УДК 004.023+004.93
DOI: 10.31891/CSIT-2020-2-5

РАДЮК П. М.
Хмельницький національний університет

ПІДХІД ДО ПРИСКОРЕННЯ НАВЧАННЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ЗА РАХУНОК НАЛАШТУВАННЯ ГІПЕРПАРАМЕТРІВ НАВЧАННЯ

За останні десятиліття впровадження методів глибокого навчання, зокрема згорткових нейронних мереж (ЗНМ) призвело до вражаючого успіху у задачах обробки статичних зображень та відео. Проте, навчання ЗНМ здебільшого ґрунтується на застосуванні наборів квазіоптимальних гіперпараметрів архітектури та навчання. Подібний підхід потребує тривалого часу навчання мережі та не гарантує задовільного результату. Тим не менш, налаштування гіперпараметрів має вирішальне значення для ефективності ЗНМ, оскільки різні гіперпараметри призводять до моделей із суттєво різними характеристиками. Невдало підібрані гіперпараметри зазвичай призводять до низької продуктивності моделі. На сьогодні, питання оптимального підбору гіперпараметрів для ЗНМ все ще невирішене. Подана робота пропонує кілька практичних підходів до налаштування гіперпараметрів, що дає змогу скоротити час навчання та підвищити точність роботи моделі. У статті розглядається функція втрат валідації тренувань під час недо- та перенавчання та наводяться вказівки щодо досягнення точки оптимізації. В роботі також розглядається питання регуляції кроку та імпульсу навчання для прискорення навчання мережі. Усі експерименти базуються на відомих наборах даних CIFAR-10 та CIFAR-100.

Keywords: швидкість навчання, розмір підвиборки набору даних, імпульс навчання, зниження ваги, гіперпараметри, згорткова нейронна мережа, точність валідації.

RADIUK P.
Khmelnytskyi National University, Ukraine

AN APPROACH TO ACCELERATE THE TRAINING OF CONVOLUTIONAL NEURAL NETWORKS BY TUNING THE HYPERPARAMETERS OF LEARNING

Over the last decade, a set of machine learning algorithms called deep learning has led to significant improvements in computer vision, natural language recognition and processing. This has led to the widespread use of a variety of commercial, learning-based products in various fields of human activity. Despite this success, the use of deep neural networks remains a black box. Today, the process of setting hyperparameters and designing a network architecture requires experience and a lot of trial and error and is based more on chance than on a scientific approach. At the same time, the task of simplifying deep learning is extremely urgent.

To date, no simple ways have been invented to establish the optimal values of learning hyperparameters, namely learning speed, sample size, data set, learning pulse, and weight loss. Grid search and random search of hyperparameter space are extremely resource intensive. The choice of hyperparameters is critical for the training time and the final result. In addition, experts often choose one of the standard architectures (for example, ResNets and ready-made sets of hyperparameters. However, such kits are usually suboptimal for specific practical tasks.

The presented work offers an approach to finding the optimal set of hyperparameters of learning ZNM. An integrated approach to all hyperparameters is valuable because there is an interdependence between them. The aim of the work is to develop an approach for setting a set of hyperparameters, which will reduce the time spent during the design of ZNM and ensure the efficiency of its work.

In recent decades, the introduction of deep learning methods, in particular convolutional neural networks (CNNs), has led to impressive success in image and video processing. However, the training of CNN has been commonly mostly based on the employment of quasi-optimal hyperparameters. Such an approach usually requires huge computational and time costs to train the network and does not guarantee a satisfactory result. However, hyperparameters play a crucial role in the effectiveness of CNN, as diverse hyperparameters lead to models with significantly different characteristics. Poorly selected hyperparameters generally lead to low model performance. The issue of choosing optimal hyperparameters for CNN has not been resolved yet. The presented work proposes several practical approaches to setting hyperparameters, which allows reducing training time and increasing the accuracy of the model. The article considers the function of training validation loss during underfitting and overfitting. There are guidelines in the end to reach the optimization point. The paper also considers the regulation of learning rate and momentum to accelerate network training. All experiments are based on the widespread CIFAR-10 and CIFAR-100 datasets.

Keywords: Learning rate; batch size; momentum; weight decay; hyperparameters; convolutional neural network; validation accuracy.

Вступ. За останнє десятиліття, набір алгоритмів машинного навчання під назвою глибоке навчання (deep learning) призвів до значних поліпшень у задачах комп'ютерного зору [10], розпізнавання та обробки природних мов [2, 3]. Це призвело до широкого застосування різноманітних комерційних продуктів, що засновані на навчанні, в різних сферах людської діяльності. Незважаючи на такий успіх, застосування глибоких нейронних мереж залишається чорним ящиком. На сьогодні процес налаштування гіперпараметрів та проектування архітектури мережі вимагає досвіду та значної кількості спроб та помилок і базується більше на випадковості, ніж на науковому підході. В той же час, завдання спрощення глибокого навчання є надзвичайно акуальними.

На сьогодні не винайдено простих способів встановлення оптимальних значеннь гіперпараметрів навчання, а саме швидкості навчання, розміру підвиборок наборві даних, імпульсу навчання та зниження ваги. Пошук за сіткою (grid search) [4] та випадковий пошук (random search) [5] простору гіперпараметрів є надзвичайно ресурсоємкими. Вибір гіперпараметрів є критичним для часу навчання та фінального результату. Крім того, фахівці часто вибирають одну зі стандартних архітектур (наприклад, ResNets [6]) та готові набори гіперпараметрів. Проте, зазвичай такі набори є неоптимальними для конкретних практичних завдань.

Подана робота пропонує підхід до пошуку оптимального набору гіперпараметрів навчання ЗНМ. Комплексний підхід до всіх гіперпараметрів є цінним, оскільки між ними є взаємозалежність. Метою роботи є розроблення підходу для налаштування набору гіперпараметрів, що дасть змогу знизити витрати часу під час проектування ЗНМ та забезпечить ефективність її роботи.

Аналіз досліджень та публікацій. Основою для поданого підходу є добре відома концепція балансу між недонавчанням та перенавчанням. Вона полягає у оцінюванні тестової та валідаційної функції втрат навчання за наявності недонавчання та перенавчання, для досягнення оптимального набору гіперпараметрів. Як тестові, так і валідаційні втрати стосуються використання даних валідації для виявлення помилки або точності, яку створює мережа під час навчання (рис. 1) [4].

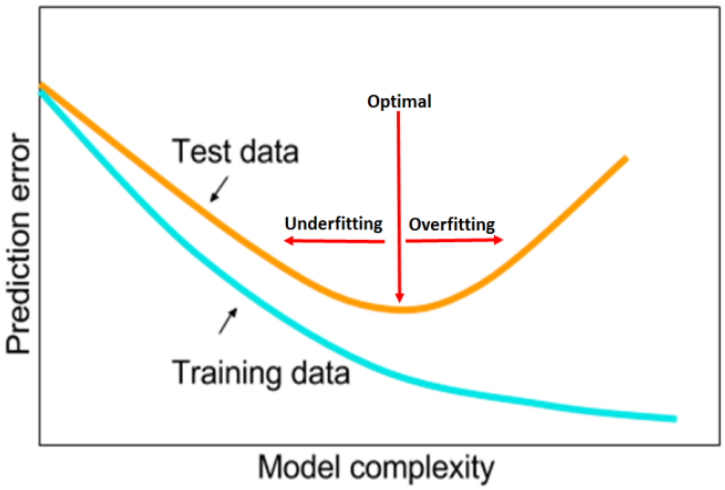


Рис. 1. Компроміс між недо- та перенавчанням. Складність моделі (вісь абсцис) стосується потужності моделі машинного навчання (зображено оптимальну потужність, що знаходиться між недо- та перенавчанням)

У роботі [6] пропонується розглянути вимоги до налаштування гіперпараметрів, використовуючи циклічні темпи навчання та циклічний імпульс. Наведені експериментальні результати вказують на те, що швидкість навчання, імпульс і регуляризація тісно пов'язані між собою, і оптимальні значення повинні визначатися разом.

У кількох останніх роботах обговорюється використання великої швидкості навчання [7], малого [8] та великого [9, 10] розміру підвиборок наборів навчальних даних. Вони демонструють, що співвідношення швидкості навчання до розміру підвиборки набору даних впливає на навчання. У роботі [11] автори досліджують розміри підвиборки та співвідносять оптимальний розмір підвиборки зі швидкістю навчання, розміром набору даних та імпульсом.

