

# Обучение с учителем

Санкт-Петербургский государственный университет  
Кафедра статистического моделирования

16 сентября 2025, Санкт-Петербург

**Машинное обучение** — это раздел искусственного интеллекта, в котором разрабатываются методы и алгоритмы, позволяющие компьютерам обнаруживать закономерности в данных и делать прогнозы без явных инструкций.

**Обучение с учителем** — один из способов машинного обучения, в ходе которого для каждого примера в обучающем наборе известно, какой результат является правильным.

**Пример задач:**

- Регрессия: предсказание стоимости недвижимости, количества продаж некоторого товара, погоды.
- Классификация: предсказание ценовой категории товара, типа изображения, болеет ли человек или нет.

Дано:

- 1 Пространство объектов  $X$  — множество описаний объектов (например, фотографии, тексты, таблицы с признаками).
- 2 Пространство ответов  $Y$  — множество меток или значений, которые нужно предсказывать (например, классы «кот»/«собака», цена товара).
- 3 Обучающая выборка  $D = \{(x_i, y_i)\}_{i=1}^n$ , где  $x_i \in X$ ,  $y_i \in Y$ .

Модель:

$$y = f(x) + \varepsilon,$$

где  $f(x)$  — некоторая фиксированная (но неизвестная) функция,  $\varepsilon$  — шум,  $E\varepsilon = 0$  и  $\varepsilon$  не зависит от  $x$ .

**Предположение:**  $f(x)$  лежит в некотором классе функций (например, в классе линейных функций).

**Задача:** по обучающей выборке  $D$  построить оценку  $\hat{f}(x)$  функции  $f(x)$  в выбранном классе функций.

Чтобы оценить, насколько хорошо модель предсказывает ответы, используется **функция потерь**  $L(y, \hat{y})$ . Она показывает, насколько велико расхождение между истинными значениями  $y$  и его предсказаниями  $\hat{y}$ .

Тогда задача машинного обучения — минимизация выбранной функции потерь:

$$L(y, \hat{y}) \longrightarrow \min .$$

В большинстве случаев вычислить точку минимума функции потерь аналитически не представляется возможным, поэтому для его нахождения прибегают к методам **детерминированной и стохастической оптимизации** (например, перебор значений по сетке, метод Ньютона и квазиньютоновские методы, (стохастический) градиентный спуск, случайный поиск).

Градиентный спуск является наиболее распространенным алгоритмом оптимизации в машинном обучении.

Пусть  $f$  — некоторая гладкая функция, у которой необходимо найти минимум. Обозначим  $p_n = -\nabla f(x_n)$  — направление антиградиента в точке  $x_n$ . Тогда

$$x_{n+1} = x_n + \alpha p_n,$$

где  $\alpha$  — гиперпараметр, отвечающий за скорость обучения.

**Условия сходимости:** выпуклость  $f$ , липшицевость  $\nabla f$ , ...

**Критерий остановки:** достижение определенного числа итераций, малая норма градиента, малое изменение значения функции.

# Модификации градиентного спуска

В машинном обучении:

$$f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x),$$

где  $f_i$  — функция потерь для  $i$ -го наблюдения.

❶ (Batch) Gradient Descent (GD):

$$p_n = -\nabla f(x_n) = -\sum_{i=1}^n \nabla f_i(x_n).$$

- ❷ Mini-batch GD: случайным образом выбирается  $m$  наблюдений и делается шаг с  $p_n = -\sum_{i=1}^m \nabla f_{j_i}(x_n)$ .
- ❸ Stochastic GD: mini-batch GD с  $m = 1$ .
- ❹ Adam (Adaptive Moment Estimation): основан на GD, каждый параметр модели имеет собственную адаптивную скорость обучения, основанную на прошлых градиентах.

Процесс обучения любого алгоритма машинного обучения выглядит следующим образом:

- 1 Выборка  $D$  предварительно разбивается на тренировочную и тестовую:  $D = D_{\text{train}} \sqcup D_{\text{test}}$ .
- 2 На тренировочных данных модель обучается: минимизируется выбранная функция потерь  $L(y, \hat{y})$ .
- 3 Также часто присутствует и валидационная выборка  $D_{\text{val}}$ , на основе которой подбираются гиперпараметры модели/производится остановка оптимизации функции потерь.

После обучения проверяется качество/обобщающая способность модели — на тестовых данных вычисляются различные метрики. Выбираются они в зависимости от задачи.

**Таблица:** Метрики для задач регрессии и классификации

Регрессия	MSE, RMSE, MAE, MAPE, WAPE
Классификация	Accuracy, Precision, Recall, F1-score, ROC AUC, PR AUC

Также имеет смысл сравнить полученные результаты с baseline предсказаниями (например, среднее в задаче регрессии и наиболее распространенная метка в задаче классификации).



Пусть целевая переменная  $y$  принимает значения  $\{-1, 1\}$ . Хотим обучить линейную модель так, чтобы плоскость, которую она задает, как можно лучше отделяла объекты одного класса от другого.

Линейный классификатор:

$$\hat{y} = \hat{f}(x; w) = \text{sign}\langle x, w \rangle.$$

Функция потерь:

$$L(y, \hat{y}) = \sum_{i=1}^n \mathbb{I}[y_i \langle x_i, w \rangle < 0] \longrightarrow \min_w.$$

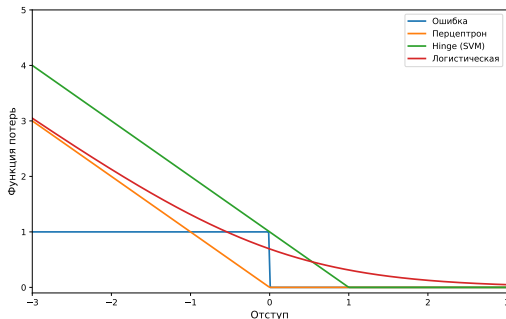
Величина  $M_i = y_i \langle x_i, w \rangle$  называется **отступом** (margin) классификатора. Абсолютная величина отступа говорит о степени уверенности классификатора.

**Проблема:** функция  $\mathbb{I}[M < 0]$  кусочно-постоянная, следовательно функцию потерь невозможно оптимизировать градиентными методами, поскольку во всех точках производная равна нулю.

**Решение:** можно мажорировать эту функцию более гладкой функцией и минимизировать функцию потерь с этой мажорирующей функцией с помощью методов численной оптимизации.

# Линейная классификация. Функции потерь

- 1 Перцептрон:  $L(M) = \max(0, -M)$  — отступы учитываются только для неправильно классифицированных объектов пропорционально величине отступа.
- 2 Hinge (SVM):  $L(M) = \max(0, 1 - M)$  — объекты, которые классифицированы правильно, но не очень «уверенно», продолжают вносить свой вклад в градиент.
- 3 Логистическая:  $L(M) = \ln(1 + e^{-M})$ .



Посмотрим на задачу классификации как на задачу предсказания вероятностей (например, предсказание «кликабельности» рекламного баннера).

**Принцип работы:** научить линейную модель предсказывать значения  $z \in \mathbb{R}$  (логиты), а затем преобразовывать их в вероятности с помощью сигмоиды:

$$z_i = \langle x_i, w \rangle = \ln \frac{p_i}{1 - p_i}, \quad p_i = \frac{1}{1 + e^{-\langle x_i, w \rangle}} = \sigma(\langle x_i, w \rangle).$$

Функция правдоподобия для распределения Бернулли:

$$p(y \mid \mathbf{X}, w) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1-y_i}.$$

Прологарифмируем:

$$\sum_{i=1}^n \left[ y_i \ln(\sigma(\langle x_i, w \rangle)) + (1 - y_i) \ln(1 - \sigma(\langle x_i, w \rangle)) \right].$$

Теперь пусть  $y \in \{-1, 1\}$ . Тогда, поскольку  $\sigma(z) = 1 - \sigma(-z)$ , логарифм правдоподобия можно представить в следующем виде:

$$\begin{aligned}\ln p(y \mid \mathbf{X}, w) &= - \sum_{i=1}^n \left[ \mathbb{I}[y_i = 1] \sigma(z_i) + \mathbb{I}[y_i = -1] (1 - \sigma(z_i)) \right] \\ &= - \sum_{i=1}^n \ln \sigma(y_i \langle x_i, w \rangle) \\ &= \sum_{i=1}^n \ln (1 + e^{-M})\end{aligned}$$

Таким образом, функцию потерь в логистической регрессии можно представить в виде функции от отступа.