Введение в машинное обучение

Михаил Иванов

Summer Freestyle Workshop

2026-08-25

0.1. Обо мне

Михаил Иванов, ассистент, МГУ им. М.В. Ломоносова.

Преподавание: машинное обучение, частотническая статистика, теория вероятностей, байесовская статистика, моделирование временных рядов, большие языковые модели.

Исследования:

- прогнозирование временных рядов с использованием смесей нормальных распределений [1] и процессов Ито [2], [3];
- смеси нормальных распределений вообще [1], [4].

0.2. План лекции

1. Основные черты машинного обучения.
2. Понятие *признака*, вектора признаков.
3. *Линейные* модели для регрессии и классификации.
4. *Нелинейные* модели для регрессии и классификации.
5. Основы статистического обучения.

1. Идеи машинного обучения

1.1. Имитация человека

Человек может выполнять сложные действия, требующие распознавания образов.

- Я вижу кошку или собаку?
- Перейти дорогу или подождать, пока машина проедет?
- О чём говорит собеседник?

Может ли это повторить компьютерный *алгоритм*?

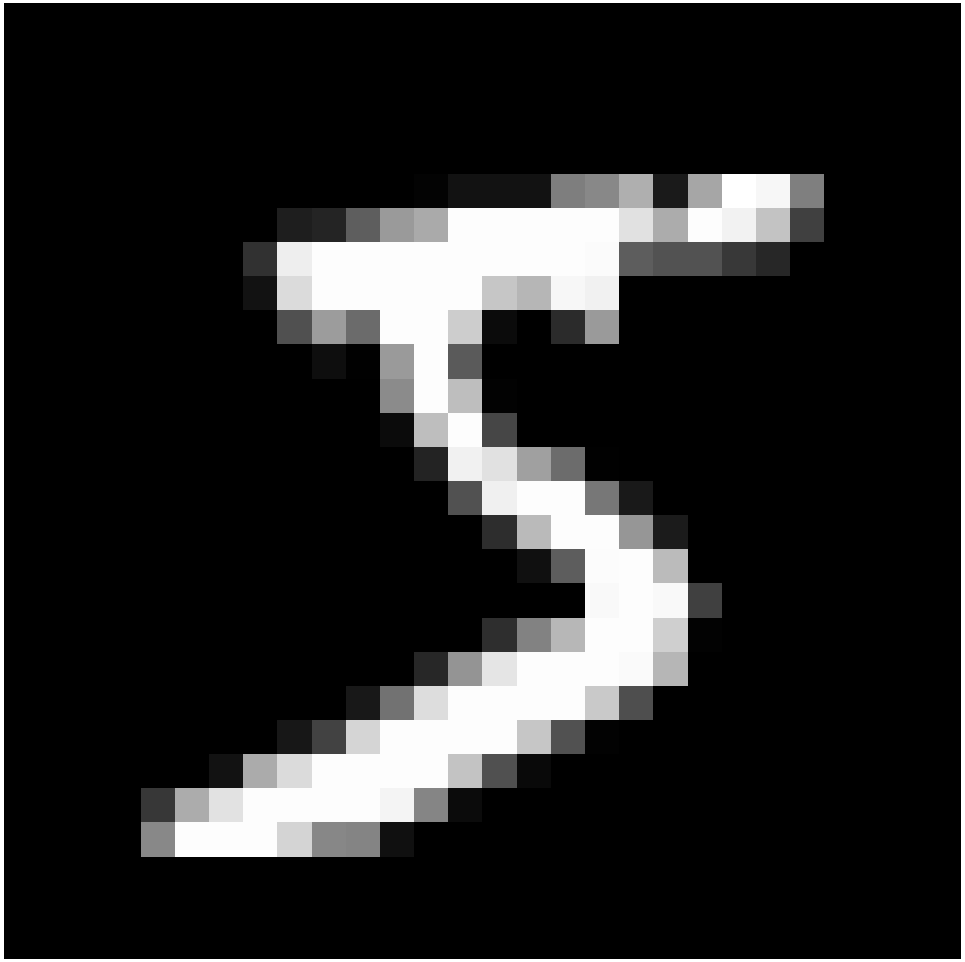

Рис. 1. Что здесь нарисовано?

1.2. Симуляция действий человека

Иногда есть чёткий алгоритм действий:

- поиск элемента в данном наборе;
- сортировка элементов;
- умножение в столбик;
- поиск производной $f'(x)$;
- доказательство некоторых теорем [5].

Иногда алгоритм неизвестен или слишком сложен.

- Что изображено на фото?[1]
- Что ответить в разговоре?
- Как научиться говорить?[2]
- Как не упасть, балансировать?

Как «научить» компьютер выполнять *посильные* действия, для которых нет алгоритма?

---

[1] Сравнивать данное фото со всем, что видел в жизни?

[2] Младенец зубрит фонетику, грамматику и слова?

1.3. Математическая формулировка

$$y = a(\text{  })$$

Выход/результат y связан с входом  неизвестной функцией a.

- Человек/природа – носитель этой неизвестной функции.
- Задача машинного обучения – *приблизить* эту функцию математической функцией/алгоритмом.

$$y \approx f(\text{  }, \theta)$$

Здесь $f(\cdot, \theta)$ – известная функция, θ – её параметры.

Будем называть $f(\cdot, \theta)$ *моделью* истинной функции.

1.4. Признаки объектов

Требуется измерить входной объект  .

- Человек: вход – результаты измерения органами чувств: зрение, слух, обоняние, осязание, ... Также память/опыт.
- Математическая функция $f(\cdot, \theta)$: вход – *числа*, результаты измерений объекта сенсорами.

Пример: рисунок $\longrightarrow$ матрица фотоаппарата $\longrightarrow$ яркость разных областей матрицы.

$$\text{  } = \mathbf{x}_n = (0, 0, 0, 0, ..., 1, 1, 0, 0, ..., 1, 1, 1, 1, ..., 0, 0, ...) \in \mathbb{R}^{784}$$

1.5. Статистическое обучение

Задача – «выучить» *закономерности* в множестве наблюдений.

$$a(\text{  }) = 5, a(\text{  }) = 5, a(\text{  }) = 5, a(\text{  }) = 8, a(\text{}) = 8, \ldots$$

Пример: как отец влияет на рост сына?

1. Выборка: много пар (отец, сын).
2. Извлечь *признак* отца (рост) и *метку* – рост сына [6].
3. Набор данных:

$$(1.90, 1.86), (\text{рост отца}, \text{рост сына}), (x_n, y_n), \ldots$$

4. По этим наблюдениям «выучить» математическую функцию/модель $f(\cdot, \theta)$, соотносящую x_n и y_n.

1.6. Фокус этой лекции

1. Обучение с учителем: $y_n = f(x_n; \theta)$. Нет обучения без учителя (кластеризация, снижение размерности) и обучения с подкреплением (отдельная лекция).
2. Параметрические модели.
3. Модели линейные по признакам. Нет заведомо нелинейных моделей (деревья, K ближайших соседей, ...).
4. Задачи:
   - Регрессия: $f : X \times \Theta \to \mathbb{R}$.
   - Бинарная классификация: $f : X \times \Theta \to \{-1, 1\}$.

2. Модели, линейные по признакам

2.1. Линейная регрессия

Метки $y_n \in \mathbb{R}$ линейно связаны с признаками $\mathbf{x}_n \in \mathbb{R}^F$:

$$y_n = b + \mathbf{w}^\top \mathbf{x}_n + \sigma\varepsilon_n$$

- ε_n – случайный шум,
- $(b, \mathbf{w})$ – параметры модели.

Используется с 1880-х: Гаусс, Гальтон [6], Пирсон [7].

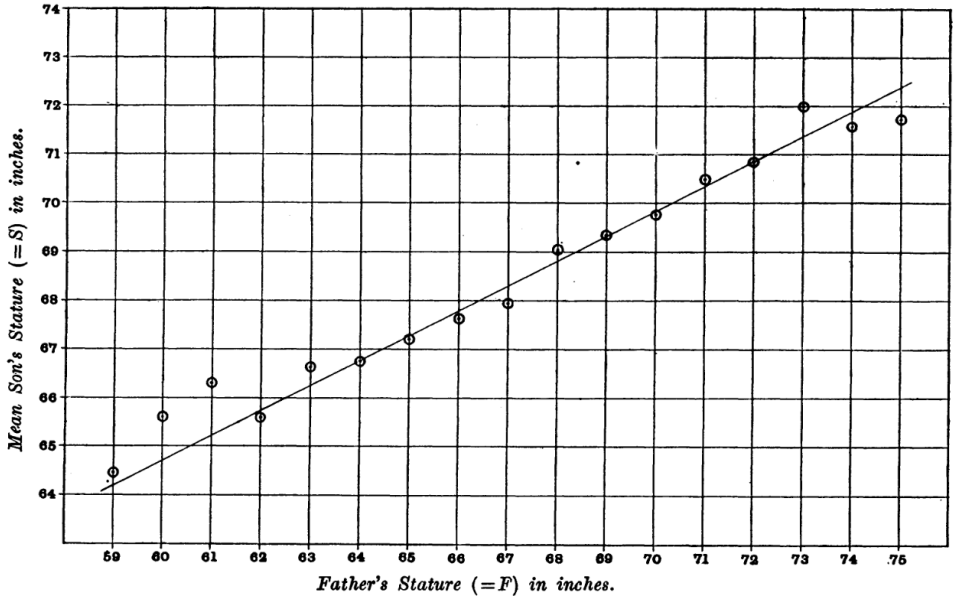

Рис. 2. Probable Stature of Son for given Father's Stature (inches).

