Глубокие нейронные сети

Техники, улучшающие обучение нейронных сетей

Лекция 8

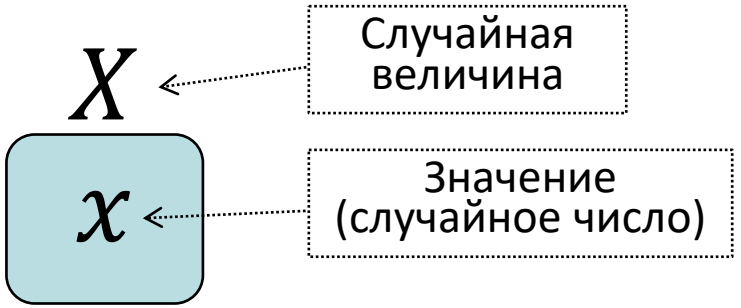
Техники, улучшающие обучение нейронных сетей

- Инициализация весов и смещений на основе нормального распределения
- Уменьшение скорости обучения η
- Градиентный спуск на основе импульса
- Алгоритмы с адаптивной скоростью обучения
- Альтернативные модели нейрона

Инициализация весов и смещений на основе нормального распределения (Weight initialization based on Gaussian distribution)

Случайная величина X

- Случайная величина X принимает случайное значение x с некоторой вероятностью
- Сумма вероятностей для всех возможных значений равна 1



Функция распределения $F(x)$ случайной величины X

$$F(x) = P(X < x)$$

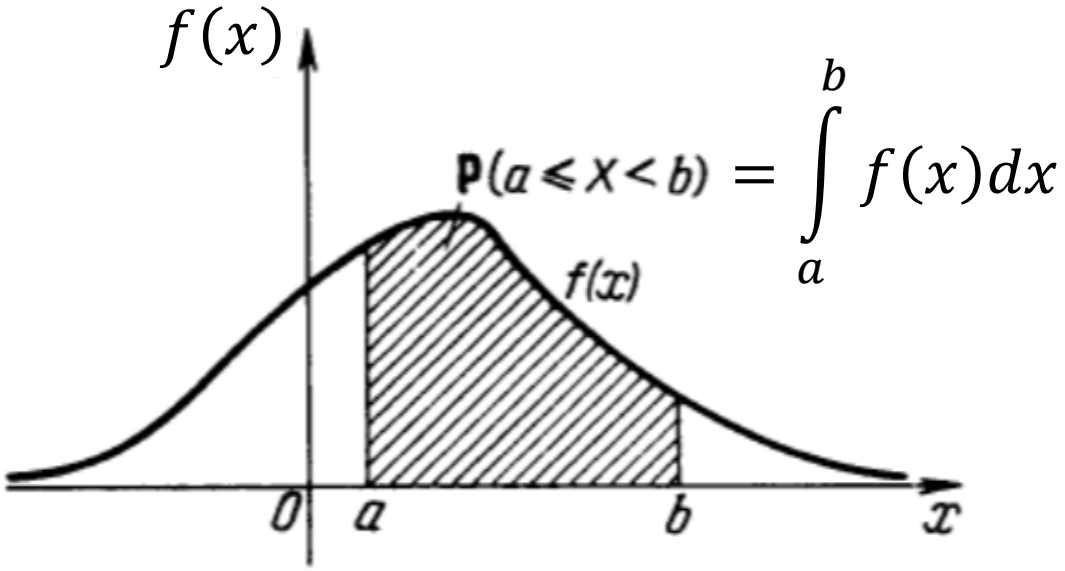
$$F: \mathbb{R} \to \mathbb{R}_{[0;1]}$$

$F(x)$ – вероятность того, что случайная величина X примет значение меньше x

Плотность вероятности $f(x)$

Функция распределения F случайной величины X имеет вид:

$$F(x) = \int_{-\infty}^{x} f(x)dx$$

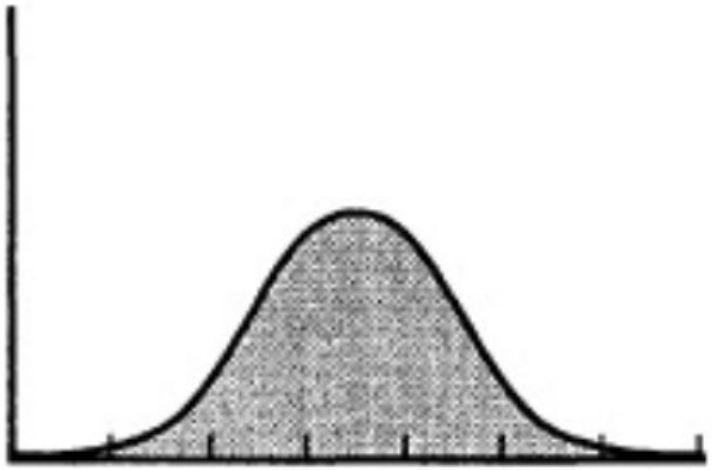


Свойство:

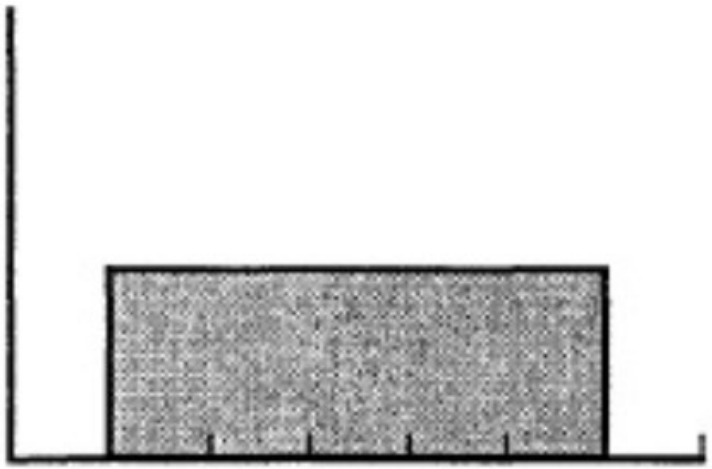
$$\lim_{x\to\infty} F(x) = 1 \Rightarrow \int_{-\infty}^{+\infty} f(x)dx = 1$$

Примеры распределений случайных величин

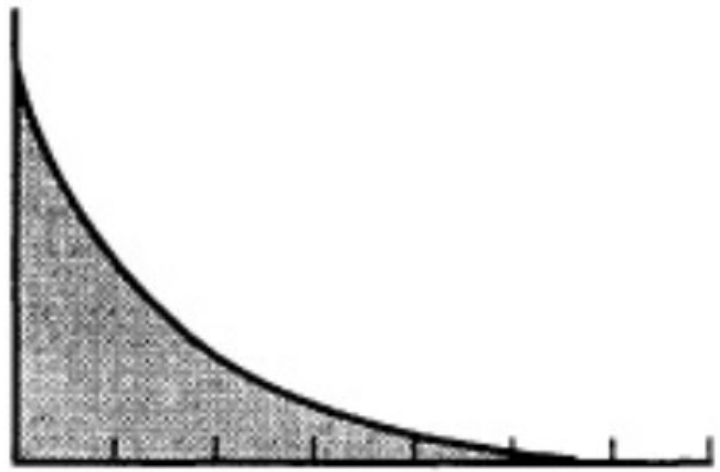
?



?



?



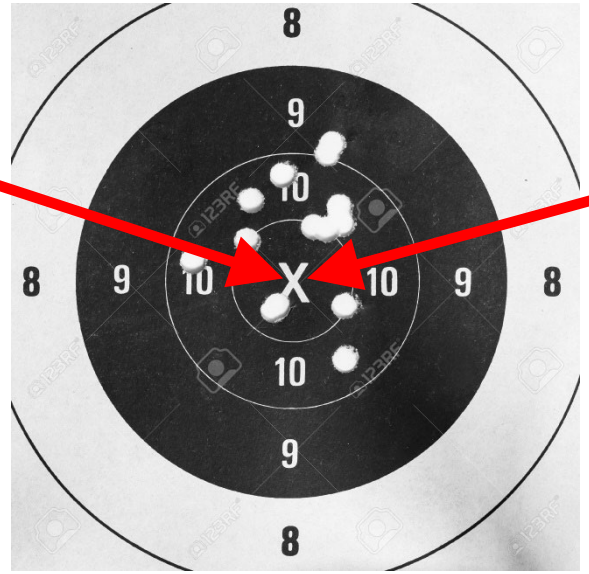
Математическое ожидание

Для непрерывной случайной величины

$$M[X] = \int_{-\infty}^{+\infty} x\, f(x) dx$$

$$M[X] = \sum_{i=1}^{n} x_i f(x_i)$$

Для дискретной случайной величины



$$M[X] \approx \frac{1}{n} \sum_{i=1}^{n} x_i$$

Для какого распределения эта формула является точной?

Дисперсия
(разброс относительно математического ожидания)

$$D[X] = \int_{-\infty}^{+\infty} (x - M[X])^2 f(x) dx$$

$$D[X] = \sum_{i=1}^{n} (x_i - M[X])^2$$

Для дискретного распределения

Маленькая дисперсия

Большая дисперсия

Нормальное распределение

Нормальное распределение имеет *плотность распределения*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}$$

Математическое ожидание:

$$M[X] = \int_{-\infty}^{+\infty} x \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} dx = a$$

Дисперсия:

$$D[X] = \int_{-\infty}^{+\infty} (x-a)^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} dx = \sigma^2$$

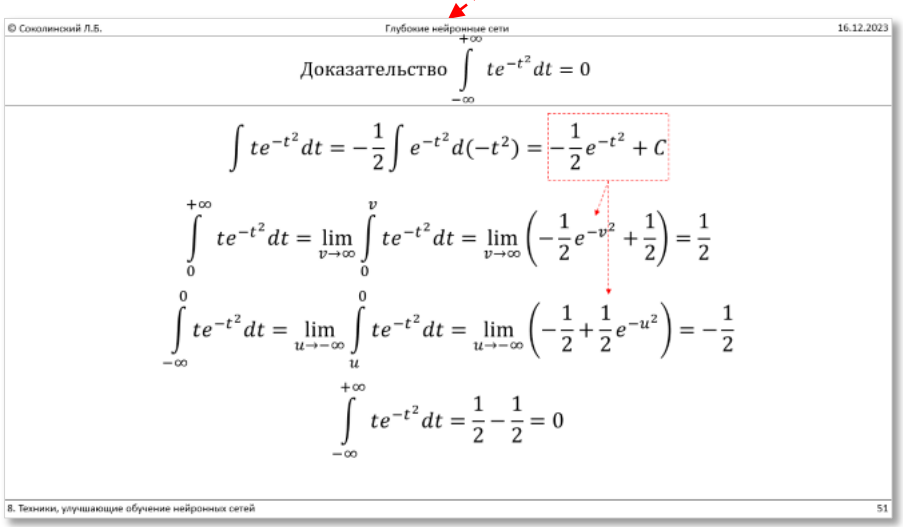
Нормальное распределение полностью определяется своими математическим ожиданием и дисперсией!

