Материал лекций по курсу «Нейросетевые модели и нейрокомпьютеры».

Лекция №1. «Задачи, решаемые нейросетевыми моделями. Основа нейросетевых моделей – нейрон. Структура, задачи, функционирование и типология нейронов.»

| Задачи | |
|---|---|
| Классификация<br>* образов<br>* последовательностей | |
| Распознавание (образов) | |
| Обобщение (образов) | |
| Прогнозирование (последовательностей) | |
| Преобразование (восстановление) | |
| Выявление основных характеристик (синтез решения) | |

Нейрон.

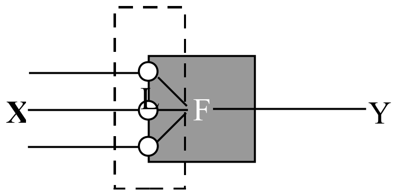
С математической точки зрения нейрон представляет собой композицию функций, задающих зависимость выходного сигнала нейрона Y от вектора входных сигналов **X**:

$$Y=F(L(\mathbf{X}))$$

$$\mathbf{X} = (x_1, x_2, x_3, x_4, \ldots x_n)$$

L(**X**) – это функция преобразования вектора входных сигналов на синапсах нейрона, F( ) – функция активации нейрона.

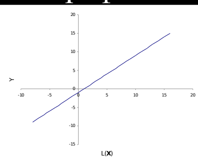
Структурно нейрон можно изобразить следующим образом: входные сигналы – компоненты вектора **X** – движутся по дендритам и поступают на синапсы, которые изображены в виде (трёх) кружочков, там над сигналами осуществляется преобразование L – результат подаётся на вход функции F, которая формирует на аксоне нейрона выходной сигнал Y.

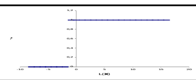
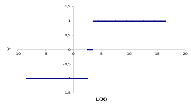
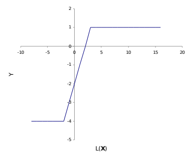
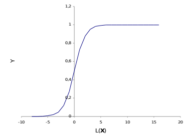
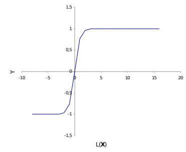
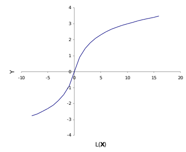
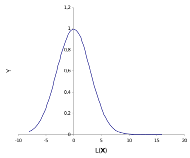



Типология входных сигналов $x_i$ – компонентов вектора **X**:

| структура сигнала | |
|---|---|
| * однопараметрический (скаляр), | |
| | мощность шкалы измерения сигнала |
| | * дискретная шкала (дискретный сигнал), |
| | * непрерывная шкала (аналоговый сигнал); |
| * многопараметрический (вектор), | |
| | физическая интерпретация характеристик многопараметрического сигнала |
| | * амплитуда, |
| | * частота (спектр), |
| | * фаза, |
| | * длительность. |

Типология возможных функций активации нейронов:

| функция | Формула | график |
|---|---|---|
| линейная<br>(вырожденная функция активации, используется при решении задач прогнозирования) | $F(\mathbf{X}) = k_* L(\mathbf{X})-T$ |  |

| | | |
|---|---|---|
| пороговая | $F(\mathbf{X}) = k_1$, при $L(\mathbf{X}) \geq T$<br>$= k_2$, при $L(\mathbf{X}) < T$ |  |
| функция знака | $F(\mathbf{X}) = k * sign(L(\mathbf{X})-T)$ |  |
| ограниченная линейная | $= k_2$, при $L(\mathbf{X}) > T_2$<br>$F(\mathbf{X}) = (k_2-k_1) * ( L(\mathbf{X})- T_1) /$<br>$(T_2-T_1)+ k_1$,<br>при $T_1 \leq L(\mathbf{X}) \leq T_2$<br>$= k_1$, при $L(\mathbf{X}) < T_1$ |  |
| сигмоидная | $F(\mathbf{X}) = 1/(1+\exp(k * L(\mathbf{X})-T))$ |  |
| гиперболический тангенс | $F(\mathbf{X}) = th(k * L(\mathbf{X})-T)$ |  |
| логарифмическая | $F(\mathbf{X}) = \ln(L(\mathbf{X})+\sqrt{(L(\mathbf{X})^2+1)})$ |  |
| радиально-базисная | $F(\mathbf{X}) = \exp(-((L(\mathbf{X})-T)/k)^2/2)$ |  |

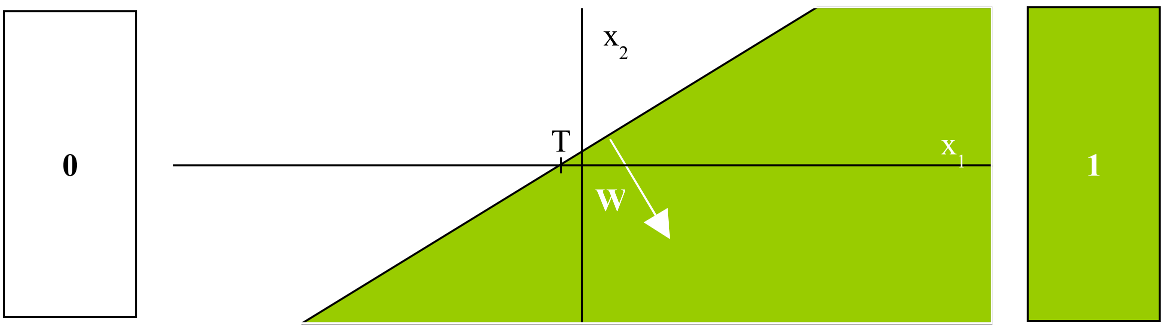
Соответственно тип нейрона определяется типом обрабатываемого сигнала и видом функции F и L.

Геометрическая интерпретация задачи нейрона.

Рассмотрим нейрон вида:

$$L(\mathbf{X}) = (\mathbf{X},\mathbf{W})\ ;$$

$$F(\mathbf{X}) = \begin{cases} 1, \text{ при } L(\mathbf{X}) \geq T; \\ 0, \text{ при } L(\mathbf{X}) < T; \end{cases}$$

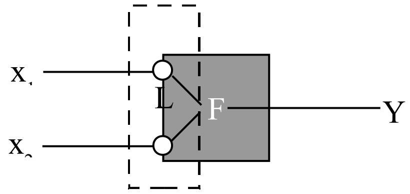



Такой нейрон разбивает пространство входных сигналов на две линейно разделённые области точек этого пространства: для точек из одной области нейрон будет давать на выходе сигнал **1**, а для точек из другой области – **0**.

