

# DEEP LEARNING

## Mạng nơ ron hồi quy (RNN)

Phạm Nguyên Khang  
pnkhang@cit.ctu.edu.vn

CAN THO, 22/12/2022

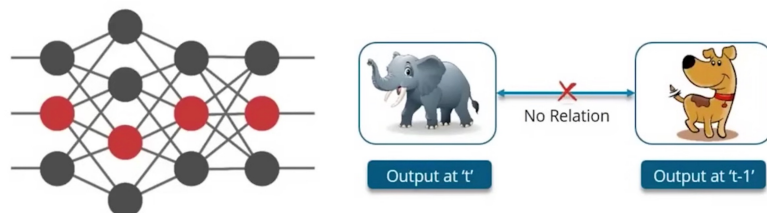
## Nội dung

- Hạn chế của mạng truyền thẳng
- Mạng RNN là gì?
- Các vấn đề của mạng RNN
- Suy giảm và bùng nổ gradient
- Mạng Bộ nhớ ngắn hạn dài (LSTM)

1

## Hạn chế của mạng truyền thẳng

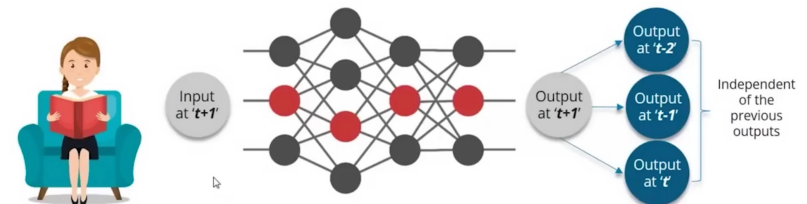
- Mạng truyền thẳng (feedforward network) được dùng để dự đoán nhãn của một tập ảnh ngẫu nhiên
- Kết quả của việc dự đoán lần thứ nhất, không ảnh hưởng đến kết quả dự đoán lần thứ hai



2

## Hạn chế của mạng truyền thẳng

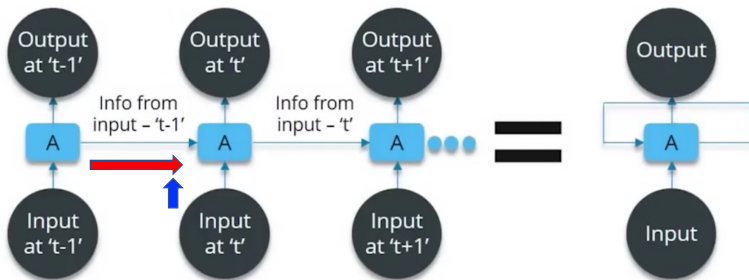
- Khi đọc sách, bạn hiểu được nhờ vào sự hiểu biết của bạn về các từ trước đó
- Không thể dự đoán được từ tiếp theo trong một câu nếu dùng mạng truyền thẳng



3

## Giải pháp

- Kết quả dự báo của **đầu ra tại thời điểm  $t-1$**  kết hợp với **đầu vào tại thời điểm  $t$**  được dùng để dự báo kết quả **đầu ra tại thời điểm  $t$** .



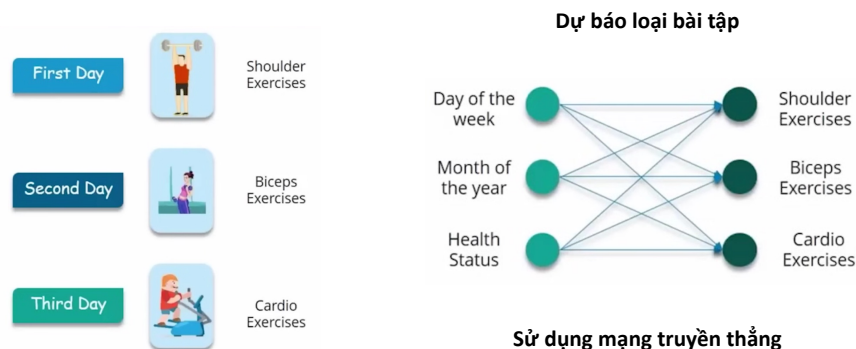
4

## Mạng RNN

- Mạng RNN là mạng nơ ron được thiết kế để nhận các mẫu (pattern) trong mỗi chuỗi dữ liệu (sequences of data), như: văn bản, gien, chữ viết tay, giọng nói, time series, giá cổ phiếu, ...
- Ví dụ:
  - Gym PT lập thời khoá biểu tập luyện cho bạn. Các bài tập được lặp lại sau mỗi 3 ngày.