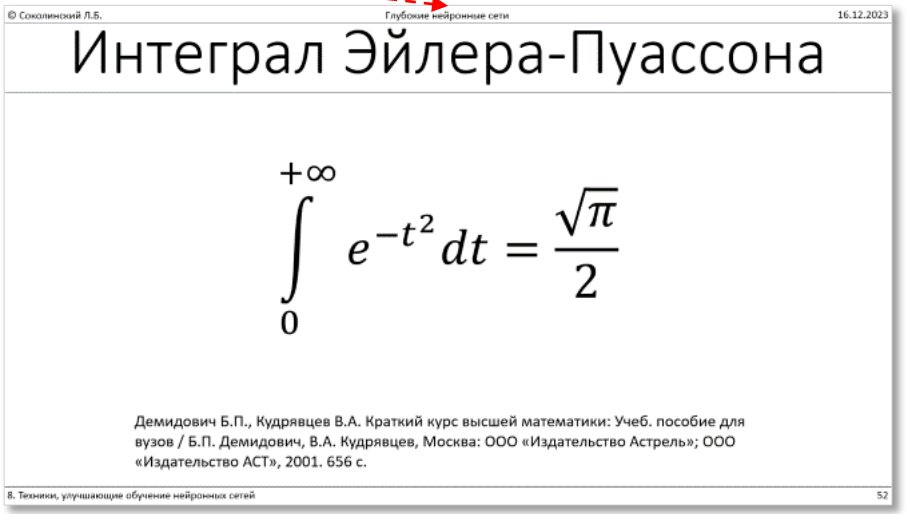
Доказательство $\int_{-\infty}^{+\infty} x \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} dx = a$

Применяя замену переменной $x = \sigma\sqrt{2}t + a$ имеем:

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} (\sigma\sqrt{2}t + a) e^{-t^2} dt = \frac{\sigma\sqrt{2}}{\sqrt{\pi}} \int_{-\infty}^{+\infty} te^{-t^2} dt + a\frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-t^2} dt$$

$$= \frac{\sigma\sqrt{2}}{\sqrt{\pi}} \int_{-\infty}^{+\infty} te^{-t^2} dt + a\frac{2}{\sqrt{\pi}} \int_{0}^{+\infty} e^{-t^2} dt = a$$





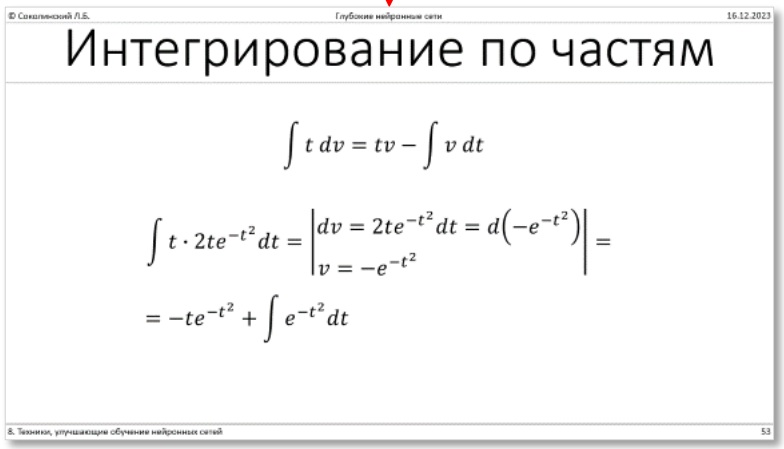
Доказательство $\int_{-\infty}^{+\infty} (x-a)^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} dx = \sigma^2$

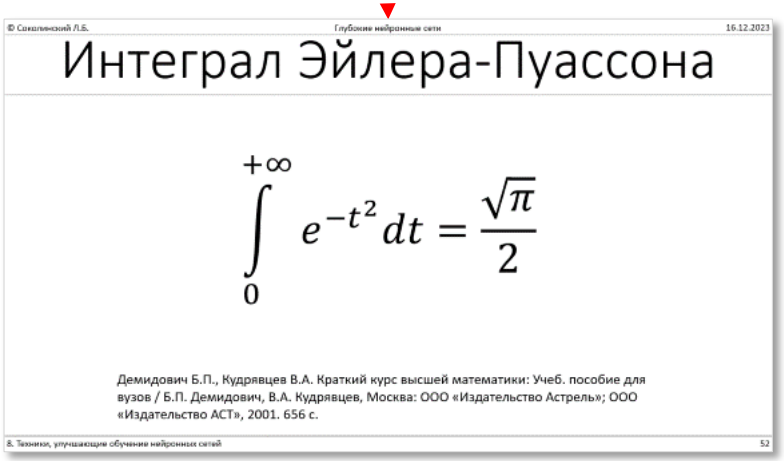
Применяя замену переменной $t = \frac{x-a}{\sigma\sqrt{2}}$ имеем:

$$\int_{-\infty}^{+\infty} (x-a)^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} dx = \frac{2\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{+\infty} t^2 e^{-t^2} dt$$

Интегрируя по частям, получим:

$$\frac{\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{+\infty} t \cdot \underbrace{2te^{-t^2}dt}_{dv} = \frac{\sigma^2}{\sqrt{\pi}} \left(-te^{-t^2} \Big|_{-\infty}^{+\infty} + \int_{-\infty}^{+\infty} e^{-t^2} dt \right) = \frac{\sigma^2}{\sqrt{\pi}} 2 \int_{0}^{+\infty} e^{-t^2} dt = \sigma^2$$





Стандартное нормальное распределение

Математическое ожидание:

$$a = M[X] = 0$$

Среднее квадратическое отклонение:

$$\sigma = \sigma[X] = \sqrt{D[X]} = 1$$

Плотность вероятности стандартного нормального распределения:

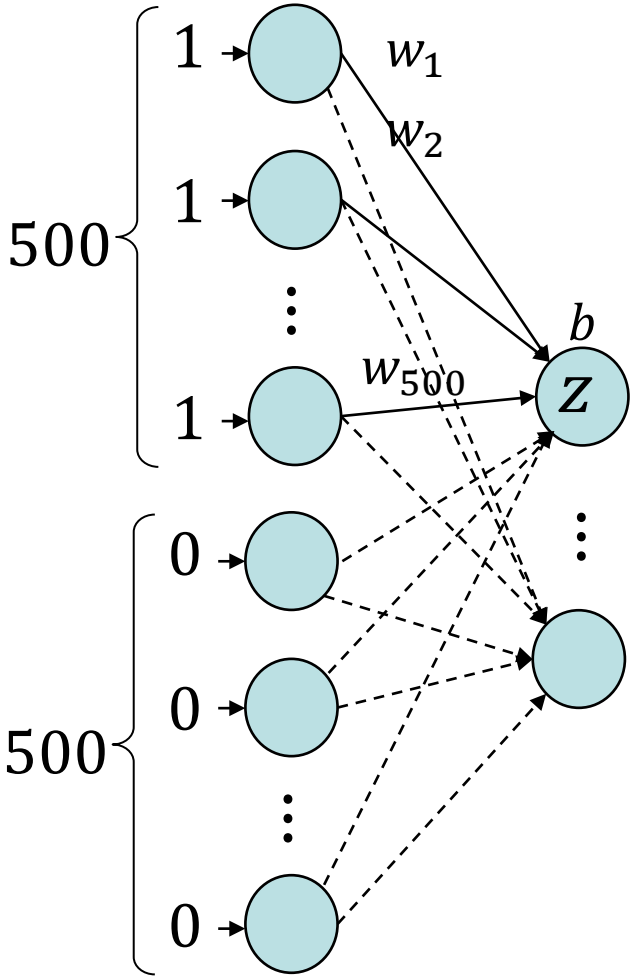
$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Случайная генерация с использованием стандартного нормального распределения

Плотность распределения:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Проблема со стандартным нормальным распределением



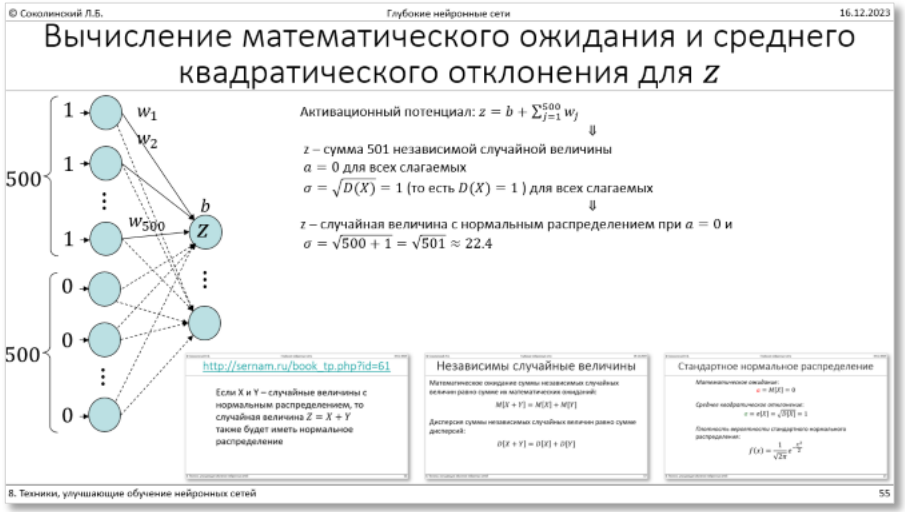
Активационный потенциал: $z = b + \sum_{j=1}^{500} w_j$

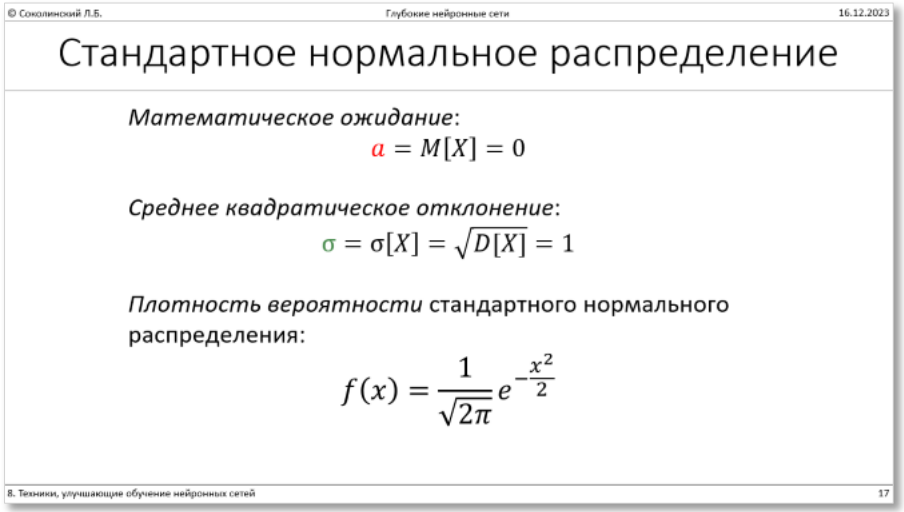
⇓

z – сумма 501 независимых случайных величин со стандартным нормальным распределением

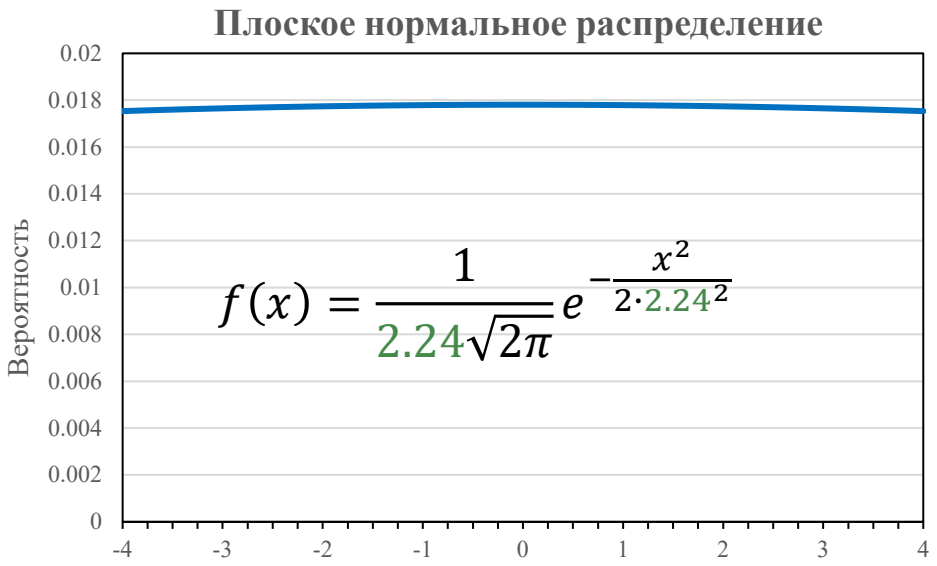
⇓

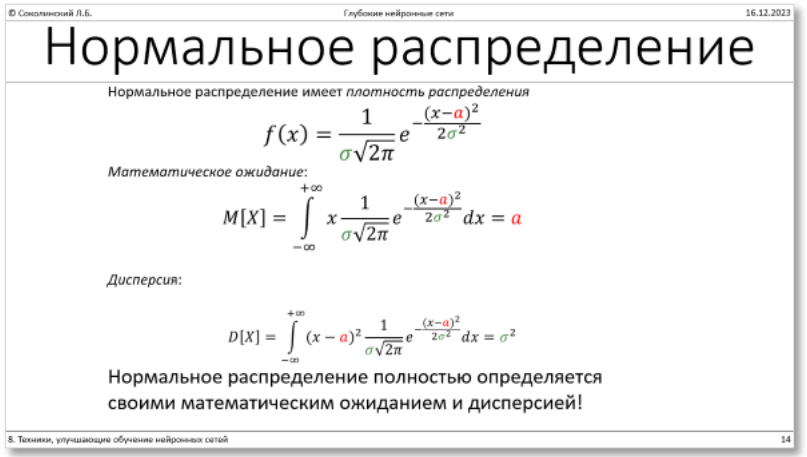
z – случайная величина с нормальным распределением при $a = 0$ и $\sigma = \sqrt{501} \approx 22.4$





