Лекция 9

Персептроны и зарождение искусственных нейронных сетей. Обучение персептрона.

Автор: д.т.н Джомартова Ш.А.

А если данные не структурированы?

Основные виды машинного обучения



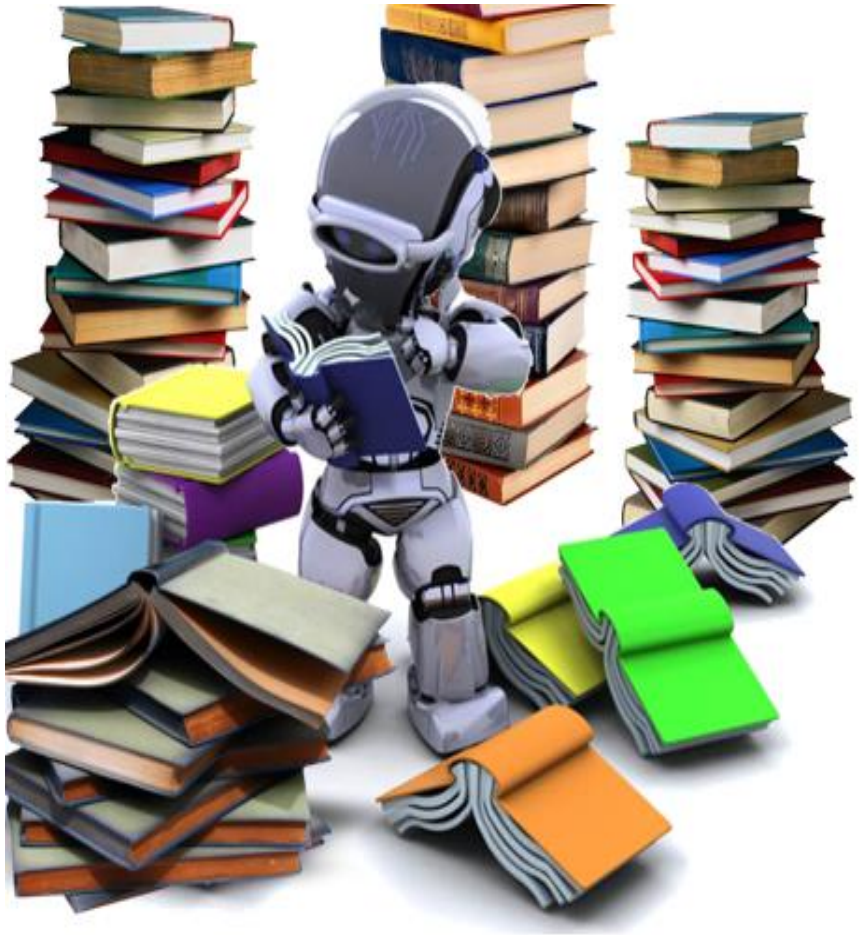

Машинное обучение
Классическое обучение
Без учителя
С учителем
Уменьшение размерности (обобщение)
t-SNE
PCA
LSA
SVD
LDA
Ridge/Lasso Regression
Обучение с подкреплением
Генетический Алгоритм
A3C
SARSA
Q-Learning
Deep Q-Network (DQN)
Ансамблевые методы
Стекинг
Беггинг
Random Forest
Бустинг
AdaBoost
XGBoost
LightGBM
CatBoost
Нейросети и глубокое обучение
Convolutional Neural Networks (CNN)
DCNN
Перцептроны (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq

Что такое глубокое обучение?

(англ. Deep learning)

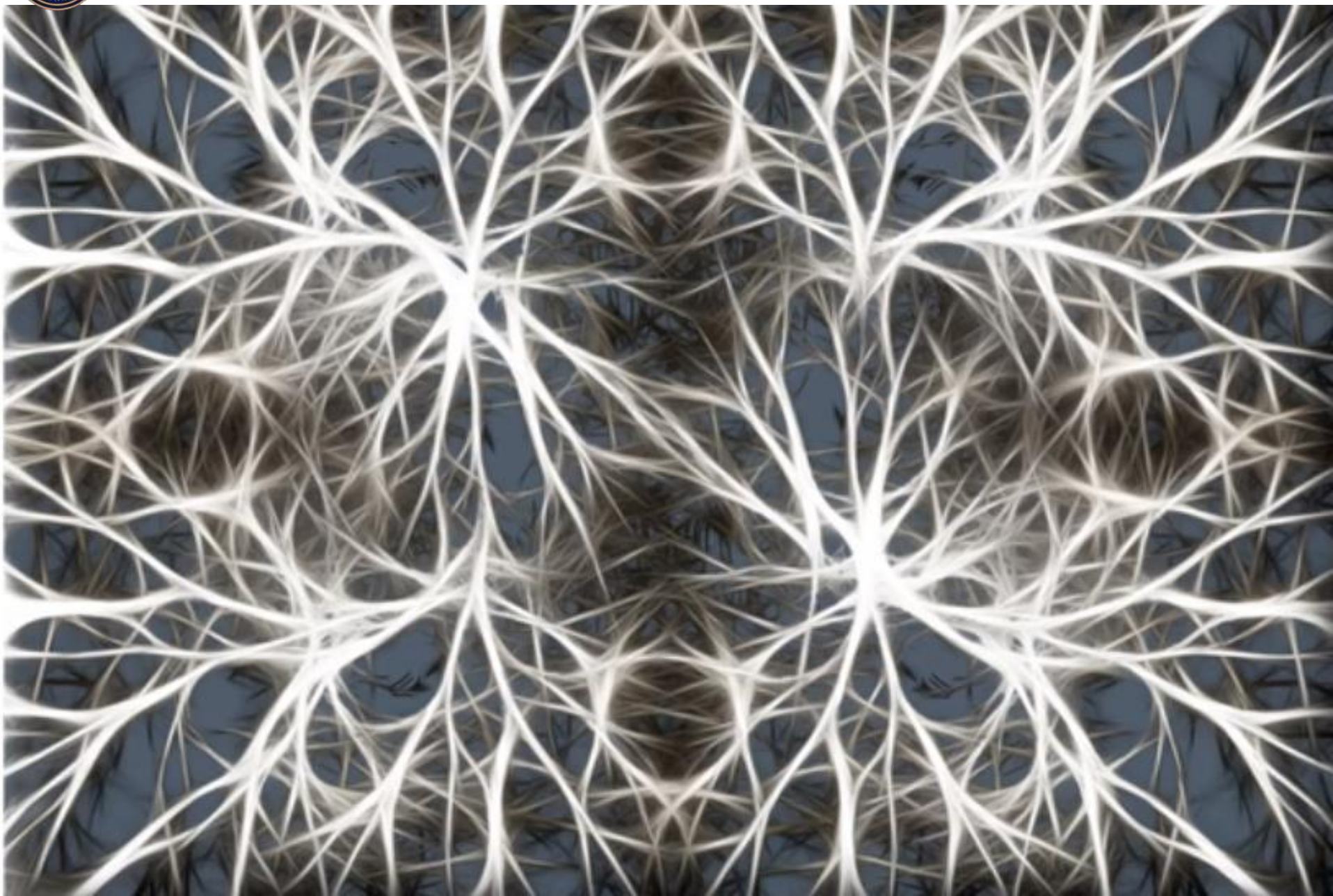
набор алгоритмов машинного обучения, основанных на изучении **множества уровней представления**.

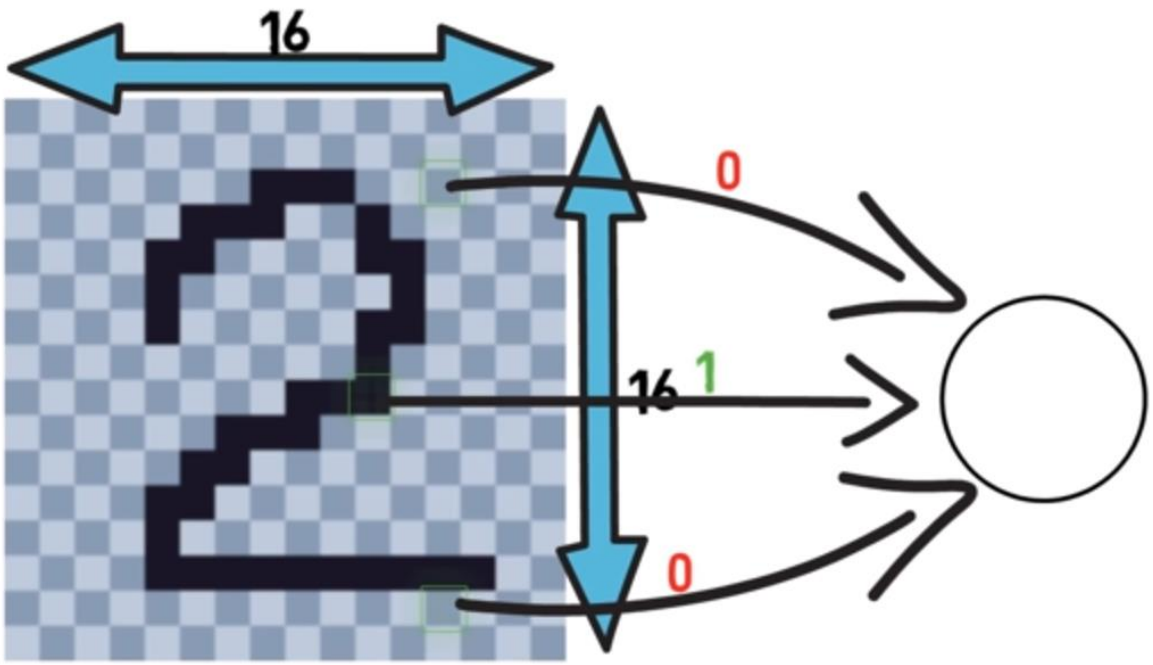
Множество уровней представления означают множество уровней абстракции.

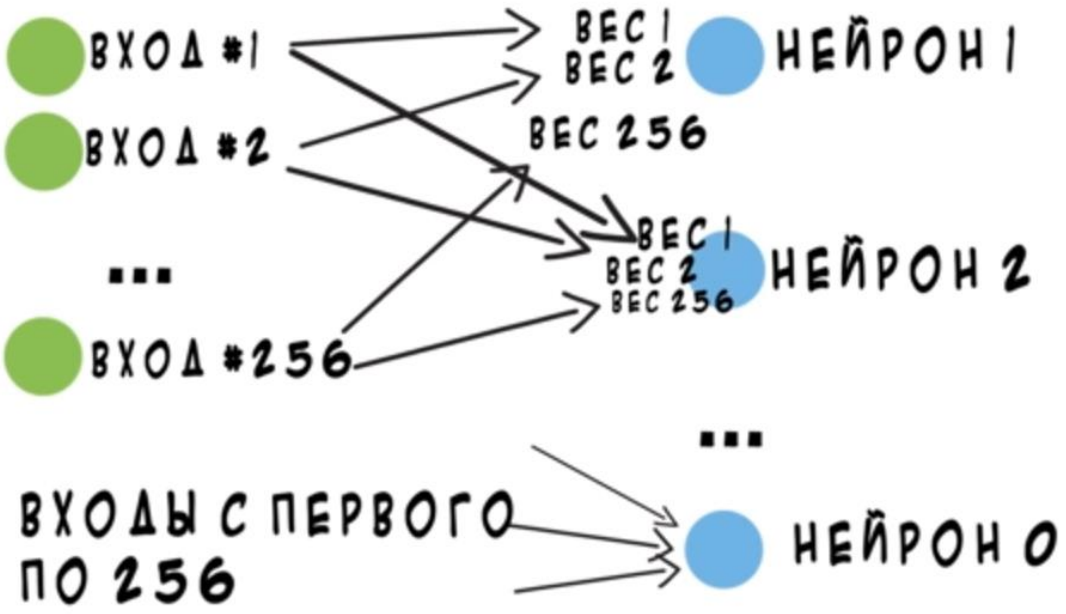


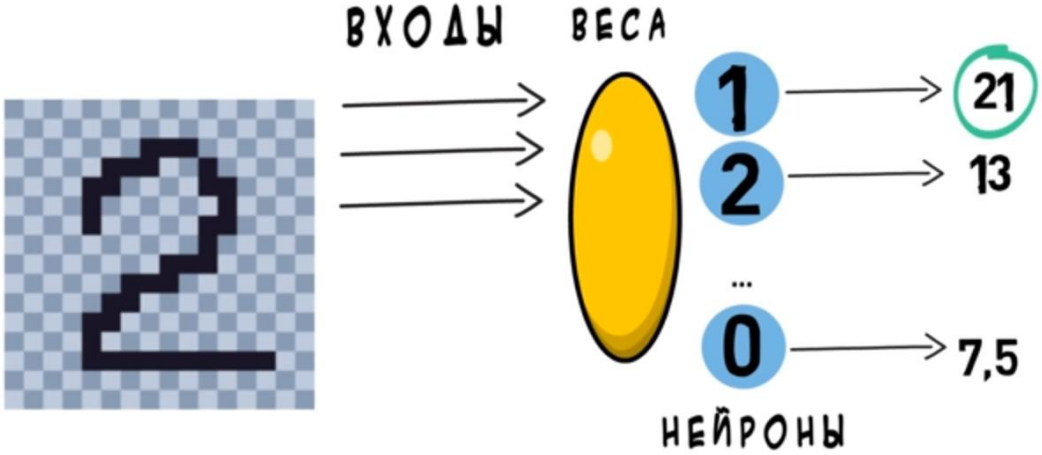
Какие задачи решает глубокое обучение?

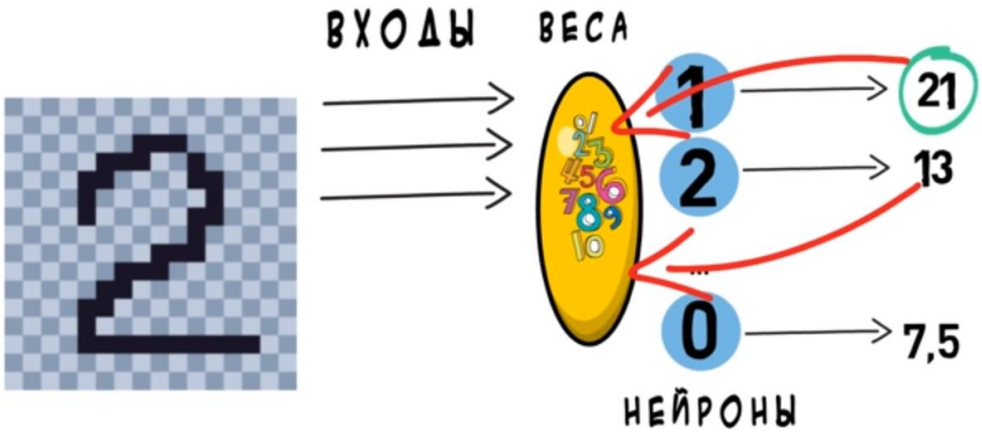
Компьютерное зрение (computer vision)
Распознавание речи (speech recognition)
Обработка естественных языков (natural language processing – NLP)
Медицинская диагностика и Биоинформатика
Финансовые приложения
Обработка текстов и документов
Защита данных и выявление мошенничества
Персональная безопасность
Рекомендации и персонализированный маркетинг
Интеллектуальные автомобили и роботы
Техническая диагностика
Информационный поиск
Интеллектуальные игры

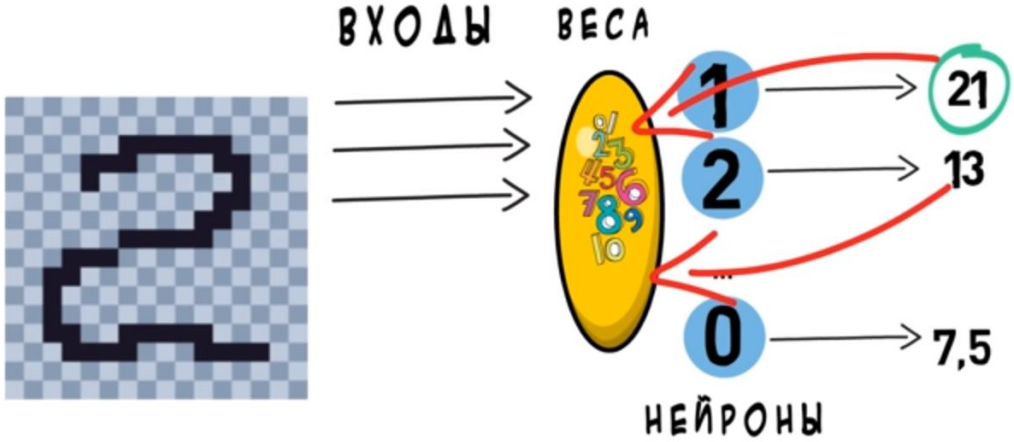



16
16
0
1
0


ВХОД #1
ВХОД #2
...
ВХОД #256
ВЕС 1
ВЕС 2
ВЕС 256
НЕЙРОН 1
ВЕС 1
ВЕС 2
ВЕС 256
НЕЙРОН 2
...
ВХОДЫ С ПЕРВОГО ПО 256
НЕЙРОН 0

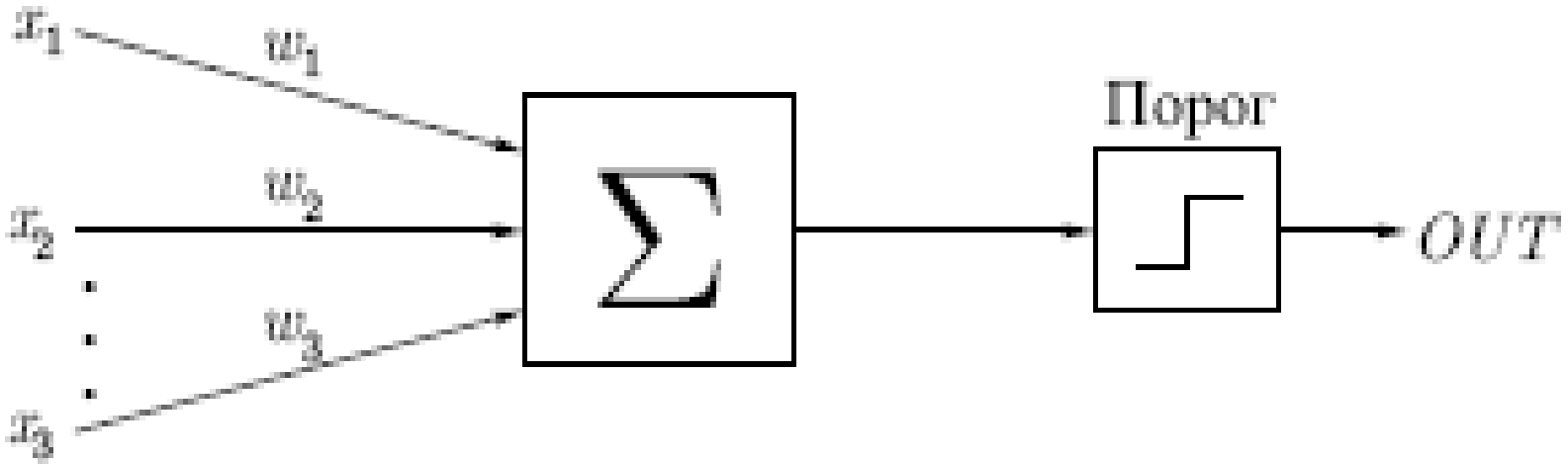

ВХОДЫ
ВЕСА
1
2
...
0
21
13
7,5
НЕЙРОНЫ


ВХОДЫ
ВЕСА
1
2
...
0
21
13
7,5
НЕЙРОНЫ


ВХОДЫ
ВЕСА
1
2
...
0
21
13
7,5
НЕЙРОНЫ

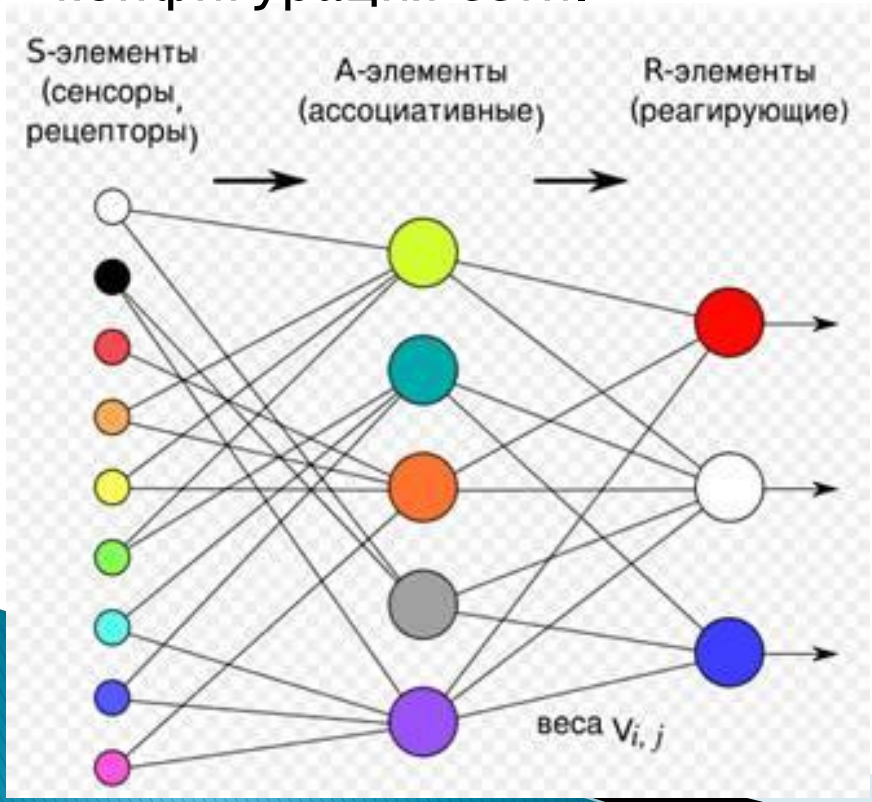
Персептрон

Перцептрон (англ. *Perceptron*) — простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов.



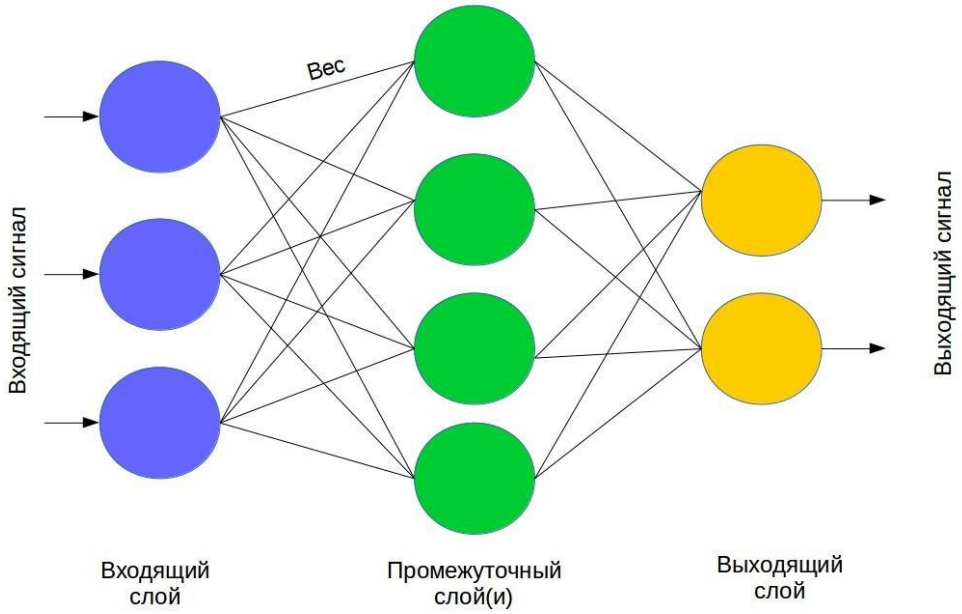
Персептрон Розенблатта

- На входе и на выходе – векторы, по вектору R решаем, что получили;
- Предназначался для распознавания образов (два класса в простейшем случае, один R-элемент);
- Функции – пороговые, значение R принадлежит [-1,1];
- «Память» сети – это веса (в общем случае, матрица весов), и конфигурация сети.





Персептрон

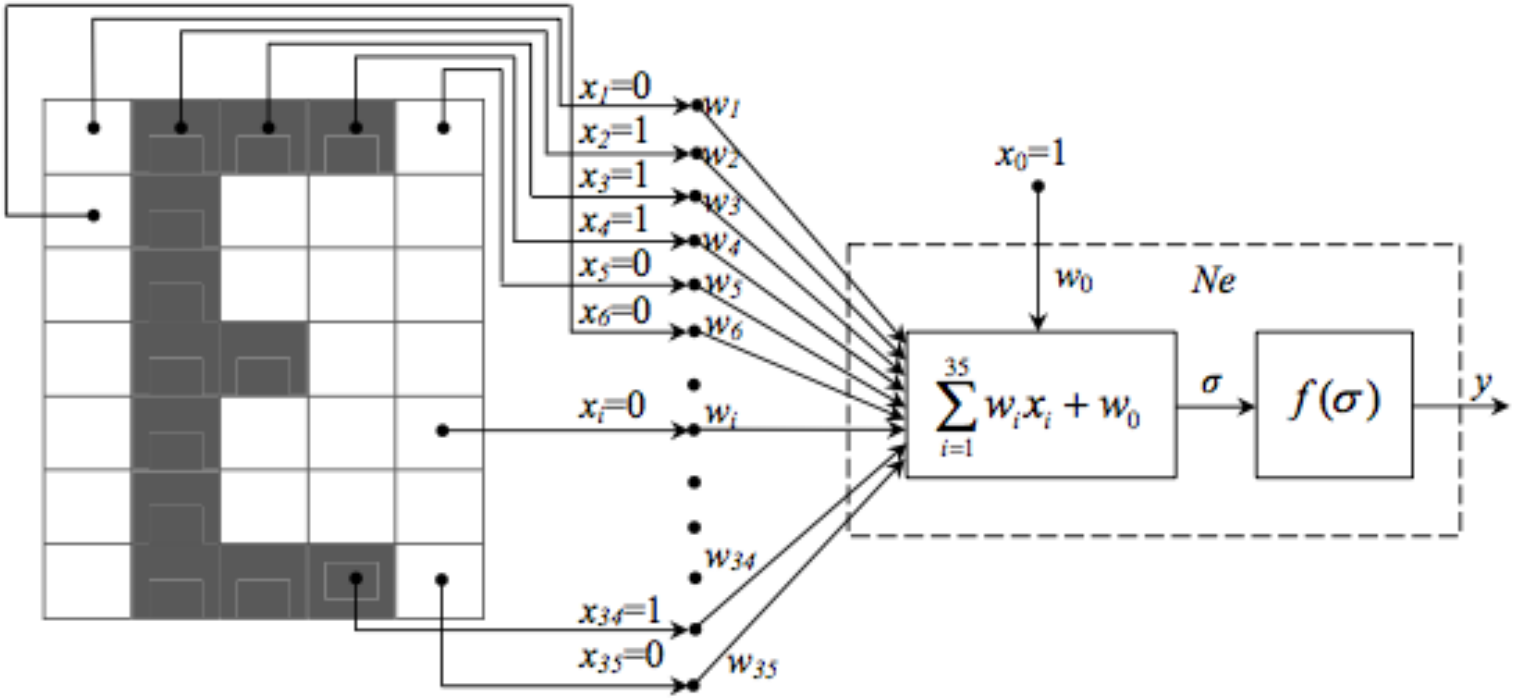


$$y = f(\sum w_2 f(\sum w_1 x))$$

Перцептрон - одна из первых моделей нейронных сетей, предложенная в 1957 году Ф. Розенблаттом как средство решения задач классификации.

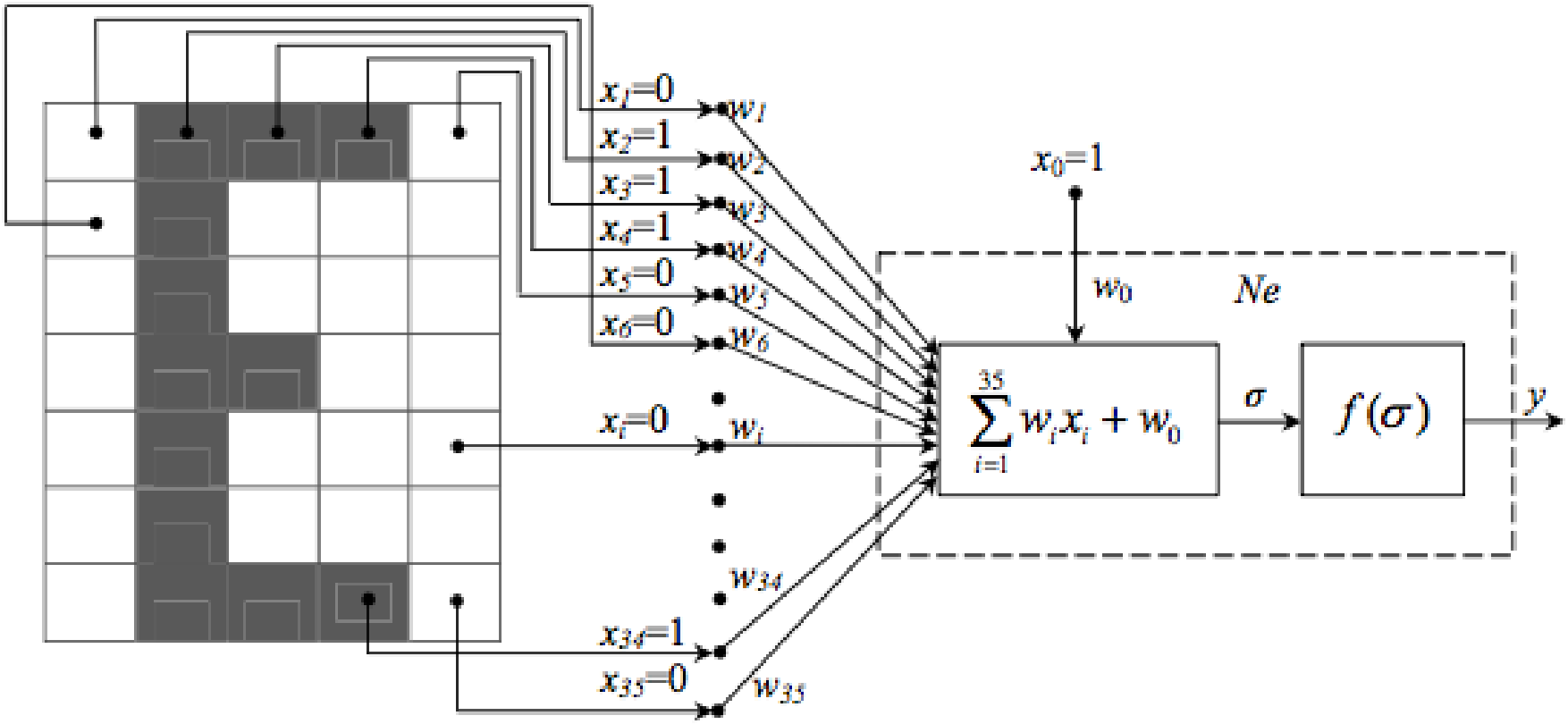
Элементарный персептрон

- Рассмотрим принцип действия персептрона на примере классификации букв русского алфавита на гласные и согласные.
- Задача персептрона заключается в том, чтобы формировать выходной сигнал *y*, равный единице, если на вход поступает гласная буква, и нулю, если согласная.



