

УДК 004.932:004.5

Шепелюк Леонід,
студент гр КІМ-21,
факультет комп'ютерних та інформаційних технологій,
Луцький національний технічний університет,
м. Луцьк, Україна
LionLinkHome@gmail.com

Шепелюк Дмитро,
студент гр. КІМ-21,
факультет комп'ютерних та інформаційних технологій,
Луцький національний технічний університет,
м. Луцьк, Україна
LionLinkHome@gmail.com

ДОСЛІДЖЕННЯ ВПЛИВУ АРХІТЕКТУРИ ТА АЛГОРИТМІВ НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ НА ЯКІСТЬ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ

Аналіз проблем методів та алгоритмів автоматизованого розпізнавання номерів автомобілів показав, що найбільш перспективно використовувати нейромережеві алгоритми, які підлаштовуються до зміни умов спостереження засобів контролю дорожнього руху.

Використання алгоритму навчання нейронної мережі з використанням зворотного поширення помилки є одним з ключових етапів у функціонуванні нейронної мережі. На цьому етапі мережа навчається шляхом визначення важливих зв'язків між її нейронами та встановлення оптимальних ваг для кожного з цих зв'язків. Процес навчання нейронної мережі включає в себе використання алгоритмів, які сприяють визначенню вагових коефіцієнтів, що з'єднують нейрони. Ці алгоритми базуються на аналізі залежності між ваговими

коефіцієнтами та кількістю навчальних прикладів, що підтверджують цю залежність.

Для першого експерименту використана модель Sequential – це лінійна послідовність шарів, яка дозволяє будувати моделі нейронних мереж шар за шаром, де дані проходять через кожен шар послідовно. з трьома повнозв'язними шарами і оптимізатором стохастичний градієнтний спуск.

```
from keras.models import Sequential

#from keras.layers.core import Dense

from keras.layers import Dense

from keras.optimizers import SGD

from keras.callbacks import EarlyStopping

from keras.callbacks import ModelCheckpoint

. . .

EPOCHS = 70

checkpointer = ModelCheckpoint(filepath='~/MyDrive/GoogleColab/point.h5',
verbose=1, save_best_only=True)

H = model.fit(trainX, trainY, validation_data=(testX, testY),

              epochs=EPOCHS, batch_size=128,

              callbacks=[checkpointer])
```

Рис 1. Фрагмент коду навчання нейронної мережі

На тренувальній виборці та валідаційній виборці точність розпізнавання зростає наближаючись до 60 %. Починаючи з 50 епохи якість розпізнавання валідаційної вибірки майже не змінюється, що є ознакою «перенавчання» мережі.

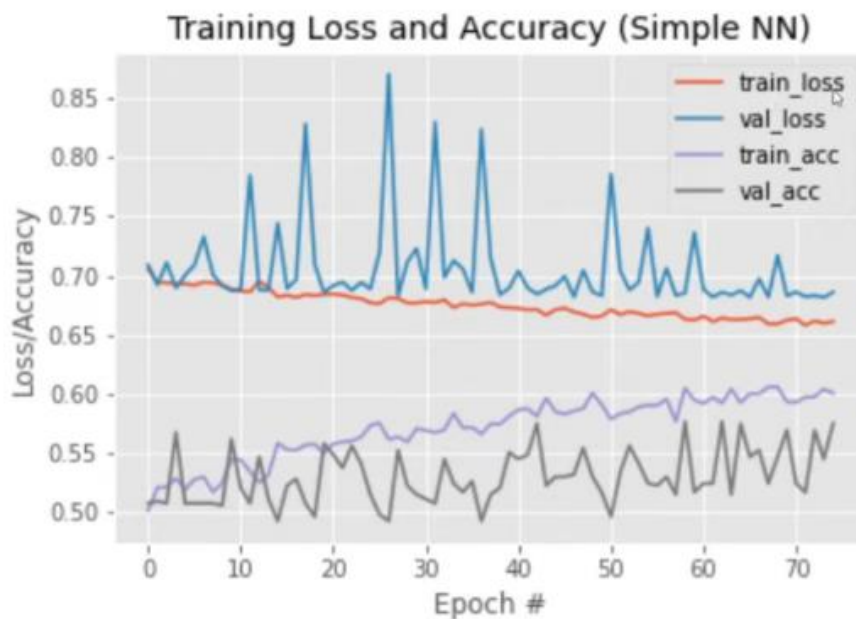


Рис. 2. Результати навчання лінійної моделі

Підвищити точність розпізнавання можна добавивши додатковий четвертий шар. Змінюючи параметри моделі такі як `batch_size`, крок градієнтного спуску, модель оптимізатора можна підвищити якість розпізнавання.

```

from keras.layers import Conv2D, Flatten, Dropout, Activation, MaxPooling2D
from keras.optimizers import Adam

model.add(Conv2D(32, (3, 3), padding="same", input_shape=(32, 32, 3)))
model.add(Activation("relu"))
model.add(Conv2D(32, (3, 3), padding="same", activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3, 3), padding="same", activation="relu"))
model.add(Conv2D(64, (3, 3), padding="same", activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
    
```

```
model.add(Dense(512, activation='relu'))  
model.add(Dropout(0.5))  
model.add(Dense(2, activation='softmax'))  
  
INIT_LR = 0.01  
  
opt = Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999, amsgrad=False)  
  
model.compile(loss="binary_crossentropy", optimizer=opt, metrics=["accuracy"])
```

Рис 3. Фрагмент коду навчання чотирьох шарової нейронної мережі

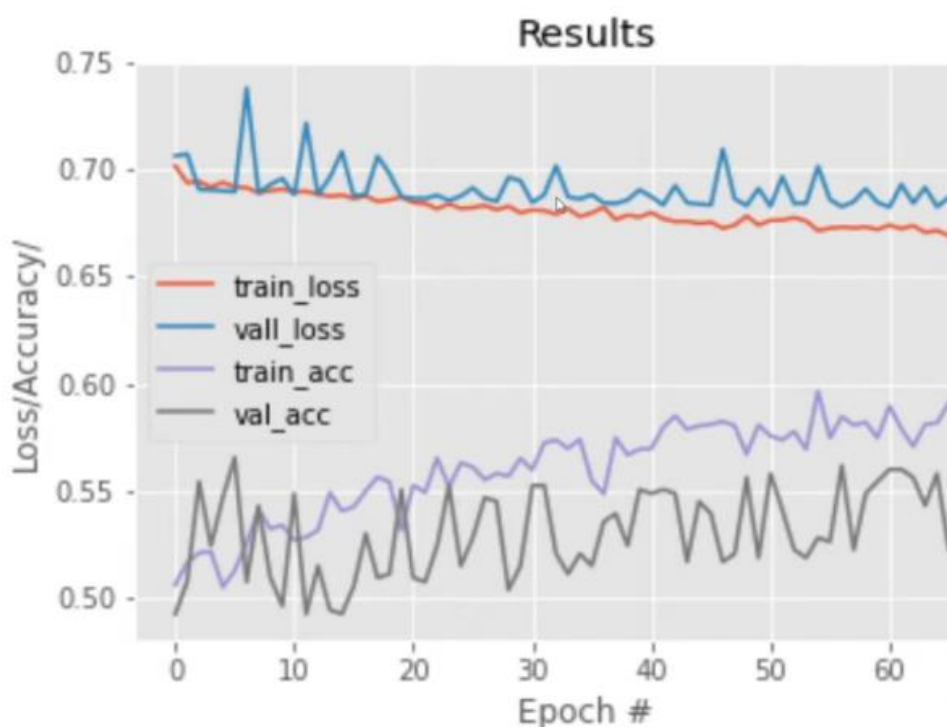


Рис. 4. Результати навчання

Використання згорткової нейронної моделі забезпечує вищу точність розпізнавання (див рис.3).

```
INIT_LR = 0.01  
  
#opt = SGD(lr=INIT_LR)  
  
opt = Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999, amsgrad=False)  
  
# model.fit(x_train, y_train, batch_size=128, epochs=20, verbose=0,  
validation_data=(X_test, Y_test), callbacks=[checkpointer])  
  
H = model.fit(trainX, trainY, validation_data=(testX, testY),
```

```
epochs=EPOCHS, batch_size=32,  
  
shuffle=True,  
  
callbacks=[checkpointer])
```

Рис 5. Фрагмент коду навчання згорткової нейронної мережі

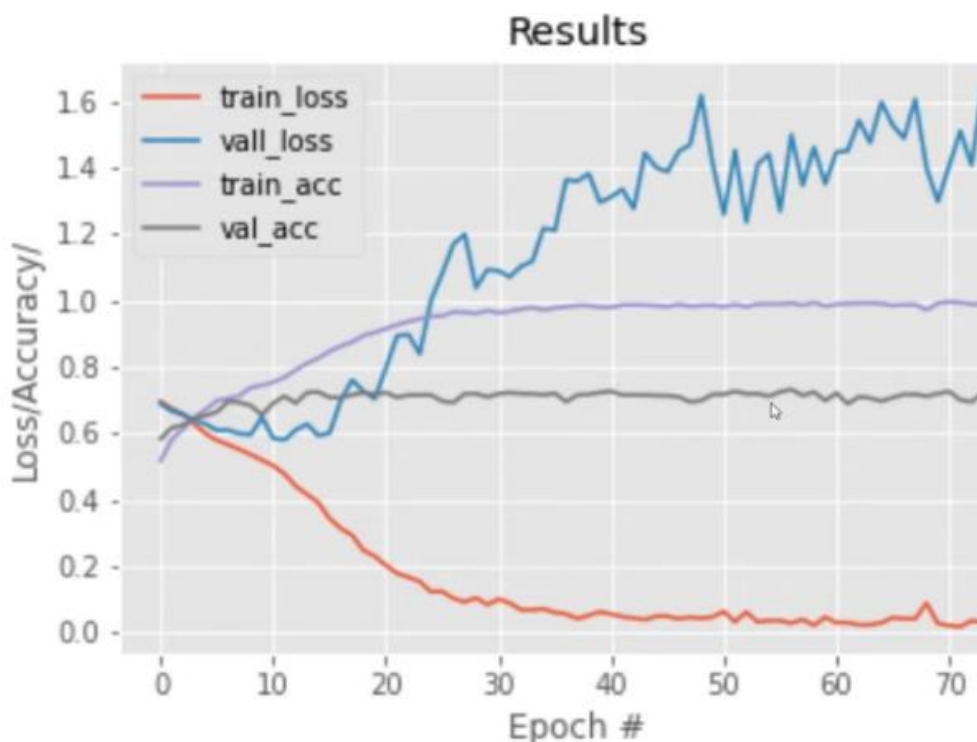


Рис. 6. Результати навчання

За даним алгоритмом, нейронна мережа повинна «тренуватись», співставляючись із вибіркою з тисячі малюнків номерних знаків. Якщо нейромережа коректно навчилася, вихідні дані ми отримуємо у вигляді символів номерного знаку.

Висновок. Задача розпізнавання – це складна завдання, яка вимагає багатьох етапів попередньої обробки зображень. Однак однією з ключових складнощ є баланс між часом обробки та точністю алгоритму розпізнавання. Недавні дослідження показали, що використання методів зниження розмірності простору ознак дозволяє підвищити середню точність класифікації з 87,7% до 90,2%.

Список використаних джерел:

1. Yilmaz K. Розумна гібридна система розпізнавання номерних знаків основана на обробці зображень використовуючи нейронну мережу та кореляцію зображення, (INISTA), Інновації в інтелектуальних системах та додатках 2011 Міжнародний симпозіум 18 Червня 2011 р.
2. Вікіпедія, Згортка (обробка зображень). URL: [https://uk.wikipedia.org/wiki/Згортка_\(обробка_зображень\)](https://uk.wikipedia.org/wiki/Згортка_(обробка_зображень))
3. Встановлення системи з розпізнаванням номерів. URL: <https://t-bezpeka.com/services/innovations/vstanovlennya-systemy-z-rozpiznavannyam-nomeriv/>.
4. wiki.uk-ua. 2021. URL: https://www.wiki.uk-ua.nina.az/Метод_зворотного_поширення_помилки.html