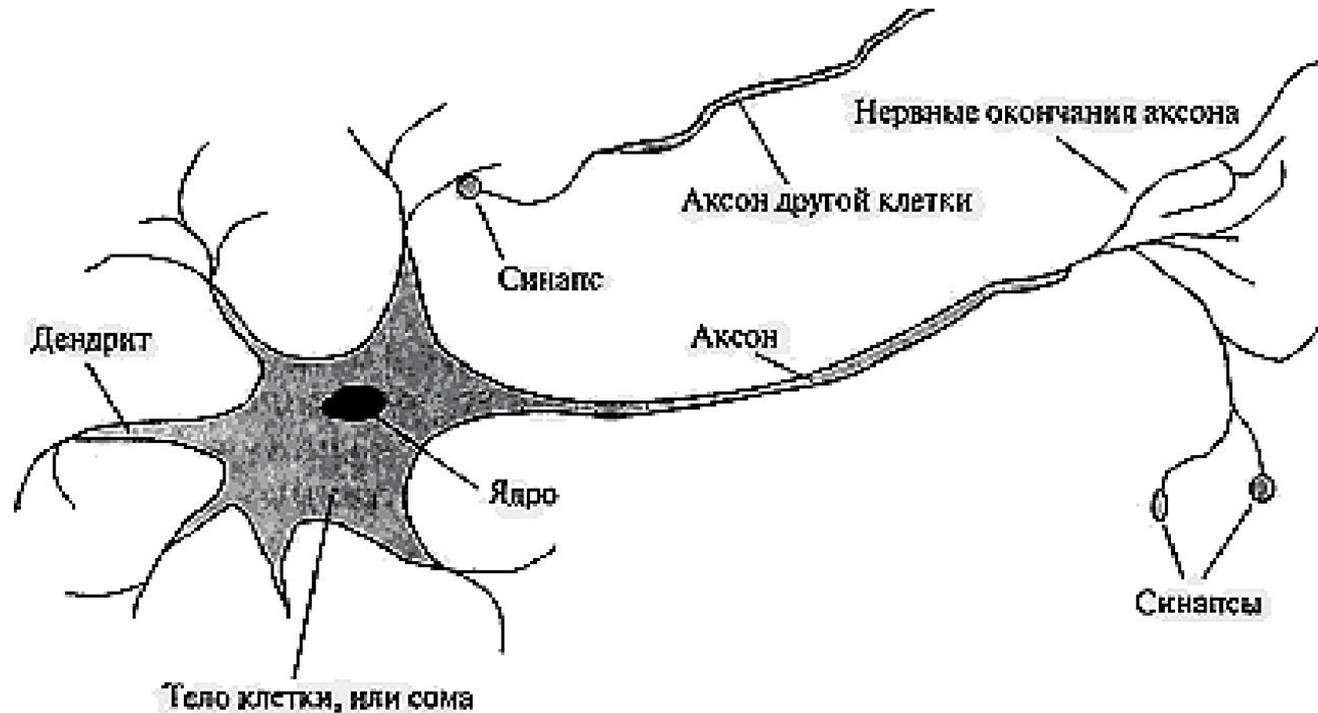
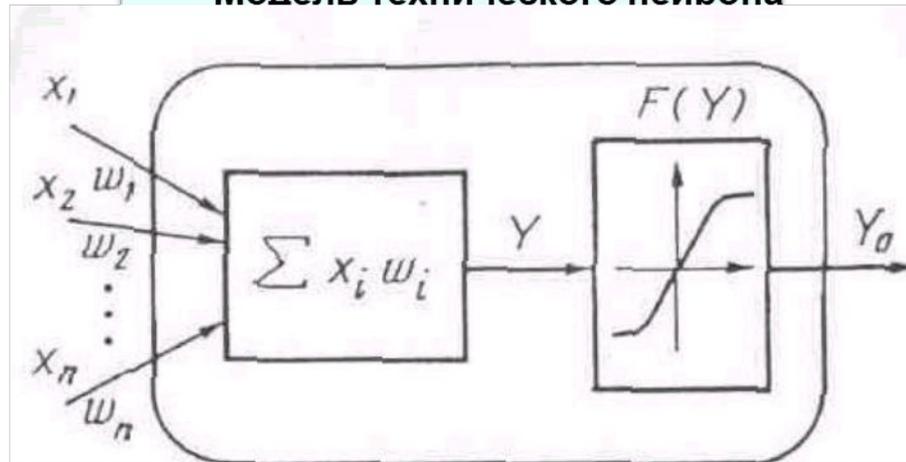


Нервная клетка или нейрон



Нейрон является особой биологической клеткой, которая обрабатывает информацию. Она состоит из тела клетки (cell body), или сомы (soma), и двух типов внешних древоподобных ветвей: аксона (axon) и дендритов (dendrites). Тело клетки включает ядро (nucleus), которое содержит информацию о наследственных свойствах, и плазму, обладающую молекулярными средствами для производства необходимых нейрону материалов. Нейрон получает сигналы (импульсы) от других нейронов через дендриты (приемники) и передает сигналы, сгенерированные телом клетки, вдоль аксона (передатчик), который в конце разветвляется на волокна (strands). На окончаниях этих волокон находятся синапсы (synapses), количество которых может составлять от 10 до 100 000.

