Practical Introduction To Neural Networks and Deep Learning

Introduction Module

Perumadura De Silva

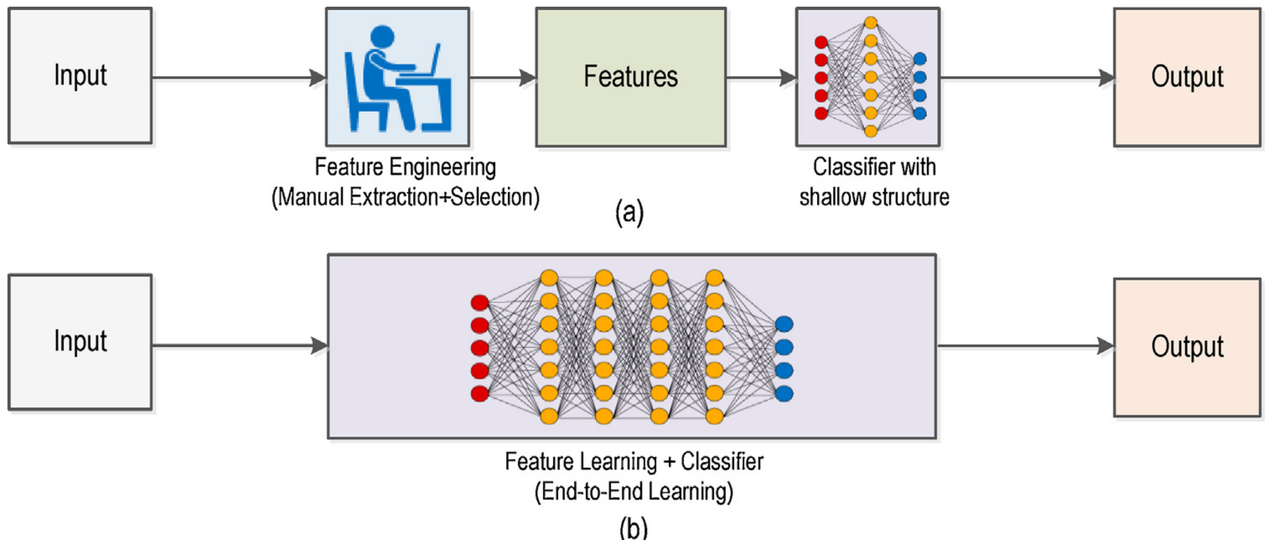
Introduction Module

Module Content

- Machine Learning: Traditional vs Modern
  - Introduction
  - Applications
- Basics of Deep learning
  - Deep Neural Networks
  - Training Neural Networks
- An Introduction To TensorFlow
  - Tensors and their flow
  - What is TensorFlow
  - TensorFlow Architecture
  - Low-Level Intro
  - Tensors
  - Tensor Operations

Machine Learning: Traditional vs Modern

Introduction

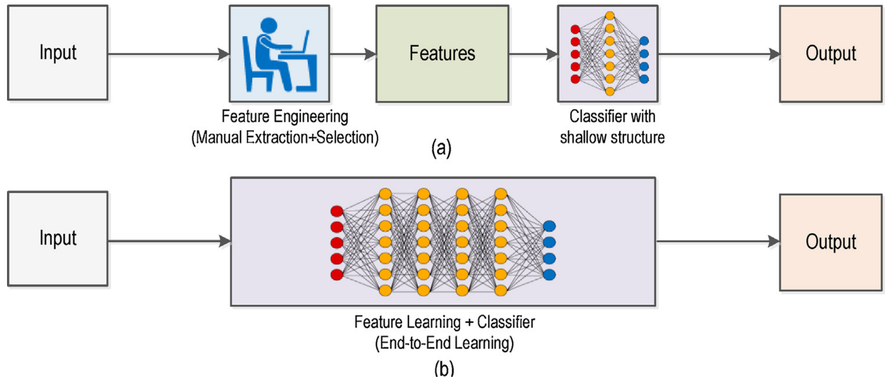


Traditional machine learning, (b) Neurocomputing
Source : ref1

Machine Learning: Traditional vs Modern

Introduction

- Traditional Methods:
  - Symbolic AI:
    - Expert systems: uses symbols and rules to process a decision
  - Traditional Machine Vision:
    - Engineer feature extractors manually
      - Sobel Filters (Edge detection)
      - HOG (Histogram of Oriented Gradients): Features for object detection

