

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

факультет прикладних інформаційних технологій та електроінженерії

(повна назва факультету)

кафедра радіотехнічних систем

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Метод інтелектуального розпізнавання зображень для підвищення
ефективності систем машинного зору

Виконав(ла): студент(ка) 6 курсу, групи РАМ-62
спеціальності _____

172 Електронні комунікації та радіотехніка

(шифр і назва спеціальності)

(підпис) Брегін С.І.
(прізвище та ініціали)

Керівник _____
(підпис) Дедів І.Ю.
(прізвище та ініціали)

Нормоконтроль _____
(підпис) Хвостівська Л.В.
(прізвище та ініціали)

Завідувач кафедри _____
(підпис) Дунець В.Л.
(прізвище та ініціали)

Рецензент _____
(підпис) Тимків П.О.
(прізвище та ініціали)

Тернопіль
2024

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

[illegible]

КАЛЕНДАРНИЙ ПЛАН

(прізвище та ініціали)

АНОТАЦІЯ

Метод інтелектуального розпізнавання зображень для підвищення ефективності систем машинного зору // Кваліфікаційна робота магістра // Брегін Сергій Ігорович // ТНТУ ім. І.Пулюя, ФПТ // Тернопіль, 2024 // с. - 69, рис. - 21, дод. - 2, бібл. - 58.

Ключові слова: НЕЙРОМЕРЕЖА, ЗОБРАЖЕННЯ, КЛАСИФІКАТОР.

В роботі обґрунтовано метод інтелектуального розпізнавання зображень для підвищення ефективності систем машинного зору. Розглянуто технологію машинного навчання, як одного із підрозділів штучного інтелекту. Наведено підходи до організації класифікації із використанням машинного навчання. Показано, що для задач розпізнавання зображень або мови найбільш підходящим методом класифікації стає використання нейромереж. Розглянуто принципи функціонування нейромереж, зокрема із застосуванням глибокого навчання. Було розроблено алгоритм для розпізнавання зображень, що використовує ансамбль нейромереж із застосуванням двох типів додаткового класифікатора. Показано ефективність роботи таких мереж.

ANNOTATION

Intelligent image recognition method to improve of the machine vision systems efficiency // Master's qualification work // Bregin S.I. // TNTU, FPT // Ternopil, 2024 // p. - 69, fig. - 21, appl. - 2, bibl. - 58.

Key words: NEURAL NETWORK, IMAGE, CLASSIFIER.

The paper substantiates the method of intelligent image recognition to increase of the machine vision systems efficiency. The technology of machine learning is considered as one of the subdivisions of artificial intelligence. Approaches to organizing classification using machine learning are presented. It is shown that for image or language recognition tasks, the most suitable classification method is the use of neural networks. The principles of functioning of neural networks are considered, in particular with the use of deep learning. An algorithm for image recognition was developed that uses an ensemble of neural networks consisting of two multilayer perceptrons and 4 deep convolutional networks, and with the use of two types of additional classifier. The efficiency of such networks is shown.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА.....	10
1.1 Поняття машинного зору.....	10
1.2 Принцип функціонування машинного зору.....	12
1.3 Структура систем ML.....	16
1.4 Класи нейромереж.....	21
1.5 Архітектура нейромереж.....	23
1.6 Створення, ініціалізація та моделювання мережі.....	25
1.7 Персептрони.....	28
1.8 Аналіз поширених бібліотек.....	30
1.9 Висновки до розділу 1.....	32
РОЗДІЛ 2. ОСНОВНА ЧАСТИНА.....	33
2.1 Алгоритм класифікації зображень.....	33
2.2 Архітектура алгоритму.....	37
2.3 Висновки до розділу 2.....	37
РОЗДІЛ 3. НАУКОВО-ДОСЛІДНА ЧАСТИНА.....	38
3.1 Вихідні дані.....	38
3.2 Попередня обробка даних.....	39
3.3 Тестовий алгоритм розпізнавання зображень в Matlab.....	40
3.4 Результати окремих моделей.....	44
3.5 Висновки до розділу 3.....	49
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	51
4.1 Охорона праці.....	51
4.2 Безпека в надзвичайних ситуаціях.....	53
4.3 Висновки до розділу.....	55
ВИСНОВКИ.....	56

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
ДОДАТКИ	

ВСТУП

Актуальність. Сьогодні комп'ютерні системи мають доступ до великого обсягу зображень та джерел даних, отриманих зі смартфонів, дорожніх камер тощо. Хоча технологія візуальної обробки інформації існує вже деякий час, більшість процесу вимагала втручання людини, займала багато часу і була пов'язана з помилками.

Сьогодні прогрес у цій галузі разом із значним збільшенням обчислювальної потужності дозволив поліпшити як масштаб, так і точність обробки даних зображень. Системи такого зору, що працюють на базі ресурсів хмарних обчислень, тепер доступні для кожного. Будь-яка організація може використовувати цю технологію для перевірки особистості, модерації контенту, аналізу потокового відео, виявлення помилок та багато іншого.

Технологія автономних транспортних засобів використовує машинний зір для розпізнавання зображень у реальному часі та побудови 3d-карт за допомогою численних камер, встановлених на автономному транспорті. Він може аналізувати зображення та ідентифікувати учасників дорожнього руху, дорожні символи, пішоходів чи перешкоди.

У напіваавтономних транспортних засобах машинний зір використовує машинне навчання (ml) для моніторингу водія. наприклад, воно шукає ознаки розсіяності, втоми та сонливості залежно від положення голови водія, руху очей та руху верхньої частини тіла. якщо технологія вловлює певні попереджувальні знаки, вона сигналізує водію і таким чином знижує можливість дорожньо-транспортної пригоди.

Вроботі проводиться розроблення методу розпізнавання зображень, зокрема дорожніх символів, для підвищення ефективності систем машинного зору із використанням нейронних мереж.

Мета: розроблення методу розпізнавання зображень для підвищення ефективності систем машинного зору. **Задачі:**

- аналіз методів розпізнавання зображень, зокрема інтелектуалізованих;
- аналіз штучних нейромереж та їх застосування для завдання розпізнавання зображень;
- розробка алгоритму розпізнавання дорожніх символів для систем машинного зору;
- оцінювання точності роботи отриманого алгоритму.

Об'єкт дослідження: процес розпізнавання зображень.

Предмет дослідження: розпізнавання дорожніх символів для систем машинного зору.

Наукова новизна: обґрунтовано структуру нейромережі для розпізнавання дорожніх символів, що може бути застосована для систем машинного зору.

РОЗДІЛ 1

АНАЛІТИЧНА ЧАСТИНА

1.1 Поняття машинного зору

Комп'ютерні системи мають доступ до великого обсягу зображень та джерел даних, отриманих зі смартфонів, дорожніх камер тощо. Такі додатки застосовують AI та машинне навчання для точної обробки цих даних з метою ідентифікації об'єктів та розпізнавання осіб, а також для класифікації, рекомендацій, моніторингу та виявлення.

Хоча технологія візуальної обробки інформації існує вже деякий час, більшість процесу вимагала втручання людини, займала багато часу і була пов'язана з помилками. Наприклад, впровадження системи розпізнавання облич у минулому вимагало від розробників вручну позначати тисячі зображень точками із ключовими даними, такими як ширина перенісся та відстань між очима. Автоматизація цих завдань вимагає значних обчислювальних потужностей, оскільки дані зображень є неструктурованими і складними для впорядкування комп'ютерами. Таким чином, візуальні програми були дорогими і недоступними для більшості організацій.

Сьогодні прогрес у цій галузі разом із значним збільшенням обчислювальної потужності дозволив поліпшити як масштаб, так і точність обробки даних зображень. Системи такого зору, що працюють на базі ресурсів хмарних обчислень, тепер доступні для кожного. Будь-яка організація може використовувати цю технологію для перевірки особистості, модерації контенту, аналізу потокового відео, виявлення помилок та багато іншого.

Численні програми такого зору застосовуються у сфері розваг, бізнесу, охорони здоров'я, транспорту та в повсякденному житті. Розглянемо деякі приклади з практики

Охорона і безпека. Уряди та підприємства використовують машинний зір для підвищення безпеки активів та об'єктів. Наприклад, камери та датчики стежать за громадськими просторами, промисловими об'єктами та місцями підвищеної безпеки. Вони надсилають автоматичні попередження, якщо відбувається щось незвичайне, наприклад несанкціоноване проникнення в заборонену зону.

Так само машинний зір може підвищити особисту безпеку як вдома, так і на робочому місці. Наприклад, технологія розпізнавання дозволяє відстежувати безліч дезпекових проблем. До них відносяться домашні камери, що працюють в реальному часі і виявляють домашніх тварин, або камери, що працюють в реальному часі і виявляють відвідувачів. На робочому місці такий моніторинг включає носіння працівниками захисних засобів, надання інформації системам оповіщення або складання звітів.

Машинний зір дозволяє аналізувати зображення та отримувати метадані для бізнес-аналітики, створюючи можливості для отримання додаткового прибутку та підвищення операційної ефективності. Наприклад, він може:

- Здійснювати автоматичну ідентифікацію дефектів якості до того, як продукція залишить завод;
- Виявляти проблеми з технічним обслуговуванням та безпекою обладнання;
- Аналізувати зображення у соціальних мережах;
- Перевіряти особу співробітників за допомогою автоматичного розпізнавання осіб.

Охорона здоров'я - одна з провідних галузей, що використовують технології машинного зору. Зокрема, аналіз медичних знімків створює візуалізацію органів та тканин, допомагаючи медичним працівникам швидко та точно ставити діагнози, що призводить до покращення результатів лікування, приклад:

- Виявлення пухлин шляхом аналізу родимок та пошкоджень шкіри
- Автоматичний рентгенівський аналіз

- Виявлення симптомів за допомогою МРТ

Технологія автономних транспортних засобів використовує машинний зір для розпізнавання зображень у реальному часі та побудови 3d-карт за допомогою численних камер, встановлених на автономному транспорті. Він може аналізувати зображення та ідентифікувати учасників дорожнього руху, дорожні знаки, пішоходів чи перешкоди.

У напівавтономних транспортних засобах машинний зір використовує машинне навчання (ml) для моніторингу водія. Наприклад, воно шукає ознаки розсіяності, втоми та сонливості залежно від положення голови водія, руху очей та руху верхньої частини тіла. Якщо технологія вловлює певні попереджувальні знаки, вона сигналізує водію і таким чином знижує можливість дорожньо-транспортної пригоди.

Програми автоматизованого зору, що забезпечують підвищення продуктивності та зниження витрат завдяки інтелектуальній автоматизації, покращують загальне функціонування сільськогосподарського сектору. Супутникові знімки, а також кадри з БПЛА допомагають аналізувати широкі ділянки землі та вдосконалювати методи ведення сільського господарства. Програми такого зору автоматизують такі завдання, як моніторинг польових умов, виявлення захворювань сільськогосподарських культур, перевірку вологості ґрунту та прогнозування погоди та врожайності. Моніторинг тварин за допомогою машинного зору – ще одна ключова стратегія розумного сільського господарства.

1.2 Принцип функціонування машинного зору

Системи застосовують технологію штучного інтелекту (AI) для імітації можливостей людського мозку, які відповідають за розпізнавання та класифікацію об'єктів. Фахівці з інформатики навчають комп'ютери розпізнавати візуальні дані, вводячи величезну кількість інформації. Алгоритми машинного навчання (ml) виявляють загальні закономірності цих

зображеннях або відео і застосовують отримані знання для точної ідентифікації невідомих зображень. Наприклад, якщо комп'ютери оброблять мільйони зображень автомобілів, вони почнуть створювати ідентичні шаблони, які дозволять точно визначити автомобіль на зображенні. Машинний зір використовує такі технології.

Глибоке навчання – це тип машинного навчання, у якому використовуються нейронні мережі (НМ). НМ глибокого навчання складаються з безлічі верств програмних модулів, які називаються штучними нейронами, які працюють разом усередині комп'ютера. Вони використовують математичні розрахунки для автоматичної обробки різних аспектів даних зображення та поступово виробляють комбіноване розуміння зображення.

Згорткові нейронні мережі (CNN) використовують систему міток для категоризації візуальних даних та розуміння всього зображення. Вони аналізують зображення в пікселях і надають кожному пікселю мітку. Це значення вводиться, щоб виконати математичну операцію, яку називають згорткою, і зробити прогнози щодо зображення. Подібно до людини, яка намагається розпізнати об'єкт на відстані, CNN спочатку визначає контури та прості форми, а потім заповнює їх додатковими деталями, такими як колір, внутрішні форми та текстура. Нарешті, він повторює процес прогнозування протягом кількох ітерацій підвищення точності.

Рекурентні нейромережі (RNN) схожі на CNN, але можуть обробляти серію зображень для пошуку взаємозв'язків. У той час як CNN використовуються для аналізу окремих зображень, RNN можуть аналізувати відео та розуміти взаємозв'язки між зображеннями.

У чому різниця між машинним зором та обробкою зображень? Обробка зображень використовує алгоритми зміни зображень, включаючи підвищення різкості, згладжування, фільтрацію чи підвищення якості. Комп'ютерний зір відрізняється тим, що він не змінює зображення, а навпаки, осмислює те, що бачить, і виконує завдання, наприклад, маркування. У деяких випадках можна використовувати обробку зображень для зміни зображення, щоб система

машинного зору могла краще його розпізнати. В інших випадках ви використовуєте зір для ідентифікації зображень або частин зображення, а потім використовуєте обробку зображень для подальшої зміни зображення.

Розглянемо кілька прикладів завдань машинного зору, які можуть реалізувати організації.

Класифікація зображень дозволяє комп'ютерам побачити зображення і точно визначити, до якого класу воно належить. Машинний зір розуміє типи зображень і відповідно маркує їх, наприклад, дерева, літаки або будівлі.

Виявлення об'єктів — це завдання машинного зору виявлення і локалізації зображень. Програма використовує класифікацію для ідентифікації, сортування та організації зображень. Виявлення об'єктів використовується у промислових та виробничих процесах для управління автономними додатками та моніторингу виробничих ліній. Виробники камер для підключених будинків та постачальники послуг також використовують функцію виявлення об'єктів для обробки відео потоків з камер у реальному часі, щоб виявляти людей та об'єкти в дійсному часі та надавати попередження кінцевим користувачам.

Відстеження об'єктів використовує моделі глибокого навчання для ідентифікації та відстеження предметів, що входять до різних категорій. Ця технологія має кілька реальних застосувань у різних галузях. Першим елементом процесу відстеження об'єктів є виявлення; навколо об'єкта створюється рамка, що обмежує, йому присвоюється ідентифікатор об'єкта, і його можна відстежувати по кадрах. Наприклад, відстеження об'єктів можна використовувати для моніторингу дорожнього руху у міському середовищі, спостереження за людьми та медичної візуалізації.

