

# Automatic Derivation Of Formulas Using Reinforcement Learning\*

MinZhong Luo

China Institute of Atomic Energy  
Peking  
luomincong@foxmail.com

Li Liu

China Institute of Atomic Energy  
Peking  
liuli@foxmail.com

## ABSTRACT

This paper presents an artificial intelligence algorithm that can be used to derive formulas from various scientific disciplines called **automatic derivation machine**. First, the formula is abstractly expressed as a multiway tree model, and then each step of the formula derivation transformation is abstracted as a mapping of multiway trees. Derivation steps similar can be expressed as a reusable formula template by a multiway tree map. After that, the formula multiway tree is eigen-encoded to feature vectors construct the feature space of formulas, the Q-learning model using in this feature space can achieve the derivation by making training data from derivation process. Finally, an automatic formula derivation machine is made to choose the next derivation step based on the current state and object. We also make an example about the nuclear reactor physics problem to show how the automatic derivation machine works.

## KEYWORDS

Automatic derivation ,Formula multiway tree,Encoding algorithm,Q-learning

### ACM Reference Format:

MinZhong Luo and Li Liu. 2018. Automatic Derivation Of Formulas Using Reinforcement Learning. In *Proceedings of ACM ICCES (Conference ICCES)*. ACM, Kuala Lumpur,Malaysia, Article 4, 8 pages. [http://www.ijist.net/ICCES2018/10.475/123\\_4](http://www.ijist.net/ICCES2018/10.475/123_4)

## 1 INTRODUCTION

The automatic derivation of formulas is a revolutionary application of AI in a particular field of scientific research. By abstracting the structure of a formula into a deducible data structure, the goal can be deduced. Previously there has been no study of formula-derived AI methods, but "automatic machine proof" [7] is a study that is similar to formula derivation. Automatic machine proof refers to the method which transforms proof-type problems into computer-computable

forms, such as polynomials [2], and then calculating it. Automatic proof is a special form of automatic formula derivation. Due to the requirement of simple calculation, machine proof generally involves only simple geometric problems [8].

The automatic formula derivation in this paper refers to the use of the AI method in various professional fields (such as materials science, computational mechanics, etc.) to make complex calculations of relevant formulas automatically to obtain valuable calculation results and solutions, which is revolutionary for traditional scientific research.

First, we re-express the formula as a multiway tree model, which is called a formula multiway tree. In the formula multiway tree, each non-leaf node is an operation symbol (for example, =, +,  $\exp(\cdot)$  Etc), while leaf nodes are algebraic and numeric symbol (eg  $x$ ,  $e^{-2.337}$ ,  $\pi$ , etc.). By devising the subtree matching algorithm,tree construction algorithm,subtree replacement algorithm of the formula, the formula can be deduced or transformed. The subtree matching algorithm and replacement algorithm of the formula multiway tree can make the formula multiway tree deform according to a certain template, which makes the complex formula can be deformed according to the simple formula multiway tree deformation method, and this deformation mode can be used for more complex formulae, which is called iterative learning for automatic formula derivation.

We have designed the encoding algorithm for the formula multiway tree, which can transform different formulas into feature vectors in a unified dimension and establish the feature space of the formulas. The eigencoding of the formula multiway tree makes:

- each formula has a unique identity;
- we can measure the similarity of the formulas. The similar formula multiway trees have a smaller spatial distance;
- the feature vectors can be used to train the learner.

We use the reinforcement learning mechanism to train the formula derivation machine. For a specific professional problem (such as calculation in theoretical mechanics), we need to prepare the relevant derivation training set, set the optional formula multiway tree transformation method to the action set  $a_i$ , and set expressions for formulas at different stages to state set  $s_i$ , and use the gradient descent method to optimize the model parameters by establishing the neural network model  $\pi_\theta(a|s)$ .

There are two main difficulties in the automatic derivation of formulas.

\*The Source Code:<https://github.com/Luomin1993/SciBot>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
*Conference ICCES, July 2018, Kuala Lumpur, Malaysia*  
© 2018 Copyright held by the owner/author(s).  
ACM ISBN 123-4567-24-567/08/06.  
[http://www.ijist.net/ICCES2018/10.475/123\\_4](http://www.ijist.net/ICCES2018/10.475/123_4)

(1) The formula itself is a highly abstract mathematical language and cannot be directly calculated like other training data for machine learning.