Процесс обучения рассматриваемого нейрона сводится к изменению его внутренних параметров – компонентов вектора **W** и порога T. Приведённая ниже формула была предложена одной из первых для обучения нейрона и известна как правило Хебба:

$$w_i(t+1) = w_i(t)+\alpha * x_i * Y,\ i = 1..n$$

здесь $w_i(t+1)$ и $w_i(t)$ – компоненты вектора **W** в момент времени t+1 и t соответственно, α – шаг обучения ($0 < \alpha \leq 1$). Чтобы использовать правило Хебба для настройки порога поступают следующим образом: переходят от рассматриваемого нами нейрона (см. рис. 1) к нейрону, изображённому на рис. 2.



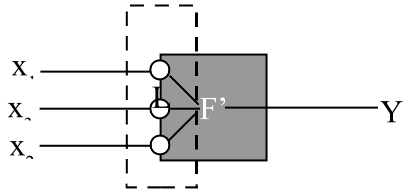




Рис. 1 Рис. 2

где

$$F'(\mathbf{X}) = \begin{cases} 1, \text{ при } L(\mathbf{X}) \geq 0; \\ 0, \text{ при } L(\mathbf{X}) < 0; \end{cases}$$

$x_3 = -1$, а $w_3 = T$. Таким образом:

$$T(t+1) = T(t)-\alpha * Y$$

здесь стоит отметить, что эту формулу целесообразно применять в случае, когда функция активации нейрона является биполярно-пороговой: $Y \in \{-1; 1\}$, или осуществляется нормализация вектора **W**, включая порог.

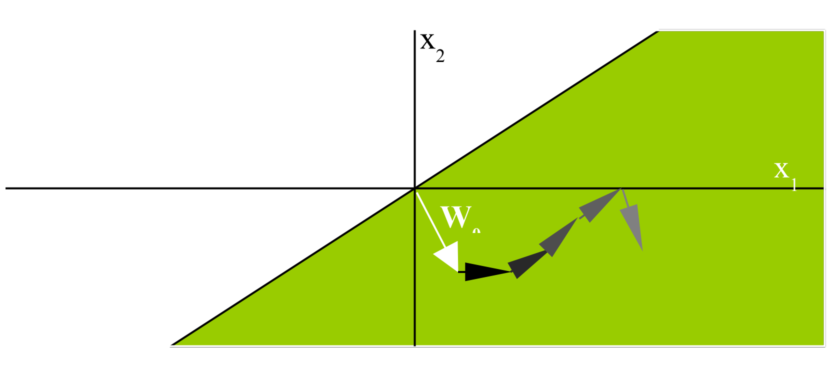
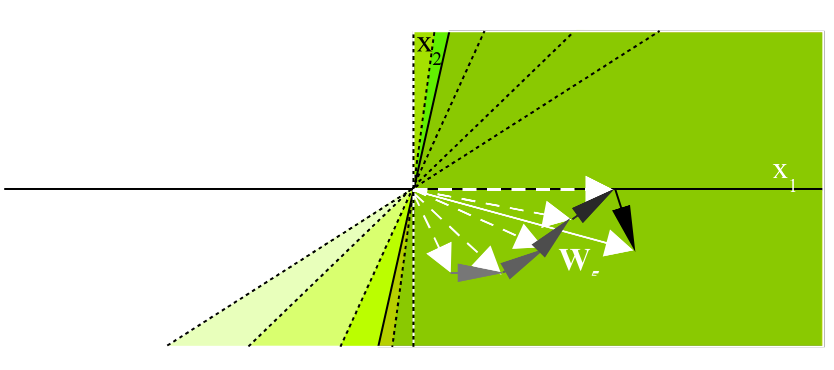
Согласно правилу Хебба процесс обучения зависит от входных и внутренних значений характеристик нейрона. Такой механизм обучения называется обучением без учителя. Если же при обучении используется эталонное (ожидаемое) значение выходного сигнала нейрона, то такой механизм обучения называется обучением с учителем. Приведём формулу для обучения с учителем известную как дельта-правило:

$$w_i(t+1) = w_i(t)+\alpha * x_i * (P-Y),\ i = 1..n$$

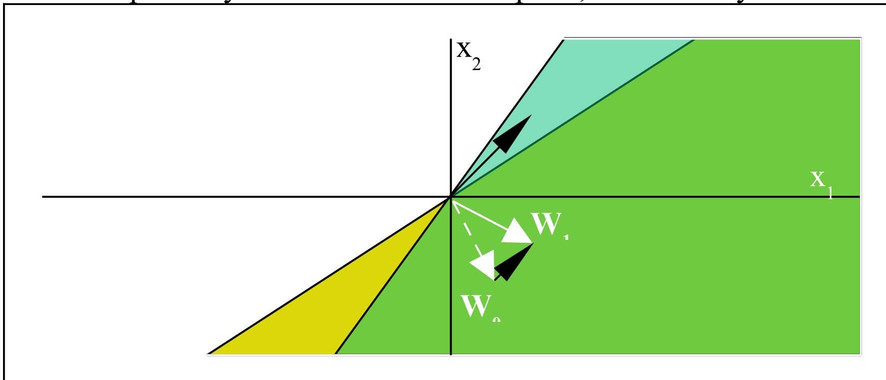
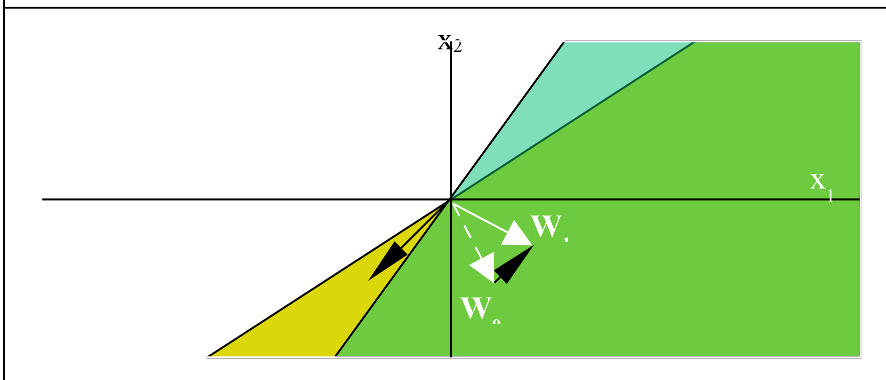
P – эталонное значение выходного сигнала нейрона.

Рассмотрим геометрическую интерпретацию правила Хебба и дельта-правила.

