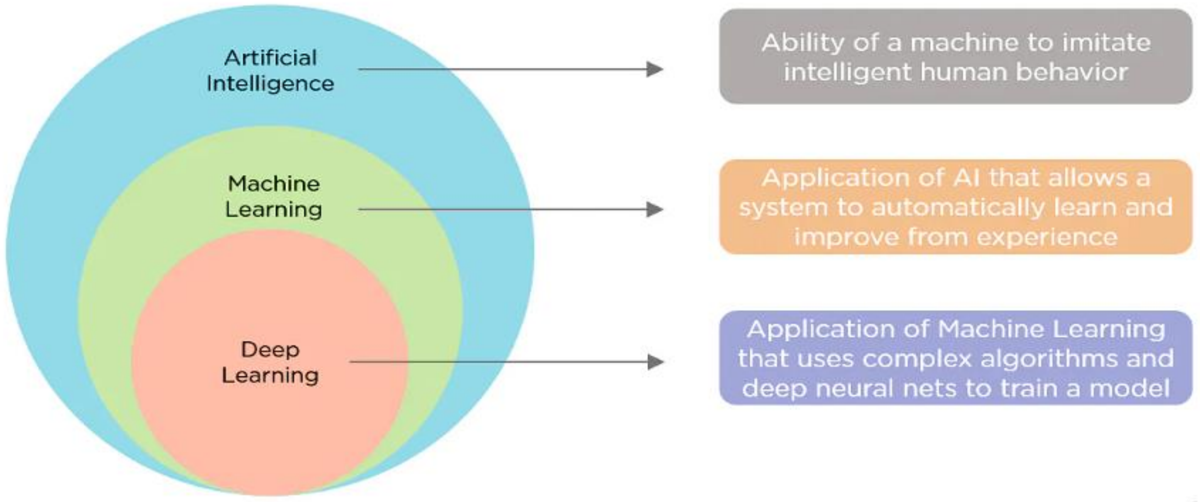
# What is Deep Learning?



Ability of a machine to imitate intelligent human behavior

Application of AI that allows a system to automatically learn and improve from experience

Application of Machine Learning that uses complex algorithms and deep neural nets to train a model

[Differences Between AI vs. Machine Learning vs. Deep Learning | Simplilearn]

# Deep Learning (contd..)

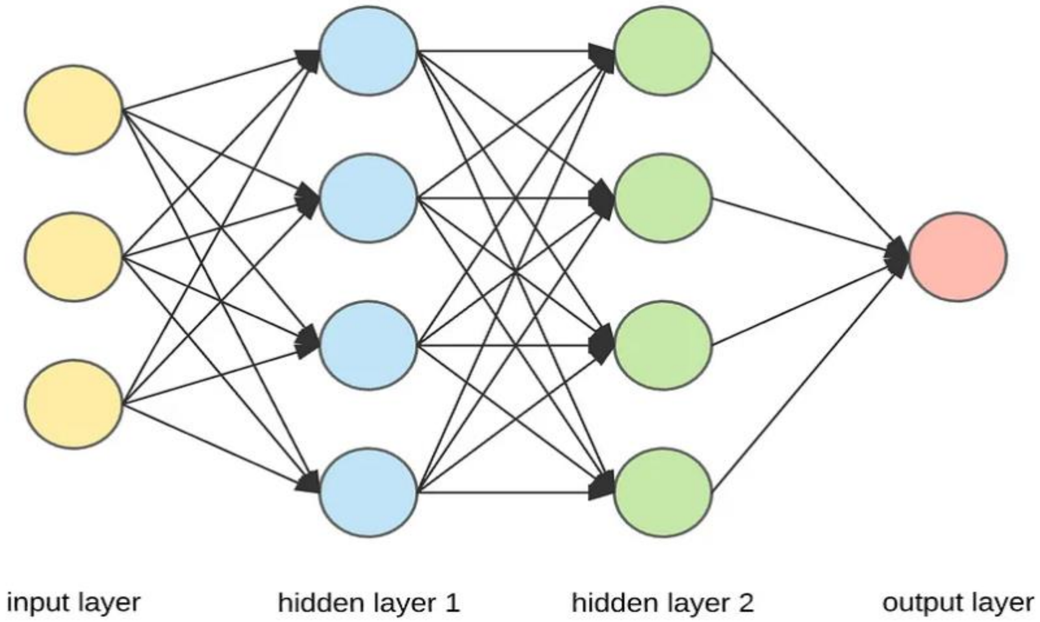
- A neural network with three or more layers is “deep”

- This neural network mimic the way human brain works with neurons

- Types of Deep Neural Networks:
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Generative Adversarial Network (GAN)

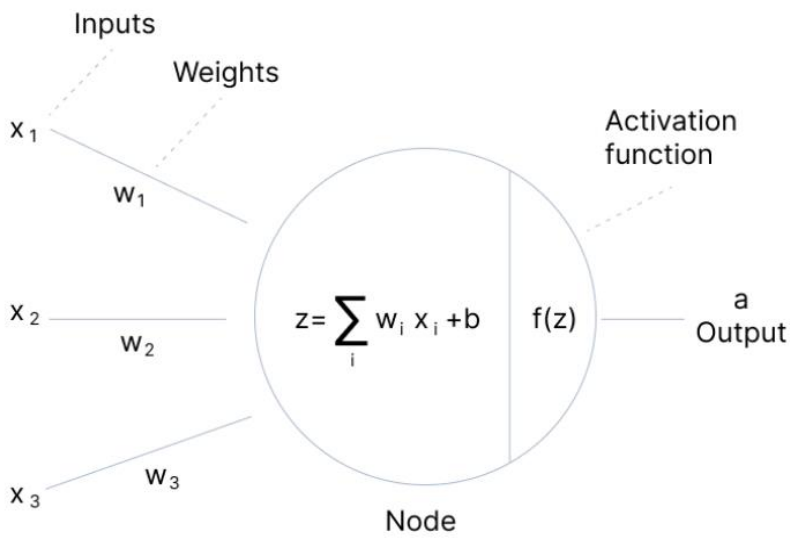
Why does deep learning have the name "deep"? (indiaai.gov.in)