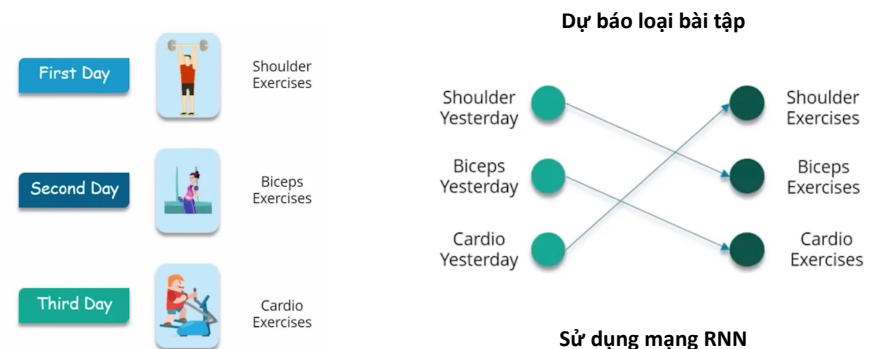
5

## Mạng RNN



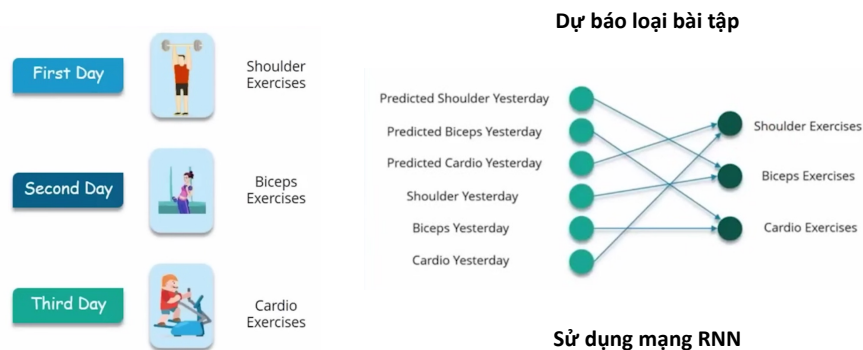
6

## Mạng RNN



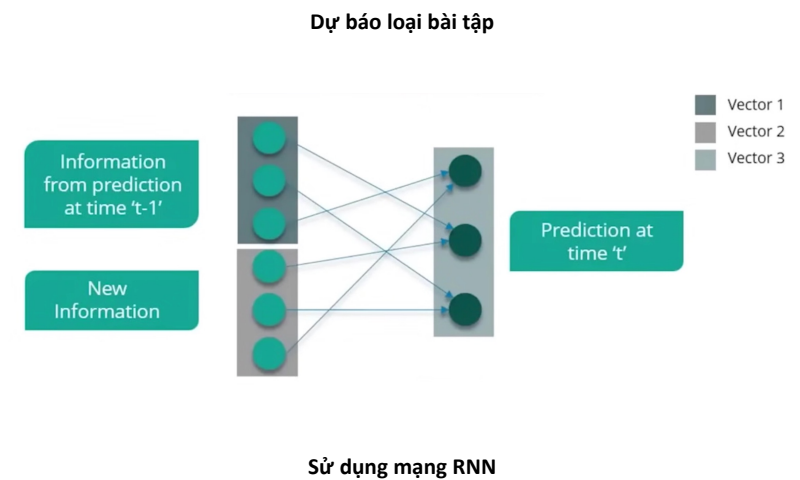
7

## Mạng RNN



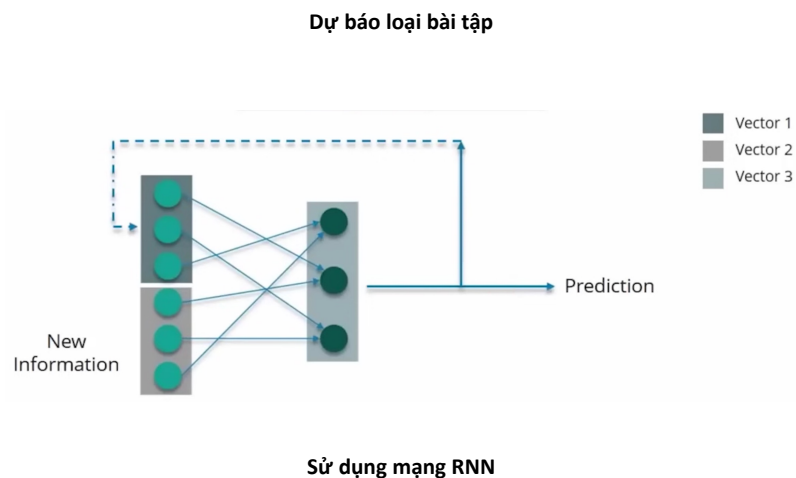
8

## Mạng RNN



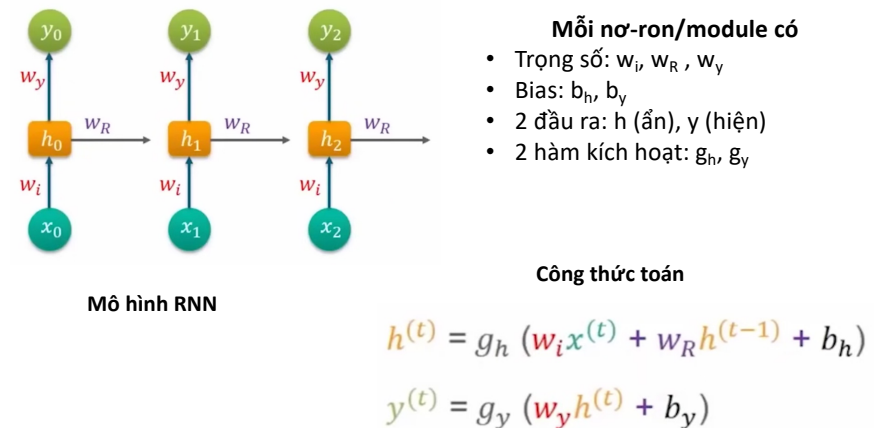
9

## Mạng RNN



10

## Mạng RNN



11

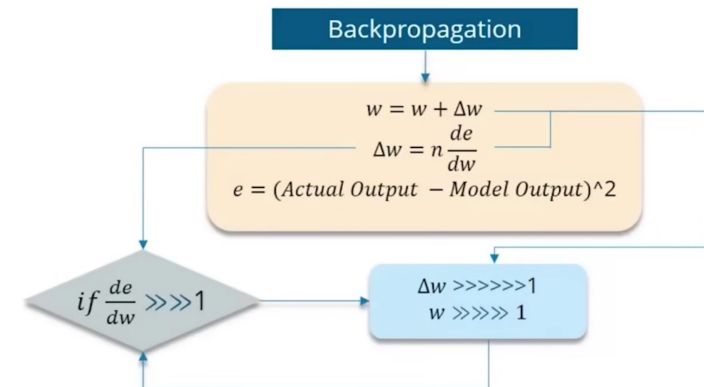
## Huấn luyện mạng RNN

- Sử dụng thuật toán lan truyền ngược theo thời gian (Backpropagation Through Time - BTT)
  - Áp dụng lan truyền ngược tại mỗi thời điểm  $t$
- Vấn đề



