Genesis Space’s Consensus Mechanism

Using Artificial Intelligence Technology for Consensus Mechanism (POC)

2017.12.31

Genesis Inc. (TBD)

Address： 91 E Brokaw Rd， San Jose， CA 95112

Phone： 516-777-0588 Email： thegenesisspace@gmail.com Website： thegenesis.space

Introduction

Mining of cryptocurrencies consumes around 70% computing power from the whole crypto network. This is also one of the reasons that cryptocurrencies are blamed. Huge amount of electricity is consumed and the environment is polluted. However, mining doesn't have any value other than POW.

On the other hand, massive data are produced globally, but these data are always fragmented. These data must be cleaned and transformed appropriately before they are analysed. The data cleaning and transformation process usually consumes lots of computing power.

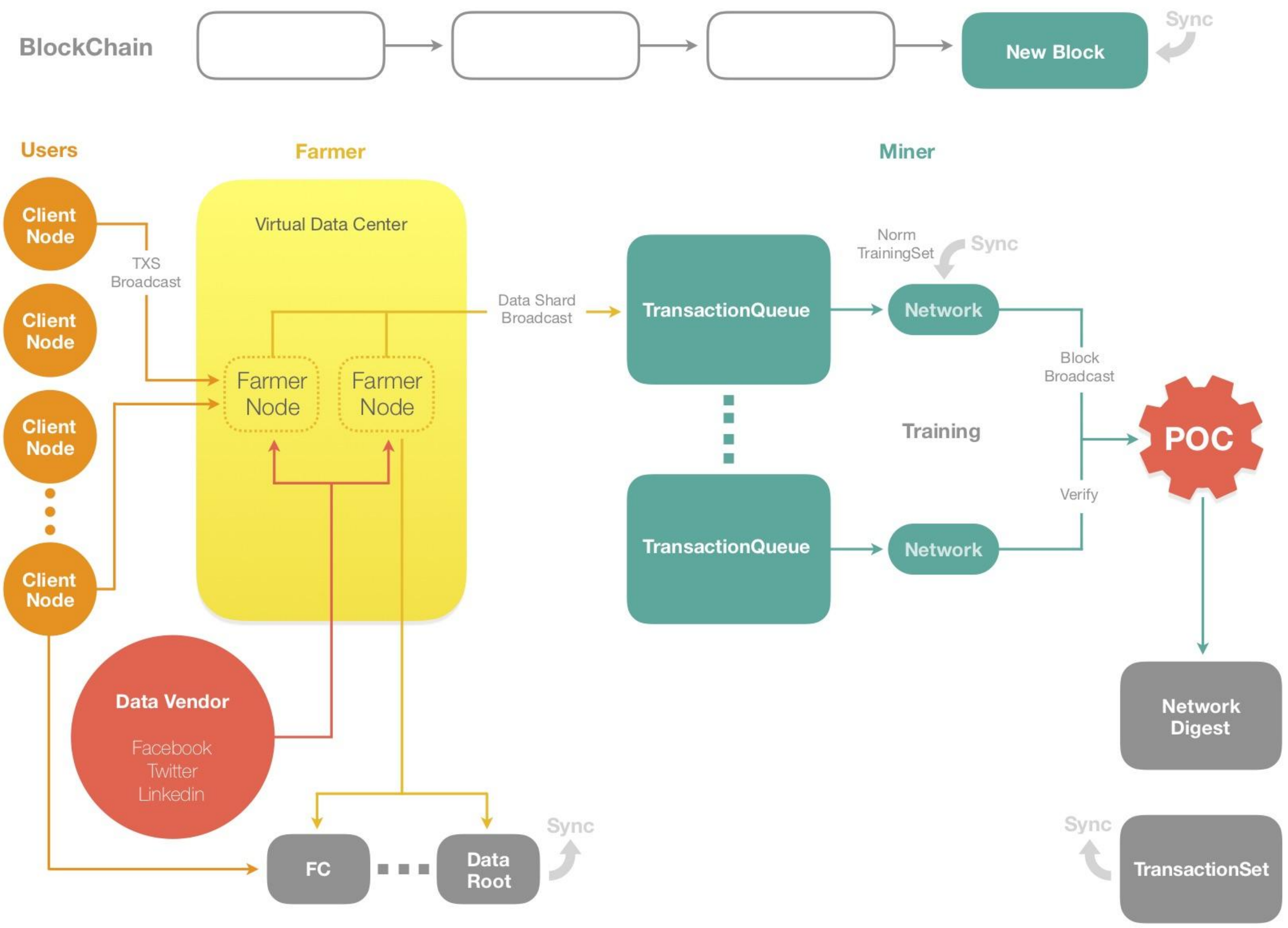
Genesis combines the mining with the data cleaning and model training process. We make the model training process of AI system work as the proof of work for Blockchain. In a nutshell, Genesis is using mining of a blockchain to gather computing power to solve practical problems.

