AI Prediction& Ecosystem White Paper EN

AIPE

AIPE TOKEN:
TABLE OF CONTENTS

ABSTRACT

'Artificial Intelligence (AI)' has been widely known since Go match between AlphaGo and 9 dan ranked and 18-time world champion Lee Sedol and it has brought great changes to the world. It started to be used in various fields all over the world. It is actively used in the field of financial services, insurance (BFSI), life sciences and education. It is used in various industries such as government, defense, manufacturing, and energy. It is also widely used in retails, communications, media, entertainment, and sports.

In fact, as the demand for intelligent business processes increases, the overall experience delivered to users is improving. In response, global companies such as Google, Microsoft, IBM, AWS, Intel and FICO (Fair Isaac Corporation, Salesforce, Baidu and SAS) are focusing on integrating AI functionality with their business applications.

According to a report released by global market research institute Research and Markets, the AI service market is expected to grow at an annual average growth rate of over 48% until 2023 (reported in July 2018). Such a growth is based on an increased demand for intelligent business processes and hope to reduce operating costs caused by large amounts of data from increasing connected device and IoT. In addition, as governments, businesses and municipalities try to improve their infrastructures and smart city programs, the overall AI service market has been growing and the development is expected to continue.

It should be noted that AI technology is a high-value-added industry that can sustain growth and is a market where real profits are generated. However, contrary to external growth and potential of AI technology, the problem it is facing with the market is still not solved properly and it is in infancy.

Therefore, AIPE decided to build an AI model based on AI Prediction. It can link AI researchers who want to test data and put it into practical use, investors who want to use AI Model Data as an indicator for cryptocurrency investment and companies that need AI developer pool and AI data. We are developing platform based on blockchain that connects and forms an interactive and interdependent relationship between researcher, investor and company and fill their needs.

All information produced and shared within the platform is securely managed through blockchain technology. AI researchers have ownership and management rights over the developed content, investors' personal information and investment information from the AI Model is protected. When they share it with company participants, they will be compensated for it.

In addition, all participants and companies can implement various On Demand Services within this platform using API provided by AI Prediction Platform. Our AIPE team aims to redistribute the Prediction Model owned by AI researchers and the data ownership and management authority of actual users through the AI Prediction Platform to those who need them so that the values in demand can be recycled by giving an appropriate compensation. Based on this, we aim to contributed to AI industry revolution.

SERVICES

3PIKS

A unique reward platform that combines AI and blockchain technology

We have developed 3PIKS that connects AI and the cryptocurrency market to provide a price prediction service by combining AI and human factor.

Participants in cryptocurrency market can refer to not only the price prediction of AI models in 3PIKS services but also the results of 3PIKS users' predictions as a form of collective intelligence. Users can predict price of the cryptocurrency along with their own AI models and receive AI Token as a reward for it. AI Token will be exchanged for AIPE Token and will be exchanged for tokens and coins of our partners that have a strategic partnership with us in the near future. In the ecosystem that we aim to construct, a variety of uses will be guaranteed in such a way.

AIBitBip

AI-based prediction signal service for cryptocurrency price change

AI models capture meaningful UP / DOWN signs to notify users in the form of signals and help their trading. Prototypes have been developed in the form of applications to improve accessibility for users and will be launched in Q2 2019. In the beginning, it will serve several dozens of models that have been developed by ourselves. We will select and add more models of researchers developed in AI Prediction Platform.

The value of AIBitBip is not just about informing users about trading. If a number of people trade using the same signal, they can turn into one big group. If a large amount of capital is injected at the same time to create a flow of price, it is expected that users can use this as a reference for trading and the risk of investment will be greatly reduced in the future. The form of an application makes high exposure for users and it is also easily accessible. This application can turn into a place to post advertisement and use referral system.

AI Auto-Trading

Auto-Trading Bot based on AI Prediction Signal

Receiving a signal through AIBitBip does not necessarily mean that anyone can make profits from trading. It will be difficult to buy or sell in real-time based on signal in the cryptocurrency market where 24-hour large-scale trading are made constantly. In order to solve this problem, we are going to service AI Auto-Trading Bot using exchanges' API. We have been trying trading through AI Prediction Signal for a long time to lead the project to success. We have experienced a lot of trials and errors, but as AI has learned repeatedly and constantly, we could observe a gradual increase in hit rate and yield.

We aim to give profits to our users by providing a service that they can use to buy and sell at the right time. In order to use the service, it is necessary to own AIPE coin over a certain period of time and share a part of the profits of auto-trading with researchers to maintain the ecosystem.

Auto trading is not just about making profits, but it should also be in a close partnership with Exchange. In order to proceed with the auto trading, numerous processes of registering and making a trade at asking prices should be repeated. Exchange plays an important role in shaping the market by securing a large volume and liquidity of cryptocurrency. Through our B2B business, we can build a mutually beneficial ecosystem for user, exchange, and development of the Prediction Platform.

