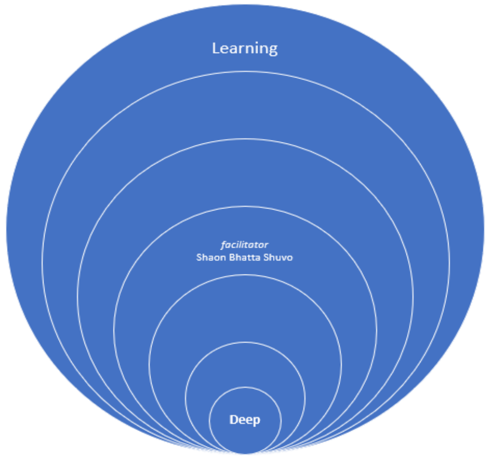


Workshop (Part2) on Deep Learning

facilitator

Shaon Bhatta Shuvo

Ph.D. Candidate, School of Computer Science

University of Windsor, Ontario, Canada.

Email: shuvos@uwindsor.ca

University of Windsor

Key Topics

- Why Convolutional Neural Networks (CNN)?
- Introduction to CNN
- CNN Steps
- Some popular CNN models
- Handwritten Character Recognition on MNIST Dataset (Hands-on)
- Concept of Underfitting, Overfitting and Regularizations
- Transfer Learning and InceptionV3 model
- Canadian Medicinal Plant Detection using Transfer Learning (Hands-on)
- How to learn Deep Learning more efficiently?

Let's Recap from Workshop Part 1

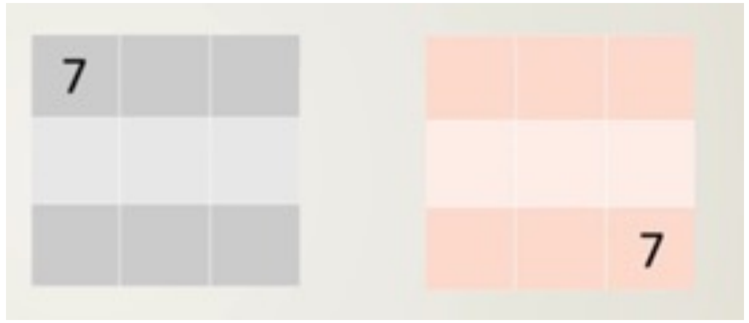
Let's visualize, log on to: **www.playground.tensorflow.org**

Hypothetical Interview of an ML Engineer :

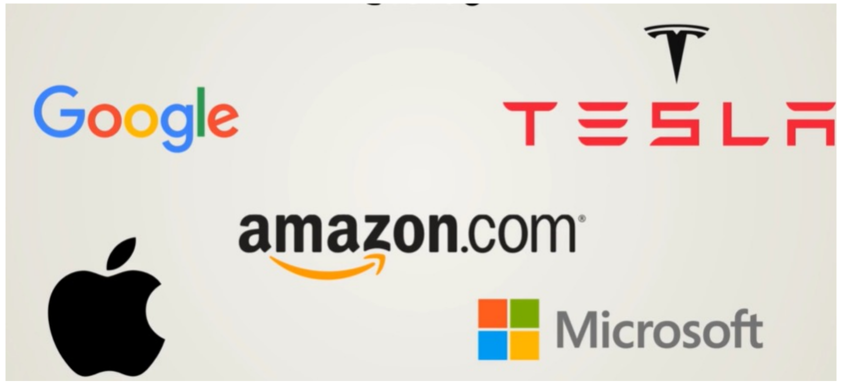
Interviewer: What is your biggest strength?
Candidate: I am an expert in Machine Learning.
Interviewer: What's 5+16?
Candidate: It's 4.
Interviewer: Not even close. It's 21.
Candidate: It's 15.
Interviewer: Close but wrong. It's still 21.
Candidate: It's 20.
Interviewer: No, it's 21.
Candidate: It's 21.
Interviewer: You're hired.

Why Convolutional Neural Networks(CNN)?

- ❑Spatial Proximities are preserved.
- ❑A detail is looked for everywhere in a photo.



- ❑Some instances of CNN application:
  - ✓Robot Vision
  - ✓Self Driving Car
  - ✓Facebook Tagging
  - ✓Apple’s Face Recognition
  - ✓Facebook messenger/snapchat filters
  - ✓Google assistant to give artificial sound in a more human like shape.
  - ✓Amazon Recommender Engine, etc.



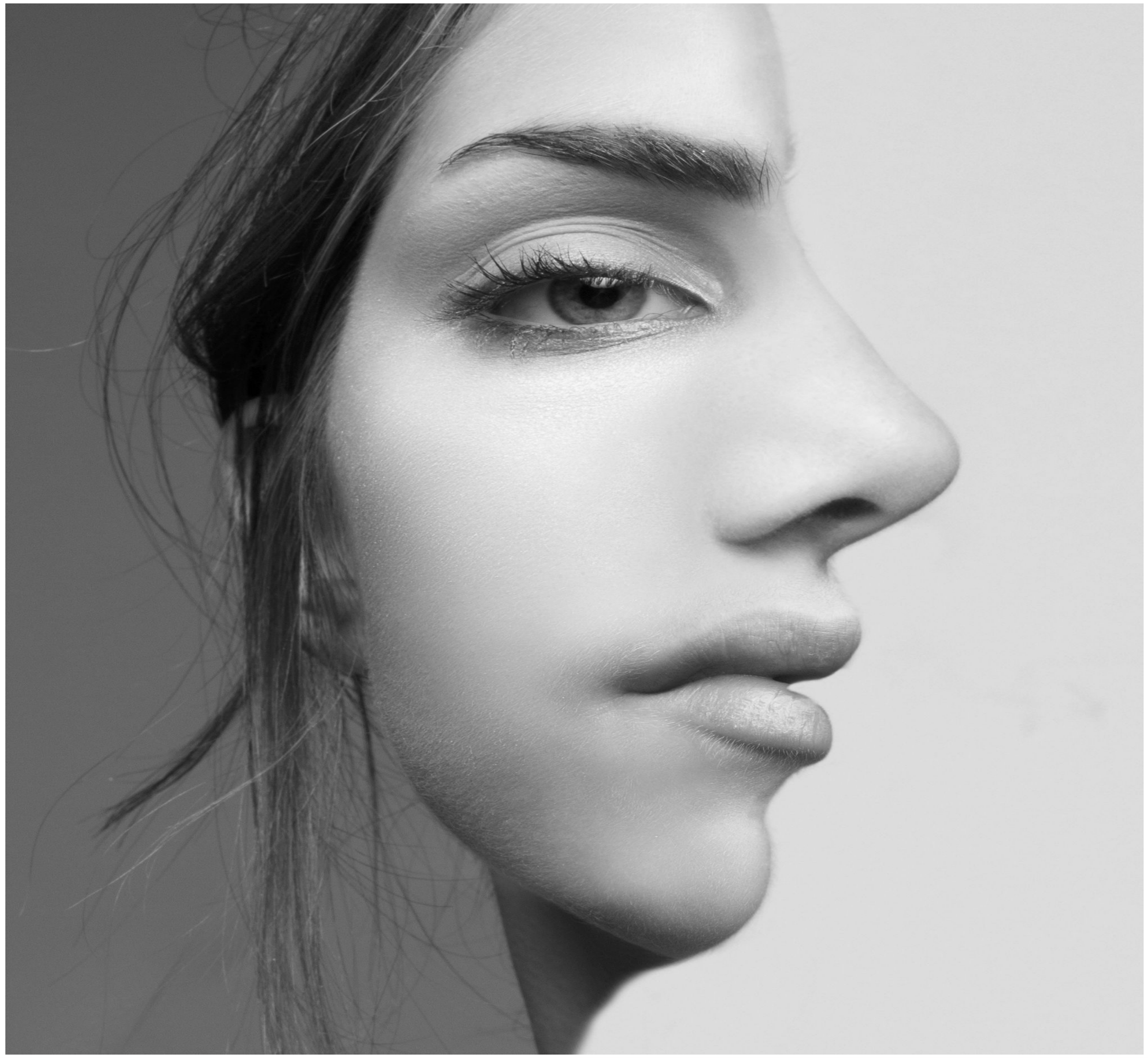
Introduction to Convolutional Neural Network (CNN)

- In deep learning, a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep neural networks, most commonly applied to analyzing visual imagery.
- Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.
- The Convolutional Neural Network (CNN) has shown excellent performance in many computer vision and machine learning problems.

[Specially, Image Classification, Image Semantic Segmentation, Object Detection in Image etc.]

Sources: https://en.wikipedia.org/wiki/Convolutional_neural_network
https://cs.nju.edu.cn/wujx/paper/CNN.pdf

Which feature does your brain identify?



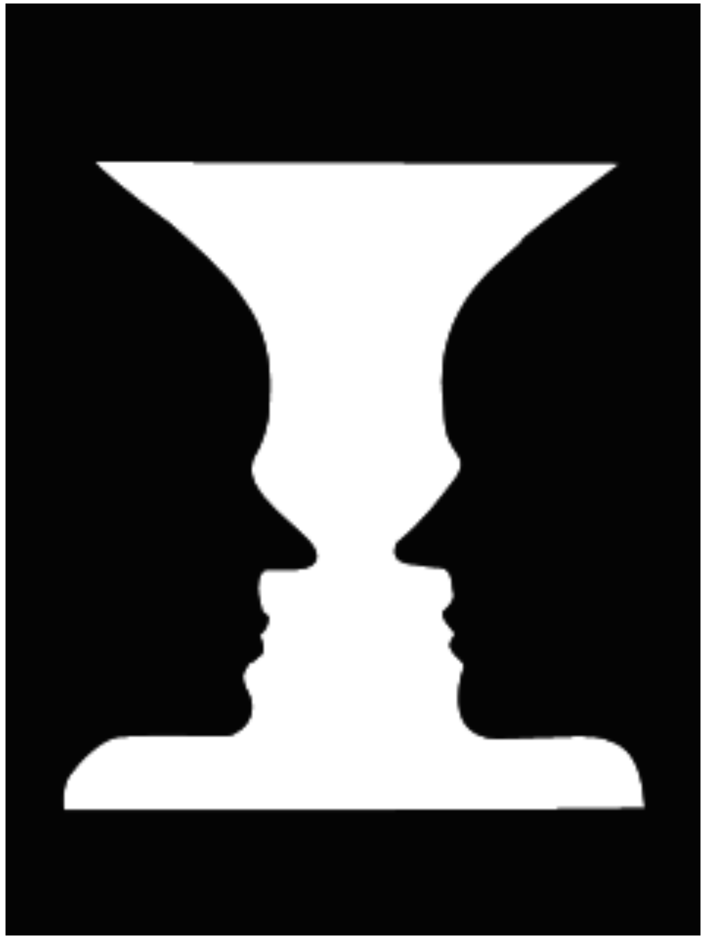
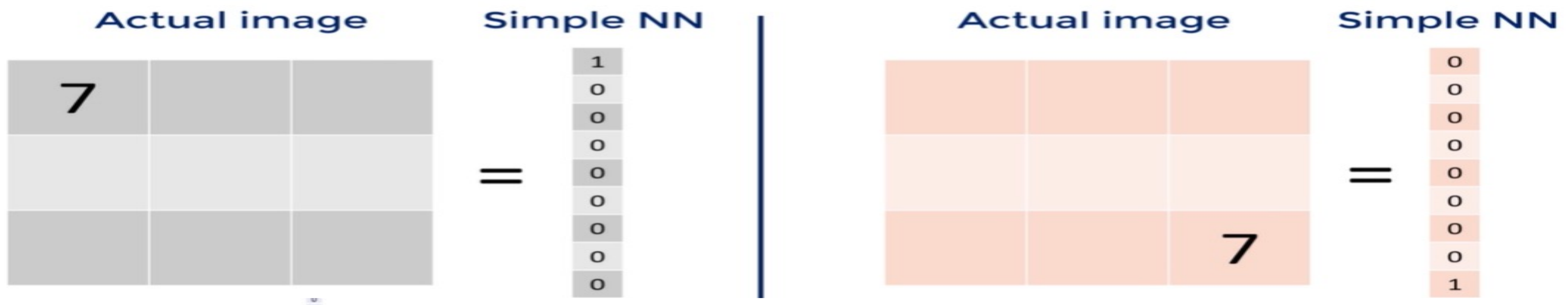
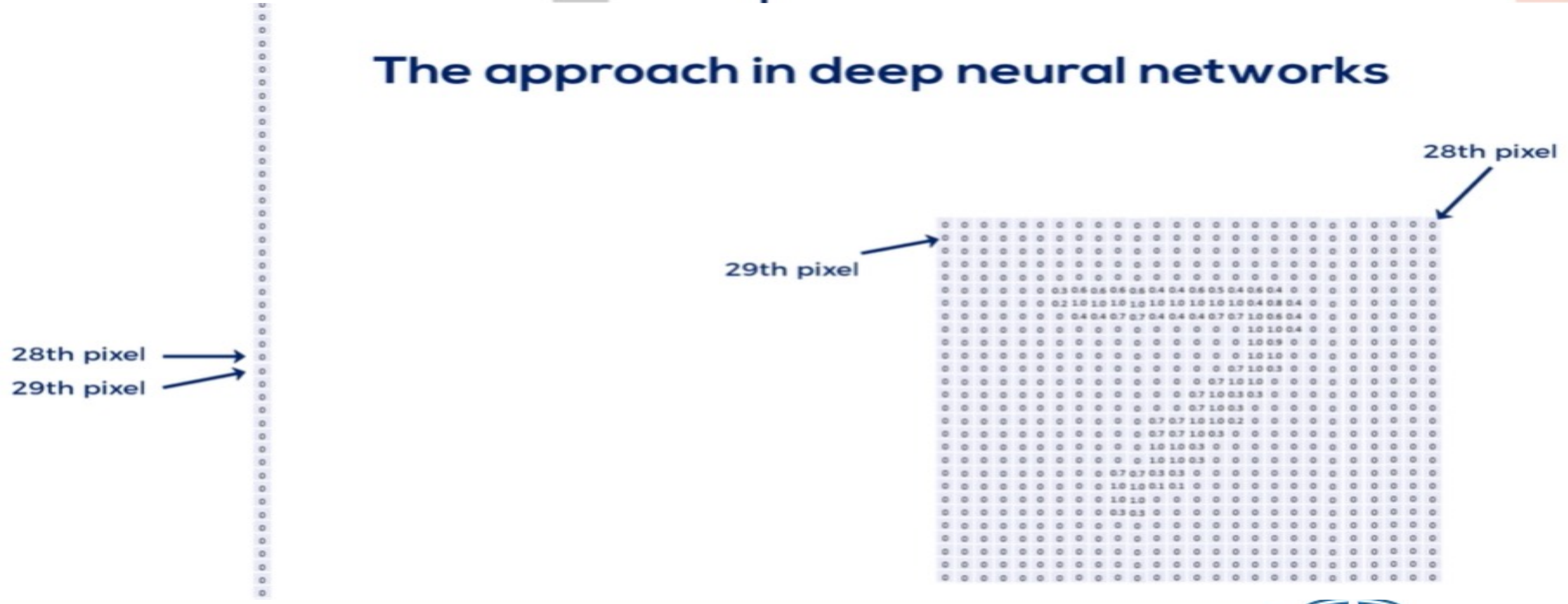


Image saved from: printerest.ca

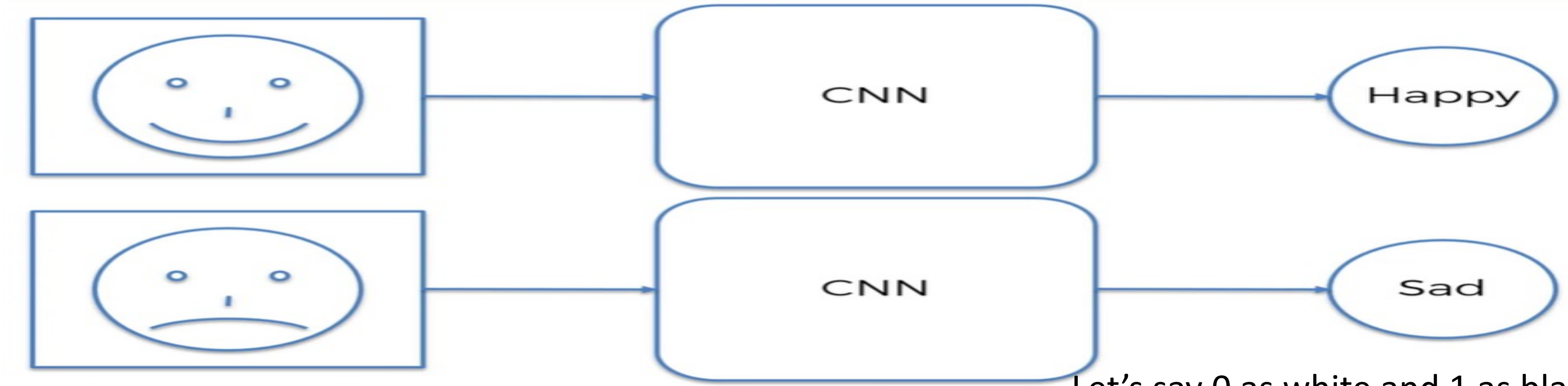
Spatial Information is very important

(Spatial proximities =location of areas relative to one another)

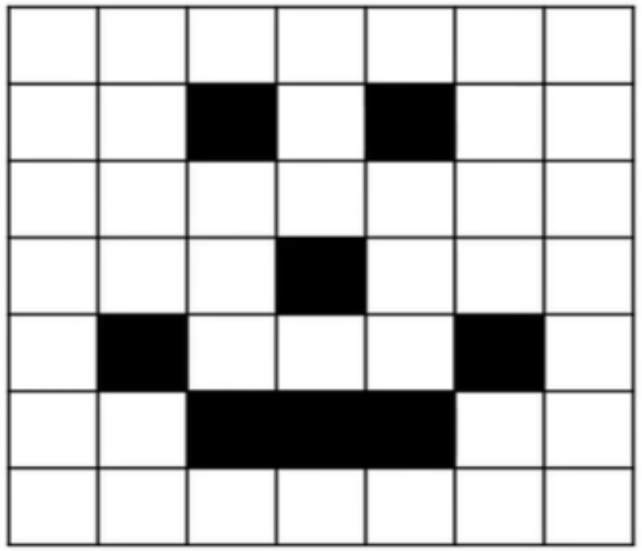


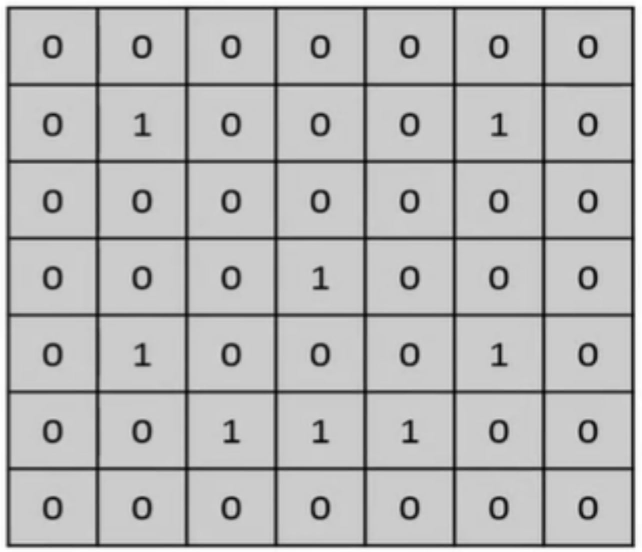


CNN as a Blackbox.



Let’s say 0 as white and 1 as black

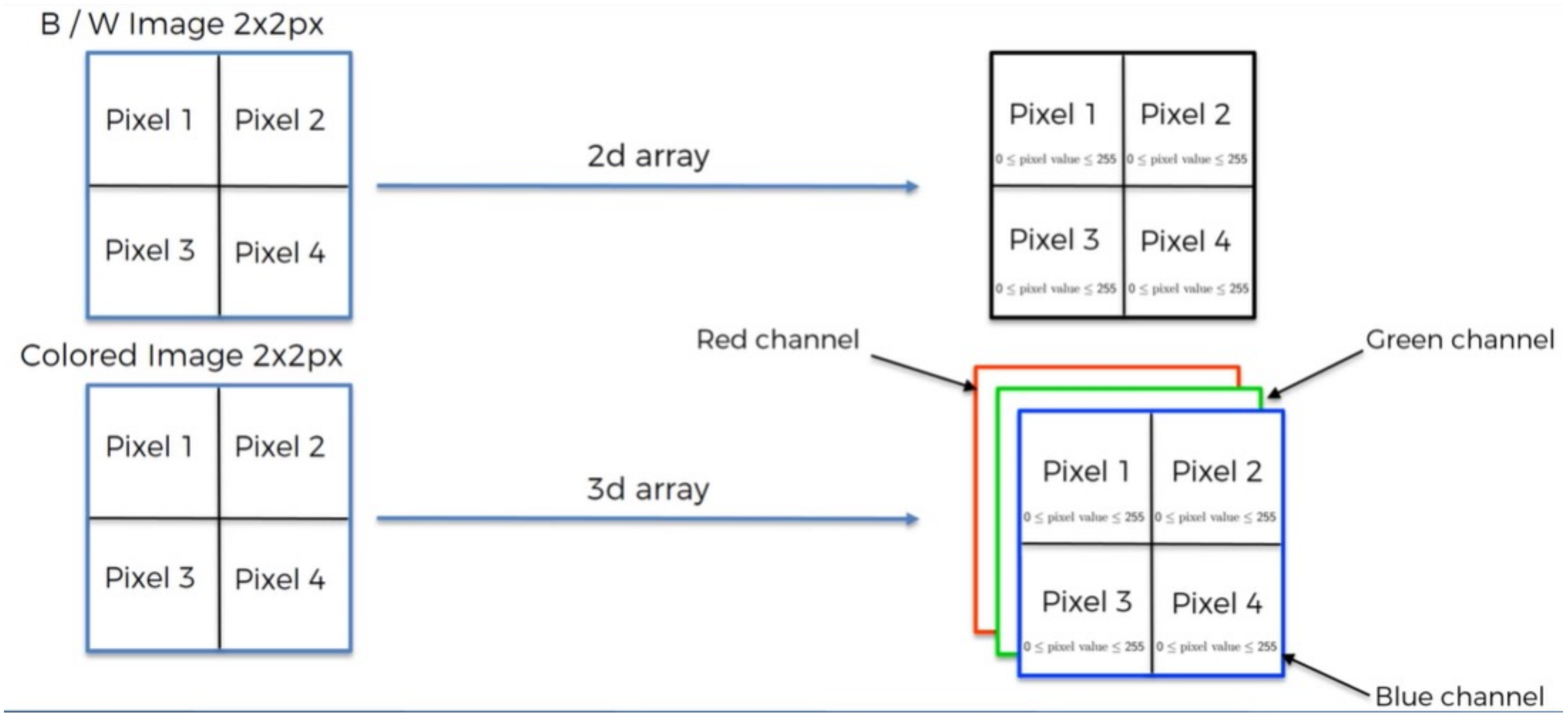




0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Image source: SuperDataScience

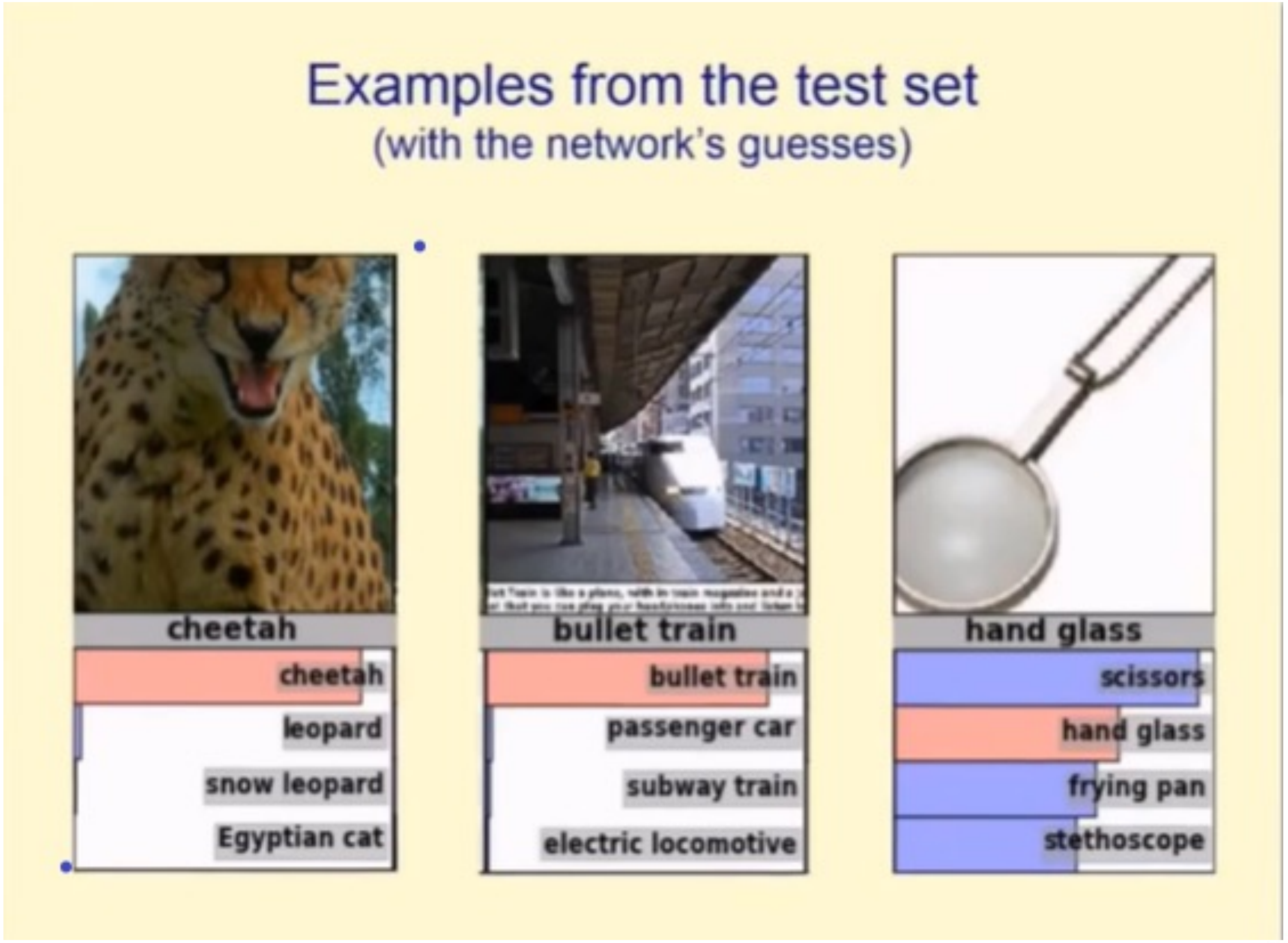
Actual Image Representation



B / W Image 2x2px
Pixel 1
Pixel 2
Pixel 3
Pixel 4
2d array
Pixel 1
$0 \leq$ pixel value ≤ 255
Pixel 2
$0 \leq$ pixel value ≤ 255
Pixel 3
$0 \leq$ pixel value ≤ 255
Pixel 4
$0 \leq$ pixel value ≤ 255
Colored Image 2x2px
Pixel 1
Pixel 2
Pixel 3
Pixel 4
3d array
Red channel
Green channel
Blue channel
Pixel 1
$0 \leq$ pixel value ≤ 255
Pixel 2
$0 \leq$ pixel value ≤ 255
Pixel 3
$0 \leq$ pixel value ≤ 255
Pixel 4
$0 \leq$ pixel value ≤ 255

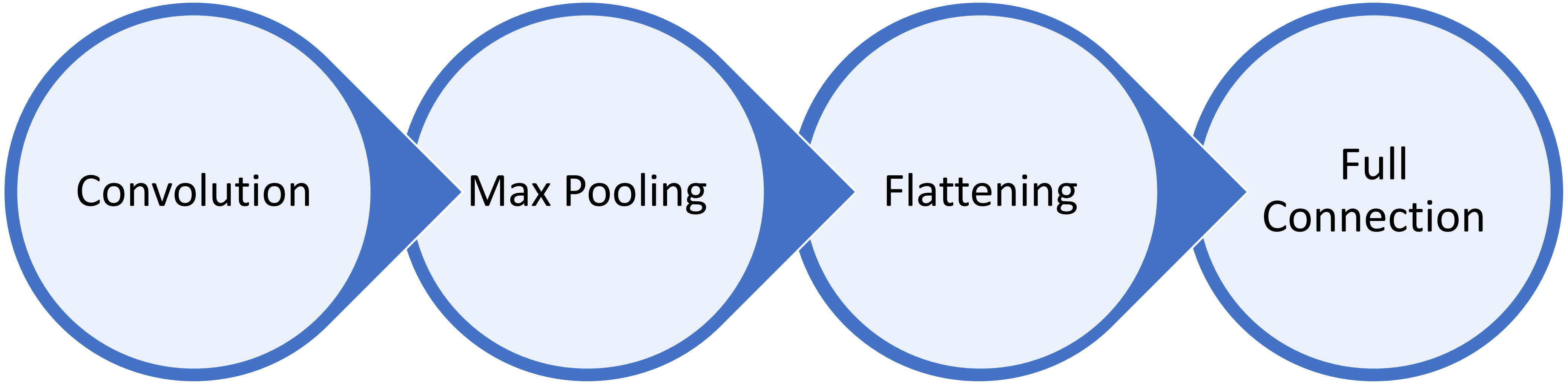
Image source: SuperDataScience

CNN predicts possible level based on probability calculation

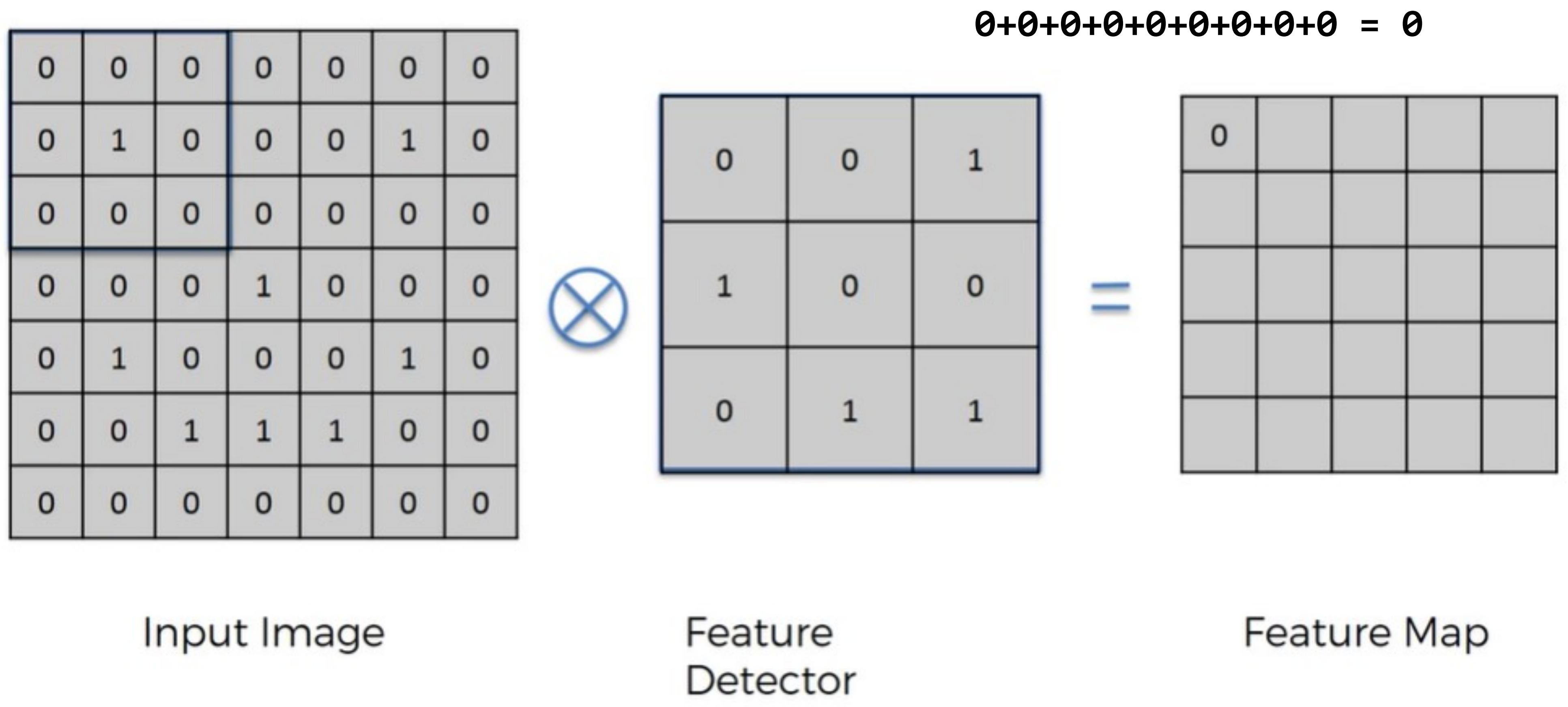


Source: A talk by Geoffrey Hinton.

CNN Steps



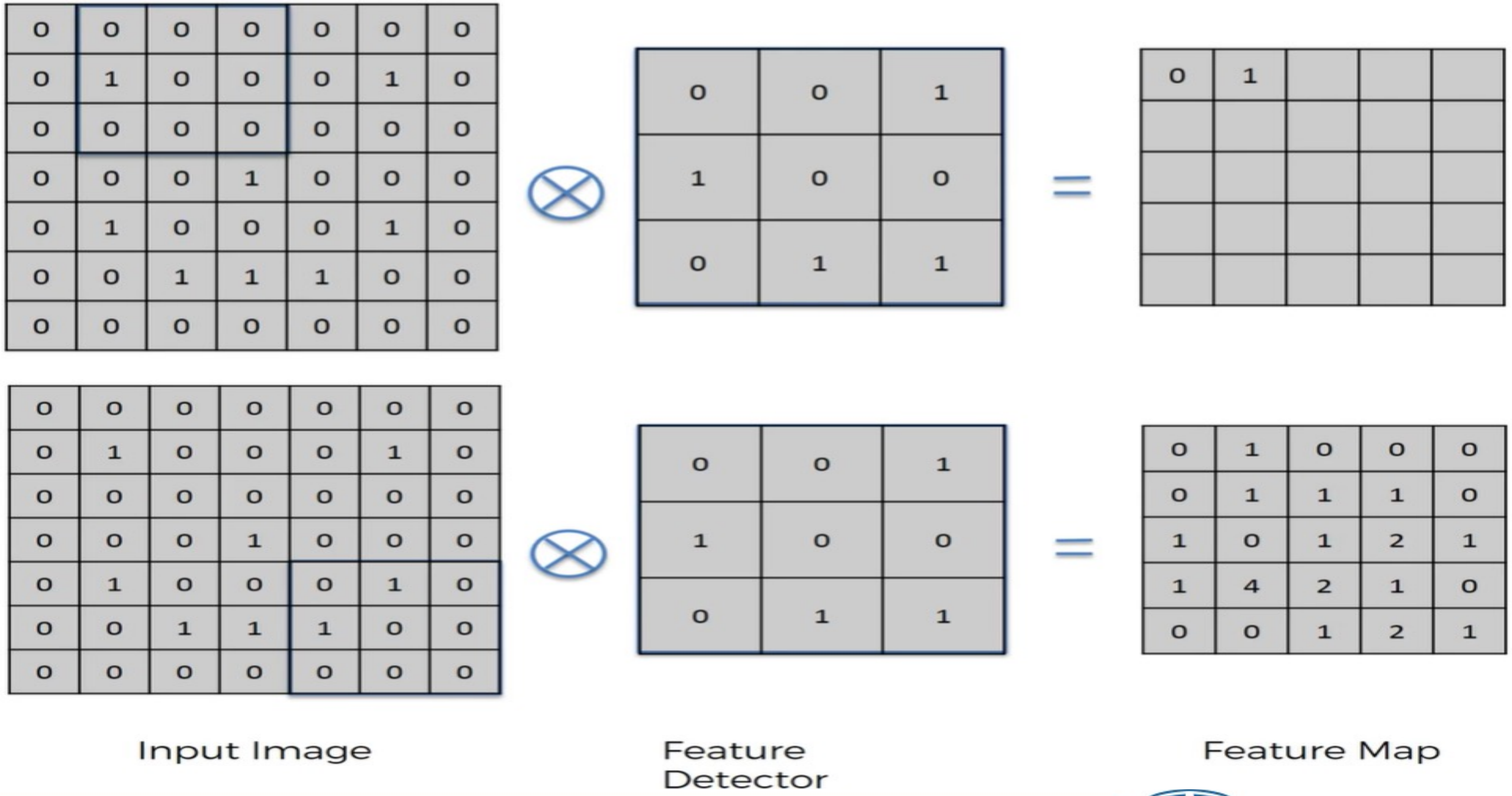
Step 1: Convolution



Element-wise (Position-wise) multiplication and then add them all to get feature map.
Feature Detector is also known as Kernel or Filter.
Features Map is also known as Convolved Feature or Activation Map.

Step 1: Convolution

Stride is the number of pixels shifts over the input matrix (vertically and horizontally). Here we choose the stride of 1.



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

0	0	1
1	0	0
0	1	1

0	1			

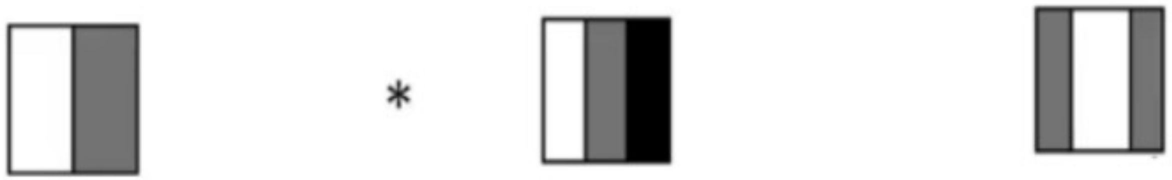
0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

0	0	1
1	0	0
0	1	1

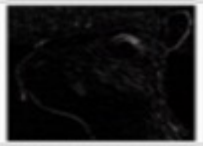
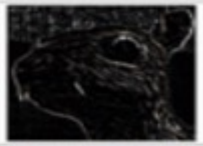
0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Step 1: Convolution [Some Common Filters]

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$

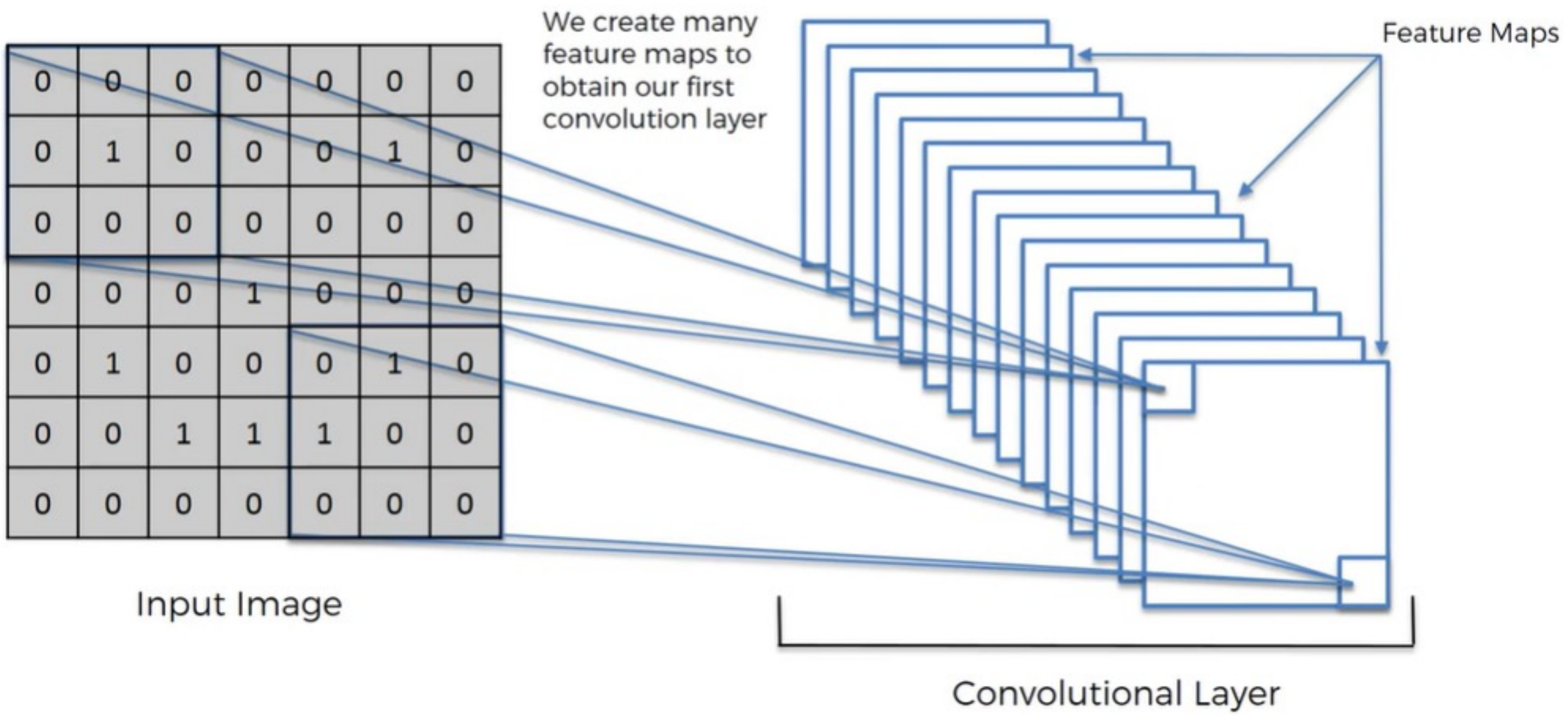


Higher pixel values represent the brighter portion of the image and the lower pixel values represent the darker portions. This is how we can detect a vertical edge in an image.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

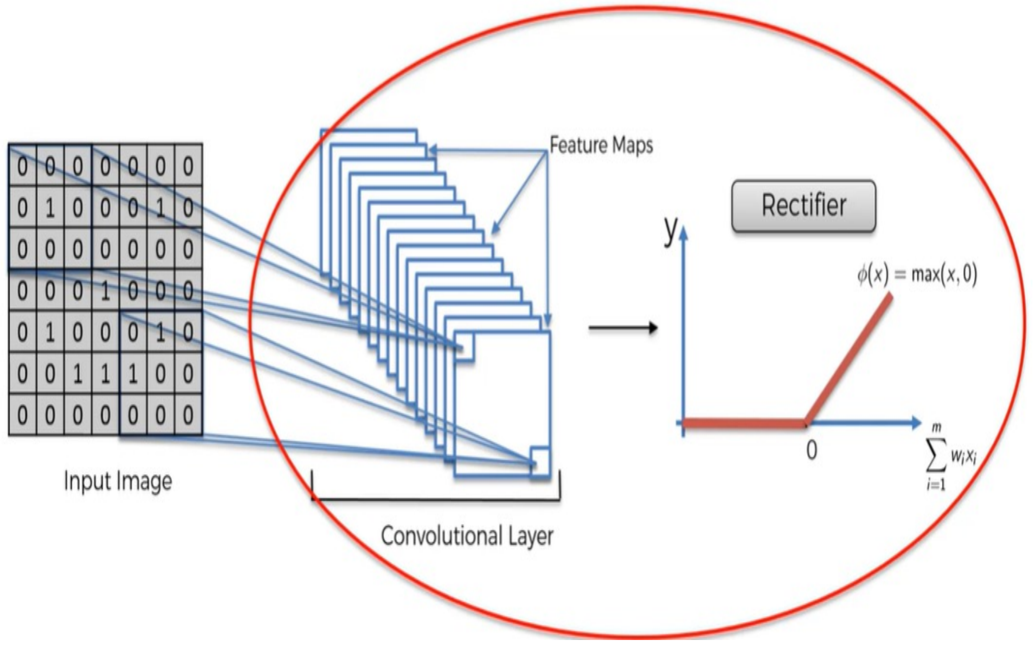
Step 1: Convolution

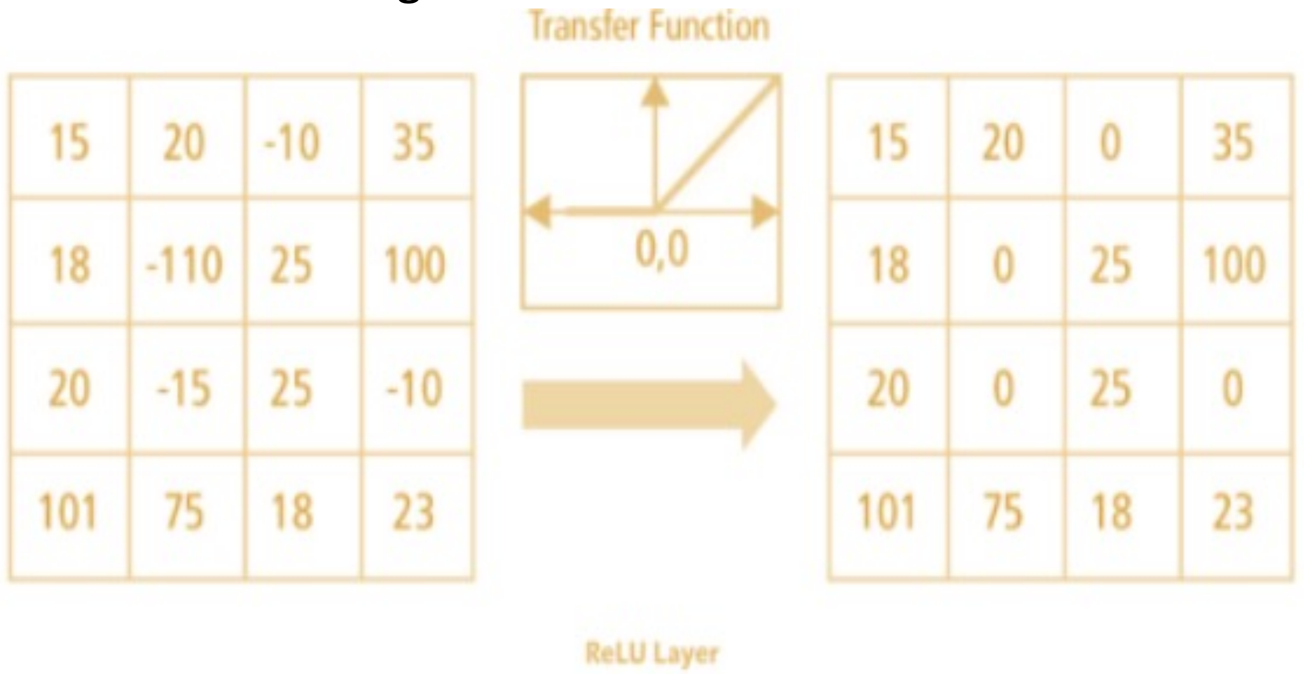
During the training the network decides which features are important to identify the images certain categories.



Step 1(B): Non-linearity (ReLU Layer)

- ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ***f(x) = max(0,x).***
- Why ReLU is important?
  - ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real-world data would want our ConvNet to learn would be non-negative linear values.





15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

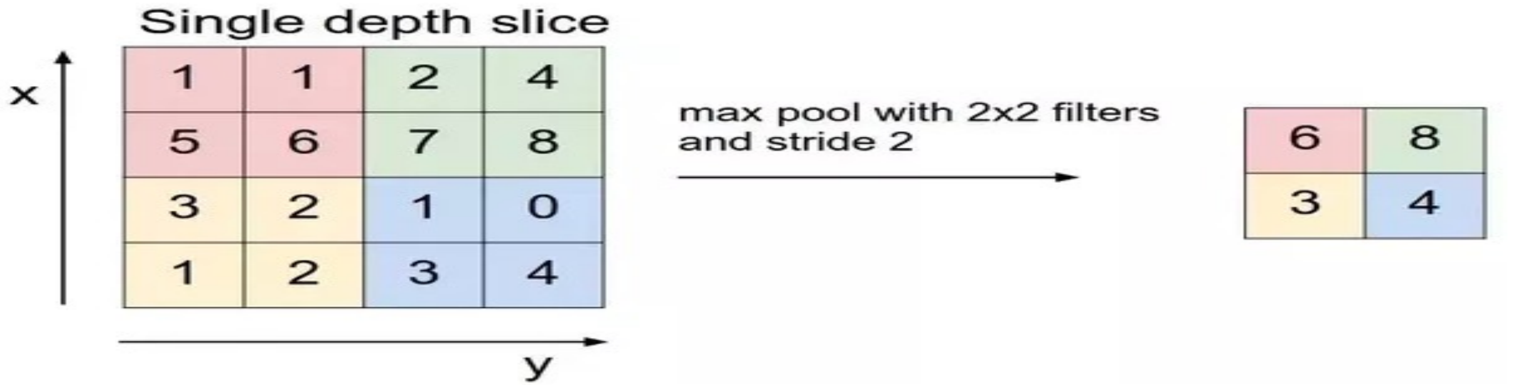
15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

Pooling Layer

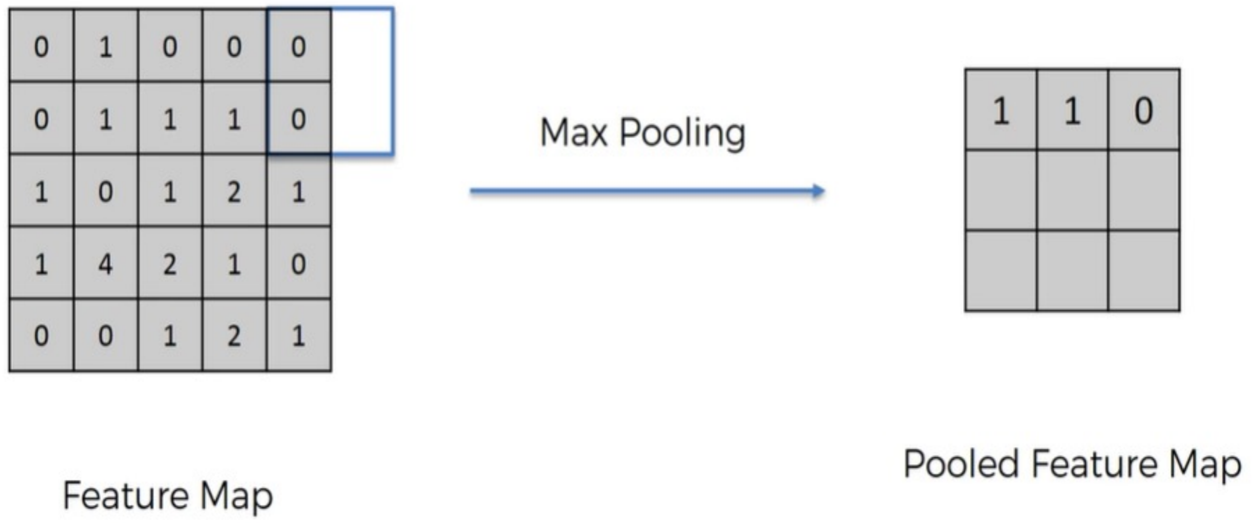
- Pooling layers section would reduce the number of parameters when the images are too large.
- Spatial pooling also called subsampling or down-sampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:
  - Max Pooling
  - Average Pooling
  - Sum Pooling
- Benefits of Pooling:
  - Reduce the size.
  - We can still able to preserve most of the relevant features irrespective of their position.
  - Reducing the number of parameters, which eventually help to prevent overfitting.

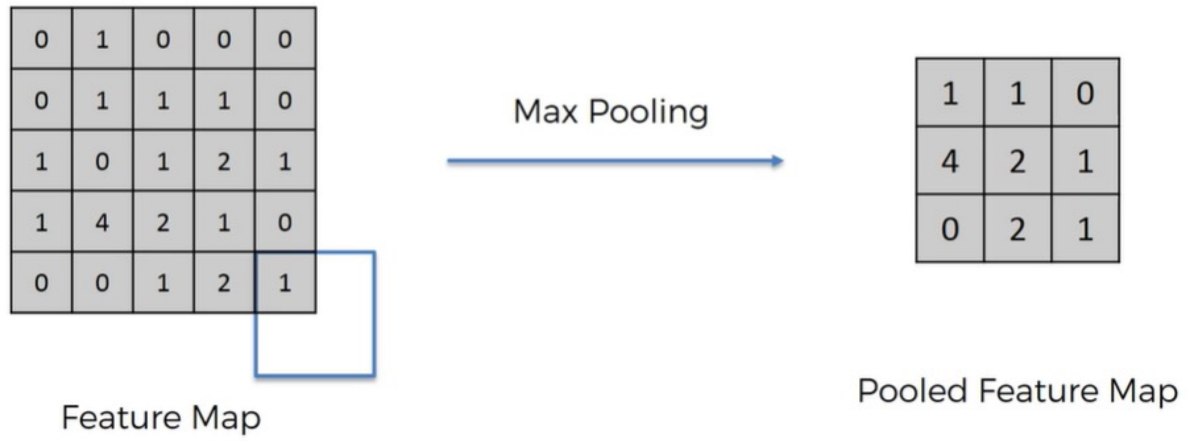
Step 2: Max Pooling

- Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.



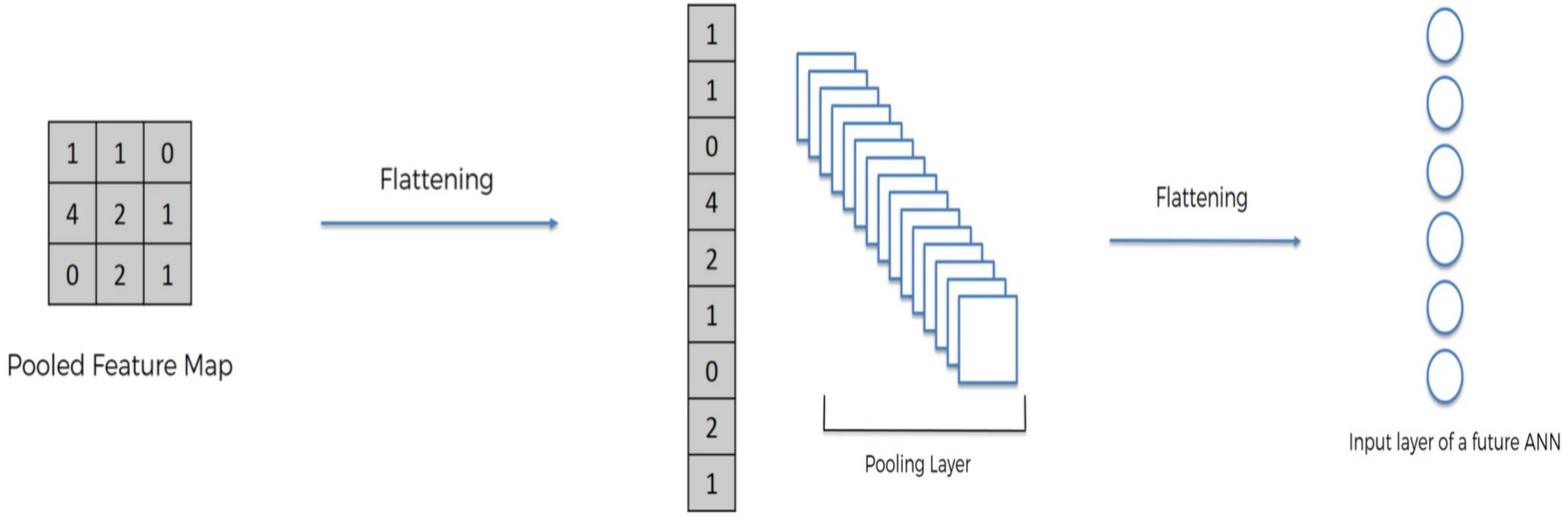
Hyperparameters for pooling layer:
1. Filter size
2. Stride
3. Max or average pooling





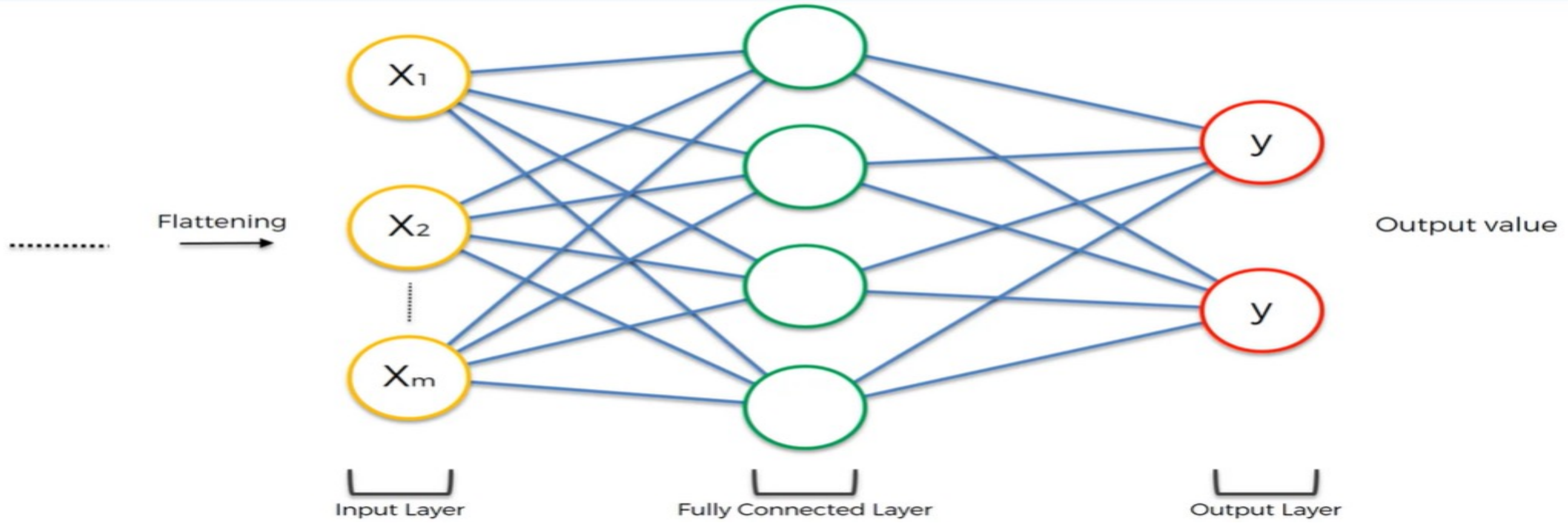
Step 3: Flattening

We need to flatten the matrix into vector, so that they can be passed as input of fully connected layer (like a Neural Network) for further processing.



Step 4: Full Connection / Fully Connected layer

- Like hidden layer in NN, in CNN we have fully connected layer (more specific type of hidden layer) .
- **In a fully connected layer**, each neuron receives input from every element **of** the previous **layer**. **In a convolutional layer**, neurons receive input from only a restricted subarea **of** the previous **layer**.
- In CNN weights, weights, filters/feature detectors are updated to reduced the loss function (Cross-entropy) to through backpropagation. We use SoftMax as activation function.



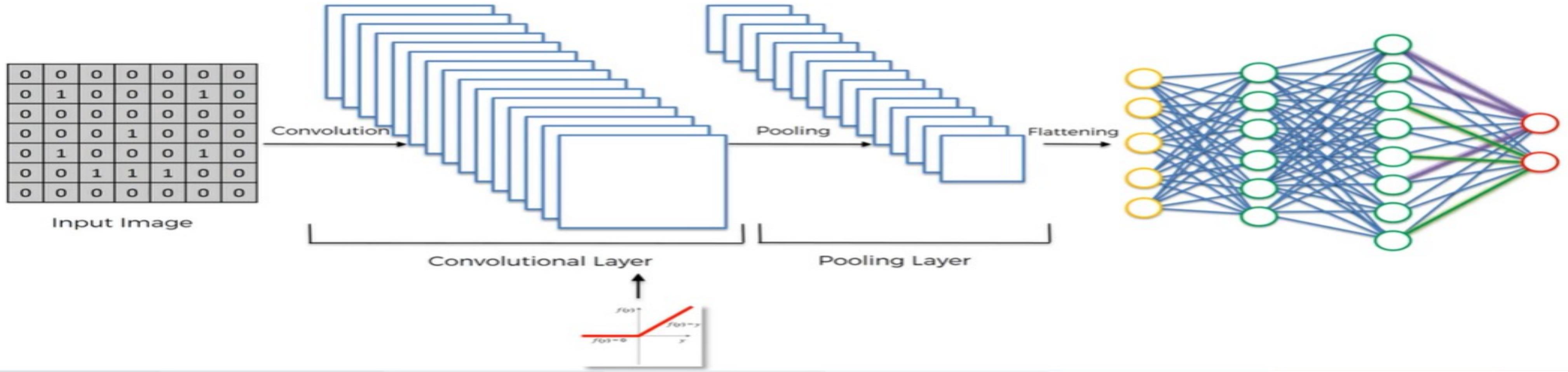
Backpropagation

We covered how an image is classified via forward pass. Next, let us inspect what happens backward. We call this **back propagation**. This is where CNN collects feedback and improves itself:

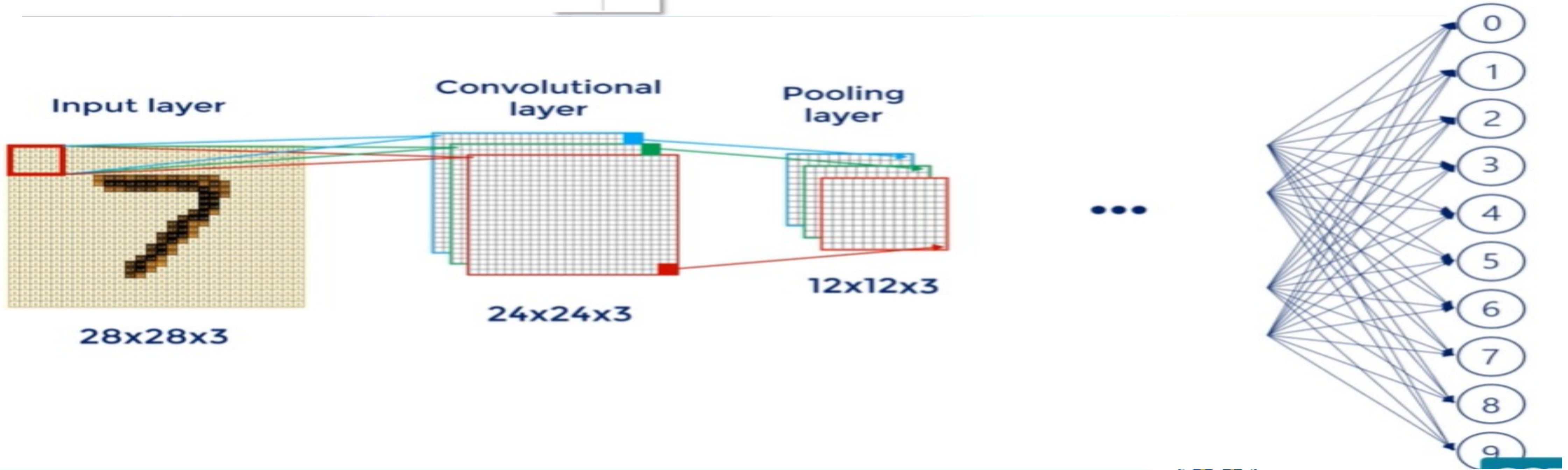
- After prediction, each layer will receive feedback from its preceding layer. Feedback will be in the form of losses incurred at each layer during prediction.
- Aim of the CNN algorithm is to arrive at optimal loss. We call this as local minima.
- Based on the feedback, network will update the weights of kernels.
- This will make the output of convolutions better when next time forward pass happens.
- When the next forward pass happens, loss will come down. Again, we will do back prop, the network will continue to adjust, a loss will further come down and process repeats.
- This forward pass followed by back prop keeps happening the number of times we choose to train our model. We call it **epochs**.

[Backpropagation involves the mathematical calculations including gradient descents]

Summary

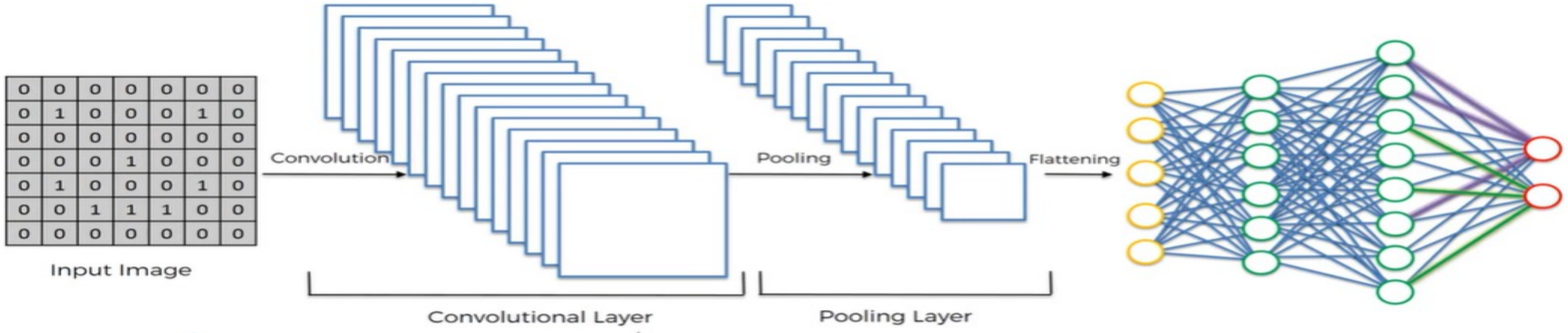


0 0 0 0 0 0 0
0 1 0 0 0 1 0
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
Input Image
Convolution
Pooling
Flattening
Convolutional Layer
Pooling Layer

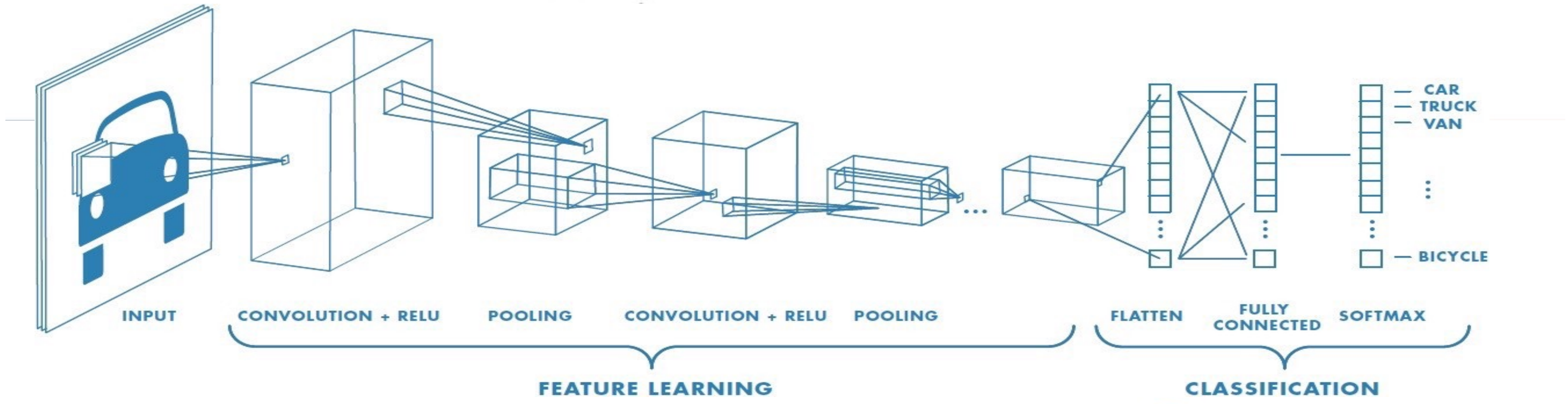


Input layer
Convolutional layer
Pooling layer
28x28x3
24x24x3
12x12x3
0
1
2
3
4
5
6
7
8
9

Summary



0 0 0 0 0 0 0
0 1 0 0 0 1 0
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
Input Image
Convolution
Pooling
Flattening
Convolutional Layer
Pooling Layer



INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
CAR
TRUCK
VAN
BICYCLE
FEATURE LEARNING
CLASSIFICATION

Some Popular CNN Models

- ***Some Common Deep CNN Architectures includes:***
  - ***LeNet-5***
  - **AlexNet**
  - **VGG-16**
  - **Inception-v1**
  - **Inception-v3**
  - **ResNet-50**
  - **Xception**
  - **Inception-v4**
  - **Incption-ResNet**
  - **ResNeXt-50**

For more details : https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

CNN hands-on

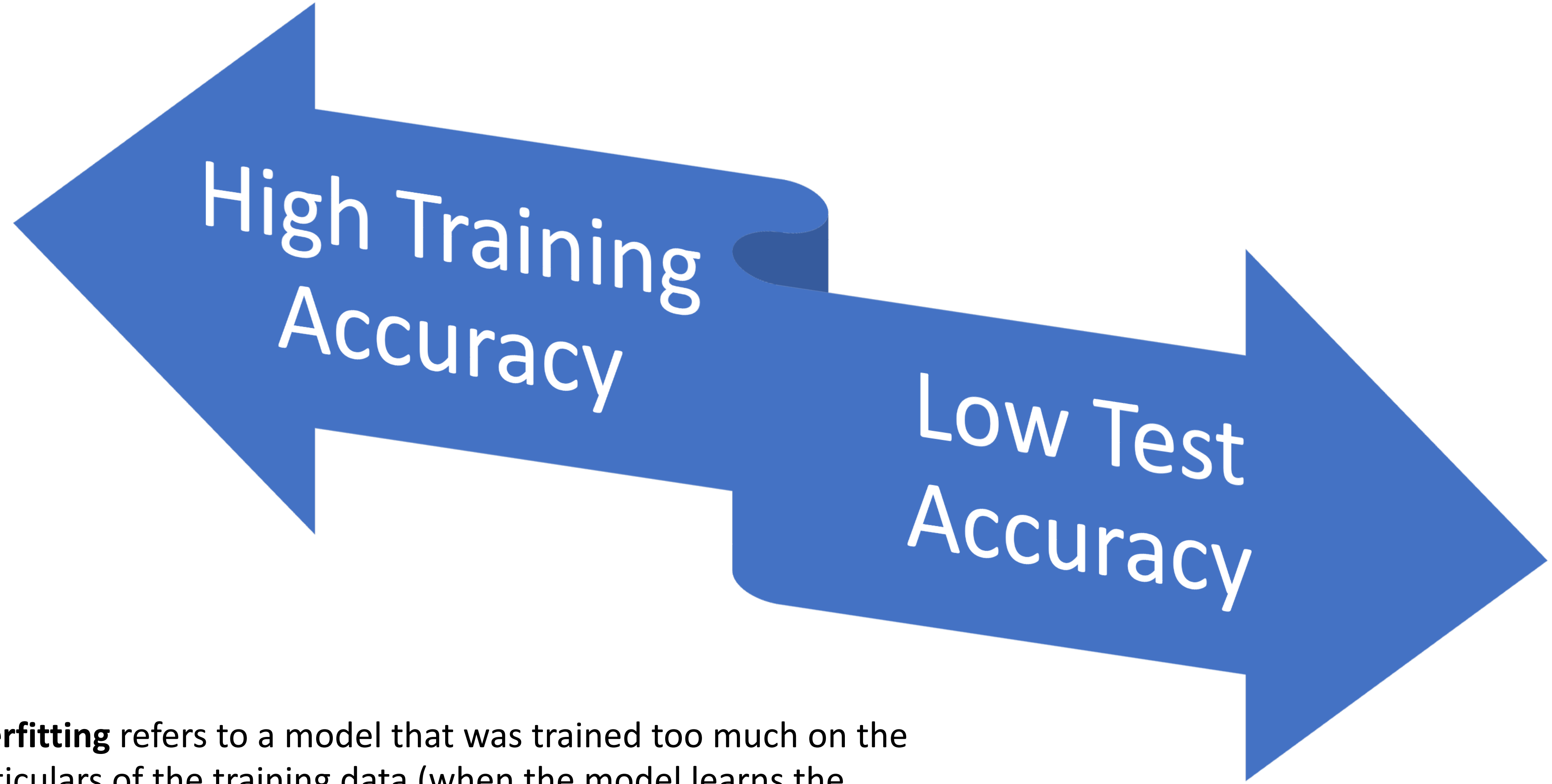
Please get the python codes from the following GitHub repository:

https://github.com/ShaonBhattaShuvo/Deep-Learning-Workshop/blob/master/Workshop%20(Part%202)/WorshopDL_(Part2).py

To visualize how handwritten characters are recognized by CNN:

https://www.cs.ryerson.ca/~aharley/vis/conv/

Concept of Overfitting



Overfitting refers to a model that was trained too much on the particulars of the training data (when the model learns the noise in the dataset).

Concept of Overfitting

Specifically, **Overfitting** occurs if the model or algorithm shows low bias but **high variance**.
Underfitting occurs if the model or algorithm shows **high bias** but low variance.

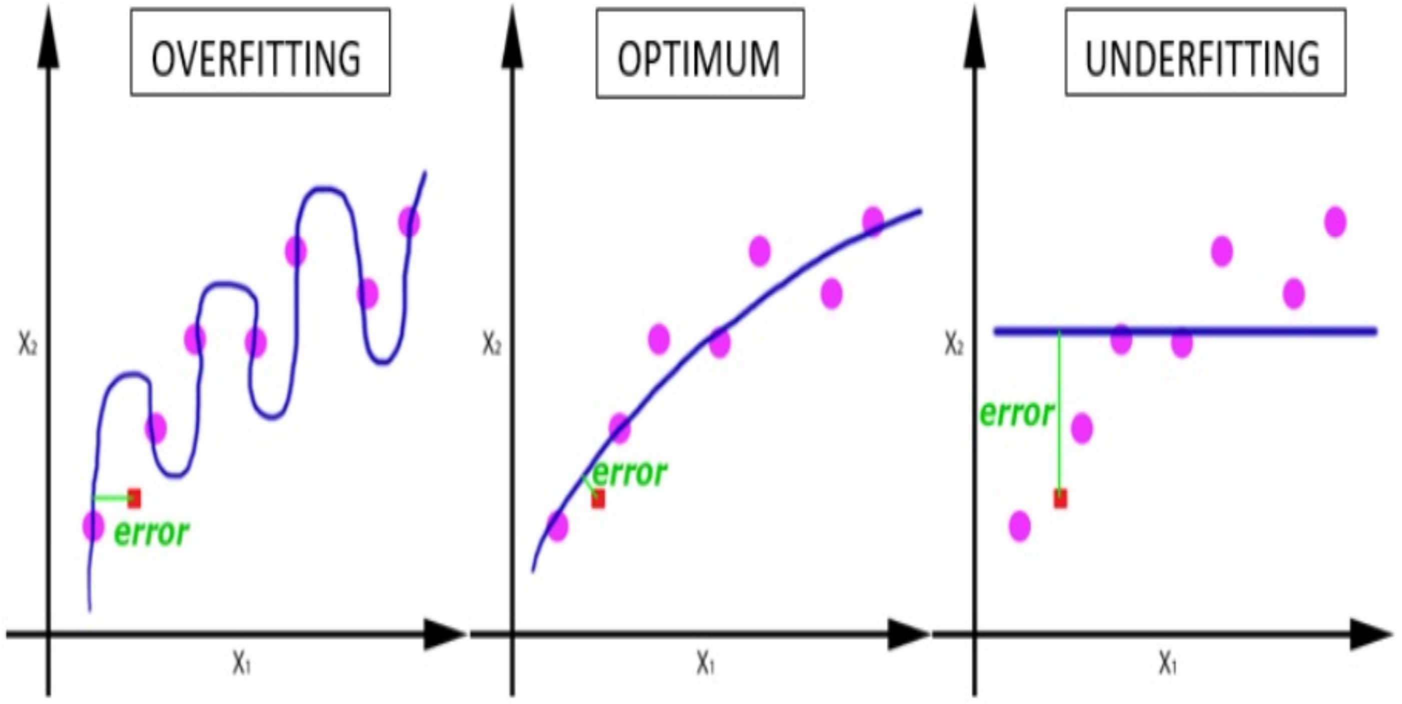


Image source: towardsdatascience.com

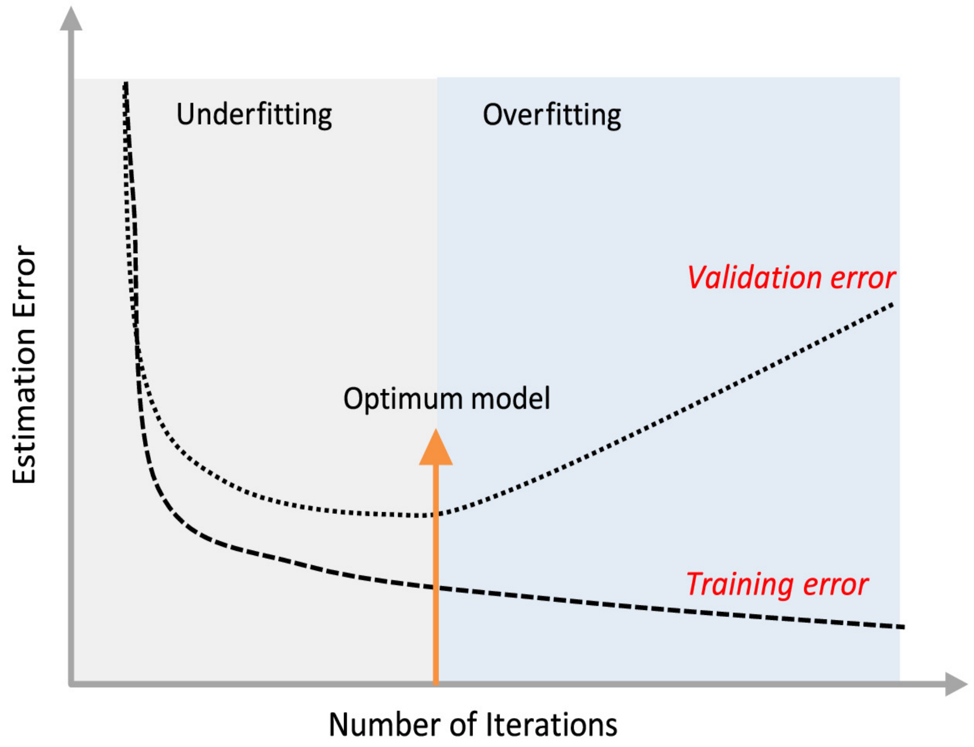
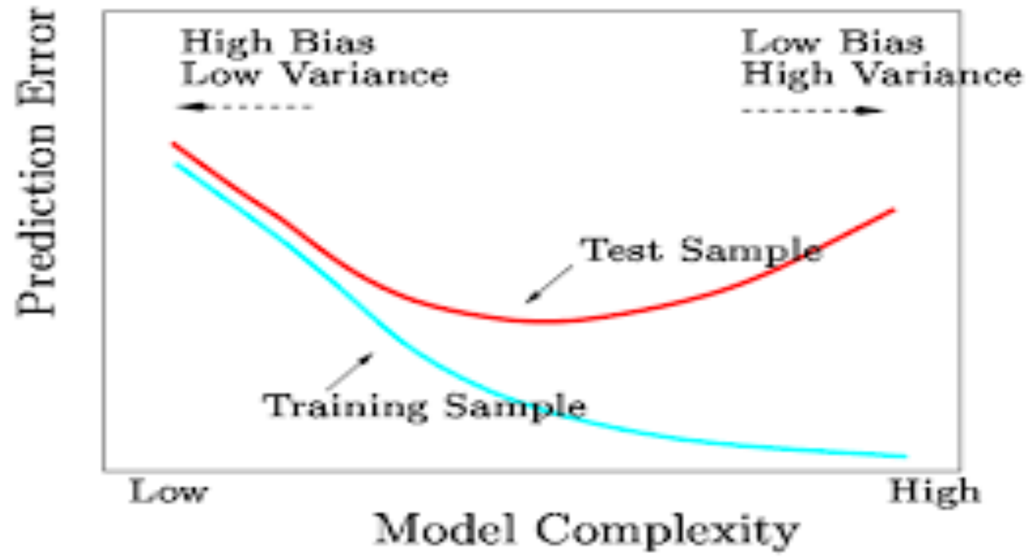


Image source: osapublishing.org

Bias and Variance

High Difference Between Training and Test Accuracy/Error is High Variance (Overfitting)

Low Training Error + High Test Error = High Variance



Low Training Accuracy/ High Training Error is High Bias (Underfitting)

High Training error = High Bias

Reducing Bias?

- Bigger Networks (More Nodes/Deep)
- Training Longer (More Iteration)

Reducing Variance?

- More Data
- **Regularization**

Regularizations: reduces the overfitting.

- L1 & L2 Regularization (Weight Decay)
  - These update the general cost function by adding another term known as the regularization term (slightly less than 0). This prevents the weights from growing too large.
  - *Cost function = Loss (say, binary cross entropy) + Regularization term.*
  - However, this regularization term differs in L1 and L2.

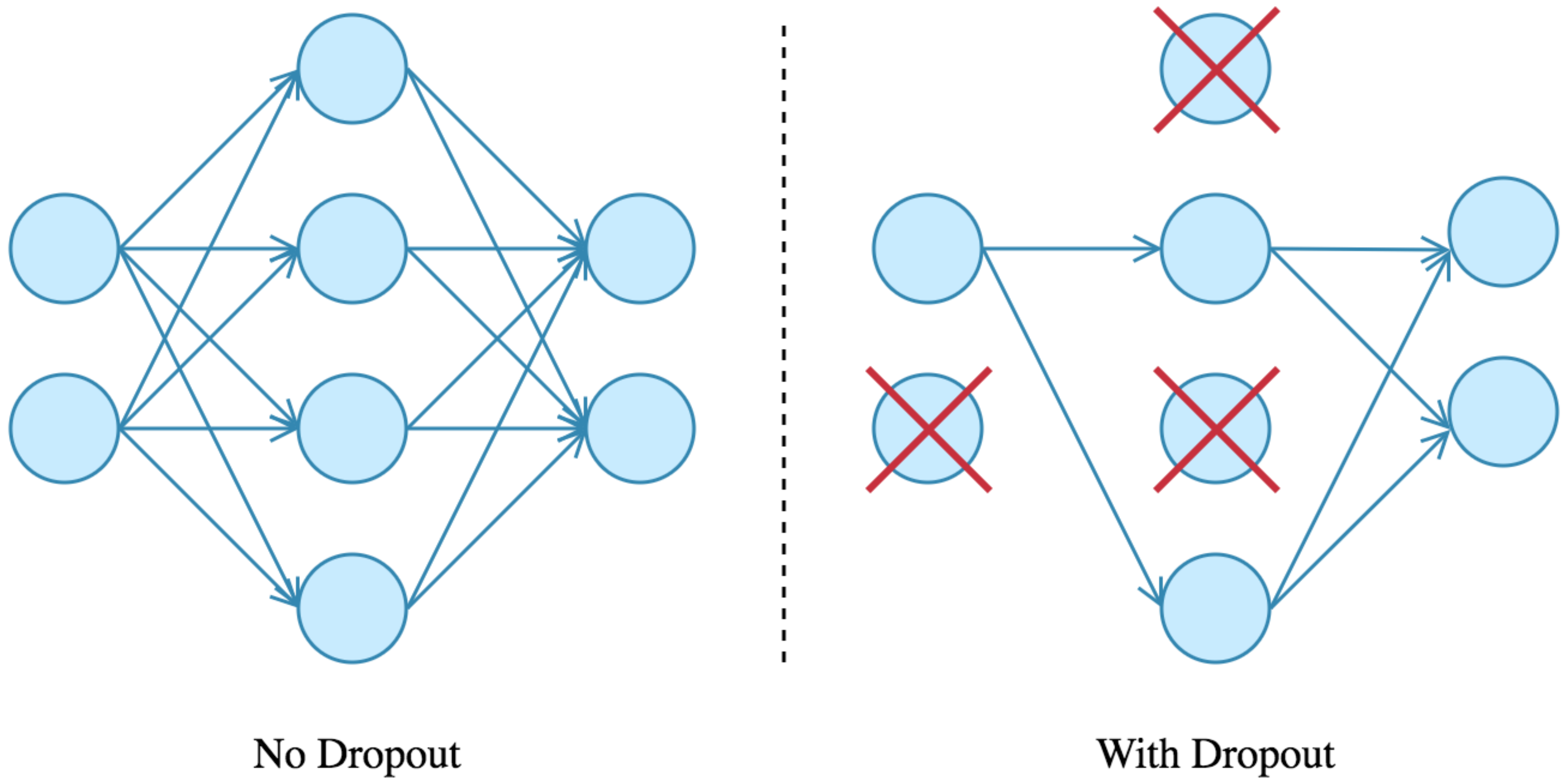
In L2, $Cost\,function = Loss + \frac{\lambda}{2m} * \sum \|w\|^2$ In L1, $Cost\,function = Loss + \frac{\lambda}{2m} * \sum \|w\|$

Here, **lambda** is the regularization parameter. It is the hyperparameter whose value is optimized for better results.
L2-Norm is also known as Square Norm or Euclidian Distance (in NN it is actually Frobenius Norm.)
L1 Norm is also known as Manhattan Distance or Taxicab Norm.

- Dropout
- Data Augmentation
- Early Stopping

[We will see details and also implement Dropout, Data Augmentation and Early Stopping in our Next Hands-on Experiment]

Dropout



Data Augmentation

- Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data.

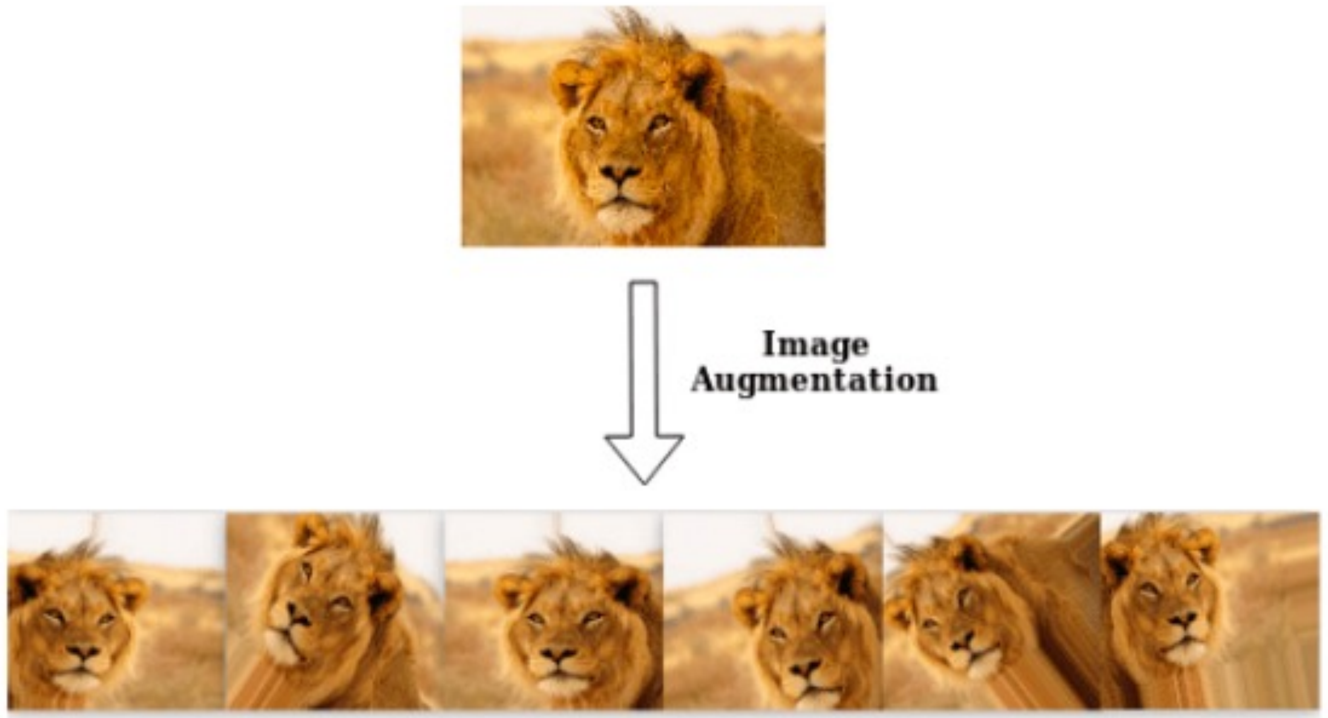


Image Source: towardsdatascience.com

Early Stopping

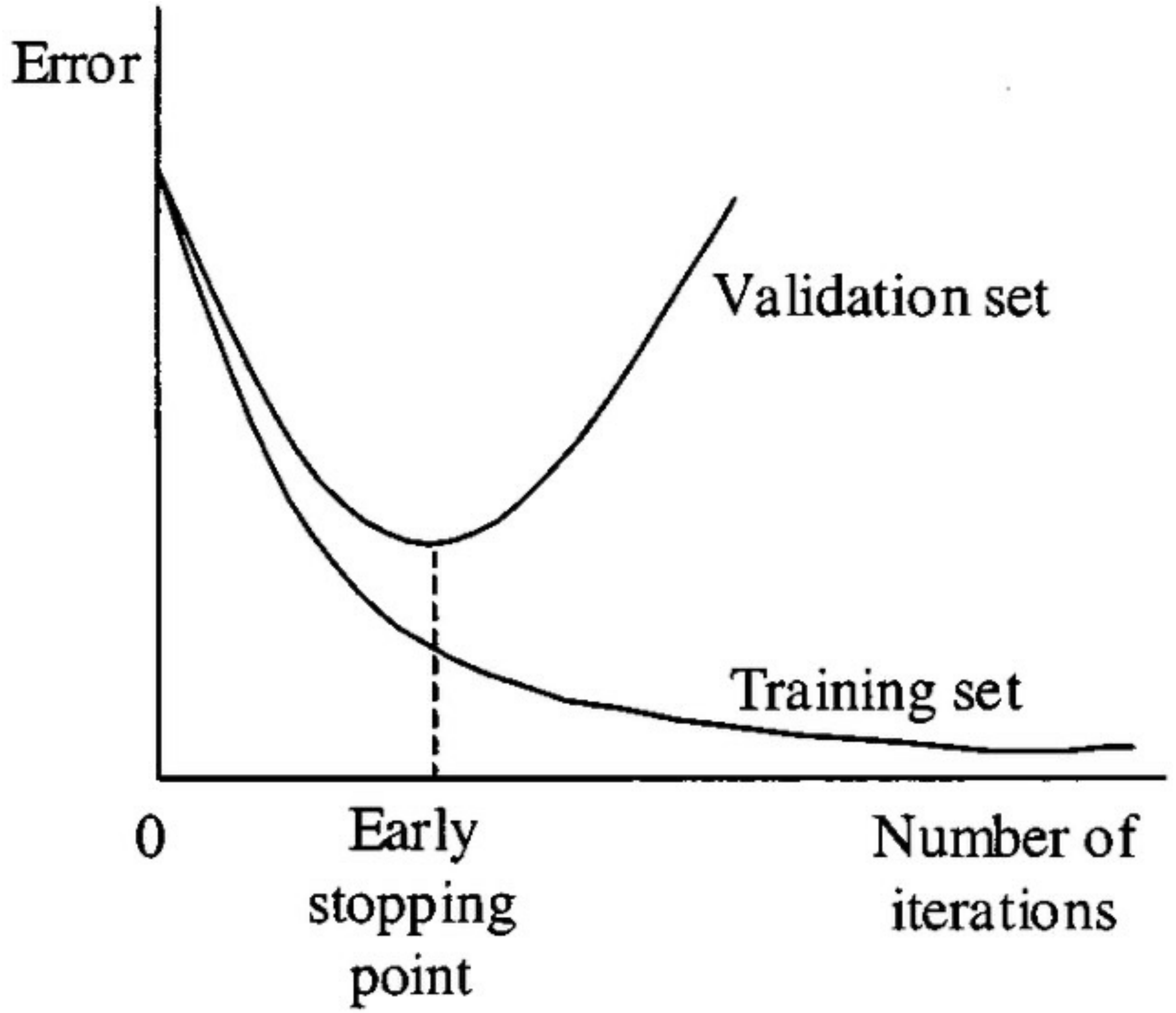
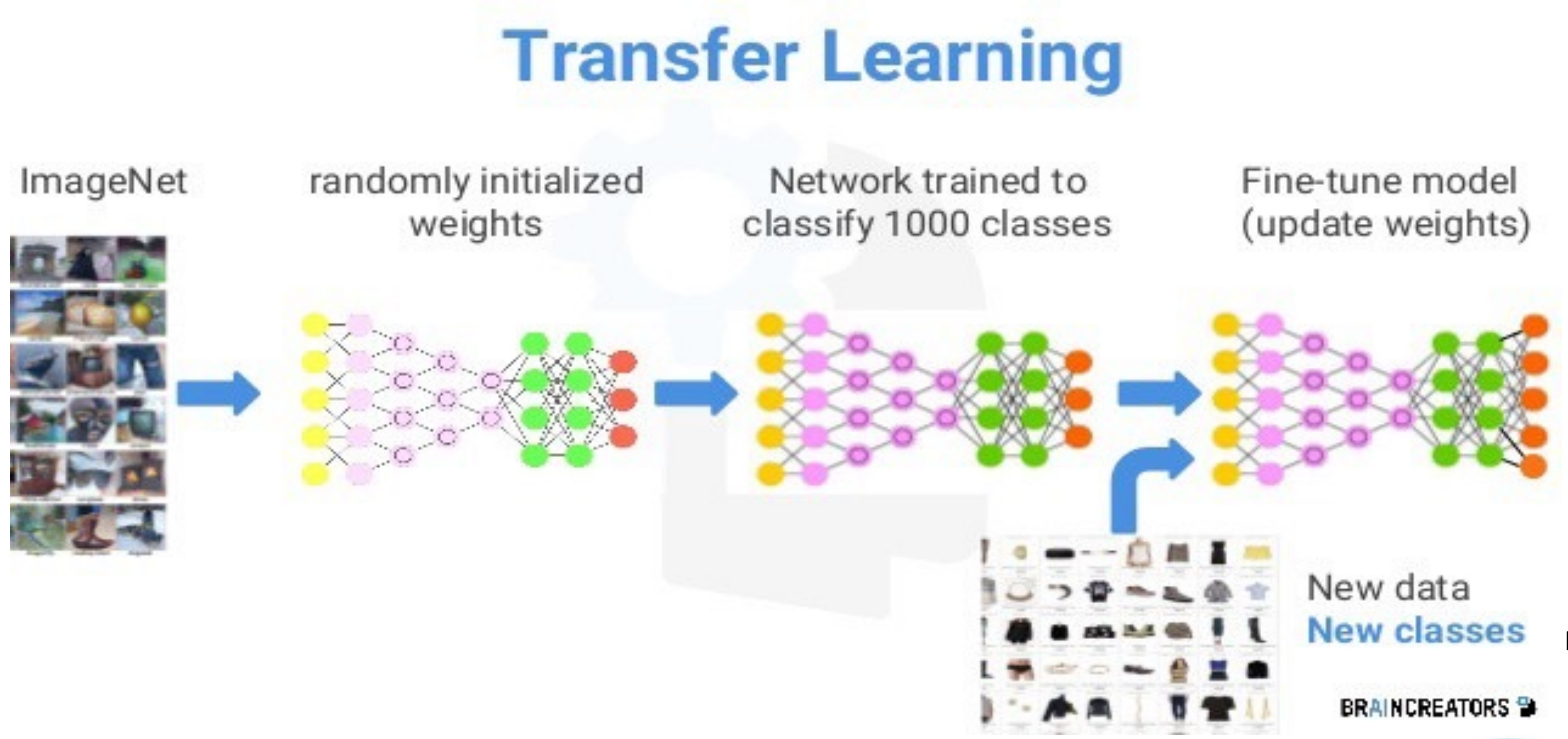


Image Source: towardsdatascience.com

Transfer Learning

- ❑Storing knowledge gained while solving one problem and applying it to a different but related problem.


Transfer Learning
ImageNet
randomly initialized weights
Network trained to classify 1000 classes
Fine-tune model (update weights)
New data
New classes
BRAINCREATORS

How to use?

-Using a pretrained model.

When to use?

-When dataset is too small to train.

Image Source: mc.ai

InceptionV3: Pretrained Model

InceptionV3 Model:

- InceptionV3 is a culmination of many ideas developed by multiple researchers over the years.
- The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, dropouts, and fully connected layers.
- The model is available in keras.application library with weights pre-trained on ImageNet.
  [The complete documentation on Inception V3 model is available at : https://keras.io/applications/]

ImageNet Dataset:

- The ImageNet has over ten million URLs of labeled images.

Transfer learning hands on

Please get the and dataset python codes from the following GitHub repository:

https://github.com/ShaonBhattaShuvo/Deep-Learning-Workshop/tree/master/Workshop%20(Part%202)

Challenges?

- Image quality is not very good
- Less number of images (small dataset)

How to learn deep learning more efficiently?

- Learn basic Machine Learning techniques.
- Ask the question why?
- Pick a project and start implementing.
- Learn from different sources.
- Mathematical Knowledge:
  - Probability and Statistics.
  - Linear algebra.
  - Calculus.
- Keep Patience

Some useful contents came in handy to prepare the lecture:

1. https://theappsolutions.com/blog/development/convolutional-neural-networks/
2. https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/
3. https://en.wikipedia.org/wiki/Convolutional_neural_network
4. https://cs.nju.edu.cn/wujx/paper/CNN.pdf
5. https://analyticsindiamag.com/deep-learning-image-classification-with-cnn-an-overview/
6. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
7. https://docs.gimp.org/2.6/en/plug-in-convmatrix.html
8. https://www.coursera.org/specializations/deep-learning#courses
9. https://www.udemy.com/course/artificial-intelligence-az/
10. https://www.udemy.com/course/machinelearning/
11. https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d
12. https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/
13. https://www.osapublishing.org/jocn/fulltext.cfm?uri=jocn-10-10-D126&id=396379
14. https://machinelearningmastery.com/transfer-learning-for-deep-learning/
15. https://www.cs.ryerson.ca/~aharley/vis/conv/