

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра комп'ютерних інтелектуальних технологій та систем
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Нейромережева обробка та розпізнавання об'єктів
в умовах кольорової близькості

(тема)

Виконав:

студент II курсу, групи КІТм-19-1
Гнібеда А.О.
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерні інтелектуальні технології
(повна назва освітньої програми)

Керівник: проф. Руденко О.Г.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТС Руденко О.Г.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
Кафедра _____ комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти _____ другий (магістерський)
Спеціальність _____ 123 – Комп'ютерна інженерія
(код і повна назва)
Тип програми _____ освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Комп'ютерні інтелектуальні технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 2020 р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Гнібеді Артему Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Нейромережева обробка та розпізнавання об'єктів в умовах кольорової близькості

затверджена наказом по університету від “ 11 ” листопада 2020 р. № 1582Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 10 грудня 2020 р.

3. Вхідні дані до роботи 1) штучні нейронні мережі; 2) нейромережева обробка;
3) розпізнавання об'єктів; 4) кольорова близькість.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) аналіз проблемної області і постановка задачі;

2) модель нейронної мережі для обробки та розпізнавання об'єктів в умовах кольорової близькості

3) програмна реалізація нейронної мережі для обробки та розпізнавання об'єктів в умовах кольорової близькості

4) висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайдів 10

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд стану проблеми та постановка задачі	11.11.20–12.11.20	
2	Аналіз літератури	12.11.20–21.11.20	
3	Вибір методів рішення для реалізації та їх обґрунтування	21.11.20–29.11.20	
4	Розробка нейронної мережі для обробки та розпізнавання об'єктів в умовах кольорової близькості	29.11.20–02.12.20	
5	Оформлення пояснювальної записки	02.12.20–10.12.20	
6	Оформлення графічної частини	10.12.20–11.12.20	

Дата видачі завдання 11 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Руденко О.Г.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 105 с., 20 рис., 1 табл., 2 дод., 21 джерел.

МОРФОЛОГІЯ, ФІЛЬТРАЦІЯ ЗОБРАЖЕННЯ, КОЛІРНИЙ ПРОСТОР, ЗГЛАДЖУВАННЯ ЗОБРАЖЕНЬ, ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ, ПЕРЦЕПТРОНОМ, БІНАРИЗАЦІЯ

Метою атестаційної роботи є проаналізувати готові рішення та розробити методи зберігання, обробки і розпізнавання об'єктів в умовах кольорової близькості.

У ході виконання атестаційної роботи були розглянуті всі сучасні рішення розробки нейронної системи і способи розпізнавання об'єктів в кольоровій близькості

ABSTRACT

Explanatory note of attestation work: 92 pages, 20 figures, 1 tables, 2 appendices, 21 sources.

MORPHOLOGY, IMAGE FILTRATION, COLOR SPACE, IMAGE SMOOTHING, IMAGE FILTRATION, PERCEPTRON, BINARIZATION

The purpose of certification work is to analyze ready-made solutions and develop methods of storage, processing and recognition of objects in terms of color proximity.

During the attestation work all modern solutions of neural system development and methods of recognition of objects in color proximity were considered.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	12
1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТА ОБЗОР ЛІТЕРАТУРИ	14
1.1 Визначення, пов'язані з теорією розпізнавання	14
1.2 Деякий історичний контекст глибокого навчання	15
1.3 Три класи глибокого навчання	23
1.3.1 Основна термінологія глибокого навчання	23
1.3.2 Трестороння категоризація	24
1.3.3 Глибокі мережі для безконтрольного або генеративного навчання ..	26
1.3.4 Глибокі мережі для контрольованого навчання	32
1.3.5 Гібридні глибинні мережі	35
1.4 Моделювання та обробка природної мови	39
1.4.1 Мовне моделювання	39
1.4.2 Обробка природної мови	46
2 МЕТОДИ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ КОМП'ЮТЕРНОГО БАЧЕННЯ	56
2.1 Вибрані програми в розпізнаванні об'єктів та комп'ютерне бачення ..	56
2.1.1 Безконтрольне або генеративне навчання особливостей	57
2.1.2 Навчання та класифікація ознак під наглядом	59
2.2 Колірний простір	66
2.2.1 Людський зір	66
2.2.2 Представлення кольору в машинній графіці	67
2.2.3 Кольорова палітра RGB	68
2.2.4 Кольорова палітра CIE LAB	69
3 МЕТОДИ ОБРОБКИ ЗОБРАЖЕННЯ ТА ЇХ ФІЛЬТРАЦІЯ	70
3.1 Порогова обробка (Thresholding)	70
3.2 Лінійні фільтри	71

3.3 Нелінійні фільтри	72
3.4 Математична морфологія	73
3.5 Метод класифікації зображення: персептрон	76
4 РОЗРОБКА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ	78
4.1 Постановка завдання	78
4.2 Використовувані технології	79
4.3 Генерація тестового набору.....	80
4.4 Фільтрація і векторизація зображень.....	81
4.5 Реалізація персептрона.....	82
ВИСНОВКИ	84
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	85
ДОДАТОК А	87
ДОДАТОК Б	94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DNN – глибинні нейронні мережі (англ., Deep Neural Net).

MLP – алгоритм стиснення без втрат (англ., Meridian Lossless Packing)

CRF – умовні випадкові поля (англ., Conditional Random Fields)

SGD – стохастичний градієнтний спуск (англ., Stochastic Gradient Descent)

GMM – узагальнений метод моментів (англ., Generalized Method of Moments)

MaxEnt – максимальна ентропія (англ., Maximum Entropy)

RBM – обмежена машина Больцмана (англ., restricted Boltzmann machine)

SPN – Спрямований (орієнтований) ациклічний граф (англ., directed acyclic graph)

FIR – фільтр зі скінченною імпульсною характеристикою (англ., Finite Impulse Response)

ELM – машини екстремального навчання (англ., Extreme Learning Machines)

ВСТУП

Теорія розпізнавання образів є одним з основних розділів кібернетики, як в теоретичному, так і в прикладному плані. Так, автоматизація деяких процесів передбачає створення пристроїв, здатних реагувати на мінливі характеристики зовнішнього середовища деякою кількістю позитивних реакцій.

Базою для вирішення завдань такого рівня є результати класичної теорії статистичних рішень, в рамках якої розроблено алгоритми визначення класу, до якого може бути віднесений об'єкт, що розпізнається. На базі статистики за рахунок застосування розділів прикладної математики, теорії інформації, методів алгебри логіки, математичного програмування і системотехніки заснована теорія розпізнавання.

Крім того, в основі теорії розпізнавання графічних образів лежать концепції теорії обробки зображень. У процесі розпізнавання кольорових об'єктів застосовується фільтрація зображень, операції математичної морфології, зміна яскравості і контрасту зображень, квантування кольору і перетворення графічних зображень в інший колірний простір.

На даний момент розпізнавання образів – одне з провідних напрямків кібернетики: воно спрощує взаємодію людини з комп'ютером і створює передумови для застосування різних систем штучного інтелекту.

В даній роботі розглядаються методи розподілу зображень на колірні складові, виділення, фільтрація і подальше розпізнавання образів з даного зображення.

1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТА ОБЗОР ЛІТЕРАТУРИ

1.1 Визначення, пов'язані з теорією розпізнавання

Розпізнавання образів (об'єктів, сигналів, ситуацій, явищ чи процесів) – завдання ідентифікації об'єкта або визначення будь-яких його властивостей по його зображенню (оптичне розпізнавання) або аудіозаписи (акустичне розпізнавання) або іншим характеристикам.

Образ – це опис об'єкта або процесу, що дозволяє виділяти його з навколишнього середовища і згрупувати з іншими об'єктами або процесами для прийняття необхідних рішень [1]. Образи мають характерні об'єктивні властивості в тому сенсі, що різні люди, які навчаються на різному матеріалі спостережень, здебільшого однаково і незалежно один від одного класифікують одні і ті ж об'єкти.

Безліч зображень, об'єднаних будь-якими загальними властивостями, називається образом. Методику віднесення елемента до якого-небудь образу називають вирішальним правилом. Гістограмою називають графічне представлення розподілу яркостей зображення.

Ознакою зображення називається його найпростіша відмінна характеристика або властивість [2]. Деякі ознаки є природними в тому розумінні, що вони встановлюються візуальним аналізом зображення, тоді як інші, так звані штучні ознаки, виходять в результаті його спеціальної обробки або вимірювань. До природних ознак належать світлота (яскравість) і текстура різних областей зображення, форма контурів об'єктів. Гістограми розподілу яскравості і спектри просторових частот дають приклади штучних ознак.

Метрика – спосіб визначення відстані між елементами деякої множини. Чим менший цей період, тим більше схожими є об'єкти (числа, функції, символи, звуки й інше). Зазвичай елементи задаються у вигляді набору чисел, а метрика – у вигляді функції. Від вибору уявлення образів і реалізації метрики

залежить ефективність програми, один алгоритм розпізнавання з різними метриками буде помилятися з різною частотою.

Навчання – процес вироблення в деякій системі тієї чи іншої реакції на групи зовнішніх ідентичних сигналів шляхом багаторазового впливу на систему зовнішнього коригування. Таке зовнішнє коригування у навчанні прийнято називати «заохоченням» і «покаранням». Механізм генерації цього коригування практично повністю визначає алгоритм навчання. Самонавчання відрізняється від навчання тим, що додаткова інформація про вірність реакції системі не повідомляється.

Адаптація – процес зміни параметрів і структури системи, а можливо – і керуючих впливів, на основі поточної інформації з метою досягнення певного стану системи при початковій невизначеності і мінливих умовах роботи.

1.2 Деякий історичний контекст глибокого навчання

Донедавна більшість методів машинного навчання та обробки сигналів використовували архітектури з неглибокою структурою. Ці архітектури, як правило, містять щонайбільше один або два шари нелінійних перетворень ознак. Прикладами дрібної архітектури є змішані гаусові моделі (GMM), лінійні або нелінійні динамічні системи, умовні випадкові поля (CRF), моделі максимальної ентропії (MaxEnt), метод опорних векторів (SVM), логістична регресія, регресія ядра, багатошарові персептрони (MLP) з єдиним прихованим шаром, включаючи машини для екстремального навчання (ELM). Наприклад, SVM використовують модель неглибокого лінійного розділення шаблону з одним або нульовим шаром перетворення об'єктів, коли використовується трюк ядра або іншим чином. (Помітними винятками є останні методи ядра, які були направлені та інтегровані в глибоке навчання; наприклад, [6]). Неглибокі архітектури продемонстрували свою ефективність у вирішенні багатьох простих або добре обмежених проблем, але їх обмежене моделювання та репрезентативна сила можуть спричинити труднощі при

роботі з більш складними реальними програмами, що включають природні сигнали, такі як людська мова, природний звук та мова, і природні образи та зорові сцени.

Однак механізми обробки інформації людини (наприклад, зір та слух) наводять на думку про необхідність глибоких архітектур для вилучення складної структури та побудови внутрішньої репрезентації з багатих сенсорних даних. Наприклад, системи людського мовлення та сприйняття оснащені чітко нашарованими ієрархічними структурами для перетворення інформації з рівня форми сигналу на лінгвістичний рівень. Подібним чином зорова система людини також має ієрархічну природу, переважно з точки зору сприйняття, але, що цікаво також, з точки зору «покоління»). Природно вірити, що сучасний рівень може бути вдосконалений у обробці цих типів природних сигналів, якщо можна розробити ефективні алгоритми глибокого навчання.

Історично концепція глибокого навчання виникла в результаті штучних досліджень нейронних мереж. (Отже, іноді можна почути дискусію про «нейронні мережі нового покоління».) Нейронні мережі з прямим зв'язком або MLP з великою кількістю прихованих шарів, які часто називають глибокими нейронними мережами (DNN), є хорошими прикладами моделей з глибокою архітектурою. Зворотне розповсюдження (BP), популярне у 1980-х роках, було відомим алгоритмом вивчення параметрів цих мереж. На жаль, одна BP тоді на практиці не працювала. для вивчення мереж з більш ніж невеликою кількістю прихованих шарів. BP базується на локальній інформації про градієнт і починається зазвичай у деяких випадкових початкових точках. Він часто потрапляє у поганий локальний оптимум, коли використовується алгоритм періодичного або навіть стохастичного градієнтного спуску BP. Серйозність значно зростає із збільшенням глибини мереж. Ця проблема частково відповідає за спрямування більшості досліджень машинного навчання та обробки сигналів від нейронних мереж до неглибоких моделей, що мають опуклі функції втрат (наприклад, SVM, CRF та MaxEnt), для яких глобальний оптимум може бути ефективно отриманим за рахунок зменшення

потужності моделювання, хоча продовжувалась робота над нейронними мережами з обмеженим масштабом та впливом.

Труднощі оптимізації, пов'язані з глибинними моделями, були емпірично полегшені, коли в двох семінарських роботах було представлено розумно ефективний, некерований алгоритм навчання [14]. У цих роботах було представлено клас глибоких генеративних моделей, який називається мережею глибоких переконань (DBN). DBN складається з набору обмежених машин Больцмана (RBM). Основним компонентом DBN є алгоритм пошарового навчання, який оптимізує ваги DBN в момент тимчасової складності, лінійно залежить від розміру та глибини мережі. Окремо і з певним подивом ініціалізація ваг MLP за допомогою відповідно налаштованого DBN часто дає набагато кращі результати, ніж для випадкових ваг. Таким чином, MLP з великою кількістю прихованих шарів або глибокими нейронними мережами (DNN), які вивчаються з попередньою підготовкою DBN без нагляду з подальшим точним налаштуванням зворотного розповсюдження, іноді в літературі також називають DBN [12].

Незалежно від розвитку RBM, у 2006 році були опубліковані дві альтернативні, не ймовірнісні, негенеративні, неконтрольовані глибокі моделі без нагляду. Одним із них є варіант автокодера з пошаровим навчанням, схожим до навчання DBN [9]. Інша модель – енергетична модель з безконтрольним вивченням розріджених надмірно повноцінних уявлень. Обидва вони можуть бути ефективно використані для попередньої підготовки глибокої нейронної мережі, подібно до DBN.

На додаток до забезпечення хороших точок ініціалізації, DBN має інші привабливі властивості. По-перше, алгоритм навчання ефективно використовує немічені дані. По-друге, його можна трактувати як імовірнісну генеративну модель. По-третє, проблема надмірної підгонки, яка часто спостерігається в моделях з мільйонами параметрів, таких як DBN, і проблема недостатньої підгонки, яка часто трапляється в глибоких мережах, може бути ефективно усунена на етапі генеральної попередньої підготовки.

Використання прихованих шарів з великою кількістю нейронів у DNN значно покращує модельну силу DNN та створює багато оптимальних конфігурацій. Навіть якщо вивчення параметрів потрапило в локальний оптимум, отриманий DNN все одно може працювати досить добре, оскільки шанс мати поганий локальний оптимум нижчий, ніж коли в мережі використовується невелика кількість нейронів. Однак використання глибоких і широких нейральних мереж викликало б великий попит на обчислювальну потужність під час навчального процесу, і це одна з причин, чому лише останніми роками дослідники почали досліджувати як глибокі, так і широкі нейронні мережі всерйоз.

Кращі алгоритми навчання та різні нелінійності також сприяли успіху DNN. Алгоритми стохастичного зниження градієнта (SGD) є найефективнішим алгоритмом, коли навчальний набір великий і надлишковий, що притаманно більшості програм. Нещодавно показано, що SGD є ефективним для розпаралелювання на багатьох машинах з асинхронним режимом [12] або на декількох графічних процесорах через конвексний BP. Крім того, SGD часто може дозволити тренуванню вистрибнути з місцевих оптимумів через шумні градієнти, оцінені з однієї або невеликої партії зразків. Інші алгоритми навчання, такі як Hessian free [15] або методи підпростору Крилова, показали подібну здатність.

Для задачі оптимізації вивчення DNN очевидно, що кращі методи ініціалізації параметрів призведуть до кращих моделей, оскільки оптимізація починається з цих початкових моделей. Однак очевидним було те, як ефективно ініціалізувати параметри DNN та як використання великих обсягів навчальних даних може полегшити проблему навчання до недавнього часу.

Процедура попередньої підготовки DBN не єдина, що дозволяє ефективно ініціювати DNN. Альтернативний неконтрольований підхід, який виконує однаково добре, полягає у попередній підготовці DNN шарів за шаром, розглядаючи кожну пару шарів як безшумний автокодер, регуляризований шляхом встановлення випадкової підмножини вхідних

вузлів на нуль. Інша альтернатива – використовувати контрактні автокодери для тієї ж мети, надаючи перевагу представленням, які є більш стійкими до варіацій введення тобто штрафуючи градієнт діяльності прихованих блоків щодо входів. Крім того, Ранзато та ін. [17] розробили симетричну машину розрідженого кодування (SESM), яка має дуже подібну архітектуру до RBM як будівельних блоків DBN. SESM також може використовуватися для ефективної ініціалізації навчання DNN. На додаток до невідконтрольної попередньої підготовки з використанням шарових процедур, контрольована попередня підготовка, або інколи її називають дискримінаційною попередньою підготовкою, також виявилася ефективною та у випадках, коли маркованих навчальних даних достатньо, ефективніше, ніж некеровані методи попередньої підготовки. Ідея дискримінаційної попередньої підготовки полягає в тому, щоб почати з одношарового MLP, навченого за алгоритмом ВР. Кожного разу, коли ми хочемо додати новий прихований шар, ми замінюємо вихідний рівень випадковим чином ініціалізованим новим прихованим і вихідним шаром і тренуємо весь новий MLP (або DNN), використовуючи алгоритм ВР.

Дослідники, які застосовують глибоке навчання мови та зору, проаналізували, що DNN фіксують у мові та зображеннях. Наприклад, застосував метод зменшення розмірності для візуалізації взаємозв'язку між векторами ознак, вивченими DNN. Вони виявили, що приховані вектори діяльності DNN зберігають структуру подібності векторів об'єктів у багатьох масштабах, і що це особливо актуально для функцій фільтрабанку. Більш досконалий метод візуалізації, заснований на генеративному процесі зверху вниз у зворотному напрямку класифікаційної мережі, нещодавно був розроблений Цайлером та Фергюсом [21] для вивчення особливостей, які фіксують глибокі згорткові мережі із даних зображення. Показано, що потужність глибинних мереж полягає у їх здатності виділяти відповідні риси та спільно здійснювати дискримінацію [15].

