YAPAY SİNİR AĞLARI, KELİME VEKTÖRLERİ VE DERİN ÖĞRENME METOTLARI İLE UYGULAMA ÖRNEKLERİ

Sedrettin ÇALIŞKAN[1]

ÖZET

Bu çalışmada, yapay sinir ağları, kelime vektörleri ve derin öğrenme metotları ile yapılan uygulamalardan bahsedilmiştir. Bu çalışmalarda kullanılan metodolojiler ve veri setleri açıklanmıştır. Yapay sinir ağları kullanılarak yapılan geri yayımlı algoritma iyileştirmesi, derin öğrenme metotları kullanarak twitter veri analizi ve kelime vektörleri kullanılarak verilen metin içerisinde geçen kelimelerin birbirleri ile olan ilişkilerinin görselleştirilmesi ve türkçe metinlerin anlam analizinin gerçekleştirilmesi anlatılmıştır. Yapılan farklı çalışmaların sonuçları, parametre analizi ve kaynak kodları makale içerisinde ve ekler kısmına konumlandırılmış bir şekilde paylaşılmıştır.

Anahtar Kelimeler: Yapay sinir ağları, geri yayımlı, kelime vektörleri, twitter, derin öğrenme.

1. GİRİŞ

Yapay sinir ağları, insan beyninin sinir hücrelerinden oluşmuş katmanlı ve paralel olan yapısının, tüm fonksiyonlarıyla beraber sayısal dünyada gerçeklenmeye çalışılan modellenmesidir. Sayısal dünya ile belirtilmek istenen donanım ve yazılımdır. Bir başka ifadeyle yapay sinir ağı hem donanımsal olarak hemde yazılım ile modellenebilir. Bu bağlamda, yapay sinir ağları ilk elektronik devreler yardımıyla kurulmaya çalışılmış ancak bu girişimi kendini yavaş yavaş yazılım sahasına bırakmıştır. Böylesi bir kısıtlanmanın sebebi; elektronik devrelerin esnek ve dinamik olarak değiştirilememesi ve birbirinden farklı olan ünitelerin bir araya getirilememesi olarak ortaya konmaktadır.[1]

Derin Öğrenme , Makine Öğrenmesi (Machine Learning) tekniklerinden sadece biri. Genel olarak kastedilen şey ise çok katmanlı Yapay Sinir Ağları'ndan başka bir şey değil. Yapay Sinir Ağları'nı temel alan sistemler önce Ses ve Konuşma Tanıma alanında mevcut sistemlerden daha iyi performans göstermeye ve hayatımıza girmeye başladılar. Aynı şekilde bugün cep telefonlarımız

[1] Mühendislik ve Fen Bilimleri Enstitüsü, Fatih Sultan Mehmet Vakıf Üniversitesi, Beyoğlu, İstanbul.
sdrttnclskn@gmail.com

sesli komutları anlarken bu teknolojiden yararlanıyor. Öz olarak derin öğrenme, büyük verinin hızlı işlemcilerde yapay zeka teknikleri ile işlenip bilgiye dönüştürülmesidir.

Word2Vec , kelimeleri vektör uzayında ifade etmeye çalışan, tahmin temelli bir modeldir.
Kelimelerin bilgisayarın anlayacağı şekilde vektörler halinde temsil edilmesini ve kelimeler arasındaki uzaklığı vektörel olarak hesaplamanızı sağlayan bir algoritma araç kitidir. Bu vektörel yapının üzerine yazılmış araçlar ile bir kelimeye en yakın kelimleri listeletebilirsiniz. Kelimeler arası ilişki kurabilirsiniz.[2]
Yapılan bu çalışmada Yapay sinir ağları kullanılarak Geri Yayılımlı Yapay Sinir Ağları Hata fonksiyonunu ve toplam hatayı minimuma yaklaştırma çalışmaları yapılmaktadır. Derin Öğrenme ile Twitter Duyarlılık Analizi yapılmakta ve Kelime Vektörleri ile de metin analizi yapılmaktadır.

Bölüm 2'de Kullanılan yöntemlerden ,Yapay Sinir Ağları, Kelime Vektörleri tanımılar ve kullanımlarından bahsedilmiştir.. **Bölüm 3**'de Yapılan uygulamlar ve uygulama sonuçlarına dair anlatımlar yapılmıştır.

2. KULLANILAN YÖNTEMLER

2.1 Yapay Sinir Ağları(YSA)

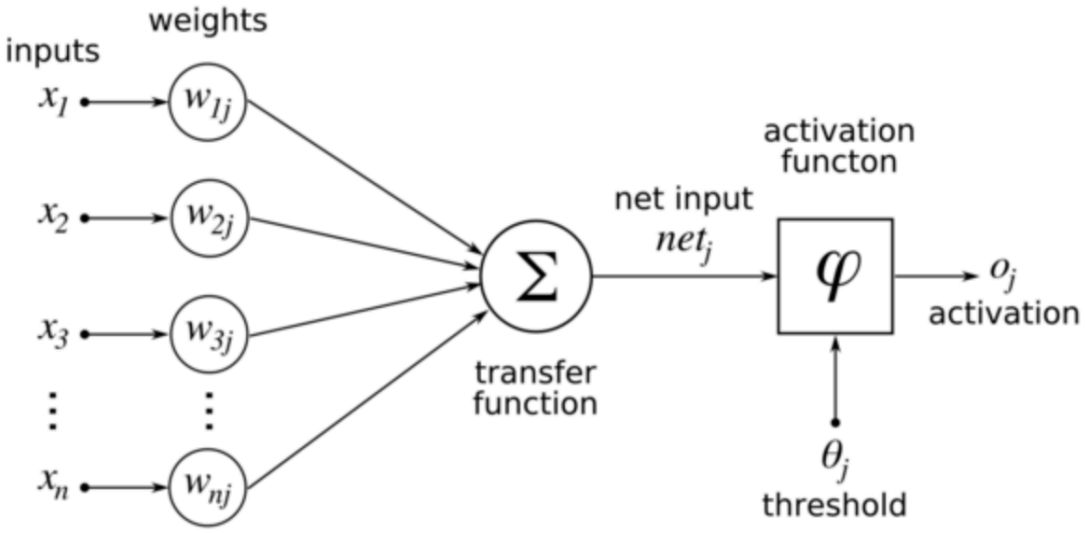
İlk yapay nöron, 1943 yılında nöropsikiyatrist Warren McCulloch ve bilim adamı Walter Pits tarafından üretilmiştir. Ancak dönemin kısıtlı olanakları nedeniyle, bu alanda çok gelişme sağlanamamıştır. Bundan sonra 1969'da Minsky ve Papert bir kitap yayınlayarak, yapay sinir ağları alanında duyulan etik kaygıları da ortadan kaldırmış ve bu yeni teknolojiye giden yolu açmışlardır. İlk gözle görülür gelişmeler ise 1990'lı yıllara dayanmaktadır.[4]

