Introduction_to_Deep_Learning

April 10, 2025

0.1 Introduction to Deep Learning:

Deep Learning is a subset of Machine Learning. Deep Learning is mostly about Neural Network. The main aim of Deep Learning or Neural Network is to mimic the human Brain. Means, It makes the machine learn like human being learn. To understand what deep learning is, we first need to understand the relationship deep learning has with machine learning, neural networks, and artificial intelligence. The best way to think of this relationship is to visualize them as concentric circles:

- At the outer most ring you have artificial intelligence. One layer inside of that is machine learning. With artificial neural networks and deep learning at the center.
- Deep learning is a more approachable name for an artificial neural network. The "deep" in deep learning refers to the depth of the network. An artificial neural network can be very shallow & deep.

#Some requirements to learn Deep Learning: * Mathematics: - Functions - Vectors - Linear Algebra -» [Metrix, Operations of Metrix] - Differential Calculus -» [Gradient, Partial Derivatives, Differentiation] - Graphs

- Programming:
  - Python with OOPs concepts
  - Code readability
- System:
  - At least i3 or i5 processor
  - At least 8gb of RAM
  - Good internet connection
- IDE / Code Editor:
  - Google Colab
  - Paperspace
  - Pycharm
  - VS code
  - jupyter notebook
  - spyder

And very last requirement is `Dedication`

1 Neural Neworks:

2 ANN: Artificial Neural Networks:-

- Neural networks are inspired by the structure of the cerebral cortex. At the basic level is the perceptron, the mathematical representation of a biological neuron. Like in the cerebral cortex, there can be several layers of interconnected perceptrons.
- ANNs are **core of deep learning**. Hence one of the most important topic to understand.
- ANNs are **versatile, scalable and powerfull**. Thus it can tackle highly complex ML tasks like classifying images, identying object, speech recognition etc.
- ANN can be single layer or Multiple layers.
- Above the ANN has just 3 layers, The green one is Input layer , Blue one is Hidden layer and the red one is output layer.
- In each neuron except input neuron we use some kinds of activation function for activating the neuron based on some threshold. Activation function also filters out the important data.
- Here each neuron passes some information to others neuron then others neuron take this information and do some calculation and pass to other , that's how it makes the neuron networks.

2.1 Biological Neuron:

- Biological Neuron produce short electrical impulses known as `action potentials` which travels through axons to the synapses which releases chemical signals i.e `neurotransmitters`.
- When a connected neuron recieves a sufficient amount of these neurotransmitters within a few milliseconds, it fires (or does not fires, think of a logic gate) its own action potential or elctrical impulse.
- These simple units form a strong network known as Biological Neural Network (BNN) to perform very complex computation task.

3 The first Artificial Neuron:

- It was in year 1943, Artificial neuron was introduced by-
  - Neurophysiologist Warren McCulloh and
  - Mathematician Walter Pitts
- They have published their work in `McCulloch, W.S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115-133 (1943). https://doi.org/10.1007/BF02478259`. read full paper at link
- They have shown that these simple neurons can perform small logical operation like OR, NOT, AND gate etc. These neuron only fires when they get two active inputs.

3.1 Deep Learning classification: -

- (1) **ANN** –> Artificial Neural Network. This ANN works on all kinds of Tabular dataset (excel sheet data). Any kinds of Regression & Classification task of Machine Learning can be solved by ANN.
- (2) **CNN** –> Convolutional Neural Network. CNN works on computer vision like Images, Videos. CNN architecture works well on these kinds of Image & videos data. Some application of CNN are:-
  - (a) Image Classification –> CNN, Transfer Learning
  - (b) Object Detection –> RCNN, FAST RCNN, FASTER RCNN, SSD, YOLO, DETECTRON
  - (c) Segmentation
  - (d) Tracking
  - (e) GAN » Generative Adversarial Network
- (3) **RNN** –> Recurrent Neural Network. RNN works whenever input data are text, time series or sequential data like Audio, Time series et cetra. Some techniques are used in RNN:-
  - RNN
  - LSTM RNN
  - Bidirectional LSTM RNN
  - Encoder, Decoder
  - Transformers
  - Bert
  - GPT1, GPT2, GPT3 #### NLP - Natural Language Processing is the part of RNN. NLP –> Text –> Vector
  - Bag of words (BOW)
  - TFIDF
  - word2vec
  - word embedding

4 Why Deep Learning?

- Computers have long had techniques for recognizing features inside of images. The results weren't always great. **Computer vision** has been a main beneficiary of deep learning. Computer vision using deep learning now rivals humans on many image recognition tasks.
- **Facebook** has had great success with **identifying faces** in photographs by using deep learning. It's not just a marginal improvement, but a game changer: "Asked whether two unfamiliar photos of faces show the same person, a human being will get it right 97.53 percent of the time. New software developed by researchers at Facebook can score 97.25 percent on the same challenge, regardless of variations in lighting or whether the person in the picture is directly facing the camera."
- **Speech recognition** is a another area that's felt deep learning's impact. Spoken languages are so vast and ambiguous. Baidu – one of the leading search engines of China – has developed a voice recognition system that is faster and more accurate than humans at producing text on a mobile phone. In both English and Mandarin.

- What is particularly fascinating, is that generalizing the two languages didn't require much additional design effort: "Historically, people viewed Chinese and English as two vastly different languages, and so there was a need to design very different features," Andrew Ng says, chief scientist at Baidu. "The learning algorithms are now so general that you can just learn."
- **Google** is now using deep learning to **manage the energy** at the company's data centers. They've cut their energy needs for cooling by 40%. That translates to about a 15% improvement in power usage efficiency for the company and hundreds of millions of dollars in savings.

`To understand more about ANN (Artificial Neural Network)` Please visit Tensorflow Playground and play with the Neural Network.

5 Installation of Tensorflow, Keras, pytorch & Basic of Google Colab

5.0.1 Tensorflow:

- If you are using google colab then you don't have to install tensorflow additionaly, Colab already has tensorflow pre-installed you have to just import that. To import tensorflow just write `import tensorflow as tf`
- But if you are using your local system like jupyter notebook then you have to install it. To install that first of all create a virtual Environment then activate your Environment and write to your anaconda promt `pip install tensorflow` it will install latest version of tensorflow for you. To check the version of tensorflow `tf.__version__` and for keras `tf.keras.__version__`

5.0.2 pytorch:

- For installing pytorch please visit their website Pytorch

5.0.3 Activate GPU on colab:

To activate GPU on colab go to the Runtime above and change the runtype type to GPU. To check your GPU just write `!nvidia-smi`

##Deep Learning Frameworks

Deep learnings is made accessible by a number of open source projects. Some of the most popular technologies include, but are not limited to, Deeplearning4j (DL4j), Theano, Torch, TensorFlow, and Caffe. The deciding factors on which one to use are the tech stack they target, and if they are low-level, academic, or application focused. Here's an overview of each:

DL4J:

- JVM-based
- Distrubted
- Integrates with Hadoop and Spark

Theano:

- Very popular in Academia

- Fairly low level
- Interfaced with via Python and Numpy

Torch:

- Lua based
- In house versions used by Facebook and Twitter
- Contains pretrained models

TensorFlow:

- Google written successor to Theano
- Interfaced with via Python and Numpy
- Highly parallel
- Can be somewhat slow for certain problem sets

Caffe:

- Not general purpose. Focuses on machine-vision problems
- Implemented in C++ and is very fast
- Not easily extensible
- Has a Python interface

  pip install tensorflow==2.0.0

  To run from Anaconda Prompt

  !pip install tensorflow==2.0.0

  To run from Jupyter Notebook

Both Tensorflow 2.0 and Keras have been released for four years (Keras was released in March 2015, and Tensorflow was released in November of the same year). The rapid development of deep learning in the past days, we also know some problems of Tensorflow1.x and Keras:

- Using Tensorflow means programming static graphs, which is difficult and inconvenient for programs that are familiar with imperative programming
- Tensorflow api is powerful and flexible, but it is more complex, confusing and difficult to use.
- Keras api is productive and easy to use, but lacks flexibility for research

Official docs link for DETAILED INSTALLATION STEPS for Tensorflow 2

```
[ ]: # Verify installation
     import tensorflow as tf
```

```
[ ]: print(f"Tensorflow Version: {tf.__version__}")
     print(f"Keras Version: {tf.keras.__version__}")
```

```
Tensorflow Version: 2.5.0
Keras Version: 2.5.0
```

Tensorflow2.0 is a combination design of Tensorflow1.x and Keras. Considering user feedback and framework development over the past four years, it largely solves the above problems and will become the future machine learning platform.

Tensorflow 2.0 is built on the following core ideas:

- The coding is more pythonic, so that users can get the results immediately like they are programming in numpy
- Retaining the characteristics of static graphs (for performance, distributed, and production deployment), this makes TensorFlow fast, scalable, and ready for production.
- Using Keras as a high-level API for deep learning, making Tensorflow easy to use and efficient
- Make the entire framework both high-level features (easy to use, efficient, and not flexible) and low-level features (powerful and scalable, not easy to use, but very flexible)

Eager execution is by default in TensorFlow 2.0 and, it needs no special setup. The following below code can be used to find out whether a CPU or GPU is in use

6 GPU/CPU Check

[ ]:
```
tf.config.list_physical_devices('GPU')
```

[ ]:
```
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

[ ]:
```
tf.config.list_physical_devices('CPU')
```

[ ]:
```
[PhysicalDevice(name='/physical_device:CPU:0', device_type='CPU')]
```

[ ]:
```
CheckList = ["GPU", "CPU"]
for device in CheckList:
    out_ = tf.config.list_physical_devices(device)
    if len(out_) > 0:
        print(f"{device} is available")
        print("details\n",out_)
    else:
        print(f"{device} not available")
```

```
GPU is available
details
 [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
CPU is available
details
 [PhysicalDevice(name='/physical_device:CPU:0', device_type='CPU')]
```

[ ]:
```

```