

# О калибровке нейронных сетей

Шарипов А.

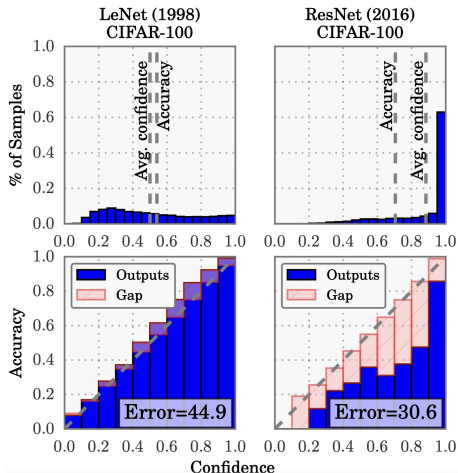
МФТИ, ФПМИ ИВТф

осень 2025

# TL;DR статьи (введение в проблематику)

В последние годы сильно выросла точность (accuracy), но пострадала откалиброванность (calibration).

Сравнение 5-слойной сети LeNet и 110-слойной сети ResNet на CIFAR-100. Мы видим, что LeNet хорошо откалиброван, поскольку уверенность близко приближается к ожидаемой точности. С другой стороны, точность ResNet лучше, но не соответствует его уверенности.



Мы рассматриваем многоклассовую классификацию.

Вход  $X \in \mathcal{X}$  и метки  $Y \in \mathcal{Y} = \{1, \dots, K\}$  — случайные величины, которые удовлетворяют истинному совместному распределению

$$\pi(X, Y) = \pi(Y|X)\pi(X).$$

## Определение

Пусть  $h$  — нейронная сеть с  $h(X) = (\hat{Y}, \hat{P})$ , где  $\hat{Y}$  — предсказание класса, а  $\hat{P}$  — соответствующая уверенность, то есть вероятность корректности.

# TL;DR статьи (определения)

## Определение

Будем говорить, что модель *идеально откалибрована*, если

$$\mathbb{P}(\hat{Y} = Y | \hat{P} = p) = p \quad \forall p \in [0, 1].$$

## Определение

Определим *точность*  $\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}(\hat{y}_i = y_i)$ , где  $\hat{y}_i$  и  $y_i$  — предсказанная и истинная метка класса на  $i$ -ом экземпляре.

## Определение

Определим *среднюю уверенность* в пределах  $B_m$  как

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i, \text{ где } \hat{p}_i \text{ — уверенность на } i\text{-ом экземпляре.}$$

## Определение

Определим *ожидаемую калибровочную ошибку* (ECE) как

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|,$$

где  $n$  — число экземпляров.

## Определение

Определим *максимальную калибровочную ошибку* (MCE) как

$$\text{MCE} = \max_{m \in \{1, \dots, M\}} |\text{acc}(B_m) - \text{conf}(B_m)|.$$

## Examples

Идеальная модель будет иметь  $\text{acc}(B_m) = \text{conf}(B_m) \forall m \in \{1, \dots, M\}$ .

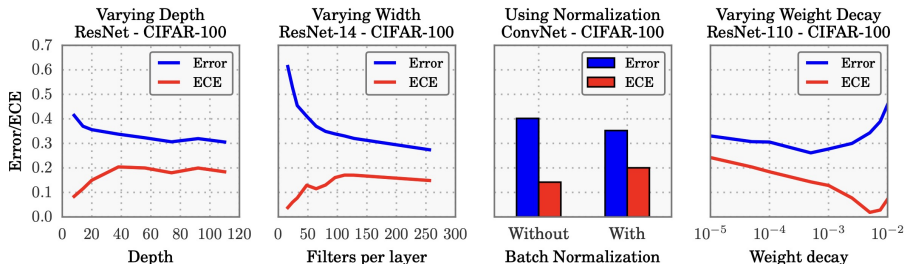
## Examples

Идеальная модель будет иметь  $\text{MCE} = \text{ECE} = 0$ .

# TL;DR статьи (причины некалиброванности)

- Мощность модели;
- Пакетная нормализация;
- Уменьшение весов;
- NLL.

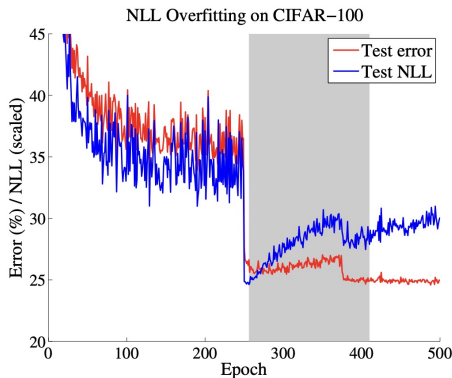
Погрешность и ECE в зависимости от глубины и ширины в сети ResNet, обученной на CIFAR-100:



# TL;DR статьи (причины некалиброванности)

- Мощность модели;
- Пакетная нормализация;
- Уменьшение весов;
- NLL.