# Artificial Neural Network



- Applied Deep Learning - Part 1: Artificial Neural Networks | by Arden Dertat | Towards Data Science

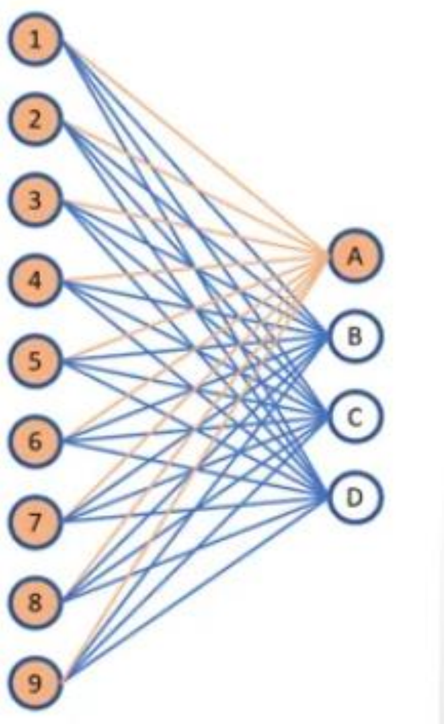
# Function of single neuron



- This weighted total is passed as an input to an “activation function” to produce the output.
- Activation functions choose whether the neuron should fire or not.

Artificial Neural Network Tutorial – Javatpoint
Activation Functions in Neural Networks [12 Types & Use Cases] (v7labs.com)

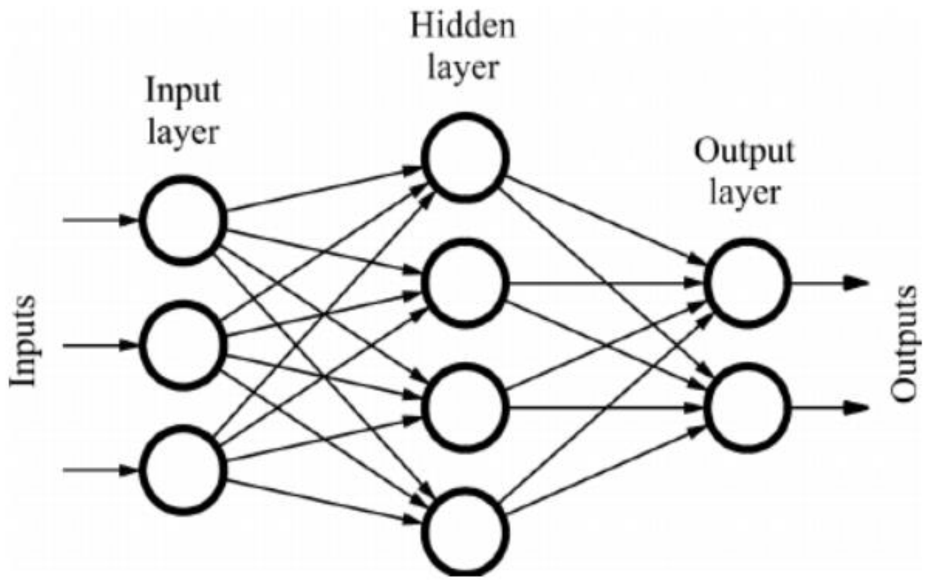
# Fully Connected Artificial Neural Network



In fully connected neural network, each output is influenced by each of the input.

Convolutional Layers vs Fully Connected Layers | by Diego Unzueta | Towards Data Science

# “Feed Forward” Neural Network and “BackPropagation” of Error



- “Feed Forward NN” is the simplest form of neural network as information is only processed in one direction.
- The initial weights are assigned randomly.
- Our job is to match the "predicted output" to the "desired output", with the given weights.
- If output is satisfactory, no changes to the weights are made.
- However, if the output does not match the desired output, then the weights need to be changed to reduce the error.
- This error is then propagated back within the whole network, one layer at a time
- The weights are updated according to the value that they contributed to the error.

- [Feed Forward Neural Network Definition | DeepAI]

# Supervised, Un-supervised and Reinforcement Learning

- Supervised Learning:
  - It functions with “data” + “labels”
  - Used for classification and prediction functions
  - Classification and Prediction can be pretty accurate and be learnt over time
  - For “Robotic Arm”, we are using supervised learning
- Un-Supervised Learning:
  - It functions with only the “data”, labels are absent
  - Used for finding hidden patterns in the data without human intervention
  - Used for “Clustering” application
  - It might give us inaccurate results, unless human intervention is done
  - It is more closer to Artificial Intelligence

Supervised vs. Unsupervised Learning: What’s the Difference? | IBM

# Supervised, Un-supervised and Reinforcement Learning

- Reinforcement Learning
  - It can be expressed in “state-action-reward” framework
  - It can be used for “long term decision making” or “repeated decision making” problems
  - Example: Vehicle Path Finding in a given environment with obstacles

- https://developer.ibm.com/learningpaths/get-started-automated-ai-for-decision-making-api/what-is-automated-ai-for-decision-making/

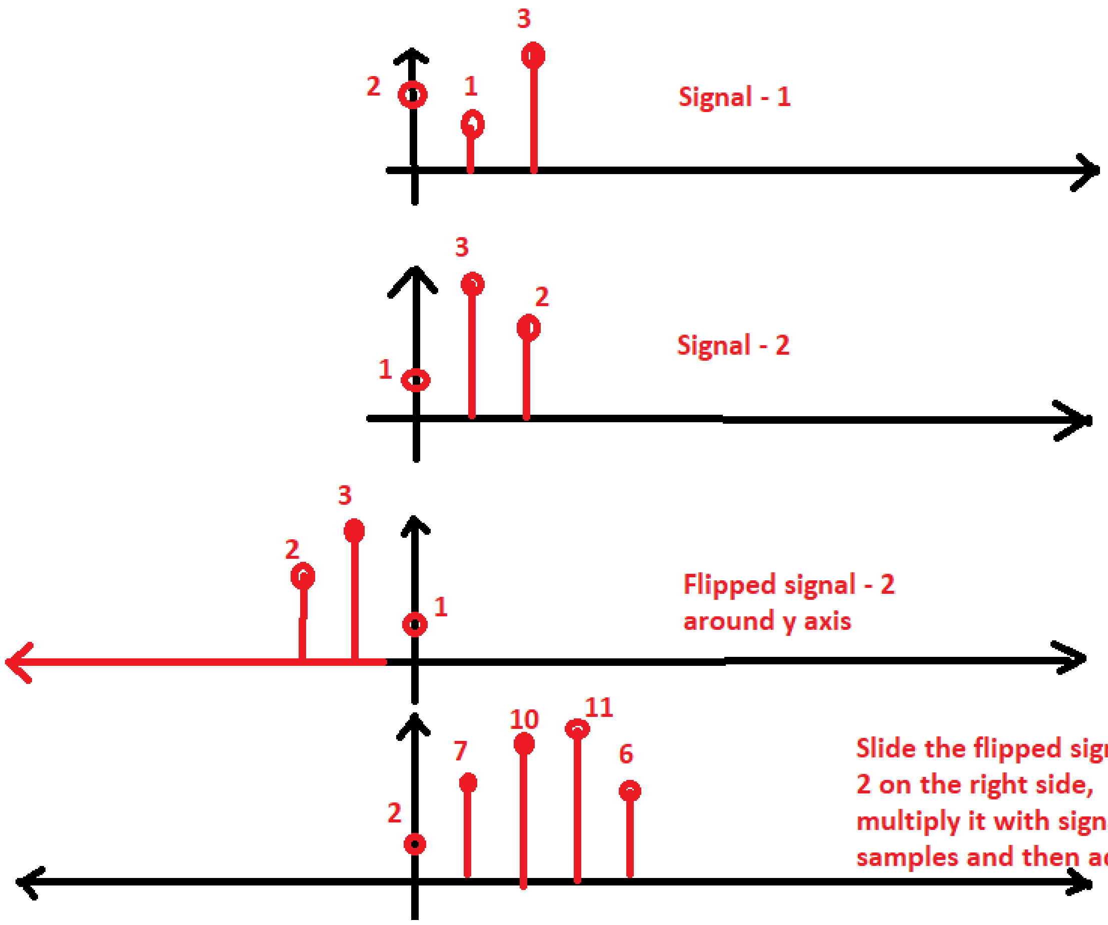
# Concept of “Convolution” in Signal Processing

- Convolution is a formal mathematical operation just as “multiplication”, “addition” and “integration”

- Addition takes two numbers and produces a third number, while convolution takes two signals and produces a third signal.

- Below is the notation of convolution
  - y[n] = x[n] * h[n]

# Example - Convolution Calculation



Slide the flipped signal 2 on the right side, multiply it with signal 1 samples and then add

1st sample = 1 x 2
2nd sample = 1 x 1 + 3 x 2
3rd sample = 1 x 3 + 3 x 1 + 2 x 2
4th sample = 3 x 3 + 2 x 1
5th sample = 3 x 2

Total sample = N1 + N2 -1 = 5

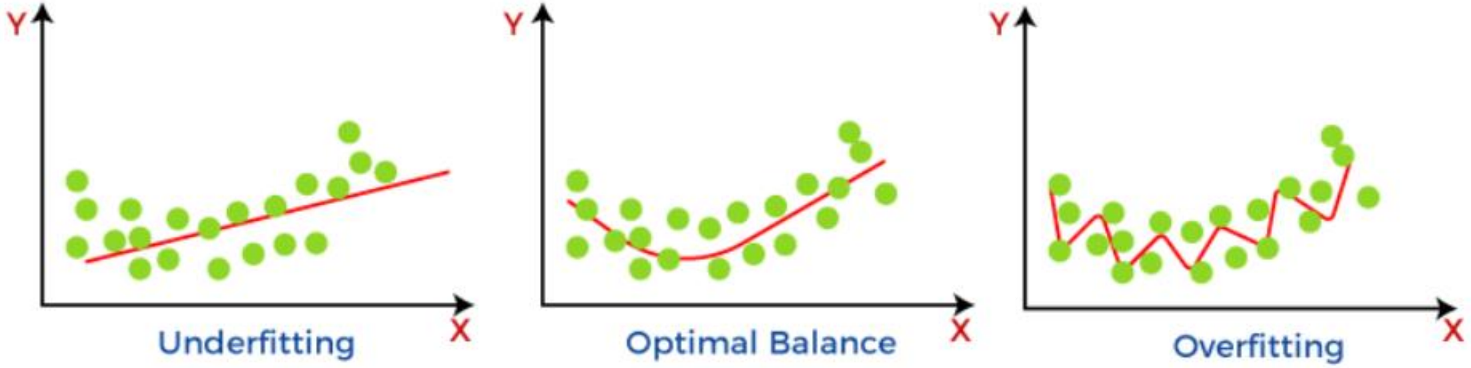
# References for Convolution

- https://uomustansiriyah.edu.iq/media/lectures/5/5_2021_11_27!10_35_40_AM.pdf
- The Scientist and Engineer's Guide to Digital Signal Processing Convolution (analog.com)
- What is the physical meaning of the convolution of two signals? - Signal Processing Stack Exchange
- Understanding Convolutions - colah's blog
- Week 4: Image Filtering and Edge Detection (sbme-tutorials.github.io)

Very interesting article - How images look after applying different kernels, especially edge detection

# OverFitting and UnderFitting

- To understand overfitting and underfitting, we need to understand “bias” and “variance”
  - Bias is the error resulted from “training set”. This error reduces the training accuracy
  - Variance is the error resulted from “testing set”. This error reduces the testing accuracy



- https://medium.com/analytics-vidhya/elucidating-bias-variance-under-fitting-and-over-fitting-273846621622
- Bias and Variance in Machine Learning - Javatpoint

# Overfitting and Underfitting (Contd.)

- Ways to reduce High Bias (Low training accuracy, low testing accuracy):
  - Increase the input features as the model is underfitted.
  - Decrease the regularization term.
  - Use more complex models, such as including some polynomial features.

