

MATLAB & Toolboxes

Вход

Ю.П.Маслобоев "Свойства и параметры нейронной сети как объекта MATLAB"

При разработке Neural Network Toolbox использованы принципы объектно-ориентированного программирования. Основной объект – нейронная сеть. В этом разделе описаны основные свойства сети.

Архитектура сети

Структура подобъектов

Функции

Параметры

Значения весов и смещений

Другие свойства

Ю.П.Маслобоев "Свойства подобъектов нейронной сети"

Входы

Слои

Выходы

Эталоны

Смещения

Входные веса

Веса слоев

MATLAB & Toolboxes

Вход

Ю.П.Маслобоев "Свойства и параметры нейронной сети как объекта MATLAB" Архитектура сети

Эти свойства определяют количество подобъектов сети (таких как слои, выходы, эталоны, смещения и веса), и каким образом они объединяются в сеть.

- **numInputs** - определяет количество входов сети.

net.numInputs - может принимать значения 0 или положительное число.

Пояснение.

Количества входов сети и размерность входа разные вещи. Количество входов определяет, сколько групп векторов принимает сеть в качестве входов. Размерность каждого входа (т.е. количество элементов в каждом векторе) определяется размерностью входа (net.inputs{i}.size). Большинство сетей имеет один вход, размерность которого определяется поставленной задачей.

Сопутствующие эффекты.

Любое изменение этого свойства приводит к изменению размера матрицы, определяющей связи входов со слоями, (net.inputConnect) и размер массива ячеек входных подобъектов (net.inputs).

- **numLayers** - определяет количество слоев сети.

net.numLayers - может принимать значения от 0 до любого положительного целого.

Сопутствующие эффекты.

Любое изменение этого свойства приводит к изменению размера каждой из булевых матриц, которые определяют связи между слоями:

net.biasConnect

net.inputConnect

net.layerConnect

net.outputConnect

net.targetConnect

и изменяют размерности каждого из массивов ячеек структур подобъектов, чьи размеры зависят от количества слоев:

net.biases

net.inputWeights

net.layerWeights

net.outputs

net.targets

А также изменяет размеры каждого из настраиваемых параметров:

net.IW

net.LW

net.b

- **biasConnect** - определяет, какие слои имеют смещения.

net.biasConnect - может принимать значения Булевой матрицы $N_l \times 1$, где N_l – количество слоев (net.numLayers). Наличие (или отсутствие) смещения у i -го слоя определяется 1 (или 0) в: net.biasConnect(i)

Сопутствующие эффекты.

Любое изменение этого свойства влияет на наличие или отсутствие структур в массиве ячеек смещений (net.biases) и, в наличии или отсутствии в массиве элементов векторов смещений (net.b).

- **inputConnect** - определяет, какие слои обладают связями со входами.

net.inputConnect – определяет Булеву матрицу $N_l \times N_i$, где N_l – количество слоев сети (net.numLayers), а N_i – количество входов сети (net.numInputs). Наличие (или отсутствие) связи i – го слоя с j – м слоем определяется 1 (или 0) в net.inputConnect(i,j).

Сопутствующие эффекты.

Любое изменение этого свойства влияет на наличие или отсутствие структур в массивах ячеек подобъектов входных весов (net.inputWeights) и на наличие или отсутствие матриц в массиве ячеек входных весов (net.IW).

- **layerConnect** - определяет какие слои связаны с другими слоями.

net.layerConnect - определяет Булеву матрицу $N_l \times N_l$, где N_l – количество слоев в сети (net.numLayers). Наличие (или отсутствие) связи к l -го слоя с j -м слоем определяется 1 (или 0) в net.layerConnect(i,j).

Сопутствующие эффекты.

Любое изменение этого свойства влияет на наличие или отсутствие структур в массивах ячеек подобъектов весов слоев (net.layerWeights) и в наличии или отсутствии матриц в массиве ячеек весов слоев (net.LW).

- **outputConnect** - определяет какие слои генерируют выходы сети.

net.outputConnect – задается в виде матрицы Булевых значений $1 \times N$, где N – количество слоев в сети (net.numLayers). Наличие (или отсутствие) выхода от i -го слоя определяется 1 (или 0) в net.outputConnect(i).

Сопутствующие эффекты.

Любые изменения этого свойства будут влиять на количество выходов сети (net.numOutputs) и наличие или отсутствие структур в массиве ячеек выходных подобъектов (net.outputs).

- **targetConnect** - определяет какие слои связаны с эталонами.

net.targetConnect - задается в виде матрицы Булевых значений $1 \times N$, где N – количество слоев в сети (net.numLayers). Наличие (или отсутствие) эталонного значения, связанного с i -м слоем определяется 1 (или 0) в net.targetConnect(i).

Сопутствующие эффекты.

Любые изменения этого свойства будут изменять количество эталонных значений сети (net.numTargets) и наличие или отсутствие структур в массиве ячеек эталонных подобъектов (net.targets).

- **numOutputs** (только для чтения) - указывает сколько выходов имеет сеть.

net.numOutputs - возвращает значение, соответствующее количеству единиц в матрице выходных соединений: numOutputs=sum(net.outputConnect).

- **numTargets** (только для чтения) - указывает сколько эталонных значений имеет сеть.

net.numTargets - возвращает значение, соответствующее количеству единиц в матрице соединений с эталонами: numTargets=sum(net.targetConnect).

- **numInputDelays** (только для чтения) - указывает какое количество временных шагов последних входов должно быть реализовано для того, чтобы смоделировать сеть.

net.numInputDelays - возвращает значение величины максимальной задержки, связанной с входными весами сети:

```
numInputDelays = 0;
```

```
for i=1:net.numLayers
```

```
for j=1:net.numInputs
```

```
if net.inputConnect(i,j)
```

```
numInputDelays = max( ...
```

```
[numInputDelays net.inputWeights{i,j}.delays]);
```

```
end
```

```
end
```

```
end
```

- **numLayerDelays** (только для чтения) - указывает какое количество временных шагов последних выходов должно быть реализовано для того, чтобы смоделировать сеть.

net.numLayerDelays - возвращает значение величины максимальной задержки, связанной с весами слоев сети:

```
numLayerDelays = 0;
```

```
for i=1:net.numLayers
```

```
for j=1:net.numLayers
```

```
if net.layerConnect(i,j)
```

```
numLayerDelays = max( ...
```

```
[numLayerDelays net.layerWeights{i,j}.delays]);
```

```
end
```

```
end
```

```
end
```

Ю.П.Маслобоев "Свойства и параметры нейронной сети как объекта MATLAB" Структура подобъекто

Эти свойства определяют массивы ячеек структур, которые определяют каждый из входов сети, слои, выходы, эталоны, смещения и веса.

