Методы машинного обучения
Метод стохастического градиента и линейные модели классификации

Воронцов Константин Вячеславович
www.MachineLearning.ru/wiki?title=User:Vokov
вопросы к лектору: k.vorontsov@iai.msu.ru

материалы курса:
github.com/MSU-ML-COURSE/ML-COURSE-25-26
орг.вопросы по курсу: ml.cmc@mail.ru

ВМК МГУ • 23 сентября 2025

Содержание

Общая постановка задачи обучения по прецедентам

Дано: X — пространство *объектов*
$X^\ell = \{x_1, \ldots, x_\ell\} \subset X$ — *обучающая выборка*, прецеденты
$a(x, w)$, $a\colon X \times W \to Y$ — параметрическая модель, гипотеза

Найти $w \in W$ — вектор параметров модели $a(x, w)$

Критерий ERM (Empirical Risk Minimization) — минимизация эмпирического риска, в общем случае с регуляризацией:

$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(w, x_i) + \tau \mathscr{R}(w) \to \min_w$$

$\mathscr{L}(w, x)$ — *функция потерь* (loss function),
тем больше, чем хуже ответ модели $a(x, w)$ на объекте x

$\mathscr{R}(w)$ — *регуляризатор*, априорные требования к модели

τ — *коэффициент регуляризации* для балансировки критериев

Функции потерь $\mathscr{L}(w, x)$ в задачах обучения с учителем

Дано: X^ℓ, каждому объекту x_i соответствует ответ $y_i = y(x_i)$

Задача регрессии: $y_i \in \mathbb{R}$

Найти: модель регрессии $a(x, w)$, вектор параметров w

Критерий ERM: $$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} L\Big(\underbrace{a(x_i, w) - y_i}_{\varepsilon_i(w)\text{ — error, ошибка}}\Big) \to \min_w,$$

где $L(\varepsilon)$ унимодальная: ε^2, $|\varepsilon|$, $\big(|\varepsilon| - c\big)_+$, и др.

Задача классификации с двумя классами: $y_i \in \{-1, +1\}$

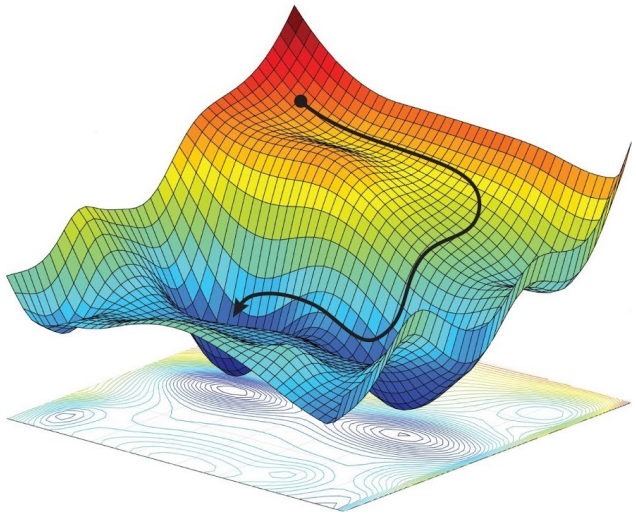
Найти: модель классификации $a(x, w) = \operatorname{sign} g(x, w)$

Критерий ERM: $$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} L\Big(\underbrace{g(x_i, w)y_i}_{M_i(w)\text{ — margin, отступ}}\Big) \to \min_w,$$

где $L(M)$ невозрастающая: $\ln(1 + e^{-M})$, $(1 - M)_+$, e^{-M}, и др.

Градиентный метод минимизации эмпирического риска

$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}(w, x_i) + \tau\mathscr{R}(w) \to \min_w$$



Метод *градиентного спуска*:

$w^{(0)} :=$ начальное приближение;

$w^{(t+1)} := w^{(t)} - h\nabla Q(w^{(t)})$

где $\nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=1}^{N}$ — *вектор градиента*,

h — *градиентный шаг*, называемый также *темпом обучения*

$$w^{(t+1)} := w^{(t)} - h\left(\frac{1}{\ell}\sum_{i=1}^{\ell}\nabla\mathscr{L}\left(w^{(t)}, x_i\right) + \tau\nabla\mathscr{R}\left(w^{(t)}\right)\right)$$

Идея ускорения сходимости: брать случайное подмножество слагаемых, или даже один объект, сразу обновляя вектор весов

Метод стохастического градиента SG (Stochastic Gradient)

Критерий ERM: $Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}(w, x_i) + \tau\mathscr{R}(w) \to \min_w$

Вход: выборка X^ℓ, параметры h, τ, λ;

Выход: вектор весов w;

инициализировать веса w_j, $j = 1, \ldots, N$;
инициализировать оценку $Q(w)$ по небольшой подвыборке;
повторять
    объект x_i выбрать из X^ℓ случайным образом;
    потеря: $\mathscr{L}_i := \mathscr{L}(w, x_i)$;
    градиентный шаг: $w := w - h\nabla\mathscr{L}(w, x_i) - h\tau\nabla\mathscr{R}(w)$;
    рекуррентная оценка критерия: $Q := \lambda\mathscr{L}_i + (1-\lambda)Q$;
пока значение Q и/или веса w не сойдутся;

H. Robbins, S. Monro. A stochastic approximation method. 1951.

Откуда взялась такая рекуррентная оценка функционала?

Проблема: вычисление оценки Q по всей выборке $x_1, \ldots, x_\ell$ намного дольше градиентного шага по одному объекту x_i.

Решение: использовать приближённую рекуррентную формулу.

Среднее арифметическое:

$$\bar{Q}_m = \tfrac{1}{m}\mathscr{L}_m + \tfrac{1}{m}\mathscr{L}_{m-1} + \tfrac{1}{m}\mathscr{L}_{m-2} + \ldots$$

$$\bar{Q}_m = \tfrac{1}{m}\mathscr{L}_m + (1 - \tfrac{1}{m})\bar{Q}_{m-1}$$

Экспоненциальное скользящее среднее (ЭСС):

$$\bar{Q}_m = \lambda\mathscr{L}_m + (1-\lambda)\lambda\mathscr{L}_{m-1} + (1-\lambda)^2\lambda\mathscr{L}_{m-2} + \ldots$$

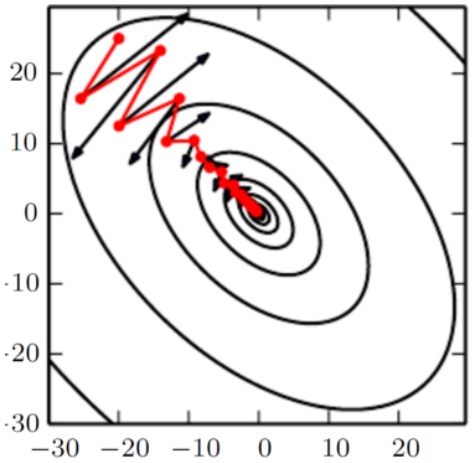
$$\bar{Q}_m = \lambda\mathscr{L}_m + (1 - \lambda)\bar{Q}_{m-1}$$

Параметр λ (порядка $\frac{1}{m}$) — *темп забывания* предыстории ряда.