- Ways to Reduce High Variance (High training accuracy, low testing accuracy):
  - Reduce the input features or number of parameters as a model is overfitted.
  - Do not use a much complex model.
  - Increase the training data.
  - Increase the Regularization term.

# TensorFlow Conv1D Layer

- How Convolutional Layers Work in Deep Learning Neural Networks? - Hong Jing (Jingles) (jinglescode.github.io)

- Conv1D
- Kernel
- Filters
- Padding
- Strides
- Epochs/Iterations: The number of times the input data is fed to the neural network, until the weights get stabilized and output accuracy reaches to expected value

# Convolution Neural Network Layer

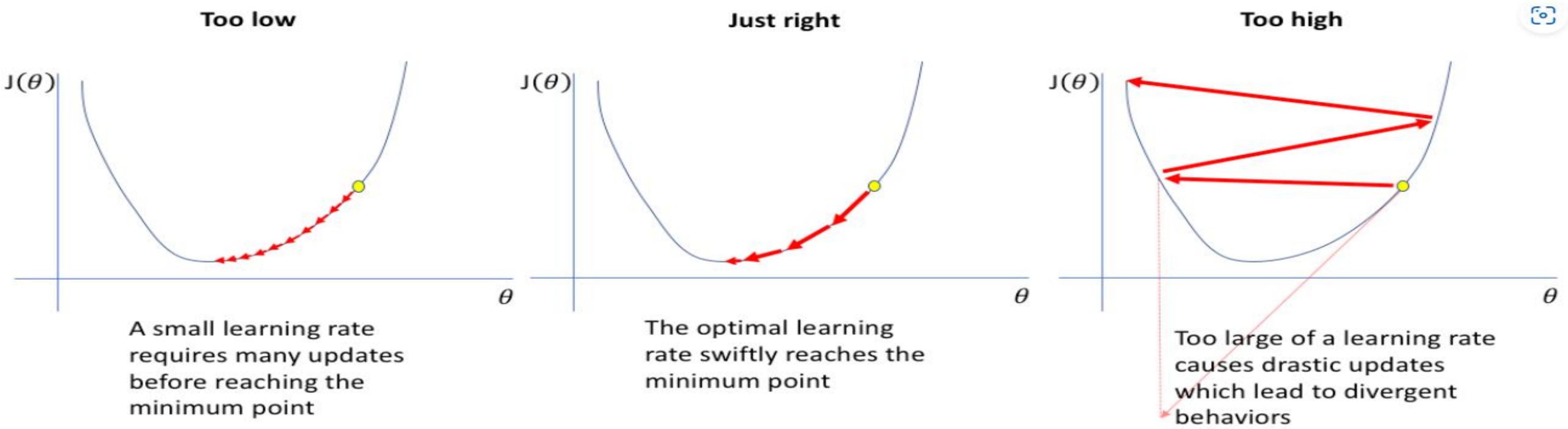
- Kernel Regularizer, Bias Regularizer, Activity (output of layer) Regularizer
  - Fine-tune the model
  - Introduce penalty, for the model to converge with minimal loss
  - Used to reduce overfitting

- Kernel Constraint – min_max_norm
  - Applies for weight matrix of the kernel
  - If weights exceed the "given value", it scales the whole weight matrix by a factor that reduces the norm to "given value"

- https://analyticsindiamag.com/kernel-regularizers-with-neural-networks/

# Convolution Neural Network

- Learning rate
  - The "learning rate" is a tuning parameter in an optimization algorithm, that determines the step size at each iteration while moving toward a minimum of a loss function.



Too low

A small learning rate requires many updates before reaching the minimum point

Just right

The optimal learning rate swiftly reaches the minimum point

Too high

Too large of a learning rate causes drastic updates which lead to divergent behaviors

[https://www.jeremyjordan.me/nn-learning-rate/](https://www.jeremyjordan.me/nn-learning-rate/)
[https://en.wikipedia.org/wiki/Learning_rate](https://en.wikipedia.org/wiki/Learning_rate)

# Convolution Neural Network

- Loss Function/Cost Function
  - Cost functions are used to estimate how badly models are performing
  - A cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and y
  - This is typically expressed as a difference or distance between the predicted value and the actual value

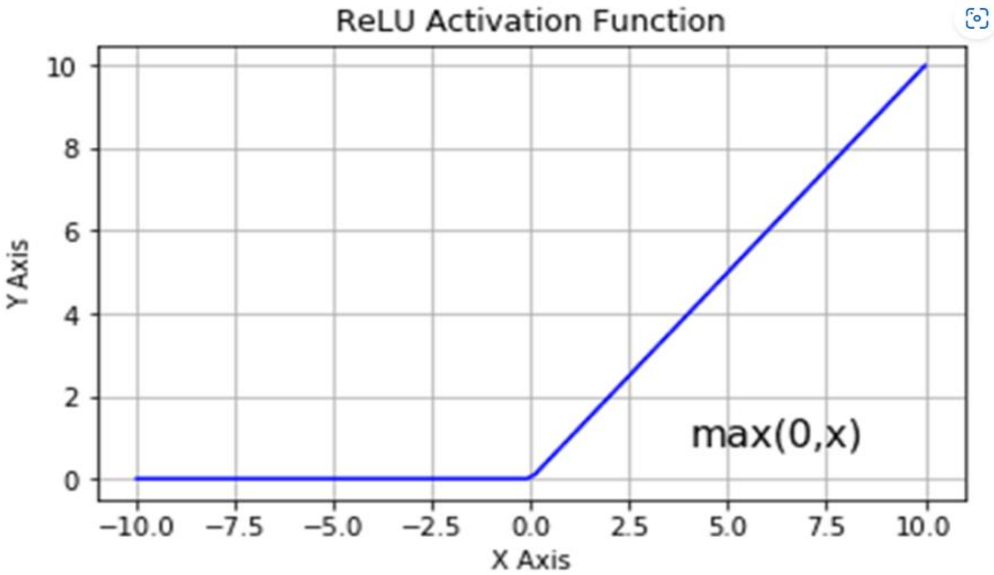
https://towardsdatascience.com/machine-learning-fundamentals-via-linear-regression-41a5d11f5220

- Activation Function
  - An Activation Function decides whether a neuron should be activated or not
  - This means that it will decide whether the neuron's input to the network is important or not, in the process of prediction
  - It is done using simpler mathematical operations

https://www.v7labs.com/blog/neural-networks-activation-functions

# Convolution Neural Network

- Activation Functions
  - ReLU: It process all positive inputs through the neuron and suppresses all negative inputs to zero. The function looks like below.

