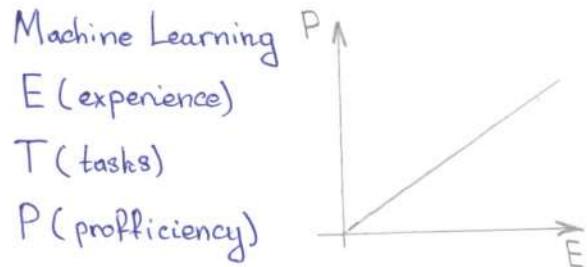


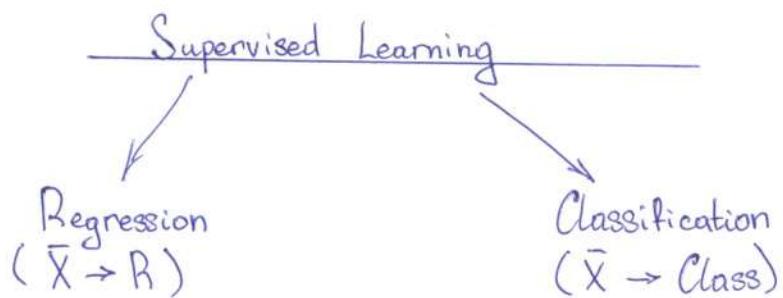
Machine Learning (ML)

- Dynamische Reaktionen (Cognitive Science, Computational Neurobiology)
- Dynamische Psychoanalyse (nicht wie Psychoanalyse)
- Reichenbachsche Reaktion (Turing Test "Imitation game" (1950)) 
- Reichenbachsche Psychoanalyse - omnipoetische



Statistical approaches : estimation - learning

- Supervised Learning : Regression ($y(\bar{x}) \rightarrow R^c$), Classification ($y(\bar{x}) \rightarrow \vec{p}_{\text{class}}$)
- Unsupervised Learning : Clusterization
- Reinforcement Learning : Agent \rightleftarrows Environment



Data

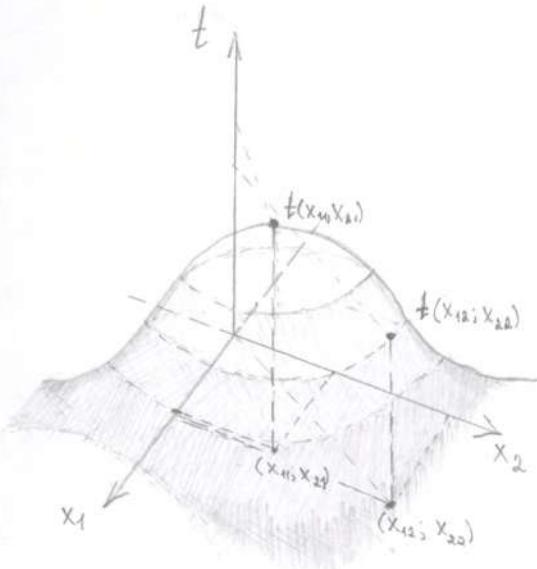
Object features - $\bar{x} \in \mathbb{R}^D$

Training sample

$$\begin{array}{l} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \vdots \\ \bar{x}_n \end{array}$$

Target values

$$\begin{array}{l} \bar{t}_1 \\ \bar{t}_2 \\ \bar{t}_3 \\ \vdots \\ \bar{t}_N \end{array}$$



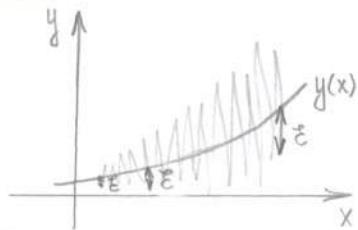
Notation
y - target
\hat{y} - predicted

Linear Regression

$$t(\bar{x}) = y(\bar{x}, \vec{w}) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

$\sigma \neq P(\bar{x}) \Rightarrow$
const.

$$y(\bar{x}, \vec{w}) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_D x_D = w_0 + \sum_{j=1}^D w_j x_j = \vec{w}^\top \bar{x}, \quad x_0 = 1$$



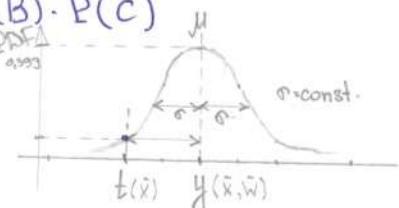
Likelihood

Общая условная плотность вероятности P при X, \vec{w}, σ^2

Events are Independent : $P(A \cap B \cap C) = P(A) \cdot P(B) \cdot P(C)$

$$P(\vec{t} | X, \vec{w}, \sigma^2) = \prod_{n=1}^N N(t_n | y(\bar{x}_n, \vec{w}), \sigma^2) \xrightarrow{x} \max$$

$$P(\vec{t} | X, \vec{w}, \sigma^2) = \prod_{n=1}^N \frac{1}{\sigma \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \left(\frac{t_n - y(\bar{x}_n, \vec{w})}{\sigma} \right)^2} \xrightarrow{x} \max$$

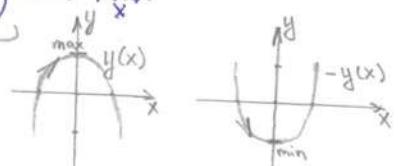


$$\ln P(\vec{t} | X, \vec{w}, \sigma^2) = \ln \prod_{n=1}^N \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{t_n - y(\bar{x}_n, \vec{w})}{\sigma} \right)^2} \xrightarrow{x} \max$$

$$= \sum_{n=1}^N \ln \frac{1}{\sigma \sqrt{2\pi}} + \ln \sigma$$

$P(x) \rightarrow \max$
 $\operatorname{argmax}_x P(x) = \operatorname{argmax}_x \log(P(x))$

$$\begin{aligned} \ln P(\vec{t} | X, \vec{w}, \sigma^2) &= \sum_{n=1}^N \ln \frac{1}{\sigma \sqrt{2\pi}} + \sum_{n=1}^N \ln e^{-\frac{1}{2} \left(\frac{t_n - y(\bar{x}_n, \vec{w})}{\sigma} \right)^2} = \\ &= \sum_{n=1}^N \ln \frac{1}{\sigma \sqrt{2\pi}} + \sum_{n=1}^N -\frac{1}{2} \left(\frac{t_n - y(\bar{x}_n, \vec{w})}{\sigma} \right)^2 = \\ &= \sum_{n=1}^N \ln \frac{1}{\sigma \sqrt{2\pi}} + \frac{1}{\sigma^2} \underbrace{\left(\sum_{n=1}^N \frac{1}{2} (t_n - y(\bar{x}_n, \vec{w}))^2 \right)}_{\text{Loss Function}} \xrightarrow{x} \min \end{aligned}$$



Loss Function

$$E(\vec{w}) = \sum_{n=1}^N \frac{1}{2} (t_n - y(\bar{x}_n, \vec{w}))^2 \xrightarrow{x} \min - \text{Mean Squared Error (MSE)}$$

Optimization problem

Adding Non-linearity to Linear Regression

$$\hat{t}_i(\vec{x}_i) = y(\vec{x}_i, \vec{w}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$y(\vec{x}_i, \vec{w}) = \vec{w}^T \vec{x} = \sum_j^D x_j w_j = w_0 + x_1 w_1 + x_2 w_2 + \dots + x_D w_D$$

$\vec{x}_i \in \mathbb{R}^D$ - sample

$$\vec{y}(X, \vec{w}) = X^{N \times D} \cdot \vec{w}^{D \times 1} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vec{x}_1 & \vec{x}_2 & \vec{x}_3 & \dots & \vec{x}_N \end{bmatrix}^{N \times 1} \cdot \begin{bmatrix} w_0 & w_1 & w_2 & \dots & w_D \end{bmatrix}^{1 \times D}$$

rows - samples
cols - features

Basis Function

$$\vec{\phi}(\vec{x}_i) = [\phi_0(\vec{x}_i) \ \phi_1(\vec{x}_i) \ \phi_2(\vec{x}_i) \ \dots \ \phi_{M-1}(\vec{x}_i)]^T$$

$$\vec{\phi} \in \mathbb{R}^M$$

$$y(\vec{x}_i, \vec{w}) = \vec{w}^T \cdot \vec{\phi}(\vec{x}_i) = w_0 + \sum_{j=1}^{M-1} w_j \cdot \phi_j(\vec{x}_i), \quad \phi_0(\vec{x}) = 1$$

Polynomial Basis Functions:

$$\underbrace{[x_1 \ x_2 \ \dots \ x_D]}_{\vec{\phi}_1}, \underbrace{[x_1^2 \ x_2^2 \ \dots \ x_D^2]}_{\vec{\phi}_2}, \underbrace{[x_1^3 \ x_2^3 \ \dots \ x_D^3]}_{\vec{\phi}_3}$$

Design Matrix

$$\Phi(X) = [\phi^T(\vec{x}_1) \ \phi^T(\vec{x}_2) \ \dots \ \phi^T(\vec{x}_N)]^T$$

$$\Phi(X) = \begin{bmatrix} \phi_0(\vec{x}_1) & \phi_1(\vec{x}_1) & \dots & \phi_{M-1}(\vec{x}_1) \\ \phi_0(\vec{x}_2) & \phi_1(\vec{x}_2) & \dots & \phi_{M-1}(\vec{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\vec{x}_N) & \phi_1(\vec{x}_N) & \dots & \phi_{M-1}(\vec{x}_N) \end{bmatrix}^{N \times M}$$

$$\vec{y}(X, \vec{w}) = \Phi(X)^{N \times M} \cdot \vec{w}^{M \times 1} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \vec{y}_1 & \vec{y}_2 & \vec{y}_3 & \dots & \vec{y}_N \end{bmatrix}^{N \times 1}$$

\vec{y} нелинейна относительно X .

Linear Regression Optimization.

Normal Equation. Analytical Solution.

$$t(\bar{x}) = y(\bar{x}, \bar{w}) + \varepsilon; \quad \varepsilon \sim N(0, \sigma^2)$$

$\Phi \neq P(\bar{x})$

$$y(\bar{x}, \bar{w}) = \bar{w}^\top \Phi(\bar{x}) = \sum_{i=0}^{M-1} w_i \phi_i(\bar{x})$$

$$P(\vec{t} | X, \bar{w}, \sigma^2) = \prod_{n=1}^N N(t_n | y(\bar{x}_n, \bar{w}), \sigma^2) \rightarrow \max_w$$

$$\bar{x} \in \mathbb{R}^D, \quad X \in \mathbb{R}^{N \times D},$$

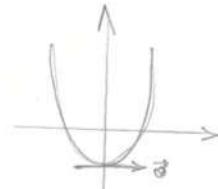
$$\Phi(X) \in \mathbb{R}^{N \times M}$$

Loss function

$$E(\bar{w}) = \sum_{n=1}^N \frac{1}{2} (t_n - y(\bar{x}_n, \bar{w}))^2 \rightarrow \min_w$$

In local minima gradient is equal to zero vector:

$$\nabla E(\bar{w}) = \vec{0}$$



Gradient is vector of partial derivatives: $\nabla y(\bar{x}) = \frac{dy}{dx_1} \vec{i} + \frac{dy}{dx_2} \vec{j} + \frac{dy}{dx_3} \vec{k}$

$$\frac{dE(\bar{w})}{d w_i} = \left(\sum_{n=1}^N \frac{1}{2} (t_n - y(\bar{x}_n, \bar{w}))^2 \right)' = \sum_{n=1}^N 2 \cdot \frac{1}{2} (t_n - y(\bar{x}_n, \bar{w})) \cdot (t_n - y(\bar{x}_n, \bar{w}))'$$

$$\frac{dE(\bar{w})}{d w_i} = -\sum_{n=1}^N (t_n - y(\bar{x}_n, \bar{w})) \cdot \phi_i(\bar{x}_n) \quad \text{- scalar}$$

$$\nabla E(\bar{w}) = -\sum_n (t_n - y(\bar{x}_n, \bar{w})) \cdot \vec{\Phi}(\bar{x}_n) = \left[\frac{dE(\bar{w})}{d w_0} \quad \frac{dE(\bar{w})}{d w_1} \quad \frac{dE(\bar{w})}{d w_2} \dots \frac{dE(\bar{w})}{d w_{M-1}} \right]^\top$$

vector of partial derivatives

$$\vec{0}^\top = -\sum_{n=1}^N (t_n - y(\bar{x}_n, \bar{w})) \cdot \vec{\Phi}^\top(\bar{x}_n) =$$

$$= -\sum_{n=1}^N t_n \vec{\Phi}^\top(\bar{x}_n) + \sum_{n=1}^N y(\bar{x}_n, \bar{w}) \vec{\Phi}^\top(\bar{x}_n) = -\sum_{n=1}^N t_n \vec{\Phi}^\top(\bar{x}_n) + \sum_{n=1}^N (y) \bar{w}^\top \vec{\Phi}(\bar{x}_n) \vec{\Phi}^\top(\bar{x}_n) =$$

$$= -\vec{t}^\top \vec{\Phi} + \bar{w}^\top \vec{\Phi}^\top \vec{\Phi} = \vec{0}^\top$$

$$\bar{w}^\top \vec{\Phi}^\top \vec{\Phi} = \vec{t}^\top \vec{\Phi}$$

$$\bar{w}^\top = \vec{t}^\top \vec{\Phi} (\vec{\Phi}^\top \vec{\Phi})^{-1}$$

$$\bar{w} = (\vec{t}^\top \vec{\Phi} (\vec{\Phi}^\top \vec{\Phi})^{-1})^\top = ((\vec{\Phi}^\top \vec{\Phi})^{-1})^\top \cdot (\vec{t}^\top \vec{\Phi})^\top = ((\vec{\Phi}^\top \vec{\Phi})^\top)^{-1} \cdot (\vec{\Phi}^\top \vec{t})$$

$$\bar{w} = (\vec{\Phi}^\top \vec{\Phi})^{-1} \vec{\Phi}^\top \vec{t}$$

Moore-Penrose Pseudo inverse

$$\vec{\Phi}^\top \vec{\Phi}^{M \times N} \cdot \vec{\Phi}^{N \times M} = \vec{A}^{M \times M} \quad \text{- depends on number of features of } \vec{x}$$

$$\vec{A}^{-1} = ?$$

\Rightarrow Gradient Descent

$$(AB)^\top = B^\top A^\top$$

$$(A^{-1})^\top = (A^\top)^{-1}$$

(Linear Regression Optimization)

$$\vec{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}$$

$$\bar{\Phi}(\bar{x}_i) = \begin{bmatrix} \varphi_0(\bar{x}_i) \\ \varphi_1(\bar{x}_i) \\ \varphi_2(\bar{x}_i) \\ \vdots \\ \varphi_{M-1}(\bar{x}_i) \end{bmatrix}$$

$$\Phi(X) = \begin{bmatrix} \varphi_0(\bar{x}_1) & \varphi_1(\bar{x}_1) & \dots & \varphi_{M-1}(\bar{x}_1) \\ \varphi_0(\bar{x}_2) & \varphi_1(\bar{x}_2) & \dots & \varphi_{M-1}(\bar{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(\bar{x}_N) & \varphi_1(\bar{x}_N) & \dots & \varphi_{M-1}(\bar{x}_N) \end{bmatrix}_{N \times M}$$

$$\sum_{n=1}^N t_n \bar{\Phi}^T(\bar{x}_n) = \begin{bmatrix} t_n \varphi_0(x_n) & t_n \varphi_1(x_n) & \dots \\ \vdots & \vdots & \vdots \\ t_n \varphi_0(x_{n+1}) & t_n \varphi_1(x_{n+1}) & \dots \end{bmatrix} = \begin{bmatrix} t_n \varphi_0(x_n) + t_{n+1} \varphi_0(x_{n+1}) & t_n \varphi_1(x_n) + t_{n+1} \varphi_1(x_{n+1}) \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}$$

1) $\vec{t}^T \cdot \Phi^*(X) = [t_1 \ t_2 \ \dots \ t_n] \begin{bmatrix} \varphi_0(\bar{x}_1) & \varphi_1(\bar{x}_1) & \dots \\ \varphi_0(\bar{x}_2) & \varphi_1(\bar{x}_2) & \dots \\ \varphi_0(\bar{x}_N) & \varphi_1(\bar{x}_N) & \dots \end{bmatrix}^{N \times M} = \left[\sum_{n=1}^N t_n \varphi_0(\bar{x}_n) \quad \sum_{n=1}^N t_n \varphi_1(\bar{x}_n) \right] \top$

2) $\sum_{n=1}^N \bar{\Phi}(\bar{x}_n) \bar{\Phi}^T(\bar{x}_n) = \sum_{n=1}^N \begin{bmatrix} \varphi_0(\bar{x}_n) \\ \varphi_1(\bar{x}_n) \\ \vdots \\ \varphi_{M-1}(\bar{x}_n) \end{bmatrix} [\varphi_0(\bar{x}_n) \ \varphi_1(\bar{x}_n) \ \dots \ \varphi_{M-1}(\bar{x}_n)] = \sum_{n=1}^N \begin{bmatrix} \varphi_0 \varphi_0 & \varphi_0 \varphi_1 & \dots & \varphi_0 \varphi_{M-1} \\ \varphi_1 \varphi_0 & \varphi_1 \varphi_1 & \dots & \varphi_1 \varphi_{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{M-1} \varphi_0 & \varphi_{M-1} \varphi_1 & \dots & \varphi_{M-1} \varphi_{M-1} \end{bmatrix} =$

$$= \Phi^T \Phi$$

 $M \times N \times N \times M = M \times M$

$$\Phi^T \Phi = \begin{bmatrix} \varphi_0(\bar{x}_1) & \varphi_0(\bar{x}_2) & \dots & \varphi_0(\bar{x}_N) \\ \varphi_1(\bar{x}_1) & \varphi_1(\bar{x}_2) & \dots & \varphi_1(\bar{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{M-1}(\bar{x}_1) & \varphi_{M-1}(\bar{x}_2) & \dots & \varphi_{M-1}(\bar{x}_N) \end{bmatrix} \cdot \begin{bmatrix} \varphi_0(\bar{x}_1) & \varphi_1(\bar{x}_1) & \dots & \varphi_{M-1}(\bar{x}_1) \\ \varphi_0(\bar{x}_2) & \varphi_1(\bar{x}_2) & \dots & \varphi_{M-1}(\bar{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(\bar{x}_N) & \varphi_1(\bar{x}_N) & \dots & \varphi_{M-1}(\bar{x}_N) \end{bmatrix}$$

Weight Decay Regularization

$$t(\bar{x}) = y(\bar{x}, \bar{w}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2), \quad \sigma^2 \neq f(\bar{x})$$

$$P(\vec{t} | X, \bar{w}, \sigma^2) = \prod_{n=1}^N N(t_n | y(\bar{x}_n, \bar{w}), \sigma^2) \xrightarrow[w]{\text{max}} \text{product - общая условная вероятность}$$

вероятностей t при X, \bar{w} и σ^2 .

$$y(\bar{x}_n, \bar{w}) = \bar{w}^\top \bar{\Phi}(\bar{x}_n)$$

$$E(\bar{w}) = \frac{1}{2} \sum_{n=1}^N (t_n - \bar{w}^\top \bar{\Phi}(\bar{x}_n))^2 + \frac{\lambda}{2} \sum_{j=1}^{N-1} |w_j|^q \xrightarrow[w]{\min}$$

- No-bias, zero mean assumption about the parameter regularization.

- standard values for q are 1 and 2.

Регуляризация накладывает давление на значение параметров модели.

$$E(\bar{w}) = \frac{1}{2} \sum_{n=1}^N (t_n - \bar{w}^\top \bar{\Phi}(\bar{x}))^2 + \frac{\lambda}{2} \bar{w}^\top \bar{w} \rightarrow \min$$

$$\frac{d\left(\frac{1}{2} \sum_{j=1}^{N-1} w_j^2\right)}{d(w_j)} = 2 \cdot \frac{1}{2} \cdot (0 + \dots + w_j) = \lambda w_j \Rightarrow \nabla = \begin{bmatrix} \lambda w_1 \\ \lambda w_2 \\ \vdots \\ \lambda w_{N-1} \end{bmatrix} = \lambda \bar{w}$$

$$\nabla E(\bar{w}) = - \sum_{n=1}^N (t_n - \bar{w}^\top \bar{\Phi}(\bar{x}_n)) \bar{\Phi}(\bar{x}_n) + \lambda \bar{w} \rightarrow \min$$

$$\bar{\Phi}^\top = \nabla E(\bar{w})^\top = -\vec{t}^\top \Phi + \bar{w}^\top \Phi^\top \Phi + \lambda \bar{w}^\top \stackrel{\Phi^\top \Phi \in \mathbb{R}^{M \times M}}{\stackrel{\Phi^\top \Phi \in \mathbb{R}^{M \times M}}{=}} -\vec{t}^\top \Phi + \bar{w}^\top (\Phi^\top \Phi + \lambda I)$$

$$\bar{w}^\top = \vec{t}^\top \Phi (\Phi^\top \Phi + \lambda I)^{-1}$$

$$\bar{w} = (\vec{t}^\top \Phi (\Phi^\top \Phi + \lambda I)^{-1})^\top = ((\Phi^\top \Phi + \lambda I)^{-1})^\top \cdot (\vec{t}^\top \Phi)^\top = \underbrace{(\Phi^\top \Phi + \lambda I)^{-1} \cdot (\Phi^\top \vec{t})}_{\text{Normal Equation}}$$

$$\nabla E(\bar{w}) = - \sum_{n=1}^N (t_n - \bar{w}^\top \bar{\phi}(\bar{x}_n)) \bar{\phi}(\bar{x}_n) + \lambda \bar{w}$$

$$\bar{\Phi} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$n \times 1 \quad n \times 1$

$$\bar{\Phi} = \nabla E(\bar{w}) = - \sum_{n=1}^N (t_n \vec{\phi}(x_n) - (\bar{w}^\top \bar{\phi}(\bar{x}_n)) \bar{\phi}(\bar{x}_n)) + \lambda \bar{w} =$$

$$= -\Phi^\top \vec{t} +$$

Therefore we should transpose

~~$$\begin{bmatrix} \Phi_0(x_1) & \Phi_0(x_2) & \dots \\ \Phi_1(x_1) & \Phi_1(x_2) & \dots \\ \Phi_2(x_1) & \Phi_2(x_2) & \dots \\ \Phi_3(x_1) & \Phi_3(x_2) & \dots \end{bmatrix} \quad \begin{bmatrix} \Phi_0(x_1) & \Phi_1(x_1) & \Phi_2(x_1) \\ \Phi_0(x_2) & \Phi_1(x_2) & \Phi_2(x_2) \\ \vdots & \vdots & \vdots \end{bmatrix}$$~~

$$\sum_{n=1}^N \bar{\phi}(\bar{x}_n) \Phi^\top(\bar{x}_n) = \left(\sum_{n=1}^N \begin{bmatrix} \Phi_0(x_n) \\ \Phi_1(x_n) \\ \Phi_2(x_n) \end{bmatrix} \right)^\top \cdot \begin{bmatrix} \Phi_0(x_1) & \Phi_1(x_1) & \Phi_2(x_1) \\ \Phi_0(x_2) & \Phi_1(x_2) & \Phi_2(x_2) \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}^{3 \times 3} =$$

$$= \sum_{n=1}^N \begin{bmatrix} \Phi_0 \Phi_0 & \Phi_0 \Phi_1 & \Phi_0 \Phi_2 \\ \Phi_1 \Phi_0 & \Phi_1 \Phi_1 & \Phi_1 \Phi_2 \\ \Phi_2 \Phi_0 & \Phi_2 \Phi_1 & \Phi_2 \Phi_2 \end{bmatrix}$$

$$\Phi(\bar{X}) = \begin{bmatrix} \Phi_0(\bar{x}_1) & \Phi_1(\bar{x}_1) & \Phi_2(\bar{x}_1) \\ \Phi_0(\bar{x}_2) & \Phi_1(\bar{x}_2) & \Phi_2(\bar{x}_2) \\ \Phi_0(\bar{x}_3) & \Phi_1(\bar{x}_3) & \Phi_2(\bar{x}_3) \end{bmatrix}_{M \times N}$$

$$\Phi^\top(\bar{X}) = \begin{bmatrix} \Phi_0(\bar{x}_1) & \Phi_0(\bar{x}_2) & \Phi_0(\bar{x}_3) \\ \Phi_1(\bar{x}_1) & \Phi_1(\bar{x}_2) & \Phi_1(\bar{x}_3) \\ \Phi_2(\bar{x}_1) & \Phi_2(\bar{x}_2) & \Phi_2(\bar{x}_3) \\ \vdots & \vdots & \vdots \end{bmatrix}_{N \times M}$$

$$\Phi^\top \cdot \Phi = \begin{bmatrix} \sum_{n=1}^N \bar{\phi}_0(\bar{x}_n) \Phi_0(x_n) & \sum_{n=1}^N \bar{\phi}_0(\bar{x}_n) \bar{\phi}_1(\bar{x}_n) & \dots \\ \sum_{n=1}^N \bar{\phi}_1(\bar{x}_n) \bar{\phi}_0(\bar{x}_n) & \sum_{n=1}^N \bar{\phi}_1(\bar{x}_n) \bar{\phi}_1(\bar{x}_n) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}_{M \times M}$$

$$w^\top \Phi^\top \Phi w = \begin{bmatrix} w_1 \sum + w_2 \sum + w_3 \sum & w_1 \sum + w_2 \sum + w_3 \sum & \dots \end{bmatrix}$$

$$w^\top \Phi^\top \Phi w + \lambda w^\top = \begin{bmatrix} w_1 \sum + w_2 \sum + w_3 \sum + w_4 \sum & \dots \end{bmatrix}$$

Методическое решение задачи регрессии

process → Data → Analyze → Estimate (Learning)

estimate unknown params of process generating data.

- Мы не знаем закономерности.

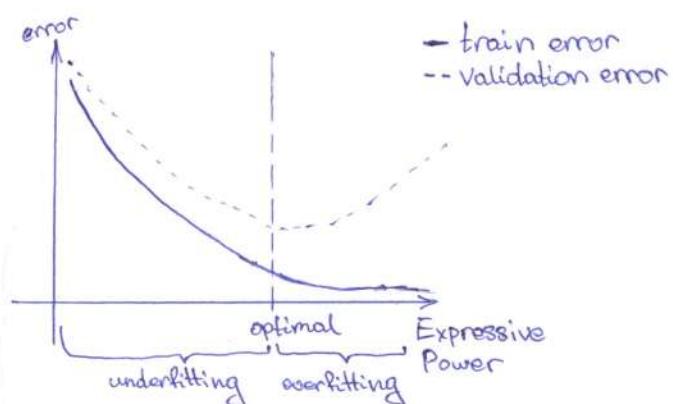
Общее изложение гаусса:

Наше гауссово уравнение;
За какой период времени;
Сколько гауссов;
Среднее (μ);
Разброс (σ^2);

Общие:

Train	Validation	Test
60%	20%	20%

- 1) Try Linear Regression $w^T \bar{x} = \hat{y}$
- 2) Try other basis functions $w^T \phi(\bar{x})$
Test on Validation dataset
- 3) Underfitting vs. Overfitting
- 4) Regularization
- 5) Проверяем гиперпараметры: φ_i, λ
- 6) Модель с наименьшей ошибкой на validation более вероятна.
- 7) Test on test dataset. - результат.



Пример норены train, validation, test

$10^{1000\dots}$ моделей

модели генерируют генераторов, среди них модели имеют любые параметры, и модели подбирают 100% accuracy на validation, но на тестовой выборке 50% acc.

Classification

$$\text{Class} = \{0, 1, 2, \dots, K\}$$

u - object (cat; patient ...)

$$\bar{x} = \mu(u) - \text{features of object (image; temperature, pain...)} \\ \bar{x} \in \mathbb{R}^D, \quad \bar{x} = [x_1, x_2, \dots, x_D]^T$$

Binary Classification ($K=2$)

$$y(\bar{x}, \bar{w}) \rightarrow \text{Class or P(Class)} \quad \rightarrow t$$

Multiclass Classification ($K > 2$)

$$\hat{y}(\bar{x}, \bar{w}) = [p_1, p_2, \dots, p_K]^T \quad \rightarrow \vec{f} - \text{one-hot encoding} \\ \sum_{k=1}^K p_k = 1, \quad p_k \geq 0, \quad k = 1, 2, \dots, K$$

p_k - вероятность принадлежности объекта
к классу k .

Confusion Matrix

$$C = \{c_{ij}\}_{K \times K}$$

$$c_{ij} = |\{u \in C_i \mid y(\bar{x}(u), \bar{w}) = j\}|$$

		y		
		0	1	
t	0	TN	FP	
	1	FN	TP	

В многоклассовой задаче можно считать
что с разными классификациями

Binary Classification Error

$$y(\bar{x}, \bar{w}) = 0, +1$$

✓ True Positive = $\{u_i \in C_1 | y(\bar{x}, \bar{w}) = +1\}$

✓ True Negative = $\{u_i \in C_0 | y(\bar{x}, \bar{w}) = 0\}$

✗ False Positive = $\{u_i \in C_0 | y(\bar{x}, \bar{w}) = +1\}$

✗ False Negative = $\{u_i \in C_1 | y(\bar{x}, \bar{w}) = 0\}$

$$\text{Accuracy} = \frac{TP + TN}{N}$$

$$\begin{aligned} TP + TN + FP + FN &= \# \\ TP + FN &= \#P \\ TN + FP &= \#N \end{aligned}$$

Problem of accuracy is in imbalanced dataset.

Zglobočitá smogu řešení, včetně řešení:
(99%) (1%)

if $y(\bar{x}, \bar{w}) = 0$, then accuracy is 99%, but it is useless.

Důležitá než rozdíl v množství poga

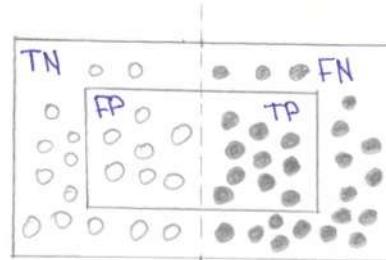
$$\alpha_N = \frac{FP}{FP + TN} \quad (\text{FPR})$$

false positive rate

$$\beta_P = \frac{FN}{FN + TP} \quad (\text{FNR})$$

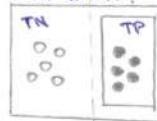
false negative rate

$$\text{Recall} + \beta_P = 1$$



○ - Negative
● - Positive

Best when:



$$\begin{aligned} \text{If model} \rightarrow 1 &\text{ then Precision} = \frac{20}{20+0} = 1 \\ \text{Recall} &= \frac{20}{20+0} = 1 \\ P_{\text{score}} &= \frac{2 \cdot 1 \cdot 1}{1+1} = \frac{2 \cdot 1 \cdot 0.5 \cdot 0.5}{1+0.5} = 0.5 \end{aligned}$$

TN učíme řešení významněji

Precision & Recall

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (\text{TPR})$$

true positive rate

$$F_1\text{-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Fscore vs. average

Sensitivity & Specificity

$$\text{Sensitivity (TPR, Recall)} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (\text{TNR})$$

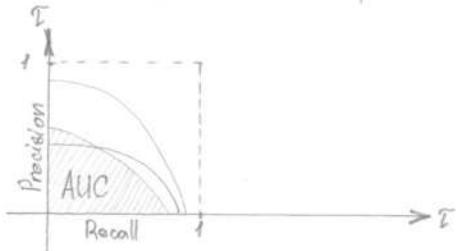
true negative rate

Precision - Recall Curve

Precision & Recall - zuarenuma gaa anpegeññoro noporoboro zuarenuma (2, may).