Метод накопления инерции (momentum)

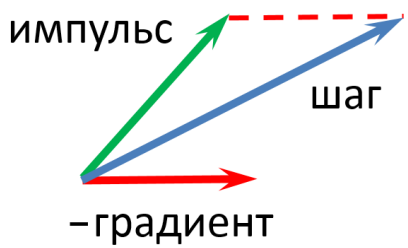
Momentum — экспоненциальное скользящее среднее градиента по последним $\approx \frac{1}{1-\gamma}$ итерациям [Б.Т.Поляк, 1964]:

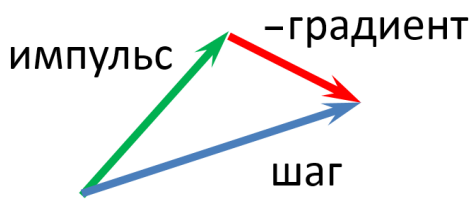
$$v := \gamma v + (1-\gamma)\nabla \mathscr{L}(w, x_i)$$
$$w := w - hv$$



NAG (Nesterov's accelerated gradient) — стохастический градиент с инерцией [Ю.Е.Нестеров, 1983]:

$$v := \gamma v + (1-\gamma)\nabla \mathscr{L}(w - h\gamma v, x_i)$$
$$w := w - hv$$





Эвристики для перебора объектов, ускоряющие сходимость

1. *перетасовка объектов* (shuffling): выбирать объекты x_i
   — попеременно из разных классов
   — самые далёкие из нескольких случайных объектов
2. пропускать «слишком хорошие» объекты
   — у которых $\mathscr{L}_i < \mathscr{L}_{\min}$
   — с вероятностью тем больше, чем меньше $\mathscr{L}_i - \mathscr{L}_{\min}$
   (при этом не тратится время на мелкие улучшения)
3. пропускать «слишком плохие» объекты, «выбросы»
   — у которых $\mathscr{L}_i > \mathscr{L}_{\max}$
   — с вероятностью тем больше, чем больше $\mathscr{L}_i - \mathscr{L}_{\max}$
   (при этом может улучшиться качество классификации)

Возможно, параметры $\mathscr{L}_{\min}$, $\mathscr{L}_{\max}$ придётся подбирать.

Эвристики для выбора градиентного шага

1. сходимость гарантируется (для выпуклых функций) при

$$h_t \to 0, \quad \sum_{t=1}^{\infty} h_t = \infty, \quad \sum_{t=1}^{\infty} h_t^2 < \infty,$$

   в частности, можно положить $h_t = 1/t$
2. *метод скорейшего градиентного спуска* основан на поиске оптимального *адаптивного шага* h^*:

$$\mathscr{L}\big(w - h\nabla\mathscr{L}(w, x_i), x_i\big) \to \min_h$$

   в частности, при квадратичной функции потерь $h^* = \|x_i\|^{-2}$
3. пробные случайные шаги для «выбивания» итерационного процесса из локальных экстремумов, с выбором лучшего
4. метод Левенберга-Марквардта (сходимость второго порядка)

Диагональный метод Левенберга-Марквардта

Метод Ньютона-Рафсона:

$$w := w - h\left(\mathscr{L}''(w, x_i)\right)^{-1}\nabla\mathscr{L}(w, x_i),$$

где $\mathscr{L}''(w, x_i) = \left(\frac{\partial^2\mathscr{L}(w,x_i)}{\partial w_j \partial w_{j'}}\right)$ — гессиан, $n\times n$-матрица

Эвристика. Считаем, что гессиан диагонален:

$$w_j := w_j - h\max\left\{\mu, \frac{\partial^2\mathscr{L}(w, x_i)}{\partial w_j^2}\right\}^{-1}\frac{\partial\mathscr{L}(w, x_i)}{\partial w_j},$$

$h > 0$ — темп обучения, можно полагать $h = 1$,
$\mu > 0$ — параметр, предотвращающий обнуление знаменателя.

Вблизи минимума сходимость второго порядка с темпом h
Вдали от минимума сходимость первого порядка с темпом $\frac{h}{\mu}$

Эвристики для инициализации весов

1. обучение очень простым и быстрым методом
2. обучение по небольшой случайной подвыборке объектов
3. мультистарт: многократные запуски из разных случайных начальных приближений и выбор лучшего решения

Эвристики для линейных моделей $a(x,w) = \sum_{j=1}^{n} w_j f_j(x) = \langle w, x \rangle$

4. $w_j := 0$ для всех $j = 0, \ldots, n$ — плохой вариант
5. $w_j := \text{random}\left(-\frac{1}{2n}, \frac{1}{2n}\right)$ — случайные значения вблизи нуля
6. $w_j := \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}$, где $f_j = \left(f_j(x_i)\right)_{i=1}^{\ell} \in \mathbb{R}^{\ell}$, $y = \left(y(x_i)\right)_{i=1}^{\ell} \in \mathbb{R}^{\ell}$

   Эта оценка w оптимальна для задачи регрессии, когда
   1) функция потерь квадратична и
   2) признаки некоррелированы, $\langle f_j, f_k \rangle = 0$, $j \neq k$

Метод SG: Достоинства и недостатки

Достоинства:

1. *простота:* относительно легко реализуется
2. *универсальность:* для любых $a(x, w)$, $\mathscr{L}(w, x)$, $\mathscr{R}(w)$
3. *поточность:* возможность обучения на потоке данных
4. *подходит для обработки больших данных:*
   — можно получить неплохое решение, успев обработать лишь малую часть обучающей выборки
   — часто оказывается быстрее и лучше более сложных и ресурсоёмких методов второго порядка

Недостатки:

1. подбор комплекса эвристик является искусством (не забыть про переобучение, застревание, расходимость)

