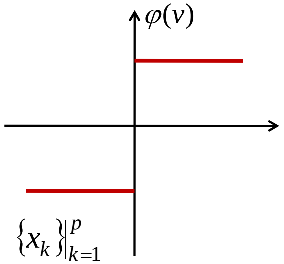
Hatırlatma

Ayrık Zaman Hopfield Ağı ile Çağrışımlı Bellek Tasarımı

<u>Kullanılan Hücre Modeli:</u>

McCulloch-Pitts



Eksik birşey var!!

$$\varphi(v)=\begin{cases}1 & v>0\\ -1 & v<0\end{cases}$$

<u>Örüntüler:</u> $x\in\{-1,1\}^n \quad \{x_k\}\big|_{k=1}^{p}$

<u>1. Aşama:</u> Belleğin Oluşturulması

n boyutlu, p tane veriden yararlanarak belleği oluşturmak için ağırlıklar belirlenmeli

$$w_{ji}=\frac{1}{n}\sum_{k=1}^{p}x_i^{(k)}x_j^{(k)},\quad i\neq j$$

- Her nöronun çıkışı diğer nöronların girişine bağlı
- kendisine geribesleme yok
- ağırlık matrisi simetrik

Hatırlatma

Ağırlıklar önceden hesaplanabilir veya

$$w_{ji}(k+1)=\alpha w_{ji}(k)+(1-\alpha)x_i(k)x_j(k),\quad 0<\alpha<1$$ ile belirlenebilir.

<u>2. Aşama:</u> Anımsama

Dinamik yapı: $x(k+1)=\varphi(v(k))=\varphi\big(Wx(k)\big)$

Verilen bir ilk koşul için durumlar dinamik yapı gereği senkron veya asenkron yenilenir

Neye karşılık düşüyor?

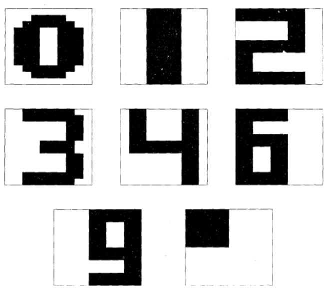
Tüm nöronlar için $x(k+1)=x(k)$ olduğunda bellekte saklanan örüntülerden birine karşılık düşen bir kararlı düğüm noktasına erişilir.

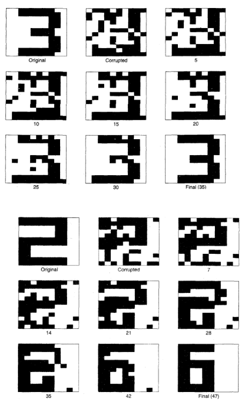
<u>Örnek:</u> $\{x_k\}\big|_{k=1}^{p}$

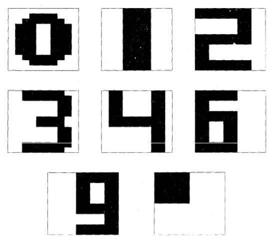
$$\{x_k\}\big|_{k=1}^{4}=\left\{\begin{bmatrix}1\\1\\-1\\-1\end{bmatrix},\begin{bmatrix}-1\\-1\\1\\1\end{bmatrix},\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix},\begin{bmatrix}-1\\1\\-1\\1\end{bmatrix}\right\}$$