- **inputs** - содержит структуры свойств для каждого из входов сети.

`net.inputs` - матрица $N_i \times 1$ ячеек входных структур, где N_i - число входов сети (`net.numInputs`).

Структура, определяющая свойства i -го сетевого входа, определена в: `net.inputs{i}`

- **layers** - содержит структуры свойств для каждого из слоев сети.

`net.layers` - массив $N_l \times 1$ ячеек входных структур, где N_l - число слоев сети (`net.numLayers`).

Структура, определяющая свойства i -го слоя определена в: `net.layers{i}`.

- **outputs** - содержит структуры свойств для каждого из выходов сети.

`net.outputs` - массив $1 \times N_o$ ячеек входных структур, где N_o - число слоев сети (`net.numLayers`).

Структура, определяющая свойства i -го выхода, определена в: `net.outputs{i}`

если соответствующее выходное соединение `net.outputConnect(i)`- 1 (или 0):.

- **targets** - содержит структуры свойств для каждого из эталонов сети.

`net.targets` – массив $1 \times N_t$ ячеек входных структур, где N_t - число слоев сети (`net.numLayers`).

Структура, определяющая свойства эталона связанного с i -ым слоем (или нулевая матрица) определена в: `net.targets{i}` если соответствующее выходное соединение `net.targetConnect(i)` - 1 (или 0).

- **biases** - содержит структуры свойств для каждого из смещений сети.

`net.biases` - массив $N_b \times 1$ ячеек, где N_b - число слоев сети (`net.numLayers`).

Структура, определяющая свойства смещений i -го слоя (или нулевая матрица) расположена в: `net.biases{i}` если соответствующее соединение для смещения `net.biasConnect(i)` - 1 (или 0).

- **inputWeights** – содержит структуры свойств для каждого из входных весов.

`net.inputWeights` - массив $N_i \times N_l$ ячеек, где N_l - число слоев сети (`net.numLayers`), а N_i - число входов сети (`net.numInputs`).

Структура, определяющая свойства весов к i -му слоя от j го слоя (или нулевая матрица) определена в: `net.inputWeights{i,j}`, если соответствующее соединение входа `net.inputConnect(i,j)` - 1 (или 0).

- **layerWeights** - содержит структуры свойств для каждого из весов слоя сети.

`net.layerWeights` - массив $N_l \times N_l$ ячеек, где N_l - число слоев сети (`net.numLayers`).

Структура, определяющая свойства связей i -го слоя с j м слоем (или нулевая матрица) определена в: `net.layerWeights{i,j}`, если соответствующая связь слоя `net.layerConnect(i,j) - 1` (или 0).

MATLAB & Toolboxes

Ю.П.Маслобоев "Свойства и параметры нейронной сети как объекта MATLAB" Функции

Эти свойства определяют алгоритмы, которые используются при адаптации, инициализации, тренировке и оценке ее функционирования.

- **adaptFcn** - определяет функцию, которая будет использована для адаптации сети.

net.adaptFcn - задается в виде названия функции адаптации сети из настоящего тулбокса:

trains – функция адаптации весов и смещений.

Функция адаптации сети используется, чтобы выполнить адаптацию всякий раз, когда вызывается adapt:

```
[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)
```

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, параметры адаптации сети (net.adaptParam) устанавливаются таким образом, чтобы содержать параметры и значения по умолчанию новой функции.

- **initFcn** - определяет функцию, используемую для инициализации матриц весов и векторов смещений сети.

net.initFcn - задается в виде названия функции инициализации сети из настоящего тулбокса:

initlay - функция инициализации весов и смещений.

Функция инициализации сети используется, чтобы выполнить инициализацию всякий раз, когда вызывается init:

```
net = init(net)
```

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, параметры сети (net.initParam) устанавливаются таким образом, чтобы содержать параметры и значения по умолчанию новой функции.

- **performFcn** – определяет функцию, используемую для оценки функционирования сети.

net.performFcn - задается в виде названия функции функционирования сети из настоящего тулбокса:

Функции функционирования

mae	Средняя абсолютная ошибка
mse	Средняя квадратичная ошибка
msereg	Средняя квадратичная ошибка w/reg
sse	Суммарная квадратичная ошибка

Функция функционирования сети используется, чтобы вычислить оценку функционирования сети всякий раз, когда вызывается train:

```
[net,tr] = train(NET,P,T,Pi,Ai)
```

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, параметры сети (net.performParam) устанавливаются таким образом, чтобы содержать параметры и значения по умолчанию новой функции.

- **trainFcn** - определяет функцию, используемую для тренировки сети.

net.trainFcn - задается в виде названия функции тренировки сети из настоящего тулбокса:

Функции тренировки

trainb	Пакетная тренировка с использованием правил обучения для весов и смещений
trainbfg	Тренировка сети с использованием квази –Ньютоновского метода BFGS
trainbr	Регуляризация Bayesian
trainc	Использование приращений циклического порядка
traincgb	Метод связанных градиентов Пауэлла-Била (Powell-Beale)
traincgf	Метод связанных градиентов Флетчера-Пауэлла (Fletcher-Powell)
traincgp	Метод связанных градиентов Полака-Рибера (Polak-Ribiere)
traingd	Метод градиентного спуска
traingda	Метод градиентного спуска с адаптивным обучением
traingdm	Метод градиентного спуска с учетом моментов
traingdx	Метод градиентного спуска с учетом моментов и с адаптивным обучением
trainlm	Метод Левенберга-Маркара (Levenberg-Marquardt)
trainoss	Одноступенчатый метод текущих
trainr	Метод случайных приращений
trainrp	Алгоритм упругого обратного распространения
trains	Метод последовательных приращений

Функция используется, чтобы тренировать сеть всякий раз, когда вызывается train:

```
[net,tr] = train(NET,P,T,Pi,Ai)
```

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, параметры сети (net.trainParam) устанавливаются таким образом, чтобы содержать параметры и значения по умолчанию новой функции.

