

Dil İşlemede Dilbilgisi Kuralları ile Çözümlemeden
Sinir Ağlarının Kullanımına Geçiş

«Constituency» Dilbilgisi Örnekleri (devam)

□ İsim Öbeği örnekleri

- ❖ Sözcüklerin nasıl bir araya gelerek gruplandıkları ifade edilir.
- ❖ En az bir tane isim içerirler.
- ❖ Bu kelimelerin bir arada gruplandığına dair hangi kanıtlar vardır?
 - ✓ Harry the Horse arrive(s)
 - ✓ a high-class spot such as Mindy's attract(s)
 - ✓ the Broadway coppers love(s)
 - ✓ the reason he comes into the Hot Box sit(s)
 - ✓ they
 - ✓ three parties from Brooklyn

Dilbilgisi (Grammar) ve Sözdizimsel Yapı (Constitutents)

- ❑ Gramer (dilbilgisi), içeriklerin (kendini oluşturan öğelerin) nasıl düzenleneceğini açıklar.
- ❑ Bizim için örtük (implicit) bilgidir.
 - ❖ Çünkü herhangi bir yerde yanlış varsa neden yanlış olduğu bu genel yapıda çıkarılamaz.
 - ❖ Sadece dile ait olan olası cümleler üretilir.
- ❑ Bu özelliği ile dilin sözdizimsel yapı, dilin anlamından (meaning) farklıdır.
 - ❖ Örnek olarak aşağıdaki cümle incelensin:

Colorless green ideas sleep furiously
 - ❖ Sözcüklerin sıralanışı doğrudur.
 - ❖ Verilmek istenen düşünce: green ve colorless sözcükleridir.
 - ❖ Vurgulanmak istenilen: That ideas sleep
 - ❖ Vurgulanmak istenilen That sleeping is done furiously
 - ❖ Yandaki cümle sıralanışında değildir: sleep green furiously ideas colorless

Dependency Parsing (Bağlılık Ayırıştırması /Çözümlemesi)

(tekrar)

- ❑Bağlılık ayırıştırması, her bir kelimeyi bir başkasına (bu kelime baş (head) olarak kabul edilir) ekleyerek belirli bir cümlenin dilbilgisi yapısı ayırıştırılır.
- ❑Bu ilişkilendirmelerin sonunda tüm bu yönlendirilmiş bağlantılar bir bağlılık ağacında toplanır.
- ❑Bu yönlendirilmiş bağlantılar, bağlılıkları sözdizimsel işlevleri tanımlayan etiketlerle daha da etkinleştirilir (doğrulukları onaylanır).

Dil Ayırıştırmanın / Çözümlemenin Kullanıldığı Alanlar

Dilbilgisi kontrolü / Grammar checkers

Diyalog Sistemleri / Dialog systems

Yüksek Kesinlikte Soru Cevap Sistemleri /High precision question answering

Adlandırılmış varlık tanıma /Named entity recognition

Cümle Sıkıştırma / Sentence compression

Fikir madenciliği / Opinion mining

Ürünler hakkında görüş çıkarma / Extracting opinions about products

Bilgisayar oyunlarında etkileşimin iyileştirilmesi / Improved interaction in computer games

Dilbilimcilerin veri bulmasına yardımcı olma / Helping linguists find data

Makine Çevirisi /Machine translation

İlişki çıkarma sistemleri /Relation extraction systems

.....

Tanırma (Recognition) ve özümleme (Parsing) Problemi

Recognition

- Bir x cümlesi ve G dilbilgisi verildiğinde, geçerli bir çözümleme ağacı (parsing tree) ile türetilir ve cümlenin tanımlanan dilbilgisine ait olduğu gösterilir.
 - ❖ Buna tanıma (recognition) denir.
 - ❖ Bu adıma göre gerçekleştirilen “*ispat*” bir çıkarım, geçerli bir ayrıştırma ağacıdır.

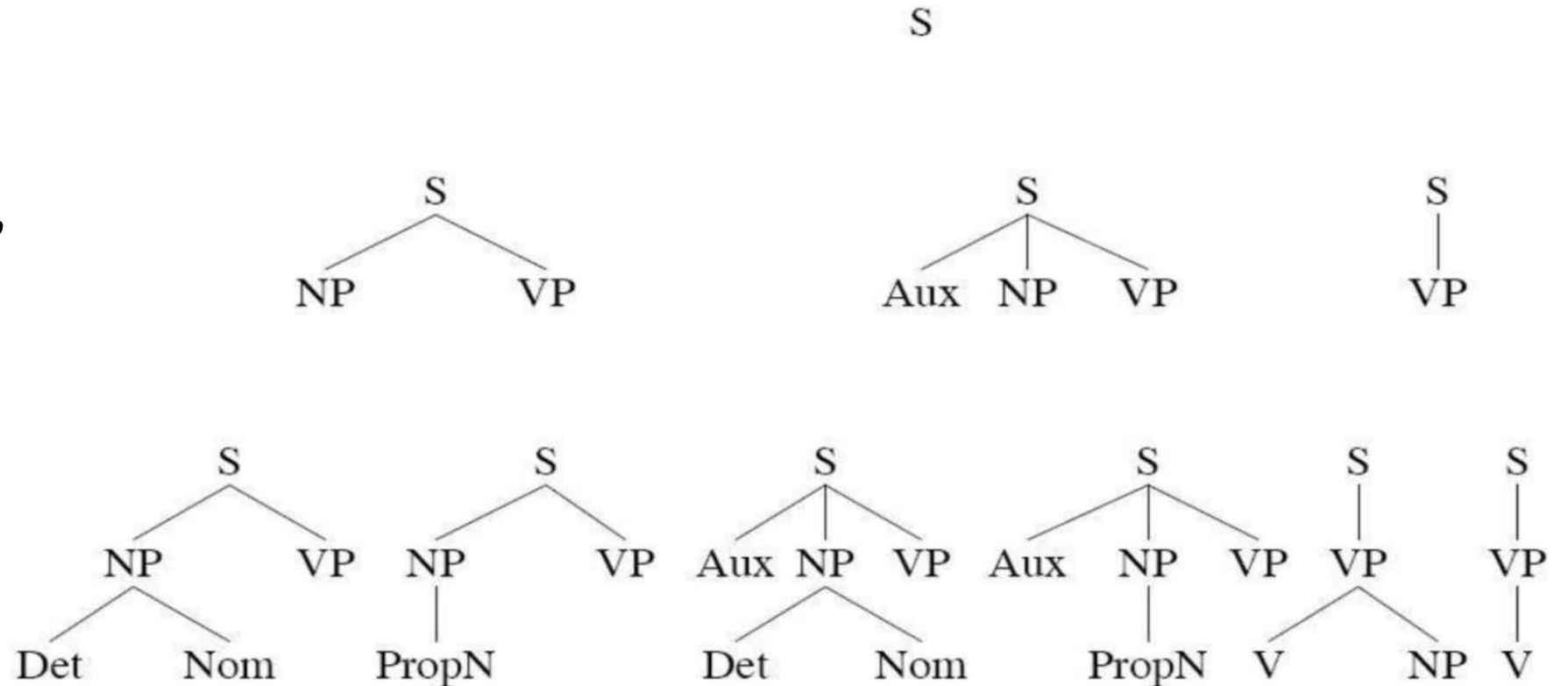
Parsing

- Bir x cümlesi bir G dilbilgisine ait ise, bunun ispatlanması istenecektir.
 - ❖ İspat, dar kapsamlı dilbilgisi kuralları için bile üstel düzeyde gerçekleştiği için oldukça zorlayıcıdır.
 - ❖ **Sonuç olarak:** Bir G dilbilgisine ait bir x cümlesi için bir veya daha fazla türetme olup olmadığının gösterilmesi gerekir.

Türetme Ağaçları: Yukarıdan Aşağı Çözümleme (1. yol)

- ❑ Türetme ağaçlarından ve dilbilgisi kurallarından sadece geçerli ağaçlar ele alınır.
- ❑ Fakat “Book that flight” cümlesi için tüm olası çözüm ağaçları alındığında, bazı çözümleme ağaçlarında kelimelerle uyumsuzluklar görülür.

“Book that flight”



Türetme Ağaçları: Aşağıdan Yukarı Çözümleme (2. yol)

Noun Det Noun
| | |
Book that flight

Verb Det Noun
| | |
Book that flight

Nominal Nominal
| | |
Noun Det Noun
| | |
Book that flight

Nominal
|
Noun
|
Verb Det Noun
| | |
Book that flight

NP
/ | \
Nominal Det Nominal
| | |
Noun that Noun
| | |
Book that flight

VP Nominal
| |
Verb Det Noun
| | |
Book that flight

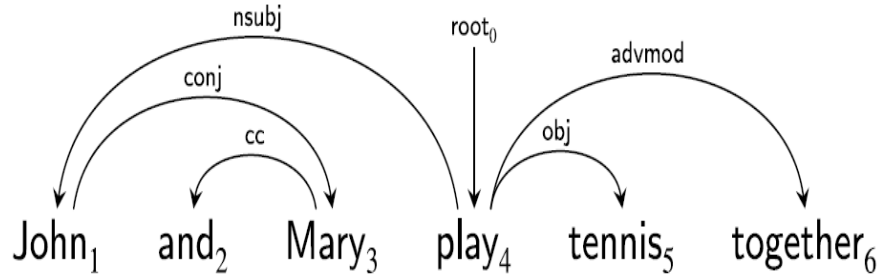
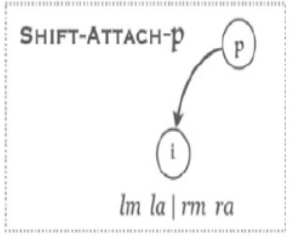
NP
/ | \
Verb Det Nominal
| | |
Book that flight

