

# Konuşma Tanıma için Yapay Öğrenme

Murat SARAÇLAR

Boğaziçi Üniversitesi  
Elektrik-Elektronik Mühendisliği Bölümü

Murat SARAÇLAR

<http://busim.ee.boun.edu.tr/~murat/>



- 1994 Bilkent U. EE (BS)
- 2000 Johns Hopkins U. (MS, PhD)
- 2000-2005 AT&T Labs – Research
- 2005- Boğaziçi U. EE
- 2011-2012 Google Inc.
- 2012-2013 IBM T.J. Watson Research Center
- 2013- Özgür Deniz 😊

# Özet

- Konuşma Tanıma
- İstatistiksel Modeller
- Yapay Öğrenme
- Derin Öğrenme

# Konuşma Tanıma

Tanımlar

Yaklaşımlar

Uygulamalar

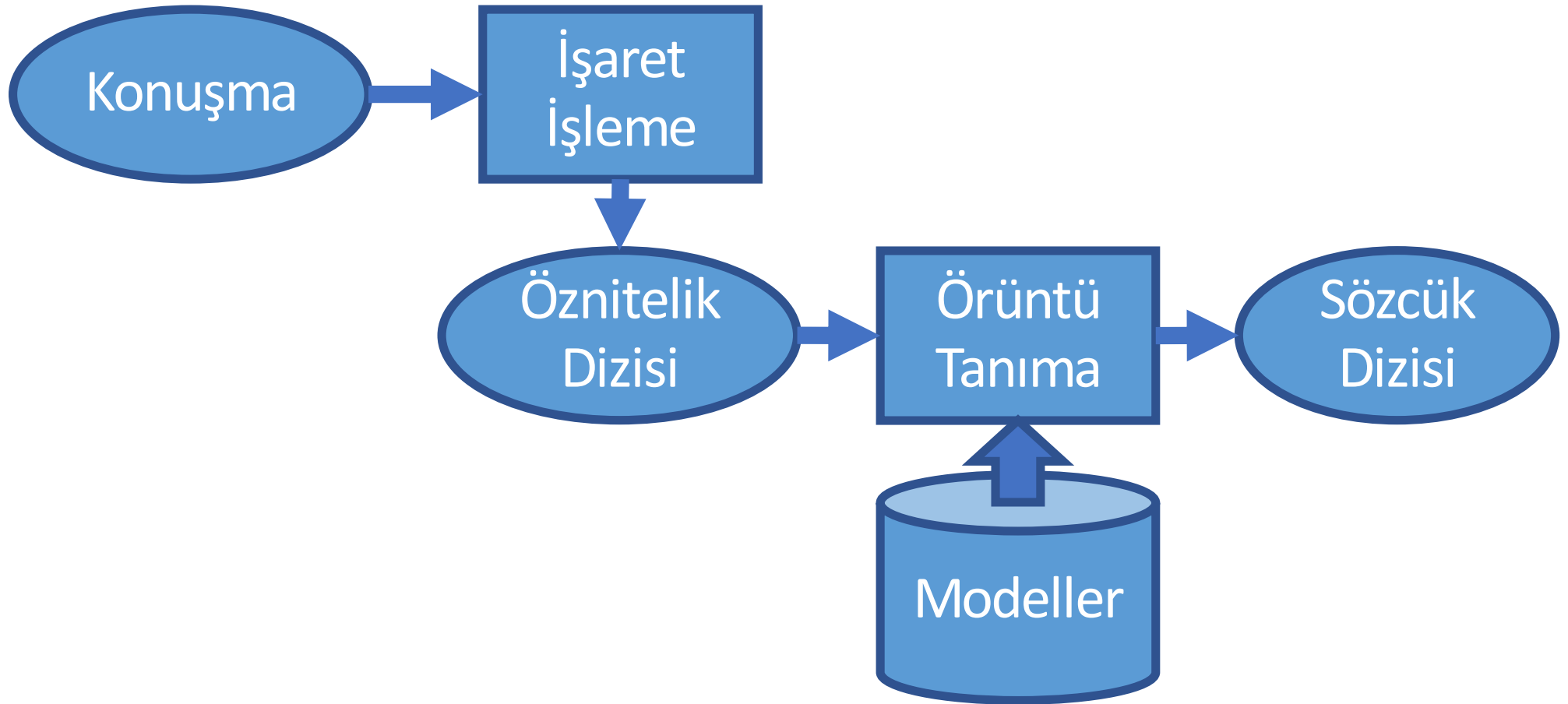
# Konuşma Tanıma: Tanımlar

- (Otomatik) Konuşma Tanıma:  
= (Automatic) Speech Recognition
- Konuşmanın yazıya dökülmesi  
(yazılandırma = transcription)

# Konuşma Tanıma: Girdi ve Çıktılar

- Girdi: Konuşma «sinyali»
  - Sayısallaştırılmış (8-16 kHz, 8-16 bit)
- Ara gösterim: Öznitelikler dizisi
  - Konuşmada bilgi sinyalin zaman-frekans içeriğindedir
  - İşaret işleme yöntemleriyle elde edilen öznitelikler
  - Vektör dizisi (yaklaşık olarak saniyede 100 tane)
- Çıktı: Kelime dizisi
  - Varsayım: sonlu dağarcık

# Konuşma Tanıma: Sistem



# Konuşma Tanıma: Yaklaşımlar

- Örüntü eşleştirme
- İstatistiksel modelleme
- Derin öğrenme



# Konuşma Tanıma: Uygulamalar

- Dikte (bilgisayar, cep telefonu, ...)
- İnsan bilgisayar (makina) arayüzü
- Telefon üzerinden etkileşim (örn. müşteri hizmetleri)
- Ses içeriğine erişim
- Akıllı asistanlar
- Sesli çeviri

# Konuşma Tanıma için İstatistiksel Yaklaşımlar

Tanımlar

Modeller

Yöntemler

# İstatistiksel Konuşma Tanıma:

## Biraz Matematiksel Notasyon

- Girdi: Akustik öznitelik vektör dizisi (A)

$$\begin{aligned}t &= 1, \dots, T \\ a_t &\in \mathbb{R}^d \\ A &= a_1, a_2, \dots, a_T\end{aligned}$$

- Çıktı: Kelime dizisi (W)

$$\begin{aligned}i &= 1, \dots, N \\ w_i &\in \mathcal{V} \\ W &= w_1, w_2, \dots, w_N\end{aligned}$$

# İstatistiksel Konuşma Tanıma

- En olası sözcük dizisi

$$\hat{W} = \arg \max_W P(W|A)$$

- Bayes kuralı yardımıyla

$$\hat{W} = \arg \max_W \frac{P(A|W)P(W)}{P(A)}$$

- Konuşma tanımanın temel denklemi:

$$\hat{W} = \arg \max_W P(A|W)P(W)$$

## $P(W)$ : Dil Modeli

- Dil modeli bir dildeki tüm cümlelere (kelime dizilerine) bir olasılık atar.
- En genel haliyle  $P(W) = \prod_{i=1}^N P(w_i | w_1, \dots, w_{i-1})$
- Tabii ki  $i$  arttıkça bütün bu koşullu olasılıkları belirlemek mümkün olmayacaktır.
- Çözüm: Geçmiş gruplamak  $h_i = \Phi(w_1, \dots, w_{i-1})$
- Böylece  $P(W) = \prod_{i=1}^N P(w_i | h_i)$

## $P(A|W)$ : Akustik Model

- Akustik model bir akustik öznitelik vektör dizisinin bir sözcük dizisine karşılık gelme (koşullu) olasılığını verir.
- Bir dildeki tüm sözcük dizileri için ayrı bir olasılık modeli kestirmek mümkün değildir.
- Kısıtlı dağarcıklar haricinde tüm sözcükler için bile ayrı bir olasılık modeli kestirmek mümkün olmayabilir.
- Bu nedenle akustik modelleme için sözcüklerden küçük birimler kullanılır.

# Söyleyiş (Telaffuz, Sesletim) Modeli: Sözcüklerden Sesçiklere

- Akustik modelleme için tercih edilen birimler sesçiklerdir (phone/phoneme).
- Sesçikler bağlam içinde modellenir. (örn. triphone)
- Sözcüklerden sesçiklere geçiş için bir söyleniş sözlüğü (pronunciation lexicon) kullanılır.
- Doğal karşılıklı konuşma için olasılıksal modeller önerilmiştir.