Genesis is the first blockchain project that introduces the concept that the model training process can be used for consensus mechanism for blockchain. POC (Proof of Comprehension), Genesis' consensus mechanism, can transform POW with no practical value to data model training process. POC has huge economic value. Wikibon predicted that the market value of big data industry is 45.4 billion. According to McKinsey's analysis, big data industry can unlock a value of 54 trillion USD globally.

Process Realization

Our purpose of creating Genesis is not to solve a problem in a specific scenario, but to create a platform for different development teams to contribute unique value to various industries. We believe that POC will be applied to scenarios that requires massive computing power.

Below is the flow chart of Genesis POC:



Data Acquisition and Model Output

Genesis' dapps, from its dapp store, are the data source for the whole system. The third-party application development teams can develop based on Genesis' dapp development platform and provide the data upload channels. Meanwhile data providers can get rewards from dapps.

Data are stored in a decentralized file storage system. In Genesis, every farmer is a maintainer for the data source. Basically, Genesis has a real-time expansion, decentralized file storage system which is maintained by all farmers.
After POC trains the model with data, dapps will show the output.

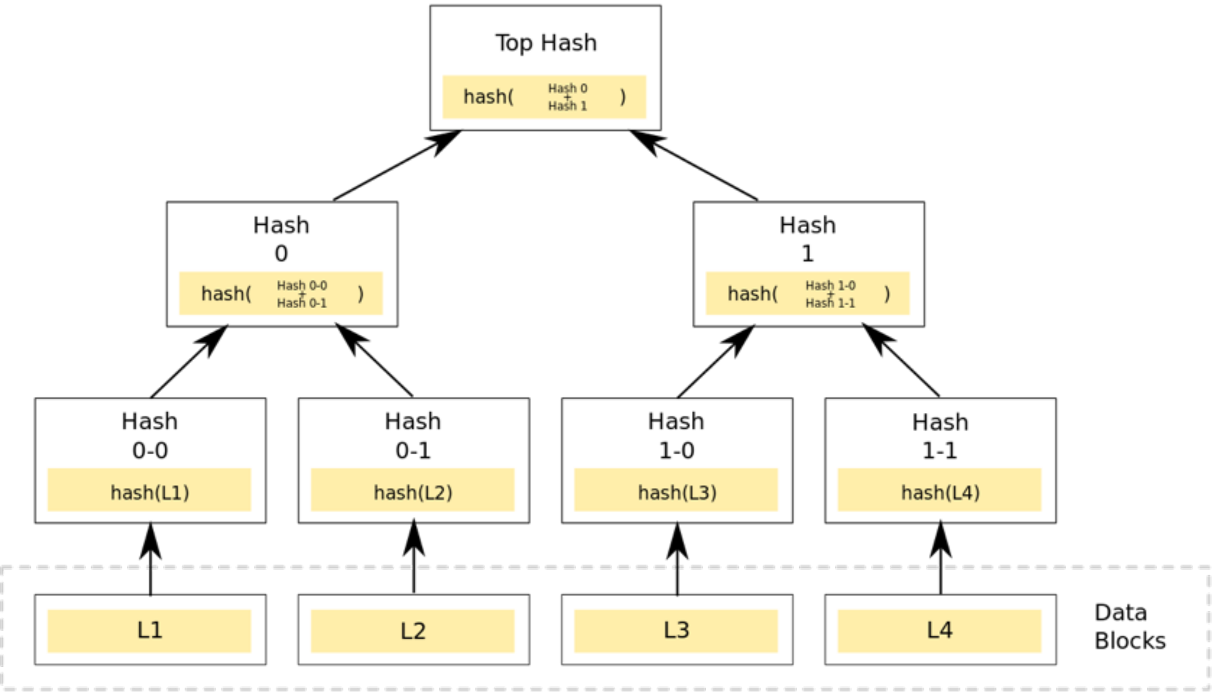
Data Storage

In Genesis, data are stored in a virtual data centre for POC.

1. Virtual Data Centre

Nodes in Genesis that are used to store data are called storage node. Storage node and normal node share the same address. Dataset are separated and copied to storage nodes. The cluster of storage nodes is called virtual data centre. Every time a user submits a transaction, he or she needs to sign a store contract with virtual data centre to rent the storage space. This technology can make sure the scalability and the data integrity.

2. Merkle Tree



