**УДК: 004.8:004.21** DOI 10.33111/mise.104.12

**Позднякович О.Є.**,
к.ф.-м.н., доцент кафедри системного аналізу та кібербезпеки,
КНЕУ імені Вадима Гетьмана

**Pozdniyakovych O.E.,**
Candidate of Physical and Mathematical Sciences, Associate Professor
of System analysis and cybersecurity
KNEU named after V. Hetman

# МЕТОДИ ОПТИМІЗАЦІЇ ГІПЕРПАРАМЕТРІВ У МАШИННОМУ НАВЧАННІ

# HYPERPARAMETER OPTIMIZATION METHODS IN MACHINE LEARNING

***Анотація**. Стаття присвячена проблемі оптимізації гіперпараметрів у машинному навчанні. Проводиться аналіз різних методів для поліпшення продуктивності моделей. У статті досліджено сучасні підходи до оптимізації гіперпараметрів у машинному навчанні, що є критично важливими для підвищення точності та узагальнюючої здатності моделей глибокого навчання. Розглянуто проблему залежності ефективності моделей від вибору гіперпараметрів, таких як швидкість навчання, параметри регуляризації та архітектурні особливості мережі. Проаналізовано переваги та обмеження основних методів оптимізації: базових (grid search, random search), байєсівських, еволюційних алгоритмів, а також методів зі змінною точністю (successive halving, Hyperband). Особливу увагу приділено використанню байєсівської оптимізації із сурогатними моделями на основі гауссових процесів, які дозволяють прогнозувати значення цільової функції та рівень невизначеності. Описано функції збору, такі як Expected Improvement, Probability of Improvement та Upper Confidence Bound, що керують процесом вибору нових гіперпараметрів. Показано, що ефективне використання обчислювальних ресурсів, зокрема через методи змінної точності, є ключовим чинником у задачах масштабного налаштування моделей. Визначено, що еволюційні алгоритми забезпечують гнучкість у дослідженні простору гіперпараметрів, хоча мають високу обчислювальну складність. У висновках підкреслено важливість автоматизації процесу налаштування гіперпараметрів, інтеграції цих методів у життєвий цикл моделі та перспективність подальших досліджень у напряму масштабованих і розподілених обчислень. Результати мають практичну цінність для побудови продуктивних моделей у середовищах з обмеженими ресурсами та високими вимогами до точності прогнозування.*
***Ключові слова**: оптимізація гіперпараметрів, машинне навчання, глибокі нейронні мережі.*

***Abstract**. The article is devoted to the problem of hyperparameter optimization in machine learning. Different methods for improving model performance are analyzed. The article explores modern approaches to hyperparameter optimization in machine learning, which is critically important for enhancing the accuracy and generalization capability of deep learning models. It addresses the*

*issue of model performance dependency on the selection of hyperparameters such as learning rate, regularization parameters, and network architecture. The study analyzes the strengths and limitations of the main optimization methods: basic (grid search, random search), Bayesian, evolutionary algorithms, and variable fidelity methods (successive halving, Hyperband). Special attention is given to Bayesian optimization using surrogate models based on Gaussian processes, which allow for predicting the value of the objective function and the level of uncertainty. Acquisition functions such as Expected Improvement, Probability of Improvement, and Upper Confidence Bound are described as key mechanisms in guiding the selection of new hyperparameters. The article shows that efficient use of computational resources, particularly through variable fidelity approaches, is a crucial factor in large-scale model tuning tasks. It is determined that evolutionary algorithms provide flexibility in exploring the hyperparameter space, albeit with high computational complexity. The conclusions emphasize the importance of automating the hyperparameter tuning process, integrating these methods into the model development lifecycle, and the prospects for future research in scalable and distributed computing. The findings are practically valuable for building high-performance models in resource-constrained environments with strict predictive accuracy requirements.*
***Keywords**: optimization of hyperparameters, machine learning, deep neural networks.*

**Постановка проблеми.** Сучасні глибокі нейронні мережі значною мірою залежать від різноманітних гіперпараметрів, які визначають архітектуру моделі, методи регуляризації та стратегії оптимізації. Важливим завданням у машинному навчанні є налаштування цих гіперпараметрів для оптимізації продуктивності.

**Аналіз останніх досліджень і публікацій.** У процесі оптимізації гіперпараметрів було встановлено, що різні конфігурації гіперпараметрів можуть по-різному впливати на результати для конкретного набору даних [1]. Очевидно, що правильне налаштування гіперпараметрів відіграє ключову роль у підвищенні ефективності моделей глибоких нейронних мереж. Дослідження показали, що методи оптимізації гіперпараметрів можна застосовувати для адаптації універсальних алгоритмів машинного навчання до конкретних завдань і сфер застосування [2]. Загальновизнано, що ретельне налаштування гіперпараметрів значно перевершує стандартні налаштування за замовчуванням, які пропоновані популярними бібліотеками машинного навчання, що покращує як продуктивність моделей, так і їхню здатність адаптуватися до різних наборів даних [3-6].

