

ЛАБОРАТОРНА РОБОТА №7

Розпізнавання образів

Тема роботи: методи розпізнавання зображень.

Мета роботи: дослідити та реалізувати методи розпізнавання зображень.

Теоретичні відомості

Образом називається деяка впорядкована сукупність *дескрипторів*, які в літературі по розпізнаванню часто називають *ознаками*. *Класом образів* (або просто *класом*) називається сукупність образів, що володіють деякими спільними рисами (властивостями) – однаковими ознаками.

Набули поширення три форми впорядкованого представлення ознак. Зокрема у вигляді:

- векторів ознак (для кількісних дескрипторів);
- символьних рядків;
- дерев.

На практиці задача класифікації найчастіше зводиться до побудови деякої гіперплощини або множини гіперплощин в просторі ознак, щоб з їх допомогою можливо було визначити якому класу належить задана множина ознак.

Класифікатор за мінімумом відстані

Прототип кожного класу визначається як вектор математичного очікування образів даного класу:

$$\mathbf{m}_j = \frac{1}{N} \sum_{x \in w_j} \mathbf{x}_j, j = 1, 2, \dots, W$$

де \mathbf{x}_j – вектори ознак образів j -го класу, N – кількість образів, W – кількість класів. Підсумовування ведеться по всіх таких векторах ознак об'єктів класу

j . Розділююча поверхня між класами ω_1 , і ω_2 у випадку використання класифікатора за мінімумом відстані задається рівнянням:

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = \mathbf{x}^T (\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m}_j) = 0.$$

Кореляційне співставлення

Для пошуку об'єкта на зображенні часто використовується кореляція. В безпосередньому вигляді її застосовують доволі рідко, оскільки вона чутлива до зміни яскравості порівнюваних зображень і з огляду на це не дозволяє досягти достатньої надійності. Натомість більшого розповсюдження набув коефіцієнт кореляції, що обчислюється за формулою:

$$\gamma(x, y) = \frac{\sum_s \sum_t [f(s, t) - \bar{f}(s, t)] \cdot [w(x + s, y + t) - \bar{w}]}{\sqrt{\sum_s \sum_t [f(s, t) - \bar{f}(s, t)]^2 \cdot \sum_s \sum_t [w(x + s, y + t) - \bar{w}]^2}},$$

де $x = 0, 1, 2, \dots, M-1$, $y = 0, 1, 2, \dots, N-1$, \bar{w} – середнє значення пікселів шаблону w (обчислюване тільки один раз), \bar{f} – середнє значення елементів зображення f в області, що збігається з поточним положенням w , а підсумовування ведеться за всіма парам координат, спільним для f та w . Коефіцієнт кореляції змінюється в діапазоні $\gamma(x, y) \in [-1; +1]$ і не залежить від зміни масштабу яскравостей f та w .

В системі MATLAB коефіцієнт кореляції між двовірними даними (зображеннями) обчислюється за допомогою функції `normxcorr2`. При використанні Python можна скористатись функцією `matchTemplate` пакету OpenCV або функцією `match_template` пакету `skimage.feature`.

Штучні нейронні мережі

Елементарною коміркою нейронної мережі є *нейрон*. Структура простого нейрона показана на рис. 7.1. Він має вектор входу \mathbf{p} , що містить R елементів p_1, p_2, \dots, p_R . Кожен вхідний елемент перемножується на

відповідні вагові коефіцієнти w_1, w_2, \dots, w_R , після чого отримані добутки додаються. Таким чином, якщо розглядати вагові коефіцієнти як вектор-рядок \mathbf{W} , то результат підсумовування зважених входних сигналів є не чим іншим, як скалярним добутком між векторами \mathbf{W} та \mathbf{p} .