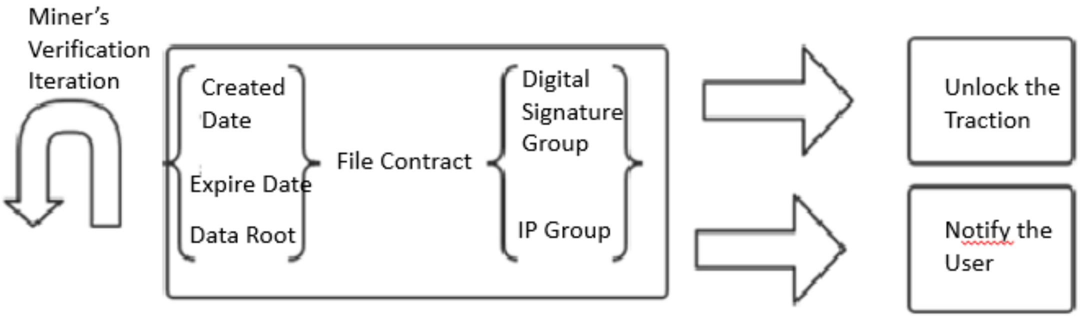
Merkle Tree is often used to verify the non-credible data source. Merkle Tree has following features:

- All leaf nodes' values are data blocks with same length.
- Values at the non-leaf nodes are calculated by all leaf nodes' values below it with a hash algorithm. If the son node of the non-leaf node is also non-leaf node, the value of this node is calculated by hashing its son-node.
- User can use one branch path of Merkle tree to synchronize data. This path is called verification path. Every verification path is verified by the root node.

3. Data Storage

Data in Genesis are stored in storage nodes in the form of Merkle Tree. The storage node broadcasts its IP to the whole network after it is invoked. Normal nodes put the more credible storage nodes in a nodes queue as a part of the virtual data centre. Then the normal nodes will synchronize their own Merkle Tree to virtual data centre and sign the storage contract after synchronization finishes.

4. Create and Publish Storage Contract



Storage contract is the protocol between storage nodes and normal nodes which rent storage space. The content of the protocol is saved as script in the payload of the contract and it includes the responsibilities each party should take. The contract need to be signed by the data provider and the virtual data centre before it is published. Once the contract is published, it cannot be edited again. The content of the contract includes:

- Head of the contract hash
- The length of the blockchain when the contract is created
- Rent cycle time (calculated by the blockchain length)
- Root of the Merkle Tree for data verification
- Digital signature group (from data provider and the virtual data centre)
- Group of storage nodes' external IP address
- Double locked script (The data uploader double lock the reward within a period through the public key combination's verification path. The miner need to use the private key to unlock.)
- Storage contract are published to blockchain as transaction.