## arg max : En olası sözcük dizisini bulma

- Arama uzayı bir dildeki tüm sözcük dizilerini içermektedir.
- Bu uzay sonsuz olduğuna göre arama işlemi verimli bir şekilde yapılmalıdır.
- Eğer arama uzayı uygun bir şekilde (örn. sonlu durum içeren bir çizge) düzenlenirse dinamik programlama kullanılabilir.

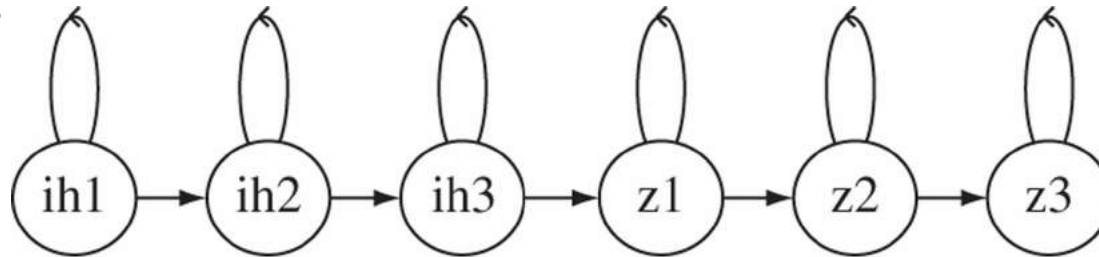


## $P(W)$ : (Görünür) Markov modelleri

- Markov varsayımı: Gelecek sadece şimdiki duruma bağlıdır, geçmişten bağımsızdır.  $P(W) = \prod_{i=1}^N P(w_i|h_i)$
- Uzak geçmişi unutursak  $n$ -gram:  $h_i = \{w_{i-n+1}, \dots, w_{i-1}\}$ 
  - *Unigram*:  $P(W) = \prod_{i=1}^N P(w_i)$
  - *Bigram*:  $P(W) = \prod_{i=1}^N P(w_i|w_{i-1})$
  - *Trigram*:  $P(W) = \prod_{i=1}^N P(w_i|w_{i-2}, w_{i-1})$
- Model parametreleri durumlar arasındaki geçiş olasılıklarıdır.
- Ağırlıklı sonlu durum makinasıyla gerçekleştirilebilir.

## $P(A|W)$ : Saklı Markov Modelleri (SMM)

- Saklı Markov modellerinde durumlar saklıdır.
- Yani hangi gözlemin hangi durumdan geldiği belli değildir.
- Her bir sesçik soldan sağa bir SMM ile modellenir.
- Sözcük modelleri sesçik modellerinin ardarda eklenmesiyle elde edilir.



Copyright © 2011 Pearson Education, Inc. publishing as Pearson Hall

## $P(A|W)$ : SMM Olasılıkları

- Söyleyiş modeli ve saklı Markov modellerinin eklenmesiyle

$W \rightarrow S = s_1, \dots, s_M$  (sonlu durum dönüştürücüsü)

$$P(A|\Lambda(S)) = \sum_Q P(A, Q|\Lambda(S))$$

$$P(A, Q|\Lambda(S)) = \prod_{t=1}^T P(q_t|q_{t-1}; \Lambda(S)) p(a_t|q_t; \Lambda(S))$$

- $P(q_t|q_{t-1})$ : Durumlar arası geçiş olasılıkları

## $P(a_t|q_t)$ : Durum Çıktı Olasılık Dağılımı

- Gauss Dağılımı (Normal dağılım)

$$p(a_t|q_t) = \mathcal{N}(a_t; \mu, \Sigma)$$

- Gauss Karışım Modelleri

$$p(a_t|q_t) = \sum_k w_k \mathcal{N}(a_t; \mu_k, \Sigma_k)$$

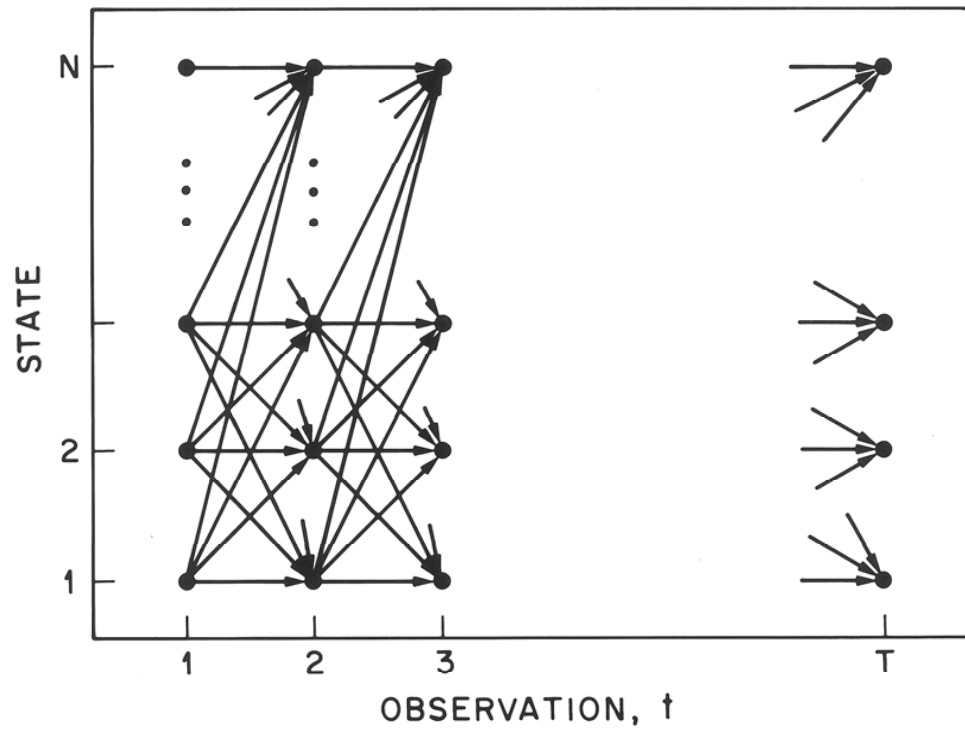
- Yapay Sinir Ağları

# Modellerin Birleştirilmesi

- Ağırlıklı Sonlu Durum Makinaları ve Dönüştürücüleri
  - H: Saklı Markov Modeller (SMM'lerden durumlara)
  - C: Bağlam Modeli (sesçiklerden SMM'lere)
  - L: Söyleyiş sözlüğü (sözcüklerden sesçiklere)
  - G: Dil modeli (sözcükler)
- Arama uzayı: HoCoLoG
- Bu uzayı arama için verimli bir hale getirmek mümkündür.  
(det, min, push)

# En iyi durum dizisinin bulunması

- Kafes yapısı (trellis)



## Viterbi Algoritması - 1

$$\delta_t(i) \equiv p(q_1^{t-1}, q_t = s_i, a_1^t)$$

1. Başlangıç:

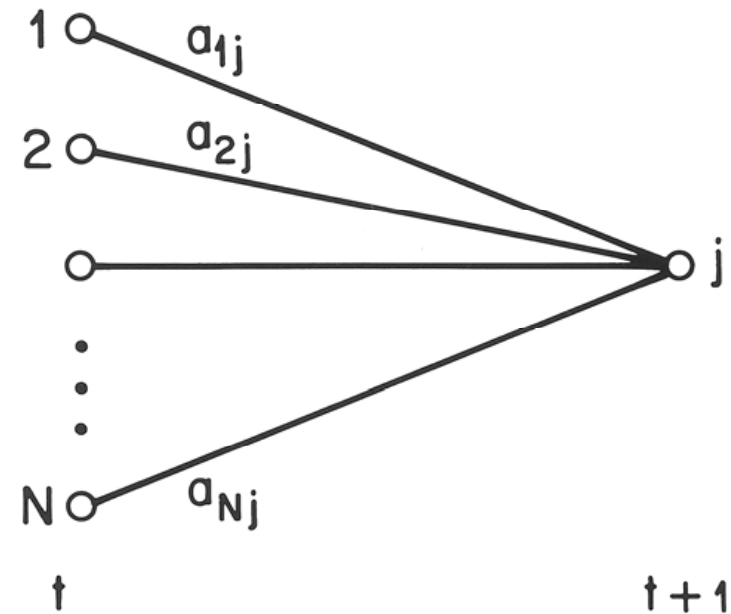
$$\delta_1(i) = p(a_1 | q_1 = s_i)$$

$$\psi_1(i) = 0$$

2. Yineleme:  $t = 1, \dots, T - 1$

$$\delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} p(a_{t+1} | q_{t+1} = s_j)$$