VP NP
| / | \
Verb Det Nominal
| | |
Book that flight

VP NP
/ | \
Verb Det Nominal
| | |
Book that flight

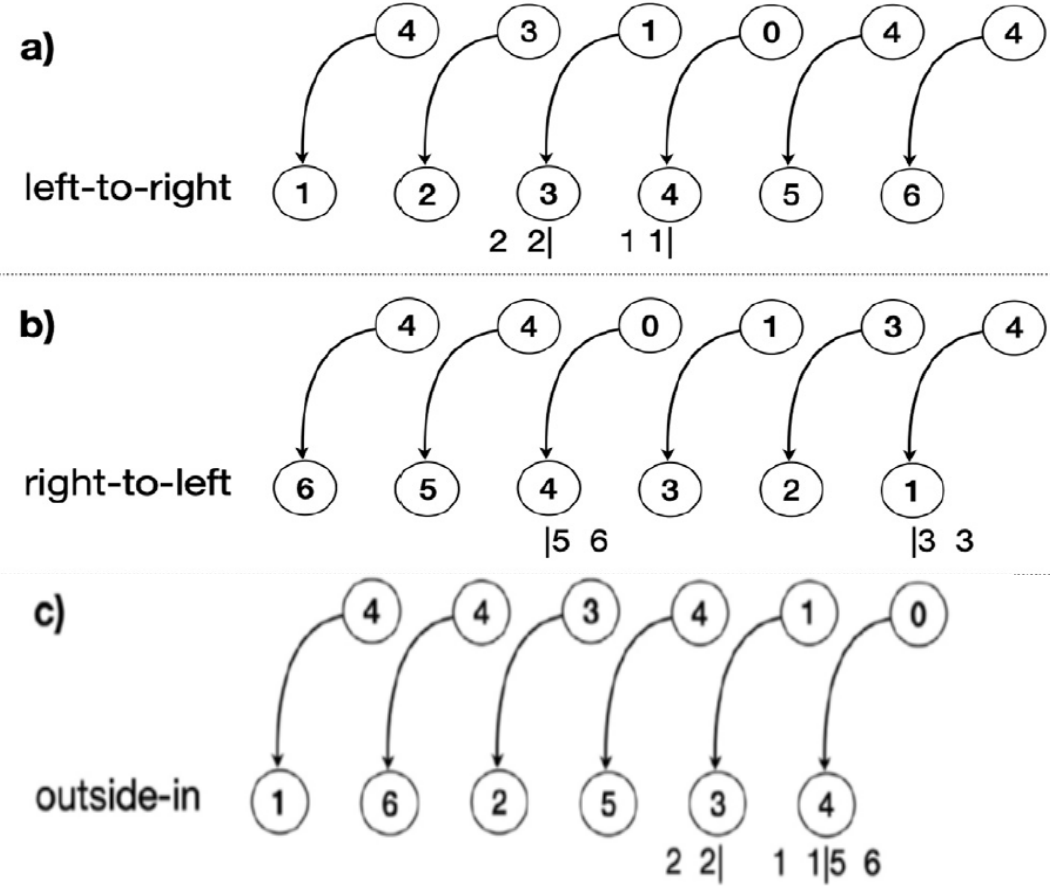
- ❑ “Book that flight” cümlesi sadece uygun dilbilgisi kuralları ile çözümlenir.
- ❑ Buradaki kuralların çoğunda uyumsuzluk görülür.

Geçiş Tabanlı /Transition Based Algoritma ile Dependency Parsing (3-farklı çözümlenme ağacı ile)



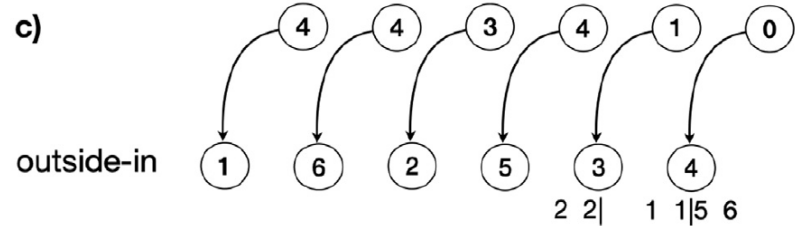
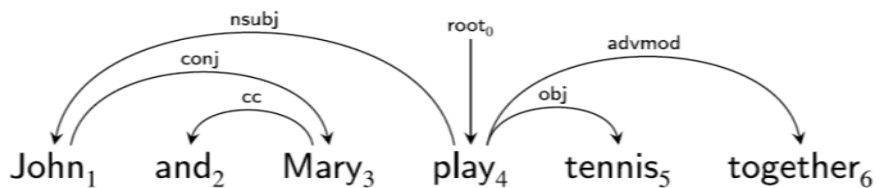
Giriş cümlesinin bağıllık ağacı

Geçiş tabanlı algoritmalar ayrıştırma sürecini ardışık çözümlenmelerle gerçekleştirir; etkili olana uygulanır ve yeni bir durum oluşturulur.

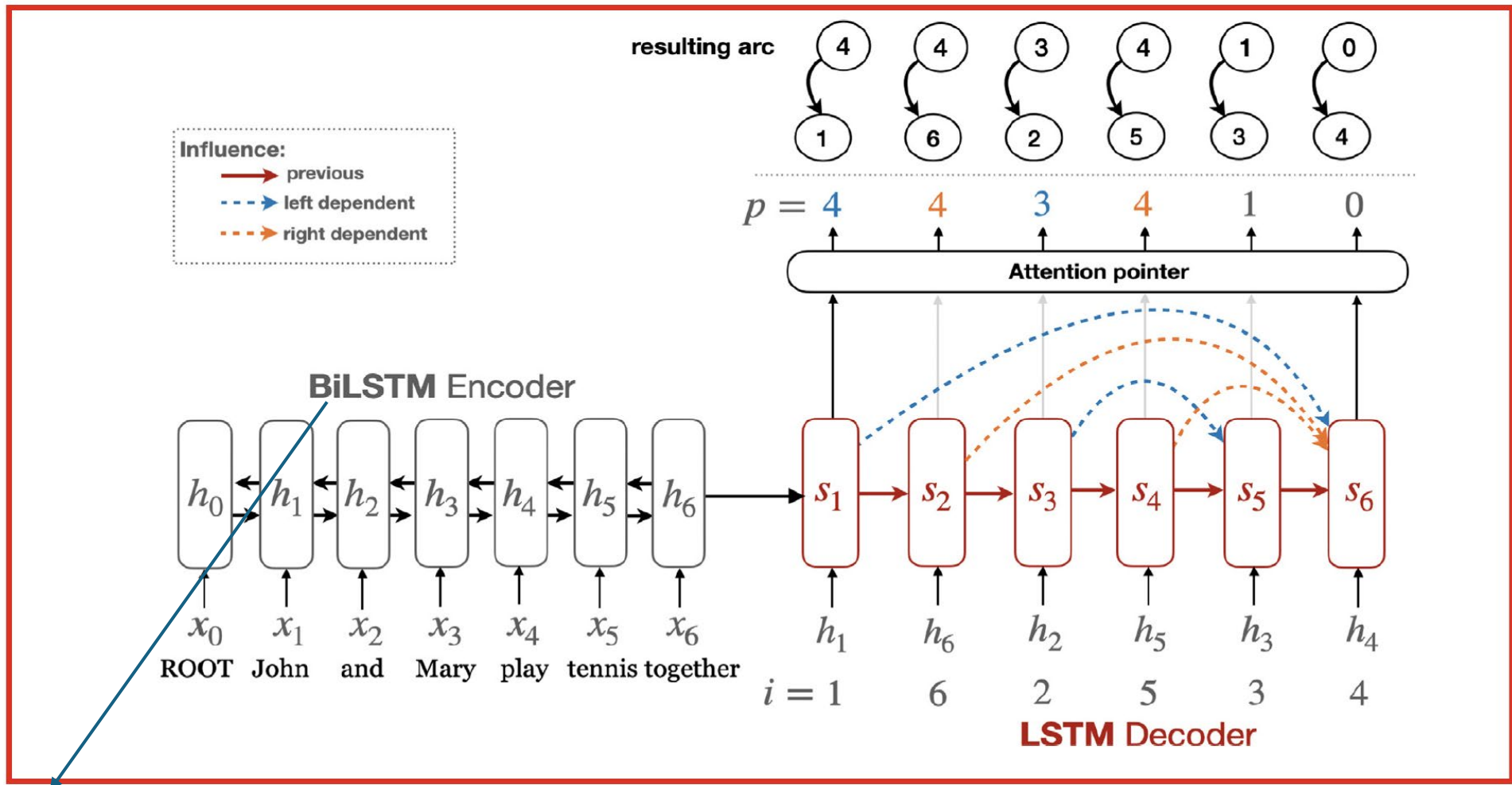


Kaynak: <https://www.sciencedirect.com/science/article/pii/S1566253522001993>

Bağlanma sırası leftmost (lm) , rightmost (rm), last-attached left (la) ve right attached right (ra) bağımlılıkları her bir kod çözücü (decoding) adımında geçerlidir.



ENCODER - DECODER Mimarisi



Bidirectional Long Short-Term Memory Kodlayıcı

LSTM Decoder: (Long Short Term Memory Decoder) kod çözücü

Uzun Kısa Süreli Bellek / LSTM

- Uzun kısa süreli bellek (LSTM), klasik tekrarlayan sinir ağlarında (recurrent neural networks – RNN) karşılaşılan kaybolan gradyan sorununu (vanishing gradient problem) hafifletmeyi amaçlar.
 - ❖ Aktivasyon fonksiyonları sayesinde giriş değerleri belirli bir aralığa indirgenebilir.
 - ❖ Bu aralık genel olarak -1 ve 1 aralığı veya 0 ve 1 aralığı olur.
 - ❖ Küçük bir alana indirgeme yapıldığı için, giriş değerlerinde büyük bir değişim olsa da, aktivasyon fonksiyonunda o kadar büyük bir değişime yol açmayabilir.
- LSTM ile, aralık uzunluğunu göz önüne almadığı için diğer RNN'lere, Hidden Markov modellerine avantajıdır.