Ще одним способом стислого введення DNN ми можемо переглянути

історію штучних нейронних мереж, використовуючи «ажіотажний цикл», який є графічним зображенням зрілості, прийняття та соціального застосування конкретних технологій. Версія графіка циклів 2012 року, складена Гартнером, і показана на рисунку 1.1. Він має намір показати, як технологія або програма буде розвиватися з часом (згідно з п'ятьма фазами: технологічний тригер, пік завищених очікувань, впадина розсарування, схил просвітлення та плато виробництва), а також забезпечити джерело розуміння для управління його розгортанням.



Рисунок 1.1 – Графік гіперциклу Гартнера, що представляє п'ять фаз технології

Застосувавши гіперцикл Гартнера до розвитку штучної нейронної мережі, було створено рисунок 1.2 для узгодження різних поколінь нейронної мережі з різними фазами, позначеними в ажіотажному циклі. Пік активності («очікування» чи «медіа-ажіотаж» на вертикальній осі) припав на кінець 1980-х – початок 1990-х років, що відповідало висоті того, що часто називають «другим поколінням» нейронних мереж. Мережа глибоких переконань (DBN) та швидкий алгоритм її навчання були винайдені в 2006 році. Коли DBN був використаний для ініціалізації DNN, навчання стало високоефективним, і це

надихнуло на подальші швидкозростаючі дослідження (фаза «просвітлення», показана на рисунку 1.2). Застосування DBN та DNN до видобутку та розпізнавання мовлення в масштабах галузі розпочалося у 2009 році, коли співпрацювали провідні наукові та промислові дослідники, що мають глибоке навчання та мовний досвід. Ця співпраця швидко розширила роботу з розпізнавання мови за допомогою методів глибокого навчання до все більших успіхів. Очікується, що висота фази «плато продуктивності», яка, на мою думку, ще не досягнута буде вищою, ніж на стереотипній кривій (обведена знаком питання на рисунку 1.2), і позначена штриховою лінією, яка рухається прямо вгору.

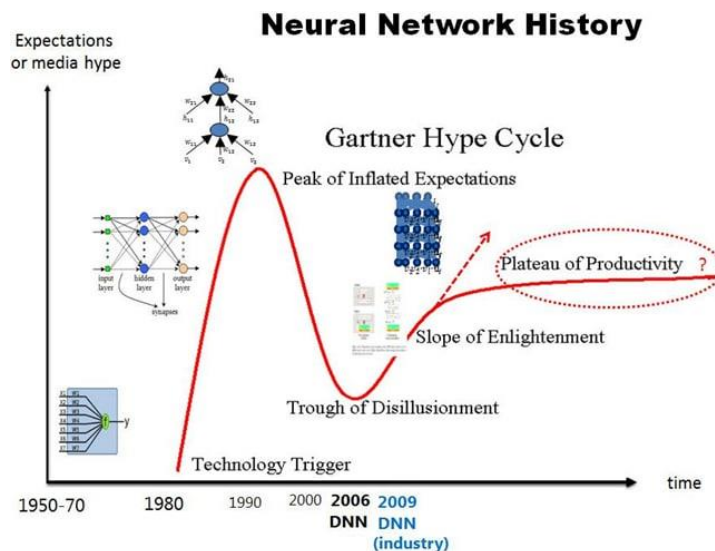


Рисунок 1.2 – Застосування графіку гіперциклів Гартнера до аналізу історії технології штучних нейронних мереж

На рисунку 1.3 показано історію розпізнавання мови, яку склав NIST, організовану шляхом побудови графіку коефіцієнта помилок у словах (WER) як функції часу для ряду складних завдань розпізнавання мови. Всі результати WER були отримані з використанням технології GMM – HMM. З рис. 1.3 виділено одне особливо складне завдання (комутатор), можна побачити рівну криву протягом багатьох років із використанням технології GMM – HMM, але після використання технології DNN WER різко падає (позначена червоною

зіркою на рис. 1.4).

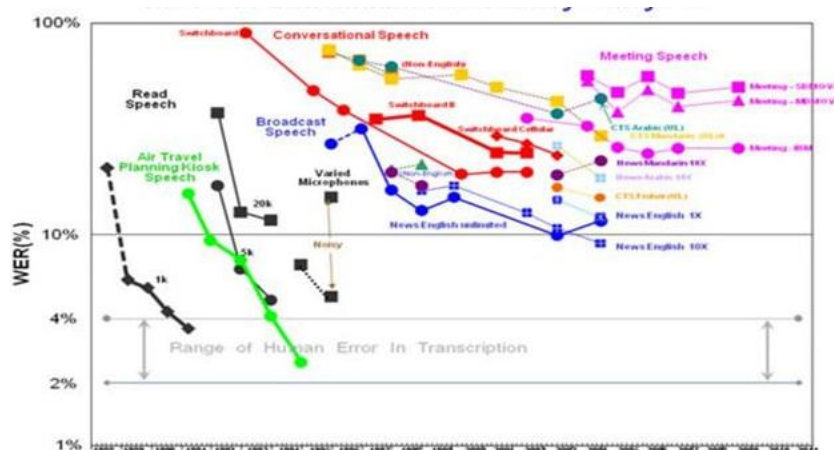


Рисунок 1.3 – Відомий сюжет NIST, що показує історичні показники помилок розпізнавання мови, досягнуті підходом GMM-HMM для ряду складних завдань розпізнавання мови

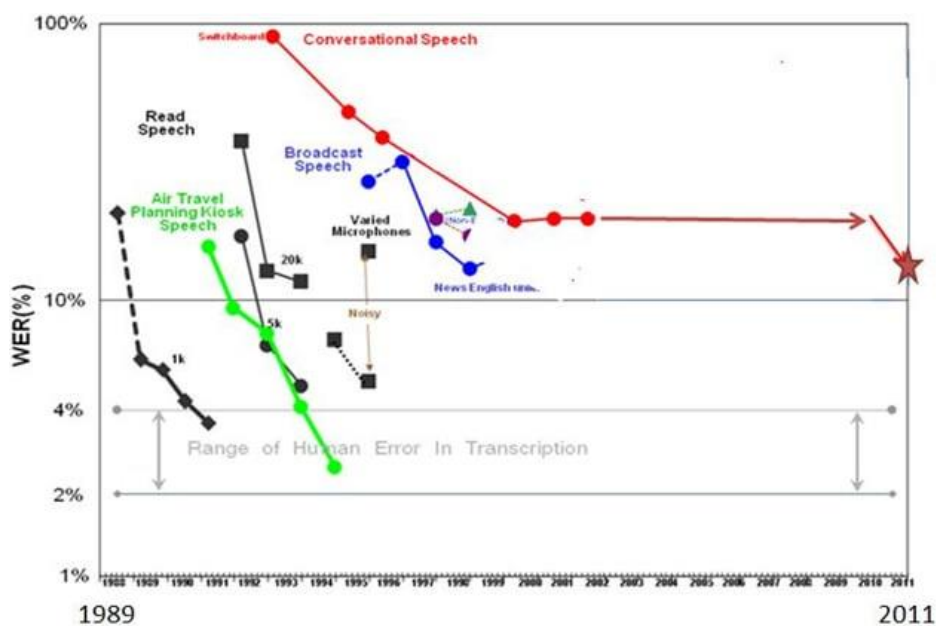


Рисунок 1.4 – Витяг WER одного завдання з рис. 1.3 та додавання суттєво нижчого WER (позначеного зірочкою), досягнутого за допомогою технології DNN

1.3 Три класи глибокого навчання

1.3.1 Основна термінологія глибокого навчання

Глибоке навчання – клас технік машинного навчання, де багато рівнів етапів обробки інформації в ієрархічних контрольованих архітектурах використовуються для некерованого вивчення особливостей та для аналізу/класифікації шаблонів. Суть глибокого навчання полягає в обчисленні ієрархічних ознак або подань даних спостережень, де особливості або фактори вищого рівня визначаються з нижчих рівнів. Сім'я методів глибокого навчання стає дедалі багатшою, охоплюючи нейронні мережі, ієрархічні ймовірнісні моделі та різноманітні алгоритми навчання без нагляду та нагляду.

Мережа глибоких переконань (DBN) – імовірнісні генеративні моделі, складені з безлічі шарів стохастичних, прихованих змінних. Два верхні шари мають неорієнтовані симетричні зв'язки між собою. Нижні шари отримують зверху вниз, спрямовані на з'єднання із шаром, що знаходиться вище.

Машина Больцмана (BM) – мережа симетрично зв'язаних нейроноподібних одиниць, які приймають стохастичні рішення щодо того, бути чи ні.

Обмежена машина Больцмана (RBM) – спеціальний тип BM, що складається з шару видимих одиниць та шару прихованих одиниць без видимо-видимих або прихованих-прихованих з'єднань.

Глибока нейронна мережа (DNN) – багатшаровий персептрон з безліччю прихованих шарів, ваги яких повністю зв'язані і часто (хоча і не завжди) ініціалізуються, використовуючи або некеровану, або контрольовану техніку попередньої підготовки. (У літературі до 2012 року DBN часто використовували неправильно, щоб позначити DNN.)

Глибокий автокодер – «дискримінаційний» DNN, вихідними цілями якого є самі дані, а не мітки класів; отже, модель без нагляду. Коли тренується за критерієм деноїзування, глибокий автокодер також є генеративною

моделлю і з нього можна взяти проби.

Розподілене представництво – внутрішнє представлення даних, що спостерігаються, таким чином, що вони моделюються, як це пояснюється взаємодією багатьох прихованих факторів. Певний фактор, засвоєний з конфігурацій інших факторів, часто може добре узагальнити нові конфігурації. Розподілені уявлення, природно, відбуваються у нейромережі «коннекціонізму», де поняття представлене зразком діяльності в ряді одиниць і де одночасно одиниця, як правило, вносить свій внесок у багато концепцій. Однією з ключових переваг такої кореспонденції багато-до-багатьох є те, що вони забезпечують надійність у відображенні внутрішньої структури даних з точки зору витонченої деградації та стійкості до пошкоджень. Інша ключова перевага полягає в тому, що вони полегшують узагальнення понять та відносин, тим самим забезпечуючи здатність міркувати.

1.3.2 Тристороння категоризація

Як було описано раніше, глибоке навчання відноситься до досить широкого класу технік машинного навчання та архітектур, з ознакою використання багатьох шарів нелінійної обробки інформації, що мають ієрархічну природу. Залежно від того, як архітектури та методики призначені для використання, наприклад, синтез/генерування або розпізнавання/класифікація, можна широко класифікувати більшість робіт у цій галузі на три основні класи:

1. Глибокі мережі для неконтрольованого або генеративного навчання, які призначені для фіксації кореляції високих порядків спостережуваних або видимих даних для цілей аналізу або синтезу структури, коли інформація про мітки цільових класів відсутня. При використанні в генеративному режимі може також бути призначена для характеристики спільних статистичних розподілів видимих даних та пов'язаних з ними класів, коли вони доступні та обробляються як частина видимих даних. В останньому випадку використання

правила Байєса може перетворити цей тип генеративних мереж на дискримінаційні для навчання.

2. Глибокі мережі для контрольованого навчання, які покликані безпосередньо забезпечити дискримінаційну силу для цілей класифікації зразків, часто характеризуючи заданий розподіл класів, обумовлений видимими даними. Дані цільової мітки завжди доступні у прямій або непрямій формі для такого навчання під контролем. Їх також називають дискримінаційними глибинними мережами.

3. Гібридні глибинні мережі, де метою є дискримінація, якій часто, суттєво, допомагають результати генеральних або неконтрольованих глибинних мереж. Цього можна досягти шляхом кращої оптимізації або/і регуляризації глибинних мереж категорії (2). Ціль також може бути досягнута, коли для оцінки параметрів у будь-якій із глибинних генеративних або неконтрольованих глибинних мереж у категорії (1) вище використовуються дискримінаційні критерії для контрольованого навчання.

Зверніть увагу, що використання «гібриду» у (3) відрізняється від того, що використовується іноді в літературі, що стосується гібридних систем розпізнавання мови, що подають вихідні ймовірності нейронної мережі в НММ [7].

За загальноприйнятою традицією машинного навчання може бути природно просто класифікувати методи глибокого навчання на глибокі дискримінаційні моделі (наприклад, глибокі нейронні мережі або DNN, періодичні нейронні мережі або RNN, згорткові нейронні мережі або CNN тощо) та генеративні/неконтрольовані моделі (наприклад, обмежена машина Больцмана або RBM, мережі глибоких переконань або DBN, глибокі машини Больцмана (DBM), регульовані автокодери тощо). Однак ця двостороння схема класифікації втрачає ключове розуміння, отримане в ході досліджень глибокого навчання, про те, як генеративні моделі або моделі без нагляду можуть значно покращити підготовку DNN та інших моделей глибокого дискримінаційного або контрольованого навчання завдяки кращій

систематизації або оптимізації. Крім того, глибокі мережі для неконтрольованого навчання не обов'язково повинні мати ймовірнісний характер або мати можливість суттєво брати вибірки з моделі (наприклад, традиційні автокодери, мережі розрідженого кодування тощо). Більш пізні дослідження узагальнили традиційні аутокодери, що відмовляють, таким чином, щоб з них можна було ефективно взяти вибірку, і таким чином вони стали генеративними моделями [6]. Тим не менше, традиційна двостороння класифікація справді вказує на декілька ключових відмінностей між глибокими мережами для некерованого та контрольованого навчання. Порівняно з цими двома, глибокі моделі контрольованого навчання, такі як DNN, як правило, більш ефективні для навчання та тестування, більш гнучкі для побудови та більш придатні для наскрізного навчання складних систем (наприклад, відсутність приблизного висновку та навчання (наприклад, поширення петельних переконань). З іншого боку, моделі глибокого безконтрольного навчання, особливо імовірнісні узагальнюючі, легше інтерпретувати, легше вбудовувати знання домену, легше складати і легше обробляти невизначеність, але вони типово нерозв'язні у висновках та навчанні для складних систем. Ці відмінності зберігаються також у запропонованій тристоронній класифікації.

1.3.3 Глибокі мережі для безконтрольного або генеративного навчання

Навчання без нагляду означає відсутність використання інформації про нагляд за конкретними завданнями (наприклад, ярлики цільових класів) у процесі навчання. Багато глибоких мереж у цій категорії можуть бути використані для змістовного генерування зразків шляхом вибірки з мереж, на прикладах RBM, DBN, DBM та узагальнених автоенкодерів, що відмовляють [9], і, отже, є генеративними моделями. Однак деякі мережі в цій категорії не можуть бути легко відібрані на прикладах мереж розрідженого кодування та оригінальних форм глибоких автокодерів, тому вони не є генеративними за

своєю суттю.

Серед різних підкласів генеративних або неконтрольованих глибинних мереж найбільш поширені глибинні моделі на основі енергії [10]. Типовим прикладом цієї безконтрольної категорії моделі є оригінальна форма глибокого автокодера. Більшість інших форм глибоких автокодерів також не мають нагледу за своєю природою, але мають досить різні властивості та реалізації. Прикладами є трансформатори автокодерів, прогностичні розріджені кодери та їх складена версія, а також шумозаглушувальні автокодери та їх версії з накопиченням.

Зокрема, у безшумових автокодерах вхідні вектори спочатку пошкоджуються, наприклад, випадковим вибором відсотка входів і встановленням їх у нулі або додаванням до них гауссового шуму. Потім параметри коригуються для прихованих вузлів кодування для реконструкції вихідних, непошкоджених вхідних даних, використовуючи такі критерії, як середньоквадратична помилка реконструкції та розбіжність KL між початковими входами та реконструйованими входами. Кодовані подання, перетворені з непошкоджених даних, використовуються як вхідні дані на наступний рівень накопиченого безшумового автокодера.

Іншим відомим типом глибоких неконтрольованих моделей із узагальнюючими можливостями є глибока машина Больцмана або DBM. DBM містить багато шарів прихованих змінних і не має зв'язків між змінними всередині того самого шару. Це окремий випадок загальної машини Больцмана (BM), яка являє собою мережу симетрично з'єднаних блоків, які вмикаються або вимикаються на основі стохастичного механізму. Маючи простий алгоритм навчання, загальні BM дуже складні для вивчення та дуже повільні у роботі. У DBM кожен рівень фіксує складні кореляції вищого порядку між діями прихованих функцій у шарі нижче. DBM мають потенціал для вивчення внутрішніх уявлень, які стають дедалі складнішими, дуже бажаними для вирішення проблем розпізнавання об'єктів та мовлення. Крім того, представлення високого рівня можуть бути побудовані на основі великої

кількості невпевнених сенсорних входів, а потім можна використовувати дуже обмежені розмічені дані, щоб лише трохи тонко налаштувати модель під конкретне завдання.

Коли кількість прихованих шарів DBM зменшується до одного, ми обмежили машину Больцмана (RBM). Як і DBM, у RBM немає прихованих до прихованих та видимих до видимих з'єднань. Основна перевага RBM полягає в тому, що шляхом складання багатьох RBM можна ефективно вивчити багато прихованих шарів, використовуючи активації функцій одного RBM як навчальні дані для наступного. Така композиція призводить до створення мережі глибоких переконань (DBN).

Стандартний DBN поширився на факторну машину Больцмана вищого порядку в нижньому шарі, що дало значні результати для розпізнавання телефонів та для комп'ютерного зору. Ця модель, яка називається середньою коваріацією RBM або mcRBM, визнає обмеження стандартної RBM у здатності представляти структуру коваріації даних. Однак важко навчити mcRBM і використовувати їх на вищих рівнях глибокої архітектури. Крім того, опубліковані вагомі результати непросто відтворити. В архітектурі, описаній у [11], параметри mcRBM у повному DBN не налаштовуються з використанням дискримінаційної інформації, яка використовується для тонкої настройки вищих шарів RBM, через високі обчислювальні витрати. Подальша робота показала, що коли використовуються функції, адаптовані для динаміків, які усувають більшу мінливість функцій, mcRBM не був корисним.

Іншою репрезентативною глибокою генеративною мережею, яка може бути використана для неконтрольованого (а також контрольованого) навчання, є мережа сумарних продуктів або SPN. SPN – це спрямований ациклічний графік із спостережуваними змінними, а операції із сумою та продуктом – як внутрішні вузли в глибокій мережі. Вузли «суми» дають суміш моделі, а вузли продукту створюють ієрархію функцій. Властивості «повноти» та «послідовності» стримують SPN бажаним способом. Навчання SPN здійснюється за допомогою алгоритму ЕМ разом із зворотним

розповсюдженням. Процедура навчання починається з щільного SPN. Потім він знаходить структуру SPN, вивчаючи її ваги, де нульові ваги означають видалені з'єднання. Основна проблема у вивченні SPN полягає в тому, що навчальний сигнал (тобто градієнт) швидко розріджується, коли він поширюється на глибокі шари. Було знайдено емпіричні рішення для пом'якшення цих труднощів. Незважаючи на безліч бажаних генеративних властивостей в SPN, важко точно налаштувати параметри, використовуючи дискримінаційну інформацію, обмежуючи її ефективність у завданнях класифікації [17]. Однак ця складність була подолана в наступній роботі, про яку повідомляється в [13], де був представлений ефективний алгоритм дискримінаційного навчання у стилі BP для SPN. Важливо, що стандартний градієнтний спуск, заснований на похідній умовної ймовірності, страждає від тієї самої проблеми дифузії градієнта, добре відомої в регулярних DNN. Фокус для полегшення цієї проблеми при вивченні SPN полягає у тому, щоб замінити граничний висновок найбільш вірогідним станом прихованих змінних та розповсюдити градієнти лише за допомогою цього «жорсткого» вирівнювання. Відмінні результати щодо малих завдань розпізнавання зображень повідомили Генс та Домінго [13].

