

## CÂU HỎI ÔN TẬP

### Yêu cầu

- Sinh viên IN QUYỂN bao gồm: code, kết quả chạy, giải thích bằng word
- Giải thích bằng viết tay trên quyển tài liệu. Có thể kèm theo mẫu giấy viết giải thích. Tô màu code, kết quả....
- Càng viết TAY nhiều càng điểm cao (chống copy + ChatGPT 🙅)

CHÚC CÁC BẠN MỘT MÙA THI ĐẠT KẾT QUẢ TỐT!

=====

1. Trình bày hiểu biết của mình về tensorflow và 5 ví dụ minh họa (không có trong Bài tập 3). Tham khảo các link dưới đây  
[https://www.geeksforgeeks.org/introduction-to-tensorflow/TensorFlow Tutorial, CME 323, 4 12 2018.pdf \(stanford.edu\)](https://www.geeksforgeeks.org/introduction-to-tensorflow/TensorFlow Tutorial, CME 323, 4 12 2018.pdf (stanford.edu))
2. Chạy code sau đây và giải thích *Linear Regression model* using TensorFlow Core API.

```
# importing the d, giải thichdependencies
import tensorflow.compat.v1 as tf
import numpy as np
import matplotlib.pyplot as plt

# Model Parameters
learning_rate = 0.01
training_epochs = 2000
display_step = 200

# Training Data
train_X =
np.asarray([3.3,4.4,5.5,6.71,6.93,4.168,9.779,6.182,7.59,2.167,
7.042,10.791,5.313,7.997,5.654,9.27,3.1])
train_y =
np.asarray([1.7,2.76,2.09,3.19,1.694,1.573,3.366,2.596,2.53,1.22
1,
2.827,3.465,1.65,2.904,2.42,2.94,1.3])
n_samples = train_X.shape[0]

# Test Data
test_X = np.asarray([6.83, 4.668, 8.9, 7.91, 5.7, 8.7, 3.1,
2.1])

# Set placeholders for feature and target vectors
X = tf.placeholder(tf.float32)
y = tf.placeholder(tf.float32)

# Set model weights and bias)
```

```

test_y = np.asarray([1.84, 2.273, 3.2, 2.831, 2.92, 3.24, 1.35,
1.03])
W = tf.Variable(np.random.randn(), name="weight")
b = tf.Variable(np.random.randn(), name="bias")

# Construct a linear model
linear_model = W*X + b

# Mean squared error
cost = tf.reduce_sum(tf.square(linear_model - y)) /
(2*n_samples)

# Gradient descent
optimizer =
tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)

# Initializing the variables
init = tf.global_variables_initializer()

# Launch the graph
with tf.Session() as sess:
    # Load initialized variables in current session
    sess.run(init)
    # Fit all training data
    for epoch in range(training_epochs):

        # perform gradient descent step
        sess.run(optimizer, feed_dict={X: train_X, y: train_y})

        # Display logs per epoch step
        if (epoch+1) % display_step == 0:
            c = sess.run(cost, feed_dict={X: train_X, y:
train_y})
            print("Epoch:{0:6} \t Cost:{1:10.4} \t W:{2:6.4} \t
b:{3:6.4}".
                    format(epoch+1, c, sess.run(W), sess.run(b)))

        # Print final parameter values
        print("Optimization Finished!")
        training_cost = sess.run(cost, feed_dict={X: train_X, y:
train_y})
        print("Final training cost:", training_cost, "W:",
sess.run(W), "b:",
            sess.run(b), '\n')

    # Graphic display
    plt.plot(train_X, train_y, 'ro', label='Original data')

```

```

plt.plot(train_X, sess.run(W) * train_X + sess.run(b),
label='Fitted line')
plt.legend()
plt.show()

# Testing the model
testing_cost = sess.run(tf.reduce_sum(tf.square(linear_model
- y)) / (2 * test_X.shape[0])),
                        feed_dict={X: test_X, y: test_y})

print("Final testing cost:", testing_cost)
print("Absolute mean square loss difference:",
abs(training_cost - testing_cost))

# Display fitted line on test data
plt.plot(test_X, test_y, 'bo', label='Testing data')
plt.plot(train_X, sess.run(W) * train_X + sess.run(b),
label='Fitted line')
plt.legend()
plt.show()

```

3. Từ tensorflow đến keras. Trình bày hiểu biết của mình về keras và 5 ví dụ minh họa. Tham khảo <https://www.activestate.com/resources/quick-reads/what-is-a-keras-model/>
4. Chạy 3 ví dụ dưới đây và giải thích các dòng lệnh theo hiểu biết của mình

=====

```

import tensorflow as tf
import numpy as np
#print("TensorFlow version:", tf.__version__)

mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10)
])
predictions = model(x_train[:1])
predictions
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
loss_fn(y_train[:1], predictions)
model.compile(optimizer='adam',
              loss=loss_fn,