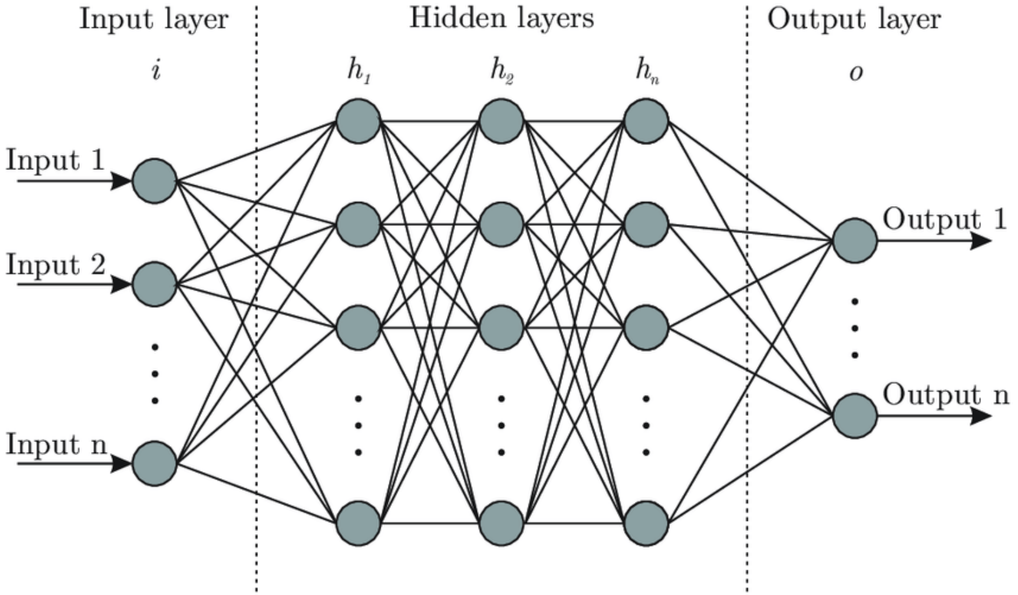


Traditional machine learning, (b) Neurocomputing
Source : ref1

Machine Learning: Traditional vs Modern

Introduction

- Neurocomputing methods:
  - Multi layer perception (MLP)
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Attention Layers



A Neural Network (ref-2)
Inputs: Data and Outputs: Predictions

Machine Learning: Traditional vs Modern

Introduction

Pros | Cons

Modern Methods	Traditional methods
End-to-End model optimisation	Manual feature engineering
Difficult to interpret the model	The model internal dynamics are known
Expensive optimisation	Requires less resources to optimise
Does not require domain specific knowledge	Expertise in a given domain is required
Robust against uncertainties	Fails under uncertain conditions

Comparison between traditional methods and Neurocomputing

Machine Learning: Traditional vs Modern

Applications

- Computer vision (Image/ Video):
  - Classification
  - Segmentation
  - Object detection
  - Enhancement and Inpainting
  - Data generation