$$S = 33.73 + 0.516F$$

1078 Cases. [7, Diagram I]

2.2. *Регрессия* к среднему: $S = 33.73 + 0.516F$

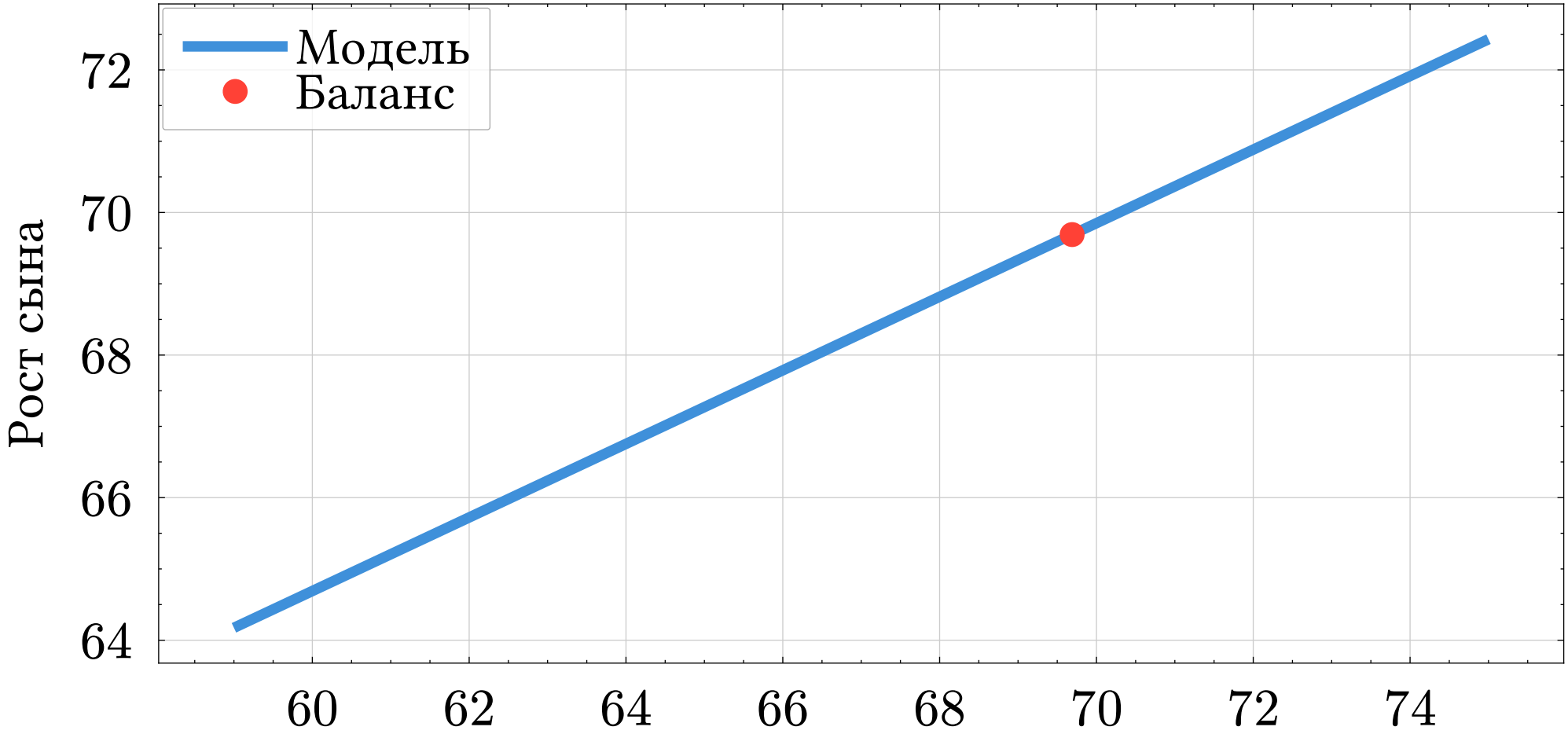

Рис. 3. Зависимость роста сына от роста отца (дюймы) [7].

2.3. Линейная регрессия: обучение

$$y_n = b + \mathbf{w}^\top \mathbf{x}_n + \sigma\varepsilon_n, \quad \varepsilon_n \sim \mathcal{N}(0, 1)$$

Минимизируем среднее *отрицательное* лог-правдоподобие:

$$\left(\hat{b}, \hat{\mathbf{w}}, \hat{\sigma}\right) = \arg\min_{b,\mathbf{w},\sigma} -\frac{1}{N}\sum_{n=1}^{N} \ln \mathcal{N}(y_n; b + \mathbf{w}^\top \varphi(\mathbf{x}_n), \sigma)$$

...или среднеквадратическую ошибку ($\arg\min_\mu \mathbb{E}(X-\mu)^2 = \mathbb{E}X$):

$$\left(\hat{b}, \hat{\mathbf{w}}\right) = \arg\min_{b,\mathbf{w}} -\frac{1}{N}\sum_{n=1}^{N} \left(y_n - \left(b + \mathbf{w}^\top \varphi(\mathbf{x}_n)\right)\right)^2$$

2.4. Бинарная классификация

Одни студенты сдают экзамен (исход $+1$), другие проваливаются (исход -1).

1. Студент.
2. Его атрибуты/признаки.
3. Точка в многомерном пространстве.

Цель: разделить два типа точек («положительные» и «отрицательные») некоей кривой.

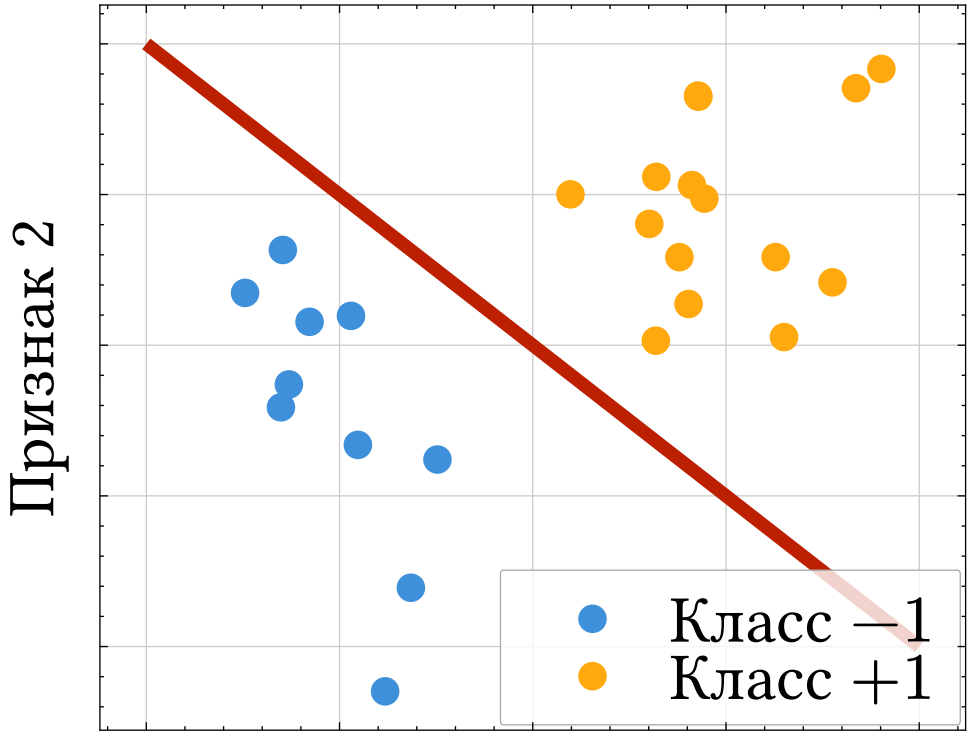

Рис. 4. Кассификация разделяющей прямой.

2.5. Бинарная классификация: моделирование

1. Студент сдаёт экзамен, набрав достаточное число баллов.
2. Устроим модель так.
   1. У каждого объекта есть «степень возбуждённости», «близость к $+1$», назовём это число $a_n \in \mathbb{R}$.
   2. «Слишком низкая» возбуждённость $\longrightarrow$ класс -1.
   3. «Достаточно высокая» возбуждённость $\longrightarrow$ класс $+1$.

«Достаточно высокая» значит «больше *отсечки* $-b$».

$$y_n = \begin{cases} -1 \text{ if } a_n \leq -b \\ +1 \text{ if } a_n > -b \end{cases} = \begin{cases} -1 \text{ if } b + a_n \leq 0 \\ +1 \text{ if } b + a_n > 0 \end{cases} = \operatorname{sign}(b + a_n)$$

2.6. Бинарная классификация: пробит и логит

Возбуждённость a_n зависит от $\mathbf{x}_n$ *линейно* и зашумлена ε_n:

$$a_n = \mathbf{w}^\top \mathbf{x}_n + \sigma\varepsilon_n$$
$$y_n = \operatorname{sign}(b + a_n)$$

Модель	Время появления	Распределение шума ε_n
Пробит	1930-е	Нормальное $\mathcal{N}(0, 1)$
Логит	1950-е	Логистическое $\mathrm{Logistic}(0, 1)$

Логит – аппроксимация пробита, см. исторический обзор в [8].

Обучение: оценка параметров $(b, \mathbf{w})$ методом максимального правдоподобия (при участии Р. Фишера).

2.7. Логит: результаты оценивания

Получаем разделяющую гиперплоскость (прямую):

$$\mathbf{x} \in \mathbb{R}^F : \hat{b} + \hat{\mathbf{w}}^\top \mathbf{x} = 0$$

Модель вероятностная, поэтому можно найти условные вероятности:

$$P(y_n = 1 \mid \mathbf{x}_n) = F\left(\hat{b} + \hat{\mathbf{w}}^\top \mathbf{x}_n\right),$$

где $F(x)$ – функция распределения шума.

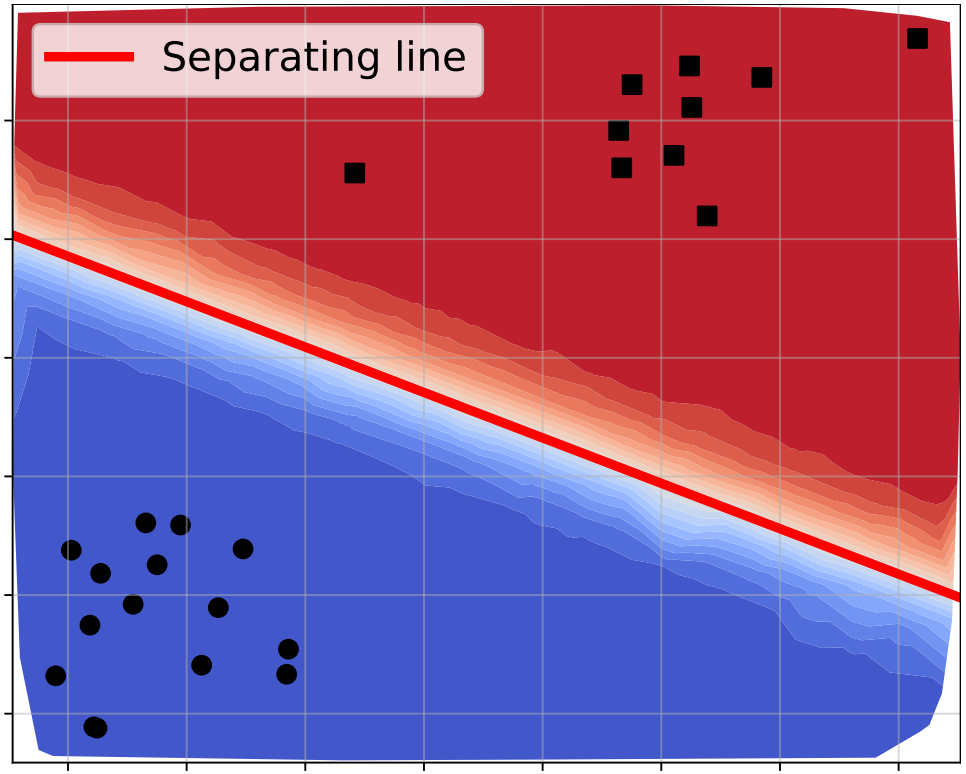

Рис. 5. Единственная разделяющая прямая.

