Сверточные нейронные сети

Вячеслав Пономарев

Высшая школа экономики
Факультет компьютерных наук

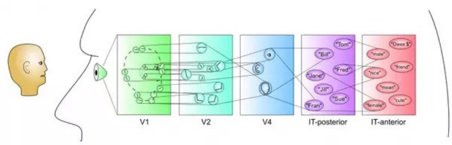
18 октября 2018 г.

Содержание

Введение

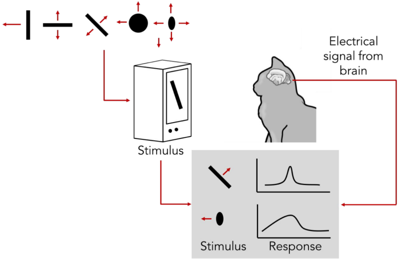
Введение

Human visual cortex



Source: https://towardsdatascience.com/understanding-convolutional-neural-networks-221930904a8e

Hubel & Wiesel experiments

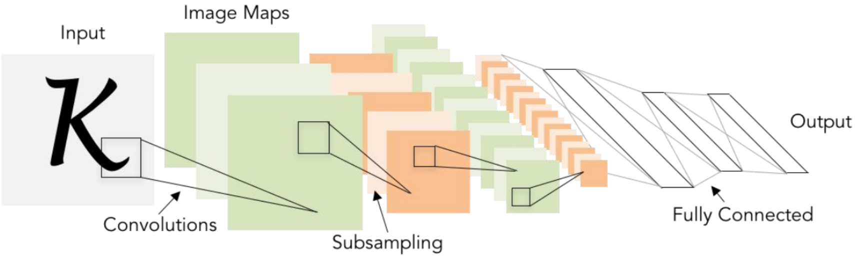


Source: Stanford CS class CS231n slides

Введение

The first Convolutional Neural Network – LeNet-5

LeCun, Bottou, Bengio, Haffner 1998



Source: Stanford CS class CS231n slides

Введение

Почему не стоит использовать полносвязную архитектуру для изображений?

- CIFAR-10 - картинки размера 32x32x3

Введение

Почему не стоит использовать полносвязную архитектуру для изображений?

- CIFAR-10 - картинки размера 32x32x3
- Уже один нейрон на первом слое полносвязной нейронной сети будет иметь $32 \cdot 32 \cdot 3 + 1 = 3073$ параметров

Введение

Почему не стоит использовать полносвязную архитектуру для изображений?

- CIFAR-10 - картинки размера 32x32x3
- Уже один нейрон на первом слое полносвязной нейронной сети будет иметь $32 \cdot 32 \cdot 3 + 1 = 3073$ параметров
- Что-то более приличное по размеру - 200x200x3

Введение

Почему не стоит использовать полносвязную архитектуру для изображений?

- CIFAR-10 - картинки размера 32x32x3
- Уже один нейрон на первом слое полносвязной нейронной сети будет иметь $32 \cdot 32 \cdot 3 + 1 = 3073$ параметров
- Что-то более приличное по размеру - 200x200x3
- $200 \cdot 200 \cdot 3 + 1 = 120\,001$ параметров

Введение

Почему не стоит использовать полносвязную архитектуру для изображений?

- CIFAR-10 - картинки размера 32x32x3
- Уже один нейрон на первом слое полносвязной нейронной сети будет иметь $32 \cdot 32 \cdot 3 + 1 = 3073$ параметров
- Что-то более приличное по размеру - 200x200x3
- $200 \cdot 200 \cdot 3 + 1 = 120\ 001$ параметров
- 256 нейронов в слое - 30 720 256 параметров

Архитектура

Сверточный слой: фильтры

Основной компонент - фильтр свертки:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Receptive field or filter size **F** - 3x3

Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Сверточный слой: фильтры

Как работает фильтр свертки?
2D - пример:

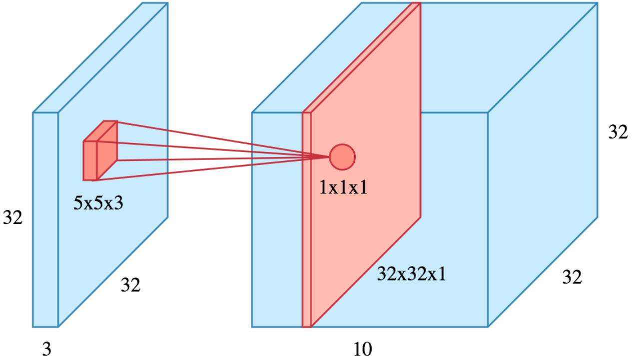
1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Сверточный слой: фильтры