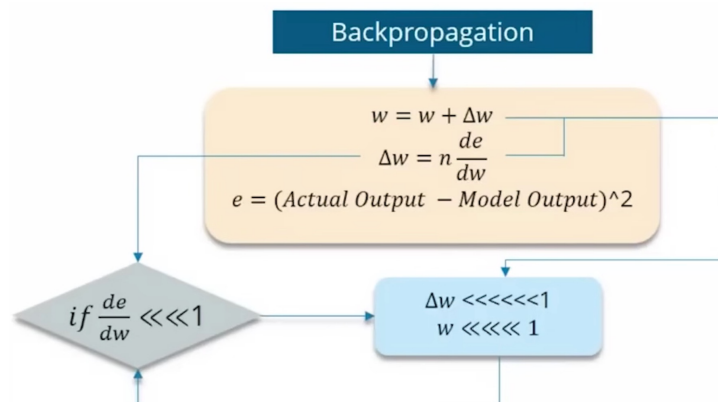
12

## Suy giảm gradient



13

## Bùng nổ Gradient



14

## Giải pháp

### Bùng nổ gradient

- Truncated BTT
  - Thay vì sử dụng lan truyền ngược từ thời điểm cuối, ta chọn thời điểm sớm hơn chút, ví dụ sau: 10 thời điểm
- Cắt gradient khi vượt quá ngưỡng
- Dùng RMSprop điều chỉnh tốc độ học

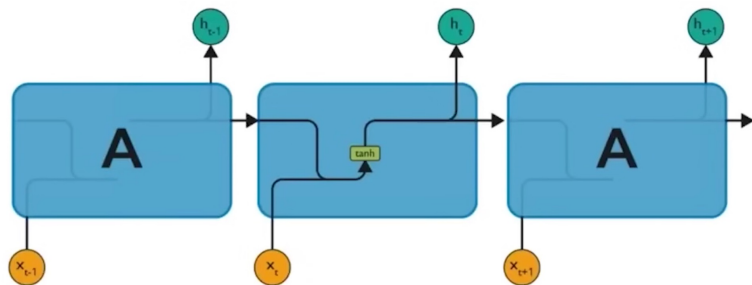
### Suy giảm gradient

- Dùng hàm kích hoạt ReLU
  - Gradient = 1
- Cắt gradient khi quá thấp
- Dùng RMSprop
- Sử dụng mạng LTMS hay GRU

15

## Mạng LSTM

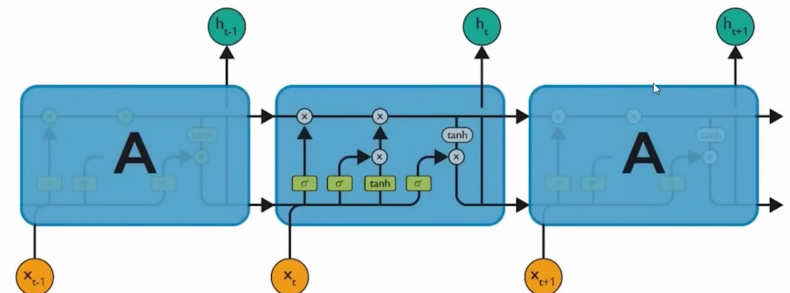
- Long Short-Term Memory (Bộ nhớ ngắn hạn lớn/dài)
  - Có khả năng học được các sự phụ thuộc dài hạn (long-term)



16

## Mạng LSTM

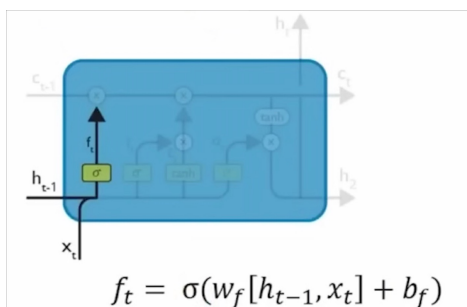
- Long Short-Term Memory (Bộ nhớ ngắn hạn lớn/dài)
  - Có khả năng học được các sự phụ thuộc dài hạn (long-term)



17

## Mạng LSTM

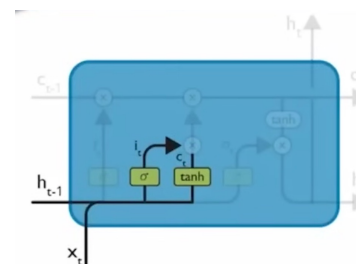
- Bước 1:
  - Xác định thông tin không cần thiết (cần quên đi) bằng cách sử dụng tầng sigmoid (cổng quên)
    - $f_t \sim 0$  quên,  $f_t \sim 1$  giữ lại



18

## Mạng LSTM

- Bước 2:
  - Xác định thông tin cần thiết để lưu lại
  - Dùng 2 tầng: tầng *sigmoid* (cổng vào) và tầng *tanh* tạo ra vector ứng viên mới.