(2) There are many basic formulas in each professional field. Man-made annotations have high professional requirements for people. For the first difficulty, the formula multiway tree can decompose the formula to the calculation-symbol level, can contain all the information of a formula, and its encoding can also be used as a training data to input learner. For the second difficulty, we can use the template mapping mechanism and iterative learning mode to artificially label as few basic formulas as possible, so that the automatic derivation machine automatically derives higher-order complex formulas based on low-order simple formula templates.

Artificial intelligence methods should not only focus on tasks such as recommendation or automatic identification in the service industry, but should also be used in more meaningful scientific research areas. By establishing the formula automatic derivation machine, we can efficiently obtain meaningful results. In the case of this paper, the neural network model  $\pi_\theta(a|s)$  is trained by constructing a first-order linear differential equation training set, which successfully derives the fission concentration equation solution of  $Pm^{149}$  in reactor physics. In accordance with the formula automatic derivation machine design principle of this paper, more complex professional scientific research problems can be solved.

## 2 FORMULA MULTIWAY TREE

### 2.1 multiway tree model

In the derivation of mathematical formulas, in order for the formula to be computed by a computer, the formula must be re-expressed in a computable form. In the research of machine proof, the proof of many geometric problems is re-expressed as the related polynomial [1], and the homogeneous differential equation can also be mapped as a polynomial to calculate, for example:

$$ay''' + by'' + cy' + d = y \Rightarrow ay^3 + by^2 + cy^1 + d = y^0$$

However, the traditional polynomial expression has great limitations. For example, if you want to use a polynomial to express a complicated formula, it is not feasible. For example, such as this formula:

$$\int y dx + \sin(y/x) + S(x) = 0$$

Therefore, the formula multiway tree is proposed to re-express the formula. The formula multiway tree is constructed from top to bottom according to the symbol priority. The computational symbol is a node, and the algebraic symbols and numerical symbols are leaf nodes, so that any complex formulas can be decomposed and re-expressed in the form of a multiway tree.

The formula is re-expressed with a multiway tree, and the formula derivation process can be based on the operation and

transformation algorithm of the multiway tree. In the derivation process, the structure and mathematical connotation of the formula follow the derivation rules strictly, eliminating errors and ambiguities, which is also easy to implement in programming languages.

For example, if the following formula needs to be expressed, the corresponding multiway tree of the formula is shown in FIG.1.

$$\delta = \int \frac{\bar{M}_i \bar{M}_j}{EI} dx$$

$$P(x, y, z, t) = dE/dS$$

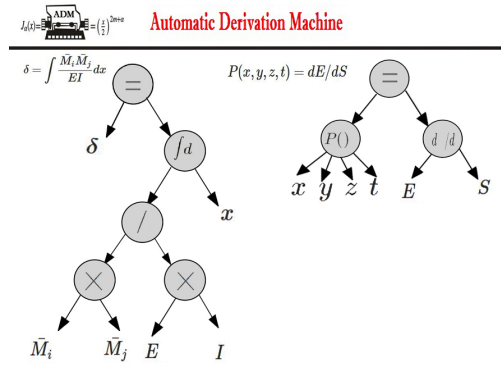


Figure 1: Using a multiway tree to express the formula, each operator is a node, and algebraic symbols or other operators that belong to this operator's scope of operation belong to this node's child nodes. Algebraic symbol nodes and numerical nodes can only be leaf nodes.

### 2.2 Subtree search algorithm

In order to complete the function of iterative learning, we propose the subtree search algorithm of the formula multiway tree. The subtree search algorithm refers to when given formula multiway tree  $F$  (such as  $\sin^2(x) + \cos^2(x) = 1$ ) and a simple template formula  $T$  (such as  $a + b = c$ ), judging whether the formula  $F$  conforms to the template formula  $T$ , that is, whether the formula multiway tree of  $F$  contains the subtree  $T$ .

The search algorithm is based on recursive implementation. Specifically, the subtree  $T$  is matched from the root of the  $F$  formula multitree, and if their symbols are the same, their child nodes would be compared, if their symbols are not the same, then start matching from the child nodes of  $F$  recursively. The specific algorithm is as follows:

For example, the formula  $dy = adx$  before derivation below is the subtree of the formula  $df(x) = \sin x dx$  before derivation, and the formula  $f(x) = \int \sin x dx$  can be deduced according to the derivation pattern of  $dy = adx \Rightarrow y = \int adx$ , which is also called the template mapping method proposed later in this paper.

