ТЕХНОСФЕРА

Лекция n1
Основы нейронных сетей

Нестеров Павел

17 марта 2016 г.

План лекции

Принципиальное отличие

Теория статистического обучения

Линейная и логистическая регрессии, expectation-maximization, naive bayes classifier, random forest, support vectior machine, gradient boosting trees и т.д.

Имитация работы мозга человека

Perceptron, cognitron, self-organizing maps, multi-layer feedforward network, convolution network, Boltzmann machine, deep neural network и т.д.

С другой стороны

Искусственная нейронная сеть

- алгоритмическая композиция (ансамбль) слабых моделей
- байесова или марковская сеть
- ???
- или любое другое *обобщение*, важны идеи лежащие в основе теории

Нервная система до нейробиологии

Рис.: 17 век, Рене Декарт о нервной системе: «Раздражение ступни передаётся по нервам в мозг, взаимодействует там с духом и таким образом порождает ощущение боли».

Нервная система в современном понимании

- В 1906 году врач и гистолог Сантьяго Рамон-и-Кахаль совместно с врачем Камилло Гольджи получают нобелевскую премию за "за работы по структуре нервной системы"; их работы заложили основы нейронной теории нервной системы и современной нейробиологии.

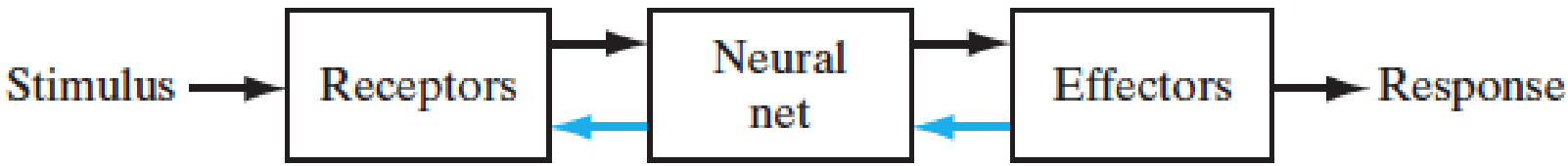
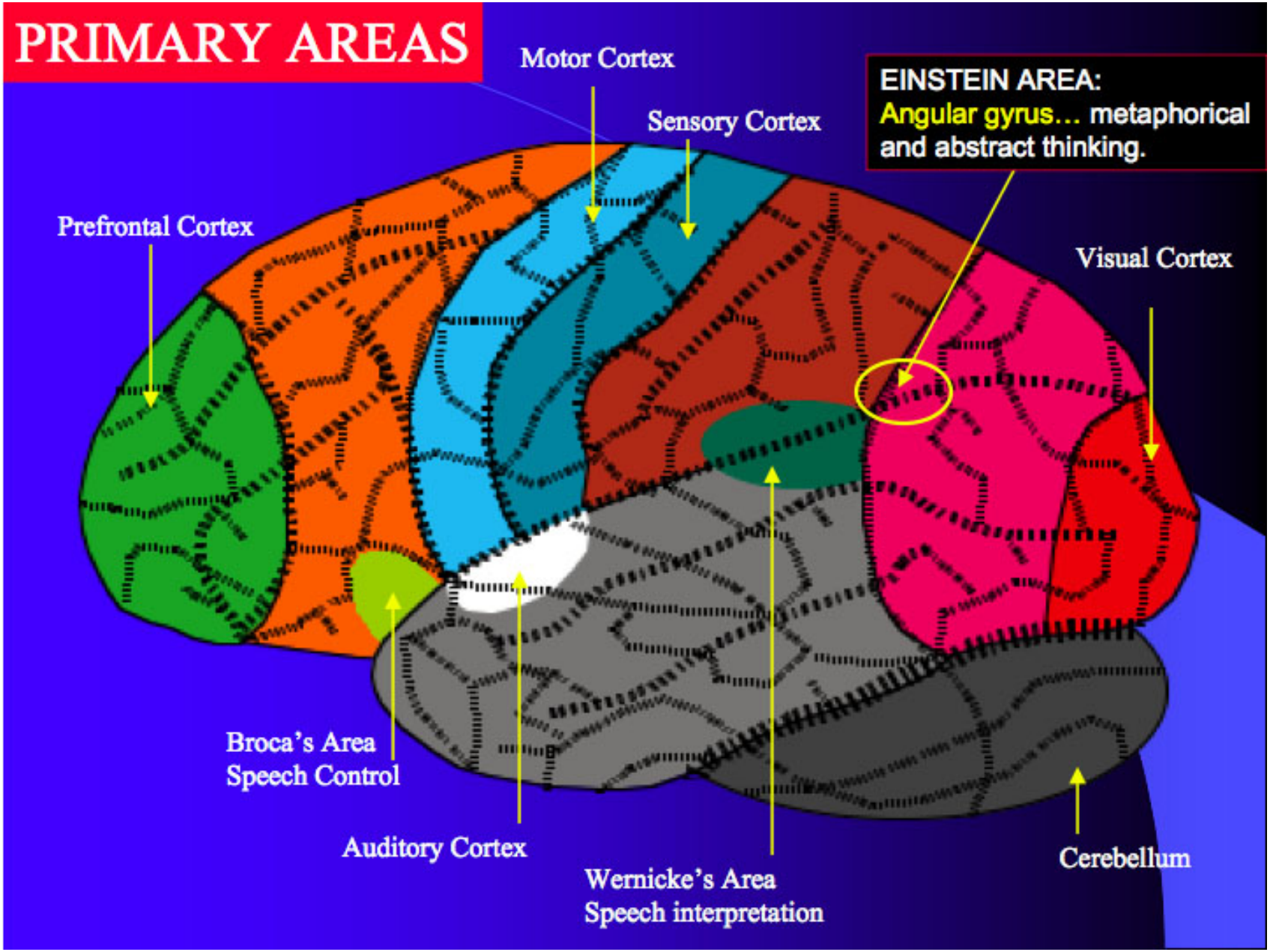


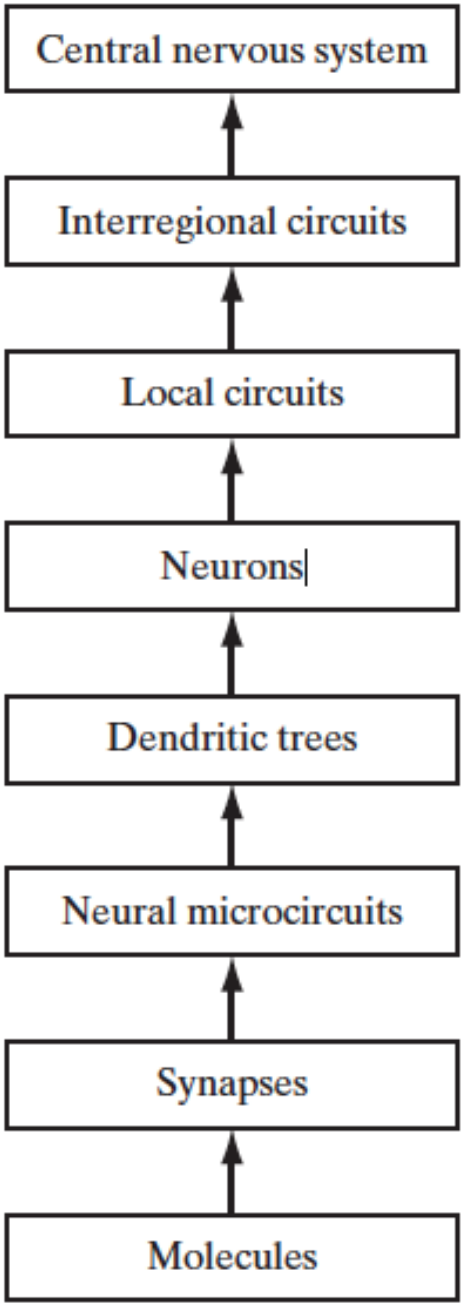
Рис.: Блок-схема нервной системы[1]

[1] Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Мозг человека, #1

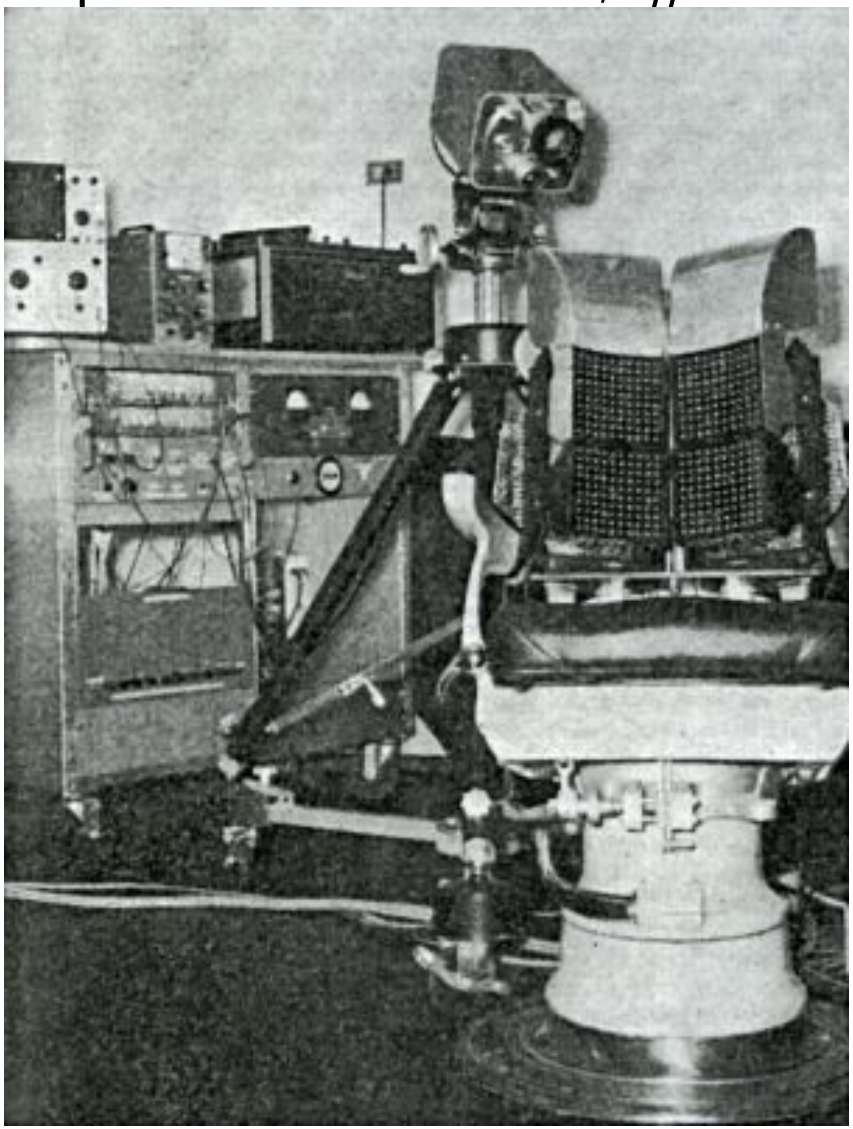


Мозг человека, #2



- 10^{11} нейронов в мозге, из них 10^9 в коре
  - CPU: *15-core Xeon IvyBridge-EX* - $4.3 \cdot 10^9$ транзисторов
  - GPU: *Nvidia Tesla GK110 Kepler* - $7.08 \cdot 10^9$ транзисторов
- нейроны (и мозг в целом) обладают **нейропластичностью** - способностью изменяться под действием опыта;
- мозг - комплексная, нелинейная система параллельной обработки данных, способная изменять структуру своих составных частей;
- мозг решает определенный тип задач значительно быстрее чем любой современный компьютер, несмотря на то, что нейрон крайне *медленная* вычислительная единица.

