PROJE RAPORU

Emre KOCABEY 202113171815

Fatma ESER 202213171402

Mustafa CEYHAN 202113171812

HABER METNİ TEMELLİ ÇOK SINIFLI SINIFLANDIRMA

1. Giriş

Bu projede, haber metinleri ve açıklamalarına dayanarak çok sınıflı (multiclass) bir metin sınıflandırma modeli geliştirilmiştir. Amaç, haber kaynaklarından toplanan "World (Dünya)", "Sports (Spor)" ve "Business (İş Dünyası)" kategorilerinden üç tanesini kullanarak, girilen metnin hangi kategoriye ait olduğunu doğru bir şekilde tahmin edebilen bir sinir ağı (RNN tabanlı LSTM) modelini oluşturmaktır.

2. Veri Seti

2.1. Veri Kaynağı ve Özellikleri

- Veri Kümesi: AG News Classification Dataset
- Kaynak: Kaggle (https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset)
- Toplam Örnek Sayısı: 120.000 (100.000 eğitim, 20.000 test)
- Kaggle Usability Score: 7.6 (yüksek kalite)
- Sınıf Sayısı: 4 (World, Sports, Business, Sci/Tech)

Bu proje için seçilen sınıflar:

1. World (Etiket 0)
2. Sports (Etiket 1)
3. Business (Etiket 2)

NOT: Sci/Tech sınıfı (Etiket 3) bu projede kullanılmamıştır; sadece üç kategori üzerinden çok sınıflı sınıflandırma yapılmıştır.

2.2. Veri Ön İşleme Adımları

1. CSV Dosyasının Hazırlanması:
   Kaggle' dan indirilen dataset çok büyük olduğu için sadece indirilen data setin train.csv kısmı kullanıldı.
2. Sınıf Filtreleme:
   Class Index sütunu 0, 1, 2, 3 değerleri alır. Biz sadece 0, 1 ve 2 etiketlerini kullanmak üzere filtre uyguladık. Yeni DataFrame, yalnızca "World", "Sports" ve "Business" örneklerini içerir.

3. Metin Oluşturma:
Title (Başlık) ve Description (Açıklama) sütunları birleştirilerek tek bir text sütunu oluşturuldu. Null değerli satırlar '' olarak dolduruldu ve birleştirme işlemi yapıldı.
4. Küçük Harfe Dönüştürme ve Temizleme:
Tüm metinler .lower() metodu ile küçültüldü. Noktalama işaretleri, özel karakterler ve sayılar ihtiyaca göre temizlenebilir.
5. Sınıf Etiketleme (One-Hot Encoding):
Class Index değerleri 0 → 0, 1 → 1, 2 → 2 olarak korundu. tf.keras.utils.to_categorical() fonksiyonu ile one-hot formata dönüştürüldü.
6. Tokenizasyon ve Pad/Truncate İşlemi:
Kelime Sözlüğü Boyutu: 10.000, Maksimum Dizi Uzunluğu: 200. Metinler texts_to_sequences() ile sayısal dizilere dönüştürüldü ve pad_sequences(..., maxlen=200) ile sabit uzunluğa getirildi.

2.3. Sınıf Dağılımı

Her bir sınıf örnek sayısının yaklaşık eşit olması modelin dengeli eğitim almasını sağlar. Aşağıda sınıfların dağılımına ilişkin grafik yerleştirilecektir:
[Sınıf Dağılımı Grafiği Buraya Eklenecek]

3. Model Mimarisi

3.1. Kullanılan Model Türü: BiLSTM (Çift Yönlü LSTM)

Neden RNN/LSTM?
- Haber metinleri sıralı bir yapıya sahiptir; LSTM hücreleri, metindeki uzun dönemli bağımlılıkları yakalamada etkilidir.
- BiLSTM (Bidirectional LSTM), metnin hem ileri hem geri bağlamını öğrenerek performansı artırır.