AI Prediction Platform

AI Research - AI Prediction Platform to improve the development environment and to obtain proven data

At present, the biggest barrier to entry in AI research is refining a large amount of data and having a computing power. It is almost impossible for an individual to set up such an environment, and it is the same for companies in need of AI model data.

AIPE introduces the AI Prediction Platform to improve the AI research and development environment. In the platform, AI researchers can easily access and use the Open Source Library environment, and companies can obtain traffic and model data. If many AI researchers make various AI models by using the data tools from the platform provided in such an environment, the market entry barriers of cloud providing companies will be lowered.

It will make it easier to manage AI models and share data systematically. This can be a starting point for new asset management trends by facilitating systematic AI model management, sharing data, reducing server usage costs, increasing the number of trading bots / algorithms, and increasing users' accessibility.

The platform infra provided by AIPE is meaningful not only in terms of AI researchers, investors, and corporates' convenience but also in terms of innovation in the value chain of the AI industry.

Exchange Solution

Many companies are challenging the development and service of cryptocurrency exchanges, but they are still struggling with problems of time & financial costs as well as lack of technical expertise & marketing.

AIPE aims to provide our own developed exchange's solution for such a problem. In addition to providing solutions for the existing centralized exchanges, we are also developing a cryptocurrency margin exchange that provides margin and leverage trading. The margin exchange under development will provide marginal futures trading similar to 'BitMEX' and will also provide marginal option trading.

There are many cryptocurrency exchange solution providers, but the companies are limited to providing to centralized exchanges only. AIPE aims to provide a cryptocurrency investment solution that supports margin trading, futures trading, marginal futures trading and marginal option trading.

AI Prediction Ecosystem

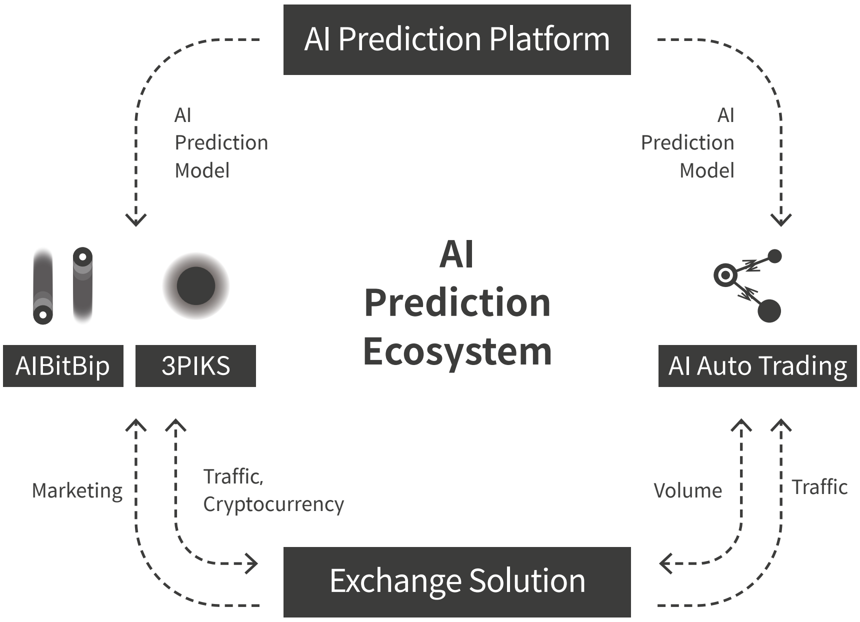
Vision of ecosystem on coexistence of AIPE's AI services

The vision that AIPE pursues is to form a unified cryptocurrency ecosystem with the unique value of each service.

We airdrop coins and tokens to users by using both 3PIKS, a reward platform, and AIBitBip, a cryptocurrency price change notification service and they can be used as a marketing channel for an exchange or a specific coin.

Also, we compensate by AI Token (tentative name) in 3PIKS, and it is in the process of being able to swap to token and coin of companies that we have a partnership with. Swapped coins can be withdrawn from exchanges that provide their marketing services, which can ensure the exchange's traffic and liquidity.

Ecosystem of AIPE does not merely include excellent AI models or provide a solution for developing exchanges. Each service has a mutual & unified relationship and value. It has a structure that circulates prediction of cryptocurrency price, currency liquidity, publicity, traffic, and market stability



AI TECHNOLOGY

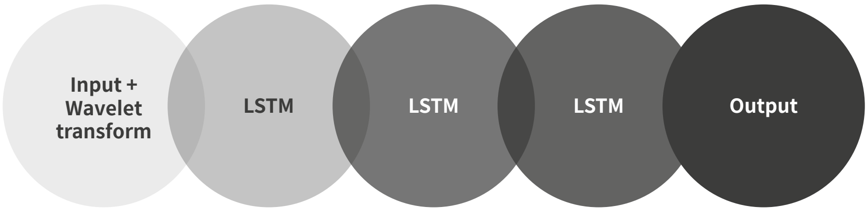
Outline

Since AI development is our main project, we are operating AI Research Lab. When developing the Prediction Model for the first time, we started from using a model of deep-learning method only, but now we use both machine-learning and deep-learning to develop models.