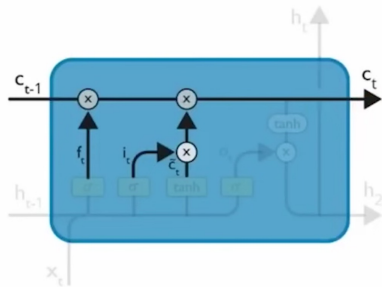
$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c)$$

19

## Mạng LSTM

- Bước 3:
  - Cập nhật trạng thái mới từ trạng thái cũ ( $c_{t-1}$ ) và ứng viên mới ( $\tilde{c}_t$ )

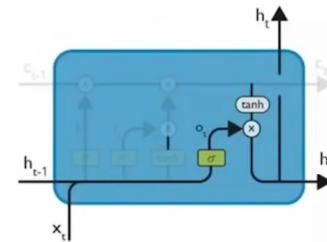


$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

20

## Mạng LSTM

- Bước 4:
  - Xác định đầu ra  $h_t$ 
    - Đưa đầu ra trước đó  $h_{t-1}$  và đầu vào hiện tại  $x_t$  qua cổng sigmoid
    - Nhân kết quả với tanh (trạng thái hiện tại -  $c_t$ )



$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

21

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv

Index	Features	Format
1	Date Time	01.01.2009 00:10:00
2	p (mbar)	996.52
3	T (degC)	-8.02
4	Tpot (K)	265.4
5	Tdew (degC)	-8.9
6	rh (%)	93.3

7	VPmax (mbar)	3.33
8	VPact (mbar)	3.11
9	VPdef (mbar)	0.22
10	sh (g/kg)	1.94
11	H2OC (mmol/mol)	3.12
12	rho (g/m ** 3)	1307.75
13	wv (m/s)	1.03
14	max. wv (m/s)	1.75
15	wd (deg)	152.3

22

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv
  - Import các thư viện cần thiết

```
import tensorflow as tf
from keras.layers import LSTM, Dense
import pandas as pd
```

23

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv
  - Đọc dữ liệu

```
csv_file = "/Users/pnkhang/DL/climate.csv"

#Đọc dữ liệu, phân cách bằng dấu phẩy
df = pd.read_csv(csv_file, sep=',')

# Lấy cột nhiệt độ (cột thứ 2)
data = df.iloc[:, [2]].values
print(data)
```

24

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv
  - Chuẩn hoá dữ liệu

```
#Chuẩn hoá dữ liệu mean = 0, std = 1
def normalize(data, train_split):
    data_mean = data[:train_split].mean(axis=0)
    data_std = data[:train_split].std(axis=0)
    return (data - data_mean) / data_std
```

```
#Chuẩn hoá và tách dữ liệu huấn luyện
train_split = int(0.715 * int(df.shape[0]))

data = normalize(data, train_split)

train_data = data[:train_split]
val_data = data[train_split:]
```

25

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv
  - Tạo dữ liệu huấn luyện

```
#Xác định phương thức dự báo
past = 720
future = 72
step = 6
batch_size = 256
```

```
start = past + future
end = start + train_split

x_train = train_data
y_train = data[start:end]
sequence_length = int(past / step)
```

26

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv
  - Tạo dữ liệu huấn luyện

```
#Xác định phương thức dự báo
past = 720
future = 72
step = 6
batch_size = 256
```

```
# Tạo dữ liệu huấn luyện
dataset_train =
tf.keras.preprocessing.timeseries_dataset_from_array(
    x_train,
    y_train,
    sequence_length=sequence_length,
    sampling_rate=step,
    batch_size=batch_size,
)
```