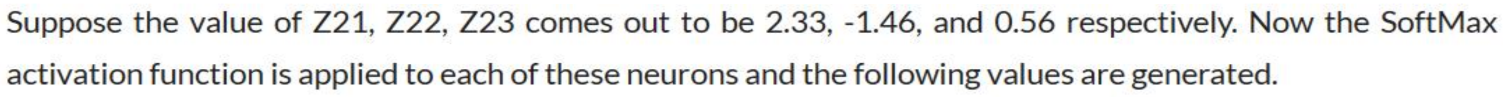


https://www.nomidl.com/deep-learning/what-is-relu-and-sigmoid-activation-function/

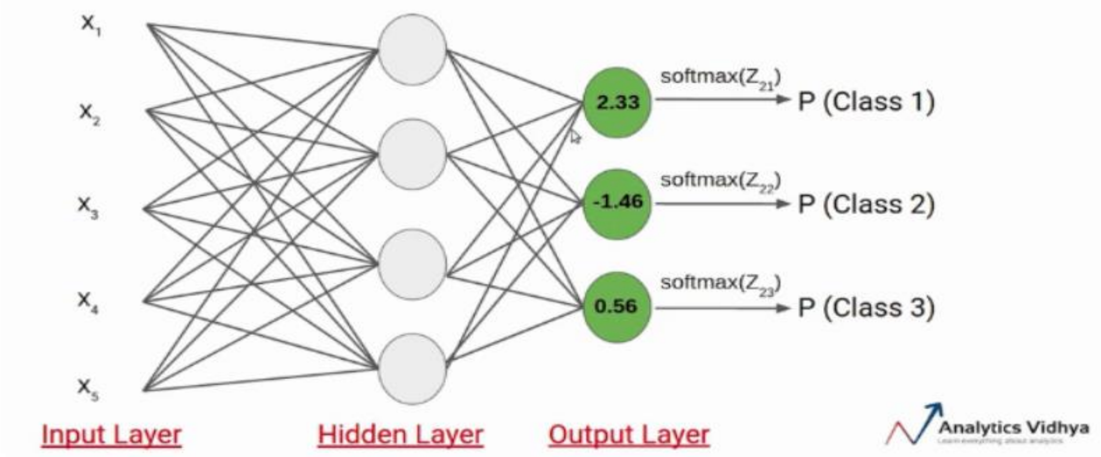
# Convolution Neural Network

- Activation Functions
  - Softmax:

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

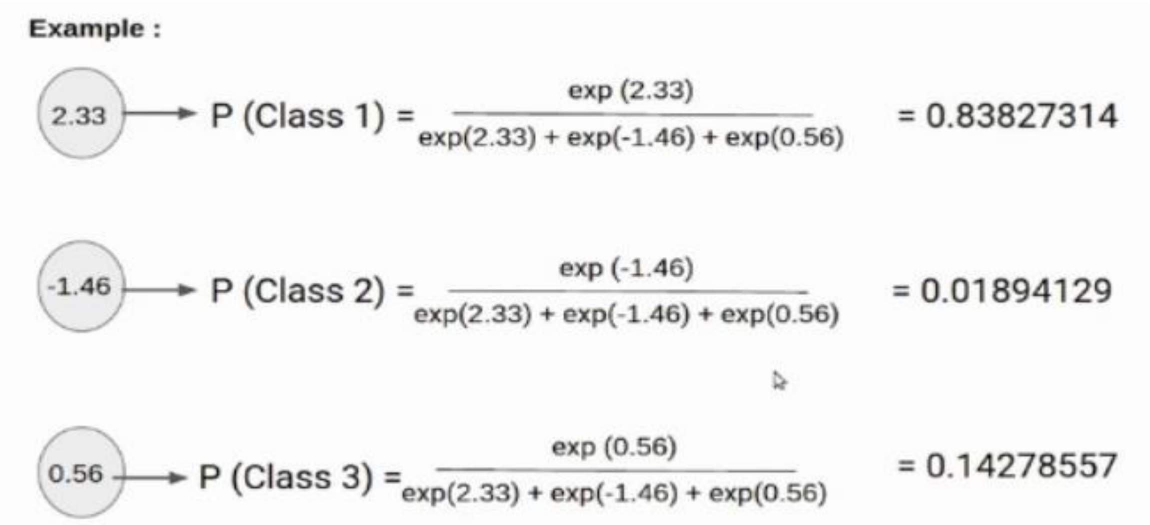


Suppose the value of Z21, Z22, Z23 comes out to be 2.33, -1.46, and 0.56 respectively. Now the SoftMax activation function is applied to each of these neurons and the following values are generated.



https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/

# Convolution Neural Network



**Example :**

$$2.33 \longrightarrow \text{P (Class 1)} = \frac{\exp\,(2.33)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)} = 0.83827314$$

$$-1.46 \longrightarrow \text{P (Class 2)} = \frac{\exp\,(-1.46)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)} = 0.01894129$$

$$0.56 \longrightarrow \text{P (Class 3)} = \frac{\exp\,(0.56)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)} = 0.14278557$$

# Convolution Neural Network

- Optimizer: Algorithm which minimizes the cost/loss function and updates the attributes of the neural network (such as weights and biases) to achieve the minimization

- Network Convergence: It is a process to get minimum error/loss and maximum accuracy

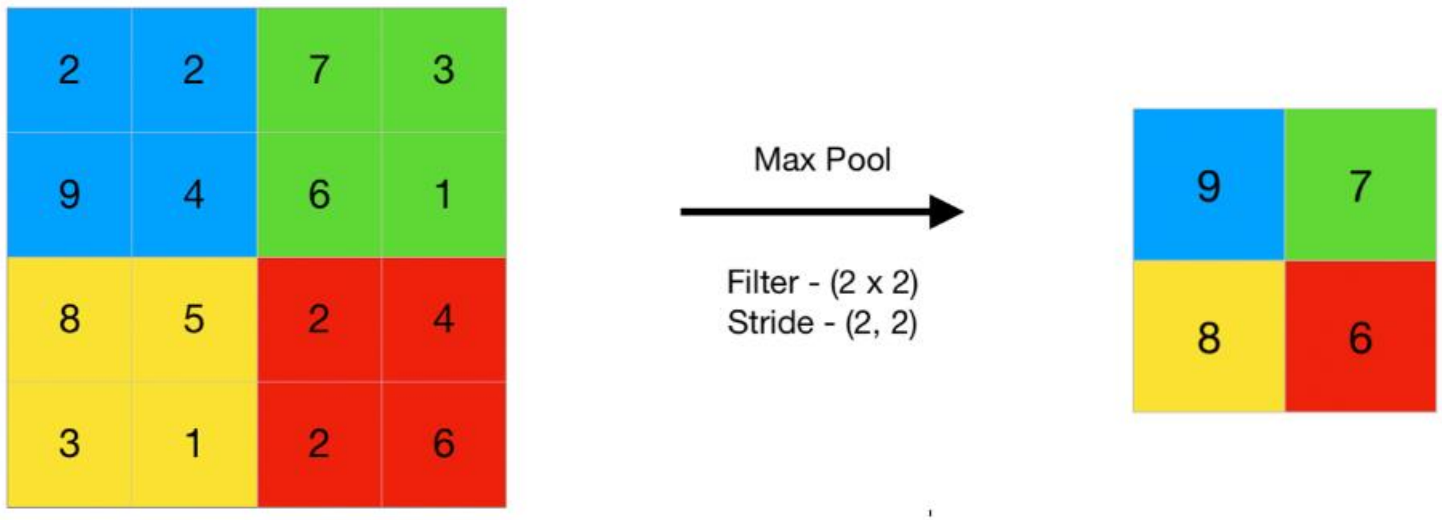
- Early stopping: Stop the training of the neural network before the number of configured iterations/epochs is reached. This is done, when the expected accuracy or loss figure is reached

- Model Checkpoint: Checkpoint for expected accuracy or loss. The model attributes at this instant can be preserved for future use of prediction

- Batch size:
  - The batch size is a number of samples processed before the model is updated.
  - The size of a batch must be more than or equal to one and less than or equal to the number of samples in the training dataset.

# Max Pooling Layer

- MaxPooling1D



|  |  |  |  |
|---|---|---|---|
| 2 | 2 | 7 | 3 |
| 9 | 4 | 6 | 1 |
| 8 | 5 | 2 | 4 |
| 3 | 1 | 2 | 6 |

Max Pool

Filter - (2 x 2)
Stride - (2, 2)

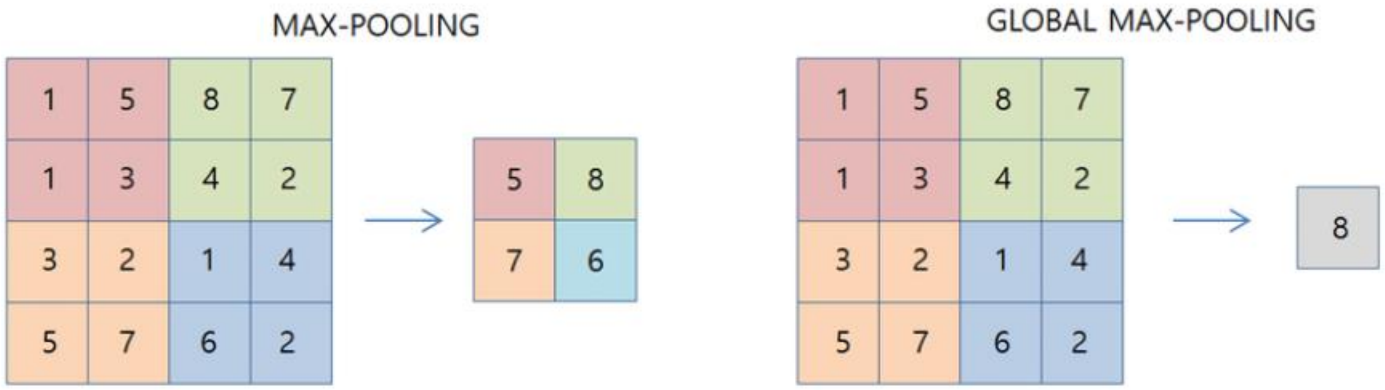
|  |  |
|---|---|
| 9 | 7 |
| 8 | 6 |

NOTE: The figure is around 2D signals (images). We need to apply it to 1D signals.

https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

# Global Max Pooling Layer

- GlobalMaxPooling1D



MAX-POOLING

| | | | |
|---|---|---|---|
| 1 | 5 | 8 | 7 |
| 1 | 3 | 4 | 2 |
| 3 | 2 | 1 | 4 |
| 5 | 7 | 6 | 2 |

→

| | |
|---|---|
| 5 | 8 |
| 7 | 6 |

GLOBAL MAX-POOLING

| | | | |
|---|---|---|---|
| 1 | 5 | 8 | 7 |
| 1 | 3 | 4 | 2 |
| 3 | 2 | 1 | 4 |
| 5 | 7 | 6 | 2 |