**Метою статті є** дослідження сучасних методів і підходів до оптимізації гіперпараметрів у машинному навчанні.

**Виклад основного матеріалу.** Задача оптимізації гіперпараметрів полягає в знаходженні такого набору гіперпараметрів, який мінімізує похибку моделі на валідаційному наборі даних

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}}{argmin}\, \mathbb{E}_{(D_{train,}, D_{valid})\sim\mathcal{D}} \boldsymbol{V}\left(L, A_{\boldsymbol{\lambda}}, D_{train,}, D_{valid}\right),$$

де $\boldsymbol{\lambda}^*$ – шуканий набір оптимальних гіперпараметрів, $\boldsymbol{\Lambda}$ — множина можливих конфігурацій гіперпараметрів, може включати такі параметри, як швидкість навчання, параметри регуляризації, кількість шарів у нейронній мережі тощо; $\mathbb{E}_{(D_{train,}, D_{valid})\sim\mathcal{D}}$ — математичне очікування за розподілом пар даних для навчання ($D_{train}$) і валідації ($D_{valid}$), які обирають із деякого розподілу $\mathcal{D}$; $\boldsymbol{V}\left(L, A_{\boldsymbol{\lambda}}, D_{train,}, D_{valid}\right)$ — функція, яка обчислює метрику оцінки (наприклад, похибку моделі) на валідаційному наборі даних, $L$ — функція втрат (наприклад, MSE, крос-ентропія), яка вимірює, наскільки добре модель передбачає результати, $A_{\boldsymbol{\lambda}}$ — алгоритм навчання, який залежить від гіперпараметрів $\boldsymbol{\lambda}$, $D_{train}$ — навчальний набір даних, $D_{valid}$ — валідаційний набір даних.

Методи оптимізації гіперпараметрів можна поділити на кілька груп на основі стратегії пошуку, використання даних і обчислювальних ресурсів.

*Базові методи.* Пошук по ґратці (grid search) — метод, що передбачає встановлення діапазонів значень для кожного гіперпараметра і перевірку всіх можливих комбінацій цих значень. Це забезпечує повний перегляд параметрів, і оптимальна конфігурація гіперпараметрів буде знайдена, якщо вона міститься в ґратці параметрів, заданій користувачем. Однак, цей метод може бути дуже трудомістким при великій кількості гіперпараметрів або широких діапазонах.

У методі випадкового пошуку (random search) [7] також визначаються діапазони значень для кожного гіперпараметра, але замість перевірки всіх можливих комбінацій, випадковим чином обирається фіксована кількість комбінацій для перевірки. Такий підхід дозволяє досліджувати ширший діапазон значень гіперпараметрів і робить це більш ефективно з точки зору обчислювальних витрат і часу, але метод випадкового пошуку може бути неефективним для великих просторів гіперпараметрів через випадковість і відсутність використання інформації з попередніх випробувань для коригування подальших.

*Байєсівські методи.* У байєсівській оптимізації ключову роль відіграють імовірнісна сурогатна модель і функція збору, які спільно розв’язують задачу вибору нових точок для оцінки цільової функції. Сурогатна модель використовується для апроксимації складної цільової функції, яку важко оцінити через обмежені обчислювальні ресурси або високу вартість оцінок. Ця модель оновлюється після кожного нового вимірювання і враховує всі попередні спостереження, дозволяє швидко робити прогнози. Основна мета

сурогатної моделі — створити спрощену ймовірнісну модель, яка може точно передбачати значення цільової функції в нових точках і оцінювати при цьому рівень невизначеності кожного прогнозу.

