



*T.C*

***SAKARYA ÜNİVERSİTESİ***  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

# **BSM 310 – YAPAY ZEKA**

**Grup üyeleri:**  
**B171210016 Kaan Gecü**  
**B171210076 Burak Bayram**  
**1612.15156 Murad Abaszada**

**Sakarya**  
**2020**

# Yinelenen Sinir Ağları (Recurrent Neural Networks)

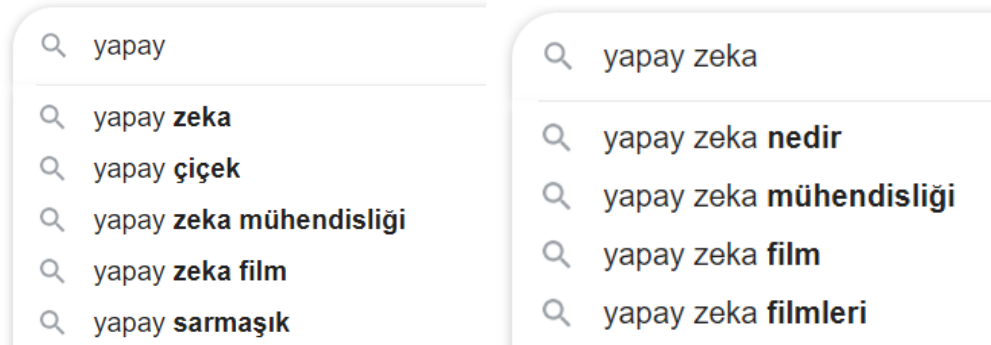
## Kısaca Sinir Ağları

Yapay Sinir Ağları insan beynindeki nöronların sinapslar ile birbiri arasında kurduğu iletişimi bilgisayar ortamında oluşturan bir hesaplama tekniğidir.

Bu öğrenme olayı verilen bir datadan çıkarım yaparak tahminler yürütülmesi ardından bu tahminlere dayanarak yeni tahminler yürütülmesi şeklinde gerçekleşir. Örnek vermek gerekirse birkaç tane yangın musluğu içeren ve içermeyen fotoğraflar verilerek sistemin bunları analiz etmesi sonucunda, verilen başka fotoğraflardaki yangın musluklarını da kırmızı, yol kenarında gibi unsurlara bakarak bulması sağlanır. Günümüzde Sinir Ağlarının kullanıldığı alanlar olan sürücüsüz arabalar, yüz tanıma algoritmaları bu şekilde ancak çok daha fazla kuralla işler.

## Yinelenen Sinir Ağları,

Yapay Sinir Ağlarının özelleşmiş bir alt dalıdır. Yinelenen Sinir Ağlarının farkı çıkışın tekrar girişe beslenmesi ile dinamik (zamana bağlı) bir yapıda tahminlerini yürütebilmesidir. Bu modelin kullanım alanı birbiri ile bağlantılı olan yazı tanıma ve söz tanıma gibi yerlerdir. Hepimizin kullandığı arama motorları bu mantıktan yararlanarak milyonlarca aramadan aldığı bilgiye göre bize arama önerisinde bulunmaktadır.

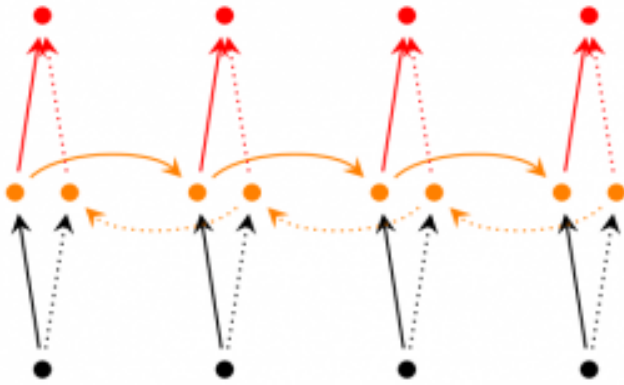


*Google üzerinde bu duruma bir örnek*

# Başlıca Birkaç Yinelenen Sinir Ağı Mimarisi

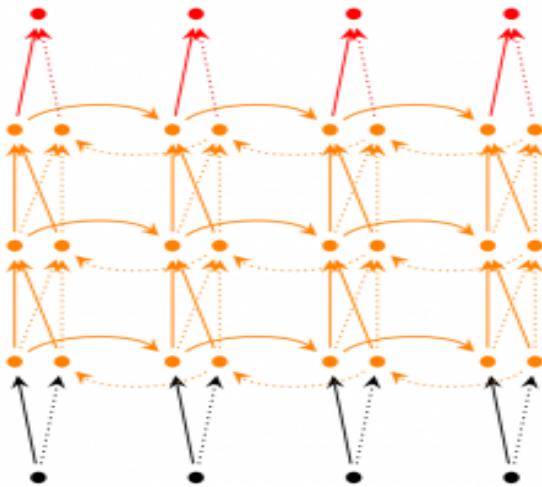
## İki Yönlü RNN'ler(Bidirectional RNNs)

T anındaki çıktıların sadece önceki düğümlerdeki çıktılarına değil gelecekteki düğümlerdeki çıktılarına da bağlı olabileceği fikrine dayanmaktadır. Örneğin, bir dizideki eksik bir kelimeyi tahmin etmek için hem sol hem de sağ içeriğe bakmak istiyorsunuz. İki yönlü RNN'ler oldukça basittir. Bunlar sadece üst üste istiflenmiş iki RNN'dir. Çıktılar daha sonra her iki RNN'nin gizli halini temel alarak hesaplanır.



## Derin(İki Yönlü) RNN'ler(Deep RNNs)

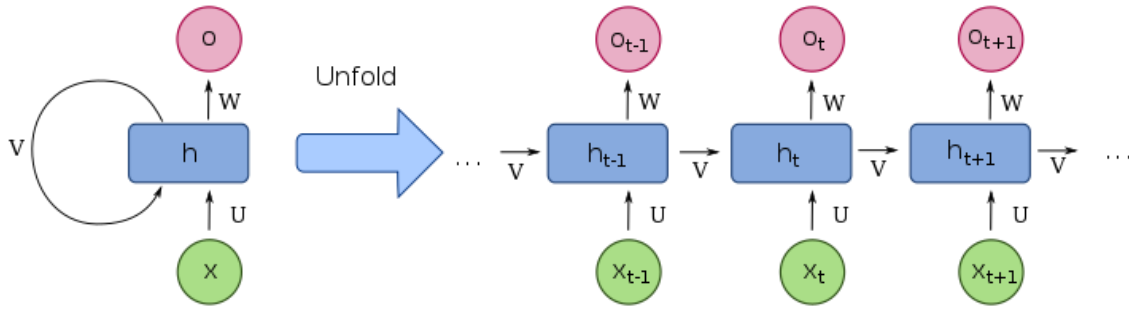
Çift Yönlü RNN'lere benzer, yalnızca zaman adımıyla birden fazla katmandan oluşur. Uygulamada bu bize daha yüksek bir öğrenme kapasitesi sağlar (ancak aynı zamanda bir çok eğitim verisine ihtiyaç duyulur).



## Tamamen Yinelenen

Daha önce de anlattığımız gibi bu ağlar nöron benzeri düğüm yapılarından oluşmaktadır. Bu düğümler farklı katmanlarda toplanmıştır, tamamen yinelenen mimaride her bir düğüm bir sonraki katmana bilgi besler. Bu düğümler üç farklı şekilde bulunurlar,

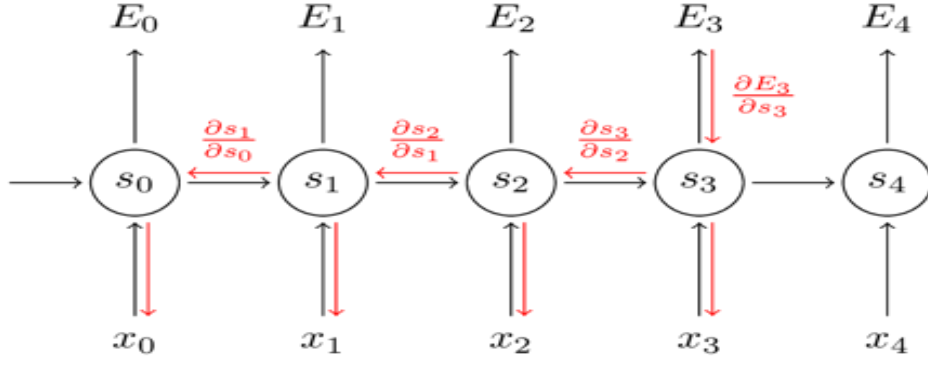
1. Giriş – Ağ dışından bilgi girişi yapılır
2. Gizli – giriş çıkış arasında bilgiyi düzenler ya da değiştirir
3. Çıkış – Sonuçların bulunduğu düğümlerdir



## Backpropagation Through Time (BPTT)

RNN'lerin amacı sıralı inputları doğru bir şekilde sınıflandırmak diyebiliriz. Bu işlemleri yapabilmek için hatanın backpropunu ve gradient descentini kullanırız. Backprop, ileri beslemeli ağlarda sonda outputtaki hatayı geriye hatanın türevini ağırlıklara dağıtılarak yapılır. Bu türev kullanılarak öğrenme katsayısı, gradient descent düzenlenerek hatayı düşürecek şekilde ağırlıklar düzenlenir.

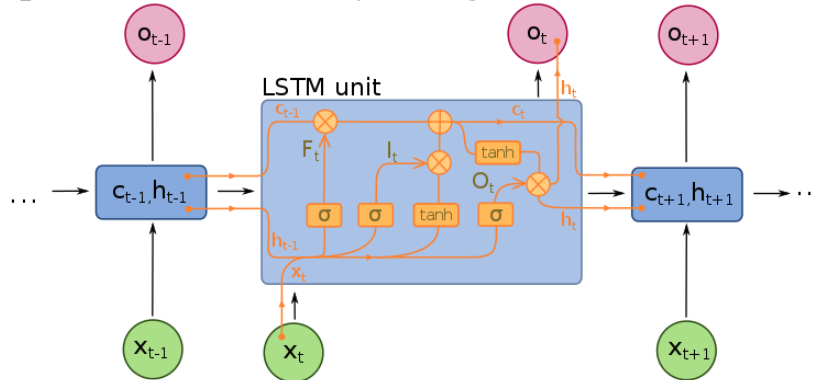
RNN için kullanılan yöntem ise BPTT diye bilinen zamana bağlı sıralı bir dizi hesaplamasının tümü için backprop uygulamasıdır. Yapay ağlar bir dizi fonksiyonu iç içe  $f(h(g(x)))$  şeklinde kullanır. Buraya zamana bağlı değişken eklendiğinde türev işlemi zincir kuralı ile çözümlenebilir.



Yukarıdaki figürde sıralı 5 girdili bir RNN yapısı gösterilmektedir. E burada oluşan hatayı ifade etmektedir. Örneğin,  $E_3$  için geri yayılım(backpropagation) yaparken yaptığımız işlemde  $w$  ağırlığına göre türevi kullanılmaktadır. Bu türevi çözebilmek için zincir kuralı ile birkaç türevin çarpımını kullanırız.

### Uzun Kısa Süreli Bellek (LSTM)

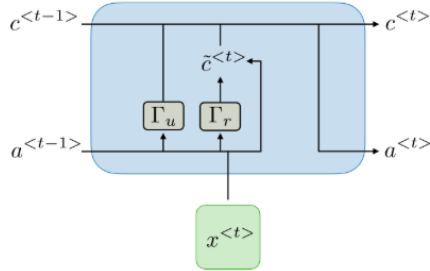
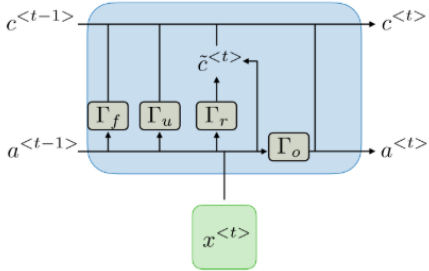
Bu sistem kaybolan gradyan probleminin önüne geçilmesi amacı ile tasarlanmıştır. Bu problem katman sayısına göre katlanarak azalan- artan olabilen çarpımsal gradyan nedeniyle uzun vadeli bağımlılıkları yakalamadaki zorluktan kaynaklanır. Bunu Unutma Kapısı adlı yapılarla gerçekleştirir. Geri yayılım sırasında oluşan bir hatanın yok olmasını engeller. Yok olmak yerine sistemin önceki zamanında sonsuza kadar giderek binlerce birim zaman önce öğrenilen olayları karşılaşılan problemde kullanabilir. Bu özellikle dilde mecaz anlamda kullanılması gibi duruma bağlı olarak farklı anlam kazanan kelimeleri ve cümle yapılarını tanımada fayda sağlar.



## Geçitli Tekrarlayan Birim (GRU)

Uzun Kısa Süreli Bellek sistemine benzer yapıdadır. Asıl farkları LSTM ye göre daha az parametre bulundurmasıdır. Çıkış kapıları yoktur. Kullanıldığı başlıca alanlardan biri polifonik müziklerin modellenmesindedir.

### LSTM ile GRU arasındaki farklar

Karakterizasyon	Geçitli Tekrarlayan Birim (GRU)	Uzun Kısa Süreli Bellek (LSTM)
$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + (1 - \Gamma_u) \star c^{<t-1>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$
$a^{<t>}$	$c^{<t>}$	$\Gamma_o \star c^{<t>}$
Bağımlılıklar		

## KAYNAKLAR

- <https://www.youtube.com/watch?v=UNmqTiOnRfg>
- <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- [https://en.wikipedia.org/wiki/Recurrent\\_neural\\_network](https://en.wikipedia.org/wiki/Recurrent_neural_network)
- <https://medium.com/@hamzaerguder/recurrent-neural-network-nedir-bdd3d0839120>