

О.В. Гавриленко, Н.В. Новіченко

ДОСЛІДЖЕННЯ ОПТИМІЗАТОРІВ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ БУДІВЕЛЬ ЗА АРХІТЕКТУРНИМ СТИЛЕМ

Анотація. У статті розглянуто задачу класифікації зображень з тісними міжкласових взаємозв'язками – класифікація архітектурних стилів будівель, де велика кількість основних рис та ознак є спільною для декількох класів. Об'єктом дослідження є алгоритм навчання нейронної мережі для розпізнавання архітектурних стилів будівель. Запропоновано метод навчання нейронної мережі для класифікації архітектурних стилів будівель за зображеннями будівель, що за меншу кількість часу навчання досягає більшої. Запропонований алгоритм оптимізатору реалізовано програмно і проведено експерименти для порівняння ефективності алгоритму.

Ключові слова: машинне навчання, нейронні мережі, оптимізатори, класифікація зображень, архітектурні стилі.

Постановка проблеми. Задача класифікації архітектурного стилю будівель відрізняється від стандартних задач класифікації наявністю міжкласових взаємозв'язків між різними стилями, тобто велика кількість основних рис та ознак є спільною для декількох стилів. З збільшенням цифрових даних необхідність у їх обробці буде зростати, адже класифікація зображень використовується в багатьох галузях, таких як освіта, безпека, результати класифікації є основою для багатьох соціально-економічних та екологічних застосувань. Точність класифікації зображень може мати життєво важливе значення якщо використовується, наприклад, в медицині.

Аналіз останніх досліджень і публікацій. Задача класифікації архітектурних стилів вже була розглянута раніше. Наприклад, у роботі «Visual pattern discovery for architecture image classification and product image search» [1] реалізовано алгоритм класифікації за 4 архітектурними стилями за допомогою виділення візуальних патернів шляхом створення просторових конфігурацій для того, що б мати можливість отримати точні результати при масштабуванні об'єкта, обертанні та деформації. Алгоритм, розроблений Z. Xu представлений у роботі «Architectural Style Classification using Multinomial Latent Logistic

Regression» [2] розроблявся з використанням якісної вибірки зображень будівель понад 20 архітектурних стилів. Вводячи поняття «ймовірність» в аналіз архітектурного стилю, автори аналізують взаємозв'язки архітектурних стилів імовірно. У роботі «Automatic architectural style recognition» представлена реалізація алгоритму NBNN для класифікації архітектурних стилів будівель [3]. Підхід для класифікації запропонований у роботі «Architectural Style Classification of Building Facade Windows» [4] полягає у виділенні текстур та форм в опис зображення.

Розглянуті роботи мають свої переваги, такі як простота реалізації, швидкість навчання, проте майже всі вони ефективні лише при малій кількості класів або ж для зображень будівель, що не мають багато деталей; інші ж вміють розрізняти лише ті будівлі, на яких є куполи, вежі або колони.

Мета дослідження. Об'єктом дослідження є процес навчання нейронної мережі - коригування ваг за допомогою ітераційного процесу, зміні значень параметрів мережі таким чином, щоб функція втрат була зведена до мінімуму. Щоб досягти мінімум функції існують алгоритми зміни ваг мережі - оптимізатори, такі як методи градієнтного спуску або Adam. Предметом дослідження є оптимізатори - алгоритми або методи, що використовуються під час навчання нейронної мережі для зміни її атрибутів, таких як ваги та швидкість навчання. Метою роботи є зменшення часу навчання нейронної мережі при збереженні точності навчання.

Викладення основного матеріалу дослідження. Задача класифікації зображення полягає у присвоєнні об'єкту (ситуації) деяким чином певний клас з існуючого переліку. Існує велика кількість методів вирішення поставленої задачі, кожен з яких має свої особливості і має різну ефективність залежно від особливостей задачі та вихідних даних. Найбільш популярними та ефективним методом вирішення задачі класифікації зображень є нейронні мережі, а саме CNN – це архітектура штучних нейронних мереж, що заснована на механізмах зорової кори ока, саме тому ця архітектура є найбільш ефективною для роботи з зображеннями. Такі нейронні мережі намагаються імітувати роботу мозку людини та процес його навчання. Так само, як мозок приймає вхідні дані, обробляє їх і генерує певний результат, так і працює нейронна мережа. Три дії - отримання вхідних даних, обробка інформації, генерування вихідних даних - представлені у вигляді шарів у нейронній мережі - вхідних, прихованих та вихідних [5].