Режим обучения персептрона

- Для того, чтобы персептрон решал требуемую задачу он должен пройти режим обучения.
- Суть режима обучения заключается в настройке весов w_i и w_0 на совокупность входных образов решаемой задачи



Режим обучения персептрона

• Обучающий набор данных для персептрона должен состоять из образцов представления знаний, которым предлагается его обучить, т.е. из букв русского алфавита.

• В процессе обучения персептрону предъявляются эти буквы и анализируется его реакция *y*.

• Если, например, на вход персептрона поступает буква «А», а выходной сигнал *y* случайно оказался равным единице, означающей, что буква гласная, то корректировать веса не нужно.

• Однако если выход неправилен и *y* равен нулю, то следует увеличить веса тех активных входов, которые способствуют возбуждению персептрона.

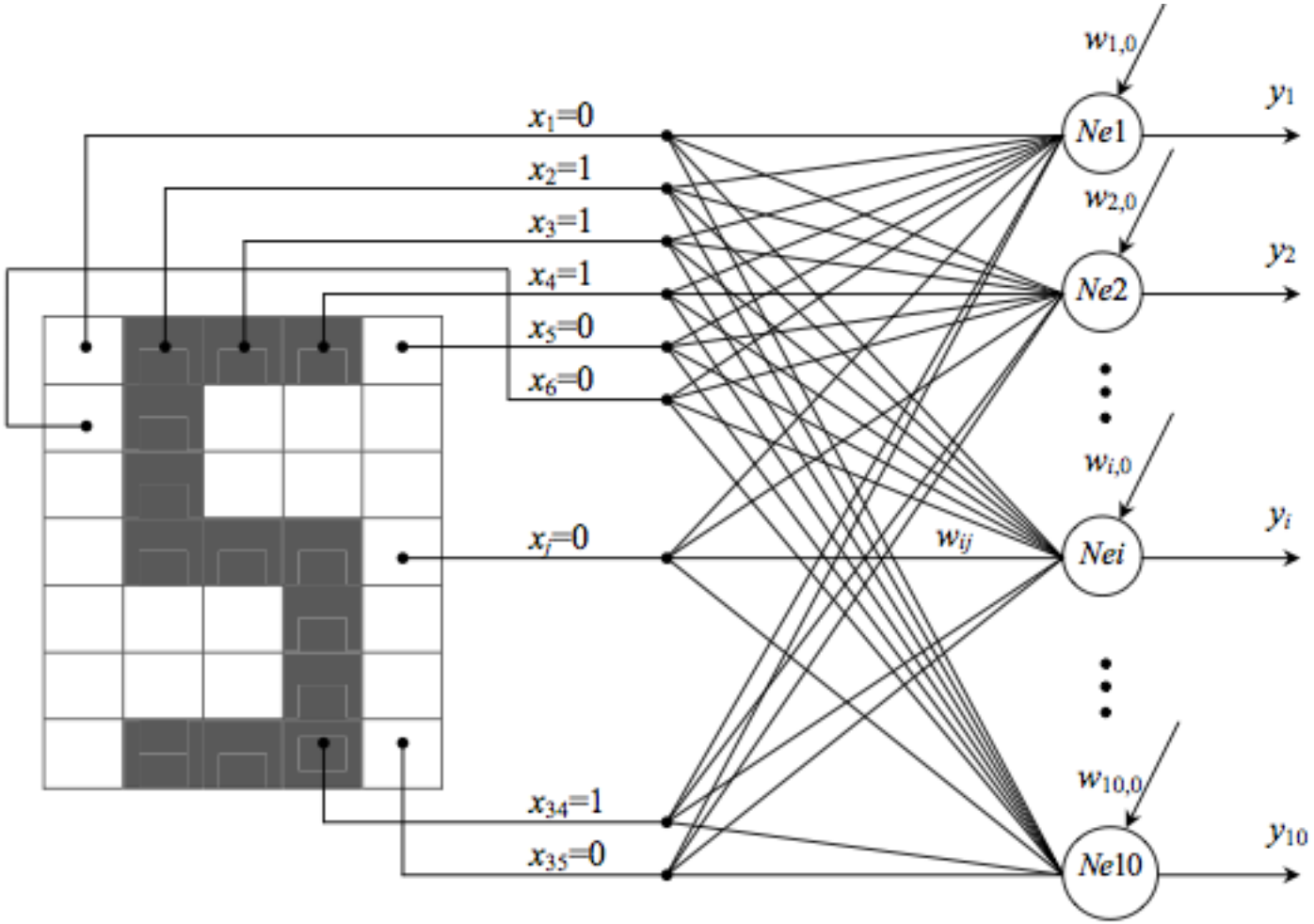
Однослойный персептрон

Развитие идеи элементарного персептрона привело к появлению однослойного персептрона и созданию алгоритма его обучения.

Данная сеть имеет 10 нейронов, организованных таким образом, чтобы каждой цифре соответствовал свой нейрон.

- Выход первого нейрона y_1 должен быть равен единице, если персептрону предъявляется цифра «1» и нулю для выходов всех остальных нейронов.
- Выход y_2 должен быть равен единице, если персептрону показывается цифра «2», при этом остальные выходы нейронов должны быть равны нулю.
- И так далее до цифры «0».

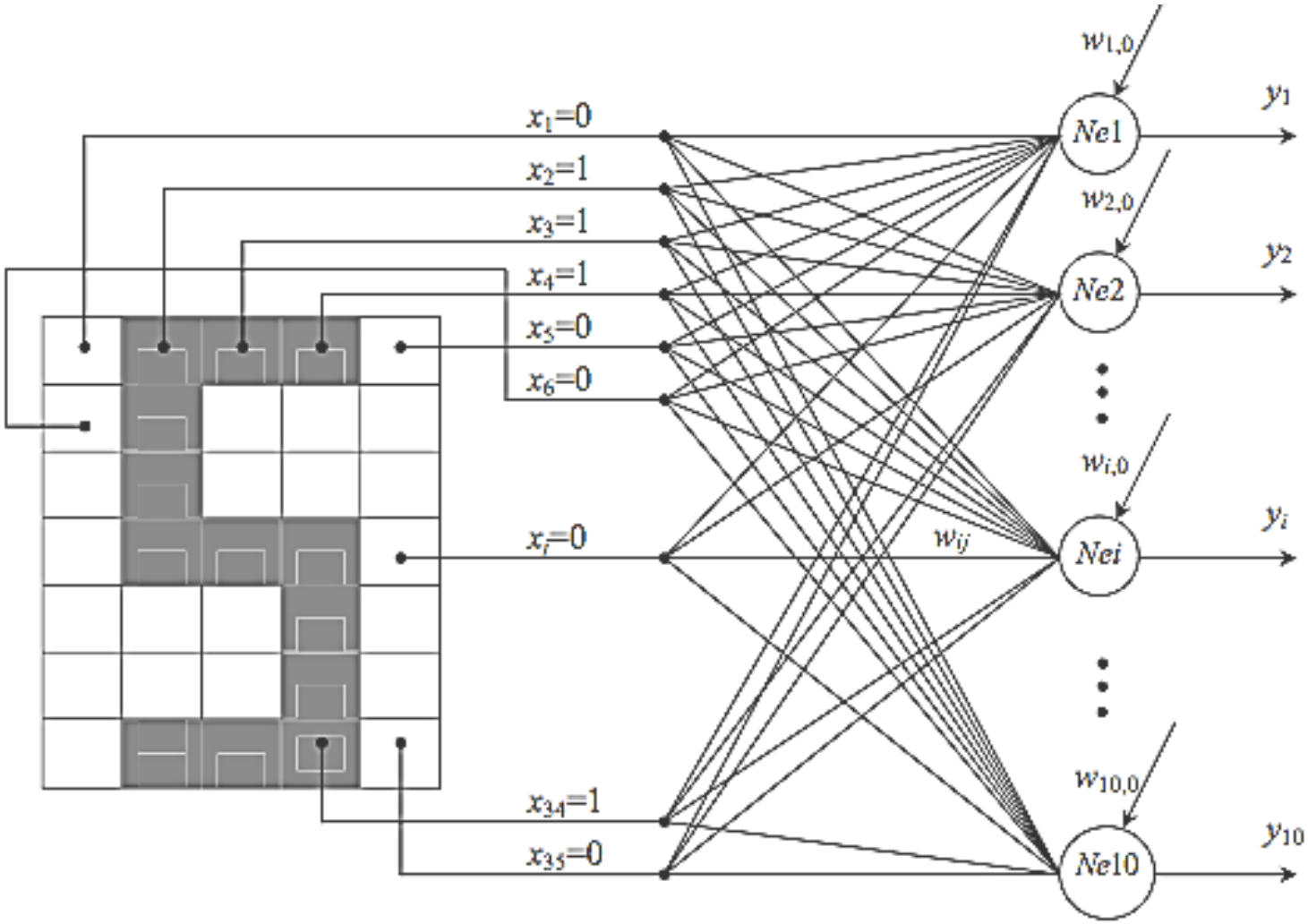
Схема однослойного персептрона, предназначенного для распознавания цифр



Алгоритм обучения

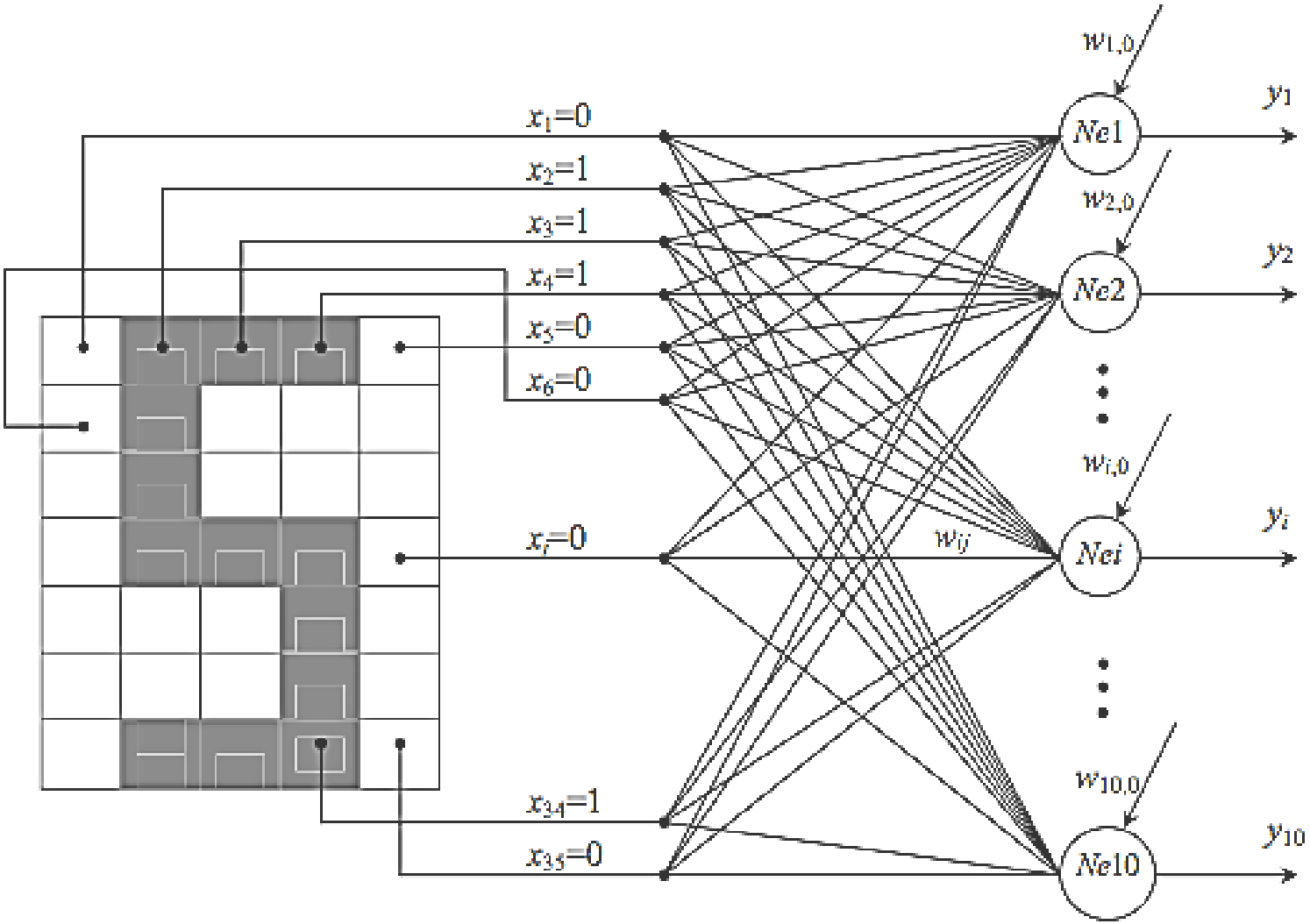
Алгоритм выглядит следующим образом:

- **Шаг 1. [Инициализация].** Всем весам персептрона w_{ij} и $w_{i,0}$ (i=1;10,j=1;35) присваиваются небольшие случайные значения из диапазона [–0,1; +0,1].



Алгоритм обучения

- **Шаг 2.** На вход персептрона подается очередной входной вектор $X[t]=\{x_1[t],x_2[t],\ldots,x_{35}[t]\}$, где t – номер итерации.



