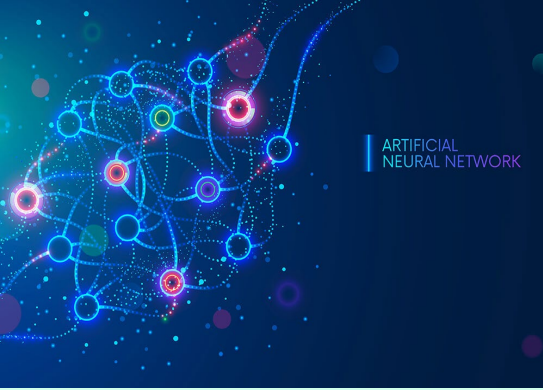




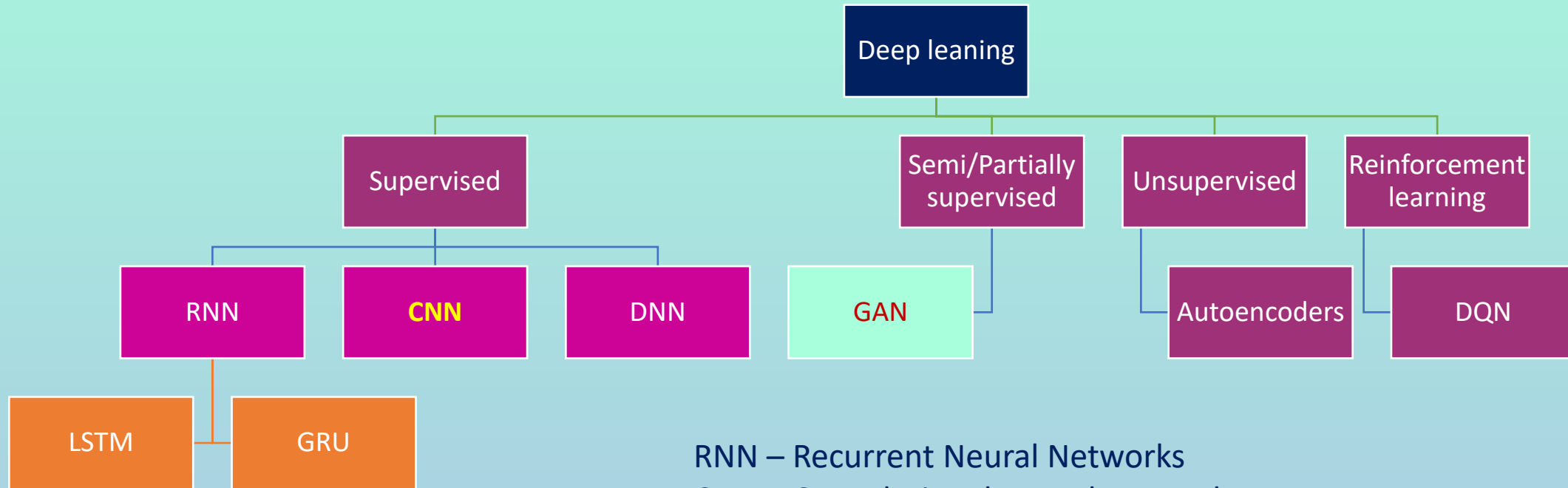
Artificial Intelligence (ACCA7026)

