Карпов В.Э.

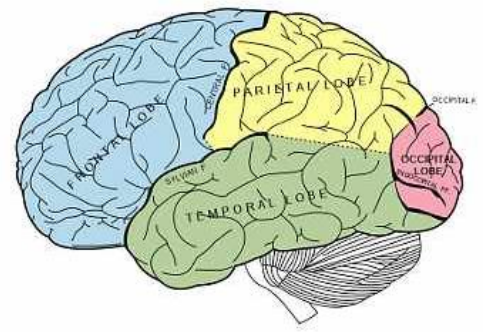
Классические нейронные сети

Краткая историческая справка

Два направления нейронного моделирования:

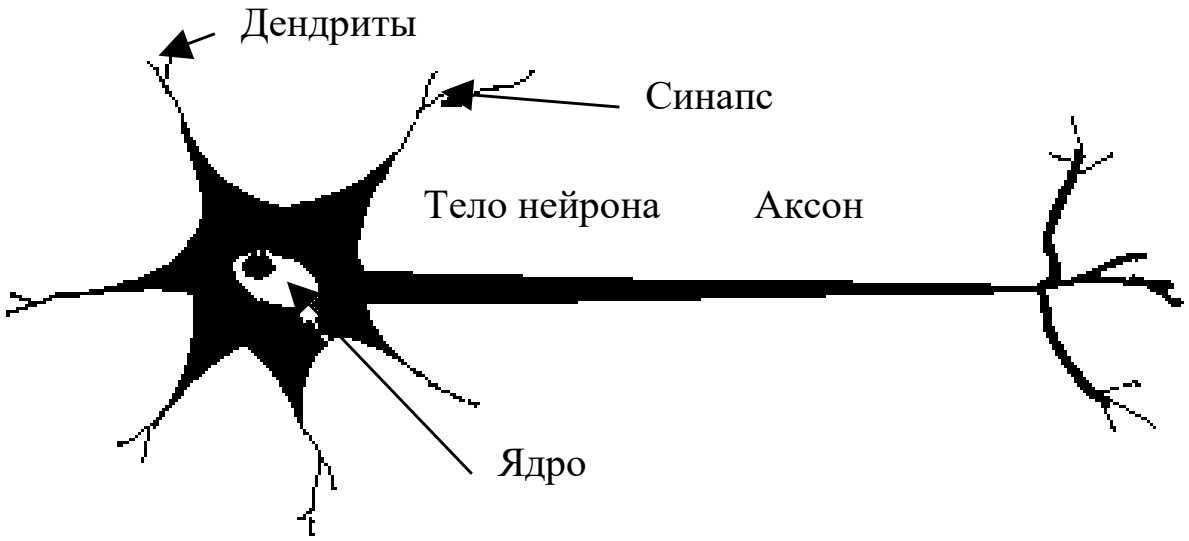
1. Понять механизмы функционирования нервной системы человека на уровне физиологии и психологии.
2. Создать ВС, выполняющие функции сходные с функциями мозга.

- Д. Хэбб в 1949 г. предложил закон обучения искусственных нейронных сетей (ИНС).
- 50-60 гг. первые искусственные нейронные сети. Минский, Розенблатт, Уидроу и др. Однослойные сети (персептроны).
- Доказательство Минского о неспособности однослойных сетей решать многие простые задачи. Перерыв в 20 лет.
- В 80-90 гг. возобновились разработки ИНС, которые показали, что Минский был слишком пессимистичен.



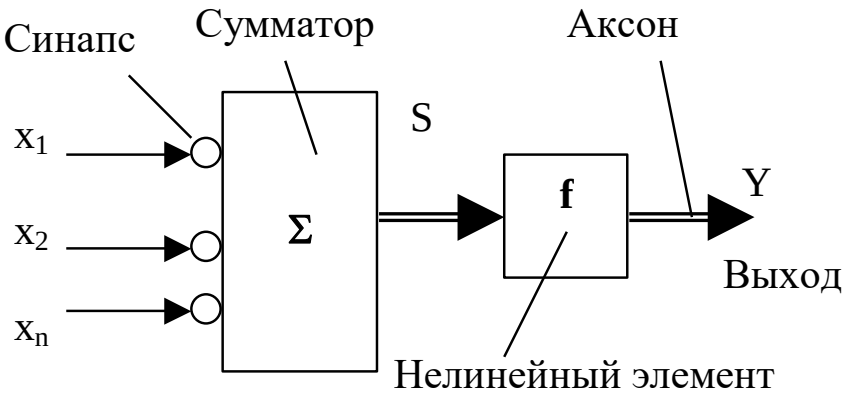
Биологические нейронные сети

- Нервная система человека имеет 100 млрд. нейронов участвующих примерно в 10^{15} передающих связях.
- Нейрон осуществляет обработку и передачу электрохимических сигналов по нервным путям, которые образуют коммуникационную



Формальная модель нейрона

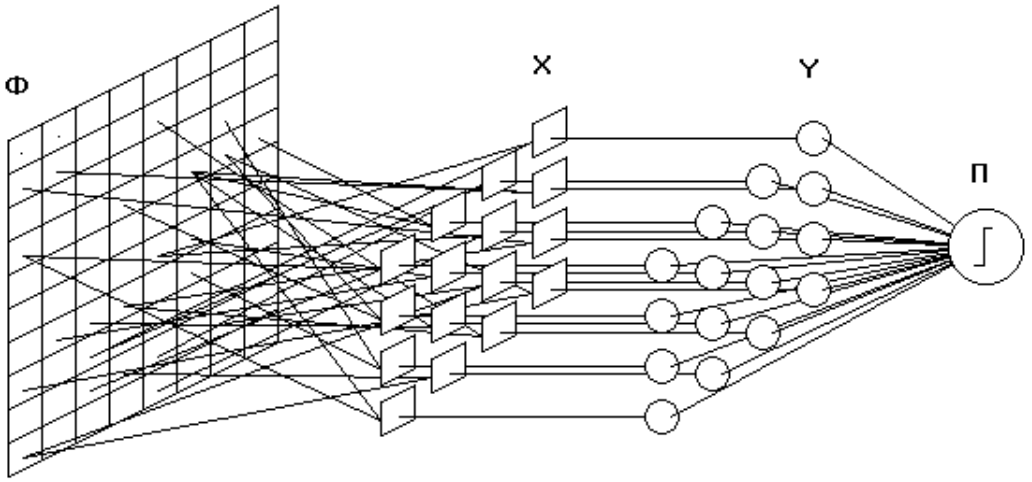
- В 1943 г. Мак-Каллок, Питс.
- $X=\{x_1,x_2,...,x_n\}$ - множество входных сигналов.
- $W=\{w_1,w_2,...,w_\eta\}$ – вес, который соответствует "силе" одной биологической синаптической связи.
- Σ- блок суммирования, соответствующий телу биологического элемента.
- S - выход блока суммирования.
- Эта модель послужила основой для создания одной из первых работ в области бионического направления ИИ – *персептрона*.



$$S = \sum_{i=1}^{n} w_i x_i$$

Персептрон

- 1957 г. (середина 1958 ?), физиолог Фрэнк Розенблатт (F.Rosenblatt).
- 1959 - демонстрация первой действующей машины «MARK-1» (Корнельская авиационная лаборатория).



