



Южный федеральный
университет



@SINECOR

Как работают нейронные сети

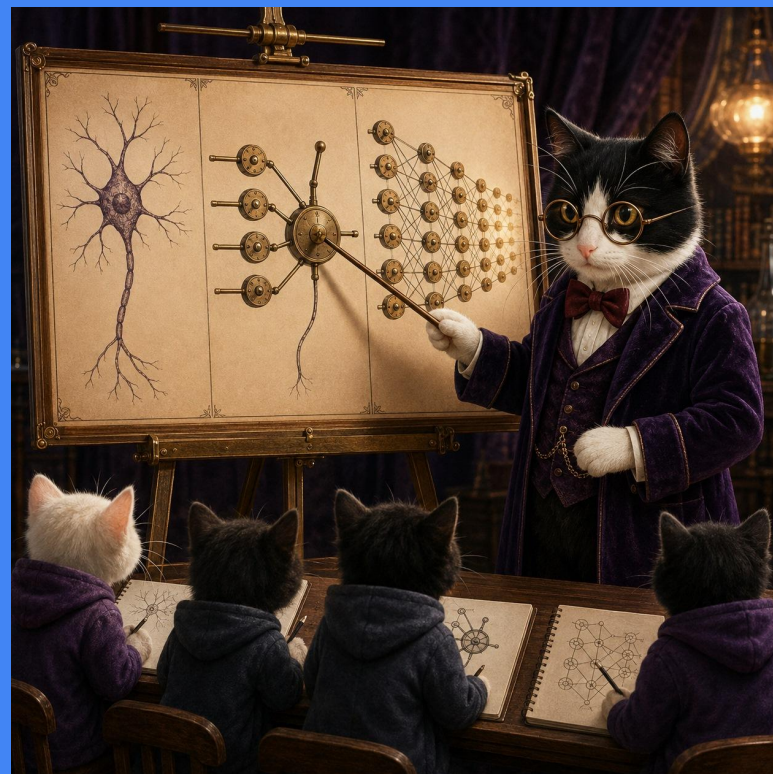


Сергей Николенко
28 сентября 2026 г.



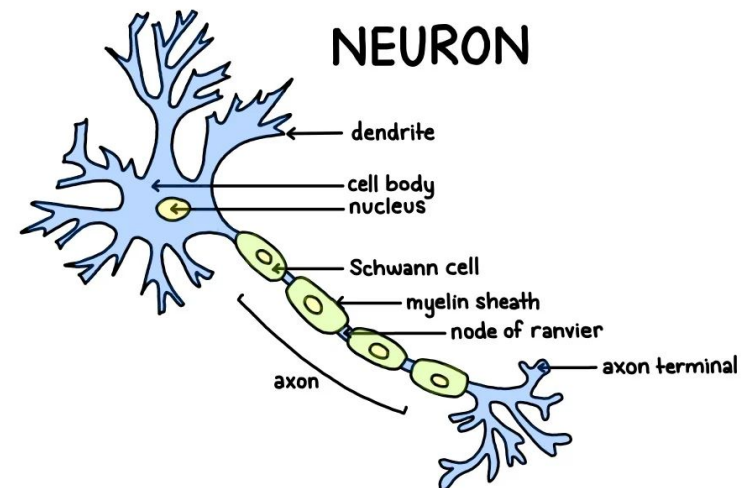
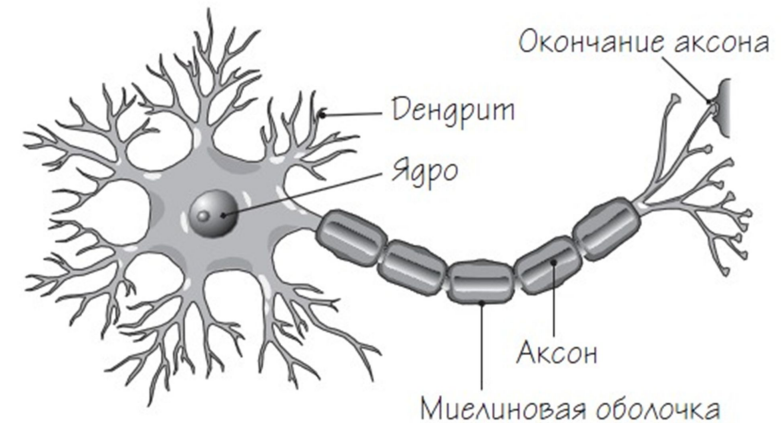
Технологии и фронтиры науки о данных

От одного нейрона к графу вычислений



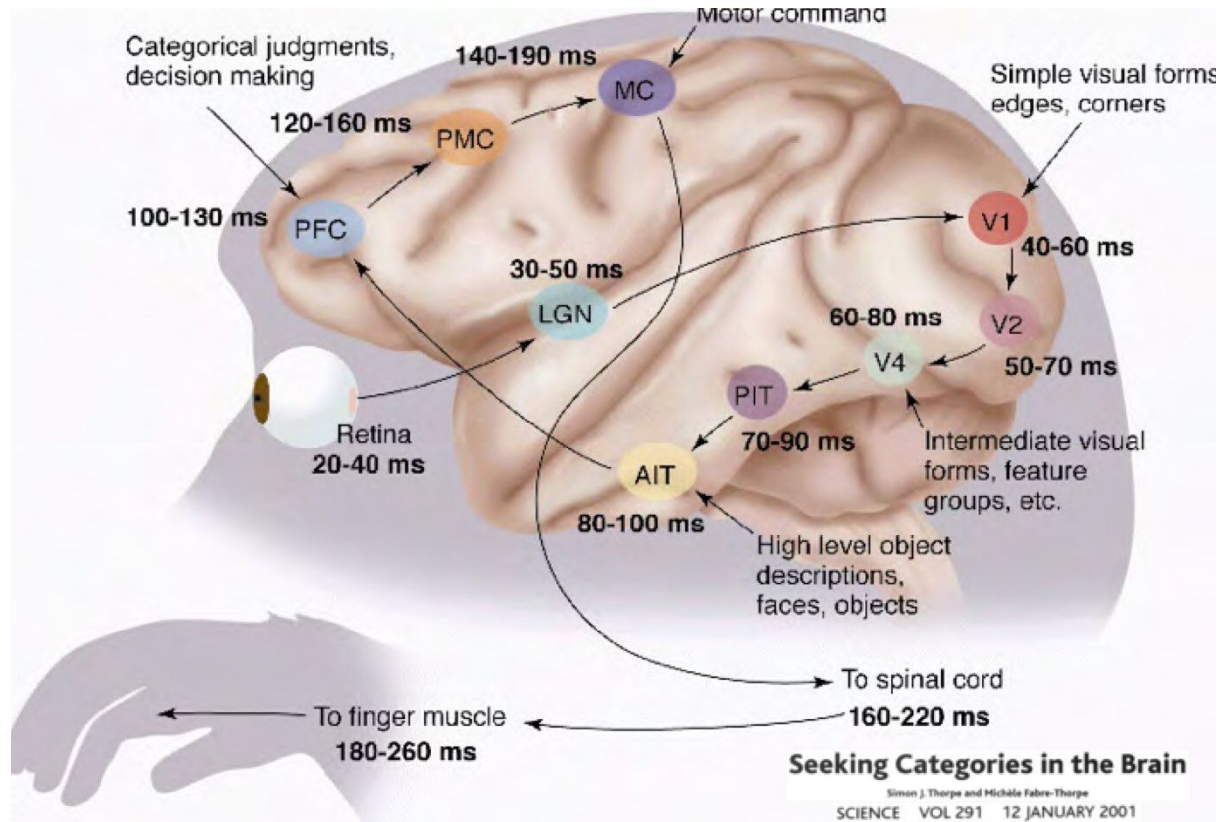
От биологии к информатике

- Мозг состоит из нейронов, которые устроены примерно одинаково
- Нейрон через дендриты получает сигнал от других нейронов, время от времени запускает сигнал по аксону, и через синапсы сигнал аксона доходит до дендритов других нейронов.
- Нейронов очень много: 10^{11} нейронов, в среднем 7000 связей у каждого, то есть 10^{15} синапсов



