

Approach. Addison-Wesley, reading, MA, 1991. 10. *Thalheim B. Fundamentals of Entity-Relationship Modeling*, Springer-Verlag, Berlin, 2000. 11. *Abiteboul, S. and Bidoit, N. Non First Normal Form Relations to Represent Hierarchically Organized Data*. In *Proc. 3rd ACM SIGACT-SIGMOD*, Waterloo, 1984, p. 191-198. 12. *Bancilhon, F. and al. Verso: A Relational Back-End Data Machine*. In *Proc. Inter., Workshop on Database Machines*, san Diego, 1982. 13. *Bidoit, N. Efficient Evaluation of Relational Queries Using "Nested Relations"*. *Rapport de recherche de l'INRIA-Rocquencourt RR-0480 -1986*. www.inria.fr/rrrt/rr-0480.html - 3k 14. *Ullman, J. Principles of Databases Systems*. Computer Science Press. 15. *Bidoit, N. Un Modele de Donnees Relationnel Non Normalise: Algebre et Interpretation*. PhD thesis, Universite de Paris-Sud, Centre d'Orsay, 1984. 16. *Maier, D., Mendelzon, A., Sagiv, Y. A Consideration on Normal Form of Not-Necessarily Normalized Relation in the Relation Model*. In *Proc. Inter. Conf. on VLDB*, Tokyo, 1977, p. 447-453. 17. *Aho, A., Sagiv, Y., Ullman J. Efficient Optimization of a Class of Relational Expressions*. *ACM Trans. On Database Systems*, 1979, 4(4): 435-454. 18. *Aho, A., Sagiv, Y., Ullman J. Equivalences Among Relational Expressions*. *Siam Journal Comp.*, 1979, 8(2): 218-246. 19. *Klug, A. On Inequality Tableaux*. Technical Report 403, Computer Science Department, University of Wisconsin-Madison, November, 1980.

УДК 004.02

М.В. Давидов, Ю.В. Нікольський
Національний університет "Львівська політехніка",
кфедра інформаційних систем та мереж

КЛАСИФІКАЦІЯ ЕЛЕМЕНТІВ ВІДЕОЗОБРАЖЕНЬ РЕАЛЬНОГО ЧАСУ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖІ

© Давидов М. В., Нікольський Ю.В., 2005

Описаний прототип системи керування виступом за допомогою набору жестів. Задачу вирішено побудовою класифікатора елементів зображення за допомогою багатошарової нейронної мережі. Модифіковано алгоритм навчання зі зворотним поширенням похибки. Запропоновано новий метод визначення еталону та навчальних прикладів.

The prototype of conference control system is described. The problem is solved by applying the neural network as classifier of image elements. New modification of the backpropagation algorithm for training the neural network was built. The new way to define the patterns and standards is used.

Постановка проблеми

Пропонуємо результати досліджень, здійснених з метою побудови прототипу системи інтерактивного керування виступом. Доповідач може застосовувати широкий спектр засобів супроводження виступу, до яких належать мультимедійний проектор, кодоскоп, проектор слайдів, системи керування освітленням приміщення, шторами, мікрофонами, засобами запису та відображення інформації: диктофоном, магнітофоном, відеокамерою, цифровим фотоапаратом тощо. Цими пристроями може керувати оператор з центрального пульта. У такому випадку для виступу виділяють спеціально обладнане приміщення з вказаним комплектом обладнання, а оператору необхідно мати сценарій виступу. У запропонованій системі, спрощену схему якої показано на рис.1, відсутня необхідність у такому обладнаному приміщенні. Систему можна скласти з наявних пристроїв у вільній конфігурації з мінімальними вимогами до вартості та сумісності їх роботи, зокрема, не потрібний центральний пульт та спеціально підготовлений оператор. Всі пристрої під'єднано до центрального сервера, який керує їхньою роботою. Доповідача постійно знімає відеокамера, комп'ютер опрацьовує відеосигнал, виділяє у ньому жести доповідача та передає керуючий сигнал на відповідний пристрій. Складовими цієї системи є доповідач, екран, відеока-

мера, програмно-апаратний комплекс та пристрої відображення матеріалів виступу (мультимедійний проектор, кодоскоп, проектор слайдів).

Розглянемо функції складових запропонованої системи.

1. Доповідач під час виступу керує комплексом пристроїв визначеним набором жестів.
2. На екран проєктують матеріали виступу. На екран також можна проєктувати жести з метою розпізнання їх проєкцій. Іншим способом керування пристроями може бути набір міток, які проєктують на екран під час виступу. У такому випадку керувати пристроями можна не набором жестів, а доторканням пальця, указки або світлової мітки відповідної ділянки екрана.
3. Відеокамера знімає доповідача та екран і передає отримане зображення для опрацювання на сервер керування програмно-апаратним комплексом.
4. Пристрої відображення матеріалів виступу подають ці матеріали на екран.
5. Програмно-апаратний комплекс складається з сервера та розміщеного на ньому програмного забезпечення. Програмно-апаратний комплекс отримує сигнал з відеокамери, опрацьовує його у реальному часі та передає керуючі сигнали на під'єднані до нього пристрої.

