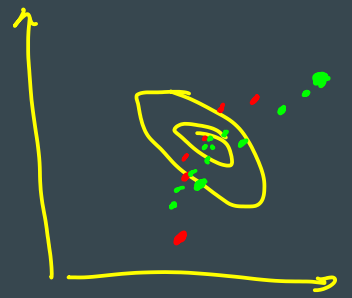
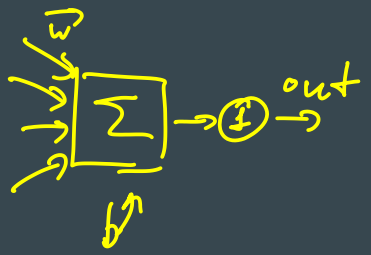
Лекция 3



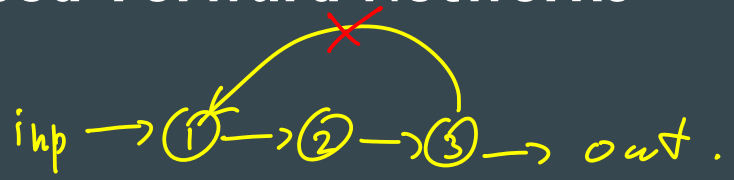
ExpressML

Повторение

Математическая модель нейрона



Feed-Forward Networks

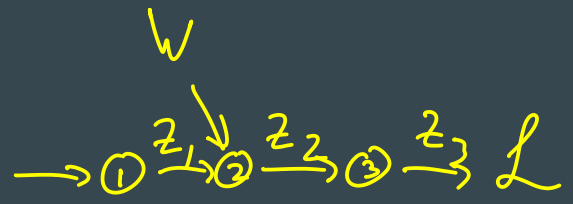


Функция потерь

$L \leftarrow$ распред., берётся из правдоподобия

Chain Rule, Обратное распространение ошибки

$f(g(x))$ $\quad \frac{\partial f(g(x))}{\partial x} = \frac{\partial f}{\partial g} \frac{\partial g}{\partial x} \cdot \ldots \cdot$



$\frac{\partial L}{\partial z_3} \cdot \frac{\partial z_3}{\partial z_2} \cdot \frac{\partial z_2}{\partial w} = \frac{\partial L}{\partial w}$

Градиентный спуск

$$\left\| \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \right\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

$$x^{t+1} = x - \alpha \cdot \nabla L$$

Нестер:

$$v^{t+1} = v \cdot \gamma + (1-\gamma) \cdot \nabla f \qquad \gamma - \text{momentum}$$

$$x^{t+1} = x - \alpha \cdot v^{t+1}$$

Adam:

$$\left.\begin{aligned} m^{t+1} &= m \cdot \beta_2 + (1-\beta_2) \cdot \|\nabla f\|_2 \\ v^{t+1} &= v\beta_1 + (1-\beta_1) \nabla f \end{aligned}\right\} \text{EMA}$$

$$x^{t+1} = x - \alpha \cdot \frac{v^{t+1}}{m^{t+1}}$$

RProp. (поэлементно)

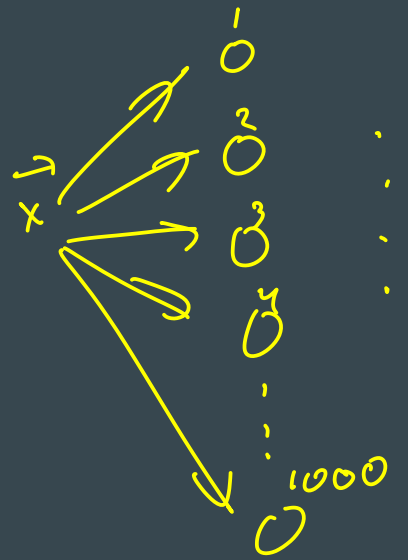
$$x^{t+1} = x - \alpha \cdot \nabla f$$

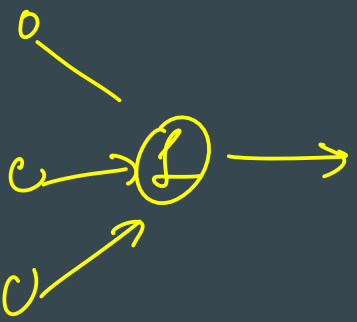
$$\alpha_i = \begin{cases} \alpha_i \cdot a & \nabla f_i \cdot \nabla f_i^{-1} < 0 \\ \alpha_i \cdot 1.2 & \nabla f_i \nabla f_i^{-1} > 0 \end{cases} \quad \alpha_i > 0$$