Нещодавні роботи [12, 13] ставлять під сумнів використання регуляризації за рахунок гіперпараметра dropout та зниження ваги. Одним із висновків цієї роботи є те, що загальна регуляризація повинна бути в рівновазі для певного набору даних та архітектури. Проведені експерименти демонструють, що їхній погляд на регуляризацію обмежений – вони лише додають регуляризацію шляхом збільшення даних, щоб замінити регуляризацію зменшенням ваги та випаданням без повного вивчення регуляризації. Існують також підходи до вивчення оптимальних гіперпараметрів шляхом диференціації градієнта щодо гіперпараметрів [14]. Підхід у цій роботі є більш простим для практики.

Мета та завдання дослідження.

Ця робота спрямована на визначення найкращого розміру підвиборки набору даних для швидкого навчання та точності розпізнавання ЗНМ. Для досягнення мети потрібно виконати наступні задачі.

1. Підготувати різні набори даних для навчання нейронної мережі.
2. Обрати гіперпараметри навчання мережі та визначити послідовність їхніх значень для використання у навчанні ЗНМ.
3. Дослідити, як значення гіперпараметрів впливає на точність розпізнавання зображень обраного набору даних.
4. Визначити найкращі бали точності навчання та запропонувати чинники або групу чинників, що об'єднують ці бали, вказуючи значення відповідних гіперпараметрів.

Тестові набори даних. Задача класифікації зображень проводиться на наборах даних CIFAR-10 та CIFAR-100 [15]. Ці набори широко застосовуються для оцінки різних архітектур ЗНМ завдяки простому використанню та задовільним результатам у порівнянні.

База даних CIFAR-10 складається з 60 000 кольорових зображень розміром 32 × 32 у 10 класах, по 6000 зображень на категорію. Є 50 000 навчальних зображень та 10 000 пробних зображень. Набір даних розділений на п'ять навчальних підвиборок та одну тестову. Тестова підвибірка містить рівно 1000 довільно вибраних зображень з кожного класу. Навчальні виборки містять решту зображень у довільному порядку. Проте деякі виборки набору даних можуть містити більше зображень одного класу, ніж інші. Між ними навчальні виборки містять рівно 5000 зображень з кожної категорії (рис. 2).

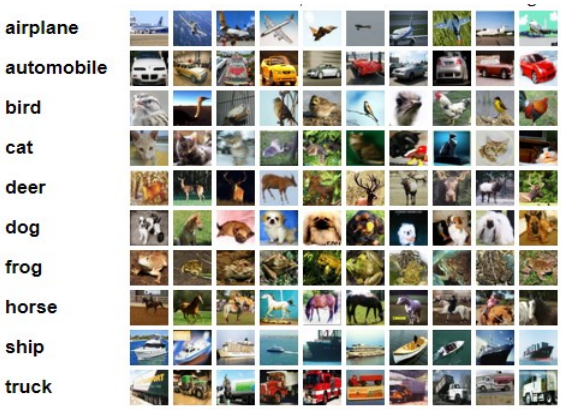


Рис. 2. Різноманітність кольорових зображень із набору даних CIFAR-10, що містить десять категорій зображень (позначені як «літак», «автомобіль», «птах», «кішка», «олень», «собака», «жаба», «кінь», «корабель», «вантажівка») [15]

Ініціалізація наборів гіперпараметрів навчання. Постановку завдання можна розглядати як пошук наборів гіперпараметрів за заданим набором даних та архітектурою.

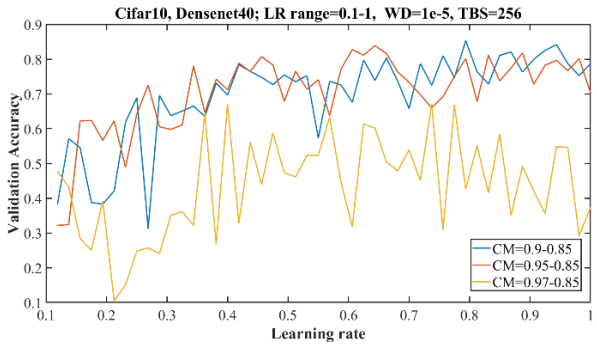
1. ***Швидкість навчання*** (learning rate, LR). Перш за все, потрібно виконати перевірку діапазону швидкості навчання на «велику» швидкість навчання. Максимальний LR залежить від архітектури (для дрібної 3-шарової архітектури велика – 0.01, а для ResNet велика – 3.0).