27

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó

- Tập dữ liệu: climate.csv

- Tạo dữ liệu kiểm chứng

```
x_end = len(val_data) - past - future
label_start = train_split + past + future
x_val = val_data[:x_end]
y_val = data[label_start:]
```

```
dataset_val =
tf.keras.preprocessing.timeseries_dataset_from_array(
    x_val,
    y_val,
    sequence_length=sequence_length,
    sampling_rate=step,
    batch_size=batch_size,
)
```

28

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó

- Tập dữ liệu: climate.csv

- Tạo mô hình

```
model = tf.keras.Sequential()
model.add(Input(shape=(sequence_length, 1)))
model.add(LSTM(32))
model.add(Dense(1))

model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=learning_rate), loss="mse")
model.summary()
```

29

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó

- Tập dữ liệu: climate.csv

- Tạo mô hình

```
model = tf.keras.Sequential()
model.add(Input(shape=(sequence_length, 1)))
model.add(LSTM(32))
model.add(Dense(1))

model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=learning_rate), loss="mse")
model.summary()
```

30

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó

- Tập dữ liệu: climate.csv

- Tạo mô hình

```
model = tf.keras.Sequential()
model.add(Input(shape=(sequence_length, 1)))
model.add(LSTM(32))
model.add(Dense(1))

model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=learning_rate), loss="mse")
model.summary()
```

Layer (type)	Output Shape	Param #
Lstm (LSTM)	(None, 32)	4352
dense (Dense)	(None, 1)	33

Total params: 4,385  
Trainable params: 4,385  
Non-trainable params: 0

31



## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv
  - Huấn luyện

```
# Huấn luyện
epochs = 10
history = model.fit(
    dataset_train,
    epochs=epochs,
    validation_data=dataset_val,
    #callbacks=[es_callback, modelckpt_callback],
)
```

32

## Ví dụ 1 – Dự báo nhiệt độ

```
1172/1172 [=====] - ETA: 0s - loss: 0.21532023-03-13 12:30:47.934179:
2023-03-13 12:30:47.985474: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_regis
1172/1172 [=====] - 61s 51ms/step - loss: 0.2153 - val_loss: 0.1596
Epoch 2/10
1172/1172 [=====] - 58s 50ms/step - loss: 0.1426 - val_loss: 0.1623
Epoch 3/10
1172/1172 [=====] - 59s 50ms/step - loss: 0.1363 - val_loss: 0.1828
Epoch 4/10
1172/1172 [=====] - 59s 51ms/step - loss: 0.1357 - val_loss: 0.1852
Epoch 5/10
1172/1172 [=====] - 59s 50ms/step - loss: 0.1360 - val_loss: 0.1810
Epoch 6/10
1172/1172 [=====] - 60s 51ms/step - loss: 0.1338 - val_loss: 0.1791
Epoch 7/10
1172/1172 [=====] - 60s 52ms/step - loss: 0.1324 - val_loss: 0.1793
Epoch 8/10
1172/1172 [=====] - 61s 52ms/step - loss: 0.1294 - val_loss: 0.1805
Epoch 9/10
1172/1172 [=====] - 60s 51ms/step - loss: 0.1299 - val_loss: 0.1769
Epoch 10/10
1172/1172 [=====] - 61s 52ms/step - loss: 0.1278 - val_loss: 0.1733
```

33

## Ví dụ 1 – Dự báo nhiệt độ

- Bài toán: Dự báo nhiệt độ của ngày thứ 6, dựa vào dữ liệu của 5 ngày trước đó
  - Tập dữ liệu: climate.csv
  - Vẽ đồ thị

