

УДК 004.6, 004.8

ВИЗНАЧЕННЯ НАЙКРАЩОГО ОПТИМІЗАТОРА ДЛЯ НЕЙРОМЕРЕЖІ ПРИ РОЗРОБЦІ СИСТЕМ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

Андріан Козинець^{ORCID}

Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 м. Львів, Україна

Козинець, А. (2025). Визначення найкращого оптимізатора для нейромережі при розробці систем автоматичної класифікації зображень. *Електроніка та інформаційні технології*, 29, 37-46. <https://doi.org/10.30970/eli.29.4>

АНОТАЦІЯ

Передумови. Вибір найкращого оптимізатора є важливим кроком у розробці ефективних систем автоматичної класифікації зображень. Зокрема, для нейромереж, побудованих на основі згорткових нейронних мереж (ЗНМ), вибір між популярними методами оптимізації, такими як Adam (Адаптивна Оцінка Моменту) та SGD (Стохастичний Градієнтний Спуск), може суттєво вплинути на їхню ефективність. Порівняння цих оптимізаторів дає змогу визначити найбільш відповідний оптимізатор для розв'язання конкретних завдань машинного навчання.

Методи та інструменти. Для навчання та тестування моделей було використано бібліотеку TensorFlow та інтерфейс для візуалізації процесу машинного навчання TensorBoard. Нейромережі тренувались на двох різних наборах даних з суттєво різними характеристиками. Для кожного набору даних було проведено навчання за допомогою двох оптимізаторів: Adam та SGD. Під час аналізу результатів враховувалися такі параметри, як точність тренування та перевірки, а також втрати під час тренування та перевірки.

Результати та обговорення. Згідно з результатами, оптимізатор SGD показав кращі показники продуктивності порівняно з Adam. Початкова точність для SGD була вищою, а також спостерігалось значне підвищення швидкості зростання точності. Крім того, остаточна точність моделі з оптимізатором SGD була вищою як на тренувальному, так і на перевірному наборах даних, що вказує на більш ефективний процес навчання з використанням цього оптимізатора.

Висновки. Визначено, що використання оптимізатора SGD призводить до кращих результатів у порівнянні з оптимізатором Adam для задач автоматичної класифікації зображень. Це підтверджується вищими показниками точності та більш швидким процесом навчання.

Ключові слова: TensorFlow, згорткова нейронна мережа, Adam, адаптивна оцінка моменту, SGD, стохастичний градієнтний спуск, GPU, TPU.

ВСТУП

У сучасному цифровому світі обробка та класифікація зображень стали необхідними складовими багатьох додатків та сервісів. З розвитком технологій машинного навчання з'являються нові можливості для автоматизації цих процесів, що значно підвищує ефективність та точність роботи з великими обсягами візуальних даних. Одним із перспективних напрямків є розробка систем автоматичної класифікації зображень, які дозволяють користувачам швидко отримувати інформацію про зміст зображень та полегшують їх пошук і категоризацію.



© 2025 Andrian Kozynets. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Класифікація зображень є процесом асоціювання ключових слів або міток з зображеннями для того, щоб описати їхній зміст. Класифікація має велике значення у різних сферах та застосуваннях, оскільки допомагає в управлінні, пошуку та аналізі зображень. Наприклад, класифікація зображення використовується у соціальних мережах, системах керування вмістом (СКВ), пошукових системах. Під час навчання моделей машинного навчання класифіковані зображення використовуються для покращення алгоритмів розпізнавання об'єктів, сцен та інших характеристик зображень. Це дозволяє створювати більш точні та ефективні системи розпізнавання.

Вибір оптимізатора під час навчання ЗНМ для класифікації зображень є надзвичайно важливим етапом, який безпосередньо впливає на час навчання, точність отриманої моделі та розбіжність між фактичними та передбаченими результатами. В статті [1] розглянуто причину через яку SGD погано працює з проблемою блочної неоднорідності, а також показано чому адаптивна оцінка моменту (Adam) показує кращі результати в цій ситуації. У [2] розглянуто гібридне використання цих двох оптимізаторів, що призводить до отримання кращих результатів навчання.