Genel anlamda YSA, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar halinde düzenlenir. Donanım olarak elektronik devrelerle veya bilgisayarlarda yazılım olarak gerçekleşebilir. Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir [3]. Turing makineleriyle temeli atılan yapay zeka üzerinde en fazla

araştırma yapılan konu “Yapay Sinir Ağları”dır. Yapay sinir ağları, temelde tamamen insan beyni örneklenerek geliştirilmiş bir teknolojidir .Bir sinir ağı, bilgiyi depolamak ve onu kullanışlı hale getirmek için doğal eğilimi olan basit birimlerden oluşan paralel dağıtılmış bir işlemcidir. İnsan beyni ile iki şekilde benzerlik göstermektedir: 1.Bilgi, öğrenme süreci yoluyla ağ tarafından elde edilir. 2. Sinaptik ağırlıklar olarak bilinen nöronlar arası bağlantı kuvvetlerini, bilgiyi depolamak için kullanır .[5]

Yapay sinir ağları başlıca; Sınıflandırma, Modelleme ve Tahmin uygulamaları olmak üzere, pek çok alanda kullanılmaktadır. Başarılı uygulamalar incelendiğinde, YSA'ların çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek sensör verilerinin olması ve problemi çözmek için matematiksel modelin ve algoritmaların bulunmadığı, sadece örneklerin var olduğu durumlarda yaygın olarak kullanıldıkları görülmektedir. Bu amaçla geliştirilmiş ağlar genellikle şu fonksiyonları gerçekleştirmektedirler; Muhtemel fonksiyon kestirimleri, sınıflandırma, ilişkilendirme veya örüntü eşleştirme, zaman serileri analizleri, sinyal filtreleme, veri sıkıştırma, örüntü tanıma, doğrusal olmayan sinyal işleme, doğrusal olmayan sistem modelleme, optimizasyon, Kontrol YSA'lar pek çok sektörde değişik uygulama alanları bulmuştur.[6]

2.1.1 Yapay Sinir Ağı Yapısı

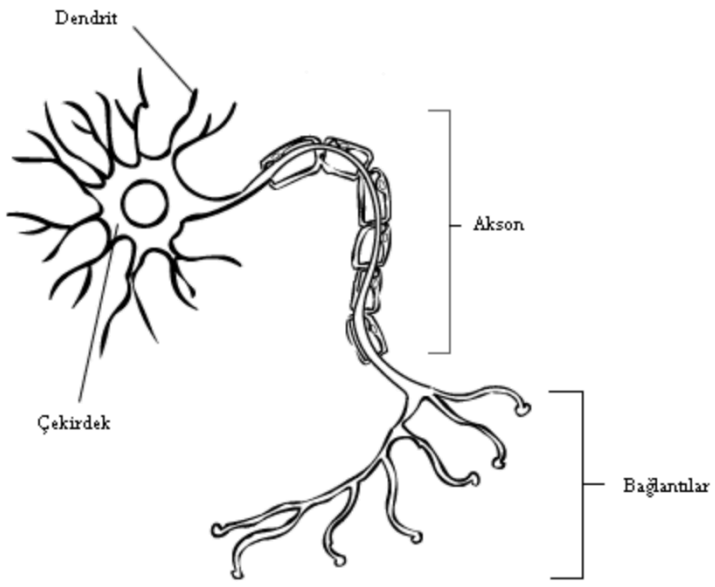


Şekil 1: Yapay sinir hücre yapısı

YSA, insan beyninin çalışma mekanizmasını taklit ederek beynin öğrenme, hatırlama genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilen mantıksal yazılımlardır. YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır.YSA, biyolojik sinir ağları taklit eden sentetik ağlardır. Yapay Sinir Ağı modelindeki terimler (Tablo 1) [7]

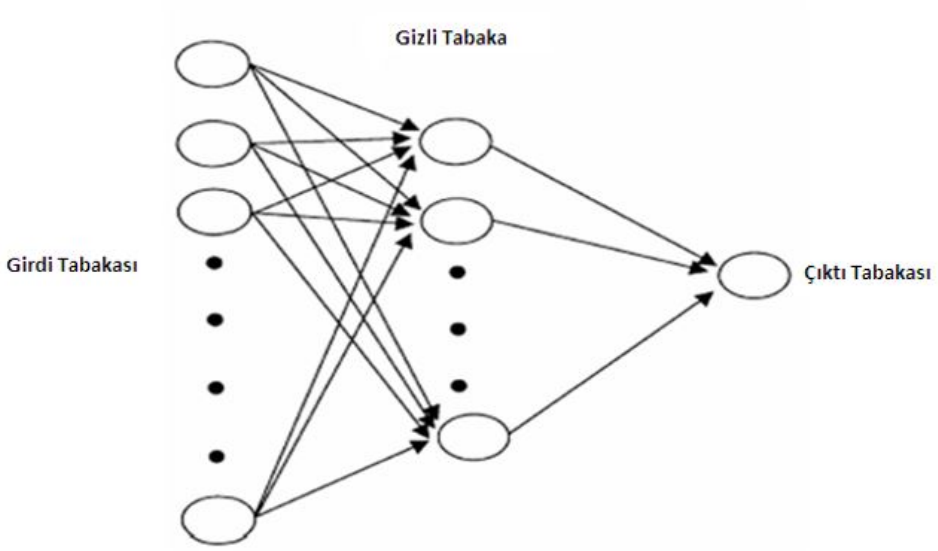
Tablo 1: Yapay Sinir Ağı modelindeki terminolojisi