Нейропластичность, #1



- эксперимент нейрофизиолога Paul Bach-y-Rita, 1969 года
- сетка 20x20 стимуляторов на расстоянии 12мм

Нейропластичность, #2

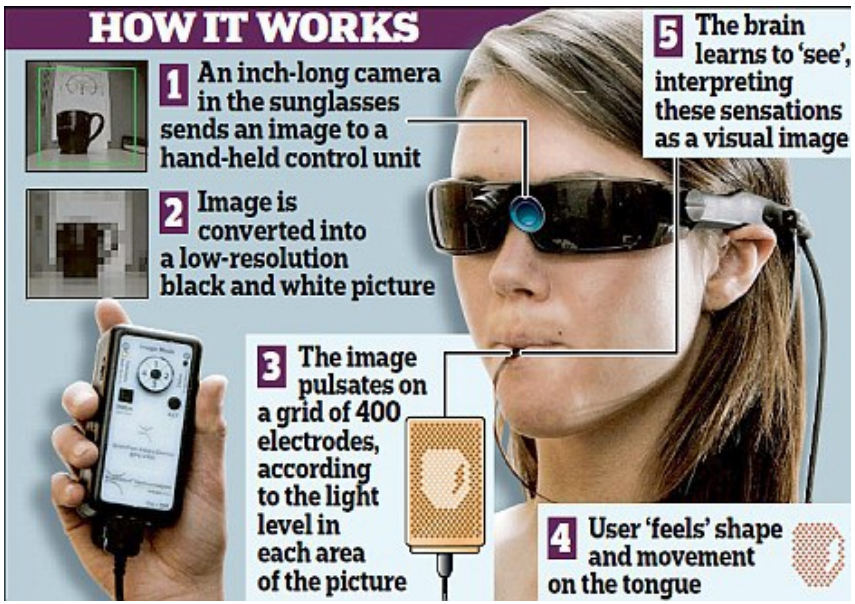
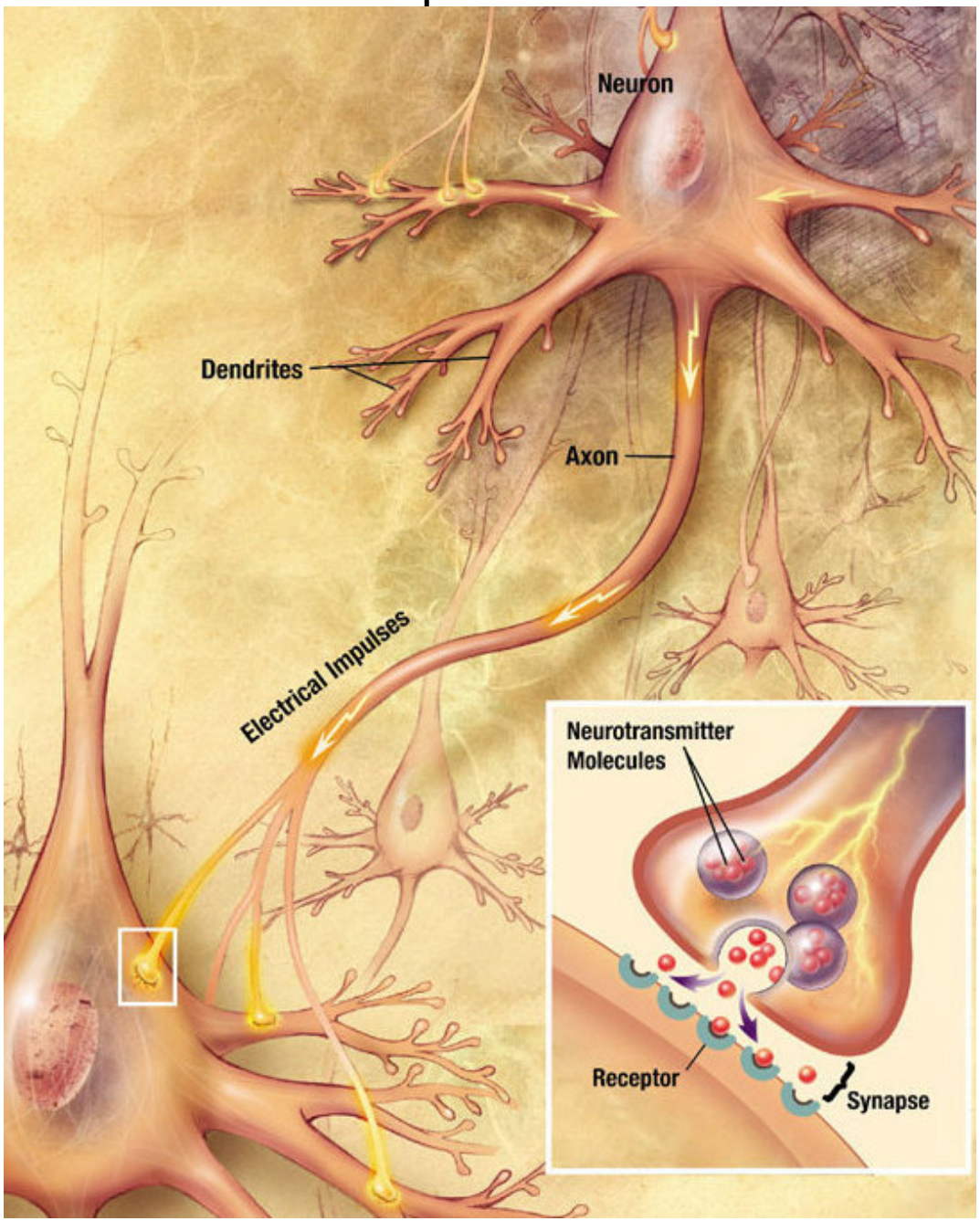


Схема биологического нейрона



Нейронные ансамбли

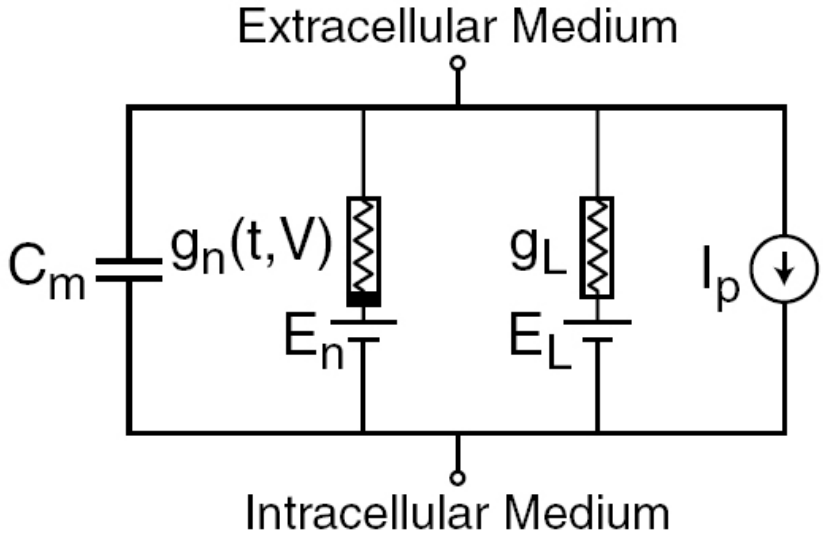
- Физиолог и нейропсихолог Дональд Хебб разработал теорию взаимосвязи головного мозга и мыслительных процессов в книге "The Organization of Behavior"(1949).
- Нейронный ансамбль - совокупность нейронов, составляющих функциональную группу в высших отделах мозга.
- Нейроансамбль - распределенный способ кодирования информации.
- Нейрон сам по себе генерирует по мимо сигнала еще и шум, но ансамбль в среднем генерирует чистый сигнал (*bagging?*).

Нейронная модель Ходжкина-Хаксли, #1

- Модель Ходжкина—Хаксли (1952 год) — математическая модель, описывающая генерацию и распространение потенциалов действия в нейронах[2].
- Потенциал действия — волна возбуждения, перемещающаяся по мембране живой клетки в процессе передачи нервного сигнала.
- Нобелевская премия по физиологии и медицине "за открытия, касающиеся ионных механизмов возбуждения и торможения в периферических и центральных участках нервных клеток"(1963 год).

[2] см. приложение Cellmembranion.gif

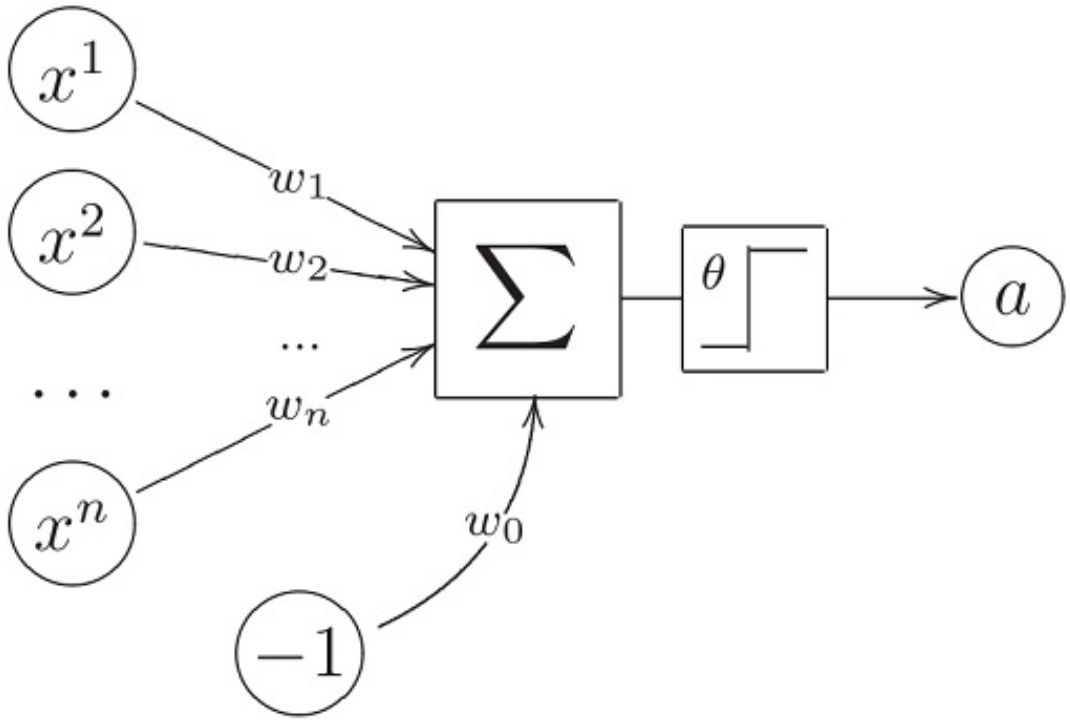
Нейронная модель Ходжкина-Хаксли, #2



Каждому элементу схемы соответствует свой биохимический аналог:

- C_m - электроемкость внутреннего липидного слоя клеточной мембраны;
- g_n - потенциал-зависимые ионные каналы отвечают за нелинейную электрическую проводимость;
- g_L - каналы мембранных пор отвечают за пассивную проводимость;
- E_x - источники напряжения побуждает ионы к движению через мембранные каналы.