От биологии к информатике

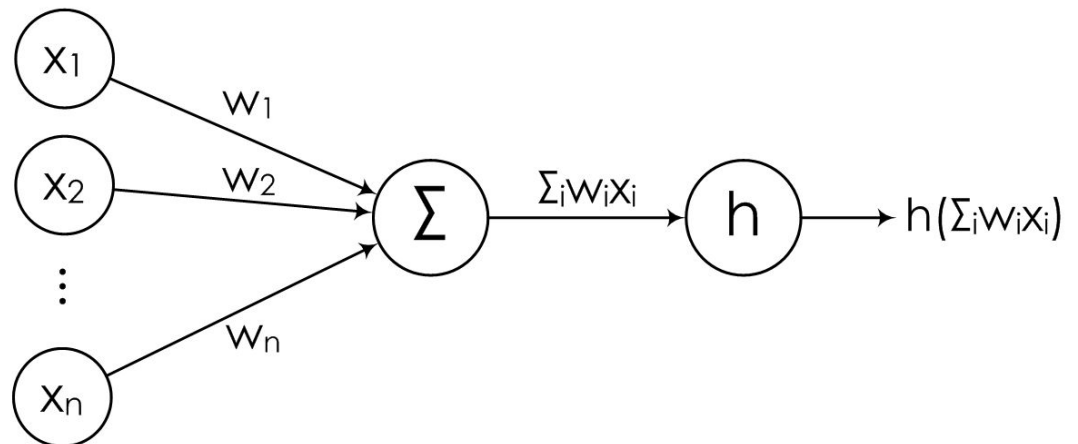
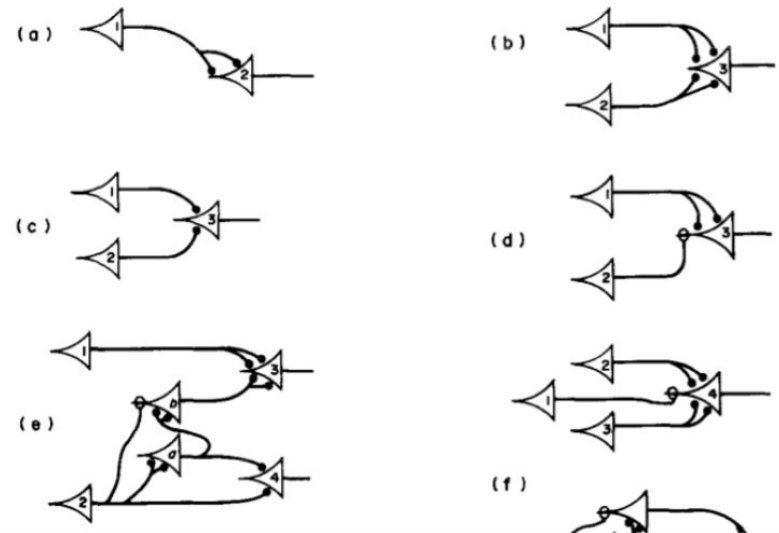
- Но «тактовая частота» этого «компьютера» очень низкая: от 10 до 200 герц.
- И при этом мы, например, распознаём лицо человека за 200-300 миллисекунд.
- Это значит, что цепочки короткие, и вычисления в мозге скорее параллельные, чем последовательные.



McCulloch, Pitts, 1943

- Искусственные нейросети появились даже раньше, чем тест Тьюринга и Дартмутский семинар
- (Маккаллох, Питтс, 1943): первая модель нейрона, которая была фактически современной!

LOGICAL CALCULUS FOR NERVOUS ACTIVITY 105



Перцептрон Розенблатта

- Как раз эту модель реализовывал и обучал первый перцептрон (Rosenblatt, 1958), о котором мы говорили в прошлый раз
- Это была первая реально работающая модель машинного обучения