2. ***Загальний розмір виборки набору даних*** (total batch size, TBS). Якщо обчислювальний пристрій має кілька графічних процесорів, загальний розмір виборки – це розмір виборки набору даних на графічному процесорі, помножений на кількість графічних процесорів.

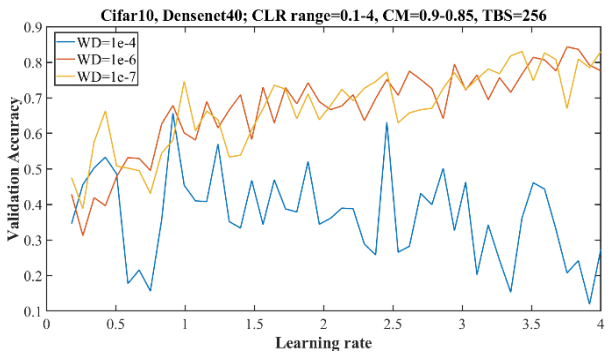
3. ***Імпульс навчання*** (momentum, M). Короткі пробіги зі значеннями імпульсу навчання 0.99, 0.97, 0.95 та 0.9 швидко покажуть найкраще значення для імпульсу. Використання циклічного імпульсу разом із тестом діапазону LR стабілізує збіжність під час використання великих значень швидкості навчання більше, ніж це робить постійний імпульс.

4. ***Зниження ваги навчання*** (weight decay, WD). Цей гіперпараметр потребує визначення належної величини. Наприклад, більш складний набір даних вимагає меншої регуляризації, тому спробуємо менші значення зменшення ваги, такі як 10^{-4}, 10^{-5}, 10^{-6}, 0. Неглибока архітектура вимагає більшої регуляризації, тому перевіримо великі значення зменшення ваги, такі як 10^{-2}, 10^{-3}, 10^{-4}.

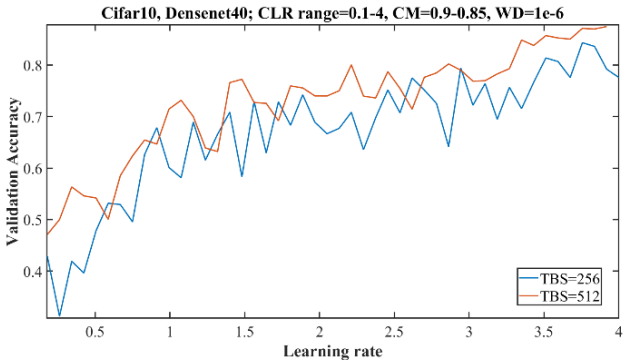
а) порівняння WD з 32-шаровим ResNet



б) порівняння імпульсу з 40-шаровим DenseNet



в) порівняння WD з 40-шаровим DenseNet



г) порівняння TBS з 40-шаровим DenseNet

Рис. 3. Ілюстрація пошуку гіперпараметрів для широких ResNet та DenseNet на CIFAR-10. Тренінг слідує випробуванню діапазону швидкості навчання (LR = 10^{-1}) для WideNet та (LR = 10^{-4} для DenseNet, а також cycling momentum (0.95–0.85). Для архітектури DenseNet точність тестування зручніше інтерпретувати для найкращого weight decay, ніж тестова втрата

Цей список узагальнює методи для оптимізації гіперпараметрів. Оптимізація гіперпараметрів може бути досить швидкою, якщо шукати підказки про втрату тесту на початку тренування. Нижче наводиться опис використання зазначеного контрольного списку для кількох архітектур на наборах даних CIFAR-10 та CIFAR-100.

ResNets на наборі даних CIFAR-10. Першим кроком для тестування гіперпараметрів за допомогою комплексної архітектури ResNet є вихід тесту діапазону LR із зменшенням імпульсу навчання в процесі кількох значень зниження ваги. Для ResNet початкові значення гіперпараметрів є такими: TBS = 512, швидкість навчання від 0.1 до 1.0 і імпульс від 0.95 до 0.85.