Sinir Sistemi	Yapay Sinir Ağı
Nöron	İşlem Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar



Şekil 2: Biyolojik sinir hücre yapısı

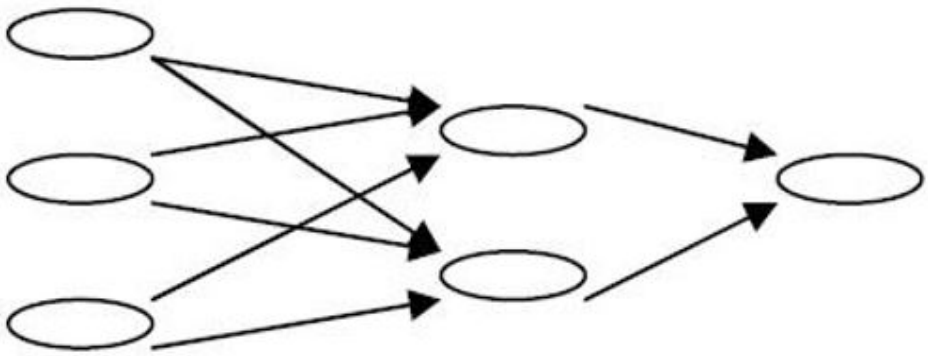
2.1.2 Yapay Sinir Ağlarının Mimari Yapısı



Şekil 3: Mimari Yapısı

Girdi, Gizli ve Çıktı tabakalarından oluşan 3 tabakalı (ya da katmanlı) ileri beslemeli bir sinir ağı modeli görülmektedir.[8]

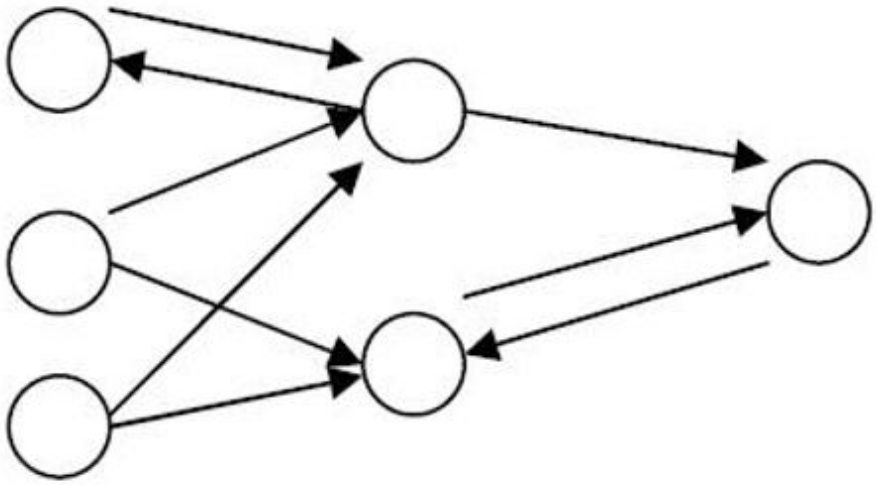
İleri Beslemeli YSA Modeli



- Tek yönlü sinyal akışı için izin verir.
- İleri beslemeli yapay sinir ağında, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir.

- Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan ara (gizli) katmandaki hücrelere iletir.
- Gizli ve çıktı tabakalarından bilginin işlenmesi ile çıkış değeri belirlenir.

Geri Beslemeli YSA Modeli

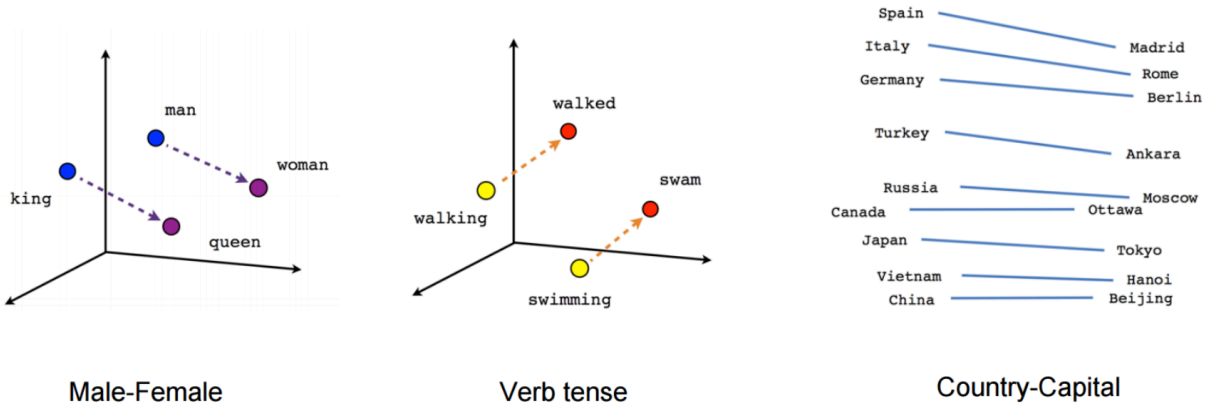


- Geri beslemeli Yapay Sinir Ağları (YSA)' da, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır.
- Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir.
- Bu çeşit sinir ağlarının dinamik hafızaları vardır. Bu yapıdaki nöronların çıkışı sadece o anki giriş değerlerine bağlı değildir ayrıca önceki giriş değerlerine de bağlıdır. Bundan dolayı, bu ağ yapısı özelikle tahmin uygulamaları için uygundur. Bu ağlar özellikle çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamıştır [9]

2.2 Kelime Vektörleri (Word2Vec)

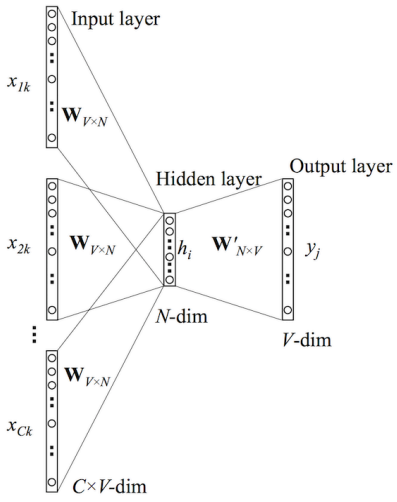
Word2Vec , kelimeleri vektör uzayında ifade etmeye çalışan ve tahmin temelli bir modeldir. Google araştırmacı Tomas Mikolov ve ekibi tarafından 2013 yılında icat edilmiştir. İki çeşit alt yöntemi

vardır: CBOW(Continous Bag of Words) ve Skip-Gram. iki yöntem de genel olarak birbirine benzemektedir. Word2Vec kelimeler arasındaki uzaklığı vektörel olarak hesaplamanızı sağlayan bir algoritma araç kitidir. Bu vektörel yapının üzerine yazılmış araçlar ile bir kelimeye en yakın kelimleri listeletebilirsiniz. Kelimeler arası ilişki kurabilirsiniz.