Modelin Temel Akışı:
1. Embedding Katmanı:
Embeddingkatmanı, her kelimeyi 128 boyutlu bir vektöre dönüştürür.
2. Çift Yönlü (Bidirectional) LSTM Katmanları:
- Birinci BiLSTM katmanı: Bidirectional(LSTM(64, return_sequences=True))
return_sequences=True ile her adımda çıktı veren LSTM, üst katmana tüm gizli durumları aktarıyor.
- Dropout (0.5)
- İkinci BiLSTM katmanı: Bidirectional(LSTM(32))
Bu katman çıktı olarak yalnızca son gizli durumu (hidden state) verir.
3. Tam Bağlı (Dense) Katmanlar:
- Dense(64, activation='relu') ara katmanı
- Dense(3, activation='softmax') → 3 sınıf çıktısı (World, Sports, Business)

Neden 2 Katman BiLSTM?
- İkinci LSTM katmanı, birinci katmandaki zaman serisi özelliklerini entegre ederek daha

yüksek seviyeli temsiller üretir.
- Dropout katmanları (her bir BiLSTM sonrası %50) ile overfitting riski azaltılır.

4. Eğitim Süreci

4.1. Eğitim / Doğrulama Ayrımı

Eğitim Verisi: Birleştirilmiş veri kümesinin %80'i
Doğrulama Verisi: Eğitim verisinin %10'u (Early Stopping ve hiperparametre ayarı için)
Test Verisi: Birleştirilmiş veri kümesinin %20'si

Stratifikasyon: Sınıf dağılımının her bölmede dengeli kalmasını sağlar.

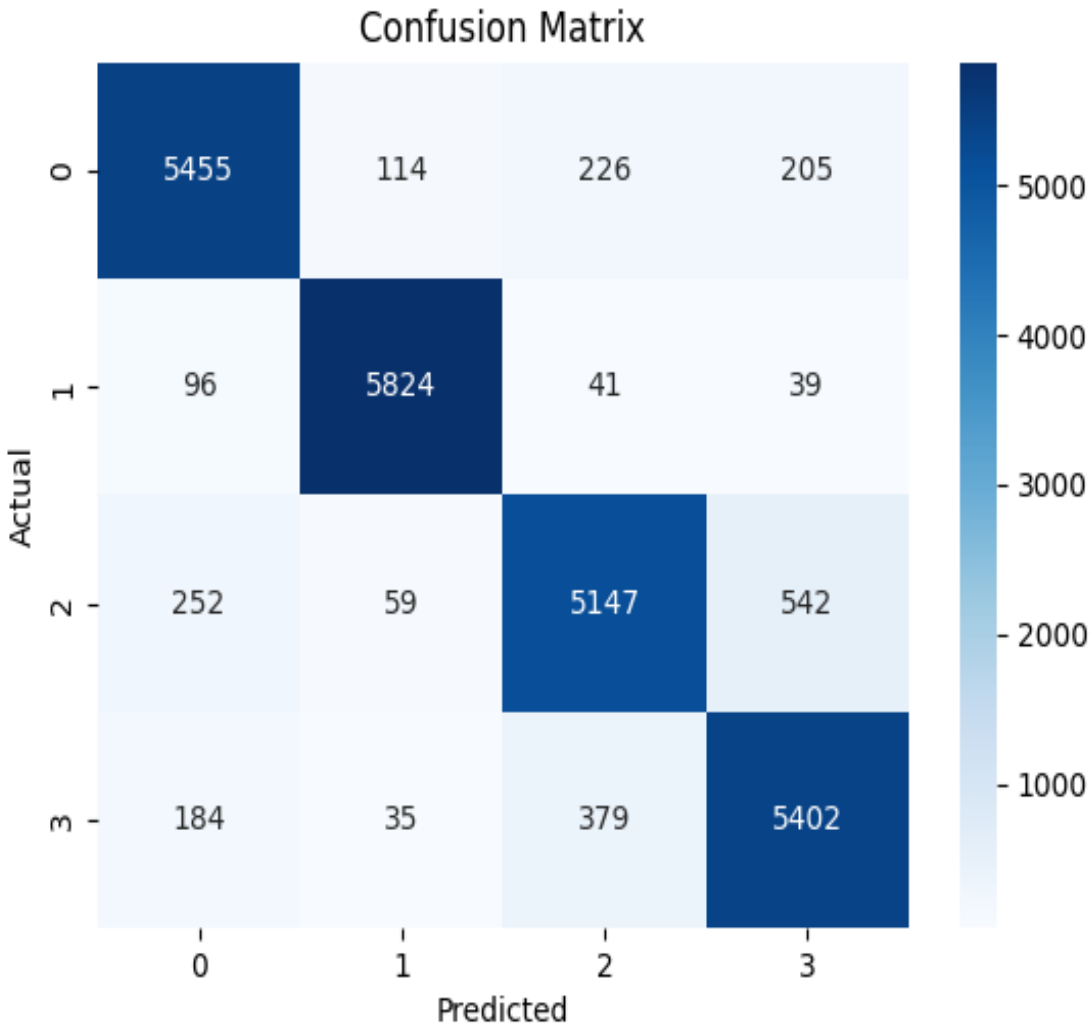
4.2. Eğitim Ayarları ve Hiperparametreler