“Имитация процессов человеческого мышления”

- 10^4 (100x100) фотоэлементов Φ_i. (+1,0).
- Элементы X_i: +1 / -1
- Если сумма приходящих от фотоэлементов Φ_i сигналов меньше порога срабатывания, то на выходе элемента выдается сигнал -1, если больше - то +1.
- Y_i. Усилители с коэффициентами λ_i.
- П - выходное устройство: +1 и -1.
- Образ 1: $z_1+z_2+z_3+...+z_n<0 \Rightarrow П=-1$
- Образ 2: $z_1+z_2+z_3+...+z_n >0 \Rightarrow П=+1$

Обучение персептрона

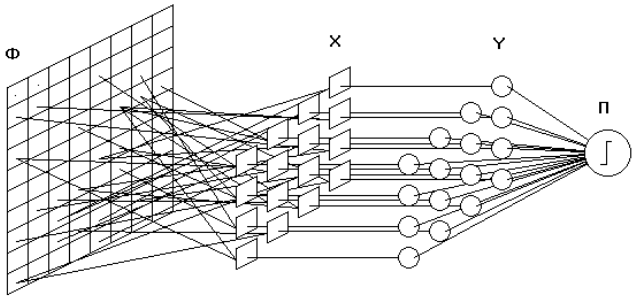
- Обучение - настройка весов и порогов: увеличивались значения весов правильно сработавших связей, уменьшались для сработавших неправильно.
- *Обучающее воздействие* в персептроне реализуется за счет изменения коэффициентов усиления λ_i всех усилителей второго слоя Y.

Закон обучения:

1. $\lambda'_i = \lambda_i + \delta y_i$
2. где λ'_i - новый коэффициент усиления.
3. δ =0 при правильной классификации, и ± 1 - при ошибочной: если изображение А отнесено к классу В, то δ=+1, а если изображение В определено как А, то δ=-1.

Розенблатт: чем больше персептрон, тем меньше он нуждается в обучении => в пределе бесконечный персептрон учить не надо, он уже полностью обучен.

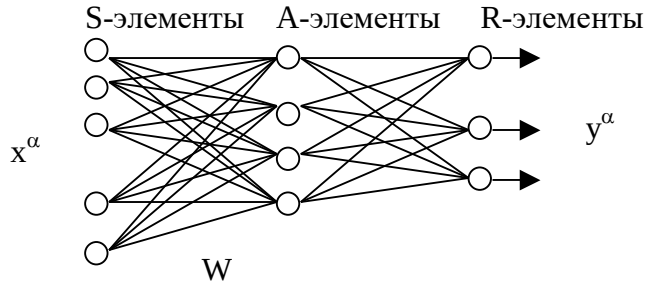
Исходя из частных признаков объектов, персептрон способен организовывать общее - их разделение на классы



Формальное описание персептрона

Обучение сети состоит в подстройке весовых коэффициентов каждого нейрона.

- Пусть имеется набор пар векторов (x^a, y^a), a=1..p - *обучающая выборка*.
- НС обучена на данной обучающей выборке, если при подаче на входы сети каждого вектора $\mathbf{x^a}$ на выходах всякий раз получается соответствующий вектор $\mathbf{y^a}$
- Метод обучения состоит в итерационной подстройке матрицы весов, последовательно уменьшающей ошибку в выходных векторах.



0. Начальные значения весов всех нейронов W(t=0) полагаются случайными.
1. Сети предъявляется входной образ x^a, в результате формируется выходной образ y'^α ($y'^\alpha \neq y^\alpha$)
2. Вычисляется вектор ошибки $\delta^\alpha = (y'^\alpha - y^\alpha)$.

   Далее изменение вектора весовых коэффициентов должно быть **пропорционально ошибке на выходе**.
3. Вектор весов модифицируется по следующей формуле: $W(t+\Delta t)=W(t)+\eta x^\alpha (\delta^\alpha)^T$. Здесь $0<\eta<1$- темп обучения.
4. Переход к П.1.

Ограниченность персептрона

- Розенблатт предположил, что НС способна к воспроизведению *любой* логической функции.
- В 1969 г. М.Минским и С.Пейпертом (M.Minsky, S.Papert) было доказано, что это не так. Были выявлены принципиальные неустранимые ограничения однослойных персептронов.
- Доказанная Розенблаттом теорема о сходимости обучения по δ-правилу говорит лишь о том, что персептрон способен обучится любому обучающему набору, который *он способен представить*.

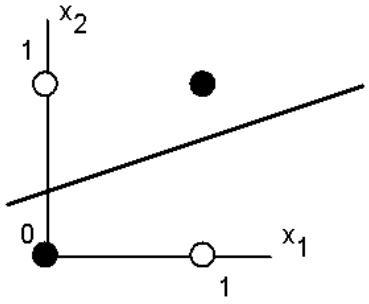
Линейная разделимость и персептронная представляемость

- Каждый нейрон персептрона является формальным пороговым элементом:

$$y_j = \begin{cases} 1, & \sum_i W_{ij} x_i > \Theta_j \\ 0, & \sum_i W_{ij} x_i \le \Theta_j \end{cases}$$

- Т.о., при заданных значениях весов и порогов, нейрон имеет определенное значение выходной активности для каждого возможного вектора входов.
- Множество входных векторов, при которых нейрон активен (y=1), отделено от множества векторов, на которых нейрон пассивен (y=0) *гиперплоскостью*, уравнение которой есть

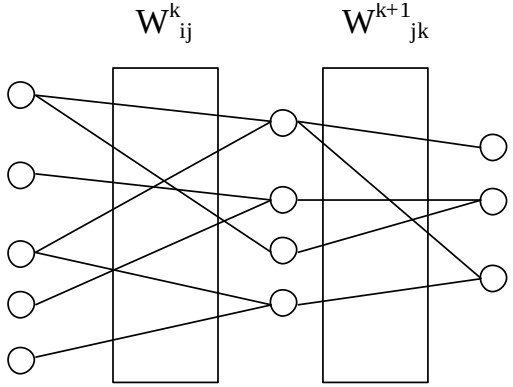
$$\sum_i W_{ij} x_i - \theta_j = 0$$



Число переменных N	Полное число возможных логических функций $(= 2^{2^N})$	Из них линейно разделимых функций
1	4	4
2	16	14
3	256	104
4	65536	1882
5	$> 10^9$	94572

Многослойные нейронные сети

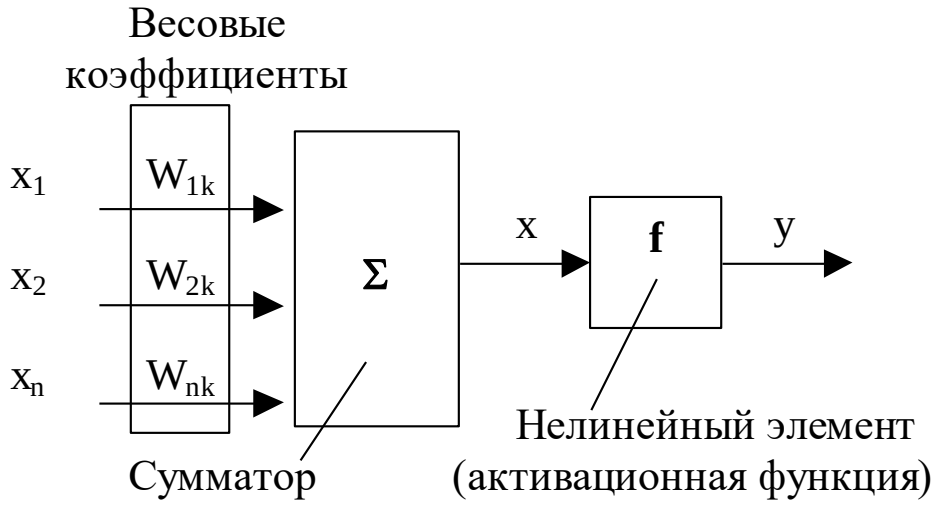
"Усовершенствование персептрона"



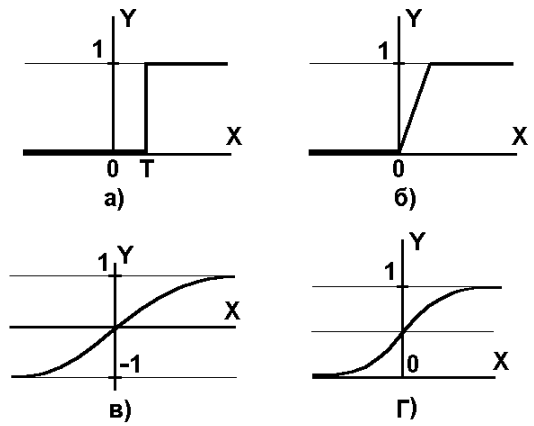
- Выход слоя k:
  $Y^k = W^k X_k$
- Эти сигналы являются входом следующего слоя k+1 и, пользуясь ассоциативностью матричного произведения, получаем:
- $Y^{k+1} = W^{k+1} X_{k+1} = W^{k+1} Y^k = W^{k+1} (W^k X_k) = (W^{k+1} W^k) X_k = W X_k$
- Следовательно, два слоя, представленных матрицами W^{k+1} и W^k можно заменить одним.