Сегментування — це алгоритм машинного зору, який ідентифікує об'єкт шляхом поділу його зображень на різні області на основі видимих пікселів. Сегментування також спрощує зображення, наприклад розміщує форму або контур предмета, щоб визначити, що це таке. При цьому сегментування також розпізнає наявність кількох об'єктів на зображенні або кадрі.

Наприклад, якщо на зображенні є кішка та собака, можна використовувати сегментування для розпізнавання цих двох тварин. На відміну від виявлення об'єкта, що будує рамку навколо об'єкта, при сегментуванні відстежуються пікселі визначення форми об'єкта, що полегшує його аналіз і маркування.

Вилучення даних зображень на основі контенту - це застосування методів машинного зору, що дозволяють шукати конкретні цифрові зображення у великих базах даних. Він аналізує метадані, такі як теги, описи, мітки та ключові слова. Семантичний пошук використовує такі команди, як "знайти фотографії будівель", для отримання відповідного контенту.

AWS надає широкий і повний набір послуг AI/ML, підключених до набору джерел даних.

Для клієнтів, які використовують фреймворки та керують власною інфраструктурою, ми оптимізуємо версії найпопулярніших фреймворків глибокого навчання, включаючи PyTorch, MXNet та TensorFlow. AWS надає великий спеціалізований портфель обчислювальних та мережевих послуг, а також послуг ML інфраструктури зберігання даних з можливістю вибору процесорів та прискорювачів, щоб допомогти клієнтам з різним бюджетом задовольнити унікальні потреби у продуктивності.

Для клієнтів, які хочуть створити стандартне рішення в галузі машинного зору для свого бізнесу, AmazonSageMaker спрощує підготовку даних, побудову, навчання та розгортання ML-моделей за будь-якого випадку використання за допомогою повністю керованої інфраструктури, інструментів та робочих процесів, у тому числі рішень без коду для бізнес-аналітиків.

Для клієнтів, яким не вистачає навичок машинного навчання, яким потрібно прискорити виведення на ринок або які хочуть додати інтелект у існуючий процес або програму, AWS пропонує низку сервісів машинного зору на основі ML. Ці сервіси дозволяють легко додавати інтелектуальні дані до програм штучного інтелекту за допомогою попередньо навчених API. Amazon Rekognition автоматизує аналіз зображень та відео за допомогою ML та

аналізує мільйони зображень, прямих трансляцій та збережених відео за лічені секунди.

1.3 Структура систем ML

Машинний зір — це технологія та методи, які використовуються для забезпечення автоматичної перевірки та аналізу на основі зображень для таких застосувань, як автоматична перевірка, керування процесами та керування роботами, зазвичай у промисловості. Машинне бачення відноситься до багатьох технологій, програмних і апаратних продуктів, інтегрованих систем, дій, методів і досвіду. Машинний зір як дисципліну системної інженерії можна вважати відмінною від комп'ютерного зору, формою інформатики. Він намагається інтегрувати існуючі технології по-новому та застосувати їх для вирішення проблем реального світу. Термін є поширеним для цих функцій у середовищах промислової автоматизації, але також використовується для цих функцій у керуванні автомобілями в інших умовах.

Загальний процес машинного зору включає планування деталей вимог і проекту, а потім створення рішення. Під час виконання процес починається зі створення зображення, після чого відбувається автоматичний аналіз зображення та вилучення необхідної інформації.

Визначення терміну «машинний зір» відрізняються, але всі включають технологію та методи, які використовуються для автоматичного отримання інформації із зображення, на відміну від обробки зображень, де результатом є інше зображення. Витягнута інформація може бути простим сигналом хорошої/поганої частини або більш складним набором даних, таких як ідентичність, положення та орієнтація кожного об'єкта на зображенні. Інформацію можна використовувати для таких додатків, як автоматичний огляд і керування роботами та процесами в промисловості, для моніторингу безпеки та керування транспортними засобами. Ця сфера охоплює велику кількість технологій, програмних і апаратних продуктів, інтегрованих систем,

дій, методів і досвіду. Машинний зір є практично єдиним терміном, який використовується для цих функцій у додатках промислової автоматизації; цей термін є менш універсальним для цих функцій в інших середовищах, таких як безпека та навігація автомобіля. Машинний зір як дисципліну системної інженерії можна вважати відмінною від комп'ютерного зору, форми базової інформатики; машинне бачення намагається інтегрувати існуючі технології по-новому та застосувати їх для вирішення реальних проблем у спосіб, який відповідає вимогам промислової автоматизації та подібних областей застосування. Термін також використовується в ширшому значенні виставками та торговими групами, такими як Асоціація автоматизованої обробки зображень та Європейська асоціація машинного зору. Це ширше визначення також охоплює продукти та програми, які найчастіше роблять обробку зображень. Основним використанням MV є автоматична інспекція та керування промисловими роботами/процесами. Останнім часом терміни комп'ютерний зір і MV зблизилися більшою мірою.

Основним використанням машинного зору є автоматичний огляд і сортування на основі зображень, а також керування роботом.

Першим кроком у послідовності автоматичної перевірки є отримання зображення, як правило, за допомогою камер, лінз і освітлення, яке було розроблено для забезпечення диференціації, необхідної для подальшої обробки. Пакети програмного забезпечення MV і програми, розроблені в них, потім використовують різні методи обробки цифрових зображень для вилучення необхідної інформації та часто приймають рішення (такі як «пройшов/не пройшов») на основі вилученої інформації.

Компоненти автоматичної інспекційної системи зазвичай включають освітлення, камеру, процесор, програмне забезпечення та пристрої виведення.

Пристрій для обробки зображень (наприклад, камера) може бути відокремленою від основного блоку обробки зображень або об'єднаною із ним, у цьому випадку комбінація зазвичай називається розумною камерою або розумним датчиком. Включення повної функції обробки в той самий корпус,

що й камера, часто називають вбудованою обробкою. У разі відокремлення з'єднання може здійснюватися зі спеціалізованим проміжним обладнанням, спеціальним пристроєм обробки або захоплювачем кадрів у комп'ютері за допомогою аналогового або стандартизованого цифрового інтерфейсу. Реалізації MV також використовують цифрові камери, здатні до прямого підключення (без фреймграбера) до комп'ютера.

У той час як звичайне (2D) зображення найчастіше використовується в MV, альтернативи включають мультиспектральне зображення, гіперспектральне зображення, зображення різних інфрачервоних діапазонів, зображення лінійного сканування, 3D зображення поверхонь і рентгенівське зображення. Ключові відмінності в MV 2D візуалізації у видимому світлі: монохроматичне проти кольорового, частота кадрів, роздільна здатність, а також те, чи є процес зображення одночасним для всього зображення, що робить його придатним для рухомих процесів.

Хоча переважна більшість програм машинного бачення розв'язується за допомогою двовимірного зображення, програми машинного бачення, що використовують 3D-зображення, займають зростаючу нішу. Найпоширенішим методом 3D-зображення є тріангуляція на основі сканування, яка використовує рух продукту або зображення під час процесу створення зображення. Лазер проектується на поверхні об'єкта. У системі машинного зору це досягається скануючим рухом, або переміщенням заготовки, або переміщенням камери та лазерної системи зображення. Лінія переглядається камерою під іншим кутом; відхилення лінії означає варіації форми. Лінії з кількох сканувань збираються в карту глибини або хмару точок. Стереоскопічне бачення використовується в особливих випадках, пов'язаних із унікальними функціями, присутніми в обох ракурсах пари камер. Інші 3D-методи, що використовуються для машинного зору, — це час польоту та сітка. Одним із методів є системи на основі масиву сітки, які використовують систему псевдовипадкового структурованого світла, яка використовується в системі Microsoft Kinect.

Після того як зображення отримано, воно обробляється. Функції центральної обробки зазвичай виконуються центральним процесором, графічним процесором, FPGA або їх комбінацією. Глибоке навчання та логічний висновок висувають вищі вимоги до продуктивності обробки. Кілька етапів обробки зазвичай використовуються в послідовності, яка закінчується бажаним результатом. Типова послідовність може починатися з таких інструментів, як фільтри, які змінюють зображення, з подальшим виділенням об'єктів, потім вилученням (наприклад, вимірюваннями, зчитуванням кодів) даних з цих об'єктів, після чого ці дані передаються або порівнюються з цільовими значеннями. Методи обробки зображень машинного зору включають:

- реєстрацію: об'єднання суміжних 2D або 3D зображень
- фільтрування (наприклад, морфологічне фільтрування)

Порогове значення: Порогове значення починається з встановлення або визначення значення сірого кольору, яке буде корисним для наступних кроків. Потім це значення використовується для розділення частин зображення, а іноді й для перетворення кожної частини зображення просто на чорно-біле залежно від того, нижче чи вище це значення від градацій сірого.

Сегментація: Розбиття цифрового зображення на кілька сегментів для спрощення та/або зміни подання зображення на щось більш значуще та легше для аналізу.

Виявлення країв: пошук країв об'єкта.

Аналіз кольору: ідентифікуйте деталі, продукти та предмети за допомогою кольору, оцінюйте якість за кольором і виділяйте елементи за допомогою кольору.

Виявлення та виділення точок: перевірка зображення на наявність окремих точок з'єднаних пікселів (наприклад, чорна діра в сірому об'єкті) як орієнтири зображення.

Нейромережа/глибоке_навчання/обробка машинного навчання: зважене та самонавчальне прийняття багатоваріантних рішень. Приблизно в 2019 році

відбувається велике розширення цього, використовуючи глибоке навчання та машинне навчання для значного розширення можливостей машинного зору. Найпоширенішим результатом такої обробки є класифікація. Прикладами класифікації є ідентифікація об'єкта, класифікація ідентифікованих об'єктів.

Розпізнавання шаблонів, включаючи зіставлення шаблонів. Пошук, зіставлення та/або підрахунок конкретних шаблонів. Це може включати розташування об'єкта, який можна обертати, частково приховувати іншим об'єктом або змінювати розмір.

Зчитування штрих-коду, Data Matrix і "2D штрих-коду".

Оптичне розпізнавання символів: автоматичне читання тексту, наприклад серійних номерів.

Вимірювання/метрологія: вимірювання розмірів об'єктів (наприклад, у пікселях, дюймах або міліметрах).

Порівняння з цільовими значеннями для визначення результату «пройшов або не пройшов». Наприклад, під час перевірки коду або штрих-коду зчитане значення порівнюється із збереженим цільовим значенням. Для вимірювання порівнюється з належним значенням і допусками. Для перевірки буквено-цифрових кодів значення OCR порівнюється з належним або цільовим значенням. Для перевірки наявності плям вимірюваний розмір плям можна порівняти з максимальними значеннями, дозволеними стандартами якості.

Звичайним результатом автоматичних систем перевірки є рішення про проходження/відмову. Ці рішення можуть, у свою чергу, запускати механізми, які відхиляють невдалі елементи або подають сигнал тривоги. Інші поширені вихідні дані включають інформацію про положення та орієнтацію об'єкта для систем керування роботом. Крім того, вихідні типи включають дані числових вимірювань, дані, зчитані з кодів і символів, підрахунок і класифікацію об'єктів, відображення процесу або результатів, збережені зображення, сигнали тривоги від автоматизованих систем моніторингу простору MV і сигнали керування процесом. Це також включає в себе інтерфейси

користувача, інтерфейси для інтеграції багатокомпонентних систем і автоматизований обмін даними.

Термін глибоке навчання має різні значення, більшість з яких можна застосувати до методів, які використовуються в машинному зорі вже понад 20 років. Однак використання терміну в «машинному зорі» почалося наприкінці 2010-х років із появою можливості успішного застосування таких методів до цілих зображень у просторі промислового машинного зору. Звичайне машинне бачення зазвичай потребує фази «фізики» рішення для автоматичного контролю машинного зору для створення надійного простого розрізнення дефектів. Прикладом «простої» диференціації є те, що дефекти темні, а хороші частини продукту світлі. Поширеною причиною, чому деякі програми були неможливими, було те, що неможливо було досягти «простого»; глибоке навчання усуває цю вимогу, по суті, «бачить» об'єкт більше, як людина, що робить тепер можливим виконання цих автоматичних застосувань. Система вивчає велику кількість зображень під час фази навчання, а потім виконує перевірку під час використання, що називається «виведенням».

Машинний зір зазвичай надає роботу інформацію про місцезнаходження та орієнтацію, щоб він міг правильно охопити продукт. Ця можливість також використовується для керування рухом, який є простішим, ніж у роботів, наприклад, контролер руху за 1 або 2 осями. Загальний процес включає планування деталей вимог і проекту, а потім створення рішення. Багато етапів процесу такі ж, як і при автоматичному огляді, за винятком того, що в результаті зосереджується на наданні інформації про положення та орієнтацію.

1.4 Класи нейромереж

Нейромережі розрізняють за структурою мережі, особливостями моделі нейрона, особливостями навчання мережі.

За структурою нейромережі діляться на неповнозв'язні (або шаруваті) та повнозв'язкові.

По сигналах, що використовуються на входах і виходах, нейронні мережі можна розділити на аналогові та бінарні.

За моделюванням часу нейронні мережі поділяються на мережі з безперервним і дискретним часом. Для програмної реалізації застосовується зазвичай дискретний час.

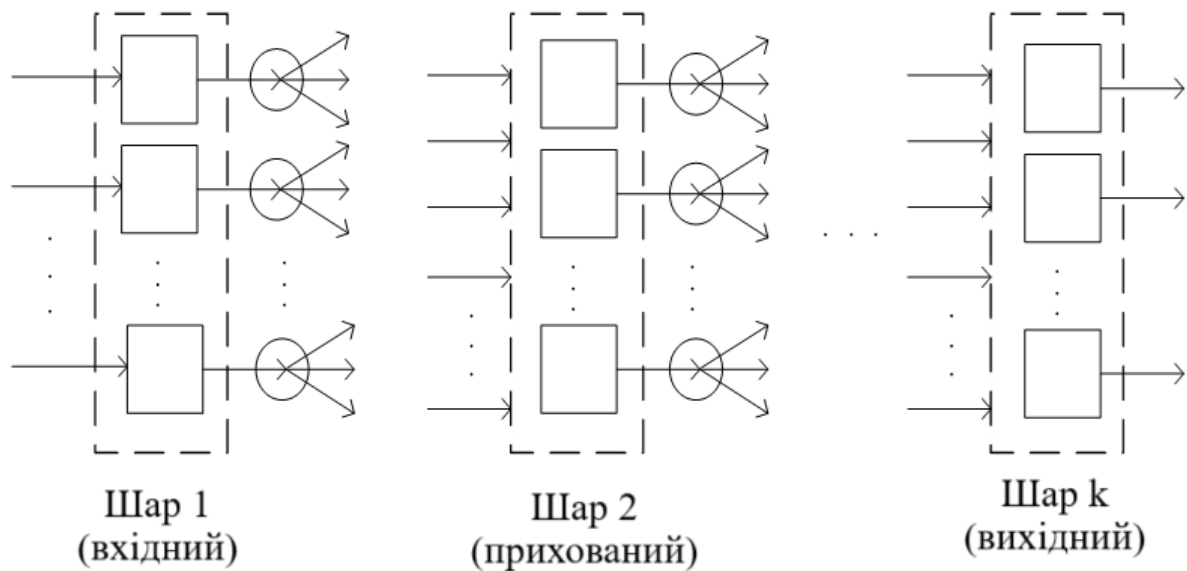


Рис. 1.1. Шарувата мережа

Залежно від функції активації нейронів виділяють однорідні та неоднорідні мережі. Однорідна, або гомогенна, нейромережа, це така, нейрони якої однаково перетворюють вихідний сигнал адаптивного суматора. Якщо ж функція активації залежить від декількох параметрів, значення яких змінюються від нейрона до нейрона, то мережу називають неоднорідною або гетерогенною.

