Convolutional Neural Networks

Convolutional Neural Networks

- **Definition:**

  A Convolutional Neural Network (CNN) is a class of deep neural networks primarily used for analyzing visual data, such as images and videos.

- **Core Idea:** CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images through the application of convolutional filters.

Applications

Widely used in

1. Image classification,

2. Object detection,

3.Facial recognition, medical image analysis, and more.

How CNN Works

- **Convolution Layer:** Applies filters to the input image to extract features such as edges, textures, and patterns.

- **Activation Function (ReLU):** Introduces non-linearity to help the network learn complex patterns.

- **Pooling Layer (Max Pooling):** Reduces the spatial dimensions of the feature maps, making the network computationally efficient while retaining important features.

How CNN Works (Cont..)

- **Fully Connected Layer:** Connects neurons from the flattened feature map to produce the final output, such as class scores.

- **Softmax Layer:** Converts the output scores into probabilities for classification tasks.

Key Features of CNNs

- **Local Connectivity:** Neurons in each layer are connected only to a local region of the input.

- **Shared Weights (Filters):** The same filter is applied across the entire input, allowing CNNs to detect patterns regardless of their position.

- **Parameter Efficiency:** Reduced number of parameters compared to fully connected networks, making CNNs faster and less prone to overfitting.

Example - Image Classification with CNN

- **Problem:** Classify images of handwritten digits (e.g., MNIST dataset).

- **Step 1:** Input the image (28x28 pixels) to the CNN.
-
- **Step 2:** Convolutional layers extract features like edges and textures.
-
- **Step 3:** Pooling layers reduce dimensionality.
-
- **Step 4:** Fully connected layers make the final prediction, e.g., determining the digit.

Real-World Applications of CNNs

- **Self-Driving Cars:** Object detection and recognition of road signs, pedestrians, and other vehicles.

- **Medical Diagnosis:** Analysis of medical images, such as X-rays or MRIs, to detect abnormalities.

- **Facial Recognition:** Identifying and verifying human faces in images and videos.

Advantages of CNNs

- **Automatic Feature Extraction:** No manual feature engineering is required.
- **High Accuracy:** Especially in image and video recognition tasks.
- **Scalability:** Can handle large datasets efficiently.

Limitations of CNNs

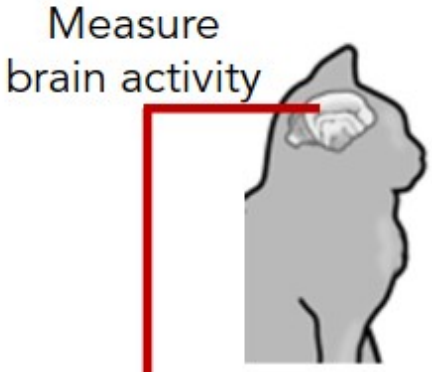
- **High Computational Cost:** Requires significant computational resources, especially for large models.
- **Data Hungry:** Needs a large amount of labeled data for training.
- **Lack of Spatial Awareness:** Struggles with recognizing relationships between objects that are far apart in the image.

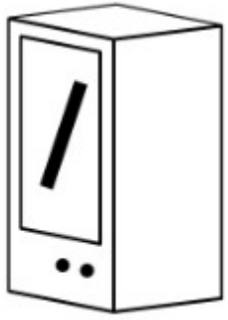
Hubel and Wiesel’s 1959 Study: The Motivation Behind CNNs

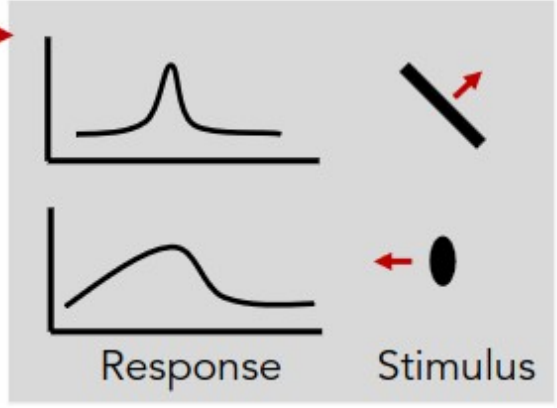
- **Background:** In 1959, neuroscientists David Hubel and Torsten Wiesel conducted groundbreaking research on the visual cortex of cats.

- **Objective:** To understand how the brain processes visual information, specifically how neurons in the visual cortex respond to different stimuli.

Hubel and Wiesel, 1959







1959
Hubel & Wiesel

Simple cells:
Response to specific rotation and orientation

Complex cells:
Response to light orientation and movement, some translation invariance

No response

Slide inspiration: Justin Johns

Key Discoveries

- **Simple Cells:** Respond to specific orientations of edges or lines in a particular location of the visual field.
- **Complex Cells:** Respond to oriented edges or lines but are less sensitive to the exact position in the field.
- **Hierarchical Processing:** Discovered that visual processing in the brain occurs in a hierarchical manner, with simple cells feeding into complex cells.

Implications of Their Findings

- **Hierarchical Pattern Recognition:** The brain processes visual information by progressively extracting more complex features.

- **Feature Detection:** Neurons act as filters that detect specific features such as edges, orientations, and movements.

- **Biological Motivation for CNNs:** This hierarchical feature extraction process inspired the design of CNNs, where layers of artificial neurons mimic the structure of the visual cortex.

Biological Inspiration for CNNs

- **Layers of CNNs vs. Visual Cortex:**

- **Convolutional Layers:** Mimic simple cells by detecting basic features like edges and lines.

- **Deeper Layers:** Mimic complex cells by detecting more abstract patterns, such as textures or shapes.

- **Local Receptive Fields:** Just like neurons in the visual cortex, CNN neurons only focus on small, localized regions of the input image, allowing them to learn spatial hierarchies of features.

Convolution- Edge detector

- **Overview:** The Sobel Edge Detector is an algorithm used to detect edges in an image using convolution.

- **Key Feature:** It emphasizes high-frequency changes (edges) by calculating the gradient of image intensity.

