Настройка нейронной сети

Марина Горлова

С чего начать?

1 Определиться с тем какую задачу решаем

2 Подготовить данные

3 Построить бейзлайн

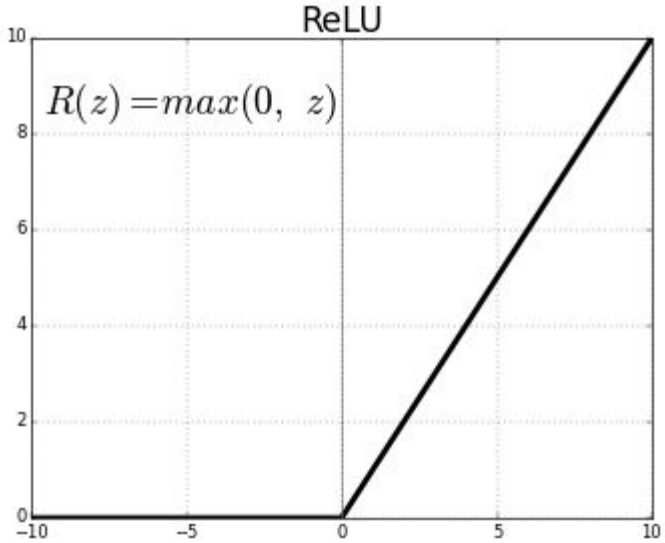
4 Перейти к более сложным моделям и настройке параметров

