# Agenda

1. Why Platform?
2. Why ML Platform?
3. Engineering Challenges in Model Life Cycle
4. Feature Engineering
5. Model Training
6. Model Serving
7. Common Model Architectures
8. Tensorflow
9. Keras

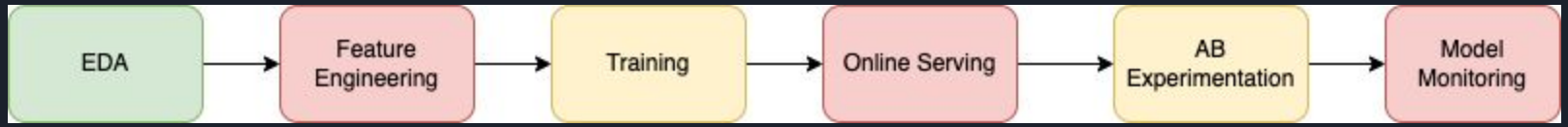
# Why Platform?

Platforms exist to enable the reuse of technology and address common problem statements across multiple applications, ensuring productivity, efficiency, scalability, and standardization throughout the organization

# Why ML Platform?

ML platforms exist to enable the reuse of technology across different datasets, feature types, and model types at various stages of the model lifecycle, thereby enhancing productivity and efficiency

# Engineering Challenges in Model Life Cycle



# Distributed Systems 101

1. Distributed Storage
2. Distributed compute
3. Distributed Joins
4. Network Co-ordination
5. Fault Tolerance

# Feature Engineering

- Feature Generation at scale - offline and nearline
- Training Data Generation on high dimensionality features
- Feature Storage for offline and real time access
- Feature consumption - latency constraints
- Feature parity in offline, nearline and online