- **Directional Sensitivity:** Detects edges in both horizontal and vertical directions using specific convolution kernels.

Convolution- Edge detector

- Sobel Edge detector
- Vertical kernel and Horizontal kernel

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Gray scale image

- 

CNN (cont..)

- **Convolution:** Element wise multiplication and summed result

  The filter slides over the input image, multiplying its values with corresponding image pixel values and summing them up to produce a single output value at each position.

- **Stride:** No. Of rows or columns shifted.

  Stride refers to the number of pixels by which the filter moves across the input image during convolution. It controls the spatial resolution of the output feature map.

Padding

- Padding refers to adding extra pixels around the input image, typically filled with zeros, before applying convolution.
- **Purpose:** To control the spatial dimensions of the output feature maps, prevent loss of information, and ensure that edge features are captured
- **Input** **output**
- (nxn) (n-k+1)X (n-k+1)

  K is filter size
- (nxn) (n-k+2p+1)X (n-k+2p+1)
- (nxn) (floor(n-k+2p/s)+1)X (floor(n-k+2p/s)+1)

Padding

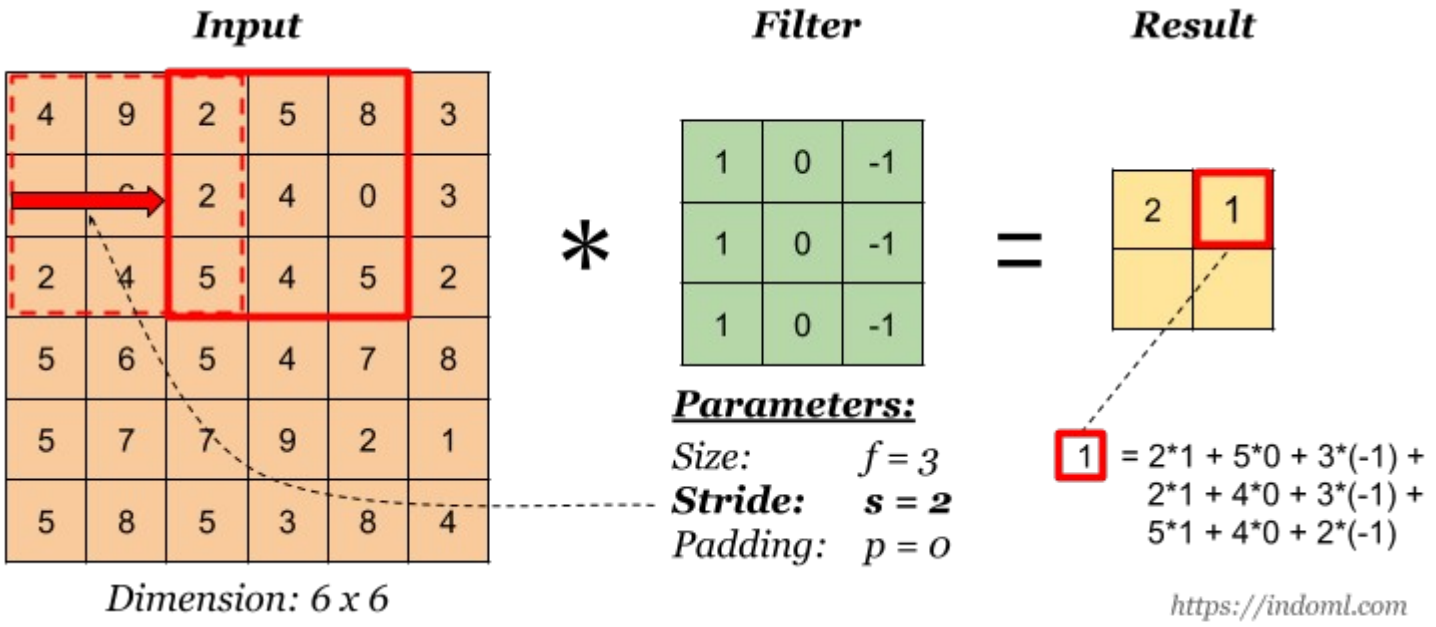
0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

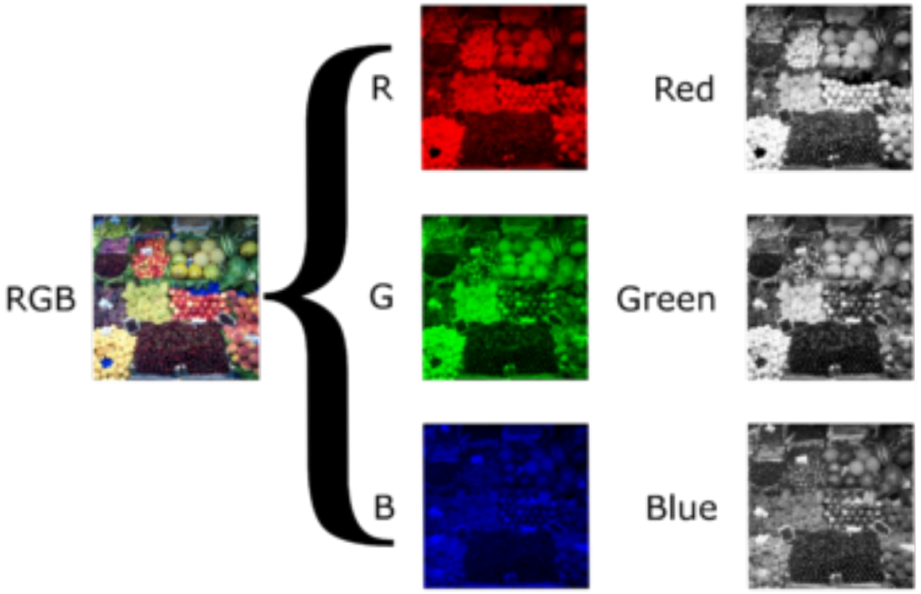
0	-1	0
-1	5	-1
0	-1	0

114	328	-26	470	158
53	266	-61	-30	344
403	116	-47	295	244
108	-135	256	-128	344
314				

