

УДК 004.85

В.В. ГНАТУШЕНКО, Т.М. ФЕНЕНКО
Український державний університет науки і технологіїО.Л. ДОРОШ
Інститут транспортних систем і технологій НАН України**РЕЗУЛЬТАТИ НАЛАШТУВАННЯ ПАРАМЕТРІВ НЕЙРОННИХ ГЛИБОКИХ
МЕРЕЖ ЩОДО РОЗПІЗНАВАННЯ FASHION MNIST DATASET**

Проведено дослідження моделей згорткової нейронної мережі (Convolutional neural network – CNN) з метою підвищення точності розпізнавання Fashion MNIST DATASET.

З огляду відомо, що розпізнавання елементів одягу набору Fashion MNIST є більш складним ніж розпізнавання набору рукопису цифр MNIST. Набор одягу Fashion-MNIST рекомендовано для досліджень різних архітектур нейронних мереж. Найкращі результати якості розпізнавання Fashion MNIST DATASET отримано за згортковою нейронною мережею.

В даній роботі було метою покращити точність розпізнавання Fashion MNIST DATASET за рахунок дослідження різних архітектур CNN та їх параметрів.

Обрано дві архітектури послідовної згорткової нейронної мережі з тих, у яких точність розпізнавання Fashion MNIST DATASET більше ніж 93%. Проведено дослідження їх архітектур та параметрів. Моделі відповідають визначенню нейронних глибоких мереж та мають різну кількість шарів. В дослідженнях моделей показано вплив параметрів `batch_size`, `validation_split`, `validation_data` на точність розпізнавання, а також варіанти розташування шару `BatchNormalization` та шару активації; вплив параметра “filters” для згорткового шару. Крім того, було використано два варіанти вибору валідаційної вибірки: перший – з набору даних для навчання (20%), а другий – набір даних тестування. При розрахунках число `epoch` навчання дорівнювало 20. В процесі навчання вирішувалось питання не допустити перенавчання за допомогою аналізу функції втрат.

Використано бібліотеки TensorFlow, Keras, мову програмування Python. Розроблено програмні модулі, які було реалізовано у хмарному сервісі Google Colab.

В результаті досліджень підтверджено заявлену у роботах інших авторів точність розпізнавання >93% Fashion MNIST DATASET та отримано покращену точність розпізнавання в 94,16% для однієї з обраних моделей. Обґрунтовано вплив параметру `batch_size` на точність розпізнавання, обрано значення `batch_size` відповідно найкращому результату розпізнавання Fashion MNIST DATASET. Продемонстровано, що збільшення кількості даних для навчання покращує точність розпізнавання при використанні параметра `valid_data==(X_test, X_test_labels)` замість `valid_split` для даних навчання. Наведені результати чисельного експеримента, які підтверджують важливість та корисність застосування методів регуляризації для вирішення проблеми перенавчання: налаштування шарів Dropout дозволило покращити точність розпізнавання.

Ключові слова: глибоке навчання, згорткова нейронна мережа, архітектура CNN, точність розпізнавання, налаштування параметрів CNN, Fashion MNIST DATASET, PYTHON, KERAS, TENSORFLOW, GOOGLE COLAB.

HNATUSHENKO Vik., FENENKO T.

Ukrainian State University of Science and Technologies, Dnipro, Ukraine

DOROSH O.

Institute of Transport Systems and Technologies of the National Academy of Sciences of Ukraine, Dnipro

**RESULTS OF NEURAL DEEP NETWORKS PARAMETER TUNING
FOR FASHION MNIST DATASET RECOGNITION**

A study of convolutional neural network (CNN) models was conducted in order to obtain better recognition quality of the Fashion MNIST DATASET.

From the review, it is known that the Fashion MNIST DATASET recognition set is more difficult than MNIST DATASET recognition. The Fashion-MNIST DATASET is recommended for research on different neural network architectures. The best Fashion MNIST DATASET recognition quality results were obtained by convolutional neural network.

In this work, the goal was to improve the recognition quality of the Fashion MNIST DATASET by studying different CNN architectures and their parameters.