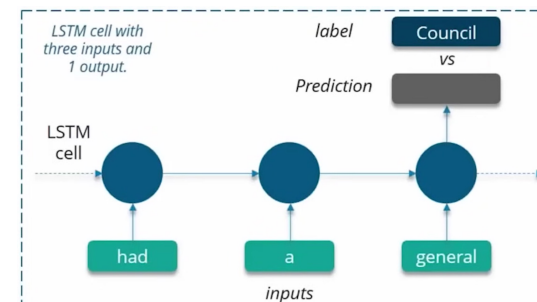
```
def visualize_loss(history, title):
    loss = history.history["loss"]
    val_loss = history.history["val_loss"]
    epochs = range(len(loss))
    plt.figure()
    plt.plot(epochs, loss, "b", label="Training loss")
    plt.plot(epochs, val_loss, "r", label="Validation loss")
    plt.title(title)
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()
    plt.show()
```

visualize\_loss(history, "Training and Validation Loss")

34

## Ví dụ 2 – Sinh từ

- Bài toán
  - Dự báo từ kế tiếp dựa vào 3 từ trước đó



- LSTM cell có:
- 3 input (timestep = 3)
  - 1 output

35

## Ví dụ 2 – Sinh từ

- Huấn luyện
  - Chuẩn bị dữ liệu

Trích từ truyện ngụ ngôn  
của Aesop

Có tất cả 112 từ khác nhau.

long ago, the mice had a general council to consider what measures they could take to outwit their common enemy, the cat. some said this, and some said that but at last a young mouse got up and said he had a proposal to make, which he thought would meet the case. you will all agree, said he, that our chief danger consists in the sly and treacherous manner in which the enemy approaches us. now, if we could receive some signal of her approach, we could easily escape from her. i venture, therefore, to propose that a small bell be procured, and attached by a ribbon round the neck of the cat. by this means we should always know when she was about, and could easily retire while she was in the neighborhood. this proposal met with general applause, until an old mouse got up and said that is all very well, but who is to bell the cat? the mice looked at one another and nobody spoke. then the old mouse said it is easy to propose impossible remedies.

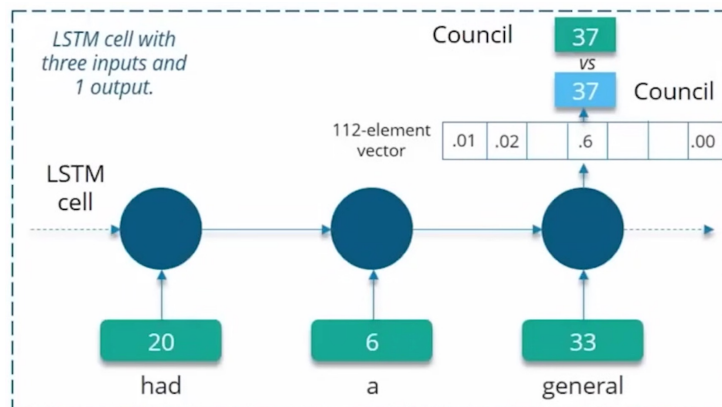
36

## Ví dụ 2 – Sinh từ

- Huấn luyện
  - Chuẩn bị dữ liệu
    - Thu thập dữ liệu: 1 đoạn văn bản
    - Gán cho mỗi từ 1 số nguyên (id)

37

## Ví dụ 2 – Sinh từ



38

## Ví dụ 2 – Sinh từ

- Import

```
import tensorflow as tf
import numpy as np
import collections
from keras.layers import LSTM, Dense
```

39

## Ví dụ 2 – Sinh từ

- Hàm đọc file

```
def read_data(fname):  
    with open(fname) as f:  
        content = f.readlines()  
        content = [x.strip() for x in content]  
        words = []  
        for line in content:  
            words.extend(line.split())  
  
    return np.array(words)
```

40

## Ví dụ 2 – Sinh từ

- Hàm tạo từ điển

```
def build_dataset(words):  
    count = collections.Counter(words).most_common()  
    word2id = {}  
    for word, freq in count:  
        word2id[word] = len(word2id)  
  
    id2word = dict(zip(word2id.values(), word2id.keys()))  
    return word2id, id2word
```

41

## Ví dụ 2 – Sinh từ

- Đọc dữ liệu và tạo từ điển

```
data = read_data('/Users/pnkhong/DL/toto.txt')  
  
print(data)  
  
w2i, i2w = build_dataset(data)  
  
vocab_size = len(w2i)  
timestep = 3
```