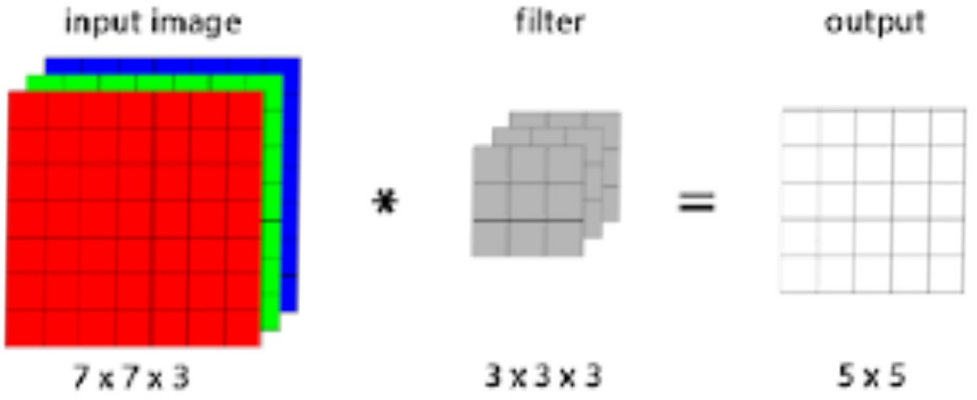
Strides


Input
Filter
Result
4 9 2 5 8 3
2 4 0 3
2 4 5 4 5 2
5 6 5 4 7 8
5 7 7 9 2 1
5 8 5 3 8 4
1 0 -1
1 0 -1
1 0 -1
2 1
Parameters:
Size: f = 3
Stride: s = 2
Padding: p = 0
1 = 2*1 + 5*0 + 3*(-1) +
2*1 + 4*0 + 3*(-1) +
5*1 + 4*0 + 2*(-1)
Dimension: 6 x 6
https://indoml.com