Şekil 4: Üç boyutlu uzayda Word2Vec örneği [10]

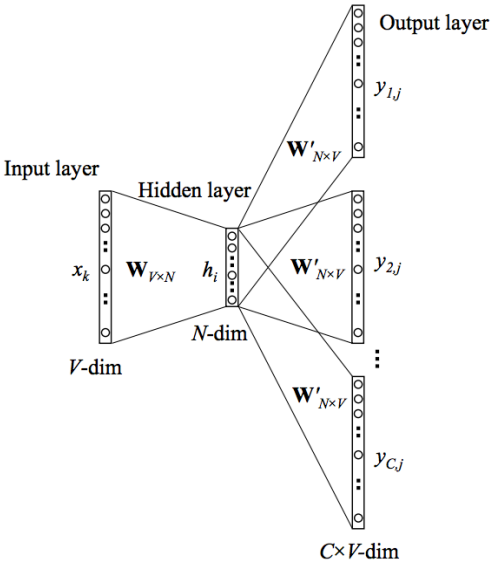
Word2Vec kelimelerin vektörel temsili için iki farklı model mimariden birini kullanabilir: CBOW ve skip-gram. CBOW ve Skip-Gram modelleri birbirlerinden output'u ve input'u alma açısından farklılaşmaktadır.



Şekil 5: CBOW Yapısı[11]

Örnek cümle: "*Gözümün önünden yorgun insanlar geçiyordu birer birer.* " Bu cümleyi input olarak alan ve ***window_size***=1 olan CBOW modeli çalışma süreci: Önce "Gözümün" kelimesini window'un merkezine oturtuyor, sonra sağındaki ve solundaki 1'er kelimeyi ayrı ayrı input olarak alıp (çünkü ***window_size***=1) merkeze oturttuğu "Gözümün" kelimesini Neural Network modeli ile

tahmin etmeye çalışmaktadır. Sonra ***window***'u 1 sağa kaydırmakta, bu sefer window'un merkezine "önünden" kelimesi gelmektedir.



Şekil 6: Skip-Gram Yapısı[11]

Aynı cümlenin ***window_size*** =1 olan Skip-gram modeli çalışma süreci: Merkezdeki kelime input olarak alınıp merkezdeki kelimeye ***window_size*** büyüklüğünden daha az yakın olan kelimeler tahmin edilmektedir. Skip-gram modeli ilk önce cümledeki "Gözümün" kelimesini merkeze oturtuyor ve "Gözümün" kelimesini kullanarak "önünden" kelimesini tahmin etmeye çalışıyor.

Sonuç olarak, CBOW modelleri genel yapısı gereği küçük datasetlerde daha iyi çalışırken, büyük datasetlerde Skip-gram daha iyi çalışmaktadır.

3. Uygulamalar

3.1 Yapay sinir ağları kullanılarak Geri Yayılımlı Yapay Sinir Ağları Uygulaması

Geri yayılımlı yapay sinir ağları iki temel aşamadan oluşur: İleri besleme ve geri yayılım. İleri besleme, ağa giriş verilerinin verildiği aşamadır. Bu aşamanın sonunda elde edilen çıkışlar hata fonksiyonuna girilir ve hatalar geriye yayılarak ağırlıklar güncellenir. Hata fonksiyonunu ve

dolayısıyla toplam hatayı minimuma yaklaştırmak için gradyan iniş metodu kullanılır. Uygulama kodlarının ekler bölümünde linkleri verilmiştir.(Ek - 1)

Veri Seti Özellikleri:

X : Her satırın bir eğitim örneği olduğu girdi veri kümesi matrisi.

y : Her satır bir eğitim örneği olduğu çıktı veri kümesi matrisi.

l0 : Giriş verisi tarafından belirtilen Ağın Birinci Katmanı

l1 : Ağın İkinci Katmanı, aksi halde gizli katman olarak bilinir.

syn0 : Ağırlıkların ilk katmanı, Synapse 0, l0'dan l1'e bağlanma.

* : Elementwise çarpma, böylece eşit boyuttaki iki vektör, özdeş boyutta bir nihai vektör oluşturmak için karşılık gelen 1-to-1 değerlerini çarpar.

- : Eşit boyuttaki son vektör oluşturmak için, eşit boyuttaki iki vektör, karşılık gelen değerleri 1'den 1'e çıkartarak, öğe temelinde çıkarma.

x.dot(y) : Eğer x ve y vektörler ise, bu bir noktalı bir üründür. Her ikisi de matris ise, matris-matris çarpımıdır. Tek bir matris ise, o zaman vektör matris çarpımı olur.

Algoritma Akışı:

Geri yayılım ağındaki öğrenme aşağıdaki adımlardan oluşur:

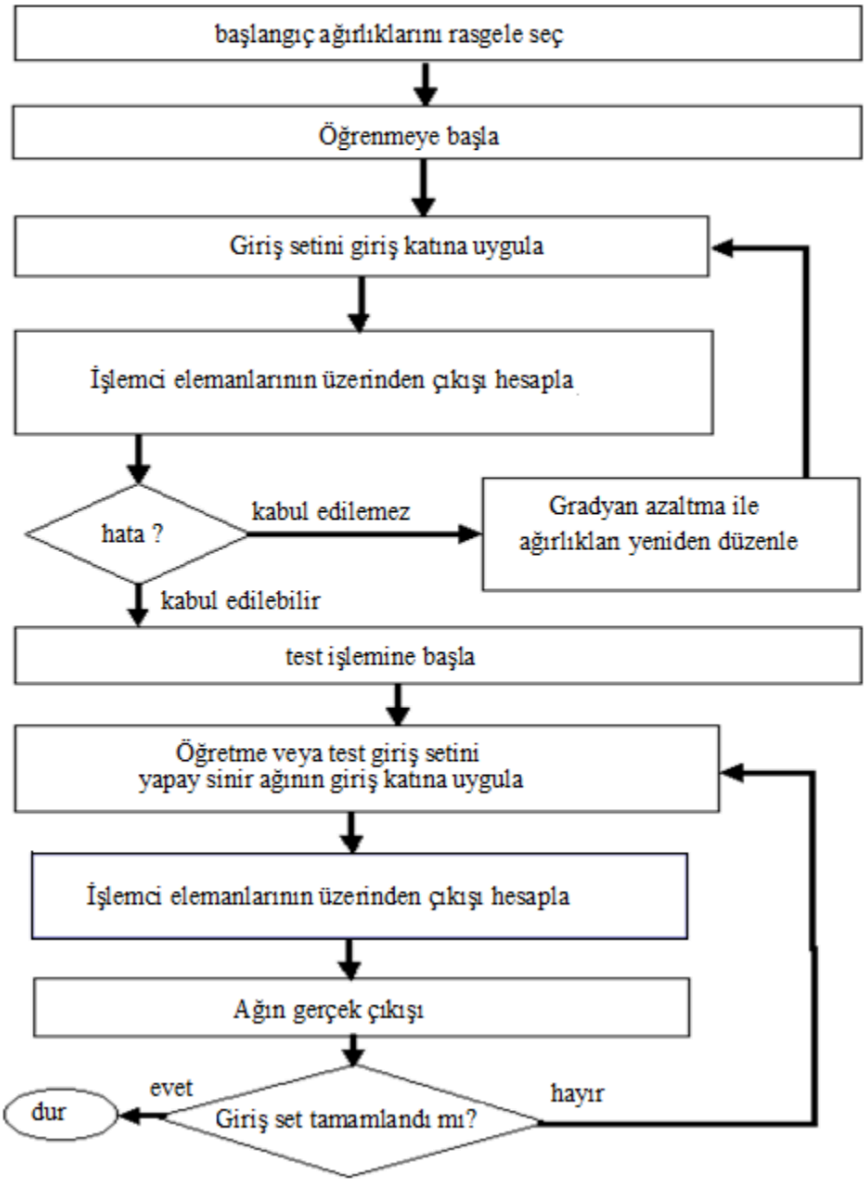
Adım 1. Eğitim kümesinden bir sonraki örneği seçme ve ağ girişine girdi vektörü uygulama.

Adım 2. Ağın çıktısını hesaplama.

Adım 3. Ağın çıktısı ile istenen hedef vektör arasındaki hatayı hesaplama.

Adım 4. Hatayı küçültecek şekilde ağın ağırlıklarını ayarlama.

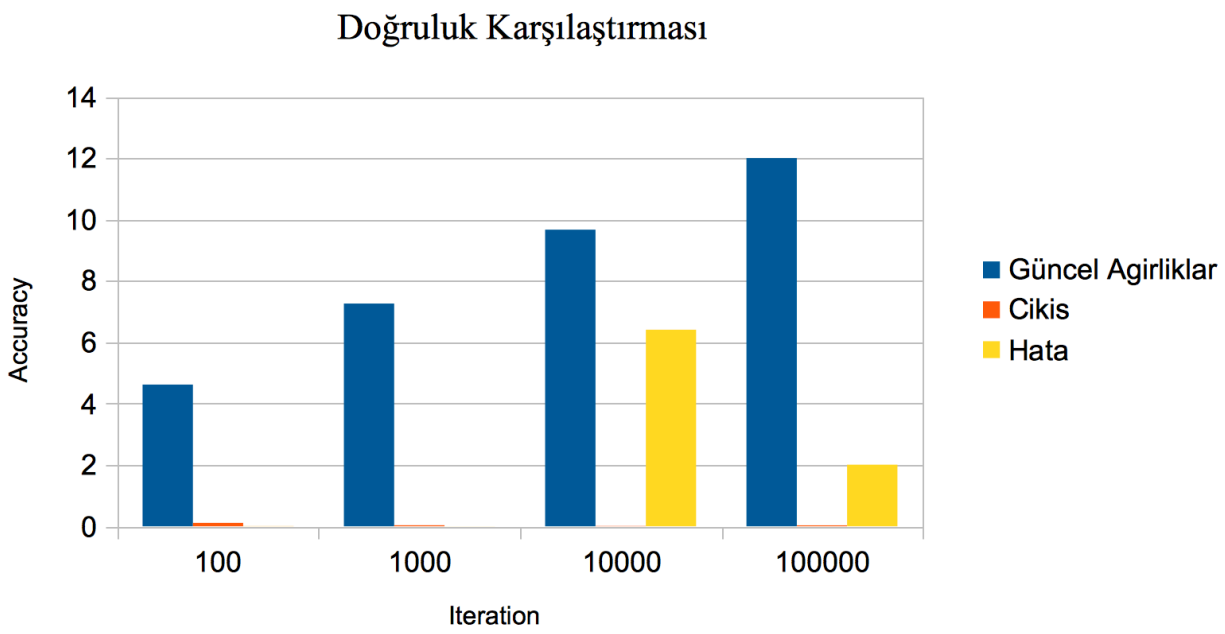
Çok katmanlı ileri beslemeli Y.S.A. modelindeki geri yayılım **Şekil 7'** de verilmiştir.



Şekil 7 : Geri Yayılımlı Yapay Sinir Ağları akış şeması [12]

Sonuçlar:

Tablo 2.'de bulunan değerler modelin eğitimi ve test edilmesi sırasında elde edilen doğruluk değerlerini göstermektedir.



