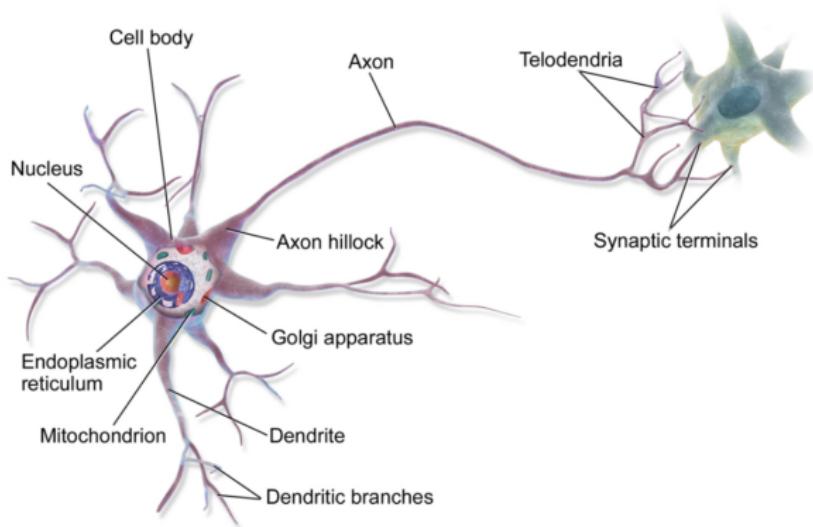


Derin Öğrenme

Dr.Furkan Göz

Aralık 11, 2023

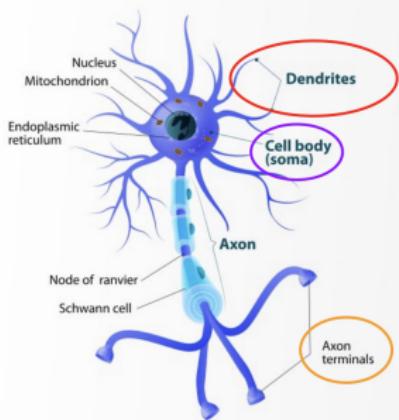


By BruceBlaus - Own work, CC BY 3.0,
<https://commons.wikimedia.org/w/index.php?curid=28761830>

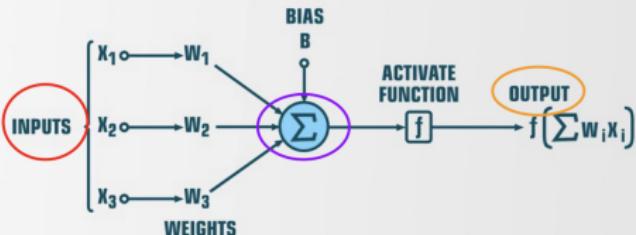
Neuron vs Artificial Neuron

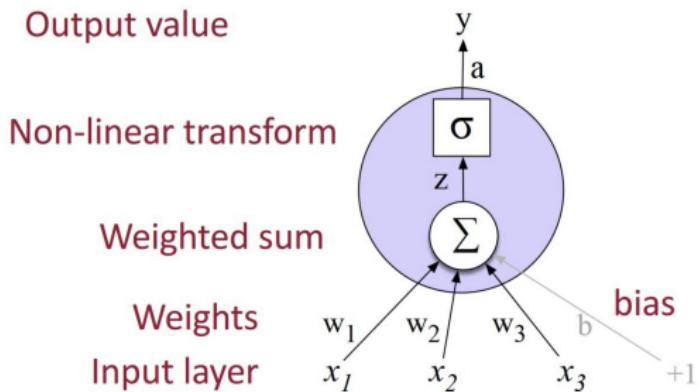
Doğal Dil İşleme ve Metin Madenciliğine Giriş

Structure of Typical Neuron



Structure of Artificial Neuron





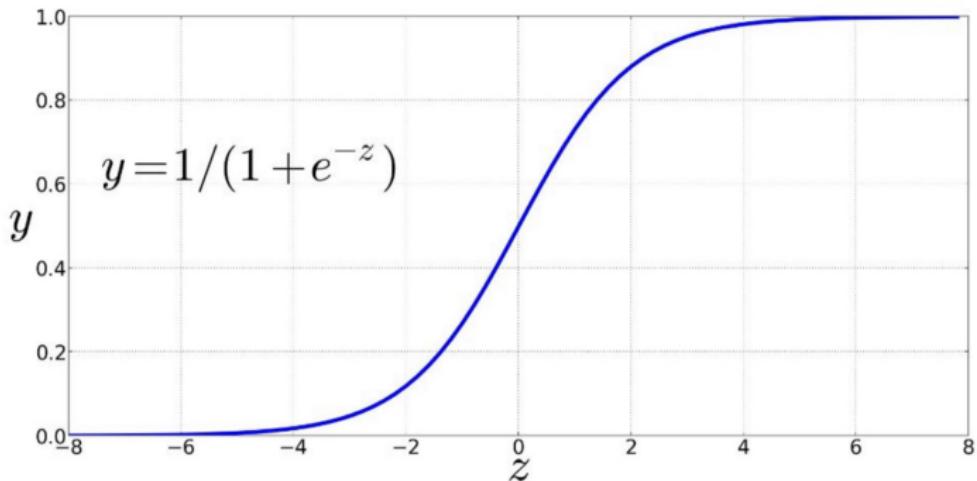
Neural unit girdilerin ağırlıkları toplamı alır ve bias terimi eklenir