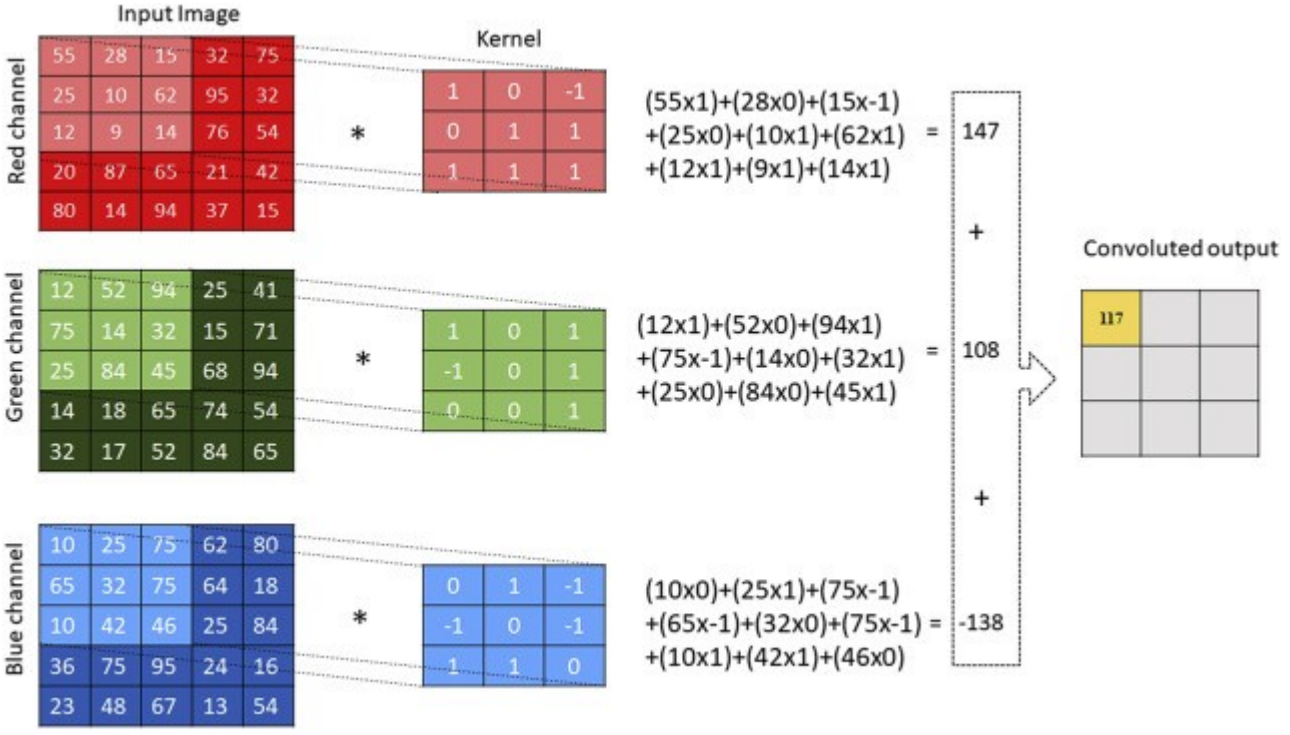
Convolution on RGB images



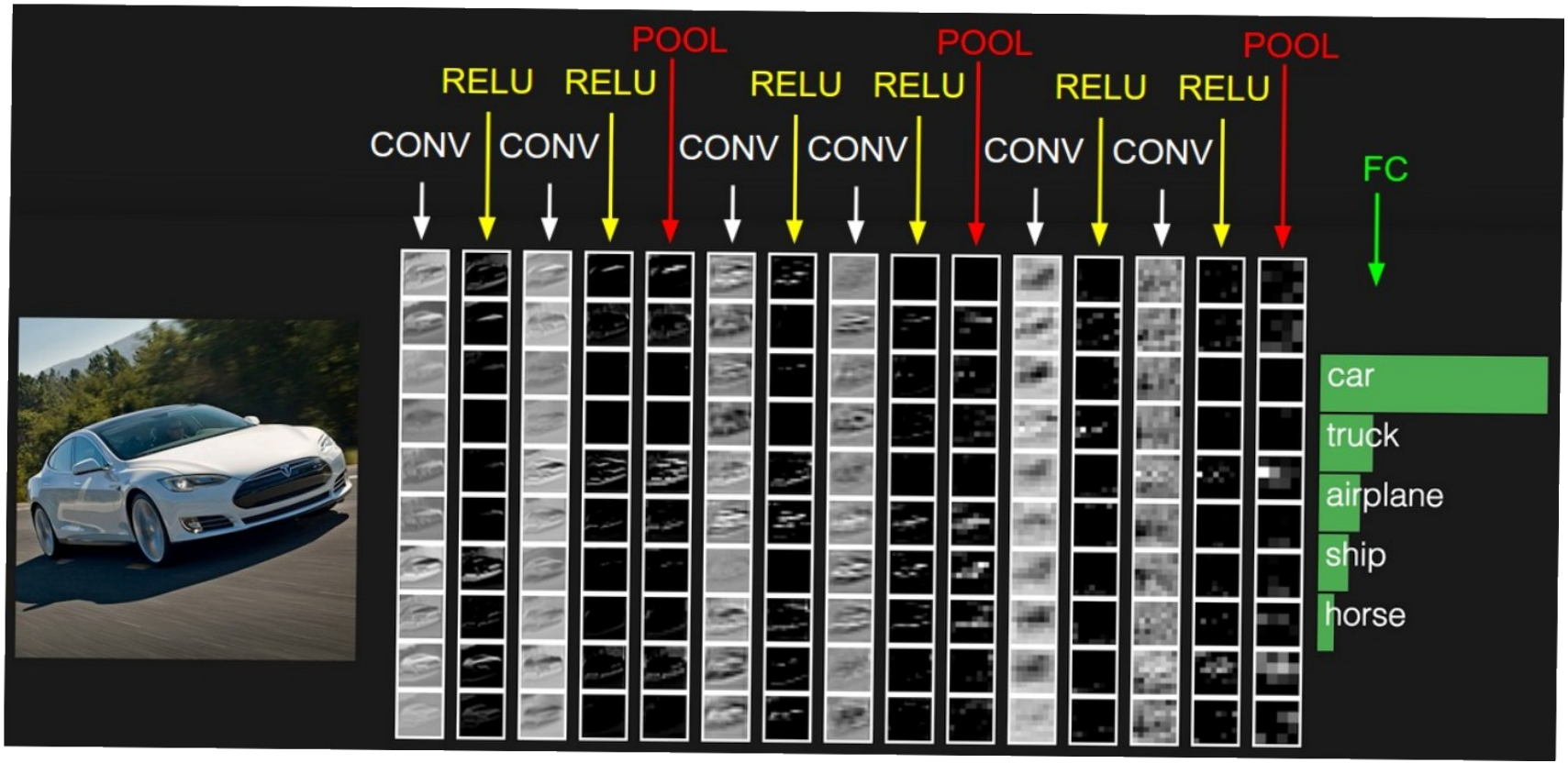
Convolution on RGB images(cont..)



Convolution on RGB images(cont..)