Нестеровский градиентный спуск

Проблемы сигмоидных нейронных сетей

Проблема инициализации весов

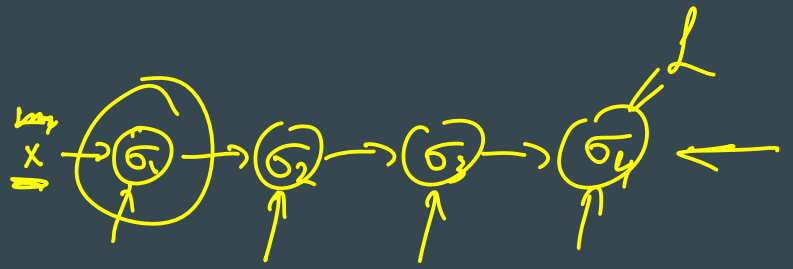




Все один инициализировал одинаковая динамика

Xavier – эмпирич.

Проблема затухания градиента (сигмоида)



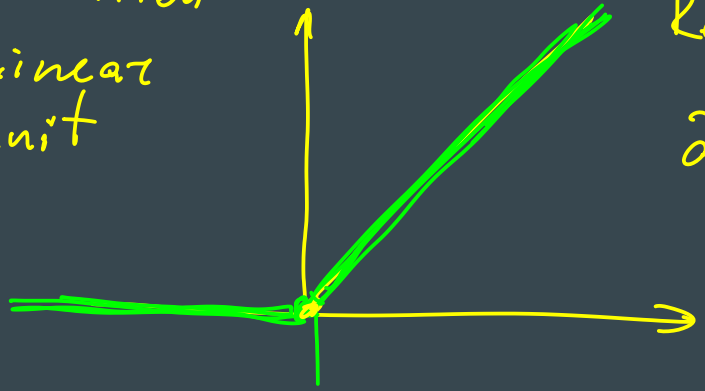
$$\frac{\partial L}{\partial x} = \sigma_4' \cdot \sigma_3' \cdot \sigma_2' \cdot \sigma_1' =$$

$$= \sigma_4(1-\sigma_4) \cdot \sigma_3(1-\sigma_3) \cdot \sigma_2(1-\sigma_2) \cdot \sigma_1(1-\sigma_1)$$