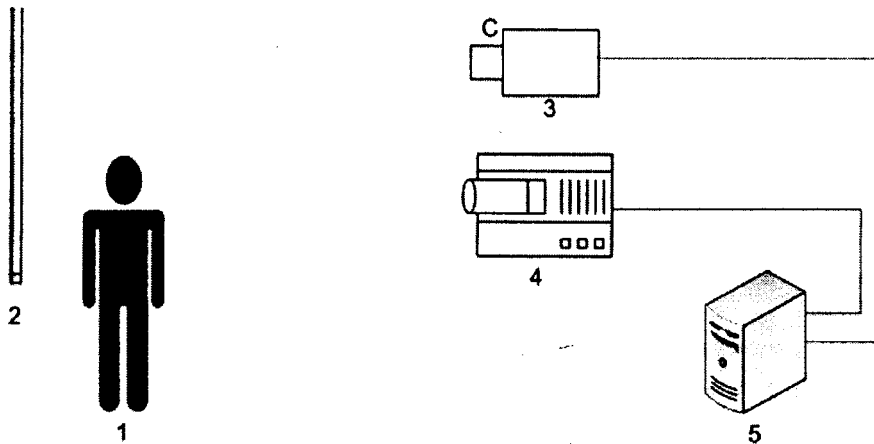


Рис. 1. Структура програмно-апаратного комплексу

Основні задачі, які вирішуються під час створення програмно-апаратного комплексу, пов'язані з опрацюванням зображення руки доповідача, розпізнаванням на цьому зображенні кінців пальців та класифікацією жестів. Даними для розв'язування цих задач є зображення, які надходять з відеокамери та проходять низку етапів опрацювання. Задачу розпізнавання зображення кінців пальців сформульовано та вирішено як задачу класифікації.

Аналіз останніх досліджень

Методи розпізнавання поділяють на детерміновані, ймовірнісні, логічні, структурно-лінгвістичні та комбіновані [1]. Детерміновані методи основані на використанні міри близькості векторів ознак досліджуваного об'єкта та спеціально створених еталонів. За основу ймовірнісних методів взято визначення ступеня належності досліджуваного об'єкта певному класу. Логічні методи для оцінювання належності об'єкта класу використовують числення висловлювань та дискретний аналіз. Структурно-лінгвістичні методи аналізують опис об'єкта як сукупності його неподільних структурних елементів. Комбіновані методи об'єднують детерміновані, ймовірнісні, логічні та структурно-лінгвістичні методи.

Зображення традиційно обробляють за такою схемою [2], у послідовності розв'язання задач щодо певних елементів зображень:

- 1) оцифрування зображення;
- 2) очищення зображення від шумів та завад;
- 3) опис шуканих елементів та створення бази еталонів для цих елементів;
- 4) пошук елементів зображення, у певному сенсі близьких до еталонних;
- 5) класифікація знайдених елементів.

Оцифрування полягає у перетворенні відеосигналу з аналогового у цифровий формат або цифрового відеосигналу у формат, який опрацюють програмно під час вирішення основної задачі розпізнавання. Зображення у сучасних системах оцифровують за допомогою цифрової камери, відеосервера або відеокарти комп'ютера. Оцифрування зображень відеокамерою характерне для веб-камер, які під'єднують безпосередньо до комп'ютера через USB порт. Також можна використовувати мережеві камери, які під'єднують до мережі Ethernet. Якщо відеосигнал надходить з аналогової відеокамери у стандарті VHS, то його також перетворюють у необхідний цифровий формат програмно або апаратно. Аналогову відеокамеру можна під'єднати до відеосервера у мережі Ethernet або до відеовходу відеокарти комп'ютера.

Зображення очищують або до, або під час комп'ютерного опрацювання. Очищення полягає в усуненні несуттєвих елементів зображення, оскільки ці елементи можуть ускладнити та збільшити час опрацювання. Також вони можуть спотворити результат, що спричинено особливостями застосування алгоритмів опрацювання. До елементів, що усувають під час очищення, належать шуми на зображеннях, нерівномірність освітлення, спотворення геометрії або кольорів зображень тощо. Для очищення зображення використовують різноманітні фільтри або методи фільтрації [3].

Опис елементів зображення полягає у створенні бази шаблонів – еталонів, за якими шукають елементи зображень. Шаблон – це набір характеристик, які дозволяють класифікувати елементи зображення віднесенням його до одного з відомих класів елементів. Для підвищення ефективності алгоритмів опрацювання зображень створювані шаблони повинні враховувати нерівномірність освітлення та фону, повороти частин зображень, їх перекриття тощо. Це збільшує кількість шаблонів та ускладнює алгоритми виділення наборів характеристик, з яких формують шаблон. Основна проблема цього етапу опрацювання зображень полягає у тому, що відсутня загальна методика створення шаблонів, а кожна нова задача вимагає розроблення такої методики.

Пошук елементів зображення полягає у знаходженні на зображенні характеристик, за якими здійснюють пошук у базі шаблонів. Характеристики знаходять послідовним переглядом зображення, виділенням його фрагментів та порівнянням їх з шаблоном. Для цього шаблон пересувають по зображенню з певним кроком, який узгоджують з розмірами шуканого елемента. При цьому виходять з таких міркувань: якщо у близькому околі елемента зображення є точки, які мають характеристики, близькі до характеристик шаблону, то ці точки вважають такими, що належать шаблону. Введення поняття близькості є предметом спеціальних досліджень.

Класифікацію виконують порівнянням характеристик знайдених елементів з характеристиками шаблонів та наданням цим елементам класифікаційної мітки або імені того шаблону, який має найближчі характеристики у введеному понятті близькості.

Суттєвий недолік наведених етапів опрацювання зображень полягає у необхідності введення великої кількості шаблонів для зображення різних об'єктів. Також потрібно вводити шаблони для знаходження однотипних об'єктів, які на зображенні мають різні розміри, орієнтацію або деформовані. Перевага наведеної схеми опрацювання полягає у математичній обґрунтованості результатів опрацювання за заданих вимог до якості результату.

Для розв'язання задачі пошуку елементів зображень як задачі класифікації пропонуємо використати алгоритми машинного навчання [4]. Метою застосування цих алгоритмів є пошук прихованих закономірностей у базі даних, яка містить характеристики зображення. Зображення, яке пройшло опрацювання, описують спеціальними класами функцій, а знайдені закономірності задають значеннями параметрів цих функцій.

