

Metinlerin 6 Duyguya Göre Sınıflandırılması İçin LSTM Tabanlı Bir Öğrenme Modeli

Burak Can Kuş
Giray Budan

Danışman: Dr. Öğr. Üyesi Mete Eminağaoğlu

Wed 22 May 2019

Sunum Planı

- Projenin Tanımı
- Amaç
- Kapsam
- Kullanılan Araçlar ve Yöntemler
- Uygulama
- Testler
- Sonuçlar
- Değerlendirme
- Demo
- Kaynakça

Projenin Tanımı

- İngilizce metinleri 6 farklı duygu kategorisine göre sınıflandıran LSTM temelli bir derin öğrenme modeli oluşturulması
- Saima Aman'ın blog yazılarından ürettiği veri seti kullanılmıştır.

Amaç

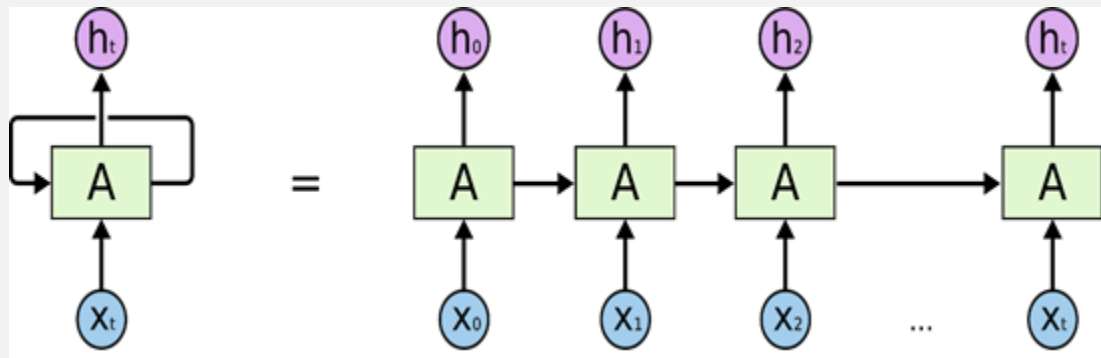
- Projenin amacı, yazı ile iletişimde cümle düzeyinde aktarılmak istenen duygunun tespit edilmesidir.
- Sınıflandırma için Paul Ekman'ın 1992 yılında kategorize ettiği 6 farklı duygu (happiness, sadness, anger, disgust, surprise, fear) temel alınır.

Proje Kapsamındaki Ana Konular

- Deep Learning
- LSTM (Long Short-Term Memory)
- Word2Vec
- Çoklu Sınıflandırma

Recurrent Neural Network (RNN)

- Verilerde sıralamanın önemli olduğu veri türleri ile çalışılırken kullanılır.
- Metinden duygu analizi veya Google Translate gibi yapay zeka destekli çeviri hizmeti verilen programların altyapılarında bulunur.
- Örnek RNN modeli:



Long Short-Term Memory (LSTM)

- RNN'lerin çok önceden olan olayları aklında tutabilmesi için farklı bir mimari yapıya ihtiyaçları vardır. Bu tür problemlerde, RNN'lerin bir çeşidi olan LSTM kullanılır.
- Aman veri setinden bir örnek;

“It was crazy, it was drunken, it was new and it was damn fun and I realized that there is less to learn from happy things but there are important messages you gain from good experiences.”

Kullanılan Araç ve Yöntemler

- Stanford Tokenizer
- GloVe: Global Vectors for Word Representation
- FastText
- PyTorch
- scikit-learn
- Batch Normalization
- Dropout
- Early Stopping

Stanford Tokenizer

- Bir tokenizer, kabaca metni kelimelere karşılık gelen token'lara böler. Yani cümlelerin kelimelerini ayırır.
- Projede, cümledeki her bir kelimenin vektörü kullanılacağı için Stanford Tokenizer ile bu ihtiyaç giderildi.

Tokenization öncesi:

I have to look at life in her perspective, and it would break anyones heart.