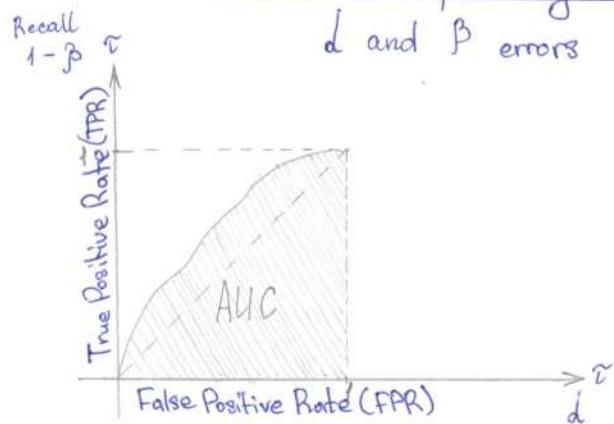
t	P_0	P_1	$\tilde{\tau} = 0,5$	$\tilde{\tau} = 0,6$	$\tilde{\tau} = 0,75$
0	0,3	0,4	1	1	0
0	0,4	0,6	1	1	0
1	0,45	0,55	1	0	0
1	0,2	0,8	1	1	1

$$\uparrow \text{Acc.} = 0,5 \quad \text{Acc.} = 0,25 \quad \text{Acc.} = 0,75$$



AUC - Area Under the Curve

Receiver Operating Characteristic (ROC)



Accuracy:

$$\text{Precision} = \frac{TP}{TP + FP}$$

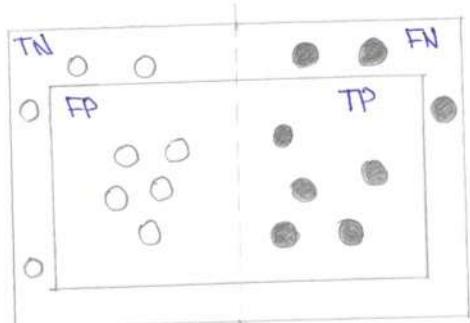
$$\text{Recall} = \frac{TP}{TP + FN} \quad (\text{TPR})$$

Error:

$$\alpha = \frac{FP}{FP + TN} \quad (\text{FPR})$$

$$\beta_p = \frac{FN}{FN + TP} \quad (\text{FNR})$$

● - positive
○ - negative



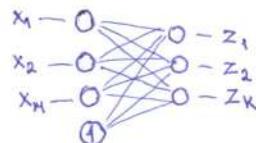
Logistic Regression (Classification)

Data

\vec{x}_i	Class Name	t_i	One-hot encoding vector
\vec{x}_1	Cat	1	[1 0 0]
\vec{x}_2	Dog	2	[0 1 0]
\vec{x}_3	Horse	3	[0 0 1]
\vec{x}_n	"String"	$t_n \in [1; K]$	$\vec{t}_n = [\vec{t}_n^{(1)} \vec{t}_n^{(2)} \dots \vec{t}_n^{(K)}]^T$ $t_n^{(t_n)} = 1, \text{ otherwise } t_n^{(k)} = 0$

Множество монотонных преобразований

$$\left\{ \begin{array}{l} Z_1 = w_{10} + w_{11}x_1 + w_{12}x_2 + \dots + w_{1M}x_M = \vec{w}_1^T \vec{x} + b_1 \\ Z_2 = w_{20} + w_{21}x_1 + w_{22}x_2 + \dots + w_{2M}x_M = \vec{w}_2^T \vec{x} + b_2 \\ \dots \\ Z_K = w_{K0} + w_{K1}x_1 + w_{K2}x_2 + \dots + w_{KM}x_M = \vec{w}_K^T \vec{x} + b_K \end{array} \right.$$

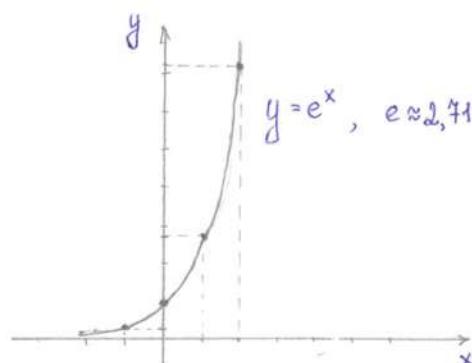


$$\begin{aligned} \vec{x} &\in \mathbb{R}^M \\ \vec{y} &\in \mathbb{R}^K \end{aligned} \quad W = \begin{bmatrix} \vec{w}_1^T \\ \vec{w}_2^T \\ \vdots \\ \vec{w}_K^T \end{bmatrix}, \quad W = (W \quad \vec{b})$$

$$W^{K \times (M+1)} \cdot \vec{x}^{(M+1) \times 1} = \vec{z}^{K \times 1}$$

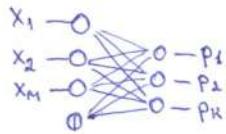
of samples

- $y_i(\vec{x}_i, W) = \text{Softmax}(W \vec{x}_i + \vec{b}) =$
- $= \text{Softmax}(W \vec{x}_i)$
- $\text{Softmax}(\vec{z}) = \left(\frac{e^{z_i}}{\sum_i e^{z_i}} \right)$
- 1) $\sum_i \frac{e^{z_i}}{\sum_i e^{z_i}} = 1$
- 2) Softmax is differentiable function



Logistic Regression. Loss Function.

$$\hat{y}_i(\vec{x}_i, \mathbb{W}) = \text{Softmax}(\mathbb{W}\vec{x}_i + \vec{b}), \quad \hat{y} \in \mathbb{R}^{K \times 1}$$



p_i - Softmax probability,
not PDF of Normal Distribution.

Вероятность правильного класса i на i -ом sample-e и
класса j на j -ом sample-e.

$$P(y_i, y_j) = y_i(\vec{x}_i, \mathbb{W}) \cdot y_j(\vec{x}_j, \mathbb{W}) - \text{Independent Events:}$$

$$P(A \text{ and } B) = P(A) \cdot P(B)$$

Maximum Likelihood

Одна из основных задач обучения логистической регрессии - это максимизация вероятности правильного класса для каждого объекта в наборе обучающих данных.

$$P(T | X, \mathbb{W}) = \underbrace{y_{x_1}^{(t_1)}(\vec{x}_1, \mathbb{W}) \cdots y_{x_n}^{(t_n)}(\vec{x}_n, \mathbb{W})}_{y_k \text{ on right class}} \rightarrow \max_{\mathbb{W}}$$

$$P(T | X, \mathbb{W}) = \prod_{n=1}^N \underbrace{\prod_{k=1}^K y_k^{(k)}(\vec{x}_n, \mathbb{W})^{t_n^{(k)}}}_{y_n^{(t_n)}(\vec{x}_n, \mathbb{W}) = \prod_{k=1}^K y_k^{(k)}(\vec{x}_n, \mathbb{W})^{t_n^{(k)}}} \rightarrow \max_{\mathbb{W}}$$

$$\begin{cases} t_n^{(k)} = 1 & \text{if } k = t_n \\ t_n^{(k)} = 0 & \text{if } k \neq t_n \end{cases}$$

$$\vec{t}_n = [0 \ 0 \ 0 \dots 1 \dots 0 \ 0 \ 0]^T - \text{one-hot encoding}$$

t_n - # of class that's correct.

$$-\ln(P(T | X, \mathbb{W})) = -\sum_{n=1}^N \sum_{k=1}^K \ln(y_n^{(k)}(\vec{x}_n, \mathbb{W})^{t_n^{(k)}}) \rightarrow \min_{\mathbb{W}}$$

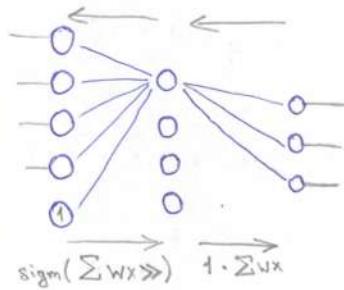
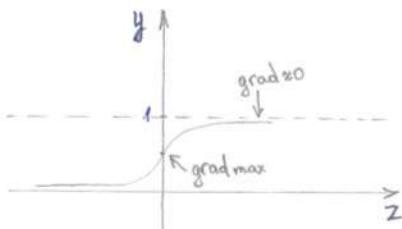
Loss Function (Числова оцінка)

$$E(\mathbb{W}) = -\sum_{n=1}^N \sum_{k=1}^K t_n^{(k)} \ln(y_n^{(k)}(\vec{x}_n, \mathbb{W})) \rightarrow \min_{\mathbb{W}}$$

Weights initialization (Glorot/Xavier)

Vanishing / Exploding Gradients Problem?

$$y(\bar{x}, \bar{w}) = \text{sigm} \left(\sum_i w_i x_i + b \right) , \quad x \sim \mathcal{N}(0, \sigma^2=1)$$



$$\begin{aligned} \text{sigm} &\rightarrow 1) x \in [0; 1] \\ &w \in [0; \frac{1}{n}] \\ &\sum_i x_i w_i = [0; 1] \end{aligned}$$

$$\begin{aligned} \tanh &\rightarrow 2) x \in [-1; 1] \approx \mathcal{N}(0, 1) \\ &w \in [-\frac{1}{n}; \frac{1}{n}] \text{ or } \mathcal{N}(0, \frac{1}{n}) \\ &\sum_i x_i w_i = [-1; 1] \end{aligned}$$

↑
we need this

$$w_i \sim \mathcal{N}(\mu=0, \sigma^2 = \left(\frac{2}{n_{in} + n_{out}} \right)^2)$$

↑ average
↑ sigma=1 exploding gradient

Balance between:

- 1) $\sum w_i x_i \gg$
- 2) Exploding gradient

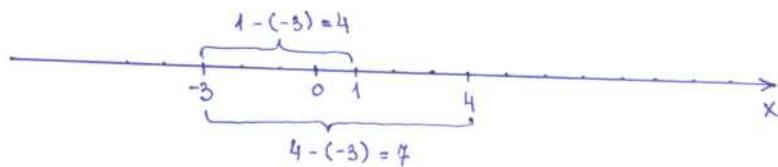
Data Normalization

$$\vec{x} = [x^{(1)} \ x^{(2)} \ \dots \ x^{(M)}]^T$$

$$\tilde{x}^{(j)} = 2 \frac{x^{(j)} - \min_i x_i^{(j)}}{\max_i x_i^{(j)} - \min_i x_i^{(j)}} - 1$$

$$\tilde{x}^{(j)} \in [-1; 1]$$

$x^{(j)}$ нормализуется значение на j-й строке вектора \vec{x}_i из $X \in \mathbb{R}^{M \times N}$



$$\tilde{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}, \quad \sigma \gg 0$$

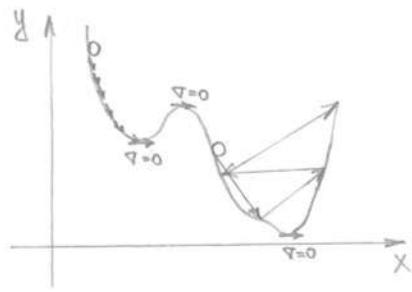
$$\tilde{x}^{(j)} \sim \mathcal{N}(\mu=0, \sigma^2=1)$$

Momentum SGD

1. $\nabla_1 \leftarrow \text{velocity}$

$$2. \boxed{\nabla_2 + \mu \cdot \nabla_1}$$

$$3. \nabla_3 + \mu \nabla_2 + \mu^2 \nabla_1 = \nabla_3 + \mu (\nabla_2 + \mu \nabla_1)$$



$$V_2 = V_1 + \mu V_0 \quad \text{no come ou we momentum change omarawumca.}$$

$$\text{velocity} = 0$$

iter

velocity += momentum

velocity += $\nabla E(W)$

$$W \leftarrow lr \cdot \text{velocity}$$

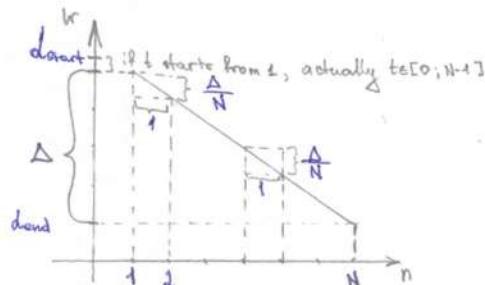
Regularized SGD

$$W \leftarrow lr \cdot \nabla E(W) + \lambda W$$

Learning Rate Decay

1. Linear

$$d_t = d_{\text{start}} - (d_{\text{start}} - d_{\text{end}}) \cdot \frac{t}{N}$$



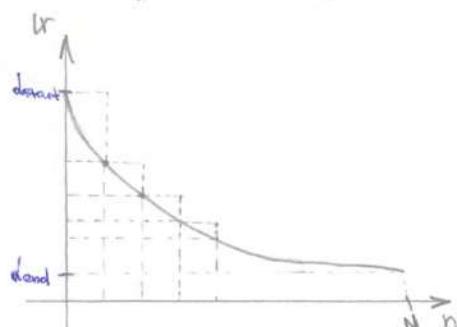
2. Exponential

$$d_t = d_{\text{start}} \cdot \sigma^t$$

$$\sigma = \left(\frac{d_{\text{end}}}{d_{\text{start}}} \right)^{\frac{1}{N-1}}$$

$$d_t = d_{\text{start}} \cdot \left(\frac{d_{\text{end}}}{d_{\text{start}}} \right)^{\frac{t}{N-1}}$$

$$\left(\left(d_{\text{start}} \cdot \frac{\sqrt[3]{\sigma}}{\text{const}} \right) \cdot \sqrt[3]{\sigma} \right) \cdot \sqrt[3]{\sigma} = d_{\text{start}} \cdot \frac{d_{\text{end}}}{d_{\text{start}}} = d_{\text{end}}$$



Dropout

- training: $0.2 \rightarrow 0, p$

- inference: input: 0.8 , we mimic Magnitude $\sim (1-p)$

Logistic Regression. Model parameters estimation.

Gradient Descent

Loss Function

$$E(W) = - \sum_{n=1}^N \sum_{k=1}^K t_n^{(k)} \cdot \ln(y_n^{(k)}(\bar{x}_n, W)) \rightarrow \min_W, \quad W = (W, b)$$

1. Стартовая инициализация весов.

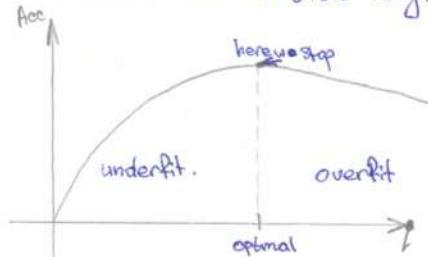
$$w_i \sim \mathcal{N}(\mu=0, \sigma^2 = \frac{1}{M})$$

2. Установка гиперпараметра:

$$W_k = W_{k-1} - \gamma \cdot \nabla E(W_{k-1}), \quad \gamma - \text{learning rate, hyperparameter.}$$

3. Метод перебора по всем весам, когда мотивом к остановке является несходимость критерия.

Максимум остановки обучения - это же гиперпараметр.



Logistic Regression. Model parameters estimation.

Loss Function

$$E(\mathbb{W}) = - \sum_{n=1}^N \sum_{k=1}^K t_n^{(k)} \ln y^{(k)}(\bar{x}_n, \mathbb{W}) \rightarrow \min_{\mathbb{W}}, \quad \mathbb{W} = (\mathbb{W}, b)$$

Dua 1-20 saemema fólofren

$$E(\mathbb{W}) = - \sum_{k=1}^K t^{(k)} \ln y^{(k)}(\bar{x}, \mathbb{W}) = - \ln y^{(t)}(\bar{x}, \mathbb{W}) \rightarrow \min_{\mathbb{W}}$$

$$\bar{y}(\bar{x}, \mathbb{W}) = \text{Softmax}(\bar{z}(\bar{x}, \mathbb{W})) = \text{Softmax}(W\bar{x} + b)$$

$$E(\mathbb{W}) = - \ln y^{(t)}(\bar{x}, \mathbb{W}) = - \ln \left(\frac{e^{z_t}}{\sum_{k=1}^K e^{z_k}} \right) = - \ln e^{z_t} + \ln \left(\sum_{k=1}^K e^{z_k} \right) = - z_t + \ln \left(\sum_{k=1}^K e^{z_k} \right) \rightarrow \min_{\mathbb{W}}$$

$$\{ z_i = \sum_j w_j x_j = \bar{w}^T \bar{x}_i + b_i \}$$

$$\bar{z} = W\bar{x} + b$$

$$\frac{dE(\mathbb{W})}{d w_{tl}} = - (w_{ti} x_i)^T + \frac{1}{\sum_{k=1}^K e^{z_k}} \cdot e^{z_t} \cdot (w_{ti} x_i)^T = -x_i + y_t^{(t)} \cdot x_i = (y_t^{(t)} - 1) x_i$$

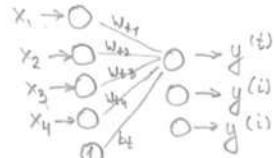
$$\frac{dE(\mathbb{W})}{d w_{il}} = -0 + y_t^{(i)} \cdot (w_{il} \cdot x_i)^T = y^{(i)} \cdot x_i, \quad i \neq t$$

$$\nabla_{\mathbb{W}} E = \begin{bmatrix} \frac{dE}{dw_{11}} & \frac{dE}{dw_{12}} & \frac{dE}{dw_{13}} & \dots & \frac{dE}{dw_{1M}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{dE}{dw_{K1}} & \frac{dE}{dw_{K2}} & \frac{dE}{dw_{K3}} & \dots & \frac{dE}{dw_{KM}} \end{bmatrix} = \begin{bmatrix} y_1 x_1 & y_1 x_2 & \dots & y_1 x_N \\ (y_t - 1)x_1 & (y_t - 1)x_2 & \dots & (y_t - 1)x_M \\ \vdots & \vdots & \ddots & \vdots \\ y_K x_1 & y_K x_2 & \dots & y_K x_N \end{bmatrix}$$

$$\nabla_{\mathbb{W}} E = (\bar{y} - \bar{t}) \cdot \bar{x}^T = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}^{K \times 1} \cdot \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}^{1 \times M} = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}^{K \times M}$$

$$\frac{dE}{db_t} = -1 + y_t = y_t - 1 \Rightarrow \frac{dE}{b} = (\bar{y} - \bar{t})$$

$$\frac{dE}{db_i} = -0 + y_i = y_i$$



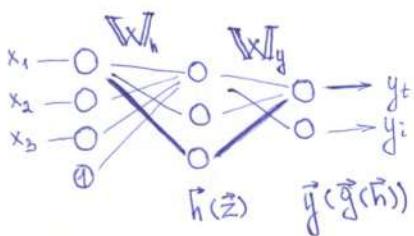
Regularization

$$\nabla_{\mathbb{W}} E = (\bar{y} - \bar{t}) \cdot \bar{x}^T + \lambda \mathbb{W}$$

$$\nabla_b E = (\bar{y} - \bar{t})$$

- Batch gradient descent - entire dataset
- Mini-batch - sample from dataset
- Stochastic - 1 element from dataset

Multi layer perceptron



```

graph LR
    z["z"] -- "n" --> h_bar["h̄"]
    h_bar -- "Sigmoid" --> g_bar["ḡ"]
    g_bar -- "Sum" --> y_hat["ȳ"]
    y_hat -- "Softmax" --> y["y"]

```

$$\vec{y}(\vec{g}(\vec{h}(\vec{x}(\vec{w}))) = \text{Softmax}(\vec{g}(\vec{h})) = \left(\frac{e^{g_i}}{\sum_{j=1}^n e^{g_j}} \right)$$

$$\begin{aligned}
 P(T|X, W_h, W_y) &= \prod_{n=1}^N \prod_{k=1}^K y^{(k)} - (\bar{x}_n, W_h, W_y)^T t_n^{(k)} \xrightarrow{\max} W_h, W_y \\
 -\ln(P(T|X, W_h, W_y)) &= -\sum_{n=1}^N \sum_{k=1}^K \ln y^{(k)} - (\bar{x}_n, W_h, W_y)^T t_n^{(k)} \xrightarrow{\min} W_h, W_y \\
 \sum_{k=1}^K \ln y^{(k)} (\bar{x}_n, W_h, W_y)^T t_n^{(k)} &= -\ln y^{(+)} (\bar{x}, W_h, W_y)^T \xrightarrow{\min} W_h, W_y
 \end{aligned}$$

$$E(W_k) = -\ln g_t^{(k)}(\bar{x}, \bar{w}_t, \bar{x}_y) = -\ln \left(\frac{e^{g_t(\bar{h})}}{\sum_{k=1}^K e^{g_t(\bar{h})}} \right) = -\ln(e^{g_t(\bar{h})}) + \ln \left(\sum_{k=1}^K e^{g_t(\bar{h})} \right) \rightarrow \min_{\bar{W}_k}$$

$$E(\bar{W}_h) = -g_t(\vec{h}) + \ln \left(\sum_{k=1}^{K-1} g_k(\vec{h}) \right) \rightarrow \min_{W_h}$$

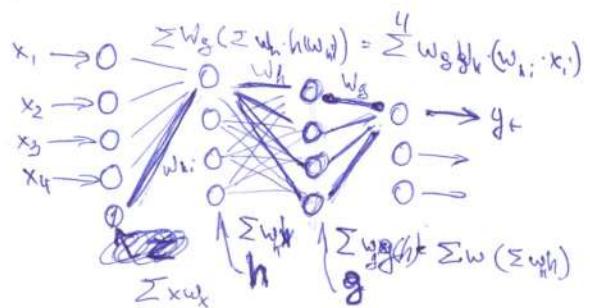
$$g_t(\vec{h}(z)) = \sum_{i=1}^H w_{ti} \cdot h_i(z_i)$$

$$h_i(z_i) = \frac{1}{1 + e^{-z_i}}$$

$$z_i(\bar{w}_n) = \sum_{i=1}^n w_{ki} \cdot x_i$$

$$h(x) = \frac{1}{1+e^{-2x}}$$

$$X(w) = \sum w_i x_i$$



Auto Grad

$\alpha = \text{Number with Grad } (3)$

$$b \approx a + u$$

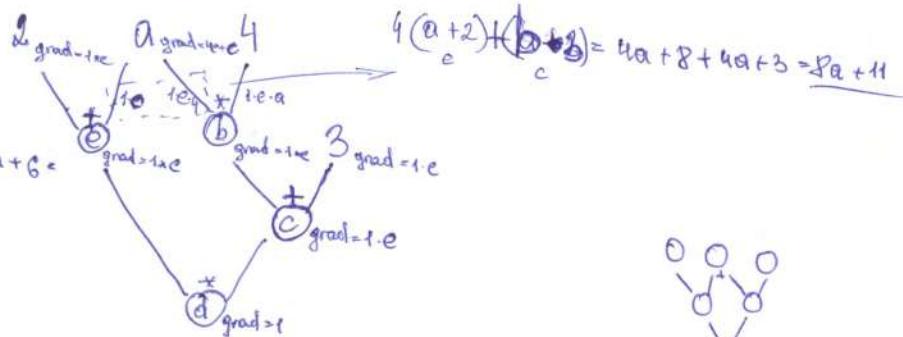
$$c = b + 3$$

$$d = c + (a+2)$$

$$d(x) = (b+3)(a+2)$$

$$\frac{df}{dx} \frac{dg}{dx} + \frac{df}{dx} \frac{dh}{dx} = 8a + 11$$

$$\boxed{\frac{dy}{dx} + \frac{du}{dx}}$$

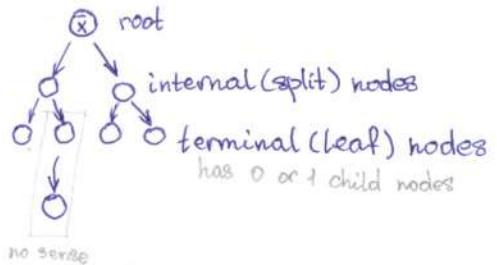


Почему все существуют различные механизмы участвующие в нескольких операциях? Это можно объяснить с помощью принципов

Decision Tree

Decision Tree ML Algorithm - набор связных классификаторов, организованных в иерархическую структуру - дерево.

Дерево - связной ориентированный граф без циклов, в котором каждая вершина имеет не более 1-го ребра.

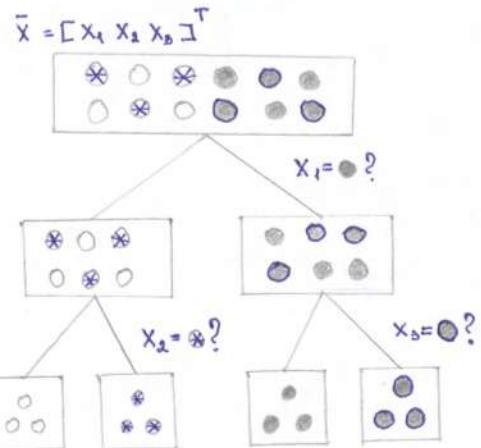


Слабый классификатор - это underfit-модель классификатор, который имеет мало параметров, высокую ошибку, опирается на небольшое кол-во features, быстро вычисляется.

Главное преимущество Decision Tree в том, что для него не нужно весь вектор характеристик (\vec{x}), в отличии от регрессии, то есть когда есть стоимость вычисления каждой характеристики, можно минимизировать издерожки \leftrightarrow максимизировать прибыль.

Например, в случае логистической регрессии для диагностирования заболевания нужно как будто сдать все возможные лаб. тесты. Но, если все сдали 1 и всё ок, то возможно оставшееся 500 сдавать не нужно.

Decision Tree легко интерпретируются.



Использует 3 параметра из 3-х.

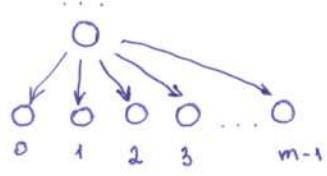
Decision Tree. Internal (split) nodes.

Split function - разделяющая функция.

$$h(\bar{x}, \theta_j) : R^D \times T \rightarrow \{0, m-1\}$$

args Domain Range

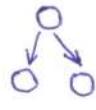
$\theta_j \in T$ - параметр разделяющей функции



Split function - функция, определяющая следующий узел из m номеров j-го узла.

If $m=2 \rightarrow$ Binary Decision Tree

$$h(\bar{x}, \theta_j) : R^D \times T \rightarrow \{0, 1\}$$



Feature Selection Function

$$\tilde{\psi}(\bar{x}) : R^D \rightarrow R^{D'}, \quad D' \leq D$$

(ps:)

$$\bar{x} = [x_1 \ x_2 \ x_3 \ x_4 \dots x_D]^T$$

$$\tilde{\psi}(\bar{x}) = [x_2 \ x_3]^T, \quad D' = 2$$

$\tilde{\psi}(\bar{x})$ - гиперпараметр,
своя отдача для каждого внутреннего узла,
благодаря чему не используется весь feature vector \bar{x} .

$\psi(\bar{x}) = x_i$ - частоты случаев.

Количество используемых характеристики - гиперпараметр,
который что имеет лучший результат будем давать выбор всех,
а что будем выбирать все.

Такие именно характеристики - это параметр.

Decision Tree . 3 types of split function.

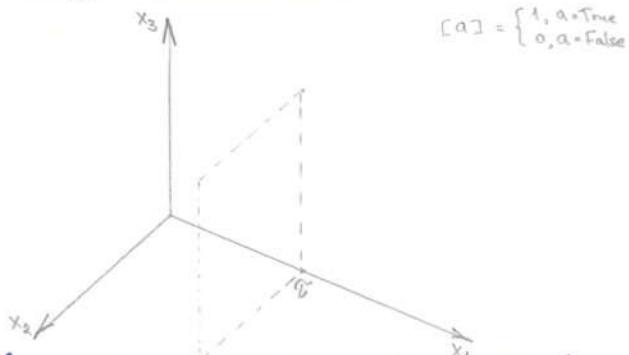
1) Default

$$h(\bar{x}, \theta_j) = [\underbrace{\varphi_1}_{-\infty} > \psi(\bar{x}) > \underbrace{\varphi_2}_{\infty}]$$

$$\theta_j = (\psi, \varphi_1, \varphi_2), \quad \psi(\bar{x}) = x_i$$

Гиперплоскость параллельная всем осям координат, кроме одной.

i.e. $x_i < c$ (brute force)



$$[a] = \begin{cases} 1, & a = \text{True} \\ 0, & a = \text{False} \end{cases}$$

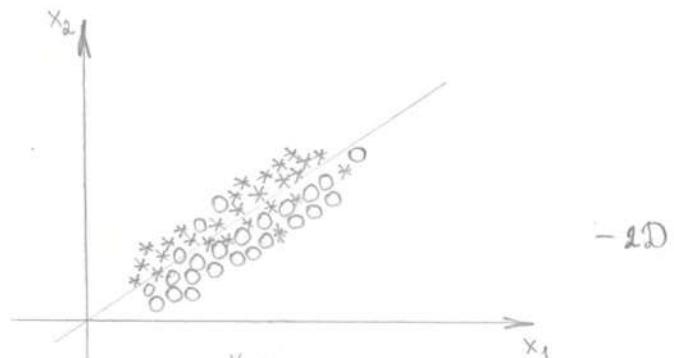
2) Linear

$$h(\bar{x}, \theta_j) = [\varphi_1 > \bar{\psi}(\bar{x})^T \bar{w} > \varphi_2]$$

$$\theta_j = (\bar{\psi}, \bar{w}, \varphi_1, \varphi_2)$$

$$\text{i.e. } x_1 w_1 + x_2 w_2 < c$$

$$x_1 w_1 + b < x_2 \quad (\text{regression})$$

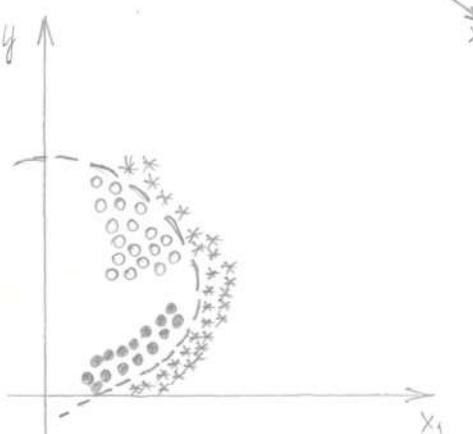
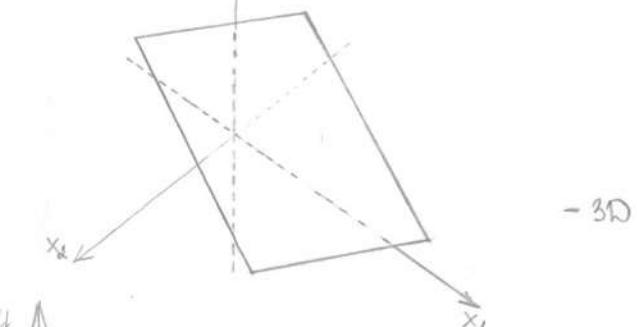


3) Non - Linear

$$h(\bar{x}, \theta_j) = [\varphi_1 > \bar{\psi}(\bar{\psi}(\bar{x}))^T \bar{w} > \varphi_2]$$

$$\theta_j = (\bar{\psi}, \bar{w}, \varphi_1, \varphi_2)$$

\$\psi\$ - basis functions.



Задачи с бинарной меткой \$y\$ (may).

Decision Tree Training.

Notation

Classification classes $C_1, C_2 \dots C_K$

S_i - множество элементов домена узла i
(также dataset-a)

$S_i^{(k)}$ - множество элементов класса k , принадлежащих узлу i .

