

МІЖДИСЦИПЛІНАРНИЙ КУРСОВИЙ ПРОЄКТ

на тему: Розробка онтології нейронної мережі

Студента (ки) 4 курсу групи ІТКН-16-8
спеціальності ІІІ КН
Дудка Б. Ю.
(прізвище та ініціали)

Керівник доц. каф. ІІІ, доц., к.т.н.
Вітько О. В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

_____	<u>Вітько О. В.</u>
(підпис)	(прізвище та ініціали)
_____	<u>Терзіян В. Я.</u>
(підпис)	(прізвище та ініціали)
_____	<u>Головянко М. В.</u>
(підпис)	(прізвище та ініціали)

Харківський національний університет радіоелектроніки

(повне найменування вищого навчального закладу)

Інститут, факультет, відділення _____ КН

Кафедра, циклова комісія _____ ІІІ

Освітньо-кваліфікаційний рівень _____ бакалавр

Спеціальність _____ 122 “Комп’ютерні науки”

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

“ _____ ” _____ 2019 року

З А В Д А Н Н Я
НА МІЖДИСЦИПЛІНАРНИЙ КУРСОВИЙ ПРОЕКТ

_____ Дудка Богдан Юрійович

(прізвище, ім’я, по батькові)

1. Тема роботи _____ розробка онтології нейронної мережі

керівник роботи _____ доц. Вітько О. В., к.т.н., доц.

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

2.Строк подання студентом проекту (роботи) _____ 23 грудня 2019 р.

3. Вихідні дані до проекту (роботи) _____ постановка задачі на атестаційну роботу бакалавра, інформаційні джерела з обраної предметної галузі, визначення термінів в предметній галузі, інформаційні джерела із галузі Semantic Web, специфікація мови OWL

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)
_____ вступ, ціль роботи, аналіз предметної галузі та постановка задачі, розробка онтології, логічне виведення за онтологією, висновки.

5. Дата видачі завдання _____ 15.10.2019

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів проекту	Термін виконання етапів проекту	Примітка
1	Видача завдання	15.10.2019	виконано
2	Аналіз предметної області	16.10.2019 — 02.11.2019	виконано
3	Постановка задачі	03.11.2019 — 13.11.2019	виконано
4	Розробка онтології	14.11.2019 — 10.12.2019	виконано
5	Формування логічних виведень за онтологією	11.12.2019 — 13.12.2019	виконано
6	Оформлення пояснювальної записки	14.12.2019 — 18.12.2019	виконано
7	Захист курсового проекту	23.12.2019	

Студент _____ Дудка Б. Ю.
(підпис) (прізвище та ініціали)

Керівник проекту _____ Вітько О. В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до міждисциплінарного курсового проекту 7-го семестру, 25 стр., 15 рис., 0 табл., 5 дж., 1 додат.

Мета курсового проекту - розробка онтології для бакалаврської роботи, а саме нейронна мережа, яка розпізнає рукописний текст і трансформує його в друкований.

Об'єкт дослідження - онтологія, процес її розробки, створення онтології для певної доменної області.

Методи дослідження - базуються на програмному додатку Protege 5.5.0. та технології OWL.

В результаті виконання курсового проекту було розроблено онтологію, яка описує базову структуру нейронної мережі.

ONTOLOGY, OWL, PROTEGE, OBJECT PROPERTY, DATA PROPERTY, NEURAL NETWORK, NEURON, МАШИННЕ НАВЧАННЯ, ОНТОЛОГІЯ, КЛАСИ

ЗМІСТ

Реферат	4
Зміст	5
Вступ	6
1 Ціль роботи	7
2 Аналіз предметної галузі та постановка задачі	8
2.1 Аналіз предметної області	8
2.2 Постановка задачі	11
3 Розробка онтології	12
3.1 Створення ієрархії класів	12
3.2 Створення Object Properties	15
3.3 Створення Data Property	16
3.4 Створення екземплярів класів	16
4 Логічне виведення за онтологією	18
Висновки	20
Перелік посилань	21
Додаток А	22

ВСТУП

Наразі Інтернет-технології стрімко розвиваються в зв'язку з технологічним бумом та бурхливою розробкою нових комп'ютерних технологій. Але при цьому забезпеченню, що розробляється все ще важко розуміти інформацію таким чином, як її сприймає людина. Тому дані потрібно якимось чином формалізувати.