Artificial Neural Networks and Deep Learning

Part 2

Dr. Seena Joseph

Deep Learning



RNN – Recurrent Neural Networks

CNN – Convolutional Neural Network

DNN – Deep Neural Network

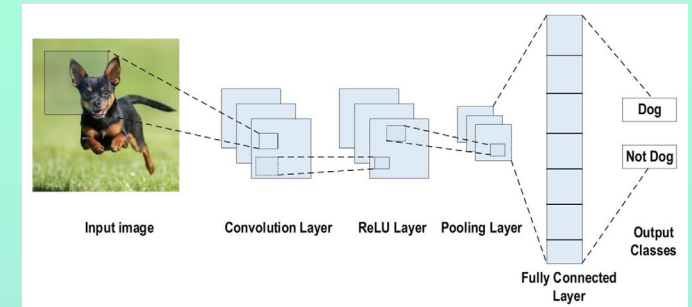
GAN – Generative Adversarial Network

DQN – Deep Q Network

LSTM – Long Short-Term Memory

GRU – Gated Recurrent Unit

CNN

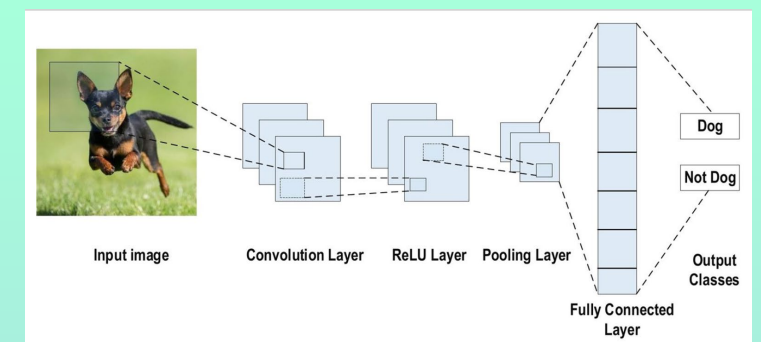


- Convolutional Neural Network
- Identify relevant features without human supervision
- Make use of 2D input data structures
- The input x of each layer in a CNN models is organised in three dimensions
 - Height, width and depth (also known as channel number) $m \times m \times r$
 - Several kernel (filters) available in each convolutional layer are denoted by k and also have three dimensions $n \times n \times q$;
 - Kernel dimension is similar to the input dimension, where the $n < m$ and $q \leq r$;
 - Kernel are the basis for the local connection and share similar parameters, bias and weight to generate k feature maps h^k
 - Convolutional layer calculates a dot product as in Equation below and then apply an activation function or non linearity to the convolutional layer

$$h^k = f(W^k * x + b^k)$$

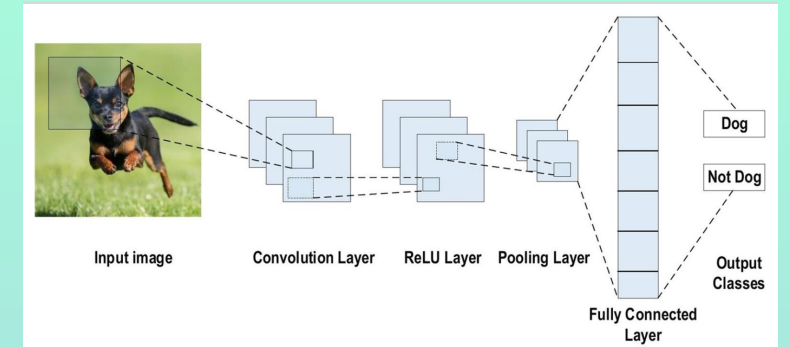
CNN

- Down sampling of feature map in the sub-sampling layer
 - This will reduce the network parameters and thereby accelerate training process and handle the overfitting issues
 - Pooling function is applied to all feature maps
- FC layer receives the mid and low level features and create high level abstraction that represents the last stage layer
- Ending layers is used to get the classification score using functions such as softmax, SVM etc;



CNN

- A CNN consists of
 - Convolutional layers
 - Pooling layers
 - Activation function
 - Fully connected layers
 - Loss functions



CNN

- CNN Layers

- Convolutional Layer

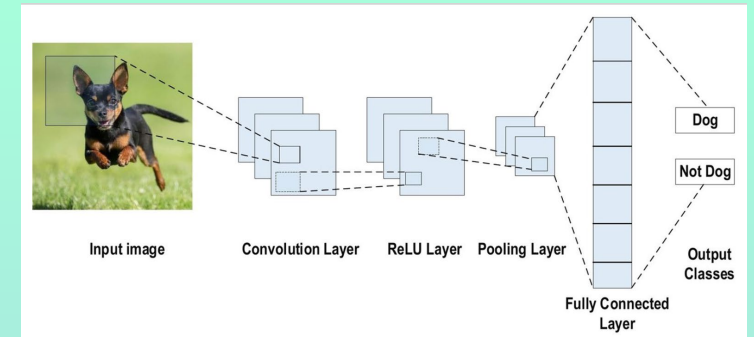
- It is the core building block of a convolutional layer.
 - Contains a collection of convolutional filters (kernels)
 - The n-dimensional input is convolved with these filters to generate the feature map

- Kernel- a grid of discrete numbers/values describe the kernel

- Each value is called as kernel weight

- Convolutional operation

- Figure 1



CNN

Kernel size area of the image

Step-1

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



0	1
-1	2



1		

Entry value to the output feature map

Kernel

Step-2

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



0	1
-1	2



1	0	

Step-3

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



0	1
-1	2



1	0	4

Step-4

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



0	1
-1	2



1	0	4
4		

Step-5

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



0	1
-1	2



1	0	4
4	1	

Source :L. Alzubaidi *et al.*, [1]

CNN

- Convolutional operation is demonstrated with an example of a 4 x 4 gray scale image with 2 x 2 kernel
- Kernel slides over the whole image vertically and horizontally
- A dot product between the input image and kernel is determined; corresponding values are multiplied and summed up to single scalar value
- The process is repeated until no further sliding is possible
- Calculated dot products are the representation of the feature maps

CNN

- The result of the convolutional layer is influenced by three key parameters.

Depth, stride and zero padding

- These parameters control the size and structure of the output after applying the convolutional filters to the image.

Depth : number of filters used in the convolutional layer.

- Each filter detects different features such as edges, shapes, texture or more complex patterns in deeper layers.
- Each filter generates a separate feature map, so the depth of the output volume is equal to the number of filters used in the convolution.
- Example: if a convolutional layer uses 32 filters, the output volume will have a depth of 32, meaning 32 feature maps are generated.

Stride : It refers to how much the filter moves (slides) over the input image during the convolution operation. The stride determines the step size for moving the filter across the input.

Zero- padding: It involves adding extra pixels (usually zeros) around the edges of the input image before applying the filter. Zero-padding ensures that the filter can fully slide across the input, even at the edges, without losing information at the boundaries.

CNN

Summarize the convolutional Layer

- Accepts an input of size $W_1 \times H_1 \times D_1$
 - Requires four hyperparameters:
 - Number of filters, K
 - their spatial extent, F
 - the stride, S
 - the amount of zero padding, P

Produces a volume of size $W_2 \times H_2 \times D_2$

where:

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1$$

(i.e. width and height are computed equally by symmetry)

$$D_2 = K$$

CNN

If the input is $32 \times 32 \times 3$ (RGB image), with a 3×3 filter ($F=3$), stride $S=1$, padding $P=1$ and 64 filters. Calculate the output dimension.

where:

$$W_2 = \frac{32 - 3 + 2 * 1}{1} + 1 = 32$$

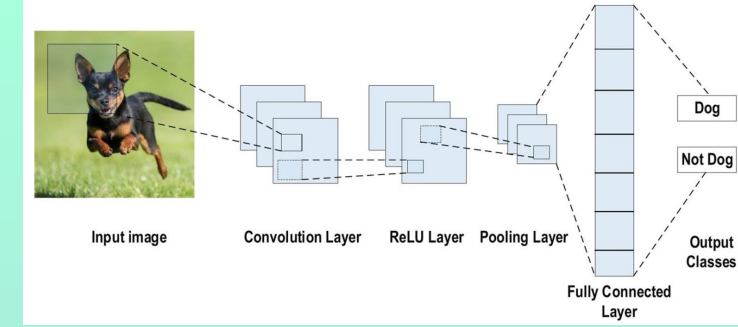
$$H_2 = \frac{32 - 3 + 2 * 1}{1} + 1 = 32$$

