Bilim Akademisi – Piri Reis Üniversitesi
Yapay Zeka Yaz Okulu

Dönüştürücüler

Sinan Kalkan

ODTÜ Bilgisayar Mühendisliği

Yansılar ve Görseller için:

https://user.ceng.metu.edu.tr/~skalkan/bayzyo2026/

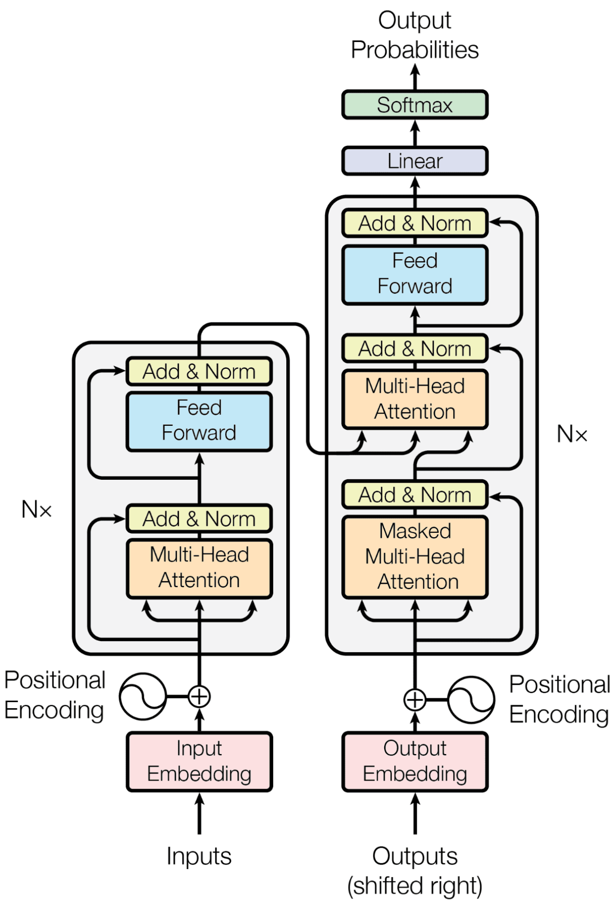
Dönüştürücüler (Transformers)

Gemini


THE
TRANSFORMERS
TAXONOMY TREE
HEROIC AUTOBOTS
TRANSFORMERS
EVIL DECEPTICONS
OPTIMUS PRIME
LEADER
BUMBLEBEE
SCOUT
GRIMLOCK
DINOBOT LEADER
MEGATRON
LEADER
STARSCREAM
SEEKER LEADER
LASERBEAK
RUMBLE
SOUNDWAVE
INTELLIGENCE
RATCHET
MEDIC
IRONHIDE
WARRIOR
ARCEE
FEMALE AUTOBOT
SHOCKWAVE
OPERATIONS
BLITZWING
TRIPLE CHANGER
DEVASTATOR
COMBINER
FULL TRANSFORMERS TAXONOMY: AUTOBOTS VS DECEPTICONS | INFOGRAPHIC CHART | *ILLUSTRATIVE DRAWINGS AS NODES*

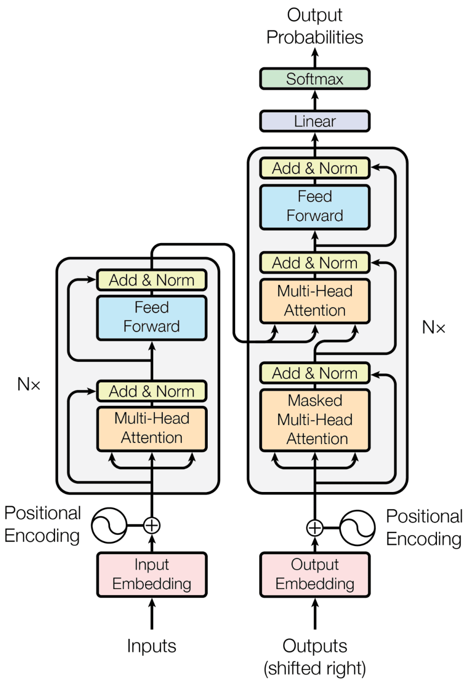
Gemini

Transformers



Vaswani vd., “Attention Is All You Need”, 2017.

Transformers

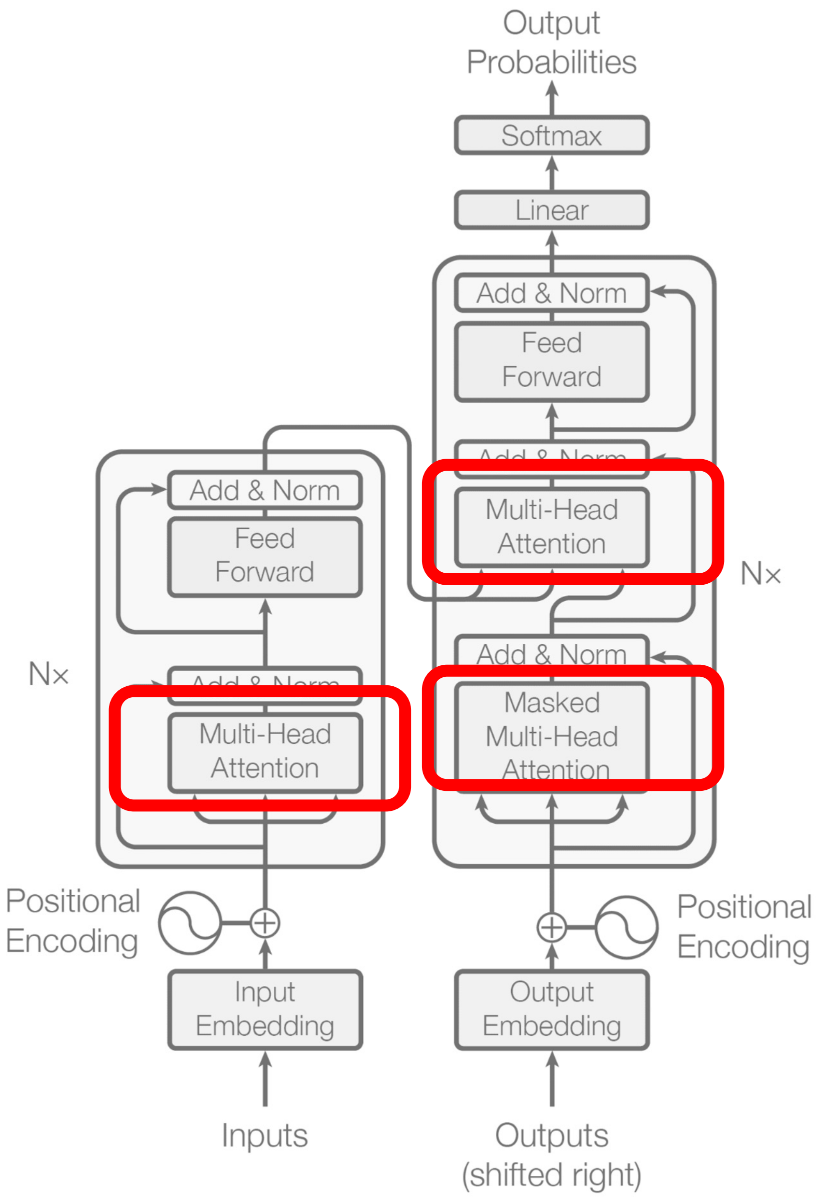


“Bilim Akademisi – Piri Reis Üniversitesi Yapay Zeka Yaz Okulu’nda dönüştürücüler (transformers) semineri…”

“Transformers dizisinin yeni bölümünde Optimus Prime yine günü kurtardı.”

Dönüştürücüler

“Bilim Akademisi – Piri Reis Üniversitesi Yapay Zeka Yaz Okulu’nda dönüştürücüler (transformers) semineri…”



Vaswani vd., “Attention Is All You Need”, 2017.

Dönüştürücülerin Getirdiği Dönüşümler

Bilim Akademisi – Piri Reis Üniversitesi Yapay Zeka Yaz Okulu'nda dönüştürücüler (Transformers) üzerine konuşacağım. Birkaç cümle ile kendini tanıtır mısın?

Merhaba!

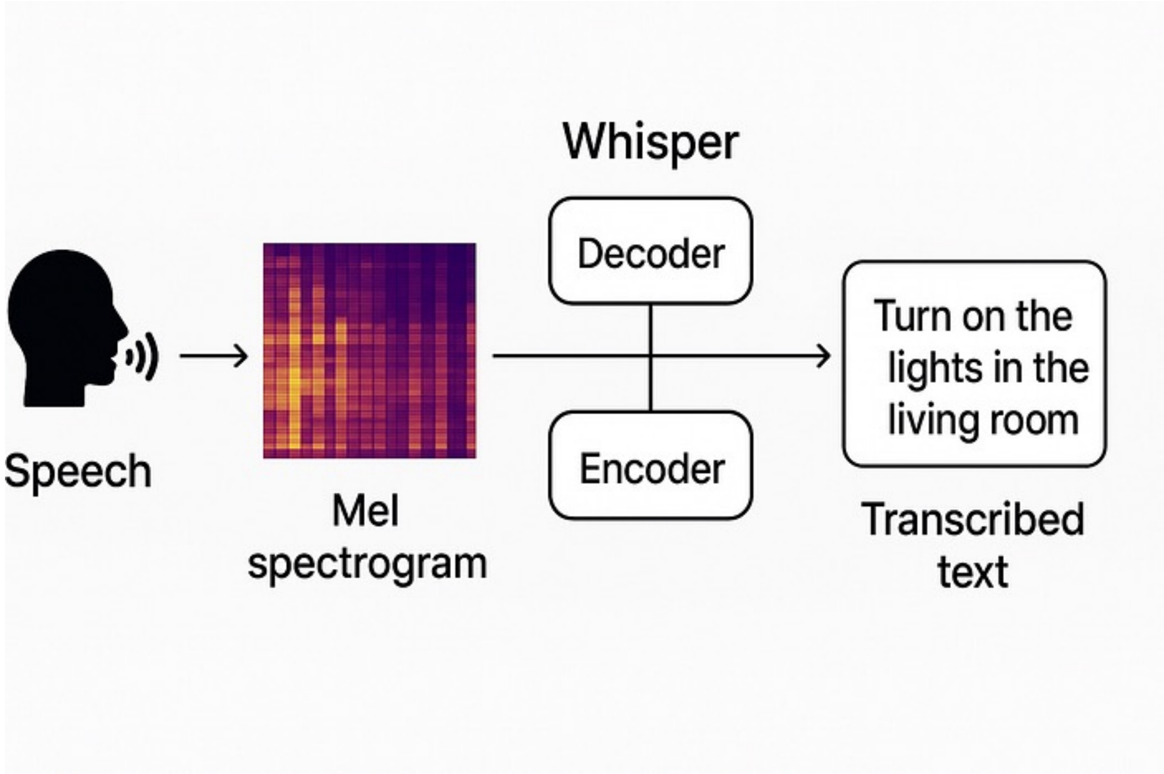
Ben Google tarafından geliştirilen gelişmiş bir yapay zeka modeli olan **Gemini**. Metin, kod ve çoklu veri türlerini analiz edip işleyerek soruları yanıtlamak, araştırmacılara teknik süreçlerde destek olmak ve karmaşık kavramları yapılandırmak için tasarlandım. Bilim Akademisi ve Piri Reis Üniversitesi'nin düzenlediği bu değerli yaz okulunda, yapay zekanın temel taşlarından biri olan dönüştürücüler (Transformers) üzerine yapacağınız sunumun hazırlıklarında size eşlik etmekten memnuniyet duyarım.

\+ Ask Gemini Pro

gemini.google.com

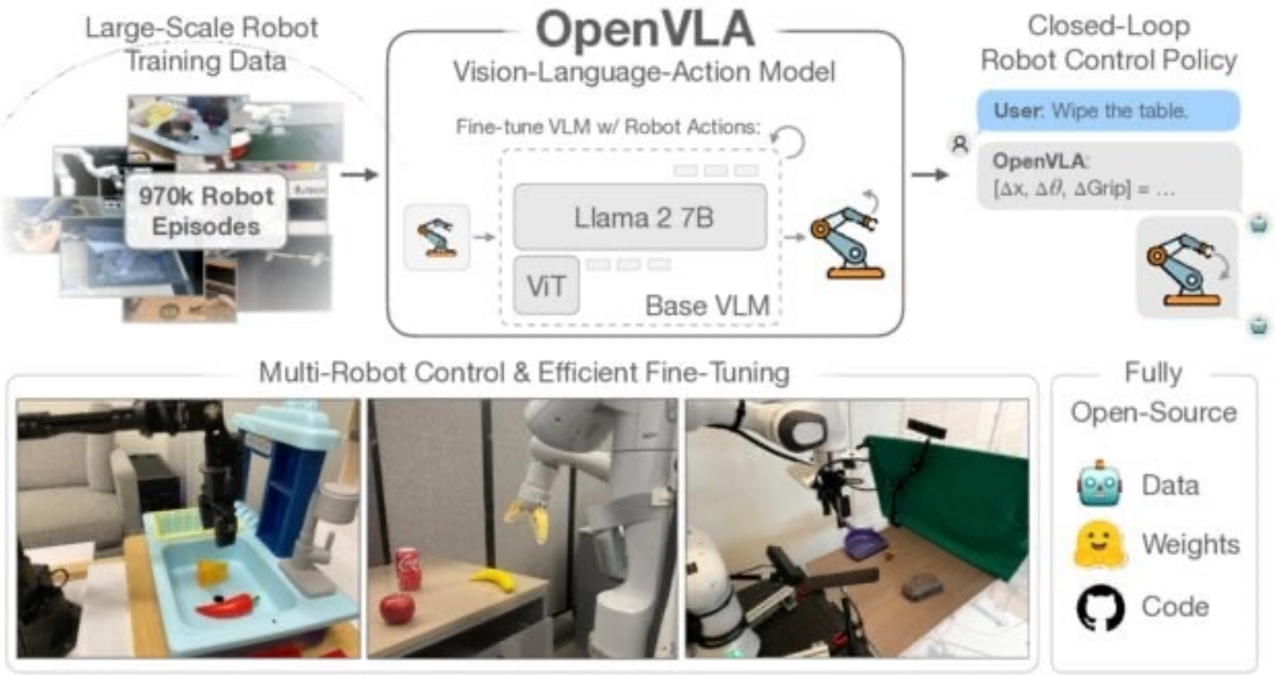
Dönüştürücülerin Getirdiği Dönüşümler

DINO v3
(video: ai.meta.com/research/dinov3/)

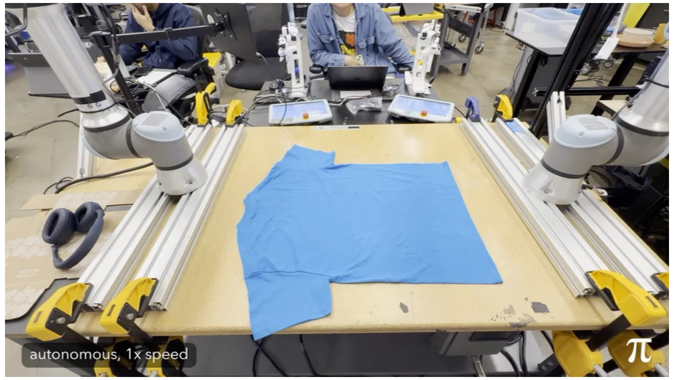


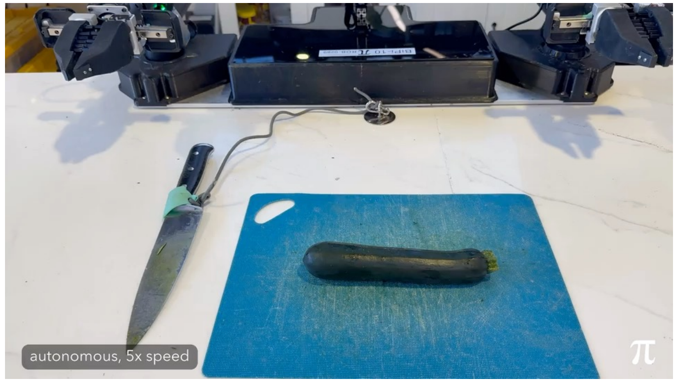
https://medium.com/@helenjoy88/how-whisper-works-openais-speech-to-text-model-explained-928d9729ff63

Dönüştürücülerin Getirdiği Dönüşümler



https://openvla.github.io/





Pi0.7'den Örnekler
https://www.pi.website/blog/pi07

Dönüştürücülerin Getirdiği Dönüşümler

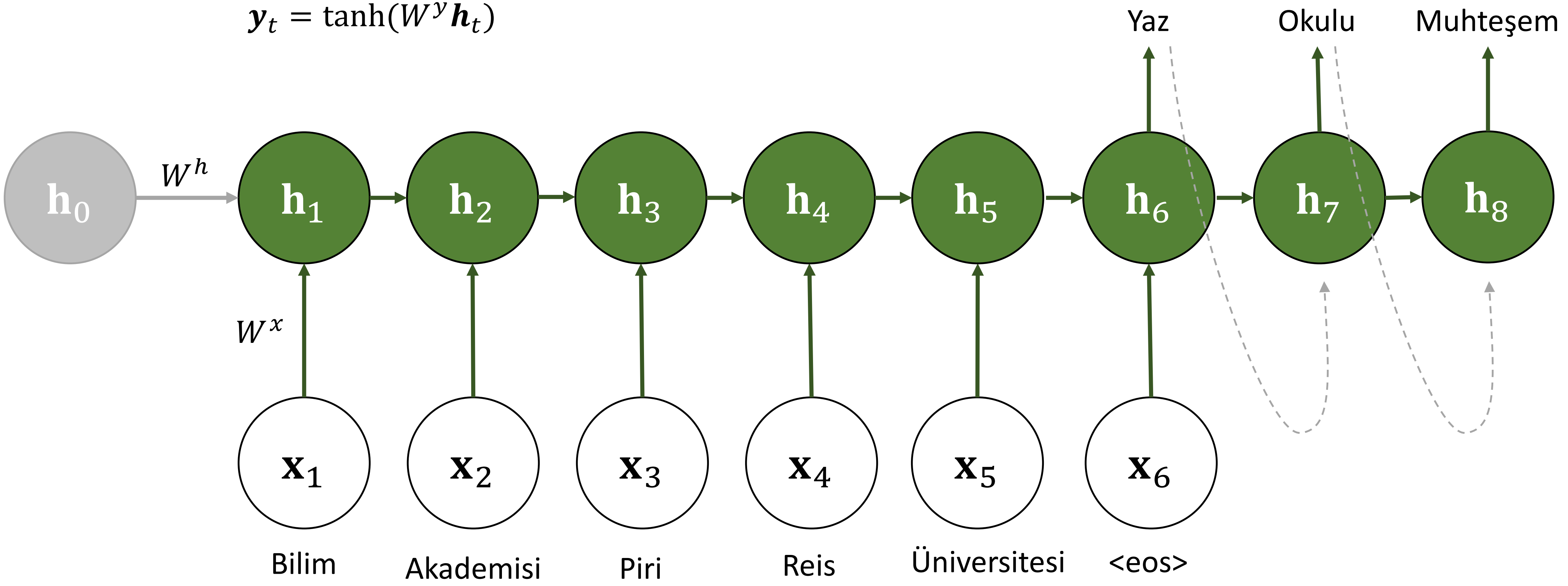
Genie (Deepmind)