$$z = b + \sum_i w_i x_i$$

$$z = \mathbf{w} \cdot \mathbf{x} + b$$

$$y = a = f(z)$$

doğrusal olmayan fonksiyonlar $f()$: sigmoid, tanh, ve ReLU



$$y = \sigma(w \cdot x + b) = \frac{1}{1 + \exp(-(w \cdot x + b))}$$

Tanh ve Relu

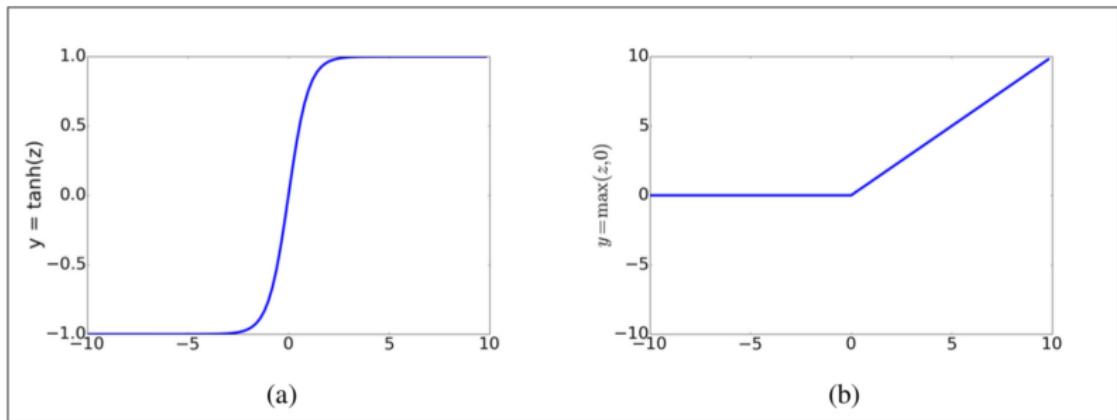


Figure 7.3 The tanh and ReLU activation functions.

Örnek



Bir neural için değerler:

$$w = [0.2, 0.3, 0.9] \quad \text{ve} \quad b = 0.5$$

Aşağıdaki girdi x için sonuç ne olur?

$$x = [0.5, 0.6, 0.1]$$

$$\begin{aligned}y &= \sigma(w \cdot x + b) = \frac{1}{1 + e^{-(w \cdot x + b)}} \\&= \frac{1}{1 + e^{-(0.5 \times 0.2 + 0.6 \times 0.3 + 0.1 \times 0.9 + 0.5)}} \\&= \frac{1}{1 + e^{-0.87}} \\&= 0.70\end{aligned}$$

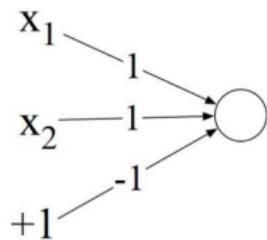
Can neural units compute simple functions of input? (Minsky and Papert (1969))

		AND		OR		XOR		
x1	x2	y	x1	x2	y	x1	x2	y
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0

Minsky, M. and S. Papert. 1969. Perceptrons. MIT Press

A very simple neural unit

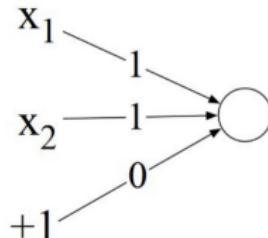
- Binary output (0 or 1)
- No non-linear activation function



AND

		AND
x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

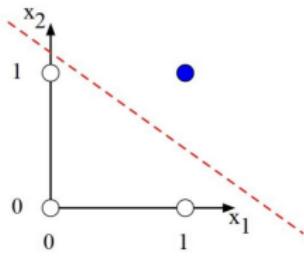
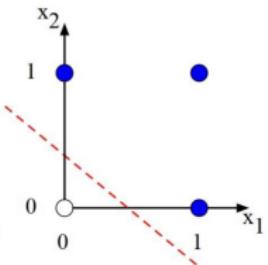
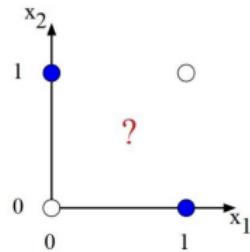
$$y = \begin{cases} 0, & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ 1, & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$