Задача регрессии

Функция активации ReLU

mse, mae

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

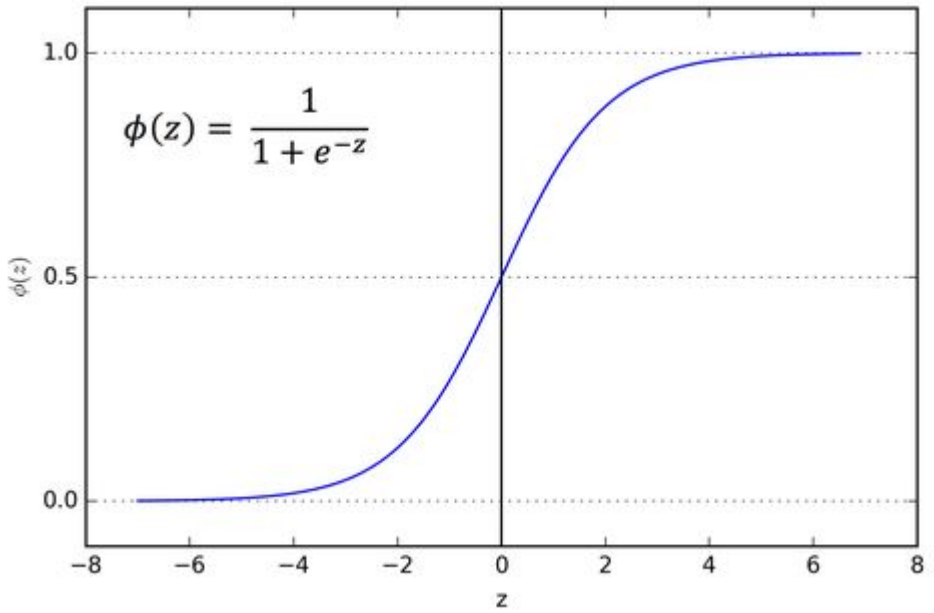


Задача бинарной классификации

сигмоидная функция активации

binary_crossentropy

$$-(y\log(p)+(1-y)\log(1-p))$$



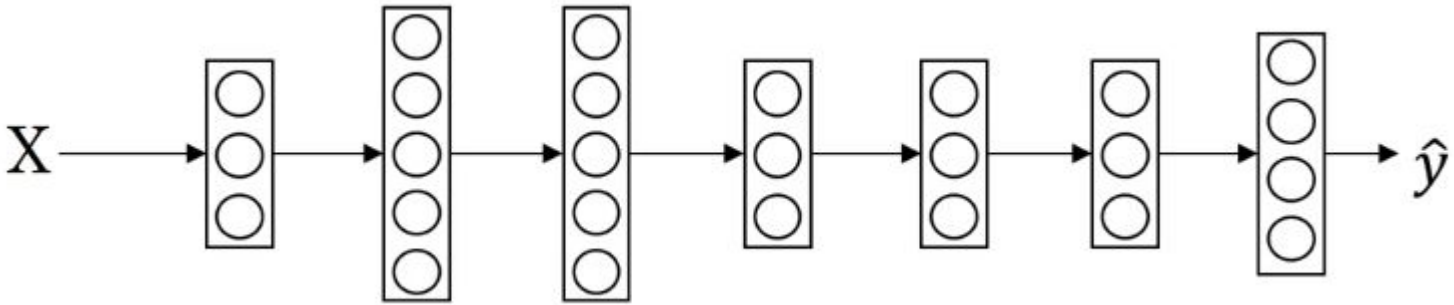
Задача многоклассовой классификации

активация softmax

category_crossentropy

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$



Подготовка данных

- тренировочный сет
- валидационный сет
- тестовый