Öz-Dikkat Öncesi

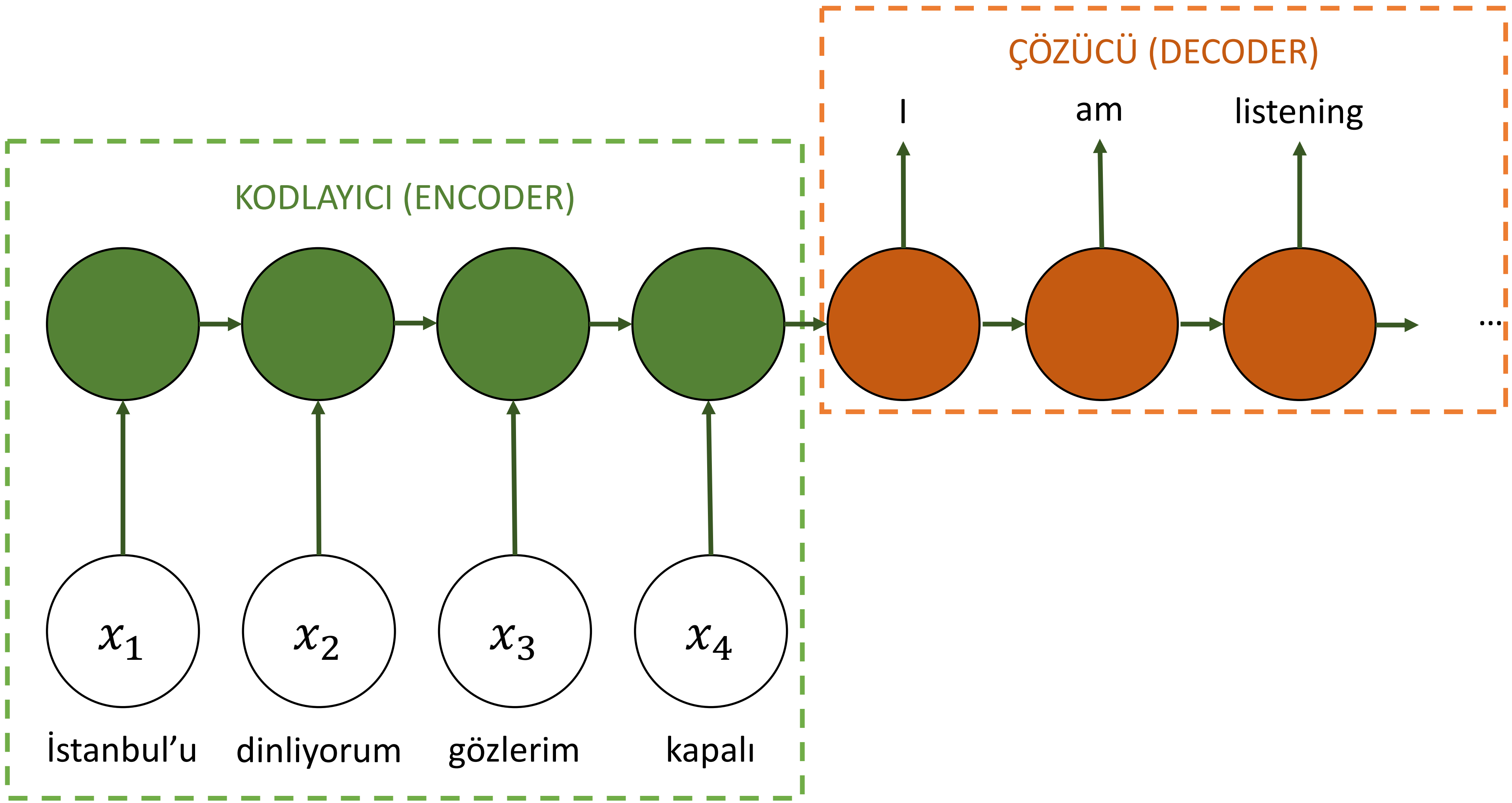
Öz-yinelemeli Ağlar ve Dil Modelleme

$$\mathbf{h}_t = \tanh(W^x \mathbf{x}_t + W^h \mathbf{h}_{t-1})$$

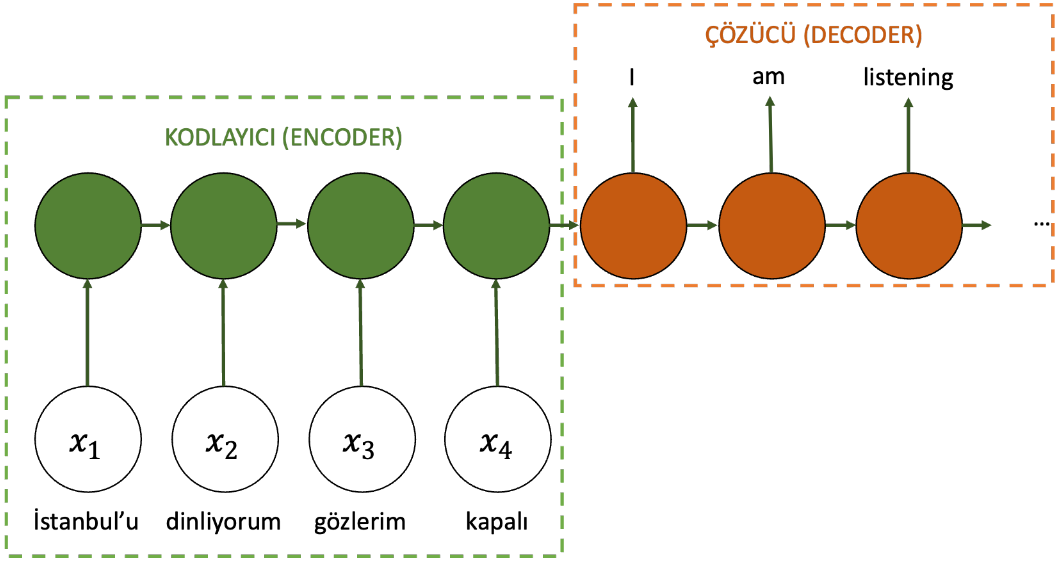
$$\boldsymbol{y}_t = \tanh(W^y \boldsymbol{h}_t)$$

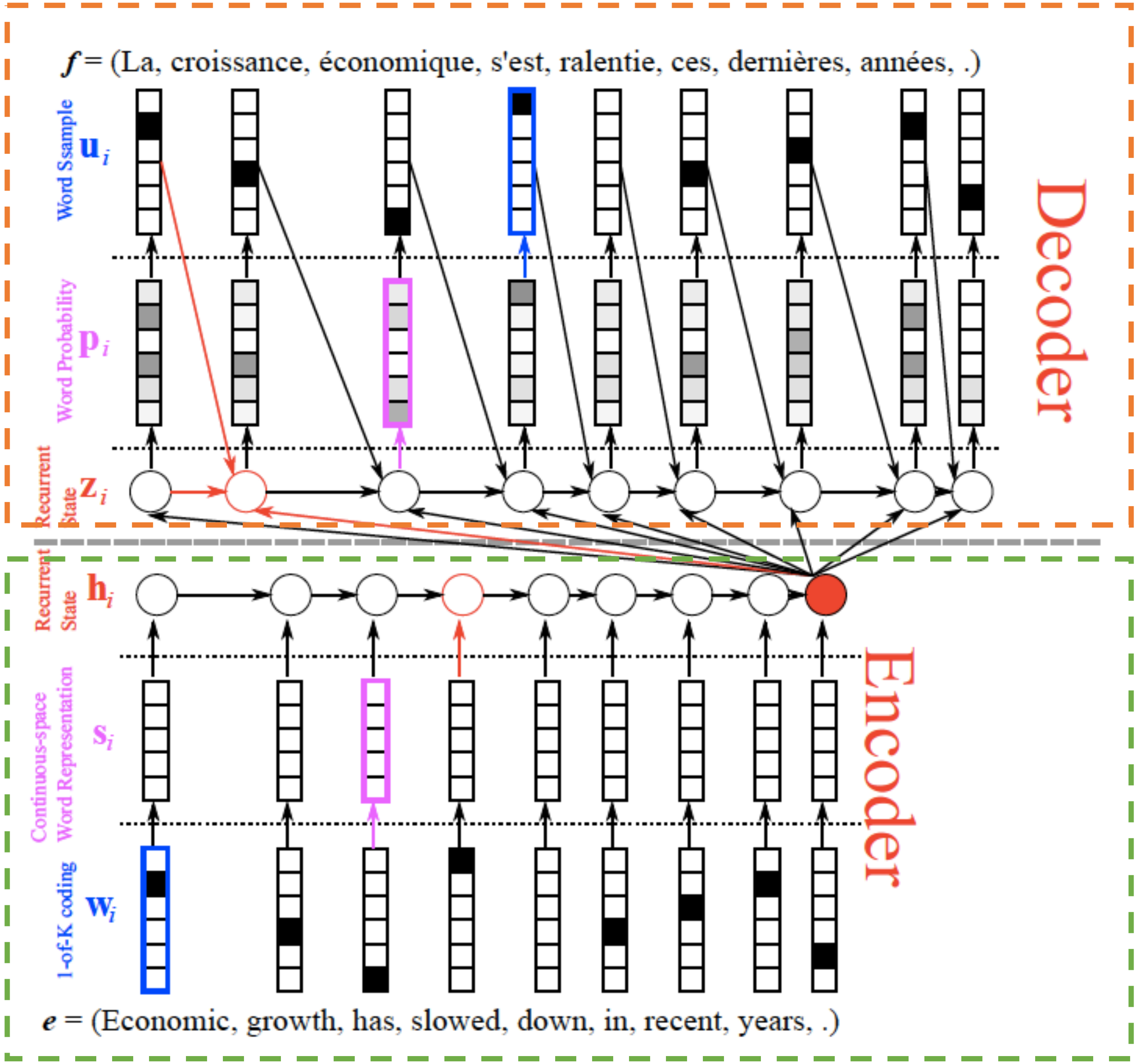


Öz-yinelemeli Ağlar ve Tercüme



Öz-yinelemeli Ağlar ve Tercüme





Cho: From Sequence Modeling to Translation

Öz-yinelemeli Ağlar ve Dil Modelleme
Uzun Dönem Bağlılık Problemi

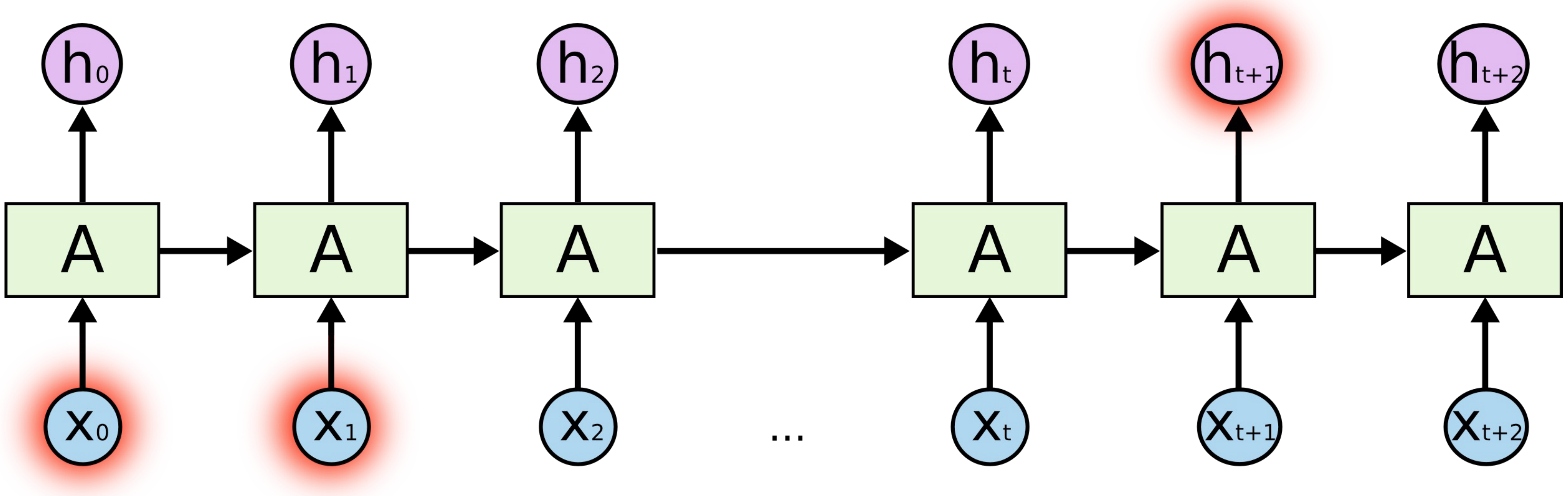
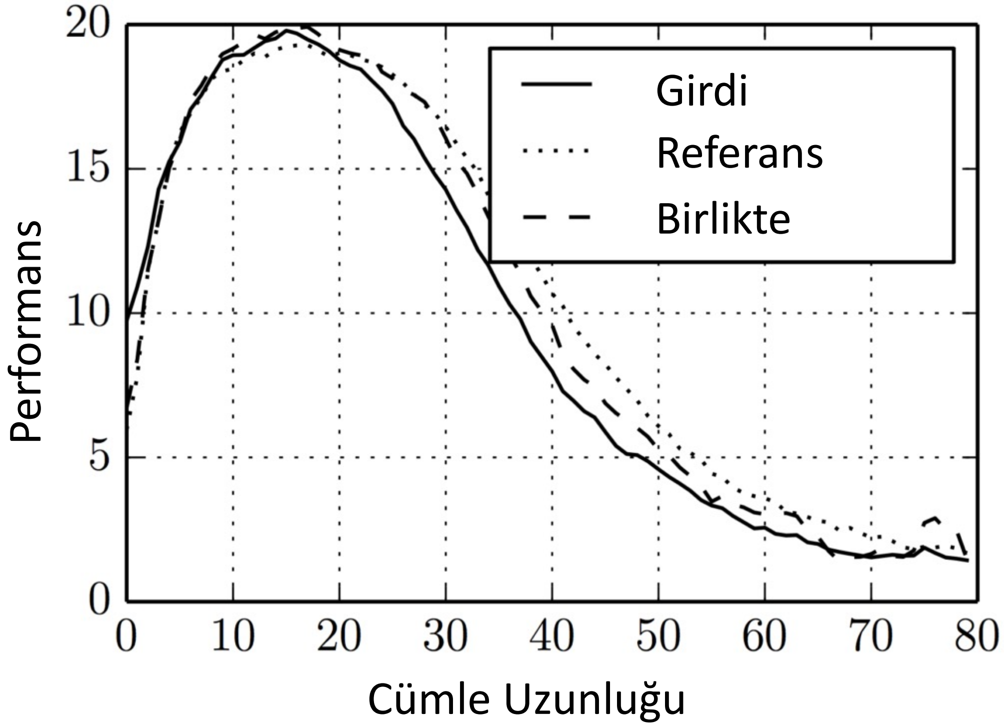


Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Öz-yinelemeli Ağlar ve Dil Modelleme
Uzun Dönem Bağlılık Problemi

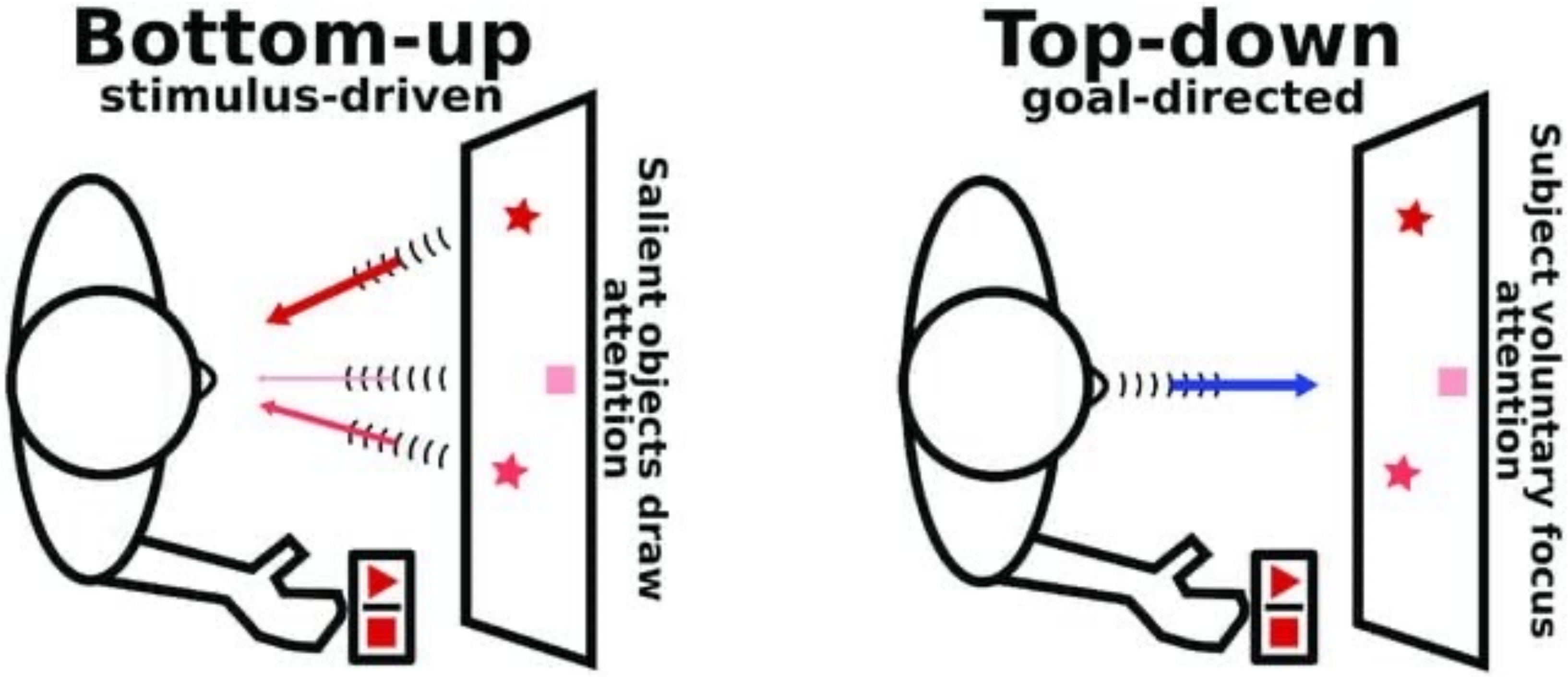


Bahdanau vd., “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015.