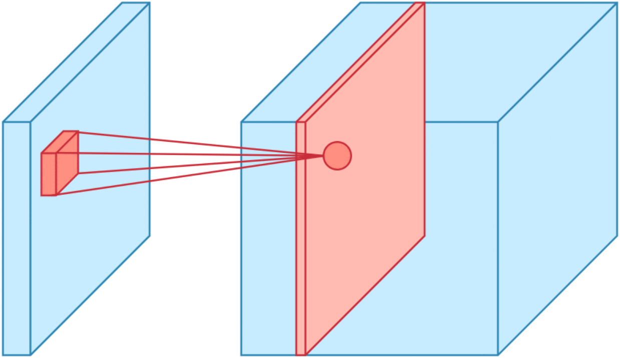
Связь нейронов с входным слоем локальная *(local connectivity)*:



Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Сверточный слой: фильтры

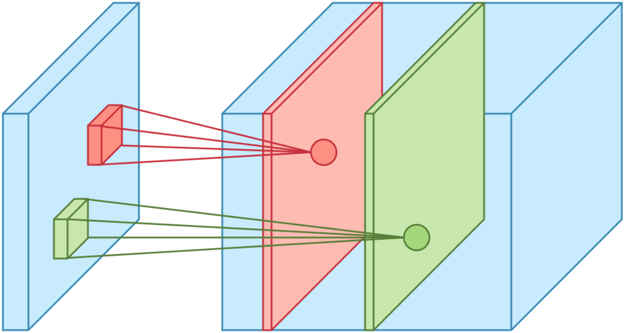
Операция свертки в действии:



Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Сверточный слой: фильтры

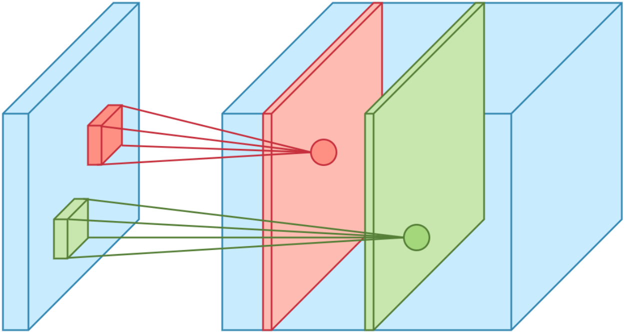
- Выход сверточного слоя - карты признаков:



Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Сверточный слой: фильтры

- Выход сверточного слоя - карты признаков:

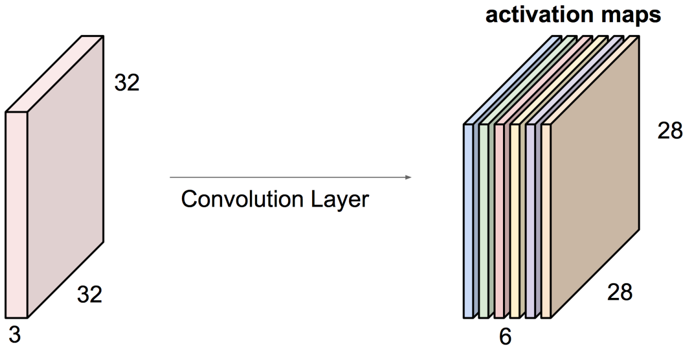


Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

- Какие размеры у выхода?

Сверточный слой: глубина выхода

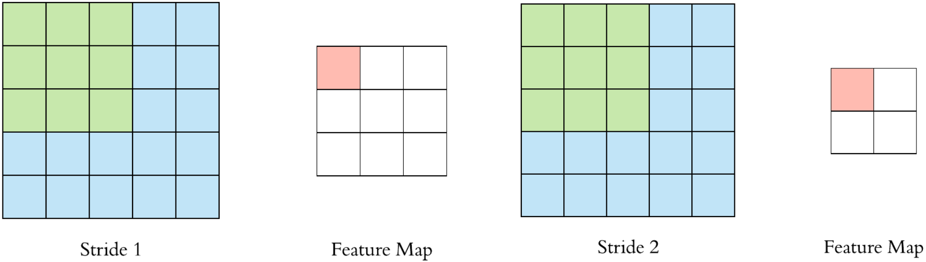
Количество фильтров $\mathbf{K} \in \mathbb{N}$