Одним з таких способів є розробка інтелектуальних моделей представлень формалізованого типу. Вони описують модель в певній доменній області за допомогою якогось загально прийнятого синтаксису. Якщо правильно розробити модель, то вона буде зрозуміла як людині, так і програмному забезпеченню, яке може працювати з цією моделлю. Способи оброблення моделей називаються інтелектуальними методами їх обробки. З їх допомогою можливо отримувати вибірки даних з моделі, додавати нові значення і т.д.

Таким чином опис якоїсь доменної області за допомогою моделей надає великий плюс - можливість зручно працювати з об'єктом доменної області, що в свою чергу допомагає автоматизувати певні процеси. Наприклад, схожим чином працюють інтернет індексатори, які "ходять" по сайтах, читають структуру HTML документу та формують відомості про той чи інший сайт, як розмістити його в пошуковій видачі, на які запити його відображати і т.д. Також таким чином можна збирати аналітичні дані і багато інших підходів.

1 ЦІЛЬ РОБОТИ

Метою даної курсової роботи виступає закріплення та розширення знань, що стосуються представлення даних і знань у структурному та семантичному вигляді, онтології, процесу її розроблення.

Окрім цього здобуті навички підкріплюються практично шляхом розроблення онтології для обраної доменної області. В даному випадку це онтологія нейронної мережі. Розроблена онтологія повинна відображати структуру нейронної мережі, класи, які описують її елементи (такі як активаційна функція, нейрони і т.д.), також описуються потрібні параметри.

Не менш важливим етапом є здобуття навичок з логічного виведення, оскільки з його допомогою демонструється функціонал розробленої онтології.

2 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Аналіз предметної області

Оскільки за останні декілька років розвиток комп'ютерних технологій кардинально збільшився в масштабах людству стала доступна можливість використання нейронних мереж. За своєю суттю нейронна мережа - це наближена копія нейронної системи людини та тварин, яку перенесли в комп'ютер [1]. Вона здатна приймати які дані, генерувати вихідні дані але головне - вона здатна навчатися.

