

УДК 004

М.Окунєв, магістр гр. КІ-22М-2,

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РАДІАЛЬНО-БАЗИСНОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАННЯ ОБРАЗІВ

У статті розроблено програмне забезпечення, яке призначено для системи радіально-базисної нейронної мережі для розпізнання образів. Метою розробки є дослідження та програмна реалізація системи радіально-базисної нейронної мережі для розпізнання образів. Об'єктом дослідження є процес радіально-базисної нейронної мережі для розпізнання образів. Предметом дослідження є методи радіально-базисної нейронної мережі для розпізнання образів. Методи дослідження базуються на методах штучного інтелекту, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи радіально-базисної нейронної мережі для розпізнання образів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. Класифікація зображень є важливою фазою в загальному процесі автоматичного анотування зображень і пошуку зображень. У цьому дослідженні ми стурбовані дизайном класифікаторів зображень, розроблених у просторі ознак, утвореному примітивами низького рівня, визначеними в налаштуваннях стандарту MPEG-7. Наша мета полягає в тому, щоб дослідити дискримінаційні властивості таких стандартних дескрипторів зображень і розглянути ефективну архітектуру класифікаторів разом із їхнім розробкою. Здатності класифікатора зображень до узагальнення мають важливе значення для його успішного використання під час пошуку та анотації зображень. Інтуїтивно очікується, що класифікатор повинен досягти високої точності класифікації невидимих зображень, які є досить «схожими» на ті, що зустрічаються в навчальному наборі. З іншого боку, ми можемо припустити, що продуктивність класифікатора не може бути гарантована у випадку зображень, які дуже сильно відрізняються від елементів навчального набору. Щоб слідувати цьому спостереженню, ми розробляємо та використовуємо концепцію локалізованої помилки узагальнення та показуємо, як вона керує дизайном класифікатора. Як класифікатор зображень ми розглядаємо використання нейронних мереж радіальної базисної функції (RBFNN). Завдяки інтенсивним експериментам ми показуємо, що отриманий класифікатор перевершує інші класифікатори, такі як багатокласові опорні векторні машини (SVM), а також «стандартні» RBFNN (тобто ті, що розроблені без вказівок, запропонованих оптимізацією локалізованої помилки узагальнення). Експериментальні дослідження виявили деякі цікаві інтерпретаційні можливості класифікаторів RBFNN, пов'язані з їх рецептивними полями.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-20] було виявлено певні прогалини у забезпеченні системи радіально-базисної нейронної мережі для розпізнання образів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи радіально-базисної нейронної мережі для розпізнання образів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем радіально-базисної нейронної мережі для розпізнання образів.

– Дослідження системи радіально-базисної нейронної мережі для розпізнання образів.

– Програмна реалізація системи радіально-базисної нейронної мережі для розпізнання образів.

Об'єктом дослідження є процес радіально-базисної нейронної мережі для розпізнання образів.

Предметом дослідження є методи радіально-базисної нейронної мережі для розпізнання образів.

Методи дослідження базуються на методах штучного інтелекту, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Починаючи з самого початку, давайте поговоримо про зображення та те, як на них дивляться комп'ютери. Думаю, більшість людей знають про пікселі. Пікселі – це одна точка на зображенні, яка визначається кольором і його положенням.

Якщо ми подивимося на приклад ліворуч, ми побачимо дуже просте зображення, що складається з (ігноруйте масштабування) чотирьох пікселів: двох синіх і двох зелених, кожен колір поширюється по діагоналі.

Якщо ми хочемо описати ці пікселі трохи технічніше, ми можемо призначити номери різним кольорам. Скажімо, *синій*=1 і *зелений*=-1.

Таким чином, ми можемо записати це зображення як послідовність із чотирьох чисел, що йде зліва направо та зверху вниз. 1, -1, -1, 1

Якщо вам це важко зрозуміти, не хвилюйтеся. Повертаючись до нашого піксельного зображення, ми можемо розмістити числа прямо на різних пікселях, щоб зробити його більш чітким. Я сподіваюся, що тепер це має більше сенсу.

