



DERİN ÖĞRENME YAKLAŞIMI İLE DUYGU DURUM ANALİZİ (KERAS ile LSTM UYGULAMASI)



GİRİŞ

Sözlük Tabanlı Yaklaşımlar

Sözlük tabanlı sistemlerde bir ön bilgi ve karar verme mekanizması olarak sözlük eklenmektedir. Sözlüğün büyüklüğü duygu-durum analizi çalışmalarının başarılarını etkilemekle birlikte, yük olarak sistemde bulunmaktadır. Daha kapsamlı sözlüklerle daha başarılı sonuçlar elde edilebilmektedir.

Makine Öğrenmesi Yaklaşımı

Makine öğrenmesi yaklaşımlarını kullanan uygulamalarda eğitim ve test verileri bulundurulmalıdır. Sistemin öğrenebilmesi için yeterli büyüklükte bir veri kümesi etiketlenmiş olarak sağlanmalıdır. Bu tarz yaklaşımdaki en büyük sorun aşırı öğrenme(overfitting) sonucu yeni verilere adaptasyonun zor olmasıdır.

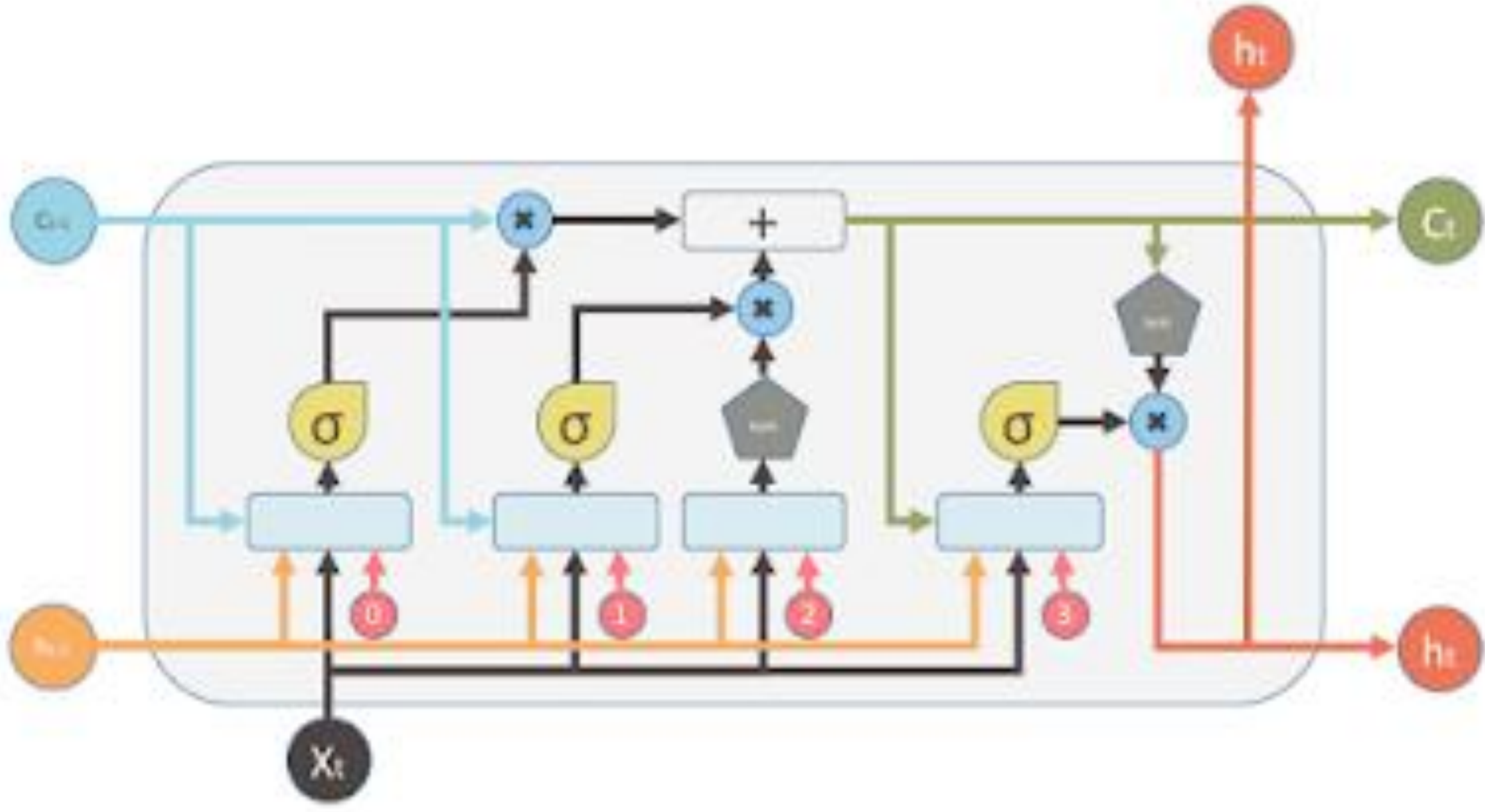
Derin Öğrenme Yaklaşımı

Derin öğrenme yaklaşımları, ilk olarak görüntü üzerinde uygulamaya başlanmış, son birkaç yıl içerisinde ise metin üzerinde başarılı olduğu anlaşılmıştır. Diğer öğrenme metodlarına göre eğitim aşaması uzun sürse de, bulut ve GPU desteği ile bu sorun aşılmış olup, özellikle Google ve Amazon tarafından sunulan bu ortamlar ve geliştirici araçları ile hızlı bir şekilde eğitim aşaması tamamlanabilmektedir. Bağımlı kütüphaneler de yeniden yüklenmeye gerek kalmadan kullanılabilirlerdir.