Tokenization sonrası:

['I', 'have', 'to', 'look', 'at', 'life', 'in', 'her', 'perspective', ',', 'and', 'it', 'would', 'break', 'anyones', 'heart', '.']

GloVe: Global Vectors for Word Representation

- GloVe, kelimeler için vektör gösterimleri elde etmek amacıyla geliştirilen unsupervised bir öğrenme algoritmasıdır.
- “you” ve “years” kelimeleri için kelime vektörü gösterimi;

```
you -0.0010919 0.33324 0.35743 -0.54041 0.82032 -0.49391 -0.32588 0.001
9972 -0.23829 0.35554 -0.60655 0.98932 -0.21786 0.11236 1.1494 0.73284
0.51182 0.29287 0.28388 -1.359 -0.37951 0.50943 0.7071 0.62941 1.0534 -
2.1756 -1.3204 0.40001 1.5741 -1.66 3.7721 0.86949 -0.80439 0.1839 -0.3
4332 0.010714 0.23969 0.066748 0.70117 -0.73702 0.20877 0.11564 -0.1519
0.85908 0.2262 0.16519 0.36309 -0.45697 -0.048969 1.1316
years 0.16962 0.4344 -0.042106 -0.63324 -0.1278 0.53668 -1.0662 -0.3262
9 -0.50079 0.10247 -0.021968 -0.35105 -0.64153 -0.42454 1.3836 -0.13543
-0.24754 0.22156 -0.65563 0.44424 0.17017 0.35816 0.56379 -0.48044 -0.
14765 -1.629 -0.31308 -0.47217 0.02659 0.47603 3.4619 0.12069 -0.045344
-0.47303 0.28569 -0.077584 -0.16447 0.7181 0.2617 -0.16841 -1.245 -0.0
76188 0.17493 0.24507 -0.63801 -0.21096 -0.49918 -0.50108 -0.7704 -0.32
234
```

FastText

- FastText de GloVe gibi kelimeler için vektör gösterimleri elde etmeye yarayan açık kaynaklı bir kütüphanedir.
- GloVe'dan farkı, sözlükte olmayan kelimeler için yakın bir vektör üretebilir.
- Projede FastText, GloVe ile karşılaştırmalı olarak kullanılmıştır ve kötü sonuç verdiği için GloVe tercih edilmiştir.

PyTorch

- PyTorch, Python için geliştirilmiş açık kaynaklı bir makine öğrenimi kütüphanesidir. Doğal dil işleme gibi alanlarda kullanımı yaygındır.
- PyTorch, iki adet high-level özellik sunar:
- Tensor işlemlerinin GPU üzerinde yapılabilmesi
- Derin sinir ağlarının otomatik olarak diferansiyel ağaçlarının oluşturulması ve backpropagation yapılması

scikit-learn

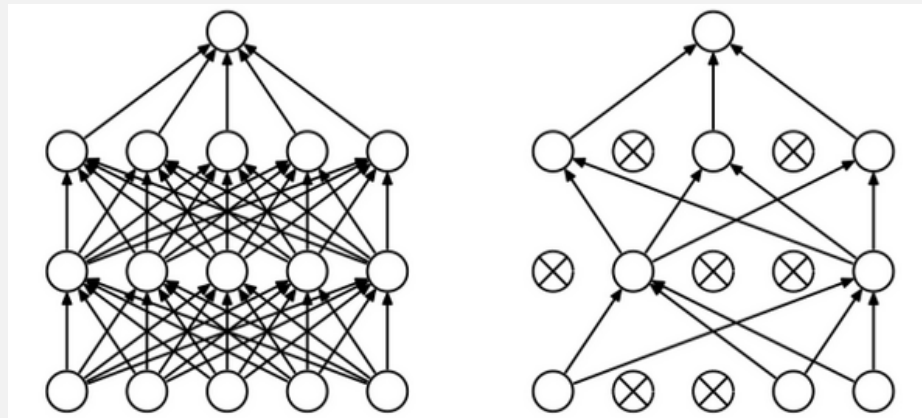
- Scikit-learn, Python programlama dili için ücretsiz makine öğrenimi kütüphanesidir.
- Projemizde cross validation ve sınıflandırma raporlarını hazırlamak için PIL (Pillow) kütüphanesi ile ortaklaşa kullanılmıştır.