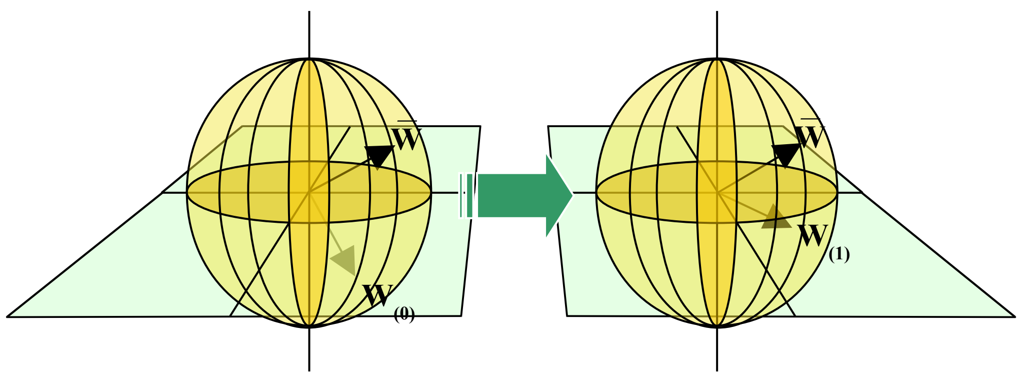
Пусть есть нейрон с двумя входами и пороговой функцией активации ($k_1$=1; $k_2$=0; T=0), каждый сигнал, поступающий на вход такого нейрона, умножается на вес – компонент вектора **W**. Рассмотрим процесс изменения согласно правилу Хебба вектора **W**, обозначив вектора **W** и **X** в момент времени **i** как $\mathbf{W_{(i)}}$ и $\mathbf{X_{(i)}}$ соответственно.

| | |
|---|---|
|   | $\mathbf{W}_{(0)}$= (2,-3); $\mathbf{X}_{(0)}$= (2,0)<br>$\mathbf{W}_{(1)}$= (4,-3); $\mathbf{X}_{(1)}$= (2,1)<br>$\mathbf{W}_{(2)}$= (6,-2); $\mathbf{X}_{(2)}$= (1,1) |
|   | $\mathbf{W}_{(3)}$= (7,-1); $\mathbf{X}_{(3)}$= (2,1)<br>$\mathbf{W}_{(4)}$= (9,0); $\mathbf{X}_{(4)}$= (1,-2)<br>$\mathbf{W}_{(5)}$= (10,-2) |

Рассмотрим обучение такого же нейрона, но только уже согласно дельта-правилу.

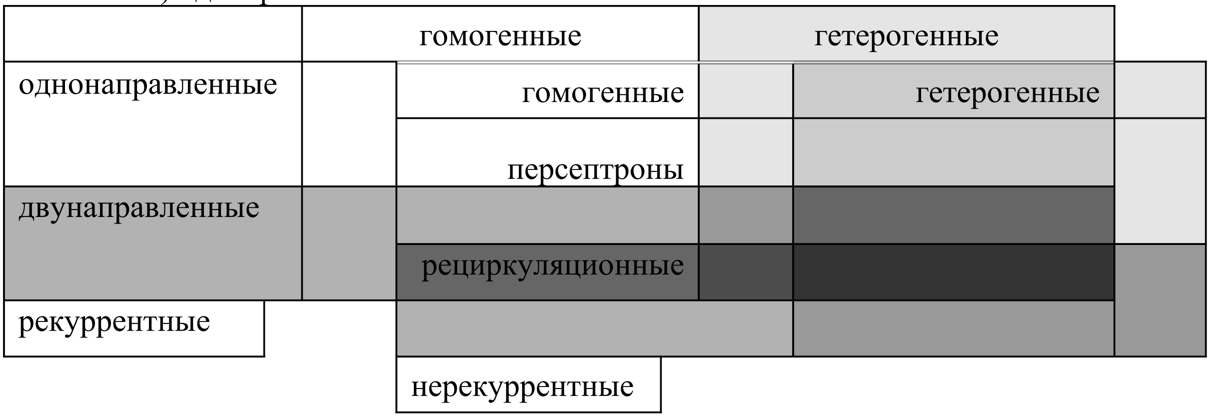
| | |
|---|---|
|   | $\mathbf{W}_{(0)}$= (2,-3);<br>$\mathbf{X}_{(0)}$= (3,3);<br>Y=0;<br>S=1<br>$\mathbf{W}_{(1)}$= (3,-2) |
|   | $\mathbf{W}_{(0)}$= (2,-3);<br>$\mathbf{X}_{(0)}$= (-3,-3);<br>Y=1;<br>S=0<br>$\mathbf{W}_{(1)}$= (3,-2) |

В случае нормализации векторов **W** и **X** геометрическая интерпретация процесса обучения согласно дельта-правилу, заключается в подстройке направления вектора **W** в направлении некоторого эталонного вектора $\bar{\mathbf{W}}$.



Лекция №2. «Слой. Типология нейросетевых моделей. Простейшие сети, персептроны. Обучение нейросетей.»

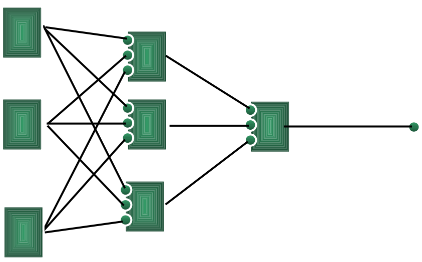
В предыдущем разделе был рассмотрен нейрон – основа нейросетевых моделей. Каждая нейронная сеть составляется из некоторого количества нейронов. Нейроны в сети организуется в слои. Слоем нейронной сети называется множество нейронов, которые могут обрабатывать один и тот же сигнал (приходящий из одного источника) одновременно.



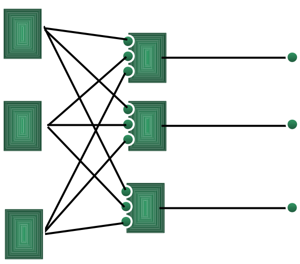

Рассмотрим типологию нейронных сетей. Сети делятся на гомогенные (содержащие однотипные нейроны) и гетерогенные. Сети только с прямым распространением информации (однонаправленные) и сети с обратным распространением информации (двунаправленные). Сети с наличием обратных связей (реккурентные) и с их отсутствием. Среди двунаправленных сетей выделяют класс рециркуляционных, а среди сетей с обратными связями – класс релаксационных сетей. Такие сети обрабатывают информацию до тех пор, пока сеть не достигнет стабильного состояния, состояния релаксации. Примером релаксационной сети является сеть Хопфилда. По количеству слоёв сети делят на однослойные, двуслойные сети и т.д. Кроме того сети могут быть классифицированы по механизму их обучения либо его отсутствию (сети с фиксированными связями). Рассмотрим некоторые классы нейронных сетей. Нейроны в сетях делят на рецепторы, эффекторы и нейроны скрытых слоёв.

