Uygulama 10. LSTM Derin Öğrenme Modelinin Uygulanması

Bu uygulama ile yaprak verisetinin LSTM ile analizi gerçekleştirilecektir. Yaprak verisetine ait bilgiler kullanılarak doksan dokuz sınıflı bir sınıflandırma işlemi gerçekleştirilecektir.

Öncelikle gerekli kütüphaneler yüklenir.

```
#kütüphanelerin yüklenmesi
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
```

Ardından veriseti yüklenir ve verideki sınıflar etiketlenir.

```
#Verilerin yüklenmesi
eğitimVeri = pd.read_csv("..../train.csv")
testVeri = pd.read_csv("..../test.csv")

#Sınıf sayısı ve etiketlerinin belirlenmesi
label_encoder =LabelEncoder().fit(eğitimVeri.species)
labels = label_encoder.transform(eğitimVeri.species)
classes = list(label_encoder.classes_)
```

Girdi ve çıktı verileri hazırlanır, standartlaştırma işlemi yapılır.

```
#Eğitim ve test verileri gereksiz bilgilerden arındırılır
eğitimVeri = eğitimVeri.drop(["id","species"],axis=1)
testVeri = testVeri.drop(["id"],axis=1)
nb_features = 192
nb_classes = len(classes)

#Verilerin standartlaştırılması
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(eğitimVeri.values)
eğitimVeri = scaler.transform(eğitimVeri.values)
```

Eğitim ve doğrulama verileri hazırlanarak kategorileştirme işlemi gerçekleştirilir.

```
#Eğitim ve doğrulama verilerinin belirlenmesi
from sklearn.model_selection import train_test_split
X_train, X_valid, y_train, y_valid = train_test_split(eğitimVeri, labels,test_size = 0.2)

#Çıktı değerlerinin kategorileştirilmesi
from tensorflow.keras.utils import to_categorical
y_train = to_categorical(y_train)
y_valid = to_categorical(y_valid)
```

Ardından girdiler lstm yapısına uygun olacak şekilde yeniden boyutlandırılır.

```
#Girdi verilerinin yeniden boyutlandırılması
X_train = np.array(X_train).reshape(792,192,1)
X_valid = np.array(X_valid).reshape(198,192,1)
```

Ardından ağın modeli oluşturulur.

```
#Modelin oluşturulması
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation,Dropout
from tensorflow.keras.layers import Flatten,LSTM,BatchNormalization

model = Sequential()
model.add(LSTM(512,input_shape=(nb_features,1)))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add((Flatten()))
model.add(Dropout(0.15))
model.add(Dense(2048, activation="relu"))
model.add(Dense(1024, activation="relu"))
model.add(Dense(nb_classes, activation="softmax"))
model.summary()
```

Daha sonra model derlenir ve eğitilir.

```
#Modelin kesinlik, f1-skor gibi değerlerin oluşturulması
from keras import backend as K

def recall_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())
    return recall

def precision_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    return precision

def f1_m(y_true, y_pred):
    precision = precision_m(y_true, y_pred)
    recall = recall_m(y_true, y_pred)
    return 2*((precision*recall)/(precision+recall+K.epsilon()))

#Modelin derlenmesi
model.compile(loss="categorical_crossentropy", optimizer = "adam", metrics=["accuracy",f1_m,precision_m,recall_m])

#Modelin eğitilmesi
score = model.fit(X_train, y_train, epochs = 100, validation_data=(X_valid,y_valid))
```

Ardından değerlendirme kriterlerinin ortalama sonuçları gösterilir.

```
#Ortalama değerlendirme kriterleri
print(("Ortalama Eğitim Kaybı: ", np.mean(model.history.history["loss"])))
print(("Ortalama Eğitim Başarımı: ", np.mean(model.history.history["accuracy"])))
print(("Ortalama Doğrulama Kaybı: ", np.mean(model.history.history["val_loss"])))
print(("Ortalama Doğrulama Başarımı: ", np.mean(model.history.history["val_accuracy"])))
print(("Ortalama F1-Skor Değeri: ", np.mean(model.history.history["val_f1_m"])))
print(("Ortalama Kesinlik Değeri: ", np.mean(model.history.history["val_precision_m"])))
print(("Ortalama Duyarlılık Değeri: ", np.mean(model.history.history["val_recall_m"])))
```