$\sigma \in [0;1]$ $\quad max: \left(\frac{1}{4}\right)^n$

ReLU, ELU — активации

Rectified Linear Unit

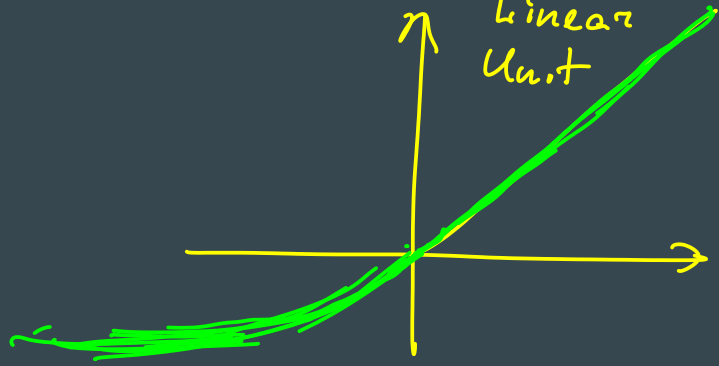


$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$$

∂

$\mathcal{L}(ReLU)$

Exponential Linear Unit



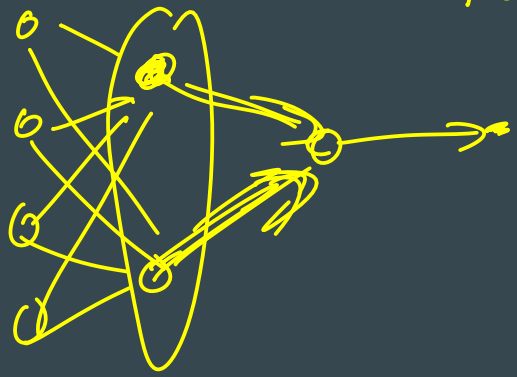
$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ e^{+x} + c & \text{if } x < 0 \end{cases} \qquad c = -1$$

Проблема переобучения

~10к параметров

1) Уменьшать # параметров

2) Drop-Out — регуляризация

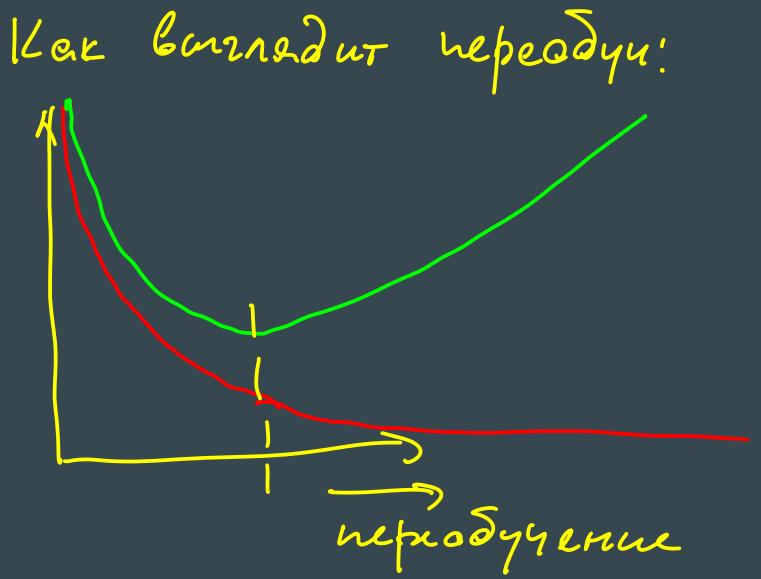


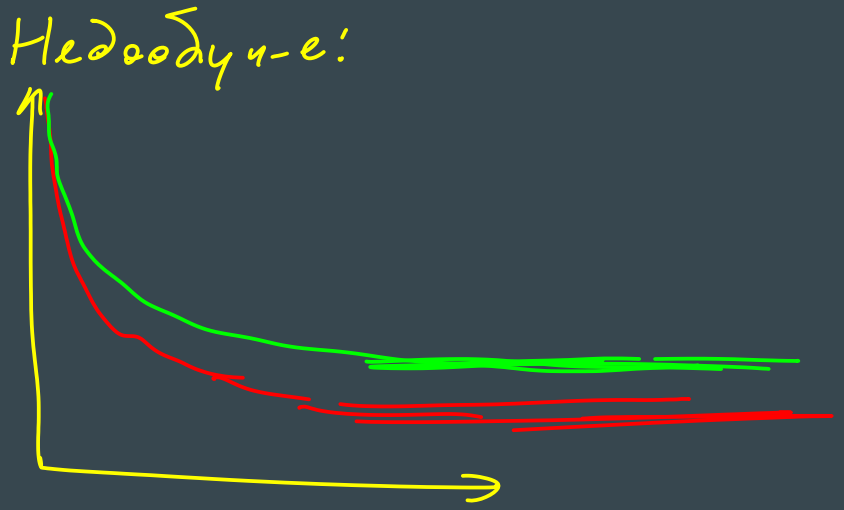
L-2

Drop-Out $p = 0.5$

3) $L + \left| L \sum_{i=1}^{N} w_i^2 \right|$ Weight-Decay $\left(\sum_{i=1}^{N} |w_i| \; L-1 \right)$

Регуляризация

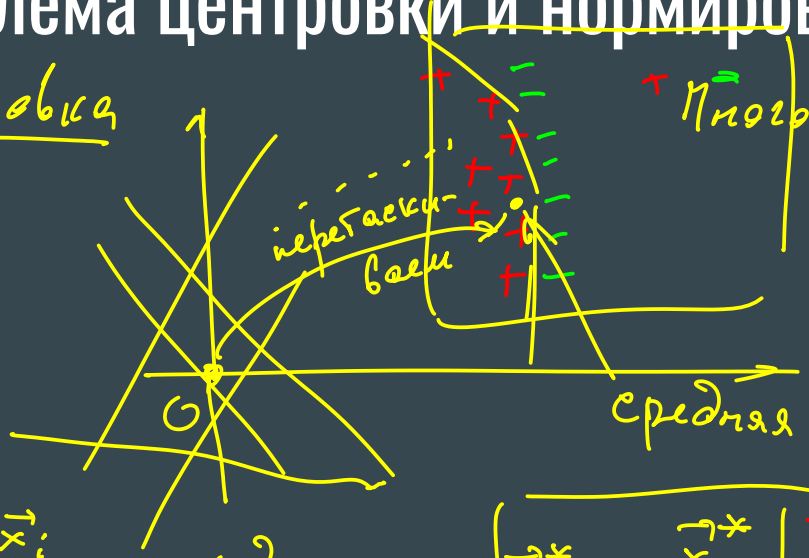




Проблема центровки и нормировки

Данные надо центрировать

Центровка



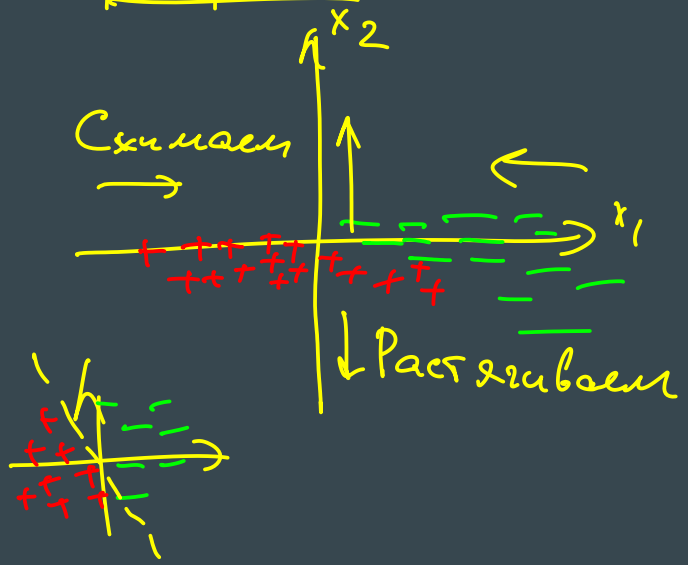
$$\vec{\mu} = \frac{\sum \vec{x}_i}{N} \quad \text{— среднее}$$