Алгоритм CNN включає в себе 2 основних процеса: згортка та об'єднання. Повний навчальний процес нейронної мережі включає два етапи: пряме і зворотне поширення.

Пряме поширення. Зображення подаються на вхідний шар у вигляді матриці цифр. Ці числові значення позначають інтенсивність пікселів на зображенні. Нейрони в прихованих шарах застосовують кілька математичних операцій над цими значеннями з використанням певних значення параметрів - ваг, які випадковим чином ініціалізуються перед початком навчання і змінюються в процесі навчання. Результат надсилається на вихідний рівень, який генерує остаточний результат класифікації.

Зворотне поширення. Після отримання результату класифікації, наступним кроком є порівняння результату з фактичним значенням. Залежно від кінцевого результату та наскільки це близьке до фактичного значення (помилки), значення параметрів мережі оновлюються. Процес прямого розповсюдження повторюється з використанням оновлених значень параметрів і генеруються нові результати поки помилка не стане мінімальною [6].

Дано довільну вибірку зображень $X_{i,j}$, що розподілені за M класами. Необхідно мінімізувати значення функції втрат, що відповідає за загально похибку мережі при навчанні. Функція втрат показує, яка частина зображень при використанні мережі може бути класифікована невірно.

Дано:

$\{C\}$ – кількість класів, архітектурних стилів;

$\{c_i\}$ – множина класів, архітектурних стилів, де $i = \overline{1 \dots C}$;

$\{X_{i,j}\}$ – зображення будівлі, що відноситься до класу $c_{X_j} \in C$;

$\{N\}$ – кількість зображень;

$\{p_{model} [X_j \in c_{X_j}]\}$ – ймовірність, що зображення $X_{i,j}$ належить до класу c_{X_j} ;

$$L = -\frac{1}{N} \sum_{i=1}^N \log p_{model} [X_j \in c_{X_j}] \rightarrow \min \quad (1)$$

Обмеження:

$i, j, c_i, N \in N$;

$2 < N, C < \alpha$;

$0 \leq L < \infty$;

На кожній ітерації необхідно змінити ваги нейронної мережі та швидкість навчання таким чином, щоб функція втрат (1) зменшилася. Нові значення ваг

та швидкості навчання визначають оптимізатори. Розглянемо деякі найбільш популярні з них.

Градiєнтний спуск (GD) - це алгоритм оптимізації, який використовується при навчанні моделі машинного навчання для пошуку локального мінімуму диференційованої функції. Він базується на опуклій функції та ітеративно налаштовує її параметри, щоб мінімізувати дану функцію до її локального мінімуму. Простота обчислень та легкість реалізації роблять його популярним, проте він має суттєві недоліки: може зупинитися в локальному мінімуму; ваги змінюються після обчислення градієнта для всього набору даних, при великому набору даних знаходження мінімумів може зайняти роки; потребує багато пам'яті для обчислення градієнта для всього набору даних [7].

Деякі недоліки градієнтного спуску вирішує алгоритм SGD. Він є розширенням алгоритму градієнтного спуску, GD для обчислення похідної функції втрачає одночасно завантажує весь набір даних з n точок. У алгоритмі SGD похідна обчислюється, приймаючи по одній точці за раз. Проте, час обчислень для 1 епохи більший за GD. Існує ще одна модифікація - SGD з імпульсом - вона має всі переваги алгоритму SGD та збігається швидше, ніж алгоритм GD. Проте, на кожному етапі потрібно обчислити ще одну змінну для кожного оновлення. Алгоритм SGD варто використовувати лише для неглибоких мереж [8].

Вище розглянуті алгоритми не впливали ніяк на швидкість навчання, тому розглянемо AdaGrad - алгоритм, що встановлює адаптивний рівень навчання для кожного з ваг. AdaGrad змінює загальну швидкість навчання на кожному часовому кроці для кожного параметра на основі минулих градієнтів. Головна перевага AdaGrad - не потрібно оновлювати швидкість навчання вручну, оскільки вона адаптивно змінюється з ітераціями. Основний недолік - накопичення квадратних градієнтів, накопичена сума постійно зростає під час навчання, внаслідок чого швидкість навчання зменшується; може виникати проблема зникаючого градієнта.