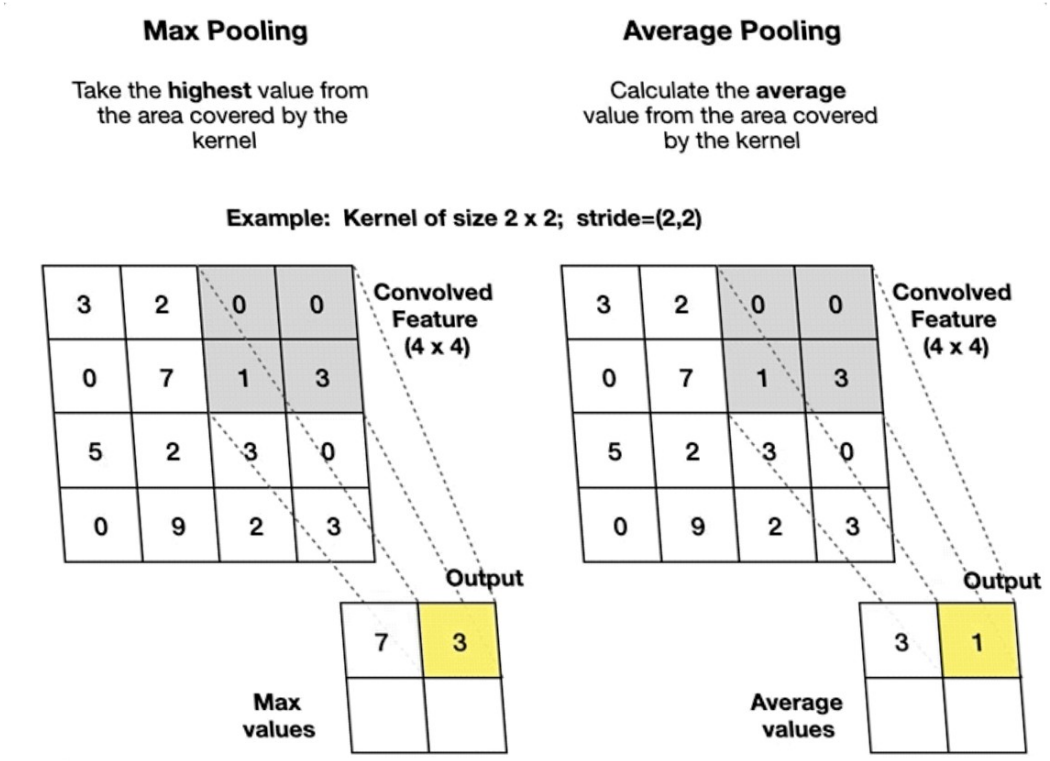
Convolution Layer



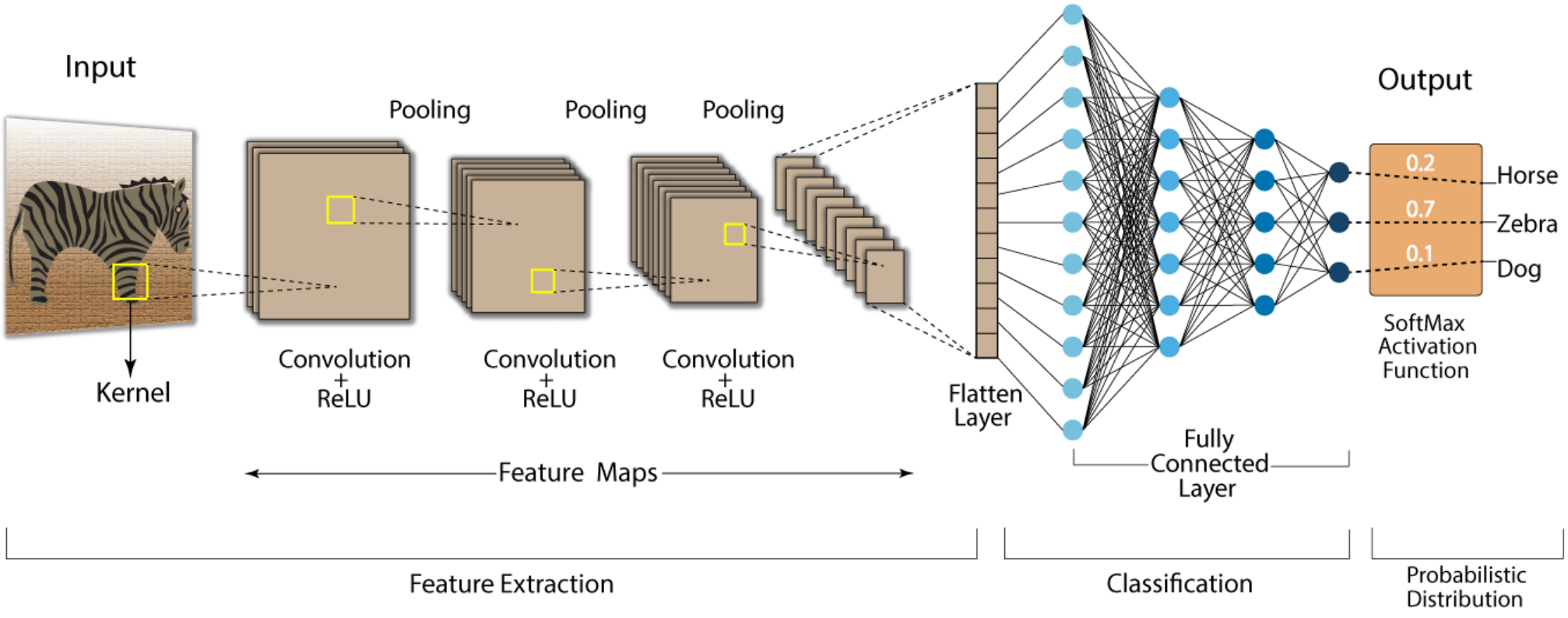
Pooling

- Downsampling operation on each feature map
- Location Invariant
- Rotation Invariant
- Scale Invariant

Pooling (Cont..)



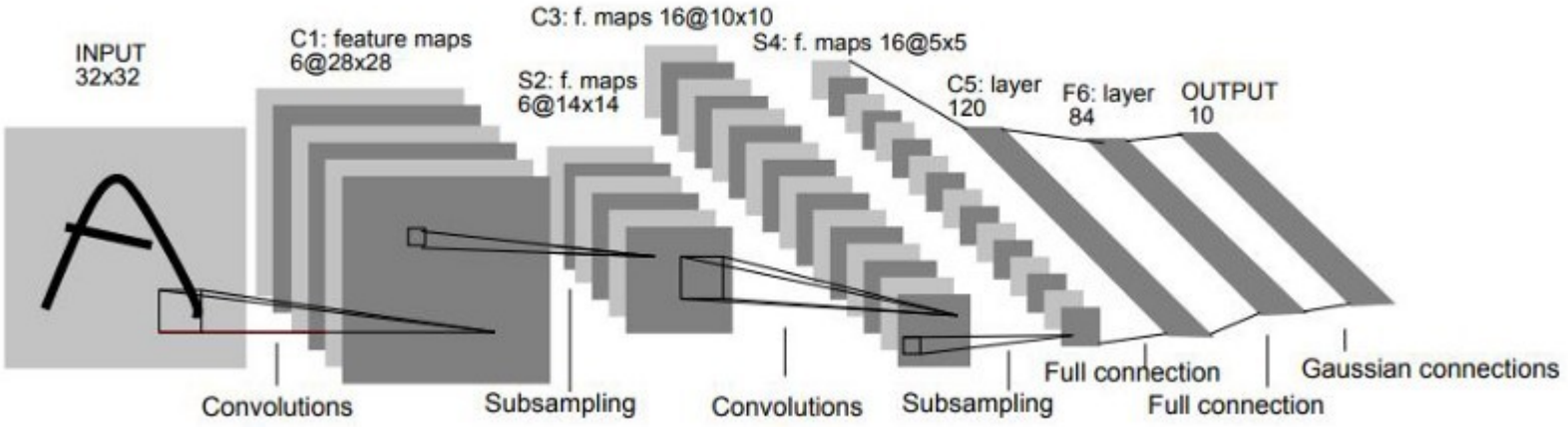
Convolution Neural Network (CNN)



LeNET

- Yann Lecunn introduced in the year 1998.
- One of the first CNN architectures.
- For detecting hand writing digits (MNIST datasets)
- Simple architecture with convolutional, pooling, and fully connected layers.
- 2 Convolutional layers, 2 Subsampling layers, 1 Fully connected layer.