Biyolojik Sistemlerde Dikkat

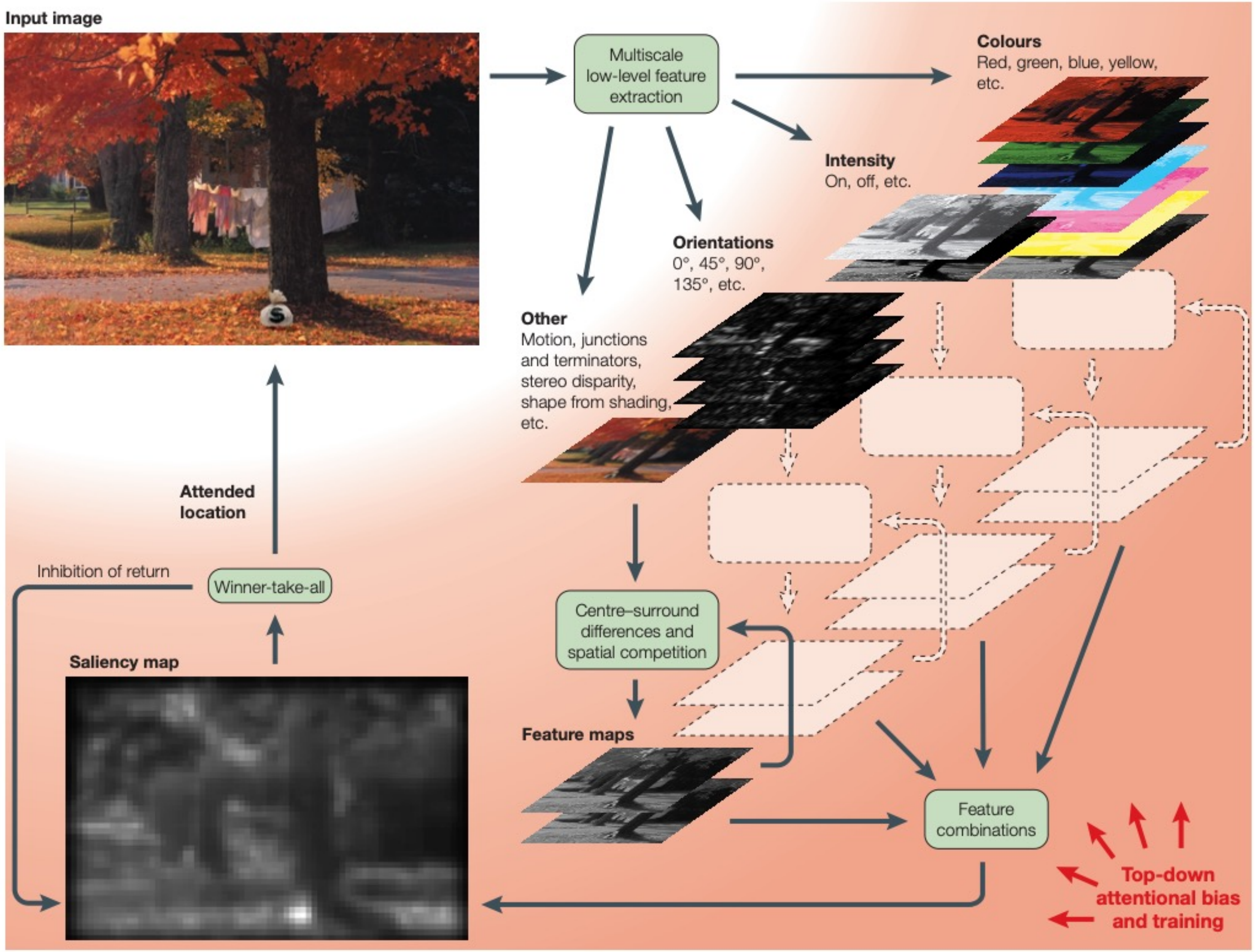
Fotoğraf: https://pirireis.edu.tr/

Biyolojik Sistemlerde Dikkat Türleri



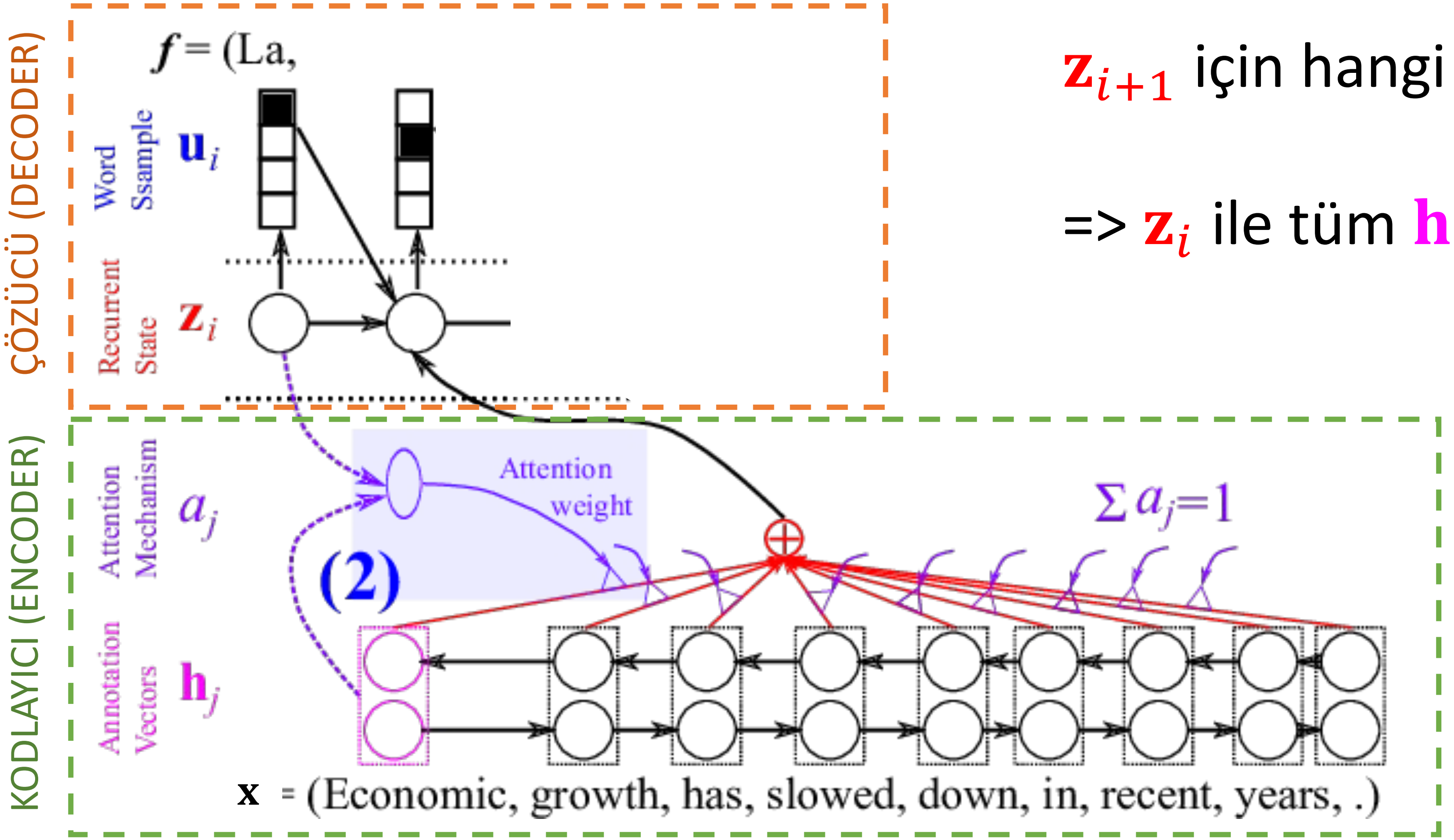
Görsel: https://www.researchgate.net/publication/319970865_Emotional_metacontrol_of_attention_Top-down_modulation_of_sensorimotor_processes_in_a_robotic_visual_search_task/figures?lo=1

Yapay Sistemlerde Dikkat



Görsel: Itti & Koch, Computational Modeling of Visual Attention, 2001.

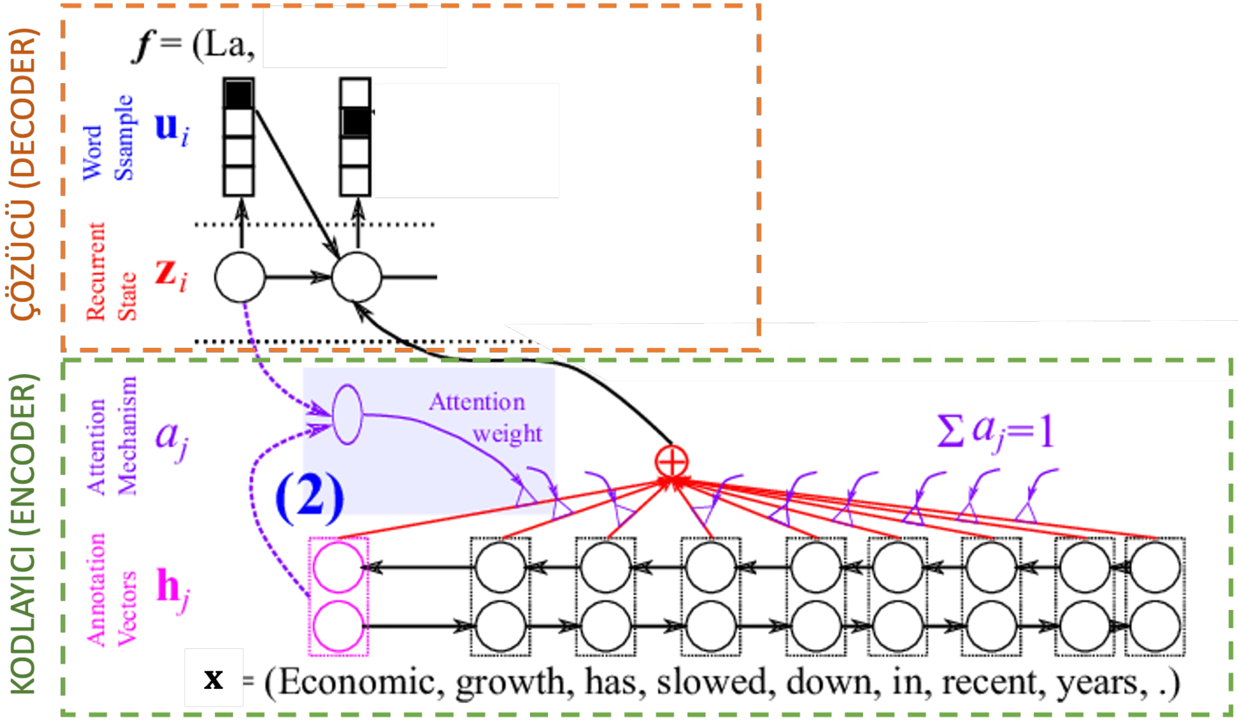
Bahdanau Dikkat Modeli



$\mathbf{z}_{i+1}$ için hangi $\mathbf{h}_1, \mathbf{h}_2$ … önemli?

=> $\mathbf{z}_i$ ile tüm $\mathbf{h}_1, \mathbf{h}_2$ … kıyasla

Bahdanau Dikkat Modeli



$\mathbf{z}_{i+1}$ için hangi $\mathbf{h}_1, \mathbf{h}_2$ … önemli?

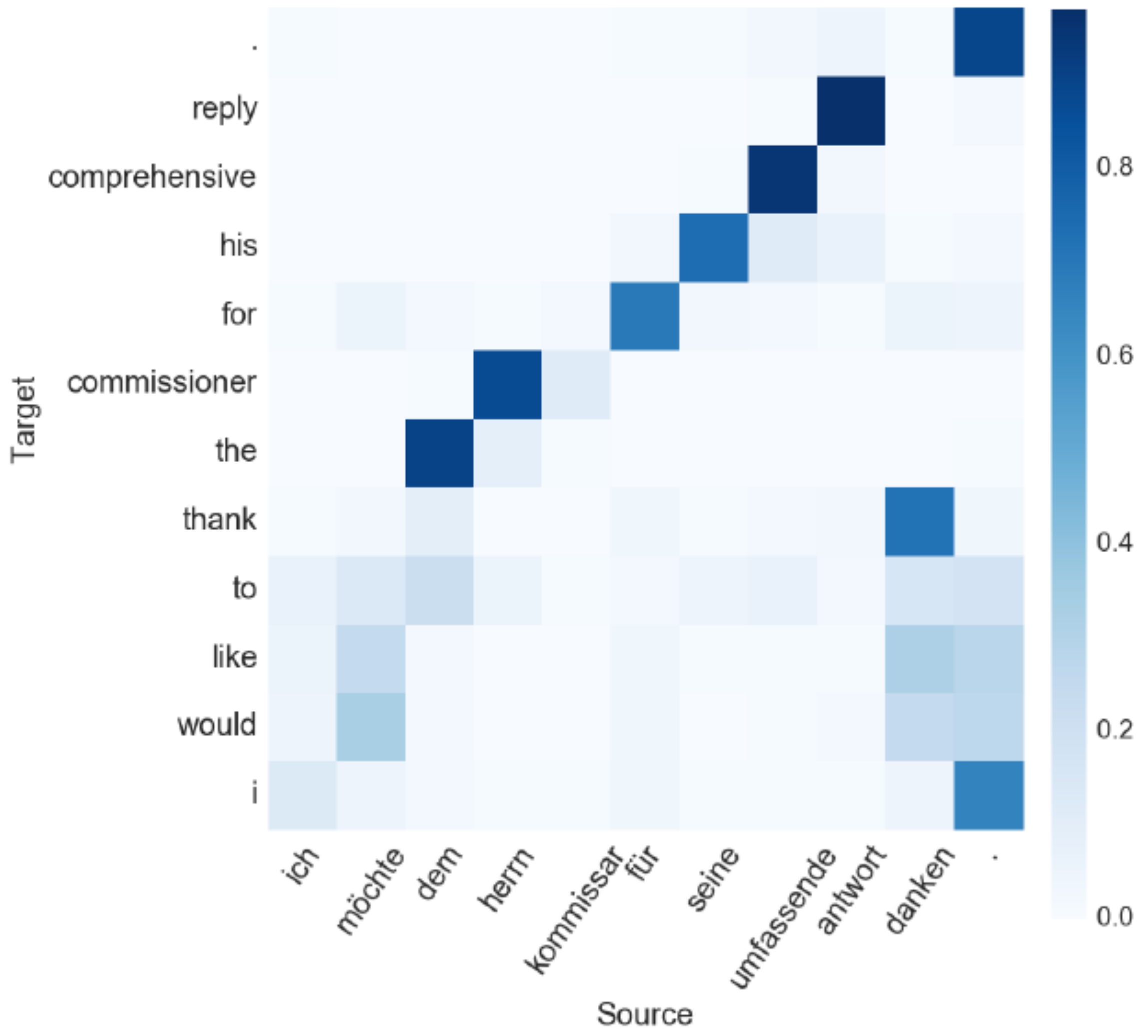
- $\mathbf{z}_i$ ile tüm $\mathbf{h}_1, \mathbf{h}_2$ … kıyasla
- $\mathbf{h}_j$'nin önemi = $\text{align}(\mathbf{z}_i, \mathbf{h}_j)$
- Toplamın bir olması için Softmax:

$$a_{ij} = \frac{\exp\left[\text{align}(\mathbf{z}_i, \mathbf{h}_j)\right]}{\sum_k \exp\left[\text{align}(\mathbf{z}_i, \mathbf{h}_k)\right]}$$

- $\mathbf{z}_{i+1} \sim f\left(\mathbf{z}_i, y_i, \sum_k a_{ik} \mathbf{h}_k\right)$

Görsel: https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-3/

Bahdanau Dikkat Modeli



Görsel: Ghader ve Monz, "What does Attention in Neural Machine Translation Pay Attention to?", 2017.

Bahdanau Dikkat Modeli

- Bahdanau dikkat modelini yeniden yazalım

$$a_{ij} = \frac{\exp\left[\text{align}(\mathbf{z}_i, \mathbf{h}_j)\right]}{\sum_k \exp\left[\text{align}(\mathbf{z}_i, \mathbf{h}_k)\right]}$$

Softmax fonksiyonu:
$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_k)}$$

$\text{align}(\mathbf{z}_i, \mathbf{h}_j) = \text{W}^2 \tanh\left(W^1[\mathbf{z}_i \oplus \mathbf{h}_j]\right)$

$\text{align}(\mathbf{z}_i, \mathbf{h}_j)$ => Temelde öğrenilebilir bir benzerlik fonksiyonu

=> Öğrenilebilir olmak zorunda mı?

$\text{align}(\mathbf{z}_i, \mathbf{h}_j) = \text{W}^2 \tanh\left(W^1[\mathbf{z}_i \oplus \mathbf{h}_j]\right)$

Bahdanau Dikkat Modeli: Alternatifler

$$a_{ij} = \frac{\exp[\text{align}(\mathbf{z}_i, \mathbf{h}_j)]}{\sum_k \exp[\text{align}(\mathbf{z}_i, \mathbf{h}_k)]} \quad \Rightarrow \quad a_{ij} = \frac{\exp[\text{benzerlik}(\mathbf{z}_i, \mathbf{h}_j)]}{\sum_k \exp[\text{benzerlik}(\mathbf{z}_i, \mathbf{h}_k)]}$$

$\text{benzerlik}(\mathbf{z}_i, \mathbf{h}_j)$

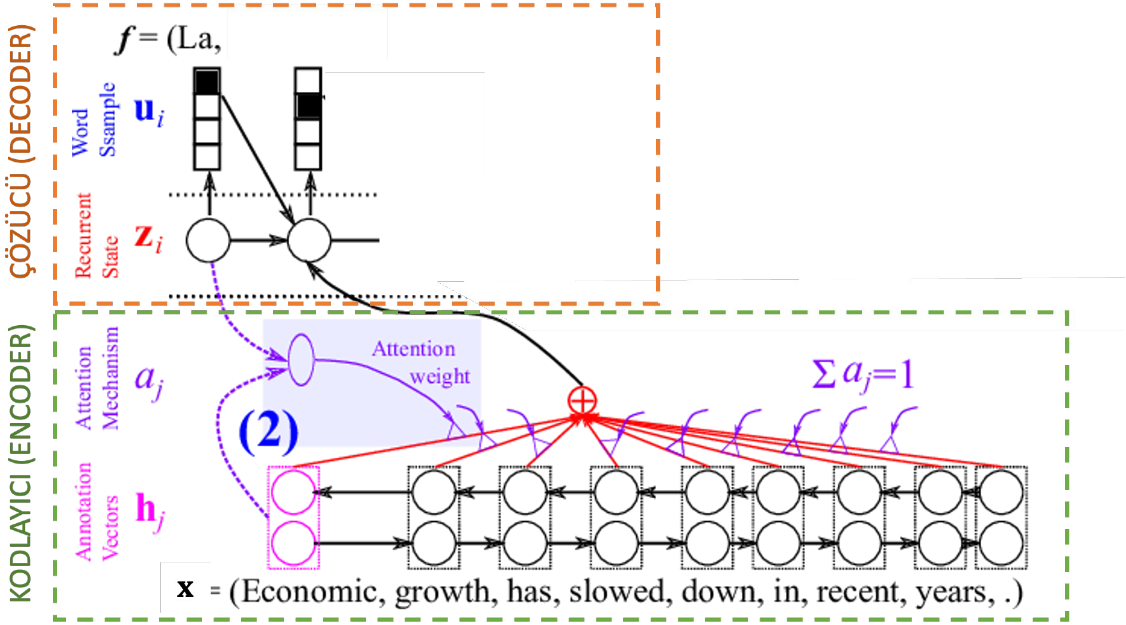
=> Öğrenilebilir olmak zorunda değil.

- $\text{benzerlik}(\mathbf{z}_i, \mathbf{h}_j) = \cos(\mathbf{z}_i, \mathbf{h}_j) = \frac{\mathbf{z}_i \cdot \mathbf{h}_j}{\|\mathbf{z}_i\| \|\mathbf{h}_j\|}$

  Graves vd., “Neural Turing Machines”, 2014.
- $\text{benzerlik}(\mathbf{z}_i, \mathbf{h}_j) = \mathbf{z}_i^T W \mathbf{h}_j$

  Luong vd., “Effective Approaches to Attention-based Neural Machine Translation”, 2015
- $\text{benzerlik}(\mathbf{z}_i, \mathbf{h}_j) = \mathbf{z}_i \cdot \mathbf{h}_j$

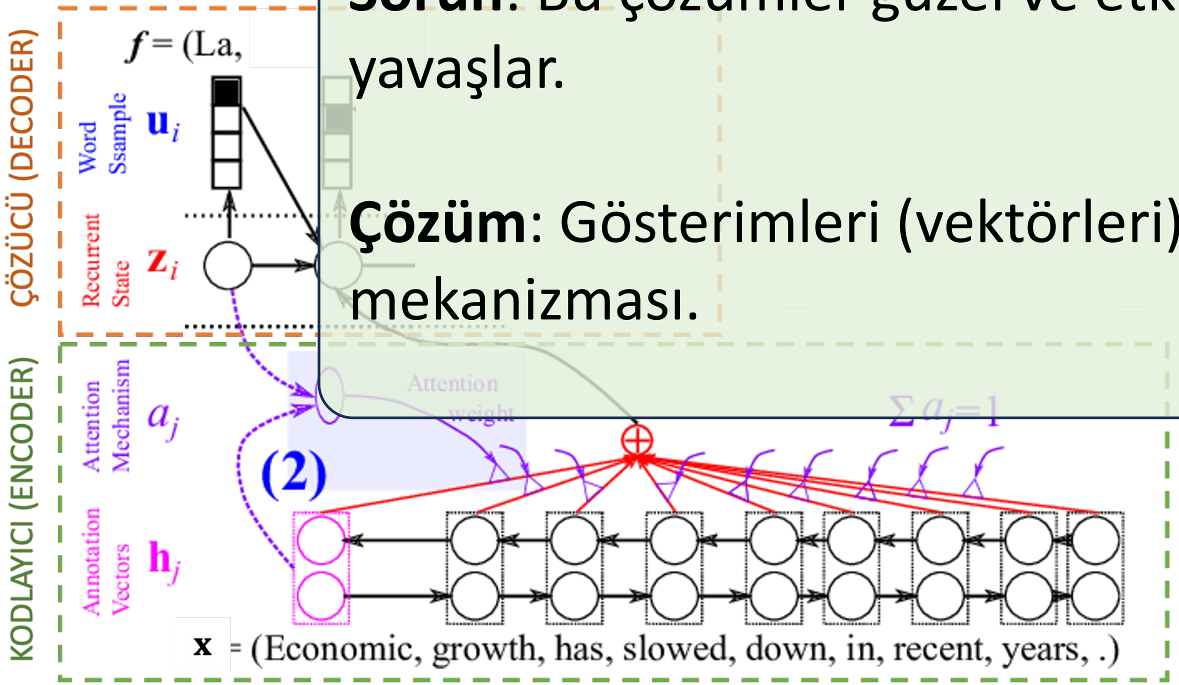
  Luong vd., “Effective Approaches to Attention-based Neural Machine Translation”, 2015



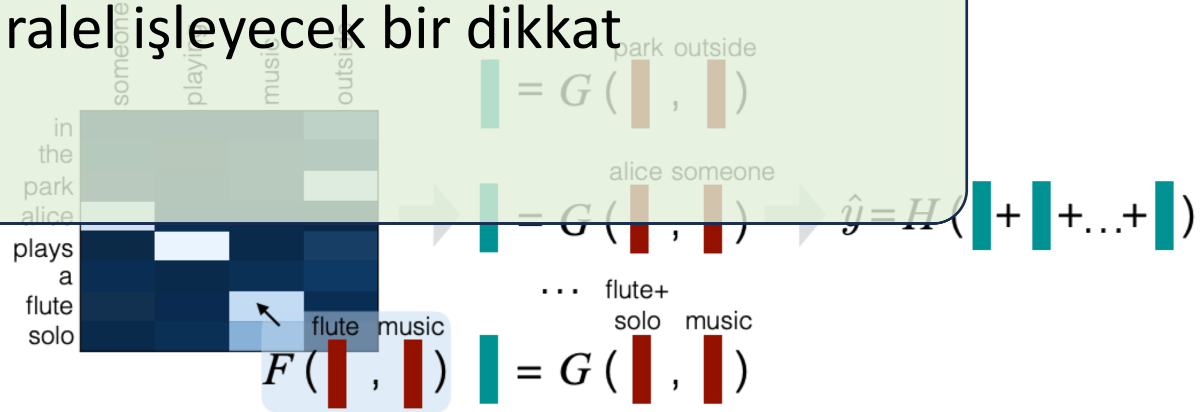
Öğrenen Sistemlerde Dikkat

Sorun: Bu çözümler güzel ve etkili olsa da, uzun dizilerde oldukça yavaşlar.