$$\psi_{t+1}(j) = \max_i \delta_t(i) a_{ij}$$



## Viterbi Algoritması - 2

$$\delta_t(i) \equiv p(q_1^{t-1}, q_t = s_i, a_1^t)$$

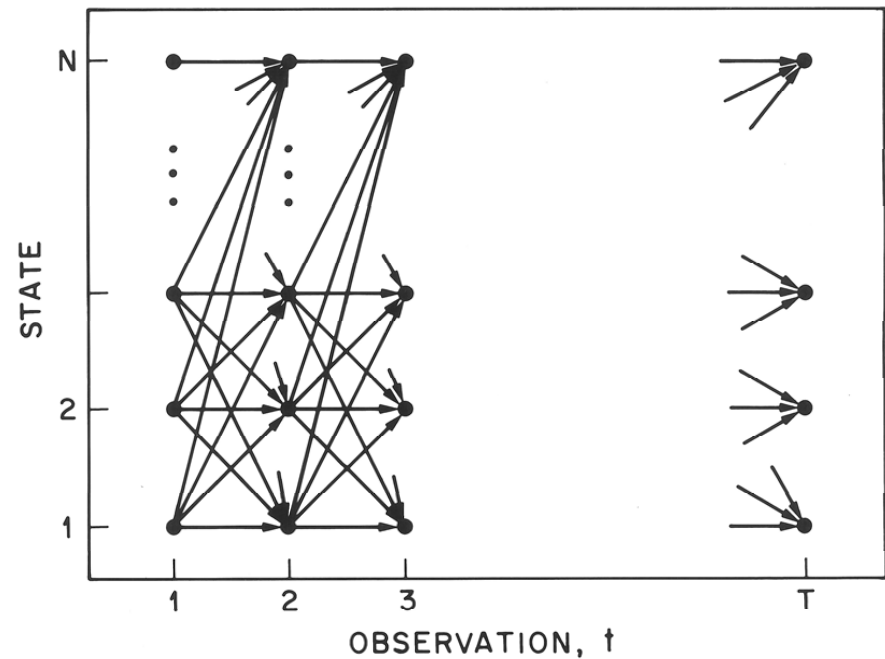
3. Son:

$$p^* = \max_i \delta_T(i)$$

$$q_T^* = \arg \max_i \delta_T(i)$$

4. Geri izleme:  $t = T - 1, \dots, 1$

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$





## En iyi sözcük dizisi

- Viterbi algoritması  $N^T$  durum dizisi içinden en iyi olanı  $N^2T$  işlemle bulur.
- Sonlu durum dönüştürücüsünde sözcük bilgileri de saklanarak en iyi sözcük dizisi de belirlenmiş olur.
- Çıktı: En iyi durum/sözcük dizisi ve zaman bilgisi

# Model Parametrelerinin Kestirimi

# Ayrık (Görünür) Markov Model Olasılıklarının Kestirimi

- En yüksek olabilirlik kestirimi verinin olasılığını en yüksek yapan model parametrelerini belirler.
- Ayrık (görünür) Markov modelleri için parametreleri elimizdeki veriyi kullanarak sayma ve bölme işlemleriyle bulabiliriz.
- Örneğin ikili (bigram) dil modeli için

$$\hat{P}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$$

## Dil Modelinde Sıfır Olasılıklarla Başa Çıkma - 1

- En yüksek olabilirlik kestirimi veride görülmeyen dizilere 0 olasılık atar.
- Dil modellemede bunu istenmez.
- Yumuşatma yöntemleri (smoothing)
  - Zenginden alıp fakire verme 😊
- Daha düşük derece koşullanmış olasılıkları kullanma
  - Aradeğerleme (interpolation)
  - Unutma (back-off)

# Dil Modellemede Yumuşatma Yöntemleri

- $N$ : derlem boyutu,  $V$ : dağarcık boyutu
- En yüksek olasılık kestirimi:  $P(w_i) = \frac{C(w_i)}{N}$
- Laplace (bir ekleme):  $P^*(w_i) = \frac{C(w_i)+1}{N+V}$
- Sabit ( $d$ ) azaltma: Eğer  $C(w_i) > 0$  ise  $P^*(w_i) = \frac{C(w_i)-d}{N}$   
Arda kalan olasılık görülmeyen sözcüklere dağıtılır.
- Good-Turing
- Kneser-Ney

## Daha düşük derece modelleri kullanma

- Ara değerlendirme (interpolation):

$$\tilde{P}(w_i|w_{i-1}) = \lambda P(w_i|w_{i-1}) + (1 - \lambda)P(w_i)$$

- Unutma (back-off):

$$\tilde{P}(w_i|w_{i-1}) = \begin{cases} C(w_{i-1}, w_i) > 0 \text{ ise} & P^*(w_i|w_{i-1}) \\ \text{değilse} & \alpha(w_{i-1})\tilde{P}(w_i) \end{cases}$$

$P^*(w_i|w_{i-1})$  azaltılmış olasılıklarıdır.

# Dil modelleme için yapay öğrenme

- En yüksek entropi (MaxEnt)
  - Öznitelik temelli bir yaklaşım:  $\phi(w_i, h_i)$
  - Doğrusal kısıtlar için üstel bir dağılım tanımlar:

$$P(w_i|h_i) = \frac{e^{\langle \alpha, \phi(w_i, h_i) \rangle}}{\sum_w e^{\langle \alpha, \phi(w, h_i) \rangle}}$$

- Ayırıcı (ayrımsayıcı – discriminative) dil modelleri
  - Sadece doğruları değil yanlışları da dikkate alır
  - Doğrusal veya log-doğrusal (= üstel) modeller

# Akustik Modellerin Kestirimi

- Üretici
  - En yüksek olabilirlik kestirimi
- Ayırıcı (ayrımsayıcı)
  - Koşullu en yüksek olabilirlik
  - En yüksek ortak bilgi kestirimi
  - En düşük (sesçik/sözcük) hata oranı kestirimi
  - En düşük Bayes riski kestirimi



# Saklı Markov Modelleri için En Yüksek Olabilirlik Kestirimi

- Baum-Welch Algoritması
  - Bir beklenti-(en)büyütme (Expectation-Maximization) Algoritması
- İki adımdan oluşan döngüsel bir yöntem
  - Beklenti adımında bir önceki döngüdeki parametreler kullanılarak logaritmik olabilirlik fonksiyonunun beklenen değeri hesaplanır. Bu değer logaritmik olabilirlik fonksiyonunun bir alt sınırıdır.
  - (En)büyütme adımında ise bu alt sınırı (en)iyileyen parametre değerleri bulunur.

## Beklenti (En)Büyütme Algoritması

- Model:  $M$ , gözlenen değişken:  $X$ , saklı değişken:  $Z$
- Log olabilirlik:  $\mathcal{L}(M|X) = \log \prod_t p(x_t|M) = \sum_t \log p(x_t|M)$
- Tüm log olabilirlik:  $\mathcal{L}_c(M|X, Z) = \sum_t \log p(x_t, z_t|M)$
- Beklenti:  $Q(M|M^i) = E[\mathcal{L}_c(M|X, Z)|X, M^i]$
- (En)büyütme:  $M^{i+1} = \arg \max_M Q(M|M^i)$
- Teorem:  $Q(M'|M) \geq Q(M|M)$  ise  $\mathcal{L}(M'|X) \geq \mathcal{L}(M|X)$

# İleri (Forward) Algoritması

$$\alpha_t(i) \equiv p(a_1^t, q_t = s_i)$$

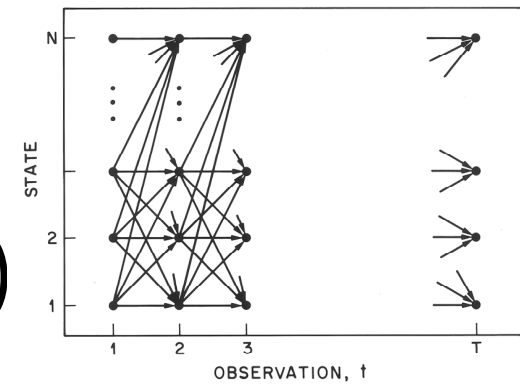
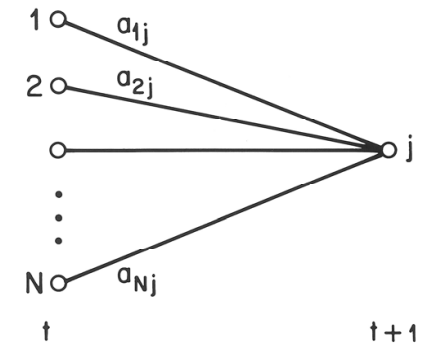
1. Başlangıç:

$$\alpha_1(i) = p(a_1 | q_1 = s_i)$$

2. Yineleme:  $t = 2, \dots, T - 1$

$$\alpha_{t+1}(j) = \left[ \sum_i \alpha_t(i) a_{ij} \right] p(a_{t+1} | q_{t+1} = s_j)$$