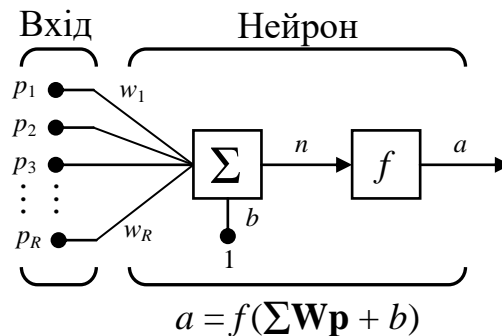


Рис. 7.1. Нейрон з векторним входом

Нейрон має зміщення b , яке підсумовується разом із зваженою сумою входів. Таким чином, сума n набуває вигляду $n = w_1 p_1 + w_2 p_2 + \dots + w_R p_R + b$ і є рівнянням гіперплощини в R -мірному просторі, причому ваги визначатимуть нахил даної гіперплощини, а зміщення – її положення. Сума n передається як аргумент до *функції активації* f , яка визначає кінцеву реакцію нейрона. Найбільш поширені функції активації наведені на рис. 7.2. Відмітимо, що в сучасних мережах у проміжних шарах найбільш часто використовуються функції типу ReLU д) та е), а у вихідному шарі функція для отримання *категорійного розподілу* softmax.

Реальна нейронна мережа може містити один або більшу кількість шарів. Одношарові нейронні мережі мають суттєве обмеження – вони можуть виконувати розпізнавання *тільки лінійно-роздільних образів*, що звужує їх застосування. Тому в більшості реальних задач розповсюдження набули багатошарові нейронні мережі, одна з яких показана на рис. 7.3. Ця мережа має R входів, S^1 нейронів в першому шарі, S^2 нейронів в другому шарі і так далі. При цьому різні шари можуть мати різне число нейронів. Виходи кожного проміжного шару виступають входами для наступного шару. Матриці ваг входного шару і прихованих (проміжних) шарів позначено \mathbf{IW} (Input Weight) і \mathbf{LW} (Layer Weight) відповідно.