Розв'язання задачі опрацювання зображень за технологією машинного навчання містить такі етапи:

- 1) формування бази даних на основі зображень;
- 2) попереднє опрацювання даних для зменшення їх обсягу та усунення несуттєвих даних;
- 3) пошук закономірностей застосуванням алгоритмів машинного навчання;
- 4) оцінювання результатів розв'язання задачі опрацювання зображень.

Виділення елементів зображення методами машинного навчання не вимагає створення набору шаблонів. Замість цього на еталонних зображеннях виділяють елементи, які використовують для

налагодження системи опрацювання зображень. Процес налагодження називають навчанням: він полягає у застосуванні алгоритмів, які формують процес послідовних наближень параметрів спеціальних функцій. Під час розв'язання задачі опрацювання побудована система знаходить на інших зображеннях елементи, подібні до етелонних.

Останнім часом особливого поширення для аналізу зображень набули нейронні мережі [3]. Перевагою методів з використанням нейронних мереж порівняно з іншими методами (опорних векторів, головних компонент, канонічних змінних тощо) вважають те, що основою методів нейронних мереж є спеціальні класи нелінійних функцій. Для цих функцій доведено, що нейронна мережа, побудована за схемою перцептрону з двома прихованими шарами, може наближати на обмеженій області із заданою точністю довільну неперервну функцію багатьох змінних [5]. Багатошаровий перцептрон [6] з активаційними функціями сигмоїдального типу [7] є найдослідженішою схемою нейромережі. У разі практичного використання нейронних мереж потрібно розв'язати низку задач: правильно підготувати дані для налагодження нейронної мережі, обрати її топологію, вибрати активаційні функції, створити навчальні приклади та обрати алгоритм навчання. Постановка цих задач неадекватно вирішуваній проблемі та відсутність узгодженості під час розв'язання спричиняють спотворення остаточного результату.

Питання підготовки даних, які використовують для навчання мережі у разі опрацювання зображень, недостатньо висвітлено у наукових публікаціях. Якщо дані вибирають випадково, то у разі значної відмінності у кількості інформації про частини загальної задачі такий вибір призводить до некоректного опрацювання частин зображення з гіршим інформаційним поданням. У таких випадках треба зменшити відмінність в описі даних про ці частини. Цю задачу пропонують розв'язувати корегуванням значення на виході мережі за допомогою матриць втрат [8].

Для опису зображень, які опрацьовують нейронною мережею [9], використовують різноманітні методи. Найпоширенішим способом опису зображень є його попиксельне подання. Другий спосіб опису – подання зображення, як результату опрацювання спеціальними фільтрами. У разі попиксельного опису на входи нейронної мережі подають значення яскравості та кольору пікселів, а пікселі обирають з околу досліджуваної точки зображення. За другим способом опису зображення попередньо опрацьовують різноманітними фільтрами. Такими фільтрами є фільтри виділення країв, горизонтальних та вертикальних смуг, Габора тощо. Опис зображення з використанням фільтрів добре показав себе у випадку розпізнавання рукописних символів.

Серед методів навчання багатошарового перцептрону треба виділити методи зворотного поширення похибки, стохастичний, спряжених градієнтів, Левенберга–Маркара [8]. У цій статті модифіковано метод зворотного поширення похибки аналогічно до [10].

Цілі статті

Для виділення зображення кінців пальців руки доповідача на відеозображенні реального часу використано індуктивні методи штучного інтелекту, або методи машинного навчання. Основна ідея цих методів полягає у виділенні прихованих закономірностей у великих масивах даних. Ці закономірності потім використовують для побудови спеціальних процедур та правил опрацювання таких масивів даних. Зокрема, у цій статті описане застосування нейронних мереж для розв'язання задачі розпізнавання на зображенні певних елементів, якими є кінці пальців доповідача. Цю задачу розв'язували у такій послідовності:

- 1) розроблено методіку формування базової множини зображень з відеоряду реального часу;
- 2) проведено попереднє опрацювання зображень з метою усунення шумів на зображенні, які виникають внаслідок стиснення або неякісного оцифрування зображення;
- 3) запропонований підхід до вирішення задачі зменшення розмірності даних усуненням складових зображення, які не повинні впливати на процес розпізнавання;
- 4) побудований класифікатор на основі нейромережі прямого поширення сигналу – багатошарового перцептрону;
- 5) розроблене спеціалізоване програмне забезпечення, яке випробуване на реальних прикладах.

Побудова класифікатора на основі нейромережі для розпізнавання зображення кінців пальців, передбачає розв'язання задач у такій послідовності:

- 1) вибір топології мережі;
- 2) вибір активаційних функцій нейронів мережі;
- 3) вибір алгоритму навчання мережі, аналіз його застосовності до розв'язування поставленої задачі та побудова адекватної модифікації алгоритму навчання;
- 4) побудова навчальних прикладів на основі зображення;
- 5) побудова еталона, з яким у процесі виконання алгоритму порівнюють вихідний сигнал мережі та використовують для коригування ваг міжнейронних зв'язків;
- 6) формулювання критерію закінчення алгоритму навчання.

Розв'язанням вказаних задач визначають якість розпізнавання зображення, яку оцінюємо, виконуючи контрольні приклади.

Основний матеріал

Побудова системи розпізнавання жестів складається з двох основних етапів. Перший з них – розпізнавання зображення кінців пальців та виділення їх серед інших елементів, другий – власне класифікація розпізнаних комбінацій пальців, яким можна поставити у відповідність змістовні команди. У цій статті наведені результати розв'язування задач, які належать до першого з цих етапів.