Image inpainting
source: ref-2

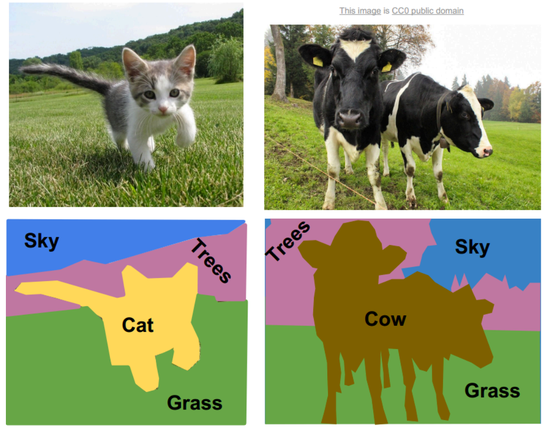
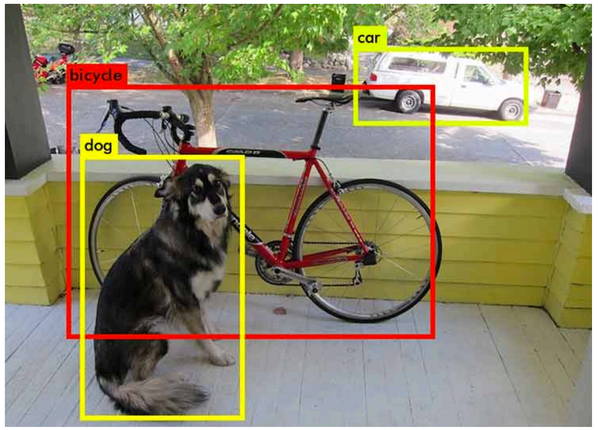


Image segmentation
source: ref-3



Object detection
source: ref-4

Machine Learning: Traditional vs Modern

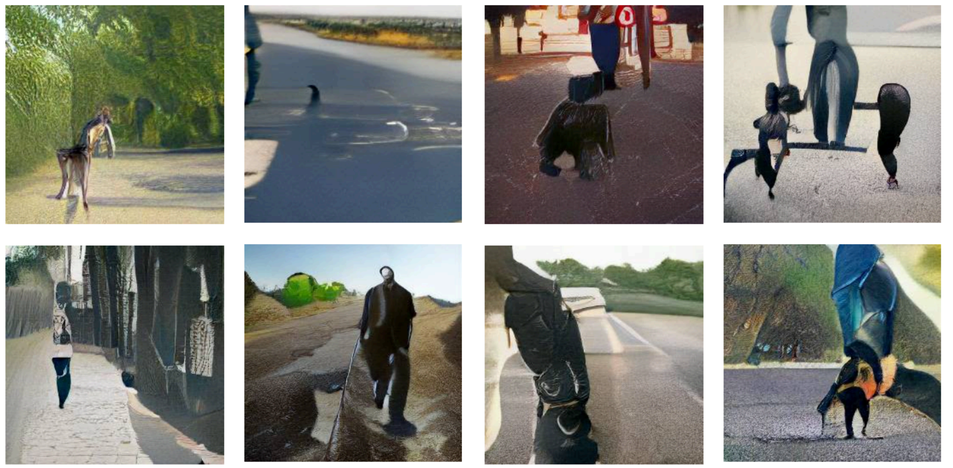
Applications

- Text data:
  - Sentiment/Token classification
  - Story generation
  - Translation
  - Visual Question and Answering:
    - Text + Image/Video
  - Text to image generation

What color are her eyes?
What is the mustache made of?

Visual Q&A
source: ref-4

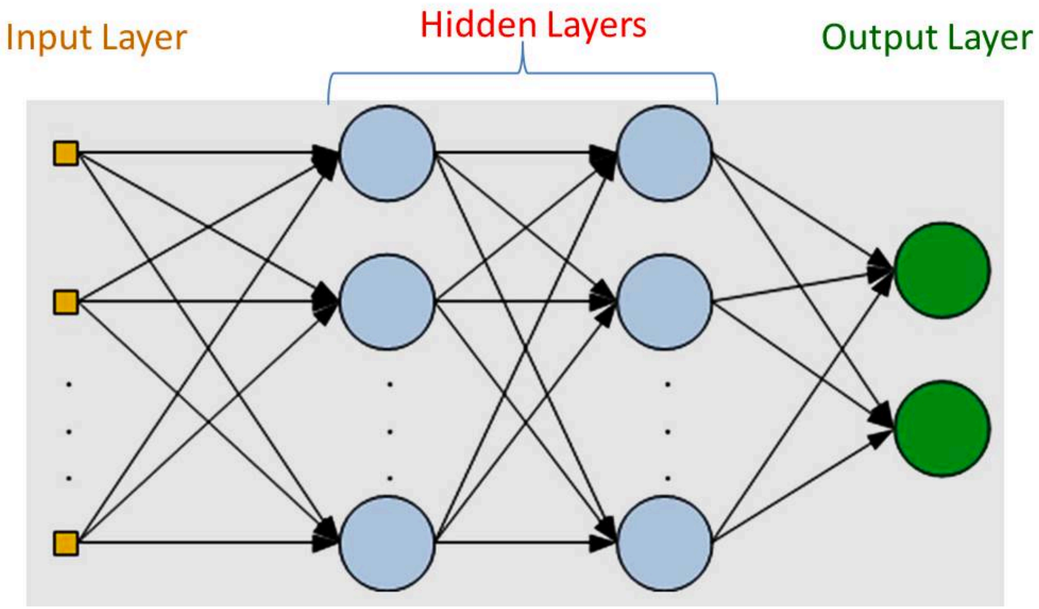
A person walking on the road with a black dog



Text to image generation:
Demo: https://huggingface.co/spaces/flax-community/dalle-mini

Basics of deep learning
Deep Neural Networks

- A feedforward network can be expressed as:
  - $y = f(x, W_\theta) = xW_\theta$
    - where, $f(.)$ the MLP function that maps inputs x to y, θ are tuneable model parameters
- Feedforward because:
  - The information flows from inputs (x) towards the outputs (y) through a set of intermediate computations (Transformation of information by means of θ)

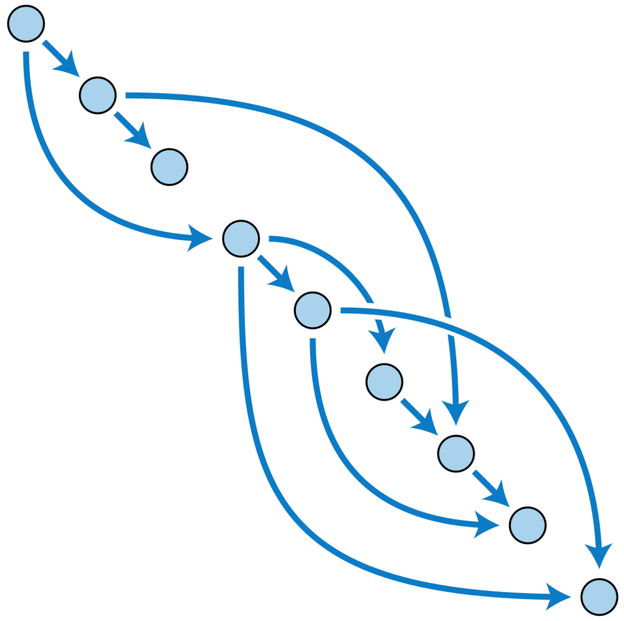


Ref.5: Multilayer perceptron (A Type of deep feed forward networks) with 2 hidden layers

Basics of deep learning

Deep Neural Networks

- Typically no feedback connection from outputs back to the network
- Only recurrent neural networks has a feedback connection
- Constructed by composing (Chaining) many functions together, hence a network
  - E.g: $y = f_{output}(f_2(f_1(f_{input}(x))))$
- Number of chaining functions determine the depth of the network (deep nets)
- Generalized by directed acyclic graph (DAG)



Ref.2: A directed acyclic graph

Basics of deep learning
Training Neural Networks

- By training we adjust parameters/weights to satisfy a task
- Training methods:
  - Supervise learning
    - Involves labels
  - Unsupervise/Semi-Supervise learning
    - No labels or few available once
  - Self-Supervise learning
    - Algorithm labels the data itself
  - Reinforcement learning
    - Involves a reward function

