

УДК 004.89

DOI <https://doi.org/10.32838/2663-5941/2021.6/17>**Мозговенко А.А.**

Таврійський державний агротехнологічний університет імені Дмитра Моторного

**Зінов'єва О.Г.**

Таврійський державний агротехнологічний університет імені Дмитра Моторного

**Гешева Г.В.**

Таврійський державний агротехнологічний університет імені Дмитра Моторного

## РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ КЛАСИФІКАЦІЇ НАВЧАЛЬНИХ ТЕКСТІВ ДИСЦИПЛІН З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

У статті розглядається програмний засіб класифікації навчальних текстів дисциплін з урахуванням нейронних мереж. Нейронні мережі дають багато можливостей і можуть відігравати суттєву роль у процесі навчання.

Текст становить собою вияв мовної діяльності людини, її результат, відбиваючи особливості комунікативного процесу на певному етапі суспільного розвитку. Автор, створюючи текст, розраховує на певне його сприймання й розуміння реципієнтом, зворотну реакцію – як безпосередню, так і опосередковану, залежно від обставин спілкування. Саме сприймання та розуміння тексту, а також здатність відреагувати на його зміст трактують як дискурс. Проте розуміння тексту з певної дисципліни може викликати складнощі у викладача, який не розбирається в предметі. Тому виникає необхідність у розробленні програмного модуля, який допоможе класифікувати текст по певних категоріях.

Нейронні мережі нині забезпечують найкращі рішення багатьох проблем у розпізнаванні зображень, розпізнаванні мовлення, обробленні природної мови та багатьох інших сферах.

З природної точки зору нейронні мережі – це звичайне уявлення про мозок: нейрони, взаємопов'язані з іншими нейронами, які утворюють мережу. Проста інформація проходить у багатьох із них, перш ніж стати справжньою річчю, як-от «порушити рукою, щоб взяти цей олівець».

Нейронна мережа – це система, яка вчиться робити прогнози, виконуючи такі кроки:

- Взяття вхідних даних;
- Складання прогнозу;
- Порівняння прогнозу з бажаним результатом;
- Налаштування внутрішнього стану для правильного прогнозування наступного разу.

Вектори, шари та лінійна регресія є одними з будівельних блоків нейронних мереж. Дані зберігаються у вигляді векторів, а за допомогою Python ви зберігаєте ці вектори в масивах. Кожен шар перетворює дані, які надходять із попереднього шару. Ви можете розглядати кожен шар як крок розробки функцій, оскільки кожен шар витягує деяке представлення даних, які були отримані раніше.

Одна цікава особливість шарів нейронної мережі полягає в тому, що ті самі обчислення можуть витягувати інформацію з будь-яких даних. Це означає, що не має значення, чи використовуєте ви графічні дані чи текстові дані. Процес отримання значущої інформації та навчання моделі глибокого навчання однаковий для обох сценаріїв.

**Ключові слова:** нейронні мережі, Python, об'єктно-орієнтоване програмування, TensorFlow, Qt Designer, машинне навчання.

**Постановка проблеми.** Розуміння тексту з певної дисципліни може викликати складнощі у викладача, який не розбирається в предметі. Тому виникає необхідність у розробленні програмного модуля, який допоможе класифікувати текст за певними категоріями.

Реалізацію програмного модуля пропонується провести за допомогою нейронних мереж. Такі мережі поступально покращують свою продук-

тивність та, якщо розглядати приклади, загалом без спеціального програмування під задачу.

