

Глубинное обучение

Оптимизация. Обучение нейросетей.

Ирина Сапарина

Оптимизация

Stochastic Gradient Descent

$$\frac{1}{n} \sum_i L(y_i, f(x_i, \theta)) \rightarrow \min_{\theta}$$

Mini-batch SGD:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{m} \sum_{i=1}^m \frac{dL(y_i, f(x_i, \theta))}{d\theta}$$

одно обновление –
 m примеров (батч)

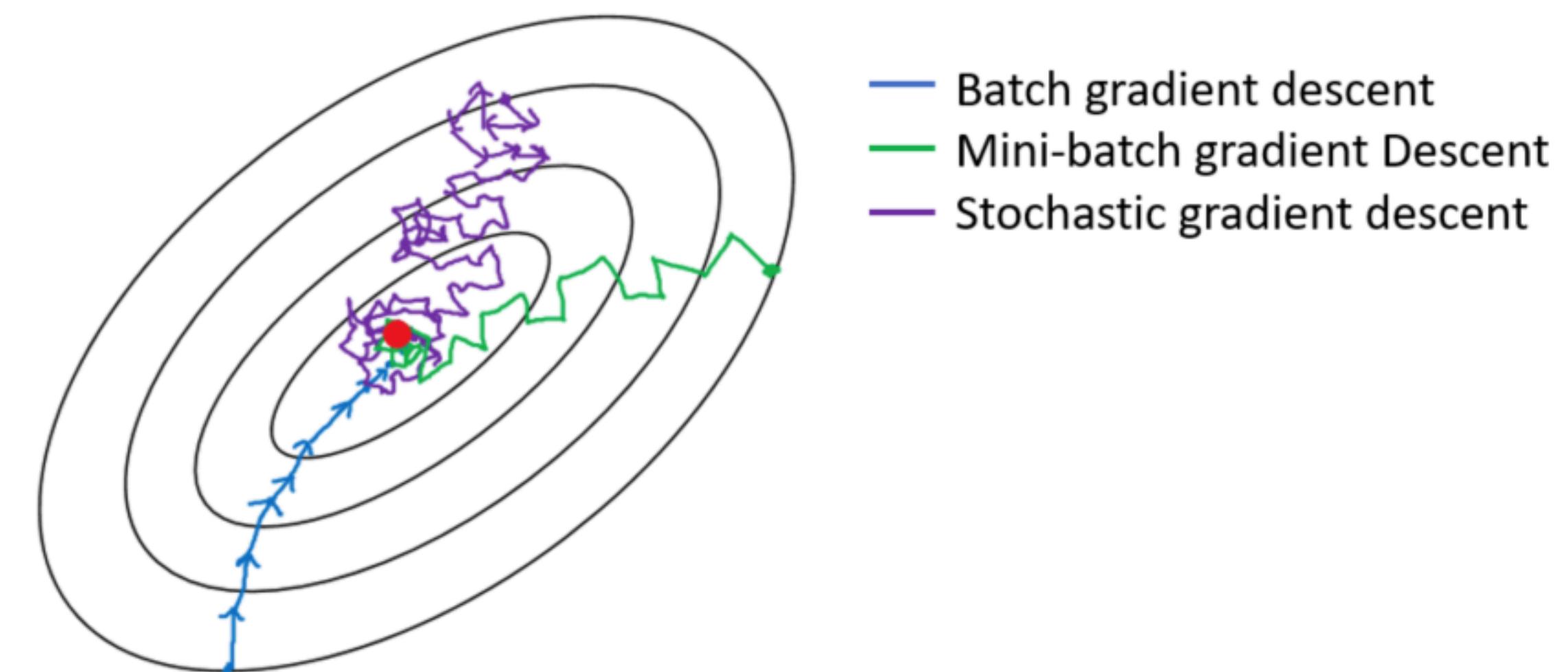


Image credit

Stochastic Gradient Descent

Проблемы:

- Градиент может быть шумным
- LR одинаковый для всех параметров и данных
- Можно застрять в локальном минимуме или седловой точке

Stochastic Gradient Descent

Проблемы:

- Градиент может быть шумным
- LR одинаковый для всех параметров и данных
- Можно застрять в локальном минимуме или седловой точке



SGD

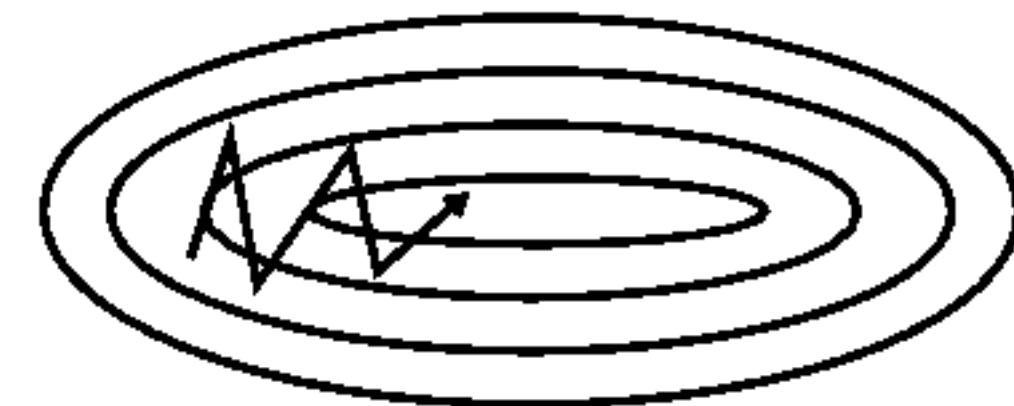
Stochastic Gradient Descent

Проблемы:

- Градиент может быть шумным
- LR одинаковый для всех параметров и данных
- Можно застрять в локальном минимуме или седловой точке



SGD



SGD + momentum

Stochastic Gradient Descent + Momentum

SGD

$$\theta_{t+1} = \theta_t - \alpha \frac{dL(\theta)}{d\theta}$$

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

Stochastic Gradient Descent + Momentum

SGD

$$\theta_{t+1} = \theta_t - \alpha \frac{dL(\theta)}{d\theta}$$

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

v - “скорость”

γ - “трение”, обычно = 0.9 ... 0.99

Stochastic Gradient Descent + Momentum

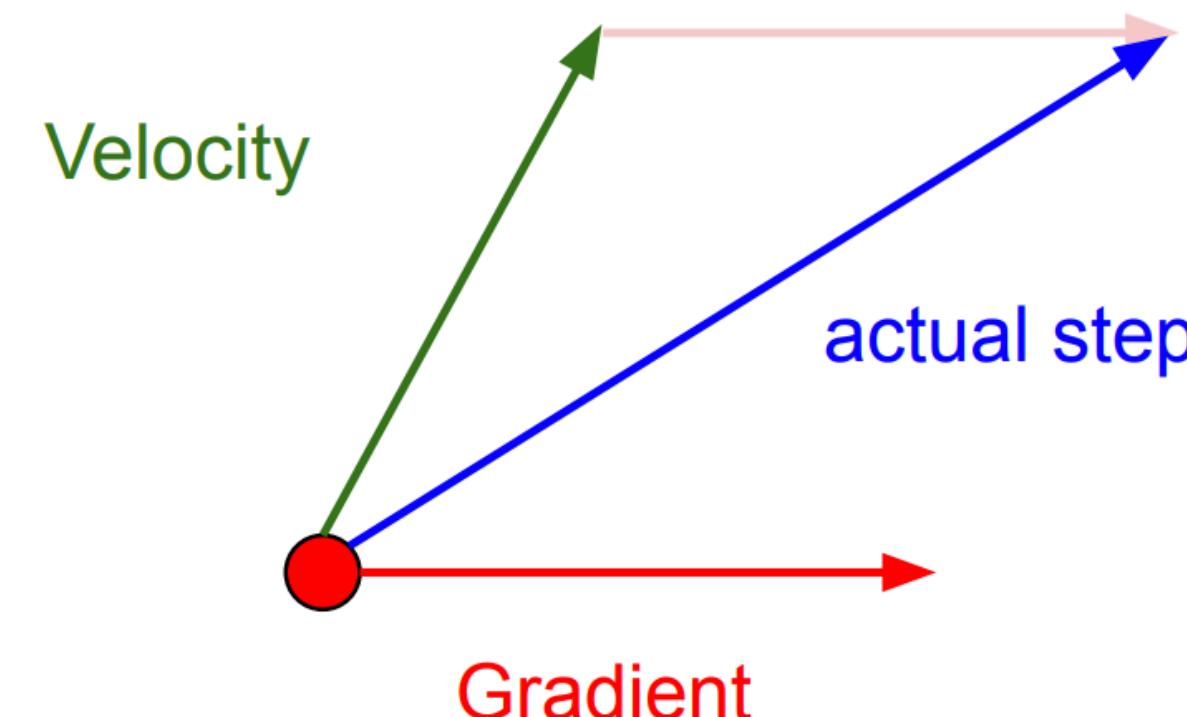
SGD

$$\theta_{t+1} = \theta_t - \alpha \frac{dL(\theta)}{d\theta}$$

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$



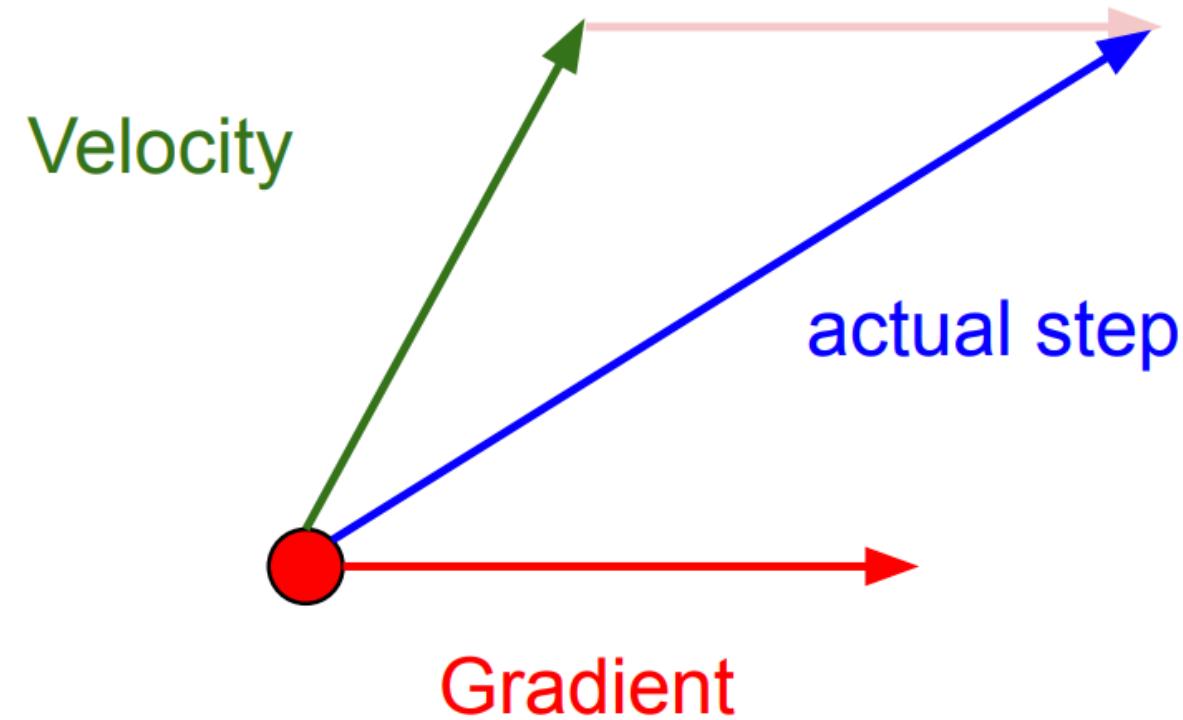
[Image credit](#)