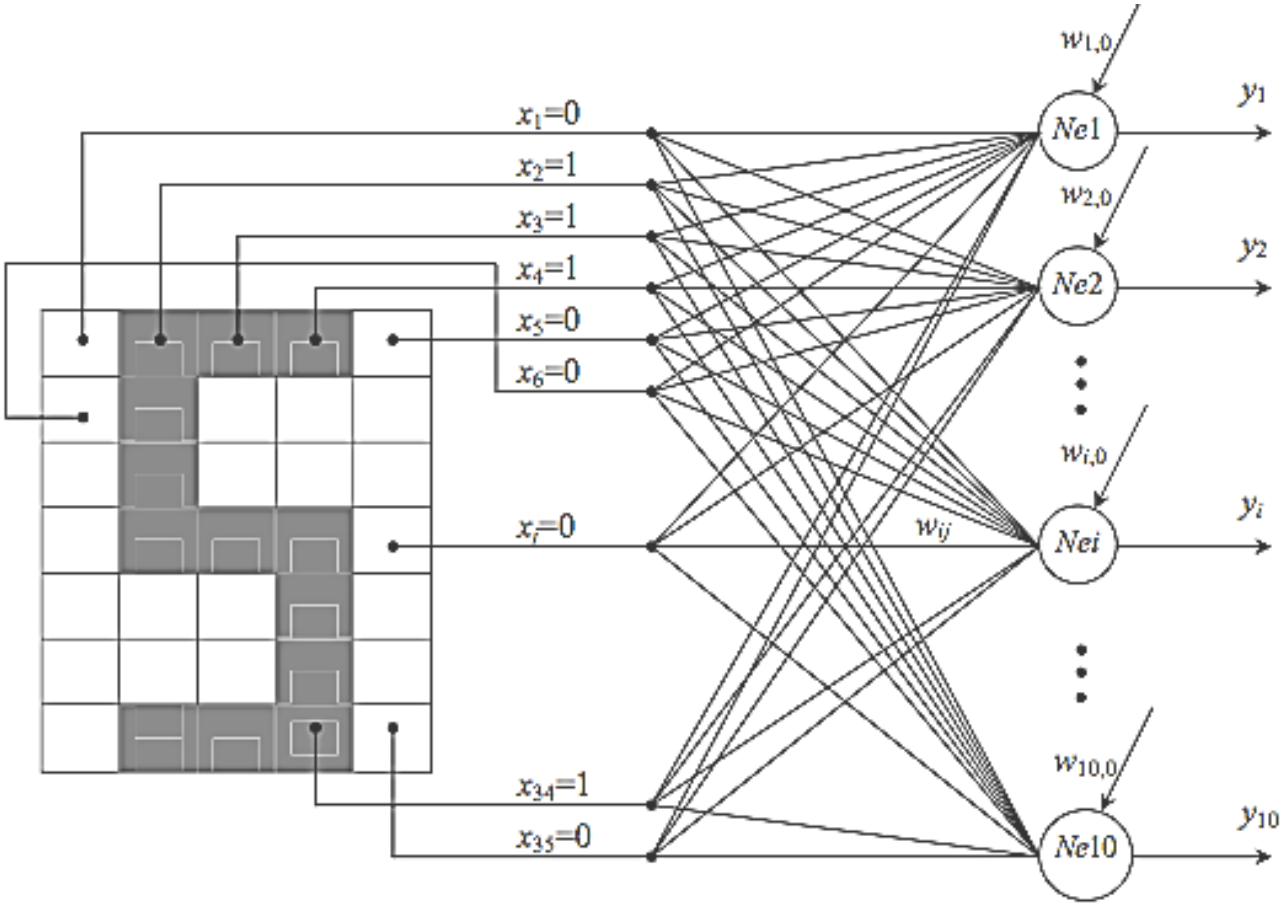
Алгоритм обучения

- Каждый из 10 нейронов выполняет взвешенное суммирование входных сигналов

$$\sigma_i[t] = \sum_{j=1}^{35} w_{ij}[t] + w_{i.0}[t]$$

и вырабатывает выходной сигнал

$$y_i[t] = \begin{cases} 1, & \text{если } \sigma_i[t] \geq 0; \\ 0, & \text{если } \sigma_i[t] < 0. \end{cases}$$



Алгоритм обучения

- **Шаг 3.** Для каждого нейрона определяется ошибка $\beta_i [t] = (d_i [t] - y_i[t])$, где $d_i[t]$ – требуемое значение выхода *i–го* нейрона, а $y_i[t]$ – полученное на шаге 2 значение *i–го* выхода.
- **Шаг 4. [Дельта–правило].** Производится модификация весовых коэффициентов персептрона в соответствии с формулами:

$$w_{ij}[t+1] = w_{ij}[t] + \Delta w_{ij}[t];$$

$$\Delta w_{ij}[t] = \alpha \cdot \beta_i[t] \cdot x_j[t];$$

$$w_{i,0}[t+1] = w_{i,0}[t] + \Delta w_{i,0}[t];$$

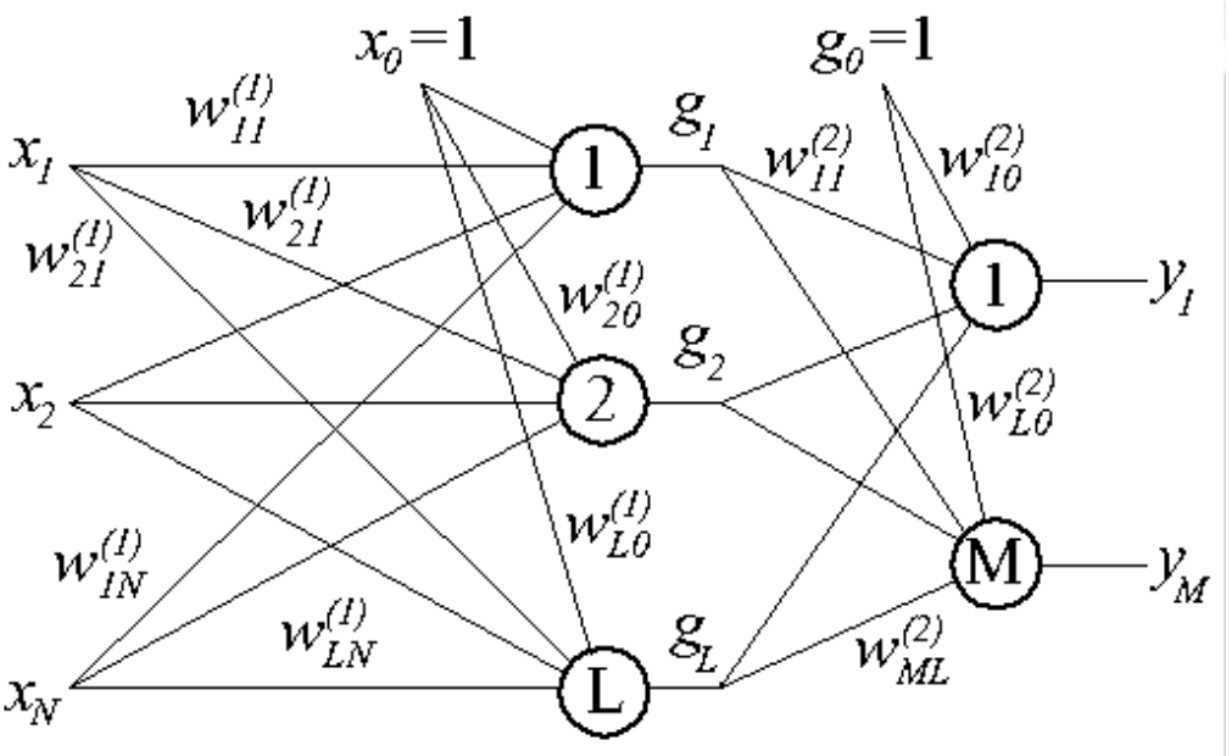
$$\Delta w_{i,0}[t] = \alpha \cdot \beta_i[t],$$

где α – коэффициент скорости обучения с помощью которого можно управлять величиной коррекции весов Δ ($0 < \alpha \leq 1$).

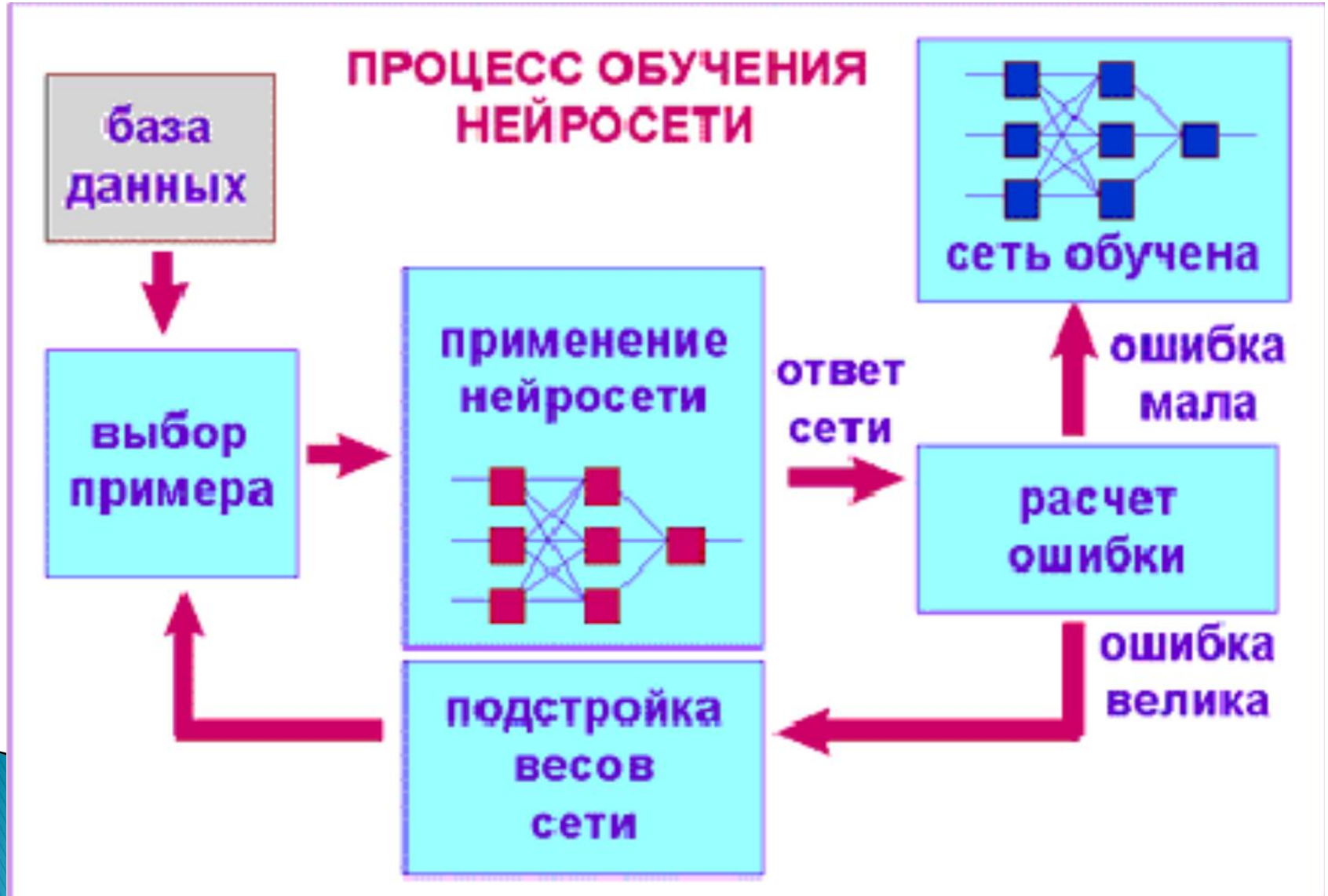
Многослойный персептрон

МП - нейронные сети прямого распространения. Входной сигнал в таких сетях распространяется в прямом направлении, от слоя к слою. Многослойный персептрон в общем представлении состоит из следующих элементов:

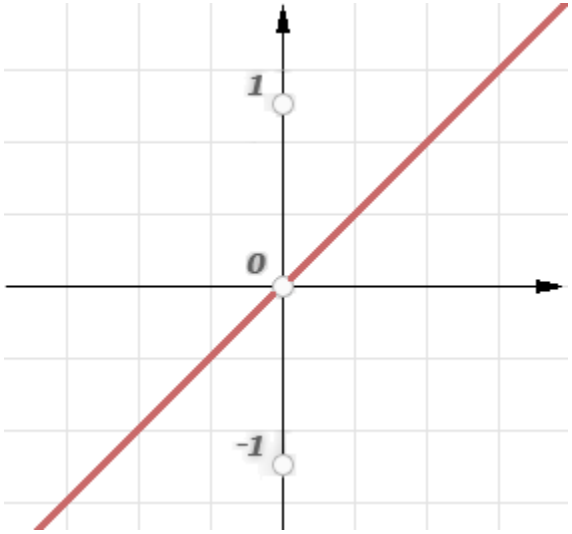
- множества входных узлов, которые образуют входной слой;
- одного или нескольких скрытых слоев вычислительных нейронов;
- одного выходного слоя нейронов.

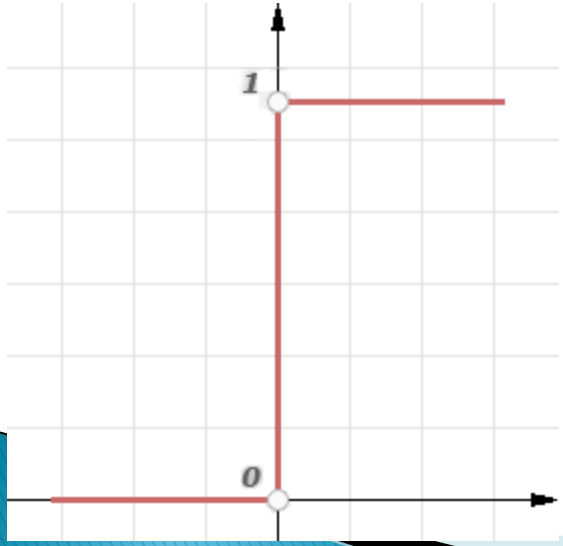


Алгоритм обучения многослойной нейронной сети методом обратного распространения ошибки



Функции активации

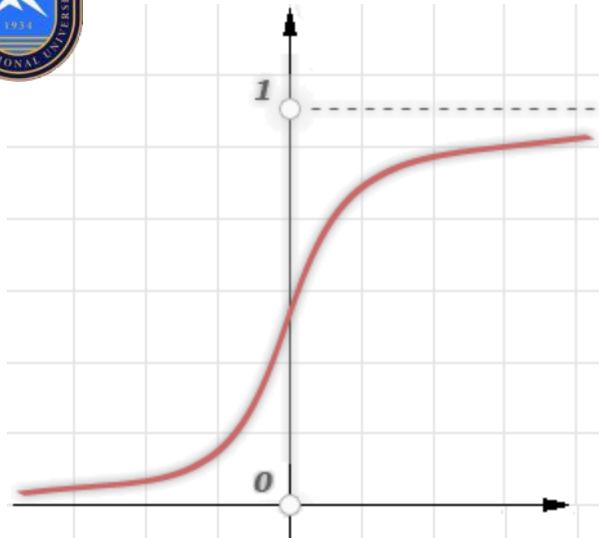




- Линейная
  - Выходы сети являются линейными комбинациями входов
- Пороговая

  Эта функция использовалась в оригинальной модели ИН.

