

# Нейроуправление



к.т.н., ассистент Фомин А.В.

# Биологические основы нейронных сетей



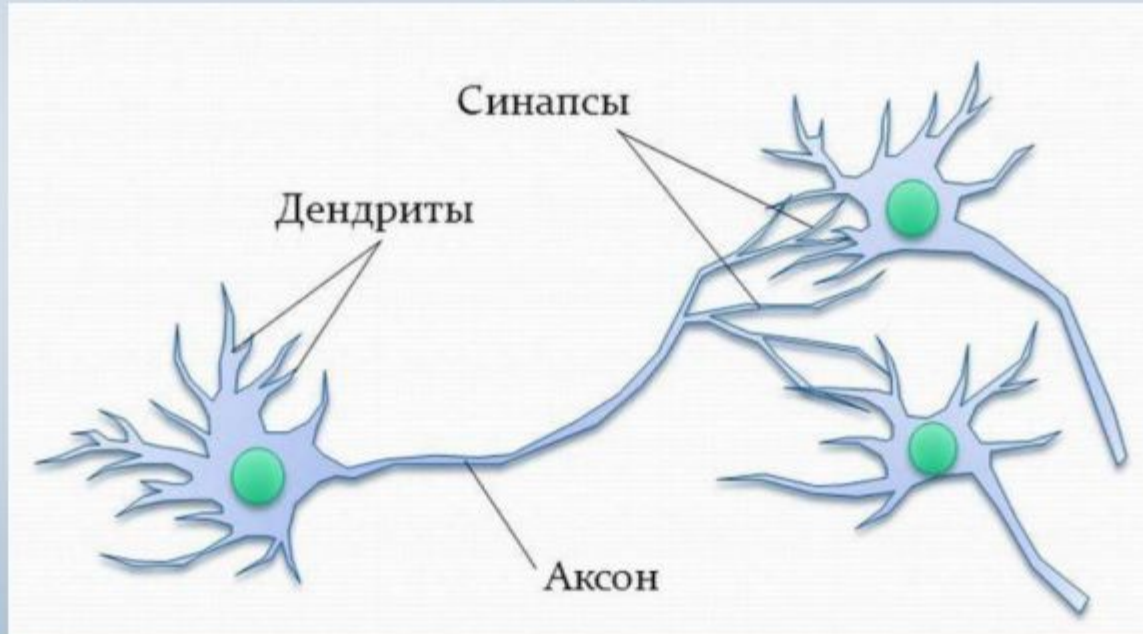
Мозг представляет собой огромную сеть, где каждым узлом является нервная клетка (нейрон).

В мозгу человека от 5 до 20 млрд. нейронов. Передача информации в мозгу, как и нервной системе в целом, осуществляется посредством нервных импульсов.



# Нейроны человеческого мозга

Передача информации в мозгу, как и нервной системе в целом, осуществляется посредством нервных импульсов. Они распространяются в направлении от тела клетки к концевому отделу аксона, который может ветвиться, образуя множество окончаний, контактирующих с другими нейронами через узкую щель – синапс



Важно, что нейроны работают не сами по себе, а иерархически объединяются. Сначала – в локальные нейронные контуры (или локальные сети), которые состоят из нейронов, выполняющих одну функцию. Локальные контуры объединяются в отделы, которые выполняют совокупность схожих функций.

# Искусственный нейрон

1. Клетка - элементарный процессор, способный к простейшей обработке информации. 2. Нейрон - элемент клеточной структуры мозга. 3. Нейрон осуществляет прием и передачу информации в виде импульсов нервной активности. 4. Природа импульсов — электрохимическая.

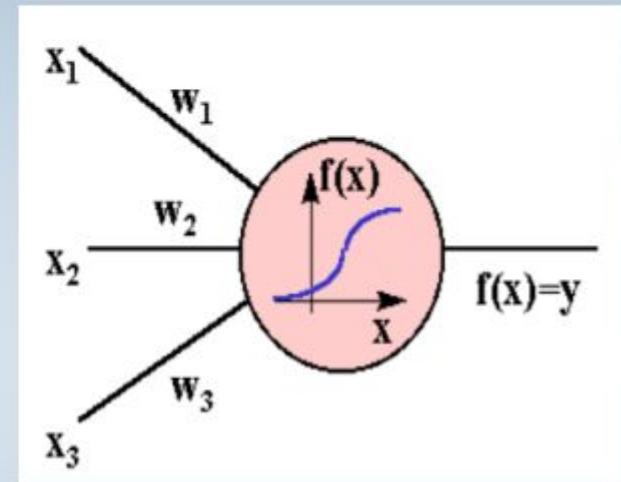
**Искусственный нейрон** — узел искусственной нейронной сети, являющийся упрощённой моделью естественного нейрона. Математически, искусственный нейрон обычно представляют как некоторую нелинейную функцию от единственного аргумента — линейной комбинации всех входных сигналов.