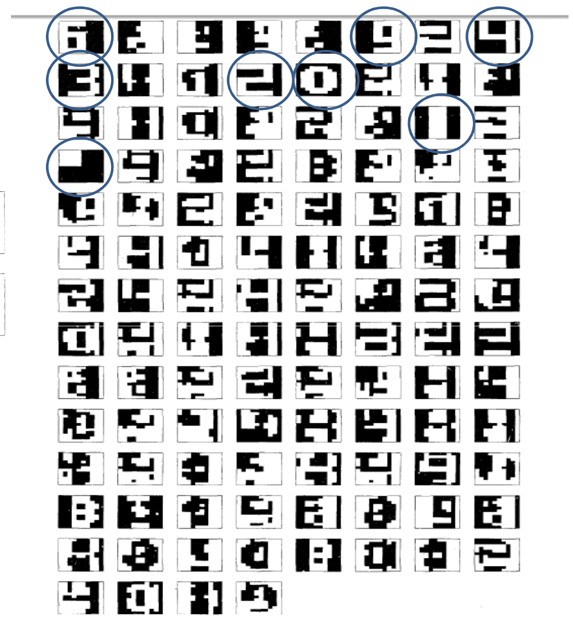
Bazı Sorular

- Gerçekten de belirlenen ağırlıklar ile istenilen kararlı denge noktalarına erişmemizi sağlayacak dinamik sistem yaratıldı mı?
- Eğer evet ise, bir bozulmuş veya eksik örüntü ile başlayarak bu örüntünün bellekteki aslına erişilebilinir mi?
- Herhangi bir ilk ilk koşul ile başlanıldığında ağa ilişkin dinamik hangi kararlı durum çözümünü verecek ?
- Küçük hata ile kaç örüntü belleğe yerleştirilebilinir?









Hopfield Ağı yakınsıyor, ama nereye?

$$x_j(k+1) = \varphi\left(\sum_{i=1}^{n} w_{ji} x_i(k) + I_j\right)$$

$$x_j^{\,d} = \varphi\left(\sum_{i=1}^{n} w_{ji} x_i^{\,d} + I_j\right)$$

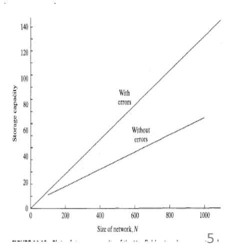
Ağırlıkları yerleştirelim: $x_j^{\,d} = \varphi\left(\frac{1}{n}\sum_{j\neq i}\sum_{k=1}^{p} x_i^{(k)} x_j^{(k)} x_i^{\,d} + I_j\right)$

$$= \varphi\left(\frac{1}{n} x_i^d \sum_{j\neq i}\left(x_i^d\right)^2 + \frac{1}{n}\sum_{j\neq i}\sum_{k\neq p}^{p} x_i^{(k)} x_j^{(k)} x_i^{\,d} + I_j\right)$$

$$x_j^{\,d} = \varphi\left(\frac{n-1}{n} x_j^d + \delta_j^d\right)$$

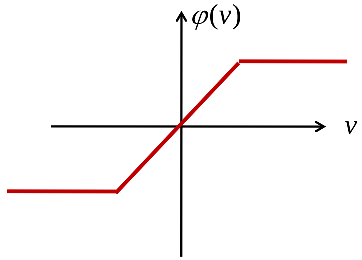
n büyük ise $x_j^{\,d} = \varphi\left(x_j^d + \delta_j^d\right)$

p > 0.38n ise bellek anlamsızlaşıyor



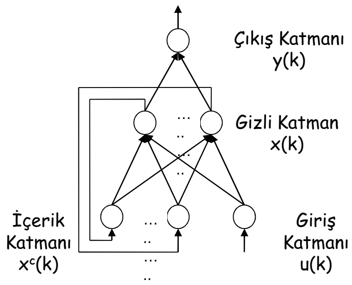
Sürekli Zaman Hopfield Ağı ile Çağrışımlı Bellek Tasarımı

$$\dot{x}_i(t) = \alpha_i x_i + \sum_{j=1}^{n} w_{ij}\varphi(x_j) + I_i)$$

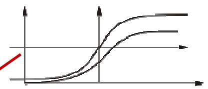


Ayrık zaman Hopfield ağındaki gibi ağırlıklar belirlenir ve diferansiyel denklem takımı çözülür.

Elman Ağı (1990)



$$y(k) = \sum_{i=1}^{n} w_i^y x_i(k)$$



$$x_i(k) = f(v_i(k))$$

$$x_j^c(k) = x_j(k-1)$$

$$v_i(k) = \sum_{j=1}^{n} w_{ij}^x x_j^c(k) + w_i^u u(k)$$

Tanıdık ama farklı
$$\begin{cases} x(k) = \Phi(x(k-1), u(k-1)) \\ y(k) = \Psi(x(k)) \end{cases}$$

Bu ağ yapısı ile ne yapabiliriz? Sistem tanıma

Dinamik sistemi tam olarak belirlemek için ağırlıkların belirlenmesi gerek, nasıl?