Source: Stanford CS class CS231n slides

Сверточный слой: отступы

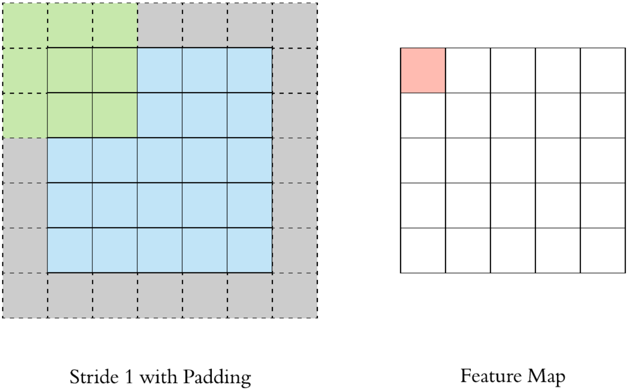
Stride $\mathbf{S} \in \{1, 2, 3, \dots\}$



Source:
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Сверточный слой: заполнение нулями

Padding $\mathbf{P} \in \{0, 1, 2, \ldots\}$



Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Сверточный слой: размер выхода

Пусть $\mathbf{W_1} \times \mathbf{H_1} \times \mathbf{D_1}$ - размер входного слоя

- $\mathbf{F}$ - размер фильтра
- $\mathbf{K}$ - количество фильтров
- $\mathbf{P}$ - padding
- $\mathbf{S}$ - stride

Тогда размер выходного слоя - $\mathbf{W_2} \times \mathbf{H_2} \times \mathbf{D_2}$

- $\mathbf{W_2} = \frac{\mathbf{W_1} - \mathbf{F} + \mathbf{2P}}{\mathbf{S}} + 1$
- $\mathbf{H_2} = \frac{\mathbf{H_1} - \mathbf{F} + \mathbf{2P}}{\mathbf{S}} + 1$
- $\mathbf{D_2} = \mathbf{K}$

Сверточный слой: количество параметров

Наивная схема:

- Вход - [227x227x3]
- $\mathbf{F} = 11$, $\mathbf{S} = 4$, $\mathbf{P} = 0$, $\mathbf{K} = 96$
- Выход - [55x55x96]
- Всего $55 \cdot 55 \cdot 96 = 290{,}400$ нейронов, каждый соединен с областью размера [11x11x3]
- Каждый нейрон имеет $11 \cdot 11 \cdot 3 + 1 = 364$ параметра
- Всего $290400 \cdot 364 = 105{,}705{,}600$ параметров

Сверточный слой: количество параметров

Parameter sharing:

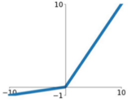
- Вход - [227x227x3]
- $\mathbf{F} = 11$, $\mathbf{S} = 4$, $\mathbf{P} = 0$, $\mathbf{K} = 96$
- Выход - [55x55x96]
- Все нейроны на одном *depth slice* используют одни и те же веса
- Всего 96 различных наборов весов
- Итого $96 \cdot 11 \cdot 11 \cdot 3 + 96 = 34{,}944$ параметра

Сверточный слой: нелинейность

Activation Functions

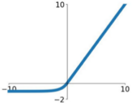
Leaky ReLU

$\max(0.1x, x)$



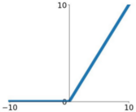
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



ReLU

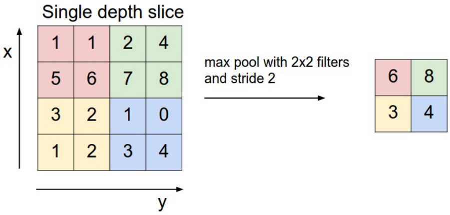
$\max(0, x)$

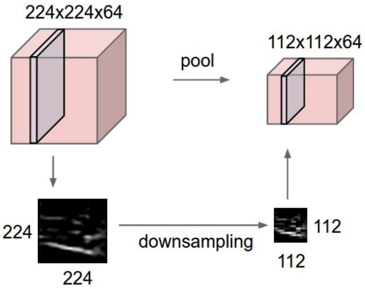


Source: Stanford CS class CS231n slides

Обычно используют:

- Max pooling
- Average pooling





Source: Stanford CS class CS231n slides

Downsampling

Как это работает?

Пусть $\mathbf{W_1} \times \mathbf{H_1} \times \mathbf{D_1}$ - размер входного слоя

- $\mathbf{F}$ - размер окна
- $\mathbf{S}$ - stride

Тогда размер выходного слоя - $\mathbf{W_2} \times \mathbf{H_2} \times \mathbf{D_2}$

- $\mathbf{W_2} = \frac{\mathbf{W_1 - F}}{\mathbf{S}} + 1$
- $\mathbf{H_2} = \frac{\mathbf{H_1 - F}}{\mathbf{S}} + 1$
- $\mathbf{D_2} = \mathbf{D_1}$

Downsampling

Зачем?