Задача бинарной классификации

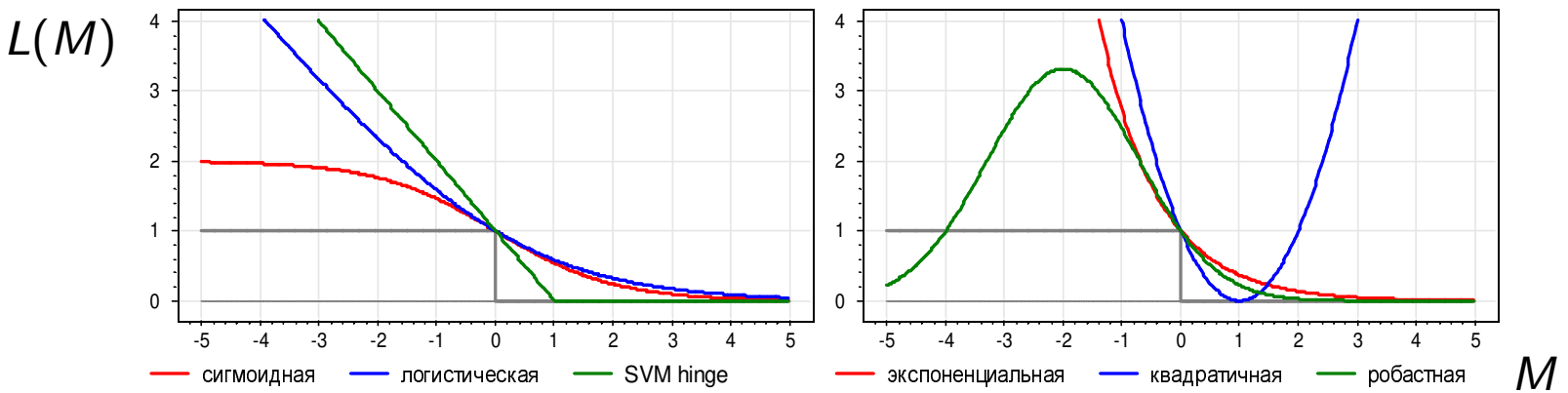
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \{-1, +1\}$

Найти: вектор w модели классификации $a(x, w) = \text{sign}\, g(x, w)$

Критерий ERM, *аппроксимированного эмпирического риска:*

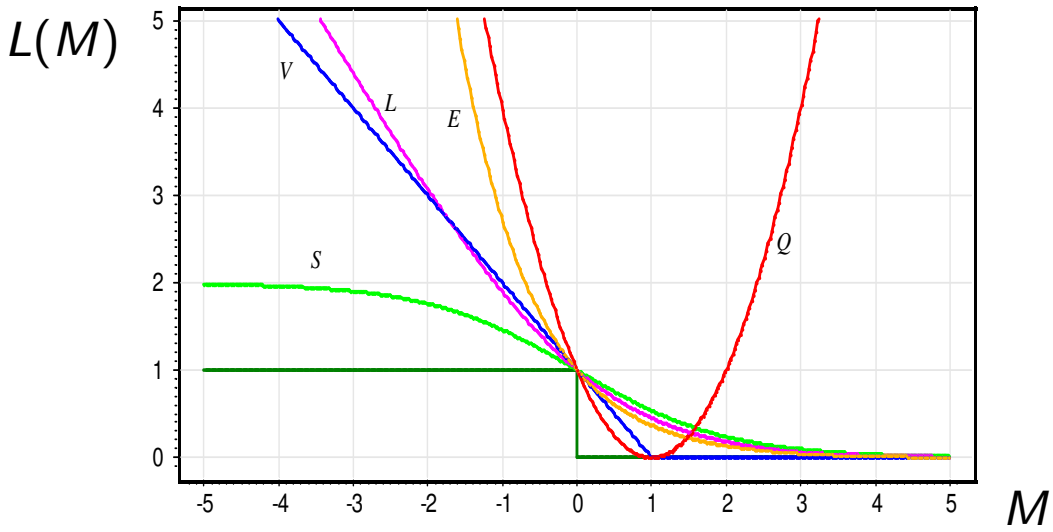
$$\sum_{i=1}^{\ell} \big[g(x_i, w)y_i < 0\big] \leqslant \sum_{i=1}^{\ell} L\big(g(x_i, w)y_i\big) \;\to\; \min_w$$

где $g(x_i, w)y_i = M_i$ — *отступ* или *зазор* (margin) между объектом x_i и границей классов $\{x\colon g(x, w) = 0\}$,
$[M < 0] \leqslant L(M)$ — функция, как правило, невозрастающая



Непрерывные аппроксимации пороговой функции потерь

Выбор функции потерь $L(M)$ порождает большое разнообразие методов, отличающихся полезными свойствами (это спойлер)



$[M < 0]$	пороговая функция потерь
$V = (1 - M)_+$	кусочно-линейная (метод опорных векторов, SVM)
$H = (-M)_+$	кусочно-линейная (правило Хэбба, Hebb's rule)
$L = \log_2(1 + e^{-M})$	логарифмическая (логистическая регрессия, LR)
$Q = (1 - M)^2$	квадратичная (линейный дискриминант Фишера, FLD)
$S = 2(1 + e^{M})^{-1}$	сигмоидная (искусственный нейронные сети, ANN)
$E = e^{-M}$	экспоненциальная (линейный ансамбль AdaBoost)

Бинарный разделяющий классификатор (margin-based classifier)

Бинарный классификатор: $a(x, w) = \text{sign}\, g(x, w), \quad Y = \{-1, +1\}$

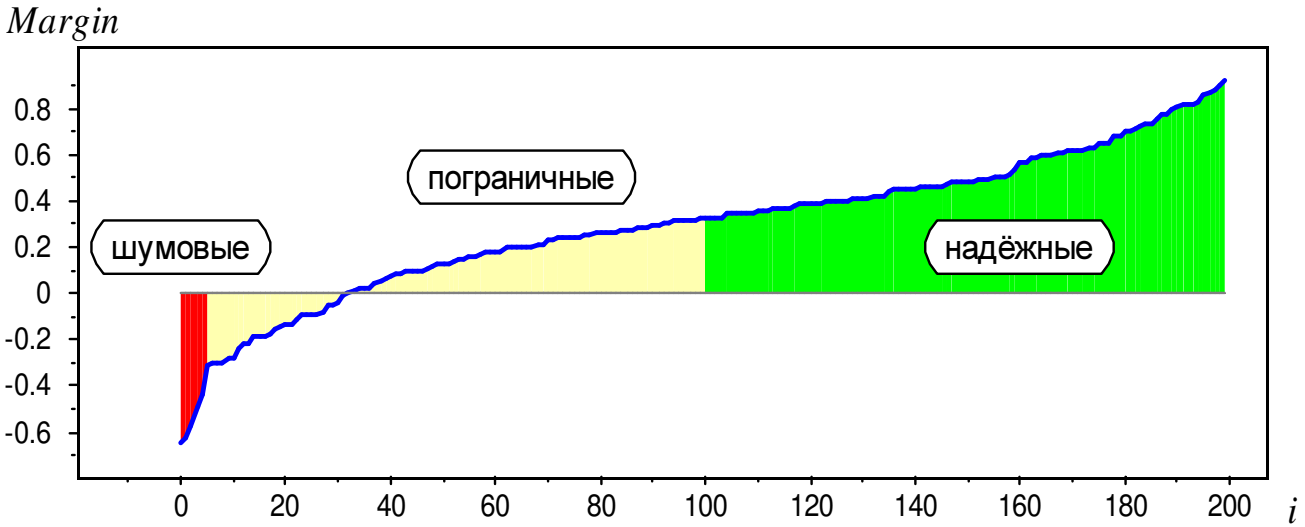
$g(x, w)$ — *разделяющая (дискриминантная) функция*