Модель технического нейрона



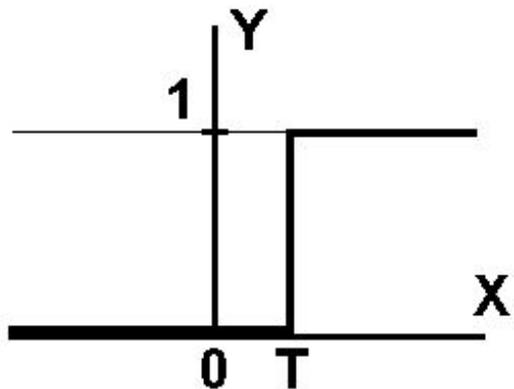
Нейрон — это элементарный преобразовательный элемент, имеющий непустое множество входов, на которые поступают сигналы x_1, x_2, \dots, x_n , суммирующий блок, блок преобразования сигнала с помощью активационной проекции и один выход Y_0 . Каждому входу приписан свой «вес» w_i (соответствующий «мере» биологической синаптической связи), который определяет силу и знак связи. Отрицательное значение веса соответствует подавлению активности соответствующего элемента, а положительной значение — усилению его активности. Функционирует нейрон в два такта, на первом такте в суммирующем блоке вычисляется величина возбуждения, полученного нейроном

$$S = \sum_{i=1}^n x_i w_i = (X, W),$$

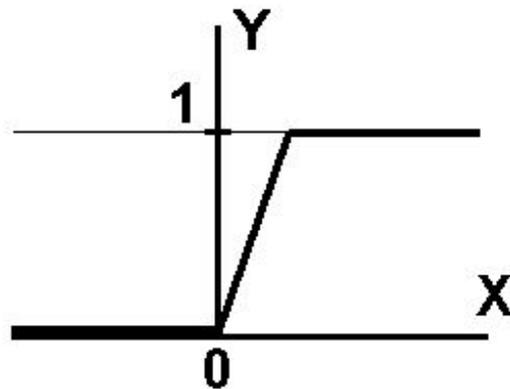
которую удобно представлять в виде скалярного произведения вектора входных сигналов на вектор весов. На втором такте суммарное возбуждение пропускается через активационную (преобразующую) функцию $F(\cdot)$ в результате чего получается выходной сигнал

$$Y_0 = F(S).$$

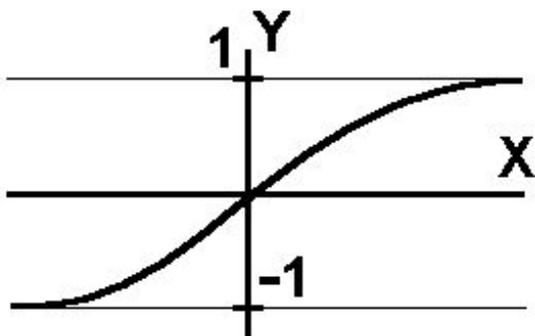
Параметрами нейрона, определяющими его работу, являются: вектор весов W , пороговый уровень и вид функции активации F .



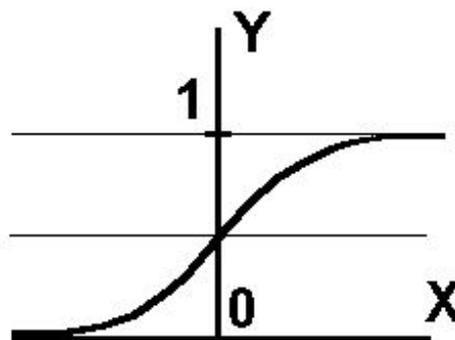
а)



б)



в)



г)

Виды функций активации: а) функция единичного скачка; б) линейный порог (гистерезис); в) гиперболический тангенс; г) сигмоид

ПРИМЕРЫ НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕМЫХ АКТИВАЦИОННЫХ ФУНКЦИЙ

1. пороговая (ступенчатая) функция

$$F(Y) = \{1, \text{ если } Y > \alpha; 0 - \text{ в противном случае } \},$$

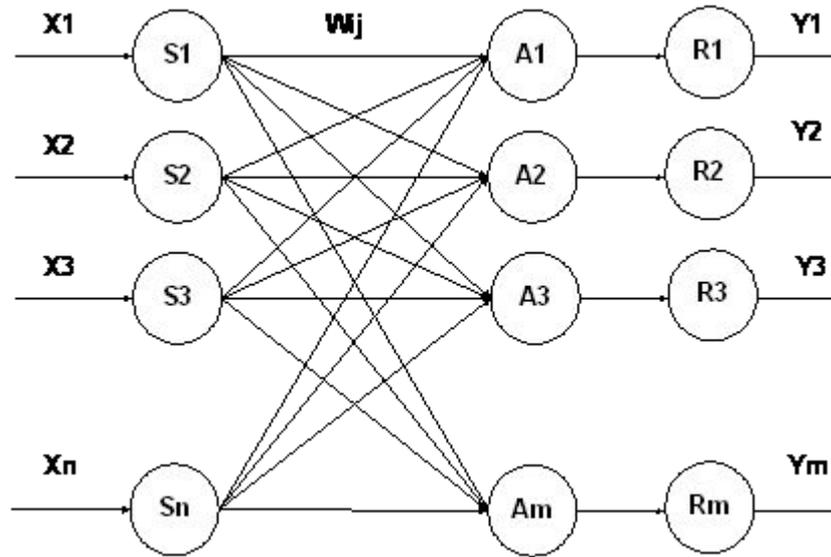
где α - некоторое пороговое значение;

2. сигмоидная (логистическая) функция

$$F(Y) = \frac{1}{1 + \exp(-aY)}$$

Мак-Калок и Питтс доказали, что при соответствующим образом подобранных весах совокупность параллельно функционирующих нейронов подобного типа способна выполнять универсальные вычисления. Здесь наблюдается определенная аналогия с биологическим нейроном: передачу сигнала и взаимосвязи имитируют аксоны и дендриты, веса связей соответствуют синапсам, а пороговая функция отражает активность сомы (тела клетки).