Batch Normalization

- Batch normalization, yapay sinir ağlarının hızını, performansını, dengesini arttırmak ve aktivasyonları ayarlayarak ve ölçekleyerek giriş katmanını normalleştirmek için kullanılır.

Dropout

- Dropout, overfitting (aşırı öğrenme) olduğu durumlarda verileri dengeleyip bu sorunu çözmek amacıyla kullanılan bir tekniktir.
- Dropout, rastgele seçilen nöronların eğitim sırasında ihmal edildiği bir tekniktir. Bazı nöronlar rastgele bir şekilde çıkarılır.



Early Stopping

- Early stopping, makine öğreniminde bir sinir ağını eğitirken overfitting'i engellemek için kullanılan bir yöntemdir. Patience değeri alır.
- Ağın overfitting'e başlamadan önce kaç kez çalıştırılabileceğini belirlemeye yarar.



Yazılım Dili ve Geliştirme Ortamı

- Python projenin tamamında yazılım dili olarak kullanılmıştır.
- Proje, ortak çalışma yürütülebilmesi açısından Google Colaboratory üzerinde yazılmıştır.
- Anaconda platformu ile de bilgisayarlarımızda Python kullanımının daha pratik hale getirilmesi sağlanmıştır.

Eğitimde Kullanılan Veri Setleri

- Emotion Annotated Dataset: Saima Aman tarafından üretilen ve sadece akademik araştırmalar için kullanıma açık olan bir veri setidir. Blog yazılarından üretilmiştir.
- Twitter Dataset: J. Yang ve J.Leskovec tarafından Twitter kullanıcıları tarafından atılan 467 milyon tweet içeren veri setidir. Bu tweetler arasından elle 373 duygu cümlesi seçilip ek bir veri seti oluşturulmuştur.