3. Son:  $P(A | \Lambda(S)) = \sum_i \alpha_T(i)$



# Geri (Backward) Algoritması

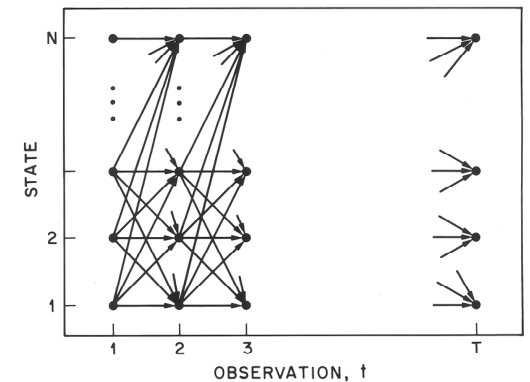
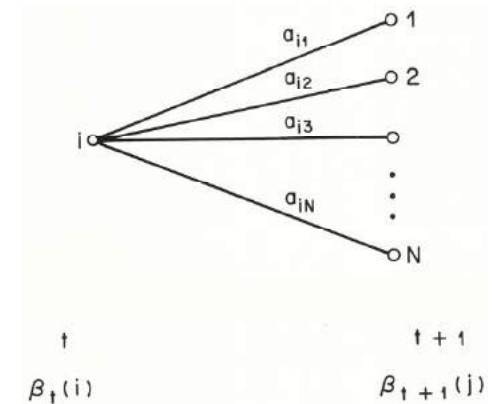
$$\beta_t(i) \equiv p(a_{t+1}^T | q_t = s_i)$$

1. Başlangıç:

$$\beta_T(i) = 1$$

2. Yineleme:  $t = T - 1, \dots, 1$

$$\beta_t(i) = \sum_j a_{ij} \beta_{t+1}(j) p(a_{t+1} | q_{t+1} = s_j)$$



## Beklenen değerlerin hesaplanması

- Beklenen durum geçiş sayıları

$$\xi_t(i, j) \equiv P(q_t = s_i, q_{t+1} = s_j | a_1^T, \Lambda)$$
$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} p(a_{t+1} | q_{t+1} = s_j) \beta_{t+1}(j)}{\sum_{i'} \sum_{j'} \alpha_t(i') a_{i'j'} p(a_{t+1} | q_{t+1} = s_{j'}) \beta_{t+1}(j')}$$

- Beklenen durum bulunma sayıları

$$\gamma_t(i) \equiv P(q_t = s_i | a_1^T, \Lambda)$$
$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i'} \alpha_t(i') \beta_t(i')} = \sum_j \xi_t(i, j)$$

# Model Parametrelerinin Güncellenmesi

- Geçiş olasılıkları

$$\hat{a}_{ij} = \frac{\sum_t \xi_t(i, j)}{\sum_t \gamma_t(i)}$$

- Gauss çıktı dağılımı için ortalama ve değişinti (varyans)

$$\hat{\mu}_i = \frac{\sum_t \gamma_t(i) a_t}{\sum_t \gamma_t(i)}$$

$$\hat{\sigma}_i^2 = \frac{\sum_t \gamma_t(i) (a_t - \hat{\mu}_i)^2}{\sum_t \gamma_t(i)}$$

# Konuşma Tanıma için Derin Öğrenme

# Derin Öğrenme ve Yapay Sinir Ağları

- «Derin» derken yapay sinir ağlarının mimarisi (katman sayısı) kastedilmektedir.
- YSA derinleştikçe veri gösterimi de öğrenilmekte ve verinin önceden işlenmesine ihtiyaç azalmaktadır.
- Konuşma tanımda Mel Frekans Kepstral Katsayıları (MFCC) yerini büyük ölçüde log(-mel) enerji özniteliklerine bırakmıştır.



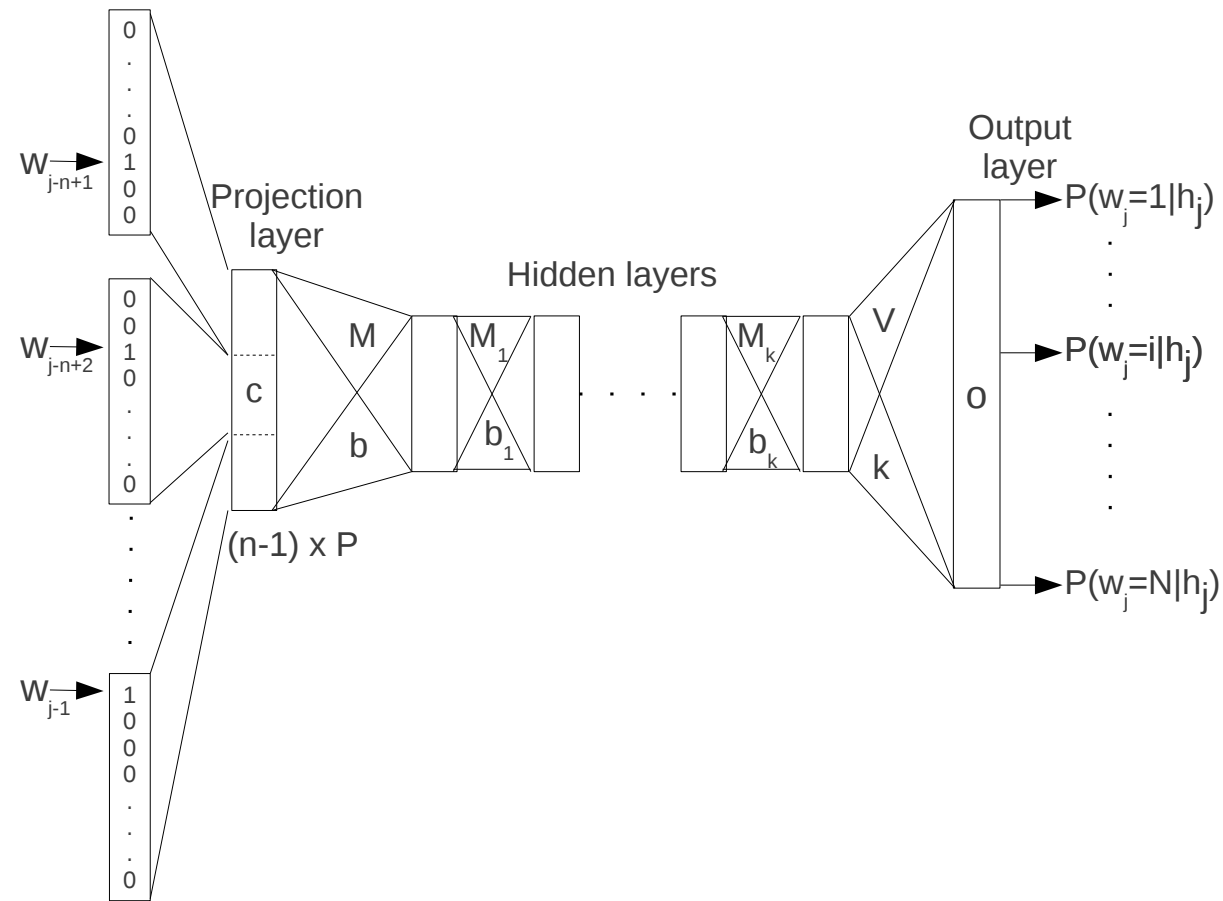
# Konuşma Tanıma için Yapay Sinir Ağları

- Bir sınıflandırma yöntemi olan yapay sinir ağları sınıflandırmanın yanı sıra sınıflar üzerinden bir olasılık dağılımı da üretebilir.
- Bunun için çıkış katmanındaki çıktıların negatif olmayan ve toplamı bir olan değerler olması gerekir.
- Bu nedenle çıkış katmanında «softmax» işlevi kullanılır.