FIG. 1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

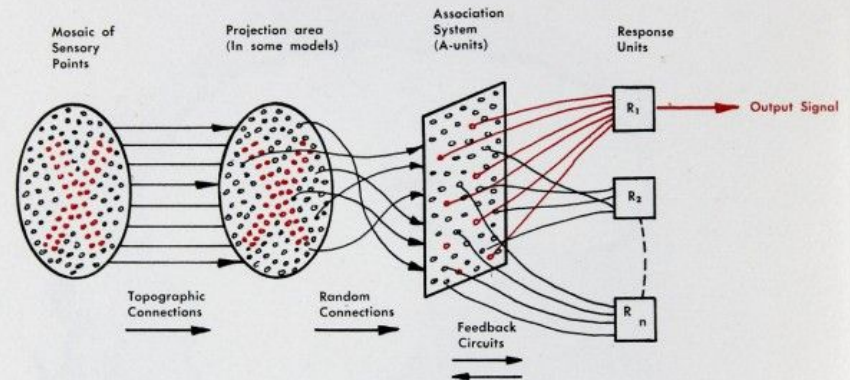
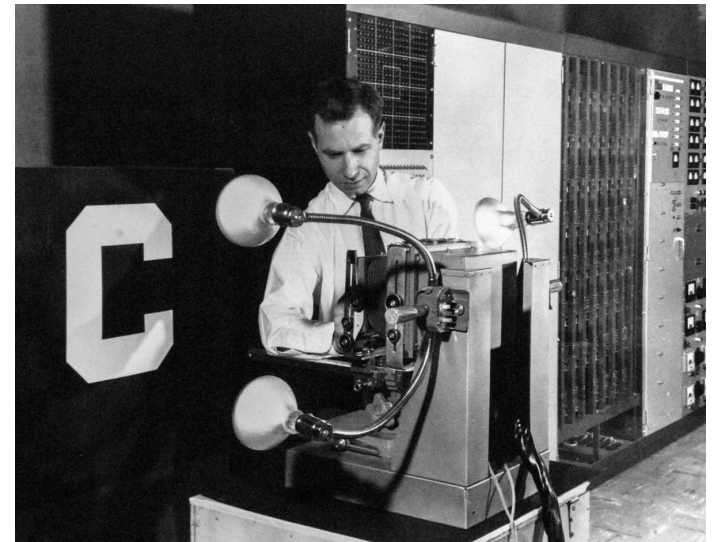
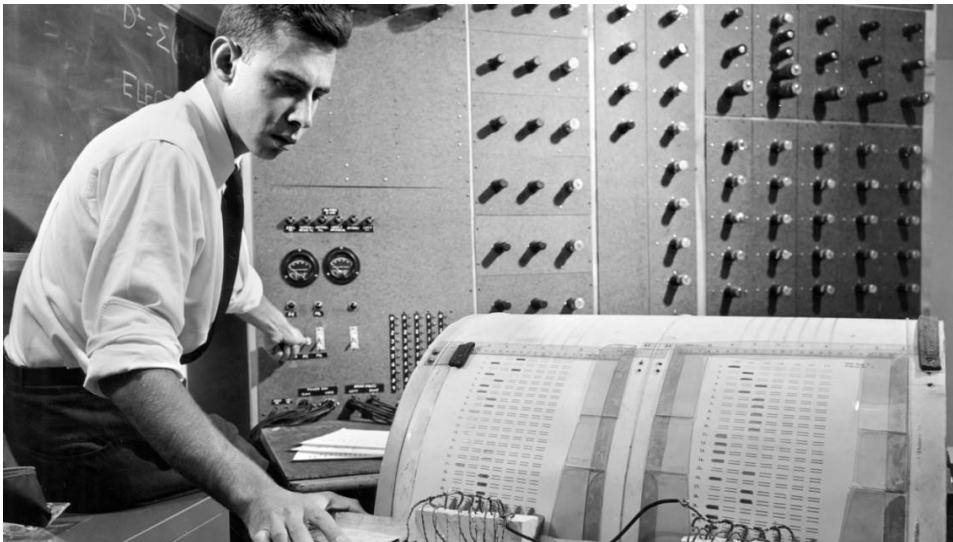
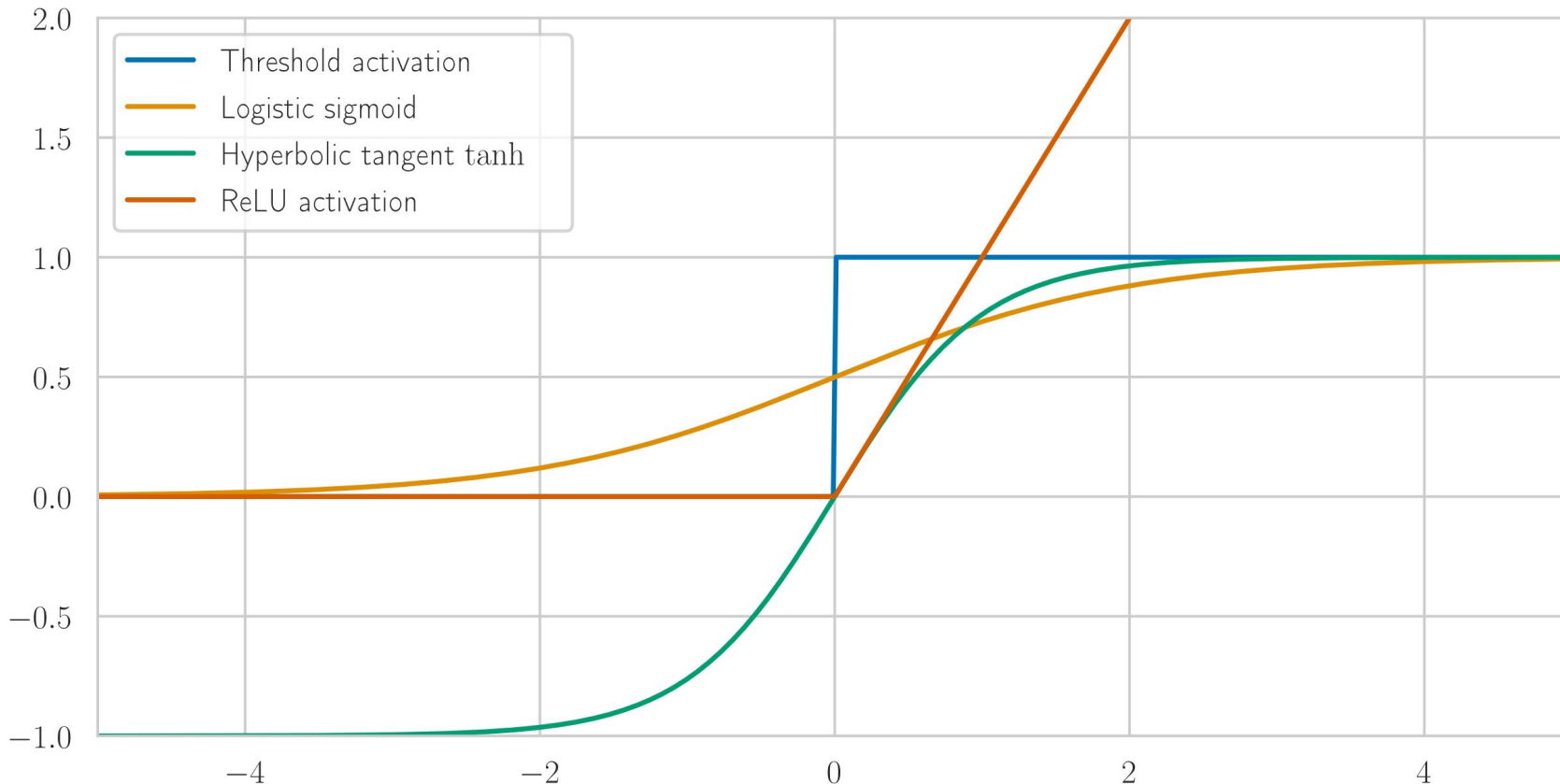


FIG. 2 — Organization of a perceptron.