$N_i = |S_i|$ - количество элементов множества S_i

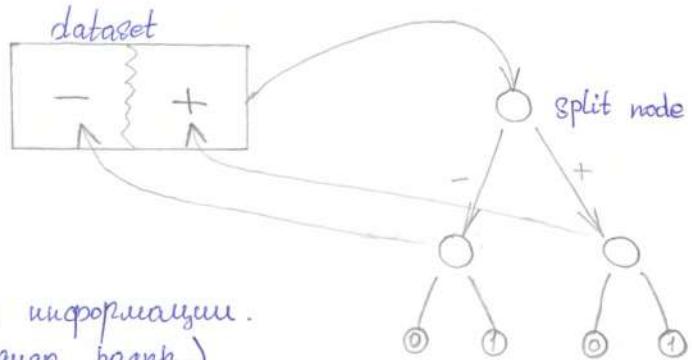
$$N_i^{(k)} = |S_i^{(k)}|$$

S_{ij} - множество элементов подмножества i принадлежащих узлу j .

$$S_{ij}^{(k)}$$

$$N_{ij} = |S_{ij}|$$

$$N_{ij}^{(k)} = |S_{ij}^{(k)}|$$

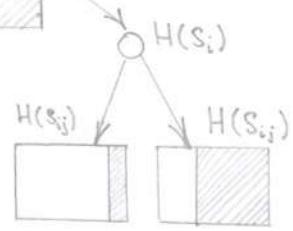
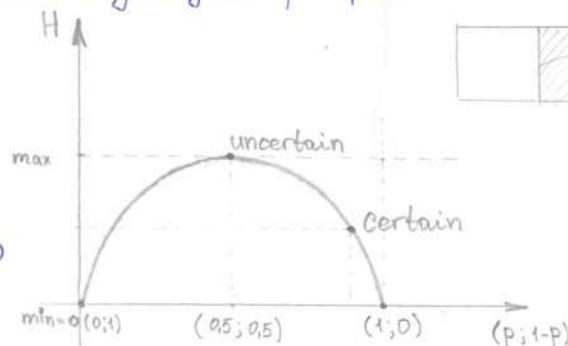


Entropy - оценка количества информации.
(Теория информации Гиббса, Бенара.)

$$H = - \sum_i p_i \log p_i$$

$$\frac{N_i^{(k)}}{N_i} = p^{(k)}$$

$$H(S_i) = - \sum_{k=1}^K \frac{N_i^{(k)}}{N_i} \cdot \log \frac{N_i^{(k)}}{N_i}$$



Information Gain
(Loss Function)

Энтропия падает при демаркиации классов / разбиении
количества классов, и наоборот растет при uniform разбиении.

$$IG = H(S_i) - \sum_j \frac{N_{ij}}{N_i} \cdot H(S_{ij}) \rightarrow \max_{\theta}$$

$$E(\theta) = \sum_j \frac{N_{ij}}{N_i} \cdot H(S_{ij}) \rightarrow \min_{\theta} \in [0; H(S_i)]$$

Найдем $\theta = (\psi, \tau)$, которое минимизирует целевую функцию.

Минимизируем uncertainty на child узлах.

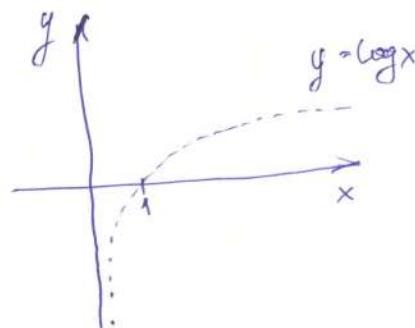
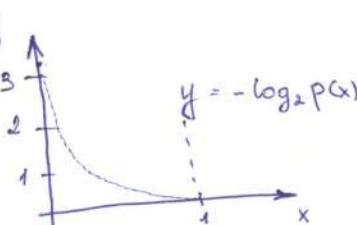


Information theory

Information

$$h(x) = -\log_2(p(x))$$

shannon, e-nats
sure events have more information.
data compression uncertainty.

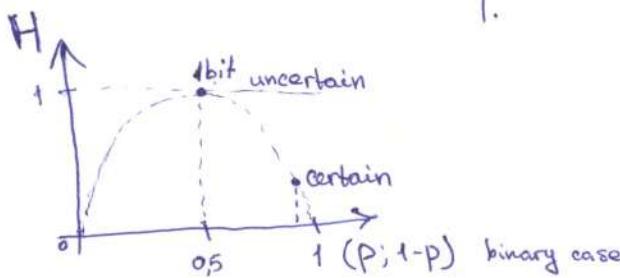


Entropy

$$H(x) = - \sum_i p_i(x) \cdot \log_2(p_i(x))$$

shannon Entropy

$$\frac{1}{2} \cdot \log_2 \frac{1}{2} = \frac{1}{2} \cdot (\log_2 1 - \log_2 2) = \\ = \frac{1}{2} (0 - 1) = -\frac{1}{2}$$



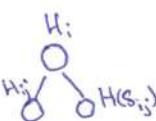
$$\frac{N_i^{(k)}}{N_i} = p^{(k)}$$

$$H(S_i) = \sum_{k=1}^K \frac{N_i^{(k)}}{N_i} \cdot \log_2 \frac{N_i^{(k)}}{N_i}$$

Деяние информации,
uncertain - мало информации.

$N_i^{(1)}$	$N_i^{(2)}$
N_i	

dataset



Information Gain

$$IG = \underbrace{H(S_i)}_{\text{useful}} - \sum_j \frac{N_{ij}}{N_i} \cdot H(S_{ij}) \rightarrow \max_{\Theta}$$

$E(\Theta)$

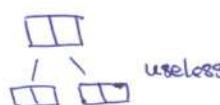
$$E(\Theta) = \sum_j \frac{N_{ij}}{N_i} \cdot H(S_{ij}) \rightarrow \min_{\Theta}$$

$\min \quad \Theta \quad \max \quad H$

минимизируемое деяние uncertainty на child языке, но есть ожидаем certain (90% равно места)

$$1) H_i(0,5; 0,5) = 1$$

$$IG = 1 - \sum_i 0,5 \cdot 1 = 0$$



$$2) IG = 1 - (0,1 \cdot 0 + 0,9 \cdot H_{ij}(0,3; 0,7)) = 1 - (0,9 \cdot (0,3 \log_2 0,3 + 0,7 \log_2 0,7)) = 1 - 0,95 \boxed{0,05}$$

~~IG < 0~~ $\Rightarrow IG \in [0; H(S_i)]$

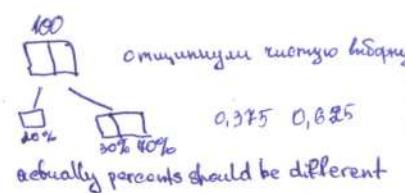
3)

$$\begin{array}{|c|c|} \hline 100 & \\ \hline 0,5 & 0,5 \\ \hline \end{array} - 50/50$$

$$\begin{array}{|c|c|} \hline 20 & 80 \\ \hline 0,7 & 0,3 \\ \hline 14 & 6 \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline 80 & \\ \hline 0,45 & 0,55 \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline 36 & 44 \\ \hline \end{array}$$



$$H(S_i) = - \sum p(x_i) \log_2 p(x_i) = -\left(\frac{1}{2} \cdot \log_2 \frac{1}{2}\right) = 1$$

$$IG = 1 - (0,2 \cdot H(S_{i1}) + 0,8 \cdot H(S_{i2})) = 1 - 0,9696 \boxed{0,03}$$

\Rightarrow same entropy for both parts

$$-0,2(0,7 \log_2 0,7 + 0,3 \log_2 0,3) = 0,2 \cdot 0,88$$

$$-0,8(0,45 \log_2 0,45 + 0,55 \log_2 0,55) = 0,8 \cdot 0,9928$$

Decision Tree. Terminal Node.

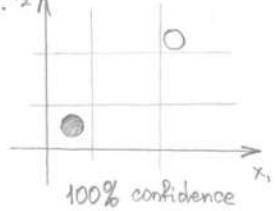
Stopping Criteria (to create terminal node)

- Достигнута заданная глубина d
- $N_i < n$
- $H(S_i) < \epsilon$

If $N_i = N_i^{(k)}$ - Такое же окончание переобученное дерево,
с малым кол-вом элементов на терминальных узлах.
Много элементов на вспомогательных узлах \rightarrow overfitting.



Random Forest: overfit + overfit = ok



$$1) \quad y = \underset{k}{\operatorname{argmax}} N_i^{(k)} \rightarrow \{0, 1, \dots, K\}$$

$$2) \quad \vec{y} = \left(\frac{N_i^{(1)}}{N_i} \quad \frac{N_i^{(2)}}{N_i} \quad \frac{N_i^{(3)}}{N_i} \dots \frac{N_i^{(K)}}{N_i} \right)^T$$

Большая уверенность позована на overfitting.

Random Forest

Random Forest - это ансамбль деревьев решений.

Ensemble classifier - это объединение нескольких классификаторов.

Если мы объединим в один алгоритм несколько разных алгоритмов, то это будет называться fusion classifier.

С помощью склонности классификаторов RF может описывать нелинейные данные.

M деревьев: T_1, T_2, \dots, T_M

$$RF(\bar{x}) = RF(T_1(\bar{x}), T_2(\bar{x}), \dots, T_M(\bar{x})) = \frac{1}{M} \sum_{j=1}^M T_j(\bar{x})$$

$$T_i(\bar{x}) = \left(\frac{N_i^{(1)}}{N_i} \quad \frac{N_i^{(2)}}{N_i} \quad \dots \quad \frac{N_i^{(K)}}{N_i} \right), \quad \bar{x} \in S_i \text{ при обучении}$$

Проблема: детерминированный процесс обучения Decision Tree на одинаковой обучающей выборке даёт одинаковые деревья.

$$T_i = T_j, \forall i, j \Rightarrow RF(\bar{x}) = T_1(\bar{x})$$

Bagging (Bootstrap Aggregation)

Метод по взятии случайного подмножества обучающей выборки.

Random Sampling with replacement, то есть можно брать один и тот же элемент несколько раз.

$$\{1, 2, 3, 4, 5, 6, 7\} \rightarrow \{3, 2, 1, 5, 5, 2, 3\}$$

Random Node Optimization

Выбор параметров split function из случайного подмножества параметров разделяния.

$$IG \rightarrow \max_{\Theta_i \in T_i}, \quad T_i \subset T$$



$P = |\mathcal{T}_i|$ - "Количество случайности", комбинация параметров Θ_i .

$$\Theta_1 = (x_1, z_1)$$

$$\Theta_2 = (x_2, z_2)$$

$$\Theta_3 = (x_1, z_3)$$

$$\Theta_4 = (x_1, z_4)$$

$$\Theta_5 = (x_2, z_5)$$

$\rightarrow \arg \max_{\Theta} IG$ - выбираем комбинацию, дающую максимальное значение IG.

- $P=1$, оптимизация нет, так себе деревья, но разнообразие
- $P=n$, если n большое, деревья будут близки к оптимальному
- $P=1$, если переделал все значения, то деревья одинаковые.

Чем больше кол-во деревьев, M, тем лучше, но после какого-то момента не будем давать улучшений, а только нагрузят.

$(n+k-1)!$	unique sets
$k! (n-1)!$	
ABC	(x_{n+1})
ABCA	$\frac{k}{k}$
permutations	$(2+3-1) = \frac{4!}{2! \cdot 2!}$
4!	$\frac{4!}{2! \cdot 2!}$
0000	
00	
000	
0000	

AdaBoost. Adaptive Boosting

Несмотря на Random Forest в том, что он присущивается к дереву, дереву. RF не имеет возможности обратить внимание на то, что есть особенность в примерах, чтобы закрыть проблему.

Углубленная обучающая выборка

Training samples	Sample weight	Class labels
x_i	w_i	t_i
x_N	w_N	t_N
\bar{x}	\bar{w}	\bar{t}

Числовая функция

Функция:

$$E(\theta) = \frac{1}{2} \sum_{i=1}^N w_i (t_i - y(\bar{x}_i; \theta))^2$$

На всех элементах, на которых значение веса, это значение будет правильным.

Классификация:

$$E(\theta) = - \sum_{i=1}^N w_i \sum_{k=1}^K t_i^{(k)} \ln y(\bar{x}_i; \theta)$$

Дерево решений:

$$IG_w = H_w(S_i) - \sum_j \frac{\tilde{N}_{ij}}{N_i} H_w(S_{ij}) \rightarrow_{\theta} \max$$

$$H_w(S_i) = - \sum_k \frac{\tilde{N}_i^{(k)}}{N_i} \log \frac{\tilde{N}_i^{(k)}}{N_i}$$

$\tilde{N}_i = \sum_{x_i \in S_i} w_i$ — не количество элементов, а сумма весов элементов.
Например, если вес нулевой, он никак не влияет на числовую функцию.

$$\tilde{t}_i = \left(\frac{\tilde{N}_i^{(1)}}{N_i} \quad \frac{\tilde{N}_i^{(2)}}{N_i} \quad \frac{\tilde{N}_i^{(3)}}{N_i} \dots \frac{\tilde{N}_i^{(K)}}{N_i} \right)^T$$

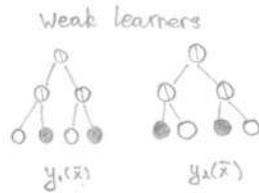
AdaBoost. Рекурсивная загара.

Бинарная загара классификации:

$$C_1, C_2; t_i \in \{-1, 1\}, i=1, 2, \dots, N$$

Слабые классификаторы:

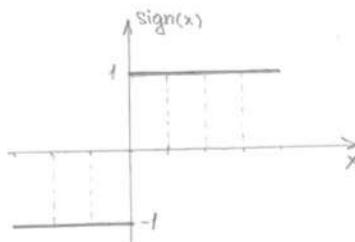
$$y_j(\bar{x}) \rightarrow \{-1, 1\}, j=1, 2, \dots, M$$



Умозрительный классификатор:

$$Y(\bar{x}) = \text{sign} \left(\sum_{j=1}^M d_j y_j(\bar{x}) \right)$$

↑
weight of
weak classifier



Если d_j отрицательно, то он переворачивает кривую

Минимизация весов обучающей выборки

На шаге j :

$$w_i^{(j)} \geq 0, \sum_{i=1}^N w_i^{(j)} = 1$$

$$w_i^{(j)} = \frac{1}{N}, i=1, 2, \dots, N$$

for balanced dataset.

$$\text{or } w_i^{(j)} = \frac{1}{2N_k}, \text{ for imbalanced dataset}$$

Минимизированная целевая функция

Основная идея, максимизировать правильные.

$$J_j = \sum_{i=1}^N w_i^{(j)} I(y_j(\bar{x}_i) \neq t_i), \text{ если } y \neq t, \text{ то вес сокращается, если нет, то не меняется}$$

Или не обратимое значение, а библиотека из exponential loss.

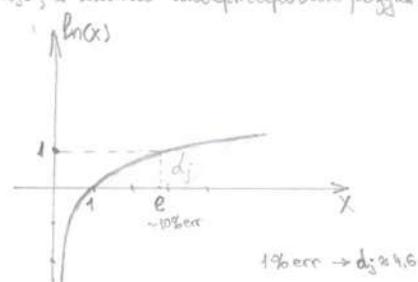
Ошибки слабого классификатора

$$\varepsilon_j = \sum_{i=1}^N w_i^{(j)} \cdot I(y_j(\bar{x}_i) \neq t_i), \text{ если } \varepsilon_j > \frac{1}{2}, \text{ то ошибок нечего, и можно использовать регуляризатор}$$

best: 1) $\ln \frac{1-\varepsilon_j}{\varepsilon_j} = \ln \infty = \infty$

random: 2) $\ln \frac{1-\frac{1}{2}}{\frac{1}{2}} = \ln 1 = 0$

worst: 3) $\ln \frac{1-1}{1} = \ln 0 = -\infty$



Обновление весов обучающей выборки

$$w_i^{(j)}, i=1, 2, \dots, N$$

$$w_i^{(j+1)} = \frac{w_i^{(j)} \cdot e^{d_j \cdot I(y_j(\bar{x}_i) \neq t_i)}}{Z_j}, i=1, 2, \dots, N$$

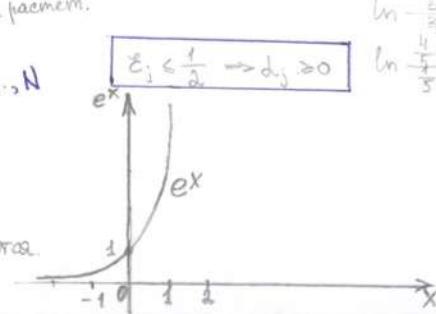
$$Z_j = \sum_{i=1}^N w_i^{(j)} e^{d_j \cdot I(y_j(\bar{x}_i) \neq t_i)}$$

- For normalization.

сумма увеличивающегося веса,

и те изменения на помощь не помогут веса уменьшиться.

if $\varepsilon = \frac{1}{2} \rightarrow \text{stop learning, nothing gonna change}$



$$\ln \frac{1-\varepsilon_j}{\varepsilon_j} = \ln \frac{1}{2} < 0$$

$$\ln \frac{1-\frac{1}{2}}{\frac{1}{2}} = \ln 1 = 0$$

Установление основ AdaBoost.

Неслабкая функция

$$E_0 = \sum_{i=1}^N e^{-\frac{1}{2}t_i Y_N^*(\bar{x}_i)} \quad - \text{exponential loss function}$$

$$Y_N^*(\bar{x}_i) = \sum_{j=1}^M d_j y_j(\bar{x}_i) \quad - \text{without sign} (\Sigma)$$

$$\cdot e^{-\frac{1}{2}t_i Y_N^*(\bar{x}_i)} ; t_i \rightarrow \{-1; 1\}$$

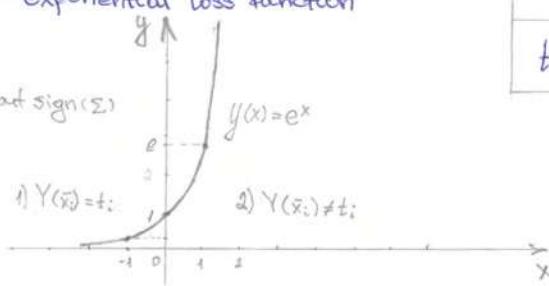
$$1) Y_N^*(\bar{x}_i) \rightarrow \{-c; c\}$$

$$\forall y_j(\bar{x}_i) = t_i$$

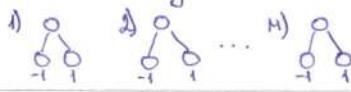
$$e^{-\frac{1}{2}|Y_N^*(\bar{x}_i)|}$$

$$2) \forall y_j(\bar{x}_i) \neq t_i$$

$$e^{+\frac{1}{2}|Y_N^*(\bar{x}_i)|}$$



weak learners $y_j(\bar{x}) \rightarrow \{-1; 1\}, j=1, \dots, M$



$$t_i \rightarrow \{-1; 1\}$$

y_1, y_2, \dots, y_{m-1} - already given weak learners

$y_m - ?$ - AdaBoost is greedy algorithm, so it cares only about currently last classifier.

$$\begin{aligned} E_0 &= \sum_{i=1}^N e^{-\frac{1}{2}t_i Y_N^*(\bar{x}_i)} = \sum_{i=1}^N e^{-\frac{1}{2}t_i (Y_{m-1}^*(\bar{x}_i) + d_m y_m(\bar{x}_i))} = \sum_{i=1}^N e^{-\frac{1}{2}t_i Y_{m-1}^*(\bar{x}_i) - \frac{1}{2}t_i d_m y_m(\bar{x}_i)} = \\ &= \sum_{i=1}^N \underbrace{e^{-\frac{1}{2}t_i Y_{m-1}^*(\bar{x}_i)}}_{W_i^{(m)}} \cdot e^{-\frac{1}{2}t_i d_m y_m(\bar{x}_i)} = \sum_{i=1}^N \tilde{w}_i^{(m)} \cdot e^{-\frac{1}{2}t_i d_m y_m(\bar{x}_i)} = \underbrace{\sum_{i=1}^N \tilde{w}_i^{(m)} e^{-\frac{1}{2}t_i d_m y_m(\bar{x}_i)}}_{\text{norm}} \rightarrow \min_{d_m y_m} \end{aligned}$$

$$\begin{aligned} W_i^{(m)} &= \frac{\tilde{w}_i^{(m)}}{\sum_m} ; \quad \tilde{w}_i^{(m)} > 0 ; \quad \sum_{i=1}^N \tilde{w}_i^{(m)} = 1 \\ \tilde{z}_m &= \sum_{i=1}^N \tilde{w}_i^{(m)} \Rightarrow \tilde{z}_m \cdot \tilde{w}_i^{(m)} = \tilde{z}_m \cdot \frac{\tilde{w}_i^{(m)}}{\sum_m} = \tilde{w}_i^{(m)} \end{aligned}$$

$$\begin{aligned} E &= \sum_{i=1}^N w_i^{(m)} \cdot e^{-\frac{1}{2}t_i d_m y_m(\bar{x}_i)} = \sum_{i=1}^N w_i^{(m)} \cdot e^{-\frac{1}{2}t_i d_m y_m(\bar{x}_i)} \underbrace{I(y_m(\bar{x}_i) \neq t_i)}_{\text{Incorrectly classified}} + \sum_{i=1}^N w_i^{(m)} \cdot e^{-\frac{1}{2}t_i d_m y_m(\bar{x}_i)} \cdot I(y_m(\bar{x}_i) = t_i) \underbrace{=}_{\text{Correctly classified}} \\ &= \sum_{i=1}^N w_i^{(m)} \underbrace{e^{+\frac{1}{2}d_m}}_{\text{const}} \cdot I(y_m(\bar{x}_i) \neq t_i) + \sum_{i=1}^N w_i^{(m)} \underbrace{e^{-\frac{1}{2}d_m}}_{\text{const}} \cdot I(y_m(\bar{x}_i) = t_i) \pm e^{-\frac{1}{2}d_m} \sum_{i=1}^N w_i^{(m)} I(y_m(\bar{x}_i) \neq t_i) = \\ &= (e^{\frac{1}{2}d_m} - e^{-\frac{1}{2}d_m}) \sum_{i=1}^N w_i^{(m)} I(y_m(\bar{x}_i) \neq t_i) + e^{-\frac{1}{2}d_m} \sum_{i=1}^N w_i^{(m)} (I(y_m(\bar{x}_i) = t_i) + I(y_m(\bar{x}_i) \neq t_i)) = \\ &= (e^{\frac{d_m}{2}} - e^{-\frac{d_m}{2}}) \sum_{i=1}^N w_i^{(m)} I(y_m(\bar{x}_i) \neq t_i) + e^{-\frac{d_m}{2}} \underbrace{\left(\sum_{i=1}^N w_i^{(m)} (I(y_m(\bar{x}_i) = t_i) + I(y_m(\bar{x}_i) \neq t_i)) \right)}_{\text{by def. is } 1} \rightarrow \min_{d_m y_m} \end{aligned}$$

$$\begin{aligned} J &= \sum_{i=1}^N w_i^{(m)} \cdot I(y_m(\bar{x}_i) \neq t_i) = \varepsilon_m \Rightarrow E_{dm} = (e^{\frac{d_m}{2}} - e^{-\frac{d_m}{2}}) \varepsilon_m + e^{-\frac{d_m}{2}} \rightarrow \min_{d_m} \\ E_{dm} &= \left(\frac{1}{2} e^{\frac{d_m}{2}} + \frac{1}{2} e^{-\frac{d_m}{2}} \right) \varepsilon_m + \left(-\frac{1}{2} e^{-\frac{d_m}{2}} \right) = 0 ; (e^{\frac{d_m}{2}} + e^{-\frac{d_m}{2}}) \varepsilon_m + e^{-\frac{d_m}{2}} = e^{\frac{d_m}{2}} \cdot e^{\frac{d_m}{2}} ; (e^{\frac{d_m}{2}} + 1) \varepsilon_m = 1 ; e^{\frac{d_m}{2}} + 1 = \frac{1}{\varepsilon_m} ; e^{\frac{d_m}{2}} = \frac{1}{\varepsilon_m} - \frac{\varepsilon_m}{\varepsilon_m} \end{aligned}$$

$$d_m = \ln \frac{1 - \varepsilon_m}{\varepsilon_m}$$

Mame uamurecne ocuofc AdaBoost. (2)

$$E_o = \sum_{i=1}^N e^{-\frac{1}{2} t_i Y_m^*(\bar{x}_i)}$$

$$Y_m^*(\bar{x}_i) = \sum_{j=1}^M d_j y_j(\bar{x}_i)$$

$$y_1, y_2, \dots, y_{m-1}$$

$$y_m - ?$$

$$E_o = \sum_{i=1}^N e^{-\frac{1}{2} t_i Y_m^*(\bar{x}_i)} = \sum_{i=1}^N e^{-\frac{1}{2} t_i (Y_{m-1}^*(\bar{x}_i) + d_m y_m(\bar{x}_i))} = \sum_{i=1}^N \underbrace{e^{-\frac{1}{2} t_i Y_{m-1}^*(\bar{x}_i)}}_{\tilde{W}_i^{(m)}} \cdot e^{-\frac{1}{2} t_i d_m y_m(\bar{x}_i)} =$$

$$= \sum_{i=1}^N \tilde{W}_i^{(m)} e^{-\frac{1}{2} t_i d_m y_m(\bar{x}_i)}$$

$$\text{when } m=1 \Rightarrow \tilde{W}_i^{(m)} = e^0 = 1 \Rightarrow \tilde{W}_i^{(m)} = \frac{1}{\sum_{i=1}^N 1} = \frac{1}{N}$$

$$\left| \begin{array}{l} \tilde{W}_i^{(m)} = \frac{\tilde{W}_i^{(m)}}{\sum_m} \Rightarrow \tilde{W}_i > 0, \sum_{i=1}^N \tilde{W}_i = 1 \\ \sum_m = \sum_{i=1}^N \tilde{W}_i^{(m)} \end{array} \right.$$

$$\bullet \quad \tilde{W}_i^{(m)} = e^{-\frac{1}{2} t_i Y_{m-1}^*(\bar{x}_i)}$$

$$\bullet \quad \tilde{W}_i^{(m+1)} = e^{-\frac{1}{2} t_i Y_m^*(\bar{x}_i)} = e^{-\frac{1}{2} t_i (Y_{m-1}^*(\bar{x}_i) + d_m y_m(\bar{x}_i))} = e^{-\frac{1}{2} t_i Y_{m-1}^*(\bar{x}_i) - \frac{1}{2} t_i d_m y_m(\bar{x}_i)} =$$

$$= \underbrace{e^{-\frac{1}{2} t_i Y_{m-1}^*(\bar{x}_i)}}_{\tilde{W}_i} \cdot e^{-\frac{1}{2} t_i d_m y_m(\bar{x}_i)} = \tilde{W}_i^{(m)} \cdot e^{-\frac{1}{2} t_i d_m y_m(\bar{x}_i)} = \dots$$

$$\left| \begin{array}{l} t_i y_m(\bar{x}_i) = 1 - 2 I(y_m(\bar{x}_i) \neq t_i) \\ \begin{cases} t_i y_m(\bar{x}_i) = 1, \text{ if } t_i = y_m(\bar{x}_i) \\ t_i y_m(\bar{x}_i) = -1, \text{ if } t_i \neq y_m(\bar{x}_i) \end{cases} \end{array} \right.$$

$$\dots = \tilde{W}_i^{(m+1)} = \tilde{W}_i^{(m)} \cdot e^{-\frac{1}{2} d_m (1 - 2 I(y_m(\bar{x}_i) \neq t_i))} = \tilde{W}_i^{(m)} \cdot e^{-\frac{1}{2} d_m + I(y_m(\bar{x}_i) \neq t_i)} = \tilde{W}_i^{(m)} \cdot e^{-\frac{1}{2} d_m \cdot d_m I(y_m(\bar{x}_i) \neq t_i)}$$

$$\text{Norm: } W_i^{(m+1)} = \frac{\tilde{W}_i^{(m+1)}}{\sum_{l=1}^N \tilde{W}_l^{(m+1)}} = \frac{\tilde{W}_i^{(m)} \cdot (e^{-\frac{1}{2} d_m}) \cdot e^{d_m I(y_m(\bar{x}_i) \neq t_i)}}{\sum_{l=1}^N \tilde{W}_l^{(m)} \cdot (e^{-\frac{1}{2} d_m}) \cdot e^{d_m I(y_m(\bar{x}_i) \neq t_l)}} \cdot \frac{e^{\frac{1}{2} d_m}}{e^{\frac{1}{2} d_m}} = \frac{\tilde{W}_i^{(m)} \cdot e^{d_m I(y_m(\bar{x}_i) \neq t_i)}}{\sum_{l=1}^N \tilde{W}_l^{(m)} \cdot e^{d_m I(y_m(\bar{x}_i) \neq t_l)}} \cdot \frac{1}{\sum_m} =$$

$$= \frac{\tilde{W}_i^{(m)} \cdot e^{d_m I(y_m(\bar{x}_i) \neq t_i)}}{\sum_{l=1}^N \frac{\tilde{W}_l^{(m)}}{\sum_m} e^{d_m I(y_m(\bar{x}_i) \neq t_l)}} \quad \boxed{Z_m}$$

$$W_i^{(m+1)} = \frac{W_i^{(m)} \cdot e^{d_m I(y_m(\bar{x}_i) \neq t_i)}}{Z_m}$$

$$1. f = \sum_{i=1}^N W_i^{(m)} \cdot I(y_m(\bar{x}_i) \neq t_i) = \varepsilon_m \rightarrow \min_{y_m}$$

$$2. d_m = \ln \frac{1 - \varepsilon_m}{\varepsilon_m}$$

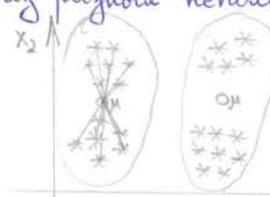
$$3. W_i^{(m+1)} = \frac{W_i^{(m)} \cdot e^{d_m I(y_m(\bar{x}_i) \neq t_i)}}{Z_m}$$

$$W_i^{(1)} = \frac{1}{N} \text{ or } \frac{1}{2 N_k} \text{ for imbalanced dataset.}$$

K-Means Clustering

Задача кластеризации - отнесение элементов выборки к одному из нескольких кластеров так, чтобы элементы из одного кластера были похожи, а из разных непохожи.

Данные: $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N$; $\bar{x} \in \mathbb{R}^D$



Кластеры: C_1, C_2, \dots, C_k

Центроиды: $\bar{\mu}_i \in \mathbb{R}^D$, $i = 1, 2, \dots, k$, то есть для каждого кластера свой центроид.

Целевая функция

$$E = \sum_{k=1}^K \sum_{\bar{x} \in C_k} \| \bar{\mu}_k - \bar{x} \|^2$$

— Euclidean Distance (d) = $\sqrt{(x_1 - x_i)^2 + (y_2 - y_i)^2}$

Инициализация центроидов: Random, min-max range, dataset elements

Назначение элементов кластерам:

$$C_j = \{ \bar{x}_i \mid d(\bar{x}_i; \bar{\mu}_j) = \min_k d(\bar{x}_i; \bar{\mu}_k), i = 1, 2, \dots, N \}, j = 1, 2, \dots, N$$

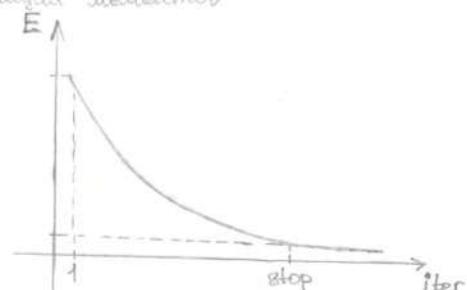
min distance no break.

