Artificial Intelligence and Electrical & Electronics Engineering: AIEEE Open Access

PRIME OPEN ACCESS

Volume 1, Issue 1

Research Article

Date of Submission: 09 June, 2025
Date of Acceptance: 29 June, 2025
Date of Publication: 15 July, 2025

Artificial Neural Networks with Extensions and Correlated Run

Mirzakhmet Syzdykov*

Satbayev University, Astana, Kazakhstan

***Corresponding Author:**
Mirzakhmet Syzdykov, Satbayev University, Astana, Kazakhstan.

Citation: Syzdykov, M. (2025). Artificial Neural Networks with Extensions and Correlated Run. *Art Intelligence and Ele & Electronics Eng: AIEEE Open Access, 1*(1), 01-07.

Abstract
In this work we are demonstrating the supervised power of artificial neural networks within the scope of performance and other loyal theoretical background, which lays in-between the theory and practice, still we don't have a definite answer if it was defined before, since the applicated role of artificial intelligence which come to the trend during the recent time, thus, in this short work we are addressing the theoretical conveniences observed since the first humanity statement on the account of the human and artificial consciousness and mimics between them which lays the ground-breaking technological revolution to its first step. We give also the definition of the relation of probabilistic models towards construction of finite automata recognizing certain patterns for the given probability set and we are giving the definition of filter which is a rational relation between the repeated sequences in non-deterministic finite automata, we also show that it can be used in constructing deterministic automataon with respect to the filter condition, the definition of the aggregator normalization form of data in a database is also given. In this work we give the new algorithm for proving facts in logical directed graph using dynamic programming approach.

Keywords: Artificial Intelligence, Neural Networks, Extensions, Theory & Background

Introduction
The statement of theorem has a long-standing history [1]. We are considering the latest research using oracle-approach of proof, as well as our result on the "as is" incompatibility of complexity classes [2,3].The oracle and Turing automata have a review in [4-6].

In the cited research, most of cultural and social aspects of the global impact of AI on social processes and social transformation are discussed with the prior question of 'what could be the obstacle or regressive factor of the reactive development of AI technology gaining more and more heights in our society' [7]. The question was well-raised before from both technical and general point of view [8,9]. However, due to the handy nature of AI it gets more applications in our life, we also know the technical details of AI such as predictive function [10-12].

Artificial neural networks are a long-standing and novel approach in primarily changing human labor factor as well as a modern invention [13-15]. In general, human brain has a memory and even short memory which could be implied into artificial intelligence [16-18].

Discussions
Since oracle is posed as a separate element of the proof, however, as we know the separation leads to a 'greedy' and unknown strategy ignoring the certificate relation 'as is'.

As we know, if there is the set of certificate relations then we can inverse and build the set of alphabetical trees of the accepted words in Turing automata which is polynomial in time, space and, thus, complexity.

Let there be the case for NP-complete problem, such that there is the case where we know that both simplex and polynomial algorithms are correct, which is a satisfying condition for the proof of the fact that P equals NP. As we know for the set of complete problems to be resolvable as giving the result of P = NP or P <> NP, there should be at least half of the problems in NP-class to be either P- or NP-complete according to Dirichlet principle.

We are to define the AI socially independent or dependent according to its prediction function, which is defined as follows:

$$f(x) = P_f\big(X \vee (Y_1 ... Y_n)\big),$$

Where x or X is an event which is addressed to AI and $Y_1 ... Y_n$ are social correlated factors according to which the answer is evaluated.

As we know, we can assume that the answer of the function $f(x)$ given by AI can be evaluated first by the human operator and only then it can be subjected to the output condition.

Let's assume that the human expectation can be assumed as either constant or the different function as follows:

$$g(x) = P_g(X, Y_1 ... Y_n).$$

We give the following criteria of the adequacy of the answer proposed by AI and further evaluated by the human operator as:

$$f(x) \leq g(x) \vee \big|\big(f(x) - g(x)\big)\big| \leq e,$$

Where e is a measure of the value when results of both functions $f(x)$ and $g(x)$ can be considered same.
Since this is a global social process, these functions are probabilistic.

As we can now introduce the translatable programming language in our environment of operating system 'LavoroL' to abstract components like independent byte-code of virtual machine or the finite automata as it was already implemented in our project 'AQL', we are to use the step wise principle of evaluation in which the virtual machine is implemented and its abstract components are run on one step of evaluation giving then the control to another processes in the same abstract, or virtual, space.