Nesterov Momentum

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$



SGD + Nesterov momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta - \gamma v_{t-1})}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

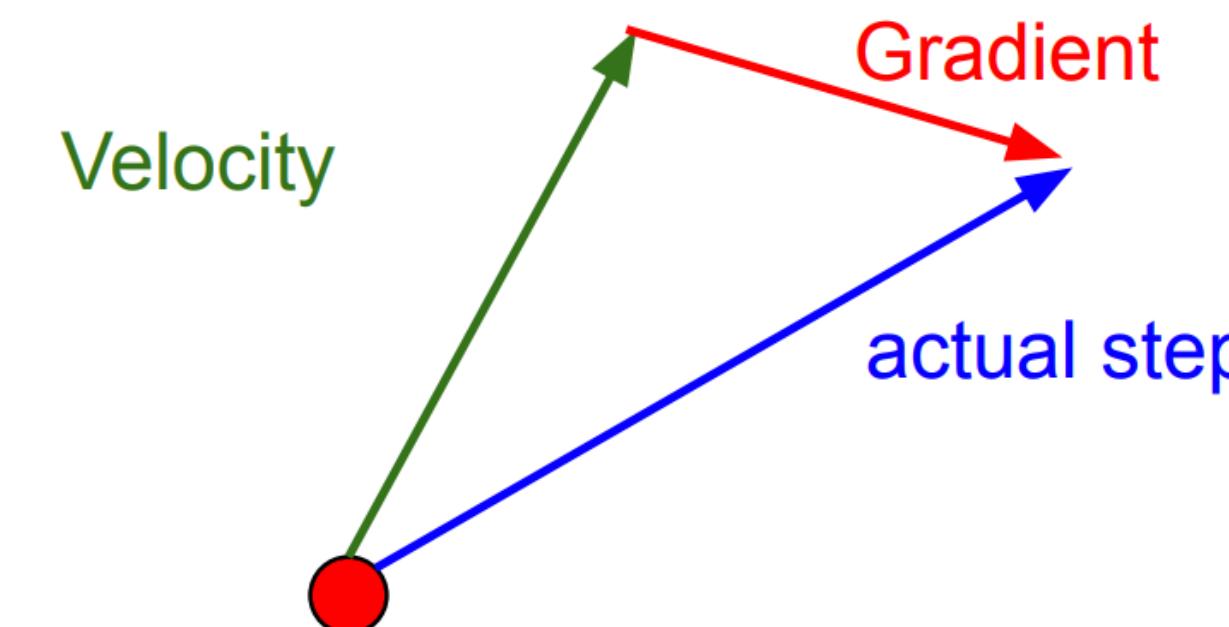


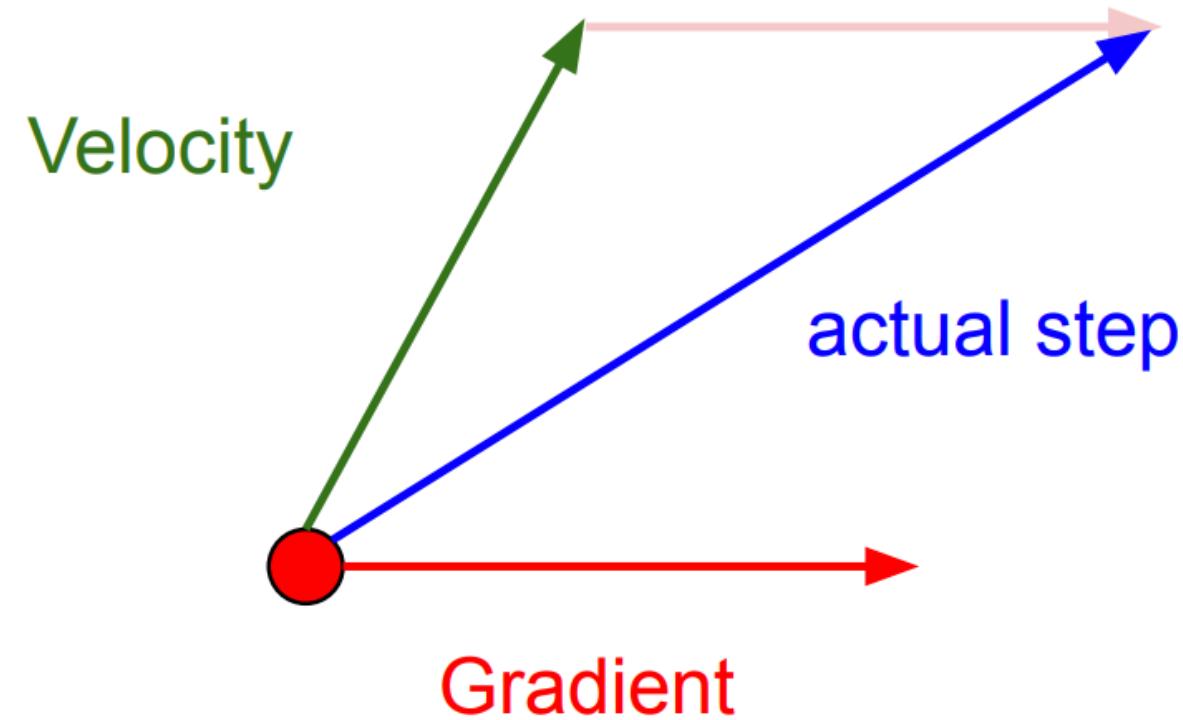
Image credit

Nesterov Momentum

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$



SGD + Nesterov momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta - \gamma v_{t-1})}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

оцениваем, какие
параметры будут

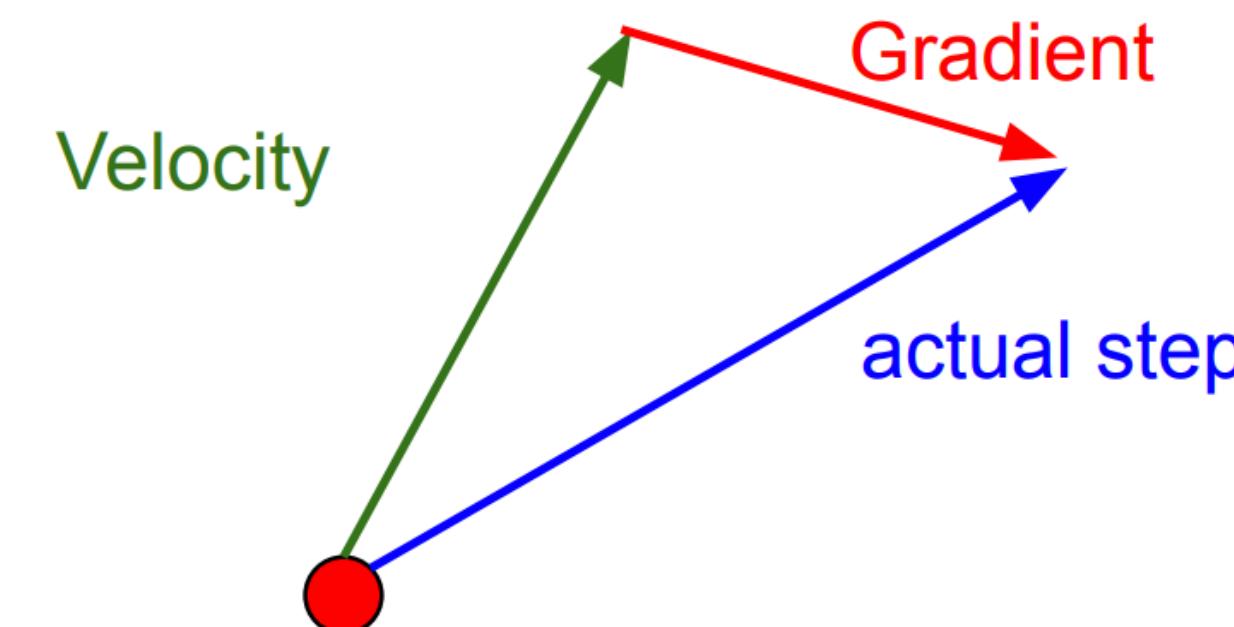
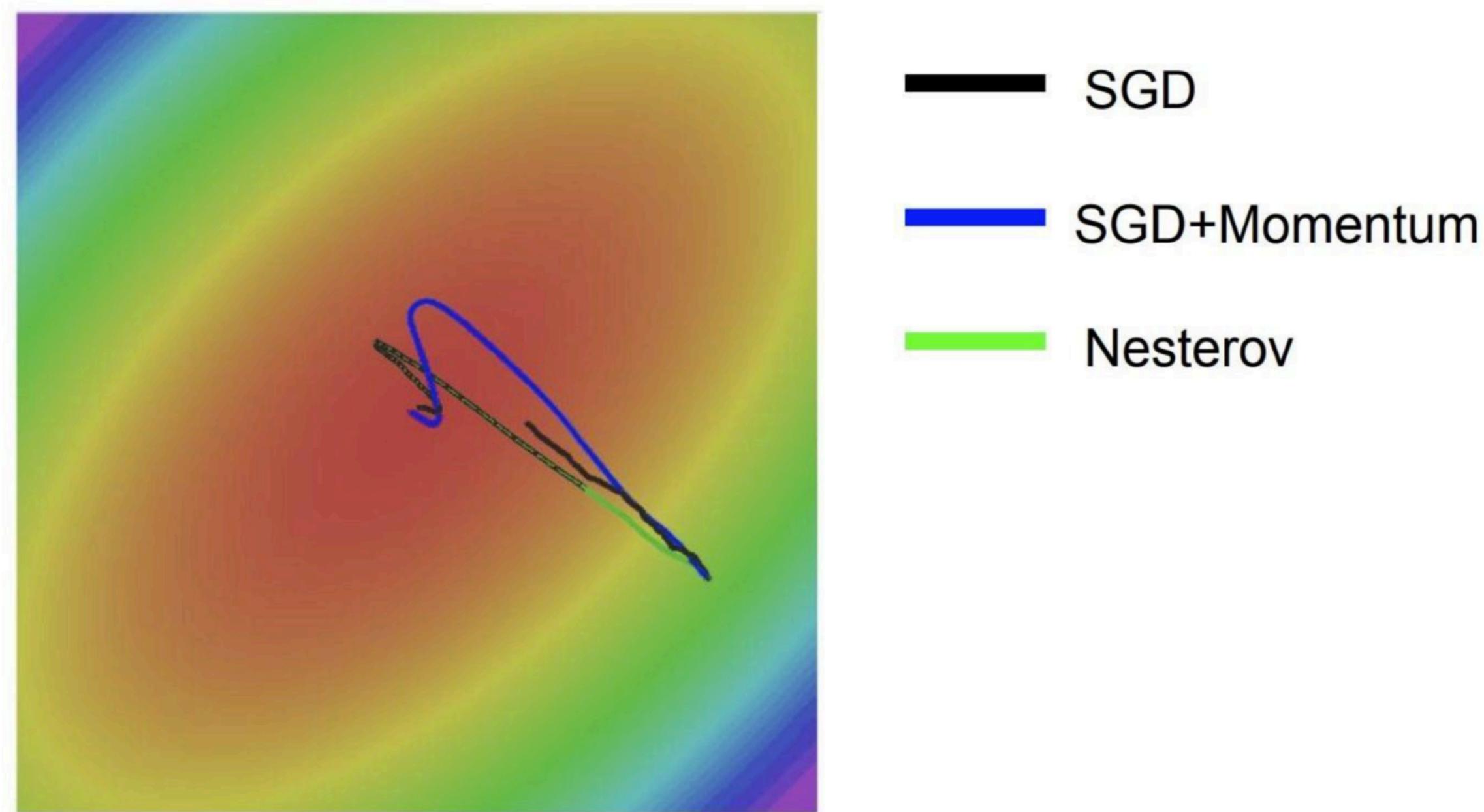


