Deep learning for Computer Vision

Dr Kien Nguyen
Email: k.nguyenthanh@qut.edu.au
Queensland University of Technology, Australia

Warning

Deep learning is new SEXY (for a reason!)

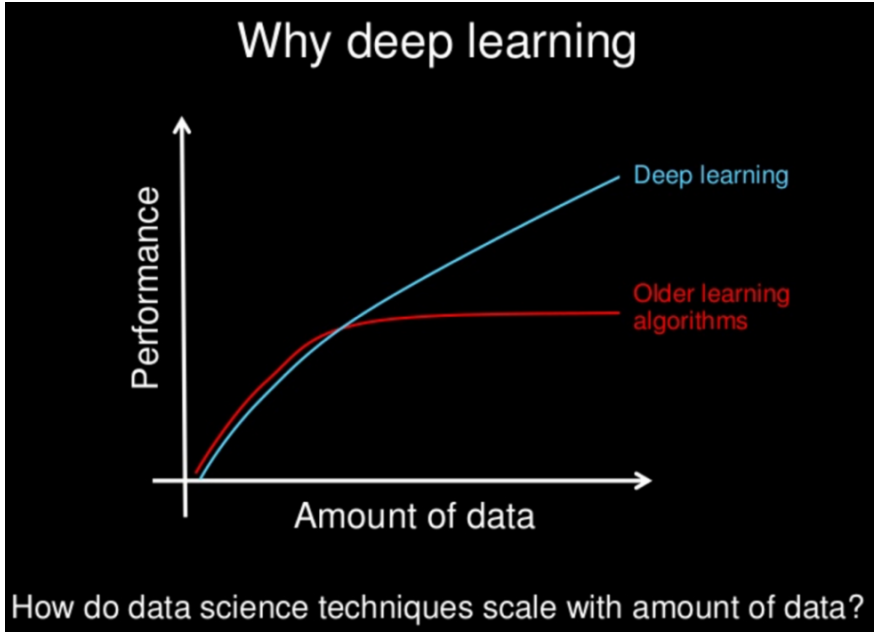
Outline

I. Why, What, Where
II. Deep learning Models
III. Fundamental blocks

Why - deep learning for CV?

Handcrafted feature representation (SIFT, SURF, Bispectral, LBP, HOG, ...) => laborious, require domain expert knowledge, no guarantee for optimal

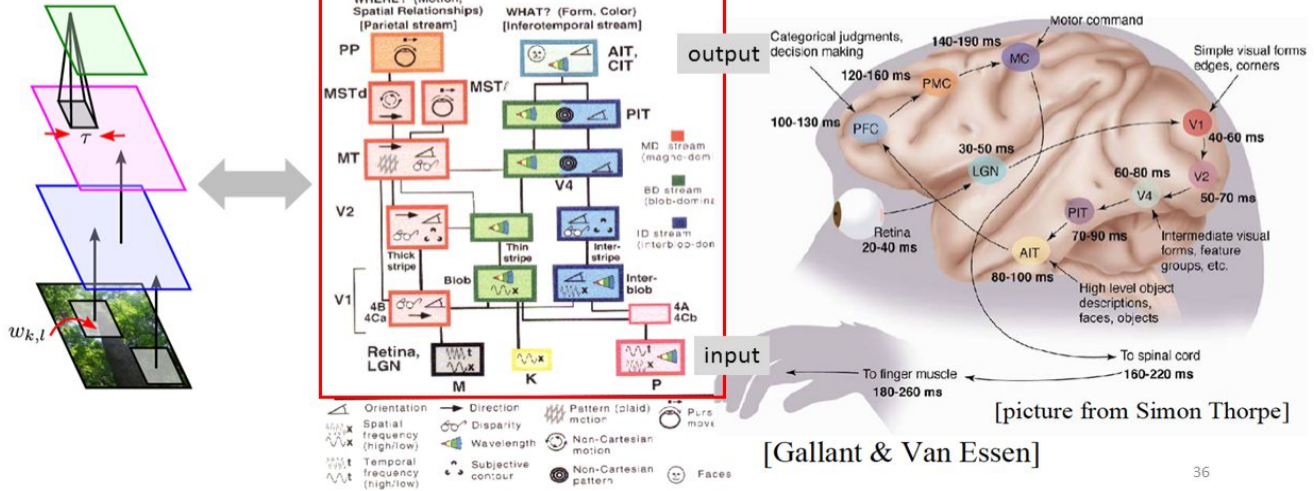
Deep learning: automatically learn feature representation directly from training data, let the data speak for themselves



What - is deep learning?

Definition: *Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks*

The Mammalian Visual Cortex Inspires CNN



What - is deep learning?

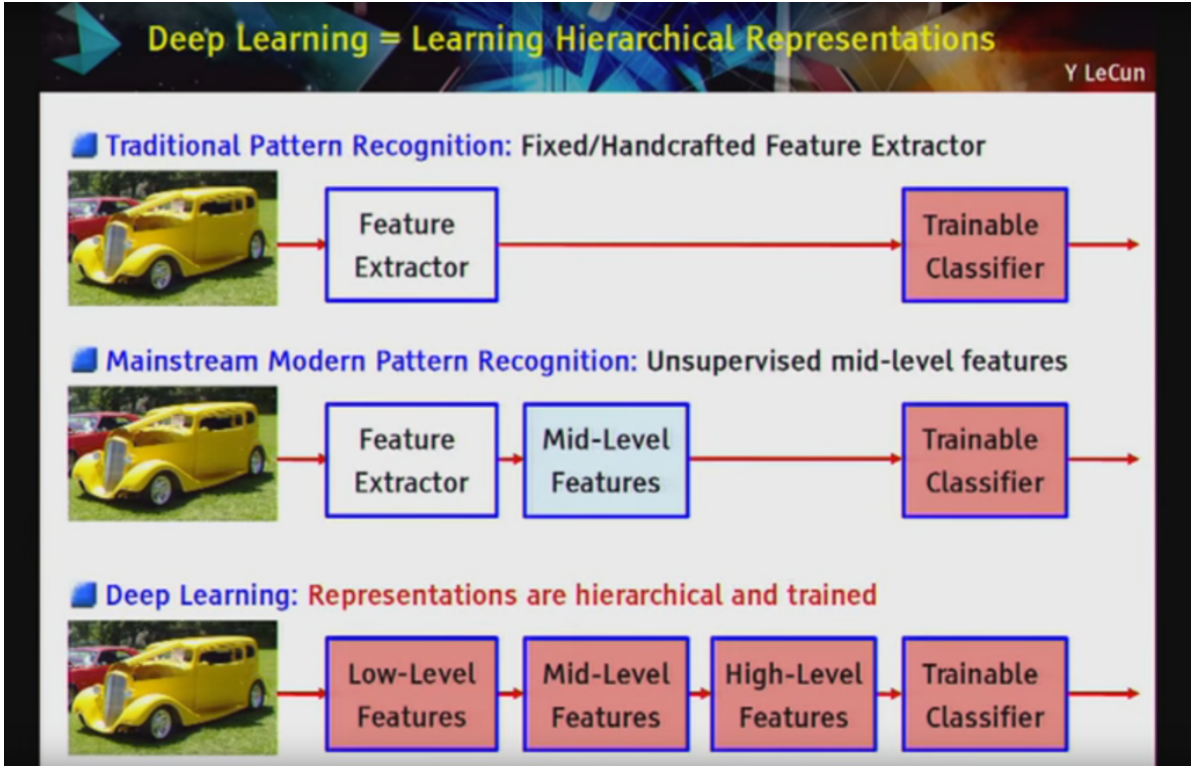
Deep learning is just large neural networks with:

- (1) more data,
- (2) more computation,
- (3) better algorithms

What - is deep learning?

Definition 1: Yann LeCun, “Accelerating Understanding: Deep Learning, Intelligent Applications, and GPUs”, 2016.

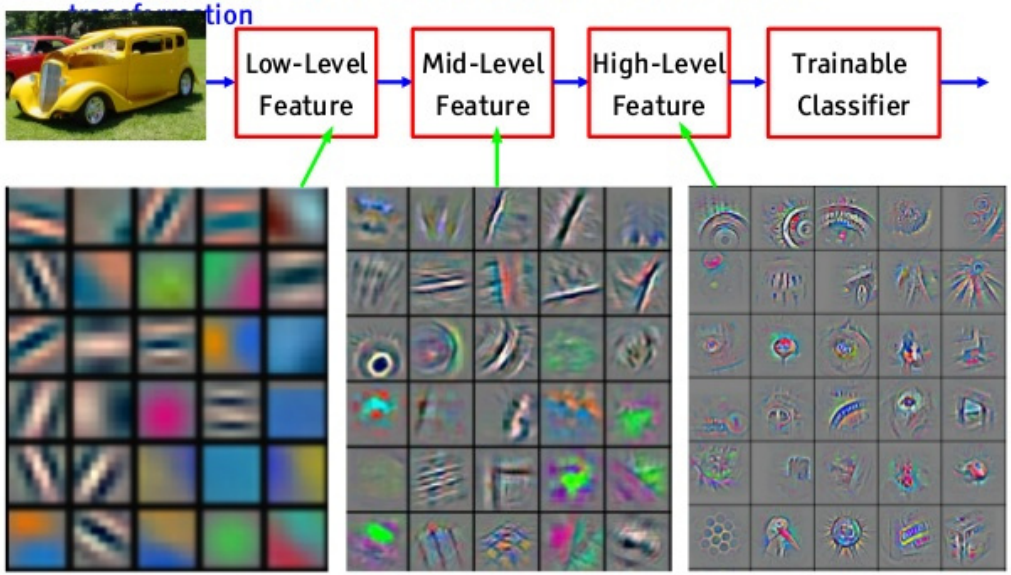
Deep learning [is] … a pipeline of modules all of which are trainable. … deep because [has] multiple stages in the process of recognizing an object and all of those stages are part of the training



What - is deep learning?

Definition 2: Yoshua Bengio, “Deep Learning of Representations for Unsupervised and Transfer Learning”, JMLR, 2012.

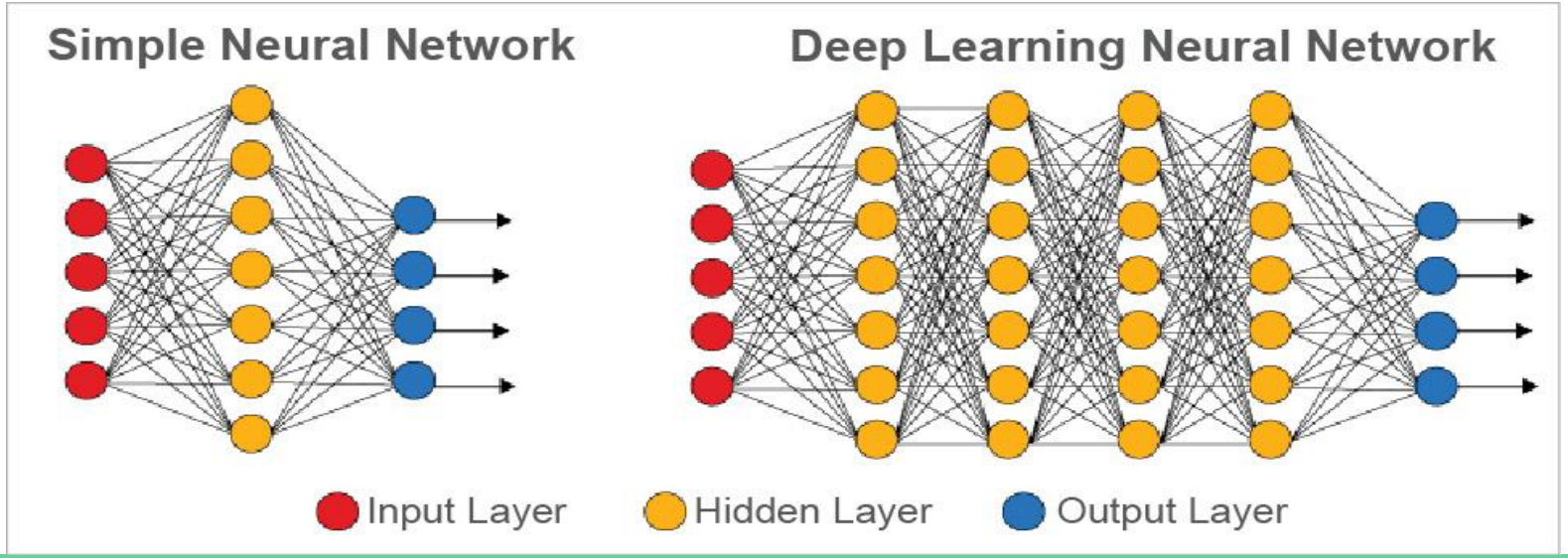
Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features



What - is deep learning?