→

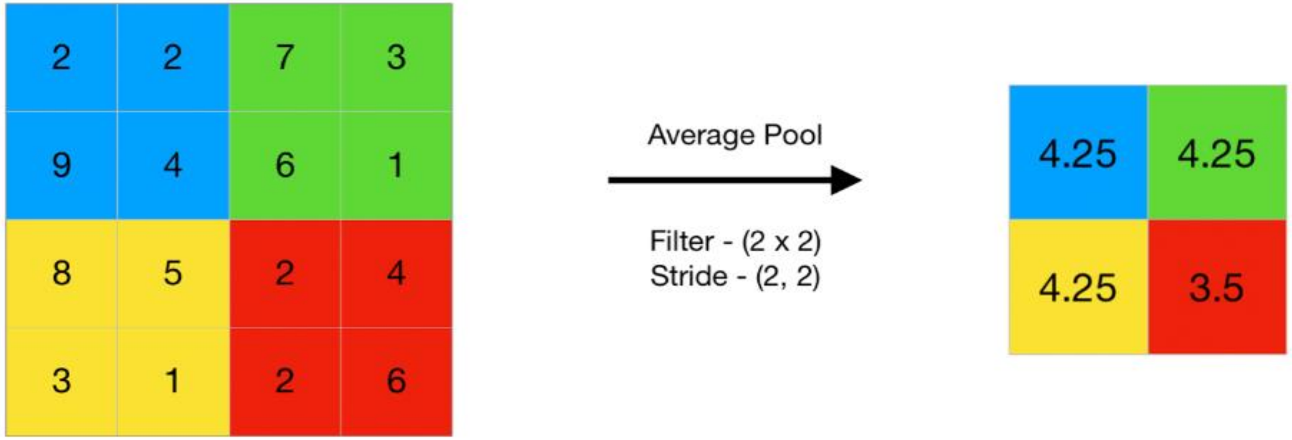
| |
|---|
| 8 |

The difference of max-pooling and global max-pooling.

NOTE: The figure is around 2D signals (images). We need to apply it to 1D signals.

# Average Pooling

- AveragePooling1D



| | | | |
|---|---|---|---|
| 2 | 2 | 7 | 3 |
| 9 | 4 | 6 | 1 |
| 8 | 5 | 2 | 4 |
| 3 | 1 | 2 | 6 |

| | |
|---|---|
| 4.25 | 4.25 |
| 4.25 | 3.5 |

1. **Code #2 : Performing Average Pooling using keras**

https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

# Global Average Pooling Layer

- Imagine a diagram, similar to Global Max Pooling slide

# Dropout Layer

- During usage of neural network with a large number of neurons, a scenario might be present where more than one neuron will extract similar features

- To avoid redundancy of features, we use Dropout

- In dropout, we randomly shut down some fraction of a layer's neurons at each training step by zeroing out the neuron values.

- The fraction of neurons to be zeroed out is known as the "dropout rate"

- Dropout reduces the size of neural network in every iteration and hence reduces overfitting

# Dense Layer

- Dense Layer is simple layer of neurons in which each neuron receives input from all the neurons of previous layer, thus called as dense.

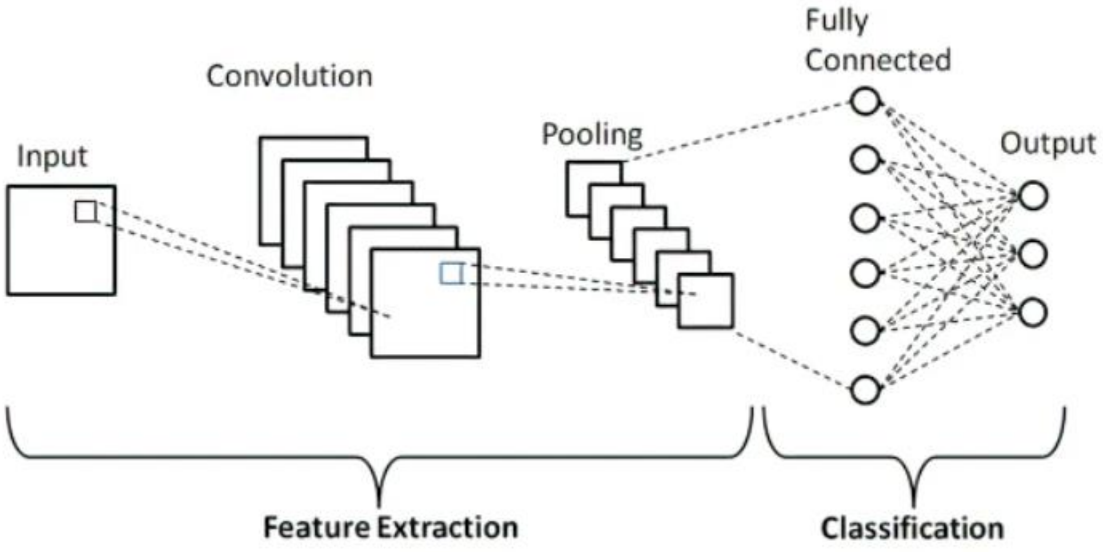
# What is “Tensor” in python?

- A tensor can be described as a n-dimensional numerical array.
- A tensor can be called a generalized matrix.
- It could be a 0-D matrix (a single number), 1-D matrix (a vector), 2-D matrix or any higher dimensional structure.
- A tensor is identified by three parameters viz., rank, shape and type.

https://www.geeksforgeeks.org/python-creating-tensors-using-different-functions-in-tensorflow/

# Convolution Neural Network - Architecture

- Convolution Layer
- Pooling Layer
- Normalization Layer
- Fully Connected Layer
- Activation Function



https://www.upgrad.com/blog/basic-cnn-architecture/

# What are “Features” when it comes to Machine Learning?

- A feature is an individual measurable property or characteristic of a phenomenon
- Choosing "informative", "discriminating" and "independent" features is a crucial element of effective algorithms in pattern recognition, classification and regression
- Briefly, "feature" is input and "label" is output
- A feature is one column of the data in your input set
- Robotic arm has 32000 features and 1 label

# Regularization

- “Regularization” refers to techniques that are used to calibrate machine learning models, in order to minimize the adjusted loss function and prevent overfitting or underfitting.

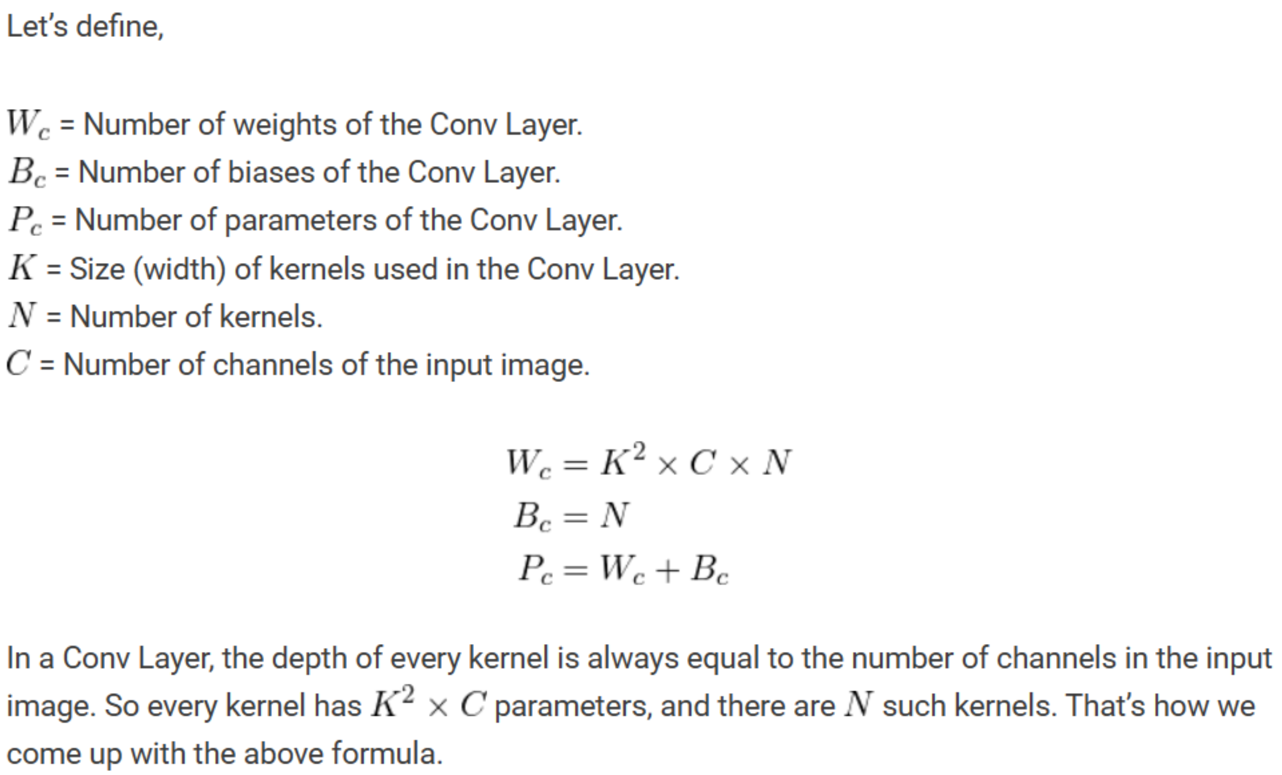
- Regularization can be implemented using a “Dropout” layer

- It can also be implemented using L1 or L2 regularization parameter while configuring Convolution Neural Network layer

https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning

# Current model of Robotic Arm

```
•  _________________________________________________________________
•  Layer (type)            Output Shape           Param #
•  =================================================================
•  conv1d (Conv1D)          (None, 248, 75)         24075

•  max_pooling1d (MaxPooling1D  (None, 82, 75)          0
•  )

•  global_average_pooling1d (G  (None, 75)             0
•  lobalAveragePooling1D)

•  dense (Dense)            (None, 32)              2432

•  dropout (Dropout)         (None, 32)              0

•  dense_1 (Dense)           (None, 3)               99

•  =================================================================
```



Let's define,

$W_c$ = Number of weights of the Conv Layer.
$B_c$ = Number of biases of the Conv Layer.
$P_c$ = Number of parameters of the Conv Layer.
$K$ = Size (width) of kernels used in the Conv Layer.
$N$ = Number of kernels.
$C$ = Number of channels of the input image.

$$
\begin{aligned}
W_c &= K^2 \times C \times N \\
B_c &= N \\
P_c &= W_c + B_c
\end{aligned}
$$

In a Conv Layer, the depth of every kernel is always equal to the number of channels in the input image. So every kernel has $K^2 \times C$ parameters, and there are $N$ such kernels. That's how we come up with the above formula.

https://learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/

NOTE: The above formula for parameter calculation is for Conv2D. For Conv1D, DO NOT calculate the square of the kernel size. Just keep the kernel size as it is.

# Introduction to LSTM (Long Short Term Memory)

- It is used in the field of Deep Learning
- These neural networks have memory
- They are capable of learning long term dependencies
- They are used especially in “sequence prediction” problems
- Some Applications
  - Handwriting Recognition
  - Video-to-text conversion

https://intellipaat.com/blog/what-is-lstm/

# Tips and Tricks

- Number of samples should be at least equal to the number of features, to create a good model
- If the data samples are less, don’t go for a more complex model
- Learning rate should be at least 10 times less than the weights and the regularization term
- Start with Python “defaults” for each parameter and then vary them, based on the output
- To begin with, adjust the “kernel size” and “learning rate” first, to see the “accuracy” and “loss” figures fall within expected limits
- GlobalMaxPooling or GlobalAveragePooling or Flatten layer is required before using Dense layers in the model
- Take a look at other models available on internet for solving different problems
- Take a look at the Tensorflow/Keras functions to find more fine-tuning parameters

# Automated Model Paramters

- Need to give an introduction of “Automated Model Parameters” file to save time