If we explain the deep-learning method roughly, it is the construction of modeling system using RNN and CNN methodology. It is very important to preprocess the relevant time series data for modeling. The preprocessing of the input data of the RNN is focused on effectively removing noise. In addition, data preprocessing applied to CNN shows an emphasis on efficiently representing patterns. Although we use the RNN and CNN methods independently, we tried modeling them complementarily. In order to compensate for the instability of the Decision Tree, the Gradient Boosting, which is one of the commonly used methods, constitutes a prediction mode as machine-learning methodology.

Since July 2017, time series data of about 100,000 cryptocurrencies have been used for training machine-learning models. Time series data was based on five features: high price, low price, market price, close price, and trading volume. As the recent research is accelerated, we use various features from feature engineering that uses technical indicators in addition to the five basic features mentioned above.

The machine-learning model tends to achieve higher accuracy when the input data is large. However, since characteristics of each model are different, and the time series data has some distinctive elements. Training models by inputting limitless amount of data is not always the best way. In fact, models showed high performances when they were trained with selective data. For example, the classification accuracy of the CNN model was the highest when the CNN model was trained with the recent 18,000 time series data, and some LSTM models showed the highest performance when the whole data was trained.



[Figure] Structure of stacked wavelet LSTM model

Next, the structure of the model we use is as follows. When using RNN methodology, stacked LSTM / GRU with multiple LSTM or GRU layers showed a good performance. Also, the RNN model with wavelet transform showed satisfactory results. The CNN model is important for enhancing the performance of the combination of the number of convolutional layers and the activation function. The more layers we had, the more over-sum problems we had, and we used the dropout and batch normalization appropriately. Mixed models with LSTM cells and convolutional layers stacked in parallel showed a good performance as well. In the case of gradient boosting, we used methodologies such as XGBoost, LightGBM and Catboost. In the case of boosting series of machine-learning models, a performance is dependent on feature selection and data size.

Introducing Main Technology

Outline

Our research team, composed of about 10 members, is working on research and methodology is as follows. First of all, we study major research papers in English, Chinese and Korean, and practice them by ourselves. We have used about 500 papers from August 2018 to December 2018. We are currently continuing to train the models that have achieved the standard in our actual implementation model and improve the usage method. Significant reference research papers can be found on the R & D blog of cosmicbc.com. We have been releasing our research results steadily and will continue to open to the public. As of December 2018, we have implemented about 100 papers in actual service.

We have researched and implemented a model by using AI cloud computing server of Amazon web service (AWS), Naver Cloud Platform (NCP), and Alibaba Cloud. We have implemented models and taught on historical data and live data for 2 to 3 weeks and check out their accuracy percentage.

Based on this, a variety of transformation processing methods are implemented focusing on models with a probability higher than a certain value, so that they can learn for more than one month.

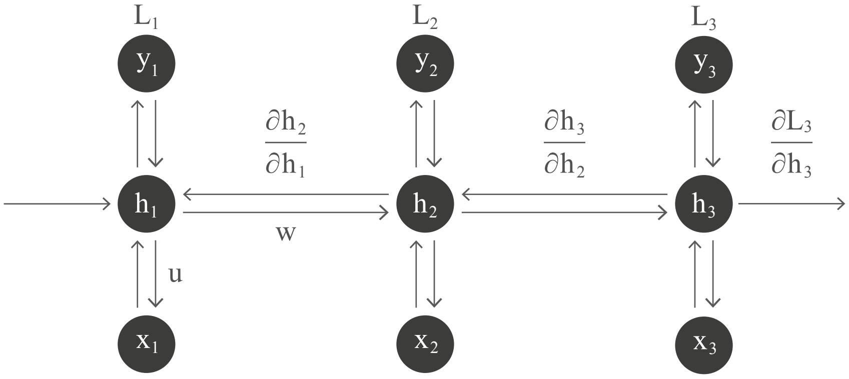
RNN

The characteristics of the Recurrent Neural Network (RNN) or the circular neural network model are fully connected neural networks. It has a memory that is not found in the CNN (Convolutional Neural Network). This is known to be suitable for solving the problem of prediction classification of sequence data. Sequence data is time series data given as an ordered sequence in each element, for example, voice, video, and text. This can be expressed in the following form.

$$x^1, x^2, x^3, \dots\dots, x^T$$

Here, the lengths of consecutive time series t = 1,2,3, ..., T are generally variable. An example of an estimation problem that deals with real sequence data is to estimate the t + 1th word given to the tth word of a sentence. Each word is strongly influenced by the sequence of previous words. It is estimated that RNN achieves highly accurate predictions by learning the dependency between words, that is, context. In the case of time series data such as cryptocurrency, it is common that previous data influences the current data. The RNN may temporarily store information and vary the response accordingly due to the memory structure described above. The 'context' of continuous time series data can be captured, and the classification problem can be handled well. That is, the RNN can be applied to solve the problem of fluctuations of cryptocurrency prices.