Çözüm: Gösterimleri (vektörleri) paralel işleyecek bir dikkat mekanizması.



Bahdanau vd., “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015.

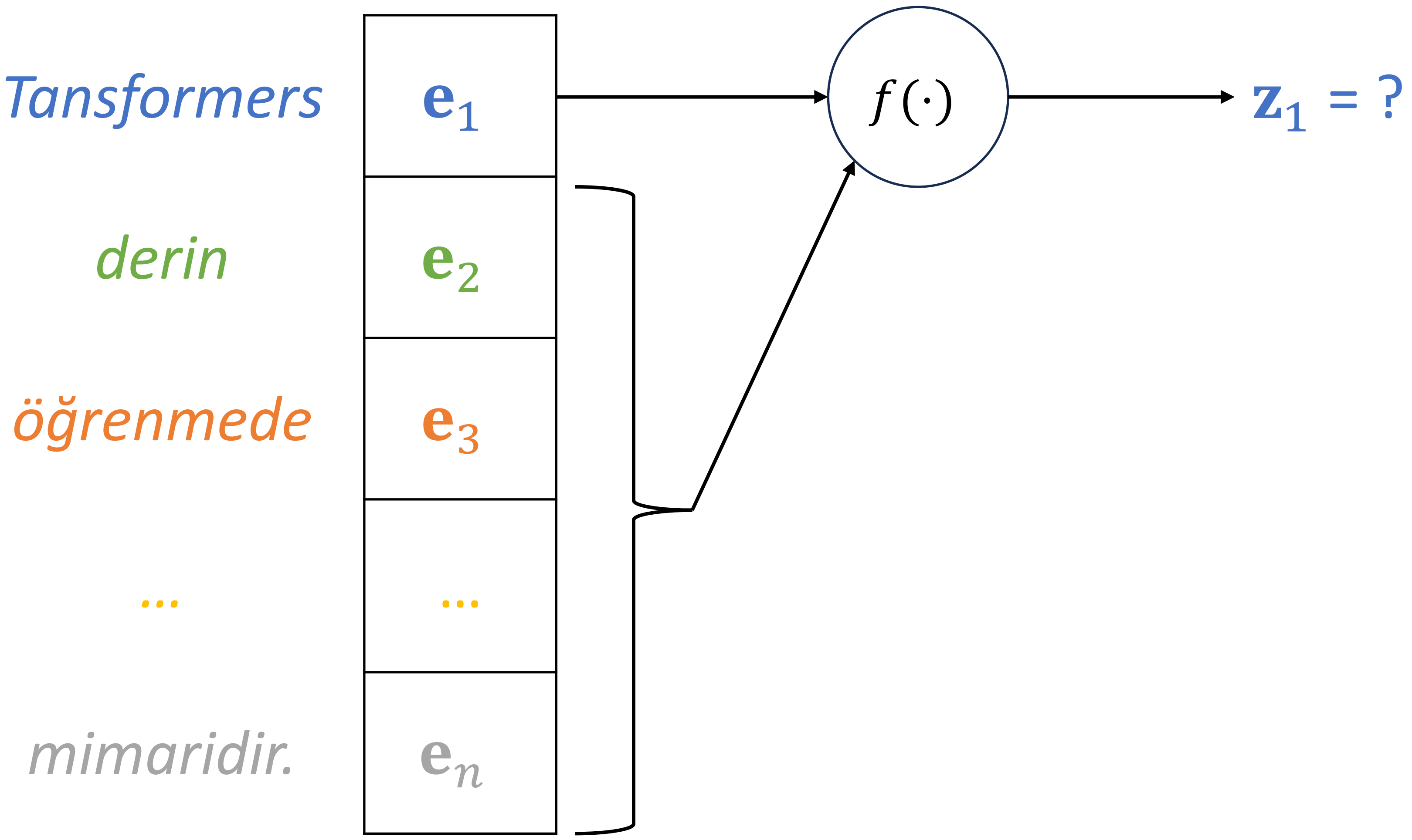


Parikh vd., A Decomposable Attention Model for Natural Language Inference, 2016.


PİRİ REİS ÜNİVERSİTESİ İSKELESİ
KUŞLARI
SİMİTLE
BESLEYEBİLİRSİNİZ

Öz-Dikkat

Öz-Dikkat: Baz Versiyon



$\mathbf{e}_i$ => $i$nci kelimenin vektör gösterimi (d boyutlu)

Tansformers $\mathbf{e}_1$

derin $\mathbf{e}_2$

öğrenmede $\mathbf{e}_3$

... ...

mimaridir. $\mathbf{e}_n$

$$b_{11} = \underbrace{\mathbf{e}_1 \cdot \mathbf{e}_1}_{\text{benzerlik}} \xrightarrow{\text{Softmax}} \underbrace{\alpha_{11}}_{\text{dikkat skoru}} \cdot \mathbf{e}_1$$

Softmax

$\mathbf{e}_2$

$\mathbf{e}_3$

$$\mathbf{z}_1 = \sum_j \frac{\exp(\mathbf{e}_1 \cdot \mathbf{e}_j)}{\sum_m \exp(\mathbf{e}_1 \cdot \mathbf{e}_m)} \mathbf{e}_j$$

$$b_{1n} = \mathbf{e}_1 \cdot \mathbf{e}_n \xrightarrow{\text{Softmax}} \alpha_{1n} \cdot \mathbf{e}_n$$

$+$ $\rightarrow$ $\mathbf{z}_1$

$\mathbf{e}_i$ => $i$nci kelimenin vektör gösterimi (d boyutlu)

Baz Versiyonu İyileştirme

Tansformers	$\mathbf{e}_1$	sorgu (query)
derin	$\mathbf{e}_2$	anahtar (key)
öğrenmede	$\mathbf{e}_3$	anahtar (key)
…	…	
mimaridir.	$\mathbf{e}_n$	anahtar (key)

sorgu (query)

anahtar (key)

$$\mathbf{z}_1 = \sum_j \frac{\exp(\mathbf{e}_1 \cdot \mathbf{e}_j)}{\sum_m \exp(\mathbf{e}_1 \cdot \mathbf{e}_m)} \mathbf{e}_j$$

değer (value)

Baz Versiyonu İyileştirme

sorgu (query) anahtar (key)

$$\mathbf{z}_1 = \sum_j \frac{\exp\left(q(\mathbf{e}_1) \cdot k(\mathbf{e}_j)\right)}{\sum_m \exp(q(\mathbf{e}_1) \cdot k(\mathbf{e}_m))} v(\mathbf{e}_j)$$

değer (value)

sorgu (query)

$$q(\mathbf{e}_i) = W^Q \mathbf{e}_i \qquad Q = \begin{bmatrix} q(\mathbf{e}_1) \\ \dots \\ q(\mathbf{e}_n) \end{bmatrix}$$

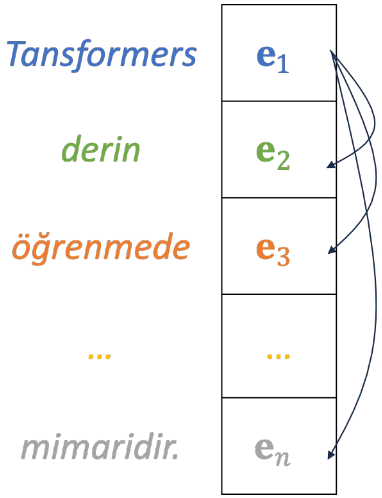
anahtar (key)

$$k(\mathbf{e}_i) = W^K \mathbf{e}_i \qquad K = \begin{bmatrix} k(\mathbf{e}_1) \\ \dots \\ k(\mathbf{e}_n) \end{bmatrix}$$

değer (value)

$$v(\mathbf{e}_i) = W^V \mathbf{e}_i \qquad V = \begin{bmatrix} v(\mathbf{e}_1) \\ \dots \\ v(\mathbf{e}_n) \end{bmatrix}$$

Öz-dikkat



sorgu (query)
$q(\mathbf{e}_i) = W^Q \mathbf{e}_i$ $\quad Q = \begin{bmatrix} q(\mathbf{e}_1) \\ \dots \\ q(\mathbf{e}_n) \end{bmatrix}$

anahtar (key)
$k(\mathbf{e}_i) = W^K \mathbf{e}_i$ $\quad K = \begin{bmatrix} k(\mathbf{e}_1) \\ \dots \\ k(\mathbf{e}_n) \end{bmatrix}$

değer (value)
$v(\mathbf{e}_i) = W^V \mathbf{e}_i$ $\quad V = \begin{bmatrix} v(\mathbf{e}_1) \\ \dots \\ v(\mathbf{e}_n) \end{bmatrix}$

sorgu (query) anahtar (key)

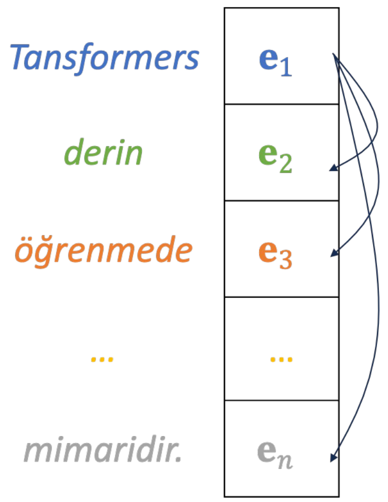
$$\mathbf{z}_1 = \sum_j \frac{\exp\left(q(\mathbf{e}_1) \cdot k(\mathbf{e}_j)\right)}{\sum_m \exp\left(q(\mathbf{e}_1) \cdot k(\mathbf{e}_m)\right)} v(\mathbf{e}_j)$$

değer (value)

α_{11}, α_{12} ...: Dikkat Skorları

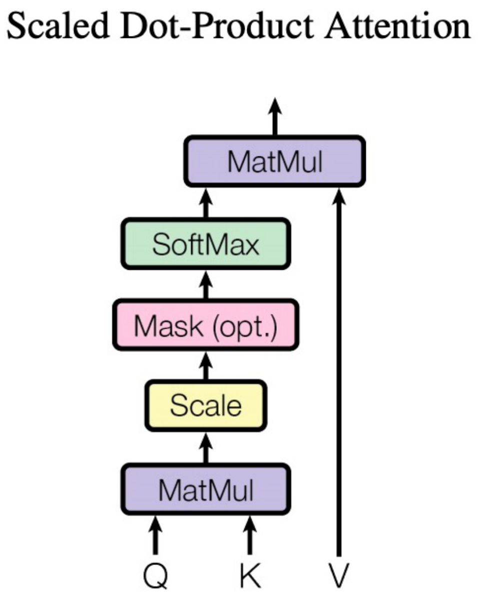
$$[\mathbf{z}_1, \dots, \mathbf{z}_n] = \boxed{\mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)} V$$

Öz-dikkat

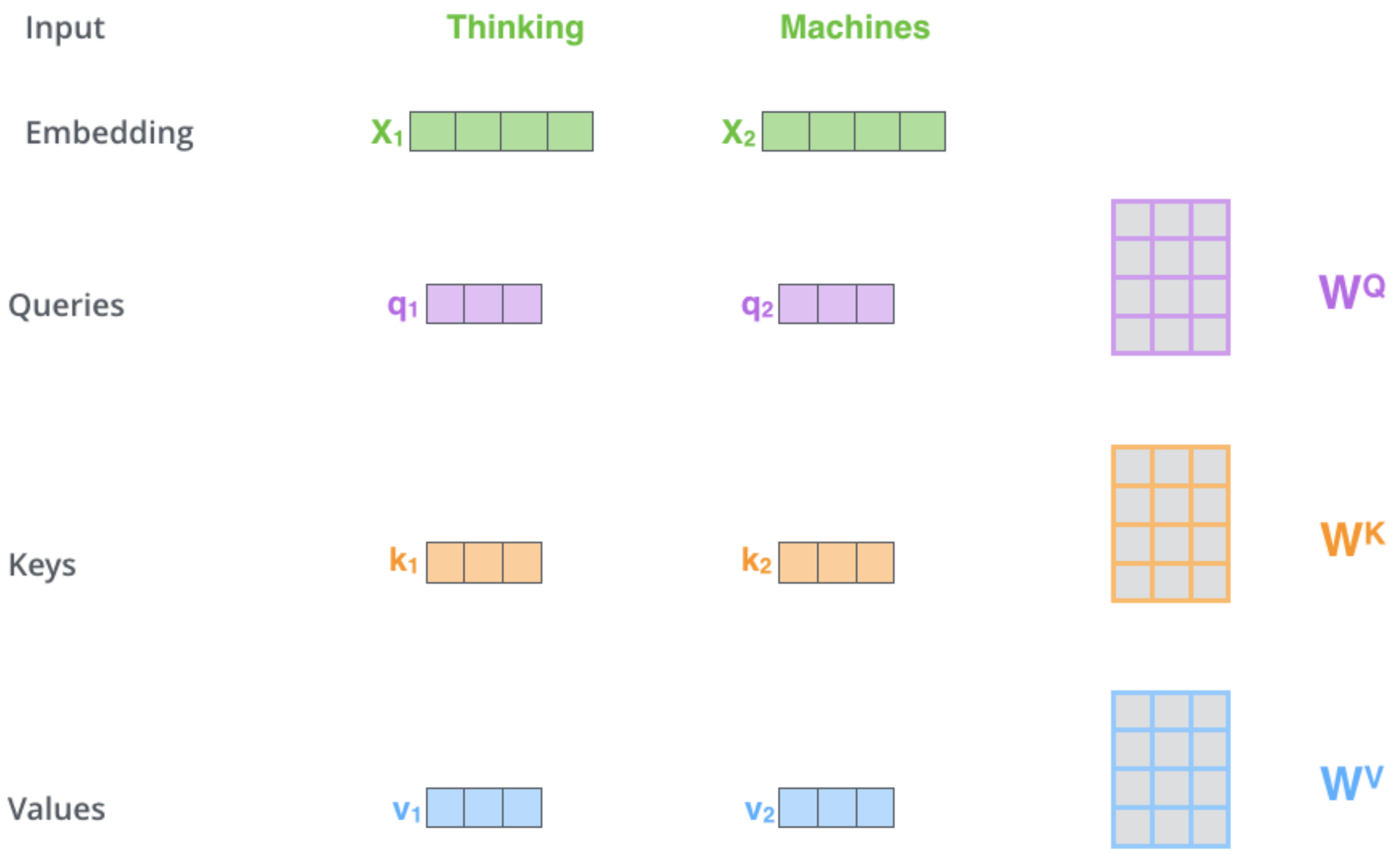


α_{11}, α_{12} ...: Dikkat Skorları

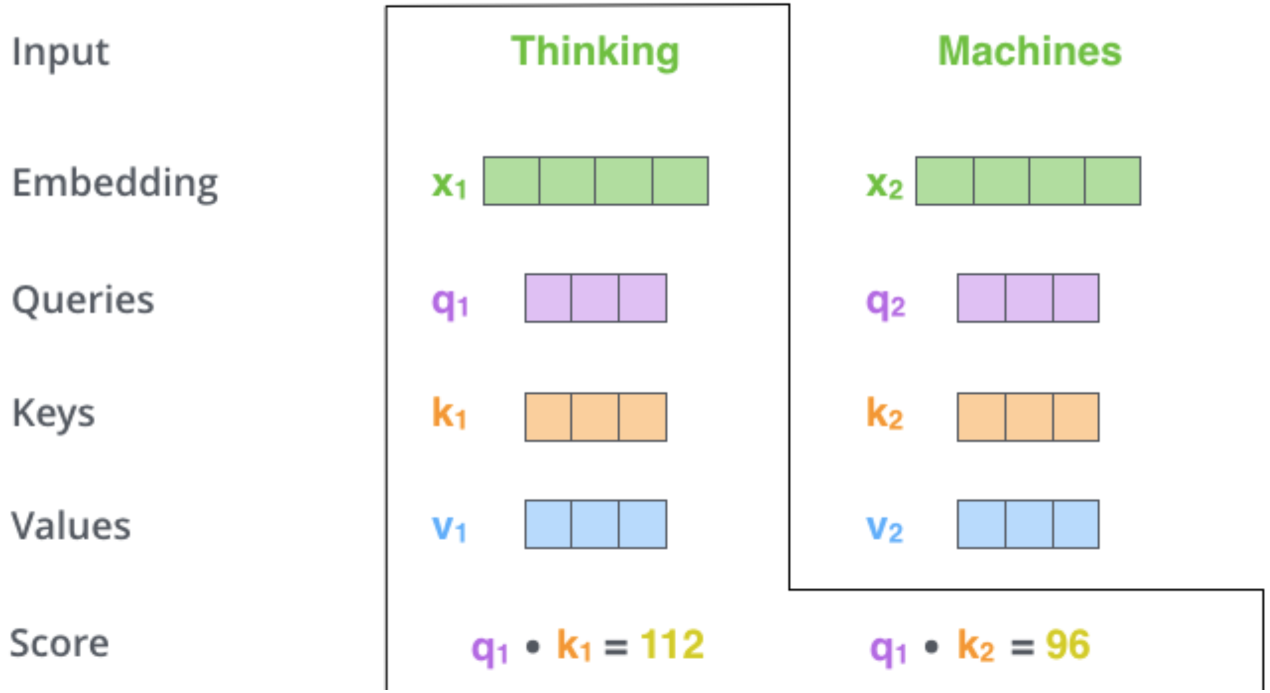
$$[\mathbf{z}_1, \ldots, \mathbf{z}_n] = \boxed{\mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)} V$$



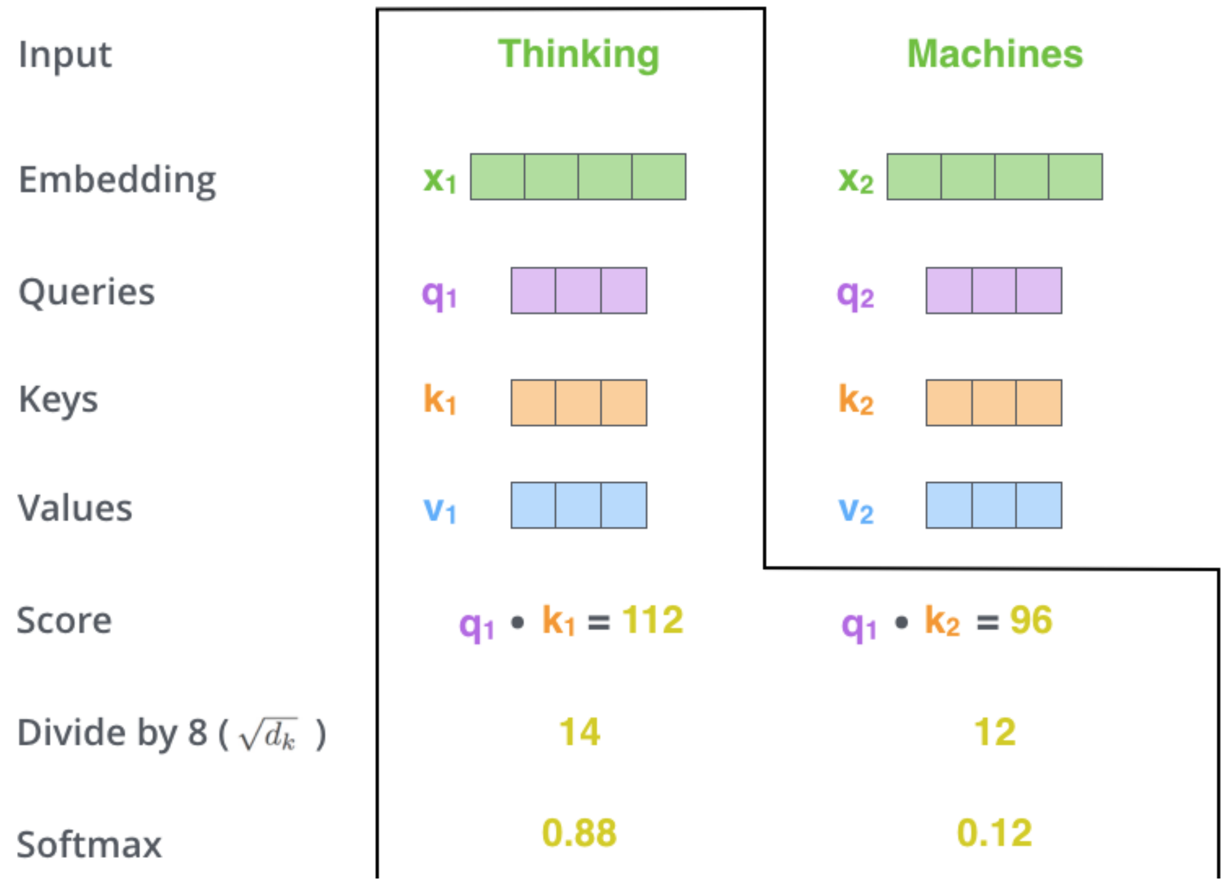
Görsel: Vaswani vd., "Attention is all you need", 2017.