Данную функцию называют функцией активации или функцией срабатывания, передаточной функцией. Полученный результат посылается на единственный выход. Такие искусственные нейроны объединяют в сети — соединяют выходы одних нейронов с входами других.



# Формальное описание искусственного нейрона

- $Y = f(X) \equiv \varphi(X \cdot W)$
- Входной вектор
  - $X = \{1, X_1, \dots, X_N\}$
- Веса нейрона
  - $W = \{w_0, w_1, \dots, w_N\}$
- Передаточная функция  $f(x)$ 
  - Скалярная
  - Дифференцируемая, ограниченная
  - Монотонно возрастающая



# Нейронные сети

Эта функция называется *функцией активации* нейрона. Нейрон реализует скалярную функцию векторного аргумента. Математическая модель нейрона: где  $w_i$  – вес синапса,  $i = 1 \dots n$ ;  $b$  - значение смещения;  $s$  - результат суммирования;  $x_i$  - компонент входного вектора (входной сигнал),  $i = 1 \dots n$ ;  $y$  - выходной сигнал нейрона;  $n$  - число входов нейрона;  $f$  - нелинейное преобразование (функция активации). Синаптические связи с положительными весами называют *возбуждающими*, с отрицательными весами - *тормозящими*.

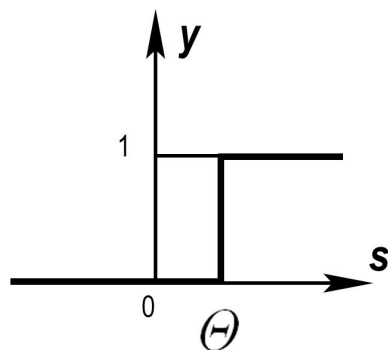
$$s = \sum_{i=1}^n w_i x_i + b,$$

$$y = f(s),$$

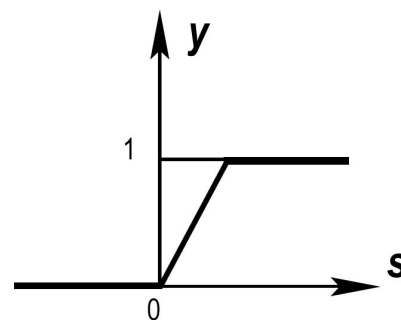
# Функции активации

Название	Формула	Область значений
Линейная	$f(s) = k s$	$(-\infty, \infty)$
Полулинейная	$f(s) = \begin{cases} k s, & s > 0, \\ 0, & s \leq 0 \end{cases}$	$(0, \infty)$
Логистическая (сигмоидальная)	$f(s) = \frac{1}{1 + e^{-as}}$	$(0, 1)$
Гиперболический тангенс (сигмоидальная)	$f(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$	$(-1, 1)$
Экспоненциальная	$f(s) = e^{-as}$	$(0, \infty)$
Синусоидальная	$f(s) = \sin(s)$	$(-1, 1)$
Сигмоидальная (рациональная)	$f(s) = \frac{s}{a +  s }$	$(-1, 1)$
Шаговая (линейная с насыщением)	$f(s) = \begin{cases} -1, & s \leq -1, \\ s, & -1 < s < 1, \\ 1, & s \geq 1 \end{cases}$	$(-1, 1)$
Пороговая	$f(s) = \begin{cases} 0, & s < 0, \\ 1, & s \geq 0 \end{cases}$	$(0, 1)$
Модульная	$f(s) =  s $	$(0, \infty)$
Знаковая (сигнатурная)	$f(s) = \begin{cases} 1, & s > 0, \\ -1, & s \leq 0 \end{cases}$	$(-1, 1)$
Квадратичная	$f(s) = s^2$	$(0, \infty)$