Two consecutive convolutional neural network architectures were selected from those with Fashion MNIST DATASET recognition quality greater than 93%. A study of their architectures and parameters was conducted. The models correspond to the definition of neural deep networks and have different number of layers. Model studies show the influence of `batch_size`, `validation_split`, and `validation_data` parameters on recognition accuracy, as well as location options for

the BatchNormalization layer and the activation layer; the effect of the “filters” parameter for the convolutional layer. In addition, two validation sample selection options were used: the first one was from the training dataset (20%) and the second one was the testing dataset. In the calculations, the number of training epochs was equal to 20. In the training process, the issue of preventing overtraining was solved using the analysis of the loss function.

TensorFlow, Keras, Python programming language were used. Software modules were developed and implemented in the Google Colab cloud service.

As a result of the research, the recognition quality >93% of the Fashion MNIST DATASET declared in the works of other authors was confirmed, and an improved recognition quality of 94.16% was obtained for one of the selected models. The influence of the batch_size parameter on the recognition quality is substantiated, and the batch_size value is chosen according to the best recognition result of the Fashion MNIST DATASET. Increasing the amount of training data has been shown to improve recognition performance when using valid_data==(X_test, X_test_labels) instead of valid_split for training data. The results of a numerical experiment are presented, which confirm the importance and usefulness of applying regularization methods to solve the retraining problem: adjusting the Dropout layers allowed to improve the recognition accuracy.

Keywords: deep learning, convolutional neural network, CNN architecture, recognition quality, CNN parameter tuning, Fashion MNIST DATASET, PYTHON, KERAS, TENSORFLOW, GOOGLE COLAB,

Постановка проблеми

Глибокі нейронні мережі (CNN) у задачах розробки систем штучного інтелекту є одним із найпопулярніших підходів і дозволяють отримати високу точність рішення. Це такі галузі досліджень, як комп'ютерний зір, розпізнавання мови, експертні системи та інші [1].

Вже більше ніж 20 років тестування задач комп'ютерного зору почали проводити на наборі MNIST рукописних символів цифр. Завдяки розробці моделі глибокого навчання на основі архітектури CNN було отримано найкращий результат розпізнавання цифр 99,7% [2], але не дало змоги отримати такі високі результати для набору даних Fashion MNIST.

Аналіз останніх досліджень і публікацій

В даний час запропоновано багато архітектур CNN, які призначені для розпізнавання зображень і тестувалися на Fashion MNIST DATASET. Найкращі результати щодо обраного набору зображень, а саме більше 93% якості розпізнавання, отримано з використанням загорткових нейронних мереж [2]. Такий результат є високим і дозволяє зробити висновок, що задача розпізнавання елементів одягу Fashion MNIST DATASET вже не є актуальною з точки зору отримання кращої якості розпізнавання. Але Fashion MNIST DATASET можна використати для тестування різних архітектур та налаштування параметрів моделей, що не втрачає актуальності. Таким чином, Набір даних Fashion MNIST часто використовують для перевірки нових ідей CNN. В [3] розглянуто 5 архітектур CNN, які теж протестовано на наборі Fashion MNIST, в результаті отримано точність розпізнавання трохи більше ніж 93%. На сайті <http://www.eamonfleming.com/> [4] наведено іншу архітектуру моделі CNN згідно якої точність розпізнавання теж більше 93%.

В [5] наведені результати розпізнавання Fashion MNIST DATASET за допомогою програмного засобу, який було розроблено з використанням мови Python, бібліотек TensorFlow та Keras та який був використаний в даній роботі для дослідження архітектур та налаштування CNN.

Мета дослідження

Метою роботи є покращення результатів розпізнавання Fashion MNIST DATASET з використанням різних архітектур CNN, налаштування параметрів та регуляризації і надання рекомендацій.

Викладення основного матеріалу дослідження

В результаті аналізу досліджень результатів розпізнавання Fashion MNIST DATASET обрано дві CNN, які відповідали точності розпізнавання 93% [3, 4]. В табл. 1 наведено інформацію щодо обраних CNN.