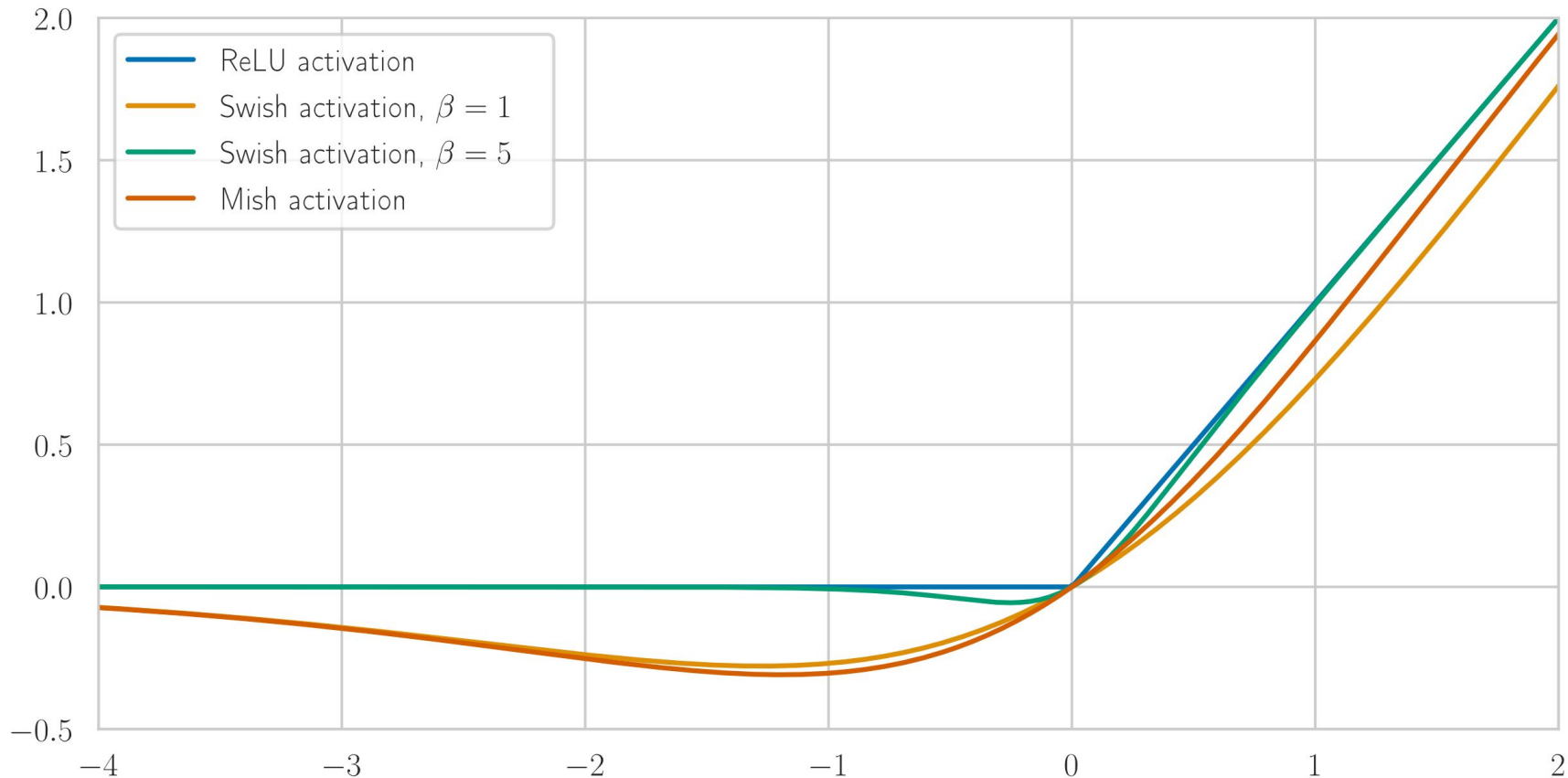
Функции активации

- В наше время появились и другие виды связей (например, механизмы внимания), но в основном сети состоят из таких же искусственных нейронов; только функции активации меняются



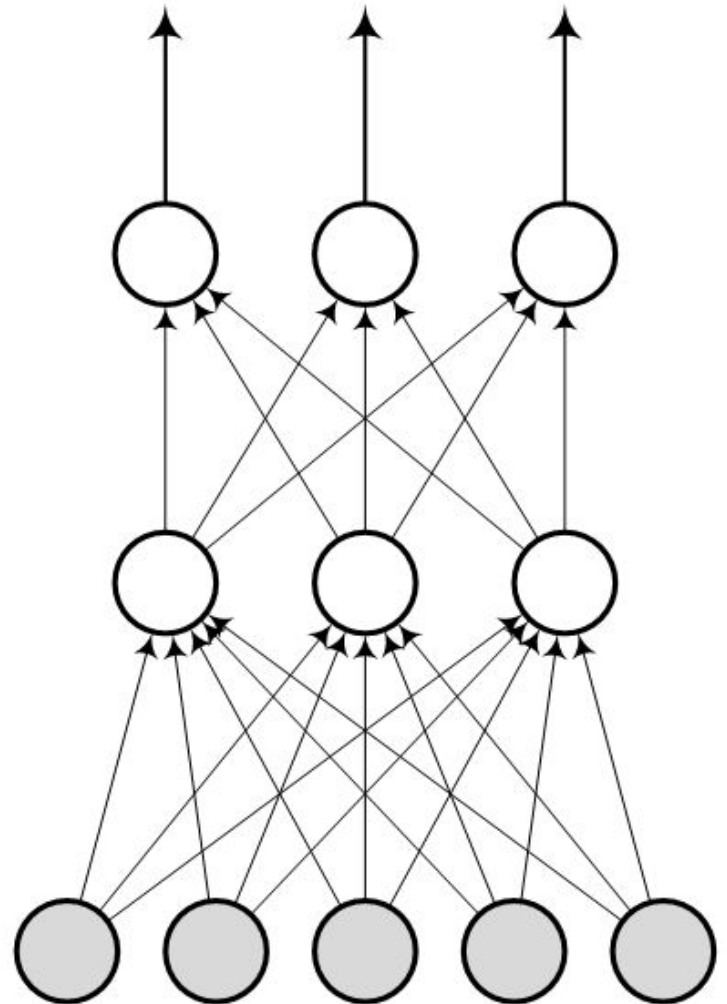
Функции активации

- В наше время появились и другие виды связей (например, механизмы внимания), но в основном сети состоят из таких же искусственных нейронов; только функции активации меняются



Коннекционизм

- Нейроны соединяются в сети
- Принцип коннекционизма: большая композиция простых объектов обретает новые свойства
- Нейросеть – это граф вычислений, узлы которого – элементарные функции
- Часто такие графы делятся на слои (layers); справа – пример простой сети с двумя слоями



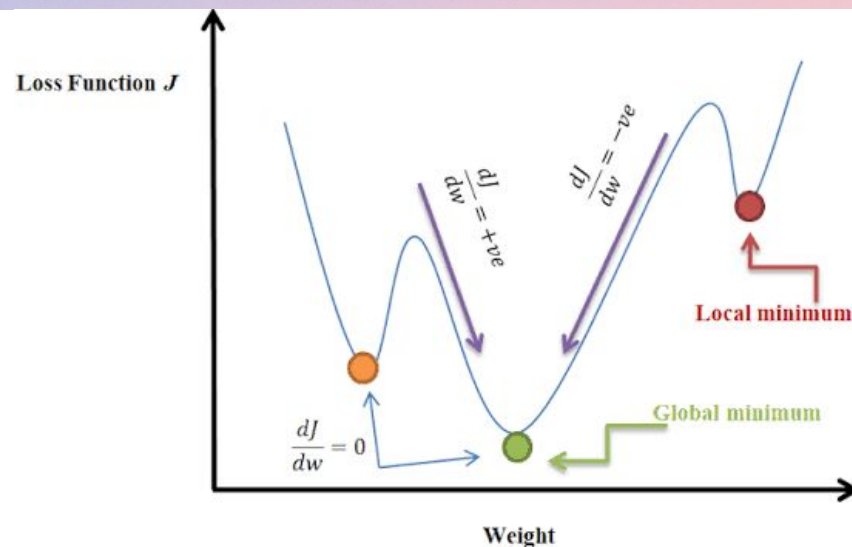
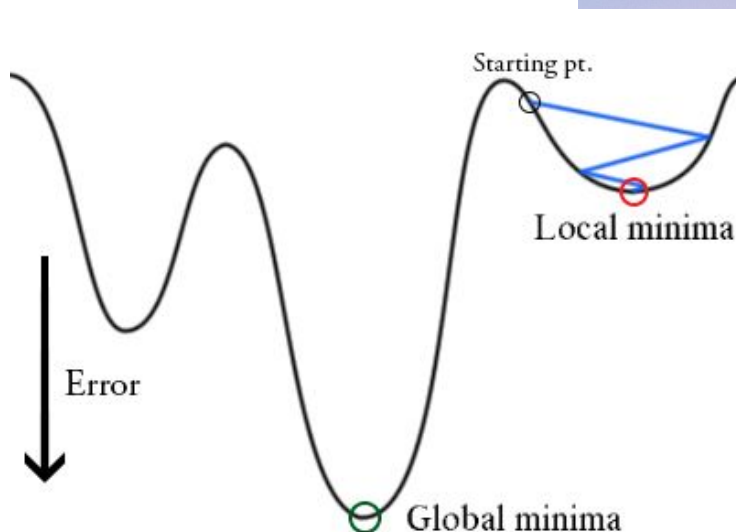
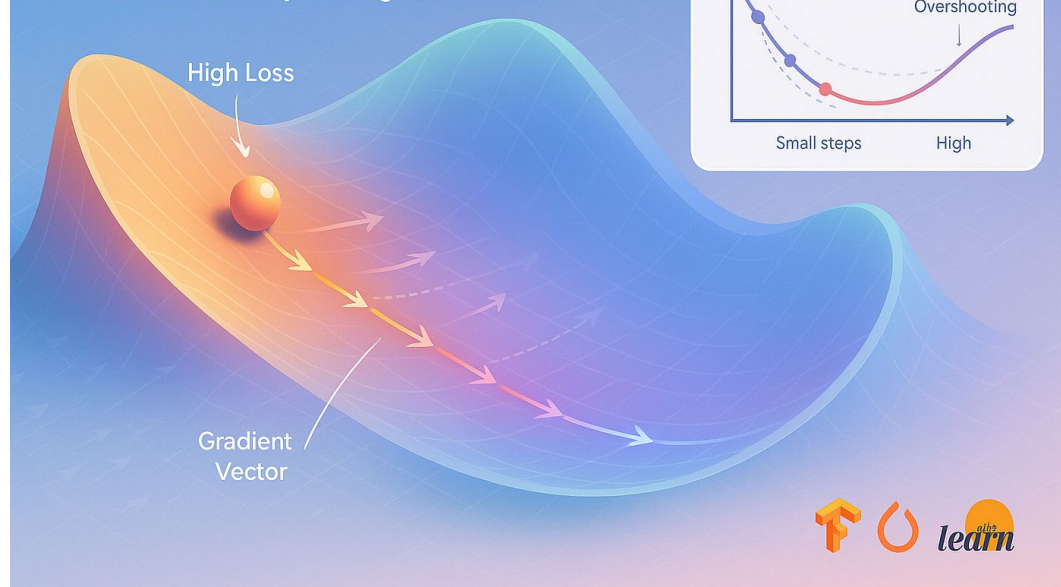
Градиентный спуск

