_i(t)\right)_{i=1}^{3} \in \boldsymbol{R}^3$, generate the corresponding output sequences for the input training sequences as follows:

$$y_1(t) = 5x_1(t) - 0.2x_3(t-1) + 0.1\varepsilon$$
$$y_2(t) = 25x_2(t-1)x_3(t-3) + 0.1\varepsilon$$

   where ε is standard normally distributed random variable.

Train an RNN to learn the above sequences by using gradient descent learning. Use a learning factor $\alpha = 0.01$ and Adam optimizer.

t8q2.ipynb

Question 2

Set random seed and initialize hyper-parameters

```
[2]  # Set random seed for reproducibility
     seed = 10
     np.random.seed(seed)
     torch.manual_seed(seed)
```

```
<torch._C.Generator at 0x7eeddd747dd0>
```

```
[3]  # Initialize hyper-parameters
     n_in = 3
     n_hidden = 5
     n_out = 2
     n_steps = 100
     n_seqs = 8
     n_iters = 10000
     lr = 0.01
```

Generate training data

```
[4]  x_train = np.random.rand(n_seqs, n_steps, n_in)
     y_train = np.zeros([n_seqs, n_steps, n_out])

     y_train[:, 1:, 0] = 5 * x_train[:, 1:, 0] - 0.2 * x_train[:, :-1, 2]
     y_train[:, 3:, 1] = 25 * x_train[:, 2:-1, 1] * x_train[:, :-3, 2]
     y_train += 0.1 * np.random.randn(n_seqs, n_steps, n_out)

     x_train = torch.tensor(x_train, dtype=torch.float32)
     y_train = torch.tensor(y_train, dtype=torch.float32)
```

$$\left.\begin{aligned} y_1(t) &= 5x_1(t) - 0.2x_3(t-1) + 0.1\varepsilon \\ y_2(t) &= 25x_2(t-1)x_3(t-3) + 0.1\varepsilon \end{aligned}\right\}$$

Question 2

The basic slice syntax is i:j:k where *i* is the starting index, *j* is the stopping index, and *k* is the step.

y_train[:,3:,1] = [$y_2(3), y_2(4), y_2(5) \ \ldots, y_2(99)$]

x_train[:,2:-1,1] = [$x_2(2), x_2(3), x_2(4) \ \ldots, x_2(98)$]

x_train[:, :-3,2] = [$x_3(0), x_3(1), x_3(2) \ \ldots, x_3(96)$]

