

ТИМОШЕНКО Павло

Національний університет «Львівська політехніка»  
<https://orcid.org/0000-0002-5510-1963>  
[pavlo.tymoshenko.knm.2018@lpnu.ua](mailto:pavlo.tymoshenko.knm.2018@lpnu.ua)

ЗАСОБА Євген

Національний університет «Львівська політехніка»  
<https://orcid.org/0000-0003-4830-8306>  
[geka.zasoba@gmail.com](mailto:geka.zasoba@gmail.com)

КОВАЛЬЧУК Олександр

Національний університет «Львівська політехніка»  
<https://orcid.org/0000-0002-0148-0320>  
[oleksandr.v.kovalchuk@lpnu.ua](mailto:oleksandr.v.kovalchuk@lpnu.ua)

ПШЕНИЧНИЙ Олександр

Національний університет «Львівська політехніка»  
<https://orcid.org/0000-0001-8823-7472>  
[oleksandr.y.pshenychnyi@lpnu.ua](mailto:oleksandr.y.pshenychnyi@lpnu.ua)

## НЕЙРОЕВОЛЮЦІЙНІ АЛГОРИТМИ ДЛЯ ГЕНЕРУВАННЯ НЕЙРОННИХ МЕРЕЖ

Вирішення інженерних задач за допомогою звичайних нейронних мереж вимагають тривалих досліджень щодо вибору архітектури та гіперпараметрів. Сильний штучний інтелект був би позбавленим таких недоліків. Подібні дослідження проводяться з використанням дуже широкого спектру підходів: наприклад, біологічного (спроби виростити мозок у лабораторних умовах), апаратного (створення нейронних процесорів) або програмного (з використанням потужностей звичайного CPU та GPU). Метою роботи є розроблення такої системи, яка дозволяла би за допомогою еволюційних підходів генерувати нейронні мережі, придатні для розв'язування задач. Подібний носить назву «нейроеволюції». До мети цієї роботи також входить дослідження особливостей можливих застосованих еволюційних стратегій.

**Ключові слова:** нейроеволюційний алгоритм, нейронна мережа, модель машинного навчання, препроцесинг даних.

TYMOSHENKO Pavlo, ZASOBA Yevgen, KOVALCHUK Olexander, PSHENYCHNYI Olexander  
Lviv Polytechnic National University

## NEUROEVOLUTIONARY ALGORITHMS FOR NEURAL NETWORKS GENERATING

Solving engineering problems using conventional neural networks requires long-term research on the choice of architecture and hyperparameters. A strong artificial intelligence would be devoid of such shortcomings. Such research is carried out using a very wide range of approaches: for example, biological (attempts to grow a brain in laboratory conditions), hardware (creating neural processors) or software (using the power of ordinary CPUs and GPUs). The goal of the work is to develop such a system that would allow using evolutionary approaches to generate neural networks suitable for solving problems. This is called "neuroevolution". The purpose of this work also includes the study of the features of possible applicable evolutionary strategies.

The object of research in this work is a neuroevolutionary approach to solving problems of machine learning. The subject of research is evolutionary strategies, neural coding methods networks in the organism's genome. The scientific novelty of the work lies in the testing of previously unused evolutionary strategies and the generalization of the obtained system to the systems of "general artificial intelligence". A system for simulating neuroevolution was created. The specifics of implementation were considered, the choice of algorithms was justified, and their work was explained. In order to perform experiments, datasets were created and methods of applying neuroevolutionary systems were developed. It was possible to choose the most optimal training parameters, to find out the relationship between them, as well as the accuracy and speed of training. It cannot be said that the models implemented within this work directly bring us closer to strong AI. They still lack their own memory as well as a certain level of complexity. For successful use, it is necessary to configure the view of the input data or perform some calculations outside the model. However, in the future, such a system can be developed, for example, to work with SNNs, or for use on special equipment

**Keywords:** neuroevolutionary algorithms, artificial neural network, machine learning model, data preprocessing.

### Вступ

Останнім часом штучний інтелект розвивається дуже потужно. Розвиток галузі уже давно не відображається в підручниках, бо кожного дня можна очікувати на вихід такої статті, що здатна перевернути бачення порядку денного з ніг на голову. Досягнутий успіх можна побачити наочно: вже зараз нас оточують голосові помічники різного стибу, розумна видача контенту у стрічках соцмереж, «розумні» фільтри в Instagram тощо. Є також і менш доступні для загалу, але більш цікаві проекти: мовна модель GPT 3 вміє спілкуватись зі співрозмовником [1], аналізувати його фрази та давати цілком осмислені відповіді; модель DALL-E 2 «розуміє» речі, зображені на поданій картинці, може змінювати контекст зображення, або й взагалі генерувати нові картинки з поданого опису [2].