<https://github.com/danifg/BottomUp-Hierarchical-PtrNet>

Table 1 — Percentage of long arcs that are leftward.

Language	Treebank	Family	Order	Size	%long	%left
Arabic	PADT	AA	VSO	6.1k	16.62	10.30
Basque	BDT	LI	SOV	5.4k	20.17	44.46
Chinese	GSD	ST	SVO	4.0k	25.31	59.44
English	EWT	IE	SVO	12.5k	16.43	25.49
Finnish	TDT	UR	SVO	12.2k	17.45	26.83
Hebrew	HTB	AA	SVO	5.2k	17.82	20.60
Italian	ISDT	IE	SVO	13.1k	16.20	22.69
Korean	GSD	KO	SOV	4.4k	14.76	78.40
Swedish	Talbanken	IE	SVO	4.3k	19.22	29.51
Turkish	IMST	TU	SOV	3.7k	14.84	70.69

❑ Universal Dependencies v2.6 ile farklı dillerde uygulamalar gerçekleştirilir.

❑ Farklı morfolojik karmaşıklığa ve sözcük bağımlılıklarındaki farklı uzunluklara göre çeşitli dil ailelerinden 10 ağaç bankası incelenmiştir.

Farklı treebank örnekleri

Family = Afro-Asiatic (AA), Indo-European (IE), Koreanic (KO), Language isolate (LI), Sino-Tibetan (ST), Turkic (TU), Uralic (UR)

Order = dominant word order

Size = number of training sentences.

%long = percentage of long arcs (length > 4) in the dev split

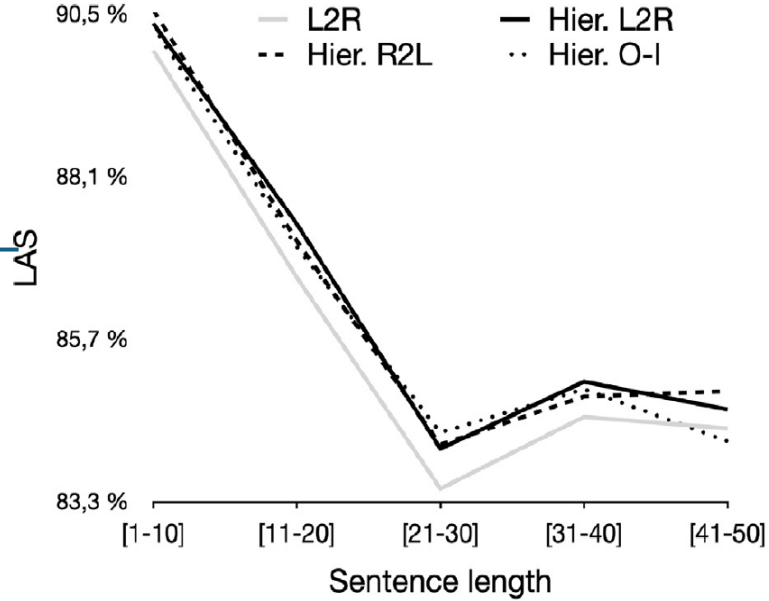
.%left = percentage of long arcs that are leftward.

Her «Transition System» için Bağımlı Bilgi ve Çözümleme Performansına Etkisi

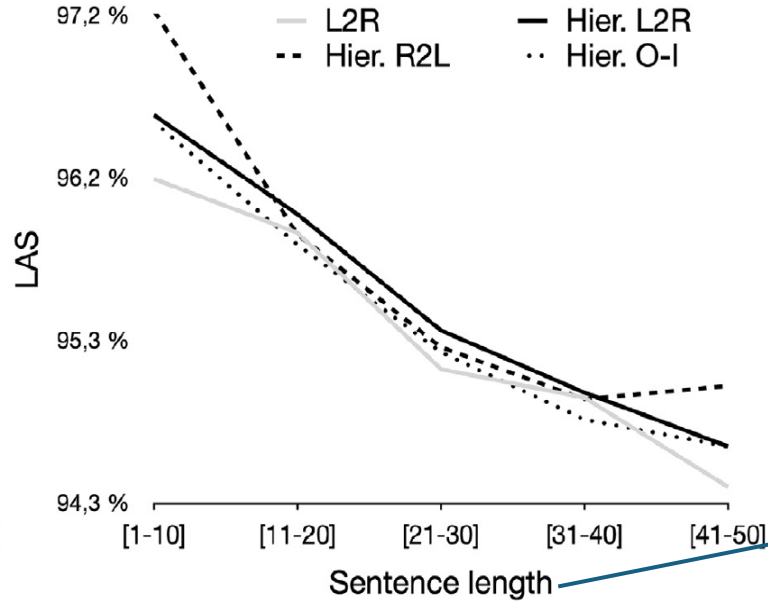
Language	Left2Right		Right2Left		Outside-in	
	all	long	all	long	all	long
Arabic	10.07	1.20	22.20	8.00	14.59	3.75
Basque	6.65	2.29	5.75	2.17	7.10	2.81
Chinese	16.34	6.17	7.99	3.83	12.60	5.34
English	7.02	1.66	4.54	2.54	6.10	2.33
Finnish	7.06	1.69	5.36	2.70	6.61	2.45
Hebrew	10.80	1.69	11.77	5.47	11.69	3.95
Italian	11.15	1.74	8.97	4.88	10.64	3.73
Korean	6.94	2.45	4.64	0.79	5.77	1.61
Swedish	10.63	2.70	7.81	4.18	9.61	3.71
Turkish	5.71	1.85	3.46	0.74	5.07	1.27

- Evrensel Bağımlılıklara (Universal Dependencies) göre ağaçlar oluşturulur.
- Her geçiş sisteminde her bir cümle için mevcut olan bağımlı tüm sözcükler (all) ve uzun aralıklı bağımlılık sayısına (long) göre değerler hesaplanır.
 - ❖ Cümle başına uzun aralıklı bağımlılık sayısının (bağımlılık uzunluğu > 4) sonuçları, kısa cümleler için yapılan hesaplama sonuçlarından çok daha düşük olacaktır.

Average
Labeled
Score



a) Sentence length - UD treebanks



b) Sentence length - PTB with BERT

Tüm geçiş (transition) sistemlerinin **cümle uzunluklarına** göre doğrulukları ve Evrensel Bağımlılık ağaç bankası ve PTB'deki temel ayrıştırıcı (çözümleyici) sonuçları

Penn Tree Bank

Hatalar, cümle uzunluğuna göre t UD veri kümelerinin birleştirilmesi ve BERT tabanlı yerleştirmelerle PTB için değerlendiriliyor. L2R, R2L ve dışarıdan içeriye (O-I) geçiş sistemlerinin cümle uzunluklarına göre doğruluğu ve sırasıyla UD ağaç bankaları ve PTB'deki temel ayrıştırıcı (çözümleyici) görülmektedir.

UD veri kümelerinde, cümle uzunluğundan bağımsız olarak doğruluk kazanımları görülmektedir. Cümleler daha uzun olduğunda iyileştirmeler daha yüksek olma eğilimindeyken, uzunluk 40 'dan büyük olduğunda bunlar daralıyor

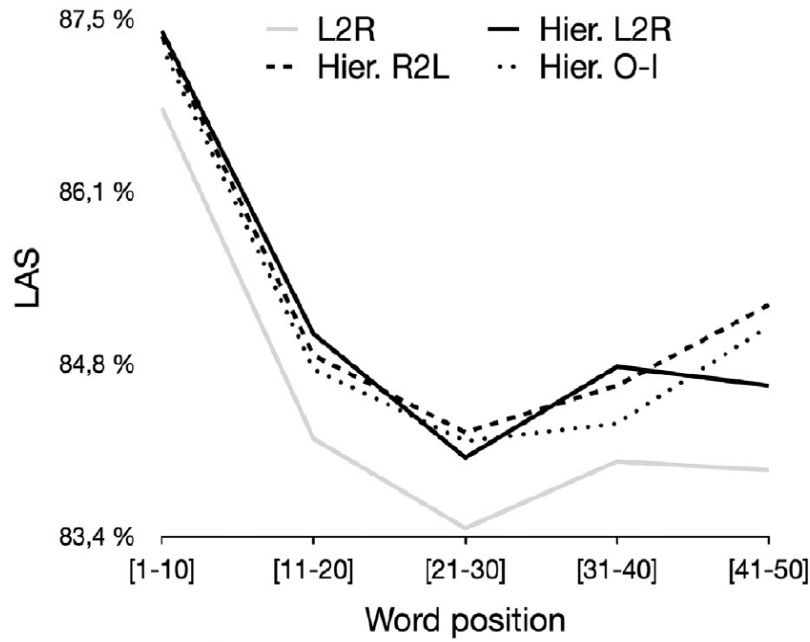
PTB'de, yeni yaklaşımlar yalnızca uzun cümlelerde değildir, en kısa olanlarda da temel çizgi geride kalmaktadır.

Hatalar, cümle uzunluğuna göre tüm Evrensel Bağımlılıklar veri kümesinin birleştirilmesi ve BERT tabanlı yerleştirmelerle PTB için karakterize edilmiştir.

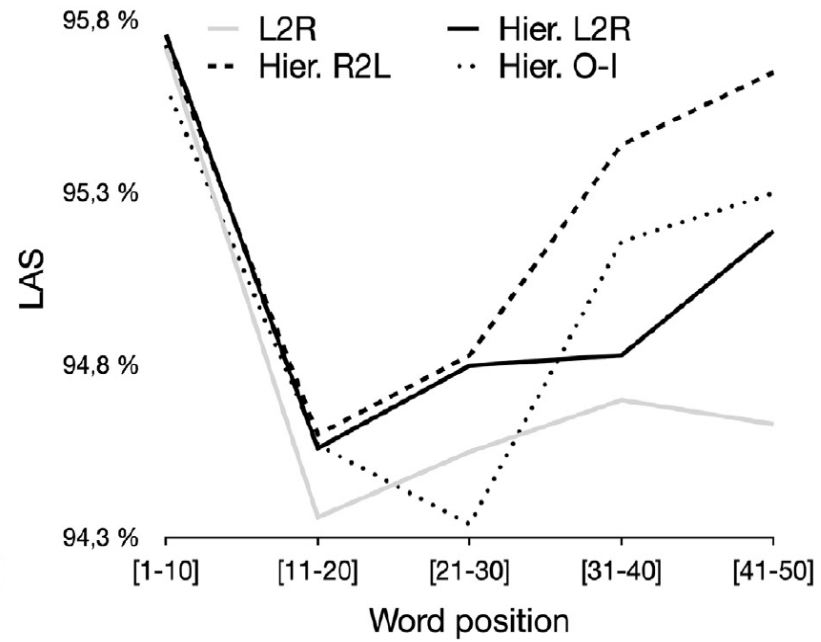
Üç geçiş sisteminin cümle uzunluklarına göre doğruluğu ve sırasıyla Evrensel Bağımlılıklar ağaç bankaları ve PTB'deki temel ayrıştırıcı görülmektedir.

R2L çözümleyicinin en uzun cümlelerde en yüksek performansı gösteriyor. D-I algoritmanın UD ağaç bankalarında uzunluğu 40'tan büyük cümlelerde ve PTB'de uzunluğu 31 ile 40 arasında olan cümlelerde düşüyor.

Average
Labeled
Score



c) Word position - UD treebanks



d) Word position - PTB with BERT

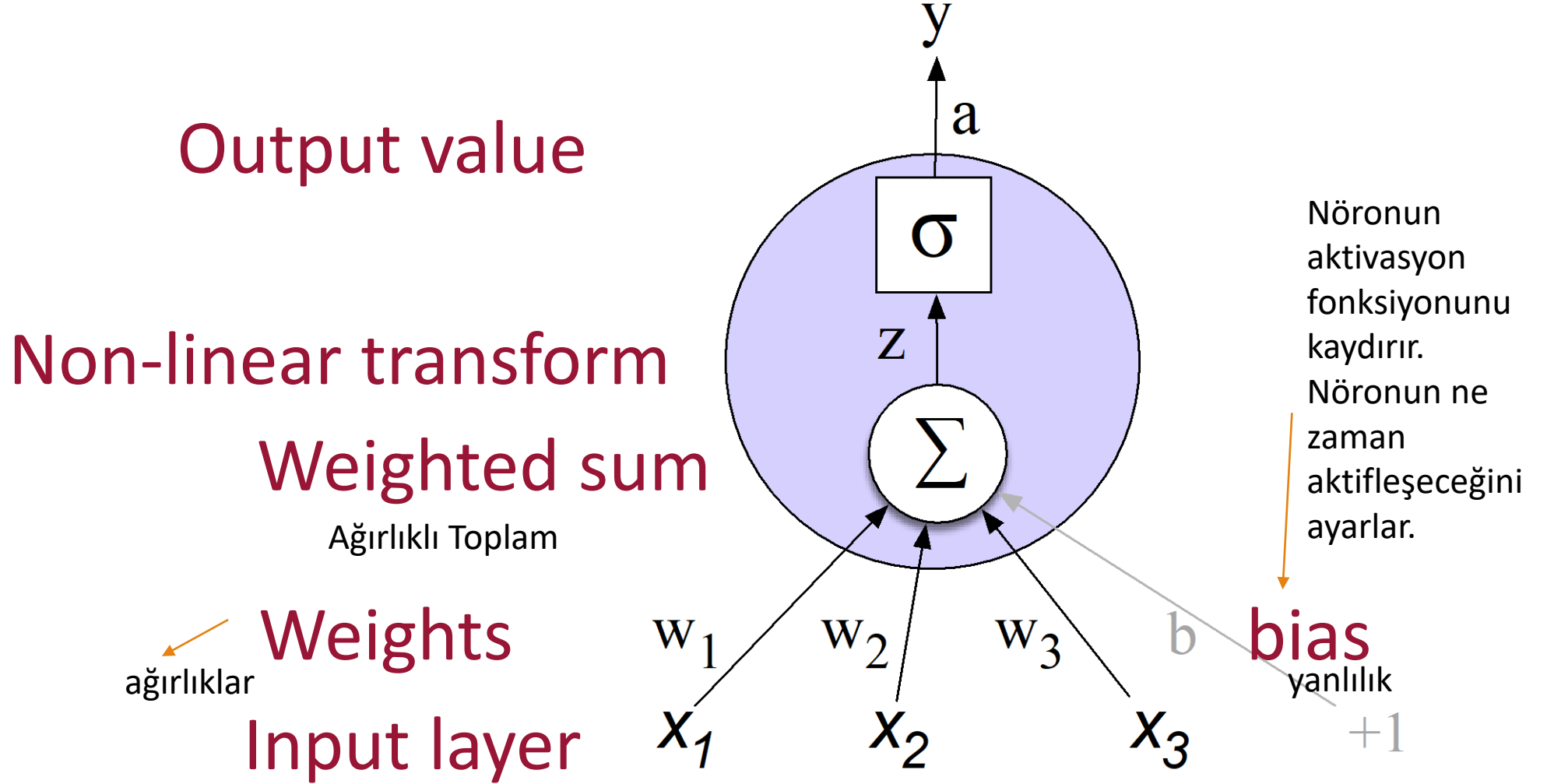
Tüm geiş (transition) sistemlerinin **kelime yerleřimlerine** gre doęrulukları ve Evrensel Baęımlılık aęa bankası ve PTB'deki temel ayrıřtırıcı (zmleyici) sonuları

- ❑ Hata yayılımının azaltılması /reduction of error propagation), **cmle iindeki kelime konumlarına gre** LAS deęiřimleri daha aık bir Őekilde grlebilir.
- ❑ Soldan saęa geiř sistemi, cmlenin son konumlarında szcklere yapılan ekler zerinde (hata yayılımından en ok etkilenen) en yksek doęrulukları elde ediyor.
- ❑ Saędan sola ve dıřarıdan ieriye yaklařımların cmlenin sonuna yakın kelimelerde soldan saęa algoritmasından nemli lde daha iyi performans gsteriyor.
- ❑ O-I modelinde PTB'de orta konumlardaki kelimelerde (21'den 30'a) ve UD veri kmelerinde, fazla nemli olmayarak 31'den 40'a uzunlukta doęrulukta nemli bir dřř grlmektedir.

Simple Neural
Networks and
Neural Language
Models

Units in Neural Networks

Neural Network



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

Ağırlıklar ve bias eğitim sırasında otomatik olarak ayarlanır.

Yapay Sinir Ağlarında Öğrenme Süreci: Ağırlıklar, Girdiler ve Bias

Öğrenme forward propagation (ileri yönlü yayılım) ve backpropagation (geri yayılım) ile gerçekleşir.

Girişlerin ağırlıklı toplamı ile v_i yanılma (bias) değeri toplanır.

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

$$z = w \cdot x + b$$

z yazılması yerine lineer olmayan bir aktivasyon fonksiyonu kullanılabilir.

$$y = a f(z)$$

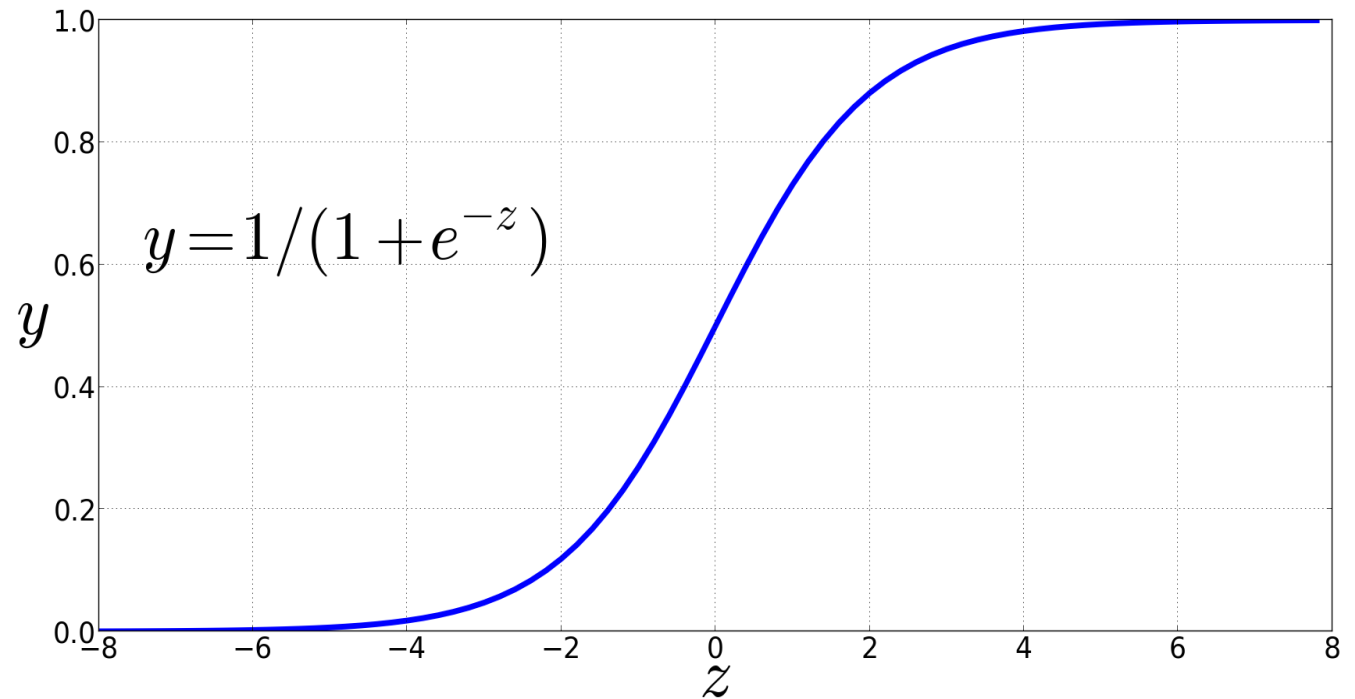
Non-Linear Activation Functions

f olarak seçilen non linear fonksiyonlardan biri sigmoid fonksiyonu logistic regression için kullanılır.

Sigmoid

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

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



Çıktıya **aktivasyon** değeri denir. 0 ile 1 arasında normalize edilmiştir.

Sayısal Örnek

$$w = [0.2, 0.3, 0.9]$$

$$b = 0.5$$

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

olsun. Sonuç:

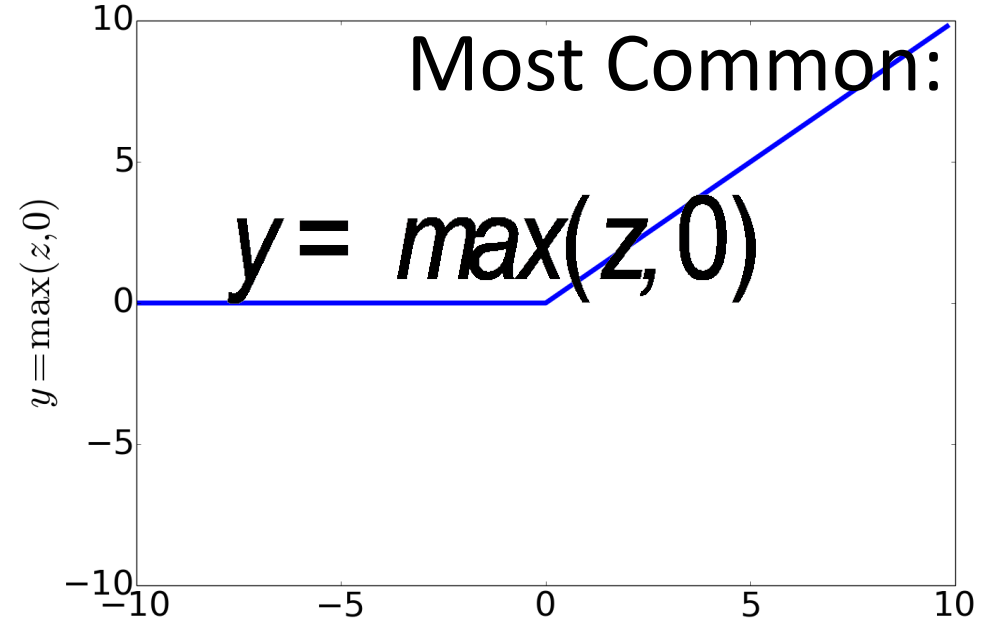
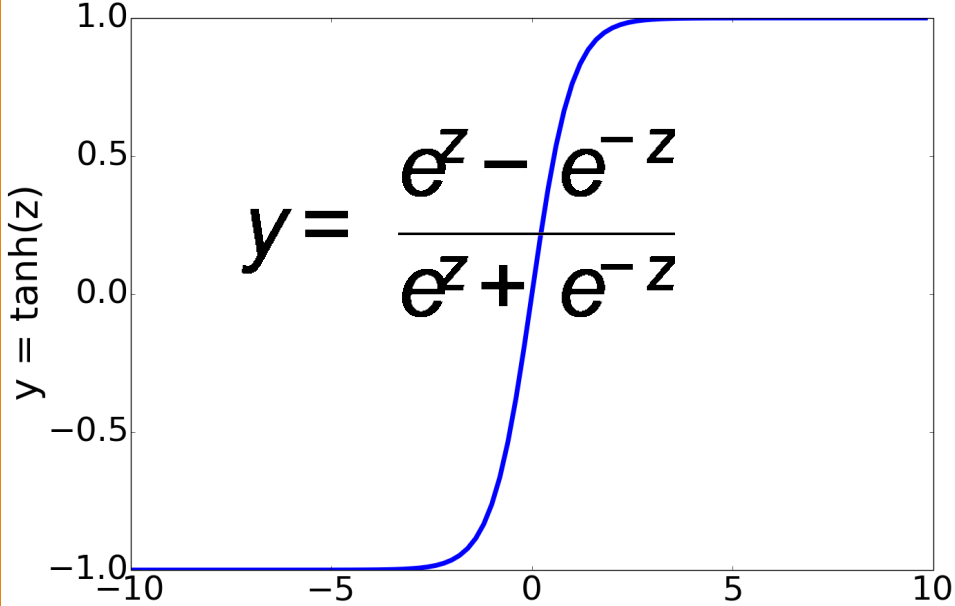
$$y = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)}} = \frac{1}{1 + e^{-(.5 * .2 + .6 * .3 + .1 * .9 + .5)}} = \frac{1}{1 + e^{-0.87}} = .70$$

Non-linear aktivasyon fonksiyonu olarak Sigmoid fonksiyonu fazla tercih edilmez.

Diğer Non-Linear Aktivasyon Fonksiyonları

tanh

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



ReLU Rectified Linear Unit /doğrultulmuş doğrusal birim

- tanh, sigmoid fonksiyonuna yakın bir fonksiyondur. Aktivasyon fonksiyonu olarak daha fazla kullanılır.
- tanh fonksiyonunun aralığı -1 ile 1 arasındadır.
- tanh fonksiyonunun bu özelliği sonuca yakınsama da daha sorunsuz çıktı üretebilir. Ortalamaya yakın değerlere eşleşme daha yüksek olasılıklıdır.
- En basit ve oldukça yaygın kullanılan aktivasyon fonksiyonu ise rectified linear unit (ReLU) fonksiyonudur. z değerinin pozitif olması durumunda (akisi durumda 0 olur) kullanılır. En fazla tercih edilen aktivasyon fonksiyonudur.

Introduction to Transformers

Transformers

LLM'ler
transformatörlerden
oluşur.

Transformatör, ileri
beslemeli ağ gibi belirli bir
ağ mimarisi türüdür.

very approximate timeline

1990 Static Word Embeddings

2003 Neural Language Model

2008 Multi-Task Learning

2015 Attention

2017 Transformer

2018 Contextual Word Embeddings and Pretraining

2019 Prompting

Problem with static embeddings (word2vec)

Word and vector are static. The embedding for a word doesn't reflect how its meaning changes in context.

The chicken didn't cross the road because it was too tired

What is the meaning represented in the static embedding for "it"?

Contextual Embeddings

- ❑ Intuition: a representation of meaning of a word should be different in different contexts!
- ❑ Contextual Embedding: each word has a different vector that expresses different meanings depending on the surrounding words
- ❑ How to compute contextual embeddings?
Attention

Contextual Embeddings

The chicken didn't cross the road because it

What should be the properties of "it"?

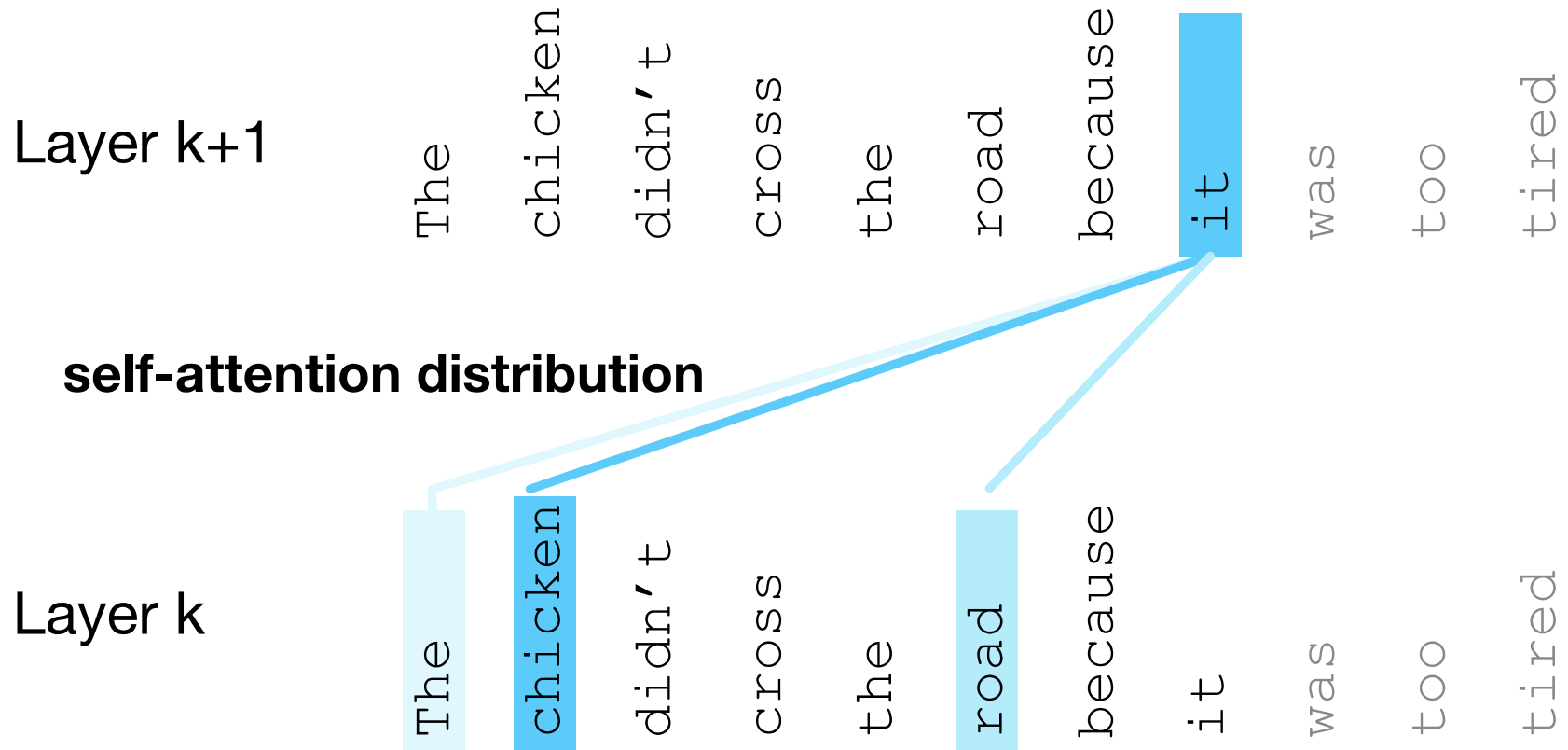
The chicken didn't cross the road because it was too **tired**

The chicken didn't cross the road because it was too **wide**

At this point in the sentence, it's probably referring to either the chicken or the street

Intuition of Attention

columns corresponding to input tokens

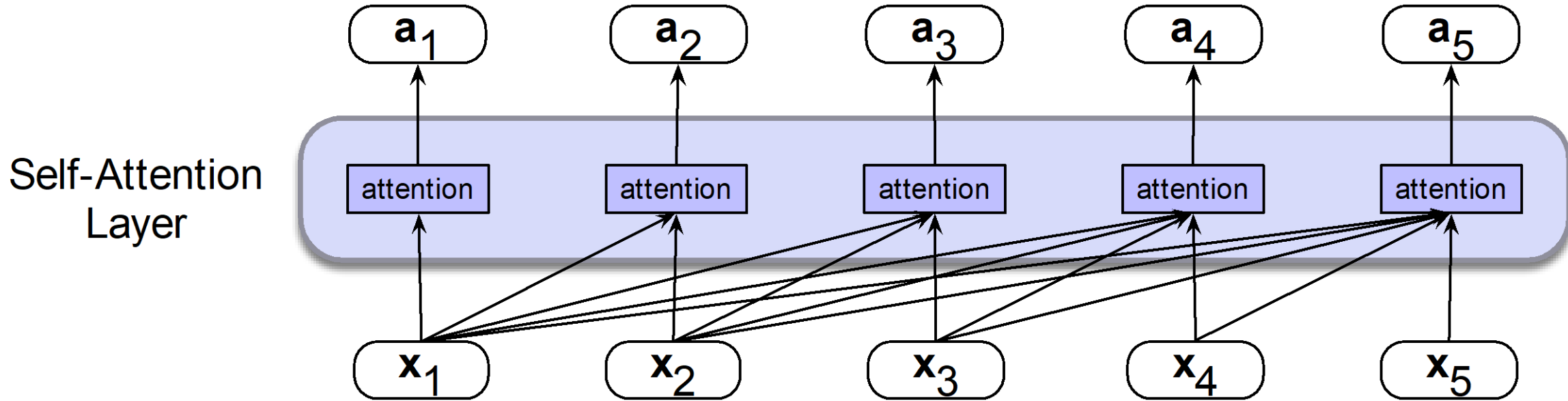


Attention definition

A mechanism for helping compute the embedding for a token by selectively attending to and integrating information from surrounding tokens (at the previous layer).

More formally: a method for doing a weighted sum of vectors.

Attention is left-to-right



A sum of prior words weighted by their similarity with the current word

Given a sequence of token embeddings:

$$\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3 \quad \mathbf{x}_4 \quad \mathbf{x}_5 \quad \mathbf{x}_6 \quad \mathbf{x}_7 \quad \mathbf{x}_i$$

Produce: \mathbf{a}_i = a weighted sum of \mathbf{x}_1 through \mathbf{x}_7 (and \mathbf{x}_i)

Weighted by their similarity to \mathbf{x}_i

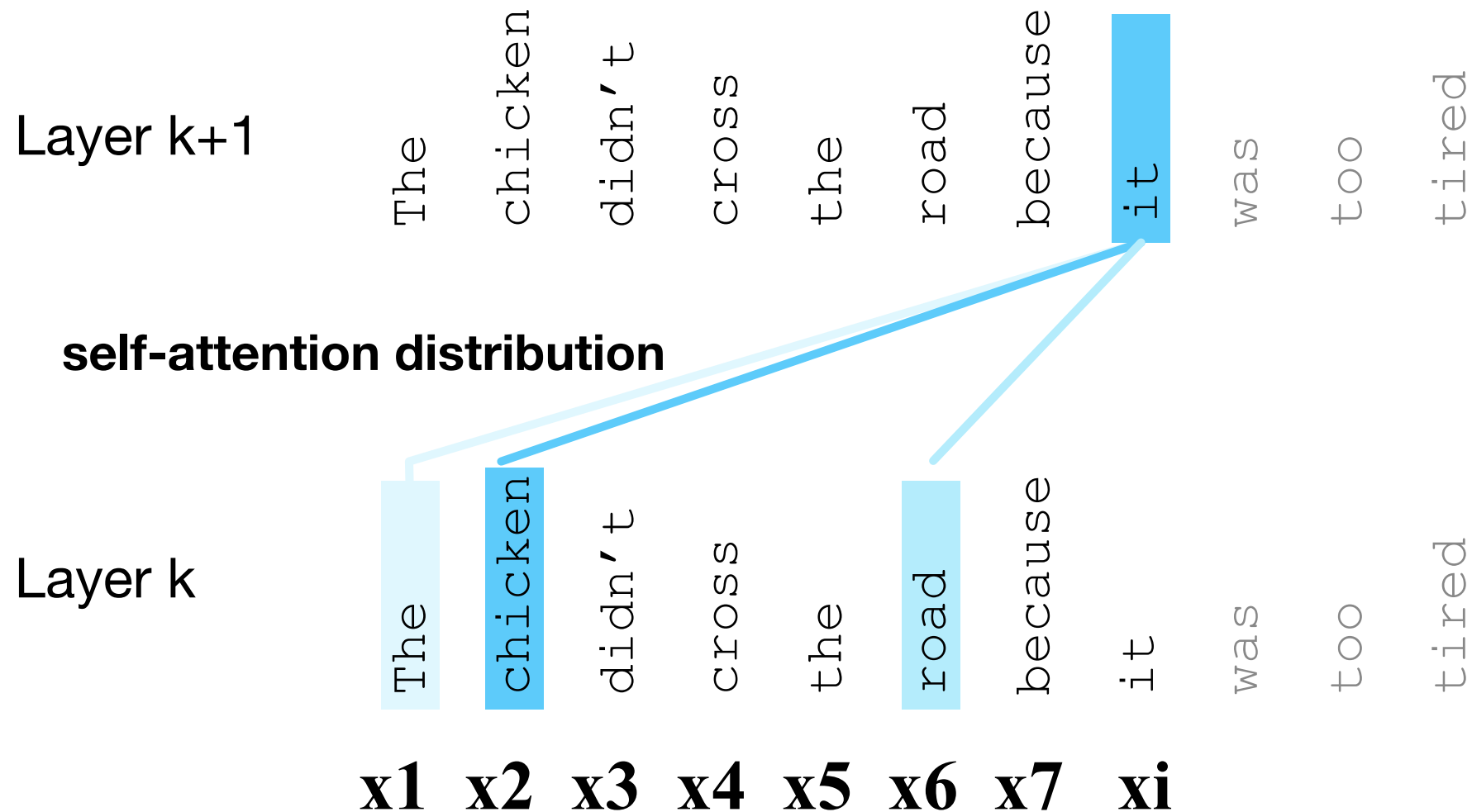
$$\text{score}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$$

$$a_{ij} = \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j)) \quad \forall j \leq i$$

