RAVEN PROTOCOL

Decentralized and Distributed Deep Learning

Whitepaper

V1.0

May 2019

Rahul Vishwakarma and Sherman Lee

Abstract

In the past decade, Artificial Intelligence (AI) has achieved remarkable results in the areas of task specific intelligence such as: Natural Language Processing, Speech Recognition, and Computer Vision among many others. Right now, we stand at a turning point in the history of Artificial Intelligence driven by the trends of improved algorithms, exponential growth in computational power, explosion of data, and the development of AI hardware such as GPUs. These technical breakthroughs are paralleled by an unprecedented adoption of AI technology to solve real world problems.

Along with those factors, the AI community has contributed greatly to the rapid development and progress in the AI ecosystem. And, we at Raven believe that the community will be increasingly important going forward.

The AI community has come far in its journey, but it is only getting started. Companies all around the globe are adopting AI faster than they have adopted any other technology ever before. Today, each of these companies are spending millions of dollars every month in building and maintaining customized deep learning solution to offer better solutions and products to their users and customers. And this is ever-increasing. Unless, this changes with the adoption of a technology like that of the Raven Protocol.

The Raven Protocol distributes heavy deep learning training in the ecosystem using blockchain and incentivizes those who contribute their compute resources in exchange for Raven Tokens, by introducing a new protocol backed by a tested deep learning training distribution algorithm.

Table of Contents

1. Introduction

Raven Protocol is a decentralized and distributed deep-learning training protocol which provides cost-efficient and faster training of deep neural networks by utilizing the compute resources in the network.

1.1 Background

From its inception Raven Protocol has been able to solve (and keeps solving) a series of bottleneck problems that startup companies like Mate Labs[1] and Rocco.ai[2] faced in their constant training and deployment of AI models. They are the first of deep-tech companies to implement the Raven Protocol, among 30+ other Zeroth.AI's portfolio companies.

Mate Labs

Mate Labs has built a machine learning platform called Mateverse[3], so that non-developers can easily build, train and deploy custom Machine Learning (ML) and Deep Learning (DL) models without writing even a single line of code. They have built an intelligent algorithm trained to select the right ML/DL architectures by understanding the problem that a user is trying to solve. They have grown organically to over 8,000 users and the server costs to support that many users keeps growing and growing.

Rocco.ai

Rocco is an AI-powered social media marketer that suggest fresh content for brands to post on Twitter and Facebook. Rocco paves way to eliminate the need to hire human social media managers for marketing products and services. They are currently building a content generation platform which leverages ML to develop social media posts. As a startup, relying on AWS for AI-training became unsustainable.

More importantly, the understated and industry-wide problems that almost every company using AI are trying to solve had no solution anywhere in sight. Until now.

1.2 Problem Statement

A. Expensive Servers.

Deep Learning is an expensive and time-consuming process. With engineer salaries shooting above 150k USD, even the maintenance and running of such intelligent algorithms are expensive. Training a simpler 19 layer Convolutional Neural Network[4] on a million images (Imagenet[5]) usually takes around 2 weeks of continuous training. This translates to 336 hours on an ec2 instance[6] @2.6 USD per hour costing roughly around 1000 USD. This is just one trained model. In this industry, one million images is a paltry figure, because there are millions of different items on which image recognition needs to automate tasks. Moreover, these models needs to be maintained constantly at least once in a week to keep the models updated. Image classification is only one such case amongst hundreds of different ways where deep learning is applied, to transform and disrupt existing workflows and businesses.

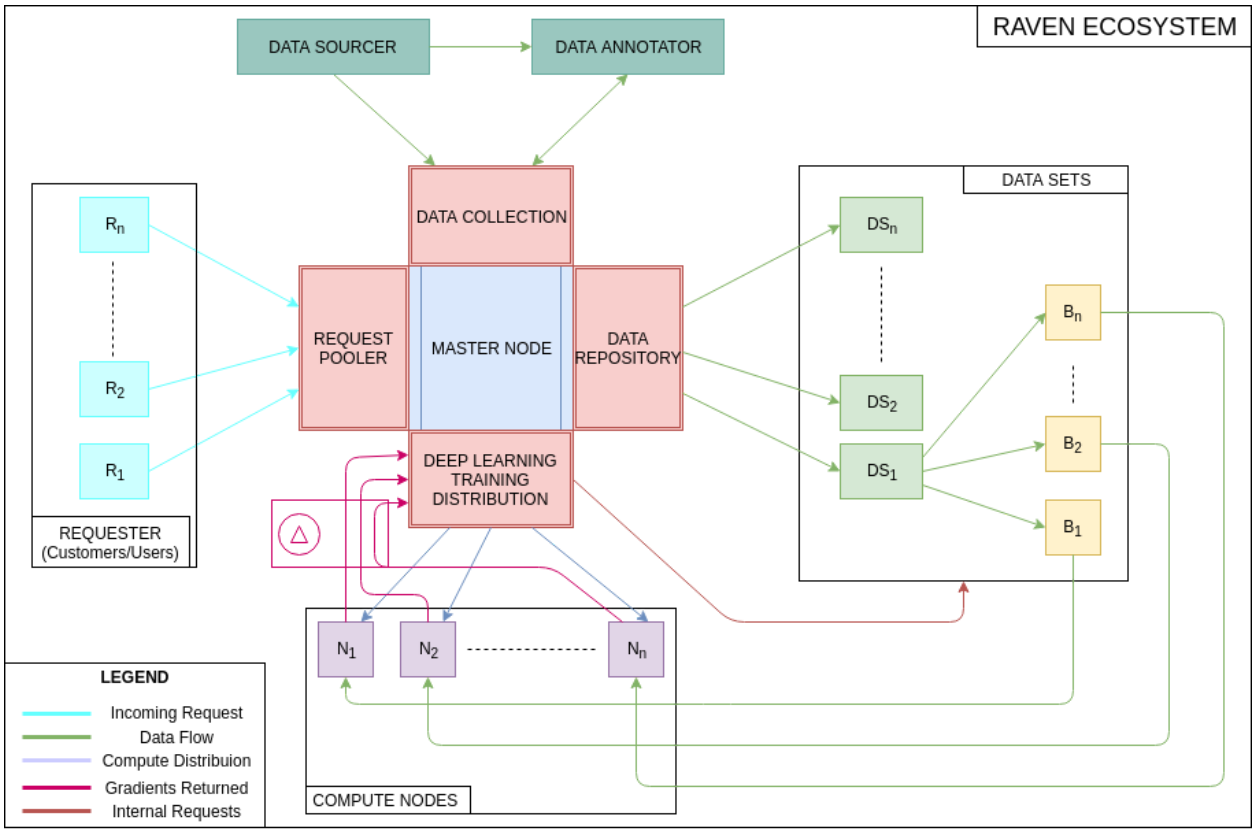
B. Training Speed

GP-GPU implementation to train the deep neural nets is one of the best features that has driven change in the overall AI ecosystem, thanks to Andrew Ng. The implementation single-handedly pushed the overall development by several years in exponentially accelerating the training of deep neural nets. NVIDIA's soaring shares is one of the biggest testament to this. NVIDIA itself is pushing the boundaries by constant development and various other programs to support the ecosystem.

Training deep neural networks on GPUs is still a localized way of training the network and can only scale so much given the limitation in the number of cores that can be fitted inside a block of a single GPU chip. We owe GPUs a lot of credit to the success and the mass scale adoption of AI today but, there is a need to match the computational acceleration used in training models with the rapid development of AI systems. Historically, proliferation of information was achieved with a network of computers, called the internet and cloud today. There's a need to speed up the training process for companies to be able to quickly provide personalized experience rather than pooling it for later; and make it cost-efficient for them.

1.3 Distributed Machine Learning and Data Sourcing & Annotation

The following diagram represents the functioning of the Raven Ecosystem and the flow of key elements in this network.



Key actors and roles within the ecosystem:

1. **Requester:** Users who want to train their deep learning models.
2. **Master Node:** Master nodes which act as a central source to integrate key elements of the ecosystem and orchestrate the distribution.
3. **Request Pooler:** Pools all the incoming requests and transfers it to the Master Node to make it ready to get distributed.
4. **Data Collection:** Collects the data from various Data Banks (here Sourcer) and pre-processes it to be stored in the standard format. Moreover, distributes it amongst Annotators who would annotate the raw data.
5. **Data Repository:** Post Collection, every data set (DS_i) flows through the Data Repository where it gets stashed into different batches (B_i) for it to be passed to different compute nodes.
6. **Deep Learning Training Distribution:** It is the powerhouse of the ecosystem. This is where the calculations that are needed to perform the training of a deep learning network is carefully distributed amongst compute nodes. It is one the most difficult tasks in deep learning training because all the calculations need to be synchronous for both forward and backward propagation. An analogy to this criticality is Flocking or Swarming where one wrong bird or insect behaving out of the order disturbs all others. Various nodes in the network return gradients that are necessary to calculate the weights of various neurons in the deep learning network.

1.4 Benefits

Decentralizing and distributing the training of deep Neural nets is an evolved form of localized AI development, much like a lot of other technologies in the past viz., the internet. With the best interests it will be accelerating the development and the adoption of AI technologies like deep learning. To count a few concrete benefits that Raven Protocol will have in the entire ecosystem:

1. **Faster Training**
2. **Cost Efficiency**
3. **Openness**
4. **Faster Updation**
5. **Standardized flow**
6. **Newer Applications**
7. **Increased experimentation**
8. **New AI-proof economy**

1.5 Marketplace/Platform

(Bridging all AI Services together, the hub for all AI development)

With Raven Protocol's innovative solution to the distribution of deep learning training, we are at a unique stage of bridging together the rest of the AI ecosystem. As such, we will develop the platform to allow all the different AI services to communicate with each other. This is enabled with the Raven Marketplace, which connects the different stakeholders in the AI ecosystem in a framework which promotes mutual benefit and proliferate creation of new applications. The primary stakeholders are as follows:

1. The people who want to **train a deep learning model**.
2. The people who want to **offer a trained deep learning model** to the network
3. The people who want to **use a trained deep learning model** on the network.
4. The people who want to **contribute excess computing resources to the training of various deep learning models**.

Further in the future, Raven Protocol will incorporate additional features that will involve more stakeholders such as, a dataset marketplace or, a data labeling and annotation marketplace. Irrespective of whether the complementary marketplaces are being built in-house, they will still be made available through integration with existing platforms within the ecosystem.

Critical to such an interoperability is the data and model format that each AI-service expects. Raven will create the layer to translate the interoperability seamlessly, as well as passthrough the costs associated with using each service. Meaning that, we will need an AI-stable coin developed at the back-end, in partnership with larger AI projects.

1.6 Token Utility

The RAVEN token will be the means to kick off all the activity in the Raven Protocol ecosystem. An AI-developer who wants to perform a training run on 1M images will need to send over RAVEN tokens to a smart contract. Contributors to the network who provide compute nodes will be rewarded with RAVEN tokens upon the successful training run, and is released via smart contract upon data and calculation verification.

Contributors running the compute nodes are ideally heavily involved in the AI community to drive forward progress in the technology. Thus, they will have a need to use those RAVEN tokens they've been rewarded to do AI training as well. The substantial value of tokens held in the blockchain gains more value as they are used in training other models. If the system is left to run on one node alone, the compute power will not be sufficient to conduct the training. The system is sustained with the use of subsequent nodes

throughout the blockchain. Thus, the flow of RAVEN tokens will be a complete circle and continue to be utilized inside of the network.

RAVEN tokens will also be used on the marketplace/platform to utilize additional services on top of the distributed AI training. Acquiring datasets, annotating data, using someone else's trained model, running an algorithm developed by the community, etc., can be accomplished by using RAVEN tokens.

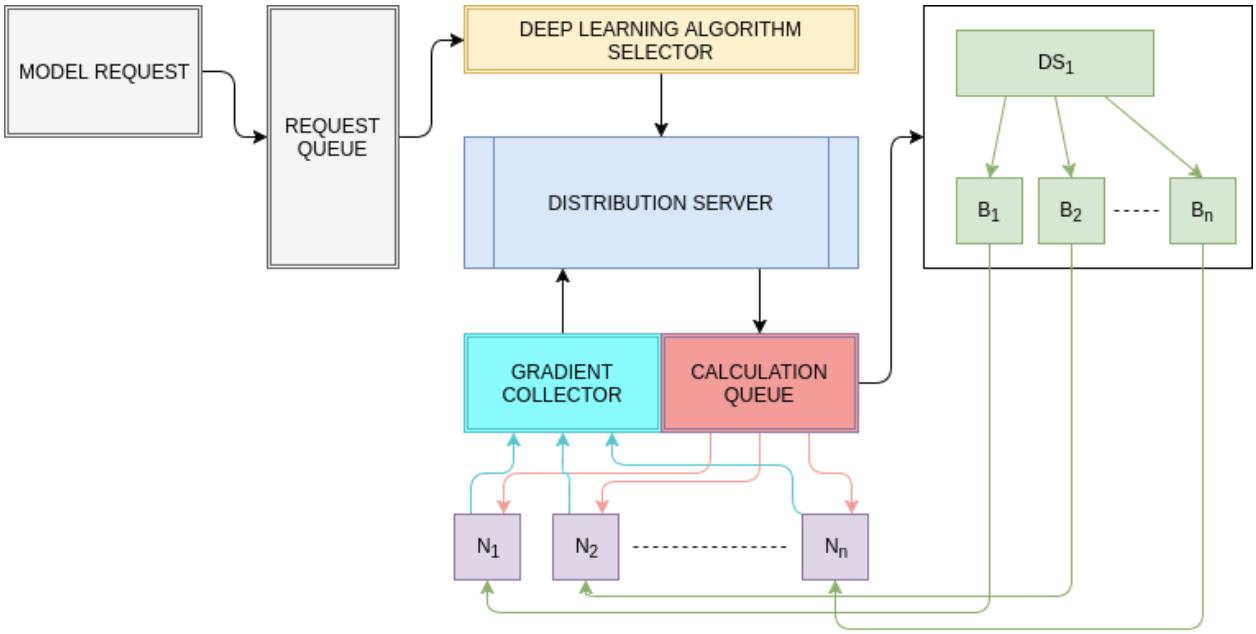
The special thing about this network is that those who contribute value to the AI-community will be able to utilize other valuable services in the ecosystem for their own AI-development needs. Thus, making the cost to do AI-development efficiently cheap and much more sustainable than paying for services like AWS.

2. Distributed Deep Learning

Raven Protocol introduces for the first time a democratic and crowd-sourced infrastructure that efficiently trains deep neural networks.

Further sharing a primer on the technology that makes distribution possible:

2.1 Technology



A Requester can request Raven Protocol to train a customized deep learning model. Once the request has been registered it gets queued in the Request Queue. Sequentially, each request is processed and goes through the Deep Learning Algorithm Selector, where an algorithm best suited for the incoming request is selected. Once the algorithm has been selected, it is passed to the Distribution Server. At the Distribution Server, a list of matrix calculation is prepared to calculate the gradients. Moreover the Distribution Server also calls for the batches (Bi) of the supplied dataset (DSi) through the Calculation Queue. The compute nodes (Ni) in the network are registered at the Distribution Server from where calculations are assigned to different nodes subject to the Calculation Queue. Different nodes return the results of different calculation in the form of gradients and it is registered at the Gradient Collector. The Gradient Collector and the Calculation Queue work in tandem, and calculations for different layers of a deep neural net is updated in the Calculation Queue seeking the status and results from the Gradient Collector via the

Distribution Server. The result of the process is a trained model with weights processed, which can then be exposed for the usage.

Following are few details related to incentivisation and data transfer:

1. Proof of Calculation

Proof of calculation will be the primary guideline for the regulation and distribution of incentives to the compute nodes (N_i) in the network. Following are the two prime deciders for the incentive distribution:

- **Speed:** Depending upon how fast a node can calculate the gradients and return it back to the Gradient Collector.

- **Redundancy:** The 3 fastest redundant calculation will only qualify for receiving the incentive. This will make sure that the gradients that are getting returned are genuine and of highest quality.

2. Compute Usage

Crowd-sourced compute resource gathering and allocation, is key to Raven Protocol and is subject to the following criteria:

- **Permissions:** Those participating by lending their systems as compute nodes (N_i) for training models, are requested explicit permission by Raven. Such awareness is created among users to differ from companies that have, in the past, taken advantage of user-ignorance. We have clear focus in being transparent and explicit on how much of the user's resources will be utilised by our network.

- **CPU/GPU allocation:** Allocating CPU/GPU resources are optional categories at the contributor's side. The allocation is also assigned more value in incentivisation as it aids training DL networks.

- **Throttling:** Resources allocated are calculated in percentages and those percentages can be throttled and controlled from the user's end. The number of calculations that are performed are in proportion with the amount of compute thus available. Allocation of resources are simultaneously registered within the blockchain, and incentivised accordingly. For example, if a user allows the network to use 5% of their computing resource, then the network would send 10 calculations per second, whereas if the user allows the network to use 10%, then the network will send 20 calculations per second.

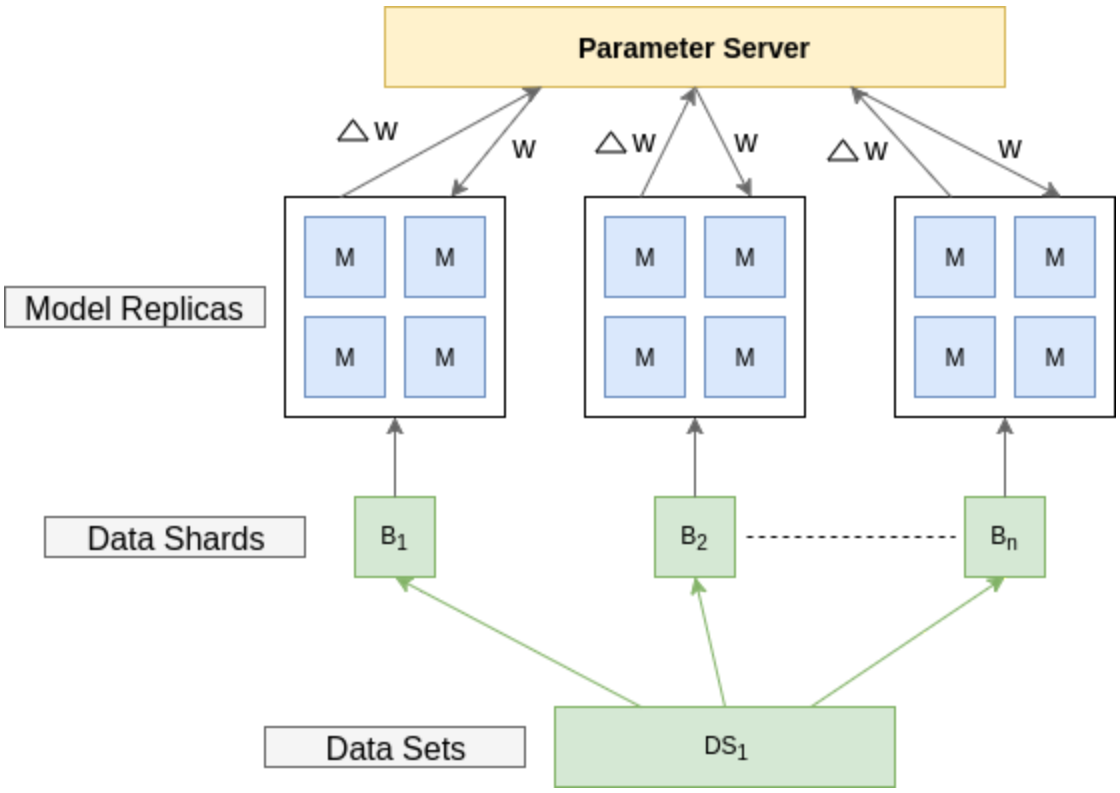
3. Comparative Analysis

Distributed deep learning is a novel methodology to be able to train gigantic models efficiently.

Deep learning is made possible today because of parallel computations. GPUs are all about doing hundreds and thousands of micro calculations at the same time. [ref: Andrew Ng's Paper].

We have come really far in terms of speeding up deep neural networks' training. There is still a long way to go from here. Developers are always coming up with creative hacks to utilize what's available at their disposal and do something exceptional with it. With a new parallelization framework, Raven Protocol is a complementary attempt to promote and help grow the AI ecosystem.

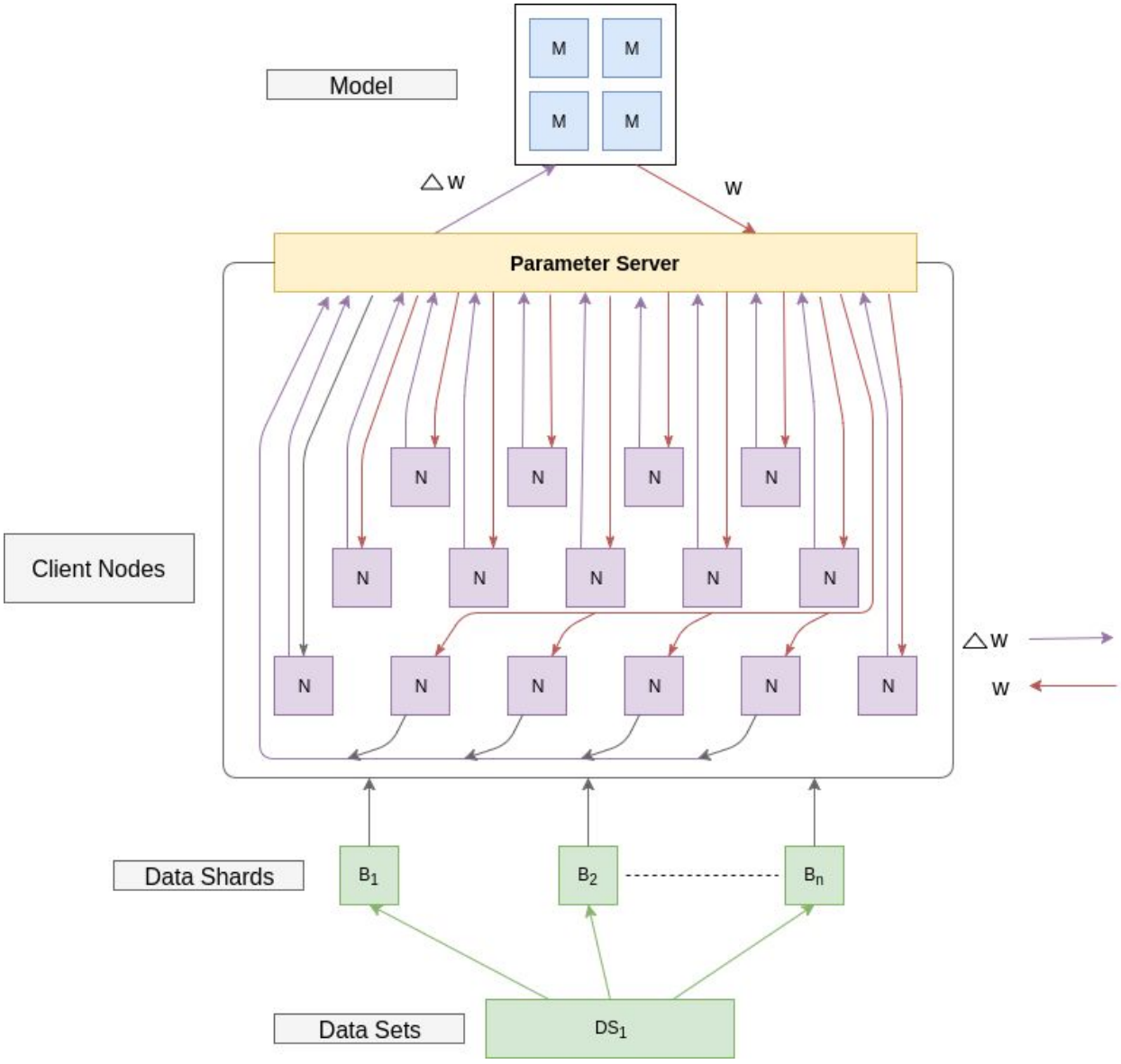
3.1 Data Parallelisation



Data Parallelisation with conventional deep learning frameworks.

In case of conventional deep learning frameworks like Tensorflow, data parallelization is an added feature and not an inherent property. Hence, to make it work, developers rely on conventional methods of processing large amounts of data, by sharding the data into several pieces. And, to make this work with deep learning, a replica of the subject deep learning architecture is kept at every client node. Finally, all the results are collated at the Parameter Server.

However, Raven Protocol attempts to do it differently by making the optimization and speed of the training process an inherent property of the platform; letting developers and researchers to focus on discovering new algorithms.

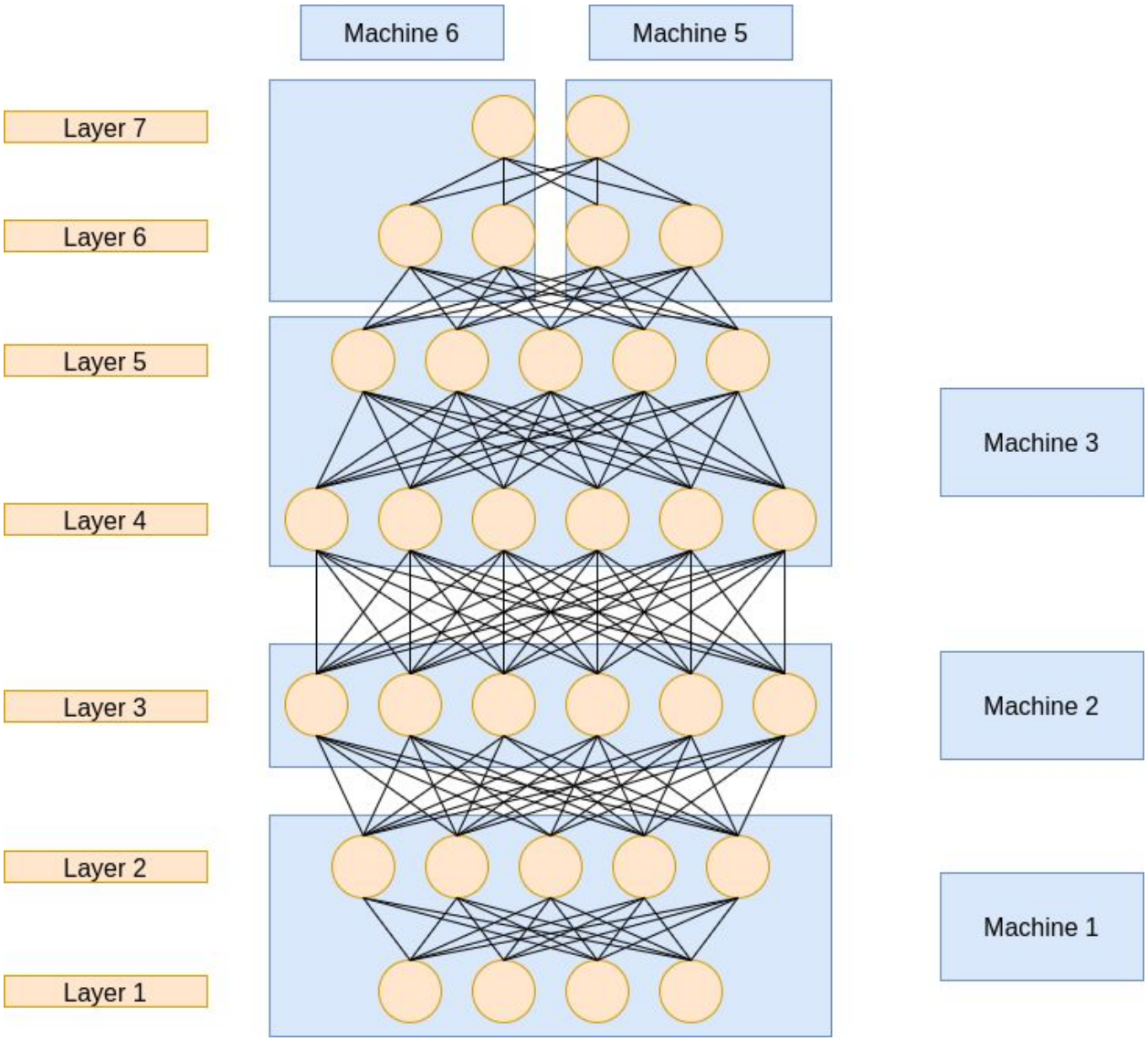


Distribution with Raven Protocol

Distribution of data within Raven, is executed by dividing the data into smaller snippets. They are then distributed to various Client Nodes. Clients, subsequently, calculate the gradients and sends it back to the Parameter Server. Something to note here is that the ML model/architecture lies at the Parameter Server and is updated dynamically with the gradients collected from every Client Node.

3.2 Model Parallelisation

Model Parallelism is another way of distributed training for large scale deep learning. Traditionally, this was achieved by keeping the dataset in a different system, which is



Model Parallelisation with conventional deep learning frameworks.

accessible to other machines, among which a deep learning architecture is spread, as shown in the figure above.

With Parallelisation in Raven, the subject DL Architecture lies at the Parameter server itself and calculation are spread out.

3.3 Graph Construction (Dynamic vs Prebuild)

All the frameworks operate on tensors and are built on the computational graph as a Directed Acyclic Graph. In most of the current and popular deep learning frameworks including Tensorflow (before Eager Execution), the computational graph is static in nature. However, frameworks like PyTorch is dynamic, giving a lot more options to researchers and developers to fiddle around with their creativity and imagination.

A major difference between the static and dynamic computation graph is that in the former, the model optimization is preset and the data substitutes the placeholder tensors whereas, in the latter the nodes in a network are executed without a need for any placeholder tensors. Dynamic computation holds a very distinguishable advantage in cases like language modelling where, the shape of the tensors are variable during the course of the training.

Raven Protocol exploits the dynamic nature of the graph computation and takes it to the next level and utilizes this property to make the micro-distribution possible.

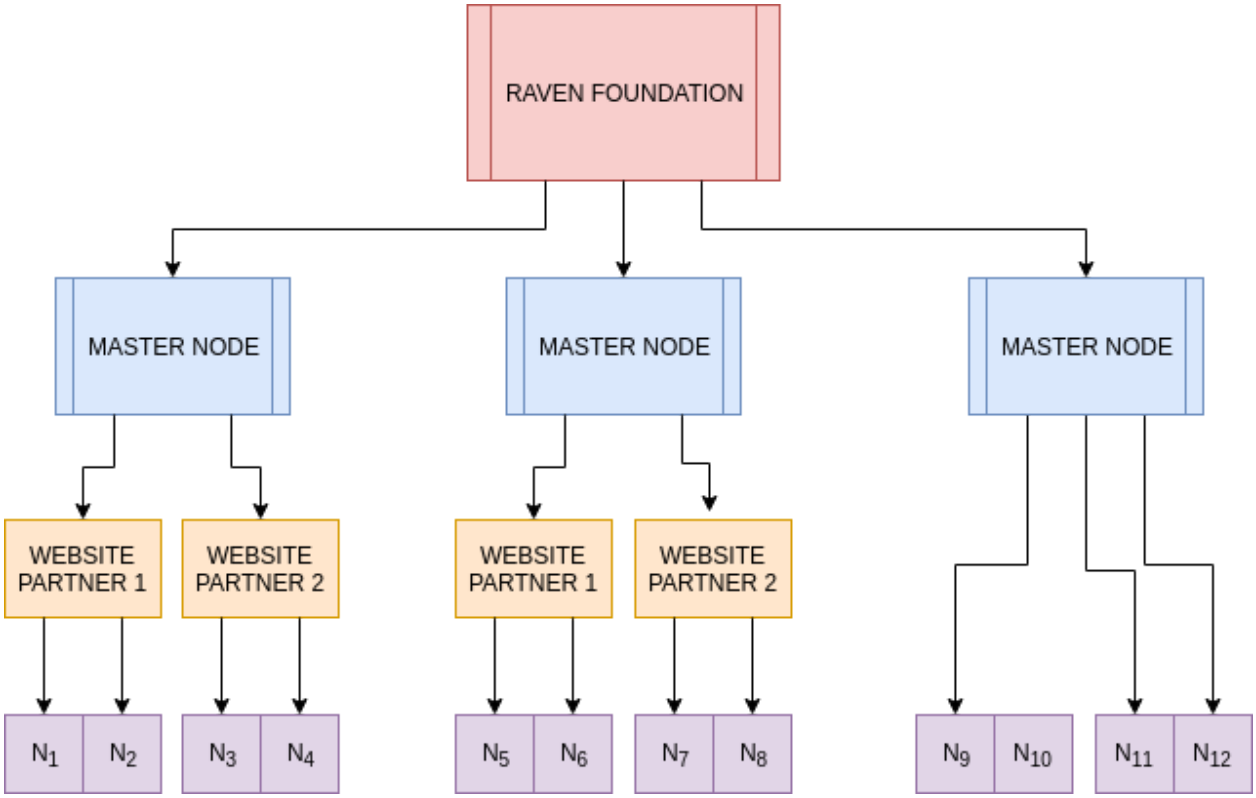
4. Transaction Framework

Raven Protocol is trying to create a self-sustaining and dynamic ecosystem for Customers who want to train their AI engines regularly to keep them updated, Contributors who would like to support the ecosystem by sharing their compute resources in the form of Computers, Smartphones or even a server rack. And, Contributors as Partners in the form of various website owners who are willing to support the Raven platform by inviting their network of users/viewers.

To help them build this cohesive relationship, Raven (RAVEN) will work as the common ground to facilitate a secure transaction that may take place inside Raven Ecosystem.

To provide a fair ground for the transactions, various costs inside the Raven Ecosystem will be pegged to USD, which will provide transparency to the customers.

Following is tentative structure of the transaction framework between key entities operating in the Raven Ecosystem:



Key Entities:

1. **Raven Protocol:** which will be responsible for the constant development, so that the customers, contributors and other partners get the seamless experience.
2. **Master Node:** which will be responsible for orchestrating the training of various deep neural networks.
3. **Website Partner:** These are consumer websites, who would want to contribute to the progress of AI ecosystem by extending the Raven Network to their users/viewers; hence it will provide compute resources.
4. **Compute Nodes:** The most important part of the whole network, they'll be sharing their compute resources which will finally deliver the promise of Raven Protocol.

Following is how the Cost to the Customer and Earnings will be calculated:

Let us assume a hypothetical training: T

Unit Calculation: C_i

Total No. of calculations needed to complete the training: $C_n = \sum_{i=1}^{n} C_i$

Operation Price per calculation: R_i

Total operational cost: $R_T = C_i + R_i$

Operating Margin: $M = 10\%\ of\ R_T$;

includes Master Node - 5% + Website Partner - 3% + Transaction fee - 2%

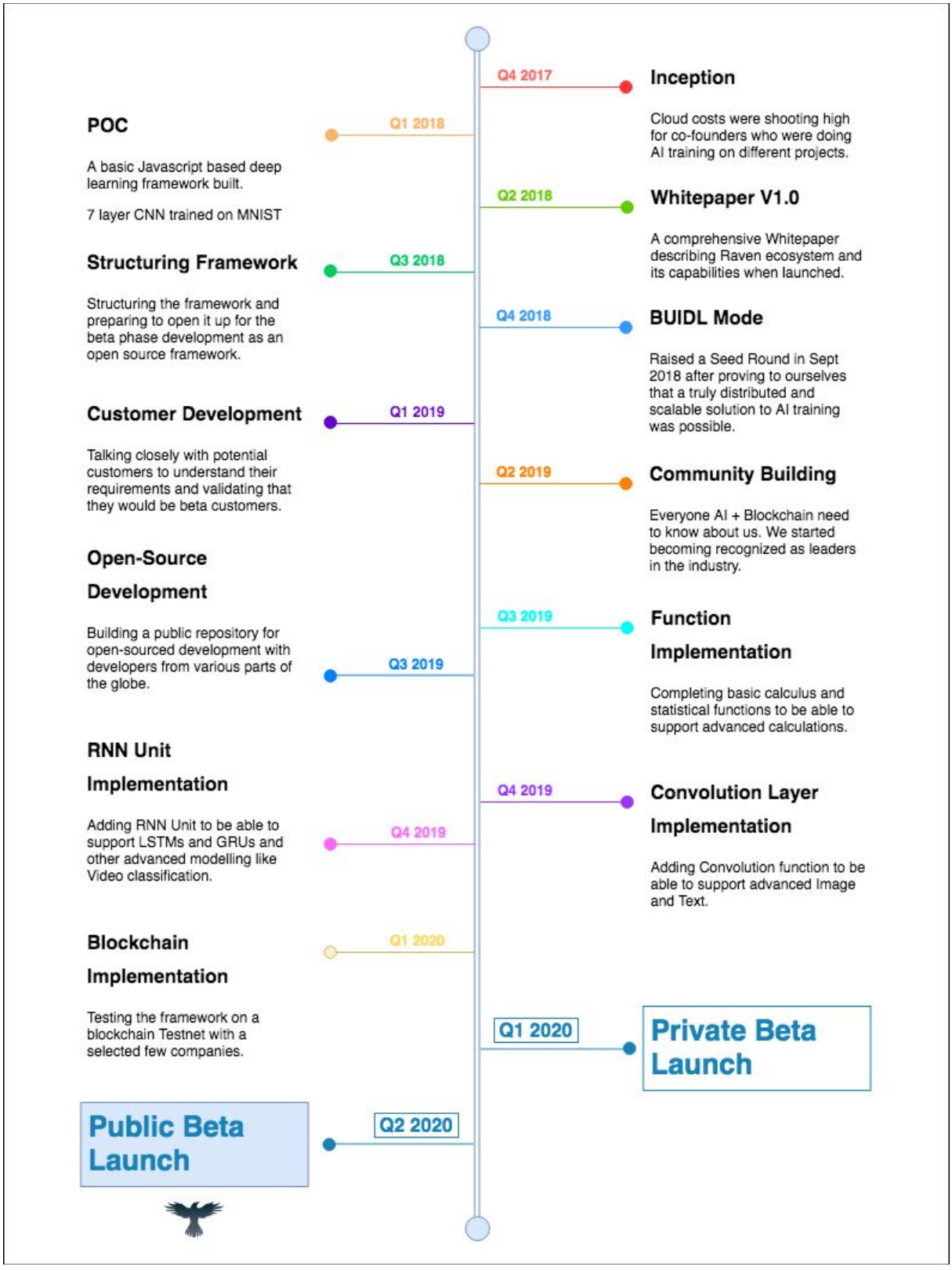
Total Cost to Customer: $R_S = R_T + M$

Total No. of (+)ve calculations, calculated at a Contributor node (N): $C_n = \sum_{i=1}^{x} C_i$

Contributor node's (N) total earnings: $\frac{\sum_{i=1}^{x} (C_i \times R_i)}{\sum_{i=1}^{n} C_i}$

Above summarizes the financial structure inside the Raven Protocol ecosystem.

5. Development Roadmap



6. Conclusion

Creating and establishing a protocol for communication among AI blockchains which allows the sharing of essential resources of compute power will set the future standards for all AI applications. The Raven Protocol will be that standard.

1. Raven Protocol provides a channel for distribution of the training of deep learning algorithms. The contributors to the network share their compute resources to enable training for users in the network and are in turn rewarded with Raven (RAVEN) coins/tokens, through the blockchain. Currently, those who enable distributed AI training in the market today, either train the entire deep neural net architecture on a single node or split the architecture into smaller segments to be then trained on servers that have ample heavy-compute power. **Raven Protocol, in contrast, enables a single deep learning architecture to be split up to the micro level and trained across the many dynamically allocated nodes within the Raven network — the first truly distributed training algorithm in the world.**

2. With a distributed network of computing resources, Raven Protocol creates a platform to connect all AI services together, bringing together the larger AI community as well. **Raven Protocol is the central hub for performing AI training and development. The platform will integrate AI services such as data exchanges, data labeling/annotation services, shared models, and implemented algorithms** that may be developed by other projects or members of the AI community. All integrations with the platform will be supported by the Raven Protocol and allow for seamless communication between services.

3. Raven Protocol opens up a platform for individuals to contribute idle compute power to the network; adding value by aiding AI models' training. They are then compensated by those who use the resources, which are shared using your own devices such as phones/laptops, or any other. **With the correct incentivisation in place, AI training can be accessible to the the masses either for free or at deep discounted rates**; this would undoubtedly push AI technology to levels unimaginable at present.

END