**Аналіз останніх досліджень і публікацій.** Загальне захоплення нейромережевими технологіями і глибинним навчанням не минуло і комп'ютерну лінгвістику – автоматичну обробку текстів на природній мові. На недавніх конференціях асоціації комп'ютерної лінгвістики ACL, головному науковому форумі в цій області,

переважну більшість доповідей було присвячено застосуванню нейронних мереж як для вирішення вже відомих задач, так і для дослідження нових, які не вирішувалися за допомогою стандартних засобів машинного навчання. Підвищена увага лінгвістів до нейронних мереж зумовлена декількома причинами. Застосування нейронних мереж, по-перше, істотно підвищує якість рішення багатьох стандартних задач класифікації текстів та послідовностей, по-друге, знижує трудомісткість під час роботи безпосередньо з текстами, по-третє, дозволяє вирішувати нові завдання (наприклад, створювати чат-боти). Водночас нейронні мережі можна вважати повністю самостійним механізмом розв'язання лінгвістичних проблем.

**Постановка завдання.** Метою роботи є розроблення програмного модуля з класифікації навчальних текстів дисциплін з використанням нейронних мереж.

Даний програмний модуль має можливості подальшого вдосконалення:

- Поліпшення інтерфейсу;
- Збільшення кількості категорій;
- Застосування моделей нейронних мереж.

**Виклад основного матеріалу дослідження.** З формальної точки зору нейронна мережа являє собою спрямований граф заданої архітектури, вершини або вузли якого називаються нейронами. На першому рівні графа знаходяться вхідні вузли, на останньому – вихідні вузли, число яких залежить від завдання. Наприклад, для класифікації на два класи на вихідний рівень мережі можна помістити один або два нейрони, для класифікації на  $k$  класів –  $k$  нейронів. Всі інші рівні у графі нейронної мережі прийнято називати прихованими шарами. Всі нейрони, що знаходяться на одному рівні, пов'язані ребрами з усіма нейронами наступного рівня, кожне ребро має вагу. Кожному нейрону ставиться у відповідність функція активації, що моделює роботу біологічних нейронів: вони «мовчать», коли вхідний сигнал слабкий, а коли його значення перевищує певний поріг, спрацьовують і передають вхідні значення далі по мережі. Завдання навчання нейронної мережі на прикладах (тобто на парах «об'єкт – правильна відповідь») полягає в пошуку ваг ребер, найкращим чином пророкують правильні відповіді. Ясно, що саме архітектура – топологія будови графа нейронної мережі – є її найважливішим параметром. Хоча формального визначення для «глибинних мереж» поки немає, прийнято вважати глибинними все нейронні мережі, що складаються з великого числа шарів або мають «нестандартні» шари (наприклад,

що містять тільки обрані зв'язки або використовують рекурсію з іншими верствами).

Прикладом найбільш успішного застосування нейронних мереж поки є аналіз зображень, однак нейромережеві технології докорінно змінили і роботу з текстовими даними. Якщо раніше кожен елемент тексту (літеру, слово або пропозицію) потрібно було описувати за допомогою безлічі ознак різної природи (морфологічних, синтаксичних, семантичних і т. д.), то тепер у багатьох задачах необхідність у складних описах пропадає. Теоретики і практики нейромережевих технологій часто говорять про «навчання поданням» (representation learning) – у сирому тексті, розбитому тільки на слова і пропозиції, нейронна мережа здатна знайти залежності й закономірності та самостійно скласти простору ознак. На жаль, у такому просторі людина нічого не зрозуміє – під час навчання нейронна мережа ставить кожному елементу тексту у відповідність один щільний вектор, що складається з деяких чисел, які представляють виявлені «глибинні» взаємозв'язки. Акцент під час роботи з текстом зміщується від конструювання підмножини ознак і пошуку зовнішніх баз знань до вибору джерел даних і розмітки текстів для подальшого навчання нейронної мережі, для якого потрібно значно більше даних у порівнянні зі стандартними методами. Саме через необхідність використовувати великі обсяги даних і через слабкість інтерпретування і непередбачуваності нейронні мережі не затребувані в реальних додатках промислового масштабу на відміну від інших, добре зарекомендували себе алгоритми навчання, такі як випадковий ліс і машини опорних векторів.