АРХИТЕКТУРА НЕЙРОСЕТИ

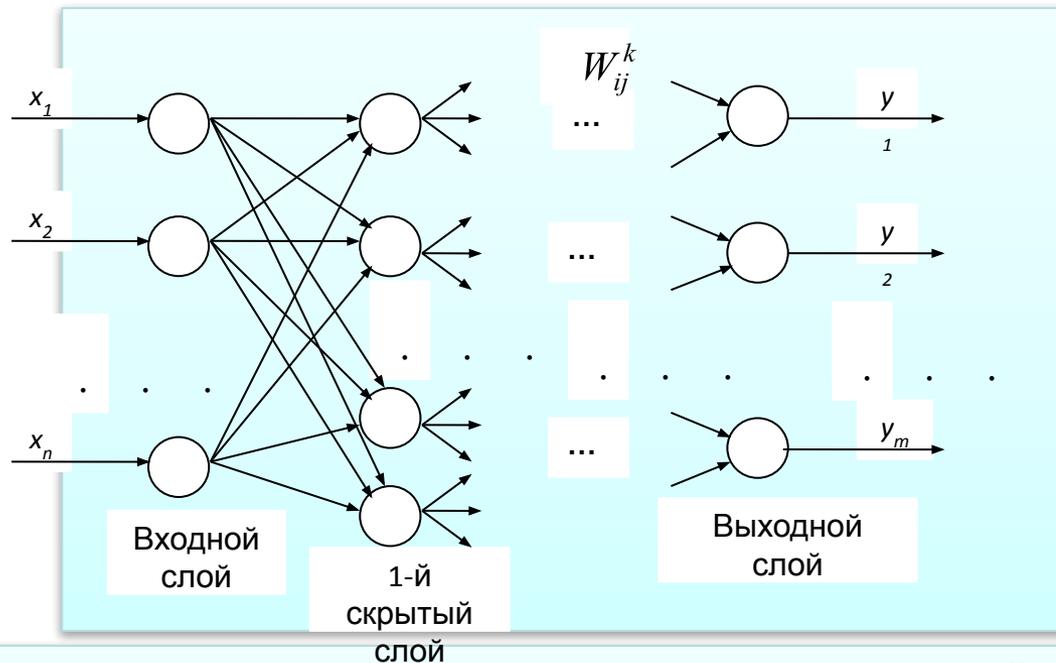


ОДНОСЛОЙНЫЙ ПЕРСЕПТРОН

Слой, обозначенный как $S_1 \dots S_n$, — это входной сенсорный слой. Его назначение состоит в том, чтобы воспринимать входные сигналы. Слой $A_1 \dots A_m$ называется ассоциативным. Именно здесь происходит непосредственная обработка информации. Слой $R_1 \dots R_m$ он называется эффекторным, и служит для передачи выходных воздействий. Особенностью этого слоя является использование в нейронах пороговой функции активации.

Персептроном, как правило, называют однослойную нейронную сеть, при этом каждый персептронный нейрон в качестве активационной функции использует функцию единичного скачка (пороговую).

СТРУКТУРА МНОГОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ



Многослойная сеть состоит из произвольного количества слоев нейронов. Нейроны каждого слоя соединяются с нейронами предыдущего и последующего слоев по принципу "каждый с каждым".

Количество нейронов (узлов) в слоях может быть произвольным, но обычно принимают во всех скрытых слоях одинаковое количество нейронов.

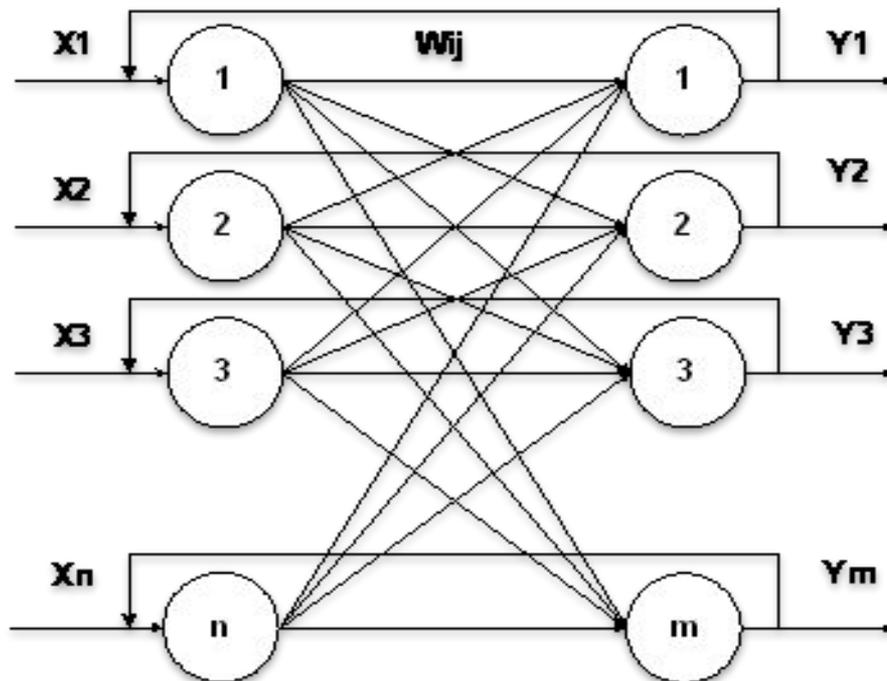
Многослойная сеть может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов.

По архитектуре связей НС могут быть сгруппированы в два класса: сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями.

РЕКУРРЕНТНАЯ СЕТЬ

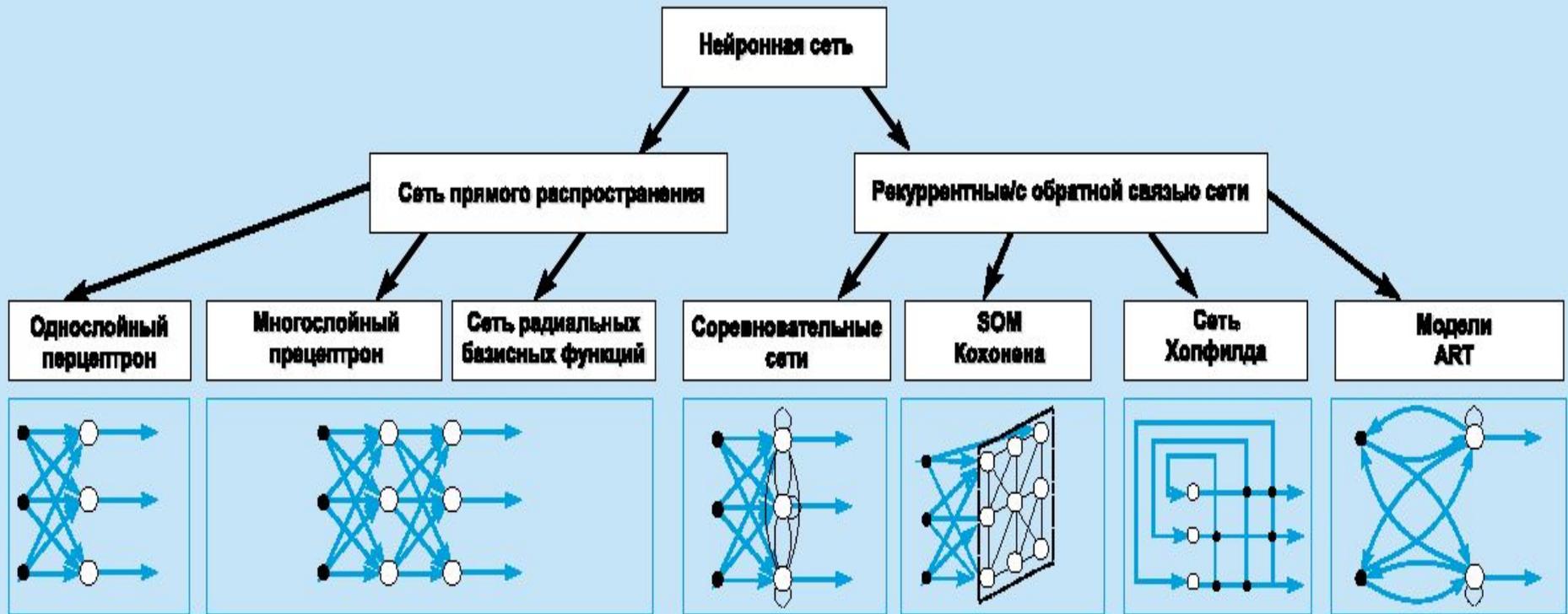
Рекуррентная сеть подает свои выходные данные обратно на свои собственные входы.

Это означает, что уровни активации сети образуют динамическую систему, которая может достигать устойчивого состояния, или переходить в колебательный режим, или даже проявлять хаотичное поведение. Более того, отклик сети на конкретные входные данные зависит от ее начального состояния, которое, в свою очередь, может зависеть от предыдущих входных данных. Поэтому рекуррентные сети (в отличие от сетей с прямым распространением) могут моделировать кратковременную память.

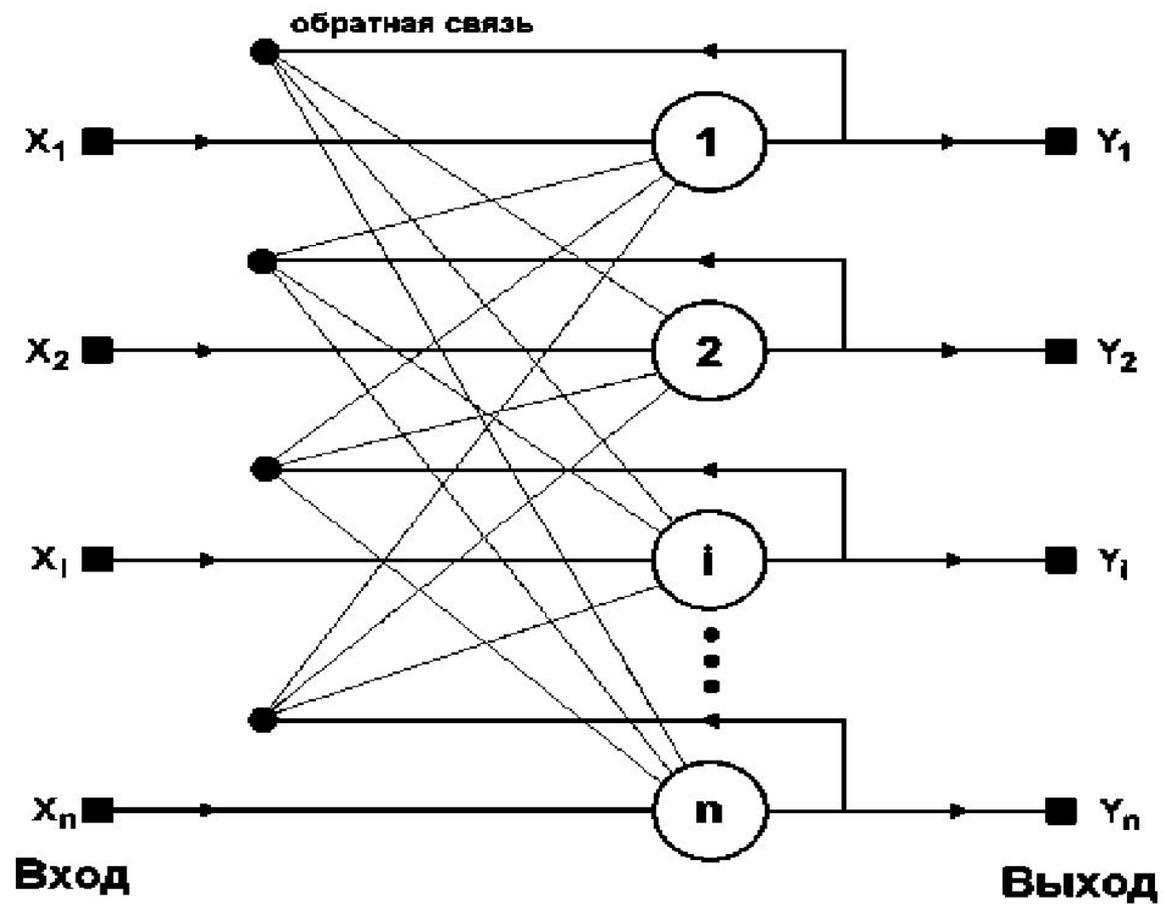


АРХИТЕКТУРА НС

По архитектуре связей НС могут быть сгруппированы в два класса : сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными СВЯЗЯМИ.



НЕЙРОННАЯ СЕТЬ ХОПФИЛДА



ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Суть процесса обучения НС заключается в выполнении следующей многошаговой процедуры:

Шаг 1. Задаётся обучающее множество (“задачник”), элементами которого являются обучающие пары

$$\{(X_1, D_1); (X_2, D_2); \dots; (X_L, D_L)\}$$

В данном случае – 1-й входной вектор (или 1-й входной образ), предъявляемый нейронной сети; 2-й – вектор эталонных (требуемых) реакций НС в ответ на 1-й входной вектор ; L – число различных обучающих пар.

Шаг 2. Устанавливается начальное состояние НС путём присваивания всем её весам некоторых случайных (малых) значений.

- Шаг 3. На вход сети подаётся входной вектор ; определяются реакции нейронов выходного слоя.
- Шаг 4. Вычисляется разность между желаемой реакцией сети и её фактическим выходом , т. е. ошибка сети, а также суммарная квадратичная ошибка .
- Шаг 5. Осуществляется коррекция весов нейронной сети таким образом, чтобы уменьшить ошибку .
- Шаг 6. Повторяются шаги 3–5 для каждой пары обучающего множества до тех пор, пока ошибка на всем множестве не достигнет малой, заранее заданной величины.

Результатом обучения является такая настройка весов синаптических связей, при которой каждому входному вектору сеть сопоставляет требуемый (или близкий к нему) выход.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

В среде прикладных программ MATLAB («Matrix Laboratory»), используемой для решения задач технических вычислений, пакет Neural Network Toolbox обеспечивает всестороннюю поддержку проектирования, обучения и моделирования множества известных нейросетевых структур, от базовых моделей персептрона до самых современных ассоциативных и самоорганизующихся сетей.

Neural Network Toolbox обеспечивает поддержку пакета Simulink, что позволяет моделировать нейросети и создавать блоки на основе разработанных нейросетевых структур.

ПРИМЕР ФОРМИРОВАНИЯ НЕЙРОСЕТЕВОЙ МОДЕЛИ

Формирование нейросетевых моделей. Основной функцией для формирования нейросетевых моделей в Simulink является функция **gensim**, записываемая в форме

gensim(net,st),

где **net** — имя созданной НС, **st** — интервал дискретизации (если НС не имеет задержек, ассоциированных с ее входами или слоями, значение данного аргумента устанавливается равным -1).

В качестве примера использования средств Simulink рассмотрим следующий.

Пусть входной и целевой векторы имеют вид

$$p = [1 \ 2 \ 3 \ 4 \ 5];$$

$$t = [1 \ 3 \ 5 \ 7 \ 9].$$

es	Type	Size	Date Modified

```
-- 21.10.09 18:31 --
-- 27.10.09 17:13 --
neural
p=[1 2 3 4 5];
t=[1 3 5 7 9];
net=newlind(p,t);
y=sim(net.p);
y=sim(net,p)
gensim(net,-1)
p=[1 2 3 4 5];
t=[1 3 5 7 9];
net=newlind(p,t);
y=sim(net,p)
```

To get started, select [MATLAB Help](#) or [Demos](#) from the Help menu.

```
>> p=[1 2 3 4 5];
>> t=[1 3 5 7 9];
>> net=newlind(p,t);
>> y=sim(net,p)

y =

    1.0000    3.0000    5.0000    7.0000    9.0000

>> |
```

Workspace

es	Type	Size	Date Modified
----	------	------	---------------

Command Window

```
To get started, select MATLAB Help or Demos from the Help menu.  
  
>> p=[1 2 3 4 5];  
>> t=[1 3 5 7 9];  
>> net=newlind(p,t);  
>> y=sim(net,p)  
  
y =  
  
1.0000 3.0000 5.0000 7.0000 9.0000  
  
>> gensim(net,-1)  
>>
```

Library: neural

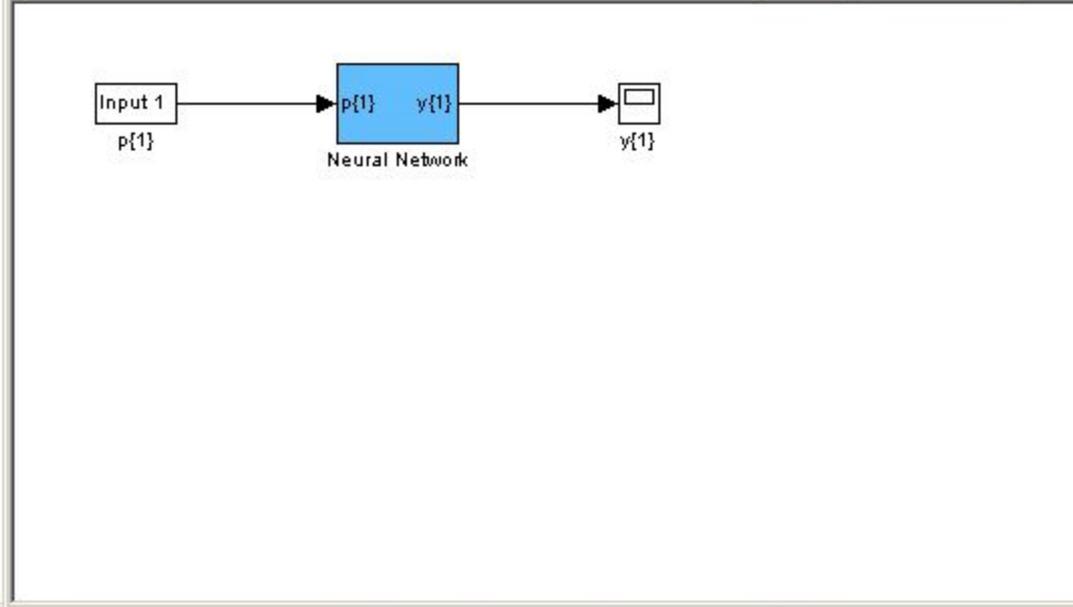
File Edit View Format Help

Transfer Functions Net Input Functions Weight Functions

Control Systems

untitled *

File Edit View Simulation Format Tools Help



```
t=[1 3 5 7 9];  
net=newlind(p,t);  
y=sim(net,p)  
gensim(net,-1)
```

Workspace

es	Type	Size	Date Modified
----	------	------	---------------

Command Window

To get started, select [MATLAB Help](#) or [Demos](#) from the Help menu.

```
>> p=[1 2 3 4 5];  
>> t=[1 3 5 7 9];  
>> net=newlind(p,t);  
>> y=sim(net,p)
```

y =

1.0000	3.0000	5.0000	7.0000	9.0000
--------	--------	--------	--------	--------

```
gensim(net,-1)
```

Source Block Parameters: p{1}

constant

Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.

Constant value:

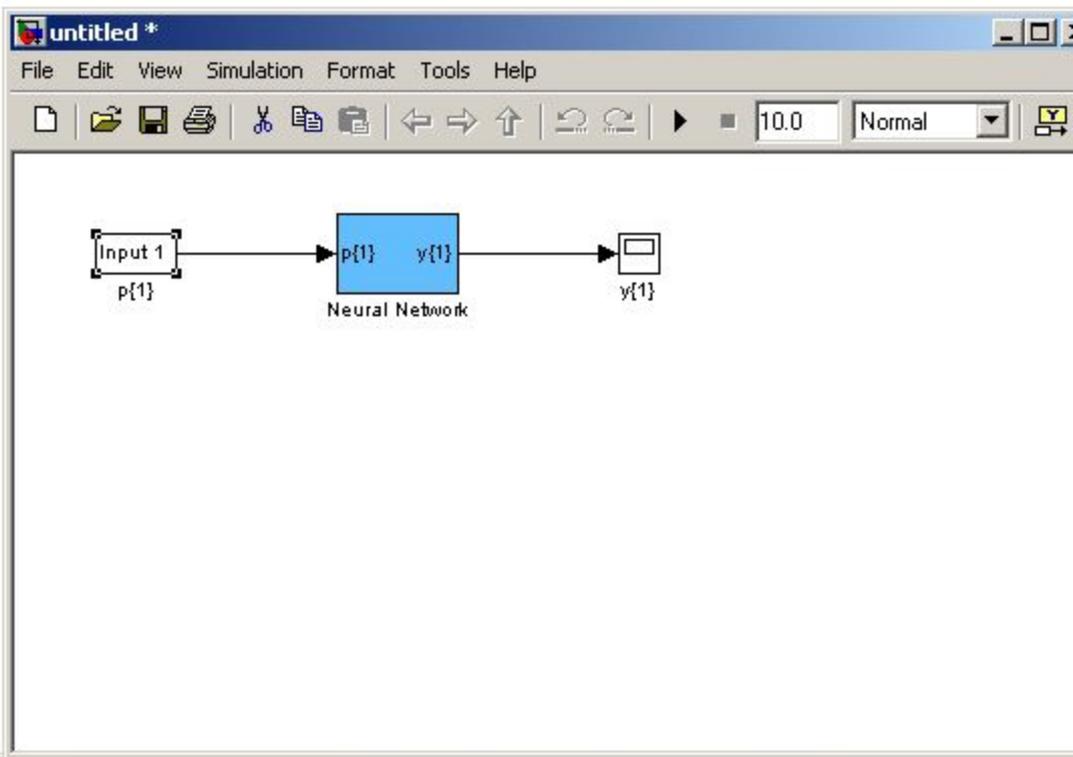
Interpret vector parameters as 1-D

Sampling mode: Sample based

Sample time:

OK Cancel Help

```
net=newlind(p,t);  
y=sim(net,p)  
gensim(net,-1)
```



Workspace

es	Type	Size	Date Modified
----	------	------	---------------

Command Window

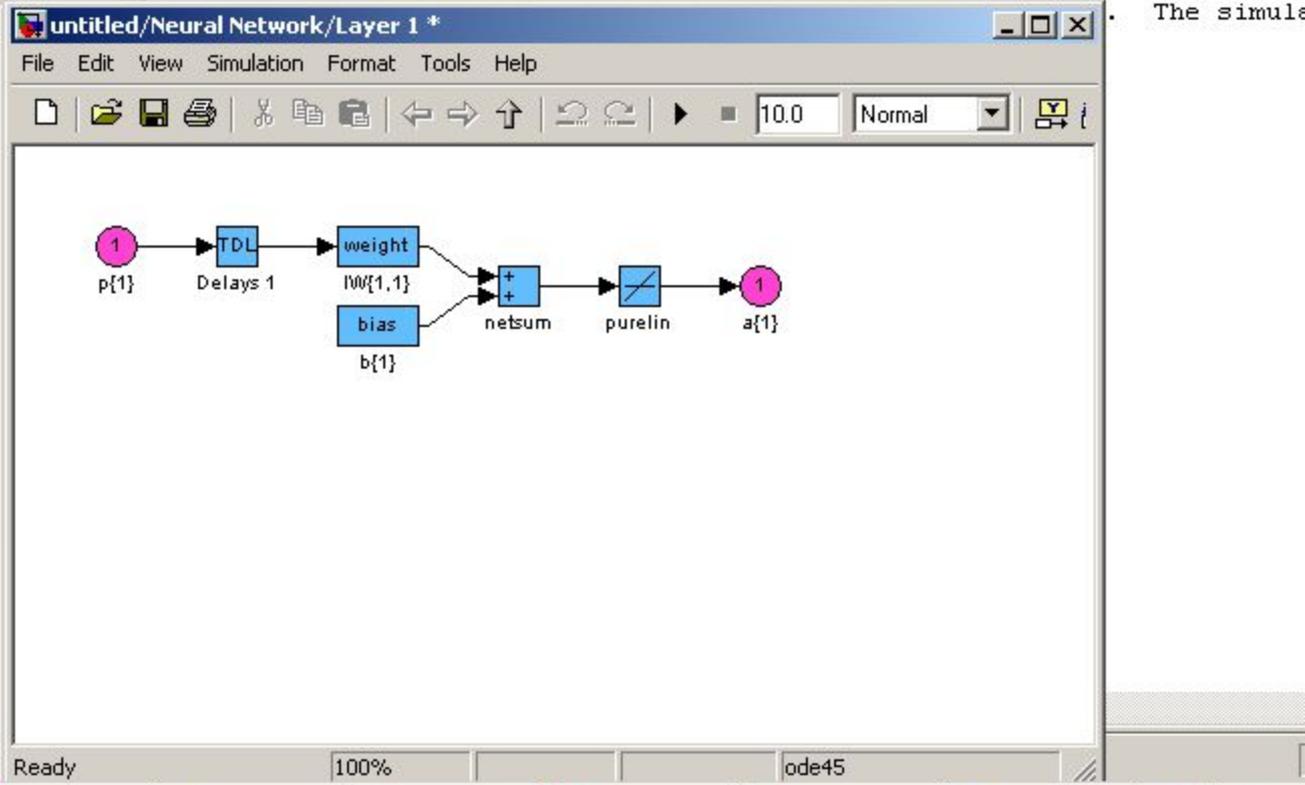
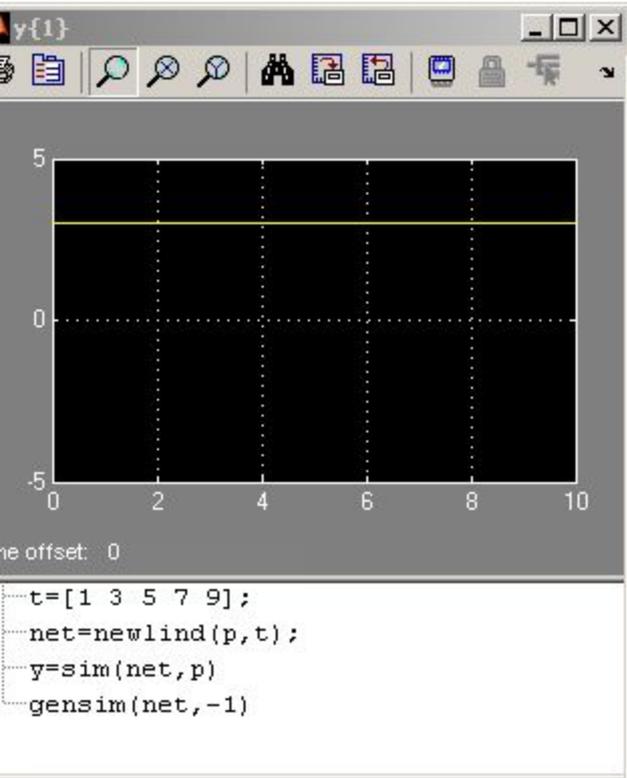
To get started, select [MATLAB Help](#) or [Demos](#) from the Help menu.

```
>> p=[1 2 3 4 5];  
>> t=[1 3 5 7 9];  
>> net=newlind(p,t);  
>> y=sim(net,p)
```

y =

1.0000	3.0000	5.0000	7.0000	9.0000
--------	--------	--------	--------	--------

```
>> gensim(net,-1)  
Warning: The model 'untitled' does not have continuous states, hence using ode45. The simulation will stop at the end of the simulation.
```



С созданной сетью можно проводить различные эксперименты, возможные в среде Simulink; вообще с помощью команды **gensim** осуществляется интеграция созданных нейросетей в блок-диаграм-

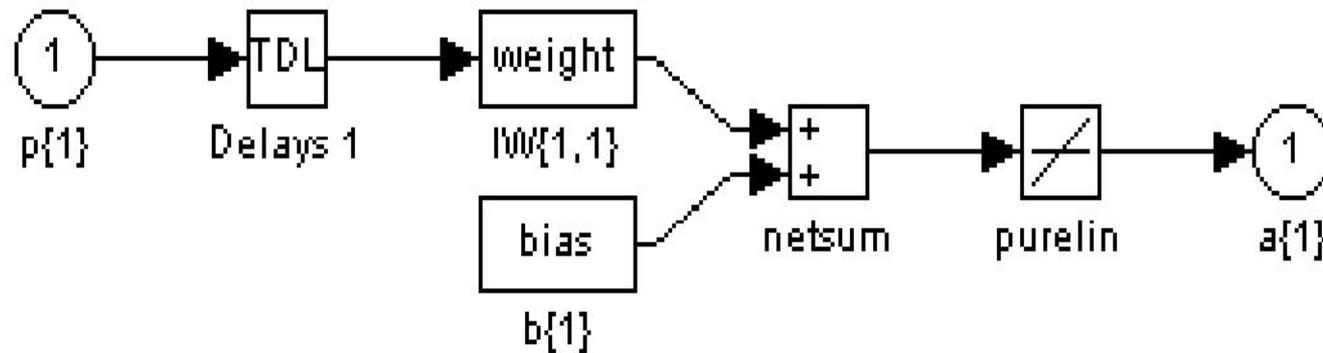


Рис. 4.30. Структура созданной НС

мы этого пакета — с использованием имеющихся при этом инструментов моделирования различных систем (например, встраивание нейросетевого регулятора в систему управления и моделирование последней и т. п.).

Search for: Go

Example: "plot tools" OR plot* tools

Contents Index Search Results Demos

- [-] Fuzzy Logic
- [-] GARCH
- [-] Genetic Algorithm and Direct S
- [-] Image Acquisition
- [-] Image Processing
- [-] Instrument Control
- [-] Link for Cadence Incisive
- [-] Link for Code Composer Studi
- [-] Link for ModelSim
- [-] Mapping
- [-] Model Predictive Control
- [-] Neural Network
 - [-] Neurons
 - [-] Simple neuron and transfer
 - [-] Neuron with vector input
 - [-] Perceptrons
 - [-] Linear Networks
 - [-] Backpropagation
 - [-] Radial Basis Networks
 - [-] Self-organizing Networks
 - [-] LVQ Networks
 - [-] Hopfield Networks
 - [-] Application Examples
 - [-] Control Systems
 - [-] Other Demos
- [-] OPC
- [-] Optimization
- [-] Partial Differential Equation
- [-] RF
- [-] Robust Control
- [-] Signal Processing



Title: Neural Network Demo: nnd2n2

Open nnd2n2.m in the Editor

Run this de

Neuron with vector input

nnd2n2
_ □ ×

File Edit View Insert Tools Desktop Window Help

Neural Network DESIGN

Two-Input Neuron

Input

p(1)

1

0

-1

w(1,1)

-2 0 2

F

Hardlim

n

6

-6

a

6

4

2

0

-2

-4

-6

p(2)

1

0

-1

w(1,2)

-2 0 2

1

b

-2 0 2

Σ

F

$a = \text{hardlim}(w^*p + b)$

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

Chapter 2

У п р а в л е н и е. Для динамической системы, заданной совокупностью $\{u(t), y(t)\}$, $u(t)$ является входным управляющим воздействием, а $y(t)$ - выходом системы в момент времени t . В системах управления с эталонной моделью целью управления является расчет такого входного воздействия $u(t)$, при котором система следует по желаемой траектории, диктуемой эталонной моделью. Примером является оптимальное управление двигателем.

СПЕЦИАЛИЗИРОВАННЫЕ ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ

- Промышленные нейроимитаторы, такие как Neural Works Professional II+ фирмы Neural Ware, или MultiNeuron, разработанный в Красноярском научном центре.
- Программный пакет Neuro Office (разработчик — ЗАО «АльфаСистем»).
- OWL (разработчик — Hyper Logic Соф.).
- Neuro Windows (разработчик — Ward Systems Group).
- NNet+ (разработчик — NeuroMetric Vision System).
- STATISTICA Neural Networks - нейронно-сетевой пакет фирмы StatSoft.