Makine Öğrenmesi

Final çalışma notları

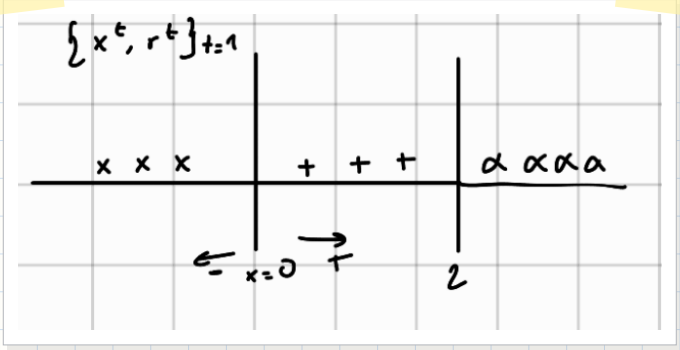
Orijinal not: Final-notu.pdf

> **UYARI** Bu belge, el yazısı ders notlarından Claude (Anthropic) yardımıyla dijitale aktarılmıştır. Yardımcı kaynak olarak hazırlanmıştır; hata veya eksik içerebilir. Lütfen orijinal notları ve ders kaynaklarını esas alın.
>
> **NOTICE** *This document was transcribed from handwritten lecture notes with the help of Claude (Anthropic). It is meant as a supplementary resource and may contain errors or omissions. Always check against the original notes and official course material.*

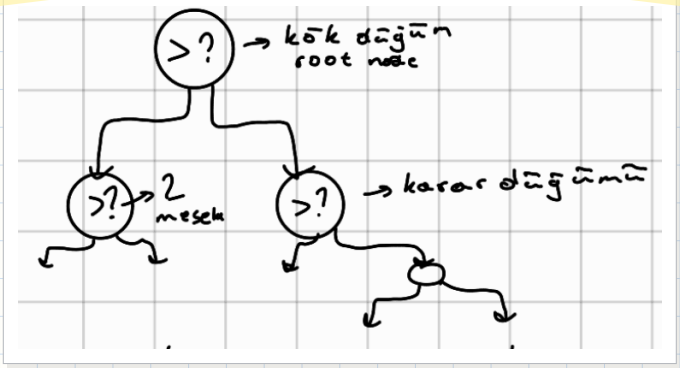
orj. s. 1

Ders6

Karar ağaçları (Decision tree)



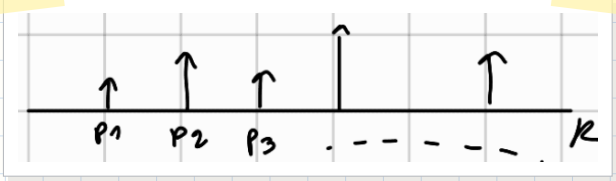
orijinal çizim, s. 1



orijinal çizim, s. 1

Amaç en sonda verileri homojen olarak ayırmak

Saflık ölçütleri (Impurity measures):



orijinal çizim, s. 1

$p_i = \frac{N_i}{N}$

en saf durum tek bir olasılığın 1 kalması

Entropy: $I_{entropy} = -\sum_{i=1}^{k} p_i \log p_i$

Gini Sabiti: $I_{Gini} = 1 - \sum_{i=1}^{k} p_i^2$

0 ile $1 - \frac{1}{k}$ arasında değerler alır

Tek bir sınıf varsa entropy'nin çok sınıf varsa max olur

dallanırken max entropy nerdeyse ordan devam et

$$\sum_{j=1}^{N} w_j I_j \qquad w_j = \frac{N_j}{N}$$

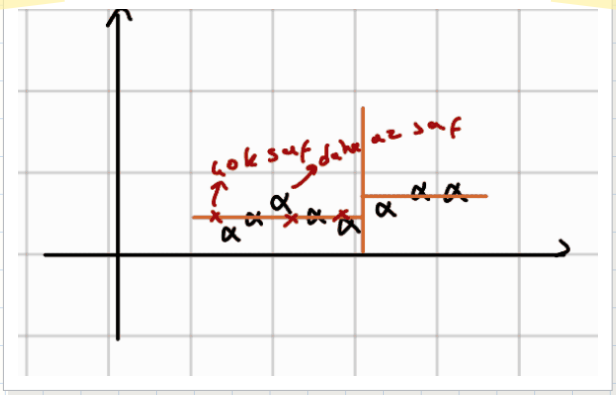
$$\varepsilon = \varepsilon_k - \varepsilon_{k-1}$$

while $\varepsilon > \gamma$ threshold dallan

$$L(\theta|x) = \sum_{j=1}^{N} w_j I_j + \underbrace{M\lambda \cdot f(\gamma)}_{\text{overfittingi engeller}}$$

? yaprak sayısı, λ = katsayı

Regresyon ağaçları



orijinal çizim, s. 1

Metrik olarak varyans kullanılabilir veya L_1 kullanılabilir. (Goodness of split)

L_1 norm aykırı değerlere hassasiyeti az



orijinal çizim, s. 1

Avantajları

- Basittir. Görsel temsili kolaydır
- Açıklanması en kolay makine öğrenmesi
- Hem sınıflandırma hem regresyon için kullanılabilir
- Doğrusal olmayan ilişkileri bulabilir

Dezavantajları

Overfitting: Çok sayıda düğümden oluşuyor ise overfitting olabilir

Kararsızlık: Verilerdeki ufak değişiklik tamamen farklı bir ağaç oluşmasına neden olabilir

Çözüm: 1- overfitting için budama yapılabilir

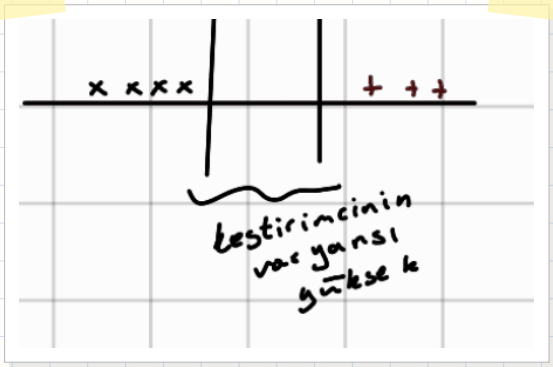
Ön budama: önceden max derinlik vererek ağacın uzaması engellenir

Son budama: Tam gelişmiş ağaçtan önemli katkı sağlamayan dallar kaldırılır

orj. s. 2

Ensemble Learning

Bagging



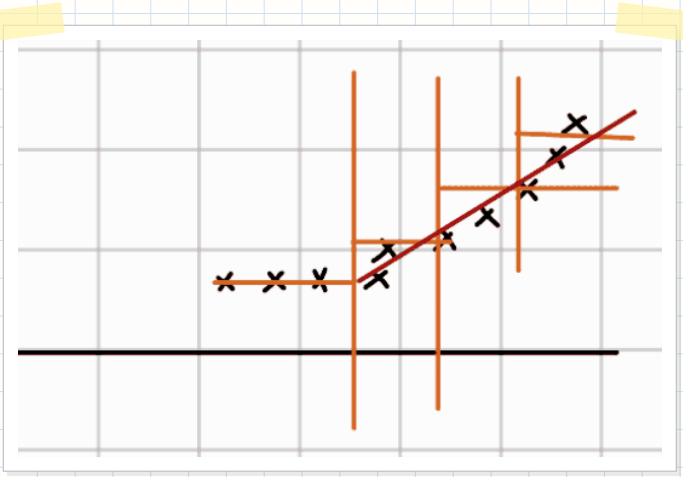
orijinal çizim, s. 2

100 tane karar ağacı alırım, yeni bir örnek geldiği zaman ortalama yapıyorum böylece kestirimcinin varyansı küçülür.

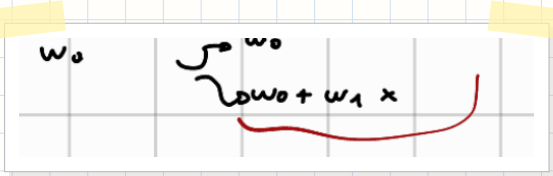
Baggingte olan eğitim seti alt eğitim setlerine bölünür. Her veri setiyle Model eğitilir, böylece alt veri setleri kadar model oluşturulur. Tahminlerin birleştirilmesi aşamasında regresyon ağaçları için ortalama alınırken sınıflandırma ağaçlarında sonuçlar oylama ile belirlenir.

Boosting

Boosting: Verinin özel bir bölgesine oturan daha iyi bir sınıflandırıcı varsa bu şekilde ?yapılabilir



orijinal çizim, s. 2



orijinal çizim, s. 2

> *Şekil:* Bu sayfa basılı materyal içeriyor (2. Yükseltme (Boosting) yazısı, kdnuggets.com'dan alıntı); dijitalleştirilmedi.
> *(orijinal notta, s. 2)*

orj. s. 3

Adaboost

$f_i(x),\ i \in \{0, \ldots, M\}$ (Aynı modelin parametrelerini değiştirip linear kombinasyonlarını kullanıyoruz)

$$y = \sum_{i=1}^{M} w_i f_i(x)$$

verinin hangi bölgesinde hangi fonksiyon daha iyi

$$L(\theta|X) = \sum_{t=1}^{N} w_t \left(\hat{r}^t - r^t\right)^2$$

↑ risk gibi

Bu modelde eğitim kümesi önce zayıf öğrenici ile eğitilir. Eğitim sonrası yanlış olarak tahminlenen örnekler bu algoritma için önemlidir. Bir sonraki eğitimde ilk tahminlemede yanlış öğrenilen eğitim verilerine daha fazla öncelik verilerek yani ağırlıkları arttırılarak tekrar eğitilir.

Ders7

Boyut Azaltma

(1) Feature selection

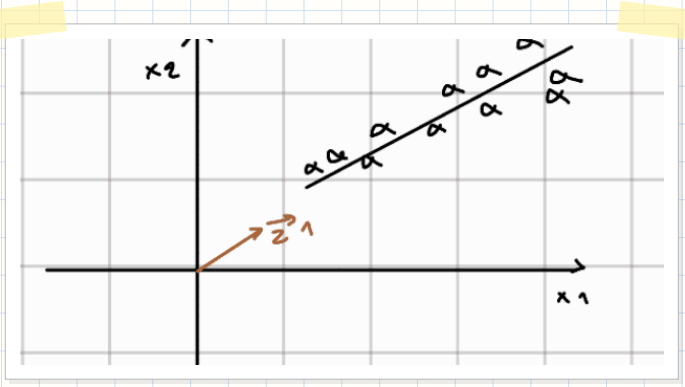
$$\{x^t, r^t\}_{t=1}^N$$

$$x^t_{N_1 \times 1} \rightarrow x^t_{N_2 \times 1} \qquad N_2 < N_1$$

(2) Subset Selection

$F = \emptyset$
while not converge:
  for $x_i \notin F$
    $F_i \leftarrow F \cup x_i$
    $L_i(x|F_i)$
  $i = \arg\min L_i(x|F_i)$
  $F \leftarrow F_i$

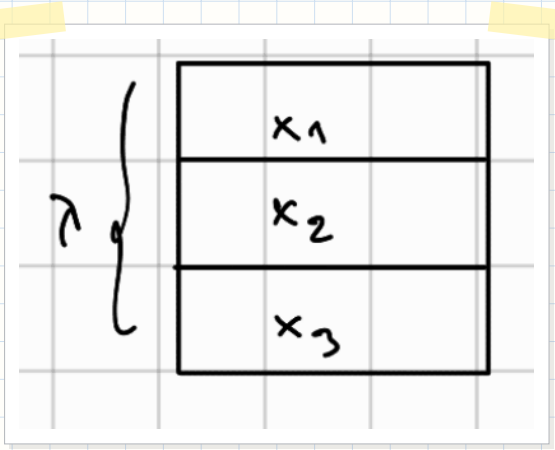
(3) PCA (Principle component Analysis)



orijinal çizim, s. 3

Feature Embedding

$$WDW^T = X \cdot X^T$$



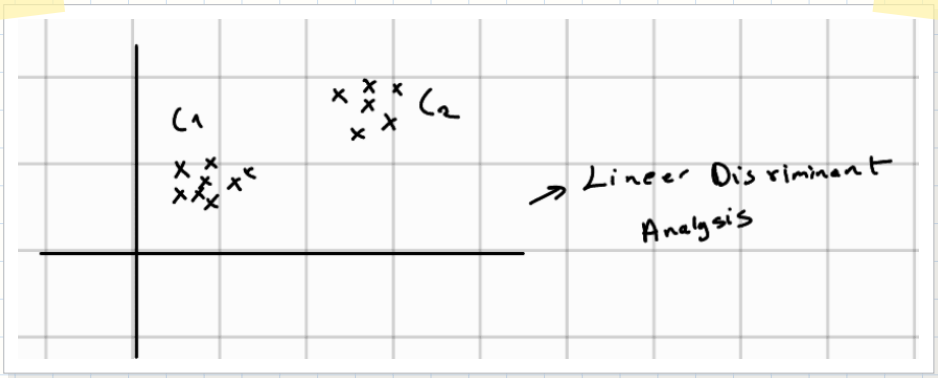
orijinal çizim, s. 3

Singular Value Decomposition

SVD: $X = [x_1\ x_2\ \cdots\ x_N]_{N\times d}$

$$= \underbrace{V}_{N\times r}\ \underbrace{\Lambda}_{r\times r}\ \underbrace{W^T}_{r\times d}$$

LDA (Linear discriminant Analysis)



orijinal çizim, s. 3

$$z = w^T x$$

$$m_1 = \frac{\sum_t w^T x^t r^t}{\sum_t r^t} = w^T \vec{m}_1$$

$$m_2 = \frac{\sum_t w^T x^t (1 - r^t)}{\sum_t (1 - r^t)} = w^T \vec{m}_2$$

$$s_1^2 = \sum_t (w^T x^t - m_1)^2 r^t$$

$$s_2^2 = \sum_t (w^T x^t - m_2)^2 (1 - r^t)$$

$$J(w) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} = \frac{(w^T \vec{m}_1 - w^T \vec{m}_2)^2}{s_1^2 + s_2^2}$$

$$= \frac{w^T (\vec{m}_1 - \vec{m}_2)(\vec{m}_1 - \vec{m}_2)^T w}{s_1^2 + s_2^2}$$

$$= \frac{w^T S_B w}{s_1^2 + s_2^2}$$

orj. s. 4

$$s_1^2 = \sum_t (w^T x^t - m_1)^2 r^t$$

$$= w^T S_1 w \qquad S_1 = \sum_t (x^t - \vec{m}_1)(x^t - \vec{m}_1)^T r^t$$

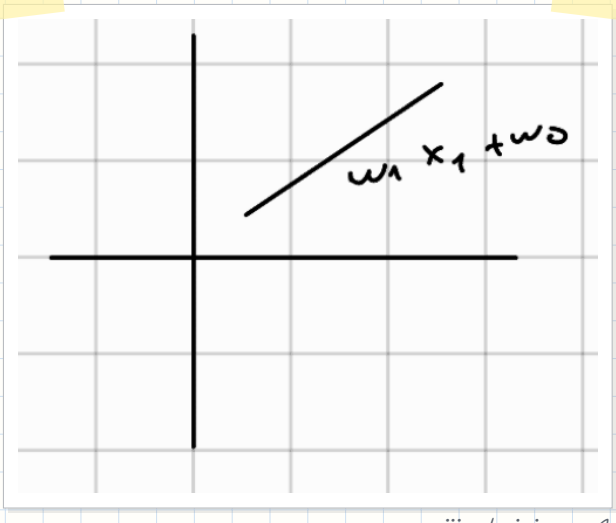
$$s_1^2 + s_2^2 = w^T S_W w = \vec{w}^T (S_1 + S_2) w$$