Завдання класифікації текстів – одне з найактуальніших для маркетологів, особливо коли йдеться про аналіз думок або відношення споживача до якогось товару чи послуги, тому дослідники постійно працюють над підвищенням якості його вирішення. Однак аналіз думок є завданням класифікації швидше пропозицій, а не текстів – у позитивному відкликанні користувач може написати одну-дві негативно забарвлених пропозиції, і їх теж важливо вміти визначати і аналізувати. Відомі труднощі в класифікації пропозицій полягають у змінній довжині входу – оскільки пропозиції в текстах бувають довільної довжини, незрозуміло, як подати їх на вхід нейронної мережі. Один із підходів, запозичений з області аналізу зображень, полягає у використанні згортальних нейронних мереж.

Процес класифікації можна проводити за допомогою нейронних мереж. Штучні нейронні

мережі – це обчислювальні системи, натхненні біологічними нейронними мережами, що становлять мозок тварин. Такі системи навчають задачі (поступально покращують свою продуктивність на них) загалом без спеціального програмування під задачу. Нейронна мережа – це мережа простих елементів, званих нейронами, які отримують вхід, змінюють свій внутрішній стан (збудження) відповідно до цього входу і виробляють вихід, залежний від входу та збудження.

Однією з особливостей нейронних мереж є їх здатність до навчання. У широкому розумінні процес навчання – це адаптація параметрів та архітектури мережі для рішення поставленої задачі шляхом оптимізації вибраного критерію якості [1]. Найчастіше до параметрів, які адаптуються, відносяться вагові коефіцієнти нейронів, рідше – архітектура мережі та параметри функції активації. Всі алгоритми навчання можна розділити на два великі класи: детерміновані та стохастичні. Для алгоритмів першого класу характерним є наявність жорстких правил дій за адаптації параметрів, а другому притаманні дії, які підкоряються деякому випадковому процесу [2].

Розвиток Інтернету та процеси глобалізації сприяли тому, що з'явилося дуже багато інформації, опрацювати яку самотужки людина фізично не в змозі. Нейронні мережі знайшли застосування у:

- аналізі та класифікуванні даних за заданими параметрами;
- формуванні аналітичних прогнозів з огляду на вхідну інформацію;
- порівнянні та розпізнаванні ідентичних даних.

Нейронні мережі найбільш зручно створювати за допомогою спеціальних пакетів. Найбільш популярні пакети – це TensorFlow, PyTorch, Microsoft Cognitive Toolkit, Theano.

TensorFlow – відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявлення та розшифрування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди. Її наразі застосовують як для досліджень, так і для розроблення продуктів Google, часто замінюючи на його ролі її закритого попередника, DistBelief. TensorFlow було початково розроблено командою Google Brain для внутрішнього використання в Google, поки її не було випущено під відкритою ліцензією Apache 2.0 9 листопада 2015 року.

Платформа спочатку розроблена командою Google Brain і використовуються в сервісах Google

для розпізнавання мови, виділення облич на фотографіях, визначення схожості зображень, відсіювання спаму в Gmail, підбору новин у Google News і організації перекладу з урахуванням сенсу. Розподілені системи машинного навчання можна створювати на типовому обладнанні завдяки вбудованій підтримці в TensorFlow рознесення обчислень на кілька CPU або GPU.

Серед застосувань, для яких TensorFlow є основою, є програмне забезпечення автоматизованого опису зображень, таке як DeepDream. 26 жовтня 2015 року Google офіційно реалізувала RankBrain, який підтримує TensorFlow. RankBrain тепер обробляє суттєве число пошукових записів, замінюючи та доповнюючи традиційні статичні алгоритми на основі результатів пошуку.

Іншим застосуванням є використання у складі програм FakeApp з метою безшовного поєднання фото- та відеозображень для створення докладних, але правдоподібних відео, відомих під назвою Deepfake.

TensorFlow надає бібліотеку готових алгоритмів чисельних обчислень, реалізованих через графи потоків даних (data flow graphs). Вузли в таких графах реалізують математичні операції або точки входу/виводу, тоді як ребра графа представляють багатовимірні масиви даних (тензори), які перетікають між вузлами. Вузли можуть бути закріплені за обчислювальними пристроями і виконуватися асинхронно, паралельно обробляючи разом усі підходящі до них тензори, що дозволяє організувати одночасну роботу вузлів у нейронній мережі за аналогією з одночасною активацією нейронів у мозку.

PyTorch – відкрита бібліотека машинного навчання на основі бібліотеки Torch, що використовують для таких застосувань, як комп'ютерне бачення та обробка природної мови. Розробляє її переважно група дослідження штучного інтелекту компанії Facebook. Вона є вільним та відкритим програмним забезпеченням, що випускають під ліцензією Modified BSD. І хоча інтерфейс Python є більш відшліфованим і головним зосередженням розробки, PyTorch також має зовнішній інтерфейс і для C++. Крім того, програмне забезпечення ймовірнісної мови програмування Pyro компанії Uber використовує PyTorch як внутрішній інтерфейс.

Microsoft Cognitive Toolkit, раніше відома як CNTK й іноді звана The Microsoft Cognitive Toolkit – це платформа глибокого навчання, розроблена Microsoft Research. Microsoft Cognitive Toolkit описує нейронні мережі як низку обчислювальних кроків через орієнтований граф.

Theano – бібліотека чисельного обчислення для Python. Обчислення в Theano виражаються NumPy-ським синтаксисом і компілюються для ефективного виконання на архітектурі або ЦП, або ГП.

Ураховуючи результати досліджень, викладені вище, і вимоги до програмних навчальних систем, такі як можливість вибору дисциплін, введення тексту та отримання результатів, метою роботи є створення програмного модуля з класифікації навчальних текстів дисциплін з використанням нейронних мереж.

З метою розроблення якісного програмного модуля з класифікації навчальних текстів дисциплін з використанням нейронних мереж вивчено закономірності процесу проектування та роботи нейронних мереж, які дозволяють ефективно використовувати їх у процесі класифікації тексту дисципліни.

Для досягнення поставленої мети в роботі необхідно вирішити такі задачі:

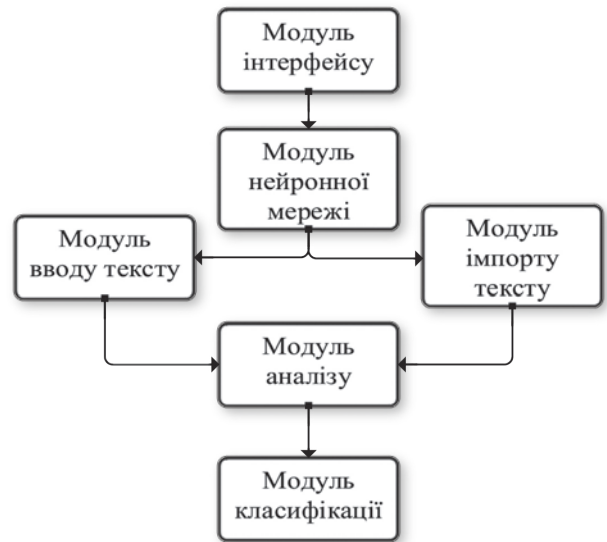
- вивчити і проаналізувати методики побудови нейронних мереж;
- проаналізувати існуючий інструментарій для створення нейронних мереж;
- розробити структурну схему програмної системи;
- розробити програмне забезпечення з класифікації навчальних текстів дисциплін із використанням нейронних мереж.

Важливою властивістю нейронних мереж, що свідчить про їхній великий потенціал і широкі прикладні можливості, є паралельна обробка інформації одночасно великою кількістю нейронів. Завдяки цьому досягається значне пришвидшення обробки інформації. Іншою, не менш важливою, особливістю нейронних мереж є здатність до навчання та узагальнення інформації.

Під час робіт із програмним модулем із класифікації навчальних текстів дисциплін з використанням нейронних мереж, який представлений у вигляді віконної програми, основним джерелом інформації є зображення на екрані монітора. Мета роботи користувача з програмним модулем полягає у прискоренні класифікації та обробки текстів з навчальних дисциплін.

Для забезпечення виконання всіх етапів процесу класифікації в складі програмного модуля з класифікації навчальних текстів дисциплін з використанням нейронних мереж ураховані функціональні складові елементи, взаємозв'язок яких між собою показаний на рис. 1.

Модуль інтерфейсу завантажує графічний інтерфейс користувача для подальшої роботи.



**Рис. 1.** Структурна схема програмного модуля з класифікації навчальних текстів дисциплін з використанням нейронних мереж

Модуль нейронної мережі завантажує вже навчену нейронну модель з конкретних дисциплін.

Модуль вводу тексту та модуль імпорту тексту відповідають за внесення тексту до програмного модуля.

Модуль аналізу вносить текст до моделі мережі та відповідає за обробку та перетворення тексту.

Модуль класифікації обробляє результат аналізу та виводить його у форму користувача.

Програмний модуль складається з графічного інтерфейсу, розробленого в середовищі Qt Designer, нейронної мережі з пакету TensorFlow та скрипту, написаного на мові програмування Python.

Для розроблення графічного інтерфейсу користувача використовувалося середовище розробки Qt Designer. Qt Designer дозволяє створювати графічні інтерфейси користувача за допомогою низки інструментів. Існує панель інструментів «Панель віджетів», у якій доступні для використання елементи інтерфейсу – віджети, такі як, наприклад, «випадає» ComboBox, «поле введення» LineEdit, «кнопка» PushButton і багато інших. Кожен віджет має свій набір властивостей, що визначається відповідним йому класом бібліотеки Qt. Властивості віджета можуть бути змінені за допомогою «Редактора властивостей». Для кожного класу властивостей віджета існує свій спеціалізований редактор. Характерною особливістю Qt Designer є підтримка візуального редагування сигналів і слотів. Так, наприклад, можна зв'язати сигнал, що генерується з переключення стану віджета CheckBox, зі слотом, що відповідає за доступність іншого віджета.

Для створення та роботи з нейронними мережами використовувався пакет TensorFlow. TensorFlow надає бібліотеку готових алгоритмів чисельних обчислень, реалізованих через графи потоків даних (data flow graphs). Вузли в таких графах реалізують математичні операції або точки входу/виводу, тоді як ребра графа представляють багатовимірні масиви даних (тензори), які перетікають між вузлами. Вузли можуть бути закріплені за обчислювальними пристроями і виконуватися асинхронно, паралельно обробляючи разом усі підходящі до них тензори, що дозволяє організувати одночасну роботу вузлів у нейронній мережі за аналогією з одночасною активацією нейронів у мозку.

Для написання функціоналу програмного модуля було вибрано мову програмування Python. Python – інтерпретована об’єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Структури даних високого рівня разом із динамічною семантикою та динамічним зв’язуванням роблять її привабливою для швидкого розроблення програм, а також як засіб поєднання наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. У мові програмування Python підтримується кілька парадигм програмування, зокрема: об’єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Багата стандартна бібліотека є однією із привабливих сторін Python. Тут є засоби для роботи з багатьма мережевими протоколами та форматами Інтернету, наприклад, модулі для написання HTTP-серверів та клієнтів, для розбору та створення поштових повідомлень, для роботи з XML тощо. Набір модулів для роботи з операційною системою дозволяє писати крос-платформні застосунки. Існують модулі для роботи з регулярними виразами, текстовими кодуваннями, мультимедійними форматами, криптографічними протоколами, архівами, серіалізацією даних, юніт-тестуванням та ін.

Архітектура програмного забезпечення (англ. Software architecture) – це структура програми або обчислювальної системи, яка включає програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними.

Під час робіт із програмним модулем із класифікації навчальних текстів дисциплін з викорис-

танням нейронних мереж, який представлений у вигляді віконної програми, основним джерелом інформації є зображення на екрані монітора.