UYGULAMA

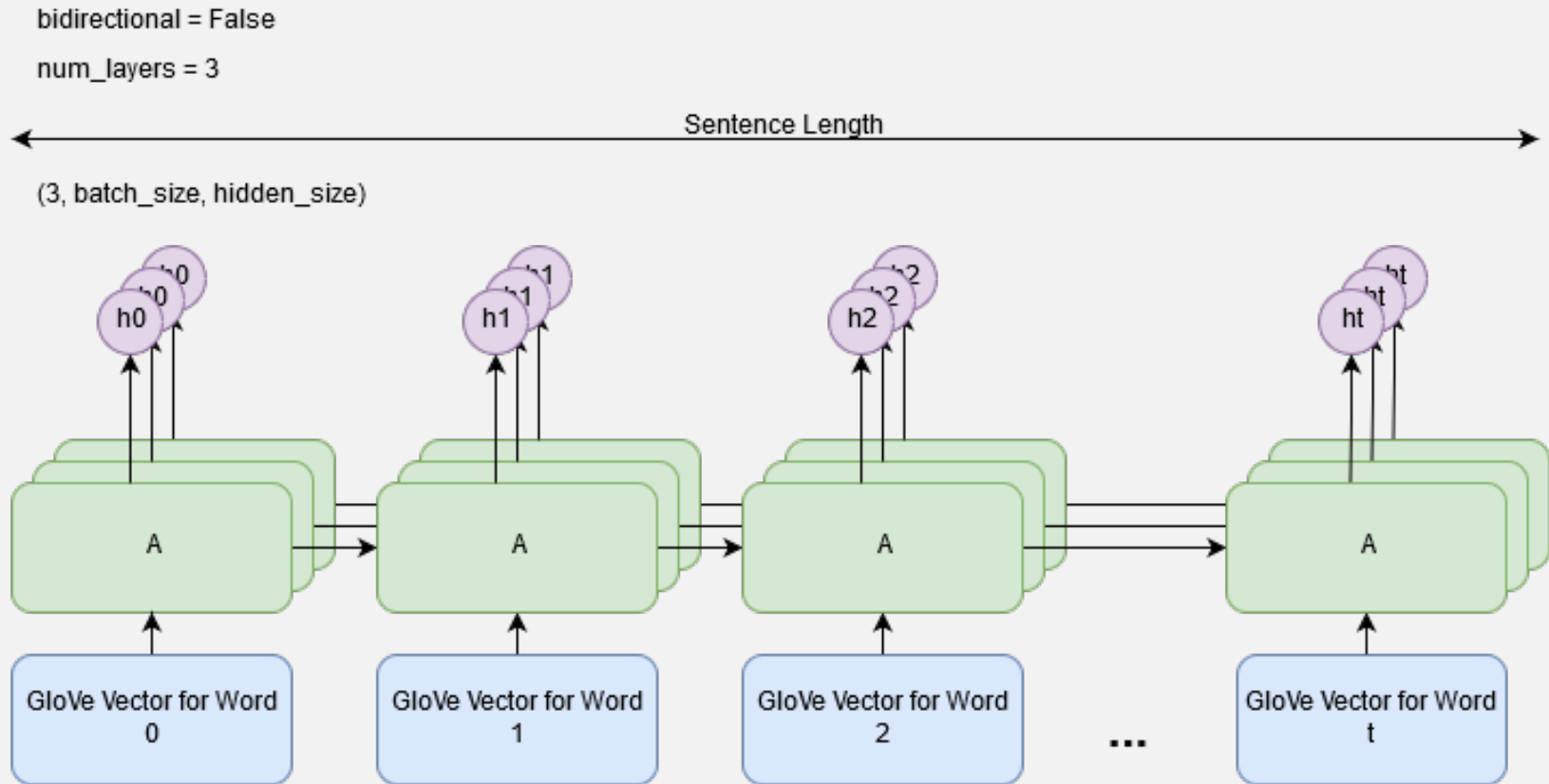
Veri Seti

- Kullanılan veri setinde, ilk etapta 536 **happiness**, 173 **sadness**, 179 **anger**, 172 **disgust**, 115 **surprise** ve 115 **fear** cümlesi olmak üzere toplam 1290 cümle vardır.
- Bu veri seti grup üyeleri tarafından filtrelenerek cümle sayısı 1011'e düşürülmüştür.

Eğitim

- Veri setindeki her bir cümle Stanford Tokenizer ile tokenlarına ayrılır.
- Elde edilen tüm tokenlara karşılık gelen kelime vektörleri GloVe vektörlerinden bulunur.
- Bu kelime vektörleri de LSTM modeline girdi olarak verilir ve eğitim yapılır.

LSTM Modeli



Eğitim

- Loss fonksiyonu olarak CrossEntropyLoss
- Optimizer olarak Adam optimizer (L2 Regularization)

Eğitim

- Her fold'da, train/test classification reports ve confidence matrices bir listeye eklenir.
- Her fold'un en iyi modelleri sonuç olarak alınır.
- Örnek:
“es_n1+b32+e100+lr0.001+hidden256+ly2+bd+bn+dp_h0.2+ dp_o0.4.pth”

Model Sonuçları

- Model değerlendirme dosyaları ise “fold 1.png”, “fold 2.png” ... “fold n.png” şeklindedir.

```
—es_n5+b128+e1000+lr0.001+hidden512+ly1+bd+dp_h0.0+dp_o0.9
  5 fold average.png
  fold 1.png
  fold 2.png
  fold 3.png
  fold 4.png
  fold 5.png
—es_n5+b128+e1000+lr0.003+hidden256+ly1+dp_h0.0+dp_o0.7
  5 fold average.png
  fold 1.png
  fold 2.png
  fold 3.png
  fold 4.png
  fold 5.png
—es_n5+b128+e1000+lr0.003+hidden256+ly2+dp_h0.3+dp_o0.9
  5 fold average.png
  fold 1.png
  fold 2.png
  fold 3.png
  fold 4.png
  fold 5.png
```

TESTLER

```
lrlist = [0.001, 0.003, 0.005]
batchsizelist = [8, 16, 32, 64, 128, 256]
hdlist = [70, 128, 256, 300, 512, 768, 1024]
lylist = [1, 2, 4]
bdlist = [True, False]
bnlist = [True, False]
dphlist = [0.0, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]
dpolist = [0, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9]
number_of_epochs = 1000
n_splits = 5
early_stopping_patience = 15
```

(Bu örnekte 60480 x 5 farklı model vardır.)