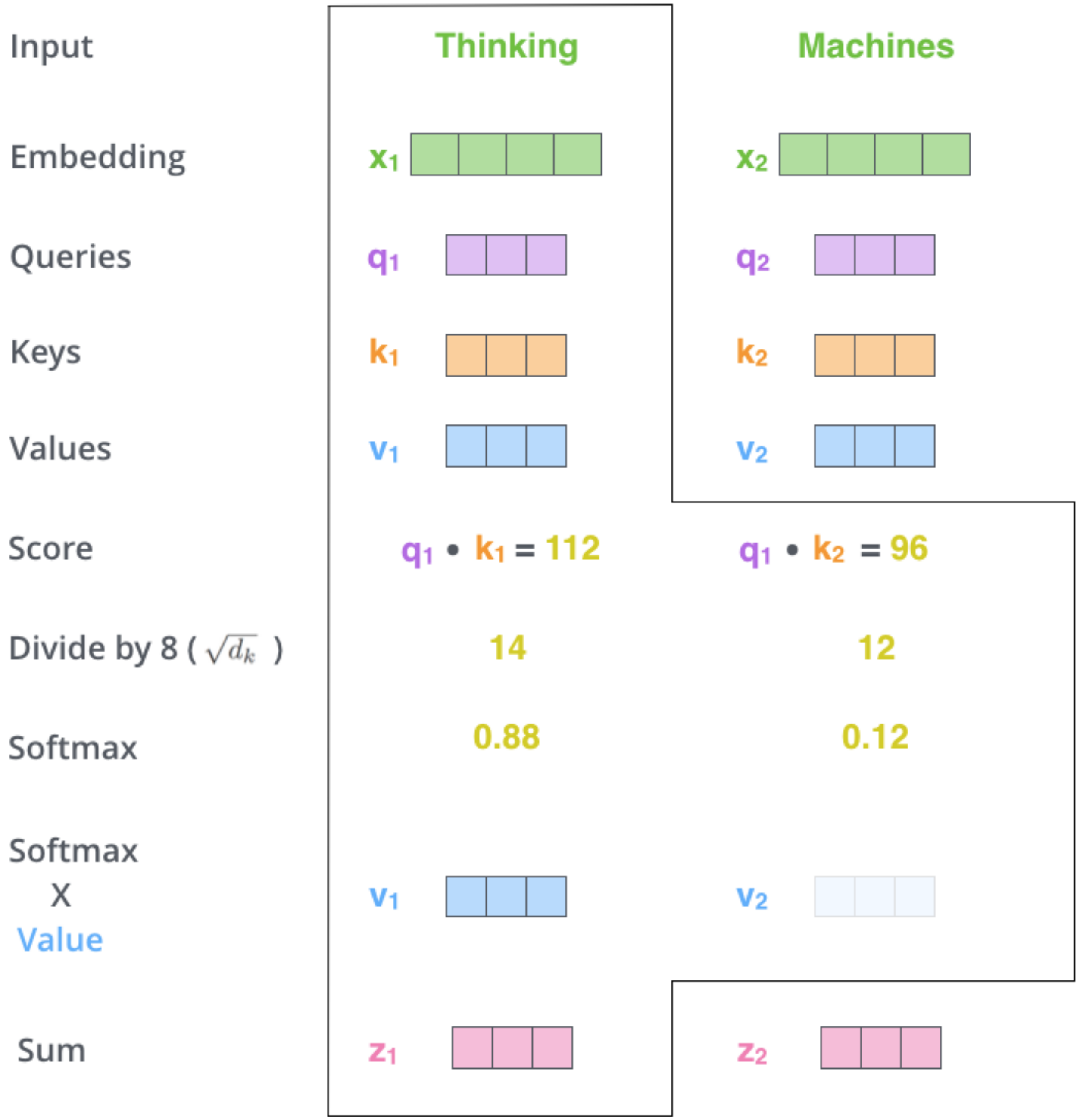
Görsel: https://jalammar.github.io/illustrated-transformer/



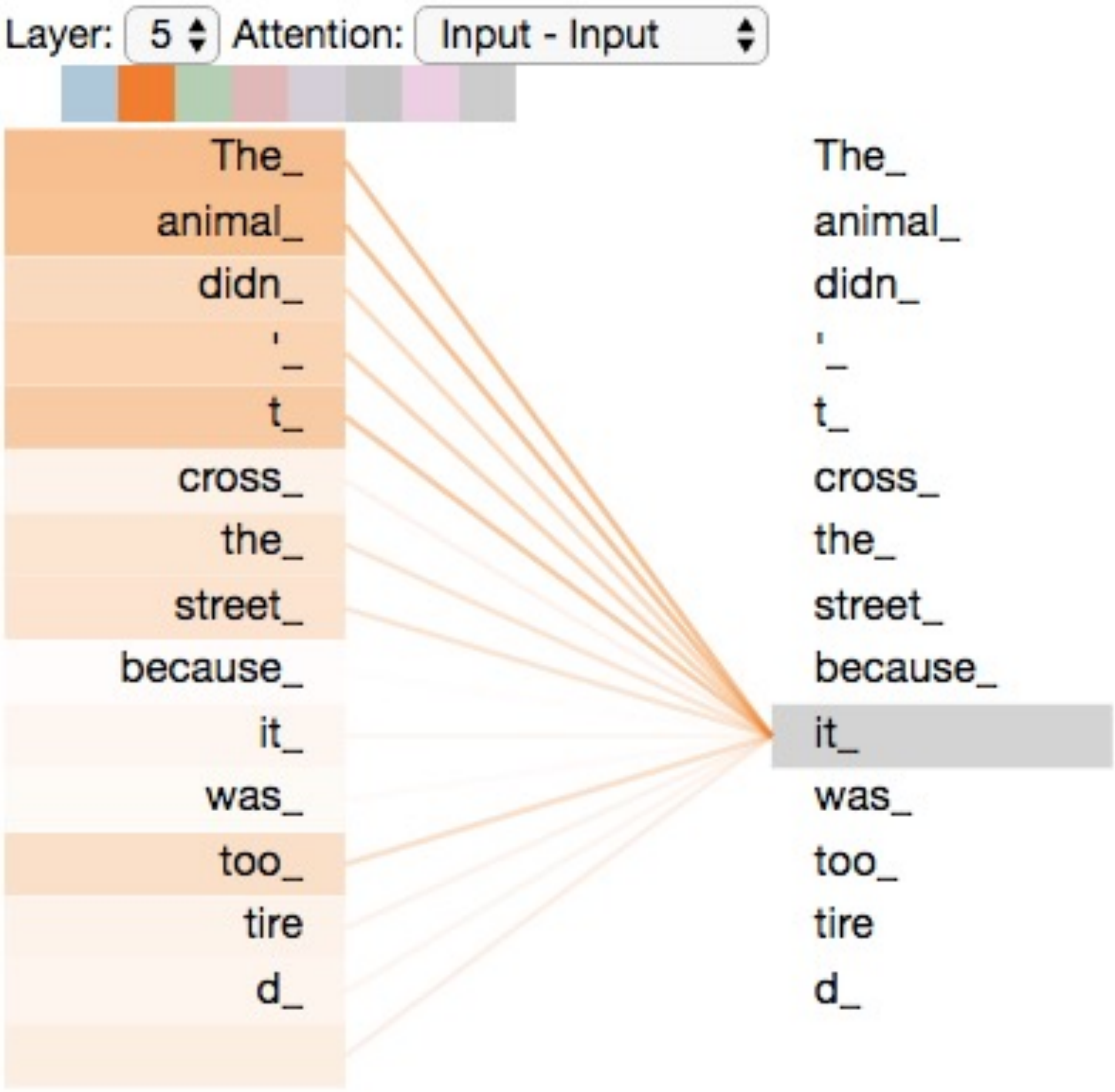
Görsel: https://jalammar.github.io/illustrated-transformer/



Görsel: https://jalammar.github.io/illustrated-transformer/



Görsel: https://jalammar.github.io/illustrated-transformer/



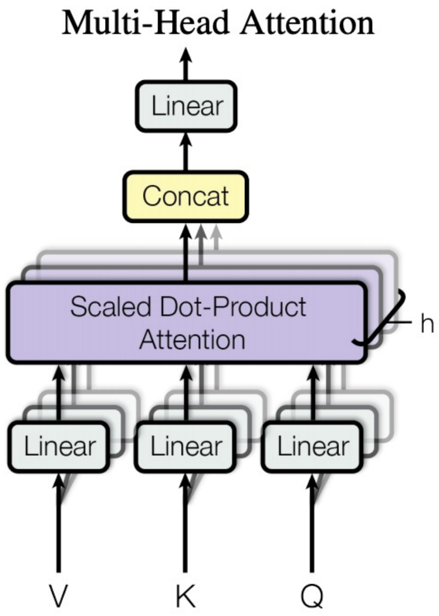
Görsel: https://jalammar.github.io/illustrated-transformer/

$$\mathbf{z}_1 = \sum_j \frac{\exp\left(q(\mathbf{e}_1)\cdot k(\mathbf{e}_j)\right)}{\sum_m \exp(q(\mathbf{e}_1)\cdot k(\mathbf{e}_m))} v(\mathbf{e}_j)$$

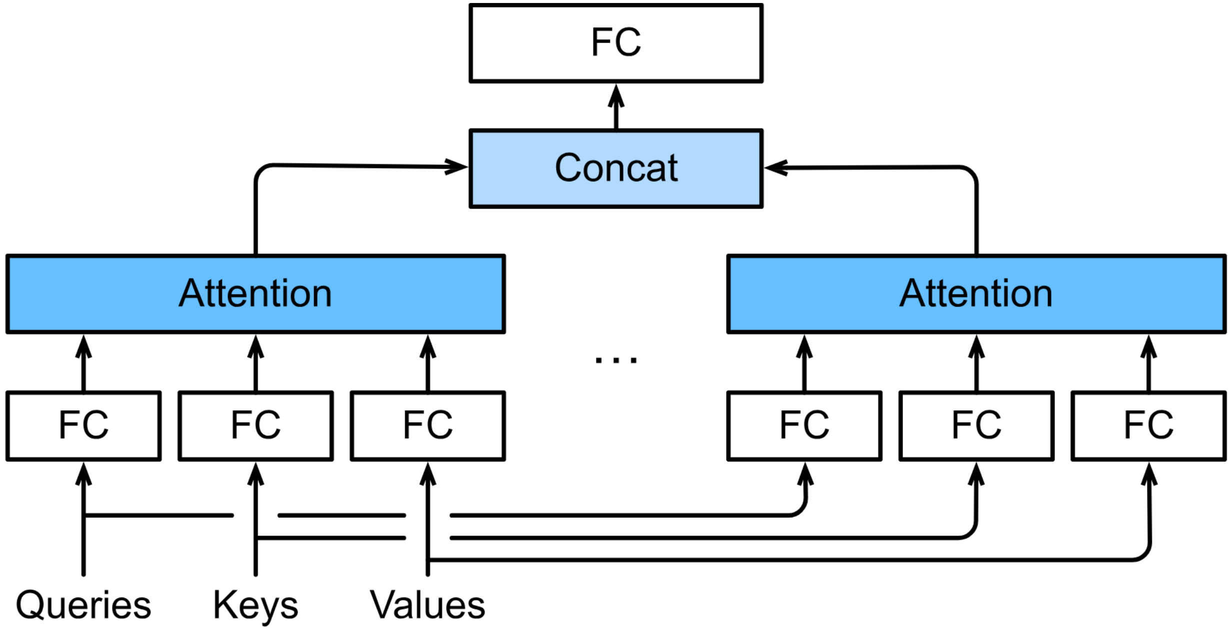
$$[\mathbf{z}_1, \ldots, \mathbf{z}_n] = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

```
import numpy as np

# 1. Cümlemiz (3 kelime, her biri 2 boyutlu)
# Kelime 1: [1, 0], Kelime 2: [0, 1], Kelime 3: [1, 1]
e = np.array([
    [1, 0], # Kelime 1
    [0, 1], # Kelime 2
    [1, 1]  # Kelime 3
])

# 2. Parametreleri rastgele değerlerle başlat
# Aslında bunları gradyan inişi ile öğreniyoruz:
W_q = np.array([[1, 0], [0, 1]])
W_k = np.array([[1, 1], [0, 1]])
W_v = np.array([[1, 2], [3, 0]])

# 3. Q, K ve V matrislerini hesapla
Q = x @ W_q
K = x @ W_k
V = x @ W_v

# 4. Benzerlikleri hesapla
benzerlik = Q @ K.T

# 5. Softmax
def softmax(x):
    return np.exp(x) / np.sum(np.exp(x), axis=-1,
keepdims=True)

dikkat_skorlari = softmax(benzerlik)

# 6. Sonuç: Bağlamı dikkate alan kelime anlamları
z = dikkat_skorlari @ V
```

Çok başlı öz-dikkat (ing. Multi-head self-attention)

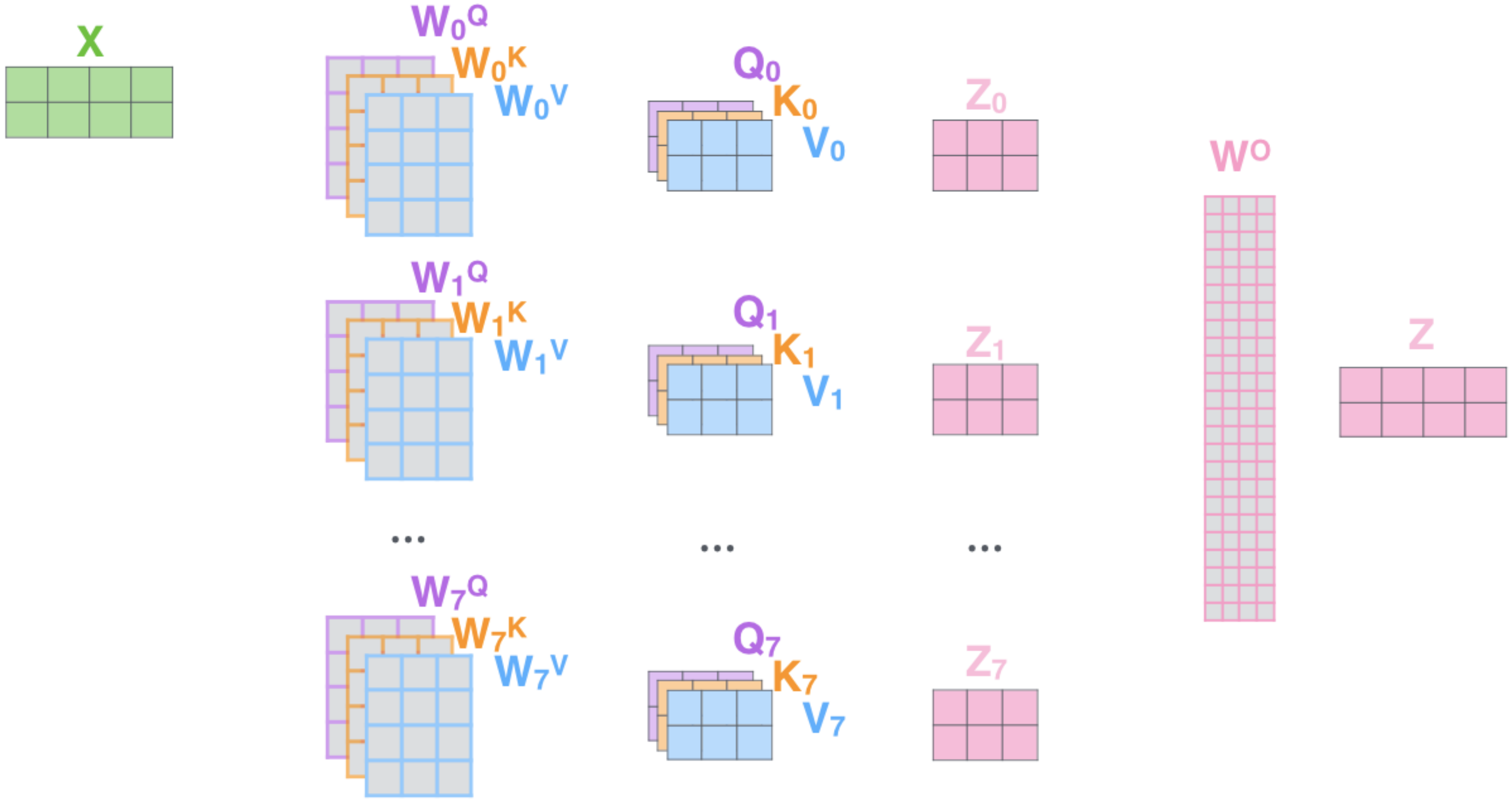


Görsel: Vaswani vd., “Attention is all you need”, 2017.



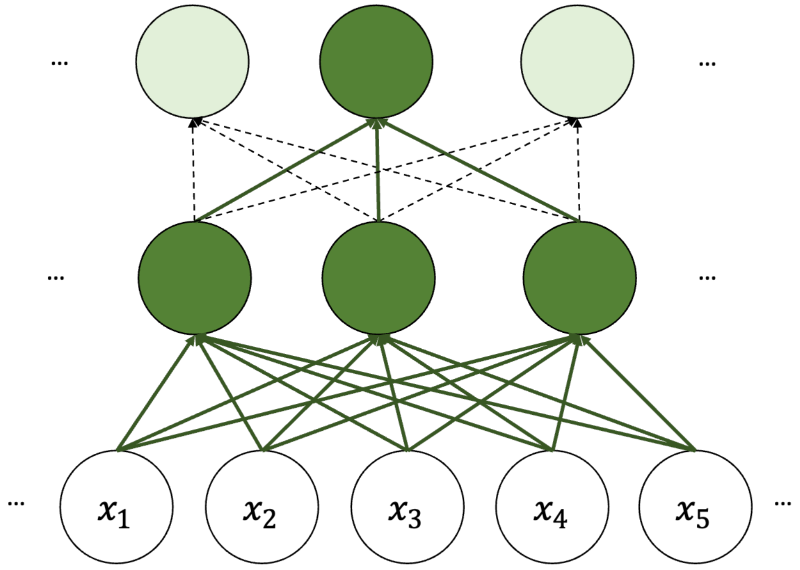
Görsel: https://d2l.ai/chapter_attention-mechanisms-and-transformers/multihead-attention.html

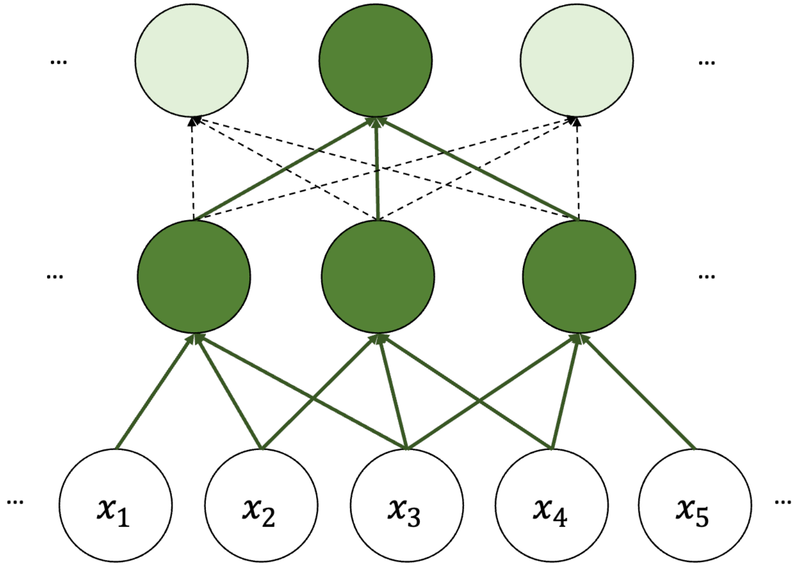
BAYZYO
Harika

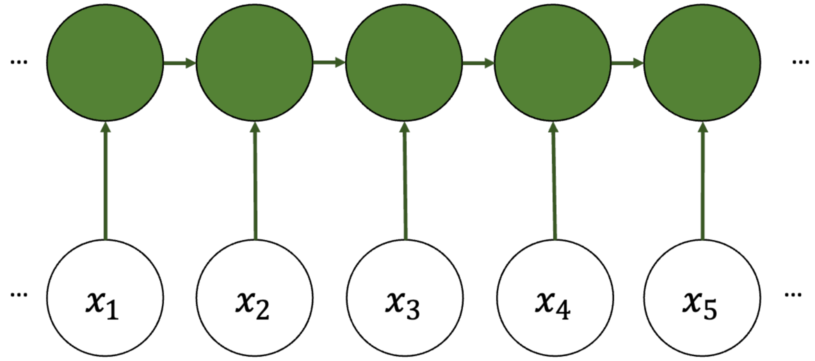


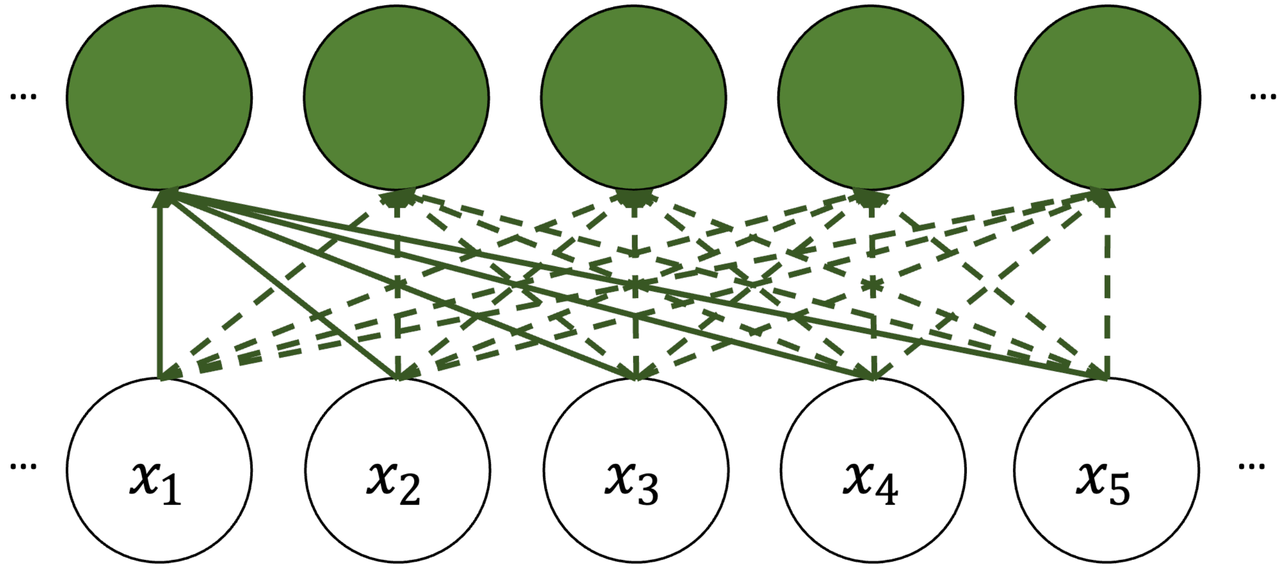
Görsel: https://jalammar.github.io/illustrated-transformer/

