MPSSDEGB
कौशल से रोजगार, समृद्धि अपार

FITT
Foundation For Innovation And Technology Transfer

SANKALP
Ministry of Skill Development & Entrepreneurship

Advanced Neural Networks

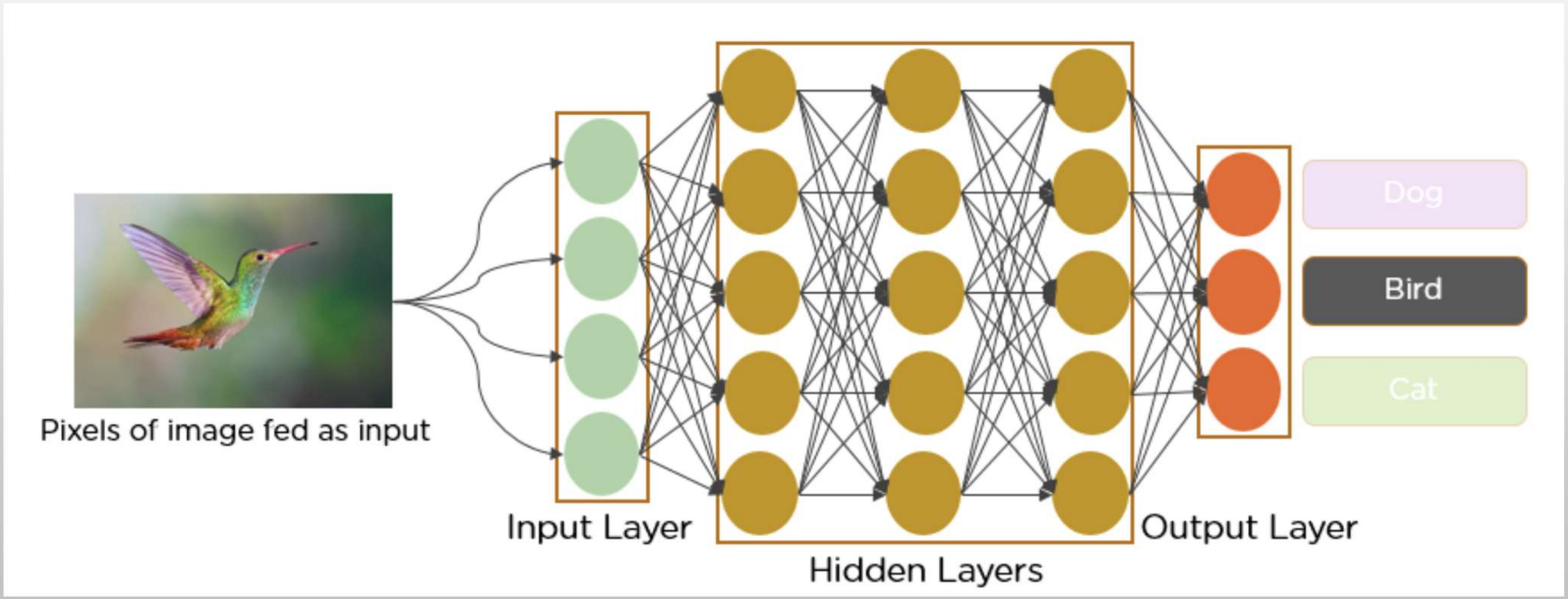
Convolutional Neural Networks (CNN)

Agenda

- CNN
- RNN

Convolutional Neural Networks (CNN)

- One of the most popular deep neural networks is Convolutional Neural Networks.
- Also known as CNN or ConvNet



Convolutional Neural Networks (CNN)

Background

- first developed and used around the 1980s.
  - recognize handwritten digits.

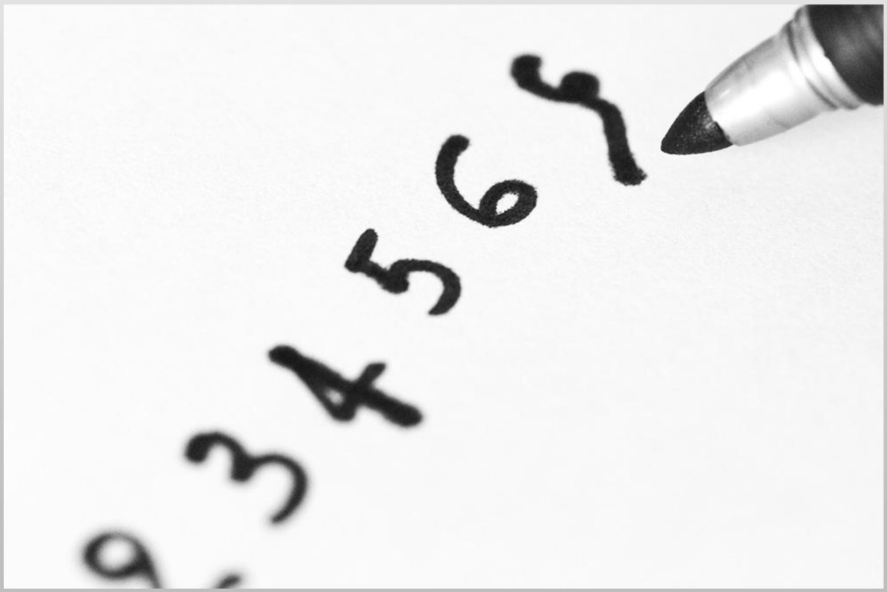
CNN

- It is a type of artificial neural network designed for processing structured grid data, such as images.
- CNNs have proven very effective in areas such as image recognition and classification.

The role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

Convolutional Neural Networks (CNN)

World's first CNN for handwriting recognition (video link)



Convolutional Neural Networks (CNN)

Basic overview of the CNN architecture:

1. Input Layer:

The input layer represents the raw data, typically an image or a sequence of data.

2. Convolutional Layers:

Convolutional layers apply convolution operations to the input data. These layers consist of filters (also called kernels) that slide over the input to extract local patterns or features.

3. Activation Function:

After each convolution operation, an activation function (commonly ReLU - Rectified Linear Unit) is applied element-wise to introduce non-linearity.

4. Pooling Layers:

Pooling layers down-sample the spatial dimensions of the data. Max pooling is a common operation, selecting the maximum value from a group of neighboring pixels.

5. Fully Connected Layers:

Fully connected layers connect every neuron from the previous layer to every neuron in the current layer. These layers often follow the convolutional and pooling layers.

6. Output Layer:

The output layer produces the final result, whether it's a classification, regression, or any other task.

Convolutional Neural Networks (CNN)

Layers in a Convolutional Neural Network

- A convolution neural network has multiple hidden layers that help in extracting information from an image.

- The four important layers in CNN are:
  - Convolution layer
  - ReLU layer
  - Pooling layer
  - Fully connected layer

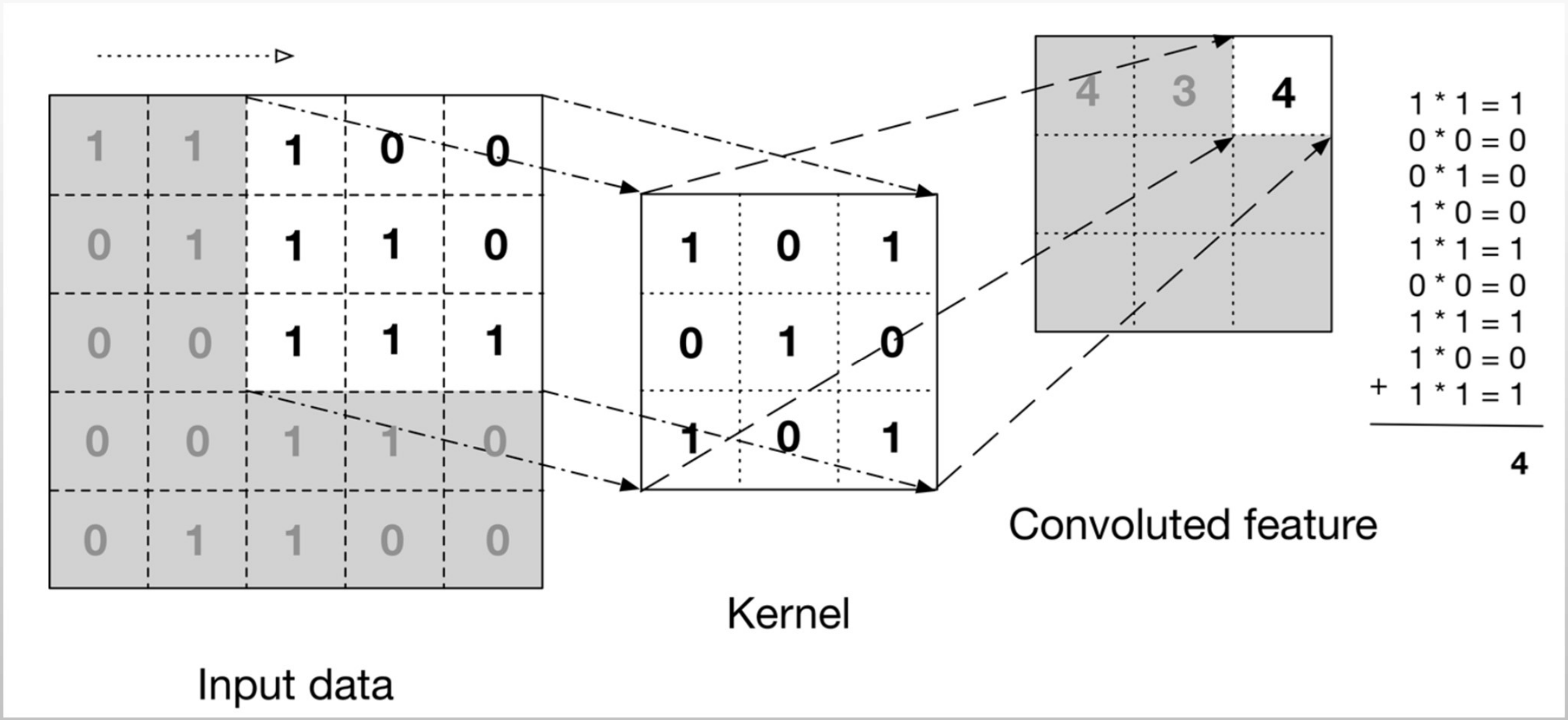
Convolutional Neural Networks (CNN)

Convolution layer

- A convolution layer has several filters that perform the convolution operation.
- Every image is considered as a matrix of pixel values.
    - Consider the following 5x5 image whose pixel values are either 0 or 1.
    - There’s also a filter matrix with a dimension of 3x3.
    - Slide the filter matrix over the image and compute the dot product to get the convolved feature matrix.

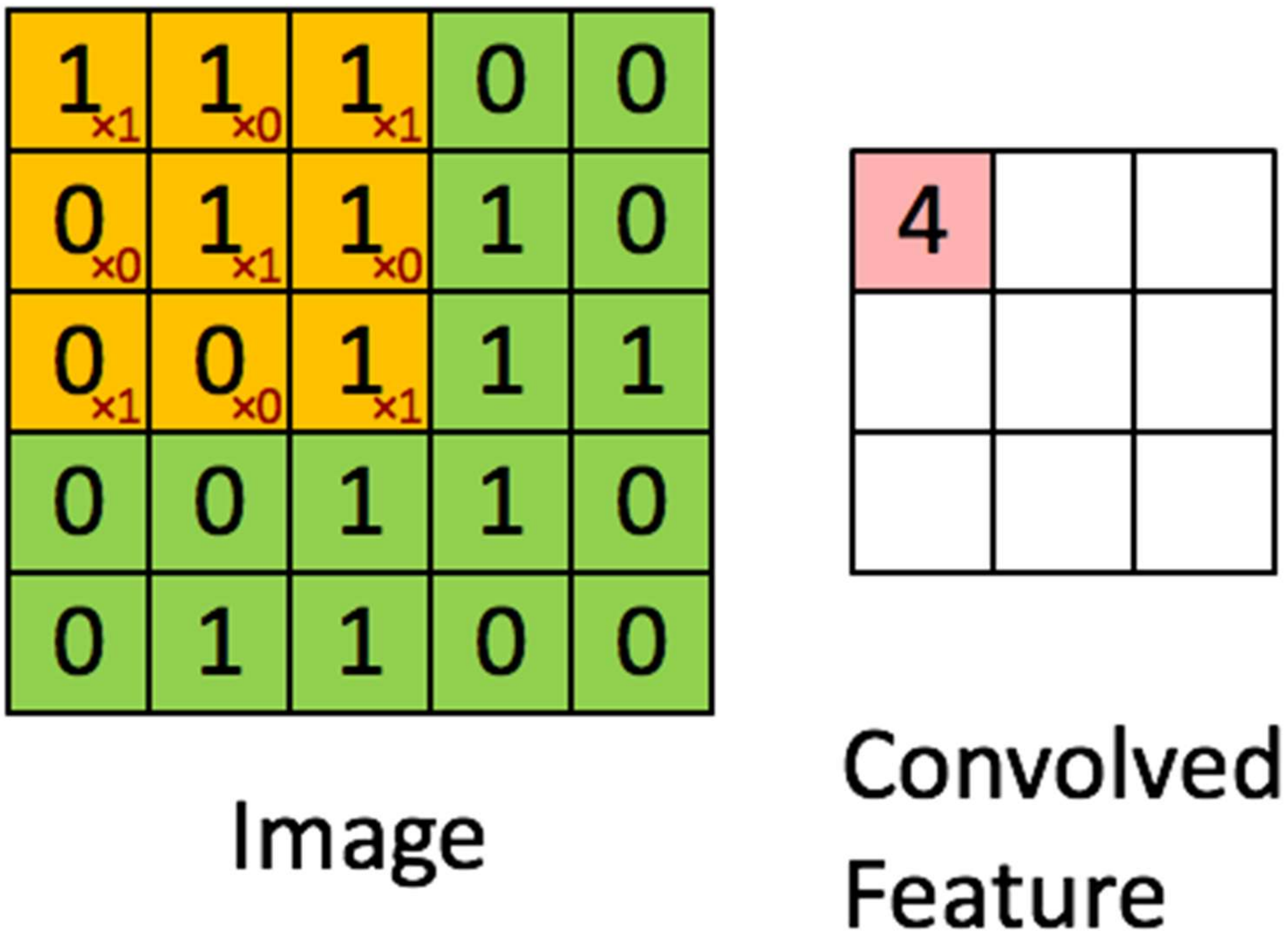
Convolutional Neural Networks (CNN)

Convolution

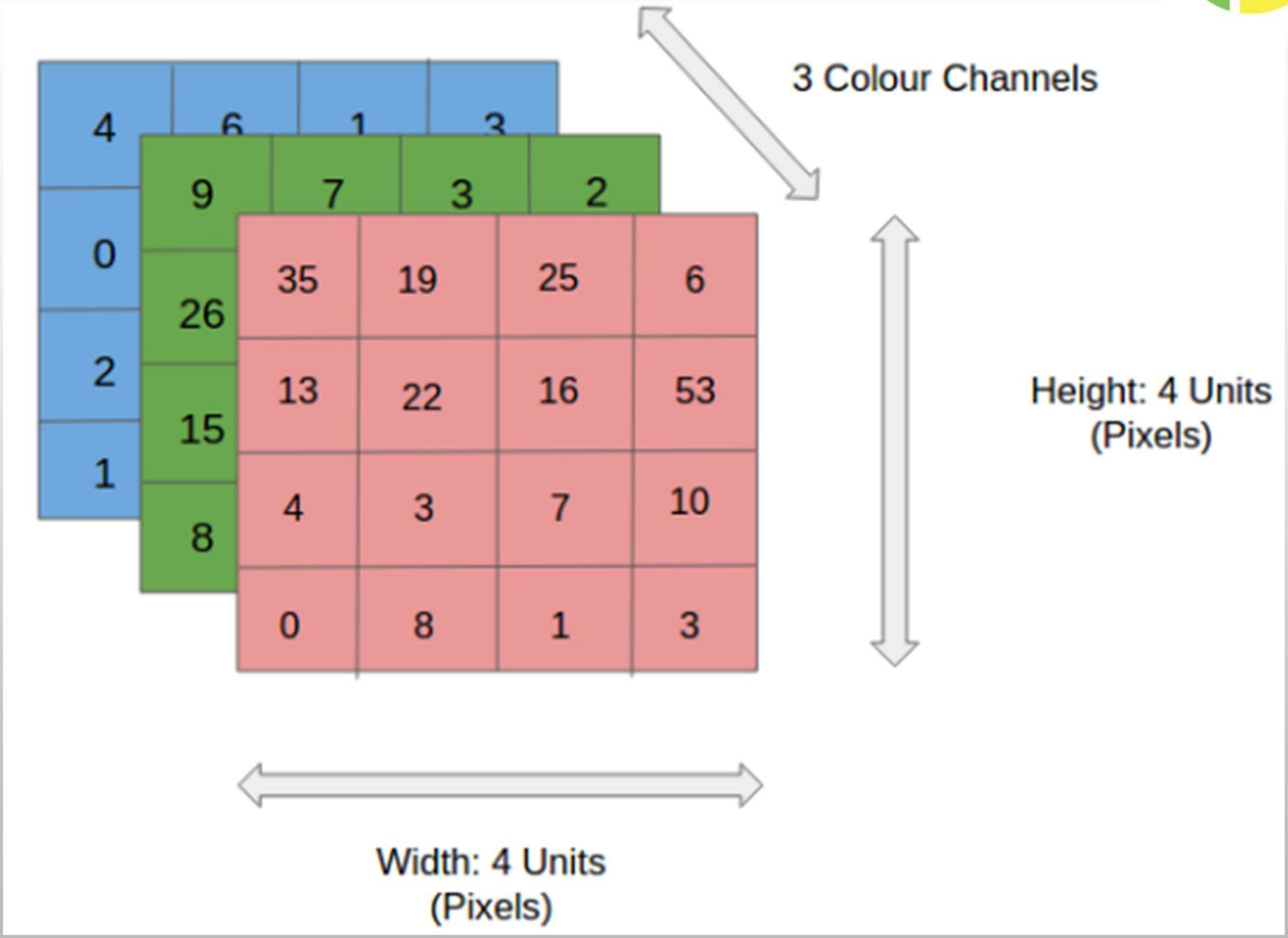


- A filter/kernel(3×3 matrix) is applied to the input image to get the convolved feature. This convolved feature is passed on to the next layer.

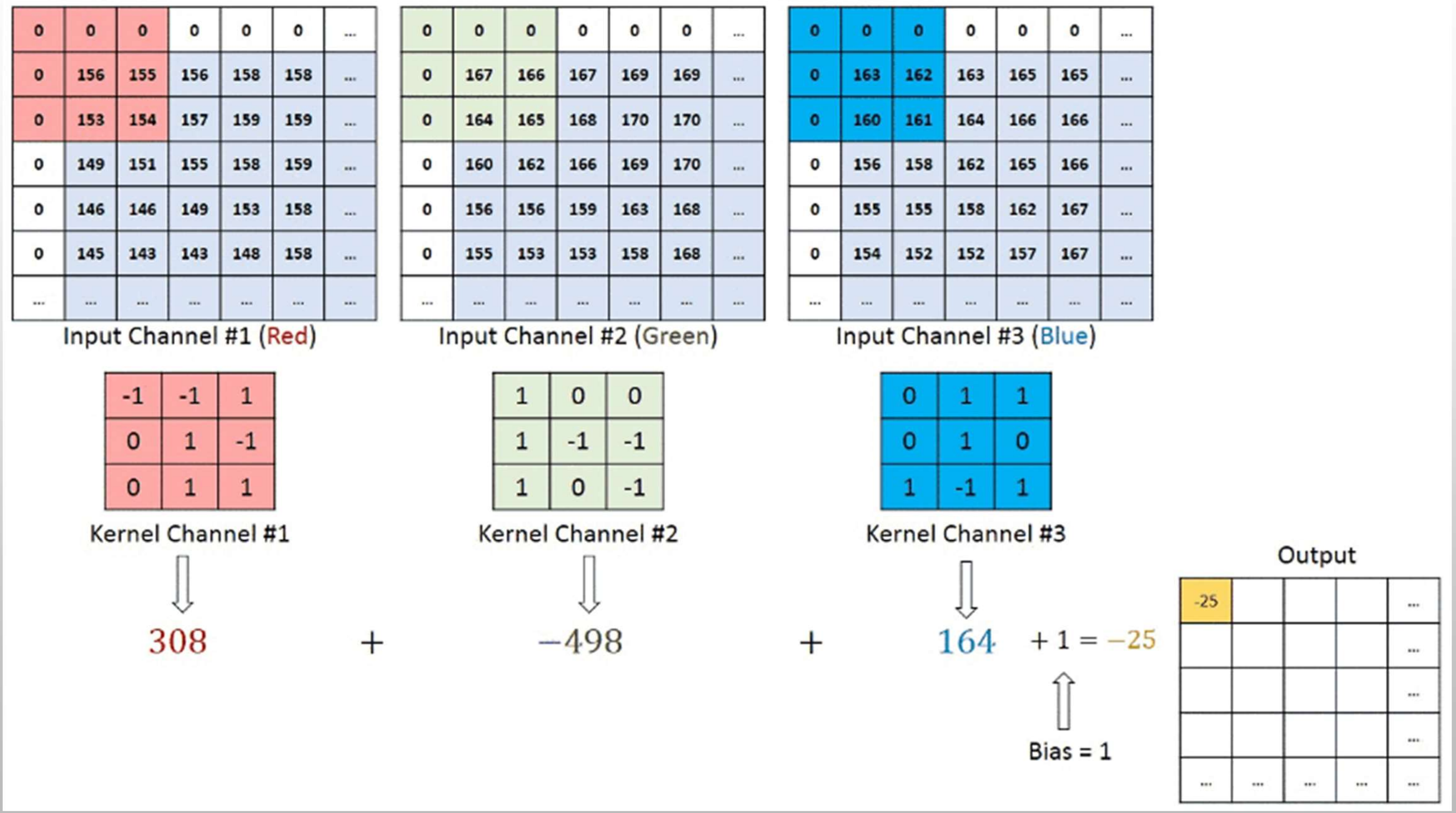
Convolutional Neural Networks (CNN)



Convolutional Neural Networks (CNN)



Convolutional Neural Networks (CNN)



0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

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

Kernel Channel #1

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

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

Kernel Channel #3

$$308 + -498 + 164 + 1 = -25$$

Bias = 1

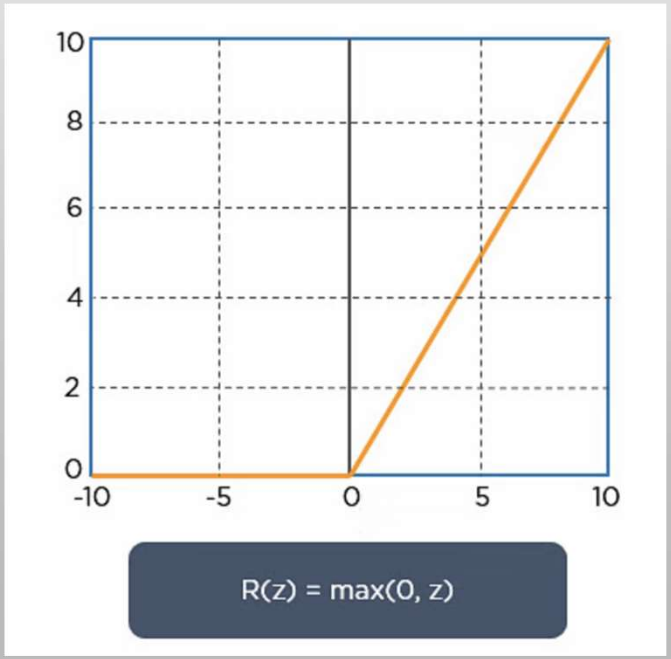
Output

-25				...
				...
				...
				...
...	...	...	...	...

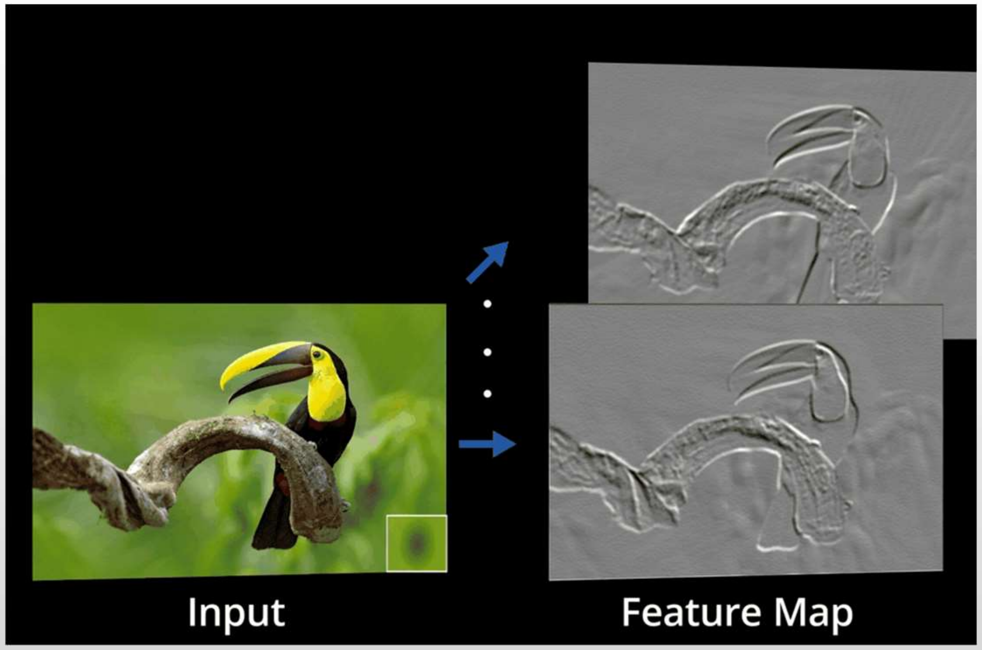
Convolutional Neural Networks (CNN)

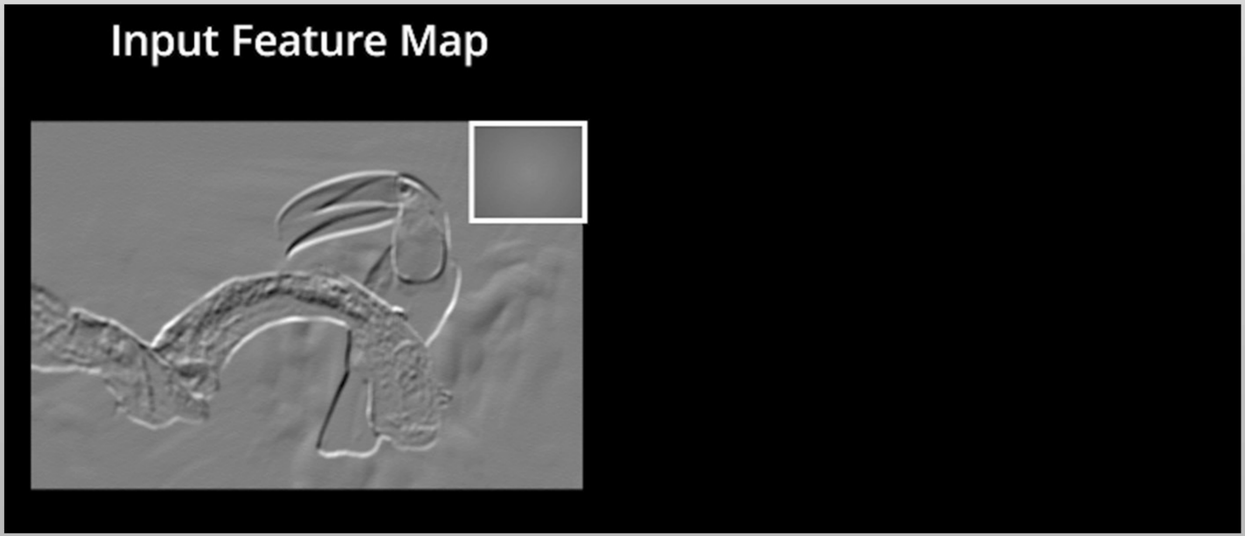
ReLU layer

- ReLU stands for the rectified linear unit.
- Once the feature maps are extracted, the next step is to move them to a ReLU layer.
- ReLU performs an element-wise operation and sets all the negative pixels to 0.
- It introduces non-linearity to the network, and the generated output is a rectified feature map.
- Below is the graph of a ReLU function:



Convolutional Neural Networks (CNN)

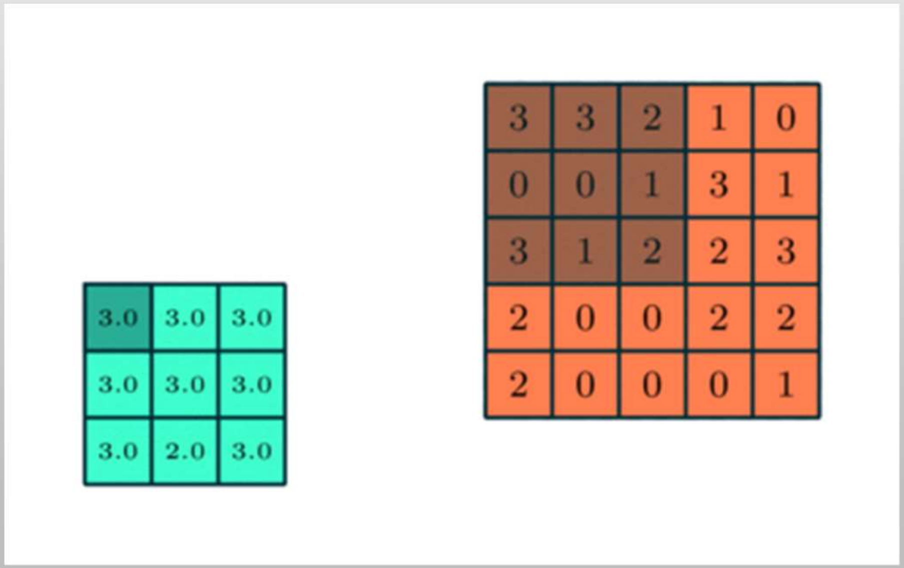




Convolutional Neural Networks (CNN)

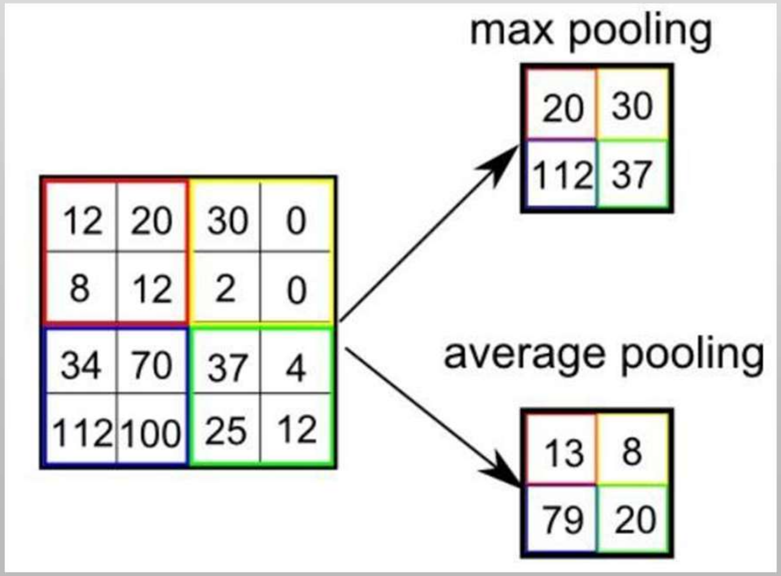
Pooling Layer

- Also known as a subsampling or downsampling layer
- It is responsible for reducing the spatial size of the Convolved Feature.
- This is to **decrease the computational power required to process the data** by reducing the dimensions.
- There are two types of pooling:
  - average pooling
  - max pooling



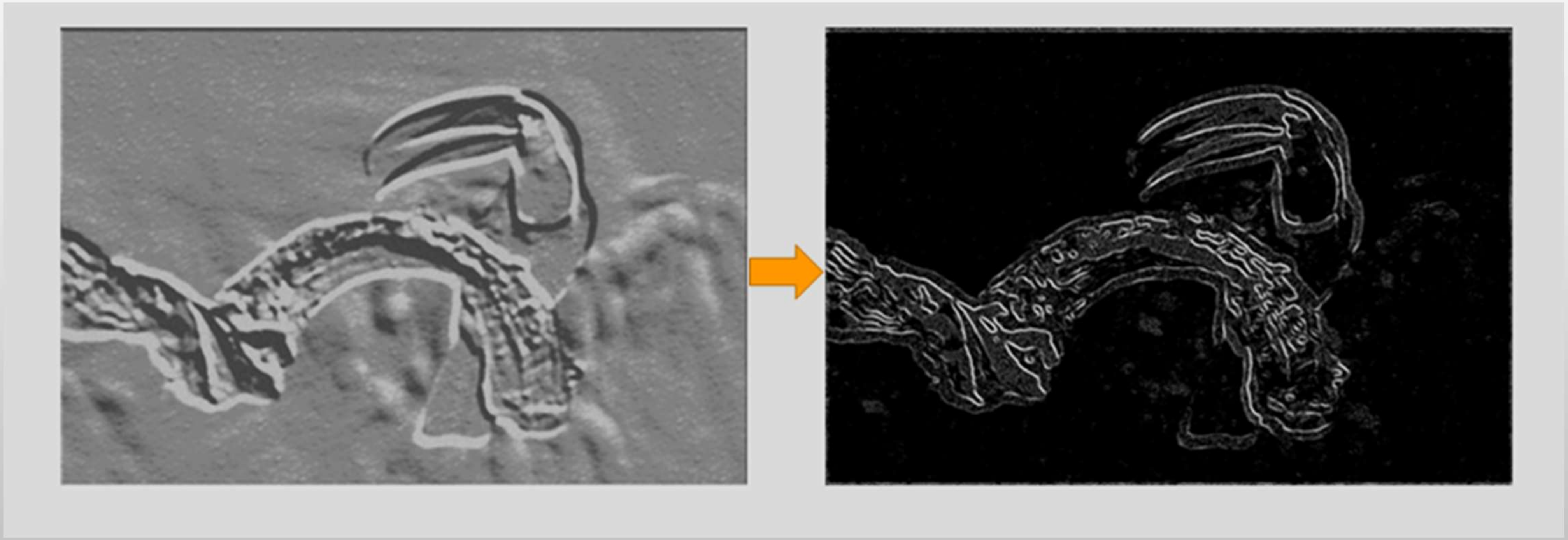
Convolutional Neural Networks (CNN)

1. **Max Pooling:**
   1. A window moves over the input volume, and the maximum value within that window is selected as the representative value for that region.
   2. Retain the most important features in a given region and discards less relevant information.
2. **Average Pooling:**
   1. Instead of selecting the maximum value, it calculates the average of the values in the window.
   2. Average pooling can smooth out the representations and provide a more generalized view of the input.



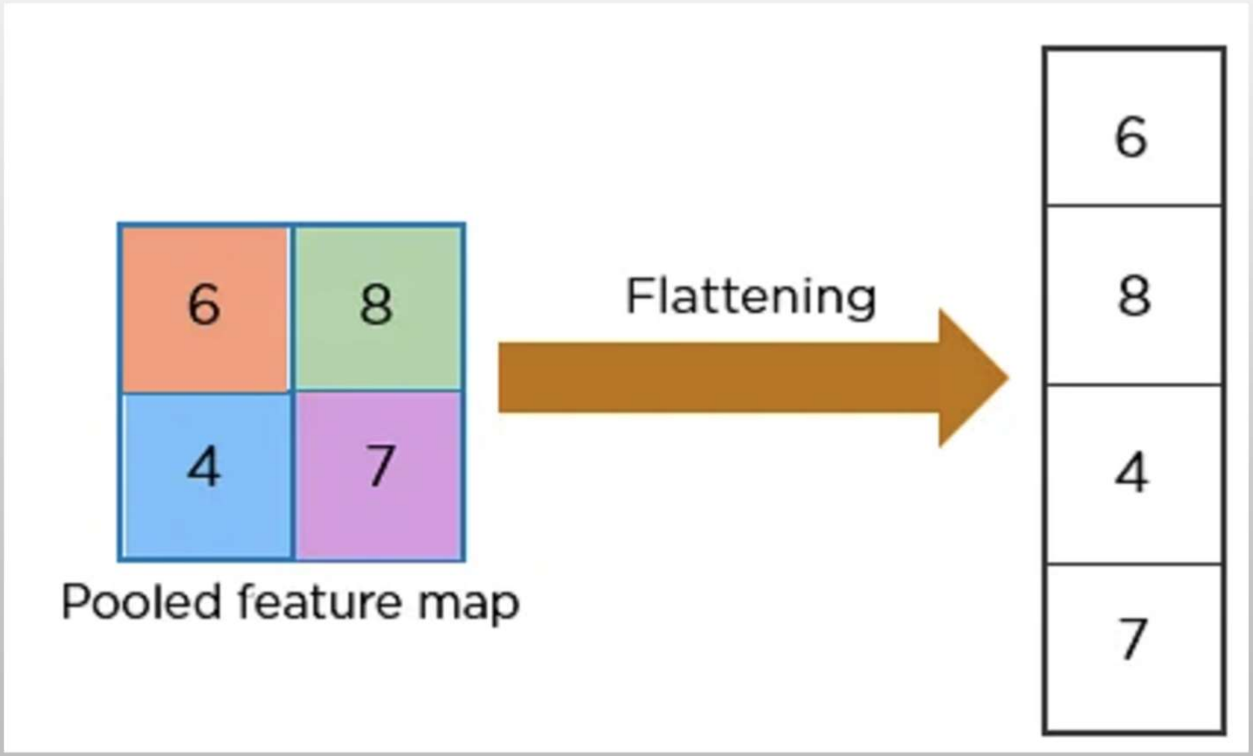
Convolutional Neural Networks (CNN)

- The convolution layer uses various filters to identify different parts of the image like edges, corners, body, feathers, eyes, beak etc. Whereas pooling layer downsamples the data i.e reduces the dimensions of the feature map



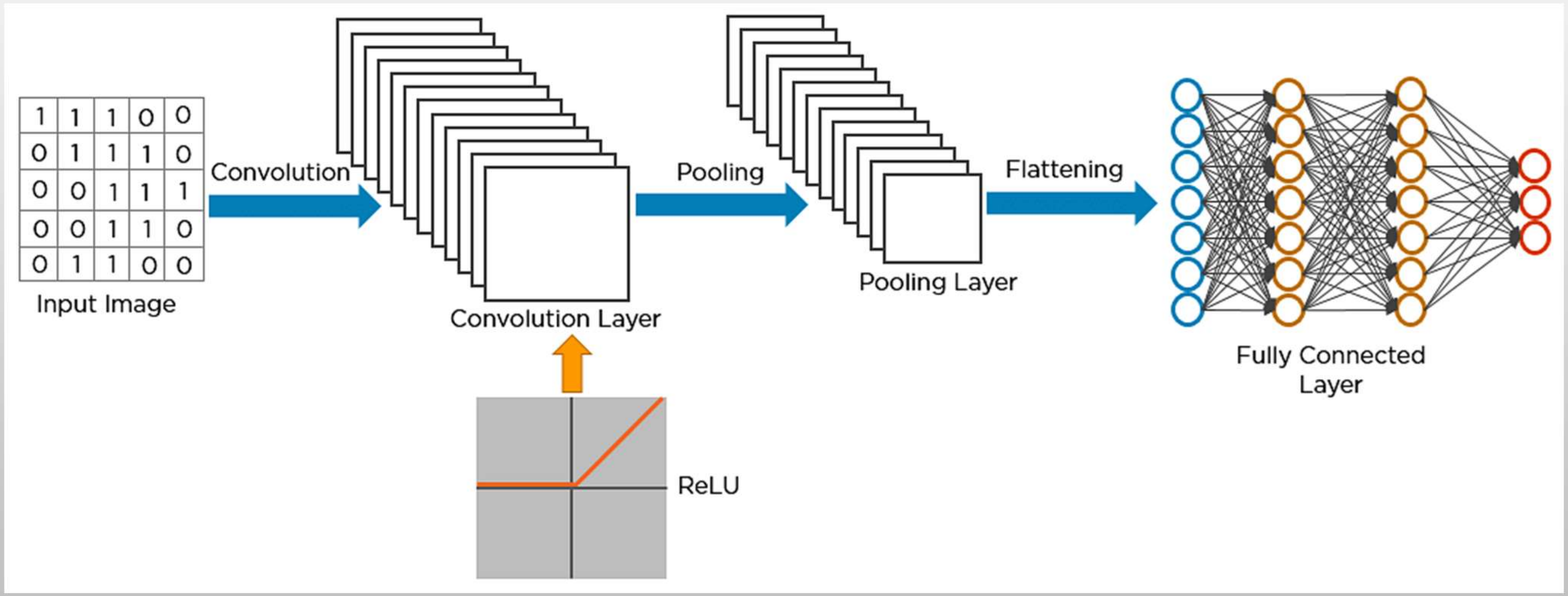
Convolutional Neural Networks (CNN)

- Flattening is used to convert all the resultant 2D arrays from pooled feature maps into a single long continuous linear vector.

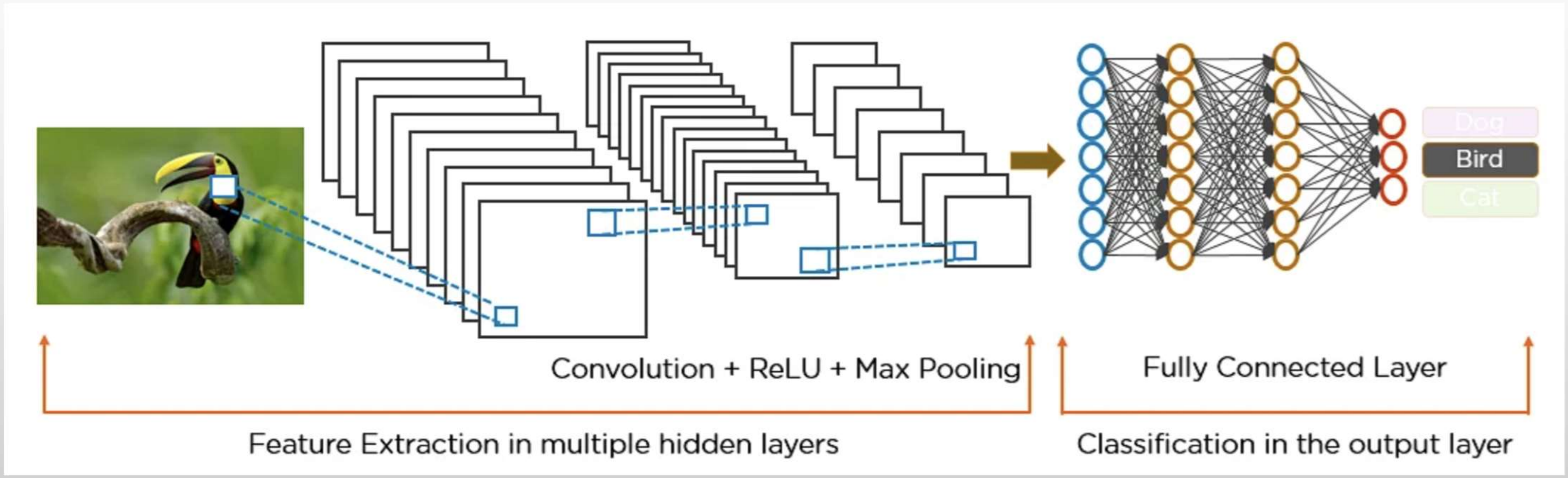


Convolutional Neural Networks (CNN)

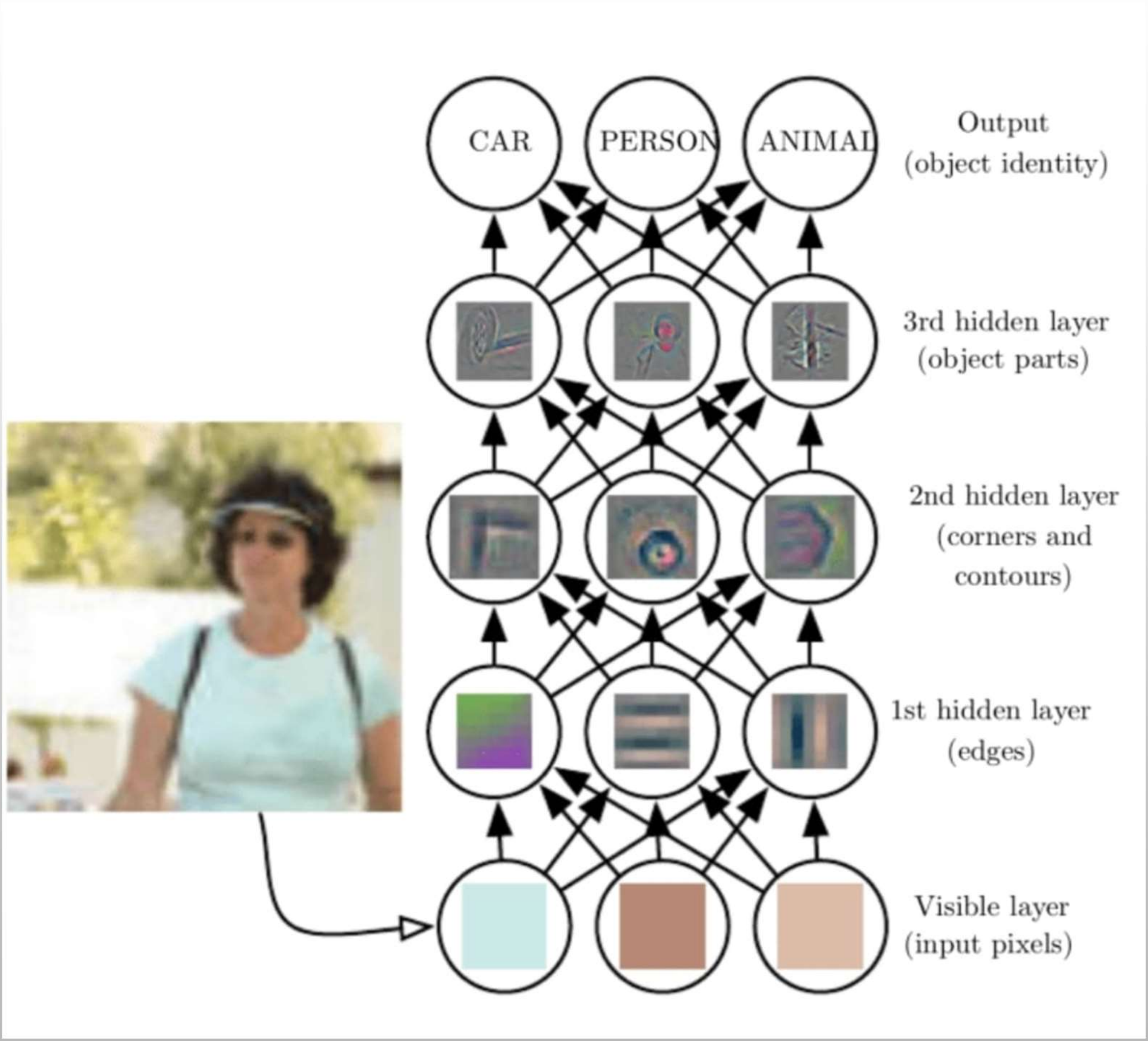
- The flattened matrix is fed as input to the fully connected layer to classify the image.



Convolutional Neural Networks (CNN)


Dog
Bird
Cat
Convolution + ReLU + Max Pooling
Fully Connected Layer
Feature Extraction in multiple hidden layers
Classification in the output layer

Convolutional Neural Networks (CNN)



Convolutional Neural Networks (CNN)

Use Cases of CNNs:

Convolutional Neural Networks have become fundamental in computer vision and are widely applied across various domains due to their ability to automatically learn hierarchical features from structured grid data like images.

1. Image Classification:
CNNs excel at image classification tasks, such as identifying objects in photographs. They have been used in competitions like ImageNet to achieve state-of-the-art results.

2. Object Detection:
For identifying and locating multiple objects within an image. Applications include autonomous vehicles, surveillance, and facial recognition systems.

3. Semantic Segmentation:
CNNs can be used for pixel-wise classification, separating an image into different segments or classes. This is valuable in medical imaging, satellite imagery analysis, and more.

4. Image Generation and Style Transfer:
CNNs can be used to generate new images, as seen in generative models like GANs (Generative Adversarial Networks). Style transfer techniques use CNNs to apply the artistic style of one image to another.

Convolutional Neural Networks (CNN)

5. Speech Recognition:
CNNs can be applied to audio data for speech recognition tasks, learning hierarchical features from audio spectrograms.

6. Natural Language Processing (NLP):
For processing sequential data like text, CNNs can be used for tasks such as sentiment analysis and text classification.

7. Medical Image Analysis:
Identifying patterns and anomalies in medical images, such as X-rays and MRIs, for diagnostics and treatment planning.

8. Video Analysis:
CNNs can be extended to video data, enabling tasks like action recognition, scene understanding, and tracking.

9. Autonomous Vehicles:
CNNs play a crucial role in computer vision systems for autonomous vehicles, helping in object detection, lane detection, and scene understanding.

10. Virtual and Augmented Reality:
CNNs contribute to creating immersive experiences by enabling facial recognition, gesture recognition, and object recognition in AR/VR applications.

Convolutional Neural Networks (CNN)

Quiz

CNN - Quiz

Recurrent Neural Networks (RNN)

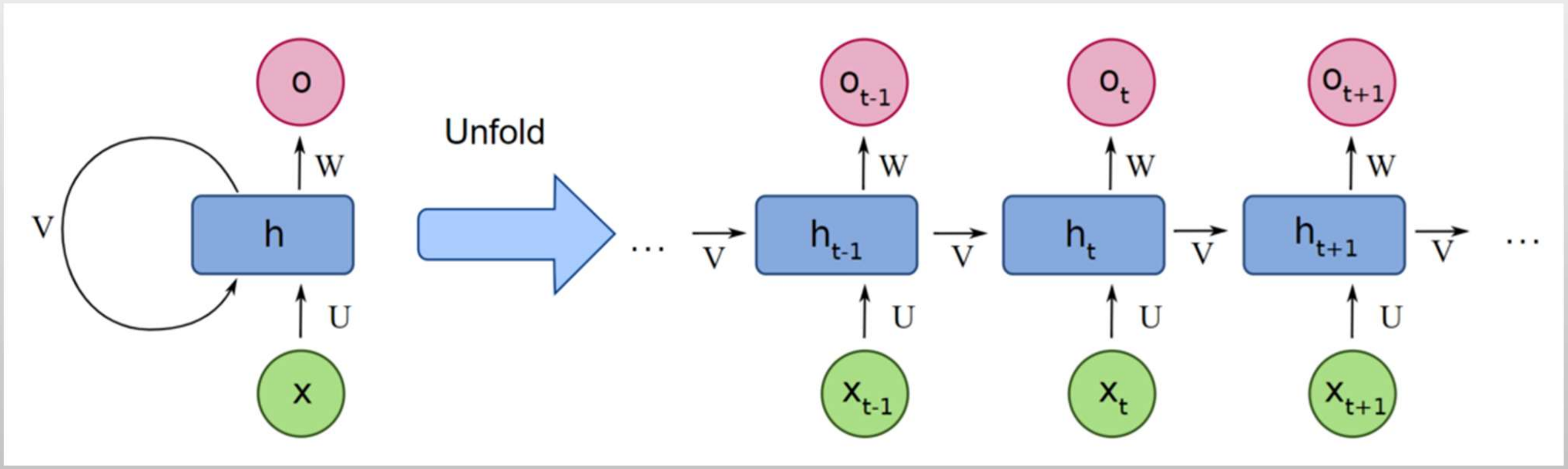
Suppose you want to predict the last word in the text:
"The clouds are in the ________".
The most obvious answer to this is the "sky".

Consider this sentence:
"I have been staying in Spain for the last 10 years…I can speak fluent ________."
The most suitable answer to this sentence is "Spanish".

- The word you predict will depend on the previous few words in context.
- Here, you need the context of Spain to predict the last word in the text.

Recurrent Neural Networks (RNN)

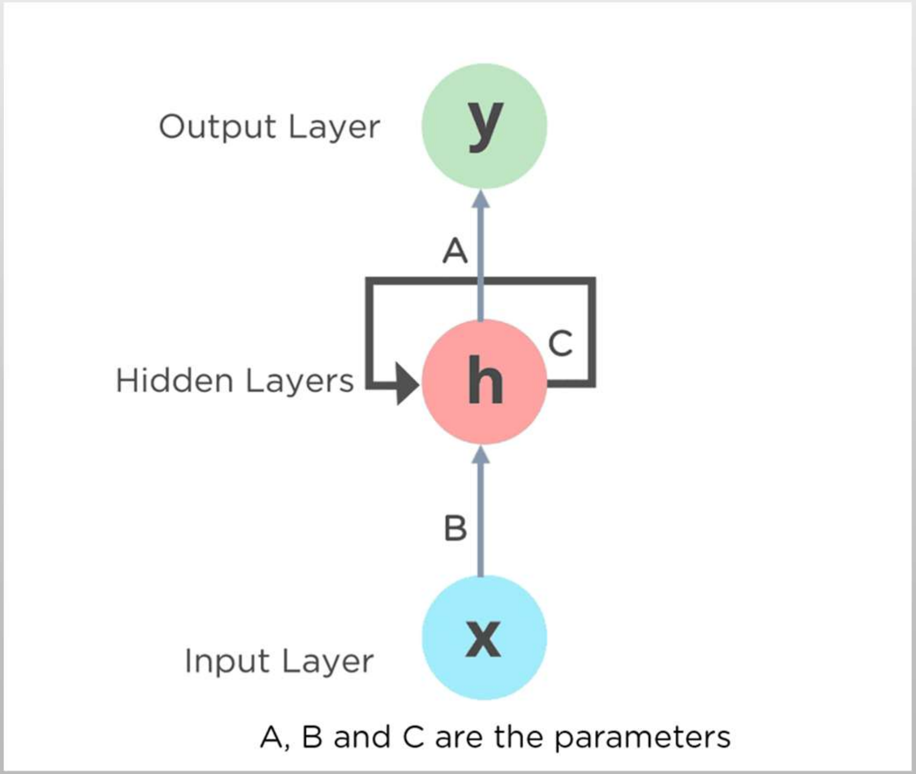
- A Deep Learning approach for modelling sequential data.
- They work especially well for jobs requiring sequences, such as time series data, voice, natural language, and other activities.



Recurrent Neural Networks (RNN)

- RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer.

- The nodes in different layers of the neural network are compressed to form a single layer of recurrent neural networks. A, B, and C are the parameters of the network.



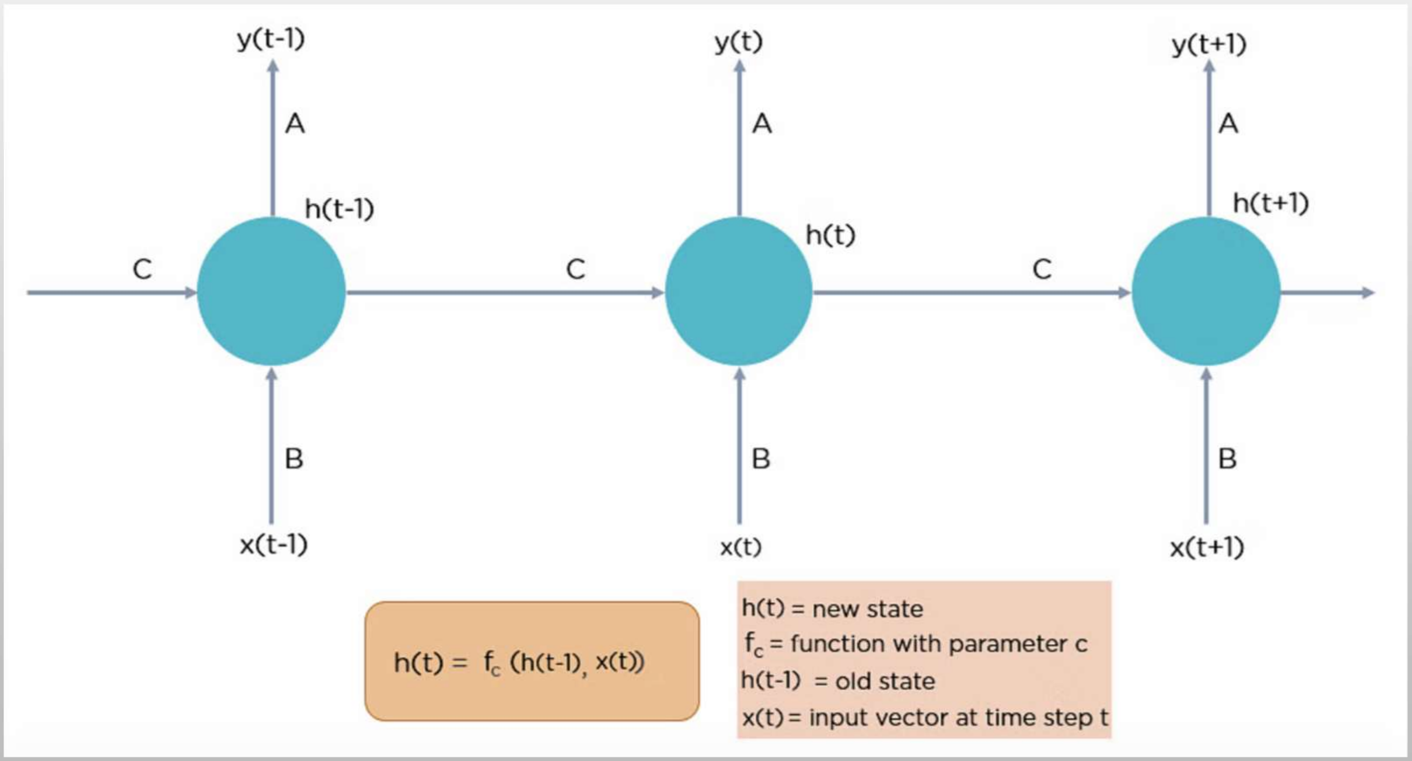
Rotate anticlockwise and compress the layers.

Recurrent Neural Networks (RNN)

Architecture

- RNNs are a type of neural network that has hidden states and allows past outputs to be used as inputs.

- They usually go like this:



y(t-1)
y(t)
y(t+1)
A
h(t-1)
h(t)
h(t+1)
C
B
x(t-1)
x(t)
x(t+1)
$h(t) = f_c(h(t-1), x(t))$
$h(t)$ = new state
f_c = function with parameter c
$h(t-1)$ = old state
$x(t)$ = input vector at time step t

"x" is the input layer
"h" is the hidden layer
"y" is the output layer
A, B, and C are the network parameters used to improve the output of the model

The output at any given time is fetched back to the network to improve on the output.

Recurrent Neural Networks (RNN)

Why Recurrent Neural Networks?

- RNN were created because there were a few issues in the feed-forward neural network:
  - Cannot handle sequential data
  - Considers only the current input
  - Cannot memorize previous inputs

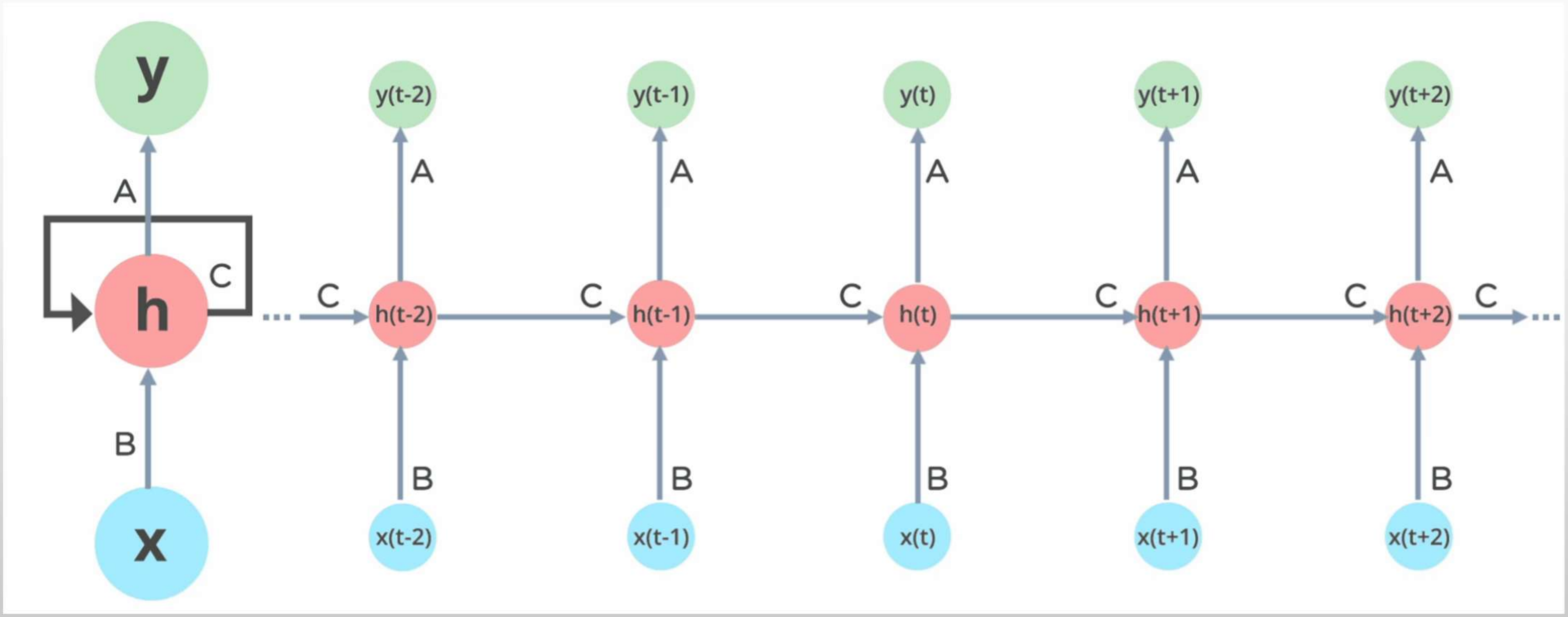
- The solution to these issues is the RNN.

- An RNN can handle sequential data, accepting the current input data, and previously received inputs.

- RNNs can memorize previous inputs due to their internal memory.

Recurrent Neural Networks (RNN)

Working



- The RNN will standardize the different activation functions and weights and biases so that each hidden layer has the same parameters.

- Then, instead of creating multiple hidden layers, it will create one and loop over it as many times as required.

Recurrent Neural Networks (RNN)

Feed-Forward Neural Networks vs Recurrent Neural Networks

- A feed-forward neural network allows information to flow only in the forward direction, from the input nodes, through the hidden layers, and to the output nodes. There are no cycles or loops in the network.
- In a feed-forward neural network, the decisions are based on the current input. It doesn’t memorize the past data, and there’s no future scope.

Recurrent Neural Networks (RNN)

Applications:

- Image Captioning
- Time Series Prediction
- Natural Language Processing
- Machine Translation

Recurrent Neural Networks (RNN)

Advantages

- Ability To Handle Variable-Length Sequences
- Memory Of Past Inputs
- Parameter Sharing
- Non-Linear Mapping
- Sequential Processing
- Flexibility
- Improved Accuracy

Recurrent Neural Networks (RNN)

Disadvantages

- Vanishing And Exploding Gradients
- Computational Complexity
- Difficulty In Capturing Long-Term Dependencies
- Lack Of Parallelism
- Difficulty In Choosing The Right Architecture
- Difficulty In Interpreting The Output

Recurrent Neural Networks (RNN)

Types of Recurrent Neural Networks

There are four types of Recurrent Neural Networks:

1.One to One

1.One to Many

1.Many to One

1.Many to Many

Recurrent Neural Networks (RNN)

One to One RNN

known as the Vanilla Neural Network. It's used for general machine learning problems, which has a single input and a single output.

One to Many RNN

has a single input and multiple outputs.

Ex: The image caption.

Many to One RNN

Takes a sequence of inputs and generates a single output.

Ex: Sentiment analysis.

Many to Many RNN

Takes a sequence of inputs and generates a sequence of outputs.

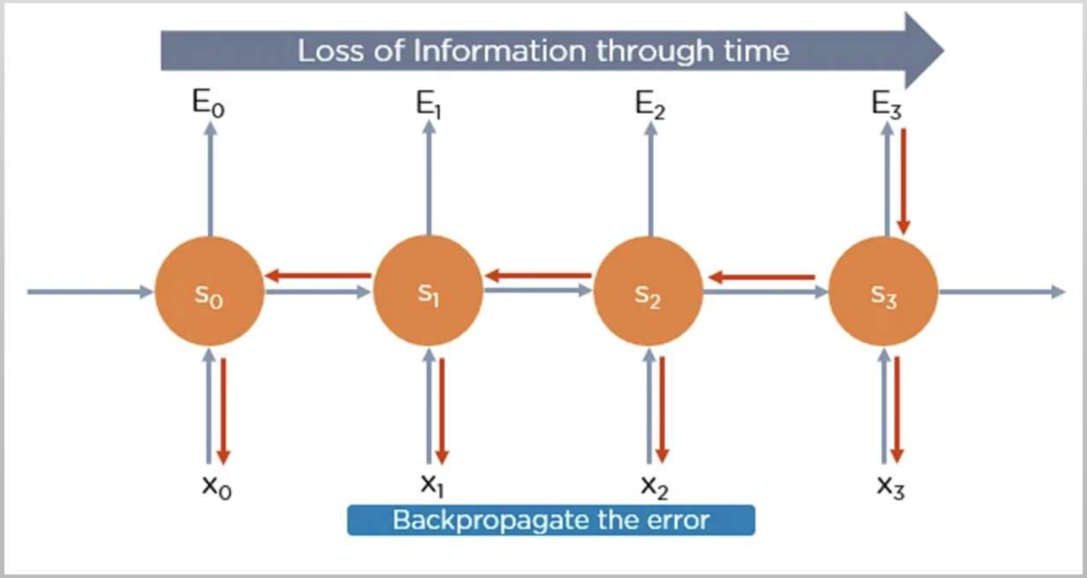
Ex: Machine translation.

Recurrent Neural Networks (RNN)

Two Issues of Standard RNNs:

1. **Vanishing Gradient Problem**

- When the gradient becomes too small, the parameter updates become insignificant.
- This makes the learning of long data sequences difficult.



Recurrent Neural Networks (RNN)

2. Exploding Gradient Problem

- If the slope tends to grow exponentially instead of decaying, this is called an Exploding Gradient.
- This problem arises when large error gradients accumulate, resulting in very large updates to the neural network model weights during the training process.
- Long training time, poor performance, and bad accuracy are the major issues in gradient problems.

Recurrent Neural Networks (RNN)

Backpropagation Through Time

- Backpropagation Through Time (BPTT) is an extension of the backpropagation algorithm for training RNNs and other sequential models.
- While traditional backpropagation is used to update weights in feedforward neural networks, BPTT is designed to handle the temporal dependencies in recurrent architectures.

Long-Term Dependencies:

- BPTT is particularly crucial for capturing and learning long-term dependencies in sequential data.
- The recurrent connections in the network allow information to be passed from one time step to the next
- BPTT ensures that the network can learn from these temporal relationships.

Recurrent Neural Networks (RNN)

Variant RNN Architectures

- There are several variant RNN architectures that have been developed over the years to address the limitations of the standard RNN architecture.
- Few examples are:

    Long Short-Term Memory (LSTM) Networks

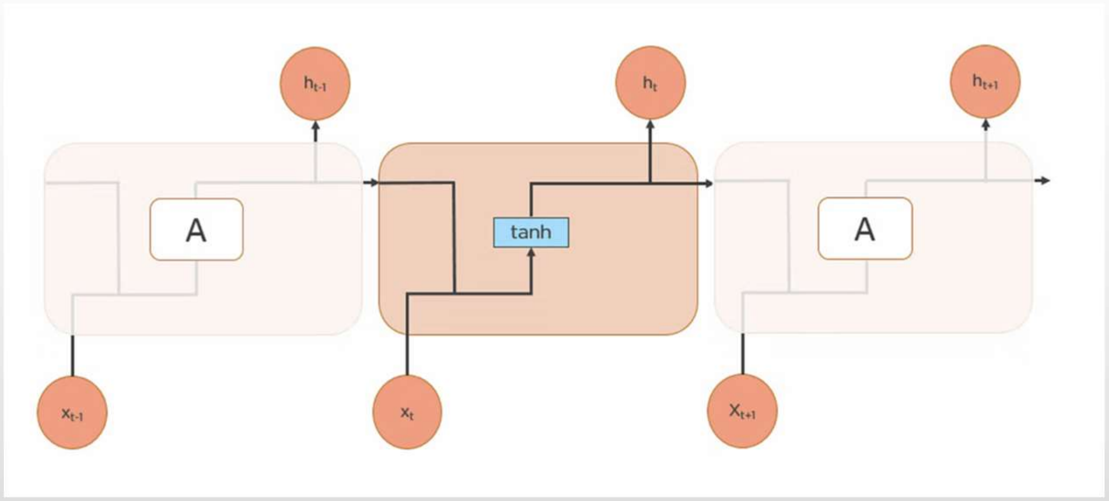
    Gated Recurrent Unit (GRU) Networks

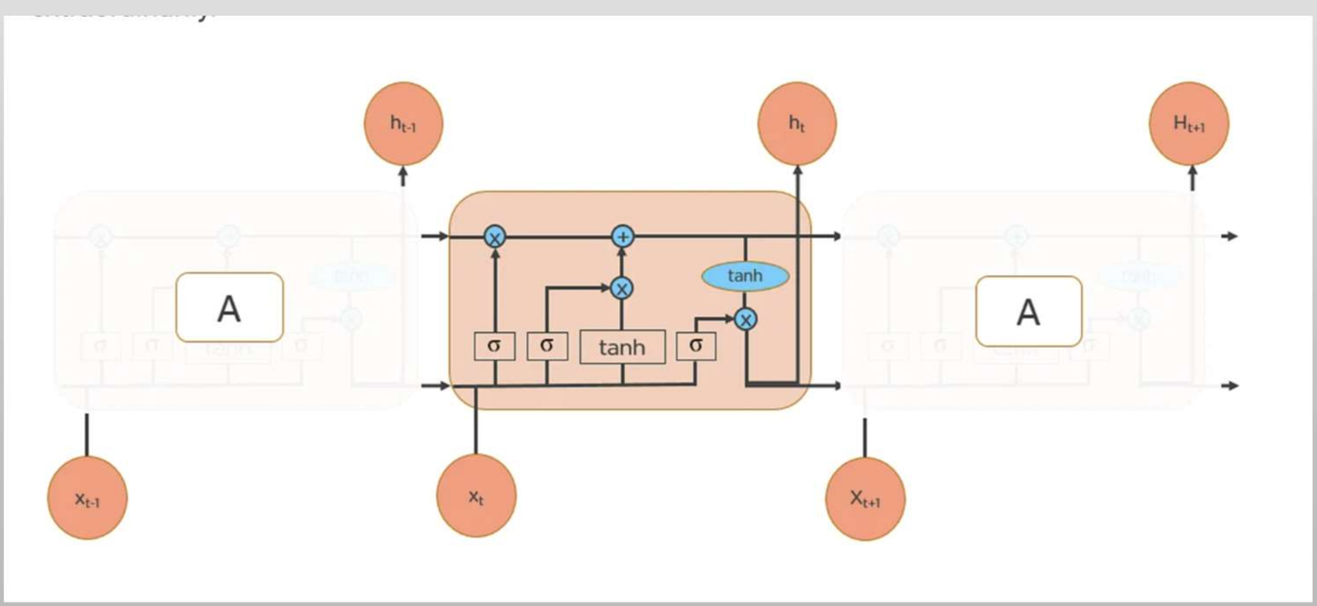
Recurrent Neural Networks (RNN)

Long Short-Term Memory (LSTM) Networks

- To handle the vanishing gradient problem
- By introducing three gating mechanisms that control the flow of information through the network:
    - input gate
    - forget gate
    - output gate
- These gates allow the LSTM network to selectively remember or forget information from the input sequence.

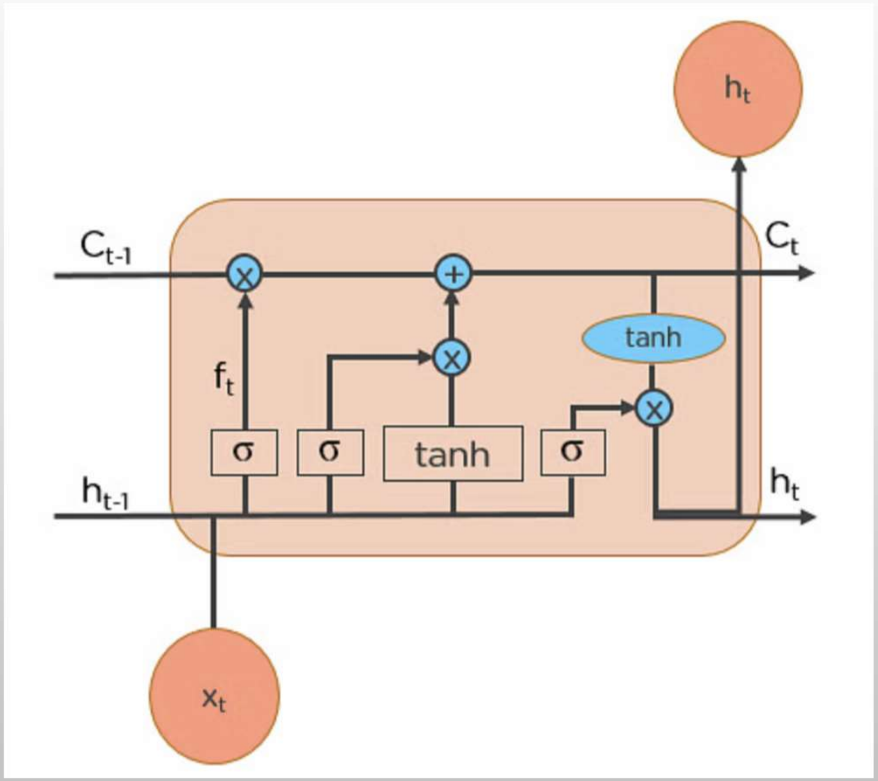
Recurrent Neural Networks (RNN)





Recurrent Neural Networks (RNN)

Workings of LSTMs in RNN



LSTMs work in a 3-step process.

Step 1: Decide How Much Past Data It Should Remember

Step 2: Decide How Much This Unit Adds to the Current State

Step 3: Decide What Part of the Current Cell State Makes It to the Output

Recurrent Neural Networks (RNN)

Gated Recurrent Unit (GRU) Networks

- to address the vanishing gradient problem.
- It has two gates:
    - the reset gate
    - the update gate
- The reset gate determines how much of the previous state should be forgotten, while the update gate determines how much of the new state should be remembered.
- This allows the GRU network to selectively update its internal state based on the input sequence.

Recurrent Neural Networks (RNN)

Use Cases of RNNs:

Recurrent Neural Networks (RNNs) are particularly well-suited for tasks involving sequential or time-series data due to their ability to capture temporal dependencies.

1. **Natural Language Processing (NLP):**
   RNNs are widely used in NLP tasks, such as language modeling, machine translation, and sentiment analysis. They can effectively model dependencies between words in a sentence.
2. **Speech Recognition:**
   RNNs can be applied to recognize spoken language and convert speech into text. Long Short-Term Memory (LSTM) networks, a type of RNN, are often used to capture long-range dependencies in audio sequences.
3. **Time Series Prediction:**
   RNNs are used in financial forecasting, stock price prediction, weather prediction, and other time series analysis tasks. They can learn patterns and dependencies in sequential data over time.
4. **Handwriting Recognition:**
   RNNs can be employed for recognizing and interpreting handwritten text. They can model the sequential nature of strokes in handwriting.

Recurrent Neural Networks (RNN)

5. Video Analysis:

RNNs are useful for video classification, action recognition, and video captioning. They can capture temporal dependencies in frames of a video sequence.

6. Healthcare Predictive Modeling:

RNNs can be applied in healthcare for predicting patient outcomes based on time-series data, monitoring vital signs, and identifying patterns in medical records.

7. Predictive Maintenance:

RNNs can be used for predicting equipment failures and maintenance needs in industries by analyzing time-series data from sensors and monitoring devices.

8. Robotics and Motion Planning:

RNNs can assist in robotic control and motion planning by learning sequences of movements. They are useful for tasks like grasping objects and navigating environments.

9. Music Generation:

RNNs can generate music sequences by learning patterns in musical data. They have been used for creating new compositions and generating melodies.

Recurrent Neural Networks (RNN)

10. DNA Sequence Analysis:

In bioinformatics, RNNs can be used for analyzing DNA sequences, predicting genetic structures, and identifying patterns related to gene expression.

11. Event Prediction in Social Media:

RNNs can analyze sequential data from social media platforms to predict events, trends, and user behavior over time.

12. Autonomous Vehicles:

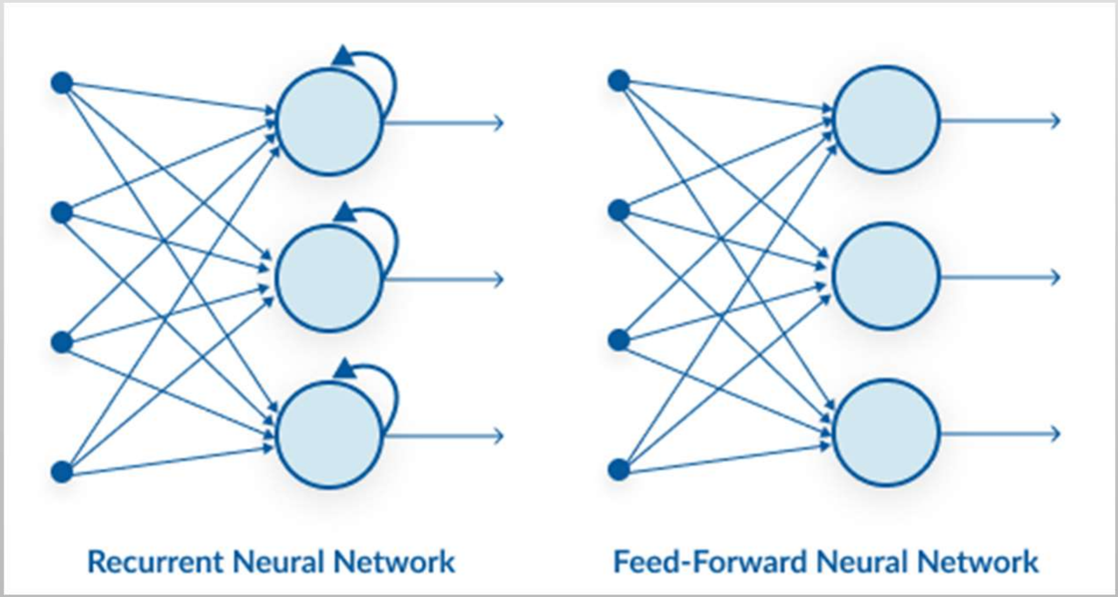
RNNs can contribute to tasks like trajectory prediction, understanding the movement of other vehicles, and predicting pedestrian behavior in autonomous vehicle systems.

13. Game Playing:

RNNs can be applied to model and predict sequences of moves in games, contributing to strategies and decision-making in games like chess or Go.

Demonstration

Demo of CNN and RNN (colab link)

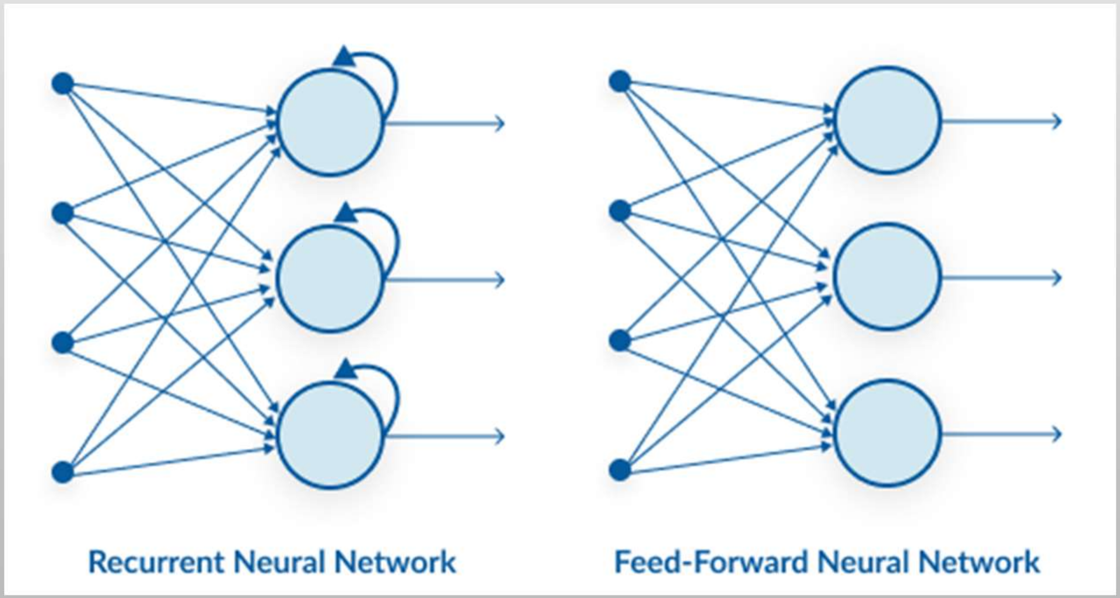


Homework

Reading material_1

Reading Material_2

Reading Material 3



References

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville published by MIT Press, 2016
2. Stanford University's Course — CS231n: Convolutional Neural Network for Visual Recognition by Prof. Fei-Fei Li, Justin Johnson, Serena Yeung
3. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939
4. https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
5. https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network
6. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn
7. https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/
8. https://towardsdatascience.com/understanding-rnns-lstms-and-grus-ed62eb584d90
9. https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e
10. https://www.cse.ust.hk/~quan/comp5421/notes/cnn.pptx
11. https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt
12. https://www.cse.iitk.ac.in/users/sigml/lec/Slides/LSTM.pdf
13. https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture6RNN.ppt

THANKS