- Самый главный метод оптимизации: нужно спускаться в сторону, обратную градиенту

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha \nabla f(\mathbf{x}_k).$$

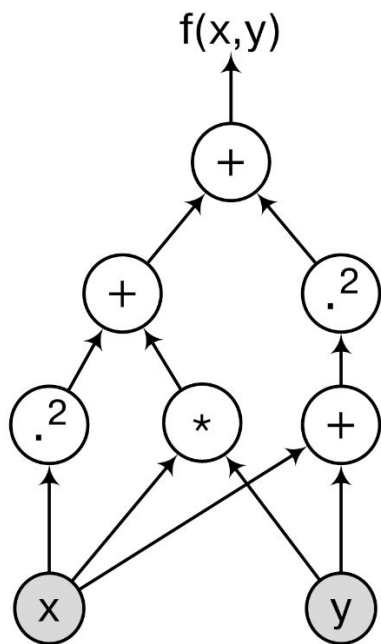
Gradient Descent:

How Models Learn by Rolling Downhill

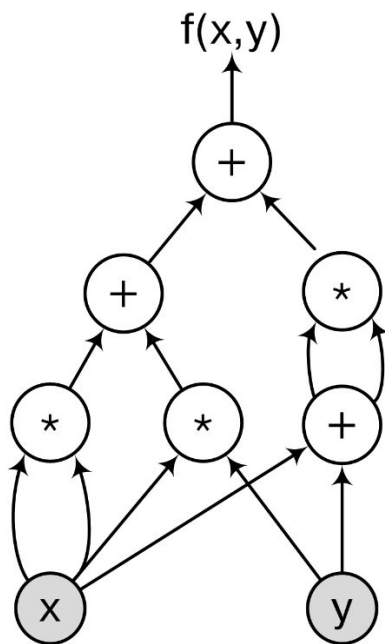


Обратное распространение ошибки

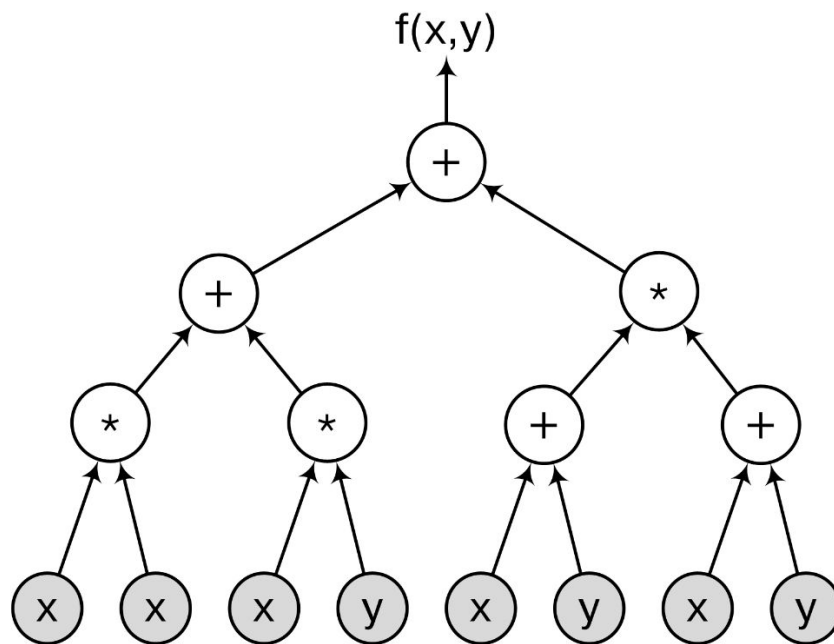
- Но как подсчитать все эти производные?..
- Рассмотрим пример:



(a)



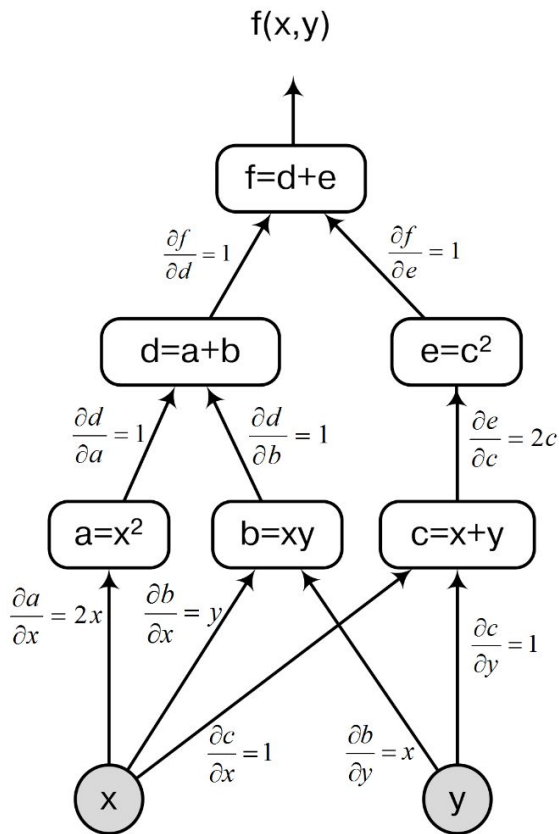
(б)



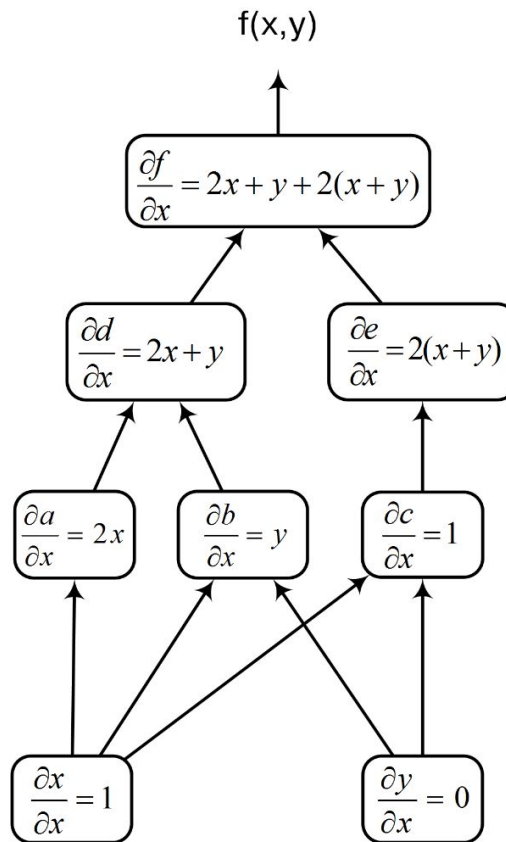
(B)

Обратное распространение ошибки

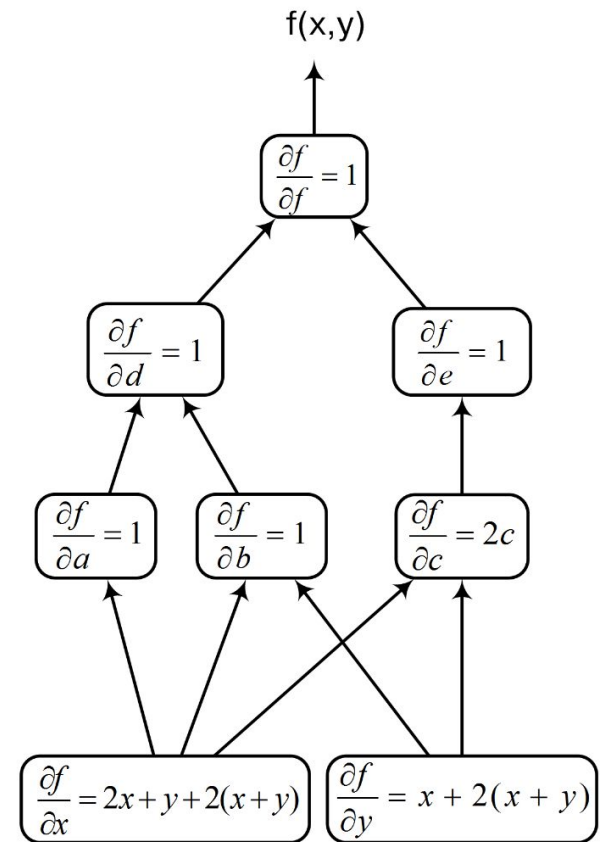
- Но как подсчитать все эти производные?..
- Рассмотрим пример:



(a)



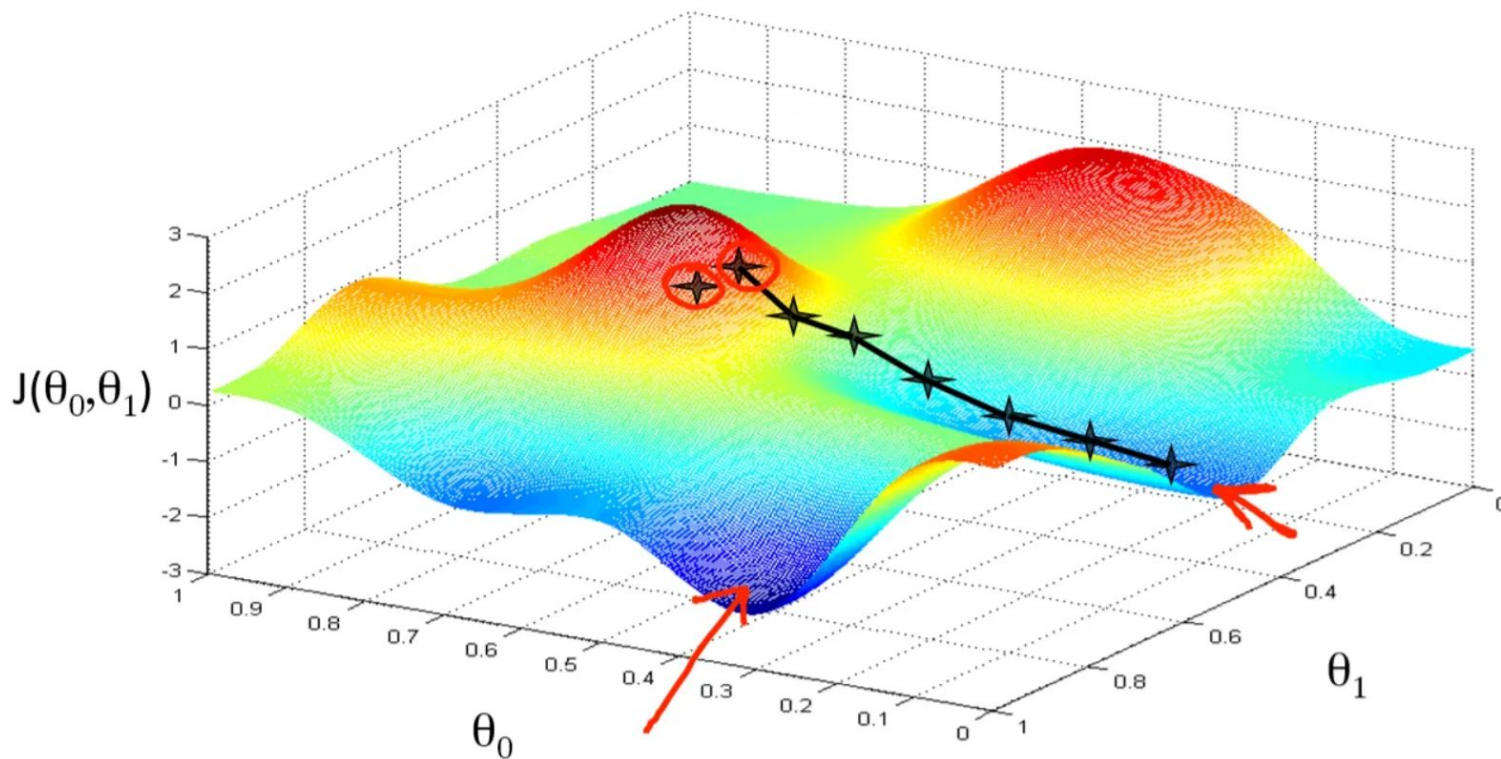
(б)



(B)

Градиентный спуск

- А потом происходит градиентный спуск. Точнее...