Під станом нейронної мережі, що може змінюватися, зазвичай розуміється:

- ваги синапсів нейронів;

- встановлення нових нейронних взаємозв'язків (властивість біологічних нейронів встановлювати нові зв'язки та ліквідувати старі називається пластичністю).

1.5 Архітектура нейромереж

Архітектура мережі складається з опису того, скільки шарів має мережа, числа нейронів, виду функції активації кожного шару та інформації про з'єднання шарів. Архітектура мережі залежить від того конкретного завдання, яке має вирішувати мережа.

Робота мережі полягає у обчисленні виходів мережі на основі відомих входів з метою формування бажаного відображення вхід/вихід. Конкретне завдання визначає кількість входів та кількість виходів мережі. Крім числа нейронів у вихідному шарі мережі, для проектувальника важлива число нейронів шару. Більше число нейронів прихованих шарів забезпечує більш потужну мережу. Якщо має бути реалізовано лінійне відображення слід використовувати нейрони з лінійними функціями активації.

У цьому випадку лінійні нейронні мережі неспроможні формувати нелінійні відображення. Використання нелінійних функцій активації дозволяє настроїти нейронну мережу на реалізацію нелінійних зв'язків вхід/вихід.

Мережі зі зміщенням дозволяють формувати складніші зв'язки між входами та виходами, ніж мережі без зміщення. Наприклад, нейрон без зміщення, коли всі входи нульові, завжди задаватиме вхід функції активації рівним нулю, проте нейрон зі зміщенням може бути навчений так, щоб за тих же умов встановити вхід функції активації довільної форми.

У багатошарових мережах часто використовуються нелінійні сигмоїдні функції активації типу логістичної або гіперболічного тангенсу (рис. 1.2).

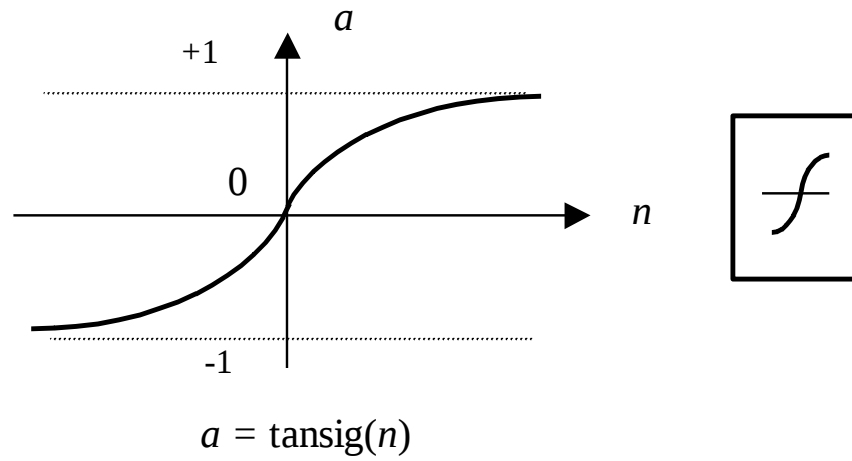


Рис. 1.2. Активуючі функції

Якщо останній шар багатошарової мережі використовує такі функції активації, то виходи мережі будуть обмежені.

Одношарова мережа з нейронами S з функціями активації logsig, що має R входів, показана на рис. 1.3.

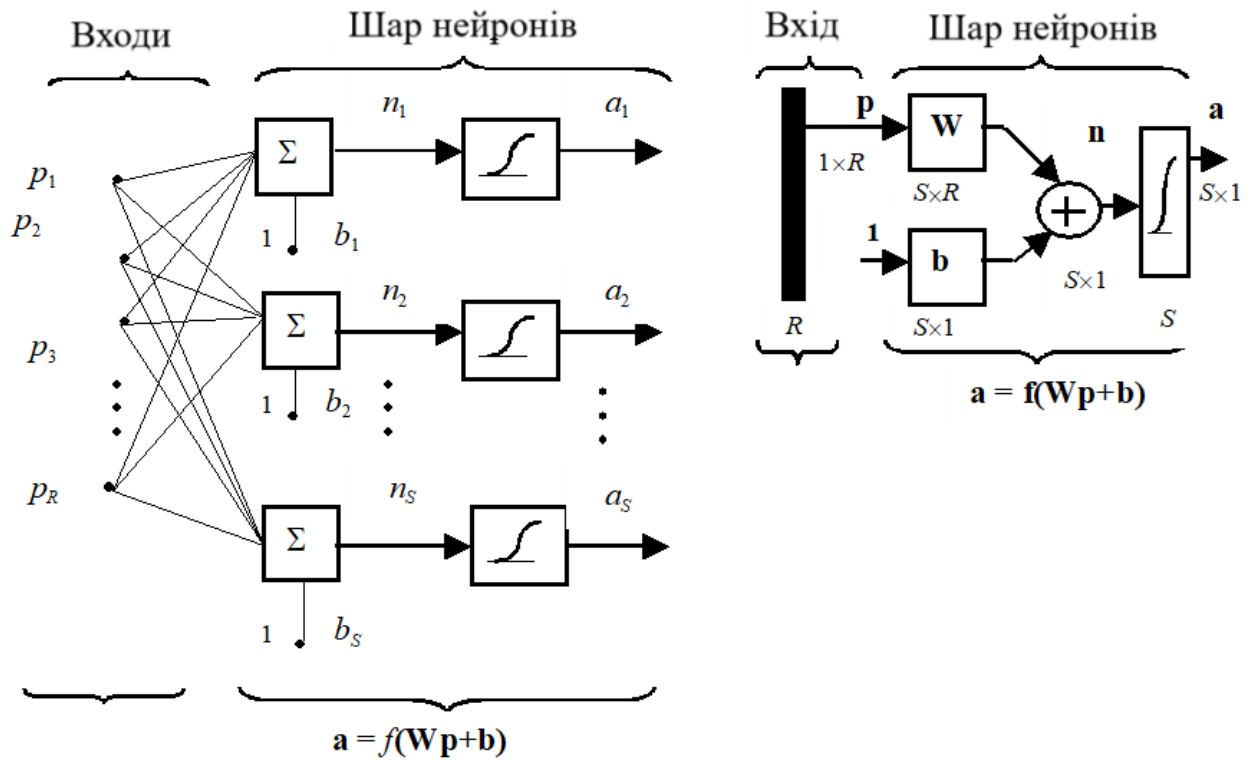


Рис. 1.3. Вигляд мережі

Для пояснення позначень у багатошарових нейронних мережах варто вивчити двошарову мережу, показану на рис. 1.4.

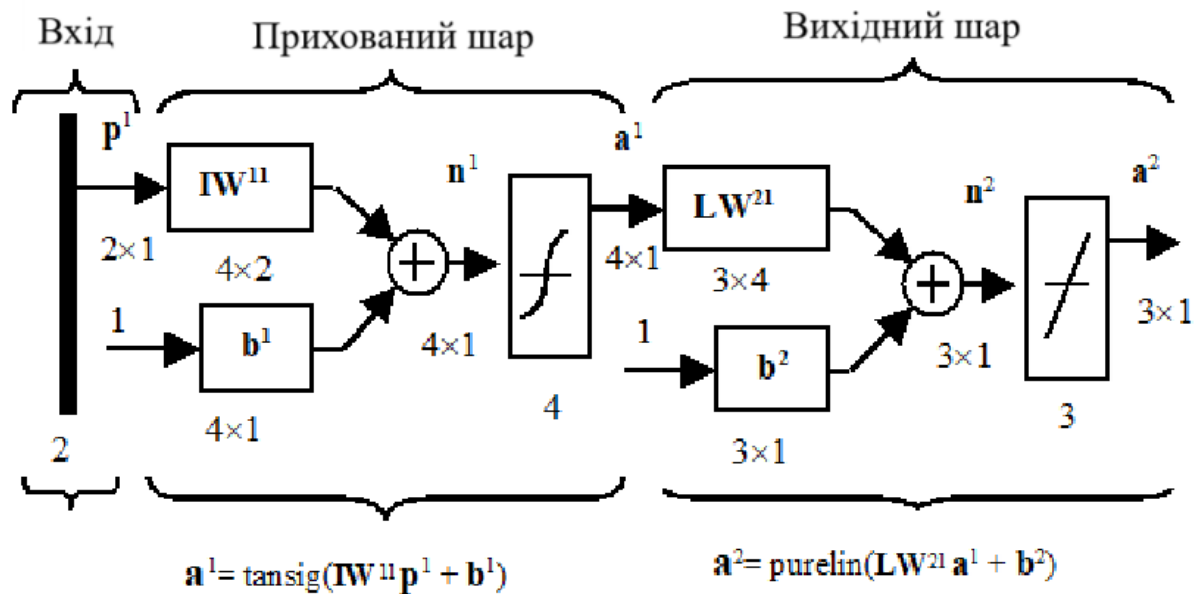


Рис. 1.4. Вигляд мережі

Така мережа застосовна для апроксимації функцій. Вона може точно відтворити довільну функцію, якщо задати достатню кількість нейронів прихованого шару.

1.6 Створення, ініціалізація та моделювання мережі

Формування архітектури мережі

Перший етап роботи з нейромережами – це створення моделі мережі. Для створення мереж із прямою передачею сигналу в NNT призначена функція `newff`. Вона має 4 вхідні аргументи та 1 вихідний аргумент – об'єкт класу `network`. Перший вхідний аргумент – це масив розміру $R2$, що містить допустимі межі значень (мінімальне та максимальне) для кожного з R елементів вектора входу; другий – масив для задання кількості нейронів кожного шару; третій – масив ядер, що містить імена функцій активації для кожного шару; четвертий – ім'я функції навчання.

Наприклад, наступний оператор створює мережу із прямою передачею сигналу:

```
net = newff([-1 2; 0 5],[3,1],{'tansig','purelin'},'traingd');
```

Ця мережа використовує 1 вектор входу з двома елементами, що мають допустимі межі значень $[-12]$ і $[05]$; має 2 шари з 3-ма нейронами в 1-му шарі та одним нейроном у 2-му шарі; використовувані функції активації: *tansig* – у першому шарі, *purelin* – у другому шарі; використовується функція навчання – *traingd*.

М-функція *newff* не тільки створює архітектуру мережі, а й ініціалізує її ваги та зміщення, готуючи нейромережу до навчання. Проте існують ситуації, коли потрібна спеціальна процедура ініціалізації мережі.

Статичні мережі. Статична нейронна мережа характеризується тим, що в її складі немає елементів запізнення і зворотних зв'язків. Розглянемо мережу, показану на рис.1.5.

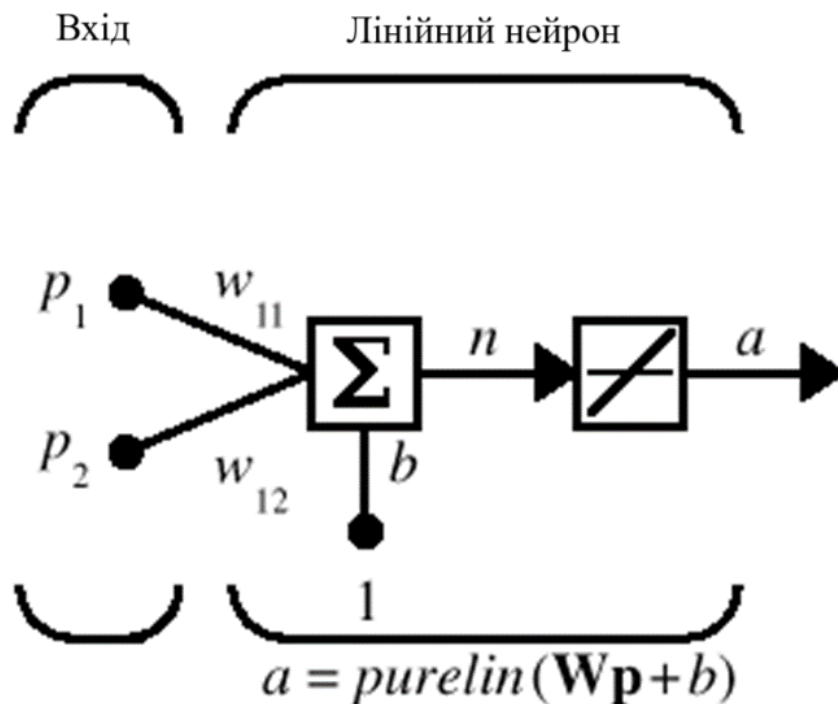


Рис. 1.5. Мережа

Для задання такої мережі призначена М-функція `newlin` із `NeuralNetworkToolbox`, яка вимагає вказати крайні значення для кожного з елементів входу; у разі вони рівні -1 і 1 відповідно, і навіть кількість верств, у разі 1 .

Припустимо, що на мережу подається така послідовність із чотирьох векторів входу:

$$\left\{ \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}.$$

Оскільки мережа статична, можна перегрупувати цю послідовність у наступний числовий масив:

$$P = [-1001; 0 -1 1 -1];$$

Змоделюємо мережу:

```
A = sim(net,P) % Моделювання мережі net із вектором входу P та виходом A
```

```
A = -1 -2 2 -1
```

Результат слід інтерпретувати так. На вхід мережі подається послідовність чотирьох сигналів, і вона генерує вектор виходу із чотирьох елементів. Результат був той самий, якби було 4 однакові мережі, що функціонують паралельно, і на кожную мережу було подано один із векторів входу і генерувався один із виходів.

Динамічні мережі. Коли мережа містить лінії затримки, вхід мережі слід розглядати як послідовність векторів, що подаються на мережу у певні моменти часу.

1.7 Персептрони

Персептрон – це одношарова нейромережа. Принцип його роботи такий:

1. Задається та обнулюється матриця ваг.
2. Навчальний образ, представлений вхідним вектором X (з 0 та 1) множиться на матрицю ваг. В результаті виходить вектор NET, розмірність якого дорівнює кількості нейронів.
3. Будуємо на основі отриманого добутку вихідний вектор (такої ж розмірності) за формулою:

$$y_i = F(\text{NET}_i),$$

де F - порогова функція. У нашому випадку це лінійна функція

$$F(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}.$$

1. Порівнюємо вектор Y , що покомпонентно вийшов, з правильною відповіддю. Під відповіддю мається на увазі вектор такої ж розмірності, як і вихідний. Його треба заздалегідь підготувати, і, зрозуміло, для кожного класу це має бути свій унікальний вектор.