Таблиця 1

Архітектури згорткових нейронних мереж

N	Позначення моделі	Архітектура CNN	Кількість нейронів	Розмір байт	Посилання
1	Cnn_93	4 згорткові шари з фільтр. (3x3) та 1 повнозв'язний шар	1 637 570	19 722 408	[4]
2	Cnn_a5	4 згорткові шари з фільтр. (3x3) та 2 повнозв'язних шари	1 676 266	20 177 840	[3] Architecture 5

У роботі реалізовано програмний код мовою Python з використанням KERAS і Tensorflow, розрахунки були проведені у Google Colab.

Для кожної з архітектур реалізовано функцію побудови моделі CNN та функцію, що містить інформацію про модель. Функція з інформацією по моделі надає такі позначення у заголовках графіків: “C” для згорткового шару Conv2D; “B” для шару пакетної нормалізації BatchNormalization; “M” для шару пулінга MaxPooling2D; “D” для шару Dropout; “F” для шару Flatten; “D” для повнозв’язного шару Dense; “A” для шару активації.

Навчання моделі виконується за викликом методу fit:

```
fit(x=None, y=None, batch_size=None, epochs=1, verbose=1, callbacks=None,
    validation_split=0.0, validation_data=None, shuffle=True,
    class_weight=None, sample_weight=None,
    initial_epoch=0, steps_per_epoch=None, validation_steps=None)
```

Основні параметри методу fit: *x* – масив навчальних даних; *y* – масив цілей (класів, міток); *batch_size* – розмір пакета навчання; *epochs* – число епох навчання неронної мережі; *verbose* – індикатор для виводу інформації в процесі навчання; *callbacks* – список процедур, що виконуються в процесі навчання; *validation_split* – значення між 0 та 1, частка валідаційних даних; *validation_data* – явна вказівка на валідаційні дані.

Модель за назвою *cnn_93* має структуру, яку показано на рис.1, отримана за методом *summary*.

При дослідженні моделі *cnn_93* було з'ясовано її особливості [4]: шар BatchNormalization нормує ваги нейронів на виході згорткового шару Conv2D; у процесі навчання валідаційні дані співпадають з тестовими даними, а саме з тими, які призначені для перевірки (тестування) моделі; після процесу навчання використовується найкраща модель, яка отримується в процесі навчання замість моделі, яка відповідає останній епосі навчання.

Проведено дослідження моделі *cnn_93* з питань: – вплив параметрів *batch_size*, *validation_split*, *validation_data* на точність розпізнавання; – варіанти розташування шару BatchNormalization та шару активації; – вплив параметра “filters” для згорткового шару Conv2D з ім'ям *name='93_1_4'*.

Результати дослідження впливу параметра *batch_size* на точність розпізнавання отримані за таких умов: – число епох навчання дорівнює 20; – валідаційні дані – з даних для навчання, параметр *validation_split=0.2*. Як видно з табл. 2 виконання однієї епохи, зі збільшенням значення *batch_size*, зменшувалося, а розмір пакету *batch_size* несуттєво вплинув на точність розпізнавання. Найкращі результати отримані при *batch_size=96*.

На рис. 2 показано графік залежності точності навчання від епох за варіантом 2 табл. 2. З графіків видно швидкість налаштування моделі: з 5-й по 10-ту епоху крива валідаційних даних практично стає паралельною осі абсцис (точність розпізнавання дорівнює приблизно 0,925).

На рис. 3 показано графік залежності функції втрат від епох за варіантом 2 табл. 2. По формі функції втрат видно, що перенавчання відсутнє, також є можливість зменшити кількість епох до 10.

Layer (type)	Output Shape	Param #
93_l_1 (Conv2D)	(None, 28, 28, 32)	320
93_l_2 (BatchNormalization)	(None, 28, 28, 32)	128
93_l_3 (Dropout)	(None, 28, 28, 32)	0
93_l_4 (Conv2D)	(None, 28, 28, 32)	9248
93_l_5 (Dropout)	(None, 28, 28, 32)	0
93_l_6 (Conv2D)	(None, 28, 28, 24)	6936
93_l_7 (Dropout)	(None, 28, 28, 24)	0
93_l_8 (Conv2D)	(None, 28, 28, 64)	13888
93_l_9 (MaxPooling2D)	(None, 14, 14, 64)	0
93_l_10 (Dropout)	(None, 14, 14, 64)	0
93_l_11 (Flatten)	(None, 12544)	0
93_l_12 (Dense)	(None, 128)	1605760
93_l_13 (Dropout)	(None, 128)	0
93_l_14 (Dense)	(None, 10)	1290
Total params: 1,637,570		
Trainable params: 1,637,506		
Non-trainable params: 64		

