РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ ЗА ДОПОМОГОЮ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Велещук Олександр Іванович
магістрант спеціальності «Середня освіта. Інформатика»,
Тернопільський національний педагогічний університет імені Володимира Гнатюка
velsashok@gmail.com

Карабін Оксана Йосифівна
кандидат педагогічних наук,
доцент кафедри інформатики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка
karabinoksana@gmail.com

Державні структури, бізнес, наукові та навчальні організації використовують велику кількість паперових документів, частина яких є рукописними. Тому існує потреба в оцифруванні таких документів, аналізі і обробці інформації яка в них міститься.

Розпізнавання рукописного тексту відноситься до задачі оптичного розпізнавання символів (англ. optical character recognition, OCR). Отриману текстову інформацію використовують для: пошуку по ключових словах, перекладу або озвучення тексту, збереження інформації в компактнішій формі.

Мета дослідження полягає у розробці веб-додатку для розпізнавання рукописних цифр. Під час розробки додатку здійснювалося порівняння трьох алгоритмів класифікації: k-найближчих сусідів (kNN), випадкового лісу (англ. random forest) і опорних векторів (англ. support vector machine) та підбір оптимальних параметрів для цих алгоритмів на наборі даних MNIST.

Оптичне розпізнавання символів відбувається у декілька етапів: сегментація тексту - розбиття на рядки, слова і окремі символи; класифікація зображення окремих символів передаються алгоритму класифікації який здійснює розпізнавання. Щоб підвищити точність підключаються словники і при спробі розпізнати наступний символ враховується те, як були розпізнанні попередні.

Для програмної реалізації використано мову програмування Python, оскільки для Python існує велика кількість бібліотек для аналізу даних і машинного навчання. Серед них було вибрано бібліотеку **scikit-learn** в якій реалізовано основні алгоритми машинного навчання

Для навчання алгоритмів використовувалася база даних MNIST (скорочення від Mixed National Institute of Standards and Technology) — база даних зразків рукописного написання цифр, яка містить 60000 зображень для навчання і 10000 зображень для тестування.

Для методу випадкового лісу здійснювався підбір параметру n_estimators, який задає кількість дерев рішень у лісі. Було створено список parameters із набором значень, серед яких потрібно було вибрати найкраще. Доля правильних відповідей для кожного значення параметру обчислювалася за допомогою перехресної перевірки із розбиттям на п’ять частин.

```
from sklearn.model_selection import cross_val_score
```

```
from sklearn.ensemble import RandomForestClassifier

parameters = [100, 200, 300, 400]
scores = []
for parameter in parameters:
    clf = RandomForestClassifier(n_estimators=parameter, n_jobs=-1, verbose=True)
    cv_scores = cross_val_score(clf, X_train, y_train, cv=5, scoring='accuracy')
    scores.append(cv_scores.mean())
    print(scores[-1])
```

Рис. 1. Фрагмент коду для підбору параметру n_estimators алгоритму «Випадковий ліс»

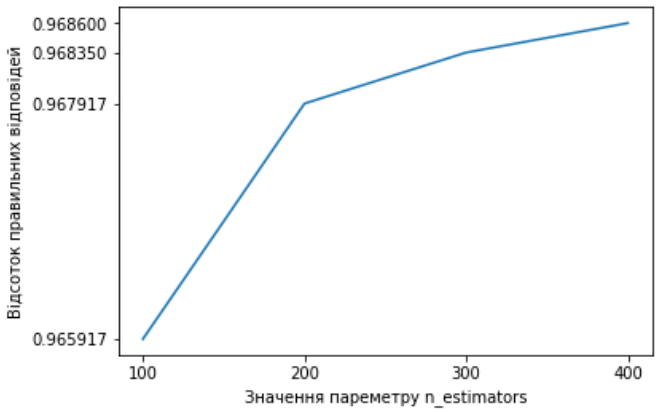

Рис. 2. Графік залежності частки правильних відповідей від параметру n_estimators

На рисунку 2 можна побачити, що із збільшенням числа n_estimators точність теж зростає. Проте варто зауважити, що збільшення n_estimators із 300 до 400 підвищить точність лише на ≈0.0003, але при цьому суттєво погіршить швидкодію.