2. При цьому порівнянні ми виконуємо такі дії (для кожної компоненти векторів):

- якщо j -та компонента результату більша за j -ту компоненту відповіді, ми віднімаємо вхідний вектор з j -того стовпця матриці ваг, тобто стовпця j -того нейрона (щоб при подальшому множенні вхідного вектора на матрицю скалярний добуток виявився меншим);

- якщо j -та компонента результату менша за j -ту компоненту відповіді, ми додаємо вхідний вектор до j -того стовпця матриці ваг.

Таким чином, вхідний вектор коригує матрицю ваги під свої значення.

3. Так само персептрон навчається на другому навчальному образі, третьому, ..., k -тому.

4. Якщо хоча б один навчальний образ змінив матрицю ваги, повторюємо все ще раз з кроку 2. Існує теорема, що якщо класи навчальних

образів розділяються, то рано чи пізно матриця перестане змінюватися (навчиться).

Одна ітерація навчання персептрона одним образом представлена на рис. 1.6:

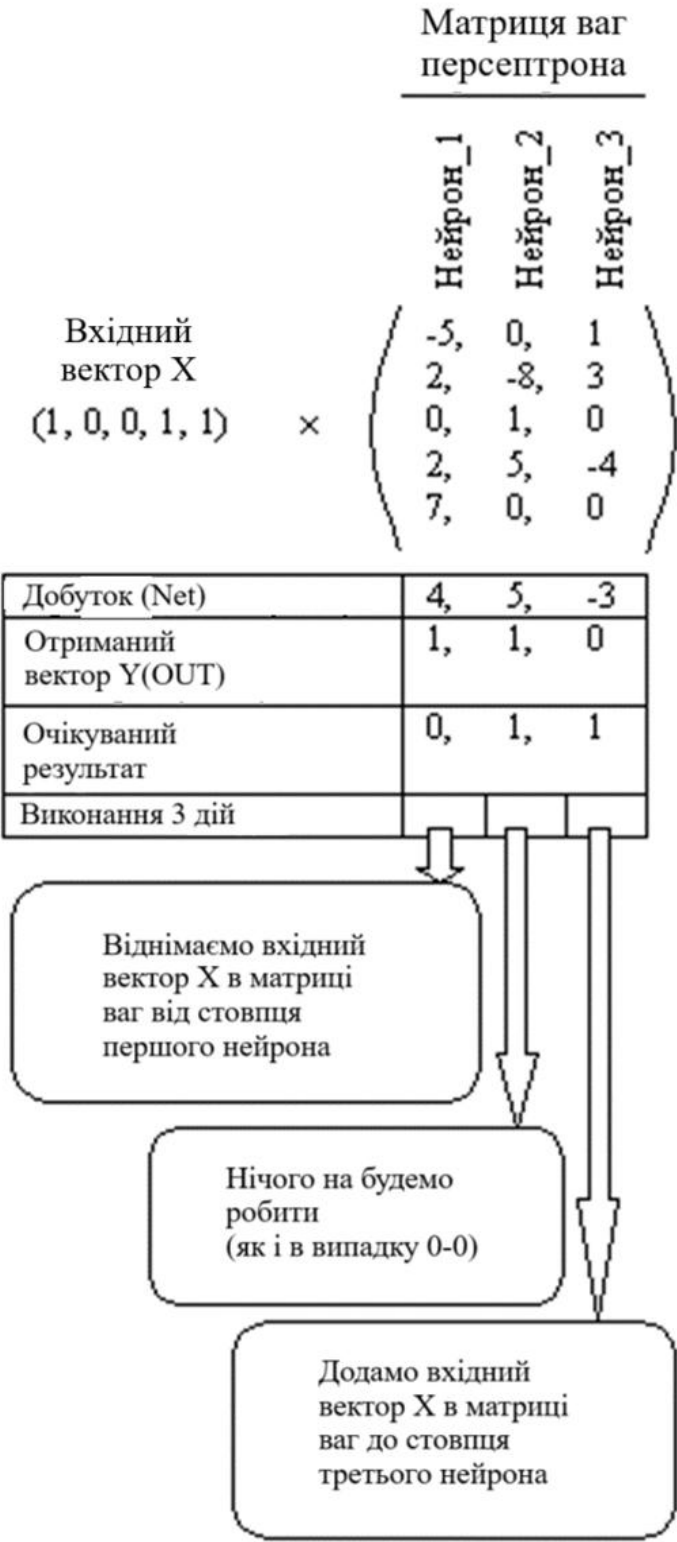


Рис. 1.6. Ітерація у персептроні

Модель Росенблата навчалася з допомогою методу корекції помилок, в якому ваги залишаються незмінними до того часу, поки вихідне значення персептрона коректне, у разі помилки вага змінюється на 1 убік, зворотну знаку помилки.

1.8 Аналіз поширених бібліотек

Для реалізації алгоритмів роботи з нейромережами в системі, що розробляється, застосовано існуючу бібліотеку. Спочатку проведено аналіз існуючого рішення програмного забезпечення реалізації алгоритмів deep learning, і за підсумками даного аналізу було здійснено вибір. Аналіз існуючих рішень складався з двох фаз: теоретичної та практичної.

У ході теоретичної фази було розглянуто ряд бібліотек:

- Deeperlearning4j – бібліотека для реалізації нейромереж та алгоритмів глибокого навчання. Бібліотека розвивається та підтримується компанією Skymin, яка також надає комерційну підтримку для цієї бібліотеки. У середині цієї бібліотеки використовується бібліотека для швидкої роботи з масивами ND4J розробки тієї ж компанії. Deeperlearning4j працює із значною кількістю мереж, серед них багат шаровий персептрон, згорткові мережі тощо.

- Torch – дозволяє дослідникам для роботи з нею використовувати набагато зручнішу скриптову мову Lua.

- Caffe – бібліотека, яка сконцентрована на ефективній реалізації алгоритмів глибокого навчання, що розробляється в першу чергу Berkley Vision and Learning Center, проте, як і всі попередні, має відкритий вихідний код. Бібліотека реалізована мовою C та працює з Python і Matlab. Підтримує повнозв'язкові та згорткові мережі, дозволяє описувати мережі у форматі набору шарів, у форматі .prototxt, підтримує обчислення на GPU. Плюсом є велика кількість передбачених моделей та прикладів, що у поєднанні з іншими

характеристиками робить бібліотеку найпростішою для старту роботи серед перерахованих вище.

Deeplearning4j є найбільш проблематичною в установці. Бібліотека Theano також є досить складною в установці та налаштуванні, проте для цієї бібліотеки існує коректна документація та приклад робочого коду. Тому, незважаючи на потенційно набагато ширші можливості цієї бібліотеки порівняно з Caffe, для проведення даного дослідження було вирішено зупинитися саме на останній як найбільш відповідній поставленим завданням.

Caffe має спрощений інтерфейс, дозволяючи легко конфігурувати та навчати нейронні мережі. Для роботи з бібліотекою робиться опис мережі у prototxt (protocol buffer definition file – мова опису даних, створена компанією Google), яка дещо схожа на формат JSON, добре структурована і зрозуміла для людини. Опис мережі є по суті послідовним описом кожного з її шарів. Як вхідні дані бібліотека може працювати з базою даних (leveldb або lmdb), in-memory даними, HDF5 файлами та зображеннями. Також є можливість використовувати для цілей розробки та тестування спеціальний вид даних, що зветься DummyData.

Бібліотека підтримує створення шарів наступних типів: InnerProduct (повнозв'язковий шар), Splitting (перетворює дані для передачі відразу на кілька вихідних шарів), Flattening (перетворює дані з багатовимірної матриці на вектор), Reshape (дозволяє змінити розмірність даних), Concatenation (перетворює дані з кількох вхідних шарів в один вихідний), Slicing та ще кілька інших. Для згорткових мереж підтримуються також спеціальні типи шарів – Convolution (шар згортки), Pooling (шар сабсемблінга) і Local Response Normalization. Також підтримуються ряд видів функції втрат, що застосовуються при навчанні мережі та функцій активації нейронів – які також конфігуруються як окремі шари мережі.

Таким чином, мережа описується декларативно у достатній простій формі. Також для роботи бібліотеки з використанням стандартних скриптів потрібно створити файл solver.prototxt, в якому описується конфігурація

навчання мережі – кількість ітерацій для навчання, learning rate, платформа для обчислень – cpu або gpu і т.д.

Навчання моделі може бути реалізовано із застосуванням вбудованих скриптів (після їх доопрацювання під поточне завдання) або вручну через написання коду з використанням python або Matlab, що надається apі мовою. При цьому існують скрипти, що дозволяють не тільки виконати навчання мережі, але також, наприклад, створити базу даних на основі переліку зображень – при цьому зображення перед додаванням до бази даних будуть приведені до фіксованого розміру і нормалізовані. Скрипти, за допомогою яких відбувається навчання, також інкапсулюють деякі допоміжні дії - наприклад, проводять оцінку поточної точності моделі через деяку кількість ітерацій і зберігають поточний стан навченої моделі файл снапшота. Використання файлів снапшота дозволяє надалі продовжити навчання моделі замість того, щоб починати спочатку, якщо виникає така необхідність, а також після деякої кількості ітерацій змінити конфігурацію моделі – наприклад, додати новий шар – і при цьому ваги вже навчених раніше шарів збережуть свої значення, що дозволяє реалізувати описаний раніше механізм пошарового навчання.

1.9 Висновки до розділу 1

Розглянуто технологію машинного бачення та машинного навчання, як одного із підрозділів штучного інтелекту. Наведено підходи до організації класифікації із використанням машинного навчання. Показано, що для задач розпізнавання зображень або мови найбільш підходящим методом класифікації стає використання нейромереж. Розглянуто принципи функціонування нейромереж, зокрема із застосуванням глибокого навчання.

Показано задачі класифікації зображень, та бібліотеки, що можуть для цього застосовуватись.

РОЗДІЛ 2

ОСНОВНА ЧАСТИНА

2.1 Алгоритм класифікації зображень

У ході вивчення теоретичного матеріалу було сформовано наступний набір ідей, які мають бути втілені у підсумковому алгоритмі:

- Використання глибоких згорткових нейромереж. Згорткові мережі стабільно показують найкращі результати в розпізнаванні зображень, у тому числі дорожніх знаків, тому їх використання в алгоритмі, що розробляється, виглядає логічним.

- Використання багат шарових персептронів. Незважаючи на загалом велику ефективність згорткових мереж, існують типи зображень, для яких багат шаровий персептрон показує кращі результати, тому було прийнято рішення використати також і даний алгоритм

- Комбінування результатів кількох моделей за допомогою додаткового класифікатора. Оскільки було вирішено використовувати щонайменше два типи нейронних мереж, потрібен спосіб формувати деякий загальний результат класифікації з урахуванням результатів кожної з них. Для цього планується використовувати додатковий класифікатор, не пов'язаний із нейронними мережами, вхідними значеннями для якого є результати класифікації кожної з мереж, а вихідним – підсумковий прогнозований клас зображення

- Застосування додаткових перетворень до вхідних даних. Для підвищення придатності вхідних зображень до розпізнавання та, відповідно, поліпшення результативності роботи класифікатора, до вхідних даних має бути застосовано кілька типів перетворень, при цьому результати кожного з них повинні бути оброблені окремою мережею, навченою для розпізнавання зображень з цим типом перетворення.

На основі всіх перерахованих вище ідей була сформована наступна концепція класифікатора зображень. Класифікатор являє собою ансамбль з 6 нейронних мереж, що функціонують незалежно: 2 багат шарові персептрони і 4 згорткові мережі. При цьому мережі одного типу відрізняються між собою типом перетворення, застосованим до вхідних даних. Дані на вході масштабуються таким чином, щоб на вході кожної мережі завжди виявлялися дані одного розміру, при цьому дані розміри можуть відрізнятися для різних мереж. Для агрегації результатів всіх мереж використовується додатковий класичний класифікатор, в якості якого було використано 2 варіанти: алгоритм J48, заснований на дереві рішень, та алгоритм kStar, що є “лінивим” класифікатором. Перетворення, що використовуються у класифікаторі:

- Бінаризація – зображення замінюється новим, з пікселів лише чорного та білого кольорів. Для виконання бінаризації використовується метод адаптивного порогового значення (adaptive thresholding). Для кожного пікселя зображення обчислюється середнє значення деякого його околу пікселів (передбачається, що зображення містить тільки відтінки сірого, для цього вихідні зображення були попередньо перетворені відповідним чином), і потім на основі обчисленого середнього значення визначається, повинен піксель вважатися чорним чи білим.

- Вирівнювання гістограми (histogram equalization) – суть методу полягає у застосуванні до гістограми зображення деякої функції, такої, щоб значення на результуючій діаграмі були розподілені максимально рівномірно. При цьому цільова функція обчислюється з урахуванням функції розподілу інтенсивності кольорів у вихідному зображенні. Приклад застосування подібної функції до гістограми зображення представлений на рис.2.1. Цей метод може застосовуватися як для чорно-білих, так і кольорових зображень – окремо для кожного компонента кольору.

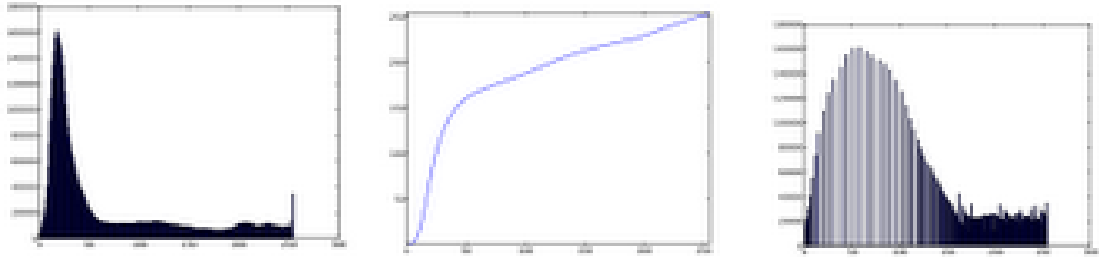


Рис. 2.1. результати застосування вирівнювання діаграми зображення

• Підсилення контрасту – для кожного пікселя зображення знаходиться локальний мінімум і максимум в деякому його околі і потім цей піксель замінюється локальним максимумом, якщо його вихідне значення ближче до максимуму або локальним мінімумом в іншому випадку. Застосовується до чорно-білих зображень.