При виборі оптимізатора для навчання ЗНМ важливо враховувати специфіку задачі та характеристики даних, а також особливості архітектури самої мережі. Однією з сучасних тенденцій є використання методів мета-оптимізації, які автоматично налаштовують параметри навчання. Це значно спрощує процес пошуку найкращих гіперпараметрів і забезпечує стабільнішу та швидшу збіжність. Наприклад, байесова оптимізація може ефективно знаходити оптимальні параметри, комбінувати різні підходи до навчання.

Методи, які комбінують SGD з адаптивними методами, такими як Adam, стають все популярнішими. Такі гібридні оптимізатори можуть знаходити більш ефективні траєкторії мінімізації функції втрат та забезпечувати баланс між швидкістю збіжності та стабільністю. Використання таких оптимізаторів знижує ризик застрягання в локальних мінімумах і покращує узагальнення на нових даних.

Окрім вибору оптимізатора, необхідно враховувати особливості апаратного забезпечення. Наприклад, сучасні графічні процесори (ГП) та тензорні процесори (ТП) значно прискорюють процес навчання завдяки своїй архітектурі, яка підходить для паралельної обробки даних. Оптимізатори, такі як Adam, добре працюють з великими наборами даних і можуть ефективно використовувати обчислювальні ресурси ГП/ТП.

Використання гібридних оптимізаторів і методів мета-оптимізації вимагає системного підходу до налаштування та тестування моделі. Це дозволяє максимально ефективно використовувати доступні обчислювальні ресурси, досягати високої точності та узагальнюваності моделей на нових даних.

МЕТОДИ ТА ІНСТРУМЕНТИ

Для розробки системи автоматичної класифікації зображень використано бібліотеку TensorFlow та два оптимізатори: Adam і SGD.

Вибір бібліотеки TensorFlow зумовлений її широким застосуванням у дослідженнях та промислових проектах, а також наявністю великої кількості інструментів і ресурсів для розробки та оптимізації моделей машинного навчання. TensorFlow надає гнучкість у налаштуванні моделей та підтримує роботу на різних платформах, що робить її зручною для використання в різних середовищах.

Оптимізатор Adam обрано через його здатність адаптивно змінювати швидкість навчання для кожного параметра окремо, що прискорює збіжність моделі та покращує її стабільність. Adam добре підходить для задач, де дані мають високу варіативність.

Оптимізатор SGD, з іншого боку, є класичним методом оптимізації, який часто використовується через свою простоту та ефективність у задачах з великими наборами даних. Він менш вибагливий до ресурсів і може забезпечити хороші результати при правильному налаштуванні швидкості навчання. Проте, порівняно з Adam, SGD може демонструвати повільнішу збіжність і більшу чутливість до вибору початкових гіперпараметрів.

У порівнянні з іншими популярними оптимізаторами, такими як RMSprop або Adagrad, вибір Adam і SGD обґрунтований їхньою здатністю забезпечити хороший баланс між швидкістю навчання та стабільністю моделі. RMSprop, наприклад, теж є адаптивним оптимізатором, але Adam об'єднує в собі переваги RMSprop та моменту, що дає йому перевагу в багатьох сценаріях.

Оптимізатори в 3НМ відіграють ключову роль у процесі навчання. Вони допомагають мережі адаптуватися до набору даних та підвищувати ефективність у вирішенні поставленої задачі. Два найпоширеніші оптимізатори: Adam та SGD – мають свої унікальні особливості та застосування.

Оптимізатор Adam [3] використовує метод ефективної стохастичної оптимізації, який вимагає лише градієнтів першого порядку з невеликими вимогами до пам'яті. Метод розраховує індивідуальні адаптивні темпи навчання для різних параметрів з оцінок першого та другого моментів градієнтів.

SGD — це оптимізаційний алгоритм, який лежить в основі багатьох моделей глибокого навчання (ГН). SGD працює шляхом поступового коригування параметрів моделі з використанням градієнта функції втрат, який обчислюється для підмножини (міні-пакета) даних навчання, а потім змінюючи міні-партію протягом певного часу, щоб вивчити весь набір даних [4].

Для дослідження ефективності оптимізаторів послідовно використано два набори даних. Перший набір даних містив 25 тисяч зображень розділених на 6 категорій. Другий набір даних містив 550 тисяч зображень, розділених на тисячу класів. Кожне зображення у наборах було попередньо підготовлено для використання.

Дослідження проводилося на п'ятишаровій нейронній мережі, яка включала три згорткових шари з ядром розміром 3x3 та попередньою активацією за допомогою функції активації ReLU (Rectified Linear Unit, Випрямлений Лінійний Вузол). Кожний згортковий шар налаштовувався для виявлення різних ознак та характеристик у вхідних зображеннях. Після згорткових шарів використовувався шар згладжування для згладжування вихідних даних перед їх подачею на повнозв'язний шар. Останній, п'ятий, шар мережі був повнозв'язний та використовувався для класифікації зображень на тисячу різних класів, що представляли собою категорії зображень. Темп навчання для обох оптимізаторів встановлено на однаковий рівень, який становив 10^{-4} . Однаковий темп забезпечує порівнянність результатів та дозволяє уникнути впливу різниці у темпі навчання на ефективність оптимізації.

Процес тренування моделі виконувався за допомогою бібліотеки TensorFlow на апаратному забезпеченні, що включає графічну карту RTX 3060 Ti з GDDR6X 8 Gb пам'яті, процесор Ryzen 5 5600 та 16 Гб оперативної пам'яті. Тренування моделі включало кілька етапів: початковий запуск моделі з початковими параметрами, проходження через набір тренувальних даних, обчислення втрат (loss), оновлення параметрів моделі для зменшення цих втрат.

Для кожної епохи модель проходила через весь набір тренувальних даних, і для другого набору даних процес займав приблизно 40 хвилин на кожну епоху. Протягом кожної епохи модель обчислювала втрати на тренувальних даних і коригувала свої параметри для мінімізації цих втрат. Після кожної епохи модель також оцінювалася на перевірному наборі даних, щоб перевірити, як добре вона узагальнює нові дані, які не були використані під час тренування. Загалом було виконано 9 епох тренування, що зайняло приблизно 7 годин 20 хвилин. Під час цього процесу

спостерігалось поступове зменшення втрат як на тренувальному, так і на перевірному наборах даних, що свідчить про ефективне навчання моделі. Результати були візуалізовані у вигляді графіків та таблиць.

Для збору та аналізу результатів навчання використовувався інструмент TensorBoard, який надає можливість візуалізації метрик навчання, таких як втрата та точність, на різних етапах процесу навчання. Це дозволяє систематично оцінювати та порівнювати продуктивність стохастичного градієнтного спуску та адаптивної оцінки моменту під час навчання моделі з метою визначення найефективнішого оптимізатора для даної задачі класифікації зображень.

На рис.1 представлено приклад декількох зображень з набору даних (ambulance, cup, strawberry) та частину назв різноманітних категорій даних.

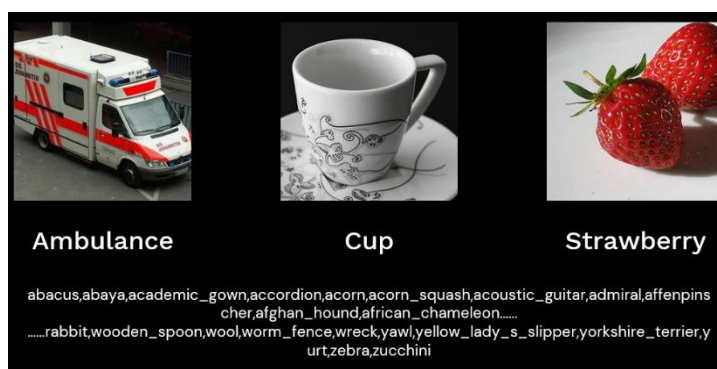


Рис.1. Зображення та назви категорій у наборі даних.

Fig. 1. Images and names of categories in the dataset.

РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

На рис.2 зображено графіки результатів тренування моделі з використанням оптимізаторів SGD та Adam. Графіки зверху показують результати для оптимізатора SGD. Ліворуч (рис.2а) відображена точність (accuracy) моделі, а праворуч (рис.2б) — втрати (loss) моделі по епохах для тренувальних та перевірних даних. Пунктирні представляють фактичні значення, тоді як суцільні лінії показують згладжені значення. Для оптимізатора SGD точність тренувальних даних поступово зростає і досягає згладженого значення 0,8297 та фактичного значення 0,8343 після 16 епох. Точність перевірних даних також зростає до згладженого значення 0,8141 і фактичного значення 0,8117. Втрати на тренувальних даних зменшуються до згладженого значення 0,4694 і фактичного значення 0,4545, а втрати на перевірних даних зменшуються до згладженого значення 0,541 і фактичного значення 0,5498. Загальний час тренування для кожної епохи становив 11,75 хвилин. Для оптимізатора Adam (графіки внизу) точність тренувальних даних зростає до згладженого значення 0,8218 та фактичного значення 0,8252 після 19 епох.

Точність моделі на перевірних даних зростає до згладженого значення 0,7929 і фактичного значення 0,805. Втрати на тренувальних даних зменшуються до згладженого значення 0,4901 і фактичного значення 0,48, а втрати на перевірних даних зменшуються до згладженого значення 0,5812 і фактичного значення 0,5479. Загальний час тренування для кожної епохи становив 13,91 хвилин. Порівнявши результати двох оптимізаторів, можна побачити, що оптимізатор SGD досягає трохи кращої точності на перевірних даних, тоді як оптимізатор Adam показує трохи менші втрати на перевірних даних. Виходячи з цих результатів, можна зробити висновок, що оптимізатор SGD є дещо кращим варіантом для цієї задачі, оскільки забезпечує вищу точність на перевірних даних.