Проблему зменшення швидкості навчання вирішує AdaDelta - замість неефективного зберігання попередніх квадратних градієнтів, сума градієнтів рекурсивно визначається як середнє усіх минулих квадратних градієнтів. RMSprop алгоритм майже ідентичний AdaDelta, так як розробили їх майже одночасно для вирішення недоліків AdaGrad [9].

Найбільш популярний алгоритм Adam можна розглядати як поєднання RMSprop та SGD з імпульсом. Він має в собі переваги обох алгоритмів і має

найменшу «вартість» навчання. Проте, якщо градієнт певної ваги великий (або сильно змінюється), то алгоритм не настільки ефективний, як SGD. [10]

Розглянемо процес оновлення ваг нейронної мережі на кожній ітерації.

$$g_t = \text{grad}(L), \quad (2)$$

$$W^{[l]} = W^{[l]-1} - \alpha g_t. \quad (3)$$

На кожній ітерації необхідно обчислити градієнт функції втрат (2) що покаже напрямок напрямок найшвидшого зростання функції втрат і перерахувати ваги мережі W враховуючи швидкість навчання α (3).

Вибір швидкості навчання α є вирішальним – якщо встановити занадто низьку, то нейронна мережа буде вчитися дуже повільно, якщо встановити занадто високою, то можна не досягти мінімуму. Для визначення швидкості навчання необхідно провести велику кількість експериментів на даних, що будуть використовуватися для навчання мережі, що може зайняти роки, тому пропонується використовувати адаптивну швидкість навчання для кожної ітерації.

Змінюючи ваги за формулою (3) ми знайдемо локальний мінімум функції втрат, проте не маємо можливості контролювати зміни ваг – робити більші «кроки» коли градієнт не змінюється на велике значення, щоб пришвидшити навчання, або ж навпаки менші щоб не «перестрибнути» мінімум.

$$m^{[l]} = \beta_1 m^{[l]-1} + (1 - \beta_1) g_t, \quad (4)$$

$$v^{[l]} = \beta_2 v^{[l]-1} + (1 - \beta_2) g_t^2. \quad (5)$$

Для цього при обчисленні градієнту, обчислемо значення ковзне середнє градієнту (4) та його квадратів (5).

$$W^{[l]} = W^{[l]-1} - \alpha \frac{m^{[l]}}{\sqrt{v^{[l]} + \epsilon}}. \quad (6)$$

Змінюючи ваги мережі ми контролюємо «крок» домножаючи зміну на швидкість навчання α . Таким чином, при малій зміні градієнту $m^{[l]}$ приблизно дорівнює $\sqrt{v^{[l]}}$, отже зміна ваг повністю контролюється швидкістю навчання. При різкій зміні градієнту $m^{[l]}$ значно менше ніж $\sqrt{v^{[l]}}$, а значить швидкість навчання зменшується.

При навчанні нейронних мереж важливо мати малі значення W , це дозволить уникнути перенавчання та створити більш узагальнену модель [6]. Існуючі методи вирішують цю проблему додавши доданок $\mu W^{[l]-1}$ у формулу розрахунку градієнту (2), проте це впливає на значення $m^{[l]}$ та $v^{[l]}$ – вони будуть відслідковувати не лише зміну градієнту але також і регуляцію ваг мережі.

$$W^{[t]} = W^{[t-1]} - \alpha \frac{\beta_1 m^{[t-1]} + (1 - \beta_1) [(g)]_t + \mu W^{[t-1]}}{\sqrt{v^{[t]} + \epsilon}} \quad (7)$$

Для вирішення цієї проблеми додамо регуляцію ваг як наведено в (7). В цьому випадку зниження ваги також нормується значенням $\sqrt{v^{[t]}}$: при великих значеннях ваг W або великому значенню градієнта (різко змінюється) значення $v^{[t]}$ також буде мати велике значення, тому великих змін ваг відбуватися не буде. Аналогічно для малих значень.

Для тестування продуктивності розробленого алгоритму в якості контрольної задачі візьмемо задачу навчання нейронної мережі для класифікації зображень будівель за архітектурним стилем. У вибірку для навчання увійшло 1033 зображень розподілених за 6 архітектурними стилями, у тестову – 504 зображення.

Розроблений оптимізатор буде порівняно з оптимізаторами Adagrad та RMSprop на архітектурах ResNet та DenseNet. Експерименти були проведені при однакових умовах: апаратне та програмне забезпечення не змінювалось, інших операцій паралельно не проводилось.