Ağırlıkların belirlenmesi çok katmanlı algılayıcıya benzer şekilde....

$$E = \frac{1}{2}\left(yd(k) - y(k)\right)^2$$ Dinamik sürece ilişkin her gözlem anındaki veri dikkate alınarak hata tanımlanıyor

Her katmana ilişkin ağırlık matrisleri en dik iniş yöntemi ile hata enazlanacak şekilde güncelleniyor

$$\Delta w = -\eta\frac{\partial E}{\partial w}$$

$$\Delta w_i^y = \eta\underbrace{\left(y_d(k) - y(k)\right)}x_i(k)$$

$$\Delta w_i^u = \eta\underbrace{\left(y_d(k) - y(k)\right)}w_i^y f'(x_i(k))u(k)$$

$$\Delta w_{ij}^x = \eta\underbrace{\left(y_d(k) - y(k)\right)}w_i^y f'(x_i(k))x_j(k-1)$$

Hata terimi her k anında tanımlanıyor

Güncelleme terimlerinin açık ifadesi:

$$-\frac{\partial E}{\partial w_i^y} = \frac{\partial E}{\partial y(k)}\frac{\partial y(k)}{\partial w_i^y} = -(y_d(k)-y(k))x_i(k)$$

$$-\frac{\partial E}{\partial w_i^u} = \frac{\partial E}{\partial y(k)}\frac{\partial y(k)}{\partial x_i(k)}\frac{\partial x_i(k)}{\partial v_i(k)}\frac{\partial v_i(k)}{\partial w_i^u} = -(y_d(k)-y(k))w_i^y\frac{\partial x_i(k)}{\partial v_i(k)}u(k)$$

$$-\frac{\partial E}{\partial w_{ij}^x} = \frac{\partial E}{\partial y(k)}\frac{\partial y(k)}{\partial x_i(k)}\frac{\partial x_i(k)}{\partial v_i(k)}\frac{\partial v_i(k)}{\partial w_{ij}^x} = -(y_d(k)-y(k))w_i^y\frac{\partial x_i(k)}{\partial v_i(k)}x_j(k-1)$$

Dinamik geriye yayılım kullanılırsa..... $v_i(k)=\sum_{j=1}^{n} w_{ij}^x x_j(k-1)+w_i^u u(k)$

$$\Delta w_{ij}^x = \eta(y_d(k)-y(k))w_i^y f'(x_i(k))\left(x_j(k-1)+\sum_{l=1}^{n} w_{il}^x\frac{\partial x_l(k-1)}{\partial w_{ij}^x}\right)$$

$$\frac{\partial x(k)}{\partial w_{ij}^x} = \frac{\partial x_i(k)}{\partial v_i(k)}\frac{\partial}{\partial w_{ij}^x}\left(\sum_{l=1}^{n} w_{il}^x x_l(k-1)+w_i^u u(k)\right) = \frac{\partial x_i(k)}{\partial v_i(k)}\left(x_j(k-1)+\sum_{l=1}^{n} w_{il}^x\frac{\partial x_l(k-1)}{\partial w_{ij}^x}\right)$$

Elman Ağı ile zamanda yapı tanıma: kelime tanıma

Elman ağı belirli bir kurala göre oluşturulmuş sembol dizisinin altında yatan kuralı öğrenebiliyor.

Bu semboller dili oluşturan sesler olarak düşünülebilir.

Harf dizisine ilişkin gösterim: 6 özellik ile elde ediliyor

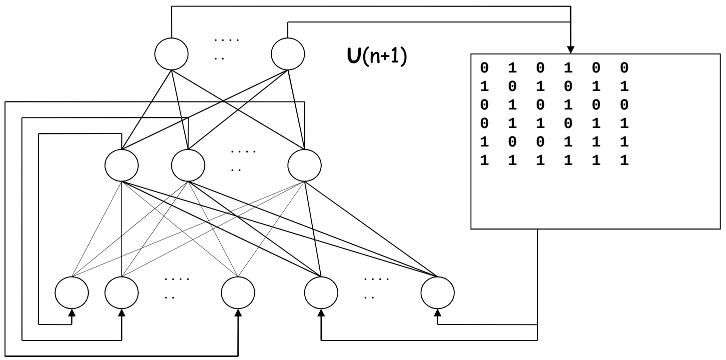
	Sessiz	Sesli	Kesikli	Yüksek	Dönüşlü	Akortlu
b	[1	0	1	0	0	1]
d	[1	0	1	1	0	1]
g	[1	0	1	0	1	1]
a	[0	1	0	0	1	1]
i	[0	1	0	1	0	1]
u	[0	1	0	1	1	1]

