

УПРАВЛЕНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

глубокие нейронные сети I

А.В. Макаренко

avm@rdcn.ru

Научно-исследовательская группа «Конструктивная Кибернетика»
Москва, Россия, www.rdcn.ru

Институт проблем управления РАН
Москва, Россия

Учебный курс – Лекция 4

29 марта 2018 г.

ИПУ РАН, Москва, Россия

- ① Глубокое обучение
 - Общие положения

- ② Глубокие нейронные сети
 - Общие положения
 - Составляющие успеха
 - Базовая архитектура сетей I
 - Обучение
 - Реализация

- ③ Заключение

Outline section

- ① Глубокое обучение
 - Общие положения

- ② Глубокие нейронные сети
 - Общие положения
 - Составляющие успеха
 - Базовая архитектура сетей I
 - Обучение
 - Реализация

- ③ Заключение

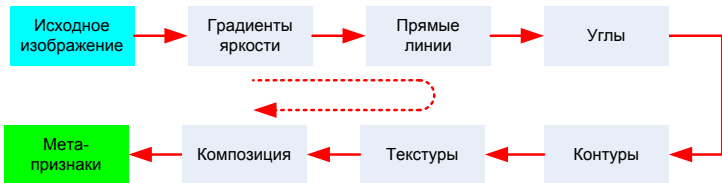
Определение понятия

ГЛУБОКОЕ ОБУЧЕНИЕ – набор алгоритмов машинного обучения, которые пытаются моделировать иерархические абстракции в данных, используя архитектуры, состоящие из каскадного множества нелинейных преобразований (фильтров).

Определение понятия

ГЛУБОКОЕ ОБУЧЕНИЕ – набор алгоритмов машинного обучения, которые пытаются моделировать иерархические абстракции в данных, используя архитектуры, состоящие из каскадного множества нелинейных преобразований (фильтров).

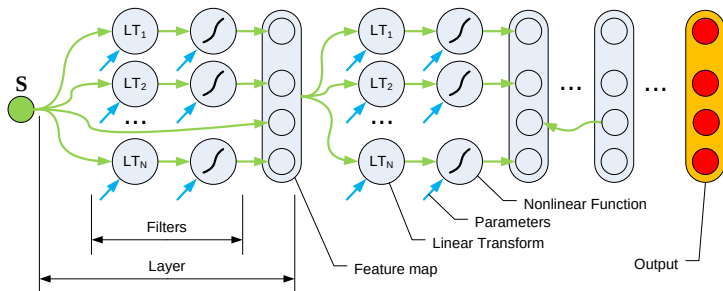
Пример иерархических абстракций в данных (распознавание изображений):



Определение понятия

ГЛУБОКОЕ ОБУЧЕНИЕ – набор алгоритмов машинного обучения, которые пытаются моделировать иерархические абстракции в данных, используя архитектуры, состоящие из каскадного множества нелинейных преобразований (фильтров).

Пример архитектуры алгоритма:

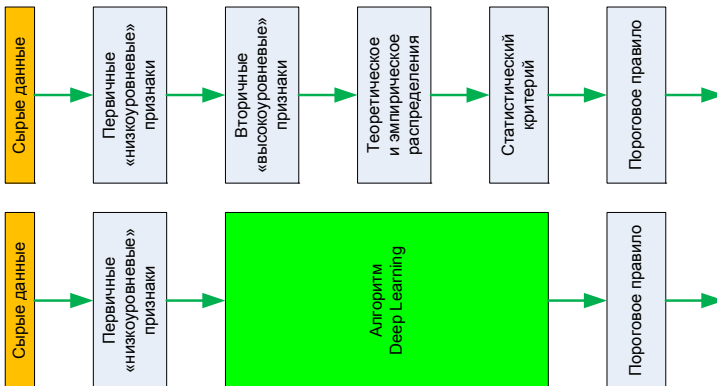


Особенности

Уникальная особенность глубокого обучения – работа с исходными данными (низкоуровневыми признаками) и самостоятельное извлечение (формирование) признаковового описания объектов.

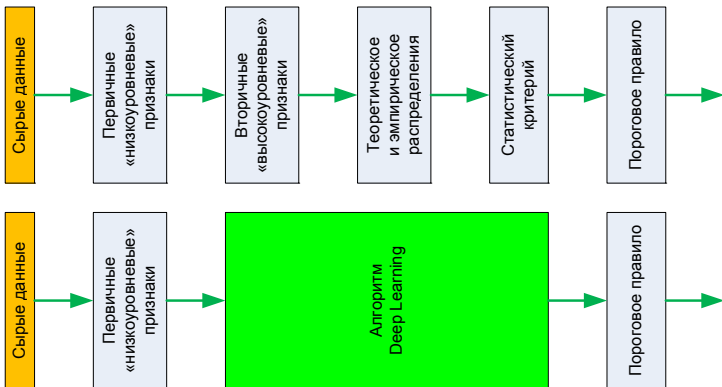
Особенности

Уникальная особенность глубокого обучения – работа с исходными данными (низкоуровневыми признаками) и самостоятельное извлечение (формирование) признаковового описания объектов.