Sıralılık örüntüsü -Sequence pattern- içeren durumlarda LSTM daha başarılıdır.

Özellikle makine öğrenmesi algoritmalarının zorlandığı doğal dil uygulamalarında derin öğrenme yaklaşımı ile başarılı sonuçlar elde edilebilmektedir.

Güçlü yanı olan nokta çarpımı ile hızlı türev almalar yapabilmesi ve veriyi güzel tanıyabilmeye ek olarak, LSTM kapılar vasıtası ile gereksiz bilgiyi unutma; eski bilgileri tutabilme(zaman faktörü) ve bilgiyi güncelleme özelliklerine sahiptir.



LSTM KAPILARI

LSTM ağlarının en önemli özellikleri:

- Giriş Kapısı (tutulacak bilgilerin belirlenmesi)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- Unutma Kapısı (gerekli olmayan bilgilerin silinmesi)

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- Çıkış Kapısı (başka üniteye girdi olarak veya çıktı olarak kullanım)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

SONUÇLAR

Tweetler üzerinden eğitilen LSTM ağları ile % 84.78 başarı sağlanmıştır. Veriseti olarak Amerika seçimlerinde twitterda #GOPDebates etiketli tweetlerden oluşan 16649 satırlık CSV dosyası kullanılmıştır

```

Epoch 1/7
7188/7188 [=====] - 19s 3ms/step - loss: 0.4364 - acc: 0.8151
Epoch 2/7
7188/7188 [=====] - 18s 2ms/step - loss: 0.3287 - acc: 0.8662
Epoch 3/7
4896/7188 [=====] - ETA: 7s - loss: 0.2858 - acc: 0.88097188/7188 [=====] - 18s 2ms/step - loss: 0.2809 -
Epoch 4/7
7188/7188 [=====] - 18s 2ms/step - loss: 0.2587 - acc: 0.9000
Epoch 5/7
5984/7188 [=====] - ETA: 2s - loss: 0.2284 - acc: 0.91067188/7188 [=====] - 18s 2ms/step - loss: 0.2298 -
Epoch 6/7
7188/7188 [=====] - 18s 2ms/step - loss: 0.1987 - acc: 0.9218
Epoch 7/7
6528/7188 [=====] - ETA: 1s - loss: 0.1912 - acc: 0.92367188/7188 [=====] - 18s 2ms/step - loss: 0.1908 -
<keras.callbacks.History at 0x7f7fb6bcf3c8>

score,accuracy = model.evaluate(X_test, Y_test, verbose = 2, batch_size = batch_size)
print("score: {:.2f}%".format(score*100))
print("Accuracy: {:.2f}%".format(accuracy*100))

score: 40.80%
Accuracy: 84.78%

[12] tweet = ['Since I buy that crazy game notebook, I can not attend any meeting and can not afford any more spending this week.']
vectorizing the tweet by the pre-fitted tokenizer instance
tweet = tokenizer.texts_to_sequences(tweet)
padding the tweet to have exactly the same shape as 'embedding_2' input
tweet = pad_sequences(tweet, maxlen=28, dtype='int32', value=0)
print(tweet)
sentiment = model.predict(tweet, batch_size=1, verbose = 2)[0]
if(np.argmax(sentiment) == 0):
    print("Your tweet is 'Negative'")
elif(np.argmax(sentiment) == 1):
    print("Your tweet is 'Positive'")

Your tweet is 'Negative'
    
```

DEMO

Aşağıdaki parametrelerle Amerikan seçimlerine ait twitter verileri ile ingilizce dili üzerinde eğitilen LSTM ağları %84.78 başarı sağlamıştır:

Öğrenme katsayısı (learning rate) : 0.001

Maksimum özellik sayısı : 2000

Drop out : 0.2

Toplu iş boyutu (batch size) : 32

Input boyutu (embedding dimension) : 128

Output boyutu (Lstm output dimension) : 196

GELECEK ÇALIŞMA

LSTM ağları ile konvolusyonel ağların hibrit sistem olarak tasarlanması ile %90 doğruluk oranları ile daha başarılı sonuçlar elde edilebilir.

REFERANSLAR

*Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." In Acoustics, speech and signal processing (icassp), 2013 IEEE International Conference on, pp. 6645-6649. IEEE, 2013. <http://ieeexplore.ieee.org/abstract/document/6638947>.

*Amodei, Dario, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." ArXiv preprint ArXiv:1512.02595 (2015).

* <https://github.com/oxford-cs-deeprnlp-2017/lectures> Deep Learning Natural Language Processing Lectures, Practicals Overview, Oxford, 2017.

*Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., and Potts, C., Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. Stanford, 2014. <http://nlp.stanford.edu/~socherr>

*<http://colah.github.io/posts/2015-08-Understanding-LSTMs>