Т.о., возможный выход из сложившейся ситуации - переход к иерархическим многослойным структурам, состоящим из ***нелинейных*** **нейронов**.

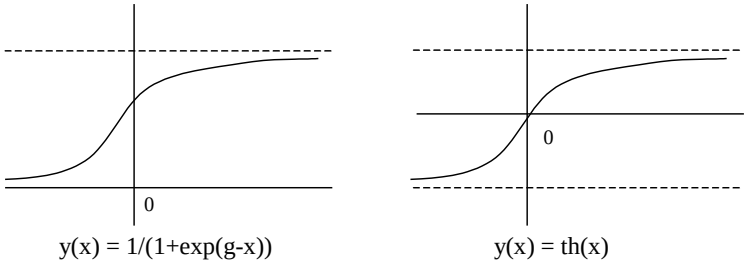
Нелинейные элементы



$$y = \frac{1}{1 + e^{-(\sum_i W_i x_i - \Theta)}}$$



- **Гиперболический тангенс**
- **Сжимающая функция**



$$f(x) = \frac{1}{1 + e^{g - x}}$$

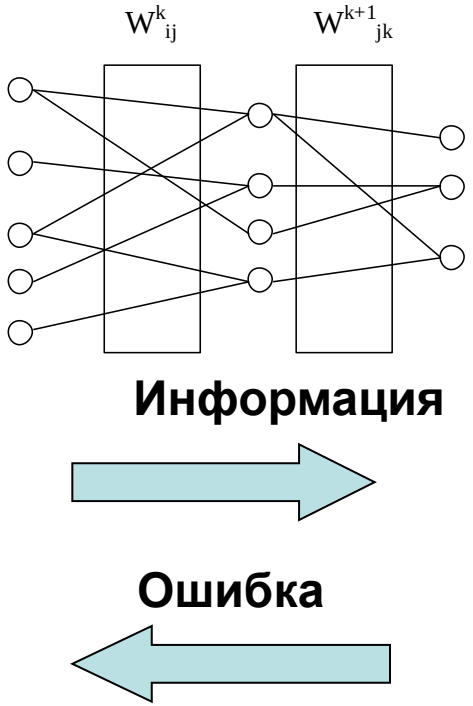
g=1/T, где T – это пороговое значение нейрона.
Такой нейрон обрабатывает как слабые сигналы, которые он усиливает, так и сильные

Обучение нейрона

1986 г., Руммельхарт и Хинтон (Rummelhart D.E., Hinton G.E.). **Алгоритм обратного распространения ошибок (error back propagation).**

- Основная идея состоит в том, как получить оценку ошибки для нейронов **скрытых** слоев.
- *Известные* ошибки нейронов выходного слоя возникают вследствие *неизвестных* пока ошибок нейронов скрытых слоев.

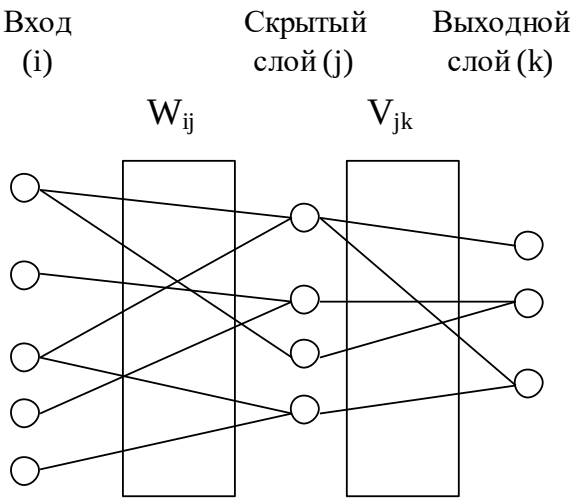
1. Чем больше значение синаптической связи между нейроном скрытого слоя и выходным нейроном, тем сильнее ошибка первого влияет на ошибку второго.
2. =>оценку ошибки элементов скрытых слоев можно получить, как взвешенную сумму ошибок последующих слоев.
3. При обучении информация распространяется от низших слоев иерархии к высшим, а оценки ошибок, делаемые сетью - в **обратном** направлении.



Математическое описание

- Хорошие свойства сигмоида

$$f(x)=\frac{1}{1+e^{-ax}}$$ f'(x) = a·f(x)·(1-f(x))



Элементы выходного слоя

$$x_k=\sum_j V_{jk}y_j \qquad y_k=f(x_k)$$

Элементы скрытого слоя

$$x_j=\sum_i W_{ij}x_i^{\alpha} \qquad y_j=f(x_j)$$

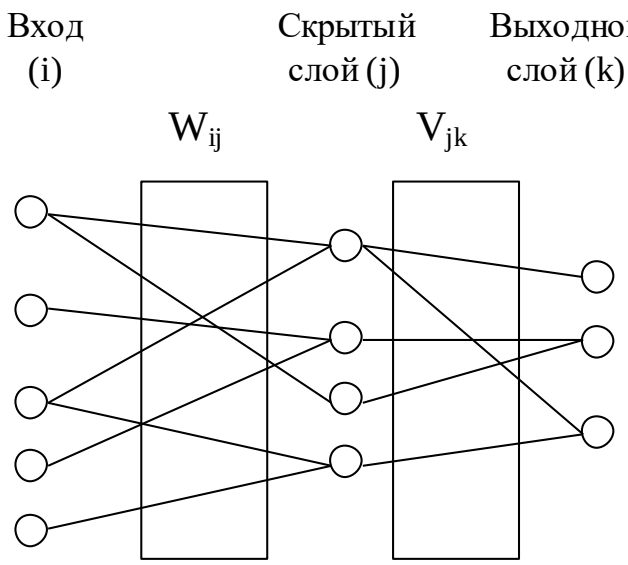
Функция ошибки распознавания входного образа $$E=\frac{1}{2}\sum(y_k-Y_k^{\alpha})^2$$

Пусть обучение должно заключаться в **минимизации функции ошибки** за счет подстройки весов выходного (k) и скрытого (j) слоев.
Классический градиентный метод оптимизации состоит в итерационном уточнении весов выходного и скрытого слоев

$$V_{jk}(t+1)=V_{jk}(t)-h\frac{\partial E}{\partial V_{jk}} \qquad W_{ij}(t+1)=W_{ij}(t)-h\frac{\partial E}{\partial W_{ij}}$$