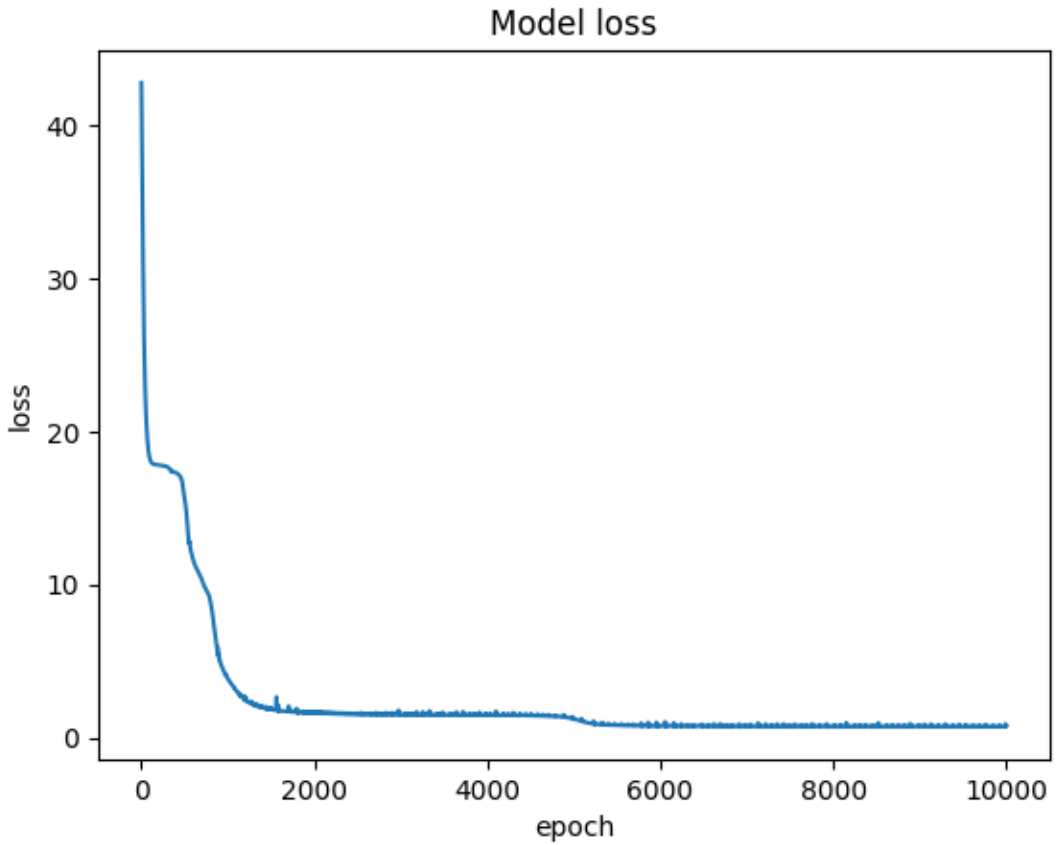
Assume *n* is the number of elements in the dimension being sliced

If *j* is not given it defaults to *n*

If *k* is not given it defaults to 1

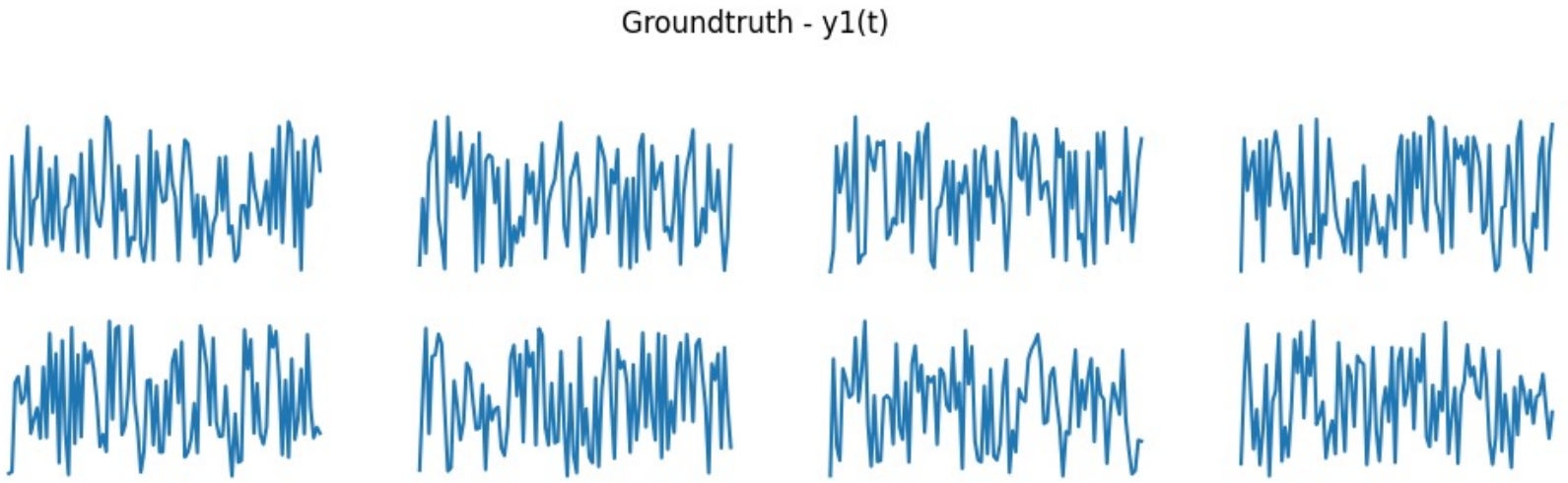
Negative *i* and *j* are interpreted as *n* + *i* and *n* + *j* where *n* is the number of elements in the corresponding dimension