OR

		OR
x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	1

■ Decision boundary

a) x_1 AND x_2 b) x_1 OR x_2 c) x_1 XOR x_2

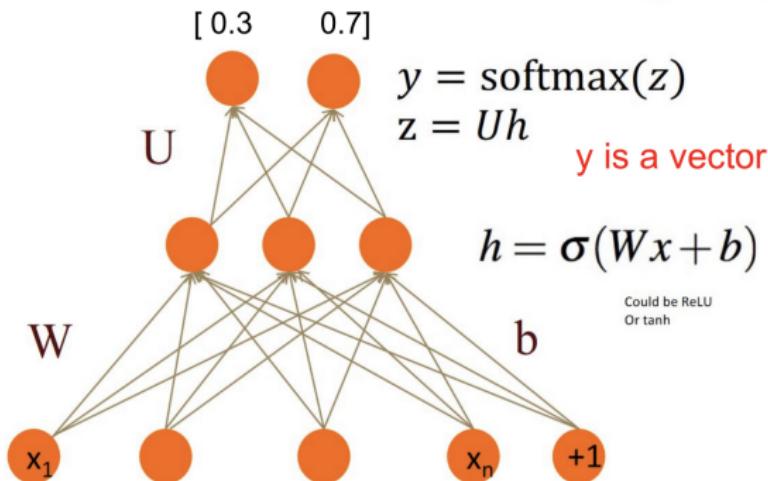
Multilayer Perceptron

$$\text{softmax}(\mathbf{z}_i) = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^d \exp(\mathbf{z}_j)} \quad 1 \leq i \leq d$$

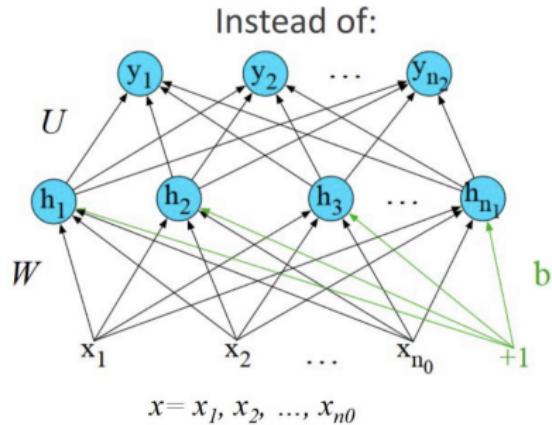
Output layer
(σ node)

hidden units
(σ node)

Input layer
(vector)

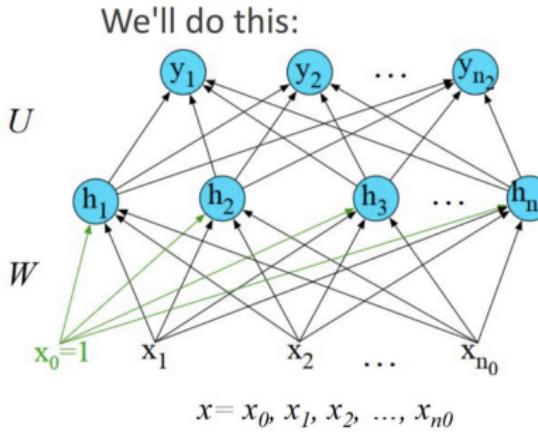


Multilayer Perceptron



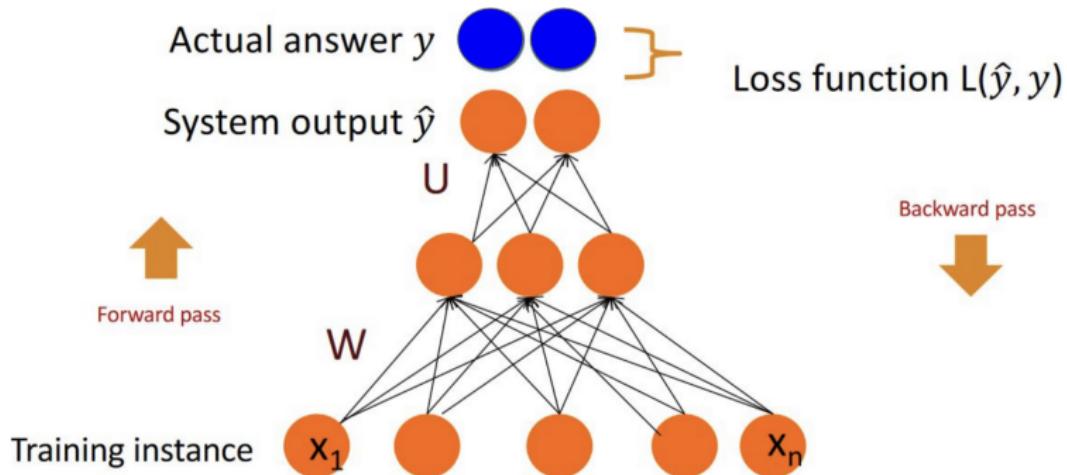
$$h = \sigma(Wx + b)$$

$$h_j = \sigma \left(\sum_{i=1}^{n_0} W_{ji} x_i + b_j \right)$$



$$h = \sigma(Wx)$$

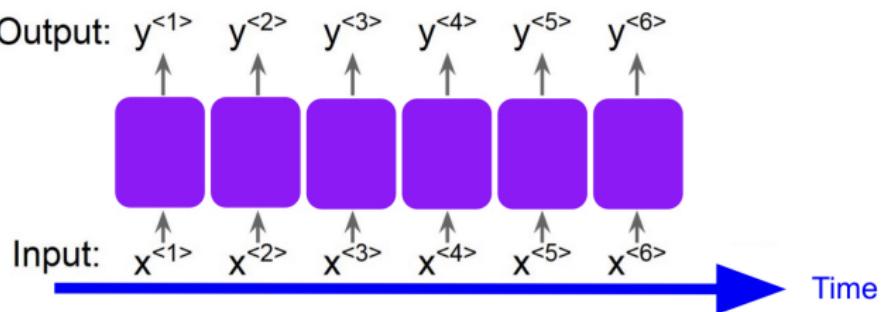
$$\sigma \left(\sum_{i=0}^{n_0} W_{ji} x_i \right)$$



Sequential Data

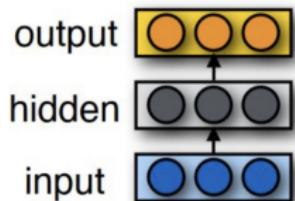
■ Zaman serisi, metin verileri

- The movie he has **not** seen is good
- The movie he has seen is **not** good

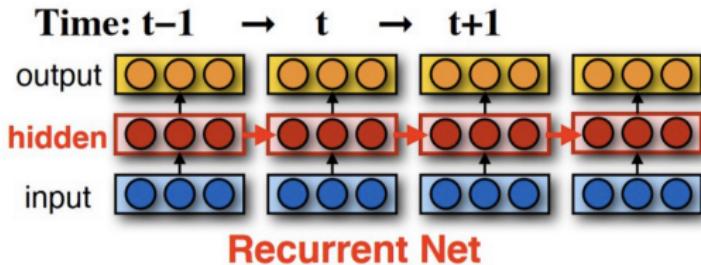


- İleri beslemeli ağlar sadece sabit boyutta girdi ve çıktıları işleyebilir.
- Tekrarlayan Sinir Ağları (RNN'ler) değişken uzunluktaki dizileri (girdi ve çıktı olarak) işleyebilir.
- RNN'lerin 3 ana çeşidi vardır ve bunlar iç yapılarına göre farklılık gösterir:
 - RNN
 - LSTM
 - GRU

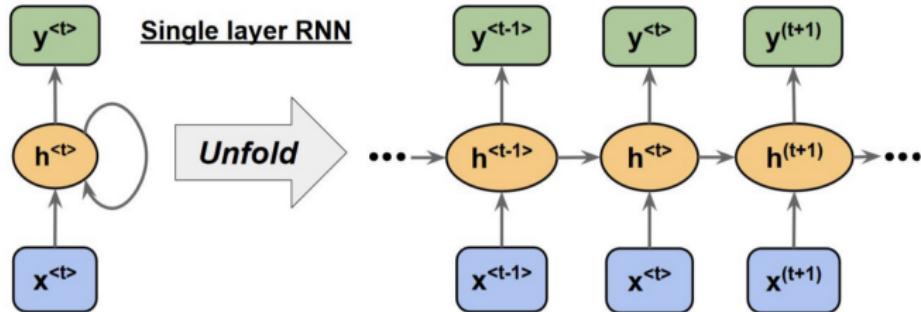
- RNN: İleri beslemeli bir ağın T kez çalıştırılarak bir dizi T çıktısı oluşturulduğu düşünülebilir.
- Yineleme: Bir önceki adımda hesaplanan gizli durum ($h(t-1)$) mevcut adımdaki gizli duruma ($h(t)$) aktarılır



Feedforward Net



- Girdi katmanı: verilerin alındığı katman.
- Gizli katman(lar): bilgiler saklanır.
- Çıktı katmanı (t zamanında): bu katman gizli katmandan işlenmiş verileri alır ve nihai tahminleri yapar.



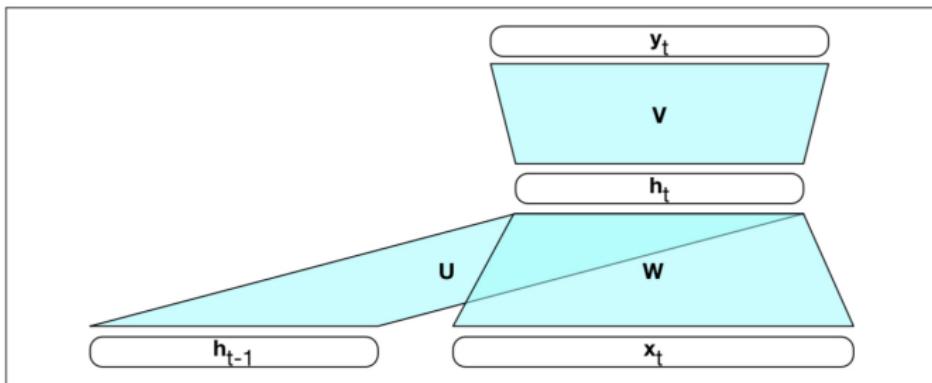
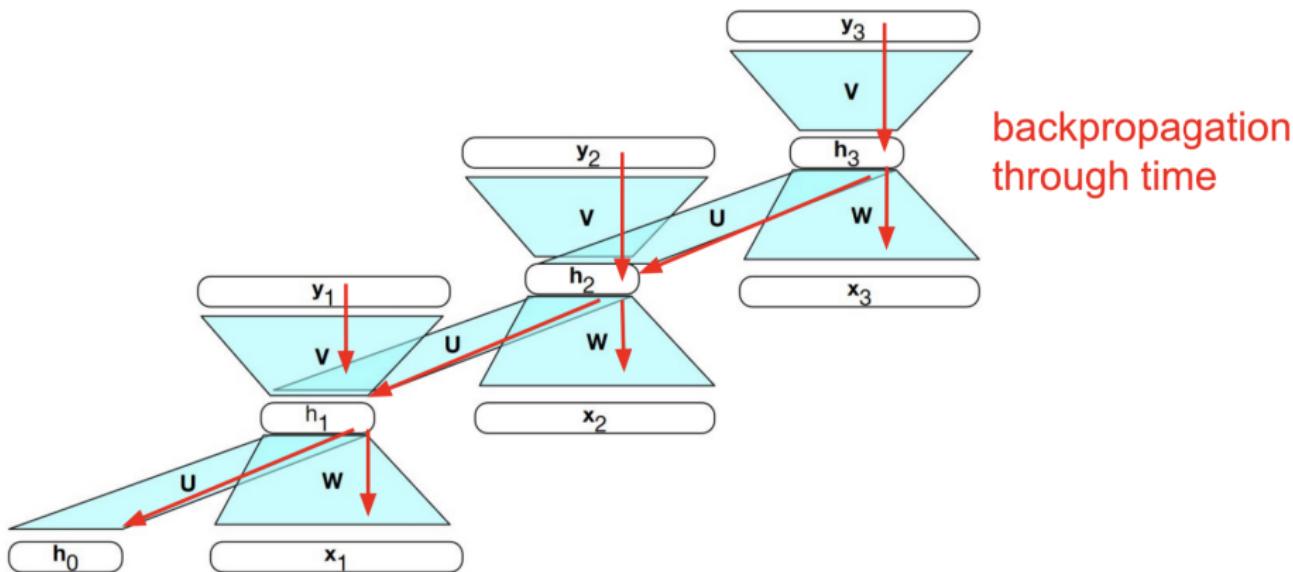


Figure 9.2 Simple recurrent neural network illustrated as a feedforward network.