$\{x : g(x, w) = 0\}$ — разделяющая поверхность между классами

$M_i(w) = g(x_i, w)y_i$ — *отступ* (margin) объекта x_i

$M_i(w) < 0 \iff$ модель $a(x, w)$ ошибается на x_i

Ранжирование объектов по возрастанию отступов $M_i(w)$:



Задача многоклассовой классификации (multiclass classification)

Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in Y$, $|Y| < \infty$

Найти: вектор $w = (w_y : y \in Y)$ модели классификации

$$a(x, w) = \arg\max_{y \in Y} g_y(x, w_y),$$

где $g_y(x, w_y)$ отделяет объекты класса y от всех остальных,
$M_{iy}(w) = g_{y_i}(x_i, w_{y_i}) - g_y(x_i, w_y)$ — *отступ* объекта x_i по классу y

Критерий ERM «каждый против всех» (one-vs.-all, OvA):

$$Q(w) = \sum_{i=1}^{\ell} \sum_{y \neq y_i} \big[M_{iy}(w) < 0\big] \leqslant \sum_{i=1}^{\ell} \sum_{y \neq y_i} L\big(M_{iy}(w)\big) \to \min_w,$$

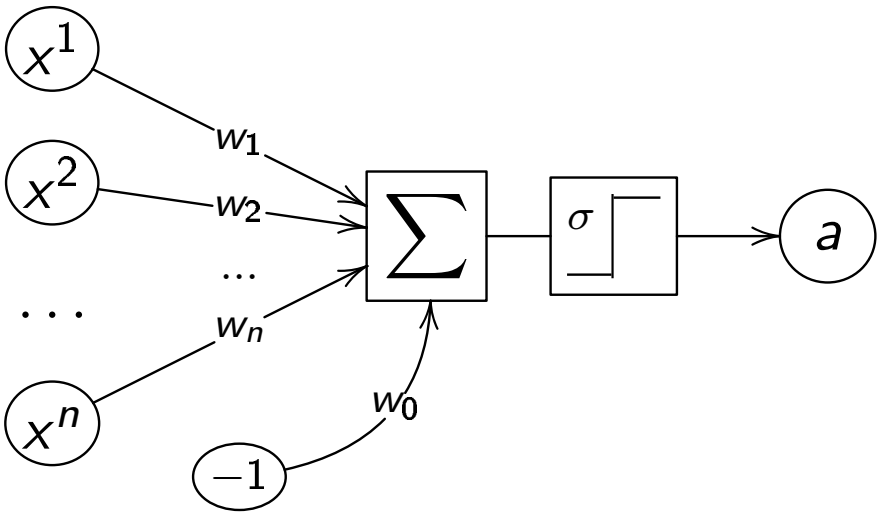
где функция $L(M)$ выбирается, как правило, невозрастающей

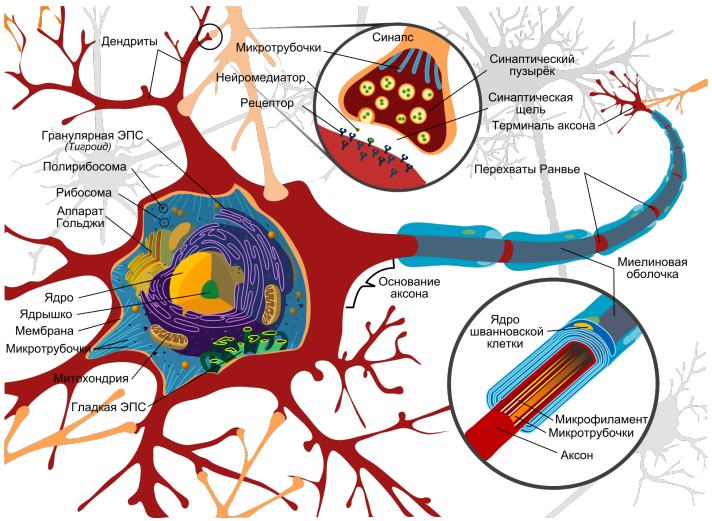
Линейный классификатор — математическая модель нейрона

Линейная модель нейрона МакКаллока-Питтса [1943]:

$$a(x,w) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right) = \sigma\big(\langle w, x\rangle - w_0\big),$$

$\sigma(z)$ — функция активации (например, sign, th или $\frac{1}{1+e^{-z}}$),
w_j — весовые коэффициенты синаптических связей,
w_0 — порог активации, он же вес признака $f_0(x) \equiv -1$





Мультиколлинеарность и переобучение в линейных моделях

Причины — те же, что и в линейных моделях регрессии:

- признаки линейно зависимы (мультиколлинеарны):
  если $\exists u \in \mathbb{R}^n$: $\forall x_i \in X^\ell \ \langle u, x_i \rangle = 0$, то решение
  не единственно и не устойчиво: $\forall \gamma \in \mathbb{R} \ \langle w + \gamma u, x_i \rangle = \langle w, x_i \rangle$
- в частности, если $\ell < n$ (объектов меньше, чем признаков)

Проявления мультиколлинеарности:

- слишком большие веса $|w_j|$ разных знаков
- вес w_j не интерпретируется как важность признака f_j
- переобучение: $Q(w^*, X^\ell) \ll Q(w^*, X^k)$

Способы устранения мультиколлинеарности и переобучения:

1. регуляризация: $\|w\| \to \min$;
2. отбор признаков: $f_1, \ldots, f_n \to f_{j_1}, \ldots, f_{j_m}, \ \ m \ll n$.
3. преобразование признаков: $f_1, \ldots, f_n \to g_1, \ldots, g_m, \ \ m \ll n$;

L_2-регуляризация (сокращение весов, weight decay)

Штраф за увеличение нормы вектора весов:

$$\widetilde{\mathscr{L}}(w, x_i) = \mathscr{L}(w, x_i) + \frac{\tau}{2}\|w\|^2 = \mathscr{L}(w, x_i) + \frac{\tau}{2}\sum_{j=1}^{n} w_j^2 \to \min_w .$$

Градиент:

$$\nabla\widetilde{\mathscr{L}}(w, x_i) = \nabla\mathscr{L}(w, x_i) + \tau w.$$

Модификация градиентного шага:

$$w := w(1 - h\tau) - h\nabla\mathscr{L}(w, x_i).$$

Методы подбора коэффициента регуляризации τ:

- hold-out или скользящий контроль
- стохастическая адаптация по сетке значений $\{\tau_k\}$

Негладкие регуляризаторы для отбора и группировки признаков

Общий вид регуляризаторов (μ — параметр селективности):

$$\sum_{i=1}^{\ell} \mathscr{L}(w, x_i) + \sum_{j=1}^{n} R_\mu(w_j) \;\to\; \min_w .$$

