

## Uygulama 11. CNN ve LSTM Derin Öğrenme Modellerinin Birleştirilmesi

Bu uygulama ile rakam verisetinin CNN ve LSTM ile analizi gerçekleştirilecektir. Rakam verisetine ait bilgiler kullanılarak on 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
rakamVeri = pd.read_csv("../train.csv")

#determine the number of classes(labels)
label_encoder = LabelEncoder().fit(rakamVeri.label)
labels = label_encoder.transform(rakamVeri.label)
classes = list(label_encoder.classes_)
```

Girdi ve çıktı verileri hazırlanır, standartlaştırma işlemi yapılır.

```
#Verilerin hazırlanması
rakamVeri = rakamVeri.drop(["label"],axis=1)
nb_features = 784
nb_classes = len(classes)

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

Eğitim ve doğrulama verileri hazırlanarak kategorileştirme işlemi gerçekleştirilir.

```
#Eğitim ve doğrulama verilerinin ayrılması
from sklearn.model_selection import train_test_split
X_train, X_valid, y_train, y_valid = train_test_split(rakamVeri,labels,test_size = 0.3)

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

Ardından girdiler CNNLSTM yapısına uygun olacak şekilde yeniden boyutlandırılır.

```
#Girdi verilerinin yeniden boyutlandırılması
X_train = np.array(X_train).reshape(29400,784,1)
X_valid = np.array(X_valid).reshape(12600,784,1)
```

Ardından ağıın modeli oluşturulur.

```
#Modelin oluşturulması
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout, Conv1D,
from tensorflow.keras.layers import MaxPooling1D, Flatten, LSTM, BatchNormalization

model = Sequential()
model.add(Conv1D(512,1,input_shape=(nb_features,1)))
model.add(Activation("relu"))
model.add(MaxPooling1D(2))
model.add(Conv1D(256,1))
model.add(Activation("relu"))
model.add(MaxPooling1D(2))
model.add(LSTM(512))
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 derlenmesi
model.compile(loss="categorical_crossentropy", optimizer = "rmsprop", metrics=["accuracy"])

#Modelin eğitilmesi
score = model.fit(X_train, y_train, epochs = 250, validation_data=(X_valid,y_valid))
```

Ardından değerlendirme kriterlerinin ortalama sonuçları gösterilir.

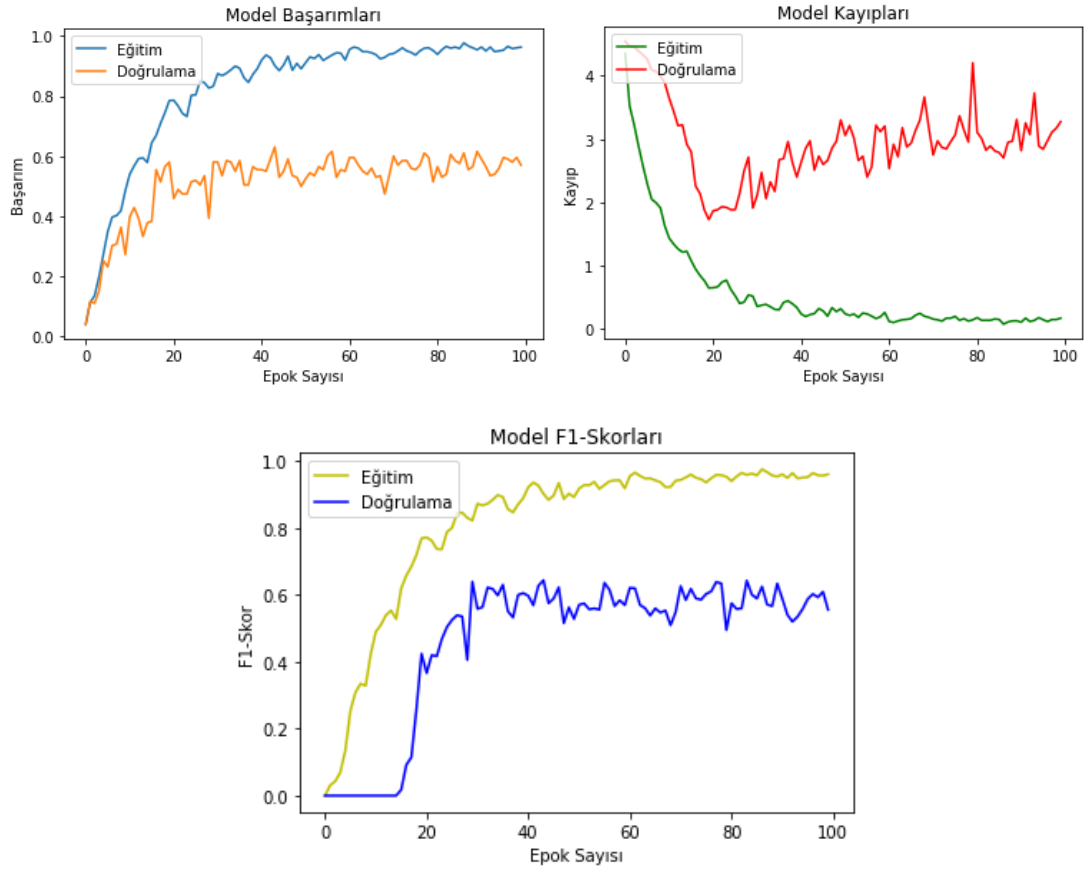
```
#Ortlama değerlerin verilmesi
print(("Ortalama Doğrulama Kaybı: ", np.mean(model.history.history["val_loss"])))
print(("Ortalama Doğrulama Başarımı: ", np.mean(model.history.history["val_accuracy"])))
```

Daha sonra sonuçlar grafik üzerinde gösterilir.

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

Örnek çıktı



## Uygulama 11. Değerlendirme Soruları

- 1) 4. Haftada verilen "telefon\_fiyat\_değişimi.csv" verisini CNNRNN modelini oluşturarak analiz ediniz. Analiz işleminiz sonunda kayıp ve başarımlarını çıktı olarak veriniz. Bunlara ek olarak sonuçları grafik üzerinde gösteriniz **(20p)**.
- 2) Kredi kartına ait verileri kullanarak CNNLSTM derin öğrenme modeli ile sınıflandırma gerçekleştiriniz. Sınıflandırma sonuçlarını başarımlarını, f1-skor, kesinlik, duyarlılık ve AUC skorlarını kullanarak değerlendiriniz. Bu değerleri grafik üzerinde gösteriniz **(40p)**. Verisetine <https://www.kaggle.com/mlg-ulb/creditcardfraud> adresinden ulaşılabilir.
- 3) Kalp rahatsızlığına ait verileri kullanarak CNNLSTM derin öğrenme modeli ile sınıflandırma gerçekleştiriniz. Sınıflandırma sonuçlarını başarımlarını, f1-skor, kesinlik, duyarlılık ve AUC skorlarını kullanarak değerlendiriniz. Bu değerleri grafik üzerinde gösteriniz. Bunlara ek olarak tSNE analizini gerçekleştirip, grafik üzerinde gösteriniz **(40p)**. Verisetine <https://www.kaggle.com/ronitf/heart-disease-uci> adresinden ulaşılabilir.