Модель МакКаллока-Питтса (1943 год)



- $a(x) = \theta\left(\sum_{j=1}^{n} w_j \cdot x_i - w_0\right)$;
- $\theta(z) = [z \geq 0] = \begin{cases} 0, z<0 \\ 1, z \geq 0 \end{cases}$ - функция Хевисайда;
- эквивалентно линейному классификатору.

Данная модель, с незначительными изменениями, актуальна и по сей день.

Правила Хебба (1949)

В своей книге Дональд Хебб описал процесс адаптирования нейронов в мозге в процессе обучения, и сформулировал базовые механизмы нейропластичности:

1. если два нейрона по разные стороны от синапсов активируются синхронно, то "вес"синапса слегка возрастает;
2. если два нейрона по разные стороны от синапсов активируются асинхронно, то "вес"синапса слегка ослабевает или синапс удаляется[3].

Эти правила легли в основу зарождающейся теории нейросетей, сегодня мы можем увидеть этот мета-алгоритм в основных методах обучения нейронных сетей.

[3] это расширенное правило, в оригинале второй части не было

Правила Хебба, математическая формулировка, #1

Допустим у нас имеется набор бинарных векторов размерности n, каждому из которых соответствует бинарный выход y:

- $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_m\}, \forall \vec{x}_i \in \{0,1\}^n$
- $Y = \{y_1, y_2, \ldots, y_m\}, \forall y_i \in \{0,1\}$
- $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

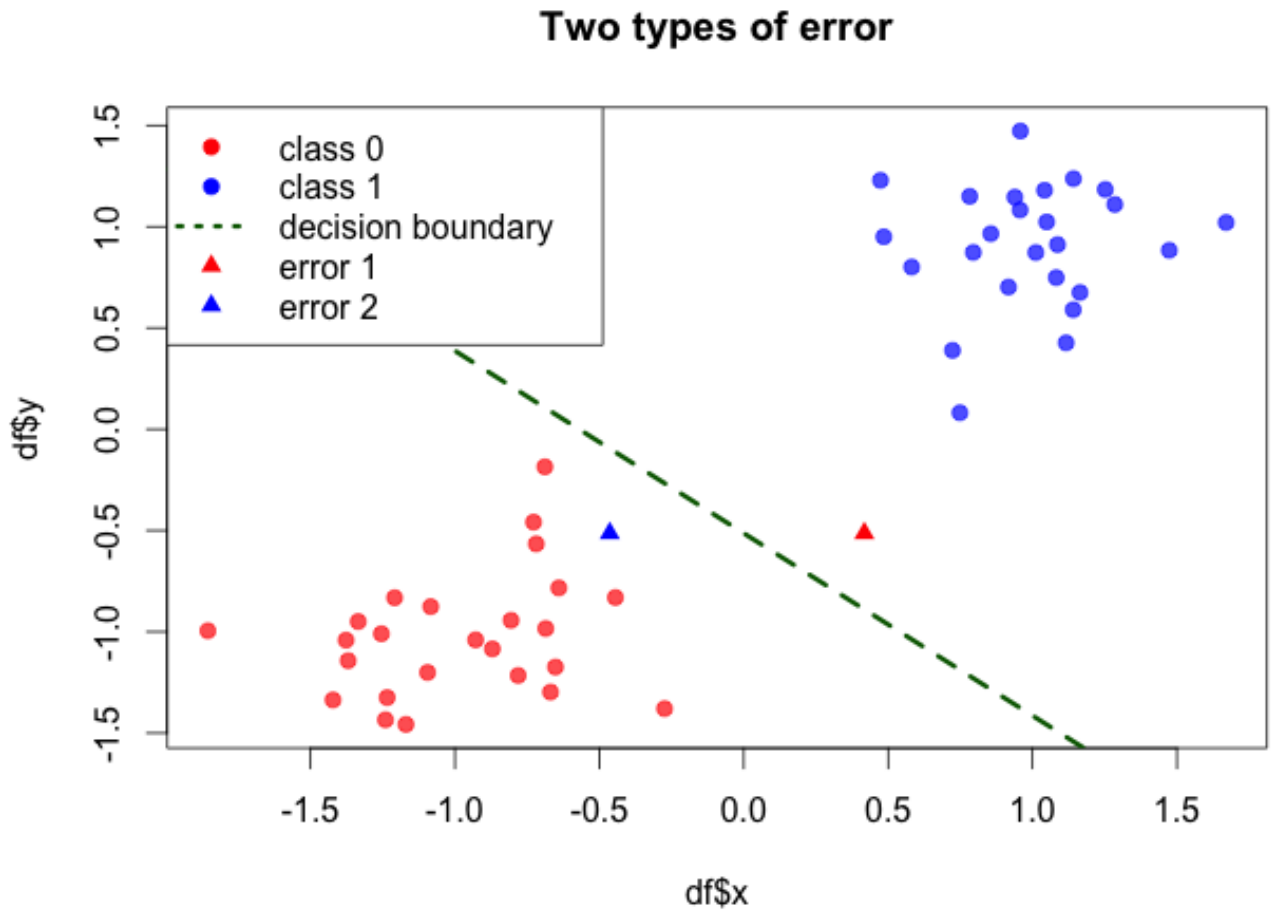
Тогда нейрон может совершить два типа ошибок:

1. $\hat{y}_i = 0 \wedge y_i = 1 \Rightarrow$ *увеличить* веса при тех входах равных 1
   - *зачем?*
2. $\hat{y}_i = 1 \wedge y_i = 0 \Rightarrow$ *уменьшить* веса при тех входах равных 1
   - *зачем?*

Правила Хебба, математическая формулировка, #2

Тогда нейрон может совершить два типа ошибок:

1. $\hat{y}_i = 0 \wedge y_i = 1 \Rightarrow$ *увеличить* веса при тех входах равных 1
   - не преодолели порог $\Rightarrow$ увеличить скалярное произведение
2. $\hat{y}_i = 1 \wedge y_i = 0 \Rightarrow$ *уменьшить* веса при тех входах равных 1
   - перешагнули порог $\Rightarrow$ уменьшить скалярное произведение



Однослойный персептрон Розенблатта (1958 год)

Нейрофизиолог Френк Розенблатт предложил схему устройства, моделирующего процесс человеческого восприятия, и назвал его "персептроном". По мимо этого:

- показал, что персептрон может выполнять базовые логические операции;
- разработал алгоритм обучения такой модели - метод коррекции ошибки;
- доказал сходимость алгоритма (теорема сходимости персептрона), но только для линейно разделимых классов;
- реализовал физический прототип такой модели;
- реализовал первый в мире нейрокомпьютер "MARK-1".

Нейрокомпьютер MARK-1

New York Times о персептроне, 8 июля, 1958

NEW NAVY DEVICE LEARNS BY DOING

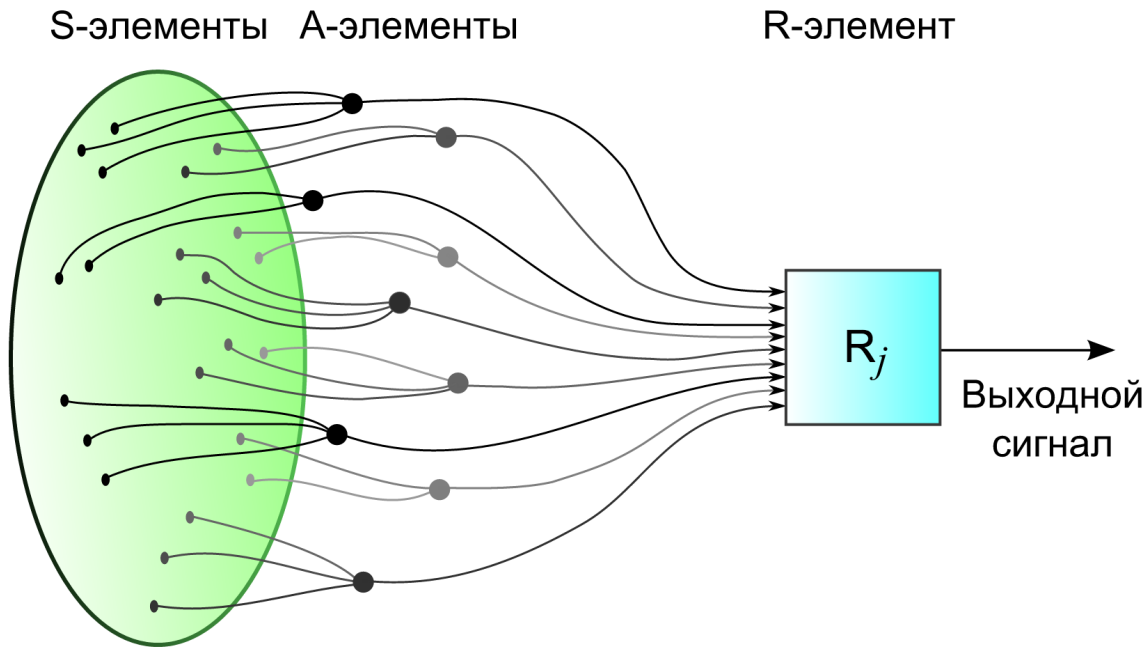
Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

