Convolutional Neural Networks (CNN): Supervised Learning

A. Supervised and unsupervised learning:

Supervised learning: As the name indicates, has the presence of a supervisor as a teacher. Basically supervised learning is when we teach or train the machine using data that is well labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of example (data) so that the supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labelled data.

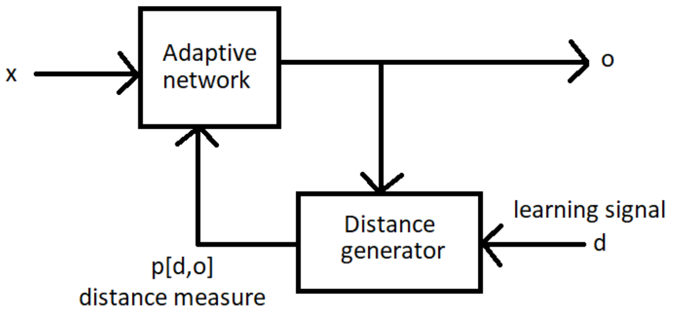
Unsupervised learning is the training of a machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.

Parameters	Supervised machine learning	Unsupervised machine learning
Input Data	Algorithms are trained using labelled data.	Algorithms are used against data that is not labelled
Accuracy	Highly accurate	Less accurate
Output	Desired output is given.	Desired output is not given.
Training data	Use training data to infer model.	No training data is used.
Complex model	It is not possible to learn larger and more complex models with supervised learning.	It is possible to learn larger and more complex models with unsupervised learning.
Called as	Supervised learning is also called classification.	Unsupervised learning is also called clustering.

Supervised learning:
Feedback is an important part in the process of learning. It exhibits the relationship pattern in the cause-and-effect path. The concept is highly elusive and somewhat paradoxical supervised learning it is assumed that at each instant of time when the input is applied, the desired response d of the system is provided by the teacher.

In the input patterns classification problems, the error can be used to modify weights.

Due to this the error decreases. A training set which is a set of input and output patterns is required for this mode of learning. If the classifications or associations are accurate then a reward is yield in the supervised learning and there is a penalty for yielding inaccurate responses. The direction of negative error gradient is estimated by the teacher and error is reduced.

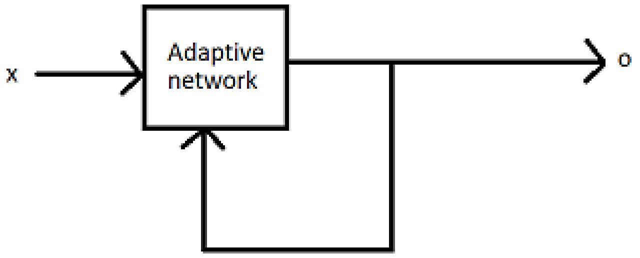


Unsupervised learning:

The technique can be differentiated from the unsupervised learning as follows. In learning without supervision, the desired response is unknown; hence, the explicit error information is not used to improve the network behaviour. No information is available to know the correctness of incorrectness of responses, hence learning must be done on the basis of observations of responses to inputs. There is a marginal or no knowledge available about these responses.

Unsupervised learning technique is often used to perform clustering.

Sometimes it is said that in unsupervised learning technique there is no presence of a teacher. But it is not that appropriate, as learning without a teacher is at all not possible. The learning mode is unsupervised where the teacher sets the goals but does not remain involved in every training step



In simple terms, in supervised learning Feedback is provided throughout.

It exhibits relationship pattern in the cause-and-effect path.

In supervised learning, every time an input is applied, the desired response of the system is provided by the teacher.

In unsupervised learning where there is no supervision, the desired response is unknown; the explicit error information is not used to improve the network behaviour.

Supervised and unsupervised learning are a part of deep learning.

Classification of deep learning:

Supervised Learning

- Artificial Neural Network
- Recurrent Neural Network

- Convolutional Neural Network

Unsupervised learning

- Self-organizing maps
- Boltzmann Machine
- Autoencoders

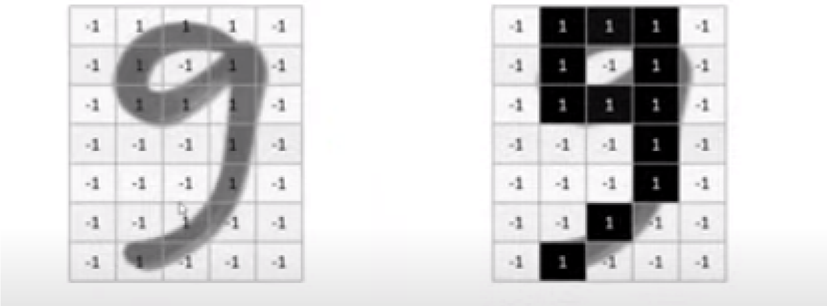
B. <u>Introduction to CNN:</u>

Convolutional Networks (LeCun, 1989), are also known as Convolutional Neural Networks or CNNs sometimes also called ConvNets.

It is a feed-forward neural network as the information moves from one layer to the next. It consists of hidden layers having convolution and pooling functions in addition to the activation function for introducing non-linearity.

- Used for image recognition and processing.
- Detect features in an image, such as edges, corners, and textures.
- Computers see an input image as an array of pixels.
- The filters are applied to small regions of the image, and as the filters move across the image, they create a set of feature maps that capture different aspects of the image.
- These feature maps are then fed into a series of convolutional layers, which combine the feature maps to create more complex representations of the image.
- Finally, the output of the convolutional layers is fed into one or more fully connected layers, which perform classification or regression tasks based on the features learned from the image.

Computer Vision:



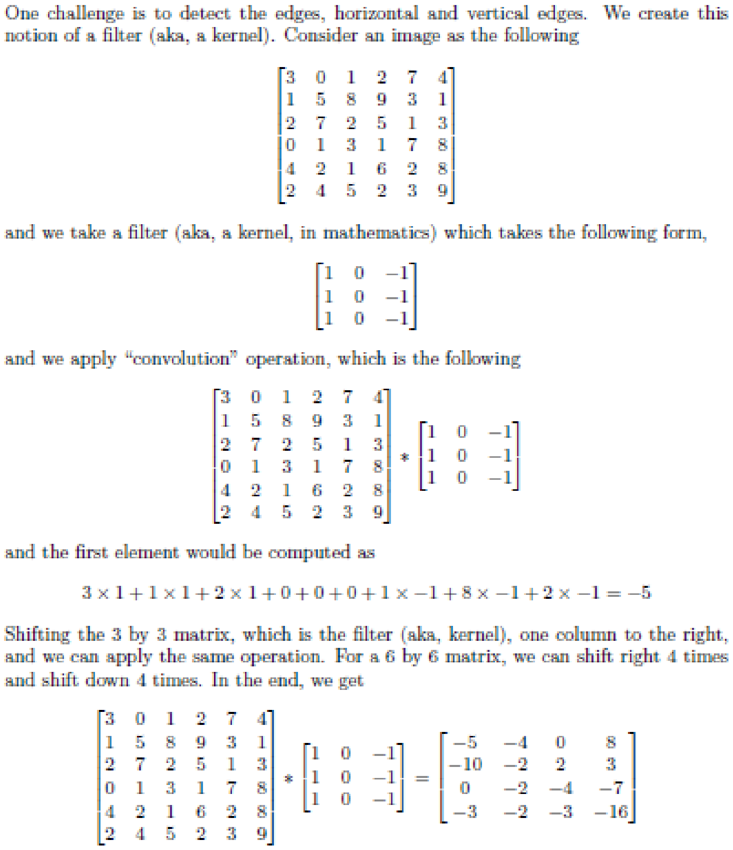
-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

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

- In computer vision, images are typically represented as a grid of pixels. Each pixel represents a small part of the image and is assigned a value that corresponds to its color or grayscale intensity.
- The pixels in an image are arranged in rows and columns to form a grid. The number of rows and columns in the grid depend on the size and resolution of the image. For example, an image with a resolution of 1280x720 has 1280 columns and 720 rows of pixels.
- When a computer processes an image, it reads the pixel values in the grid and uses them to analyze and manipulate the image. In a CNN, the filters are applied to small regions of the image, which correspond to a small set of adjacent pixels. By analyzing the values of these pixels, the filters can detect features such as edges, corners, and textures.

Convolution networks eg:

CNNs are a specialized kind of neural networks for processing data with a known, grid-like topological arrangement. For example, a time-series data, is like a single dimensional grid that collect samples at regular intervals of time and image data, is a two dimensional grid of pixels.



One challenge is to detect the edges, horizontal and vertical edges. We create this notion of a filter (aka, a kernel). Consider an image as the following

$$\begin{bmatrix} 3 & 0 & 1 & 2 & 7 & 4 \\ 1 & 5 & 8 & 9 & 3 & 1 \\ 2 & 7 & 2 & 5 & 1 & 3 \\ 0 & 1 & 3 & 1 & 7 & 8 \\ 4 & 2 & 1 & 6 & 2 & 8 \\ 2 & 4 & 5 & 2 & 3 & 9 \end{bmatrix}$$

and we take a filter (aka, a kernel, in mathematics) which takes the following form,

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

and we apply "convolution" operation, which is the following

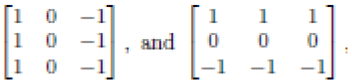
$$\begin{bmatrix} 3 & 0 & 1 & 2 & 7 & 4 \\ 1 & 5 & 8 & 9 & 3 & 1 \\ 2 & 7 & 2 & 5 & 1 & 3 \\ 0 & 1 & 3 & 1 & 7 & 8 \\ 4 & 2 & 1 & 6 & 2 & 8 \\ 2 & 4 & 5 & 2 & 3 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

and the first element would be computed as

$$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 + 0 + 0 + 1 \times -1 + 8 \times -1 + 2 \times -1 = -5$$

Shifting the 3 by 3 matrix, which is the filter (aka, kernel), one column to the right, and we can apply the same operation. For a 6 by 6 matrix, we can shift right 4 times and shift down 4 times. In the end, we get

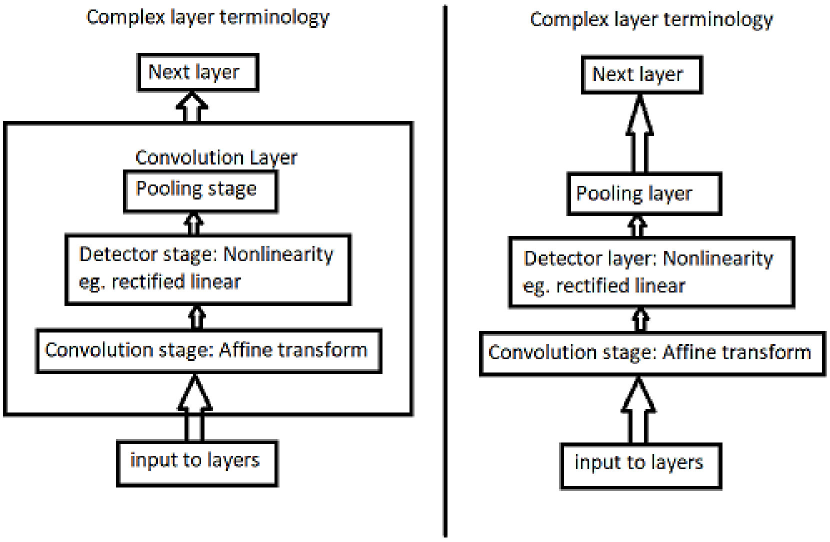
$$\begin{bmatrix} 3 & 0 & 1 & 2 & 7 & 4 \\ 1 & 5 & 8 & 9 & 3 & 1 \\ 2 & 7 & 2 & 5 & 1 & 3 \\ 0 & 1 & 3 & 1 & 7 & 8 \\ 4 & 2 & 1 & 6 & 2 & 8 \\ 2 & 4 & 5 & 2 & 3 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -5 & -4 & 0 & 8 \\ -10 & -2 & 2 & 3 \\ 0 & -2 & -4 & -7 \\ -3 & -2 & -3 & -16 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \text{ and } \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}.$$

Vertical and horizontal filters(Kernel).

CNN first learns to recognize the components of an image and then combine these components (pooling) to recognize the larger structure of the whole image.


Complex layer terminology
Next layer
Convolution Layer
Pooling stage
Detector stage: Nonlinearity eg. rectified linear
Convolution stage: Affine transform
input to layers
Complex layer terminology
Next layer
Pooling layer
Detector layer: Nonlinearity eg. rectified linear
Convolution stage: Affine transform
input to layers

The components of typical CNN layer

There are two commonly used sets of terminology for describing these layers.

Complex layer: In this terminology, the convolutional net is viewed as a small number of relatively complex layers, with each layer having many "stages." In this terminology, there is a one-to-one mapping between kernel tensors and network layers.

Simple layer: In this terminology, the convolutional net is viewed as a larger number of simple layers; every step of processing is regarded as a layer in its own.

4.1 Motivation behind Convolution:

In mathematics and signal processing, convolution is a mathematical operation on two functions that produces a third function, which expresses how the shape of one function is modified by the other.

It is a specialized type of linear operation.

Properties:

1. Sparse interaction
2. Parameter sharing
3. Equivariant representation

Parameter sharing is a feature detector (such as a vertical edge detector) that is useful in one part of the image is probability useful in another part of the image. Sparse interaction means that, in each layer, each output value depends only a small number of inputs.
The layers of convolution neural network will exhibit a property of equivariance to translation. That means if we changed the input in a way, the output will also get changed in the same way.

4.2 The convolution operation

Convolution is an operation on two functions of a real-valued argument.

Suppose a location of a spaceship is tracked with a laser sensor. The laser sensor provides a single output x(t), indicating the position of the spaceship at time t.

Both x and t are real-valued, such that we can get a different reading from the laser sensor at any instant in time. Suppose that laser sensor is somewhat noisy.

To obtain a less noisy estimate of the spaceship's position several measurements will need to be averaged. More recent measurements are more relevant, so it should be a weighted average that gives more weight to recent measurements.

This can be done with a weighting function w(a), where a is the age of a measurement. If such a weighted average operation is applied at every moment, a new function s is obtained providing a smoothed estimate of the position of the spaceship.

s(t) = x(a) w(t-a) da

This operation is called as convolution. The convolution operation is typically denoted with an asterisk:

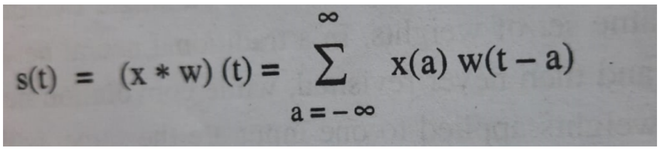
s(t) = (x*w) (t)

Here, w needs to be a valid probability density function, or the output is not a weighted average Also, w needs to be 0 for all negative arguments, or it will look into the future, which is presumably beyond our capabilities.

These limitations are particular to this example.

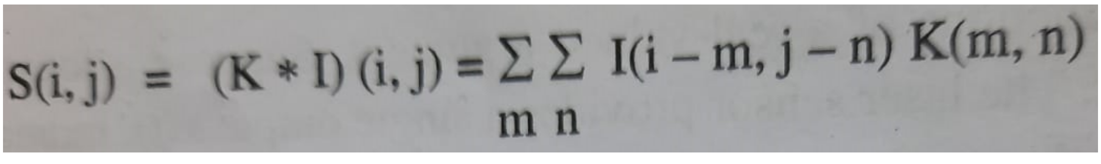
In convolutional network terminology, the first argument (in this example, the function x) to the convolution is often referred to as the input and the second argument (in this example, the function w) as the kernel. The output is sometimes referred to as the feature map.

Usually, when we work with data on a computer, time will be discretized, and the sensor will provide data at regular intervals. In this example, it might be more realistic to assume that the laser provides a measurement once per second. The time index t can then take on only integer values. If it is assumed that x and w are defined only on integer t, discrete convolution can be defined as follows:



$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)\, w(t-a)$$

We often use convolutions over more than one axis at a time. For example, if we use a two- dimensional image I as our input, we probably also want to use a two-dimensional kernel K:



$$S(i, j) = (K * I)(i, j) = \sum_{m} \sum_{n} I(i-m, j-n)\, K(m, n)$$

Convolution is commutative (changing the order of the operands does not change the final result.), meaning we can equivalently write:

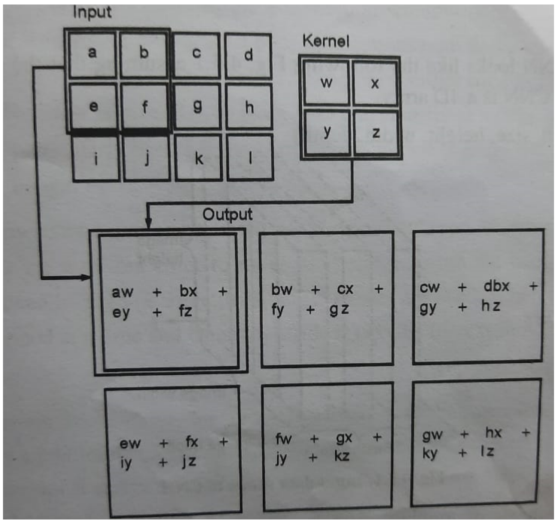
$$S(i, j) = (K * I)(i, j) = \sum_{m} \sum_{n} I(i - m, j - n) K(m, n)$$

The latter formula is more straightforward to implement in a machine learning library because there is less variation in the range of valid values of m and n.
While the commutative property is useful for writing proofs, it is no of a neural network implementation. Instead, many neural network libraries implement a related function called the cross-correlation, which is same as the convolution but without flipping the kernel:

$$S(i, j) = (I * K)(i, j) = \sum_{m} \sum_{n} I(i + m, j + n) K(m, n)$$

Mathematically, the convolution and cross-correlation operations can be defined as:
Convolution: (f * g)[n] = ∑ k f[k] g[n-k]
Cross-correlation: (f ⨂ g)[n] = ∑ k f[k] g[n+k]

Many machine learning libraries implement cross-correlation but call it convolution. In the context of machine learning, the learning algorithm will learn the appropriate values of the kernel in the appropriate place, so an algorithm based on convolution with kernel flipping will learn a kernel that is flipped relative to the kernel learned by an algorithm without the flipping.

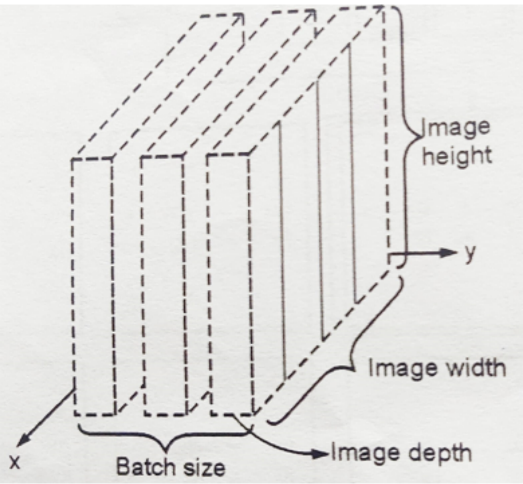


Convolution without kernel flipping.

4.3 Terminologies

1. Input shape
   The input data to CNN looks like the following assuming the data is a collection of images.

Input to CNN is a 4D array.
Input_shape = (batch_size, height, width, depth)

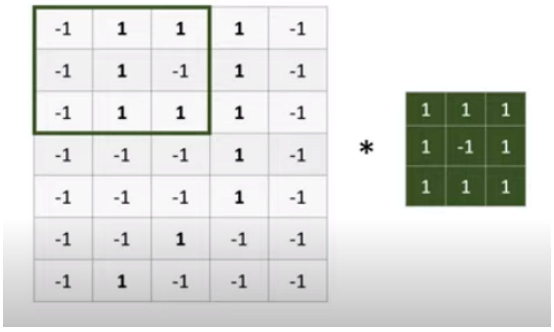


Batch size is the number of training examples in one forward/backward pass and depth of the image, is the number of color channels. RGB image has a depth of 3, and the greyscale image has depth of 1.

2. Output shape
The output of the CNN is also a 4D array.
output shape (batch_size, height, width, depth)

Where batch size would be the same as input batch size but the other 3 dimensions of the image might change depending upon the values of filter, kernel size, and padding.

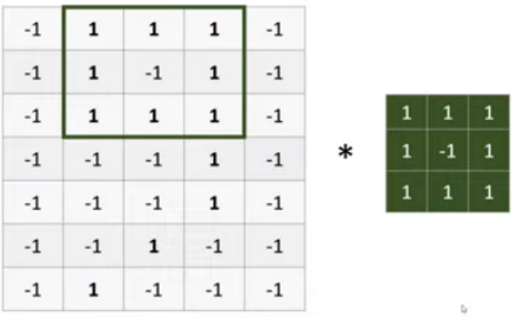
3. Filter
In a convolution neural network, input data is convolved over with a filter which is used to extract features. Filter or a kernel is a matrix that moves over the image pixel data (input). It performs a dot product with that particular region of the input data and outputs the matrix of the dot product.



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

*

1	1	1
1	-1	1
1	1	1



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

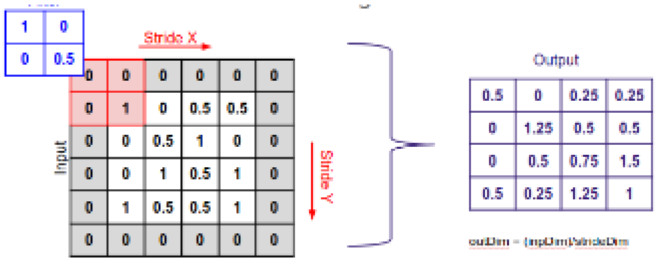
*

1	1	1
1	-1	1
1	1	1

4. Padding

Padding works by extending the area of an image processed by a convolutional neural network. The kernel is the neural network filter which moves across the image, scanning each pixel and converting data into a smaller, or sometimes larger, format. In order to assist the kernel with processing the image, padding is added to the frame of the image to allow for more space for the kernel to cover the image. Adding padding to an image processed by a CNN allows for more accurate analysis of images.

In simple words, Padding in convolution refers to the process of adding extra pixels or values around the edges of an input image before applying a convolutional operation. The main purpose of padding is to preserve the spatial dimensions of the input image and to avoid reducing the size of the output feature map.



1	0
0	0.5

0	0	0	0	0	0
0	1	0	0.5	0.5	0
0	0	0.5	1	0	0
0	0	1	0.5	1	0
0	1	0.5	0.5	1	0
0	0	0	0	0	0

0.5	0	0.25	0.25
0	1.25	0.5	0.5
0	0.5	0.75	1.5
0.5	0.25	1.25	1

5. Stride

The number of rows and columns traversed per slide are referred as stride. It can be thought as by how many pixels we want our filter to move as it slides across the image. Strides of 1, both for height and width can be used or sometimes, larger strides are used. When the stride is one the filter is moved to one pixel at a time and when the stride is two the filter is moved to two pixels at a time.

6. Tensors

In machine learning applications, the input is usually a multidimensional array of data and the kernel is usually a multidimensional array of parameters that are adapted by the learning algorithm. These multidimensional arrays are called as tensors.