YEESI 104 : Machine Vision in Agriculture

Deep Learning for Image Classification

YEESI Innovation Lab

Sokoine University of Agriculture

June 17, 2022

Introduction

This lesson covers the basics of deep learning and also a practical example on image classification.

Lesson Objectives: At the end of the lesson,students should be able to: -

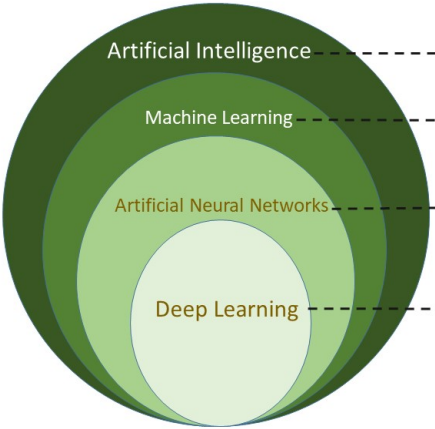
- Explain different Deep Learning concepts
- Install and use Deep Learning frameworks such as Tensorflow
- Perform Data Preprocessing
- Train and Optimize Deep Learning models

Lesson Contents

The following will be covered in this lesson:

1. Basics of Deep Learning and Neural Networks
2. Environment Setup (tensorflow)
3. Deep Learning with deep neural networks.
4. Model design with tensorflow/keras.
5. Model training testing.

Basics of AI, ML, ANN and DL



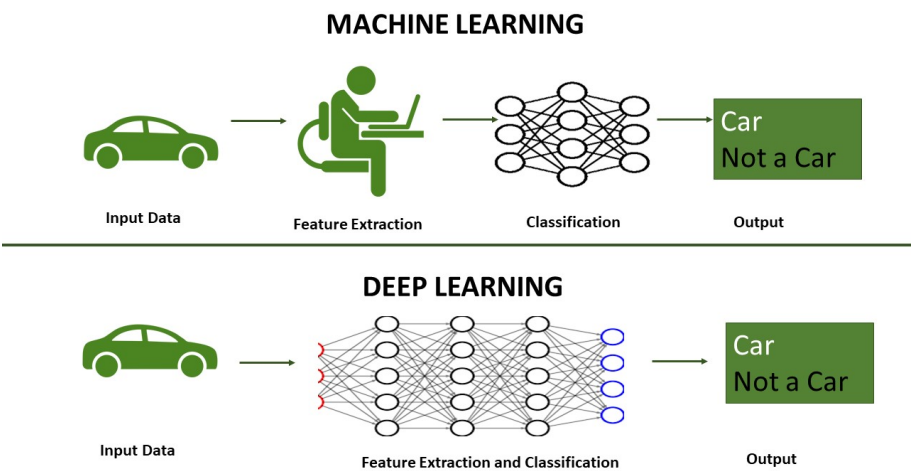
Theory and development of intelligent machines that can simulate human thinking, capability and behaviour

An application or subset of AI that allows machines to learn from data without being programmed explicitly.

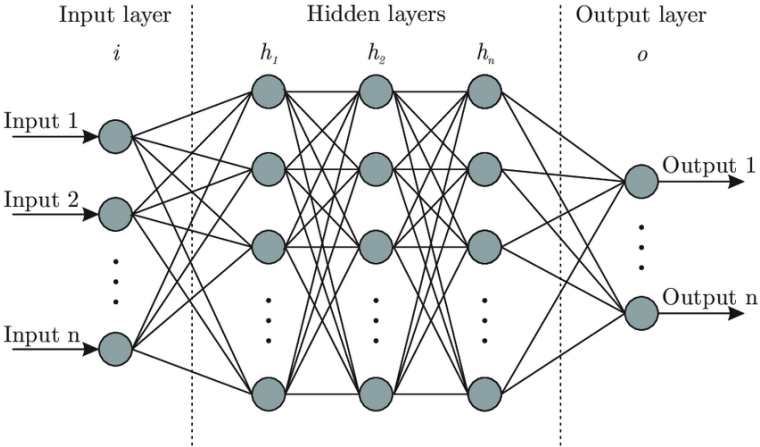
A computational model that mimics the way nerve cells work in the human brain i.e. Brain inspired ML models

A subset of ML based on artificial neural networks with representation learning.

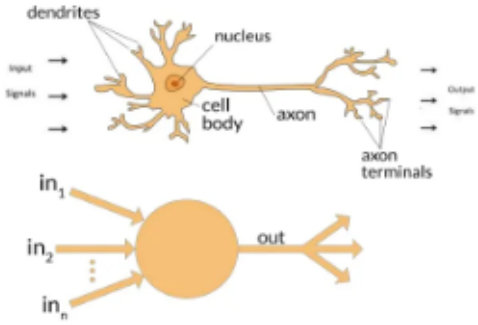
Deep Learning Vs Traditional Machine Learning



Artificial Neural Networks

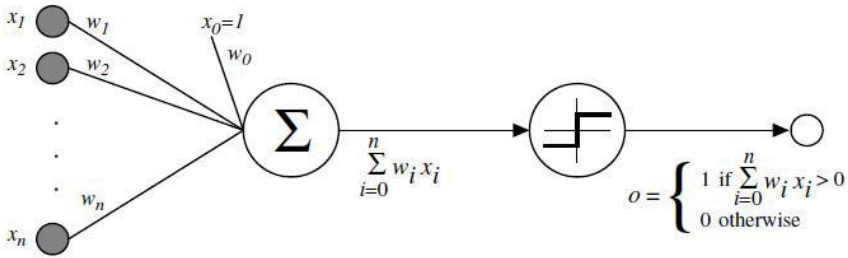


What is a Neural Net?



- Also known as perceptron.The simplest form of ANN is the Perceptron, a model with one layer only, very similar to the linear regression model.

A Simplified mathematical model of how the neurons operate



From: http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/

continuation

Input			Weights
x1	0.5	x	w1
x2	0.2	x	w2
x3	0	x	w3

Target
1
0

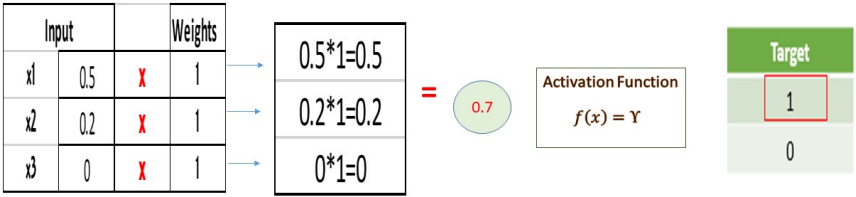
Input			Weights
x1	0.5	x	1
x2	0.2	x	1
x3	0	x	1

0.5*1=0.5
0.2*1=0.2
0*1=0

= 0.7

Target
1
0

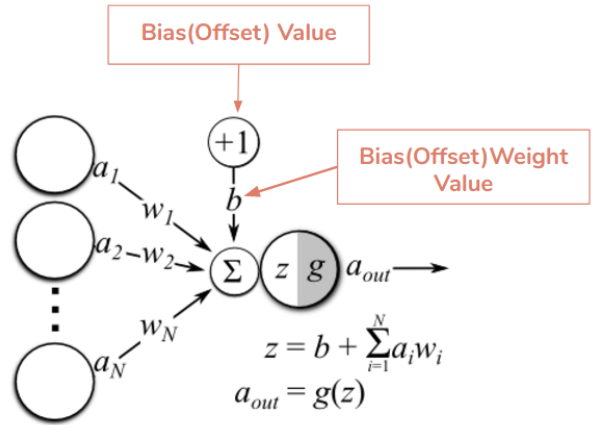
continuation



Weights and Bias

- Inside each neuron, the linear combination of inputs and weights includes also a bias, similar to the constant in a linear equation
- Weights control the signal (or the strength of the connection) between two neurons. In other words, a weight decides how much influence the input will have on the output.
- The bias unit guarantees that even when all the inputs are zeros there will still be an activation in the neuron.
- Without these spare bias weights, our model has quite limited "movement" while searching through solution space.

Weights and Biases



Activation Functions

- The activation function defines the output of that node.
- They decide whether a neuron should be activated or not and introduce non-linear transformation to a neural network
- The activation function is a mathematical "gate" in between the input feeding the current neuron and its output going to the next layer.
- Activation functions in output layers of ML models mostly squash the value between a bounded range like 0 to 1.
- Activation used in hidden layers of neural networks provide non-linearity

Activation Functions

- Examples of Output layer Activation functions
  - Sigmoid - binary classification
  - Softmax - multiclass classification
- Examples of Hidden layer Activation functions
  - Rectified Linear Unit (ReLu)
  - Exponential Linear Unit
- This link gives a detailed description of commonly used activation functions.
- Another link

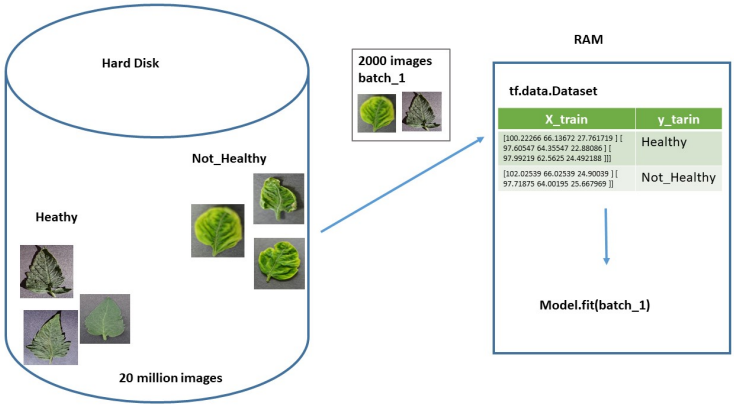
Tensorflow

- TensorFlow is a software library or framework, designed by Google to implement machine learning and deep learning concepts in the easiest manner.
- In this tutorial TensorFlow and Keras will be used.

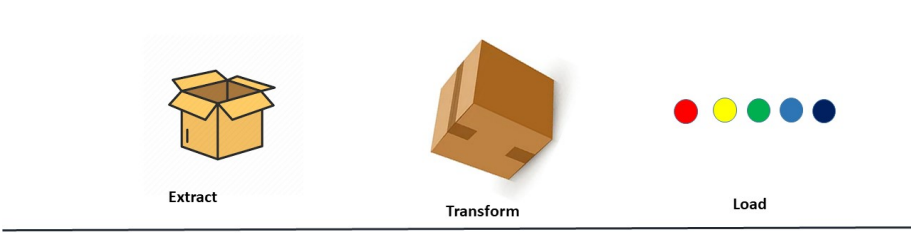
Tensorflow

- Tensors are used as the basic data structures in TensorFlow language.
- Tensors are defined as multidimensional array or list.
- Install TensorFlow through the terminal:*pip install tensorflow*
- OR In Anaconda you can create a new environment and add tensorflow, keras and matplotlib packages

tf.data: Fast, flexible, and easy-to-use input pipelines



tf.data API


Extract
Transform
Load
tf.data
Input Pipeline
Dataset
Extract
Transform
Load
Consume data (from memory, local disk, GCS, S3...etc)
Shuffle examples
Create mini-batches
Model