Definition 3: Jurgen Schmidhuber, “Deep Learning in Neural Networks: An Overview”, arxiv 2014.

At which problem depth does Shallow Learning end, and Deep Learning begin? Discussions with DL experts have not yet yielded a conclusive response to this question. […], let me just define for the purposes of this overview: problems of depth > 10 require Very Deep Learning



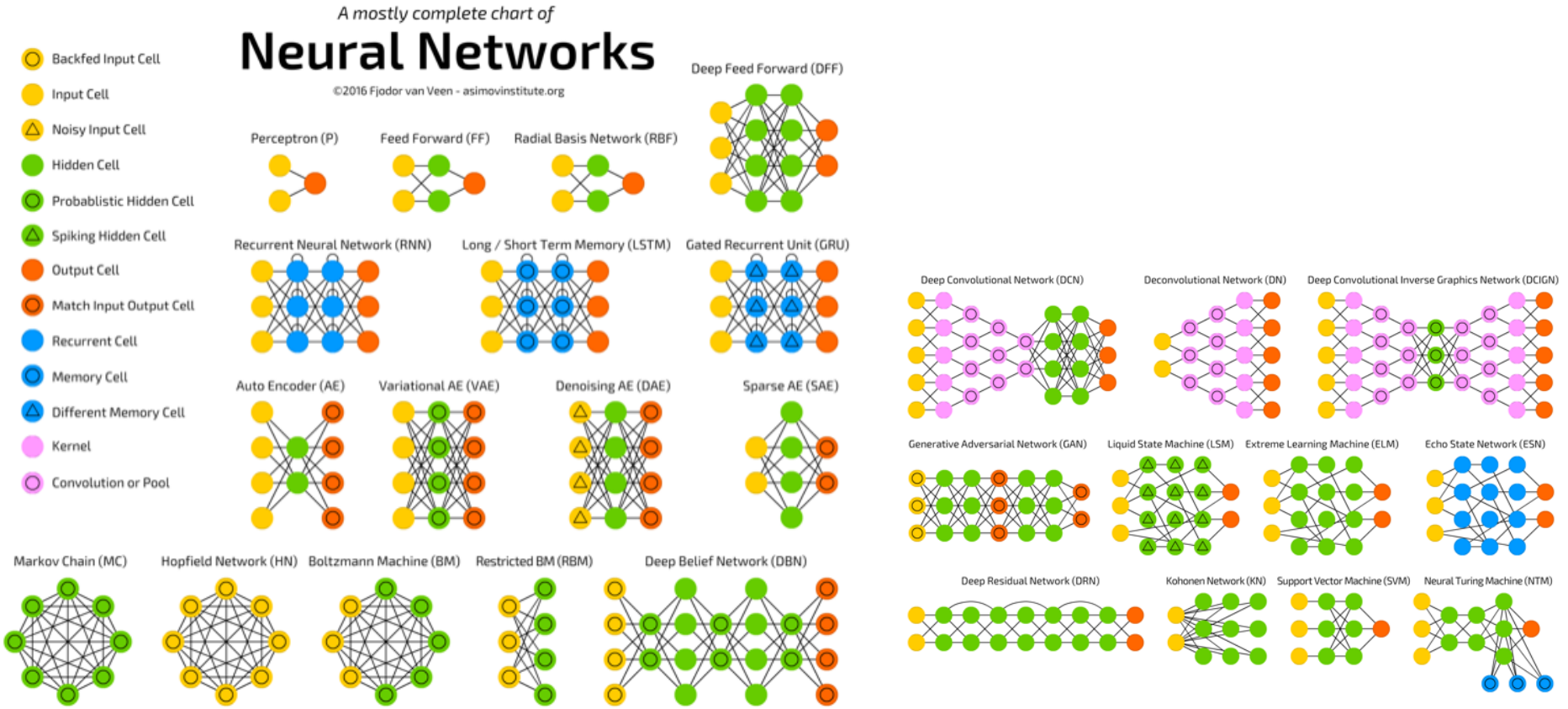
What - is deep learning?

Why is called “deep learning”?

Geofrey Hinton is a pioneer in the field of artificial neural networks and co-published the first paper on the backpropagation algorithm for training multilayer perceptron networks.

He may have started the introduction of the phrasing “*deep*” to describe the development of large artificial neural networks in his 2006 paper “A Fast Learning Algorithm for Deep Belief Nets”, which they describe an approach to training “deep” (as in a many layered network) of restricted Boltzmann machines.

What - is deep learning?



Where – Applications in CV

1. Image Classification
2. Object Detection & Localization
3. Image Segmentation
4. Image Captioning
5. Visual Question Answering
6. Pix2pix - Neural Style Transfer, Image Generating

Image Classification

Flower

Elephant

Port

Hallowen pumpkin

Dog

Image Classification

ballplayer, baseball player

scuba diver

groom, bridegroom

Image Classification

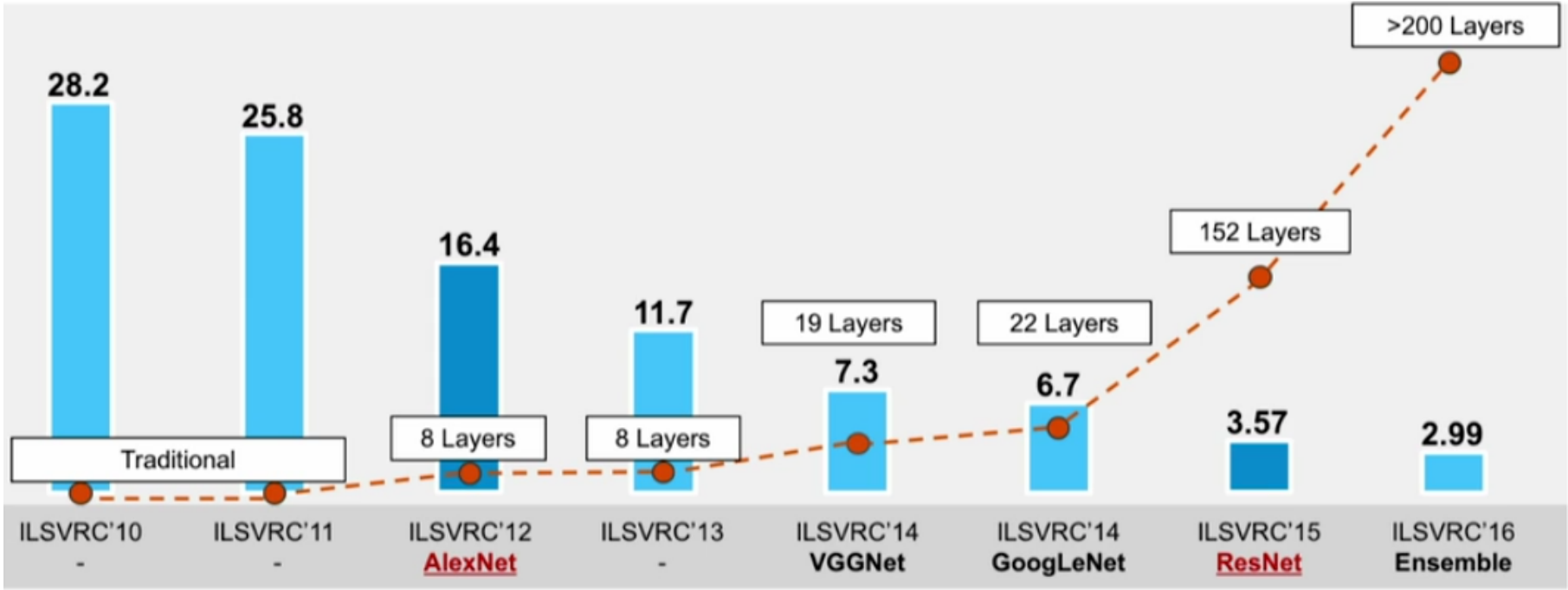


Image Classification

AlexNet

VGG

Google Inception

Microsoft ResNet

Facebook DenseNet

Google MobileNet

Image Detection & Localization

Faster RCNN

YOLO

SSD

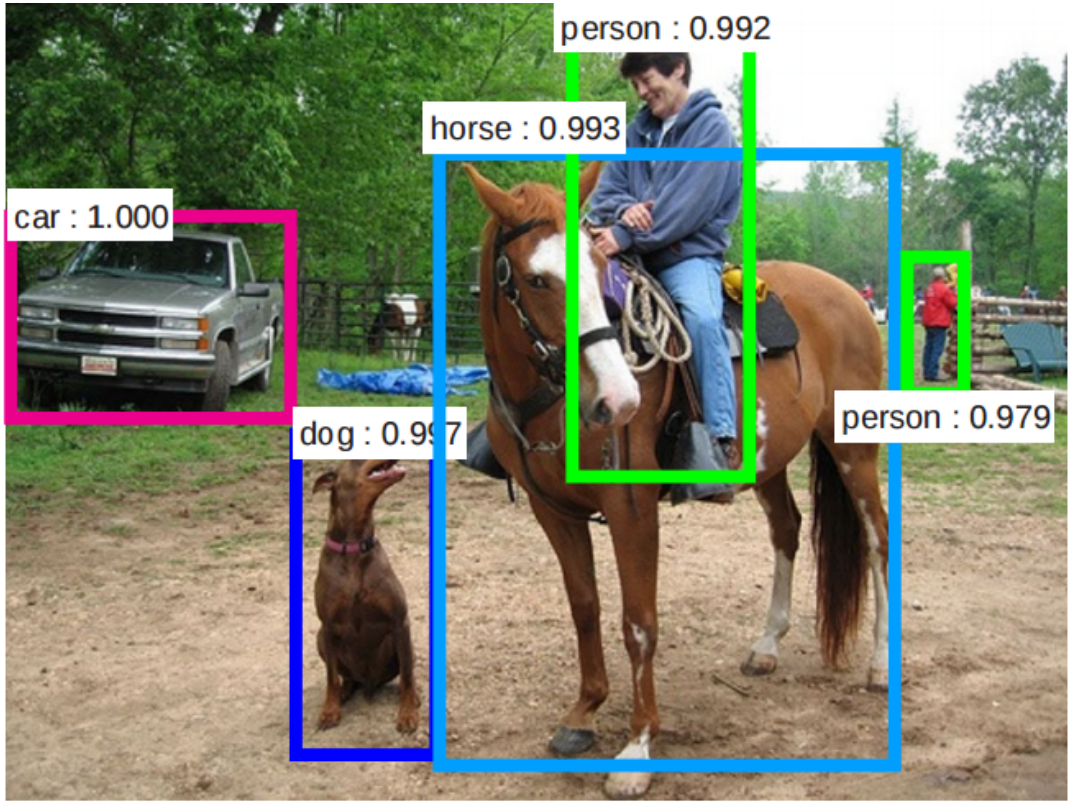


Image Detection & Localization

Mask RCNN

Semantic Segmentation

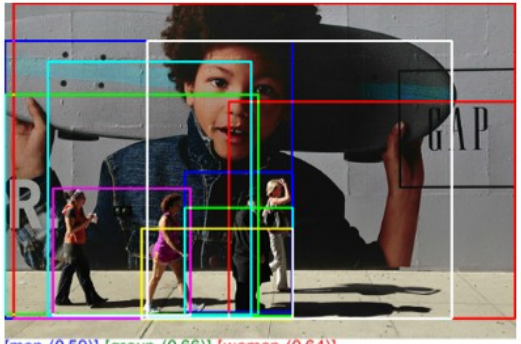
Semantic Segmentation

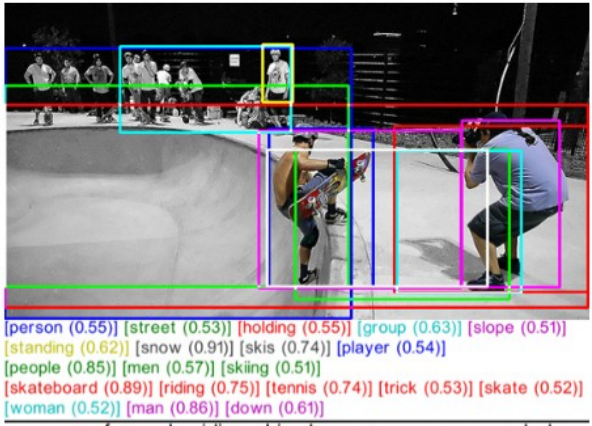
https://www.youtube.com/watch?v=L0V6zmGP_oQ

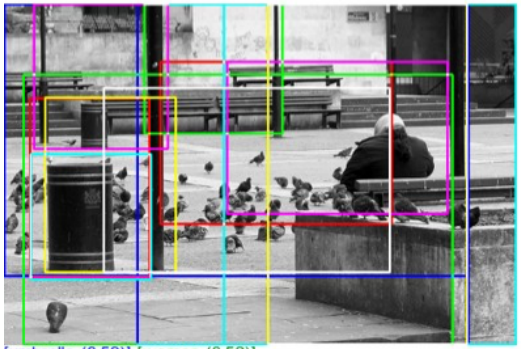
Image Captioning

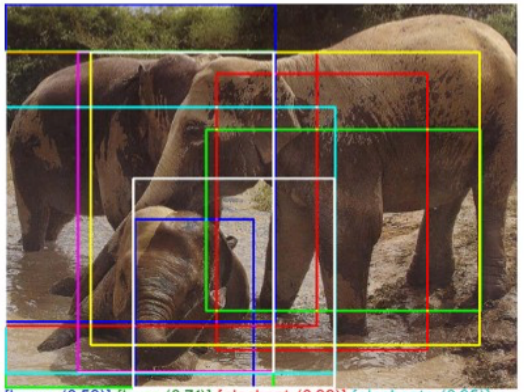
Describes without errors	Describes with minor errors	Somewhat related to the image
		
A person riding a motorcycle on a dirt road.	Two dogs play in the grass.	A skateboarder does a trick on a ramp.
		