Пропорция, в которой разбиваем данные, зависит от их количества,

задачи и способа валидации

Нормализация данных

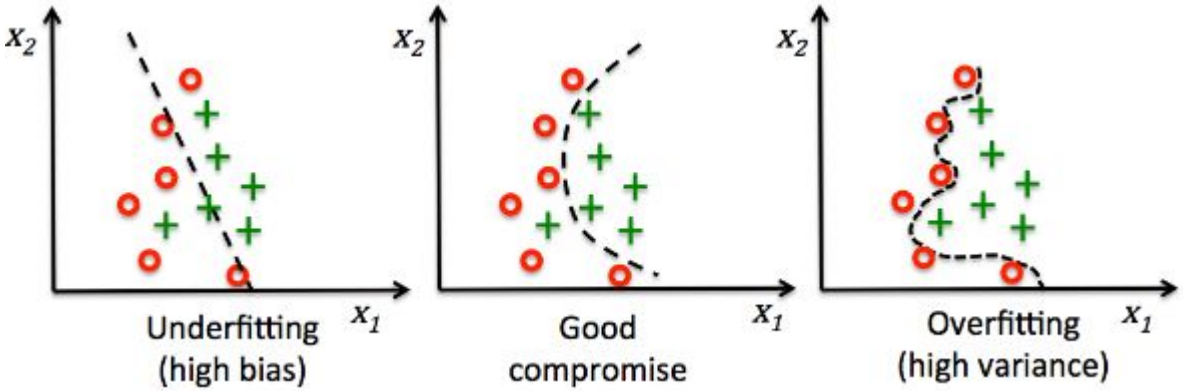
Bias / Variance

bias

- большая сеть
- больше epoch

variance

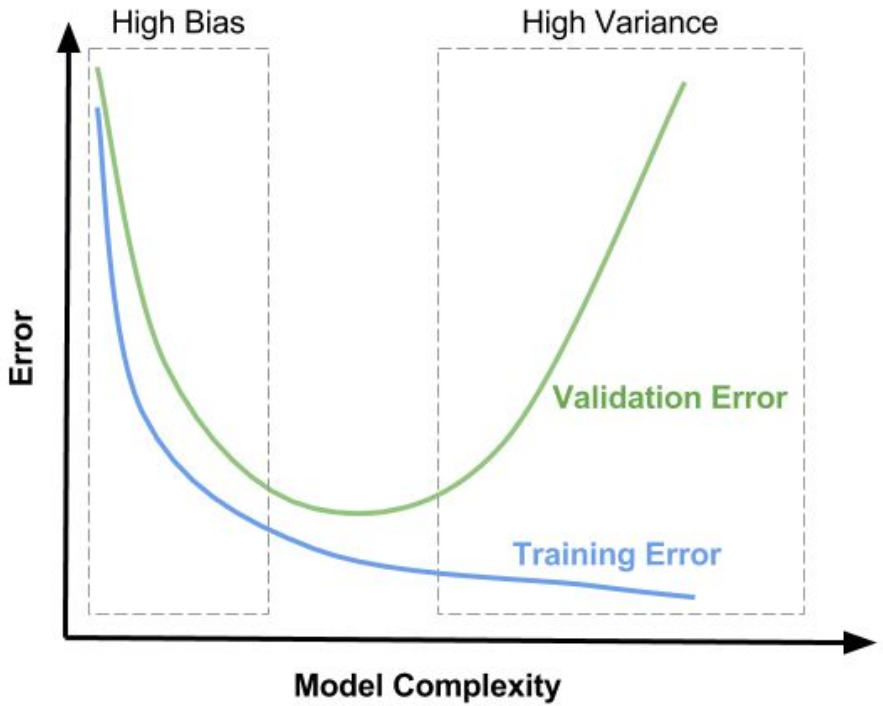
- больше данных
- регуляризация



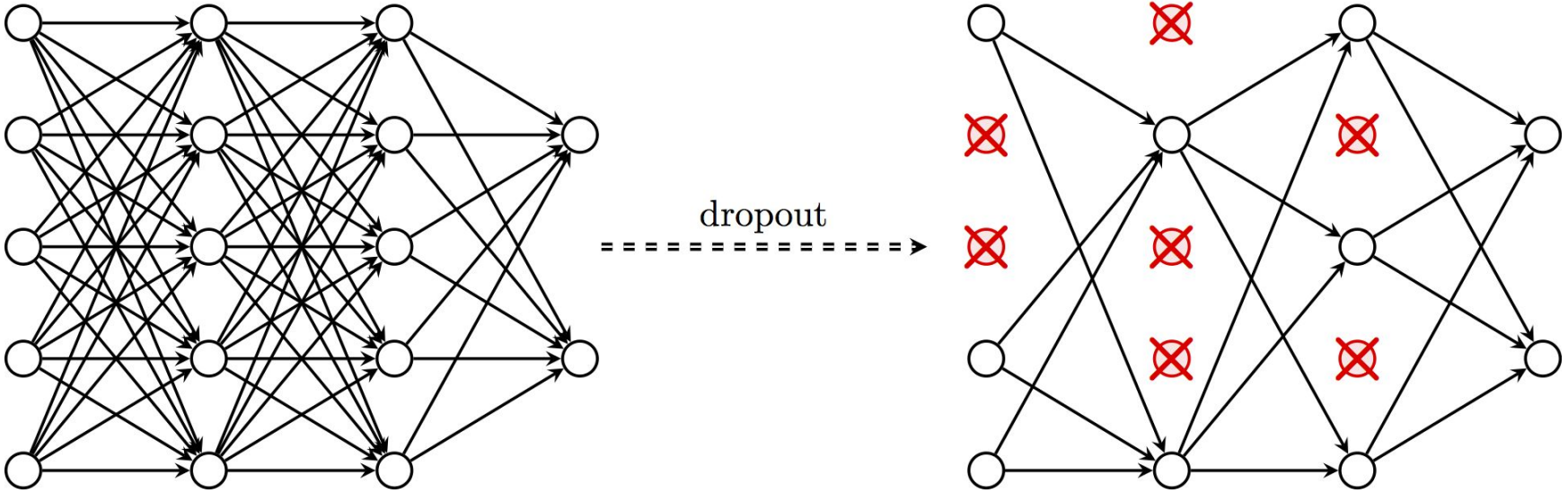
Регуляризация

L1, L2 регуляризация

Dropout



Dropout



Dropout

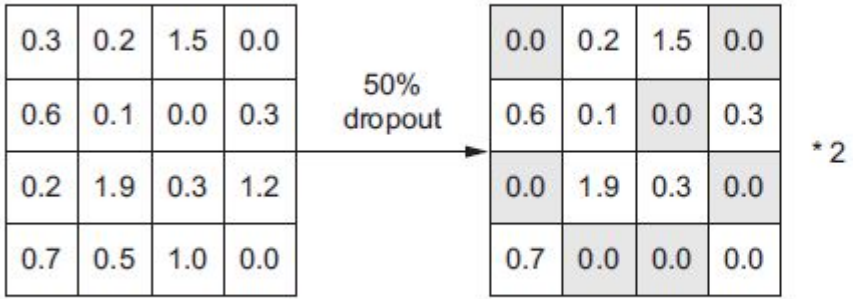


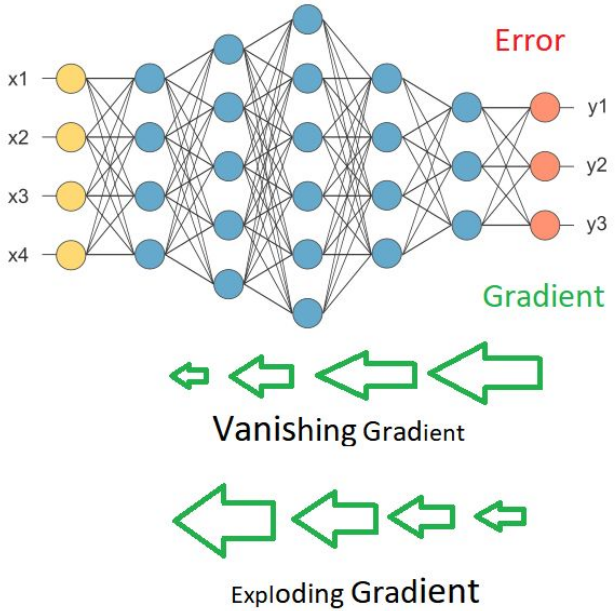
Figure 4.8 Dropout applied to an activation matrix at training time, with rescaling happening during training. At test time, the activation matrix is unchanged.

Vanishing / Exploding gradient

Vanishing gradient - веса на близких

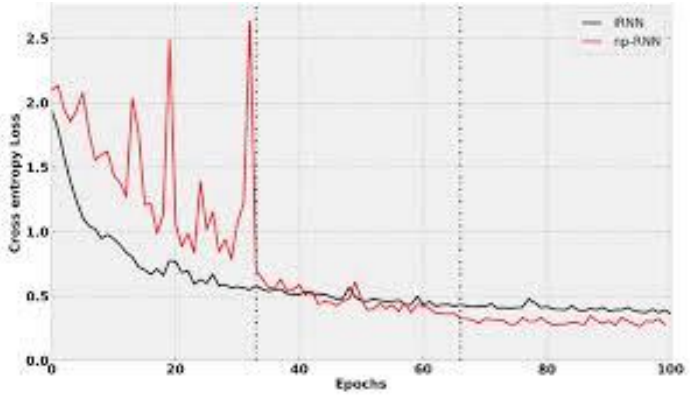
к X слоях становятся близкими к 0

Exploding gradient - веса резко растут



Exploding gradient

Определить можно по training loss

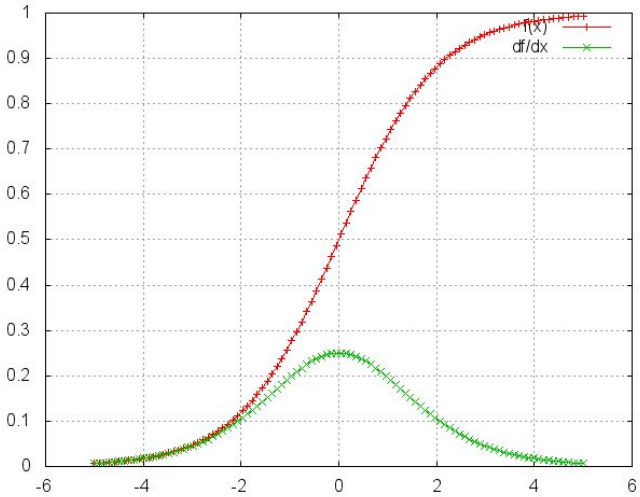


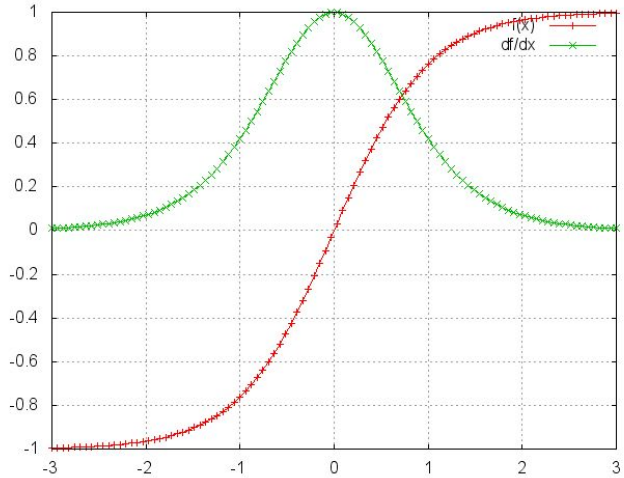
решение проблемы регуляризация и clipping, уменьшить learning rate

Vanishing gradient

Нет надежного способа определить эту проблему

Sigmoid и tanh activation могут усугубить проблему



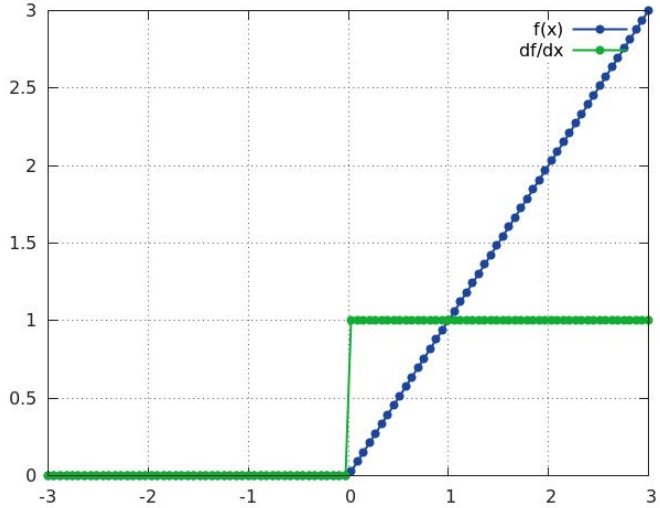


Vanishing gradient

Решение: использовать ReLU и его модификации на скрытых слоях

Поменять инициализацию весов

- N(0,1)*0.001
- N(0,1) *0.001 *1/sqrt(n)
- N(0,1) *0.001 *sqrt(2/n)
- N(0,1) *0.001 *sqrt(2/(n_{input}+ n_{output}))



Оптимизация mini-batch gradient descent

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)}).$$

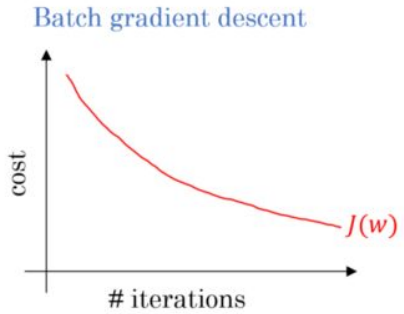
pros

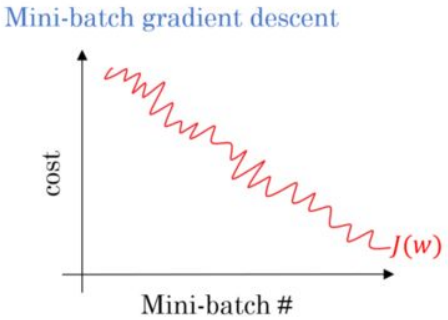
- меньше застревает в локальных минимумах
- эффективнее обучается
- требует меньше памяти

cons

- не оптимальное решение

  на каждом шаге



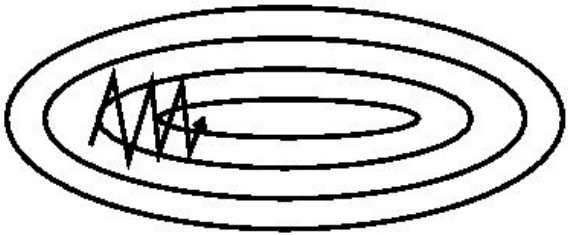


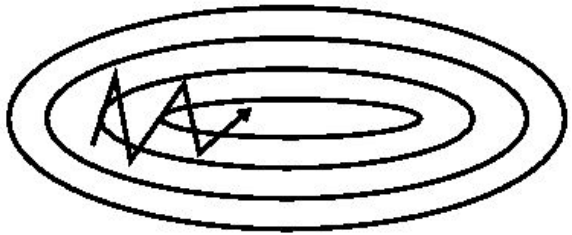
Оптимизация

- Застревание в локальных минимумах или седловых точках
- Плато чередуются с регионами сильной нелинейности
- Некоторые параметры обновляются значительно реже других, особенно когда в данных встречаются информативные, но редкие признаки
- Слишком маленькая скорость обучения заставляет алгоритм сходиться очень долго и застревать в локальных минимумах, слишком большая — «пролетать» узкие глобальные минимумы или вовсе расходиться

Оптимизация: momentum

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$
$$\theta = \theta - v_t$$





Сохраняет информацию о предыдущем направлении

Оптимизация: AdaGrad

$$g_t \equiv \nabla_\theta J(\theta_t)$$

$$G_t = G_t + g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} g_t$$

Уменьшает степень обновления параметров, которые часто обновляются

Комбинация этих двух идей - Adam

Настройка гиперпараметров

learning rate и другие параметры оптимизации

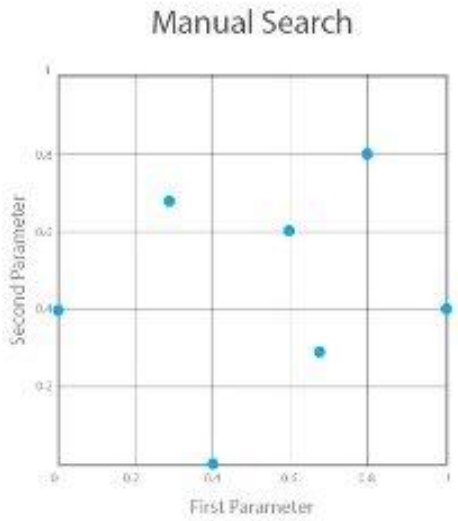
mini-batch

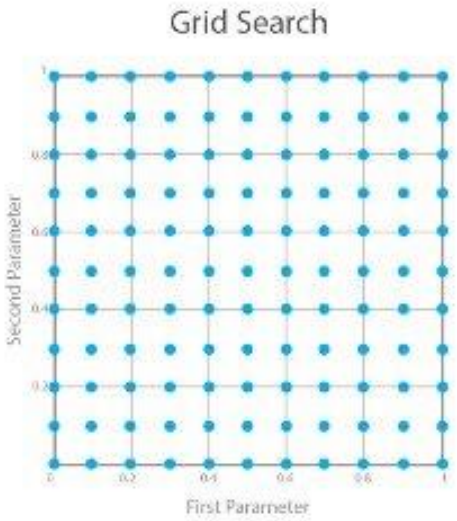
количество элементов на скрытых слоях

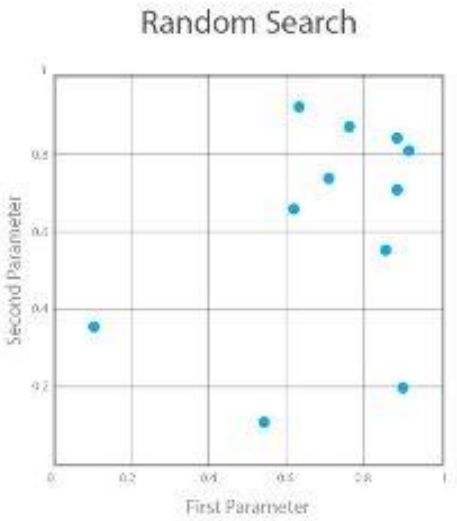
количество слоев

learning rate decay

Настройка гиперпараметров







Batch normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

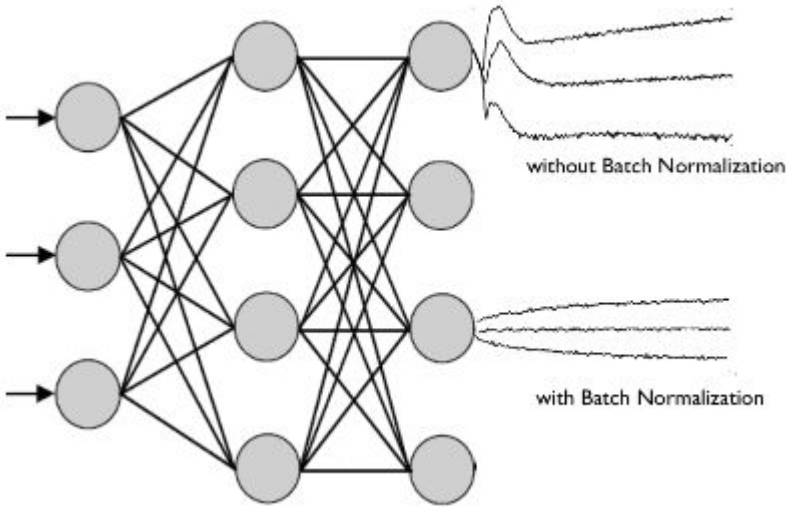
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.



На тесте используются экспоненциально взвешенные значения по mini-batch