شبکه‌های عصبی مصنوعی

درس ۲

مدل نرون و معماری‌های شبکه

Neuron Model and Network Architectures

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

http://courses.fouladi.ir/nn

Neuron Model and Network Architectures

شبکه‌های عصبی مصنوعی

مدل نرون و معماری‌های شبکه

۱

نظریه و مثال‌ها

مغز انسان

HUMAN BRAIN

مغز: شبکه‌ای بسیار بزرگ از عصب‌ها (نرون: سلول عصبی)
۱۰۰ میلیارد نرون
۱۰ هزار اتصال برای هر نرون

شبکه‌ی عصبی مصنوعی:
الگوبرداری از شبکه‌ی عصبی طبیعی

نرون طبیعی

NATURAL NEURON

سوما
Soma

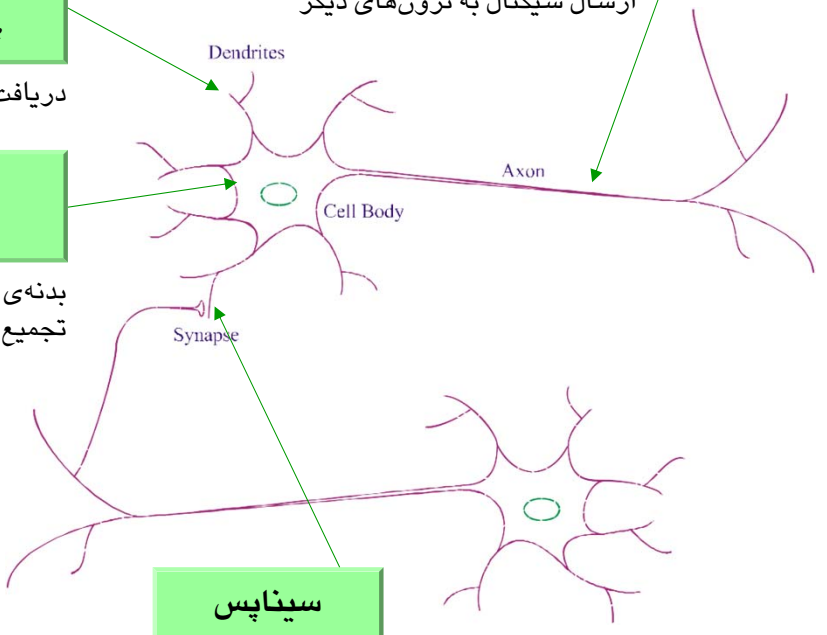
ارسال سیگنال به نرون‌های دیگر

دندریت
Dendrite

دریافت سیگنال از سایر نرون‌ها

سوما
Soma

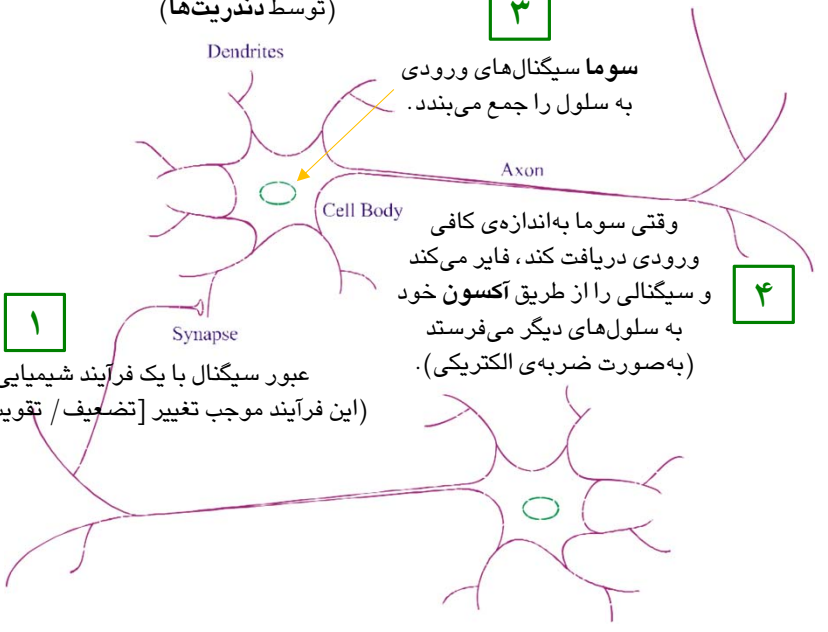
بدنه‌ی سلول:
تجمیع سیگنال‌های ورودی به سلول



سیناپس
Synapse

محل اتصال به آکسون نرون دیگر

عملکرد نرون طبیعی



شبکه‌ی عصبی مصنوعی

ARTIFICIAL NEURAL NETWORK (ANN)

شبکه‌ی عصبی مصنوعی:

یک سیستم پردازش اطلاعات

- دارای ویژگی‌هایی مشترک با شبکه‌ی عصبی طبیعی
- تعمیم‌یافته‌ی مدل ریاضی بازشناسی انسان بر اساس نروبیولوژی

فرضیات پایه‌ی ANN

- پردازش اطلاعات در نرون‌ها: اجزای ساده با تعداد فراوان
- انتقال سیگنال‌ها از طریق اتصالات بین نرون‌های شبکه
- هر اتصال یک وزن دارد (سیگنال انتقال‌یافته از آن در آن ضرب می‌شود.)
- هر نرون یک تابع فعال‌سازی (معمولاً غیرخطی) دارد:
  (این تابع بر روی ورودی‌های نرون اعمال می‌شود تا خروجی تولید شود.)

شبکه‌ی عصبی مصنوعی

ویژگی‌های مشخص‌کننده

ARTIFICIAL NEURAL NETWORK (ANN)

ویژگی‌های مشخص‌کننده‌ی یک ANN		
	ساختار / معماری شبکه *Network Architecture / Structure*	الگوی اتصال نرون‌ها
	تابع فعال‌سازی *Activation Function*	برای هر نرون می‌تواند متفاوت باشد.
	الگوریتم آموزش (یادگیری) *Training Algorithm (Learning)*	روش تعیین وزن‌های اتصالات شبکه

تطبیق شبکه‌های عصبی طبیعی با شبکه‌های عصبی مصنوعی

ANN vs. NNN

شبکه‌ی عصبی طبیعی *Natural Neural Network (NNN)*	شبکه‌ی عصبی مصنوعی *Artificial Neural Network (ANN)*
دندریت	اتصال بین نرون‌ها
تغییر سیگنال ورودی با عبور از فاصله‌ی سیناپسی	ضرب ورودی در وزن‌های شبکه
تجمیع سیگنال‌های ورودی در سوما	جمع وزن‌دار سیگنال‌های ورودی در نرون
فایر شدن سلول و ارسال سیگنال به آکسون	تولید خروجی توسط تابع فعال‌سازی