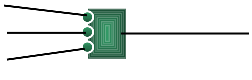
Начнём с нейросетевых моделей, исследованных Ф. Розенблаттом в качестве значительно упрощённых моделей фрагментов мозга и получивших название персептронов. По Розенблатту персептроны – это сети составленные из S-, A- и R-элементов. Сенсорный (S-элемент) способен выдавать сигнал 1 либо 0. Ассоциативный элемент (A-элемент) – обычный нейрон ($L(\mathbf{X}) = (\mathbf{X},\mathbf{W})$) с бинарно-пороговой функцией активации. Реагирующий элемент (R-элемент) – обычный нейрон с биполярно-пороговой функцией активации. Фрэнк Розенблатт выделял персептроны с последовательными связями, персептроны с перекрёстными связями и персептроны с обратными связями.

Рассмотрим трёхслойный по Розенблатту персептрон с последовательными связями. Он имеет следующую структуру:



Элементарным альфа-персептроном называется разновидность трёхслойного персептрона, у которого не меняются веса связей между S- и A-элементами. Класс задач, решаемый таким персептроном,





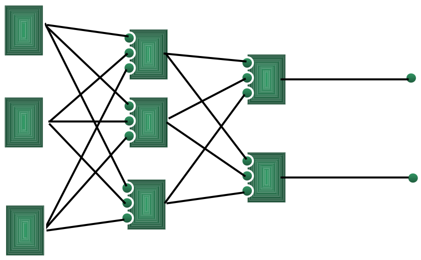
эквивалентен классу задач, решаемых каждым нейроном однослойной сети или просто нейроном, поэтому такую модель можно назвать сетью весьма условно. Такую сеть можно обучать как с помощью правила Хебба, так и с помощью дельта-правила.

Элементарный альфа-персептрон не способен решить задачу «исключающего или». Благодаря трудам Розенблатта доказано, что для случая, когда обучение происходит согласно дельта-правилу, элементарный альфа-персептрон успешно обучается за конечное количество шагов обучения на заданной выборке, когда экземпляры из выборки предъявляются в случайном порядке, но через гарантированно конечный интервал времени, кроме того такой персептрон может обучаться и в случае случайного знака ошибки (подкрепления).

Гамма-персептроном согласно Розенблатту называют персептрон, в котором сумма весов одного нейрона постоянна. Розенблатт рассматривал модели, в которых веса связей между сенсорным и ассоциативным слоем подчиняются нормальному распределению (НРВ (нормальное распределение входов) и РРВ (раздельное (для тормозящих и возбуждающих) распределение входов)), исследовал способности к классификации и распознавания образов трёхслойными персептронами, обобщающие свойства персептронов с четырьмя слоями и персептронов с перекрёстными связями. Видел будущее в применении персептронов с обратными связями.

Рассмотрим нейросетевую модель, которая способна решать более сложную задачу. Например, задачу определения принадлежности точки заданному выпуклому многоугольнику.

Пусть есть двуслойная сеть с бинарно-пороговой функцией активации. Если веса и пороги трёх нейронов первого слоя имеют следующие значения:

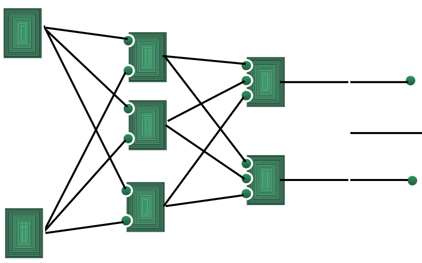


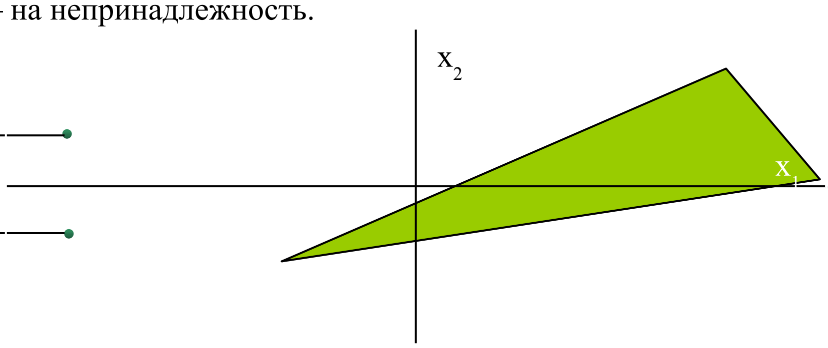
| | |
|---|---|
| $\mathbf{W}_1^{[1]} = (1,-2);$ | $T_1^{[1]} = 2$ |
| $\mathbf{W}_2^{[1]} = (-1,-1);$ | $T_2^{[1]} = -29$ |
| $\mathbf{W}_3^{[1]} = (-1.1,7.5);$ | $T_3^{[1]} = -27.6$ |

а веса нейронов второго слоя имеют значения:

| | |
|---|---|
| $\mathbf{W}_1^{[2]} = (1,1,1);$ | $T_1^{[2]} = 3$ |
| $\mathbf{W}_2^{[2]} = (-1,-1,-1);$ | $T_2^{[2]} = -2$ |

то получим сеть, первый нейрон выходного (второго) слоя которой будет реагировать на принадлежность точки треугольнику с вершинами (20, 9), (28,5, 0.5), (-9,-5), а второй нейрон – на непринадлежность.