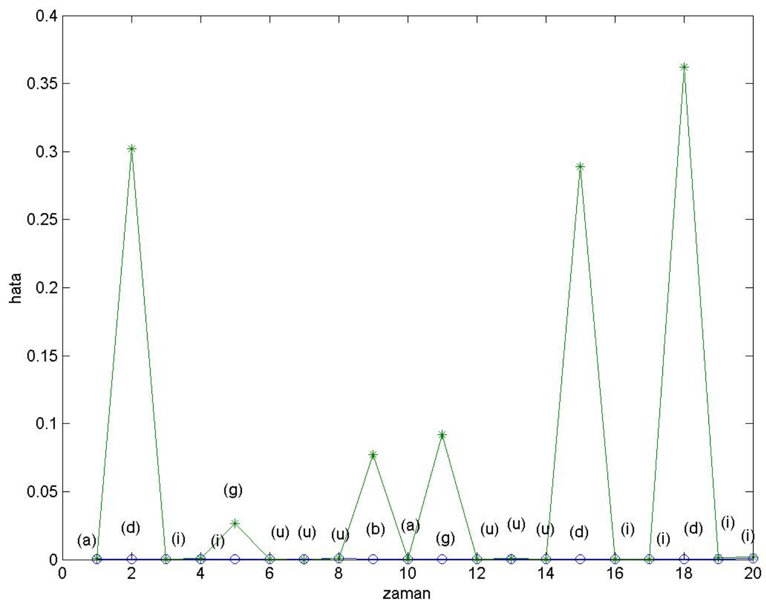
Mahmut Meral Bitirme Ödevi, 2003

önce üç sessiz harften biri rasgele olarak seçilip sesli harfler aşağıdaki kurala göre araya eklenmiştir.

b -> ba
d -> dii
g -> guuu

Örneğin rasgele seçilen sessiz harf dizisi **dbgbddg**… ise oluşan harf dizisi: **diibaguuubadiidiiguuu**… şeklindedir





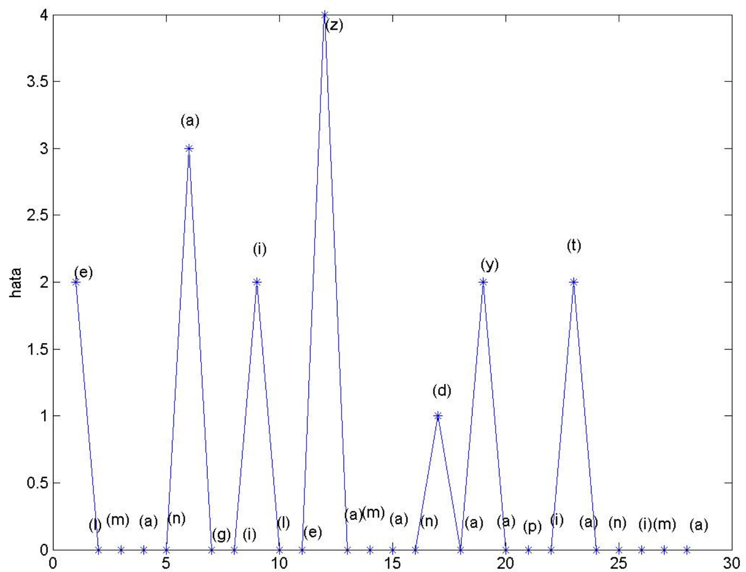
Elman Ağı ile kelime dizisi tanıma

Elman ağı, ses sembollerinden (harflerden) oluşan kelimeler kullanılarak bir harf dizisinin içinden anlamlı harf dizilerini de (kelimeleri) ayırabilir

Uygulamada on üç farklı harften oluşan altı farklı kelime kullanılmaktadır. Harfler beş bitlik vektörler olarak kodlanmıştır. Kelimelerin uzunluğu üç ile yedi harf arasında değişmektedir.

