

# Функции потерь

Лекция 2

# Зачем нужны функции потерь?

Данные → Модель → Предсказание → Loss →  
→ Обратная связь → Новые параметры

*Loss = мера ошибки. Без неё обучение невозможно*



# Определение функции потерь

$$L(y, \hat{y}) = \ell(y, \hat{y})$$

$y$  – истинное значение

$\hat{y}$  – предсказание модели

$\ell$  – правило вычисления  
ошибки

Свойства функции потерь:

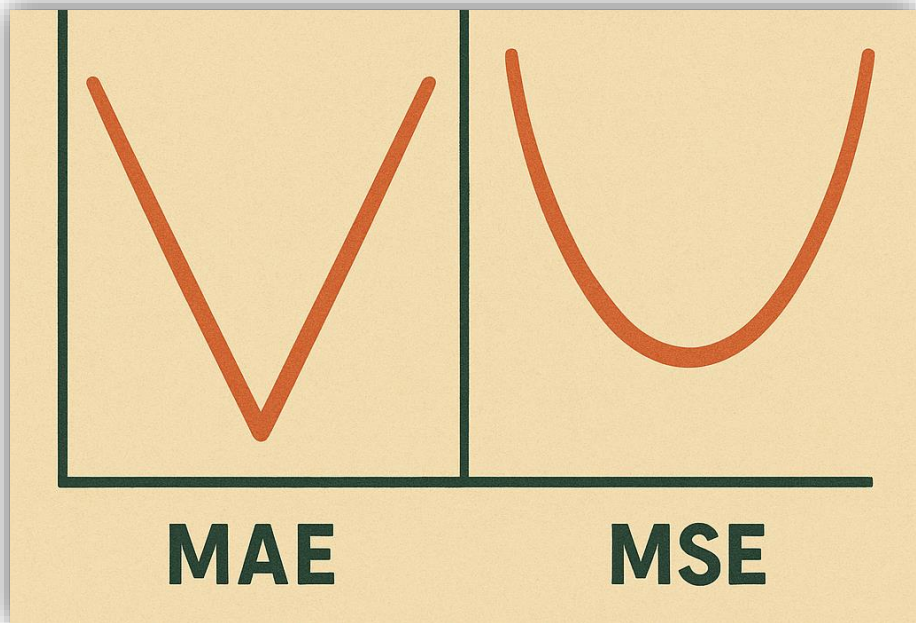
- $L \geq 0$
- $L = 0$
- Дифференцируема
- Интерпретируема

# Функции потерь. MAE и MSE

MAE (средняя абсолютная ошибка)    MSE (среднеквадратичная ошибка)

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$



RMSE  
(корень из среднеквадратичной ошибки)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

# Negative Log-Likelihood (NLL Loss)

$$L_{NLL} = - \sum_{i=1}^n \log p(y_i | x_i, \theta)$$

$$L_{NLL} = - \frac{1}{N} \sum_{i=1}^n \log p(y_i | x_i, \theta)$$

$x_i$  — входные данные,

$y_i$  — истинная метка для примера  $i$ ,

$N$  — количество примеров в выборке,

$\theta$  — параметры(веса) модели,

$p(y_i | x_i, \theta)$  — вероятность правильного класса, предсказанная моделью.

# Кросс-энтропия (Cross-Entropy Loss)

$$L_{CE} = - \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log p_{i,c}$$

$$L_{CE} = - \frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log p_{i,c}$$

$C$  — количество классов в задаче классификации.

$y_{i,c}$  — one-hot индикатор (1, если правильный класс =  $c$ , иначе 0).

$p_{i,c}$  — вероятность для класса  $c$ , которую модель присвоила классу  $c$ .

В сумме остаётся только правильный класс.

# Hinge Embedding Loss

## (Функция потерь для SVM)

$$L_{hinge} = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i \cdot f(x_i))$$

$y_i \in \{-1, +1\}$  — правильный класс,  $f(x_i)$  — предсказание модели

- Используется для задач бинарной классификации
- Наказывает модель, если правильный класс «слишком близко к границе»
- Идея следующая - правильный класс должен иметь минимальный зазор (margin)

# Margin Ranking Loss

$$L = \frac{1}{N} \sum_{i=1}^N \max \left( 0, -y_i \cdot \left( f \left( x_i^{(1)} \right) - f \left( x_i^{(2)} \right) \right) + m \right)$$

$f \left( x_i^{(1)} \right), f \left( x_i^{(2)} \right)$  — предсказания для пары объектов,  
 $y_i \in \{-1, +1\}$  — какой объект должен быть «лучше»,  
 $m$  — margin (обычно = 1).

- Используется для задач ранжирования и сравнения пар объектов
- Цель - правильный объект должен иметь больший скор, чем неправильный, хотя бы на минимальный зазор (margin)
- Штраф: если разница меньше зазора



# Triple Margin Loss

$$L = \frac{1}{N} \sum_{i=1}^N \max(0, \|f(a_i) - f(p_i)\|_2 - \|f(a_i) - f(n_i)\|_2 + m)$$

$a_i$  (anchor) — «якорь», объект, относительно которого сравниваем.

$p_i$  (positive) — положительный пример того же класса, что и anchor.

$n_i$  (negative) — отрицательный пример из другого класса.

$f(\cdot)$  — модель, которая преобразует объект в вектор-признаков (embedding).

$\|\cdot\|_2$  — L2-норма, то есть евклидово расстояние между векторами.

- Используется, когда нужно научить модель сравнивать объекты в эмбединговом пространстве.
- Работает с тройками: anchor (опорный объект), positive (такой же класс) и negative (другой класс).
- Задача: сделать расстояние между anchor и positive меньше, чем между anchor и negative.
- Применяется в задачах распознавания лиц, поиска похожих объектов, рекомендательных системах и metric learning.

# Расстояние Кульбака — Лейблера (KL divergence)

$$D_{KL}(P \parallel Q) = \sum_{i=1}^K P(i) \log \frac{P(i)}{Q(i)}$$

$P$  — истинное распределение (например, one-hot или soft label)

$Q$  — предсказанное моделью распределение.

- Показывает, насколько сильно одно распределение вероятностей отличается от другого.
- Измеряет, сколько «информации» теряется, если вместо реального распределения использовать предсказанное.
- Чем меньше значение, тем ближе предсказание к истине.
- Широко используется в машинном обучении: при обучении вариационных автоэнкодеров, в языковых моделях, при аппроксимации распределений.

# Функции потерь: итоги

Loss = язык обратной связи, который показывает модели, где и насколько она ошиблась

- **Регрессия**
  - MSE / RMSE — сильный штраф за крупные ошибки
  - MAE — устойчивость к выбросам
- **Классификация**
  - NLL / Cross-Entropy — работа с вероятностями классов
  - Hinge — уверенные границы
- **Метрическое обучение**
  - Margin Ranking / Triplet — «свой ближе, чужой дальше»
- **Распределения**
  - KL-расстояние — сравнение формы предсказаний с реальностью

Нет универсального решения — выбор loss зависит от задачи, данных и целей