42

## Ví dụ 2 – Sinh từ

- Tạo dữ liệu huấn luyện

```
X, Y = [], []  
for i in range(timestep, len(data)):  
    X.append([w2i[data[k]] for k in range(i-timestep, i)])  
    Y.append(w2i[data[i]])
```

```
encoded_data = [w2i[x] for x in data]  
X = encoded_data[:-1]  
Y = encoded_data[timestep:]  
  
train_data = tf.keras.preprocessing.timeseries_dataset_from_array(  
    X, Y, sequence_length=timestep, sampling_rate=1  
)
```

43

## Ví dụ 2 – Sinh từ

- Tạo mô hình và huấn luyện

```
model = tf.keras.Sequential()
model.add(LSTM(512, return_sequences=True,
               input_shape=(timestep, 1)))
model.add(LSTM(512, return_sequences=False))
model.add(Dense(vocab_size))

model.summary()
```

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

model.fit(train_data, epochs=500)
```

44

## Ví dụ 2 – Sinh từ

- Tạo mô hình và huấn luyện

```
model = tf.keras.Sequential()
model.add(LSTM(512, return_sequences=True,
               input_shape=(timestep, 1)))
model.add(LSTM(512, return_sequences=False))
model.add(Dense(vocab_size))

model.summary()
```

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

model.fit(train_data, epochs=500)
```

45

## Ví dụ 2 – Sinh từ

- Tạo mô hình và huấn luyện

```
2/2 [=====] - 0s 19ms/step - loss: 0.1252 - accuracy: 0.9261
Epoch 496/500
2/2 [=====] - 0s 21ms/step - loss: 0.1260 - accuracy: 0.9489
Epoch 497/500
2/2 [=====] - 0s 20ms/step - loss: 0.1105 - accuracy: 0.9375
Epoch 498/500
2/2 [=====] - 0s 19ms/step - loss: 0.1145 - accuracy: 0.9659
Epoch 499/500
2/2 [=====] - 0s 20ms/step - loss: 0.1070 - accuracy: 0.9489
Epoch 500/500
2/2 [=====] - 0s 24ms/step - loss: 0.0999 - accuracy: 0.9602
```

46

## Ví dụ 2 – Sinh từ

- Dự báo

```
def encode(sent):
    return [[w2i[w] for w in sent.split()]]

pred = model.predict(encode("had a general"))
pred_word = i2w[np.argmax(pred)]
print(pred_word)

pred = model.predict(encode("a general
council"))
pred_word = i2w[np.argmax(pred)]
print(pred_word)
```

```
2023-03-13 13:13:11
council
to
```

47

## Thực hành 1

- Làm lại bài tập dự báo thời tiết và bài tập sinh từ

48

## Thực hành 2

- Mở rộng bài toán dự báo thời tiết với nhiều cột dữ liệu hơn

- p (mbar)
- T (degC)
- rh (%)
- VPact (mbar)
- VPdef (mbar)
- H2OC (mmol/mol)
- rho (g/m \*\* 3)

```
csv_file = "/Users/pnkhang/DL/climate.csv"

#Đọc dữ liệu, phân cách bằng dấu phẩy
df = pd.read_csv(csv_file, sep=',')

# Lấy cột nhiệt độ (cột thứ 2)
data = df.iloc[:, 2].values
print(data)
```

- Gợi ý: Thay chỗ dữ liệu [2] bằng mảng [2, 3, ... ]

49

## Thực hành 3

- Thu tập một tập dữ liệu văn bản lớn hơn. Huấn luyện mô hình.
- Tạo giao diện cho phép nhập vào 3 từ và sinh ra từ 10 từ kế tiếp từ 3 từ nhập vào:
  - Lấy 3 từ nhập vào sinh ra từ thứ 4.
  - Lấy các từ 2, 3, 4 sinh ra từ thứ 5
  - Cứ như thế cho đến từ thứ 10.

50

# THANK YOU



51