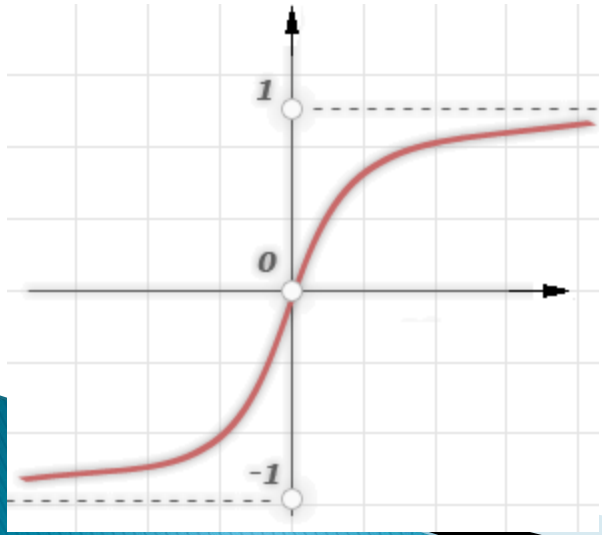
  + имеет центрированный аналог (sign x)
  - не дифференцируема



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

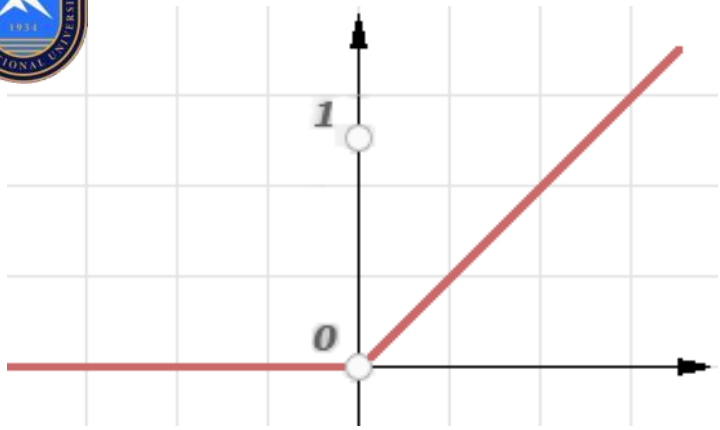
  Долгое время считалась функцией, лучше всего описывающей работу нейрона.

  + дифференцируема
  - имеет порог насыщения



- Tanh $f(x) = \tanh\left(\frac{\alpha x}{2}\right) = \frac{1-e^{-\alpha x}}{1+e^{-\alpha x}}$

  + дифференцируема
  + центрована
  - имеет порог насыщения



- ReLU (rectified linear unit)

$$f(x) = \max(0, x)$$

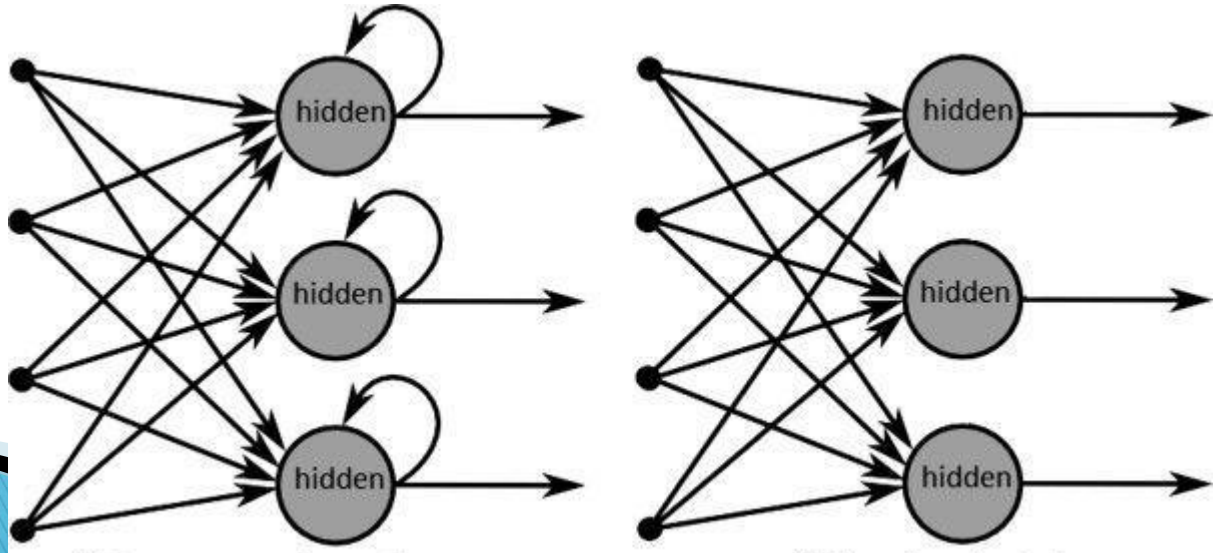
В настоящее время самая широко используемая функция активации в силу своей простоты.

Также недавние исследования показывают, что она правильнее описывает работу биологических нейронов

+ дифференцируема
+ не имеет порога насыщения
+ быстро вычисляется
- не центрована
- чувствительна к инициализации

Слои

- Слой – совокупность нейронов сети, объединяемых по особенностям их функционирования. В плоскослоистых сетях это группа нейронов, имеющих один и тот же набор входов и не соединенных между собой.
- По виду связи между слоями сети делят на
  - Сети прямого распространения (FFNN)
  - Рекуррентные сети (RNN)



(a) Recurrent neural network (b) Forward neural network

Обучение сети

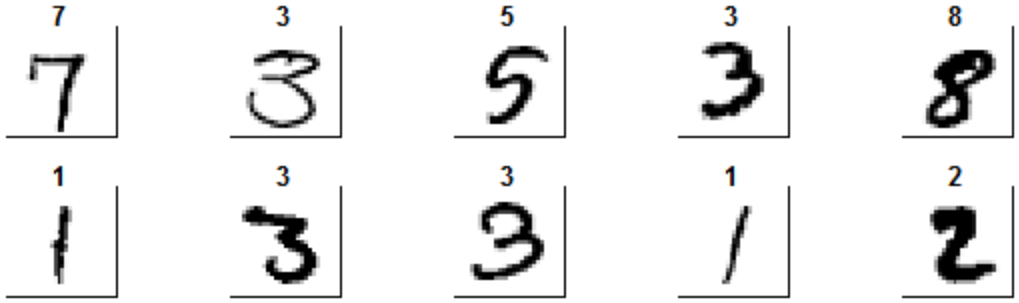
- Наиболее распространенный метод обучения нейронной сети – **метод обратного распространения ошибки**.

  Он был впервые описан в 1974 г. А.И. Галушкиным.

- Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

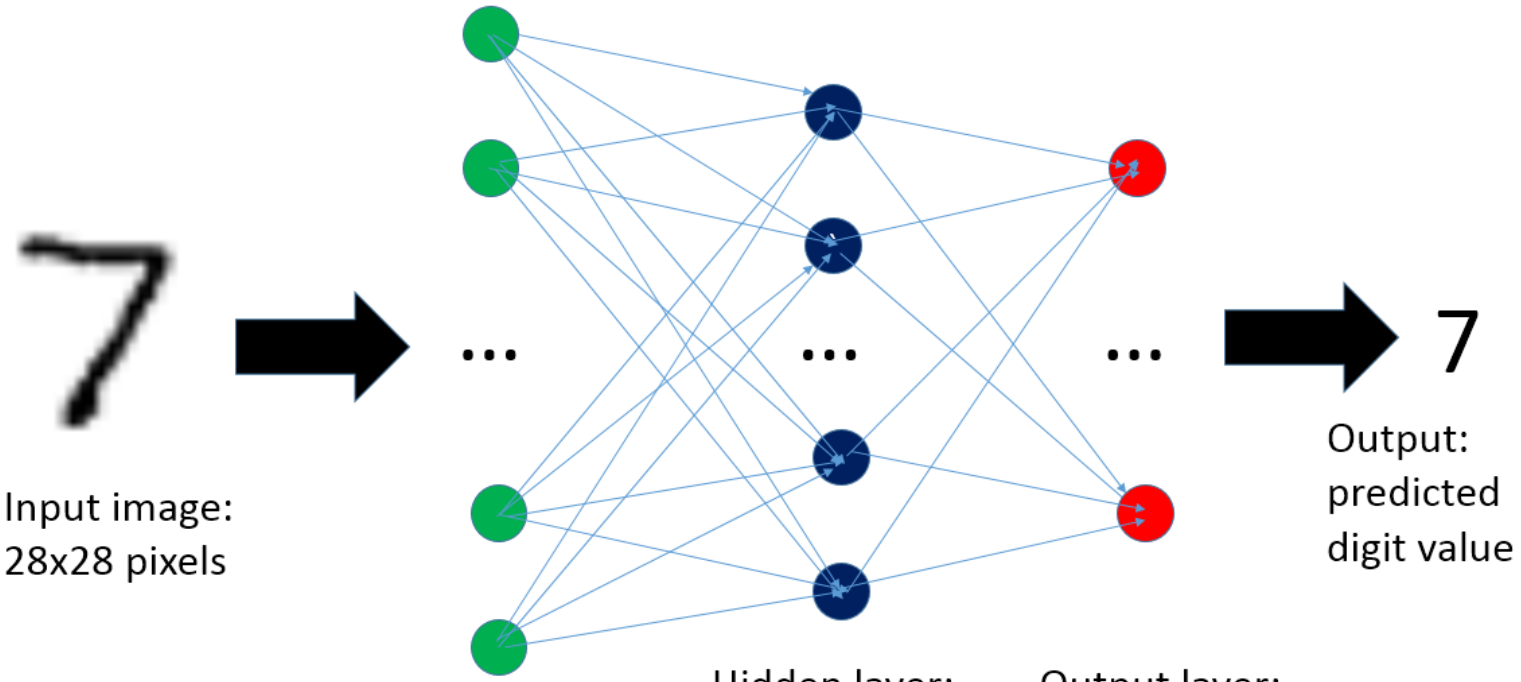
Обучающая выборка

$$D \left\{ \begin{array}{|c|c|} \hline x_1 & \tilde{y}_1 \\ \hline x_2 & \tilde{y}_2 \\ \hline \vdots & \\ \hline x_N & \tilde{y}_N \\ \hline \end{array} \right.$$



Выборка – набор размеченных входных векторов (т.е. таких, для которых известен правильный ответ), по которому производится настройка сети.

Прямой ход



$$\sum_{i=1}^{784} w_{i,n}^{(1)} x_i \rightarrow f(S_n)$$

Функция потерь

Функция потерь — функция, по значению которой можно оценить работу сети.

Две наиболее часто используемых функции потерь:

- среднеквадратичная ошибка (MSE):

$$E = \frac{1}{2}\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2$$

- логистическая (log loss):

$$E = -\frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i \cdot \log(y_i) + (1 - \tilde{y}_i) \cdot \log(1 - y_i))$$

Обратный ход

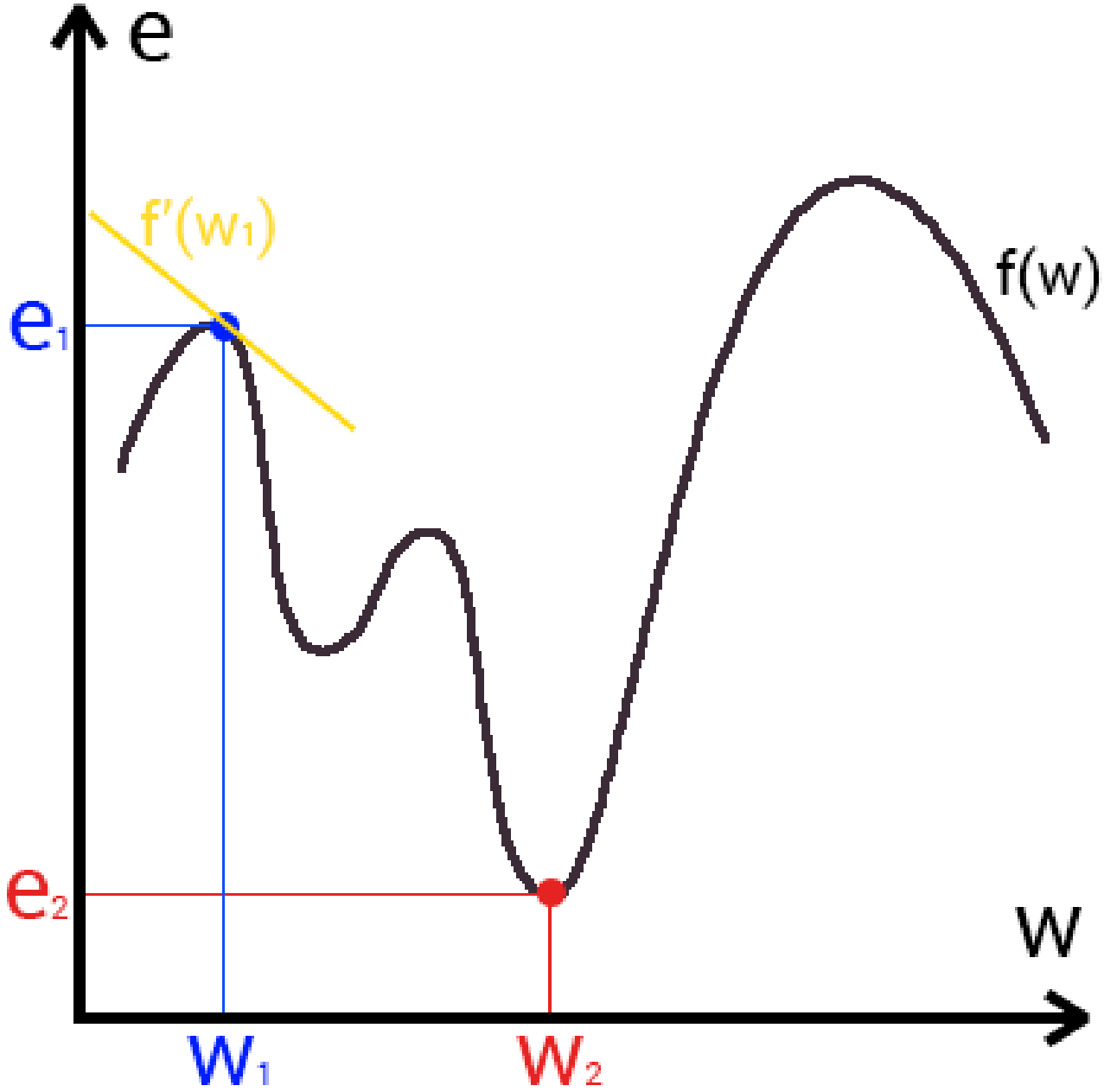
Будем минимизировать функцию потерь

$$E = \frac{1}{2}\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2$$

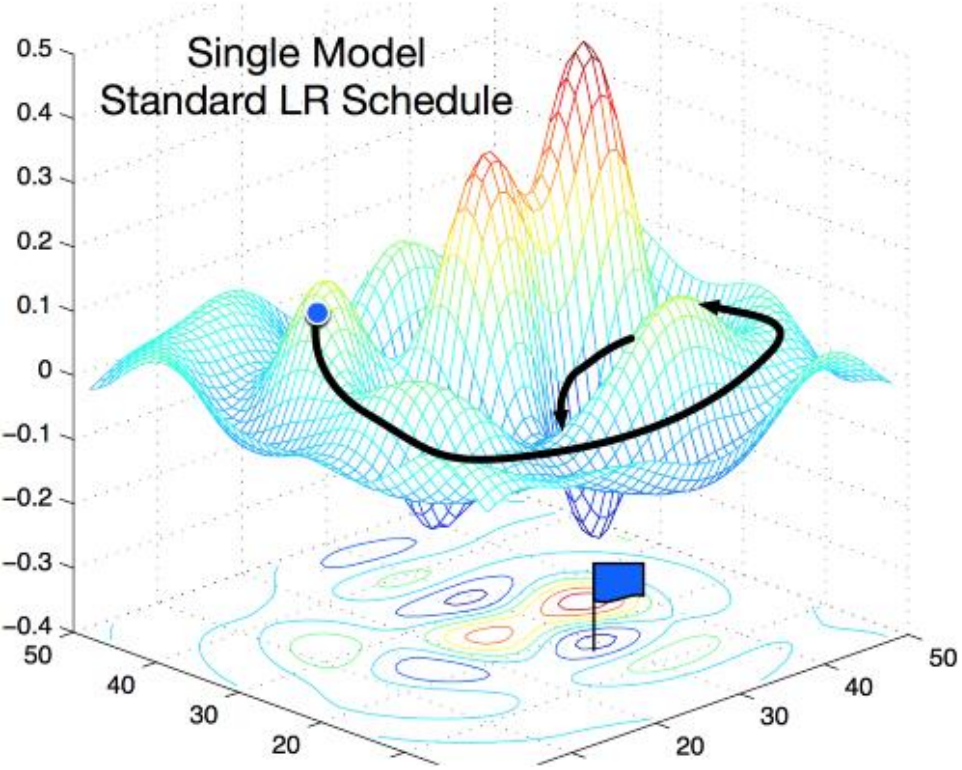
методом стохастического градиентного спуска