Öz-Dikkat, Evrişim, Özyineleme, Algılayıcı Kıyaslaması



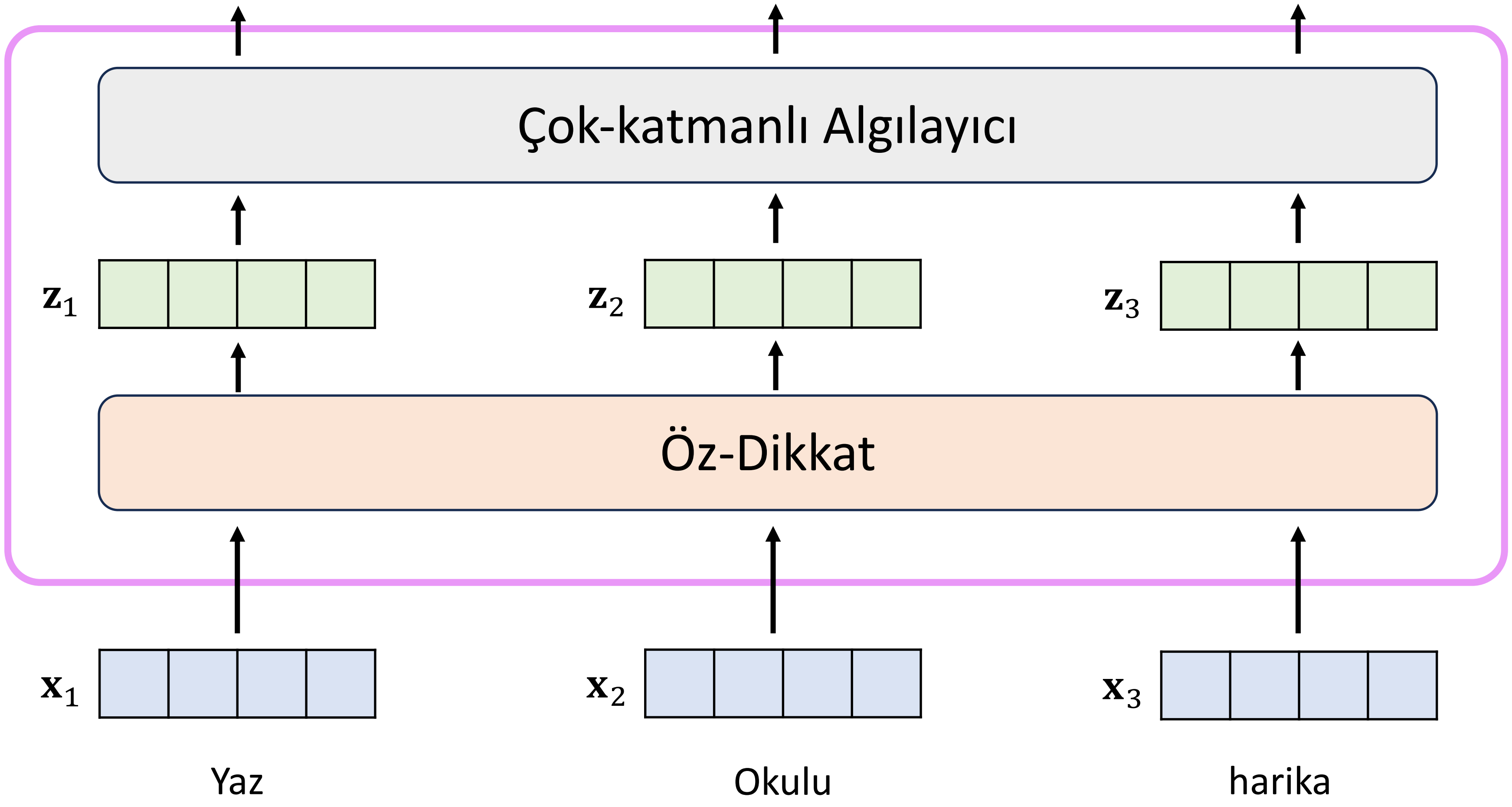




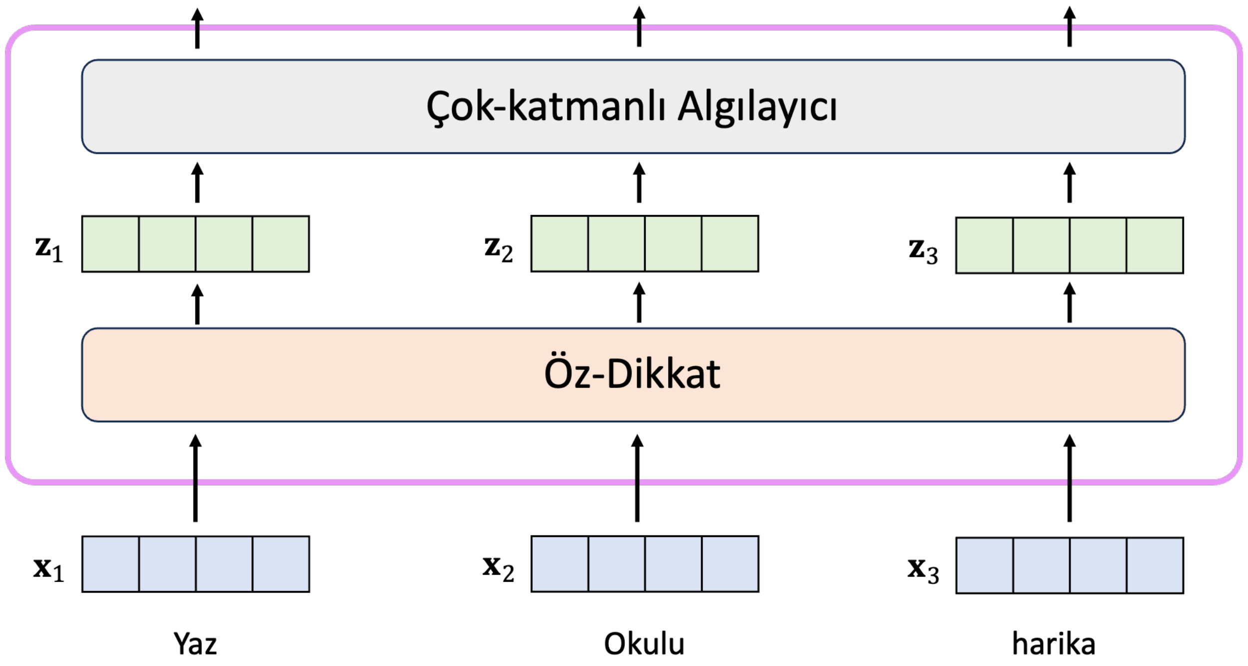


Dönüştürücüler ve Türleri

Dönüştürücüler ve Türleri



Dönüştürücüler ve Türleri



$$\mathbf{z}_1 = \sum_j \frac{\exp\left(q(\mathbf{e}_1) \cdot k(\mathbf{e}_j)\right)}{\sum_m \exp(q(\mathbf{e}_1) \cdot k(\mathbf{e}_m))} v(\mathbf{e}_j)$$

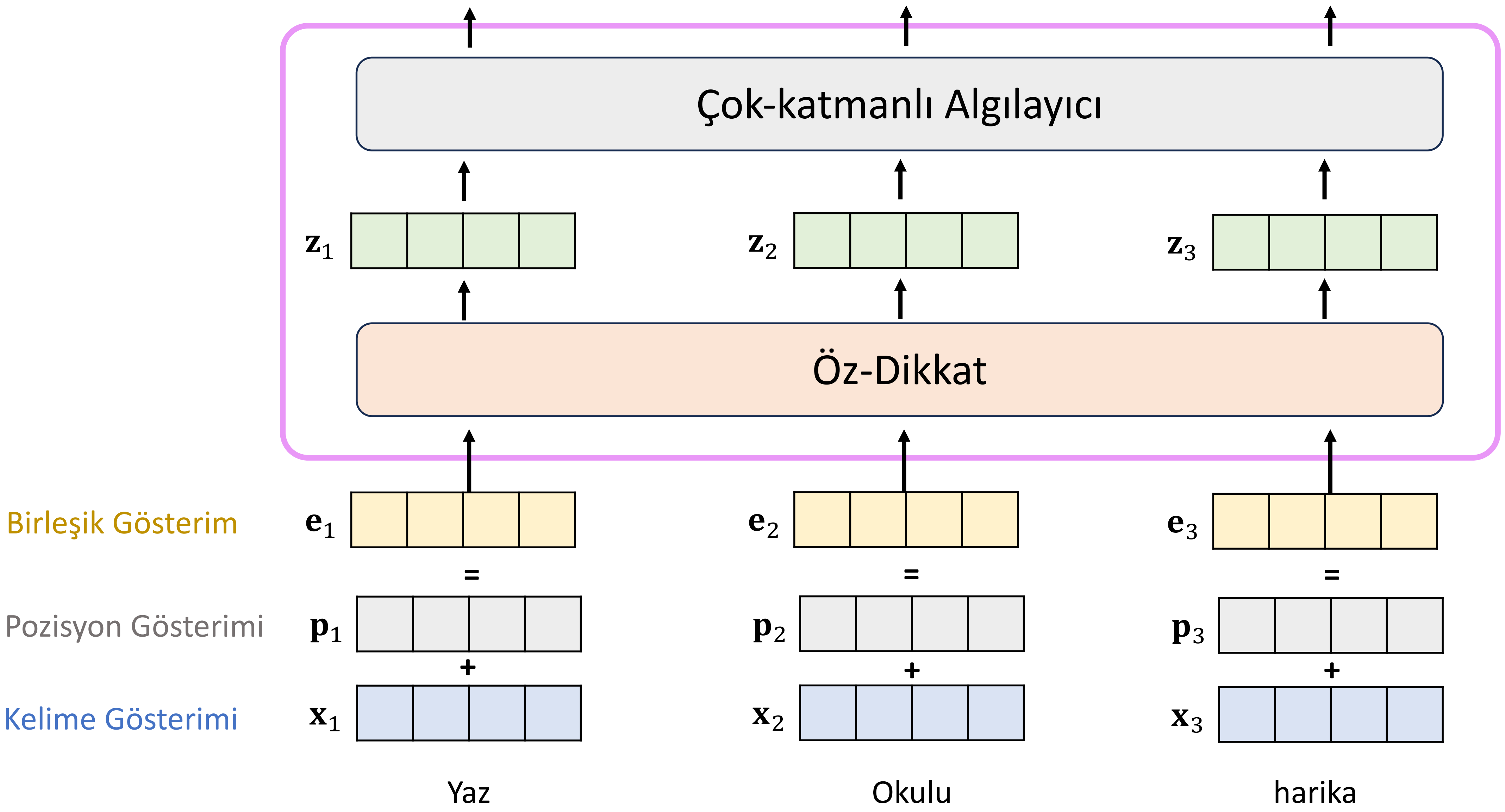
Pozisyon bilgisi yok!

Aynı kelimeler farklı sıralandığında farklı anlamlar ifade eder:

- **“man bites dog”**

  ve

- **“dog bites man”**


Çok-katmanlı Algılayıcı
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_3$
Öz-Dikkat
Birleşik Gösterim
$\mathbf{e}_1$
$\mathbf{e}_2$
$\mathbf{e}_3$
=
Pozisyon Gösterimi
$\mathbf{p}_1$
$\mathbf{p}_2$
$\mathbf{p}_3$
+
Kelime Gösterimi
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
Yaz
Okulu
harika

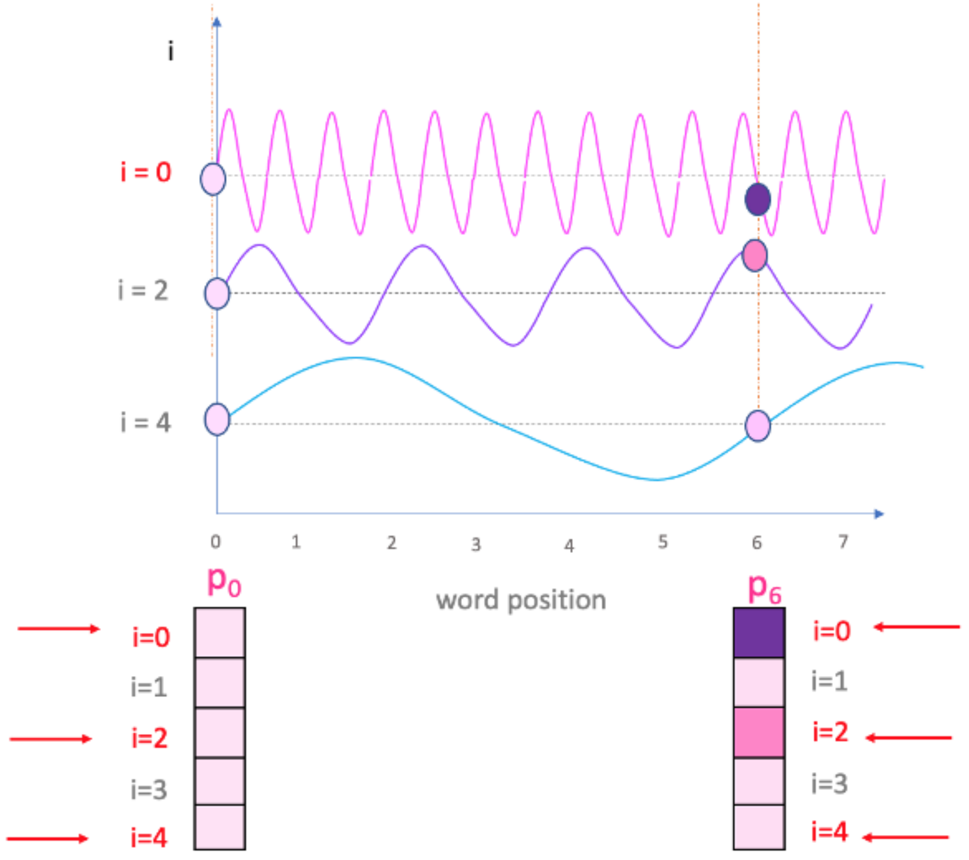
Pozisyon Gösterimleri

Zaman

	Indeks 2	Indeks 1	Indeks 0
Gösterim 0	0	0	0
Gösterim 1	0	0	1
Gösterim 2	0	1	0
Gösterim 3	0	1	1
Gösterim 4	1	0	0
Gösterim 5	1	0	1
Gösterim 6	1	1	0
Gösterim 7	1	1	1

Pozisyon Gösterimleri

$$p_{i,2t} = \sin(i/10000^{2t/d})$$
$$p_{i,2t+1} = \cos(i/10000^{2t/d})$$



Zaman

	Indeks 2	Indeks 1	Indeks 0
Gösterim 0	0	0	0
Gösterim 1	0	0	1
Gösterim 2	0	1	0
Gösterim 3	0	1	1
Gösterim 4	1	0	0
Gösterim 5	1	0	1
Gösterim 6	1	1	0
Gösterim 7	1	1	1

