

**BURSA TEKNİK ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**ENDÜSTRİ MÜHENDİSLİĞİ**

**MAKİNE ÖĞRENMESİNE GİRİŞ**

**PROJE KONUSU:**

**YAPAY SİNİR AĞLARI, PERCEPTRON, RNN ALGORİTMASI**

**HAZIRLAYAN:**

**EMİRHAN ERDİNÇ**

**19392380009**

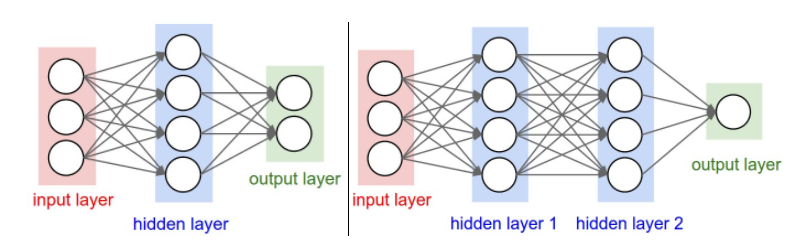
**DERSİN HOCASI: KADER NİKBAY OYLUM**

**İÇİNDEKİLER:**

1. **YAPAY SİNİR AĞI NEDİR?**
2. **Yapay Sinir Ağlarının Avantajları**
3. **Yapay Sinir Ağlarının Kullanım Alanları**
4. **Yapay Sinir Ağları Tarihi:**
5. **Yapay Sinir Ağ Modelleri**
6. **Perceptron**
7. **RNN NEDİR**
8. **RNN Tarihçesi**
9. **RNN Çeşitleri**
10. **RNN KULLANIM ALANLARI**
11. **KAYNAKÇA**

**YAPAY SİNİR AĞI NEDİR?**

Yapay sinir ağları (YSA), insan [beyninin](https://tr.wikipedia.org/wiki/Beyin) bilgi işleme tekniğinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. YSA ile basit [biyolojik](https://tr.wikipedia.org/wiki/Biyolojik) sinir sisteminin çalışma şekli taklit edilir. Yani biyolojik [nöron](https://tr.wikipedia.org/wiki/N%C3%B6ron) hücrelerinin ve bu hücrelerin birbirleri ile arasında kurduğu sinaptik bağın dijital olarak modellenmesidir. Nöronlar çeşitli şekillerde birbirlerine bağlanarak ağlar oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözlemleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.





**YAPAY SİNİR AĞLARI ELEMANLARININ İNSAN SİNİR SİSTEMİNDE KARŞILIKLARI**

**Yapay Sinir Ağlarının Avantajları**

* Yapay Sinir Ağları bir çok hücreden meydana gelir ve bu hücreler eş zamanlı çalışarak karmaşık işleri gerçekleştirir.
* Öğrenme kabiliyeti vardır ve farklı öğrenme algoritmalarıyla öğrenebilirler.
* Görülmemiş çıktılar için sonuç (bilgi) üretebilirler. Gözetimsiz öğrenim söz konusudur.
* Örüntü tanıma ve sınıflandırma yapabilirler. Eksik örüntüleri tamamlayabilirler.
* Hata toleransına sahiptirler. Eksik veya belirsiz bilgiyle çalışabilirler. Hatalı durumlarda dereceli bozulma (graceful degradation) gösterirler.
* Paralel çalışabilmekte ve gerçek zamanlı bilgiyi işleyebilmektedirler.

Yapay sinir ağları başlıca teşhis, sınıflandırma, tahmin, kontrol, veri ilişkilendirme, veri filtreleme, yorumlama gibi alanlarda kullanılmaktadır. Hangi problem için hangi ağın daha uygun olduğunu belirlemek için ağların özellikleri ile problemlerin özelliklerini karşılaştırmak gerekir.

**Yapay Sinir Ağlarının Kullanım Alanları**

* Hesaplamalı Finans(Computational finance): Kredi skorlaması(credit scoring), Algoritmik Ticaret(algorithmic trading)
* Görüntü işleme ve bilgisayarla görü (image processing and computer vision): Yüz tanıma(face recognition), hareket tanıma (motion detection), nesne tanıma (object detection)
* Hesaplamalı biyoloji (Computational biology): Tümör bulma (tumor detection), İlaç keşfi (durg discovery), DNA dizilimi (DNA seqencing)
* Enerji üretimi (Energy production): Fiyat ve yük tahmini (price and load forecasting)
* Otomotiv, havacılık ve üretim (Automotive, aerospace and manufacturing): Öngörücü bakım (predictive maintenance)
* Doğal dil işleme (Natural language processing): Sesli Asistan (voice assistant), Duygu analizi (emotion analysis)

Bir yapay sinir hücresi beş bölümden oluşmaktadır;

**1.Girdiler:** Girdiler nöronlara gelen verilerdir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir.

**2. Ağırlıklar:** Yapay sinir hücresine gelen bilgiler girdiler üzerinden çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilinmektedir.

**3.Toplama Fonksiyonu (Birleştirme Fonksiyonu):**Toplama fonksiyonu bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur

**4**.**Aktivasyon fonksiyonu:**Önceki katmandaki tüm girdilerin ağırlıklı toplamını alan ve daha sonra bir çıkış değeri (tipik olarak doğrusal olmayan) üreten ve bir sonraki katmana geçiren bir fonksiyondur. (örneğin, ReLU veya sigmoid ).

**5.Çıktılar**: Aktivasyon fonksiyonundan çıkan değer hücrenin çıktı değeri olmaktadır. Her hücrenin birden fazla girdisi olmasına rağmen bir tek çıktısı olmaktadır. Bu çıktı istenilen sayıda hücreye bağlanabilir.

**Yapay Sinir Ağları Tarihi:**

1949 yılında McGill Üniversitesi’nde bir psikolog olan Donald Hebb, “The Organization of Behaviour” isimli kitabında, yapay hücrelerden oluşan bir YSA’nın ağırlık değerlerini geliştiren bir öğrenme kuralını açıklamıştır (Hebb, 1949, 60-78). Hebb Öğrenme Kuralı denilen bu kural, günümüzde de birçok öğrenme kuralının temelini oluşturmaktadır (Öztemel, 2006, 37). Farely ve Clark tarafından 1954 yılında rassal ağlar ile adaptif tepki üretme kavramı ortaya atılmıştır (Anderson ve McNeill, 1992, 17). Bu kavram, 1958 yılında Rosenblatt ve 1961 yılında Caianiello tarafından geliştirilmiştir. Özellikle Rosenblatt tarafından geliştirilen algılayıcı (perceptron) model, daha sonraları geliştirilerek yapay sinir ağlarında devrim niteliğinde olan çok katmanlı algılayıcıların temelini oluşturmuştur

1950’li yıllarda bilgisayarların gelişmesiyle birlikte, insanın düşünce yapısıyla ilgili teorilerin temellerinin modellenmesi olanaklı hale gelmiştir. IBM araştırma laboratuarları araştırmacılarından Nathanial Rochester, bir sinir ağı simülasyonu oluşturma çabalarına öncülük etmiştir. İlk girişim başarısız olmasına rağmen sonraki girişimler başarılı olmuştur. Bu aşamadan sonra, geleneksel hesaplama yöntemlerinin araştırılması yerini sinirsel hesaplama yöntemlerinin araştırılmasına bırakmıştır (Anderson ve McNeill, 1992, 17). Rochester ve ekibi, o zamana kadar “düşünen makineleri” kendi çalışmalarını kanıt göstererek savunmaktaydılar. 1956 yılında yapılan ve daha sonraları Yapay Zeka (YZ) ve YSA (Yapay Sinir Ağları) üzerine konuşmaların ve dolayısıyla kanıtların artmasını sağlamış olan “Yapay Zeka Dartmouth Yaz Araştırma Projesi” yapılmıştır. Bu araştırmanın sonuçlarından biri, hem YZ ve hem de YSA araştırmalarına ilgi gösterilmesini teşvik etmesidir (Anderson ve McNeill, 1992, 17).

**Yapay Sinir Ağ Modelleri**

Tek katmanlı algılayıcılar, çok katmanlı algılayıcılar, ileri beslemeli yapay sinir ağları ve geri beslemeli yapay sinir ağlar; yapay sinir ağ modelleridir.

**Çok Katlı Algılayıcılar (ÇKA)**  
Algılayıcı ve Adaline yöntemleri doğrusal olmayan çözümler üretemediği için hem mimari hem de eğitim algoritması açısından iyileştirilmiş Çok Katmanlı Algılayıcı (MLP) ağı önerilmiştir. Mimari açıdan doğrusal olmayan aktivasyon fonksiyonuna sahip birçok nöronun birbirine hiyerarşik olarak bağlandığı bir yapıya sahip olan MLP, Algılayıcı ve Adaline yöntemlerinin avantajları yanı sıra geri-yayılım adındaki öğrenme sistemini kullanmaktadır ve genel olarak yapay sinir ağları ileri beslemeli ve geri beslemeli ağlar olarak ikiye ayrılmaktadır.

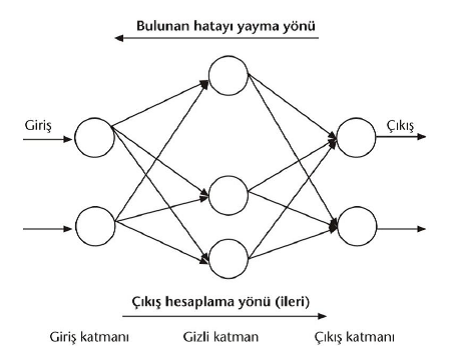
**Tek Katmanlı Algılayıcılar:**

Tek katmanlı ağlar sadece girdi ve çıktıdan oluşur. Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusaldır ve 1 veya -1 değerlerini almaktadır. Eğer çıktı 1 ise birinci sınıfa, -1 ise ikinci sınıfa kabul edilmektedir.

**İleri Beslemeli Ağlar**  
İleri beslemeli sinir ağları tek yönlü sinyal akışı için izin verir. Ayrıca, ileri beslemeli sinir ağları çoğu katmanlar halinde organize edilmektedir. Üç katmanlı ileri beslemeli sinir ağı bir örneği Şekil 7’de gösterilmiştir. Bu şebeke giriş düğümleri, iki gizli katman ve bir çıkış katmandan oluşur. Tipik aktivasyon fonksiyonları Şekil 4’te gösterilmiştir. Bu sürekli aktivasyon fonksiyonları katmanlı ağlar gradyan tabanlı eğitimi için izin verir (Wilamowski, 2003).

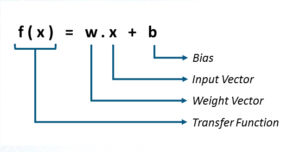
İleri beslemeli yapay sinir ağında, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan ara (gizli) katmandaki hücrelere iletir. Bilgi, ara ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar, doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı yapay sinir ağının, orta katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir. En çok bilinen algoritma olan geriye yayılım öğrenme algoritması, bu tip yapay sinir ağların eğitiminde etkin olarak kullanılmaktadır. Ağa, hem örnekler hem de örneklerden elde edilmesi gereken çıktılar (beklenen çıktılar) verilmektedir. Ağ kendisine gösterilen örneklerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra gösterilen benzer örnekler için bu çözüm uzayı sonuçlar ve çözümler üretebilmektedir.

**Geri Beslemeli Yapay Sinir Ağları**  
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 yapısı ile geri beslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli YSA yapıları elde edilebilir. Aşağıda bulunan Şekil 8’de iki katmanlı ve çıkışlarından giriş katmanına geri beslemeli bir YSA yapısı görülmektedir (Kabalcı, 2015).



**Perceptron**

Makine öğreniminde, algılayıcı, ikili sınıflandırıcıların denetimli öğrenimi için bir algoritmadır. İkili bir sınıflandırıcı, bir sayılar vektörüyle temsil edilen bir girdinin belirli bir sınıfa ait olup olmadığına karar verebilen bir fonksiyondur.



Yukarıdaki şekilde tanımlanan ağın yapay sinir ağındaki karşılığı perceptron’dur. Bu fonksiyonda yukarıda gösterildiği üzere W değeri ağırlık parametresi, x değeri girdi, b değeri bias ve y değeride ağın çıktısı olarak tanımlanmaktadır. Burada x girdi değerimiz, örneğin kedi resimlerini tanıyorsak kedi resmine ait matrisi, y ise bu resmin kediye ne kadar benzediğine dair skoru verir. Parametrelerimiz olan W ağırlık ve b bias değerlerini bu çıktı skorunu iyileştirmek için kullanılır. Bu anlamda çok katmanlı yapay sinir ağlarında yada derin öğrenme de yaptığımız temel şey modelimiz için en iyi skoru verecek w ve b parametre değerlerini hesaplamaktır.

**RNN NEDİR**

Yinelemeli sinir ağı, düğümler arası bağların zamansal bir dizi doğrultusunda [yönlü çizge](https://tr.wikipedia.org/wiki/Y%C3%B6nl%C3%BC_%C3%A7izge) oluşturduğu bir [yapay sinir ağı](https://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1) çeşididir. Yaygın olarak İngilizce kısaltması olan RNN ([İngilizce](https://tr.wikipedia.org/wiki/%C4%B0ngilizce): Recurrent neural network) olarak anılır. [İleri beslemeli sinir ağından](https://tr.wikipedia.org/w/index.php?title=%C4%B0leri_beslemeli_sinir_a%C4%9F%C4%B1&action=edit&redlink=1) türetilen RNN yöntemi, bir iç durum belleği kullanarak değişik uzunluktaki dizileri işleyebilir. Bu sayede [yazı tanıma](https://tr.wikipedia.org/w/index.php?title=Yaz%C4%B1_tan%C4%B1ma&action=edit&redlink=1) ve [konuşma tanıma](https://tr.wikipedia.org/wiki/Konu%C5%9Fma_tan%C4%B1ma) gibi problemlere uygulanabilir. Teorik olarak [Turing makinesine](https://tr.wikipedia.org/wiki/Turing_makinesi) denk ([Turing-complete](https://tr.wikipedia.org/w/index.php?title=Turing-complete&action=edit&redlink=1)) olan yinelemeli sinir ağları, herhangi uzunluktaki bir [girdiyi](https://tr.wikipedia.org/wiki/Girdi) işleyebilen herhangi bir programı çalıştırabilir.

**RNN Tarihçesi**

Yinelemeli sinir ağları [David Rumelhart](https://tr.wikipedia.org/w/index.php?title=David_Rumelhart&action=edit&redlink=1)'ın 1986 yılındaki çalışmasına dayanır.[[8]](https://tr.wikipedia.org/wiki/Yinelemeli_sinir_a%C4%9F%C4%B1#cite_note-8) Hopfield ağı denen özel bir RNN türü de [John Hopfield](https://tr.wikipedia.org/w/index.php?title=John_Hopfield&action=edit&redlink=1) tarafından 1982 yılında geliştirilmiştir. 1993 yılında, bir RNN çalışması 1000'den fazla katman gerektiren bir “çok derin öğrenme” görevini başarmıştır. [Long short-term memory](https://tr.wikipedia.org/wiki/Long_short-term_memory" \o "Long short-term memory) (LSTM) ağları [Hochreiter](https://tr.wikipedia.org/w/index.php?title=Sepp_Hochreiter&action=edit&redlink=1" \o "Sepp Hochreiter (sayfa mevcut değil)) ve [Schmidhuber](https://tr.wikipedia.org/w/index.php?title=J%C3%BCrgen_Schmidhuber&action=edit&redlink=1" \o "Jürgen Schmidhuber (sayfa mevcut değil)) tarafından 1997 yılında geliştirilmiş ve çeşitli uygulama alanlarında en iyi performansları kaydetmiştir.

**RNN Çeşitleri**

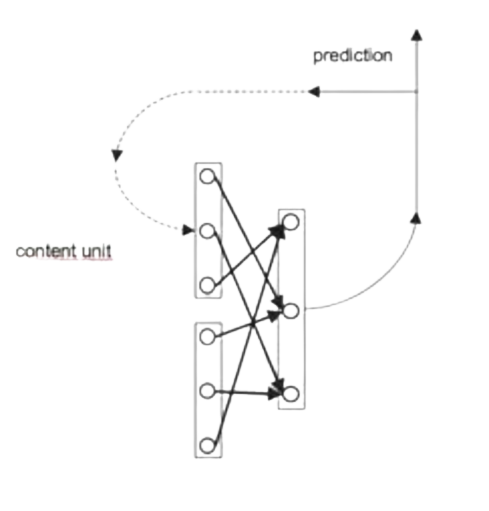
**Tam Yinelemeli**

Tam yinelemeli sinir ağlarında tüm nöronların çıktısı tüm nöronların girdisine bağlanır. En genel RNN mimarisi budur, çünkü diğer tüm mimariler, buradaki bazı bağların ağırlıkları sıfırlanarak elde edilebilir. RNN'ler iki farklı biçimde gösterilir: kapalı biçimde, [özyineleme](https://tr.wikipedia.org/wiki/%C3%96zyineleme) bağlantıları [düğümlerin](https://tr.wikipedia.org/wiki/D%C3%BC%C4%9F%C3%BCm_(matematik)) kendilerinin bir sonraki adımdaki durumuna olan bağlantılarıdır; açılmış biçimde, düğümlerin her zaman adımındaki durumları ayrı ayrı gösterilir.

**Geçitli Yinelemeli**

Geçitli yineleme birimi ([İngilizce](https://tr.wikipedia.org/wiki/%C4%B0ngilizce): gated recurrent unit, GRU) 2014 yılında önerilmiş bir yinelemeli ağ birimidir. Bu birimler, nöronlar arasındaki geçişi düzenleyen bir takım öğeler barındırır. [LSTM](https://tr.wikipedia.org/wiki/LSTM" \o "LSTM)'e benzer şekilde unutma kapısı bulunur, ancak GRU yapıları genellikle daha basittir. [Polifonik](https://tr.wikipedia.org/wiki/Polifonik) müzik ve konuşma sinyali modelleme gibi işlerde LSTM'e benzer bir başarıyla çalışır.

**RNN’İN ÇALIŞMA MANTIĞI**

Diyagramda görüldüğü gibi, RNN’in işlem döngüsünde gizli katmandan çıkan sonuç hem çıktı üretir, hem de içerik birimlerine yazılır. Bu şekilde, her yeni girdi, önceki girdilerin işlenmesi sonucu üretilmiş birlikte işlenir. Farklı zamanlarda belleğe alınan veriler arasında korelasyon bulunuyorsa buna “uzun vade” bağımlılık denir. RNN, bu uzun vade bağımlılıkların arasındaki ilişkiyi hesaplayabilen bir ağdır. 

**RNN’İN AVANTAJLARI**

* Herhangi bir uzunluktaki girdilerin işlenmesi imkanı
* Girdi büyüklüğüyle artmayan model boyutu
* Geçmiş bilgileri dikkate alarak hesaplama
* Zaman içinde paylaşılan ağırlıklar

**RNN’İN DEZAVANTAJLARI**

* Yavaş Hesaplama
* Uzun zaman önceki bilgiye erişme zorluğu
* Mevcut durum için gelecekteki herhangi bir girdinin düşünülememesi

**RNN KULLANIM ALANLARI**

**Dil Modelleme ve Metin Oluşturma**

Bir kelime dizisi verildiğinde, önceki kelimelerde verilen her bir kelimenin olasılığını tahmin etmek istiyoruz. Dil Modelleri, bir Cümlenin Makine Çevirisi için önemli bir girdi olan(yüksek olasılıklı cümleler genellikle doğru olduğundan) ölçmeyi mümkün kılar. Bir sonraki kelimeyi tahmin etmenin bir yan etkisi, çıktı olasılıklarından örnekleme yaparak yeni bir metin üretmemizi sağlayan bir üretken model elde etmemizdir. Ve eğitim verilerimizin ne olduğuna bağlı olarak her türlü şeyi üretebiliriz. Dil Modellemede, girdimiz tipik olarak bir kelime dizisidir (örneğin, tek-sıcak vektörler olarak kodlanır) ve çıktımız tahmin edilen kelime dizisidir.

**Makine Çevirisi**

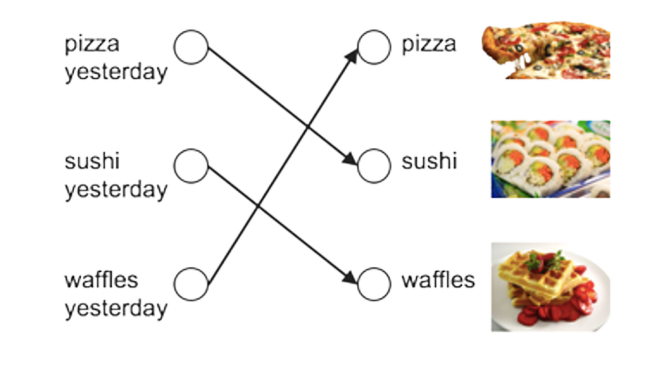
Makine Çevirisi, kaynak dilimizde (örneğin Almanca) bir dizi kelime girdiğimiz için dil modellemesine benzer. Hedef dilimizde (veya İngilizce) bir kelime dizisi çıkarmak istiyoruz. Temel bir fark, sadece tüm giriş dizisini yakalamamız gerektiğidir, çünkü çevrilmiş cümlenin ilk sözcüğü, tam giriş dizisinden yakalanabilir.

**Konuşma Tanıma**

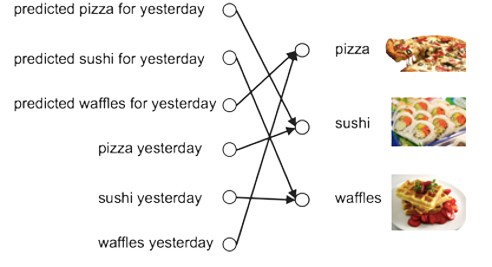
Ses dalgasından gelen akustik sinyallerin bir giriş dizisi verildiğinde, fonetik bölümlerin bir dizisini olasılıkları ile birlikte tahmin edebiliriz.

RNN ayrıca Geleneksel sinir ağı, Müzik üretimi, Duygu sınıflandırma, İsim varlık tanıma gibi alanlarda da kullanılır.

**RNN ÖRNEĞİ**



Bir aşçının 3 farklı yemeği sırasıyla yaptığını düşünelim. 1. gün pizza yaptıysa 2. gün sushi 3. gün waffle yapıyor olsun. Bir sonraki gün ne yapacağını tahmin etmemiz gerekiyorsa öncelikle nasıl bir problemle karşı karşıya olduğumuz anlayarak, bu probleme uygun bir metot kullanmamız gerekir. Burada yemek bir sıraya göre geldiği için, yani önceki gün yapılan yemeğin sonraki günde etkisi olacağı için uygulanması gereken metot recurrent neural network metodudur. Bu sayede elimizdeki bilgilerle yeni bilgi olmasa dahi birkaç hafta sonraki günde yapılacak yemeği dahi tahmin edebiliriz.



KAYNAKÇA:

* <https://tr.wikipedia.org/wiki/Yinelemeli_sinir_a%C4%9F%C4%B1>
* <https://medium.com/deep-learning-turkiye/rnn-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-9e5d572689e1>
* <https://www.elektrikport.com/makale-detay/yinelemeli-sinir-aglari-(rnn)-nedir/23277#ad-image-0>
* <https://medium.com/@buse.kara7998/perceptron-nedi%CC%87r-a15a7c8e5276>
* <https://matkafasi.com/111517/perceptron-nedir>
* <https://tr.wikipedia.org/wiki/Perceptron>
* <https://www.datasciencearth.com/yapay-sinir-aglarinin-temel-yapisi/>
* <https://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1>
* <https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/>
* <https://devhunteryz.wordpress.com/2018/06/30/derin-ogrenme-perceptron-ogrenme-algoritmasi/>
* <https://atcold.github.io/pytorch-Deep-Learning/tr/week06/06-3/>