

Основы практического использования нейронных сетей.

Лекция 7. Методы повышения эффективности
алгоритмов обучения для глубоких НС.

Дмитрий Буряк.
к.ф.-м.н.
dyb04@yandex.ru

Batch normalization

- ❑ Нормализация + декорреляция входных данных → повышение эффективности обучения.
- ❑ Нарушение полученных свойств входных векторов для промежуточных данных во внутренних слоях (internal covariance shift).
- ❑ Идея: проводить предобработку входных данных для каждого внутреннего слоя.
- ❑ Оптимизация вычислительных затрат → нормализация внутренних данных (без декорреляции).

Batch normalization. Алгоритм

Input: Network N with trainable parameters Θ ;
subset of activations $\{x^{(k)}\}_{k=1}^K$

Output: Batch-normalized network for inference, $N_{\text{BN}}^{\text{inf}}$

- 1: $N_{\text{BN}}^{\text{tr}} \leftarrow N$ // Training BN network
- 2: **for** $k = 1 \dots K$ **do**
- 3: Add transformation $y^{(k)} = \text{BN}_{\gamma^{(k)}, \beta^{(k)}}(x^{(k)})$ to $N_{\text{BN}}^{\text{tr}}$ (Alg. 1)
- 4: Modify each layer in $N_{\text{BN}}^{\text{tr}}$ with input $x^{(k)}$ to take $y^{(k)}$ instead
- 5: **end for**
- 6: Train $N_{\text{BN}}^{\text{tr}}$ to optimize the parameters $\Theta \cup \{\gamma^{(k)}, \beta^{(k)}\}_{k=1}^K$
- 7: $N_{\text{BN}}^{\text{inf}} \leftarrow N_{\text{BN}}^{\text{tr}}$ // Inference BN network with frozen parameters
- 8: **for** $k = 1 \dots K$ **do**
- 9: // For clarity, $x \equiv x^{(k)}, \gamma \equiv \gamma^{(k)}, \mu_{\mathcal{B}} \equiv \mu_{\mathcal{B}}^{(k)}$, etc.
- 10: Process multiple training mini-batches $\mathcal{B}$, each of size m , and average over them:

$$\mathbb{E}[x] \leftarrow \mathbb{E}_{\mathcal{B}}[\mu_{\mathcal{B}}]$$

$$\text{Var}[x] \leftarrow \frac{m}{m-1} \mathbb{E}_{\mathcal{B}}[\sigma_{\mathcal{B}}^2]$$
- 11: In $N_{\text{BN}}^{\text{inf}}$, replace the transform $y = \text{BN}_{\gamma, \beta}(x)$ with

$$y = \frac{\gamma}{\sqrt{\text{Var}[x] + \epsilon}} \cdot x + \left(\beta - \frac{\gamma \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}}\right)$$
- 12: **end for**

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1, \dots, x_m\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

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

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

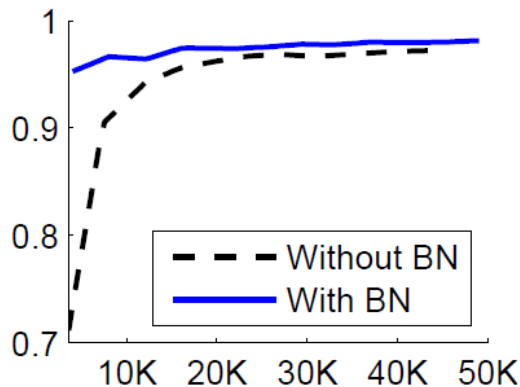
$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

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

Batch normalization. Примеры.

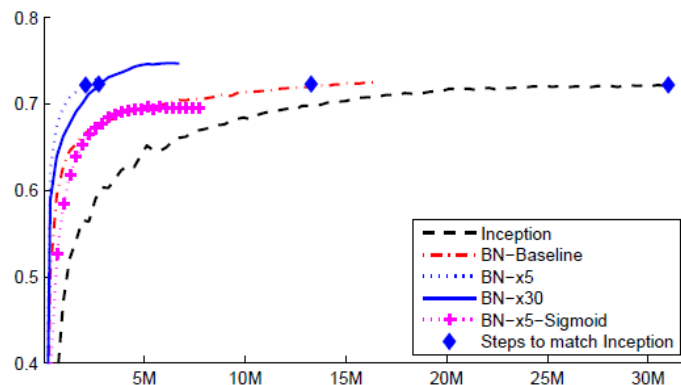
❑ Классификация MNIST

❑ Сеть 784x100x100x100x10,
50000 обучающих примеров



❑ Классификация ImageNet

❑ Сеть $13.6 \cdot 10^6$ параметров,
1000 классов



Model	Steps to 72.2%	Max accuracy
Inception	$31.0 \cdot 10^6$	72.2%
BN-Baseline	$13.3 \cdot 10^6$	72.7%
BN-x5	$2.1 \cdot 10^6$	73.0%
BN-x30	$2.7 \cdot 10^6$	74.8%
BN-x5-Sigmoid		69.8%

Регуляризация L2

- ❑ Регуляризация - метод предотвращения переобучения НС.
- ❑ Введение штрафа для больших весов.

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2$$

- ❑ λ - коэффициент регуляризации.

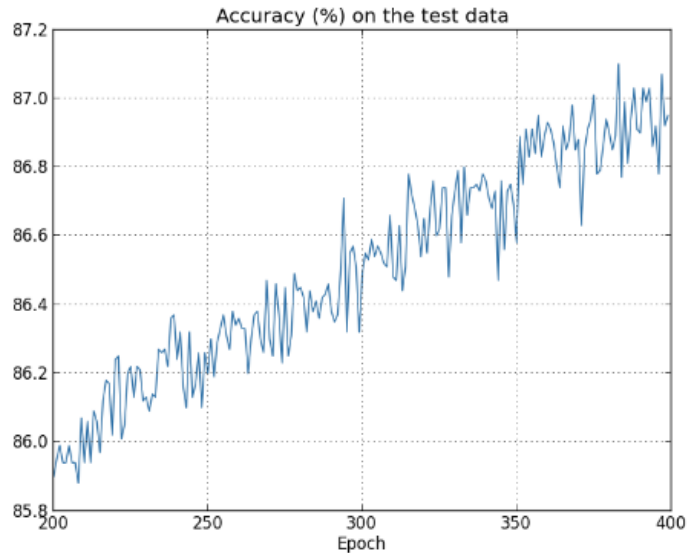
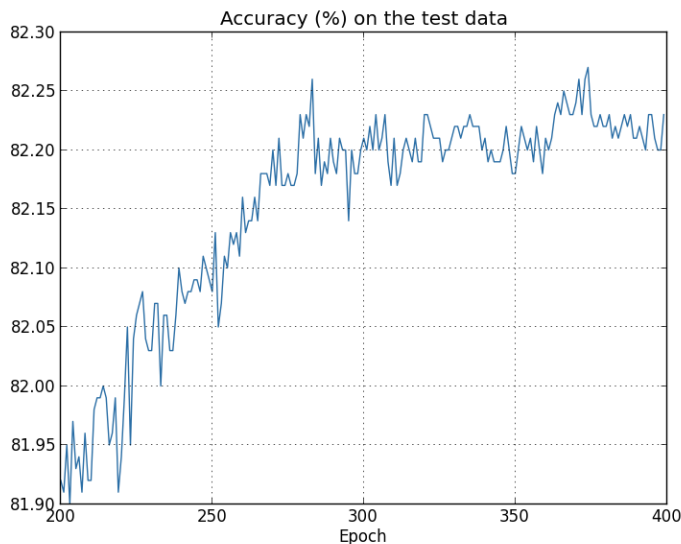
$$\begin{aligned} \frac{\partial C}{\partial w} &= \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w & b &\rightarrow b - \eta \frac{\partial C_0}{\partial b} \\ \frac{\partial C}{\partial b} &= \frac{\partial C_0}{\partial b} & w &\rightarrow w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} w \\ & & &= \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w} \end{aligned}$$

- ❑ Масштабирование веса перед коррекцией по градиентному спуску.

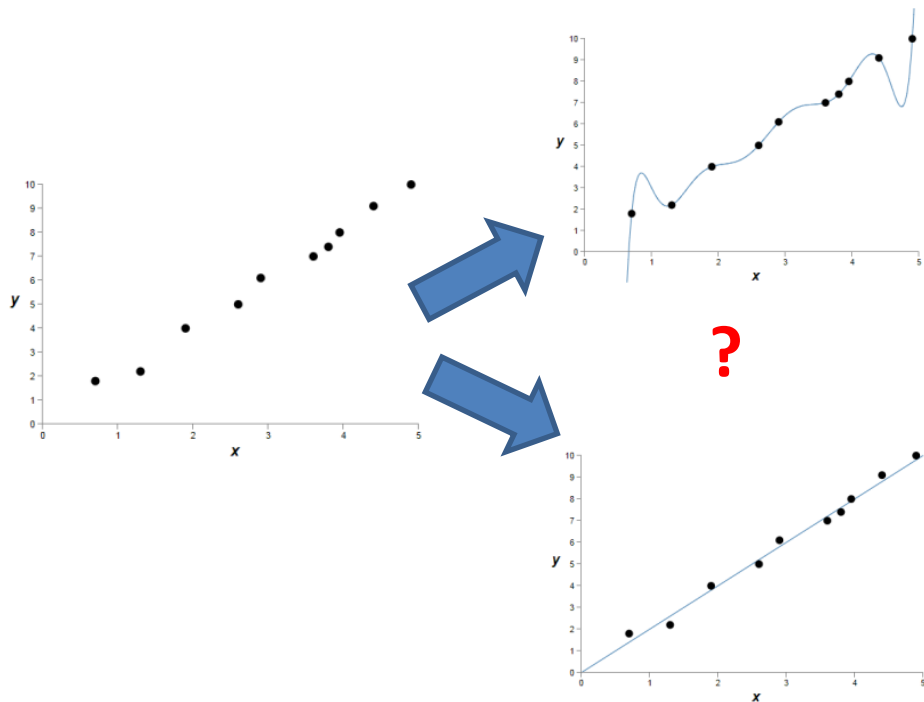
Пример применения регуляризации L2

❑ Классификация MNIST

❑ Сеть 784x30x10, 1000 обучающих примеров



Регуляризация → снижение переобучения



- ❑ Нет однозначного решения без дополнительной информации.
- ❑ Большие значения параметров → увеличение чувствительности к шуму.

$$y = a_0x^9 + a_1x^8 + \dots$$

$$y = a_0x + a_1$$

Регуляризация L1

- ❑ Введение штрафа для больших весов.

$$C = C_0 + \frac{\lambda}{n} \sum_w |w|$$

- ❑ λ - коэффициент регуляризации.

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} \text{sgn}(w) \quad w \rightarrow w' = w - \frac{\eta \lambda}{n} \text{sgn}(w) - \eta \frac{\partial C_0}{\partial w}$$

- ❑ Уменьшение веса на фиксированную величину

- ❑ Для регуляризации L2 значение уменьшения веса зависит от его величины.

$$w \rightarrow w' = w \left(1 - \frac{\eta \lambda}{n} \right) - \eta \frac{\partial C_0}{\partial w}$$

Регуляризация Max-norm

- ❑ Ограничения нормы вектора весов для каждого нейрона.

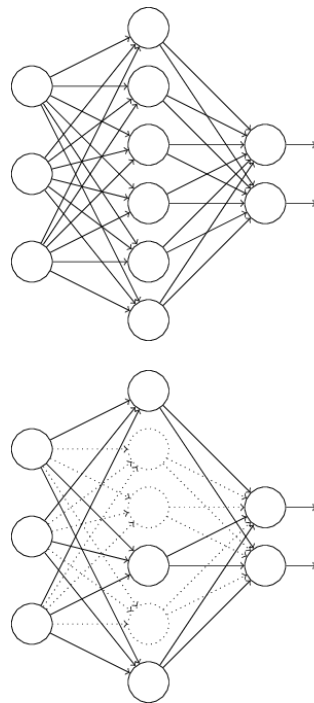
$$\|\tilde{\mathbf{w}}\|_2 \leq c$$

- ❑ c - гиперпараметр.
- ❑ Реализация через нормировку вектора весов при невыполнении неравенства.
- ❑ Эффективна при совместном использовании с Dropout

Dropout

- ❑ Инструмент регуляризации.
- ❑ Модификация архитектуры сети в процессе обучения.
- ❑ Упрощенная схема Dropout
 1. Временно удалить из НС половину случайно выбранных внутренних нейронов с соответствующими связями.
 2. Провести итерацию обучения на пакете: обновление связей оставшихся нейронов.
 3. Восстановить удаленные нейроны и их связи.
 4. Повторить п. 1 – 3.
- ❑ Перед применением сети уменьшить внутренние веса в 2 раза.

N. Srivastava et al. Dropout: A simple way to prevent neural networks from overfitting, 2014.



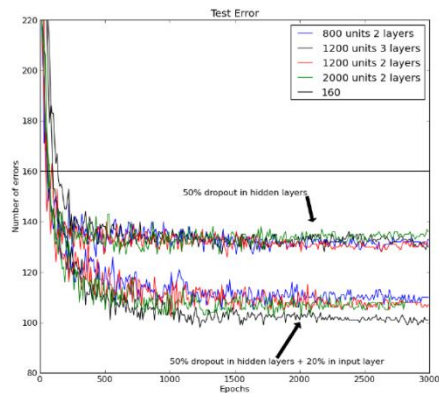
Dropout. Пример использования.

□ MNIST.

Входной вектор 784 элемента.

10 классов.

10000 тестовых изображений.

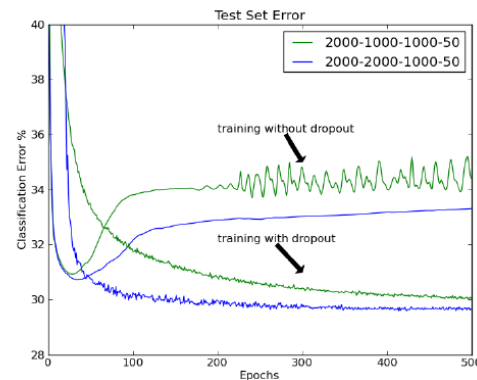
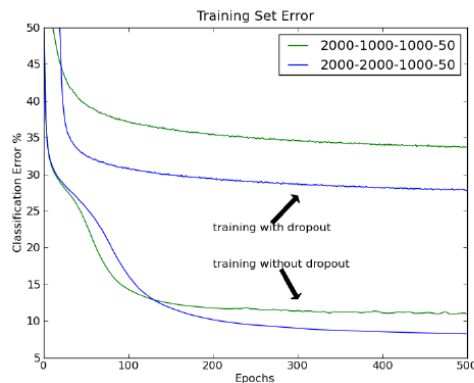


□ Reuters. Классификация документов.

Входной вектор 2000 элементов.

50 классов.

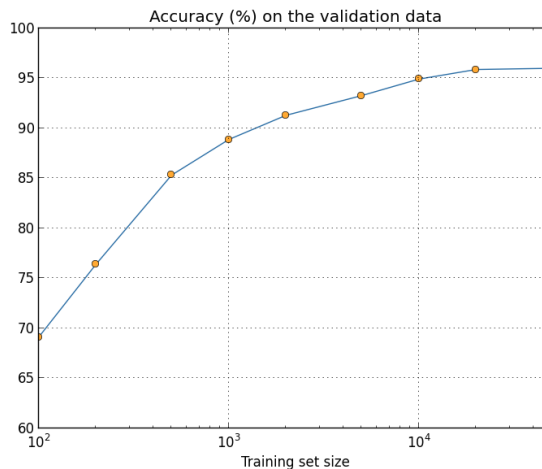
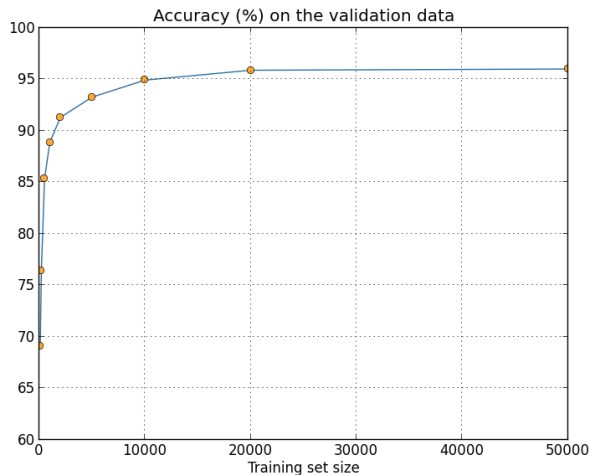
~200000 тестовых документов.



G.E. Hinton et al, Improving neural networks by preventing co-adaptation of feature detectors, 2012

Увеличение обучающей выборки.

- ❑ Способствует росту обобщающей способности НС



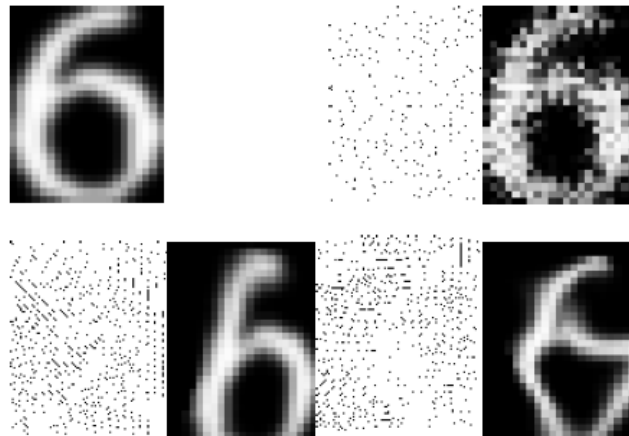
Сеть
784x30x10

- ❑ «Естественное» получение новых примеров часто сопряжено со сложностями

Методы генерации обучающих примеров.

□ MNIST, MLP 784x800x10

Тип искажения	Ошибка на тестовой выборке
Нет	1.6%
аффинные преобразования	1.1%
«эластичные» искажения	0.4%



P. Y. Simard et al. Best practices for convolutional neural networks applied to visual document analysis, 2003

Типы сверток (1)

❑ Входной тензор: $W \times H \times C_I$;

❑ Ядро: $K_1 \times K_2$;

❑ Выходной тензор: $W \times H \times C_O$;

❑ Обычная свертка

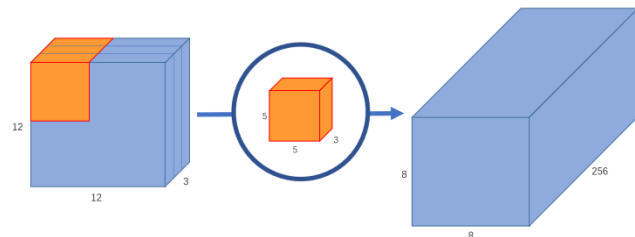
- Число операций: $K_1 \times K_2 \times C_I \times W \times H \times C_O$;

❑ Свертка Depthwise Separable

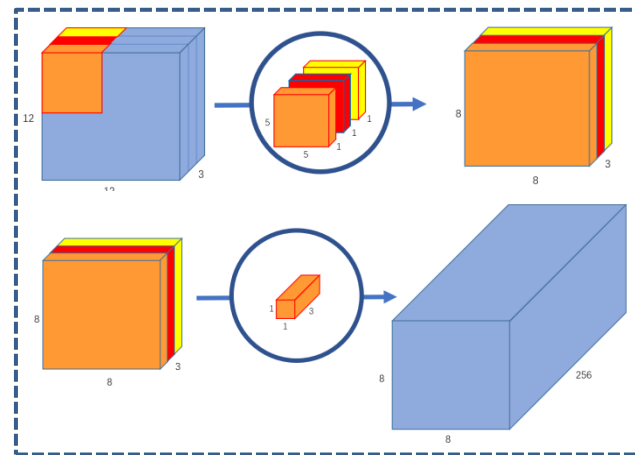
- Depthwise, число операций: $K_1 \times K_2 \times C_I \times W \times H$;
- Pointwise, число операций: $C_I \times W \times H \times C_O$;
- Общее число операций: $K_1 \times K_2 \times C_I \times W \times H + C_I \times W \times H \times C_O$;

+: Вычислительная эффективность

-: Потеря точности



Обычная свертка, 1228800 операций

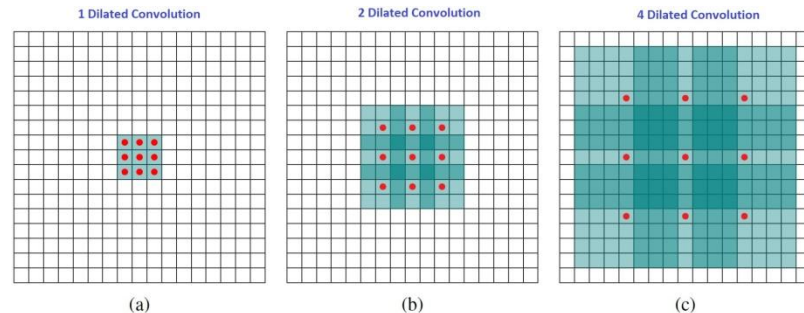


Depthwise Separable, 52952 операций

Типы сверток (2)

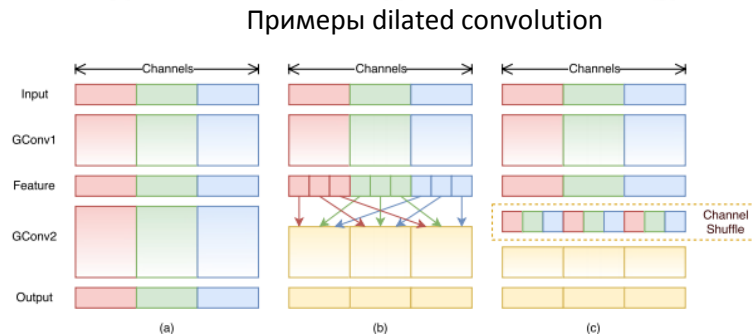
❑ Dilated convolution

- Прореживание рецептивного поля
- Увеличение рецептивного поля с сохранением числа параметров и операций.



❑ Shuffle convolution

- Поканальные свертки с дальнейшим перемешиванием результирующих карт.



а. Поканальная свертка
b, c. Shuffle convolution