Ідея тут проста: ми хочемо математично описати те, що бачимо. Насправді цифри відображатимуть або фактичний колір (HDR має більше одного мільярда кольорів), або діапазон кольорів, представлених за допомогою одного числа (наприклад, коли ми використовуємо зображення у відтінках сірого).

Оскільки ми визначили основи математики нашого розпізнавання образів, які візерунки ми можемо розпізнати? Ну, у нас лише чотири пікселі та два кольори, але ми хочемо, щоб це було просто, чи не так?

У будь-якому випадку, наші варіанти - це два наступних шаблони. Один починається зеленим кольором у верхньому лівому куті, а інший – синім відповідно.

Що таке нейронна мережа?

Перш ніж почати реалізацію нашої власної нейронної мережі, ми повинні уточнити, що таке нейронна мережа насправді. За своєю суттю нейронна мережа – це серія ланцюжкових математичних операцій. Нейронні мережі створені безпосередньо за принципом роботи нашого мозку, звідси й назва.

У будь-якому випадку, нейронна мережа – це мережа взаємопов'язаних «нейронів» (як і наш мозок), які взаємодіють, обробляють і передають інформацію один одному.

Ці нейрони розділені на набір шарів. Зазвичай фахівці говорять про три шари:

- Вхідний рівень: вхідні дані (чотири пікселі).
- Прихований шар: математична магія.
- Вихідний рівень: результат мережі.

У нашому прикладі кількість вхідних елементів є кінцевою. Це означає, що прихований шар, справжня магія, може бути простим і легким для розуміння. Вихідним шаром було б розпізнане зображення, якби він міг його розпізнати.

Для простоти ми лише хочемо навчити його розпізнавати, чи є діагональний зсув кольорів між зеленим і синім. Це означає, що нічого, окрім двох наведених вище прикладів, не буде розпізнано.

Розпізнавання образів через нейронну мережу

Створюючи нейронну мережу, ми маємо почати з певного типу вхідних даних. У нашому випадку ми маємо набір із восьми зображень, кожне по чотири пікселі з комбінацією наших двох кольорів

Що б ми не створили, воно має розпізнавати лише перші два зображення. Зображення перших двох зображень для побудови зображень котів, тоді як решта зображень нудних равликів (не засуджуючи тих, хто любить равликів, просто кажу!).

Вхідний шар

Наш перший крок – створити вхідний шар. Щоб представити наше зображення, нам потрібні чотири нейрони як вхідні дані. Ви можете розглядати ці нейрони як властивості будь-якого роду.

Уявіть, що ви хочете визначити, чи потрібна кімната прибирання. У цьому випадку у нас буде не чотири пікселі, а багато 4 властивостей, пов'язаних із прибиранням кімнати, наприклад Чи кімната вже чиста? Це складське приміщення? (Приміщення для зберігання ніколи не прибирають; запитайте на горіщі!) Чи справді я маю пісосос? Нарешті, чи відбувається відключення електроенергії, яке не дасть мені зависнути?

У будь-якому випадку, повертаючись до наших чотирьох пікселів, ми б представили їх як набір із чотирьох вхідних нейронів.

Прихований шар(и)

Прихований шар отримує дані від вхідного шару. Прихований шар сам по собі може містити багато кроків. Насправді багато кроків у прихованому шарі не є рідкістю. Скільки залежить від того, скільки параметрів ми хочемо проаналізувати. У нашому випадку ми хочемо знати саме одну властивість зображення. Є зсув кольору по діагоналі чи ні?

Це означає, що для цього дуже простого прикладу наш прихований шар буде однією математичною функцією та результатами як безпосереднім результатом нашої функції розпізнавання. Але яку математичну магію нам тут використати?

Фактичні нейронні мережі можуть мати складніші алгоритми, але ми хочемо дотримуватися самої основи нашої математики, додавання та віднімання.

Застосовуючи будь-яку з цих двох операцій до кожного з чотирьох вхідних нейронів, ми маємо потенціал із 16 комбінацій.

Як бачите, потенційні кандидати також позначені як працюють чи ні. Робота в цьому випадку означає, що обчислення дасть відмінні результати для нашого набору з восьми вхідних зображень.

Два кандидати, яких ми могли б використати, виглядають дивно знайомими, просто використовуючи символи замість наших чисел. Це не завжди так, але для цього дуже простого прикладу лише з одним параметром це має сенс. Шаблон, який ми хочемо розпізнати, – це шаблон, за яким його можна розпізнати.

Насправді комп'ютери використовувалися б для пошуку «найкращої» комбінації кандидатів. У зв'язку з експоненціальним зростанням кількості комбінацій під best розуміють евристичний підхід. Інженери починали б з однієї комбінації та, у багатьох ітераціях, дещо її коригували, використовуючи один із кращих результатів для наступної ітерації. У якийсь момент вони можуть зупинитися і вважати, що вони досить близькі до «найкращого» рішення.

Щоб показати експоненціальне зростання, ось таблиця зростання або кількості пікселів, яку ми хочемо розпізнати за один цикл, або кількості потенційних математичних операцій. Комбінації швидко розбігаються.

Нарешті, кожен нейрон передає цей результат «функції активації». Функція активації використовується для визначення виходу нейрона. Він отримує результат обчислення та застосовує певний тип функціональних можливостей прийняття рішень. Трохи спрощено, він вирішує, «активований» нейрон чи ні.

Вихідний рівень

Вихідний рівень об'єднує виходи з прихованого шару та обчислює остаточний прогноз. У той час як прихований шар може мати багато кроків обчислень, вихідний рівень є одним кроком.

У цьому наборі вихідний рівень має два основних призначення:

1. Точка остаточного рішення: вихідний рівень – це місце, де мережа приймає остаточне рішення. Після того, як дані проходять через приховані шари (і вхідний рівень, якщо на те пішло), вихідний рівень обробляє цю інформацію для створення остаточного результату.

2. Інтерпретація результату: вихідний рівень перетворює внутрішнє представлення мережі у формат, зрозумілий людям. Це може бути простий вихід, як-от 0 або 1 , істинний чи хибний, а також більш складні результати, такі як ймовірності, класифікація, результати регресії тощо.

У нашому простому прикладі вихід буде класифіковано простим порівнянням, якщо кінцевий результат дорівнює нулю (нерозпізнаний), нижче нуля (верхній лівий, нижній правий) і вище нуля (верхній правий, нижній лівий). Не надивовижніша з усіх класифікацій, але це те, чого ми хотіли досягти.

Запуск алгоритму виявлення

Тепер, коли ми знаємо основні поняття, настав час спробувати запустити першу ітерацію нашої нейронної мережі та подивитися, як вона оброблятиме наші 4-піксельні зображення.

1. Вхідний рівень: значення кожного пікселя (-1 для зеленого, 1 для синього) передається в мережу.

2. Прихований рівень: кожен нейрон у цьому шарі отримує вхідні дані від усіх чотирьох пікселів, виконує певну магію (ми поговоримо про це за секунду й поки що ігноруємо), а потім застосовує функцію активації, щоб визначити її результат. У нашому простому випадку функція активації перевіряє вихід нейрона на нерівний нуль. Нуль означає, що діагональний перемикач кольорів не виявлено, одиниця означає, що наша нейронна мережа знайшла одиницю.

3. Вихідний рівень: цей рівень об'єднує результати прихованого шару для остаточного прогнозування. У нашому випадку він може виводити значення, близьке до 1 для верхньої лівої та нижньої правої діагоналей і близьке до 0 для іншої діагоналі.

Ваги та упередження

Поки що я приховував від вас одну важливу інформацію. Кожне з'єднання між нейронами має пов'язану з ним «вагу». Ці ваги визначають, наскільки важливою є кожна інформація. Крім того, у кожного нейрона є «упередження» – уявіть це як схильність нейрона спрацьовувати незалежно від його вхідних даних.