За традиційним підходом для розв'язування задачі розпізнавання вводять шаблони елементів, які шукають на зображенні. Оскільки положення пальців, їх розмір та орієнтація можуть змінюватись на відеозображенні реального часу в широких діапазонах, то традиційний підхід вимагає великої кількості шаблонів, що уповільнить процес розв'язування.

Задачу розпізнавання зображення кінців пальців руки інтерпретували, як задачу класифікації елементів зображення на такі, що вважаються зображеннями кінців пальців, та на відмінні від них. Для класифікації елементів зображення використано нейромережевий класифікатор. У разі застосування методів машинного навчання відповідна програма опрацювання буде правилом розпізнавання, яке є функцією ваг міжнейронних зв'язків.

Задачу розпізнавання, як задачу класифікації, вирішують у два етапи – на першому навчають мережу на навчальних прикладах, а на другому – навчену мережу використовують для класифікації нових прикладів. Запропоновано використати нейромережу прямого поширення, яка зображена на рис. 2. Мережа має $n+1$ входів x_i , ($i = 0, 1, 2, \dots, n$), $Q-1$ прихований шар нейронів та вихідний шар, який містить один нейрон. На вхід мережі подають *приклади* – вектори чисел довжини $n+1$. На вхід x_0 завжди подають одиницю. У разі навчання мережі ці приклади називатимемо *навчальними*. Залежно від задачі, що вирішують, кількість прихованих шарів та кількість нейронів у шарі можна змінювати для підвищення якості розпізнавання. Вплив кількості прихованих шарів та кількості нейронів у кожному шарі на якість розпізнавання у цій роботі не досліджувався.

Нейронну мережу навчають за принципом “навчання з вчителем”, за яким кожному навчальному прикладу ставлять у відповідність певний зразковий елемент – еталон. Розбіжність між прикладом та еталоном використовують для уточнення міжнейронних зв'язків мережі. Тому до початку навчання необхідне створення навчальних прикладів та відповідних їм еталонів.

Навчальні приклади та еталон створювали з кадрів відеозображення. Перший крок цього створення полягав у попереднього опрацювання відеокadrів та зменшенні обсягу інформації про зображення на них. Оскільки дослідження проводились з метою створення прототипу, то здійснювалося відеознімання лише руки з мінімізацією елементів фону. Після цього з відеоряду вибирали кожний десятий кадр, який був записаний у форматі RGB. Для створення навчальних прикладів та еталонів вибрано 16 кадрів розміру 320×240 пікселів. Надалі розміри прикладів та еталонів зменшували, зменшуючи зображення до розміру 128×96 пікселів методом бікубічної інтерполяції інструментальними засобами програми Adobe Photoshop. Такий розмір зображення обраний через те, що подальше його зменшення значно ускладнило візуальне визначення кінців пальців.

Наступний крок попереднього опрацювання зображення полягав у зменшенні кількості компонент кожного з пікселів зображення. Кадри з відеокaмери у форматі RGB мають три компоненти

кольору – червону, зелену та синю. Експериментально встановлено, що жодна з цих компонент не дає змоги відрізнити руку від деталей фону. Через це запропоновано перейти до подання кольору у форматі YCbCr. Цей формат широко застосовують у комп'ютерній графіці, зокрема, у форматі JPG. Формат YCbCr містить три компоненти: Y – яскравість, Cb – насиченість синього, Cr – насиченість червоного. Перехід від формату RGB до YCbCr здійснюють за формулами:

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B;$$

$$Cb = -0,1687 \cdot R - 0,3313 \cdot G + 0,5 \cdot B + 0,5;$$

$$Cr = 0,5 \cdot R - 0,4187 \cdot G - 0,0813 \cdot B + 0,5.$$

Значення усіх компонент формату YCbCr задають з інтервалу $[0, 1]$. Встановлено, що руку найкраще відрізнити від деталей фону у компоненті Cr. Компоненту Cr тестового зображення показано на рис. 4, б, а її значення подане у градаціях сірого кольору.

Всі відеокадри було перетворено до формату YCbCr та використано лише компоненту Cr, що дало змогу втричі зменшити обсяг інформації. Додаткова перевага компоненти Cr полягає у тому, що вона несе інформацію про колір, а не про яскравість. Тому це дає можливість вирішити проблему нерівномірності освітлення. Інших методів попереднього опрацювання зображень не застосовували.

Мережу навчають на множині прикладів, побудованих на основі фрагментів відеокадрів. Навчальні приклади формують з пікселів зображення кінців пальців. Для цього на частині кадрів було позначено кінці пальців. Належність об'єкта до певного класу можна задавати значеннями 0 або 1 [11]. Оскільки кінець пальця займає на зображенні кілька пікселів, то запропоновано виділяти область на кадрі, у якій знаходимо кінець пальця, а з пікселів цієї області утворювати еталон. Для цього кожному пікселю вказаної області надане число, якому відповідає інтенсивність кольору. Це число взяте з інтервалу $[0,1]$ та має зміст ступеня належності пікселя кінцю пальця. Інтенсивність кольору задана у градаціях сірого, де 0 відповідає чорному кольору, а 1 – білому. Інтенсивність кожного пікселя можна змінювати, уточнюючи тим положення кінця пальця. Розпізнавання кінців пальців тепер розглядатимемо як класифікацію пікселів зображення на такі, які можна або не можна вважати кінцями пальців.