Нормальное распределение для z при $a = 0$ и $\sigma = 22.4$





Высока вероятность, что

$$z \gg 1 \text{ или } z \ll -1$$

$$\Downarrow$$

Высока вероятность, что выходной сигнал скрытого нейрона будет очень близок к 1 или 0

$$\Downarrow$$

Высока вероятность, что скрытый нейрон будет плохо обучаться

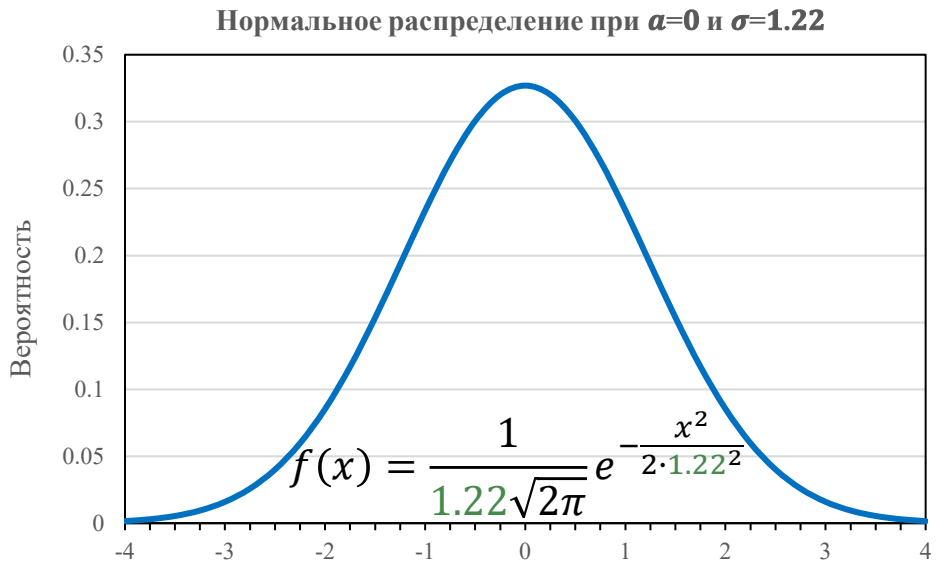
Решение проблемы

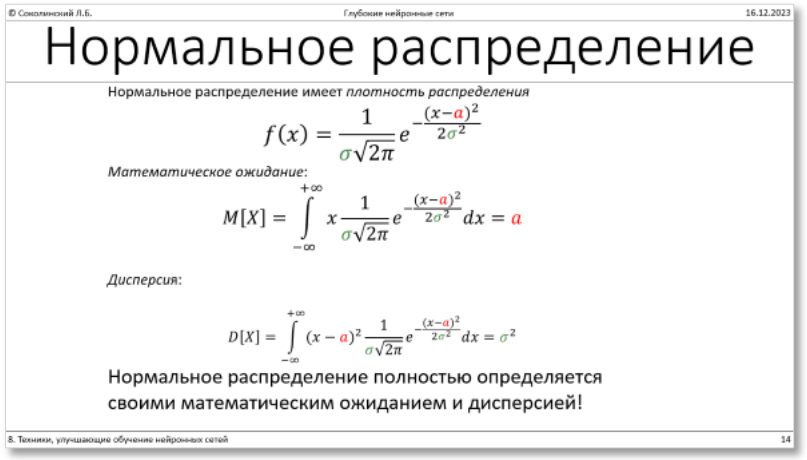
- Инициализировать веса w_j с помощью случайной величины с нормальным распределением при $a = 0$ и $\sigma = \sqrt{D(X)} = 1/\sqrt{1000}$
- Инициализировать b с помощью случайной величины с нормальным распределением при $a = 0$ и $\sigma = 1$

$$\Downarrow$$

z – случайная величина с нормальным распределением при $a = 0$ и $\sigma = \sqrt{\frac{500}{1000} + 1} = \sqrt{3/2} \approx 1.22$

Нормальное распределение для z при $a = 0$ и $\sigma = 1.22$





Низка вероятность, что $z \gg 1$ или $z \ll -1$

⇓

Низка вероятность, что выходной сигнал скрытого нейрона будет очень близок к 1 или 0

⇓

Низка вероятность, что скрытый нейрон будет плохо обучаться

Общий случай: скрытый нейрон имеет n входных весов

- Инициализировать веса w_j с помощью случайной величины с нормальным распределением при $a = 0$ и $\sigma = 1/\sqrt{n}$
- Инициализировать b с помощью случайной величины с нормальным распределением при $a = 0$ и $\sigma = 1$

Уменьшение скорости обучения η

- Скорость обучения η – самый важный параметр в стохастическом градиентном спуске
- На практике скорость обучения нужно уменьшать на каждой итерации:

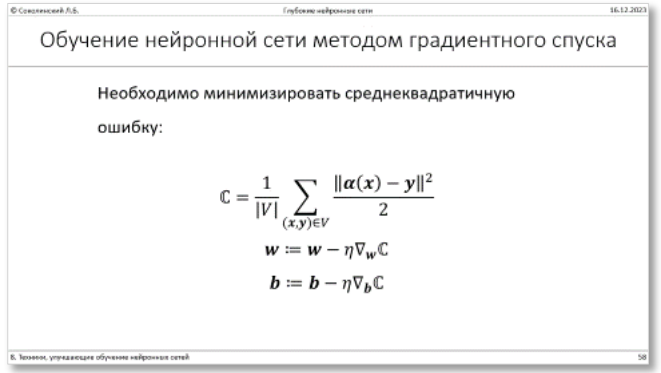
$$\eta_1 > \eta_2 > \cdots > \eta_i > \cdots$$

- Метод:
  - Уменьшаем скорость обучения линейно до итерации *K*:

$$\eta_i = \left(1 - \frac{i}{K}\right)\eta_0 + \frac{i}{k}\eta_K$$

  - Начиная с итерации K скорость обучения остается постоянной:

$$\eta_{i>K} = \eta_K$$



Выбор K, η_0 и η_K

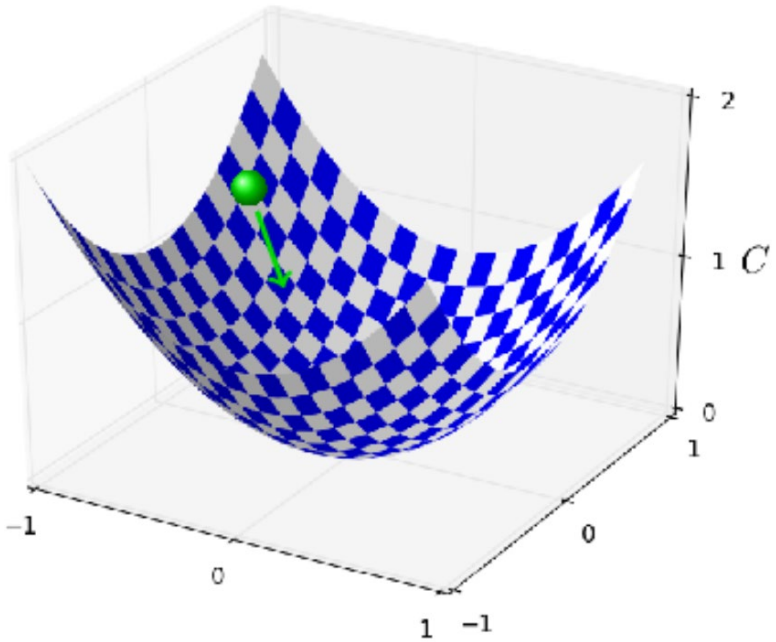
$$100 < K < 1000$$

η_0 – максимальная скорость на первых 10 итерациях, при которой процесс сходится

$$\eta_K = \frac{\eta_0}{100}$$

Алгоритмы-оптимизаторы с адаптивной скоростью обучения

Импульсный метод (Momentum)



$\boldsymbol{v}$ – скорость движения шарика по поверхности (начальная скорость равна нулю)

μ – момент движения (величина, обратно пропорциональная трению шарика о поверхность):

$$0 \leq \mu \leq 1$$

$$\boldsymbol{v_w} \coloneqq \mu \boldsymbol{v_w} - \eta \nabla_{\boldsymbol{w}} \mathbb{C}$$

$$\boldsymbol{v_b} \coloneqq \mu \boldsymbol{v_b} - \eta \nabla_{\boldsymbol{b}} \mathbb{C}$$

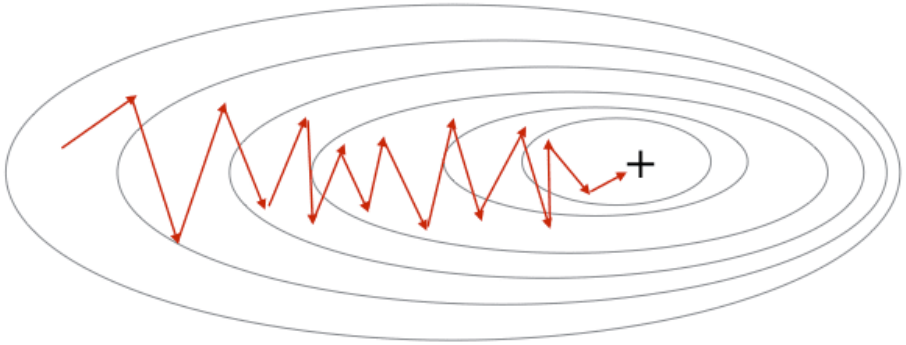
$$\boldsymbol{w} \coloneqq \boldsymbol{w} + \boldsymbol{v_w}$$

$$\boldsymbol{b} \coloneqq \boldsymbol{b} + \boldsymbol{v_b}$$

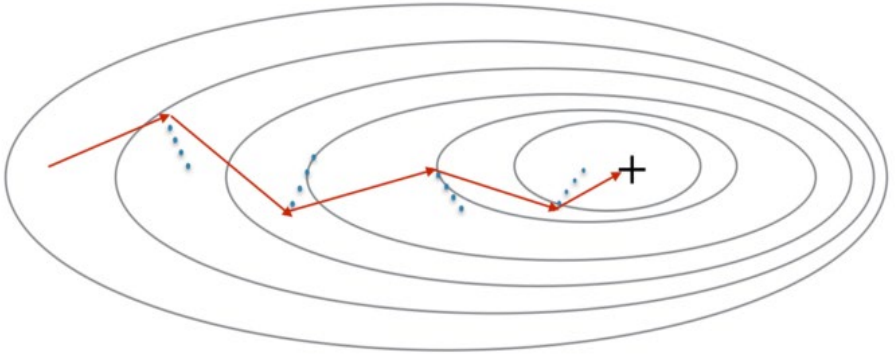
На практике обычно задают μ равным 0.5, 0.9 или 0.99.

SGD vs Momentum

Stochastic Gradient Descent



Momentum



Метод Нестерова (Nesterov Accelerated Gradient)

$\boldsymbol{v}$ – скорость движения шарика по поверхности (начальная скорость равна нулю)

μ – момент движения (величина, обратно пропорциональная трению шарика о поверхность): $0 \leq \mu \leq 1$

$$\acute{\boldsymbol{w}} \coloneqq \boldsymbol{w} + \mu \boldsymbol{v_w}$$

$$\acute{\boldsymbol{b}} \coloneqq \boldsymbol{b} + \mu \boldsymbol{v_b}$$

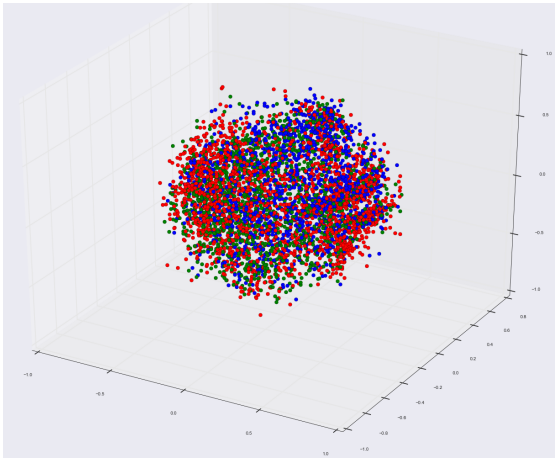
$$\boldsymbol{v_w} \coloneqq \mu \boldsymbol{v_w} - \eta \nabla_{\boldsymbol{w}} \mathbb{C}(\acute{\boldsymbol{w}})$$

$$\boldsymbol{v_b} \coloneqq \mu \boldsymbol{v_b} - \eta \nabla_{\boldsymbol{b}} \mathbb{C}(\acute{\boldsymbol{b}})$$

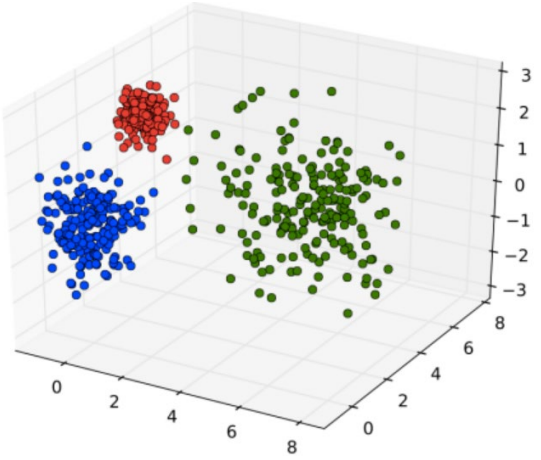
$$\boldsymbol{w} \coloneqq \boldsymbol{w} + \boldsymbol{v_w}$$

$$\boldsymbol{b} \coloneqq \boldsymbol{b} + \boldsymbol{v_b}$$

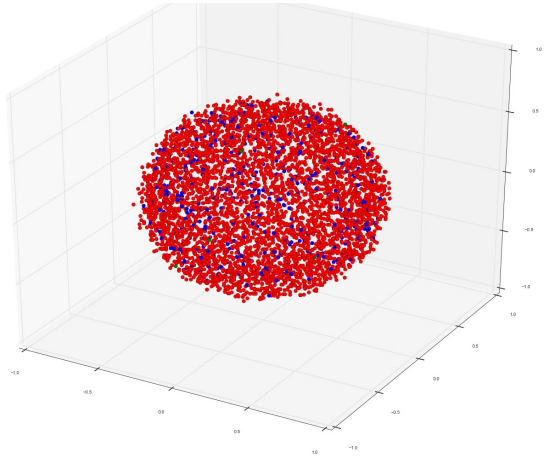
Плотные, разреженные и перекошенные обучающие данные



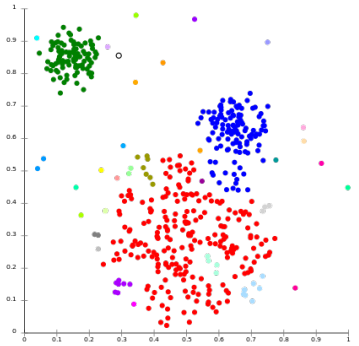
Плотные равномерные



Разреженные равномерные

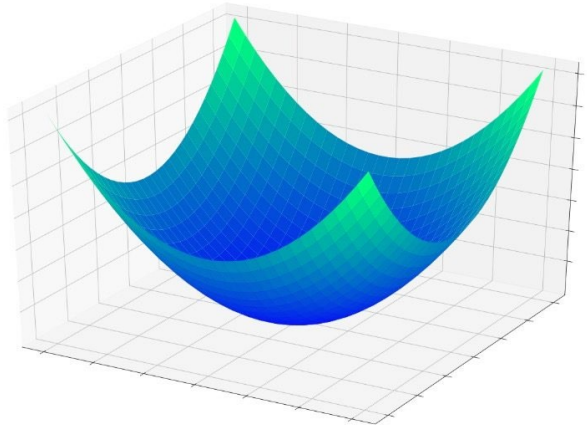


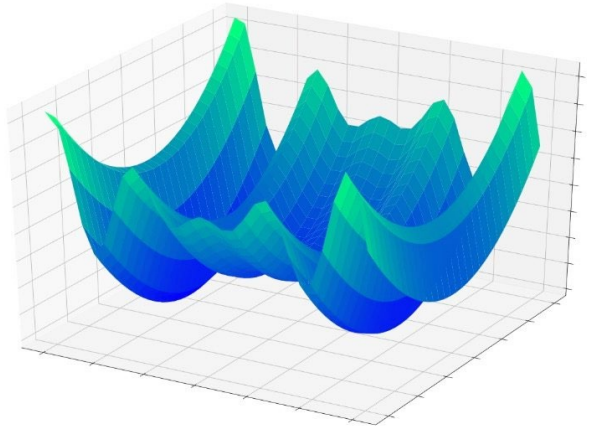
Плотные перекошенные

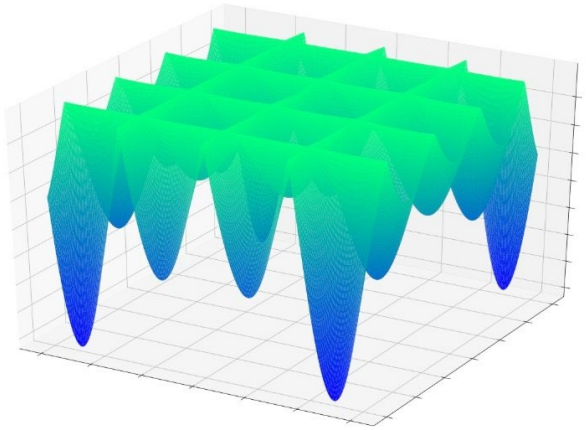


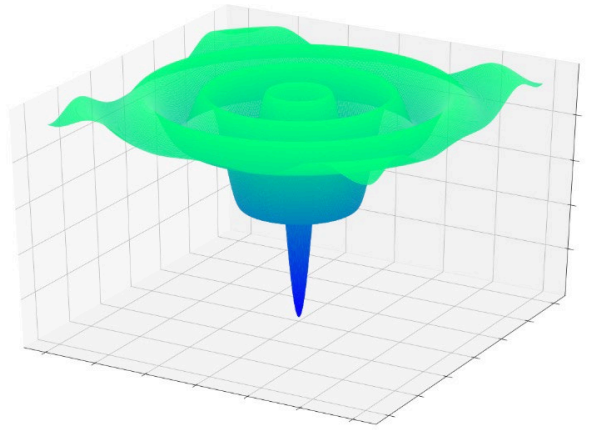
Разреженные перекошенные

Различные поверхности для поиска минимума ошибки

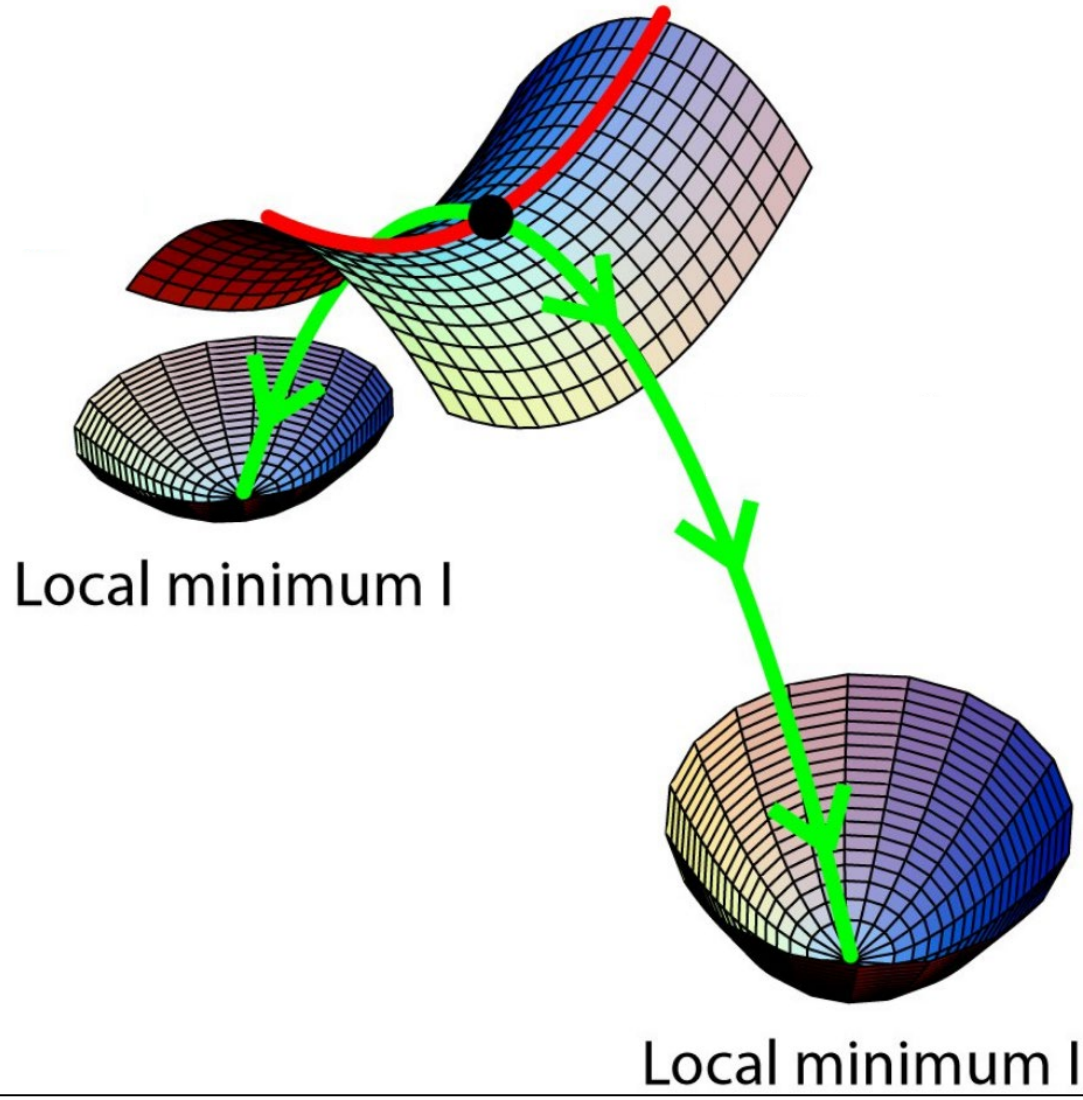








Седловидная впадина



AdaGrad (адаптивный градиентный спуск)

Алгоритм AdaGrad по отдельности адаптирует скорости обучения всех параметров модели, умножая их на коэффициент, обратно пропорциональный квадратному корню из суммы всех прошлых значений квадрата градиента:

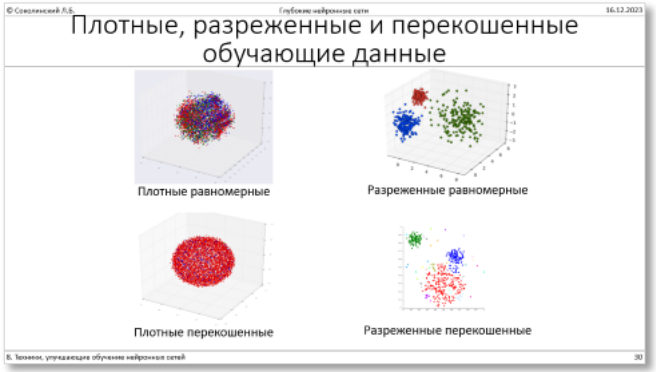
$$\boldsymbol{g}_{\boldsymbol{w}}^{(0)} = \mathbf{1} \qquad \boldsymbol{g}_{\boldsymbol{b}}^{(0)} = \mathbf{1}$$

$$\boldsymbol{g}_{\boldsymbol{w}}^{(i)} = \boldsymbol{g}_{\boldsymbol{w}}^{(i-1)} + \left(\nabla_{\boldsymbol{w}}\mathbb{C}\left(\boldsymbol{w}^{(i-1)}\right)\right)^2 \qquad \boldsymbol{g}_{\boldsymbol{b}}^{(i)} = \boldsymbol{g}_{\boldsymbol{b}}^{(i-1)} + \left(\nabla_{\boldsymbol{b}}\mathbb{C}\left(\boldsymbol{b}^{(i-1)}\right)\right)^2$$

$$h_{w_j}^{(i-1)} = \frac{1}{\sqrt{g_{w_j}^{(i-1)}}} \qquad h_{b_k}^{(i-1)} = \frac{1}{\sqrt{g_{b_k}^{(i-1)}}}$$

$$\boldsymbol{w}^{(i)} = \boldsymbol{w}^{(i-1)} - \boldsymbol{h}_{\boldsymbol{w}}^{(i-1)}\eta\nabla_{\boldsymbol{w}}\mathbb{C}\left(\boldsymbol{w}^{(i-1)}\right)$$

$$\boldsymbol{b}^{(i)} = \boldsymbol{b}^{(i-1)} - \boldsymbol{h}_{\boldsymbol{b}}^{(i-1)}\eta\nabla_{\boldsymbol{b}}\mathbb{C}\left(\boldsymbol{b}^{(i-1)}\right)$$



Преимущества и недостатки AdaGrad

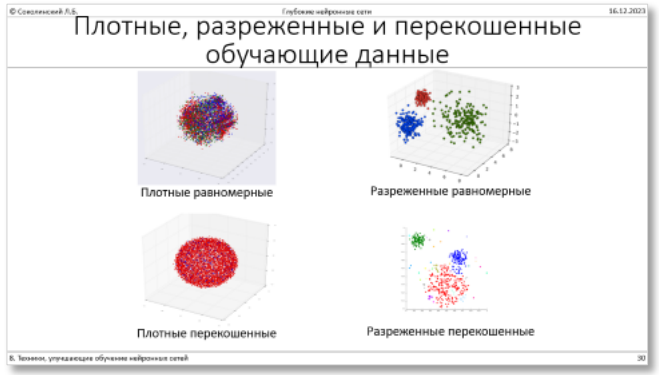
⊕ Для каждого параметра (вес w или смещение b) скорость подбирается индивидуально за счет g_w и g_b (можно без подбора взять $\eta = 0.01$)