2.8. Классификация: перцептрон [9], [10]

Франк Розенблатт, Cornell Aeronautical Laboratory, около 1960.

Модель сети нейронов в мозге [9].

1. «Stimuli impinge on a retina of sensory units (S-points), which ... respond on an all-or-nothing basis.»
2. «Impulses are transmitted to a set of association cells (A-units)...»
3. «If the algebraic sum of ... impulse intensities i_k is equal to or greater than the threshold (θ) of the A-unit, then the A-unit fires, again on an all-or-nothing basis.»

A-unit (модель нейрона) срабатывает (передаёт сигнал), когда $\sum_k i_k \geq \theta$, как в пробите и логите.

2.8. Классификация: перцептрон [9], [10]

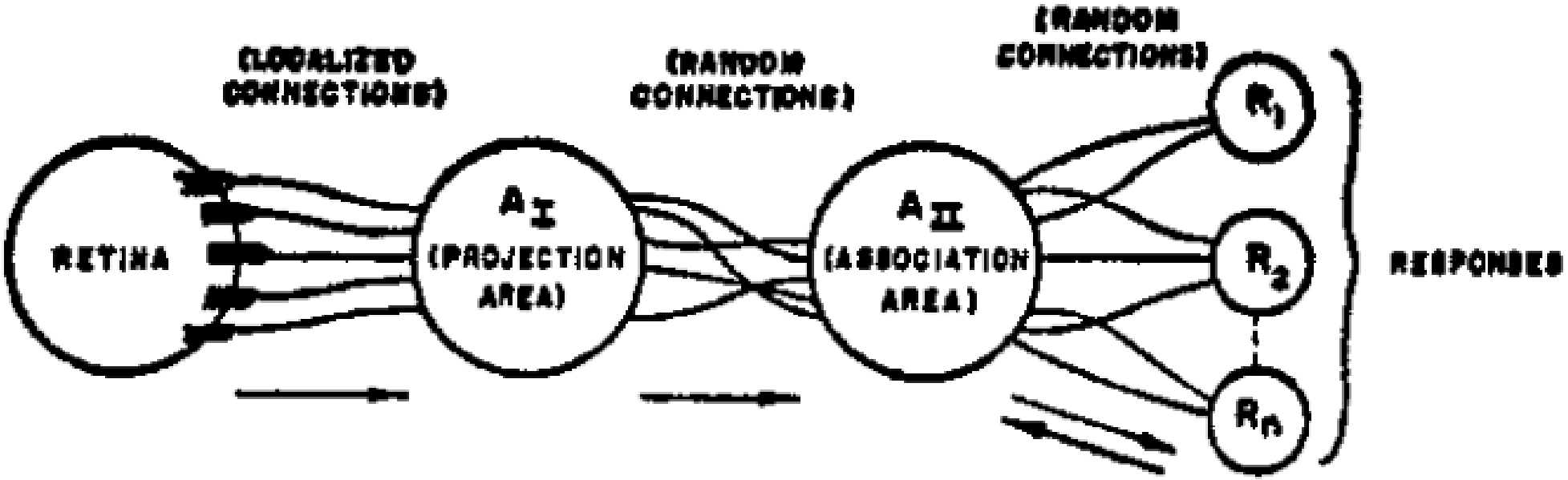

Рис. 6. Перцептрон Розенблатта [9, fig. 1].

Слои: сетчатка (сенсоры), prjection area, association area, выход.

Допустимы двунаправленные подключения между нейронами.

2.8. Классификация: перцептрон [9], [10]

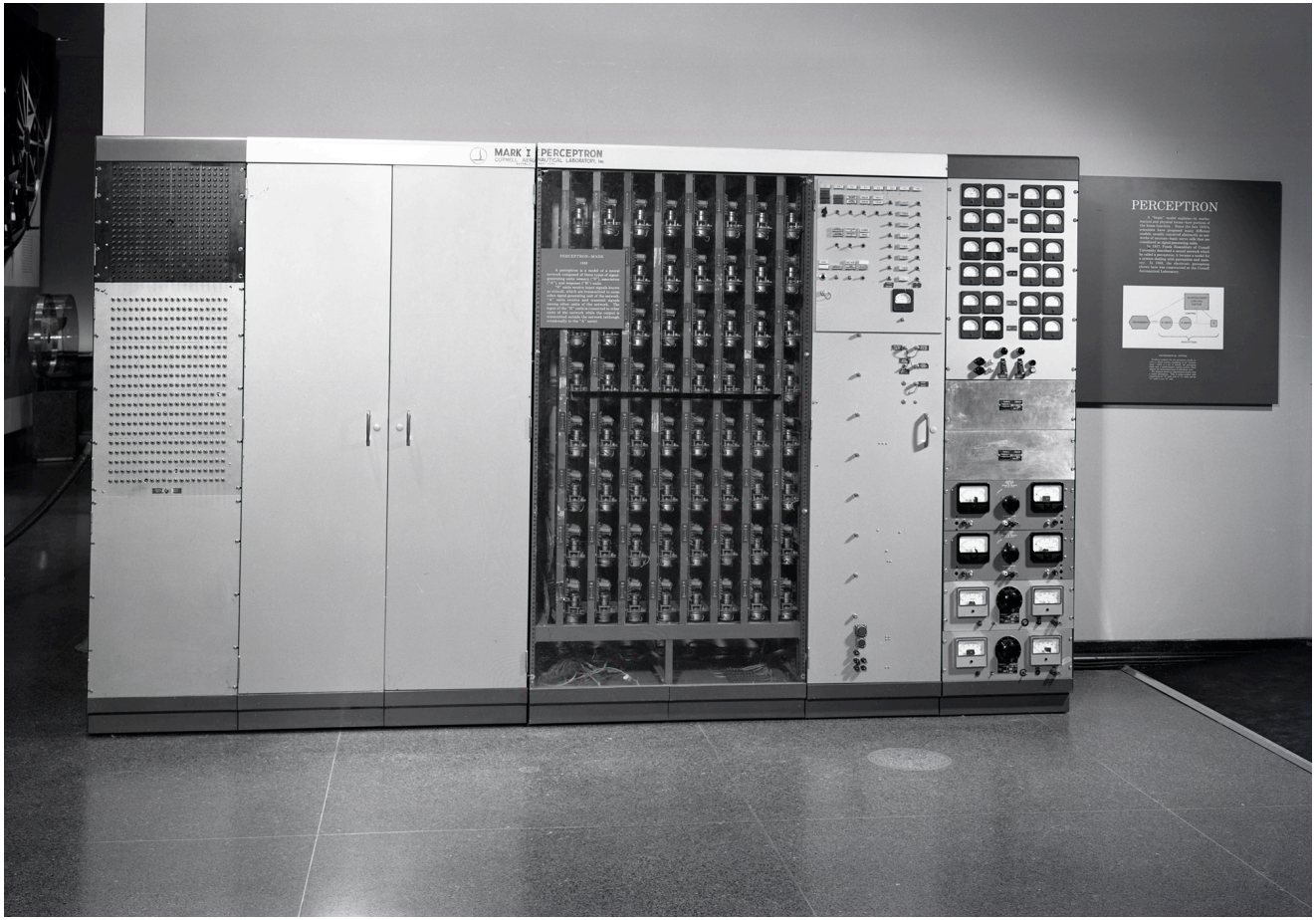

Рис. 7. Mark I Perceptron – реальный компьютер. Источник.

2.9. Простой перцептрон [11, sec. 9.a]

Выход единственного нейрона для бинарной классификации:

$$y_n = \begin{cases} -1 \text{ if } \mathbf{w}^\top \mathbf{x}_n < -b \\ +1 \text{ if } \mathbf{w}^\top \mathbf{x}_n > -b \end{cases} = \text{sign}\left(b + \mathbf{w}^\top \mathbf{x}_n\right)$$

- $\mathbf{x}_n = \left(x_{n,1}, ..., x_{n,F}\right)$ – интенсивности импульсов от входных нейронов (сенсоров). В оригинале: $x_{n,f} \in \{-1, 1\}$.
- $\mathbf{w} = \left(w_1, ..., w_F\right)$ – «ценности» входных нейронов: «если у A-unit высокая ценность, все его исходящие импульсы считаются более эффективными, сильными...» [9].

2.10. Простой перцептрон: обучение

1. Начальные значения: $b = 0$, $\mathbf{w} = (0, 0, \ldots, 0)$.
2. Получить признаки $\mathbf{x}_n$ и метку $y_n \in \{-1, +1\}$ нового наблюдения.
3. Сделать прогноз: $\hat{y}_n = \text{sign}\big(b + \mathbf{w}^\top \mathbf{x}_n\big)$.
   1. Если $y_n = \hat{y}_n$, перейти к пункту 2.
   2. Если $y_n = -1$, но $\hat{y}_n = +1$, то нейрон перевозбудился. *Ослабить* связи со входными нейронами:

$$b = b - 1, \quad \mathbf{w} = \mathbf{w} - \mathbf{x}_n$$

   3. Если $y_n = +1$, но $\hat{y}_n = -1$, то нейрон недовозбудился. *Усилить* связи со входными нейронами:

$$b = b + 1, \quad \mathbf{w} = \mathbf{w} + \mathbf{x}_n$$

4. Перейти пункту 2, пока все прогнозы не станут верными.

2.10. Простой перцептрон: обучение

Например, ослабление связи снижает «возбуждённость» (преактивацию) выходного нейрона для стимулов $\mathbf{x}_n$:

$$(\mathbf{w} - \mathbf{x})_n^\top \mathbf{x}_n < \mathbf{w}^\top \mathbf{x}_n$$

Если данные линейно разделимы, алгоритм сойдётся и найдёт *произвольную* разделяющую гиперплоскость за конечное число шагов [12].

1. Начальные значения: $b = 0$, $\mathbf{w} = (0, 0, ..., 0)$.
2. Получить признаки $\mathbf{x}_n$ и метку $y_n \in \{-1, +1\}$ нового наблюдения.
3. Сделать прогноз: $\hat{y}_n = \text{sign}(b + \mathbf{w}^\top \mathbf{x}_n)$.
   1. Если $y_n = \hat{y}_n$, перейти к пункту 2.
   2. Если $y_n = -1$, но $\hat{y}_n = +1$, то нейрон перевозбудился. *Ослабить* связи со входными нейронами:
   $$b = b - 1, \quad \mathbf{w} = \mathbf{w} - \mathbf{x}_n$$
   3. Если $y_n = +1$, но $\hat{y}_n = -1$, то нейрон недовозбудился. *Усилить* связи со входными нейронами:
   $$b = b + 1, \quad \mathbf{w} = \mathbf{w} + \mathbf{x}_n$$
4. Перейти пункту 2, пока все прогнозы не станут верными.

2.11. Различные разделяющие линии

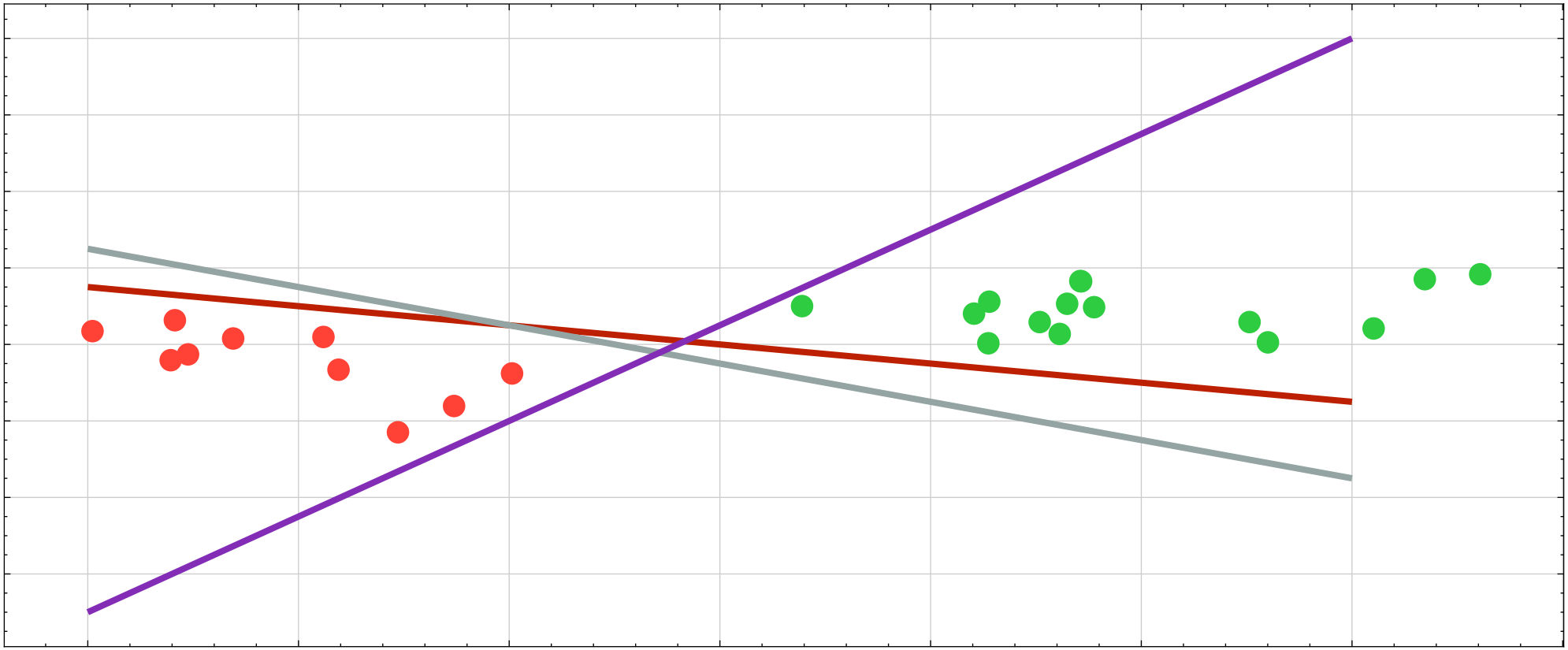

Рис. 8. Алгоритм обучения простого перцептрона может сойтись к любой из этих разделяющих линий.

2.12. Разделяющая гиперплоскость

Геометрическое место точек $H = \{\mathbf{x} : b + \mathbf{w}^\top \mathbf{x} = 0\}$ со свойствами:

1. если $b + \mathbf{w}^\top \mathbf{x}_n > 0$, n-й объект имеет класс $y_n = +1$;
2. если $b + \mathbf{w}^\top \mathbf{x}_n < 0$, n-й объект имеет класс $y_n = -1$.

Итоговое свойство разделяющей гиперплоскости:

$$y_n \times (b + \mathbf{w}^\top \mathbf{x}_n) > 0 \quad \forall n$$

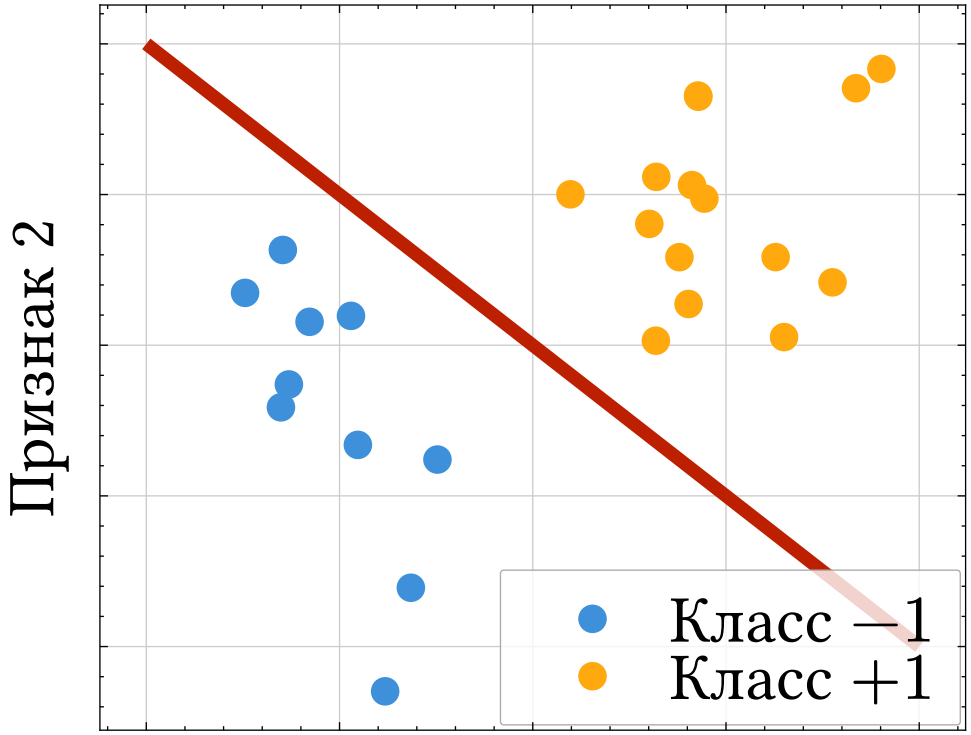

Рис. 9. Кассификация разделяющей прямой.

2.13. Метод опорных векторов [13], [14]

Советские математики
В.Н. Вапник и А.Я. Червоненкис,
1980-1990-е.

Метод выучивает:

1. разделяющую гиперплоскость...
2. ...которая *наиболее далека* от ближайших точек.

А простой перцептрон учит *произвольную* разделяющую гиперплоскость.

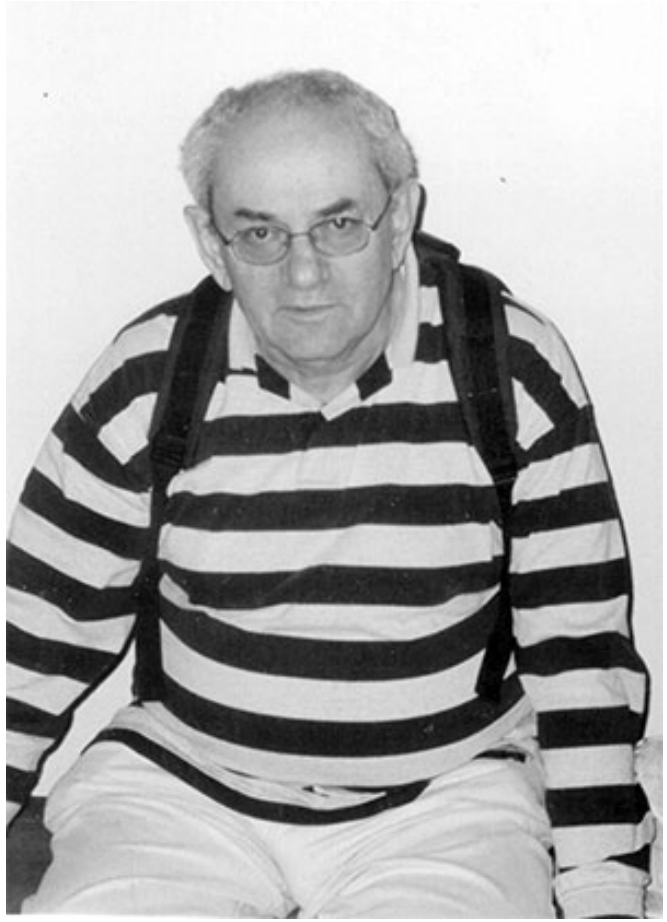

Рис. 10. В.Н. Вапник

2.13. Метод опорных векторов [13], [14]

Разделяющая линия, которая *наиболее далека* от ближайших точек. Линия с максимальным *зазором* (англ. margin).

$$\max_{b,\mathbf{w}} \frac{1}{\|\mathbf{w}\|}$$

$$\text{s.t. } y_n \cdot (b + \mathbf{w}^\top \mathbf{x}_n) > 1 \; \forall n$$

Единственное решение $(b^*, \mathbf{w}^*)$.

Точки на краю зазора – *опорные вектора.*

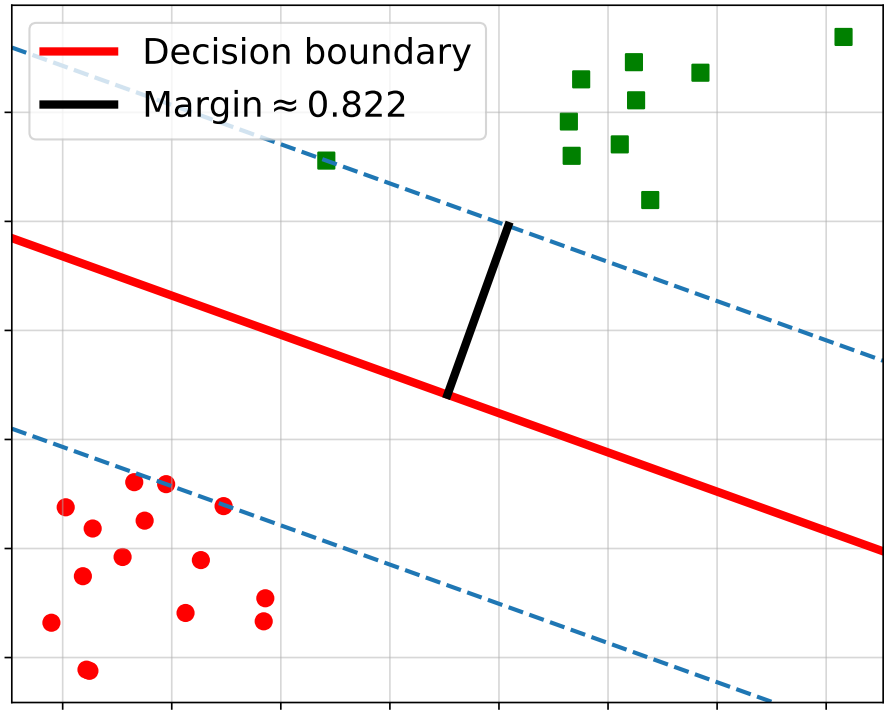

Рис. 11. Единственная разделяющая прямая с максимальным зазором.

2.14. Метод опорных векторов: обучение

Найти параметры гиперплоскости $(b^*, \mathbf{w}^*)$, решив оптимизационную задачу:

$$\max_{b,\mathbf{w}} \frac{1}{\|\mathbf{w}\|}$$

$$\text{s.t. } y_n \cdot (b + \mathbf{w}^\top \mathbf{x}_n) > 1 \; \forall n$$

Или, что (почти) то же самое (т.н. «мягкий» зазор):

$$\min_{b,\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \max(0, 1 - y_n \cdot (b + \mathbf{w}^\top \mathbf{x}_n))$$

Это *не* отрицательное лог-правдоподобие!

2.15. Если данные *почти* линейно разделимы

1. Пробит и логит: смещённая разделяющая прямая.
2. Простой перцептрон: бесконечный цикл.
3. Метод опорных векторов: обобщение с «мягким» зазором.

Допускается неверная классификация некоторых наблюдений.

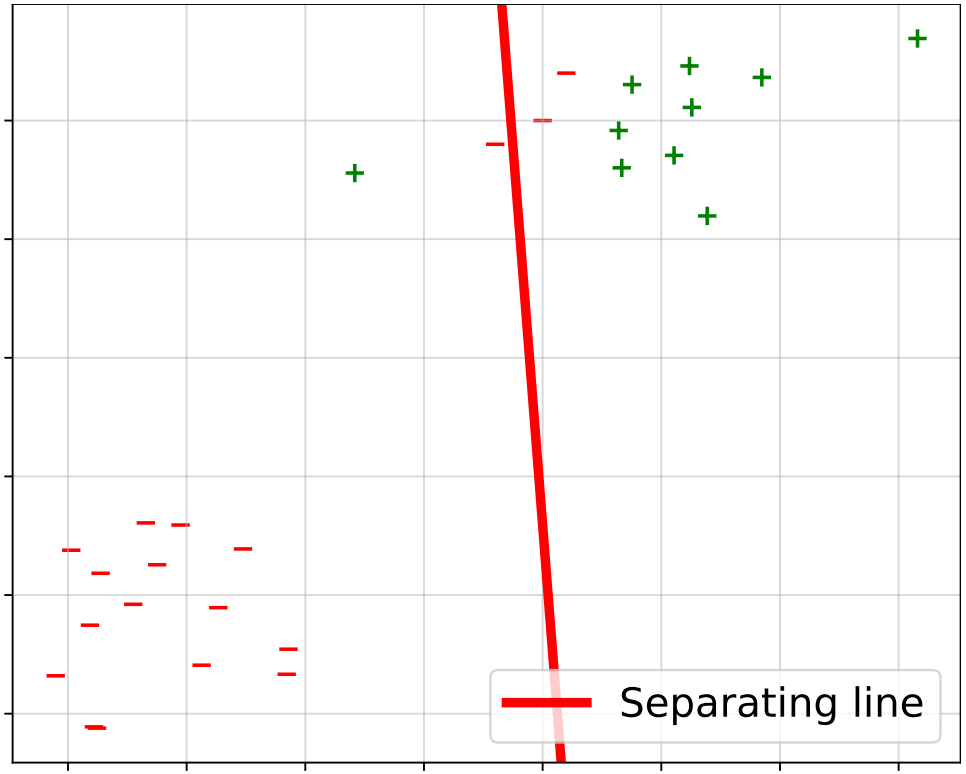

Рис. 12. Логит: линейно неразделимые данные.

3. Линейно неразделимые данные

3.1. Нелинейные взаимосвязи: регрессия

Нелинейная взаимосвязь:

$$y_n = \sin(x_n) + \sigma\varepsilon_n,$$

нужны нелинейные модели.

Шаг к нелинейности – **многочлены** относительно x_n:

$$y_n = b + w_1 x_n + w_2 x_n^3 + w_3 x_n^5 + \ldots$$

Линейно по *новым признакам*:

$$\mathbf{x}_n = (x_n, x_n^3, x_n^5, x_n^7) \in \mathbb{R}^4$$

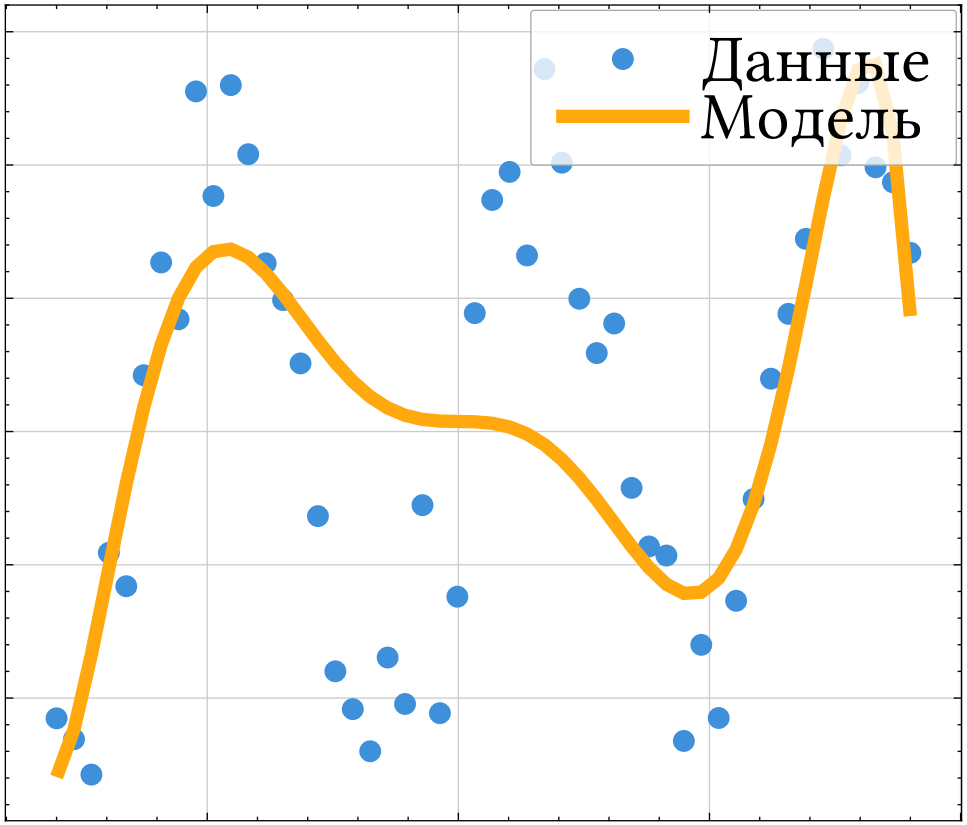

Рис. 13. Многочлен 7 степени нелинеен по *исходным* x_n.

3.2. «Извлекатели признаков»

Два типа признаков:

1. $\mathbf{x}_n \in \mathbb{R}^F$ – измерения реального мира сенсорами;
2. $\varphi(\mathbf{x}_n) \in \mathbb{R}^G$ – фиксированные нелинейные преобразования исходных измерений (обычно $G \gg F$).

$$\varphi(\mathbf{x}_n) = \big(x_{n,1}, x_{n,2}, \sin\big(x_{n,1}\big), x_{n,1}x_{n,2}, \exp\big(x_{n,2}\big), \ldots\big)$$

- «Извлекатель» $\varphi(\mathbf{x}_n)$ делает модель *нелинейной* по исходным признакам $\mathbf{x}_n$.
- Модель остаётся линейной (!) по новым признакам $\varphi(\mathbf{x}_n)$.

$\varphi : \mathbb{R}^F \to \mathbb{R}^G$ помогают улавливать нелинейные взаимосвязи между признаками $\mathbf{x}_n$ и метками y_n, используя линейные модели.

3.3. Нелинейные взаимосвязи: классификация

Разделяющая гиперплоскость может не существовать.

Решение: использовать нелинейные $\varphi : \mathbb{R}^F \to \mathbb{R}^G$.

$$y_n = \operatorname{sign}\big(b + \mathbf{w}^\top \varphi(\mathbf{x}_n)\big)$$

Обучающие алгоритмы те же, теперь применяются к новым *многомерным* векторам признаков $\varphi(\mathbf{x}_n)$ вместо $\mathbf{x}_n$.

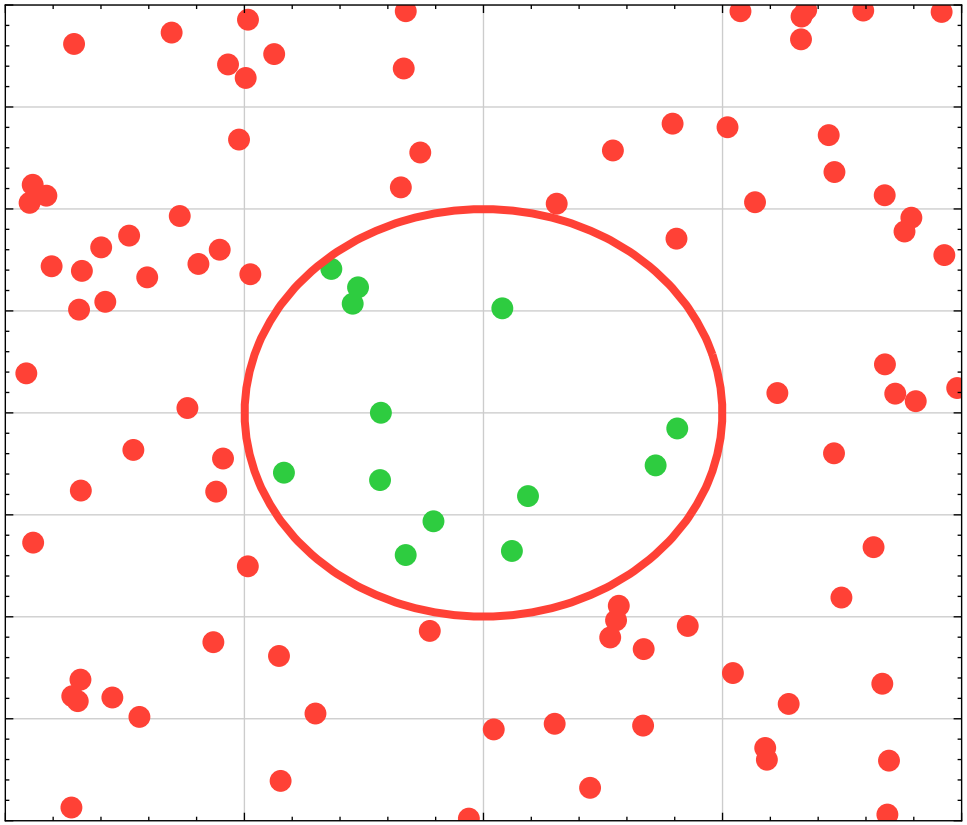

Рис. 14. Разделяющая кривая, нелинейная по $\big(x_{n,1}, x_{n,2}\big)$.

3.4. Нелинейности в методе опорных векторов

1. Преобразованные признаки $\varphi(\mathbf{x}_n) \in \mathbb{R}^G$ многомерные.
2. Их долго вычислять, они могут занимать много памяти.

Прогноз метода опорных векторов:

$$\hat{y}(\mathbf{x}) = \operatorname{sign}\Big(\hat{b} + \hat{\mathbf{w}}^\top \varphi(\mathbf{x})\Big) = \operatorname{sign}\left(\hat{b} + \sum_n \hat{\alpha}_n y_n \varphi(\mathbf{x}_n)^\top \varphi(\mathbf{x})\right)$$

- Подавляющее большинство $\hat{\alpha}_n = 0$.
- Для *опорных векторов* признаков $\hat{\alpha}_n \neq 0$.
- Иногда $k(\mathbf{x}_n, \mathbf{x}) = \varphi(\mathbf{x}_n)^\top \varphi(\mathbf{x})$ можно вычислить очень быстро, даже для бесконечномерных признаков $\varphi(\mathbf{x}) \in \mathbb{R}^\infty$.
- Функции $k(\mathbf{x}_n, \mathbf{x}) = \varphi(\mathbf{x}_n)^\top \varphi(\mathbf{x})$ называются *ядрами*.

3.5. Ядра: многомерные скалярные произведения

Полиномиальное ядро:

$$k(\mathbf{x}, \mathbf{y}; p) = \left(1 + \mathbf{x}^\top \mathbf{y}\right)^p$$

На самом деле вычисляет многомерное скалярное произведение справа.

Гауссово ядро:

$$\exp\left(-\frac{\gamma}{2}\|\mathbf{x} - \mathbf{y}\|^2\right) = \varphi(\mathbf{x}; \gamma)^\top \varphi(\mathbf{y}; \gamma),$$

где $\varphi(\mathbf{x}; \gamma) \in \mathbb{R}^\infty$ [15].

$$x_1^{10}y_1^{10} + 10x_1^9y_1^9 + 10x_1^9x_2y_2y_1^9 + 45x_1^8y_1^8 + 45x_1^8x_2^2y_2^2y_1^8 + 90x_1^8x_2y_2y_1^8 + 120x_1^7y_1^7 + 120x_1^7x_2^3y_2^3y_1^7 +$$
$$360x_1^7x_2^2y_2^2y_1^7 + 360x_1^7x_2y_2y_1^7 + 210x_1^6y_1^6 + 210x_1^6x_2^4y_2^4y_1^6 + 840x_1^6x_2^3y_2^3y_1^6 + 1260x_1^6x_2^2y_2^2y_1^6 +$$
$$840x_1^6x_2y_2y_1^6 + 252x_1^5y_1^5 + 252x_1^5x_2^5y_2^5y_1^5 + 1260x_1^5x_2^4y_2^4y_1^5 + 2520x_1^5x_2^3y_2^3y_1^5 + 2520x_1^5x_2^2y_2^2y_1^5 +$$
$$1260x_1^5x_2y_2y_1^5 + 210x_1^4x_2^6y_2^6y_1^4 + 1260x_1^4x_2^5y_2^5y_1^4 + 210x_1^4y_1^4 + 3150x_1^4x_2^4y_2^4y_1^4 + 4200x_1^4x_2^3y_2^3y_1^4 +$$
$$3150x_1^4x_2^2y_2^2y_1^4 + 1260x_1^4x_2y_2y_1^4 + 120x_1^3x_2^7y_2^7y_1^3 + 840x_1^3x_2^6y_2^6y_1^3 + 2520x_1^3x_2^5y_2^5y_1^3 + 4200x_1^3x_2^4y_2^4y_1^3 +$$
$$120x_1^3y_1^3 + 4200x_1^3x_2^3y_2^3y_1^3 + 2520x_1^3x_2^2y_2^2y_1^3 + 840x_1^3x_2y_2y_1^3 + 45x_1^2x_2^8y_2^8y_1^2 + 360x_1^2x_2^7y_2^7y_1^2 + 1260x_1^2x_2^6y_2^6y_1^2 +$$
$$2520x_1^2x_2^5y_2^5y_1^2 + 3150x_1^2x_2^4y_2^4y_1^2 + 2520x_1^2x_2^3y_2^3y_1^2 + 45x_1^2y_1^2 + 1260x_1^2x_2^2y_2^2y_1^2 + 360x_1^2x_2y_2y_1^2 +$$
$$10x_1x_2^9y_2^9y_1 + 90x_1x_2^8y_2^8y_1 + 360x_1x_2^7y_2^7y_1 + 840x_1x_2^6y_2^6y_1 + 1260x_1x_2^5y_2^5y_1 + 1260x_1x_2^4y_2^4y_1 + 840x_1x_2^3y_2^3y_1 +$$
$$360x_1x_2^2y_2^2y_1 + 10x_1y_1 + 90x_1x_2y_2y_1 + x_2^{10}y_2^{10} + 10x_2^9y_2^9 + 45x_2^8y_2^8 + 120x_2^7y_2^7 +$$
$$210x_2^6y_2^6 + 252x_2^5y_2^5 + 210x_2^4y_2^4 + 120x_2^3y_2^3 + 45x_2^2y_2^2 + 10x_2y_2 + 1$$
$$\|$$
$$\left(x_1^{10}, \sqrt{10}x_1^9, \sqrt{10}x_1^9x_2, \sqrt{45}x_1^8, \sqrt{90}x_1^8x_2, \sqrt{120}x_1^7, \sqrt{120}x_1^7x_2^3, ..., \sqrt{10}x_2, 1\right) \begin{pmatrix} y_1^{10} \\ \sqrt{10}y_1^9 \\ \sqrt{10}y_1^9y_2 \\ \sqrt{45}y_1^8 \\ \sqrt{90}y_1^8x_2 \\ \sqrt{120}y_1^7 \\ \sqrt{120}y_1^7y_2^3 \\ \vdots \\ \sqrt{10}y_2 \\ 1 \end{pmatrix}$$

Рис. 15. Полиномиальное ядро, $p = 10; \mathbf{x}, \mathbf{y} \in \mathbb{R}^2$.

3.6. Нелинейная разделяющая кривая: гауссово ядро

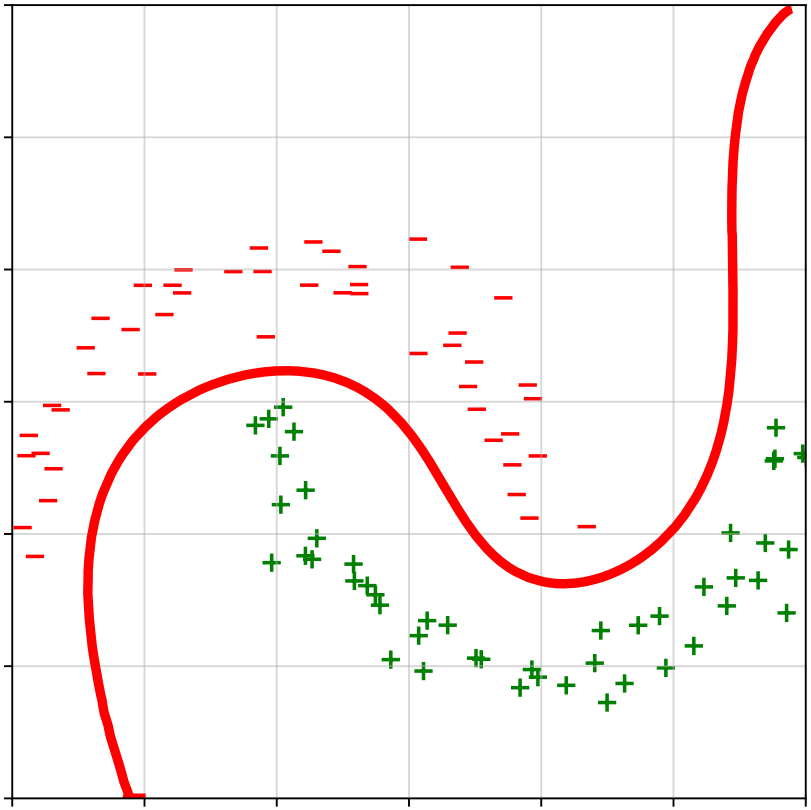

Рис. 16. Метод опорных векторов: разделяющая гиперплоскость в бесконечномерном признаковом пространстве.

4. Обучаемое извлечение признаков

4.1. Многослойный перцептрон

Простой перцептрон.

$$y_n = \operatorname{sign}\big(b_4 + \mathbf{w}^\top \varphi(\mathbf{x}_n)\big),$$

где $\varphi : \mathbb{R}^F \to \mathbb{R}^G$ – фиксированная функция, «извлекатель признаков».

Многослойный перцептрон.

$$y_n = \operatorname{sign}\big(b_4 + \mathbf{w}^\top \varphi_1(\mathbf{x}_n)\big)$$

$$\varphi_1(\mathbf{x}) = \operatorname{sign}\big(\mathbf{b}_1 + W_1 \varphi_2(\mathbf{x})\big) \in \mathbb{R}^9$$

$$\varphi_2(\mathbf{x}) = \operatorname{sign}\big(\mathbf{b}_2 + W_2 \mathbf{x}\big) \in \mathbb{R}^{50}$$

Функции φ имеют параметры, которые «обучаются» вместе с параметрами $(b, \mathbf{w})$.

Нейроны пересылают друг другу сигналы.

4.1. Многослойный перцептрон

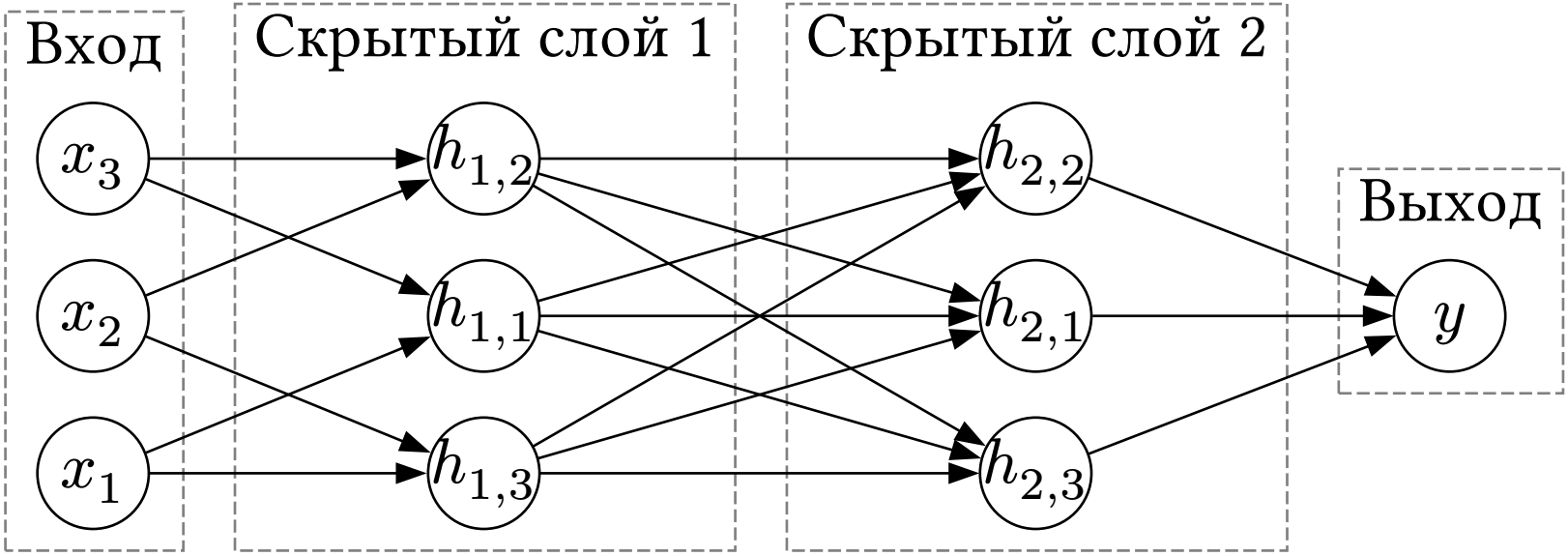

Рис. 17. Исходные признаки $\mathbf{x}_n$ – входные сигналы.
Новые признаки – выход промежуточных слоёв.
Выходной сигнал: $y \in \{-1, +1\}$.

4.2. Многослойный перцептрон: обучение

Обобщение логита:

$$\varphi_1(\mathbf{x}) = \operatorname{sign}(\mathbf{b}_1 + W_1\varphi_2(\mathbf{x})) \in \mathbb{R}^9$$

$$\varphi_2(\mathbf{x}) = \operatorname{sign}(\mathbf{b}_2 + W_2\mathbf{x}) \in \mathbb{R}^{50}$$

$$a_n = b + \mathbf{w}^\top\varphi_1(\mathbf{x}_n) + \sigma\varepsilon_n$$

$$y_n = \operatorname{sign}(a_n) \in \{-1, +1\}$$

Метод максимального правдоподобия?

$$\frac{d}{dx}\operatorname{sign}(x) = 0$$

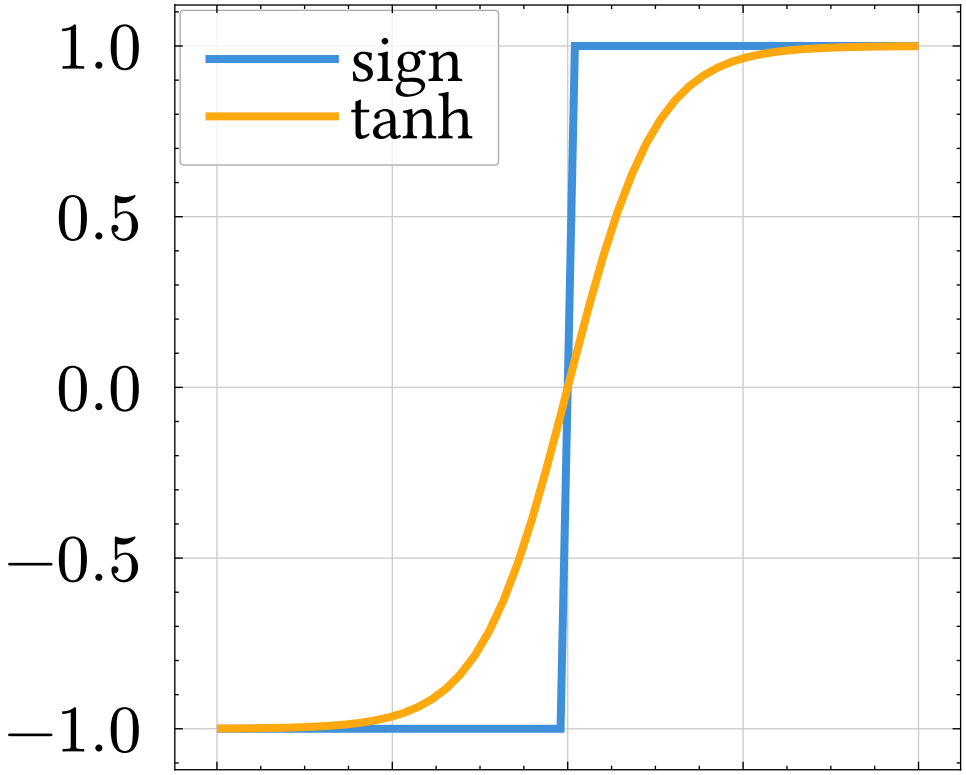

Рис. 18. $\tanh(x)$ – дифференцируемая аппроксимация $\operatorname{sign}(x)$.

4.3. Больше обучаемых признаков

1. Многослойный перцептрон (Розенблатт, 1960-е).
2. Свёрточные нейросети (Япония, 1980-е [16]; США, 2012 [17]).
   - $\mathbf{x}_n$ – изображение (матрица), $\varphi(\mathbf{x}_n; \theta)$ применяет операцию свёртки (по факту – опять скалярное произведение).
3. Рекуррентные нейросети (Розенблатт, потом возрождение в 1990-х: [18], [19]).
   - $\mathbf{x}_n$ – текущее наблюдение в *последовательности*, $\varphi(\mathbf{x}_n; \theta)$ обновляет «память» о прошлых значениях.

Параметры θ функции $\varphi(\mathbf{x}_n; \theta)$ *обучаются* вместе с $(b, \mathbf{w})$.

4.4. Признаки, выученные свёрточными сетями

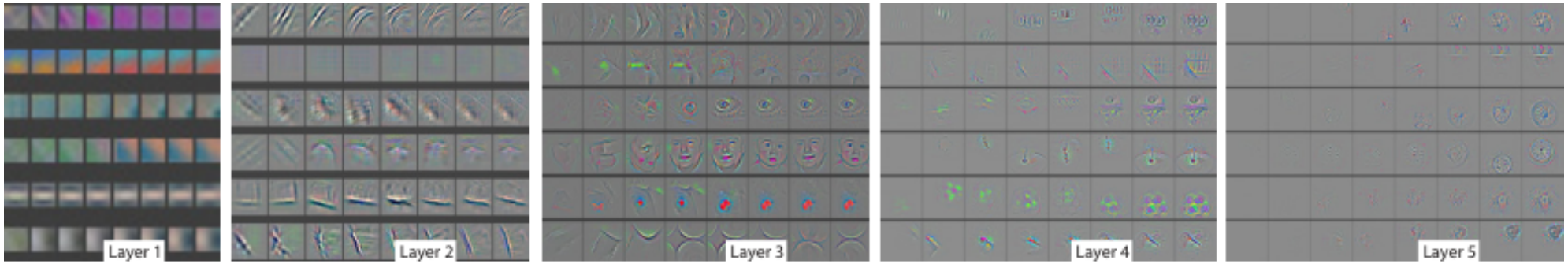

Рис. 19. Признаки (степени активации нейронов), выученные всё более глубокими слоями свёрточной нейросети [20, Fig. 4].

Они становятся всё более сложными: прямые линии, кривые, лица, …

Задача нейросети – многоклассовая классификация.

5. Обучающие алгоритмы

5.1. Метод максимального правдоподобия

Пусть шум в линейной регрессии ε_n имеет плотность $p(x;\theta)$.

Лог-правдоподобие $(y_1, ..., y_N)$ условно на признаки $(\mathbf{x}_1, ..., \mathbf{x}_N)$:

$$\ln L(b, \mathbf{w}, \theta) = \sum_{n=1}^{N} \ln p(y_n - (b + \mathbf{w}^\top \varphi(\mathbf{x}_n)); \theta)$$

Оптимизационные алгоритмы обычно *минимизируют*:

$$\left(\hat{b}, \hat{\mathbf{w}}, \hat{\theta}\right) = \arg\min_{b,\mathbf{w},\theta} \frac{1}{N} \sum_{n=1}^{N} -\ln p(y_n - (b + \mathbf{w}^\top \varphi(\mathbf{x}_n)); \theta)$$

Это выборочное среднее. Константа $1/N$ не влияет на оптимум.

5.2. Минимизация риска, обобщение

$$\left(\hat{b}, \hat{\mathbf{w}}, \hat{\theta}\right) = \arg\min_{b,\mathbf{w},\theta} \frac{1}{N} \sum_{n=1}^{N} -\ln p(y_n - (b + \mathbf{w}^\top \varphi(\mathbf{x}_n)); \theta)$$

Оценки параметров ищутся по выборке $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_n, y_n)\}_n$, но нам интересна вся популяция (генеральная совокупность).

Модель должна мало ошибаться для *любого* наблюдения из того же распределения, в т.ч. для новых наблюдений не в $\mathcal{D}_{\text{train}}$:

$$\left(\hat{b}, \hat{\mathbf{w}}, \hat{\theta}\right) \in \arg\min_{b,\mathbf{w},\theta} \mathbb{E}\left[-\ln p(Y - (b + \mathbf{w}^\top \varphi(X)); \theta) \mid X\right]$$

Это матожидание как функция параметров называется *риск*.
Цель: минимизировать риск, минимизируя *выборочный* риск.

5.3. Минимизация эмпирического риска

«Обучение» – минимизация эмпирического риска по параметрам.

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} l\left(y_n, \underbrace{f(\mathbf{x}_n; \theta)}_{\hat{y}_n} \right) + \lambda_N \mathcal{R}(\theta),$$

где $\mathcal{R}(\theta) \to \infty$, когда модель «слишком сложная».

Модель	**Функция потерь $l(y_n, \hat{y}_n)$**
Линейная регрессия	$(y_n - \hat{y}_n)^2$
Пробит, логит	$-\ln F(y_n \cdot \hat{y}_n)$
Простой перцептрон	$\max(0, -y_n \cdot \hat{y}_n)$
Метод опорных векторов	$\max(0, 1 - y_n \cdot \hat{y}_n)$
Любая вероятностная модель	$-\ln p(y_n \mid \hat{y}_n)$

5.4. Выбор модели

Есть обучающая выборка:

$$\mathcal{D}_{\text{train}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$$

Оценили $\hat{\theta}$, минимизируя эмпирический риск на $\mathcal{D}_{\text{train}}$.

«Хорошая» модель должна давать малый риск на любых данных $\mathcal{D}_{\text{val}}$ из *того же распределения*.

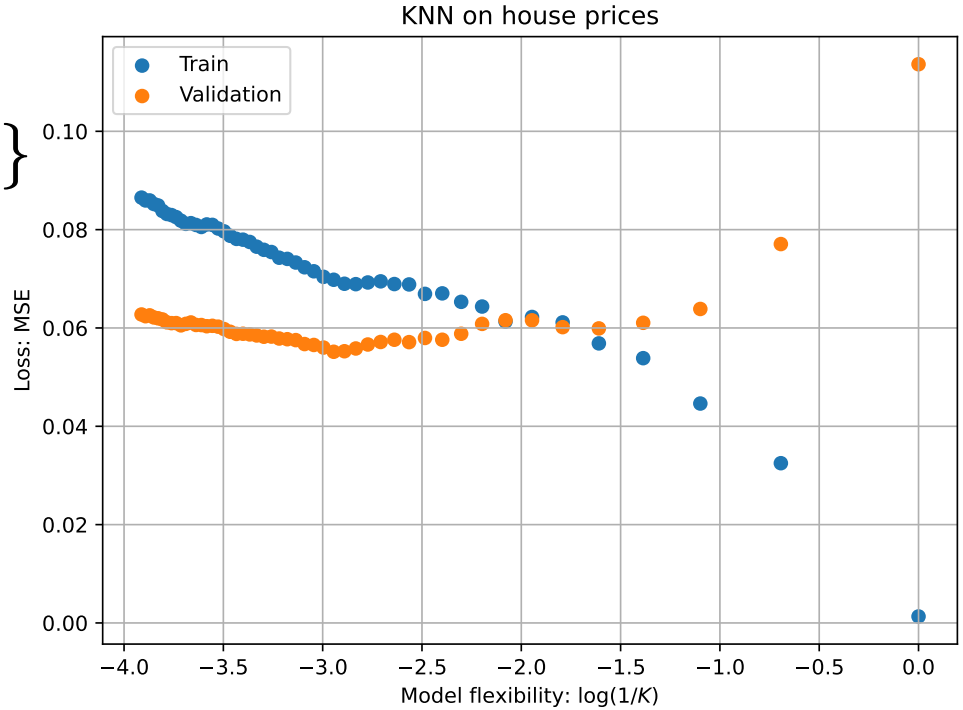

Рис. 20. Сложная модель даёт нулевую ошибку на $\mathcal{D}_{\text{train}}$, но большую ошибку на $\mathcal{D}_{\text{val}}$.

6. Выводы

6.1. Эконометрика – часть машинного обучения

Многие модели выглядят так:

$$y_n = g\big(b + \mathbf{w}^\top \varphi(\mathbf{x}_n; \theta) + \sigma\varepsilon_n\big)$$

- $y_n \in \mathbb{R}$ (регрессия) или y_n – метка класса (классификация).
- $\mathbf{x}_n \in \mathbb{R}^F$ содерит F признаков/измерений объекта.
- $(b, \mathbf{w}, \sigma)$ – параметры модели.
- $g(\cdot)$ переводит $\mathbb{R}$ в метки классов.
- ε_n – опциональный случайный шум (σ может быть нулём).
- $\varphi : \mathbb{R}^F \to \mathbb{R}^G$ извлекает из $\mathbf{x}_n$ новые, полезные признаки.

Нейросеть – линейная регрессия с обучаемыми «извлекаторами признаков».

6.2. Литература

[1] M. A. Ivanov, «Forecasting Itô-type Processes Using Features Based on Dynamic Gaussian Mixtures», *Moscow University Computational Mathematics and Cybernetics*, т. 50, вып. S1, сс. S31–S38, июл. 2026, doi: 10.3103/S0278641926700160.

[2] M. Ivanov, V. Korolev, и A. Vakshin, «Time-Series Forecasting by Statistical State-Dependent Reconstruction of Coefficients of Itô-Type Processes», *Mathematics*, т. 14, вып. 11, с. 1788, май 2026, doi: 10.3390/math14111788.

[3] V. Korolev *и др.*, «Extraction of Informative Statistical Features in the Problem of Forecasting Time Series Generated by Itô-Type

6.2. Литература

Processes». Просмотрено: 7 май 2026 г. [Онлайн]. Доступно на: https://arxiv.org/abs/2604.16865

[4] M. A. Ivanov и V. Y. Korolev, «Quasi-Exponentiated Mixed Normal Distributions», *Theory of Probability & Its Applications*, т. 71, вып. 2, сс. 248–254, авг. 2026, doi: 10.1137/S0040585X97T992951.

[5] «Lean (proof assistant)». Просмотрено: 23 август 2026 г. [Онлайн]. Доступно на: https://en.wikipedia.org/wiki/Lean_(proof_assistant)

6.2. Литература

[6] F. Galton, «Regression Towards Mediocrity in Hereditary Stature», *The Journal of the Anthropological Institute of Great Britain and Ireland*, т. 15, с. 246, 1886, doi: 10.2307/2841583.

[7] K. Pearson и A. Lee, «On the Laws of Inheritance in Man: I. Inheritance of Physical Characters», *Biometrika*, т. 2, вып. 4, с. 357, ноя. 1903, doi: 10.2307/2331507.

[8] J. Cramer, «The Origins of Logistic Regression», *SSRN Electronic Journal*, 2003, doi: 10.2139/ssrn.360300.

6.2. Литература

[9] F. Rosenblatt, «The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.», *Psychological Review*, т. 65, вып. 6, сс. 386–408, 1958, doi: 10.1037/h0042519.

[10] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, DC: Spartan books, 1962.

[11] H. D. Block, «The Perceptron: A Model for Brain Functioning. I», *Reviews of Modern Physics*, т. 34, вып. 1, сс. 123–135, янв. 1962, doi: 10.1103/RevModPhys.34.123.

6.2. Литература

[12] A. Novikoff, «On Convergence Proofs on Perceptrons», в *Proceedings of the Symposium on the Mathematical Theory of Automata*, Polytechnic institute of Brooklyn, апр. 1962, сс. 615–622. [Онлайн]. Доступно на: https://cs.uwaterloo.ca/~y328yu/classics/novikoff.pdf

[13] В. Вапник, *Восстановление зависимостей по эмпирическим данным*. М.: Наука, 1979.

[14] C. Cortes и V. Vapnik, «Support-Vector Networks», *Machine Learning*, т. 20, вып. 3, сс. 273–297, сен. 1995, doi: 10.1007/BF00994018.

6.2. Литература

[15] «Why Gaussian radial basis function maps the examples into an infinite-dimensional space?». Просмотрено: 25 август 2026 г. [Онлайн]. Доступно на: https://stackoverflow.com/a/23581657

[16] K. Fukushima, «Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position», *Biological Cybernetics*, т. 36, вып. 4, сс. 193–202, апр. 1980, doi: 10.1007/BF00344251.

[17] A. Krizhevsky, I. Sutskever, и G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks», в *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, и K. Weinberger, Ред., Curran Associates, Inc.,

6.2. Литература

2012. Просмотрено: 19 август 2026 г. [Онлайн]. Доступно на: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[18] J. L. Elman, «Finding Structure in Time», *Cognitive Science*, т. 14, вып. 2, сс. 179–211, мар. 1990, doi: 10.1207/s15516709cog1402_1.

[19] S. Hochreiter и J. Schmidhuber, «Long Short-Term Memory», *Neural Computation*, т. 9, вып. 8, сс. 1735–1780, ноя. 1997, doi: 10.1162/neco.1997.9.8.1735.

[20] M. D. Zeiler и R. Fergus, «Visualizing and Understanding Convolutional Networks», в *Computer Vision – ECCV 2014*, т. 8689,

6.2. Литература

D. Fleet, T. Pajdla, B. Schiele, и T. Tuytelaars, Ред., Cham: Springer International Publishing, 2014, сс. 818–833. doi: 10.1007/978-3-319-10590-1_53.