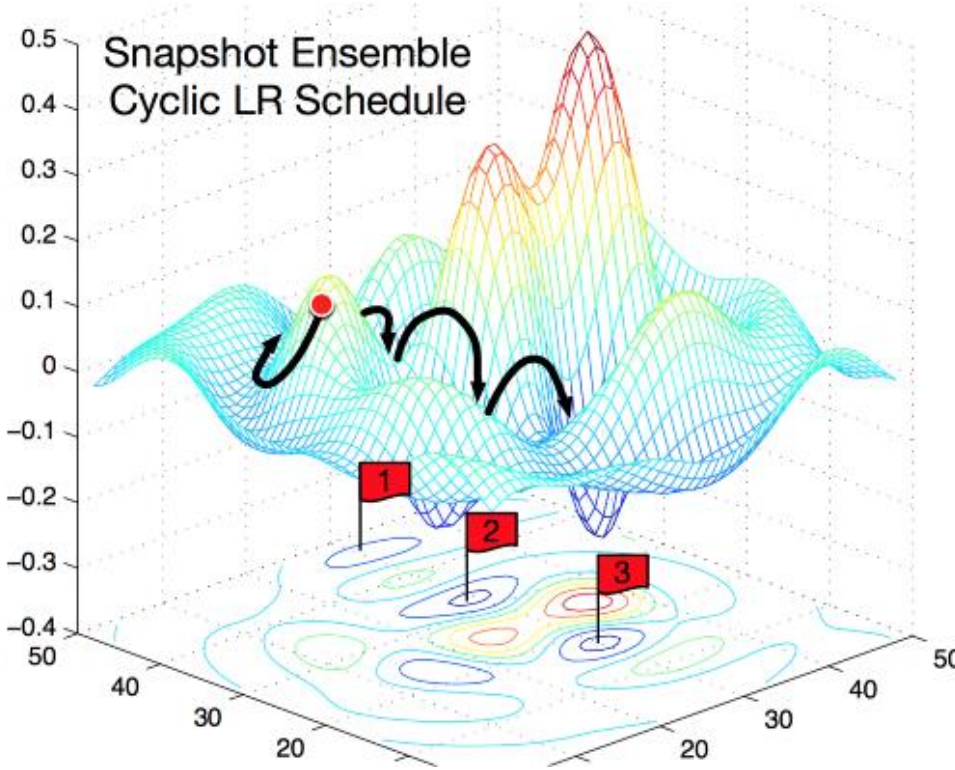
$$w_{ij} = w_{ij} + \Delta w_{ij}$$

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

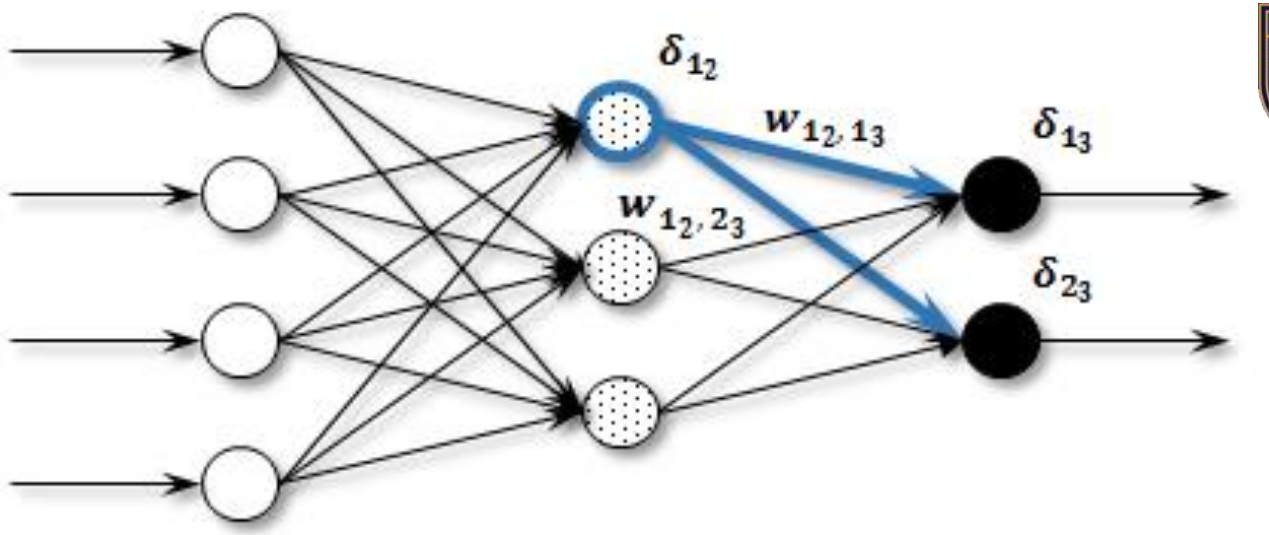

e
f'(w1)
f(w)
e1
e2
w
w1
w2

«Спуск» по поверхности ошибки





Итого:



- Для последнего слоя:

$$\delta_j = (y_j - \tilde{y}_j) f'(S_j)$$

- Для внутренних слоев:

$$\delta_j = \left(\sum_k \delta_k w_{jk} \right) f'(S_j)$$

- Для всех:

$$\Delta w_{ij} = -\eta \delta_j y_i$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

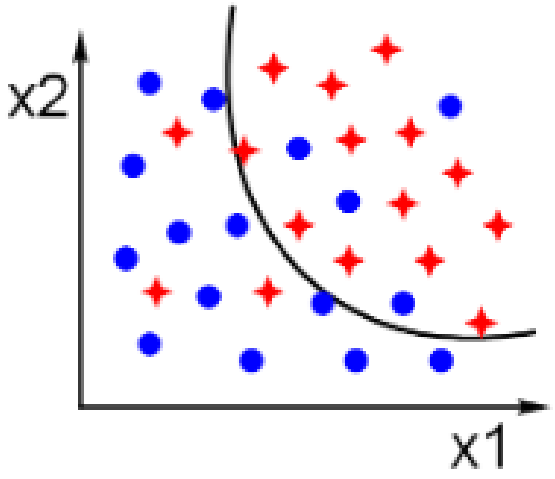
Проблемы обучения

- Паралич сети – сеть перестает обучаться.
- Переобучение
- Недообучение

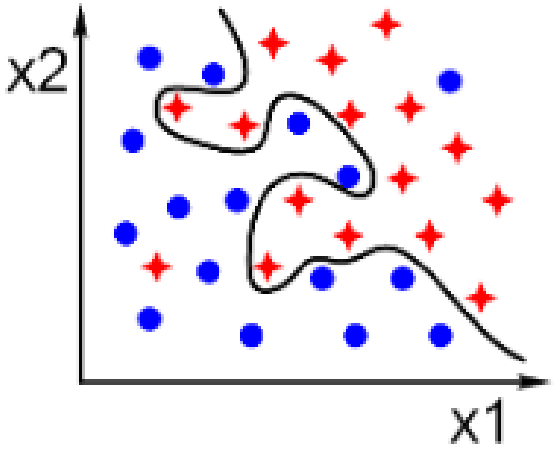
Причины:

- затухающий градиент
- взрывающийся градиент
- неправильный выбор гиперпараметров

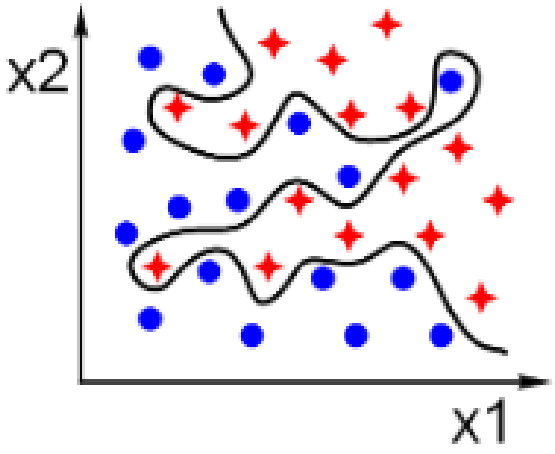
Недообучение



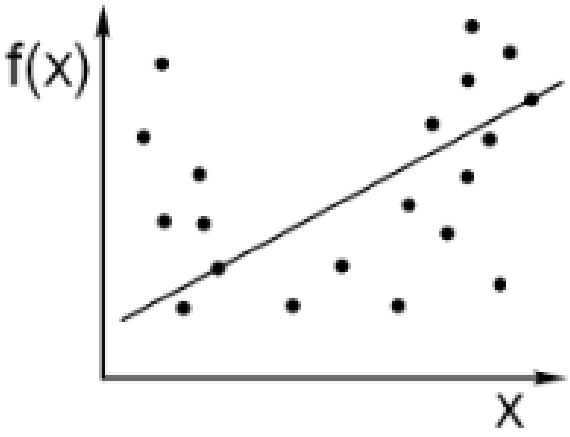
Оптимум



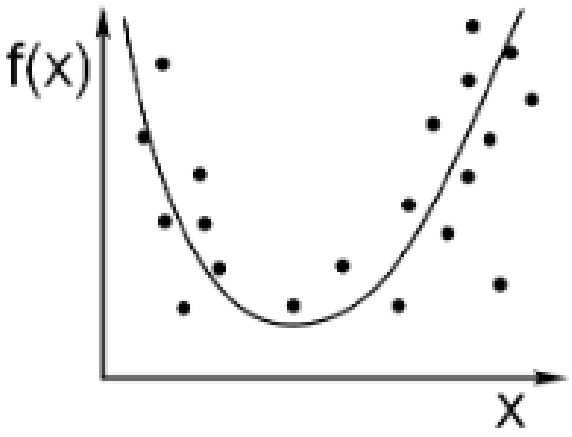
Переобучение



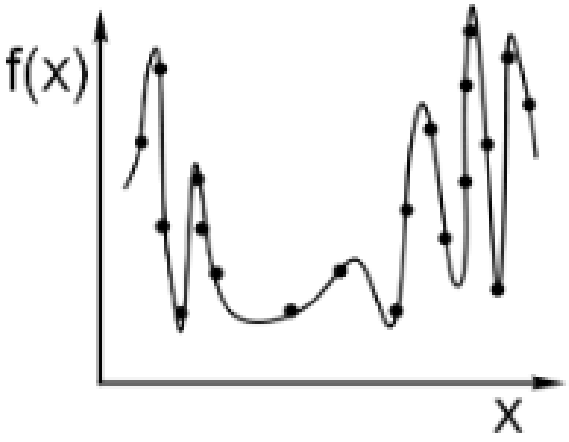
Недообучение



Оптимум



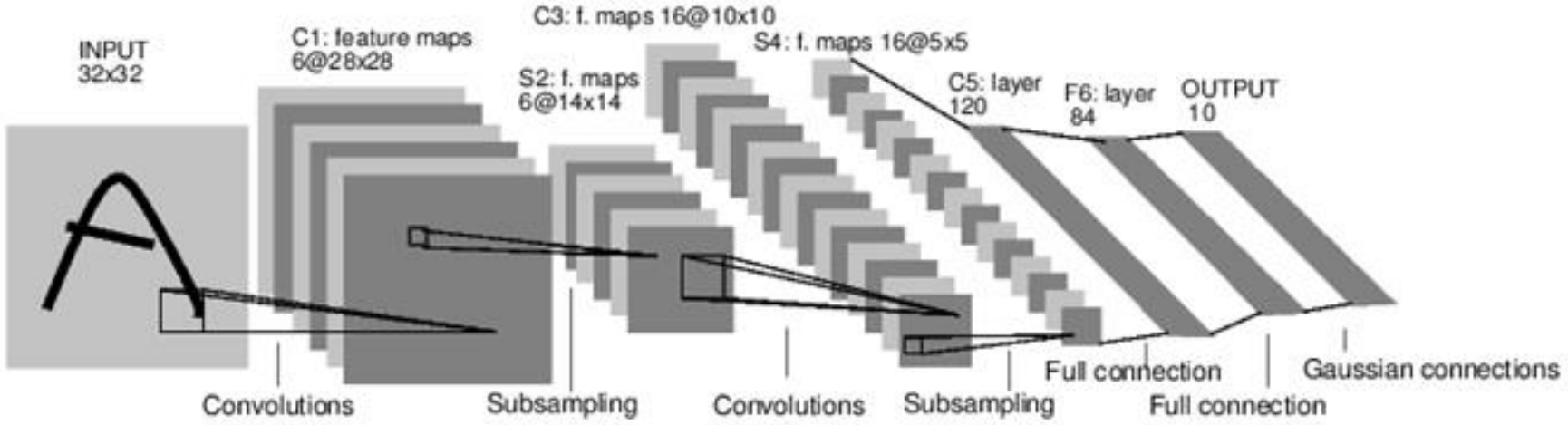
Переобучение



Контроль обучения

- Кросс-валидация
- Регуляризация
  - штраф за большие веса
  - dropout
  - batch norm
- Работа с обучающей выборкой