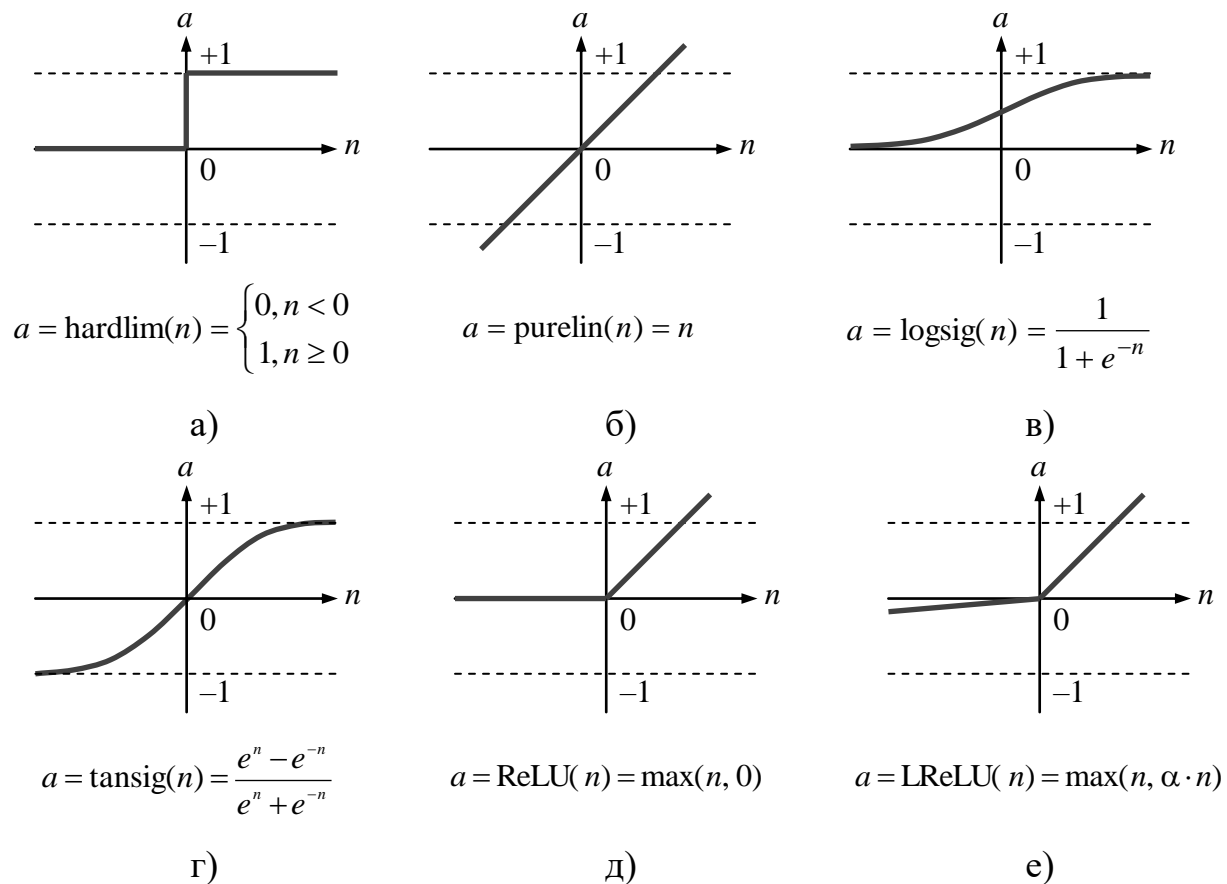


Рис. 7.2. Найбільш поширені функції активації:

- а) функція Хевісайда;
- б) лінійна функція активації;
- в) логістична функція активації (сигмоїд);
- г) тангенс гіперболічний;
- д) функція ReLU (Rectified Linear Unit);
- е) функція Leaky ReLU (LReLU)