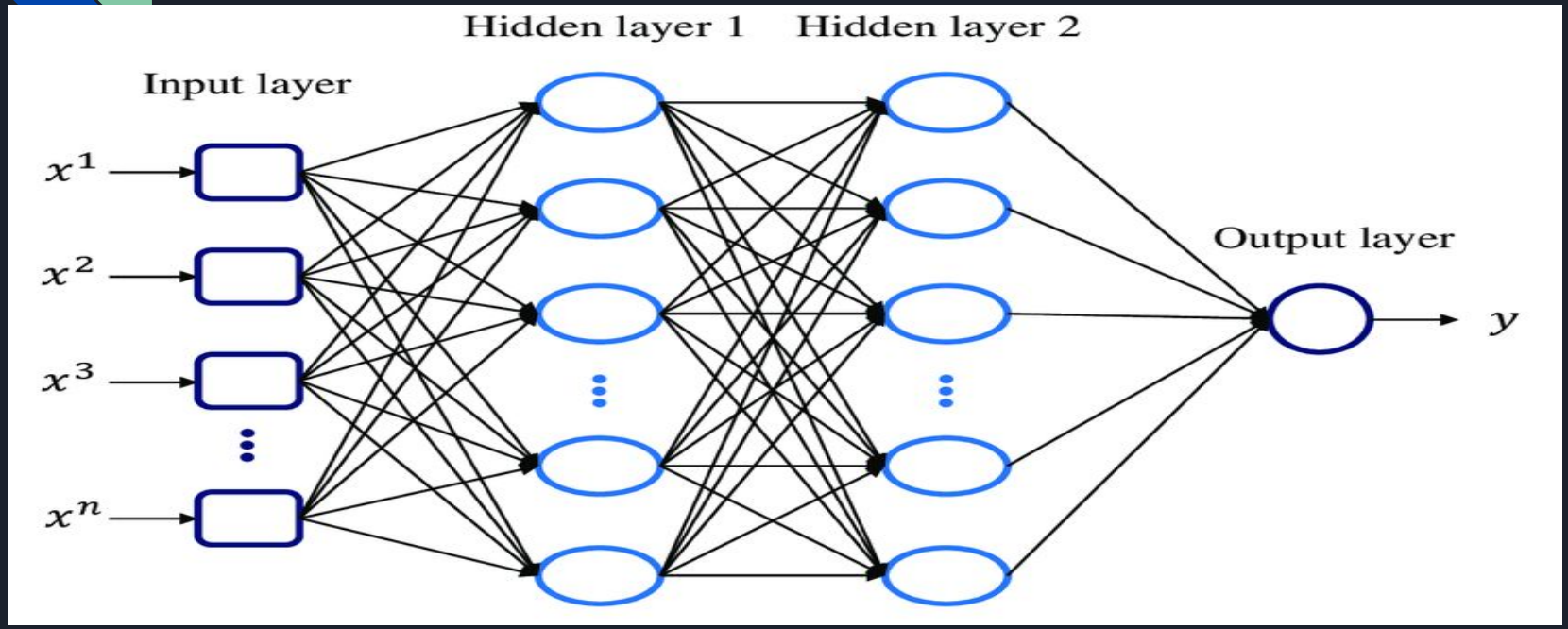
# Model Training

- Backprop and update of Gradients
- Data Parallelism and Model parallelism
- Synchronization of gradients in multiple nodes
- Distributed Graph execution and update
- Hyper parameter tuning
- Fault tolerance and stragglers
- Hardware utilization and optimizations

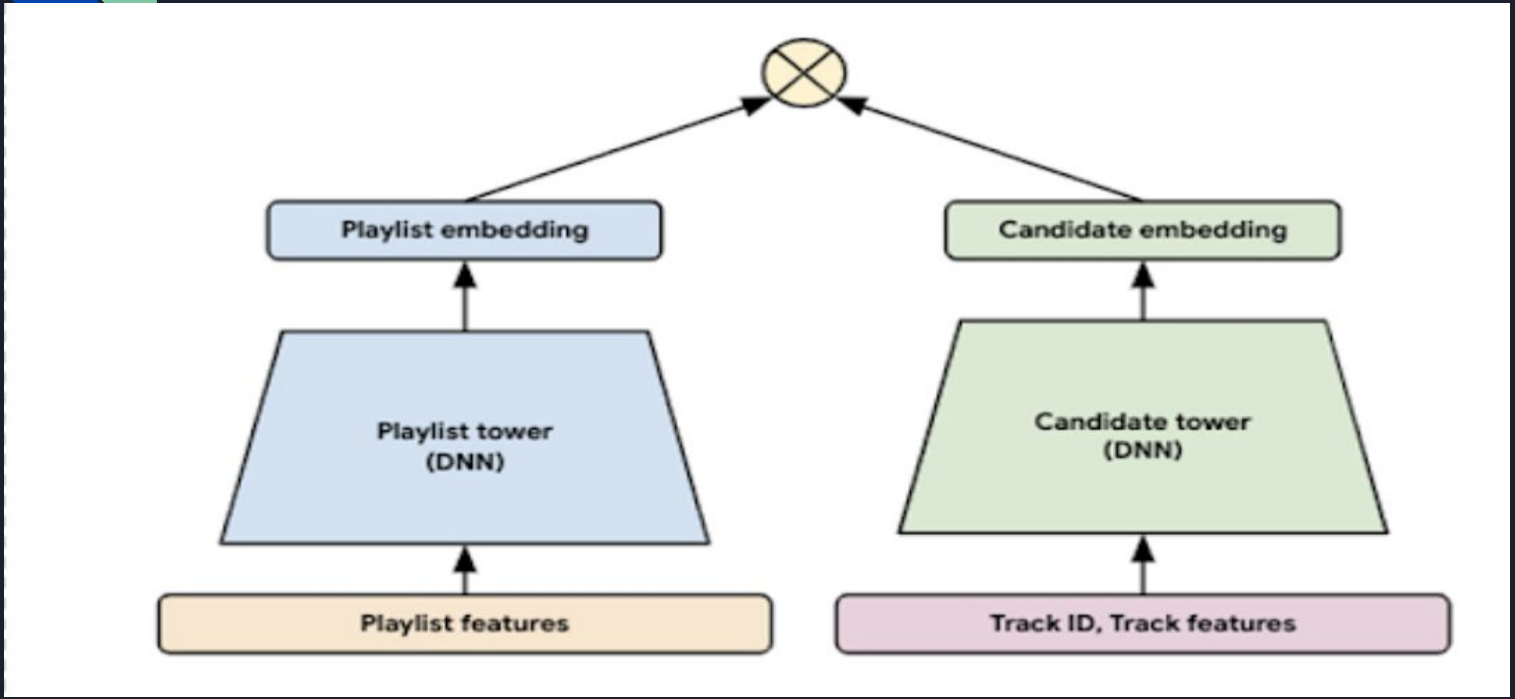
# Model Serving

- Running predictions at strict latency constraints
- Server optimizations at scale.
- Reusing model predictions across requests
- Model compression and graph optimizations
- Efficient execution of the graph
- Batch execution of requests
- Hardware utilization and optimization
- Logging and monitoring