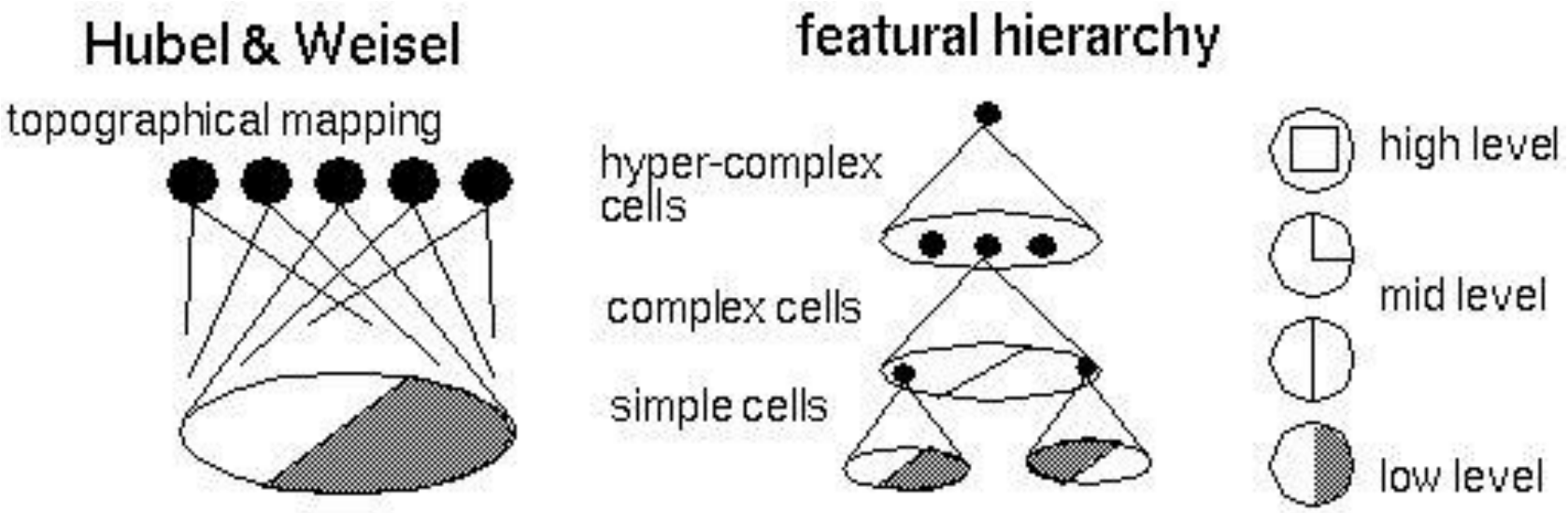
Сверточные нейронные сети



A Full Convolutional Neural Network (LeNet)

Сверточная нейронная сеть (*CNN*) — специальная архитектура нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание изображений. Идея заключается в чередовании сверточных слоев и субдискретизирующих слоев.

Понимание работы CNN



Показано, что мозг обрабатывает визуальную информацию иерархически: сначала находят границы, углы, а на более глубоких слоях – сложные объекты.

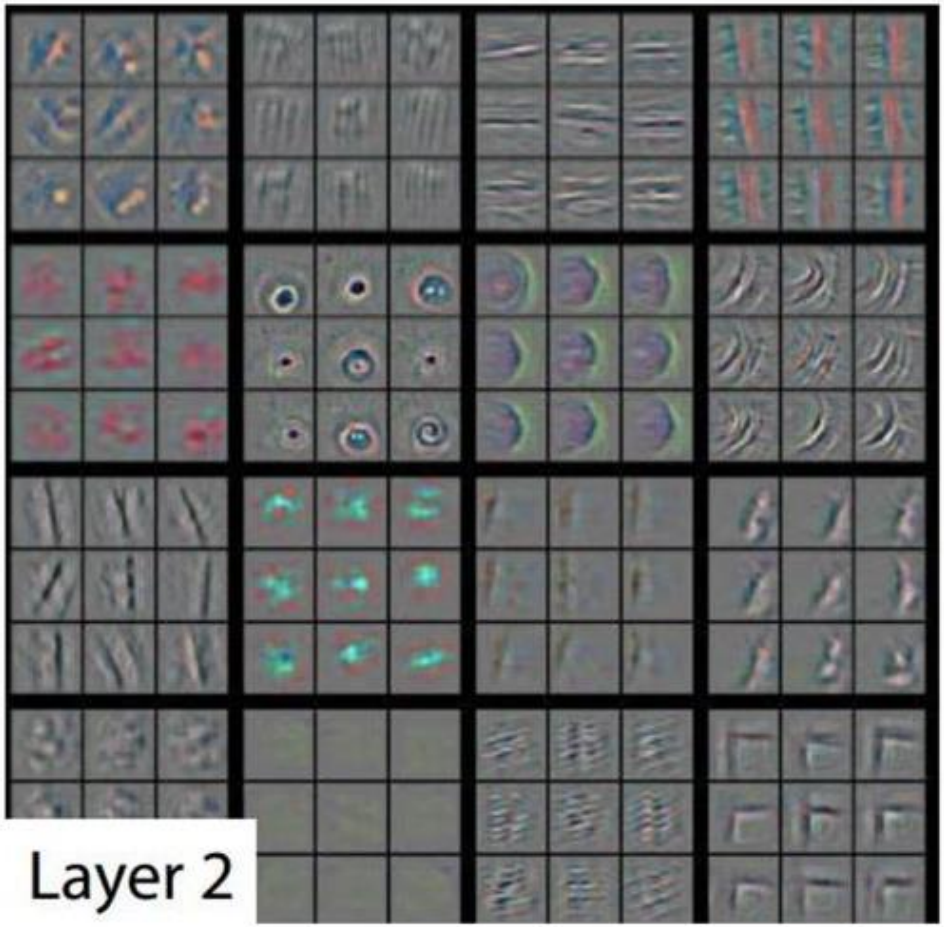
Выучиваемые признаки



Layer 1

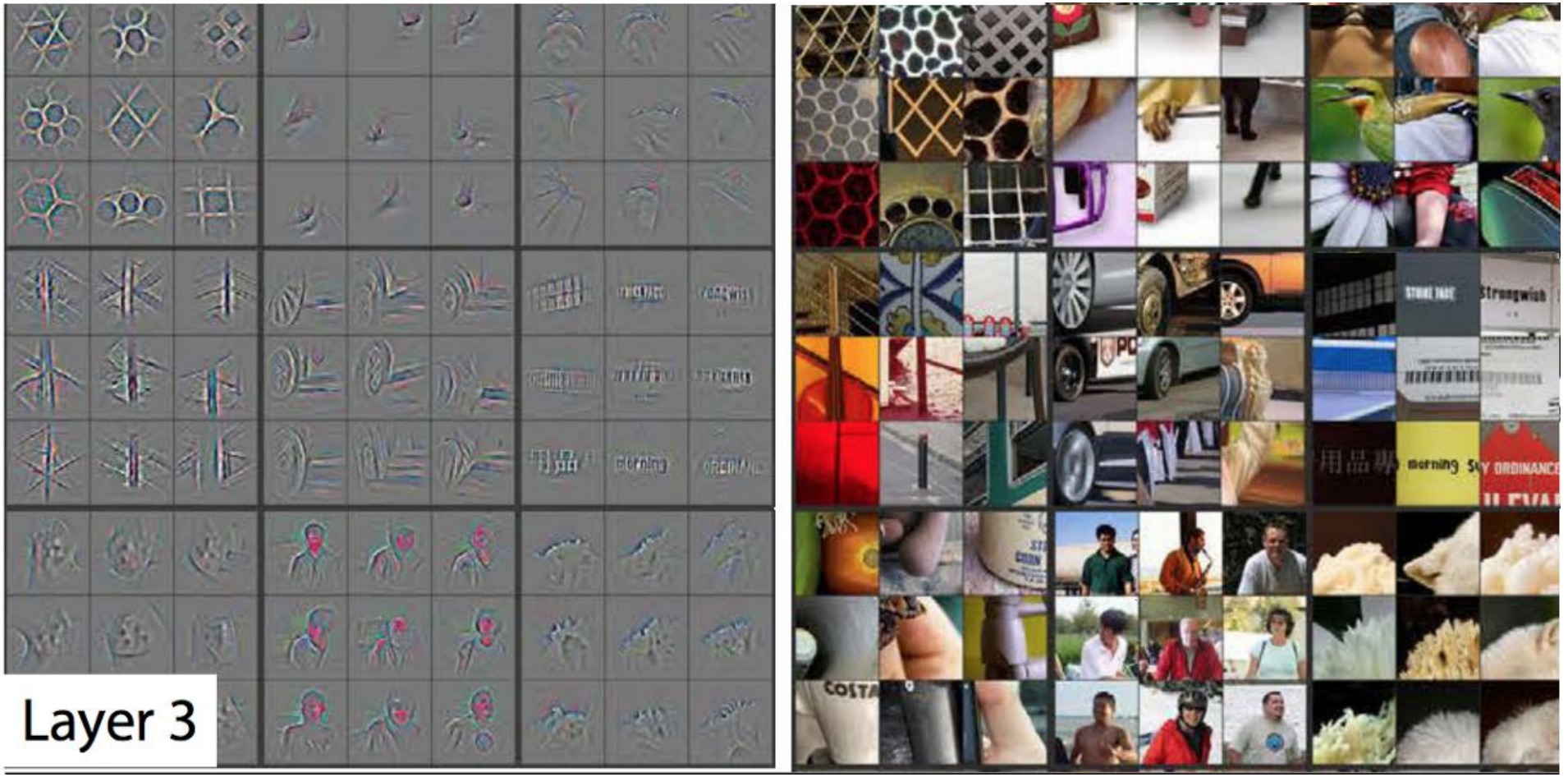


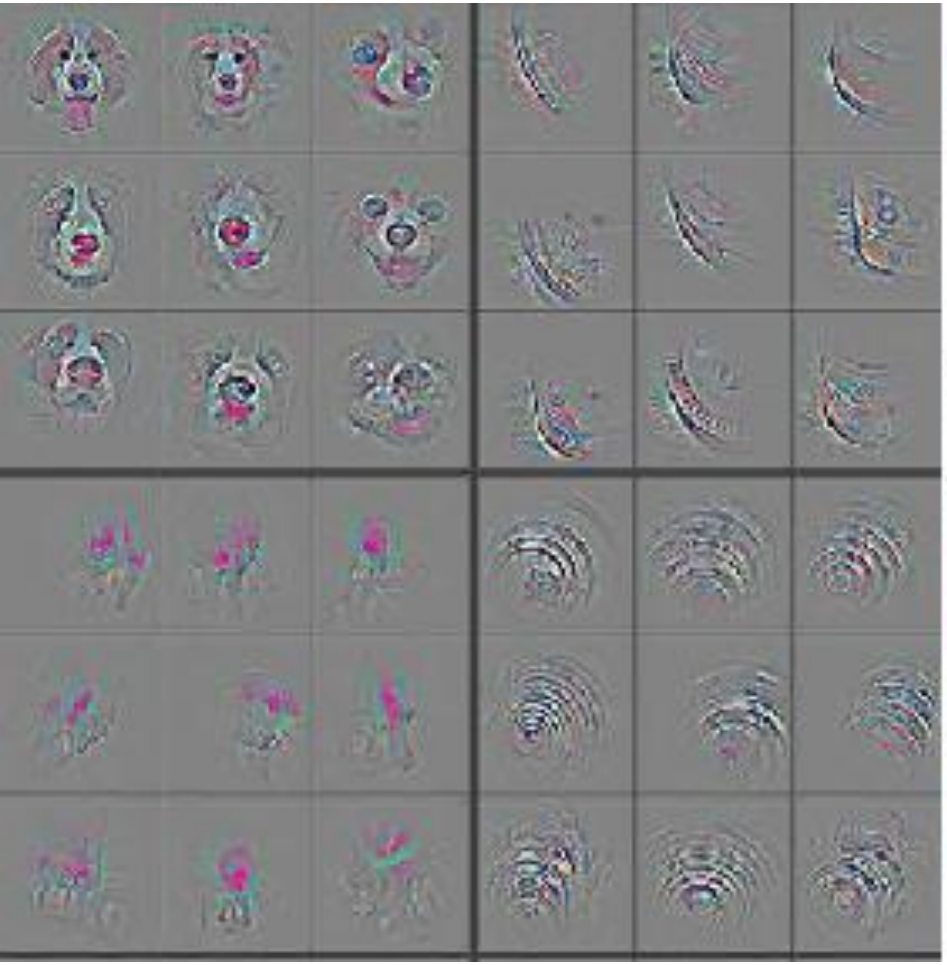
На рисунке показаны куски изображения, которые больше всего были ответственны за то, чтобы активировать нейрон на первом слое.



Layer 2




Layer 3





Layers 4, 5

Рекуррентные нейронные сети

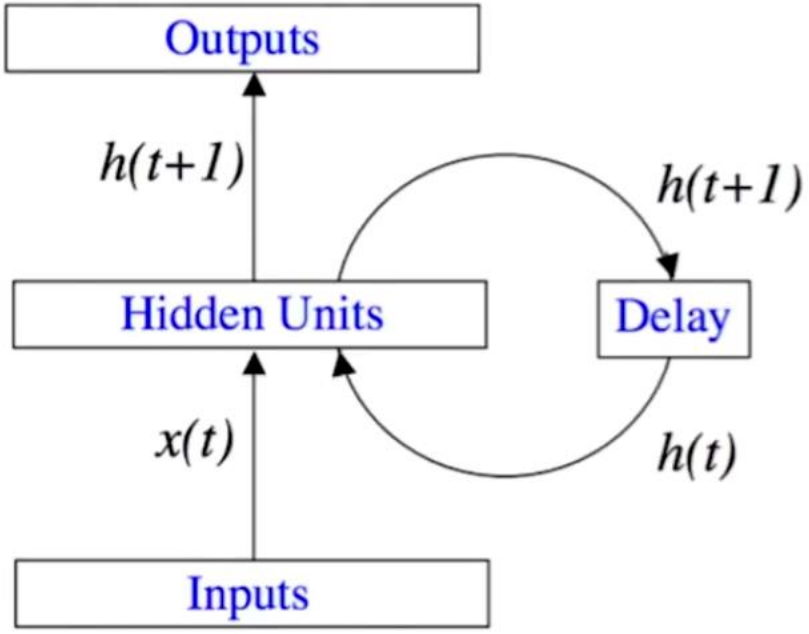


Рис.: RNN с задержкой на скрытом слое

- Рекуррентная нейронная сеть (RNN) — архитектура нейронных сетей, где связи между элементами образуют направленный цикл.
- Наличие таких циклов делает эту архитектуру идеальной для обработки последовательностей и данных, распределенных во времени.

- Все биологической нейронной сети – рекуррентные
- RNN моделирует динамическую систему
- Универсальная теорема аппроксимации говорит, что с помощью RNN можно смоделировать поведение любой динамической системы
- Существует много алгоритмов обучения RNN без явного лидера.