Şekil 8: Eğitim doğruluk oranlarının karşılaştırılması

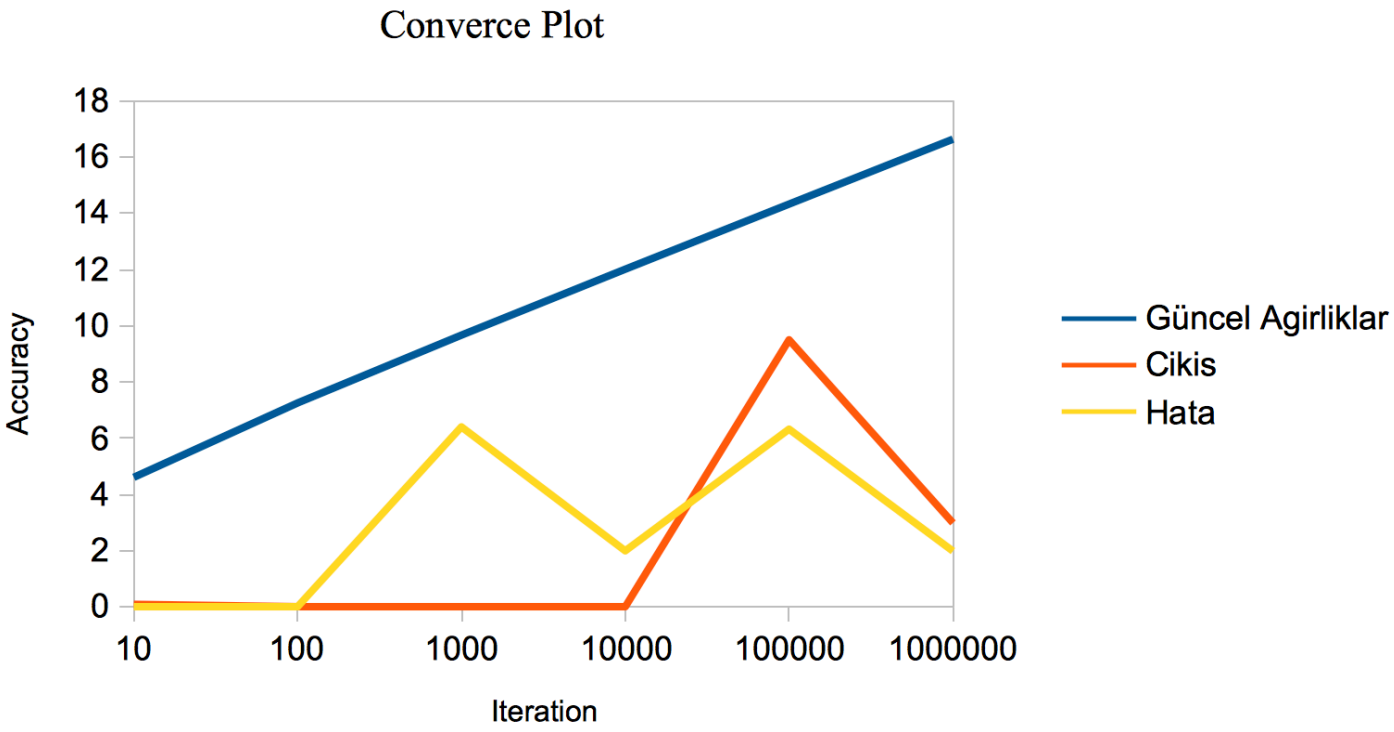
```
Güncel Agirliklar
[[ 9.67299303]
 [-0.2078435 ]
 [-4.62963669]]
Cikis
[[ 0.00966449]
 [ 0.00786506]
 [ 0.99358898]
 [ 0.99211957]]
Hata
[[ -9.66449169e-05]
 [ -7.86505967e-05]
 [  6.41101950e-05]
 [  7.88042678e-05]]
```

Güncel ağırlıklarımız: Öğrendiğimiz değerler

Çıkış değerleri : Sistemin kendi öğrenme değerleri

Hata : Kendi öğrenme sürecindeki hata payı

Şekil 9: 1000 iteration ‘da doğruluk değerleri



Şekil 10: Eğitim doğruluk oranlarının yakınsama grafiği

3.2 Derin Öğrenme Metotları kullanılarak Twitter Duyarlılık Analizi Uygulaması

Bu uygulamada belirtilen anlam kümesi ile ilgili atılan tweet'lerin anlamsal analizi yapılacaktır. Verilen cümlenin olumlu yada olumsuz bir cümle olduğu sınıflandırılacaktır. Yapılan çalışmanın uygulama kodu ekler bölümünde bulunmaktadır.(Ek- 2)

Veri Seti Özelikleri

wiki = TextBlobg("") : Twitter da aranacak olacak kelime dizisi

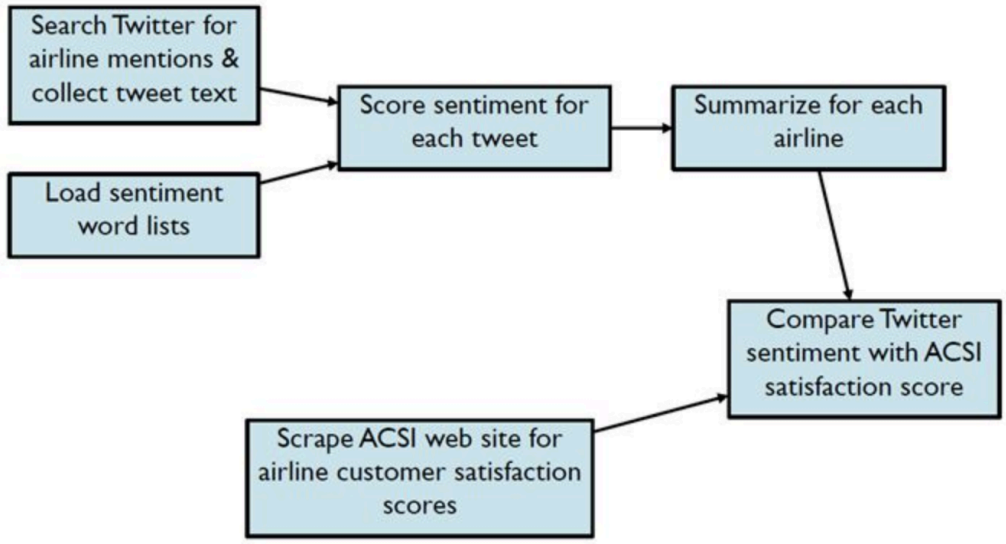
wiki.tags : Twitter da arancak olan kelime dizisin string'lere ayırıp tag'ler haline getirilmesi.

wiki.words : Dizi string 'leri.

wiki.sentiment.polarty = -1 < sentiment < 1 : Duyarlılık aralığımız.

public_tweets = api.serach(''): Tweet oluşturma, tweet silme ve twitter kullancısı bulma parametreleri kullanılabilir.