Для забезпечення виконання всіх етапів процесу класифікації в складі програмного модуля з класифікації навчальних текстів дисциплін з використанням нейронних мереж ураховані функціональні складові елементи, взаємозв’язок яких між собою показаний на рис. 2.



Рис. 2. Структурна схема архітектури програмного модуля

Інтерфейс відповідає за графічне відображення роботи програмного модуля.

Текст представляє собою засоби вводу та імпорту тексту до програмного модуля.

Нейронна мережа – це навчений у пакеті TensorFlow на мові програмування Python модуль, який додає функціонал із класифікації тексту до програмного модуля.

Оброблення і вивід надає змогу користувачу оцінити результати роботи нейронної мережі.

Програмна частина надає функціонал та об’єднує інші частини для коректної роботи всього програмного модуля.

Програмний модуль має віконний інтерфейс, розроблений за допомогою інструментарію Qt Designer і мови програмування Python. Інтерфейс приймає вхідні дані за допомогою таких пристроїв, як комп’ютерна клавіатура та миша й забезпечує графічний висновок на моніторі комп’ютера. У GUI-дизайні широко використовуються, як мінімум, два різні принципи: об’єктно-орієнтовані інтерфейси користувача (ООUI) й інтерфейси, орієнтовані на додатки.

Вікно форми має фіксований розмір і дорівнює 700\*500 пікс, а також включена можливість зміни розміру вікна.

Головна форма модуля (рис. 3) складається з трьох областей:

- введення тексту;
- завантаження нейронної мережі;
- вивід результату.



Рис. 3. Інтерфейс користувача

Для запуску навчальної системи необхідно запустити файл PyText.exe, який знаходиться в головному каталозі. Після запуску файлу «PyText.exe» на екрані з'являється вікно програмного модуля (рис. 4).



Рис. 4. Інтерфейс програмного модуля

Після запуску користувач вводить або імпортує текст до області вводу та натискає кнопку Generate. Після чого текст передається до нейронної моделі, проводиться оброблення, пошук ключових слів та прогнозування.

Після оброблення тексту результати роботи передаються до області виводу. Результати роботи програми надані на рисунку 6.

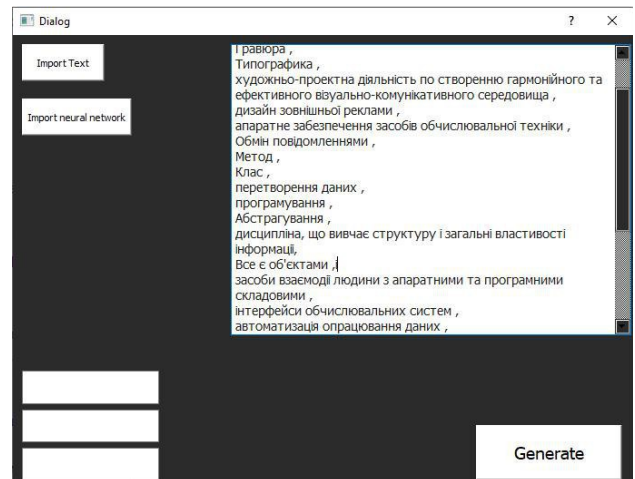


Рис. 5. Інтерфейс програмного модуля з введеним текстом

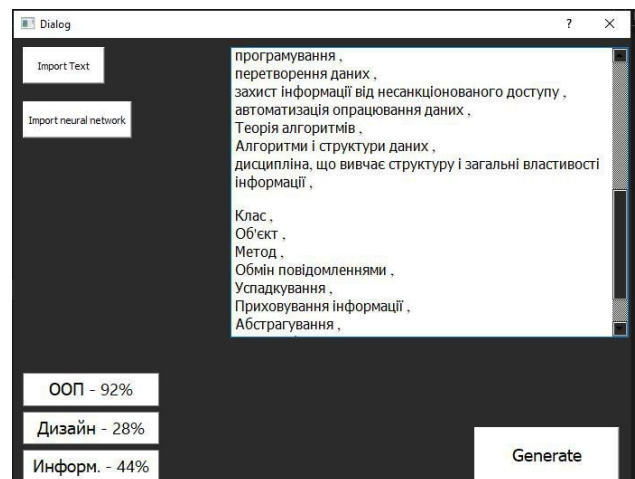


Рис. 6. Інтерфейс програмного модуля з виведеними результатами

**Висновки.** У роботі розглядається можливість розроблення програмного модуля з класифікації навчальних текстів дисциплін з використанням нейронних мереж.

Запропонований програмний модуль із класифікації навчальних текстів дисциплін з використанням нейронних мереж має можливості подальшого вдосконалення за рахунок поліпшення інтерфейсу, збільшення кількості категорій та застосування вбудованих моделей нейронних мереж.

#### Список літератури:

1. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применения. Харьков : ТЕЛТЕХ, 2004. 369 с.
2. Комашинский В.И., Смирнов Д.А. Нейронные сети и их применение в системах управления и святы. Москва : Горячая Линия – Телеком, 2003. 98 с.
3. Лемешевский С.В. Практическое применение численных методов на языке Python. Москва : Диалектика, 2016. 102 с.

4. Прохоронюк Н.А. Python Самое необходимое. Санкт Петербург : «БХВ-Петербург». 2016. 462 с.
5. Седжвик Р., Уэйн К, Дондеро Р. Программирование на языке Python : учебный курс. Москва : Диалектика, 2017. 736 с.
6. Субботін С.О., Олійник А.О., Олійник О.О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей : монографія ; під заг. ред. С.О. Субботіна. Запоріжжя : ЗНТУ, 2009. 375 с.
7. Федоров Д.Ю. Основы программирования на примере языка Python. СПбГЭУ, 2016. 154 с.
8. Яхьева Г.Э. Нечеткие множества и нейронные сети : учебное пособие. Москва : Интернет – Университет Информационных Технологий ; БИНОМ. Лаборатория знаний, 2006. 316 с.

**Mozgovenko A.A., Zinovieva O.G., Heshcheva H.V. DEVELOPMENT OF SOFTWARE FOR CLASSIFICATION OF EDUCATIONAL TEXTS OF DISCIPLINES USING NEURAL NETWORKS**

*The article considers the software tool for the classification of educational texts of disciplines taking into account neural networks. Neural networks provide many opportunities and can play a significant role in the learning process.*

*The text is a manifestation of human language activity, its result, reflecting the peculiarities of the communicative process at a certain stage of social development. The author, creating the text, counts on a certain perception and understanding by the recipient, the feedback - both direct and indirect, depending on the circumstances of communication. Perception and understanding of the text, as well as the ability to respond to its content, are interpreted as a discourse. However, understanding a text in a particular discipline can cause difficulties for a teacher who does not understand the subject. Therefore, there is a need to develop a software module that will help classify the text into certain categories.*

*Neural networks now provide the best solutions to many problems in image recognition, speech recognition, natural language processing, and many other areas.*

*By nature, neural networks are a common view of the brain: neurons that are interconnected with other neurons that make up the network. Simple information passes in many of them before it becomes a real thing, such as "break your hand to take this pencil."*

*A neural network is a system that learns to make predictions by following these steps:*

- Taking input data
- Making a forecast
- Comparison of the forecast with the desired result
- Adjust the internal state for correct forecasting next time

*Vectors, layers, and linear regression are some of the building blocks of neural networks. The data is stored as vectors, and with Python, you store these vectors in arrays. Each layer converts coming from the previous layer. You can design the skin as some form of a layer, extracting the presented data that was used before.*

*One feature of neural network layers is that the calculation itself can obtain information from any data. This means that it doesn't matter if you use graphics or text. The process of obtaining meaningful information and learning the model of deep learning is the same for both scenarios.*

**Key words:** *neural networks, Python, object-oriented programming, TensorFlow, Qt Designer, machine learning.*