5. Implementation of storage chain and storage contract

The sidechain for maintaining storage contracts is called storage chain. Storage chain is maintained by miners. When miners create new storage blocks, they will verify the data that are pointed by part of the current storage contracts. If the verification succeeds, miners will publish a transaction with a special verification path to the main chain and the transaction will copy the double locked script and unlock it again. This process is called the execution of the contract. The client side is monitoring the transactions and the execution of the contract. The storage contract will be removed from side chain once it is expired. When the end date of contract is approaching, user can choose to extend the rent period.

6. RS code correction

The virtual data centre's storage nodes can provide wrong data pack because of network jitter or data corruption. Genesis adopts RS code to correct data. RS code is a simple but useful algebraic programming mode. RS code is widely used to correct errors in different digital communications and storage systems.

POC (proof of comprehension)

Genesis's POC mechanism is based on deep neural networks' model optimization and logic verification. Below is the detailed description of how it is designed.

1. Deep Neural Network

Machine learning based on DNN (Deep Neural Network) [1,2,3] has already been proved to be effective in the field of image recognition [4,5] and NLP (Natural Language Processing) [6,7]. Neural networks model emulates human brain's synapses for information processing and it is mainly used for supervised learning [2]. By training the model with labelled data, neural networks can derive the reflection from the data features to labels and it can be used for classification problems. Adding more neural network layers can make the model identify more abstract and advanced labelled feature.

For example, the deep neural network model in figure 1, by inputting the data into the input layer and implementing the feature extraction in hidden layers, output layer gets the label for each data record. The relationship between layers are linear transformations and each neuron represents one linear transformation. Deep neural network can be used to solve many practical problems. For example, it can be used to do human facial recognition. Different human face images can be inputted into the model and it will tell you whether there is a human face inside the image [1,9].

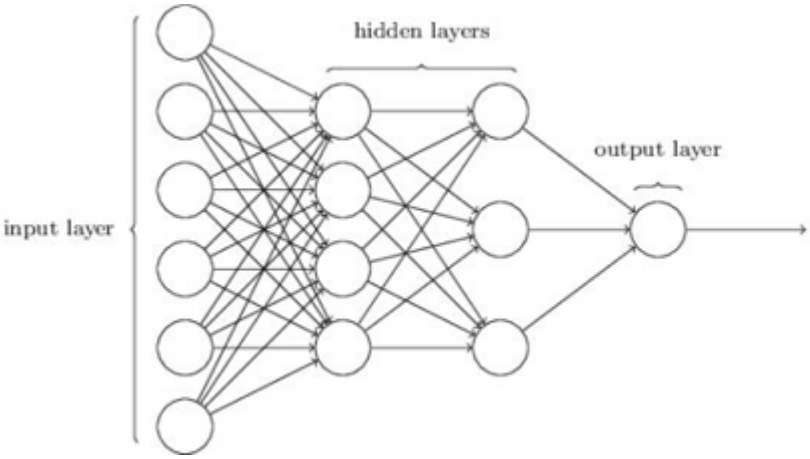


Figure 1 Deep Neural Network

Neural networks algorithms can be further divided into Convolutional Neural Network for image recognition, Recurrent Neural Network with long short term memory for Natural Language Processing and Generative Adversarial Network which is widely used for producing image, video, natural language and music.

2. Deep Neural Network’s Parameters Optimization

Except deciding on the network structure (number of layers and number of neurons in each layer), how to train the model with training data set (data with its label) to get a set of optimized parameters is also a significant part of implementing Deep Neural Network algorithm. Specifically, this training process can be defined as: given a dataset D = {X,Y}, X={X_1,...X_N} is the input dataset and Y={y1,…,yN} is the corresponding label dataset. Our optimization purpose is to find a hypothesis function that defines the relationship between X and Y. After we get a set of the optimized parameters, we can use it to classify the new data into different labels. To achieve this purpose, we usually define a cost function L(F(X)， Y) and train the model with different set of parameters to minimize the cost function. Squared loss and cross-entropy are cost functions that are used more frequently. The formula for squared loss is $\sum_{i=1}^{k} \quad (yi - \widehat{yi})$^2 and it's usually for regression model. Cross entropy's formula is $-\sum_{i=1}^{k} \quad (yi \; log \; \widehat{yi})$ and it's for classification model. Because deep neural networks contain lots of parameters, it's very hard to optimize the parameters directly. One solution is backpropagation algorithm [10]. A deep neural network can be regarded as a piled linear function with a non-linear function. For a network with L layers (h1, …, hL), each layer can be written as:

F_l(x) = W^(l+1)h^l

h^l = q(W^(l)h^(l-1))

q() is the non-linear function (invoke function). q() can be sigmoid, tanh, RelU algorithms etc [2,11]. To update W^L, the common method is gradient decent. For parameters on each layer, we use the rule below to update.

$$W_{t+1}^{(l)} \leftarrow W_t^{(l)} - \eta \nabla_{W_t^{(l)}} \mathcal{L}(\mathbf{F}(\,\mathbf{x}\,), y_t) \quad \forall l = 1, \ldots, L+1$$

η is the predefined learning rate. When we train the model, in most cases we will select a relatively big learning rate like 0.1. Then we gradually lower the learning rate to make the model more accurate. Because deep neural network has a huge number of parameters, even though we use different regularization methods (Convolutional Neural Network [16], or Dropout technology [17]) to reduce the number of parameters, the model still need to be trained with massive data and the gradient decent will slow the computing speed. One standard method that can mitigate this issue is stochastic gradient decent [1,2,3,14,15].

Stochastic gradient decent randomly selects a subset of the training data every time it updates parameters. In this way, it can have a good performance even with a limited number of iteration updates [14,15].

3. Test the performance of deep neural network algorithm

After we finish training the model, we need to test the model's performance. For testing model, we need to use a different set of data other than the training data set. To avoid the over-fitting [2,11,12], we also need to keep a set of unused data for cross-validation. As we can see from the figure 2, a model that has good performance in training set can have an instable performance in new data set. Only when the model have good performance in testing dataset, it can be used to solve real tasks.

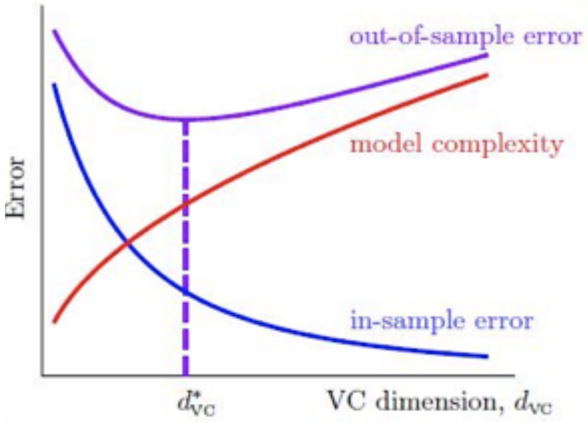


Figure 2. Model could have Over-fitting risk

4. Model Training and Proof of Work

Genesis' POC mechanism is based on deep neural network's optimization and testing logic. POC verifies miners' work through the performance of the model. In POC, every transaction includes a subset from the original dataset (divided into training dataset and testing dataset). When a miner receives the model trained by the previous block, he or she can get the gradient decent status of the model from the transaction and update the model. One important note is that because the data from the transaction is randomly selected from the original dataset, the gradient decent of the training process is as same as the stochastic gradient decent. Genesis uses SGX technology to carry the training model and the training dataset. SGX also significantly improves the efficiency of POC's work of security and proof of work.

SGX

1. SGX definition

SGX's full name is Intel Software Guard Extensions. SGX is an extension of Intel Architecture and can be used to improve software's security. Unlike other security methods which identify and isolate all male software on the platform, SGX encapsulates all legit software's operations into an enclave to protect them from malware's attack. Neither privileged nor non-privileged software have rights to access enclave. It means that once the software or data is in the enclave, even the operating system or VMM (Hypervisor) cannot influence the code and data inside enclave. Enclave's security border only contains CPU and itself. The enclave created by SGX can be regarded as a TEE (Trusted Execution Environment). SGX is slightly different from ARM TrustZone (TZ). TZ divides the CPU into two isolated operating environments (security environment and normal environment). These two environments communicate with each other using SMC commands. However, SGX operates multiple secure enclaves in one CPU. Also, TZ can achieve the same effect by dividing the CPU into multiple isolated environments.