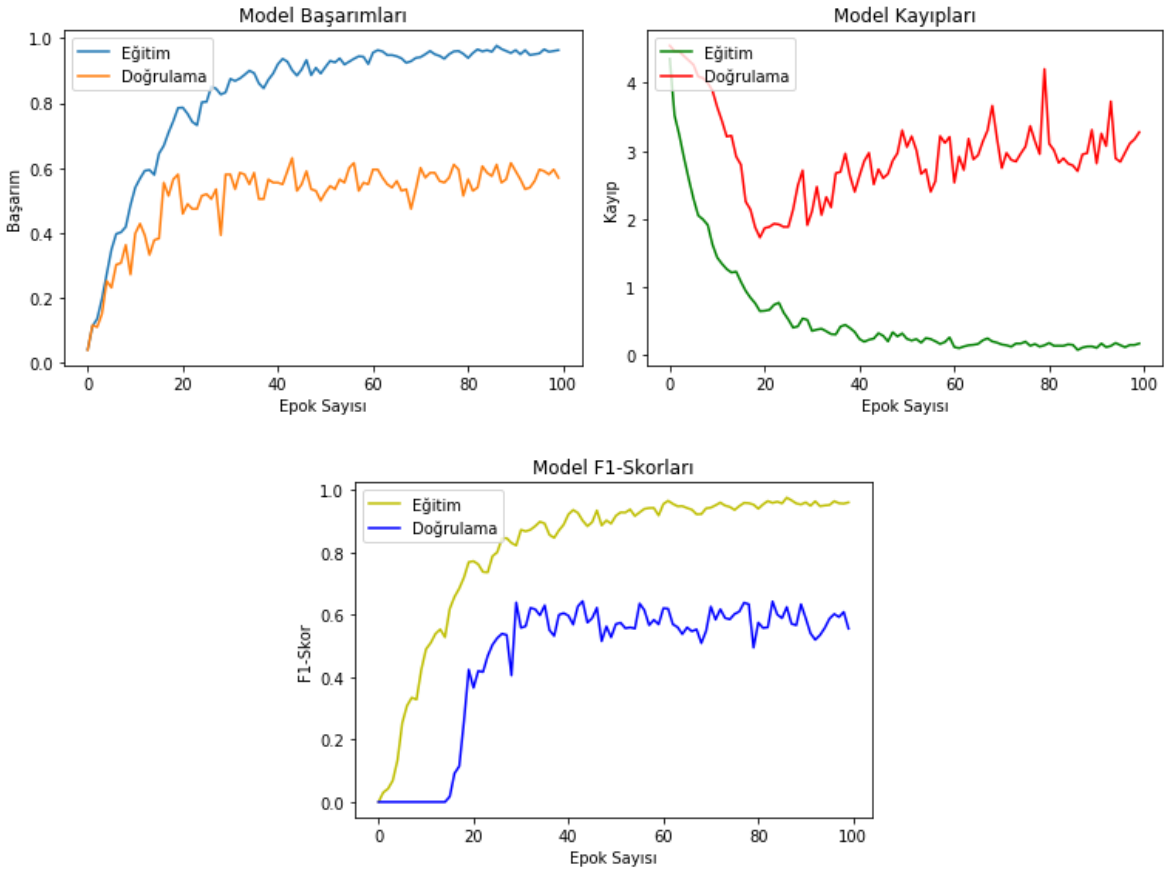
Daha sonra sonuçlar grafik üzerinde gösterilir.

```
#Eğitim ve doğrulama değerlerinin grafik üzerinde gösterilmesi
import matplotlib.pyplot as plt
plt.plot(model.history.history["accuracy"])
plt.plot(model.history.history["val_accuracy"])
plt.title("Model Başarımları")
plt.ylabel("Başarım")
plt.xlabel("Epok Sayısı")
plt.legend(["Eğitim","Doğrulama"], loc="upper left")
plt.show()

plt.plot(model.history.history["loss"],color="g")
plt.plot(model.history.history["val_loss"],color="r")
plt.title("Model Kayıpları")
plt.ylabel("Kayıp")
plt.xlabel("Epok Sayısı")
plt.legend(["Eğitim","Doğrulama"], loc="upper left")
plt.show()

plt.plot(model.history.history["f1_m"],color="y")
plt.plot(model.history.history["val_f1_m"],color="b")
plt.title("Model F1-Skorları")
plt.ylabel("F1-Skor")
plt.xlabel("Epok Sayısı")
plt.legend(["Eğitim","Doğrulama"], loc="upper left")
plt.show()
```

Örnek çıktı



Uygulama 10. Değerlendirme Soruları

1) 4. Haftada verilen "telefon_fiyat_değişimi.csv" verisini LSTM modelini oluşturarak analiz ediniz. Analiz işleminiz sonunda kayıp ve başarım değerlerini çıktı olarak veriniz. Bunlara ek olarak sonuçları grafik üzerinde gösteriniz **(25p).**

2) Kaktüs bitkisine ait verileri kullanarak LSTM derin öğrenme modeli ile sınıflandırma gerçekleştiriniz. Sınıflandırma sonuçlarını başarım, f1-skor, kesinlik, duyarlılık ve AUC skorlarını kullanarak değerlendiriniz. Bu değerleri grafik üzerinde gösteriniz **(50p)**.Verisetine https://www.kaggle.com/c/aerial-cactus-identification/data adresinden ulaşılabilir.

3) Sayılara ait verileri kullanarak LSTM derin öğrenme modeli ile sınıflandırma gerçekleştiriniz. Sınıflandırma sonuçlarını başarım, f1-skor, kesinlik, duyarlılık ve AUC skorlarını kullanarak değerlendiriniz. Bu değerleri grafik üzerinde gösteriniz. Bunlara ek olarak tSNE analizini gerçekleştirip, grafik üzerinde gösteriniz **(50p).** Verisetine https://www.kaggle.com/c/digit-recognizer adresinden ulaşılabilir.