Гауссові процеси [8] є однією з найпоширеніших моделей для побудови таких сурогатних моделей. Вони не тільки прогнозують значення цільової функції, але й забезпечують розподіл імовірності для кожного прогнозу, що включає середнє значення та дисперсію. Гауссові процеси є ймовірнісними моделями, які представляють функції як випадкові величини, що визначені на основі гауссових розподілів. Головна перевага гауссових процесів у тому, що вони можуть оцінювати невизначеність своїх прогнозів. Основні компоненти таких процесів:

- *Прогноз для точки*. Прогнозоване значення функції в точці $\lambda$ — це нормальний розподіл із середнім значенням $\mu(\lambda)$ та дисперсією $\sigma^2(\lambda)$: $f(\lambda) \sim N(\mu(\lambda), \sigma^2(\lambda))$.
- *Оновлення моделі після отримання нових даних*. Кожне нове спостереження дозволяє оновити модель, коригуючи її прогнози на основі попередніх даних. Для цього використовується коваріаційна матриця $K$, що оцінює залежність між різними точками. Ядрова функція $k(\lambda, \lambda')$ (наприклад, радіально-базисна функція або інші) використовується для визначення подібності між точками $\lambda$ і $\lambda'$. Передбачення середнього і дисперсії можуть бути отримані таким чином:

$$\mu(\lambda) = k(\lambda, X)(k(X, X) + \sigma_n^2 I)^{-1} y,$$

$$\sigma^2(\lambda) = k(\lambda, \lambda) - k(\lambda, X)(k(X, X) + \sigma_n^2 I)^{-1} k(X, \lambda),$$

де $X$ — матриця точок спостережень, $k(\lambda, X)$ і $k(X, \lambda)$ — коваріації між новою точкою і точками в навчальній вибірці, $k(X, X)$ — коваріаційна матриця для точок в наборі даних $X$, $k(\lambda, \lambda)$ — коваріація нової точки $\lambda$ з самою собою, $y$ — спостережувані значення цільової функції, $\sigma_n^2$ — дисперсія шуму, $I$ — одинична матриця.

Крім гауссових процесів, існують інші підходи до побудови сурогатних моделей:

- *Дерева рішень і випадкові ліси*. Ці моделі використовують деревоподібні структури для розділення даних і створення апроксимацій. Вони можуть бути швидшими для великих наборів даних.
- *Методи на основі поліноміальних або радіально-базисних функцій*. Ці методи застосовуються для більш гладких функцій, де відомі властивості форми функції.

Функція збору прогнозує корисність нових точок завдяки балансу між дослідженням нових гіперпараметрів і використанням уже відомих хороших значень. Байєсівська оптимізація активно використовує цей принцип для пошуку глобальних оптимумів.

Найбільш поширеними функціями збору є Expected Improvement (EI), Probability of Improvement (PI) і Upper Confidence Bound (UCB).

Функція збору Expected Improvement (EI) [9] оцінює очікуване поліпшення відносно поточного найкращого значення цільової функції $f_{min}$. Вона максимізує очікування поліпшення в новій точці $\lambda$

$$EI(\lambda) = \mathbb{E}[\max(f_{min} - f(\lambda), 0)] =$$
$$= \big(f_{min} - \mu(\lambda)\big)\Phi\left(\frac{f_{min} - \mu(\lambda)}{\sigma(\lambda)}\right) + \sigma(\lambda)\varphi\left(\frac{f_{min} - \mu(\lambda)}{\sigma(\lambda)}\right),$$

де $\mu(\lambda)$ — передбачене значення сурогатної моделі в точці $\lambda$, $\sigma(\lambda)$ — стандартне відхилення передбачення в точці $\lambda$, Ф — функція стандартного нормального розподілу, $\varphi$ — щільність стандартного нормального розподілу.

EI поєднує в собі як вибір точок, які можуть поліпшити поточне рішення, так і врахувує невизначеності, пов'язані з новими точками.

Функція збору Probability of Improvement (PI) максимізує ймовірність того, що нова точка $\lambda$ приведе до поліпшення порівняно з поточним мінімумом $f_{min}$

$$PI(\lambda) = \Phi\left(\frac{f_{min} - \mu(\lambda)}{\sigma(\lambda)}\right),$$

де $\mu(\lambda)$ — передбачене значення сурогатної моделі в точці $\lambda$, $\sigma(\lambda)$ — стандартне відхилення передбачення в точці $\lambda$, Ф — функція стандартного нормального розподілу.

PI орієнтована на поліпшення вже відомих рішень.

Функція збору Upper Confidence Bound (UCB) [10] обирає точки, де комбінація передбаченого значення $\mu(\lambda)$ і його невизначеності $\sigma(\lambda)$ максимальні

$$UCB(\lambda) = \mu(\lambda) + k\,\sigma(\lambda),$$

де $\mu(\lambda)$ — передбачене значення сурогатної моделі в точці $\lambda$, $\sigma(\lambda)$ — стандартне відхилення передбачення в точці $\lambda$, $k$ — параметр, який контролює баланс між вибором точок з високою невизначеністю і вибором точок з високим передбаченим значенням.

Ці функції збору використовують комбінацію передбачень сурогатної моделі та оцінки невизначеності, щоб оптимізувати пошук у просторі параметрів.

Байєсівські методи мають декілька недоліків, хоча вони є популярними та ефективними:

- *Висока обчислювальна складність*. Побудова та оновлення сурогатної моделі можуть бути обчислювально витратною, особливо при збільшенні кількості даних або гіперпараметрів. Побудова коваріаційної матриці стає повільною і складною для великої кількості точок.
- *Обмеження за розміром даних*. Методи, засновані на гауссових процесах, не підходять для задач із тисячами точок через суттєве зниження продуктивності.
- *Чутливість до вибору сурогатної моделі*. Якість оптимізації суттєво залежить від якості використовуваної сурогатної моделі. Якщо сурогатна модель не відповідає дійсності, ефективність оптимізації знижується.
- *Повільний процес навчання*. Байєсівська оптимізація проводить пошук на основі вже спостережуваних даних. Хоча вона ефективна для невеликої кількості ітерацій, велика кількість гіперпараметрів можуть зробити процес повільним.

Незважаючи на ці недоліки, байєсівські методи є потужним інструментом для оптимізації гіперпараметрів, особливо у разі обмежених обчислювальних ресурсів або коли потрібно мінімізувати кількість викликів цільової функці.

*Еволюційні алгоритми.* Ці алгоритми [11] використовують для оптимізації гіперпараметрів, застосовуючи принципи біологічної еволюції, такі як відбір, схрещування і мутація. У межах алгоритму кожну конфігурацію гіперпараметрів розглядають як можливий розв’язок задачі.

На кожному етапі оцінюється якість рішень за допомогою функції пристосованості, наприклад, на основі точності моделі на валідаційному наборі даних. Найкращі рішення відбирають і комбінують для генерації нових конфігурацій гіперпараметрів із використанням механізмів схрещування (об’єднання параметрів двох рішень) і мутації (випадкова зміна окремих параметрів).

Цей процес повторюється поки не буде досягнуто оптимального рішення або виконано критерії завершення, такі як задана кількість ітерацій.

Еволюційні алгоритми ефективні в задачах із великим числом гіперпараметрів і складними взаємозв’язками між ними. Недоліки полягають у високій обчислювальній складності при значних роз-

мірах популяції та залежності від якості функції пристосованості, а також ефективності операцій схрещування і мутації.

*Методи оптимізації зі змінною точністю.* Збільшення розмірів наборів даних, ускладнення моделей є основною перешкодою для успішної оптимізації гіперпараметрів, оскільки вони роблять оцінку продуктивності моделі більш витратною.

Популярним підходом для прискорення налаштування гіперпараметрів стало використання невеликої підмножини даних. Це досягається завдяки навчанню на обмеженій кількості ітерацій, роботі з частиною ознак або виконанню лише одного чи кількох етапів перехресної перевірки. Методи змінної точності формалізують ці евристики, спираючись на так звані спрощені апроксимації реальної функції втрат, яку необхідно мінімізувати. Такі апроксимації являють собою баланс між точністю оптимізації та часом обчислень.

Методи змінної точності включають підходи, які аналізують і моделюють криві навчання [12, 13], щоб вирішити, чи потрібно виділяти додаткові ресурси або припиняти навчання для поточної конфігурації гіперпараметрів. Криві навчання можуть являти собою, наприклад, зміну якості однієї й тієї самої конфігурації на підмножинах даних, що збільшуються, або залежність продуктивності ітераційного алгоритму від кількості виконаних ітерацій (або кожної $i$-ї ітерації, якщо обчислення метрики продуктивності вимагає значних ресурсів). Точність методів залежить від здатності коректно прогнозувати криві навчання. Невірні прогнози можуть призвести до відхилення потенційно успішних моделей.

Метод послідовного ділення навпіл (successive halving) — це метод оптимізації гіперпараметрів, який розподіляє обчислювальні ресурси (наприклад, час навчання, кількість даних, кількість епох) між безліччю конфігурацій гіперпараметрів. Основна ідея методу полягає в тому, щоб почати з великої кількості конфігурацій гіперпараметрів, для цього надати кожній з них невелику кількість ресурсів. Поступово відсікати найменш успішні конфігурації і виділити тим, що залишилися, більше ресурсів. У результаті знайти найкращу конфігурацію, що дозволить заощадити обчислювальні ресурси. Головний недолік методу — потрібно заздалегідь визначити загальний бюджет ресурсів, і якщо він обраний неправильно, результати можуть бути неефективними або неточними.

Hyperband [14] — складніша версія методу послідовного ділення навпіл. Основна ідея цього методу полягає в тому, щоб ефективно розподілити обмежений бюджет обчислювальних ресу-

рсів між множиною гіперпараметрів. Метод комбінує підхід послідовного ділення навпіл, який відсіває менш перспективні конфігурації на ранніх етапах, з адаптивною стратегією розподілу ресурсів. Hyperband розглядає різні сценарії розподілу бюджету — від тестування великої кількості конфігурацій з невеликими ресурсами до глибокого аналізу невеликої кількості конфігурацій. Це дає йому змогу одночасно досліджувати широкий спектр гіперпараметрів і заглиблюватися в найперспективніші варіанти, забезпечуючи баланс між широтою і глибиною оптимізації. Однак метод hyperband може бути неефективним за високих витрат на кожну ітерацію навчання або за значних рівнів шуму в даних, що знижує точність відбору конфігурацій.

**Висновки та перспективи подальшого дослідження.** Вибір відповідних методів оптимізації гіперпараметрів значно впливає на продуктивність моделей машинного навчання. Різні методи, такі як байєсівська оптимізація, еволюційні алгоритми демонструють переваги залежно від структури завдання та обсягу даних. Методи змінної точності, такі як hyperband і successive halving, забезпечують компроміс між часом виконання і якістю моделі, що робить їх корисними для завдань з обмеженими ресурсами.

Включення методів оптимізації гіперпараметрів у стандартні цикли розроблення моделей спрощує роботу з великими і складними моделями, що мінімізує потребу в ручному налаштуванні.

Незважаючи на успіхи в галузі оптимізації гіперпараметрів, залишаються проблеми машинного навчання, які не були безпосередньо розв’язані через розмір простору конфігурацій, витрати на оцінку окремих моделей, і які можуть вимагати нових підходів [15].

З огляду на важливість паралельних обчислень, перспективним напрямком є розробка методів, що повною мірою використовують можливості великомасштабних обчислювальних кластерів.

## Бібліографічні посилання

1. Kohavi, R., John, G.: Automatic Parameter Selection by Minimizing Estimated Error. In: Prieditis, A., Russell, S. (eds.) Proceedings of the Twelfth International Conference on Machine Learning, pp. 304–312. Morgan Kaufmann Publishers (1995).

2. Escalante, H., Montes, M., Sucar, E.: Particle Swarm Model Selection. Journal of Machine Learning Research 10, 405–440 (2009).

3. Mantovani, R., Horvath, T., Cerri, R., Vanschoren, J., Carvalho, A.: Hyper-Parameter Tuning of a Decision Tree Induction Algorithm. In: 2016 5th Brazilian Conference on Intelligent Systems (BRACIS). pp. 37–42. IEEE Computer Society Press (2016).

4. Olson, R., La Cava, W., Mustahsan, Z., Varik, A., Moore, J.: Data-driven advice for applying machine learning to bioinformatics problems. In: Proceedings of the Pacific Symposium in Biocomputing 2018. pp. 192–203 (2018).

5. Sanders, S., Giraud-Carrier, C.: Informing the Use of Hyperparameter Optimization Through Metalearning. In: Gottumukkala, R., Ning, X., Dong, G., Raghavan, V., Aluru, S., Karypis, G., Miele, L., Wu, X. (eds.) 2017 IEEE International Conference on Big Data (Big Data). IEEE Computer Society Press (2017).

6. Thornton, C., Hutter, F., Hoos, H., Leyton-Brown, K.: Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In: Dhillon, I., Koren, Y., Ghani, R., Senator, T., Bradley, P., Parekh, R., He, J., Grossman, R., Uthurusamy, R. (eds.) The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13). pp. 847– 855. ACM Press (2013).

7. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research 13, 281–305 (2012).

8. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. The MIT Press (2006).

9. Balcan, M., Weinberger, K. (eds.): Proceedings of the 33rd International Conference on Machine Learning (ICML'17), vol. 48. Proceedings of Machine Learning Research (2016).

10. Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. W. (2012). Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory, 58(5), 3250-3265.

11. Simon, D.: Evolutionary optimization algorithms. John Wiley & Sons (2013).

12. Kohavi, R., John, G.: Automatic Parameter Selection by Minimizing Estimated Error. In: Prieditis, A., Russell, S. (eds.) Proceedings of the Twelfth International Conference on Machine Learning, pp. 304–312. Morgan Kaufmann Publishers (1995).

13. Provost, F., Jensen, D., Oates, T.: Efficient progressive sampling. In: Fayyad, U., Chaudhuri, S., Madigan, D. (eds.) The 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99). pp. 23–32. ACM Press (1999).

14. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A.: Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research 18(185), 1–52 (2018).

15. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115(3), 211–252 (2015).

*Статтю подано до редакції: 03.12.2024*