

BÀI TẬP DeepLearning 4

Yêu cầu

- Sinh viên copy code từng phần code ra file doc và giải thích
- Bài tập này xem như chuẩn bị cho Phần 2: Deep Learning
- **Đánh giá tuần 23/10 ----30/10**

1. Chạy chương trình sau và giải thích

#0. Giải thích các thư viện

```
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
```

#1. Chuẩn bị data: sinh viên vào google gõ tên file và load về máy mình

```
dataset = np.loadtxt("pima-indians-diabetes.csv", delimiter=",")
```

```
# split into input (X) and output (Y) variables
```

#2. Giải thích X,Y các chỉ số trong[], thử thay đổi 1 chỉ số

```
#khác xem sao. Cho ví dụ
```

```
X = dataset[:,0:8]
```

```
Y = dataset[:,8]
```

```
# create model
```

#3. Xây dựng model. Giải thích các dòng lệnh sau

```
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
```

#4. compile và giải thích 3 tham số loss, optimizer, metrics

```
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

```
# Fit the model
```

#5. fit model: giải thích các tham số...epoch, batch...

```
history = model.fit(X, Y, validation_split=0.33, epochs=150,
batch_size=10, verbose=0)
```

```
# list all data in history
```

#6. Giải thích history

```
print(history.history.keys())
```

```
# summarize history for accuracy
```

#7. Giải thích Biểu đồ

```
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
#8. Giải thích biểu đồ
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()
=====

```

2. Sử dụng code đã học ở Câu 1 để show các biểu đồ trong code sau và giải thích

```

===

#Import required packages
from keras.models import Sequential
from keras.layers import Dense
import numpy as np

# Getting the data ready
# Generate train dummy data for 1000 Students and dummy
testfor 500
#Columns :Age, Hours of Study &Avg Previous test scores
np.random.seed(2018) #Setting seed for reproducibility
train_data, test_data = np.random.random((1000, 3)),
np.random.random((500, 3))
#Generate dummy results for 1000 students : Whether Passed
(1) or Failed (0)
labels = np.random.randint(2, size=(1000, 1))
#Defining the model structure with the required layers,
# ofneurons, activation function and optimizers
model = Sequential()
model.add(Dense(5, input_dim=3, activation='relu'))
model.add(Dense(4, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
#Train the model and make predictions

```

```

model.fit(train_data, labels, epochs=10, batch_size=32)
#Make predictions from the trained model
predictions = model.predict(test_data)

```

=====

3. Thêm phần show và chạy code sau

===

```

import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation

# Generate dummy training dataset
np.random.seed(2018)
x_train = np.random.random((6000,10))
y_train = np.random.randint(2, size=(6000, 1))
# Generate dummy validation dataset
x_val = np.random.random((2000,10))
y_val = np.random.randint(2, size=(2000, 1))
# Generate dummy test dataset
x_test = np.random.random((2000,10))
y_test = np.random.randint(2, size=(2000, 1))
#Define the model architecture
model = Sequential()
model.add(Dense(64, input_dim=10,activation = "relu"))
#Layer 1
model.add(Dense(32,activation = "relu")) #Layer 2
model.add(Dense(16,activation = "relu")) #Layer 3
model.add(Dense(8,activation = "relu")) #Layer 4
model.add(Dense(4,activation = "relu")) #Layer 5
model.add(Dense(1,activation = "sigmoid")) #OutputLayer
#Configure the model
model.compile(optimizer='Adam',loss='binary_crossentropy',metrics=['accuracy'])
#Train the model
model.fit(x_train, y_train, batch_size=64, epochs=3,
validation_data=(x_val,y_val))

#evaluate(x=None, y=None, batch_size=None,
verbose=1,sample_weight=None, steps=None)

```

```
print(model.evaluate(x_test,y_test))
print(model.metrics_names)
#print 10 predictions
pred = model.predict(x_test)
pred[:10]
```

=====

4. Chạy ví dụ trong Chap 3 ([0.2]) và bổ sung phần show() các biểu đồ + Giải thích như Cây 1,2,3
5. Chạy ví dụ trong Chap 4 ([0.2]) và bổ sung phần show() các biểu đồ + Giải thích như Cây 1,2,3, 4