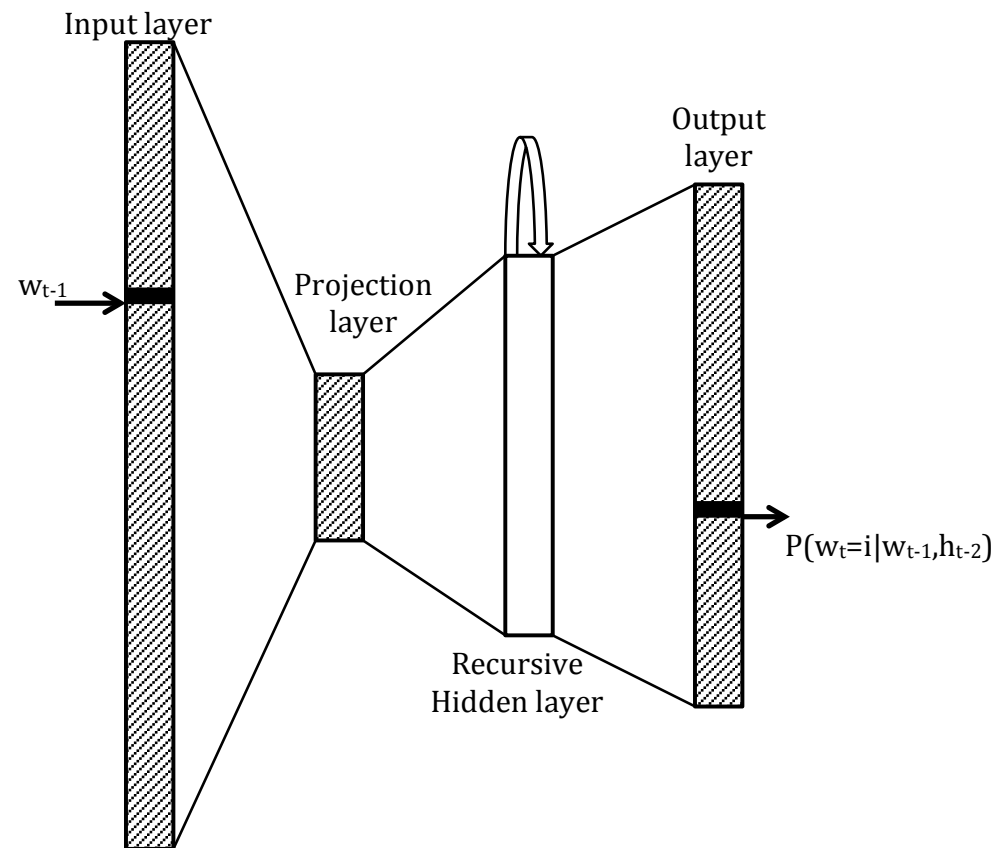
## Dil Modelleme için Yapay Sinir Ağları

- Giriş katmanında geçmişe ait sözcük(ler)  $w_{i-n+1}, \dots, w_{i-1}$  sadece o sözcüğe karşılık gelen girdi 1, diğerleri 0 olacak şekilde oluşturulan vektörleri,
- İlk (doğrusal) ortak katmanda sözcüklerin sürekli bir uzaydaki vektör gösterimleri,
- Çıkış katmanında da tahmin edilen sözcükler  $w_i$  bulunur.
- Bu istenen koşullu olasılığı verir:  $P(w_i | w_{i-n+1}, \dots, w_{i-1})$

# İleri Beslemeli Sinir Ağları

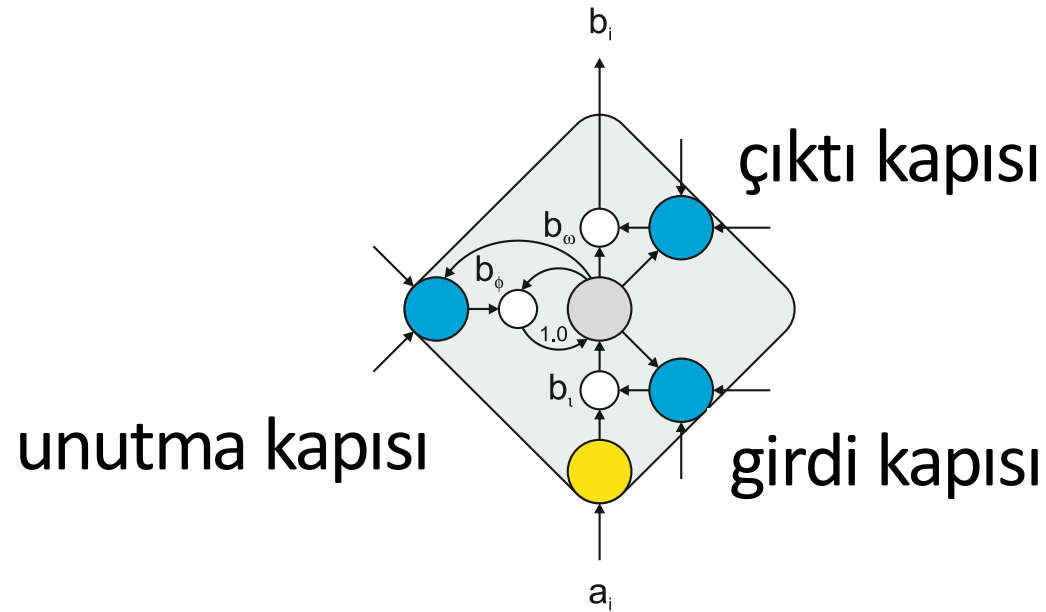


# Yinelemeli (Özyineli) Sinir Ağları

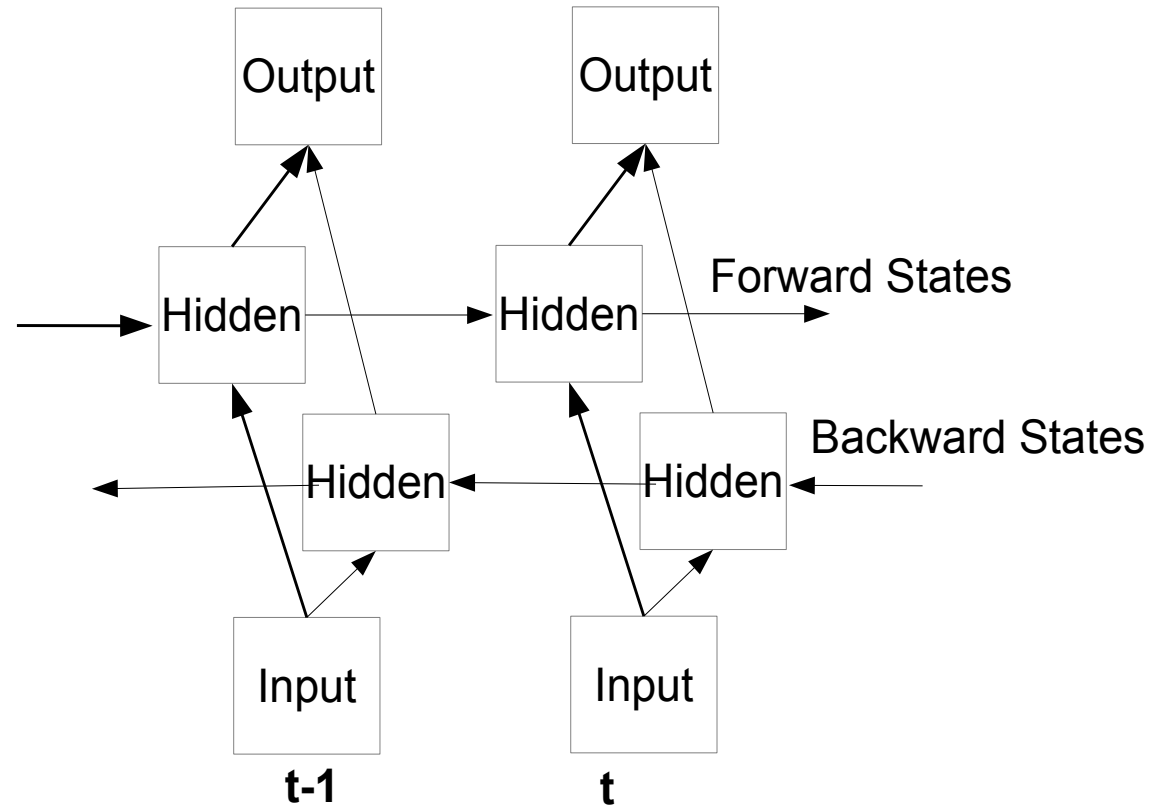
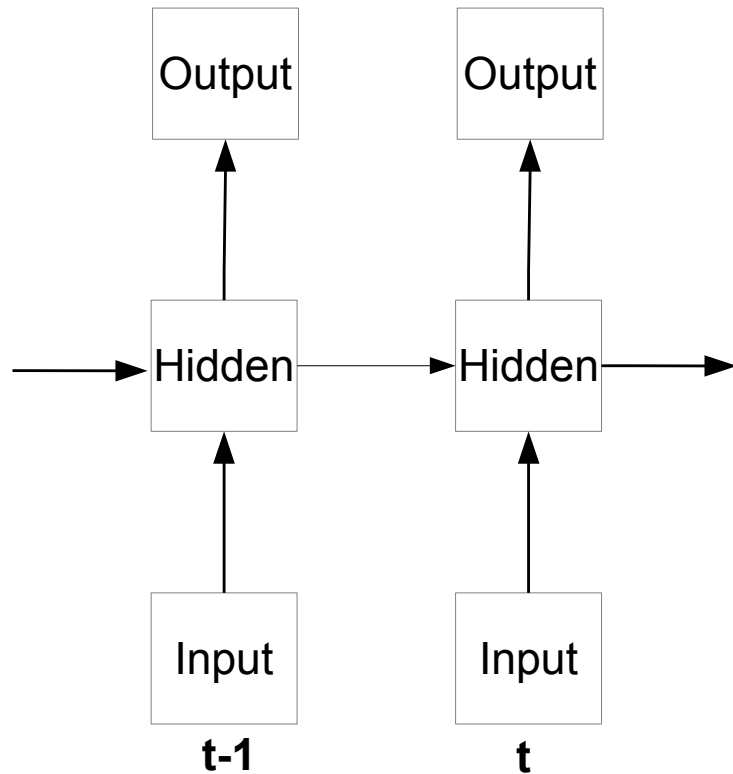