Altı kelimeden rasgele 450 kelime uzunluklu bir dizi oluşturdu. Daha sonra kelime dizisi 2106 harf uzunluğunda bir harf dizisine çevrilerek beş bitlik vektörler şeklinde kodlandı

Giriş		Çıkış	
e	00101	l	01000
l	01000	m	01100
m	01100	a	00001
a	00001	n	01101
n	01101	a	00001
a	00001	ğ	00101
ğ	00101	ı	00110
ı	00110	i	00111
i	00111	l	01000
l	01000	e	00100
e	00100	z	10010
z	10010	a	00001
a	00001	m	01100
m	01100	a	00001
a	00001	n	01101
n	01101	d	00010
d	00010	a	00001
a	00001	y	10001
y	10001	a	00001
a	00001	p	01110
p	01110	ı	00111
ı	00111	t	01111
t	01111	a	00001
a	00001	n	01101
n	01101	ı	00110
ı	00110	m	01100
m	01100	a	00001
a	00001		



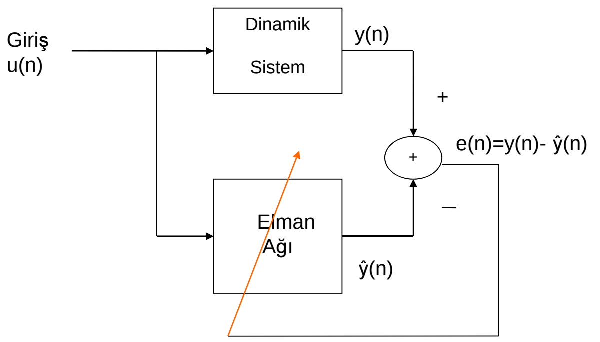
Kelimeler arası sınırlarda iki ve üzerinde hatalı bit oluşmaktadır. Bir nokta hariç ara noktalarda ise hatalı bit yoktur.

.

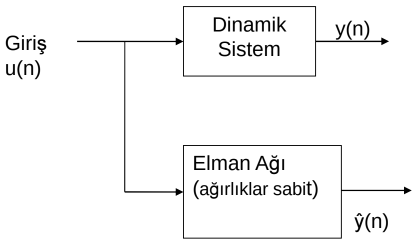
Sadece “zamanda” kelimesindeki “d” harfinin tahmininde bir hatalı bit oluşmaktadır.

Bu durumun sebebi “elman” ve “zamanda” kelimelerinin her ikisinde de “man” dizisinin bulunmasıdır. ”zamanda” kelimesindeki “man” dizisinden sonra “d” gelmesi beklenirken, “elman” kelimesindeki “man” dizisinden sonra altı kelime içinden herhangi birinin ilk harfi (e,a,i,z,y,t) gelebilmektedir.

Bu durum eğitimi olumsuz yönde etkileyerek hatanın yeterince azalmasını önlemektedir

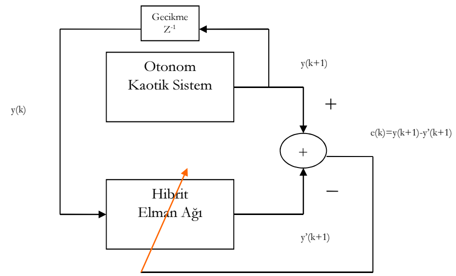


Eğitim



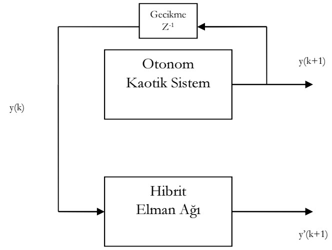
Test ve
Dinamik sistemin kullanılması

Hibrit Elman Ağı ile Kaotik Sistem Tanıma

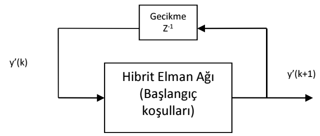


Kaotik zaman serisi tanımada kullanılan hibrit Elman ağının eğitimi

Elman ağının kaotik sistem çıkışıyla uyarılması gerekli



Bir adım sonrasının öngörümü



Ağın otonom davranışı

Elman Ağı ile Sistem Tanıma Uygulamaları

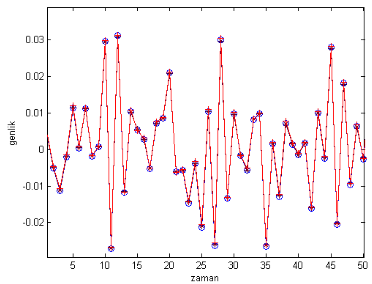
- Dinamik Sistem Tanıma
  Billings Sistemi

$$y(k) = (0.8 - 0.5e^{-y^2(k-1)})y(k-1) - (0.3 + 0.9e^{-y^2(k-1)})y(k-2) + 0.1\sin(\pi y(k-1)) + e(k)$$