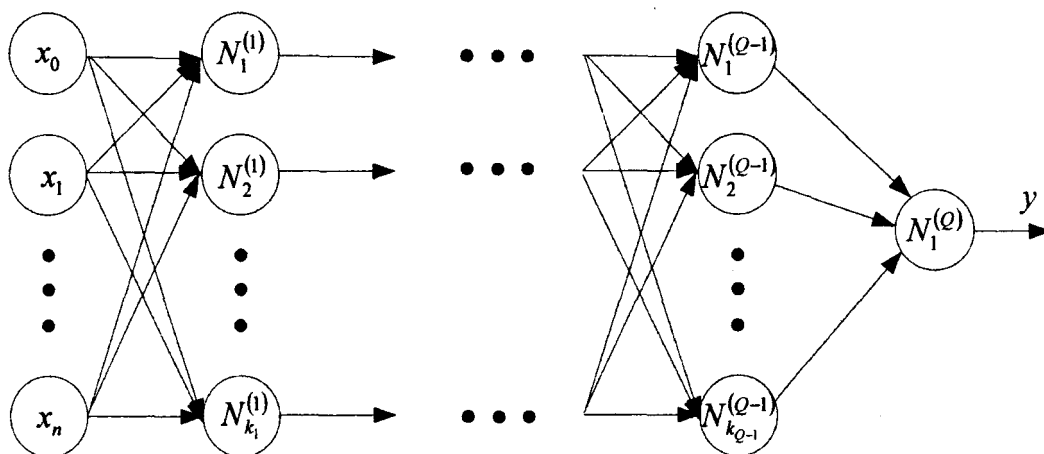


Рис. 2. Топологія нейронної мережі прямого поширення сигналу, використана для розв'язування задачі класифікації елементів зображень

Побудову прикладів виконано на основі зображень розміру 128×96 пікселів. Для цього кожний його піксель розглядаємо як такий, що його розташовано у центрі квадратного околу зі стороною у n пікселів. Для формування прикладів піксели околу проглядаємо зліва направо та згори донизу. Число n підбрано під час навчання мережі. Найкращі результати розпізнавання отримано при $n = 7$. Інтенсивність пікселя з центру околу вважаємо еталонном.

Нейронна мережа обчислює функцію, значеннями аргументів якої є компоненти прикладів, а значенням функції – ступінь належності пікселя кінцю пальця. Для розв'язування задачі розпізнавання у такій постановці використано багатошаровий перцептрон [5]. Активаційною функцією нейронів обрано непарну сигмоїдальну функцію $F(x) = \frac{1}{1 + e^{-x}} - 0.5$. Вибір такої активаційної функції нейронів за-

пропоновано [6] для уникнення уповільнення збіжності градієнтного методу мінімізації функції ваг нейронів, який є основою методу зворотного поширення похибки, використаний для навчання нейромережі. Таке уповільнення пов'язане з поганою обумовленістю матриці других похідних цієї функції [12].

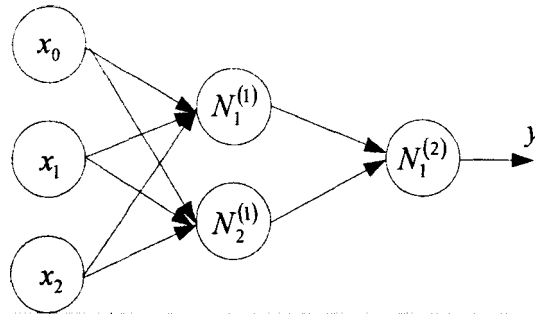


Рис. 3. Нейромережа, використана для обчислення функції XOR

Для навчання нейронної мережі побудовано модифікацію алгоритму зворотного поширення похибки, у якій, аналогічно [6], введено випадковий доданок у формулі корекції ваг міжнейронних зв'язків. Результат навчання нейромережі за побудованим алгоритмом порівнювали з результатом навчання за звичайним алгоритмом зворотного поширення похибки на прикладі обчислення функції XOR [13]. Експерименти здійснені для мережі з непарними активаційними функціями нейронів, зображеної на рис.3. Лише 20% нейромереж, навчених за звичайним методом зворотного поширення похибки, правильно обчислили функцію XOR, а навчені за модифікованим алгоритмом – у 99 спробах навчання зі ста.

Для формулювання алгоритму введемо позначення:

Q – номер вихідного шару нейронів;

q – номер шару нейронів, $q = 1, 2, \dots, Q$;

k_q – кількість нейронів у шарі q , $k_Q = 1$;

N – кількість навчальних прикладів;

T – кількість циклів навчання;

t – момент часу виконання алгоритму навчання, $t = 0, 1, 2, \dots, N \cdot T$;

$y = y(t)$ – вихід мережі – значення, обчислене мережею в момент t ;

$d = d(t)$ – еталонне значення, з яким порівнюємо вихід мережі в момент t ;

Решта параметрів та змінних визначено, якщо це не вказано явно, в момент t ;

$x_j^{(q)}$, $j = 0, 1, \dots, k_{q-1}$, – входи нейронів шару q ;

$x_0^{(q)} = 1$, $x_j^{(1)} = x_j$ – входи мережі;

$w_{ij}^{(q)}$ – вага міжнейронного зв'язку нейронів $N_i^{(q-1)}$ та $N_j^{(q)}$;

$s_j^{(q)} = \sum_{i=0}^{k_{q-1}} w_{ij}^{(q)} x_i^{(q)}$ – зважена сума входів нейрона $N_j^{(q)}$, $q = 1, 2, \dots, Q$, $j = 1, 2, \dots, k_q$;

$y_j^{(q)}$ – значення, обчислене нейроном $N_j^{(q)}$, $y_j^{(q)} = F(s_j^{(q)})$;

$x_j^{(q)} = y_j^{(q-1)}$, $j = 1, \dots, k_{q-1}$, $q = 2, 3, \dots, Q$ – входи нейронів шару q , які дорівнюють виходам нейронів шару $q-1$;

$F'(s_j^{(q)})$ – похідна активаційної функції нейрона $F(x)$ у точці $s_j^{(q)}$;

η – параметр, який регулює швидкість навчання мережі. Цей параметр вибираємо за

формулою $\eta = \frac{\eta_0(TN - t) + \eta_1 t}{TN}$, де η_0 та η_1 – параметри методу – початкова та кінцева швидкості

навчання, відповідно;

$Rand(a, b)$ – псевдовипадкове число, розподілене за рівномірним законом в інтервалі $[a, b]$.