ویژگی‌های مشترک شبکه‌های عصبی طبیعی و شبکه‌های عصبی مصنوعی

تحمل‌پذیری نقص
Fault Tolerance

تعمیم‌پذیری
Generalization

مقاوم بودن در برابر نویز
Robustness

پردازش موازی
Parallel Processing

پردازش توزیع‌شده
Distributed Processing

حافظه‌ی توزیع‌شده
Distributed Memory

قابلیت یادگیری
Learning Capability

مدل نرون و معماری‌های شبکه

۱

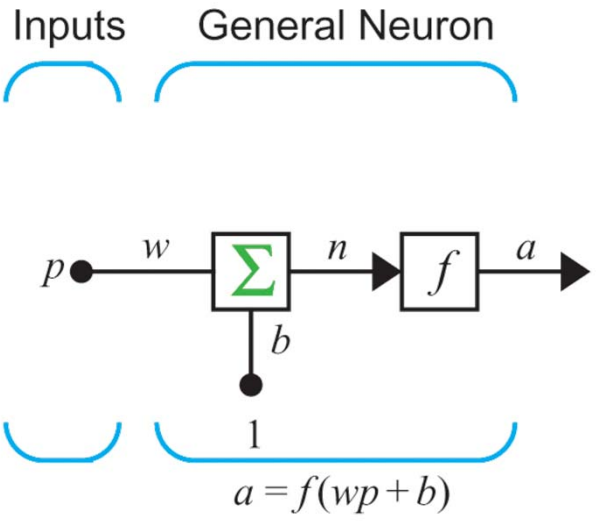
مدل نرون

نمادگذاری

NOTATION

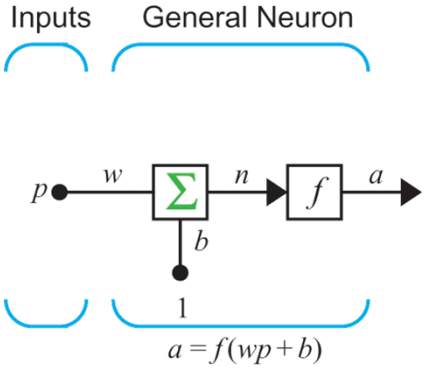
اسکالرها	ایتالیک کوچک	*a, b, c*
بردارها	سیاه کوچک	**a, b, c**
ماتریس‌ها	سیاه بزرگ	**A, B, C**

Single-Input Neuron



نرون تک–ورودی

SINGLE-INPUT NEURON



w	وزن	weight
p	ورودی (ادراک)	percept (input)
b	بایاس (آفست)	bias (offset)
$n = wp + b$	ورودی خالص	net input
f	تابع انتقال (فعال‌سازی)	transfer function (activation)
a	خروجی (کنش)	action (output)

$$a = f(wp + b) = f(n)$$

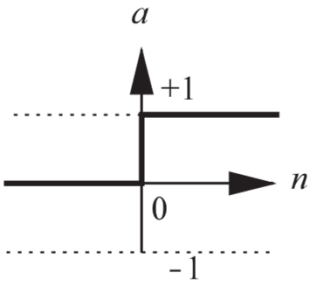
بایاس مانند یک وزن برای ورودی واحد 1 عمل می‌کند.

توابع انتقال

TRANSFER FUNCTIONS

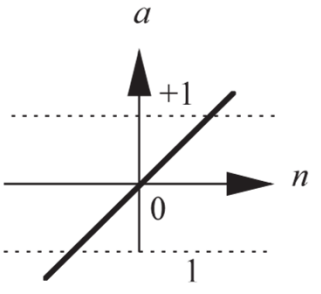
انواع مختلفی از توابع انتقال (خطی/غیرخطی) وجود دارد:
انتخاب تابع خاص بر اساس مشخصات مسئله‌ی مورد نظر برای نرون

حد سخت
Hard Limit



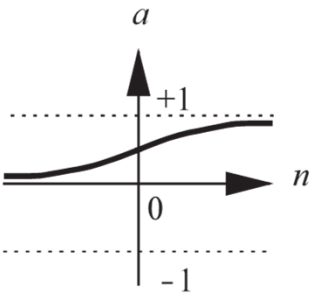
$a = hardlim(n)$

خطی
Linear



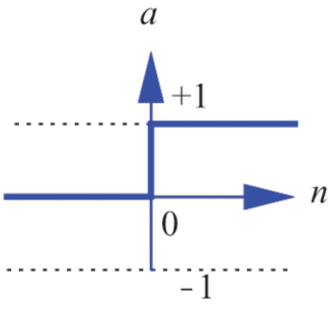
$a = purelin(n)$

سیگموئید لگاریتمی
Log-Sigmoid



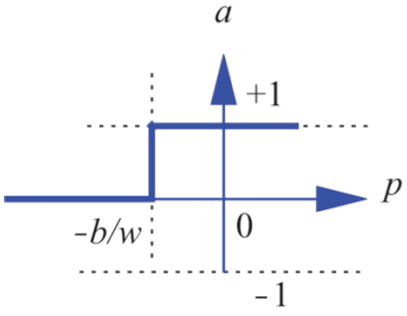
$a = logsig(n)$

Transfer Functions



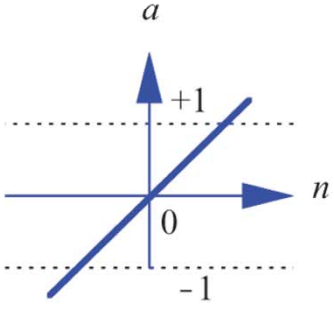
$a = hardlim(n)$

Hard Limit Transfer Function



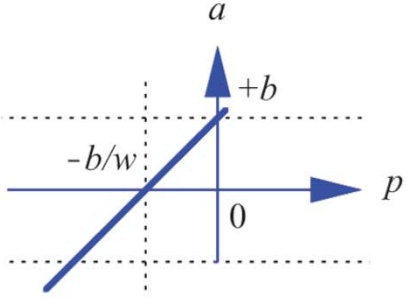
$a = hardlim(wp + b)$

Single-Input *hardlim* Neuron



$a = purelin(n)$

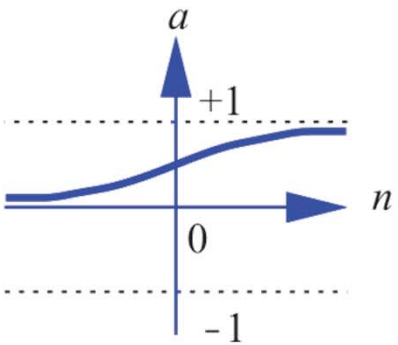
Linear Transfer Function

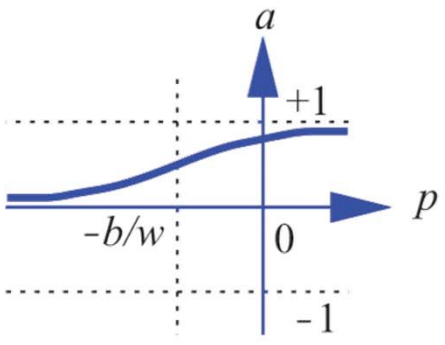


$a = purelin(wp + b)$

Single-Input *purelin* Neuron

Transfer Functions





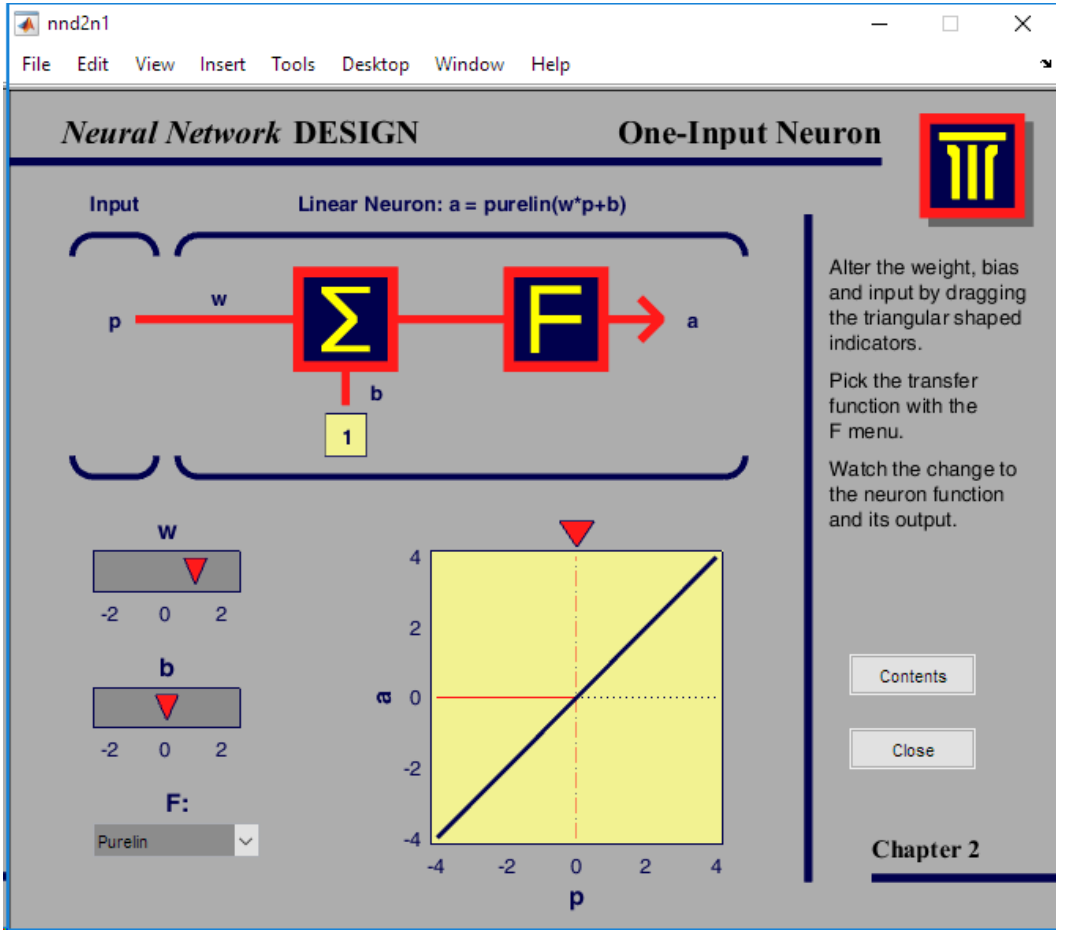
$a = logsig(n)$

Log-Sigmoid Transfer Function

$a = logsig(wp + b)$

Single-Input *logsig* Neuron

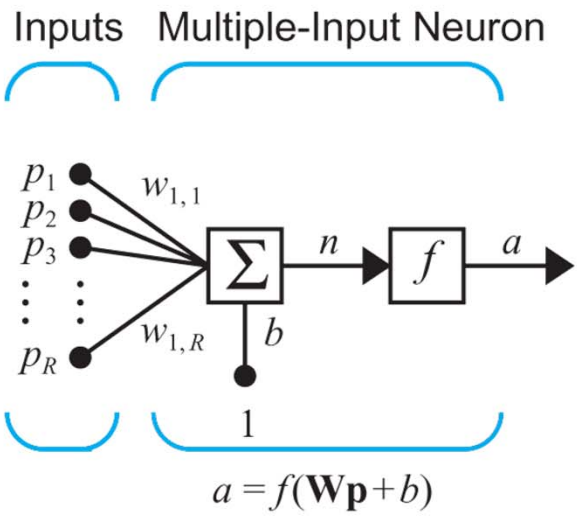
$$a = \frac{1}{1+e^{-n}}$$

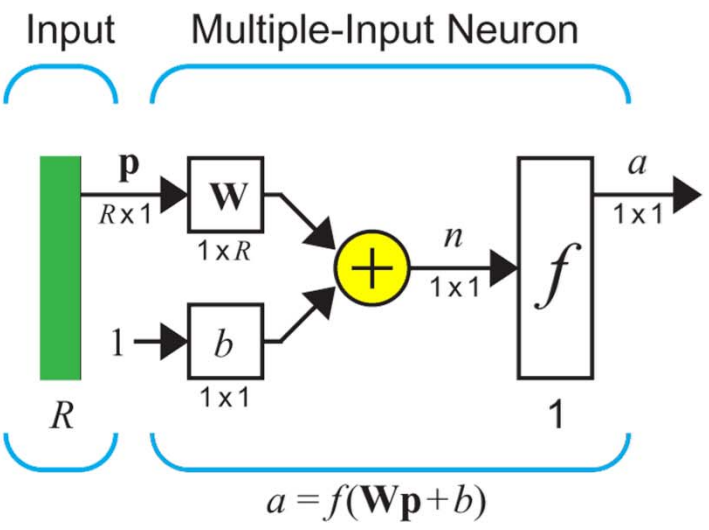


MATLAB

```
>> nnd2n1
```

Multiple-Input Neuron



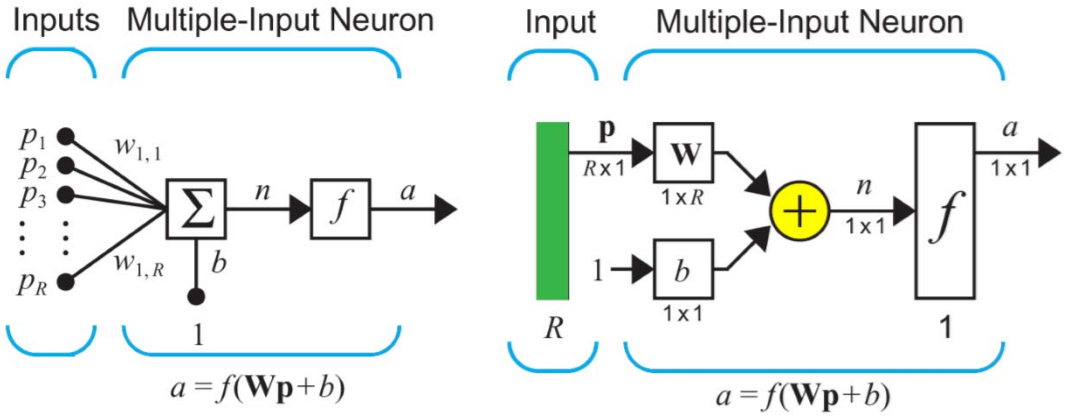


Abreviated Notation

نرون چند–ورودی