Akış Diyagramı



Şekil 11: Twitter duyarlılık analiz akış şeması[13]

Sonuçlar:

Örnek alınan kelime dizisi , wiki = TextBlobg (“İlem demek ilmi geleneğin yuvası demektir.”).

Kelime dizi ile ilgili bir kelime(search (‘ilem’)) aradığımızda bululan tweet kümesi aşağıdaki gibidir.

```
-

Yer: İlmi Etüdler Derneği
Zaman: YARIN (13 Ocak Cumartesi) - 17:30

Engin Koca, Yeni doğa felsefesini…
Sentiment(polarity=0.0, subjectivity=0.0)

İLEM SUNUMLARI 131

Yer: İlmi Etüdler Derneği
Zaman: YARIN (13 Ocak Cumartesi) - 17:30

Engin Koca, Yeni doğa fels… https://t.co/PXD0Ul1Mxr
Sentiment(polarity=0.0, subjectivity=0.0)

RT @ilemihtisas: İLEM SUNUMLARINDA YARIN (13 Ocak Cumartesi), 17:30

"Modern Fiziğin Doğuşu"

https://t.co/e8kUTIxecP

#ilemsunumları #ile…
Sentiment(polarity=0.2, subjectivity=0.3)

RT @ilemihtisas: İLEM SUNUMLARINDA YARIN (13 Ocak Cumartesi), 17:30

"Modern Fiziğin Doğuşu"

https://t.co/e8kUTIxecP

#ilemsunumları #ile…
Sentiment(polarity=0.2, subjectivity=0.3)
```

#ilem ile ilgili atılan tweet’ler de olumlu ,olumsuz mesajlarda **polarity** ve **subjectivity** -1 ile 1 arasında değiştiğini görüyoruz.

3.3 Kelime Vektörleri (Word2Vec) ile Metin Analizi Uygulaması

Bu çalışmada kelime vektörleri kullanılarak verilen metinlerin analizi gerçekleştirilmiştir. Metin içerisinde geçen kelimelerin birbirleri ile olan ilişkileri hesaplanarak tahminlerde bulunulmuş ve kelimeler arası uzaklık ve yakınlıklar grafik üzerinde görselleştirilmiştir. Yapılan çalışmanın uygulama kodu ekler bölümünde bulunmaktadır.(Ek- 3)

Veri Seti Özellikleri:

Veri seti olarak *"Şakir KOCABAŞ'ın, Yapay Zeka ve Bilim Felsefesi"* makalesi kullanılmıştır. **Word2Vec** Parametreleri:

1. **Sg**: Eğitim algoritmasını tanımlar. Ön Tanımlı olarak "0" verilir. "0" ise "CBOW" yöntemi, "1" ise "skip-gram" yöntemi kullanılır. Çalışmamızda "skip-gram" kullanılmıştır.

2. **Seed**: Rastgele sayı üretir.

3. **Workers**: Modeli eğitmek için kullanılacak iş parçacıkları sayısı.

4. **Size**: Özellik vektörünün boyutudur. Kelimelerin temsil edileceği vektör boyutu.

5. **Min_count**: Toplam frekanstan düşük olan bütün kelimeler yok sayılır.

6. **Window**: Cümle içerisinde geçerli ve tahmin edilen kelime arasındaki maksimum uzaklıktır. Yani seçilen kelime ile ilişkili olan kelimeler aranırken, seçilen kelimenin sağındaki ve solundaki kelimelerden kaçar tanesinin inceleneceğini belirtir.

Algoritma Akışı:

1. "Data" klasöründen "Ş*akir KOCABAŞ*" makalesinin olduğu veri seti okunmaktadır.

2. Veri seti istenmeyen karakterlerden temizlenir. Sadece harf ve rakamlar tutulur. Özel karakterler veri setinden temizlenmektedir.

3. Veri seti içerisindeki cümleler analiz edilerek, cümlelerde geçen bütün kelimeler ayrıştırılır ve elde edilen kelimelerin frekans değerleri belirlenmektedir.

4. Word2Vec eğitim modeli “Word2Vec Parametreleri” bölümünde belirtilen değerlere göre oluşturulmaktadır.

5. Oluşturulan model örnek veri seti ile eğitilmektedir.

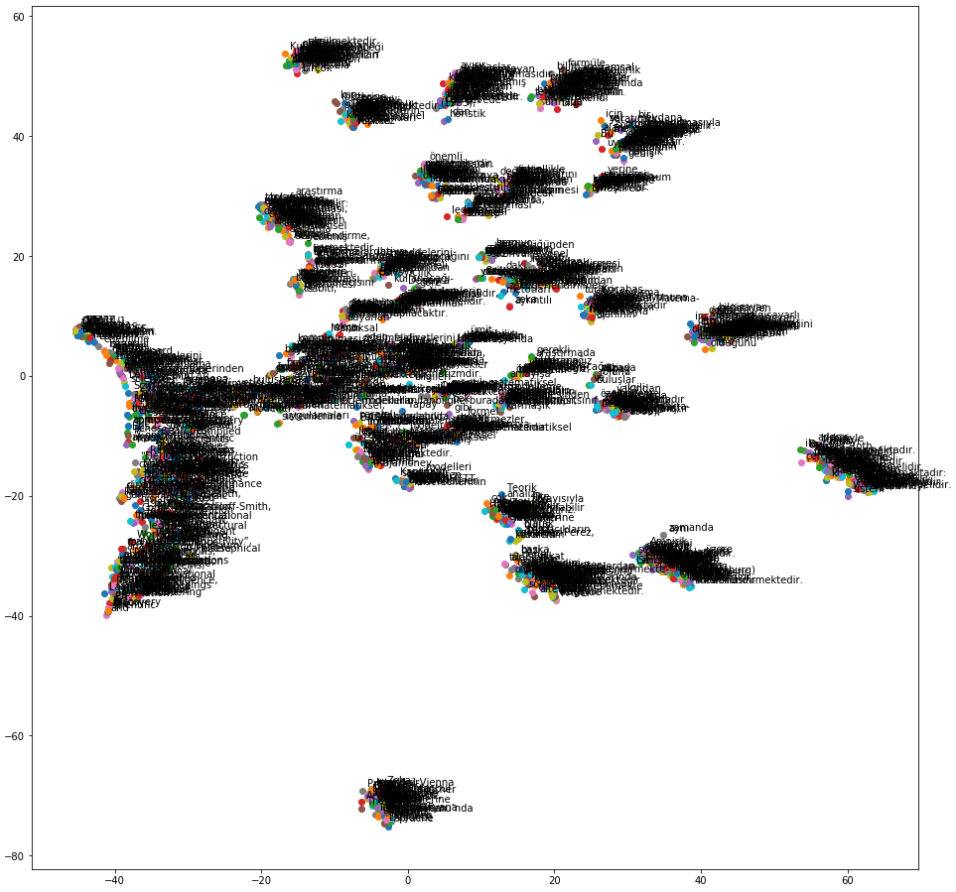
6. Model eğitildikten sonra veri setinde geçen bütün kelimelerin grafik üzerinde gösterilmek üzere “x” ve “y” noktaları belirlenir. Yani bütün kelimelerin birbirleri ile olan yakınlık ve uzaklık ilişkileri belirlenmiş olmaktadır.