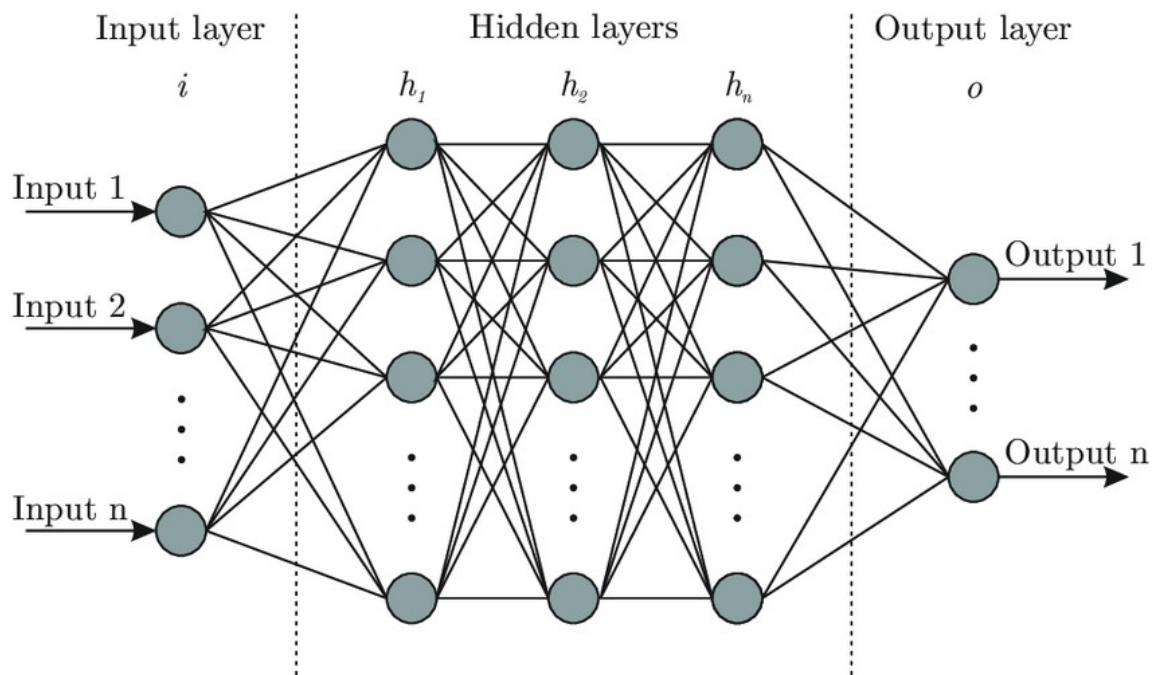


Рисунок 2.1 - структура типової нейронної мережі

Структуру нейронної мережі можна описати наступним чином - є вхідні рівень, який носить в собі дані. У кожного вхідного значення окрім самих даних є же ваговий коефіцієнт, який відображає на скільки це значення вагоме під час класифікації. Є вихідний рівень, який несе в собі згенеровану

мережею відповідь. І є приховані рівні. Саме за допомогою цих даних мережа приймає те чи інше рішення.

Кожен рівень складається з нейронів - ділянки, які обробляють вхідні дані [2]. Далі за допомогою активаційної функції вихідні дані трансформуються і передаються наступному нейрону. Активаційна функція допомагає позбутися такої проблеми, як лінійність досліджень, що дає нам більшу гнучкість та можливість обробляти більш складні та менш структуровані набори даних. Інформація між нейронами передається за допомогою синапсів.

На початку своєї роботи нейронна мережа не має жодного поняття про об'єкт, який вона вивчатиме. Так, якщо мережа аналізує фото і знаходить на ньому котів, то на початку дослідження вона не знає нічого про котів - їх особливості побудови, поведінки і т.д. Все це вона дізнається на етапі навчання. На даному етапі мережі подається набір тренувальних даних, вона повинна класифікувати об'єкт на фото. Якщо мережа класифікувала об'єкт правильно, то процес навчання продовжується. Інакше вона "штрафується".

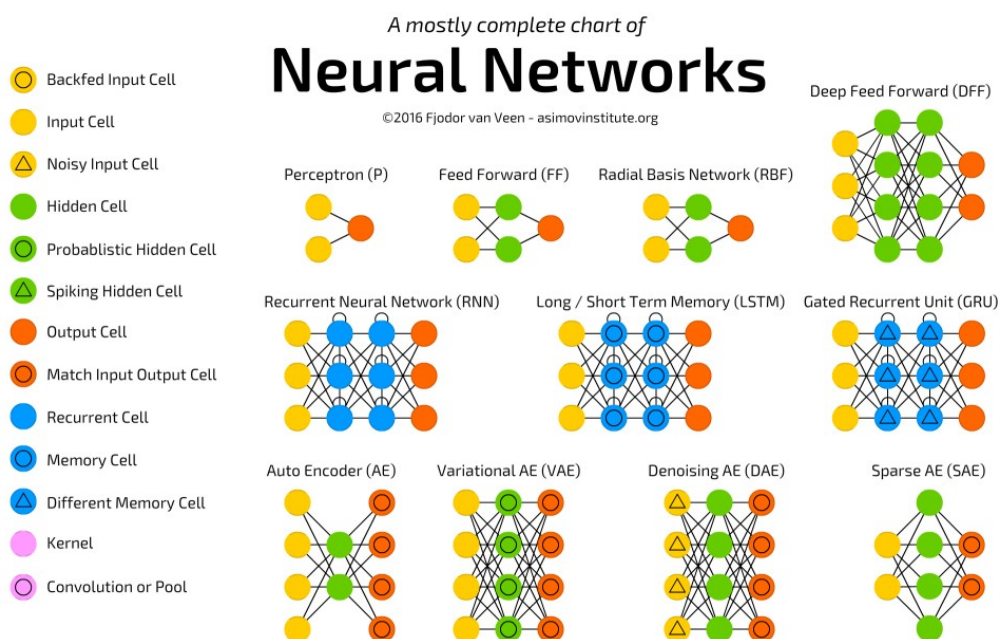


Рисунок 2.2 - види нейронних мереж та їх структура

Як було сказано раніше, нейронні мережі користуються великим попитом на даний момент. Їх використовують в медицині, науці, в автомобілях, телефонах і т.д. Також популярною задачею для нейронних мереж є розпізнання об'єктів на зображенні, зокрема розпізнавання тексту.

Мережа отримує на вхід зображення з текстом і повинна коректно конвертувати його в друкований текст. Дана технологія зараз є майже в кожному перекладі на телефоні. Більш складною є задача розпізнавання рукописного тексту. Основна складність в тому, що якщо мережу навчати тільки на одному прикладі, то вона може помилково або й зовсім хибно розпізнати текст написаний іншою людиною, адже у кожної людини почерк відрізняється. Таким чином складність даної задачі кардинально збільшується. При цьому дана задача все ще потребує рішення адже її рішення все ще є затребуваною.

Наприклад, людина написала вам листа але ви не знаєте мову, якою цей лист написаний. За допомогою нейронної мережі не потрібно вручну писати слова мови, яка вам не відома. Можна просто сфотографувати текст, надати

його нейронній мережі, а вона в свою чергу виконає всю роботу. Також її можна використати при написанні конспекту. Написаний від руки конспект можна легко трансформувати в електронний без потреби витрачання годин часу на повторний його набір.

На даний момент на ринку вже існує велика кількість подібних сервісів, враховуючи продукти таких гігантів, як Google (TensorFlow), Amazon і т.д. Їх API можна використовувати для впровадження функціоналу з розпізнавання тексту в власний проект. Мінусом цього підходу виступає малий функціонал, розширення якого потребує доплати. Тому в якості експерименту можна створити власну мережу.

2.2 Постановка задачі

Виходячи з описаної вище інформації, розроблення нейронної мережі для розпізнавання рукописного тексту все ще є актуальною проблемою, яка потребує вирішення. Цей проект може бути використаний в безлічі сервісів в якості стороннього додатку або ж існувати самостійно.

Але як і більшість проектів, окрім теоретичних знань дана задача потребує планування та схематичного уявлення продукту, що розробляється. Для вирішення цієї проблеми було прийняте рішення розробити онтологію типової нейронної мережі. Дана онтологія допоможе з'ясувати чи наявне правильне розуміння щодо того, що таке нейронна мережа, який у неї принцип побудови, її складові і т.д.

3 РОЗРОБКА ОНТОЛОГІЇ

Перш за все потрібно уточнити поняття онтології. Онтологія - це загальноприйнята концептуалізація певної доменної області яка містить базис для опису цієї області і визначає способи та методи взаємодії між компонентами цієї області [3]. Крім цього вона включає домовленості про представлення доменної області. Перш за все при розробці онтології потрібно створити ієрархію класів

3.1 Створення ієрархії класів

Для початку потрібно описати базову структуру ієрархії. Клас Artificial Neural Network виступає класом, який агрегує в собі всі складові нейронної мережі. Він знаходиться в ієрархії Model.

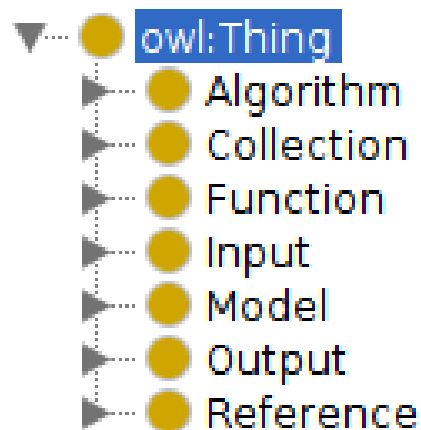


Рисунок 3.1 - базова ієрархія класів

Мережа має такі складові:

- 1) **Algorithm** - алгоритми, які використовуються в мережі. Це може бути як алгоритм ініціалізації, так і тренувальний алгоритм.

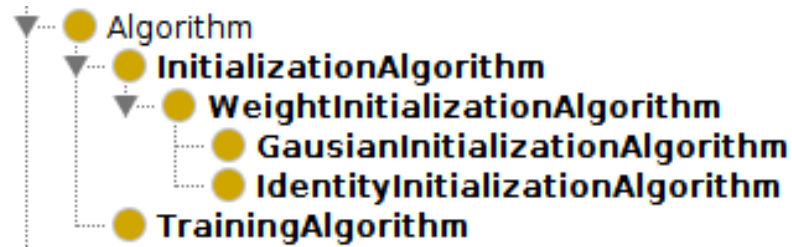


Рисунок 3.2 - ієрархія класів Algorithm

- 2) **Collection** - описує об'єкти, які мають одночасно декілька різних екземплярів. Наприклад Layer. Одночасно мережа може мати декілька рівнів, серед яких Input, Output і т.д.



Рисунок 3.3 - ієрархія класів Collection

- 3) **Function** - описує основні математичні компоненти мережі. Серед них нейрони, активаційна функція, функція втрат [4] і т.д.

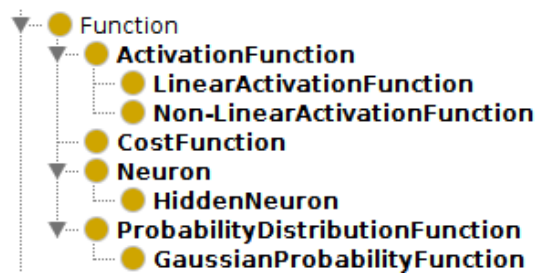


Рисунок 3.4 - ієрархія класів Function

- 4) Input - описує об'єкти, які мережа може приймати на вхід в якості вхідних даних. Наприклад, це може бути фотографія або текст.



Рисунок 3.5 - ієрархія класів Input

- 5) Output - тип вихідних даних мережі. Ця ієрархія класів описує вигляд, до якого мережа приводить результат. Наприклад, друкований текст.



Рисунок 3.6 - ієрархія класів Output

- 6) Reference - опис даних, які зберігаються в нейронах. Це може бути вагомість даного значення порівняно з іншими значеннями, а також набір початкових параметрів мережі (кількість рівнів, швидкість навчання і т.д.) [5].

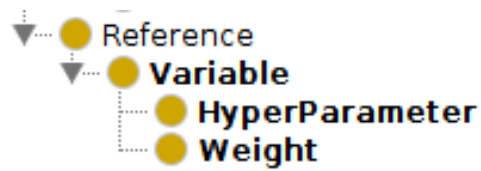


Рисунок 3.7 - ієрархія класів Reference

3.2 Створення Object Properties

Об'єктні властивості слугують для створення полей в класах. Основні поля мережі - вона повинна мати хоча б один рівень, в свою чергу нейрон повинен бути частиною хоча б одного рівня і частиною самої мережі. Крім цього у нейронна повинна бути активаційна функція. Також мережа повинна мати алгоритм ініціалізації.

Виходячи з описаних фактів маємо наступну ієрархію об'єктів:

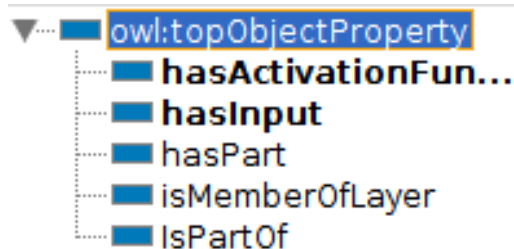


Рисунок 3.8 - згенеровані властивості об'єкту

Крім цього, для класів були додані обмеження, наприклад:

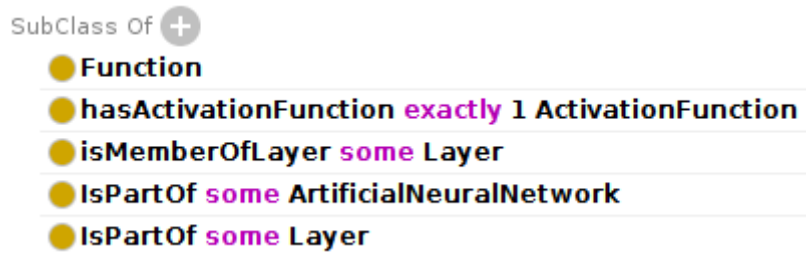


Рисунок 3.9 - обмеження класу Neuron

3.3 Створення Data Property

Властивості даних описують примітивні значення полів класу. В нашому випадку це можуть бути значення функції втрат, початкова кількість прихованих рівнів та значення вагових коефіцієнтів.

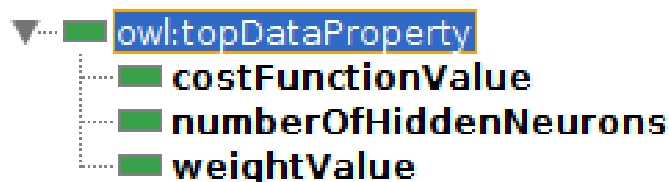


Рисунок 3.10 - властивості даних

3.4 Створення екземплярів класів

Як результат розробленої ієрархії класів були створені екземпляри класів, потрібних для створення нейронної мережі. В екземпляри входять: декілька екземплярів рівнів, нейрони, алгоритм ініціалізації, декілька активаційних функцій, тестові входні дані і т.д.

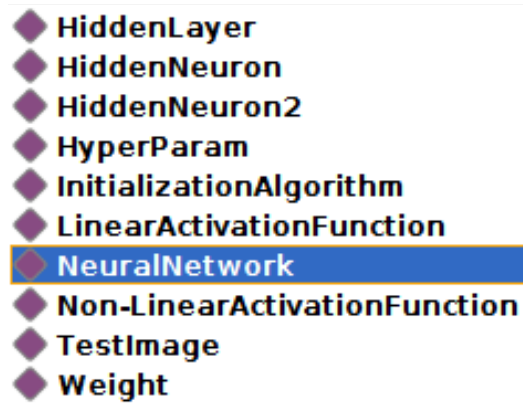


Рисунок 3.11 - перелік створених екземплярів

Крім цього, для екземплярів були задані об'єктні властивості та властивості даних.

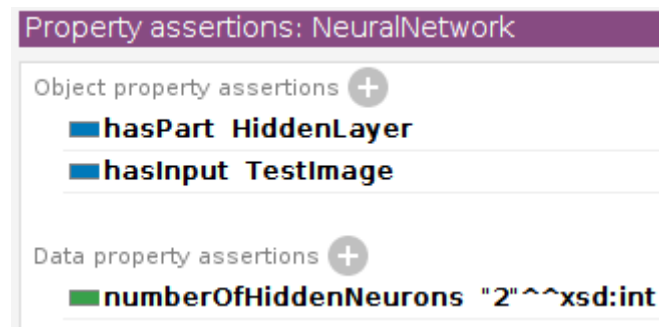


Рисунок 3.12 - приклад заповнених властивостей екземпляру

4 ЛОГІЧНЕ ВИВЕДЕННЯ ЗА ОНТОЛОГІЄЮ

В результаті розроблення онтології було отримано ієрархію класів, яка відображає структуру нейронної мережі. Окрім цього онтологія була провалідована вбудованим валідатором HermiT.

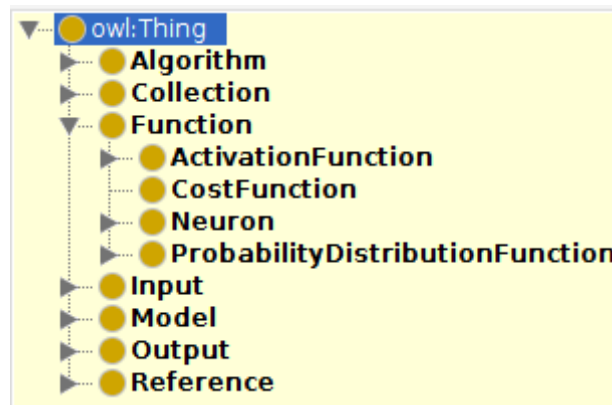


Рисунок 4.1 - результат валідації онтології

Розроблена ієрархія може в подальшому слугувати в якості джерела даних про доменну область. Інформацію з неї можна отримувати за допомогою певних запитів. Серед них:

- 1) Отримання повної ієрархії класів за класом;
- 2) Отримання вибірки даних на основі певного атрибуту або характеристики;
- 3) Валідація даних на відповідність заданим обмеженням;

Окрім приведених вище запитів спеціальні бібліотеки (наприклад, Jena) дозволяють не просто читати інформацію з готової онтології, а й розширювати її. Розглянемо більш детально описані вище запити.

Що стосується першого запиту, то він може бути використаний у випадку, коли користувачу потрібно дізнатися про всі доступні класи в даний момент. Наприклад, які алгоритми ініціалізації або який тип активаційної функції можуть бути використані. Таким чином він розуміє з чим працює.

Запити другого типу можуть бути використані для отримання набору даних за якимсь довільною характеристикою. Вони можуть бути використані для отримання статистичних даних, валідації і т.д. Наприклад, отримати екземпляри, у яких кількість рівнів більше або дорівнює трьом.

Крім цього вони можуть бути задіяні для отримання класів з певними характеристиками. Так, можна отримати перелік класів нейронів, які мають конкретну активаційну функцію.

Запити третього типу використовуються для валідації вже розробленої або модифікованої онтології. Наприклад, вони можуть слугувати для перевірки наявності певної частини нейронної мережі в її екземплярі або перевірити валідність переданого типу властивостей даних. Дані функції відіграють важливу роль адже не дають можливості вийти за рамки розробленої онтології.

ВИСНОВКИ

Під час розроблення даного курсового проекту було отримано більш поглиблені знання в предметній області, а саме нейронні мережі. Набуті знання будуть в майбутньому використані для написання бакалаврської роботи. Серед отриманих знань можна виділити ознайомлення з поняттям нейронних мереж, ознайомлення з принципами побудови нейронних мереж, їх основними типами.

Окрім теоретичних відомостей отримані знання були застосовані практично шляхом розроблення онтології для обраної доменної області. В результаті цього була створена ієрархія класів, яка відображає структуру нейронної мережі. Для класів були задані об'єктні властивості, що характеризують дані, які може агрегувати екземпляр певного класу. Крім цього, для деяких класів були задані властивості даних, які зберігають примітивні типи з даними класу. Були накладені обмеження на певні класи. Таким чином вони описують допустиму структуру мережі.

В результаті для побудованої онтології в якості прикладу були створені екземпляри потрібних класів. Також онтологія була провалідована вбудованим валідатором.

ПЕРЕЛІК ПОСИЛАНЬ

1. Штучна нейронна мережа [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0.
2. Google. Neural Networks: Structure [Електронний ресурс] / Google – Режим доступу до ресурсу: <https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/anatomy>.
3. Ontology (information science) [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Ontology_\(information_science\)](https://en.wikipedia.org/wiki/Ontology_(information_science)).
4. Everything you need to know about Neural Networks [Електронний ресурс] – Режим доступу до ресурсу: <https://hackernoon.com/everything-you-need-to-know-about-neural-networks-8988c3ee4491>.
5. Radhakrishnan P. What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network? [Електронний ресурс] / Pranoy Radhakrishnan – Режим доступу до ресурсу: <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>.

ДОДАТОК А
ФРАГМЕНТ OWL ОНТОЛОГІЇ

Приклад декларації класу ArtificialNeuralNetwork:

```

<owl:Class rdf:about="http://www.semanticweb.org/bohdan/ontologies/2019/11/ neural-
network#ArtificialNeuralNetwork">
  <rdfs:subClassOf          rdf:resource="http://www.semanticweb.org/bohdan/ontologies
/2019/11/neural-network#ComputationalModel"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="http://www.semanticweb.org/bohdan/ontologies/2019/11/neural-
network#hasPart"/>
      <owl:someValuesFrom
        rdf:resource="http://www.semanticweb.org/bohdan/ontologies/2019/11/neural-
network#Layer"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty    rdf:resource="http://www.semanticweb.org/bohdan/ontologies/2019
/11/neural-network#hasInput"/>
      <owl:qualifiedCardinality
        rdf:datatype="http://www.w3.org/2001/XMLSchema#non
NegativeInteger">1</owl:qualifiedCardinality>
      <owl:onClass
        rdf:resource="http://www.semanticweb.org/bohdan/ontologies/2019          /11/neural-
network#Input"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

```

Приклад створення екземпляру InitializationAlgorithm:

```
<owl:NamedIndividual          rdf:about="http://www.semanticweb.org/bohdan/ontologies/2019/11/neural-network#InitializationAlgorithm">
    <rdf:type                  rdf:resource="http://www.semanticweb.org/bohdan/ontologies/2019/11/neural-network#InitializationAlgorithm"/>
    <IsPartOf                  rdf:resource="http://www.semanticweb.org/bohdan/ontologies/2019/11/neural-network#NeuralNetwork"/>
</owl:NamedIndividual>
```

Приклад створення властивості об'єкту:

```
<owl:ObjectProperty          rdf:about="http://www.semanticweb.org/bohdan/ontologies/2019/11/neural-network#IsPartOf"/>
```