When training RNN, we use Gradient Descent like fully connected neural network. Two methods can be applied to calculate the derivative of the error with respect to the weight of each layers. One is Realtime Recurrent Learning (RTRL), and the other is Back Propagation Through Time (BPTT). The former is more efficient in memory usage, while the latter is faster in computation.


L_1
L_2
L_3
y_1
y_2
y_3
$\frac{\partial h_2}{\partial h_1}$
$\frac{\partial h_3}{\partial h_2}$
$\frac{\partial L_3}{\partial h_3}$
h_1
h_2
h_3
w
u
x_1
x_2
x_3

[Figure] BTPP
※Source: Hands-On Reinforcement Learning with Python by Sudharsan Ravichandiran

BPTT develops the RNN in the time direction and changes it to the same structure as the fully connected neural network, and then performs backpropagation calculation. As shown in the figure above, the RNN of each time is unfolded to express the return path of the intermediate layer as a combination to the middle layer unit at consecutive times. The developed neural network becomes the existing fully connected neural network, and the gradient can be calculated through the backpropagation. We can define a delta that propagates to the previous layer like an existing backpropagation.

$$\delta_2^{(3)} = \frac{\partial L_3}{\partial z_3} = \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial s_2}\frac{\partial s_2}{\partial z_2}$$

Here $z_2 = Ux_2 + Ws_1$

If we apply the same method at the previous time step, it is

$$\delta_1^{(3)} = \frac{\partial L_3}{\partial z_1} = \frac{\partial L_3}{\partial h_2}\frac{\partial h_2}{\partial s_1}\frac{\partial s_1}{\partial z_1} = \delta_2^{(3)}\frac{\partial h_2}{\partial s_1}\frac{\partial s_1}{\partial z_1}$$

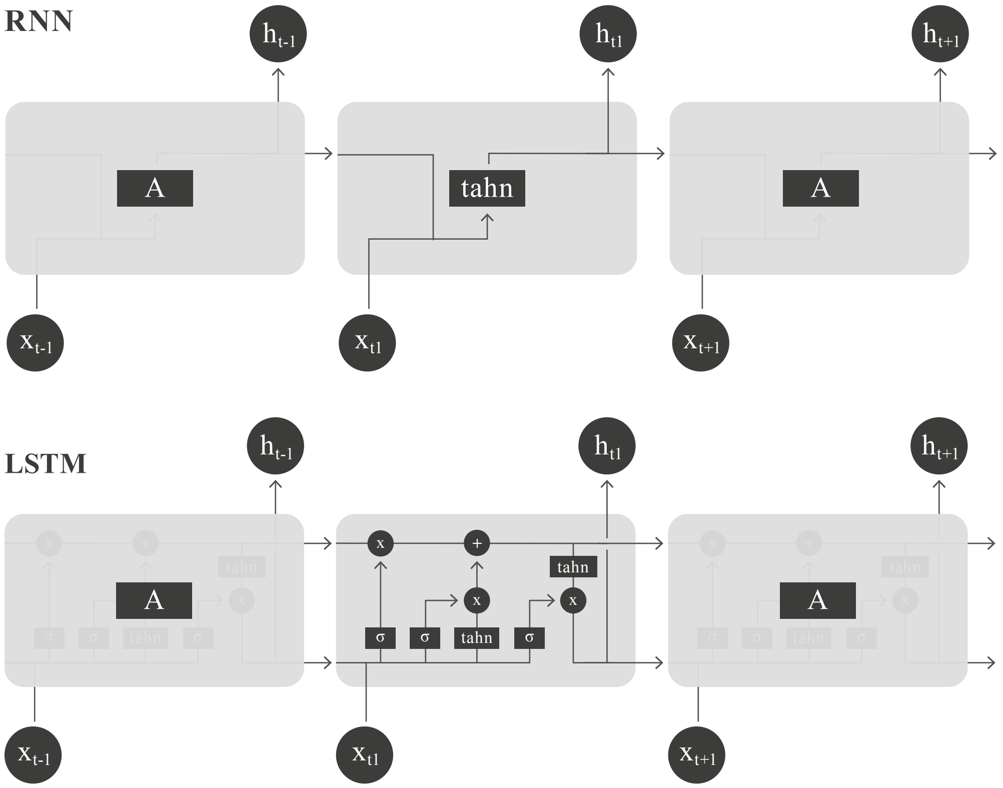
As described earlier, the RNN can capture and estimate context of sequence data, but it is sensitive to the length of the context.