7. Kelimeler için belirlenen “x” ve “y” noktaları koordinat düzlemine yerleştirilmektedir.

8. Kelimeler arasındaki ilişkiler grafik üzerinde görselleştirilmektedir.

9. İstenirse elle verilen kelime ile benzerlik gösteren kelimeleri belirlemek üzere benzerlik (similarity) fonksiyonu kullanılmaktadır.

Sonuçlar:



Şekil 12: Veri setinde kelimelerin birbirleri ile olan ilişkilerinin genel görünümü.

Bu çalışmada kullanılan kelimelerin birbirleri ile olan benzerlikleri, farklılıkları, verilen iki kelimenin birbirine ne kadar benzer oldukları belirlenmektedir.

model.similarity (‘zeka','bilim') ——->> *0.88671410881046953*
model.similarity(‘yaratıcılık','yapay') ——->> *0.90104469558231726*
model.similarity(‘bilgi','metafizik') ———>> *0.89918061140991656*

Şekil 13*:* Benzer kelimelerin yakınlığını göstermektedir.

Şekil 13‘te kelimelere atanan vektörler arasındaki kosinus mesafelerine (yakınlıklar/benzerlikler) bakacak olursak, model benzer kelimelerin yakınlığını yüksek, benzer olmayan kelimelerin de yakınlığını düşük olarak tanımlamaktadır.

```
model.most_similar('metafizik')
```

```
[('ontolojik', 0.9992651343345642),
 ('listede', 0.9992560148239136),
 ('motivasyonu', 0.999171257019043),
 ('bkz.', 0.9990996718406677),
 ('eğilimler', 0.9989954233169556),
 ('etkileyebileceği', 0.9989340901374817),
 ('Âlem', 0.9988895058631897),
 ('şimdiki', 0.9988783001899719),
 ('son', 0.9988157749176025),
 ('yer', 0.9987466931343079)]
```

Şekil 14 : “Metafizik” kelimesi ile benzer olan kelimeler ve benzerlik oranları.

Şekil 14’de bulunan sonuçlara bakıldığında verilen kelime ile “ontolojik, motivasyonu, son, yer ,eğilimler” kelimelerinin yüksek oranda benzer oldukları görülmektedir. Metin bağlamına bakıldığında “ontolojik”, “Alem”, ‘‘yer” kavramların arasında güçlü bir anlam ilişkisi olduğu tespit edilmektedir.

KAYNAKÇA

1. Cinsdiki, M. (1997). *Neural Network Solutions for ATM Routing & Multicasting Problems.* Yayınlanmış Yüksek Lisans Tezi, Ege Üniversitesi, Fen Bilimleri Enstitisü, İzmir.

2. https://medium.com/@bakiiii/deep-learning-ile-t%C3%BCrk%C3%A7e-film-yorumlar%C4%B1ndan-duygu-ve-puan-tahmini-78a606d86dde, *erişim tarihi: 08. 01.2018*

3. Der: Becerikli, Y. *Yapay Sinir Ağları ile Duygu Analizi.* Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği.

4. Ergezer, H.& Dikmen, M.& Özdemir, E. (2003). *Yapay Sinir Ağları Ve Tanıma Sistemleri Pivolka.* Uygulamalı Yerbilimleri Dergisi, 2(6), s.71-79, Kocaeli.

5. Ataseven, B. (2013) *Yapay Sinir Ağları ile Öngörü Modellemesi.* Dergipark, s.101-115, İstanbul.

6. Çayıroğlu, İ. *İleri Algoritma Analizi-5 Yapay Sinir Ağları.* Karabük Üniversitesi Mühendislik Fakültesi.

7.-http://kod5.org/yapay-sinir-aglari-ysa-nedir/, *erişim tarihi: 09.01.2018*

8. http://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari, *erişim tarihi : 09.01.2018*

9. Aşkın, D. & İskender, İ. & Mamızadeh, A. (2011) *Farklı Yapay Sinir Ağları Yöntemlerini Kullanarak Kuru Tip Transformatör Sargısının Termal Analizi.* Gazi Üniv. Müh. Mim. Fak. Der.Cilt 26, No 4, 905-913, Ankara.

10. https://www.tensorflow.org/tutorials/word2vec , *erişim tarihi: 10.01.2018*

11. https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/, *erişim tarihi: 10.01.2018*

12. Doğan, G. (2010). *Yapay Sinir Ağları Kullanılarak Türkiye'deki Özel Bir Sigorta Şirketinde Portföy Değerlendirmesi.* Yayınlanmış Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara.

13. Altun, İ.& Dündar, S. (2005). *Yapay Sinir Ağları ile Trafik Akım Kontrolü.* Deprem Sempozyumu, Kocaeli.

EKLER

EK -1: https://github.com/sdrttnclskn/YapaySinirAglari-GeriYayilimUygulamasi

EK -2 : https://github.com/sdrttnclskn/DataScience-TwitterSentimentAnalysis

EK -3 : https://github.com/sdrttnclskn/Word2Vec-TextAnalysis