Метод обратного распространения ошибок

(Руммельхарт и Хинтон (Rummelhart D.E., Hinton G.E.), 1986)



$$x_k = \sum_j V_{jk} y_j \qquad y_k = f(x_k)$$

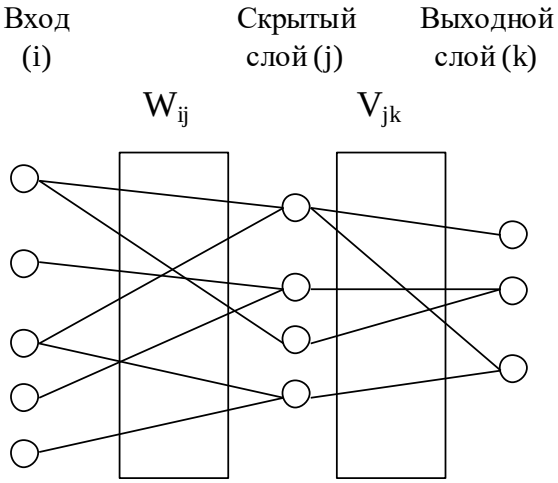
$$x_j = \sum_i W_{ij} x_i^{\alpha} \qquad y_j = f(x_j)$$

$$f(x) = \frac{1}{1 + e^{-ax}}$$

Функция ошибки $$E = \frac{1}{2} \sum (y_k - Y_k^{\alpha})^2$$

Градиентный метод оптимизации

Итерационное уточнение весов выходного и скрытого слоев:



$$V_{jk}(t+1) = V_{jk}(t) - h\frac{\partial E}{\partial V_{jk}}$$

$$W_{ij}(t+1) = W_{ij}(t) - h\frac{\partial E}{\partial W_{ij}}$$

$$\frac{\partial E}{\partial V_{jk}} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial x_k}\frac{\partial x_k}{\partial V_{jk}} \qquad \frac{\partial E}{\partial y_k} = y_k - Y_k^{\alpha} = \delta_k \qquad \frac{\partial y_k}{\partial x_k} = y_k \cdot (1 - y_k)$$

(свойство сигмоидальной функции f(x): f’(x)=f(x)·(1-f(x)))

$$\frac{\partial x_k}{\partial V_{jk}} = y_j$$

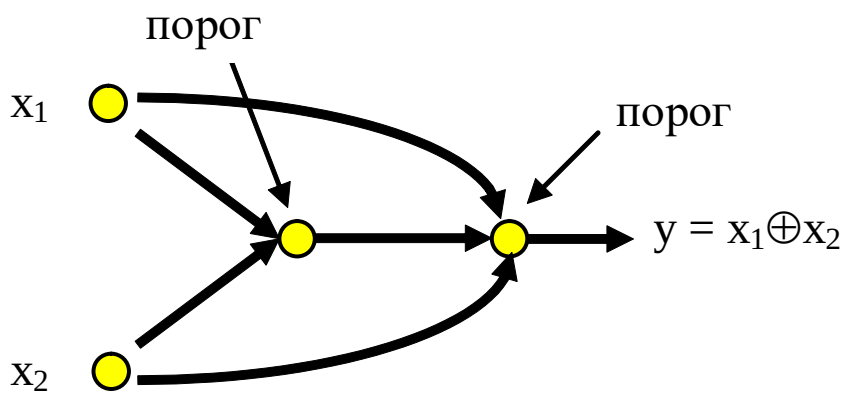
$$\frac{\partial E}{\partial V_{jk}} = \delta_k \cdot y_k \cdot (1 - y_k) \cdot y_j \qquad \frac{\partial E}{\partial W_{ij}} = [\sum \delta_k \cdot y_k \cdot (1 - y_k) \cdot V_{jk}][y_j \cdot (1 - y_j) \cdot x_i^{\alpha}]$$

Решение задачи "Исключающего ИЛИ"

$$\frac{\partial E}{\partial y_j} = \sum \frac{\partial E}{\partial x_k}\frac{\partial x_k}{\partial y_j} = \sum \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial x_k}\frac{\partial x_k}{\partial y_j} = \sum \delta_k \cdot y_k \cdot (1 - y_k) \cdot V_{jk}$$

$$\frac{\partial E}{\partial W_{ij}} = [\sum \delta_k \cdot y_k \cdot (1 - y_k) \cdot V_{jk}][y_j \cdot (1 - y_j) \cdot x_i^{\alpha}]$$

1400 итераций.



Особенности обучения

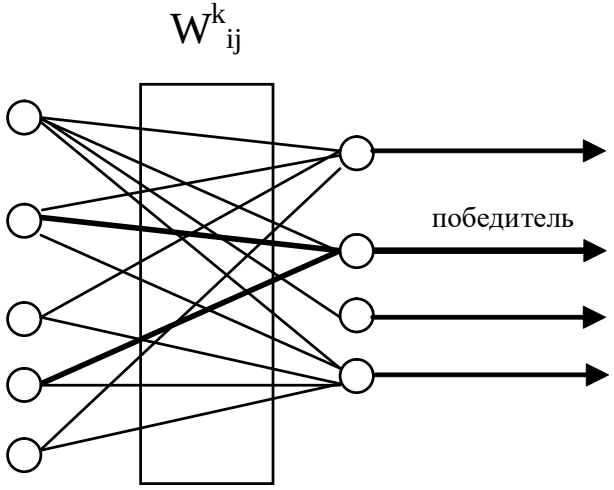
- обучение сводится к решению задачи оптимизации функционала ошибки градиентным методом
- попадание в локальный экстремум и т.н. "паралич сети"
- медленная сходимость метода обратного распространения

Карта самоорганизации Кохонена

Модель Т.Кохонена (T.Kohonen, 1982, 1984).

- Обобщение предъявляемой информации.
- В результате работы НС получается образ, представляющий собой карту распределения векторов из обучающей выборки. Таким образом, в модели Кохонена выполняется решение задачи нахождения кластеров в пространстве входных образов.

- При предъявлении сети входного вектора возбуждается единственный нейрон, наиболее точно соответствующий входному образу.
- Механизм конкуренции между нейронами.

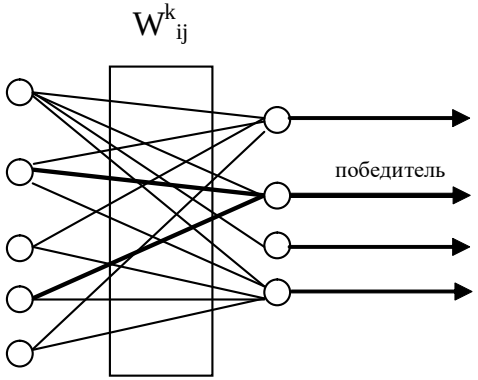


$$W^t = W^{t-1} + \Delta W^t$$

$$\Delta W^t = \eta \cdot \left(x^{t-1} - W^{t-1} \right)$$

Принцип работы

- Сеть обучается без учителя на основе самоорганизации.
- По мере обучения вектора весов нейронов стремятся к центрам кластеров - групп векторов обучающей выборки.
- Сеть относит новый предъявленный образ к одному из сформированных кластеров, указывая тем самым категорию, к которой он принадлежит.



- Сеть состоит из одного слоя нейронов.
- Число входов каждого нейрона равно размерности входного образа.
- Количество нейронов определяется той степенью подробности с которой требуется выполнить кластеризацию набора входных образов.

$$d_m = \sum_{i=1}^{N} (x_i(t) - W_i^m(t))^2$$

- Обучение начинается с задания случайных значений матрице связей W.
- Далее происходит процесс самоорганизации, состоящий в модификации весов при предъявлении на вход векторов обучающей выборки.

Проецирование многомерного пространства в пространство с более низкой размерностью

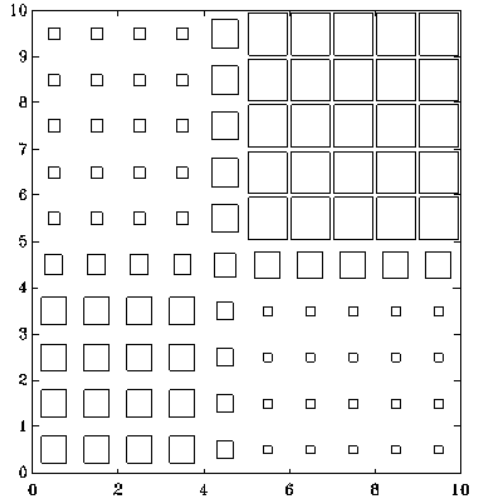
Обучение

1. Для каждого нейрона m определяется его расстояние до вектора входа d_m
2. Выбирается нейрон m=m*, для которого это расстояние минимально.
3. На текущем шаге обучения t будут модифицироваться только веса нейронов из окрестности нейрона m*.
4. Первоначально в окрестности любого из нейронов находятся все нейроны сети.
5. Далее эта окрестность сужается.
6. В конце этапа обучения подстраиваются только веса ближайшего нейрона.
7. Темп обучения h(t)<1 с течением времени уменьшается.

Образы обучающей выборки предъявляются последовательно, и каждый раз происходит подстройка весов.

$$d_m = \sum_{i=1}^{N} (x_i(t) - W_i^m(t))^2$$

Пример карты Кохонена. Размер каждого квадратика соответствует степени возбуждения соответствующего нейрона



Звезды Гроссберга

Гроссберг (S. Grossberg), 1969

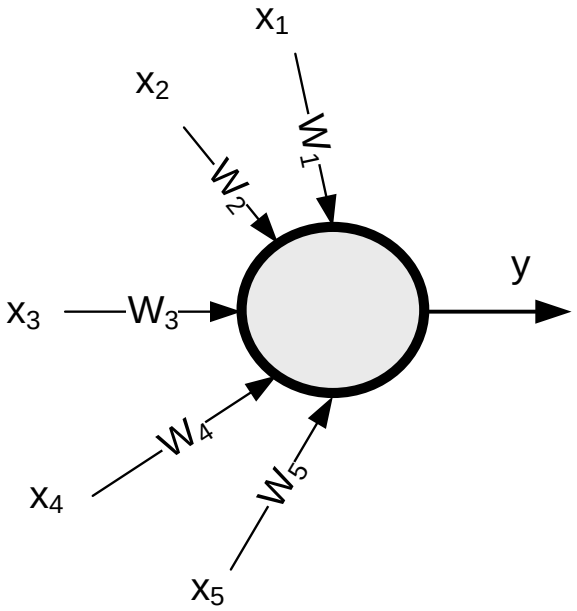
Входная звезда

N входов $X_1..X_N$, которым соответствуют веса $W_1..X_N$, и один выход Y, являющийся взвешенной суммой входов.

Входная звезда обучается выдавать сигнал на выходе всякий раз, когда на входы поступает определенный вектор.

Т.о., входная звезда является детектором совокупного состояния своих входов.

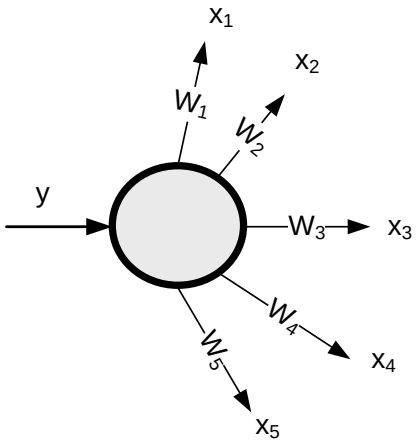
Процесс обучения представляется в итерационной форме:



$$W_i(t+1) = W_i(t) + \alpha \cdot \left(X_i - W_i(t)\right)$$

Выходная звезда

- **Выходная звезда** Гроссберга выполняет противоположную функцию - функцию командного нейрона, выдавая на выходах определенный вектор при поступлении сигнала на вход.
- Нейрон этого типа имеет один вход и M выходов с весами $W_{1..M}$, которые обучаются по формуле:



$$W_i(t+1) = W_i(t) + \beta \cdot \left(Y_i - W_i(t)\right)$$

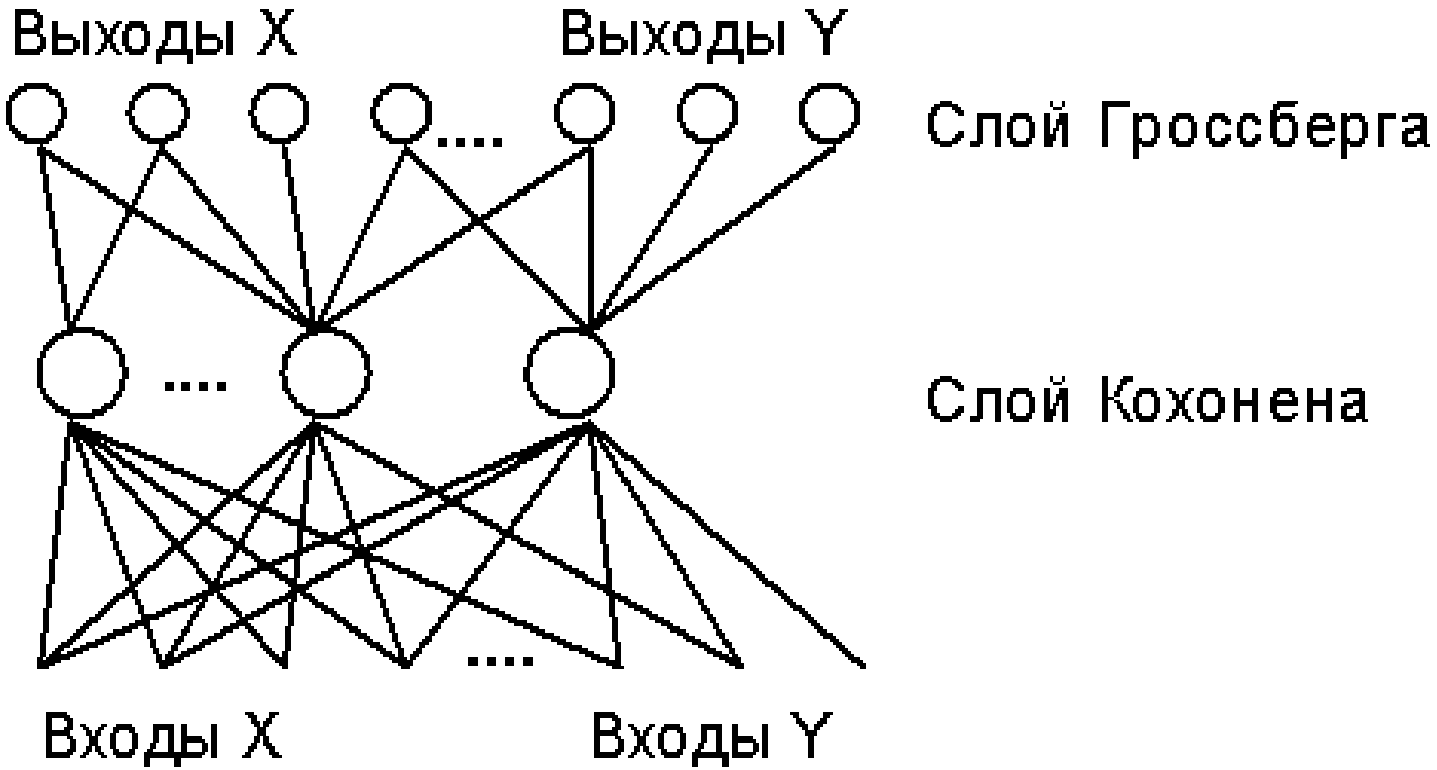
Итерационный процесс будет сходиться к собирательному образу, полученному из совокупности обучающих векторов.

Особенность звезд - **локальность** памяти. Каждая входная звезда помнит "свой" относящийся к нему образ и игнорирует остальные.

Каждой выходной звезде присуща также конкретная командная функция. Образ памяти связывается с определенным нейроном, а не возникает вследствие взаимодействия множества нейронов в сети.

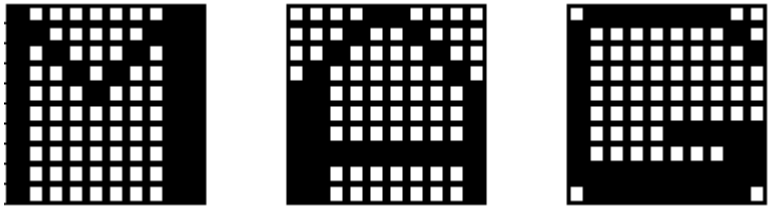
Сети встречного распространения

- Возможность обобщения информации сети Кохонена и простота обучения выходной звезды Гроссберга



СПИНОВЫЕ СТЕКЛА

- Дж.Хопфилд. Динамическая система с ассоциативной памятью - "спиновое стекло".
- N элементов - т.н. *изинговых спинов*, каждый из которых может находиться в одном их двух состояний: $s_i = \pm 1$.
- Состояние системы характеризуется энергией взаимодействующих спинов и с течением времени система стремится минимизировать эту энергию за счет изменения состояний спинов.
- Спиновое стекло может содержать много состояний с минимумами полной энергии E, отвечающих различным спиновым конфигурациям. Таким образом, спиновое стекло "хранит в памяти" большое число картин - образов.
- Запись изображения осуществляется с помощью формирования матрицы синаптических связей J согласно правилу Хебба. После формирования этой матрицы системе предъявляется распознаваемое изображение. Степень соответствия этого изображения тому, что было записано определяется величиной полной энергии E.



Энергия взаимодействия

- N элементов, каждый из которых может находиться в одном их двух состояний: $s_i = \pm 1$.
- Энергия взаимодействия пары спинов

  $\delta E_{ij} = -J_{ij} s_i s_j$
- Полная энергия E взаимодействующих спинов

  $E \to \min$

$$E = -\frac{1}{2}\sum J_{ij} s_i s_j$$

J_{ij} - элементы матрицы взаимодействия

Запись образов

- M различных образов, каждый из которых характеризуется набором ориентированных спинов $\{\varepsilon_i(m)\}$, m=1,..,M, причем различные наборы *ортогональны*:

$$N^{-1}\sum_{j=1}^{N}\varepsilon_j^{(m)}\varepsilon_j^{(m')}=\delta_{mm'}$$

<u>Минимизация энергии E</u>

$J_{ij}=\varepsilon_i\varepsilon_j$.

($\delta E_{ij}=-J_{ij}\,\varepsilon_i\varepsilon_j=-\varepsilon^2_i\varepsilon^2_j=-1 \Rightarrow \delta E_{ij}$ минимальна)

Правило Хебба:

$$J_{ij}=\sum_{m=1}^{M}\varepsilon_i^{(m)}\varepsilon_j^{(m)}$$

Запись дополнительного образа M+1:

$J(M+1)_{ij}=J(M)_{ij}+\varepsilon_i^{(M+1)}\,\varepsilon_j^{(M+1)}$

Ложные образы

- Отклонения от требования строгой ортогональности приводят к появлению дополнительных случайных вариаций в поле h:

$$E = -\left(\frac{1}{2}\right)\sum h_i s_i \qquad h_i = \sum_j J_{ij} s_j$$

Запись новых образов

- Насколько система ("старая" матрица) распознает "новый" вектор

$$\varepsilon_i^{//} = N^{-1} \sum_j J_{ij}^{(M)} \varepsilon_j^{(M+1)}$$

- Если новый вектор $\varepsilon^{(M+1)}$ ортогонален ко всем старым, то $\varepsilon^{||}=0$. Если он уже хранится в памяти, то $\varepsilon^{||} = \varepsilon^{(M+1)}$ M изменять не надо

$$J_{ij}^{(M+1)} = J_{ij}^{(M)} + \varepsilon_j^r \varepsilon_j^{(M+1)} \qquad \varepsilon_j^r = \varepsilon_j^{(M+1)} - \varepsilon_j^{||}$$

$$J_{ij}^{(M+1)} = J_{ij}^{(M)} + \xi \varepsilon_i^r \varepsilon_j^{(M+1)}$$

$$J_{ij}^{(M+1)} = J_{ij}^{(M)} + \varepsilon_i^r \varepsilon_j^r / \varepsilon^r \varepsilon^r \qquad \varepsilon^r \varepsilon^r \equiv N^{-1} \sum \left(\varepsilon_j^r \right)^2$$

$J'_{ij} = \text{sign}(J_{ij})$

Предварительная ортогонализация

Насколько система ("старая" матрица) распознает "новый" вектор:

$$\varepsilon_i^{\prime\prime} = N^{-1} \sum_j J_{ij}^{(M)} \varepsilon_j^{(M+1)}$$

Если новый вектор $\varepsilon^{(M+1)}$ ортогонален ко всем старым, то $\varepsilon^{||}=0$

$$J_{ij} = \sum_{m_1, m_2}^{M} \left(R_{m_1, m_2}^{-1} \right) \varepsilon_i^{(m_1)} \varepsilon_j^{(m_1)} \qquad R_{m_1, m_2}^{-1} = N^{-1} \sum_i^N \varepsilon_i^{(m_1)} \varepsilon_{ij}^{(m_1)}$$

Ложные образы и навязчивые идеи

- 1) Записанные в память образы воспроизводятся системой с некоторыми искажениями.
- 2) M/N=0.05: "фазовый переход“ => образ может эволюционировать с течением времени к одному из ложных.
- 3) M/N=0.4: второй "фазовый переход“ все притягивающие конфигурации соответствуют только ложным образам.

Реализация модели

- Пусть есть сеть из N попарно связанных элементов (нейронов). Каждый нейрон - это автомат с двумя состояниями $s_i = \pm 1$. Связь между автоматами i и j характеризуется весовым коэффициентом J_{ij}.
- Пусть s_i^n - состояние всех элементов в момент времени n.

1. Выбираем произвольный элемент i;
2. Вычисляем действующее на него поле h: $h_i = \sum_j J_{ij} s_j^n$
3. Новое состояние элемента $s_i^{n+1} = \text{sign } h_i^n$

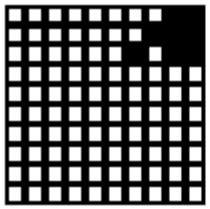
   Или $$s_i^{n+1} = sign\left(\sum_{j=1}^{N} J_{ij} s_j^n \right)$$

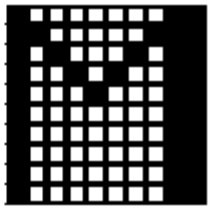
А Б

Примеры использования

1. Распознавание фраз. При этом символы кодируются по битам.
2. Распознавание символов. Изображение символов кодируется по точкам. В силу корреляции изображений букв, предпочтительнее использовать предварительную ортогонализацию

$$J_{ij} = \sum_{m_1,m_2}^{M} \left(R^{-1}_{m_1,m_2} \right) \varepsilon_i^{(m_1)} \varepsilon_j^{(m_2)} \qquad R_{m_1,m_2} = N^{-1} \sum_{i}^{N} \varepsilon_i^{(m_1)} \varepsilon_i^{(m_2)}$$





Оценка отклонений

- При любом конечном значении отношения M/N записанные в память образы воспроизводятся системой с некоторыми **искажениями**. Всегда имеются ложные устойчивые образы, представляющие собой суперпозицию из M записанных в память картин.
- Примерно при M/N=0.05 в модели наблюдается **"фазовый переход"**: минимумы энергии, отвечающие истинным образам, становятся менее глубокими, чем те, которые соответствуют ложным образам. => Даже содержащий незначительные случайные искажения предъявленный образ может эволюционировать с течением времени не к ближайшему из записанных образов, а к одному из ложных.
- Примерно при M/N=0.4 в системе происходит второй "фазовый переход", после которого записанные картины перестают отвечать минимумам энергии, а все притягивающие конфигурации соответствуют **только ложным образам**.

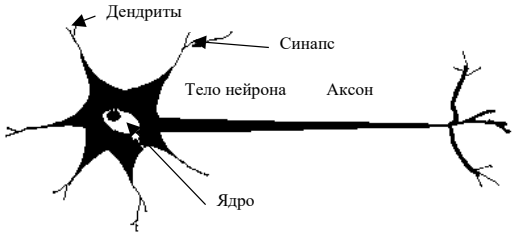
Задачи для нейронных сетей

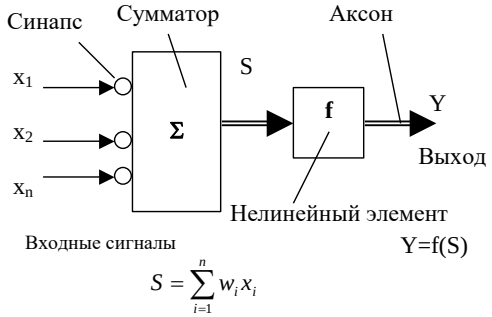
НС являются универсальными аппроксиматорами функций многих переменных

- построение функции по конечному набору значений;
- оптимизация;
- построение отношений на множестве объектов;
- распределенный поиск информации и ассоциативная память;
- фильтрация;
- сжатие информации;
- идентификация динамических систем и управление ими;
- нейросетевая реализация классических задач и алгоритмов вычислительной математики: решение систем линейных уравнений, решение задач математической физики сеточными методами и др.

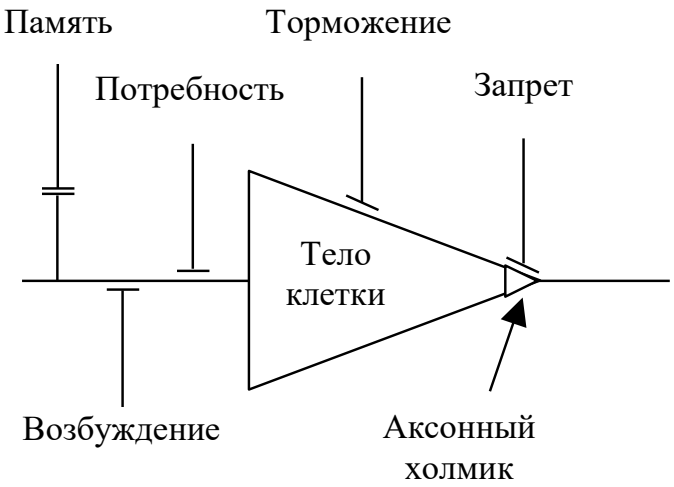
Большие пирамидные нейроны

- Спинной мозг и решение интеллектуальных задач



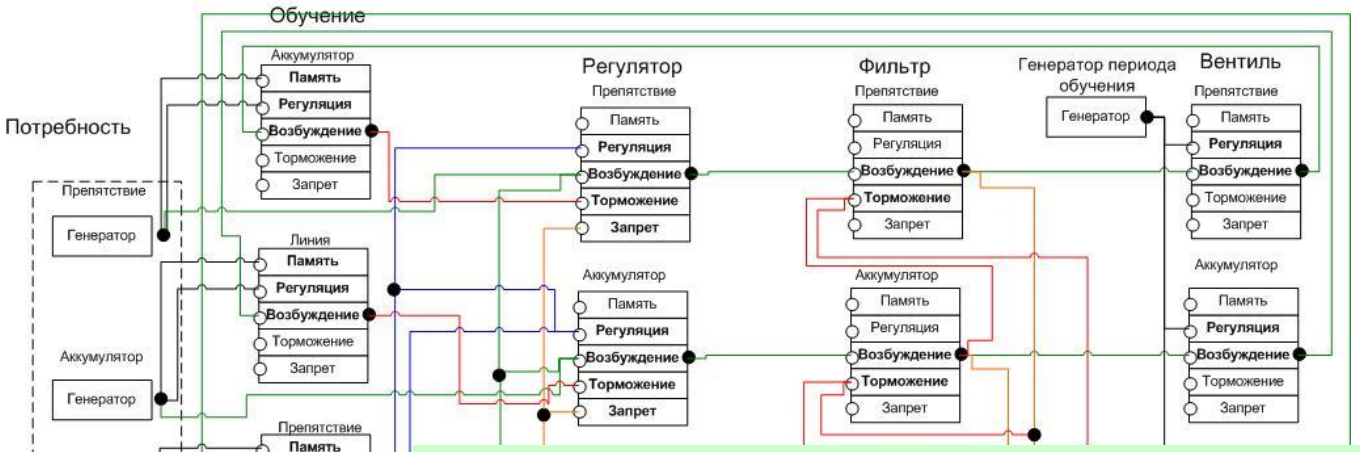


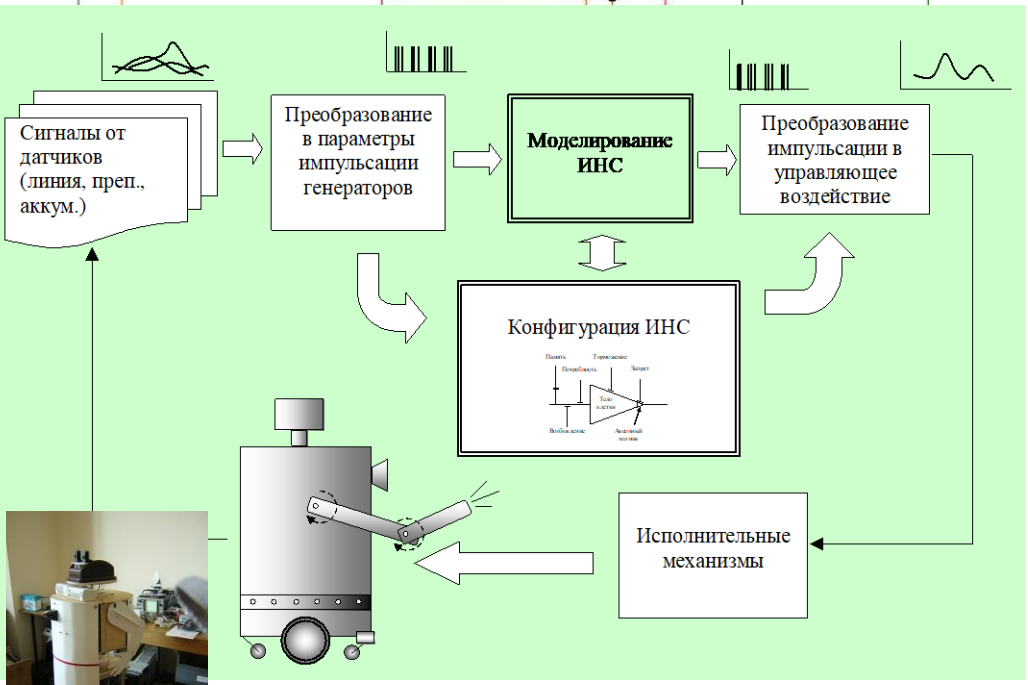
- **Большой пирамидный нейрон** (интеллектуальный нейрон), В.Б. Вальцев

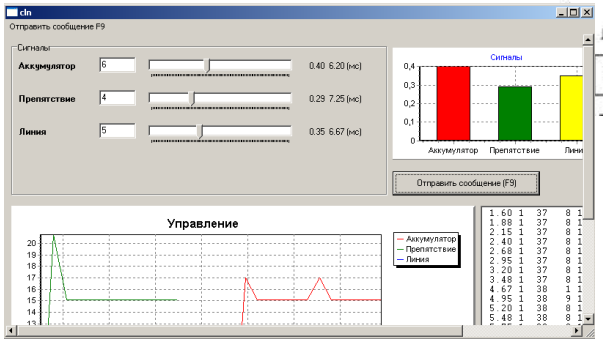


Интеллектуальный нейрон. Брейнпьютер

Задача планирования поведения робота







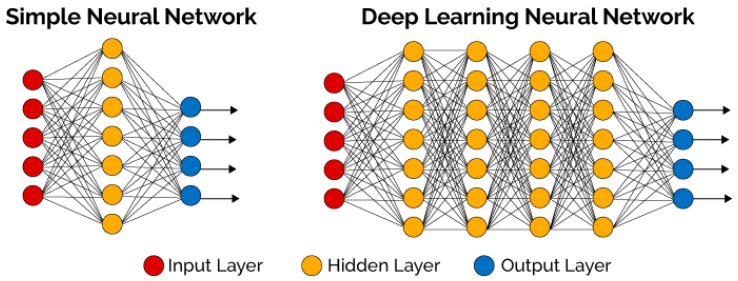
НС сегодня

1. НС – универсальный аппроксиматор:

X→[НС]->Y

2. От маленького молотка к паровому молоту:

- большие вычислительные мощности
- много данных (примеры…)

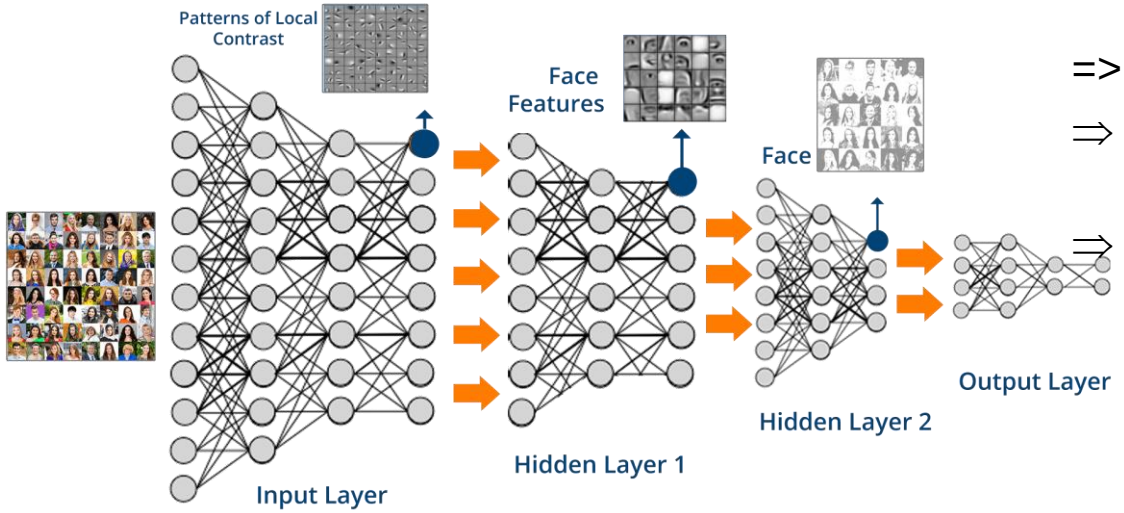


1. Нет понимания сути интеллекта
2. Нет моделирования процесса мышления:

=> нет универсальности

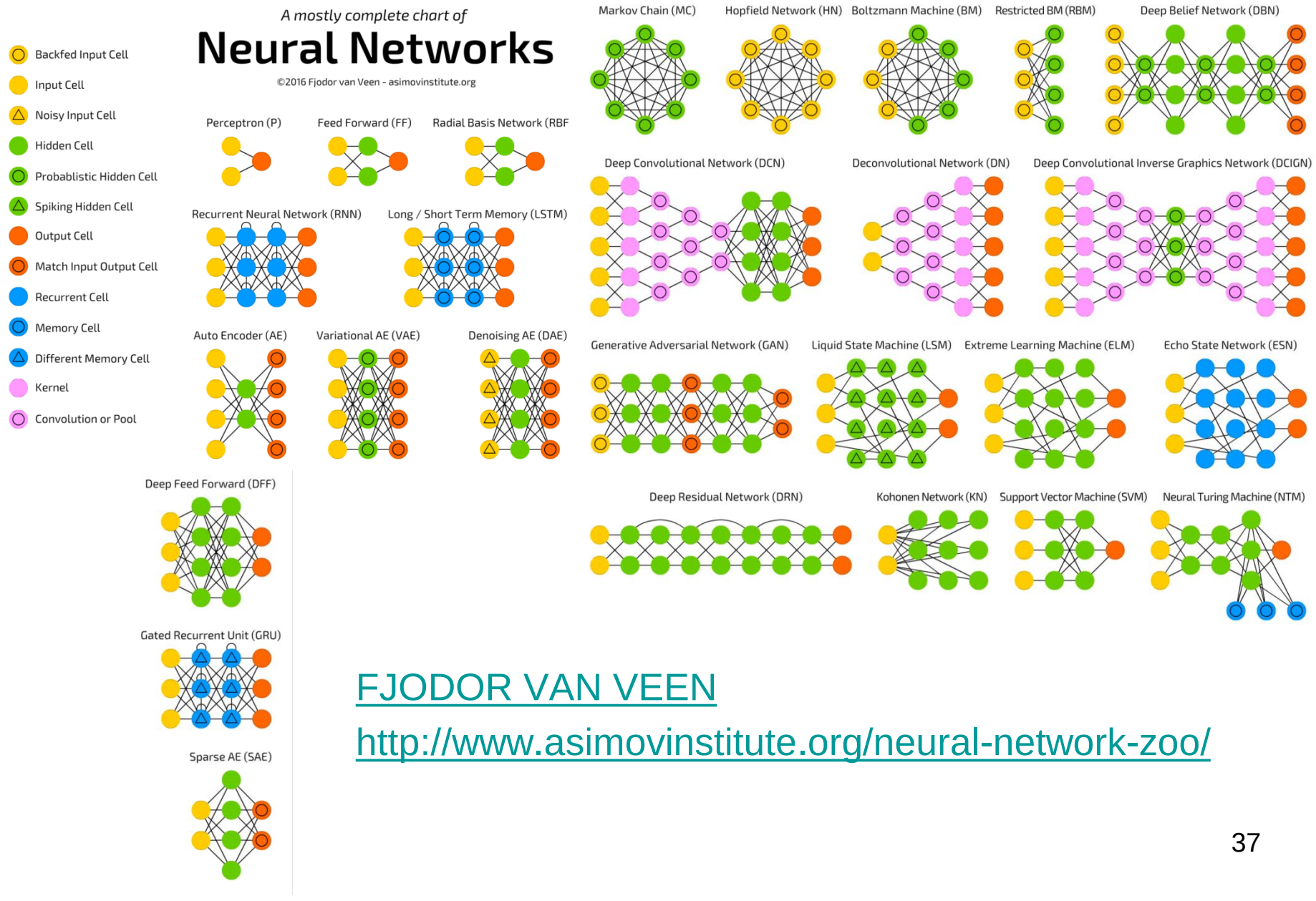
⇒ нет рефлексии и , как следствие, объяснения

⇒ НС не приближают нас к пониманию того, как устроен человек и как строить «умные машины»



Нейробиология честнее и конструктивнее: от клеточного уровня – к функциональным системам (ТФС)

THE NEURAL NETWORK ZOO



FJODOR VAN VEEN

http://www.asimovinstitute.org/neural-network-zoo/