Схематично загальна схема отриманого класифікатора представлена на рис. 2.2:

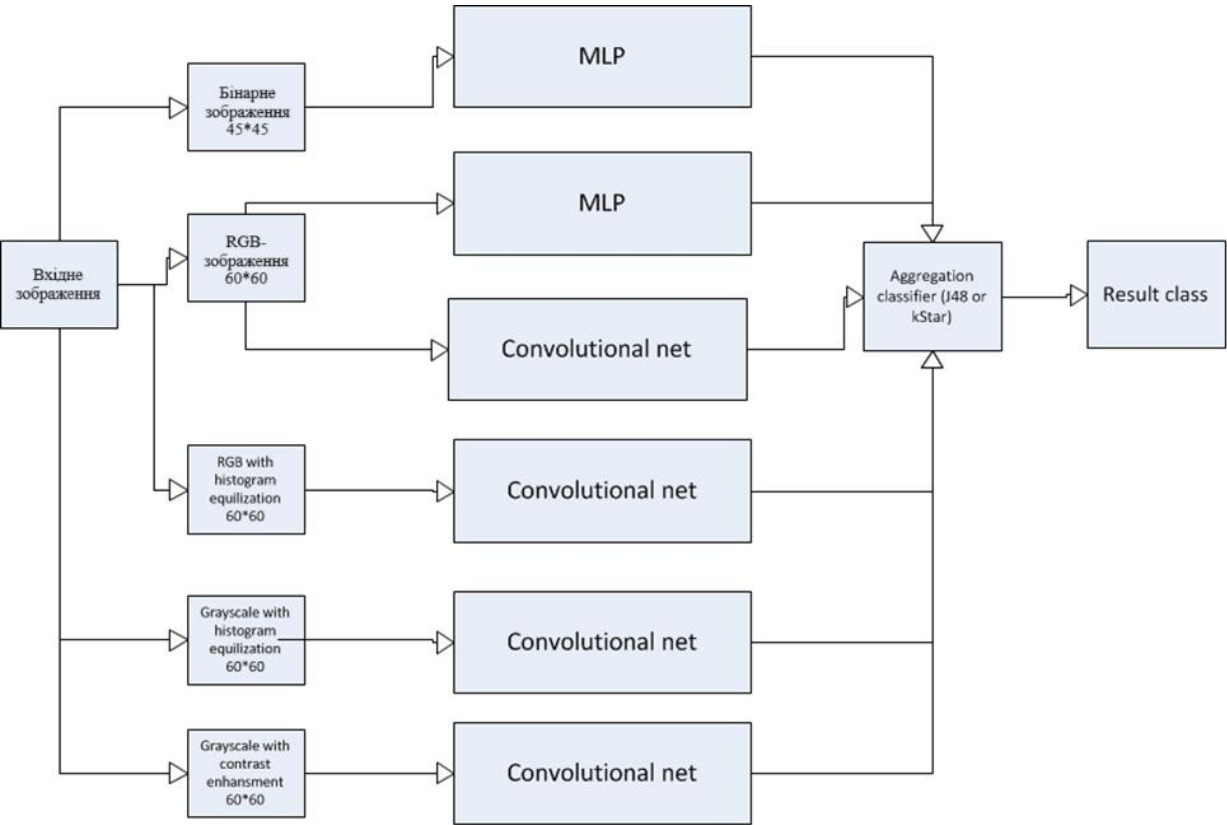


Рис. 2.2. Підсумкова схема класифікатора

Для реалізації частини моделі, що відповідає за перетворення вхідних даних та нейронні мережі, використовується мова Python і бібліотека Caffe.

Обидва багатошарові персептрони містять 4 приховані шари, і в цілому їх конфігурація описується наступним чином:

- Вхідний шар
- Шар 1, 1500 нейронів
- Шар 2, 500 нейронів
- Шар 3, 300 нейронів
- Шар 4, 150 нейронів
- Вихідний шар

Що стосується згорткових мереж, то за основу їхньої архітектури була взята відома мережа LeNet, розроблена для класифікації зображень з датасету ImageNet. При цьому, мережа була модифікована.

- 3 шари згортки з розмірами ядра 9, 3 та 3 відповідно
- 3 шари сабсемплінгу
- 3 пов'язані шари розмірами 100, 100 і 43 нейрона

Схема цієї мережі представлена на рис. 2.3.

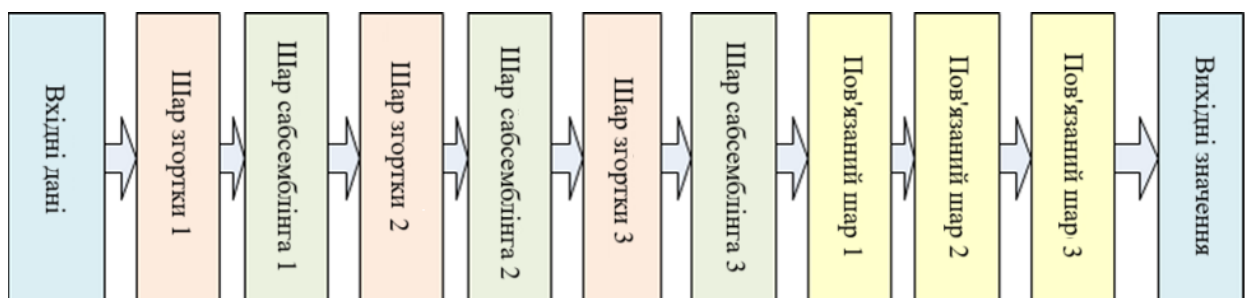


Рис. 2.3. Схема згорткової мережі

Кожна з нейронних мереж, що формують модель, навчається окремо. Після навчання нейронних мереж спеціальний скрипт мовою Python для кожної з мереж для кожного із зображень навчальної множини отримує результат класифікації у вигляді списку ймовірностей кожного з класів, вибирає два найбільш ймовірні класи і записує отримані значення разом із

реальним значенням класу зображення у файл. Отриманий файл потім передається як навчальна множина класифікатору (J48 і kStar), реалізованому в бібліотеці Weka. Відповідно, подальша класифікація проводиться з використанням цієї бібліотеки.

2.2 Архітектура алгоритму

Модулі нейронних мереж та класифікації вже містять навчені моделі. При цьому для навчання моделей надається інтерфейс адміністратора, який є по суті набір скриптів мовою python для навчання нейронних мереж та консольну утиліту мовою Java для навчання підсумкового класифікатора. Передбачається, що ці інструменти не повинні використовуватися часто або непрофесійними користувачами.

Відкритим залишається питання практичних результатів роботи класифікатора, що використовується в даному алгоритмі.

2.3 Висновки до розділу 2

На основі проведеного огляду для розроблення алгоритму класифікації зображень передбачено використання глибоких згорткових нейромереж, багатошарових персептронів, комбінування результатів кількох моделей за допомогою додаткового класифікатора, застосування додаткових перетворень до вхідних даних. була сформована наступна концепція класифікатора зображень. Класифікатор являє собою ансамбль з 6 нейронних мереж, що функціонують незалежно: 2 багатошарові персептрони і 4 згорткові мережі.

РОЗДІЛ 3

НАУКОВО-ДОСЛІДНА ЧАСТИНА

3.1 Вихідні дані

Як вхідні дані в даному дослідженні був використаний вже згаданий раніше датасет GTSRB. Бібліотека має більше 50000 різнокласових зображень. При цьому кількість зображень, що належать різним класам, є різною (рис. 3.1).

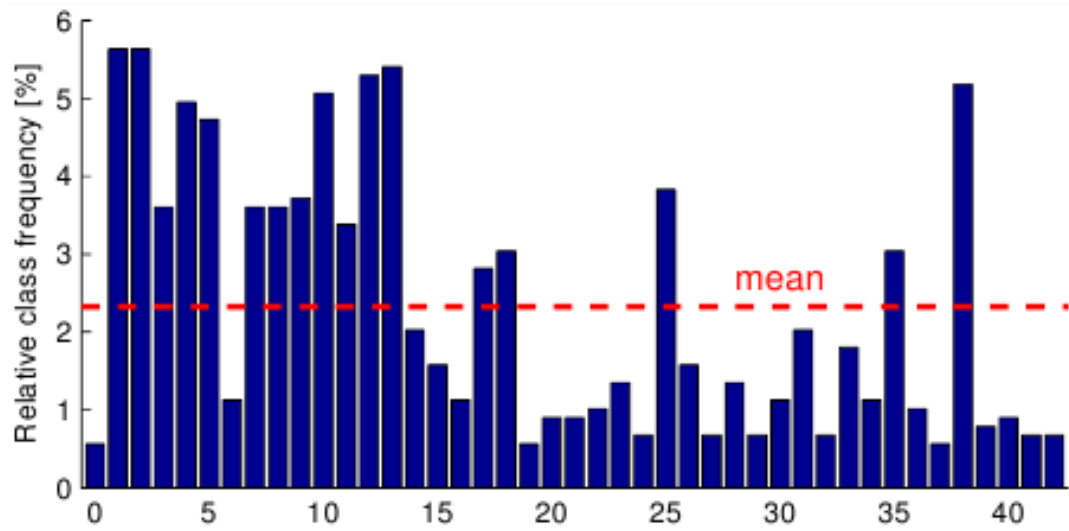


Рис. 3.1. Розподіл кількості зображень за класами

Розміри вхідних зображень також відрізняються. Загальний розподіл розмірів зображень представлено на рис. 3.3.

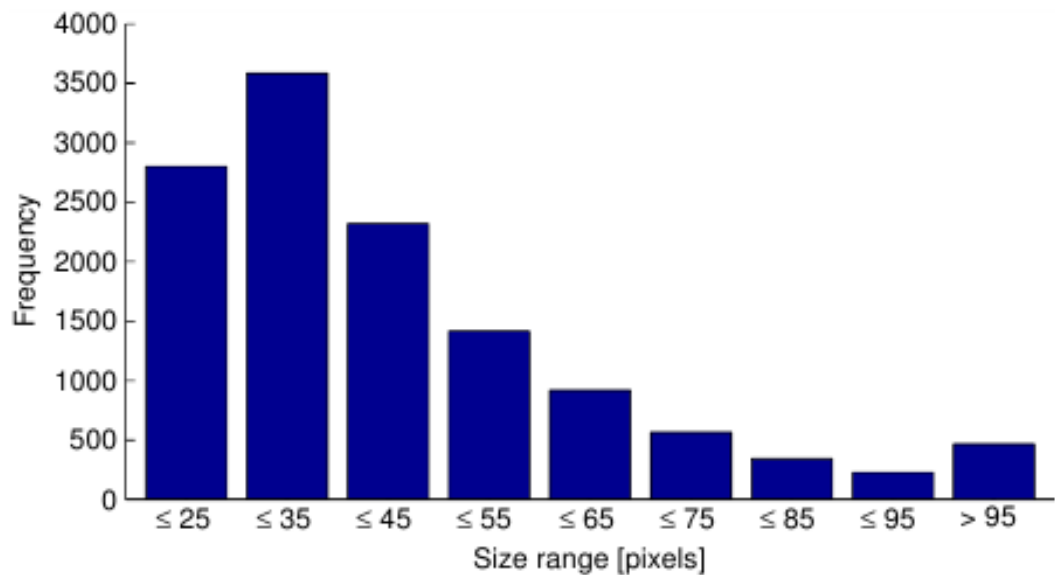


Рис. 3.3. Розподіл розмірів зображень

3.2 Попередня обробка даних

Перед початком роботи вихідні дані були відповідним чином підготовлені - перетворені з формату PPM у формат JPEG, з яким вмiє працювати бібліотека Caffe. Даліше зображення були випадково розбиті на навчальну та тестову множини у співвідношенні 80:20%, а також масштабовані. В алгоритмі класифікації використовуються зображення двох розмірів – 45x45 (для навчання багат шарового перспетрона на бінаризованих даних) та 60x60, тому для кожного зображення навчальної та тестової множини були створені екземпляри цих двох розмірів. Також до кожного з зображень були застосовані зазначені раніше перетворення (бінаризація, нормалізація гістограми, підвищення контрастності), і вже отримані зображення були збережені в базі даних LMDB (Lightning Memory-Mapped Database). Такий спосіб зберігання даних забезпечує найбільш швидку та зручну роботу бібліотеки Caffe. Для перетворення зображень застосовувалися бібліотеки Python Imaging Library (PIL) та scikit-image. Приклади отриманих після проведення кожного з перетворень зображень представлені на рис. 3.4.

Збережені в базі даних зображення використовувалися надалі для безпосереднього навчання нейронних мереж.



Рис. 3.4. Результати застосування перетворень до зображення

Що стосується навчання нейронних мереж, то кожна з мереж була навчена окремо і результати її роботи були оцінені, а потім було побудовано та навчено підсумковий класифікатор. Спочатку створена найпростіша мережа, що є персептроном з одним прихованим шаром. Розгляд цієї мережі мав дві мети – вивчення роботи з бібліотекою Caffe на простому прикладі формування деякого бенчмарку для більш предметної оцінки результатів роботи інших мереж в порівнянні з нею

3.3 Тестовий алгоритм розпізнавання зображень в Matlab

У системі Matlab є інструмент NNtool, що має графічний інтерфейс користувача, який істотно полегшує завдання моделювання нейронних мереж.

Скрипт prprob задає матрицю X із 26 стовпцями, один для кожної літери алфавіту. Кожен стовець має 35 значень, які можуть бути або 1 або 0. Кожен стовець має 35 значень і задає 5×7 бітовий масив букви.

Матриця T є матрицею ідентичності 26×26 , яка відображає ці 26 вхідних векторів у ці 26 класів.

```
[X, T] = prprob;
```

Тут A , перша літера, побудована з квадратів-пікселів.

```
plotchar(X(:,1))
```

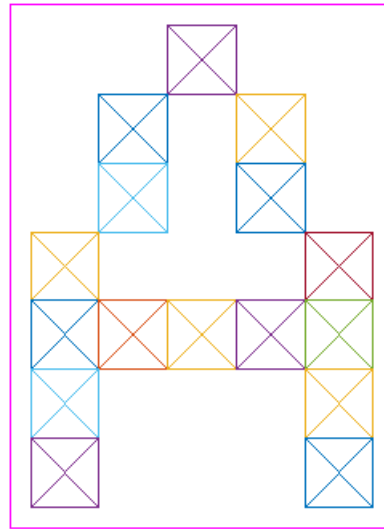


Рис. 3.5. Вхідне зображення

Створення першої нейронної мережі

Щоб вирішити цю проблему, ми будемо використовувати нейронну мережу прямого зв'язку, налаштовану на розпізнавання образів із 25 прихованими нейронами.