Модифікований алгоритм зворотного поширення похибки складається з таких кроків.

Якщо це не вказане спеціально, то вважаємо, що всі параметри обчислені у момент часу t .

Крок 1. Встановити початкові ваги зв'язків $w_{ij}^{(q)} = \text{Rand}(-0.5, +0.5)$, $i = 1, 2, \dots, k_{q-1}$, $j = 1, 2, \dots, k_q$, $q = 1, 2, \dots, Q$, та покласти $t_0 = 1$.

Крок 2. Обчислити середньоквадратичну похибку мережі на всіх навчальних прикладах

$$\varepsilon = \left(\frac{\sum_{r=1}^N (y_r - d_r)^2}{N} \right)^{1/2}, \text{ де } d_r - \text{еталонне значення, яке відповідає прикладу з номером } r. \text{ Тут } y_r -$$

значення виходу мережі за подання на її вхід навчального прикладу з номером r .

Крок 3. Покласти $\tilde{w}_{ij}^{(q)} = w_{ij}^{(q)}$ та $n_0 = 1$, $i = 1, 2, \dots, k_{q-1}$, $j = 1, 2, \dots, k_q$, $q = 1, 2, \dots, Q$.

Крок 4. Подати на входи мережі довільний навчальний приклад та обчислити виходи $y_j^{(q)}$ нейронів, $q = 1, 2, \dots, Q$, $j = 1, 2, \dots, k_q$.

Крок 5. Обчислити коефіцієнт $\delta_1^{(Q)}$ для останнього шару

$$\delta_1^{(Q)} = (y_1^{(Q)} - d) F'(s_1^{(Q)}); j = 1, 2, \dots, k_Q$$

Крок 6. Обчислити коефіцієнти $\delta_j^{(q)}$

$$\delta_j^{(q)} = F'(s_j^{(q)}) \cdot \sum_{r=1}^{k_{q+1}} \delta_r^{(q+1)} w_{jr}^{(q+1)}, q = Q-1, Q-2, \dots, 1, j = 1, 2, \dots, k_q$$

Крок 7. Обчислити приріст ваг мережі $\Delta w_{ij}^{(q)} = -\eta \delta_j^{(q)} x_i^{(q)}$, $i = 1, 2, \dots, k_{q-1}$, $j = 1, 2, \dots, k_q$.

Крок 8. Перерахувати ваги зв'язків нейронів

$$w_{ij}^{(q)}(t+1) = w_{ij}^{(q)}(t) + \Delta w_{ij}^{(q)} + \text{Rand}(-\varepsilon, +\varepsilon), i = 1, 2, \dots, k_{q-1}, j = 1, 2, \dots, k_q.$$

Крок 9. Виконати кроки 4–8 для $n_0 = 2, 3, \dots, N$.

Крок 10. Обчислити $\tilde{\varepsilon} = \left(\frac{\sum_{r=1}^N (y_r - d_r)^2}{N} \right)^{1/2}$. Якщо $\tilde{\varepsilon} > \varepsilon$, то покласти $w_{ij}^{(q)} = \tilde{w}_{ij}^{(q)}$,

$i = 1, 2, \dots, k_{q-1}$, $j = 1, 2, \dots, k_q$, $q = 1, 2, \dots, Q$, інакше – покласти $\varepsilon = \tilde{\varepsilon}$.

Крок 11. Виконати кроки 3–10 для $t_0 = 2, 3, \dots, T$.

Мережу навчають поданням на її вхід навчального прикладу, вибраного випадково з множини навчальних прикладів, та порівнянням обчисленого мережею значення з відповідним йому еталоном. Приріст ваг міжнейронних зв'язків, отриманий на кроці 7, використаний для їх перерахунку на кроці 8. При розв'язуванні поставленої задачі виявилось доцільним навчати мережу лише на частині прикладів, оскільки використання всіх прикладів уповільнювало навчання.

Оскільки елементи зображення, які належить розпізнавати, займають відносно малу площу на кадрі, нейромережа вважає їх шумом і відкидає. Тому множини еталонів та відповідних їм навчальних прикладів було розбито на групи. Для цього інтервал $[0, 1]$ розбили на інтервали однакової довжини, а кожен інтервал було утворено з еталонів, які потрапляли до одного такого інтервалу. Після цього на вхід мережі подавали однакову кількість прикладів з кожної групи. Це забезпечило у процесі навчання однакове представництво всіх градацій ступеня належності.

Навчену нейромережу використано для розпізнавання зображення кінців пальців на новому зображенні. Для цього створено відповідний приклад для кожного пікселя цього зображення та подано на вхід мережі. На виході мережі отримано значення ступеня належності цього пікселя кінцю пальця. Величина ступеня належності є яскравістю цього пікселя на зображенні.

Після цього виділено найяскравіші піксели, вважаючи, що вони вказують положення кінця пальця. Піксели з яскравістю, значно меншою від максимальної, не розглядались. Серед решти пікселів вибирають найяскравіший у квадраті зі стороною 5 пікселів, у який потрапляють точки зображення одного об'єкта. Розмір квадрата підібрано так, щоб відстань між об'єктами, що розпізнаються, завжди була більшою за довжину сторони цього квадрата.

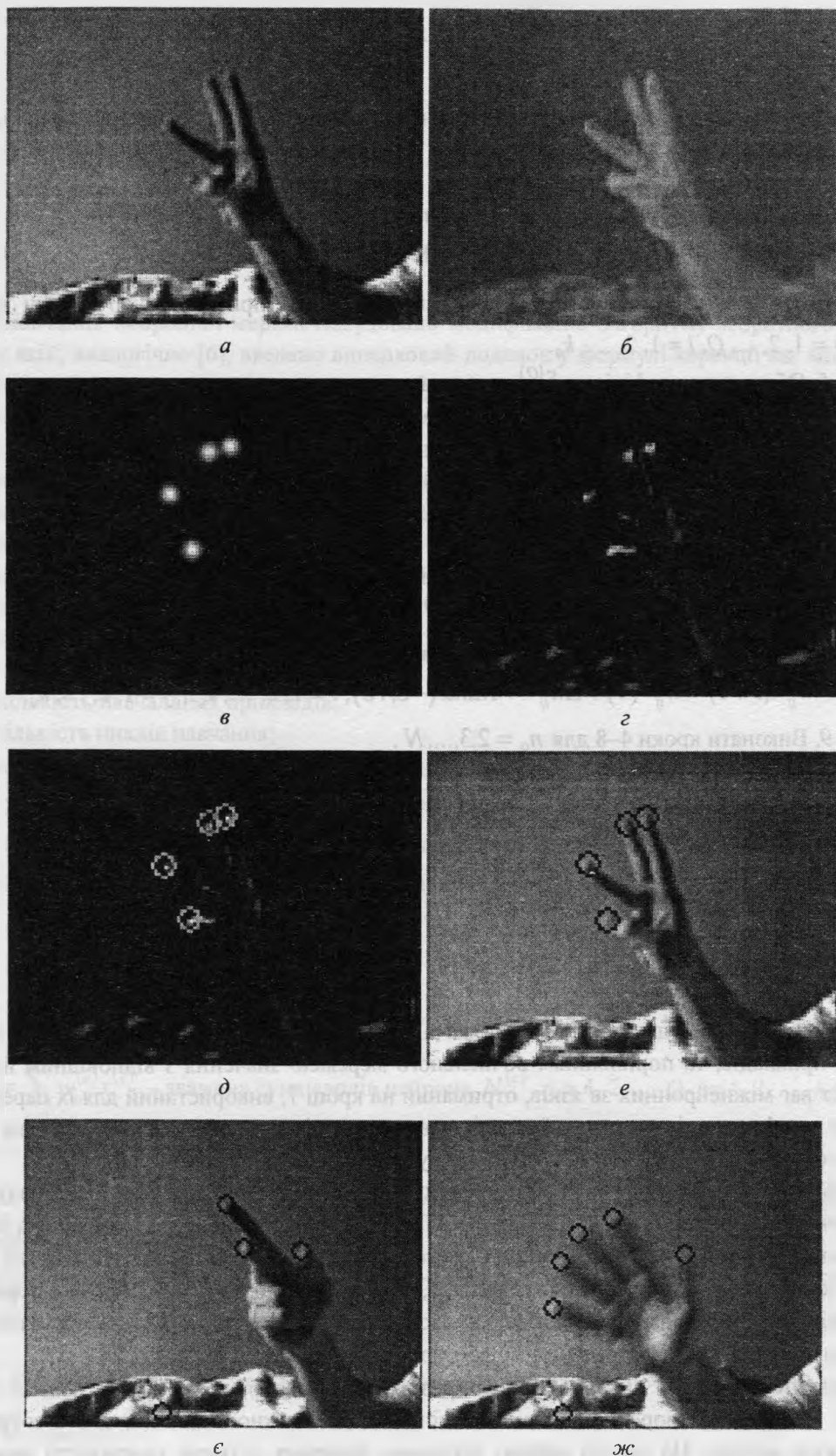


Рис. 4. Послідовність етапів розпізнавання кінців пальців

На рис. 4 показано послідовність етапів розпізнавання кінців пальців. На рис. 4, а початкове зображення – один кадр з відеоряду, отриманого з відеокамери, після зменшення його до розміру 128×96 пікселів та використаного для побудови навчальних прикладів. На рис. 4, б показано компоненту C_r кольорового простору $YCbCr$, виділену з заданого зображення та подану у градаціях сірого. Максимальному значенню цієї компоненти відповідає білий колір, а мініимальному – чорний. На рис. 4, в видалене зображення руки, а залишено області, якими окреслені кінці пальців. Величина яскравості у межах кожної області визначає положення кінця пальця – білий колір відповідає найімовірнішому його положенню. Області зроблено розмитими, оскільки точно визначити, що вважати кінцем пальця, неможливо. На рис. 4, г показано результат розпізнавання кінців пальців на зображенні з рис. 4, б застосуванням навченої нейронної мережі. На рис. 4, д позначено області з максимальною яскравістю, які знайдені на рис. 4, г. Знайдені на попередньому рисунку максимумами яскравості накладені на початкове зображення для перевірки правильності розпізнавання. Результат такого накладання показаний на рис. 4, е. Можна зауважити, що мережа правильно розпізнала кінці пальців. Мережу, навчену на зображенні, показаному на рис. 4, а, застосовано для розпізнавання кінців пальців на інших кадрах. Результати такого розпізнавання показано на рис. 4, є та 4, ж.

Висновки

Запропонований прототип системи керування доповідями не вимагає спеціальної обчислювальної та дорогої відеотехніки. Для роботи системи розпізнавання достатньо веб-камери із роздільною здатністю 320×240 пікселів та комп'ютера з процесором Celeron 1,2 МГц. Як показали проведені експерименти з вибору топології мережі та параметрів навчання, задовільні результати отримані для нейронної мережі, яка має один прихований шар з 5 нейронами. Час опрацювання одного кадру становив 0,15с.