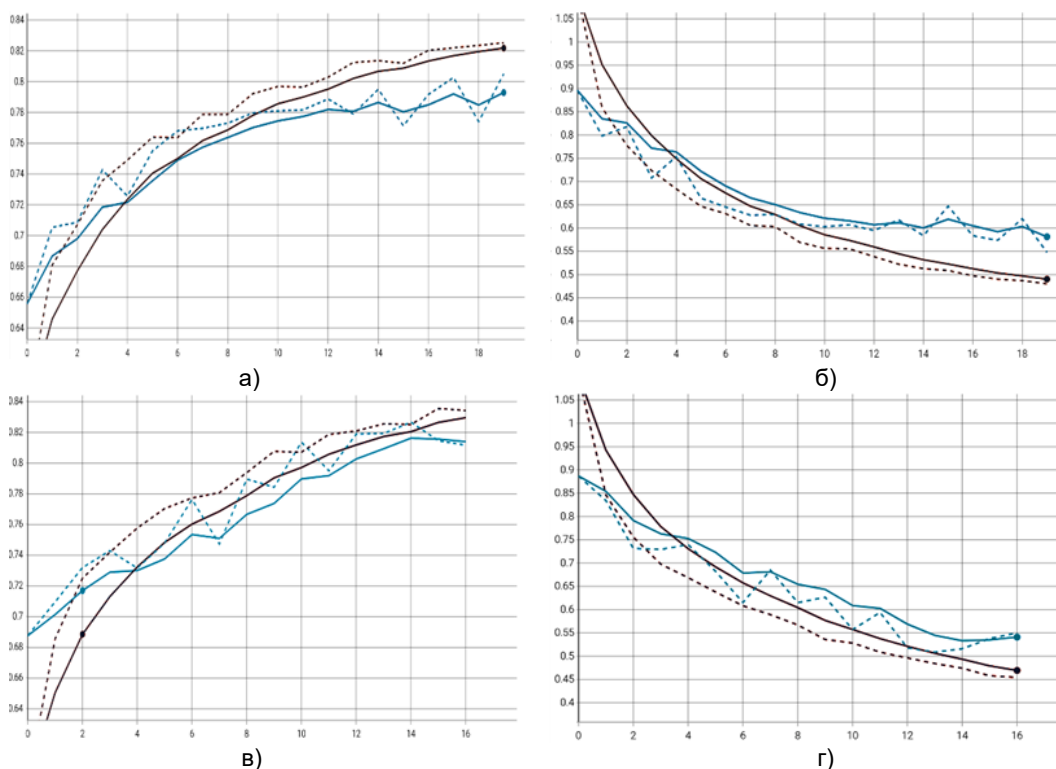


Рис.2. Графіки залежності точності та втрат від кількості епох навчання (перший набір даних): точність оптимізатора SGD (а), втрати оптимізатора SGD (б), точність оптимізатора Adam (в), втрати оптимізатора Adam (г). Коричневі лінії – тренувальні дані, а сині — перевірні. Пунктирні лінії – фактичні значення, суцільні лінії – згладжені значення.

Fig. 2. Graphs of the dependence of accuracy and loss on the number of training epochs (first dataset): accuracy of the SGD optimizer (a), losses of the SGD optimizer (б), accuracy of the Adam optimizer (в), losses of the Adam optimizer (г). Brown lines – training data, blue ones – validation data. Dashed lines – actual data, solid ones – smoothed data.

Крім того, час тренування для SGD був трохи коротшим (11,75 хвилин за 16 епох) порівняно з Adam (13,91 хвилин за 19 епох), що також є перевагою. Враховуючи вищу точність і менший час тренування, оптимізатор SGD виглядає кращим вибором для даного завдання. Сумарно отримані результати для влучності та втрат моделей отримані на першому тренувальному та перевірному наборі даних представлено у таблиці 1.

На рис.3 представлено графіки результатів точності і втрат для тренувальних та перевірних даних другого набору для оптимізаторів SGD та Adam. Для оптимізатора SGD (Рис. 3а – точність, Рис. 3б – втрати) темно-сині лінії показують тренувальні дані, а світло-сині — перевірні. Пунктирні лінії представляють фактичні значення, тоді як суцільні лінії показують згладжені значення. Точність тренувальних даних поступово зростає до згладженого значення 0,2479, а фактичне значення становить 0,2663. Перевірна точність також збільшується, досягаючи згладженого значення 0,239 і фактичного значення 0,2498.

Таблиця 1. Кінцеві значення тренування моделі для першого набору даних.
Table 1. Final values of model training using the first dataset.

		Оптимізатор			
		SGD		Adam	
		Фактичні значення	Згладжені значення	Фактичні значення	Згладжені значення
Точність	Тренувальна	0,8343	0,8297	0,8252	0,8218
	Перевірні	0,8117	0,8141	0,805	0,7929
Втрати	Тренувальні	0,4545	0,4694	0,48	0,4901
	Перевірні	0,5498	0,541	0,5479	0,5812
Час(хв)		11,75		13,91	
К-сть циклів навчання		16		19	

Втрати на тренувальних даних зменшуються до згладженого значення 3,7209 і фактичного – 3,59, тоді як перевірні втрати зменшуються до згладженого значення 3,8222 і фактичного – 3,7522. Загальний час тренування становить 7,282 год.

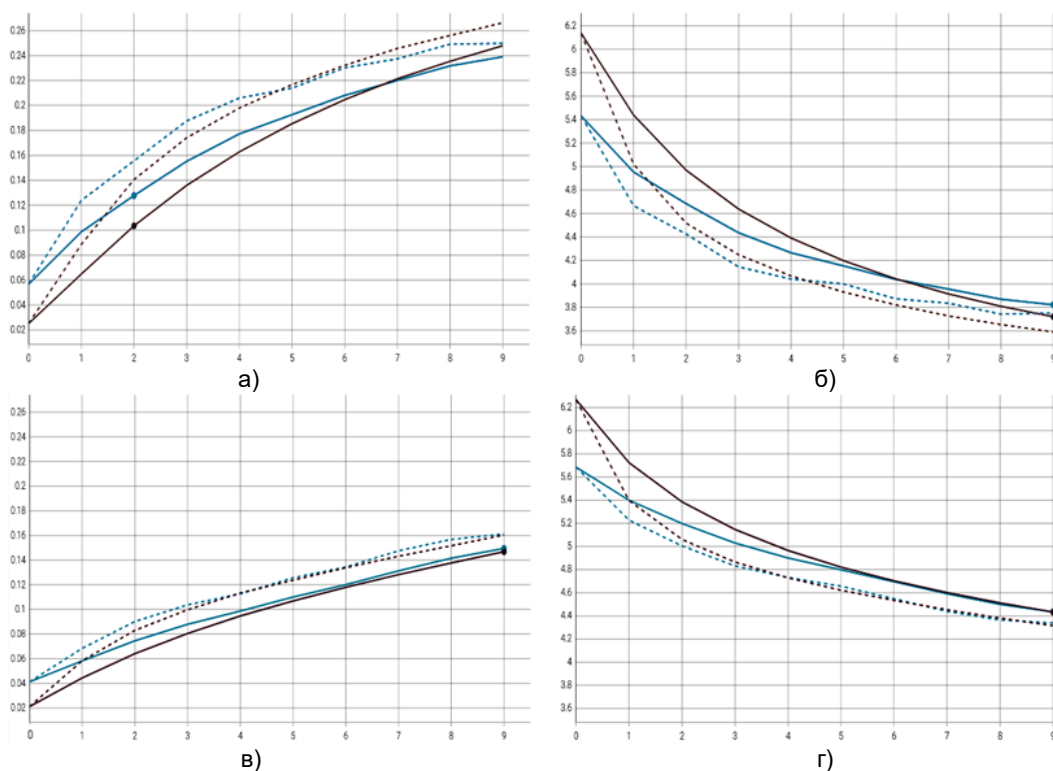


Рис.3. Графіки залежності точності та втрат від кількості епох навчання (другий набір даних): точність оптимізатора SGD (а), втрати оптимізатора SGD (б), точність оптимізатора Adam (в), втрати оптимізатора Adam (г). Коричневі лінії – тренувальні дані, а сині – перевірні. Пунктирні лінії – фактичні значення, суцільні – згладжені значення.

Fig. 3. Graphs of the dependence of accuracy and loss on the number of training epochs (second dataset): accuracy of the SGD optimizer (a), losses of the SGD optimizer (b), accuracy of the Adam optimizer (v), losses of the Adam optimizer (r). Brown lines – training data, blue ones – validation data. Dashed lines – actual data, solid ones – smoothed data.

Для оптимізатора Adam (графіки внизу, зліва – точність, справа - втрати) тем-но-сині лінії також представляють тренувальні дані, а світло-сині — перевірні. Пунктирні лінії відображають фактичні значення, тоді як суцільні лінії показують згладжені значення. Точність тренувальних даних збільшується до згладженого значення 0,1468, а фактичне значення становить 0,1604. Перевірна точність також збільшується до згладженого значення 0,1495 і фактичного значення 0,1612. Втрати на тренувальних даних зменшуються до згладженого значення 4,4333 і фактичного значення 4,3156, тоді як перевірні втрати зменшуються до згладженого значення 4,4347 і фактичного значення 4,3387. Загальний час тренування становить 7,396 годин. Сумарно отримані результати для влучності та втрат моделей отримані на другому тренувальному та перевірному наборі даних представлені у таблиці 2.

Таблиця 2. Кінцеві значення тренування моделі використовуючи другий набір даних.

Table 2. Final values of model training using the second dataset.

		Оптимізатор			
		SGD		Adam	
		Фактичні значення	Згладжені значення	Фактичні значення	Згладжені значення
Точність	Тренувальна	0,2663	0,2479	0,1604	0,1468
	Перевірна	0,2498	0,239	0,1612	0,1495
Втрати	Тренувальні	3,59	3,7209	4,3156	4,4333
	Перевірні	3,7522	3,8222	4,3387	4,4347
Час(год)		7,28		7,4	
К-сть циклів навчання		9		9	

Виходячи з цих результатів, видно, що обидва оптимізатори показують зростання точності та зменшення втрат із збільшенням кількості епох. Однак, SGD досяг кращої точності як на тренувальних, так і на перевірних даних у порівнянні з Adam. Також втрати для SGD були нижчими, що вказує на ефективніше навчання моделі з використанням цього оптимізатора.

На основі даних можна визначити деякі характеристики результатів моделі навчання:

- Зростання точності з кожною епохою навчання, наприклад, від 0,02 до 0,16 з першої до десятої епохи навчання. Це свідчить про те, що модель поступово вдосконалюється під час навчання.
- Стабільність точності на етапі перевірки, наприклад, від 0,04 до 0,161 з першої до десятої епохи перевірки. Це може свідчити про те, що модель добре узгоджується з новими даними після навчання.
- Точність навчання на початку зазвичай невелика, але з кожною новою епохою вона зростає. Наприклад, у першій епосі точність для навчання та перевірки становить 0,02 та 0,04, відповідно, а у десятій епохі ці величини зростають до 0,16 та 0,161, відповідно.
- Потенційне перенавчання спостерігається, якщо точність на навчальному наборі даних значно перевищує точність на перевірному наборі. Однак, у

представленому випадку точності на обох наборах досить близькі, що може свідчити про відсутність перенавчання або його незначний вплив.

Загалом, результати показують, що модель покращується з кожною новою епохою навчання, і точність на перевірки підтверджує стабільність та ефективність моделі.

Тестування роботи моделі для класифікації зображень було проведено для різноманітних зображень. Тестування виконувалось на нейромережі з використанням оптимізатора SGD, який забезпечує кращі результати порівняно з оптимізатором Adam. На рис.4а показано зображення автомобіля швидкої допомоги (з набору даних для навчання (рис.1)), та результат його розпізнавання у форматі JSON (запис об'єктів JavaScript).



(a)



(б)

Рис.4. Тестові зображення та визначені категорії: класифікація зображення швидкої допомоги (а), класифікація зображення легкового автомобіля (б).

Fig. 4. Test images and defined categories: ambulance image classification (a), car image classification (б).

Із рис. 4а видно, що першою визначеною категорією є автомобіль швидкої допомоги (ambulance), тобто категорія визначена цілком вірно. Також, оскільки в автомобіля швидкої допомоги є певні елементи схожості з поліцейським мікроавтобусом, то одразу наступною йде ця категорія. Нижче на рис.4б показано вибране для тесту зображення автомобіля, яке має складний ракурс, тому нейронна мережа визначила, що це зображення позашляховика, хоча і наступна категорія визначена правильно.

ВИСНОВОК

У результаті виконаної роботи було створено систему автоматичної класифікації зображень на основі згорткової нейронної мережі з використанням бібліотеки TensorFlow та системи візуалізації TensorBoard для збереження метрик навчання. Порівнювалися два популярні оптимізатори: Adam та SGD.

Виходячи з аналізу таблиць результатів, оптимізатор SGD показав кращу продуктивність у порівнянні з Adam. Початкова точність тренування та перевірки для SGD була трохи вищою, а швидкість зростання точності суттєво перевершувала

Adam, досягаючи значно вищих значень на 10-й епосі. Остаточна точність моделі з оптимізатором SGD виявилася вищою як на тренувальному, так і на перевірному наборі даних, що свідчить про більш ефективне навчання. Крім того, SGD демонстрував швидше зменшення втрат, знову ж таки забезпечуючи нижчі кінцеві значення втрат порівняно з Adam. Додатково слід зауважити, що час тренування для SGD був трохи меншим, що також є його перевагою порівняно з оптимізатором Adam. Для першого набору даних час тренування для SGD був 11,75 хвилин за 16 епох порівняно з Adam – 13,91 хвилин за 19 епох. В обох випадках моделі не демонстрували значної різниці між тренувальними та перевірними наборами, що свідчить про відсутність перенавчання та добру здатність до узагальнення. Таким чином, для даних моделей та наборів даних оптимізатор SGD був визначений як більш ефективний, забезпечуючи кращу продуктивність та стабільність роботи системи автоматичної класифікації зображень.

ЛІТЕРАТУРА

- [1] Y.Zhang *Why Transformers Need Adam: A Hessian Perspective*/ C.Chen, T.Ding, Z.Li, R.Sun, Z.Luo// Cornell Univer. Machine Learning arXivLabs: [arXiv:2402.16788](https://arxiv.org/abs/2402.16788)
- [2] N. Landro *Mixing ADAM and SGD: a Combined Optimization Method*/ I.Gallo, R.L.Grassa// Cornell Univer. Machine Learning arXivLabs: [arXiv:2011.08042](https://arxiv.org/abs/2011.08042)
- [3] Diederik P. Adam: A Method for Stochastic Optimization/ Kingma, Jimmy Ba// Cornell Univer. Machine Learning arXivLabs: [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- [4] P. J. Kamali *Stochastic Gradient Descent outperforms Gradient Descent in recovering a high-dimensional signal in a glassy energy landscape*/ P. Urbani// Cornell Univer. Machine Learning arXivLabs: [arXiv:2309.04788](https://arxiv.org/abs/2309.04788)

DETERMINATION OF THE BEST OPTIMIZER FOR A NEURONETWORK IN THE DEVELOPMENT OF AUTOMATIC IMAGE TAGGING SYSTEMS

Andrian Kozynets 

*Ivan Franko National University of Lviv
50 Dragomanov St., Lviv 79005, Ukraine*

ABSTRACT

Background. Choosing the best optimizer is an important step in developing efficient automatic image classification systems. In particular, for neural networks based on convolutional neural networks (CNNs), the choice between popular optimization methods such as Adam (Adaptive Moment Estimation) and SGD (Stochastic Gradient Descent, SGD) can significantly affect their performance. Comparing these optimizers allows us to determine the most suitable optimizer for solving specific machine learning problems.

Methods and tools. The TensorFlow library and the TensorBoard machine learning visualization interface were used to train and test the models. The neural networks were trained on two different datasets with significantly different characteristics. For each dataset, training was performed using two optimizers: Adam and SGD. When analyzing the results, such parameters as training and validation accuracy, as well as training and validation losses, were taken into account.

Results and Discussion. According to the results, the SGD optimizer showed better performance indicators compared to Adam. The initial accuracy for SGD was higher, and a significant excess of the accuracy growth rate was observed. In addition, the final accuracy

of the model with the SGD optimizer was higher on both the training and validation datasets, which indicates a more efficient training process using this optimizer.

Conclusions. It was determined that the use of the SGD optimizer leads to better results compared to the Adam optimizer for automatic image classification tasks. This is confirmed by higher accuracy indicators and a faster training process.

Keywords: TensorFlow, Convolutional Neural Network, Adaptive Moment Estimation, Stochastic Gradient Descent, GPU, TPU.