MULTIPLE-INPUT NEURON

R تعداد ورودی‌ها



نمادگذاری خلاصه‌ی شبکه به‌صورت برداری
(بیان ابعاد ماتریس‌ها و توابع زیر آنها)

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \cdots + w_{1,R}p_R + b$$

$$n = \mathbf{W}\mathbf{p} + b$$

$$\mathbf{W} = [w_{1,1}\ w_{1,2}\ \ldots\ w_{1,R}]_{1\times R} \qquad \mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix}_{R\times 1}$$

نمادگذاری

وزن

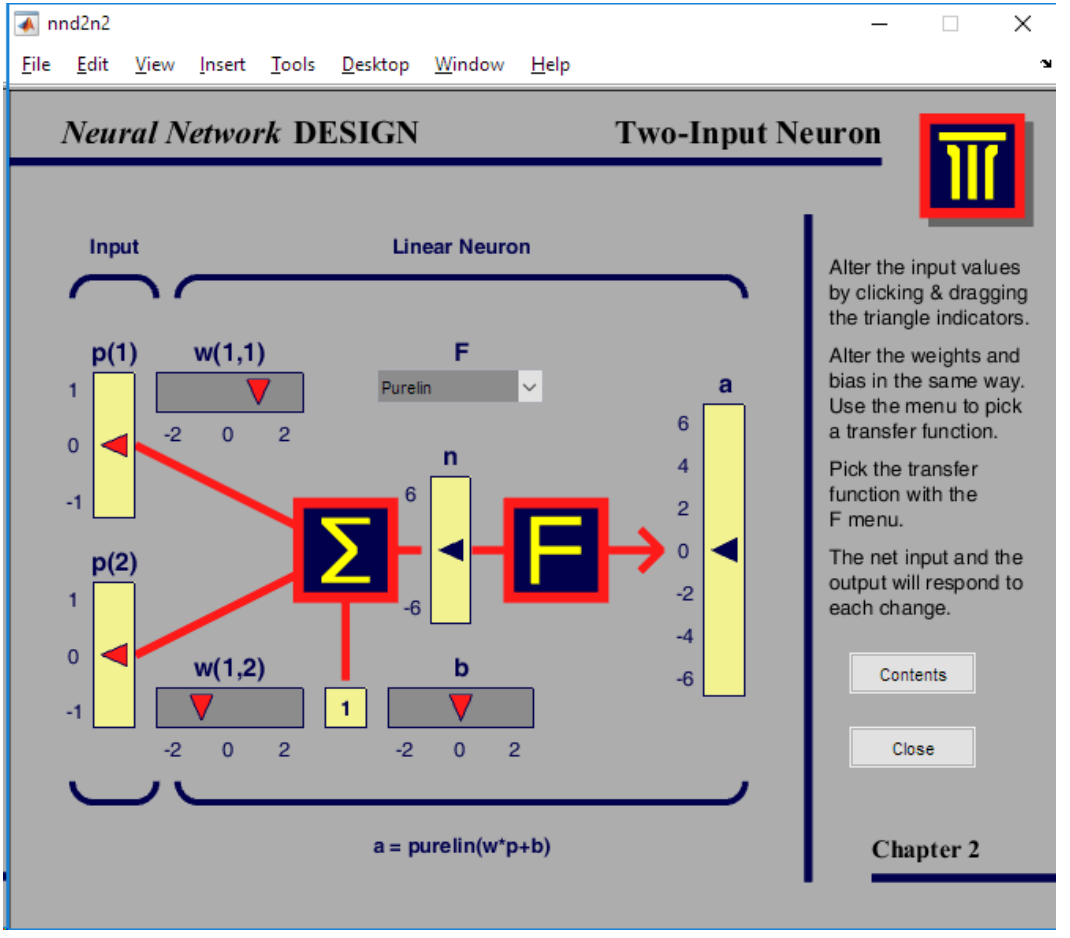
NOTATION

وزن ورودی j–ام نرون i–ام

$$w_{i,j}$$

اندیس اول: شماره‌ی نرون مقصد وزن

اندیس دوم: مبدأ سیگنال ورودی به نرون (شماره‌ی ورودی)


nnd2n2
File Edit View Insert Tools Desktop Window Help
Neural Network DESIGN
Two-Input Neuron
Input
Linear Neuron
p(1)
p(2)
w(1,1)
w(1,2)
F
Purelin
n
a
b
1
Alter the input values by clicking & dragging the triangle indicators.
Alter the weights and bias in the same way. Use the menu to pick a transfer function.
Pick the transfer function with the F menu.
The net input and the output will respond to each change.
Contents
Close
a = purelin(w*p+b)
Chapter 2

```
>> nnd2n2
```