Рис. 3, *а* ілюструє пошук за сіткою гіперпараметра *WD* з трьома прогонами, кожен з різними значеннями *WD* 10^{-3}, 10^{-4} та 10^{-5}. На рис. 3 видно, що 10^{-3} дає низьку продуктивність, а 10^{-4} трохи краще, ніж 10^{-5}. Четверте випробування із зменшенням ваги, встановленим на $3\cdot 10^{-5}$, аналогічно як 10^{-4}, так і 10^{-5}, вказуючи, що будь-яке значення в цьому діапазоні демонструє хороші результати.

У таблиці 1 наведено кінцевий результат тренінгу з виявленими гіперпараметрами, використовуючи графік швидкості навчання за один прогін з межами швидкості навчання від 0.1 до 1.0.

Таблиця 1

Фінальна точність та стандартне відхилення для різних наборів даних та архітектур; загальний розмір виборки (TBS) для всіх експериментів становив 512

Dataset	Architecture	CLR	CM	WD	Epochs	Accuracy, %
CIFAR-10	ResNet	0.1	0.9	10^{-4}	100	85.6 ± 0.7
CIFAR-10	ResNet	0.1	0.9	10^{-4}	200	86.9 ± 0.6
CIFAR-10	ResNet	0.1	0.9	10^{-4}	800	90.1 ± 0.8
CIFAR-10	ResNet	0.1-0.5	0.95-0.85	10^{-4}	25	84.3 ± 0.6
CIFAR-10	ResNet	0.1-1	0.95-0.85	10^{-4}	50	89.5 ± 0.2
CIFAR-10	ResNet	0.1-1	0.95-0.85	10^{-4}	100	88.7 ± 0.2
CIFAR-10	DenseNet	0.1	0.9	10^{-4}	100	92.4 ± 0.1
CIFAR-10	DenseNet	0.1	0.9	10^{-4}	200	91.7 ± 0.2
CIFAR-10	DenseNet	0.1	0.9	10^{-4}	400	92.1 ± 0.3
CIFAR-10	DenseNet	0.1-4	0.95-0.85	10^{-4}	75	89.8 ± 0.7
CIFAR-10	DenseNet	0.1-4	0.95-0.85	10^{-4}	100	91.5 ± 0.3
CIFAR-10	DenseNet	0.1-4	0.95-0.85	10^{-4}	150	90.8 ± 0.1
CIFAR-100	ResNet-56	0.005	0.9	10^{-4}	100	60.2 ± 0.5
CIFAR-100	ResNet-56	0.005	0.9	10^{-4}	200	60.3 ± 0.8
CIFAR-100	ResNet-56	0.005	0.9	10^{-4}	400	61.0 ± 0.3
CIFAR-100	ResNet-56	0.1-0.5	0.95-0.85	10^{-4}	25	61.4 ± 0.2
CIFAR-100	ResNet-56	0.1-0.5	0.95-0.85	10^{-4}	50	64.6 ± 0.8
CIFAR-100	ResNet-56	0.09-0.9	0.95-0.85	10^{-4}	100	66.3 ± 0.4

За 100 ітерацій мережа ResNet-32 сходиться і забезпечує тестову точність $88.7\,\%\pm0.2$. Для порівняння, стандартний метод навчання досягає точності лише 90.1 ± 0.8 за 800 ітерацій. Подібний результат демонструє суперконвергенцію для ResNets.

DenseNets на наборі даних CIFAR-10.

Такий же експеримент було проведено над архітектурою DenseNet з 40 шарами. Проте пошук гіперпараметрів для DenseNet є більш складним завданням, ніж для ResNets. Першим кроком для тестування гіперпараметрів за допомогою архітектури DenseNet є запуск тесту діапазону LR з кількома максимальними значеннями імпульсу. Імпульс, встановлений на 0.99, але значення 0.97, 0.95 та 0.9 показані на рис. 3, *б*. Хоча, як правило, легше інтерпретувати тестові втрати для пошуку найкращих значень гіперпараметрів, для архітектури DenseNet точність тестування легше інтерпретувати, ніж тестові втрати. Для архітектури DenseNet менші значення імпульсу працюють краще, тому в наступних тестах використовувався діапазон від 0.9 до 0.85. Далі було перевірено діапазон швидкості навчання. Як показано на рис. 3.в), архітектура DenseNet стабільна навіть під час діапазону швидкості навчання від 0.1 до 4.0.