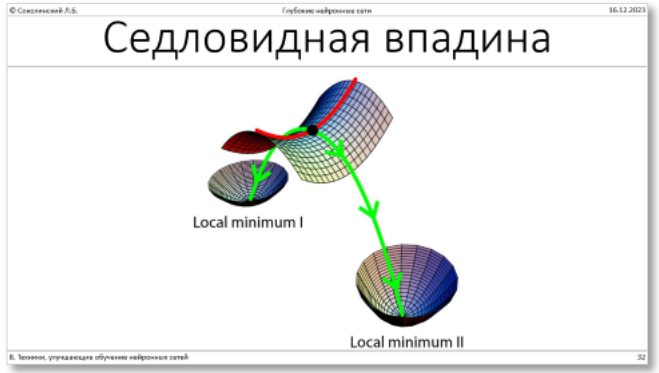
⊕ Редким, но важным признакам со временем отдается приоритет (метод подходит для разреженных и перекошенных обучающих данных)

⊖ g_w и g_b могут быстро увеличиваться, приводя к существенному замедлению скорости обучения

⊖ Начальная скорость обучения η одинакова для всех параметров (она может оказаться хороша для одних размерностей, но плоха для других)

⊖ Метод чувствителен к локальным минимумам и седловидным впадинам



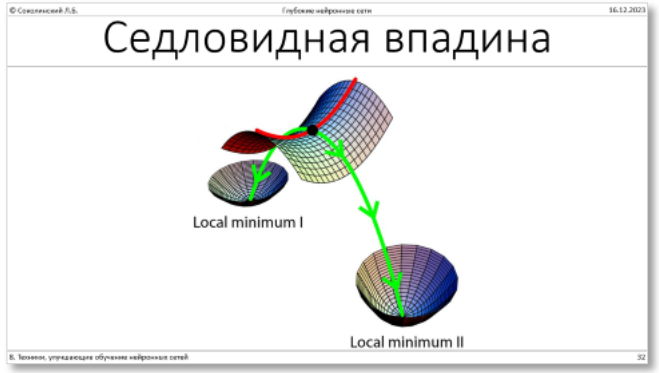


Алгоритм RMSProp

В алгоритме RMSProp используется *экспоненциально затухающее среднее*, т. е. далекое прошлое отбрасывается, чтобы повысить скорость сходимости после обнаружения седловидной впадины. Для этого вводится новый параметр $\rho \in [0; 1)$, регулирующий скорость затухания.

$$\boldsymbol{g}_w^{(0)} = \mathbf{1}; \qquad \boldsymbol{g}_b^{(0)} = \mathbf{1};$$

$$\boldsymbol{g}_w^{(i)} = \rho \boldsymbol{g}_w^{(i-1)} + (1-\rho)\left(\nabla_{\boldsymbol{w}} \mathbb{C}\left(\boldsymbol{w}^{(i-1)}\right)\right)^2; \quad \boldsymbol{g}_b^{(i)} = \rho \boldsymbol{g}_b^{(i-1)} + (1-\rho)\left(\nabla_{\boldsymbol{b}} \mathbb{C}\left(\boldsymbol{b}^{(i-1)}\right)\right)^2$$



$$h_{w_j}^{(i-1)} = \frac{1}{\delta + \sqrt{g_{w_j}^{(i-1)}}}; \qquad h_{b_k}^{(i-1)} = \frac{1}{\delta + \sqrt{g_{b_k}^{(i-1)}}}$$

$$\boldsymbol{w}^{(i)} = \boldsymbol{w}^{(i-1)} - \boldsymbol{h}_{\boldsymbol{w}}^{(i-1)} \eta \nabla_{\boldsymbol{w}} \mathbb{C}\left(\boldsymbol{w}^{(i-1)}\right)$$

$$\boldsymbol{b}^{(i)} = \boldsymbol{b}^{(i-1)} - \boldsymbol{h}_{\boldsymbol{b}}^{(i-1)} \eta \nabla_{\boldsymbol{b}} \mathbb{C}\left(\boldsymbol{b}^{(i-1)}\right)$$

δ – небольшая константа для обеспечения численной устойчивости (по умолчанию 10^{-7})

Алгоритм Adam (адаптивные моменты)

$\eta = 0.001$ // скорость обучения

$\rho_1 = 0.9; \quad \rho_2 = 0.999$ // коэффициенты затухания

$\delta = 10^{-8}$ // небольшая константа для обеспечения устойчивости

$\boldsymbol{s}_w^{(0)} = \mathbf{0}; \quad \boldsymbol{s}_b^{(0)} = \mathbf{0}$ // 1-й момент

$\boldsymbol{r}_w^{(0)} = \mathbf{0}; \quad \boldsymbol{r}_b^{(0)} = \mathbf{0}$ // 2-й момент

$\boldsymbol{g}_w^{(i)} = \nabla_{\boldsymbol{w}} \mathbb{C}\left(\boldsymbol{w}^{(i-1)}\right); \quad \boldsymbol{g}_b^{(i)} = \nabla_{\boldsymbol{b}} \mathbb{C}\left(\boldsymbol{b}^{(i-1)}\right)$ // градиенты

$\boldsymbol{s}_w^{(i)} = \rho_1 \boldsymbol{s}_w^{(i-1)} + (1-\rho_1)\boldsymbol{g}_w^{(i)}; \quad \boldsymbol{s}_b^{(i)} = \rho_1 \boldsymbol{s}_b^{(i-1)} + (1-\rho_1)\boldsymbol{g}_b^{(i)}$ // смещенная оценка 1-го момента

$\boldsymbol{r}_w^{(i)} = \rho_2 \boldsymbol{r}_w^{(i-1)} + (1-\rho_2)\boldsymbol{g}_w^{(i)} \circ \boldsymbol{g}_w^{(i)}; \quad \boldsymbol{r}_b^{(i)} = \rho_2 \boldsymbol{r}_b^{(i-1)} + (1-\rho_2)\boldsymbol{g}_b^{(i)} \circ \boldsymbol{g}_b^{(i)}$ // смещенная оценка 2-го момента

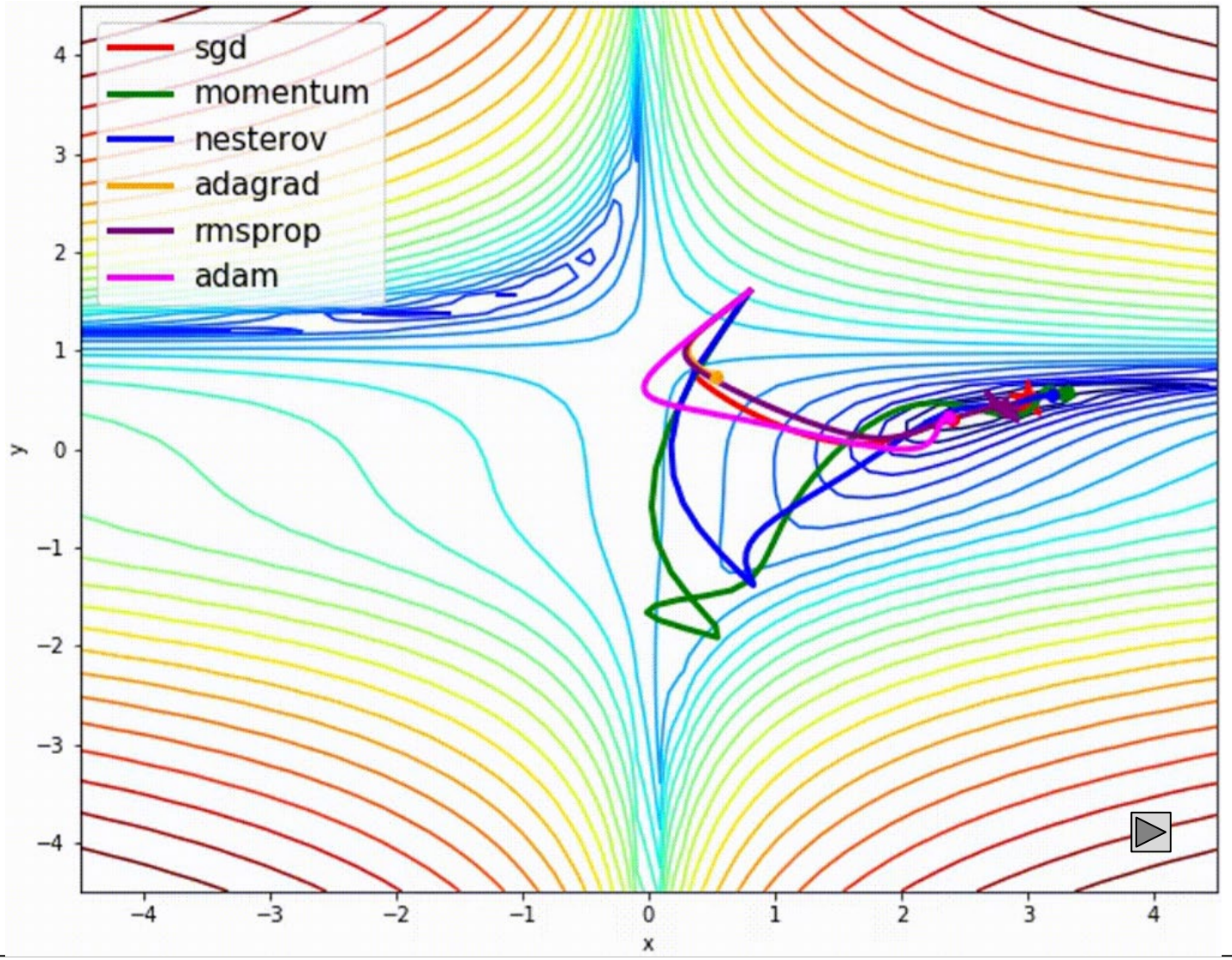
$\hat{\boldsymbol{s}}_w = \frac{\boldsymbol{s}_w^{(i)}}{1-\rho_1}; \quad \hat{\boldsymbol{s}}_b = \frac{\boldsymbol{s}_b^{(i)}}{1-\rho_1}$ // корректировка смещения 1-го момента

$\hat{\boldsymbol{r}}_w = \frac{\boldsymbol{r}_w^{(i)}}{1-\rho_2}; \quad \hat{\boldsymbol{r}}_b = \frac{\boldsymbol{r}_b^{(i)}}{1-\rho_2}$ // корректировка смещения 2-го момента

$$\boldsymbol{w}^{(i)} = \boldsymbol{w}^{(i-1)} - \eta \frac{\hat{\boldsymbol{s}}_w}{\sqrt{\hat{\boldsymbol{r}}_w} + \delta}; \qquad \boldsymbol{b}^{(i)} = \boldsymbol{b}^{(i-1)} - \eta \frac{\hat{\boldsymbol{s}}_b}{\sqrt{\hat{\boldsymbol{r}}_b} + \delta}$$

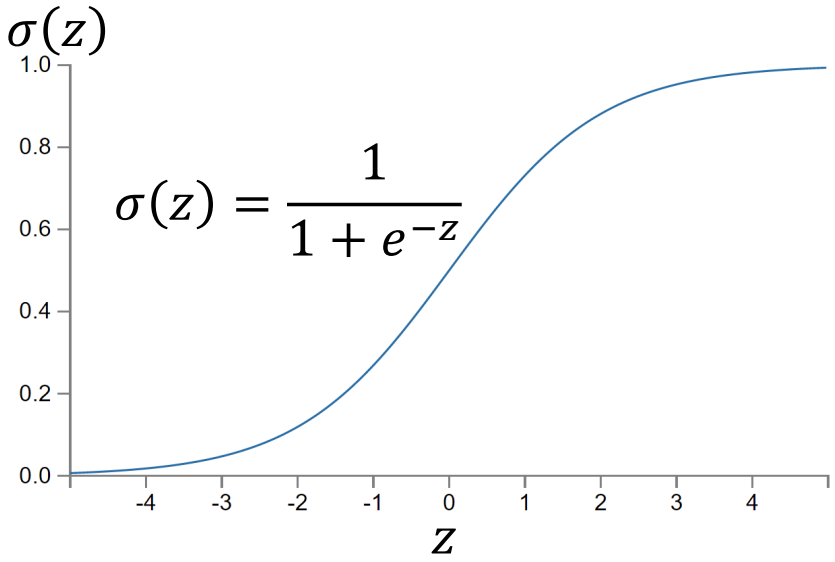
Adam соединяет преимущества RMSProp и Momentum

Сравнение алгоритмов-оптимизаторов

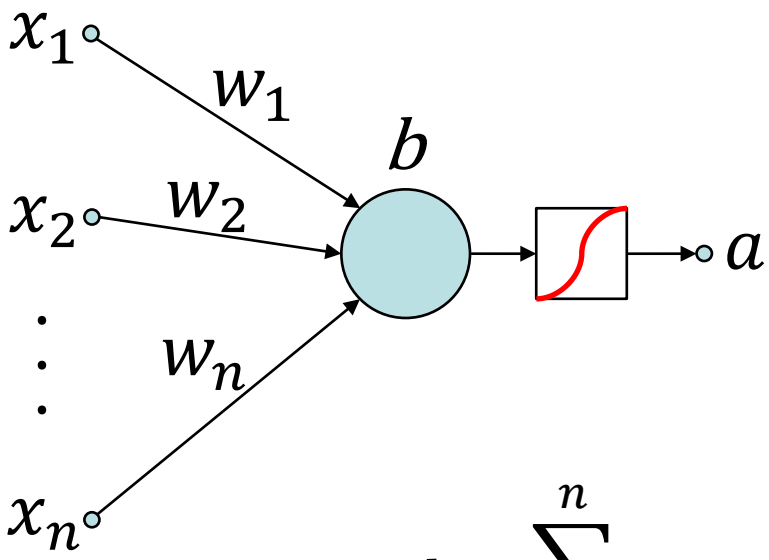


Альтернативные модели нейрона (Other models of artificial neuron)

Сигмоид (Sigmoid)



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

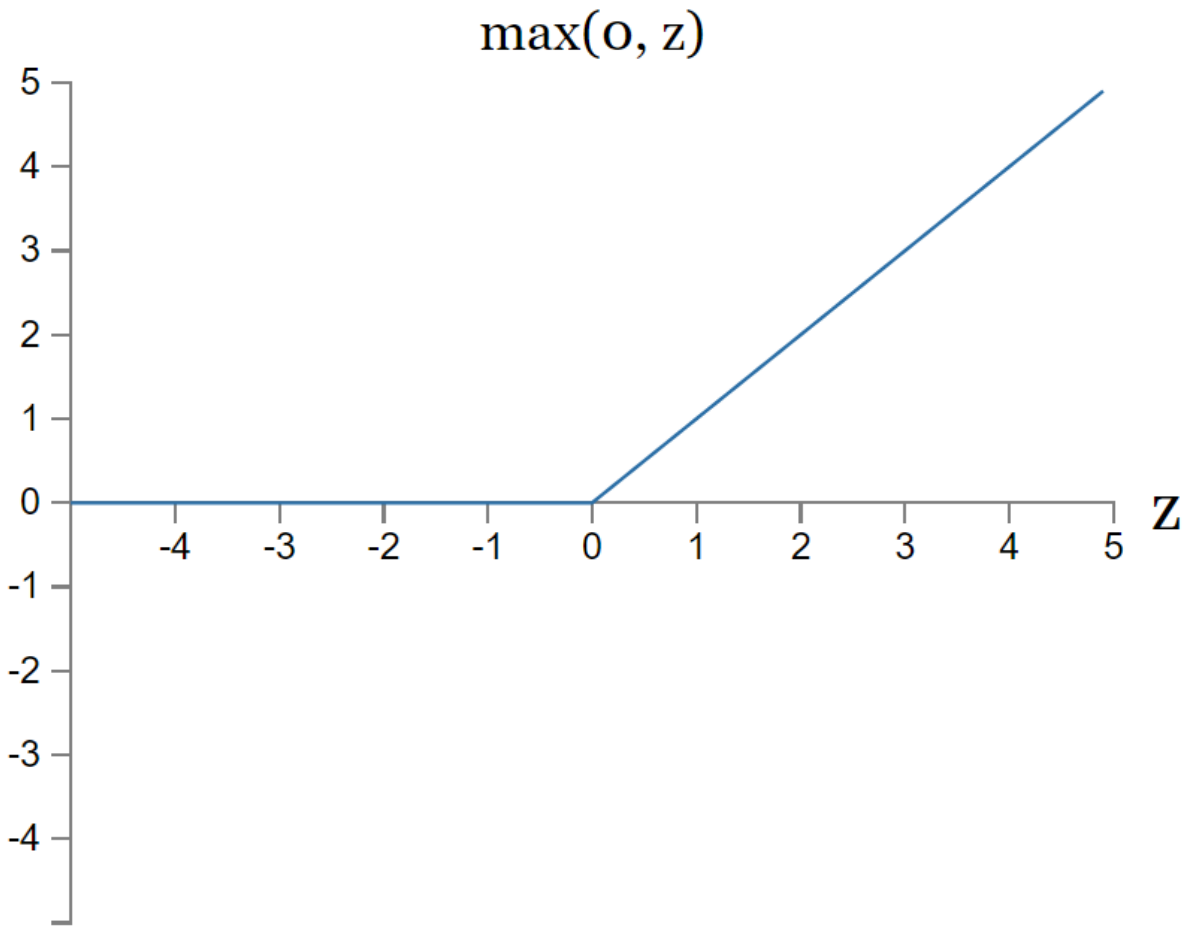


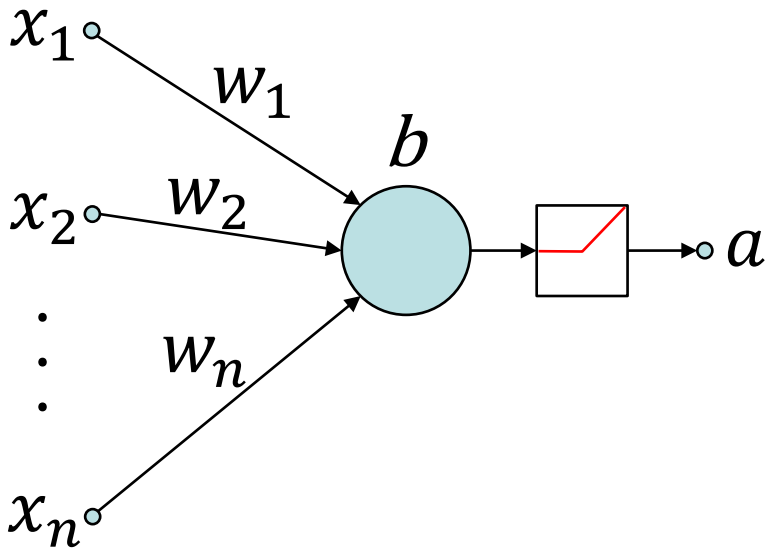
$$z = b + \sum_{i=1}^{n} w_i x_i$$

$$a = \frac{1}{1 + e^{-b-\sum_{i=1}^{n} w_i x_i}}$$

\+ Соответствует биологическому нейрону
\+ Отсутствует проблема мертвых нейронов
\+ Дифференцируемость в нуле
\+ Симметричность относительно нуля
\+ Выходной сигнал ограничен сверху
– Отсутствует инвариантность относительно умножения на константу
– Низкая вычислительная эффективность
– Проблема исчезающего градиента

Линейный выпрямитель (ReLU – Rectified Linear Unit)

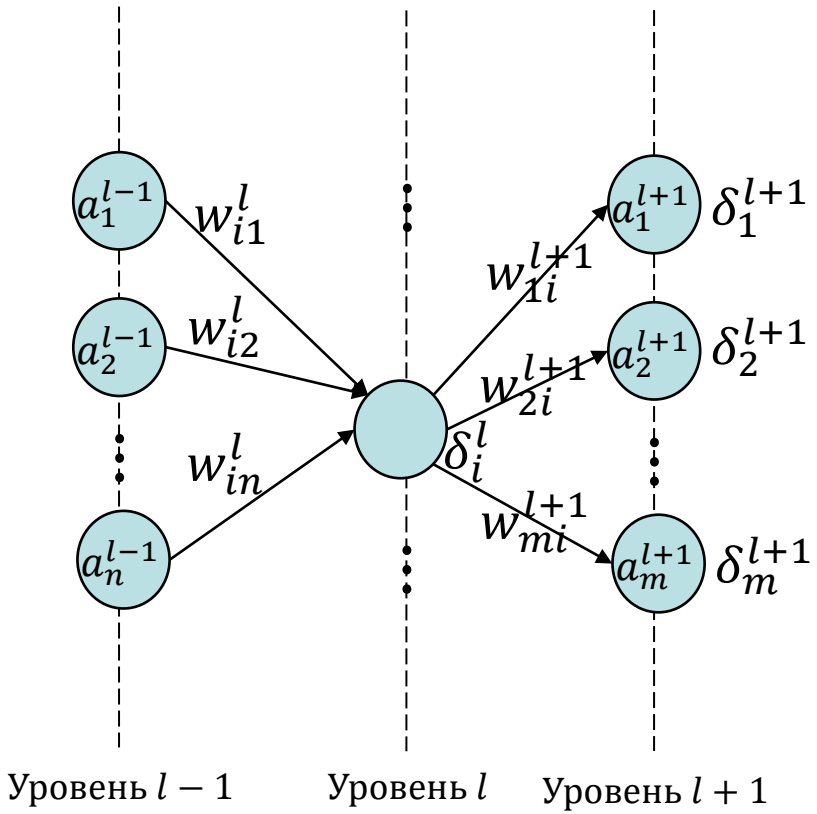




$$z = b + \sum_{i=1}^{n} w_i x_i$$

$$f(z) = \max(0, z)$$

Особенности линейного выпрямителя ReLU



Метод обратного распространения ошибки:

$$\frac{\partial C}{\partial w_{ij}^l} = \delta_i^l \cdot a_j^{l-1}$$

$$\delta_i^l = \sum_j w_{ji}^{l+1} \delta_j^{l+1} f'\left(z_i^l\right)$$

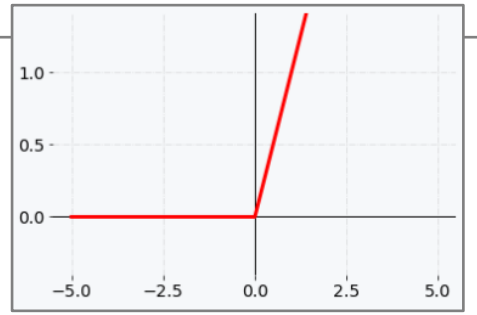
$$\Downarrow$$

Если $z_i^l > 0$, то $f'\left(z_i^l\right) = 1 \Rightarrow$ загоризонталивание не происходит $\Rightarrow$ не происходит замедление обучения

Если $z_i^l < 0$, то $f'\left(z_i^l\right) = 0 \Rightarrow$ обучение не происходит

ReLU – наиболее популярная функция активации в современных глубоких нейронных сетях

Сильные и слабые стороны линейного выпрямителя ReLU



Преимущества

1. **Соответствие биологическому нейрону**
   (в мозге не возникает отрицательных электрических потенциалов)
2. **Разреженная активация** (при использовании нормального распределения для начальной инициализации весов около 50% скрытых нейронов будут иметь ненулевой выходной сигнал)
3. **Быстрая обучаемость** (отсутствует проблема исчезающего градиента)
4. **Высокая вычислительная эффективность** (необходимы только сравнения, сложения и умножения)
5. **Инвариантность относительно умножения на константу** ($\max\{0; hz\} = h\max\{0; z\}$ при $h \geq 0$)

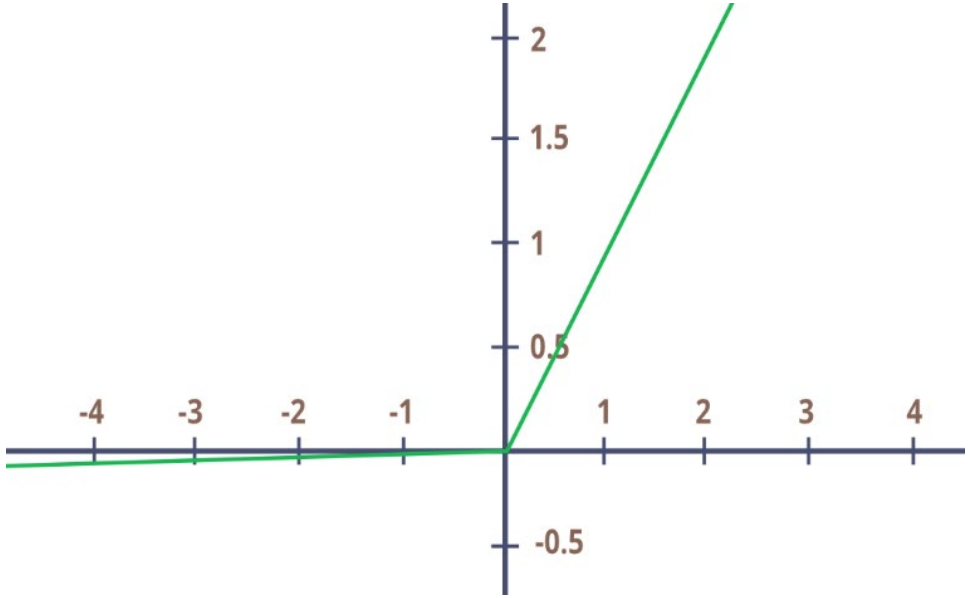
Потенциальные проблемы

1. **Недифференцируемость в нуле** (положить производную в нуле равной 0 или 1)
2. **Несимметричность относительно нуля**
3. **Выходной сигнал неограничен сверху** (использовать *softmax* для выходного слоя)
4. **Возникновение мертвых ReLU нейронов**

Проблема мертвых ReLU нейронов

- ReLU нейроны иногда могут попадать в состояния, при которых они становятся неактивными практически для всех прецедентов из обучающей выборки
- В этом случае нейроны перестают обучаться и «умирают»
- Бывают ситуации, когда количество мертвых ReLU нейронов может стать настолько большим, что это будет существенно снижать эффективность нейронной сети
- Проблема мертвых ReLU нейронов обычно возникает при слишком большой скорости обучения

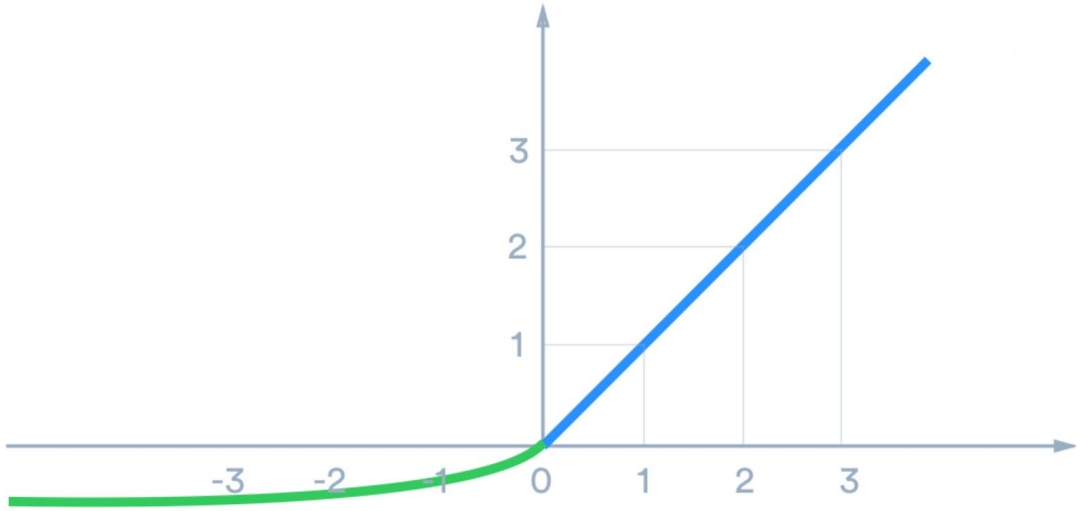
Линейный выпрямитель с утечкой (Leaky ReLU)



$$f(z) = \begin{cases} z & |z > 0 \\ 0.01z & |z \leq 0 \end{cases}$$

\+ Решает проблему мертвых нейронов
\+ Высокая вычислительная эффективность
\+ Инвариантность относительно умножения на константу
– Недифференцируемость в нуле
– Несоответствие биологическому нейрону
– Несимметричность относительно нуля
– Выходной сигнал неограничен сверху

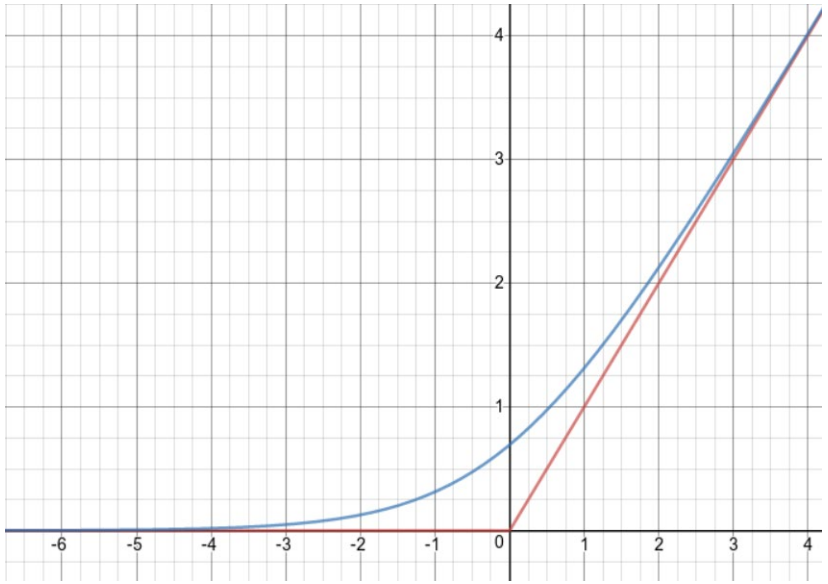
Экспоненциальная линейная функция (Exponential Linear Unit, ELU)



$$f(z) = \begin{cases} z & |z > 0 \\ e^z - 1 & |z \leq 0 \end{cases}$$

\+ Решает проблему мертвых нейронов
\+ Дифференцируемость в нуле
– Отсутствует инвариантность относительно умножения на константу
– Несоответствие биологическому нейрону
– Несимметричность относительно нуля
– Выходной сигнал неограничен сверху
– Низкая вычислительная эффективность

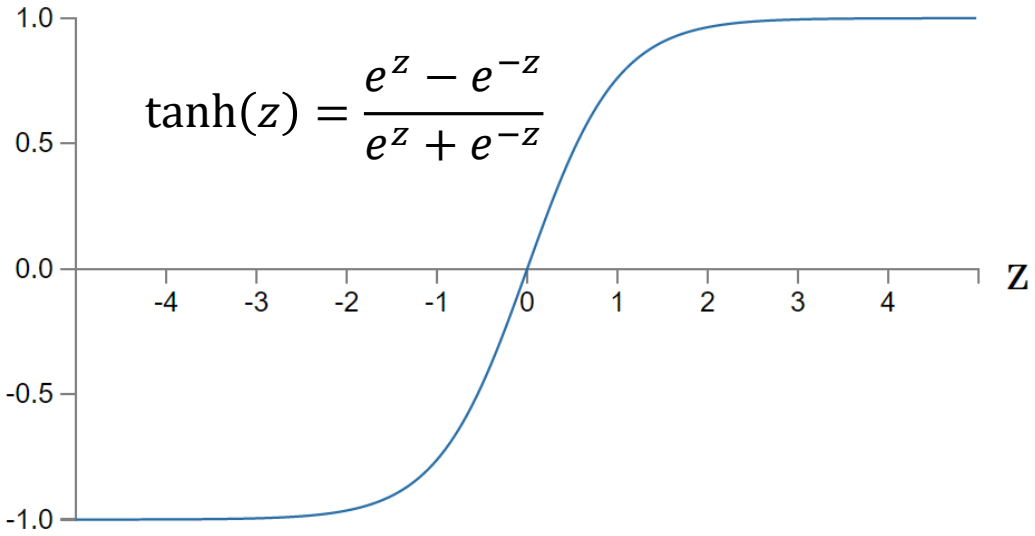
SoftPlus



$$f(z) = \ln(1 + e^z)$$

\+ Решает проблему мертвых нейронов
\+ Дифференцируемость в нуле
\+ Соответствие биологическому нейрону
– Отсутствует инвариантность относительно умножения на константу
– Несимметричность относительно нуля
– Выходной сигнал неограничен сверху
– Низкая вычислительная эффективность

Нейрон на основе гиперболического тангенса (tanh neuron)

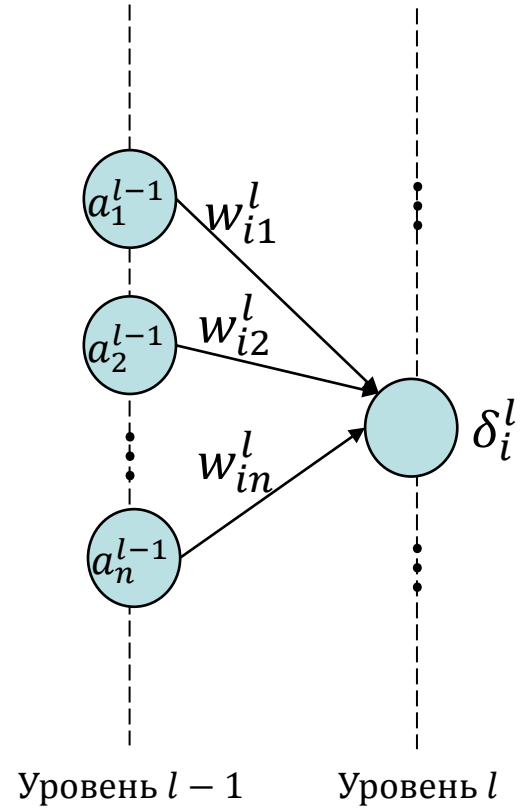


$$\tanh(z) \equiv \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$= 2\frac{1}{1 + e^{-2z}} - 1 = 2\sigma(2z) - 1$$

\+ Решает проблему мертвых нейронов
\+ Дифференцируемость в нуле
\+ Симметричность относительно нуля
\+ Выходной сигнал ограничен сверху
– Отсутствует инвариантность относительно умножения на константу
– Несоответствие биологическому нейрону
– Низкая вычислительная эффективность

Отличие гиперболического тангенса от сигмоида



Метод обратного распространения ошибки:

$$\frac{\partial C}{\partial w_{ij}^l} = \delta_i^l \cdot a_j^{l-1}$$

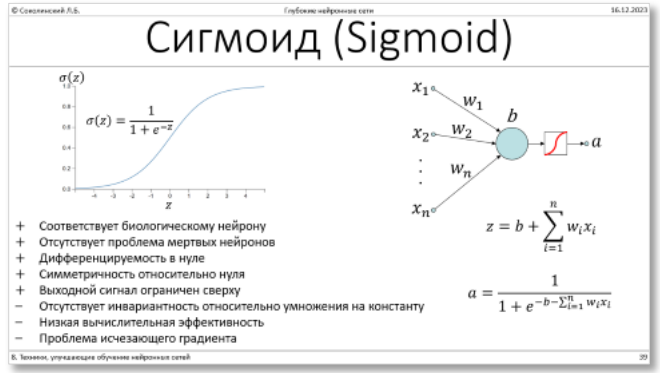
Сигмоид $\Rightarrow a_j^{l-1} > 0$

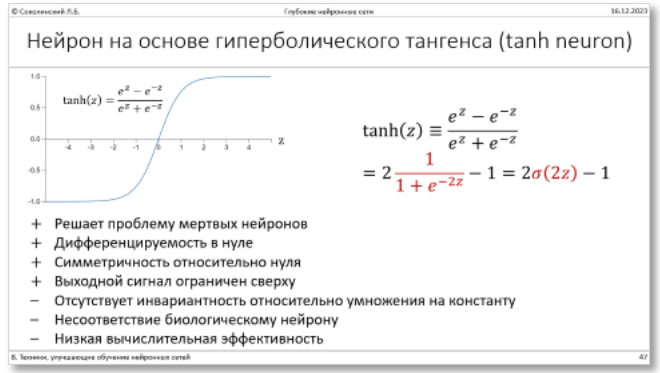
$\Downarrow$

Если $\delta_i^l < 0$, то все $w_{i1}^l, \cdots, w_{in}^l$ должны увеличиваться

Если $\delta_i^l > 0$, то все $w_{i1}^l, \cdots, w_{in}^l$ должны уменьшаться

При использовании ***tanh*** часть $w_{i1}^l, \cdots, w_{in}^l$ могут уменьшаться, а часть увеличиваться в зависимости от знака $a_1^{l-1}, \cdots, a_n^{l-1}$





Конец лекции 8

Вспомогательные слайды

Доказательство $\int_{-\infty}^{+\infty} te^{-t^2}dt = 0$

$$\int te^{-t^2}dt = -\frac{1}{2}\int e^{-t^2}d(-t^2) = -\frac{1}{2}e^{-t^2} + C$$

$$\int_0^{+\infty} te^{-t^2}dt = \lim_{v\to\infty}\int_0^{v} te^{-t^2}dt = \lim_{v\to\infty}\left(-\frac{1}{2}e^{-v^2} + \frac{1}{2}\right) = \frac{1}{2}$$

$$\int_{-\infty}^{0} te^{-t^2}dt = \lim_{u\to-\infty}\int_u^{0} te^{-t^2}dt = \lim_{u\to-\infty}\left(-\frac{1}{2} + \frac{1}{2}e^{-u^2}\right) = -\frac{1}{2}$$

$$\int_{-\infty}^{+\infty} te^{-t^2}dt = \frac{1}{2} - \frac{1}{2} = 0$$

Интеграл Эйлера-Пуассона

$$\int_0^{+\infty} e^{-t^2} dt = \frac{\sqrt{\pi}}{2}$$

Демидович Б.П., Кудрявцев В.А. Краткий курс высшей математики: Учеб. пособие для вузов / Б.П. Демидович, В.А. Кудрявцев, Москва: ООО «Издательство Астрель»; ООО «Издательство АСТ», 2001. 656 с.

Интегрирование по частям

$$\int t\,dv = tv - \int v\,dt$$

$$\int t \cdot 2te^{-t^2}dt = \left| \begin{matrix} dv = 2te^{-t^2}dt = d\left(-e^{-t^2}\right) \\ v = -e^{-t^2} \end{matrix} \right| =$$

$$= -te^{-t^2} + \int e^{-t^2}dt$$

$$\int_{-\infty}^{+\infty} te^{-t^2}dt = 0$$

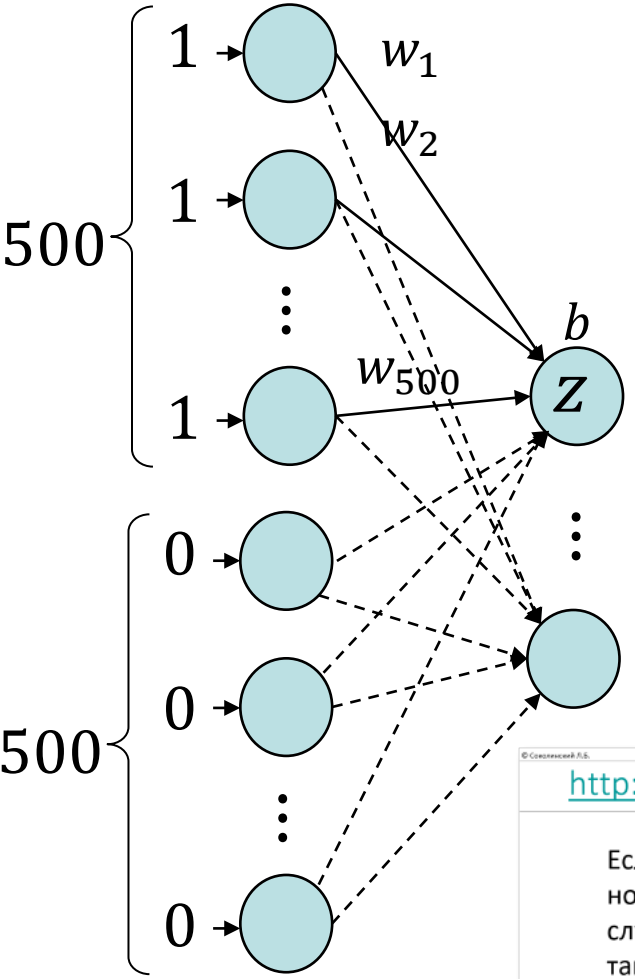
$$\int te^{-t^2}dt = -\frac{1}{2}\int e^{-t^2}d(-t^2) = -\frac{1}{2}e^{-t^2} + C$$

$$\int_{0}^{+\infty} te^{-t^2}dt = \lim_{v\to\infty}\int_{0}^{v} te^{-t^2}dt = \lim_{v\to\infty}\left(-\frac{1}{2}e^{-v^2} + \frac{1}{2}\right) = \frac{1}{2}$$

$$\int_{-\infty}^{0} te^{-t^2}dt = \lim_{u\to-\infty}\int_{u}^{0} te^{-t^2}dt = \lim_{u\to-\infty}\left(-\frac{1}{2} + \frac{1}{2}e^{-u^2}\right) = -\frac{1}{2}$$

$$\int_{-\infty}^{+\infty} te^{-t^2}dt = \frac{1}{2} - \frac{1}{2} = 0$$

Вычисление математического ожидания и среднего квадратического отклонения для z



Активационный потенциал: $z = b + \sum_{j=1}^{500} w_j$

$\Downarrow$

z – сумма 501 независимой случайной величины

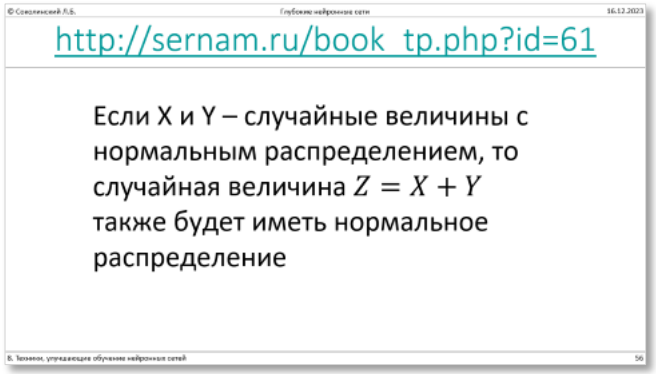
$a = 0$ для всех слагаемых

$\sigma = \sqrt{D(X)} = 1$ (то есть $D(X) = 1$) для всех слагаемых

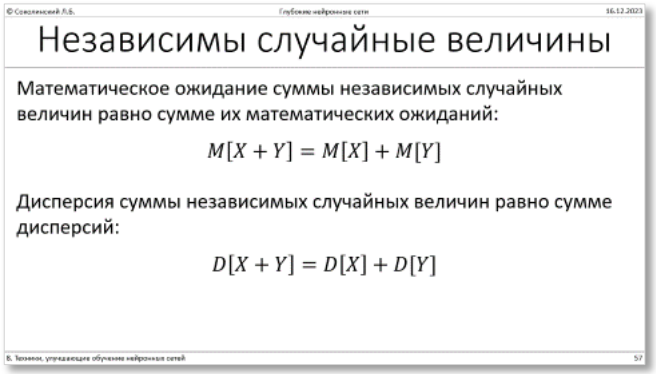
$\Downarrow$

z – случайная величина с нормальным распределением при $a = 0$ и

$\sigma = \sqrt{500 + 1} = \sqrt{501} \approx 22.4$


http://sernam.ru/book_tp.php?id=61

Если X и Y – случайные величины с нормальным распределением, то случайная величина $Z = X + Y$ также будет иметь нормальное распределение

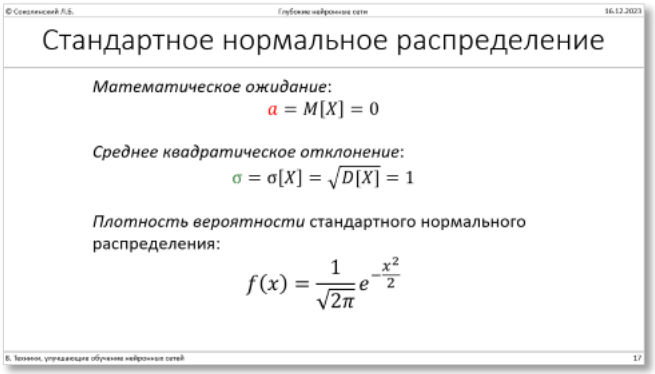

Независимы случайные величины

Математическое ожидание суммы независимых случайных величин равно сумме их математических ожиданий:

$$M[X + Y] = M[X] + M[Y]$$

Дисперсия суммы независимых случайных величин равно сумме дисперсий:

$$D[X + Y] = D[X] + D[Y]$$


Стандартное нормальное распределение

Математическое ожидание:

$$a = M[X] = 0$$

Среднее квадратическое отклонение:

$$\sigma = \sigma[X] = \sqrt{D[X]} = 1$$

Плотность вероятности стандартного нормального распределения:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

http://sernam.ru/book_tp.php?id=61

Если X и Y – случайные величины с нормальным распределением, то случайная величина $Z = X + Y$ также будет иметь нормальное распределение

Независимы случайные величины

Математическое ожидание суммы независимых случайных величин равно сумме их математических ожиданий:

$$M[X + Y] = M[X] + M[Y]$$

Дисперсия суммы независимых случайных величин равно сумме дисперсий:

$$D[X + Y] = D[X] + D[Y]$$

Обучение нейронной сети методом градиентного спуска

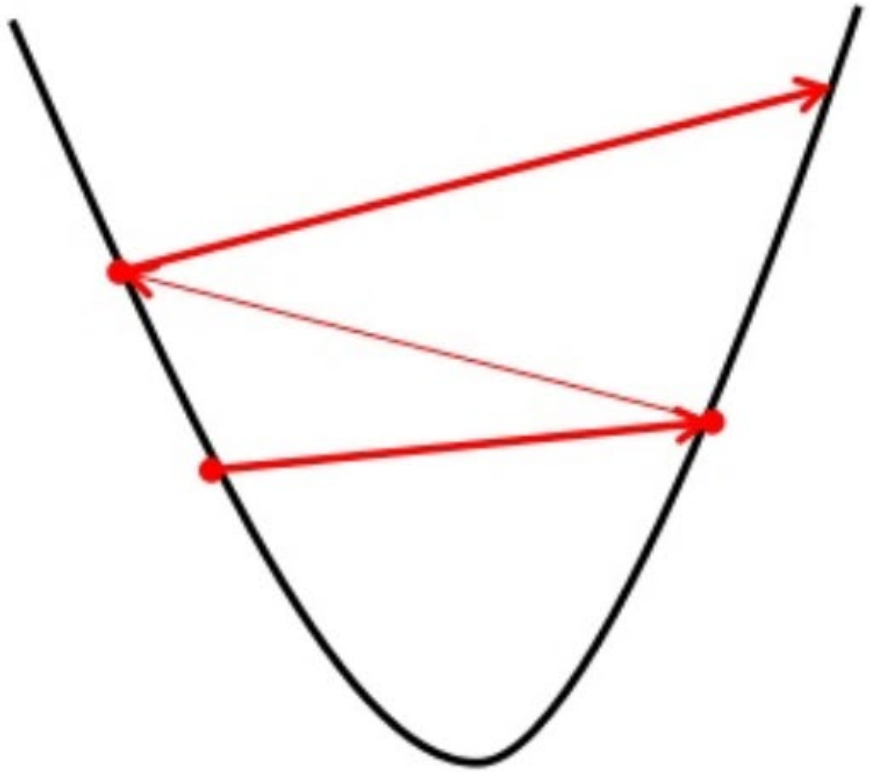
Необходимо минимизировать среднеквадратичную ошибку:

$$\mathbb{C} = \frac{1}{|V|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in V} \frac{\|\boldsymbol{\alpha}(\boldsymbol{x}) - \boldsymbol{y}\|^2}{2}$$

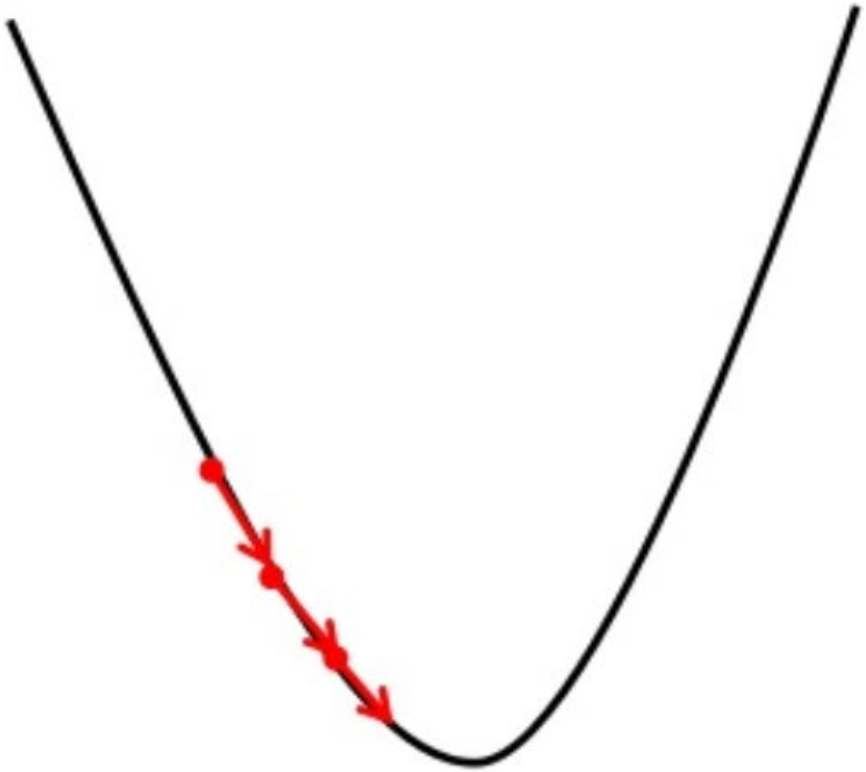
$$\boldsymbol{w} := \boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} \mathbb{C}$$

$$\boldsymbol{b} := \boldsymbol{b} - \eta \nabla_{\boldsymbol{b}} \mathbb{C}$$

Выбор скорости обучения η



Большая скорость обучения:
метод расходится (ошибка увеличивается)



Малая скорость обучения:
метод медленно сходится

Высокая скорость обучения предотвращает остановку в локальном минимуме