Görsel: https://www.youtube.com/watch?v=dichIcUZfOw

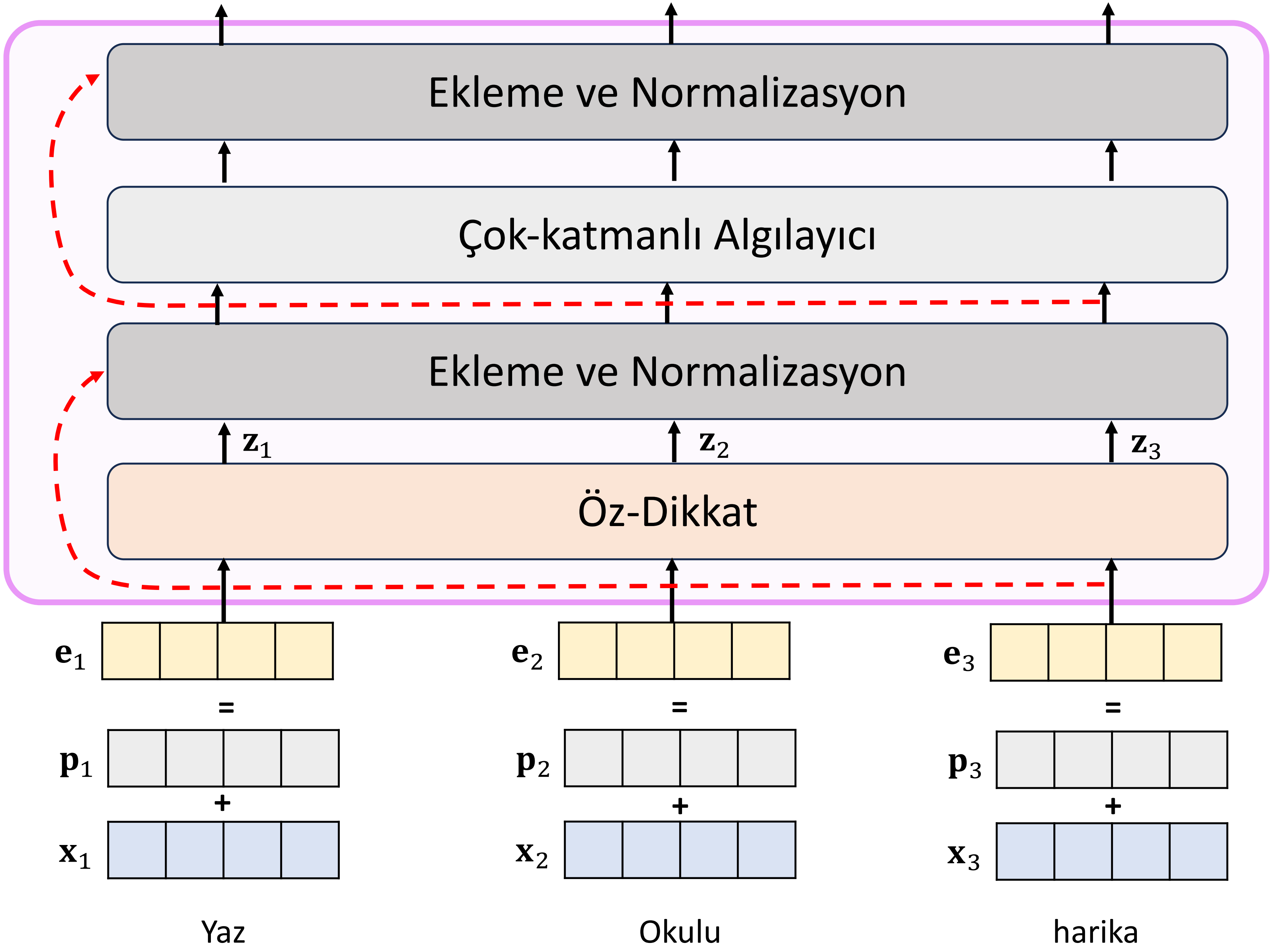
Pozisyon Gösterimleri

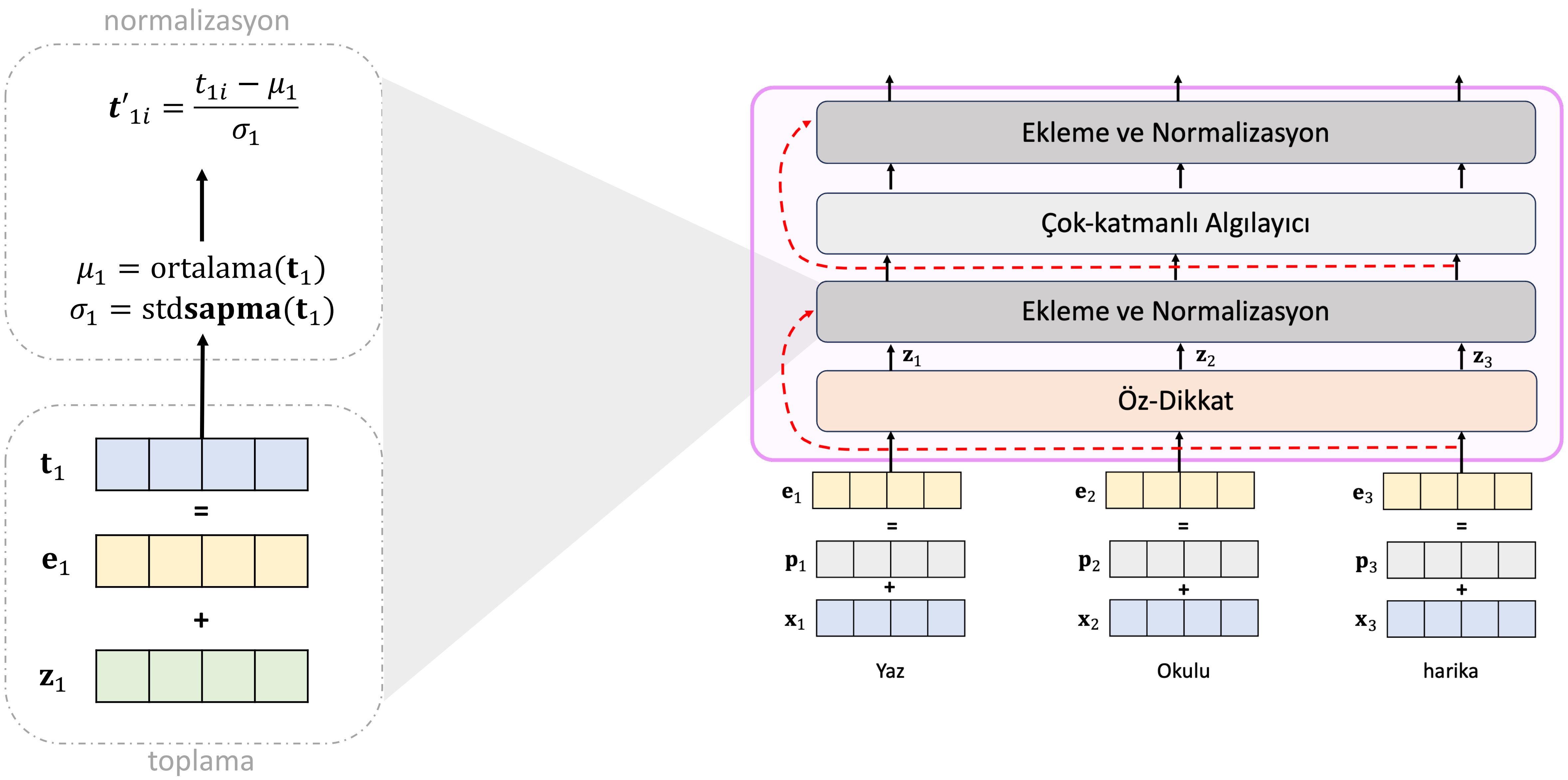
- Pozisyon gösterimlerini öğrenebiliriz
- Her adım türevlenebilir olduğu için gradyan inişi (gradient descent) ile bu gösterimleri güncelleyebiliriz

Zaman

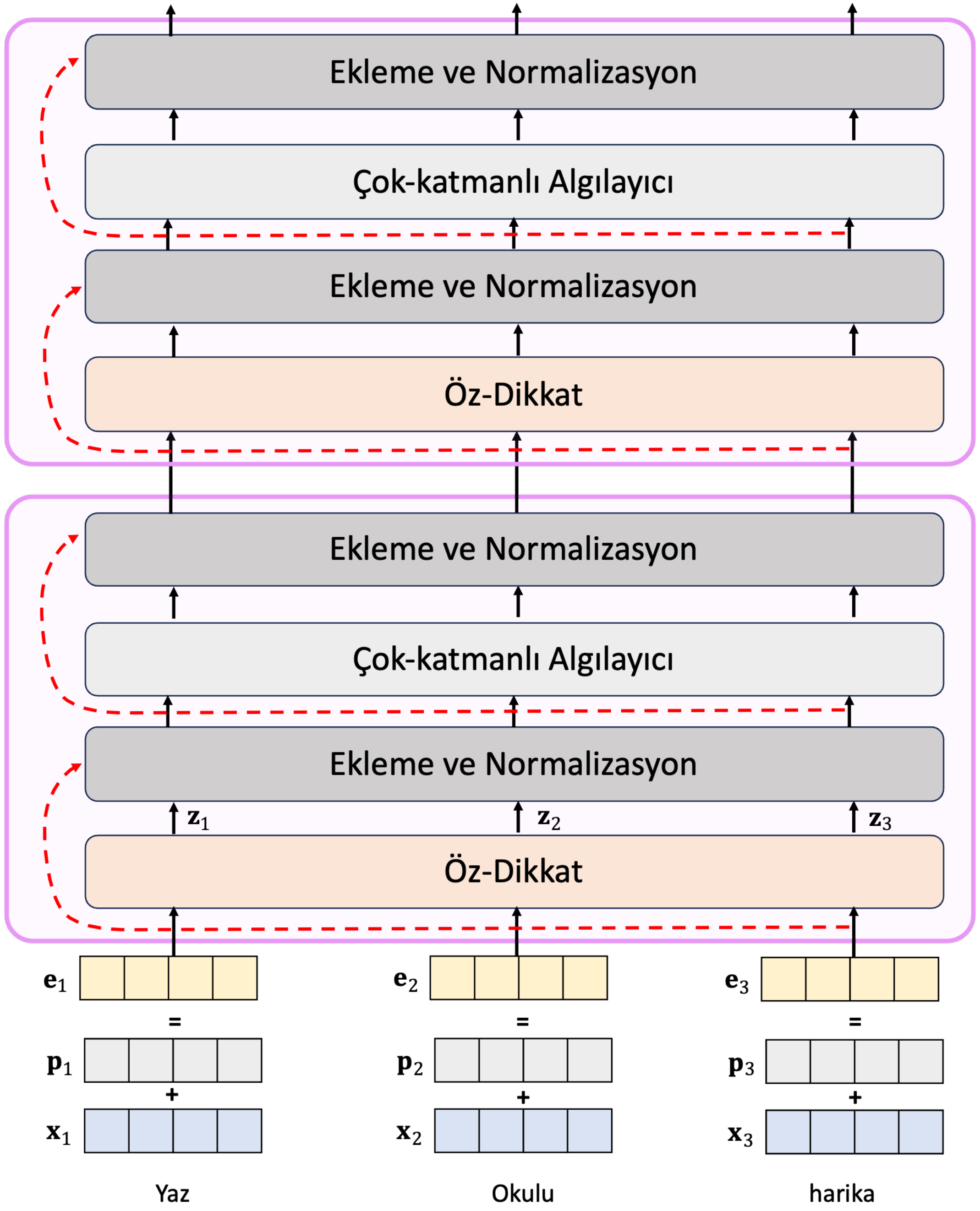
	Indeks 2	Indeks 1	Indeks 0
Gösterim 0	p_{02}	p_{01}	p_{00}
Gösterim 1	p_{12}	p_{11}	p_{10}
Gösterim 2	p_{22}	p_{21}	p_{20}
Gösterim 3	p_{32}	p_{31}	p_{30}
Gösterim 4	p_{42}	p_{41}	p_{40}
Gösterim 5	p_{52}	p_{51}	p_{50}
Gösterim 6	p_{62}	p_{61}	p_{50}
Gösterim 7	p_{72}	p_{71}	p_{70}

p_{ij}: öğrenilebilir parametre


Ekleme ve Normalizasyon
Çok-katmanlı Algılayıcı
Ekleme ve Normalizasyon
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_3$
Öz-Dikkat
$\mathbf{e}_1$
$\mathbf{e}_2$
$\mathbf{e}_3$
=
$\mathbf{p}_1$
$\mathbf{p}_2$
$\mathbf{p}_3$
+
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
Yaz
Okulu
harika

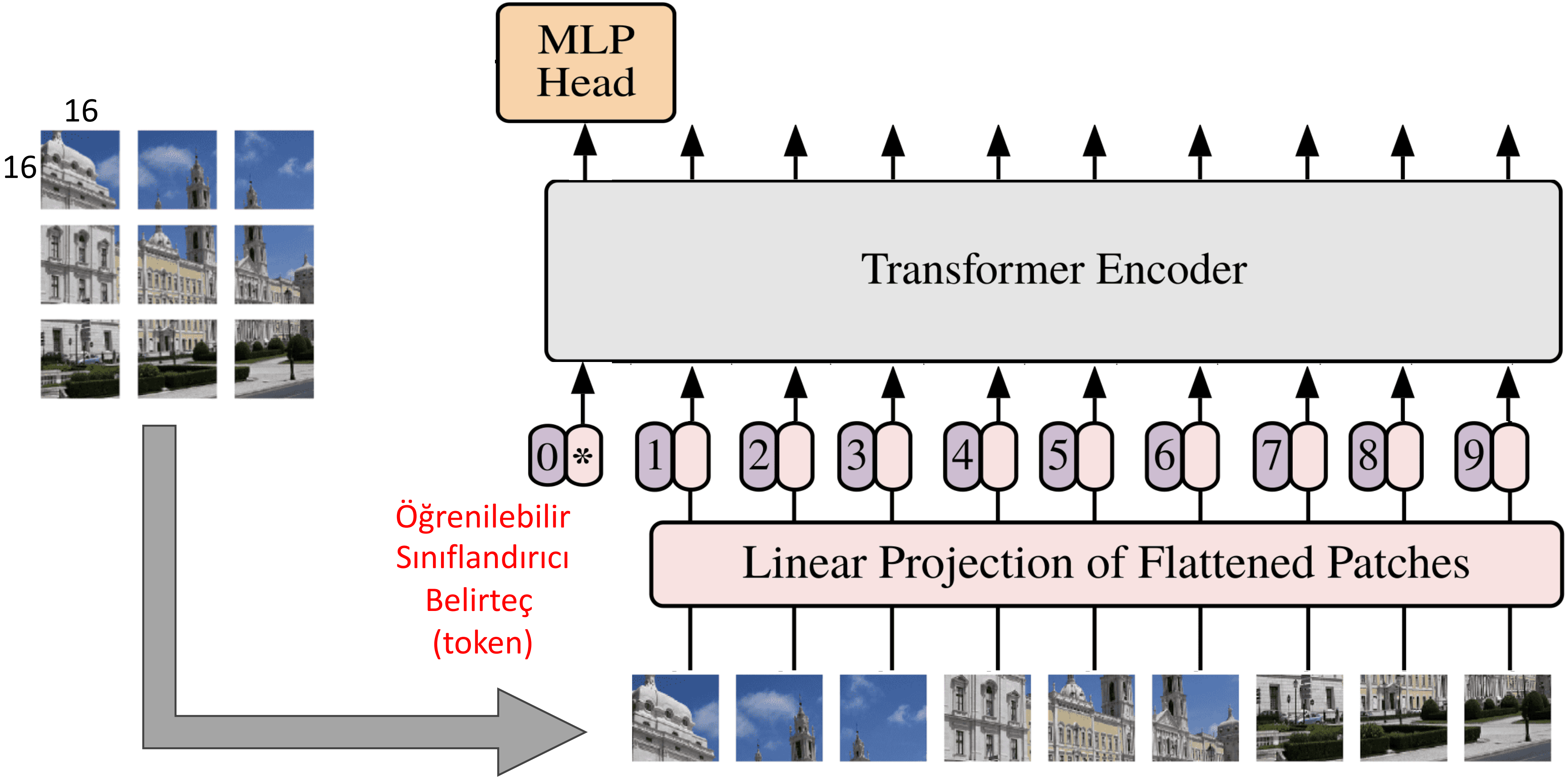

normalizasyon
$t'_{1i} = \frac{t_{1i} - \mu_1}{\sigma_1}$
$\mu_1 = \text{ortalama}(\mathbf{t}_1)$
$\sigma_1 = \text{std}\mathbf{sapma}(\mathbf{t}_1)$
$\mathbf{t}_1$
=
$\mathbf{e}_1$
+
$\mathbf{z}_1$
toplama
Ekleme ve Normalizasyon
Çok-katmanlı Algılayıcı
Ekleme ve Normalizasyon
$\mathbf{z}_1$ $\mathbf{z}_2$ $\mathbf{z}_3$
Öz-Dikkat
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{e}_3$
$\mathbf{p}_1$ $\mathbf{p}_2$ $\mathbf{p}_3$
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$
Yaz
Okulu
harika

Kodlayıcı (Encoder)

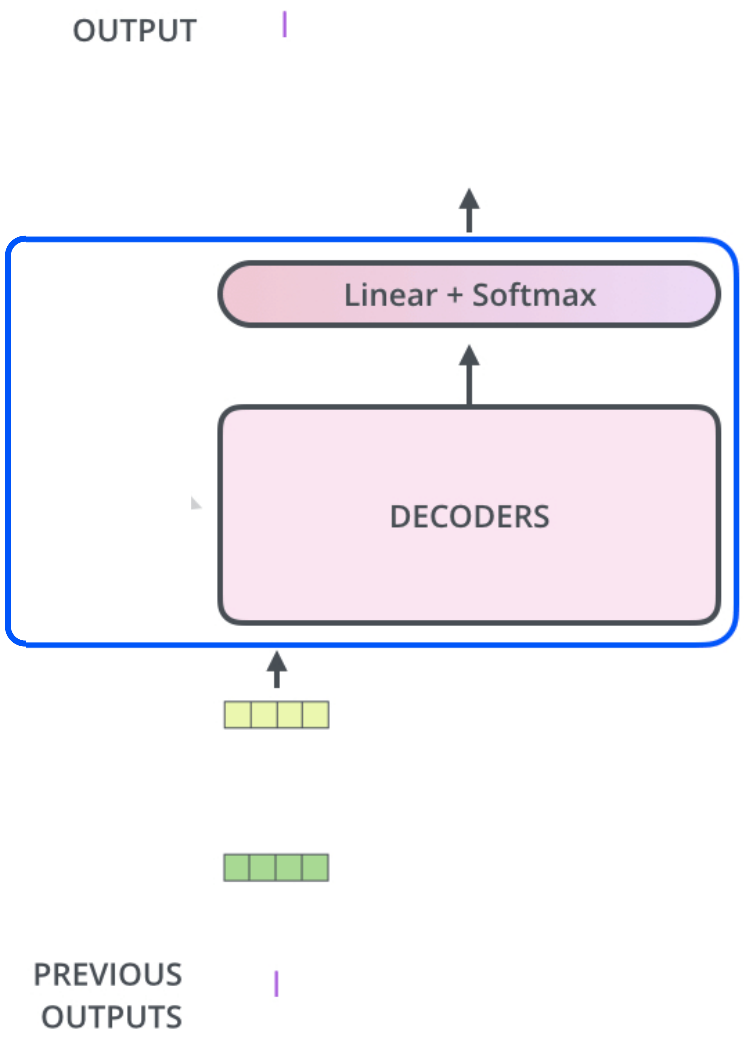


Örnek: Vision Transformers (ViT)

Dosovitskiy vd., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", 2020.

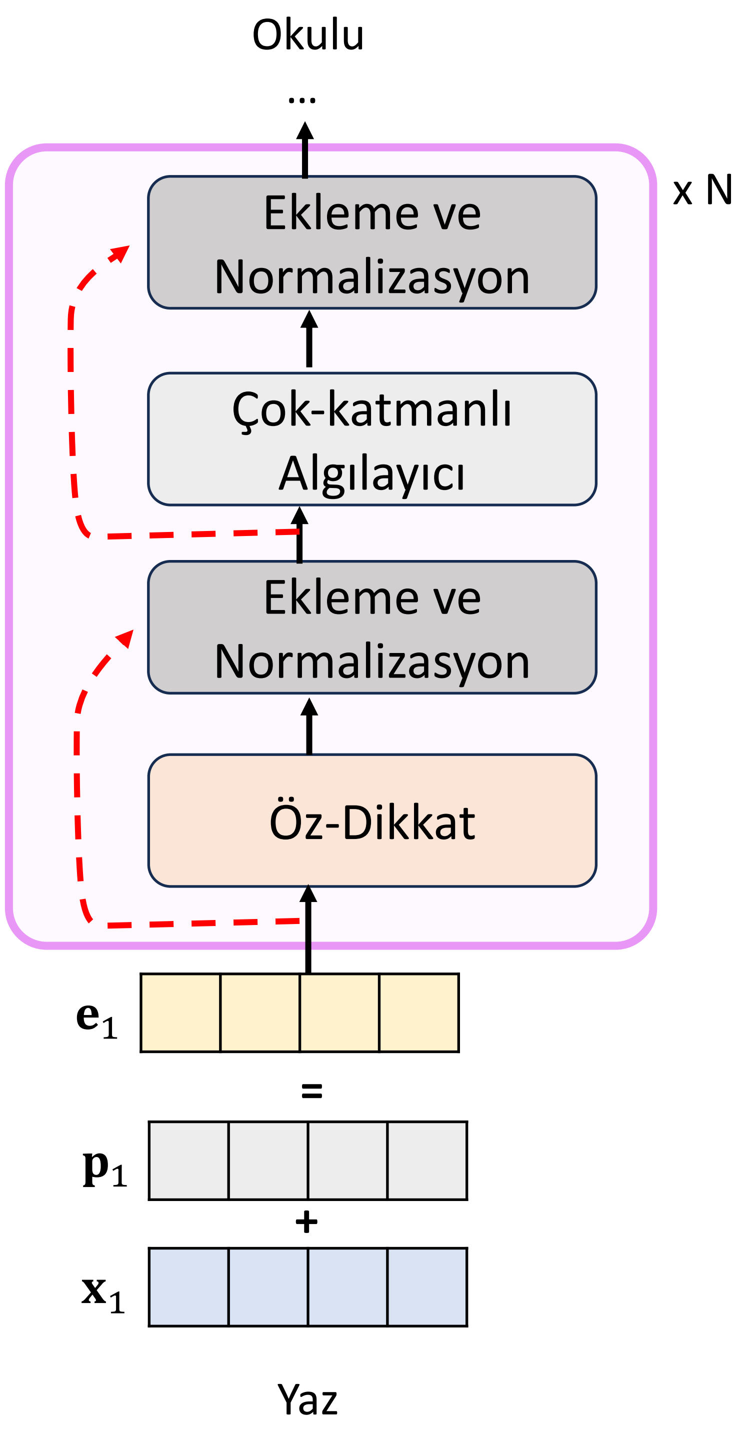


Çözücü (Decoder)