Setup Google Colab

An Introduction To TensorFlow

Perumadura De Silva
Abhiram Kolli

An Introduction To TensorFlow

Tensors and their flow

- Tensors
  - A n-dimensional array consisting of primitive values
  - Rank of a tensor: number of dimensions
  - Shape: tuple of integers
  - In TensorFlow, 1D (Vector), 2D (Matrix)…nD (Tensor) are called Tensors
- Flow

Operation: x element wise multiply x

- $x \dashrightarrow x^2$

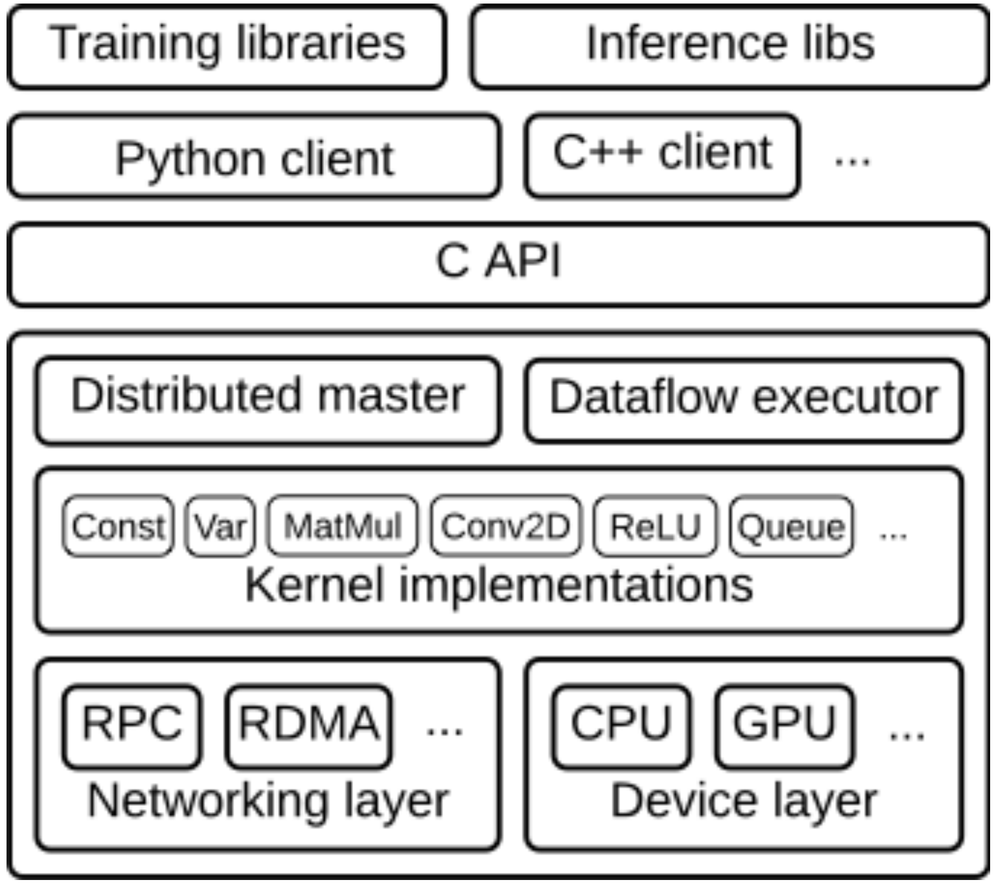
An Introduction To TensorFlow

What is TensorFlow?