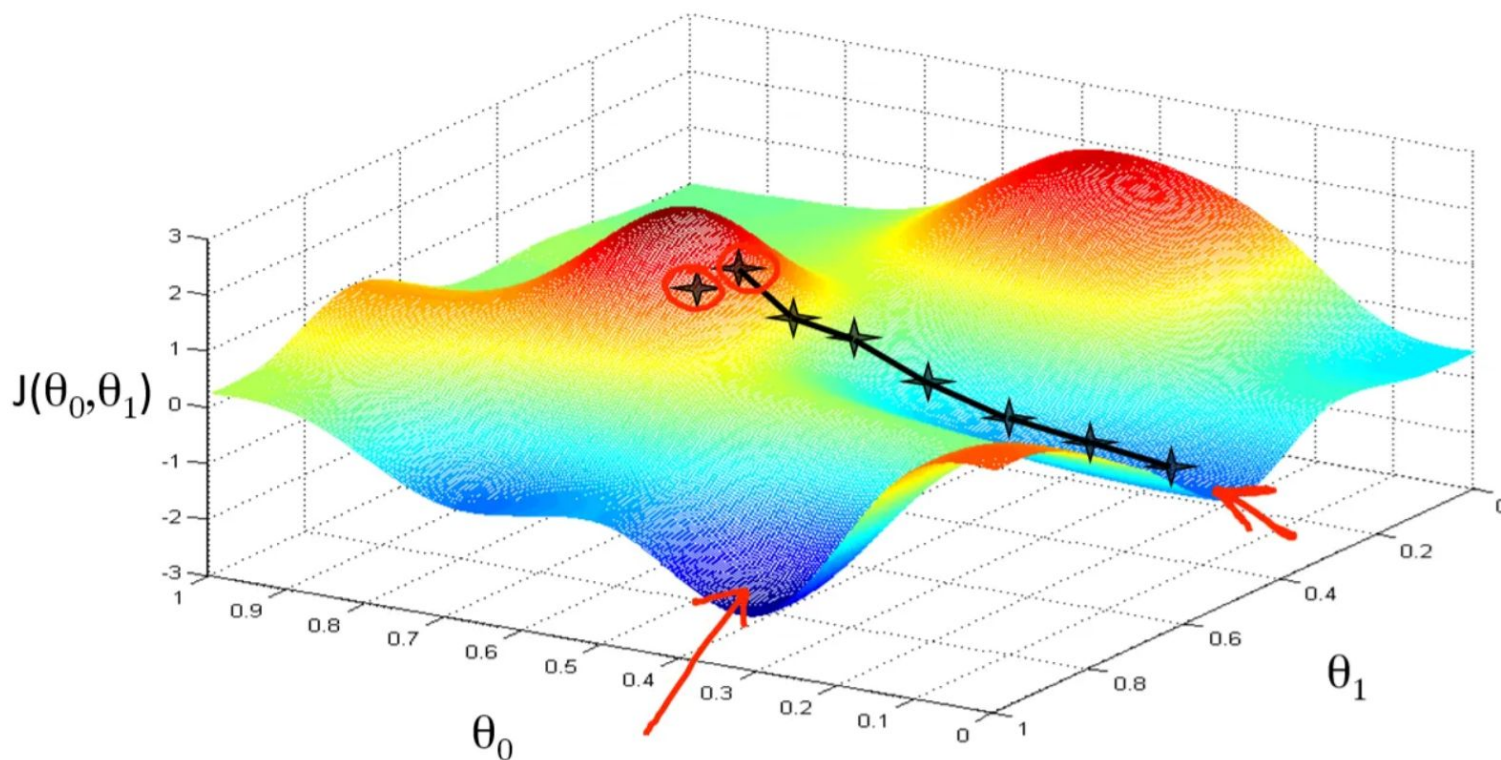


Алгоритмы стохастического градиентного спуска



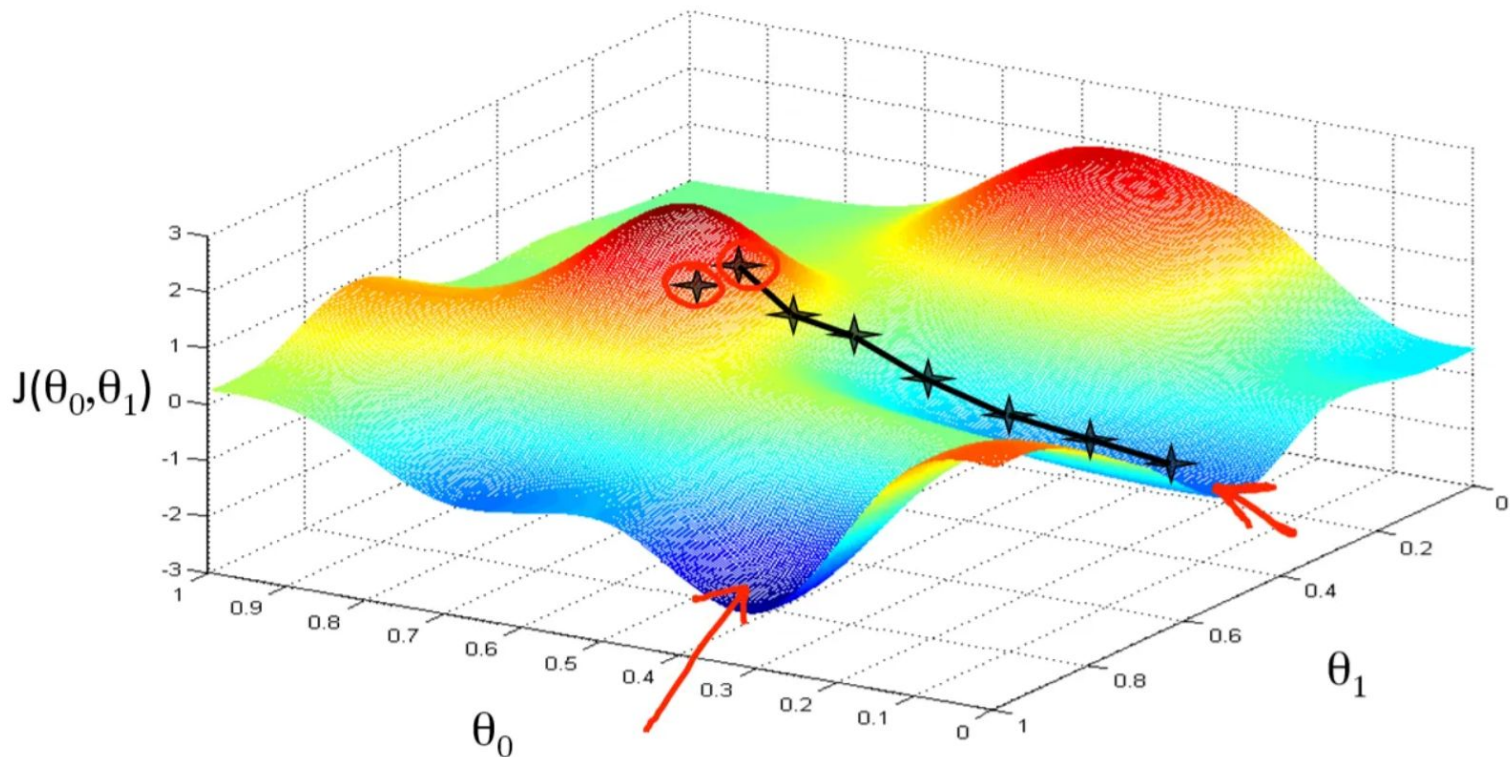
Градиентный спуск

- Мы уже обсудили, как устроены нейросети, как происходит обратное распространение ошибки, то есть как можно считать градиенты
- А потом происходит градиентный спуск, о котором мы уже говорили. Точнее...