---

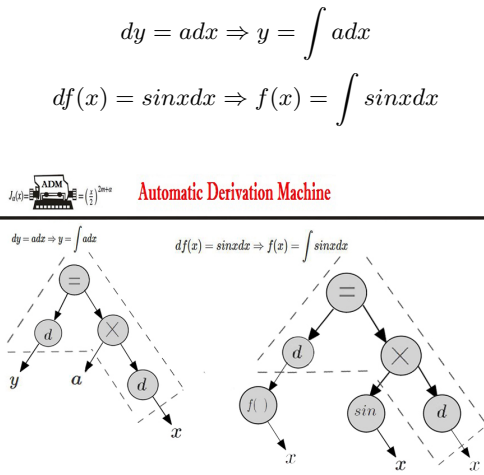
**Algorithm 1** subtree search algorithm  $find(root, sub)$

---

**Input:** Specific formula root node  $root$ ; root node of the formula subtree  $sub$ ; Specific node matching function  $TreeHasSub(root, sub)$  **Output:** if the template formula  $sub$  is the subtree of  $root$

- 1: **if** their symbols are the same:  $root.sym=sub.sym$  **then**
- 2:     **if**  $TreeHasSub(root, sub)$  **then** return true;
- 3:     **end if**
- 4: **end if**
- 5: **for** node in  $root.subNodes$  **do**
- 6:     return  $find(node, sub)$ ;
- 7: **end for**
- 8: return result

---

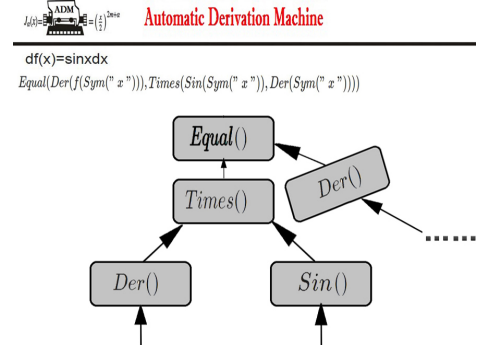


**Figure 2:** Using formula  $dy = adx$  as a subtree, performing subtree search algorithm in other formulas to find out if this subtree is included. For example, formula  $df(x) = sinxdx$  contains this subtree, which is equivalent to,  $df(x) = sinxdx$  is a specific case of the template formula  $dy = adx$ .

### 2.3 Construct declaration method

Because we need to manually build a large number of basic formulas for training in the later period, so when declaring a specific multiway tree for a formula, we need an efficient declaration method that meets people's thinking. Therefore, we propose a multiway tree construction method. Here, we only need to use a construct function in the programming language (C++ language is used in this article) to do the declaration of a formula. The construct function returns an instance of a formula multiway tree (in this article, the program returns the c++ point of the root node of a formula multiway tree). For example, we declare the following formula:

$$df(x) = sinxdx \Rightarrow Equal(Der(f(Sym("x"))), Times(Sin(Sym("x")), Der(Sym("x"))))$$



**Figure 3:** Expressions based on multiway trees are also suitable for using code expressions. Each operator is abstracted as a functional expression, which allows top-down use of code to express arbitrarily complex formulas easily.

## 3 FORMULA DERIVATION IMPLEMENTATION

### 3.1 Template mapping method

We propose the template mapping method that allows the formula to be deduced from the current state to the next step. Specifically, for example, a one-step derivation of the formula  $e^x sinx = m(x)t$ :

$$e^x sinx = m(x)t \Rightarrow e^x sinx/t = m(x)$$

This can be seen as done according to the template formula  $a = bc \Rightarrow a/c = b$ . First, the formula  $e^x sinx = m(x)t$  judges if the subtree  $a = bc$  is included, and then the template  $a = bc \Rightarrow a/c = b$  can be used to complete its derivation:  $e^x sinx = m(x)t \Rightarrow e^x sinx/t = m(x)$ .