Возьмём точку (12,0) внутри треугольника, получим $\mathbf{X} = (12,0)$:

| | |
|---|---|
| $L(\mathbf{X})_1^{[1]} = 12*1+0*(-2)+(-1)*2 = 14;$ | $Y_1^{[1]} = 1$ |
| $L(\mathbf{X})_2^{[1]} = 12*(-1)+0*(-1)+(-1)*(-29) = 17;$ | $Y_2^{[1]} = 1$ |
| $L(\mathbf{X})_3^{[1]} = 12*(-1.1)+0+(-1)*(-27.6) = 14.4;$ | $Y_3^{[1]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 1*1+1*1+1*1+(-1)*3 = 0;$ | $Y_1^{[2]} = 1$ |
| $L(\mathbf{X})_2^{[2]} = 1*(-1)+1*(-1)+1*(-1)+2 = -1;$ | $Y_2^{[2]} = 0$ |

Возьмём точку (0,0) вне треугольника, получим **X** = (0,2):

| | |
|---|---|
| $L(\mathbf{X})_1^{[1]} = 0*1+0*(-2)+(-1)*2 = -2;$ | $Y_1^{[1]} = 0$ |
| $L(\mathbf{X})_2^{[1]} = 0*(-1)+0*(-1)+(-1)*(-29) = 29;$ | $Y_2^{[1]} = 1$ |
| $L(\mathbf{X})_3^{[1]} = 0*(-1.1)+0+(-1)*(-27.6) = 27.6;$ | $Y_3^{[1]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 0*1+1*1+1*1+(-1)*3 = -1;$ | $Y_1^{[2]} = 0$ |
| $L(\mathbf{X})_2^{[2]} = 0*(-1)+1*(-1)+1*(-1)+2 = 0;$ | $Y_2^{[2]} = 1$ |

Нейроны выходного слоя такой сети могут решать задачу «исключающее или», но не могут определить принадлежность точки невыпуклой замкнутой или несвязной геометрической фигуре.

Сеть, решающую задачу «исключающее или» можно задать следующим образом: веса и пороги двух нейронов первого слоя имеют следующие значения:

| | |
|---|---|
| $\mathbf{W_1^{[1]}} = (-1,-1);$ | $T_1^{[1]} = -0.5$ |
| $\mathbf{W_2^{[1]}} = (1,1);$ | $T_2^{[1]} = 1.5$ |

а веса нейрона второго слоя имеют значения:

| | |
|---|---|
| $\mathbf{W_1^{[2]}} = (-1,-1);$ | $T_1^{[2]} = 0$ |

Действительно:

| | |
|---|---|
| $L(\mathbf{X})_1^{[1]} = 0*(-1)+0*(-1)+(-1)*(-0.5) = 0.5;$ | $Y_1^{[1]} = 1$ |
| $L(\mathbf{X})_2^{[1]} = 0*1+0*1+(-1)*1.5 = -1.5;$ | $Y_2^{[1]} = 0$ |
| $L(\mathbf{X})_1^{[2]} = 1*(-1)+0*(-1)+(-1)*0 = -1;$ | $Y_1^{[2]} = 0$ |
| $L(\mathbf{X})_1^{[1]} = 0*(-1)+1*(-1)+(-1)*(-0.5) = -0.5;$ | $Y_1^{[1]} = 0$ |
| $L(\mathbf{X})_2^{[1]} = 0*1+1*1+(-1)*1.5 = -0.5;$ | $Y_2^{[1]} = 0$ |
| $L(\mathbf{X})_1^{[2]} = 0*(-1)+0*(-1)+(-1)*0 = 0;$ | $Y_1^{[2]} = 1$ |
| $L(\mathbf{X})_1^{[1]} = 1*(-1)+0*(-1)+(-1)*(-0.5) = -0.5;$ | $Y_1^{[1]} = 0$ |
| $L(\mathbf{X})_2^{[1]} = 1*1+0*1+(-1)*1.5 = -0.5;$ | $Y_2^{[1]} = 0$ |
| $L(\mathbf{X})_1^{[2]} = 0*(-1)+0*(-1)+(-1)*0 = 0;$ | $Y_1^{[2]} = 1$ |
| $L(\mathbf{X})_1^{[1]} = 1*(-1)+1*(-1)+(-1)*(-0.5) = -1.5;$ | $Y_1^{[1]} = 0$ |
| $L(\mathbf{X})_2^{[1]} = 1*1+1*1+(-1)*1.5 = 0.5;$ | $Y_2^{[1]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 0*(-1)+1*(-1)+(-1)*0 = -1;$ | $Y_1^{[2]} = 0$ |

Рассмотрим сеть, выходные нейроны которой способны распознавать принадлежность точки произвольной геометрической фигуре, граница которой представляет произвольную ломанную линию.

| | |
|---|---|
| $\mathbf{W_1^{[1]}} = (1,0);$ | $T_1^{[1]} = -1$ |
| $\mathbf{W_2^{[1]}} = (0,-1);$ | $T_2^{[1]} = -1$ |
| $\mathbf{W_3^{[1]}} = (-1,0);$ | $T_3^{[1]} = -1$ |
| $\mathbf{W_4^{[1]}} = (0,-1);$ | $T_4^{[1]} = 1$ |
| $\mathbf{W_5^{[1]}} = (-1,0);$ | $T_3^{[1]} = -2$ |
| $\mathbf{W_6^{[1]}} = (0,1);$ | $T_4^{[1]} = -2$ |
| $\mathbf{W_1^{[2]}} = (1,1,1,-1,0,0);$ | $T_1^{[2]} = 3$ |
| $\mathbf{W_2^{[2]}} = (0,0,-1,1,1,1);$ | $T_2^{[2]} = 3$ |
| $\mathbf{W_1^{[3]}} = (1,1);$ | $T_1^{[3]} = 1$ |

Проверим **X** = (0, 0):

| | |
|---|---|
| $L(\mathbf{X})_1^{[1]} = 0*1+0*0+(-1)*(-1) = 1;$ | $Y_1^{[1]} = 1$ |
| $L(\mathbf{X})_2^{[1]} = 0*0+0*(-1)+(-1)*(-1) = 1;$ | $Y_2^{[1]} = 1$ |
| $L(\mathbf{X})_3^{[1]} = 0*(-1)+0*0+(-1)*(-1) = 1;$ | $Y_3^{[1]} = 1$ |
| $L(\mathbf{X})_4^{[1]} = 0*0+0*(-1)+(-1)*1 = -1;$ | $Y_4^{[1]} = 0$ |
| $L(\mathbf{X})_5^{[1]} = 0*(-1)+0*0+(-1)*(-2) = 2;$ | $Y_5^{[1]} = 1$ |
| $L(\mathbf{X})_6^{[1]} = 0*0+0*1+(-1)*(-2) = 2;$ | $Y_6^{[1]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 1+1+1+0+0+0-3 = 0;$ | $Y_1^{[2]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 0+0-1+0+1+1-3 = -2;$ | $Y_2^{[2]} = 0$ |

| | |
|---|---|
| $L(\mathbf{X})_1^{[3]} = 1*1+0*1+(-1)*1 = 0;$ | $Y_1^{[3]} = 1$ |

$\mathbf{X} = (1.5, -1.5)$

| | |
|---|---|
| $L(\mathbf{X})_1^{[1]} = 1.5+(-1.5)*0+(-1)*(-1) = 2.5;$ | $Y_1^{[1]} = 1$ |
| $L(\mathbf{X})_2^{[1]} = 1.5*0+1.5+(-1)*(-1) = 2.5;$ | $Y_2^{[1]} = 1$ |
| $L(\mathbf{X})_3^{[1]} = -1.5+(-1.5)*0+(-1)*(-1) = -0.5;$ | $Y_3^{[1]} = 0$ |
| $L(\mathbf{X})_4^{[1]} = 1.5*0+1.5+(-1)*1 = 0.5;$ | $Y_4^{[1]} = 1$ |
| $L(\mathbf{X})_5^{[1]} = -1.5+(-1.5)*0+(-1)*(-2) = 0.5;$ | $Y_5^{[1]} = 1$ |
| $L(\mathbf{X})_6^{[1]} = 1.5*0-1.5+(-1)*(-2) = 0.5;$ | $Y_6^{[1]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 1+1+0-1+0+0-3 = -2;$ | $Y_1^{[2]} = 0$ |
| $L(\mathbf{X})_1^{[2]} = 0+0+0+1+1+1-3 = 0;$ | $Y_2^{[2]} = 1$ |
| $L(\mathbf{X})_1^{[3]} = 0*1+1*1+(-1)*1 = 0;$ | $Y_1^{[3]} = 1$ |

$\mathbf{X} = (1.5, 0)$

| | |
|---|---|
| $L(\mathbf{X})_1^{[1]} = 1.5*1+0*0+(-1)*(-1) = 2.5;$ | $Y_1^{[1]} = 1$ |
| $L(\mathbf{X})_2^{[1]} = 1.5*0+0*(-1)+(-1)*(-1) = 1;$ | $Y_2^{[1]} = 1$ |
| $L(\mathbf{X})_3^{[1]} = 1.5*(-1)+0*0+(-1)*(-1) = -0.5;$ | $Y_3^{[1]} = 0$ |
| $L(\mathbf{X})_4^{[1]} = 1.5*0+0*(-1)+(-1)*1 = -1;$ | $Y_4^{[1]} = 0$ |
| $L(\mathbf{X})_5^{[1]} = 1.5*(-1)+0*0+(-1)*(-2) = 0.5;$ | $Y_5^{[1]} = 1$ |
| $L(\mathbf{X})_6^{[1]} = 1.5*0+0*1+(-1)*(-2) = 2;$ | $Y_6^{[1]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 1+1+0+0+0+0-3 = -1;$ | $Y_1^{[2]} = 0$ |
| $L(\mathbf{X})_1^{[2]} = 0+0+0+0+1+1-3 = -1;$ | $Y_2^{[2]} = 0$ |
| $L(\mathbf{X})_1^{[3]} = 0*1+0*1+(-1)*1 = -1;$ | $Y_1^{[3]} = 0$ |

$\mathbf{X} = (0, -1.5)$

| | |
|---|---|
| $L(\mathbf{X})_1^{[1]} = 0*1+(-1.5)*0+(-1)*(-1) = 1;$ | $Y_1^{[1]} = 1$ |
| $L(\mathbf{X})_2^{[1]} = 0*0+(-1.5)*(-1)+(-1)*(-1) = 2.5;$ | $Y_2^{[1]} = 1$ |
| $L(\mathbf{X})_3^{[1]} = 0*(-1)+(-1.5)*0+(-1)*(-1) = 1;$ | $Y_3^{[1]} = 1$ |
| $L(\mathbf{X})_4^{[1]} = 0*0+(-1.5)*(-1)+(-1)*1 = 0.5;$ | $Y_4^{[1]} = 1$ |
| $L(\mathbf{X})_5^{[1]} = 0*(-1)+(-1.5)*0+(-1)*(-2) = 2;$ | $Y_5^{[1]} = 1$ |
| $L(\mathbf{X})_6^{[1]} = 0*0+(-1.5)*1+(-1)*(-2) = 0.5;$ | $Y_6^{[1]} = 1$ |
| $L(\mathbf{X})_1^{[2]} = 1+1+1-1+0+0-3 = -1;$ | $Y_1^{[2]} = 0$ |
| $L(\mathbf{X})_1^{[2]} = 0+0-1+1+1+1-3 = -1;$ | $Y_2^{[2]} = 0$ |
| $L(\mathbf{X})_1^{[3]} = 0*1+0*1+(-1)*1 = -1;$ | $Y_1^{[3]} = 0$ |

Следовательно для того, чтобы распознать принадлежность точки произвольному невыпуклому многоугольнику или более сложной фигуре, персептрону требуется три и более слоя.

Стоит заметить, что возможности нерекуррентной гомогенной нейросети с линейной функцией активации ничем не отличаются от возможностей однослойной сети с такой же функцией активации, таким образом: необходимым свойством нерекуррентных гомогенных многослойных сетей является нелинейность функции активации составляющих её нейронов. Задача обучения многослойных нейронных сетей при наличии эталонного значения (образа) сложнее задачи обучения нейрона.

Дана сеть – определить на принадлежность точки какой фигуре реагирует выходной нейрон сети.

Вариант 1.

| | |
|---|---|
| $\mathbf{W_1^{[1]}} = (1,2);$ | $T_1^{[1]} = -1$ |
| $\mathbf{W_2^{[1]}} = (4,-1);$ | $T_2^{[1]} = -1$ |
| $\mathbf{W_3^{[1]}} = (-1,-2);$ | $T_3^{[1]} = 1$ |
| $\mathbf{W_4^{[1]}} = (3,-3);$ | $T_4^{[1]} = 1$ |
| $\mathbf{W_5^{[1]}} = (-1,2);$ | $T_3^{[1]} = -1$ |
| $\mathbf{W_6^{[1]}} = (4,1);$ | $T_4^{[1]} = -1$ |

| | |
|---|---|
| $\mathbf{W}_1^{[2]} = (1,1,0,1,0,1);$ | $T_1^{[2]} = 2$ |
| $\mathbf{W}_2^{[2]} = (1,0,1,0,1,1);$ | $T_2^{[2]} = 2$ |
| $\mathbf{W}_1^{[3]} = (1,2);$ | $T_1^{[3]} = 1$ |
| $\mathbf{W}_2^{[3]} = (-2,1);$ | $T_2^{[3]} = 0$ |
| $\mathbf{W}_1^{[4]} = (1,-1);$ | $T_1^{[4]} = 0.5$ |

Вариант 2.

| | |
|---|---|
| $\mathbf{W}_1^{[1]} = (1,0);$ | $T_1^{[1]} = -1$ |
| $\mathbf{W}_2^{[1]} = (3,-4);$ | $T_2^{[1]} = 0$ |
| $\mathbf{W}_3^{[1]} = (0,-2);$ | $T_3^{[1]} = 1$ |
| $\mathbf{W}_1^{[2]} = (0,0,-1);$ | $T_4^{[1]} = 1$ |
| $\mathbf{W}_2^{[2]} = (0,-1,0);$ | $T_3^{[1]} = 0$ |
| $\mathbf{W}_3^{[2]} = (-1,0,0);$ | $T_4^{[1]} = 0$ |
| $\mathbf{W}_4^{[2]} = (1,0,-1);$ | $T_1^{[2]} = -1$ |
| $\mathbf{W}_5^{[2]} = (-1,0,1);$ | $T_2^{[2]} = -1$ |
| $\mathbf{W}_1^{[3]} = (1,-1,1,1,1);$ | $T_1^{[3]} = 1$ |

Рассмотрим алгоритм обучения многослойных сетей известный под названием алгоритма обратного распространения ошибки. Его необходимым условием выступает дифференцируемость используемых функций активации нейронов сети. Этот алгоритм использует градиентный метод наискорейшего спуска для минимизации среднеквадратической ошибки сети, вычисляемой по формуле.

$$E = ½ * \sum(P_i-Y_i^{[p]})^2$$

$$w_{ij}^{[q]}(t+1) = w_{ij}^{[q]}(t)-\alpha * dE/dw_{ij}^{[q]},\ i = 1..n_q,\ q = 1..p$$

$$dE/dw_{ij}^{[q]} = (dE/dy_j^{[q]})*(dy_j^{[q]}/dS_j^{[q]})*(dS_j^{[q]}/dw_{ij}^{[q]}),\ q = 1..p$$

$$dE/dy_i^{[q]} = \sum(dE/dy_j^{[q+1]})*(dy_j^{[q+1]}/dS_j^{[q+1]})*(dS_j^{[q+1]}/dy_i^{[q]}),\ q = 1..p-1$$

$$y_i^{[q-1]} = dS_j^{[q]}/dw_{ij}^{[q]};$$

$$w_{ij}^{[q]}(t+1) = w_{ij}^{[q]}(t)-\alpha*(dE/dy_j^{[q]})*(dy_j^{[q]}/dS_j^{[q]})*y_i^{[q-1]},\ i = 1..n_q,\ q = 1..p$$

$$T_j^{[q]}(t+1) = T_j^{[q]}(t)+\alpha*(dy_j^{[q]}/dS_j^{[q]})*(dE/dy_j^{[q]}),\ i = 1..n_q,\ q = 1..p$$

Алгоритм и недостатки метода обратного распространения ошибки.
Адаптивный шаг обучения.
Метод многократного распространения ошибки. Алгоритм.
Послойное обучение. Метод и алгоритм.
Рециркуляционные PCA-сети. Обучение. Задача сжатия образов.
Рекуррентные сети (Джордана-Элмана).
Линейные сети и прогнозирование числовых последовательностей. Задача выбора окна.
Релаксационные сети. Сеть Хопфилда. Сеть Хэмминга.
…
Энергия сети Хопфилда.
Рассмотрим сеть Хопфилда.

$$y_i(t+1)=th(S_i(t))$$

$$S_i(t)= \sum W_{ij}Y_j-T_i$$

Чтобы изменение состояния и входной активности нейрона приводило к уменьшению энергии, выходное значение i-го нейрона должно быть пропорционально градиенту энергии:

$$y_i(t) = -dE(y_i(t))/dS_i(t)$$

так как, если $dS_i(t)>0$, то $y_i(t+1) \rightarrow 1 > 0$, а если $dS_i(t)<0$, то $y_i(t+1) \rightarrow -1 < 0$, поэтому $dE(y_i(t)) < 0$.

$$E(y_i(t)) = -\int y_i(t)dS_i = -y_i(t)*S_i(t)+\int S_i(t)dy_i$$

$$S_i(t) = F^{-1}(y_i(t))$$
$$E(y_i(t)) = -\int y_i(t)dS_i = -y_i(t)*S_i(t)+\int F^{-1}(y_i(t))dy_i$$
$$\Delta E(y_i(t+1)) = E(y_i(t+1))-E(y_i(t));$$
Для асинхронного режима:
$$\Delta E(y_i(t+1)) = -\Delta y_i(t+1)*S_i(t)+\int F^{-1}(y_i(t))dy_i$$
$$\Delta E(y_i(t+1)) = -\Delta y_i(t+1)*(S_i(t)-F^{-1}(\varepsilon));\ y_i(t+1) \leq \varepsilon \leq y_i(t)$$
$$\Delta E(y_i(t+1)) = \Delta y_i(t+1)*(F^{-1}(\varepsilon)-S_i(t))$$
Пусть $S_i(t) > S_i(t-1)$, тогда $y_i(t+1) > y_i(t)$, $\Delta y_i(t+1) > 0$ и $F^{-1}(\varepsilon) < S_i(t)$. Получаем $\Delta E(y_i(t+1)) < 0$.
Пусть $S_i(t) < S_i(t-1)$, тогда $y_i(t+1) < y_i(t)$, $\Delta y_i(t+1) < 0$ и $F^{-1}(\varepsilon) > S_i(t)$. Вновь получаем $\Delta E(y_i(t+1)) < 0$.
$$E(t)=\sum E(y_i(t))$$
Для сети с дискретным временем и состоянием в асинхронном режиме:
$$E(y_i(t)) = -y_i(t)*S_i(t);\ S_i(t) = -\Delta E(y_i(t+1))/\Delta y_i(t+1)$$
В синхронном режиме:
$$E(y_i(t)) = -y_i(t-1)*S_i(t)$$
Другие способы обучения сети Хопфилда. Метод проекций:
$$W=XX^+$$
$X^+$ – псевдоинверсия матрицы X. Если обучающие векторы линейно независимы, то
$$W=X(X^TX)^{-1}X^T$$
$$W^{(i)}=W^{(i-1)}+1/(\ [x^{(i)}]^Tx^{(i)}-[x^{(i)}]^TW^{(i-1)}x^{(i)})*[W^{(i-1)}x^{(i)}-x^{(i)}]*[W^{(i-1)}x^{(i)}-x^{(i)}]^T;\ W^{(0)}=0$$
Метод Δ-проекций – это градиентная форма алгоритма минимизации определённой особым образом функции. В соответствии с этим способом веса подбираются рекуррентно с помощью циклической процедуры, многократно повторяемой на всём множестве обучающих выборок:
$$W \leftarrow W+\eta/N[x^{(i)}-Wx^{(i)}][x^{(i)}]^T$$
Коэффициент $\eta$ – это константа обучения, выбираемая обычно из интервала 0,7-0,9. Его смысл тот же, что и в случае многослойных сетей. В отличие от обычного метода проекций метод Δ-проекций предполагает многократное предъявление всех р бучающих выборок вплоть до стабилизации значений весов. Процесс обучения завершается, когда изменения вектора весов становятся меньше априорно принятого значения $\varepsilon$.
Линейная ассоциативная память.
Дана обучающая выборка: $X_1,X_2,\ldots X_L$.
$$(X_i, X_j) = 0;\ i \neq j;\ (X_i, X_i) = 1;$$
$$W = \sum X_k^TY_k;\ w_{ij} = \sum x_i^ky_j^k;$$
$$Y=X_iW=\sum X_i(X_j)^TY+X_i(X_i)^TY=0+1*Y$$
Двунаправленная ассоциативная память.
Синхронный характер функционирования. Позволяет кодировать множества двух взаимосвязанных векторов x[n], y[p].
$$W=\sum x_i^Ty_i;\ i=1..m$$
Энергия:
$$E(k)=-x(k)Wy(k)^T$$
При обучении по правилу Хебба: $L=m/(4*\ln(m));\ m = \sqrt{(\min(n,p))}$.
Модификация правила Коско:
$$W=\sum x_i^Ty_i+(q-1)x_j^Ty_j$$
инкрементируем q, пока пара значений $x_j$, $y_j$ не начнёт распознаваться.
$$E'(i)= -x(i)Wy(i)^T-(q-1)\ x(i)x_j^Ty_jy(i)^T$$

$q \geq \max(1, \varepsilon_{0i}^{A}/2p+1, \varepsilon_{0i}^{B}/2n+1)$, где $\varepsilon_{0i}^{A} = \max(E_{0i}^{A}-E'_{0i})$, $\varepsilon_{0i}^{B} = \max(E_{0i}^{B}-E'_{0i})$ – равны максимальным разностям энергии i-го вектора и векторов, отстоящих от него на расстояние Хэмминга, равное 1, во множествах A и B соответственно.

Модифицированная структура двунаправленной ассоциативной памяти.

$$W_f=[WW_y];\ W_b=[W^TW_x]$$

Пусть p' и n' обозначают количество обучающих пар, для которых в процессе распознавания получены неправильные ответы для векторов y и x соответственно. Индексами y и x будем обозначать процессы, приводящие к формированию ошибочных векторов y и x соотвественно. Если $(x_i, y_i)$ является очередной k-й обучающей парой, для которой $f(x_iW) \neq y_i$, то принимается $y'_{ik}=1$, $y'_{ij}=0$ для $j \neq k$ ($k = 1, 2, \dots, p'$). Если для $(x_i, y_i)$ выполняется условие $f(x_iW) = y_i$, то $y'_{ik}=0$ для $k = 1, 2, \dots, p'$. Компоненты $y'_{ik}$ образуют вектор $y'_i$ длиной p'. Аналогичным образом для процессов, распомтраняющихся в противоположном направлении, при замене векторов y на x можно получить векторы $x'_i$ длиной n'. Корректирующие матрицы $W_x$ и $W_y$ формируются согласно формулам:

$$W_y = \sum x_i^T y'_i$$

$$W_x = \sum y_i^T x'_i$$

На следующем шаге создаются матрицы дополнительных узлов сети $T_y$ и $T_x$, причём

$$T_y = \sum q_y y'^T_j y_j$$

$$T_x = \sum q_x x'^T_j x_j$$

Параметры $q_y$ и $q_x$ подбираются таким образом, чтобы они соответствовали условиям:

$$q_x > n(m-2)-2\min(\sum d_H(x_i,x_j));\ x_i, x_j \text{ – вектора из обучающей выборки}$$

$$q_y > p(m-2)-2\min(\sum d_H(y_i,y_j));\ y_i, y_j \text{ – вектора из обучающей выборки}$$

При использовании введённых обозначений получаем:

$$y_1=f(x_0W+g_y(x_0W_y)T_y)$$

$$x_1=f(y_1W+g_x(y_1W_x)T_x)$$

$$\dots$$

$$y_f=f(x_{f-1}W+g_y(x_{f-1}W_y)T_y)$$

$$x_f=f(y_fW+g_x(y_fW_x)T_x)$$

Векторы значений функций активации:

$$g_y(v) = [g_y(v_1), g_y(v_2), \dots, g_y(v_{p'})]$$

$$g_x(v) = [g_x(v_1), g_x(v_2), \dots, g_x(v_{n'})]$$

подбираются следующим образом:

$$g_y(v_i)=(1 \text{ для } v_i>n-\varepsilon_1;\ 0 \text{ – иначе});$$

$$g_x(v_i)=(1 \text{ для } v_i>p-\varepsilon_2;\ 0 \text{ – иначе});$$

$$0 << \varepsilon_1 << 2*\min(d_H(x_i,x_j));$$

$$0 << \varepsilon_2 << 2*\min(d_H(y_i,y_j)).$$

Машина Больцмана.

$P_i(S)=1/(1+\exp(-\Delta E_i/T))$; $T(t+1)=a(T)$

$T(t)=T_0/(1+\log(t))$

Алгоритм функционирования машины Больцмана.

1. Устанавливаются в начальное состояние нейронные элементы и температура сети t = 0.
2. Случайным образом из невыбранных выбирается нейронный элемент с номером i. Добавить его в выбранные.
   $y_i = (1 \text{ random} \leq y_i';\ 0 \text{ random} > y_i')$; $y_i'=1/(1+\exp(-S_i/T(t)))$
3. Вычисляется $\Delta E_i$.
4. Если $\Delta E_i < 0$, то $y_i(t+1) = y_i$.

5. Если $\Delta E_i \geq 0$, то с вероятностью $P_i$ $y_i(t+1) = y_i$, иначе – $y_i(t+1) = y_i(t)$;
6. Повторить с пункта 2, если остались невыбранные.
7. Проверяется условие равновесия ($T(t)=0$ и $\Delta E_i < 0$ для всех i). Если оно не выполняется, то $t = t+1$ и все нейроны сети невыбранные и перейти на шаг 2.