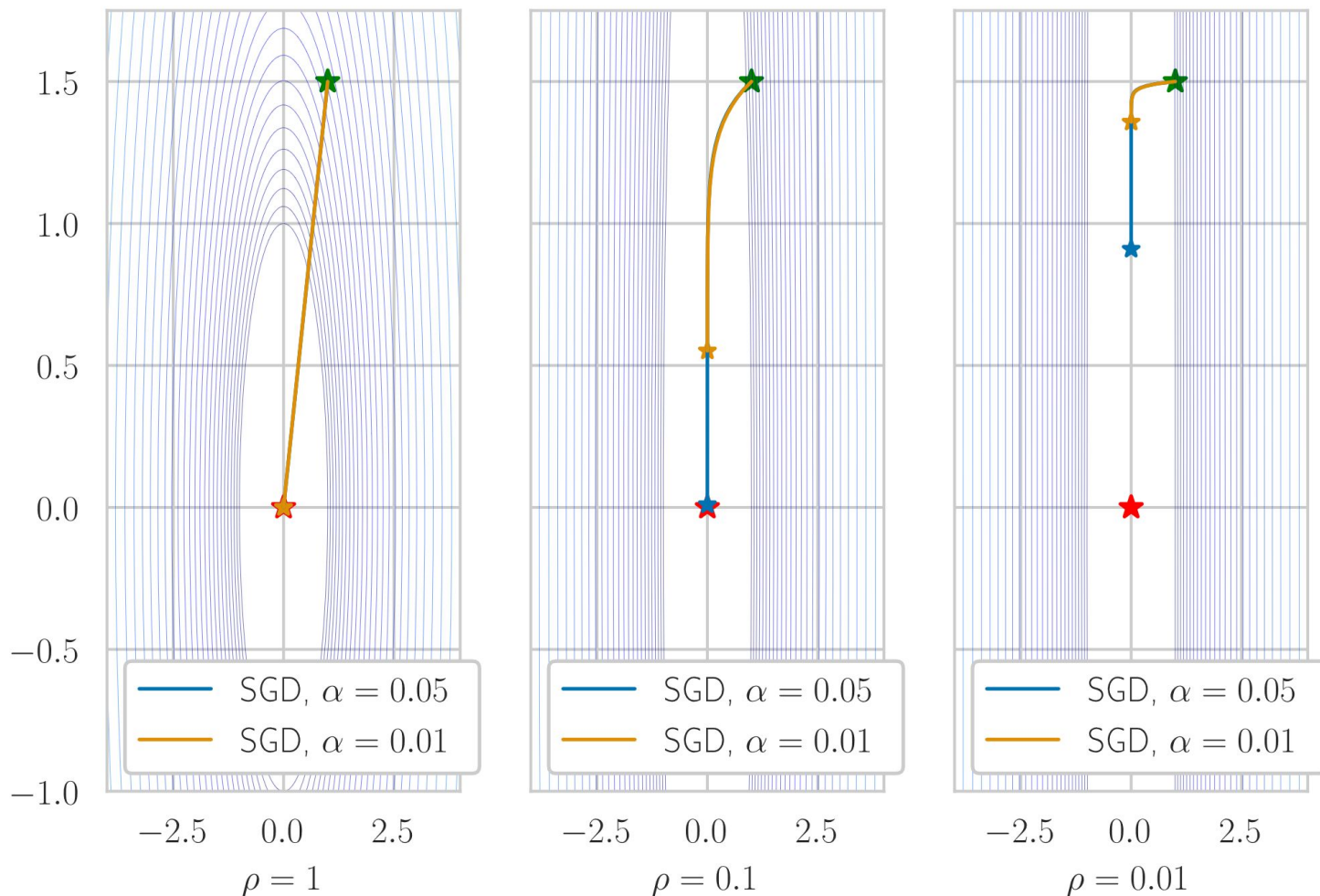
Градиентный спуск

- Мы уже обсудили, как устроены нейросети, как происходит обратное распространение ошибки, то есть как можно считать градиенты
- А потом происходит градиентный спуск, о котором мы уже говорили. Точнее, *стохастический градиентный спуск* (stochastic gradient descent, SGD), потому что вычислить всю функцию ошибки было бы слишком дорого



Градиентный спуск

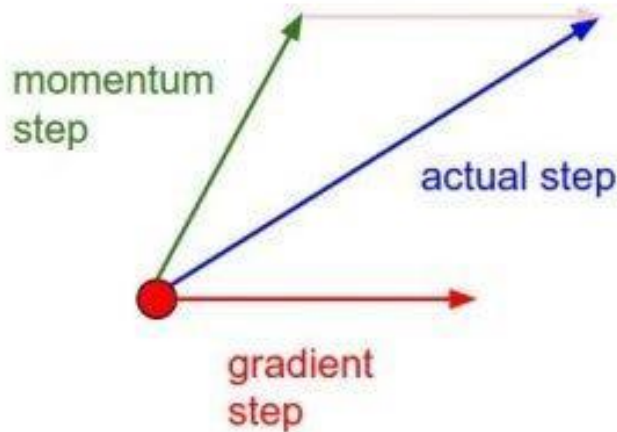
- Главные проблемы начинаются в высоких размерностях, при разном масштабе переменных:



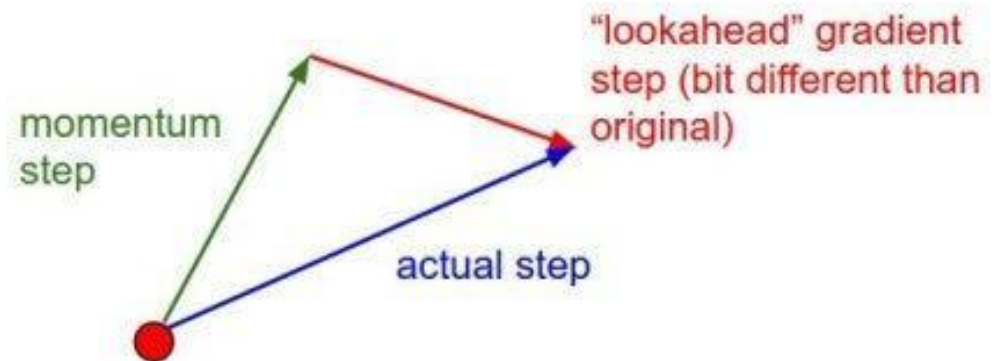
Варианты градиентного спуска

- Варианты градиентного спуска — *метод моментов*
- И *моменты Нестерова*
- Но это только начало, в современном DL много адаптивных алгоритмов...

Momentum update



Nesterov momentum update



Варианты градиентного спуска

- От Adagrad...

$$\mathbf{w}_{t+1,i} = \mathbf{w}_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

- ...через RMSprop и Adadelta до Adam...

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \\ u_t &= \frac{\eta}{\sqrt{v + \epsilon}} m_t. \end{aligned}$$

- ...а сейчас ещё Muon, Lion, и на этом
этом
наверняка
прогресс
ещё не
закончился

```
def train(w, g, m, v, lr):  
    g = clip(g, lr)  
    g = arcsin(g)  
    m = interp(g, v, 0.899)  
    m2 = m * m  
    v = interp(g, m, 1.109)  
    abs_m = sqrt(m2)  
    update = m / abs_m  
    wd = w * 0.4602  
    update = update + wd  
    lr = lr * 0.0002  
    m = cosh(update)  
    update = update * lr  
    return update, m, v
```

Algorithm 2 Muon

Require: Learning rate η , momentum μ

- 1: Initialize $B_0 \leftarrow 0$
 - 2: **for** $t = 1, \dots$ **do**
 - 3: Compute gradient $G_t \leftarrow \nabla_{\theta} \mathcal{L}_t(\theta_{t-1})$
 - 4: $B_t \leftarrow \mu B_{t-1} + G_t$
 - 5: $O_t \leftarrow \text{NewtonSchulz5}(B_t)$
 - 6: Update parameters $\theta_t \leftarrow \theta_{t-1} - \eta O_t$
 - 7: **end for**
 - 8: **return** θ_t
-

Спасибо за
внимание!



@SINECOR