LeNet


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

LeNet

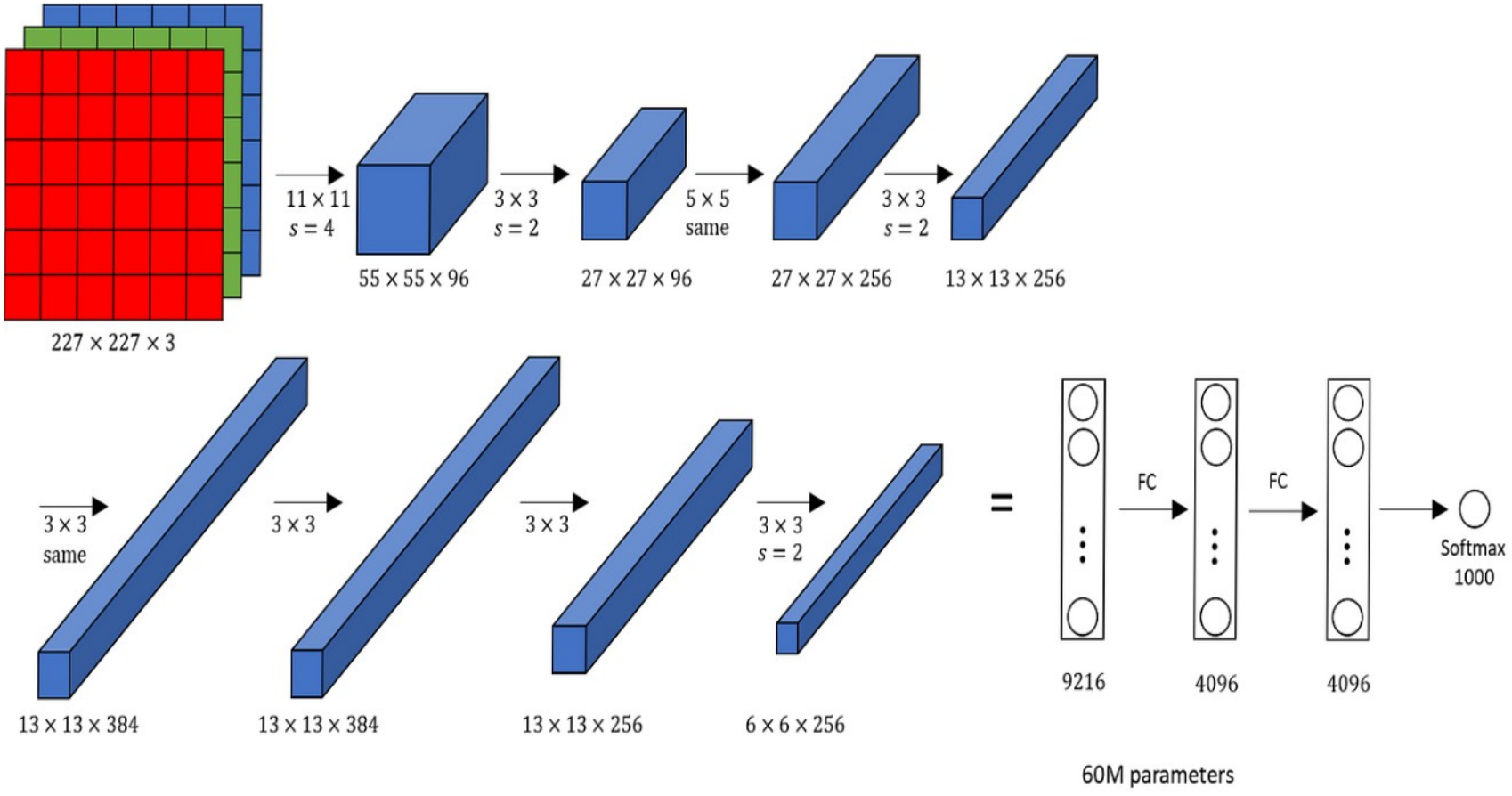
- Only 60K parameters
- General structure:

  conv->pool->conv->pool->FC->FC->output
- Different filters look at different channels
- Sigmoid and Tanh nonlinearity

AlexNet

- Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton.
- Won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012.
- Introduced ReLU activation for faster training.
- Used Dropout to reduce overfitting and GPU acceleration for training.
- 5 Convolutional layers, 3 Max-pooling layers, 3 Fully connected layers.
- 60 million parameters and 650,000 neurons.

AlexNet


227 × 227 × 3
11 × 11
s = 4
55 × 55 × 96
3 × 3
s = 2
27 × 27 × 96
5 × 5
same
27 × 27 × 256
3 × 3
s = 2
13 × 13 × 256
3 × 3
same
13 × 13 × 384
3 × 3
13 × 13 × 384
3 × 3
13 × 13 × 256
3 × 3
s = 2
6 × 6 × 256
=
9216
FC
4096
FC
4096
Softmax
1000
60M parameters

AlexNet

- Trained on GTX 580 GPU with only 3 GB of memory.
- •Network spread across 2 GPUs, half the neurons (feature maps) on each GPU.
- CONV1, CONV2, CONV4, CONV5:

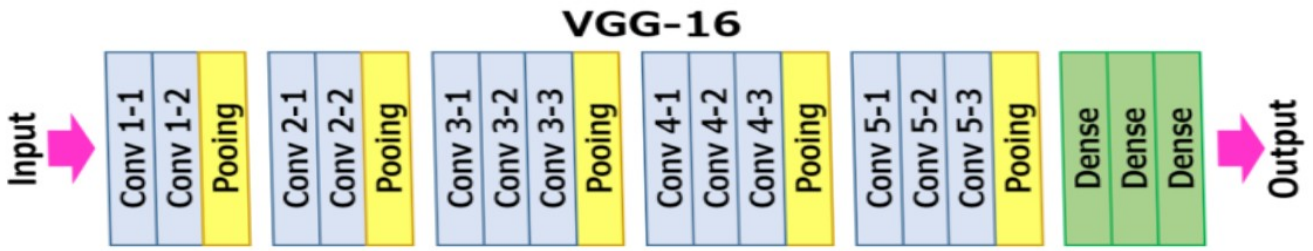
  Connections only with feature maps on same GPU.
- CONV3, FC6, FC7, FC8:

  Connections with all feature maps in preceding layer, communication across GPUs.