Не зважаючи на свої успіхи, сама концепція «штучного інтелекту» часто зіштовхується з критикою із боку філософів. Наприклад, у своїй відомій роботі «Is the Brain's Mind a Computer Program» (укр. «Розум мозку – комп'ютерна програма?») Джон Серль, як заперечення можливості здатності комп'ютера мислити, наводить експеримент із «китайською кімнатою». З цього експерименту випливає, що «одного вміння

маніпулювати символами недостатньо, щоб гарантувати знання, сприйняття, розуміння, мислення; а оскільки комп'ютери, як такі, – це засоби, що маніпулюють символами, існування однієї комп'ютерної програми недостатньо, щоби можна було говорити про наявність знання» [3]. Мовляв, комп'ютер оперує лише синтаксисом, а той не породжує семантику. Серль також називає сучасний штучний інтелект «слабким» або «вузьким». Справді, він є слабким у тому сенсі, що створені моделі здатні вирішувати лише одну конкретну виділену їм задачу і позбавлені будь-якої свідомості. Хоча, заради справедливості можна зауважити, що дискусія на тему того, чим є свідомість є окремою темою.

#### Аналіз літературних джерел

Є й більш адекватні визначення сильного штучного інтелекту. Наприклад, за [4] такий інтелект повинен могли пояснити причину обраного рішення. А от за [5], цей інтелект повинен володіти семантичною мапою світу. Це хоч і потрібна, однак надлишкова вимога, позаяк здатний пояснити свої рішення організм володіє семантикою уже за визначенням. Без сумніву, цей інтелект повинен постійно навчатися, експериментувати та робити помилки. За [6] ШІ повинен вміти комунікувати з людиною природною мовою. Однак, це теж не обов'язкова вимога й обґрунтування її необхідності руйнуються так само просто, як і твердження, що свідомість не може існувати поза межами людської мови. Уже з'ясовано, що собаки для нас є істотами, достатньо інтелектуальними для того, щоби їхній інтелект можна було вважати «сильним». А проте, вони не можуть висловлювати своїх думок бодай через те, що у них відсутня ділянка мозку, відповідальна за такі задачі. Є й узагалі глухонімі люди, які не знають жодної мови. Це не заважає їм чинити виважено й інтелектуально. Отже, мислення не обов'язково повинно приймати вигляд якихось логічних мовних конструкцій. Ймовірно, здатність ословлюватись мовою, носієм якої є істота – це просто навичка, – така сама, як і їзда на велосипеді. Якщо це правда, то для ШІ доцільно будувати системи, які оперують насамперед образами, а от лінгвістичну інтерпретацію власного мислення та його результатів виконують опціонально. Окрім того, такий підхід дозволить створити системи, здатні швидко переходити між людськими мовами – якби така система існувала, вона вирішила б усі задачі машинного перекладу раз і назавжди.

При створенні системи сильного ШІ доцільно брати приклад з природи та імітувати ті механізми, які демонструють досить непоганий інтелект – наприклад, мозок [7]. Найперша проблема, з якою ми можемо зустрітись при спробах імітувати мозок, – паралелізм виконання. Справа в тім, що у сучасній фонНейманівській структурі обчислювальних машин паралелізм забезпечується перемиканням контексту – призупиненням інших задач для виконання однієї. Тобто, така машина може виконувати лише послідовну обробку інформації. Навіть якщо ми використаємо  $N$  процесорів одночасно, це все ще буде  $N$ -потоківна обчислювальна машина, в той час, як у мозку людини всі 86 мільярдів нейронів однаково задіяні у мисленнєвих процесах, готові реагувати на будь-які подразники та взаємодіяти між собою [8]. Очевидно, для відтворення мозку нам потрібна машина, здатна симулювати роботу нейронів у реальному часі. Далі подано список деяких пристроїв, над якими сьогодні триває розробка, і які вирішують цю проблему.

- Akida NSoC – нейроморфний процесор від Brainchip. Стверджується, що він може працювати з 1.2 млн. нейронів та 10 млрд синаптичних зв'язків. Призначений для оптимізації роботи згорткових та повнозв'язних нейронних мереж.

- Neurogrid – процесор, що симулює роботу біологічного мозку. Використовує аналогове обчислення. Може симулювати до мільйона нейронів та до 6 млрд синаптичних зв'язків у режимі реального часу.