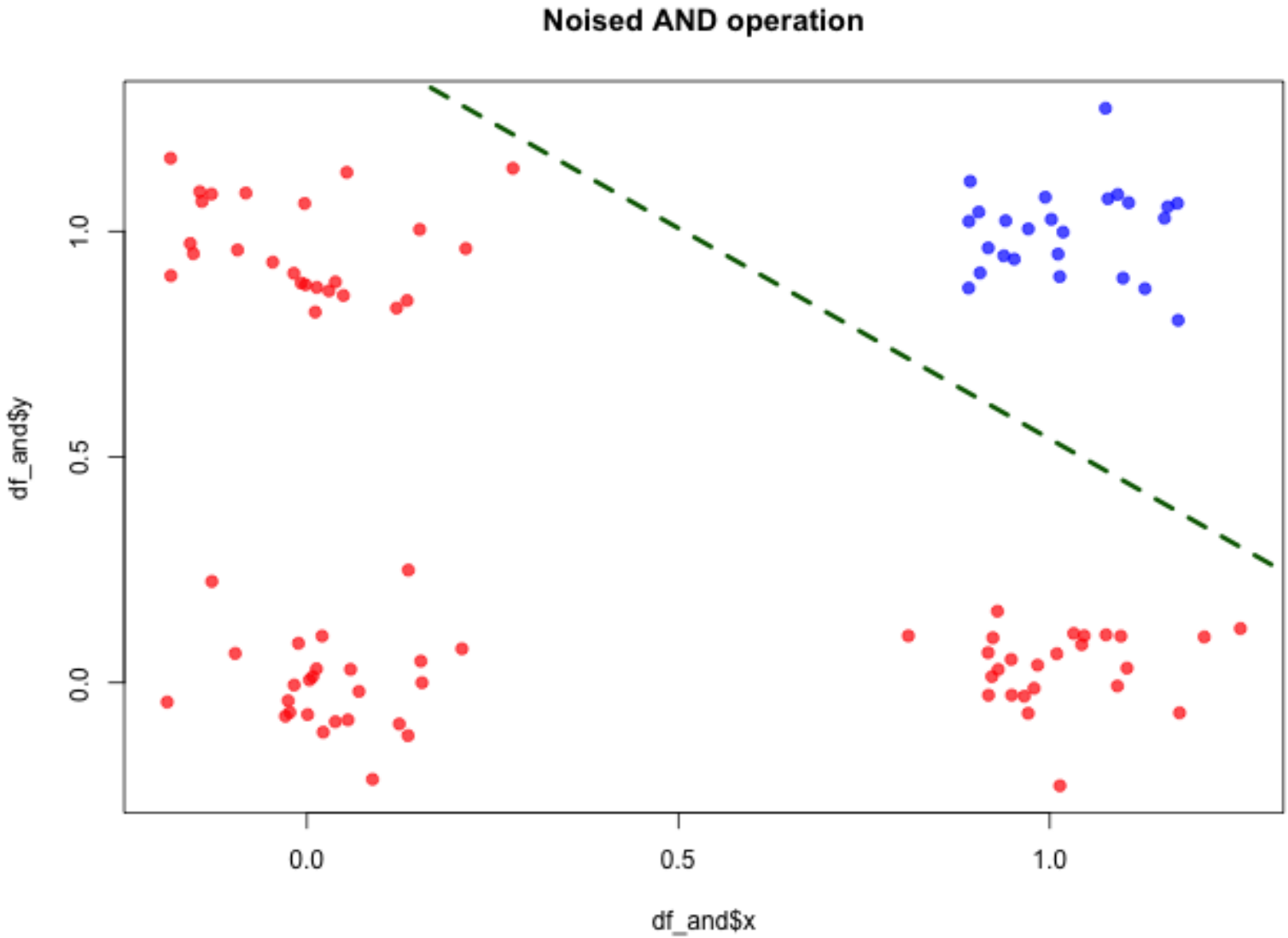
Элементарный персептрон



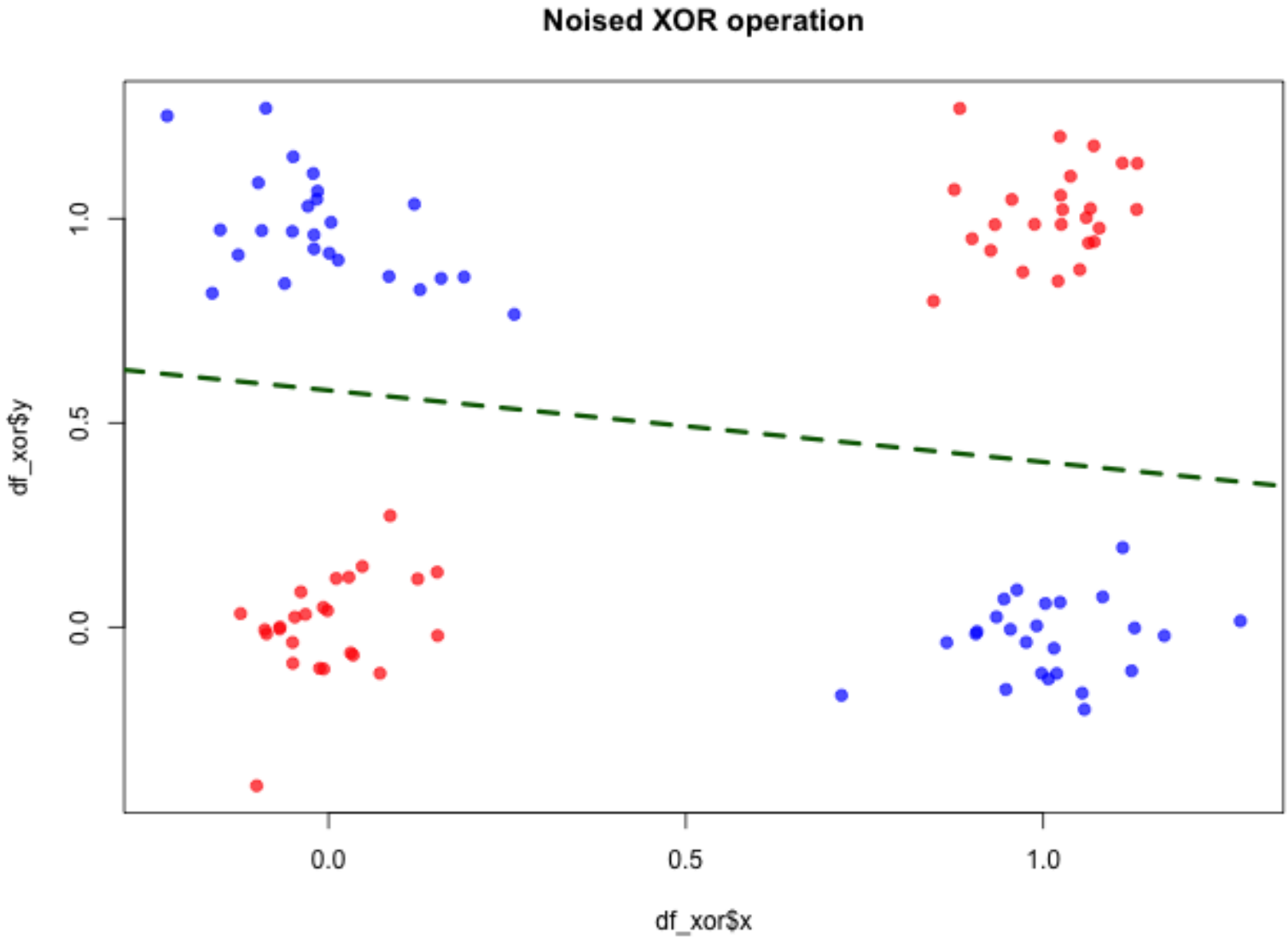
Метод коррекции ошибки

- $\hat{y}_i = 0 \wedge y_i = 1 \Rightarrow \Delta w = \eta(n) \cdot x(n)$
- $\hat{y}_i = 1 \wedge y_i = 0 \Rightarrow \Delta w = -\eta(n) \cdot x(n)$
  - $\eta(n)$ - скорость обучения, зависящая от итерации
  - $x(n)$ - входной образ на итерации n

Недостатки элементарного персептрона, AND



Недостатки элементарного персептрона, XOR



Анимация сходимости для операций AND и XOR

- операция OR - 1layer-net-or.gif
- операция AND - 1layer-net-and1.gif
- операция XOR - 1layer-net-xor.gif [4]

[4] http://theclevermachine.wordpress.com/2014/09/11/a-gentle-introduction-to-artificial-neural-networks/

Доказательства неэффективности нейронных сетей

- В 1969 году математик и исследователь ИИ Марвин Минский провел детальный математический анализ персептрона и опубликовал формальное доказательство ограниченности этой модели.
- *"**There is no reason to suppose** that any of these virtues carry over to the many-layered version. Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgement that the extension to multilayer systems is sterile."*[5]
- Отсутствие преимуществ + ограничения модели в итоге поубавили интерес научного сообщества к нейронным сетям, требовалось, что то принципиально новое

[5] Персептроны, Марвин Минский в соавторстве с Сеймуром Папертом, MIT Press, 1969

Период "забвения"

Исследования *искусственных* нейросетей не спеша продолжаются, но в режиме поиска чего-то нового:

- 1972: Т. Кохонен разрабатывает новый тип нейросетей, для задачи manifold embedding и topology preserving mapping;
- 1975-1980: К. Фукусима разрабатывает когнитрон и неокогнитрон, совершенно новый тип многослойной сверточной сети, созданной по аналогии со строением зрительной системы;
- 1982: Дж. Хопфилд разрабатывает новый тип нейросети с обратными связями, выполняющей функции ассоциативной памяти;
- 1986: Дэвид Румельхарт, **Дж. Хинтон** и Рональд Вильямс разрабатывают *вычислительно эффективный* многослойных нейросетей - метод обратного распространения ошибки (именно работа этих авторов возрождает интерес к нейронным сетям в мире).

Теорема универсальной аппроксимации[6]

Введем следующие обозначения:

- $\phi(x)$ - не тождественная, ограниченная и монотонно возрастающая функция
- I_n - n-мерный единичный гиперкуб
- $C(I_n)$ - множество непрерывных функций на I_n

$\Rightarrow \forall f \wedge \forall \epsilon > 0 \exists$

- $m \in \mathbb{N}$
- $\{\beta_i\}_{i=1\ldots m}, \forall \beta_i \in \mathbb{R}$
- $\{\alpha_i\}_{i=1\ldots m}, \forall \alpha_i \in \mathbb{R}$
- $\{w_{ij}\}_{j=1\ldots n, i=1\ldots m}, \forall w_{ij} \in \mathbb{R}$

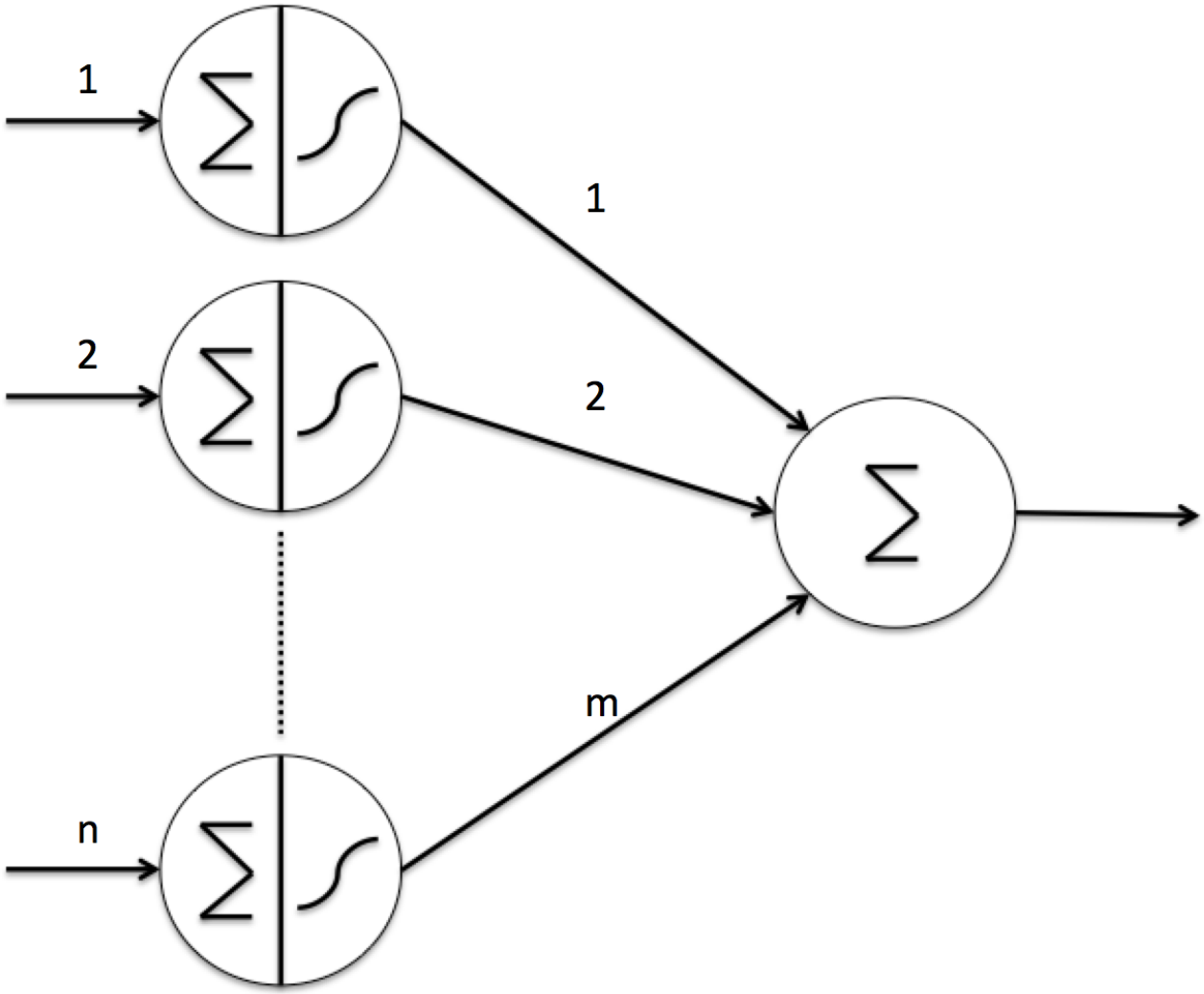
$\wedge \exists F(x_1, \ldots, x_n) = \sum_{i=1}^{m} \alpha_i \phi\left(\sum_{j=1}^{n} w_{ij} \cdot x_j + \beta_i\right)$:

- $|F(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)| < \epsilon$

Какая архитектура нейросети удовлетворяет такой формулировке?