# Uzun Kısa-Sürelili Bellek (LSTM)

- Bir özyineli (yinelemeli) sinir ağı türüdür.
- Daha uzun etkileşimleri modelleyebilir.



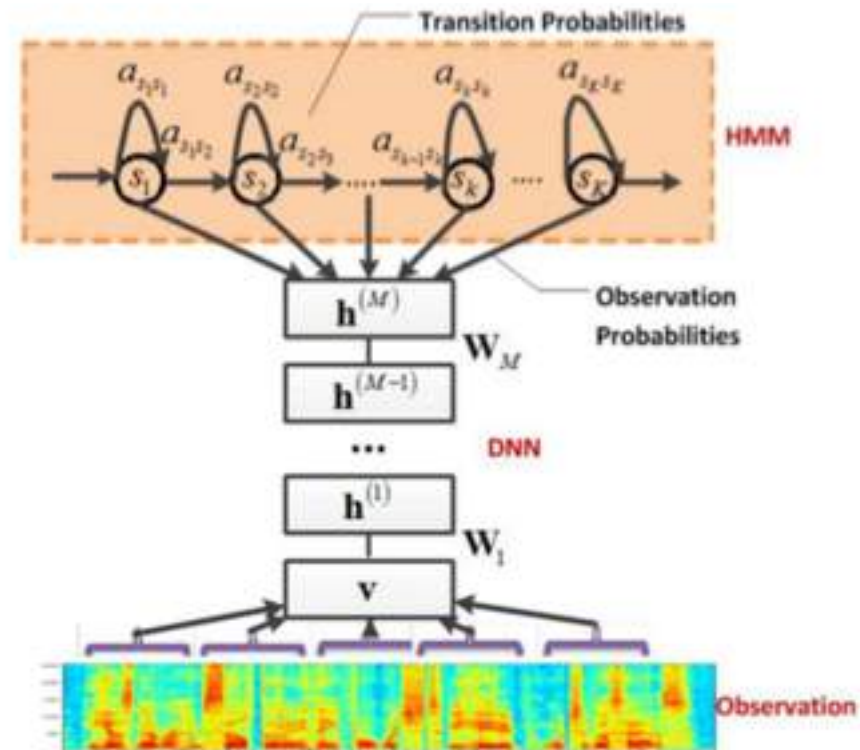
# Tek ve Çift Yönlü Yinelemeli Sinir Ağları



# Akustik Modellemede Yapay Sinir Ağları

- Giriş katmanında bir zamana ait öznitelik vektörü  $a_t$
- Çıkış katmanında da Markov modelinin durumları  $q_t$  bulunur.
- Bu bir sonsal olasılık verir:  $P(q_t|a_t)$
- Bayes kuralıyla durum çıktı olasılık dağılımı elde edilir.  
$$p(a_t|q_t) \propto P(q_t|a_t)/P(q_t)$$

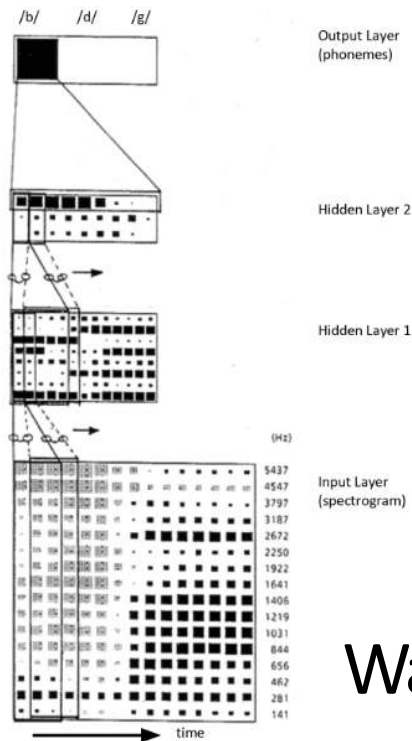
# İleri Beslemeli Sinir Ağları



Dahl, George E., et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." *Audio, Speech, and Language Processing, IEEE Transactions on* 20.1 (2012): 30-42.

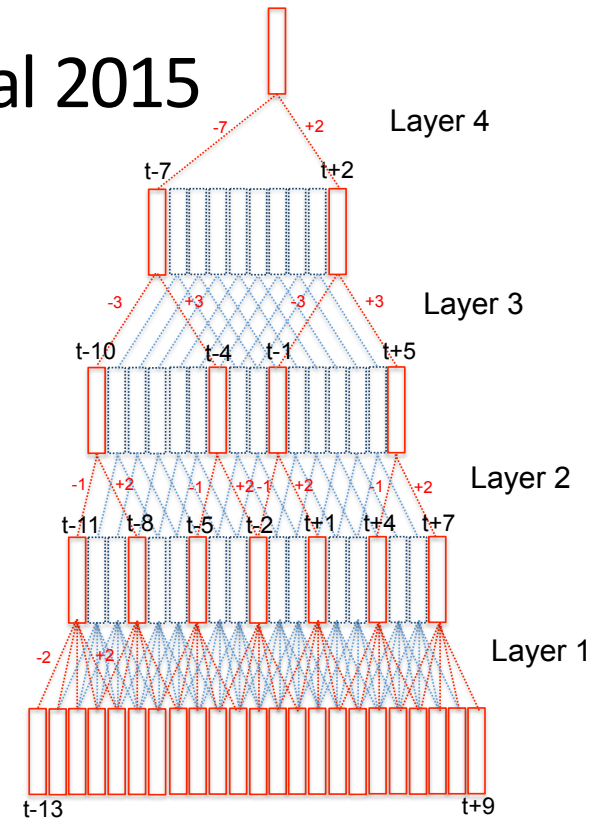


# Zaman Gecikmeli Sinir Ağları

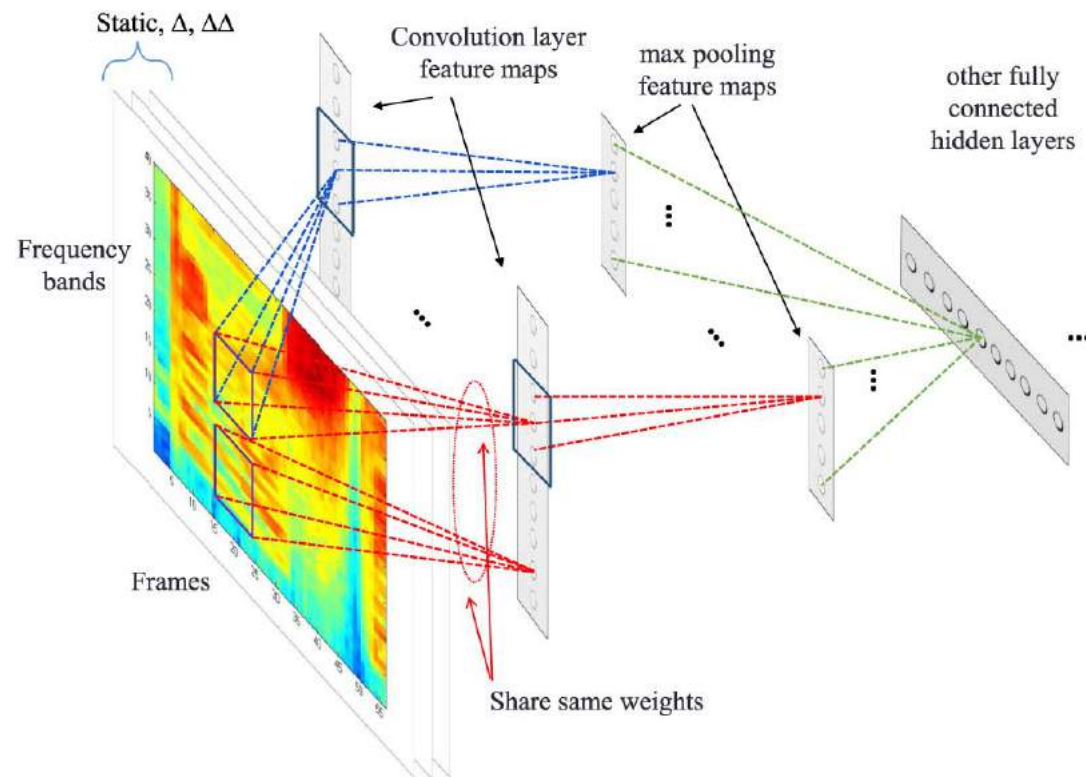


Waibel et al 1989

Peddinti et al 2015

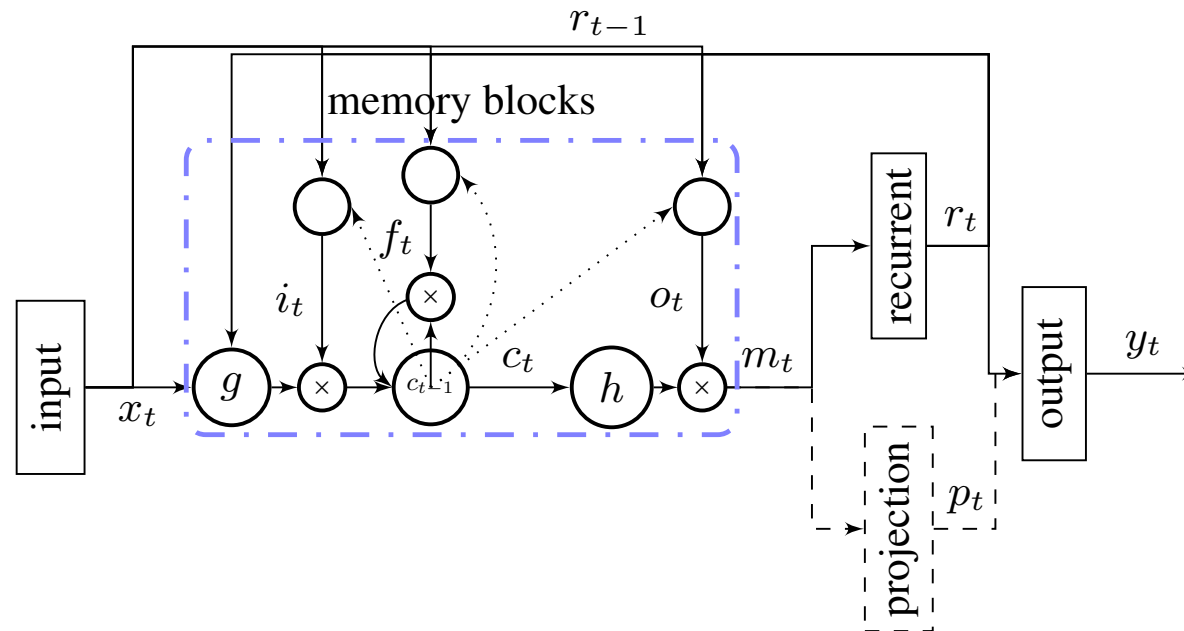


# Evrişimsel Sinir Ağları



Abdel-Hamid et al, «Convolutional Neural Networks for Speech Recognition, IEEE/ACM TASLP 22(10), Oct 2014.

# Yinelemeli Sinir Ağları: LSTM



Haşim Sak, Andrew Senior, Françoise Beaufays, «LSTM based RNN architectures for LVCSR», 2014.

# Yapay Sinir Ağlarında Parametre Kestirimi

- Kestirimde çeşitli eniyileme yöntemleri kullanılmaktadır.
- Yaygın olarak (küçük grup) bayır inişi yöntemleri kullanılır:

$$w \leftarrow w - \eta \nabla E(w)$$

- Hata Geri Yayma (Back-Propagation) yönteminde eğitim hesaplanırken türevler için zincir kuralından yararlanılır.

$$\text{Gizli katman } z_h = f(w_h^T x) \quad \text{Çıktı } y_i = g(v_i^T z)$$

$$\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_h} \frac{\partial z_h}{\partial w_{hj}}$$

## YSA Kestiriminde Amaç İşlevleri - 1

- Çapraz Entropi  $H(p, q) = -\sum_i p_i \log q_i$

$$\mathcal{F}_{CE} = -\sum_u \sum_t \log y_{ut}(s_{ut})$$

- Dizisel İşlevler

- En büyük ortak bilgi (MMI)

$$\mathcal{F}_{MMI} = \sum_u \log \frac{p(A_u|S(W_u))^{\kappa} P(W_u)}{\sum_W p(A_u|S(W))^{\kappa} P(W)}$$

## YSA Kestiriminde Amaç İşlevleri - 2

- Dizisel İşlevler

En küçük Bayes riski (MBR)

$$\mathcal{F}_{MBR} = \sum_u \log \frac{\sum_W p(A_u|S(W))^{\kappa} P(W) A(W, W_u)}{\sum_{W'} p(A_u|S(W'))^{\kappa} P(W')}$$

Burada  $A(W, W_u)$  ham doğruluğu ifade eder.

- En küçük sesçik hatası (MPE): doğru sesçik sayısı
- Durum seviyesi en küçük Bayes riski: doğru durum sayısı

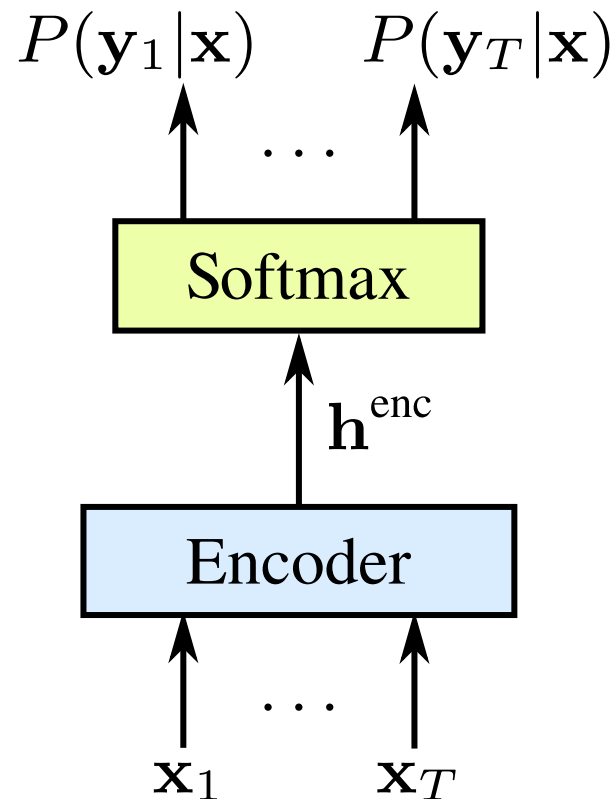
Vesely et al, «Sequence-discriminative training of deep neural networks», Interspeech 2013.

## Baştan Sona (Uçtan Uca) Konuşma Tanıma

- Son yıllarda oldukça alt düzey öznitelikler dizilerinden (ve hatta konuşma sinyalinden) doğrudan harf/sesçik/sözcük dizileri üreten sistemler önerilmiştir.
- Bu sistemler diziden diziye (seq2seq) modeller kullanırlar.
- İlk olarak girdi gilyazıcı tarafından gömülü bir gösterime çevrilir, daha sonra da gizçözücü tarafından çıktılar üretilir.
- Bu yaklaşım genelde daha çok veri (ve daha az bilgi?) gerektirmektedir.