$$\vec{x}^*_{ij} = \vec{x}_{ij} - \vec{\mu} \quad \text{— центровка}$$

$$\sigma^2_i = \frac{(x^*_{ij})^2}{N-1} \quad \text{— координата}$$

$$\vec{x}^*_{n.y.} = \frac{\vec{x}^*_{ij}}{\vec{\sigma}}$$

Нормировка



Батч-нормализация

$\rightarrow \mu, \sigma$ вычисляем

$\tilde{\mu}^{t+1} = \gamma\tilde{\mu} + (1-\gamma)\mu^t$

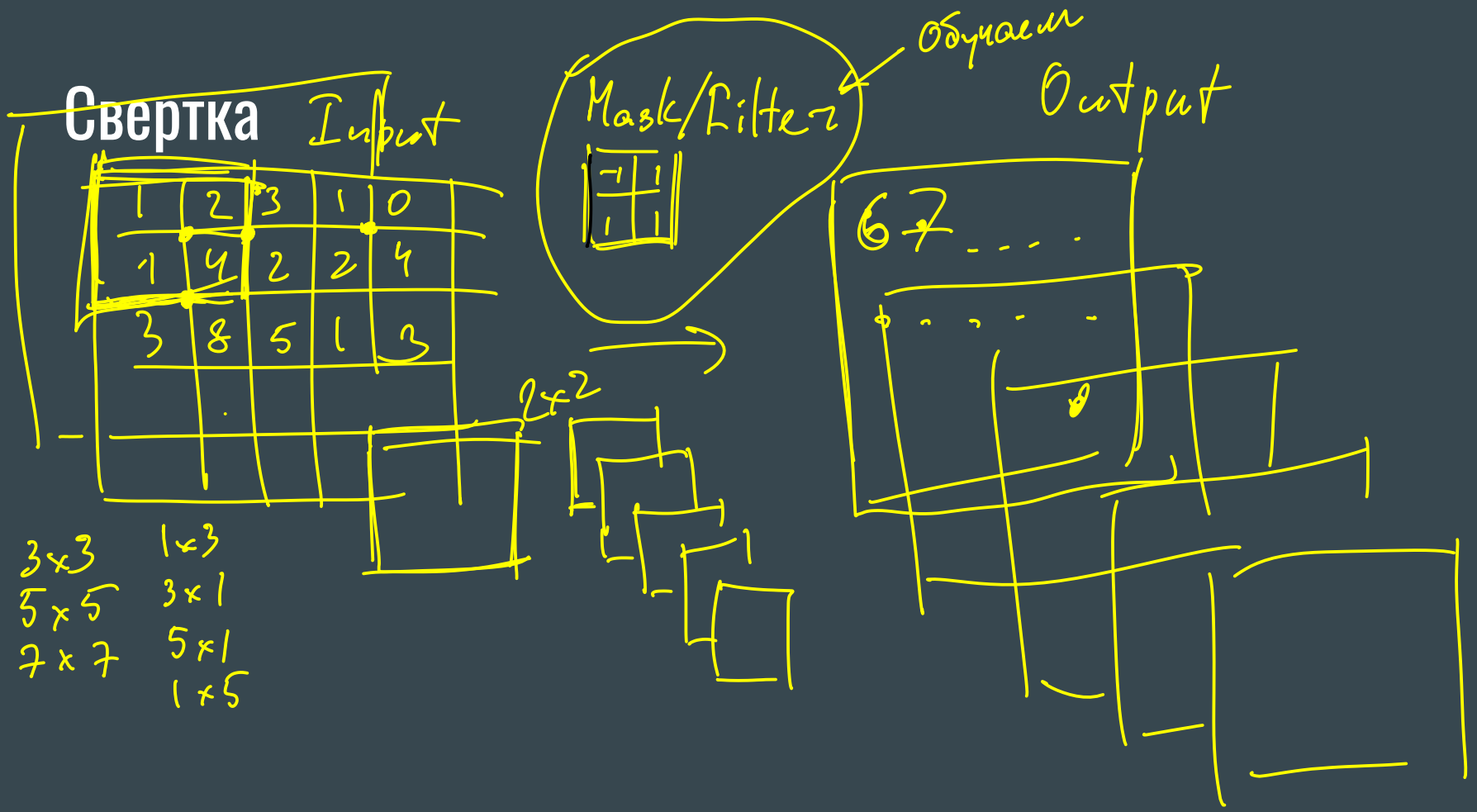
$\tilde{\sigma}^{t+1} = \gamma\tilde{\sigma} + (1-\gamma)\sigma$

$$\vec{x}^* = \frac{\vec{x} - \tilde{\vec{\mu}}}{\tilde{\vec{\sigma}}} \cdot C_1 + C_2$$