Тестовая ошибка и NLL на 110-слойной ResNet с переменной глубиной на CIFAR-100 во время обучения.



# TL;DR статьи (методы калибровки – 1)

Для начала разберемся с бинарными классификаторами.  
Для простоты считаем, что ответ модели – уверенность в положительном классе.

## Histogram binning

$$\min_{\theta_1, \dots, \theta_M} \sum_{m=1}^M \sum_{i=1}^n \mathbb{I}(a_m \leq \hat{p}_i < a_{m+1}) (\theta_m - y_i)^2.$$

## Isotonic regression

$$\min_{\substack{\theta_1, \dots, \theta_M \\ a_1, \dots, a_{M+1}}} \sum_{m=1}^M \sum_{i=1}^n \mathbb{I}(a_m \leq \hat{p}_i < a_{m+1}) (\theta_m - y_i)^2,$$

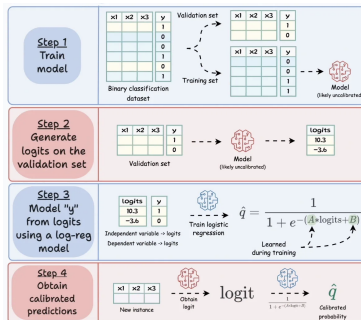
при условии  $0 = a_1 \leq a_2 \leq \dots \leq a_{M+1} = 1, \theta_1 \leq \theta_2 \leq \dots \leq \theta_M$ .

## Bayesian Binning into Quantiles (BBQ)

$$\mathbb{P}(S = s|D) = \frac{\mathbb{P}(D|S = s)}{\sum_{s' \in S} \mathbb{P}(D|S = s')}.$$

## Platt scaling

Логистической регрессией ищем  $a, b \in \mathbb{R}$ . Далее  $\hat{q} = \sigma(az + b)$ .



## TL;DR статьи (мультиклассовый случай)

Теперь  $K > 2$ . Для входа  $x_i$  выход  $\hat{y}_i$ , а значение уверенности  $\hat{p}_i$ . В свою очередь  $z_i$  — логиты, где  $\hat{y}_i = \operatorname{argmax}_k z_i^{(k)}$  и  $\hat{p}_i$  обычно получается из softmax функции  $\sigma_{SM}$ :

$$\sigma_{SM}(z_i)^{(k)} = \frac{\exp(z_i^{(k)})}{\sum_{j=1}^K \exp(z_i^{(j)})}, \quad \hat{p}_i = \max_k \sigma_{SM}(z_i)^{(k)}.$$

Цель получить откалиброванную уверенность  $\hat{q}_i$  и (возможно новое) предсказание класса  $\hat{y}_i'$  от  $y_i, \hat{y}_i, \hat{p}_i, z_i$ .

Extension of binning methods (может быть применено к histogram binning, isotonic regression, and BBQ.)

Рассмотрим задачу как  $K$  one-versus-all задач. Вектор вероятностей

$[\hat{q}_i^{(1)}, \dots, \hat{q}_i^{(K)}]$ , где  $\hat{q}_i^{(k)}$  — откалиброванная вероятность класса  $k$ .

$\hat{y}_i'$  —  $\operatorname{argmax}$  вектора, а  $\hat{q}_i'$  —  $\max$  вектора, нормированное на  $\sum_{k=1}^K \hat{q}_i^{(k)}$ .

## TL;DR статьи (мультиклассовый случай)

### Matrix and vector scaling (мультиклассовые расширения для Platt scaling)

Линейное преобразование  $Wz_i + b$  для логитов

$$\hat{q}_i = \max_k \sigma_{SM}(Wz_i + b)^{(k)}, \quad \hat{y}_i' = \operatorname{argmax}_k (Wz_i + b)^{(k)}.$$

Параметры  $W$  и  $b$  оптимизированны с учетом NLL на валидационном наборе.

### Temperature scaling (простейшее расширение для Platt Scaling)

Один скалярный параметр  $T > 0$  для всех классов. Для логита  $z_i \hookrightarrow$

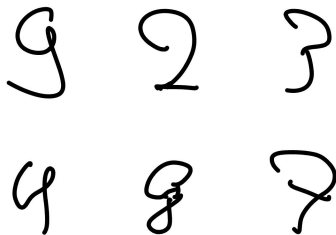
$$\hat{q}_i = \max_k \sigma_{SM}(z_i/T)^{(k)}.$$

Параметр  $T$  оптимизирован с учетом NLL на валидационном наборе.