(i.e. width and height are computed equally by symmetry)

$$D_2 = K = 64$$

Output dimension is **$32 \times 32 \times 64$**

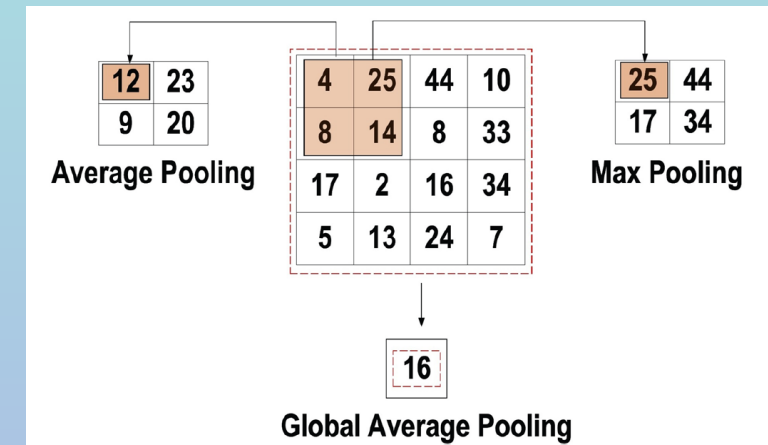
CNN



- CNN Layers

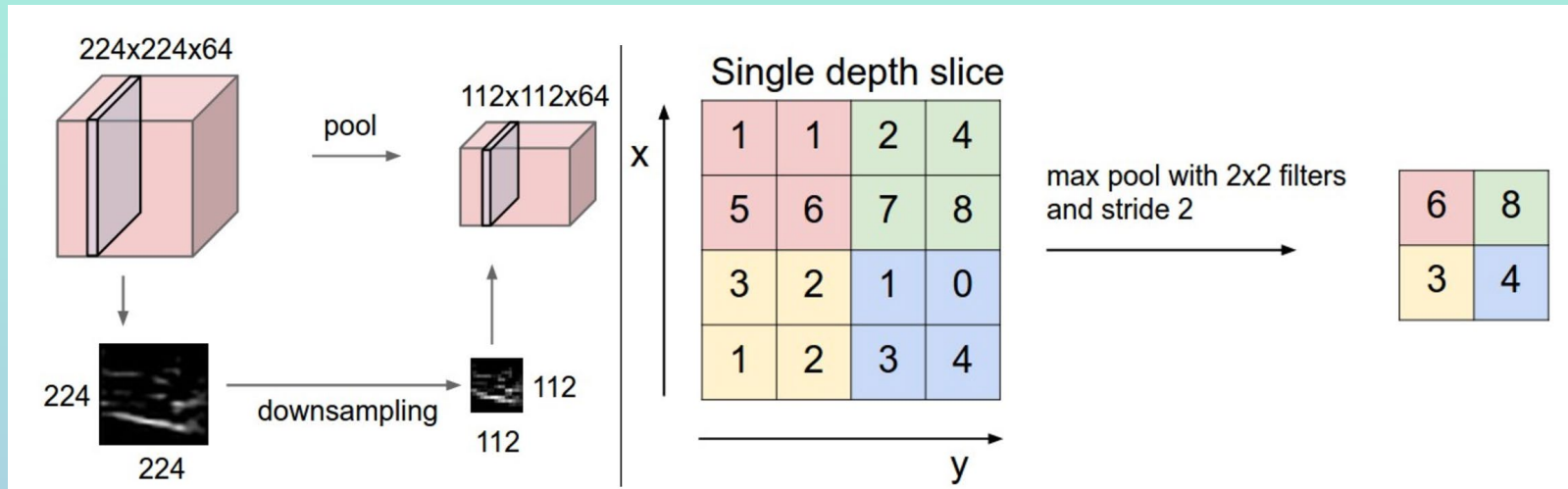
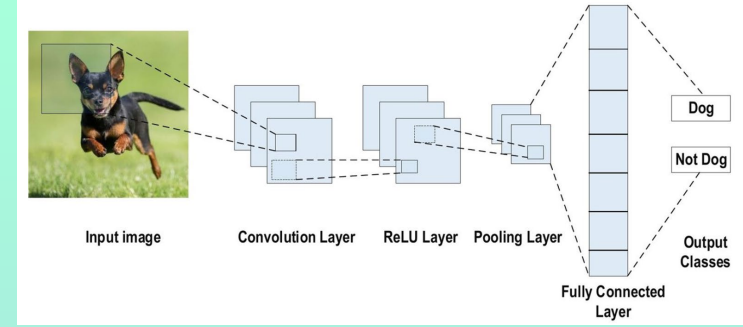
- **Pooling Layer** : Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting.

- Sub sampling of the feature map
 - Reduce the large size feature maps to create a small feature maps
 - Types of pooling methods:
 - Tree pooling, gated pooling, max pooling, min pooling, average pooling, global average pooling (GAP) and global max pooling

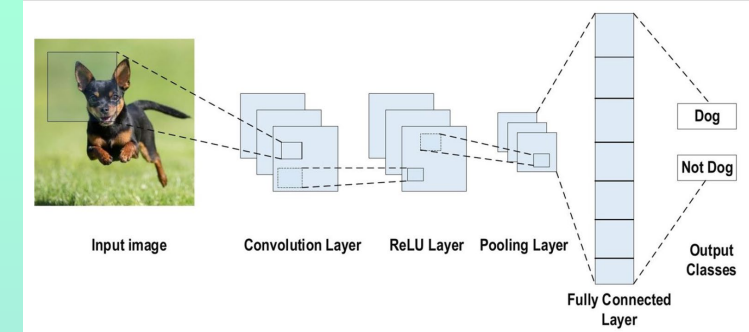


CNN

- Pooling layer



CNN



- CNN Layers

- **Activation Function**

- Non-linear activation layer is employed after all layers with weights (FC layer and Convolutional layer)
 - The function is to map the input value to output value. The input value is created by computing weighted sum of the neuron input with bias. This input value is mapped to an output value based on the activation function and this value determines whether a neuron is activated or not.

- **Fully connected layer**

- Normally located at the end of the CNN
 - Each neuron is connected to all other neurons
 - Inputs for this layer is from the last pooling or convolutional layer.
 - Utilised the CNN classifier

- **Loss function**

- Quantify or measure the error or discrepancy between the predicted output of NN and the actual target values. This error reveals the difference between the actual output and the predicted one. And it will be optimized through the CNN learning process.

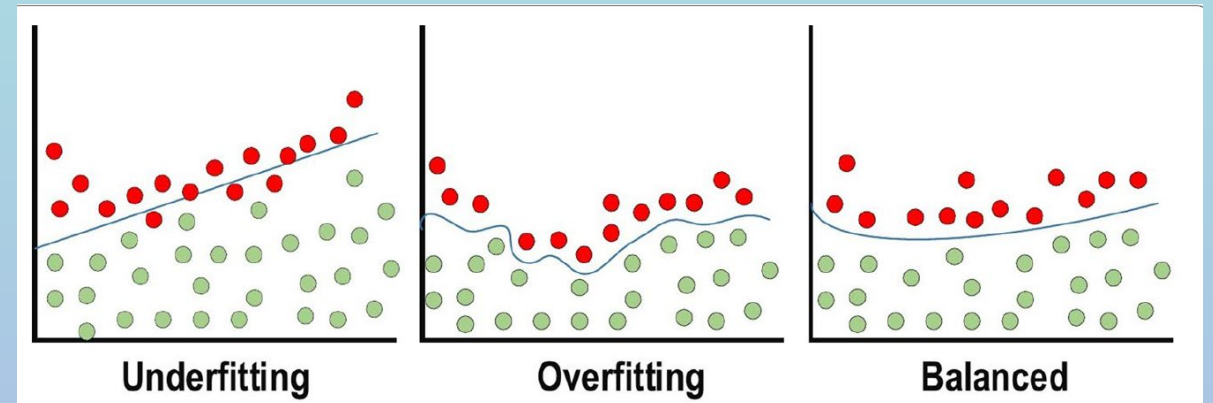
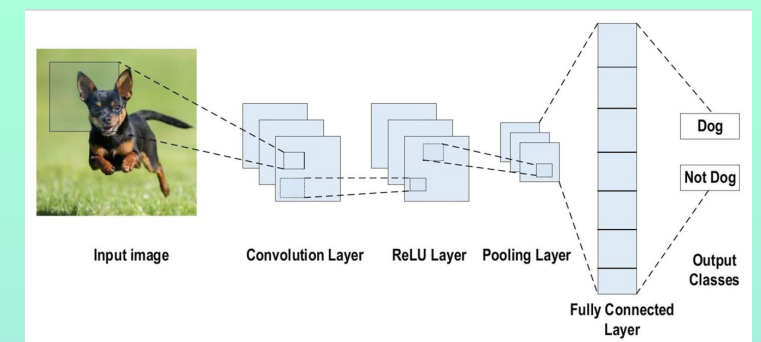
CNN