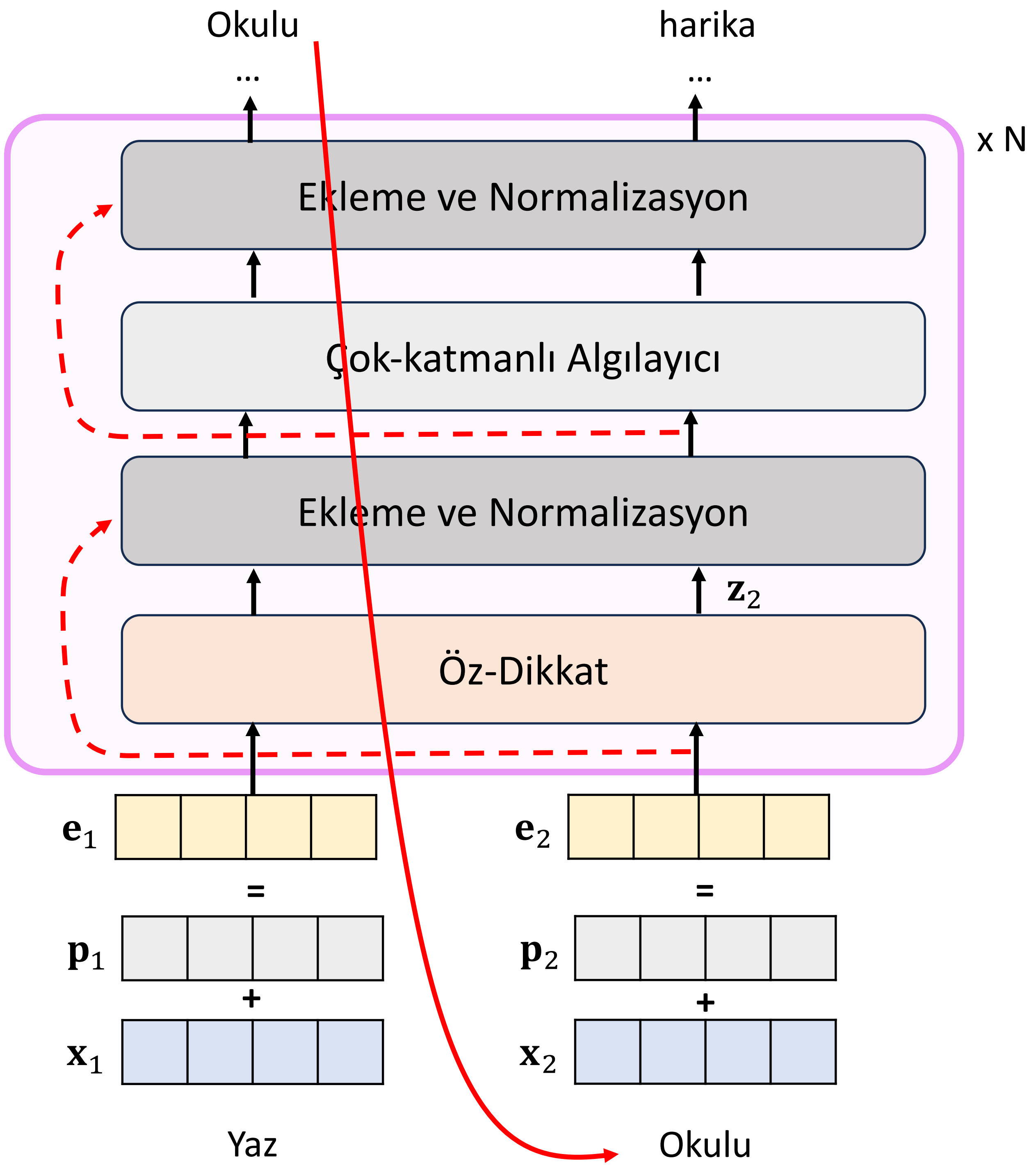


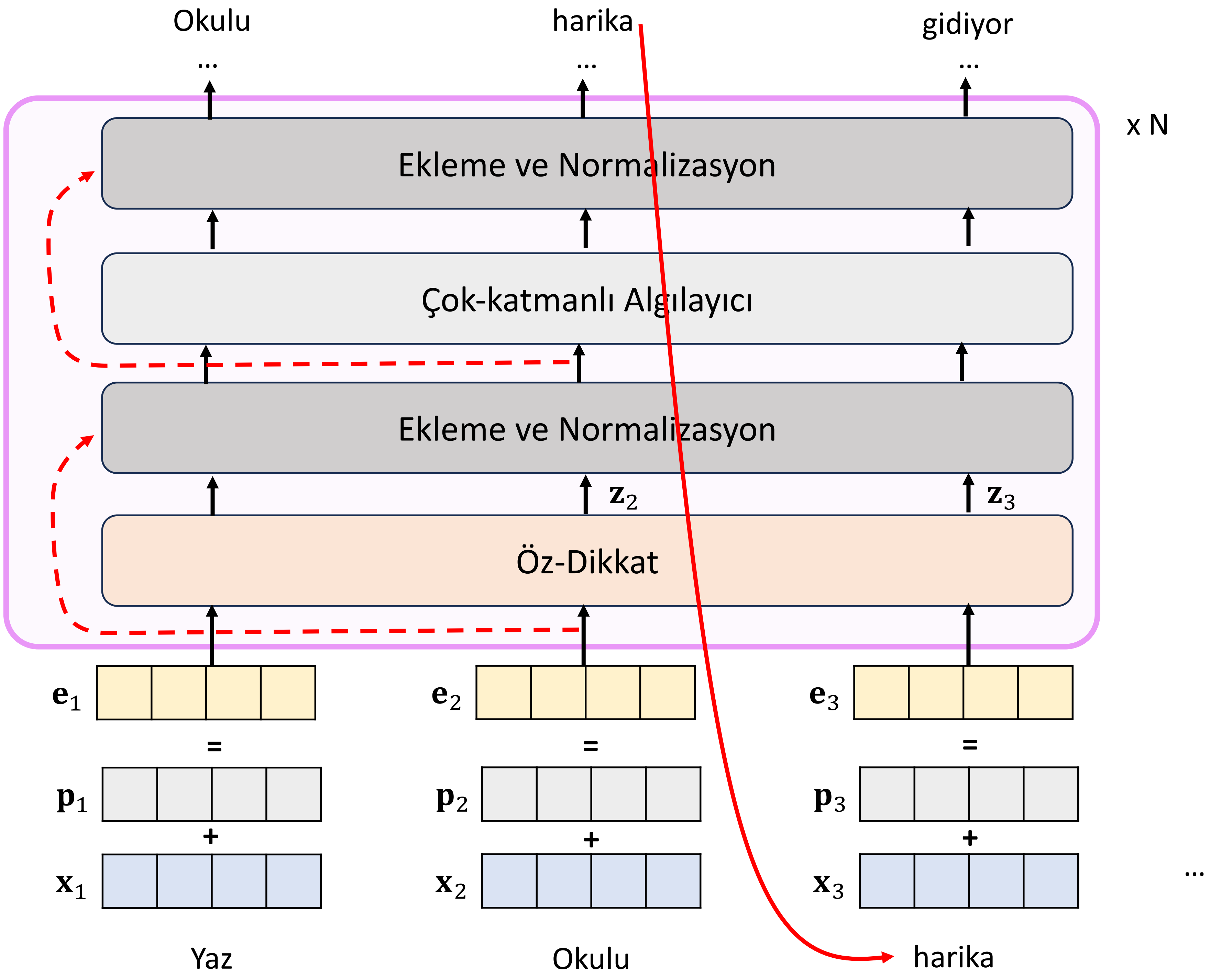
Görsel: https://jalammar.github.io/illustrated-transformer/

Çözücü (Decoder)

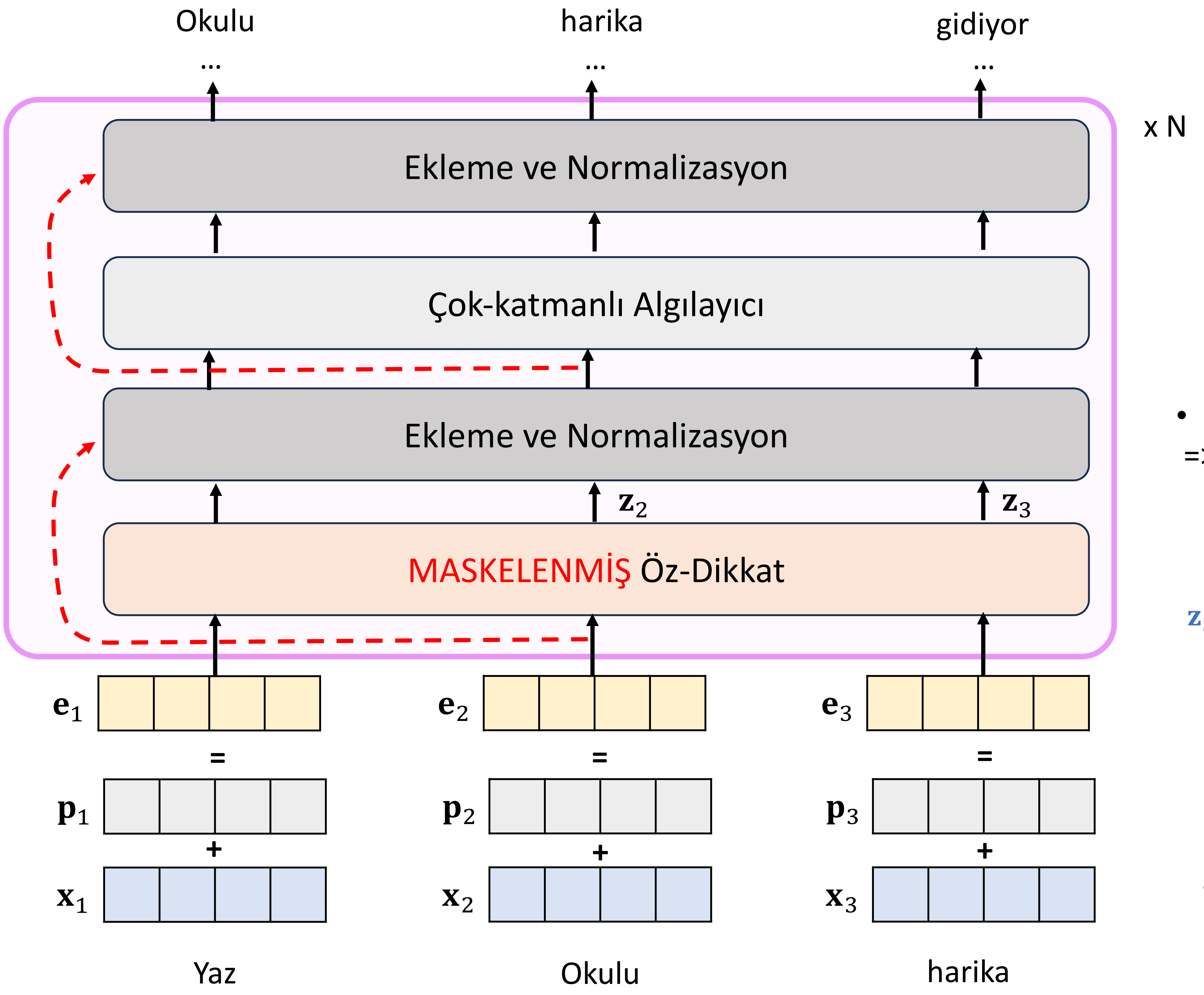


Çözücü (Decoder)





Çözücü (Decoder)



Çözücü (Decoder)

Maskelenmiş Öz-Dikkat Eğitim Esnasında

- İleri adımları hesaba katma.

=> O adımların dikkat skorunu sıfırla

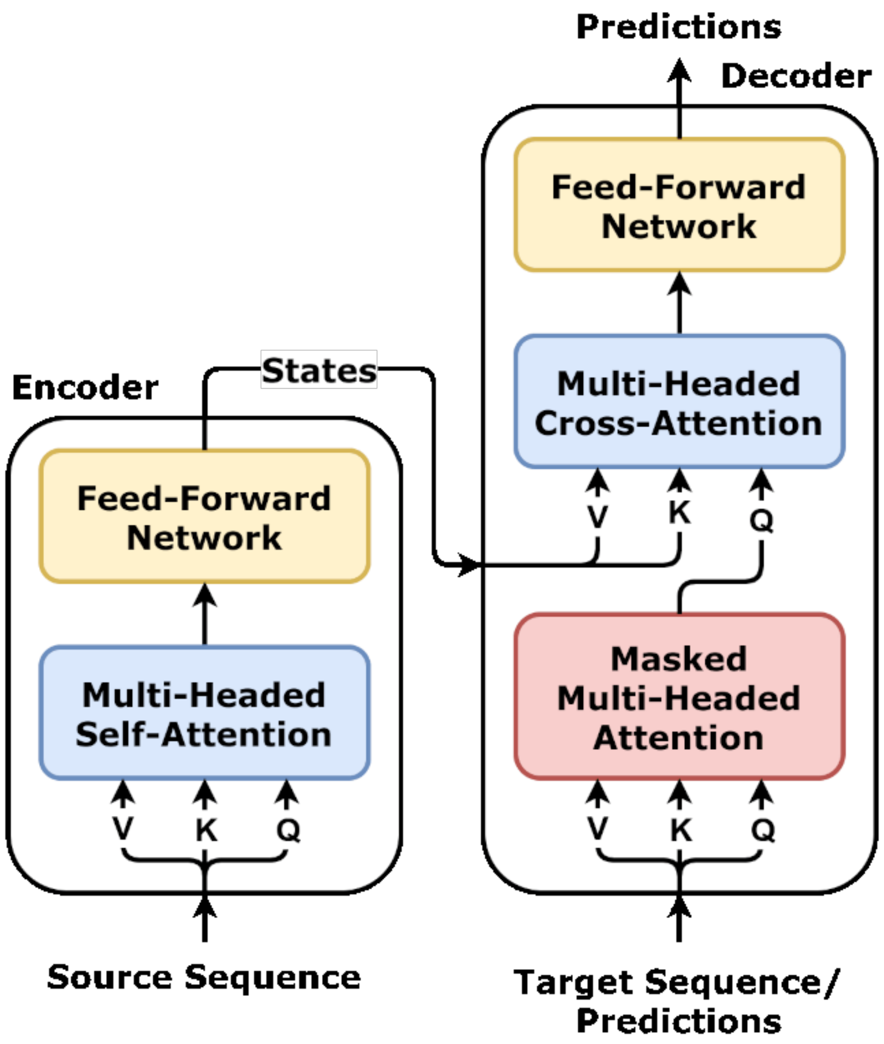
$$j > i \Rightarrow -\infty$$

$$\mathbf{z}_i = \sum_j \frac{\exp\left(\boxed{q(\mathbf{e}_i)\cdot k(\mathbf{e}_j)}\right)}{\sum_m \exp\left(\boxed{q(\mathbf{e}_i)\cdot k(\mathbf{e}_m)}\right)} v(\mathbf{e}_j)$$

$$m > i \Rightarrow -\infty$$

...

Kodlayıcı-Çözücü Dönüştürücüler



Görsel: https://commons.wikimedia.org/wiki/File:Transformer,_one_encoder-decoder_block.png

Dönüştürücülerin Geleceği

Dönüştürücülerin Geleceği

Sorunlar / Zorluklar

✗ **Hesaplama Maliyeti**

Öz-dikkat mekanizmasının girdi uzunluğuna göre kuadratik (n adet token için $\mathcal{O}(n^2)$) büyümesi.

✗ **Hafıza Darboğazı**

Çıkarım (inference) anında KV matrisleri nedeniyle hafıza tüketiminin hızla artması.

✗ **Bağlam Kaybı**

Girdi uzadıkça modelin ortadaki bilgileri unutması (*"Lost in the middle"* fenomeni).

✗ **Veri İhtiyacı**

Sıfırdan eğitimin devasa ölçekte yüksek kaliteli veriye ihtiyaç duyması.

Alternatifler / İyileştirmeler

✓ **Alternatif Mekanizmalar**

Lineer Öz-dikkat ve State Space Models (Örn: Mamba).

✓ **Donanım Farkındalığı (Hardware-Aware)**

FlashAttention ile I/O darboğazının aşılması.

✓ **Mimari Verimlilik**

Çok-başlı Dikkatte KV matrislerini paylaşma.

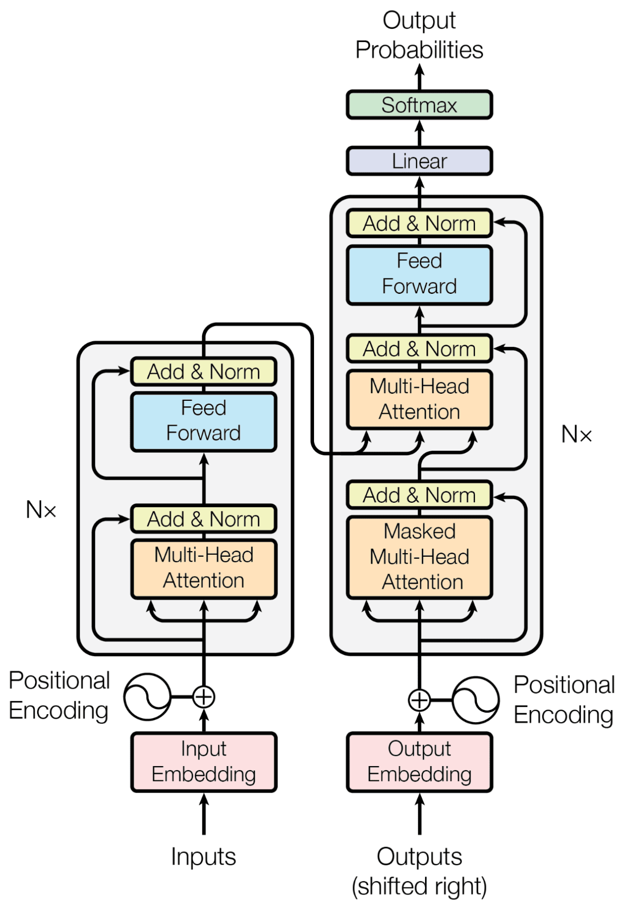
✓ **Yeni Paradigmalar**

Test zamanı öğrenme / hesaplama

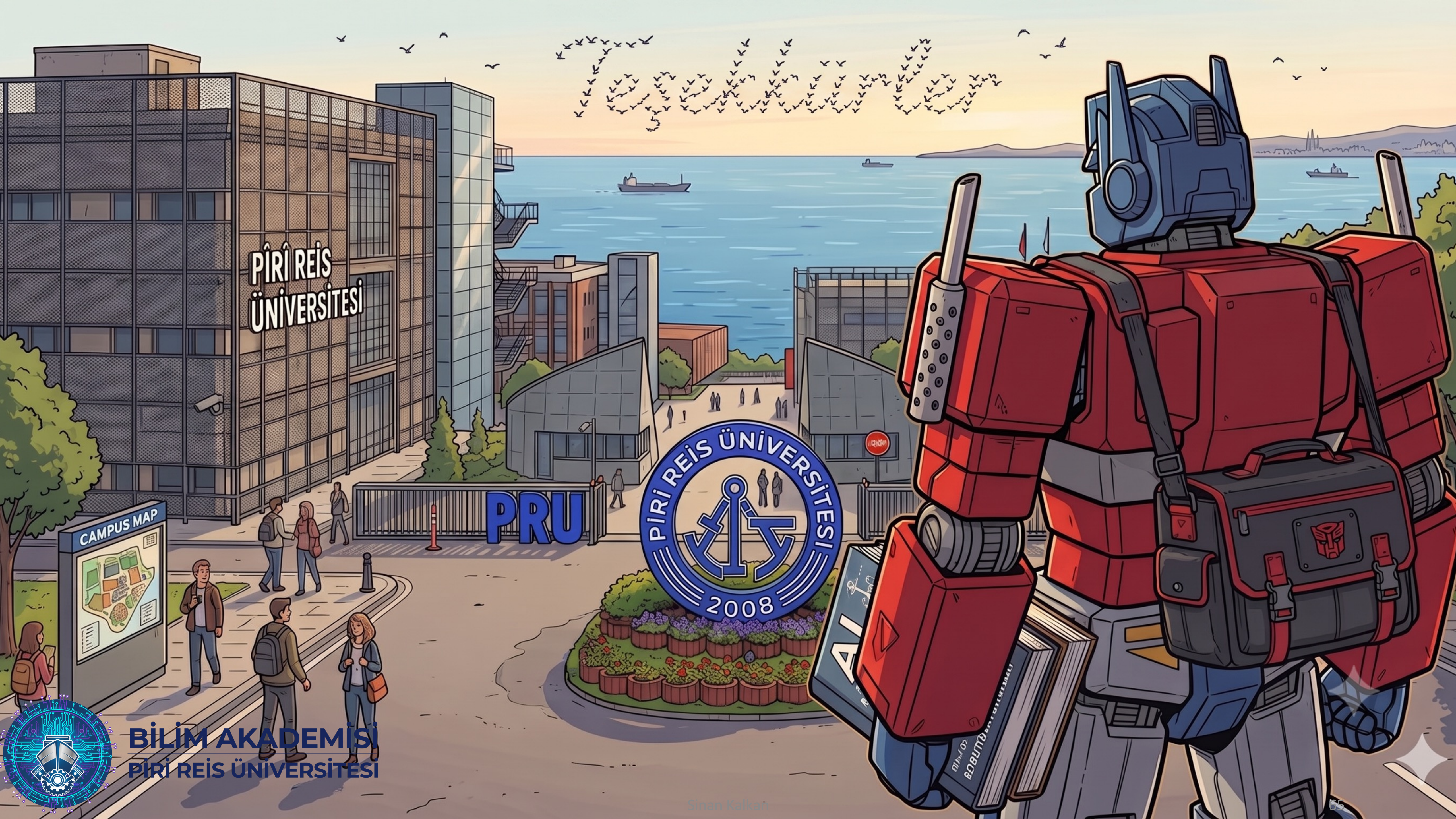
Sonuç

- Dönüştürücüler, öz-dikkat mekanizması sayesinde karmaşık durumlarda/problemlerde en uygun girdiyi seçme yeteneğine sahip

- Performanslarının uzun dizilerde artırılması, hesaplama kaynağı ihtiyacının iyileştirilmesi üzerinde çalışılan konular arasında



Vaswani vd., "Attention Is All You Need", 2017.



BİLİM AKADEMİSİ
PİRİ REİS ÜNİVERSİTESİ