- Наличие признака важнее чем его точная позиция
- Понижаем размерность выходного слоя
- Уменьшаем количество обучаемых параметров сети
- Тем самым уменьшаем время обучения
- Контролируем переобучение

Downsampling

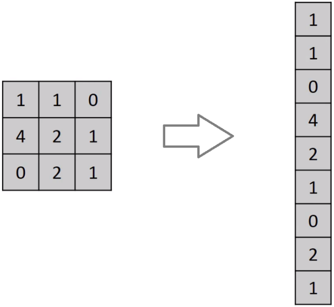
Зачем?

- Наличие признака важнее чем его точная позиция
- Понижаем размерность выходного слоя
- Уменьшаем количество обучаемых параметров сети
- Тем самым уменьшаем время обучения
- Контролируем переобучение

Впрочем, есть подходы к проектированию сети без слоя pooling:
Striving for Simplicity: The All Convolutional Net
[Springenberg, Dosovitskiy, Brox, Martin, 2015]

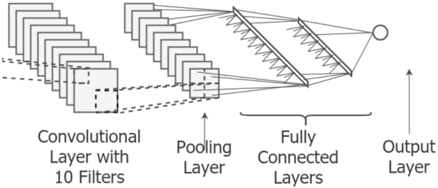
Полносвязные слои

Flatten



Source: https://rubikscode.net/2018/02/26/introduction-to-convolutional-neural-networks

Последние слои CNN



Source: https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-convolution-neural-networks-e3f054dd5daa

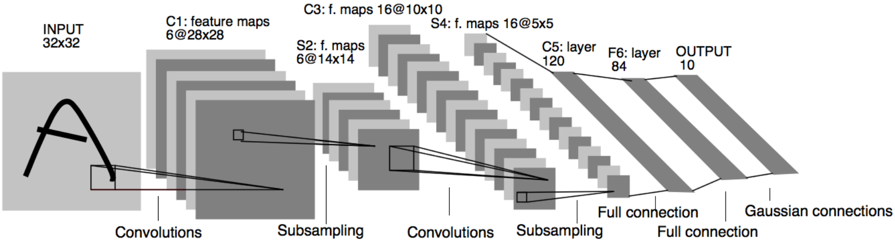
Стандартная архитектура целиком

INPUT→[[CONV→RELU]*N→POOL?]*M →[FC→RELU]*K→FC

Стандартная архитектура целиком

INPUT→[[CONV→RELU]*N→POOL?]*M →[FC→RELU]*K→FC

LeNet-5 architecture:



Source: Gradient-based learning applied to document recognition [Y. Lecun et al., 1998]

Обучение

Backpropagation: градиент по фильтру

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \textbf{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

Backpropagation: градиент по фильтру

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{22}}$$

Source: https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Backpropagation: градиент по фильтру

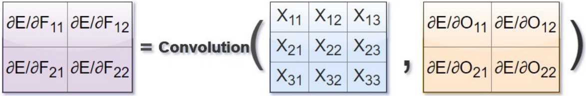
$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$
$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$
$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}X_{11} + \frac{\partial E}{\partial O_{12}}X_{12} + \frac{\partial E}{\partial O_{21}}X_{21} + \frac{\partial E}{\partial O_{22}}X_{22}$$
$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}X_{12} + \frac{\partial E}{\partial O_{12}}X_{13} + \frac{\partial E}{\partial O_{21}}X_{22} + \frac{\partial E}{\partial O_{22}}X_{23}$$
$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}X_{21} + \frac{\partial E}{\partial O_{12}}X_{22} + \frac{\partial E}{\partial O_{21}}X_{31} + \frac{\partial E}{\partial O_{22}}X_{32}$$
$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}}X_{22} + \frac{\partial E}{\partial O_{12}}X_{23} + \frac{\partial E}{\partial O_{21}}X_{32} + \frac{\partial E}{\partial O_{22}}X_{33}$$

Source: `https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e`

Backpropagation: градиент по фильтру



$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}} X_{11} + \frac{\partial E}{\partial O_{12}} X_{12} + \frac{\partial E}{\partial O_{21}} X_{21} + \frac{\partial E}{\partial O_{22}} X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}} X_{12} + \frac{\partial E}{\partial O_{12}} X_{13} + \frac{\partial E}{\partial O_{21}} X_{22} + \frac{\partial E}{\partial O_{22}} X_{23}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}} X_{21} + \frac{\partial E}{\partial O_{12}} X_{22} + \frac{\partial E}{\partial O_{21}} X_{31} + \frac{\partial E}{\partial O_{22}} X_{32}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}} X_{22} + \frac{\partial E}{\partial O_{12}} X_{23} + \frac{\partial E}{\partial O_{21}} X_{32} + \frac{\partial E}{\partial O_{22}} X_{33}$$

Source: https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$
$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$
$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}}F_{11} + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$
$$\frac{\partial E}{\partial X_{12}} = \frac{\partial E}{\partial O_{11}}F_{12} + \frac{\partial E}{\partial O_{12}}F_{11} + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$
$$\frac{\partial E}{\partial X_{13}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}F_{12} + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$
$$\frac{\partial E}{\partial X_{21}} = \frac{\partial E}{\partial O_{11}}F_{21} + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}F_{11} + \frac{\partial E}{\partial O_{22}}0$$
$$\frac{\partial E}{\partial X_{22}} = \frac{\partial E}{\partial O_{11}}F_{22} + \frac{\partial E}{\partial O_{12}}F_{21} + \frac{\partial E}{\partial O_{21}}f_{12} + \frac{\partial E}{\partial O_{22}}F_{11}$$
$$\frac{\partial E}{\partial X_{23}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}F_{22} + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}F_{11}$$
$$\frac{\partial E}{\partial X_{31}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}F_{21} + \frac{\partial E}{\partial O_{22}}0$$
$$\frac{\partial E}{\partial X_{32}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}F_{22} + \frac{\partial E}{\partial O_{22}}F_{21}$$
$$\frac{\partial E}{\partial X_{33}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}F_{22}$$

Source:
https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Backpropagation: градиент по входу

$$\begin{bmatrix} \partial E/\partial X_{11} & \partial E/\partial X_{12} & \partial E/\partial X_{13} \\ \partial E/\partial X_{21} & \partial E/\partial X_{22} & \partial E/\partial X_{23} \\ \partial E/\partial X_{31} & \partial E/\partial X_{32} & \partial E/\partial X_{33} \end{bmatrix} = \textbf{Full\_Convolution}\left(\begin{bmatrix} \partial E/\partial O_{11} & \partial E/\partial O_{12} \\ \partial E/\partial O_{21} & \partial E/\partial O_{22} \end{bmatrix}, \begin{bmatrix} F_{22} & F_{21} \\ F_{12} & F_{11} \end{bmatrix} \right)$$

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}} F_{11} + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{12}} = \frac{\partial E}{\partial O_{11}} F_{12} + \frac{\partial E}{\partial O_{12}} F_{11} + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{13}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} F_{12} + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{21}} = \frac{\partial E}{\partial O_{11}} F_{21} + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} F_{11} + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{22}} = \frac{\partial E}{\partial O_{11}} F_{22} + \frac{\partial E}{\partial O_{12}} F_{21} + \frac{\partial E}{\partial O_{21}} f_{12} + \frac{\partial E}{\partial O_{22}} F_{11}$$

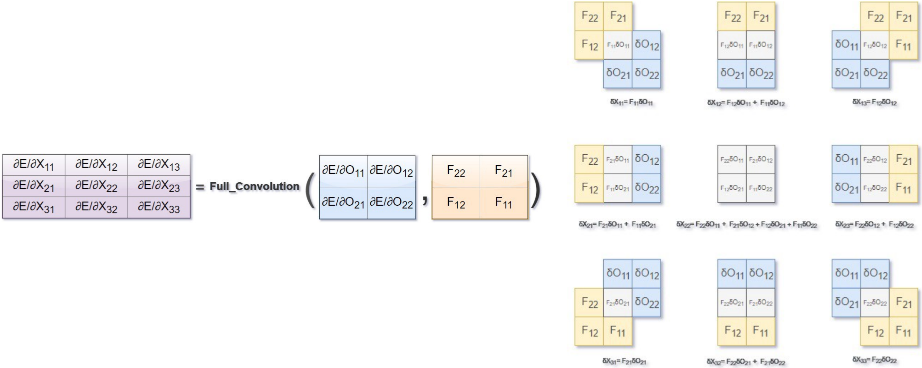
$$\frac{\partial E}{\partial X_{23}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} F_{22} + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} F_{11}$$

$$\frac{\partial E}{\partial X_{31}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} F_{21} + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{32}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} F_{22} + \frac{\partial E}{\partial O_{22}} F_{21}$$

$$\frac{\partial E}{\partial X_{33}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} F_{22}$$

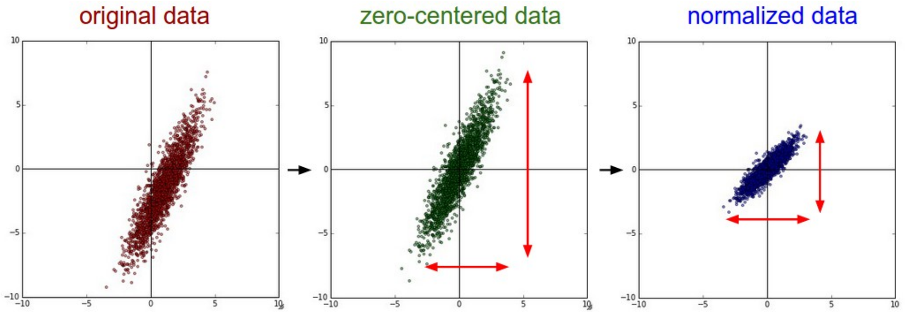
Source:
https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e



Source:
https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Предобработка данных

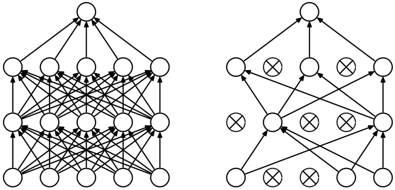
- Mean subtraction
- Normalization
- PCA / Whitening - не используются в CNN



Source: Stanford CS class CS231n slides

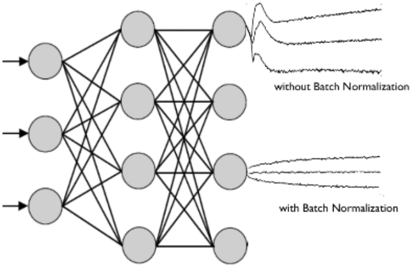
Регуляризация

Dropout


(a) Standard Neural Net (b) After applying dropout.

Source: Dropout: A Simple Way to Prevent Neural Networks from Overfitting [N. Srivastava et al., 2014]

Batch Normalization



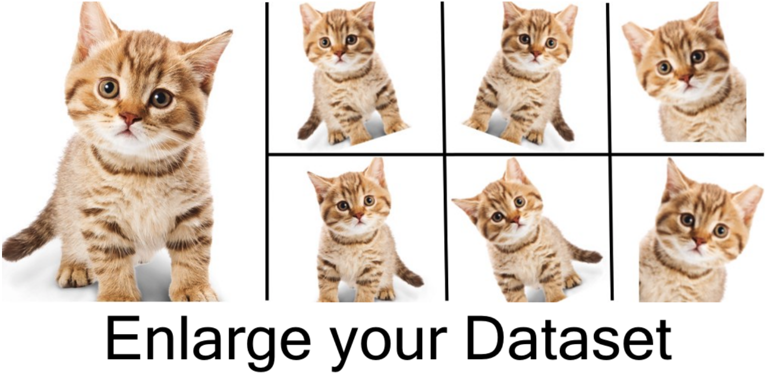
Source: https://calculatedcontent.com/2017/06/16/normalization-in-deep-learning/

Data augmentation

- Придумали "state-of-the-art" архитектуру под задачу?
- Но очень мало данных?
- Выход есть!

Data augmentation

- Придумали "state-of-the-art" архитектуру под задачу?
- Но очень мало данных?
- Выход есть!



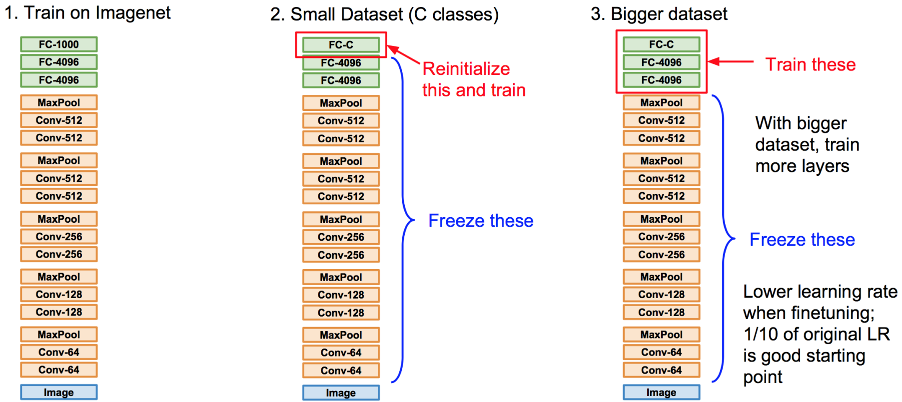
Source: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Data augmentation

Популярные техники:

- Flip
- Rotation
- Scale
- Crop
- Translation
- Color jitter (Randomize contrast and brightness)
- Gaussian Noise
- and so on...

Transfer learning

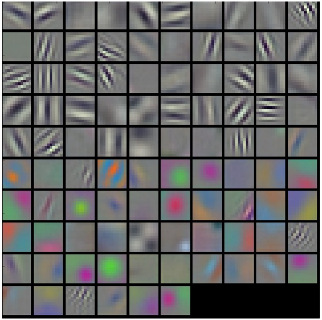


Source: Stanford CS class CS231n slides

Визуализация

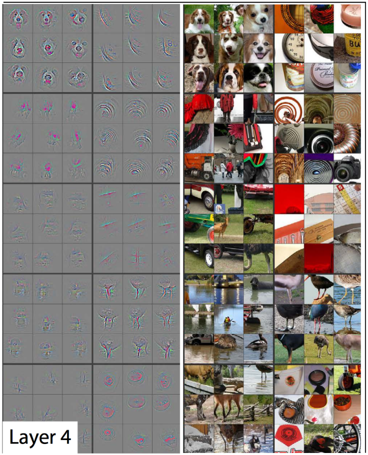
Визуализация

Фильтры первого слоя



Source: Stanford CS class CS231n slides

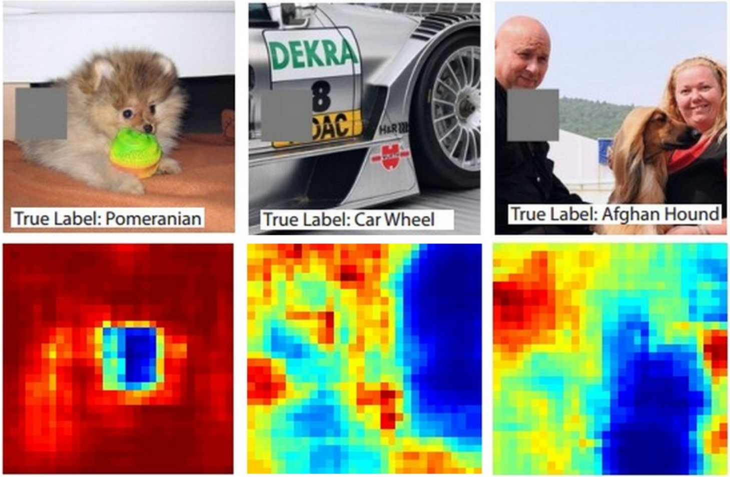
Reconstructed patterns that cause high activations



Source: Visualizing and Understanding Convolutional Networks [Matthew D. Zeiler, Rob Fergus, 2013]

Визуализация

Probability of the correct class as a heatmap



Source: Visualizing and Understanding Convolutional Networks [Matthew D. Zeiler, Rob Fergus, 2013]

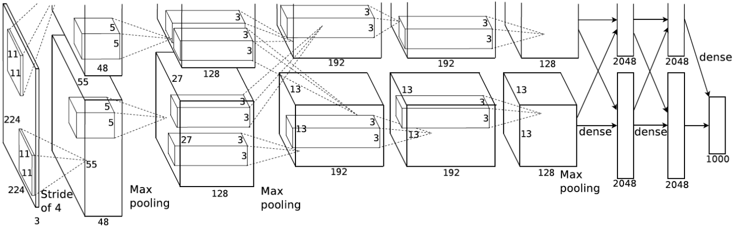
Современные архитектуры

Современные архитектуры

- ImageNet – 14,197,122 images, 21841 synsets indexed
- ILSVRC – ImageNet Large Scale Visual Recognition Challenge (1000 classes)

AlexNet

- 60M parameters
- ReLU activations
- Dropout
- Data augmentation
- 2 GPUs / 1 week
- ILSVRC-2012 winner - top-5 error of 15.3%



Source: ImageNet Classification with Deep Convolutional Neural Networks [Alex Krizhevsky et al., 2012]

VGGNet

- Simple - 3x3 filters only
- Deep - 16/19 layers
- Data augmentation
- 4 GPUs / 2-3 weeks
- ILSVRC-2014 2nd place - top-5 7.3% error

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

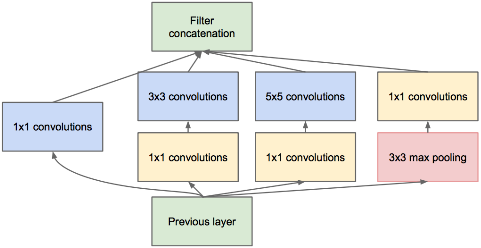
Table 2: **Number of parameters** (in millions).

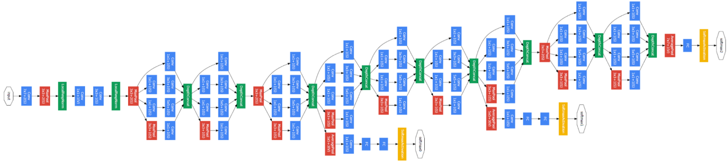
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Source: Very Deep Convolutional Networks for Large-Scale Image Recognition [Simonyan & Zisserman, 2014]

GoogleNet

- 9 inception modules
- Very deep ≈ 100 layers in total
- ≈5M parameters
- "Few high-end GPUs" / 1 week
- ILSVRC-2014 winner - top-5 6.7% error



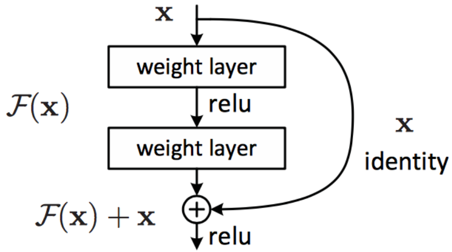


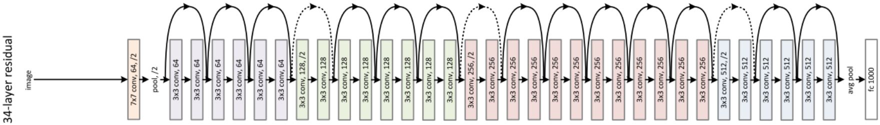
Source: Going Deeper with Convolutions [C. Szegedy1, W. Liu et al., 2015]

"Well said, Leo, well said"

Microsoft ResNet

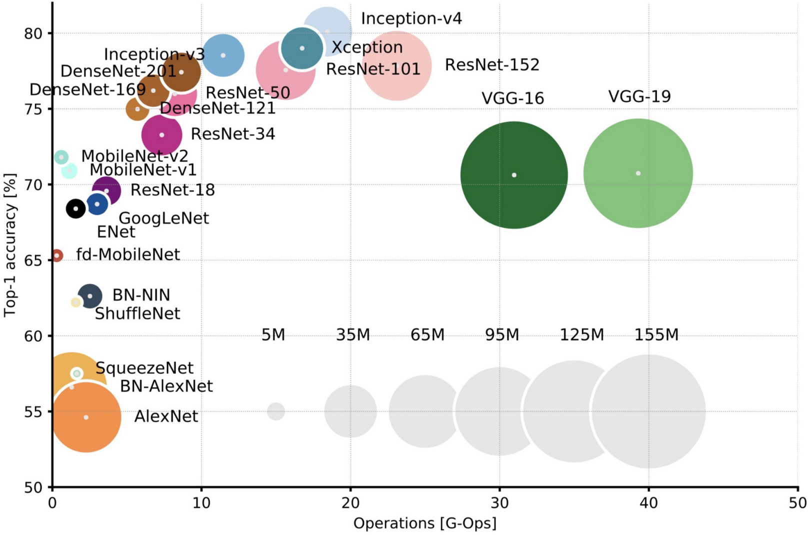
- 152 layers (tried 1202 layers)
- Residual block
- 8 GPUs / 2-3 weeks
- ILSVRC-2015 winner - top-5 3.57% error





Source: Deep Residual Learning for Image Recognition [Kaiming He, Xiangyu Zhang et al., 2015]

Сравнение

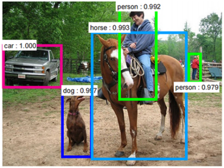


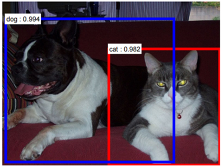
Source: https://towardsdatascience.com/neural-network-architectures-156e5bad51ba

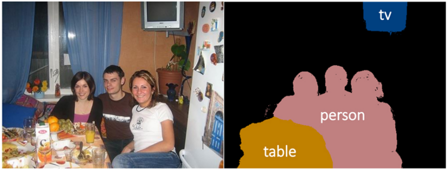
Области применения

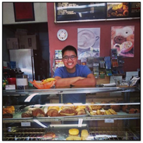
Области применения

- Image Classification
- Object Detection
- Video Classification
- Object Tracking
- Edge Detection
- Semantic Segmentation
- Human Pose Estimation
- Image Captioning
- Question Answering (based on image)
- Image Generation









What kind of store is this?	bakery bakery pastry	art supplies grocery grocery
Is the display case as full as it could be?	no no no	no yes yes

How many bikes are there?	2 2 2	3 4 12
What number is the bus?	48 48 48	4 46 number 6

man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

boy is doing backflip on wakeboard.

Source: `https://skymind.ai/wiki/convolutional-network`

Итоги

- Основные компоненты CNN - сверточный слой, pooling слои, полносвязные слои
- В одном сверточном слое используют набор фильтров
- Градиент по матрице весов или входного слоя можно вычислить, применив свертку
- Batch Normalization и Data augmentation для регуляризации
- Transfer learning – способ добиться хорошего качество на малом объеме данных
- Современные архитектуры очень глубокие, применяются нестандартные подходы к проектированию

Источники

- Stanford CS class CS231n lecture notes
- Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2
- The 9 Deep Learning Papers You Need To Know About, https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html
- Forward And Backpropagation in Convolutional Neural Network, https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e