- Regularization of CNN

- Prevent overfitting and improve the generalization performance of the network

- Regularization techniques

- Dropout
 - Drop-weights
 - Data augmentation : Artificially increase the size and diversity of a training dataset by applying random transformations to the input data.

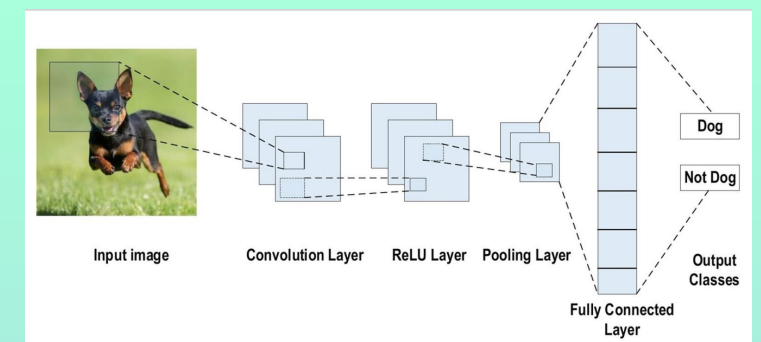


Source :L. Alzubaidi *et al.*, [1]

CNN

- CNN Learning Process

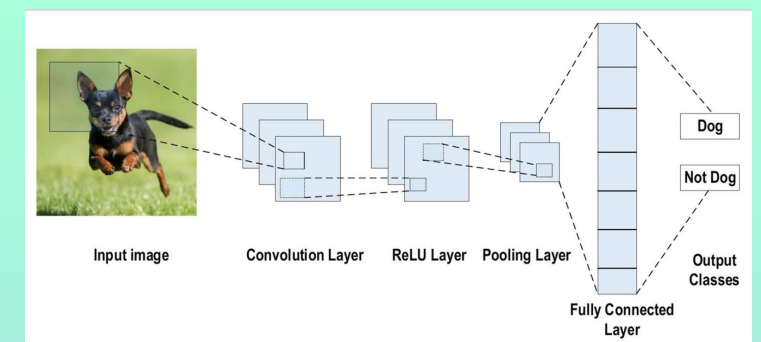
- Learning algorithm selection (optimizer) and use of enhancements
 - Learning rate : step size of the parameter update
 - Training epoch : complete repetition of the parameter update
 - Gradient descent or Gradient based learning algorithm
 - Batch gradient Descent
 - Stochastic gradient descent
 - Mini-batch gradient descent
 - Momentum
 - Adaptive momentum estimation (Adam)



CNN

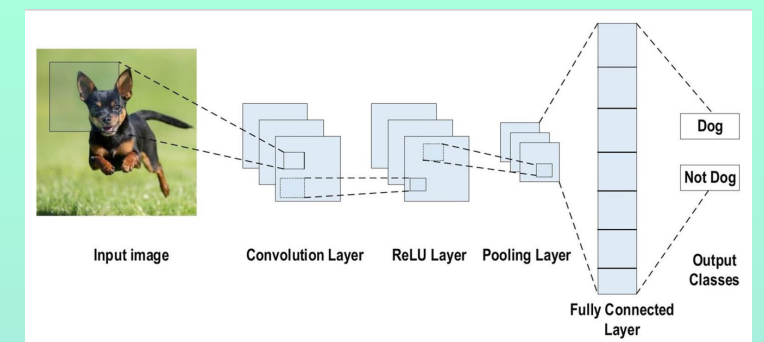
- Benefits of CNN

- Weight sharing feature that reduces the number of network parameters ; help generalisation and also helps in handling overfitting
- Concurrently learning the feature extraction layers and the classification layer causes the model output to be both highly organized and highly reliant on the extracted features.
- Large scale network implementation is much easier with CNN



CNN

- Practical DEMO
 - CNN Practical – Jupyter Note



CNN

- CNN Architecture

Model	Main finding	Depth	Dataset	Error rate	Input size	Year
AlexNet	Utilizes Dropout and ReLU	8	ImageNet	16.4	$227 \times 227 \times 3$	2012
NIN	New layer, called 'mlpconv', utilizes GAP	3	CIFAR-10, CIFAR-100, MNIST	10.41, 35.68, 0.45	$32 \times 32 \times 3$	2013
ZfNet	Visualization idea of middle layers	8	ImageNet	11.7	$224 \times 224 \times 3$	2014
VGG	Increased depth, small filter size	16, 19	ImageNet	7.3	$224 \times 224 \times 3$	2014
GoogLeNet	Increased depth, block concept, different filter size, concatenation concept	22	ImageNet	6.7	$224 \times 224 \times 3$	2015

Source :L. Alzubaidi *et al.*, [1]

CNN

• CNN Architecture

Model	Main finding	Depth	Dataset	Error rate	Input size	Year
Inception-V3	Utilizes small filtersize, better feature representation	48	ImageNet	3.5	$229 \times 229 \times 3$	2015
Highway	Presented the multipath concept	19, 32	CIFAR-10	7.76	$32 \times 32 \times 3$	2015
Inception-V4	Divided transform and integration concepts	70	ImageNet	3.08	$229 \times 229 \times 3$	2016
ResNet	Robust against overfitting due to symmetry mapping-based skip links	152	ImageNet	3.57	$224 \times 224 \times 3$	2016
Inception-ResNet-v2	Introduced the concept of residual links	164	ImageNet	3.52	$229 \times 229 \times 3$	2016
FractalNet	Introduced the concept of Drop-Path as regularization	40,80	CIFAR-10	4.60	$32 \times 32 \times 3$	2016
			CIFAR-100	18.85		
WideResNet	Decreased the depth and increased the width	28	CIFAR-10	3.89	$32 \times 32 \times 3$	2016
			CIFAR-100	18.85		

CNN

- CNN Architecture

Model	Main finding	Depth	Dataset	Error rate	Input size	Year
Xception	A depthwise convolution followed by a pointwise convolution	71	ImageNet	0.055	$229 \times 229 \times 3$	2017
Residual attention neural network	Presented the attention technique	452	CIFAR-10, CIFAR-100	3.90, 20.4	$40 \times 40 \times 3$	2017
Squeeze-and-excitation networks	Modeled inter-dependencies between channels	152	ImageNet	2.25	$229 \times 229 \times 3$ $224 \times 224 \times 3$ $320 \times 320 \times 3$	2017
DenseNet	Blocks of layers; layers connected to each other	201	CIFAR-10, CIFAR-100, ImageNet	3.46, 17.18, 5.54	$224 \times 224 \times 3$	2017
Competitive squeeze and excitation network	Both residual and identity mappings utilized to rescale the channel	152	CIFAR-10 CIFAR-100	3.58 18.47	$32 \times 32 \times 3$	2018

CNN

- CNN Architecture

Model	Main finding	Depth	Dataset	Error rate	Input size	Year
MobileNet-v2	Inverted residual structure	53	ImageNet	–	$224 \times 224 \times 3$	2018
CapsuleNet	Pays attention to special relationships between features	3	MNIST	0.00855	$28 \times 28 \times 1$	2018
HRNetV2	High-resolution representations	–	ImageNet	5.4	$224 \times 224 \times 3$	2020

- Architecture in 2021, 2022, 2023 and 2024 ?

Reference

- [1] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, pp. 1–74, 2021.