Ю.П.Маслобоев "Свойства и параметры нейронной сети как объекта MATLAB"

Параметры

- **adaptParam** - определяет параметры и значения текущей функции адаптации.

net.adaptParam - поля этого свойства зависят от текущей функции адаптации (net.adaptFcn). Вызовите вышеупомянутую ссылку, чтобы увидеть поля текущей функции адаптации.

Вызовите help на текущую функцию адаптации, чтобы получить описание каждого из полей: help(net.adaptFcn).

- **initParam** - определяет параметры и значения текущей функции инициализации.

net.initParam - поля этого свойства зависят от текущей функции инициализации (net.initFcn). Вызовите вышеупомянутую ссылку, чтобы увидеть поля текущей функции адаптации.

Вызовите help на текущую функцию инициализации, чтобы получить описание каждого из полей: help(net.initFcn)

- **performParam** - определяет параметры и значения текущей функции функционирования.

net.performParam - поля этого свойства зависят от текущей функции функционирования (net.performFcn). Вызовите вышеупомянутую ссылку, чтобы увидеть поля текущей функции функционирования.

Вызовите help на текущую функцию функционирования, чтобы получить описание каждого из полей: help(net.performFcn)

- **trainParam** - определяет параметры и значения текущей функции тренировки.

net.trainParam - поля этого свойства зависят от текущей функции тренировки (net.trainFcn). Вызовите вышеупомянутую ссылку, чтобы увидеть поля текущей функции функционирования.

Вызовите help на текущую функцию тренировки, чтобы получить описание каждого из полей: help(net.trainFcn)

Математика\Neural Network Toolbox

Ю.П.Маслобоев "Свойства и параметры нейронной сети как объекта MATLAB" Значения весов и смещений

Эти свойства определяют модифицируемые параметры сети: матрицы весов и векторы смещений.

- **IW** - определяет матрицы входных весов.

`net.IW` – массив $N_i \times N_i$ ячеек, где N_i - число слоев в сети (`net.numLayers`), а N_i - число входов сети (`net.numInputs`).

Матрица весов для связи i го слоя с j м входом (или нулевая матрица []) описывается в: `net.IW{i,j}`, если соответствующее входное соединение. `net.inputConnect(i,j) - 1` (или 0).

Матрица весов имеет количество строк, равное размеру слоя, к которому она относится (`net.layers{i}.size`). Содержит количество столбцов равное произведению размерности входа на количество задержек, ассоциированных с весом:

`net.inputs{j}.size * length(net.inputWeights{i,j}.delays)`

Эти размеры могут также быть получены из свойств входных весов:

`net.inputWeights{i,j}.size`

- **LW** - определяет матрицы весов связей между слоями.

`net.LW` - массив $N_i \times N_i$ ячеек, где N_i - число слоев в сети (`net.numLayers`).

Матрица весов для связи i го слоя. с j м слоем (или нулевая матрица []) представлена в: `net.LW{i,j}`, если соответствующее соединение слоя `net.layerConnect(i,j) - 1` (или 0).

Матрица весов имеет количество строк, равное размеру слоя, к которому она относится (`net.layers{i}.size`). Содержит количество столбцов равное произведению размерности слоя на количество задержек, ассоциированных с весом:

`net.layers{j}.size * length(net.layerWeights{i,j}.delays)`

Эти размеры могут также быть получены из свойств веса слоя:

`net.layerWeights{i,j}.size`

- **b** - определяет векторы смещений для каждого слоя со смещением.

`net.b` - массив $N_i \times 1$ ячеек, где N_i - число слоев в сети (`net.numLayers`).

Вектор смещения для i -го слоя (или нулевая(пустая) матрица []) определен в `net.b{i}` если соответствующее соединение смещения `net.biasConnect(i) - 1` (или 0).

Число элементов в векторе смещения всегда равно размеру слоя, с которым он связан (`net.layers{i}.size`). Этот размер может также быть получен из свойств смещения: `net.biases{i}.size`.

Ю.П.Маслобоев "Свойства и параметры нейронной сети как объекта MATLAB" Другие свойства

Единственное другое свойство - свойство для данных пользователя.

- **userdata** - Это свойство обеспечивает место для пользователей, чтобы добавить пользовательскую информацию о сети.

net.userdata

Только одно поле предопределено. Оно содержит секретное сообщение всем пользователям Neural Network Toolbox:

net.userdata.note

MATLAB & Toolboxes

Ю.П.Маслобоев "Свойства подбъектов нейронной сети" Входы

Эти свойства определяют структуру i-го сетевого входа. - `net.inputs{i}`

- **range** - определяет диапазоны изменения каждого из элементов i-го входа сети

`net.inputs{i}.range` - значение этого свойства может быть установлено в виде матрицы $R_i \times 2$, где R_i - число элементов входа (`net.inputs{i}.size`), и каждый элемент в столбце 1 меньше чем элемент, следующий за ним в столбце 2.

Каждая j-ая строка определяет минимальное и максимальное значения j-го входного элемента, в следующем порядке:

`net.inputs{i}(j,:)`

Использование.

Некоторые функции инициализации используют диапазоны входных значений для вычисления соответствующих начальных значений матриц входных весов.

Сопутствующие эффекты.

Всякий раз, когда число строк в этом свойстве изменяется, размерности входов (`net.inputs{i}.size`) изменяются таким образом, чтобы избежать несоответствий. Количество весовых коэффициентов каждого ввода (`net.inputWeights{:,i}.size`) и размерности их весовых матриц (`net.IW{:,i}`) также изменяются.

- **size** - определяет число элементов в i-ом входе сети

`net.inputs{i}.size` - может принимать значение 0 или положительное целое число.

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, диапазоны входов (`net.inputs{i}.ranges`), входные веса (`net.inputWeights{:,i}.size`) и матрицы весов (`net.IW{:,i}`) изменяют размерности чтобы избежать несоответствий.

- **userdata** - предоставляет место для пользователей, чтобы ввести дополнительную информацию об i-м входе сети.