VGG-16

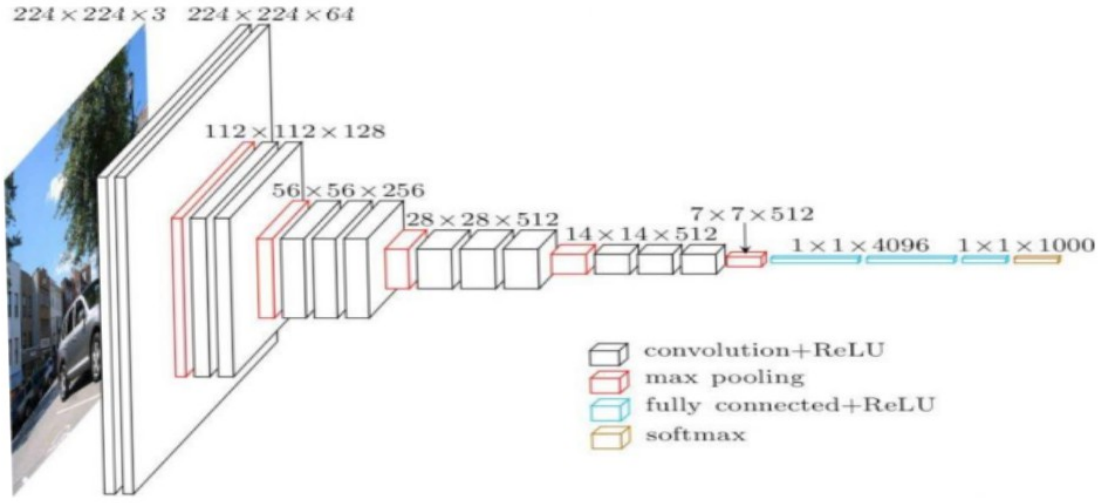
- Visual Geometry Group, Oxford in the year of 2014
- Emphasized simplicity and depth with smaller (3x3) convolution filters.
- Deeper network with up to 19 layers.
- Improved accuracy over previous models by increasing depth.
- 16-19 layers with repeated convolutional and pooling blocks.
- Emphasis on using uniform filter size throughout the network.

VGG-16 Architecture

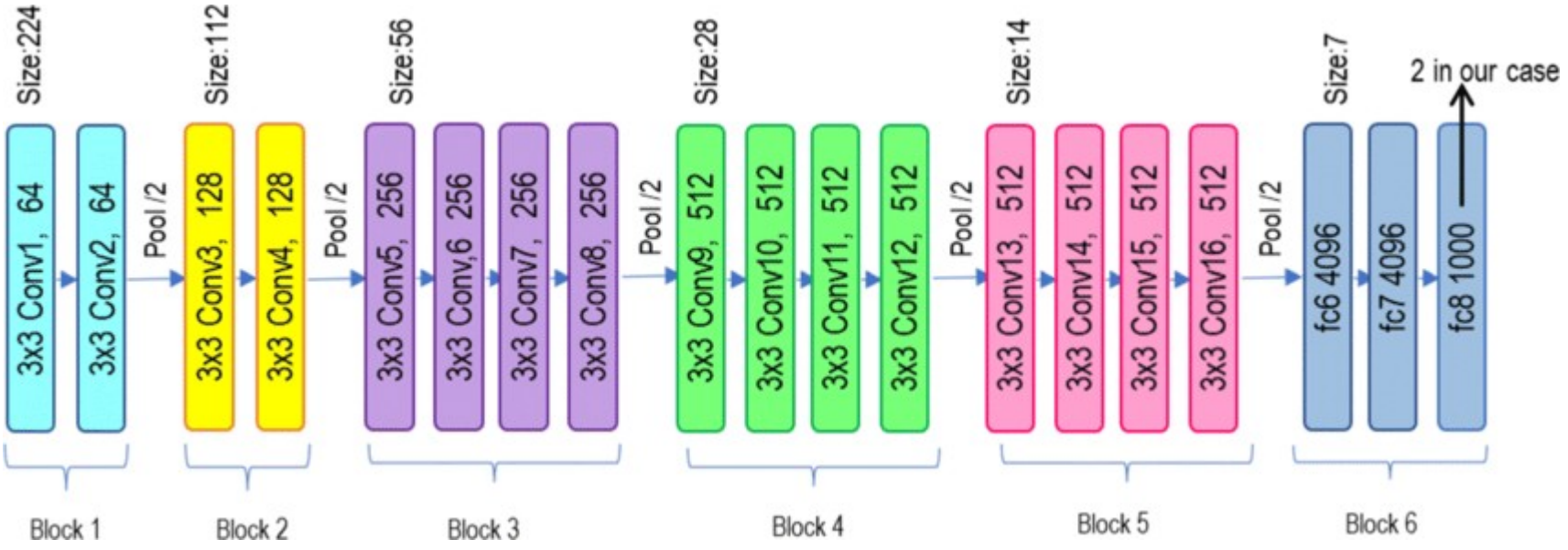


The Architecture

The architecture depicted below is VGG16.



VGG-19 Architecture

- 

VGG-16

- ILSVRC’14 2nd in classification, 1st in localization
- Similar training procedure as AlexNet
- Use VGG16 or VGG19 (VGG19 only slightly better, more memory)
- Trained on 4 Nvidia Titan Black GPUs for two to three weeks

VGG-16

- Convolutional neuralnetworks have to have a **deep network of layers in order for this hierarchical representation** of visual data to work

