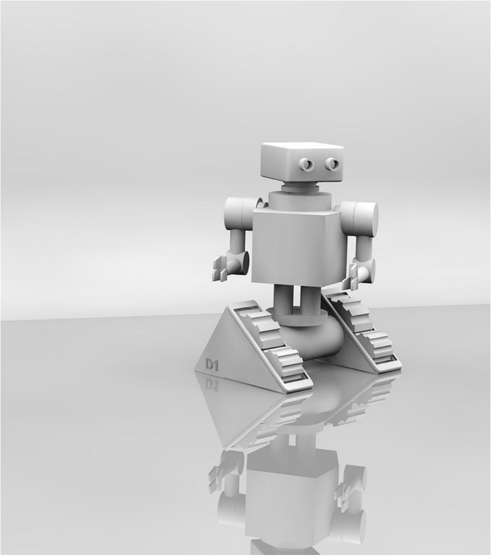


CPSC 481
Artificial Intelligence

Dr. Mira Kim
Mira.kim@fullerton.edu

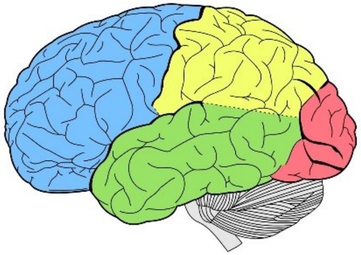
What we will cover this week

- Artificial Neural Network (ANN)
- Convolutional Neural Network (CNN)

BIOLOGICAL NEURAL NETWORK

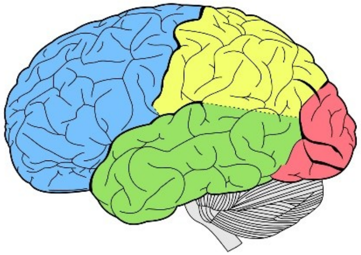
Brain and Machine

- Brain
  - Pattern Recognition
  - Association
  - Complexity
  - Noise Tolerance
- Computation
  - Many, Slow, Unreliable Processors acting in Parallel



- Machine
  - Calculation
  - Precision
  - Logic
- Von Neumann architecture uses a Single Processing Unit.
  - Tens of Millions of Operations per second
  - Absolute Arithmetic Precision

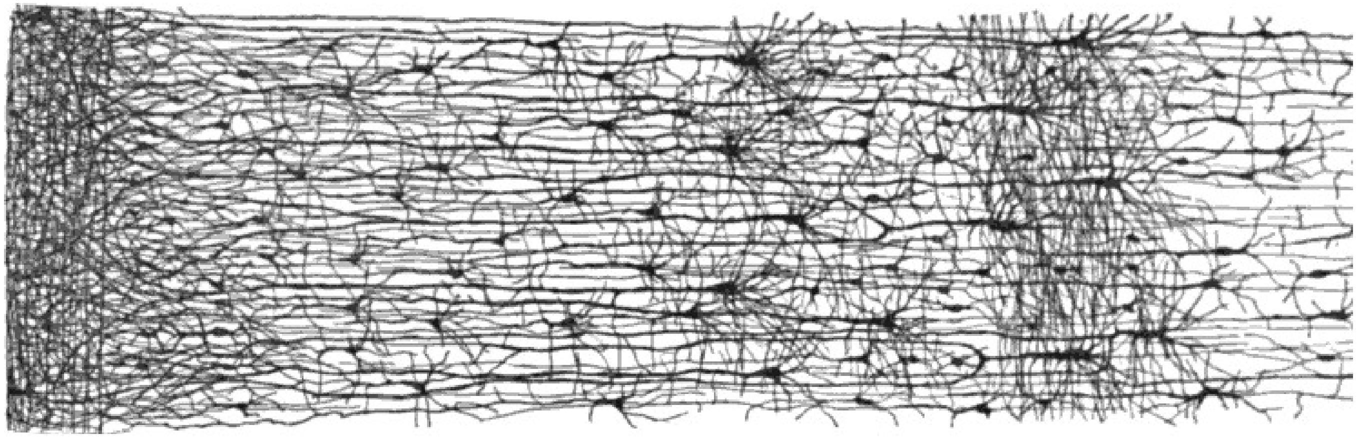
Features of Brain



- # of Neurons: ~ 10^{10} (10 Billion)
- Connections per Neuron: about 7,000
- Neuron Switching Time: ~ 0.001 Second
  - On average, several thousand connections.
- Compensates for Complex Problems Massive Parallelism
- Scene Recognition Time: ~ 0.1 second
- Die Off Frequently, Never Renewed
  - 2 special places in brain can give birth to new neurons.

Biological Neural Networks (BNN)

- Group of biological neurons interconnected by the ends of the neuron
- Multi Layer of Neural Networks, i.e., Neural Circuit



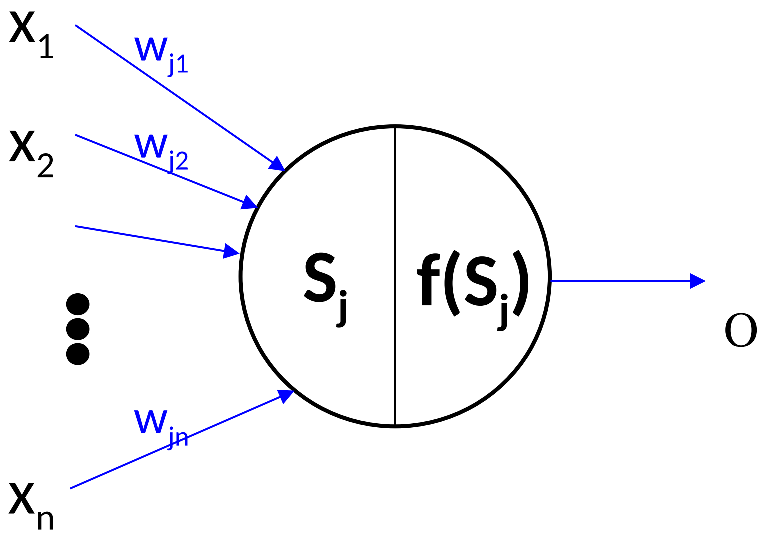
ARTIFICIAL NEURON

Artificial Neuron

- Definition
  - Mathematical Function based on Concept of Biological Neuron
  - Artificial Unit of activating computation with input
- Characteristics
  - It uses activation functions to activate neuron.
    - If the result of a function is high enough, the neuron becomes activated.
  - The type of output in neuron is numeric.

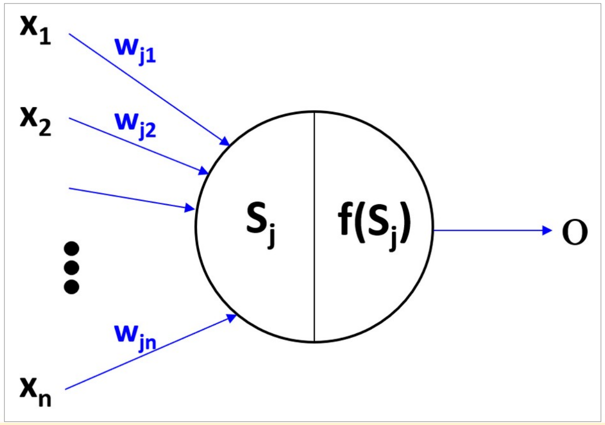
Artificial Neuron

- Called Perceptron
- Input
  - Composed of artificial inputs
- Output
  - Producing an output



