Нейронные сети: как заставить их обучаться

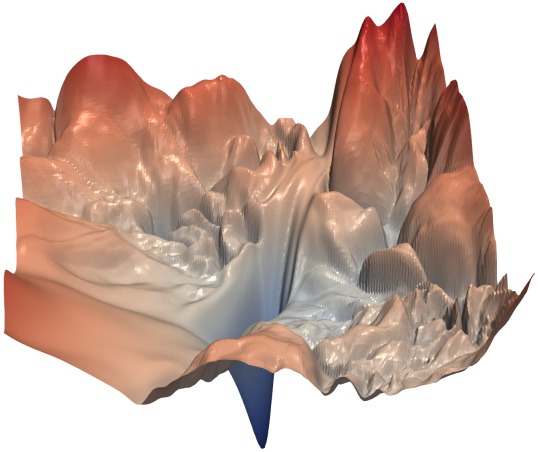
Константин Архипенко

7 октября 2020 г.

ИСП РАН

Почему сложно обучать нейросети

ИСП РАН



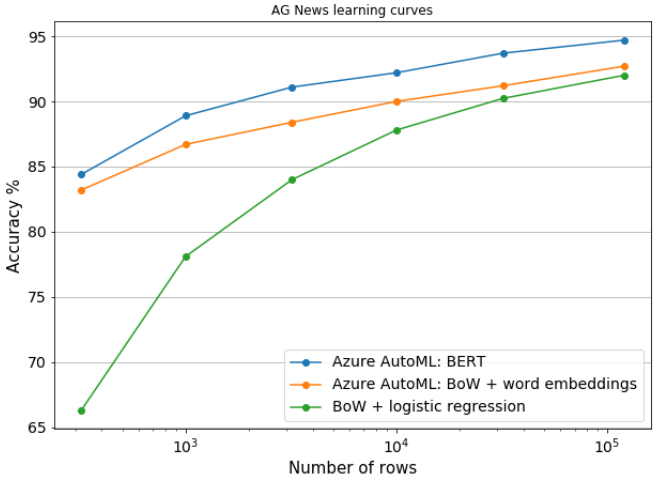
[1] Hao Li et al. “Visualizing the Loss Landscape of Neural Nets” NeurIPS (2018)

Почему сложно обучать нейросети

ИСП РАН

- Много “плохих” локальных минимумов и седловых точек
- Переобучение
- Transfer learning полностью не спасает

Get more data



Аугментация данных

Fig. 1. Exemplar applications of image transformations available in Albumentations

2

- Для картинок: повороты, размытие, масштабирование, сгенерированные GAN, ...

[2] Buslaev et al. “Albumentations: Fast and Flexible Image Augmentations” (2020)

Аугментация для NLP

ИСП РАН

Augmenter	Target	Augmenter	Action	Description
Textual	Character	KeyboardAug	substitute	Simulate keyboard distance error
Textual		OcrAug	substitute	Simulate OCR engine error
Textual		RandomAug	insert, substitute, swap, delete	Apply augmentation randomly
Textual	Word	AntonymAug	substitute	Substitute opposite meaning word according to WordNet antonym
Textual		ContextualWordEmbsAug	insert, substitute	Feeding surroundings word to BERT, DistilBERT, RoBERTa or XLNet language model to find out the most suitlabe word for augmentation

Аугментация для NLP[3]

ИСП РАН

	Sentence
Original	The quick brown fox jumps over the lazy dog
Synonym (PPDB)	The quick brown fox climbs over the lazy dog
Word Embeddings (word2vec)	The easy brown fox jumps over the lazy dog
Contextual Word Embeddings (BERT)	Little quick brown fox jumps over the lazy dog
PPDB + word2vec + BERT	Little easy brown fox climbs over the lazy dog

[3] https://github.com/makcedward/nlpaug

- Пример — логистическая регрессия (бинарная классификация)
- Maximum likelihood estimation:

$$\frac{1}{N}\sum_n \log P(y = y_n|\mathbf{x}_n, \mathbf{w}, w_0) \to \max_{\mathbf{w}, w_0}$$

$$\hat{y} = P(y = 1|\mathbf{x}, \mathbf{w}, w_0) = \sigma(w_0 + \langle \mathbf{w}, \mathbf{x}\rangle)$$

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

- Переход к минимизации кросс-энтропии:

$$\frac{1}{N}\sum_n \text{xent}(y_n, \hat{y}_n(\mathbf{x}_n, \mathbf{w}, w_0)) \to \min_{\mathbf{w}, w_0}$$

$$\text{xent}(y, \hat{y}) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

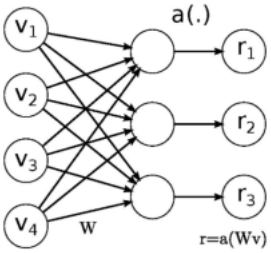
Регуляризация

- Уравнение разделяющей гиперплоскости: $w_0 + \langle \mathbf{w}, \mathbf{x} \rangle = 0$
- Умножение уравнения на 2 не меняет саму гиперплоскость
- Но увеличение $\mathbf{w}$ делает модель более чувствительной к выбросам
- Логистическая регрессия с ℓ_2-регуляризацией:

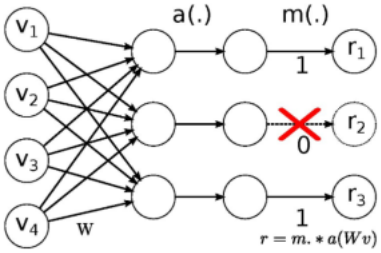
$$\frac{1}{N} \sum_n \text{xent}(y_n, \hat{y}_n(\mathbf{x}_n, \mathbf{w}, w_0)) + \frac{\lambda}{2} \|\mathbf{w}\|_2 \to \min_{\mathbf{w}, w_0}$$

- В нейронных сетях: регуляризация dense слоев и сверточных фильтров

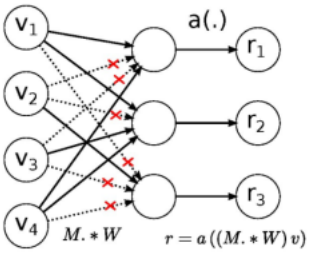
Dropout и DropConnect



No-Drop Network



DropOut Network



DropConnect Network

Dropout и DropConnect: dense слой

ИСП РАН

- p — drop rate, $a(\cdot)$ — функция активации, $\mathbf{W}$ — матрица параметров
- Dropout:

$$m_i \sim \text{Bernoulli}(p)$$

$$\mathbf{r} = \frac{1}{1-p}\mathbf{m} \odot a(\mathbf{W}\mathbf{v})$$

- DropConnect:

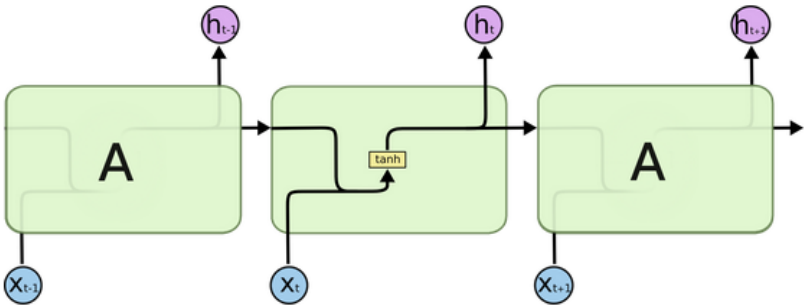
$$m_{ij} \sim \text{Bernoulli}(p)$$

$$\mathbf{r} = a((\mathbf{M} \odot \mathbf{W})\mathbf{v})$$

- До появления Transformer — SOTA языковые модели
- Vanilla RNN:

$$h_t = f(Wx_t + Uh_{t-1} + b)$$

$$h_0 = 0$$

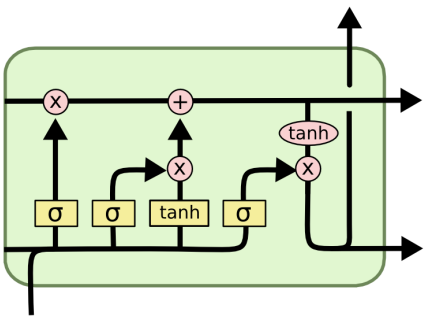


$$h_t = f(Wx_t + Uh_{t-1} + b)$$

- Проблемы с градиентом:

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial h_T}\frac{\partial h_T}{\partial W} \qquad \frac{\partial h_T}{\partial W} = \left(\prod_{t=1}^{T-1}\frac{\partial h_{t+1}}{\partial h_t}\right)\cdot\frac{\partial h_1}{\partial W}$$

- Величина (норма) произведения зависит от нормы матрицы U. Если последняя слишком отклоняется от 1, получим vanishing/exploding gradients



$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i)$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f)$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o)$$

$$\widetilde{c}_t = \tanh(W^c x_t + U^c h_{t-1} + b^c)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t$$

$$h_t = \tanh(c_t) \odot o_t$$

AWD-LSTM

ИСП РАН

- Merity et al. “Regularizing and Optimizing LSTM Language Models” (2017)
- Применяется DropConnect к рекуррентным матрицам ($\mathbf{U}^i, \mathbf{U}^f, \mathbf{U}^c, \mathbf{U}^o$)
- Sample weight — отношение длины обрезанной последовательности к длине исходной
- Weight tying
- Activity regularization

AWD-LSTM: трюки

ИСП РАН

- Weight tying: используются одни и те же веса для embedding и softmax слоев
- Activity regularization: ℓ_2-регуляризация скрытых состояний $\mathbf{h}_t$
- Temporal activity regularization: ℓ_2-регуляризация разницы между $\mathbf{h}$ для соседних timestep:

$$\text{TAR} = \beta \|\mathbf{h}_t - \mathbf{h}_{t-1}\|_2$$

- Вывод: SOTA модели часто содержат специфичные трюки, касающихся регуляризации и оптимизации
- Сейчас рекуррентные нейросети для моделирования языка вытеснены Трансформерами

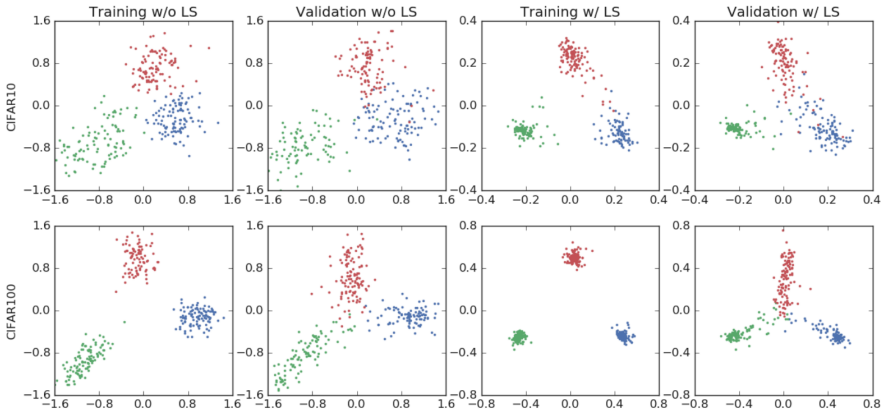
Label smoothing[4]

ИСП РАН

- Классификация на K классов
- Ground truth — one-hot векторы $\mathbf{y}_n$ с единицей в позиции номера класса
- Заменяем единицу в этих векторах на $1 - \alpha$, а все нули — на $\frac{\alpha}{K-1}$

[4] Müller et al. “When Does Label Smoothing Help?” (2019)

Label smoothing

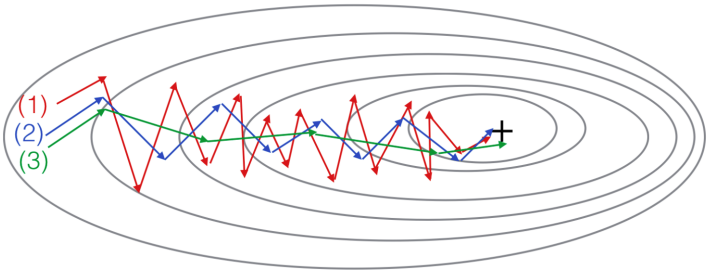


- Stochastic gradient descent с momentum:

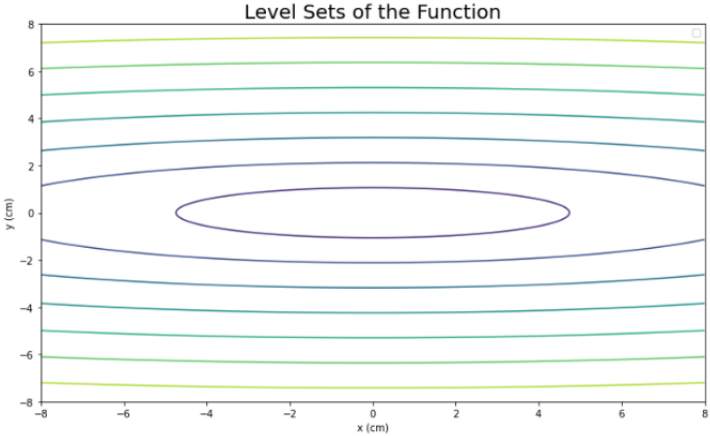
$$\mathbf{v}_t \leftarrow \gamma \mathbf{v}_{t-1} - \eta \nabla_{\boldsymbol{\theta}} L(B, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{t-1}, B=B_t}$$

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} + \mathbf{v}_t$$

- Типичное значение $\gamma = 0.9$



- Мотивация:



- Скользящее среднее квадрата градиента:

$$\mathrm{E}g_t^2 = \beta \mathrm{E}g_{t-1}^2 + (1-\beta) g_t^2$$

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{\eta}{\sqrt{\mathrm{E}g_t^2 + \varepsilon}} \cdot g_t$$

- Вторая производная — связь с кривизной по направлению
- Adagrad: сумма вместо скользящего среднего, эффективная learning rate быстро падает (недостаток)

Adam[5]

$$\mathbf{g}_t \leftarrow \nabla_{\boldsymbol{\theta}} L(B, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{t-1}, B=B_t}$$

$$\widetilde{\boldsymbol{\phi}}_t \leftarrow \beta_1 \widetilde{\boldsymbol{\phi}}_{t-1} + (1-\beta_1) \cdot \mathbf{g}_t$$

$$\widetilde{\boldsymbol{\psi}}_t \leftarrow \beta_2 \widetilde{\boldsymbol{\psi}}_{t-1} + (1-\beta_2) \cdot \mathbf{g}_t^2$$

$$\boldsymbol{\phi}_t \leftarrow \widetilde{\boldsymbol{\phi}}_t / (1-\beta_1^t)$$

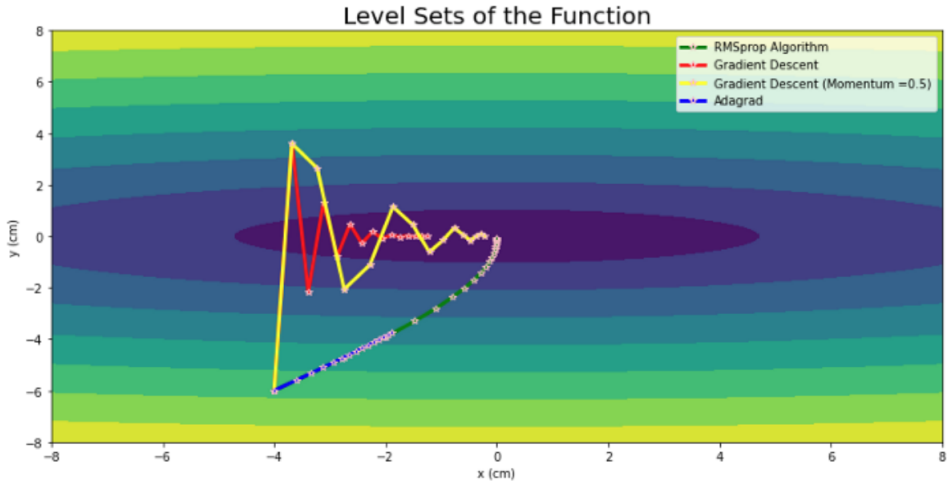
$$\boldsymbol{\psi}_t \leftarrow \widetilde{\boldsymbol{\psi}}_t / (1-\beta_2^t)$$

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha \boldsymbol{\phi}_t / (\sqrt{\boldsymbol{\psi}_t} + \varepsilon)$$

- Скользящие средние градиента и квадрата градиента
- Проблема холодного старта: деление на $(1-\beta)^t$

[5] Kingma & Ba “Adam: A Method for Stochastic Optimization” (2014) ICLR

Адаптивные методы vs SGD



Адаптивные методы vs SGD

ИСП РАН

- SGD с momentum обычно требуется больше времени для сходимости, но он менее подвержен over-shooting (как Adam)
- Сказанное выше — типичное поведение, не всегда оно так
- Есть статья, где предлагается динамическое переключение с Adam на SGD[6]

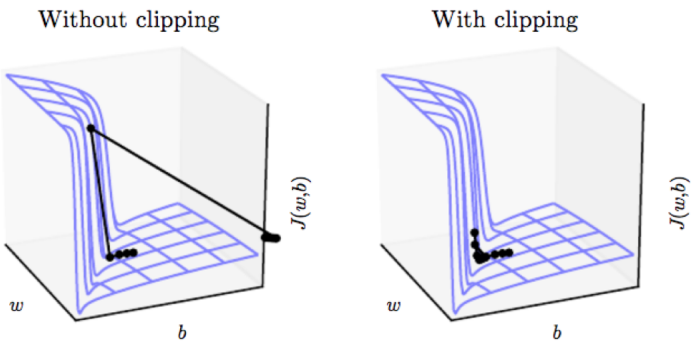
[6] Keskar & Socher “Improving Generalization Performance by Switching from Adam to SGD” (2017)

Другие трюки для оптимизации

ИСП РАН

- Gradient clipping: если норма градиента больше c, то

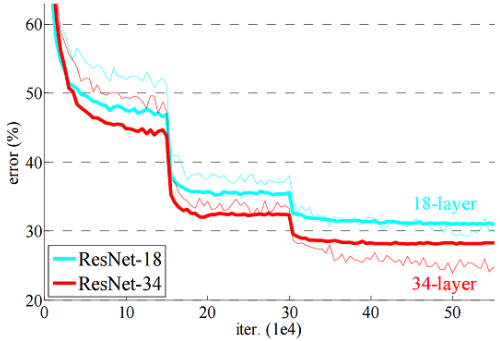
$$\mathbf{g} \leftarrow c \cdot \frac{\mathbf{g}}{\|\mathbf{g}\|}$$



Другие трюки для оптимизации

ИСП РАН

- Learning rate schedule:



- Похожий эффект имеет изменение batch size во время обучения[7]
- Early stopping

[7] Smith et al. “Don’t Decay the Learning Rate, Increase the Batch Size” ICLR 2018

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

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

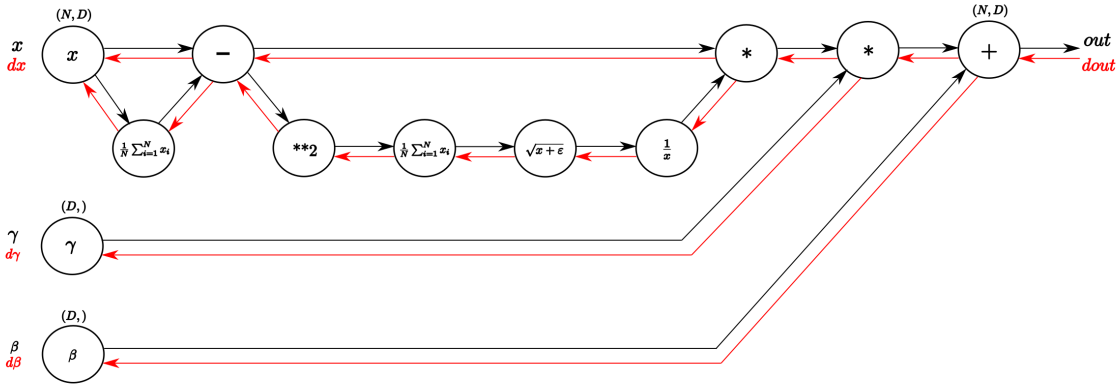
$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

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

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

- Batch norm после функции активации часто дает результаты лучше

Batch norm: граф вычислений



Почему batch norm работает?

ИСП РАН

- Раньше считалось, что помогает справиться с internal covariate shift[8] (постоянное изменение распределения входа следующего слоя при изменении весов предыдущего)
- Более новый результат: делает функцию потерь более гладкой[9]

Theorem 4.1 (The effect of BatchNorm on the Lipschitzness of the loss)**.** *For a BatchNorm network with loss $\widehat{\mathcal{L}}$ and an identical non-BN network with (identical) loss $\mathcal{L}$,*

$$\left|\left|\nabla_{\boldsymbol{y}_j}\widehat{\mathcal{L}}\right|\right|^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(\left|\left|\nabla_{\boldsymbol{y}_j}\mathcal{L}\right|\right|^2 - \frac{1}{m}\left\langle \mathbf{1}, \nabla_{\boldsymbol{y}_j}\mathcal{L}\right\rangle^2 - \frac{1}{m}\left\langle \nabla_{\boldsymbol{y}_j}\mathcal{L}, \hat{\boldsymbol{y}}_j\right\rangle^2\right).$$

[8] Ioffe & Szegedy “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift” (2015)

[9] Santurkar et al. “How Does Batch Normalization Help Optimization? “ (2018)

Lemma 4.5 (BatchNorm leads to a favourable initialization). *Let W^* and $\widehat{W}^*$ be the set of local optima for the weights in the normal and BN networks, respectively. For any initialization W_0*

$$\left|\left|W_0 - \widehat{W}^*\right|\right|^2 \leq ||W_0 - W^*||^2 - \frac{1}{||W^*||^2}\left(||W^*||^2 - \langle W^*, W_0\rangle\right)^2,$$

if $\langle W_0, W^*\rangle > 0$, where $\widehat{W}^*$ and W^* are closest optima for BN and standard network, respectively.

- Доказательство: $BN((a\mathbf{W})\mathbf{x}) = BN(\mathbf{W}\mathbf{x}) \quad \forall a > 0$, поэтому для любого минимума в non-BN сети $\mathbf{W}^*$ точка $\hat{\mathbf{W}} = k\mathbf{W}^*$ будет минимумом в BN-сети
- Возьмем $k = \frac{\langle \mathbf{W}^*, \mathbf{W}_0\rangle}{||\mathbf{W}^*||^2}$. Раскроем квадраты норм в $||\mathbf{W}_0 - \hat{\mathbf{W}}||^2 - ||\mathbf{W}_0 - \mathbf{W}^*||^2$, получим отрицательное число $-||\mathbf{W}^*||^2 \cdot (1-k)^2$
- Последнее верно для произвольного $\hat{\mathbf{W}}$, значит, и для $\hat{\mathbf{W}}^*$