```

```
        metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test, verbose=2)
```

=====

<https://www.tensorflow.org/tutorials/images/cnn>

```
import tensorflow as tf
```

```
from tensorflow.keras import datasets, layers, models
```

```
import matplotlib.pyplot as plt
```

```
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()
```

```
# Normalize pixel values to be between 0 and 1
```

```
train_images, test_images = train_images / 255.0, test_images / 255.0
```

```
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
               'dog', 'frog', 'horse', 'ship', 'truck']
```

```
plt.figure(figsize=(10,10))
```

```
for i in range(25):
```

```
    plt.subplot(5,5,i+1)
```

```
    plt.xticks([])
```

```
    plt.yticks([])
```

```
    plt.grid(False)
```

```
    plt.imshow(train_images[i])
```

```
    # The CIFAR labels happen to be arrays,
```

```
    # which is why you need the extra index
```

```
    plt.xlabel(class_names[train_labels[i][0]])
```

```
plt.show()
```

=====

<https://www.tensorflow.org/tutorials/keras/classification>

```
import tensorflow as tf
```

```
# Helper libraries
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
print(tf.__version__)
```

```
fashion_mnist = tf.keras.datasets.fashion_mnist
```

```
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
               'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

```

train_images.shape
plt.figure()
plt.imshow(train_images[0])
plt.colorbar()
plt.grid(False)
plt.show()

train_images = train_images / 255.0
test_images = test_images / 255.0
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap=plt.cm.binary)
    plt.xlabel(class_names[train_labels[i]])
plt.show()

```

## Xử lý ngôn ngữ tự nhiên Chollet, Chapter 6: Deep learning for text and sequences

5. Trình bày quá trình học với keras. Tham khảo <https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/>
6. Chạy ví dụ sau và giải thích

```

=====
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt
import numpy as np
# load pima indians dataset
dataset = np.loadtxt("pima-indians-diabetes.csv", delimiter=",")
# split into input (X) and output (Y) variables
X = dataset[:,0:8]
Y = dataset[:,8]
# create model
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# Fit the model
history = model.fit(X, Y, validation_split=0.33, epochs=150, batch_size=10, verbose=0)
# list all data in history
print(history.history.keys())
# summarize history for accuracy

```

```

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
=====

```

7. Xử lý ngôn ngữ tự nhiên: Trình bày hiểu biết về xử lý ngôn ngữ tự nhiên. Chollet, Chapter 6:  
Deep learning for text and sequences. Tham khảo  
<https://machinelearningmastery.com/what-are-word-embeddings/>  
<https://www.geeksforgeeks.org/word-embeddings-in-nlp/>  
<https://www.turing.com/kb/guide-on-word-embeddings-in-nlp>

8. Chạy ví dụ sau và giải thích

```

=====
from keras.datasets import imdb
from keras import preprocessing
from keras.layers import Embedding
#from tensorflow.keras.models import Sequential
#from tensorflow.keras.layers import Dense
from keras.models import Sequential
from keras.layers import Flatten, Dense

import matplotlib.pyplot as plt
import numpy as np

max_features = 10000
maxlen = 20
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
x_train = preprocessing.sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = preprocessing.sequence.pad_sequences(x_test, maxlen=maxlen)

model = Sequential()
model.add(Embedding(10000, 8, input_length=maxlen))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
model.summary()
history = model.fit(x_train, y_train,

```

```

epochs=10,
batch_size=32,validation_split=0.2)

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

```

9. Word Embedding là gì? Trình bày hiểu biết của mình và các ứng dụng của word embedding (3 trang). Áp dụng cho phân loại text: Trình bày kiến thức (3 trang) và code, thêm phần show như Bài tập 5, chạy code với 3 kiến trúc khác nhau (thêm layer, neuron ) và nêu nhận xét outputs trên Biểu đồ <https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/>
10. Áp dụng cho phân loại review restaurant: Trình bày kiến thức (3 trang) và code, thêm phần show như Bài tập 5, chạy code với 3 kiến trúc khác nhau (thêm layer, neuron ) và nêu nhận xét outputs trên Biểu đồ  
<https://medium.com/analytics-vidhya/understanding-embedding-layer-in-keras-bbe3ff1327ce>  
<https://medium.com/analytics-vidhya/understanding-embedding-layer-in-keras-bbe3ff1327ce>  
<https://saturncloud.io/blog/understanding-embedding-layers-in-keras-a-comprehensive-guide/>  
<https://www.kaggle.com/code/rajmehra03/a-detailed-explanation-of-keras-embedding-layer>
11. Chạy các ví dụ trong Chapter 5 (Chollet) và giải thích
12. Chạy các ví dụ trong Chapter 6 (Chollet) và giải thích