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

$$\frac{\partial J(w)}{\partial w} = 0$$

$$= \frac{w^T (\vec{m}_1 - \vec{m}_2)}{w^T S_W w}$$

Ders8



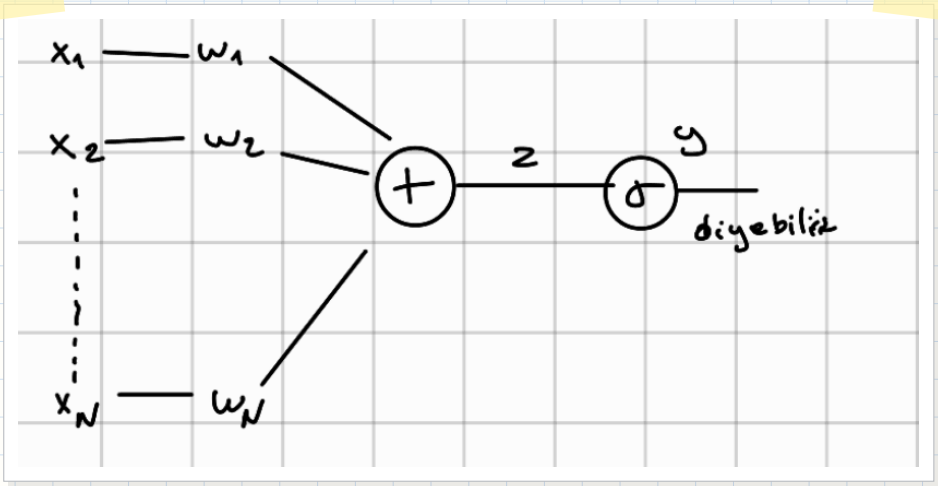
orijinal çizim, s. 4

$$\vec{w}^T x + b = \hat{r}^t$$

$$L(\theta|X) = \sum_{t=1}^{N} (\hat{r}^t - r^t)^2$$

türevini alıp 0'a eşitlenerek regresyon

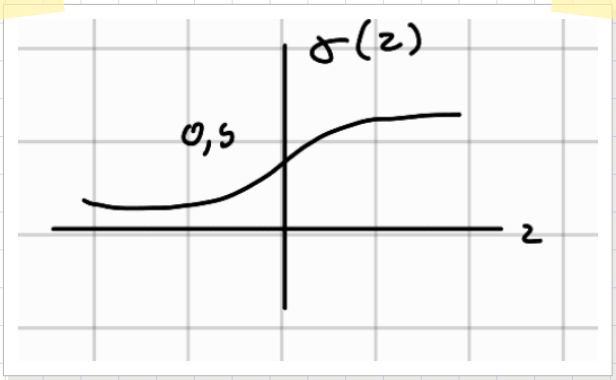
Başka yöntem



orijinal çizim, s. 4

Sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



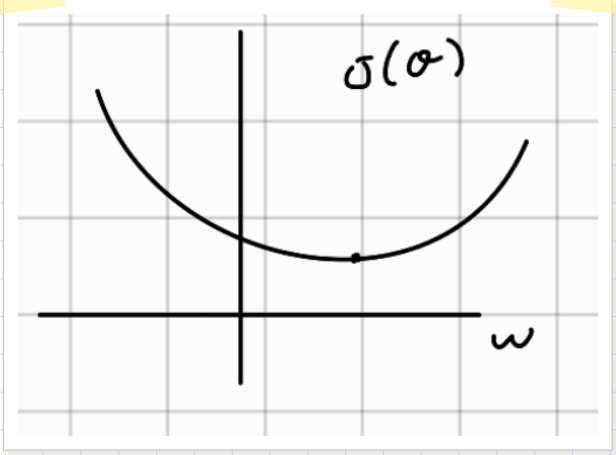
orijinal çizim, s. 4

Bu sefer iteratif olarak ilerliyoruz

gradient descent

$$w_i \leftarrow w_i - l \frac{\partial J}{\partial w_i}$$

J: maliyet fonksiyonu, l: ? lr



orijinal çizim, s. 4

Back Propagation

$$\hat{y}_i = \sigma(z_i) = \frac{1}{1+e^{-z_i}} \qquad z_i = w^T \cdot x_i + b$$

$$J(w) = -\frac{1}{N}\sum_{i=0}^{N} \underbrace{y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)}_{\text{cross entropy}}$$

$$\frac{\partial J(w)}{\partial w} = \frac{\partial J(w)}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial w}$$

? okunamadı

$$\frac{\partial J(w)}{\partial \hat{y}_i} = \frac{-y_i}{\hat{y}_i} + \frac{1-y_i}{1-\hat{y}_i} \qquad \frac{\partial \hat{y}_i}{\partial z_i} = \hat{y}_i(1-\hat{y}_i)$$

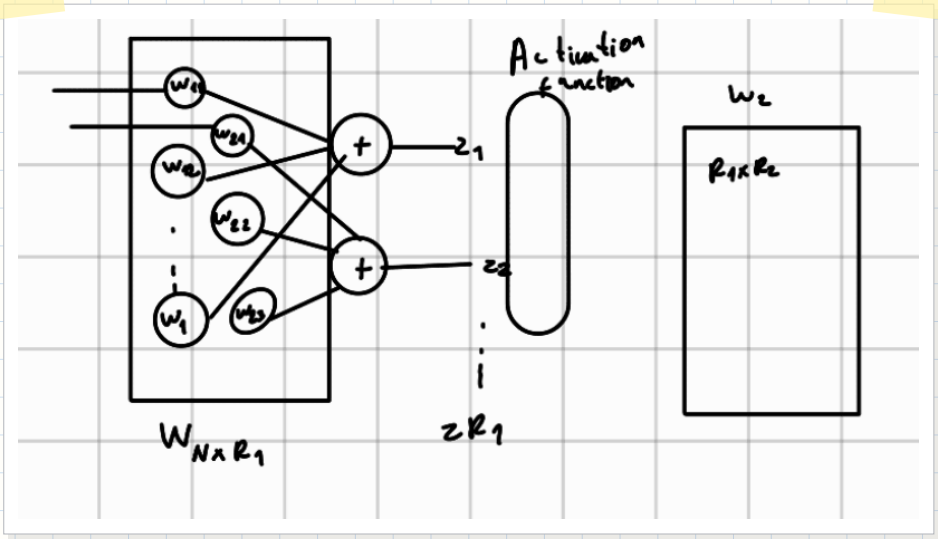
$$\frac{\partial z_i}{\partial w} = x_i$$

$$\frac{\partial J(w)}{\partial w} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)x$$

sonra bunu ekleyeceğiz, geri ? yayıyoruz

$$\frac{\partial J(w)}{\partial b} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)$$

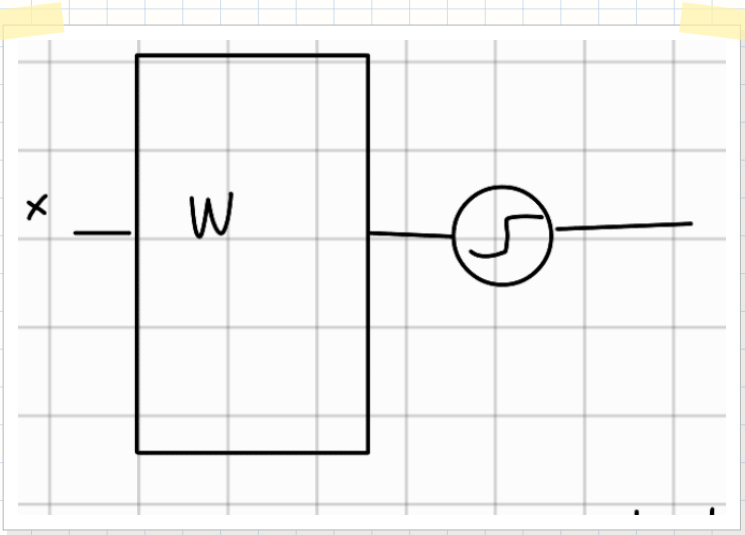
$$x = \{x^t, r^t\}_{t=1}^{M} \qquad P(c_0) = \hat{y}_i, \quad P(c_1) = (1-\hat{y}_i)$$



orijinal çizim, s. 4

orj. s. 5

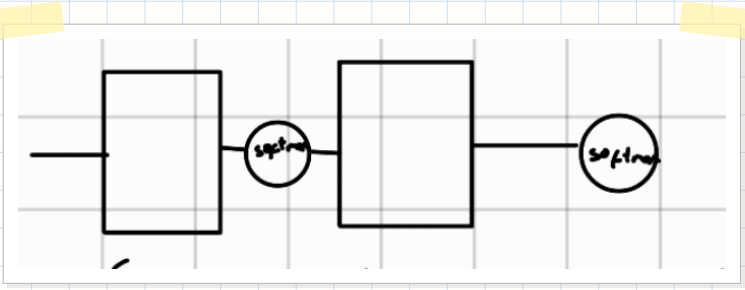
Activation layers



orijinal çizim, s. 5

(1) softmax

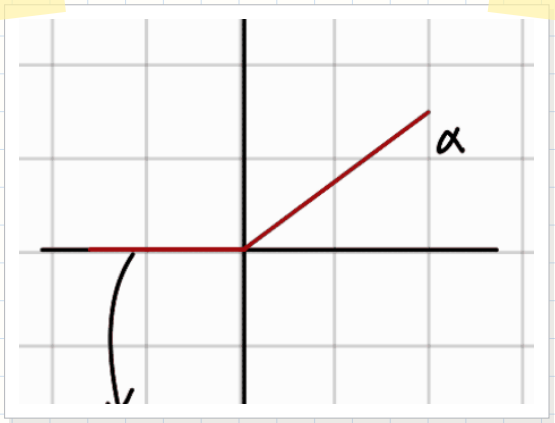
$$\text{softmax} = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} \qquad \text{olasılık dağılımı}$$



orijinal çizim, s. 5

Sonsuza giderken türev 0 o yüzden iyi çalışmaz. Bu yüzden genelde softmax en son katmanda kullanılır

(2) Relu

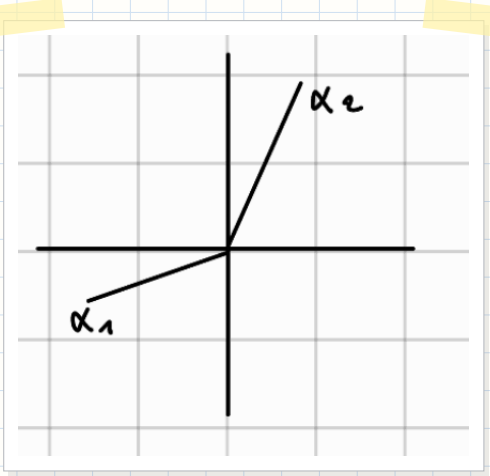


orijinal çizim, s. 5

a = b şeklinde gösteriyoruz (non-linear)

negatif tarafta yine türev 0, dead neuron sorununa yol açıyor (Vanishing gradient)

(3) Leaky Relu



orijinal çizim, s. 5

non linearity

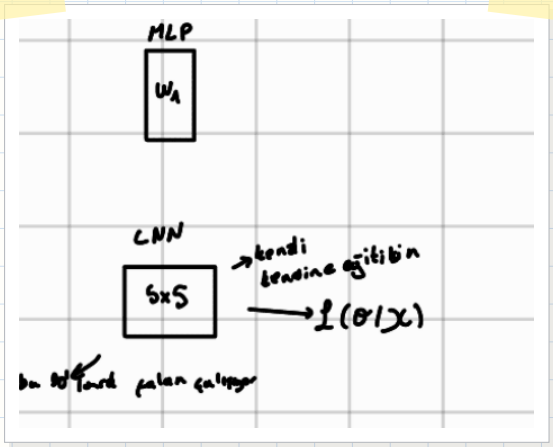
(4) tanh

$$\tanh = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

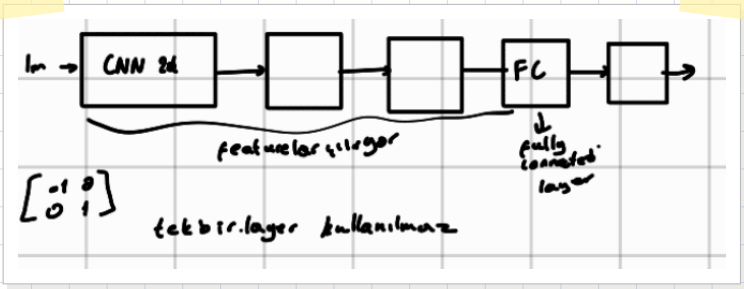
bunda da türevler $+\infty$, $-\infty$ da 0'a yaklaşıyor

> **Şekil:** Tanh fonksiyonunun basılı grafiği (ekran görüntüsü); dijitalleştirilmedi.
> *(orijinal notta, s. 5)*

Ders9

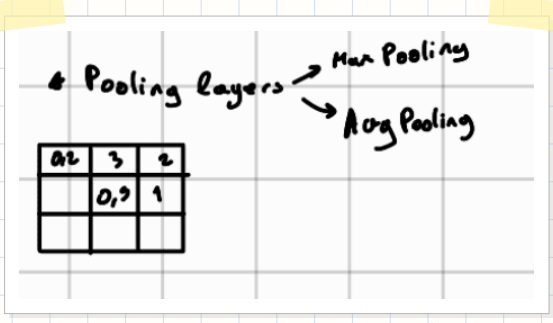


orijinal çizim, s. 5



orijinal çizim, s. 5

tek bir layer kullanmaz



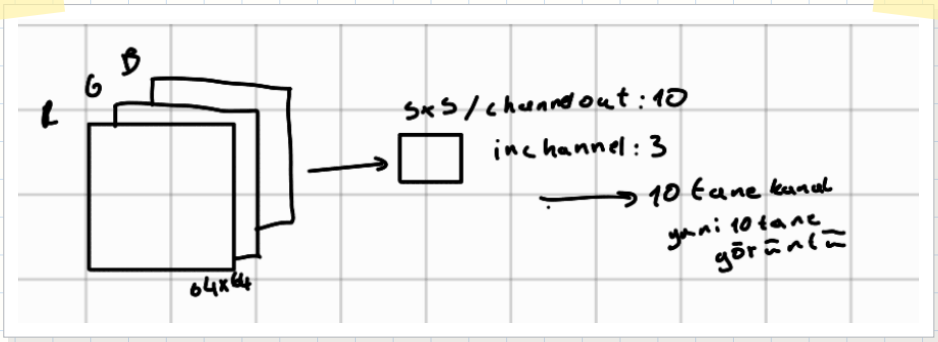
orijinal çizim, s. 5

kernel size, filtre sayısı

Feature extraction için çok kullanılır

Odaklanıp işimize yarayacağımız verilerin her şeyi bundan değil.

orj. s. 6



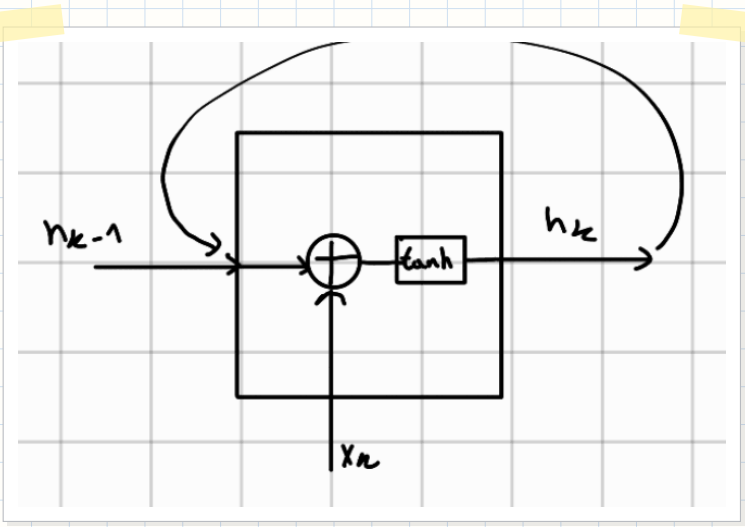
orijinal çizim, s. 6

(1) feature extraction: Öznitelik çıkarma
(2) Özniteliklerin konumlarından bağımsız
(3) Less parameters
eğitimi ?okunamadı göre daha zor

Transfer learning

Elimizdeki veriler ne kadar kaliteliyse o kadar kolay sınıflandırıcımız olur. Yolo feature extraction'ı çok iyi yapıyor ve iyi parçalara ayırıyor

Time series prediction



orijinal çizim, s. 6

Patlayan Gradyan (Exploding Gradient):

Geri yayılım sırasında gradyanlar çok büyük değerlere ulaşır ve bu durum ağırlıkların aşırı güncellenmesine neden olur. Modelin kararsız hale gelmesine yol açar. Eğitim süreci başarısız olur

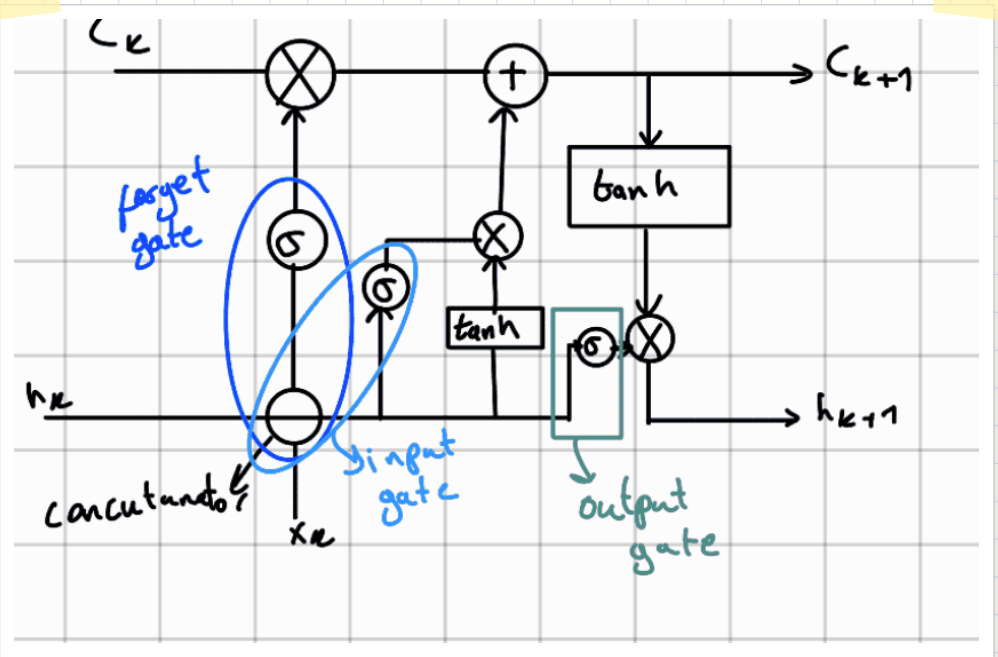
Yok olan Gradyan (Vanishing Gradient):

Geriye yayılım sırasında gradyanlar çok küçük değerlere düşer. Bu durumda ağırlıklar güncellenemez ve model, özellikle uzun dizilerde zaman bağımlılıklarını öğrenemez.

Nedeni ise RNN geri yayılım sırasında zincir kuralını kullanır. Bu süreçte zincirleme türev alınır:

$$\frac{\partial \mathcal{L}}{\partial W} = \prod_t \frac{\partial h_t}{\partial h_{t-1}}$$

LSTM



orijinal çizim, s. 6

Encoding

ham bir veriyi sayıya çevirmek

kedi : 0 köpek 1

One hot Encoding

R:0 G:1 B:2

$$R = [1,0,0]$$
$$G = [0,1,0]$$
$$B = [0,0,1]$$

orj. s. 7

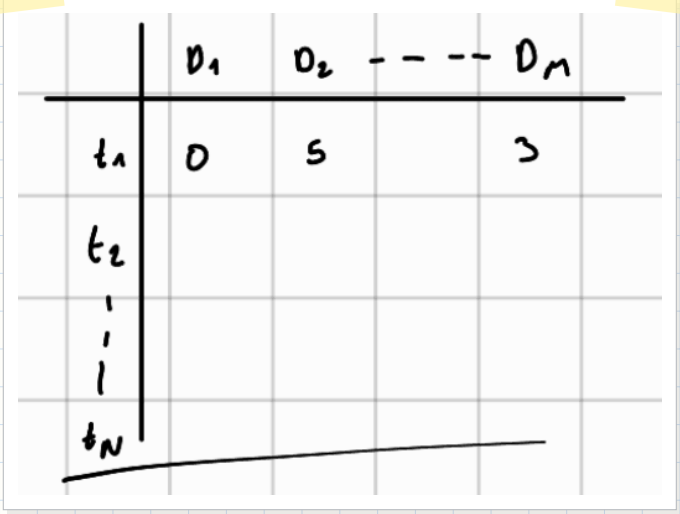
Ali topu at
↪ tokenization

Al/i /to/pu/ /a/t/

gözetimsiz olarak entropy kullanılır

king - man = queen - woman

en basit şekilde nasıl vektöre çeviririm



orijinal çizim, s. 7

TF-IDF

$TF(t, d)$

$IDF(t)$

Termfreq (TF)

bazı çok kullanılan kelimeler var

NLP (Natural language processing)

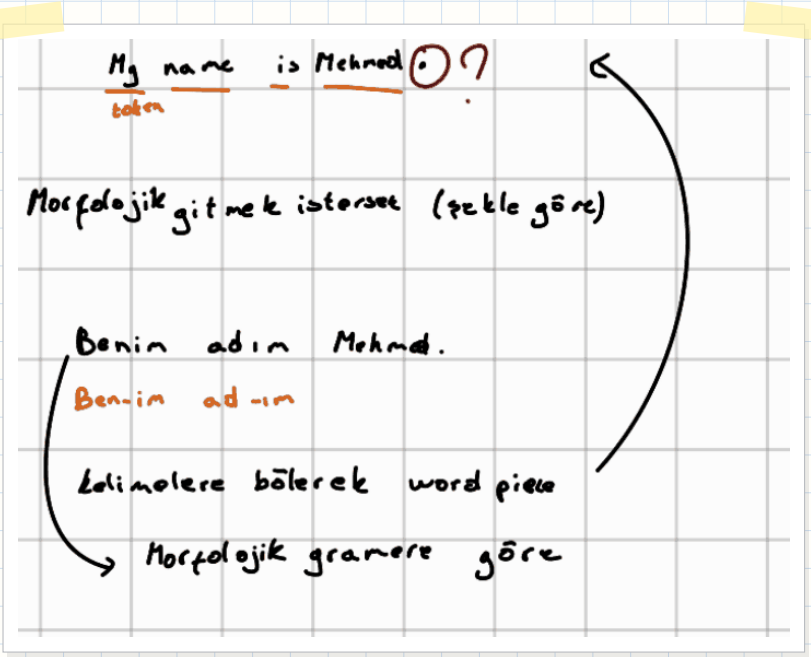
Tokenization

Morfolojik gitmek istersek (?eke göre)

Benim adım Mehmet.

Kelimelere bölerek word piece

↪ Morfolojik gramere göre



orijinal çizim, s. 7

Türkçede 10000 kelime

10.000

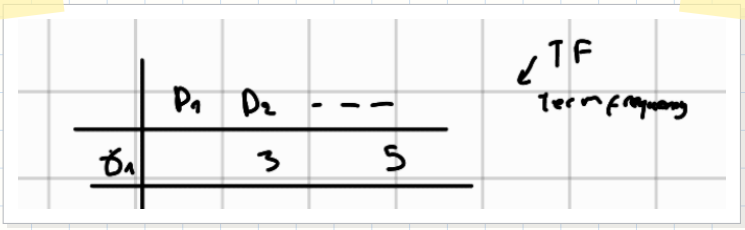
ahmed → 1000

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

One hot encoding

TF-IDF

Elimizde dokümanlar var



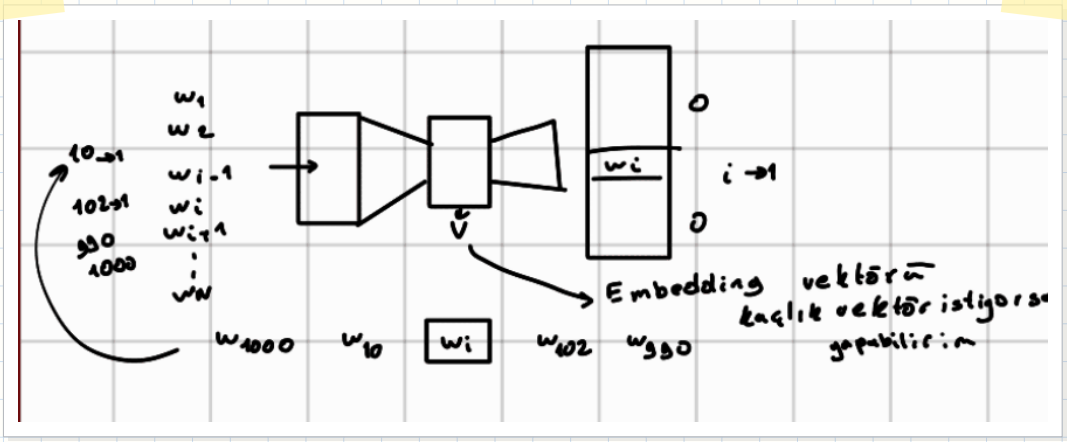
orijinal çizim, s. 7

Bow TF

Tüm dökümanlarda iki kelime aynı frekansla geçmemeli

Word2Vec

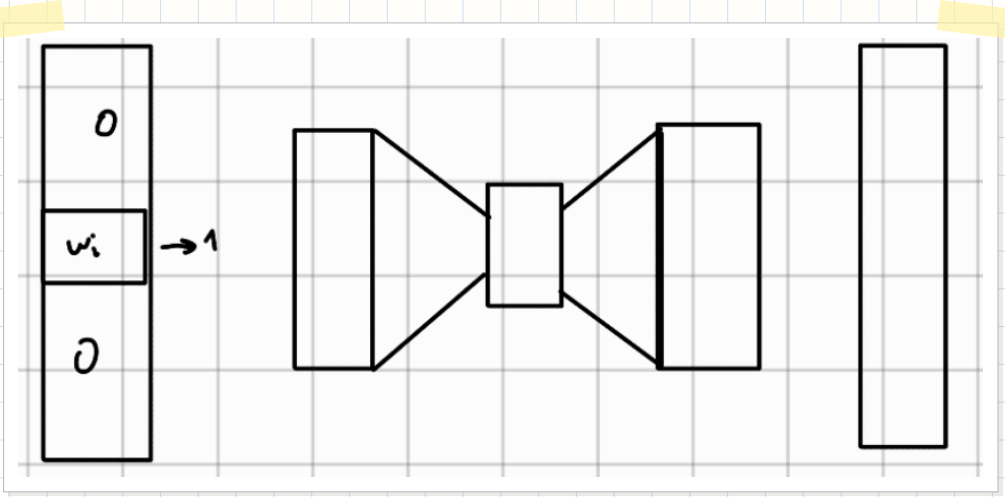
CBOW Continuous Bag of Words



orijinal çizim, s. 7

Skip gram / n-gram **2 gram iki komşuluk**

tersten de yapabiliriz



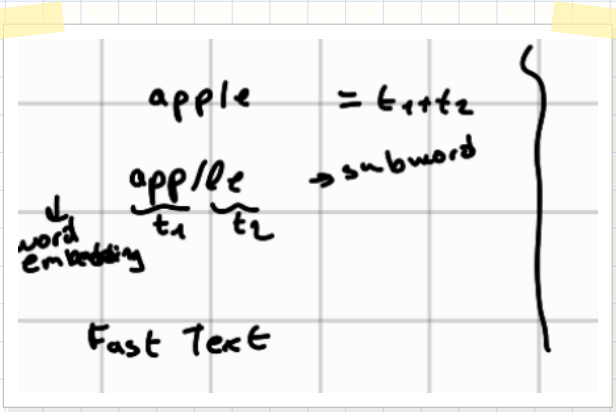
orijinal çizim, s. 7

Global vectors for word Representation (Glove)

$$J = \sum_{i,j} f(x_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$

i'inci döküman, j'inci kelimenin geçme frekansı

bunu bulmaya çalışıyoruz, kelimenin ağırlığı

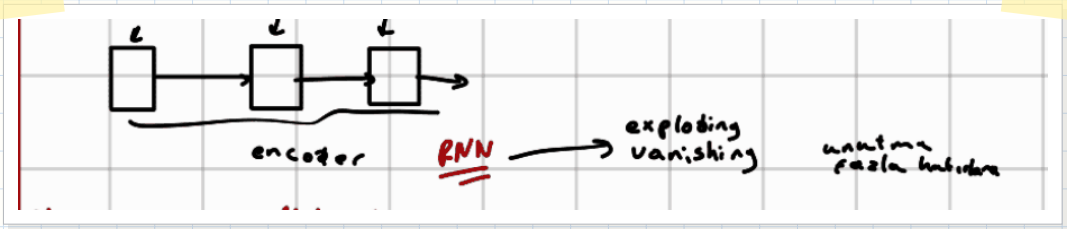


orijinal çizim, s. 7

Static embedding

Covid geldi yeni kelime

Benim adım Mehmet / Seq 2 Seq ✓ Supervised



orijinal çizim, s. 7

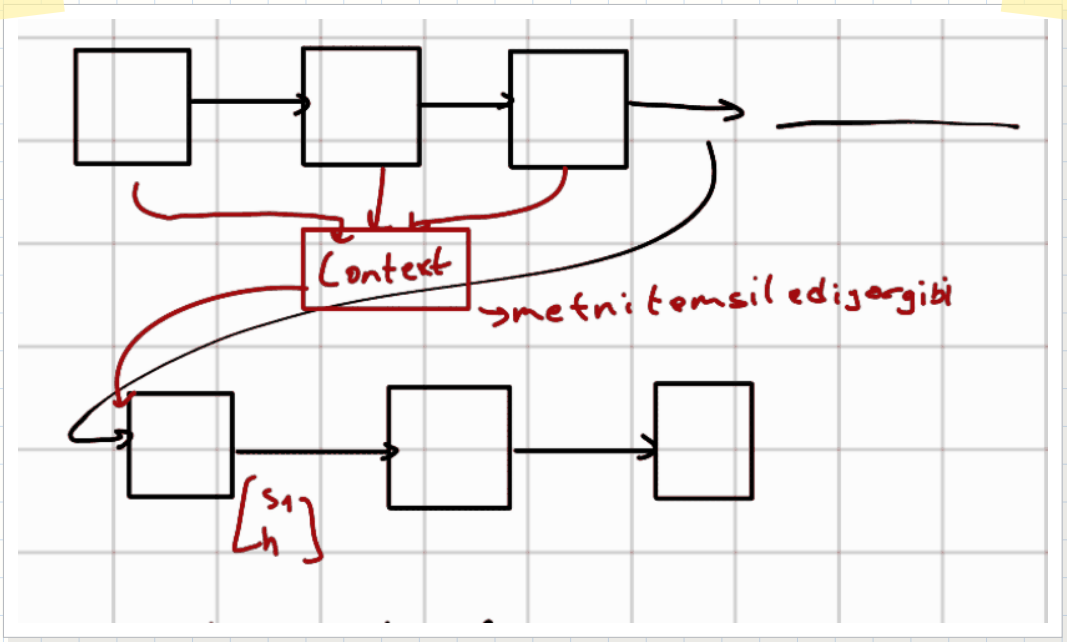
My name is Mehmet **sonraki güncellemede LSTM**

w_1 $\tilde{w}_2$ $\tilde{w}_3$ $\hat{w}_4$

Bu metin kısa olmadıktan sonra context'i yakalayamıyor.

orj. s. 8

Attention Mechanism



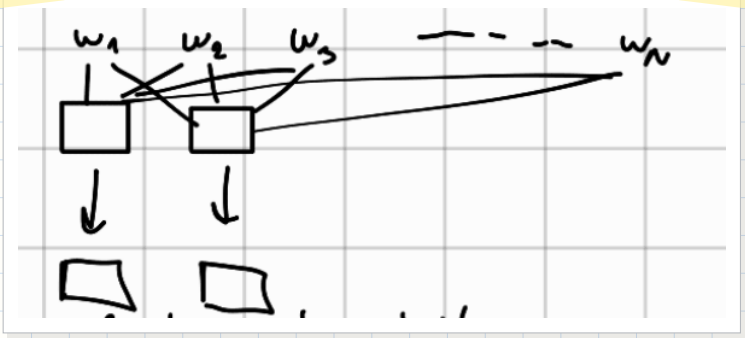
orijinal çizim, s. 8

kelimeleri hidden layerle çarpıyoruz

$w \cdot h \xrightarrow{\text{softmax}} [\alpha_1, \alpha_2, \ldots, \alpha_{100}]$

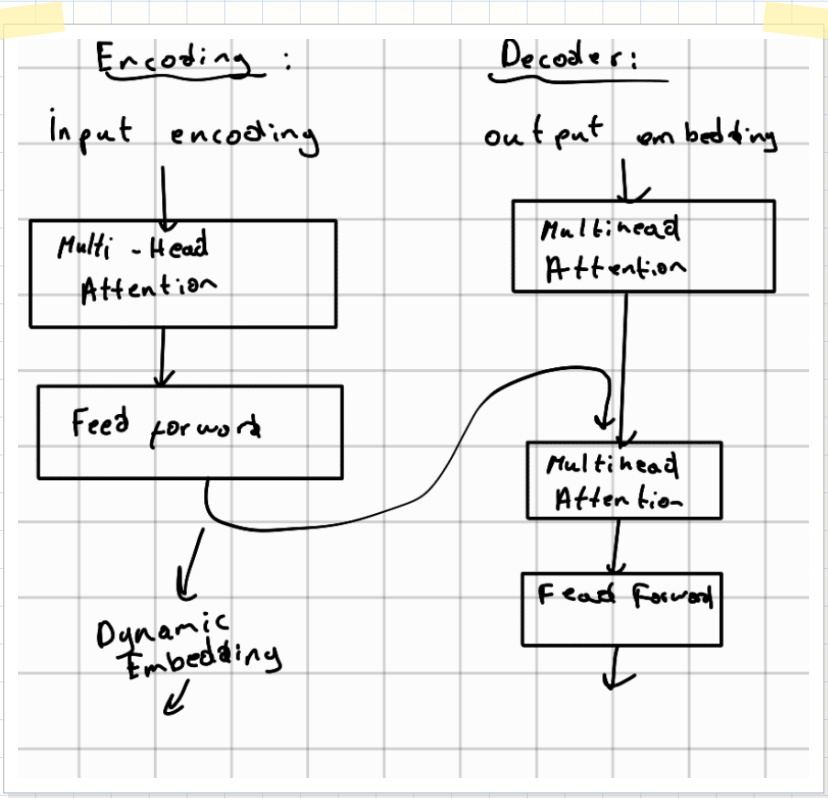
hangi kelime ne kadar önemli.

Sonra döngü yerine MLP koyuyoruz



orijinal çizim, s. 8

paralelde çalışabiliyor



orijinal çizim, s. 8

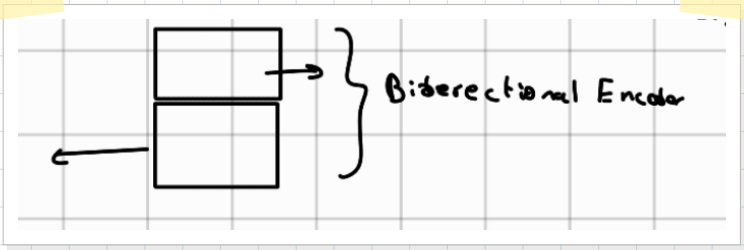
BERT: Sadece Encoder, kelimeleri öğrenebilen

BERT NLP'nin ?*gelişimi*

BERT: Bidirectional Encoder Representations from Transformers

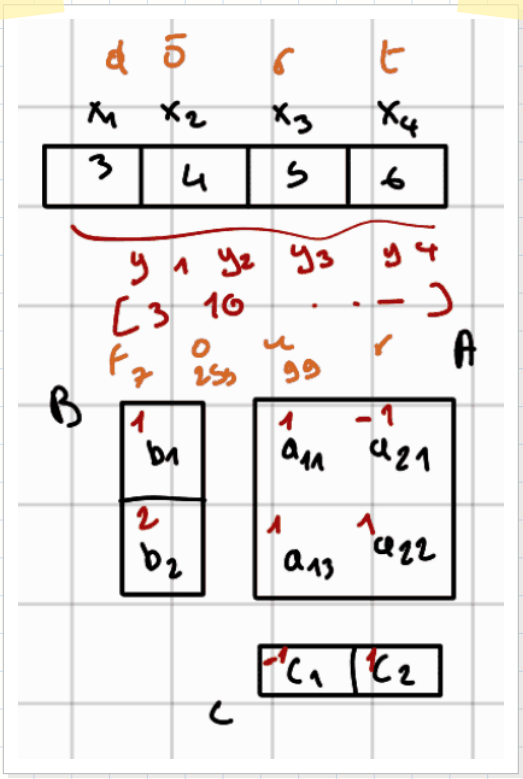
ELMO: Embeddings from language model

context mekanizmasıyla

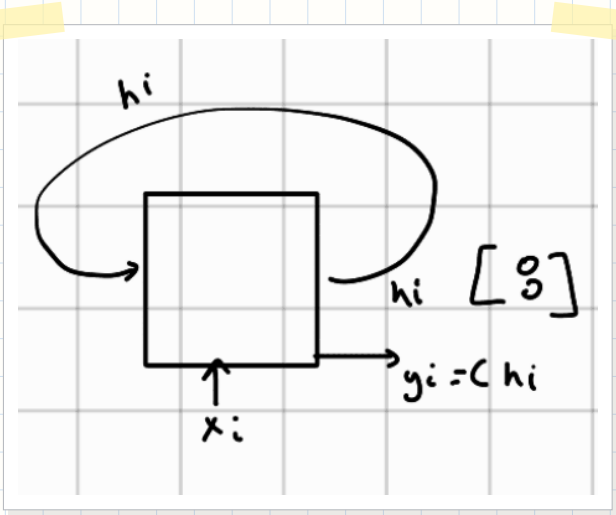


orijinal çizim, s. 8

RNN



orijinal çizim, s. 8



orijinal çizim, s. 8

$$(1)\quad [B\,;\,A]\begin{bmatrix} x_t \\ h_i \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

$$C h_i = [\,3\,] = y_i$$

$$(2)\quad \begin{bmatrix} 1 & 1 & -1 \\ 2 & 1 & 1 \end{bmatrix}\begin{bmatrix} 4 \\ 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 17 \end{bmatrix}$$

$$y_2 = 16$$

MSE = ?

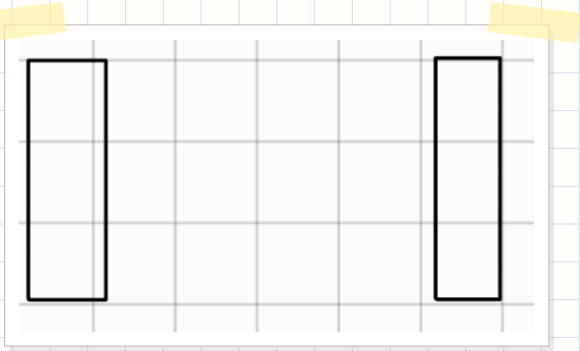
$$\hat{y} = f\left(CA\begin{bmatrix} x \\ h \end{bmatrix}\right)$$

$$L = (\hat{y} - y)^2$$

orj. s. 9

I run

I university bus run



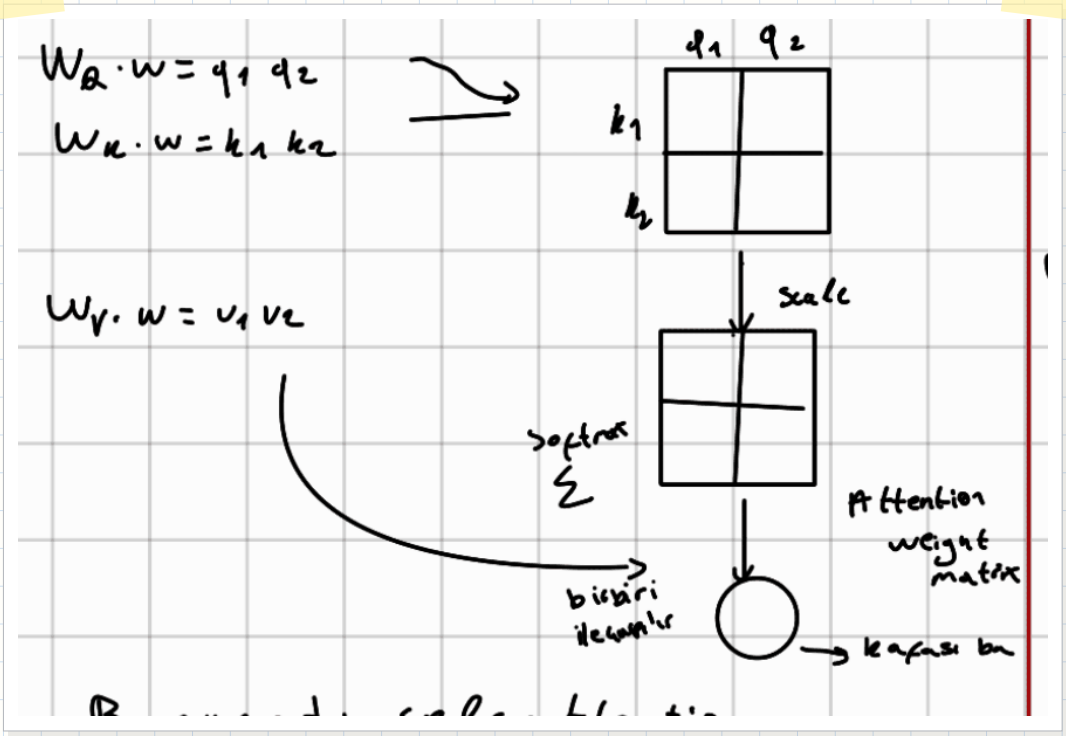
orijinal çizim, s. 9

Q = Query attention = $\text{softmax}\left(\frac{k^T \cdot Q}{\sqrt{d_k}}\right) V$ (key'in boyutu)

k = key

V = Value

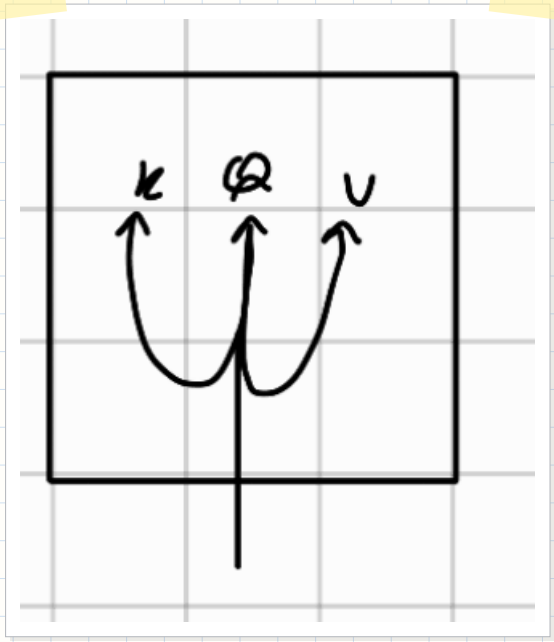
$W_Q \cdot w = q_1 q_2$
$W_K \cdot w = k_1 k_2$
$W_V \cdot w = v_1 v_2$



orijinal çizim, s. 9

Bunun adı self attention

Makalelerde böyle görürsünüz



orijinal çizim, s. 9

→ Multihead ile olabilir

Q: Ne aradığımız

k: Ne ile ilgili

V: Neyi kullanacağız

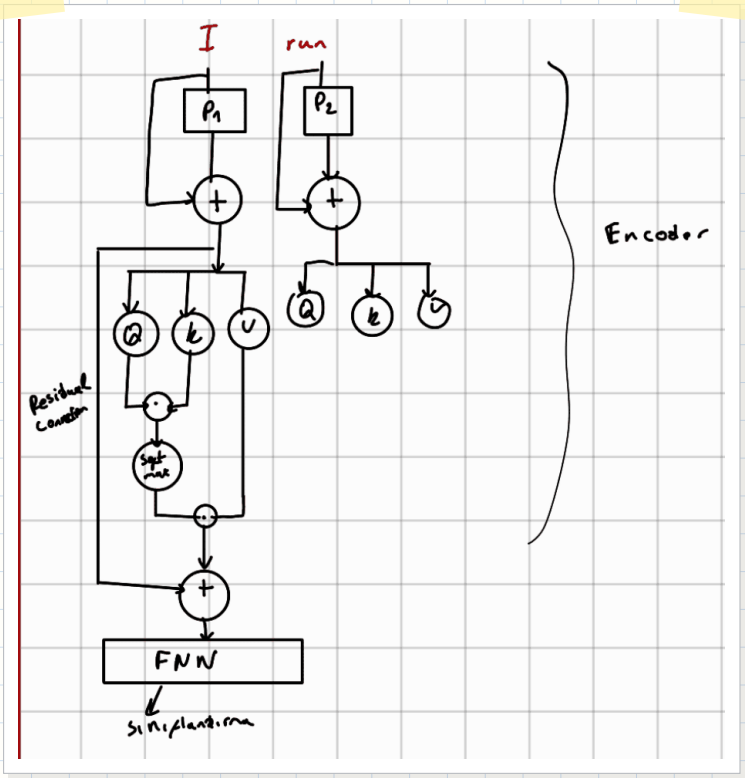


orijinal çizim, s. 9

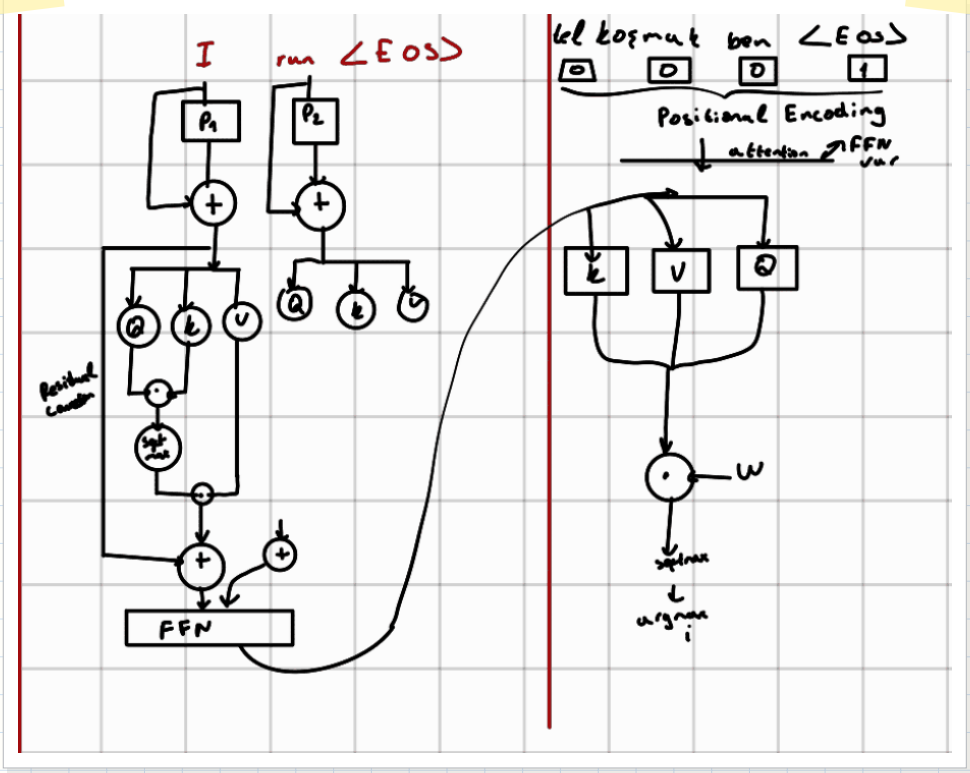
bunu × W_1 → positional encoding matris

$$x_I + W_1 x_I = \bar{x}_I$$

positional embedding



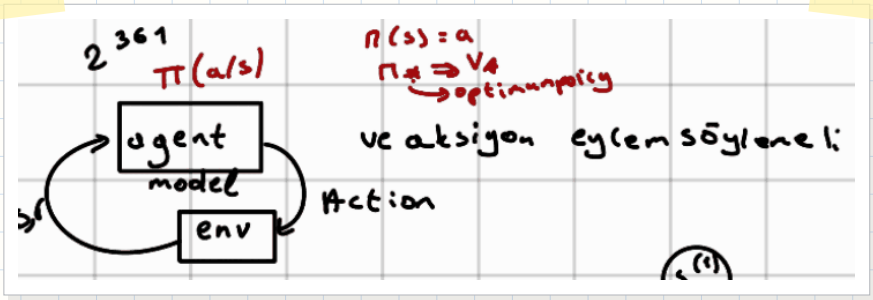
orijinal çizim, s. 9



orijinal çizim, s. 9

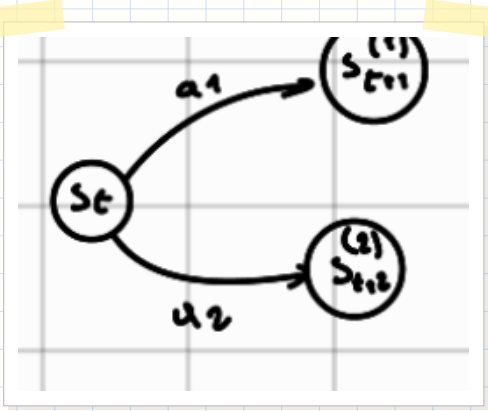
orj. s. 10

$$P(s_{t+1}, r_{t+1} \mid s_t, a_t)$$



orijinal çizim, s. 10

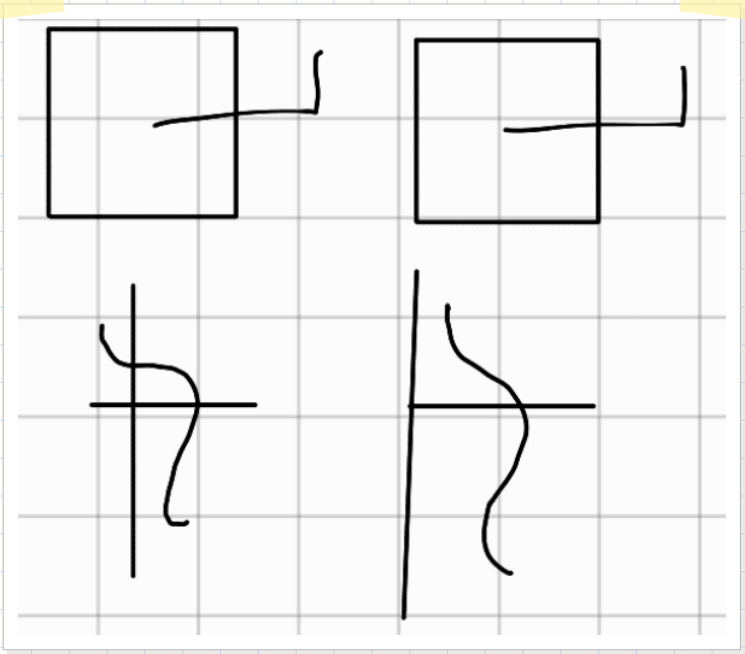
ve aksiyon ? seçmeyi öğrenmeli



orijinal çizim, s. 10

- Model tabanlı

Modelsiz



orijinal çizim, s. 10

Exploration

Exploitation

? Dilemması

$$V(s) = E\{G_t \mid S_t\}$$

$$G_t = \sum_{n=1} r_{t+n}$$

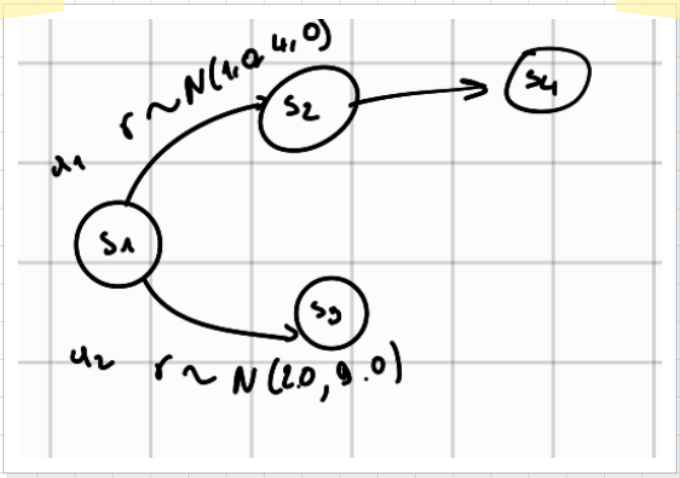
$V(s) = E\{G_t \mid S_t\}$ nasıl hesaplanır

$s_0, a_0, r_1, s_1, a_1, r_2, \cdots$

Gelecekte alacağım ödüllerin toplamı

$$V(s) = E\{G_t \mid S_t\}$$

$$Q(s,a) = E\{G_t \mid s_t, a\}$$



orijinal çizim, s. 10

(1) $\boxed{\begin{aligned} Q(s_1, a_1) &= 0 \\ Q(s_1, a_2) &= 0 \end{aligned}}$

SARSA $0 < \alpha < 1$

$$Q(s_1, a_1) = Q(s_1, a_1) + \alpha(r + V(s_2) - Q(s_1, a_1))$$

reward ↗ ? $Q(s_2, a_2)$

$$= 2$$

Policy gradient theorem

Actor critic (value, policy birleşimi)

Bunun dışında

Multi agent reinforcement learning

$$G_t = \sum_{n=1}^{T} \gamma^{n-1} r_{t+n} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3}$$

$0 < \gamma \leq 1$ $\qquad = r_{t+1} + G_{t+1}$

$$V(s_t) = r_{t+1} + V(s_{t+1})$$