# Практическая часть (предварительный анализ)

Обучим ResNet-18 и LeNet на MNIST, предварительно упростив задачу до бинарной классификации (9 vs не-9), и посмотрим что происходит с калибровкой там.

Почему была выбрана 9 наглядно:



# Практическая часть (предварительный анализ)

Сделаем Reliability Diagram:

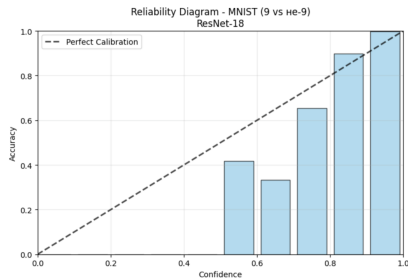
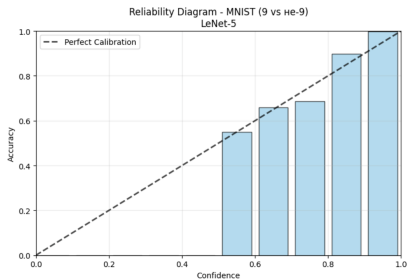
```
bins = np.linspace(0, 1, M + 1)

for i in range(M):
    low = bins[i]
    high = bins[i + 1]
    bin_mask = (confidences >= low) & (confidences <
        high)
    bin_size = np.sum(bin_mask)

    if bin_size > 0:
        bin_acc = np.mean(y_true[bin_mask] == y_pred[
            bin_mask])
        bin_conf = np.mean(confidences[bin_mask])
    else:
        bin_acc = 0
        bin_conf = (low + high) / 2
```

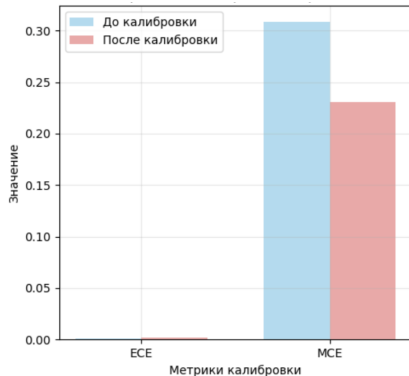
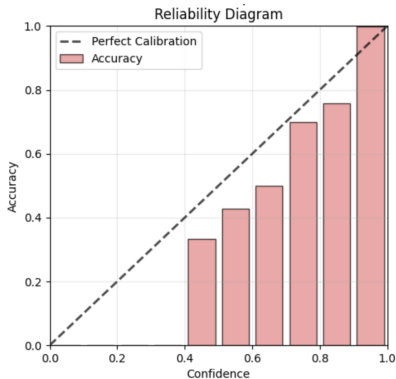
# Практическая часть (предварительный анализ)

Соответственно имеем



# Практическая часть (Isotonic Regression)

Возьмем реализацию из `sklearn.isotonic IsotonicRegression`

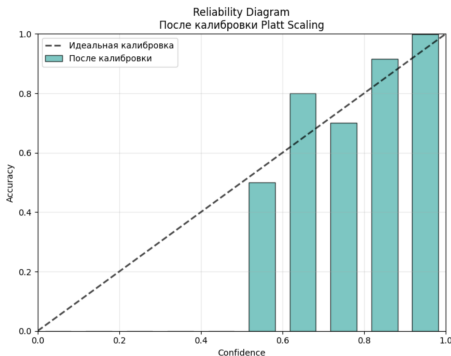


$$\delta ECE \approx -23\%, \delta MCE \approx 25\%.$$

# Практическая часть (Platt Scaling)

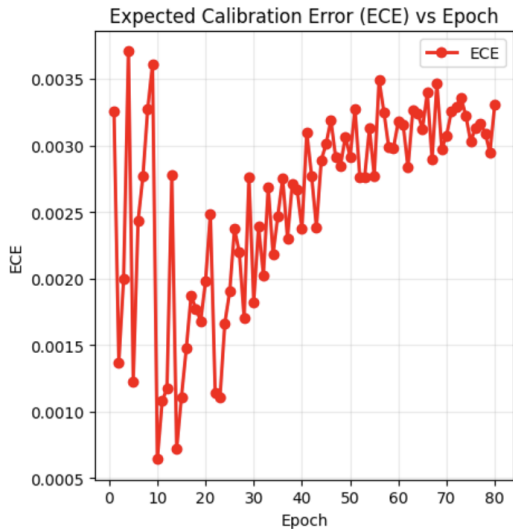
Возьмем реализацию

- `sklearn.calibration CalibratedClassifierCV`
- `sklearn.linear_model LogisticRegression`



$\delta ECE \approx 35\%$ .

# Практическая часть (one more thing)



# Практическая часть (доп)

Также рассмотрим LeNet и ResNet на MNIST в задаче многоклассовой классификации. Соответственно имеем