Повторювані нейронні мережі (RNN) можна розглядати як інший клас глибинних мереж для неконтрольованого (а також контрольованого) навчання, де глибина може бути такою великою, як довжина послідовності вхідних даних. У режимі безконтрольного навчання RNN використовується для прогнозування послідовності даних у майбутньому, використовуючи попередні зразки даних, і жодна додаткова інформація про клас не використовується для навчання. RNN є дуже потужним для моделювання даних послідовності (наприклад, мови чи тексту), але до недавнього часу вони широко не використовувались, частково тому, що їм важко навчитися захоплювати довгострокові залежності, що породжує проблеми зникнення градієнта або вибуху градієнта, які були відомі на початку 1990-х. Тепер із цими проблемами можна впоратися легше. Недавні успіхи в оптимізації

Hessian-free також частково подолали цю проблему, використовуючи наближену інформацію другого порядку або оцінки стохастичної кривизни. У більш пізній роботі [15], RNNs які пройшли навчання з використанням оптимізації Hessian-free, використовуються як генеративна глибинна мережа в задачах моделювання мови на рівні символів, де введені керовані зв'язки, що дозволяють поточним вхідним символам передбачити перехід від одного латентного вектора стану до наступного. Продемонстровано, що такі генеративні моделі RNN здатні генерувати послідовні символи тексту. Зовсім недавно Бенджо і Суцкевер досліджували варіації алгоритмів оптимізації стохастичного градієнтного спуску при навчанні генеративних RNN і показали, що ці алгоритми можуть перевершити безгесівські методи оптимізації. Міколов та ін. повідомили про чудові результати щодо використання RNN для мовного моделювання. Зовсім недавно Месніл та Яо та повідомив про успіх RNN у розумінні розмовної мови.

Існує довга історія досліджень розпізнавання мовлення, де механізми виробництва людської мови використовуються для побудови динамічної та глибокої структури в імовірнісних генеративних моделях; для всебічного огляду. Зокрема, ранні роботи, описані в [12], узагальнили та розширили загальноприйнятту неглибоку та умовно незалежну структуру НММ шляхом накладення динамічних обмежень у вигляді поліноміальної траєкторії на НММ параметри. Варіант цього підходу був нещодавно розроблений з використанням різних методів навчання для змінних у часі параметрів НММ та з додатками, розширеними до надійності розпізнавання мови [21]. Подібні траєкторії НММ також складають основу для параметричного синтезу мови. Подальша робота додала новий прихований шар до динамічної моделі, щоб чітко врахувати цільові, артикуляційно-властиві властивості в генерації людської мови. Більш ефективна реалізація цієї глибокої архітектури з прихованою динамікою досягається за допомогою нерекурсивних або кінцевих імпульсних характеристик (FIR) фільтрів. Наведені вище глибокоструктуровані генеративні моделі мовлення можуть бути показані як

особливі випадки більш загальної динамічної моделі мережі та ще більш загальної динамічної графічної моделі.

Графічні моделі можуть містити безліч прихованих шарів для характеристики складних взаємозв'язків між змінними у генерації мовлення. Озброєний потужною графікою артикуляційні властивості у поколінні мовлення людини. Більш ефективна реалізація цієї глибокої архітектури з прихованою динамікою досягається за допомогою нерекурсивних або кінцевих імпульсних характеристик (FIR) фільтрів. Наведені вище глибокоструктуровані генеративні моделі мовлення можуть бути показані як особливі випадки більш загальної динамічної моделі мережі та ще більш загальної динамічної графічної моделі. Озброєний потужною графікою інструмент моделювання, глибока архітектура мовлення нещодавно успішно застосовується для вирішення дуже складної проблеми одноканального багатомовного розпізнавання мови, де змішана мова є видимою змінною, тоді як незмішана мова стає представленою в новий прихований шар у глибинній генеративній архітектурі. Глибокі генеративні графічні моделі справді є потужним інструментом у багатьох додатках завдяки своїй здатності вбудовувати знання про домен. Однак вони часто використовуються з невідповідними наближеннями у висновках, навчанні, прогнозуванні та дизайні топології, що все впливає із властивої нерозв'язності цих завдань для більшості реальних програм. Ця проблема була розглянута в останніх роботах Стоянова, що забезпечує цікавий напрямок для створення глибоких генеративних графічних моделей, які можуть бути більш корисними на практиці в майбутньому. Ще більш радикальний спосіб боротьби з цією нерозбірливістю був запропонований Бенджо [9], де потреби у маргіналізації прихованих змінних взагалі уникають.

Стандартні статистичні методи, що використовуються для широкомасштабного розпізнавання та розуміння мови, поєднують (неглибокі) приховані марківські моделі для мовної акустики з вищими рівнями структури, що представляють різні рівні ієрархії природної мови. Цю

комбіновану ієрархічну модель можна належним чином розглядати як глибоку генеративну архітектуру. Ці ранні глибинні моделі були сформульовані як спрямовані графічні моделі, відсутній ключовий аспект «розподіленого представлення», втілений у новіших глибинних генеративних мережах DBN та DBM.

Динамічні або тимчасово рекурсивні генеративні моделі, засновані на архітектурах нейронних мереж, для моделювання руху людини та в [20] для розбору природної мови та природної сцени. Остання модель особливо цікава, оскільки алгоритми навчання здатні автоматично визначати оптимальну структуру моделі. Це контрастує з іншими глибокими архітектурами, такими як DBN де вивчаються лише параметри, тоді як архітектури потрібно попередньо визначити. Зокрема, як повідомляється в [19], рекурсивна структура, яка зазвичай зустрічається на зображеннях природних сцен та в реченнях природною мовою, може бути виявлена за допомогою архітектури передбачення структури максимального поля. Показано, що одиниці, що містяться на зображеннях чи сентенціях, ідентифікуються, а також визначається спосіб взаємодії цих одиниць між собою, утворюючи ціле.

1.3.4 Глибокі мережі для контрольованого навчання

Багато дискримінаційних методів для контрольованого навчання в обробці сигналів та інформації є неглибокими архітектурами, такими як НММ та умовні випадкові поля (CRF). CRF – це суттєво дискримінаційна архітектура, яка характеризується лінійним співвідношенням між вхідними ознаками та перехідними ознаками. Неглибокий характер CRF найбільш чітко проявляється через еквівалентність, встановлену між CRF та дискримінаційно підготовленими моделями Гауса та НММ. Зовсім недавно глибокоструктуровані CRF були розроблені шляхом розміщення вихідних даних у кожному нижчому шарі CRF разом з вихідними вхідними даними на його вищий рівень [21]. Різні версії глибоко структурованих CRF успішно

застосовуються до розпізнавання телефонів, ідентифікація розмовної мови та обробка природної мови. Однак, принаймні для завдання розпізнавання телефону, продуктивність глибокоструктурованих CRF, які є суто дискримінаційними (негенеративними), не змогла зрівнятися з гібридним підходом за участю DBN.

Морган дає чудовий огляд інших основних існуючих дискримінаційних моделей розпізнавання мовлення, заснованих головним чином на традиційній нейронній мережі або архітектурі MLP, використовуючи навчання зворотного розповсюдження із випадковою ініціалізацією. Це аргументує важливість як збільшення ширини кожного шару нейронних мереж, так і збільшення глибини. Зокрема, клас моделей глибоких нейронних мереж лежить в основі популярного «тандемного» підходу, де результат розвитку дискримінаційно засвоєної нейронної мережі розглядається як частина змінної спостереження в НММ.

Повторювані нейронні мережі (RNN) використовувались як генеративна модель. RNN також можуть бути використані як дискримінаційна модель, де на виході є послідовність міток, пов'язана з послідовністю вхідних даних. Зверніть увагу, що такі дискримінаційні RNN або моделі послідовностей застосовувались до мови давно з обмеженим успіхом. У [7] НММ навчався спільно з нейронними мережами, з дискримінаційним імовірнісним критерієм навчання. У [18] окремий НММ був використаний для сегментації послідовності під час тренування, а НММ також використовувався для перетворення результатів класифікації RNN у послідовності міток. Однак використання НММ для цих цілей не використовує весь потенціал RNN.

Нещодавно було запропоновано набір нових моделей та методів, які дозволяють самим RNN виконувати класифікацію послідовностей, одночасно вбудовуючи в модель довгострокову короткочасну пам'ять, усуваючи необхідність попереднього сегментування дані навчання та для подальшої обробки результатів. В основі цього методу лежить ідея інтерпретації виходів RNN як умовного розподілу по всіх можливих послідовностях міток з

урахуванням вхідних послідовностей. Потім може бути виведена диференційована цільова функція для оптимізації цих умовних розподілів за правильними послідовностями міток, де сегментація даних виконується автоматично алгоритмом. Ефективність цього методу була продемонстрована в завданнях розпізнавання рукописного вводу та в невеликому мовленнєвому завданні

Іншим типом дискримінаційної глибинної архітектури є згорткова нейронна мережа (CNN), в якій кожен модуль складається із згорткового шару та шару об'єднання. Ці модулі часто складаються один на інший або DNN поверх нього, щоб сформувати глибоку модель. Згорнутий шар поділяє багато ваг, а шар об'єднання подає вибірки вихідних даних згорткового шару і зменшує швидкість передачі даних із нижчого шару. Спільне використання ваги в згортковому шарі, разом із відповідним чином обраними схемами об'єднання, надає CNN деяких властивостей «незмінності» (наприклад, незмінність трансляції). Стверджувалося, що така обмежена «незмінність» чи еквівалентність не є адекватною для складних завдань розпізнавання зразків, і можуть знадобитися більш принципові способи обробки більш широкого діапазону незмінності. Тим не менше, CNN були визнані високоефективними та широко використовувались у комп'ютерному зорі та розпізнаванні зображень. Зовсім недавно, із відповідними змінами від CNN, призначеними для аналізу зображень, до тих, що враховують специфічні властивості мовлення, CNN також вважається ефективним для розпізнавання мови.

Корисно зазначити, що нейронна мережа із затримкою часу (TDNN) [15], розроблена для раннього розпізнавання мови, є особливим випадком і попередником CNN, коли розподіл ваги обмежений одним із двох вимірів, тобто часом розміру, а шару об'єднання немає. Лише недавно дослідники виявили, що інваріантність часового виміру менш важлива, ніж інваріантність частотно-вимірювального рівня для розпізнавання мови. Ретельний аналіз основних причин описаний в [12], а також нова стратегія для проектування рівня об'єднання CNN, продемонстрована як більш ефективна, ніж усі

попередні CNN у розпізнаванні телефонів.

Також корисно зазначити, що модель ієрархічної тимчасової пам'яті (HTM) є ще одним варіантом і розширенням CNN. Розширення включає наступні аспекти:

- 1) вводиться часовий або часовий вимір, який служить інформацією про «нагляд» щодо дискримінації (навіть для статичних зображень);
- 2) використовуються як інформаційні потоки знизу вгору, так і зверху вниз, а не лише знизу вгору в CNN;
- 3) байєсівський імовірнісний формалізм використовується для злиття інформації та для прийняття рішень.

Навчальна архітектура, розроблена для розпізнавання мовлення знизу вгору, заснована на виявленні з використанням методики DBN – DNN, також може бути віднесена до категорії дискримінаційних або категорія глибокої архітектури під наглядом. У цій архітектурі немає наміру та механізму характеризувати спільну ймовірність даних та розпізнавання цілей мовних атрибутів та телефону та слів вищого рівня. Найновіша реалізація цього підходу базується на DNN, або нейронних мережах з багатьма рівнями, що використовують навчання зворотного розповсюдження. Один проміжний рівень нейронної мережі у реалізації цієї основи виявлення явно представляє мовні атрибути, які є спрощеними сутностями з «атомних» одиниць мови. Спрощення полягає у видаленні тимчасово перекриваючих властивостей мовних атрибутів або подібних до артикуляції рис. Очікується, що вбудовування таких більш реалістичних властивостей у майбутню роботу покращить точність розпізнавання мови.

1.3.5 Гібридні глибинні мережі

Термін «гібрид» для цієї третьої категорії відноситься до глибокої архітектури, яка або включає, або використовує як генеративні, так і дискримінаційні компоненти моделі. У існуючих гібридних архітектурах

генеративний компонент здебільшого використовується для допомоги при дискримінації, що є кінцевою метою гібридної архітектури. Як і чому генеративне моделювання може допомогти у дискримінації, можна дослідити з двох точок зору:

- точка зору оптимізації, де генеративні моделі, які навчаються без нагляду, можуть забезпечити чудові точки ініціалізації в дуже нелінійних задачах оцінки параметрів (з цієї причини введено загальновживаний термін «попередня підготовка» у процесі глибокого навчання);
- перспектива регуляризації, де моделі без нагляду під дією можуть ефективно надавати пріоритет набору функцій, представлених моделлю.

Дослідження [13], забезпечило глибокий аналіз та експериментальні докази, що підтверджують обидві вищезазначені точки зору.

DBN, генеративну глибинну мережу для неконтрольованого навчання, про яку йдеться в п.1.3.3, можна перетворити та використовувати як початкову модель DNN для контрольованого навчання з тією ж структурою мережі, яка в подальшому піддається дискримінаційній підготовці або доопрацюванню надані цільові мітки. Коли DBN використовується таким чином, ми розглядаємо цю модель DBN–DNN як гібридну глибинну модель, де модель, навчена з використанням неконтрольованих даних, допомагає зробити дискримінаційну модель ефективною для контрольованого навчання.

Інший приклад гібридної глибинної мережі розроблений, де ваги DNN також ініціалізуються з генеративного DBN, але додатково налаштовуються за допомогою дискримінаційного критерію рівня послідовності, що є умовною ймовірністю послідовності міток з урахуванням вхідної послідовності ознак, замість загальновживаного критерію перехресної ентропії на рівні кадру. Це можна розглядати як поєднання статичного DNN із неглибокою дискримінаційною архітектурою CRF. Можна показати, що такий DNN–CRF еквівалентний гібридній глибокій архітектурі DNN та HMM, параметри яких вивчаються спільно, використовуючи критерій максимальної взаємної інформації (MMI) повної послідовності між усією послідовністю міток та

вхідною послідовністю ознак . Тісно пов'язаний метод навчання в повній послідовності, розроблений і впроваджений для набагато більших завдань, проводиться нещодавно з успіхом для неглибокої нейронної мережі і для глибокої.

Варто вказати на зв'язок між вищезазначеною стратегією попередньої підготовки/тонкої настройки, по'язаної з гібридними глибокими мережами, та дуже популярною технікою навчання мінімальних телефонних помилок (MPE) для НММ. Щоб ефективність навчання MPE була ефективною, параметри потрібно ініціалізувати, використовуючи алгоритм (наприклад, алгоритм Баума-Велча), який оптимізує генеративний критерій (наприклад, максимальна ймовірність). Цей тип методів, що використовує підготовлені параметри з максимальною ймовірністю, щоб допомогти у дискримінаційному навчанні НММ, можна розглядати як «гібридний» підхід до навчання неглибокої моделі НММ.

Поряд із використанням дискримінаційних критеріїв для навчання параметрів у генеративних моделях, як у наведеному вище прикладі навчання НММ, ми тут обговорюємо той самий метод, який застосовується для вивчення інших гібридних глибинних мереж. Генеративна модель RBM вивчається з використанням дискримінаційного критерію ймовірностей заднього позначення класу. Тут вектор мітки поєднується з вектором вхідних даних, щоб сформувати комбінований видимий шар у RBM. Таким чином, RBM може служити самостійним рішенням проблем класифікації, і автори вивели дискримінаційний алгоритм навчання RBM як неглибоку генеративну модель. У пізнішій роботі Ранзато глибока генеративна модель DBN із керованим випадковим полем Маркова (MRF) на найнижчому рівні вивчається для вилучення ознак, а потім для розпізнавання складних класів зображень, включаючи оклюзії. Генеративна здатність DBN сприяє виявленню того, яка інформація фіксується, а що втрачається на кожному рівні представлення в глибинній моделі. Відповідне дослідження щодо використання дискримінаційного критерію емпіричного ризику для підготовки глибоких

графічних моделей

Подальшим прикладом гібридних глибинних мереж є використання генеративних моделей DBN для попередньої підготовки глибоких згорткових нейронних мереж. Як і повністю підключений DNN, про який говорили раніше, попередня підготовка також допомагає поліпшити продуктивність глибоких CNN за випадкової ініціалізації. Попередня підготовка DNN або CNN з використанням набору регуляризованих глибоких автокодерів [9], включаючи автоматичні кодектори, що скорочують звук, скорочувальні автокодери та розріджені автокодери, також є подібним прикладом категорії гібридних глибинних мереж.

Останній приклад, наведений тут для гібридних глибинних мереж, базується на ідеї та роботі [15], де одне із завдань дискримінації (наприклад, розпізнавання мови) видає результат (текст), який служить вхідним матеріалом для другого завдання дискримінації (наприклад, машинного перекладу). Загальна система, що надає функціональність мовного перекладу – перекладу мови однією мовою на текст іншою мовою, – це двоступенева глибока архітектура, що складається з генеративних та дискримінаційних елементів. Обидві моделі розпізнавання мови (наприклад, НММ) та машинного перекладу (наприклад, фразове відображення та немонотонне вирівнювання) носять генеративний характер, але всі їх параметри вивчаються для дискримінації остаточного перекладеного тексту з урахуванням мовних даних. Структура, описана в [13], дозволяє наскрізну оптимізацію продуктивності загальної глибинної архітектури, використовуючи уніфіковану структуру навчання. Цей гібридний підхід глибокого навчання може бути застосований не лише до перекладу мови, а й до всіх мовленнєво-орієнтованих та, можливо, інших завдань з обробки інформації, таких як пошук мовної інформації, розуміння мови, розуміння та пошук міжмовної мови/ тексту тощо.