Theoretically, it is possible to keep information of all previous time steps from time t, but dependence over a long time cannot be learned. In reality, the maximum amount of time that can be reflected in the output of the RNN is determined to be about 10 time steps in the past.

This limit is given to the neural network with a large number of layers when the gradient is calculated by the backpropagation method. The gradient value diverges or disappears as the layer goes back up. In other words, even if the number of RNN layers is small, it is equivalent to handling many layers in the backpropagation calculation, and the gradient value is also easily diverted or lost. This is the reason why it is difficult to handle long sequences in an RNN. Short-term memory can be realized, but it is difficult to realize long-term memory.

LSTM and GRU

In the early 1990s, Hochreiter, Schmidhuber, and Bengio introduced several ways to solve this problem through a theoretical review of the issues that occurred in the past. The most successful of these is known as Long Short Term Memory (LSTM). The LSTM is a structure in which each unit of the middle layer is composed of elements called memory units as compared with the basic RNN described above. The rest of the structure is exactly same as the existing RNN.



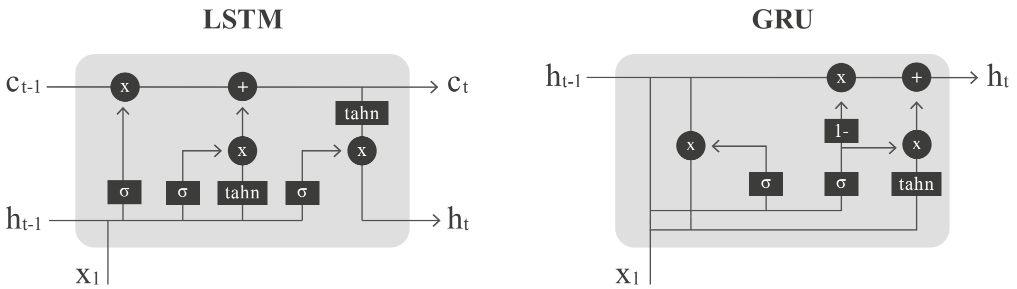
[Figure] Comparison of RNN and LSTM Structure
※Source: https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/

It is easy to understand when imagining a cell-state that serves as a conveyor belt. A way to move information across multiple time steps has been added. The information extracted at some point in the time series moves up to the conveying belt and moves down to the time step at the required point. LSTM does this by storing information for later and preventing the gradual loss of old signals in the process.

The formula for the LSTM cell is as follows. Here, ⊙ is the Hadamard product operator, which means multiplication by element.

$$f_t = \sigma(W_{xh\_f}x_t + W_{hh\_f}h_{t-1} + b_{h\_f})$$
$$i_t = \sigma(W_{xh\_i}x_t + W_{hh\_i}h_{t-1} + b_{h\_i})$$
$$o_t = \sigma(W_{xh\_o}x_t + W_{hh\_o}h_{t-1} + b_{h\_o})$$
$$g_t = \tanh(W_{xh\_g}x_t + W_{hh\_g}h_{t-1} + b_{h\_g})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot \tanh(c_t)$$

Forget gate f_t is a gate to forget past information. The value sent from forget gate is the value that the sigmoid applied to xt and h_{t-1}. If this value is 0, it forgets the information of the previous state, and if it is 1, it remembers completely. The “input gate $i_t \odot g_t$” is for storing the current information. The input gate computes the value of Hadamard product operation after applying sigmoid with x_t and h_{t-1} and applying hyperbolic tangent (tanh) as the same input. In short, LSTM solves the long term memory problems of RNN and has been used successfully in many areas. Since the success of LSTM, many variants of LSTM have been studied and published in the academic community. Among them, Gated Recurrent Unit (GRU), introduced by Cho Kyung-hyun in 2014 is remarkable.



[Figure] GRU
※Source: http://sqlml.azurewebsites.net/2017/08/12/recurrent-neural-network/

As shown in the figure above, we can see the structure of GRU is simpler than that of LSTM. In detail, like LSRM, the amount of information is controlled by using the gate, but how to control the gate is slightly different.

When we solve this by formula, it is

$$z_t = \sigma(x_t U^z + h_{t-1} W^z)$$

$$r_t = \sigma(x_t U^r + h_{t-1} W^r)$$

$$\tilde{h}_t = \tanh(x_t U^h + (r_t * h_{t-1}) W^h)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

The reset gate r_t has a value between 0 and 1. If the value is close to 0, it forgets all past information, and if it is 1, it stores all past information. The last line of the formula controls the update to the next state. $h_{t-1}$is the past information. $\tilde{h}_t$is the current information. And it is the update gate, z_t, which decides how to assemble it.