[6] Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Универсальный аппроксиматор



В чем проблема универсального аппроксиматора, исходя из условий теоремы?

Демонстрация сходимости нейросети с одним скрытым слоем

- операция XOR - 2layer-net-xor.gif
- бинарная классификация - 2layer-net-ring.gif
- аппроксимация функции sin - 2layer-net-regression-sine.gif
- аппроксимация функции abs - 2layer-net-regression-abs.gif[7]

[7] http://theclevermachine.wordpress.com/2014/09/11/a-gentle-introduction-to-artificial-neural-networks/

Многоснойная нейронная сеть прямого распространения

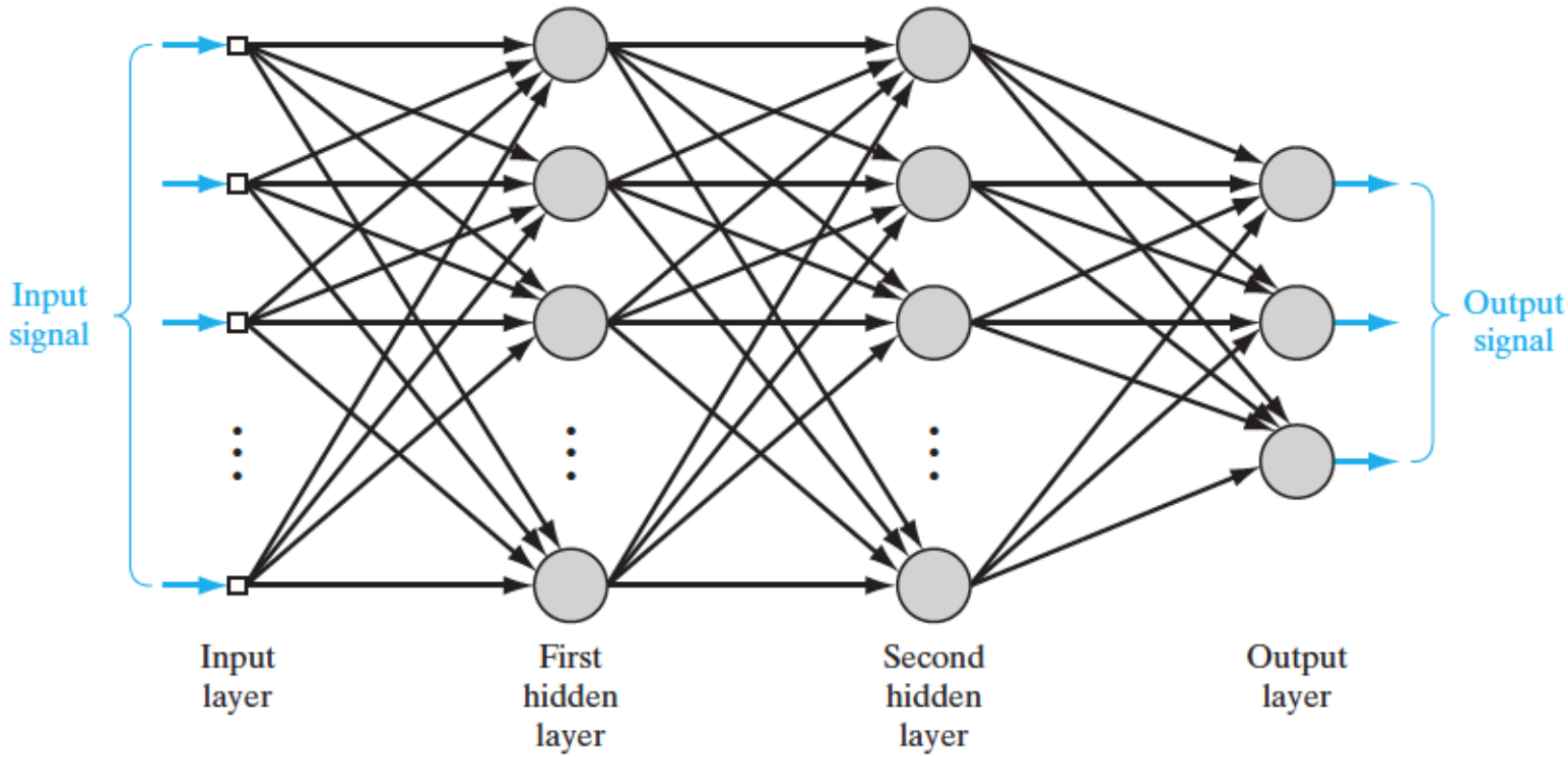


Рис.: Архитектура сети с двумя скрытыми слоями[8]

[8] Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Отличие персептрона Румельхарта от персептрона Розенблатта

- Нелинейная функция активации;
- один и более скрытых слоев (до работ Хинтона по ограниченной машине Больцмана, на практике не использовали более двух скрытых слоев, а чаще всего один);
- сигналы на входе и на выходе не обязательно бинарные;
- произвольная архитектура сети (в рамках многослойности);
- ошибка сети интерпретирует в смысле некоторой меры, а не как число неправильных образов в режиме обучения.

Модифицированная модель нейрона МакКаллока-Питтса

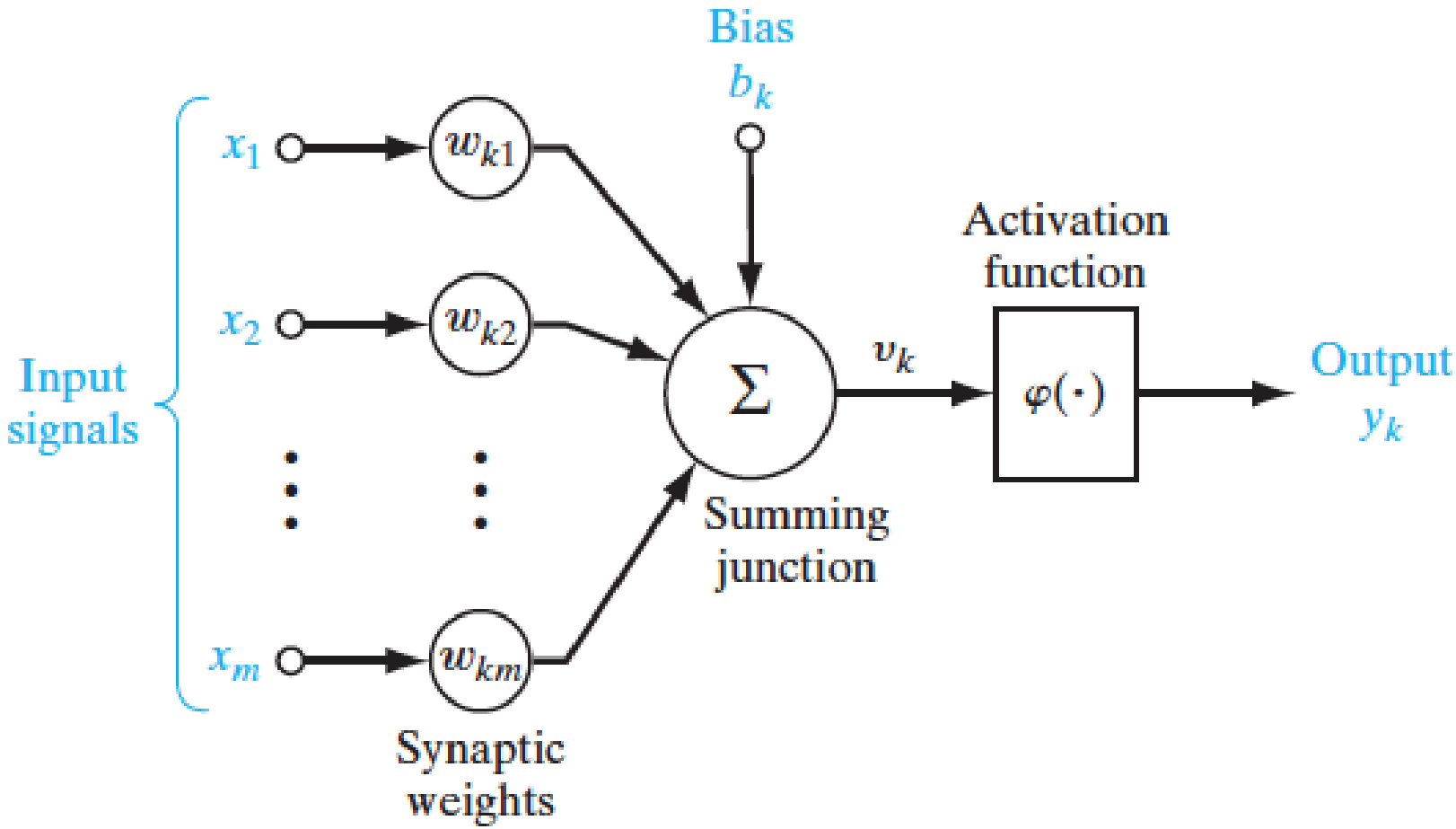


Рис.: Схема искусственного нейрона[9]

[9] Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Функция активации

Задача функции активации - ограничить амплитуду выходного значения нейрона; чаще всего для этого используется одна из сигмоидальных (S-образных) функций:

- логистическая функция: $f(z) = \dfrac{1}{1 + e^{-a \cdot z}}, \forall a \in \mathbb{R}$
- гиперболический тангенс: $f(z) = \dfrac{e^{a \cdot z} - e^{-a \cdot z}}{e^{a \cdot z} + e^{-a \cdot z}}, \forall a \in \mathbb{R}$
- rectifier: $f(z) = \max(0, x) \approx \ln(1 + e^{z})$