`net.inputs{i}.userdata`

Одно поле предопределено. Оно содержит сообщение ко всем пользователям Neural Network Toolbox:

`net.inputs{i}.userdata.note`

MATLAB & Toolboxes

Ю.П.Маслобоев "Свойства подбъектов нейронной сети" Слои

Эти свойства определяют структуру каждого i-го слоя сети - `net.layers{i}`

- **dimensions** - определяет физические размеры i-го слоя нейронов. Возможность упорядочивать нейроны слоя многомерным способом важна для самоорганизующихся карт.

`net.layers{i}.dimensions` - может быть присвоено значение вектора-строки из нулей или положительных целых элементов. При этом произведение всех элементов равно числу нейронов в слое (`net.layers{i}.size`).

Использование.

Размерности слоя используются, чтобы вычислять позиции нейронов в слое (`net.layers{i}.positions`) с помощью функция топологии слоя (`net.layers{i}.topologyFcn`).

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, размеры слоев (`net.layers{i}.size`) изменяются, чтобы избежать несоответствий. Позиции нейронов в слоях (`net.layers{i}.positions`) и расстояния между нейронами (`net.layers{i}.distances`) также модифицируются.

- **distanceFcn** - определяет функцию, используемую для вычисления расстояния между нейронами в i-ом слое (`net.layers{i}.distances`) по их позициям в слое (`net.layers{i}.positions`). Расстояния между нейронами используются самоорганизующимися картами.

`net.layers{i}.distanceFcn` - название любой функции расстояния, из набора функций тулбокса:

Функции расстояний

<code>boxdist</code>	Расстояние между двумя векторами положения.
<code>dist</code>	Весовая функция Евклидова расстояния
<code>linkdist</code>	Связанная функция расстояния
<code>mandist</code>	Весовая функция расстояния Манхэттена

Сопутствующие эффекты.

Всякий раз, когда это свойство изменяется, расстояние между нейронами слоя (`net.layers{i}.distances`) модифицируется.

- **distances** (только для чтения) - определяет расстояния между нейронами в i-ом слое. Эти расстояния используются самоорганизующимися картами.

`net.layers{i}.distances` - возвращает значение результата применения функции расстояния слоя (`net.layers{i}.distanceFcn`) к позициям нейронов слоя (`net.layers{i}.positions`)

- **initFcn** - это свойство определяет функцию инициализации, используемую для инициализации i-го слоя, в случае, если функция инициализации сети (`net.initFcn`) – `initlay`.

`net.layers{i}.initFcn` - название любой функции инициализации слоя из тулбокса.

Функции инициализации слоя

initnw	Функция инициализации Нгуен-Видроу (Nguyen-Widrow)
initwb	Функция инициализации по весам и смещениям

Если установлена функция инициализации сети `initlay`, то функция указанная этим свойством используется, чтобы инициализировать веса и смещения слоя в случае, когда вызывается функция `init`:

```
net=init(net)
```

- **netInputFcn** - это свойство определяет использование входной функции сети для вычисления i-го входа слоя сети задаваемого взвешенными входами и смещениями.

`net.layers{i}.netInputFcn` – название любой входной функции сети из тулбокса.

Входные функции сети

netprod	Функция произведения входов
netsum	Функция суммирования входов

Входная функция сети используется для моделирования сети, когда вызывается функция `sim`:

```
[Y,Pf,Af]=sim(net,P,Pi,Ai)
```

- **positions** (только для чтения) - это свойство определяет позиции нейронов в i-ом слое. Эти позиции используются самоорганизующимися картами.

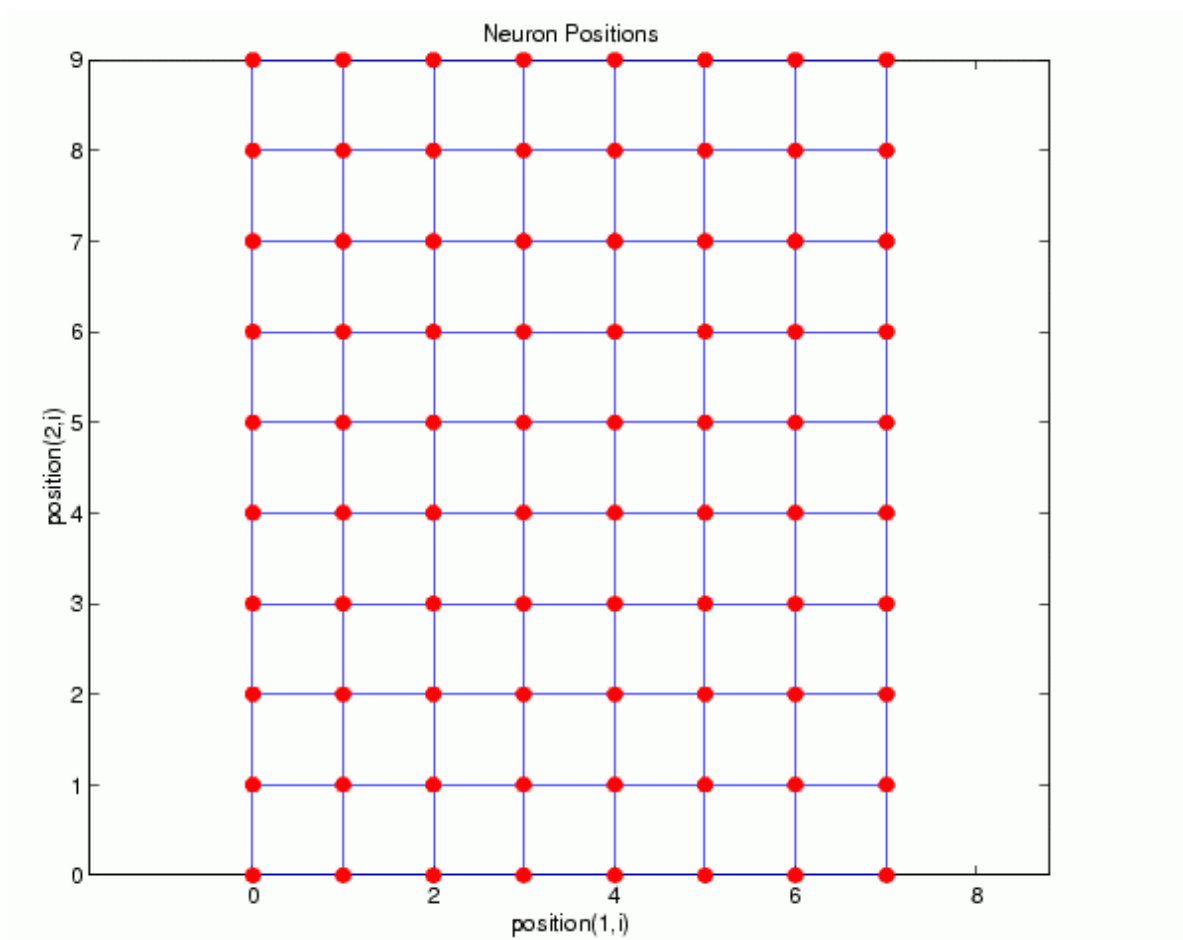
`net.layers{i}.positions` - возвращает значение результата применения функции топологии слоя (`net.layers{i}.topologyFcn`) к позициям размерности слоя (`net.layers{i}.dimensions`).

Графическое представление.

Используйте `plotsom`, чтобы нарисовать картину расположения нейронов слоя.

Например, если нейроны первого слоя сети размещены с размерностями (`net.layers{1}.dimensions`) = [4 5] и функция топологии (`net.layers{1}.topologyFcn`) = `hextop`, то позиции нейронов расположены так, как показано на рисунке, полученном в результате выполнения функции:

```
plotsom(net.layers{1}.positions)
```



- **size** - определяет число нейронов в i-ом слое.

`net.layers{i}.size` - можно присваивать значения 0 или положительное целое число.

Сопутствующие эффекты.

Всякий раз, когда это свойство изменяется, размеры входных весов к слою (`net.inputWeights{i,:}.size`) весовых коэффициенты к слою (`net.layerWeights{i,:}.size`), или от слоя (`net.inputWeights{i,:}.size`), а также смещений слоя (`net.biases{i}.size`) изменяются.

Размерности соответствующих весовых матриц (`net.IW{i,:}`, `net.LW{i,:}`, `net.LW{:,i}`) и смещений (`net.b {i}`) также изменяются.

Изменение этого свойства также изменяет размер выхода слоя (`net.outputs{i}.size`) и размерность вектора эталона (`net.targets{i}.size`) если они присутствуют.

Наконец, когда это свойство изменено, размерность нейронов слоя (`net.layers{i}.dimension`) устанавливается на то же самое значение. (Это приводит к одномерному расположению нейронов. Если требуется другое расположение, установите свойство размерности непосредственно, вместо того, чтобы использовать `size`.)

- **topologyFcn** – это свойство определяет функцию, используемую для вычисления позиций нейронов i-го слоя по размерностям слоя (`net.layers{i}.dimensions`).

`net.topologyFcn` - название любой функции топологии из тулбокса

Топологические функции

<code>gridtop</code>	Топологическая функция в виде сеточного слоя
<code>hextop</code>	Топологическая функция в виде гексагонального слоя
<code>randtop</code>	Топологическая функция в виде случайного слоя

Сопутствующие эффекты.

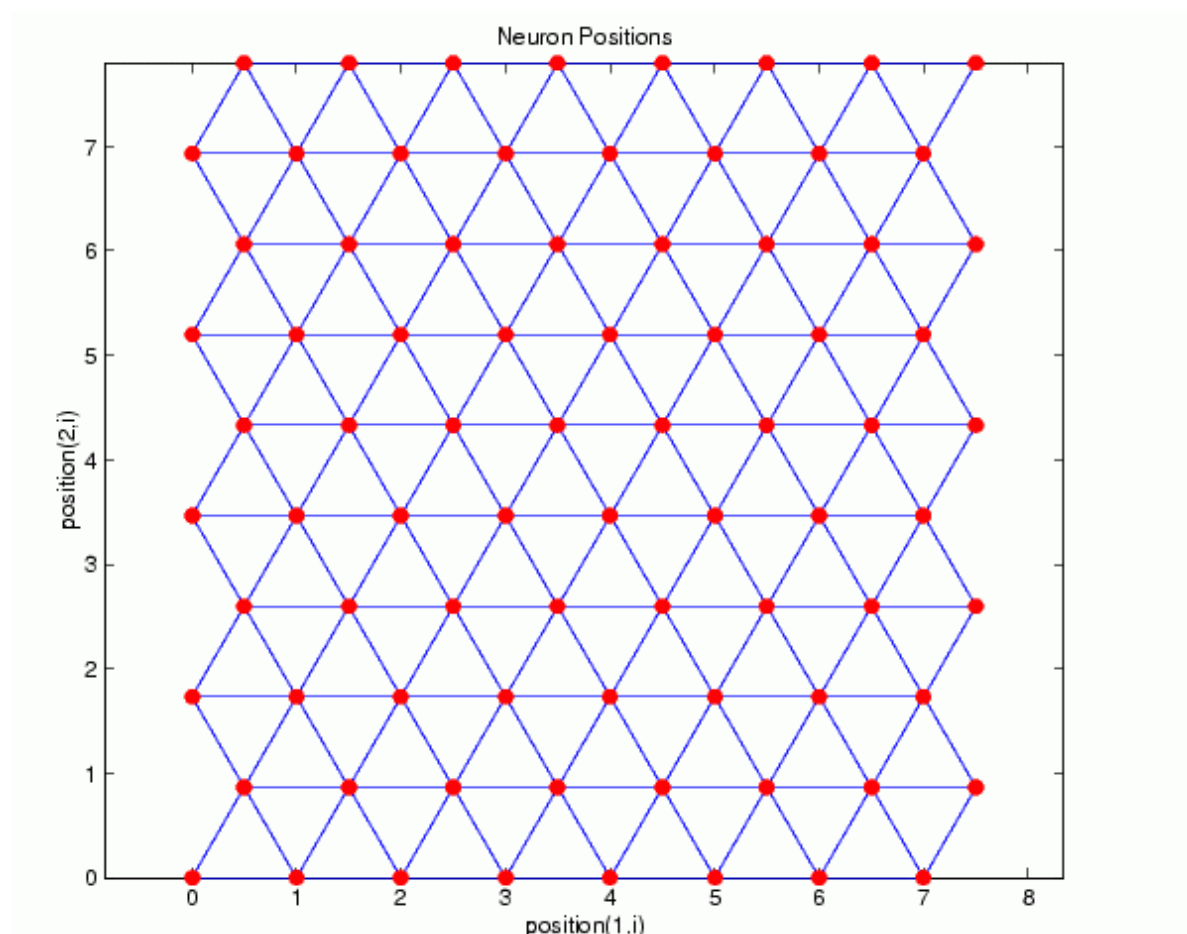
Всякий раз, когда это свойство изменяется, позиции нейронов слоя (`net.layers{i}.positions`) модифицируются.

Графическое представление.

Используйте `plotsom`, чтобы нарисовать картину расположения нейронов слоя.

Например, если нейроны первого слоя сети размещены с размерностями `(net.layers{1}.dimensions) = [8 10]` и функция топологии `(net.layers{1}.topologyFcn) = randtop`, то позиции нейронов расположены так, как показано на рисунке, полученном в результате выполнения функции:

```
plotsom(net.layers{1}.positions)
```



- **transferFcn** - эта функция определяет функцию активации, используемую для вычисления выхода *i*-го слоя по известному входу слоя.

`net.layers{i}.transferFcn` - качестве этой функции можно использовать любую функцию активации из тулбокса:

Функции активации

compet	Конкурирующая функция активации
hardlim	Ступенчатая функция активации
hardlims	Ступенчатая симметричная функция активации
logsig	Сигмоидная (логистическая) функция активации
poslin	Положительная линейная функция активации
purelin	Линейная функция активации
radbas	Радиальная базисная функция активации
satlin	Насыщающаяся функция активации
satlins	Симметричная насыщающаяся линейная функция активации
softmax	Функция активации, уменьшающая диапазон входных значений
tansig	Функция активации гиперболический тангенс
tribas	Треугольная функция активации

Функция активации используется, чтобы моделировать сеть когда вызывается функция `sim`:

```
[Y,Pf,Af] = sim(net,P,Pi,Ai)
```

- **userdata** - это свойство предоставляет место для пользователей, чтобы ввести дополнительную информацию об *i*-м слое сети.

```
net.layers{i}.userdata
```

Одно поле предусмотрено. Оно содержит сообщение ко всем пользователям Neural Network Toolbox.

```
net.layers{i}.userdata.note
```

Ю.П.Маслобоев "Свойства подбъектов нейронной сети" Выходы

- **size** (только для чтения) - возвращает число элементов в i-м выходе слоя

`net.outputs{i}.size` - возвращает размер i-го слоя (`net.layers{i}.size`)

- **userdata** - это свойство предоставляет место для пользователей, чтобы ввести дополнительную информацию об i-м слое сети.

`net.outputs{i}.userdata`

Одно поле предусмотрено. Оно содержит сообщение ко всем пользователям Neural Network Toolbox.

`net.outputs{i}.userdata.note`

Ю.П.Маслобоев "Свойства подбъектов нейронной сети" Эталоны

- **size** (только для чтения) - возвращает число элементов в эталоне i-го слоя

`net.targets{i}.size` - возвращает размер i-го слоя (`net.layers{i}.size`)

- **userdata** - это свойство предоставляет место для пользователей, чтобы ввести дополнительную информацию об эталоне i-го слоя сети.

`net.targets{i}.userdata`

Одно поле предопределено. Оно содержит сообщение ко всем пользователям Neural Network Toolbox.

`net.targets{i}.userdata.note`



Ю.П.Маслобоев "Свойства подбъектов нейронной сети" Смещения

- initFcn** - это свойство определяет функцию, используемую для инициализации вектора смещений i -го слоя в случае, если используется функция инициализации сети `initlay` и функция инициализации слоя - `initwb`

`net.biases{i}.initFcn` – задается в виде названия функции инициализации смещений из настоящего тулбокса:

Функции инициализации смещений

<code>initcon</code>	“Сознательная” функция инициализации
<code>initzero</code>	Инициализация с установкой нулевых значений смещений и весов
<code>rands</code>	Инициализация с установкой симметричных случайных значений весов и смещений

Эта функция используется, чтобы вычислять начальное значение вектора смещений для i -го слоя (`net.b{i}`), при вызове `init` в том случае, когда в качестве функции инициализации сети (`net.initFcn`) выбрана `initlay` и в качестве функции инициализации i -го слоя (`net.layers{i}.initFcn`) выбрана `initwb`.

`net = init(net)`

- learn** – это свойство определяет, будет ли изменен i -й вектор смещения в процессе обучения и адаптации.

`net.biases{i}.learn` – может быть установлен в 0 или 1, разрешая или запрещая модификацию смещений при вызове функций `adapt` или `train`:

`[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)`

`[net,tr] = train(NET,P,T,Pi,Ai)`

- learnFcn** - это свойство определяет функцию, используемую для модификации вектора весов i -го слоя в процессе тренировки, в случае, если в качестве функции тренировки сети выбраны `trainb`, `trainc`, или `trainr` а также в процессе адаптации, если в качестве функции адаптации сети выбрана `trains`.

`net.biases{i}.learnFcn` – название функции обучения из настоящего тулбокса.

Функции обучения

<code>learncon</code>	Обучающая функция смещений
<code>learnkd</code>	Обучающая функция градиентного спуска
<code>learnkdgm</code>	Обучающая функция Хэбба
<code>learnp</code>	Обучающая функция смещений и весов перцептрона
<code>learnpn</code>	Обучающая функция нормализованных смещений и весов перцептрона
<code>learnwh</code>	Правило обучения Уидроу-Хоффа (Widrow-Hoff)

Функция обучения модифицирует i -й вектор смещения (`net.b {i}`) при вызове `train`, в случае, если в качестве функции тренировки (`net.trainFcn`) выбраны `trainb`, `trainc`, или `trainr`, а также при вызове `adapt`, если в качестве функции адаптации (`net.adaptFcn`) выбрана `trains`.

```
[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)
```

```
[net,tr] = train(NET,P,T,Pi,Ai)
```

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, параметры обучения смещений (`net.biases{i}.learnParam`) устанавливаются таким образом, чтобы содержать поля и значения по умолчанию новой функции.

- **learnParam** - это свойство определяет параметры значения обучения для текущей функции обучения смещений *i*-го слоя.

```
net.biases{i}.learnParam
```

Поля этого свойства зависят от текущей функции обучения (`net.biases{i}.learnFcn`). Вызовите вышеупомянутую ссылку, чтобы ознакомиться с полями текущей функции обучения.

Вызовите справку для текущей функции обучения, чтобы получить описание каждого из полей:

```
help(net.biases{i}.learnFcn)
```

- **size** (только для чтения) - возвращает размер вектора смещения *i*-го слоя.

`net.biases{i}.size` – всегда возвращает размер *i*-го слоя (`net.layers{i}.size`).

- **userdata** - это свойство предоставляет место для пользователей, чтобы ввести дополнительную информацию об эталоне *i*-го слоя сети.

```
net.biases{i}.userdata
```

Одно поле предопределено. Оно содержит сообщение ко всем пользователям Neural Network Toolbox.

```
net.biases{i}.userdata.note
```

MATLAB & Toolboxes

Ю.П.Маслобоев "Свойства подбъектов нейронной сети" Входные веса

- **delays** - это свойство определяет дискретную задержку между j-м входом и его коэффициентом связи (весом) с i-м слоем.

`net.inputWeights{i,j}.delays` – может принимать значение в виде вектора-строки из нулей или целых положительных чисел.

Сопутствующие эффекты.

Всякий раз, когда это свойство изменено, размер весов (`net.inputWeights{i,j}.size`) и размерность матрицы весов (`net.IW{i,j}`) также изменяются.

- **initFcn** – это свойство определяет функцию, используемую для инициализации матрицы весовых коэффициентов (весов) между j-м входом и i-м слоем, в случае когда в качестве функции инициализации сети используется `initlay` а в качестве функции инициализации i-го слоя используется `initwb`.

`net.inputWeights{i,j}.initFcn` - задается в виде названия функции инициализации веса из настоящего тулбокса.

Функции инициализации весов

<code>initzero</code>	Инициализация с установкой нулевых значений весов и смещений
<code>midpoint</code>	Инициализация с установкой средних значений весов
<code>randnc</code>	Инициализация с установкой нормализованных значений столбцов весовых матриц.
<code>randnr</code>	Инициализация с установкой нормализованных значений строк весовых матриц.
<code>rands</code>	Инициализация с установкой симметричных случайных значений весов и смещений.

Эта функция используется, для вычисления исходной матрицы для весовых коэффициентов (весов) между j-м входом и i-м слоем (`net.IW{i,j}`) когда вызвана функция `init` в том случае, если в качестве функции инициализации сети (`net.initFcn`) используется `initlay` а в качестве функции инициализации i-го слоя (`net.layers{i}.initFcn`) используется `initwb`.

`net = init(net)`

- **learn** - это свойство определяет будет ли матрица весов к i-му слою от j-го входа изменена в процессе обучения и адаптации.

`net.inputWeights{i,j}.learn` – устанавливается в 0 или 1, разрешая или запрещая модификацию весов в процессе обучения, если вызваны `adapt` или `train`.

`[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)`

`[net,tr] = train(NET,P,T,Pi,Ai)`

- **learnFcn** - это свойство определяет функцию, используемую для модификации матрицы весов к i-му слою от j-го входа в процессе тренировки, в случае, если в качестве функции тренировки сети выбраны `trainb`, `trainc`, или `trainr` а также в процессе адаптации, если в качестве функции адаптации сети выбрана `trains`.

`net.inputWeights{i,j}.learnFcn` - задается в виде названия функции обучения весов из настоящего тулбокса

Функции обучения весов

learngd	Обучающая функция градиентного спуска
learngdm	Обучающая функция градиентного спуска с учетом моментов
learnh	Обучающая функция Хэбба
learnhd	Обучающая функция Хэбба с учетом затухания
learnis	Обучающая функция instar
learnk	Обучающая функция Кохонена
learnlv1	Обучающая функция LVQ1
learnlv2	Обучающая функция LVQ2
learnos	Обучающая функция outstar
learnp	Обучающая функция смещений и весов перцептрона
learnpn	Обучающая функция нормализованных смещений и весов перцептрона
learnsom	Обучающая функция самоорганизующейся карты весов
learnwh	Обучающая функция Уидроу-Хоффа (Widrow-Hoff)

Функция обучения модифицирует весовую матрицу i -го слоя от j -го входа ($\text{net.IW}\{i,j\}$) при вызове `train`, в случае, если в качестве функции тренировки (`net.trainFcn`) выбраны `trainb`, `trainc`, или `trainr`, а также при вызове `adapt`, если в качестве функции адаптации (`net.adaptFcn`) выбрана `trains`.

`[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)`

`[net,tr] = train(NET,P,T,Pi,Ai)`

- **learnParam** - это свойство определяет параметры и значения обучения для текущей функции обучения веса i -го слоя от j -го входа.

`net.inputWeights{i,j}.learnParam` - поля этого свойства зависят от текущей функции обучения (`net.inputWeights{i,j}.learnFcn`). Обратитесь к вышеупомянутой ссылке чтобы увидеть поля текущей функции обучения.

Вызовите `help` на текущей функции обучения, чтобы получить описание каждого из полей:

`help(net.inputWeights{i,j}.learnFcn)`

- **size** (только для чтения) - это свойство определяет размерность матрицы весов i -го слоя от j -го входа сети.

`net.inputWeights{i,j}.size` - оно всегда устанавливается в виде двухэлементного вектора-строки, указывающего число строк и столбцов соответствующей матрицы весов (`net.IW{i,j}`). Первый элемент равен размеру i -го слоя (`net.layers{i}.size`).

Второй элемент равен произведению длины векторов задержек весов на размер j -го входа.

`length(net.inputWeights{i,j}.delays) * net.inputs{j}.size`

- **userdata** - это свойство предоставляет место для пользователей, чтобы ввести дополнительную информацию об эталоне i -го слоя сети.

`net.inputWeights{i}.userdata`

Одно поле предопределено. Оно содержит сообщение ко всем пользователям Neural Network Toolbox.

`net.inputWeights{i}.userdata.note`

- **weightFcn** - это свойство определяет функцию, используемую для применения к весам от входа к j -му слою.

`net.inputWeights{i,j}.weightFcn` - задается в виде названия функции весов из настоящего тулбокса

Весовые функции

dist	Евклидово расстояние
------	----------------------

dotprod	Весовая функция в виде скалярного произведения
mandist	Весовая функция – расстояние Манхеттена
negdist	Весовая функция – отрицательное расстояние
normprod	Нормированное скалярное произведение

Весовые функции используются, когда вызывается функция `sim` для моделирования сети.

$[Y, Pf, Af] = \text{sim}(\text{net}, P, Pi, Ai)$



Ю.П.Маслобоев "Свойства подбъектов нейронной сети" Весы слоев

- **delays** – это свойство определяет дискретную задержку между j-м слоем и его коэффициентом связи (весом) с i – м слоем.

`net.layerWeights{i,j}.delays` – может принимать значение в виде вектора-строки из нулей или целых положительных чисел.

- **initFcn** – это свойство определяет функцию, используемую для инициализации матрицы весовых коэффициентов (весов) между j-м слоем и i-м слоем, в случае когда в качестве функции инициализации сети используется `initlay` а в качестве функции инициализации i-го слоя используется `initwb`.

`net.layerWeights{i,j}.initFcn` – задается в виде названия функции инициализации веса из настоящего тулбокса.

Функции инициализации весов

<code>initzero</code>	Инициализация с установкой нулевых значений весов и смещений
<code>midpoint</code>	Инициализация с установкой средних значений весов
<code>randnc</code>	Инициализация с установкой нормализованных значений столбцов весовых матриц.
<code>randnr</code>	Инициализация с установкой нормализованных значений строк весовых матриц.
<code>rand</code>	Инициализация с установкой симметричных случайных значений весов и смещений.

Эта функция используется, для вычисления исходной матрицы для весовых коэффициентов (весов) между i-м слоем и j-м слоем (`net.LW{i,j}`) когда вызвана функция `init` в том случае, если в качестве функции инициализации сети (`net.initFcn`) используется `initlay` а в качестве функции инициализации i-го слоя (`net.layers{i}.initFcn`) используется `initwb`.

`net = init(net)`

- **learn** – это свойство определяет будет ли матрица весов к i-му слою

от j-го слоя изменена в процессе обучения и адаптации.

`net.layerWeights{i,j}.learn` – устанавливается в 1 или 0, разрешая или запрещая модификацию весов в процессе обучения, если вызваны `adapt` или `train`.

`[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)`

`[net,tr] = train(NET,P,T,Pi,Ai)`

- **learnFcn** – это свойство определяет функцию, используемую для модификации матрицы весов к i-му слою от j-го слоя в процессе тренировки, в случае, если в качестве функции тренировки сети выбраны `trainb`, `trainc`, или `trainr` а также в процессе адаптации, если в качестве функции адаптации сети выбрана `trains`.

`net.inputWeights{i,j}.learnFcn` – задается в виде названия функции обучения весов из настоящего тулбокса

Функции обучения весов

<code>learngd</code>	Обучающая функция градиентного спуска
<code>learngdm</code>	Обучающая функция градиентного спуска с учетом моментов

learnh	Обучающая функция Хэбба
learnhd	Обучающая функция Хэбба с учетом затухания
learnis	Обучающая функция instar
learnk	Обучающая функция Кохонена
learnlv1	Обучающая функция LVQ1
learnlv2	Обучающая функция LVQ2
learnos	Обучающая функция outstar
learnp	Обучающая функция смещений и весов перцептрона
learnpn	Обучающая функция нормализованных смещений и весов перцептрона
learnsom	Обучающая функция самоорганизующейся карты весов
learnwh	Обучающая функция Уидроу-Хоффа (Widrow-Hoff)

Функция обучения модифицирует весовую матрицу i-го слоя от j-го слоя (net.LW{i,j}) при вызове train, в случае, если в качестве функции тренировки (net.trainFcn) выбраны trainb, trainc, или trainr, а также при вызове adapt, если в качестве функции адаптации (net.adaptFcn) выбрана trains.

[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)

[net,tr] = train(NET,P,T,Pi,Ai)

- **learnParam** - это свойство определяет параметры и значения обучения для текущей функции обучения веса i-го слоя от j-го слоя.

net.layerWeights{i,j}.learnParam

Поля этого свойства зависят от текущей функции обучения

(net.layerWeights{i,j}.learnFcn).

Вызовите help на текущей функции обучения, чтобы получить описание каждого из полей.

help(net.layerWeights{i,j}.learnFcn)

- **size** (только для чтения) – это свойство определяет размерность матрицы весов i-го слоя от j-го слоя сети.

net.layerWeights{i,j}.size - устанавливается в виде двухэлементного вектора-строки, указывающего число строк и столбцов соответствующей матрицы весов (net.LW{i,j}). Первый элемент равен размеру i-го слоя (net.layers{i}.size).

Второй элемент равен произведению длины векторов задержек весов на размер j-го слоя.

length(net.layerWeights{i,j}.delays) * net.inputs{j}.size

- **userdata** - это свойство предоставляет место для пользователей, чтобы ввести дополнительную информацию об (i,j)-весах слоев.

net.layerWeights{i}.userdata

Одно поле предопределено. Оно содержит сообщение ко всем пользователям Neural Network Toolbox.

net.layerWeights{i}.userdata.note

- **weightFcn** – это свойство определяет функцию, используемую для применения к весам от i-го слоя к j-му слою.

net.layerWeights{i,j}.weightFcn - задается в виде названия функции весов из настоящего тулбокса

Весовые функции

dist	Евклидово расстояние
dotprod	Весовая функция в виде скалярного произведения

mandist	Весовая функция – расстояние Манхеттена
negdist	Весовая функция – отрицательное расстояние
normprod	Нормированное скалярное произведение

Весовые функции используются, когда вызывается `sim` для моделирования сети:


```
[Y,Pf,Af] = sim(net,P,Pi,Ai)
```

Во первых по этому много информации на форуме и в интернете, во вторых вопрос слишком общий на него можно ответить НС только и умеют что аппроксимировать ф-ии.

Ну а если по делу:

1) Берете в правую руку MatLAB

2) левой набираете следующий код

 **Без подсветки**

```
1: X=0:0.1:10;
2: Y=sin(X);
3: net = newff([0 10],[5 1],{'tansig' 'purelin'});
4: net=train(net,X,Y);
5: X2=1:0.15:8;
6: Y2=sim(net,X2);
7: plot(Y2);
```

Это двуслойная нейросеть прямого распространения с сигмоидальной функцией скрытого слоя которая аппроксимирует функцию $\sin(x)$

для более сложных операций вам нужно почитать литературу по НС типа Калана...