شبکه‌های عصبی مصنوعی

مدل نرون و معماری‌های شبکه

۲

معماری‌های شبکه

معماری شبکه

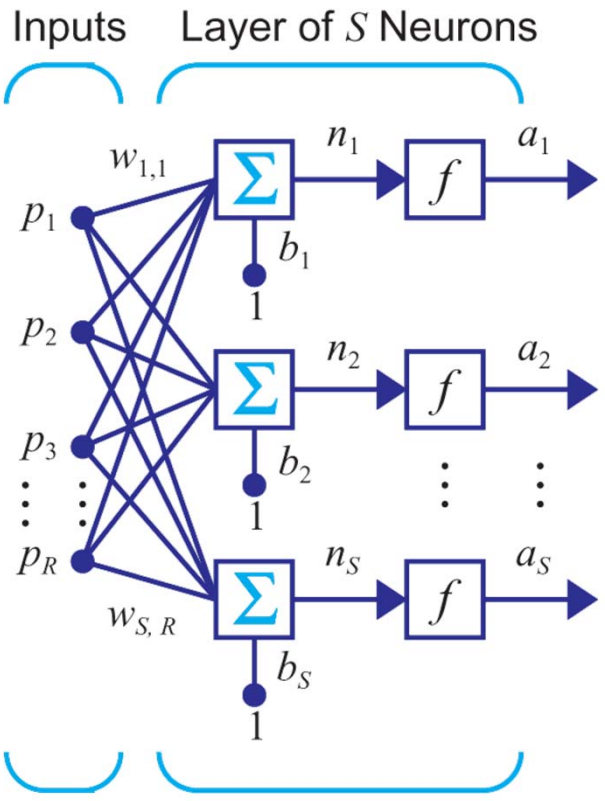
NETWORK ARCHITECTURE

معماری شبکه *Network Architecture*	چگونگی اتصال نرون‌ها به یکدیگر

معماری شبکه معمولاً از الگوی **لایه‌ای** پیروی می‌کند.

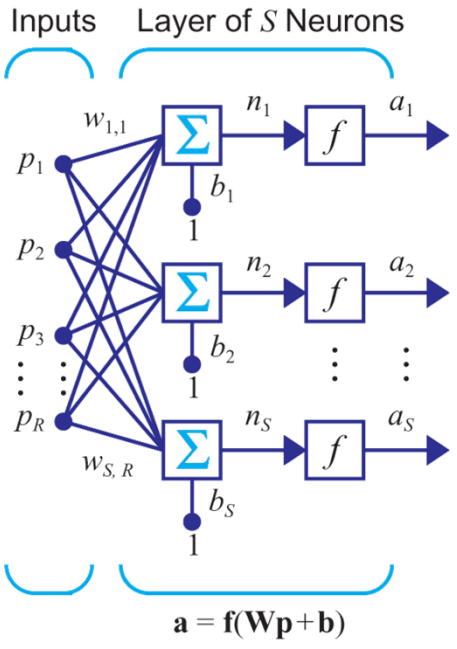
لایه *Layer*	مجموعه‌ای از نرون‌ها با ورودی‌های مشترک

Layer of Neurons



یک لایه از نرون‌ها

LAYER OF NEURONS



R	تعداد ورودی‌ها
S	تعداد نرون‌ها

لزوماً $S = R$ نیست.

$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p}+\mathbf{b})$$

$\mathbf{a}_{S\times 1}$

$\mathbf{f}_{S\times 1}$

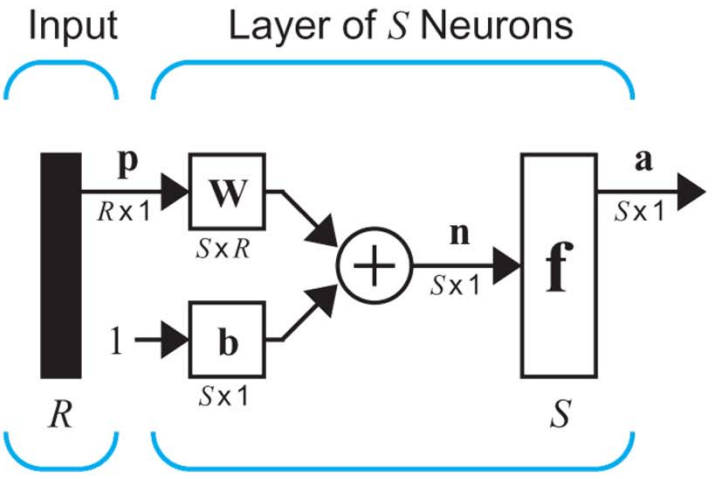
$\mathbf{W}_{S\times R}$ ← هر سطر مربوط به یک نرون؛ هر ستون مربوط به یک ورودی

$\mathbf{p}_{R\times 1}$

$\mathbf{b}_{S\times 1}$

در یک لایه، تابع انتقال هر نرون می‌تواند با نرون دیگر متفاوت باشد.

Abbreviated Notation



$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b})$$

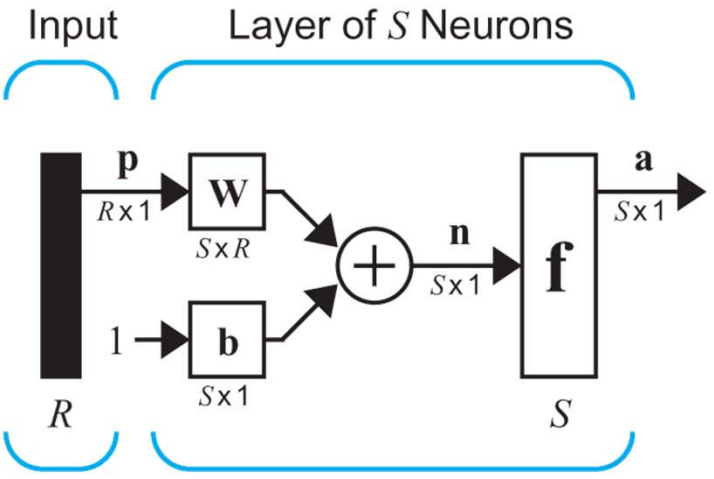
$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$

یک لایه از نرون‌ها

نمادگذاری خلاصه‌شده

LAYER OF NEURONS

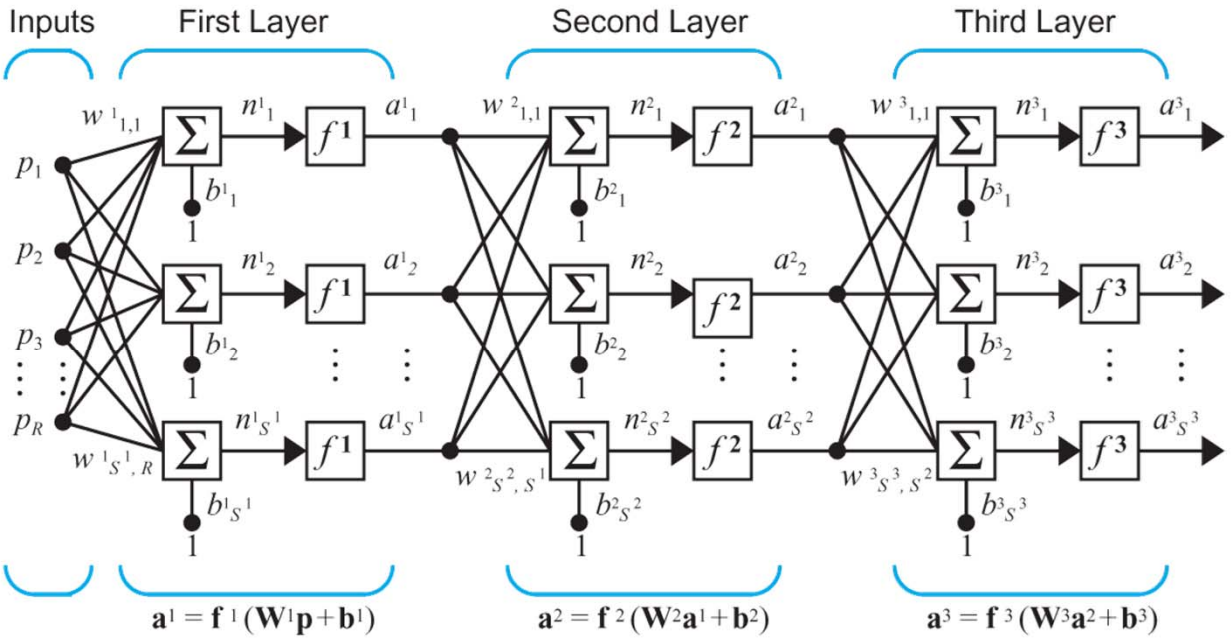


$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p}+\mathbf{b})$$

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$

Multilayer Network



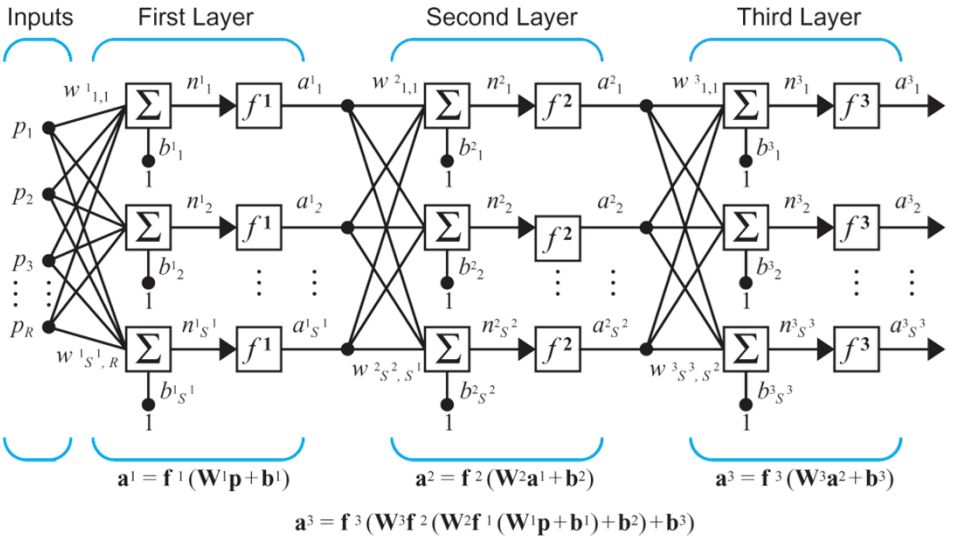
$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{W}^2\mathbf{a}^1+\mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{a}^2+\mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$$

چند لایه از نرون‌ها

شبکه‌های چند لایه

MULTIPLE LAYERS OF NEURONS



$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$$

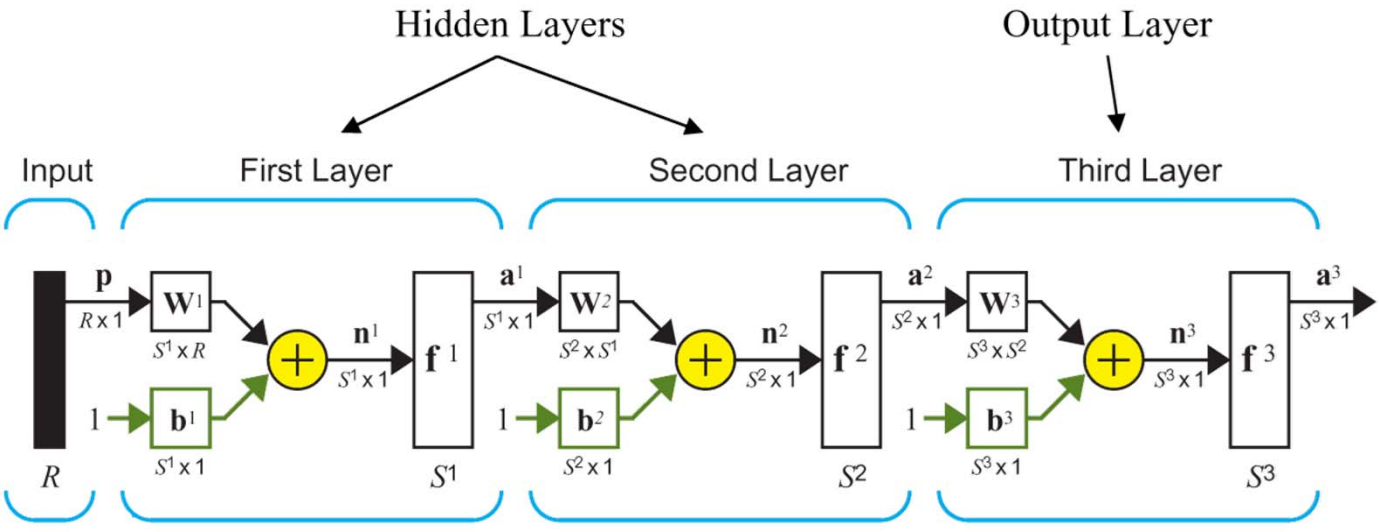
هر لایه ماتریس وزن خود را دارد: $\mathbf{W}^i$ = ماتریس وزن لایه‌ی i $\quad \mathbf{W}^1, \mathbf{W}^2, \mathbf{W}^3, \cdots$

هر لایه بـردار بایاس خود را دارد: $\mathbf{b}^i$ = بردار بایاس لایه‌ی i $\quad \mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3, \cdots$

ورودی هر لایه، خروجی لایه‌ی قبلی است. $\quad \mathbf{p}^i = \mathbf{a}^{i-1}, \quad \mathbf{p}^1 = \mathbf{p}$

تابع انتقال یک لایه می‌تواند با لایه‌ی دیگر متفاوت باشد.

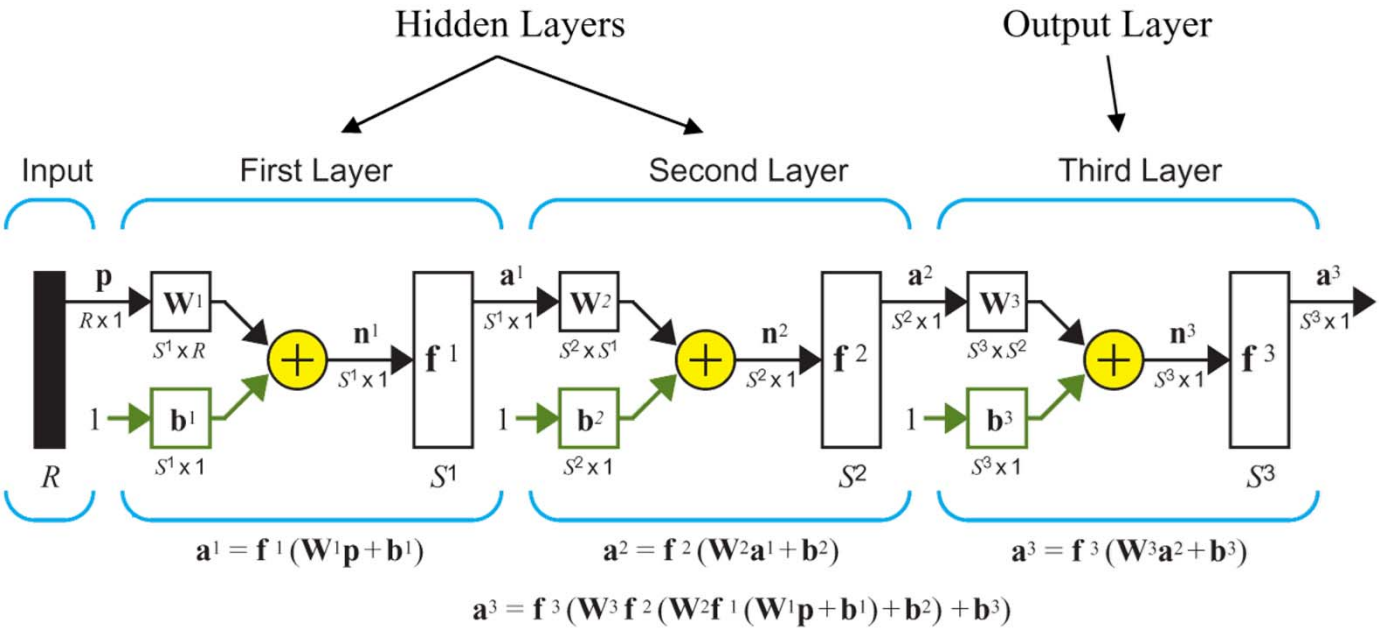
Abreviated Notation



چند لایه از نرون‌ها

شبکه‌های چند لایه: نمادگذاری خلاصه‌شده

MULTIPLE LAYERS OF NEURONS



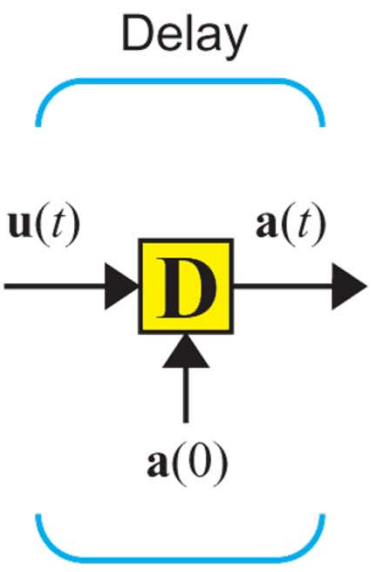
شبکه‌های چند لایه

MULTILAYER NETWORKS

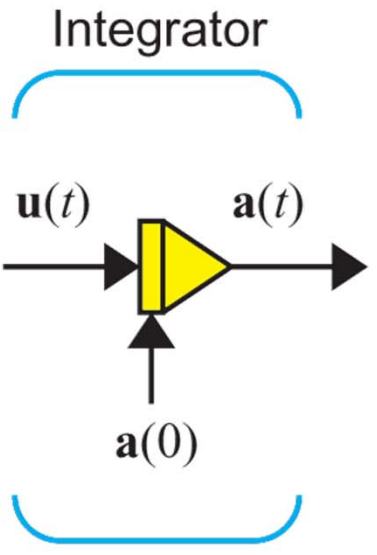
راهنماهایی برای مشخص‌سازی شبکه:

تعداد نرون‌های لایه‌ی اول	تعداد ورودی‌ها
تعداد نرون‌های لایه‌ی آخر	تعداد خروجی‌ها
توابع انتقال لایه‌ی خروجی	بر اساس مشخصه‌های مطلوب سیگنال خروجی
تعداد نرون‌های لایه‌ی مخفی	؟؟
تعداد لایه‌های مخفی	؟؟ (معمولاً ۲ یا ۳؛ ۴ به بالا در شبکه‌های عمیق)

Delays and Integrators



$$\mathbf{a}(t) = \mathbf{u}(t - 1)$$



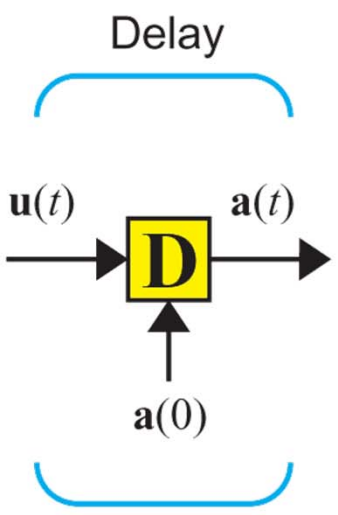
$$\mathbf{a}(t) = \int_0^t \mathbf{u}(\tau)\, d\tau + \mathbf{a}(0)$$

شبکه‌های بازگشتی

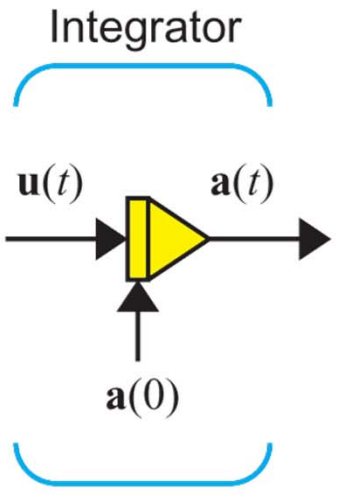
RECURRENT NETWORKS

شبکه‌های بازگشتی (دارای فیدبک)

در شبکه‌های بازگشتی از عناصر تأخیر و انتگرال‌گیر استفاده می‌شود:

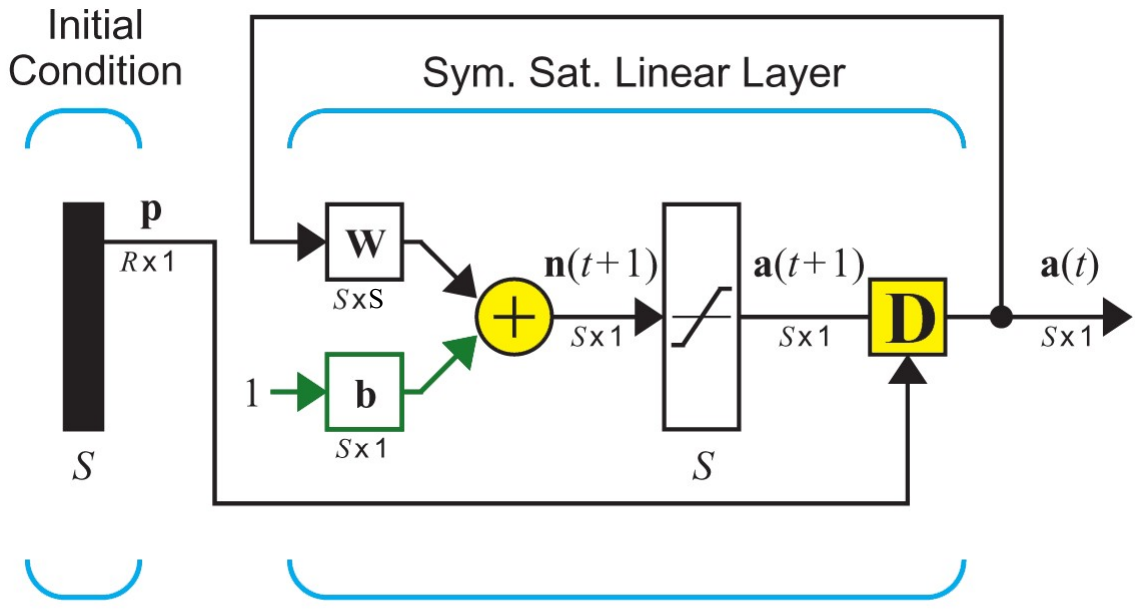


$$a(t) = u(t-1)$$



$$a(t) = \int_0^t u(\tau)\, d\tau + a(0)$$

Recurrent Network



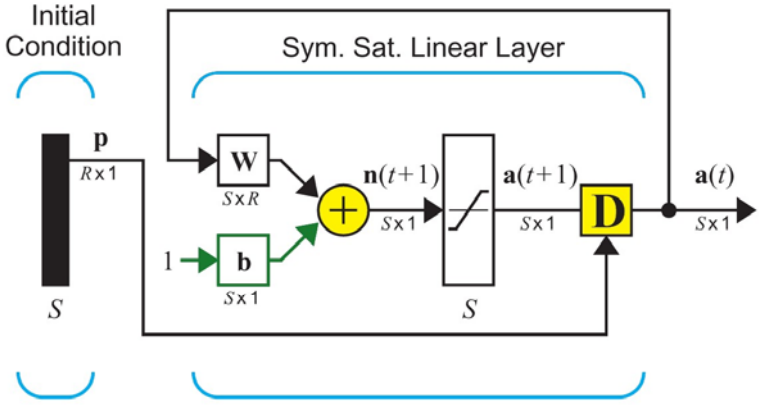
$$\mathbf{a}(0) = \mathbf{p} \qquad \mathbf{a}(t+1) = \mathbf{satlin}(\mathbf{W}\mathbf{a}(t)+\mathbf{b})$$

$$\mathbf{a}(1) = \mathbf{satlins}(\mathbf{W}\mathbf{a}(0)+\mathbf{b}) = \mathbf{satlins}(\mathbf{W}\mathbf{p}+\mathbf{b})$$

$$\mathbf{a}(2) = \mathbf{satlins}(\mathbf{W}\mathbf{a}(1)+\mathbf{b})$$

شبکه‌های بازگشتی

RECURRENT NETWORKS



$$\mathbf{a}(1) = \mathbf{satlins}(\mathbf{W}\mathbf{a}(0) + \mathbf{b}) = \mathbf{satlins}(\mathbf{W}\mathbf{p} + \mathbf{b})$$

$$\mathbf{a}(2) = \mathbf{satlins}(\mathbf{W}\mathbf{a}(1) + \mathbf{b})$$

شبکه‌های بازگشتی، دارای فیدبک (پس‌خور) است: برخی از خروجی‌ها به ورودی‌ها متصل می‌شوند.

شبکه‌های بازگشتی نسبت به شبکه‌های پیش‌خور به‌طور بالقوه قدرت‌مندتر هستند
و می‌توانند رفتارهای زمانی را نشان بدهند.

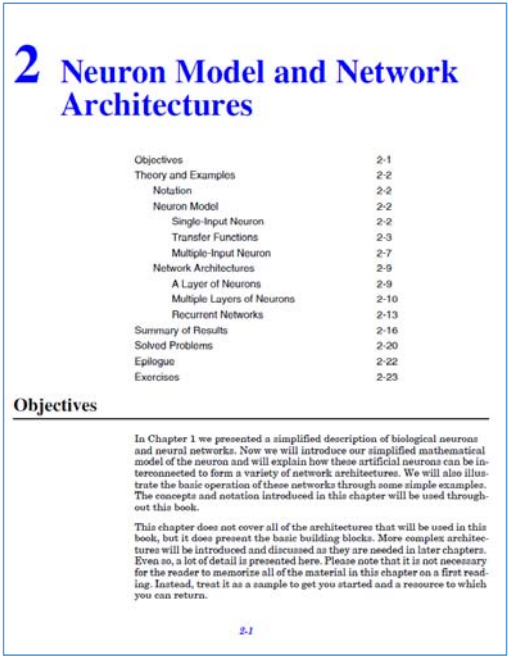
شبکه‌های عصبی مصنوعی

مدل نرون و معماری‌های شبکه

۳

منابع

منبع اصلی



2 Neuron Model and Network Architectures

Objectives

In Chapter 1 we presented a simplified description of biological neurons and neural networks. Now we will introduce our simplified mathematical model of the neuron and will explain how these artificial neurons can be interconnected to form a variety of network architectures. We will also illustrate the basic operation of these networks through some simple examples. The concepts and notation introduced in this chapter will be used throughout this book.

This chapter does not cover all of the architectures that will be used in this book, but it does present the basic building blocks. More complex architectures will be introduced and discussed as they are needed in later chapters. Even so, a lot of detail is presented here. Please note that it is not necessary for the reader to memorize all of the material in this chapter on a first reading. Instead, treat it as a sample to get you started and a resource to which you can return.

2-1

Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesus,
Neural Network Design,
2nd Edition, Martin Hagan, 2014.
Chapter 2

Online version can be downloaded from: http://hagan.okstate.edu/nnd.html