Image credit

Nesterov Momentum



[Image credit](#)

Stochastic Gradient Descent

Проблемы:

- Градиент может быть шумным
- LR одинаковый для всех параметров и данных
- Можно застрять в локальном минимуме или седловой точке

AdaGrad

Идея: адаптивный learning rate

- небольшой lr для часто обновляемых параметров и большой для редких

AdaGrad

Идея: адаптивный learning rate

- небольшой lr для часто обновляемых параметров и большой для редких

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

Градиент для i -го параметра
на шаге t

AdaGrad

Идея: адаптивный learning rate

- небольшой lr для часто обновляемых параметров и большой для редких

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

Градиент для i -го параметра
на шаге t

$$G_{t,i} = \sum_{\tau=1}^t g_{\tau,i}^2$$

“кэш” градиентов

AdaGrad

Идея: адаптивный learning rate

- небольшой lr для часто обновляемых параметров и большой для редких

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

Градиент для i -го параметра
на шаге t

$$G_{t,i} = \sum_{\tau=1}^t g_{\tau,i}^2$$

“кэш” градиентов

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

AdaGrad

Идея: адаптивный learning rate

- небольшой lr для часто обновляемых параметров и большой для редких

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

Градиент для i -го параметра
на шаге t

$$G_{t,i} = \sum_{\tau=1}^t g_{\tau,i}^2$$

“кэш” градиентов

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

AdaGrad

Идея: адаптивный learning rate

- небольшой lr для часто обновляемых параметров и большой для редких

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

Градиент для i -го параметра
на шаге t

$$G_{t,i} = \sum_{\tau=1}^t g_{\tau,i}^2$$

“кэш” градиентов

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

может стать ~ 0

RMSProp

Идея: адаптивный learning rate

- небольшой lr для часто обновляемых параметров и большой для редких

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

Градиент для i -го параметра
на шаге t

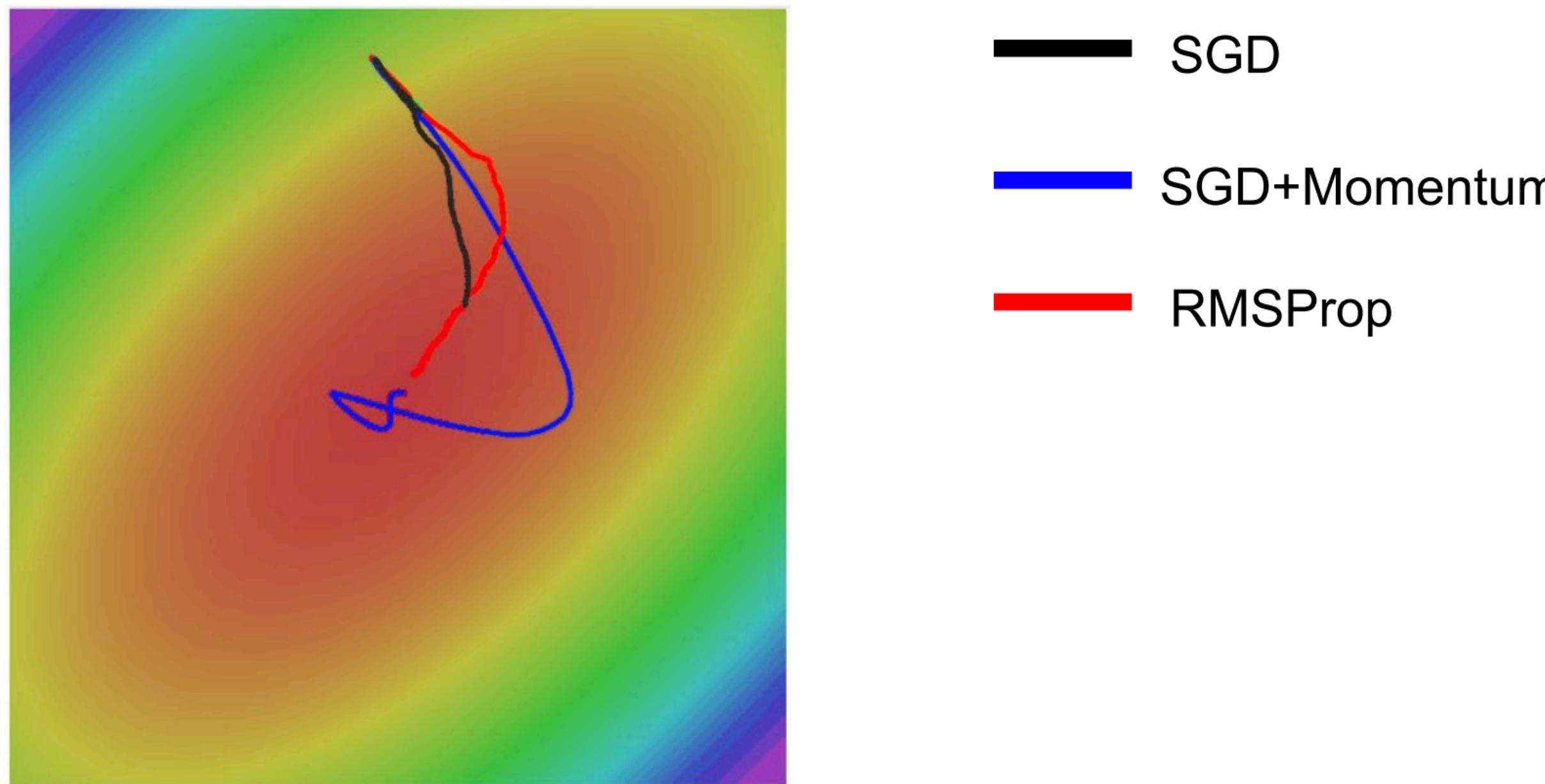
$$G_{t,i} = \beta G_{t-1,i} + (1 - \beta) g_{t,i}^2$$

“кэш” градиентов, **exponential smoothing**

$$\beta = 0.9$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

RMSProp



[Image credit](#)

Adam

Соединим идеи AdaGrad/RMSProp и Momentum

RMSProp

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

$$G_{t,i} = \beta G_{t-1,i} + (1 - \beta) g_{t,i}^2$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

Adam

Соединим идеи AdaGrad/RMSProp и Momentum

RMSProp

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

$$G_{t,i} = \beta G_{t-1,i} + (1 - \beta) g_{t,i}^2$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

Adam

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{G_t + \epsilon}} v_t$$

Adam

Соединим идеи AdaGrad/RMSProp и Momentum

RMSProp

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

$$G_{t,i} = \beta G_{t-1,i} + (1 - \beta) g_{t,i}^2$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

SGD + momentum

$$v_t = \gamma v_{t-1} + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

Adam

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{G_t + \epsilon}} v_t$$

Adam

Соединим идеи AdaGrad/RMSProp и Momentum

RMSProp

$$g_{t,i} := \frac{dL(\theta_{t,i})}{d\theta_{t,i}}$$

$$G_{t,i} = \beta G_{t-1,i} + (1 - \beta) g_{t,i}^2$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

SGD + momentum

$$v_{t+1} = \gamma v_t + \alpha \frac{dL(\theta)}{d\theta}$$

$$\theta_{t+1} = \theta_t - v_t$$

Adam

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t$$

$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{G_t + \epsilon}} v_t$$

Adam

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t$$

$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t^2$$

$$v_0 = G_0 = 0$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{G_t + \epsilon}} v_t$$

Какие будут первые шаги?

Adam

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t$$

$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t^2$$

$$v_0 = G_0 = 0$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_1^t} \quad \hat{G}_t = \frac{G_t}{1 - \beta_2^t}$$

bias correction

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{G}_t + \epsilon}} \hat{v}_t$$

Adam

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t$$

$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

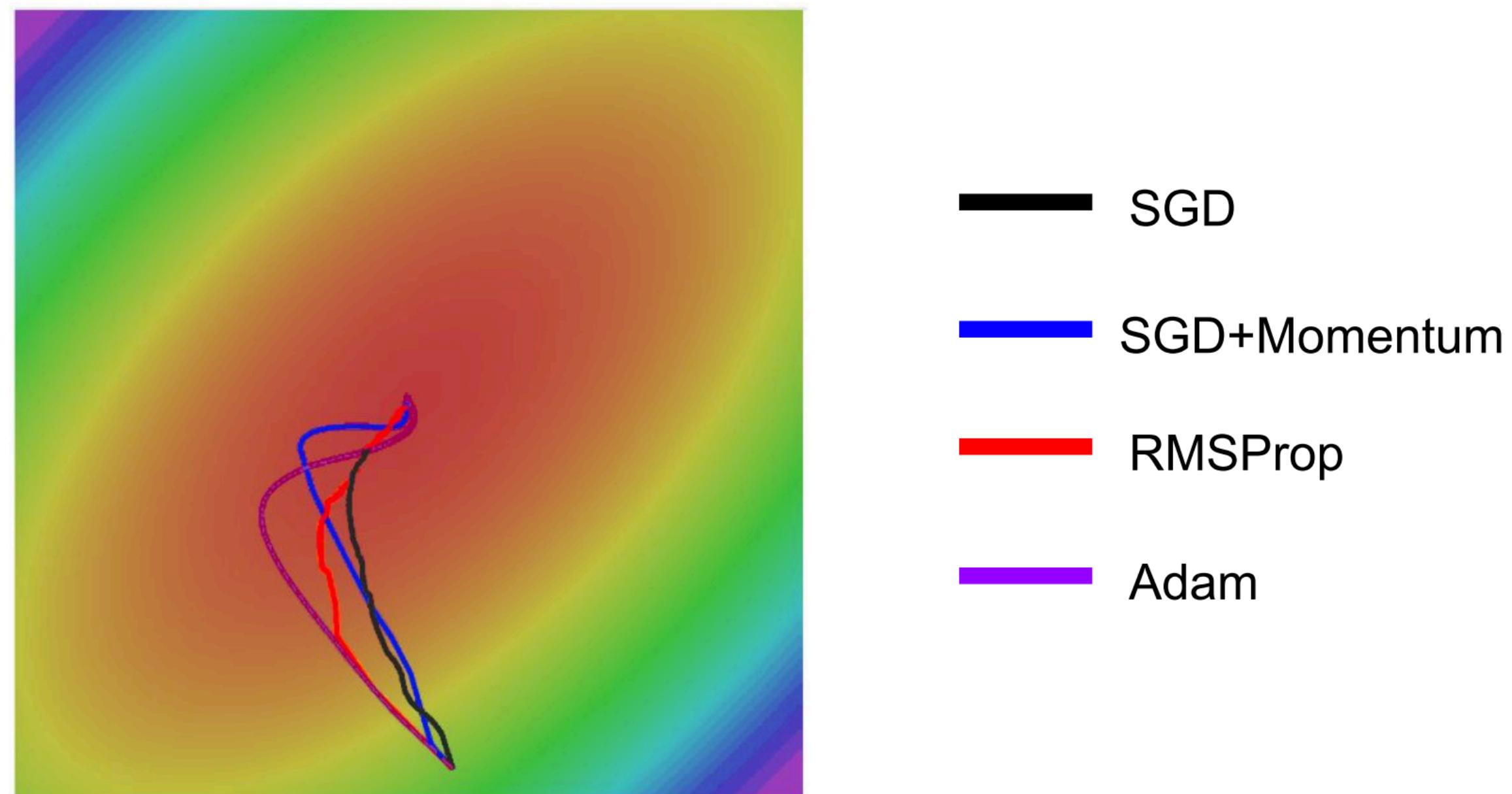
$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t^2$$

$$v_0 = G_0 = 0$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_1^t} \quad \quad \hat{G}_t = \frac{G_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{G}_t + \epsilon}} \hat{v}_t$$