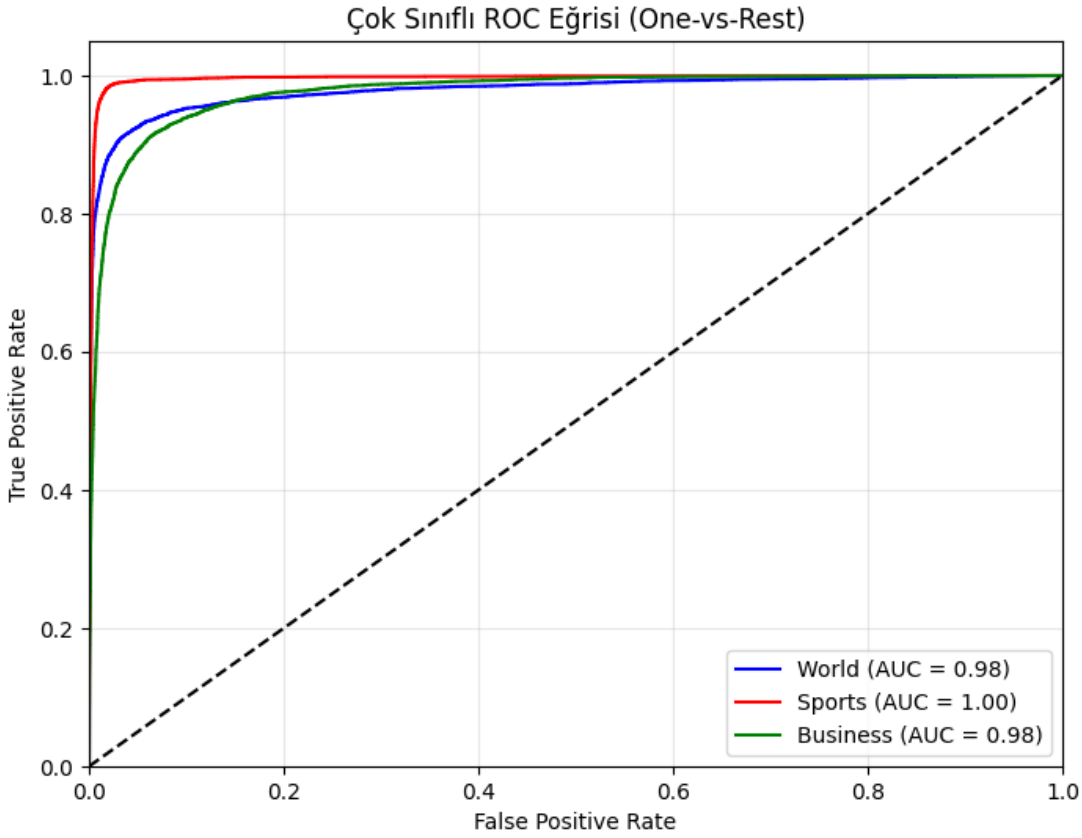
- Epoch: 5 (gerektikçe artırılabilir / EarlyStopping ile ayarlanabilir)
- Batch Size: 64 (CPU'da verimlilik için dengeli)
- Optimizer: Adam (learning_rate default = 0.001)
- Loss: categorical_crossentropy (çok sınıflı softmax çıktısı)
- Metrics: ['accuracy']

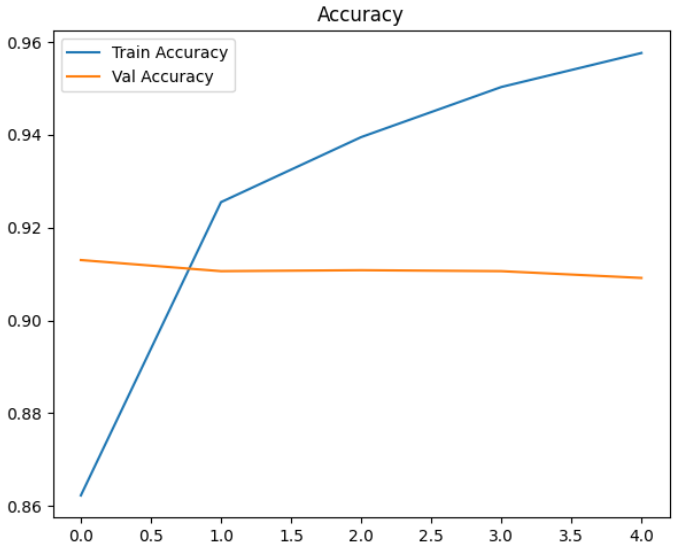
4.3. Erken Durdurma ve Model Kaydetme

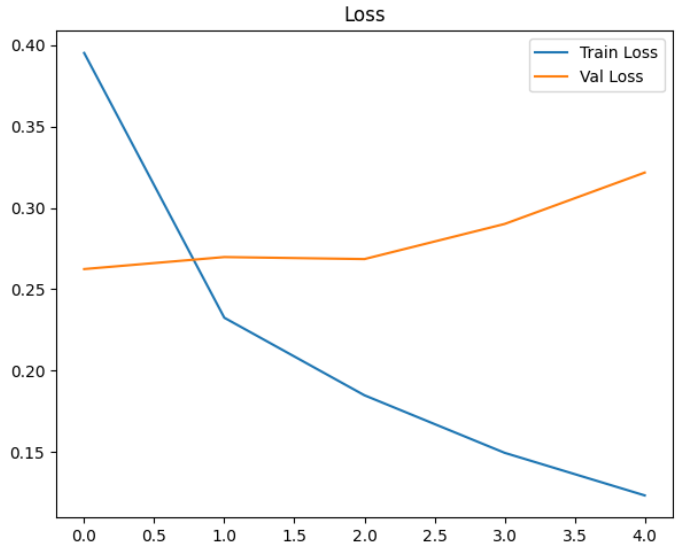
EarlyStopping ve ModelCheckpoint, eğitim süresini kısaltarak en iyi ağırlıkları saklamayı

```
Test Loss: 0.3195, Test Accuracy: 0.9095
750/750 ━━━━━━━━━━━━━━━━━━━━ 11s 15ms/step

Classification Report:

              precision    recall  f1-score   support

           0       0.91      0.91      0.91      6000
           1       0.97      0.97      0.97      6000
           2       0.89      0.86      0.87      6000
           3       0.87      0.90      0.89      6000

    accuracy                           0.91     24000
   macro avg       0.91      0.91      0.91     24000
weighted avg       0.91      0.91      0.91     24000
```


Confusion Matrix
Actual
Predicted
0
1
2
3
5455
114
226
205
96
5824
41
39
252
59
5147
542
184
35
379
5402
5000
4000
3000
2000
1000


Çok Sınıflı ROC Eğrisi (One-vs-Rest)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
World (AUC = 0.98)
Sports (AUC = 1.00)
Business (AUC = 0.98)


Accuracy
Train Accuracy
Val Accuracy
0.86
0.88
0.90
0.92
0.94
0.96
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Loss
Train Loss
Val Loss
0.15
0.20
0.25
0.30
0.35
0.40
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0