W1 = 0.5

$$f(x) = \begin{cases} 1 & x \geq 0.5 \\ 0 & x < 0.5 \end{cases}$$





W2 = -0.5









W3 = 0.5



1



0





W1 = 0.5

$$x = 0.5 * 1 + -0.5 * 0 + 0.5 * 0 = 0.5$$

$$f(x) = \begin{cases} 1 & x \geq 0.5 \\ 0 & x < 0.5 \end{cases}$$





W2 = -0.5







W3 = 0.5

1

0









0.25

0.25

0

0.5

-0.4

0.9

-1

1

$$f(x) = \begin{cases} 1 & x \geq 0.5 \\ 0 & x < 0.5 \end{cases}$$

1

0





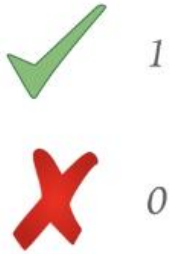


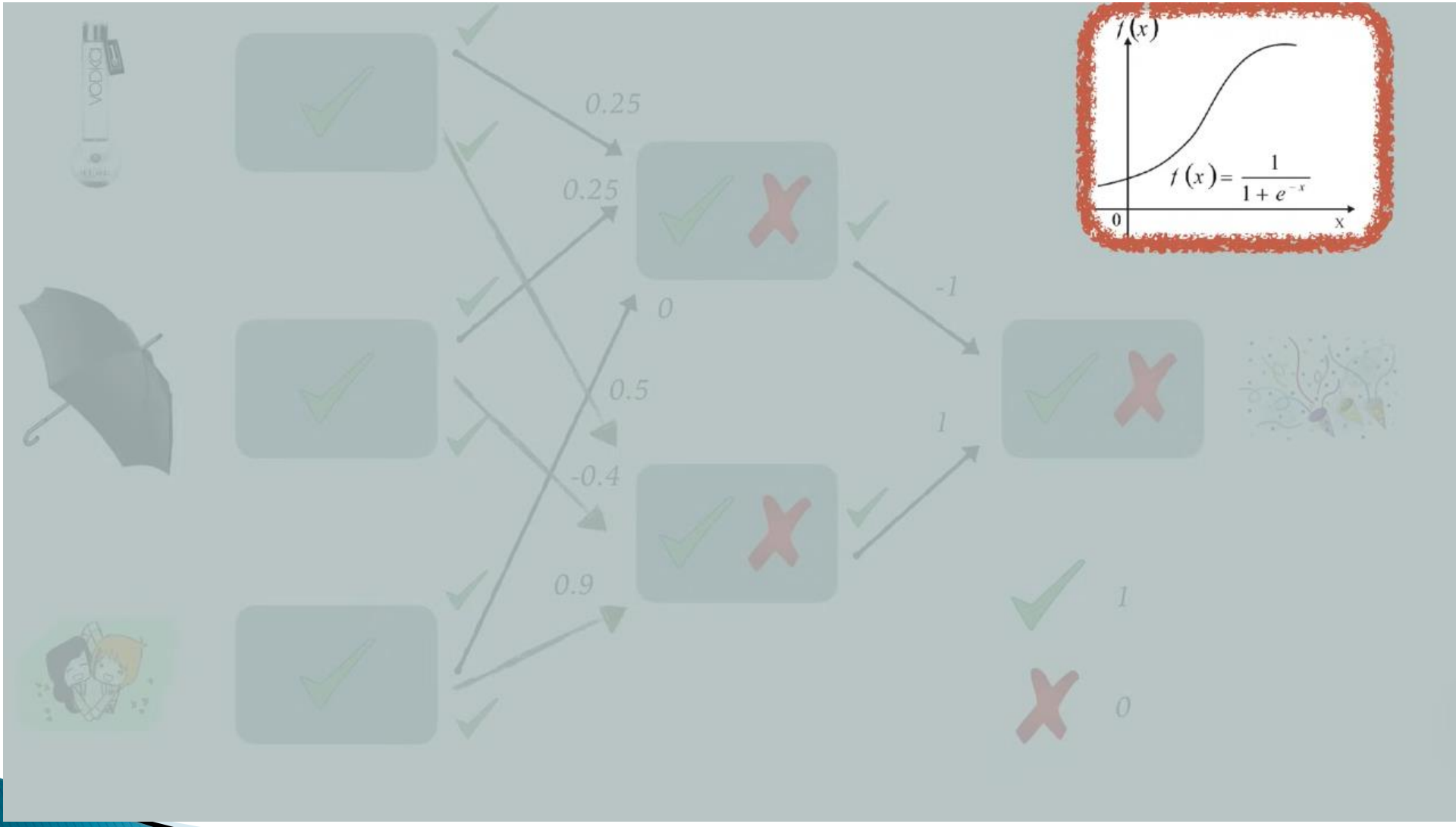


$$f(x) = \begin{cases} 1 & x \geq 0.5 \\ 0 & x < 0.5 \end{cases}$$







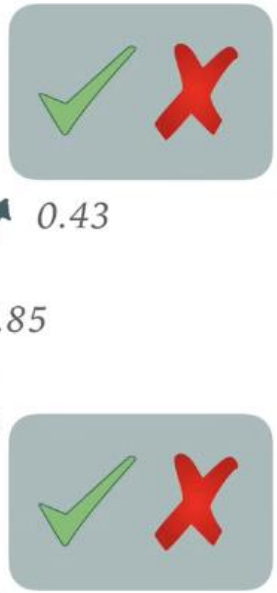

0.25
0.25
0
0.5
-0.4
0.9
-1
1
1
0
$f(x)$
$f(x)=\frac{1}{1+e^{-x}}$
0
x

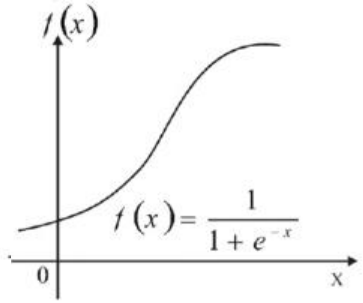




















0.79

0.44

0.43

0.85

0.43

0.29

0.5

0.52

1

0

$1*0.79 + 1*0.44 + 0*0.43 = 1.23$

$sigmoid(1.23) = 0.77$













0.79

0.44

0.43

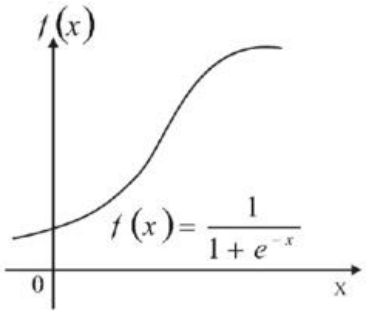
0.85

0.43

0.29

0.77

0.78



0.5

0.52

0.69



expected:

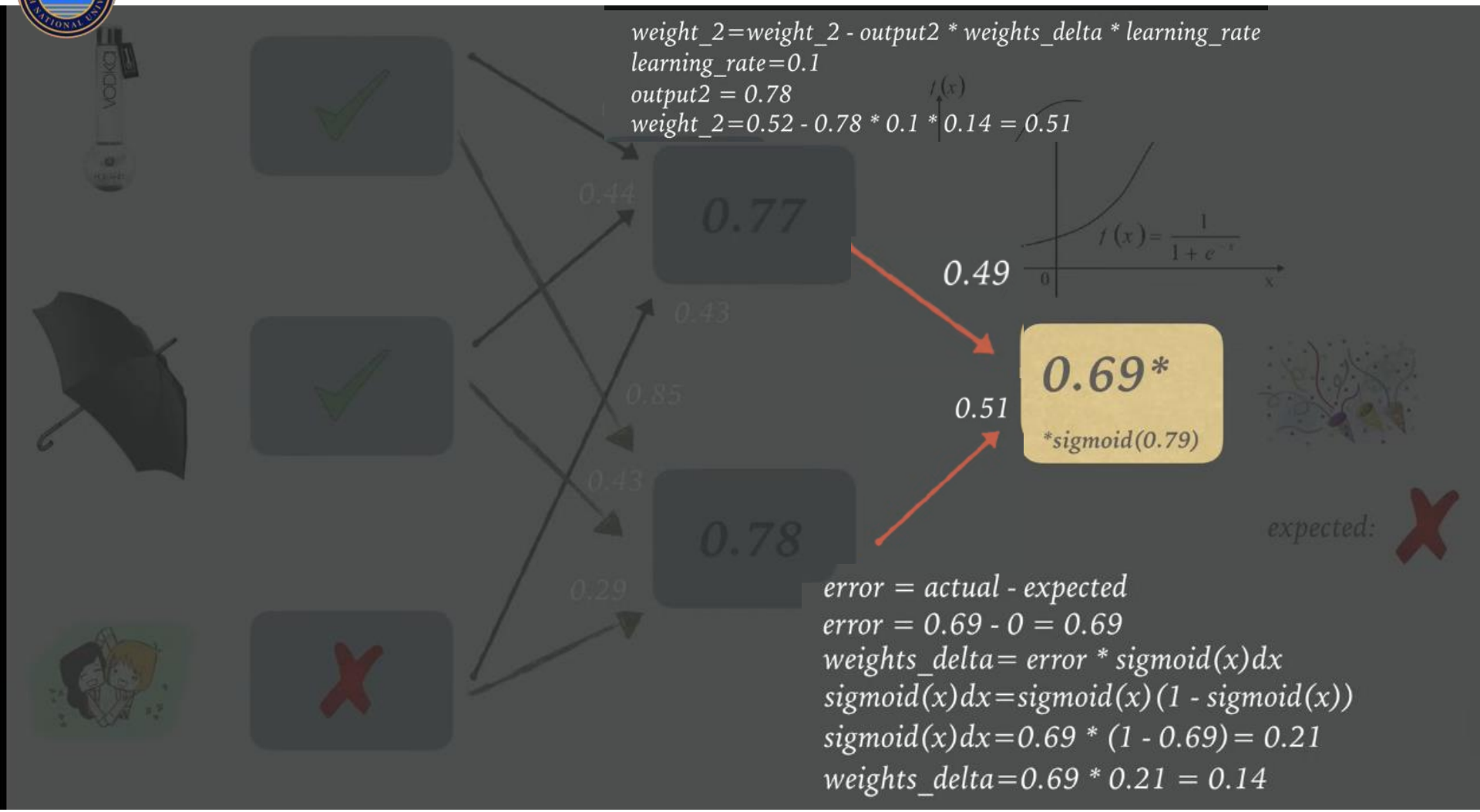


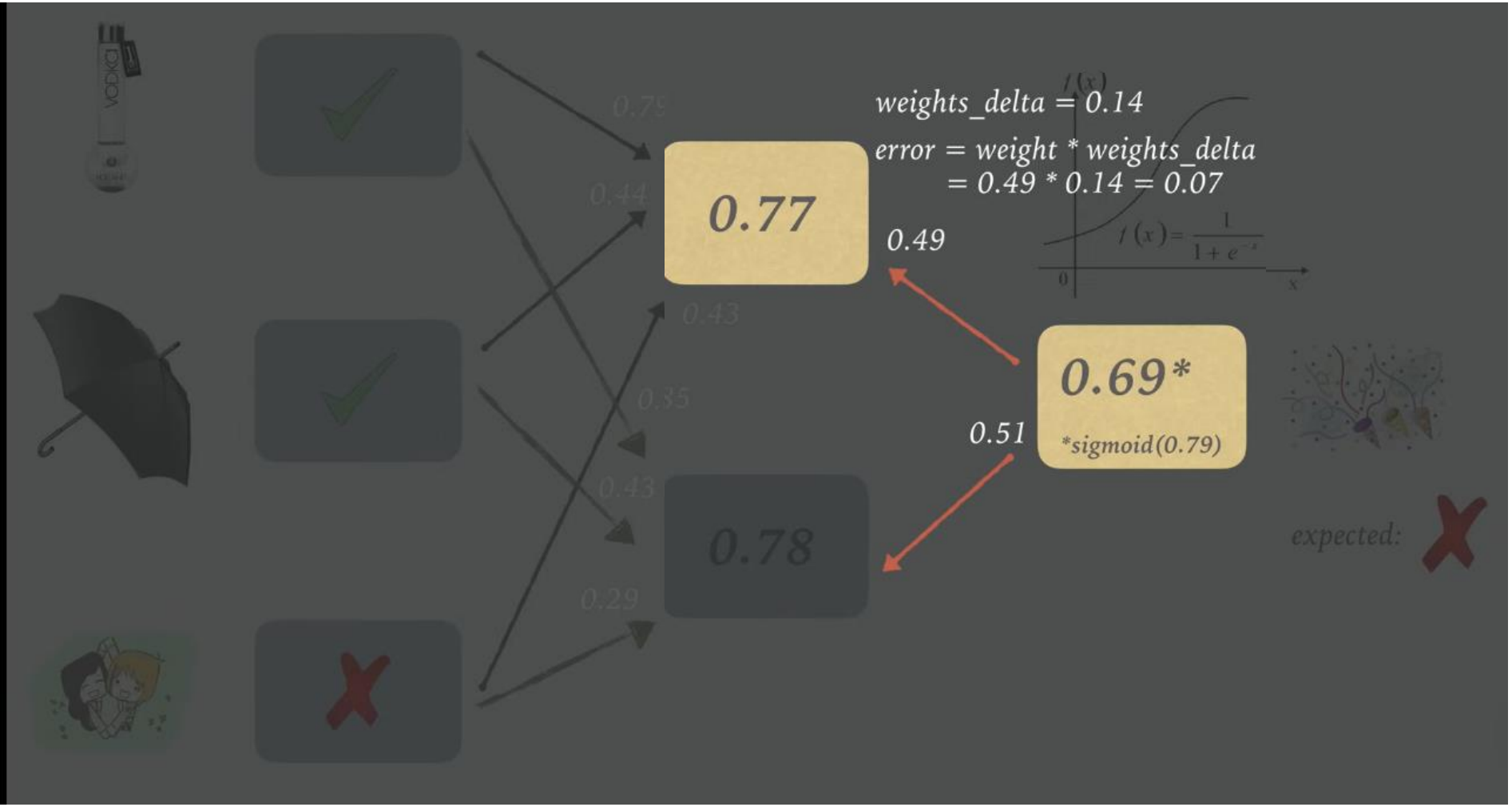
$0.77*0.5 + 0.78*0.52 = 0.79$

$sigmoid(0.79) = 0.69$

$1*0.85 + 1*0.43 + 0*0.29 = 1.28$

$sigmoid(1.28) = 0.78$


weight_2=weight_2 - output2 * weights_delta * learning_rate
learning_rate=0.1
output2 = 0.78
weight_2=0.52 - 0.78 * 0.1 * 0.14 = 0.51
0.44
0.77
0.43
0.85
0.43
0.78
0.29
0.49
0.51
0.69*
*sigmoid(0.79)
expected:
error = actual - expected
error = 0.69 - 0 = 0.69
weights_delta= error * sigmoid(x)dx
sigmoid(x)dx=sigmoid(x)(1 - sigmoid(x))
sigmoid(x)dx=0.69 * (1 - 0.69)= 0.21
weights_delta=0.69 * 0.21 = 0.14


0.77
0.78
0.69*
*sigmoid(0.79)
weights_delta = 0.14
error = weight * weights_delta
= 0.49 * 0.14 = 0.07
0.49
0.51
0.75
0.44
0.43
0.35
0.43
0.29
expected:

Спасибо за внимание!