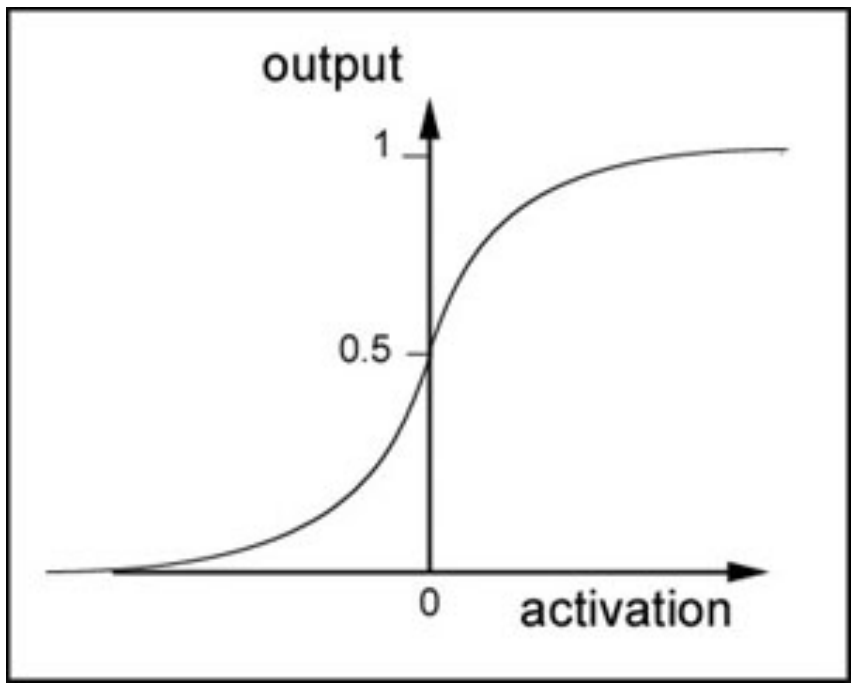
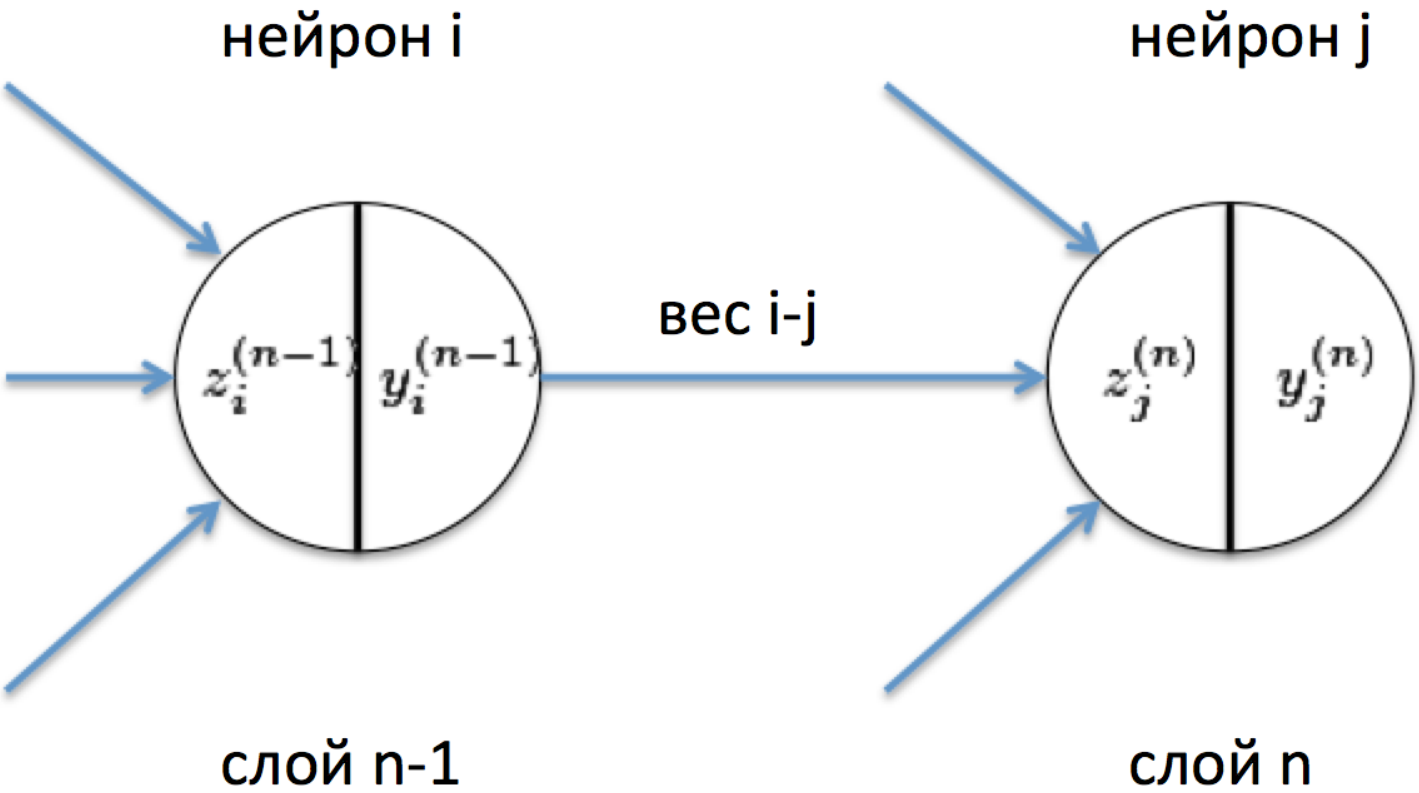


Рис.: Логистический сигмоид

Backprop, обозначения #1

$$z_j^{(n)} = b_j^{(n)} + \sum_{i=1}^{N_{n-1}} w_{ij}^{(n)} x_i^{(n)} = \sum_{i=0}^{N_{n-1}} w_{ij}^{(n)} x_i^{(n)} \quad (1)$$

$$y_k^{(n)} = f_k^{(n)} \left(z_k^{(n)} \right) \quad (2)$$



Backprop, обозначения #2

Обычное обучение с учителем:

- дан набор данных $D = \left\{ \left(x_1, t_1\right), \left(x_2, t_2\right), \ldots, \left(x_{|D|}, t_{|D|}\right) \right\}, x_i \in \mathbb{R}^{N_{\text{INPUT}}}, y_i \in \mathbb{R}^{N_{\text{OUTPUT}}}$
- необходимо построить такое отображение (нейросеть) $f_{\text{NETWORK}} : X \to Y$, которое минимизирует некоторый функционал ошибки $E : \mathbb{R}^{N_{\text{OUTPUT}}} \times \mathbb{R}^{N_{\text{OUTPUT}}} \to \mathbb{R}$, например
  - Евклидово расстояние для задачи регрессии
  - средний логарифм функции правдоподобия распределения Бернулли для задачи классификации среди пересекающихся классов
  - кросс-энтропия для задачи классификации среди *не*пересекающихся классов

Градиентный спуск, #1

Алгоритм backprop - это модификация классического градиентного спуска. Параметрами модели являются только веса всех нейронов сети:

$$\delta_{ij}^{(n)} = -\eta \frac{\partial E\left(\vec{y}^{(n)}, \vec{t}\right)}{\partial w_{ij}^{(n)}} \tag{3}$$

- η - скорость обучения (спуска, learning rate)
- $\vec{y}^{(n)}$ - вектор выходов нейросети (выходы последнего слоя)
- $\vec{t}$ - ожидаемые выходы нейросети для текущего примера

Есть идеи?

Градиентный спуск, #2

- $\frac{\partial E}{\partial w_{ij}^{(n)}} = \frac{\partial E}{\partial z_j^{(n)}} \frac{\partial z_j^{(n)}}{\partial w_{ij}^{(n)}}$
- *???*

Градиентный спуск, #3

- $\dfrac{\partial E}{\partial w_{ij}^{(n)}} = \dfrac{\partial E}{\partial z_j^{(n)}} \dfrac{\partial z_j^{(n)}}{\partial w_{ij}^{(n)}}$
- $\dfrac{\partial z_j^{(n)}}{\partial w_{ij}^{(n)}} = \sum_i \dfrac{\partial w_{ij}^{(n)} x_i^{(n-1)}}{\partial w_{ij}^{(n)}} = x_i^{(n-1)}$

В итоге получим:

$$\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \tag{4}$$

Градиентный спуск, выходной слой, #1

- $\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}}$
- $E\left(y(z), t\right)$???

Градиентный спуск, выходной слой, #2

$$\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial y_j^{(n)}} \frac{\partial y_j^{(n)}}{\partial z_j^{(n)}} \tag{5}$$

Таким образом при условии дифференцируемости целевой функции и функции активации, вычисление градиента любого из весов выходного слоя становится легко решаемой задачей.

Градиентный спуск, любой скрытый слой, #1

$$\begin{aligned}
\frac{\partial E}{\partial w_{ij}^{(n)}} &= x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \\
&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial z_j^{(n)}} \\
&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial y_j^n} \frac{\partial y_j^n}{\partial z_j^n}
\end{aligned}$$

- сумматор следующего слоя зависит только от выходов текущего слоя, а выходы текущего зависят только от сумматоров текущего слоя
- $\frac{\partial z_k^{(n+1)}}{\partial y_j^n}$ *???*

Градиентный спуск, любой скрытый слой, #2

$$\begin{aligned}
\frac{\partial E}{\partial w_{ij}^{(n)}} &= x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \\
&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial z_j^{(n)}} \\
&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial y_j^n} \frac{\partial y_j^n}{\partial z_j^n}
\end{aligned}$$

- сумматор следующего слоя зависит только от выходов текущего слоя, а выходы текущего зависят только от сумматоров текущего слоя
- $\frac{\partial z_k^{(n+1)}}{\partial y_j^n} = \sum_i \frac{\partial w^{(n+1)_{ik}} y_i^n}{\partial y_j^n} = w_{ik}^{(n+1)}$

Градиентный спуск, любой скрытый слой, #3

$$
\begin{aligned}
\frac{\partial E}{\partial w_{ij}^{(n)}} &= x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \\
&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial z_j^{(n)}} \\
&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial y_j^n} \frac{\partial y_j^n}{z_j^n} \\
&= x_i^{(n-1)} \sum_k w_{ik}^{(n+1)} \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial y_j^n}{\partial z_j^n}
\end{aligned}
$$

Обратный проход, #1

Получилась формула градиента для скрытых слоев:

$$\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \sum_k w_{ik}^{(n+1)} \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial y_j^n}{\partial z_j^n}$$

Выполнив замены $\delta_k^{(n)} = \frac{\partial E}{\partial z_k^{(n+1)}}$ и $c_k = x_i^{(n-1)} \frac{\partial y_j^n}{\partial z_j^n} w_{ik}^{(n+1)}$, получим следующее:

$$\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \frac{\partial y_j^n}{\partial z_j^n} \sum_k w_{ik}^{(n+1)} \delta_k^{(n)} = \sum_k c_k \delta_k^{(n)}$$

- в итоге получилась линейная функция от $\delta_k^{(n)}$, c_k - фиксирована после прямого прохода;
- $\delta_k^{(n)}$ - локальный градиент/ошибка нейрона *(она как раз и распространяется обратно)*;
- для текущего слоя n, ошибка $\delta_k^{(n)} = \frac{\partial E}{\partial z_k^{(n+1)}}$ уже подсчитана!