# Bağlantıcı Zamansal Sınıflandırma

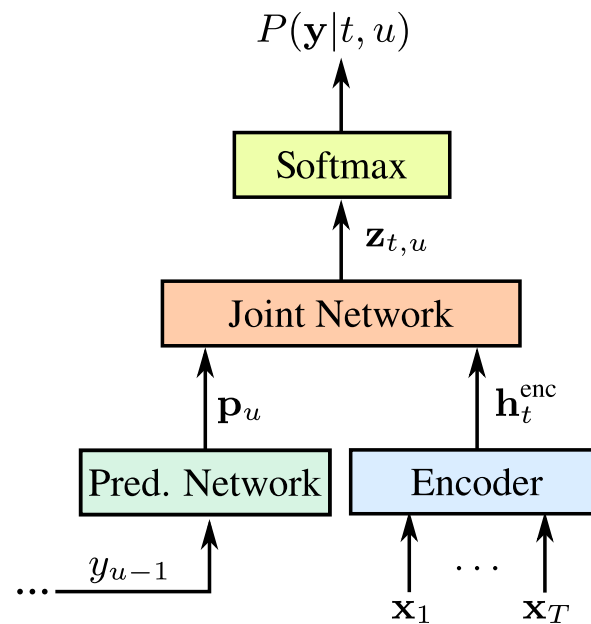
## Connectionist Temporal Classification (CTC)



Prabhavalkar et al, «A Comparison of Sequence-to-Sequence Models for Speech Recognition», Interspeech 2017.



# Özyineli Dönüştürücü Sinir Ağı



Prabhavalkar et al, «A Comparison of Sequence-to-Sequence Models for Speech Recognition», Interspeech 2017.

# Dinle, Dikkat et, Yaz

## Chan et al, «Listen, Attend and Spell», 2015

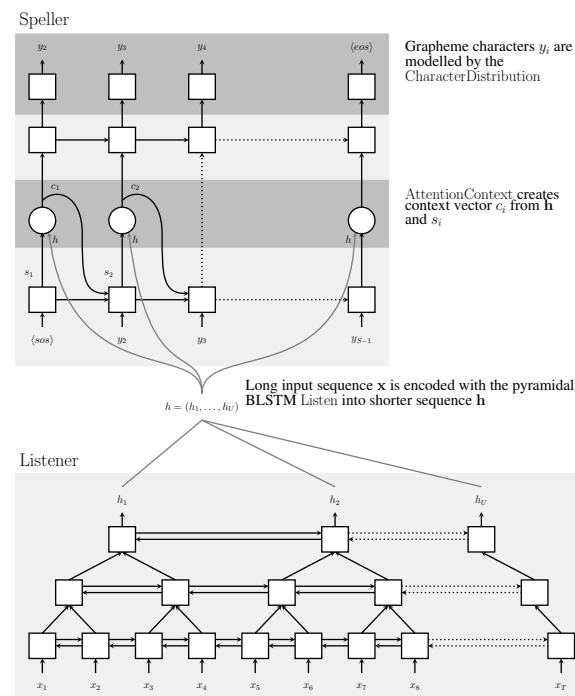
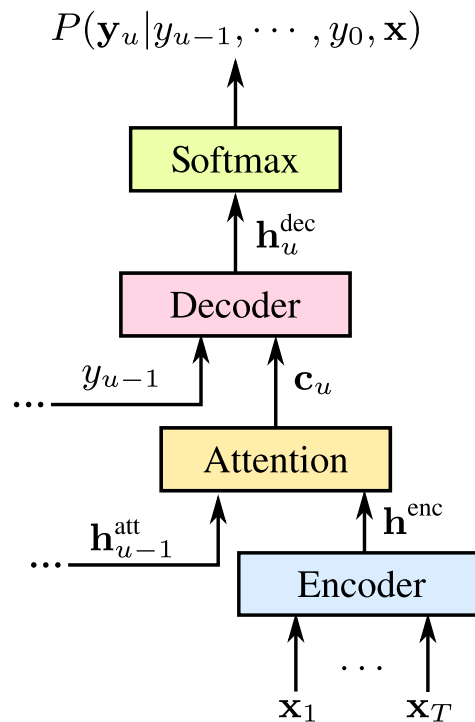


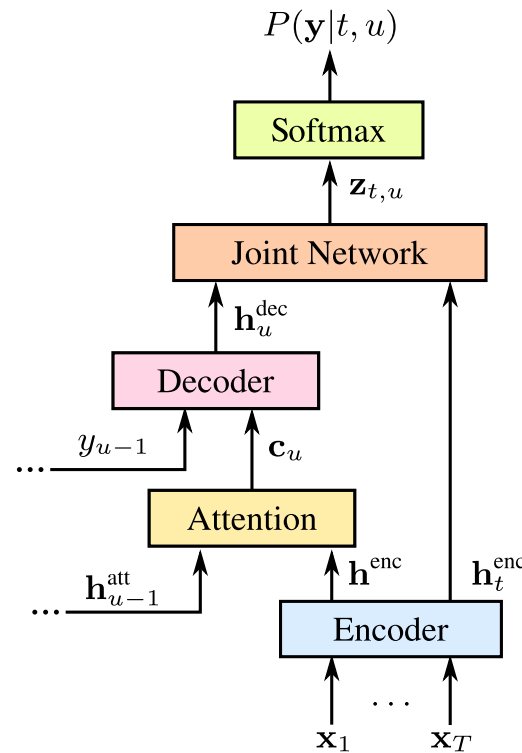
Figure 1: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence  $x$  into high level features  $h$ , the speller is an attention-based decoder generating the  $y$  characters from  $h$ .

# Dikkat Tabanlı Model



Prabhavalkar et al, «A Comparison of Sequence-to-Sequence Models for Speech Recognition», Interspeech 2017.

# Dikkat İçeren Özyineli Dönüştürücü Sinir Ağı

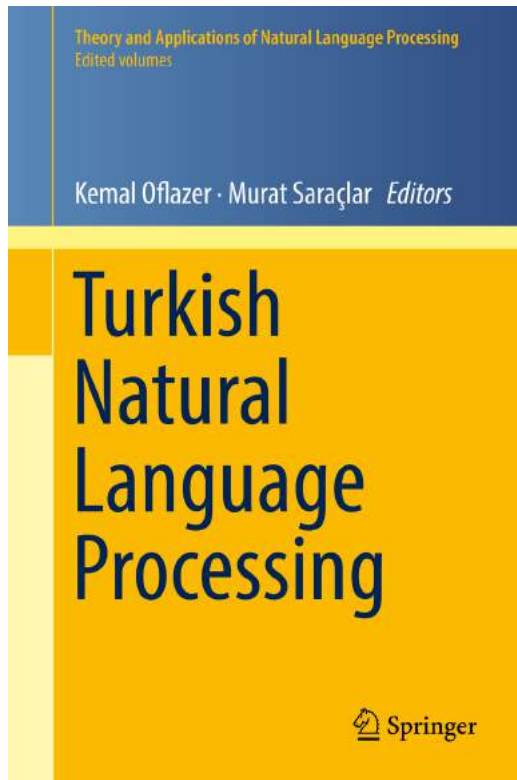


Prabhavalkar et al, «A Comparison of Sequence-to-Sequence Models for Speech Recognition», Interspeech 2017.

# 3 Temmuz Salı: Ses ve Konuşma İşleme Günü

- **IEEE Signal Processing Society Distinguished Industry Speaker**  
Speech Recognition: What's Left? *Michael Picheny*
- Türkçe için Konuşma Tanıma ve Derin Öğrenmeyle Dil Modelleme *Ebru Arısoy*
- Tek ve Çok Kanallı Ses Kaynağı Ayırma için Derin Öğrenme *Hakan Erdoğan*
- Duygulanımsal Konuşma ve İsmar Modelleri için Derin Öğrenme *Engin Erzin*
- Konuşma Sentezi *Cenk Demiroğlu*
- Karma Gerçeklik için Ses Etkileşimleri *Cumhur Erkut*

Pek Yakında ...



## **Turkish Natural Language Processing** Kemal Oflazer, Murat Saraçlar *Editors*

- Ch. 4: Language Modeling for Turkish Text and Speech Processing  
Arısoy and Saraçlar
- Ch. 5: Turkish Speech Recognition  
Arısoy and Saraçlar