Рис. 1. Структура моделі cnn_93

Таблиця 2

Модель cnn_93 (validation_split=0.2 epochs=20)

№ варіанту	1	2	3	4	5
batch_size	64	96	128	256	512
Час виконання однієї епохи (сек)	15	15	13	11	10
Точність моделі на тесті	0.9253	0.9273	0.9268	0.9262	0.9259
Точність найкращої моделі на тесті	0.9251	0.9302	0.9268	0.9261	0.9262

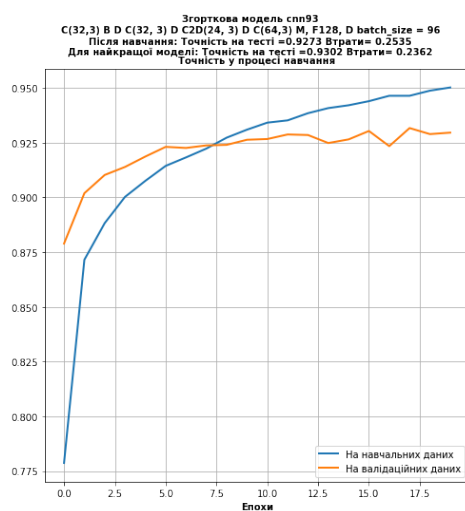


Рис. 2. Точність навчання-модель cnn_93

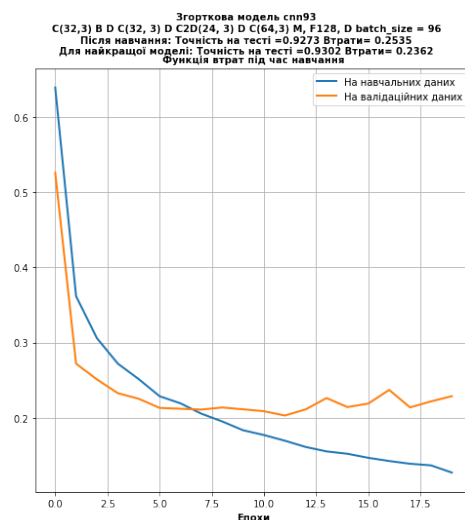


Рис. 3. Функція втрат-модель cnn_93

Висновки щодо дослідження впливу параметрів batch_size, validation_split, validation_data для моделі cnn_93: при розпізнаванні зображень елементів одягу набору Fashion MNIST: рекомендовано використовувати batch_size=96 і як валідаційні дані застосовувати тестові дані: validation_data=(X_test, X_test_labels).

В подальших дослідженнях моделі cnn_93 змінимо розташування шару активації та шару BatchNormalization. Автори шару BatchNormalization, який нормує ваги нейронів [6], рекомендують встановити його перед викликом функції активації:

```
cnn.add(Conv2D(32, kernel_size=3, activation=None,
input_shape=(28,28,1),padding='same'))
cnn.add(BatchNormalization())
cnn.add(Activation('relu'))
```

Згідно з рекомендаціями у шарі Conv2D з ім'ям name='93_l_1' встановимо activation=None та після шару BatchNormalization додамо шар активації `cnn_simple_model.add(tf.keras.layers(Activation('relu'))`.

Таблиця 3

Модель cnn_93 (validation_data, batch_size=96, epochs=20)

	1	2
Варіант	Conv2D (...activation='relu' ...) BatchNormalization	Conv2D (...activation=None ...) BatchNormalization Activation('relu')
Точність моделі на тесті	0.9269	0.9265
Точність найкращої моделі на тесті	0.9317	0.9307

Порівняльний аналіз розпізнавання зображень Fashion MNIST з різними варіантами розташування шару BatchNormalization надані у табл. 3. Варіант «1» підключення шару BatchNormalization після шару Conv2D(...activation='relu'...) є найкращім.