Особенности

Уникальная особенность глубокого обучения – работа с исходными данными (низкоуровневыми признаками) и самостоятельное извлечение (формирование) признаков описания объектов.



МЕТАОБУЧЕНИЕ – программа самостоятельно учится как лучше ей учиться.

История

Автор термина Deep Learning:



Рина Дехтер (Rina Dechter),
род. 1950 г.

R. Dechter, “Learning While Searching in Constraint-Satisfaction-Problems”,
Proceedings of the 5th National
Conference on Artificial Intelligence,
vol. 1. Philadelphia, PA, August 11-15,
1986. [Читать...](#)

История

Автор термина Deep Learning:



Рина Дехтер (Rina Dechter),
род. 1950 г.

R. Dechter, “Learning While Searching in Constraint-Satisfaction-Problems”,
Proceedings of the 5th National
Conference on Artificial Intelligence,
vol. 1. Philadelphia, PA, August 11-15,
1986. [Читать...](#)

Основная парадигма Глубокого Обучения (на данный момент):

Глубокие нейронные сети.

Outline section

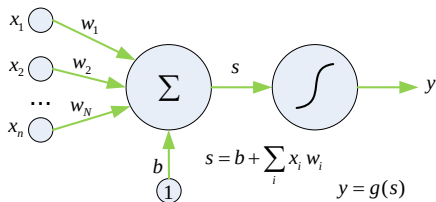
- ① Глубокое обучение
 - Общие положения
- ② Глубокие нейронные сети
 - Общие положения
 - Составляющие успеха
 - Базовая архитектура сетей I
 - Обучение
 - Реализация
- ③ Заключение

Определение понятия

ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ – это многослойные искусственные нейронные сети, с числом внутренних (скрытых) слоёв более одного.

Определение понятия

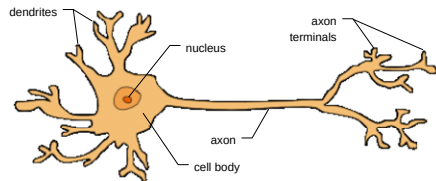
ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ – это многослойные искусственные нейронные сети, с числом внутренних (скрытых) слоёв более одного.



$$g : y = \begin{cases} 1, & s \geq 0, \\ 0, & s < 0. \end{cases} \text{ – персептрон;}$$

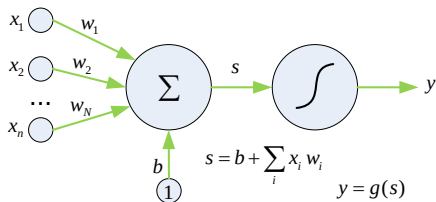
$$g : y = \frac{1}{1 + e^{-\beta s}} \text{ – сигмоидальный;}$$

$$g : y = \begin{cases} 1, & s > 0, \\ -1, & s \leq 0. \end{cases} \text{ – адалайн;}$$



Определение понятия

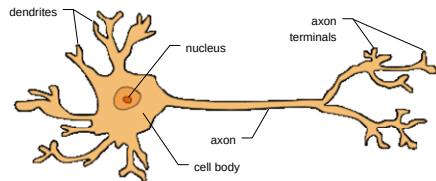
ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ – это многослойные искусственные нейронные сети, с числом внутренних (скрытых) слоёв более одного.



$$g : y = \begin{cases} 1, & s \geq 0, \\ 0, & s < 0. \end{cases} \text{ – персептрон;}$$

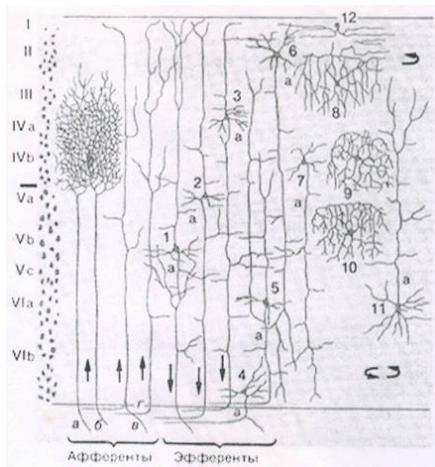
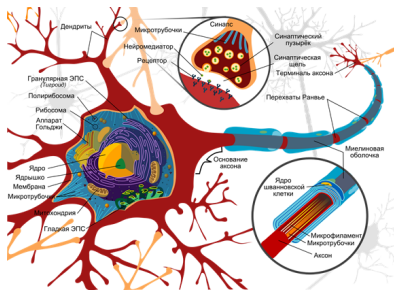
$$g : y = \frac{1}{1 + e^{-\beta s}} \text{ – сигмоидальный;}$$

$$g : y = \begin{cases} 1, & s > 0, \\ -1, & s \leq 0. \end{cases} \text{ – адалайн;}$$



Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002.