Основними результатами виконаної роботи можна вважати метод побудови еталону та навчальних прикладів введенням ступеня належності пікселів зображення шуканому об'єкту. Розроблений новий алгоритм утворення навчальних прикладів розбиттям їх множини на групи за значенням еталону та вибором однакової кількості прикладів з кожної групи, що дало змогу збільшити якість навчання мережі. Час навчання нейронної мережі на 16 навчальних зображеннях, з яких створено $16 \cdot 128 \cdot 96 = 196608$ навчальних прикладів, становив 15 секунд.

Тестування системи показало високу якість розпізнавання. Під час тестування системи нею розпізнано всі кінці пальців. Аналіз об'єктів, які навчена нейромережа розпізнала як кінці пальців, показав, що 80 % з них дійсно є кінцями пальців. Решту 20 % становили інші елементи зображення. Отриманий результат вимагає додаткових досліджень з метою усунення цієї особливості роботи системи.

Майбутні дослідження спрямовуватимуться на пошук оптимальних параметрів системи розпізнавання. Також досліджуватимуть вплив параметрів алгоритму навчання мережі на якість розпізнавання, на вибір оптимального співвідношення між топологією мережі, якістю та швидкістю навчання.

1. Белозерский Л.А. Основы построения систем распознавания образов – Донецкий Государственный институт искусственного интеллекта, 1997. 2. Русин Б.П. Системы синтеза, обработки та розпізнавання складноструктурованих зображень – Львів: Вертикаль, 1997. – 264 с. 3. Форсайт Д.А., Понс Ж. Компьютерное зрение, современный подход. – М.: Издательский дом "Вильямс", 2004. – 928с. 4. Mitchell T. Machine Learning. McGraw-Hill Companies, Inc. 1997. 5. Горбань А.Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сибирский журнал вычислительной математики. – 1998. – Т. 1.– № 1. – С.12-24. 6. Круглов В. В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с. 7. Fravolini M. L., Campa G., Napolitano M., La Cava M. Comparison of Different Growing Radial Basis Functions Algorithms for Control Systems Applications. 2000. <http://www2.cemr.wvu.edu/~gcampa/pdf/acc02f8.pdf>. 8. Bishop C. Neural Networks for Pattern Recognition. Oxford University Press. 1995. 9. LeCun Y., Matan O., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D., Baird H.S. Handwritten Zipcode Recognition With Multilayer

Networks, Proc. of International Conference on Pattern Recognition, Atlantic City, 1990. 10. Rowley H.A., Baluja S., Kanade T. Rotation invariant neural-network based face detection. – Proceedings, Computer Vision and Pattern Recognition, 1998. 11. Мисюров А. В. Практика використання штучних нейронних мереж для розпізнавання рукодрукованих символів. <http://www.ocrai.narod.ru/>. 12. Пшеничный Б.Н., Данилин Ю.М. Численные методы в экстремальных задачах. – М.: Наука, 1975. 13. Jain A.K., Jianchang Mao, Mohiuddin K.M. Artificial Neural Networks: A Tutorial. Computer. Vol.29, No.3, March 1996, pp. 31-44.

УДК 004.413

П.І. Жежнич

Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

РОЗШИРЕННЯ ER-ДІАГРАМ ДЛЯ ПРОЕКТУВАННЯ ЧАСОВИХ БАЗ ДАНИХ

© Жежнич П.І., 2005

Розглянуто підходи до розширення існуючих ER-діаграм нотації Баркера для проектування часових реляційних баз даних. Запропоноване розширення є сумісним з традиційними ER-діаграмами і забезпечує відображення довільних часових аспектів бази даних.

The approaches to existing Barker’s ER-diagrams extension for temporal database design are proposed. The extension is compatible with traditional ER-diagrams. It provides proper representation of any time aspects of databases.

Постановка проблеми

Модель “сутність–зв’язок” (Entity-Relationship, ER), незважаючи на різноманітні синтаксиси та деякі семантичні відмінності, є досить поширеною як в дослідницькій сфері, так і в області розробки програмного забезпечення. Ця модель є простою для сприйняття та використання. ER-діаграми забезпечують наочне відображення проекту бази даних. Вони зосереджують увагу на структурних аспектах схеми бази даних, а не на аспектах поведінки, що дає змогу їх широко використовувати для документування програмного забезпечення з пристосуванням для багатьох користувачів.

ER-модель можна використовувати для вирішення декількох споріднених задач

- Аналіз – моделювання предметної області (ПО).
- Проектування – опис схеми бази даних для комп’ютерної системи.
- Реалізація – створення схеми бази даних за допомогою СКБД, яка безпосередньо підтримує ER-модель.

Часові аспекти схем баз даних складно відображаються за допомогою ER-діаграм, оскільки при повному моделюванні часових аспектів фактор часу приводить до невизначеностей та поганого сприйняття ER-діаграм. Тому деякі розробники просто ігнорують всі часові аспекти їх ER-діаграм, залишаючи лише спеціальні коментарі. Отже, неможливе автоматизоване відображення ER-діаграм до реляційних таблиць бази даних, і врахування часових аспектів відбувається вручну. Такі ER-діаграми належно не документують схеми часових реляційних баз даних і не є прийнятними для розробників програмного забезпечення.

Дослідниками було запропоновано понад 10 часових ER-моделей, основними серед яких є [7]:

- Часова розширена ER-модель.
- Часова ER-модель.
- Модель зв’язків, атрибутів, ключів та сутностей.

У цих моделях розглядаються два основні незалежні часові аспекти: дійсний час та час транзакції [7]. Дійсний час події ПО – це час, коли подія відбулася і впливає на стан предметної