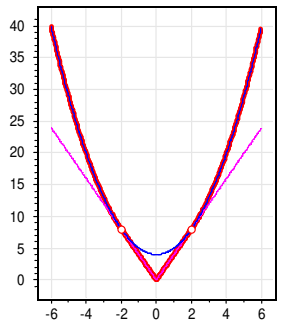
Регуляризаторы с эффектами отбора и группировки признаков:

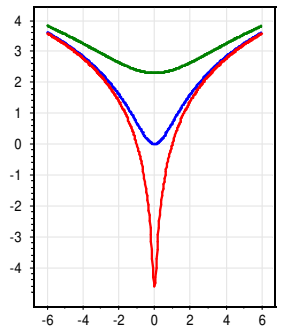
LASSO (L_1): $R_\mu(w) = \mu|w|$

Elastic Net: $R_\mu(w) = \mu|w| + \tau w^2$

Support Feature Machine (SFM):

$$R_\mu(w) = \begin{cases} 2\mu|w|, & |w| \leqslant \mu; \\ \mu^2 + w^2, & |w| \geqslant \mu; \end{cases}$$



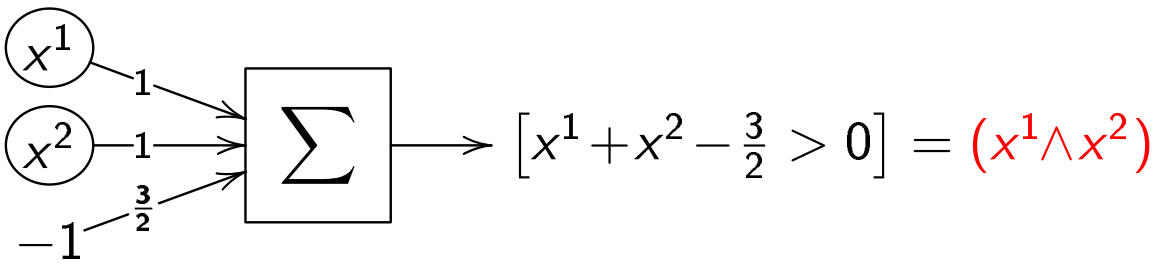


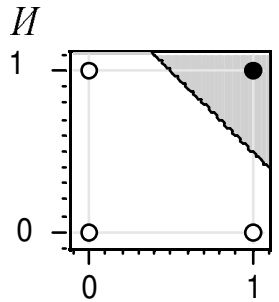
Relevance Feature Machine (RFM):

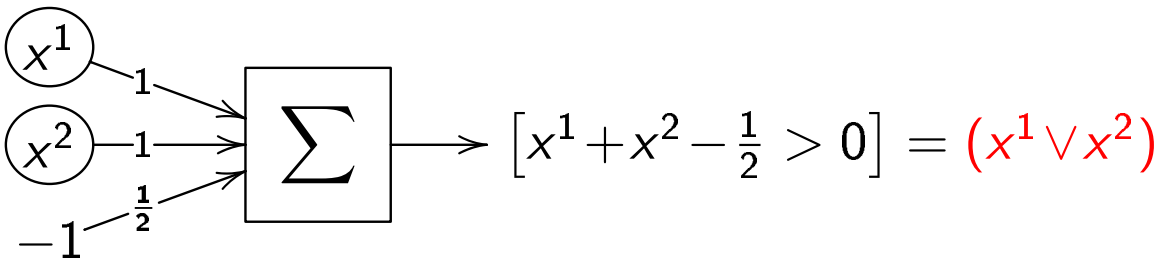
$$R_\mu(w) = \ln\left(\mu w^2 + 1\right)$$

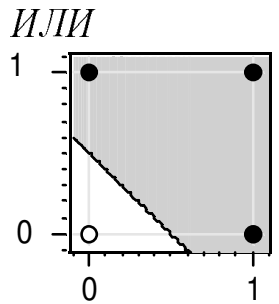
Реализация булевых функций линейным нейроном

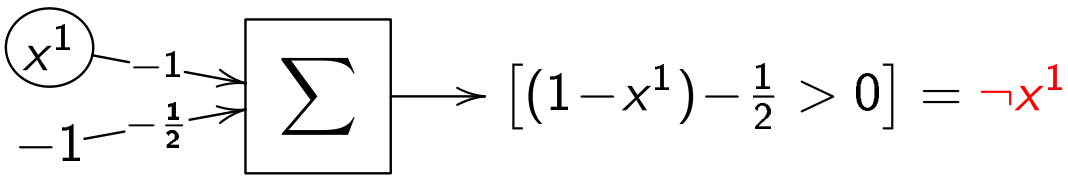
Функции И, ИЛИ, НЕ бинарных признаков x^1, x^2:

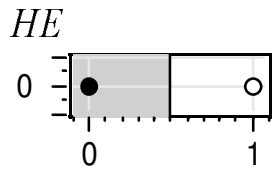












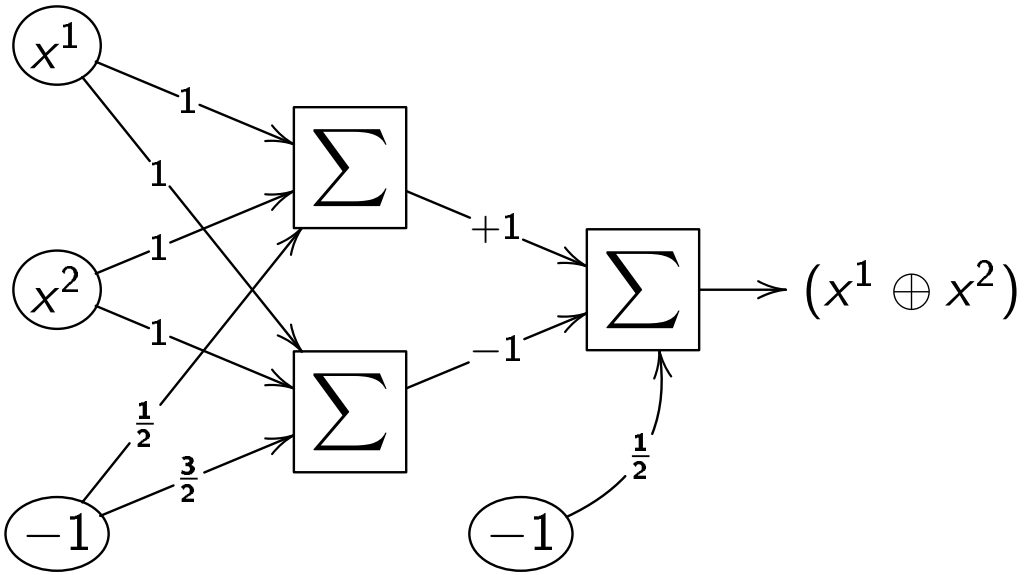
Дизъюнктивная нормальная форма $\Longleftrightarrow$ двухслойная нейросеть

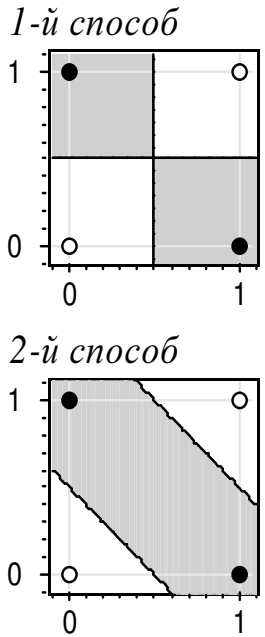
Ограничение линейных моделей: функция XOR

Функция $x^1 \oplus x^2 = [x^1 \neq x^2]$ не реализуема одним нейроном.

Два способа реализации:

- Добавлением нелинейного признака (feature generation):
  $x^1 \oplus x^2 = \left[x^1 + x^2 - 2x^1x^2 - \frac{1}{2} > 0\right]$;
- Сетью (двухслойной суперпозицией) функций И, ИЛИ, НЕ:
  $x^1 \oplus x^2 = \left[(x^1 \vee x^2) - (x^1 \wedge x^2) - \frac{1}{2} > 0\right]$.





Персептрон Розенблатта (1957)

Mark-1 — первый нейрокомпьютер (1960)
для распознавания цифр и фигур
Обучение — метод коррекции ошибки
Архитектура — двухслойная сеть

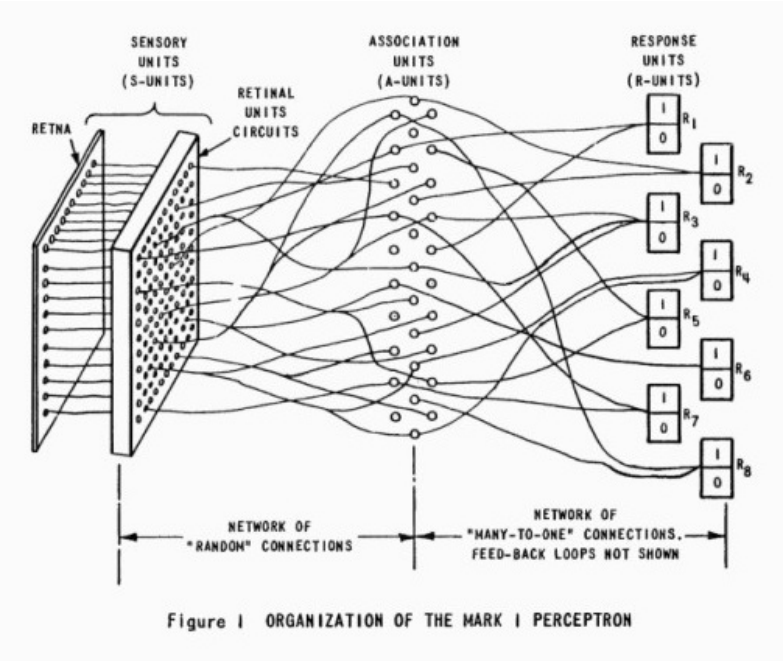
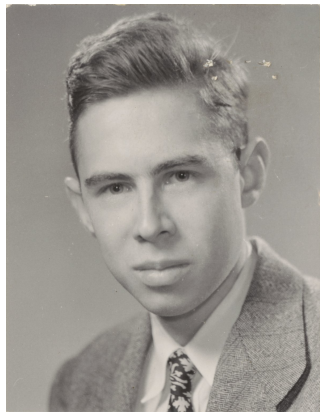


Figure 1 ORGANIZATION OF THE MARK I PERCEPTRON



Фрэнк Розенблатт
(1928–1971)

Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга. 1965 (1962)

Любую ли функцию можно представить нейросетью?

Решение тринадцатой (из 23) проблем Гильберта (1900):

Теорема [Колмогоров, 1956; Арнольд, 1957]

Любая непрерывная функция n аргументов на единичном кубе $[0,1]^n$ представима в виде суперпозиции непрерывных функций одного аргумента и операции сложения:

$$f(x_1,\ldots,x_n) = \sum_{k=1}^{2n+1} \Phi_k\left(\sum_{j=1}^{n} \varphi_{jk}(x_j)\right),$$

где Φ_k, φ_{jk} — непрерывные функции, и φ_{jk} не зависят от f.

Имеет ли теорема Колмогорова отношение к нейросетям?

А.Н.Колмогоров. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных. 1956.

В.И.Арнольд. О функции трех переменных. 1957.

Имеет ли теорема Колмогорова отношение к нейросетям?

Вроде да:

- структура суперпозиции соответствует двухслойной сети
- имеются универсальные аппроксимационные свойства

На самом деле — нет:

- это точное представление; нам достаточно аппроксимации
- функции Φ_k, φ_{jk} не гладкие и сложно строятся
- нет ни весов W, ни оптимизационной задачи обучения
- число слоёв 2 и число нейронов $[2n+1, n]$ фиксированы

Но можно обобщить конструкцию эвристически (KAN):

- любое число слоёв, любая ширина слоёв
- вместо функций Φ_k, φ_{jk} — одномерные сплайны (с весами)
- использовать стандартные методы обучения (BackProp)

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Двухслойные сети — универсальные аппроксиматоры функций

Функция $\sigma(z)$ — *сигмоида*, если $\lim\limits_{z\to-\infty}\sigma(z)=0$ и $\lim\limits_{z\to+\infty}\sigma(z)=1$.

Теорема Цыбенко (universal approximation theorem, 1989)

Если $\sigma(z)$ — непрерывная сигмоида, то для любой непрерывной на $[0,1]^n$ функции $f(x)$ существуют такие значения параметров H, $\alpha_h\in\mathbb{R}$, $w_h\in\mathbb{R}^n$, $w_0\in\mathbb{R}$, что двухслойная сеть

$$a(x)=\sum_{h=1}^{H}\alpha_h\sigma\big(\langle x,w_h\rangle-w_0\big)$$

равномерно приближает $f(x)$ с любой точностью ε:

$$\big|a(x)-f(x)\big|<\varepsilon,\ \text{для всех } x\in[0,1]^n.$$

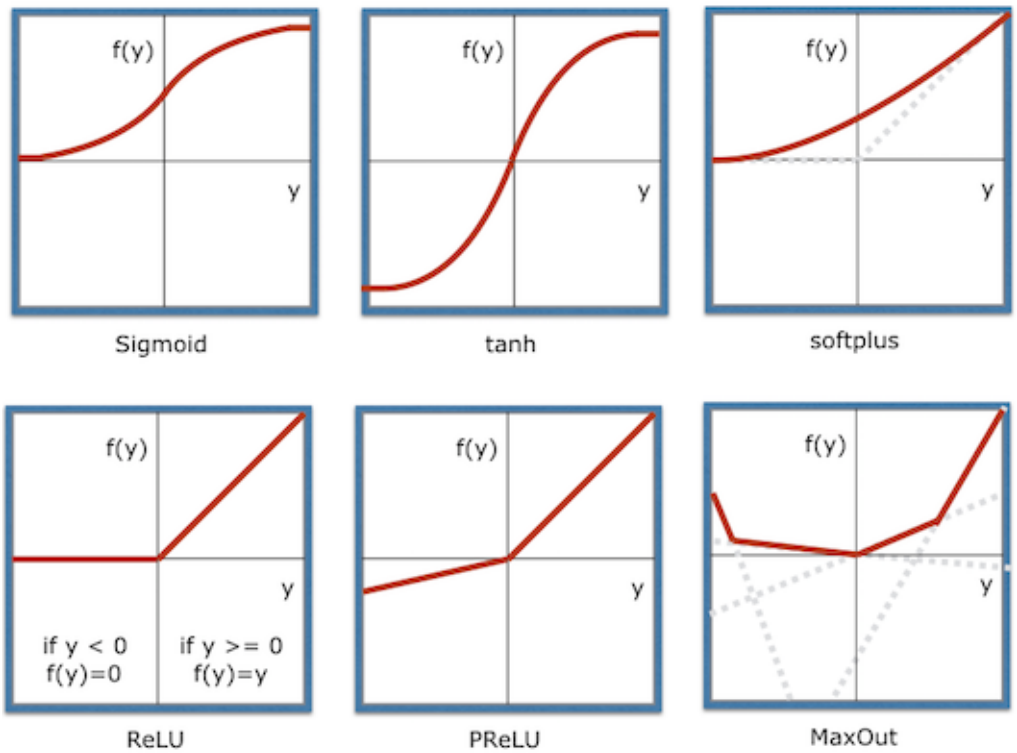
George Cybenko. Approximation by Superpositions of a Sigmoidal function. Mathematics of Control, Signals, and Systems. 1989.

Нелинейные функции активации

Функции $\sigma(y) = \frac{1}{1+e^{-y}}$ и $\text{th}(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$ могут приводить к затуханию градиентов или «параличу сети»

Функция положительной срезки (rectified linear unit, ReLU)

$$\text{ReLU}(y) = \max\{0, y\}; \qquad \text{PReLU}(y) = \max\{0, y\} + \alpha \min\{0, y\}$$



Многослойный персептрон (Multilayer Perceptron, MLP)

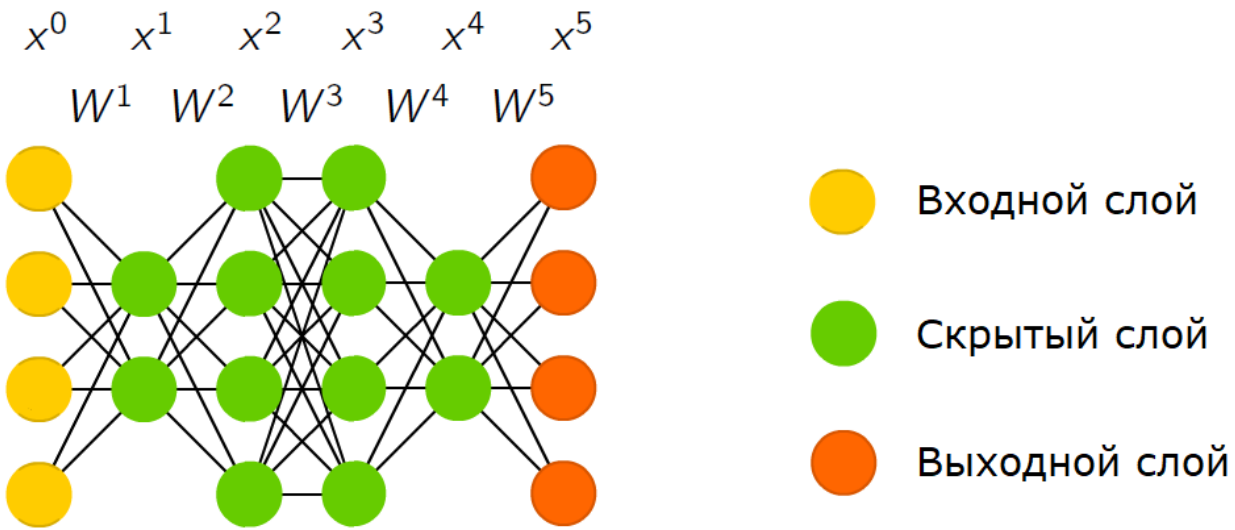
Полносвязная сеть, H_l — число нейронов в слое $l = 1, \ldots, L$

$x^0 = x = \big(f_j(x)\big)_{j=0}^{n}$ — вектор признаков на входе сети, $H_0 = n$

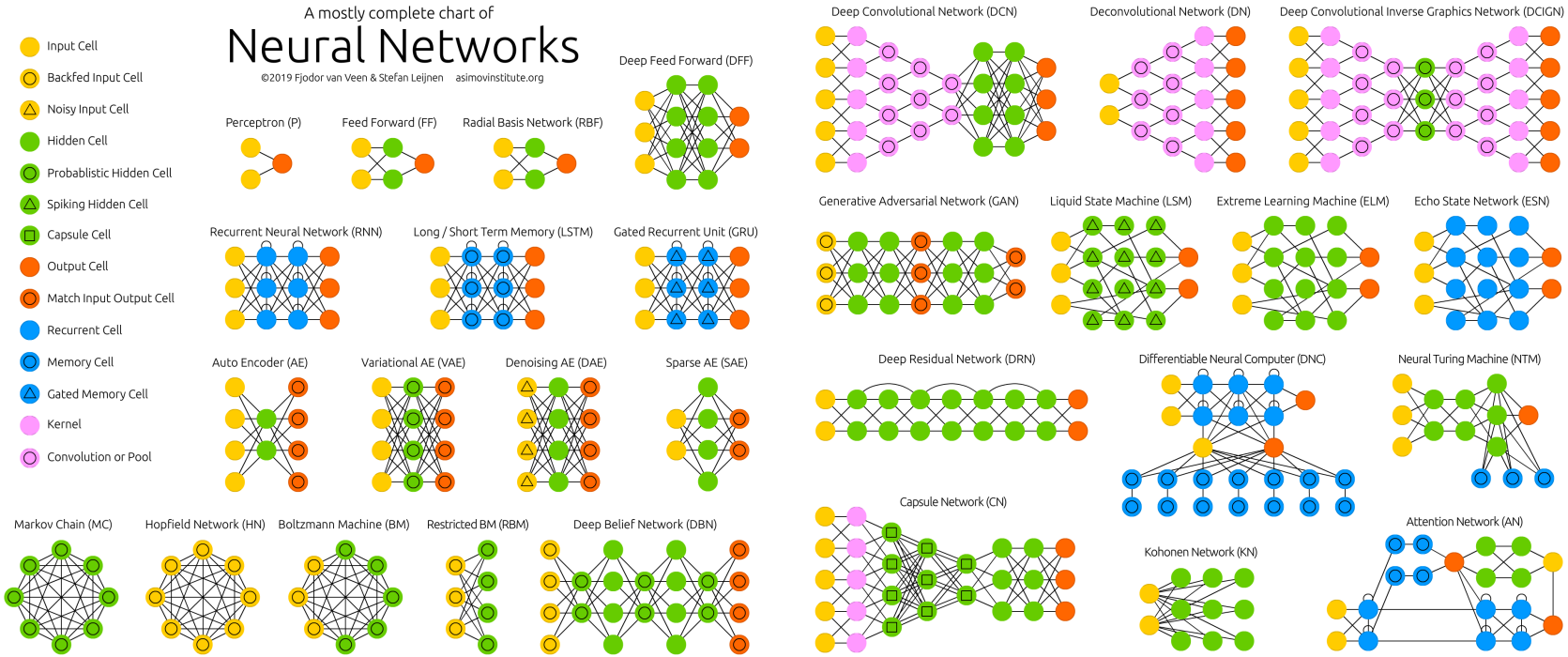
На выходе слоя — новое векторное представление объекта:

$$x^l = \sigma^l\big(W^l x^{l-1}\big) \quad \text{или} \quad x_h^l = \sigma_h^l\left(\sum_{k=0}^{H_{l-1}} w_{kh}^l x_k^{l-1}\right), \quad h = 1, \ldots, H_l,$$

где $W^l = (w_{kh}^l)$ — матрица весов размера $(H_{l-1}+1) \times H_l$

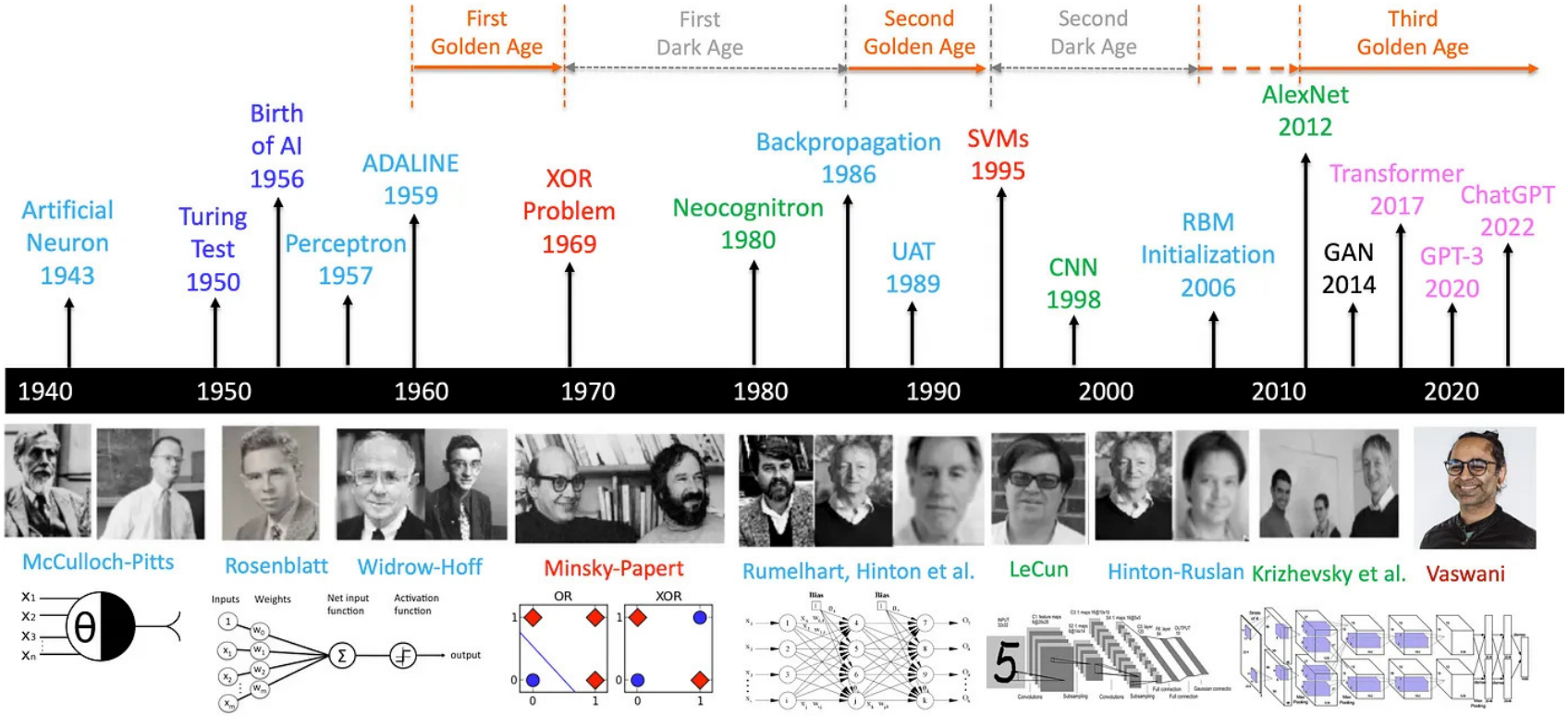


Зоопарк архитектур нейронных сетей



- *Архитектура сети* — структура слоёв и связей между ними, позволяющая наделять сеть нужными свойствами
- Все (почти) архитектуры обучаются методом BackProp — это SG + быстрое вычисление градиента + эвристики

Основные вехи развития нейронных сетей (AI winters)



Минский М., Пайперт С. Персептроны. 1971 (1969)
Галушкин А. И. Синтез многослойных систем распознавания образов. 1974
Ивáхненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965
Rummelhart D. et al. Learning internal representations by error propagation. 1986
Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012
Vaswani A. et al. Attention is all you need. 2017

Резюме в конце лекции

- *Искусственная нейронная сеть* (ANN) — скорее обучаемый векторизатор данных, чем модель работы мозга
- *Метод стохастического градиента* (SG, SGD)
  — общий подход к задачам машинного обучения
  — подходит для обучения по большим данным
  — лежит в основе BackPropagation для обучения ANN
- *Аппроксимация пороговой функции потерь* $L(M)$
  — общий подход к обучению моделей классификации
  — штраф за приближение к границе увеличивает зазор между классами, повышая надёжность классификации
- *Регуляризация* снижает переобучение, возникающее в линейных моделях из-за мультиколлинеарности
- *Эвристики* — полезные приёмы, обоснованные не теорией, а успешными воспроизводимыми экспериментами