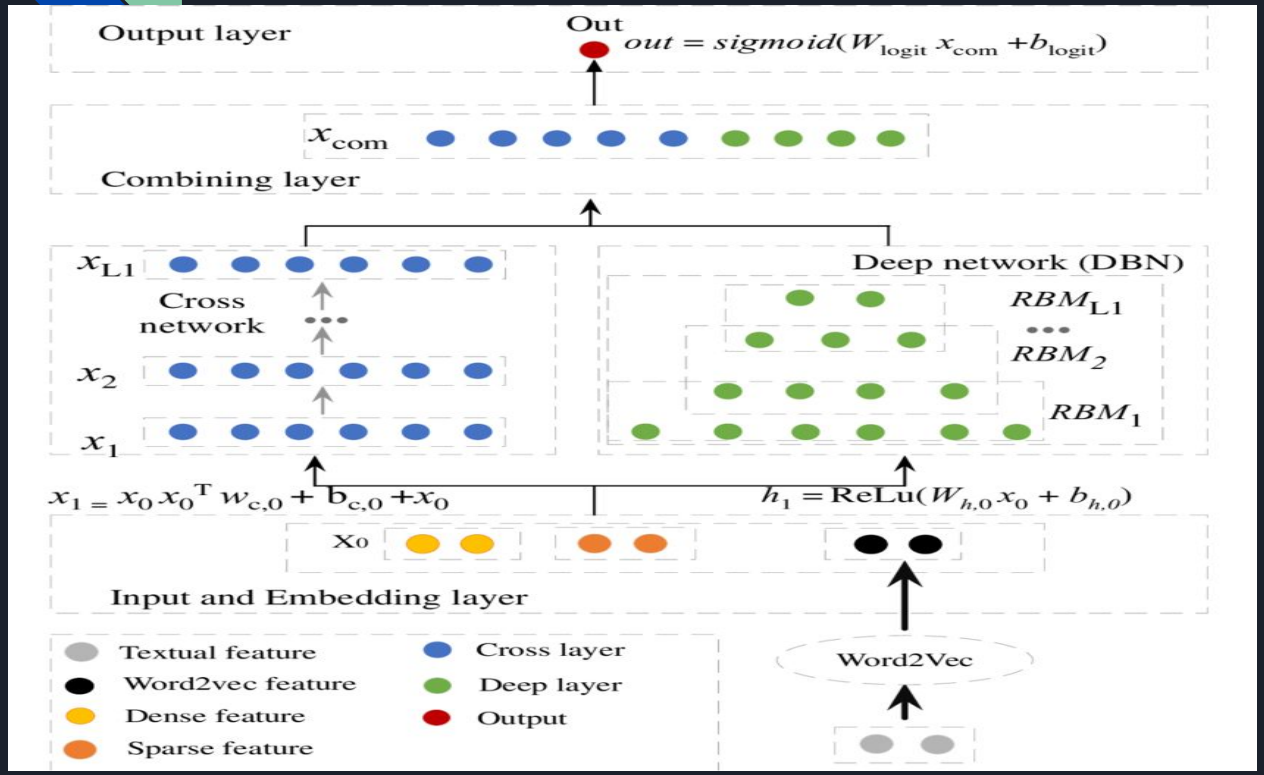
# Model Architecture - MLP



# Model Architecture - Two tower



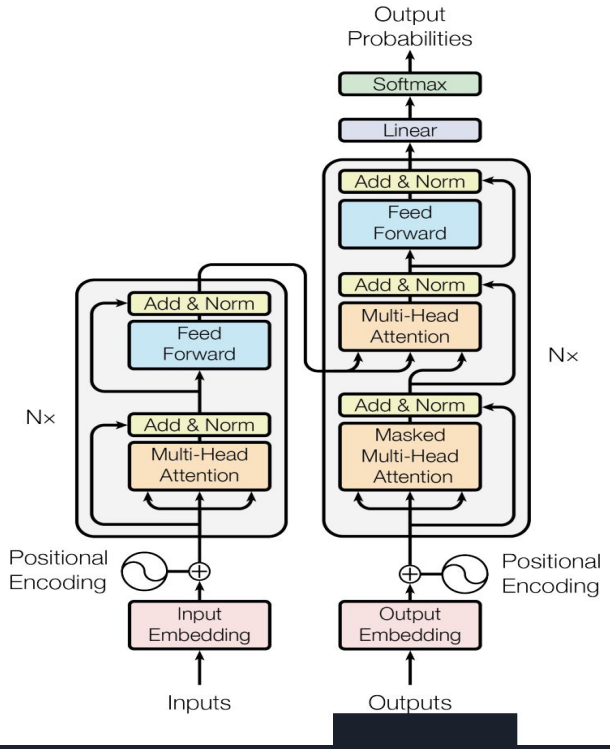
# Model Architecture - DCN



# Model Architecture - Transformer

BERT

Encoder



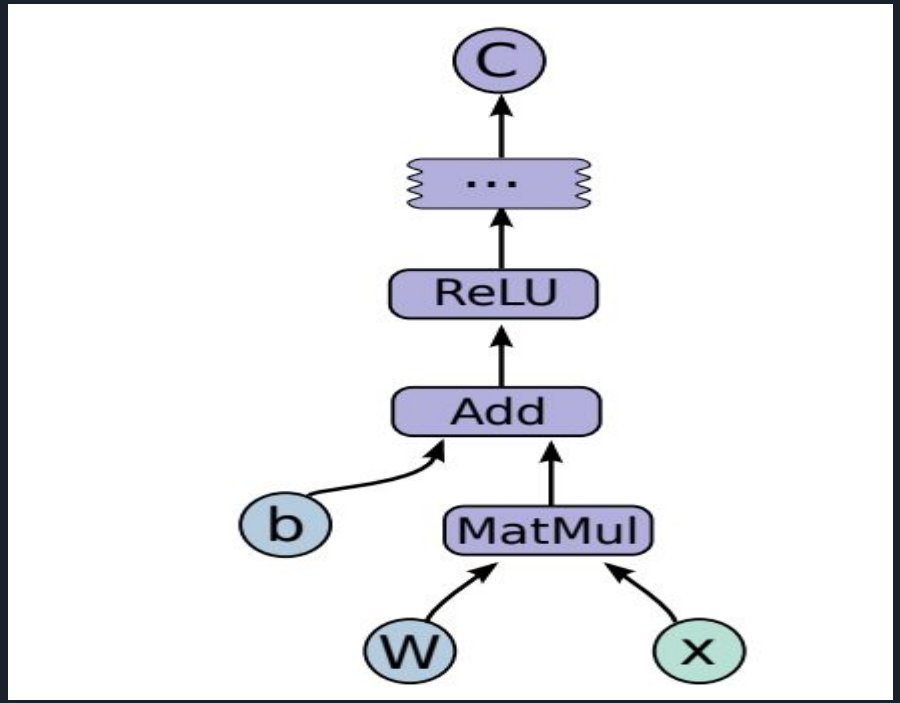

GPT

Decoder

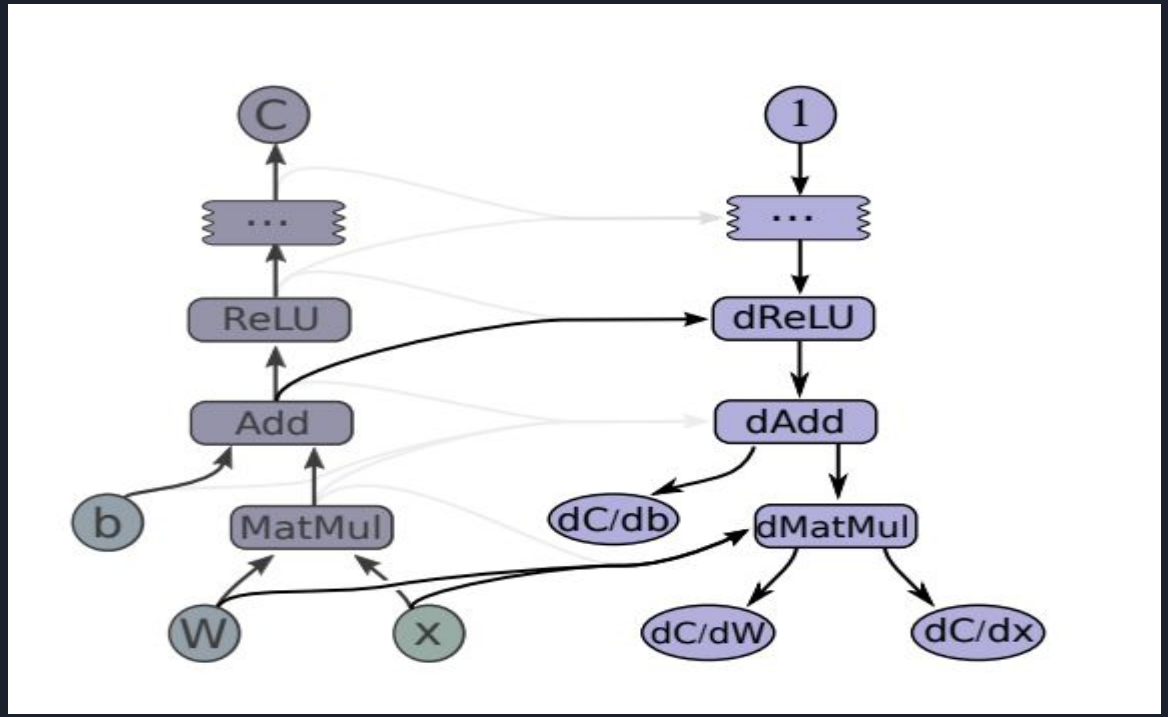
# Tensor Flow

Flexible and scalable dataflow graph execution framework optimized for machine learning operations and specialized hardware resources.

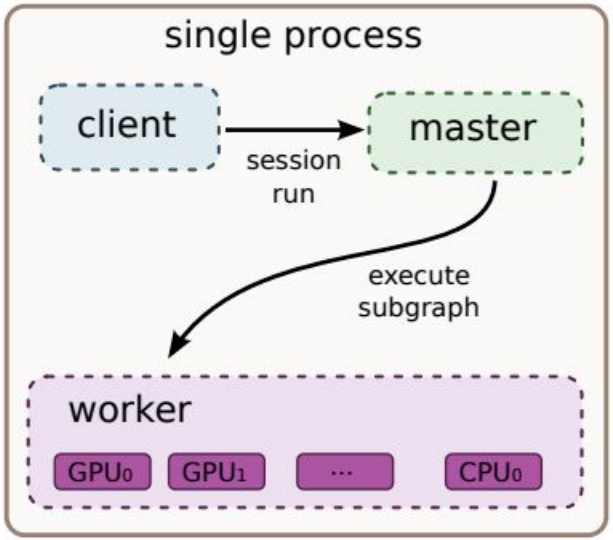
# TensorFlow - DataFlow Graph

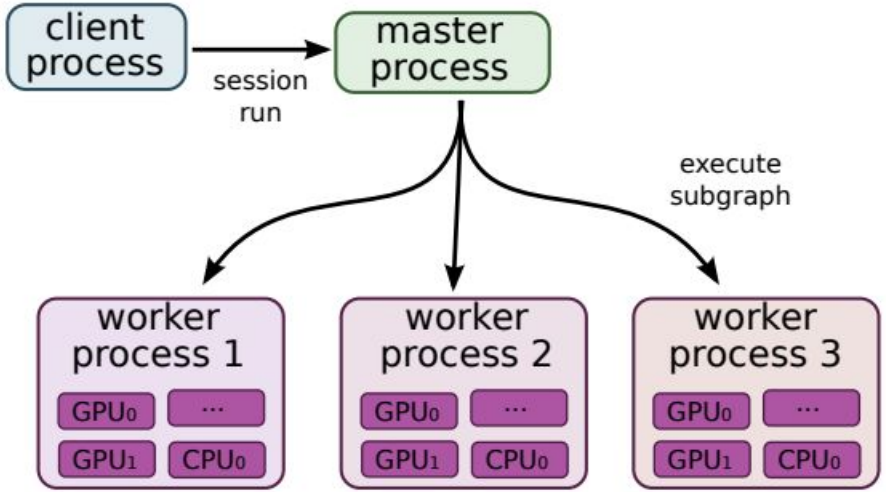


# Tensorflow - BackProp

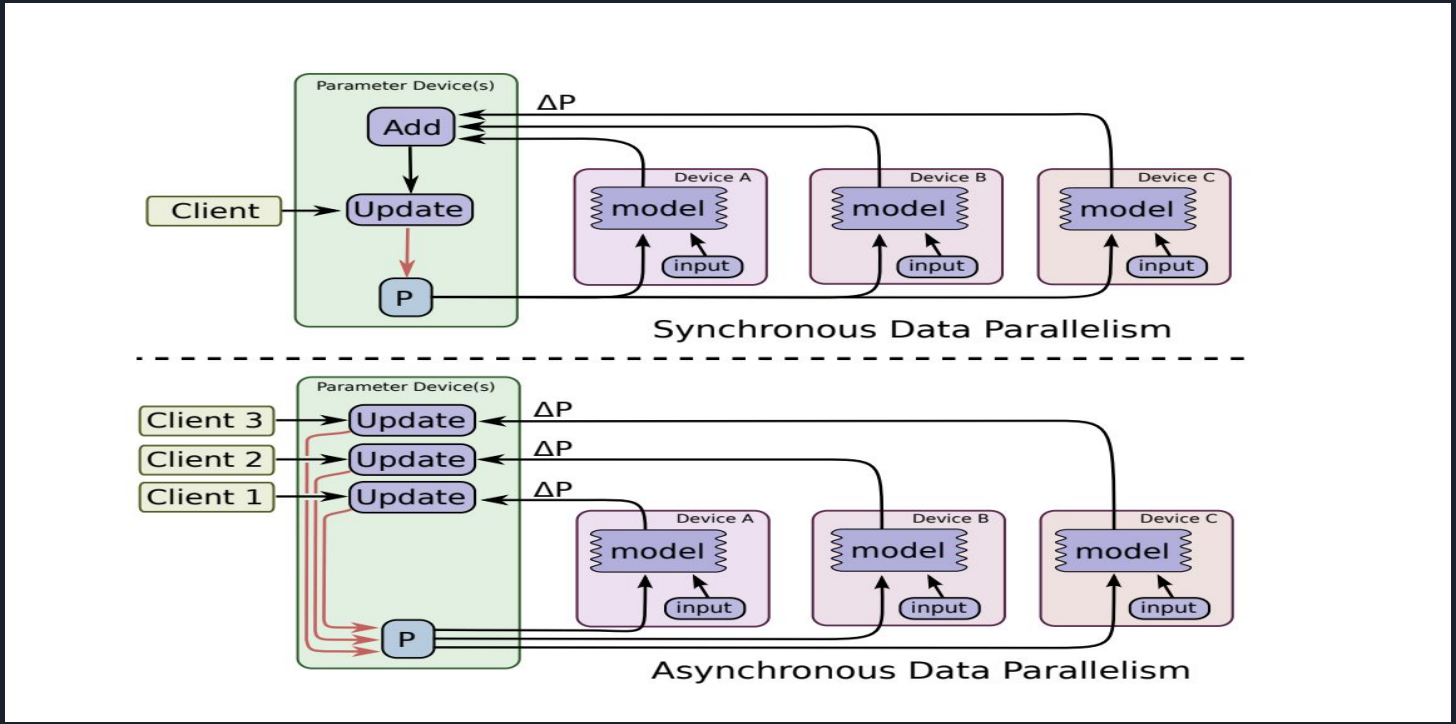


# TensorFlow - Distributed Execution



# TensorFlow - Data Parallelism



# TensorFlow - Model Parallelism

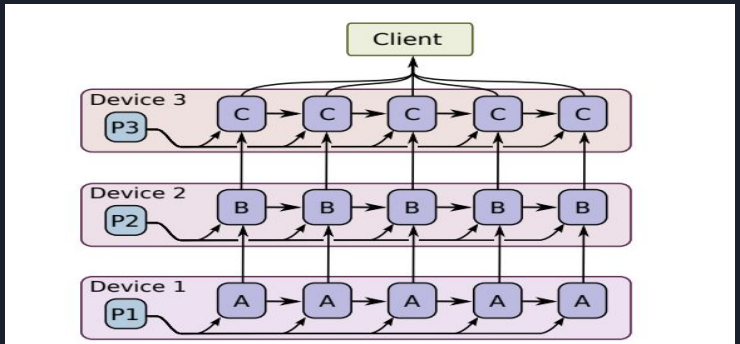
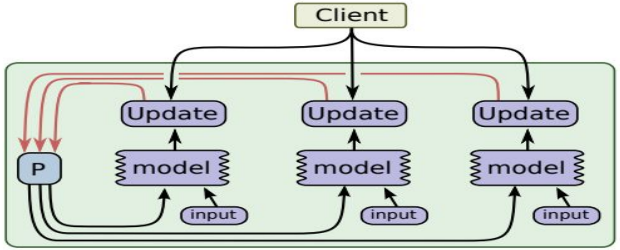



Figure 8: Model parallel training



# Keras

- Simplified Model Definition
- Base Layer and Model constructs
- Graph Definition and Submission
- Modular and Extensible
- Predefined Layers and Operations