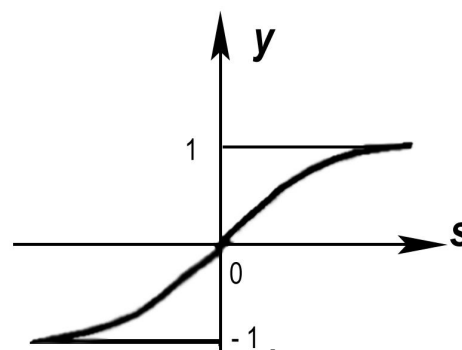
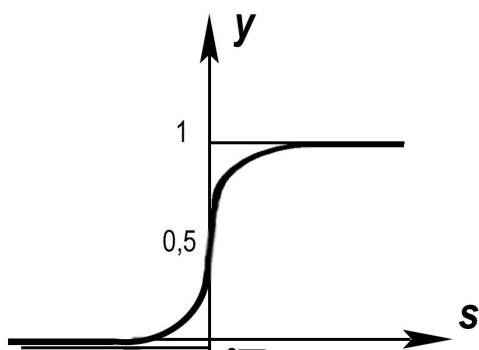
# Функции активации



а)



б)



**Рис.** Примеры активационных функций:

а - функция единичного скачка; б - линейный порог (гистерезис);

в - сигмоид (логистическая функция);

г - сигмоид (гиперболический тангенс)



# Нейронные сети

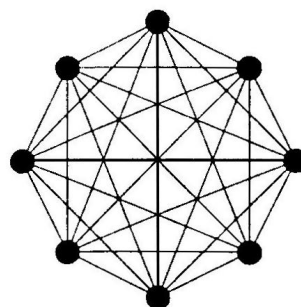
## Классификация нейронных сетей

Нейронная сеть представляет собой совокупность нейронов, определенным образом соединенных друг с другом и с внешней средой с помощью связей, определяемых весовыми коэффициентами. В зависимости от функций, выполняемых нейронами в сети, можно выделить три их типа:

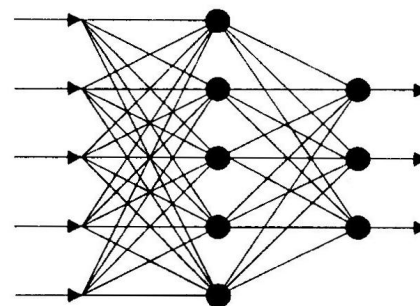
- *входные нейроны*, на которые подается вектор, кодирующий входное воздействие или образ внешней среды; в них обычно не осуществляется вычислительных процедур, а информация передается с входа на выход;
- *выходные нейроны*, выходные значения которых представляют выходы нейронной сети;
- *промежуточные нейроны*, составляющие основу нейронных сетей.

# Основные типы нейронных сетей:

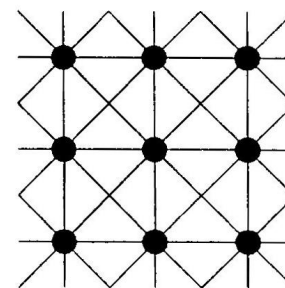
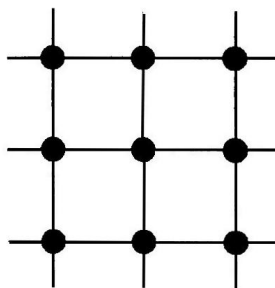
- полносвязные (рис. а);
- многослойные или слоистые (рис.б);
- слабосвязные (с локальными связями) (рис. в).



а)



б)



в)

## Полносвязанные:

- В полносвязных нейронных сетях каждый нейрон передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

# Многослойные сети :

В многослойных нейронных сетях нейроны объединяются в слои. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в слое может быть любым и не зависит от количества нейронов в других слоях. В общем случае сеть состоит из  $Q$  слоев, пронумерованных слева направо. Внешние входные сигналы подаются на входы нейронов входного слоя (его часто нумеруют как нулевой), а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько скрытых слоев. Связи от выходов нейронов некоторого слоя  $q$  к входам нейронов следующего слоя ( $q+1$ ) называются последовательными. Внутри одного слоя используется одна и та же функция активации