Використовуючи *TBS* = 256 (обрано відповідно до роботи [9]), комбіновану швидкість навчання та імпульс (0.95–0.85), три прогони з різними значеннями занепаду ваги 10^{-3}, 10^{-4} та 10^{-5} знову показує, що 10^{-3} дає низьку продуктивність, а 10^{-5} – найкращу. Отже, були випробувані менші значення занепаду ваги, а результати показані на рис. 3, *в*). На цьому рисунку видно, що занепад ваги на 10^{-4} дає більше

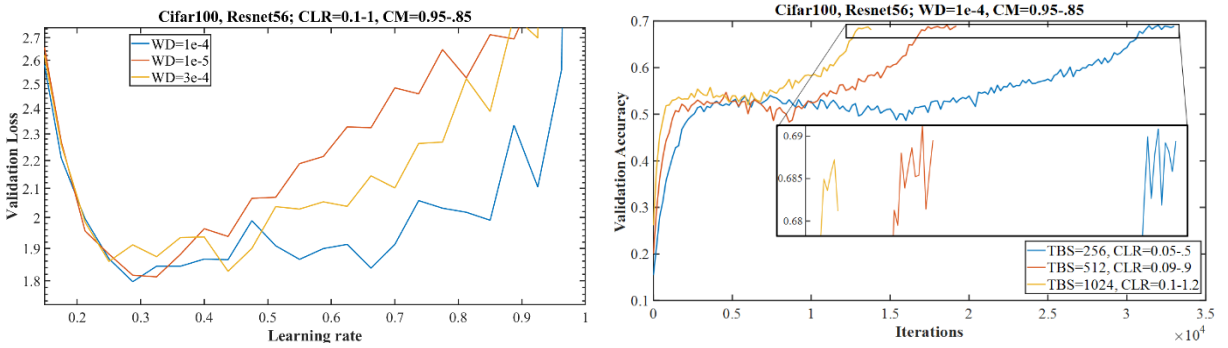
недостатньої точності, ніж менші значення завади ваги. Крім того, стохастичність кривої означає, що складність архітектури додає регуляризації, тому зменшення спаду ваги має інтуїтивний сенс. Рис. 3, *в* показує, що значення занепаду ваги 10^{-6} здається приблизно правильним.

Наступний експеримент проведений щодо TBS (рис. 3, *г*). На графіку порівнюється TBS від 256 до 512. Цей рисунок вказує на те, що TBS розміром 512 працює краще, ніж 256. Більший розмір виборки також зменшує регуляризацію, що відображається на трохи менш шумній кривій.

У таблиці 2 наведено кінцеві результати точності навчання з виявленими гіперпараметрами, використовуючи один цикл LR з межами швидкості навчання від 0.1 до 4.0 та циклічним імпульсом в діапазоні від 0.9 до 0.85.

ResNets на наборі даних CIFAR-100.

Ці експерименти використовували ту саму архітектуру ResNet-56, що і для CIFAR-10. Для CIFAR-10 гіперпараметрами з найкращими значеннями стали: LR зі значенням 0.1-1, TBS 512, циклічний M 0.95–0.85 та WD 10^{-4}. На рис. 4, *а* показані валідаційні втрати під час зниження ваги для значень $3\cdot 10^{-4}$, 10^{-4} та 10^{-5}. Найкраще значення для зниження ваги становить 10^{-4}, оскільки як більші, так і менші значення призводять до більших втрат.



***а*) пошук зниження ваги** **б) пошук загального розміру вибірки набору даних**

Рис. 4. Пошук гіперпараметрів для набору даних CIFAR-100 з архітектурою ResNet-56

Рис. 4, *б* порівнює криві навчання точності для трьох розмірів вибірки, а саме 256, 512, 1024. Кількість ітерацій тренувань було скориговано, щоб забезпечити подібний час виконання. У цьому випадку точність знаходиться в межах стандартних відхилень одна від одної.

Таблиця 1 порівнює остаточну точність навчання із політикою покрокового навчання та навчання із політикою швидкості навчання на 1 цикл. Результати навчання за політикою швидкості навчання на 1 цикл значно вищі, ніж результати політики щодо ступеня навчання. Крім того, кількість епох, необхідних для навчання, зменшується на порядок (тобто, навіть лише за 25 епох, точність вища за 1 цикл, ніж 800 ітерацій із політикою поступового навчання).

Висновки. У роботі досліджується вплив набору гіперпараметрів навчання глибоких нейронних мереж на точність класифікації зображень. В результаті проведених есперименів, запропоновано декілька ефективних способів налаштування гіперпараметрів, що значно підвищує ефективність роботи ЗНМ. Зокрема, робота демонструє як оцінити валідаційну та тестову функцію втрат, щоб уникнути недонавчання та перенавчання. У роботі наведено декілька кроків, послідовне виконання яких може привести до точки оптимального балансу, та пояснює, як можна збільшити або зменшити learning rate та momentum для прискорення навчання. Проведені експерименти підтверджують важливість балансування усіх способів регуляризації для кожного набору даних та архітектури. Гіперпараметр зниження ваги використовується як зразковий регулятор, щоб показати, як його оптимальне значення тісно пов'язане з швидкістю та імпульсом навчання.

Подальші дослідження будуть присвячені вивченню таких елементів навчання ЗНМ, як ефекти даних, збільшення даних, вплив глибини та ширини мережі, а також інші форми регуляризації.

Література

1. Liu, L., Ouyang, W., Wang, X., et al. Deep learning for generic object detection: A survey. International Journal of Computer Vision. 2020. Vol. 128, No. 2. P. 261–318. https://doi.org/10.1007/s11263-019-01247-4

2. Nassif, A. B., Shahin, I., Attili, I., et al. Speech recognition using deep neural networks: A systematic review. IEEE Access. 2019. Vol. 7. P. 19143–19165. https://doi.org/10.1109/ACCESS.2019.2896880

3. Otter, D. W., Medina, J. R., Kalita, J. K. A survey of the usages of deep learning for natural language processing. IEEE Transactions on Neural Networks and Learning Systems. 2020. P. 1–21. https://doi.org/10.1109/TNNLS.2020.2979670

4. Shekar, B. H., Dagnew, G. Grid search-based hyperparameter tuning and classification of microarray cancer data: Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP-2019), 2019. P. 1–8. https://doi.org/10.1109/ICACCP.2019.8882943

5. Torres, J. F., Gutiérrez-Avilés, D., Troncoso, A., et al. Random hyper-parameter search-based deep neural network for power consumption forecasting: Advances in Computational Intelligence, Cham , Springer International Publishing, 19. P. 259–269. https://doi.org/10.1007/978-3-030-20521-8_22

6. Szegedy, C., Ioffe, S., Vanhoucke, V., et al. Inception-v4, Inception-ResNet and the impact of residual connections on learning: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 4–10 February 2017 , AAAI Press, 17. P. 4278–4284. https://doi.org/10.5555/3298023.3298188

7. Smith, L. N. Cyclical learning rates for training neural networks: Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV-2017), 03.June.17. P. 464–472. https://doi.org/10.1109/WACV.2017.58

8. Smith, L. N., Topin, N. Super-convergence: very fast training of neural networks using large learning rates: Proc.SPIE, 10.May.19. https://doi.org/10.1117/12.2520589

9. Radiuk, P. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. Information Technology and Management Science. 2017. Vol. 20, No. 1. P. 20–24. https://doi.org/10.1515/itms-2017-0003

10. Hu, Z., Xiao, J., Tian, Z., et al. A variable batch size strategy for large scale distributed DNN training: 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 19. P. 476–485. https://doi.org/10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00074

11. Lounici, K., Meziani, K., Riu, B. Optimizing generalization on the train set: A novel gradient-based framework to train parameters and hyperparameters simultaneously // arXiv:1901.08644 [cs.NE]. 2020.

12. Garbin, C., Zhu, X., Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. Multimedia Tools and Applications. 2020. Vol. 79, No. 19. P. 12777–12815. https://doi.org/10.1007/s11042-019-08453-9

13. Golatkar, A. S., Achille, A., Soatto, S. Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence: Advances in Neural Information Processing Systems 32 (NIPS-2019), 2019. P. 10678–10688.

14. Zhang, H., Sun, J., Xu, Z. On hyper-parameter tuning for stochastic optimization algorithms // arXiv:1901.08644 [cs.NE]. 2020.

15. Krizhevsky, A. Learning multiple layers of features from tiny images. Master's Thesis, The University of Toronto, 2009. 60 p.

Надійшла / Paper received: 14.09.2020
Надрукована / Paper Printed : 03.11.2020