- Loihi – нейроморфний 14 нм процесор від Intel, що може симулювати 8 млн. нейронів та є приблизно у 1000 разів потужнішим і у 10 тис. разів енергоефективнішим, ніж звичайний CPU.

- SpiNNaker – система, що складається з 30 тис. чіпів, кожен з яких має 18 ядер та 128 Мб RAM-пам'яті. В сумі система здатна симулювати 0.5 млн нейронів, працювати в реальному часі, і споживає 30 кВт енергії.

- BrainScaleS 1 – система, здатна симулювати 4 млрд. нейронів біологічно подібних нейронів та 109 пластичних синаптичних зв'язків.

#### Основний матеріал

Нейронна мережа, що проектується цією системою, не має чіткої шарової структури. Це означає, що у нас немає деякого вхідного шару з фіксованим розміром, на який можна подати дані. Натомість мережа на кожному етапі еволюції випадковим чином може відрощувати або знищувати нейрони, які вважаються вхідними. Перед створенням мережі для неї задаються параметри того, яка максимальна кількість вхідних та вихідних нейронів може розвинути в процесі еволюції. Кожен нейрон мережі нумерований.

Припустимо, що ми маємо мережу з максимальною кількістю вхідних нейронів  $|I^U|$ , і подаємо на вхід вектор даних  $D^I$ . Обов'язкова умова такого вектора –  $|D^I| < |I^U|$ . Якщо на якомусь етапі еволюції мережа змогла розвинути лише вхідні нейрони  $I \subseteq I^U$  і ми знаємо їхні порядкові номери за деяким відношенням  $\text{num} : I^U \rightarrow Z$ , то це означатиме, що вхідними даними такої нейронної мережі буде вектор  $D^I$ , елементи якого визначені так:

$$\begin{cases} \hat{D}_i^I = D_i^I, & \text{якщо } \exists I_j \in I : \text{num}(I_j) = i, \\ 0 & \text{в іншому разі.} \end{cases}$$

Неформально кажучи, на вхідні дані є такі обмеження: їхня розмірність не повинна перевищувати кількості вхідних нейронів. При цьому, сама мережа сприйматиме тільки ті елементи сприйматиме даних, для яких у неї існують відповідні вхідні нейрони.

Нейронну мережу можна натренувати для даних будь-якої природи. Для цього було згенеровано дані для кількох задач.

У задачі «Лабіринт» нейронна мережа керує гравцем, що ходить по лабіринту. Маємо безкінечну двовимірну матрицю  $L$ , на якій прямокутник розміром  $n \times m$  заповнений значеннями, що задовольняють умові  $\forall i \in 0, n, j \in 0, m : L_{ij} \in \{0, 1\}$ . Усі клітинки, які перебувають за межами цього прямокутника заповнені значенням 1. На кожному ході гравець може перебувати у позиції  $P_{ij}$  тільки якщо  $L_{ij} = 0$ . Якщо гравець вирішує ступити на деяку клітинку, в якій міститься 1, то вважається, що він програв. Мапа генерується таким чином, щоби від точки  $(0, \lfloor \frac{m}{2} \rfloor)$  завжди існувала хоча б одна доріжка до точки  $(n, \lfloor \frac{m}{2} \rfloor)$ , по якій можна дійти, здійснюючи кроки  $(i \pm 1, j)$  та  $(i, j \pm 1)$ .

У складених лабіринтах містяться компоненти (1). Для того, щоби орієнтуватись у ділянці (а), мережі доведеться навчитись оминати перешкоди. Ділянка (б) містить глухий кут – мережа повинна виявити таку ситуацію, і зрозуміти, що їй варто зробити крок назад. У випадку (в) мережі доведеться робити вибір, яким має бути її наступний крок, і, можливо, повертатись назад за потреби.

$$\begin{matrix}
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 \dots & 0 & 0 & 0 & \dots & \dots & 0 & 0 & 0 & \dots & \dots & 1 & 0 & 1 & \dots \\
 \dots & 0 & 1 & 0 & \dots & \dots & 1 & 0 & 1 & \dots & \dots & 0 & 0 & 0 & \dots \\
 \\
 \dots & 0 & 0 & 0 & \dots & \dots & 1 & 1 & 1 & \dots & \dots & 1 & 0 & 1 & \dots \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 \text{(а)} & & & & & \text{(б)} & & & & & & \text{(в)} & & & & 
 \end{matrix} \tag{1}$$