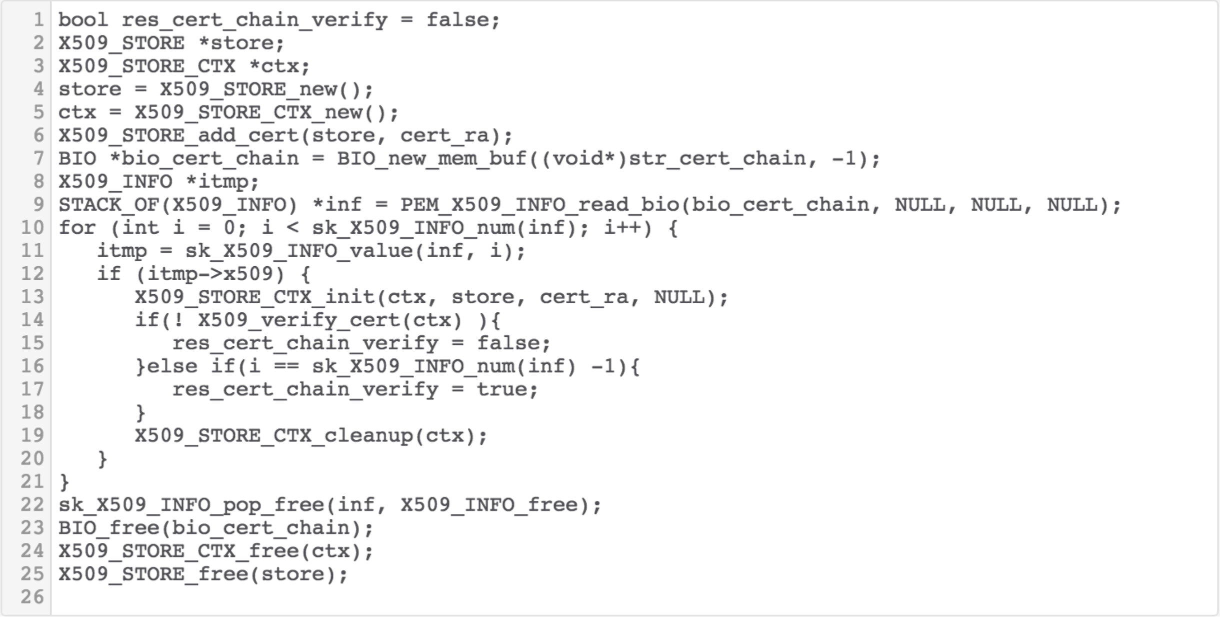
2. SGX's fundamentals

SGX protects the address space of applications. SGX uses the processor's commands to divide a part in CPU called EPC and reflect the enclaves of applications' address space to EPC. This part of CPU is encrypted and it uses the storage control unit in CPU to encrypt and transform the address.

How SGX protect the CPU: when the processor accesses the data in enclave, CPU automatically switches to enclave mode. In enclave mode, every access to CPU need to go through extra hardware examination. Because the data is stored in EPC, to prevent storage attack, the RAM in ECP are encrypted by MEE. Only when the RAM in EPC enters CPU package, it will be decrypted. And it will be encrypted again if it is put back to EPC.