Обновление центроидов

$$\bar{\mu}_j = \frac{1}{|C_j|} \sum_{\bar{x} \in C_j} \bar{x} : \quad \text{- среднее значение всех входящих элементов}$$

Критерии остановки

- обновление центроидов прекратилось
- падение целевой функции норма прекратилось
- достигнуто заранее заданное кол-во итераций



$$E = \sum_{k=1}^K \sum_{\bar{x} \in C_k} \| \bar{\mu}_k - \bar{x} \|^2 \rightarrow \min_{\bar{\mu}}$$

$$1. \bar{x} \rightarrow \mu$$

2. k, C_k решение по центроиду кластера принимается независимо

$$E_k = \sum_{\bar{x} \in C_k} \| \bar{\mu}_k - \bar{x} \|^2 = \sum_{\bar{x} \in C_k} \sum_{i=1}^D (\mu_k^{(i)} - x^{(i)})^2 \rightarrow \min_{\bar{\mu}}$$

$$\frac{dE_k}{\mu_k^{(i)}} = \sum_{\bar{x} \in C_k} 2 (\mu_k^{(i)} - x^{(i)}) = 0 .$$

$$\sum_{\bar{x} \in C_k} (\mu_k^{(i)} - x^{(i)}) = \sum_{\bar{x} \in C_k} \mu_k^{(i)} - \sum_{\bar{x} \in C_k} x^{(i)} = 0 ; \quad \sum_{\bar{x} \in C_k} \mu_k^{(i)} = |C_k| \mu_k^{(i)} = \sum_{\bar{x} \in C_k} x^{(i)} \Rightarrow$$

$$\Rightarrow \mu_k^{(i)} = \frac{1}{|C_k|} \cdot \sum_{\bar{x} \in C_k} x^{(i)} \Rightarrow \mu_i = \frac{1}{|C_j|} \cdot \sum_{\bar{x} \in C_j} \bar{x}$$

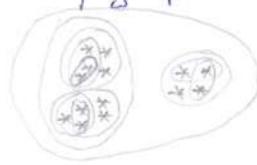
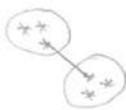
Agglomerative Clustering

$d(\bar{x}_i, \bar{x}_j) = R$ (Ме барга мөнкү ноктумалык расстояние: муз. пронзегение).

Ноңмененное обезгүйнелүү расстояние би расстояние Бакшылого разомера.

1. Single link

$$d(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$



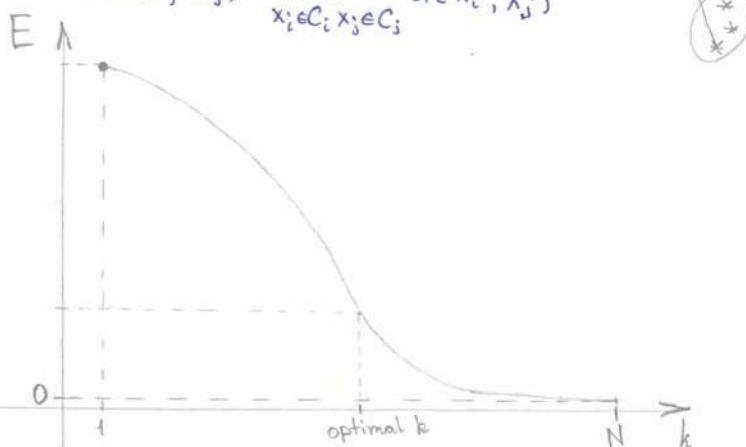
2. Average link

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \cdot \sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j)$$



3. Complete link

$$d(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$



Basic Linear Algebra

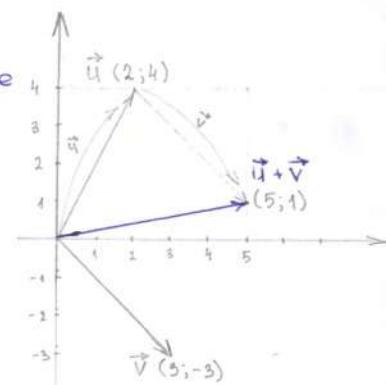
$\vec{u} + \vec{v}$ - elementwise addition of vectors, it's like stepping in space

$c \cdot \vec{v}$ - scaling of vector

The span of \vec{u} and \vec{v} is a set of all linear combinations:

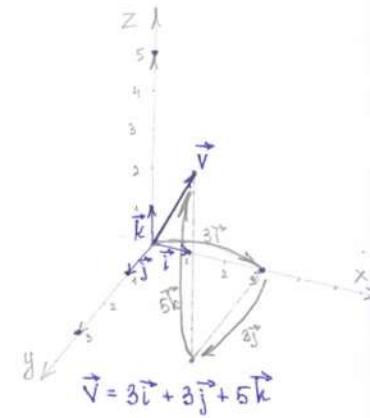
$$c_1 \vec{u} + c_2 \vec{v}$$

Linearly dependent vectors: $\vec{u} = c_1 \vec{v} + c_2 \vec{w}$



Basis vectors of a vector space is a set of linearly independent vectors $\{\vec{i}, \vec{j}, \vec{k}\}$ that span the full space:

$$\vec{v} = c_1 \vec{i} + c_2 \vec{j}$$



Linear Transformations

grid lines remain parallel and evenly spaced

$L(\vec{v}) \rightarrow$ transformed \vec{v}

We can describe any linear transformation for all vectors using the transformed basis vectors:

$$L(\vec{v}) = c_1 L(\vec{i}) + c_2 L(\vec{j}) = \begin{bmatrix} L(\vec{i})_x & L(\vec{j})_x \\ L(\vec{i})_y & L(\vec{j})_y \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$L(\vec{v}) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

shear rotation composition

$$ABC = A(BC) = (AB)C$$

$$AB \neq BA$$

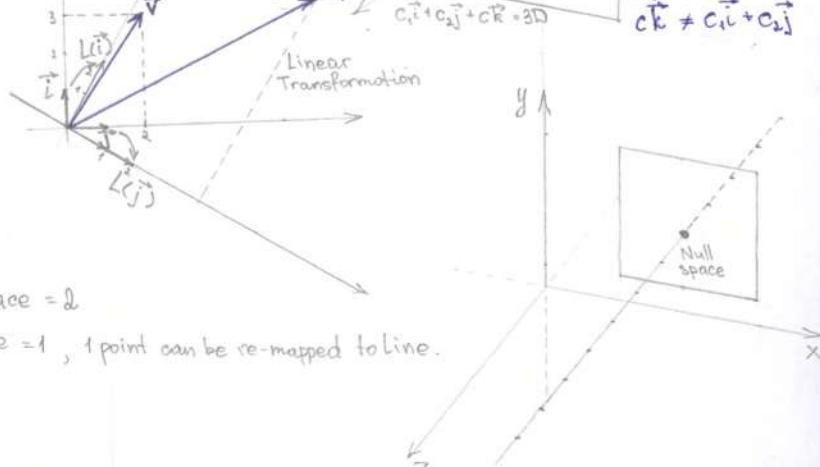
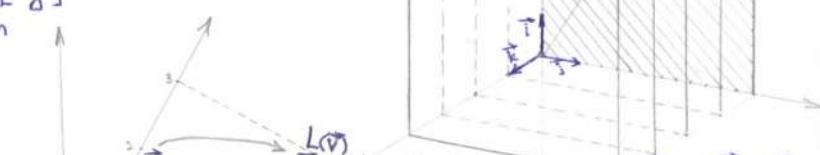
$$\begin{bmatrix} i_x & j_x \\ i_y & j_y \\ i_z & j_z \end{bmatrix} \quad 2D \rightarrow 3D \quad \text{Full rank 2}$$

$\text{Det} = \infty$

$$\begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \end{bmatrix} \quad 3D \rightarrow 2D \quad \text{Rank = 2}$$

Column space = 2

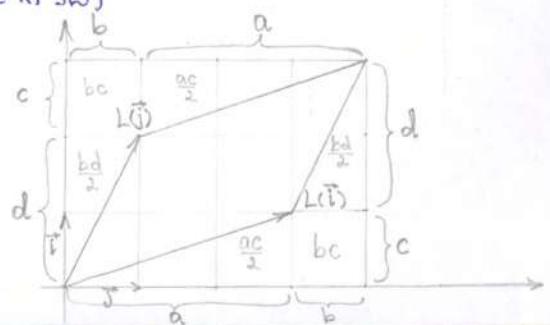
$\text{Null space = 1, 1 point can be re-mapped to line. (kernel)}$



Determinant

determinant - scale of area of linear transformation (or volume in 3D)

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad + bc - ac - bd = (a+b)(c+d) - bc - bc - ac - bd = ad - bc$$



Dot product of vectors

$$\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot \cos(\vec{a}, \vec{b})$$

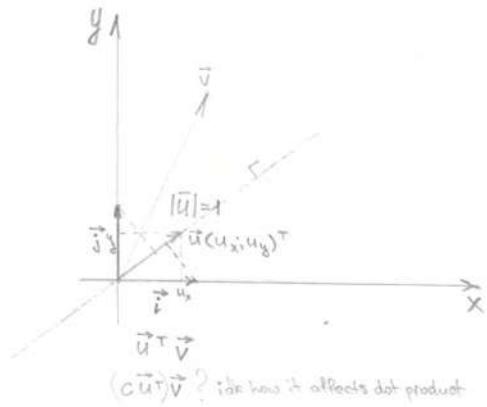
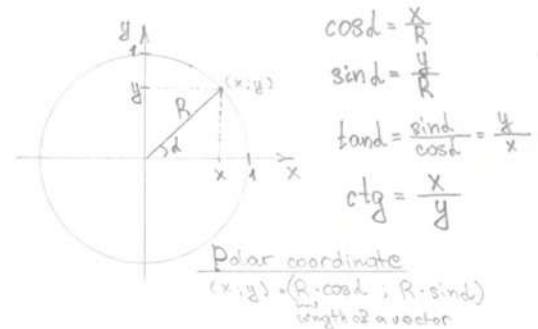
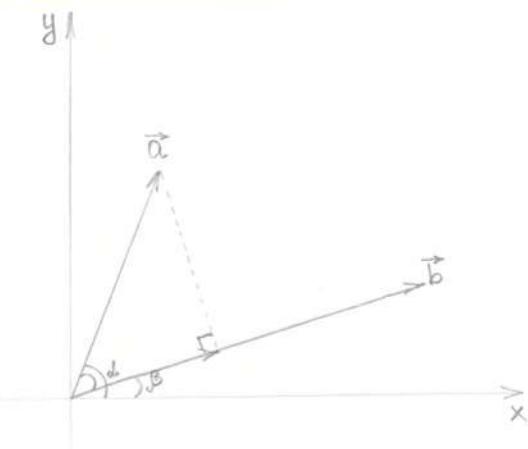
$$\vec{a} \cdot \vec{b} = \vec{a}^T \vec{b} = \vec{b}^T \vec{a} = [i \ j] \begin{bmatrix} x \\ y \end{bmatrix} = xi + yj$$

$$\vec{a} = |\vec{a}| \cdot (\cos \alpha; \sin \alpha)^T$$

$$\vec{b} = |\vec{b}| \cdot (\cos \beta; \sin \beta)^T$$

$$\begin{aligned}\vec{a}^T \vec{b} &= |\vec{a}| |\vec{b}| \cos \alpha \cos \beta + |\vec{a}| |\vec{b}| \sin \alpha \sin \beta = \\ &= |\vec{a}| |\vec{b}| \cos(\alpha - \beta)\end{aligned}$$

$$\cos(\alpha - \beta) = \cos \alpha \cos \beta + \sin \alpha \sin \beta$$



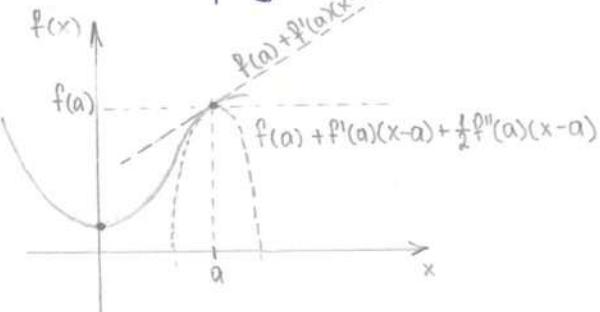
Taylor Series

$$f(x) = P(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

$$P^{(n)}(a) = f^{(n)}(a)$$

Значение функции f more a
и производные высоких степеней
(изменение изменения роста функции)

Approximation of non-polynomial functions (f)
with polynomials (P).



$$P(x) = C_0 + C_1(x-a) + C_2(x-a)^2 + C_3(x-a)^3 + C_4(x-a)^4 + \dots$$

$$0) P(a) = C_0 + C_1 \underbrace{(a-a)}_0 = C_0 = f(a)$$

$$1) P^{(1)}(a) = (0 + C_1(1) + 2C_2(a-a) + \dots) = C_1 = f'(a)$$

$$2) P^{(2)}(a) = 0 + 0 + 2C_2 + 3 \cdot 2 \cdot C_3 \underbrace{(a-a)}_0 + 4 \cdot 3 \cdot C_4 \underbrace{(a-a)}_0^2 + \dots = 2C_2 = f^{(2)}(a)$$

$$\Rightarrow C_2 = \frac{f^{(2)}(a)}{2}$$

$$3) P^{(3)}(a) = 0 + 0 + 0 + 3 \cdot 2 \cdot 1 \cdot C_3 + 4 \cdot 3 \cdot 2 \cdot C_4 \underbrace{(a-a)}_0 + \dots = 3 \cdot 2 \cdot 1 \cdot C_3 = f^{(3)}(a)$$

$$\Rightarrow C_3 = \frac{f^{(3)}(a)}{3!}$$

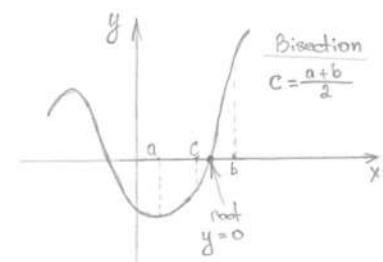
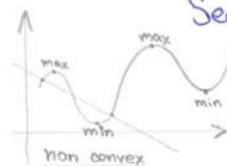
$$P(x) = f(a) + f'(a)(x-a) + \frac{1}{2} f^{(2)}(a)(x-a)^2 + \frac{1}{6} f^{(3)}(a)(x-a)^3 + \frac{1}{24} f^{(4)}(a)(x-a)^4 + \dots$$

Newton's Method

1. Root-finding method. Other: Bisection (Interval halving)

Secant, Brent's methods

2. Optimization method



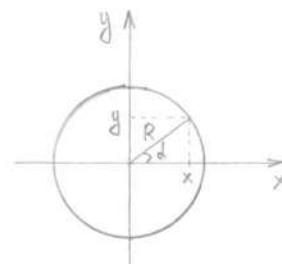
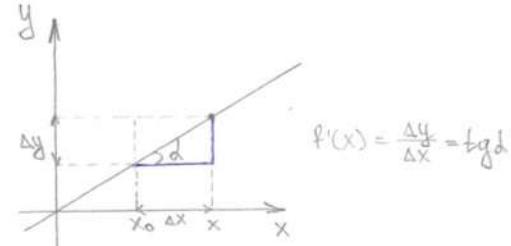
Derivative

$$y = f(x)$$

$$f'(x) = \frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$$\text{tgd} = \frac{\Delta y}{\Delta x} = \mathcal{V}$$

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$



Tangent Line

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

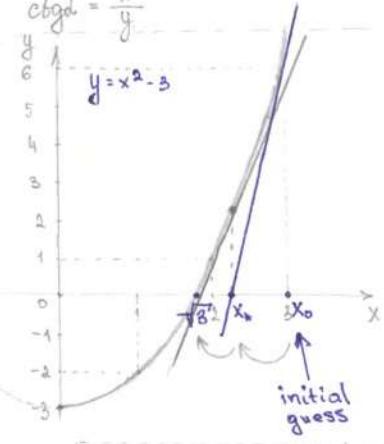
$$\Delta x = x - x_0$$

$$f'(x_0) = \frac{f(x_0 + (x - x_0)) - f(x_0)}{x - x_0} \Rightarrow f'(x_0)(x - x_0) = f(x) - f(x_0)$$

$$f(x) = f(x_0) + f'(x_0)(x - x_0)$$

$$S_0 + \mathcal{V} \cdot \Delta t$$

$$\begin{aligned} \sin d &= \frac{y}{R} \\ \cos d &= \frac{x}{R} \\ \text{tgcd} &= \frac{\sin d}{\cos d} = \frac{y}{x} \\ \text{ctgdc} &= \frac{x}{y} \end{aligned}$$



Newton's root finding method (iterative)

$$f'(x_0) = \frac{f(x) - f(x_0)}{x - x_0} \Rightarrow x - x_0 = \frac{f(x) - f(x_0)}{f'(x_0)}, \quad f'(x) \neq 0$$

$$f(x) = 0 - \text{root}$$

$$x_n = x_0 - \frac{f(x_0)}{f'(x_0)}, \quad \text{until } |x_{n+1} - x_n| < \epsilon,$$

Newton's optimization method

$y'(x) = 0$ - critical points (minima/maxima),
so we can try to find roots of $y'(x)$, i.e. $y'(x) = 0$.

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

$$\begin{bmatrix} \frac{d^2 f}{dx_1^2} & \frac{d^2 f}{dx_1 dx_2} \dots \frac{d^2 f}{dx_1 dx_D} \\ \frac{d^2 f}{dx_2 dx_1} & \frac{d^2 f}{dx_2^2} \dots \frac{d^2 f}{dx_2 dx_D} \\ \dots \\ \frac{d^2 f}{dx_D dx_1} & \frac{d^2 f}{dx_D dx_2} \dots \frac{d^2 f}{dx_D^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{df}{dx_1} \\ \frac{df}{dx_2} \\ \dots \\ \frac{df}{dx_D} \end{bmatrix} = \begin{bmatrix} \frac{df}{dx_1} \\ \frac{df}{dx_2} \\ \dots \\ \frac{df}{dx_D} \end{bmatrix}$$

$$\begin{bmatrix} \frac{d^2 f}{dx_1^2} & \frac{d^2 f}{dx_1 dx_2} \dots \frac{d^2 f}{dx_1 dx_D} \\ \frac{d^2 f}{dx_2 dx_1} & \frac{d^2 f}{dx_2^2} \dots \frac{d^2 f}{dx_2 dx_D} \\ \dots \\ \frac{d^2 f}{dx_D dx_1} & \frac{d^2 f}{dx_D dx_2} \dots \frac{d^2 f}{dx_D^2} \end{bmatrix} \begin{bmatrix} \frac{df}{dx_1} \\ \frac{df}{dx_2} \\ \dots \\ \frac{df}{dx_D} \end{bmatrix} = \begin{bmatrix} \frac{df}{dx_1} \\ \frac{df}{dx_2} \\ \dots \\ \frac{df}{dx_D} \end{bmatrix}$$

Multidimensional case:

$$\vec{x}_1 = \vec{x}_0 - H_{x_0}^{-1} \nabla f(\vec{x}_0)$$

Hessian Matrix

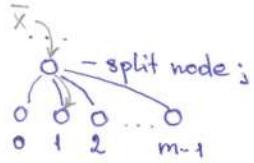
f twice-differentiable.

Decision Tree for Regression Task

- target variables: $t_1, t_2, \dots, t_N \in \mathbb{R}$ $t(\bar{x}) = y(\bar{x}) + \epsilon$

- sample: $\bar{x}_i \in \mathbb{R}^D$

- split function: $h(\bar{x}, \theta_j) : \mathbb{R}^D \times T \rightarrow \{0, 1, \dots, m-1\}$
 - $\theta_j \in T$ - split function parameters
 - usually $m=2$ i.e. Binary Decision Tree



- feature selection function: $\psi(\bar{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}, D' \leq D$
 - usually $D'=1$ i.e. $\psi(\bar{x}) = x_i$

$$\bar{x} = [x_1, x_2, \dots, x_D]^T$$

$$\bar{\psi}(\bar{x}) = [x_2, x_5]^T, D'=2$$

- Split function types remain the same, for example, default hyperplane:

$$h(\bar{x}, \theta_j) = [\tilde{x}_1 > \psi(\bar{x}) > \tilde{x}_2] - \text{Indicator Function}$$

$$\theta_j = (\psi, \tilde{x}_1, \tilde{x}_2)$$

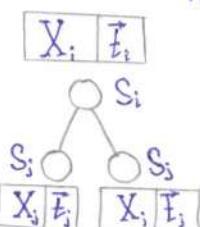
$$\psi(\bar{x}) = x_i$$

$[a] = \begin{cases} 1, & \text{if } a = \text{true} \\ 0, & \text{if } a = \text{false} \end{cases}$

Loss function

$$L = V(S_i) - \sum_j \frac{N_{ij}}{N_i} V(S_j) \rightarrow \max_{\theta} \quad (\text{we use variance instead of entropy})$$

$$V(S_i) = \frac{1}{N_i} \sum_{l=1}^{N_i} (t_i^{(l)} - \underbrace{m_i}_{\substack{\text{prediction} \\ \text{computed only once}}})^2$$



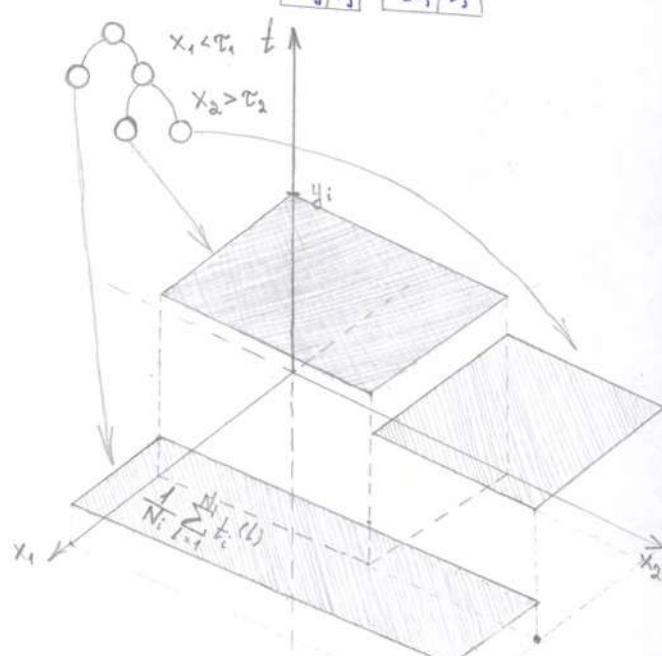
Stopping criteria to create terminal node

- depth d has been reached
- $N_i < n$
- $V(S_i) < \epsilon$

Terminal node (prediction)

$$y_i = \frac{1}{N_i} \sum_{l=1}^{N_i} t_i^{(l)} - \text{average}$$

additionally, we could return variance.



Gradient Boosting: Regression

Объединение нескольких персекторов в единий персектор.

Weak regressor - Decision Tree for Regression Task.
(weak learner in general)

Умственный персектор

$$F(\bar{x}) = \sum_{j=0}^M h_j f_j(\bar{x}) = f_0 + \sum_{j=1}^M h_j f_j(\bar{x}) \quad - \text{сумма} \text{ умственных персекторов,}$$

$h_0 = 1 \qquad f_0 - \text{const}$

$h_j = h \in [0; 1]$
as learning rate

компьютерное представление остатка (residuals).

Loss Function (any differentiable loss function) $L(F_m(\bar{x}), t) = 0 \Leftrightarrow F_m(\bar{x}) = t, \forall i$
 $L_2 \text{ or } L_1$

$$L_2 = \frac{1}{2} \sum_{i=1}^N (F(\bar{x}_i) - t_i)^2$$

$\nabla \exists i : F_m(\bar{x}_i) \neq t_i \Rightarrow L > 0$
but $\bar{x}_i = \tilde{x}_i$ and $t_i \neq \tilde{t}_i \Rightarrow F(\bar{x}_i) = \frac{\tilde{t}_i + t_i}{2}$

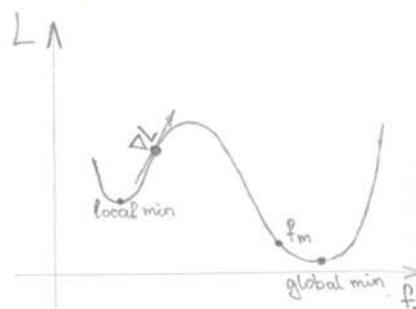
$$L \rightarrow \min_{f_0} : \left(\frac{1}{2} \sum_{i=1}^N (f_0 - t_i)^2 \right)'_{f_0} = \sum_{i=1}^N (f_0 - t_i) = Nf_0 - \sum_{i=1}^N t_i = 0 \Rightarrow f_0 = \frac{1}{N} \sum_{i=1}^N t_i$$

$$\begin{aligned} L \rightarrow \min_{f_m} : L &= \frac{1}{2} \sum_{i=1}^N (F_m(\bar{x}_i) - t_i)^2 = \frac{1}{2} \sum_{i=1}^N (F_{m-1}(\bar{x}_i) + f_m(\bar{x}_i) - t_i)^2 = \\ &= \frac{1}{2} \sum_{i=1}^N (f_m(\bar{x}_i) - \underbrace{(t_i - F_{m-1}(\bar{x}_i))}_{\tilde{t}_i^{(m)} - \text{residuals}})^2 = \frac{1}{2} \sum_{i=1}^N (f_m(\bar{x}_i) - \tilde{t}_i^{(m)})^2 \end{aligned}$$

" y настолько умного персектора f_m свой набор членов переменных $\tilde{t}_i^{(m)}$ - residuals, скомбинировано настолько персектор f_m самым разумно.

Gradient Boosting называется Gradient Descent ($f_m - (-\nabla L)$, $h - lr$), а сума остатков, это f_m уходит в глобальный минимум.

Поменяйте f_m можно overfit-симба, мы же хотим чтобы она лучше всего соответствовать гипотезе в Random Forest.



Gradient Boosting: Classification

Умозрение на aucunamop

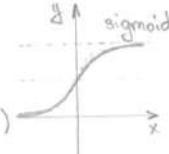
$$y(\bar{x}) = \delta(F(\bar{x})) = \delta(f_0 + \underbrace{d \sum_{j=1}^M f_j(\bar{x})}_{\text{const}}), \quad f_j - \text{regression tree}$$

Sigmoid (δ)

$$\text{Softmax}(\bar{x}) = \left(\frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \right), \quad K=2$$

$$\begin{aligned} \text{Softmax}(\bar{x}) &= \left(\frac{e^{x_0}}{e^{x_0} + e^{x_1}}, \frac{e^{x_1}}{e^{x_0} + e^{x_1}} \right) = \left(\frac{1}{1 + e^{x_1 - x_0}}, \frac{1}{e^{x_0 - x_1} + 1} \right) = |x_1 - x_0 = t| = \\ &= \left(\underbrace{\frac{1}{1 + e^t}}_{1 - \delta}, \underbrace{\frac{1}{1 + e^{-t}}}_{\delta} \right) \end{aligned}$$

$$\frac{1}{1 + e^t} = \frac{1 + e^{-t} - e^t}{1 + e^t} = 1 - \frac{e^t}{1 + e^t} = 1 - \frac{1}{e^{-t} + 1} = 1 - \delta(t) = \delta(-t)$$



Loss function

$$P(E|X) = \prod_{n=1}^N \prod_{k=1}^K y^{(k)}(\bar{x}_n)^{t_n^{(k)}} \rightarrow \max$$

$\vec{y} = [30\% \ 70\%]$ - probability vector

$$L = - \sum_{n=1}^N \sum_{k=1}^K t_n^{(k)} \ln y^{(k)}(\bar{x}_n) \rightarrow \min, \quad K=2$$

$$L = - \sum_{n=1}^N \sum_{k=1}^2 t_n^{(k)} \ln y^{(k)}(\bar{x}_n) = - \sum_{n=1}^N (t_n^{(0)} \ln y^{(0)}(\bar{x}_n) + t_n^{(1)} \ln y^{(1)}(\bar{x}_n)), \quad \delta(x) \in [0; 1] - \text{scalar}$$

$$L = - \sum_{n=1}^N ((1-t_n) \ln(1-\delta(\bar{x}_n)) + t_n \ln \delta(\bar{x}_n)) \rightarrow \min$$

$\overbrace{t_n = C_0}^1 \quad \overbrace{t_n = C_1}^1 \quad \text{Одна из них должна быть}$

$$y(\bar{x}) = \delta(f_0)$$

$$L = - \sum_{n=1}^N ((1-t_n) \ln(1-\delta(f_0)) + t_n \ln \delta(f_0)) \rightarrow \min_{f_0}$$

$$L' = - \sum_{n=1}^N ((1-t_n) \frac{1}{1-\delta(f_0)} (-\delta(f_0))' + t_n \frac{1}{\delta(f_0)} (\delta(f_0))') =$$

$$= - \sum_{n=1}^N ((1-t_n) \frac{1}{1-\delta(f_0)} (-\delta(f_0)(1-\delta(f_0)))' + t_n \frac{1}{\delta(f_0)} \delta(f_0)(1-\delta(f_0))') =$$

$$= - \sum_{n=1}^N ((1-t_n) \cdot (-\delta(f_0)) + t_n (1-\delta(f_0))) = \sum_{n=1}^N (-\delta(f_0) + t_n \delta(f_0) + t_n - t_n \delta(f_0)) =$$

$$= - \sum_{n=1}^N (-\delta(f_0) + t_n) = 0 \quad -\text{is it minimum? proof} \rightarrow \text{if } f'' = +C$$

$$\sum_{n=1}^N \delta(f_0) = \sum_{n=1}^N t_n = N \delta(f_0)$$

$$\delta(f_0) = \frac{1}{N} \sum_{n=1}^N t_n = \frac{1}{1 + e^{f_0}} = p_1 \Rightarrow$$

bias, dataset balance

$$0. (a^x)' = a^x \ln a \cdot (x)'$$

$$1. (\log_a f(x))' = \frac{1}{f(x) \ln a} \cdot f'(x)$$

$$2. \delta'(x) = (1 \cdot (1+e^{-x})^{-1})' = 0 + (-1(1+e^{-x}) \cdot (-e^{-x}))' =$$

$$= -\frac{1}{(1+e^{-x})^2} \cdot (e^{-x})' (-x)' =$$

$$= \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1 \cdot (e^{-x})}{(1+e^{-x})(1+e^{x})} = \delta(x) \cdot \frac{-e^{-x} + 1 - 1}{1 + e^{-x}} =$$

$$= \delta(x) \cdot \frac{1 + e^{-x} - 1}{1 + e^{-x}} = \boxed{\delta(x)(1 - \delta(x))}$$

$$\uparrow \downarrow \rightarrow \left(\sum_{n=1}^N \delta(f_0) - t_n \right)' = \sum_{n=1}^N \delta(f_0)(1 - \delta(f_0)) = +C$$

$$1 + e^{f_0} = \frac{1}{p_1}$$

$$e^{f_0} = \frac{1}{p_1} - 1 = \frac{1 - p_1}{p_1} = \frac{p_0}{p_1}$$

$$f_0 = \ln \frac{p_0}{p_1}$$

Gradient Boosting: Classification (2)

Умовний наступникамоф

$$y(\bar{x}) = \delta(F(\bar{x})) = \delta(f_0 + \sum_{j=1}^M f_j(\bar{x})) , \quad f_j - \text{regression tree}$$

$$\delta(f_0) = \frac{1}{N} \sum_{n=1}^N t_n \longleftrightarrow f_0 = \ln \frac{p_0}{p_1}$$

Loss function

$$L = - \sum_{n=1}^N ((1-t_n)(\ln(1-y(\bar{x}_n)) + t_n \ln y(\bar{x}_n))) \rightarrow \min$$

Given: f_0, f_1, \dots, f_{m-1}

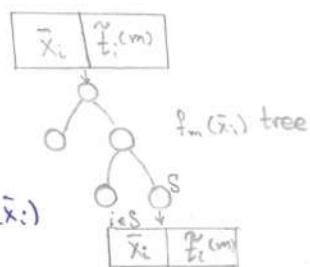
f_m - ?

$$\begin{cases} F_m(\bar{x}_i) = F_{m-1}(\bar{x}_i) + f_m(\bar{x}_i) \\ F_m(\bar{x}_i) = F_{m-1}(\bar{x}_i) - \delta L'(\delta(F_{m-1}(\bar{x}_i)))_{F_{m-1}} \\ f_m = -L' \end{cases}$$

$L'(\delta(F_{m-1}(\bar{x}_i)))$ - спосіб пошуку цільової функції L функції $F_{m-1}(\bar{x}_i)$, який у змозі зменшити L погодно симулює $F_{m-1}(\bar{x}_i)$ в обраних напрямках.

$$\begin{aligned} -L'(\delta(F_{m-1}(\bar{x}_i))) &= ((1-t_i) \ln(1-y(\bar{x}_i)) + t_i \ln y(\bar{x}_i))'_{F_{m-1}} = (\delta(F_{m-1}(\bar{x}_i)))'_{F_{m-1}} = 1, \text{ as } x' = 1 \\ &= ((1-t_i) \ln(1-\delta(F_{m-1}(\bar{x}_i))) + t_i \ln \delta(F_{m-1}(\bar{x}_i)))' = \\ &= (1-t_i) \frac{1}{1-\delta(F_{m-1}(\bar{x}_i))} \cdot (-\delta(F_{m-1}(\bar{x}_i))) (1-\delta(F_{m-1}(\bar{x}_i))) + t_i \frac{1}{\delta(F_{m-1}(\bar{x}_i))} \cdot \delta(F_{m-1}(\bar{x}_i)) (1-\delta(F_{m-1}(\bar{x}_i))) = \\ &= (1-t_i) (-\delta(F_{m-1}(\bar{x}_i))) + t_i (1-\delta(F_{m-1}(\bar{x}_i))) = \\ &= -\delta(F_{m-1}(\bar{x}_i)) + t_i \delta(F_{m-1}(\bar{x}_i)) + t_i - t_i \delta(F_{m-1}(\bar{x}_i)) = \\ &= t_i - \delta(F_{m-1}(\bar{x}_i)) = \tilde{t}_i^{(m)} - \text{label for } f_m(\bar{x}_i) \text{ tree} \end{aligned}$$

$$\tilde{t}_i^{(m)} = t_i - \delta(F_{m-1}(\bar{x}_i))$$



we try to approximate $\tilde{t}_i^{(m)}$ by the underfitting weak regressor tree $f_m(\bar{x}_i)$

but even if it will approximate 100% i.e. $f_m(\bar{x}_i) = \tilde{t}_i^{(m)}, \forall i$

$\delta(F_{m-1}(\bar{x}_i) + \tilde{t}_i^{(m)}) \neq t_i$, it does not guarantee convergence $t_i = \delta(F(\bar{x}_i)), \forall i$

$\tilde{t}_i^{(m)}$ - some направлення. Існа розмір нормалює це засобом, зробив б зупинку настанні residuals. Також б оголошила terminal value.

$$L'(\delta(F_{m-1}(\bar{x}_i) + f_m(\bar{x}_i))) \underset{f_m}{=} 0 \quad \text{we can't just return average of } \tilde{t}_i^{(m)}, \text{ too slow convergence}$$

Gradient Boosting: Classification (3)

$$y(\bar{x}) = \delta(F(\bar{x})) = \delta(f_0 + d \sum_{j=1}^M f_j(\bar{x})) , \quad f_j - \text{regression tree}$$

$$\delta(f_0) = \frac{1}{N} \sum_{n=1}^N t_n \longleftrightarrow f_0 = \ln \frac{P_0}{P_1}$$

Given: f_0, f_1, \dots, f_{m-1}

$f_m - ?$

$$\begin{cases} F_m(\bar{x}_i) = F_{m-1}(\bar{x}_i) + d f_m(\bar{x}_i) \\ F_m(\bar{x}_i) = F_{m-1}(\bar{x}_i) - \lambda L'(\delta(F_{m-1}(\bar{x}_i)))_{F_{m-1}} \end{cases}$$

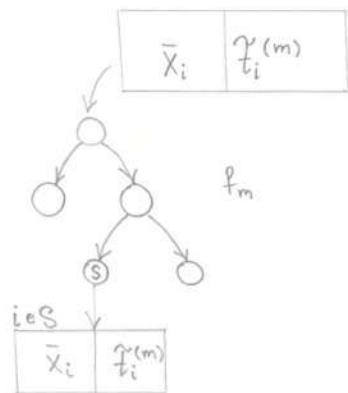
$$f_m(\bar{x}_i) = -L'(\delta(F_{m-1}(\bar{x}_i)))_{F_{m-1}}$$

$$-L'(\delta(F_{m-1}(\bar{x}_i)))_{F_{m-1}} = t_i - \delta(F_{m-1}(\bar{x}_i)) = \tilde{t}_i^{(m)}$$

residuals, we try to minimize Variance at the nodes of the f_m regression tree at these values

$\tilde{t}_i^{(m)}$ - direction for minimising L

$\delta(F_{m-1}(\bar{x}_i) + \tilde{t}_i^{(m)}) \neq t_i$, so we can't just return average on terminal node of f_m tree.



$$L(\delta(F_{m-1}(\bar{x}) + f_m(\bar{x}))) = -\sum_{n=1}^N ((1-t_n) \ln(1-\delta(F_{m-1}(\bar{x}_n) + f_m(\bar{x}_n))) + t_n \ln \delta(F_{m-1}(\bar{x}_n) + f_m(\bar{x}_n))) \rightarrow \min_{f_m}$$

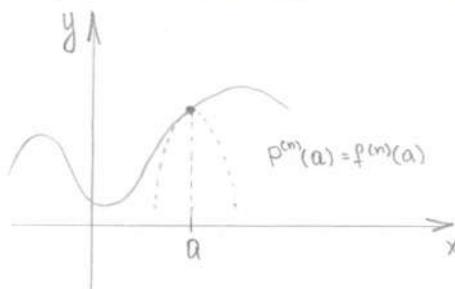
we can't take the derivative of f_m , because we would need f_m in δ but we don't know it.

Taylor Series

$$f(x) = P(x) = \sum_{n=0}^N \frac{f^{(n)}(a)}{n!} (x-a)$$

$$P(x+a) = \sum_{n=0}^N \frac{P^{(n)}(a)}{n!} ((x+a)-a)^n$$

$$f = L, \quad a = F_{m-1}, \quad x = f_m$$



$$L(\delta(F_{m-1}(\bar{x}_i) + f_m(\bar{x}_i))) \approx \sum_{n=0}^2 \frac{L^{(n)}(F_{m-1})}{n!} f_m^n = \underbrace{L(\delta(F_{m-1}(\bar{x}_i)))}_{C_0} + \underbrace{L^{(1)}(\delta(F_{m-1}(\bar{x}_i))) f_m}_{C_1} + \underbrace{\frac{1}{2} L^{(2)}(\delta(F_{m-1}(\bar{x}_i))) f_m^2}_{C_2}$$

$$L'_{f_m}(\delta(F_{m-1}(\bar{x}_i) + f_m(\bar{x}_i))) \approx L^{(1)}(\delta(F_{m-1}(\bar{x}_i))) + L^{(2)}(\delta(F_{m-1}(\bar{x}_i))) \cdot f_m = 0$$

$$f_m = \frac{-L^{(1)}(\delta(F_{m-1}(\bar{x}_i)))_{F_{m-1}}}{L^{(2)}(\delta(F_{m-1}(\bar{x}_i)))_{F_{m-1}}}$$

- Looks like Newton's Optimization Method and it is

$L' = 0$ - minima, we try to find roots of L'

but it's not guaranteed to be global minima, because L is not convex function

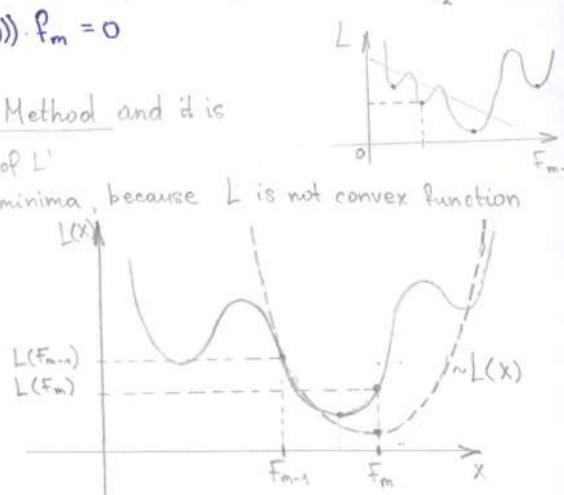
$$F_m(\bar{x}_i) = F_{m-1}(\bar{x}_i) + f_m(\bar{x}_i)$$

$\sim L'(\bar{x}_i) = 0$ local minimum with approximation error

$\sim L(\bar{x}_i) > 0$ but minima

$$L'(\bar{x}_i) = \emptyset$$

$$L(\bar{x}_i) > 0 \quad \text{close to minima}$$



Gradient Boosting: Classification (4)

$$y(\bar{x}) = \delta(F(\bar{x})) = \delta(f_0 + d \sum_{j=1}^M f_j(\bar{x}))$$

$$\delta(f_0) = \frac{1}{N} \sum_{n=1}^N t_n \longleftrightarrow f_0 = \ln \frac{p_0}{p_1}$$

$$\tilde{t}_i^{(m)} = t_i - \delta(F_{m-1}(\bar{x}_i))$$

$$f_m(\bar{x}) = -\frac{L'(\delta(F_{m-1}(\bar{x}_i)))}{L''(\delta(F_{m-1}(\bar{x}_i)))}, \forall i \in S$$

$$L(\delta(F_{m-1}(\bar{x}))) = -\sum_{n=1}^N ((1-t_n) \ln(1-\delta(F_{m-1}(\bar{x}_n))) + t_n \ln \delta(F_{m-1}(\bar{x}_n)))$$

$$\begin{aligned} L'_{F_{m-1}}(\delta(F_{m-1}(\bar{x}))) &= -\sum_{\substack{n=1 \\ n \in S}}^N ((1-t_n) \frac{1}{1-\delta(F_{m-1}(\bar{x}_n))} \cdot (-\delta(F_{m-1}(\bar{x}_n))(1-\delta(F_{m-1}(\bar{x}_n))) + \\ &\quad + t_n \frac{1}{\delta(F_{m-1}(\bar{x}_n))} \cdot \delta(F_{m-1}(\bar{x}_n))(1-\delta(F_{m-1}(\bar{x}_n)))) = \\ &= -\sum_{n=1}^N ((1-t_n)(-\delta(F_{m-1}(\bar{x}_n))) + t_n(1-\delta(F_{m-1}(\bar{x}_n)))) = \\ &= -\sum_{n=1}^N (-\delta(F_{m-1}(\bar{x}_n)) + t_n \delta(F_{m-1}(\bar{x}_n)) + t_n - t_n \delta(F_{m-1}(\bar{x}_n))) = \\ &= \sum_{n=1}^N (\delta(F_{m-1}(\bar{x}_n)) - t_n) \end{aligned}$$

$$L''_{F_{m-1}}(\delta(F_{m-1}(\bar{x}))) = \sum_{\substack{n=1 \\ n \in S}}^N \delta(F_{m-1}(\bar{x}_n))(1-\delta(F_{m-1}(\bar{x}_n)))$$

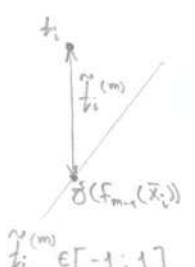
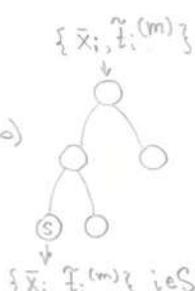
$$f_m(\bar{x}) = \frac{\sum_{n=1}^N (\tilde{t}_n - \delta(F_{m-1}(\bar{x}_n)))}{\sum_{\substack{n=1 \\ n \in S}}^N \delta(F_{m-1}(\bar{x}_n))(1-\delta(F_{m-1}(\bar{x}_n)))} \in [0; \infty) \quad \in (0; 0.25]$$

$$\sim L'(\delta(F_{m-1}(\bar{x}_i)) + f_m(\bar{x}_i)) = 0$$

$\sim L(\delta(F_{m-1}(\bar{x}_i)) + f_m(\bar{x}_i)) > 0$ but minima of $\sim L$

$$L'(\delta(F_{m-1}(\bar{x}_i)) + f_m(\bar{x}_i)) = \emptyset$$

$L(\delta(F_{m-1}(\bar{x}_i)) + f_m(\bar{x}_i)) > 0$ somewhere close to minima according to approximated $\sim L$



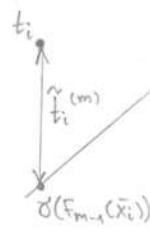
Gradient Boosting: Classification . Summary.

$$y(\bar{x}) = \delta(F(\bar{x})) = \delta(f_0 + d \sum_{j=1}^N f_j(\bar{x}))$$

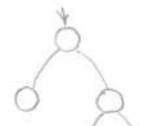
$$f_0 = \frac{p_0}{P_1} \iff \delta(f_0) = \frac{1}{N} \sum_{n=1}^N t_n$$

$$\tilde{t}_i^{(m)} = t_i - \delta(F_{m-1}(\bar{x}_i)) \in [-1, 1]$$

$$f_m(\bar{x}) = \frac{\sum_{i \in S} (t_i - \delta(F_{m-1}(\bar{x}_i)))}{\sum_{i \in S} \delta(F_{m-1}(\bar{x}_i))(1 - \delta(F_{m-1}(\bar{x}_i)))} \in [0; \infty)$$



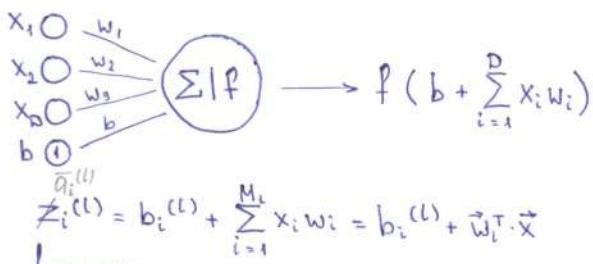
$$\{\bar{x}_i, \tilde{t}_i^{(m)}\}$$



$$\{\bar{x}_i, \tilde{t}_i^{(m)}\}, i \in S$$

Multilayer Perceptron (MLP)

Neuron



f - activation function:

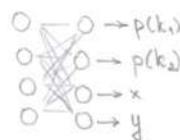
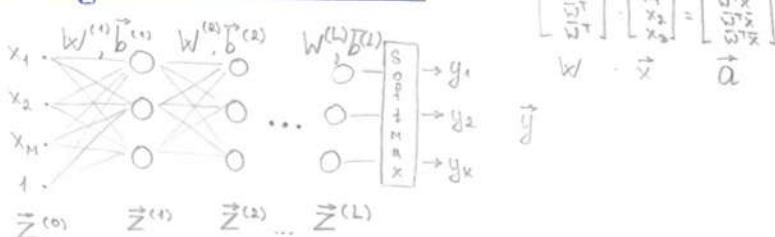
Sigmoid: $f(a) = \frac{1}{1+e^{-a}}$

Tanh: $f(a) = \frac{e^{2a}-1}{e^{2a}+1}$

ReLU: $f(a) = \max(0; a)$

Leaky ReLU: $f(a) = \max(\alpha a; a)$

Layer



M_l - number of neurons

$$\begin{aligned} \vec{Z}^{(l)} &= \vec{f}(\vec{a}^l) = \vec{f}(\vec{a}^{(l)}) = \vec{f}(\vec{W}^{(l)} \cdot \vec{Z}^{(l-1)} + \vec{b}^{(l)}) = \\ &= \vec{f}(\vec{b}^{(l)} + \vec{W}^{(l)} \cdot \vec{f}(\vec{b}^{(l-1)} + \vec{W}^{(l-1)} \cdot \vec{f}(\dots, \vec{x}))) \end{aligned}$$

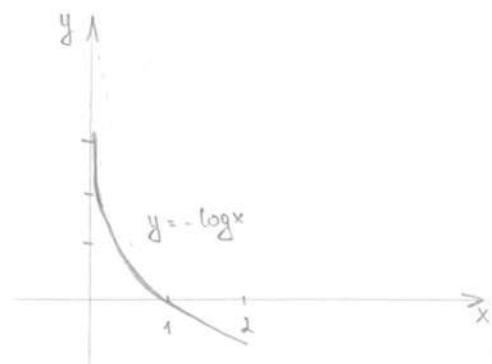
$$Y(\vec{x}, \mathbb{W}) = \text{Softmax}(\vec{Z}^{(L)}(\vec{x}, \mathbb{W}))$$

$$z_i > z_j \rightarrow y_i > y_j$$

Loss function

$$L(\mathbb{W}) = - \sum_{i=1}^N \sum_{k=1}^K t_i^{(k)} \ln Y_k(\vec{x}_i, \mathbb{W}), \quad t_i^{(k)} = \begin{cases} 1, & \text{if } k=t \\ 0, & \text{otherwise} \end{cases}$$

$$\vec{t} = (0, 0, 1, 0, 0, \dots)^T$$



1. Weights initialization

$$w_i \sim \mathcal{N}(\mu=0, \sigma^2 = \frac{2}{n_i})$$

$$2. \quad w_n = w_{n-1} - \gamma \nabla L(w_{n-1})$$

3. Stop criteria

MLP Backprop (Jacobian)

$$E(\bar{y}(\bar{x})) \quad \nabla_E E = 1$$

$$\bar{y}(\bar{x}), \quad \nabla_y E = \left[\frac{dE}{dy_1}, \frac{dE}{dy_2}, \dots, \frac{dE}{dy_M} \right]^T$$

$\nabla_x E - ?$

$$\bar{y}(\bar{x}) = \begin{cases} y_1(\bar{x}) = b_1 + \sum_{j=1}^M w_{j1}x_j \\ y_2(\bar{x}) = b_2 + \sum_{j=1}^M w_{j2}x_j \\ y_3(\bar{x}) = b_3 + \sum_{j=1}^M w_{j3}x_j \end{cases}$$

$\Rightarrow \bar{y}_1 \rightarrow y_1$
 $\Rightarrow \bar{y}_2 \rightarrow y_2$
 $\Rightarrow \bar{y}_3 \rightarrow y_3$

$$\nabla_x E = \left[\underbrace{\sum_{i=1}^M \frac{dE}{dy_i} \frac{dy_i}{dx_1}}_{\text{Why sum?}} \quad \sum_{i=1}^M \frac{dE}{dy_i} \frac{dy_i}{dx_2} \dots \right]^T \quad \frac{dE}{dx_1} = \underbrace{\frac{dE}{dy_1} \cdot \frac{dy_1}{dx_1}}_{\frac{dE}{dy_1}} + \underbrace{\frac{dE}{dy_2} \cdot \frac{dy_2}{dx_1}}_{\frac{dE}{dy_2}} + \underbrace{\frac{dE}{dy_3} \cdot \frac{dy_3}{dx_1}}_{\frac{dE}{dy_3}}$$

$$\text{Jacobian} = \frac{d\bar{y}}{dx} = \left[\left[\frac{dy_1}{dx_1}, \frac{dy_1}{dx_2}, \dots \right] \quad \left[\frac{dy_2}{dx_1}, \frac{dy_2}{dx_2}, \dots \right] \quad \left[\frac{dy_3}{dx_1}, \frac{dy_3}{dx_2}, \dots \right] \right]$$

$\frac{df}{dx} = \frac{dy}{dx} \cdot \frac{df}{dy} \rightarrow \frac{df}{dx} = \frac{df}{dy_1} \frac{dy_1}{dx} + \frac{df}{dy_2} \frac{dy_2}{dx}$

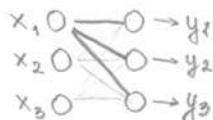
$$\nabla_x E = \text{Jacobian}^T \cdot \nabla_y E$$

$x_0 \rightarrow y_1 \rightarrow \dots \rightarrow y_2 \rightarrow \dots \rightarrow y_3$

Матрица
законов
произвольных

Why sum?

$$\frac{dE}{dx_1} = \sum_{i=1}^M \underbrace{\frac{dE}{dy_i} \frac{dy_i}{dx_1}}_{\text{Chain rule}}$$



$$E = \sum_{i=1}^M (t^{(i)} - y^{(i)}(x_1, x_2, x_3))^2$$

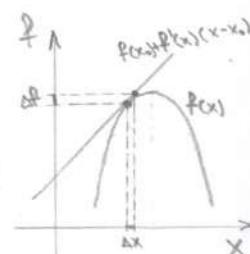
Сумма вкладов $y_i(x_1)$ отвечает за изменение.

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{\Delta f}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{df}{dx}$$

$$f'(x) = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{f(x_0 + x - x_0) - f(x_0)}{x - x_0} = \frac{f(x) - f(x_0)}{x - x_0} \Rightarrow f(x) = f(x_0) + f'(x_0)(x - x_0)$$

$$f'(x) = \frac{\Delta f}{\Delta x} \Rightarrow \Delta f = f'(x) \Delta x \quad \rightarrow df = f'(x) \cdot dx$$

$$df = \frac{\partial f}{\partial x} \cdot dx + \frac{\partial f}{\partial y} \cdot dy$$

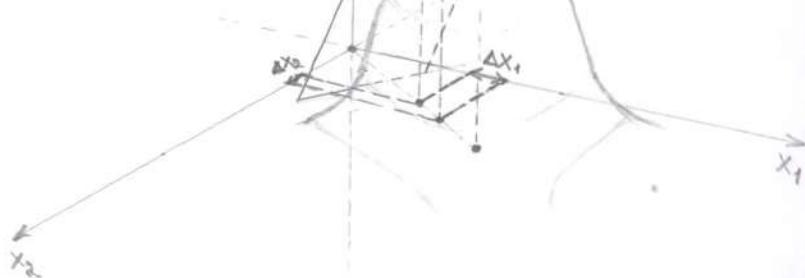


$$f(x, y)$$

$$\Delta f = [f(x_0, y_0) + f_x(x_0, y_0) \Delta x + f_y(x_0, y_0) \Delta y] - f(x_0, y_0) = f_x(x_0, y_0) \Delta x + f_y(x_0, y_0) \Delta y$$

$$df = \frac{\partial f}{\partial x} \cdot dx + \frac{\partial f}{\partial y} \cdot dy$$

$$E(y_1(x_1), y_2(x_1), \dots)$$



Classification MLP Backprop⁽¹⁾ Loss function

$\vec{y} = \text{Softmax}(\vec{z}^{(L)})$, $\vec{t} = [0010\dots0]$ one-hot encoding.

$$p(T|X, W) = \prod_{n=1}^N \prod_{k=1}^K t_n^{(k)} y^{(k)}(\vec{x}_n, W)^{t_n^{(k)}} \rightarrow \max$$

$$\ln p(T|X, W) = \sum_{n=1}^N \sum_{k=1}^K t_n^{(k)} \ln y^{(k)}(\vec{x}_n, W) \rightarrow \max$$

$$E = - \sum_{n=1}^N \sum_{k=1}^K t_n^{(k)} \ln y^{(k)}(\vec{x}_n, W) \rightarrow \min$$

Forward pass

$$\vec{y} = \text{Softmax}(\vec{z}^{(L)}) = \text{Softmax}(\vec{f}(\vec{a}^{(L)})) = \text{Softmax}(\vec{f}(\vec{b}^{(L)} + W^{(L)} \vec{z}^{(L-1)}))$$

$$\vec{z}^{(0)} = \vec{x}$$

$$\vec{z}^{(0)} \rightarrow \vec{a}^{(1)} \rightarrow \vec{z}^{(1)} \rightarrow \dots \rightarrow \vec{a}^{(L)} \rightarrow \vec{z}^{(L)} \rightarrow \vec{y} \rightarrow E$$

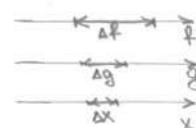
Backward pass

$$\underbrace{\nabla_E E}_{1} \rightarrow \underbrace{\nabla_y E}_{\ln(y)} \rightarrow \underbrace{\nabla_{z^{(L)}} E}_{\text{Softmax}(z^{(L)})} \rightarrow \underbrace{\nabla_{a^{(L)}} E}_{\nabla_{W^{(L)}} E} \rightarrow \dots \rightarrow \underbrace{\nabla_{a^{(1)}} E}_{\nabla_{W^{(1)}} E} \rightarrow \underbrace{\nabla_{b^{(1)}} E}_{\nabla_{b^{(1)}} E}$$

$$E = - \sum_{k=1}^K t^{(k)} \ln y^{(k)}(\vec{x}, W) \rightarrow \min$$

Chain Rule:

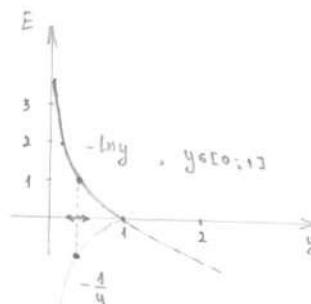
$$\begin{aligned} f'(g(x))_x &= f'(g) g \cdot g'(x)_x \\ \frac{df}{dx} &= \frac{df}{dg} \cdot \frac{dg}{dx} \end{aligned}$$



$$\left\{ \begin{array}{l} \frac{dE}{dy^{(i)}} = (-\ln y^{(i)}(\vec{x}, W))'_y = -\frac{1}{y^{(i)}(\vec{x}, W)}, \quad i = t_k - \text{ground truth} \\ \frac{dE}{dy^{(i)}} = 0, \quad i \neq t_k \end{array} \right.$$

$$\vec{t} = [0 \ 0 \ \dots \underset{t_k}{1} \ \dots \ 0 \ 0]^T$$

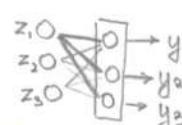
$$\nabla_y E = [0 \ 0 \ \dots \ -\frac{1}{y^{(t_k)}(\vec{x}, W)} \ \dots \ 0 \ 0]^T$$



отрицательный наклон
значение нужно увеличивать y,
чтобы уменьшить E

$$\vec{y} = \text{Softmax}(\vec{z}^{(L)})$$

$$\frac{dE}{dz_i^{(L)}} = \frac{dE}{dy_i} \cdot \frac{dy_i}{dz_i^{(L)}}$$



$$\Rightarrow \frac{dE}{dz_i^{(L)}} = \sum_{j=1}^M \frac{dE}{dy_j} \cdot \frac{dy_j}{dz_i^{(L)}}$$

$$\nabla_{z^{(L)}} E = (\frac{d\vec{y}}{dz^{(L)}})^T \cdot \nabla_y E$$

где $\frac{d\vec{y}}{dz^{(L)}}$ называется Jacobian $-y^{(j)}y^{(i)}$

$$\nabla_{z^{(L)}} E = \vec{y} - \vec{t}$$

$$\nabla_z E = \begin{bmatrix} 0,1 \\ 0,5-1 \\ 0,4 \end{bmatrix} = \begin{bmatrix} 0,1 \\ -0,5 \\ 0,4 \end{bmatrix}$$

Несколько градиентов E, нужно выбрать в каком направлении $\nabla_z E$

$$-\nabla_z E = \begin{bmatrix} -0,1 \\ 0,5 \\ -0,4 \end{bmatrix}$$

Уменьшаем коэффициенты ошибки

$$\begin{bmatrix} z_1, 0 \\ z_2, 0 \\ z_3, 0 \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}, \quad \begin{bmatrix} j_1 = 0 \\ j_2 = 1 \\ j_3 = 0 \end{bmatrix}$$

Classification MLP Backprop⁽²⁾ (Jacobian)

$$\nabla_y E$$

$$\bar{y} = \text{Softmax}(\bar{z}^{(L)}) = \left(\frac{e^{z_i^{(L)}}}{\sum_{u=1}^M e^{z_u^{(L)}}}, \frac{e^{z_2^{(L)}}}{\sum_{u=1}^M e^{z_u^{(L)}}}, \frac{e^{z_3^{(L)}}}{\sum_{u=1}^M e^{z_u^{(L)}}} \right)$$

For example $z_1^{(L)}$ on $y^{(1)}$

$$i=j \quad \frac{dy^{(j)}}{dz_i^{(L)}} = \left(\frac{e^{z_j^{(L)}}}{\sum_{u=1}^M e^{z_u^{(L)}}} \right)' = \underbrace{(e^{z_j^{(L)}})^1}_{(e^{z_1^{(L)}})^1} \underbrace{\frac{1}{\sum_{u=1}^M e^{z_u^{(L)}}}}_{e^{z_1^{(L)}}} + \underbrace{(e^{z_j^{(L)}})}_{e^{z_1^{(L)}}} \underbrace{\left(\frac{-1}{(\sum_{u=1}^M e^{z_u^{(L)}})^2} \cdot e^{z_i^{(L)}} \right)}_{e^{z_1^{(L)}}} = \\ = y^{(j)} + (-y^{(j)} \cdot y^{(i)}) = \boxed{y^{(j)} - (y^{(j)})^2}$$

$z_1^{(L)}$ on $y^{(2)}$

$$i \neq j \quad \frac{dy^{(j)}}{dz_i^{(L)}} = \left(\frac{e^{z_j^{(L)}}}{\sum_{u=1}^M e^{z_u^{(L)}}} \right)' = \underbrace{(e^{z_j^{(L)}})^1}_{0} \underbrace{\frac{1}{\sum_{u=1}^M e^{z_u^{(L)}}}}_{e^{z_1^{(L)}}} + \underbrace{e^{z_j^{(L)}}}_{e^{z_1^{(L)}}} \cdot \underbrace{\left(\frac{-1}{(\sum_{u=1}^M e^{z_u^{(L)}})^2} \cdot e^{z_i^{(L)}} \right)}_{e^{z_1^{(L)}}} = \\ = 0 - y^{(j)} \cdot y_i = \boxed{-y^{(j)} y_i}$$

$$\left(\frac{u}{v} \right)' = (u \cdot v^{-1})' = u' v^{-1} + u \cdot (-1 v^{-2}) \\ (f(g(x)))' = f'(g) \cdot g'(x)$$

$$\text{Jacobian} = \frac{d\bar{y}}{dz^{(L)}} = \left[\left[\frac{d\bar{y}}{dz_1^{(L)}} \right] \left[\frac{d\bar{y}}{dz_2^{(L)}} \right] \dots \right] = \left[\begin{array}{c|c|c|c} \frac{dy_1}{dz_1^{(L)}} & \frac{dy_1}{dz_2^{(L)}} & \dots & \\ \frac{dy_2}{dz_1^{(L)}} & \frac{dy_2}{dz_2^{(L)}} & \dots & \\ \vdots & \vdots & \ddots & \\ \frac{dy_M}{dz_1^{(L)}} & \frac{dy_M}{dz_2^{(L)}} & \dots & \end{array} \right]$$

$$\frac{d\bar{y}}{dz^{(L)}} = \left[\begin{array}{cccc} y^{(1)} - (y^{(1)})^2 & -y^{(1)} y^{(2)} & -y^{(1)} y^{(3)} & \dots \\ -y^{(2)} y^{(1)} & y^{(2)} - (y^{(2)})^2 & -y^{(2)} y^{(3)} & \dots \\ -y^{(3)} y^{(1)} & -y^{(3)} y^{(2)} & y^{(3)} - (y^{(3)})^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ -y^{(M)} y^{(1)} & -y^{(M)} y^{(2)} & -y^{(M)} y^{(3)} & \dots \end{array} \right] \frac{1}{dz^{(L)}}$$

$$\nabla_{z^{(L)}} E = \left(\frac{d\bar{y}}{dz^{(L)}} \right)^T \nabla_y E \quad , \quad \nabla_y E = [0 \ 0 \ \dots \ -\frac{1}{y^{(t_k)}} \ \dots \ 0 \ 0]^T$$

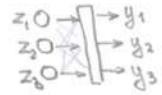
$$\nabla_{z^{(L)}} E = \left[\begin{array}{c} 0 + (-y^{(1)} y^{(t_k)}) \cdot (-\frac{1}{y^{(t_k)}}) = y^{(1)} \\ 0 + (-y^{(2)} y^{(t_k)}) \cdot (-\frac{1}{y^{(t_k)}}) = y^{(2)} \\ 0 + (-y^{(3)} y^{(t_k)}) \cdot (-\frac{1}{y^{(t_k)}}) = -1 + y^{(t_k)} \cdot y^{(t_k)-1} \\ \vdots \\ y^{(M)} \end{array} \right] = \bar{y} - \bar{t}$$

t_k -th row

$$1) -y^{(1)} y^{(t_k)}, \text{ когда } z_i^{(L)}, i \neq t_k$$

$$2) y^{(t_k)} - y^{(t_k)} y^{(t_k)}, \text{ когда } z_i^{(L)}, i = t_k$$

Other rows = 0



Regression MLP Backprop

$$f(\bar{x}) = y(\bar{x}, \bar{w}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

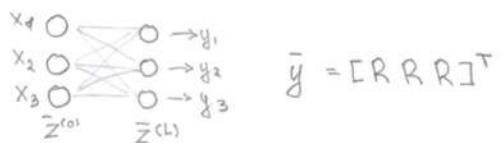
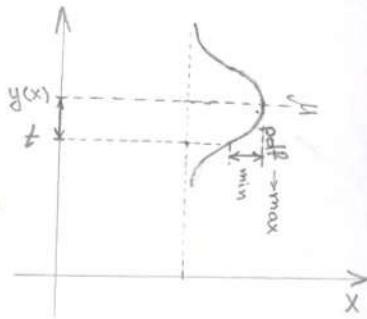
$$P(T|X, W) = \prod_{n=1}^N P(t_n | y(\bar{x}_n, W), \sigma^2) = \prod_{n=1}^N \frac{1}{\sigma \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \left(\frac{t_n - y(\bar{x}_n, W)}{\sigma} \right)^2} \rightarrow \max$$

$$\begin{aligned} \ln P(T|X, W) &= \sum_{n=1}^N \ln \left(\frac{1}{\sigma \sqrt{2\pi}} \right) + \sum_{n=1}^N \left(-\frac{1}{2} \left(\frac{t_n - y(\bar{x}_n, W)}{\sigma} \right)^2 \cdot \ln e \right) = \\ &= \underbrace{N \ln \left(\frac{1}{\sigma \sqrt{2\pi}} \right)}_{\text{const}} - \underbrace{\frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{n=1}^N (t_n - y(\bar{x}_n, W))^2}_{\min} \rightarrow \max \end{aligned}$$

$$E = \frac{1}{2} \sum_{n=1}^N (t_n - y(\bar{x}_n, W))^2 \rightarrow \min$$

for multiple outputs: $\bar{y} \in \mathbb{R}^M$

$$E = \frac{1}{2} \sum_{n=1}^N \sum_{j=1}^M (t_n^{(j)} - y^{(j)}(\bar{x}_n, W))^2 \rightarrow \min$$



Linear activation function: $f(a) = a$

Forward pass

$$\bar{x} = \bar{z}^{(0)}$$

$$\begin{aligned} \bar{z}^{(0)} &\rightarrow \bar{a}^{(1)} \rightarrow \bar{z}^{(1)} \rightarrow \bar{a}^{(2)} \rightarrow \bar{z}^{(2)} \rightarrow \dots \rightarrow \bar{z}^{(L)} \rightarrow y \rightarrow E \\ \bar{y} &= \bar{z}^{(L)} = f(\bar{a}^{(L)}) = f(\bar{b}^{(L)} + W^{(L)} \cdot \bar{z}^{(L-1)}) \end{aligned}$$

Backward pass

$$\begin{array}{ccccccccc} \nabla_E E & \rightarrow & \nabla_y E & \rightarrow & \nabla_{z^{(L)}} E & \rightarrow & \nabla_{a^{(L)}} E & \rightarrow & \nabla_{z^{(L-1)}} E \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \bar{y} & = & \bar{z}^{(L)} & & & & & & \end{array}$$

$$\begin{array}{c} \nabla_{W^{(L)}} E \\ \nabla_{b^{(L)}} E \end{array}$$

$$\begin{array}{c} \nabla_{W^{(L-1)}} E \\ \nabla_{b^{(L-1)}} E \end{array}$$

$$\begin{array}{c} \nabla_{W^{(1)}} E \\ \nabla_{b^{(1)}} E \end{array}$$

$$E = \frac{1}{2} \sum_{n=1}^N \sum_{j=1}^M (t_n^{(j)} - z_j^{(L)})^2 \rightarrow \min$$

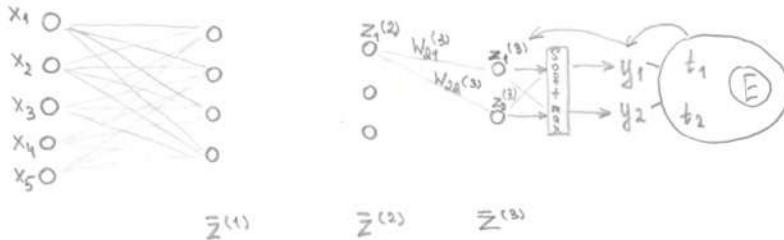
$$\frac{dE}{dz_k^{(L)}} = \frac{1}{2} \sum_{n=1}^N \sum_{j=1}^M 2 \cdot (t_n^{(j)} - z_j^{(L)}) \cdot (t_n^{(j)} - z_j^{(L)})' \Big|_{z_k^{(L)}} = \sum_{n=1}^N \sum_{j=1}^M (z_j^{(L)} - t_n^{(j)}) \mathbf{I}(j=k) = \sum_{n=1}^N (z_k^{(L)} - t_n^{(k)})$$

$\begin{cases} -1, & \text{if } j=k \\ 0, & \text{otherwise} \end{cases}$

$$\nabla_{z^{(L)}} E = \bar{z}^{(L)} - \bar{t}$$

$\Rightarrow \circ \rightarrow z_1^{(L)}$	$\rightarrow f^{(1)}$
$\Rightarrow \circ \rightarrow z_2^{(L)}$	$\rightarrow f^{(2)}$
$\Rightarrow \circ \rightarrow z_3^{(L)}$	$\rightarrow f^{(3)}$

Backprop (1)



$$\bar{z}^{(l)} = \tilde{f}(\bar{a}^{(l)}) = \tilde{f}(b^{(l)} + W^{(l)} \bar{z}^{(l-1)})$$

$$\frac{dE}{da} = \frac{dE}{dy} \cdot \frac{dy}{dz} \cdot \frac{dz}{da} = \frac{dE}{dz} \cdot \underbrace{\frac{dz}{da}}_{\stackrel{f=z}{1}} - \frac{dE}{dz} \cdot \frac{db}{da}$$

$$\nabla_{z^{(l)}} E, \quad \bar{z}^{(l)} = \tilde{f}(\bar{a}^{(l)})$$

$$z_i^{(l)} = \text{ReLU}(a_i^{(l)})$$

$$\frac{dz}{da} = \frac{dz}{df_1} \frac{df_1}{da} + \frac{dz}{df_2} \frac{df_2}{da} \dots \Rightarrow \nabla_{a^{(l)}} \bar{z} = \left[\sum_{i=1}^M \frac{dz}{df_i} \frac{df_i}{da_1} \quad \sum_{i=1}^M \frac{dz}{df_i} \frac{df_i}{da_2} \dots \right]^T, \text{ but } f_2(a_2) \neq f_2(a_1) \\ \Rightarrow f'_2(a_1) = 0$$

$$\nabla_{a^{(l)}} E = \left(\frac{d\bar{f}}{da} \right)^T \cdot \nabla_{z^{(l)}} E$$

$$\nabla_{a^{(l)}} E = \left[\frac{d\bar{f}_1}{da_1} \frac{dE}{dz_1^{(l)}} \quad \frac{d\bar{f}_2}{da_2} \frac{dE}{dz_2^{(l)}} \dots \right]^T$$

$$\frac{d\bar{f}}{da} = \begin{bmatrix} \frac{d\bar{f}_1}{da_1} & \frac{d\bar{f}_1}{da_2} & \dots & \frac{d\bar{f}_1}{da_M} \\ \frac{d\bar{f}_2}{da_1} & \frac{d\bar{f}_2}{da_2} & \dots & \frac{d\bar{f}_2}{da_M} \\ \dots & \dots & \dots & \frac{d\bar{f}_M}{da_M} \end{bmatrix} = \begin{bmatrix} \frac{d\bar{f}_1}{da_1} & 0 & \dots & 0 \\ 0 & \frac{d\bar{f}_2}{da_2} & \dots & 0 \\ \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \frac{d\bar{f}_M}{da_M} \end{bmatrix}$$

$$\left(\frac{d\bar{f}}{da} \right)^T \cdot \nabla_{z^{(l)}} E = \left[\frac{d\bar{f}_1}{da_1^{(l)}} \frac{dE}{dz_1^{(l)}} \quad \frac{d\bar{f}_2}{da_2^{(l)}} \frac{dE}{dz_2^{(l)}} \dots \right]^T$$

$$\nabla_{a^{(l)}} E = \underbrace{\bar{f}'(\bar{a}^{(l)})}_{\text{elementwise}} \circ \nabla_{z^{(l)}} E = \left(\frac{d\bar{f}}{da} \right)^T \nabla_{z^{(l)}} E = \nabla_{a^{(l)}} E \quad - \text{activation functions}$$

$$\frac{dE}{db_i^{(l)}} = \frac{dE}{dz_i^{(l)}} \cdot \frac{dz_i^{(l)}}{da_i^{(l)}} \cdot \frac{da_i^{(l)}}{db_i^{(l)}} = \frac{dE}{da_i^{(l)}} \cdot \frac{da_i^{(l)}}{db_i^{(l)}} = \frac{dE}{da_i^{(l)}} \quad a(b) = b + Wz^{(l-1)} \\ a'(b) = (b)_b^T = 1$$

$$\nabla_{b^{(l)}} E = \left[\frac{dE}{da_1^{(l)}} \quad \frac{dE}{da_2^{(l)}} \quad \frac{dE}{da_3^{(l)}} \dots \right]^T = \nabla_{a^{(l)}} E$$

$$\nabla_{z^{(l)}} E = \nabla_{a^{(l)}} E$$

$\nabla_{z^{(l)}} E$ - loss function

$\nabla_{a^{(l)}} E$ - activation function

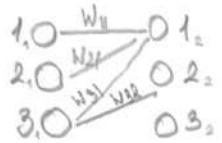
$$\bar{z} = \tilde{f}(\bar{a}) = \tilde{f}(b + Wz^{(l-1)})$$

$$\frac{dE}{da} = \frac{dE}{dz} \cdot \underbrace{\frac{dz}{da}}_{\stackrel{f=z}{1}} \rightarrow \frac{dE}{db} = \frac{dE}{dz} \frac{dz}{da} \cdot \frac{da}{db} = \boxed{\frac{dE}{dz} \cdot f'(a) \cdot a'(b)}$$

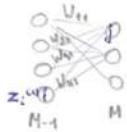
Backprop (2)

$$\bar{z}^{(l)} = \bar{f}(\bar{a}^{(l)}) = \bar{f}(\bar{b}^{(l)} + W^{(l)} \bar{z}^{(l+1)})$$

$$W^{(l)} = \begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41} \\ w_{12} & w_{22} & w_{32} & w_{42} \\ \dots & \dots & \dots & \dots \\ w_{1M} & w_{2M} & \dots & w_{NM} \end{bmatrix}$$



$$\nabla_{z^{(l)}} E, \nabla_{a^{(l)}} E, \nabla_{b^{(l)}} E$$



$$\bar{z}^{(l-1)} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}$$

$$a_i^{(l)} = b_i^{(l)} + \sum_{j=1}^{N-1} w_{ij} \cdot z_j^{(l-1)}$$

$$a_j(w_i) = b_j + \sum_{m=1}^{N-1} w_{im} \cdot z_m^{(l-1)}, \quad a^t(w_{i,j}) = 0 + (0 + (w_{i,j} \cdot z_i^{(l-1)})^t) = z_i^{(l-1)}$$

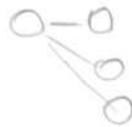
$$\frac{dE}{dw_{ij}} = \frac{dE}{da_j^{(l)}} \cdot \frac{da_j^{(l)}}{d w_{ij}} = \frac{dE}{da_j^{(l)}} \cdot z_i^{(l-1)}$$

$$\nabla_{W^{(l)}} E = \begin{bmatrix} \frac{dE}{d w_{11}} & \frac{dE}{d w_{12}} & \dots & \frac{dE}{d w_{1M}} \\ \frac{dE}{d w_{21}} & \frac{dE}{d w_{22}} & \dots & \frac{dE}{d w_{2M}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dE}{d w_{M1}} & \frac{dE}{d w_{M2}} & \dots & \frac{dE}{d w_{MM}} \end{bmatrix} = \begin{bmatrix} \frac{dE}{da_1^{(l)}} \frac{da_1^{(l)}}{d w_{11}} & \frac{dE}{da_1^{(l)}} \frac{da_1^{(l)}}{d w_{12}} & \dots & \frac{dE}{da_1^{(l)}} \frac{da_1^{(l)}}{d w_{1M}} \\ \frac{dE}{da_2^{(l)}} \frac{da_2^{(l)}}{d w_{21}} & \frac{dE}{da_2^{(l)}} \frac{da_2^{(l)}}{d w_{22}} & \dots & \frac{dE}{da_2^{(l)}} \frac{da_2^{(l)}}{d w_{2M}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dE}{da_N^{(l)}} \frac{da_N^{(l)}}{d w_{M1}} & \frac{dE}{da_N^{(l)}} \frac{da_N^{(l)}}{d w_{M2}} & \dots & \frac{dE}{da_N^{(l)}} \frac{da_N^{(l)}}{d w_{MM}} \end{bmatrix} = \begin{bmatrix} \frac{dE}{da_1^{(l)}} \cdot z_1^{(l-1)} & \frac{dE}{da_1^{(l)}} \cdot z_2^{(l-1)} \\ \frac{dE}{da_2^{(l)}} \cdot z_1^{(l-1)} & \frac{dE}{da_2^{(l)}} \cdot z_2^{(l-1)} \\ \vdots & \vdots \\ \frac{dE}{da_N^{(l)}} \cdot z_1^{(l-1)} & \frac{dE}{da_N^{(l)}} \cdot z_2^{(l-1)} \end{bmatrix}$$

$$\nabla_{W^{(l)}} E = \nabla_{a^{(l)}} E \cdot (\bar{z}^{(l)})^T = \begin{bmatrix} \frac{dE}{da_1^{(l)}} \\ \frac{dE}{da_2^{(l)}} \\ \vdots \\ \frac{dE}{da_N^{(l)}} \end{bmatrix} [z_1^{(l-1)} \ z_2^{(l-1)}]$$

$$\nabla_{z^{(l-1)}} E = ?$$

$$\begin{aligned} \frac{dE}{dz_1^{(l-1)}} &= \frac{dE}{da_1^{(l)}} \frac{da_1^{(l)}}{dz_1^{(l-1)}} + \frac{dE}{da_2^{(l)}} \frac{da_2^{(l)}}{dz_1^{(l-1)}} + \dots \\ &= w_{11} + w_{12} + w_{13} \dots \end{aligned}$$



$$\nabla_{z^{(l-1)}} E = W_l^T \cdot \nabla_{a^{(l)}} E$$

Forward

$$\bar{z}^{(0)} \rightarrow \bar{a}^{(1)} \rightarrow \bar{z}^{(1)} \rightarrow \dots \rightarrow \bar{z}^{(l-1)} \rightarrow \bar{a}^{(l)} \rightarrow \bar{z}^{(l)} \rightarrow y \rightarrow E$$

Backward

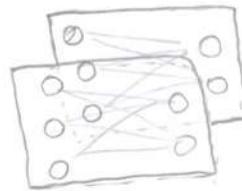
$$\nabla_E \rightarrow \nabla_y E \rightarrow \nabla_{\bar{z}^{(l)}} E \rightarrow \nabla_{a^{(l)}} E \rightarrow \nabla_{\bar{a}^{(l)}} E \rightarrow \dots \rightarrow \nabla_{\bar{z}^{(1)}} E \rightarrow \nabla_{a^{(1)}} E$$

$$\begin{array}{c} \searrow \\ \nabla_{W^{(l)}} E \\ \searrow \\ \nabla_{b^{(l)}} E \end{array}$$

$$\begin{array}{c} \nearrow \\ \nabla_{W^{(1)}} E \\ \nearrow \\ \nabla_{b^{(1)}} E \end{array}$$

$$W \cdot \vec{v}$$

$$\begin{bmatrix} \textcircled{1} & \textcircled{2} & \textcircled{3} \\ \textcircled{4} & \textcircled{5} & \textcircled{6} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} ? \\ ? \end{bmatrix}$$



$$(W \cdot \vec{v})_V^I = W^T \cdot \nabla_{\vec{v}} \quad \begin{bmatrix} W^T \\ \nabla_{\vec{v}} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

$$W \cdot V = \underbrace{\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}}_k^{m \times k}$$

$$\begin{bmatrix} W & V \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}}_{\nabla}^{2 \times 3}$$

$$(W \cdot V)_V^I = W^T \nabla_{\vec{v}}, \text{ where } \nabla_{\vec{v}}^{2 \times 3}$$

$$(W \times)_W^I = \nabla \cdot \vec{x}^T$$

$$\begin{bmatrix} \textcircled{1} & \textcircled{2} & \textcircled{3} \\ \textcircled{4} & \textcircled{5} & \textcircled{6} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \otimes \\ \otimes \\ \otimes \end{bmatrix}^{2 \times 1}$$

$$\begin{bmatrix} \otimes \\ \otimes \end{bmatrix}^{2 \times 1} \otimes \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}^{1 \times 3} = \begin{bmatrix} \otimes & \times \\ \otimes & \times \end{bmatrix}^{2 \times 3}$$

$$(W \times)_W^I = \nabla \cdot \vec{x}^T$$

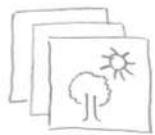
$$\begin{bmatrix} W \\ \nabla \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} \textcircled{1} & \textcircled{2} \\ \textcircled{3} & \textcircled{4} \end{bmatrix}^{3 \times 2} = \underbrace{\begin{bmatrix} \textcircled{1} & \textcircled{2} \\ \textcircled{3} & \textcircled{4} \end{bmatrix}}_{\nabla}^{2 \times 2}$$

$$\begin{bmatrix} \textcircled{1} & \textcircled{2} \\ \textcircled{3} & \textcircled{4} \end{bmatrix} \begin{bmatrix} \textcircled{1} & \textcircled{2} \\ \textcircled{3} & \textcircled{4} \end{bmatrix}^{X^T} \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}^{3 \times 1} = \begin{bmatrix} \textcircled{1} & \textcircled{2} \\ \textcircled{3} & \textcircled{4} \end{bmatrix}^{X \times X}$$

$$W \cdot V^I = W^T \cdot \nabla$$

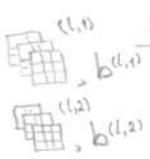
$$W^I \cdot V = \nabla \cdot V^T$$

CNN



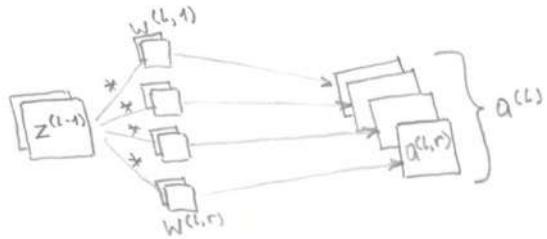
Input Tensor

$$Z^{(L-1)} \in \mathbb{R}^{m_{L-1} \times n_{L-1} \times r_{L-1}}$$



Convolution Filters

$$w^{(l,r)} \in \mathbb{R}^{\tilde{m}_l \times \tilde{n}_l \times r_l}, r = 1, 2, \dots, r_l$$



Activation maps

$$a^{(l,r)} \in \mathbb{R}^{m_l \times n_l \times 1}, r = 1, 2, \dots, r_l$$

$$a^{(l)} \in \mathbb{R}^{m_l \times n_l \times r_l}$$

$$Z^{(L-1)} * w^{(l)} = a^{(l)}$$

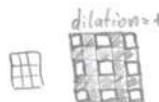
Feature maps ($Z^{(l,r)} = f(a^{(l,r)})$)

$$Z^{(l,r)} \in \mathbb{R}^{m_l \times n_l}$$

$$Z^{(l)} \in \mathbb{R}^{m_l \times n_l \times r_l} \text{ - output tensor}$$

Stride - waż

dilation - brzegi przekształcane na komórki



Convolution actually Cross-Correlation (no flip)

$$\text{Image } (I)^{H \times W}$$

Kernel $(K)^{M \times N}$ - convolution filter

$I[i,j]$ - element

$K[m,n]$ - element

$$(I * K)_{[i,j]} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I[i+m, j+n] \cdot K[m, n] + b$$

elementwise

Generally

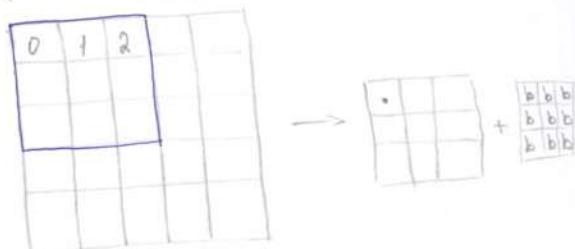
$$\square a^{(l,r)} = Z^{(l-1)} * w^{(l,r)} + b^{(l,r)}, r = 1, 2, \dots, r_l$$

$$\square a^{(l)} = Z^{(l-1)} * w^{(l)} + b^{(l)}$$

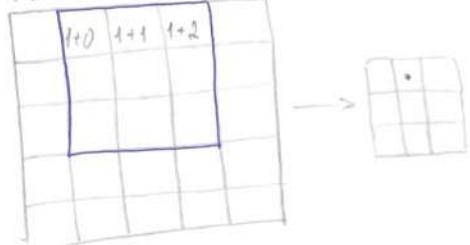
$$Z^{(l)} = f(a^{(l)}), f - \text{ReLU}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \xrightarrow{f} \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}$$

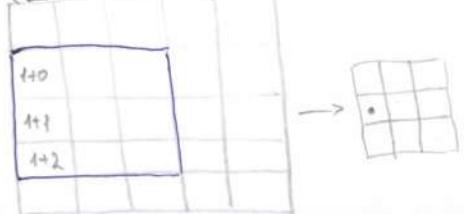
$$(I * K)_{[0,0]}$$



$$(I * K)_{[0,1]}$$

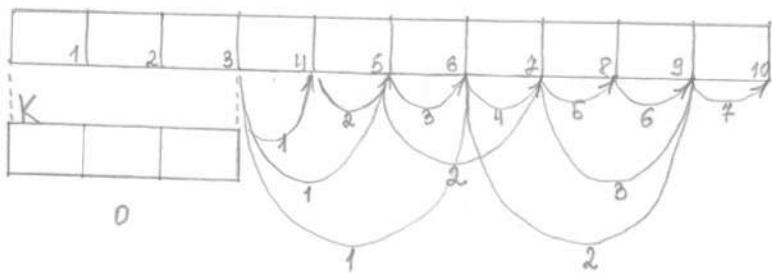


$$(I * K)_{[1,0]}$$



CNN. Output shape

W

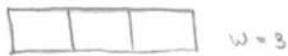


2p - padding

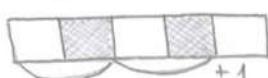
d - dilation , default = 1

Dilation, как расширение языка

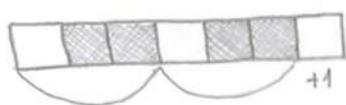
$$K=3: d=1$$



$$d=2$$

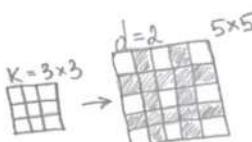


d-3



$$\boxed{d(K-1) + 1}$$

K = 2

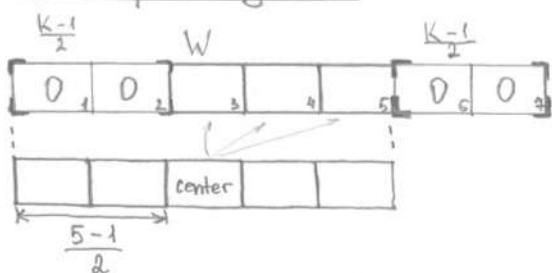


$$\boxed{\quad} \quad \boxed{\quad} \quad \boxed{\quad} = \frac{1}{\alpha}$$



$$1_r) = 4$$

Zero padding

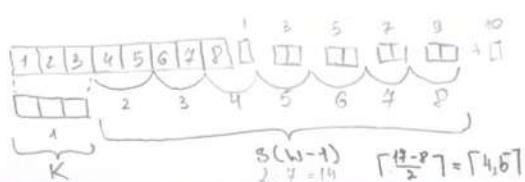


$$K = 2n+1 \in \{3, 5, 7, 9, \dots\}$$

Чтобы нарисовать $W_{out} = h$ нейроно-
заначий $\frac{K-1}{2}$ нейрона по разные:

$$P = \frac{K-1}{2}$$

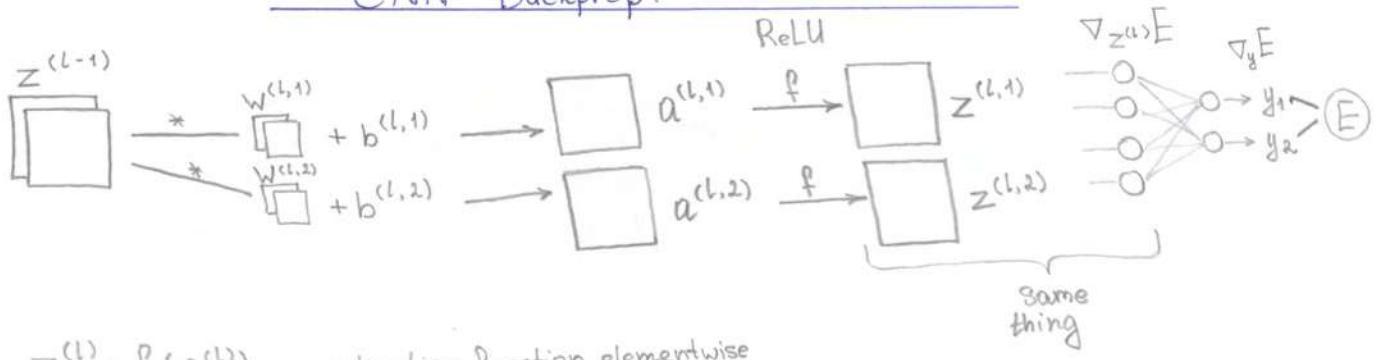
$$P = \left\lceil \frac{S(W-1) - W + K}{2} \right\rceil$$



В общем законе $\delta_{\text{БЛ}} K + 3(w-1)$

$$\lceil \frac{2-4}{3} \rceil = \lceil 1.5 \rceil = 2$$

CNN Backprop.



$$z^{(l)} = f(a^{(l)}) \quad - \text{activation function elementwise}$$

$$\nabla_{a^{(l)}} E = \left(\frac{d\bar{f}}{d\bar{a}^{(l)}} \right)^T \cdot \nabla_{z^{(l)}} E \quad \frac{dE}{da} = \frac{dE}{dz} \cdot \underbrace{\frac{dz}{df}}_{\bar{f}} \cdot \frac{df}{da}$$

$$\begin{bmatrix} \frac{df_1}{da_1} & \frac{df_2}{da_1} & \dots \\ \frac{df_1}{da_2} & \frac{df_2}{da_2} & \dots \\ \dots & \dots & \dots \end{bmatrix} = \begin{bmatrix} [\bar{f}'(a^{(l,1)})] & 0 & 0 \\ 0 & [\bar{f}'(a^{(l,2)})] & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

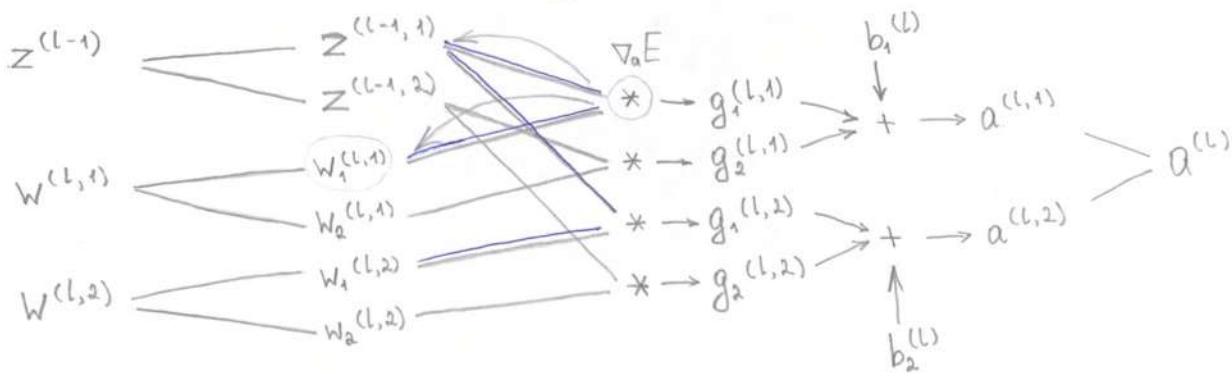
ausgeschaut, was wir für zurückfließende $a^{(l)}$ brauchen

$$\nabla_{a^{(l)}} E = f'(a^{(l)}) \circ \nabla_{z^{(l)}} E$$

elementwise

$$\begin{aligned} & \begin{pmatrix} f(a_{11}^{(l,1)}) & f(a_{12}^{(l,1)}) & \dots \\ f(a_{21}^{(l,1)}) & f(a_{22}^{(l,1)}) & \dots \\ \dots & \dots & \dots \end{pmatrix} = \\ & = \begin{bmatrix} \frac{dE}{dz^{(l,1)}} \cdot \frac{df}{da_{11}^{(l,1)}} & \dots \\ \frac{dE}{dz^{(l,1)}} \cdot \frac{df}{da_{21}^{(l,1)}} & \dots \end{bmatrix} \end{aligned}$$

Backprop through convolutions $(Z * w)_w$



$$g_k^{(l,r)} = z^{(l-1,k)} * w_k^{(l,r)}$$

$$a^{(l,r)} = \sum_k g_k^{(l,r)} + b_r^{(l)}$$

$$a = \sum_k g_k = \sum (z + w) \Rightarrow \frac{dE}{dw} = \frac{dE}{da} \cdot \frac{da}{dw}$$

$$\begin{array}{c} k+1 \\ \hline \end{array} \quad \begin{array}{c} k+1 \\ \hline k+2 \\ \hline \end{array} \quad \begin{array}{c} k+1 \\ \hline k+2 \\ \hline \end{array} = \boxed{} \quad a^{(l,r)}$$

$\nabla_{W_1(u,v)} E = ?$

$$\begin{matrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{matrix} \times \begin{matrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{matrix} = \begin{matrix} (a_{11} & a_{12} \\ a_{21} & a_{22}) \end{matrix}$$

The diagram illustrates the decomposition of W_M into four components: A_{11} , A_{12} , A_{21} , and A_{22} . The relationships are represented by arrows originating from W_M and pointing to each component, with intermediate variables Z_{11} , Z_{12} , Z_{21} , and Z_{22} positioned along the paths.

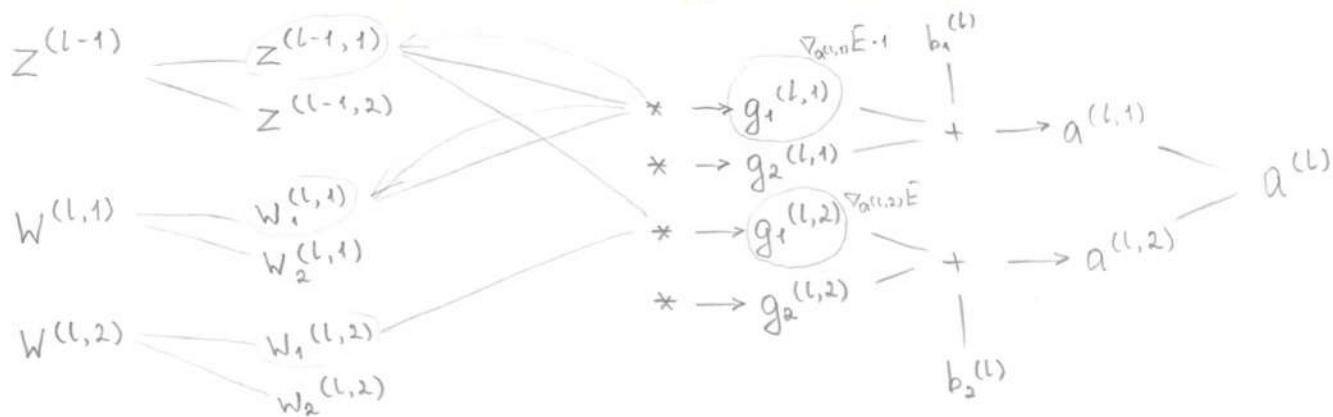
$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix}_{3 \times 3} \times \begin{bmatrix} \frac{\partial E}{\partial a_{11}} & \frac{\partial E}{\partial a_{12}} \\ \frac{\partial E}{\partial a_{21}} & \frac{\partial E}{\partial a_{22}} \end{bmatrix}_{2 \times 2} = \begin{bmatrix} \frac{\partial E}{\partial w_{11}} & \frac{\partial E}{\partial w_{12}} \\ \frac{\partial E}{\partial w_{21}} & \frac{\partial E}{\partial w_{22}} \end{bmatrix}_{2 \times 2}$$

$$Z^{(l-1)} * w^{(l,r)} \quad Z^{(l-1,k)} * \nabla_{w_k^{(l,r)}} E = \nabla_{w_k^{(l,r)}} E$$

$$\underbrace{\nabla_{W_k(l,r)}}_{\text{by channel}} E = Z^{(l-1,k)} * \nabla_{a(l,r)} E, \text{ gaivame nurojmo ciavumis b menzopz}$$

$$\nabla_{W^{(l,n)}} E = Z^{(l-1)} * \nabla_{a^{(l,n)}} E$$

Backprop through Convolution $(Z * w)_z'$



$$Z^{(l-1)} * W^{(l)} + b^{(l)}, f, f = 3 \times 3, m_1 \times n_1$$

$$\begin{cases} Z^{(l-1,1)} * W_1^{(l,1)} = g_1^{(l,1)} \\ Z^{(l-1,2)} * W_1^{(l,2)} = g_1^{(l,2)} \end{cases} \quad a' = g'$$

$$\nabla_{Z^{(l-1,1)}} E - ?$$

$$\tilde{W}_k^{(l,r)} = W_k^{(l,r)} \cdot \text{flip}[1]$$

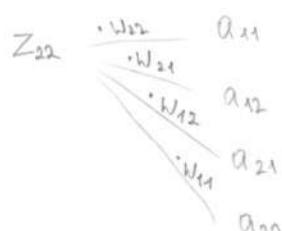
$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & \textcircled{Z}_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} * \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

1 channel

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \nabla a_{11} & \nabla a_{12} & 0 \\ 0 & \nabla a_{21} & \nabla a_{22} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} W_{22} & W_{21} \\ W_{12} & W_{11} \end{bmatrix}$$

$\xrightarrow{k=2}$ $P = \frac{W}{2}$

$$\sim \nabla_{Z^{(l-1,k)}} E = \text{ZeroPad}(\nabla_{a^{(l,r)}} E) * \tilde{W}_k^{(l,r)}$$

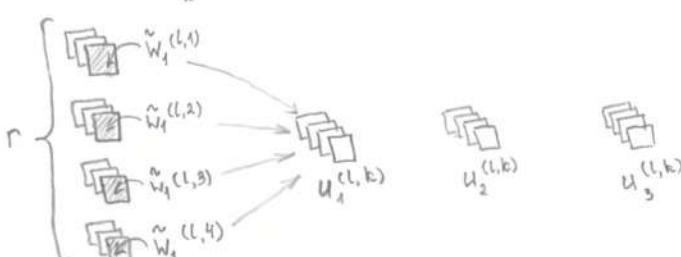


$$\Rightarrow \frac{\partial E}{\partial Z_{22}} = \frac{\partial E}{\partial a_{11}} \cdot W_{22} + \frac{\partial E}{\partial a_{12}} \cdot W_{21} + \frac{\partial E}{\partial a_{21}} \cdot W_{23} + \frac{\partial E}{\partial a_{22}} \cdot W_{12} + \frac{\partial E}{\partial a_{23}} \cdot W_{11}$$

$\nabla_a E \cdot \tilde{W}$ like

$$\boxed{\nabla_{Z^{(l-1,k)}} E = \sum_r \text{ZeroPad}(\nabla_{a^{(l,r)}} E) * \tilde{W}_k^{(l,r)}} = \text{ZeroPad}(\nabla_{a^{(l,r)}} E) \cdot \underbrace{U^{(l,k)}}_{[W_1^{(l,1)} \ W_1^{(l,2)} \dots \ W_1^{(l,r)}]}$$

$$U_r^{(l,k)} = \tilde{W}_k^{(l,r)}$$

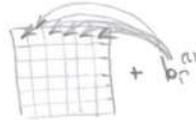


Это интересно бывает на практике использовать `PyTorch` в conjunction with `AutoGrad`.
И если базис `ZeroPad` $\nabla_a E * \tilde{W} = (Z * W)_z'$

Backprop through convolution $(Z * W + b \cdot 1)_b^l$

$$\begin{bmatrix} Z_1^{(l-1)} \\ \vdots \\ Z^{(l-1)} \end{bmatrix} * \begin{bmatrix} W^{(l,r)} \\ \vdots \\ b_r^{(l)} \end{bmatrix} = \begin{bmatrix} a_r^{(l)} \end{bmatrix}$$

$$a_r^{(l,r)} = Z^{(l-1)} * W^{(l,r)} + b_r^{(l)}$$

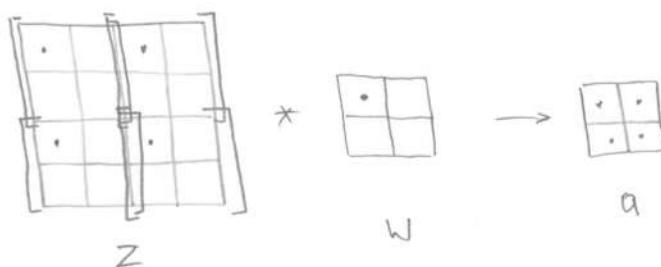


$$\frac{dE}{db_r^{(l)}} = \frac{dE}{da_{11}^{(l)}} \cdot 1 + \frac{dE}{da_{12}^{(l)}} \cdot 1 + \dots$$

$$\Rightarrow \nabla_{b^{(l,r)}} E = \underset{\substack{\uparrow \\ \text{elements of arg.}}}{\text{Sum}} (\nabla_{a^{(l,r)}} E), \quad r = 1, \dots, r^l$$

Backprop through convolution with stride

1)



$$\text{Dilate}(a) = \begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$$

$$\nabla_w E = Z^{(l-1)} * \text{Dilate}(\nabla_a E)$$

2)

$$\begin{matrix} Z_{11} & Z_{12} & Z_{13} & Z_{14} & Z_{15} \\ Z_{21} & Z_{22} & Z_{23} & Z_{24} & Z_{25} \\ Z_{31} & Z_{32} & Z_{33} & Z_{34} & Z_{35} \\ Z_{41} & Z_{42} & Z_{43} & Z_{44} & Z_{45} \\ Z_{51} & Z_{52} & Z_{53} & Z_{54} & Z_{55} \end{matrix} * \begin{matrix} W_{11}^4 & W_{12}^4 & W_{13}^3 \\ W_{21}^1 & W_{22}^2 & W_{23}^2 \\ W_{31}^2 & W_{32}^1 & W_{33}^1 \end{matrix} = \begin{matrix} a_{11}^1 & a_{12}^2 \\ a_{21}^3 & a_{22}^4 \end{matrix}$$

$$\begin{matrix} Z_{33} & \cdot w_{33} & a_{11} \\ & \cdot w_{31} & a_{12} \\ & \cdot w_{43} & a_{21} \\ & \cdot w_{11} & a_{22} \end{matrix}$$

$$\frac{dE}{dZ_{33}} = \underbrace{\frac{dE}{da_{11}} \cdot w_{33}} + \underbrace{\frac{dE}{da_{12}} \cdot w_{31}} + \underbrace{\frac{dE}{da_{21}} \cdot w_{43}} + \underbrace{\frac{dE}{da_{22}} \cdot w_{11}}$$

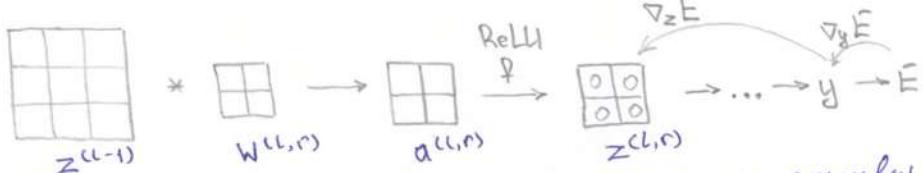
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{dE}{da_{11}} \end{bmatrix} * \begin{matrix} W_{33} & W_{32} & W_{31} \\ W_{23} & W_{22} & W_{21} \\ W_{13} & W_{12} & W_{11} \end{matrix} = \nabla_Z E$$

\tilde{w}

$$\text{ZeroPad}(\text{Dilate}_{S=1}(\nabla_a E))$$

CNN Learnability

1 Channel



Схематичное описание обработка матрицы активированных feature map.

$$\nabla_{z^{(l,r)}} E \text{ not zero} \quad \begin{array}{c} E=y+t \\ y=0, t=1 \\ \text{GMP} \end{array}$$

$$\nabla_{z^{(l,r)}} E = \nabla_y E$$

$$\nabla_{a^{(l,r)}} E$$

$$\text{ReLU}'(a^{(l,r)}) = \begin{cases} 1, & \text{if } a > 0 \\ 0, & \text{if } a \leq 0 \end{cases}$$

$$Z^{(l-1)} * W^{(l,r)} = a^{(l,r)}$$

$$\nabla_{W^{(l,r)}} E$$

$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} * \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Бес беса блоком на новый шаги

$$\begin{array}{c} \frac{\partial E}{\partial W_{11}} = \frac{\partial E}{\partial Z_{11}} \frac{\partial Z_{11}}{\partial a_{11}} + \frac{\partial E}{\partial Z_{21}} \frac{\partial Z_{21}}{\partial a_{11}} + \dots \\ \frac{\partial E}{\partial W_{11}} = \frac{\partial E}{\partial a_{11}} \end{array}$$

$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} * \begin{bmatrix} \frac{\partial E}{\partial a_{11}} & \frac{\partial E}{\partial a_{12}} \\ \frac{\partial E}{\partial a_{21}} & \frac{\partial E}{\partial a_{22}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial W_{11}} & \frac{\partial E}{\partial W_{12}} \\ \frac{\partial E}{\partial W_{21}} & \frac{\partial E}{\partial W_{22}} \end{bmatrix}$$

ReLU'

Если $\nabla_a E = 0$, то схема не обрабатывается.

Если оно оно значение активировано $\rightarrow \frac{\partial E}{\partial a_{ij}} > 0$ то схема обрабатывается.

$$\nabla_{z^{(l-1)}} E = \text{ZeroPad}(\nabla_{a^{(l,r)}} E) * W^{(l,r)}$$

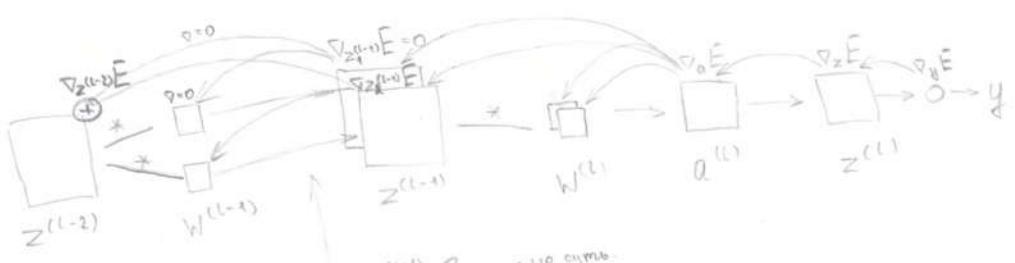
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\partial E}{\partial a_{11}} & \frac{\partial E}{\partial a_{12}} & 0 \\ 0 & \frac{\partial E}{\partial a_{21}} & \frac{\partial E}{\partial a_{22}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} W_{22} & W_{21} \\ W_{12} & W_{11} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial Z_{11}} & \frac{\partial E}{\partial Z_{12}} & \frac{\partial E}{\partial Z_{13}} \\ \frac{\partial E}{\partial Z_{21}} & \frac{\partial E}{\partial Z_{22}} & \frac{\partial E}{\partial Z_{23}} \\ \frac{\partial E}{\partial Z_{31}} & \frac{\partial E}{\partial Z_{32}} & \frac{\partial E}{\partial Z_{33}} \end{bmatrix}$$

Здесь снова если оно оно значение активировано, то немедленно предыдущие слои будут обрабатываться. (?)

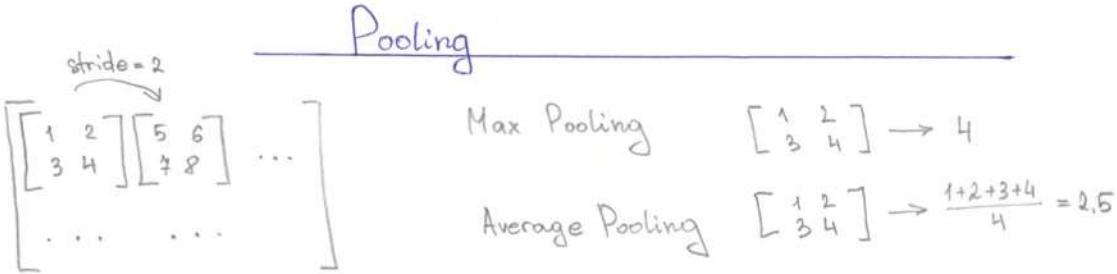
$\nabla z^{(l-1)}$ может заменяться на каналы, но он тоже будет суммироваться с зеркальным образом каналов и обрабатываться.

$$(a^{(l-1,r)})' = \underbrace{\frac{\partial E}{\partial z^{(l-1)}}}_{\nabla_{z^{(l-1)}} E} \cdot \text{ReLU}'(a^{(l-1,r)})$$

$$\nabla_{z^{(l-1)}} E$$



Если мы a^{(l-1)} глобальные, то не суммируются.



stride = kernel size

Global Pooling

kernel size = input tensor size

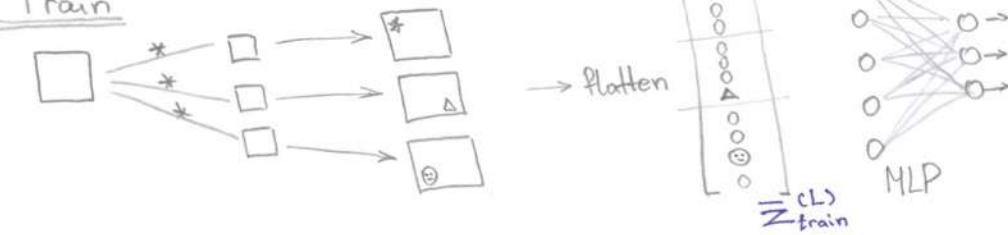
$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

Global Max Pooling $\stackrel{(GMP)}{\rightarrow} 8$

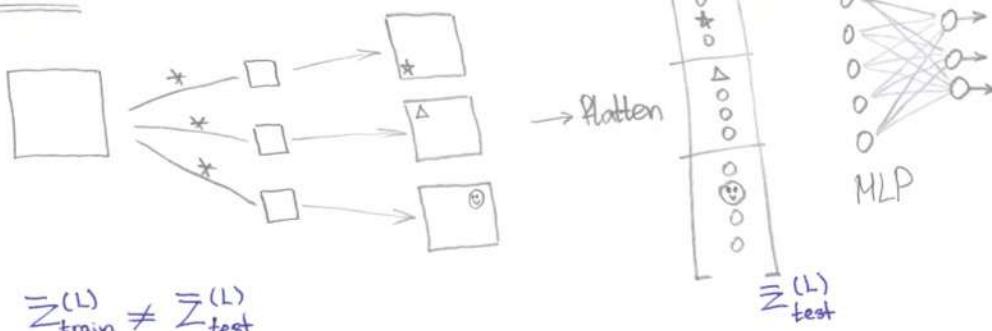
Global Average Pooling $\stackrel{(GAP)}{\rightarrow} \frac{8}{16}$

Why to use GMP

Train



Test

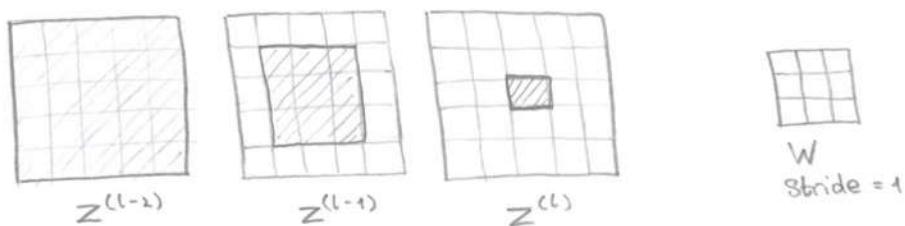


CNN обрабатывает, использует одинаковые feature maps,
но если использовать flatten, то мы имеем один проблему, как Image \rightarrow MLP.

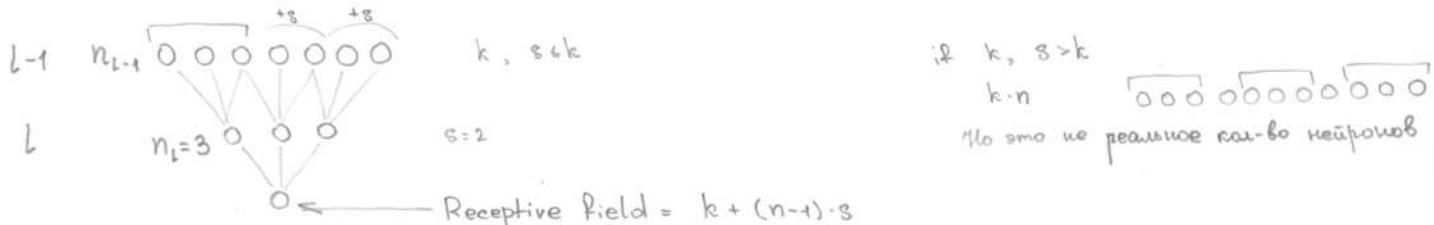
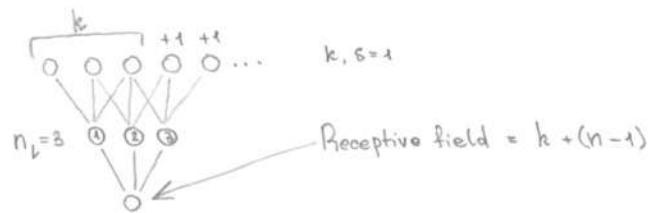
→ Fully Convolutional Neural Network

Receptive Field

$1 \rightarrow 3 \rightarrow 5$

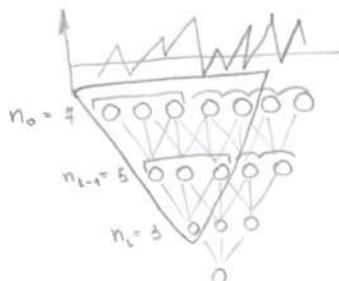


Receptive field - это множество пикселей изображения, от которых зависит активация определенного нейрона.



$$n_{L-1} = k + (n_L - 1) \cdot s$$

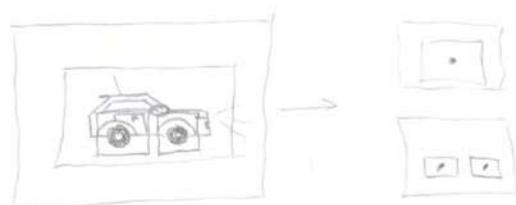
n_L - количество нейронов $L-1$ -го слоя, будущий нейрон



Dilated convolution
увеличиваём Receptive field
но...

знач receptive field нейрона,
может ли это определять cf
сигнального слоя за $O(1)$? Кажется, нет

В зависимости от Receptive field нейрон может распознавать
какие-то определенного размера, или относительное расположение параметра.



CTC for OCR



1. Classify 1M words. Problems: 1) if not in Domain, we can't classify, 2) A lot of images per each word

2. Slice into letters. Problems: 1) Hard task, 2) Annotation for each letter

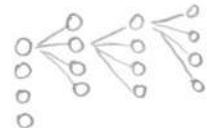
3. Seq-to-seq. Problem: frame-wise alignment

End-to-end solution - Connectionist Temporal Classification

Уговаривая газа Speech Recognition, Audio → Text
u RNN

Мы используем Fully CNN

- L - конечное алфавит ($ABC, \dots, ?$ etc.)
- $\mathcal{X} = L^*$ - множество последовательностей символов L
 $\{A\}, \{B\}, \{C\}, \{AB\}, \{BA\}, \{ABC\}, \{ACB\}, \{BAC\}, \{BCA\}, \{CAB\}, \{CBA\}$
Но если можно рассматривать, как слова.
- $\mathcal{X} = (R^m)^*$ - множество последовательностей m -мерных векторов.
 $\{[1], [1], [1], [1], \dots\}$
Но если это изображение $\in R^{m \times n}$, высотой m и произвольной шириной.
- S - множество обучающих примеров из распределения $P_{x, z}$
 $f(x) \rightarrow \mathcal{X}$
 $f((R^m)^*) \rightarrow L^*$
 $f(\text{Image}) \rightarrow L^*$



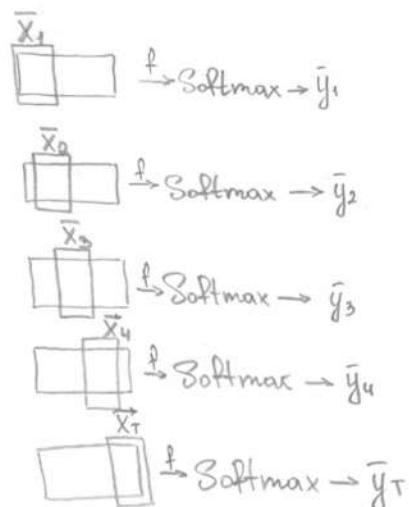
$x = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \dots, \bar{x}_T)$ - Input sequence

$z = (z_1, z_2, z_3, \dots, z_u)$ - Target sequence

$u \leq T$

$(x, z) \in S, x \in \mathcal{X}, z \in \mathcal{Z}$

$f(x) \rightarrow z$



T - кол-во шагов

- W W - - O O O - R - D D D D -

$Z = \text{WORD}$

Training a Classifier

L - alphabet

$\Sigma = L^*$ - sequence of letters

$\mathcal{X} = (R^n)^*$ - sequence of image parts

S - ~~mnogoecmbo~~ $D_{\mathcal{X} \times \Sigma}$

$x = (\bar{x}_1, \dots, \bar{x}_T)$ - input sequence (image)

$z = (z_1, \dots, z_u)$ - target sequence (word)

$u \leq T$

$(x, z) \in S, x \in \mathcal{X}, z \in \Sigma$

Train a classifier on S dataset

$$h : \mathcal{X} \rightarrow \Sigma$$

images words

Label Error Rate наиболее важную
(LER)

S' - validation dataset

$$\text{LER}(h, S') = \frac{1}{|S'|} \sum_{(x, z) \in S'} \text{ED}(h(x), z) \rightarrow \min$$

$\text{ED}(a, b)$ - минимальное кол-во операций вставки, удаления и замены, необходимых для нарушения последовательности b из a .

$$\text{ED}(a = \text{bankt}, b = \text{bank}) = 1$$

ED - Euclidean Distance?

Levenshtein Distance?

CTC

$L' = L \cup \{\text{blank}\}$, blank - нейтральный символ, межбуквенное пространство

L'^* - множество последовательностей.

i.g. $I = (l_1, l_2, l_3, \dots, l_n)$ - target последовательность, $I \in L^*$

$I' = (-, l_1, -, l_2, -, l_3, -, \dots, -, l_n, -)$, $I' \in L'^*$

$$|I'| = 2 \cdot |I| + 1$$

CNN $\rightarrow - C - N - N -$

$\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \dots, \bar{x}_T)$ - input sequence

$h(\bar{x}_t) \xrightarrow[\text{Softmax}]{\text{Softmax}} R^{L'}$ - i.e. предсказание на каждую Sybby алфавита + blank

$y_{\pi_t}^t$ π_t - symbol from L'

Для каждого из t Softmax-ов y нас нет информации о labelах, y нас нет разницы по Sybbam.

Assumption (strong)

$$P(\pi | x) = \prod_{t=1}^T y_{\pi_t}^t, \forall \pi \in L'^T$$

вероятности
символа π_t на шаге

$$\begin{bmatrix} \vec{y}^1 \\ y_1^1 \\ y_2^1 \\ y_3^1 \\ y_4^1 \\ \vdots \end{bmatrix}, \begin{bmatrix} \vec{y}^2 \\ y_1^2 \\ y_2^2 \\ y_3^2 \\ y_4^2 \\ \vdots \end{bmatrix}$$

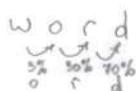
$$\bar{\pi} = [\pi_1, \pi_2, \pi_3, \dots]^T$$

Symbol
number

- Наш Sybby y $y_{\pi_t}^t$ имеет значение верхний и нижний индекс несущими, \vec{y}^t - вектор на шаге t

$y_{\pi_t}^t$ - конкретное значение вектора

- Прямоугольные последовательности π - это произведение уверенности вхождения в символах текущих шагов, которое она предсказываем. Наш Independent Events, но это концепция Dependent



Но, такие образы мы можем легко распознавать слова с ошибками или фрагментами

CTC

$$L' = L \cup \{\text{blank}\} - \text{alphabet} \quad \bar{L} = L'^*$$

L'^* - посредством символов $\text{Symbol} + \{\text{blank}\}$

$\bar{I} \in L^*$, for example $\bar{I} = \text{hello}_5$

$$\bar{I}' \in L'^* \quad \bar{I}' = -h-e-l-l-o-, |\bar{I}'| = 2|\bar{I}|+1$$

$$x = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \dots, \bar{x}_T), T - \text{width} \quad x \in \mathcal{X}$$

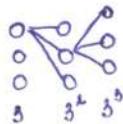
$h(\bar{x}_t) \rightarrow R^{|\bar{L}'|}$ вероятность на каждого символа $\in L'$

$$h(\bar{x}_t) = \bar{y}^t$$

path

$$\bar{\pi} = [\pi_1, \pi_2, \pi_3, \dots, \pi_T]^T, \pi_t - \text{symbol}$$

$$p(\bar{\pi} | x) = \prod_{t=1}^T p_{\pi_t}^t, \forall \pi_t \in L^T$$



$$\bar{L}^T \in L'^*$$

↑
посред. символ T

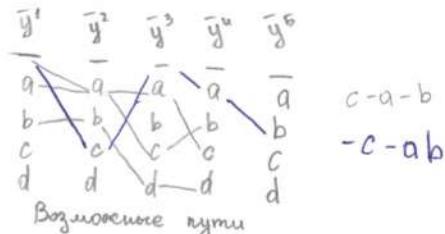
Decoding

$$B: \bar{L} \rightarrow \mathcal{X}$$

1. Удаление небольшой: AAA → A

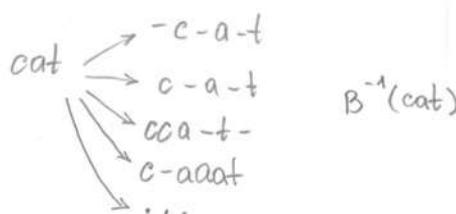
2. Удаление blanks: -AA-A → AAA

$$B(-h-e-e-l-l-l-o) \rightarrow \text{hello}$$



$$p(\bar{I} | x) = \sum_{\bar{\pi} \in B^{-1}(L)} p(\bar{\pi} | x) \quad - \text{Likelihood}$$

Вероятность посредственности I при x - это сумма вероятностей всех предсказанных посредственостей, которые характеризуются в I , а не в одном конкретном пути.



Безумное количество возможных путей.

$$h(x) = \underset{I \in L^T}{\operatorname{argmax}} p(\bar{I} | x) \quad - \text{наиболее вероятная посредственность, среди}$$

Forward - Backward Algorithm

$$d_t(s) \stackrel{\text{def}}{=} \sum_{\pi \in N^T : \pi_{1:t} = s} \prod_{t'=1}^t y_{\pi_{t'}}^{t'}$$

$B(\pi_{1:t}) = \underbrace{B(L'_{1:s})}_{\in L'}$

$$N^T = L'^T, \quad L'^T \in L'^*$$

Сума правдоподібності всіх підмножин, які видаються в $L'_{1:s}$

$L = \text{hello}$

$L' = -h-e-l-l-o -$

$L'_{1:3} = -h-$

$\pi = h-ee-ll-l-oo -$

$d_t(s)$ - правдоподібність того, що перші t символів будуть генеруватися в перші $\sum_{s \in L'}^t$ символів L' .

$$B(\bar{y}^1 \bar{y}^2 \bar{y}^3 \dots \bar{y}^T)$$

$$B(-l_1 - l_2 - l_3 - \dots - l_n)$$

\bar{y}^i - softmax i.e. $[P(\text{blank}), P(A), P(B), P(C), \dots, P(Z)]^T$

$$d_1(1) = y_b^1$$

$$d_1(2) = y_{l_1}^1$$

$$\bar{y}^1 \downarrow \\ -l_1 -$$

наде норм. ваги всіх в першій
помітці, які видаються в l_1

$$d_1(s) = 0, \forall s > 2$$

Однаку прогнозуваннями ми не можеми прогнозувати на базі символів

Recursion

Forward - Backward Algorithm

$$\bar{d}_t(s) \stackrel{\text{def}}{=} d_{t-1}(s) + d_{t-1}(s-1)$$

$$d_t(s) = \begin{cases} \bar{d}_t(s) y_{l_s^t}^t, & \text{if } l_s^t = b, \text{ or } l_{s-1}^t = l_s^t \\ (\bar{d}_t(s) + d_{t-1}(s-1)) y_{l_s^t}^t, & \text{otherwise} \end{cases}$$

— не запускаем нонекамс blank.
— гониммеритий суграви.

1) $d_t(s) = \bar{d}_t(s) y_{l_s^t}^t$, if $l_s^t = b$ - H-E-L-L-O-

 $\bar{d}_t(s) = d_{t-1}(s) + d_{t-1}(s-1)$

$d_t(s)$ — веоашиаси, эмо нефре t омремов Syggm gerogupobamica б нефре s символ l^t .

$\bar{d}_t(s)$ — веоашиаси моро, эмо $t-1$ омремов Syggm gerogupobamica б s , мбо $s-1$ символ l^t .

• $d_{t-1}(s)$

$$\pi_{1:t-1} = \dots \underset{\text{---}}{\text{H}} \underset{\text{---}}{\text{H}} \underset{\text{---}}{\text{E}} \underset{\text{---}}{\text{E}} \dots$$

$$l_{1:s}^t = \underset{s}{\leftarrow \text{H-E-}} \quad \underset{t}{\rightarrow} \quad \text{nothing changes}$$

• $d_{t-1}(s-1)$

$$\pi_{1:t-1} = \dots \underset{\text{---}}{\text{H}} \underset{\text{---}}{\text{H}} \underset{\text{---}}{\text{E}} \underset{\text{---}}{\text{E}} \dots$$

$$l_{1:s-1}^t = \underset{s-1}{\leftarrow \text{H-E-}} \quad \underset{s}{\rightarrow} \quad$$

2) $d_t(s) = \bar{d}_t(s) y_{l_s^t}^t$, if $l_s^t = l_{s-1}^t$ - E-M-M-A

$$\bar{d}_t(s) = d_{t-1}(s) + d_{t-1}(s-1)$$

• $d_{t-1}(s)$

$$\pi_{1:t-1} = \dots \underset{\text{---}}{\text{E}} \underset{\text{---}}{\text{E}} \underset{\text{---}}{\text{M}} \underset{\text{---}}{\text{M}} \underset{\text{---}}{\text{M}} \dots$$

$$l_{1:s}^t = \underset{s}{\leftarrow \text{E-M-M-}} \quad \underset{t}{\rightarrow} \quad \text{nothing changes}$$

• $d_{t-1}(s-1)$

$$\pi_{1:t-1} = \dots \underset{\text{---}}{\text{E}} \underset{\text{---}}{\text{E}} \underset{\text{---}}{\text{M}} \underset{\text{---}}{\text{M}} \underset{\text{---}}{\text{M}} \dots$$

$$l_{1:s-1}^t = \underset{s-1}{\leftarrow \text{E-M-}} \quad \underset{s}{\rightarrow} \quad \underset{t}{\text{M}}$$

Forward - Backward Algorithm

$$\bar{d}_t(s) = d_{t-1}(s) + d_{t-1}(s-1)$$

$$d_t(s) = \sum_{\pi \in NT: \pi_{1:t} = l_s^t} y_{l_s^t}^{\pi_t} \\ B(\pi_{1:t}) = B(l_{1:s}^t)$$

$$d_t(s) = \begin{cases} \bar{d}_t(s) \cdot y_{l_s^t}^t & , \text{ if } l_s^t = b \text{ or } l_{s-2}^t = l_s^t \\ (\bar{d}_t(s) + d_{t-1}(s-2)) \cdot y_{l_s^t}^t & , \text{ otherwise} \end{cases}$$

3) General case

$$d_t(s) = (\bar{d}_t(s) + d_{t-1}(s-2)) \cdot y_{l_s^t}^t = (\underbrace{d_{t-1}(s)}_{\cdot d_{t-1}(s)} + \underbrace{d_{t-1}(s-1)}_{\cdot d_{t-1}(s-1)} + \underbrace{d_{t-1}(s-2)}_{\cdot d_{t-1}(s-2)}) \cdot y_{l_s^t}^t$$

$$\pi_{1:t-1} = -HH--EE--L \left\{ \begin{array}{c} L \\ t \end{array} \right.$$

$$l_{1:s}^t = \xleftarrow[s=6]{} -H-E-L$$

$$\cdot d_{t-1}(s-1)$$

$$-H-E-L$$

$$\pi_{1:t-1} = -HH--EE \left\{ \begin{array}{c} L \\ t \end{array} \right. -H-E-L$$

$$l_{1:s-1}^t = \xleftarrow[s-1=5]{} -H-E-$$

$$\cdot d_{t-1}(s-2)$$

$$-H-EL$$

Дополнительной строкой, когда нет
максимума разрешимо пропустить blank.

$$\pi_{1:t-1} = --HH--EE \left\{ \begin{array}{c} L \\ t \end{array} \right.$$

$$l_{1:s-2} = -H-E \xleftarrow[s-2=4]{} -$$

B maxey cyrrax $B(\pi_{1:t}) = HEL$, mo eamъ noслегобавленности geroglyfomca $l_{1:s}^t$

$$d_t(s) = \sum_{\pi \in NT: \pi_{1:t} = l_s^t} y_{l_s^t}^{\pi_t} \prod_{t'=1}^{t-1} y_{l_{s-t'}^t}^{\pi_{t'}} = |\pi_t = l_s^t| = y_{l_s^t}^t \cdot (\underbrace{d_{t-1}(s)}_{B(\pi_{1:t}) = B(l_{1:s}^t)} + \underbrace{d_{t-1}(s-1)}_{B(\pi_{1:t}) = B(l_{1:s}^t)} + \underbrace{d_{t-1}(s-2)}_{B(\pi_{1:t}) = B(l_{1:s}^t)})$$

Forward - Backward Algorithm

Likelihood of particular path

$$P(\bar{\pi} | x) = \prod_{t=1}^T y_{\pi_t}^t$$

Likelihood of target sequence \bar{I}

$$P(\bar{I} | x) = \sum_{\pi \in B^{-1}(\bar{I})} P(\bar{\pi} | x) = \sum_{\pi \in B^{-1}(\bar{I})} \prod_{t=1}^T y_{\pi_t}^t = \sum_{\substack{\pi \in N^T : \\ B(\pi_{1:T}) = B(I_1^T)}} \prod_{t=1}^T y_{\pi_t}^t$$

сума (беск) нрабогоногобуй боех нумеи, генодуруюшуса \bar{I} .

Likelihood of subsequence $I'_{1:s}$

$$d_t(s) = \sum_{\pi \in N^T : t'=1}^t y_{\pi_{t'}}^t = \sum_{\pi \in N^T : t'=1}^t y_{\pi_t}^t \cdot \prod_{t'=1}^{t-1} y_{\pi_{t'}}^t$$

$$B(\pi_{1:t}) = B(I'_{1:s}) \quad B(\pi_{1:t'}) = B(I'_{1:s})$$

$$d_1(1) = y_b^t$$

$$d_1(2) = y_{L_1}^t$$

$$d_1(s) = 0, \forall s > 0$$

$$\bar{d}_t(s) = d_{t-1}(s) + d_{t-1}(s-1)$$

$$d_t(s) = \begin{cases} y_{l'_s}^t \cdot \bar{d}_t(s), & \text{if } l'_s = b \text{ or } l'_s = l'_{s-1} \\ y_{l'_s}^t \cdot (d_t(s) + d_{t-1}(s-1)), & \text{otherwise} \end{cases}$$

-H-	{ -	-L-L	{ L
-H	{ -	-L-	{ L
-H-E-L	{ L		
-H-E-	{ L		
-H-E	{ L		

Likelihood of target sequence \bar{I}'

$$P(\bar{I}' | x) = d_T(|\bar{I}'|) + d_T(|\bar{I}'|-1)$$

$-H-I^-$ $-H-I^-$

we can skip last blank

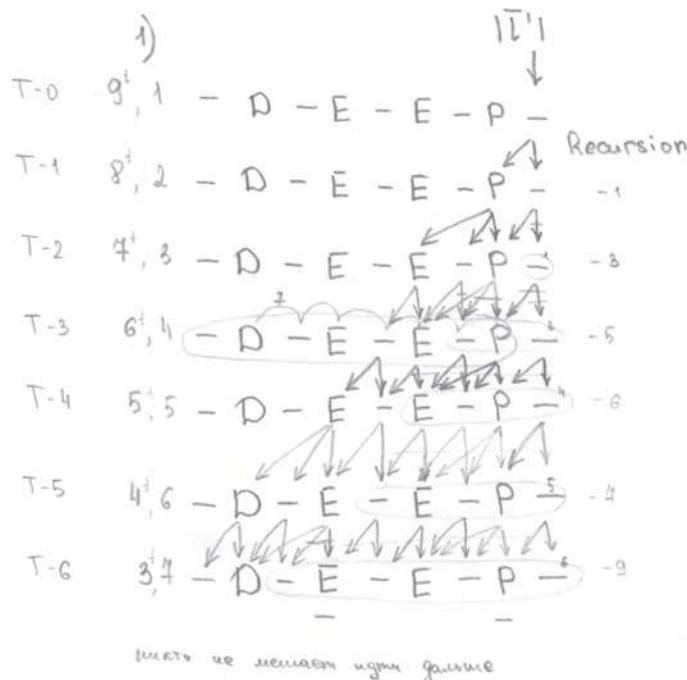
Image $\boxed{\quad}$
 $m \times n$
 $n > m$

9

Forward - Backward Algorithm

$$P(\bar{l} | x) = d_T(|\bar{l}'|) + d_{T-1}(|\bar{l}'|-1)$$

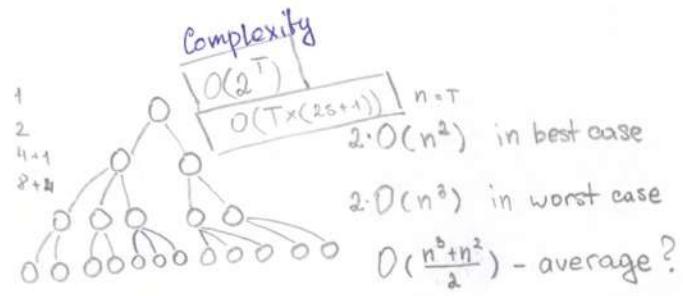
$$d_t(s) = \begin{cases} y_{l'_s}^t \cdot (d_{t-1}(s) + d_{t-1}(s-1)), & \text{if } l'_s = b \text{ or } l'_s = l'_{s-2} \\ y_{l'_s}^t \cdot (d_{t-1}(s) + d_{t-1}(s-1) + d_{t-1}(s-2)), & \text{otherwise} \end{cases}$$



мы не меняем шаги генерации

Если мы будем выбирать blank нормально, например, то наше какого-то символа мы не успеем учесть чтобы распознать, потому что, каскад, можно отбросить некоторые символы.

$$\begin{aligned} d_T(|\bar{l}'|) &= y_b^T \cdot (d_{T-1}(|\bar{l}'|) + d_{T-1}(|\bar{l}'|-1)) \\ 1) d_{T-1}(|\bar{l}'|) &= y_{l'_{111}}^{T-1} (d_{T-2}(|\bar{l}'|) + d_{T-2}(|\bar{l}'|-1)) \\ 2) d_{T-1}(|\bar{l}'|-1) &= y_{l'_{1111}}^{T-1} (d_{T-2}(|\bar{l}'|-1) + d_{T-2}(|\bar{l}'|-2) + d_{T-2}(|\bar{l}'|-3)) \end{aligned}$$



$\frac{l_1 + l_2}{n^t n^t n^t n^t}$
or average just $O(n^3)$
но это не проходит в некоторых
 $T \times T = (S-1) \times (S-1)$
С этим типом можно работать
так как при этом можно
беско ограничения

$$T \leq |\bar{l}'|$$

$$1) \text{ if } d_t(s), t < S-1 \text{ no } 100\% \text{ ff}$$

$$t: 1 \ 2 \ 3 \ 4 \quad s: 1 \ 2 \ 3 \ 4 \ 5 \quad \text{but } s: 1 \ 2 \ 3 \ 4 \ 5 \ 6$$

$$2) \text{ In example } d_{T-2}(|\bar{l}'|) = d_T(9) \text{ не выполняется (это неправильно)}$$

$d_T(8)$ выполняется, но из-за того что $l'_8 \neq b$ and $l'_8 \neq l'_{8-2}$, иначе мы не можем использовать $d_T(9)$ никак не выполняется для s_8 , если s_8 не $E-E$.

To есть не последовательные символы называемые 2 шаги, а не на один

$$t + (\text{как-то не подходит}) < S-1$$

$$t < S-1 - (2)$$

$$t < S-3$$

$$7 < 9 - 3 = 6$$

$$7 < 11 - 3 = 8$$

Мы берём 2 шага генерации, кроме того когда символы $l_t \neq l_{t-2}$ and $l_t \neq b$

Pruning

if in $d_t(s)$

$\Rightarrow t < S-1 -$ как-то не подходит, кроме первого завершения битый (0)

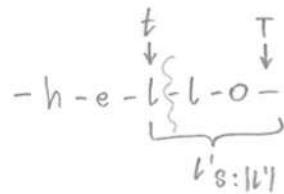
не S , а остальные могут быть лучше

Минимум 1 символ = m pixels $\boxed{0}$ $\boxed{1 \dots m}$ - $\boxed{0 \ 0}$ $\boxed{0 \ 0}$ $\boxed{0 \ 0}$

Forward-Backward Algorithm

$$\beta_t(s) = \sum_{\pi \in NT} \prod_{t'=t}^T y_{\pi_{t'}}^{t'}$$

$\beta(\pi_{t:T}) = \beta(l'_s: l'_T)$



$$\beta_T(l'_T) = y_b^T$$

- h - e - l - l - o -
↑
 $t = |l'|$

$$\beta_T(l'_{|l'|-1}) = y_{l'_{|l'|-1}}^T$$

- h - e - l - l - o -
↑
 $b = |l'| - 1$

$$\beta_T(s) = 0, \forall s < |l'| - 1$$

Не имеем озаглав.

$\beta_t(s)$ - бераамасы озаглав сүйрөнүү, көрүнүү өткөрүү

$$\bar{\beta}_t(s) = \beta_{t+1}(s) + \beta_{t+1}'(s+1)$$

$$\beta_t(s) = \begin{cases} \bar{\beta}_t(s) \cdot y_{l'_s}^t, & \text{if } (y_{l'_s}^t) l'_s = b \text{ or } l'_{s+2} = l'_s \\ (\bar{\beta}_t(s) + \beta_{t+1}(s+2)) y_{l'_s}^t, & \text{otherwise} \end{cases}$$

$$\beta_{t+1}(s)$$

- { - l - l - o - }

if $l'_s = b$ or $l'_{s+2} = l'_s$

$$\beta_{t+1}(s+1)$$

- { l - l - o - }

но жарысдаам професияны blank

$l'_{s+1:|l'|}$

$$\beta_{t+1}(s+2)$$

e { l - l - o - }

при обмене паспорта,

и жарысдаам професияны blank

e { - l - l - o - }

$l'_{s:|l'|}$

Forward - Backward Algorithm

$\prod_x x = 0$, $x < 1$ - underflow

$$C_t = \sum_s d_t(s) \quad \hat{d}_t(s) = \frac{d_t(s)}{C_t}$$

$$D_t = \sum_s \beta_t(s) \quad \hat{\beta}_t(s) = \frac{\beta_t(s)}{D_t} \quad - \text{Normalized}$$

нормированием коэффициентов

$\hat{d}_t(s), \hat{\beta}_t(s)$ - вероятность того, что на t очередь будущим
(s -й символ)

$\hat{d}_t(s)$ - будущим не будущие s символы l'

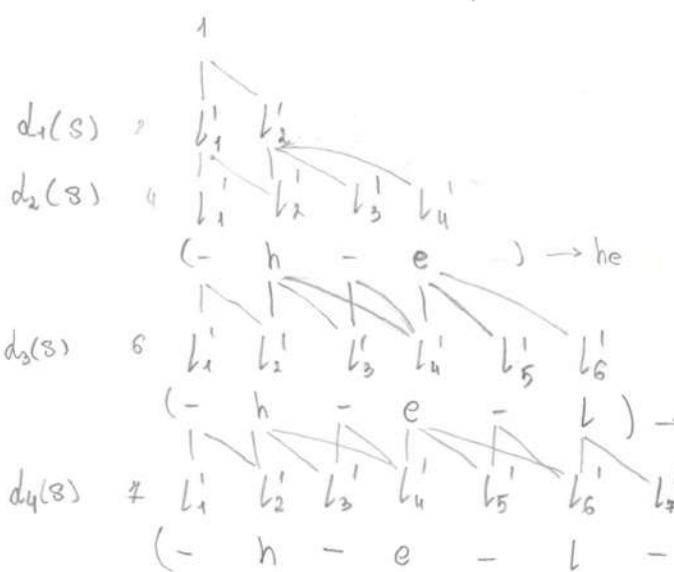
$\hat{\beta}_t(s)$ - будущим ожидания s символа g конца l'

Причина, что в любом случае будущим для $d_t(s)$
и $\beta_t(s)$. Сложность $O(T \cdot S)$.

Но, если $d_1(2), d_1(3) \dots d_1(S) = 0$ и в маине $t=1$.

Последнее $t < S-1-n$ $d_t(s)=0$, нет уже будущем, не имеет еще этого паттерна

И не более? Обратно паттерн на 2, но если $l'_S = b$ or $l'_S = l'_{S-2}$,
то паттерн на 1 есть



2t - (если $l_t = l_{t+1}$ то t)

Но, если на первом паттерне есть гапы,
на самом деле требуется $d_t(s)$ и $\beta_t(s)$
гапы.

~~$d_t(l')$~~
 β

Forward - Backward Algorithm

Loss Function

$$D^{ML}(S, \mathcal{N}_w) = - \sum_{(x,z) \in S} \ln(p(z|x))$$

$$P(S, \mathcal{N}_w) = \prod_{(x,z) \in S} p(z|x) \rightarrow \max$$

$$-\ln(P(S, \mathcal{N}_w)) = - \sum_{(x,z) \in S} \ln(p(z|x)) \rightarrow \min$$

D - objective function

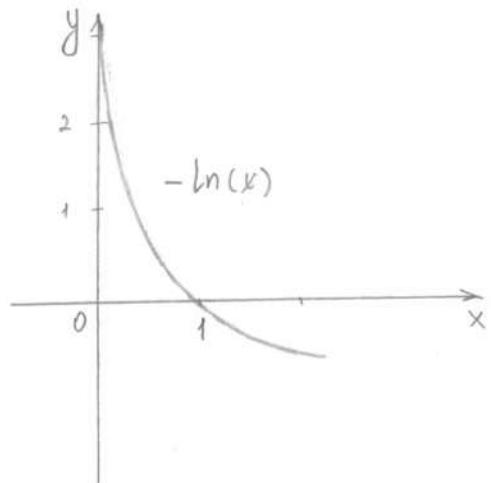
ML - maximum likelihood

S - dataset

\mathcal{N}_w - model

x - input sequence.

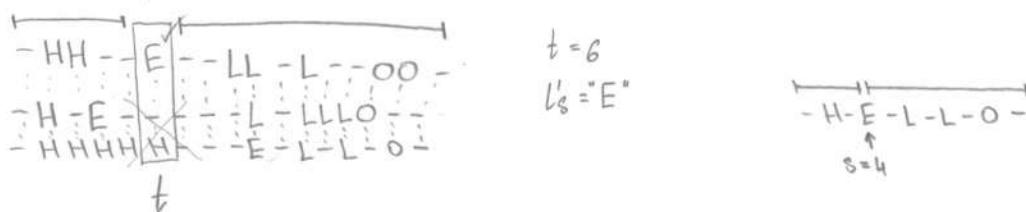
z - target sequence.



$$\frac{d D^{ML}(z|x, z), \mathcal{N}_w}{d y_k^t} = \frac{-d \ln(p(z|x))}{d y_k^t} \quad \text{partial derivative by some } y_k^t$$

Forward - Backward Algorithm

$$d_t(s) \cdot \beta_t(s) = \sum_{\pi \in B^{-1}(L)} y_{L_s^t}^t \cdot \prod_{t'=1}^T y_{\pi_{t'}^t}^{t'}$$



dt(3) - биржевые газы в градусах по С., морские $t'_{1:3}$

$\beta_+(3)$ - вероятность га + yragans om 3 go konusa , mo eeme $L'_s:1111$

To express $\pi_t = b'_s$, make note $b \cup b_{df}(s) \cup b_{\beta_t}(s)$ are

постається блокування b'_3 і рекурсивно проходить по всім відносимим предикатам підмнж, але наступну разу в зустрічі $\beta_t(s)$.

$$d_t(s) = \sum_{\pi \in N^+} \prod_{t'=1}^t y_{\pi t'}^{s'}$$

$$B(\bar{v}_1;+) = B(l'_1;+)$$

$$\beta_t(s) = \sum_{\pi \in N^t} \prod_{t'=t}^T y_{\pi_{t'}}^{t'}$$

$$\beta(\pi_{t:T}) = \beta(l_s^t : |l_s^t|)$$

$$d_t(s) = \sum_{\pi_t \in \Pi^T} y_{\pi_t}^t \prod_{t'=1}^{t-1} y_{\pi_{t+}}^{t'}$$

$$B(\pi_{1:t}) = B(l'_{1:t-5})$$

$$\beta_t(s) = \sum_{\pi \in N^T} y_{\pi_t}^t \prod_{t'=t+1}^T y_{\pi_{t'}}^{t'}$$

$$B(\pi_{t-T}) = B(l'_{S+1}l')$$

$$\begin{aligned}
 d_t(S) \cdot \beta_t(s) &= (abc + fgh) \cdot (cde + hij) = abc \cdot cde + fgh \cdot hij + \underbrace{abc \cdot hij + fgh \cdot cde}_{\text{необязательные}} = \\
 &= c(\underbrace{abcde}_{\substack{\text{full} \\ \text{word} \\ \text{path}}}) + h(\underbrace{fghij}_{\substack{\text{full} \\ \text{word} \\ \text{path}}}) + 0
 \end{aligned}$$

$$d_t(s) \cdot \beta_t(s) = \sum_{\pi \in B^{-1}(l)} \underbrace{y_{\pi_t}^t}_{\pi_t = l_s^t} \cdot \underbrace{\prod_{t'=1}^T y_{\pi_{t'}}^{t'}}_{p(\pi | x)} \quad \begin{array}{l} \text{const} \\ \downarrow \\ t = \text{const} \\ s = \text{const} \end{array} \quad \begin{array}{l} d_1(3) = 0 \\ d_2(5) = 0 \\ \dots \end{array}$$

$$p(\pi | x) = \prod_{t=1}^T y_{\pi_t}^t, \forall \pi \in L^{1:T}, L^{1:T} \in L^{1:T}$$

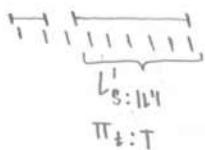
$$d_t(s) \cdot \beta_t(s) \cdot \frac{1}{y_{\pi_t}^t} = \sum_{\pi \in B^{-1}(l)} p(\pi | x)$$

1) $p(l | x) = \sum_{\pi \in B^{-1}(l)} p(\pi | x) =$, но мы не можем вычислить

2) $p(l | x)_t = \sum_{s=1}^{|L'|} \frac{1}{y_{l_s^t}^t} \cdot d_t(s) \cdot \beta_t(s)$, важность априори $p(l | x)$
но мы не можем $t = \text{const}$, так как $1 \neq 2$
 $\sum_{s=1}^{|L'|} \sum_{\pi \in B^{-1}(l)} p(\pi | x), t$ $E = O^{ML}(S, \theta_0) = -\sum_{l \in S} \ln(p(l | x))$

Мы отображаем $m \cdot t$ в номинальную и несогласованную
 форму, так что $d_t(s) = 0$, ибо $\beta_t(s) = 0$,
 соответственно $p(l | x)_t = 0 \dots ?$

Но $d_1(3) = 0 \quad \beta_1(3) > 0$



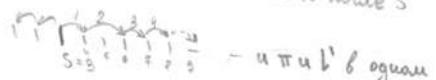
Для любого t , $p(l | x)_t > 0$, если первый $t-s-1-n$ градус узла равен β

π_t

$$T-t \geq (|L'|-s)-1-n?$$

если $=$ одинаков - 1 (- одинаков?)

одинаковы



$3-3=0$ - можно засчитать бланка youth
 $6-1=5$ - бланк
 $5-1=4$ - skip

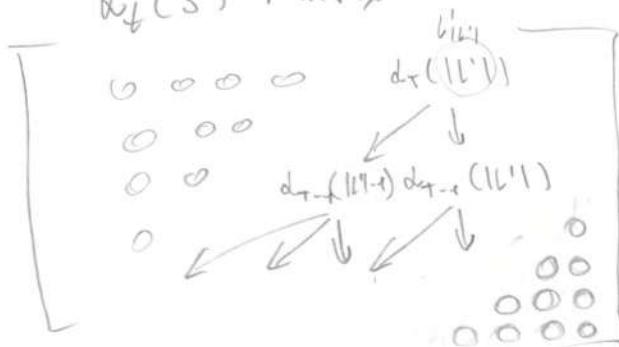
$$\nabla E_y^t = \begin{bmatrix} y_k^{t1} \\ y_k^{t2} \\ y_k^{t3} \\ \vdots \end{bmatrix}$$

$$\frac{dO^{ML}}{dy_k^t} = \frac{d(-\sum \ln(p(l | x)))}{dy_k^t}$$

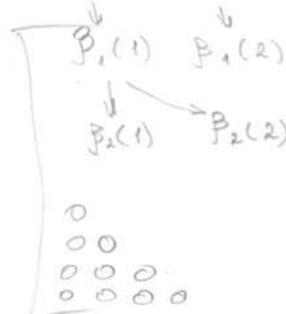
for t of 1 word

$$\nabla E_y = \sum_t \nabla E_y^t$$

$d_t(s)$ Matrix



$\beta_t(s)$



$$- h - e - l - f - o -$$

\uparrow
 $s=8$

char[]

Рекурсивно вспомогаю амплиут $d_t(s)$ и $\beta_t(s)$.

hel - b - o -
h - e l - l o

gumbos
 $d_t(4)$
 $d_t(3)$

$$y_k^t = \sum_{s: l_s^t = k} d_t(s) \beta_t(s)$$

Как быстро отыскать $s: l_s^t = k$?

Будет некак? можно лучше?

Так же сумма где находят буквы направления нумер.



- Почему нужно d и β ?
- Включают ли пробел? 80% да нет, ищите RNN.
- В конце нужны levenshtein? или наиболее правдоподобные группы символов.
- Противоречие ' b' , ' g ' и регулярные выражения? почему регулярные выражения распознаваются?

CTC Loss . Forward - Backward Algorithm

$$P(\bar{\pi} | x) = \prod_{t=1}^T y_{\pi_t}^t \quad - \text{Likelihood of particular path } \bar{\pi}$$

$\forall \pi \in L^{1:T}$

$$h(x = (R^m)^*) = z$$

$$h(\bar{x}_t) = \bar{y}^t, \bar{y}^t \in R^{1|L'|}$$

\bar{y}^t - softmax i.e. $[P(\text{blank}), P(L_1), \dots]$

$$P(\bar{l} | x) = \sum_{\bar{\pi} \in B^{-1}(l)} P(\bar{\pi} | x)$$

$$\mathcal{X} = L^*$$

$$L' = \{\text{blank}\} \cup L, \quad \mathcal{Z} = L'^*$$

$B: \mathcal{Z} \rightarrow \mathcal{X}$ - Decoding

$$h(x) = \operatorname{argmax}_{\bar{l} \in L'^*} P(\bar{l} | x) \quad - \text{Classifier (Greedy)}$$

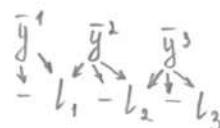
$$d_t(s) = \sum_{\bar{\pi} \in N^T: t'_1 = s} \prod_{t'=1}^t y_{\pi_{t'}}^{t'} \quad - \text{Сума ннабогодобие мого, кмо нябре } t \text{ симбаов Sygym genogupobamse b nябре } s \text{ симбаов } L'.$$

$B(\pi_{s:t}) = B(L'_{s:t})$

$$d_1(1) = y_b^1$$

$$d_1(2) = y_{L_1}^1$$

$$d_1(s) = 0, \forall s > 2$$



Очум ннегнагашен не можем ннекајамо 2 симбаа

$$\bar{d}_t(s) = d_{t-1}(s) + d_{t-1}(s-1) \quad - l_1 - \{ - \quad - l_1 - l_2 \} l_2$$

$$P(\bar{l} | x) = d_T(1|L'|) + d_T(1|L'|-1)$$

$$d_t(s) = \begin{cases} \bar{d}_t(s) \cdot y_{L_s}^t & , \text{ if } l_s' = b \text{ or } l_s' = l_{s-1}' \\ (\bar{d}_t(s) + d_{t-1}(s-2)) y_{L_s}^t & , \text{ otherwise} \end{cases}$$

$$\beta_t(s) = \sum_{\bar{\pi} \in N^T: t'_s = s} \prod_{t'=t}^T y_{\pi_{t'}}^{t'}, \quad \beta_T(1|L'|) = y_b^T \quad - l_1 - \{ l_2 \}$$

$B(\pi_{s:T}) = B(L'_{s:T})$

$$\beta_T(1|L'|-1) = y_{L_{T-1}}^T$$

$$\beta_T(s) = 0, \forall s < |L'| - 1 \quad - l_1 - l_2 -$$

$$\bar{\beta}_t(s) = \beta_{t+1}(s) + \beta_{t+1}(s+1)$$

$$\beta_t(s) = \begin{cases} y_{L_s}^t \bar{\beta}_t(s) & , \text{ if } l_s' = b \text{ or } l_s' = l_{s+2}' \\ y_{L_s}^t (\bar{\beta}_t(s) + \beta_{t+1}(s+2)) & , \text{ otherwise} \end{cases}$$

$$- \{ - l - l - 0 -$$

$$O^{ML}(S, N_w) = - \sum_{(x, z) \in S} \ln(P(z|x))$$

$$d_t(s) \cdot \beta_t(s) = \sum_{\pi \in B^{-1}(l)} y_{L_s}^t \underbrace{\prod_{t'=1}^T y_{\pi_{t'}}^{t'}}_{P(\pi | x)} \quad - H-E-E-L-L-L-O-$$

$\pi_t = l_s$

$$P(l | x) = \sum_{s=1}^{|L'|} \sum_{\pi \in B^{-1}(l)} P(\pi | x) = \sum_{s=1}^{|L'|} \frac{1}{y_{L_s}^t} \cdot d_t(s) \cdot \beta_t(s) \quad P'(l | x) = \sum \frac{1}{y_{L_s}^t} d_t(s) \beta_t(s)$$

$$\frac{dp(l | x)}{dy_k^t} = \frac{1}{(y_k^t)^2} \cdot \sum_{s \in \text{lab}(l, k)} d_t(s) \beta_t(s)$$

$$\frac{dO^{ML}(S, N_w)}{dy_k^t} = \frac{d(-\ln p(l | x))}{dy_k^t} = - \frac{1}{p(l | x)} \frac{dp(l | x)}{dy_k^t}$$

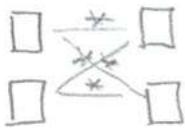
$$\text{lab}(l, k) = \{s : l_s' = k\}$$

most symbols
set s where

For each word we'll get matrix of derivatives $T \times |L'| \rightarrow [S_i], [S_j] \dots$
B cause we берга ессе бе Sygym, она Sygym пай澎湃мий о-ми
Cuorcuomby O(T \times S)



w/2 h/2



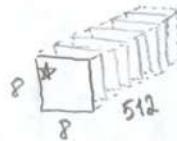
Вч. сложность осталась такой же

$$3^1 + 2^2 + 2^3 + 2^4 \\ \quad \quad \quad 5 \quad 7 \quad 9$$

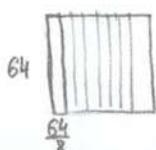
$$\begin{array}{r}
 1 \left(\begin{array}{r} 3 \\ 5 \\ \hline 10 \end{array} \right) \\
 2 \left(\begin{array}{r} 9 \\ 12 \\ 4 \\ 14 \\ \hline 28 \end{array} \right) \\
 3 \left(\begin{array}{r} 5 \\ 32 \\ 64 \\ \hline 100 \end{array} \right) \\
 4 \left(\begin{array}{r} 66 \\ 88 \\ \hline 126 \end{array} \right)
 \end{array}$$

Receptive Field

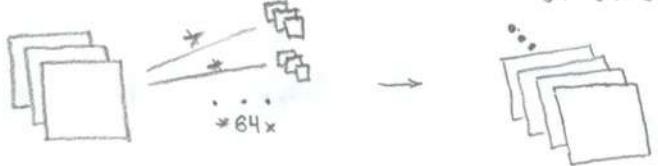
3	$8 \times 64 \times 64$
5	$64 \times 64 \times 64$
10	
12	$64 \times 32 \times 32$
14	$128 \times 32 \times 32$
28	
30	$128 \times 16 \times 16$
32	$256 \times 16 \times 16$
64	
66	$256 \times 8 \times 8$
68	$512 \times 8 \times 8$



2) GMP(1, *)

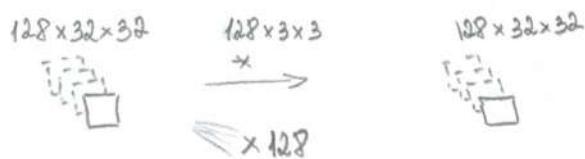
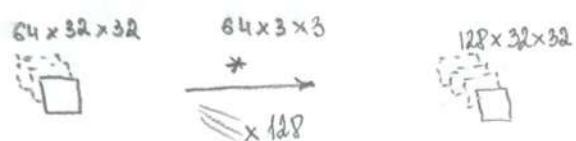
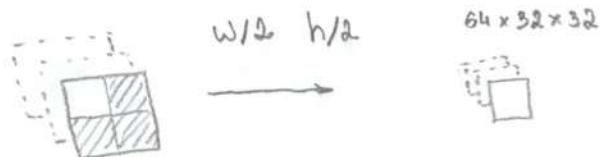


RGB 64×64 + padding



VGG

Max Pool (2×2)



Computational Complexity remains the same

Машындың оғыту 8-жада анында арамасуның
өзгірілген алғашкы түрде таңдаіттың және
менниң қабылдаіттың алгоритмдер мен деформерді
тәжірибелі әрасандың штампекті қосыннан.