Після підбору параметрів і навчання моделі для кожного алгоритму була побудована матриця помилок. Матрицю помилок для алгоритму «Випадковий Ліс» можна побачити на рисунку 3. Рядки і стовпці цієї матриці позначені числами від 0 до 9. На перетині *i*-го рядка та *j*-го стовпця стоїть число об'єктів які насправді належать до класу і, але алгоритм відніс їх класу *j*. Наприклад, на перетині рядка із індексом 2 і стовпця з індексом 0 стоїть число 5. Це означає, що 5 об'єктів які насправді були двійками алгоритм класифікував як 0. По діагоналі розташовані кількості правильно класифікованих об'єктів.

Для створення матриці помилок була використана функція confusion_matrix із модуля sklearn.metrics.

```
clf = RandomForestClassifier(n_estimators=300, n_jobs=-1, verbose=True)
y_pred = clf.fit(X_train, y_train)
cnf_matrix = confusion_matrix(y_test, y_pred)
```

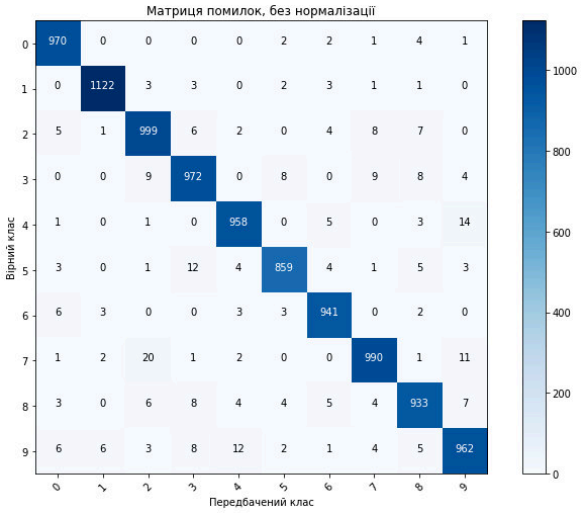

Рис. 3. Матриця помилок для алгоритму «Випадковий Ліс»

У результаті роботи було здійснено підбір оптимальних параметрів алгоритмів k-найближчих сусідів, випадкового лісу і опорних векторів.

Для алгоритму випадкового лісу здійснювався підбір параметру n_estimators, який задає кількість дерев рішень у лісі. Найкращим компромісом між швидкістю роботи і якістю виявилося значення 300.

Для алгоритму k найближчих сусідів найкращими виявилися такі параметри: n_neighbors = 4, weights = «distance». Параметр n_neighbors задає кількість сусідів, а weights = «distance» вказує на те, що вклад сусіда в результат класифікації буде обернено пропорційний відстані до нього.

Зважаючи на велику кількість даних, для алгоритму опорних векторів було вибране лінійне ядро, яке є найефективнішим з точки зору складності обчислень. Для цього алгоритму здійснювався підбір параметру C, який регулює баланс між шириною розділювальної смуги і кількістю об’єктів які її порушують. Найкращим виявилося значення параметра C = 0.001.

За допомогою бібліотеки matplotlib для кожного алгоритму побудовано графіки залежності долі правильних відповідей від значення параметру. Для аналізу якості роботи алгоритмів було побудовано матриці помилок. Після підбору параметрів для кожного алгоритму була обчислена доля правильних відповідей на 10 000 зображеннях для перевірки із набору даних MNIST. Ось результати цієї перевірки: випадковий ліс: 0.9706; K найближчих сусідів: 0.9714; алгоритм опорних векторів із лінійним ядром: 0.9087. Розроблено веб-додаток для розпізнавання рукописних цифр введених користувачем. В додатку реалізовано можливість вибору алгоритму який буде використовуватися для розпізнавання.

Список використаних джерел:

1. A function to load numpy arrays from the MNIST data files. URL: https://gist.github. com/tylerneylon/ce60e8a06e7506ac45788443f7269e40. (дата звернення: 07.04.2018)

2. Hello World - Machine Learning Recipes #1. URL: https://youtu.be/cKxRvEZd3Mw. (дата звернення: 01.04.2018)

3. K-Nearest Neighbor Classification MNIST Handwritten Digits. URL: https://youtu. be/ooQtUaCExa8. (дата звернення: 02.04.2018)

4. Machine learning in Python with scikit-learn. URL: https://www.youtube. com/playlist?list=PL5-da3qGB5ICeMbQuqbbCOQWcS6OYBr5A. (дата звернення: 12.04.2018)

5. Machine Learning with Python - from Model to Web-Service by Aleksei Tiulpin. URL: https://youtu.be/9sFUlR-CS7Y. (дата звернення: 03.04.2018)

6. Support Vector Machine Learning MNIST Handwritten Digits. URL: https://youtu.be/s8q_OQBJpwU. (дата звернення: 02.04.2018)

7. Введение в машинное обучение. URL: https://www.coursera.org/learn/vvedenie-mashinnoe-obuchenie. (дата звернення: 02.04.2018)

8. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. *Концепции, инструменты и техники для создания интеллектуальных систем*: пер. з англ. Москва.: вид. дім «Вільямс», 2018. 688 с.

9. Метод опорных векторов Лекция № 7 курса «Алгоритмы для Интернета». URL: http://yury.name/internet/07ianote.pdf. (дата звернення: 06.05.2018).

10. Соколов Е. А. Лекция 7. Решающие деревья. URL: https://github.com/esokolov/ ml-course-hse /blob/master/2017-fall/lecture-notes/lecture07-trees.pdf. (дата звернення: 10.04.2018)

АКТУАЛЬНІ ПИТАННЯ СИСТЕМИ ПРОФЕСІЙНОЇ ПІДГОТОВКИ WEB-ДИЗАЙНЕРІВ В РАМКАХ СУЧАСНИХ ТЕНДЕНЦІЙ ПРОЕКТУВАННЯ ІНТЕРНЕТ РЕСУРСІВ

Вельгач Андрій Володимирович
кандидат фізико-математичних наук,
викладач кафедри інформатики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка
velgandr@fizmat.tnpu.edu.ua

На ринку освітніх послуг присутній великий попит на навчання основам web-дизайну. На сьогоднішній день немає встановленого розуміння того, яке місце в професійній підготовці з різних спеціальностей має займати навчання web-дизайну; до цих пір не виявлено структури і змісту інформаційно-професійної компетентності сучасних фахівців в системі вищої освіти; не розроблені ефективні педагогічні технології навчання web-дизайну у закладах вищої освіти (ЗВО). Аналіз науково-педагогічної літератури та дисертаційних досліджень [1-4] показав, що проблема організації і розробки інтерактивних технологій інформаційно-орієнтованої підготовки web-дизайнерів в системі вищої освіти, без сумніву, є актуальною і недостатньо розробленою.

У сучасному суспільстві існує об’єктивна потреба в кваліфікованих кадрах, здатних реалізувати потенціал сучасних інтернет-технологій для вирішення різних професійних завдань. Одним з ключових фахівців, які беруть участь в проектуванні інтернет-ресурсів (веб-сайтів) є веб-дизайнер. Аналіз професійних завдань фахівців, які беруть участь в роботі над сайтом, дозволив уточнити поняття веб-дизайну, під яким розумітимемо галузь веб-розробки, спрямовану на графічне оформлення і проектування призначеного для користувача інтерфейсу з метою забезпечення високих споживчих властивостей і естетичних якостей інтернет-ресурсу. Виділення в напрямку підготовки «веб-дизайн» вузьких спеціалізацій відповідно до розв’язуваних професійних завдань того чи іншого учасника команди веб-розробників (візуальний дизайнер, верстальник, фахівець з