# Многослойная сеть

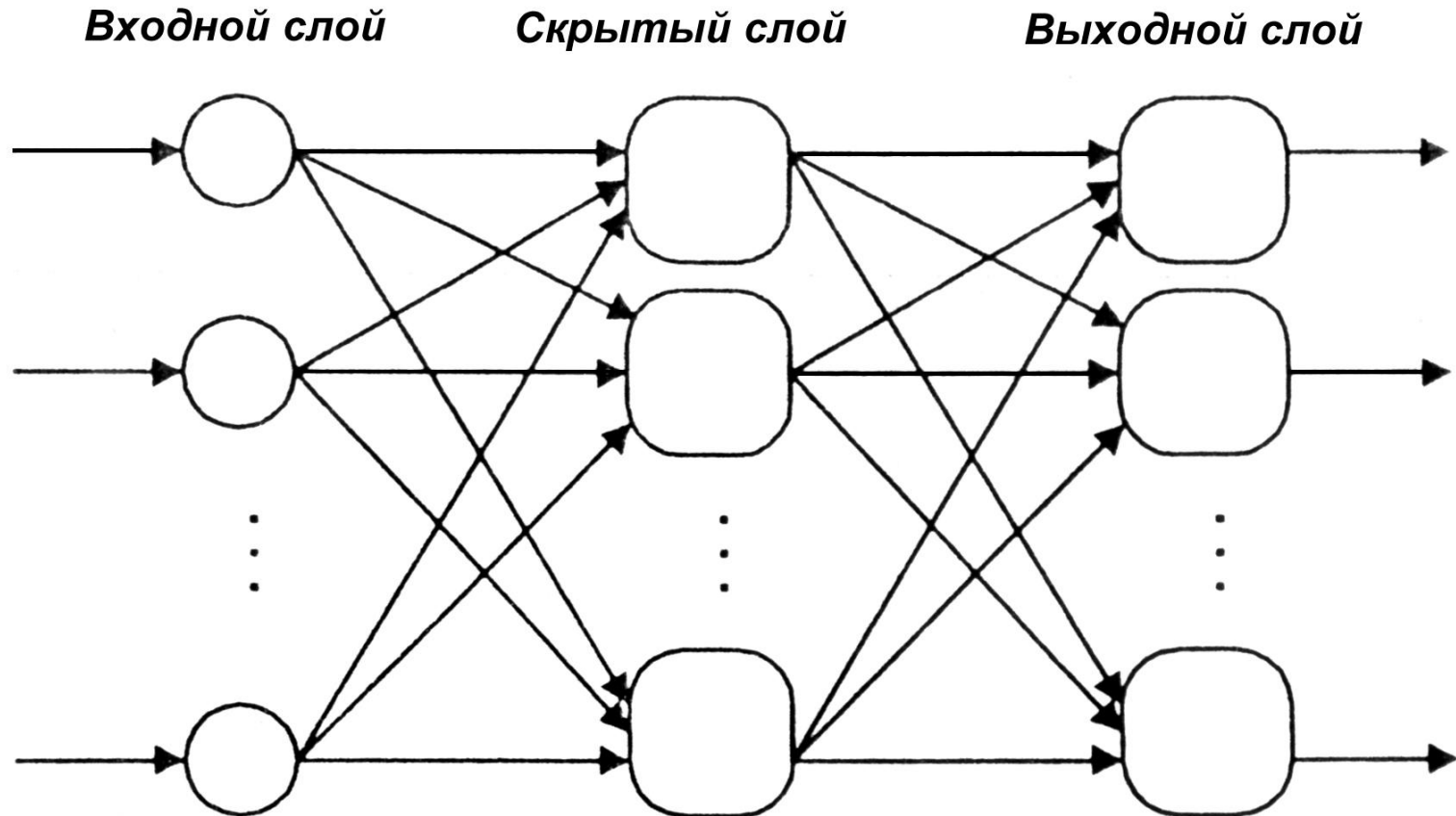


Рис. Многослойная (двухслойная) сеть прямого распространения

# Применение нейронных сетей

Многие задачи распознавания образов (зрительных, речевых), выполнения функциональных преобразований при обработке сигналов, управления, прогнозирования, идентификации сложных систем, сводятся к следующей математической постановке. Необходимо построить такое отображение  $X \rightarrow Y$ , чтобы на каждый возможный входной сигнал  $X$  формировался правильный выходной сигнал  $Y$ . Отображение задается конечным набором пар (<вход>, <известный выход>). Число этих пар (обучающих примеров) существенно меньше общего числа возможных сочетаний значений входных и выходных сигналов. Совокупность всех обучающих примеров носит название обучающей выборки.

# Применение нейронных сетей

- Управление технологическими процессами
- Распознавание образов
- Обработка естественного языка
- Машинный перевод
- Нелинейное управление и робототехника
- Машинное зрение, виртуальная реальность и обработка изображений
- Теория игр и стратегическое планирование
- Диагностика
- ИИ в играх и боты в компьютерных играх
- Машинное творчество
- Сетевая безопасность