### 3.2 Derivation by template mapping method

Template mapping can be seen as an abstraction of the specific derivation steps. It describes the transformation of the template formula  $T_1$  to the template formula  $T_2$ , and the template formula describes the simplest formula multiway tree that conforms to this derivation pattern. The template map can be described as:

$$M(T_1) = T_2$$

Where  $M(\cdot)$  represents some kind of derived transformation,  $T_1$  represents the template formula before derivation, and  $T_2$  represents the template formula after derivation. For example, addition and subtraction can be described as:

$$a + b = c \Rightarrow a = c - b$$

---

**Algorithm 2** Replacement algorithm  
*Replace*(*root*, *sub1*, *sub2*)

---

**Input:** Specific Formula A and the node *root* that coincides with the root node of *sub1*; **Template** formula *sub1* that matches formula A; **Template** formula *sub2* to be transformed. **Output:** Derived result formula *sub<sub>copy</sub>*

- 1: construct *Map* = *makeMap*(*root*, *sub1*) ,find the part of the formula that corresponds to each symbol in the template formula.
  - 2: construct *sub<sub>copy</sub>*, as a copy of *sub2*.
  - 3: **for** Traverse every corresponding node in *Map* **do**
  - 4:     Replace the corresponding node on *sub<sub>copy</sub>* with the corresponding specific part of the formula;
  - 5: **end for**
  - 6: return *sub<sub>copy</sub>*
- 

$$M(\text{tree}(a + b = c)) = \text{tree}(a = c - b)$$

Any formula derivation can be done based on the most basic template mapping. That is, the derivation of the formula  $F_1$  can be performed in a one-step derivation according to the mode of  $T_1 \Rightarrow T_2$ . This derivation method that depends on the template mapping can be expressed as  $\text{Trans}(F_1, T_1, T_2) = \text{Trans}(F_1, T_1, M(T_1)) = F_2$ . such as, for the formula  $m(x) + s(y) = T(x, y)$ , according to the template mapping  $a + b = c \Rightarrow a = cb$ , the transformation is:

$$m(x) + s(y) = T(x, y) \Rightarrow m(x) = T(x, y) - s(y)$$

$$\begin{aligned} \text{Trans}(\text{tree}(m(x) + s(y) = T(x, y)), \text{tree}(a + b = c), \text{tree}(a = c - b)) \\ = \text{tree}(m(x) = T(x, y) - s(y)) \end{aligned}$$

This method is difficult to solve the implicit conversion with physical connotation like  $\oint \frac{m}{p} dx \Rightarrow \oint \frac{dx}{v}$ . The solution to this problem is to break it down into several steps to complete, and need to add auxiliary symbols to complete, specifically:

$$\begin{aligned} \oint \frac{m}{p} dx &= S \xrightarrow{ab/c=d \Rightarrow a=dc/b} \oint dx = S \xrightarrow{p/m} \oint \frac{p}{m} dx \\ \oint dx &= Sv \xrightarrow{ab/c=d \Rightarrow a=dc/b} \oint \frac{dx}{v} = S \end{aligned}$$

### 3.3 Formula multiway tree replacement algorithm

In order to implement the function of deducing by template mapping, we propose the replacement algorithm of the formula multiway tree. After the formula  $F_1$  finds that the template formula  $T_1$  is its subtree, the replacement algorithm for the corresponding node of the formula  $F_1$  and the template formula  $T_1$  can be performed on the template formula  $T_2$  (this step is equivalent to the one in the mathematical operation: substituting another formula into the formula for calculation.), the final replaced formula  $T_2$  is the deduced result  $T_2$ . The specific algorithm is:

Based on the template mapping, we propose iterative learning. The iterative learning mechanism refers to the initial basic formula derivation template  $T_0 = (t_01, t_02)$ , where  $T_0$  is a simple enough formula transformation, and  $T_1 = (t_11, t_12)$  is a formula derived from  $T_0$  or a number of map templates of the same class containing  $T_0$ . By the mechanism like this,  $T_{n-1}$  can lead to a more complex formula transformation method  $T_n$ . Such formula derivation based on iterative transformation is called iterative learning.

## 4 FORMULA MULTIWAY TREE ENCODING

### 4.1 Benefits of feature encoding

Although the formula multiway tree can achieve derivation by searching for subtrees and replacement algorithms, formulas using multiway tree expressions cannot be input into the learner directly for training fitting and error metrics. When we need to measure the similarity of the following two formulas, the tree model is not intuitive:

$$te^x + m\cos x, te^{-x} - a\sin x$$

Therefore, we propose coding algorithm to encode the formula multiway tree and transform the formula multiway tree into the corresponding feature vectors. Which has three advantages:

- Each formula has a unique identifier.
- The similarity of the formula can be measured, the similar formula multiway trees have a smaller spatial distance.
- The formula multiway tree can be transformed into a feature vector to train the learner.

### 4.2 Encoding method

The specific encoding method is to assign an integer tag to each operator that appears in the formulas, do the similar Breadth-first traversal to the formula multiway tree from the top, and put the integer encoding of its nodes and child nodes into the feature vector. The specific process is as follows:

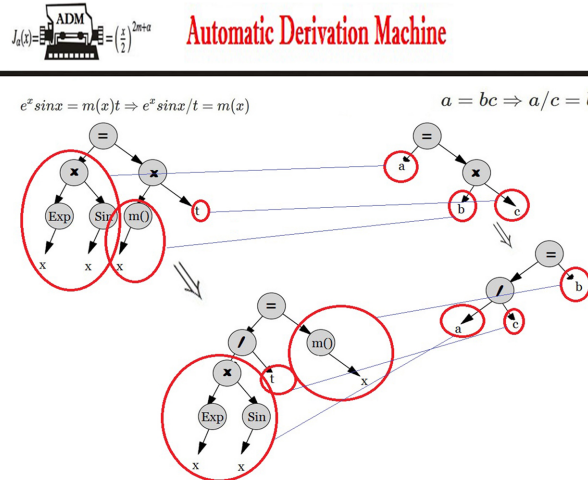


Figure 4: Replace algorithm. Step 1: find the corresponding nodes of the specific formula on the subtree *sub1* of the corresponding mode, and build the mapping relationship between them. Step 2: traverse this map and implement the replacement on the *sub2\_copy*. Step 3: mount the *sub2\_copy* back to the original tree and complete a one-step derivation.

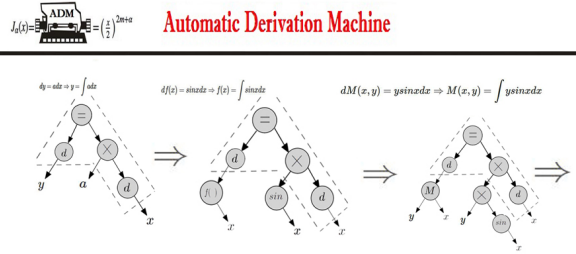


Figure 5: Using the basic template derivation formulas, the derived formulas continue to be used as templates for derivation, which is called iterative learning. Iterative learning can lead to complex formulas starting from the simplest formulas transformation.

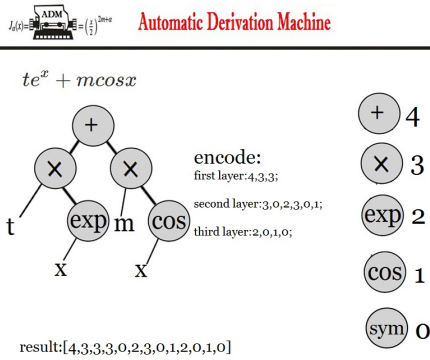


Figure 6: The multiway tree encoding algorithm is a top-down, breadth-first algorithm that assigns integer coding to all symbols firstly. Both the algebraic and numeric symbols are encoded as 0. Then the sub-layers are encoded from left to right.

### Algorithm 3 Formula multiway tree encoding

**Input:** formula multiway tree root node *root*;  
**Globe:** Maximum encoding length  $L_{max}$ ; Global encoding sequence  $V$ ; Global encoding set  $N_i$   
**Output:** feature vector  $V$

- 1: *encode*(*root*)
- 2: judge  $L_V < L_{max}$
- 3: clear temporary sequence  $S: S = []$ ;
- 4: Insert the root node code at the end of the temporary sequence:  $S.push(N_{root})$
- 5:
- 6: **for** *donode*  $\in$  *root.SubNodes*:  $S.push(N_{node})$
- 7: **end for**
- 8:  $V.push(S)$ ;
- 9:
- 10: **for** *donode*  $\in$  *root.SubNodes*: if *node.type* is not "Sym": *encode*(*node*);
- 11: **end for**
- 12: finally, insert  $L_{max} - L_V$  0 after  $V$ ;

### 4.3 Difference measure

After obtaining the eigenvectors of the two formula multiway trees, we need to define the calculation method for measuring the difference  $dis(tree_1, tree_2)$  between them. First of all, we have ensured that all the eigenvectors are of equal dimensions, and then compare each bit of them. If they are not equal, the comparison result is 1, otherwise, the comparison result is 0. Finally, the comparison result is the difference between the two formulas. For example, for the difference value calculation of formulas below:

$$dis(tree(te^x + m \cos x), tree(te^{-x} - a \sin x))$$

$$\begin{aligned}
&= dis([4, 3, 3, 3, 0, 2, 3, 0, 1, 2, 0, 1, 0], \\
&[5, 3, 3, 3, 0, 2, 3, 0, 6, 2, 3, 6, 0, 3, 0, 0]) \\
&= 4
\end{aligned}$$

Defining the formula multiway tree encoding method and differential value calculation method is of great significance to the follow-up learner training and prediction function. The encoding method can extract the features of the formula multiway tree, and convert it into a feature vector which can be calculated using the linear algebra method. The difference value calculation method can be used to complete the error convergence of the neural networks model.

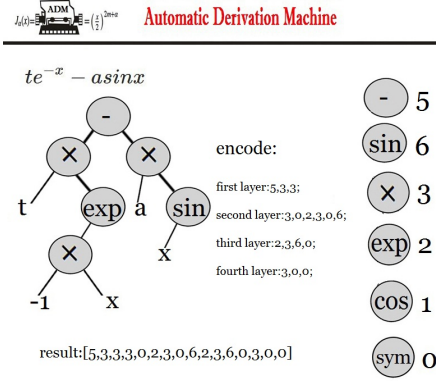


Figure 7: Formula multiway tree encoding of another tree.

## 5 LEARNING METHOD OF FORMULA DERIVATION

### 5.1 Reinforcement learning mode

We choose the reinforcement learning method to train the formula derivation machine. The reinforcement learning mode can be described as extracting an environment from the task to be completed, abstracting the state, action, and instantaneous reward received from performing the action [6]. The key elements of reinforcement learning are: environment, reward, action, and state. With these elements, a reinforcement learning model can be established. The problem of reinforcement learning is to obtain an optimal strategy for a specific problem, so that the reward obtained under this strategy is maximized. The so-called strategy is actually a series of actions, that is, sequential data.

We use the Q-learning learning model [5] in this paper. Q-learning's learning model needs to update a table named Q-table continuously. In this table,  $Q(s, a)$  represents the action  $a$  we choose at the state  $s$ , and Q-learning's learning process is to update every value in this table, that is, the expected earnings when taking action  $a$  under any state  $s$ . The specific update method is:

$$Q(s, a) = R(s, a) + \gamma MAX_{a'} Q(s', a')$$

Through continuous updating by training data, a converged Q-table can be finally used to select the action  $a$  with the largest expectation of revenue for the specific state  $s$ . The action can be thought of as a transformation of the state  $s$  into  $s'$ , which can be expressed as  $a(s) = s'$ .

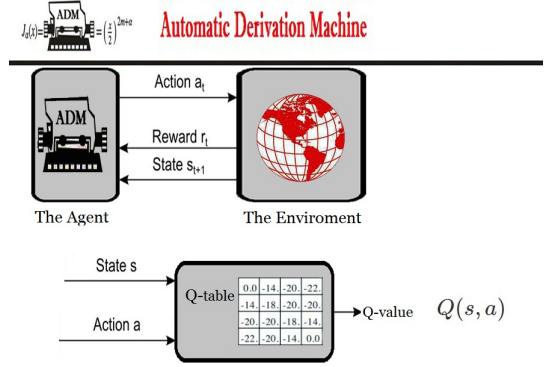


Figure 8: Q-learning is a state-based decision model, where each pair of  $(s, a)$  has a corresponding Q-value quantification in the Q-table to take the plausibility of acting  $a$  under the state  $s$ , and Q-table is the target for learning.

### 5.2 Reinforcement learning mode for automatic formula derivation

When the reinforcement learning mode is used to solve the problem of automatic formula derivation, the state set of the environment corresponds to the form of the formula multiway tree at different stages of derivation [3], and the agent-selectable action set corresponds to all the basic templates that can be selected when formulas deduced. The action can be taken as the template mapping  $(T_{at}, T_{bt})$  converts  $F_t$  to the formula  $F_{t+1}$ , which is the conversion method  $R(F_t, (T_{at}, T_{bt})) = F_{t+1}$ . Specifically it can be expressed as:

$$S = s1, s2... = F1, F2...$$

$$A = a1, a2... = (T_{a1}, T_{b1}), (T_{a2}, T_{b2})...$$

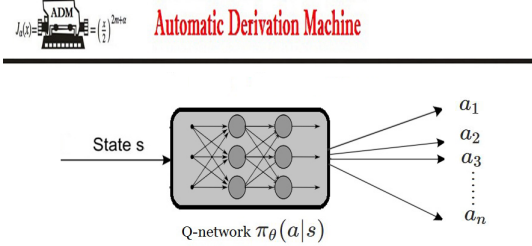
Therefore, the automatic derivation machine needs to calculate the decision probability  $\pi_\theta(a|s)$  at each step, that is, select the action  $a$  with the largest expected return (the one that is closest to the problem target) according to the existing state  $s$ . Then take the subtree search and replace algorithm to achieve a one-step formula derivation. The specific form of this probability:

$$\pi_\theta(a|s) = \pi_\theta((T_{at}, T_{bt})|F_t)$$

The neural network model and gradient descent method are used to realize the calculation of  $\pi_\theta(a|s)$  (in fact, any



other learning method can be used here to find this probability, and the accuracy may be better than neural network). The eigenvector of the formula multiway tree  $F_t$  is taken as input, and the output is the probability of the selectable action  $a$ .



**Figure 9: Using neural networks to learn the state-based decision map  $\pi_\theta(a|s)$ , the output is the probability distribution in the entire action space. The selected action is the action corresponding to the output probability with maximum value.**

### 5.3 Training of formula derivation

It is necessary to construct a training set about formula derivation before learning begins. The input element in each round of training is a formula  $F_t$  at a certain step. The output element of the training set is the optimal transformation selection  $(T_{at}, T_{bt})$ . That is,  $(X, y) = F_t, (T_{at}, T_{bt})$ .

Therefore, a complete formula derivation example can be decomposed into multiple derivation steps. Each step is a training sample, so a complete formula derivation can be converted into multiple training samples. A complete formula inference example can be expressed as:

$$F_0 \xrightarrow{(T_{a0}, T_{b0})} F_1 \cdots F_t \xrightarrow{(T_{at}, T_{bt})} F_{t+1} \cdots$$

The derivation of the solution to a certain physical equation velocity can be expressed as:

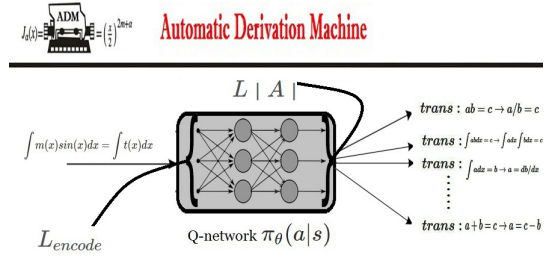
$$mv^2/2 + E = Q \xrightarrow{(a+b=c, a=c-b)} mv^2/2 = Q - E \cdots$$

$$v^2 = 2(Q - E)/m \xrightarrow{(a^2=b, a=\sqrt{b})} v = \sqrt{2(Q - E)/m} \cdots$$

The specific learning error improvement uses a gradient descent method:

$$\theta \leftarrow \theta - \epsilon \delta L(\pi_\theta(a|s), a^*)$$

In addition, in the training data, the probability of transformation method  $a^* = (T_{at}, T_{bt})$  needs to be converted into one-hot vector, which is  $a^* = (0, 0, 0, \dots, 1, \dots, 0)$ .



**Figure 10: Use neural networks to learn  $\pi_\theta(a|s)$ . The result of the output is the probability of taking each formula transformation.**

## 6 PROBLEM EXAMPLES

Now we use the above method to solve a practical problem, solving the concentration equation of  $Pm^{149}$  in reactor physics, which is a first-order linear differential equation:

$$dN_{Pm}(t)/dt = \gamma \sum \phi - \lambda N_{Pm}(t)$$

Firstly we need to construct the formula derivation training set for the first-order linear differential equation. The form of the first-order linear differential equation is like  $dy/dx + P(x)y = Q(x)$ . We hope that the auto-derivation can not only solve the concentration equation, it can also solve all first-order linear differential equations. So  $P(x)$  or  $Q(x)$  is equivalent to a collection of functions, that is:

$$P(x)_i = a, ax, ax^2 \dots ax^n \dots e^x, \sin x \dots$$

$$Q(x)_i = a, ax, ax^2 \dots ax^n \dots e^x, \sin x \dots$$

The training set  $D = (X_i, y_i)$  is constructed using equation inference examples containing these functions. The input element of the training set is a formula  $X_i = F_t$  at a certain step, and the output element of the training set is the best transformation chooses  $y_i = (T_{at}, T_{bt})$ , and finally we get the conditional probability  $\pi_\theta(a|s) = \pi_{\text{Theta}}(y|X)$  to make a derivation decision.

In the derivation process, each deduction will evaluate the next template selection according to the value of  $\pi_\theta(a|s)$ , and then transform the formula multiway tree according to the subtree matching and replacement algorithm, and then proceed to the next step until the final goal formula multiway tree is reached.

$$dN_{Pm}(t)/dt = \gamma \sum \phi - \lambda N_{Pm}(t) \xrightarrow{(a/b=c, a=cb)} \cdots$$

$$dN_{Pm}(t) = (\gamma \sum \phi - \lambda N_{Pm}(t))dt \xrightarrow{\cdots} \cdots$$

$$\int 1/(\gamma \sum \phi - \lambda N_{Pm}(t)) = \int dt \xrightarrow{(S=\int dt, S=t)} \cdots$$

$$\int 1/(\gamma \sum \phi - \lambda N_{Pm}(t)) = t \xrightarrow{\cdots} \cdots$$

**Table 1: Encoding of operators appearing in formulas in training data.**

Class	Encoding of operators
Compute symbols	<i>Sym/Num</i> , +, −, ×, =, $\int$ , $d$ , $\sum$ , $\phi$ , $/$ , $\sqrt{\cdot}$ , $d$ , $\ln(\cdot)$ , $\exp(\cdot)$ , $d/d$
Encoding number	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

## 7 DISCUSS

In order to obtain an automatic derivation machine that can derive mathematical formulas for various professional scientific research fields, we express the mathematical formulas in a multiway tree firstly and propose a subtree search algorithm and replacement algorithm for the formula multiway trees. Implementing the formula’s iterative learning function, we defined a method of formula derivation based on a simple formula mapping template. That is, only the simplest basic formula transformation set is given, and the subtree search algorithm and replacement algorithm of the formula multiway tree are used to make complex formulas follow the simple formula’s transformation rules. In order to measure the difference of the formulas and facilitate the learner learning the formula’s connotation, we propose an algorithm for encoding the formula multiway tree, transform the formula multiway tree into a feature vector. Finally, the neural network is constructed using the reinforcement learning model, and the gradient descent method is used to train the network to make the correct transformation decision according to the current formula state.

Although it is possible to deduce the correct answer, the training process is still very complicated because of the large amount data of neural network training required. Constructing a training set by man is a relatively large project, and the specific formula is more complicated such as a differential equation, so the requirements for training data producers are also higher than the requirements of the general machine learning data producers. Therefore, we should consider using smarter methods to generate formula data. For example, if writing a formula for training formulas according to certain rules, making training data will be more efficiently and accurately.

The encoding method of the formula multiway tree is still flawed, although the encoding method in this paper can measure the similarity of the formula structure, it can not measure the similarity between the calculated symbols. The encoding method of this paper will determine that the difference between + and  $E(x, y)$  and difference between + and − is equivalent, but in fact, for the specific problem, the difference and connotation between the symbols are not equivalent. Learning-based encoding methods like word2vec [4] will improve the above problem.

The formula multiway tree model and formula automatic derivation machine is a preliminary study for the automation and intelligence of scientific research, but it proposes a relatively efficient framework for re-expressing the abstract mathematical formulas of different scientific research fields

and establishing a general decision model to further derivation. Such a learning framework can be used for more complex and difficult scientific research from cross-domains, what needs to be done manually is to set basic rules and patterns, then using automatic derivation can be more efficient than artificially iterative learning and derivation. We hope that the automatic derivation machine will achieve meaningful derivation in specific scientific research fields.

## REFERENCES

- [1] Harith Al-Sahaf, Bing Xue, and Mengjie Zhang. 2017. A Multitree Genetic Programming Representation for Automatically Evolving Texture Image Descriptors. (2017).
- [2] George E. Collins and Hoon Hong. 1998. Partial Cylindrical Algebraic Decomposition for Quantifier Elimination. (1998).
- [3] Gregory Farquhar, Tim Rocktäschel, Maximilian Igl, and Shimon Whiteson. 2018. TreeQN and ATreeC, Differentiable Tree Planning for Deep Reinforcement Learning. (2018).
- [4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. (2013).
- [5] David Silver and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. (2016).
- [6] Richard S. Sutton. 1995. Generalization in reinforcement learning, successful examples using sparse coarse coding. (1995).
- [7] Alfred Tarski. 1951. A Decision Method for Elementary Algebra and Geometry. (1951).
- [8] Wen Jun Wu. 1986. Basic principles of mechanical theorem proving in elementary geometries. (1986).