3. SGX' s Remote Verification

Intel's SGX not only can offer hardware protection to offline applications, but to online applications, like DRM application. Banks' transaction system can provide "client side prove its legitimacy to server" capability. We call this capability Remote Attestation. In this process, the hardware and software's platform information from the client side and the fingerprint information relating to enclave are sent to the developer's server (Service Provider). Then the developer's server transforms the information to SGX's remote verification server (Attestation Service). SGX's remote verification server verifies whether the information is valid and sends back the result to the developer's server. After receiving the result, developer's server can know the credibility of the client side who sends the verification request and take the next step. The execution in code is shown below:



```
bool res_cert_chain_verify = false;
X509_STORE *store;
X509_STORE_CTX *ctx;
store = X509_STORE_new();
ctx = X509_STORE_CTX_new();
X509_STORE_add_cert(store, cert_ra);
BIO *bio_cert_chain = BIO_new_mem_buf((void*)str_cert_chain, -1);
X509_INFO *itmp;
STACK_OF(X509_INFO) *inf = PEM_X509_INFO_read_bio(bio_cert_chain, NULL, NULL, NULL);
for (int i = 0; i < sk_X509_INFO_num(inf); i++) {
   itmp = sk_X509_INFO_value(inf, i);
   if (itmp->x509) {
      X509_STORE_CTX_init(ctx, store, cert_ra, NULL);
      if(! X509_verify_cert(ctx) ){
         res_cert_chain_verify = false;
      }else if(i == sk_X509_INFO_num(inf) -1){
         res_cert_chain_verify = true;
      }
      X509_STORE_CTX_cleanup(ctx);
   }
}
sk_X509_INFO_pop_free(inf, X509_INFO_free);
BIO_free(bio_cert_chain);
X509_STORE_CTX_free(ctx);
X509_STORE_free(store);
```

4. SGX and POC

After obtaining the optimized model from SGX, miners need to pack the model parameters, quote generated from SGX and the testing score together in a new block. The newly generated block still cannot be broadcast to all normal users till it is verified by part of the miner cluster. When miners (verifiers) receive the testing request, they can decide which blocks they want to verify (based on the past credit score of the miner who sent the request) and test with the data from the transaction. If it is valid, verifiers will add points to the credit score of the miner who sent the request and put the verification score, weighted average score of the block:

$$\bar{x} = \frac{x_1 f_1 + x_2 f_2 + \cdots + x_k f_k}{n}$$

and the variance：

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - x)^2}{n}$$

to the block with their digital signature. If the verification fails, the credit score of the miner who sent the request will decline. When a block's number of successful verification surpasses a certain number, this block is regarded as valid and allowed to be broadcast to the whole network. Users judge based on blocks' average score and variance. Hence, blocks that have better performance (higher average, lower variance) are more likely to be synchronized,

Higher the POC, a block is more likely to be synchronized on different nodes. As a result, transactions with high-quality content are synchronized to the blockchain quickly and miners who participate in these transactions' verification are more likely to get rewards. Meanwhile, the model is processing huge amount of data without a label. This is equivalent to creating a new dataset with label. The whole process can be seen as a symbol of how efficiently Genesis uses computing power and it has a huge practical value to research and training of new models in different fields.

Reference

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[2] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436.

[3] Goodfellow I, Bengio Y, Courville A, et al. Deep learning[M]. Cambridge: MIT press, 2016.

[4] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 815-823.

[5] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017: 2980-2988.

[6] dos Santos C, Gatti M. Deep convolutional neural networks for sentiment analysis of short texts[C]//Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 2014: 69-78.

[7] Kouloumpis E, Wilson T, Moore J D. Twitter sentiment analysis: The good the bad and the omg![J]. Icwsm, 2011, 11(538-541): 164.

[8] Hush D R, Horne B G. Progress in supervised neural networks[J]. IEEE signal processing magazine, 1993, 10(1): 8-39.

[9] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[10] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533.

[11] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of the 27th international conference on machine learning (ICML-10). 2010: 807-814.

[12] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks[C]//Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. 2011: 315-323.

[13] Abadi M, Barham P, Chen J, et al. TensorFlow: A System for Large-Scale Machine Learning[C]//OSDI. 2016, 16: 265-283.

[14] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.

[15] Bottou L. Large-scale machine learning with stochastic gradient descent[M]//Proceedings of COMPSTAT'2010. Physica-Verlag HD, 2010: 177-186.

[16] LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series[J]. The handbook of brain theory and neural networks, 1995, 3361(10): 1995.

[17] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

Originality Announcement

POC consensus mechanism is Genesis team’s innovation. Genesis team has commissioned lawyers to apply for a global patent. Any unauthorized copy, imitation, modify, intercept and quotes are infringement. The team reserves the right to take further legal action.