Для порівняння використаємо наступні критерії: загальний час навчання та зміна значень функції втрат. На рисунку 1 зображена зміна функції втрат при навчанні. Запропонований алгоритм показав кращий результат на архітектурах ResNet та DenseNet ніж Adagrad та RMSprop. В середньому алгоритм показав в 1.5 рази меншу функцію втрат при завершенні навчання.

На рисунку 2 зображена кількість часу затрачена для навчання при 20 епохах. Навчання з використанням запропонованого алгоритму в середньому в 1.13 раз швидше, ніж з використанням інших алгоритмів.

Висновки. Аналіз існуючих алгоритмів та методів показав недоліки що є критичними для вирішення задачі класифікації зображень. Запропонований алгоритм вирішує частину недоліків існуючих методів: він не потребує велику кількість обчислень, так як сума градієнтів рекурсивно визначається як середнє усіх минулих квадратних градієнтів; швидкість навчання є адаптивною що дозволяє пришвидшити навчання коли градієнт функції не швидко змінюється; алгоритм можна використовувати для обчислення групи спостережень, що зменшить вимоги до пам'яті та кількості обчислень; алгоритм підходить для глибоких мереж; метод регулює значення ваг нейронної мережі, не дозволяючи їм значно збільшуватися, що робить обчислення ефективними як у SGD для

неглибоких мереж; метод враховує вимоги до значень ваг нейронної мережі – вони мають бути малими для уникнення перенавчання.

Запропонований метод навчання нейронної мережі для класифікації архітектурних стилів будівель за зображеннями будівель за меншу кількість часу навчання досягає більшої точності. В середньому запропонований алгоритм показав в 1.5 рази менше значення функції втрат при завершенні навчання та в 1.13 раз швидше, ніж з використанням інших алгоритмів.