Question 2



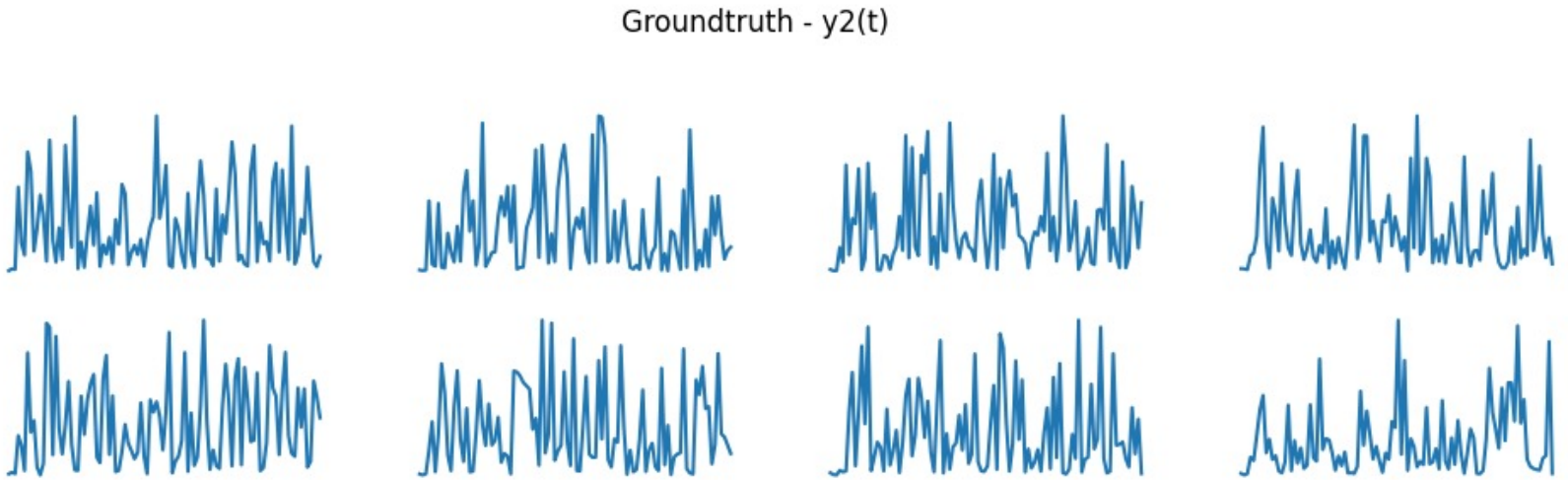
Question 2

$$y_1(t) = 5x_1(t) - 0.2x_3(t-1) + 0.1\varepsilon$$

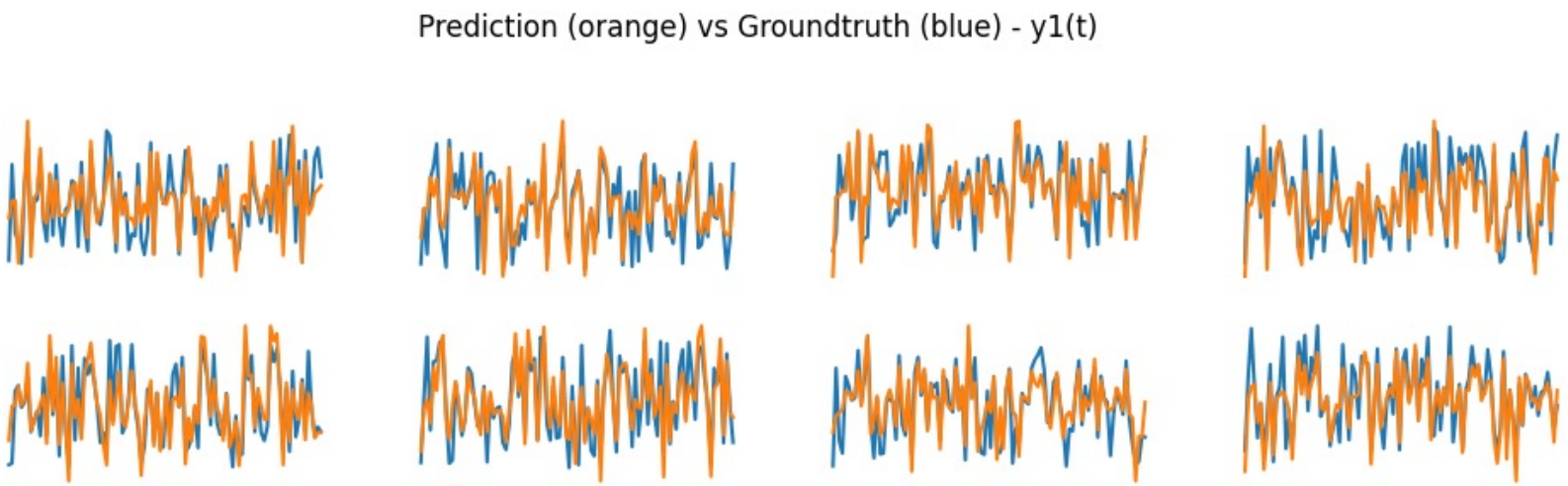


Question 2

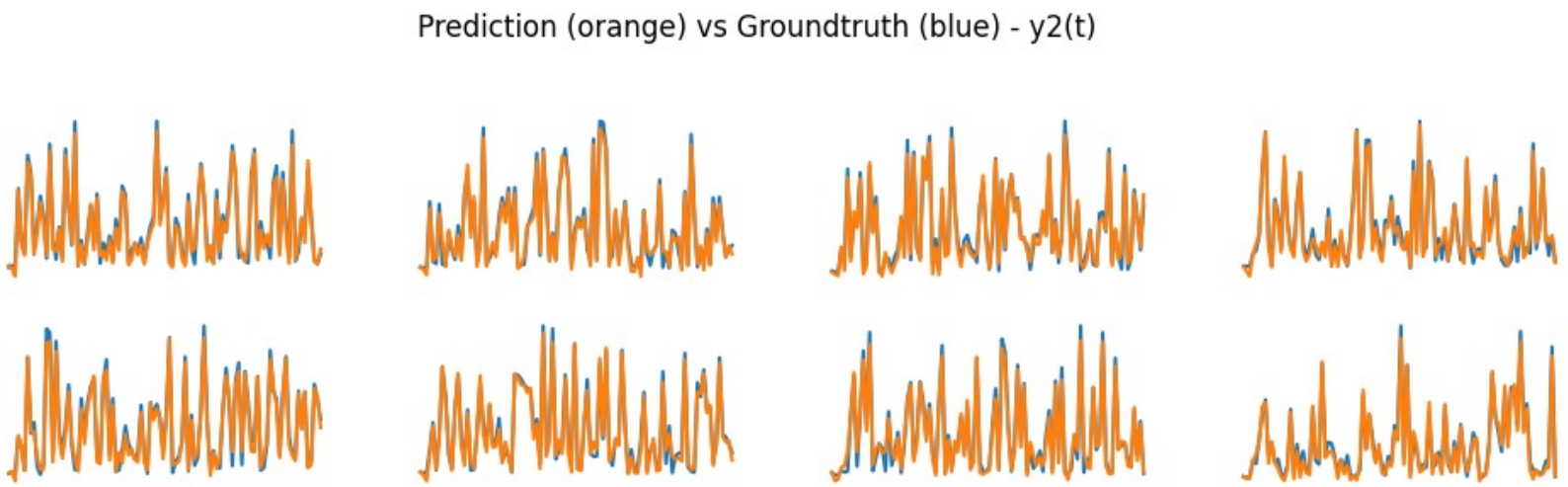
$$y_2(t) = 25x_2(t-1)x_3(t-3) + 0.1\varepsilon$$



Question 2



Question 2



Question 3

3. Design an LSTM layer with 10 units to map the following input and output sequences:

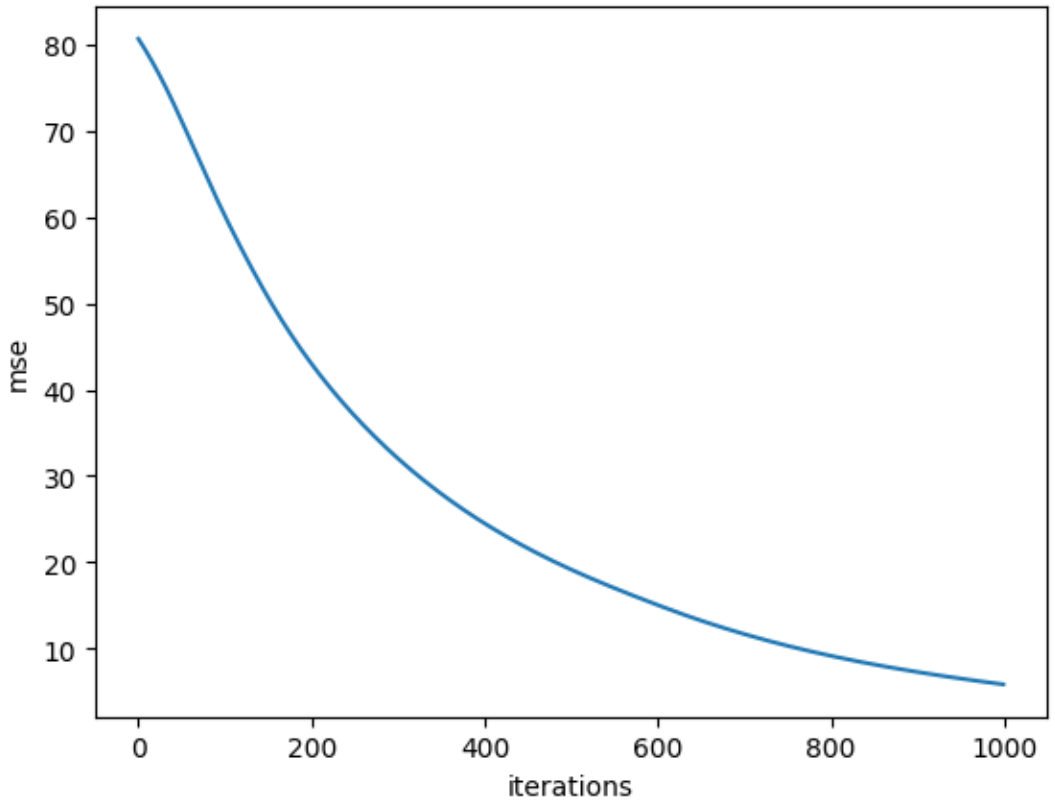
Input $\boldsymbol{x}$	Output $\boldsymbol{y}$
(1 2 5 6)	(1 3 7 11)
(5 7 7 8)	(5 12 14 15)
(3 4 5 7)	(3 7 9 12)

Plot the learning curves at a rate $\alpha = 0.001$.

Find the output sequences for the following input sequences:

(1 2 3 4)
(4 5 6 7)

Question 3



Question 3

Find the output sequences for the following input sequences:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \end{pmatrix}$$

```
test_x = torch.tensor(
    [[[1], [2], [3], [4]],  # 1, 3, 5, 7
    [[4], [5], [6], [7]]], dtype=torch.float32)  # 4, 9, 11, 13
out = predictor(test_x)
print(out)
```

```
tensor([[0.9988, 2.7404, 6.3064, 8.8883], [4.6289,
8.4632, 9.4166, 9.5583]], grad_fn=<SqueezeBackward1>)
```

Question 4

4. Design a character RNN layer with 10 GRU units to learn the sentiments about a movie, given in table 1.

Convert the input text into character ids and set the maximum text length to be 40. Train the network using gradient descent learning with the Adam optimizer and at a learning rate $\alpha = 0.001$. Use dropouts at the hidden layer of GRU at a probability $p = 0.7$.

Plot the cost and the accuracies against epochs during training.

Question 4

Table 1

Input	*Sentiment*
I did not like the movie	negative
The movie was not good	negative
I watched the movie with great interest	positive
I have seen better movies	negative
Good to see that movie	positive
I am not a fan of movies	negative
I liked the movie great	positive
The movie was of interest to me	positive
I thought they could show interesting scenes	negative
The movie did not have good scenes	negative
Family did not like the movie at all	negative

After training, find the likelihoods of the sentiments of the following statements:

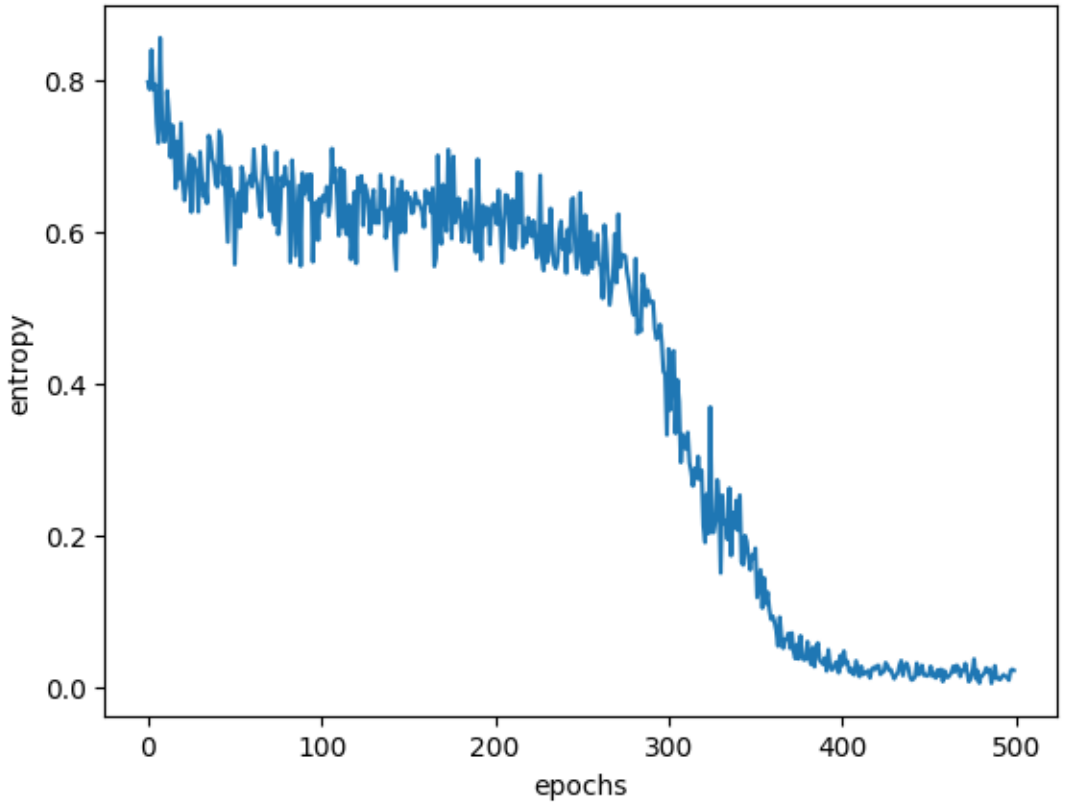
The movie was not interesting to me

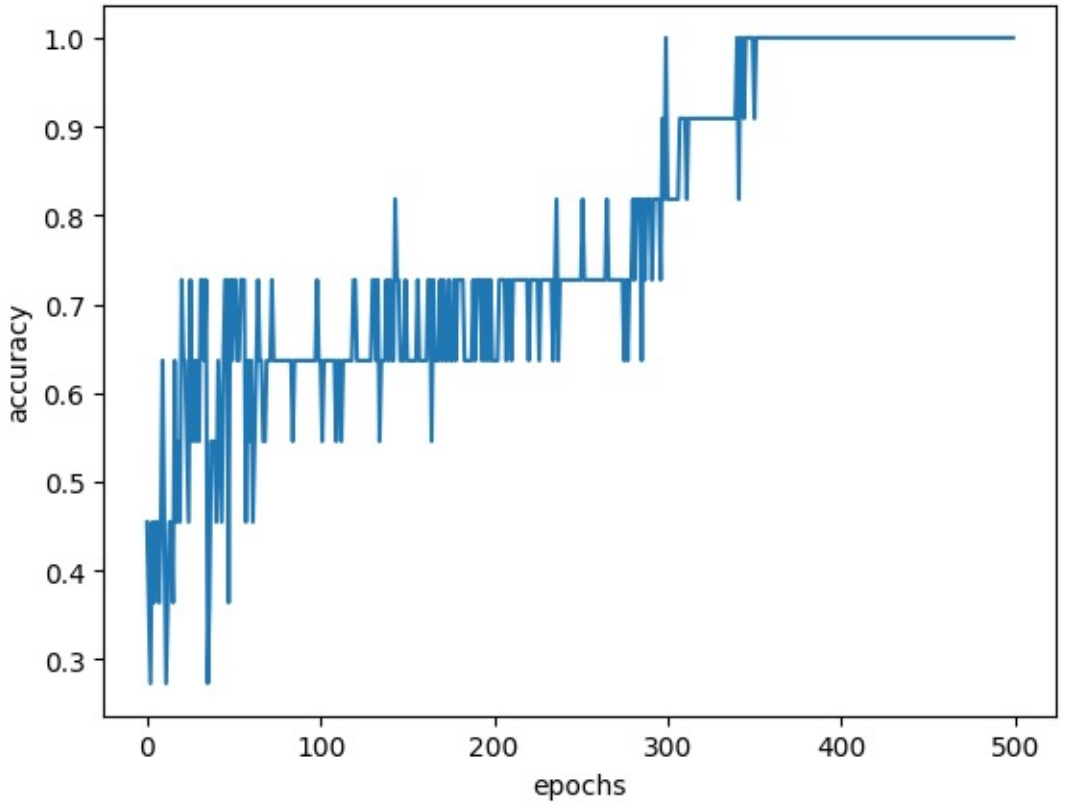
I liked the movie with great interest

Question 4

- Input: ‘Good to see that movie’
- Translated input as a list of ids:
  [7 14 14 4 0 17 14 0 16 5 5 0 17 8 1 17 0 12 14 19 9 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
- Targets:
  [0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0]
- Input text -> preprocess to max length -> convert to IDs

Question 4


entropy
0.0
0.2
0.4
0.6
0.8
0
100
200
300
400
500
epochs


accuracy
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0
100
200
300
400
500
epochs

Question 4

```
test_data = ['The movie was not interesting to me',
'I liked the movie with great interest']
```

```
test_logits = model(x_test, drop_rate=0)
probs = nn.functional.softmax(test_logits, dim=1)
print(probs)
```

```
tensor(
[[0.0119, 0.9881],
[0.4359, 0.5641]])
```

‘The movie was not interesting to me’ is a negative sentiment with a probability of 0.9881

‘I liked the movie with great interest’ is a positive sentiment with a probability of 0.4359 (hasn’t trained well)