У задачі «Космічний корабель» нейронна мережа управляє космічним кораблем, який неперервно рухається з кроком  $(i + 1, j)$  у просторі  $S$  розміром  $n \times m$ , де  $m \gg n$ . У цьому просторі  $\forall j, i \in \{1, m\} : S_{ij} = 1$ . Мережа може бачити наступний до себе стовпчик. В просторі розташовані «астероїди», задача мережі – оминати їх. Якщо корабель потрапив на клітинку зі значенням 1, то гра завершується. Щоби виграти, мережа може переміщувати корабель у будь-який інший рядок. Приклад ігрового поля подано як (2).

$$\begin{bmatrix}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \dots \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\
 \triangleright & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & \dots \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \dots \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \dots
 \end{bmatrix} \tag{2}$$

Приклад вхідних даних при поданій ситуації на ігровому полі показано на (3). На цьому прикладі прямо перед кораблем не розташовано жодних перешкод, тому він може продовжувати летіти не змінюючи курсу. Як видно, вектор вхідних даних доповнюється одиницями таким чином, щоби рядок перед кораблем завжди був відцентрований.

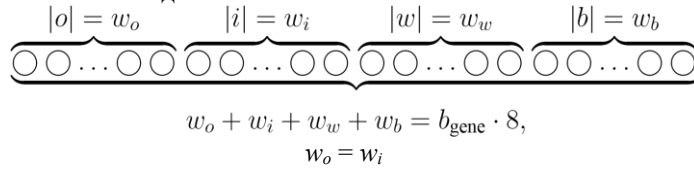
$$\begin{bmatrix}
 1 & 1 & \dots \\
 \triangleright & 0 & \dots \\
 0 & 0 & \dots \\
 0 & 0 & \dots \\
 1 & 1 & \dots
 \end{bmatrix} \rightarrow \left[ \begin{array}{cccccccc}
 \underline{1} & 1 & 1 & 0 & 0 & 0 & 1 & \\
 & & & \Delta & & & & 
 \end{array} \right] \tag{3}$$

Нейронні мережі також тестувались і на класичній задачі класифікації. Для цього було згенеровано  $n$ -вимірні сепарабельні дані з  $(n - 1)$  ознаками та однією змінною для передбачення, що може приймати цілочисельні значення з проміжку  $[0, c]$ . Такі дані утворюють множину

$$\begin{aligned}
 D &= \{ \mathbf{d}^{(0)}, \dots, \mathbf{d}^{(N)} \}, \\
 \mathbf{d}^{(i)} &= [x_1^{(i)}, \dots, x_{n-1}^{(i)}, y^{(i)}]; \\
 \forall i, j : (x_j^{(i)} &\sim \mathcal{U}(0, c)) \ \& \ (x_j^{(i)} \in \mathbb{R}) \\
 y^{(i)} &\in [0, c] \subset \mathbb{Z},
 \end{aligned}$$

У цій системі ген – це  $(b_{\text{gene}} \cdot 8)$ розрядне число, що складається з  $b_{\text{gene}}$  байтів. Система збудована таким чином, що у гені можна закодувати від 1 до  $8b_{\text{gene}}$  різних значень, виділяючи під них певні біти.

Конкретно для симулювання еволюції нейронних мереж у кожному гені закодовується 4 значення у такому порядку: порядковий номер вихідного та вхідного вузлів, сила зв'язку та деяка біасконстанта. У послідовності бітів це має такий вигляд:



Таким чином кожен ген у геномі визначає параметри синаптичного зв'язку між двома нейронами. Кількість вхідних  $n_i$  та вихідних  $n_o$  нейронів для мережі є параметрами моделі. Якщо порядковий номер гена кодується ( $w_o$ )бітним числом, то це означає, що у моделі всього може існувати ( $2^{w_o} - 1$ ) нейронів. Розподіл нейронів на вхідні, вихідні та проміжні продемонстровано у табл. 1.

Таблиця 1

**Розподіл типів нейронів за порядковими номерами**

| Тип нейронів | Проміжок порядкових номерів        | Приведення у номер всередині типу   |
|--------------|------------------------------------|-------------------------------------|
| Вхідні       | [1, $n_i$ ]                        | $\text{num}(n) = n$                 |
| Вихідні      | [ $n_i \cdot 2^{w_o} - 1 - n_o$ ]  | $\text{num}(n) = n - 2^{w_o} - n_o$ |
| Проміжні     | [ $2w_o \quad n_o, 2w_o \quad 1$ ] | $\text{num}(n) = n \quad n_i$       |

Також параметром моделі є число  $p_w \in R$ . Воно визначає межі  $[-p_w, p_w]$ , в яких може міститися вага синаптичного зв'язку. У гені зберігається число  $w$ , що набуває значень з проміжку  $[0, 2^{w_w}] \subset Z$ . Для того, щоби нормалізувати значення ваги до потрібних меж, виконується перетворення (4).

$$w^*(w) = p_w(-1 + w \cdot 2^{-w_w+1}) \tag{4}$$

Аналогічне перетворення відбувається і з параметром  $b$ , для якого також вводиться параметр  $p_b$ .