$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{x}_j$$

Intuition of Attention

columns corresponding to input tokens

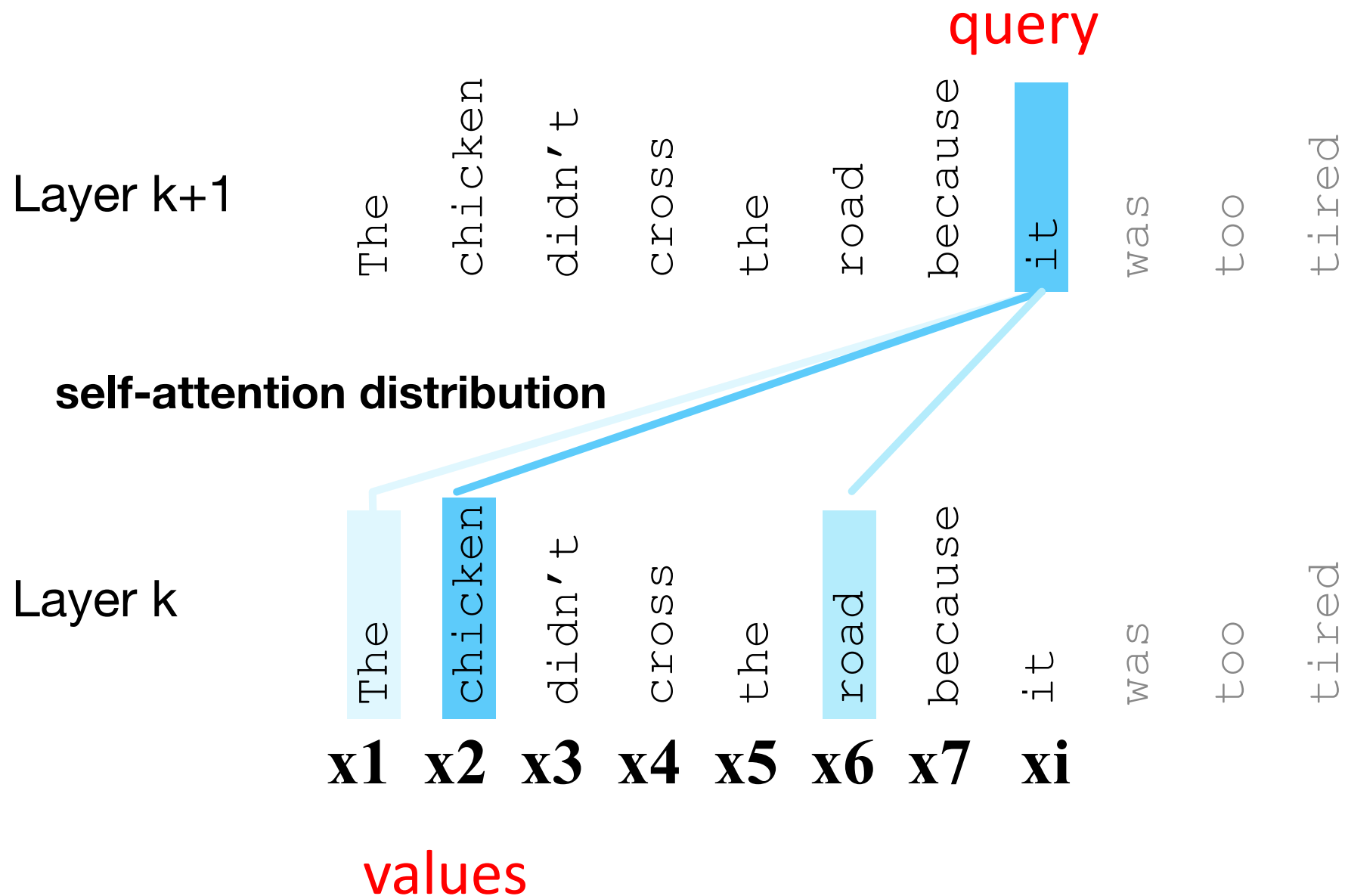


An Actual Attention Head: slightly more complicated

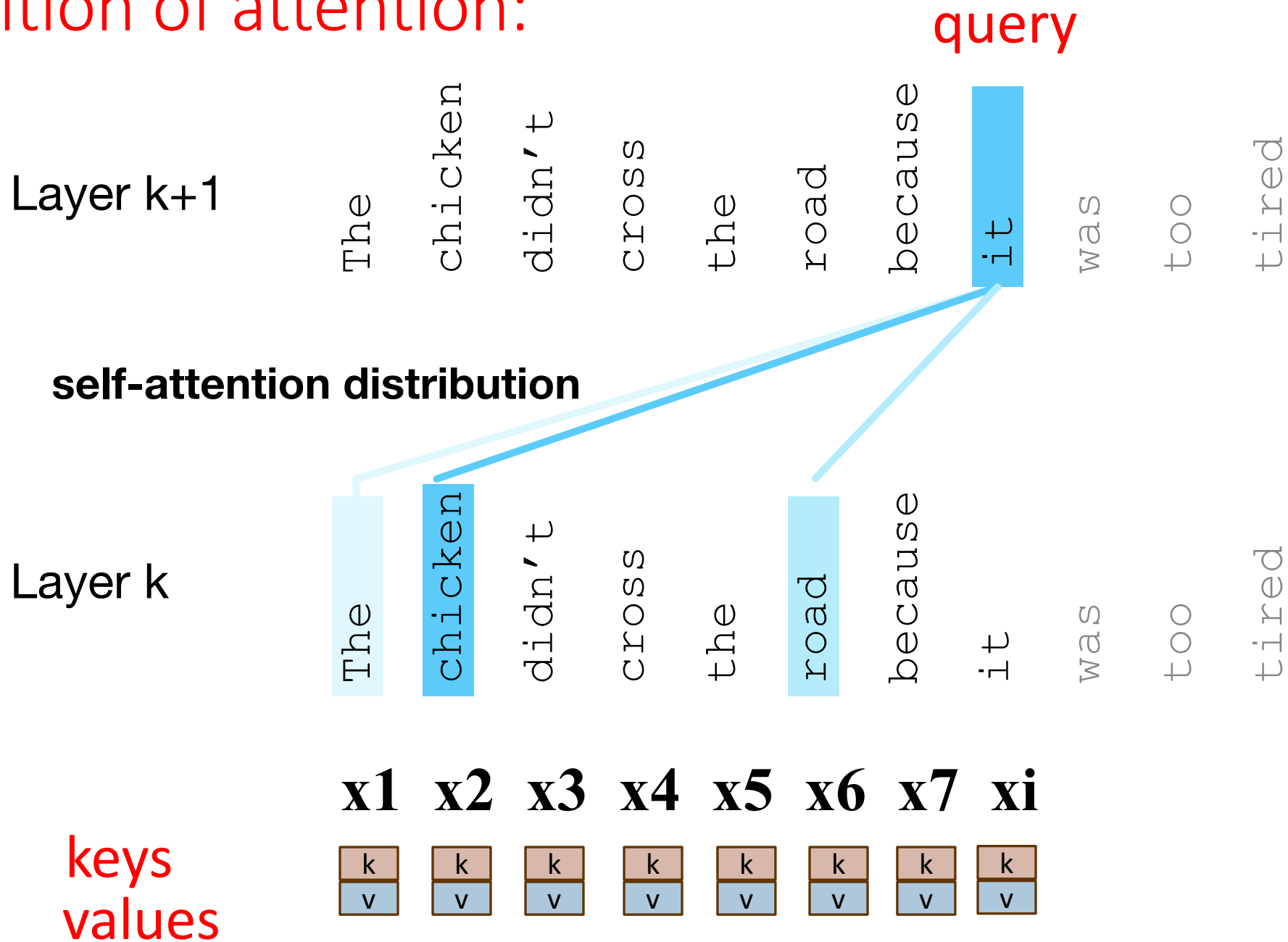
High-level idea: instead of using vectors (like x_i and x_4) directly, we'll represent 3 separate roles each vector x_i plays:

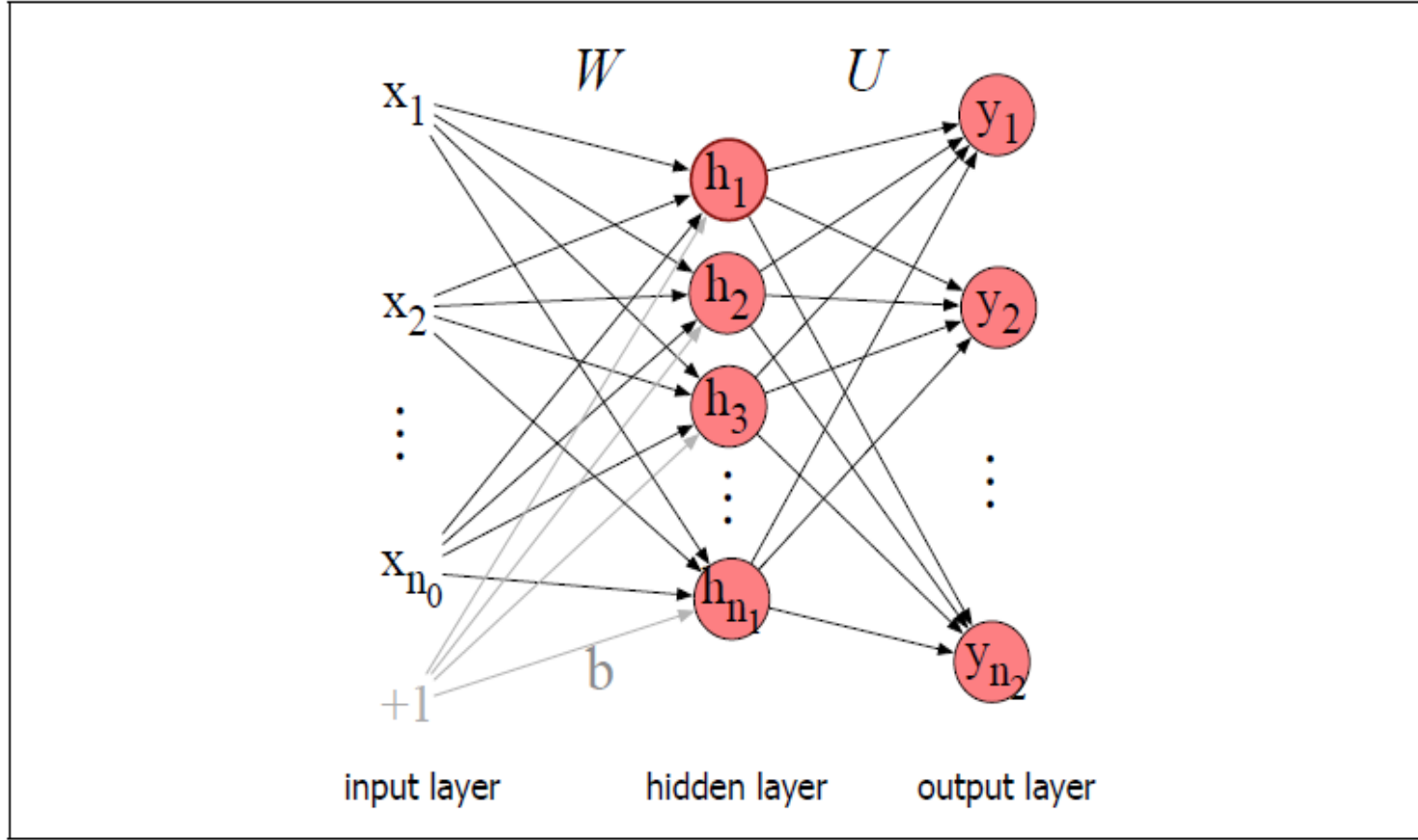
- **query**: *As the current element* being compared to the preceding inputs.
- **key**: *as a preceding input* that is being compared to the current element to determine a similarity
- **value**: a value of a preceding element that gets weighted and summed

Attention intuition



Intuition of attention:





2-düzeyle ileri yayımlı ağ
Gizli (hidden) katman ve çıktı katmanı
Giriş katmanı, katman olarak sayılmaz.