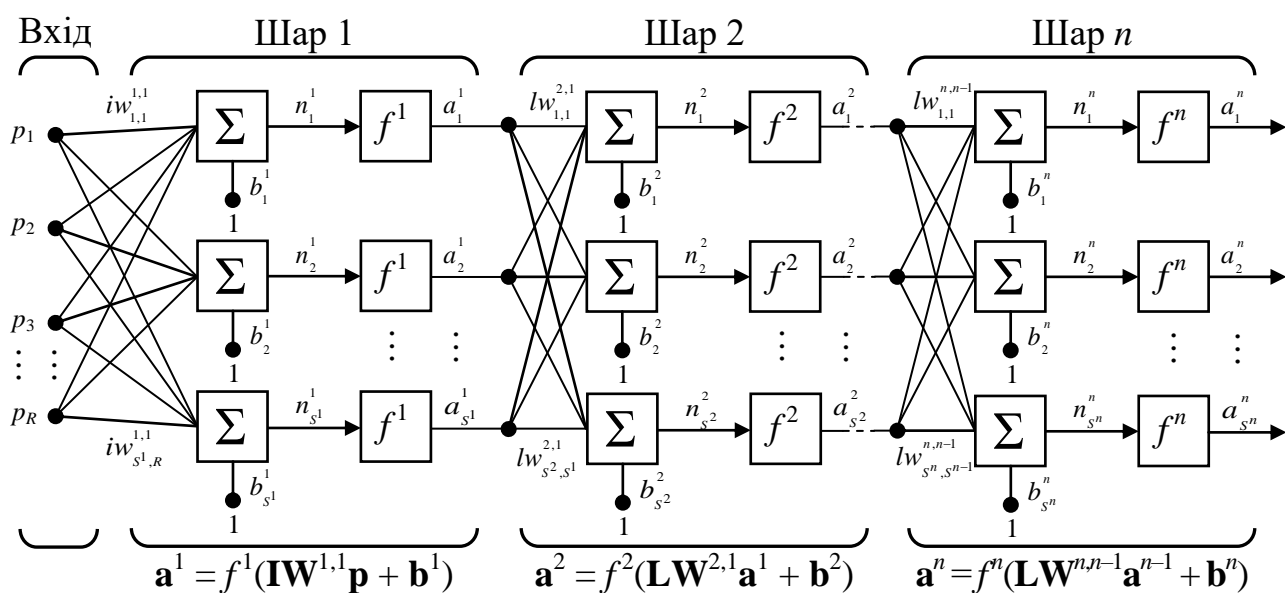


Рис. 7.3. Повнозв'язна багатошарова нейронна мережа

Навчання сучасних багатошарових нейронних мереж здійснюється методом *зворотного розповсюдження помилки (backpropagation)*. Коротко суть даного алгоритма полягає в тому, що мережі на вхід подається деякий образ(и) і оцінюється її реакція – значення на виході. Далі за допомогою спеціальної функції – *функції втрат* визначається *помилка мережі* – оцінка того, наскільки мережа дає некоректний відгук на даних, що були подані на її вхід. На основі знайденої величини помилки, починаючи з вихідного шару розраховується *градієнт помилки* відносно параметрів кожного нейрона. Для оцінки градієнтів параметрів у проміжних (прихованих) шарах використовується *ланцюгове правило диференціювання*, при цьому також оцінюються градієнти значень виходів нейронів у проміжних шарах. Знайдені градієнти можуть усереднюватись по певній невеликій підвибірці (minibatch) навчальних образів. Далі, на основі знайдених градієнтів, користуючись методом *стохастичного градієнтного спуску* (Stochastic Gradient Descent – SGD) виконується переналаштування параметрів нейронів, так щоб мінімізувати помилку мережі.

Серед функцій втрат (помилки), що використовуються при розпізнаванні образів, найбільшого розповсюдження набули: сума квадратів різниць, функція втрат SVM (hinge loss) та взаємна-ентропія. В загальному випадку функція втрат може бути будь-якою і залежить від конкретної задачі, що вирішується.

Найбільш розповсюдженими модифікаціями методу стохастичного градієнтного спуску (SGD), що застосовуються при навчанні сучасних нейронних мереж є: стохастичний градієнтний спуск з імпульсом (momentum), метод Нестерова, AdaGrad, RMSProp та Adam.

На сьогодні повнозв'язні нейронні мережі, які мають структуру подібну до тої, що показана на рис. 7.3 в чистому вигляді майже не використовуються. Натомість в сучасних задачах комп'ютерного зору застосовуються *глибокі* нейронні мережі зі згортковими шарами. Глибокими

умовно вважаються мережі, які мають щонайменше 5-7 шарів. На рис. 7.4 показано один з найпростіших варіантів мереж даного типу.

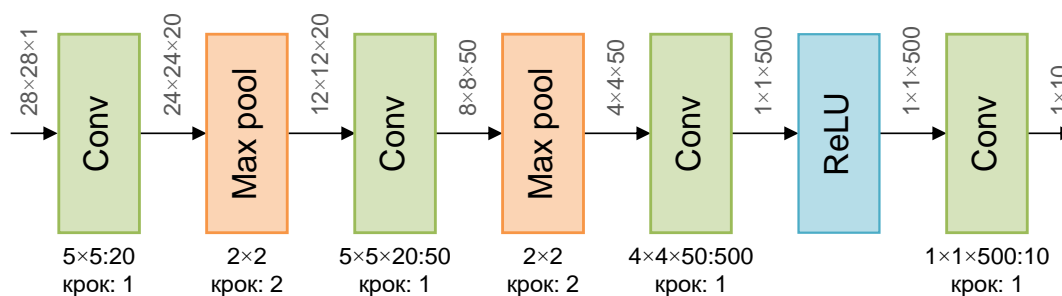


Рис. 7.4 – Проста LeNet-подібна згорткова нейронна мережа

У наведеній вище штучній нейронній мережі використані наступні позначення: *Conv* – згорткові шари, *Max pool* – шари об'єднання за максимумом, *ReLU* – функція активації $a = \max(n, 0)$. Позначення під кожним із шарів мають формат: $x \times y \times z : n$, $x \times y : n$ та $x \times y$ і означають наступне: $x \times y \times z$ та $x \times y$ – розміри фільтру в шарі; n – кількість фільтрів у шарі; *крок* – зміщення фільтрів у пікселях під час обчислення згортки або при об'єднанні за максимумом. Над стрілочками, що сполучають шари, стоять розмірності масивів (тензорів) даних, які передаються між відповідними шарами.

Навчання та тестування згорткових нейронних мереж в середовищі MATLAB можна виконувати за допомогою зовнішнього безкоштовного пакету, що має назву MATCONVNET. Даний пакет необхідно скачати, а потім скопіювати згідно інструкцій наданих на офіційному сайті MATCONVNET (<http://www.vlfeat.org/matconvnet/>). Для роботи із сучасними нейронними мережами у Python існує чимало готових пакетів, серед яких можна обрати будь-який – за власним бажанням. Тим не менш рекомендованими є пакети PyTorch (<https://pytorch.org>) або TensorFlow (<https://www.tensorflow.org>).

Створення звичайних повнозв'язних багатошарових мереж (багатошарових перцептронів) у системі MATLAB можна здійснювати функцією **feedforwardnet** (в старих версіях **newff**), їх навчання виконувати функцією **train**, а застосування вже навченої мережі – функцією **sim**.

Зауважимо, що для надійного навчання нейронних мереж їх навчальні вибірки мають містити в залежності від задачі від *декількох десятків тисяч до декількох десятків мільйонів* різних прикладів.

Порядок виконання роботи

1. Згідно наведених нижче завдань здійснити розпізнавання зображень.
2. Дослідити як параметри методів розпізнавання впливають на отримуваний результат.
3. Представити процедури розпізнавання зображень у вигляді файла-скрипта.

Завдання

Виконати обробку заданих зображень:

1. Вибрати неоднорідну область на вихідному зображенні та зберегти її. Спотворити вихідне зображення додавши шуми, незначне обертання, афінні (або проєктивні) деформації, тощо. Здійснити пошук збереженої області на спотвореному зображенні за допомогою кореляційного співставлення (файл – cameraman.tif, що входить до складу MATLAB).
2. В обраному пакеті для навчання нейронних мереж (MATCONVNET, PyTorch, Tensorflow чи іншому) створити штучну згорткову нейронну мережу (подібну до тої, що показана на рис. 7.4) для розпізнавання рукописних цифр на навчальній базі MNIST¹ (всі перелічені пакети для навчання нейронних мереж мають у своєму складі відповідний приклад). Протестувати навчену мережу на прикладах.
3. За бажанням дослідити інші наявні в MATLAB методи машинного навчання для розпізнавання зображень.

¹ Базу MNIST можна завантажити за посиланням: <http://yann.lecun.com/exdb/mnist/>

Запитання для самоконтролю

1. Наведіть означення понять «образ», «ознака» та «клас». В якому вигляді зазвичай представляють ознаки.
2. У чому полягає суть розпізнавання основанийого на співставленні. Назвіть метрики (міри) відстані, що набули найбільшого застосування при обробці зображень.
3. Як здійснюється класифікація за мінімумом відстані. Що є прототипом образу в даному виді класифікатора.
4. В чому полягає перевага коефіцієнту кореляції в порівнянні зі звичайною кореляцією.
5. Одношаровий персептрон – в чому суть процедури навчання. Який основний недолік одношарового персептрону.
6. Наведіть найбільш поширені функції активації штучних нейронів.
7. Наведіть рівняння в матричному вигляді, за яким обчислюється відгук багатошарової нейронної мережі (для спрощення достатньо буде навести рівняння для двошарової або тришарової мережі).
8. За якою процедурою здійснюється навчання багатошарового персептрона.
9. Наведіть найбільш розповсюджені види нейронних мереж.
10. Опишіть інші відомі Вам методи машинного навчання.