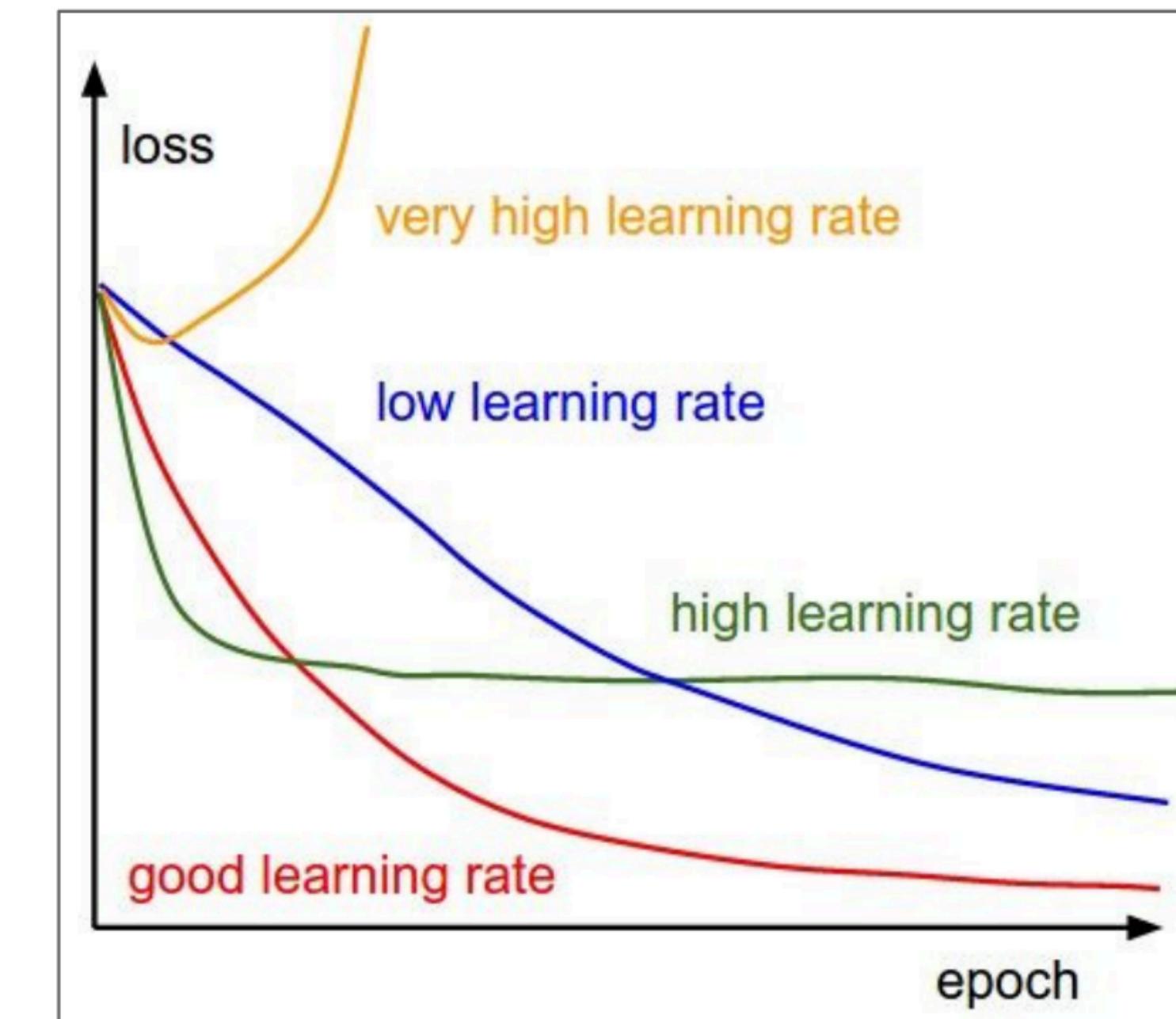
Adam



[Image credit](#)

Выводы

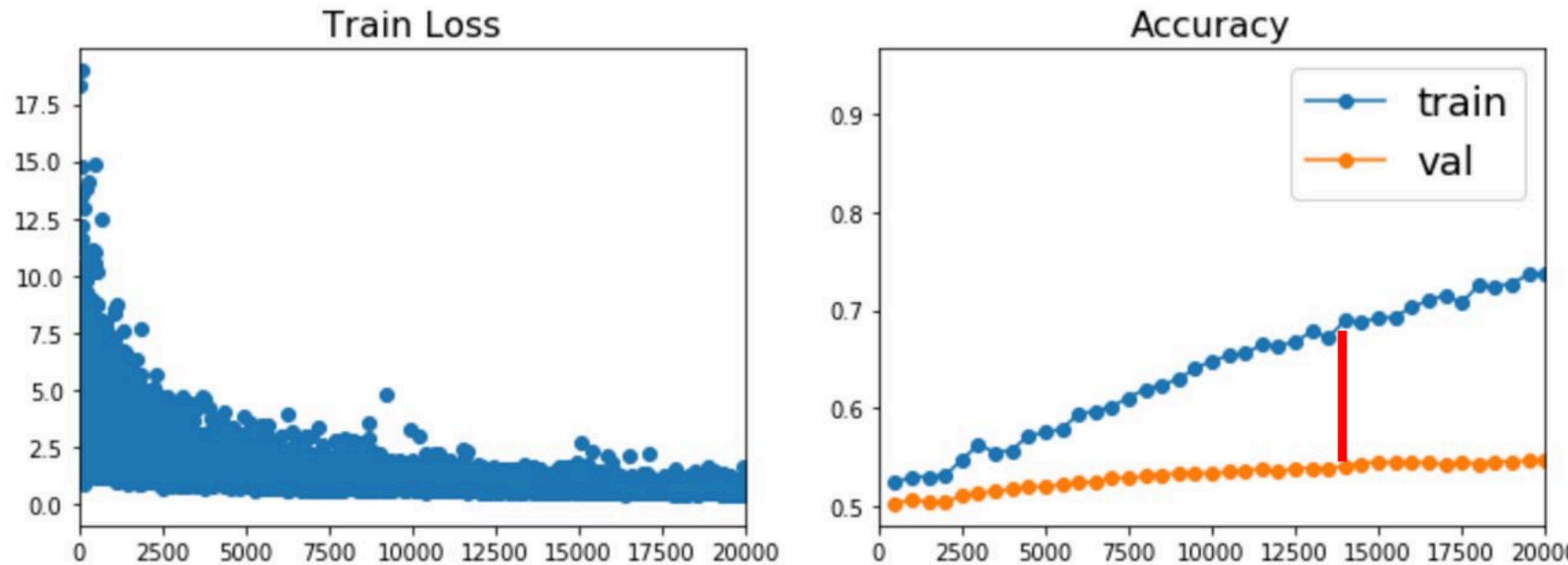
- Adam - хороший выбор для начала
- Learning rate - важный параметр
- LR Scheduler может помочь



[Image credit](#)

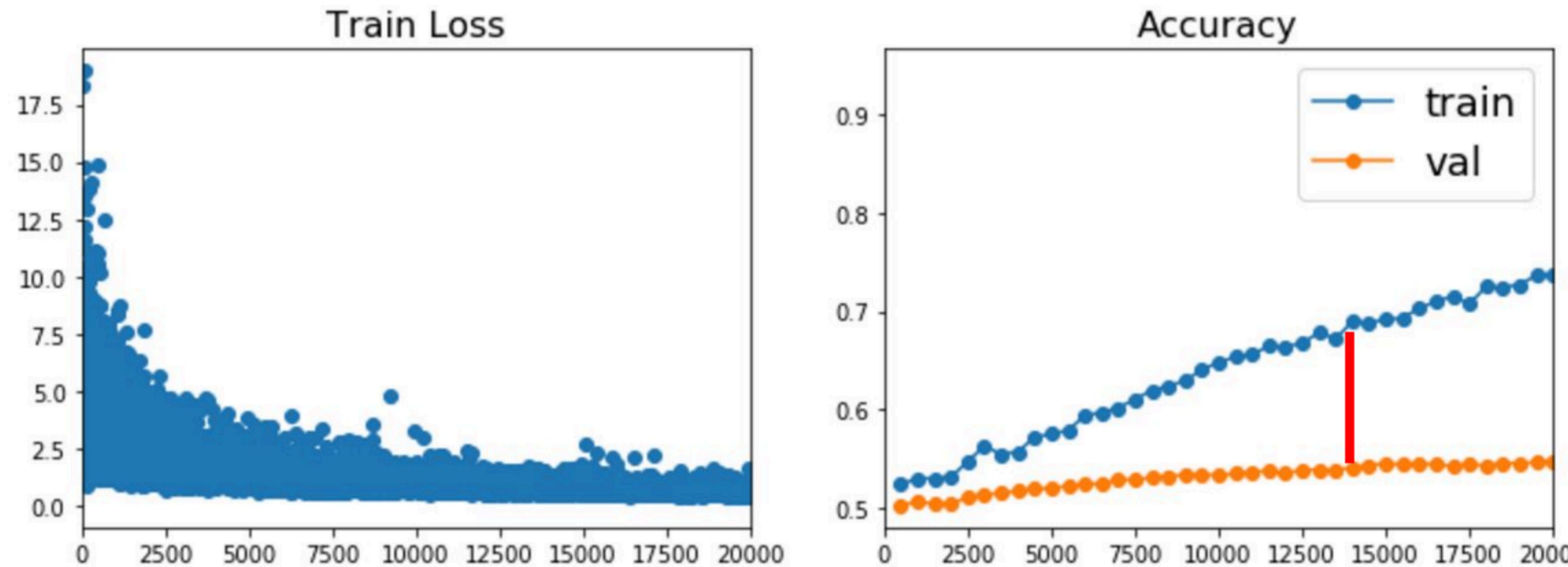
Обучение нейросетей

Обучение



Train loss падает, метрика на train улучшается - модель обучается!

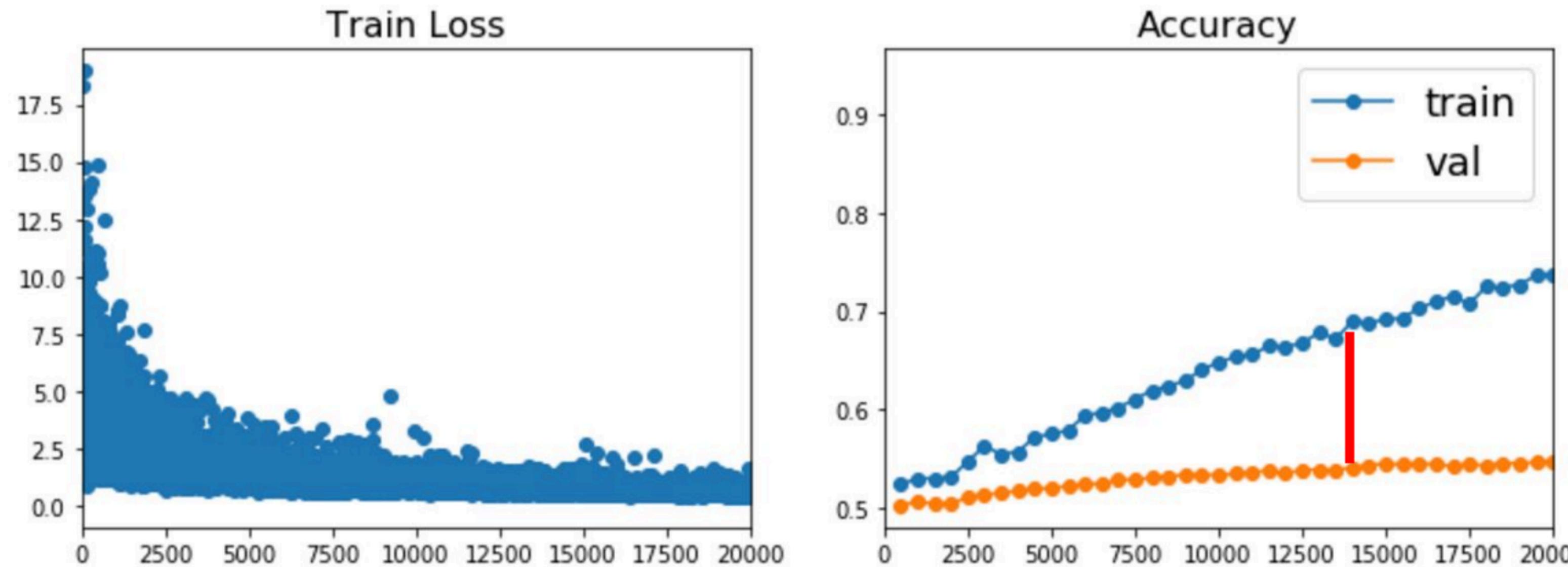
Обучение



Train loss падает, метрика на train улучшается - модель обучается!