- $h_t = g(Uh_{t-1} + Wx_t)$
- $y_t = f(Vh_t)$
- $y_t = \text{softmax}(Vh_t)$

function FORWARDRNN(**x**,*network*) **returns** output sequence **y**

```
h0 ← 0
for i ← 1 to LENGTH(x) do
    hi ← g(Uhi-1 + Wxi)
    yi ← f(Vhi)
return y
```



backpropagation
through time

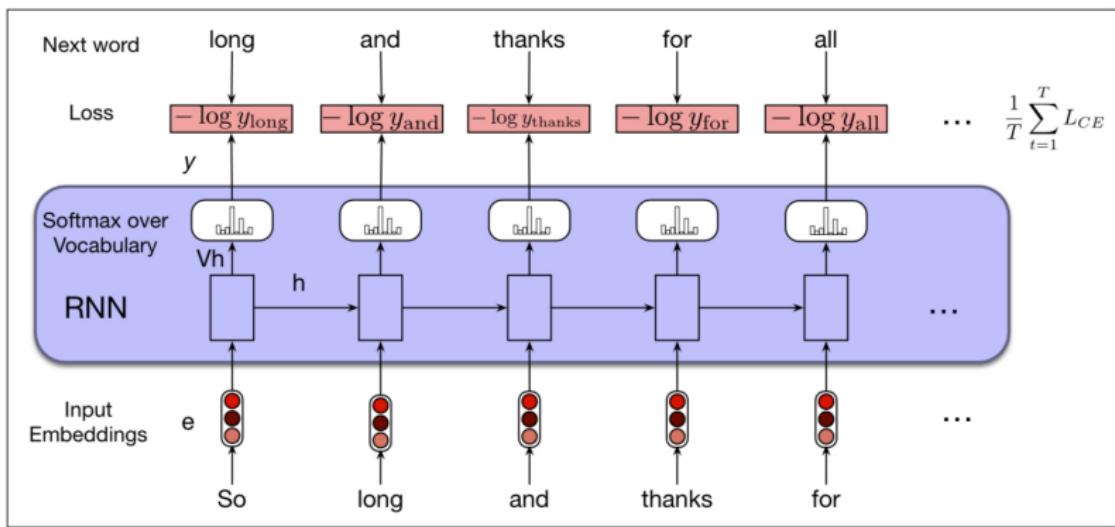


Figure 9.6 Training RNNs as language models.

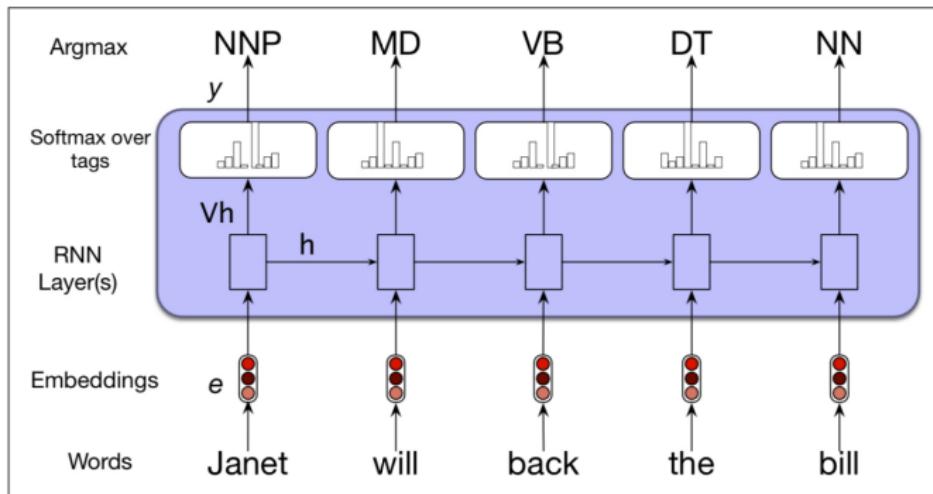


Figure 9.7 Part-of-speech tagging as sequence labeling with a simple RNN. Pre-trained word embeddings serve as inputs and a softmax layer provides a probability distribution over the part-of-speech tags as output at each time step.

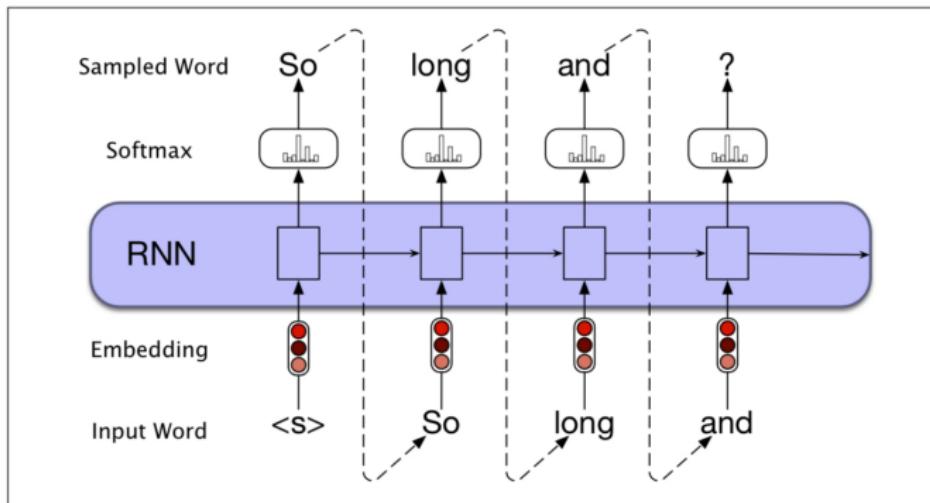
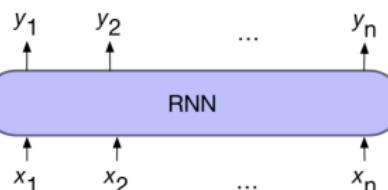
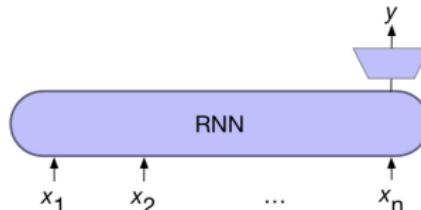


