

PHÁT TRIỂN HỆ THỐNG MINH

Bài tập intsyst_cuoicung_lớp_nhóm_hotensinhvien

Câu 1 (2đ)

- Chạy code được cho trong **CODE_1**, copy Hình sang file DOC và giải thích các , đánh giá
 - Epoch, batch
 - Train, validation set
 - Các đồ thị Loss và accuracy có ý nghĩa thế nào
 - Thêm tập test vào và đánh giá sử dụng MAE, MSE, RMSE để giải thích
- Tạo mô hình **7_layer** bằng cách thêm 7 layer vào code, thêm 100 neuron và dropout và giải thích như Câu 1
- Giải thích CNN, RNN, LSTM
- Thay bởi mô hình CNN, RNN, LSTM với ≥ 5 layer và 100 neuron. So sánh, đánh giá 3 mô hình với **7_layer** với các độ đo MAE, MSE, RMSE. Giải thích dựa trên các plot sinh ra được

Câu 2 (2đ) Dự đoán điểm và phân loại sinh viên A+, A, B+....

- Tự sinh tập DATA với 1000 sinh viên trong đó **500 Môn 1** (10%, 10%, 20%, 60%), **500 Môn 2**(10%, 20%, 20%, 50%)
- Sử dụng mô hình 7_layer, CNN, RNN, LSTM với ≥ 5 Layer và 100 neuron, cơ chế dropout...để dự đoán điểm và phân loại
- Giải thích các mô hình
- Đánh giá và xây dựng ứng dụng mobile, web

Câu 3 (3đ)

Xây dựng Hệ dự đoán giá nhà Hà đông với DATA cho trên trang face

- Tiền xử lý data, bổ sung thuộc tính khu vực, đường
- Tự xây dựng kiến trúc mạng với 7 Layer
- Sử dụng CNN, RNN, LSTM với 7 Layers. Giải thích các khái niệm liên quan
- Thay bởi mô hình CNN, RNN, LSTM với 7 layer và 100 neuron. So sánh, đánh giá 3 mô hình với **7_layer** với các độ đo MAE, MSE, RMSE. Giải thích dựa trên các plot sinh ra được
- Triển khai mô hình được chọn và phát triển một ứng dụng trên Mobile hay web để người dùng nhập: hướng nhà, khu vực, diện tích...Heej thống hiển thị và hướng dẫn đến vị trí qua GMap

Câu 4 (2đ)

Nhận dạng ảnh để điểm danh trong lớp (data trên face).

Câu 5 (2đ)

Phân tích comment của khách hàng trên Tiki, Shpee để xác định sở thích, xu hướng và gợi ý sản phẩm (sinh viên tùy chọn sản phẩm)

- Xây dựng 2 tập DATA để thử nghiệm đánh giá
- Tiền xử lý DATA: các bước và kết quả
- Sử dụng kỹ thuật BERT để phân tích các comment

4. Xây dựng 3 mô hình deep learning cho xử lý ngôn ngữ tự nhiên
5. Đánh giá các mô hình và chọn ra mô hình để triển khai
6. Xây dựng tư vấn dựa trên đánh giá sản phẩm của người dùng

=====

CODE_1 (Cho Câu 1)

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_classification
import numpy as np

# 1. Tạo dataset giả lập (phân loại nhị phân)
X, y = make_classification(n_samples=1000, n_features=20, n_classes=2, random_state=42)

# 2. Chia tập dữ liệu thành train (80%) và validation (20%)
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# 3. Xây dựng mô hình MLP đơn giản
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(X_train.shape[1],))) # Lớp đầu vào
model.add(Dense(32, activation='relu')) # Lớp ẩn
model.add(Dense(1, activation='sigmoid')) # Lớp đầu ra cho phân loại nhị phân

# 4. Compile mô hình
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# 5. Huấn luyện mô hình với dữ liệu train và validate đồng thời
history = model.fit(
    X_train, y_train,
    epochs=20,           # Số lượng epoch
    batch_size=32,       # Kích thước batch
    validation_data=(X_val, y_val), # Sử dụng tập validation
    verbose=1           # Hiển thị chi tiết quá trình huấn luyện
)

# 6. Đánh giá mô hình trên tập validation sau huấn luyện
val_loss, val_accuracy = model.evaluate(X_val, y_val)
print(f"Validation Loss: {val_loss:.4f}")
print(f"Validation Accuracy: {val_accuracy:.4f}")

import matplotlib.pyplot as plt

# Vẽ đồ thị Loss (mất mát) giữa train và validation
```

```
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
# Vẽ đồ thị Accuracy (độ chính xác) giữa train và validation
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```