Оскільки нейронна мережа ініціалізується випадковими початковими вагами, результати після навчання дещо змінюються під час кожного запуску прикладу. Щоб уникнути цієї випадковості, випадкове початкове число призначене для відтворення однакових результатів щоразу.

```
setdemorandstream(pi);
net1 = feedforwardnet(25);
view(net1)
```

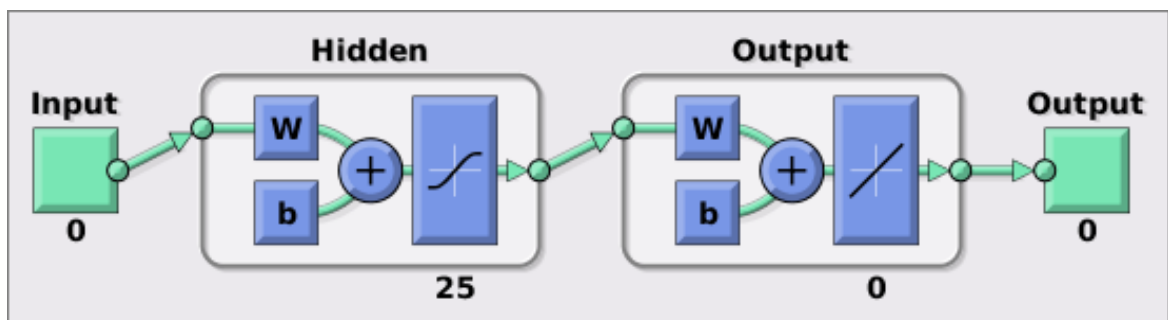


Рис. 3.6. Структура нейромережі

Навчання першої нейронної мережі

Функція навчання розділяє дані на набори для навчання, перевірки та тестування. Набір перевірки використовується для зупинки мережі до того, як вона переповнить навчальні дані, таким чином зберігаючи хороше узагальнення. Тестовий набір діє як абсолютно незалежний показник того, наскільки добре мережа може працювати на нових зразках.

Навчання припиняється, коли мережа більше не буде вдосконалювати набори для навчання або перевірки.

```
net1.divideFcn = "";
net1 = train(net1,X,T,nnMATLAB);
```

Навчання другої нейронної мережі

Ми хотіли б, щоб мережа розпізнавала не тільки ідеально сформовані літери, але й зашумлені версії літер. Тож ми спробуємо навчити другу мережу на шумових даних і порівняти її здатність робити висновки з першою мережею.

Тут створюється 30 шумних копій кожної букви Xn. Значення обмежені мінімальним і максимальним значеннями між 0 і 1. Відповідні ваги Tn також наведені.

```
numNoise = 30;
Xn = min(max(repmat(X,1,numNoise)+randn(35,26*numNoise)*0.2,0),1);
Tn = repmat(T,1,numNoise);
```

Ось шумова версія A.

```
figure
plotchar(Xn(:,1))
```

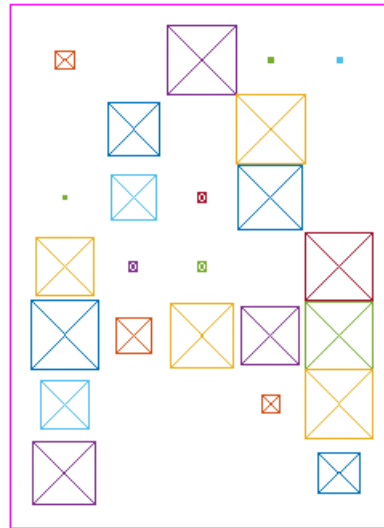


Рис. 3.7. Зашумлене зображення

Тут друга мережа створена та навчена.

```
net2 = feedforwardnet(25);
```

```
net2 = train (net2, Xn, Tn, nn MATLAB);
```

Тестування обох нейронних мереж

```
noiseLevels = 0:0.05:1;
```

```
numLevels = length(noiseLevels);
```

```
percError1 = zeros(1,numLevels);
```

```
percError2 = zeros(1,numLevels);
```

```
for i = 1:numLevels
```

```
    Xtest
```

```
    min(max(repmat(X,1,numNoise)+randn(35,26*numNoise)*noiseLevels(i),0),1);
```

```
    Y1 = net1(Xtest);
```

```
    percError1(i) = sum(sum(abs(Tn-compnet(Y1))))/(26*numNoise*2);
```

```
    Y2 = net2(Xtest);
```

```
    percError2(i) = sum(sum(abs(Tn-compnet(Y2))))/(26*numNoise*2);
```

```
end
```

```
figure
```

```
plot(noiseLevels,percError1*100,'--',noiseLevels,percError2*100);
```

```
title('Percentage of Recognition Errors');
```

```

xlabel('Noise Level');
ylabel('Errors');
legend('Network 1','Network 2','Location','NorthWest')

```

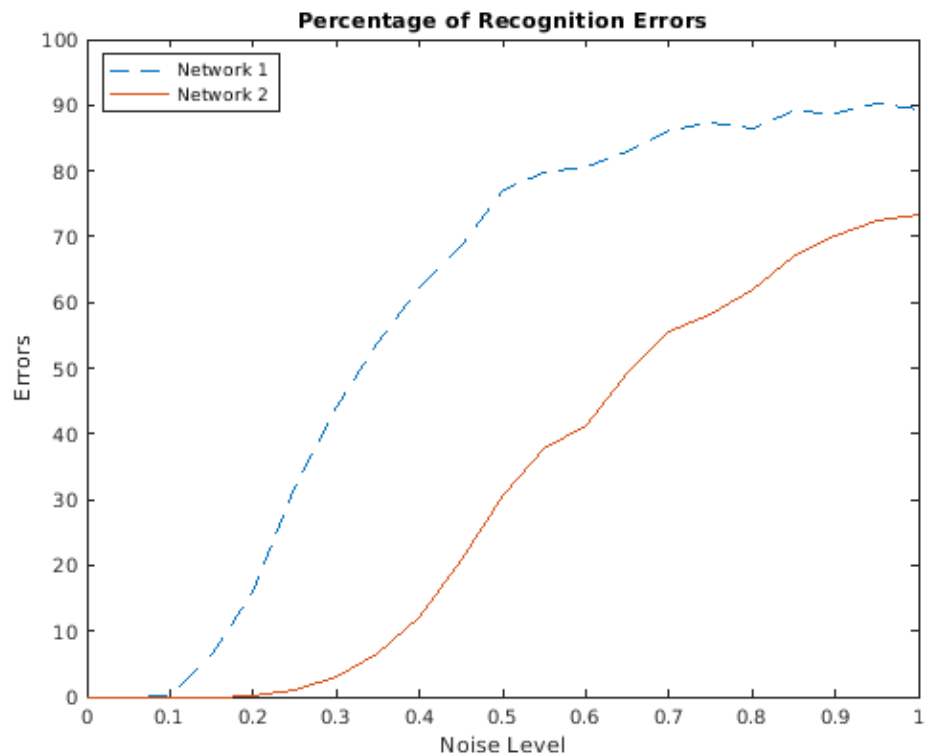


Рис. 3.8. Результати роботи двох примітивних нейромереж

Мережа 1, навчена без шуму, має більше помилок через шум, ніж фактично об'єднана з мережею 2, яка навчена з шумом.

3.4 Результати окремих моделей

До реалізованих моделей відносяться:

- Нейромережа з 1 прихованим шаром
- Багатошарова нейромережа
- Багатошарова нейромережа, побудована на основі бінаризованих

даних

- Згорткова мережа, побудована на основі вихідних даних
- Згорткова мережа, побудована на основі RGB – даних після вирівнювання діаграми
- Згорткова мережа, побудована на основі greyscale – даних після вирівнювання діаграми
- Згорткова мережа, побудована на основі greyscale – даних після підсилення
- Комбінована модель.

Нейромережа з 1 прихованим шаром, хоча не відноситься до моделей deep learning, проте виявляється дуже корисна для реалізації. До безперечних переваг даної моделі відноситься легкість її побудови і висока швидкість навчання.

Ця модель була побудована для вихідних кольорових зображень розміру 45*45 пікселів, при цьому прихований шар містив 500 нейронів. Навчання мережі зайняло близько 30 хвилин, і результуюча точність передбачення дорівнювала 59,7%.

Друга побудована модель – це багатошарова пов'язана нейронна мережа. Дана модель була побудована для бінаризованої та кольорової версій зображень меншого формату та містила 4 приховані шари. Конфігурація мережі описується так:

- Вхідний шар
- Шар 1, 1500 нейронів
- Шар 2, 500 нейронів
- Шар 3, 300 нейронів
- Шар 4, 150 нейронів
- Вихідний шар

Схематично модель цієї мережі зображено на рис. 3.9.

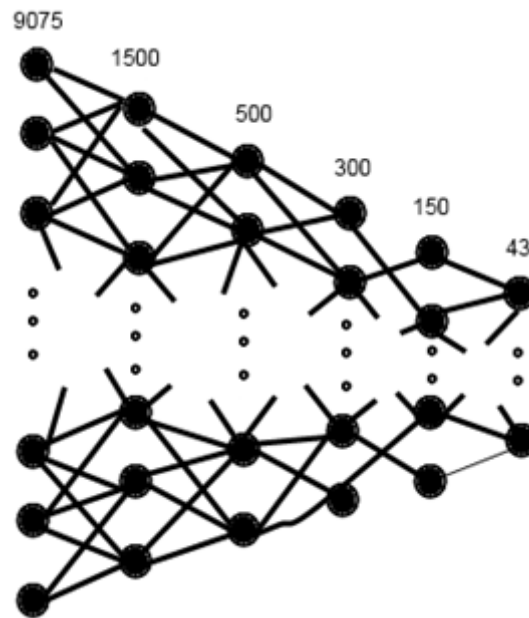


Рис. 3.9. Схема багатошарового персептрону

Підсумкова точність отриманої моделі дорівнює 66,1% для бінаризованих зображень та 81,5% для кольорових. Однак – що виправдовує побудову моделі для бінаризованих зображень, незважаючи на її меншу точність – знайшлася низка зображень, для яких саме бінаризована модель змогла визначити правильний клас.

Інші побудовані моделі так чи інакше базуються на згорткових мережах, так як саме такі мережі показали найбільшу ефективність у задачах розпізнавання класу зображень. За основу архітектури нейронної мережі було взято відому мережу LeNet, розроблену для класифікації зображень з датасета ImageNet. Однак для відповідності розглянутим зображенням, які мають значно менший розмір, мережа була модифікована. Короткий опис архітектури мережі:

- 3 шари згортки з розмірами ядра 9, 3 та 3 відповідно
- 3 шари сабсемплінгу
- 3 пов'язані шари розмірами 100, 100 і 43 нейрона

Ця мережа була окремо навчена на вихідних зображеннях більшого розміру, зображеннях після вирівнювання гістограми (колір збережений), зображеннях після вирівнювання гістограми приведених до чорно-білого

вигляду та, нарешті, чорно-білих зображеннях з підсиленням контрастом. Результати навчання представлені у таблиці 3.1:

Таблиця 3.1.

Результати навчання згорткової мережі

	Вихідне зображення	Вирівнювання гістограми (кольорове)	Вирівнювання гістограми (чорно-біле)	Підсилення контрасту (чорно-біле)
Точність	87,7%	86,7%	89,1%	82%

Можна бачити, що найкращі результати показала мережа, побудована з урахуванням чорно-білих зображень після вирівнювання гістограми. Це тому, що в процесі вирівнювання діаграми якість зображень, наприклад, відмінності між зображенням і фоном і загальний ступінь яскравості покращилися, в той же час зайва інформація, що міститься в кольорі і не несе істотного смислового навантаження, але зашумляє зображення та ускладнює класифікацію – була усунена.

Після побудови вищеописаних моделей було побудовано та навчено класифікатор, що комбінує результати всіх 6 моделей. Як алгоритм для підсумкового класифікатора було розглянуто 2 алгоритми: J48 та kStar. Загальна схема навчання підсумкового алгоритму виглядала так:

1. Навчити з використанням методу ЗПП кожену мережу на навчальній множині (множина одна і та ж для всіх мереж, але до зображень застосовані різні перетворення)

2. Для кожного екземпляра навчальної множини отримати по два найбільш ймовірні класи в порядку зменшення ймовірності від кожної мережі, зберегти отриманий набір та позначку класу

3. Використовувати отриманий набір даних – 12 атрибутів та мітку класу – як навчальну множину для підсумкового класифікатора

4. Оцінити точність отриманої моделі: для кожного екземпляра тестової множини отримати по два найбільш ймовірні класи в порядку зменшення ймовірності від кожної мережі та результуюче передбачення класу

Було обчислено результуючу точність комбінованого алгоритму: 93% при використанні алгоритму J48 та 94.8% - при використанні KStar.

При цьому алгоритм, заснований на дереві рішень, показує трохи гірші результати, проте має дві важливі переваги: по-перше, отримане в результаті роботи алгоритму дерево наочно демонструє логіку класифікації і помагає зрозуміти дані (котра з мереж дає найточніші передбачення для певного типу знаків і тому її прогноз однозначно визначає результат), по-друге – після побудови моделі даний алгоритм дозволяє здійснювати класифікацію нових даних дуже швидко, тому що для класифікації потрібно зробити лише один прохід по дереву зверху вниз. Що стосується алгоритму KStar, то в ході його роботи побудови моделі фактично не відбувається, а класифікація заснована на пошуку найбільш схожих екземплярів серед навчальної вибірки. Тому даний алгоритм, хоч і класифікує дані, але не надає при цьому жодної додаткової інформації для них, а головне – класифікація кожного екземпляра може вимагати значної кількості часу, що може бути неприйнятним для завдань, де потрібно дуже швидко отримати результат, наприклад, при розпізнаванні дорожніх знаків під час керування автомобілем.

На рис. 3.10 представлений графік навчання мережі.

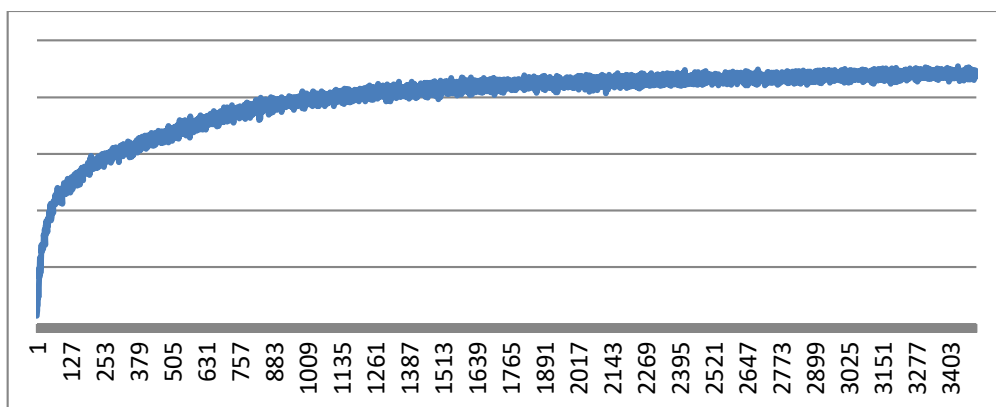


Рис. 3.10. Графік навчання згорткової мережі

Таблиця 3.2

Аналіз роботи алгоритмів

Алгоритм	Точність, %	Час навчання
Персептрон з одним прихованим шаром	59,7	30 хвилин
Багатошаровий персептрон для бінарних даних	66,1	1 година 30 хвилин
Багатошаровий персептрон для RGB-даних	81,5	5 годин
Згорткова мережа для RGB-даних	87,7	2 години
Згорткова мережа для RGB-даних з вирівнюванням гістограм	86,7	3 години
Згорткова мережа для greyscale-даних з вирівнюванням гістограм	89,1	4 години 20 хвилин
Згорткова мережа для greyscale-даних з підсиленням контрастності	82	4 години 30 хвилин (сумарний час навчання мереж)
Комбінована модель з використанням алгоритму J48	93	20 години 20 хвилин
Комбінована модель з використанням алгоритму KStar	94,8	20 години 20 хвилин

Встановлено, що за допомогою згорткових нейромереж та комбінованого алгоритму, побудованого на їх основі, вдалося отримати коректні результати, а запропонований підхід із застосуванням додаткового класифікатора для комбінації результатів нейронних мереж має безліч можливостей для подальшого вдосконалення.