Биологические аналоги

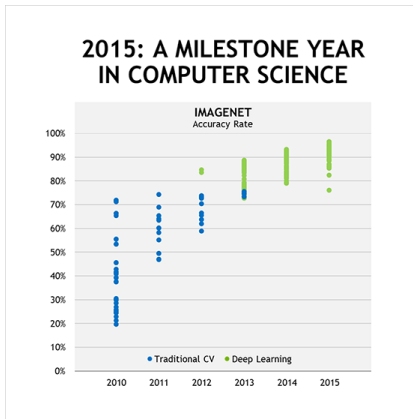


Начало новейшей истории

[ImageNet](#) – база данных аннотированных изображений, предназначенная для отработки и тестирования алгоритмов распознавания образов и машинного зрения. Для категоризации объектов на изображениях используется семантическая сеть [WordNet](#). База данных определяет 1000 классов и по состоянию на 2016 год содержит около 10 млн. изображений.

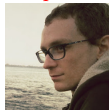
Начало новейшей истории

ImageNet – база данных аннотированных изображений, предназначенная для отработки и тестирования алгоритмов распознавания образов и машинного зрения. Для категоризации объектов на изображениях используется семантическая сеть **WordNet**. База данных определяет 1000 классов и по состоянию на 2016 год содержит около 10 млн. изображений.



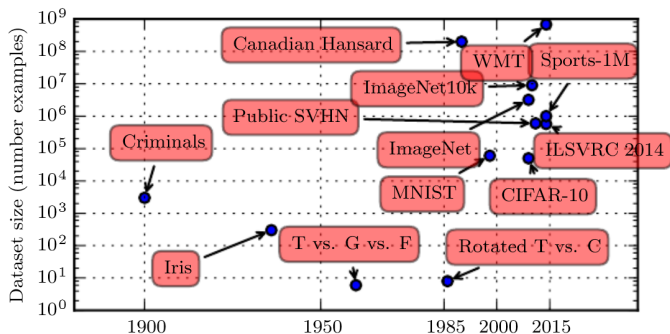
Alex Krizhevsky в 2012 году в соревновании по распознаванию изображений ImageNet применил подход на основе глубоких нейронных сетей. Его сеть AlexNet победила с существенным отрывом. Это дало исходный толчок к буму Deep Learning, который мы наблюдаем в настоящее время.

A. Krizhevsky, I. Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks.



Данные для обучения

Тотальный перевод информации в электронную (машиночитаемую) форму. Наборы данных для обучения и тестирования алгоритмов машинного обучения увеличиваются в размерах, повышается их качество и разнообразие. Стоимость добычи данных существенно снижается.



Источник и расшифровка диаграммы: I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

Аппаратные вычислительные ресурсы



ASCI White (ноябрь 2000 – июнь 2002),
12.3 Тфлопс, 110 миллионов долларов
США, 6 МВт, 106 тонн.



GPU NVIDIA Volta (3-й квартал 2017),
15.0 Тфлопс, 1700 долларов США,
300 Вт, 370 г.

Программное обеспечение

С программным обеспечением в области Deep Learning сложилась уникальная ситуация, кардинально отличающаяся от принятых «правил игры» в других научно-технических областях:

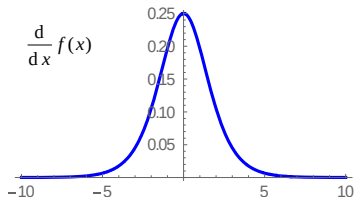
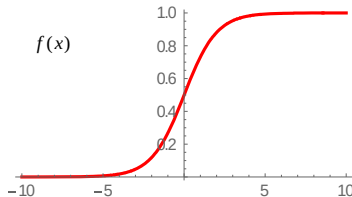
- Подавляющее большинство библиотек и фреймворков – бесплатно.
- Исходный код основных библиотек и фреймворков – открыт.
- Обучающие материалы – бесплатны и свободно доступны.
- Широкие и отзывчивые группы поддержки.

Caffe, Theano, Google TensorFlow, Microsoft CNTK, ...

Решение проблемы затухающего градиента

Активационная функция: Logistic Sigmoid

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



Аргументы «ЗА»:

- Это классика...
- Именно такие профили сигнала в мозгу...
- Функция имеет зоны насыщения
- Сигмоида гладкая и дифференцируема

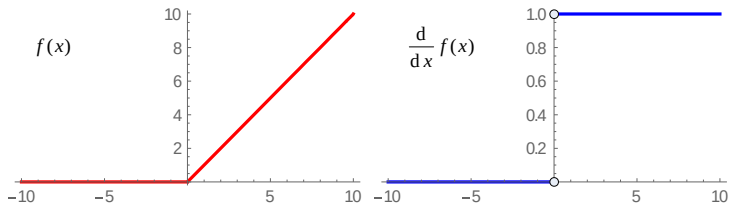
Аргументы «ПРОТИВ»:

- Функция дорогая в вычислительном плане
- Сигмоида имеет малый рабочий отрезок
- И не соответствует профилям сигнала в мозгу...