1.4 Моделювання та обробка природної мови

Дослідження в галузі обробки мови, документів та тексту останнім часом набувають все більшої популярності у співтоваристві з обробки сигналів, і Технічний комітет з обробки мовлення та обробки мов IEEE визначив однією з основних сфер уваги. Застосування глибокого навчання в цій галузі розпочалося з мовного моделювання (LM), де метою є надати ймовірність будь-якій довільній послідовності слів чи інших мовних символів (наприклад, букв, символів, телефонів тощо). Обробка природної мови (NLP) або обчислювальна лінгвістика також має справу з послідовностями слів чи інших лінгвістичних символів, але завдання набагато різноманітніші (наприклад, переклад, синтаксичний розбір, класифікація тексту тощо), не зосереджуючись на забезпеченні ймовірностей мовних символів. Зв'язок полягає в тому, що LM часто є важливим і дуже корисним компонентом систем NLP. На сьогодні заявки на NLP є однією з найактивніших сфер досліджень глибокого навчання, і глибоке навчання також вважається одним із перспективних напрямків дослідницької спільноти NLP. Однак перетин між дослідниками глибокого навчання та NLP поки що не настільки великий, як у сферах застосування мови чи зору. Це частково тому, що вагомі докази переваги глибокого вивчення сучасних методів NLP не було настільки сильним, як розпізнавання мовлення чи візуального об'єкта.

1.4.1 Мовне моделювання

Мовні моделі (LM) є найважливішою частиною багатьох успішних застосувань, таких як розпізнавання мови, пошук текстової інформації, статистичний машинний переклад та інші завдання NLP. Традиційні методи оцінки параметрів у LM засновані на підрахунку N-грамів. Незважаючи на відомі слабкі місця N-грами та величезні зусилля дослідницьких спільнот у багатьох сферах, N-грами залишались найсучаснішими, доки нейронні мережі

та методи, що базуються на глибокому навчанні, не суттєво знизили сум'яття LM, одного загального (але не остаточного) показника якості LM, за кількома стандартними завданнями базового рівня.

Перш ніж ми обговоримо LM на основі нейронних мереж, ми відзначимо використання ієрархічних байєсівських пріоритетів у побудові глибокої та рекурсивної структури для LM [16]. Зокрема, процес Пітмана-Йора експлуатується як байєсівський пріор, з якого будується глибока (чотири шари) імовірісна генеративна модель. Він пропонує принциповий підхід до згладжування LM, включаючи розподіл степенного закону для природної мови. Як обговорювалося в пункті 1.3, такий тип впровадження попередніх знань легше досягти у генеративному ймовірнісному моделюванні, ніж у дискримінаційній установці на основі нейронної мережі. Наведені результати щодо зменшення незрозумілості LM не настільки сильні, як ті, що досягнуті нейромережевими LM, про які ми поговоримо далі.

Існує довга історія використання (неглибоких) нейронних мереж прямого розповсюдження в LM, які називаються NNLM. Використання DNN однаково для LM з'явилося нещодавно. LM – це функція, яка фіксує помітні статистичні характеристики розподілу послідовностей слів у природній мові. Це дозволяє робити імовірнісні передбачення наступного слова, наведеного до попередніх. NNLM використовує здатність нейронної мережі засвоювати розподілені уявлення, щоб зменшити вплив розмірності. Оригінальний NNLM зі структурою нейронних мереж, що подається вперед, працює наступним чином: вхід N -грам NNLM сформований з використанням історії фіксованої довжини $N-1$ слово. Кожне з попередніх $N-1$ слів кодується за допомогою дуже рідкісного кодування $1-3-V$, де V – розмір словникового запасу. Потім ця ортогональна репрезентація слів на $1-3-V$ проектується лінійно на нижній вимірний простір, використовуючи матрицю проекції, спільну для слів на різних позиціях в історії. Цей тип розподіленого подання слів у безперервному просторі називається «вбудовуванням слів», що дуже відрізняється від загальної символічної або локалістичної презентації [9]. Після проекційного

шару використовується прихований шар з нелінійною функцією активації, який є або гіперболічною дотичною, або логістичною сигмоїдою. Потім вихідний рівень нейронної мережі слідує за прихованим шаром, при цьому кількість вихідних одиниць дорівнює розміру повного словникового запасу. Після навчання мережі активації вихідного рівня представляють «N -грам» розподіл ймовірностей LM.

Основна перевага NNLM над традиційним підрахунком N-грам LM полягає в тому, що історія більше не розглядається як точна послідовність $N-1$ слова, а скоріше як проекція всієї історії на якийсь нижчий мірний простір. Це призводить до зменшення загальної кількості параметрів у моделі, які необхідно тренувати, що призводить до автоматичної кластеризації подібних історій. Порівняно з класом N-грами LM, NNLM відрізняються тим, що проектують усі слова в один і той же низьковимірний простір, в якому між різними словами може бути багато ступенів подібності. З іншого боку, NNLM мають набагато більшу обчислювальну складність, ніж N-грам LM.

Давайте знову подивимося на сильні сторони NNLM з точки зору розподілених представлень. Розподілене представлення символу – це вектор ознак, що характеризують значення символу. Кожен елемент у векторі бере участь у поданні значення. За допомогою NNLM покладається на алгоритм навчання, щоб розкрити значущі, безперервні функції. Основна ідея полягає в тому, щоб навчитися пов'язувати кожне слово у словнику з безперервним значенням векторного подання, яке в літературі називається вбудованим словом, де кожне слово відповідає точці в просторі ознак. Можна уявити, що кожен вимір цього простору відповідає семантичній чи граматичній характеристиці слів. Надія в тому, що функціонально подібні слова стають ближчими один до одного в цьому просторі, принаймні за деякими напрямками. Таким чином, послідовність слів може бути перетворена в послідовність цих вивчених векторів ознак. Нейронна мережа вчиться відображати цю послідовність векторів ознак до розподілу ймовірностей за наступним словом у послідовності. Підхід розподіленого представлення до

LM має ту перевагу, що дозволяє моделі добре узагальнювати послідовності, які не входять до набору послідовностей навчальних слів, але подібні за своїми ознаками, тобто розподіленим представленням. Оскільки нейронні мережі прагнуть зіставити сусідні входи із сусідніми виходами, передбачення, що відповідають послідовностям слів із подібними ознаками, відображаються на подібні передбачення.

Вищезазначені ідеї NNLM були впроваджені в різні дослідження, деякі з яких включають глибокі архітектури. Ідея структурувати ієрархічно вихідні дані NNLM для обробки великих словників була представлена в [7]. Для моделювання мови було використано часовий механізм управління ринком. На відміну від традиційної N-грамної моделі, факторизований RBM використовує розподілені уявлення не лише для контекстних слів, але й для слів, що передбачаються. Цей підхід узагальнено для більш глибоких структур, як повідомляється в [17]. Описує NNLM зі структурованим вихідним рівнем (SOUL–NNLM), де глибина обробки в LM фокусується на репрезентації вихідних даних нейронної мережі. Рисунок 1.5 ілюструє архітектуру SOUL–NNLM з ієрархічною структурою у вихідних шарах нейронної мережі, яка має однакову архітектуру зі звичайною NNLM аж до прихованого рівня. Ієрархічна структура вихідного словника мережі має вигляд дерева кластеризації, показаного праворуч від рис. 1.5, де кожне слово належить лише одному класу і закінчується одним листовим вузлом дерева. В результаті ієрархічної структури SOUL–NNLM дає можливість навчати NNLM з повним, дуже великим словниковим запасом. Це дає переваги перед традиційною NNLM, яка вимагає коротких списків слів для того, щоб здійснити ефективне обчислення під час навчання.

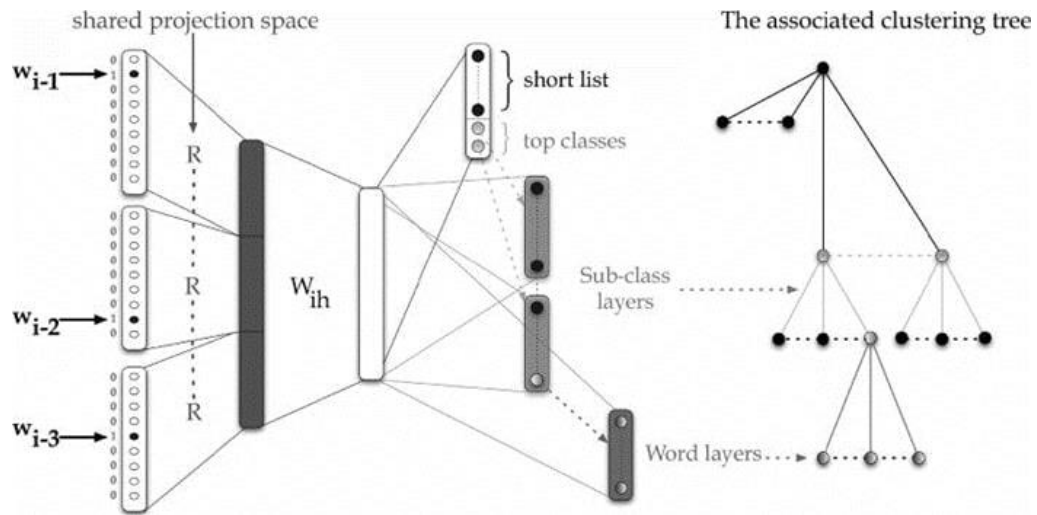


Рисунок 1.5 – Архітектура SOUL-NNLM з ієрархічною структурою у вихідних шарах нейронної мережі

Як ще один приклад неймережєвих LM, [15], використовує RNN для побудови широкомасштабної мови моделі, що називаються RNNLM. Основна різниця між зворотною передачею та періодичною архітектурою для LM – це різні способи представлення історії слів. Щодо NNLM, що передається вперед, історія все ще складається лише з кількох слів. Але для RNNLM ефективно подання історії вивчається з даних під час навчання. Прихований шар RNN представляє всю попередню історію, а не тільки $N - 1$ попередні слова, таким чином модель теоретично може представляти довгі контекстні моделі. Ще однією важливою перевагою RNNLM перед аналогом прямої передачі є можливість представляти більш вдосконалені шаблони в послідовності слів. Наприклад, шаблони, які покладаються на слова, які могли траплятися на різних місцях в історії, можуть бути кодовані набагато ефективніше за допомогою періодичної архітектури. Тобто RNNLM може просто запам'ятати якесь конкретне слово в стані прихованого шару, тоді як NNLM для перенаправлення потрібно буде використовувати параметри для кожного конкретного положення слова в історії.

RNNLM навчається з використанням алгоритму зворотного поширення у часі (рис. 1.6), щоб показати під час навчання, як RNN розгортається як мережа глибокого зворотного зв'язку (з трьома кроками в часі назад).

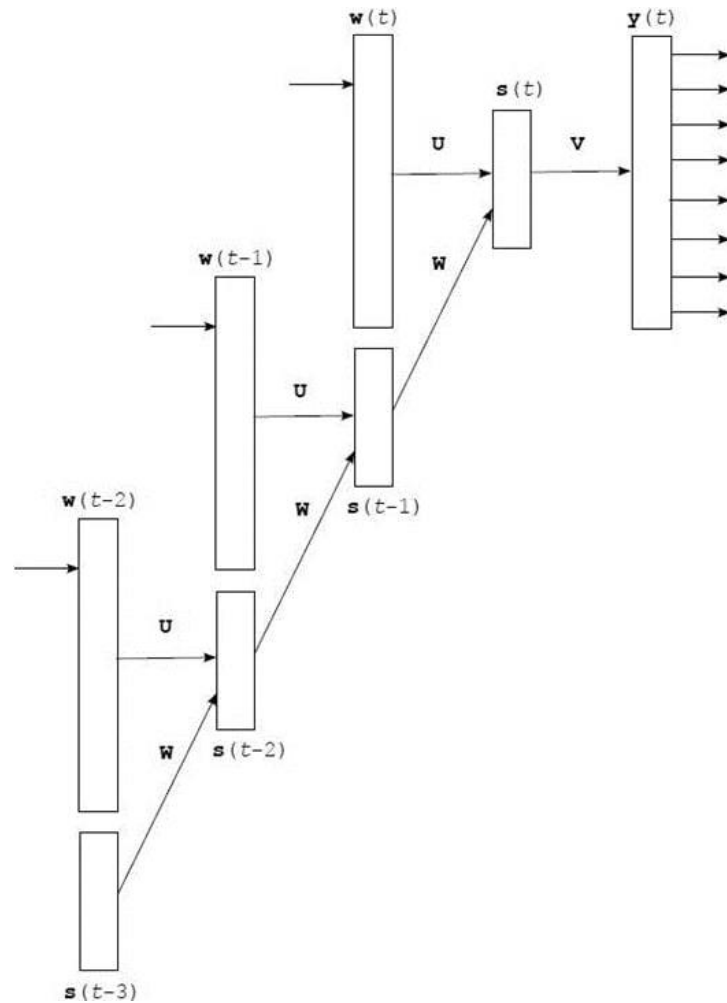


Рисунок 1.6 – Під час навчання RNNLM RNN розгортається в глибоку мережу прямого зворотного зв'язку

Тренування RNNLM досягає стабільності та швидкої конвергенції, що допомагає обмежувати зростаючий градієнт у навчанні RNN. Схеми адаптації для RNNLM також розробляються шляхом сортування навчальних даних щодо їх відповідності та навчання моделі під час обробки тестових даних. Емпіричні порівняння з іншими найсучаснішими підрахунками N-грамми LM демонструють набагато кращі показники RNNLM в мірі збентеження, як повідомляється в [17].

Окрему роботу із застосування RNN до LM з одиницею символів замість слів можна знайти в [14]. Продемонстровано багато цікавих властивостей, таких як прогнозування довгострокових залежностей (наприклад, відкриття та

закриття цитат у абзаці). Однак корисність символів замість слів як одиниць у практичному застосуванні незрозуміла, оскільки це слово є таким потужним поданням для природної мови. Зміна слів на символи в LM може обмежити більшість практичних сценаріїв застосування, і навчання стане складнішим. Наразі моделі рівня слова залишаються вищими.

У останніх роботах розробили швидкий і простий алгоритм навчання для NNLM. Незважаючи на свою чудову продуктивність, NNLM використовуються менш широко, ніж стандартно N-грами LM через набагато довший час навчання. Зазначений алгоритм використовує метод, який називається оцінкою контрастності шуму, або NCE, щоб досягти набагато швидшого навчання для NNLM, при цьому складність часу не залежить від розміру словника; отже, в NNLM використовується плоский, а не структурований по дереву вихідний рівень. Ідея NCE полягає у проведенні нелінійної логістичної регресії для розрізнення спостережуваних даних та деяких штучно створених шумів. Тобто, щоб оцінити параметри в моделі щільності спостережуваних даних, ми можемо навчитися розрізняти вибірки з розподілу даних та вибірки з відомого розподілу шуму. Як важливий особливий випадок, NCE є особливо привабливим для ненормованих розподілів (тобто вільних від функцій розділення в знаменнику). Для того, щоб ефективно застосовувати NCE для ефективного навчання NNLM, спочатку формулюють навчальну проблему як таку, яка приймає цільову функцію як розподіл слова з точки зору скорингової функції. Тоді NNLM можна розглядати як спосіб кількісної оцінки сумісності між історією слова та наступним словом-кандидатом за допомогою функції оцінки. Таким чином, цільова функція для навчання NNLM стає піднесенням функції підрахунку, нормованою на одну і ту ж константу для всіх можливих слів. Видаляючи дорогий коефіцієнт нормалізації, показано, що NCE прискорює навчання NNLM на порядок.

Подібна концепція NCE використовується в роботі [15], яка називається негативною вибіркою. Це застосовується до спрощеної версії NNLM, з метою

побудови вбудовування слів замість обчислення ймовірностей послідовностей слів. Вбудованість слів – важлива концепція для програм NLP.

1.4.2 Обробка природної мови

Машинне навчання протягом багатьох років є домінуючим інструментом NLP. Однак використання машинного навчання в NLP здебільшого обмежувалось чисельною оптимізацією ваг для поданих людиною подань та особливостей з текстових даних. Метою глибокого або репрезентативного навчання є автоматична розробка функцій або подань із вихідного текстового матеріалу, що підходять для широкого кола завдань NLP.

Нещодавно було показано, що методи глибокого навчання, засновані на нейронних мережах, добре виконують різні завдання NLP, такі як моделювання мови, машинний переклад, позначення частини мови, розпізнавання іменованих сутностей, аналіз настроїв та виявлення парафрази. Найбільш привабливим аспектом методів глибокого навчання є їх здатність виконувати ці завдання без зовнішніх ресурсів, розроблених вручну, або інтенсивної інженерії функцій. З цією метою глибоке навчання розробляє і використовує важливу концепцію під назвою «вбудовування», яка стосується представлення символічної інформації в тексті природної мови на рівні слова, рівня фрази і навіть рівня речень з точки зору безперервних векторів.

Ранні роботи, що висвітлювали важливість вбудовування слів, походять, хоча оригінальна форма походить як побічний продукт мовного моделювання. Сировинні символічні репрезентації слів трансформуються з розріджених векторів за допомогою кодування 1-з- V з дуже великим розміром (тобто розміром словника V або його квадратом або навіть його кубічним) у низькорозмірні, реальні значення векторів через нейронної мережі, а потім використовується для обробки наступними рівнями нейронної мережі. Ключовою перевагою використання неперервного простору для

представлення слів (або фраз) є його розподілений характер, що дозволяє спільно використовувати або групувати подання слів із подібним значенням. Такий розподіл неможливий в оригінальному символічному просторі, побудованому за допомогою 1-3-V кодування з дуже високим виміром, для представлення слів. Без нагляду навчання використовується там, де «контекст» слова використовується як навчальний сигнал у нейронних мережах. У роботі [18] нейронна мережа навчена виконувати вбудовування слів. Пізніші роботи пропонують нові способи вивчення вбудованих слів, які краще фіксують семантику слів шляхом включення як локального, так і глобального контекстів документів та кращого врахування омонімії та полісемії шляхом вивчення кількох вбудовувань на слово. Крім того, є вагомі докази того, що використання RNN може також забезпечити емпірично хорошу ефективність у вивченні вбудованих слів. Хоча використання NNLM, метою яких є передбачення майбутніх слів у контексті, також спонукає вбудовування слів як його побічний продукт, набагато простіші способи досягнення вбудовувань можливі без необхідності передбачення слів. Нейронні мережі, що використовуються для створення вбудованих слів, потребують набагато менших вихідних одиниць, ніж величезний розмір, як правило, необхідний для NNLM.

У тій же ранній статті про вбудовування слів Колоберт та Уестон розробили та застосували згорткову мережу як загальну модель для одночасного вирішення ряду класичних проблем, включаючи тегування часткової мови, фрагментування, позначення імен сутності, ідентифікацію семантичної ролі та подібну ідентифікацію слів. Пізніші роботи надалі розвивали швидкий, суто дискримінаційний підхід до синтаксичного аналізу, заснований на глибокій періодичній згортковій архітектурі надають вичерпний огляд способів застосування уніфікованих архітектур нейронних мереж та пов'язаних з ними алгоритмів глибокого навчання для вирішення проблем NLP «з нуля», що означає, що для вилучення особливостей не використовуються традиційні методи NLP. Тема цього напряму роботи –

уникати конкретних завдань, «Рукотворна» інженерія функцій, забезпечуючи при цьому універсальність та уніфіковані функції, побудовані автоматично з глибокого навчання, застосовного до всіх завдань з обробки природної мови. Системи, описані в [11], автоматично вивчають внутрішні репрезентації або вбудовування слів із величезної кількості в основному немаркованих навчальних даних під час виконання широкого кола завдань NLP.

Недавня робота Міколова отримує вбудовування слів шляхом спрощення NNLM, описаного пункті 1.4.1. Встановлено, що NNLM можна успішно навчати у два етапи. По-перше, безперервні вектори слів вивчаються за допомогою простої моделі, яка виключає нелінійність у верхньому шарі нейронної мережі та поділяють рівень проекції для всіх слів. А по-друге, N-грам NNLM навчається поверх векторів слів. Отже, після видалення другого кроку в NNLM проста модель використовується для вивчення вбудовування слів, де простота дозволяє використовувати дуже великий обсяг даних. Це породжує модель вбудовування слів, яка називається моделлю безперервної сумки слів (CBOW), яка показана на рис.1.7. Окрім того, оскільки мета перестав обчислювати ймовірності послідовностей слів, як у LM, система вбудовування слів тут робиться більш ефективною, не тільки передбачаючи поточне слово на основі контексту, але також здійснюючи зворотне передбачення, відоме як модель «Пропустити грам», що показано на рис.1.7.

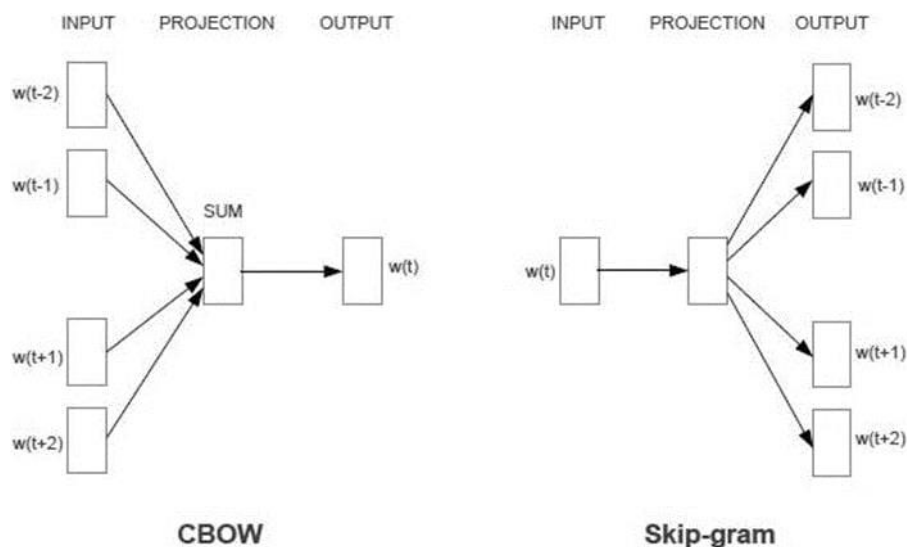


Рисунок 1.7 – Архітектура CBOW та Skip-gram

Паралельно із вищезазначеним розвитком, демонструють, що навчання NCE легких моделей вбудовування слів є високоефективним способом вивчення високоякісних подань слів, подібно до дещо ранніх легких LM. Отже, результати, які раніше вимагали дуже значної апаратної та програмної інфраструктури, тепер можуть бути отримані на одному робочому столі з мінімальним навантаженням на програмування та з використанням менше часу та даних. Ця остання робота також показує, що для навчання репрезентації лише п'ять зразків шуму в NCE можуть бути достатніми для отримання сильних результатів для вбудовування слів, набагато менших, ніж необхідних для LM. Автори також використовували «обернену модель мови» для обчислення вкладених слів, подібно до того, як модель Skip-gram використовується в [17].

Обмеження попередньої роботи над вбудовуванням слів тим, що ці моделі будувались лише з локальним контекстом і одним представленням на слово. Вони розширили моделі локального контексту до тієї, яка може включати глобальний контекст із повних речень або цілого документа. Ця розширена модель враховує омонімію та полісемію шляхом вивчення декількох вкладок для кожного слова. У попередній роботі тієї ж дослідницької групи, для побудови глибокої архітектури була розроблена рекурсивна нейронна мережа з локальним контекстом. Мережа, незважаючи на відсутність глобального контексту, вже виявилася здатною до успіху злиття слів природної мови на основі засвоєних семантичних перетворень їх оригінальних рис. Цей підхід до глибокого навчання забезпечив чудову ефективність розбору природної мови. Цей же підхід також продемонстрував, що він був досить успішним в аналізі природних зображень сцени. У відповідних дослідженнях подібна рекурсивна глибинна архітектура використовується для виявлення парафрази та для прогнозування розподілу настроїв з тексту.

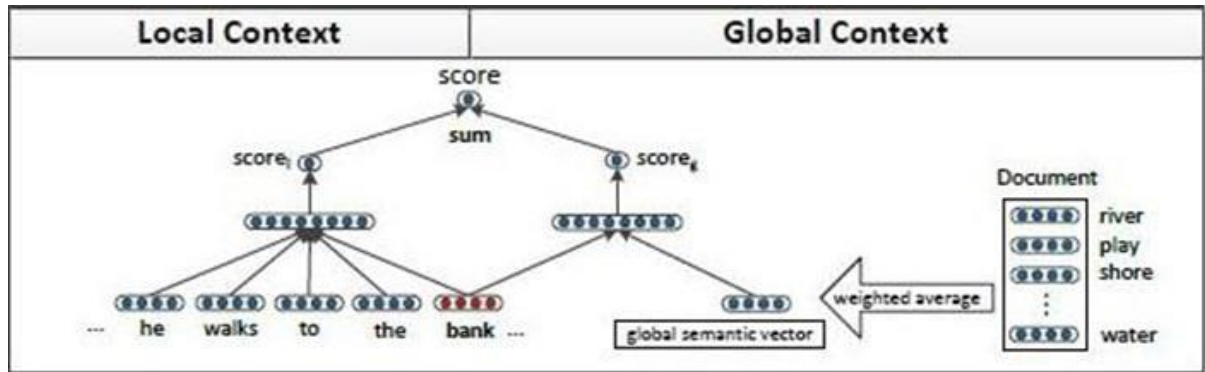


Рисунок 1.8 – Розширена модель вбудовування слів із використанням рекурсивної нейронної мережі, яка враховує не тільки локальний контекст, а й глобальний контекст. Глобальний контекст витягується з документа та розміщується у вигляді глобального семантичного вектора як частина вхідних даних до оригінальної моделі вбудовування слів із локальним контекстом

Зараз ми звернемося до вибраних застосувань методів глибокого навчання, включаючи використання архітектур нейронних мереж та вбудовування слів до практично корисних завдань NLP. Машинний переклад – одне з таких завдань, яке дослідники NLP виконують протягом багатьох років, засноване, як правило, на дрібних статистичних моделях. Роботи, описані в [19], є, мабуть, першим вичерпним звітом про успішне застосування нейромережевих мовних моделей із вбудованими словами, навчених на графічному процесорі, для великих задач машинного перекладу. Вони вирішують проблему великої складності обчислень і пропонують рішення, яке дозволяє навчити 500 мільйонів слів за 20 годин. Повідомляються про сильні результати, з озадаченням, що знизився з 71 до 60 в LM, а відповідна оцінка BLEU отримала 1,8 бала, використовуючи нейромережеві мовні моделі з вбудованими словами, порівняно з найкращими відступними LM.

Більш недавнє дослідження щодо застосування методів глибокого навчання до машинного перекладу, де компонент перекладу фраз, а не компонент LM у системі машинного перекладу замінюється моделями

нейронних мереж із вбудованим семантичним словом. Як показано на рис. 1.9 для архітектури цього підходу, пара джерел (позначена як f) і ціль (позначається e) фрази проектується у безперервно значущі векторні подання у низьковимірному прихованому семантичному просторі (позначається двома у векторами). Тоді оцінка їх перекладу обчислюється на відстані між парою в цьому новому просторі. Проекція виконується двома глибокими нейронними мережами, ваги яких вивчаються на даних паралельних тренувань. Навчання спрямоване на безпосередню оптимізацію якості наскрізних результатів машинного перекладу. Експериментальна оцінка була проведена щодо двох стандартних завдань перекладу Europarl, що використовуються спільнотою NLP, англійсько-французької та німецько-англійської. Результати показують, що нова семантична модель перекладу фраз суттєво покращує роботу сучасної фразової статистичної системи машинного перекладу, що призводить до виграшу, близького до 1.0 BLEU.

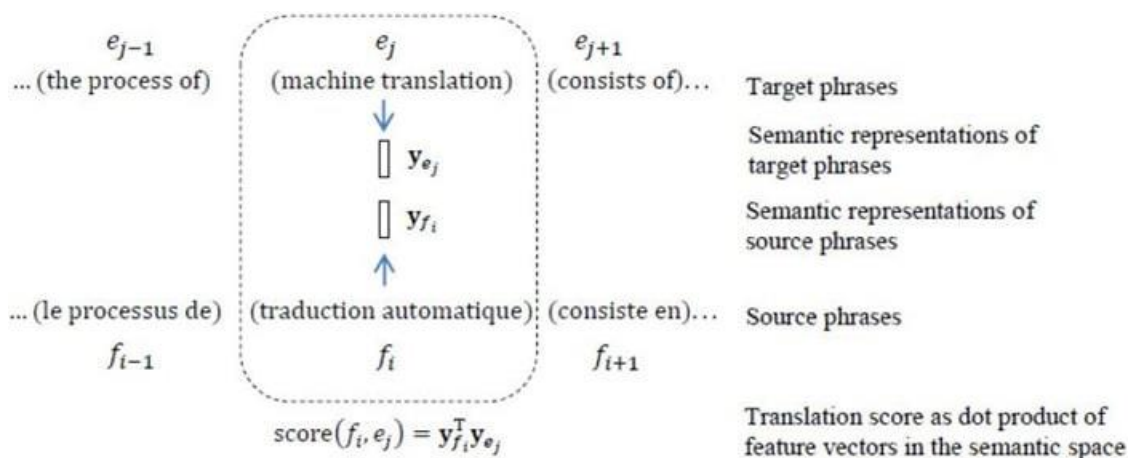


Рисунок 1.9 – Ілюстрація основного підходу, описаного для машинного перекладу

Відповідний підхід до машинного перекладу був розроблений Швенком. Оцінка ймовірностей моделі перекладу системи машинного перекладу на основі фраз проводиться за допомогою нейронних мереж. Імовірність перекладу пар фраз вивчається за допомогою подань у безперервному просторі, індукованих нейронними мережами. Здійснено спрощення, яке

розкладає ймовірність перекладу фрази або речення на добуток n -грамних ймовірностей, як у стандартному n -грамна модель мови. Жодне спільне подання фрази мовою оригіналу та перекладеної версії цільовою мовою не використовується.

Ще один глибокий підхід до машинного перекладу. Як і в інших підходах, корпус слів в одній мові порівнюється з одним і тим же корпусом слів, перекладеним на іншу, а слова та фрази в таких двомовних даних, що мають подібні статистичні властивості, вважаються рівнозначними. Нова техніка є запропонована, що автоматично формує словники та таблиці фраз, які перетворюють одну мову на іншу. Він не покладається на версії одного і того ж документа різними мовами. Натомість він використовує методи аналізу даних для моделювання структури вихідної мови, а потім порівнює її зі структурою цільової мови. Показана методика перекладу відсутніх слів та фраз шляхом вивчення мовних структур на основі великих одномовних даних та відображення між мовами з невеликих двомовних даних. Він базується на вбудованих словах з векторним значенням і вивчає лінійне відображення між векторними просторами вихідної та цільової мов.

Раніше дослідження щодо застосування методів глибокого навчання з DBN було запропоновано в [13] для нападу на проблему машинної транслітерації, що набагато простіше, ніж машинне перекладання. Цей тип глибоких архітектур та навчання може бути узагальнений для більш складної проблеми машинного перекладу, але жодної подальшої роботи не надходило. В якості ще однієї ранньої програми NLP, застосовано DNN (у статті названі DBN) для виконання завдання маршрутизації викликів на природній мові. DNN використовують неконтрольоване навчання для виявлення декількох шарів функцій, які потім використовуються для оптимізації дискримінації. Встановлено, що безконтрольне виявлення функцій робить DBN набагато менш схильними до переоснащення, ніж нейронні мережі, ініціалізовані випадковими вагами. Ненаглядне навчання також полегшує тренування нейронних мереж з безліччю прихованих шарів.

Одним із найцікавіших завдань NLP, яке нещодавно вирішувалось методами глибокого навчання, є заповнення бази знань (онтологія), що є важливим фактором у відповіді на запитання та багатьох інших додатках щодо NLP. Рання робота в цьому просторі з'явилася, де запроваджено процес автоматичного вивчення структурованих розподілених вкладених баз знань. Запропоновані подання у безперервно-значущому векторному просторі є компактними і їх можна ефективно вивчити з великомасштабних даних сутностей та відносин. Використовується спеціалізована архітектура нейронних мереж, узагальнення «сіамської» мережі. У подальшій роботі, що зосереджується на багатореляційних даних [9], семантична модель енергії узгодження пропонується для вивчення векторних подань для як сутності, так і відносини. Більш пізня робота застосовує альтернативний підхід, заснований на використанні нейронних тензорних мереж, для атаки на проблему міркувань через великий спільний графік знань для класифікації відносин. Графік знань представлений у вигляді трійки відносин між двома сутностями, і автори прагнуть розробити модель нейронної мережі, придатну для висновку про такі відносини. Модель, яку вони представили, являє собою нейронну тензорну мережу, лише з одним шаром. Мережа використовується для представлення сутностей у векторах із фіксованою вимірністю, які створюються окремо шляхом усереднення попередньо навчених векторів вбудовування слів. Потім він засвоює тензор із нещодавно доданим елементом взаємозв'язку, який описує взаємодію між усіма прихованими компонентами в кожному з відносин. Нейронна тензорна мережа показана на рис. 1.10, де кожна пунктирна рамка позначає один із двох зрізів тензора. Експериментально показує, що ця тензорна модель може ефективно класифікувати невидимі взаємозв'язки в WordNet та FreeBase.

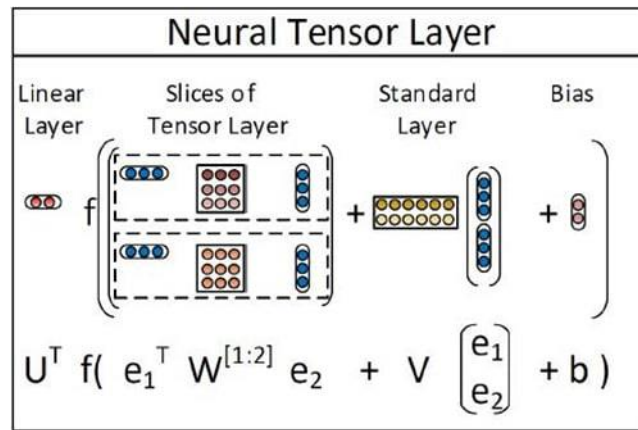


Рисунок 1.10 – Ілюстрація нейронної тензорної мережі з двома співвідношеннями, показаними як два зрізи в тензорі.

На рисунку 1.10 тензор позначається як $W^{[1:2]}$. Мережа містить білінійний тензорний шар, який безпосередньо пов'язує дві сутності (показано як e_1 і e_2) у трьох вимірах. Кожна пунктирна рамка позначає один із двох зрізів тензора.

Як остаточний приклад глибокого навчання, який успішно застосовується до NLP, ми обговорюємо тут додатки для аналізу настроїв на основі рекурсивної глибокої моделі [18]. Аналіз настрою – це завдання, яке спрямоване на оцінку позитивної чи негативної думки за допомогою алгоритму на основі вхідної текстової інформації. Як ми обговорювали раніше в цьому пункті, вбудовування слів у семантичний простір, досягнуте моделями нейронних мереж, було дуже корисним, але їм складно висловити значення довгих фраз принципово. Для аналізу настроїв із вхідними даними, як правило, з багатьох слів та фраз, модель вбудовування вимагає властивостей композиції. З цією метою Сочер розробив рекурсивну нейронну тензорну мережу, де кожен шар побудований аналогічно тому, що нейронна тензорна мережа з ілюстрацією, зображеною на рисунку 1.10. Рекурсивна побудова повної мережі, що демонструє властивості композиції, слідує конструкції [20] для звичайної, не тензорної мережі. Під час навчання на ретельно побудованій базі даних аналізу настроїв показано, що рекурсивна нейронна тензорна мережа перевершує всі попередні методи за кількома показниками. Нова

модель підвищує рівень техніки в точності позитивної/ негативної класифікації з одного речення з 80% до 85,4%. Точність прогнозування дрібнозернистих ярликів настроїв для всіх фраз досягає 80,7%, покращення на 9,7% порівняно з базовими лініями функцій.

2 МЕТОДИ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ КОМП'ЮТЕРНОГО БАЧЕННЯ

2.1 Вибрані програми в розпізнаванні об'єктів та комп'ютерне бачення

За останній час було досягнуто колосального прогресу у застосуванні методів глибокого навчання до комп'ютерного зору, особливо в області розпізнавання об'єктів. Успіх глибокого навчання в цій галузі зараз загальноновизнаний спільнотою комп'ютерного зору. Це друга область, в якій застосування методів глибокого навчання є успішним, слідуючи області розпізнавання мовлення.

Протягом багатьох років розпізнавання об'єктів у комп'ютерному зорі покладається на власноруч розроблені функції, такі як SIFT (перетворення інваріантної шкали характеру) та HOG (гістограма орієнтованих градієнтів), подібне до залежності розпізнавання мови від розроблених вручну функцій, таких як MFCC та PLP. Однак такі функції, як SIFT та HOG, фіксують лише низькорівневу крайню інформацію. Дизайн функцій для ефективного захоплення інформації середнього рівня, наприклад перетину країв або представлення високого рівня, таких як частини об'єктів, стає набагато складнішим. Поглиблене навчання має на меті подолати такі виклики шляхом автоматичного навчання ієрархій візуальних особливостей як без нагляду, так і під контролем безпосередньо з даних. Огляд нижче класифікує багато методів глибокого навчання, що застосовуються до комп'ютерного зору, на два класи:

- 1) неконтрольоване вивчення особливостей, де глибоке навчання використовується для вилучення лише функцій, які згодом можуть передаватися відносно простому алгоритму машинного навчання для класифікації чи інших завдань;

- 2) контрольовані методи навчання, коли наскрізне навчання

застосовується для спільної оптимізації компонентів екстрактора та класифікатора характеристик повної системи, коли доступні великі обсяги маркованих навчальних даних.

2.1.1 Безконтрольне або генеративне навчання особливостей

Коли маркованих даних відносно мало, показано, що безконтрольні алгоритми навчання засвоюють корисні ієрархії візуальних особливостей. Насправді, до демонстрації чудових успіхів архітектур CNN з контрольованим навчанням у конкурсі ImageNet 2012 року, більша частина роботи із застосування методів глибокого навчання до комп'ютерного зору проводилася з вивчення функцій без нагляду. Оригінальний невидимий глибокий автокодер, який використовує попередню підготовку до DBN, був розроблений та продемонстрований Хінтоном та Салахутдіновим [14] з успіхом у задачах розпізнавання зображень та зменшення розмірності (кодування) MNIST, маючи лише 60 000 зразків у навчальному наборі. Крім того, розроблено модифікований DBN, де модель верхнього шару використовує машину Больцмана третього порядку [17]. Цей тип DBN застосовується до бази даних NORB – тривимірного завдання розпізнавання об'єктів. Повідомляється про рівень помилок, близький до найкращого опублікованого результату для цього завдання. Зокрема, показано, що DBN істотно перевершує неглибокі моделі, такі як SVM. Розроблено дві стратегії для підвищення стійкості DBN. По-перше, розріджені зв'язки в першому шарі DBN використовуються як спосіб регулювання моделі. По-друге, розробляється імовірнісний алгоритм зменшення шуму. Показано, що обидва методи ефективно впливають на підвищення стійкості проти оклюзії та випадкових шумів у завданні з шумним розпізнаванням зображень. DBN також успішно застосовуються для створення компактних, але значущих зображень з метою пошуку. У цьому великому завданні отримання зображень колекції підходи глибокого навчання також дали значні результати. Крім того,

про використання умовно-часового DBN для відеопослідовності та синтезу руху людини повідомлялося в [20]. Умовні RBM та DBN роблять ваги RBM та DBN асоційованими з фіксованим часовим вікном, зумовленим даними з попередніх кроків часу.

Методи глибокого навчання мають багату родину, включаючи ієрархічні ймовірнісні та генеративні моделі (нейронні мережі чи інше). Одним з останніх прикладів цього типу, розробленим і застосованим до наборів даних виразів обличчя, є стохастичні нейромережі прямого зворотного зв'язку, які можна ефективно вивчити і які можуть спричинити багатий багаторежимний розподіл у вихідному просторі, що неможливо за допомогою стандартної, детермінованої нейронної мережі. На рисунку 2.1 показано архітектуру типової стохастичної нейронної мережі з чотирма прихованими шарами зі змішаними детермінованими та стохастичними нейронами (ліворуч), що використовується для моделювання багаторежимних розподілів, зображених праворуч. Стохастична мережа тут є глибокою, спрямованою графічною моделлю, де процес генерації починається з введення x , нейронної грані, і генерує вихід y , вираз обличчя. У експериментах з класифікацією виразів обличчя вивчені приховані некеровані функції, генеровані з цієї стохастичної мережі, додаються до пікселів зображення і допомагають отримати вищу точність базового класифікатора на основі умовного RBM/ DBN [20].

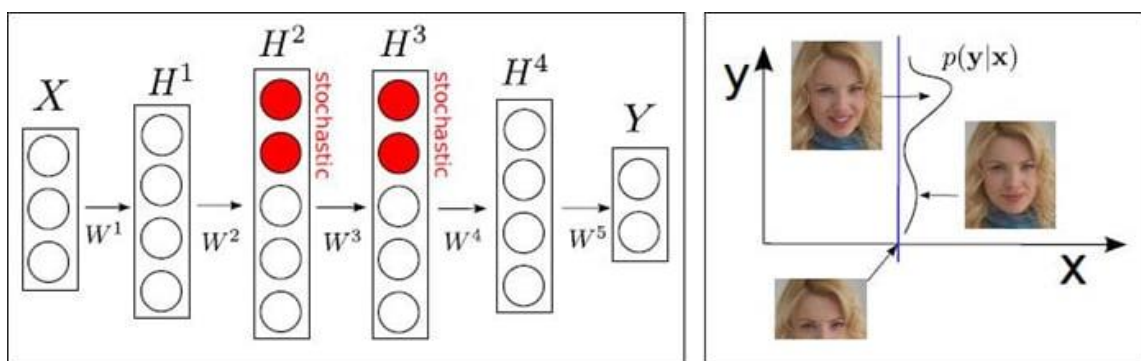


Рисунок 2.1 – Типова архітектура стохастичної нейронної мережі з чотирма прихованими шарами

Мабуть, найбільш помітною роботою в категорії безконтрольного глибокого вивчення функцій для комп'ютерного зору (до недавнього сплеску роботи на CNN), дев'ятишарового локально з'єданого розрідженого автокодера з об'єднанням і локальною нормалізацією контрасту. Модель має мільярд з'єднань, навчених набору даних з 10 мільйонами зображень, завантажених з Інтернету. Безконтрольні методи вивчення особливостей дозволяють системі тренувати детектор обличчя без необхідності позначати зображення як такі, що містять обличчя чи ні. А контрольні експерименти показують, що цей функціональний детектор надійний не тільки для перекладу, але також для масштабування та обертання поза площиною.

Інший набір популярних досліджень про глибоке вивчення глибоких функцій для комп'ютерного зору базується на моделях глибокого розрідженого кодування [16]. Цей тип глибоких моделей дав найсучасніші результати точності для завдань розпізнавання об'єктів ImageNet до появи архітектур CNN, озброєних контрольованим навчанням для спільного навчання та класифікації характеристик, до яких ми зараз звертаємося.

2.1.2 Навчання та класифікація ознак під наглядом

Походження застосувань глибокого навчання для розпізнавання об'єктів можна простежити у CNN на початку 90-х. Архітектури, засновані на CNN, у режимі навчання під контролем викликають великий інтерес до комп'ютерного зору з жовтня 2012 року, незабаром після оприлюднення результатів конкурсу ImageNet. Це головним чином завдяки величезному виграшу точності розпізнавання порівняно з конкуруючими підходами, коли доступні великі обсяги маркованих даних для ефективної підготовки великих мереж CNN за допомогою високопродуктивних обчислювальних платформ, схожих на GPU. Подібно до того, як методи глибокого навчання, засновані на DNN, перевершили попередні найсучасніші підходи до розпізнавання мовлення у ряді базових завдань, включаючи розпізнавання телефону,

розпізнавання мови з великим словником, надійне розпізнавання мови та багатомовне розпізнавання мови, засновані на CNN. Методи глибокого навчання продемонстрували те саме у наборі завдань комп'ютерного зору, включаючи розпізнавання об'єктів на рівні категорії, виявлення об'єктів та семантичну сегментацію.

Основна архітектура CNN [16] показана на рисунку 2.1. Щоб врахувати відносну незмінність просторового співвідношення в типових пікселях зображення щодо місця розташування, CNN використовує згортковий шар з локальними сприйнятливими полями і з прив'язаними вагами фільтра, подібний до двовимірних FIR-фільтрів при обробці зображень. Потім вихідні дані FIR-фільтрів передаються через нелінійну функцію активації для створення карт активації, за яким слідує інший нелінійний рівень (позначений як «субдискретизація» на рис. 2.2), який зменшує швидкість передачі даних, забезпечуючи інваріантність дещо іншим вхідним зображенням. Вихід шару об'єднання надходить до кількох повністю з'єднаних шарів, як у DNN.

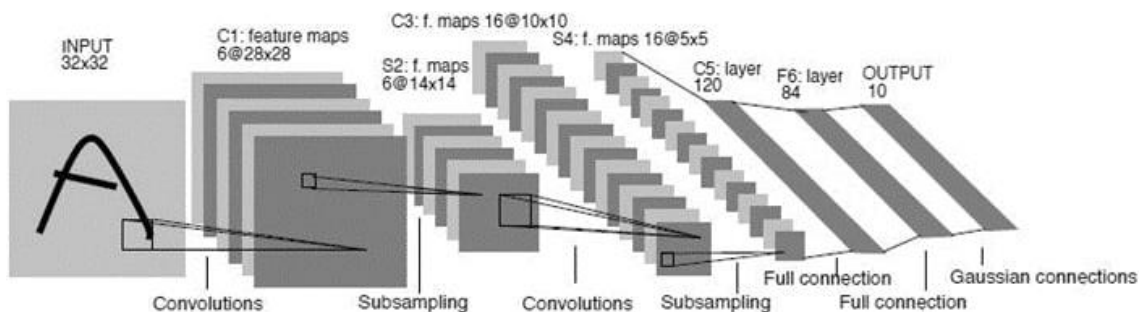


Рисунок 2.2 – Оригінальна згорткова нейронна мережа, яка складається з багат шарової змінної згортки та об'єднання шарів, за якими слідує повністю пов'язані шари

Глибокі моделі зі структурою згортки, такі як CNN, були визнані ефективними і використовуються в комп'ютерному зорі та розпізнаванні зображень з 90-х років. Найвизначніший прогрес був досягнутий у конкурсі LSVRC ImageNet 2012 року, в якому завдання полягає у підготовці моделі з 1.2 мільйонами зображень з високою роздільною здатністю для класифікації

невидимих зображень до одного з 1000 різних класів зображень. На тестовому наборі, що складається з 150 тис. зображень, глибокий підхід CNN, довів рівень помилок значно нижчий, ніж попередній сучасний рівень. Використовуються дуже великі глибокі CNN, що складаються з 60 мільйонів ваг і 650 000 нейронів, і п'яти згорткових шарів разом з шарами, що об'єднують максимум. Додаткові два повністю з'єднаних шари, як у DNN, описаному раніше, використовуються поверх шарів CNN. Незважаючи на те, що всі вищезазначені структури були розроблені окремо в попередніх роботах, їх найкраще поєднання становило основну частину успіху. Загальну архітектуру глибокої системи CNN зображено на рисунку 2.3.

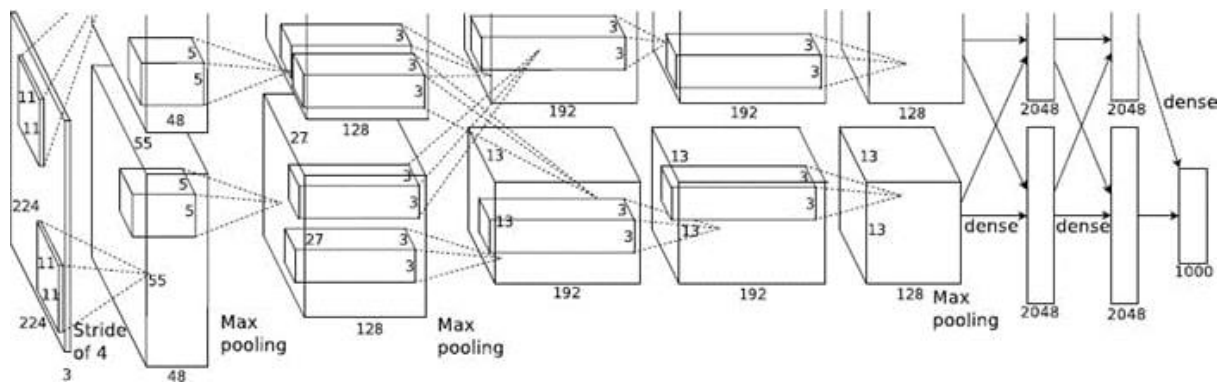


Рисунок 2.3 – Архітектура системи глибокого CNN, яка перемогла у 2012 році у конкурсі ImageNet

Два додаткові фактори сприяють остаточному успіху. Перший – це потужна техніка регуляризації, яка називається «відсівання». Зокрема, Уорд-Фарлі та інші [20] проаналізував ефекти розплутування відсіву та показав, що це допомагає, оскільки різні члени пакета поділяють параметри. Застосування тих самих методів «відсіву» також є успішним для деяких завдань розпізнавання мовлення. Другим фактором є використання ненасичуючих нейронів або випрямлених лінійних одиниць (ReLU), які обчислюють як $f(x) = \max(x, 0)$, що значно прискорює загальний навчальний процес, особливо за допомогою ефективної реалізації графічного процесора. Ця система глибокого CNN досягла переможного рейтингу помилок на 5-му

місці, який становив 15,3%, використовуючи додаткові навчальні дані з випуску ImageNet Fall 2011, або 16,4%, використовуючи лише подані навчальні дані в ImageNet 2012, що значно нижче, ніж 26,2%, досягнене другим найкращим система, яка поєднує оцінки з багатьох класифікаторів, використовуючи набір функцій, створених вручну, таких як вектори SIFT та Fisher.

Сучасні показники з використанням підходу глибокого CNN, додатково вдосконалюються ще одним значним результатом протягом 2013 року, використовуючи подібний підхід, але з більшими моделями та більшими обсягами навчальних даних. Підсумок – 5 найкращих коефіцієнтів помилок тестів від 11 найефективніших команд, що беруть участь у конкурсі ImageNet ILSVRC 2013 року, наведено на рисунку 2.4, а найкращий результат змагань 2012 року – найкращий для базової лінії.

Тут ми бачимо швидке скорочення помилок у тому самому завданні з найнижчого рівня помилок до 2012 року – 26,2% (ненейронні мережі) до 15,3% у 2012 році та далі до 11,2% у 2013 році, як досягнутих завдяки глибокій технології CNN. Цікаво також зауважити, що всі основні роботи в конкурсі ImageNet ILSVRC 2013 року базуються на підходах глибокого навчання. Наприклад, система Adobe, показана на рис. 2.4, базується на глибокому CNN [20], включаючи використання відсіву. Мережева архітектура модифікована, щоб включати більше фільтрів та з'єднань. Під час тестування показник питомості зображень використовується для отримання 9 культур із вихідних зображень, які поєднуються із стандартними п'ятьма зерновими культурами. Система NUS використовує непараметричний адаптивний метод для поєднання результатів від кількох неглибоких та глибоких експертів, включаючи глибокі CNN, ядерні та GMM методи.

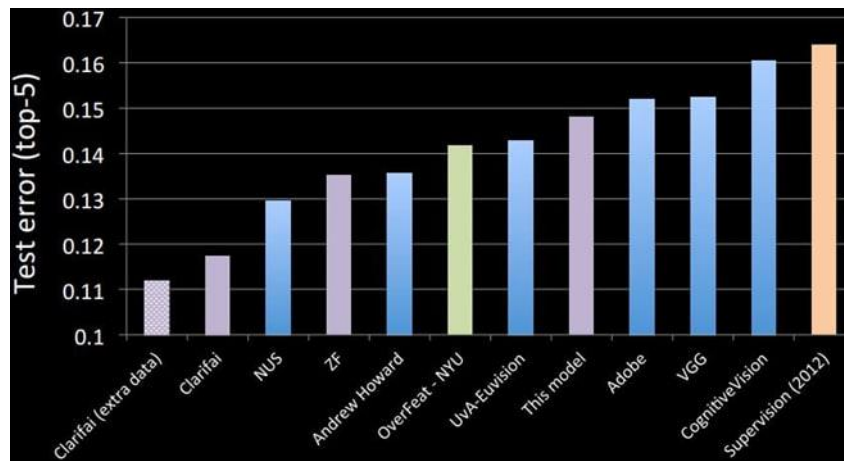


Рисунок 2.4 – Підсумкові результати проекту ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013), що представляють найсучаснішу ефективність систем розпізнавання об’єктів

Система VGG використовує комбінацію глибокої векторної мережі Фішера та глибокої CNN. Система ZF базується на поєднанні великої мережі CNN з різноманітними архітектурами. Вибору архітектур допомагала візуалізація особливостей моделі за допомогою деконволюційної мережі, як описав Цайлер та ін. [21]. Система CognitiveVision використовує схему класифікації зображень, засновану на архітектурі DNN. Метод натхненний когнітивною психофізикою про те, як система зору людини спочатку вчиться класифікувати категорії базового рівня, а потім вчиться класифікувати категорії на підпорядкованому рівні для дрібновизначеного розпізнавання об’єктів. Нарешті, найефективніша система, названа Clarifai на рис.2.4, базується на великій та глибокій CNN з регуляризацією відсіву. Це збільшує кількість навчальних даних шляхом вибіркової вибірки зображень до 256 пікселів. Система містить загалом 65 млн параметрів. Кілька таких моделей були усереднені разом для подальшого підвищення продуктивності. Основною новинкою є використання методу візуалізації, заснованого на деконволюційних мережах, як описано в [21], щоб визначити, що робить глибинну модель ефективною, на основі якої була обрана потужна глибинна архітектура.

Хоча глибокий CNN продемонстрував надзвичайну ефективність

класифікації завдань розпізнавання об'єктів, до недавнього часу не було чіткого розуміння того, чому вони виконують настільки добре. Цайлер та Фергюс провели дослідження, щоб вирішити саме цю проблему, а потім використали отримане розуміння для подальшого вдосконалення систем CNN, що дало чудові характеристики, як показано на рисунку 10.4 із позначками «ZF» та «Clarifai». Розроблена нова техніка візуалізації, яка дає уявлення про функцію проміжних шарів особливостей глибокого CNN. Ця техніка також проливає світло на роботу всієї мережі, яка діє як класифікатор. Техніка візуалізації заснована на деконволюційній мережі, яка відображає нейронні дії в проміжних шарах вихідної згорткової мережі назад у вхідний простір пікселів. Це дозволяє дослідникам вивчити, який шаблон введення спочатку спричинив дану активацію на картах функцій.

Рисунок 2.5 (верхня частина) ілюструє, як деконволюційна мережа приєднана до кожного зі своїх шарів, забезпечуючи тим самим замкнутий цикл назад до пікселів зображення як вхідного до вихідного CNN. Інформаційний потік у цьому замкнутому циклі такий: по-перше, вхідне зображення подається на глибокий CNN у зворотному напрямку, щоб обчислювати характеристики на всіх шарах. Для вивчення даної активації CNN всі інші активації на рівні встановлюються на нуль, а карти функцій передаються як вхідні дані до приєданого рівня деконволюційної мережі. Потім виконуються послідовні операції, протилежні обчисленню прямої подачі в CNN, включаючи роз'єднання, випрямлення та фільтрування. Це дозволяє реконструювати активність у шарі, що спричинив обрану активацію. Ці операції повторюються до досягнення вхідного рівня. Під час роз'єднання, необерненість операції максимального об'єднання в CNN становить вирішується за допомогою приблизного зворотного, де розташування максимумів у кожній області об'єднання реєструється у наборі змінних «перемикач». Ці перемикачі використовуються для розміщення реконструкцій з верхнього шару у відповідні місця, зберігаючи структуру стимулу. Ця процедура показана в нижній частині рис. 2.5.

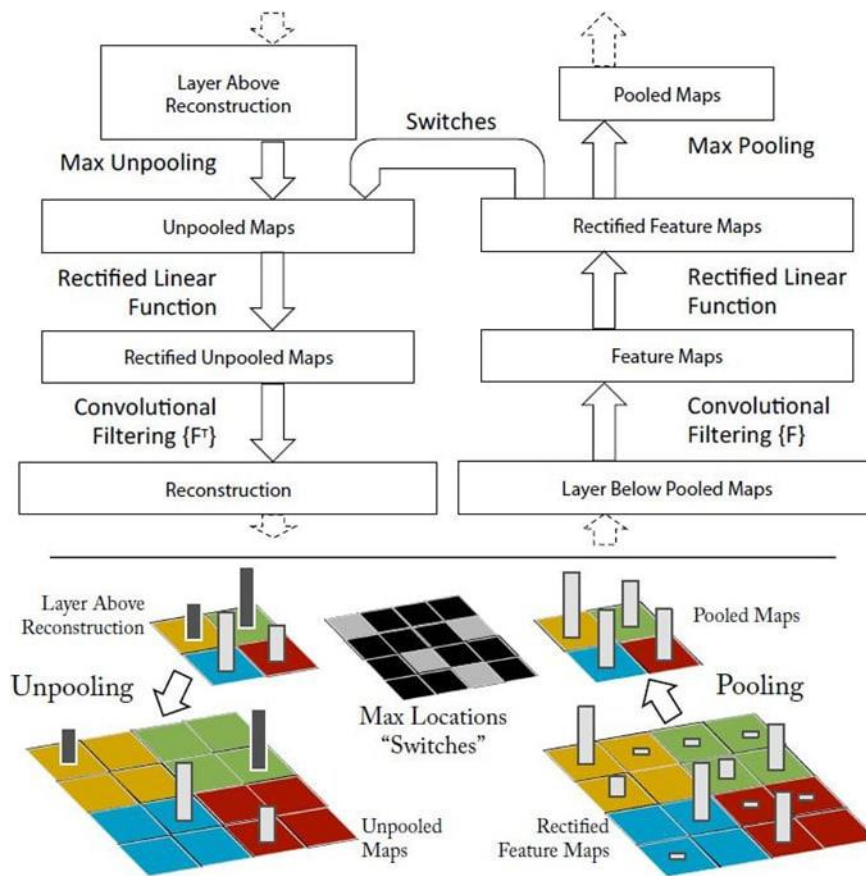


Рисунок 2.5 – Прикріплення деконволюційної мережі до відповідного шару CNN

Нарешті, найновіше дослідження контрольованого навчання для комп'ютерного зору показує, що глибока архітектура CNN є не лише успішною для класифікації об'єктів/ зображень, про яку вже йшлося в цьому розділі, але й успішною для виявлення заперечень на цілих зображеннях [13]. Завдання виявлення істотно складніше, ніж завдання класифікації.

Глибоке навчання зробило величезний прорив у комп'ютерному зорі, незабаром після його успіху в розпізнаванні мови. Наразі це контрольована парадигма навчання, заснована на глибокій архітектурі CNN та пов'язаних з нею класифікаційні техніки, які справляють найбільший вплив, продемонстровані результатами змагань ImageNet за 2012 та 2013 рр. Ці методи можна використовувати не лише для розпізнавання об'єктів, але й для багатьох інших завдань комп'ютерного зору. Були певні суперечки щодо причин успіху цих методів глибокого навчання, заснованих на CNN, та щодо

їх обмежень. Досі залишається відкритим багато питань щодо того, як ці методи можна пристосувати до певних програм комп'ютерного зору та як масштабувати моделі та навчальні дані. Для досягнення довгострокового та остаточного успіху в комп'ютерному зорі, ймовірно, знадобиться навчання без нагляду. З цією метою потрібно вирішити багато відкритих проблем у некерованому вивченні функцій та глибокому навчанні, а також провести набагато більше досліджень.

2.2 Колірний простір

2.2.1 Людський зір

Для людини колір – психологічне відчуття, викликане віддзеркаленням світла від об'єкта. Це відчуття виникає в мозку при порушенні та гальмуванні клітин, що чутливі до світла – рецепторів очної сітківки. В оці людини містяться два типи світлочутливих клітин, які називаються фоторецепторами: високочутливі палички і менш чутливі колбочки.

Палички функціонують в умовах відносно низької освітленості і відповідають за дію механізму нічного зору, однак при цьому забезпечують тільки нейтральне в колірному відношенні сприйняття дійсності. Колбочки працюють при більш високих рівнях освітленості або яскравості, ніж палички і відповідають за механізм денного зору, відмінною рисою якого є здатність забезпечення колірного зору. Колбочки відповідають червоній, зеленій і синій частині спектра і часто називаються довгими (L), середніми (M) і короткими (S) згідно довжинам хвиль, до яких вони найбільш чутливі. Кожне колірне відчуття людини може бути представлено у вигляді суми відчуттів трьох основних кольорів: червоного, зеленого і синього. Суб'єктивне сприйняття кольору залежить від багатьох параметрів: яскравості, швидкості зміни яскравості, колірної температури.

2.2.2 Представлення кольору в машинній графіці

Поняття кольору виникає при описі сприйняття очима людини електромагнітних хвиль в певному діапазоні частот. Людина сприймає хвилі довжиною від 400 нм – фіолетовий колір, до 700 нм – червоний колір [3]. Таким чином, найбільш загальним описом світлового потоку може служити його спектральна функція.

Піки на кривих чутливості відповідають червоному, зеленому і синьому кольорам. При цьому слід зауважити, що сприйнятливість до синього кольору значно нижче, ніж до двох інших. Також важливою властивістю сприйняття світла людиною є його лінійність: при освітленні двома джерелами світла зі спектральними функціями $I_1(\lambda)$ та $I_2(\lambda)$, людина сприймає їх як одну спектральну функцію, яка дорівнює сумі $I(\lambda) = I_1(\lambda) + I_2(\lambda)$. Цей факт називається законом Грассмана [3]. Так як області сприйняття для різних типів колб перекриваються, то виникають метамери – потоки хвиль з різними спектральними характеристиками, але сприймаються як такі, що мають один і той же колір.

У машинній графіці колірним простором називають математичну модель представлення кольору, в якій кожен колір являє собою координату в деякому просторі базисних кольорів. Для більшості колірних просторів відображення координат простору в кольори є бієктивне, хоча в загальному випадку таке відображення сюр'єктивне.

Кольороподілом називають технологічний етап відтворення кольорового зображення, при якому світло складного спектрального складу поділяється на кілька монохромних складових, кожна з яких містить інформацію тільки про один колір або іншому параметрі колірного простору. Отримані в результаті кольороподілу зображення називаються колірними каналами.

2.2.3 Кольорова палітра RGB

З розглянутої вище моделі людського зору випливає, що досить обґрунтованою є тривимірна колірна модель RGB, в якій базовими кольорами є червоний, зелений і синій відповідно. Кожна координата кольору – ціле число, що належить відрізку $[0,255]$. Таким чином, модель RGB містить $256^3 = 1677721$ кольорів. Ця модель характеризується властивістю адитивності в тому сенсі, що додавання двох кольорів і буде створювати новий колір, обчислений за формулою (2.1).

$$\frac{\text{sign}(255-x)+1}{2}(x-255)+255, x = (r_1 + r_2, g_1 + g_2, b_1 + b_2). \quad (2.1)$$

Векторам $(x, x, x), x \in [0,255]$ в системі RGB відповідають кольору градації сірого, причому нульовому вектору $(0,0,0)$ відповідає чорний колір, а вектору $(255,255,255)$ – білий. Коли один з компонентів досить великий тобто лежить у відрізку $[200,255]$, а два інші у $[0,50]$ тобто малі, то одержуваний відтінок близький до домінуючого кольору. Інакше, якщо величина двох будь-яких кольорів велика, а того, що залишився – мала, то одержувані відтінки називають вторинними кольорами. Їх назви можна побачити в таблиці 2.1.

Таблиця 2.1 – Домінуючі та відповідні їм вторинні кольори

Домінуючі кольори	Відповідні їм вторинні кольори
R, G	Yellow (Жовтий)
R, B	Magenta (Пурпуровий)
G, B	Cyan (Блакитний)

Колірна модель RGB знайшла широке поширення в техніці, використовується в моніторах і проекторах.

2.2.4 Кольорова палітра CIE LAB

CIE LAB (також позначається як $L * a * b$) – колірний простір, введений міжнародною комісією з освітлення (фр. Commission internationale de l'éclairage) в 1976 році. Простір являє собою тривимірний простір R^3 , в якому можна представити нескінченне число кольорів, включаючи ті, що невидимі людському оку. Для цифрового представлення, CIE LAB відображається як обмежений тривимірний цілочисловий простір. Часто L лежить в межах $[0,100]$, a і b в межах $[-128,127]$ або $[0,255]$.

Величина L відповідає за яскравість точки: $L = 0$ представляє чорний колір, $L = 100$ – білий. Значення $a = 0$ і $b = 0$ відповідають нейтрально сірим відтінкам. Вісь a представляє червоно-зелену компоненту кольору, зелені кольори знаходяться на негативних значеннях абсцис, червоні – на позитивних. Аналогічно b представляє жовто-блакитні кольори, блакитні на негативній частині прямої, жовті – на позитивній. У разі якщо $a, b \in [0,255]$, нулем відліку вважається точка 128.

У моделі, між елементами L, a, b задані нелінійні відносини, призначені для імітації нелінійного відгуку. Крім того, рівномірні зміни компонентів у колірному просторі $L * a * b$ прагнуть відповідати рівномірним змінам кольору, що приймається людиною, тому відносні відстані або відмінності в сприйнятті між будь-якими двома кольорами в просторі $L * a * b$ можна апроксимувати, розглядаючи кожен колір як точку в тривимірному просторі. Тоді відстань між кольорами буде визначатися через Евклідовому відстані між відповідними точками [5].

З МЕТОДИ ОБРОБКИ ЗОБРАЖЕННЯ ТА ЇХ ФІЛЬТРАЦІЯ

3.1 Порогова обробка (Thresholding)

Порогова обробка – найпростіший метод сегментації чорно-білих зображень. Замінює кожен піксель зображення на білий (255), якщо значення кольору більше, ніж певний поріг. Якщо значення пікселя менше порога – колір пікселя замінюється на чорний (0). Метод також називають бінаризація. Якщо об'єкт, який нас цікавить має білий колір і розташований на чорному тлі або навпаки, то визначення точок об'єкта представляє собою тривіальне завдання встановлення порога по середній яскравості. Поріг середньої яскравості – значення, з яким порівнюється яскравість кожного пікселя.

Приклад: алгоритм автоматичного підбору порога.

- 1) задати початкове наближення – поріг (наприклад, використовувати половину яскравості зображення);
- 2) розділити зображення на дві частини: область, в якій яскравість пікселів менше або дорівнює пороговій; область, в якій яскравість пікселів більше порогової;
- 3) обчислити середню яскравість на кожній області;
- 4) обчислити новий поріг як середнє середніх яскравостей на обчислених областях;
- 5) якщо новий поріг відрізняється від попереднього не більше, ніж на задану малу величину – то поріг обчислений і дорівнює новому порогу. В іншому випадку замінити значення порога новим і повторити другий крок.

Крім евристичних методів пошуку порога, широко застосовуються і статистичні методи, такі як метод Оцу. Метод Оцу, який передбачає наявності двох піків, що виділяються на гістограмі. Метод знаходить таке значення порога при яких він знаходиться на рівній відстані від піків. Приклад роботи методу Оцу можна побачити на рис 3.1.



Рисунок 3.1 – Приклад бінаризації методом Оцу

Коли зображення освітлено нерівномірно, використовується адаптивна порогова обробка. Суть її в тому, що поріг задається не глобально для всього зображення, а обчислюється локально для деякої області. Це дозволяє виділити контури зображення там, де бінаризація з фіксованим для всього зображення порогом впоратися не може.

3.2 Лінійні фільтри

Згладжуючі фільтри роблять зображення нечіткими або розмитими. Зокрема для згладжування контурів фігур використовуються фільтр, так звана згортка. Його суть полягає в тому, що кожен піксель зображення замінюється на деяке середнє значення оточуючих його пікселів. Кожен фільтр має своє ядро – матрицю коефіцієнтів, на яку множаться сусідні пікселі цільового зображення. Це ядро може мати різну розмірність, в залежності від неї збільшується або зменшується інтенсивність фільтра.

Найпростішим фільтром згортки є фільтр усереднення [2]. У нього квадратне ядро, кожен елемент дорівнює $\frac{1}{(n*m)}$, де $n*m$ – розмірність ядра. Ядро унормовано, щоб процедура подавлення шуму не викликала зміщення середньої яскравості обробленого зображення. Інший приклад – розмиття по Гаусу. Його ядро для пікселя (m, n) розмірності (u, v) радіуса r обчислюється

за формулою:

$$y(m, n) = \frac{1}{2\pi r^2} \sum_{u,v} e^{\frac{-(u^2+v^2)}{2r^2}} x(m+u, n+v) . \quad (3.1)$$

3.3 Нелінійні фільтри

Зображення може пошкоджуватися шумами і перешкодами різного походження, наприклад шумом відеодатчика, шумом зернистості фотоматеріалів і помилками в каналі передачі. Їх вплив можна мінімізувати, користуючись класичними методами статистичної фільтрації. Інший можливий підхід заснований на використанні евристичних методів просторової обробки. Шуми відеодатчиків або помилки в каналі передачі зазвичай проявляються на зображенні як розрізнені зміни ізольованих елементів, що не володіють просторовою кореляцією. Спотворені елементи часто досить помітно відрізняються від сусідніх елементів. Це спостереження послужило основою для багатьох алгоритмів, що забезпечують подавлення шуму.

Медіанна фільтрація – метод нелінійної обробки сигналів, запропонована Тьюкі. Особливо ефективний для фільтрації білого шуму. Одновимірний медіанний фільтр являє собою вікно, яке охоплює непарне число елементів, зображення. Центральний елемент замінюється медіаною всіх елементів зображення у вікні. Медіаною дискретної послідовності $a_1 \dots a_n$ для непарного n є той її елемент, для якого існують $\frac{n-1}{2}$ елементів, менших або рівних йому за величиною, і $\frac{n-1}{2}$ елементів, більших або рівних йому за величиною [2].

Вікно переміщується уздовж сигналу, що фільтрується і обчислення повторюються. На відміну від фільтра усереднення центральний піксель зображення не обчислюється, а замінюється деяким пікселем з вікна, що збільшує якість фільтрації. Приклад використання медіанного фільтра можна

побачити на рисунку 3.2. Медіанний фільтр є нелінійним тому медіана суми двох довільних послідовностей не дорівнює сумі їх медіан, що в ряді випадків може ускладнювати математичний аналіз сигналів.

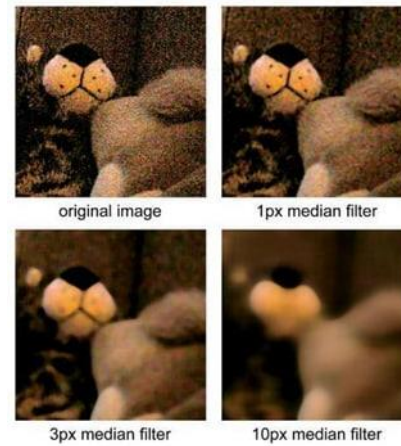


Рисунок 3.2 – Приклад використання медіанного фільтра до зашумлення зображення з трьома різними значеннями радіусу вікна фільтрації

3.4 Математична морфологія

Теорія і техніка морфологічного аналізу і обробки зображень заснована на теорії множин. Розглянемо бінарну морфологію: зображення представляється у вигляді прямокутних бінарних матриць, де одиниця означає білий колір, а нуль – чорний. Для кожної морфологічної операції, так само, як і для фільтрів, необхідно ядро, яке в даному випадку називається структурним елементом [4]. Основні структурні елементи можна побачити на рисунку 3.3.

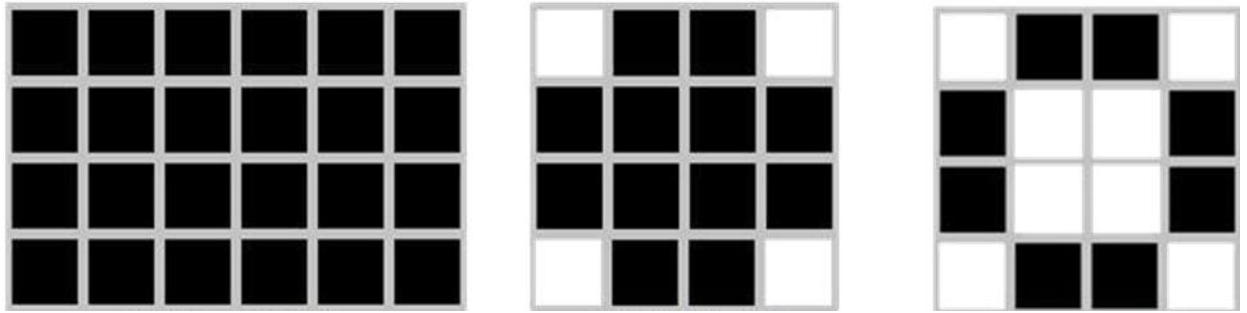


Рисунок 3.3 – Основні структурні елементи математичної морфології:

- 1) BOX [H, W] – прямокутник заданого розміру; 2) DISK [R] – диск заданого розміру; 3) RING [R] – кільце заданого розміру.

Нехай $P(x, y) \in N^2$ – координати пікселів зображення.

Базові операції:

– Перенесення (3.2)

$$P_t = \{x + t | x \in P\}. \quad (3.2)$$

Операція зміщує кожен піксель зображення на вектор t .

– Нарощування (3.3)

$$A \oplus B = \bigcup_{b \in B} A_b. \quad (3.3)$$

Структурний елемент B застосовується до всіх пікселів бінарного зображення. Кожен раз, коли початок координат (центр) структурного елементу поєднується з одиничним бінарним пікселем, до всього структурного елементу застосовується перенесення і подальше роз'єднання (логічне АБО) з відповідними пікселями бінарного зображення. Результати логічного складання записуються в результуюче бінарне зображення, яке спочатку ініціалізується нульовими значеннями.

– Ерозія (3.4)

$$A \ominus B = \{B_z \subseteq A\} \quad (3.4)$$

Якщо в деякій позиції кожен одиничний піксель структурного елементу збігається з одиничним пікселем бінарного зображення, то виконується роз'єднання центрального пікселя структурного елементу з відповідним пікселем вихідного зображення. В результаті застосування операції ерозії всі об'єкти, менші, ніж структурний елемент, стираються, об'єкти, з'єднані тонкими лініями, стають роз'єднаними і розміри всіх об'єктів зменшуються.

– Замикання (3.5)

$$A \cdot B = (A \oplus B) \ominus B \quad (3.5)$$

Операція замикання «закриває» невеликі внутрішні «дірки» в зображенні, і прибирає поглиблення по краях області. Якщо до зображення застосувати спочатку операцію нарощування, то ми зможемо позбутися від малих дірок і щілин, але при цьому відбудеться збільшення контуру об'єкта. Уникнути цього збільшення дозволяє операція ерозія, виконана відразу після нарощування з тим же структурним елементом.

– Розмикання (3.6)

$$A \circ B = (A \ominus B) \oplus B \quad (3.6)$$

Операція ерозії корисна для видалення малих об'єктів і різних шумів, але у цій операції є недолік – всі, об'єкти що лишилися зменшуються в розмірі. Цього ефекту можна уникнути, якщо після операції ерозії застосувати операцію нарощування з тим же структурним елементом. Розмикання відсіває всі об'єкти, менші ніж структурний елемент, але при цьому допомагає уникнути сильного зменшення розміру об'єктів. Також розмикання ідеально підходить для видалення ліній, товщина яких менше, ніж діаметр структурного елементу. Також важливо пам'ятати, що після цієї операції контури об'єктів стають більш гладкими.

3.5 Метод класифікації зображення: персептрон

Для вирішення завдання класифікації отриманих відфільтрованих зображень дуже ефективний метод класифікації, заснований на персептроні. Персептрон, в свою чергу, заснований на штучних нейронах – модель клітини мозку.

Штучний нейрон – зважений суматор, єдиний вихід якого визначається через його входи і матрицю ваг за формулою

$$y = f(u), u = \sum_{i=1}^n x_i w_i + x_0 w_0 A \circ B = (A \ominus B) \oplus B \quad (3.7)$$

Тут функція u – називається індукованим локальним полем, $f(u)$ – функцією активації або пороговою функцією, а x_i і w_i – відповідно є сигнали на входах нейрона і вага входів. Функція активації визначає залежність сигналу на виході нейрона від зваженої суми сигналів на його входах. У більшості випадків вона є монотонно зростаючою і має область значень $[0,1]$ або $[-1,1]$, проте існують винятки.

Персептрон складається з трьох основних структурних елементів: вхідного, прихованого і вихідного шару. Прихованих шарів може бути декілька. Кожен шар в персептроні пов'язаний з наступним, причому зв'язок повний тобто кожен нейрон першого шару пов'язаний з кожним нейроном наступного. Ребра, що з'єднують шари нейронів називають синапсами.

– Вхідний шар, традиційно, передає вхідний сигнал в кожен нейрон першого прихованого шару;

– Приховані шари складаються з нейронів, зазвичай в якості функції активації використовується

$$f(s) = \frac{1}{1+e^{-2as}} \quad (3.8)$$

– Вихідний шар також складається з нейронів, виходи вихідного шару

інтерпретуються як вихідне значення персептрона.

Щоб передбачити деякі події або, наприклад, визначити цифру, зображену на зображенні, на входи персептрона подається значення, зазвичай в межах $[0,1]$, що представляє вхідні дані. Дані з вхідного шару за допомогою синапсів подаються в перший прихований шар нейронів. Пройшовши всі приховані шари, сигнал потрапляє у вихідний шар, значення на виходах, яких називають виходом нейронної мережі або значенням нейронної мережі.

Навчання персептрона – обчислення таких коефіцієнтів для всіх нейронів крім вихідних, при яких значення нейронної мережі буде близькою до бажаного. Для навчання будь-якої нейромережі необхідний деякий досвід, який для персептрона представляється набором розмічених тестових даних. Розміченими вони називаються тому, що кожному вхідному набору ставиться у відповідність правильна відповідь, який нейронна мережа повинна видати в якості відповіді.

Для навчання персептрона ефективний метод зворотного поширення помилки. Для реалізації цього процесу необхідно поняття метрики. Метрика – функція, що визначає відстань між точками деякого простору, в нашому випадку простору відповідей R^n $[0,1]$, де n – кількість вихідних нейронів.

Навчання відбувається наступним чином: на вхід персептрона подається елемент навчальної вибірки, мережа обчислює своє вихідне значення, обчислюється помилка вихідного шару – відстань між правильною відповіддю і відповіддю нейронної мережі. Методом зворотного поширення помилки послідовно обчислюються значення помилок для всіх прихованих шарів персептрона, починаючи з останнього, а потім ваги штучних нейронів, ґрунтуючись на величині помилки, коригуються.

Перевага цього методу полягає в тому, що він може навчити всі шари нейронної мережі, і його легко прорахувати локально для кожного шару. Однак цей метод є дуже довгим, до того ж, для його застосування активаційна функція нейронів повинна диференціюватися.

4 РОЗРОБКА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ

4.1 Постановка завдання

Написати програму, яка б розпізнавала числа з тесту Ішіхара рис. 4.1.

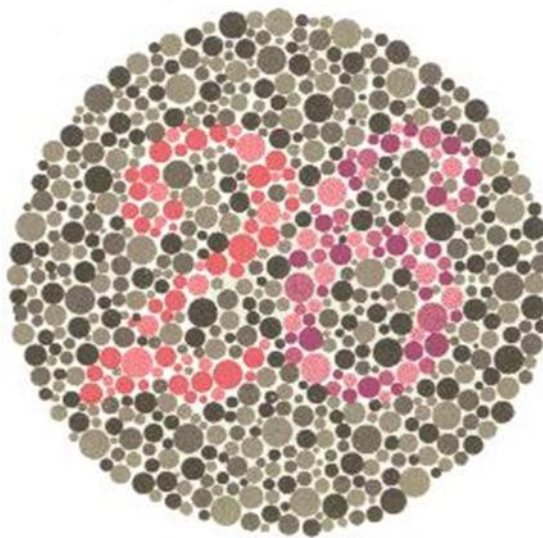


Рисунок 4.1 – Оригінальна картка тесту Ішіхара.

Тест Ішіхара – перший успішний тест сприйняття кольору, що складається з так званих псевдо-ізохроматических карток. Названий на честь Японського вченого Shinobu Ishihara, який опублікував цей тест у 1917 році.

Тест містить картки, на кожній з яких зображена окружність, що включає в себе точки (кола) різних кольорів і розмірів. Однак точки пофарбовані так, що деякі з них виділяють фон, інші – цифру. Люди з нормальним колірним сприйняттям можуть відокремити цифру від фону. Люди з дихроматичним зором можуть зробити помилкові висновки, так як бачать тільки два з трьох доступних колірних каналу. Люди з монохроматичним зором не зможуть розрізнити цифру.

Проведення тесту Ішіхара:

Людині яку тестують дається три секунди на те, щоб визначити цифру з

картки. Їй забороняється радитися або чіпати картки. Деякі з карток «легкі»: розпізнати цифру може навіть людина з монохромним зором. Щоб уникнути заучування карток вони перемішуються, так що «легкі» картки подаються упереміш зі звичайними.

Тест Ішіхара широко поширений завдяки високій точності і простоті тестування.

4.2 Використовувані технології

OpenCV – бібліотека алгоритмів комп'ютерного зору з відкритим вихідним кодом. До першої версії розроблялася в Центрі розробки програмного забезпечення Intel (російською командою в Нижньому Новгороді). OpenCV написана на мові високого рівня (C/C ++, Python) і містить алгоритми для інтерпретації зображень, калібрування камери за зразком, усунення оптичних спотворень, визначення подібності, аналіз переміщення об'єкта, визначення форми об'єкту та стеження за об'єктом, 3D-реконструкції, сегментації об'єкта, розпізнавання жестів і т.д.

Набула широкого поширення завдяки відкритості та можливості безкоштовного використання як в навчальних, так і комерційних цілях.

Keras – це бібліотека з відкритим вихідним кодом, що дозволяє легко створювати нейронні мережі. Бібліотека сумісна з TensorFlow, Theano і іншими бібліотеками машинного навчання. Tensorflow і Theano є найбільш часто використовуваними платформами для розробки алгоритмів глибокого навчання, але вони досить складні у використанні.

Keras навпаки надає простий і зручний спосіб створення моделей глибокого навчання. Її творець, François Chollet, розробив її для того, щоб максимально прискорити і спростити процес створення нейронних мереж. Він зосередив свою увагу на розширюваності, модульності, мінімалізмі і підтримки Python. Бібліотека keras внесла великий внесок в комерціалізацію глибокого навчання і штучного інтелекту, оскільки містить сучасні алгоритми

глибокого навчання, які раніше були не тільки недоступними, а й непридатними для використання.

Для тренування перцептрона і виявлення закономірностей кожної окремої цифри необхідно згенерувати вхідний набір даних. Для створення тестового набору була використана модифікована безкоштовна програма `ishihara_generator`. Також використовувалися шрифти `google fonts`.

4.3 Генерація тестового набору

Перший етап: генерація карток Ішіхара.

За допомогою програми `convert-im6` були згенеровані зображення цифр різних шрифтів (30 випадково вибраних з 2000 року наявних шрифтів). Таким чином, цифри на картках в тестовій вибірці мають різну форму, що не дасть перцептрону перенавчитися на єдиному шрифті. Картки генеруються в формі квадрата. Для генерації карток було вибрано 4 колірні теми, зображені на рис. 4.2.

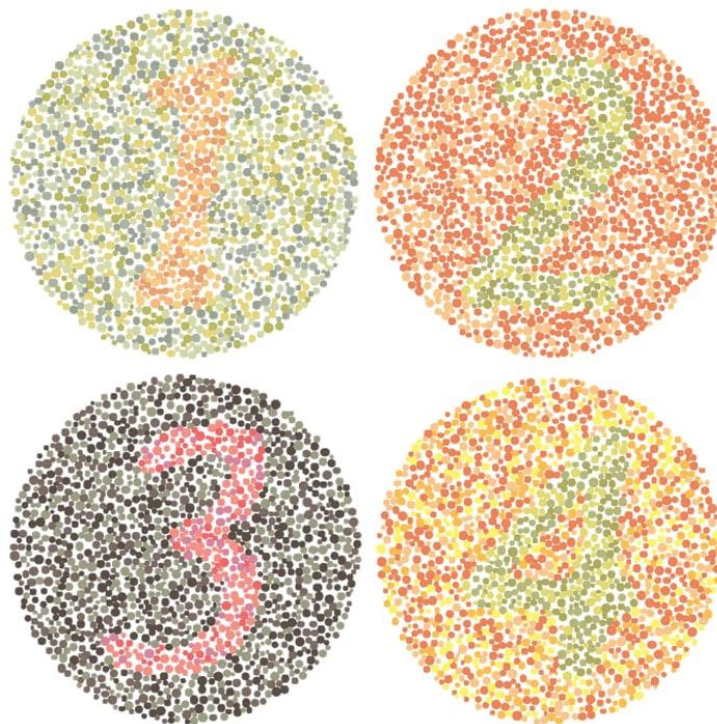


Рисунок 4.2 – Приклади згенерованих карток.

Після генерації зображення зашумлять: до яскравості кожного пікселя кожного каналу додавалося $-50 - 50$ одиниць. В підсумку було згенеровано 1400 карток Ішіхара, що цілком достатньо для навчання.

4.4 Фільтрація і векторизація зображень

Для фільтрації зображень була використана бібліотека OpenCV, а саме функції, які реалізують бінаризацію, фільтрацію і методи математичної морфології. На першому етапі проводилася бінаризація зображень. Всі зображення ділилися на три категорії:

1) Канал А в середньому темніше, ніж середина діапазону. Це означає, що на зображенні переважає зелений або синій колір. Отже, виходячи з наявних колірних тем, ми маємо червонувато-бежеву цифру на зеленому тлі. У каналі А червоні кольори дуже яскраві, так що для відділення цифри досить застосувати до цього каналу бінаризацію з порогом трохи вище середини діапазону.

Якщо ж в середньому канал А світлий, розглядається середнє значення яскравості каналу В

2) За умови світлого каналів А і В отримуємо середній колір в області червоного. З колірних тем видно, що цифра швидше за все буде зеленою. Червоний канал буде світлим, але зелена цифра буде досить темною на ній. Тому до червоного каналу застосовується фільтр з високим порогом, а після зображення інвертується.

3) В останньому випадку отримаємо середній колір в районі сірого і фіолетово-рожевого кольору. У цьому випадку цифра може бути червоно-рожевою на сірому тлі. Таким чином, в каналі А спостерігається сильний кольоровий контраст, і метод бінаризації Оцу добре відокремлює цифру від фону.

В підсумку картки перетворюються в чорно-білі зображення, де на чорному тлі білими точками виділялася форма цифр. Наступний етап

фільтрації об'єднує ці точки в одну безперервну фігуру. Ця дія виконується за допомогою математичної морфології поетапно:

Спочатку зображення очищається від дрібних шумів шляхом замикання з невеликим ядром (одиначна матриця 2×2). Після цього кола об'єднуються в безперервну фігуру шляхом замикання з одиничним ядром розміру 12×12 .

До отриманого зображення, для згладжування ребер цифр, застосовується медіанний фільтр.

Отримані зображення методами openCV стискаються до розміру 28×28 . Так як всі тестові картки мають квадратну форму, то стислі зображення не мають геометричних спотворень. Розмір стиснутого зображення оптимальний: на ньому зберігаються всі необхідні для розпізнавання ознаки тієї або іншої цифри. Збільшення розміру вхідного зображення тягне за собою нелінійне зростання складності навчання і не гарантує збільшення точності розпізнавання чисел.

4.5 Реалізація персептрона

Для реалізації персептрона використовувалося готове рішення – Фреймворк keras. Персептрон, за допомогою якого розпізнавалися відфільтровані тести, мав таку архітектуру:

- Вхідний шар 28×28 входів
- Прихований шар з 512 елементів, функція активації – relu

$$relu(x) = \max(\varepsilon, x), \varepsilon \geq 0. \quad (4.1)$$

- Вихідний шар з 10 елементів, функція активації – softmax

$$softmax(z)_i = \frac{z_i}{\sum_{k=0}^n z_k}, z = \{z_i\}, softmax(z) = \{softmax(z)_i\},$$

$$i = 0, n, relu(x) = \max(\varepsilon, x), \varepsilon \geq 0. \quad (4.2)$$

- В якості опції помилки використовується перехресна ентропія

$$H(t, o) = - \sum_k t_i \log(o_i). \quad (4.3)$$

Де t – вірна відповідь, o – результат нейронної мережі.

В якості алгоритму навчання використовується RMSProp – вдосконалений метод градієнтного спуску, що включає в себе метод бегущого середнього і адаптивність.

ВИСНОВКИ

У роботі були розглянуті методи представлення, розпізнавання і обробки зображень, розглянуті операції математичної морфології, реалізовано додаток, вирішальна тест Ішіхара. Тестування було проведено на згенерованому зашумленому наборі карт Ішіхара. Етап вибору колірних каналів, відділення цифр від фону і відновлення їх форми пройшов задовільно. Вся вибірка була розділена на тестову і навчальну в співвідношенні 600/800. В результаті навчання персептрона на навчальній вибірці, точність розпізнавання на тестових даних досягла 99%.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Чабан. Л.Н. Теория и алгоритмы распознавания образов. Учебное пособие. М.: МИИГАиК. 2004. 70с.
2. Прэтт У. Цифровая обработка изображений. М.: Мир, 1982. Т.2. — 928 с.
3. Алгоритмические основы построения растровой графики. URL <https://www.intuit.ru/studies/courses/993/163/lecture/4491> (дата звернення: 20.11.2020).
4. Форсайт Д., Понс Ж.. Компьютерное зрение. Современный подход. /изд. М.: Вильямс, 2004. — 928 с.
5. Jain, Anil K. (1989). Fundamentals of Digital Image Processing. New Jersey, United States of America: Prentice Hall. ISBN 0-13-336165-9. С 64–67.
6. O. Aslan, H. Cheng, D. Schuurmans, and X. Zhang. Convex two-layer modeling. In Proceedings of Neural Information Processing Systems (NIPS). 2013. 204 с.
7. J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and O'Shaughnessy. Research developments and directions in speech recognition and understanding. IEEE Signal Processing Magazine, 26(3):75–80, May 2009. 118 с.
8. Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 38:1798–1828, 2013. 89 с.
9. Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In Proceedings of Neural Information Processing Systems (NIPS). 2006. С. 328–329.
10. Y. Bengio, E. Thibodeau-Laufer, and J. Yosinski. Deep generative stochastic networks trainable by backprop. arXiv 1306:1091, 2013. also accepted to appear in Proceedings of International Conference on Machine Learning (ICML), 2014. 188 с.

11. Y. Cho and L. Saul. Kernel methods for deep learning. In Proceedings of Neural Information Processing Systems (NIPS) 2009. C. 342–350.
12. L. Deng. Computational models for speech production. In Computational Models of Speech Pattern Processing. Springer Verlag, 1999. C 371–372.
13. L. Deng, G. Tur, X. He, and D. Hakkani-Tur. Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In Proceedings of IEEE Workshop on Spoken Language Technologies. December 2012. 411 c.
14. G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006. 230 c.
15. Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: Fast feature extraction and SVM training. In Proceedings of Computer Vision and Pattern Recognition (CVPR). 2011. 320 c.
16. G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, July 2006. C. 504–507.
17. M. Ranzato, Y. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In Proceedings of Neural Information Processing Systems (NIPS). 2007. 112 c.
18. S. Rennie, H. Hershey, and P. Olsen. Single-channel multi-talker speech recognition — graphical modeling approaches. *IEEE Signal Processing Magazine*, 2010. C. 66–80.
19. O. Vinyals, Y. Jia, L. Deng, and T. Darrell. Learning with recursive perceptual representations. In Proceedings of Neural Information Processing Systems (NIPS). 2012. 131 c.
20. M. Wohlmayr, M. Stark, and F. Pernkopf. A probabilistic interaction model for multi-pitch tracking with factorial hidden markov model. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4), May 2011. 654 c.
21. M. Zeiler. Hierarchical convolutional deep learning in computer vision. Ph.D. Thesis, New York University, January 2014. C. 245- 247.