GoogleNet

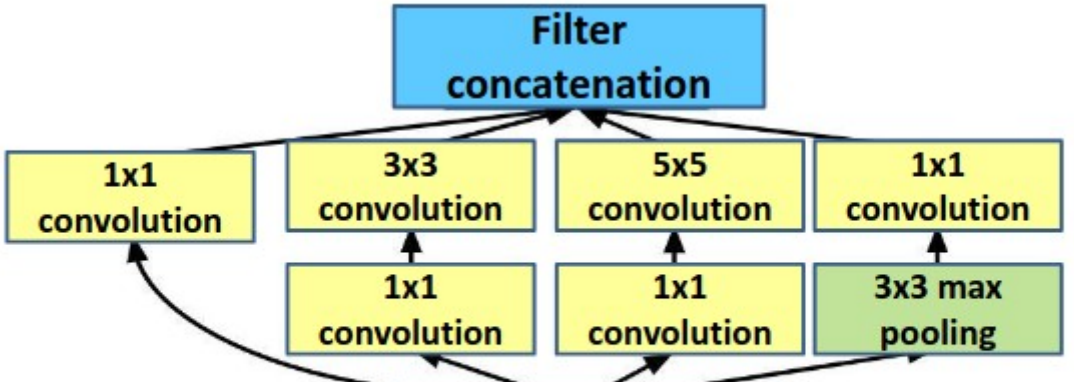
- Going Deeper with Convolutions - Christian Szegedy et
- al.; 2015
- ILSVRC 2014 competition winner
- Also significantly deeper than AlexNet
- x12 less parameters than AlexNet
- Focused on computational efficiency

GoogleNet

- Google Research team
- Introduced Inception modules that allow **multiple filter sizes to operate at the same level.**
- Efficient architecture with fewer parameters compared to traditional CNNs.
- Uses 1x1 convolutions for dimensionality reduction.
- Inception modules combine convolutions and pooling within the same layer.
- 22 layers deep with a complex multi-branch architecture.

GoogleNet

- **Inception Module:** design a good local network topology (network within a network) and then stack these modules on top of each other



GoogLe Net

- Improved utilization of the computing resources inside the network.
- Design that allows for increasing the depth and width of the network while keeping the computational budget constant

GoogleNet

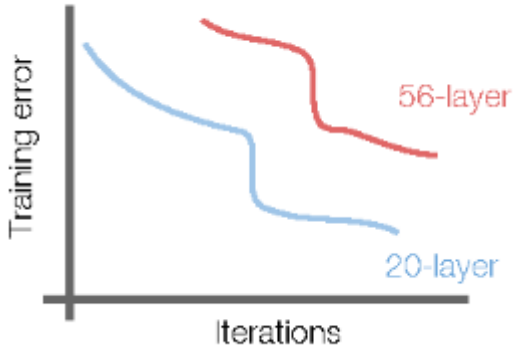
- Introduced the idea that CNN layers **didn’t always have to be stacked up sequentially**.
- Coming up with the Inception module, the authors showed that a creative structuring of layers can lead to improved performance and computationally efficiency.

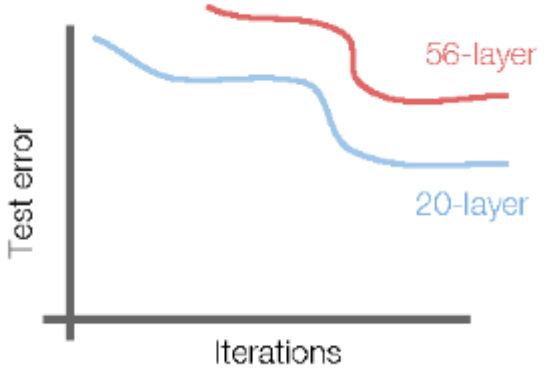
ResNet

- Extremely deep network – 152 layers
- Deeper neural networks are more difficult to train.
- Deep networks suffer from vanishing and
- exploding gradients.
- Present a residual learning framework to ease the training of networks that are substantially deeper than those used previously.

ResNet

- What happens when we continue stacking deeper layers on a convolutional neural network?



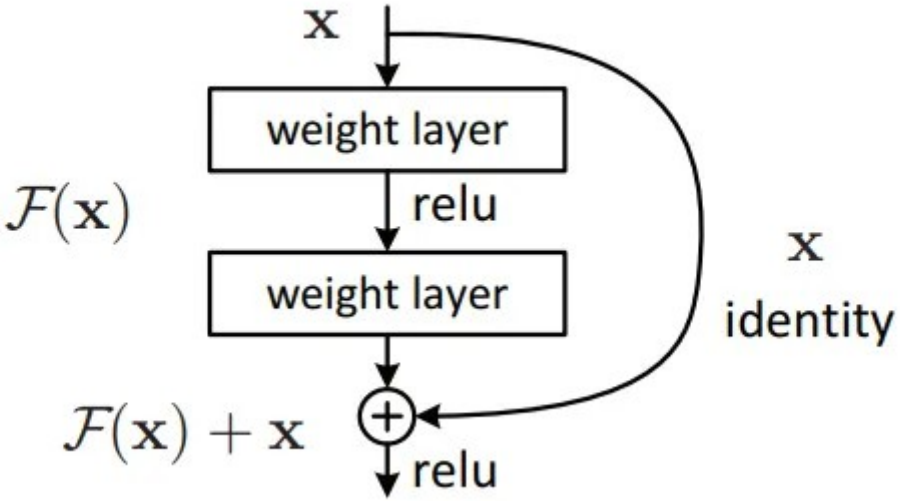


ResNet

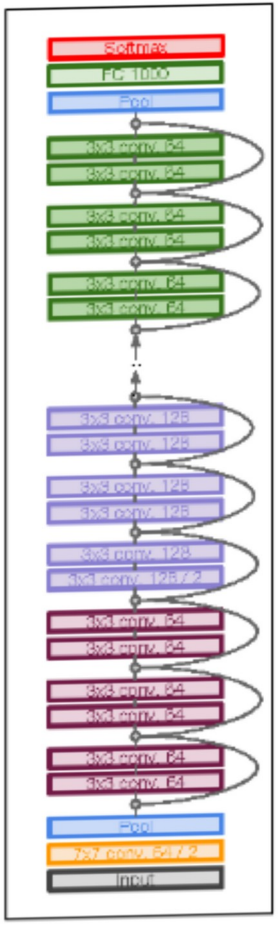
- Microsoft Research team in the year of 2015
- Introduced the concept of Residual Learning with **skip connections.**
- Overcomes the vanishing gradient problem in deep networks.
- Can scale up to hundreds of layers without degradation.
- Uses residual blocks with skip connections (identity mapping).

ResNet

- Skip Connections
- Adding layers it should not hurt the performance.



ResNet



Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes