GNY.IO's DECENTRALIZED DISTRIBUTED MACHINE LEARNING BLOCKCHAIN

One benefit of blockchain's decentralized technology is its application beyond cryptocurrencies. We are applying the techology to machine learning – to enable a smarter "thinking" blockchain process. Most machine learning models that have been used have an underlying assumption that some global knowledge is required by its algorithms to function (see fig 1). These are centralized distributed, where the entire data set is loaded into a cloud of distributed nodes so that the two main functions of machine learning are performed in one large library of functions ETL/Exploration and Model Training/Parameter Tuning. Spark, Watson, Azure all use this platform based approach and we are also biased towards methods that have been known to work.

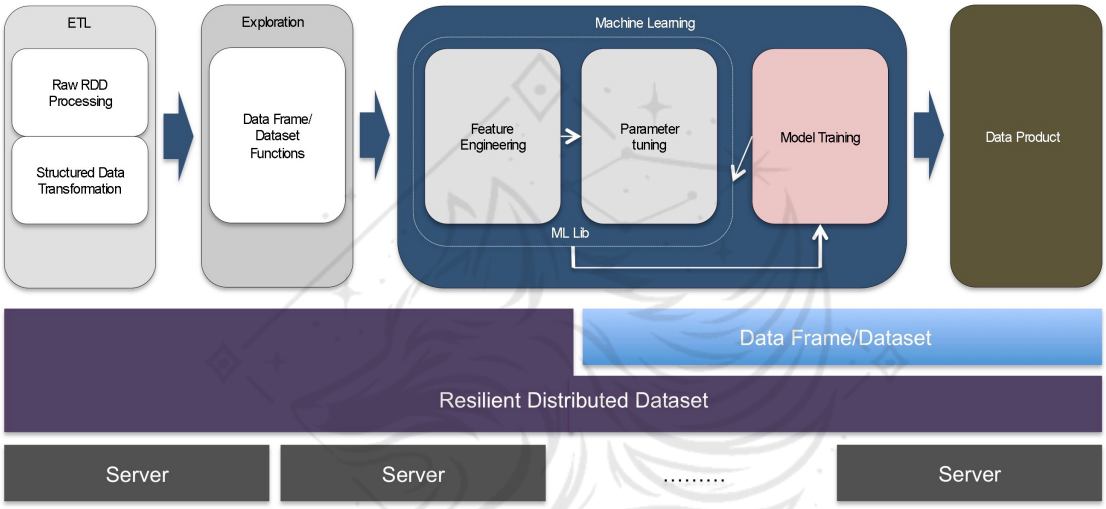


Figure 1: A typical Spark machine learning architecture

gny.io's DistributedDeep Learning breaks this pattern of one large platform library by creating two of the smallest configurable self-learning unsupervised neural net nodes – ETL node and ML node – and distributing these nodes into each block of the block chain to have them teach themselves the solution to each problem. The one conceptual problem though of machine learning is that error detection requires global knowledge that gets back-propagated to its constituents. This requirement though is fixed in gny.io's DistributedDeep Learning systems. Therefore we prefer the term 'localized models' and not 'model free models', even though both mean a measure of lack of intelligence of constituent parts. In general though, this is similar to the concept of 'parsimony', in that we seek the simplest of mechanisms that give rise to emergent properties of prediction. Gny.io is not a crushed up small machine learning library; it is one node of a huge distributed brain.

Gny.io has configurable ETL microservices and configurable machine learning microservices that can read the entire chain of data or it can read the current block data. Gny.io's microservices uses deep learning which is a class of machine learning algorithms in the form of a neural network that uses a cascade of layers (tiers) of processing units to extract features from data and make predictive guesses about new data. The smallest ML node system varies the weights and biases to see if a better outcome is obtained using a neural network (see Fig 2).

Archived copy · cryp2nova.com

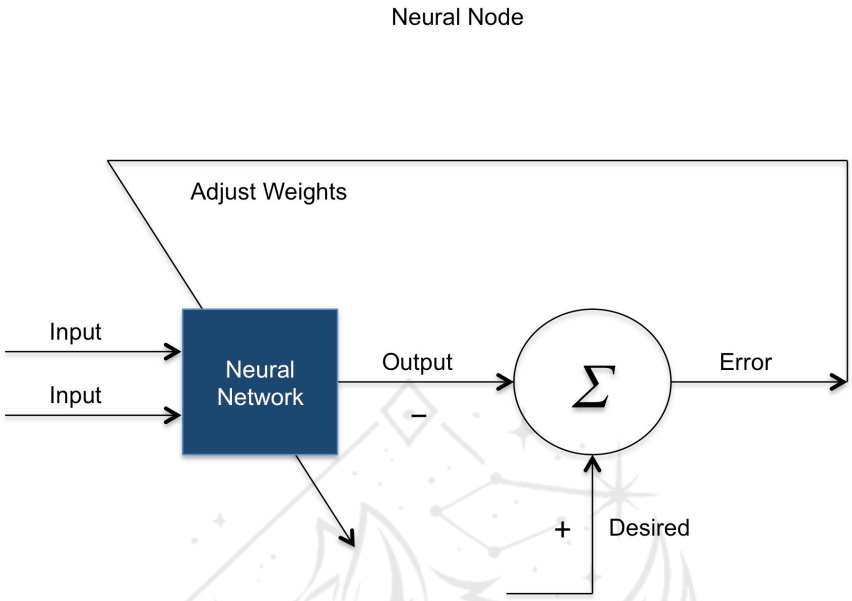


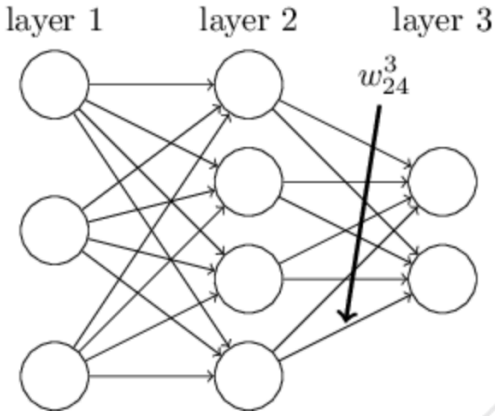
Figure 2: Neural node in a neural network

At the heart of Gny.io is the special backpropagation algorithm. Backpropagation of errors and gradient descent are some of optimization methods used to calculate the error contribution of each blockchain node after a batch of data is processed. Gny.io uses a variation of structured streaming AI and parquet data format - this sets the weights in the neuron. This means that the neurons change their connection and their weights to lesson the error. Gny.io will determine the normal rules of the system by itself with unsupervised learning. Adjusted weights is what is configurable.

Backpropagation is an expression for the partial derivative $\partial C/\partial w$ of the cost function C with respect to any weight w (or bias b) in the network. The expression tells us how quickly the cost changes when we change the weights and biases. And so backpropagation isn't just a fast algorithm for learning. It actually gives us detailed insights into how changing the weights and biases, changes the overall behavior of the network.

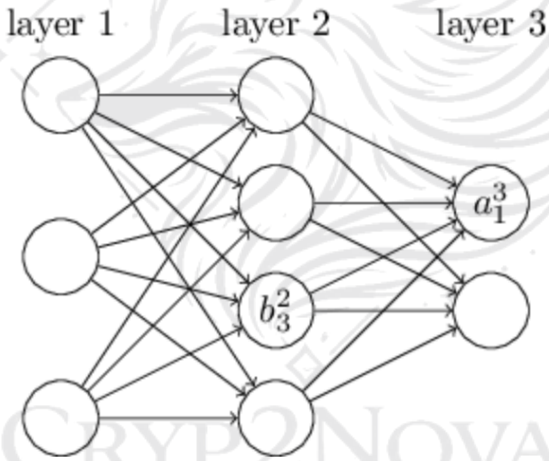
Let W(l)jk be the weight we are trying to find to kth neuron in the (l−1)th layer to the jth neuron in the lth layer. So, for example, the diagram below shows the weight on a connection from the fourth neuron in the second layer to the second neuron in the third layer of a network:

Archived copy · cryp2nova.com



w^l_{jk} is the weight from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer

Let *blj* for the bias of the *j*th neuron in the *l*th layer. And we use *alj* for the activation of the *j*th neuron in the *l*th layer. The following diagram shows examples of these notations in use:



The goal of backpropagation is to compute the partial derivatives *∂C/∂w* and *∂C/∂b* of the cost function *C* with respect to any weight *w* or bias *b* in the network. The cost function is expressed as:

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2$$

Archived copy · cryp2nova.com

where: *n* is the total number of training examples; the sum is over individual training examples, *x*; *y*=*y*(*x*) is the corresponding desired output; *L* denotes the number of layers in the network; and *aL*=*aL*(*x*) is the vector of activations output from the network when *x* is input.

What Gny.io does is compute the partial derivatives *∂Cx/∂w* and *∂Cx/∂b* for a single training example of in last blockchain connection. This allows us to use the equations to *design* activation functions which have particular desired learning properties.

Each block in the blockchain contains one neuron and the blockchain is the neural net - each smart contract is an account that has its own code - so with the neural nets.

Unlike Watson, Azure or Spark there is no controlling machine learning manager cluster and no data lake. The neural nets are self organizing and self correcting. Gny.io takes the data partition idea of parquet to its logical conclusion and performs massive parallel pattern learning that sets the weights of the neuron.



Figure 3: Gny.io's parquet data format

Archived copy · cryp2nova.com

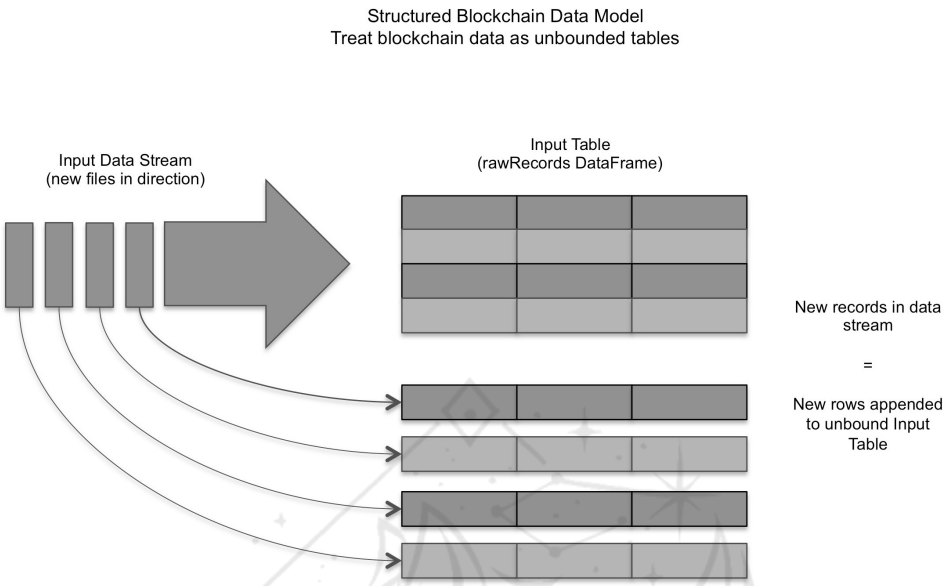


Figure 4: Parquet data flow

Gny.io's gradient descent optimization is Deep Learning. Deep learning is an advanced form of Artificial Intelligence and a powerful set of techniques for learning in neural networks. Neural networks and deep learning currently provide the best solutions to many problems in image recognition, speech recognition, and natural language processing,

Gny.io is a fusion of Deep Learning technology on to a blockchain. The blockchain, being a distributed transparent consensus based peer-to-peer network that has been shown to be extremely resilient to adversarial attacks. The blockchain itself creates the neural net. Gny.io is the first to apply this technique in blockchain.

Gny.io uses a Probabilistic Graph Model (PGM) approach to DL. In the PGM approach, the neural nodes construct a probabilistic graph that defines the relationship's different variables. The approach uses Monte-Carlo sampling to construct Bayesian consistent distributions for the variables. We then use Deep Learning to learn from this synthesized data.

Archived copy · cryp2nova.com

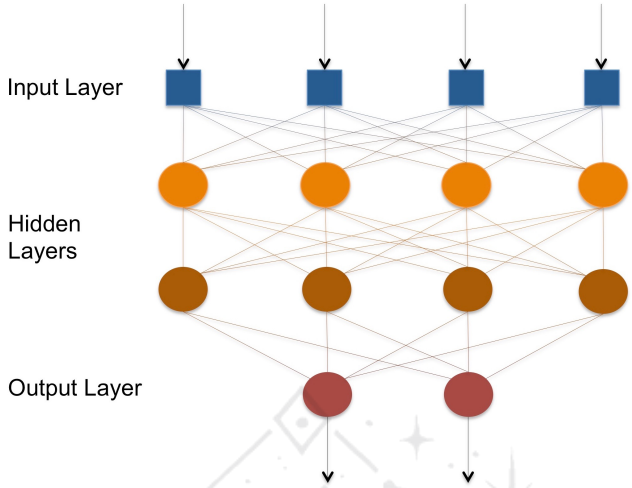


Figure 5: The "Layer Cake" of a neural network

Gny io has:

Embeddedness - The neural nets do not exist in isolation and are embedded in a much larger blockchain ecosystem.

Redundancy - Duplication of components may be inefficient however it provides the mechanism to handle the unexpected. In addition, functional redundancy offers a way to repurpose components to reduce costs.

Heterogeneity - Different predictive machines make it possible to react to a more diverse range of change as well as avoid correlated behavior that can lead to total system failure. Diversity is required for evolutionary learning and adaptation.

Modularity - Decoupling of components act like firewalls between components and help mitigate against total collapse. Individual component damage can be tolerated while the integrity of other components are preserved. In general, a distributed loosely coupled system has higher survivability that a centralized tightly coupled system.

Adaptation - A system needs to be sufficiently flexible and agile to adjust to changes in the environment. Adaptive approaches that involve simulation, selection, and amplification of successful strategies are important. Self-learning is requirement to achieve adaptability.

GNY blockchain application platform

Archived copy · cryp2nova.com

GNY is a blockchain application platform, which will help enterprise or developers to build their own blockchain application in JavaScript.

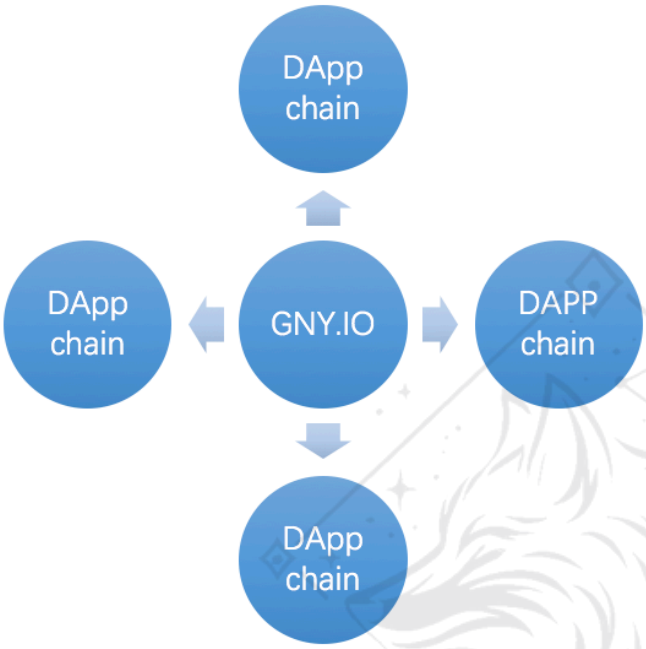


Figure 6: GNY.IO blockchain architecture

Each DApp chain is a parallel blockchain to the GNY core chain. GNY core offers the SDKs and APIs to help developers to develop their own blockchain. GNY also offers different consensus algorithms as components.

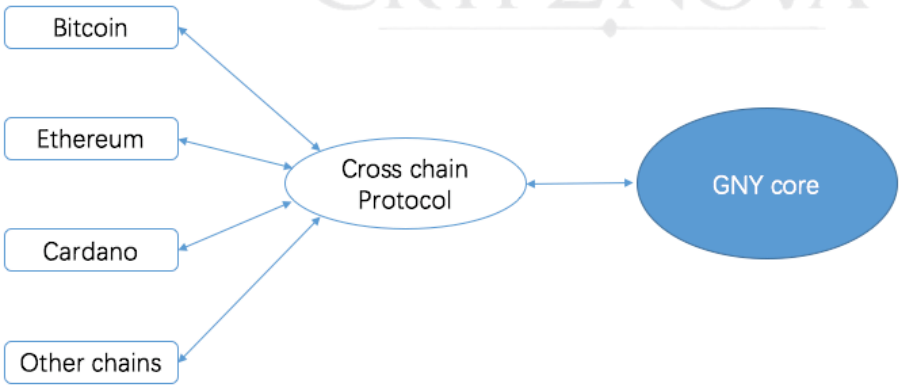


Figure 7. cross chain protocol in GNY

Archived copy · cryp2nova.com

We believe the future of blockchain is a world that different blockchains can be connected easily. We will develop the cross chain protocol, which will connect GNY core to many other mainstream blockchains. Every DApp developed based on GNY can import other tokens like BTC, ETH to their DApp.

GNY.IO will adapt the DPoS + BFT consensus algorithm, which will have 21 super nodes who maintain the network. There are many other lightweight nodes who will handle transactions to the network.

CRYP2NOVA

Archived copy · cryp2nova.com