# Обучение нейронной сети

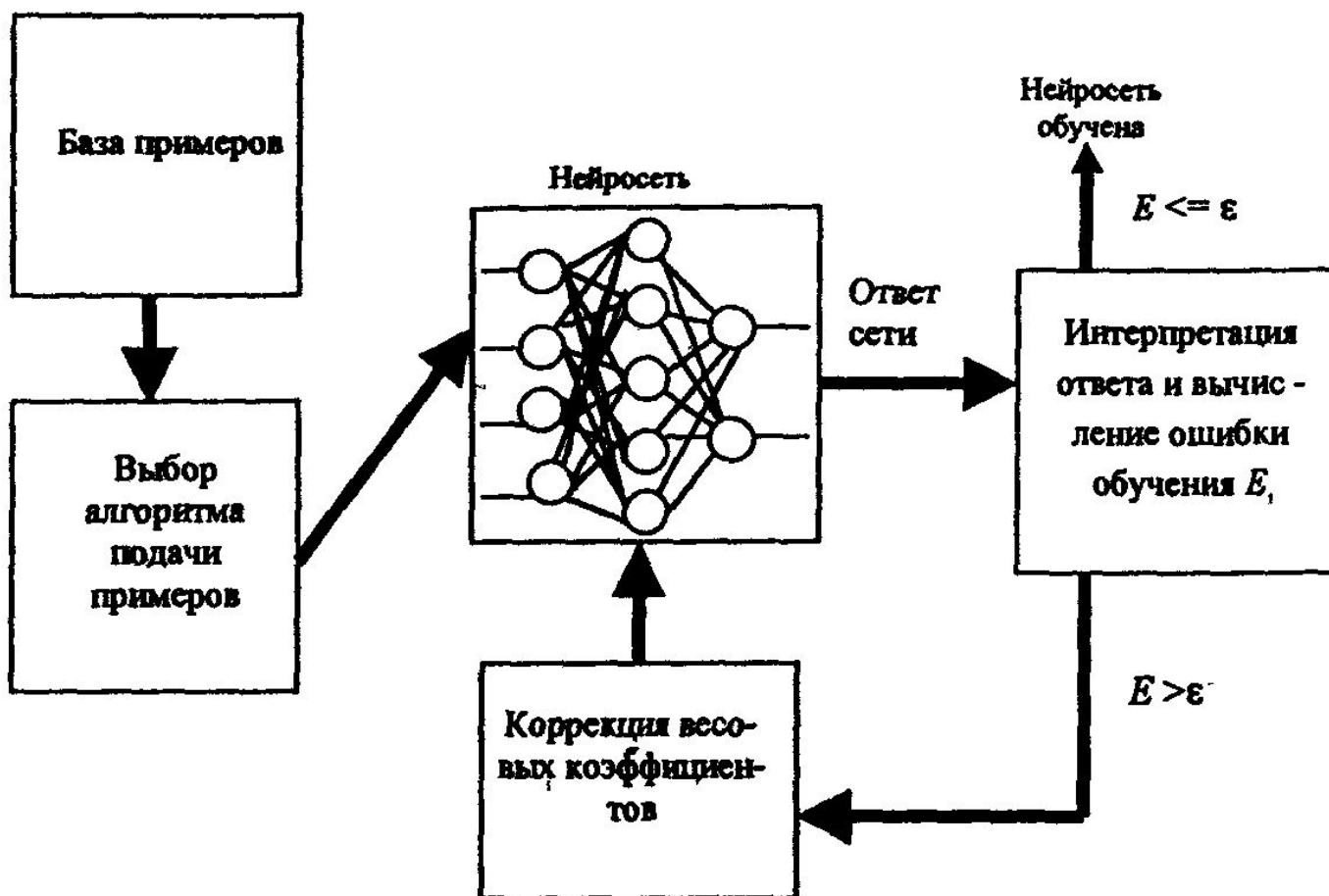
Способность к обучению является фундаментальным свойством мозга. В контексте ИНС процесс обучения может рассматриваться как настройка архитектуры сети и весов связей для эффективного выполнения специальной задачи. Обычно нейронная сеть должна настроить веса связей по имеющейся обучающей выборке. Функционирование сети улучшается по мере итеративной настройки весовых коэффициентов

Существуют три парадигмы обучения:

- 1) **"с учителем"**: нейронная сеть располагает правильными ответами (выходами сети) на каждый входной пример. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам. Усиленный вариант обучения с учителем предполагает, что известна только критическая оценка правильности выхода нейронной сети, но не сами правильные значения выхода.
- 2) **"без учителя"** (самообучение) - не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или корреляции между образцами в системе данных, что позволяет распределить образцы по категориям
- 3) **смешанная**: часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.



# Обучение нейронной сети



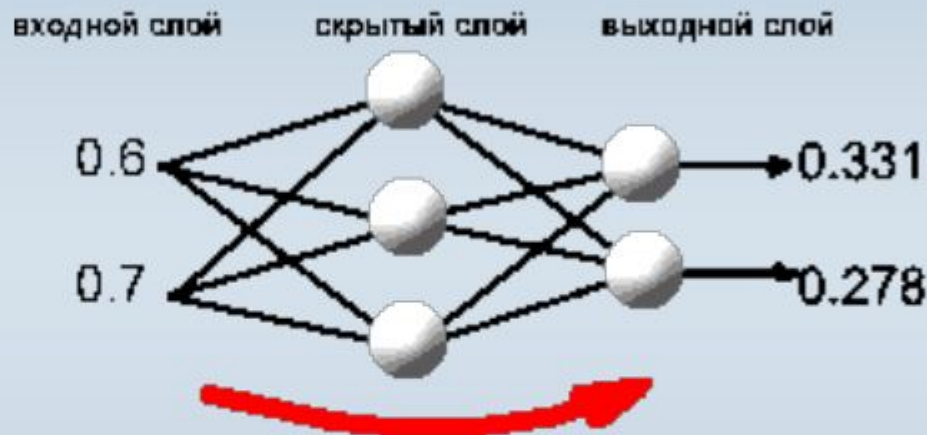
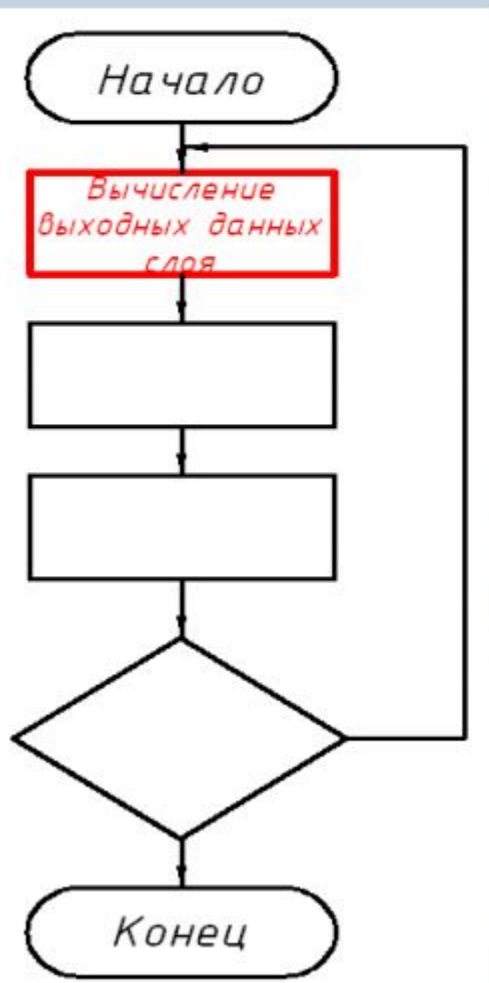
# Обучение нейронной сети

Вначале определенным образом устанавливаются значения весовых коэффициентов межнейронных связей. Затем из базы данных в соответствии с некоторым правилом поочередно выбираются примеры (пары обучающей выборки  $X_i, Y_i$  : входной вектор  $X_i$  подается на вход сети, а желаемый результат  $Y_i$  на выход сети). Если ошибка велика, то осуществляется подстройка весовых коэффициентов для ее уменьшения, Это и есть процедура обучения сети. В стандартной ситуации описанный процесс повторяется до тех пор, пока ошибка не станет меньше заданной, либо закончится время обучения.

Простейший способ обучения НС - по очереди менять каждый весовой коэффициент сети таким образом, чтобы минимизировалась ошибка сети. Этот способ является малоэффективным, целесообразнее вычислить совокупность производных ошибки сети по весовым коэффициентам - градиент ошибки по весам связей и изменить все веса сразу на величину, пропорциональную соответствующей производной. Один из возможных методов, позволяющих вычислить градиент ошибки, алгоритм обратного распространения - наиболее известен в процедурах обучения НС.

# Алгоритм обучения нейронных сетей

## Шаг 1

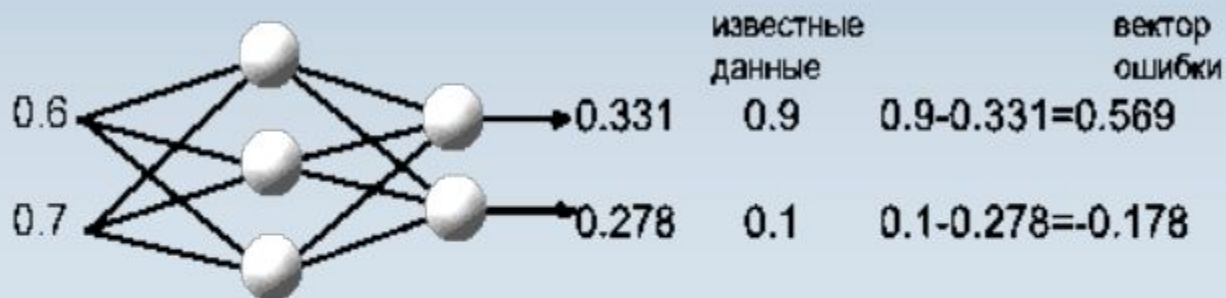
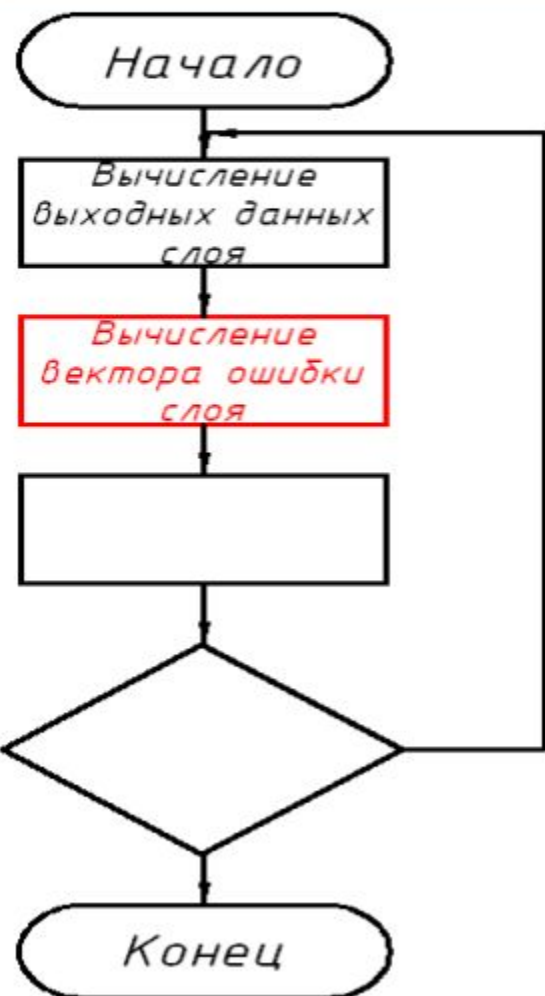


На начальном этапе величины матрицы  $W$  заданы произвольным образом. У нас есть набор тестовых объектов, и мы знаем какие выходные данные (то есть правильный ответ) соответствуют нашим исходным данным.

1. На первом этапе мы подаем последовательность наших данных и рассчитываем что получается на выходе при начальных произвольных величинах матрицы  $W$ .

# Алгоритм обучения нейронных сетей

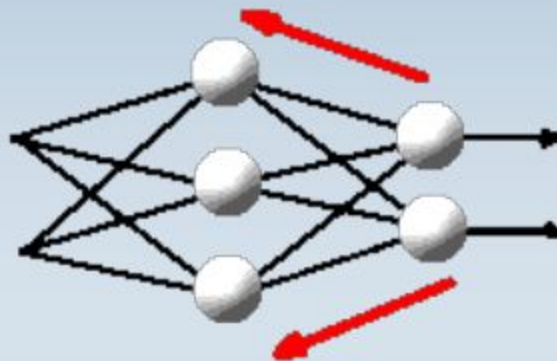
## Шаг 2



На втором этапе определяем разницу между правильными ответами и теми что получились при использовании произвольной матрицы  $W$ . То есть строим вектор ошибки.

# Алгоритм обучения нейронных сетей

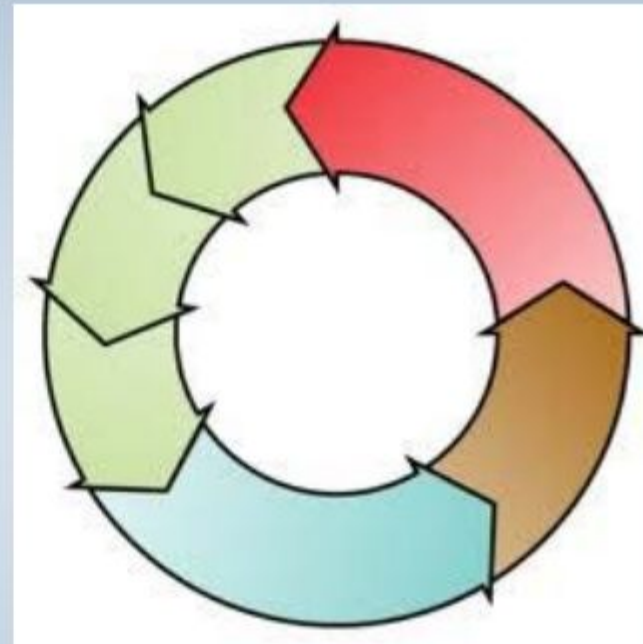
## Шаг 3



На данном этапе модифицируем наши веса, таким образом, что бы минимизировать наш вектор ошибки. То есть изменим наши веса, и снова подадим исходные данные на вход сети с уже изменёнными весам.

# Алгоритм обучения нейронных сетей

## Множество итераций

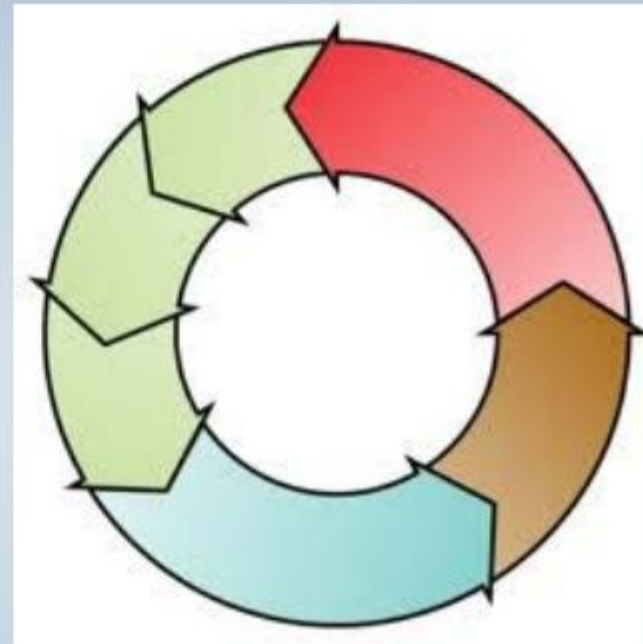


Сеть обучается путем предъявления каждого входного набора данных и последующего распространения ошибки. Цикл повторяется много раз.

**После того как сеть обучится, на вход можно подавать другие данные. Соответственно сеть будет осуществлять классификацию данных на основе полученной матрицы  $W$ .**

# Алгоритм обучения нейронных сетей

## Множество итераций



Сеть обучается путем предъявления каждого входного набора данных и последующего распространения ошибки. Цикл повторяется много раз.

**После того как сеть обучится, на вход можно подавать другие данные. Соответственно сеть будет осуществлять классификацию данных на основе полученной матрицы  $W$ .**

# Метод обратного распространения ошибки

BP – это итеративный градиентный алгоритм обучения многослойной НС без обратных связей. В такой сети на каждый нейрон первого слоя подаются все компоненты входного вектора. Все выходы скрытого слоя  $m$  подаются на слой  $m+1$  и т.д., т.е. сеть является полносвязной. При обучении ставится задача минимизации ошибки НС, которая определяется методом наименьших квадратов:

$$E(W) = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2, \quad (1)$$

где  $y_j$  - значение  $j$ -го выхода НС;

$d_j$  - желаемое значение  $j$ -го выхода;  $p$  - число нейронов в выходном слое.



# Недостатки метода ВР

**Медленная сходимость процесса обучения.** Сходимость ВР строго доказана для дифференциальных уравнений, т.е. для бесконечно малых шагов в пространстве весов. Но бесконечно малые шаги означают бесконечно большое время обучения. Следовательно, при конечных шагах сходимость алгоритма обучения не гарантируется.

**Переобучение.** Высокая точность, получаемая на обучающей выборке, может привести к неустойчивости результатов на тестовой выборке. Чем лучше сеть адаптирована к конкретным условиям (к обучающей выборке), тем меньше она способна к обобщению и экстраполяции. В этом случае сеть моделирует не функцию, а шум, присутствующий в обучающей выборке. Это явление называется переобучением. Кардинальное средство борьбы с этим недостатком - использование подтверждающей выборки примеров, которая используется для выявления переобучения сети. Ухудшение характеристик НС при работе с подтверждающей выборкой указывает на возможное переобучение

**«Ловушки», создаваемые локальными минимумами.** Детерминированный алгоритм обучения типа ВР не всегда может обнаружить глобальный минимум или выйти из него. Одним из способов, позволяющих обходить «ловушки», является расширение размерности пространства весов за счет