Лекция 9

Введение в нейронные сети

Машинное обучение · 2026

Лектор

Елизавета Власова

Практики

Кирилл Захаров
Михаил Подсытник
Игорь Толстокулаков

Задача бинарной классификации

Предсказываем, уйдёт ли клиент в следующем месяце.

Что известно о клиенте

Признак	Значение
Недель без активности, x_1	3
Обращений за помощью, x_2	1

Модель получает признаки x.

$$P(y = 1 \mid x) = ?$$

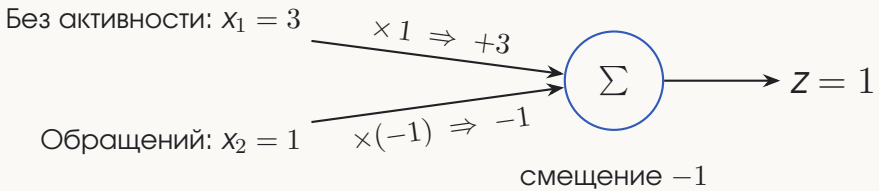
$y = 1$ — клиент уйдёт.

$y = 0$ — останется.

Ищем вероятность ухода.

От признаков к оценке

Для нашего примера возьмём веса $w_1 = 1$, $w_2 = -1$ и смещение $b = -1$.



$$z = w_1 x_1 + w_2 x_2 + b = 3 - 1 - 1 = 1.$$

z — линейная оценка (логит). Это ещё не вероятность.

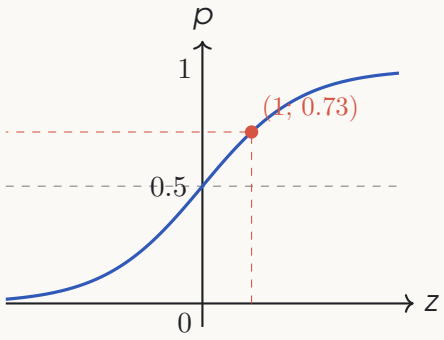
Сигмоида: от оценки к вероятности

$$p = \sigma(z) = \frac{1}{1 + e^{-z}}.$$

Для нашего клиента $z = 1$:

$$p = \sigma(1) \approx 0.73.$$

$z = 0 \Rightarrow p = 0.5$; чем больше z, тем больше p.



Численный пример

Те же веса; меняется только число недель без активности.

	x_1	Логит $z = x_1 - 1 - 1$	$p = \sigma(z)$
Клиент A	1	-1	0.27
Клиент B	2	0	0.50
Наш клиент	3	1	0.73

Чем дольше нет активности, тем выше оценка вероятности ухода при этих весах.

От вероятности к классу

При выбранном пороге t:

$$\hat{y} = \begin{cases} 1, & p \geq t, \\ 0, & p < t. \end{cases}$$

При $t = 0.5$: $p \geq 0.5$ тогда и только тогда, когда $w^\top x + b \geq 0$.

Граница решения $w^\top x + b = 0$ — прямая в двумерном пространстве признаков.

Порог можно выбрать иначе, если цена ошибок различается.

Где проходит граница?

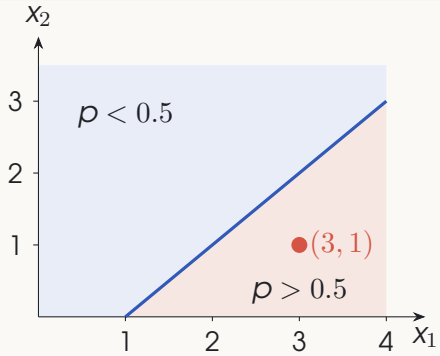
Та же модель: $z = x_1 - x_2 - 1$.

При пороге 0.5 граница:

$$x_2 = x_1 - 1.$$

Наш клиент $(3, 1)$:
$z = 1$, $p \approx 0.73$.

Прогноз: уйдёт.



Как выбираются веса

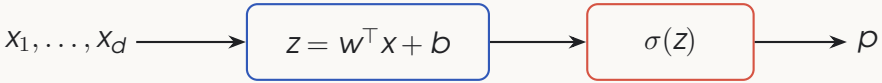
На обучающих данных знаем ответы y_i; модель выдаёт вероятности p_i.

$$L(w, b) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log p_i + (1 - y_i)\log(1 - p_i)\right].$$

Это log loss, или binary cross-entropy. Веса w и смещение b подбираются так, чтобы уменьшить среднюю ошибку, например градиентным спуском.

При $y = 1$: $p = 0.9$ даёт потерю 0.105, а $p = 0.1$ — 2.303.