A group of young people playing a game of frisbee.	Two hockey players are fighting over the puck.	A little girl in a pink hat is blowing bubbles.

Image Captioning









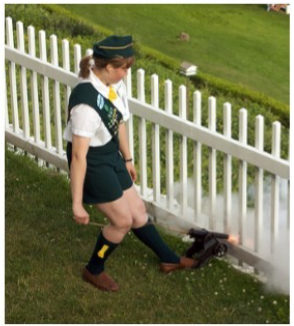
Visual Question Answering

What vegetable is on the plate?
Neural Net: broccoli
Ground Truth: broccoli

What color are the shoes on the person's feet ?
Neural Net: brown
Ground Truth: brown

How many school busses are there?
Neural Net: 2
Ground Truth: 2

What is on top of the refrigerator?
Neural Net: magnets
Ground Truth: cereal



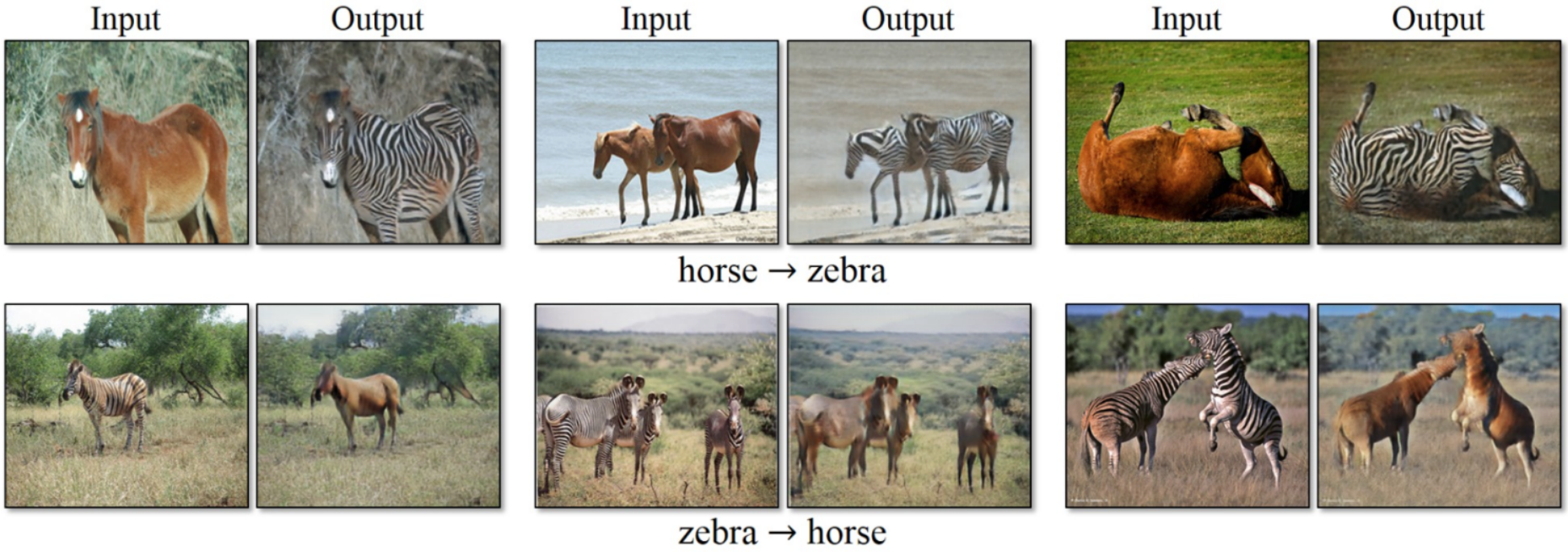
What uniform is she wearing?
Neural Net: shorts
Ground Truth: girl scout

What is the table number?
Neural Net: 4
Ground Truth:40

Neural Style Transfer

Neural Style Transfer

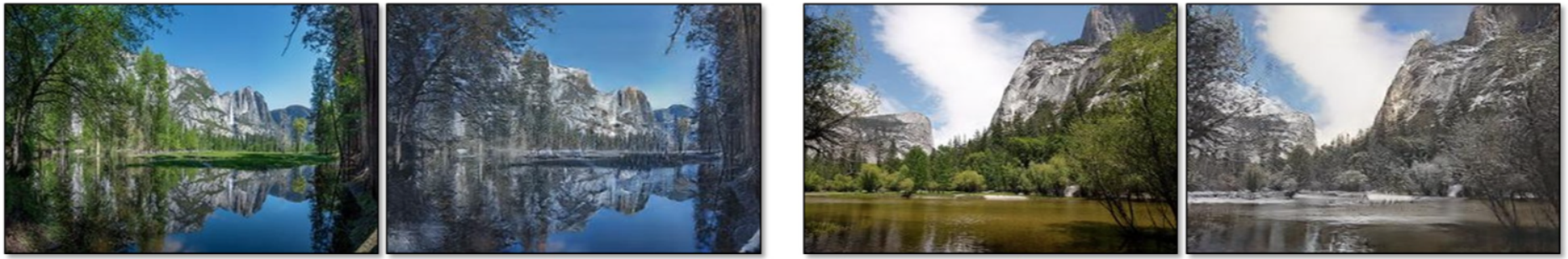
Object Transfiguration



Neural Style Transfer

Season Transfer

winter Yosemite → summer Yosemite



summer Yosemite → winter Yosemite

Neural Style Transfer

https://www.youtube.com/watch?v=9reHvktowLY

Neural Style Transfer

https://www.youtube.com/watch?v=MVBe6_o4cMI

The common paradigm (Andrej Ka)

III. Deep learning Models

III.1 Fully connected Neural Network

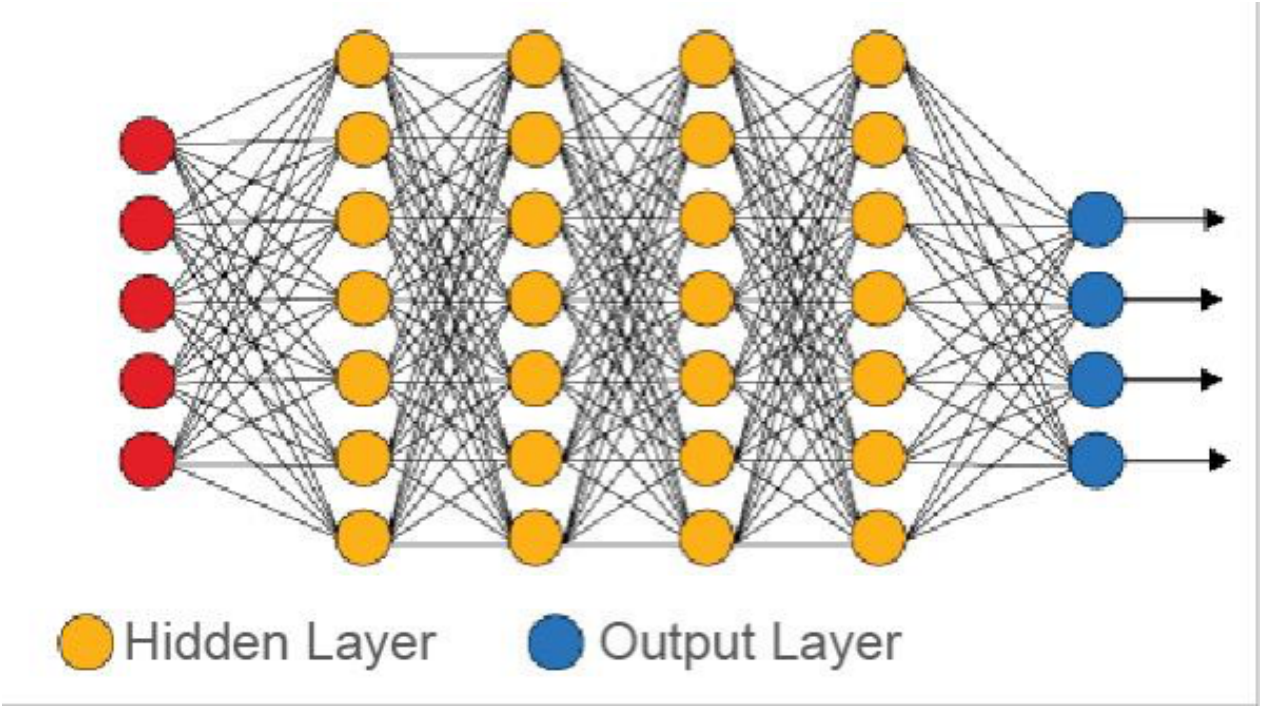
III.2 Convolutional Neural Network

III.3 Recurrent Neural Networks

III.4 AutoEncoder

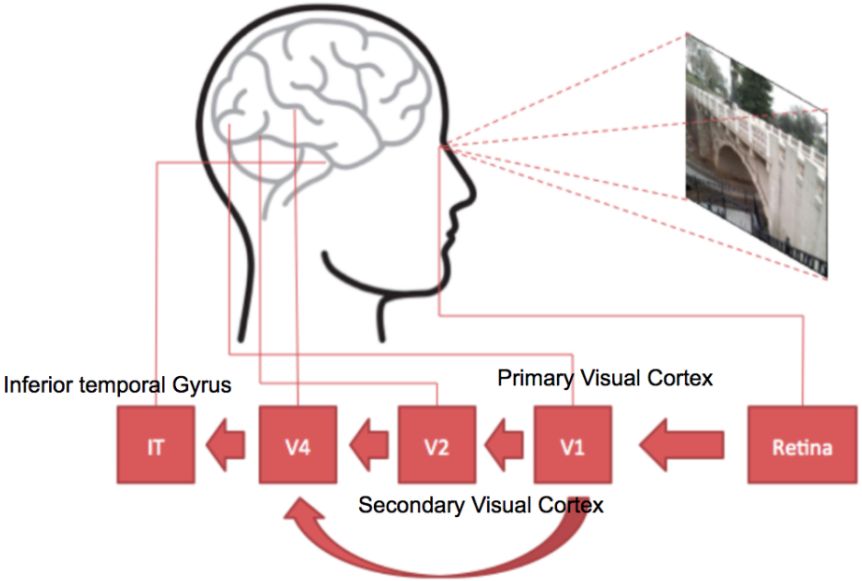
III.5 Generative Adversarial Network

III.1 Fully connected Neural Network



III.2 Convolutional Neural Network - CNN

A specific type of Neural Network for image processing by mimicking human's vision (visual cortex) system



V1: Edge detection, etc.

V2: Extract simple visual properties (orientation, spatial frequency, color, etc)

V4: Detect object features of intermediate complexity

TI: Object recognition.

III.2 Convolutional Neural Network - CNN

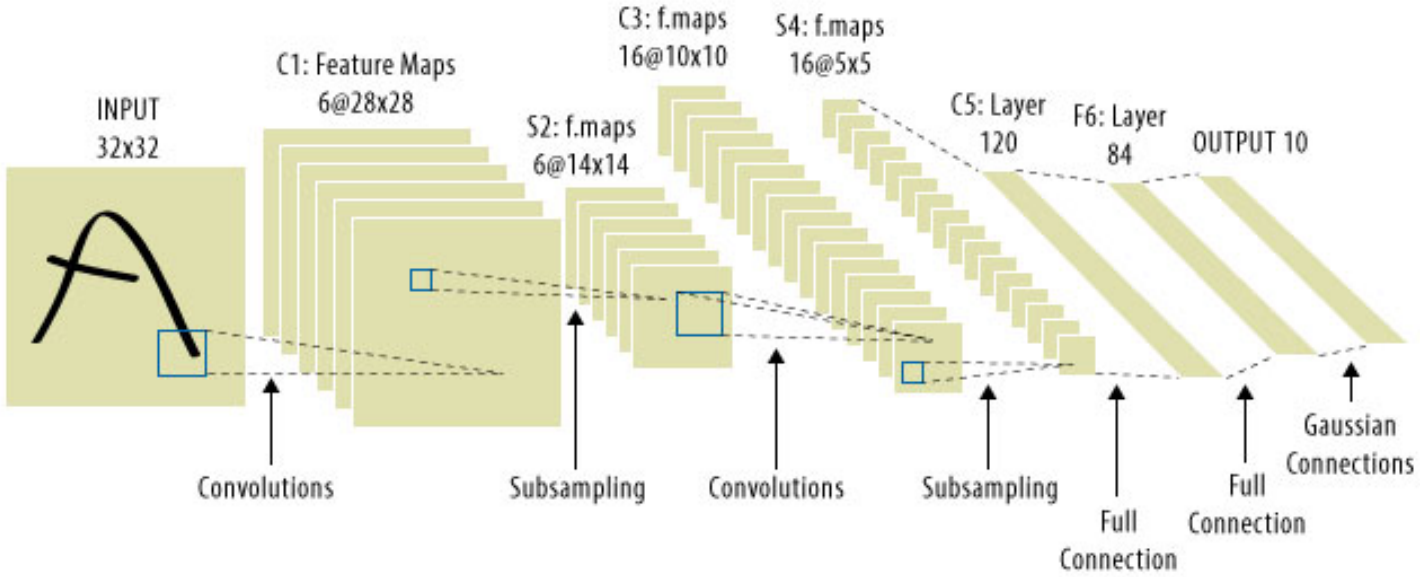
The key concept here is Convolution, which is a mathematical term, here referring to an operation between two matrices.

The convolutional layer has a fixed small matrix defined, also called kernel or filter. As the kernel is sliding, or convolving, across the matrix representation of the input image, it is computing the element-wise multiplication of the values in the kernel matrix and the original image values.

Specially designed kernels can process images for common purposes like blurring, sharpening, edge detection and many others, fast and efficiently.

III.2 Convolutional Neural Network - CNN

Convolutional and pooling/sub-sampling layers act like the V1, V2 and V4 visual cortex units, responding to feature extraction. The object recognition reasoning happens in the later fully-connected layers which consume the extracted features.



III.3 Recurrent Neural Network - RNN

A sequence model is usually designed to transform an input sequence into an output sequence that lives in a different domain. Recurrent neural network, short for “RNN”, is suitable for this purpose and has shown tremendous improvement in problems like handwriting recognition, speech recognition, and machine translation

RNN has the capability to process long sequential data and to tackle tasks with context spreading in time. The model processes one element in the sequence at one time step. After computation, the newly updated unit state is passed down to the next time step to facilitate the computation of the next element.

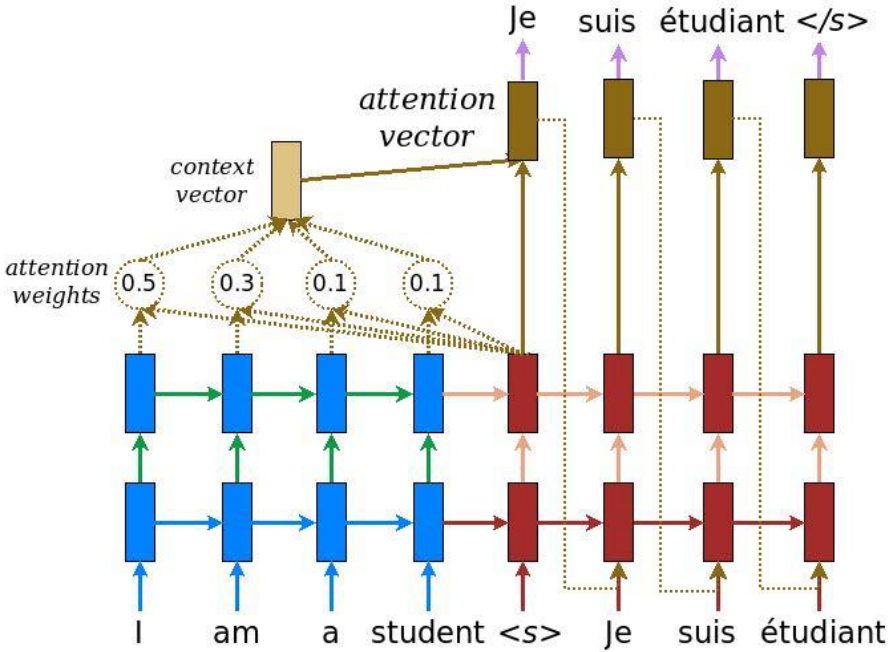
III.3 Recurrent Neural Network - RNN

Google’s speech recognition (for English) is now almost as accurate as humans

https://www.youtube.com/watch?v=D5VN56jQMWM

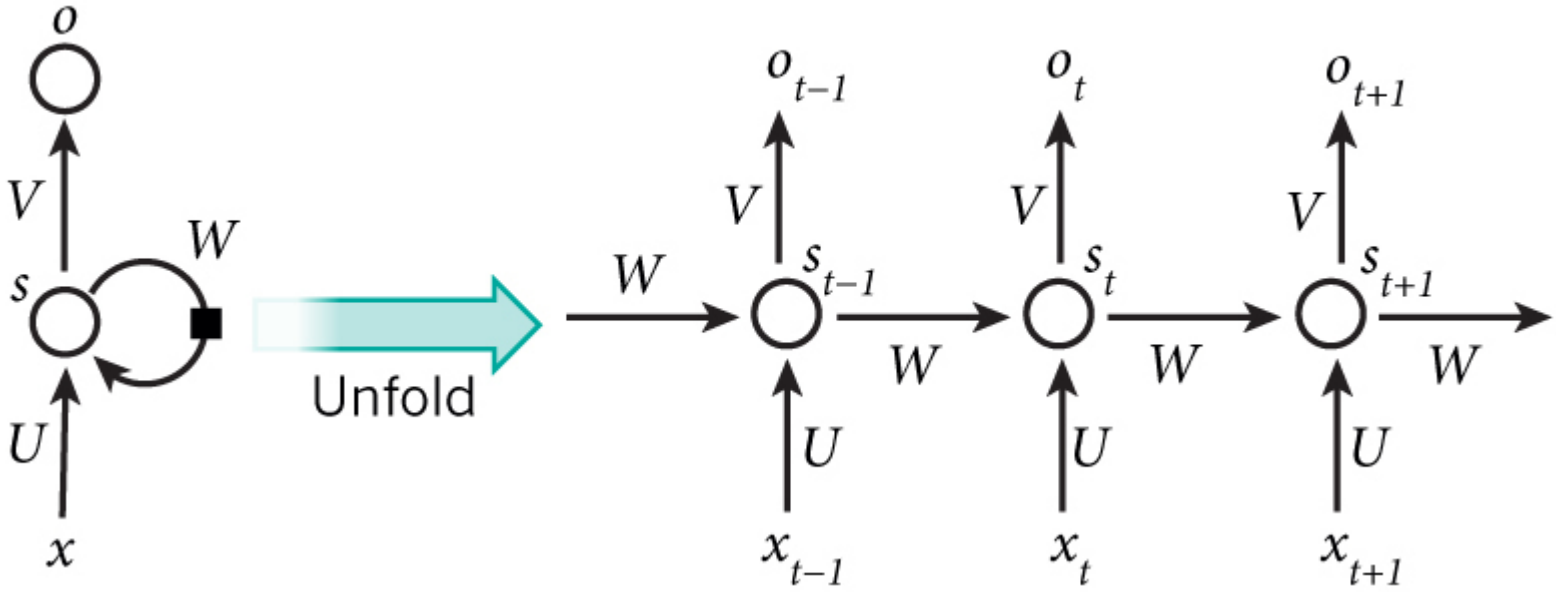
III.3 Recurrent Neural Network - RNN

Google's Neural Machine Translation



III.3 Recurrent Neural Network - RNN

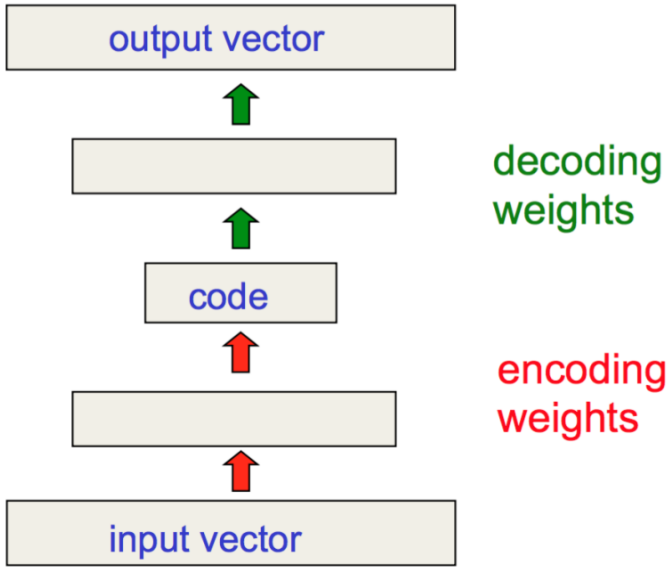
Imagine the case when an RNN model reads all the Wikipedia articles, character by character, and then it can predict the following words given the context



III.4 Autoencoder

Different from the previous models, autoencoders are for unsupervised learning. It is designed to learn a **low-dimensional** representation of a **high-dimensional** data set, similar to what Principal Components Analysis (PCA) does

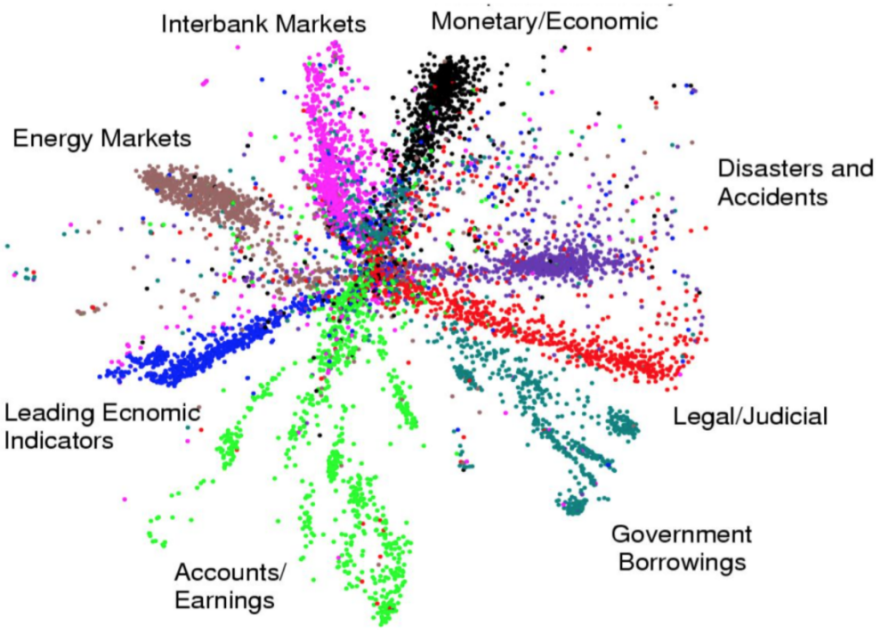
The autoencoder model tries to learn an approximation function $f(x) \approx x$ to reproduce the input data. However, it is restricted by a bottleneck layer in the middle with a very small number of nodes. With limited capacity, the model is forced to form a very efficient encoding of the data, that is essentially the low-dimensional code we learned



III.4 Autoencoder

Image compression

Compress documents on a variety of topics



III.5 Generative Adversarial Network

Generative adversarial network, short for “GAN”, is a type of deep generative models. GAN is able to create new examples after learning through the real data.

The famous deep learning researcher Yann LeCun gave it a super high praise: Generative Adversarial Network is the most interesting idea in the last ten years in machine learning.

III.5 Generative Adversarial Network

GAN comprises two independent models: the **Generator** and the **Discriminator**.

- The generator produces fake images and sends the output to the discriminator model.
- The discriminator works like a judge, as it is optimized for identifying the real photos from the fake ones.
- The generator model is trying hard to cheat the discriminator while the judge is trying hard not to be cheated. This interesting zero-sum game between these two models motivates both to develop their designed skills and improve their functionalities. Eventually, we take the generator model for producing new images.

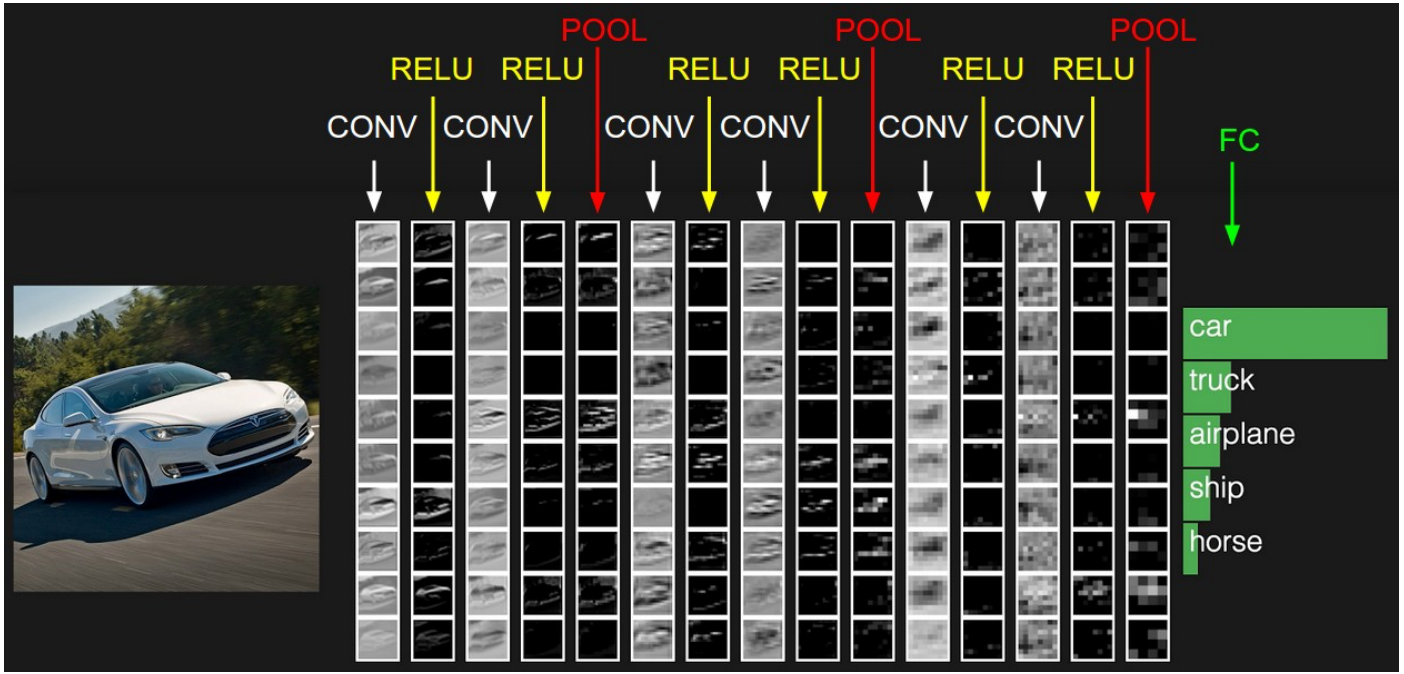
III.5 Generative Adversarial Network

Applications:

- Generate synthetic data
- Neural Style Transfer

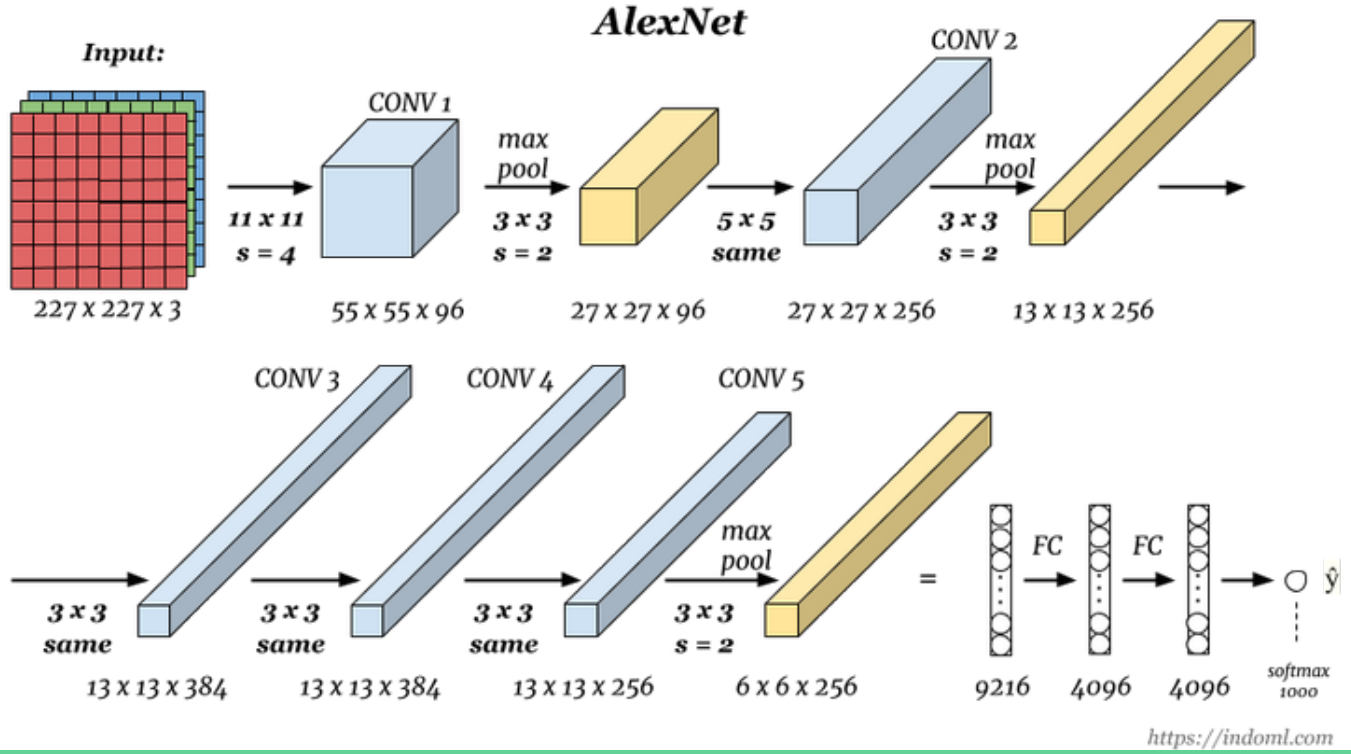
IV. Fundamental blocks

A typical architecture of a CNN



IV. Fundamental blocks

A typical architecture of a CNN



IV. Fundamental blocks

CONV

POOL

ReLU

FC

CONV - Convolution

CONVolve an image with a kernel/filter: I * W = O

Simply speaking, consider CONV as a pattern searching

2	2	1	1	0
2	1	2	0	2
2	0	1	1	0
2	0	1	0	2
0	0	1	2	1

0	-1	-1
1	-1	-1
1	1	1

CONV - Convolution

0	-1	-1
1	-1	-1
1	1	1

2	2	1	1	0
2	1	2	0	2
2	0	1	1	0
2	0	1	0	2
0	0	1	2	1

1		

CONV - Convolution

2	2	1	1	0
2	1	2	0	2
2	0	1	1	0
2	0	1	0	2
0	0	1	2	1

0	-1	-1
1	-1	-1
1	1	1

1	3	

CONV - Convolution

0	-1	-1
1	-1	-1
1	1	1

2	2	1	1	0
2	1	2	0	2
2	0	1	1	0
2	0	1	0	2
0	0	1	2	1

1	3	-2

CONV - Convolution

0	-1	-1
1	-1	-1
1	1	1

2	2	1	1	0
2	1	2	0	2
2	0	1	1	0
2	0	1	0	2
0	0	1	2	1

1	3	-2
1		

CONV - Convolution

2	2	1	1	0
2	1	2	0	2
2	0	1	1	0
2	0	1	0	2
0	0	1	2	1

0	-1	-1
1	-1	-1
1	1	1

1	3	-2
1	-4	-2
-1	-3	-4

Important:

1. A kernel/filter serves as a pattern searcher.

2. Each CNN has million of kernels/filters. Each serves as a weak classifier. When combined, they provide a very powerful classifier.

Convolution intuition

- Biologically inspired
- Local connectivity

- Three main hyper-parameters:
  - Padding
  - Stride
  - Depth

Padding

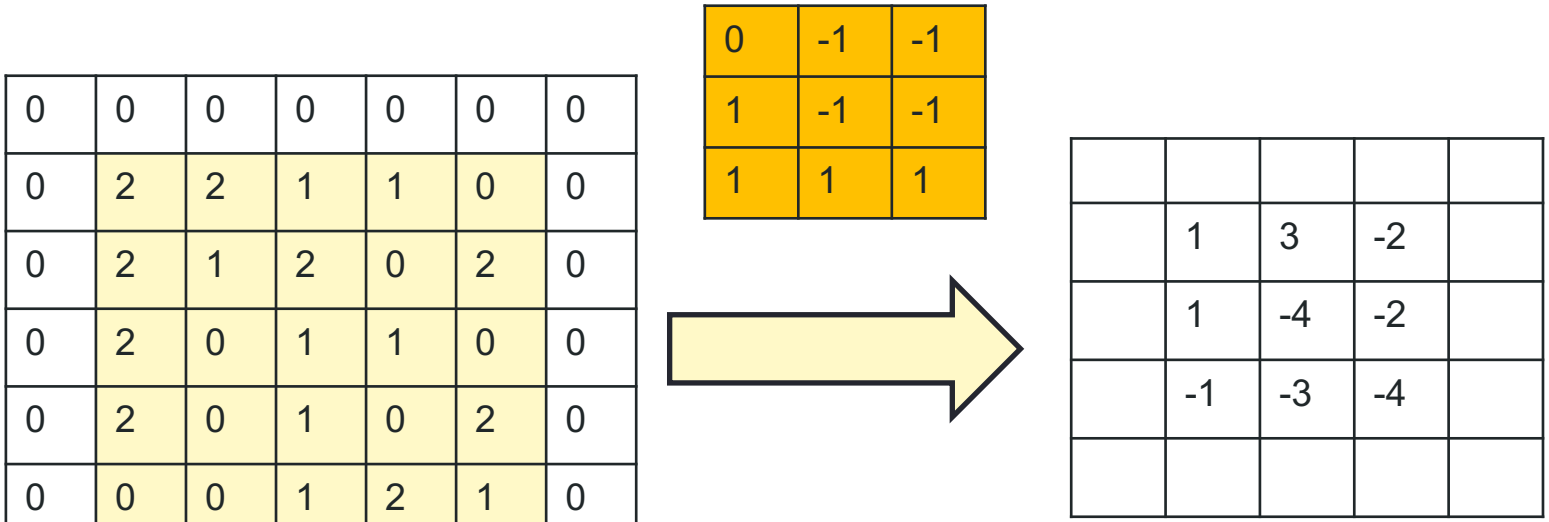
2	2	1	1	0
2	1	2	0	2
2	0	1	1	0
2	0	1	0	2
0	0	1	2	1

0	-1	-1
1	-1	-1
1	1	1

1	3	-2
1	-4	-2
-1	-3	-4

Padding

Zero-padding



Padding

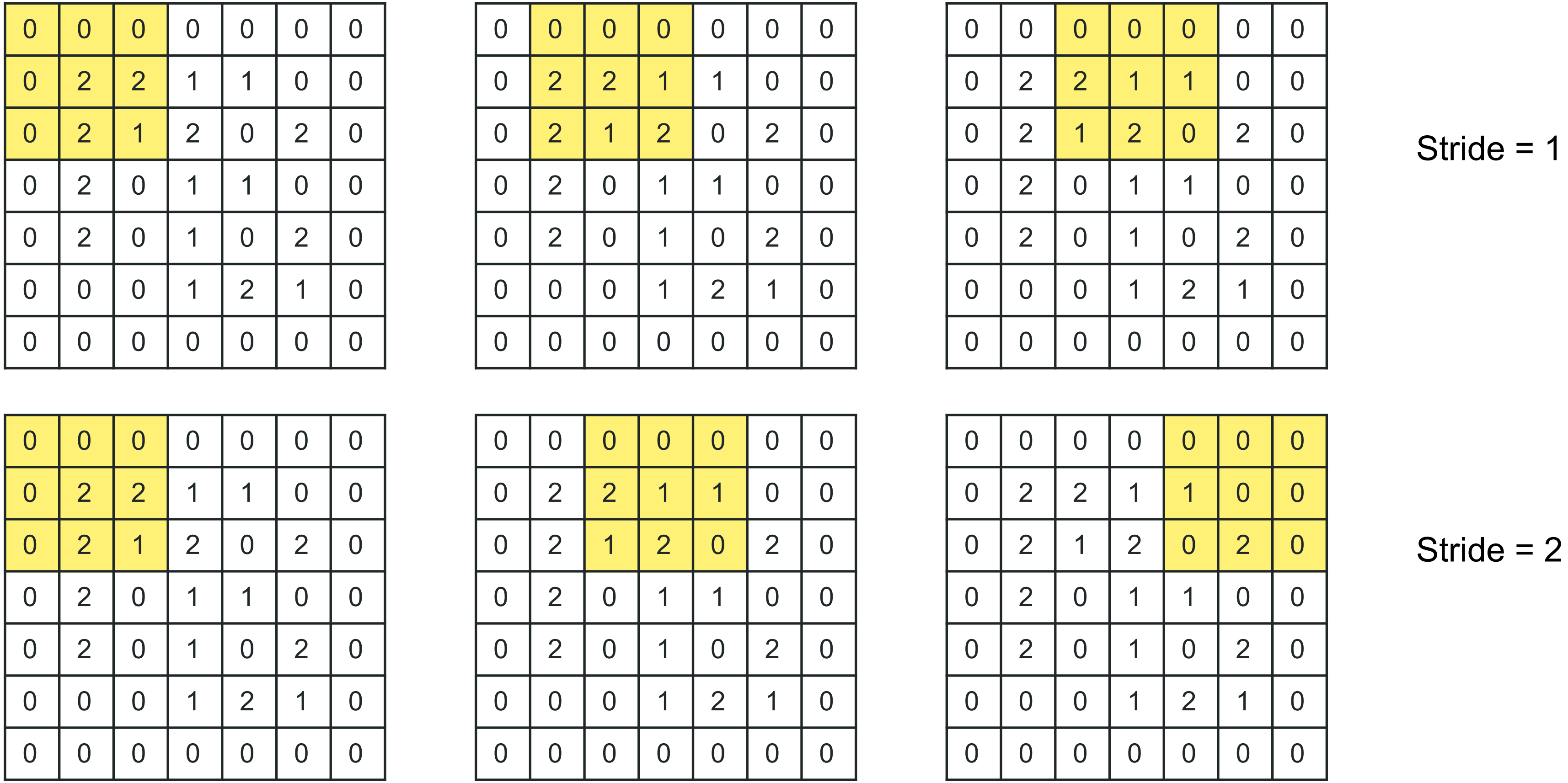
Same-padding

2	2	2	1	1	0	0
2	2	2	1	1	0	0
2	2	1	2	0	2	2
2	2	0	1	1	0	0
2	2	0	1	0	2	2
0	0	0	1	2	1	1
0	0	0	1	2	1	1

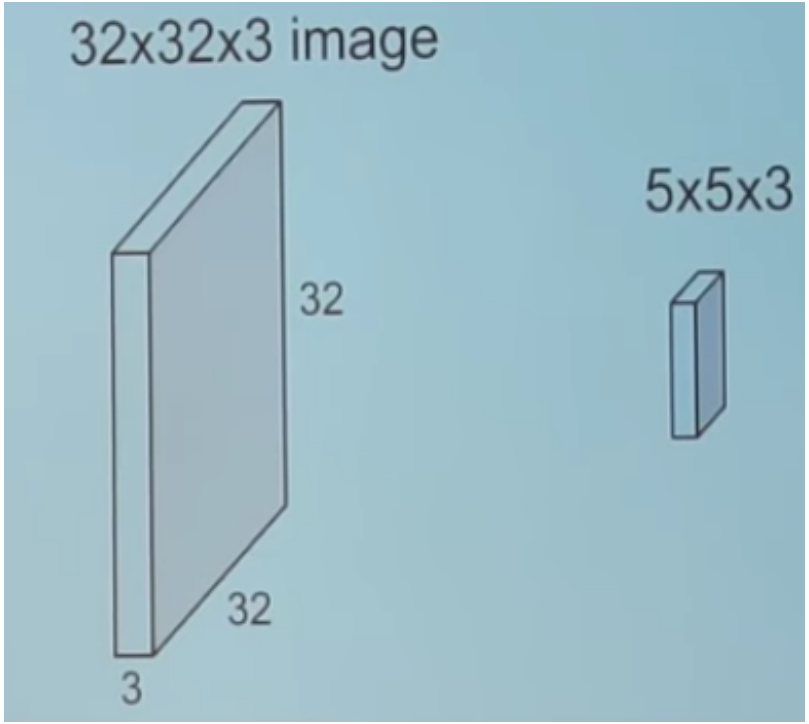
0	-1	-1
1	-1	-1
1	1	1

	1	3	-2	
	1	-4	-2	
	-1	-3	-4	

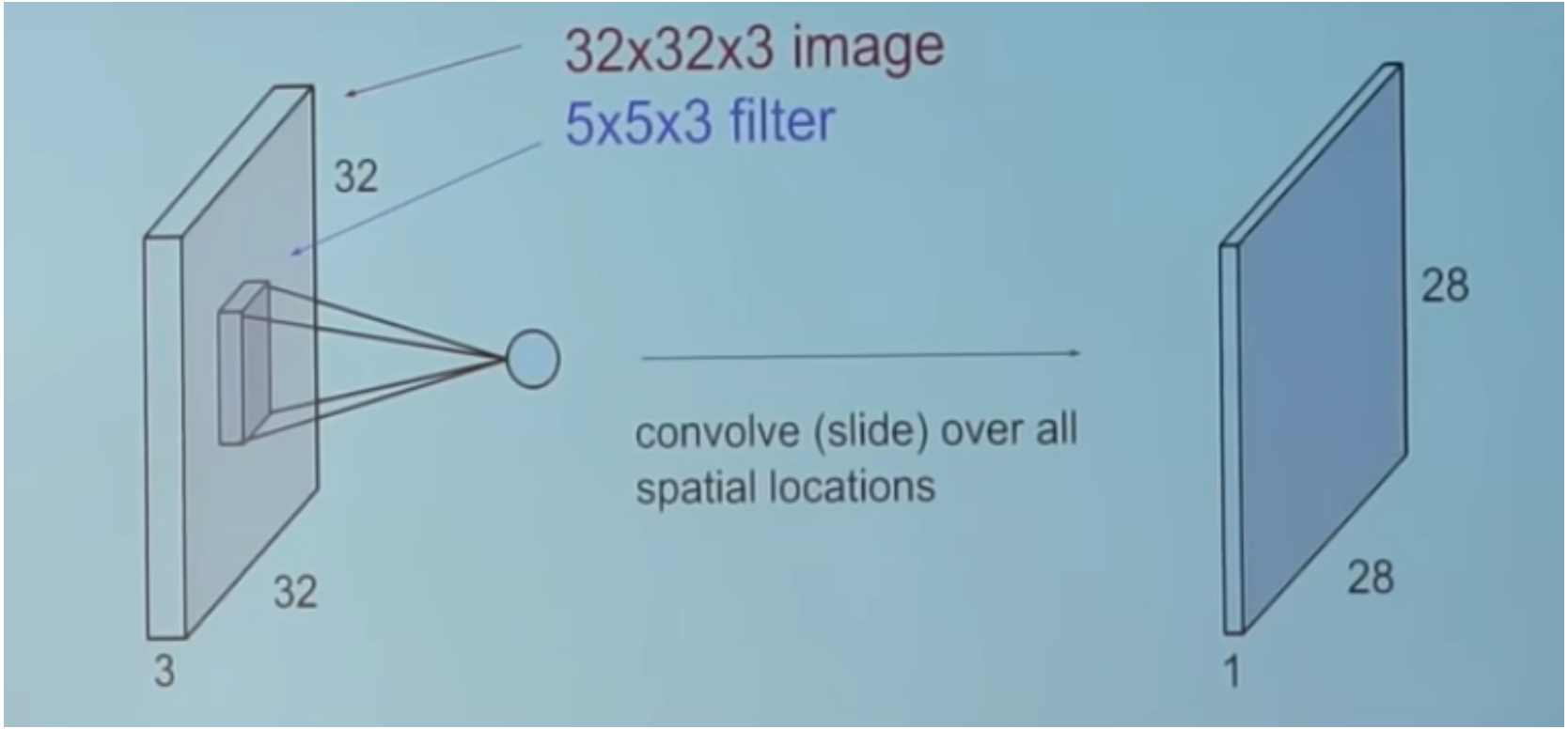
Stride



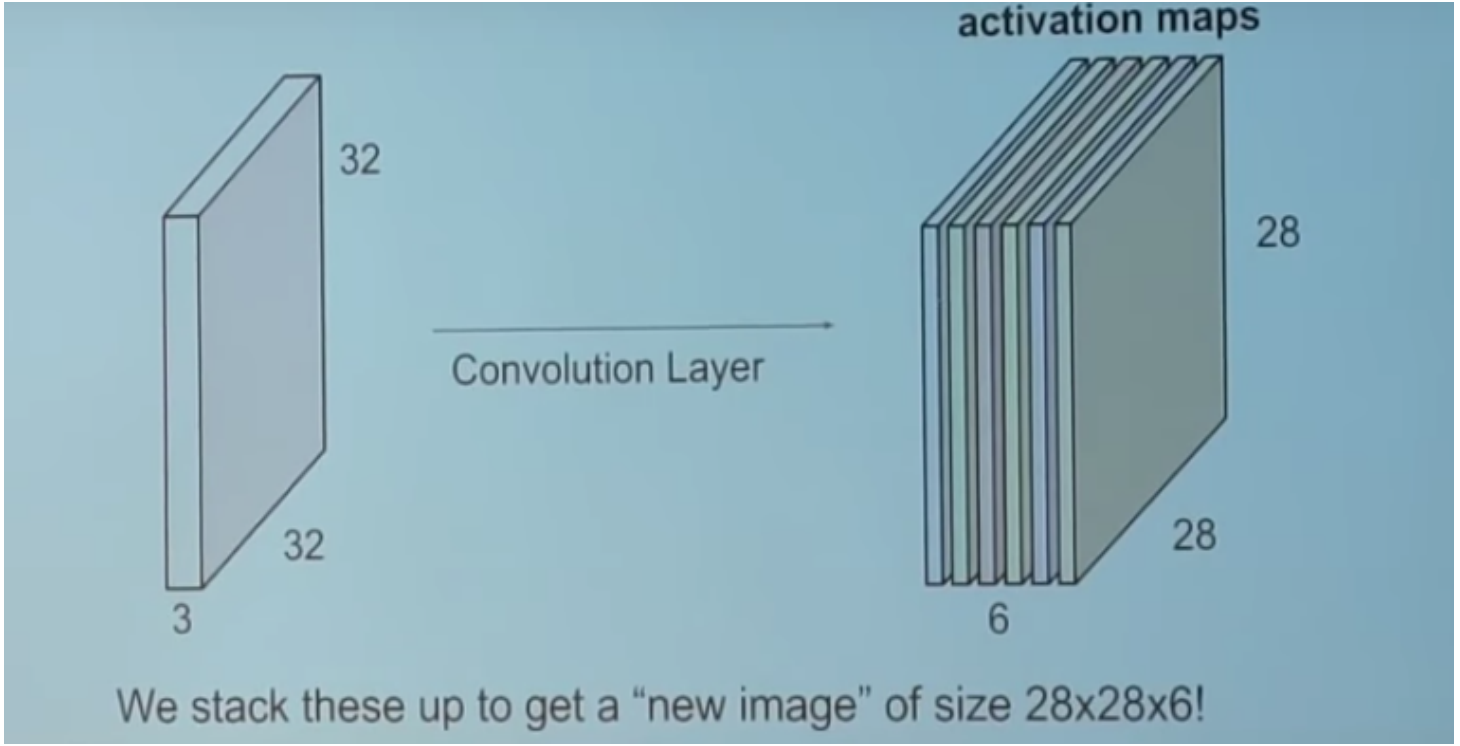
Depth - Tensor convolution



Depth- Tensor convolution



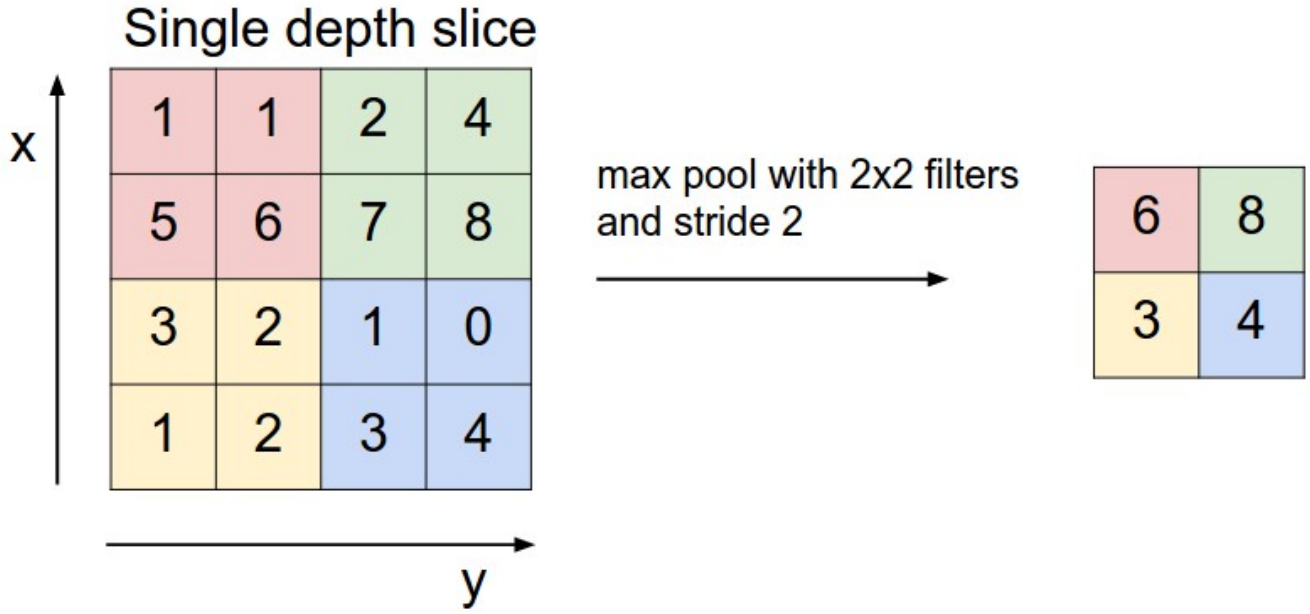
Depth - Tensor convolution



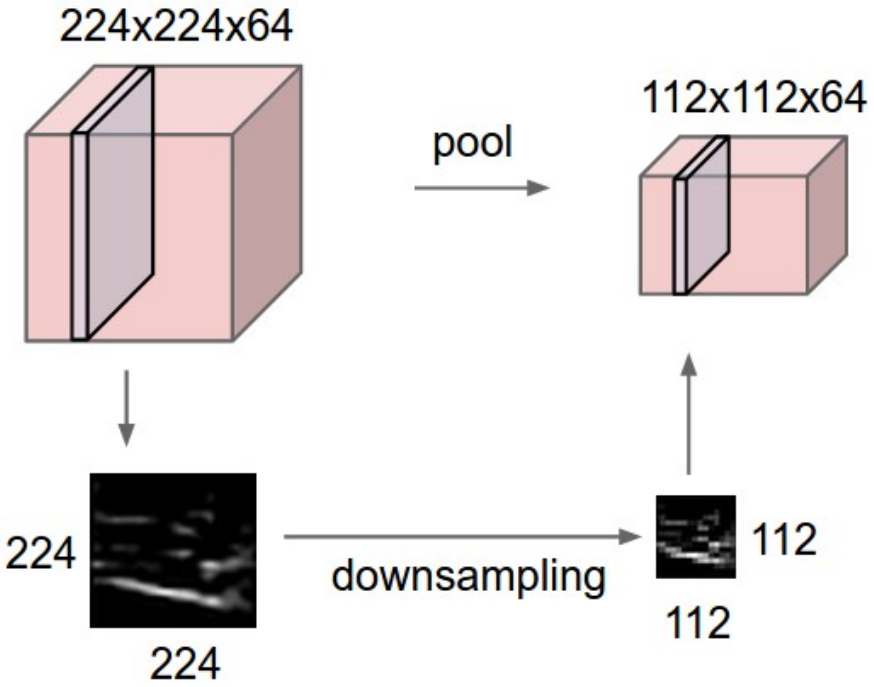
All together CONV layer

- Accepts a volume of size $W1 \times H1 \times D1$
- Requires four hyper-parameters:
  - Number of filters K, their spatial extent F, the stride S, the amount of zero padding P.
- Produces a volume of size $W2 \times H2 \times D2$ where:
  - $W2=(W1-F+2P)/S+1$
  - $H2=(H1-F+2P)/S+1$ (i.e. width and height are computed equally by symmetry)
  - $D2=K$

POOL - pooling



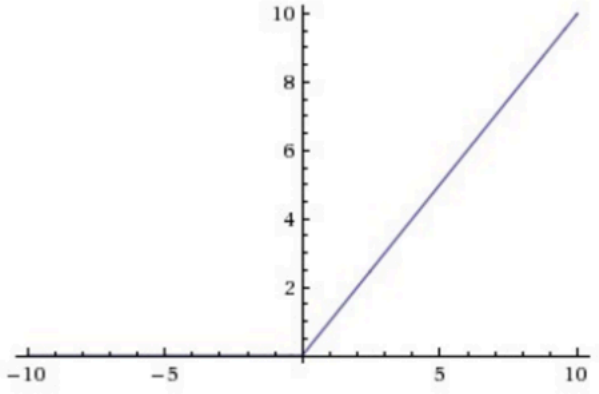
POOL - pooling



ReLU - Rectified Linear Unit

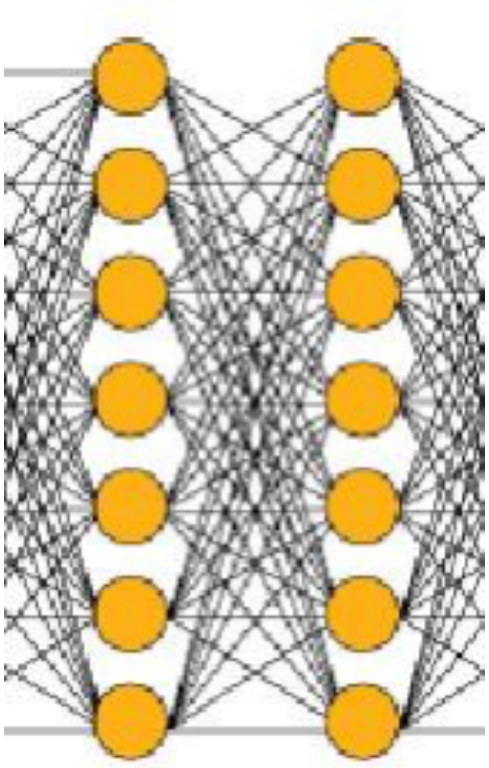
- ReLU will apply an elementwise activation function, such as the max(0,x) thresholding at zero. This leaves the size of the volume unchanged

Output = Max(zero, Input)



- The purpose of ReLU is to introduce non-linearity in our ConvNet, since most of the real-world data we would want our ConvNet to learn would be non-linear

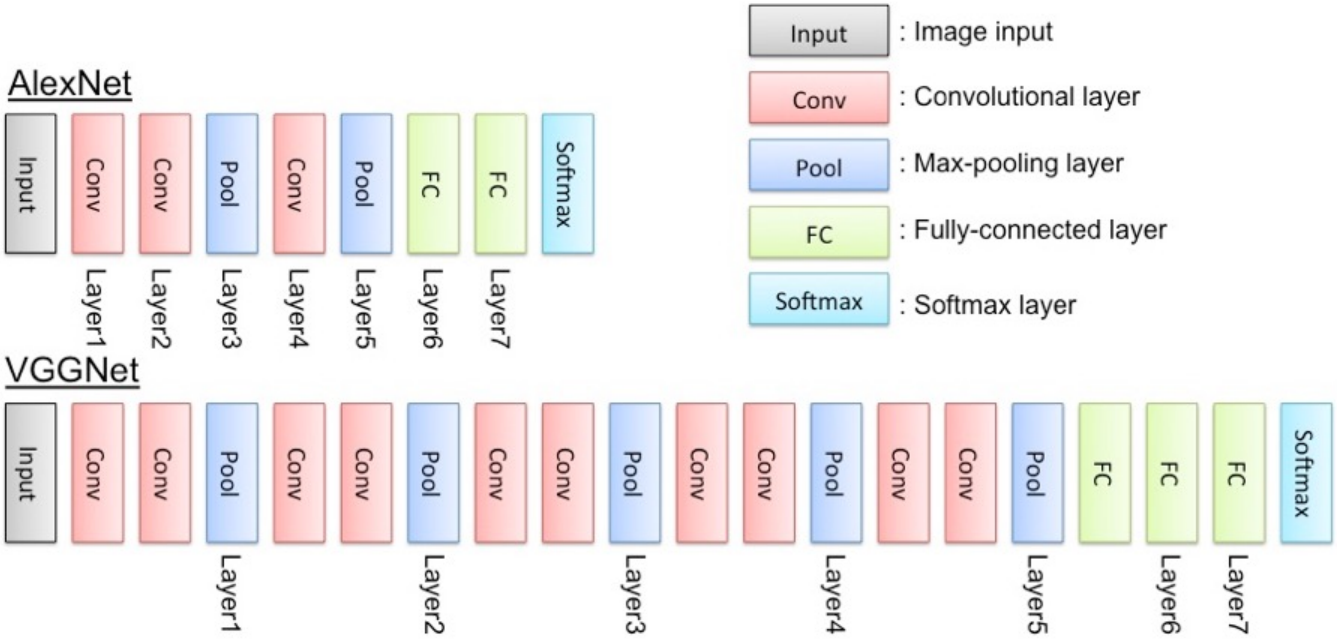
FC – Fully Connected



CNN Architectures

- Layer Patterns
- INPUT -> [[CONV -> RELU]*N -> POOL?]*M -> [FC -> RELU]*K -> FC

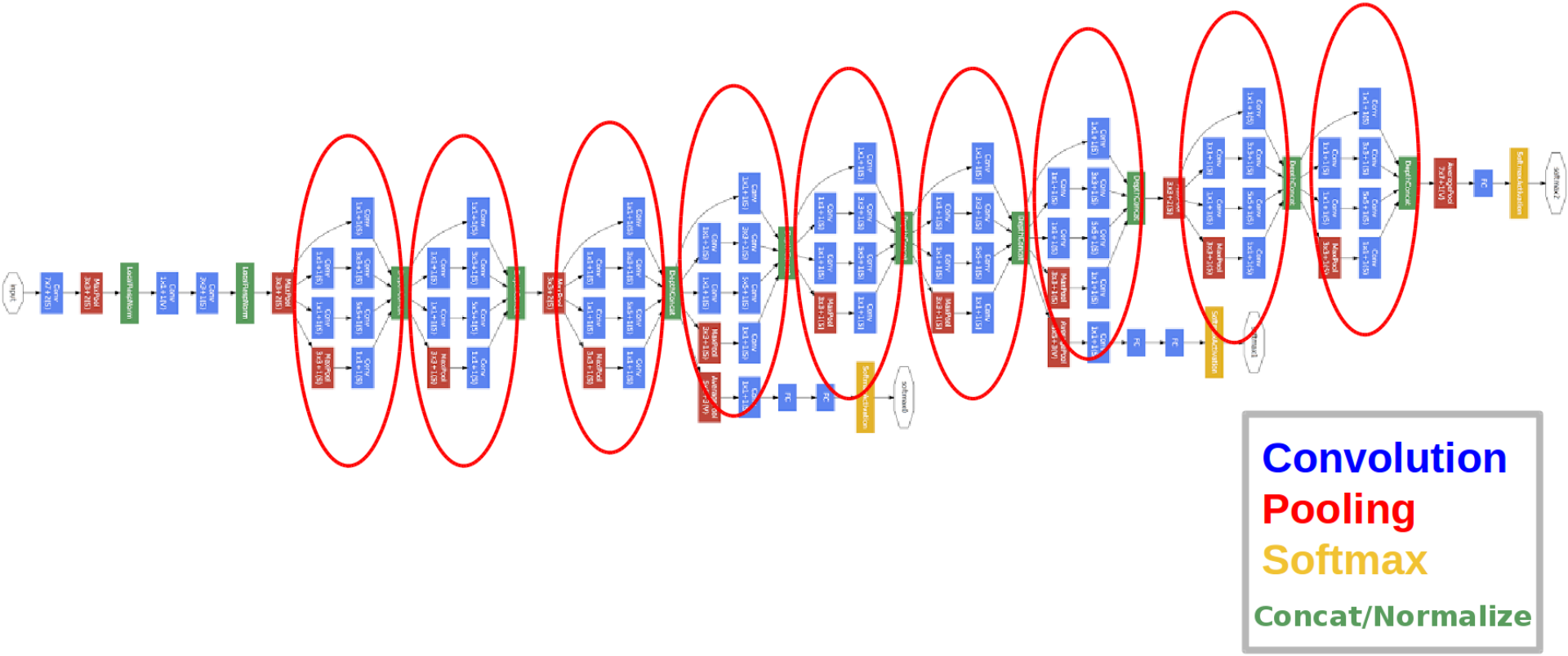
_ VGGNet의 ConvNet configuration



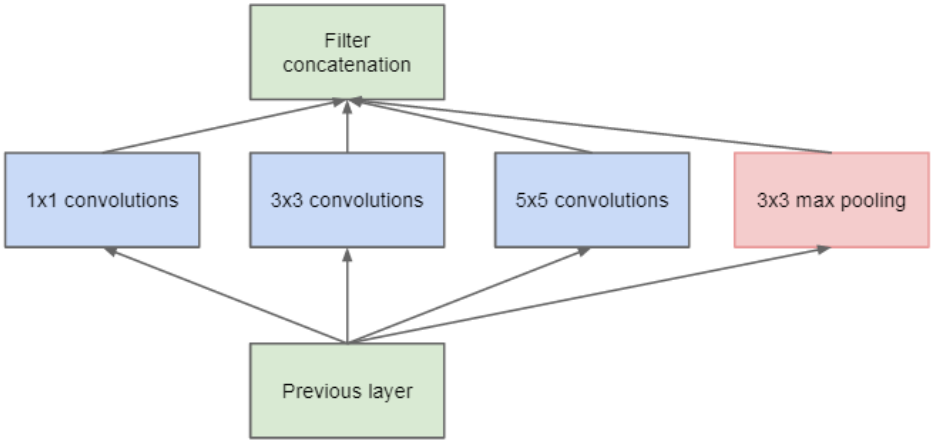
VGG


3x3 conv, 64
3x3 conv, 64
pool/2
3x3 conv, 128
3x3 conv, 128
pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
fc 4096
fc 4096
fc 4096
Size:224
Size:112
Size:56
Size:28
Size:14
Size:7

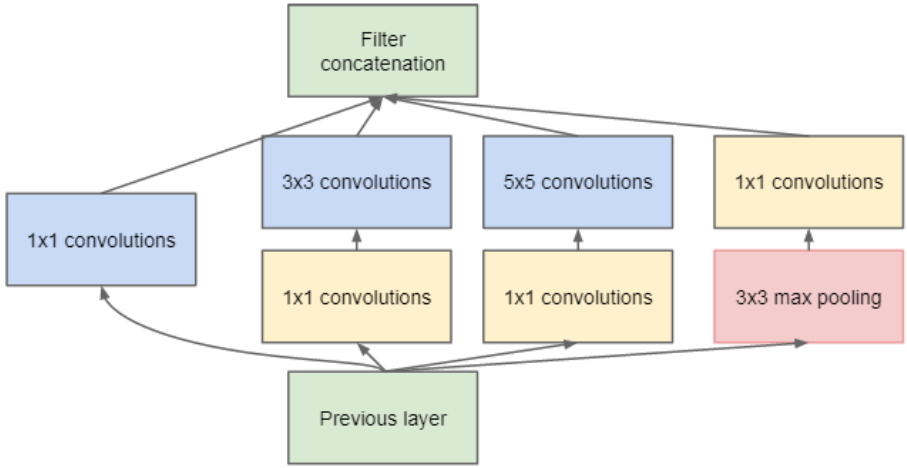
Google's Inception



Inception module

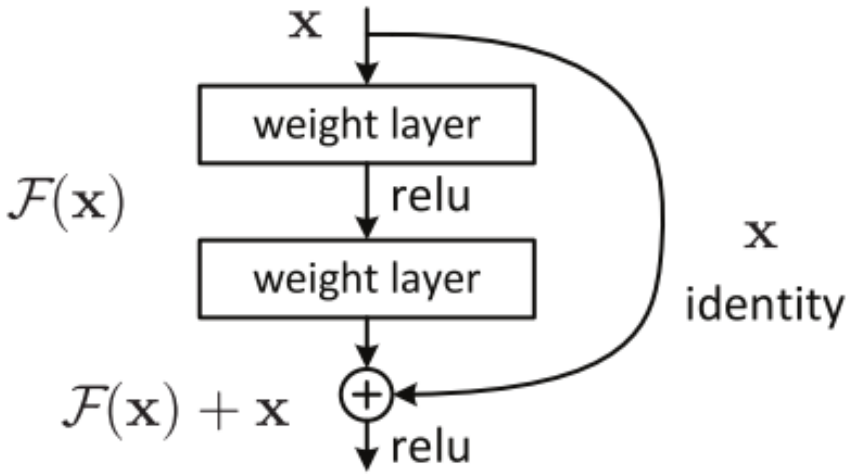


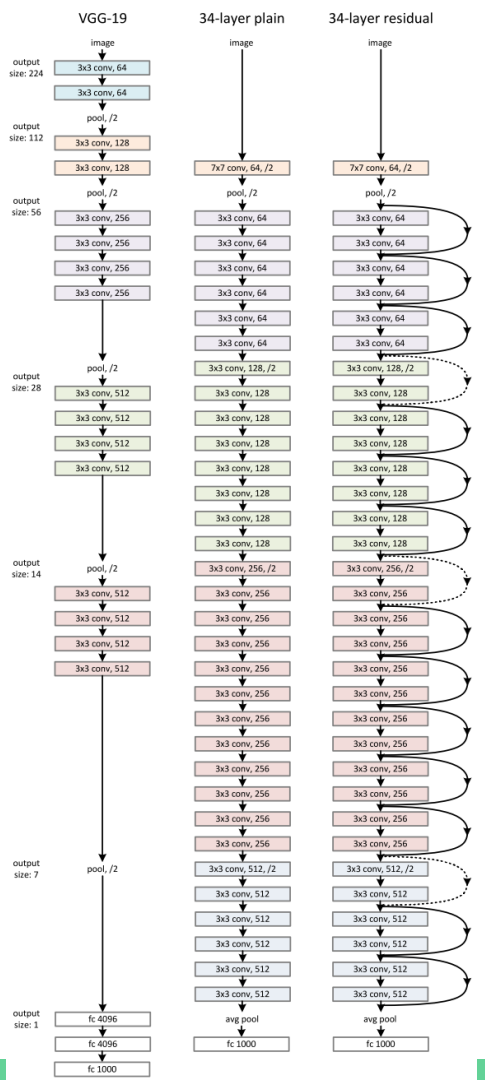
(a) Inception module, naïve version



(b) Inception module with dimension reductions

Microsoft's ResNet





ResNet variants

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Facebook's DenseNet

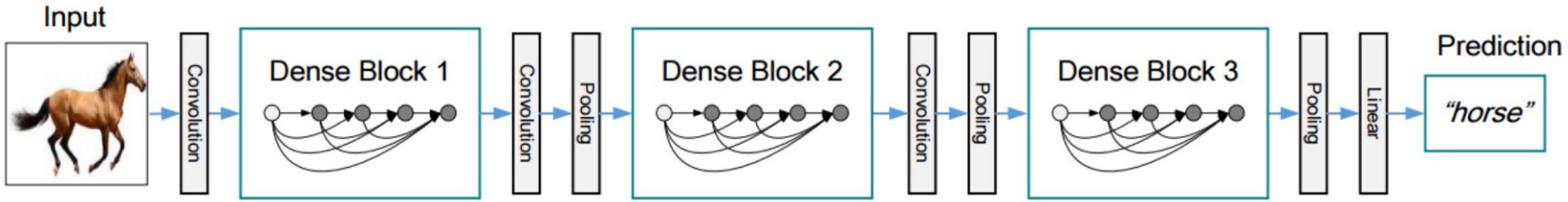
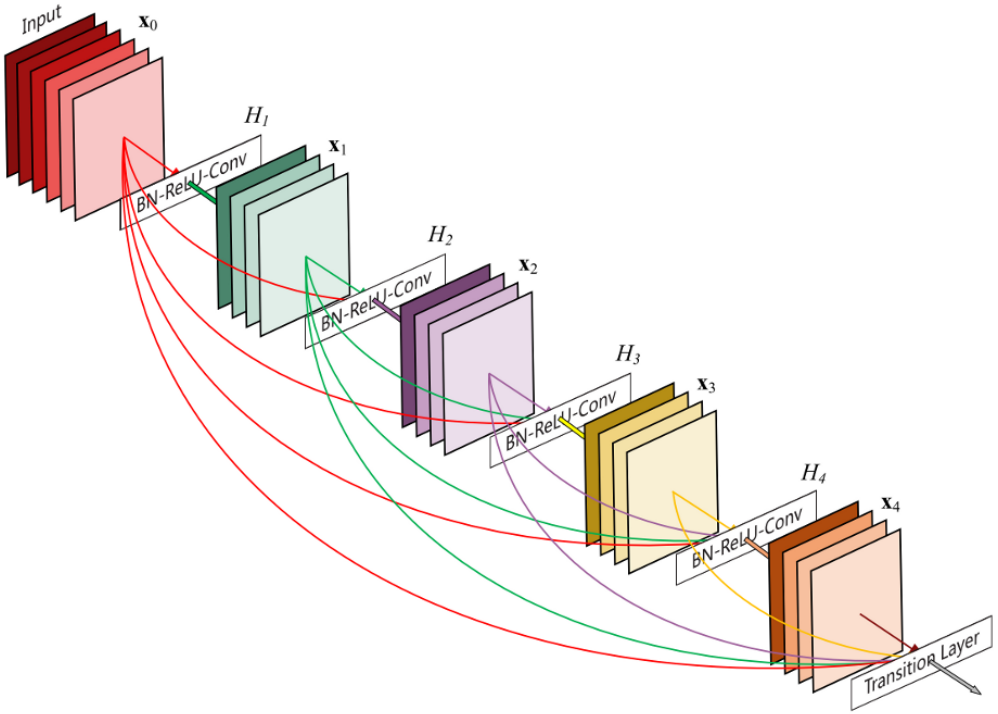


Figure 2. A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature map sizes via convolution and pooling.

Dense block


Input
$\mathbf{x}_0$
H_1
BN-ReLU-Conv
$\mathbf{x}_1$
H_2
BN-ReLU-Conv
$\mathbf{x}_2$
H_3
BN-ReLU-Conv
$\mathbf{x}_3$
H_4
BN-ReLU-Conv
$\mathbf{x}_4$
Transition Layer

DenseNet variants

Layers	Output Size	DenseNet-121($k = 32$)	DenseNet-169($k = 32$)	DenseNet-201($k = 32$)	DenseNet-161($k = 48$)
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 36
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 24
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Table 1. DenseNet architectures for ImageNet. The growth rate for the first 3 networks is $k = 32$, and $k = 48$ for DenseNet-161. Note that each "conv" layer shown in the table corresponds the sequence BN-ReLU-Conv.

Demo 1

- Architecture of state-of-the-art CNNs
  - AlexNet
  - VGG
  - Inception
  - ResNet
  - DenseNet
  - MobileNet

Demo 2

- Building your own CNN & Training from scratch
- Using pre-trained models
- Transfer Learning