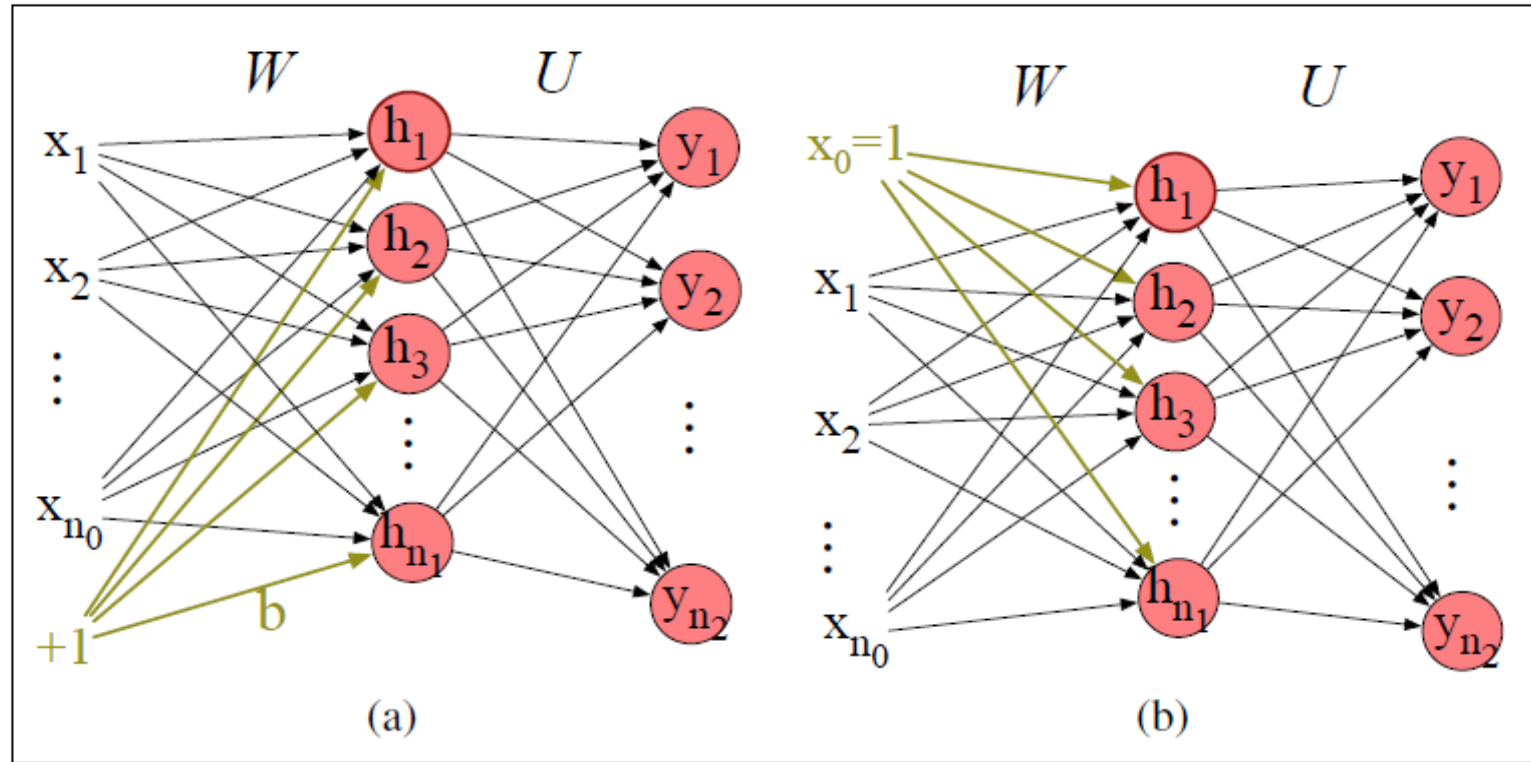
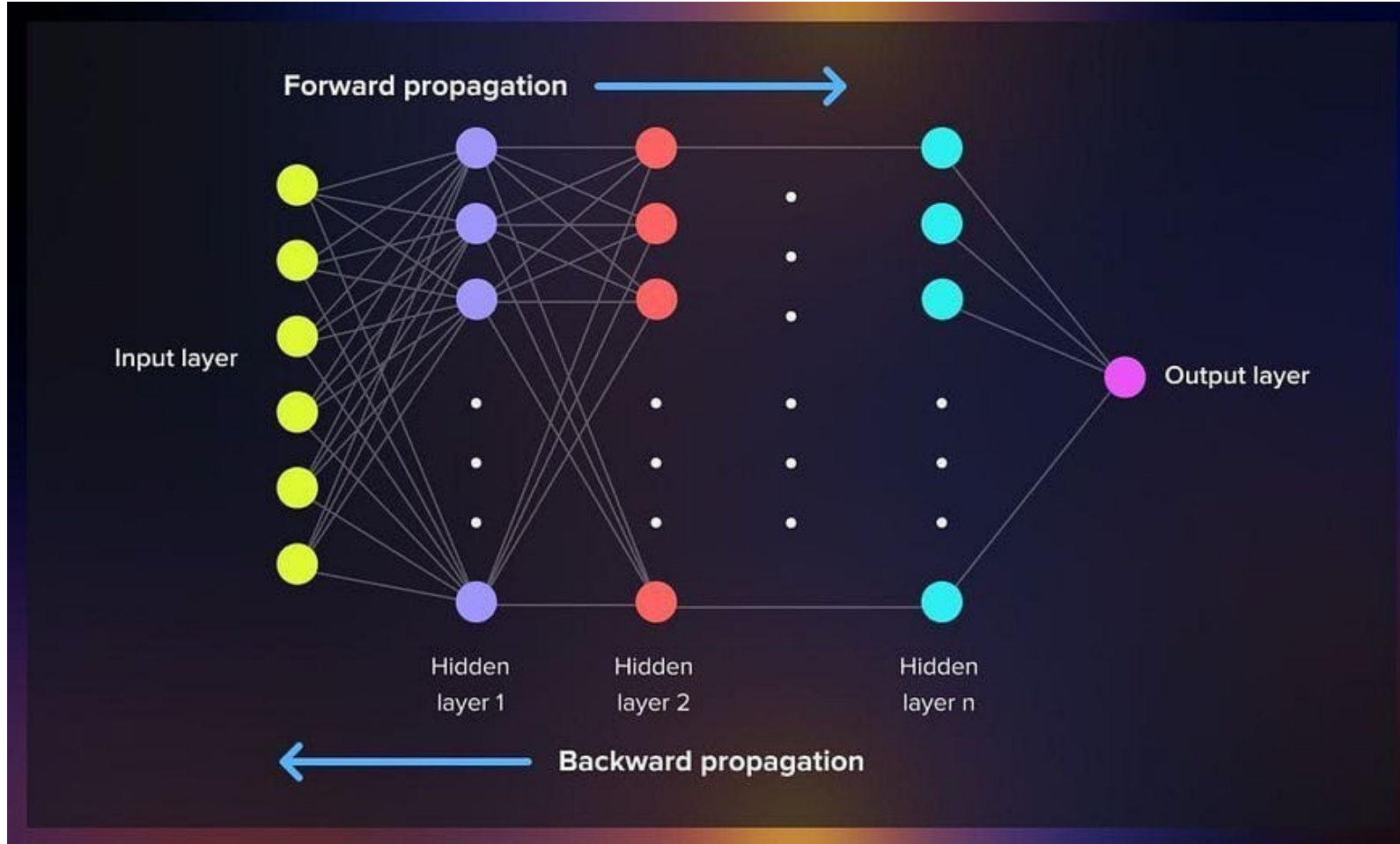


Figure 7.9 Replacing the bias node (shown in a) with $x_0=1$ (b)

bias düğümü, x_0 düğümü ile yer değiştirdi(b)

Yapay Sinir Ağları

İleri Yönlü Yayılım ve Geriye Doğru Yayılım



- ❑ İleri yönlü yayılım, girdi verilerinin ağ katmanları boyunca ilerleyerek nihai çıktıya dönüştüğü süreçtir.
 - ❖ Bu süreçte her bir nöron, ağırlıklar, bias ve aktivasyon fonksiyonları kullanarak hesaplamalar yapar.
- ❑ Geriye doğru yayılım, hataları tespit edip, bu hataları minimize etmek amacıyla ağırlıkların güncelleme sürecidir.

Duygu Analizi Örneği

