Task 19: Introduction to Deep Learning, Building Block of Deep Learning, A look on Neural Network, Tensor Operations .

Artificial intelligence, machine learning, and deep learning :

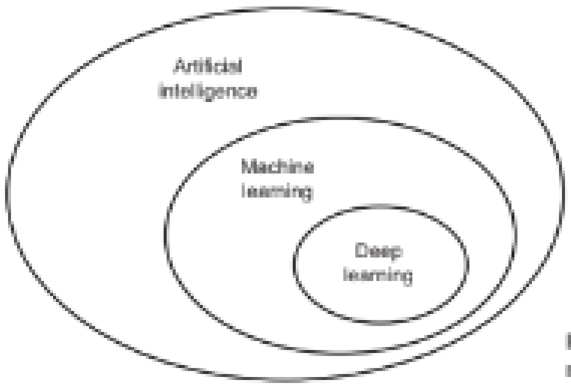

Figure 1.1 Artificial intelligence, machine learning, and deep learning

AI: the effort to automate intellectual tasks normally performed by humans**.**

ML: If a computer can go beyond the rules we give and perform beyond those rules by learning on its own and performing a specified task .With machine learning, humans input data as well as the answers expected from the data, and out come the rules. These rules can then be applied to new data to produce original answers. (ML can predict/find rules to execute a task)

machine learning is about mapping inputs (such as images) to targets (such as the label “cat”),

Symbolic AI: humans input rules (a program) and data to be processed according to these rules, and out come answers.

- ML Deals with large datasets unlike mathematical statistics but it is related to it.
- ML, DL ideas are proven by observation and experiment more often than theoretically.
- ML needs **input data points**, **expected output** and **function to measure the algo accuracy** .
- ML, DL learns the different and useful representation of the input data by transforming it.
- DL : Another take on learning representations from data by the use of successive umber of layers that tells the increasing representations
- Deep in DL : successive layers of representation.
- Depth of Model: Number of layers in DL.
- Successive layers are learned : through neural networks
- Weights: Layers use weights to play with the input data , parameters used in transforming the input data.

- How to find appropriate weights: Through loss function ., we initialize the weights mostly as 0 or give some initial random number than loss function to adjust the weights .

__Loss Function in Deep Learning__ : also called the objective function, measures the distance of the predicted output from a model to the true/original output(what you wanted the network to output) and computes a distance score, capturing how well the network has done

To Control the output of a neural network, you need to be able to observe it first and that's why we need Loss function that measures the score .

Then we used the score as a feedback signal to adjust the value of weights in the model.

Core Building Block of NN : Layers .

- Processes the data or can be called as a filter for data .
- Transforms the input data in more useful form.

To make the network ready for training, we need to pick three more things, as part of the compilation step:

A loss function—How the network will be able to measure its performance on the training data, and thus how it will be able to steer itself in the right direc-
tion.

An optimizer—The mechanism through which the network will update itself based on the data it sees and its loss function.

Metrics to monitor during training and testing—Here, we'll only care about accu-
racy (the fraction of the images that were correctly classified).

Tensors: Container for numerical Data, Matrix is a 2d Tensor.

- Dimensions called an axis in Tensors as it can contain number of dimensions.
- Scalar, 0D tensor . e.g : 12
- Vectors, 1D , e.d : [11,20,23]
- 5-dimensional vector is different from 5D tensors.

- A 5D vector has one axis with five dimensions along it, while a 5D tensor has five axes and can have any number of dimensions along each axis. The term "dimensionality" can refer to the number of entries along a specific axis or the number of axes in a tensor. It's more accurate to refer to a tensor with five axes as a tensor of rank 5, but the notation "5D tensor" is common nonetheless.

Tensors used : for neural network maths.

- Single value = 0D tensor
- Array = 1D tensor
- Single image = matrix/2D tensor
- Video = multi dimensional tensor/ndarray

Tensors are designed for hardware acceleration so that nn can do all the math .
Key attributes:

A tensor is defined by three key attributes:
Number of axes (rank)—For instance, a 3D tensor has three axes, and a matrix has two axes. This is also called the tensor's ndim in Python libraries such as Numpy.

Shape—This is a tuple of integers that describes how many dimensions the ten-
sor has along each axis. For instance, the previous matrix example has shape

(3, 5), and the 3D tensor example has shape (3, 3, 5). A vector has a shape with a single element, such as (5,), whereas a scalar has an empty shape, ().

Data type (usually called dtype in Python libraries)—This is the type of the data contained in the tensor; for instance, a tensor's type could be float32, uint8, float64, and so on. On rare occasions, you may see a char tensor. Note that string tensors don't exist in Numpy (or in most other libraries), because tensors live in preallocated, contiguous memory segments: and strings, being variable length, would preclude the use of this implementation.

Tensor Operations:
- Tensor slicing

```
>>> my_slice = train_images[10:100]
>>> print(my_slice.shape)
(90, 28, 28)
```

- Data batches

```
batch = train_images[:128]
```
And here's the next batch:
```
batch = train_images[128:256]
```
And the nth batch:
```
batch = train_images[128 * n:128 * (n + 1)]
```

- **Element wise operation :**
  Addition and relu(an activation function) operations are element wise operation. operations that are applied independently to each entry in the tensors being considered.

Element-wise operations: These operations are performed between tensors of the same shape, applying the operation to each corresponding element in the tensors.

- Broadcasting: Extending the shape of one tensor to match the shape of another tensor for arithmetic operations.
- Tensor Dot : Tensordot (also known as tensor contraction) sums the product of elements from a and b over the indices specified by axes
- Tensor Reshaping: Changing the shape of a tensor without altering its data.