Обратный проход, #2

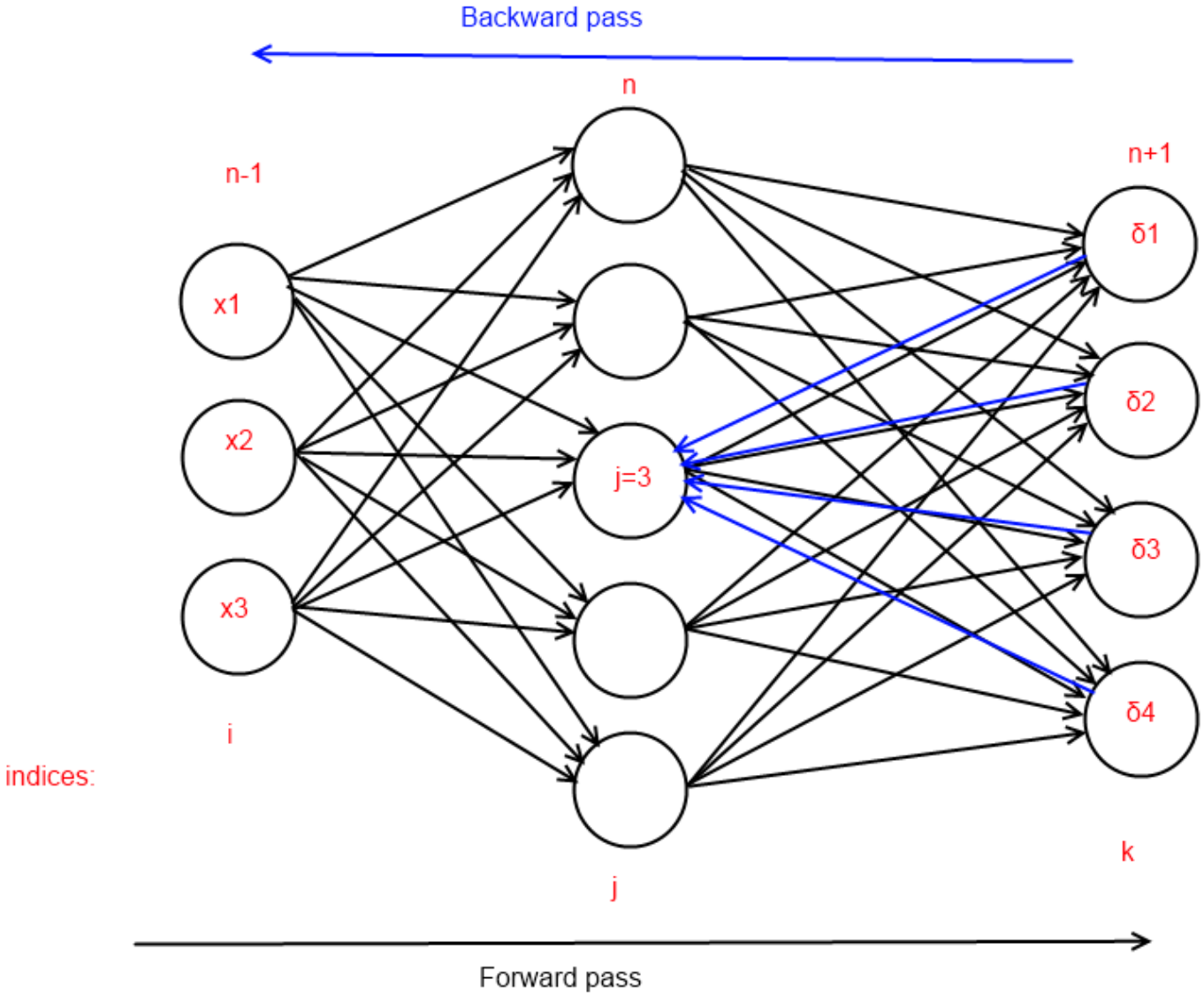


Рис.: Схема прямого (нелинейного) и обратного (линейного) распространения сигнала в сети

Flowing graph

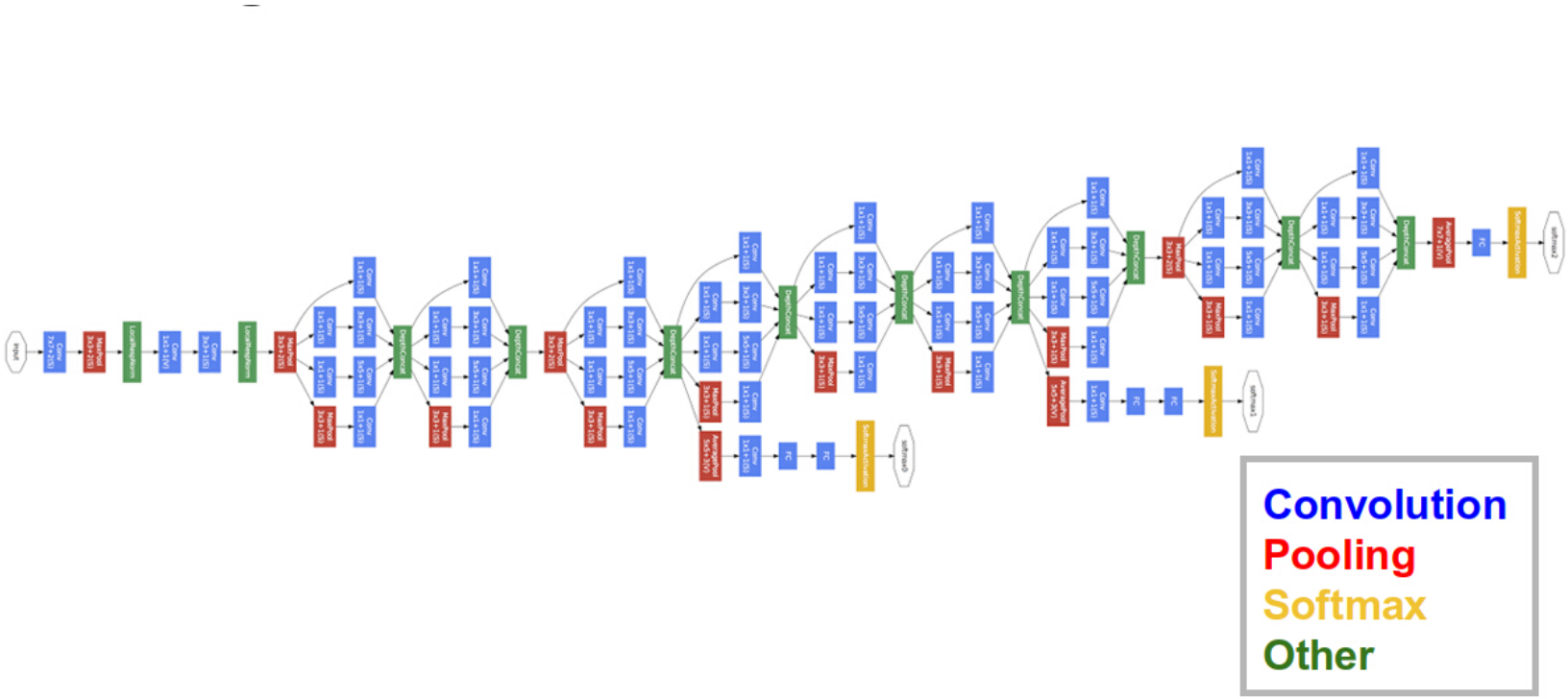


Рис.: Inceptron GoogLeNet[10]

[10] Going Deeper with Convolutions (Christian Szegedy, Wei Liu, ...)

Некоторые функции стоимости, #1

Среднеквадратичная ошибка:

- $E = \frac{1}{2} \sum_{i \in \text{OUTPUT}} (t_i - y_i)^2$
- $\frac{\partial E}{\partial y_i}$???

Логарифм правдоподобия Бернулли:

- $E = -\sum_{i \in \text{OUTPUT}} (t_i \log y_i + (1 - t_i) \log (1 - y_i))$
- $\frac{\partial E}{\partial y_i}$???

Некоторые функции стоимости, #2

Среднеквадратичная ошибка:

- $E = \frac{1}{2} \sum_{i \in \text{OUTPUT}} (t_i - y_i)^2$
- $\frac{\partial E}{\partial y_i} = y_i - t_i$

Логарифм правдоподобия Бернулли:

- $E = -\sum_{i \in \text{OUTPUT}} (t_i \log y_i + (1 - t_i) \log (1 - y_i))$
- $\frac{\partial E}{\partial y_i} = \frac{t_i}{y_i} - \frac{1 - t_i}{1 - y_i}$

Некоторые функции активации, #1

Логистическая функция:

- $f(z) = \dfrac{1}{1 + e^{-a \cdot z}}$
- $\dfrac{\partial f}{\partial z}$???

Гиперболический тангенс:

- $f(z) = \dfrac{e^{a \cdot z} - e^{-a \cdot z}}{e^{a \cdot z} + e^{-a \cdot z}}$
- $\dfrac{\partial f}{\partial z}$???

Некоторые функции активации, #2

Логистическая функция:

- $f(z) = \dfrac{1}{1 + e^{-a \cdot z}}$
- $\dfrac{\partial f}{\partial z} = a \cdot f(z) \cdot (1 - f(x))$

Гиперболический тангенс:

- $f(z) = \dfrac{e^{a \cdot z} - e^{-a \cdot z}}{e^{a \cdot z} + e^{-a \cdot z}}$
- $\dfrac{\partial f}{\partial z} = a \cdot \left(1 - f^2(z)\right)$

Режимы обучения

- online learning
- batch learning
- full-batch learning

Регуляризация в нейронной сети, #1

Что это и зачем?

- $E_R = E\left(\vec{y}, \vec{t}\right) + R(W)$

Примеры L1 и L2 регуляризации:

- $R_{L1}(W) = \sum_{ijn} \left| w_{ij}^{(n)} \right|$
- $\frac{\partial R_{L1}(W)}{\partial w_{ij}^{(n)}}$ *???*
- $R_{L2}(W) = \frac{1}{2} \sum_{ijn} \left(w_{ij}^{(n)} \right)^2$
- $\frac{\partial R_{L2}(W)}{\partial w_{ij}^{(n)}}$ *???*

