

BÀI TẬP 4: DEEP LEARNING

Due date: 11:30PM, 18/09/2023

4.1. 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 \(stanford.edu\)](#)

4.2. Giải thích các dòng đánh dấu # trong Bài tập 3.5 (> 5 trang)

4.3. 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/>

4.4. Chạy 3 ví dụ dưới đây và giải thích các dòng lệnh theo hiểu biết của mình (>5 trang)

=====

```
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()
```