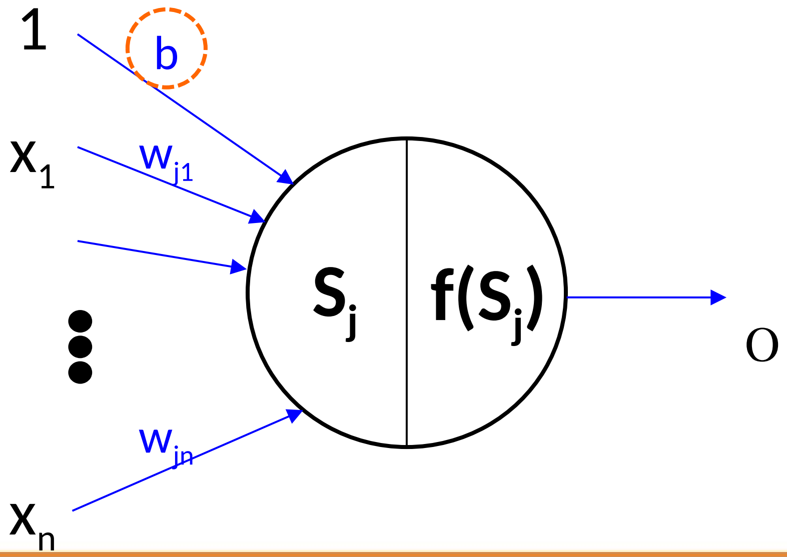
Artificial Neuron

- Each neuron has weighted inputs from other neurons
- Each neuron has a threshold value.
- If the activation level exceeds the threshold, the neuron fires.
  - The unit performs a weighted sum of its inputs, subtracts its threshold value, and gives its activation level.
    - Linear Threshold



Artificial Neuron

- Weight, w_i
  - Strength of the connection for a neuron
  - Can be updated during training neural network.
  - Backpropagation
    - To update the weight using value of errors (or loss) in each training step.
- Bias, b
  - Weight from Bias Neuron
  - Can be updated by backpropagation.



Example) Possibility of rain

- Predicting the possibility of rain based on the barometric pressure.
- We have a dataset that includes days with the barometric pressure and whether it rained or not on those days.

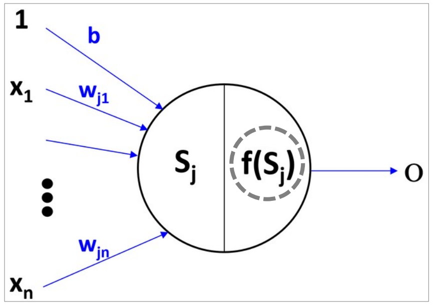
- **Input Layer -** Input x: Barometric pressure reading
- **Hidden Layer:**
  - Weight w: Weight assigned to our input x. It determines how much influence the barometric pressure will have on the prediction.
    - Example: If w=0.8, and our barometric pressure is x=0.5, the product is $0.8*0.5=0.4$.
  - Bias b: An offset or intercept added to the weighted input before it is passed through the activation function.
    - Example: If the bias b=0.2, it is added to the weighted input to give $0.4+0.2=0.6$.
- **Output Layer:**
  - The final activation that decides if it will rain or not, let’s say using a sigmoid function to output a probability.

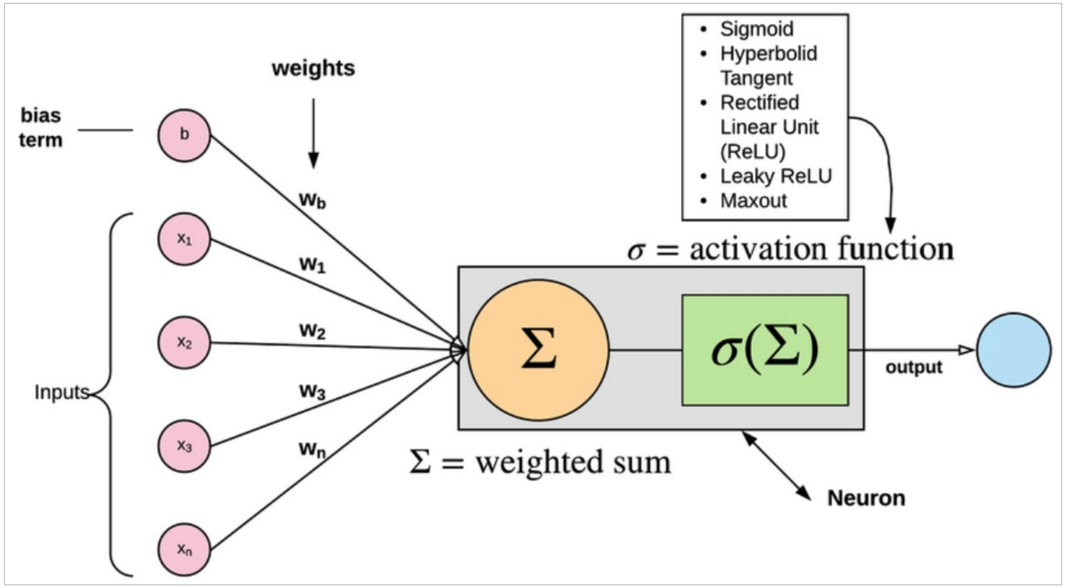
Example) Possibility of rain

- **Output = *f* (*w*·*x*+*b*)**, where:
  - *f* is the activation function, such as a sigmoid function.
  - *w*·*x* is the weighted input.
  - *b* is the bias.
- **Initial Scenario:**
  - initially, *w*=0.5 and *b*=0
  - The input barometric pressure *x*=0.8. The weighted sum is 0.5∗0.8=0.4
  - Since the bias is 0, the total input to the activation function is 0.4
  - After passing through an activation function, suppose our output is 0.6, indicating a 60% chance of rain.
- **Learning Process:**
  - Through training, we discover that the prediction is not accurate. The network should have predicted a higher chance of rain based on the data.
  - Backpropagation will calculate the gradient of the loss (error) and update *w* and *b* accordingly.
  - Let’s say after one round of backpropagation, *w* is updated to 0.6 and *b* to 0.1
- **Updated Scenario:**
  - Now, the weighted sum is 0.6∗0.8=0.48
  - Adding the bias, we get 0.48+0.1=0.58
  - The activation function might now output 0.641, indicating a 64.1% chance of rain, which is closer to the true pattern in the data.

Artificial Neuron

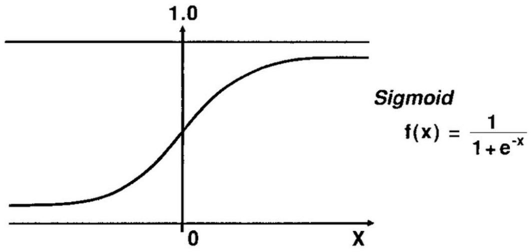
- Activation Function
  - It defines an output of a neuron for the given inputs.
  - The output judges whether activating the neuron or not.
    - Value of 0 does not activate the neuron.



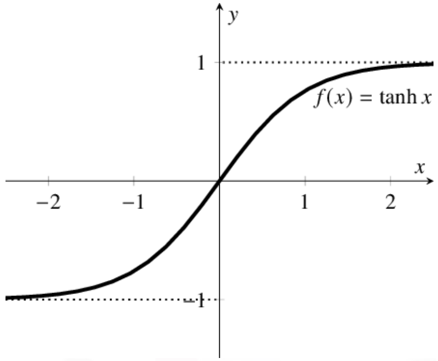


Artificial Neuron

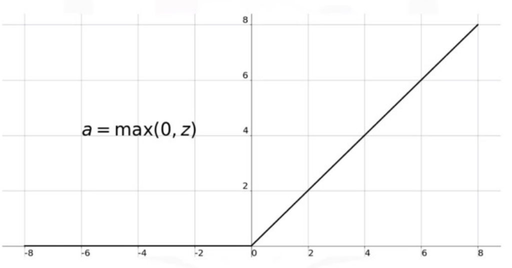
- Representative Activation Functions
  - Sigmoid Function
    - Range of Value: [0, 1]



  - Hyperbolic Tangent
    - Range of Value: [-1, 1]

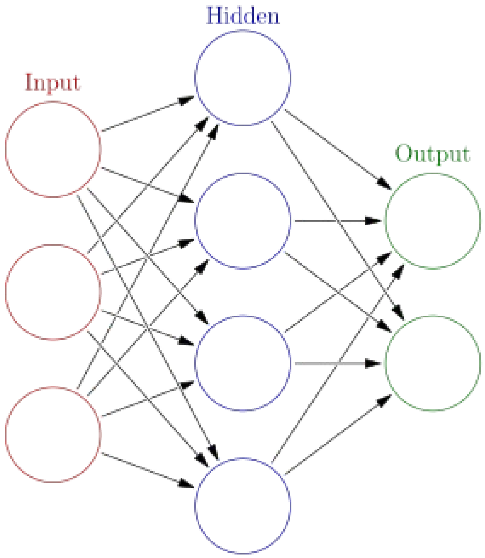


  - Relu (Rectified Linear Unit)
    - Range of Value: [0, ∞]



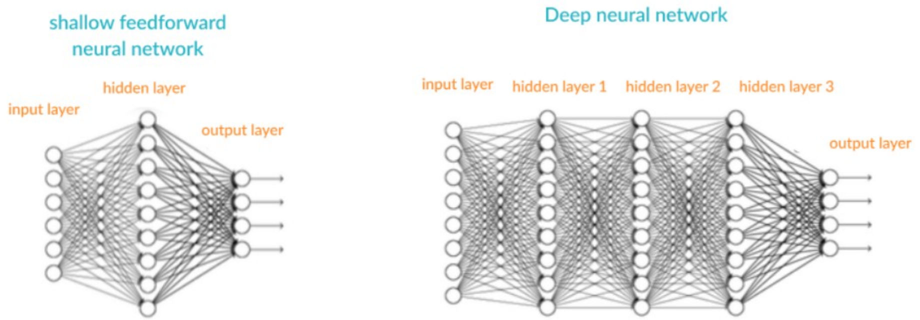
Artificial Neural Network (ANN)

- ANN, also called Neural Network
  - A supervised learning system computing system consisting of artificial neurons, that simulates the biological neural network.
  - Each neuron makes a simple decision and feeds the decisions to other neurons.
- ANN consists of Layers.
  - Input Layer
    - Accepts the inputs of the model.
  - Hidden Layer
    - The layers between input layer and output layer
  - Output Layer
    - Generates predictions.



Deep Neural Network (DNN)

- DNN
  - – Has 2 or more hidden layers that process inputs.



- Benefits
  - – DNN are more accurate than Shallow Neural Networks.
  - – Can improve the accuracy with more hidden layers.
  - – Additional layers are useful up to a limit of 9-10.
    - Beyond that, their predictive power starts to decline.

Applications of ANN

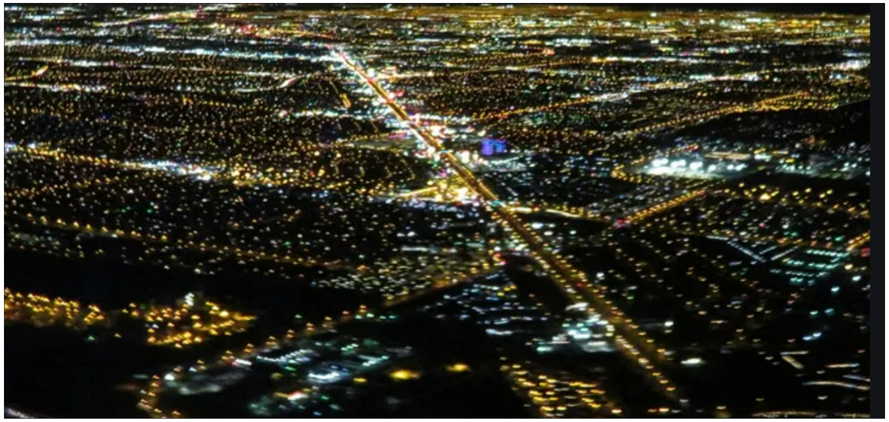
Commonly used in supervised learning where the model is trained on a labeled dataset.

1. **Image and Visual Recognition** - Used for facial recognition, image classification, and object detection in images and videos.
2. **Natural Language Processing (NLP)** – Used for machine translation, sentiment analysis, text summarization, and chatbots.
3. **Medical Diagnosis -** Can process medical images for diagnostic purposes, identify patterns in patient data to predict diseases, and assist in personalized medicine.
4. **Autonomous Vehicles** – Can process inputs from various sensors to perceive the environment and make driving decisions in self-driving cars.
5. **Gaming and Entertainment** - Used in the development of AI in games for non-player character (NPC) behavior, and game outcome prediction.
6. **Weather Forecasting** - Analyze datasets to improve the accuracy of weather predictions
7. **Manufacturing and Production** - Used for predictive maintenance, quality control, supply chain optimization, and manufacturing process control.

CONVOLUTIONAL NEURAL NETWORK (CNN)

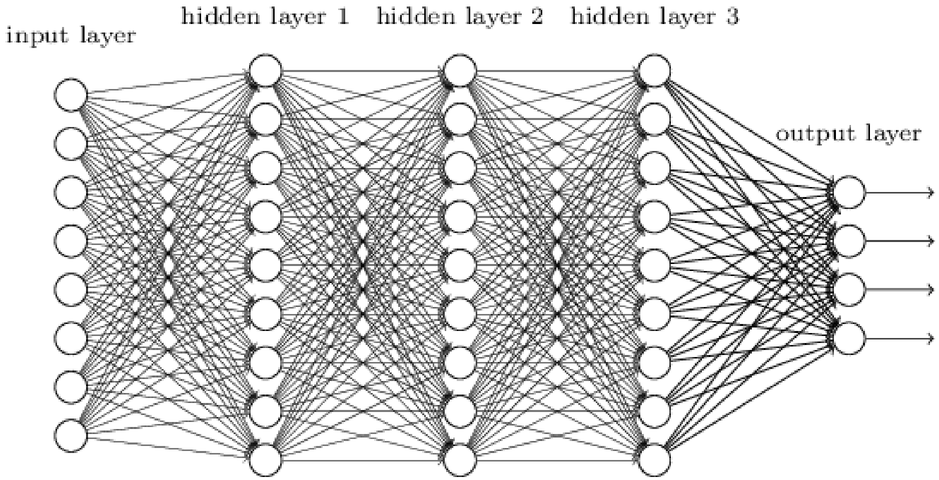
Convolutional Neural Network (CNN)

- Convolutional Neural Network (CNN)
  - A type of Neural Network used to make predictions on spatial information of input data.
- Spatial Information
  - Information the Topological, Geometric, or Geographic properties



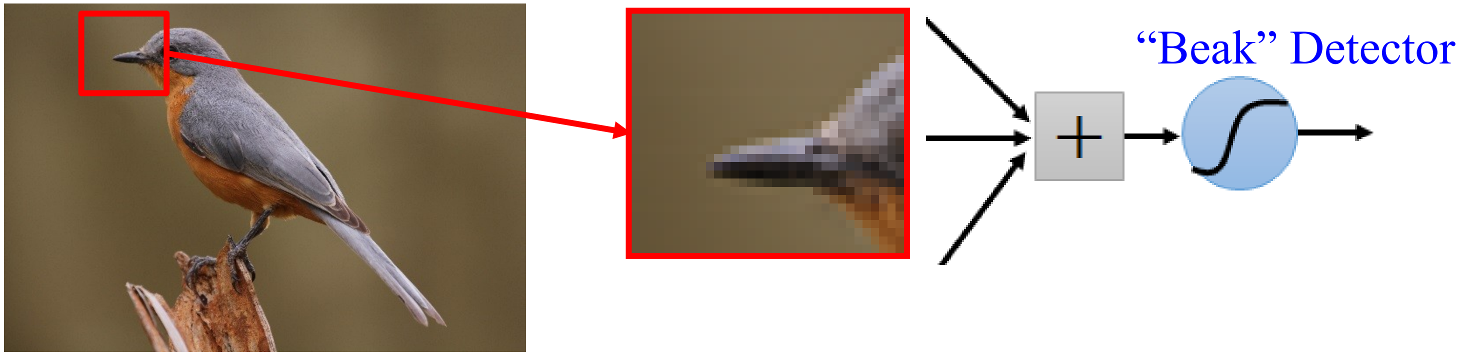
Motivation Towards CNN

- Small Neural Network Model is good for;
  - Training Efficiency
  - Prediction Efficiency
- Do we really need all the edges of a Fully Connected NN Model?
  - For some application problems, we may not need to use all the edges.



Motivation Towards CNN

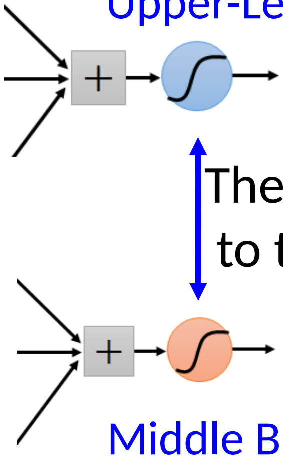
- Consider the problem of ‘Learning an Image’
  - Some patterns in an image are much smaller than the whole image.
  - Some patterns are useful and sufficient enough to recognize an image.
    - Beak for Birds



  - A small region is represented with fewer parameters.

Motivation Towards CNN

- Observation on Images
  - A same pattern could appear in different places in the image.
  - They can be compressed!



Upper-Left Beak Detector

They can be compressed to the same parameters.

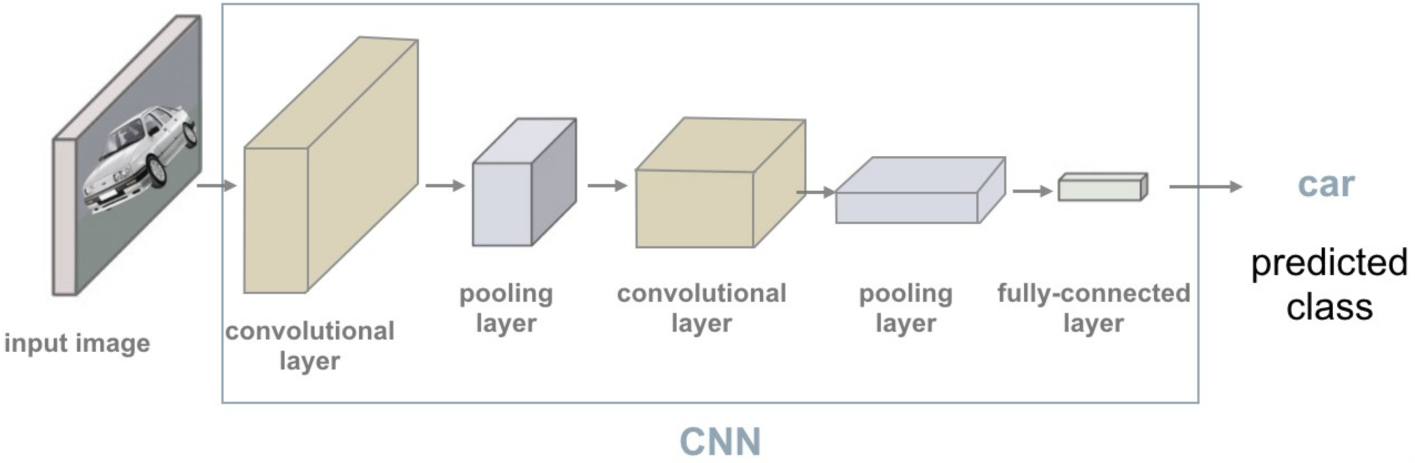
Middle Beak Detector

Motivation Towards CNN

- What about training a lot of such “small” detectors?
- What about letting each detector move around the image?
- Convolutional Neural Network (CNN)
  - To apply the two observations

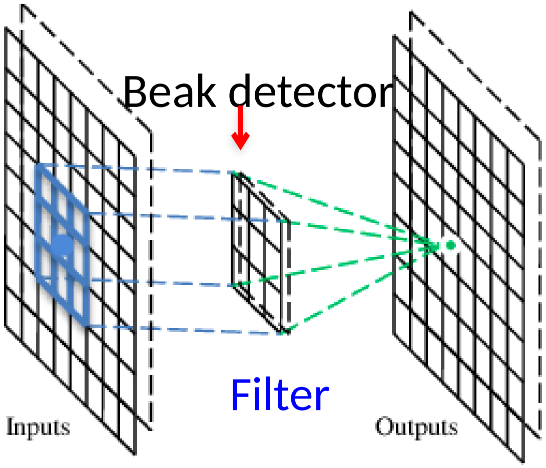
Convolutional Neural Network (CNN)

- Layers in CNN
  - Convolutional Layer - Applies filters to the input to create feature maps, highlighting important visual features
  - Pooling Layer - Reduces the spatial dimensions (height and width) of the feature maps
  - Flattening Layer - Converts the 2D feature maps into a 1D vector
  - Fully Connected Layer - Receives the flattened vector and perform classification



Convolutional Layer

- Layer performing Convolutional Operation
- What is Convolution?
  - A convolution is a linear operation that involves the multiplication of a set of weights with the input, much like a traditional neural network
  - A convolutional layer applies a convolution operation to the input and passes the result to the next layer.
  - A convolution is to convert all the pixels in its Receptive Field into a single value.
- Filter (kernel)



Input Image and Filters

6 x 6 image

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

Each filter detects a small pattern (3 x 3).

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

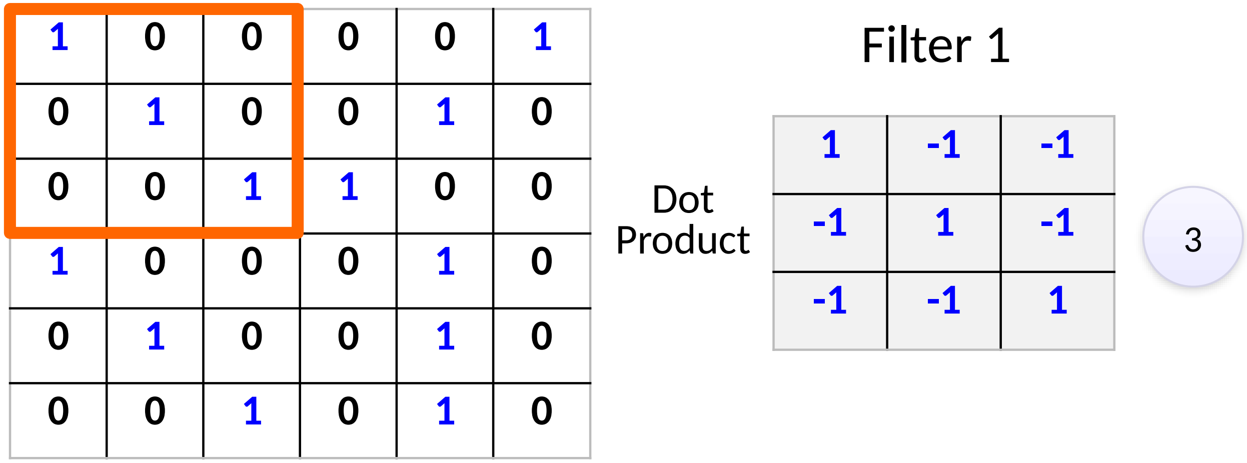
-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮

Applying Filter 1

Stride=1

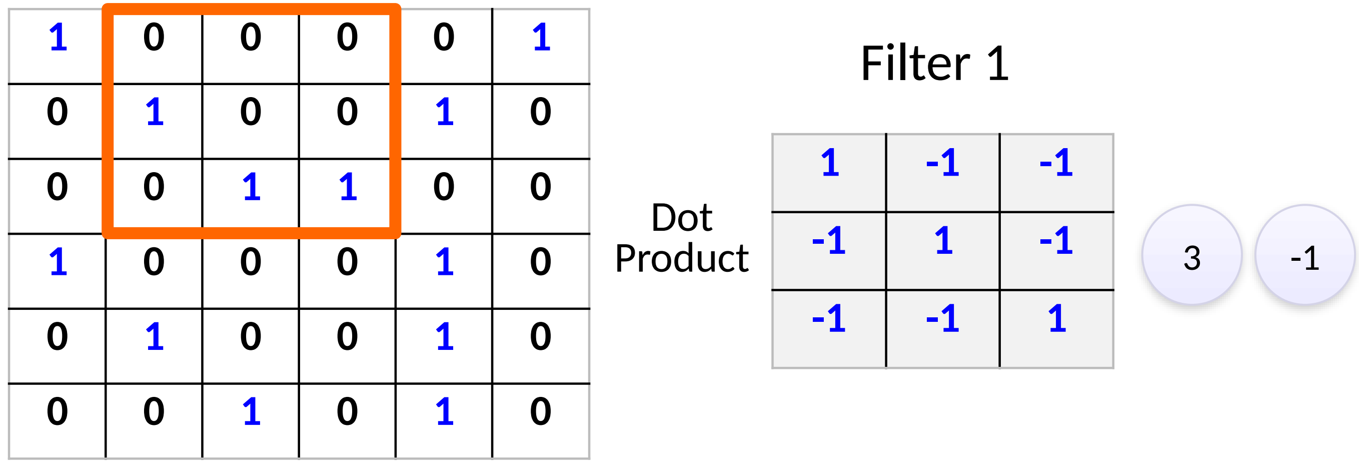


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Dot Product

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1



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

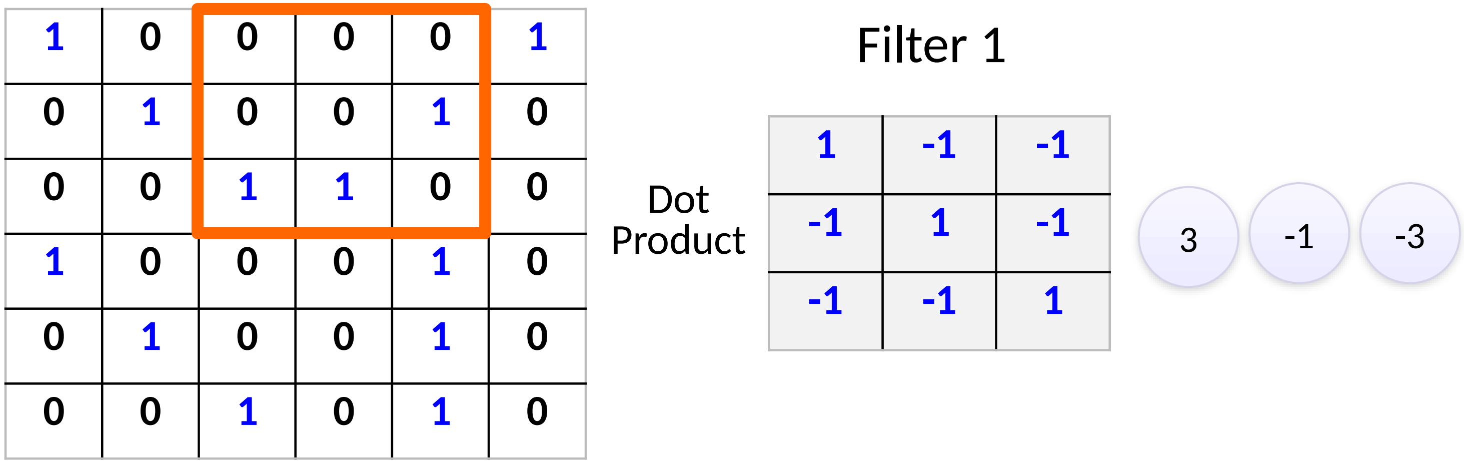
Dot Product

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

Applying Filter 1

Stride=1

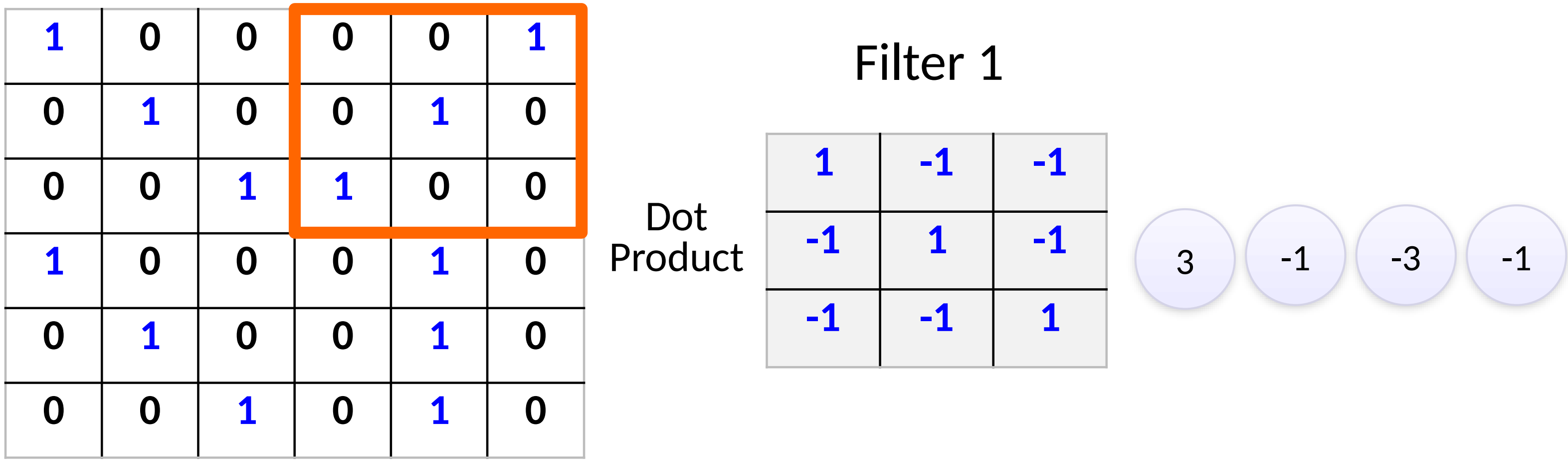


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Dot Product

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1



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

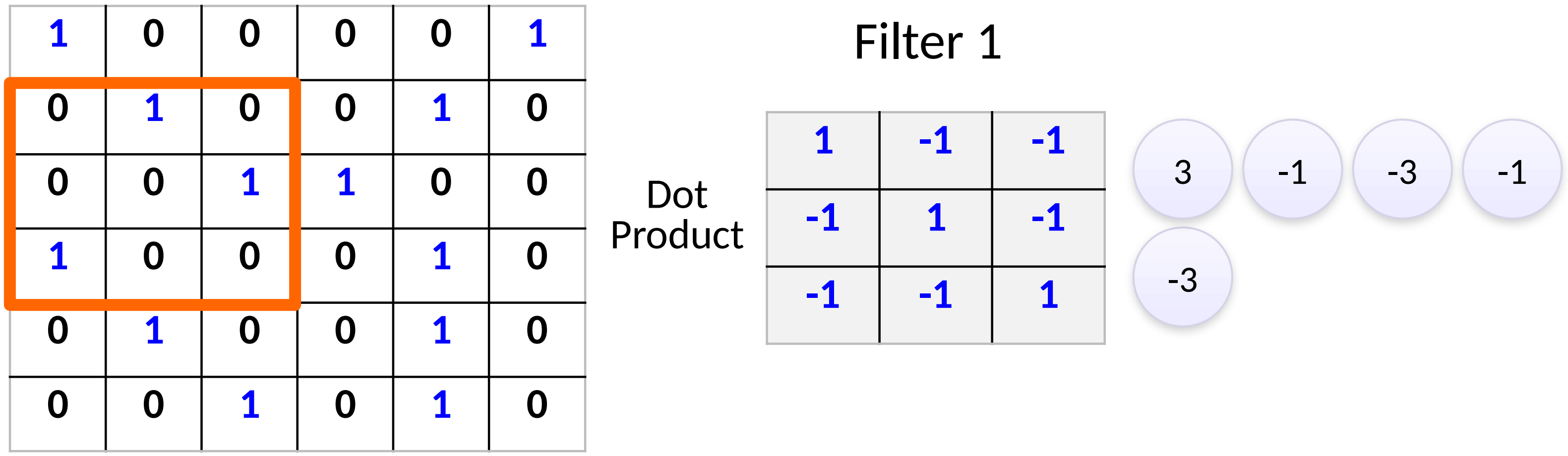
Dot Product

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

Applying Filter 1

Stride=1

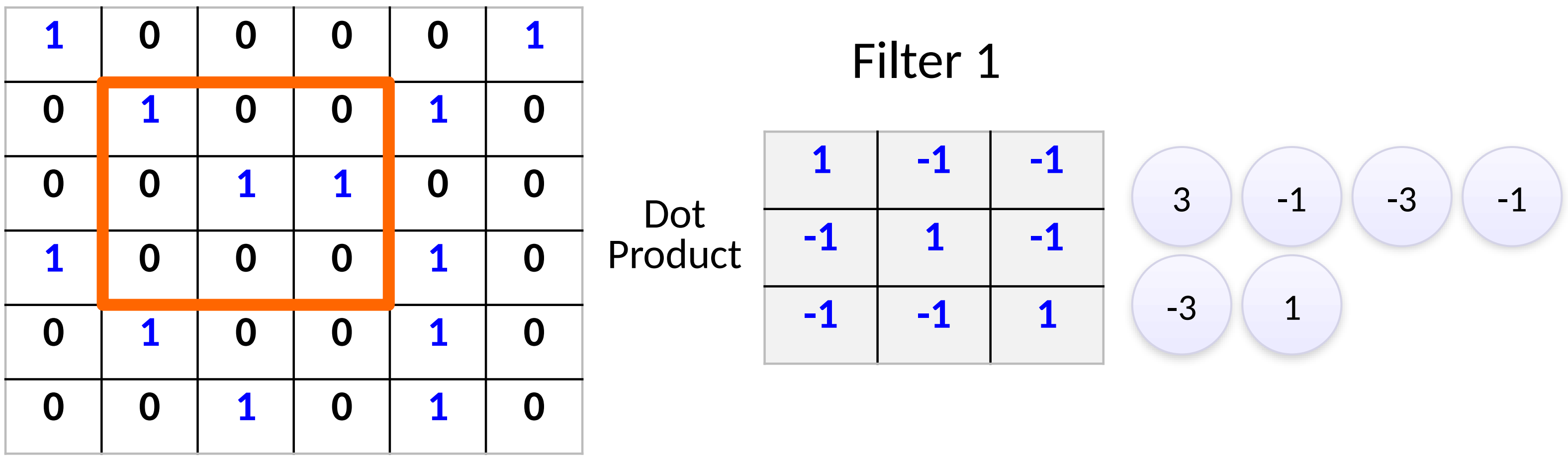


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Dot Product

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1



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

Dot Product

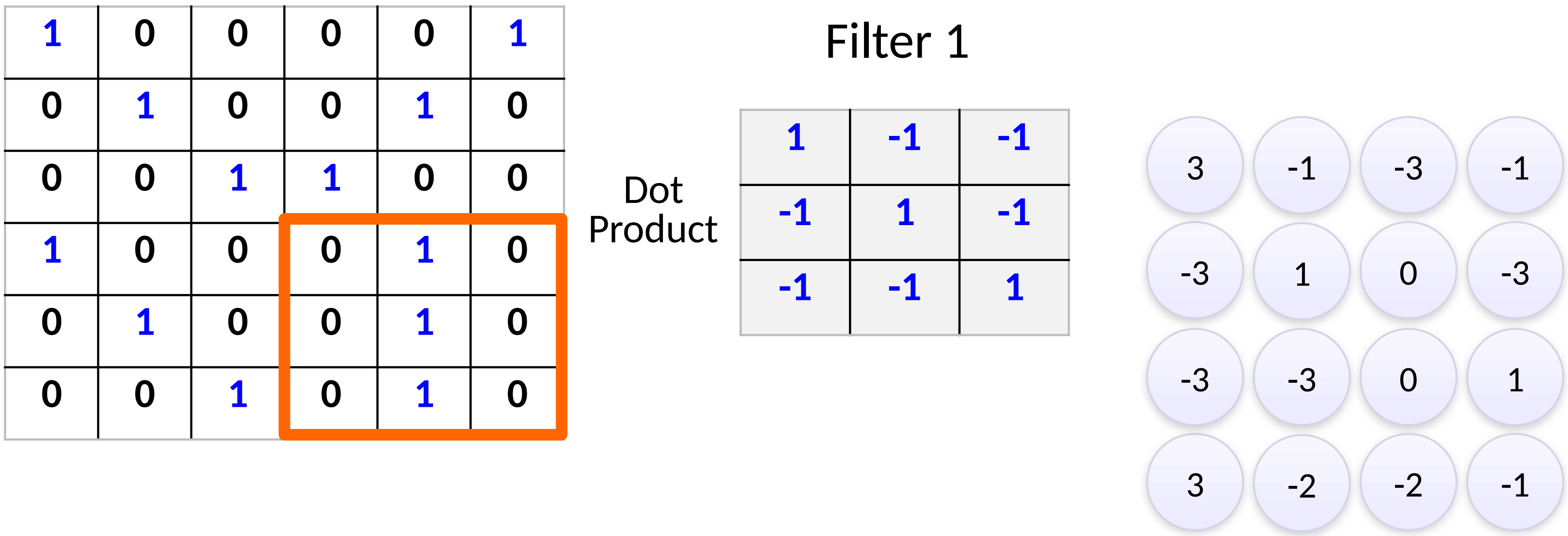
Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

Applying Filter 1

Stride=1

⋮



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

Dot Product

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Applying Filter 2

Stride=1

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

Filter 2

-1	1	-1
-1	1	-1
-1	1	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Feature Map

Stride=1

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

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

Filter 2

-1	1	-1
-1	1	-1
-1	1	-1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

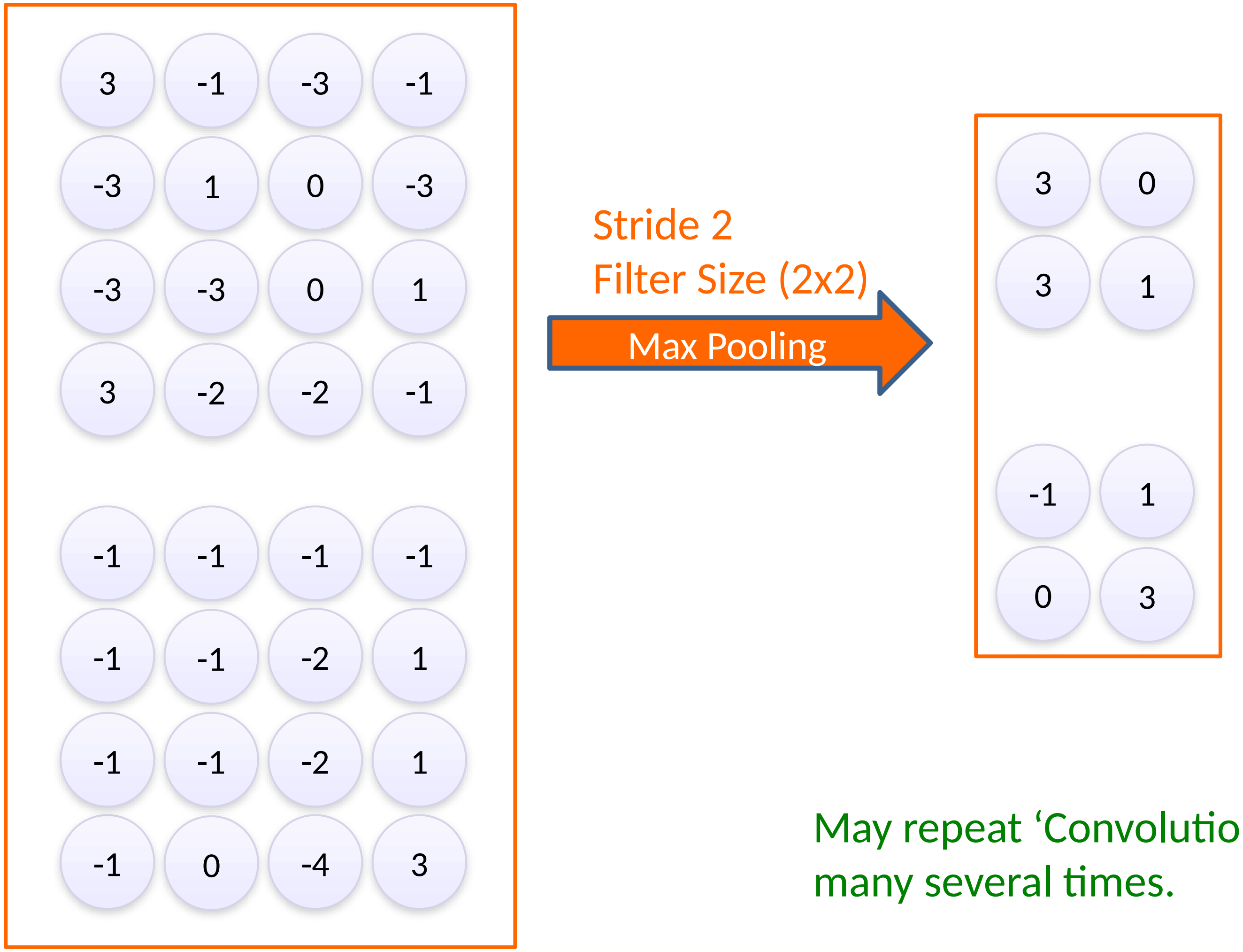
-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

(2 x 4 x 4) Matrix

Applying Max Pooling

Feature Map after Convolution
(2 x 4 x 4) Matrix

Feature Map after Max Pooling
(2 x 2 x 2) Matrix



3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Stride 2
Filter Size (2x2)

Max Pooling

3	0
3	1

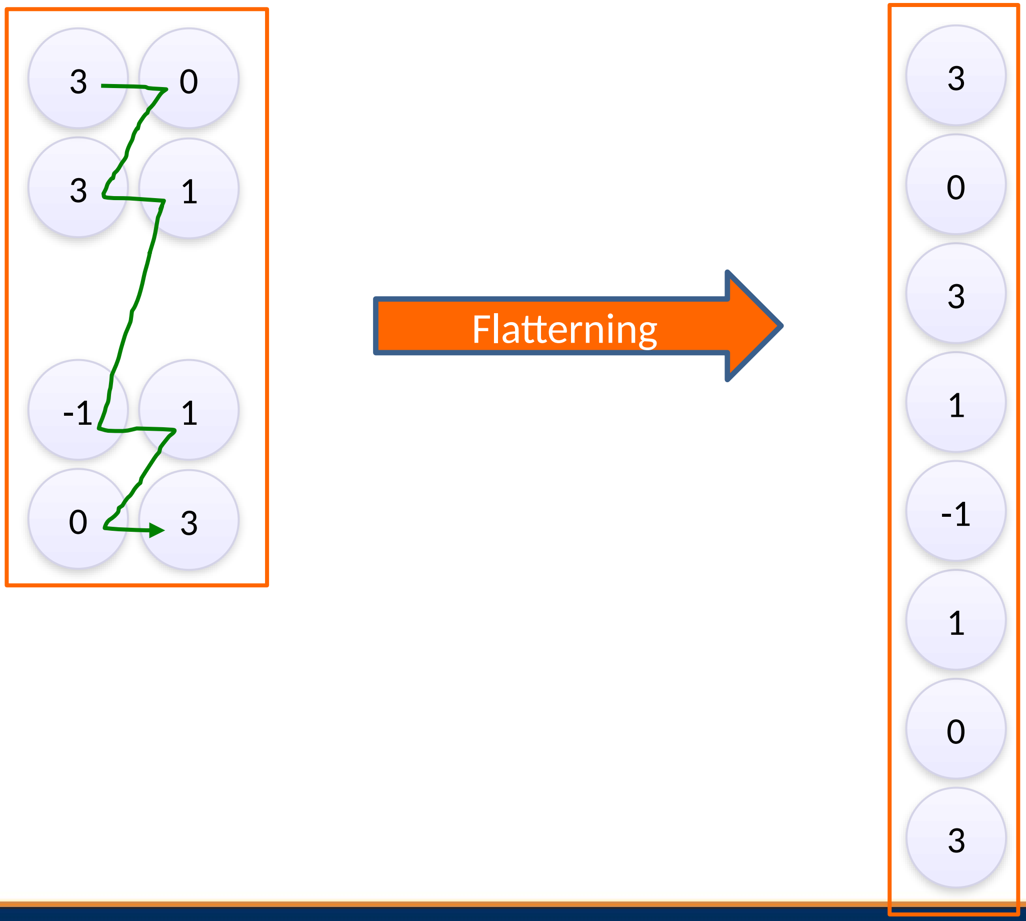
-1	1
0	3

May repeat ‘Convolution & Pooling’ many several times.

Applying Flattening

Feature Map after Max Pooling
(2 x 2 x 2) Matrix

Feature Map after Flattening
(1 x 8) Matrix

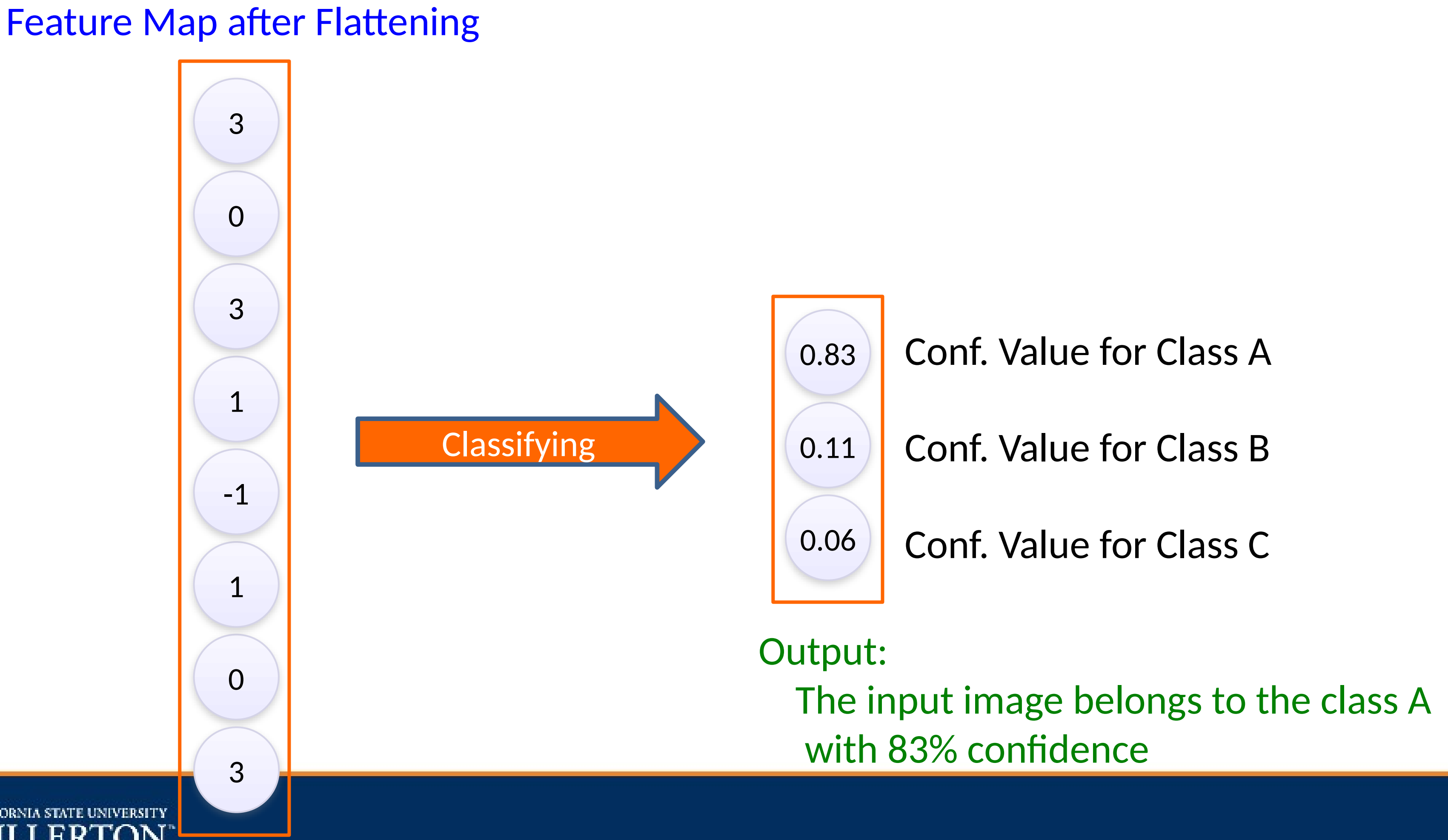


Feeding into Neural Network

- Feed the Flattening Result
  - Into a Neural Network Model that classifies.
- Example
  - To classify the input image among 3 classes
  - Fully Connected Layer
    - # of Neurons: 3 (Same as the # of Classes)
    - Each neuron returns the probability of each class to classify, i.e.,*Confidence Value* of each class.

Feeding into Neural Network

- To return the highest confidence value from the 1st neuron

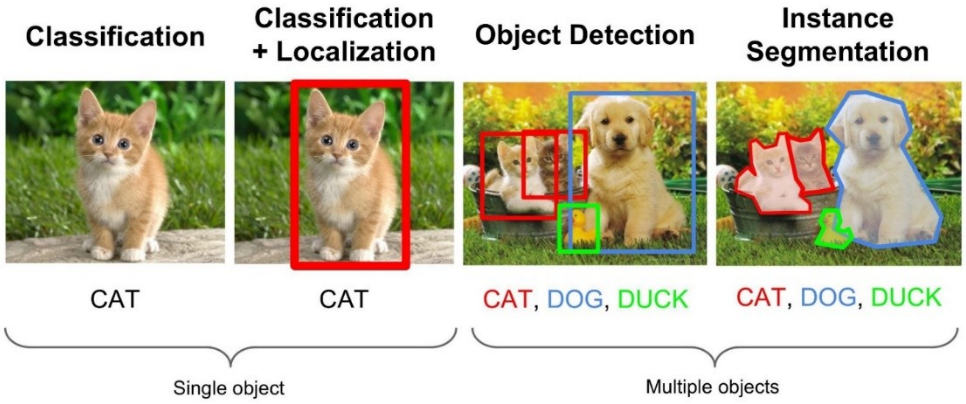


Output:

The input image belongs to the class A with 83% confidence

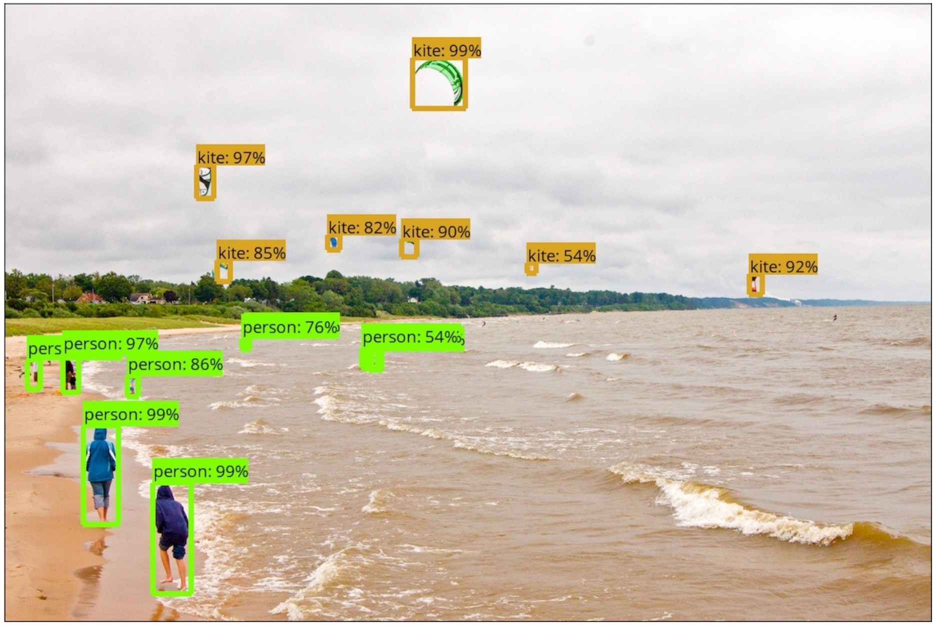
Applications of CNN

- CNN can be used for
  - Classification
  - Localization
  - Object Detection
  - Instance Segmentation
  - Image Generation
- Types of Data processed
  - Image
  - Video
  - Text



Applications of CNN

- Object Detection
  - To detect object's location in input image
    - To detect people and kits in input image



https://github.com/tensorflow/models/tree/master/research/object_detection

References

- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition