

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики

Курсова робота

освітній ступінь – магістр

**на тему: «Порівняння нейронної та баєсової мереж для класифікації
даних в реальному часі»**

Виконав: студент 1-го року
навчання

Спеціальності
122 Комп'ютерні науки

Студента Салій А. С.

Керівник Ющенко Ю. О.
старший викладач, к.т.н.

«_____» _____
20____ р.

Київ 2021

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Освітній ступінь бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-наукова програма магістр

ЗАТВЕРДЖУЮ

Викладач кафедри інформатики ,

старший викладач, к.т.н.

Ющенко Ю.О.

“ ____ ” _____ 20__ року

ЗАВДАННЯ

ДЛЯ КУРСОВОЇ РОБОТИ СТУДЕНТУ

Салій Анні Сергіївні

1. Тема роботи “Порівняння нейронної та баєсової мереж для класифікації даних в реальному часі”
2. Керівник роботи Ющенко Ю.О., старший викладач, к.т.н.
2. Строк подання студентом роботи 11.05.2021
3. План роботи

Зміст ГЧ до курсової роботи:

Індивідуальне завдання

Вступ

Огляд теоретичного матеріалу і здійснення дослідження

Висновки

Список літератури

Додатки (за необхідністю)

ГРАФІК ПІДГОТОВКИ КУРСОВОЇ РОБОТИ ДО ЗАХИСТУ

Календарний план виконання роботи:

№ п/п	Назва етапу	Термін виконання етапу
1.	Отримання завдання на курсову роботу.	20.11.2020
2.	Огляд технічної літератури за темою роботи.	05.12.2020
3.	Вивчення методів аналізу даних.	19.12.2020
4.	Написання умов до текстової частини роботи.	19.01.2021
5.	Написання умов до практичної частини проекту.	21.01.2021
6.	Написання першої частини курсової роботи.	22.02.2021
7.	Перегляд змісту роботи з керівником.	23.02.2021
8.	Написання другої частини курсової роботи.	24.03.2021
9.	Перегляд змісту роботи з керівником.	25.03.2021
10.	Написання висновків курсової роботи.	28.03.2021
11.	Перегляд змісту роботи з керівником.	30.03.2021
12.	Створення слайдів для доповіді та написання доповіді.	02.04.2021
13.	Аналіз отриманих результатів з керівником.	04.04.2021
14.	Корегування роботи .	09.04.2021
15.	Остаточне оформлення роботи та слайдів.	09.05.2021

Графік узгоджено «___» _____ 20___ р.

Науковий керівник Ющенко Ю. О.

Виконавець курсової роботи Салій А. С.

Анотація

У даній роботі розглядаються відмінності нейронних мереж та баєсових мереж, більш конкретно для завдань класифікації даних у реальному часі, та проводимо теоретичне та практичне порівняння між ними. Почнемо з короткого ознайомлення з баєсовими мережами. Надається також огляд нейронних мереж. Потім пропонуємо кілька ідей щодо того, який підхід є кращим у випадку класифікації даних у реальному часі. Внесок цієї роботи включає: дослідження літератури про відмінність нейронних мереж та баєсових; проведення експериментів, який підхід краще застосовувати, коли працюємо з даними у реальному часі.

Зміст

Анотація	4
ВСТУП	6
РОЗДІЛ 1. Теоретична частина	9
1.1. Історія виникнення баєсової мережі	9
1.2. Термін баєсової мережі	9
1.3. Приклад баєсової мережі	11
1.4. Семантика баєсової мережі	13
1.5. Наївні Баєси	16
РОЗДІЛ 2. Теоретична частина	18
2.1. Історія нейронних мереж	18
2.2. Термін нейронна мережа	19
2.3. Важливість нейронних мереж	22
2.4. Типи нейронних мереж	23
РОЗДІЛ 3. Практична частина	25
3.1. Порівняння	25
3.2. Експерименти	28
Висновки	32
Список використаної літератури	33

ВСТУП

Нейронні та баєсові мережі успішно використовуються в різних класифікаційних завданнях протягом останніх кількох десятиліть. Протягом останніх кількох років інтерес до нейронних мереж був надзвичайно збільшений, і вони почали використовуватися у переважній більшості галузей, включаючи зображення, мову, обробку сигналів. В даний час дослідники та спеціалісти намагаються застосовувати нейронні мережі майже в усіх сферах та системах, включаючи системи, що працюють з даними в реальному часі. Класифікація - одна з найпопулярніших проблем машинного навчання, і існує безліч алгоритмів, які намагаються її вирішити. Саме цим і обґрунтована **актуальність** обраної теми роботи. Дана робота зосереджена в основному на класифікації даних у режимі реального часу. Говорячи дані в режимі реального часу, маємо на увазі, що їх повна коректність залежить не тільки від логічної коректності, але й від часу, в який вони використовуються. Багато прикладів таких даних є у фінансовій сфері. Як приклад таких даних можуть бути ціни на акції, які швидко змінюються з часом. Іншим прикладом, який вважається популярним завданням класифікації, може бути система виявлення шахрайства з фінансовими (наприклад, кредитними картками) транзакціями. Іншою проблемою класифікації даних у режимі реального часу було б прийняття рішення про затвердження кредиту, тобто чи має особа мати право брати кредит у банку чи ні. У всіх вищезазначених випадках правильність даних та рішення про затвердження сильно залежать від часу. Таким чином, модель машинного навчання повинна бути адаптивною до змін. В іншому випадку навчена модель через деякий час може не мати відношення до нещодавно отриманих даних. У даній роботі також працюємо з даними в режимі реального часу.

Хоча, деякі підходи використання нейронних мереж були запропоновані для вирішення завдання класифікації, є деякі питання та проблеми, на які поки не дано відповіді у випадку використання нейронних мереж. У рамках цієї статті ми зосередимось на цих питаннях та **проблемах**, спробуємо порівняти два підходи та надати відповіді. Нейронні мережі стали більш популярними в промисловості, ніж баєсові. Однак існують певні занепокоєння та питання без відповіді щодо цього типу використання нейронних мереж. Особливо нейронні мережі дуже часто використовуються в завданнях класифікації, і спеціалісти часто не зважають на той факт, що баєсові мережі можуть бути кращим рішенням з кращою продуктивністю та точністю для конкретної проблеми. Крім того, перед вибором типу мережі потрібно врахувати деякі фактори, такі як прозорість алгоритму, теоретичне обґрунтування, відсутність значень у даних, обмеження підходу, що контролюється, побудова мережі та час навчання, адаптивність у режимі реального часу.

Мета: представити відмінності нейронних мереж та байєсівських мереж, більш конкретно для завдань класифікації даних у реальному часі, та провести теоретичне та практичне порівняння між ними. Запропонувати кілька ідей щодо того, який підхід є кращим у випадку класифікації даних у реальному часі.

Мотивація написання та **новизна** даної роботи виходить із проведеного дослідження, представленого в розділі першому.

У **першому розділі** знайомимося з баєвською мережею та кількома іншими поняттями, які використовуються в даній роботі, знайомимося з історією терміна баєсівської мережі, розглядаємо деякі приклади побудови баєвської мережі.

У другому розділі знайомимося з нейронною мережею, дізнаємось про важливість та типи нейронних мереж.

У третьому розділі після деякого теоретичного порівняння представляємо деякі експериментальні результати.

РОЗДІЛ 1. Теоретична частина

Для того, щоб дати пояснення щодо міркувань стосовно баєсової мережі та дати більше розуміння того, чому мережа дає певний результат, нам потрібна певна довідка щодо цих мереж. Розділ структурований наступним чином. У розділі 1.1 описана історія баєсової мережі. У розділі 1.2 розкрито термін баєсової мережі. Приклад баєсової мережі наведено в розділі 1.3, після чого детальніше розглядаємо семантику баєсової мережі у розділі 1.4. У розділі 1.5 ознайомлюємося з Наївним Баєсом.

1.1. Історія виникнення баєсової мережі

Термін Баєсова мережа був введений Йудою Перлом у 1985 році, щоб підкреслити:

- Часто суб'єктивної природи вхідної інформації.
- Покладання на баєсове обумовлювання як основу для уточнення інформації.
- Відмінність між причинними та доказовими способами міркування.

В кінці 1980-х років праці Йуди Перла «Імовірнісне міркування в інтелектуальних системах» та Річарда Неаполітана «Імовірнісне міркування в експертних системах» підсумували властивості баєсових мереж та утвердили баєсові мережі як область дослідження.

1.2. Термін баєсової мережі

Баєсові мережі - це широко використовуваний клас імовірнісних графічних моделей. Вони складаються з двох частин: структури та параметрів. Структура являє собою спрямований ациклічний графік (DAG), який виражає умовні незалежності та залежності між випадковими змінними, пов'язаними з вузлами. Параметри складаються з умовних розподілів ймовірностей, пов'язаних з кожним вузлом. Баєсова мережа - це компактне,

гнучке та інтерпретоване представлення спільного розподілу ймовірностей. Це також корисний інструмент у відкритті знань, оскільки спрямовані ациклічні графіки дозволяють представляти причинно-наслідкові зв'язки між змінними. Як правило, баєсова мережа навчається з даних.

Баєсова мережа - це імовірнісна графічна модель, яка приймає форму спрямованого ациклічного графіка (DAG) у поєднанні з розподілом ймовірностей. Формально баєсовою мережею є кортеж $B = (G, P)$, де G - ациклічний графік, а P - розподіл ймовірностей. Графік G сам по собі являє собою кортеж $(V(G), A(G))$, а $V(G)$ являє собою набір вузлів $\{V_1, \dots, V_n\}$ та $A(G)$ набір ребер $A(G) \subseteq V(G) \times V(G)$.

Вузли $V(G)$ у баєсовій мережі представляють статистичні змінні разом із умовним розподілом ймовірності за цими вузлами, враховуючи його батьків $P(V_i | \pi(V_i))$. Для вузлів $V(G)$ без батьківських даних умовний розподіл ймовірності дорівнює попередньому розподілу ймовірностей $P(V_i / \emptyset) = P(V_i)$. Баєсова мережа - це представлення спільного розподілу ймовірностей за набором статистичних змінних.

$$P(V_1, \dots, V_n) = \prod_i^n P(V_i | \pi(V_i)).$$

Ми, як правило, зацікавлені в апостеріорного ймовірностей, тобто ймовірність стану після введення показань в мережі. Оцінку ймовірності змінних можна отримати кількома способами, наприклад, зі статистичних даних, літератури про домен або від експертів. Алгоритми, пов'язані з такою мережею, здатні обчислити будь-який $P(H / E)$ для вузла гіпотези H , що свідчить про E . Така мережа здатна кодувати цілі розподіли ймовірностей за всіма можливими комбінаціями змінних. Імовірність $P(H / E)$ по суті обчислюється теоремою Баєса:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}.$$

Для обчислення розподілу ймовірностей були розроблені алгоритми, які роблять висновок щодо таких мереж.

Імовірнісний висновок означає розповсюдження наявних доказів та оновлення розподілу ймовірностей вузлів у мережі. Імовірнісний висновок у баєсових мережах загалом NP-важкий, але існуючі алгоритми можуть обчислювати ймовірності в поліноміальному часі в мережах обмеженої топології.

Для того, щоб зробити наведені вище визначення більш чіткими, наступна частина охоплює невеликий приклад баєсової мережі.

1.3. Приклад баєсової мережі

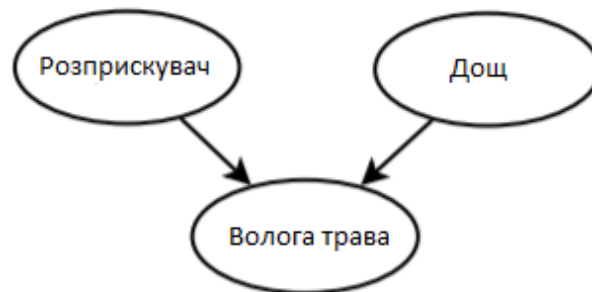


Рисунок 1. Граф для прикладу баєсової мережі

На рисунку 1 показаний графік невеликої баєсової мережі, що складається з трьох вузлів і двох країв. У цьому прикладі нас цікавить ймовірність того, що завтра трава буде мокрою. Це може бути результатом розприскувача або результатом дощу, або може бути результатом обох. Спочатку ми припускаємо, що стан кожного вузла може бути лише *істинним* або *хибним*,

тому або йшов дощ, або ні. У більш складних мережах можна визначити більше двох станів для кожного вузла.

Для кожного вузла можна призначити ймовірності цим значенням. Наприклад, звіт про погоду стверджував, що завтра існує лише десять відсотків ймовірності дощу, тож ймовірність буде $P(\text{дощ} = \text{істина}) = 0,1$ і $P(\text{дощ} = \text{незначний}) = 0,9$. Оскільки хочемо, щоб трава була мокрою, запрограмували розприскувач таким чином, що він повинен змочувати траву завтра, але завжди можливо, що він несправний, і тому ми все ще маємо шанс $P(\text{розприскувач} = \text{false}) = 0,05$.

Присвоєння ймовірностей значенням вузла працює дещо по-іншому, оскільки на результат впливають можливі значення його двох батьківських вузлів. Для кожної комбінації значень трьох змінних нам потрібно визначити ймовірність. Подивившись рисунок 2, ми можемо побачити

значення, присвоєні цим комбінаціям.

Розприскувач	Дощ	Волога Трава					
$P(\text{Істина}) = 0.95$	$P(\text{Істина}) = 0,1$	Розприскувач	Істина	Істина	Хиба	Хиба	
$P(\text{Хиба}) = 0.5$	$P(\text{Хиба}) = 0,9$	Дощ	Істина	Хиба	Істина	Хиба	
		$P(\text{Істина})$	=	1	1	1	0
		$P(\text{Хиба})$	=	0	0	0	1

Рисунок 2. Таблиці ймовірностей

Якщо пішов дощ або розприскувач працював, ймовірність того, що трава мокра, рівно 1, а в іншому випадку - 0. Для зручності прикладу ми припускаємо, що інших можливостей немає. Тому трава не буде мокрою, коли розприскувач не працюватиме або не йтиме дощ.

Якщо ввести цей приклад в інструмент баєсової мережі, такий як GeNIe, можемо обчислити, наприклад, ймовірність намокання трави за цими таблицями ймовірностей. У цьому випадку $P(\text{Волога трава} = \text{true})$ дорівнює 0,955, і це, мабуть, саме те, що очікувалось. Незважаючи на те, що є невелика ймовірність дощу, трава, ймовірно, все ще буде мокрою через розприскувач, і це саме те, що кодує мережа Баєса.

У баєсових мережах ми маємо можливість вводити спостереження за певними змінними. Це робиться, коли ми повністю знаємо значення вузла в мережі.

1.4. Семантика баєсової мережі

Баєсова мережа містить ациклічний диграф, що означає, що краї мають напрямки, і на графіку неможливо мати цикли. Коли між двома вузлами є ребро, це показує, що вони можуть впливати на ймовірність один одного. Якщо є дані про наявність у мережі вузлів, це може мати прямий вплив на розподіл ймовірностей інших вузлів. Коли ми розглядаємо цілий ланцюжок вузлів, наявність доказів може зробити так, щоб цей вплив було запобігано. Це називається блокуванням, що означає, що вплив вузла доказів уздовж однієї сторони ланцюга не змінює жодного розподілу ймовірностей для вузлів з іншого боку ланцюга. Щоб дати більше розуміння, ми розглянемо визначення блокування.

Визначення 1

Нехай $G = (V(G), A(G))$ - ациклічний диграф і нехай s - ланцюг у G між $V_i \in V(G)$ та $V_j \in V(G)$.

Ланцюг s блокується (можливо порожнім) набором вузлів $W \subseteq V(G)$, якщо $V_i \in W$ або $V_j \in W$, або s містить три послідовні вузли X_1, X_2, X_3 , для яких (принаймні) виконується одна умова:

- ребра (X_2, X_1) та (X_2, X_3) знаходяться на ланцюжку s , а $X_2 \in W$;
- ребра (X_1, X_2) і (X_2, X_3) знаходяться на ланцюжку s , а $X_2 \in W$;
- ребра (X_1, X_2) та (X_3, X_2) знаходяться на ланцюжку s , а $\sigma^*(X_2) \cap W = \emptyset$.

Тут $\sigma^*(X_2)$ - це набір усіх нащадків X_2 , включаючи сам X_2 , а ребро (X_i, X_j) являє собою спрямоване ребро від вузла X_i до вузла X_j .

На рисунку 3 можемо побачити графічне зображення умов, згаданих у визначенні 1.

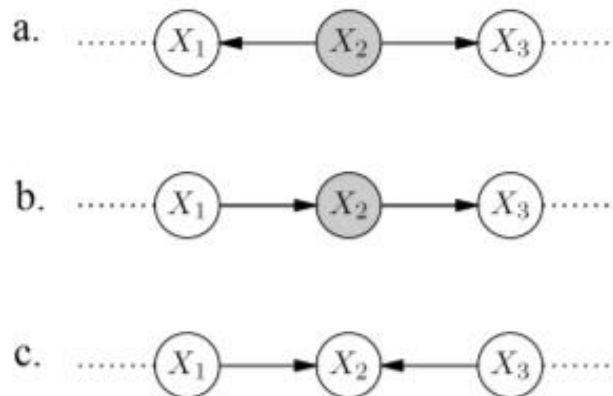


Рисунок 3. Розрізнення випадків блокування ланцюга

У перших двох ланцюгах ми бачимо, що $X_2 \in W$, що означає, що цей вузол є частиною блокуючого набору W . Зверніть увагу, що випадок b симетричний, і ми можемо повернути всі ребра, щоб опинитися в одному загальному випадку. У третьому випадку ми бачимо, що X_2 та всі його нащадки (у даному випадку порожній набір \emptyset) відсутні у наборі блокування W . Концепція блокування узагальнює цілі набори вузлів, якщо вони задовольняють запропонованим умовам.

З концепцією блокування одинарних ланцюгів ми визначаємо поняття d -поділу, де ми зацікавлені в блокуванні всіх ланцюгів між двома наборами вузлів. Отже, нас цікавить набір вузлів, які блокують усі ланцюги між цими двома наборами вузлів, і тому можна сказати, що вони не впливають на розподіл ймовірностей один одного. Визначимо d -поділ наступним чином:

Визначення 2

Нехай $G = (V(G), A(G))$ - ациклічний диграф. Нехай $X, Y, Z \subseteq V(G)$ - набори вузлів у G . Набір вузлів Z , як кажуть, d -відокремлює набори вузлів X і Y у G , позначений як $\langle X / Z / Y \rangle_G^d$, якщо для кожного вузла $V_i \in X$ і кожного вузла $V_j \in Y$ кожен ланцюжок від V_i до V_j у G заблокований Z .

За допомогою критерію d -поділу можна визначити, чи можуть вузли впливати один на одного на апостеріорні ймовірності, враховуючи дані, введені в баєсову мережу. Коли ми встановлюємо, що два вузли розділені d , це означає, що вони умовно незалежні з огляду на вузли в наборі блокування. Баєсові мережі використовують свій графік, щоб представити відношення незалежності між змінними, асоційованими з вузлами, а розділення d служить для зчитування заяв про незалежність з графіка. У багатьох випадках краї представляють причинно-наслідковий зміст, але в принципі це не так.

Баєсові технології стали популярними та усталеними, що демонструють численні компанії. Деякі сфери, в яких успішно застосовуються класифікатори баєсової мережі, такі: обчислювальна техніка та робототехніка, медицина та охорона здоров'я, економіка, фінанси та банківська справа, екологічні науки.

Структуру баєсових мереж можуть визначити або експерти з проблемних галузей, або з навчальної вибірки даних. У випадку великих наборів даних

з багатьма параметрами навчання структури може зайняти багато часу. З цієї причини існує кілька баєсових мереж із заздалегідь визначеною структурою, які пришвидшують або усувають процес навчання структури.

1.5. Наївні Баєси

Наївні Баєси - одна з баєсових мереж із заздалегідь визначеною структурою. У наївному класифікаторі Баєса кожна змінна ознаки має змінну класу єдиним батьківським елементом. Це означає, що структура є фіксованою, і єдиним завданням, що бере участь у навчанні, є оцінка параметрів. Тут передбачається, що всі ознаки є умовно незалежними з огляду на значення класу або більш формально, $p(\prod_{i=1}^n x_i | y) = \prod_{i=1}^n p(x_i | y)$, де x_i - це i -та ознака, y - змінна класу, а n - кількість ознак.

Приклад наївного Баєса представлений на рисунку 6 (а). Існують також деякі розширення наївних Баєсів, і найпопулярнішим є доповнені дерева наївного Баєса (tree augmented naive Bayes - TAN). Тут кожна змінна ознаки може мати максимум одну змінну ознаки як батьківську (крім змінної класу).

Однак єдине, що потрібно зробити, це знайти оптимальні єдині зв'язки між функціями. Приклад доповненого дерева наївного Баєса представлений на рисунку 6 (b).

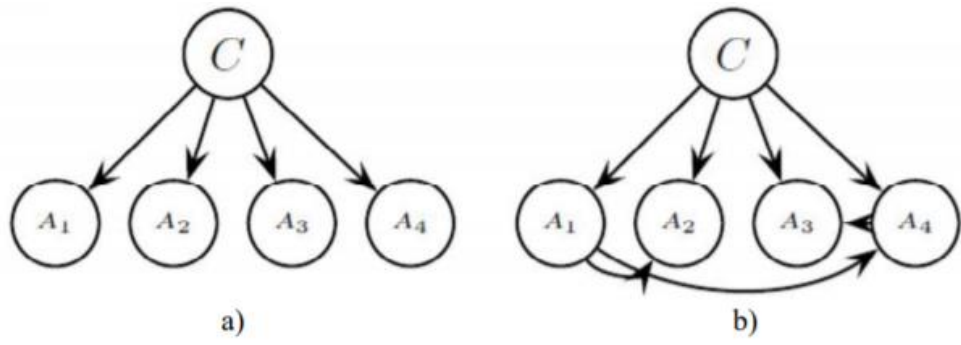


Рисунок 4. Приклади наївного Баєса (а) та доповнених дерев наївного Баєса (б).

РОЗДІЛ 2. Теоретична частина

В даній роботі використовуються нейронні мережі. Цей розділ опише прості нейронні мережі. У розділі 2.1 описана історія нейронної мережі. Нейронні мережі є методом керування даними, і в останні роки стали популярними, частково завдяки збільшенню доступних даних, а також оптимізації апаратного забезпечення комп'ютера. Ці мережі використовувались для вирішення безлічі таких проблем, як класифікація зображень, генерація тексту та виявлення об'єктів. Основним будівельним блоком глибинної мережі є єдиний перцептрон, що пояснюється в розділі 2.2. Про важливість дізнаємось з розділу 2.3, а з типами нейронних мереж ознайомимось в розділі 2.4.

2.1. Історія нейронних мереж

Історія нейронних мереж довша, ніж думає більшість людей. Хоча ідею "машини, яка думає" можна простежити ще у стародавніх греків, ми зупинимось на ключових подіях, що призвели до еволюції мислення навколо нейронних мереж, яке протягом багатьох років зменшувалось і стало популярним.

Перша нейронна мережа була задумана Уорреном Мак-Каллоком та Уолтером Піттсом у 1943 році. Вони написали основний документ про те, як можуть працювати нейрони, та змодельювали свої ідеї, створивши просту нейронну мережу за допомогою електричних ланцюгів.

Ця проривна модель відкрила шлях для досліджень нейронних мереж у двох напрямках:

- Біологічні процеси в мозку.
- Застосування нейронних мереж до штучного інтелекту (ШІ).

Дослідження ШІ швидко прискорилися, і Кунігіко Фукусіма розробив першу справжню багатoshарову нейронну мережу в 1975 році.

Початковою метою підходу до нейронної мережі було створення обчислювальної системи, яка могла б вирішувати такі проблеми, як людський мозок. Однак з часом дослідники зосередили свою увагу на використанні нейронних мереж для відповідності конкретним завданням, що призвело до відхилень від суворо біологічного підходу. Відтоді нейронні мережі підтримують різноманітні завдання, включаючи комп'ютерний зір, розпізнавання мови, машинний переклад, фільтрацію соціальних мереж, гру в настільні та відеоігри та медичну діагностику.

Коли розміри структурованих та неструктурованих даних збільшувались до рівнів великих даних, люди розробляли системи глибокого навчання, які, по суті, є нейронними мережами з багатьма рівнями. Глибоке навчання дозволяє збирати та видобувати все більші та більші дані, включаючи неструктуровані дані.

2.2. Термін нейронна мережа

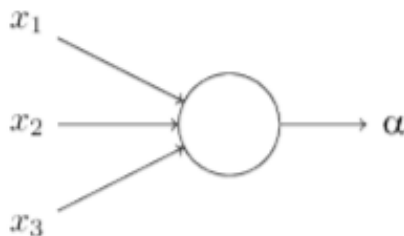


Рисунок 5. Один перцептрон з трьома входами.

Нейронна мережа складається з декількох штучних нейронів (вони ж перцептронів), з'єднаних як графічні вузли. Кожен з цих вузлів матиме вихідну активацію a відповідно до функції:

$$a = f\left(\sum_{i=1}^N \mathbf{W}_i x_i + \mathbf{b}\right)$$

де N - кількість входів, W - вектор ваг і b зсувів, які вивчаються параметрами моделі. Потім це обертається функцією активації f для збільшення складності моделі, оскільки вона додає нелінійність. Поширені приклади функції активації включають сигмовидну або випрямлену лінійну одиницю. Однак окремих нейрон не особливо корисний, оскільки він занадто спрощений, щоб навчитися функціональності вищого рівня.

Натомість ці нейрони часто об'єднуються у шари, і ці шари складаються один до одного, утворюючи багатошаровий перцептрон. Багатошаровий перцептрон - це основна форма глибокої нейронної мережі (DNN), де кожен шар повністю щільний - тобто всі вузли з одного шару з'єднані з кожним вузлом попереднього шару. Існує три типи рівнів: вхідний рівень, прихований і вихідний рівні. Вхідні та вихідні рівні є пояснювальними, а приховані рівні - це всі рівні посередині, які не потрапляють безпосередньо на вхідні чи вихідні дані. Простий багатошаровий перцептрон видно на рисунку 2. DNN навчені для використання як великі нелінійні функції, і зазвичай вони навчені вирішувати або проблеми регресії, або класифікації. Проблеми регресії полягають у тому, що модель намагається отримати один вихід як число з плаваючою комою, прикладом якого є оцінка ціни певної частки з урахуванням історичних даних. Проблеми класифікації полягають у тому, що модель присвоює єдину мітку вхідним даним із заданого набору міток, прикладом якої є сортування кеглі за їх кольором.

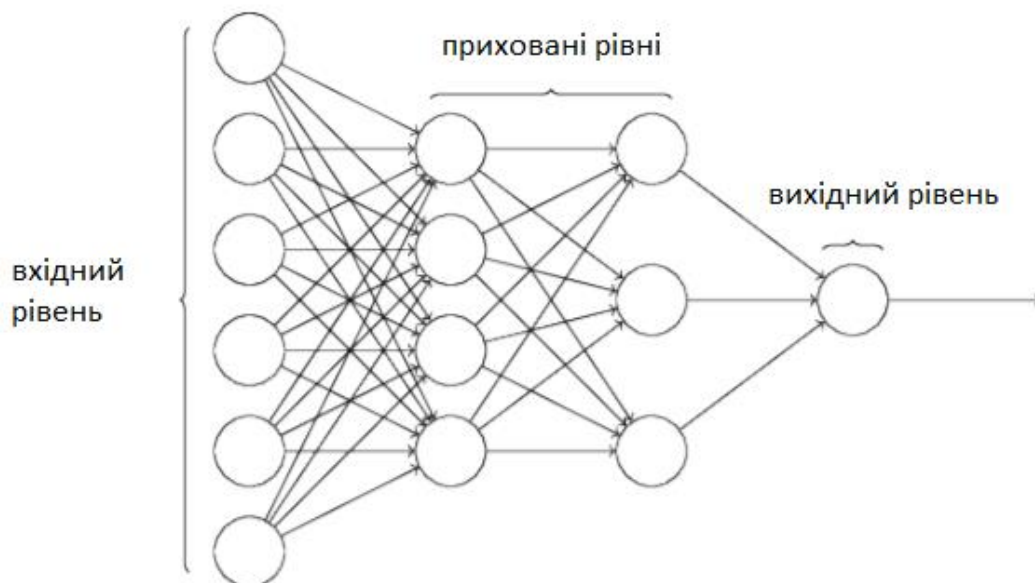


Рисунок 6. Базовий багатошаровий перцептрон. Тут багато нейронів об'єднуються, утворюючи шари, і багато шарів об'єднуються, утворюючи остаточну мережу.

Ці мережі навчаються за допомогою оптимізаційного процесу, відомого як зворотне розповсюдження.

DNN спочатку ініціалізується випадковими вагами, і висновок виконується в мережі шляхом введення навчальної вибірки. Потім дані будуть поширюватися вперед по вузлах, поки не буде вироблено остаточне рішення. Це рішення буде порівняно з основними даними, які є правильним позначенням для цих вхідних даних. Це спричиняє помилку мережі, яку генерує якась функція втрат - як правило, перехресні втрати ентропії для мережі категоризації. Диференціюючи кожен шар мережі, можна знайти градієнт, що дозволяє оптимізатору оновлювати ваги мережі для цього навчального прикладу. Це повторюється для кожного зразка в навчальному наборі. Як і у всіх оптимізаторів, це робиться з дуже малим розміром кроку. Якщо розмір кроку дуже великий, ваги можуть надуватися, і модель не сходиться. Для прискорення навчального процесу в якості навчальних даних можна використовувати одночасно декілька зразків у процесі, який називається пакетним, де підсумовуються ваги оновлення.

2.3. Важливість нейронних мереж

Нейронні мережі також ідеально підходять, щоб допомагати людям вирішувати складні проблеми в реальних ситуаціях. Вони можуть вивчати та моделювати взаємозв'язки між вхідними та вихідними даними, які є нелінійними та складними; робити узагальнення та умовиводи; розкрити приховані стосунки, закономірності та передбачення; та моделювати дуже мінливі дані (такі як дані фінансових часових рядів) та відхилення, необхідні для прогнозування рідкісних подій (наприклад, виявлення шахрайства). Як результат, нейронні мережі можуть покращити процеси прийняття рішень у таких сферах, як:

- Виявлення шахрайства з кредитними картками
- Оптимізація логістики транспортних мереж
- Розпізнавання символів та голосу, також відоме як обробка природної мови
- Медична та діагностика хвороб
- Цільовий маркетинг
- Фінансові прогнози щодо цін на акції, валюти, опціонів, ф'ючерсів, банкрутства та рейтингів облігацій
- Робототехнічні системи управління
- Прогнозування електричного навантаження та споживання енергії
- Контроль процесу та якості
- Ідентифікація хімічної сполуки
- Оцінка екосистеми
- Комп'ютерний зір для інтерпретації необроблених фотографій та відео (наприклад, у медичній візуалізації та робототехніці та розпізнаванні обличчя)

2.4. Типи нейронних мереж

Існують різні види глибоких нейронних мереж - і кожна має свої переваги та недоліки, залежно від використання. Приклади включають:

Згорткові нейронні мережі (CNN) містять п'ять типів шарів: вхідний, згортковий, об'єднаний, повністю зв'язаний та вихідний. Кожен шар має певне призначення, наприклад підведення підсумків, підключення або активація. Згорткові нейронні мережі популяризували класифікацію зображень та виявлення об'єктів. Однак CNN застосовуються і в інших сферах, таких як обробка природних мов та прогнозування.

Нейронні мережі, що повторюються (RNN), використовують послідовну інформацію, таку як дані з позначкою часу із сенсорного пристрою або вимовлене речення, що складається з послідовності термінів. На відміну від традиційних нейронних мереж, всі входи до періодичної нейронної мережі не незалежні один від одного, і вихід кожного елемента залежить від обчислень попередніх елементів. RNN використовуються в програмах прогнозування та часових рядів, аналізі настроїв та інших текстових додатках.

Нейронні мережі прямого зв'язку, в яких кожен персептрон в одному шарі з'єднаний з кожним персептроном з наступного шару. Інформація подається вперед від одного шару до наступного лише в прямому напрямку. Циклів зворотного зв'язку немає.

Нейромережі автоенкодера використовуються для створення абстракцій, званих кодерами, створеними з заданого набору входів. Хоча подібні до більш традиційних нейронних мереж, автокодери прагнуть моделювати вхідні дані самі, і тому метод вважається неконтрольованим. Передумовою автокодерів є десенсибілізація нерелевантного та сенсибілізація відповідного. У міру додавання шарів формуються подальші абстракції на

вищих шарах (шари, найближчі до точки, в яку вводиться шар декодера). Потім ці абстракції можуть бути використані лінійними або нелінійними класифікаторами.

РОЗДІЛ 3. Практична частина

Хоча як нейронні, так і баєсові мережі вирішують подібні класифікаційні проблеми і можуть бути представлені у вигляді графів, між ними існують суттєві відмінності, які представлені будуть нижче. Після деякого теоретичного порівняння в цьому розділі представлені деякі експериментальні результати.

3.1. Порівняння

Нейронні мережі - це дискримінаційні алгоритми, баєсові мережі - породжувальні. У випадку завдань класифікації дискримінаційні алгоритми намагаються знайти межі прийняття рішень між класами (тобто функцією $h(x)$) і навчитися $p(y/x)$ безпосередньо, де y представляє клас, а x - ознаки. Новий тренувальний приклад класифікується в один з класів відповідно до

$$y \in \begin{cases} \text{class 1, } h(x) \geq T, \\ \text{class 2, } h(x) < T \end{cases}$$

формули, де T - порогове значення. На відміну, породжувальні алгоритми намагаються будувати моделі для кожного класу на основі їх особливостей, і вони навчаються $p(y/x)$ і $p(y)$. У цьому випадку є ще один додатковий крок для обчислень $p(y/x)$, який виконується за теоремою Баєса:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Тут $p(y)$ попередня ймовірність, яка може вплинути на результат, і є кілька підходів, як її вибрати.

Нейромережам не вистачає теоретичного обґрунтування, вузли (нейрони) і ребра не мають жодного значення окремо. Мережа отримує вхідні дані, виконує обчислення і видає вихідні дані. Дуже часто нейронні мережі

називають підходом чорної скриньки. Дотепер не можна пояснити, чому мережа дала такий результат. Хоча нейронні мережі успішно застосовуються до величезної кількості проблем, ці теоретичні прогалини часто перешкоджають використанню нейронних мереж у деяких галузях.

Причина в тому, що спеціалісти в галузях майже завжди хочуть знати не тільки точність алгоритмів, але й пояснення щодо прогнозування. У випадку баєсових мереж усі вузли та ребра мають цілком конкретне значення. Вузли відповідають особливостям набору даних та взаємозв'язкам між ними. Цей підхід дає можливість зрозуміти спосіб роботи мереж та пояснити, чому мережа передбачила такий результат. У разі класифікації даних у режимі реального часу (наприклад, виявлення шахрайства з фінансовими операціями) баєсові мережі мають перевагу над нейронними мережами.

Відсутність теорії та доказів у випадку нейронних мереж включає не тільки тренування та прогнозування нейронних мереж, але також попередню обробку даних, наприклад, роботу з відсутніми значеннями. Існує кілька підходів до заповнення відсутніх значень у наборі даних, однак вони також є емпіричними і можуть, а можуть не працювати на практиці. У випадку баєсових мереж існують теоретично доведені алгоритми (такі, як максимізація очікувань), які призначають відсутні значення, і це гарантує найбільш оптимальний результат. Ці алгоритми також добре працюють на практиці.

Ще однією відмінністю є обмеження нейронних мереж бути контрольованим підходом [11]. Формально контрольоване навчання можна визначити наступним чином: нехай $X = (x_1, \dots, x_n)$ - це набір з n прикладів, де $x_i \in X$ для всіх $i \in [n]: = \{1, \dots, n\}$. Мета полягає в тому, щоб навчитися відображенню від x до y , даючи тренувальний набір, складений з пар (x_i, y_i) .

Однак у системах, що працюють з даними в режимі реального часу, ніхто не може гарантувати, що всі дані будуть позначені та u_i будуть доступні для всіх прикладів. Таким чином, було б непогано мати можливість вчитися як на позначених, так і на непозначених даних. У випадку з нейронними мережами всі непозначені дані слід видалити із набору даних, оскільки неможливо включити їх у тренувальний процес. Однак у випадку баєсових мереж змінну класу можна відстежувати як звичайну особливість і використовувати алгоритми введення відсутніх даних (наприклад, максимізацію очікувань). Це врешті-решт дозволяє вчитися як на позначених, так і не на позначених даних. Таким чином, для цієї властивості баєсові мережі є кращими у випадку класифікації даних у реальному часі, ніж нейронні мережі.

Структури нейронних мереж визначаються дослідниками / розробниками шляхом експериментів та налаштування. Не існує теоретично доведеного правила, яке дозволяє побудувати найефективнішу нейронну мережу для певної проблеми. Після проведення експериментів та вибору структури нейронної мережі, мережу слід тренувати для навчання ваг. Це може зайняти досить тривалий час, залежно від розміру навчального набору, кількості функцій та апаратних можливостей.

У випадку баєсових мереж структуру можуть визначити експерти або дізнатися з даних. Існує кілька алгоритмів, що виконують навчання за баєсовою структурою мережі. Після побудови баєсової мережі немає необхідності в навчанні параметрів (ваг), оскільки вони автоматично засвоюються. Однак у випадку передбачень нейронна мережа буде працювати швидше, ніж баєсова. Крім того, ваги та вихід в нейронних мережах є конкретними дійсними числами, але в баєсових мережах вони є розподілами ймовірностей, що представляють ідею невизначеності.

Останній пункт стосується адаптивності. У випадку з даними реального часу це дуже важливий момент. Після навчання нейронної мережі оновити ваги буде неможливо без перенавчання, що забирає багато часу. Незважаючи на те, що в літературі є деякі запропоновані методи, які намагаються обійти шлях, вони не повністю виявились дієвими. У випадку баєсових мереж ваги можна оновлювати на льоту без будь-яких додаткових зусиль. Це дає моделі можливість адаптуватися до постійно отримуваних даних у режимі реального часу.

3.2. Експерименти

Впроваджуємо наївного Баєса та доповнене дерево наївного Баєса. Після проведення експериментів порівнюємо результати з результатами нейронних мереж. Реалізуємо баєсові мережі в R за допомогою пакета `bnlearn`. Як датасет беремо “Дані про кредит Німеччини”. Для порівняння експериментів з уже відомими результатами збираємося виміряти ефективність моделі за оцінкою F_1 :

$$F_1 = 2 * \frac{\text{влучність} * \text{повнота}}{\text{влучність} + \text{повнота}}$$

Тридцять експериментів було проведено для класифікатора з перетасовкою даних. Результати наведені в таблиці:

Експерименти	Оцінка F_1	
	№	Наївний Баєс
1	0.79	0.75

2	0.818	0.772
3	0.799	0.757
4	0.83	0.782
5	0.819	0.773
6	0.813	0.769
7	0.807	0.764
8	0.83	0.782
9	0.83	0.791
10	0.827	0.779
11	0.798	0.756
12	0.83	0.784
13	0.825	0.778
14	0.815	0.791
15	0.805	0.762
16	0.79	0.75
17	0.794	0.753
19	0.797	0.755
20	0.798	0.756
21	0.83	0.789
22	0.812	0.767

23	0.795	0.754
24	0.79	0.75
25	0.79	0.791
26	0.808	0.765
27	0.816	0.771
28	0.818	0.772
29	0.82	0.774
30	0.82	0.774

Експерименти показують, що найкращий показник F_1 для наївного Байєса становить до 0,83, а для доповнених дерев наївного Баєса - до 0,79. Порівняння найкращих і найгірших результатів з уже відомими результатами нейронних мереж представлено в таблиці нижче:

	Наївний Баєс	Доповнені дерева наївного Баєса	Нейронні мережі
Найкращий результат	0.83	0.79	0.92
Найгірший результат	0.79	0.75	0.88

Як можемо бачити з результатів, незважаючи на переваги перед нейронними мережами, на практиці баєсові мережі не працювали так добре, як нейронні мережі для цієї конкретної проблеми. В даний час це може бути компромісом і питанням вибору, який підхід використовувати в завданні класифікації даних у реальному часі. Наша майбутня робота буде присвячена підвищенню точності баєсових мереж шляхом їх поєднання з дискримінаційними алгоритмами, такими як нейронні мережі.

Висновки

Як перший внесок, надали ретельний огляд та опис літератури баєсової мережі та нейронної (розділи другий та перший). Другий внесок - представили порівняння та показали відмінності нейронних та баєсових мереж для завдання класифікації даних у реальному часі. Баєсові мережі мають міцну теоретичну базу і вважаються надійними. Хоча нейронним мережам не вистачає теоретичних доказів і вони є більш емпіричними (підхід чорної скриньки). Однак наші експерименти показали, що нейронні мережі дають кращі результати для класифікації даних у режимі реального часу. Таким чином, це питання вибору, який підхід вибрати для завдання - баєсові мережі з помірною точністю, але теоретично обґрунтованими, або нейронні мережі з кращою точністю, але розглядаються як чорна скринька. Наші майбутні дослідження будуть зосереджені на пошуку способів поєднання та побудови гібридної моделі з дискримінаційних та породжувальних алгоритмів, що підвищить точність та покращить продуктивність.

Список використаної літератури

1. Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.
2. Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.
3. Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497.
4. Ф. В. Дженсен. "Графи баєсівських мереж та рішень". Спрингер. 2001.
5. Gordon, N. J. (1994). *Bayesian Methods for Tracking*. PhD thesis, Imperial College, University of London.
6. Girosi, F., Jones, M., and Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269.
7. Neubig, G. (2017). *Neural machine translation and sequence-to-sequence models: A tutorial*.
8. Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491.
9. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
10. Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian Additive Regression Trees. *Annals of Applied Statistics*, 4(1):266–298.
11. R. M. Neal, "Bayesian learning for neural networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 1995.
12. S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*. Cambridge, U.K.: Cambridge Univ. Press, Jul. 2015.
13. Scutari M., *Learning Bayesian Networks with the bnlearn R // Journal of Statistical Software*. - 2010. - Vol. 35. - P. 1-22.

14. B Zheng, Y H Chang, X H Wang, W F Good and D Gur, "Application of a Bayesian belief network in a computer-assisted diagnosis scheme for mass detection", Proc SPIE on Med Imaging, pp. 3661-3167, 1999.
15. S. Kostadinov, How Recurrent Neural Networks Work, 2017, [online] Available: towardsdatascience.com.
16. uk.wikipedia.org/wiki/Бассова_мережа – 09.11.2021.
17. Kim, Y. (2014). "Convolutional Neural Networks for Sentence Classification". In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751.
18. Samarasinghe, S. (2007). Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition. Taylor & Francis.
19. Orloff, J., & Bloom, J. (2014). Comparison of frequentist and Bayesian inference.
20. Stephenson, T. (2000). An introduction to Bayesian network theory and usage. Idiap Research Report, 31.