Логистическая регрессия как нейрон



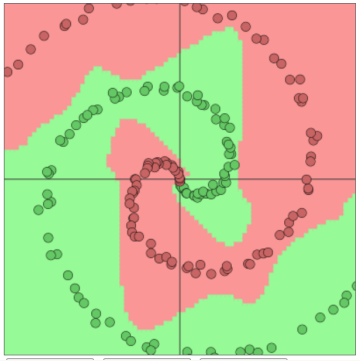
$$x \longrightarrow z = w^\top x + b \longrightarrow p = \sigma(z).$$

Это один нейрон с сигмоидной активацией. Далее соединим несколько таких вычислительных блоков в сеть.

Одного нейрона недостаточно

Для бинарной классификации граница $p = 0.5$ задаётся уравнением $w^\top x + b = 0$.

Это прямая или гиперплоскость: сложную границу ей не выразить.



Полносвязная сеть

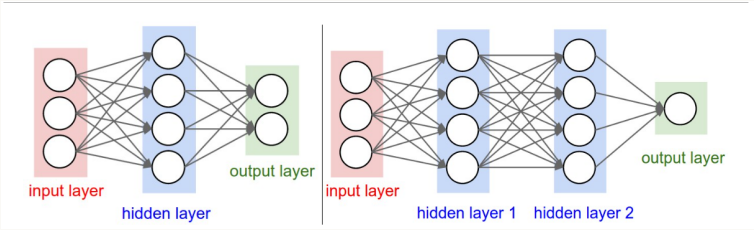
От одного нейрона к нескольким слоям

1

Нейронная сеть как функция

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^k, \qquad \theta = \{W^{(1)}, b^{(1)}, \ldots\}.$$

На входе — вектор признаков x. Выход зависит от задачи: число, вероятности классов или вектор признаков.

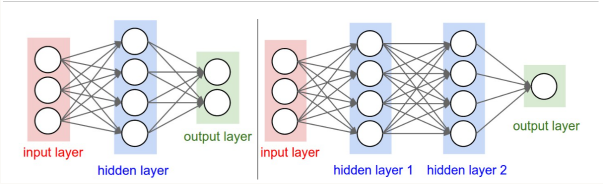


Полносвязный слой

$$h^{(l)} = \phi\left(W^{(l)} h^{(l-1)} + b^{(l)}\right), \qquad h^{(0)} = x.$$

$W^{(l)}$ — веса, $b^{(l)}$ — смещения, ϕ — функция активации.

Каждый нейрон слоя получает выходы всех нейронов предыдущего слоя.



Размерности и число параметров

Вход: 4 признака. Скрытый слой: 3 нейрона. Выход: 2 числа.

$$W^{(1)} \in \mathbb{R}^{3\times 4}, \quad b^{(1)} \in \mathbb{R}^{3}, \quad W^{(2)} \in \mathbb{R}^{2\times 3}, \quad b^{(2)} \in \mathbb{R}^{2}.$$

Сколько обучаемых параметров?

Ответ: $3 \cdot 4 + 3 + 2 \cdot 3 + 2 = 23.$

Пример: XOR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Класс 1 получается, когда входы различаются. Можно ли отделить единицы от нулей одной прямой?

Нет: для решения требуется нелинейная граница.

Зачем нужна нелинейность

Два линейных преобразования подряд:

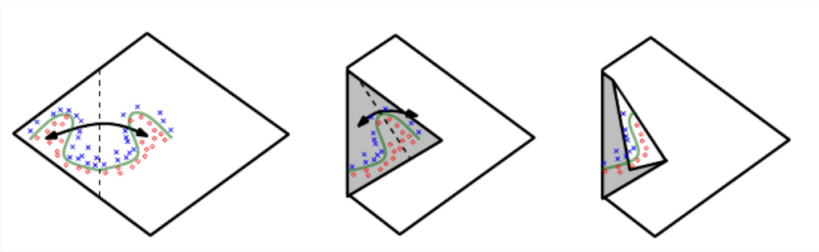
$$h = W_1 x + b_1, \qquad z = W_2 h + b_2.$$

Подставим первое во второе:

$$z = (W_2 W_1) x + (W_2 b_1 + b_2).$$

Получился один линейный слой. Нелинейность между слоями позволяет выразить более сложные зависимости.

Как сеть меняет представление



Каждый слой строит новые признаки из признаков предыдущего. В новом пространстве разделение может стать проще.

Универсальная аппроксимация

Сеть с одним достаточно широким скрытым слоем и подходящей нелинейностью может с произвольной точностью приблизить непрерывную функцию на компакте.

$$f(x) \approx \sum_{j=1}^{m} a_j \, \phi(w_j^\top x + b_j).$$

Теорема говорит о существовании параметров, но не гарантирует, что обучение их найдёт или что сеть хорошо обобщит результат.

Функции активации

Нелинейность внутри сети

2

Сигмоида

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

Значения от 0 до 1: удобно для вероятности бинарного класса.

При больших $|z|$ производная близка к нулю.

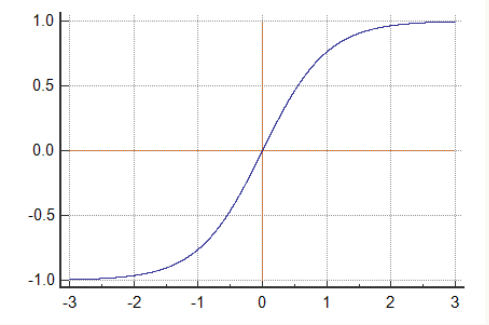


Tanh

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

Значения от -1 до 1; функция центрирована вокруг нуля.

На насыщенных участках градиент также мал.

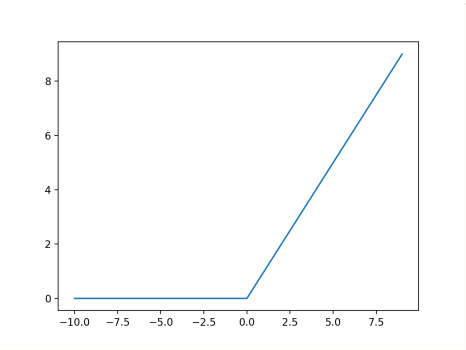


ReLU

$$\mathrm{ReLU}(z) = \max(0, z).$$

При $z > 0$ производная равна 1; при $z < 0$ равна 0.

Если нейрон постоянно находится в отрицательной области, его градиент может оставаться нулевым.



Leaky ReLU и GELU

Leaky ReLU

$$\phi(z) = \begin{cases} z, & z > 0, \\ \alpha z, & z \leq 0. \end{cases}$$

Ненулевой наклон при отрицательных значениях.

GELU

$$\mathrm{GELU}(z) = z\Phi(z).$$

Плавная активация; встречается в трансформерах. Φ — функция распределения стандартной нормали.

Выходной слой и задача

Задача	Выход	Функция потерь
Бинарная классификация	sigmoid	BCE
Несколько классов	softmax	cross-entropy
Регрессия	линейный выход	MSE или MAE

Внутренние и выходной слои могут использовать разные активации: выход должен соответствовать смыслу предсказания.

Обучение сети

Предсказание, ошибка и обновление весов

3

Softmax для нескольких классов

$$p_k = \frac{e^{z_k}}{\sum_j e^{z_j}}.$$

Логиты $z_1, \ldots, z_K$ превращаются в вероятности: $p_k > 0$, $\sum_k p_k = 1$.

Если логиты $(0, 0, 0)$, вероятности равны $(1/3, 1/3, 1/3)$.

Функция потерь

Для объекта с истинной меткой y и предсказанием $\hat{y}$ функция потерь $\ell(y, \hat{y})$ измеряет ошибку.

На обучающей выборке минимизируем среднее:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell\big(y_i, f_\theta(x_i)\big) \longrightarrow \min_\theta .$$

BCE и cross-entropy

Бинарная классификация: $y \in \{0, 1\}$, вероятность класса 1 равна p.

$$\ell = -y \log p - (1 - y) \log(1 - p).$$

Несколько классов: c — правильный класс, p_c — его вероятность.

$$\ell = -\log p_c.$$

Уверенное неверное предсказание штрафуется сильнее.

Потери в регрессии

$$\mathrm{MSE} = \frac{1}{N}\sum_i (y_i - \hat{y}_i)^2, \qquad \mathrm{MAE} = \frac{1}{N}\sum_i |y_i - \hat{y}_i|.$$

MSE сильнее реагирует на крупные ошибки; MAE линейно зависит от их величины.

Huber loss сочетает квадратичный штраф для малых ошибок и линейный — для больших.

Один шаг обучения

1. Выбираем мини-батч объектов.
2. Вычисляем предсказания: **forward**.
3. Считаем среднюю ошибку на батче.
4. Находим производные по параметрам: **backward**.
5. Обновляем веса: $\theta \leftarrow \theta - \eta \nabla_\theta L$.

η — learning rate, размер шага.

Мини-батчи и SGD

$$g_B = \frac{1}{|B|} \sum_{i \in B} \nabla_\theta \ell_i(\theta), \qquad \theta \leftarrow \theta - \eta g_B.$$

B — подмножество объектов. Градиент батча дешевле полного, но изменяется от батча к батчу.

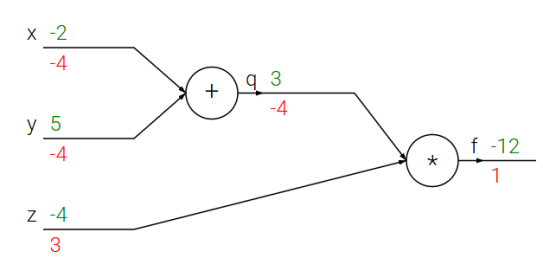
Эпоха — проход по обучающей выборке; шаг оптимизатора — обновление после одного батча.

4

Обратное распространение

Вычислительный граф и цепное правило

Вычислительный граф



Forward: вычисляем значения слева направо. Backward: передаём производные справа налево по цепному правилу.

Цепное правило

Если $u = g(x)$, а $L = f(u)$, то

$$\frac{dL}{dx} = \frac{dL}{du}\frac{du}{dx}.$$

Локальная производная показывает, как изменится выход операции при изменении её входа.

В графе с несколькими путями вклады производных складываются.

Численный пример: forward

Пусть $x = 2$, $w = 3$, $b = 1$, правильный ответ $y = 4$.

$$z = wx + b = 7, \qquad L = (z - y)^2 = 9.$$

Как изменится ошибка при увеличении w? Найдём $\partial L / \partial w$.

Численный пример: backward

$$\frac{\partial L}{\partial z} = 2(z - y) = 6, \qquad \frac{\partial z}{\partial w} = x = 2.$$

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial w} = 12.$$

При $\eta = 0.01$ получаем $w_{\text{new}} = 3 - 0.01 \cdot 12 = 2.88$.

Положительная производная означает: локально рост w увеличивает ошибку.

Градиенты в многослойной сети

$$z^{(l)} = W^{(l)} h^{(l-1)} + b^{(l)}, \qquad h^{(l)} = \phi(z^{(l)}).$$

Градиент предыдущего слоя зависит от градиента следующего слоя, весов и производной активации.

Промежуточные производные переиспользуются: один проход назад вычисляет градиенты всех параметров.

Затухающие градиенты

При обратном проходе производные перемножаются. Если их величины часто меньше 1, сигнал к ранним слоям может стать очень маленьким.

Причины и диагностика: насыщение sigmoid/tanh, глубина сети, инициализация; отслеживаем нормы градиентов.

5

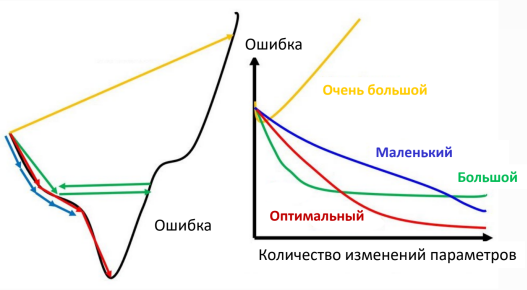
Оптимизация и регуляризация

Как сделать обучение устойчивым

Learning rate

Слишком большой шаг: колебания или расходимость.

Слишком маленький: медленное обучение.



Momentum и Adam

Momentum: накапливаем направление прошлых градиентов и сглаживаем шум.

$$v_t = \beta v_{t-1} + g_t, \qquad \theta_{t+1} = \theta_t - \eta v_t.$$

Adam: учитывает и среднее градиента, и его квадрат; шаг адаптируется по координатам. Оба метода требуют выбора learning rate.

Переобучение

Train loss продолжает уменьшаться, а validation loss после минимума растёт.

Сеть всё лучше подстраивается под обучающие данные, но качество на новых данных ухудшается.

Следим за validation-метрикой и сохраняем лучший checkpoint.

Dropout

Во время обучения случайно обнуляем часть активаций; маска меняется от шага к шагу.

При проверке и применении модели dropout выключен.

Это помогает уменьшить переобучение, но слишком сильный dropout мешает обучению.

Регуляризация и остановка

L2-регуляризация: добавляем штраф $\lambda \|W\|_2^2$ к функции потерь.

Early stopping: останавливаем обучение, когда качество на validation перестаёт улучшаться; возвращаем лучший checkpoint.

Для честной итоговой оценки используем отдельный test.

Теоретический минимум

- Логистическая регрессия, логит, сигмоида, порог, log loss.
- Нейрон, веса, смещение, функция активации.
- Полносвязный слой, скрытый слой, MLP; роль нелинейности.
- Логиты, sigmoid и softmax; BCE, cross-entropy, MSE.
- Функция потерь, мини-батч, эпоха, градиентный спуск.
- Вычислительный граф, цепное правило, forward и backpropagation.
- Learning rate, momentum, Adam, переобучение, dropout и early stopping.

Продолжим изучать глубокое обучение