- Veri setindeki büyük-küçük harflerin performansa etkisi olmamıştır.
- Batch size 16 iken ve 2 layer'lı modellerde 64 ve 128 iken performans iyileşmiştir.
- Modelin bidirectional olup olmaması performansa gözle görülür bir etki etmemiştir.
- Learning rate 0.003 iken daha iyi sonuçlar elde edilmiştir.
- Modelin büyüklüğü arttıkça başarı düşmüştür.
- Layer sayısı arttıkça performans olumsuz etkilenmiştir.
- Dropout arttıkça performans da artmıştır.

SONUÇLAR

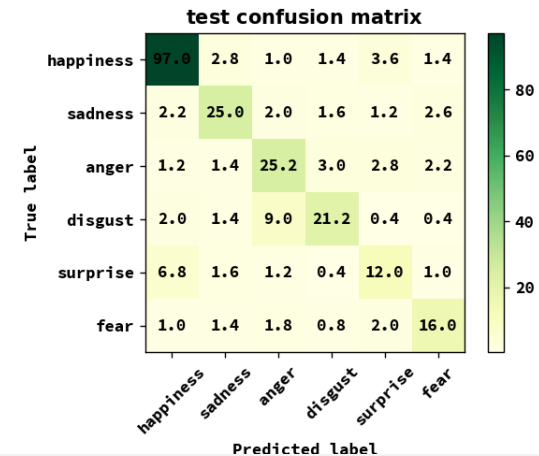
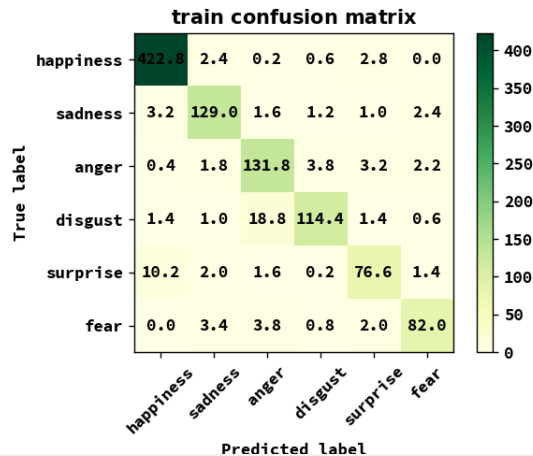
Alınan En Başarılı 5-Fold Crossover Sonuçları

LEARNING RATE = 0.003
BATCH SIZE = 16
HIDDEN_SIZE = 512
1 LAYERS
BIDIRECTIONAL NO
BATCHNORM NO
DROPOUT HIDDEN = 0.0
DROPOUT OUTPUT = 0.9

5 FOLD AVERAGE
EPOCH 28

	TRAIN DATA			
	precision	recall	f1-score	support
happiness	0.968	0.986	0.976	428.800
sadness	0.924	0.930	0.926	138.400
anger	0.846	0.922	0.878	143.200
disgust	0.946	0.832	0.882	137.600
surprise	0.876	0.834	0.842	92.000
fear	0.932	0.892	0.910	92.000
micro avg	0.928	0.928	0.928	1032.000
macro avg	0.914	0.900	0.904	1032.000
weighted avg	0.930	0.928	0.928	1032.000

	TEST DATA			
	precision	recall	f1-score	support
happiness	0.884	0.908	0.892	107.200
sadness	0.746	0.722	0.734	34.600
anger	0.632	0.700	0.658	35.800
disgust	0.752	0.620	0.676	34.400
surprise	0.558	0.522	0.524	23.000
fear	0.684	0.696	0.686	23.000
micro avg	0.760	0.760	0.760	258.000
macro avg	0.708	0.694	0.696	258.000
weighted avg	0.766	0.760	0.756	258.000