Результати впливу параметра “filters” для згорткового шару Conv2D з ім'ям name='93_l_4' надані у табл. 4 (batch_size=96, epochs=20, validation_data). Для найкращої моделі точність становить 93,35% для filters=24; для згорткового шару Conv2D з ім'ям name='93_l_4' значення параметра filters можна задавати в діапазоні від 18 до 64.

Таблиця 4

Результати дослідження впливу фільтру, модель cnn_93

Варіант	1	2	3	4	5
filters	18	24	32	64	96
Точність моделі на тесті	0.9314	0.9335	0.9269	0.931	0.9251
Точність найкращої моделі на тесті	0.9314	0.9335	0.931	0.931	0.927

Аналогічно дослідженню моделі cnn_93, проведено дослідження моделі cnn_a5 (табл. 1). Структура моделі cnn_a5, яка отримана за методом summary, – на рис. 4.

Layer (type)	Output shape	Param #
a5_l_1 (Conv2D)	(None, 28, 28, 32)	320
a5_l_2 (Conv2D)	(None, 28, 28, 32)	9248
a5_l_3 (MaxPooling2D)	(None, 14, 14, 32)	0
a5_l_4 (Dropout)	(None, 14, 14, 32)	0
a5_l_5 (Conv2D)	(None, 14, 14, 64)	18496
a5_l_6 (Conv2D)	(None, 14, 14, 64)	36928
a5_l_7 (MaxPooling2D)	(None, 7, 7, 64)	0
a5_l_8 (Dropout)	(None, 7, 7, 64)	0
a5_l_9 (Flatten)	(None, 3136)	0
a5_l_10 (Dense)	(None, 512)	1606144
a5_l_11 (Dropout)	(None, 512)	0
a5_l_12 (Dense)	(None, 10)	5130
Total params: 1,676,266		
Trainable params: 1,676,266		
Non-trainable params: 0		

Рис. 4. Структура моделі cnn_a5

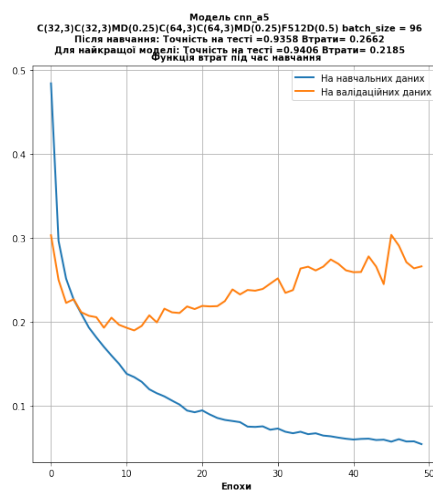
Вплив параметра `batch_size` на точність розпізнавання при прийнятих умовах розрахунку (`validation_split=0.2 epochs=20`) аналогічний як і при дослідженні моделі `cnn_93`. Але тут вдалося отримати точність розпізнавання трохи вищу, ніж у роботі [3], з якої було взяту архітектуру глибокої мережі, за рахунок налаштування гіперпараметрів.

Таблиця 5

Модель `cnn_a5`, `validation_data=(X_test, X_test_labels)`

Варіант	1	2	3	4	5
<code>batch_size</code>	48	64	96	128	256
Час виконання однієї епохи (сек)	15	12	11	10	8
Точність моделі на тесті	0.9319	0.9321	0.9401	0.9365	0.9357
Точність найкращої моделі на тесті	0.9350	0.9351	0.9401	0.9377	0.9365

Як видно з табл. 5, при `batch_size=96`, валідаційних даних співпадаючих з тестовими, отримано точність 94%.

Рис. 5. Функція втрат в процесі навчання `cnn_a5`

Проведено більш ретельний аналіз моделі `cnn_a5` при `batch_size=96` та `validation_data=(X_test, X_test_labels)` шляхом більш тривалого навчання моделі за кількістю `epochs=50`. Результати наведені на рис. 5, де надано графік залежності функції втрат від епох. У разі ми бачимо перенавчання моделі.

Для усунення явища перенавчання застосуємо один із методів регуляризації (загрублення) моделі, а саме змінимо параметри шарів Dropout (з ім'ям `name="a5_l_4"` та з `name="a5_l_8"`). Результати розрахунків наведено у табл. 6

Таблиця 6

Модель `cnn_a5` (`validation_data`, `batch_size=96`, `epochs=50`)

Варіант	1	2	3	4	5	6	7
Dropout "a5_l_4" "a5_l_8"	0.25	0.3	0.35	0.4	0.45	0.5	0.6
Точність моделі на тесті	0.9358	0.9343	0.9394	0.9377	0.938	0.9383	0.9360
Точність найкращої моделі на тесті	0.9406	0.9404	0.9404	0.9414	0.9416	0.9413	0.9343

Висновок щодо дослідження впливу параметрів шарів Dropout (з ім'ям name=a5_l_4 та з name=a5_l_8) при batch_size=96, validation_data, epochs=50 для моделі cnn_a5: точність найкращої моделі на тесті складає **94.16%** при встановленні параметра шарів Dropout з іменами "a5_l_4" та "a5_l_8" на рівні 0.45.

Графіки функції втрат (рис. 6) та точності навчання (рис. 7) відповідні моделі cnn_a5 (табл. 6, вар. 5).

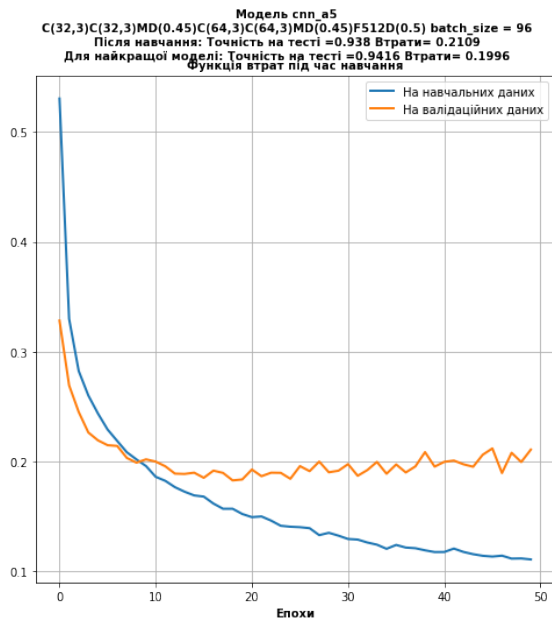


Рис. 6. Функція втрат – модель cnn_a5

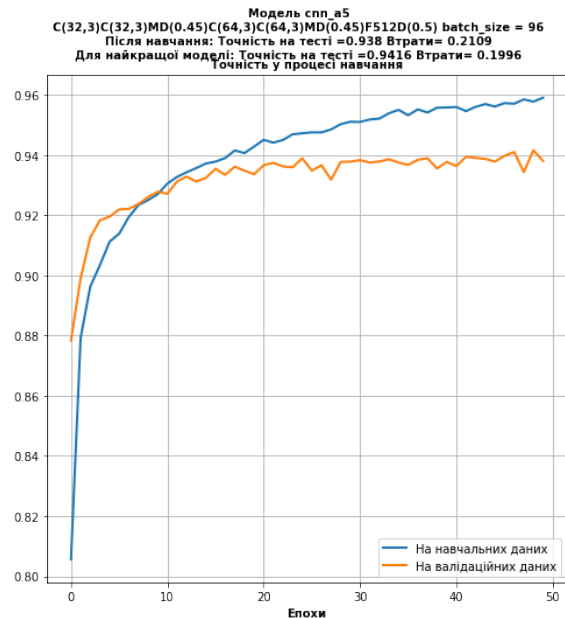


Рис. 7. Точність навчання – модель cnn_a5

Висновки по дослідженню моделі cnn_a5: обрані параметри batch_size=96 та параметри шарів Dropout з іменами "a5_l_4" та "a5_l_8" дозволили отримати точність розпізнавання 94.16%, без перенавчання, що краще ніж наведено в [3] результату розпізнавання, який дорівнює 93.56%.

Висновки

Проведено розпізнавання зображень Fashion MNIST DATASET за методами глибокого навчання та дослідження моделей двох архітектур нейронних мереж.

Підтверджено заявлену у роботах [3, 4] точність розпізнавання зображень на даних Fashion-MNIST 93%. Для моделі cnn_a5, яка відповідає CNN архітектурі 5 роботи [3], отримана точність розпізнавання в 94,16% завдяки налагодженню параметрів та регуляризації.

Обґрунтовано вплив параметру batch_size на точність розпізнавання, обрано значення batch_size відповідно найкращому результату розпізнавання зображень набору Fashion-MNIST. Підтверджено, що збільшення кількості даних для навчання покращує точність розпізнавання при використанні параметра valid_data==(X_test, X_test_labels) замість valid_split для даних навчання.

Рекомендації для подальшого дослідження обраних моделей:

- для збільшення кількості даних під час навчання можна використовувати методи Tensorflow для генерації додаткових навчальних даних (зразків) шляхом незначних змін (розміру, повороту, зміщення тощо);
- для автоматичного настроювання гіперпараметрів використовувати інструмент keras-tuner.

Список використаної літератури

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. London: MIT Press, 2016. 800 с.
2. Datasets [Електронний ресурс] – Режим доступу. – URL: <https://www.kaggle.com/> (дата звернення 10.09.2022).
3. Shivam S. Kadam, Amol C. Adamuthe, Ashwini B. Patil. CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset. Journal of Scientific Research. 2020. Vol. 64. № 02. P. 374-384. DOI:10.37398/JSR.2020.640251
4. Convolutional neural net tuning with the FASHION MNIST dataset[Електронний ресурс] – Режим доступу. – URL: <http://www.eamonfleming.com/projects/fashion-mnist.html>
5. Гнатушенко В. В., Дорош Н. Л., Фененко Т. М. Розпізнавання зображень набору FASHION MNIST методами глибокого навчання. Прикладні питання математичного моделювання. 2021. Т.4. № 1. С. 78-85. <https://doi.org/10.32782/KNTU2618-0340/2021.4.1.8>
6. Sergey Ioffe, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Електронний ресурс] – Режим доступу. – URL: <https://arxiv.org/abs/1502.03167>

References

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. London: MIT Press.
2. Datasets [Elektronnyi resurs] – Rezhym dostupu. – URL: <https://www.kaggle.com/>
3. Shivam, S. Kadam, Amol, C. Adamuthe, & Ashwini, B. (2020). Patil. CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset. Journal of Scientific Research. 64 (02), 374-384. DOI:10.37398/JSR.2020.640251
4. Convolutional neural net tuning with the FASHION MNIST dataset [Elektronnyi resurs] – Rezhym dostupu. – URL: <http://www.eamonfleming.com/projects/fashion-mnist.html>
5. Hnatushenko, V. V., Dorosh, N. L., & Fenenko, T. M. (2021). Rozpiznavannia zobrazhen naboru FASHION MNIST metodamy hlybokoho navchannia. Prykladni pytannia matematychnoho modeliuvannia. 4 (1), 78-85. <https://doi.org/10.32782/KNTU2618-0340/2021.4.1.8>
6. Sergey, Ioffe, & Christian, Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Elektronnyi resurs] – Rezhym dostupu. – URL: <https://arxiv.org/abs/1502.03167>

Гнатушенко Вікторія Володимирівна – д.т.н., професор кафедри інформаційних технологій та систем Українського державного університету науки і технології (м. Дніпро). e-mail: vvitagnat@gmail.com, ORCID: 0000-0001-5304-4144.

Фененко Тетяна Михайлівна – старший викладач кафедри інформаційних технологій та систем Українського державного університету науки і технології (м. Дніпро). e-mail: fenenkot@gmail.com, ORCID: 0000-0002-7631-3148.

Дорош Ольга Володимирівна – молодший науковий співробітник Інституту транспортних систем і технологій Національної академії наук України. e-mail: olchik.dorosh@gmail.com, ORCID: 0009-0004-9053-0386.