

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра інформаційно-обчислювальних систем і управління

Новосад Станіслав Олександрович

Метод прогнозування результатів виборів на основі  
нейронної мережі з нечіткою логікою/ Method of predicting  
election results based on a neural network with fuzzy logic

спеціальність: 122 - Комп'ютерні науки  
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи  
КНМ-21  
С.О.Новосад

---

Науковий керівник:  
к.т.н., проф. В.В.Кочан

---

Кваліфікаційну роботу  
допущено до захисту:  
«\_\_\_» \_\_\_\_\_ 20\_\_\_ р.  
Завідувач кафедри  
\_\_\_\_\_ М.П. Комар

ТЕРНОПІЛЬ - 2021

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	5
ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ВИБОРІВ .....	9
1.1 Особливості прогнозування результатів виборів та існуючі розробки.....	9
1.2 Архітектури нейронних мереж для прогнозування .....	13
1.3 Вибір напрямку дослідження та постановка завдання .....	22
Висновки до розділу 1 .....	25
2 ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ВИБОРІВ .....	27
2.1 Основні методи навчання нейронних мереж .....	27
2.2 Використання нейронних мереж і нечіткої логіки для прогнозування .....	32
2.3 Використання бібліотек при розробці моделі мережі .....	35
Висновки до розділу 2.....	38
3 ЕСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ .....	39
3.1 Підготовка набору даних для навчання мережі.....	39
3.2 Побудова моделі уподобань.....	45
3.3 Оцінка результатів експерименту .....	51
Висновки до розділу 3.....	54
ВИСНОВКИ .....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	56
Додаток А    Фрагмент коду створення і використання мережі <b>Error!</b>	

**Bookmark not defined.**

Додаток Б .....**Error! Bookmark not defined.**

Додаток В Довідка про використання результатів дипломної роботи  
.....**Error! Bookmark not defined.**

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ANN (artificial neural network) – штучна нейронна мережа

AlexNet – вид згорткової нейронної мережі

Back-propagation – метод зворотного поширення похибки

CNN (convolutional neural network) – згорткова нейронна мережа

Learning Rate – темп навчання

Stochastic Gradient Descent – стохастичний градієнтний спуск

НМ - нейронна мережа

ЗНМ - згорткова нейронна мережа

ШНМ - штучна нейронна мережа

## ВСТУП

Людина завжди хоче зазирнути в майбутнє, або хоч привідкрити пелену часу. Ще з давніх часів у міфах і билинах згадуються оракули, віщуни, гадалки які передбачували долі, крім того існували і спеціальні інструменти та засоби для передбачень. З часом до цього залучили і науку щоб визначити закономірності перебігу процесів в часі, адже значна кількість із них піддаються формалізації. На процеси впливають певні фактори, зміна значення яких і дозволяє зробити передбачення з певною долею ймовірності. Але не всі процеси піддаються формалізації, як правило до них відносяться ті процеси і явища, що залежать від значної кількості факторів, кожен з яких змінює вагу свого впливу з часом. Крім того, значення факторів можуть бути взагалі відсутніми, не варто, також, забувати і зовнішній вплив. Передбачення, або прогноз впливає на процес прийняття рішень, тому чим він точніший, тим більший ефект від запроваджених заходів, а хибний прогноз призводить до прийняття неправильних рішень.

До сфер, де прогнозування має велике значення відносяться вибори. Виборчий процес є соціально-політичним явищем і на нього впливають різні чинники, як об'єктивні так і суб'єктивні. Одним із головних чинників, що впливає на кінцевий результат, є електоральна поведінка. До вивчення електоральної поведінки в США та Західній Європі спонукало запровадження виборчого права і отримання права голосу громадянами, основними питаннями при цьому були які мотиви виборця при виборі і які фактори впливають на його поведінку. Електоральні дослідження призначенні для прогнозування результатів виборів і вивчення поведінки виборців. Політичні процеси моделювати значно складніше, ніж природні процеси. В політиці людині доводиться діяти в умовах браку достовірної інформації. Використання моделей за умов обмеженої інформації дає змогу прогнозувати розвиток досліджуваного об'єкта.

На сьогоднішній день виділено три „класичних” моделі електоральної

поведінки, де виділяється основним одним із факторів електоральної поведінки, при цьому інші фактори не відкидаються взагалі, але їх вплив вважається незначним.[1-5] Соціологічна модель виділяє основним фактором належність виборця до певної соціальної групи, при цьому на вибір впливає оточення та ЗМІ. Соціально-психологічна модель визначає ключовим механізмом групової ідентифікації, тобто партійну приналежність або симпатію виборця, тут також виділяється афективна поведінка виборця що базується на мотиві протесту. Останньою з класичних моделей є теорія раціонального вибору – базується на раціональній поведінці особи, при цьому виборець виступає в ролі споживача політичної послуги яку пропонують учасники виборів. Вивчення “класичних” моделей вказує на те, що їх фактично неможливо застосувати їх в Україні і взагалі на пострадянському просторі не врахувавши специфіки, але вони дають можливість виділити ключові фактори, що впливають на результати виборів.

**Актуальність теми** полягає у важливості результатів виборів та їх впливу на національну і світову економіку і як наслідок, потребі їх передбачення для політиків, бізнесу і громадян, та складності отримання класичними методами на основі великої кількості факторів результатів прогнозування.

**Метою даної роботи** є розробка методу прогнозування результатів виборів з застосуванням нейронних мереж.

**Об’єктом дослідження** є процес організації нейромережевого прогнозування результатів виборів.

**Предмет дослідження** – методи і моделі прогнозування електоральної поведінки та результатів виборів.

**Методи дослідження** – методи машинного навчання з використанням тренувальної множини;

**Новизна отриманих результатів** полягає у формалізації і обґрунтуванні методу прогнозування, що використовуються в соціальних дослідженнях електоральної поведінки.

**Практичне значення** Реалізовано метод прогнозування результатів виборів з використанням нейронних мереж. Представлено експериментальні дослідження.

**Публікації та апробація.** Результати кваліфікаційної роботи апробовані та опубліковані в матеріалах:

Матеріали міжнародної науково-практичної конференції «Актуальні проблеми науки, освіти та технологій: теорія та практика», Полтава, 16 грудня, 2021р

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ВИБОРІВ

## 1.1 Особливості прогнозування результатів виборів та існуючі розробки

Для досліджень в галузі економіки на сьогодні характерним є запровадження сучасних математичних методів, а також розробка новітніх комп'ютерних технологій, що дають можливість проводити дослідження складних явищ і процесів. Прогнозування – чи не найнеобхідніша, але разом із тим і складна задача інтелектуального аналізу даних. До основних проблем прогнозування можна віднести брак якості та кількості вхідних даних, змінами середовища протікання процесу, а також суб'єктивні фактори. Прогноз завжди характеризується певною похибкою, на що впливає вибрана модель прогнозу, та повнота вхідних даних.

Сфера передбачення результатів виборів цікава так як, з одного боку важлива для національної та світової економіки і суспільства в цілому, а з іншого через складність реалізації, адже на неї впливає політична система країни, неоднакового впливу факторів в різних країнах та політичного контексту в світі і в країні зокрема. Залежно від факторів, доступних даних, мети отримани результатів прогнозування може вирішувати різні задачі. Суб'єктами виступають кандидати на певні посади різного масштабу, кандидати на пост президента, та політичні об'єднання та партії. Метою може слугувати визначення шансів на перемогу кандидата, ймовірності подолання прохідного бар'єру, передбачення кількості голосів на підтримку кандидата, а також визначення ступеню впливу факторів на вибір електорату.

Перші моделі, призначені для прогнозування результатів виборів, почали використовуватись у 1950-х і більш широко поширились у середині 1990-х. На теперішній час можна виділити два різновиди таких моделей: перший полягає в визначенні ймовірності перемоги кандидата спираючись на числові соціально-економічні показники; в основі другого



використовуються дані соціологічних опитувань.

Донедавна основними методами, що використовувались у прогнозуванні, були статистичні, які базувались на накопиченні і обробці даних.

За час свого використання, вони довели свою ефективність за умови достатнього об'єму даних. Основою їх використання вибірка даних значного розміру для декількох змінних в ній, тоді вони мають високу точність у передбаченні результату.

До переваг цих методів відноситься можливість застосування на різних стадіях виборів якщо нема потреби враховувати значну кількість факторів, точність статистичних прогнозів напряму залежить від достовірності даних зібраних в процесі дослідження. Також важливою є повнота не тільки даних, а також і групи подій, тобто кожна подія містить всю групу значень, і в результаті експерименту отримує лише одне значення із групи. При використанні статистичного прогнозування застосовуються різні методи, розглянемо їх детальніше.

Прогнозування на основі результатів, отриманих при громадських опитуваннях дає різні результати, що залежить від якості проведення опитувань і по різному показує себе в різних країнах. Проблема в тому, що висвітлюється думка певної частини електорату і лише в визначений момент часу. Громадська думка за деякий час може кардинально змінитися, при цьому проведення опитувань повторно є затратним з економічної точки і все рівно не дає точності. В процесі опитування люди, що дають відповідь щодо своїх вподобань не відчувають відповідальності в цей момент за свій вибір, тому їх відповіді не завжди відображають за кого насправді будуть голосувати, навіть напередодні виборів. Але коли з різних причин зібрати інші дані з задовільною точністю немає можливості, то в таких випадках цей метод є непоганим варіантом.

Використання соціально-економічних показників базується на використанні поєднання політичної ситуації в країні, показників

економічного стану та особистих уподобань при моделюванні поведінки виборців. Як приклад може виступати модель Девіда Сандерса [6] що опиралась на бачення громадянами благополуччя в економічному плані і дані щодо вибору Консервативної партії на загальних виборах 1991 року у Великій Британії. На підході, де використовуються певні економічні показники разом із врахуванням рейтингів кандидатів та партій базується багато праць щодо прогнозування результатів виборів у США. Основною проблемою цієї моделі є використання опитувань, що неточно відображають а інколи і спотворюють реальну громадську думку. Використання нейронних мереж дозволить використовувати більше соціальних і економічних факторів, а також визначати між ними нелінійні залежності.

Ще один популярний метод моделювання - за результатами попередніх виборів, що асоціює відношення виборці до певної політичної сили та результатами виборів в минулі роки. При цьому відпадає необхідність проведення соціальних опитувань та отримати дані для прогнозування результатів в майбутньому відразу після виборів. Таку модель використав Гельмут Норпот [7], при цьому враховуючи результати кандидата передвиборах, оскільки це впливає на успіх кандидата у виборах. Модель чудово себе зарекомендувала, отримавши правильні результати практично всіх виборів з 1912 року.

Метод з використанням експертних систем ґрунтується на думці політичних експертів, які дають відповіді на коло питань щодо виборів, економічного розвитку, реформ, та політики правлячої партії. При цьому часто враховується така ж експертна думка про попередні вибори. При цьому також нема необхідності проводити опитування, але точність і ефективність прогнозу в значній мірі залежать від компетентності експертів.

Щоб збільшити точність прогнозування практикують одночасне застосування декількох моделей, дозволяє підвищити точність окремих методів та вносити корективи. Прикладом може бути робота А. Грейфа [8], що для отримання прогнозу результатів виборів в 2013 році в Німеччині

використав відповіді експертів, числові моделі та ринкові прогнози і дані опитувань. Використання одночасно декількох методів значно збільшило їх точність.

Окрім статистичних методів в прогнозуванні також використовуються експертні системи, ідея систем полягає в представленні системі знань експерта чи групи експертів в певній галузі, що адаптуються для прийняття логічних рішень, тобто вона здатна в подальшому автоматизувати роботу та замінити експерта, прибравши таким чином людський фактор.

В основі такої системи є база знань, вона найбільш придатна для прогнозування результатів виборів за даними минулих років, де може експертів у відповідях на ряд питань щодо економічної та політичної ситуації. Типова схема функціонування експертної системи представлена на рисунку 1.1..



Рисунок 1.1 – Схема функціонування експертної системи.

Такі системи можуть розширюватись в подальшому, доповнюючи наявні знання новими. До недоліків таких систем можна віднести неможливість вилучити повністю людський фактор та функціонувати автономно, а нова інформація потребує опрацювання експертами.

Проблемою є також і складність обрання та збору даних для наповнення бази знань.

Основні проблеми класичних методів пов'язані із потребою врахуванням значної кількості параметрів. Для підтримання точності потрібно збільшувати кількість спостережень але і в цьому випадку велика кількість факторів може знижувати точність моделі. Крім того, фактори та вихідні значення можуть мати складні, невідомі наперед зв'язки, в тому числі нелінійні.

Важливим є те, що при прогнозуванні не тільки беруться до уваги попередні значення величини, але крім того береться до уваги вплив додаткових факторів у вигляді часових рядів . Тобто, прогнозування в даному випадку проводиться шляхом опрацювання декількох часових рядів.

Потрібно враховувати, що в моделі прогнозу прогнозована величина має менше значення ніж передбачення змін її поведінки. Методи статистичного прогнозування, що традиційно використовуються, не враховують повною мірою фактори, що змінюють прогнозовану величину в часі. Для прогнозування можна використовувати штучні нейронні мережі, що дозволить прогнозувати результати за деякими критеріями, які будуть визначатися нейромережею.

Після дослідження електоральної поведінки та методів прогнозування результатів виборів, необхідно визначитись з типом і архітектурою нейронної мережі та напрямком дослідження.

## 1.2 Архітектури нейронних мереж для прогнозування

На початку 40-х років XX століття Уоррен Маккаллох і Уолтер Пітс створили побудовану на алгоритмах просту систему яка імітувала функції людського мозку, це послужило початком розвитку нейронних мереж. Френк Розенблат в 1957 році задумав перцептрон, призначений для виконання

складних завдань розпізнавання, цей революційний алгоритм надав прискорення розробкам в цій галузі [С1]. Після цього прогрес гальмувався через брак обчислювальних потужностей, що необхідні для обробки великих обсягів даних. І лише на початку ХХІ з ростом потужності технічних засобів, а також завдяки мережі Інтернет створення величезних наборів даних з яких можна отримувати інформацію вчені отримали те було необхідно для продовження роботи. Основна перевага в унікальній здатності нейронної мережі отримувати результат із складних або неточних даних, виявляючи закономірності та тенденції, що занадто заплутані для інших комп'ютерних моделей.

Перш ніж, перейти до розгляду архітектур нейронних мереж, необхідно розібратись з самим поняттям "нейронна мережа". До цих пір дослідники не дійшли згоди щодо визначення нейронної мережі. У літературі зустрічається безліч варіантів, наприклад: «Нейронна мережа - паралельний розподілений процесор, здатний самостійно отримувати дані з інформації, що надходить. Робота такої мережі нагадує роботу мозку, так як знання отримуються за допомогою процесу тренування, а отримані знання зберігаються не в окремому елементі, а розподілені по всій мережі»[14].

Хайкин С. пропонує такий варіант: «Нейронна мережа - система, що складається з безлічі простих обчислювальних елементів, що працюють паралельно. Результат роботи мережі визначається структурою мережі, силою зв'язків, а також видом обчислень, виконуваних кожним елементом»[15].

В статті Васенкова Д.В. зустрічається така інтерпретація: «Нейронна мережа - система, що складається з великого числа простих обчислювальних елементів. Результат роботи кожного елемента залежить тільки від його внутрішнього стану. Всі елементи працюють незалежно один від одного, тобто без синхронізації з іншими елементами»[16].

Як видно визначення подібні, і в залежності від "спеціалізації" автора виділяється певна особливість до основного тлумачення, яка відображає

сферу застосування мереж.

Повертаючись до історії розвитку, розглянемо основні архітектури що створенні за час існування нейронних мереж. Нейронна мережа, або штучна нейронна мережа, побудована за принципом функціонування нервових клітин організму. Ключовим елементом нервової системи (якщо точніше людського мозку) є нейрон, його структура представлена на рисунку 1.3.

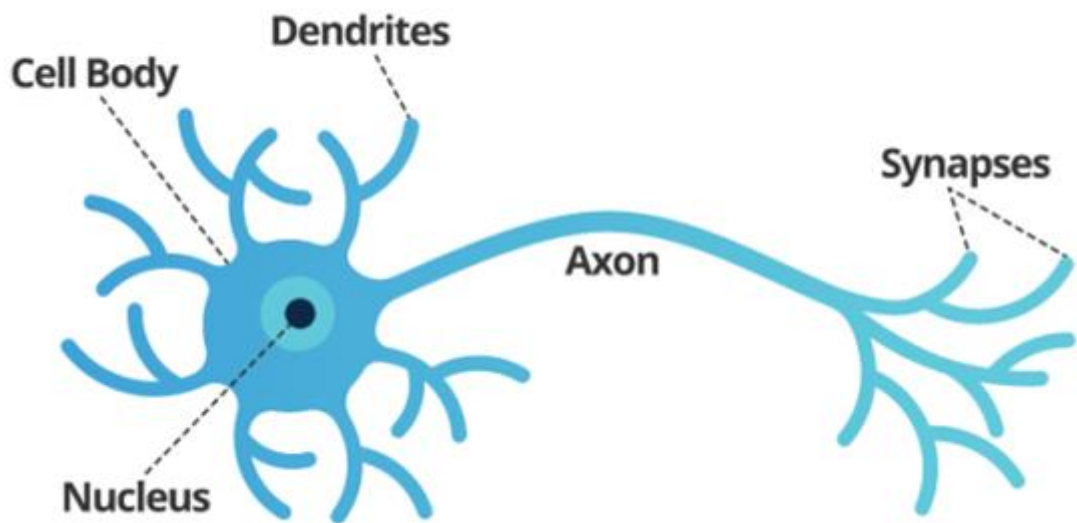


Рисунок 1.3 – Структура нейрону мозку людини

Основою побудови нейронної мережі і його основним обчислювальним елементом є штучний, або формальний нейрон, що містить кілька входів і один вихід[С5]. Схематично штучний нейрон зображено на рисунку 1.4.

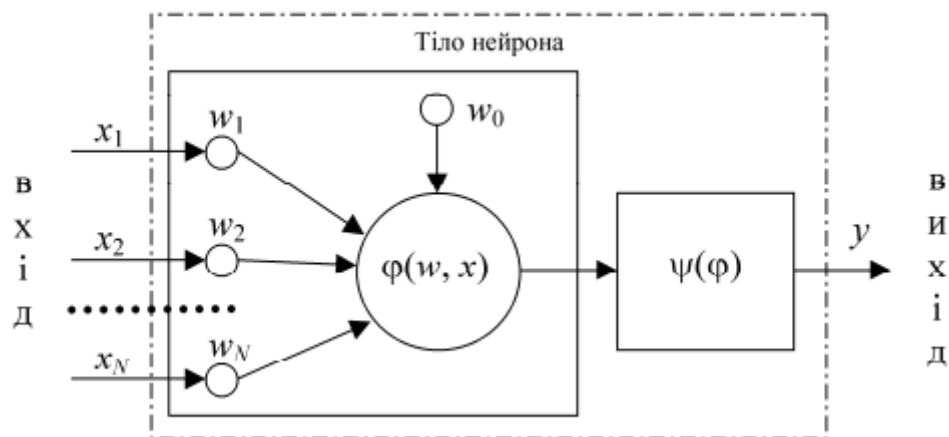


Рисунок 1.4 - Формальний нейрон.

На вхід одношарового нейрона надходить вхідний вектор – набір вхідних сигналів  $x = \{x_j\}$ ,  $j = 1, 2, \dots, N$ , де  $N$  – кількість входів.

Першою дієздатною моделлю, як вже згадувалось раніше, вважається перцептрон Розенбалата, який лежить в основі різних типів нейромереж і складається з одного нейрона.

Сучасні нейронні мережі, виправдовуючи свою назву, складаються з великої кількості нейронів пов'язаних певним чином. Зв'язки між нейронами формують структуру, або архітектуру мережі. В процесі еволюції створювались різні базові структури для вирішення певного класу задач, мережі отримали шари, втому числі приховані, в них змінювались зв'язки та ін.. F. Van Veen в своїй роботі The Neural Network Zoo[18] провів класифікацію основних відомих видів нейронних мереж, що представлена на рисунку 1.5. Представлення у такому виді не показує як вони використовуються, з точки зору процесу навчання відмінностей ще більше, але після відображення вузлів на схемі зв'язки принаймі набувають певного сенсу. Класифікація не є точною, але відображає загальну картину і дозволяє оцінити масштаби даної сфери.

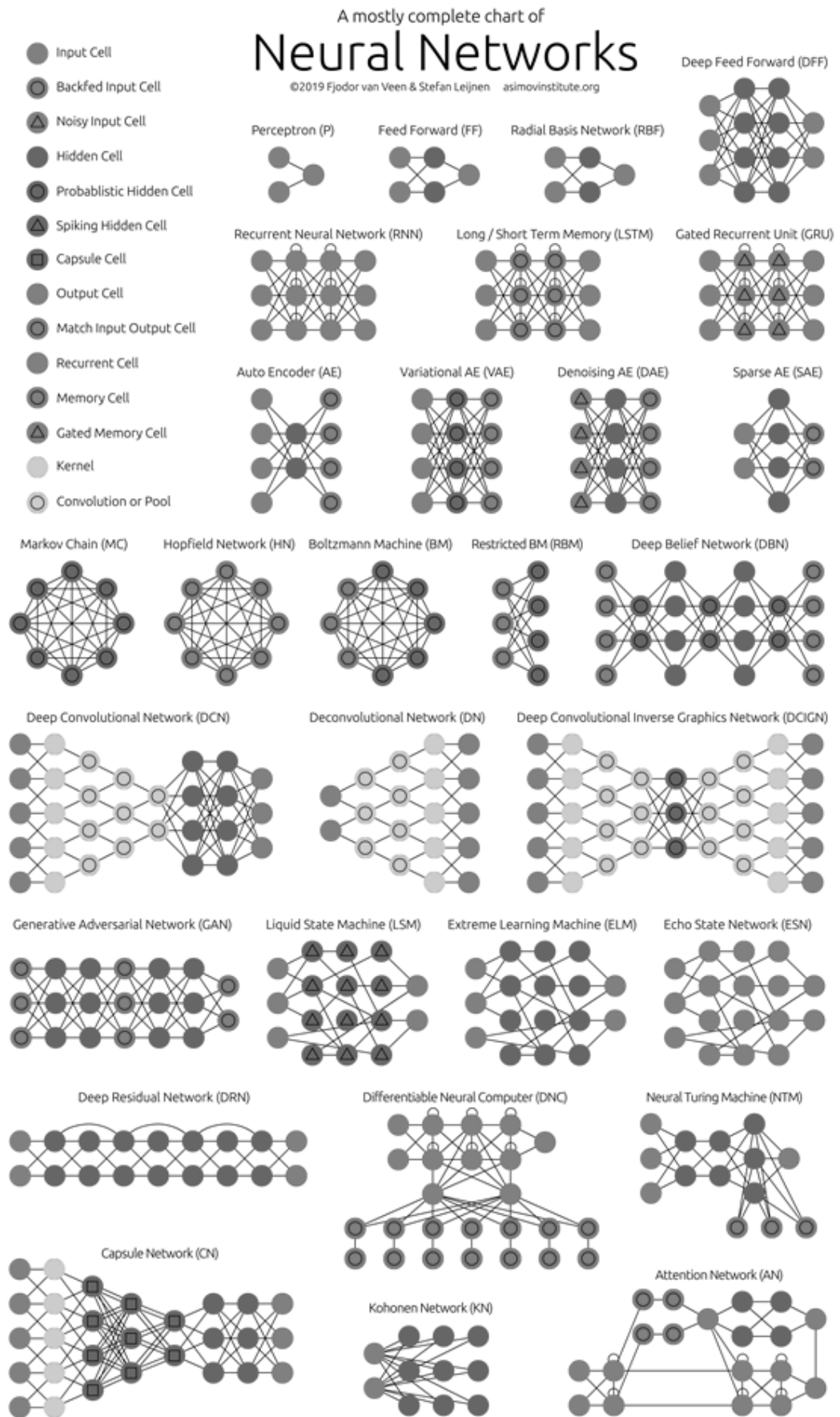


Рисунок 1.5 – Основні види нейронних мереж (The Neural Network Zoo)[5]



Для детального розгляду виберемо ключові архітектури, які є найбільш популярними і поширеними. Не можна оминати увагою багатошарові перцептрони, що є удосконаленням перцептрона Розенблатта, вони відносяться до мереж прямого поширення, де сигнал проходить від входу до виходу в одному напрямку. В цьому випадку нейрони розташовуються шари: вхідний, один або декілька прихованих та вихідний(рисунок 1.6). Зв'язки встановлені між нейронами попереднього і наступного шарів, без зв'язків в межах шару.



Рисунок 1.6 – Багатошаровий перцептрон

. Багатошаровий перцептрон став основою для розробки моделей неронних мереж різного призначення і з різним функціоналом нейронів, зв'язками та алгоритмами навчання.

Використання одного ефективних алгоритмів, так званого зворотнього поширення похибки призвело до виникнення цілого однойменного класу багатошарових мереж (рисунок 1.7). Ця архітектура дає можливість моделювання функції не залежно від її складності, вона складається з вхідного і вихідного шарів та хоча б одного прихованого( як правило один або два).

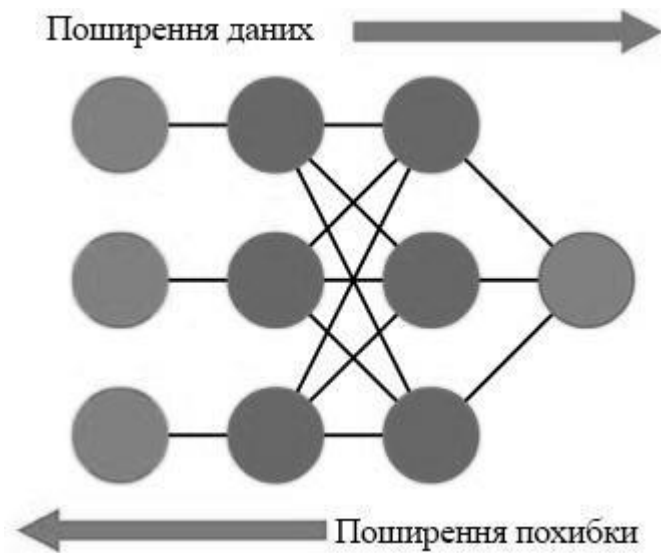
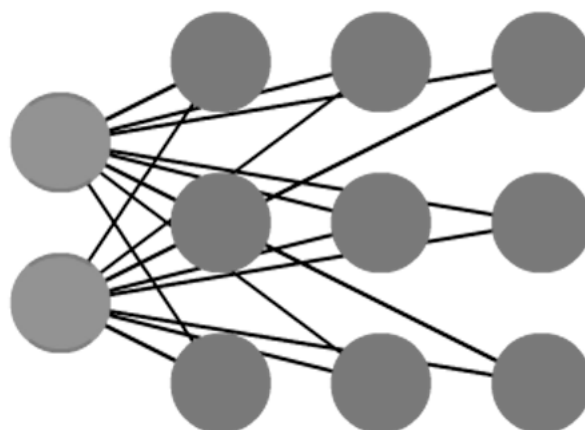


Рисунок 1.7 – Типова мережа зворотнього поширення похибки

Нейрони розміщені пошарово з прямою передачею сигналу. Алгоритм виконується циклічно, ітерації циклу прийнято називати епохами. Кожна епоха – це почергова подача на вхід навчальної множини, порівняння отриманих значень з еталонними і обчислення похибки кожного прикладу, і загальної похибки ітерації, що використовується для корекції ваг. Навчання закінчується через певну кількість епох, або досягненні певного рівня похибки, та коли вона не зменшується кілька епох поспіль. Значення похибки передається в зворотному напрямку.

Ще один тип нейронної мережі - мережа Кохонена[19], ризначена для кластеризації, вона складається з двох шарів - вхідного і вихідного , його ще називають «прошарок Кохонена» (рисунок 1.8). Зв'язки між нейронами такі ж як в багатошаровому перцептроні.



### Рисунок 1.9 – Мережа Кохонена

Кількість вхідних нейронів відповідає кількості ознак об'єкта, в процесі навчання вони розподіляють сигнал по вихідному шару, вихідних нейронів стільки ж як і кількість кластерів, які необхідно побудувати. Алгоритму навчання полягає в присвоєнні нейрону з максимальним значенням на виході 1, а решті – 0, що відносить об'єкт до того кластеру, з яким асоціюється цей нейроном.

Зміна напрямку руху сигналу призвела до появи рекурентних нейронних мереж зі змінною часу. Нейрони отримують інформацію як від попереднього шару, так і від самого себе з минулої ітерації. При цьому порядок подання вхідних даних при навчанні має значення(рисунок 1.9).

В принципі, рекурентні мережі можуть використовуватися в багатьох областях, оскільки даних можна представити у вигляді послідовності. Завдяки їм ми маємо можливість користуватись машинним перекладом текстів і синтезом мови. Основна сфера застосування це вирішення завдань, що пов'язані з послідовностями - голосові, текстові або музичні.

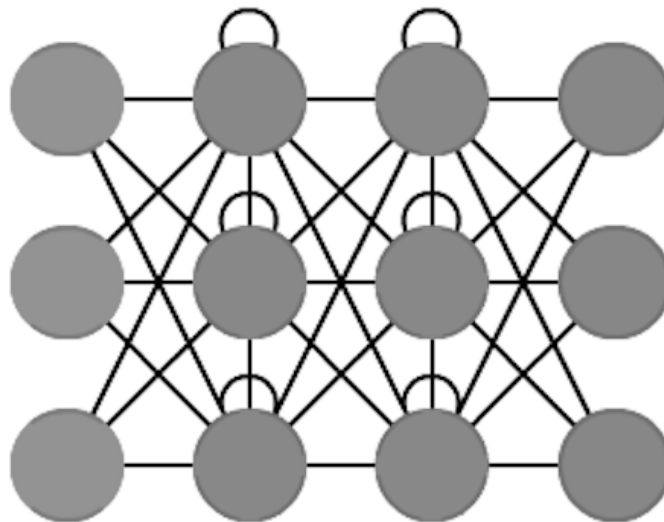


Рисунок 1.9 – Рекурентна нейронна мережа

Особливим видом є згорткові нейронні мережі і глибинні згорткові нейронні мережі (рисунок 1.10), вони кардинальні відрізняються від

інших[20].



Рисунок 1.10 – Згорткова нейронна мережа

На сьогодні це найпопулярніший вид мереж, що застосовується для розпізнавання і ідентифікації об'єктів, перенесення стилю, створення ефектів і підвищення якості фото, тобто практично скрізь, де є відео або зображення і в основному, виконують операції класифікації .

Провідним дослідником ЗНМ є Yann LeCun. У 1998 році за результатами досліджень опубліковано роботу[20], в якій описана архітектура однією з перших згорткових нейронних мереж LeNet, на її основі розроблено низку архітектур, але базова структура у всіх аналогічна LeNet.

Варті уваги розгорткові нейронні мережі, або зворотні графічні мережі, вони є оберненими до ЗНМ[21] (рисунок 1.11). Як видно з назви, мережа генерує зображення об'єкта, який відноситься до класу, ідентифікатор якого подано на вхід.



Рисунок 1.11 – Розгорткова нейронна мережа

Розглянуті архітектури є базовими для використання в різних областях, вибір тієї чи іншої архітектури для реалізації залежить від предметної області, доступного набору даних і їх формату, а також мети використання результатів.

### 1.3 Вибір напряму дослідження та постановка завдання

Нейронні мережі, як засіб розробки володіють такими атрибутами:

- адаптивне навчання: здатність моделювати складні і нелінійні відносини спираючись на отримані знання;
- самоорганізація: здатність класифікувати і кластеризувати значні обсяги даних;
- робота в реальному часі: застосування нейронних мереж для управління безпілотними автомобілями, літальними апаратами та дронами;
- прогноз: здатність НМ на основі моделей прогнозувати має широке застосування в різних галузях;
- відмовостійкість: здатність відновлюватись - нейронні мережі можуть заповнити пробіли, коли значна частина мережі втрачена або відсутня.

До основних завдань, що вирішують нейронні мережі можна віднести:

- класифікація: нейронні мережі відносять шаблони або набори даних в

заздалегідь визначені класи;

- прогноз: вони визначають майбутній результат на основі вхідних даних;

- кластеризація: вони ідентифікують унікальну особливість властиву даним і класифікують її, не маючи попередніх даних;

- зв'язування: нейронна мережа «запам'ятовує» шаблони. При подачі на вхід незнайомої версії патерну, пов'язує її з найбільш подібною в пам'яті.

Найбільшим аспектом нейронних мереж є те, що після тренування навчання не закінчується, а продрвжується в процесі їх використання.

Структури нейронної мережі залежить від особливостей та складності завдання. Для деяких задач існує оптимальна конфігурація мережі, але якщо жодна з існуючих мереж не підходить, потрібна розробка нової структури, яка залежить від конкретної задачі.

В даний час існує два підходи до реалізації ШНМ - апаратна та програмна. Програмна програючи апаратній в швидкості роботи та автономності, має переваги, в простоті використання та можливості інтеграції інформаційно-керуючу систему.

Незважаючи, що роботи по реалізації нейронних мереж проводяться багато років, знайти придатну для використання в розроблюваній системі бібліотеку програм, що дозволяє моделювати нейронну мережу, досить складно.

Серед популярних засобів моделювання штучних нейронних мереж можна виділити:

- 1) Matlab Neural Network.
- 2) Stuttgart Neural Network Simulator;
- 3) NeuroSolutions;
- 4) Fast Artificial Neural Network Library;
- 5) Jets Neural Library;
- 6) Excel Neural Package.

Як правило це готові продукти або бібліотеки, написані для

використання в середовищі Unix. Найпоширенішим інструментом є Matlab Neural Network Toolbox. Нейронні мережі в Matlab, через можливість моделювання широкого класу нейронних мереж, громіздкі в реалізації та мають не високу швидкість при вирішенні конкретних завдань, крім того цей пакет є платним.

Всі інші перераховані пакети для моделювання НМ мають обмеження як по кількості шарів, так і синоптичних ваг. Все це ускладнює їх застосування при прогнозуванні та в інших вузькоспеціалізованих предметних областях.

Очевидно, що при використанні нейронної мережі, оптимізована під конкретне завдання програмна реалізація ефективніша, ніж універсальні пакети моделювання мереж.

При застосуванні нейронних мереж слід враховувати, що їх ефективність пов'язана з точністю даних, та їх відповідністю задачі та цілі прогнозування. Вони здатні встановлювати між факторами нелінійні та складні зв'язки, але не можуть знайти зв'язки, якщо їх немає, або можуть бути неефективними, якщо фактори не є інформативними для прогнозування. На представленій вибірці певні фактори можуть погіршувати якість прогнозування. Перш за все можна не врахувати деякі важливі фактори, або їх різну важливість в тестовій і тренувальній вибірках. Так навіть при однакових наборах факторів використання даних із однієї країни для прогнозування виборів в іншій можна не врахувати специфічні у цій країні фактори, що впливають на результат виборів. Така ж ситуація може бути із застарілими даними чи наборами факторів, як при використанні даних виборів минулих років. Буває некоректно збалансованою тренувальна вибірка коли туди потрапить багато випадків із однаковим результатом і в результаті мережа буде віддавати перевагу такому результату. Прямий поділ даних на тренувальну та тестову вибірки може не врахувати залежності результату від параметру, за яким дані були відсортовані. Щоб уникнути цієї проблеми, при великому наборі даних, їх перемішують випадковим чином

перед поділом. Не слід забувати і про інформативність факторів, що представлені у вибірці, що можна перевірити наявність кореляції між зміною фактору та результату на вибірці при великій кількості записів.

Аналіз існуючих методів прогнозування показав, що їх застосування пов'язане з деякими труднощами. Одні дають значні похибки в оцінці величин, інші не мають широкого застосування через складний математичний апарат..

Моделювання є складним через те що існуючі методи прогнозування не можуть працювати з неповними або неточними даними, водночас на практиці доводиться обходитись саме такою інформацією. Тому потрібні нові методи для прогнозування, які б могли використовувати дані різного роду, в тому числі з неповними вхідними даними.

Визначені технології дозволяють організувати апарат прогнозування, але значний вплив на розробку методу прогнозування має набір даних. В умовах сталої демократії такі дані є однорідними і легко піддаються адаптації для створення тренувальної і тестової вибірок. Зовсім інша ситуація в Україні, за часи незалежності кілька разів мінялось виборче законодавство, повністю змінені виборчі округи. При такій ситуації дані про вибори в різні роки не спів розмірні, певні фактори втратили свою інформативність, тому використовувати їх необхідно тільки після попередньої обробки, тому основними даними лишаються дані соціологічних досліджень.

## Висновки до розділу 1

При написанні розділу:

1. Досліджено предметну область, зокрема розглянуті класичні моделі електоральної поведінки.
2. Визначені існуючі методи прогнозування та їх переваги та недоліки.



3. Розглянуті основні архітектури нейронних мереж.
4. Визначено основні інструменти розробки методу прогнозування.
5. Обґрунтована актуальність теми та зроблена постановка задачі, що передбачає застосування нейронних мереж для розробки методу прогнозування результатів виборів.

## 2 ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ВИБОРІВ

### 2.1 Основні методи навчання нейронних мереж

Нейронні мережі використання є досить широким, але спочатку необхідно, щоб провести навчання ( тренування) мережі[23]. Класифікація нейронних мереж в залежності від характеру навчання є наступною:

- Навчання з вчителем. В тренувальній множині розміщуються очікувані результати разом з даними.
- Навчання без вчителя на вхід подаються вхідні вектори при цьому в процесі навчання, ваги мережі підлаштовуються так щоб близькі за значенням вектори давали однакові виходи. При цьому подібні вхідні вектори об'єднуються в класи.
- Навчання з підкріпленням, система навчається, взаємодіючи з зовнішнім середовищем, що подає сигнали підкріплення, це можна вважати варіантом навчання з вчителем.

Навчити нейромережу – значить, повідомити їй, чого від неї очікуємо. При навчанні мереж ми діємо абсолютно аналогічно(рисунок 2.1). Існує навчальна вибірка. Подаючи дані на вхід мережі, ми одержуємо відповідь, не обов'язково вірну. Вірна відповідь відома тому на виході вірного класу значення має бути максимальним.



Рисунок 2.1 - Ілюстрація процесу навчання НМ

Розрахувавши відхилення отримуємо числа (по кількості класів) – вектор помилки. Алгоритм навчання використовуючи вектору помилки проводить поправки для ваг мережі.

Після проходження достатньої кількості прикладів ваги мережі стабілізуються і відповіді мережі правильні. Функція помилки поступово мінімізується і коли стає незначною або досягає нуля, навчання зупиняється. Вся інформація про задачу для мережі, є лише в прикладах, тому результат навчання залежить від їх кількості і охоплення задачі ними.

З точки зору математики. в процесі функціонування нейронна мережа генерує вихід  $Y$  на основі входу  $X$ , за функцією  $Y = G(X)$ . При відомій архітектурі мережі вид функції  $G$  визначається синаптичними вагами і зсувами мережі[24]. Якщо рішенням є функція  $Y = F(X)$ , задана парами даних  $(X_1, Y_1), (X_2, Y_2), \dots, (X_N, Y_N)$ , для яких  $Y_k = F(X_k)$  ( $k = 1, 2, \dots, N$ ), то завдання навчання - синтез функції  $G$ , близької до  $F$  при помилці  $E$ , при цьому навчання нейронної мережі перетворюється багатовимірну оптимізацію дуже великої розмірності.

Найчастіше використовують алгоритми[24]:

- градієнтний алгоритм (метод найшвидшого спуску);;
- метод Ньютона, метод Гауса-Ньютона, метод Льовенберга-Марквардта і інші;

- стохастичної оптимізації (метод Монте-Карло);
- глобальної оптимізації(перебір значень змінних, що впливають на функцію помилки  $E$ ).

Алгоритм зворотного розповсюдження найчастіше використовується для навчання послідовних багатошарових НМ. Нейрони проводять зважене сумування вхідних сигналів; враховуючи зсув нейрона. Активаційна функція проводить нелінійне перетворення.

У багатошарових мережах відомі лише оптимальні значення на виході, і багатошаровий персептрон вже неможливо навчити, використовуючи лише величини помилок. Тому використовується градієнтний метод пошуку мінімуму функції помилки в протилежному (зворотньому) напрямку щодо сигналів. Цей алгоритм навчання НМ одержав назву процедури зворотного розповсюдження. Функція помилки є сумою квадратів помилки мереж. Елементи вектора-градієнта розраховуються за своєрідним видом похідних функцій активації сигмоїдального типу [25]. Алгоритм діє циклічно, на кожній епосі на вхід подаються навчальні спостереження, вихідні значення мережі порівнюються з бажаними і вираховується помилка, потім проводиться корегування вагів, після чого всі дії повторюються. Приведемо схему алгоритму (рисунок 2.2).

Обчислення у мережі виконуються пошарово. В блоці 5 отримується значення помилки, при цьому кожний з виходів віднімається з цільового вектора, вона використовується в блоці 7 для корегування ваг.

Блоки 3-4 це «прохід вперед», адже сигнал розповсюджується по від входу до виходу. Кроки 5-7 складають «зворотний прохід», тут обчислюваний сигнал помилки іде назад по мережі.



Рисунок 2.2 – Схема алгоритму зворотного розповсюдження помилки

У алгоритмі обчислюється вектор градієнта поверхні помилок. Він вказує напрямок найкоротшого спуску з даної точки, якщо ми просуватись по ньому, помилка зменшиться. Така послідовність приведе до мінімуму одного із типів. При великому кроці збіжність буде швидшою, але можна перестрибнути через рішення.

Зниження точності глобального мінімуму функції похибки свідчить, що не всі системи можна оптимізувати за швидкістю його пошуку. При навчанні нейронна мережа обчислює функцію похибки та коригує ваги

кожного шару, тому при збільшенні кількості шарів, кількість ітерацій також зростає. Визначення параметрів нейронної мережі є важливою при створенні навчальної моделі. В основному необхідно зосередитись на двох основних: кількості шарів та темпі навчання.

Мережі глибинного навчання, в основному використовують оптимізатор стохастичного градієнтного спуску, він дозволяє задати швидкість навчання повідомляючи оптимізатору, наскільки перемістити ваги до градієнта. При низькій швидкості тренування буде надійнішим, але на оптимізацію потрібно буде багато часу через невеликі кроки. В іншому випадку навчання може не сходитися через великий крок оптимізатор перевищує мінімум.

Починати потрібно з відносно великої швидкості навчання, оскільки на початку ваги не оптимальні, а потім швидкість потрібно зменшувати, щоб більш точно оновлювати ваги.

Існують різні способи вибору відправної точки для встановлення темпу навчання. Найкращий полягає в опробуванні декількох різних значень і виборі того що дає найкращу мінімізацію втрат, не зменшуючи швидкості тренувань.

Розглянутий алгоритм навчання за допомогою зворотного розповсюдження передбачає наявність зовнішньої ланки, надаючи мережі не лише вхідні, а також і цільові образи. Алгоритми, побудовані за таким принципом називаються алгоритмами навчання з вчителем. Для їх функціонування необхідна присутність експертів, що на етапі підготовки створюють для кожного вхідного еталонний вихідний.

Головне, що робить навчання без вчителя привабливим – це його «самостійність». Процес навчання, як і у разі навчання з вчителем, полягає в підстроюванні вагів мережі.

## 2.2 Використання нейронних мереж і нечіткої логіки для прогнозування

Перевага нейронних мереж очевидна через їх універсальність та простоту. Також вони не вимогливі до набору даних, самі визначають і встановлюють зв'язки та формують їх структуру. Тому одну нейронну мережу можна використати в схожих задачах прогнозування результатів виборів, при умові, що в факторах та даних будуть незначні відмінності. Для ефективного використання потрібен правильний вибір параметрів мережі та архітектури, і фільтрація та попередня обробка даних з перевіркою їх інформативності.

Використання нейронних мереж для прогнозування результатів виборів проводилось різними вченими, так Горбан та Ваксман виконали роботу для Світового конгресу з нейронних мереж де вивчали питання прогнозування виборів на основі нейронних мереж, де вони прогнозували результати виборів президента США в 1992 році, за основу було взято дані за 1860-1980 роки [9], Г. С. Гілл в своїй роботі побудував модель із застосуванням двошарового перцептрона для прогнозування виборів до нижньої палати парламенту Індії - Лок Сабха при обмежених, через їх недоступність, даних про попередні вибори та неможливістю проведення опитувань, враховуючи кількість населення [10]. Для визначення прогнозу результатів виборів у Великій Британії Р. Борисюк, Г. Борисюк з колегами в якості даних використали відповіді експертів на питання щодо виборів 1835-1997 років [11].

В силу специфіки прогнозування, як науки своє місце тут знайшли і мережі із нечіткою логікою (FN, Fuzzy Network) або нейро-нечіткі мережі – це поєднання НМ та системи із нечіткою логікою. Їх математичне представлення отримується в результаті представлення функції приналежності в якості функції активації, а норм нечіткої логіки як виду нейронів з визначеними операціями.

Такі мережі можна описати формою апроксимації або теоремою Цибенко [12] таким чином:  $Y(x) = \sum_{i=1}^N f_i(x) * \theta_i$ .

Мережі із нечіткою логікою є універсальними апроксиматорами, вони об'єднують переваги нейронних мереж в швидкості навчання та можливості відтворити схему людського мислення. Такі системи дозволяють досягнути таких показників як інтерпретованість та точність, звісно в реальності одночасно обидва недосяжні, тому при моделюванні нейронечіткої мережі необхідно вибирати основну ціль.

Останнім часом перебуває в тренді проблематика генетичних алгоритмів, нейронних мереж і нечітких систем, які відносяться до області науки, що в англійській літературі характеризується терміном Computational Intelligence (рисунок 1.12). [22].



Рисунок 2.3 – Computational Intelligence

Їх задачі можуть розглядатися поза межами зв'язків між ними, але їх взаємозалежність є надзвичайно важливою. В практиці застосування поєднання технологій може відбуватись різними шляхами, Можливе



використання однієї технології в якості допоміжної або контролюючої, або паралельне вкочання в інтегрованих системах

В нечітких нейронних мережах результат отримується на основі апарату нечіткої логіки, а дл функцій приналежності параметри встановлюються в результаті навчання нейронної мережі.

Модуль нечіткого управління представляється в формі багат шарової мережі, в якій шари виконують функції елементів системи нечіткого виведення (рисунок. 2.4).

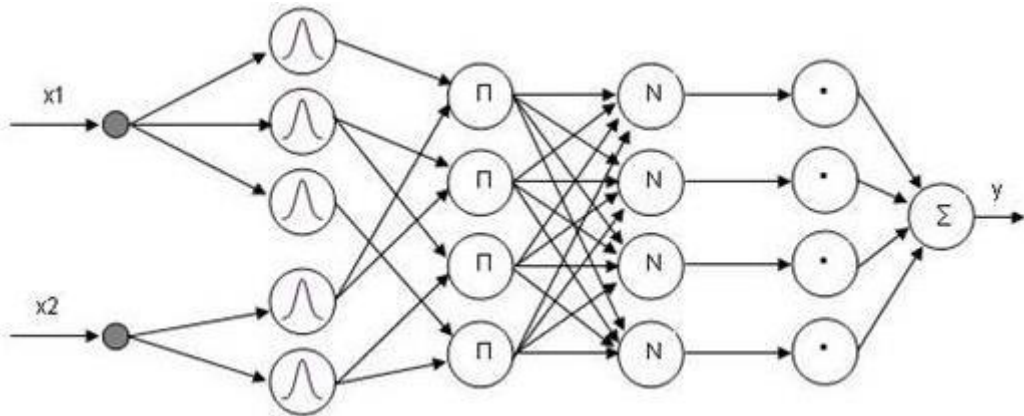


Рисунок 2.4 – Структура нейро-нечіткої мережі

До основних властивостей нейро-нечітких мереж відноситься:

- нейро-нечіткі мережі ґрунтуються на нечітких системах, які навчаються за тих самих методів що й нейромережі;
- нейро-нечітка мережа як правило є багат шаровою. Перший - вхідний шар, середній містить нечіткі правила, а останній містить вихідні змінні;
- нейро-нечітка мережа в будь-який момент може бути представлена як система нечітких правил;

Як видно, нейрон-нечіткі мережі розширюють функціонал нейронних мереж і коло задач, які можуть виконуватись.

## 2.3 Використання бібліотек при розробці моделі мережі

Вирішення завдань побудови мереж використовують бібліотеки (фреймворки) серед них можна виділити TensorFlow, Torch, Theano, Caffe та інші. Однією з цікавих реалізацій бібліотек є Keras, яка є надбудовою над TensorFlow і Theano. В бібліотеках містяться вже готові шари, функції обробки і алгоритми оптимізації та моделі різних нейронних мереж.

На сьогоднішній день у реалізації нейронних мереж важливу роль відіграють, спеціалізовані пакети, містять вже заздалегідь реалізовані базові структури для зручної розробки. Найбільш часто використовуються такі популярні бібліотеки, як Theano, Tensorflow та Torch.

Варто відзначити, що Torch реалізує інтерфейс для мови Lua, що є не дуже популярно. Theano and Torch показують приблизно однакову продуктивність. Окрім описаної вище незручності використання бібліотеки Torch, існує ще одна, також пов'язана зі зручністю розробки. Інші два інструменти здатні працювати з бібліотекою Keras, що надає визначені наперед алгоритми та структури. За результатами порівняння та аналізу документації бібліотек було прийнято рішення використовувати для програмної реалізації нейронної мережі зв'язок бібліотек Tensorflow[26] і Keras.

Даний продукт створювався з метою запобігання труднощів в розробці нейронних мереж, пов'язаних із синтаксичними особливостями конкретних пакетів. В січні 2016 року даний інструмент було придбано компанією Google і адаптовано як основне надлаштування для Tensorflow.

Keras — це API глибокого навчання, написаний на Python, який працює поверх платформи машинного навчання TensorFlow. Він був розроблений з метою надати можливість швидкого експериментування. Швидкий перехід від ідеї до результату є ключем до хорошого дослідження. Keras - це:

Просто -- але не спрощено. Keras зменшує когнітивне навантаження розробника, щоб ви могли зосередитися на тих частинах

проблеми, які дійсно важливі.

Гнучкість — Keras використовує принцип прогресивного розкриття складності: прості робочі процеси мають бути швидкими та легкими, а розширені робочі процеси мають бути можливими за допомогою чіткого шляху, який базується на тому, що ви вже навчилися.

Потужність — Keras забезпечує потужну продуктивність і масштабованість: його використовують організації та компанії

Keras дає можливість інженерам та дослідникам повною мірою скористатися можливостями масштабованості та кросплатформності TensorFlow 2: ви можете запускати Keras на TPU або на великих кластерах графічних процесорів, а також можете експортувати свої моделі Keras для запуску у браузері або на мобільному пристрої. пристрій.

#### Keras і TensorFlow 2

TensorFlow 2 — це наскрізна платформа машинного навчання з відкритим кодом. Ви можете розглядати це як рівень інфраструктури для диференційованого програмування. Він поєднує в собі чотири ключові здібності:

- Ефективне виконання низькорівневих тензорних операцій на CPU, GPU або TPU.
- Обчислення градієнта довільних диференційованих виразів.
- Масштабування обчислень для багатьох пристроїв, таких як кластери із сотень графічних процесорів.
- Експорт програм ("графів") у зовнішні середовища виконання, такі як сервери, браузери, мобільні та вбудовані пристрої.

Keras/TensorFlow сумісні з:

- Python 3.6–3.9
- Ubuntu 16.04 або новішої версії
- Windows 7 або новішої версії
- macOS 10.12.6 (Sierra) або новішої версії.

Keras — це високорівневий API TensorFlow 2: доступний,

високопродуктивний інтерфейс для вирішення проблем машинного навчання, зосереджений на сучасному глибокому навчанні. Він надає основні абстракції та будівельні блоки для розробки та доставки рішень машинного навчання з високою швидкістю ітерацій.

Keras дає можливість інженерам та дослідникам повною мірою скористатися можливостями масштабованості та кросплатформенності TensorFlow 2: ви можете запускати Keras на GPU або на великих кластерах графічних процесорів, а також можете експортувати свої моделі Keras для запуску у браузері або на мобільному пристрої. пристрій.

Бібліотека Theano [27] була призначена для розширення мови Python, яка дозволяє ефективно розрахувати математичні вирази, що містять багатомірні масиви. В бібліотеці реалізується базовий набір інструментів для побудови нейронних мереж. Процес створення моделі та визначення її параметрів вимагає написання об'ємного коду, що включає реалізацію класу моделі, самостійного визначення її параметрів, реалізації методів, що визначають функцію помилок, правило обрахування градієнтів, способи зміни ваги нейронів. Бібліотека Caffe[28] реалізована на мові програмування C++. Топологія нейромереж, вихідні дані та спосіб навчання задаються у прототиповому форматі. Побудова мережі виконується з простою та зручністю. Бібліотека Caffe є самою розповсюдженою бібліотекою глибинних навчальних програм. Через те, що в останні роки методи машинного навчання отримали велику популярність і застосовуються в безлічі різних областей, існує великий вибір бібліотек для реалізації моделей машинного навчання.

Проте з таких загальних критеріїв неможливо зробити висновок, який саме фреймворк застосовувати, оскільки вони не відповідають на питання, чи складно створювати систему із застосуванням цього фреймворку, наскільки швидко навчатиметься система.

Для реалізації нейронної мережі була обрана мова програмування Python. Python – гнучка мова програмування високого рівня, створена в 1991

році, що ставить за свої цілі низький «рівень входу» та легкість результуючих програм. Мова програмування Python має кілька версій. Python є об'єктно-орієнтованою мовою програмування високого рівня. Додаткову зручність в процесі розробки дає динамічна типізація, а також значний функціонал як вбудованих бібліотек, так і додаткових модулів та пакетів модулів. Зручним є те, що для мови програмування Python вже існує велика кількість модулів, в тому числі орієнтованих на розробку нейронних мереж, крім того вона є кросплатформенною, тобто може працювати на різних операційних системах та апаратних платформах, що позбавляє необхідності в адаптації програмного продукту та знижує вартість розробки.

#### Висновки до розділу 2

На основі проведених досліджень можна зробити висновки, що прогнозування результатів виборів звичними методами є досить трудомістким процесом, який не завжди дає позитивний результат, найкращі результати при цьому отримуються при використанні нейронних мереж. Але специфіка задачі не дозволяє в повній мірі скористатись перевагами в основному через відмінності технологій і наборів даних в різних країнах і регіонах. В результаті:

1. Визначено основні архітектури неронних мереж
2. Досліджено алгоритми навчання нейронної мережі, зокрема методу зворотного поширення похибки
3. Досліджено основні бібліотеки

## 3 ЕСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

### 3.1 Підготовка набору даних для навчання мережі

Процес навчання нейронної мережі залежить від якості вхідних даних, як вже зазначалось статистика виборів в Україні мало придатна для створення повноцінного прогнозу через свій малий обсяг і різну структурованість в кожному періоді через зміни виборчого законодавства. Тому скористаємось для створення мережі статистикою виборів в США Дані, які використовуються для навчання нейронної мережі, отримані з Кооперативного дослідження виборів до Конгресу -Cooperative Congressional Election Study (CCES), що є одним з найбільших політичних досліджень в США. Цей набір дані містить інформацію опитування респондентів із загального змісту CCES (п = 531,755), що поєднують в собі весь доступний загальний контенті datasets до 2020 року. Він включає вибрані стандартизовані змінні, включаючи демографічні показники, географію, вибір голосування, підтвержене голосування, схвалення представників, економіку тощо. Опитування проводиться серед обраних респондентів у всіх штатах кожного року, а в рік виборів в два етапи. Датасет був зібраний Шіро Курівакі в форматі Harvard Dataverse, і є у вільному доступі, його буде використано для побудови нейронної мережі визначення уподобань виборців. Крім того для використання в інформаційних системах дані CCES для зручності представляються в розділеному виді по основних показниках у формі фалів .csv, розділених по категоріях, для нас представляють інтерес файли county\_facts.csv, що містить демографічні та інші дані, county\_facts\_dictionary.csv – словник факторів, та primary\_results.csv – результати виборів.

Такий формат зумовлений тим, що система виборів в США є непрямую, двоступеневою. Президент обирається колегією виборників кількість яких пропорційна населенню штату, кожні 4 роки. Перемога у виборах визначається більшістю голосів.

Якщо більшості ніхто не отримав, президента обирають члени Палати представників серед кандидатів з найбільшою кількістю. В останні десятиліття переважна кількість голосів залишається розподіленою між всього двома кандидатами. Така система характерна для федеративної республіки і при цьому врахована думка усіх штатів та округів

Для прогнозування результатів виборів по різних округах використано їх економічні та демографічні показники. До відносяться чисельність населення та динаміка її зміни, частка(відсоток) по вікових групах, расах та національностях, середній дохід, особливості господарств, кількість та форма власності фірм.

До використання для прогнозування саме нейронних мереж спонукає велика кількість факторів впливу, що можуть мати нелінійні залежності, що може викликати труднощі для класичних методів прогнозування.

Задача прогнозування є задачею класифікації, оскільки полягає в прогнозуванні того, який кандидат із можливих буде обраний в кожному окремому окрузі. Більшість голосів дісталась представникам двох партій, тож задача приймає вигляд бінарної класифікації.

Для використання в моделюванні вихідні дані повинні попередньо оптимізовані для коректної роботи з ними нейронної мережі.

Вхідні дані містять не актуальні для прогнозування текстові змінні, числові дані ненормалізовані та деякі фактори можуть не мати впливу на вибір. При оптимізації із демографічних даних було видалено підсумки по штатах, адже ми будемо працювати із даними лише по окремих округах. Приклад частини вихідних даних із файлу `county_facts.csv` наведено в таблиці 3.1.

В наборах даних були наявні входження із пустими значеннями в деяких стовпцях, є наслідком помилок в процесі збору вихідних даних, отже їх можна видалити.

Закодовані для компактності назви факторів демографічних даних повернуто до повноцінного виду за допомогою словника (таблиця 3.2)

Таблиця 3.1 – Частина вихідних демографічних даних

fips	area_name	state_abbreviation	Pop_2014_count	Pop_2010_base_count	Pop_change_pct	Pop_2010_count
1001	Autauga County	AL	55395	54571	1.5	54571
1003	Baldwin County	AL	200111	182265	9.8	182265
1005	Barbour County	AL	26887	27457	-2.1	27457
1007	Bibb County	AL	22506	22919	-1.8	22915
1009	Blount County	AL	57719	57322	0.7	57322
1011	Bullock County	AL	10764	10915	-1.4	10914
1013	Butler County	AL	20296	20946	-3.1	20947
1015	Calhoun County	AL	115916	118586	-2.3	118572
1017	Chambers County	AL	34076	34170	-0.3	34215
1019	Cherokee County	AL	26037	25986	0.2	25989
1021	Chilton County	AL	43931	43631	0.7	43643
1023	Choctaw County	AL	13323	13858	-3.9	13859
1025	Clarke County	AL	24945	25840	-3.5	25833
1027	Clay County	AL	13552	13932	-2.7	13932
1029	Cleburne County	AL	15080	14972	0.7	14972
1031	Coffee County	AL	50909	49948	1.9	49948
1033	Colbert County	AL	54543	54428	0.2	54428
1035	Conecuh County	AL	12670	13228	-4.2	13228
1037	Coosa County	AL	10886	11758	-7.4	11539
1039	Covington County	AL	37914	37765	0.4	37765
1041	Crenshaw County	AL	13977	13906	0.5	13906
1043	Cullman County	AL	81289	80410	1.1	80406
1045	Dale County	AL	49484	50251	-1.5	50251
1047	Dallas County	AL	41711	43820	-4.8	43820
1049	DeKalb County	AL	71065	71115	-0.1	71109
1051	Elmore County	AL	80977	79296	2.1	79303
1053	Escambia County	AL	37733	38319	-1.5	38319
1055	Etowah County	AL	103531	104427	-0.9	104430
1057	Fayette County	AL	16874	17241	-2.1	17241
1059	Franklin County	AL	31601	31709	-0.3	31704
1061	Geneva County	AL	26712	26790	-0.3	26790
1063	Greene County	AL	8553	9045	-5.4	9045
1065	Hale County	AL	15184	15760	-3.7	15760
1067	Henry County	AL	17190	17302	-0.6	17302
1069	Houston County	AL	104193	101547	2.6	101547
1071	Jackson County	AL	52665	53226	-1.1	53227
1073	Jefferson County	AL	660793	658350	0.4	658466
1075	Lamar County	AL	14086	14564	-3.3	14564
1077	Lauderdale County	AL	93096	92709	0.4	92709
1079	Lawrence County	AL	33477	34339	-2.5	34339
1081	Lee County	AL	154255	140296	9.9	140247
1083	Limestone County	AL	90787	82782	9.7	82782
1085	Lowndes County	AL	10580	11299	-6.4	11299
1087	Macon County	AL	19425	21450	-9.4	21452
1089	Madison County	AL	350299	334811	4.6	334811
1091	Marengo County	AL	20110	21029	-4.4	21027
1093	Marion County	AL	30271	30777	-1.6	30776



Таблиця 3.2 – Фрагмент словника параметрів

column_name	description
Pop_2014_count	Population, 2014 estimate
Pop_2010_base_count	Population, 2010 (April 1) estimates base
Pop_change_pct	Population, percent change - April 1, 2010 to July 1, 2014
Pop_2010_count	Population, 2010
Age_under_5_pct	Persons under 5 years, percent, 2014
Age_under_18_pct	Persons under 18 years, percent, 2014
Age_over_65_pct	Persons 65 years and over, percent, 2014
Sex_female_pct	Female persons, percent, 2014
Race_white_pct	White alone, percent, 2014
Race_black_pct	Black or African American alone, percent, 2014
Race_native_pct	American Indian and Alaska Native alone, percent, 2014
Race_asian_pct	Asian alone, percent, 2014
Race_island_pct	Native Hawaiian and Other Pacific Islander alone, percent, 2014
Race_mixed_pct	Two or More Races, percent, 2014
Race_hispanic_pct	Hispanic or Latino, percent, 2014
Race_white_not_hispanic_pct	White alone, not Hispanic or Latino, percent, 2014
Pop_same_house_pct	Living in same house 1 year & over, percent, 2009-2013
Pop_foreign_born_pct	Foreign born persons, percent, 2009-2013
Pop_nonenglish_home_pct	Language other than English spoken at home, pct age 5+, 2009-2013
Pop_hs_grad_pct	High school graduate or higher, percent of persons age 25+, 2009-2013
Pop_college_grad_pct	Bachelor's degree or higher, percent of persons age 25+, 2009-2013
Pop_veterans_count	Veterans, 2009-2013

На відсотків голосів за кандидатів від демократичної та республіканської партій для кожного округу було створено змінну для класифікації із значенням 0 та 1, що відповідають перемозі того чи іншого кандидата.

Засобами бібліотеки pandas було виконано ліве зовнішнє приєднання набору демографічних даних та набору даних про результати виборів (таблиця 3.3).

В об'єднаній набір попали різні за своєю природою змінні, відповідно ці дані потребували нормалізації.

Дані, представлені у вигляді відсотків в 0 до 100, були приведені до вигляду відсотків на інтервалі  $[0;1]$ .

Інші числові дані, такі як кількість населення і подібні також були нормалізовані до інтервалу  $[0;1]$  за допомогою масштабування особливостей.

Таблиця 3.3 – Фрагмент даних результатів голосування

state	state_abbreviatio n	county	fips	party	candidate	votes	fraction_votes
Alabama	AL	Autauga	1001.0	Democrat	Bernie Sanders	544	0.182
Alabama	AL	Autauga	1001.0	Democrat	Hillary Clinton	2387	0.8
Alabama	AL	Baldwin	1003.0	Democrat	Bernie Sanders	2694	0.3289999999999999
Alabama	AL	Baldwin	1003.0	Democrat	Hillary Clinton	5290	0.647
Alabama	AL	Barbour	1005.0	Democrat	Bernie Sanders	222	0.078
Alabama	AL	Barbour	1005.0	Democrat	Hillary Clinton	2567	0.9059999999999999
Alabama	AL	Bibb	1007.0	Democrat	Bernie Sanders	246	0.1969999999999999
Alabama	AL	Bibb	1007.0	Democrat	Hillary Clinton	942	0.755
Alabama	AL	Blount	1009.0	Democrat	Bernie Sanders	395	0.386
Alabama	AL	Blount	1009.0	Democrat	Hillary Clinton	564	0.551
Alabama	AL	Bullock	1011.0	Democrat	Bernie Sanders	178	0.066
Alabama	AL	Bullock	1011.0	Democrat	Hillary Clinton	2451	0.9129999999999999
Alabama	AL	Butler	1013.0	Democrat	Bernie Sanders	156	0.065
Alabama	AL	Butler	1013.0	Democrat	Hillary Clinton	2196	0.9209999999999999
Alabama	AL	Calhoun	1015.0	Democrat	Bernie Sanders	1425	0.218
Alabama	AL	Calhoun	1015.0	Democrat	Hillary Clinton	5011	0.765
Alabama	AL	Chambers	1017.0	Democrat	Bernie Sanders	312	0.095
Alabama	AL	Chambers	1017.0	Democrat	Hillary Clinton	2899	0.8859999999999999
Alabama	AL	Cherokee	1019.0	Democrat	Bernie Sanders	268	0.249
Alabama	AL	Cherokee	1019.0	Democrat	Hillary Clinton	712	0.6609999999999999
Alabama	AL	Chilton	1021.0	Democrat	Bernie Sanders	289	0.2460000000000000
Alabama	AL	Chilton	1021.0	Democrat	Hillary Clinton	860	0.731
Alabama	AL	Choctaw	1023.0	Democrat	Bernie Sanders	273	0.124
Alabama	AL	Choctaw	1023.0	Democrat	Hillary Clinton	1772	0.802

В результаті набір даних все ще містив дуже значну кількість змінних на впливаючих на вибір по округах. Для виділення таких змінних розрахована кореляція цих змінних із відсотком голосів за одного

кандидатів, частину значень кореляції наведено в таблиці 3.4

Таблиця 3.2 – Кореляція між демографічними показниками та відсотком голосів за кандидата

Демографічний показник	Кореляція
fips	-0.056901
Population_2014	0.349021
Percent_Under_18_Years_Old	-0.068978
Percent_65_Years_and_Older	-0.304966
Percent_Females	0.187450
Percent_White_Alone	-0.590672
Percent_Black_or_African_American	0.511695
Percent_American_Indian_or_Alaska_Native	0.078051
Percent_Asian	0.434992
Percent_Native_Hawaiian_and_Other_Pacific_Islander	0.119524
Percent_Two_or_More_Races	0.167839
Percent_Hispanic_or_Latino	0.181303
Percent_White_Alone_not_Hispanic_or_Latino	-0.580653
Percent_Living_in_same_house_1_year_and_over	-0.168737
Percent_Foreign_born_persons	0.392028
Percent_Language_other_than_English_spoken_at_home	0.316960
Percent_High_school_graduate_or_higher	0.009090
Percent_Bachelors_degree_or_higher	0.436231
Percent_Veterans	0.371094
Mean_travel_time_to_work_minutes	0.101217
Housing_units	0.367998
Percent_Homeownership_rate	-0.469049
Housing_units_in_multi_unit_structures	0.582949
Median_value_of_owner_occupied_housing_units	0.447902
Number_Households	0.365203
Persons_per_household	0.15581

На основі цих даних було видалено показники, що мають дуже низьку кореляцію тобто незначний вплив на результати.

Після цього, отриманий кінцевий набір даних було поділено на навчальний та тестовий набори у співвідношенні 80% до 20%. Контрольний набір буде створюватись із навчального автоматично в ході навчання моделі та становить 10% від навчального набору даних.

### 3.2 Побудова моделі уподобань виборців

Побудова мережі прогнозу результатів виборів дозволяє використати різні підходи і методики. Враховуючи, що результати виборів для кожного їх виду відрізняються, так само буде відрізнятися і мета створення мережі.

Напередодні виборів, політичні аналітики, прогнозисти та інші зацікавлені сторони намагаються отримати якомога більше даних для передбачення результатів, але прогнозування результатів виборів не єдиний варіант. Розглядаючи політичну структуру США можна виділити двопартійну систему, де існує дві конкуруючі партії, але вони, хоч і найпотужніші, але не єдині гравці на політичному ринзі.

Як вже згадувалось вище на результати виборів впливає електоральна поведінка, яка в результаті виражається політичними уподобання виборців. Для створення нейронної мережі скористаємось пакетом RStudio, а саме мовою R. Для цього використаємо бібліотеку Keras, і хоч вона написана для Python, без проблем інтегрується в RStudio.

Для початку підключаємо пакети(рисунок 3.1).

```
install.packages("keras")
install.packages("tidyr")
install.packages("ggplot2")
install.packages("dplyr")
install.packages("fastDummies")
```

Рисунок 3.1 – Підключення пакетів

В якості даних використаємо вже згаданий пакет CCES зібраний Шіро Курівакі, підключаємо його (рисунок 3.2).

```
d <- readRDS("cumulative_2006_2018.Rds")
dim(d)

[1] 452755 73
```

Рисунок 3.2 – Завантаження даних

В результаті проведено імпорт з розмірністю 452 755 рядки і 73 стовпці. Повного переліку факторів для мережі нам не потрібно, тому виберемо лише ті фактори, які представляють інтерес для моделі, а також виключимо з набору порожні дані(рисунок 3.3.).

```
dd <- dd %>%
  select(st, dist, cong, # geography
         gender, birthyr, age, educ,
         race, faminc, marstat, # demographics
         newsint, # news interest
         approval_pres, # approval
         ideo5, # ideology
         voted_pres_16 # presidential vote
  )
dd <- dd[complete.cases(dd),]
```

Рисунок 3.3 – Фільтрація факторів

Результат виконання представлено на рисунку 3.4.

```
> dd
# A tibble: 46,206 x 13
  st   dist cong gender age educ race faminc marstat newsint approval_pres ideo5 voted_pres_16
  <chr> <int> <int> <int> <int> <int> <int> <fct> <int> <int> <int> <fct> <fct>
1 TX    29  115    2   45    3    2 10k - 20k     2     7     3 Not Sure    Hillary Clinton
2 OH     1  115    2   58    6    1 150k+         1     3     2 Conservative Donald Trump
3 KY     3  115    2   66    6    1 10k - 20k     3     1     4 Liberal     Hillary Clinton
4 AZ     9  115    2   88    2    1 20k - 30k     4     2     2 Moderate    Donald Trump
5 PA     3  115    2   59    6    1 10k - 20k     3     1     3 Liberal     Donald Trump
6 NC    11  115    1   71    2    1 50k - 60k     1     2     1 Very Conservative Donald Trump
7 PA     3  115    1   86    5    1 60k - 70k     4     1     1 Conservative Donald Trump
8 TX     3  115    2   80    6    1 60k - 70k     4     1     2 Conservative Donald Trump
9 AZ     5  115    2   45    5    1 120k - 150k  1     1     1 Very Conservative Donald Trump
10 IL    5  115    2   60    3    1 20k - 30k     2     2     4 Moderate    Hilary Clinton
# ... with 46,196 more rows
```

Рисунок 3.4 – Результат виконання фільтрації

Наступним кроком є перетворення даних, для цього скористаємось функцією `dummy_cols` щоб перекодувати не числові змінні(рисунок 3.5)

```
cat_vars <- colnames(dd)
cat_vars <- cat_vars[!cat_vars %in% c("age", "voted_pres_16")]
all_data <- dummy_cols(dd,
  select_columns = cat_vars,
  remove_first_dummy = TRUE,
  remove_selected_columns = TRUE)
```

### Рисунок 3.5 – Перекодування змінних

Тако перетворюємо нашу залежну змінну з уподобаннями `voted_pres_16` в числовий вектор з цілими числами (починаючи з нуля) для кожного з кандидатів і після цього видаляємо змінну `voted_pres_16` з нашого набору даних (рисунок 3.6)

```
all_labels <- dd$voted_pres_16 %>%
  as.factor() %>%
  unclass() - 1

all_data <- all_data %>%
  select(-voted_pres_16) %>%
  as.matrix()
```

### Рисунок 3.6 – Перекодування і видалення результуючої змінної

Після всіх операцій над даними необхідно розділити набір даних на навчальну вибірку (90%) і тестову вибірку (10%), щоб після навчання моделі можна було перевірити її продуктивність незнаних даних

```
party_levels <- levels(all_labels)
elems <- sample(1:nrow(all_data),
              round(0.1*nrow(all_data)),
              replace = F)

# training data
train_data <- all_data[-elems,]
train_labels <- all_labels[-elems]

levels(train_labels) <- levels(all_labels)

# test data
test_data <- all_data[elems,]
test_labels <- all_labels[elems]

levels(test_labels) <- party_levels
```

### Рисунок 3.7 – Розділення вибірки

Тепер можна приступити до побудови моделі, згідно обраної мети необхідно побудувати модель класифікації, де кожен кандидат представляє класифікаційну категорію (всього 5 категорій). Вхідний шар отримує пояснювальні змінні, він з'єднаний з нейронами в прихованому шарі. Нарешті, вихідний шар складається з 5 нейронів, по одному для кожної категорії.

```
# Define model
model <- keras_model_sequential()
model %>%
  layer_dense(units = 100, activation = "relu",
              input_shape = dim(train_data)[2]) %>%
  layer_dense(units = length(party_levels), activation = 'softmax')
```

### Рисунок 3.8 – Побудова моделі

Функція активації для першого етапу (вхід до прихованого шару) є ReLu, тоді як функція активації для другого етапу (прихований до вихідного шару) є softmax.

Тепер ми приступаємо до компіляції та навчання моделі. Алгоритм оптимізатора, який ми будемо використовувати тут, - це *adam*, алгоритм адаптивної оптимізації, який зазвичай використовується для навчання глибоких нейронних мереж. З навчальної вибірки виділимо 1/5 навчальних даних для моделі, щоб ітеративно обчислювати помилку перевірки (рисунок 3.9).

Алгоритм буде працювати з нейронною мережею протягом 500 епох, алеї може зупинитись раніше, якщо продуктивність моделі не збільшиться протягом 20 безперервних епох.

```
##### Compile the model #####
model %>% compile(
  optimizer = 'adam',
  loss = 'sparse_categorical_crossentropy',
  metrics = c('accuracy')
)

##### Train the Model #####
early_stop <- callback_early_stopping(monitor = "val_loss",
                                       patience = 20)

model %>% fit(train_data,
             train_labels,
             validation_split = 1/5,
             callbacks = list(early_stop),
             epochs = 500)
```

Рисунок 3.9 – Створення і навчання моделі

Після навчання нашої моделі необхідно оцінити її за допомогою тестових даних, роблячи прогноз та перевіряючи продуктивність моделі (рисунок 3.10).

```
##### Evaluate Model #####
score <- model %>% evaluate(test_data, test_labels, verbose = 0)

cat('Test loss:', score$loss, "\n")
cat('Test accuracy:', score$acc, "\n")
```

Рисунок 3.10 – Тестування моделі

Після виконання отримали наступні результати(рисунок 3.11) Втрати тесту і точність 0,4882 і 0,8461 відповідно.

```
> cat('Test loss:', score$loss, "\n")
Test loss: 0.4882013
> cat('Test accuracy:', score$acc, "\n")
Test accuracy: 0.8461039
```

Рисунок 3.11 – Результати тестування моделі

Детальне уявлення про продуктивність моделі показано рисунку 3.12, що містить матрицю невідповідностей нашої моделі. Правильні показники



класифікації (висока точність по діагоналі та низькі значення за межами) пофарбовані в зелений колір, а неправильні показники класифікації (низькі значення на діагоналі та високі значення за межами) пофарбовані в червоний колір.



Рисунок 3.12 – Матриця невідповідностей моделі

Якщо уважно подивитись на матрицю невідповідностей, то видно що модель не робила правильних прогнозів для категорій «Не голосував» та «Не впевнений/Не пам'ятаю». Це пов'язано з невеликою кількістю спостережень, що належали до цих категорій порівняно з іншими категоріями.

Крім того, може здатися дивним, що модель мала дуже низьку продуктивність, відносячи виборців до категорії «інші». Тим не менш, важливо відзначити, що ця категорія включає дуже різноманітний пул кандидатів. Ці кандидати мають різноманітну політичну ідеологію та представляють неоднорідну суміш уподобань виборців та демографії.

Отже, ми побудували модель, яка прогнозує уподобання виборців, що є попереднім етапом перед прогнозуванням результатів виборів.

Використавши прогнозування уподобань по окремих округах можна отримати набір результатів для використання в мережі прогнозування.

### 3.3 Оцінка результатів експерименту

В задачі прогнозування результатів виборів, враховуючи велику кількість факторів для проведення прогнозування, поставлена мета побудувати модель із високою точністю прогнозу, без перенавчання та не дуже високої складності. архітектура побудованої моделі у вигляді визначених шарів представлена на рисунку 3.13.

В результаті отримано нейронну мережу прямого поширення, що складається з чотирьох основних шарів, двох прихованих шарів, вхідного та вихідного. Шари моделі організовані повнозв'язними, нейрони цих шарів зв'язані з кожним нейроном попереднього шару. Активаційною функцією обрано функцію ReLU через її біологічну відповідність і розріджену активаці, крім того вона є масштабованою та дуже швидкою для обчислень і на відміну від сигмоїди не має проблем із зникненням градієнту.

Щоб мережа не перенавчалась використано метод регуляризації Dropout (метод виключення). Для прискорення роботи та стабільності мережі використовується вбудований в Keras алгоритм пакетної нормалізації (Batch Normalization). він нормалізує входи шарів нейронної мережі за допомогою повторного центрування та масштабування. Алгоритм передбачає додатковий крок нормалізації.

```

Model: "sequential"
Layer (type)                Output Shape                Param #
-----
dense (Dense)                (None, 32)                  1056
batch_normalization (BatchN (None, 32)                  128
activation (Activation)      (None, 32)                  0
dense_1 (Dense)              (None, 64)                  2112
batch_normalization_1 (Batch (None, 64)                  256
activation_1 (Activation)    (None, 64)                  0
dropout (Dropout)           (None, 64)                  0
dense_2 (Dense)              (None, 32)                  2080
batch_normalization_2 (Batch (None, 32)                  128
activation_2 (Activation)    (None, 32)                  0
dense_3 (Dense)              (None, 1)                   33
-----
Total params: 5,793
Trainable params: 5,537
Non-trainable params: 256
None

```

Рисунок 3.13 – Представлення шарів моделі

Оптимізація при навчанні відбувається за рахунок використання алгоритму стохастичного градієнтного спуску, що використовує оцінку градієнта замість його реального значення. В процесі навчання мережі від вхідних даних із навчальною вибіркою відділяється 10% для контрольної вибірки для перевірки точності моделі.

Для аналізу результатів роботи програми проведемо дослідження за кількості епох рівній 1000, що є компромісним значенням між точністю результатів та швидкістю навчання моделі. Лістинг програми представлений додатку Б.

Для демонстрації точності роботи моделі також можна використати матрицю невідповідностей моделі, що представлена на рисунку 3.14.

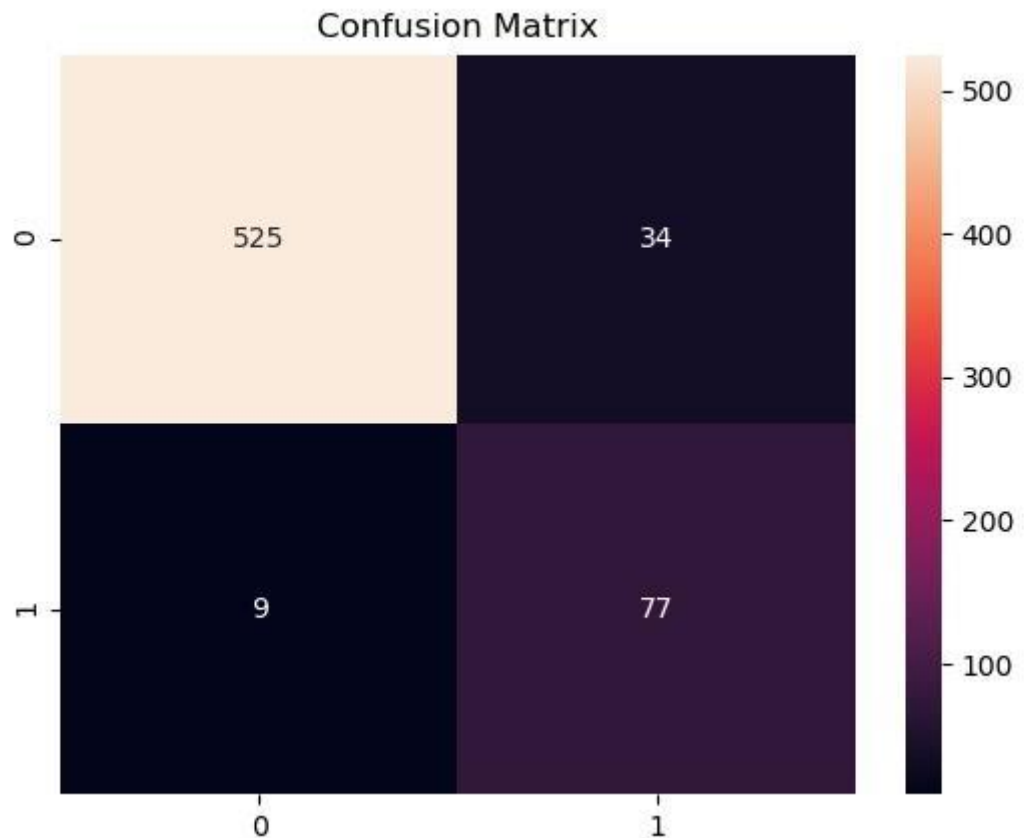


Рисунок 3.14 – Матриця невідповідностей моделі

В представленій матриці значення на перетині 0 і 0 та 1 і 1 відповідають правильно отриманим значенням прогнозу відповідно для класів 0 та 1, елементи що знаходяться на перетині 0 і 1 та 1 і 0 – відповідають неправильно отриманим прогнозам значень відповідних класів 0 і 1

Загальна точність мережі склала 95,7% при отриманій точності на контрольній вибірці 90,7%. Мережа не є схильною до перенавчання. За допомогою бібліотеки Scikit-learn отримані статистичні показники якості моделі, що зображені на рисунку 3.15.

	precision	recall	f1-score	support
0	0.98	0.94	0.96	559
1	0.69	0.90	0.78	86
accuracy			0.93	645
macro avg	0.84	0.92	0.87	645
weighted avg	0.94	0.93	0.94	645

Рисунок 3.15 – Метрики якості моделі

Показник precision вказує на точність позитивних прогнозів, recall – відображає яка частина позитивних результатів була спрогнозована правильно, f1-score – визначає відсоток позитивних прогнозів який був вірним, і support відображає значення кількості входжень в кожному класі.

В результаті прогнозування, в числовому вигляді отримано значення в 534 за клас 0 проти 111 за клас 1, що відповідає перемозі представника республіканської партії з 82,7% голосів по округах.

### Висновки до розділу 3

1. Проведено попередню обробку(підготовку) вхідних даних .
2. Оптимізовано структуру вхідних даних.
3. Сформовано структуру нової мережі і проведено її реалізацію на Python .
4. Представленні результати експериментальних досліджень виконаної роботи

## ВИСНОВКИ

У даній роботі представлений метод прогнозування результатів виборів, при цьому розглянуті архітектури нейронних мереж. Представлений процес налаштування моделі мережі для вирішення задачі, а також набір тренувальних даних для навчання нейронної мережі. В результаті нейронна мережа показала, точність 95,7% при отриманій точності на контрольній вибірці 90,7%. Це є хорошим результатом з урахуванням відносно невеликих обчислювальних потужностей і невеликого часу, витраченого на навчання мережі. Отримана таким чином нейронна мережа може бути використана в виборчих перегонах і соціальних дослідженнях.

В результаті виконання роботи розроблено метод прогнозування результатів виборів на основі нейронних мереж з нечіткою логікою, при цьому:

1. Досліджено архітектури нейронних мереж, оцінено їх переваги і недоліки,
2. Вибрано середовище для реалізації нейронної мережі, здійснено обґрунтування вибору фреймворку Keras.
3. Реалізовано запропоновану структуру нейронної мережі та проведено її навчання.
4. Проведено експериментальні дослідження, отримано результати похибки та точності (акуратності) для навчальної вибірки, контрольної та тестової вибірки.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мелешкина Е. Ю. Исследование электорального поведения: теоретические модели и проблемы их применения // Политическая наука. – 2001. – № 2. – С. 190–211.
2. Позняк Д. Соціально-психологічні механізми політичної свідомості виборців // Соціальна психологія. – 2004. – № 1. – С. 42–56.
3. Мелешкина Е.Ю. “Воронка причинности” в электоральных исследованиях // Полис. – 2002. – № 5. – С. 46–53.
4. Вебер М. Избранные произведения / Общ. ред. Ю.Н. Давидова. – М.: Прогресс, 1990. – 808 с.
5. Пушкарева Г. В. Изучение электорального поведения: контуры когнитивной модели // Полис. – 2003. – № 3. – С. 120–129.
6. Sanders, D. Government Popularity and the Next General Election. *Political Quarterly. The Political Quarterly*. 1991. Vol. 62. PP. 235-261
7. Norpoth, H. Primary Colors: A Mixed Blessing for Al Gore. *Political Science & Politics*. 2001. Vol. 34. PP. 45-48.
8. Graefe A. German election forecasting: Comparing and combining methods for 2013. *German Politics*. 2015. Vol. 24. PP. 195-204.
9. Gorban A.N., Waxman Cory. Neural Networks For Political Forecast. *WCNN'95: Proceedings of the World Congress on Neural Networks*, Washington DC, USA, July 1995. PP. 176-178
10. Gill G. S. Election result forecasting using two layer perceptron network. *Journal of Theoretical and Applied Information Technology*. 2005. November 4. PP. 144-146.
- 11 Roman Borisyuk, Galina Borisyuk, Colin Rallings, Michael Thrasher Forecasting the 2005 General Election: A Neural Network Approach *The British Journal of Politics and International Relations*, vol. 7, 2: pp. 199-209.
12. Cybenko G. V. Approximation by Superpositions of a Sigmoidal function. *Mathematics of Control Signals and Systems*. 1989. Vol. 2. PP. 303-314

13. Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга. Москва: Мир, 1965. 478 с
14. Дивеев А.И., Софронова Е.А. — Основы генетического программирования Учебно-методическое пособие — М.: Изд-во РУДН, 2006.
15. Хайкин С. Нейронные сети: полный курс, 2-е изд., испр. : Пер. с англ. - М. : ООО "И.Д. Вильямс", 2006. - 1104 с.
16. Васенков Д.В. Методы обучения искусственных нейронных сетей // Компьютерные инструменты в образовании. - СПб.: Изд-во ЦПО "Информатизация образования", 2007, №1, С. 20-29.
17. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С.О.Субботін. – Житомир: Вид. О.О.Євенок, 2020. – 184 с
18. Van Veen, F. & Leijnen, S. (2019). The Neural Network Zoo. Retrieved from <https://www.asimovinstitute.org/neural-network-zoo>.
19. Kohonen, Teuvo. "Self-organized formation of topologically correct feature maps." *Biological cybernetics* 43.1 (1982): 59-69.
20. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, 1(4):541-551, Winter 1989.
21. Zeiler, Matthew D., et al. "Deconvolutional networks." *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.
22. Осовский С. Нейронные сети для обработки информации / С. Осовский ; пер. с польского И.Д. Рудинского. – М. : Финансы и статистика, 2002. – 344 с.
23. Dai J. Convolutional feature masking for joint object and stuff segmentation / J. Dai, K. He, J. Sun. // *CVPR*. – 2015.
24. Zeiler M. Visualizing and Understanding Convolutional Networks / M. Zeiler, R. Fergus. // *ICCV*. – 2015.
25. S. Sharma. Activation functions: neural networks // Режим доступа: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>



26. TensorFlow: A System for Large-Scale Machine Learning / Martín Abadi, Paul Barham // 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16) - November 2–4, 2016 - Savannah, GA, USA – P. 266 - 270

27. PyTorch [Электронный ресурс] : [Веб-сайт]. – Режим доступа: <https://pytorch.org/about/>

28. Caffe Deep learning framework by BAIR [Электронный ресурс] : [Веб-сайт]. – Режим доступа: <http://caffe.berkeleyvision.org/>

29. Why use Keras? [Online]. Available: <https://keras.io/>.

30. Обучение нейронной сети [Электронный ресурс] / – Режим доступа : <https://neuronus.com/theory/nn/238-obucheniya-nejronnoi-seti.html>.

31. Обучение нейронной сети [Электронный ресурс] / – Режим доступа: <http://www.aiportal.ru/articles/neural-networks/learning-neunet.html>.

32. A Step by Step Backpropagation Example [Электронный ресурс] / – Available: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example>

33. All About Using Jupyter Notebooks and Google Colab – Available: <https://www.datasciencecentral.com/profiles/blogs/all-about-using-jupyter-notebooks-and-google-colab>

34. Бесплатная GPU Tesla K80 для ваших экспериментов с нейросетями – Available: <https://habr.com/ru/post/348058/>