- Kaotik Zaman Serisi Tanıma
  Feigenbaum Sistemi

$$x(k+1) = 4x(k)[1 - x(k)]$$

Dinamik Sistem Tanıma: Billings Sistemi



Klasik ve hibrit Elman ağlarının billings sistemi tanıma testi

o-: gerçek sistem çıkışı

+-: klasik Elman ağının çıkışı

*-: hibrit Elman ağının çıkışı

	Ortalama Karasel Hata **MSE**	İşaret Hata Oranı **SER**
RBF-Elman	2.43×10^{-8}	86.60dB
Klasik Elman	2.76×10^{-7}	62.30dB

$$SER = 10\log_{10}\frac{MSS}{MSE}$$

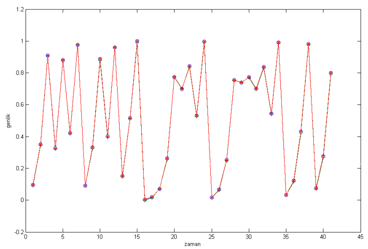
MSE: Ortalama karesel hata

SER: İşaret hata oranı

MSS: İşaretin ortalama karesel değeri

Kaotik Zaman Serisi Tanıma: Faigenbaum Sistemi

Bir Adım Sonrasının Öngörümü Testi



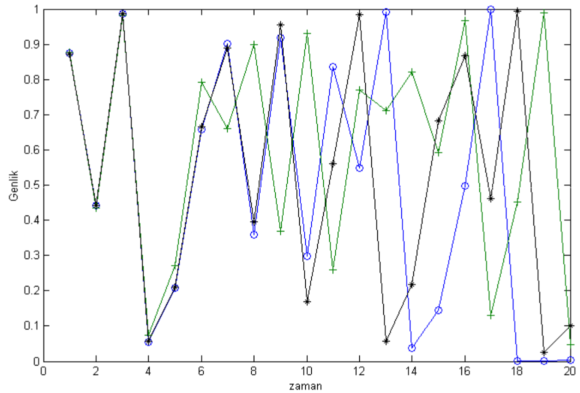
Klasik ve hibrit Elman ağlarının bir adın sonrasının öngörümü testi
o-: gerçek sistem çıkışı
+-: klasik Elman ağının çıkışı
*-: hibrit Elman ağının çıkışı

	Ortalama Karasel Hata MSE	İşaret Hata Oranı SER
RBF-Elman	$2.68x10^{-7}$	61.48dB
Klasik Elman	$5.52x10^{-6}$	48.34dB
MLP*	$1.09x10^{-5}$	45.36dB

* Haykin S.(Editor), *Kalman Filtering and Neural Networks*, John Wiley & Sons, 2001

Kaotik Zaman Serisi Tanıma: Faigenbaum Sistemi

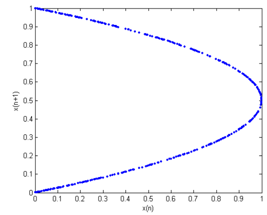
Uzun Erimli Öngörüm Testi



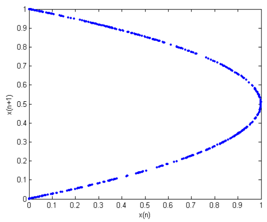
Klasik ve hibrit Elman ağlarının öngörüm erimi karşılaştırması.
o-: gerçek degeri, +-: klasik, *-: hibrit Elman ağının çıkışını göstermektedir.

Durum Port

- Ağın otonom davranışı kaotiktir.
- İki sistemin benzerliği çekicilerine bakılarak görülebilir.
- 1. dereceden sistemin çekicisi x(n)-x(n+1) çizdirilerek elde edilir.



Gerçek sistemin çekicisi



Ağın çekicisi