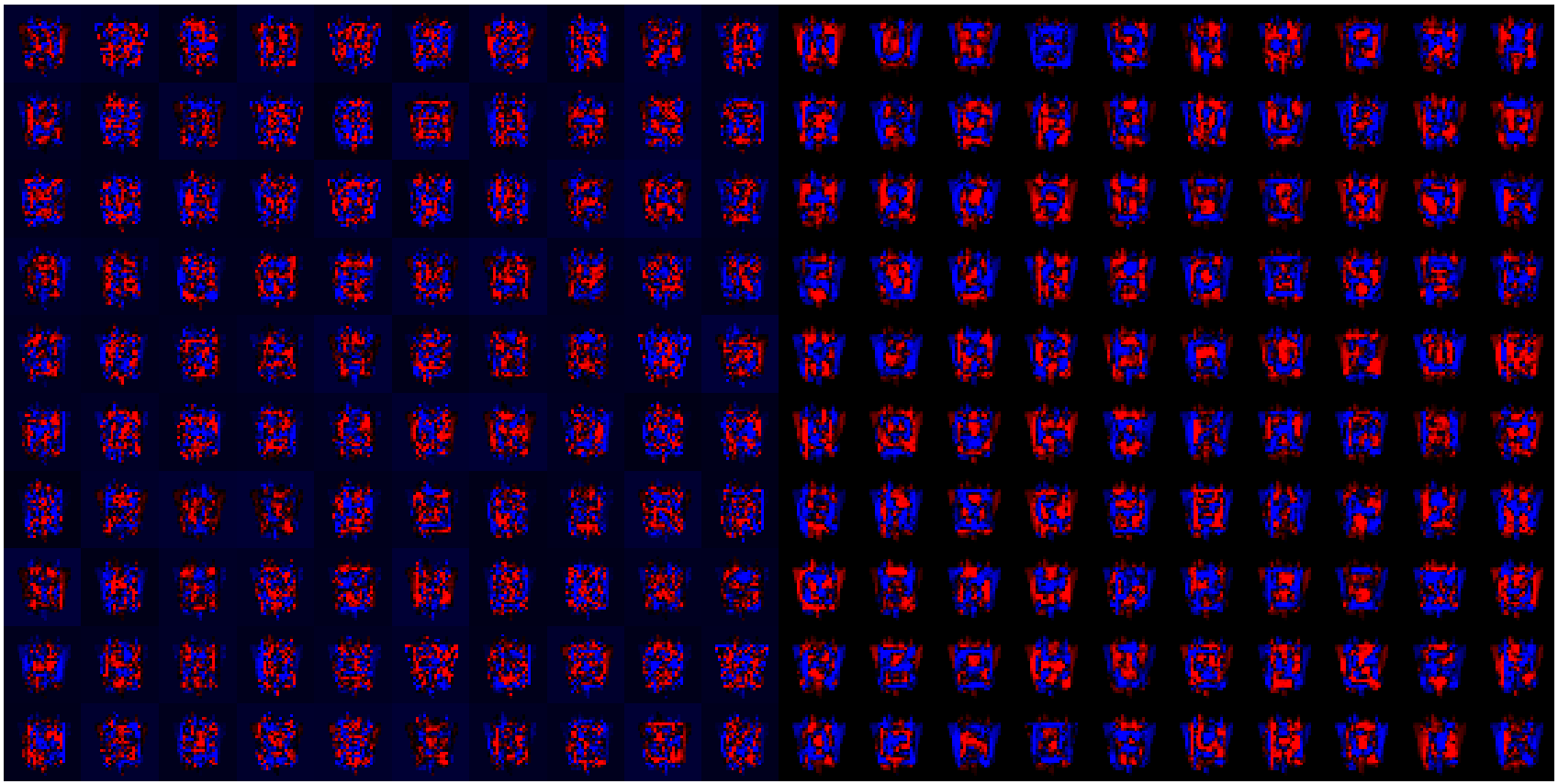
Регуляризация в нейронной сети, #2

- $E_R = E\left(\vec{y}, \vec{t}\right) + \lambda \cdot R(W)$

Примеры L1 и L2 регуляризации:

- $R_{L1}(W) = \sum_{ijn} \left| w_{ij}^{(n)} \right|$
- $\frac{\partial R_{L1}(W)}{\partial w_{ij}^{(n)}} = \mathrm{SIGN}\left(w_{ij}^{(n)}\right)$
- $R_{L2}(W) = \frac{1}{2} \sum_{ijn} \left(w_{ij}^{(n)}\right)^2$
- $\frac{\partial R_{L2}(W)}{\partial w_{ij}^{(n)}} = w_{ij}^{(n)}$

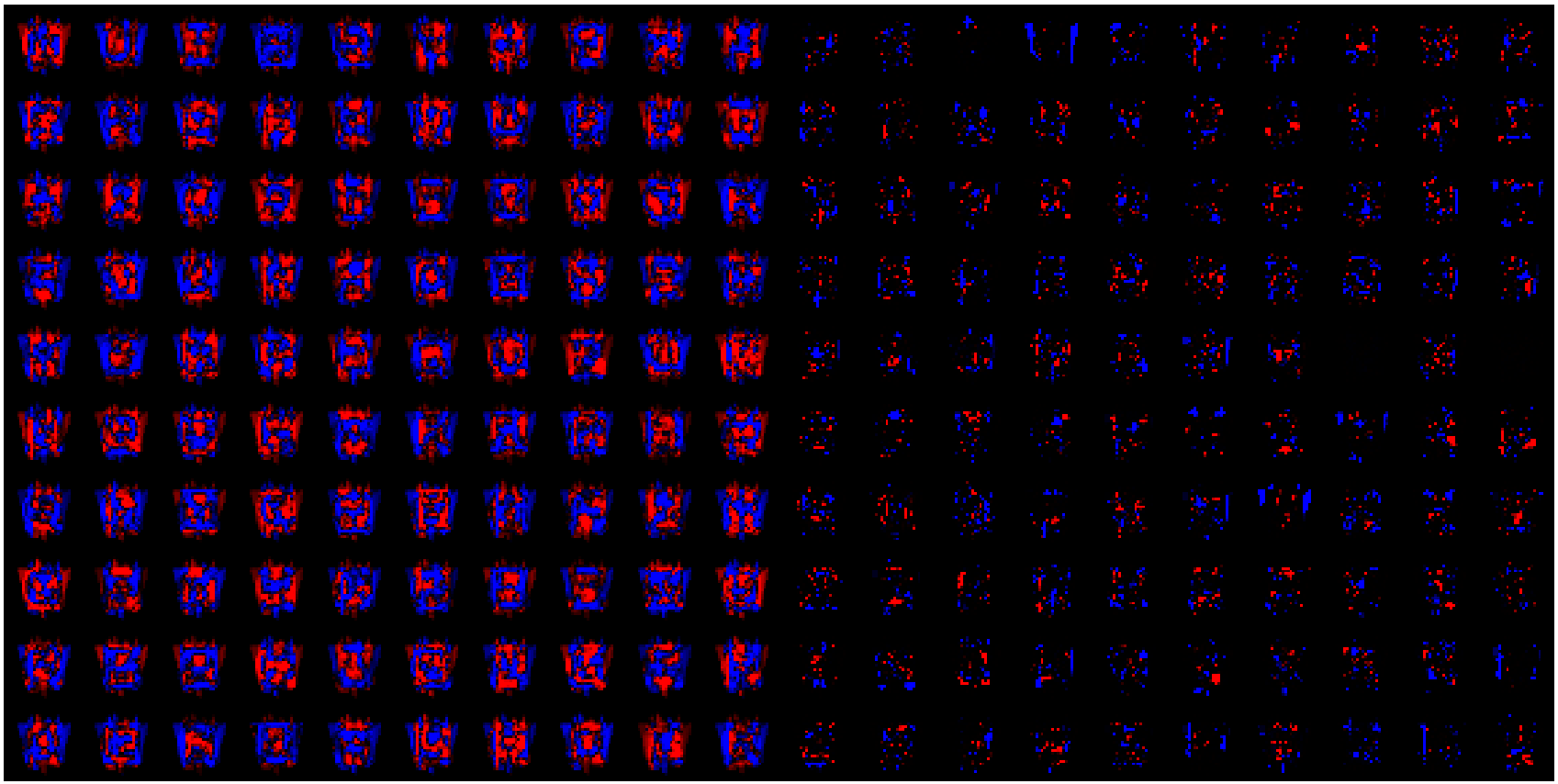
Регуляризация в нейронной сети, #2



(a) RBM, no reg (b) RBM, L2 reg

Рис.: Иллюстрация эффекта регуляризации

Регуляризация в нейронной сети, #3



(a) RBM, L2 reg

(b) RBM, L1 reg

Рис.: Иллюстрация эффекта регуляризации

Критерий остановки

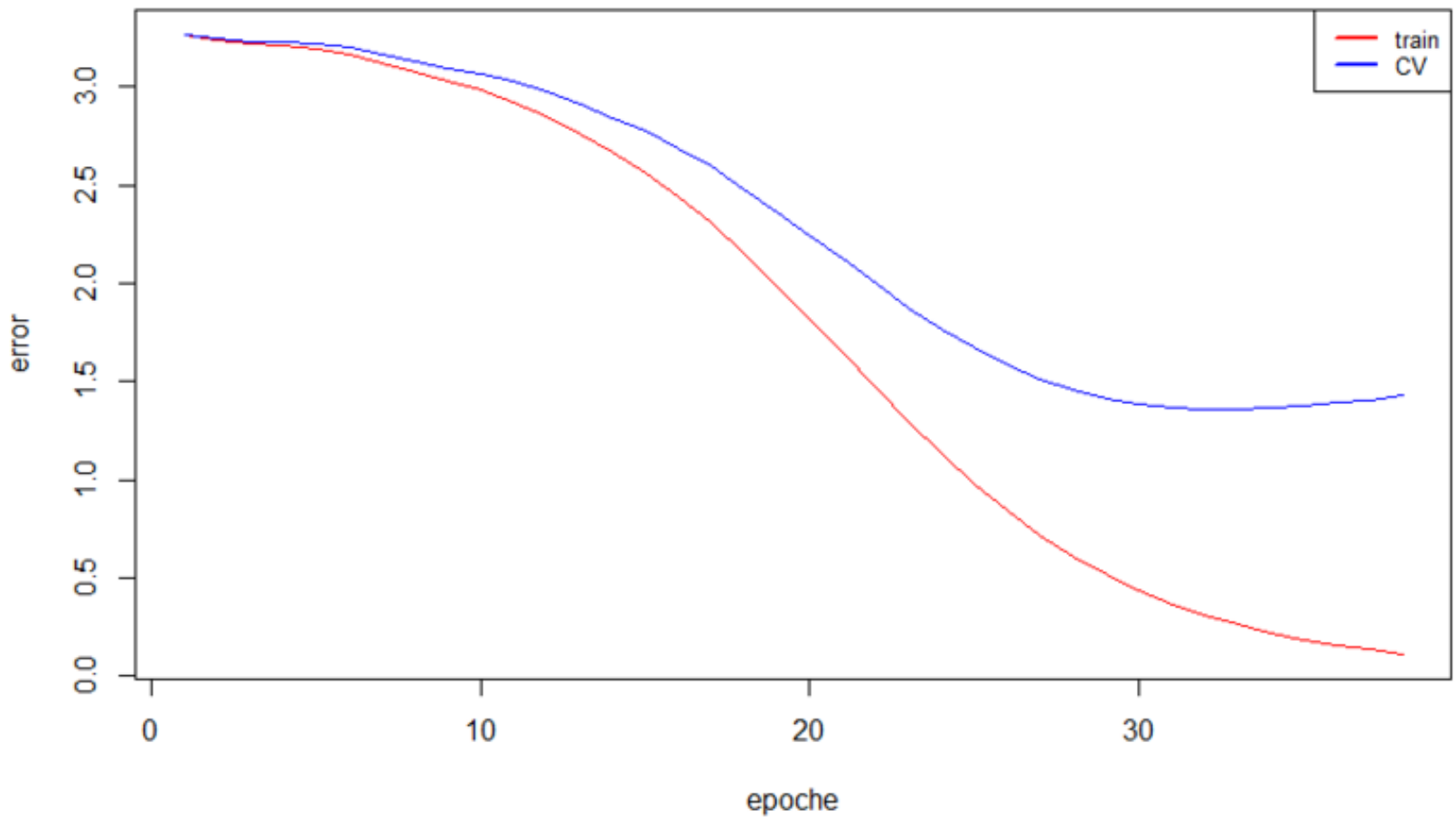


Рис.: Кроссвалидация

Ускорение сходимости

Добавление момента обучения:

$$\Delta w_{ij}(\tau) = \eta \left(\mu \Delta w_{ij} \left(\tau - 1 \right) + \nabla w_{ij} \right) \tag{6}$$

Локальная скорость обучения:

$$\delta_{ij}^{(n)} = -\eta \cdot r_{ij}^{(n)} \cdot (\cdots) \tag{7}$$

$$r_{ij}^{(n)} = \begin{cases} r_{ij}^{(n)} = b + r_{ij}^{(n)}, \nabla w_{ij}^{(n)}(\tau - 1) \cdot \nabla w_{ij}^{(n)}(\tau) > 0 \\ r_{ij}^{(n)} = p \cdot r_{ij}^{(n)} \end{cases} \tag{8}$$

где

- b - аддитивный бонус
- p - мультпликативный штраф
- $b + p = 1$
- естьсмысл добавить верхнюю и нижнюю границы для значения $r_{ij}^{(n)}$

Планы

- softmax слой в сети прямого распространения
- обучение без учителя;
- стохастический нейрон и стохастическая нейросеть;
- ограниченная машина Больцмана.
- глубокие сети

Вопросы