Let's define the state space as $S = \{s_1, ..., s_m\}$, also the weighted function f: $(s_i, s_j) \rightarrow R$, where (s_i, s_j) is a transition between states in our model, then the heuristics function h is defined as follows:

$$h(s_i, s_j) = P\Big(\big(s_i, s_j\big) \in T : T = (s, t) \Leftrightarrow \sum_{(s,t)} f\,(a, b) \rightarrow argmin_f f(s, t)\Big).$$

From the definition of heuristics function we also can supply the additional matrix of feasible transitions by analogy of ant colony optimization. In the first case, we can use the product of the automata for regular language like "$e*$", where he is a symbol from alphabet set, as we know this construction has an exponential complexity. For the second case, we can show that even optimal construction of product automata has a product complexity of the size of states in automata – this happens for either single alphabet or any alphabet with closures.

As we know at each step the determinant is computed within the multiplication of (-1) or (+1) and a cell number at this position, however, if, for example, we would translate the quadratic matrix to the set where each symbol has a given position as if this would be the number of size n*n, the determinant of such value will be invariant towards for each cell in a row and each cell in column, which states the relation between the space N2, or Nm in general case.

Let's consider the most optimal strategy by introducing the queue with a given query l, where l is a length for which the paths in queue have length less or equal to this value. As we can see the number of active candidates in queue depends on the value l, however, for general case this problem is actually NP-complete if, for example, the regular path value, known as a sum of the edges of it, is sparse to the value l, which has a pre-defined step of operation.

We simply for each variable construct the map (..., *, *, ..., a, a, a, ...) where "*" stands for any symbol in binary alphabet {a, b} and "a" stands for the clauses where the variable is included either as negation or "as is". Then we construct the "and"-expression for all the variables and perform the DFA construction, after construction we can analyze each clause position in the map as either bi-polar or represented by single truth terminal "a", if there's only path then the clause is satisfied, else there is both sides of the value of clause and it's, thus, not determined.

By tackling tags, we can perform, the memory consideration for active variable choices according to the union operator. First for each variable in clause we are building the regular expression in the form of "...a..." for positive and "...b..." - for negative case, where our alphabet is dual.

Further for each of regular expression representing clause, we form the intersection as a Berry-Sethi automata, giving the abstract node on the output. Then, this output is subjected to the modified subset construction, which we have studied long time ago.

The complexity of the algorithm is $O(n*m*log(m))$, where n is a number of variables and m is a number of clauses, the logarithm shows that we build an optimal intersection operator tree. Before proceeding to the algorithm description, we have been absolutely correct in intractability of NP-complete problem like MAX-SAT using the present-day natural logic of computation. We build the same network as per using regular expressions and Brzozowski's derivatives with optimizations in our "Regex+" framework. The concurrency principle can be used if the same state of qubits could be saved in different paths of our function – this gives the linear algorithm.

This is a typical sigmoid function, used to give the probability of the fact in artificial neural network:

$$f(x) = \frac{1}{1+e^{-x}}.$$

The memory factor in terms of artificial intelligence could be defined as a re-definition of sigmoid function $f(x)$:

$$f(x,m) = \frac{1*m}{1+e^{-x}}, wherem = \{0...1\}.$$

Here the memory parameter m could be used in modeling natural processes of cognition where there is a definition of the context as if this could a typical Turing test for determining if the conversation is between human or artificial intelligence.

Since from hierarchy theorem we know that considered complexity classes form a strict hierarchy, we proved later that FIA solve the problem in linear time, we, thus, get the contradiction and in our corollary P = EXPTIME, i.e. polynomial complexity class is equal to exponential.

We know that for Schneider's canonical forms we can construct a solution verifier operating in time $O(1)$, which requires linear construction time O(n) to determine the parameter "t", the solution, thus gives the same answer towards the membership problem for the given regular expression:

$$f(r) = \{r(t) = 'b' \Rightarrow accept, rejectotherwise\}.$$

Thus, our function for regular expression "r" is either true or false according to the verifying condition.

Let's assume that P is a strict subset of EXPTIME complexity class, then:

$$P \subset EXPTIME \Rightarrow O\big(f(r)\big) = O\big(2^{|r|}\big).$$

However, as we know from our established fact:

$$O\big(f(r)\big) = O(1 + |(r)|) \neq O\big(2^{|r|}\big) \Rightarrow P \not\subset EXPTIME.$$

The above output is a strict contradiction towards hierarchy theorem and, with some assumption, towards the inequivivalence of complexity classes P and EXPTIME. For the next, we will show that these complexity classes are equal according to the complexity notation of the solution verifier.

So, this follows "as is", let's first assume the following fact:

$$P \subseteq NP \subseteq EXPTIME.$$

From this fact it follows that:

$$O(P) = O(|r|) \leq O(NP) \leq O(EXPTIME) = O\big(|2^{|r|}|\big).$$

However, as we have devised the unary and linear solution verifier, it follows that:

$$O(f(P)) = O(|r|) \leq O(f(NP)) \leq O(f(EXPTIME)) = O(1 + |r|) \Rightarrow O(P) = O(NP) = O(EXEXPTIME).$$

The above relation is a contradiction, thus:

$$P = NP = EXPTIME.$$

For our final and non-formal proof, we state that if the exponential problem can be solved in linear time and verified in unary, then, it follows that there's no differentiating relation between complexity classes as they collapse towards the singular point of complexity with measure $O(1)$.

As our conjecture is defined as well:

$$f(P) = NP.$$

Or in other words, there exist a function such that it converges P-class towards NP-class of complexity which defines intractable or the problems which cannot be solved in any visible amount of time. We have used the extended regular expressions and re-writing in "Regex+" software package, which is, in turn, a framework, in order to solve MAX-SAT problem on the defined set of tests.

The results have shown that it takes constant exponential time of number of variables in SAT expression in order to find any feasible solution or, simply, apply "halt" operation and output the negative result in case if there's no solution for this problem.

Let's assume that P = NP:

$$P = NP \Rightarrow f^{-1}(NP) = P.$$

As we know due to subset construction:

$$f^{-1}(NP) = P \Leftrightarrow O(NP) = O(P) \Rightarrow O(NP) = O(N) \neq O(2^N).$$

The above statement is a contradiction, thus:

$$P \neq NP.$$

For discrete fractal we know that:

$$f(n) = n! = \prod_{i=1}^{n} i.$$

The deviation of the function $f(n)$ projected on discrete plane is defined as well:

$$f'(n) = \frac{n! - (n-1)!}{1} = (n-1)(n-1)!.$$

Let's assume that our number is projected on to the real axis with infinite set:

$$|R| \gg |N|.$$

Then the discrete factorial will be defined as follows:

$$f(x, y) = \lim_{dx \to 0} \prod_{x}^{y} (x + i \cdot dx) = \lim_{dx \to 0} \left(x + \frac{(y-x)}{dx} \cdot dx\right)! = \left(x + (y - x)\right)! = \frac{y!}{x!}.$$

For the above outcome we have used C++ program – it shows that the above statement is also true for the $x > 1$ and for $x < 1$:

$$f(x, y) = 0.$$

Thus, as the mean of function $f(0, 1)$ will be as follows:

$\frac{\sum f(0,1)}{3} = \frac{2}{3}.$

For another binary constant we have:

$\frac{\sum f(0,2)}{3} = \frac{4}{3} \wedge \frac{\sum f(1,2)}{3} = \frac{3}{2}.$

The Collatz conjecture is one of the unsolved problems in mathematics and we give the full proof from both points by using residue functions.

Let's assume the following for Collatz function $f(n)$:

$$O(f(n)) \leq n.$$

Or, in other words, the length of the sequence of Collatz function is less than the number itself.

Let's consider another case where the number has least number of powers of two, $2^m - 1$:

$$O(f(n)) = O(f(2^m - 1)) \approx 3^m > 2^m - 1.$$

The above statement is a contradiction, thus, Collatz function cannot be approximated in the given number of steps.

To make a full proof of Collatz conjecture we will use the numbers modulo two in form: 2*k, 2*k + 1.

For the even numbers we have:

$$f(2 \cdot k) = f(k) = f(2 \cdot m + 1).$$

For odd numbers:

$$f(2 \cdot k + 1) = f(6 \cdot k + 4) = f(3 \cdot k + 2) = f(6 \cdot m + 2) = f(3 \cdot m + 1) = f(m).$$

Thus, we have the Collatz function defined for both even and odd numbers. The probability automata are a long-standing question, recently the question was raised again.

We define the function over the languages f:

$$f : P(\ldots) \rightarrow E.$$

As it can be seen the above function is defined for the probability of events which can be terminal or non-terminal as well as the symbols in regular language defined by the alphabet E.

$$f(a)=L(a), f(a \mid b)=L(a+b), f(a \mid (b, c, d))=L(b \cdot c \cdot d \cdot a), f(a \wedge b)=L(a) \wedge L(b).$$

The above notation can be used in proving and simulation of probability events on finite automata. We have defined the general function which can be used in defining the regular languages and giving the definition of the regular probabilistic languages.

Counter automata were studied before, recent research shows that they can be used in a variety of problem solving. The filter itself is a boolean function in subset construction, which can be implemented "on the fly" by considering the certain stable configuration of states in non-deterministic finite automaton within the counter. We have shown the new model of computation within the counter automata.

Database normalization forms is a way of representing it in a more complex and reliable way, recently the new normalization form was introduced based upon the set of input data. We present the data which could be aggregated according to the function in a row.

We define the set of aggregator functions for the row r as:

$F=\{f: r \rightarrow R\}$.

According to this function we aggregate the sum, production or any other operating according to function the set of functions *G*:

$$G = \{ t \in T : g_f(t) = g_t(f(r))\},$$

Where T is a set of tables where the row is to be aggregated and, thus, represent the final normalized form according to statistical computation. We have defined the new way of representing data according to aggregator normalization form, which is designed in order to store final data as a single row for the defined list of facts in data.

The boolean logic is a way of expressing the boolean expressions, recently the new methodology was proposed. The satisfiability problem is another concept for which we give the weighted MAX-SAT algorithm based upon the estimation of sets. We simply go through the graph by constructing strongly connected components and searching the new rule once all the incoming facts are satisfied.

As per our latest research the regular grammars and languages defined by regular expressions can be used as well in order to represent the logical processes within general and stochastic model. It can be noted that weight of any variable can be approximated and made tractable within its counted rate in satisfied conditions in boolean logical formula, based upon this we can give the weights with maximum cost and apply a greedy algorithm for searching the correct assignment of boolean variables in each of the clauses. We have given a concise definition of what is to be proved using logical rules or what is to be satisfied in the MAX-SAT problem, both cases illustrate the classical example of the machine approach in giving the solution to the stated problems.

Conclusion

We have given the main theorem on social impact of AI and its correctness, we have also concluded that no proper AI answer can be considered 'as is' due to social impact and, thus, is to be measured with probabilistic accuracy – this can be viewed as either a solution to the increasing role of AI in our life as well as an argument towards human interaction and AI evaluation. We have also given all the necessary definitions of the artificial intelligence with memory factor.

References

1. Cook, S. (2003). The importance of the P versus NP question. *Journal of the ACM (JACM), 50*(1), 27-29.
2. Muñoz de la Cuesta, Javier. (2025). Joint Theory: Self-Reference, Externality, and Infinite Elementsin Computability and Complexity. 10.13140/RG.2.2.11812.39047.
3. Louiz, A. (2023). A Proof That the Set of NP-problems is Bigger Than the Set of P-problems by Using a Logical Consideration. *WSEAS Transactions on Computers, 22*, 159-170.
4. Alman, J., & Chen, L. (2022). Efficient construction of rigid matrices using an NP oracle. *SIAM Journal on Computing*, (0), FOCS19-102.
5. Dueñas-Díez, M., & Pérez-Mercader, J. (2019). How chemistry computes: Language recognition by non-biochemical chemical automata. From finite automata to turing machines. *Iscience, 19*, 514-526.
6. Dori, S., & Landau, G. M. (2006). Construction of Aho Corasick automaton in linear time for integer alphabets. *Information Processing Letters, 98*(2), 66-72.
7. Zaniewska, Katarzyna. (2025). AI Narratives to Shape the Future Introduction: Defining ProblemSpace (What). 10.13140/RG.2.2.18693.67042.
8. Hwang, T. (2018). Computational power and the social impact of artificial intelligence. arXiv preprint arXiv:1803.08971.
9. Hagerty, A., & Rubinov, I. (2019). Global AI ethics: a review of the social impacts and ethical implications of artificial intelligence. *arXiv preprint arXiv:1907.07892*.
10. Sheikh, S. (Ed.). (2020). *Understanding the role of artificial intelligence and its future social impact*. IGI Global.
11. Erdélyi, O. J., & Goldsmith, J. (2018, December). Regulating artificial intelligence: Proposal for a global solution. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 95-101).
12. Adams, S. J., Henderson, R. D., Yi, X., & Babyn, P. (2021). Artificial intelligence solutions for analysis of X-ray images. *Canadian Association of Radiologists Journal, 72*(1), 60-72.
13. Yegnanarayana, B. (2009). Artificial neural networks. PHI Learning Pvt. Ltd.
14. Krenker, A., Bešter, J., & Kos, A. (2011). Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications*. InTech, 1-18.
15. Wu, Y. C., & Feng, J. W. (2018). Development and application of artificial neural network. *Wireless Personal Communications, 102*, 1645-1656.
16. Neves, G., Cooke, S. F., & Bliss, T. V. (2008). Synaptic plasticity, memory and the hippocampus: a neural network approach to causality. *Nature reviews neuroscience, 9*(1), 65-75.
17. Aram, Z., Jafari, S., Ma, J., Sprott, J. C., Zendehrouh, S., & Pham, V. T. (2017). Using chaotic artificial neural networks to model memory in the brain. *Communications in Nonlinear Science and Numerical Simulation, 44*, 449-459.

18. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., ... & Hassabis, D. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature, 538*(7626),471-476.