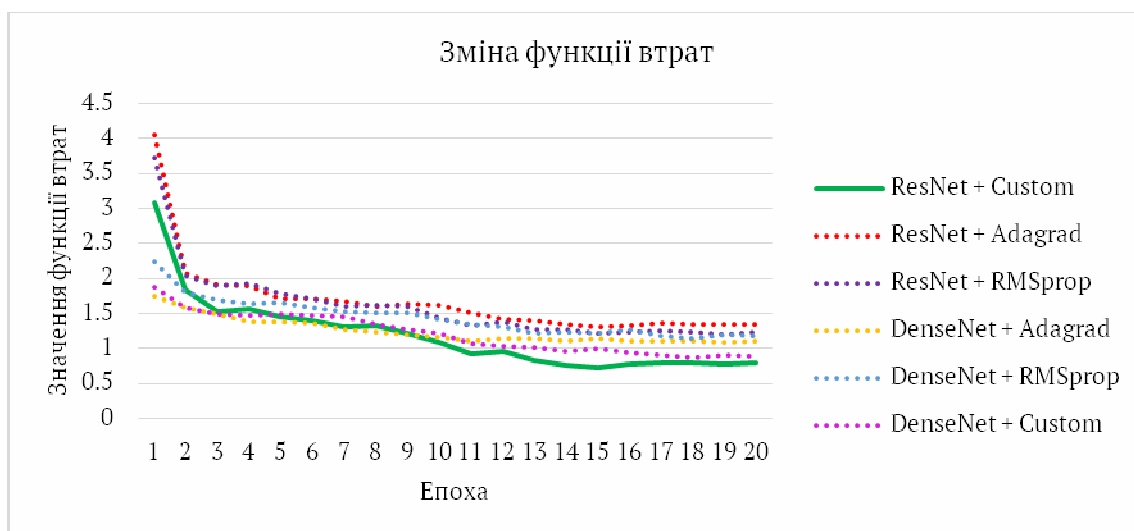


Рисунок 1 – Змінна значення функції втрат при навчанні мережі

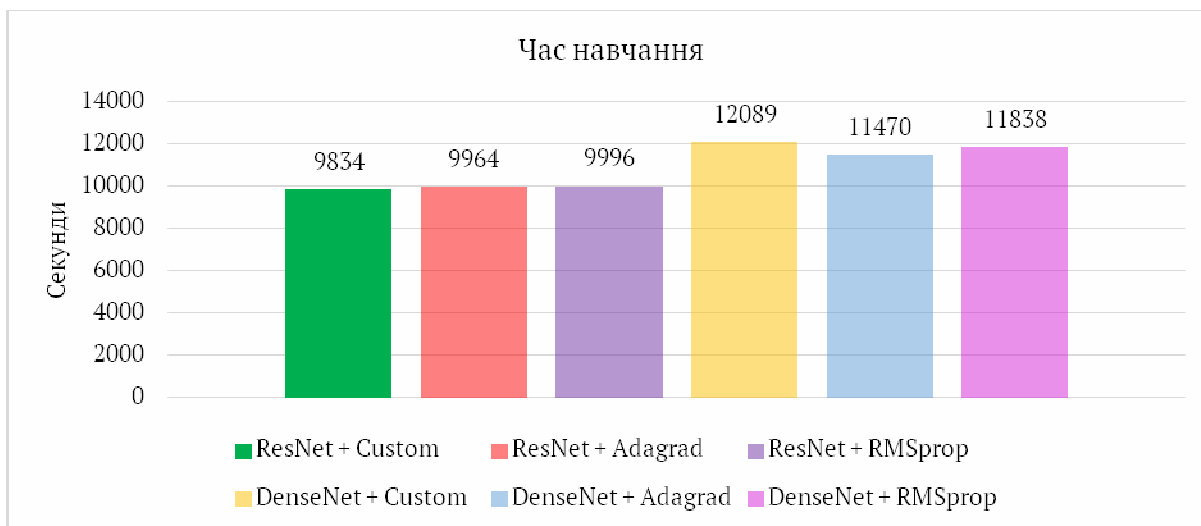


Рисунок 2 – Загальний час навчання мережі

ЛІТЕРАТУРА / LITERATURE

1. Chu W.T., Tsai M.H.: Visual pattern discovery for architecture image classification and product image search. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, p. 27. ACM (2012).
2. Z. Xu, D. Tao, Y. Zhang, J. Wu, A. Tsoi, "Architectural Style Classification using Multinomial Latent Logistic Regression", European Conference on Computer Vision (ECCV2014), 2014.
3. M. Mathias (a), A. Martinovic (a), J. Weissenberg (b), S. Haegler (b), L. Van Gool (a, b) "AUTOMATIC ARCHITECTURAL STYLE RECOGNITION" (a) PSI/VISICS, Department of Electrical Engineering, KU Leuven, Kasteelpark Arenberg 10/02441, 3001 Heverlee, Belgium - (markus.mathias, andelo.martinovic, luc.vangool)@esat.kuleuven.be (b) Computer Vision Laboratory, ETH Zurich, Sternwartstrasse 7, 8092 Zurich, Switzerland - (shaegler, julienw, vangool)@vision.ee.ethz.ch, 2011, Trento, Italy, 6 pages.
4. Gayane Shalunts, Yll Haxhimusa and Robert Sablatnig: «Architectural Style Classification of Building Facade Windows».-Vienna University of Technology, Institute of Computer Aided Automation Computer Vision Lab, Institute of Computer Graphics and Algorithms, Pattern Recognition and Image Processing Lab, 2011.
5. Shahar G. Neural Networks for Image Recognition: Methods, Best Practices, Applications [Електронний ресурс] – Режим доступу до ресурсу: <https://missinglink.ai/guides/computer-vision/neural-networks-image-recognition-methods-best-practices-applications/>.
6. Sorokina K. Image Classification with Convolutional Neural Networks [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8>.
7. Sebastian Ruder. An overview of gradient descent optimization algorithms [Електронний ресурс] Режим доступу до ресурсу: <https://ruder.io/optimizing-gradient-descent/>.
8. Vitaly Bushaev. Stochastic Gradient Descent with momentum [Електронний ресурс] - Режим доступу до ресурсу: <https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d>.
9. Helton Maia Peixoto, Rafael Magalhães. Artificial Neural Networks and Efficient Optimization Techniques for Applications in Engineering / Helton Maia Peixoto, Rafael Magalhães – 2011.

10. Ke Li, Jitendra Malik. Learning to Optimize Neural Nets – 2017.

REFERENCES

1. Chu W.T., Tsai M.H.: Visual pattern discovery for architecture image classification and product image search. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, p. 27. ACM (2012).
2. Z. Xu, D. Tao, Y. Zhang, J. Wu, A. Tsoi, "Architectural Style Classification using Multinomial Latent Logistic Regression", European Conference on Computer Vision (ECCV2014), 2014.
3. M. Mathias (a), A. Martinovic (a), J. Weissenberg (b), S. Haegler (b), L. Van Gool (a, b) "AUTOMATIC ARCHITECTURAL STYLE RECOGNITION" (a) PSI/VISICS, Department of Electrical Engineering, KU Leuven, Kasteelpark Arenberg 10/02441, 3001 Heverlee, Belgium - (markus.mathias, andelo.martinovic, luc.vangool)@esat.kuleuven.be (b) Computer Vision Laboratory, ETH Zurich, Sternwartstrasse 7, 8092 Zurich, Switzerland - (shaegler, julienw, vangool)@vision.ee.ethz.ch, 2011, Trento, Italy, 6 pages.
4. Gayane Shalunts , Yll Haxhimusa , and Robert Sablatnig: «Architectural Style Classification of Building Facade Windows». – Vienna University of Technology, Institute of Computer Aided Automation Computer Vision Lab, Institute of Computer Graphics and Algorithms, Pattern Recognition and Image Processing Lab, 2011.
5. Shahar G. Neural Networks for Image Recognition: Methods, Best Practices, Applications [Elektronniy resurs] — Rezhim dostupa k state: <https://missinglink.ai/guides/computer-vision/neural-networks-image-recognition-methods-best-practices-applications/>.
6. Sorokina K. Image Classification with Convolutional Neural Networks [Elektronniy resurs] — Rezhim dostupa k state: <https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8>.
7. Sebastian Ruder. An overview of gradient descent optimization algorithms [Elektronniy resurs] — Rezhim dostupa k state: <https://ruder.io/optimizing-gradient-descent/>.
8. Vitaly Bushaev. Stochastic Gradient Descent with momentum [Elektronniy resurs] — Rezhim dostupa k state: <https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d>.
9. Helton Maia Peixoto, Rafael Magalhães. Artificial Neural Networks and Efficient Optimization Techniques for Applications in Engineering / Helton Maia Peixoto, Rafael Magalhães – 2011.

**Исследование оптимизаторов нейронной сети для классификации изображений
зданий по архитектурному стилю**

В статье рассмотрена задача классификации изображений с тесными межклассовыми взаимосвязями - классификация архитектурных стилей зданий, где большое количество основных черт и признаков является общими для нескольких классов. Объектом исследования является алгоритм обучения нейронной сети. Предложен метод обучения нейронной сети для классификации архитектурных стилей зданий по изображения, который за меньшее количество времени обучения достигает большей точности. Предложенный алгоритм оптимизатора реализовано программно и проведено эксперименты для сравнения эффективности алгоритма.

Neural network optimizers for buildings architectural style classification task

The article considers the problem of classification of images with close interclass relationships - the classification of architectural styles of buildings, where a large number of basic features and characteristics are common to several classes. The object of the research is a neural network learning algorithm for recognizing architectural styles of buildings. A proposed method of neural network training achieves higher accuracy. The proposed optimizer algorithm was implemented and tested to compare the efficiency of the algorithm.

Analysis of existing algorithms and methods has shown shortcomings that are critical to solving the problem of image classification. The advanced algorithm solves some of the shortcomings of existing methods: it does not require a large number of calculations, as the sum of gradients is recursively defined as the average of all past square gradients; the learning speed is adaptive which allows to accelerate learning when the gradient of the function does not change quickly; the algorithm can be used to calculate a group of observations, which will reduce the requirements for memory and the number of calculations; the algorithm is suitable for deep networks; the method regulates the values of the weights of the neural network, not allowing them to increase significantly, which makes the calculations as effective as in SGD for shallow networks; the method takes into account the requirements for the values of the weights of the neural network - they must be small to avoid retraining.

The proposed method of learning the neural network for the classification of architectural styles of buildings achieves greater accuracy in less learning time. On average, the algorithm showed 1.5 times less value of the loss function at the end of training and 1.13 times faster than using other algorithms.

Гавриленко Олена Валеріївна – доцент кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут ім. І. Сікорського».

Новіченко Неля Валеріївна – студентка кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут ім. І. Сікорського».

Гавриленко Елена Валерьевна - доцент кафедры информатики и программной инженерии Национального технического университета Украины «Киевский политех-ночной институт им. И. Сикорского ».

Новиченко Неля Валерьевна - студентка кафедры информатики и программной инженерии Национального технического университета Украины «Киевский политех-ночной институт им. И. Сикорского ».

Gavrylenko Olena – Associate Professor of the Department of Informatics and Software Engineering of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».

Novichenko Nelia – student of the Department of Informatics and Software Engineering of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».