Решение проблемы затухающего градиента

Активационная функция: ReLU Function

$$f(x) = \begin{cases} x & x > 0, \\ 0 & \text{otherwise.} \end{cases}$$



Аргументы «ЗА»:

- Функция очень дешёвая в вычислительном плане
- ReLU имеет широкий рабочий отрезок

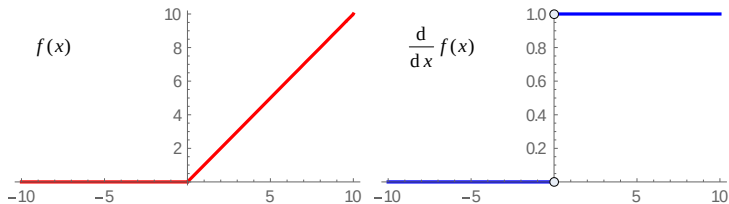
Аргументы «ПРОТИВ»:

- Не дифференцируема в 0
- Не ограничена справа
- И ни чему не соответствует...

Решение проблемы затухающего градиента

Активационная функция: ReLU Function

$$f(x) = \begin{cases} x & x > 0, \\ 0 & \text{otherwise.} \end{cases}$$



Аргументы «ЗА»:

- Функция очень дешёвая в вычислительном плане
- ReLU имеет широкий рабочий отрезок

Аргументы «ПРОТИВ»:

- Не дифференцируема в 0
- Не ограничена справа
- И ни чему не соответствует...

Новая функция – это важный вклад в решение проблемы затухания градиента в глубине сети при её обучении.

Решение проблемы переобучения

Техника **Data augmentation** – расширение обучающей выборки посредством контролируемых искажений исходных данных. Структура искажений определяется характером данных и сутью решаемой задачи.

Решение проблемы переобучения

Техника Data augmentation – расширение обучающей выборки посредством контролируемых искажений исходных данных. Структура искажений определяется характером данных и сутью решаемой задачи.

Техника Dropout – на каждой итерации обучения выкидываем часть нейронов скрытых слоев, вместе с их входящими и исходящими весами, а после завершения итерации – возвращаем. После окончания обучения умножаем все веса на нормализующий коэффициент.

Оригинальная работа: Hinton G.E. et al. Improving neural networks by preventing co-adaptation of feature detectors. ArXiv: [1207.0580](https://arxiv.org/abs/1207.0580).

Решение проблемы переобучения

Техника Data augmentation – расширение обучающей выборки посредством контролируемых искажений исходных данных. Структура искажений определяется характером данных и сутью решаемой задачи.

Техника Dropout – на каждой итерации обучения выкидываем часть нейронов скрытых слоев, вместе с их входящими и исходящими весами, а после завершения итерации – возвращаем. После окончания обучения умножаем все веса на нормализующий коэффициент.

Оригинальная работа: Hinton G.E. et al. Improving neural networks by preventing co-adaptation of feature detectors. ArXiv: [1207.0580](#).

Техника DropConnect – на каждой итерации обучения обнуляем часть входных весов нейронов скрытых слоев, а после завершения итерации – возвращаем. После окончания обучения умножаем все веса на нормализующий коэффициент.

Оригинальная работа: Wan Li et al. Regularization of Neural Networks using DropConnect. [ICML'2013](#).

Решение проблемы мультиклассовых решений

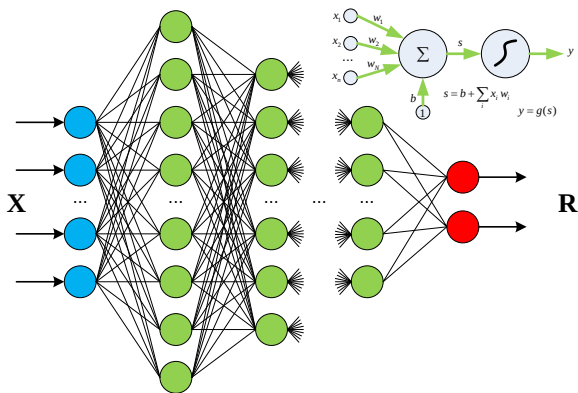
Оценивание апостериорных вероятностей в случае решения задачи мультиклассовой классификации ($K \geq 2$) – через обобщение логистической функции – функция softmax:

$$\sigma(\mathbf{y})_j = \frac{e^{\mathbf{y}_j}}{\sum_{i=1}^K e^{\mathbf{y}_i}}, \quad j = \overline{1, K}$$

См. например: Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006. Читать....

Полносвязные сети

Полносвязная нейронная сеть (Fully Connected Neural Network, FCNN) формируется как последовательная комбинация элементарных нейронов.



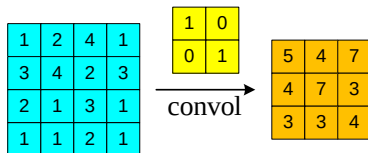
Задаётся: топология сети, активационная функция.

Обучается: веса и смещения нейронов.

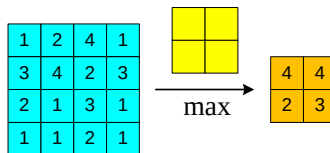
Свёрточные сети I

Свёрточная нейронная сеть (Convolution Neural Network, CNN) формируется как последовательная комбинация свёрточных слоёв и слоёв пуллинга.

Свёрточный слой:



Слой пуллинга:



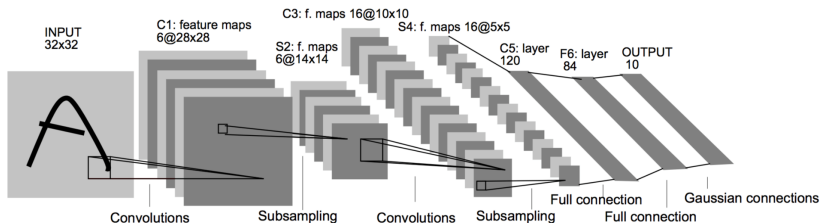
Задаётся: топология сети (topology), размеры (shape), шаг (stride) и отступы (paddings) свёрточных ядер, размеры, шаг, отступы и тип пуллинга (max, average), активационная функция.

Обучается: веса и смещения нейронов в свёрточных слоях.

Техн. информация: Dumoulin V., Visin F. A guide to convolution arithmetic for deep learning. ArXiv: [1603.07285](https://arxiv.org/abs/1603.07285).

Свёрточные сети I

Свёрточная нейронная сеть (Convolution Neural Network, CNN) формируется как последовательная комбинация свёрточных слоёв и слоёв пулинга.



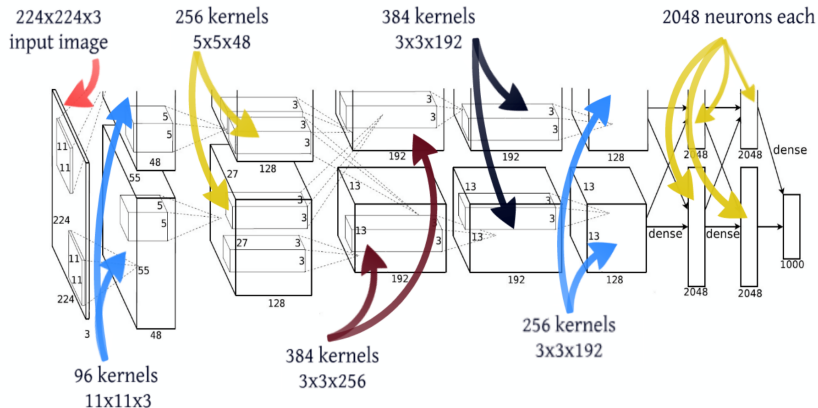
1998 г., Ян ЛеКунн, LeNet, ≈ 60 тыс. обучаемых параметров

Свёртки: 5x5, stride: 1x1; Пулинг: 2x2, stride: 2x2.

Внедрение на почте США для распознавания индексов.

Свёрточные сети I

Свёрточная нейронная сеть (Convolution Neural Network, CNN) формируется как последовательная комбинация свёрточных слоёв и слоёв пулинга.

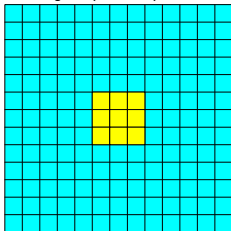


2012 г., Алекс Крижевский, AlexNet, ≈ 60 млн. обучаемых параметров

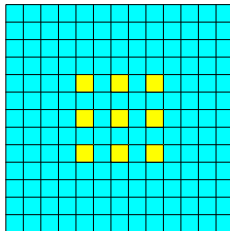
Свёрточные сети II

Dilated convolution:

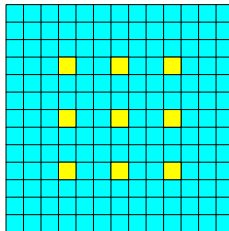
D=1, regular (standard) convol



D=2



D=3

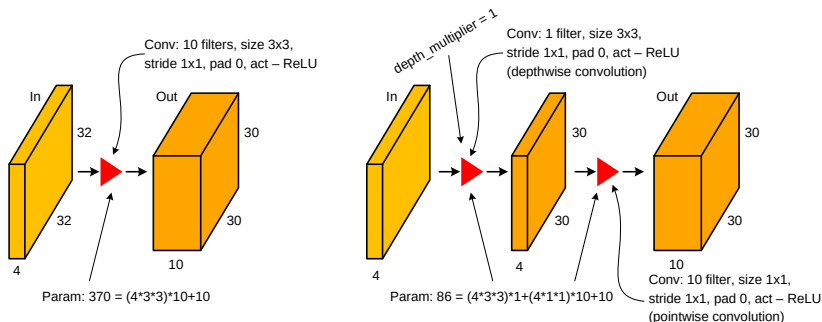


Смысл модификации: расширение рецептивного поля ядра при сохранении кол-ва параметров, ArXiv: [1609.03499](https://arxiv.org/abs/1609.03499).

Примечание: не путать с параметром stride обычной регулярной свёртки.

Свёрточные сети III

Depthwise separable convolution:



Смысл модификации: декомпозиция свёрточных ядер на малоразмерные составляющие (аналог Inception Block).

Свёрточные сети IV

Transposed convolution:

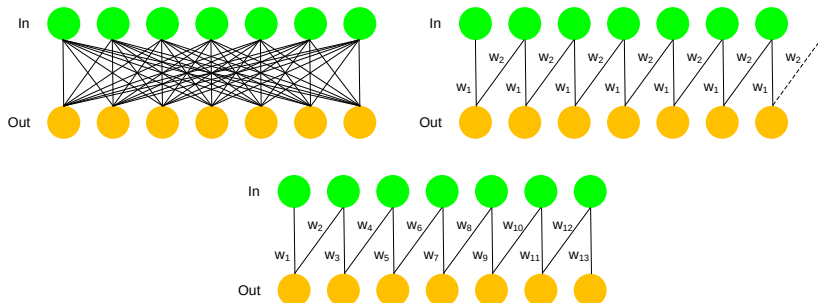
in: 5x5, out: 2x2,
shape: 3x3, stride: 2x2.

in: 2x2, out: 5x5,
shape: 3x3, stride: 2x2.

Смысл модификации: построение «Deconvolutional Networks», [Читать....](#)

Свёрточные сети V

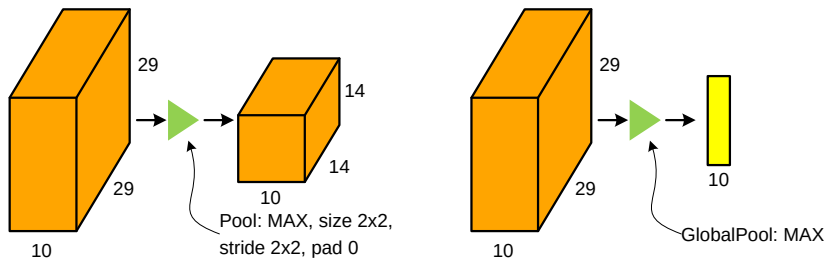
Locally Connected (unshared convolution):



Смысл модификации: поиск локализованных малоразмерных объектов.

Свёрточные сети VI

Global Pooling (Max, Average):

Смысл модификации: построение fully convolved networks, ArXiv: [1312.4400](https://arxiv.org/abs/1312.4400).

Свёрточные сети VII

Batch Normalization¹:

In: $\mathbf{B}$ – мини-батч данных $\mathbf{B} = \{x_1, x_2, \dots, x_M\}$.

Out: $y_i = \text{BN}_{\gamma, \beta}(x_i)$. Обучается: γ, β .

$$\mu_{\mathbf{B}} = \text{M}[\mathbf{B}], \quad \sigma_{\mathbf{B}}^2 = \text{D}[\mathbf{B}], \quad \hat{x}_i = \frac{x_i - \mu_{\mathbf{B}}}{\sigma_{\mathbf{B}} + \varepsilon}, \quad \text{BN}_{\gamma, \beta} : y_i = \gamma \hat{x}_i + \beta.$$

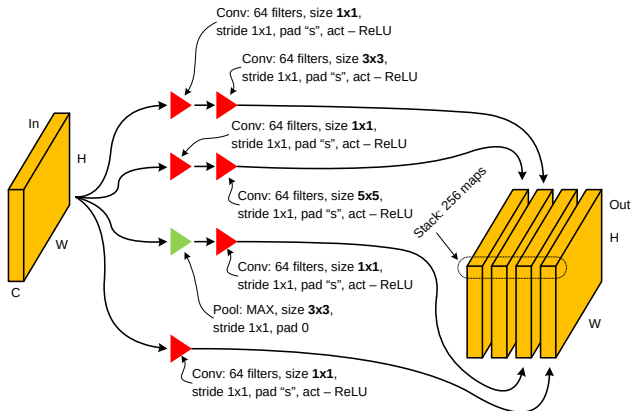
Назначение слоя: ускорение сходимости процесса обучения, возможно использовать более высокие значения *learning rate*, ArXiv: [1502.03167](#).

Примечание: Batch Normalization не является заменой Dropout.

¹Размещение информации об этом слое в разделе свёрточных сетей имеет скорее исторические причины, нежели технические.

Свёрточные сети VIII

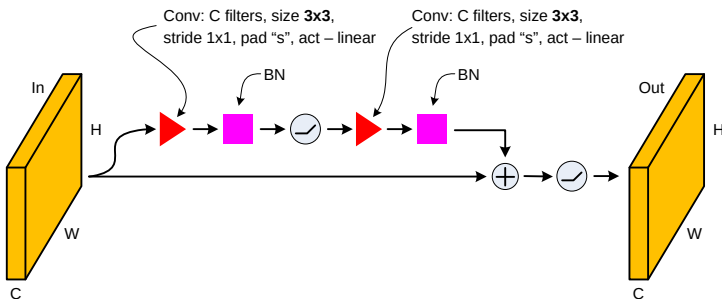
Inception:



Смысл модификации: построение вычислительно «дешёвой» сети с эффективной много-масштабной обработкой, ArXiv:[1409.4842](#), [1512.00567](#).

Свёрточные сети IX

Residual blocks:

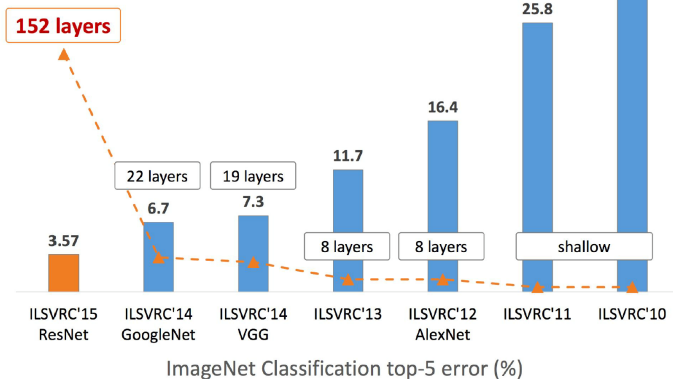


Смысл модификации: облегчение решения оптимизационной задачи при аппроксимировании сложной нелинейной функции, ArXiv: [1512.03385](https://arxiv.org/abs/1512.03385), [1603.05027](https://arxiv.org/abs/1603.05027).

Свёрточные сети IX

Residual blocks:

Revolution of Depth



Свёрточные сети X

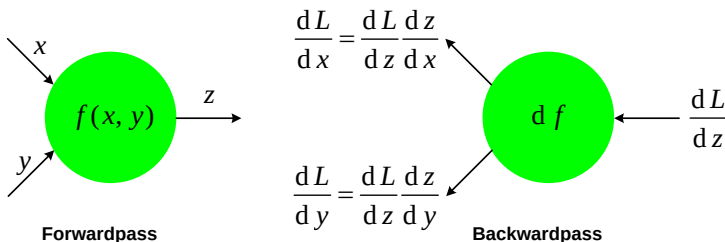
Дальнейшие эксперименты:

- Вставка блоков Residual connection внутрь блоков Residual connection (Resnet in Resnet), ArXiv: [1603.08029](#).
- Сеть строится из совокупности фильтров и нелинейностей по самоподобной архитектуре, при этом идея сквозных Residual connection отвергается, но используется регуляризация по типу DropConnect для отдельных блоков на разных масштабах (FractalNet), ArXiv: [1605.07648](#).
- Сборка ResNet сети из нескольких ResNet сетей (Multilevel Residual Networks), ArXiv: [1608.02908](#).
- Последовательное соединение групп слоёв (5 шт.) через блоки Residual connection (DenseNet), ArXiv: [1608.06993](#).
- Очередная архитектура над блоками Inception из которых формируются блоки Residual connection последовательно объединяемые в сеть (PolyNet), ArXiv: [1611.05725](#).
- ...

Алгоритм обратного распространения ошибки



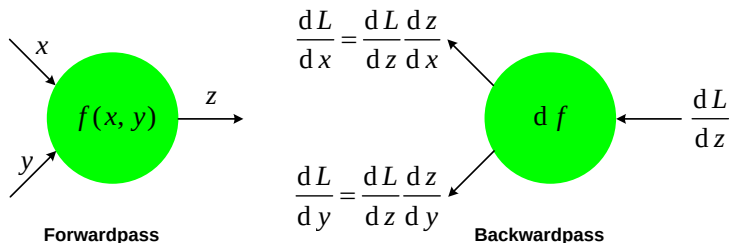
Алгоритм обратного распространения ошибки



$$\delta_{ij}^{(n)} = -\eta \frac{\partial L}{\partial \omega_{ij}^{(n)}}, \quad \omega_{ij}^{(n)}|_{k+1} = \omega_{ij}^{(n)}|_k + \delta_{ij}^{(n)}.$$

n – номер слоя, j – нейрон n -го слоя, i – нейрон $n - 1$ -го слоя

Алгоритм обратного распространения ошибки



$$\delta_{ij}^{(n)} = -\eta \frac{\partial L}{\partial \omega_{ij}^{(n)}}, \quad \omega_{ij}^{(n)}|_{k+1} = \omega_{ij}^{(n)}|_k + \delta_{ij}^{(n)}.$$

n – номер слоя, j – нейрон n -го слоя, i – нейрон $n - 1$ -го слоя

$$L = -\sum_{j=1}^M z'_j \ln z_j, \quad Q = -\frac{1}{N^b} \sum_{n=1}^{N^b} \sum_{j=1}^M z'_j{}^{(n)} \ln z_j^{(n)}, \quad z_i = \frac{e^{y_i}}{\sum_{j=1}^M e^{y_j}},$$

$$\frac{\partial L}{\partial z_j} = -\frac{z'_j}{z_j}, \quad \frac{\partial L}{\partial y_i} = z_i - z'_i.$$

Алгоритм обратного распространения ошибки

$$L = - \sum_{j=1}^M z'_j \ln z_j, \quad Q = - \frac{1}{N^b} \sum_{n=1}^{N^b} \sum_{j=1}^M z_j'^{(n)} \ln z_j^{(n)}, \quad z_i = \frac{e^{y_i}}{\sum_{j=1}^M e^{y_j}},$$

$$\frac{\partial L}{\partial z_j} = - \frac{z'_j}{z_j}, \quad \frac{\partial L}{\partial y_i} = z_i - z'_i.$$

$N^b = 1$ – online (стохастическое) обучение;

$N^b = N^{\text{Tr}}$ – full-batch обучение;

$1 < N^b < N^{\text{Tr}}$ – mini-batch обучение.

Алгоритм обратного распространения ошибки

$$L = - \sum_{j=1}^M z'_j \ln z_j, \quad Q = - \frac{1}{N^b} \sum_{n=1}^{N^b} \sum_{j=1}^M z_j'^{(n)} \ln z_j^{(n)}, \quad z_i = \frac{e^{y_i}}{\sum_{j=1}^M e^{y_j}},$$

$$\frac{\partial L}{\partial z_j} = - \frac{z'_j}{z_j}, \quad \frac{\partial L}{\partial y_i} = z_i - z'_i.$$

$N^b = 1$ – online (стохастическое) обучение;

$N^b = N^{\text{Tr}}$ – full-batch обучение;

$1 < N^b < N^{\text{Tr}}$ – mini-batch обучение.

Описание различных оптимизаторов: Sebastian Ruder, An overview of gradient descent optimization algorithms. [Читать...](#)

Keras: CNN-MNIST I

Описание нейросети:

```
47 model = Sequential()
48 model.add(Conv2D(32, kernel_size = (3, 3),
49                 activation = 'relu',
50                 input_shape = input_shape))
51 model.add(Conv2D(64, (3, 3), activation = 'relu'))
52 model.add(MaxPooling2D(pool_size = (2, 2)))
53 model.add(Dropout(0.25))
54 model.add(Flatten())
55 model.add(Dense(128, activation = 'relu'))
56 model.add(Dropout(0.5))
57 model.add(Dense(num_classes, activation = 'softmax'))
```

Пример взят из официального репозитория Keras: [GitHub](#).

Keras: CNN-MNIST II

Компилирование вычислительного графа:

```
59 model.compile(loss = keras.losses.categorical_crossentropy,  
60               optimizer = keras.optimizers.Adadelta(),  
61               metrics = ['accuracy'])
```

Пример взят из официального репозитория Keras: [GitHub](#).

Keras: CNN-MNIST II

Компилирование вычислительного графа:

```
59 model.compile(loss = keras.losses.categorical_crossentropy,  
60               optimizer = keras.optimizers.Adadelta(),  
61               metrics = ['accuracy'])
```

Запуск процесса обучения сети:

```
63 model.fit(x_train, y_train,  
64          batch_size = batch_size,  
65          epochs = epochs,  
66          verbose = 1,  
67          validation_data = (x_test, y_test))
```

Пример взят из официального репозитория Keras: [GitHub](#).

Keras: CNN-MNIST III

Тестирование сети:

```
68 score = model.evaluate(x_test, y_test, verbose = 0)
69 print('Test loss:', score[0])
70 print('Test accuracy:', score[1])
```

Пример взят из официального репозитория Keras: [GitHub](#).

Outline section

- ❶ Глубокое обучение
 - Общие положения

- ❷ Глубокие нейронные сети
 - Общие положения
 - Составляющие успеха
 - Базовая архитектура сетей I
 - Обучение
 - Реализация

- ❸ Заключение

Контрольная работа

Задание для слушателей:

- 1 Скачать и исследовать структуру и статистические свойства датасета fashion-mnist: [GitHub](#).
- 2 Установить фреймворк Keras: [keras.io](#).
- 3 Изучить построение моделей в **Sequential** и **Model** режимах.
- 4 Изучить разделы: [Core Layers](#), [Convolutional Layers](#), [Pooling Layers](#), [Merge Layers](#), [Losses](#), [Metrics](#), [Optimizers](#), [Activations](#), [FAQ](#).
- 5 Переделать пример [CNN-MNIST](#) под датасет fashion-mnist.
- 6 Внедрить в переделанную сеть Inception модуль, исследовать показатели качества нейросети (Confusion Matrix, F1-мера).
- 7 Попробовать добиться максимальных показателей качества нейросети (по F1-мере).