CNN

CNN is a neural network modeling technique mainly applied to image recognition. It has been surprisingly solving what has been regarded as a problem of computer vision (recognition of object categories, and, etc.). The problem of existing fully connected neural networks is that the number of weight and bias needed to recognize the small sized handwritten explosively is increased. For example, imagine there is a fully connected neural network with 100 neurons in a single hidden layer to recognize a 16 x 16 handwriting. A total of 28,326 weights and biases are required for this network. When the hidden layer is further stacked in this neural network, the number of learning parameters becomes much larger. In addition, when a cursive script image is vertically / horizontally moved, rotated, or deformed, there will a problem that new learning data must be inserted.

To solve this problem, unlike fully connected neural networks, CNNs have a special layer where only certain units are bonded between adjacent layers. In this particular layer, it is operated related to image processing such as convolution and pooling.

The development of CNN got a hint from neuroscience knowledge about the visual cortex in the animal's brain. More specifically, it was inspired by the locality of the receptive field of the nerve cell to the visual cortex and the presence of simple cells and complex cells. Some neurons are only excited by the horizon image, and other neurons respond to images from other angles. We also found that some neurons have large receptive fields, so they respond to more complex patterns combined with low-level patterns. By using this, representative CNN models such as LeNet, AlexNet, VGG, and GoogLeNet have been made.

The basic structure of CNN can be summarized as follows. The neurons in the first convolutional layer that receive the input image are not connected to all the pixels in the input image but are connected only to the pixels in the receptive field of the convolutional layered neuron. However, the product of the multiplications performs a linear operation. Therefore, nonlinear activation functions such as ReLU (Rectified Linear Unit) are connected to the convolutional layer to learn more complex nonlinear relations. The pooling layer then creates a subsample of the input image to reduce computation, memory usage, and number of parameters. Depending on the characteristics of the data, it is possible to stack various layers with a combination of the convolutional layer + activation function + pooling layer. Normally, CNN attaches more than one layer of full connectivity before output, which allows to map the last layer of a high-dimensional feature to an image classification.

As mentioned above, the most important component of CNN is a convolutional layer. Let's say that there is a black and white image of WXW size, and the index of each pixel is (i, j). There is a HXH filter, and the pixel index of the filter is (p, q) and p, q = 0, ... , H-1. Here, H<W. Strictly speaking, the convolutional layer can be defined as follows.

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i-p,\ j-p} h_{qp}$$

The computation of the pooling layer following the convolutional layer can be briefly described as follows. Let P_{ij} be the set of pixels contained in the HXH-sized square area around the pixel (i, j) of the input image of size WXWXK. By applying appropriate padding, you can build a P_{ij} that contains the edges of the image as well. For each pixel in P_{ij}, a pixel value u_{ijk} is obtained by using a pixel value and having H^2 in each channel k independently. Typically, the method of obtaining the pixel values is max pooling and average pooling.

Max pooling is a method of selecting the maximum value of H^2 pixel values and it is calculated as follows.

$$u_{ijk} = \max_{(p,q)\in P_{ij}} z_{pqk}$$

The average pooling takes an average of H^2 pixel values and takes it as pixel values.

$$u_{ijk} = \frac{1}{H^2} \sum_{(p,q)\in P_{ij}} z_{pqk}$$

Finally, the Lp pooling that includes max and average pooling is as follows. The arithmetic operation of pooling is executed in parallel for each channel of the input image. That is, the number of output channels of the pooling layer is equal to the number of channels of the input image. Lp pooling can be expressed as

$$u_{ijk} = \left(\frac{1}{H^2} \sum_{(p,q)\in P_{ij}} z_{pqk}^{p} \right)^{\frac{1}{p}}$$

And it can be easily seen that if P = 1, it is an average pooling and if P = ∞, it is a max pooling. In the price prediction classified problem, the movement of time series data can be patterned and analyzed. The model constructed by using CNN turned to have a strong predictive power.

Gradient Boosting

Gradient Boosting is a compound of Gradient Descent and Boosting described above. Boosting is a method of growing a relatively simple Decision Tree (or a learner), learning it in a sequential way, and growing another Decision Tree based on the result. Gradient Boosting can use any function that can be differentiated as a loss function, as we can analogize from the name Gradient Descent Method.

The algorithm of the Gradient Bosting can be expressed as:

$$h_0(x) = \underset{\gamma}{\arg\min} \sum_{i=1}^{n} L(y_i, \gamma)$$

First, the learner (h) is initialized.m = 1,2 ...M. Repeat the following 1, 2, 3 and 4

$$r_{im} = -\left[\frac{\partial L(y_i, h(x_i))}{\partial h(x_i)}\right]_{h(x)=h_m-1(x),} \quad i = 1,2, \dots ,m$$

1. Calculate the pseudo-residual.

$$\{(x_i, r_{im}) \mid i = 1,2, \dots ,n\}$$

2. The base learner h_m (x)fits the pseudo-residual.
The training data is like above.

$$\gamma_m = \underset{\gamma}{\arg\min} \sum_{i=1}^{n} L(y_i, h_{m-1}\gamma(x_i)+\gamma h_m(x_i))$$

3. The following multiplier $-_m$ is calculated.

$$h_m(x) = h_{m-1}\gamma(x)+\gamma_m h_m(x)$$

4. Update the learner.

$$h_M(x) = \sum_{i=1}^{M} \gamma_m h(x)+h_0(x)$$

The final model is selected as follows.

In the case of the price movement classified problem, if exponential loss function is used, it is $r_{im}=y_i exp(-y_i h_{m-1}(x_i))=y_i w_{im}$. If w_im is classified, it is e^{-1}. If w_im is misclassified, it is e^{-1}. A larger weight is given to the misclassified data and the process, and it proceeds to the next step. Since the pseudo residual is a kind of residual, the generalization of boosting is the process of minimizing the loss function according to the steepest gradient descent method when the loss function is defined.

Models based on gradient boosting are typically XGBoost, LightGBM, and Catboost.

First, XGBoost can: use parallel processing for fast learning and prediction; provide a variety of custom optimization options; prevent overfitting by automatic pruning using greedy algorithm; and have a good connectivity with another algorithm.

Next, the biggest advantage of LightGBM is that the learning time is shorter than XGBoost. Memory usage is also relatively low. It was developed two years later than XGBoost, and it is said that the advantages are inherited and the disadvantages are complemented. In the partitioning method focusing on leaf of LightGBM, the tree is not balanced, and the leaf node having the maximum loss value is continuously partitioned, resulting in a longer tree and an asymmetrical ruled tree. However, this can minimize the prediction error loss rather than the balanced tree partitioning method.

Finally, Catboost solved the pre-processing problem of prediction shift and categorical variables due to the existing target or data leakage by substituting the concept of ordering principle. When we describe the former in more details, we used the ordered boosting technique in calculating leaf values. The existing gradient boosting is reusing the data used in the current model when creating a new tree, which is vulnerable to excessive summing. Unlike the conventional method, the leaf value is obtained first in a reverse order, and the tree structure is selected.

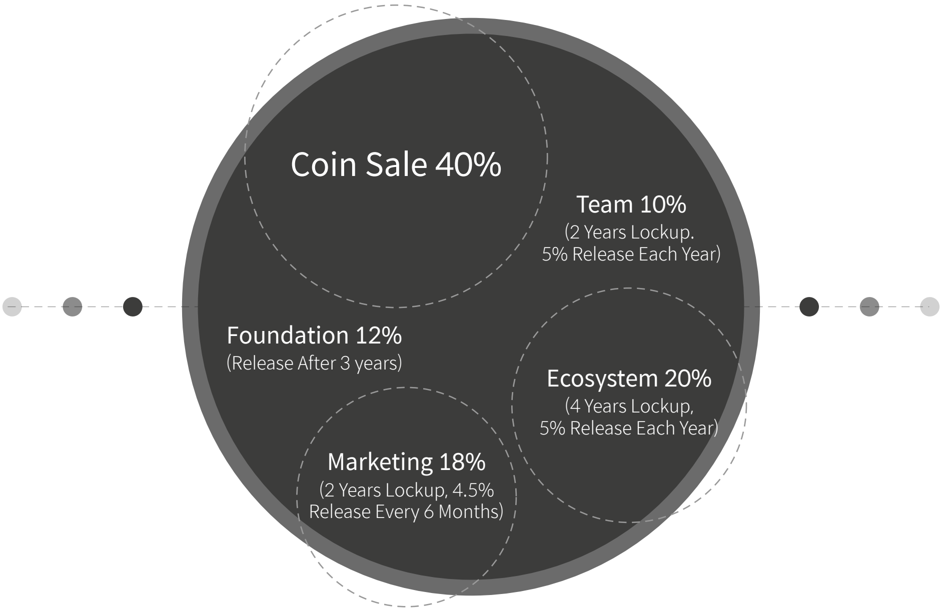
TOKEN ECONOMY

Coin Structure

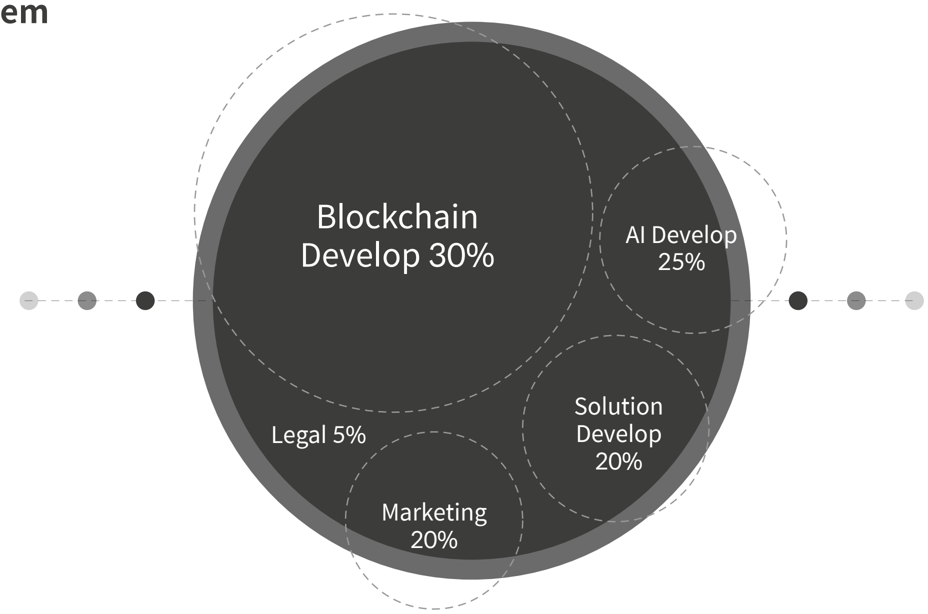
Total Volume Issued	12 billion
Price per Unit	1 AIPE = about 0.02 USD
Coin Name	AIPE Token
Coin Symbol	AIPE
Standard	ERC-20
Decimal	18
Hard Cap	412,500,000 AIPE
Soft Cap	137,500,000 AIPE

Allocation

Issuing AIPE is to create an ecosystem based on platform development and AI prediction. ICO and IEO participants will receive a distributed amount equal to 40% of the total volume issued. 30% of the remainder, excluding coin sale, is used for AI Prediction Platform construction and marketing required for the formation of the ecosystem in the beginning. The AIPE Foundation equivalent of 12% is used to build up the infrastructure as the platform is further developed and the platform is expanded. 12% of foundation volume and 10% of distribution volume will be distributed sequentially.

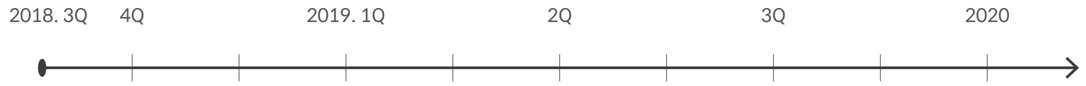


Ecosystem



SERVICE	INFO
Blockchain Develop	Maintaining AI Develop Platform and actively utilizing it to guarantee the authorities of developed models by studying blockchain. Supporting research to store AI Prediction Database in Block as a means of ensuring public confidence.
AI Develop	Supporting environment formation and prediction model development for AI researchers' education and research. Researching big data analysis' algorithm based on machine learning.
Solution Develop	Product development for integrated solution. Developing 3PIKS, AIBitBip, AI Auto-Trading Bot, Exchange, and Margin Exchange.
Marketing	Considering blockchain industry is a highly competitive region, actively utilizing various marketing channels to attract users and works for settlement of the platform. Aiming to establish a stabilization of our platform.
Legal	Handling of copyrights and related laws arising from service development and operation.

ROADMAP



Product	**AI Prediction Challenge**	**3PIKS**	**AIBitBip, AI Auto-Trading**	**AI Prediction Exchange**	**AI Prediction Platform**
Coin	-	Whitepaper	Coin Sale	Listing	Mainnet

PARTNERS

AIPE Partners

NBP NAVER BUSINESS PLATFORM

UZBIT

CoinBene

BLOCK INSIDE

UZBIT Partners

KOBEA

CARDANO

EMURGO

infinito wallet

AlphaPoint

NOTICE & CAUTION

The AIPE project white paper is designed to provide a detailed description of the project's overall content and roadmap. This white paper is not intended to be an investment solicitation, and AIPE will not be responsible for any harms, losses, debts or other financial issues incurred to anyone who reads this white paper. Please note that AIPE has no responsibility for any compensation.

If a reader of this white paper uses it for his or her decision making activities (including, using it as reference), please note again that AIPE will not be responsible for harms, compensations or any other liability.

As the white paper of AIPE project is written "as is," we do not guarantee that any content contained in the white paper will be accurate or unchangeable in the future.

As it does not guarantee any contents or statements of this white paper, AIPE has no legal responsibility for those who read it. For example, AIPE does not guarantee that a white paper is based on legitimate rights, not infringing on the rights of a third party, commercially valuable or useful, appropriate for readers who have a certain goal, and contains no error in the content of the white paper, and so on. The scope of liability waiver is not limited to the example mentioned.