- A high-level framework that allows GPGPU using python
- TensorFlow core is written in C++
- TensorFlow APIs available in Python, Java, C++, Go, Javascript
- Your Code ->
  - Compiled to an intermediate representation (Compute Graph)
  - Graph edges are tensors and nodes are operations (Ops)
  - Generation of corresponding CUDA kernels (NVIDIA GPUs)
  - GPU Device execution

An Introduction To TensorFlow

TensorFlow: Architecture



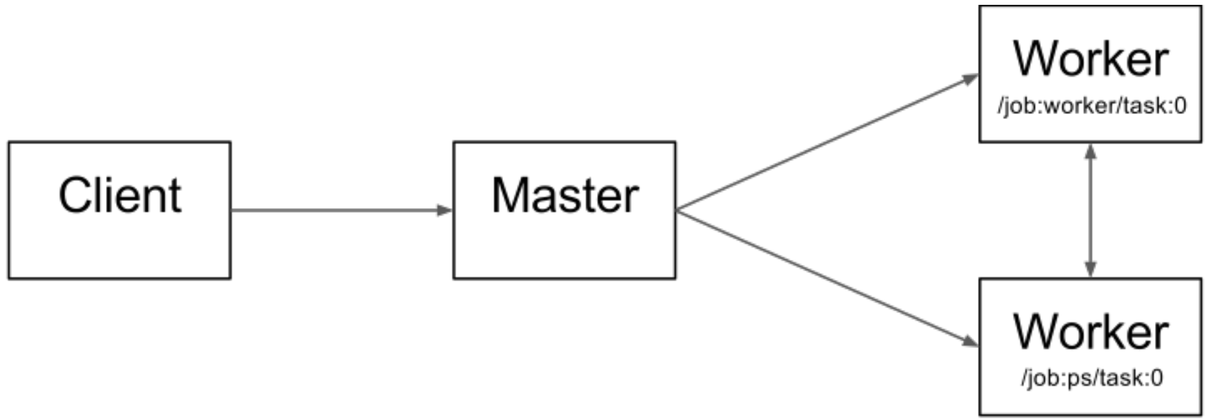
An Introduction To TensorFlow

TensorFlow Architecture

- Client: High-level wrapper that defines the TF graph (Python, Java, Go)
- Distributed Master:
  - Prune a subgraph from the main graph and partition the subgraph into multiple pieces and assign them to different devices via worker services.
  - Local case: Similar to distributed master but communicate only with local devices
- Worker Services: Schedule graph operations on devices (e.g Cuda kernels on Nvidia GPUs)
- Kernel implementation: Individual graph ops (e.g Conv, Matmul)