обучаемые параметры

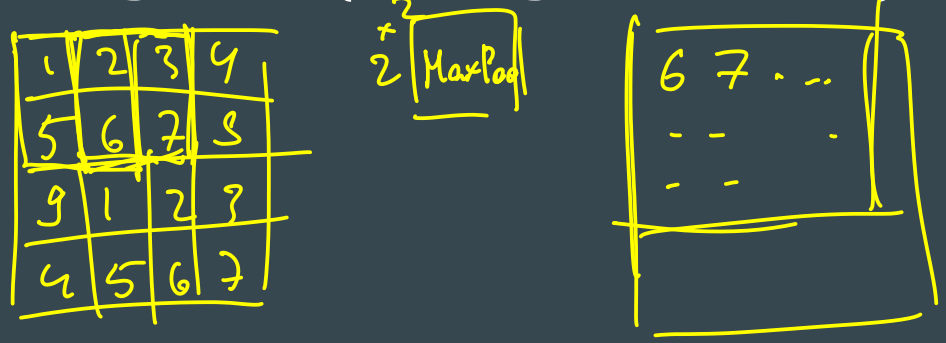
Свертночные Нейронные Сети

Свертка


Input
Mask/Filter
обучаем
Output
1 2 3 1 0
1 4 2 2 4
3 8 5 1 3
2x2
67 ...
3x3
5x5
7x7
1x3
3x1
5x1
1x5

Сверточный слой

Pooling слой (Average, Max, Min)



LeNet архитектура (1998)

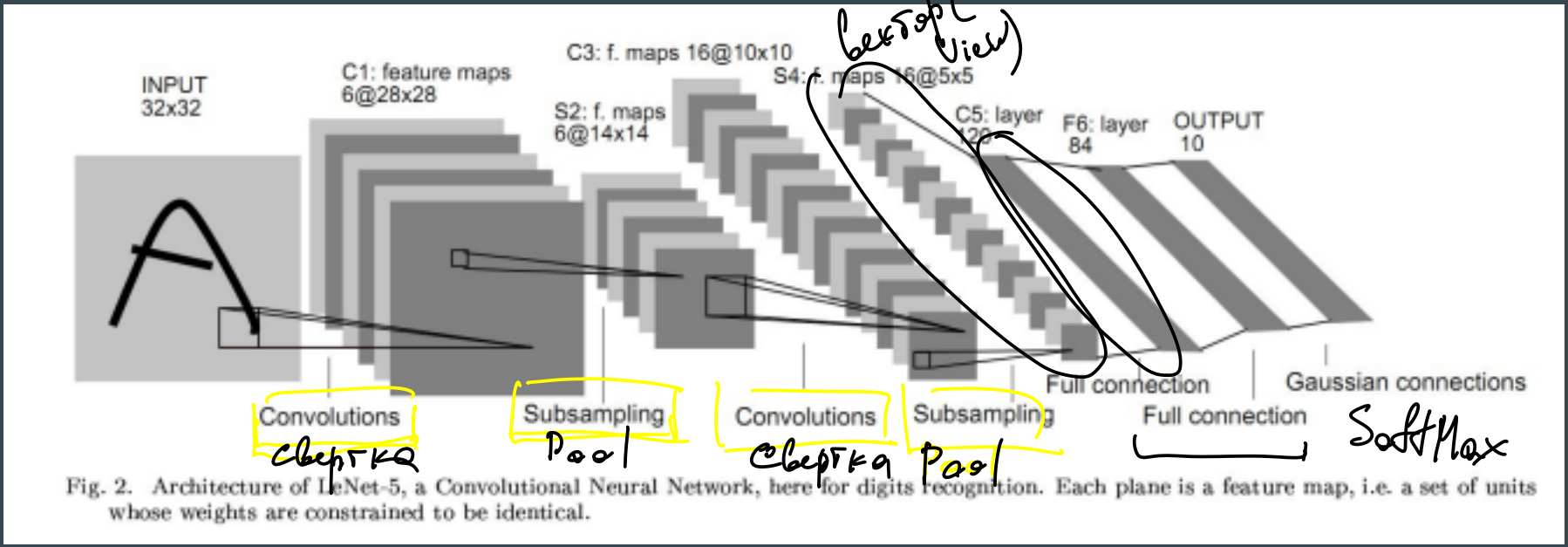


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

AlexNet архитектура