Дисципліна «Методи глибинного навчання»

Лабораторна робота №4

Тема: Розпізнавання двовимірних кольорових об’єктів за допомогою згорткової нейронної мережі

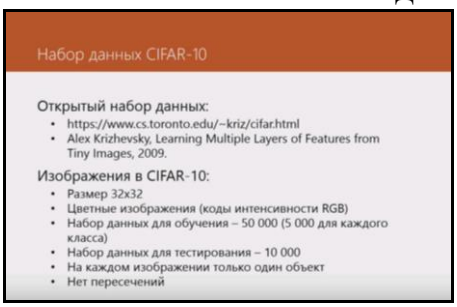
Мета: Отримання практичних навичок з розробки програмного забезпечення для реалізації згорткової нейронної мережі, призначеної для розпізнавання двовимірних кольорових об’єктів.

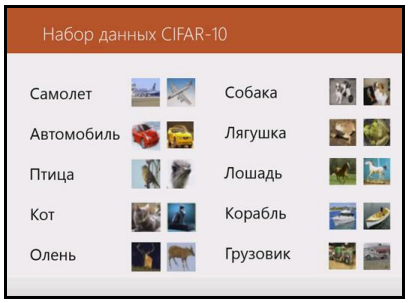
Теоретичні відомості

- Ознайомитись з матеріалами лекцій № 9-10.
- Ознайомитись з даними літературних джерел:

1. Кулаков Ю., Терейковська Л., Терейковський І. Спосіб застосування згорткової нейронної мережі для розпізнавання особи і емоцій користувача за клавіатурним почерком. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, Вип. 2 (38), 2019 р., С. 9-17.
DOI: https://doi.org/10.20535/2074-9481.2(38).2019.232654.
http://pnzzi.kpi.ua/article/view/232654
2. Терейковська, Л. (2020). Метод нейромережевого розпізнавання емоцій по зображенню обличчя. Комп’ютерно-інтегровані технології: освіта, наука, виробництво, (40), 146-152. https://doi.org/10.36910/6775-2524-0560-2020-40-22
http://cit-journal.com.ua/index.php/cit/article/view/171
3. Liudmyla Tereikovska, Ihor Tereikovskyi, Aiman Beketova, Gabit Karaman and Nadiia Makovetska. Recognition of Speaker’s Emotion by Squeezenet Convolutional Neural Network. Journal of Theoretical and Applied Information Technology. 15th March 2021. Vol.99. No 5. Pages 1139-1148.
http://www.jatit.org/volumes/Vol99No5/12Vol99No5.pdf

- Ознайомитись з базою даних CIFAR-10.





Хід виконання роботи:

1. Підготовка інструментальних засобів.

Для виконання лабораторної роботи рекомендується використовувати мову програмування Python (модуль numpy, бібліотеки Keras та TensorFlow), dataset cifar10. Комп’ютер має бути під’єднаний до мережі Internet.

2. Створення проєкту.

В лабораторній роботі використовується згорткова нейрона мережа, структура якої показана на рис. 1.

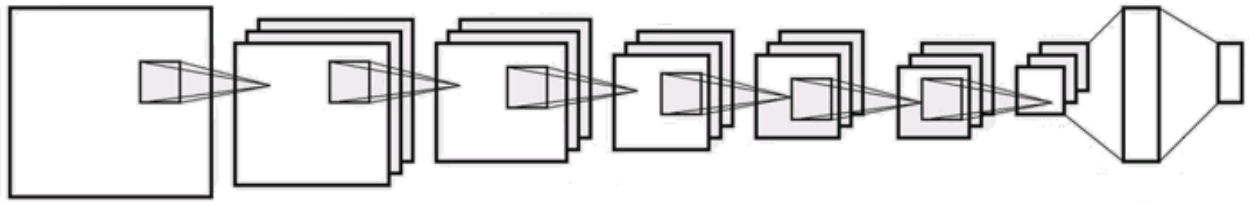


Рис. 2. Структура використаної згорткової нейронної мережі

2.1. Створити програму для реалізації навчання нейромережевої моделі (код наведений в додатку 1).

2.2. Створити програму для реалізації режиму розпізнавання нейромережевої моделі (код наведений в додатку 2).

3. Навчання нейромережевої моделі.

Запустити програму для навчання нейромережевої моделі. Зафіксувати термін та точність навченої нейромережевої моделі.

4. Тестування нейромережевої моделі.

Запустити програму для розпізнавання. Дослідити можливості навченої моделі в аспекті розпізнавання різних зображень.

5. Оформити звіт з лабораторної роботи.

Додаток 1

```
import numpy
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Flatten, Activation
from keras.layers import Dropout
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.utils import np_utils
from keras.optimizers import SGD

numpy.random.seed(42)

# Завантаження даних
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
# Розмір міні-вибірки
batch_size = 32
# Кількість класів зображень
nb_classes = 10
# Кількість епох навчання
nb_epoch = 25
# Розмір зображення
img_rows, img_cols = 32, 32
# Кількість каналів: RGB
img_channels = 3

# Нормалізація даних
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255

Y_train = np_utils.to_categorical(y_train, nb_classes)
Y_test = np_utils.to_categorical(y_test, nb_classes)

# Створення нейромережевої моделі
model = Sequential()
# Перший шар згортки
model.add(Conv2D(32, (3, 3), padding='same',
                 input_shape=(32, 32, 3), activation='relu'))
# Другий шар згортки
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
# Перший шар субдискретизації
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
# Перший шар Dropout
model.add(Dropout(0.25))

# Третій шар згортки
model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
# Четвертий шар згортки
model.add(Conv2D(64, (3, 3), activation='relu'))
# Другий шар субдисктеризації
model.add(MaxPooling2D(pool_size=(2, 2)))
# Другий шар Dropout
model.add(Dropout(0.25))
# Шар перетворення вхідних даних
model.add(Flatten())
# Повнозв'язний шар
model.add(Dense(512, activation='relu'))
# Третій шар Dropout
model.add(Dropout(0.5))
# Вихідний шар
model.add(Dense(nb_classes, activation='softmax'))

# Параметри оптимізації
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy',
              optimizer=sgd,
              metrics=['accuracy'])
# Навчання  моделі
model.fit(X_train, Y_train,
              batch_size=batch_size,
              epochs=nb_epoch,
              validation_split=0.1,
              shuffle=True,
              verbose=2)

# Оцінка якості навчання на тестових даних
scores = model.evaluate(X_test, Y_test, verbose=0)
print("Accuracy on test data: %.2f%%" % (scores[1]*100))
# Збереження моделі
model.save('my_model.h5')
```

Додаток 2

```
import numpy
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Flatten, Activation
from keras.layers import Dropout
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.utils import np_utils
from keras.optimizers import SGD
from keras.models import load_model

model = load_model('my_model.h5')
# x - назва файлу з попередньо обробленим зображенням
# для обробки слід використовувати засоби модулю numpy
model.predict(x)
```