An Introduction To TensorFlow

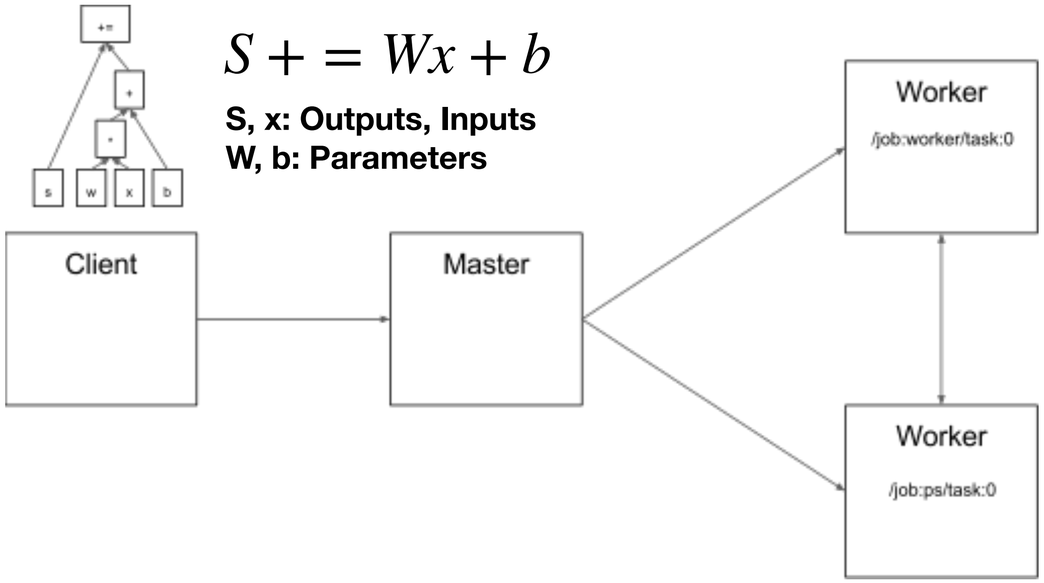
TensorFlow Architecture



An Introduction To TensorFlow

TensorFlow Architecture

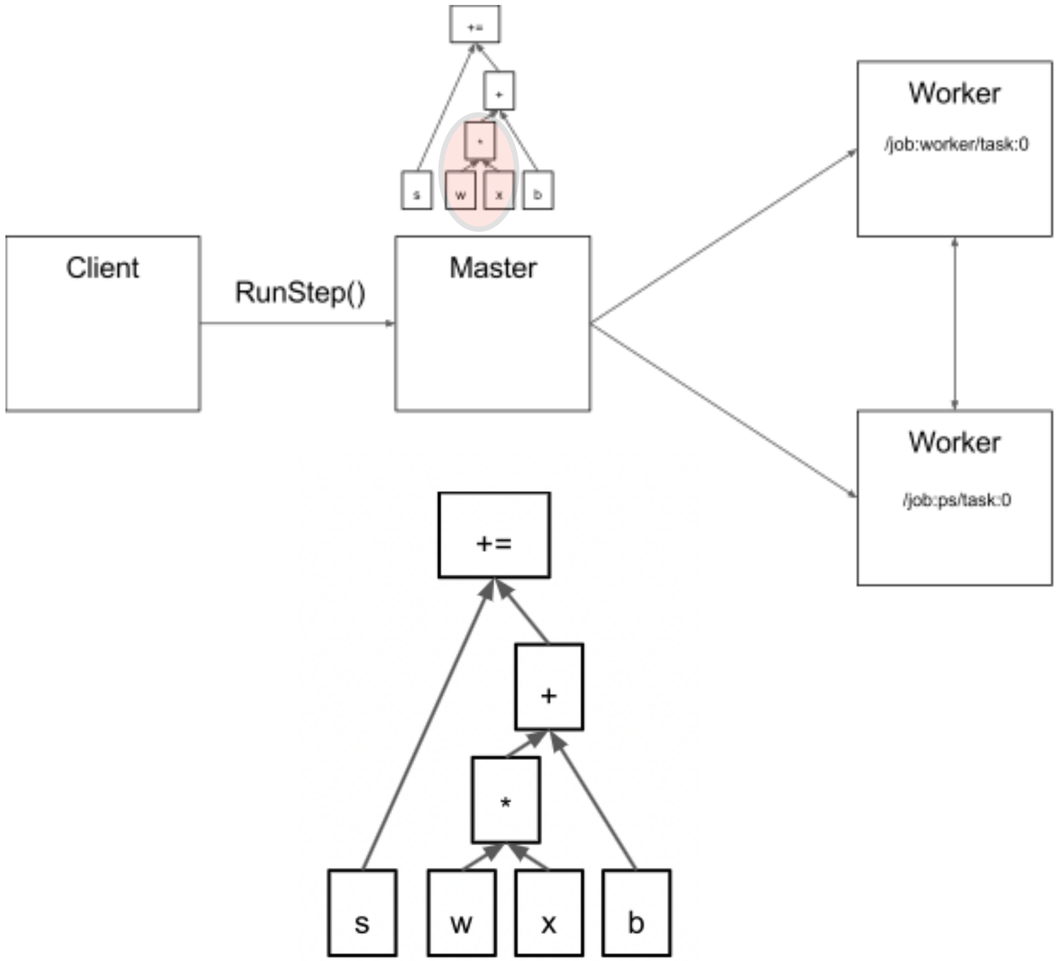
- Client
  - Defines a graph and send it to Master as a tf.GraphDef protocol buffer for evaluation.
  - PROTOCOL BUFFER: A data structure serialization method
    - https://developers.google.com/protocol-buffers
  - Serializing: Data structure to Memory Buffer (Series of bytes)
  - Why Protocol?: Serializing and Deserialized using the defined rules



An Introduction To

TensorFlow Architecture

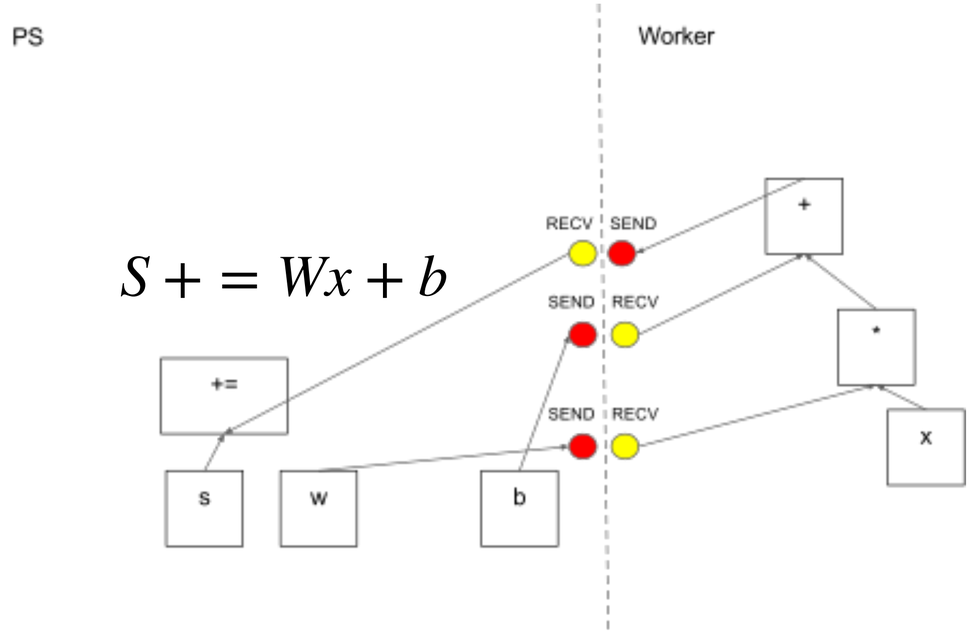
- Master
  - Partition the graph and get the subgraph that corresponds to the node on evaluation request (Session run)
  - Subgraphs are partitioned to get the ops and params



An Introduction To TensorFlow

TensorFlow Architecture

- Worker Service
  - Schedules the execution of the graph operations
  - Establishes the communications between tasks
  - Manages the communication between CPU, GPU as well as local GPUs



An Introduction To TensorFlow

TensorFlow Architecture

- Kernel Implementations
  - Example kernels: mathematical ops, array manipulations and state management
  - The implemented kernels are optimized for a specific device (CPU, GPU, TPU)
  - For efficiency TensorFlow uses the CuDnn library
  - Low-level optimized custom ops are possible with TensorFlow C++ implementation

Reference

- Ref-1: Deep learning for smart manufacturing: Methods and applications, Section 2, Page 147
- Ref-2: https://de.wikipedia.org/wiki/Inpainting#/media/Datei:Restoration.jpg
- Ref-3: https://tariq-hasan.github.io/concepts/computer-vision-semantic-segmentation/
- Ref-4: https://medium.com/analytics-vidhya/yolo-object-detection-343a430f3b48
- Ref-4: https://towardsdatascience.com/visual-question-answering-with-deep-learning-2e5e7cbfdcd4
- Ref-5: https://www.groundai.com/project/multi-stage-jamming-attacks-detection-using-deep-learning-combined-with-kernelized-support-vector-machine-in-5g-cloud-radio-access-networks/1