Для вимірювання пристосованості моделі у цій задачі використовувалась метрика точності. Результати експерименту подано у табл. 2.

Таблиця 2

**Залежність точності від параметрів тренування на 1000-й та 5000-й епохах. Використано 6-вимірні дані, 10 класів.  $w_o = w_i = 16$ ,  $w_w = w_b = 16$ . Кількість проміжних нейронів – 65520.**

| Мутація $\mu$ | Змішування $\beta$ | Розмір вибірки $b$ , % | 1000 епох | 5000 епох |
|---------------|--------------------|------------------------|-----------|-----------|
| 0.1           | 0.1                | 10                     | 23%       | 31%       |
| 0.3           | 0.1                | 10                     | 73%       | 85%       |
| 0.5           | 0.1                | 10                     | 42%       | 67%       |
| 0.3           | 0.5                | 10                     | 37%       | 51%       |
| 0.3           | 0.8                | 10                     | 37%       | 44%       |
| 0.3           | 0.5                | 20                     | 87%       | 93%       |
| 0.3           | 0.5                | 40                     | 32%       | 51%       |

Також в межах цього експерименту проведено дослідження того, як розмір комбінації для кросоверингу впливає на швидкість тренування. Це показано у табл. 3.

Таблиця 3

**Залежність точності від розміру комбінацій для кросоверингу.  $\mu = 0.3$ ,  $\beta = 0.5$ ,  $b = 20$**

| Розмір комбінацій | 1000 епох | 5000 епох |
|-------------------|-----------|-----------|
| 1                 | 13%       | 21%       |
| 2                 | 87%       | 93%       |
| 3                 | 92%       | 95%       |
| 4                 | 91%       | 94%       |

5

89%

93%

### Висновки

У межах цієї роботи було створено систему для симуляції нейроеволюції. Було розглянуто особливості імплементації, обґрунтовано вибір алгоритмів та пояснено їхню роботу.

Задля виконання експериментів було створено набори даних та розроблено методи застосування систем нейроеволюції. Вдалося підібрати найбільш оптимальні параметри тренування, з'ясувати зв'язок між ними, а також точністю та швидкістю тренування.

Не можна сказати, що моделі, реалізовані у межах цієї роботи безпосередньо наближають нас до сильного ШІ. Їм все ще бракує власної пам'яті а також певного рівня складності. Для вдалого використання доводиться конфігурувати вигляд вхідних даних або ж проводити деякі обчислення поза межами моделі. Однак у подальшому таку систему можна розвинути, наприклад, для роботи з SNN, або для використання на спеціальному обладнанні.

### References

1. Garcia A., Arbelaitz O., Linaza M.T., Vansteenwegen P., Souffriau W. (2010). Personalized Tourist Route Generation. In: Daniel F., Facca F.M. (eds) Current Trends in Web Engineering. ICWE 2010. Lecture Notes in Computer Science, vol 6385. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-16985-4\\_47](https://doi.org/10.1007/978-3-642-16985-4_47)
2. Ramesh Aditya Zero Shot Text to Image Generation. arXiv.org. URL: <https://arxiv.org/abs/2102.12092> (03.06.2022).
3. Searle J. R. Is the Brain's Mind a Computer Program? Scientific American. 1990. T. 262, № 1. С. 26–31. URL: <https://doi.org/10.1038/scientificamerican0190-26> (03.06.2022).
4. Russell S. J. Artificial intelligence: A modern approach. 2 ed. Upper Saddle River, N.J.: Prentice Hall, 2003. 1080 p.
5. Stubblefield W. A., George F. Luger Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 3 ed. Addison Wesley Publishing Company, 1997. 868 p.
6. Russell S. J. Artificial intelligence: A modern approach. 2 ed. Upper Saddle River, N.J.: Prentice Hall, 2003. 1080 p.
7. Butz M. V. Towards Strong AI. KI Künstliche Intelligenz. 2021. T. 35, № 1. P. 91—101. URL: <https://doi.org/10.1007/s13218-021-00705-x> (23.05.2022).
8. Frederico A. C. Azevedo Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled up primate brain. The Journal of Comparative Neurology. 2009. T. 513, № 5. P. 532—541. URL: <https://doi.org/10.1002/cne.21974> (23.05.2022).