Figure 9.9 Autoregressive generation with an RNN-based neural language model.

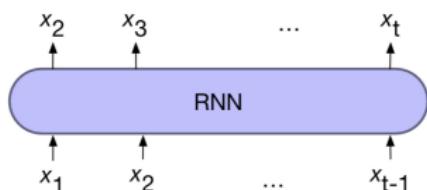
- Language modeling and generation
 - Otomatik tamamlama
 - Makine çevirisi
- Sınıflandırma (örn. duygusal analizi)
- Etiketleme (örn. kelime türü etiketleme)



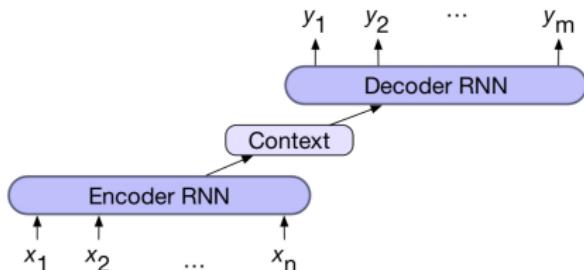
a) sequence labeling



b) sequence classification



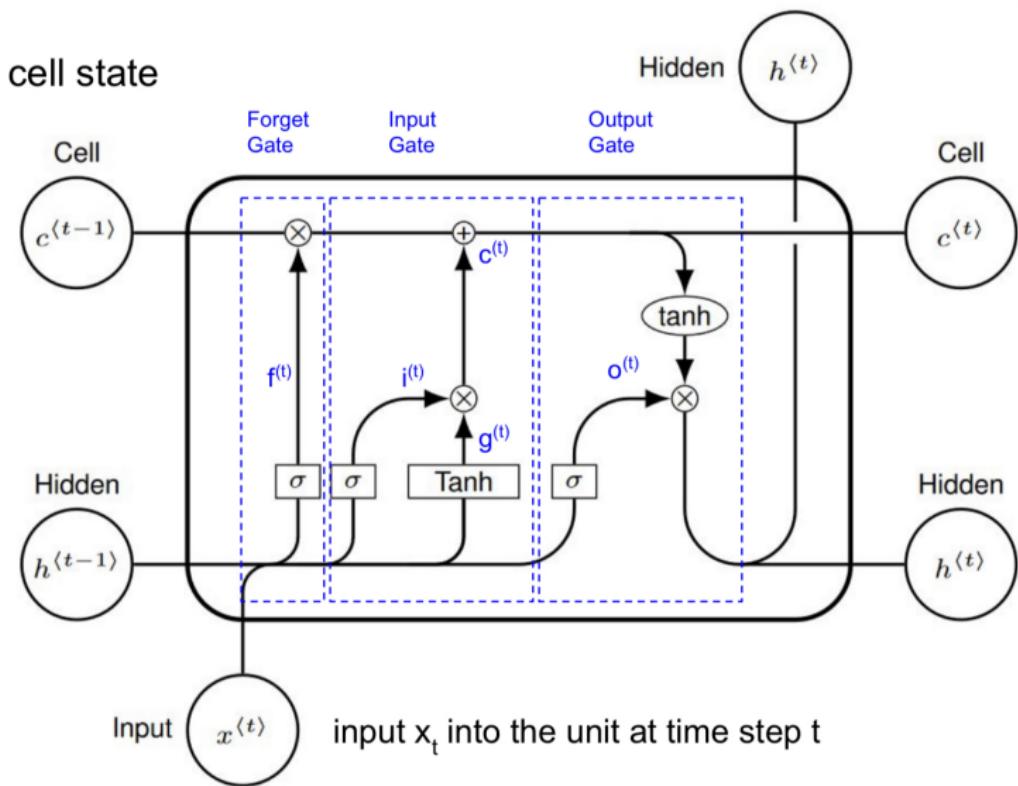
c) language modeling



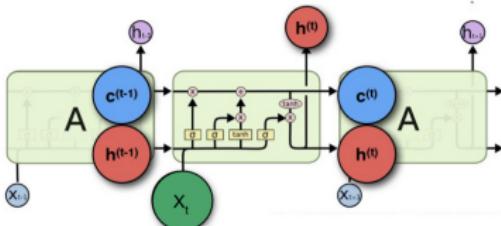
d) encoder-decoder

- RNN'lerin işleme noktasından uzak bilgileri kullanması zordur.
- Gizli durumlar genellikle en son girdi dizisinin parçalarına daha çok önem verir.
- RNN'lerde kritik bilgilerin ileri taşınmasında yetersizlikler görülür
 - Vanishing gradient problemi
- Çözüm olarak, ağır gereksiz bilgileri unutması ve önemli bilgileri hatırlaması sağlanır.
 - LSTM ve GRU gibi yapılar çözüm olarak önerilmiştir.

- **Uzun Dönem Bağımlılıkların Öğrenilmesi:** Geleneksel RNN uzun süreli bağımlılıkları öğrenmekte zorlanır. LSTM, özel tasarlanmış hücre yapısı ile bu sorunu çözer. Bilgiyi uzun süreler boyunca saklayabilen ve gerektiğinde bu bilgiyi tekrar ağır diğer kısımlarına aktarabilen hafıza hücreleri içerir.
- **Gradyan Kaybolması ve Patlaması:** RNN ağır derinleşmesiyle geriye doğru yapılan yayılım sırasında gradyanların çok küçülmesi (tanh) veya çok büyümesi (relu) sorunu yaşanır. LSTM, içerdiği kapı mekanizmaları (forget, input, ve output kapıları) ile bu sorunu çözer. Bu kapılar hücrenin ne kadar bilgiyi saklayacağını ve ne kadarını dışarı vereceğini düzenler.



- **Input** $x_{(t)}$: Zaman adımı t 'de LSTM birimine giren veri.
- **Hidden State** $h_{(t-1)}$ **ve** $h_{(t)}$: Sırasıyla önceki ve şuandaki zaman adımının durumlarını temsil eder.