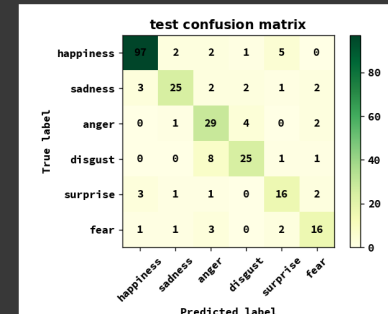
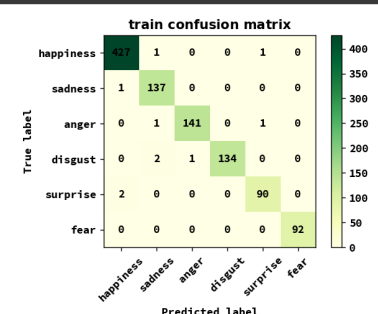
En Başarılı Fold'un Sonucu



LEARNING RATE = 0.003
BATCH SIZE = 16
HIDDEN_SIZE = 256
1 LAYERS
BIDIRECTIONAL NO
BATCHNORM NO
DROPOUT HIDDEN = 0.0
DROPOUT OUTPUT = 0.7

FOLD 2/5
EPOCH 26/1000

	TRAIN DATA					TEST DATA			
	precision	recall	f1-score	support		precision	recall	f1-score	support
happiness	0.99	1.00	0.99	429	happiness	0.93	0.91	0.92	107
sadness	0.97	0.99	0.98	138	sadness	0.83	0.71	0.77	35
anger	0.99	0.99	0.99	143	anger	0.64	0.81	0.72	36
disgust	1.00	0.98	0.99	137	disgust	0.78	0.71	0.75	35
surprise	0.98	0.98	0.98	92	surprise	0.64	0.70	0.67	23
fear	1.00	1.00	1.00	92	fear	0.70	0.70	0.70	23
micro avg	0.99	0.99	0.99	1031	micro avg	0.80	0.80	0.80	259
macro avg	0.99	0.99	0.99	1031	macro avg	0.75	0.76	0.75	259
weighted avg	0.99	0.99	0.99	1031	weighted avg	0.81	0.80	0.81	259



DEĞERLENDİRME

Literatürdeki Benzer Çalışmaların Karşılaştırılması

- Arora'nın ISEAR veri setinde test sonuçları ile bizim Aman veri setindeki test sonuçlarımız
Arora'nın en başarılı LSTM modeli: 0.64 accuracy
Bizim en başarılı LSTM modelimiz: 0.75 accuracy
- ISEAR veri setinin tamamında (guilt ve shame hariç)
Agrawal'ın en başarılı modeli: 0.45 avg f-score
Bizim en başarılı modelimiz: 0.44 avg f-score
- Aman'ın veri setinde (test sonuçlarında)
Aman'ın en başarılı ML modeli: 0.58 avg f-score
Bizim en başarılı modelimiz: 0.70 avg f-score

Demo

Kaynakça

- S. Aman & S. Szpakowicz, 2007, Identifying Expressions of Emotion in Text, V. Matousek, P. Mautner (eds.): Proc 10th International Conf. on Text, Speech and Dialogue TSD 2007, Plzeň, Czech Republic, Lecture Notes in Computer Science 4629, Springer, 196-205.
- S. Aman, 2007, Recognizing Emotions in Text, Master of Computer Science, University of Ottawa.
- S. Hochreiter & J. Schmidhuber, 1997, Long short-term memory, Neural computation, 9(8):1735–1780.
- J. Yang, J. Leskovec, 2011, Patterns of Temporal Variation in Online Media, ACM International Conference on Web Search and Data Mining (WSDM '11)
- Ebba Cecilia Ovesdotter Alm, 2008, Affect in text and speech, PhD Dissertation, Urbana, IL: University of Illinois at Urbana-Champaign.
- Ameeta Agrawal, 2011, Unsupervised emotion detection from text using semantic and syntactic relations, 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, pages 346-353. IEEE Computer Society, December.
- Jeffrey Pennington et al., 2014, GloVe: Global Vectors for Word Representation, Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014) 12

Teşekkür Ederiz

Sorularınız?