Мережа починається з випадкових ваг і зміщень. Під час навчання нейронної мережі ви пропускаєте через нейрони багато різних тестових даних і порівнюєте результат із попередньо визначеним. Відповідно до результату (незалежно від того, була розпізнана собака чи ні), нейронна мережа регулює свої ваги та зміщення, оптимізуючи себе для кращого розпізнавання шаблонів. Але як тренінг мережі знає, як їх налаштувати? Ось де з'являється наша наступна концепція.

Градiєнтний спуск

Уявіть, що ви із зав'язаними очима намагаєтесь знайти найнижчу точку на горбистій місцевості. Ймовірно, ви робите маленькі кроки, відчуваючи землю під ногами, щоб визначити, чи йдете ви вниз чи в гору. По суті, це те, що градієнтний спуск робить для нашої нейронної мережі.

Мережа робить прогноз, порівнює його з правильною відповіддю, а потім коригує його ваги та зміщення, щоб зменшити помилку. Він робить це знову і знову, поступово підвищуючи свою точність. Це процес, який називається «навчання».

Чим більше зображень ми передаємо під час фази навчання, тим кращою буде модель. Після навчання конфігурація моделі експортується та може бути завантажена в інші пристрої (наприклад, мобільні телефони). Навчання моделі зазвичай потребує досить масивних систем зберігання та обчислювальних ресурсів, що запускають мільйони й мільярди навчальних зображень. Пам'ятайте, що тут ми використовуємо лише один параметр, фактичні нейронні мережі мають від десятків тисяч до сотень тисяч, а іноді навіть мільйони.

Масштабування

Як ми можемо додати додаткові функції та продовжувати розвивати цю мережу? А як щодо мережі, яка могла б відрізнити синю діагональну лінію, що починається вгорі чи внизу ліворуч? Це було б легко зробити. Тепер ми знаємо, що існують ваги, тому ми можемо ними скористатися.

Щоб досягти цього, ми додаємо другий вихід. Перший вихід буде запускатися кожного разу, коли буде розпізнано зображення, тоді як другий вихід буде запускатися, лише якщо перший піксель синій. Використовуючи ці два вихідні дані, ми можемо зробити висновок, чи ми виявили діагональну лінію та чи маємо зелену чи синю діагональну лінію, що починається вгорі ліворуч (двійковий вибір).

У будь-якому випадку, хоча наш базовий 4-піксельний приклад є чудовою відправною точкою, розпізнавання зображень у реальному світі набагато складніше. Сучасні нейронні мережі, які використовуються для таких завдань, як розпізнавання обличчя або виявлення об'єктів на фотографіях, використовують мільйони параметрів і навчаються на величезних наборах даних. Однак фундаментальні принципи залишаються незмінними.

Нейронні мережі: особливості та проблеми

Одним із найпотужніших аспектів нейронних мереж є їх здатність автоматично вивчати функції. У нашому простому прикладі мережа може навчитися звертати увагу на конкретні комбінації пікселів, які вказують на діагональну лінію.

У більш складних мережах ранні рівні можуть навчитися виявляти краї або прості форми, тоді як більш глибокі рівні поєднують їх, щоб розпізнавати складніші візерунки, такі як обличчя чи об'єкти. Саме це ієрархічне навчання робить нейронні мережі настільки ефективними для завдань розпізнавання зображень.

Однак, хоча нейронні мережі є потужними, вони не позбавлені проблем:

1. Переобладнання: іноді мережа стає занадто спеціалізованою у своїх навчальних даних і погано працює на нових, небачених прикладах.
2. Обчислювальні ресурси: Навчання складних мереж вимагає значної обчислювальної потужності та часу.
3. Ресурси зберігання: набори даних для навчання вимагають великого обсягу пам'яті з високою пропускнуою здатністю, щоб не відставати від поточних установок навчання на основі GPU та наступного покоління.
4. Якість і кількість даних: тут застосовна стара приказка «сміття входить, сміття виходить». Для ефективного навчання мережам потрібні великі обсяги високоякісних різноманітних даних.
5. Можливість інтерпретації: на відміну від простіших алгоритмів, може бути важко точно зрозуміти, чому нейронна мережа прийняла певне рішення.

Типи нейронних мереж

Хоча ми говорили про нейронні мережі так, ніби існує лише один тип, насправді це не так. Існує багато різних підходів до глибокого навчання на основі нейронної мережі. Однак найбільш помітними є три конкретні типи, причому багато сучасних додатків використовують гібридні підходи, поєднуючи різні типи нейронних мереж, щоб використовувати їхні сильні сторони.

Нейронні мережі прямого зв'язку (FNN або FFNN)

Найпростішим типом нейронної мережі, який ми використовували в нашому 4-піксельному прикладі, є нейронна мережа прямого зв'язку, скорочена як FNN. У нейронній

мережі прямого зв'язку інформація тече лише в одному напрямку: від вхідного рівня через прихований(-і) шар(и) до вихідного рівня. У мережі немає петель і циклів.

FNN чудово підходять для простих завдань класифікації. Вони також є чудовим способом послужити основою для розуміння більш складних мереж.

Однак вони мають обмеження, коли йдеться про обробку послідовних даних або захоплення просторових зв'язків у зображеннях. Проте, хоча їх можна використовувати для розпізнавання зображень (як ми робили вище), варіанти використання зазвичай обмежуються дуже простими зображеннями.

Згорткові нейронні мережі (CNN)

Коли справа доходить до розпізнавання зображень, згорткові нейронні мережі є зірками шоу. Згорткові нейронні мережі (зазвичай скорочені як CNN) спеціально розроблені для обробки сіткових даних, таких як зображення (піксельні сітки).

Ключовою особливістю CNN є згортковий рівень, який застосовує фільтри (або ядра) для виявлення таких функцій, як грані, текстури та форми. У міру поглиблення мережі вона може розпізнавати складніші шаблони, що робить CNN надзвичайно хорошими в таких завданнях, як виявлення об'єктів і розпізнавання облич.

Згорткові нейронні мережі є робочими конячками розпізнавання зображень. Вони мають дивовижну здатність кодувати складні просторові візерунки в ієрархічні подання ознак.

Повторювані нейронні мережі (RNN)

Останнім великим підходом є рекурентна нейронна мережа або RNN.

Хоча RNN зазвичай не використовуються для розпізнавання статичного зображення, про них варто згадати через їх важливість в обробці послідовних даних. RNN мають цикли, які дозволяють інформації зберігатися, що робить їх ідеальними для завдань, пов'язаних із прогнозуванням часових рядів або послідовності.

У контексті комп'ютерного зору RNN можуть бути корисними для аналізу відео, де важливий часовий зв'язок між кадрами. Їх також можна об'єднати з CNN для створення потужних моделей для таких завдань, як субтитри до зображень або класифікація відео.

Майбутнє розпізнавання зображень

У цьому розділі ми розглянули багато базових напрацювань, від основ обробки інформації нейронними мережами до градієнтного спуску як найпоширенішого підходу до навчання нейронної мережі.

Хоча наш початковий приклад був спрощеним, він ілюструє фундаментальні концепції, які лежать в основі навіть найдосконаліших систем розпізнавання зображень. Незалежно від того, чи маєте ви справу з чотирма пікселями чи чотирма мільйонами, основні ідеї ваг, зміщень і навчання шляхом виправлення помилок залишаються незмінними.

Оскільки світ продовжує вдосконалювати архітектуру нейронних мереж і методи навчання, можливості систем розпізнавання зображень швидко розширюються. Від додатків для мобільних телефонів, медичної діагностики до автономних транспортних засобів, додатків величезна кількість і зростає.

Величезна кількість цифрових зображень, які сьогодні стали всюдисущими, вимагають посиленних зусиль для створення ефективних потреб для їх автоматичних анотацій і механізмів пошуку. Класифікація цифрових зображень стає одним із фундаментальних видів діяльності, який можна розглядати як фундаментальну передумову для всіх інших процесів обробки зображень. У класифікації зображень ми могли б слідувати загальній парадигмі розпізнавання образів. У розпізнаванні образів кожен об'єкт описується сукупністю ознак, що формує багатовимірний простір, у якому відбуваються всі дії розрізнення. На цьому етапі стають доступними різні класифікатори, як лінійні, так і нелінійні, включаючи машини опорних векторів (SVM), лінійні класифікатори, поліноміальні класифікатори, радіально-базисні нейронні мережі (RBFNN), нечіткі системи на основі правил тощо. Незалежно від того, який класифікатор був використаний обрано, формування відповідної характеристики стає першорядним. Проблема формування простору

ознак у випадку зображень ще складніша. З одного боку, у нас є багато різних альтернатив. З іншого боку, різноманітність зображень сприяє підвищенню рівня складності та складності. На зображеннях ми зустрічаємо різноманітні зображення, що демонструють різні форми, кольори, текстуру тощо, але належать до одного класу. Зображення можна описати інтенсивністю кольору кожного пікселя або ще краще деякими дескрипторами. У цьому дослідженні наша мета полягає в тому, щоб вивчити та кількісно визначити дискримінаційні властивості дескрипторів зображень MPEG-7 у проблемах класифікації. Вони досліджуються в поєднанні з двома основними категоріями класифікаторів, такими як SVM і RBFNN.

На жаль, стає очевидним, що будь-який класифікатор, який вимагає високої точності навчання, може не досягти хорошої здатності до узагальнення. Оскільки як цільові результати, так і розподіли невидимих зразків невідомі, неможливо обчислити помилку узагальнення прямим шляхом. Існує два основні підходи до оцінки помилки узагальнення, а саме аналітична модель і перехресна перевірка (CV). Загалом, аналітичні моделі обмежують помилку узагальнення для будь-яких невидимих зразків і не розрізняють навчені класифікатори з однаковою кількістю ефективних параметрів, але різними значеннями параметрів. Таким чином, межі похибок, надані цими моделями, зазвичай є вільними [1]. Основною проблемою аналітичних моделей є оцінка кількості ефективних параметрів класифікатора, які можна вирішити за допомогою VC-вимірностей [2]. VC-розмірність класифікатора визначається як найбільша кількість вибірок, які можуть бути розбиті цим класифікатором [2]. Однак для нелінійних класифікаторів, наприклад нейронних мереж, можна знайти лише вільні межі розмірів VC, і це накладає серйозні обмеження на застосовність аналітичних моделей до нелінійних класифікаторів, за винятком SVM [3]. Незважаючи на те, що CV використовує справжні цільові результати для невидимих зразків, це займає багато часу для великих наборів даних, і класифікатори *CL* повинні бути навчені *C*-кратному *CV* і *L* вибору параметрів класифікатора. Методи *CV* оцінюють очікувану помилку узагальнення замість її обмеження, тому вони не гарантують, що остаточно побудований класифікатор матиме хорошу здатність до узагальнення [1].

У класифікації зображень можна не очікувати, що класифікатор, навчений використовувати одну категорію зображень (скажімо, тварин), правильно класифікуватиме зображення, що надходять з деяких інших категорій (наприклад, овочів). У цьому випадку можна переглянути навчальний набір даних, додавши навчальні зразки овочів і перенавчаючи класифікатор, щоб включити новий клас зображень. Наприклад, у нашому наборі даних ми маємо зображення корови, але не літака, тому ми не могли очікувати, що класифікатор, навчений за допомогою нашого набору даних, правильно розпізнає літак. Очікується, що класифікатор зображень добре працюватиме для тих класів, які використовувалися для його навчання, припускаючи, що зображення, що належать до того самого класу, концептуально подібні та такі, що їхні значення дескрипторів також мають бути подібними. Тобто невидимі зразки, подібні до навчальних зразків, з точки зору *sur*-типу відстані в просторі ознак, меншого за заданий поріг, вважаються більш важливими. Таким чином, при оцінці узагальнювальних можливостей класифікаторів зображень можна не враховувати ті зображення, які абсолютно не схожі на існуючі в навчальному наборі.

Загалом, проблеми класифікації зображень є проблемами класифікації з кількома класами, і важко знайти класифікатор із хорошими властивостями узагальнення. У цій роботі ми прагнемо знайти класифікатор зображень, що має кращу здатність до узагальнення та інтерпретацію стосовно предметних знань у класифікації зображень. Ми зосереджуємося на пошуку оптимальної кількості сприйнятливих полів для RBFNN, щоб класифікувати зображення з меншою помилкою узагальнення до невидимих зображень.

Поняття та реалізація локалізованої помилки узагальнення

Враховуючи очікуване розмаїття зображень, які підлягають класифікації, можна легко уявити, що не існує алгоритму класифікації, здатного здійснити класифікацію з нульовою помилкою. Це просте та дуже інтуїтивно зрозуміле спостереження стосується зображень, які сильно відрізняються від тих, які класифікатор показував на етапі навчання. Іншими

словами, ми визнаємо, що будь-який класифікатор має деякі обмежені можливості узагальнення.

Дизайн архітектури RBFNN

У подальшому ми обмежимося RBFNN з рецептивними полями Гауса. Ми застосовуємо стандартний алгоритм кластеризації (скажімо, k-засоби, самоорганізуючі карти тощо), щоб знайти розташування сприйнятливих полів мережі. Як правило, це робиться після фіксації кількості рецептивних полів. Вибір цього числа не є тривіальним завданням, і його відповідний вибір впливає на здатність мережі до узагальнення.

У цьому дослідженні, керуючись концепцією, що семантично схожі зображення повинні демонструвати подібність у просторі ознак, ми запропонували застосування моделі помилки локалізованого узагальнення до класифікації зображень. Ця модель фіксує помилку узагальнення для невидимих зразків, подібних до навчальних зразків. Експериментальні результати показують, що RBFNN, навчений з використанням мінімізації локалізованої помилки узагальнення, перевершує «стандартний» RBFNN і багатокласовий SVM.

Розробка структурної схеми

Для реалізації завдання розпізнання образів використовується радіально-базисна нейронна мережа.

Описані вище багатосарові мережі сигмоїдального типу з математичної точки зору виконують апроксимацію функції декількох змінних $X=RN$ у безліч вихідних змінних $Y=RM$. Оскільки сигмоїдальна функція, що грає роль функції активації нейронів, має ненульове значення на всьому діапазоні вхідних даних, то в перетворенні мережею вхідних даних у вихідні беруть участь багато хто (якщо не все) її нейрони. Внаслідок цього апроксимація сигмоїдальними (і, природно, лінійними) нейронами називається глобальною апроксимацією.

Навчання радіальної мережі

Процес навчання радіальної мережі розпадається на два етапи:

- підбор параметрів радіальної функції f_i для кожного радіального нейрона (у випадку функції Гауса це центр C_i і параметр ширини s_i);
- підбор ваг вихідного шару нейронів.

При цьому другий етап значно простіше першого, оскільки зводиться до обчислення вираження $W=G^+*D$, де основні обчислювальні витрати – розрахунок псевдоінверсії матриці Гріна G .

Завдання відшукування параметрів радіальних функцій Гауса для всіх нейронів першого шару у свою чергу розпадаються на дві підзадачі:

- визначення центрів C_i ;
- розрахунок параметрів ширини s_i .

Зрозуміло, що основними вимогами до розташування центрів C_i в області визначення вхідних даних X є:

- повнота покриття області визначення;
- рівномірність розподілу.

Саме цим вимогам відповідає рішення по кластеризації даних, що дається нейронною мережею із самоорганізацією на основі конкуренції. Отже, алгоритми навчання, використовувані в цих мережах для відшукування усереднених векторів у кластерах даних, безпосередньо застосовні й у радіальних мережах для відшукування центрів радіальних функцій.

Після визначення місця розташування всіх центрів радіальних функцій C_i відбувається підбор параметрів s_i , що визначають величину області охопту, у якій значення радіальної функції перевищує граничне значення ϵ . Такий підбор повинен забезпечити, з одного боку, покриття всього простору вхідних даних i , з іншого боку, незначність перекриття сусідніх зон.

На рисунку 1 представлена структурна схема системи управління радіально-базисної нейронної мережі для розпізнання образів, під якою будемо розуміти середовище, у яку

вкладений ОУ, у свою чергу утримуючий у собі УС. Як видно з рисунка, можна затверджувати, що УС управляє не тільки ОУ, але всією системою. Під середовищем у системі можна розуміти різні об'єднання об'єктів. Будемо називати середовищем W сукупність об'єктів, що лежать поза УС; середовищем S – сукупність об'єктів, що лежать поза ОУ; середовищем U – всю систему.

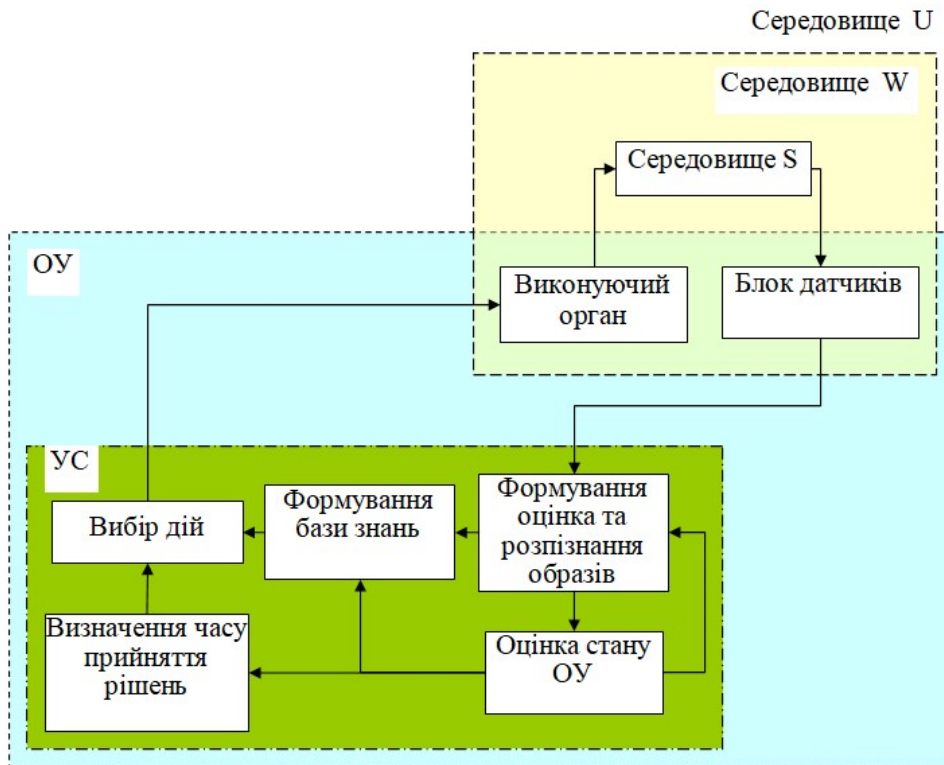


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів радіально-базисної нейронної мережі для розпізнання образів. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем радіально-базисної нейронної мережі для розпізнання образів; Досліджена система радіально-базисної нейронної мережі для розпізнання образів; На основі отриманих результатів досліджень створена програмна реалізація системи радіально-базисної нейронної мережі для розпізнання образів. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання радіально-базисної нейронної мережі для розпізнання образів. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». CEUR Workshop Proceedings Volume 3664, 2024, Pages 11-23.
2. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». Lecture Notes in Networks and Systems, 2023, 729 LNNS, pp. 104–112.
3. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». Advanced Information Systems, 2023, 7(2), pp. 49-56.
4. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based

- monitoring system for critical infrastructures». CEUR Workshop Proceedings, Volume 3530, 2023, pp. 256-265.
5. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.
 6. Smirnov, O., Neskorođieva, T., Fedorov, E., Rudakov, K., Neskorođieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12.
 7. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.
 8. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskiy, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34.
 9. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems. Volume 152, 2021, Pages 85-98.
 10. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.
 11. Smirnov O., Neskorođieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207.
 12. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.
 13. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
 14. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
 15. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.
 16. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.
 17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.
 18. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.
 19. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
 20. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.
 21. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019.
 22. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.
 23. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.
 24. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.