- **Cell State** $c_{(t-1)}$ **ve** $c_{(t)}$: LSTM'nin uzun vadeli hafızasını temsil eder ve bilgiyi zaman içinde bir adımdan diğerine aktarır.
- **Forget Gate**: Hücre durumunun hangi kısmının unutulacağına karar verir. Sigmoid fonksiyonu σ ile 0 ile 1 arasında bir değer üretir (0'a yakın değerler unutulur).
- **Input Gate**: Yeni bilginin hücre durumuna ne kadarının ekleneceğini belirler.
- **Output Gate**: Mevcut hücre durumu $c_{(t)}$ ve giriş $x_{(t)}$ 'e bağlı olarak gizli durum $h_{(t)}$ 'un nasıl güncelleneceğine karar verir.



At time t , the LSTM cell reads in

- a c -dimensional **previous cell state vector** $c^{(t-1)}$
- a h -dimensional **previous hidden state vector** $h^{(t-1)}$
- a d -dimensional **current input vector** $x^{(t)}$

At time t , the LSTM cell returns

- a c -dimensional **new cell state vector** $c^{(t)}$
- a h -dimensional **new hidden state vector** $h^{(t)}$ (which may also be passed to an output layer)

- **Gated Recurrent Unit (GRU):** RNN'lerin uzun dönemli bağımlılıkları öğrenme sorunlarını çözmek için tasarlanmıştır.
- **Update Gate:** LSTM'deki unutma ve girdi kapılarının işlevlerini birleştirir ve hücrenin önceki durumunun ne kadarının korunacağına ve yeni bilgiyle ne kadarının güncelleneceğine karar verir.
- **Reset Gate:** Hücrenin geçmiş durumu ile yeni girdi arasındaki bağlantının ne kadar sıfırlanacağını belirleyerek yeni girdilerin önceki durumla ne kadar bütünlüğe sahip olabileceğini karar verir.
- **Gizli Durum:** Gizli durum, update ve reset kapılarının çıktıları ile mevcut girdinin birleştirilmesiyle güncellenir.
- **Parametre ve Bellek Kullanımı:** GRU'nun daha az kapısı olduğu için daha az parametreye sahiptir ve daha az bellek kullanır, bu da bazı durumlarda daha hızlı eğitim ve daha hızlı tahminlere yol açar.