Проблема: на валидации качество сильно хуже

Обучение



Train loss падает, метрика на train улучшается - модель обучается!

Проблема: на валидации качество сильно хуже

Переобучение, нужна регуляризация

Регуляризация

$$\frac{1}{n} \sum_i L(y_i, f(x_i, \theta)) + \lambda R(\theta) \rightarrow \min_{\theta}$$

$$R(\theta) = \sum_l \|\theta_l\|^2$$

L2-регуляризация (weight decay)

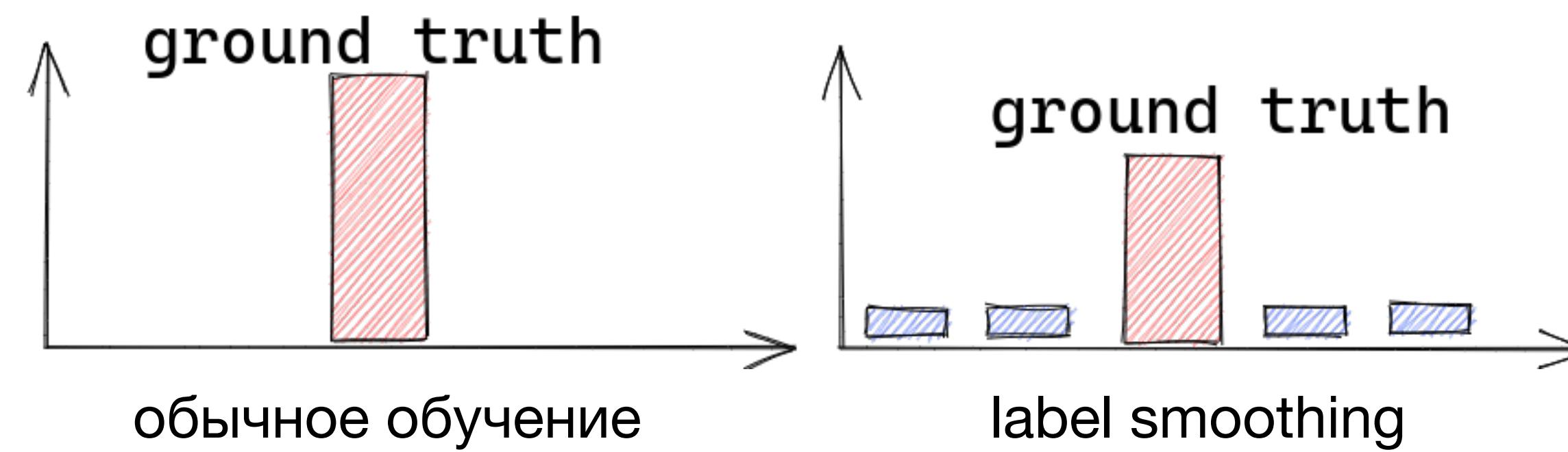
$$R(\theta) = \sum_l |\theta_l|$$

L1-регуляризация

$$R(\theta) = \sum_l |\theta_l| + \|\theta_l\|^2$$

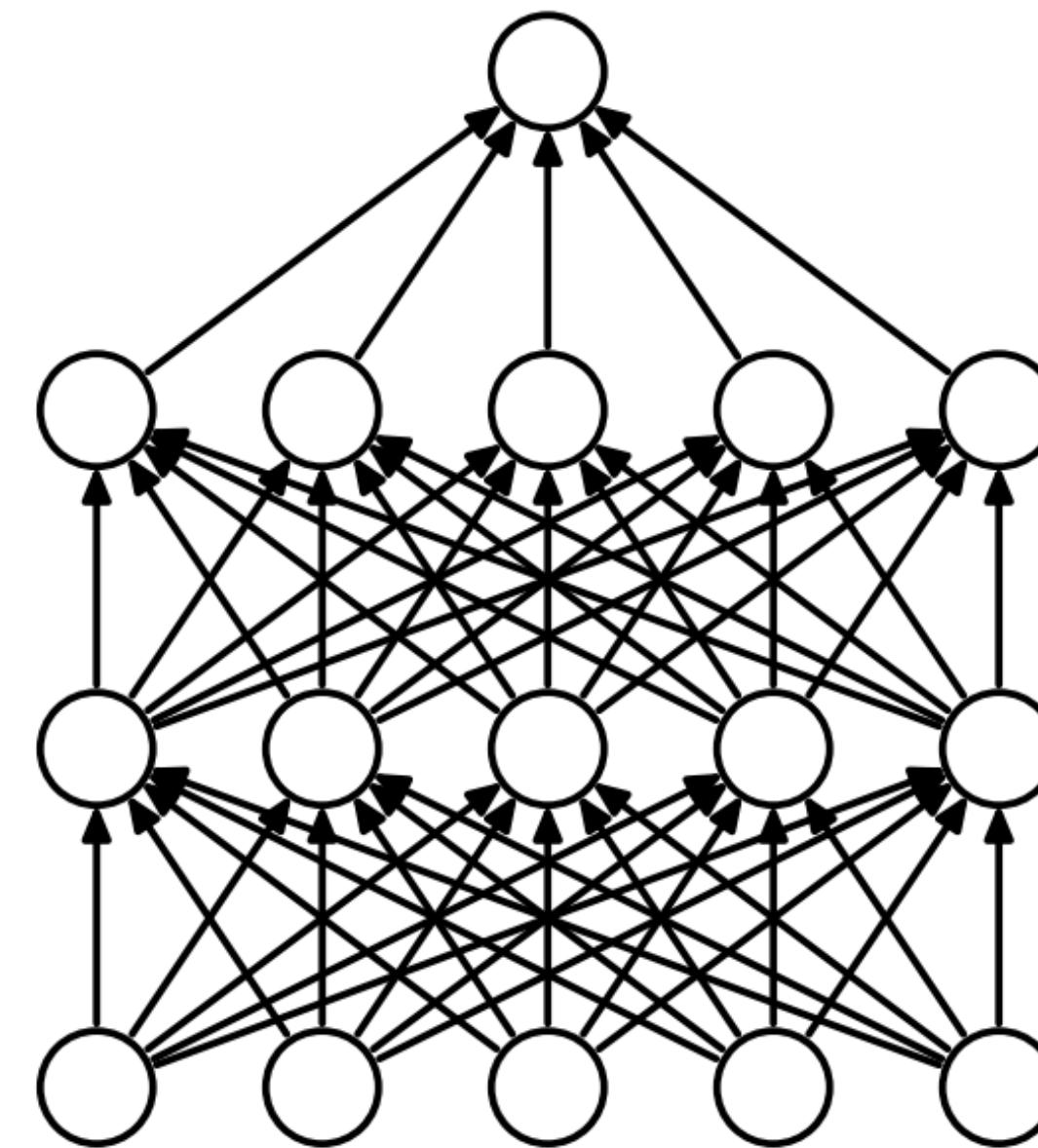
Elastic Net (L1 + L2 -регуляризация)

Регуляризация: label smoothing

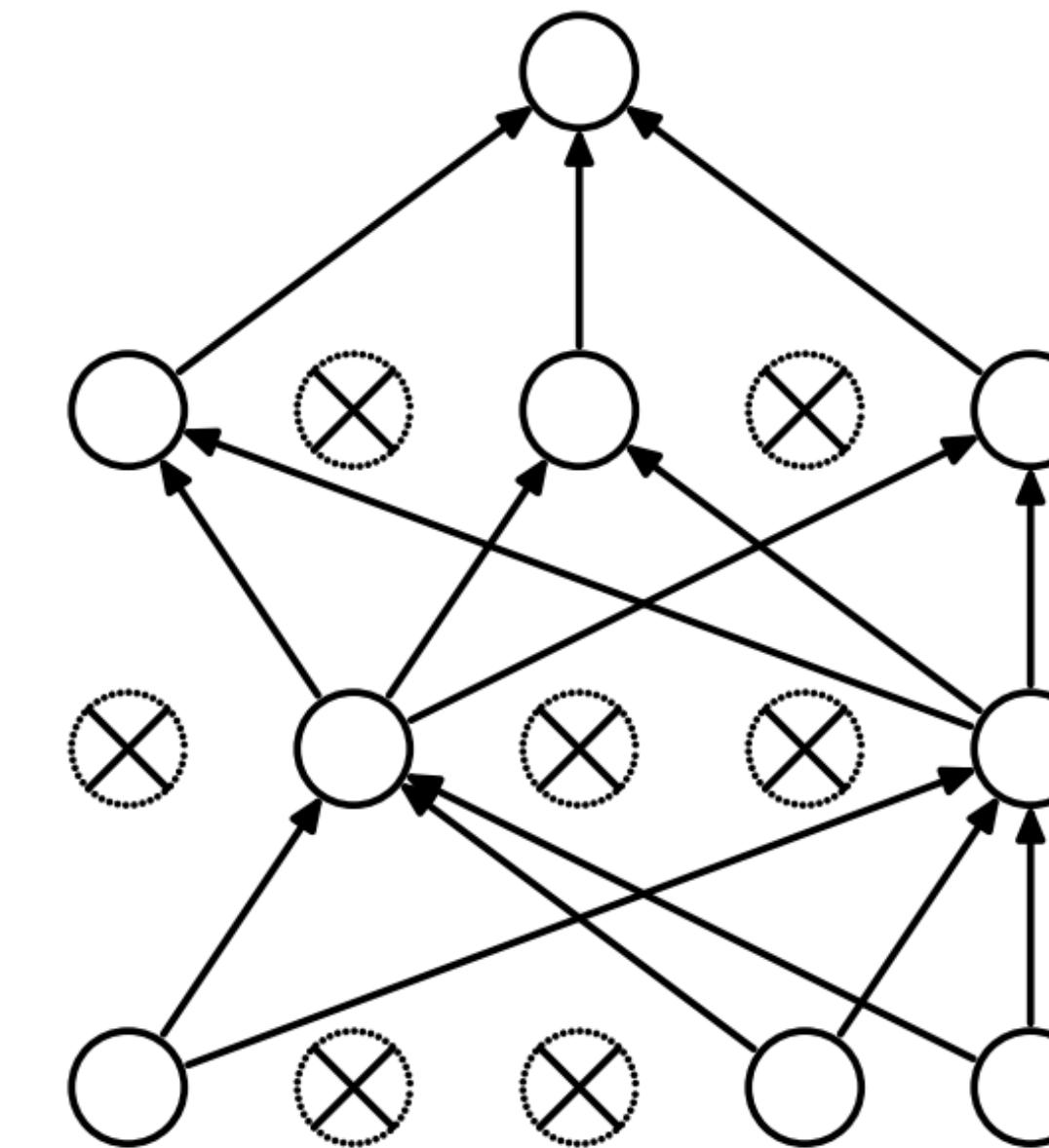


Target распределение: вероятность правильной метки α , остальных $1 - \alpha$

Регуляризация: dropout



(a) Standard Neural Net

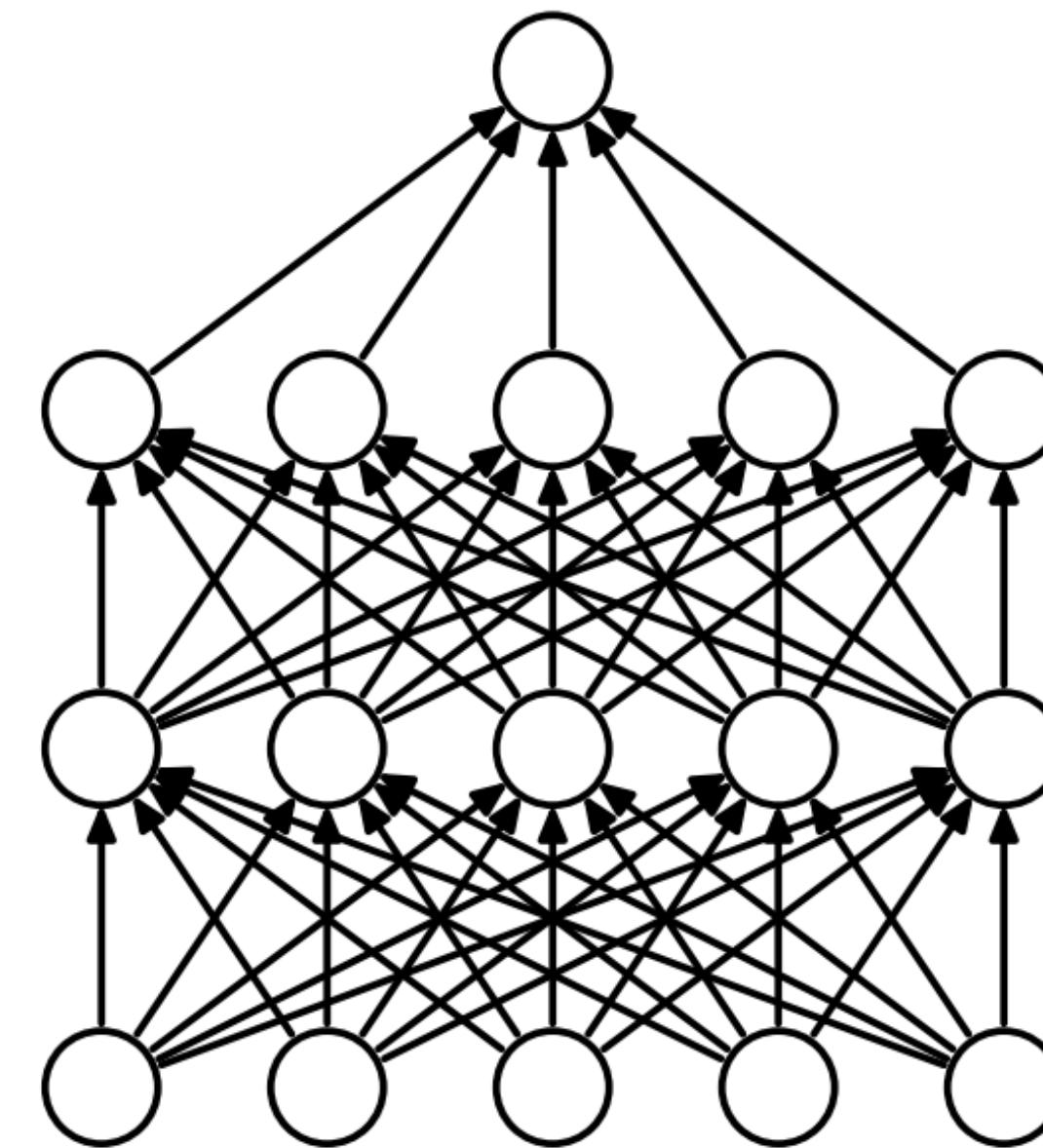


(b) After applying dropout.

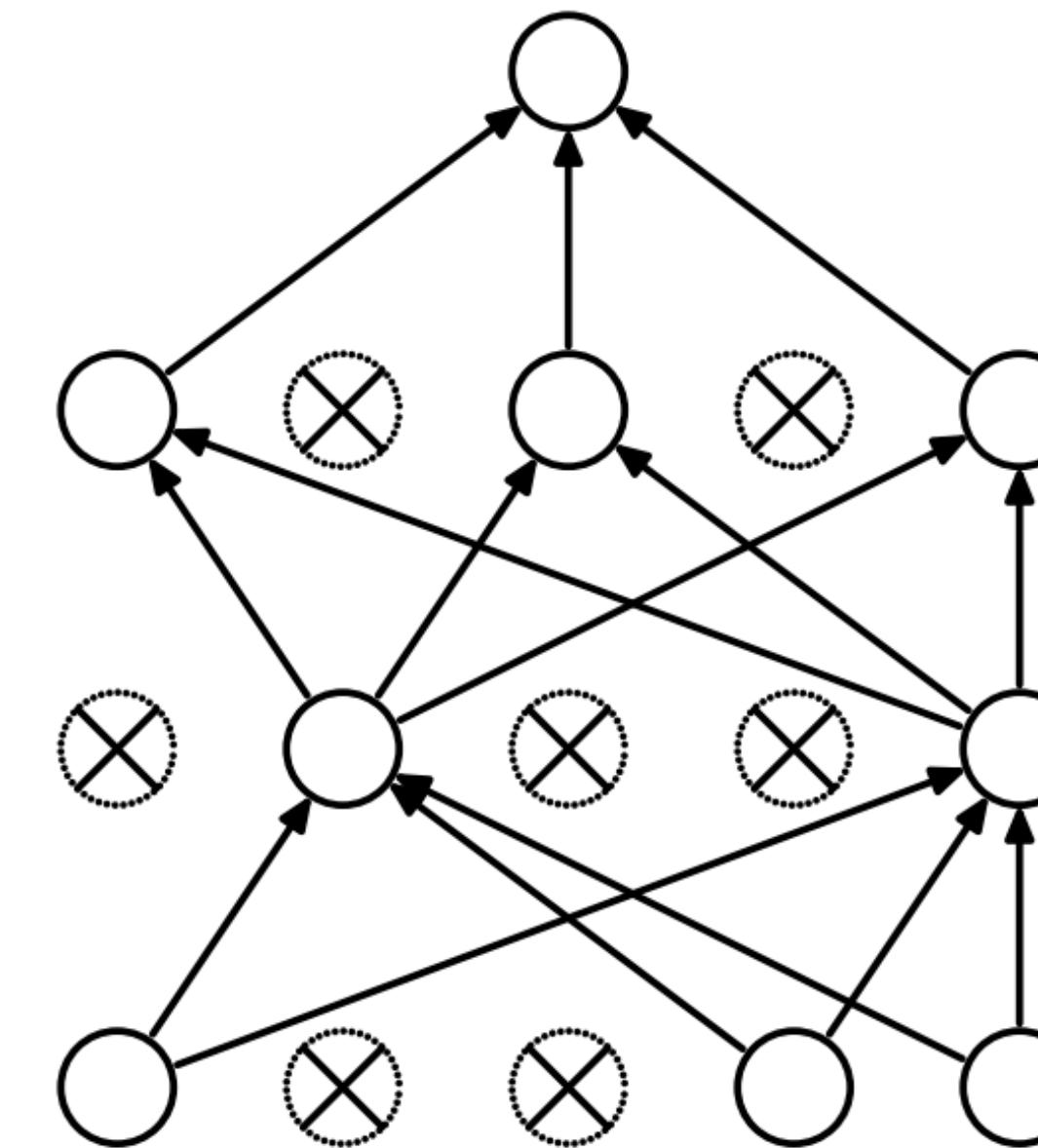
[Image credit](#)

Случайно выкидываем нейроны на forward pass с **вероятностью p**

Регуляризация: dropout



(a) Standard Neural Net

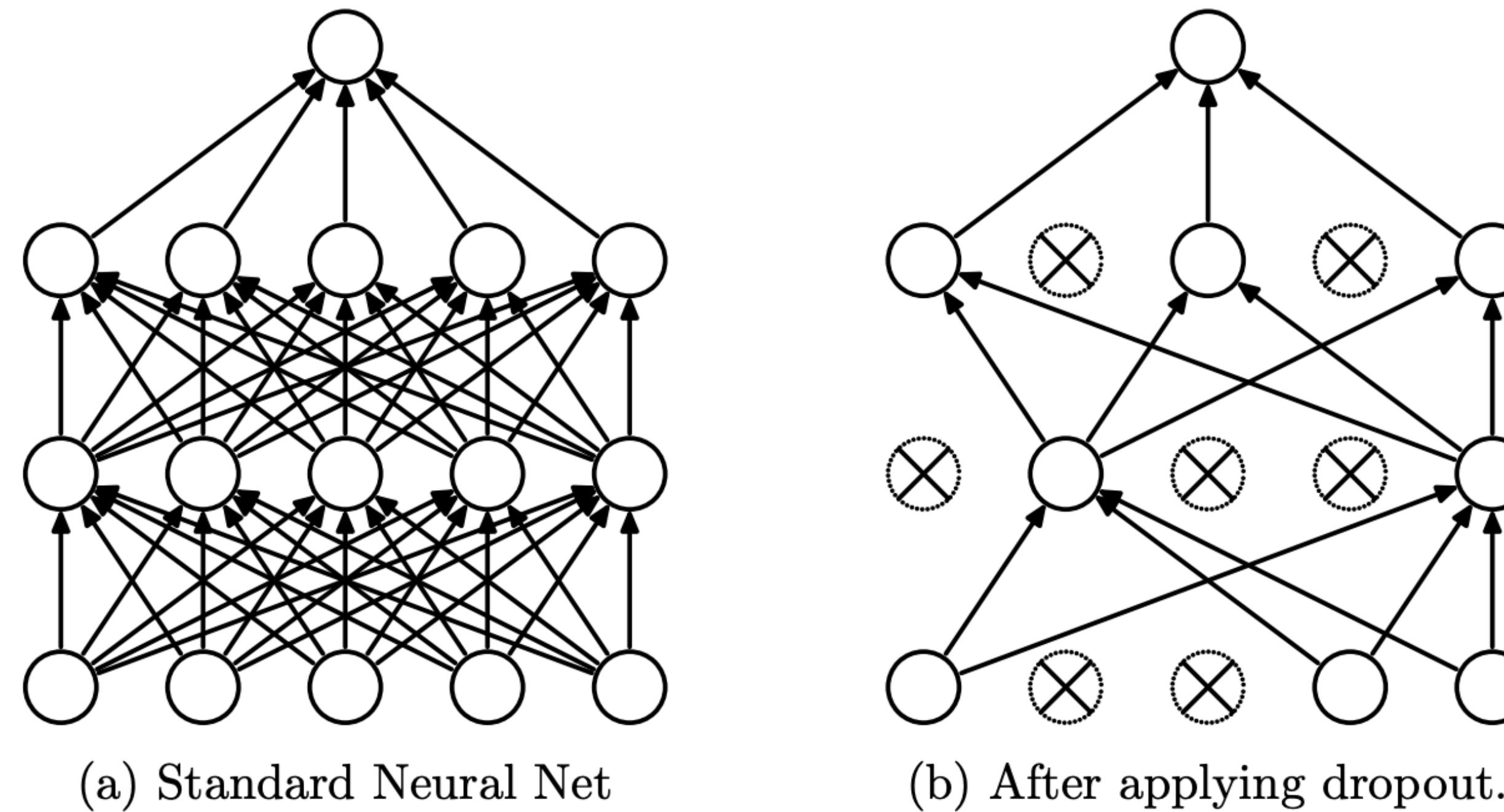


(b) After applying dropout.

[Image credit](#)

Случайно выкидываем нейроны на forward pass с **вероятностью** p
Во время обучения и теста работает по-разному!

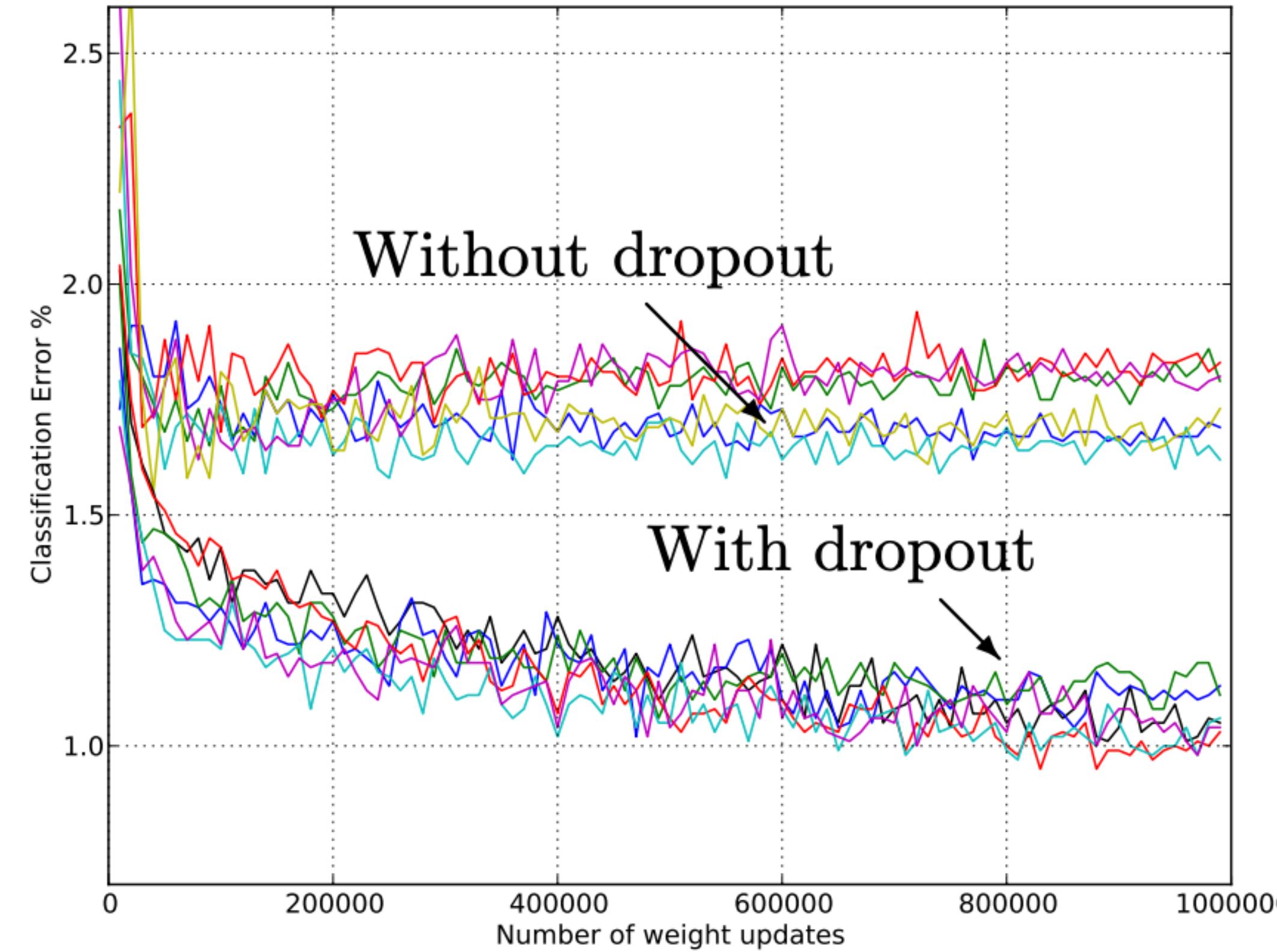
Регуляризация: dropout



Случайно выкидываем нейроны на forward pass с **вероятностью** p
Обучение: умножаем оставшиеся нейроны на $\frac{1}{1-p}$

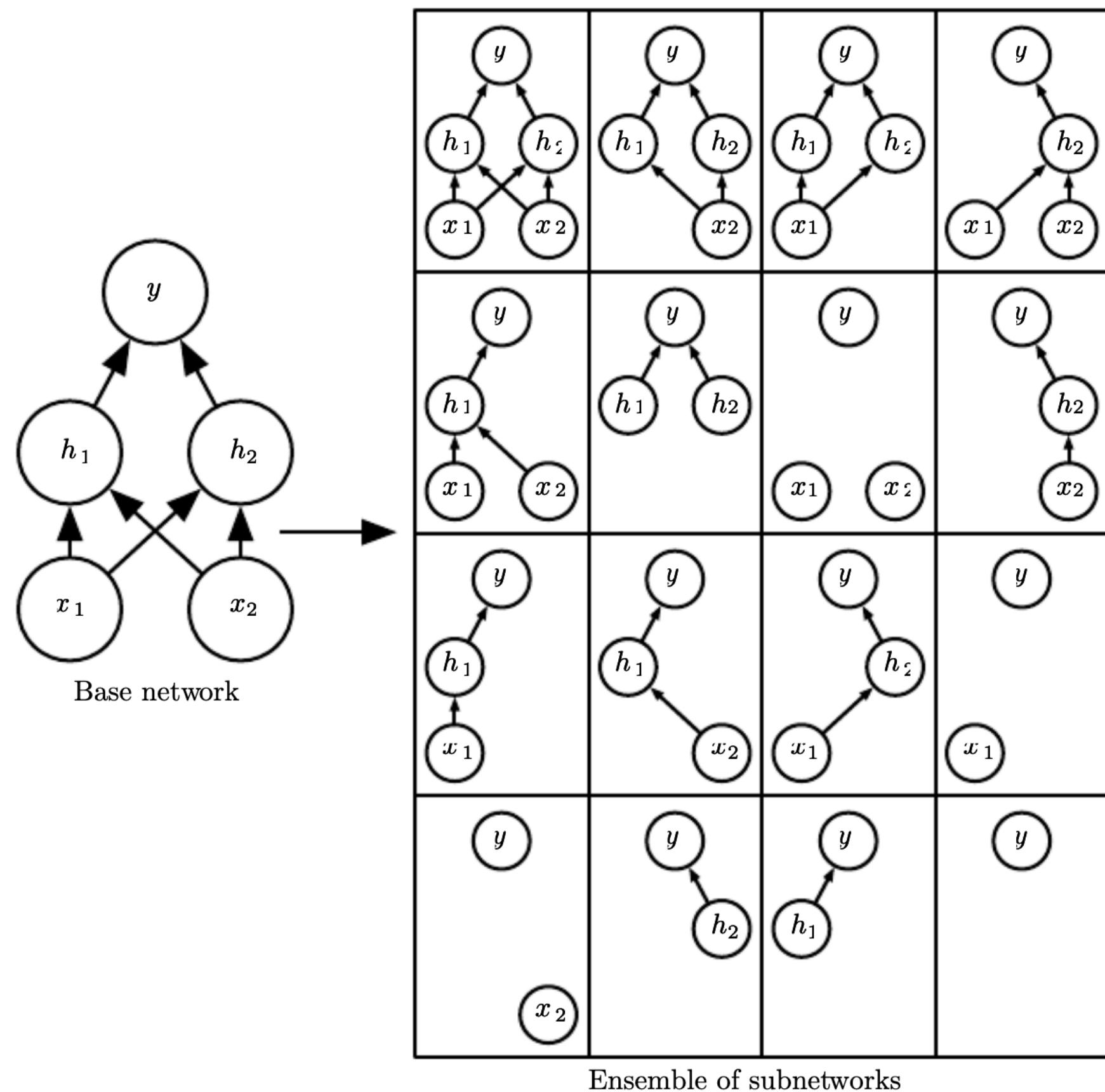
[Image credit](#)

Регуляризация: dropout



[Image credit](#)

Регуляризация: dropout



Обучение ансамбля нейросетей
(shared параметры)

Image credit

Регуляризация

- Из классического ML (L1/L2 регуляризация, label smoothing)
- Early stopping
- Dropout
- Дополнительные задачи для обучения
- Больше данных, меньше модель
- Аугментации

Batch Normalization

Рассмотрим один (не первый) слой нейросети:

во время обучения меняется распределение входа (**internal covariate shift**)

- меняется вход в саму нейросеть (разные объекты)
- меняются веса предыдущих входов

Batch Normalization

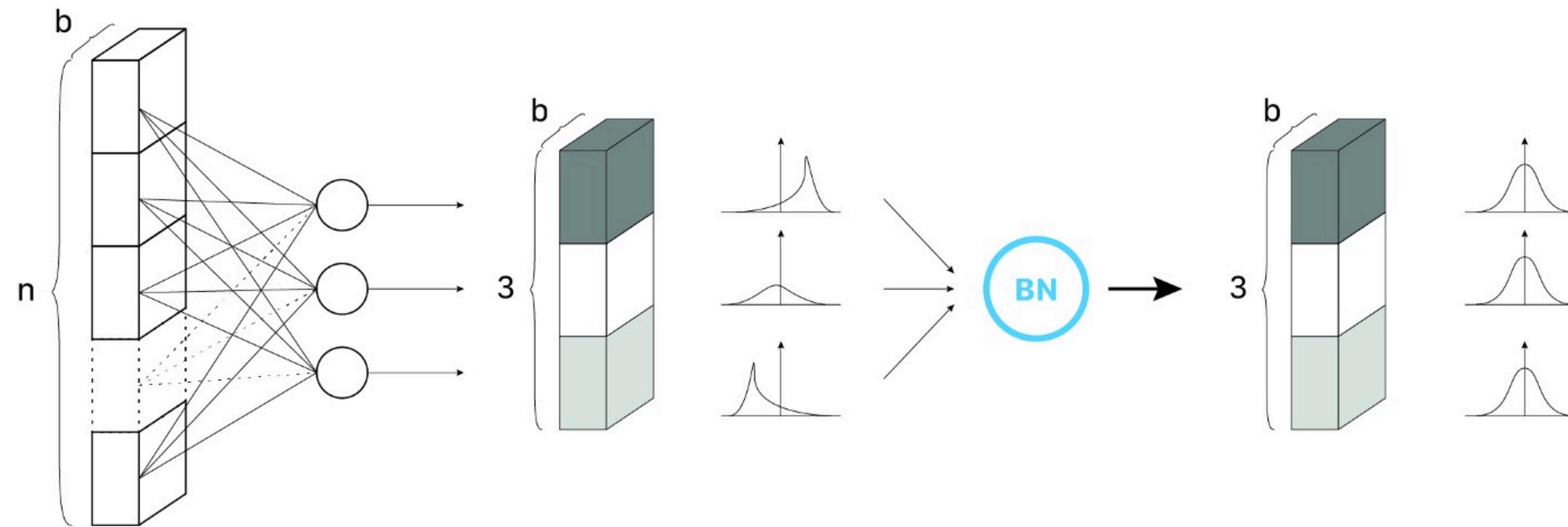
Рассмотрим один (не первый) слой нейросети:

во время обучения меняется распределение входа (**internal covariate shift**)

- меняется вход в саму нейросеть (разные объекты)
- меняются веса предыдущих входов

Batch normalization: будем нормировать среднее и дисперсию входов

Batch Normalization



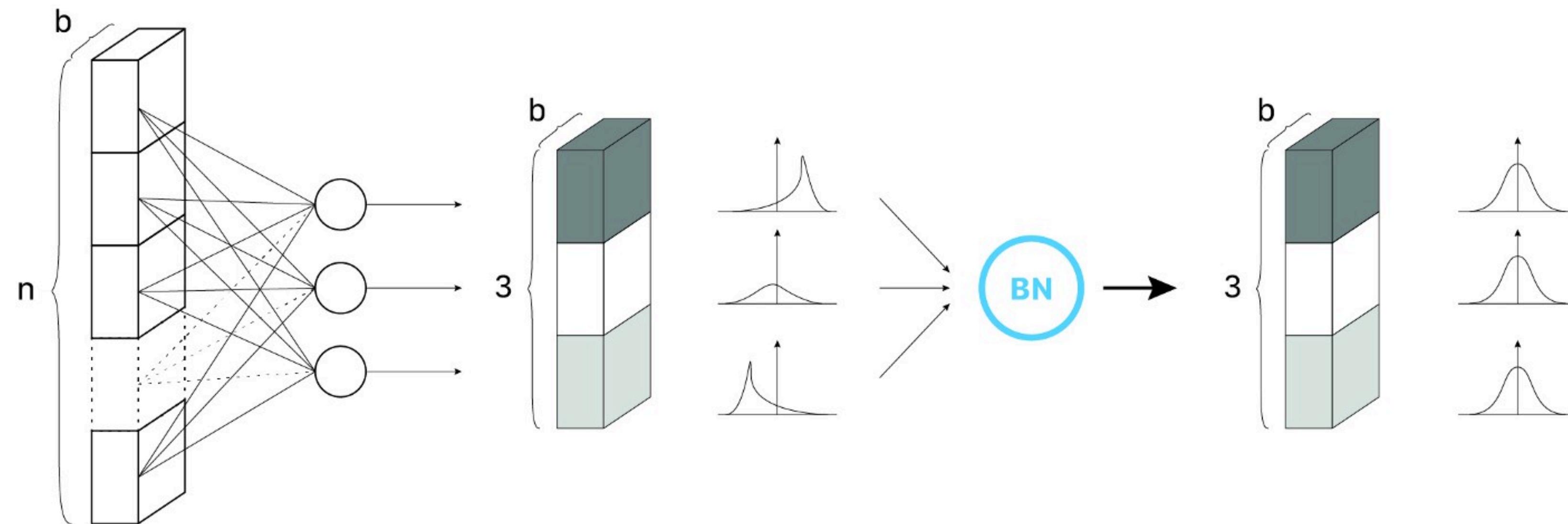
[Image credit](#)

Batch normalization: будем нормировать среднее и дисперсию входов

x_1, \dots, x_m input mini-batch

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad \sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

Batch Normalization



[Image credit](#)

Batch normalization: будем нормировать среднее и дисперсию входов

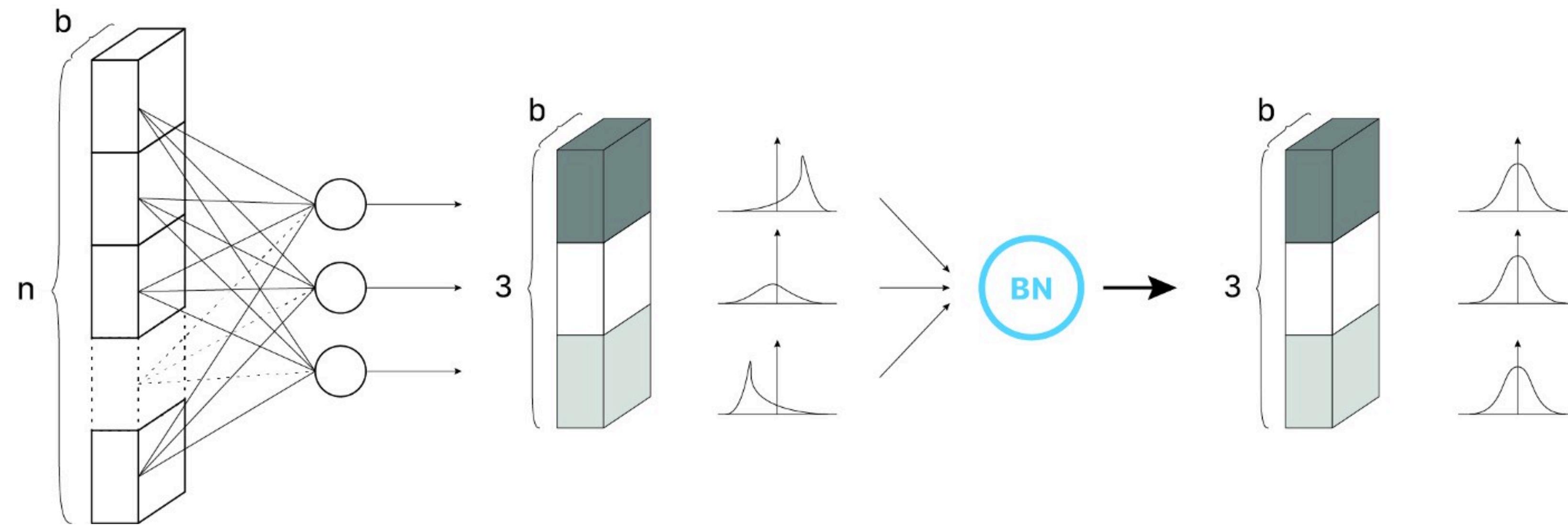
x_1, \dots, x_m input mini-batch

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad \sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$



$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Batch Normalization



[Image credit](#)

Batch normalization: будем нормировать среднее и дисперсию входов

x_1, \dots, x_m input mini-batch

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad \sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$



$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Identity transformation?

Batch Normalization

x_1, \dots, x_m input mini-batch

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta \quad \gamma, \beta \text{ - обучаемые параметры}$$

Batch Normalization

x_1, \dots, x_m input mini-batch

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$$

- зависит от батча,
как использовать на teste?

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta \quad \gamma, \beta \text{ - обучаемые параметры}$$

Batch Normalization

x_1, \dots, x_m input mini-batch

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad \hat{\mu} = \alpha \mu_B + (1 - \alpha) \hat{\mu}$$

- для обучения

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad \hat{\sigma}^2 = \alpha \sigma_B^2 + (1 - \alpha) \hat{\sigma}^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta$$

Batch Normalization

x_1, \dots, x_m input mini-batch

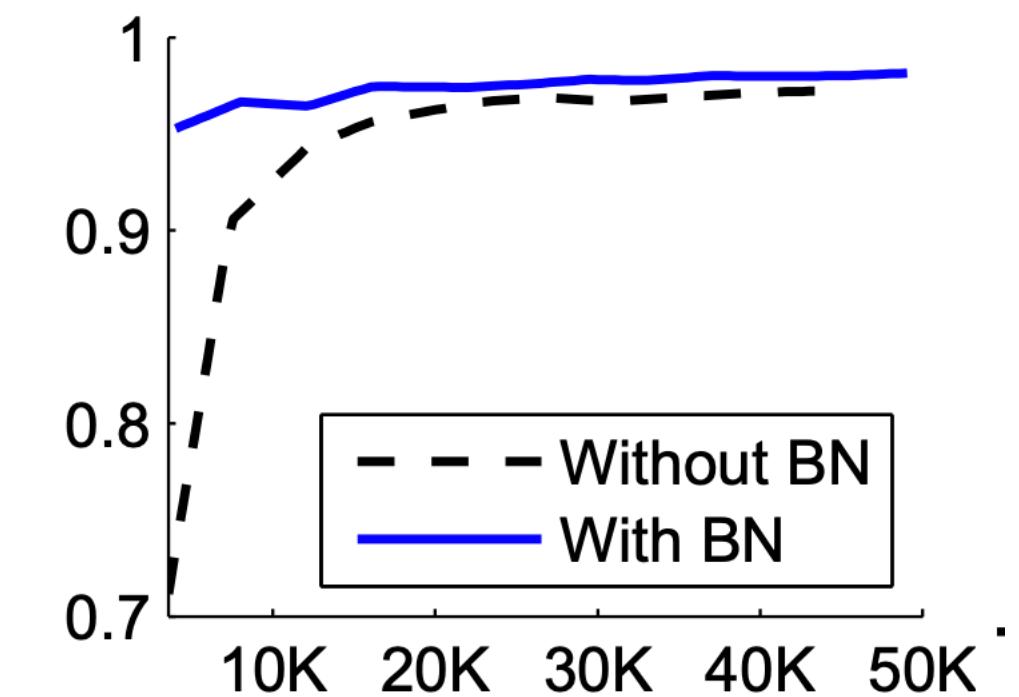
$$\hat{x}_i = \frac{x_i - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \epsilon}}$$

- для теста

$$y_i = \gamma \hat{x}_i + \beta$$

Batch Normalization

- Быстрее сходимость
- Позволяет использовать значения γ больше
- Нейросети менее чувствительны к инициализации



Обычно используется между Linear/Conv и нелинейностью

Batch Normalization

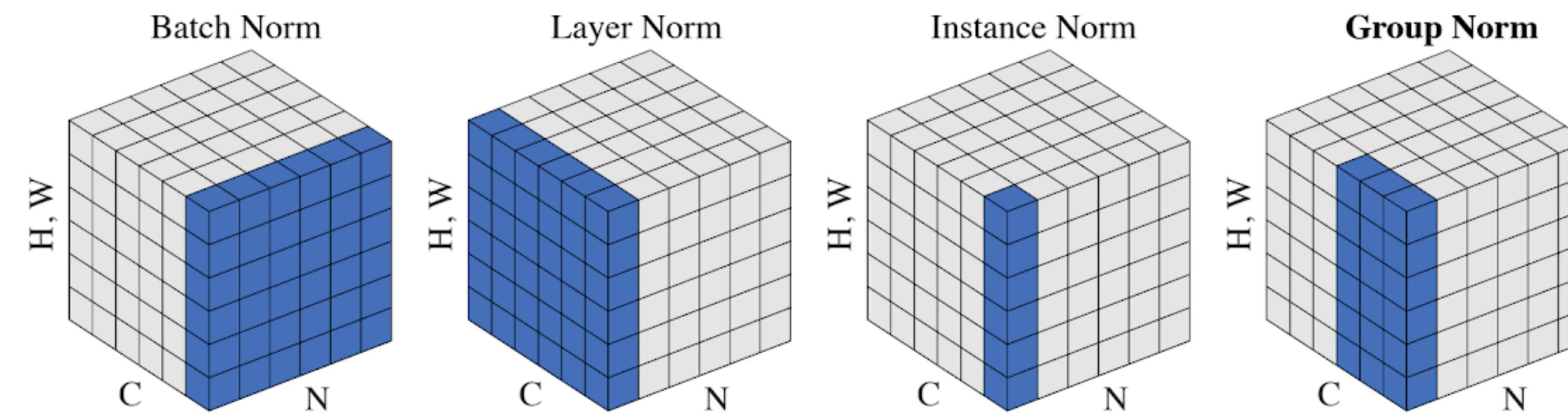


Figure 2. **Normalization methods.** Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels.