

Derin Öğrenme Deep Learning

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Simon Haykin, Neural Networks and Learning Machines, Prentice Hall, 2016." kitabı kullanılarak hazırlanmıştır.

İçerik

- ▶ Recurrent neural networks
- ▶ RNN'lerin yapısı
- ▶ RNN'lerde ileri geçiş
- ▶ RNN eğitimi
- ▶ RNN mimarileri
- ▶ RNN uygulamaları

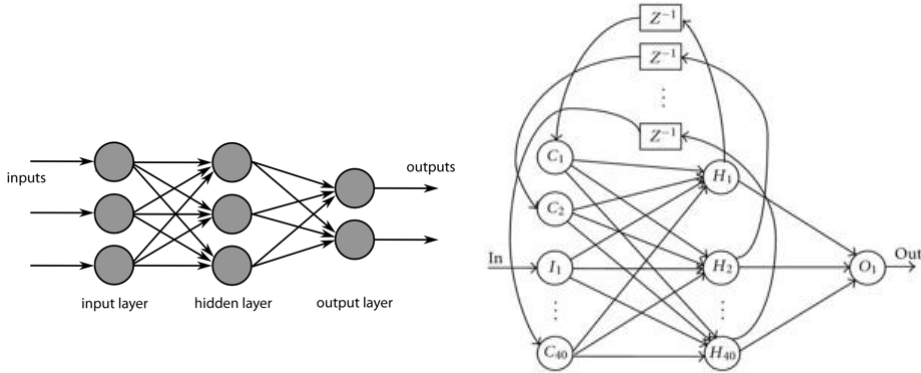
Recurrent neural networks

- ▶ Tüm problemler **sabit uzunlukta girişler ve çıkışlar ile ifade edilemez.**
- ▶ Konuşma tanıma, zaman serileri ve tahmin gibi problemler bilgiyi saklama ve kullanmayı gerektirir.
- ▶ Örneğin, girişteki bit dizisinde **1 sayısı çift ise EVET, tek ise HAYIR çıkışı** üreten sistemde önceki bilgiyi saklamak gerekir (1000010101 -> YES, 100011 -> NO).
- ▶ Bazı problemlerde sabit uzunlukta giriş her zaman mümkün olmayabilir ve giriş boyutu öncekilerden farklı olabilir.
- ▶ Tekrarlayan sinir ağları (**recurrent neural networks**) **önceki çıkışı** veya **gizli katmanın önceki durumlarını** da giriş olarak alır.
- ▶ Herhangi bir t zamanındaki **giriş, geçmiş bilgisi ile anlık girişin birleşimidir.**

3

Recurrent neural networks

- ▶ Klasik yapay sinir ağlarında **önceki durumlar veya girişler arasında ilişkilendirme yapılmaz.**
- ▶ RNN'lerde **önceki girişler veya durumlar ile ilişkilendirme yapılır.**



4

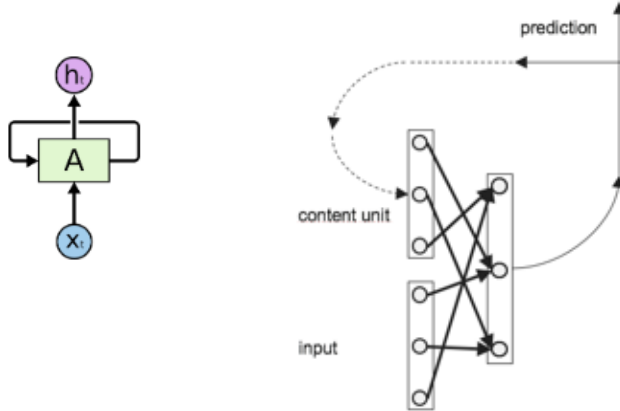
İçerik

- ▶ Recurrent neural networks
- ▶ RNN'lerin yapısı
- ▶ RNN'lerde ileri geçiş
- ▶ RNN eğitimi
- ▶ RNN mimarileri
- ▶ RNN uygulamaları

5

RNN'lerin yapısı

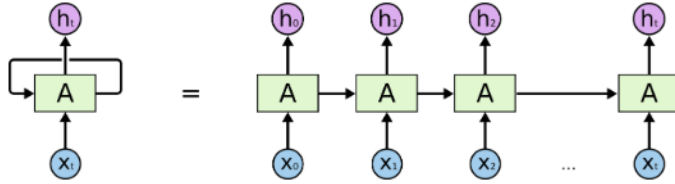
- ▶ RNN'ler döngüye sahiptir.
- ▶ Şekilde, **A** bir sinir ağını, x_t girişleri ve h_t ise çıkışı göstermektedir.



6

RNN'lerin yapısı

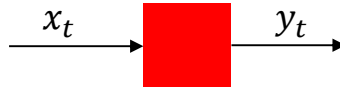
- ▶ Bir RNN, çok sayıda sinir ağı kopyası olarak düşünülebilir.
- ▶ Her sinir ağı bilgiyi sonrakine (girişe) aktarır.



7

RNN'lerin yapısı

- ▶ Basit ileri beslemeli ağlarda **her çıkış kendi girişi için hesaplanır.**



$$y_0 = f(W_x X_0)$$

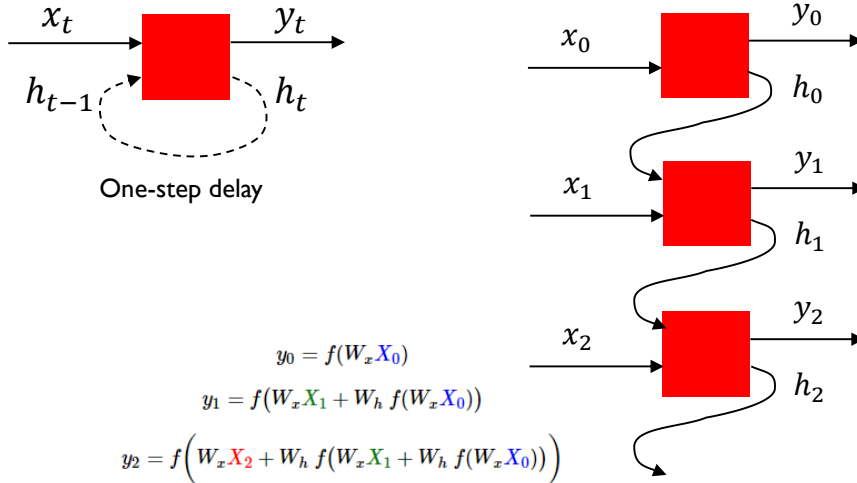
$$y_1 = f(W_x X_1)$$

$$y_2 = f(W_x X_2)$$

8

RNN'lerin yapısı

- RNN'lerde **her çıkış, kendi girişi ve önceki durumdan elde edilen girişe bağlı olarak hesaplanır.**



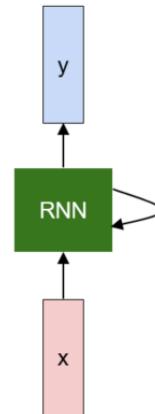
9

RNN'lerin yapısı

- Her zaman aralığında **aynı fonksiyon** ve **aynı parametreler** kullanılır.
- Katmanlar arasında ağırlıklar paylaşılarak kullanılır.**

$$h_t = f_W(h_{t-1}, x_t)$$

new state h_t is calculated using the old state h_{t-1} and the input vector x_t at some time step, through a function f with parameters W .



10

RNN'lerin yapısı

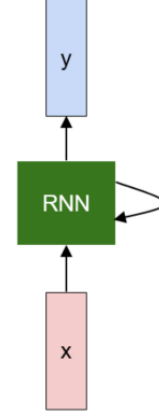
- ▶ RNN'lerde **önceki durum bilgisi belirli bir ağırlıkta sonraki çıkışları etkiler.**

$$h_t = f_W(h_{t-1}, x_t)$$



$$h_t = \tanh(W_{hh}h_{t-1} + W_{hx}x_t)$$

$$y_t = W_{hy}h_t$$

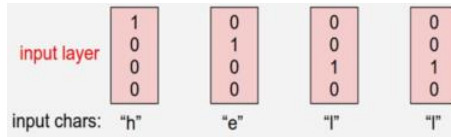


11

RNN'lerin yapısı

Örnek

- ▶ Sözlükte 4 harf olsun {h, e, l, o}.
- ▶ "hello" kelimesi için RNN oluşturalım.
- ▶ Giriş için **harfler vektöre dönüştürülür.**
- ▶ Kelimedeki her **harfin kendi sırası için 1, diğerleri 0** olacak şekilde giriş vektörleri oluşturulur.



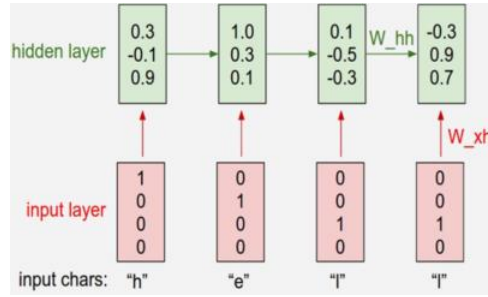
12

RNN'lerin yapısı

Örnek - devam

- Gizli katman çıkışları transfer fonksiyonu ile hesaplanır.

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

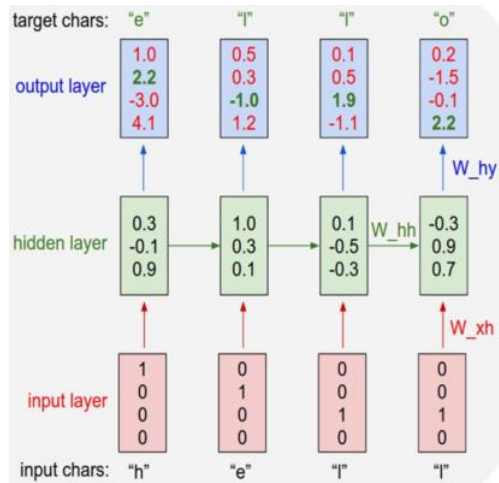


13

RNN'lerin yapısı

Örnek - devam

- Çıkışta hedeflenen vektöre göre hata hesaplanır.
- "h" karakteri girildikten sonra sonrakinin "e" olma olasılığı.
- "e" karakteri girildikten sonra sonrakinin "i" olma olasılığı.
- "i" karakteri girildikten sonra sonrakinin "i" olma olasılığı.
- "i" karakteri girildikten sonra sonrakinin "o" olma olasılığı.

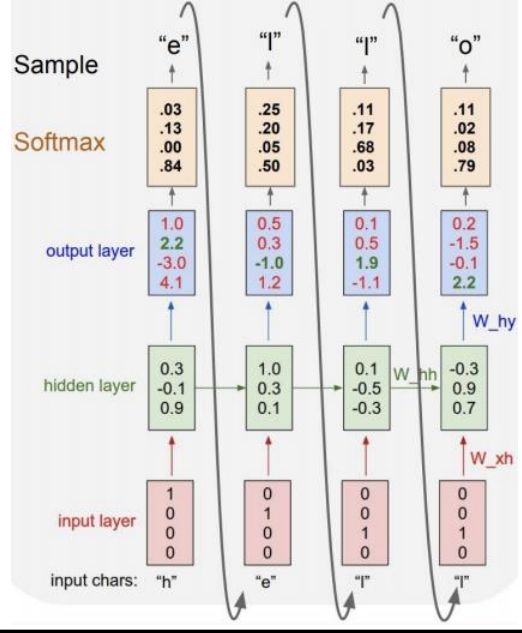


14

RNN'lerin yapısı

Örnek - devam

- ▶ Çıktılar girişe aktarılarak bir kelime/cümle oluşturulabilir.

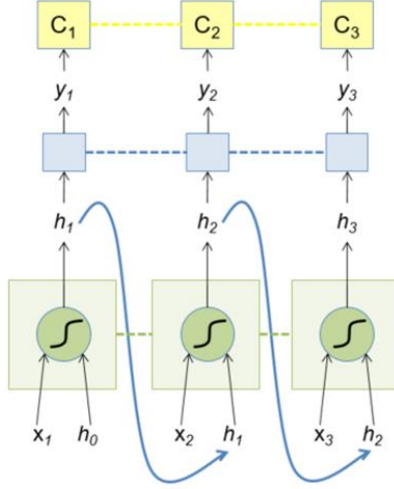


İçerik

- ▶ Recurrent neural networks
- ▶ RNN'lerin yapısı
- ▶ RNN'lerde ileri geçiş
- ▶ RNN eğitimi
- ▶ RNN mimarileri
- ▶ RNN uygulamaları

RNN'lerde ileri geiş

- ▶ Önceki ıkış sonraki giriş ile birleştirilerek yeni ıkış hesaplanır.



$$h_t = \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

$$y_t = F(h_t)$$

$$C_t = \text{Loss}(y_t, T_t)$$

----- indicates shared weights

17

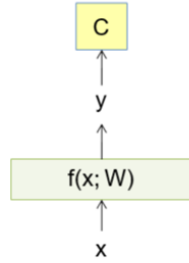
İerik

- ▶ Recurrent neural networks
- ▶ RNN'lerin yapısı
- ▶ RNN'lerde ileri geiş
- ▶ **RNN eğitimi**
- ▶ RNN mimarileri
- ▶ RNN uygulamaları

18

RNN eğitimi

- ▶ RNN'ler için eğitim backpropagation through time (BPTT) ile yapılır.
- ▶ **İstenen çıkıştaki hataya göre ağırlıklar değiştirilir.**



$$y = f(x; W)$$
$$C = \text{Loss}(y, y_T)$$

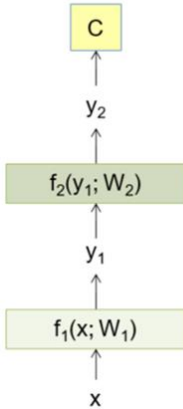
$$W \leftarrow W - \eta \frac{\partial C}{\partial W}$$

$$\frac{\partial C}{\partial W} = \left(\frac{\partial C}{\partial y} \right) \left(\frac{\partial y}{\partial W} \right)$$

19

RNN eğitimi

- ▶ **Çok katmanlı yapılarda** da geri yayılım yapılarak **ağırlıklar değiştirilir.**



$$y_1 = f_1(x; W_1)$$
$$y_2 = f_2(y_1; W_2)$$
$$C = \text{Loss}(y, y_T)$$

$$W_2 \leftarrow W_2 - \eta \frac{\partial C}{\partial W_2}$$

$$W_1 \leftarrow W_1 - \eta \frac{\partial C}{\partial W_1}$$

$$\frac{\partial C}{\partial W_2} = \left(\frac{\partial C}{\partial y_2} \right) \left(\frac{\partial y_2}{\partial W_2} \right)$$

$$\frac{\partial C}{\partial W_1} = \left(\frac{\partial C}{\partial y_1} \right) \left(\frac{\partial y_1}{\partial W_1} \right)$$

$$= \left(\frac{\partial C}{\partial y_2} \right) \left(\frac{\partial y_2}{\partial y_1} \right) \left(\frac{\partial y_1}{\partial W_1} \right)$$

20

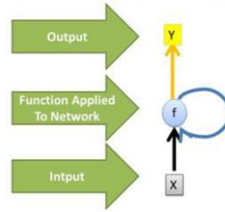
İçerik

- ▶ Recurrent neural networks
- ▶ RNN'lerin yapısı
- ▶ RNN'lerde ileri geçiş
- ▶ RNN eğitimi
- ▶ **RNN mimarileri**
- ▶ RNN uygulamaları

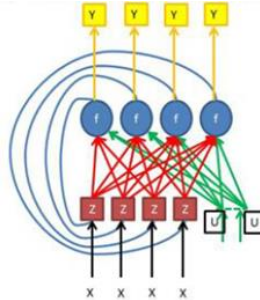
21

RNN mimarileri

- ▶ **Basit RNN mimarisi** aşağıdaki şekildedir.
- ▶ Giriş, çıkış ve önceki durumun aktarımı yapılır.



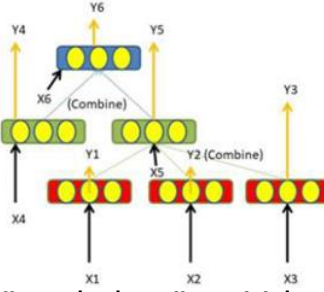
- ▶ **Fully connected RNN'lerde** tüm çıkışlardan girişlere aktarım yapılır.



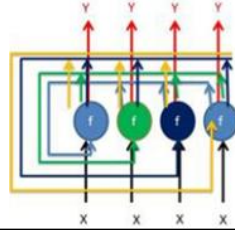
22

RNN mimarileri

- ▶ **Recursive sınır ağlarında** istenen katmana giriş yapılabilir ve istenen katmandan çıkış alınabilir.



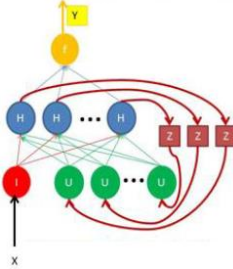
- ▶ **Hopfield ağında** tüm çıkışlar tüm girişlere aktarılır.



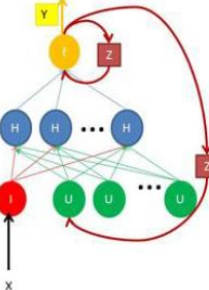
23

RNN mimarileri

- ▶ **Elman ağında** ara katman çıkışı girişlere aktarılır.



- ▶ **Jordan ağında** çıkışlar girişlere aktarılır.



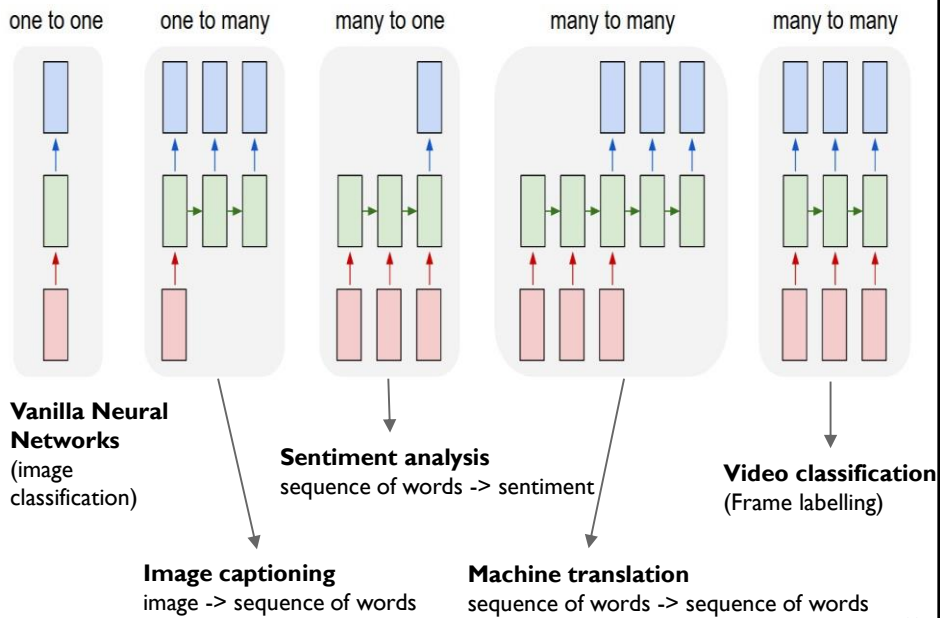
24

İçerik

- ▶ Recurrent neural networks
- ▶ RNN'lerin yapısı
- ▶ RNN'lerde ileri geçiş
- ▶ RNN eğitimi
- ▶ RNN mimarileri
- ▶ RNN uygulamaları

25

RNN uygulamaları

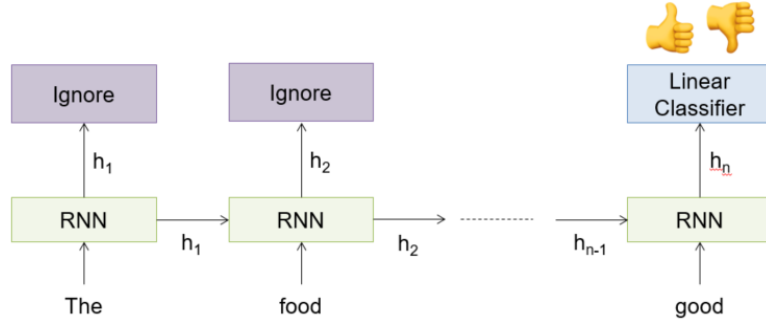


26

RNN uygulamaları

Sentiment Classification

- ▶ Çok sayıda cümle ile eğitilir.
- ▶ Daha sonra girilen cümleler için duygu durum sınıflandırması yapılır.
- ▶ Tek çıkış alınıp diğerleri göz ardı edilebilir.

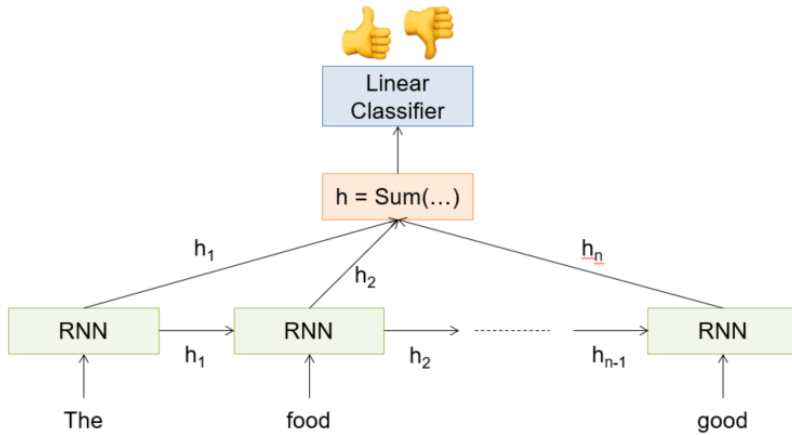


27

RNN uygulamaları

Sentiment Classification

- ▶ Tüm çıkışların toplamı birleştirilerek kullanılabilir.

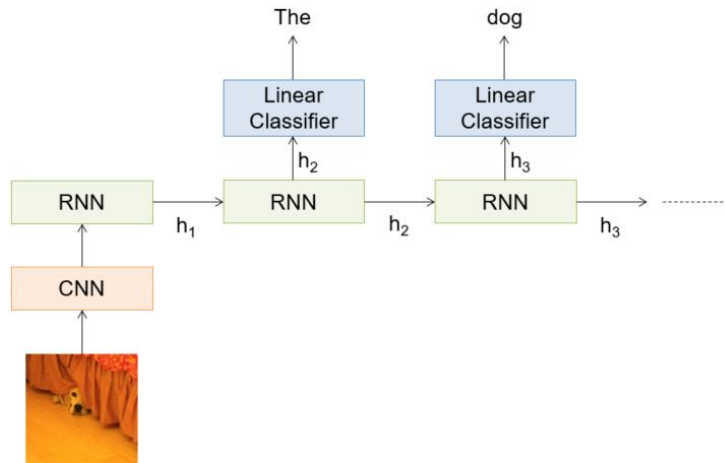


28

RNN uygulamaları

Image Captioning

- ▶ CNN ile birlikte görüntü etiketleme uygulamalarında kullanılmaktadır.



29

RNN uygulamaları

Image Captioning

- ▶ RNN ile görüntü etiketleme uygulamaları.

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A herd of elephants walking across a dry grass field.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A close up of a cat laying on a couch.



30

Ödev

- ▶ RNN'lerin NLP alanındaki uygulamasını içeren SCI/E dergilerinde yayınlanmış bir makale hakkında ödev hazırlayınız.