3.5 Висновки до розділу 3

Було розроблено алгоритм класифікації зображень, зокрема для задачі класифікації дорожних символів, що використовує ансамбль нейромереж, що складається з двох багатошарових персептронів і 4 глибоких згорткових мереж, і із застосуванням двох типів додаткового класифікатора.

Найкраща точність класифікації досягається при використанні ансамблю нейронних мереж і класифікатора KStar, а серед окремих мереж

найкращі результати показала згорткова мережа, що використовує попереднє перетворення зображення і відтінки.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Заходи електробезпеки при роботі з виробом

При роботі з виробом необхідно дотримуватись загальні правила техніки безпеки. Залежно від способу захисту обслуговуючого персоналу від удару електричним струмом блок живлення можна віднести до I або II класу відповідно до діючого стандарту.

Правила техніки безпеки:

- при підозрі на несправність під час підготовки блоку до роботи необхідно відключити його від мережі електроживлення (знеструмити). Несправний блок категорично забороняється експлуатувати. Підозріння на несправність виникає із-за виникнення підозрілих шумів, тріску, запахів і тому подібне.

- заземлення на труби опалювання і водопроводу не можна вважати задовільним, оскільки завжди є вірогідність, що в іншому приміщенні на цю ж трубу заземлений блок із значним витоком струму, який може поширитися на пацієнта і обслуговуючий апаратуру персонал.

- якщо одночасно використовується декілька блоків, вони повинні мати одну точку заземлення. Не можна підключати блоки до землі послідовно, в цьому випадку утворюється "петля" заземлення, по якій циркулюють струми витоку.

- заміна патронів, вилок і інших з'єднувачів повинна робитися лише фахівцями, хоча на перший погляд робота здається дуже простою.

Надання першої медичної допомоги при електроударах

Першу допомогу потерпілому від удару електричним струмом потрібно надавати максимально швидко і правильно. Необхідно, насамперед, людину, що постраждала, звільнити від струму, оскільки тривалість дії впливає на

тяжкість електротравми. Перш за все треба відключити подачу електроенергії до місця події за допомогою рубильника або вимикача. Якщо вимикача поруч немає, а напруга живлячої лінії не перевищує 1000 В (напруга в побутових електричних мережах не перевищує 220 В), то звільняють потерпілого за допомогою сухого предмету не провідного електрика: палиці, дошки, сухої частини одягу, наприклад підлоги піджака або коміра куртки. Відтягаючи потерпілого за одяг, потрібно потурбуватися про власну ізоляцію. Для цієї мети підійде сухий шарф, надітий на руку, суконна або шкіряна кепка; можна встати на гумовий автомобільний килимок або прихопити нею потерпілого; можна перерубати дроти сокирою з сухою дерев'яною рукояткою. Перурабати або перекушувати дроти необхідно поодиноці, кожен фазу окремо. При напрузі в мережі вище 1000 В слід надіти діелектричні рукавички і боти, діяти ізолюючою штангою або виробити замикання дротів накоротко, накинувши на них гнучкий дріт, що має чималий перетин, аби не перегорів при виникненні струму короткого замикання. Іншими словами, потрібне спеціальне устаткування.

У всіх випадках удару електричним струмом потрібно викликати швидку медичну допомогу. Якщо потерпілий в свідомості, але був в непритомності, або якщо знаходиться в несвідомому стані, але дихання і пульс збережені, то його треба покласти, розстібнути одяг, зігріти тіло і створити спокійну обстановку довкола. Аби потерпілий не захлинувся від блювоти, голову необхідно обернути набік. Коли потерпілий приходить в свідомість, йому забороняють підійматися і ходити до приїзду швидкої медичної допомоги. Зазвичай таких потерпілих доставляють в стаціонарне відділення і декілька днів спостерігають за їх станом.

Якщо після звільнення від дії електричного струму потерпілий не дихає або дихання у нього рідке, неглибоке, і шкірні покриви поступово синіють, то необхідно провести штучне дихання. Дихання з рота в рот або з рота в ніс забезпечують вступ в легені рятованого необхідної кількості повітря, придатного для дихання. Вдування повітря здійснюється через марлю або

носову хустку. При цьому методі штучного дихання видно, чи потрапляє повітря в легені рятованого, на вдиху у нього підіймаються ребра, а видих відбувається пасивно, за рахунок природної еластичності грудної клітки.

4.2 Безпека в надзвичайних ситуаціях

Пожежна безпека на підприємстві з виготовлення телекомунікаційних систем [2]

Пожежа - неконтрольоване горіння поза спеціальним вогнищем, яке призводить до матеріальної шкоди.

Пожежна безпека – стан об’єкта, при якому з регламентованою ймовірністю виключається можливість виникнення та розвиток пожежі і впливу на людей її небезпечних факторів, а також забезпечується захист матеріальних цінностей.

Причинами пожеж та вибухів на підприємстві є порушення правил і норм пожежної безпеки, невиконання Закону “Про пожежну безпеку”.

Небезпечними факторами пожежі і вибуху, які можуть призвести до травми, отруєння, загибелі або матеріальних збитків є відкритий вогонь, іскри, підвищена температура, токсичні продукти горіння, дим, низький вміст кисню, обвалення будинків і споруд.

За стан пожежної безпеки на підприємстві відповідають її керівники, начальники цехів, майстри та інші керівники.

На підприємствах існує два види пожежної охорони: професійна і воєнізована. Воєнізована охорона створюється на об’єктах з підвищеною небезпекою. Крім того на підприємствах для посилення пожежної охорони організовуються добровільні пожежні дружини і команди, добровільні пожежні товариства і пожежно-технічні комісії з числа робітників та службовців. При Міністерстві внутрішніх справ існує управління пожежної охорони (УПО) і його органи на місцях. До складу УПО входить Державний пожежний нагляд який здійснює:

Контроль за станом пожежної безпеки

Розробляє і погоджує протипожежні норми і правила та контролює їх виконання в проектах і безпосередньо на об'єктах народного господарства

Проводить розслідування і облік пожеж

Організовує протипожежну профілактику.

Протипожежна профілактика – це комплекс організаційних і технічних заходів, які спрямовані на здійснення безпеки людей, на попередження пожеж, локалізацію їх поширення, а також створення умов для успішного гасіння пожежі.

Відповідальним керівником робіт по ліквідації пожеж і аварій на підприємстві є головний інженер. Начальник структурного підрозділу, в якому виникла пожежа, є відповідальним виконавцем робіт по її ліквідації.

Протипожежні вимоги до будинків і споруд [1]

Виходячи з властивостей речовин і матеріалів, умов їх застосування і обробки і у відповідності із ОНТП 24-86 “Визначення категорій приміщень і будівель по вибухопожежній і пожежній небезпеці” приміщення по вибухопожежній і пожежній небезпеці діляться на п'ять категорій – А, Б, В, Г, Д.

До категорії А належать приміщення, де перебувають спалимі та легкозаймисті рідини з температурою спалаху, що не перевищує 28°C, а також речовини і матеріали здатні вибухати і горіти при взаємодії з водою, киснем або одне з одним; при утворенні вибухонебезпечних сумішей розвивається розрахунковий надлишковий тиск вибуху 5 кПа.

До категорії Б належать приміщення, в яких є пил та волокна, легкозаймисті рідини з температурою спалаху понад 28°C та спалимі рідини в такій кількості, що можуть утворюватися вибухонебезпечні пилоповітряні та пароповітряні суміші, при займанні яких розвивається розрахунковий надлишковий тиск вибуху 5 кПа.

До категорії В належать приміщення, де перебувають спалимі та важкоспалимі рідини, тверді спалимі та важкоспалимі речовини та матеріали

(в тому числі пил та волокна), а також речовини і матеріали які здатні при взаємодії з водою, киснем повітря та одне з одним тільки горіти (за умови, що ці приміщення не відносяться до категорії А чи Б).

До категорії Г належать приміщення, в яких є неспалимі речовини та матеріали в гарячому, розпеченому або розплавленому стані, а також спалимі гази, рідини та тверді речовини, які спалюються або утилізуються як паливо; процес їх обробки супроводжується виділенням променевої теплоти, іскор та полум'я.

До категорії Д належать приміщення, в яких є неспалимі речовини та матеріали у холодному стані.

На розвиток пожежі у приміщеннях та спорудах значно впливає здатність окремих будівельних елементів чинити опір впливу тепла, тобто їх вогнестійкість.

Вогнестійкість – здатність будівельних конструкцій чинити опір дії високої температури, утворенню наскрізних тріщин та поширенню вогню в умовах пожежі і виконувати при цьому свої звичайні експлуатаційні функції. Вогнестійкість конструкцій будівель характеризується межею вогнестійкості.

Межа вогнестійкості – це час, на протязі якого конструкція може витримати дію вогню, а потім вже починається деформація.

Всі будівлі і споруди за ступенем вогнестійкості за СНиП 2.01.02-85 поділяють на 5 ступеней.

Будинок може належати до того або іншого ступеня вогнестійкості, якщо значення меж вогнестійкості і меж поширення вогню усіх конструкцій не перевищує значень вимог СНиП 2.01.02-85. [1]

4.3 Висновки до розділу

В розділі «Охорона праці та безпека в надзвичайних ситуаціях» описано заходи електробезпеки при роботі з виробом, а також питання пожежної безпеки на підприємстві.

ВИСНОВКИ

Розглянуто технологію машинного зору та машинного навчання, як одного із підрозділів штучного інтелекту. Наведено підходи до організації класифікації із використанням машинного навчання. Показано, що для задач розпізнавання зображень або мови найбільш підходящим методом класифікації стає використання нейромереж. Розглянуто принципи функціонування нейромереж, зокрема із застосуванням глибокого навчання.

Показано задачі класифікації зображень, та бібліотеки, що можуть для цього застосовуватись.

На основі проведеного огляду для розроблення алгоритму класифікації зображень проведено використання глибоких згорткових нейромереж, багатошарових персептронів, комбінування результатів кількох моделей за допомогою додаткового класифікатора, застосування додаткових перетворень до вхідних даних.

Було розроблено алгоритм для розпізнавання зображень на прикладі задачі розпізнавання дорожніх символів, що використовує ансамбль нейромереж, що складається з двох багатошарових персептронів і 4 глибоких згорткових мереж, і із застосуванням двох типів додаткового класифікатора.

Були оцінені окремо результати кожної з мереж, що входять до нього, і підсумкова точність алгоритму для двох типів додаткового класифікатора. За результатами експериментів, найбільша точність розпізнавання досягається при використанні ансамблю нейронних мереж і класифікатора KStar, а серед окремих мереж найкращі результати показала згорткова мережа, що використовує попереднє перетворення зображення і відтінки.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Al-Azawi M. A. N. Neural Network Based Automatic Traffic Signs Recognition //International Journal of Digital Information and Wireless Communications (IJDIWC). – 2011. – Т. 1. – №. 4. – С. 753-766.
2. Baldi P. Autoencoders, Unsupervised Learning, and Deep Architectures //ICML Unsupervised and Transfer Learning. – 2012. – Т. 27. – С. 37-50.
3. Bahlmann C. et al. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information //Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. – IEEE, 2005. – С. 255-260.
4. Bastien F. et al. Theano: new features and speed improvements //arXiv preprint arXiv:1211.5590. – 2012.
5. Bengio Y., Goodfellow I., Courville A . Deep Learning. – MIT Press, book in preparation
6. Bergstra J. et al. Theano: A CPU and GPU math compiler in Python //Proc. 9th Python in Science Conf. – 2010. – С. 1-7.
7. Broggi A. et al. Real time road signs recognition //Intelligent Vehicles Symposium, 2007 IEEE. – IEEE, 2007. – С. 981-986.
8. Canny J. A computational approach to edge detection //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 1986. – №. 6. – С. 679-698.
9. Ciresan D., Meier U., Schmidhuber J. Multi-column deep neural networks for image classification //Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. – IEEE, 2012. – С. 3642-3649.
10. Ciresan D. et al. A committee of neural networks for traffic sign classification //Neural Networks (IJCNN), The 2011 International Joint Conference on. – IEEE, 2011. – С. 1918-1921.

11. Cireşan D. C. et al. Deep big multilayer perceptrons for digit recognition //Neural Networks: Tricks of the Trade. – Springer Berlin Heidelberg, 2012. – C. 581-598.
12. Daugman J. G. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression //Acoustics, Speech and Signal Processing, IEEE Transactions on. – 1988. – T. 36. – №. 7. – C. 1169-1179.
13. Gao X. W. et al. Recognition of traffic signs based on their colour and shape features extracted using human vision models //Journal of Visual Communication and Image Representation. – 2006. – T. 17. – №. 4. – C. 675-685.
14. Goodfellow I. J. et al. Pylearn2: a machine learning research library //arXiv preprint arXiv:1308.4214. – 2013.
15. Han J., Kamber M., Pei J. Data mining: Concepts and techniques. – Morgan kaufmann, 2006.
16. Harris C., Stephens M. A combined corner and edge detector //Alvey vision conference. – 1988. – T. 15. – C. 50.
17. Houben S. et al. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark //Neural Networks (IJCNN), The 2013 International Joint Conference on. – IEEE, 2013. – C. 1-8.
18. Huang F. J., LeCun Y. Large-scale learning with svm and convolutional netw for generic object recognition //2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2006.
19. Huttenlocher D. P., Ullman S. Object recognition using alignment //Proc. ICCV. – 1987. – T. 87. – C. 102-111.
20. Jia, Yangqing. "Caffe: An open source convolutional architecture for fast feature embedding." h ttp://caffe. berkeleyvision. org (2013).
21. Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks //Advances in neural information processing systems. – 2012. – C. 1097-1105.

22. Lafuente-Arroyo S. et al. Traffic sign classification invariant to rotations using support vector machines //Proceedings of Advanced Concepts for Intelligent Vision Systems, Brussels, Belgium. – 2004.
23. LeCun Y., Bengio Y. Convolutional networks for images, speech, and time series //The handbook of brain theory and neural networks. – 1995. – T. 3361. – C. 310.
24. LeCun Y. et al. Learning algorithms for classification: A comparison on handwritten digit recognition //Neural networks: the statistical mechanics perspective. – 1995. – T. 261. – C. 276.
25. Masci J. et al. Stacked convolutional auto-encoders for hierarchical feature extraction //Artificial Neural Networks and Machine Learning–ICANN 2011. – Springer Berlin Heidelberg, 2011. – C. 52-59.
26. Matan O. et al. Handwritten character recognition using neural network architectures //Proceedings of the 4th USPS Advanced technology Conference. – 1990. – C. 1003-1011.
27. McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity //The bulletin of mathematical biophysics. – 1943. – T. 5. – №. 4. – C. 115-133.
28. Minsky M., Seymour P. Perceptrons. – 1969.
29. Mitchell T. Generative and discriminative classifiers: naive Bayes and logistic regression, 2005 //Manuscript available at [http://www. cs. cm. edu/~tom/NewChapters. html](http://www.cs.cmu.edu/~tom/NewChapters.html).
30. Moutarde F. et al. Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular Traffic Signs Recognition system //Intelligent Vehicles Symposium, 2007 IEEE. – IEEE, 2007. – C. 1122-1126.
31. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain //Psychological review. – 1958. – T. 65. – №. 6. – C. 386.

32. Ruta A., Li Y., Liu X. Real-time traffic sign recognition from video by class-specific discriminative features //Pattern Recognition. – 2010. – Т. 43. – №. 1. – С. 416-430.
33. Sermanet P., LeCun Y. Traffic sign recognition with multi-scale convolutional networks //Neural Networks (IJCNN), The 2011 International Joint Conference on. – IEEE, 2011. – С. 2809-2813.
34. Serra J. Introduction to mathematical morphology //Computer vision, graphics, and image processing. – 1986. – Т. 35. – №. 3. – С. 283-305.
35. Stallkamp J. et al. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition //Neural networks. – 2012. – Т. 32. – С. 323-332.
36. Zaklouta F., Stanciulescu B. Real-time traffic sign recognition using spatially weighted HOG trees //Advanced Robotics (ICAR), 2011 15th International Conference on. – IEEE, 2011. – С. 61-66.
37. Математичне та комп'ютерне моделювання електрокардіосигналів у системах голтерівського моніторингу / Л.Є. Дедів, А.С. Сверстюк, І.Ю. Дедів, М.О. Хвостівський, В.Г. Дозорський, Є.Б. Яворська. – Львів: Видавництво «Магнолія - 2006», 2021. – 120 с. ISBN 978-617-574-218-1
38. Математичне моделювання, методи та програмне забезпечення опрацювання дихальних шумів у комп'ютерних аускультативних діагностичних системах / І.Ю. Дедів, А.С. Сверстюк, Л.Є. Дедів, В.Г. Дозорський, М.О. Хвостівський. – Львів: Видавництво «Магнолія - 2006», 2021. – 126 с. ISBN 978-617-574-219-8
39. Методичні рекомендації з оформлення кваліфікаційних робіт бакалавра за спеціальністю 172 «Телекомунікації та радіотехніка» уклад.: Дунець В.Л., Хвостівський М.О. Дедів І.Ю. Тернопіль: ТНТУ імені Івана Пулюя, 2021 р. – 72с.
40. Дозорський В.Г., Дозорська О.Ф., Дедів Л.Є., Дедів І.Ю., Паньків І.М., Яворська Є.Б. Структура системи відбору біосигналів для задачі

відновлення комунікативної функції людини. Вісник Хмельницького національного університету: технічні науки. – Хмельницький: редакція журналу "Вісник Хмельницького національного університету". – 2019. - №2(271) – с. 183-186.

41. Khvostivska L.V., Osukhivska H.M., Khvostivskyi M.O., Dediv S.Y. Development of methods and algorithms for a stochastic biomedical signal period calculation in medical computer diagnostic systems. Visnyk NTUU KPI Serii A - Radiotekhnika Radioaparaturubuduvannia, /Категорія В/ 2019. Вип. 79. С. 78-84. doi: 10.20535/RADAP.2019.79.78-84.

42. Dozorska O., Yavorska E., Dozorskyi V., Pankiv I., Dediv L. Dediv I. The Method of Indirect Restoration of Human Communicative Function. Proc. of the 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), CADSM'2019, (pp. 19–22). Polyana-Svalyava (Zakarpattia), UKRAINE 978-1-7281-0053-1/19.

43. Дозорська О.Ф., Яворська Є.Б., Дозорський В.Г., Дедів Л.Є., Дедів І.Ю. Метод виявлення ознак основного тону в структурі електроміографічних сигналів для задачі компенсації порушеної комунікативної функції людини», Вісник НТУУ "КПІ". Серія Радіотехніка, Радіоапаратуробудування, (81), с. 56-64. doi: 10.20535/RADAP.2020.81.56-64.

44. Khvostivska L., Khvostivskyi M., Dunetc V., Dediv I. Mathematical and Algorithmic Support of Detection Useful Radiosignals in Telecommunication Networks. 2nd International Workshop on Information Technologies: Theoretical and Applied Problems, ITTAP 2022. CEUR Workshop Proceedings. Ternopil 22-24 November 2022. Vol 3309, P. 314-318. ISSN 1613-0073.

45. Гевко О.В., Дозорський В.Г., Дедів Л.Є., Дедів І.Ю., Дозорська О.Ф. Структурний синтез вібромасажної апаратури. Перспективні технології та прилади, № 20, Луцьк, 2022. – с. 23-31.

46. Dozorskyi V., Dediv I., Sverstiuk S., Nykytyuk V., Karnaukhov A. The Method of Commands Identification to Voice Control of the Electric Wheelchair. Proceedings of the 1st International Workshop on Computer Information

Technologies in Industry 4.0 (CITI 2023). CEUR Workshop Proceedings. Ternopil, Ukraine, June 14-16, 2023. P.233-240. ISSN 1613-0073.

47. Khvostivska L., Khvostivskyi M., Dediv I., Yatskiv V., Palaniza Y. Method, Algorithm and Computer Tool for Synphase Detection of Radio Signals in Telecommunication Networks with Noises. Proceedings of the 1st International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2023). CEUR Workshop Proceedings. Ternopil, Ukraine, June 14-16, 2023. P.173-180. ISSN 1613-0073.

48. Khvostivska L., Khvostivskyi M., Dunets V., Dediv I. Mathematical, algorithmic and software support of synphase detection of radio signals in electronic communication networks with noises. Scientific Journal of TNTU (Tern.), vol 111, no 3, 2023. pp. 48–57.

49. Основи технології радіоелектронних апаратів : навчальний посібник / Р. А. Ткачук, В. Г. Дозорський, Л. Є. Дедів, І. Ю. Дедів. - Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017. - 336 с.

50. Паляниця Ю.Б., Марценюк А.С., Дунець В.Л., Бучинський В.М., Паламар М.І. Дрон з блоком надвисоких частот для виявлення та знешкодження вибухових пристроїв та мін. Матеріали III Міжнародної наукової конференції молодих учених та студентів «Воєнні конфлікти та техногенні катастрофи: історичні та психологічні наслідки» Тернопільського національного технічного університету імені Івана Пулюя: зб. тез доповідей, 20-21.04.2023 р. Тернопіль: ТНТУ, 2023. С. 158-159. ISBN 978-617-7875-32-0.

51. Дунець В.Л., Хвостівська Л.В., Паляниця Ю.Б. Математичне, алгоритмічне та програмне забезпечення оцінювання завадозахищеності каналів зв'язку з балансною модуляцією. Збірник наукових праць Вісник НУВГП, серія технічні науки, випуск 4 (104), 2023. - С. 95-107. ISSN: 2306-5478

52. Гевко І.Б., Дунець В.Л., Паляниця Ю.Б., Марценюк А.С., Химич Г.П. Дрон з імпульсним квантовим генератором для знешкодження об'єктів

ураження. Матеріали IV Міжнародної наукової конференції «Воєнні конфлікти та техногенні катастрофи: історичні та психологічні наслідки» Тернопільського національного технічного університету імені Івана Пулюя: зб. тез доповідей, 18-19.04.2024 р. Тернопіль: ТНТУ, 2024

53. Khymych, H., Dunets, V., Duda, S., Palaniza, Y., Kornieiev, K. Dual-Polarization Yagi Antenna for Meter Wavelength Range. *Radioelectronics and Communications Systems*, 66(11), 2023, pp. 609–615. ISSN 0735-2727. DOI: 10.3103/S0735272722080039.

54. Lyashuk, Oleg, Mykola Stashkiv, Yurii Palianytsia, Roman Khoroshun and Dmytro Mironov. Fourier Model Fitting for Vibration Analysis of Car Wheel Suspension Enhanced Damping System. *Proceedings of the 4rd International Workshop on Information Technologies: Theoretical and Applied Problems*, 23-25 October 2024, Ternopil, Ukraine. ITTAP-2024, 2024. ISSN 1613-0073

55. Паляниця Ю.Б., Сверстюк А.С., Шадріна Г.М. Математичне та комп'ютерне моделювання фонокардіосигналів для удосконалення кардіодіагностичних систем / Ю.Б. Паляниця, А.С. Сверстюк, Г.М. Шадріна – Львів: Видавництво «Магнолія - 2006», 2020. – 106 с. ISBN 5-211-05310-9.

56. Dozorskyi V., Dediv L., Kovalyk S., Dozorska O., Dediv I. (2024) Design of the endoskeleton of a biocontrolled hand prosthesis. *Scientific Journal of TNTU (Tern.)*, vol. 115, no 3, pp. 100-111.

57. Техноекоекологія та цивільна безпека. Частина «Цивільна безпека». Навчальний посібник / В.С. Стручок, – Тернопіль: ТНТУ ім. І.Пулюя, 2022. – 150 с.

58. Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної бо та заочної (дистанційної) форм навчання / В.С.Стручок. — Тернопіль: ФОП Паляниця В. А., 2022. — 156 с.

ДОДАТКИ

УДК 621.395.625.6

Ірина Дедів, канд. техн. наук, доц., Маркіян Пліс, Валентин Степанов, Сергій Брегін, Володимир Лотоцький

Тернопільський національний технічний університет імені Івана Пулюя

ЗАДАЧА ОЦІНЮВАННЯ РОЗБІРЛИВОСТІ МОВИ ДЛЯ ПОКРАЩЕННЯ ЯКОСТІ СИСТЕМ ГРОМАДСЬКОГО ОПОВІЩЕННЯ

Iryna Dediv, Ph.D, Assoc. Prof., Markiyan Plis, Valentyn Stepanov, Sergii Bregin, Volodymyr Lototskyi

THE TASK OF SPEECH INTELLIGIBILITY ASSESSMENT FOR IMPROVING THE QUALITY OF PUBLIC ANNOUNCEMENT SYSTEMS

Сьогодні в Україні, в умовах війни, особливо гостро стоїть питання своєчасного оповіщення громадян про ймовірні небезпечні непередбачувані ситуації, зокрема повітряні ракетні атаки, обмеження руху та пересування через ймовірності вибухів зокрема, виникнення пожеж чи значні викиди шкідливих речовин в повітря, воду тощо. З цією метою використовуються системи громадського оповіщення, що є організованими спеціалізованими системами, які призначені для своєчасної передачі певних сигналів та інформаційних повідомлень стосовно цивільного захисту до відповідних органів влади, підприємств, організацій, установ та населення. Ці системи включають в себе технологічні підходи та способи оповіщення, спеціалізоване обладнання, прилади та канали зв'язку. Метою роботи систем оповіщення є забезпечення максимально швидкого інформування якомога більшої кількості людей про безпосередню небезпеку. Громадські (публічні) оповіщення здійснюються дистанційно шляхом використання електромеханічних сирен (зовні), мереж мовлення всіх частотних типів, систем телевізійного та мобільного зв'язку. Основна функція, яку виконує система оповіщення в аварійній ситуації, – це трансляція мовних повідомлень, спрямованих на запобігання паніці людей, та передача інформації про напрям руху для евакуації.

В поширених сьогодні варіантах побудови систем оповіщення як основний інформаційний сигнал про виникнення надзвичайної ситуації використовується мовний чи голосовий сигнал, що являє собою голосове повідомлення, яке повинне бути однозначно донесене та сприйняте людьми чи персоналом, що перебуває в зоні дії цієї системи. При цьому, часто для підвищення ефективності технічних засобів оповіщення, зокрема підвищення випромінюваної гучномовцями потужності, проводиться попередня обробка таких сигналів, зокрема фільтрація, що передбачає подавлення високочастотних та низькочастотних складових сигналу-повідомлення та підсилення середньочастотних складових. Відповідно, часто системи оповіщення транслюють готосові повідомлення в ділянці саме середніх частот голосового сигналу. Але це в свою чергу разом із впливом зовнішніх факторів значно впливатиме на якість того голосового повідомлення, яке будуть чути люди, для яких воно транслюється. Ці і додаткові фактори значно спотворюватимуть сигнали повідомлення та впливатимуть на ступінь сприйняття оточуючими людьми таких повідомлень від системи оповіщення.

В цьому плані важливим є контроль якості роботи системи з можливістю коригування параметрів чи характеристик голосових повідомлень для забезпечення надійного та однозначного їхнього сприйняття оточуючими людьми. Власне для цього і потрібно розробити новий чи обґрунтувати вибір відомого методу оцінювання розбірливості мови.

Приклад структури нейронної мережі
Багатошарова мережа з одним прихованим шаром

```
name: "CaffeNet"
layer {
  name: "data"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    mirror: true
    crop_size: 30
    mean_file: "/D:/imagenet_mean_train_complete.binaryproto"
  }
  data_param {
    source: "/ D:/data_complete/ilsrvrc12_train_lmdb"
    batch_size: 10
    backend: LMDB
  }
}
layer {
  name: "data"
  type: "Data"
  top: "data"
  top: "label"
  include {
```

```
phase: TEST
}
transform_param {
  mirror: false
  crop_size: 30
  mean_file: "/ D:/imagenet_mean_val_complete.binaryproto"
}
data_param {
  source: "/ D:/data_complete/ilsrvrc12_val_lmdb"
  batch_size: 1
  backend: LMDB
}
}
layer {
  name: "fc1"
  type: "InnerProduct"
  bottom: "data"
  top: "fc1"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  inner_product_param {
    num_output: 500
    weight_filler {
      type: "gaussian"
```

```
std: 0.005
}
bias_filler {
type: "constant"
value: 0
}
}
}
layer {
name: "fc2"
type: "InnerProduct"
bottom: "fc1"
top: "fc2"
param {
lr_mult: 1
decay_mult: 1
}
param {
lr_mult: 2
decay_mult: 0
}
inner_product_param {
num_output: 43
weight_filler {
type: "gaussian"
std: 0.005
}
bias_filler {
type: "constant"
value: 1
```

```
}  
}  
}  
layer {  
  name: "accuracy"  
  type: "Accuracy"  
  bottom: "fc2"  
  bottom: "label"  
  top: "accuracy"  
  include {  
    phase: TEST  
  }  
}  
layer {  
  name: "loss"  
  type: "SoftmaxWithLoss"  
  bottom: "fc2"  
  bottom: "label"  
  top: "loss"  
}
```