

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВНУТРІШНІХ СПРАВ
КРЕМЕНЧУЦЬКИЙ ЛЬОТНИЙ КОЛЕДЖ**

Циклова комісія технічного обслуговування авіаційної техніки

ТЕКСТ ЛЕКЦІЇ

**з навчальної дисципліни «Інформаційні системи контролю та діагностики газотурбінних двигунів»
обов'язкових компонент освітньо-професійної програми
першого (бакалаврського) рівня вищої освіти
Технічне обслуговування та ремонт повітряних суден і авіадвигунів**

за темою – Навчання штучних нейронних мереж.

Харків 2022

ЗАТВЕРДЖЕНО

Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від 30.08.2022 № 8

СХВАЛЕНО

Методичною радою Кременчуцького
льотного коледжу Харківського наці-
онального університету внутрішніх
справ
Протокол від 22.08.2022 № 1

ПОГОДЖЕНО

Секцією Науково-методичної ради
ХНУВС з технічних дисциплін
Протокол від 29.08.2022 № 8

Розглянуто на засіданні циклової комісії технічного обслуговування авіаційної техніки, протокол від 10.08.2022 № 1

Розробник: старший викладач циклової комісії технічного обслуговування авіаційної техніки, к.т.н., спеціаліст вищої категорії, викладач-методист, Владов С.І.

Рецензенти:

1. Завідувач кафедри транспортних технологій Кременчуцького національного університету імені Михайла Остроградського, доктор технічних наук, професор Мороз М.М.
2. Викладач циклової комісії аеронавігації Кременчуцького льотного коледжу Харківського національного університету внутрішніх справ, кандидат технічних наук, старший науковий співробітник Тягній В.Г.

План лекції

1. Поняття про навчання ШНМ.
2. Правило навчання Гебба (корелятивне, співвідносне навчання).
3. Дельта-правило.
4. Градієнтні методи навчання.
5. Одношаровий персептрон.
6. Будова персептрона.
7. Навчання персептрона.
8. Алгоритм зворотного поширення помилки навчання багатошарових ШНМ прямого поширення.

Рекомендована література:

Основна

1. Нечипоренко О. М. Основи надійності літальних апаратів : навчальний посібник. Київ : НТУУ «КПІ», 2010. 240 с.
2. Глибовець М. М., Олецький О. В. Системи штучного інтелекту. Київ : КМ Академія, 2002. 366 с.
3. Литвин В. В., Пасічник В. В., Яцишин Ю. В. Інтелектуальні системи : підручник. Львів: «Новий Світ – 2000», 2020. 406 с.

Допоміжна

4. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Запоріжжя : ЗНТУ, 2008. 341 с.
5. Руденко О. Г., Бодянський Є. В. Штучні нейронні мережі : навчальний посібник. Харків: ТОВ «Компанія СМІТ», 2006. 404 с.
6. Нестеренко О. В., Савенков О. І., Фаловський О. О. Інтелектуальні системи підтримки прийняття рішень : навчальний посібник. Київ : Національна академія управління, 2016. 188 с.
7. Вахнюк С.В. Технологія створення програмних та інтелектуальних систем: навчальний посібник. Суми : УАБС НБУ, 2011. 254 с.
8. Шаров С. В., Лубко Д. В., Осадчий В. В. Інтелектуальні інформаційні системи : навчальний посібник. Мелітополь: Вид-во МДПУ ім. Б. Хмельницького, 2015. 144 с.

ТЕКСТ ЛЕКЦІЇ

8.1. Поняття про навчання ШНМ

Характерною властивістю ШНМ є її здатність до навчання, що полягає у *виробленні правильної реакції на подані їй різні вхідні сигнали*. Існують такі можливості навчання ШНМ:

- зміна конфігурації мережі шляхом утворення нових або виключення деяких існуючих зв'язків між нейронами;
- зміна елементів матриці зв'язку (ваг);
- зміна характеристик нейронів (виду й параметрів активаційної функції й т. д.).

Найбільшого поширення сьогодні отримав підхід, при якому структура мережі задається апріорно, а мережа навчається шляхом настроювання матриці зв'язків (вагових коефіцієнтів) W . Від того, наскільки вдало побудована ця матриця, залежить ефективність даної мережі. У цьому випадку навчання полягає у зміні за певною процедурою елементів матриці W при послідовному поданні мережі деяких векторів, що навчають.

У зв'язку з цим штучний нейрон може бути представлений у такий спосіб (рис. 8.1).



Рис. 8.1. Модель штучного нейрона

У процесі навчання ваги стають такими, що під час надходження вхідних сигналів мережа виробляє відповідні необхідні вихідні сигнали. Розрізняють навчання з учителем і без учителя. Перший тип навчання припускає, що є «учитель», що задає пари, які навчають — для кожного вхідного вектора, що навчає, необхідний вихід мережі. Для кожного вхідного вектора, що навчає, обчислюється вихід мережі, порівнюється з відповідно необхідним, визначається помилка виходу, на основі якої й коректуються ваги. Пари, що навчають, подаються мережі послідовно й ваги уточнюються доти, поки помилка за такими парами не досягне необхідного рівня.

Цей вид навчання неправдоподібний з біологічної точки зору. Дійсно, важко уявити зовнішнього «учителя» мозку, що порівнює реальні й необхідні реакції того, кого навчають, і коригує його поведінку за допомогою негативного зворотного зв'язку. Більш природним є навчання без учителя, коли мережі подаються тільки вектори вхідних сигналів, і мережа сама, використовуючи деякий алгоритм навчання, підстроювала б ваги так, щоб при поданні їй досить близьких вхідних векторів вихідні сигнали були б однаковими. У цьому випадку в процесі навчання виділяються статистичні властивості множини вхідних векторів, що навчають, і відбувається об'єднання близьких (подібних) векторів у класи. Подання мережі вектора з даного

класу викликає її певну реакцію, яка до навчання є непередбаченою. Тому в процесі навчання виходи мережі мають трансформуватися в деяку зрозумілу форму. Це не є серйозним обмеженням, оскільки зазвичай нескладно ідентифікувати зв'язок між вхідними векторами й відповідною реакцією мережі.

Існує ще один вид навчання — з підкріпленням (reinforcement learning), при якому також передбачається наявність учителя, що не підказує, однак, мережі правильної відповіді. Учитель тільки повідомляє, правильно чи неправильно відпрацювала мережа поданий образ. На основі цього мережа корегує свої параметри, збільшуючи значення ваг зв'язків, що правильно реагують на вхідний сигнал, і зменшуючи значення інших ваг.

Сьогодні існує велика кількість алгоритмів навчання. Деякі з них розглядатимуться пізніше, тут же коротко зупинимося на найбільш відомих.

8.2. Правило навчання Гебба (корелятивне, співвідносне навчання)

Більшість сучасних алгоритмів навчання виросло із правила Гебба. Наприкінці 40-х років XX ст. років Д. О. Гебб теоретично встановив, що асоціативна пам'ять у біологічних системах викликається процесами, що змінюють зв'язки між нервовими клітинами. Відповідно до установленого їм правила, що називається «правилом Гебба», при одночасній активації (порушенні) двох нейронів синаптична сила (вага їхнього зв'язку) зростає. Таким чином, часто використовувані зв'язки в мережі підсилюються, що пояснює феномен звички й навчання повторенням.

У ШНМ зростання синаптичної сили еквівалентне збільшенню ваги зв'язку між нейронами i та j на величину

$$\Delta w_{ij} = \gamma x_i y_j,$$

де x_i — вихід i -го та вхід j -го нейронів; y_j — вихід j -го нейрона; γ — коефіцієнт, що впливає на швидкість навчання.

Правило Гебба використовується у зв'язках асоціативної пам'яті, а також у деяких інших, заснованих на навчанні без учителя (без підкріплення). У мережах асоціативної пам'яті приймають $y = x$. У гетероасоціативних мережах x й y в загальному випадку різняться.

8.3. Дельта-правило

Це важливе правило навчання було запропоновано Б. Уїдроу й М. Е. Гоффом і найбільше відповідає одношаровим ШНМ прямого поширення. Ідея його полягає в тому, що якщо під час навчання мережі можна встановити розбіжність між її бажаною й наявною реакціями, ця розбіжність може бути усунута або зменшена шляхом зміни певним чином вагових коефіцієнтів зв'язку. Для цього й використовується дельта-правило, відповідно до якого зміна ваги зв'язку між i -м й j -м нейронами визначається у такий спосіб:

$$\Delta w_{ij} = \gamma x_i (y_j^* - y_j),$$

де x_i — вихід попереднього i -го нейрона; y_j^*, y_j — бажана й реальна реакції j -го нейрона відповідно; γ — коефіцієнт, що впливає на швидкість навчання.

Якщо різниця $(y_j^* - y_j)$ мала, тобто реакція j -го нейрона незначною мірою відрізняється від бажаної, зміна ваги зв'язку між цими нейронами також буде

незначною.

8.4. Градієнтні методи навчання

Багато методів навчання засновано на мінімізації деякої цільової (вартісної, енергетичної й т. д.) функції I , що являє собою звичайно деяку опуклу функцію. Якщо використовувані функції активації $f(\cdot)$ диференційовані, зручно застосовувати градієнтні методи мінімізації. У цьому випадку корекція ваг зв'язку між i -м й j -м нейронами відбувається за правилом

$$\Delta w_{ij} = -\gamma \nabla_{\mathbf{w}} I(\mathbf{w}),$$

де γ — коефіцієнт, що впливає на швидкість навчання;

$$\nabla_{\mathbf{w}} I(\mathbf{w}) = \frac{\partial I(\mathbf{w})}{\partial w_{ij}}.$$

Ці методи найчастіше використовуються при контрольованому навчанні, коли відома необхідна реакція нейронів \mathbf{y}^* .

Більшість градієнтних методів засновано на мінімізації квадратичного функціонала

$$I(\mathbf{w}) = 0,5e^2(k) = 0,5(y^*(k) - \mathbf{w}^T(k)\mathbf{x}(k))^2.$$

У цьому випадку

$$\nabla_{\mathbf{w}} I(\mathbf{w}) = -e(k)\mathbf{x}(k).$$

Таким чином приходимо до алгоритму методу найменших квадратів (МНК)

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \gamma e(k)\mathbf{x}(k).$$

Існують різні рекомендації з вибору γ . Так, у теорії стохастичної апроксимації, що вивчає особливості роботи алгоритмів такого типу за наявності завад вимірів, цей коефіцієнт вибирається змінним і таким, що задовольняє умовам Дворецького

$$\lim_{k \rightarrow \infty} \gamma(k) = 0, \sum_{k=1}^{\infty} \gamma(k) = \infty, \sum_{k=1}^{\infty} \gamma^2(k) < \infty,$$

зміст яких полягає в тому, що для забезпечення збіжності послідовності у деяку точку \mathbf{w}^* довжина кроку $\gamma(k)$ має з одного боку спадати досить повільно (для забезпечення власне збіжності), а з іншого — досить швидко (з метою придушення завад). Таким умовам відповідає, наприклад, ряд $\gamma(k) = \gamma_0 k^{-\alpha}$, де γ_0 — деяка константа, $0,5 < \alpha \leq 1$. У теорії стохастичної апроксимації немає рекомендацій з вибору константи γ_0 , крім її позитивності. Теорія оптимальної фільтрації, тісно пов'язана з теорією стохастичної апроксимації, дозволяє вибрати цю константу, що виявляється залежною від статистичних характеристик сигналів і завад.

8.5. Одношаровий перцептрон

8.5.1. Будова перцептрона

Значний інтерес до перцептронів викликаний роботою Ф. Розенблатта, у якій він

досліджував нейромережеву модель сітківки (RETINA) — фотоперцептрон. Згодом такий підхід широко використовувався для моделювання обробки оптичних сигналів. Фотоперцептрон зображений на рис. 8.2 і складається, відповідно до концепції Розенблатта, із трьох шарів, що послідовно здійснюють попередню обробку (розбивання) образу, оцінку його характеристик і розпізнавання:

- сітківка (RETINA);
- асоціативний шар;
- вихідний шар.

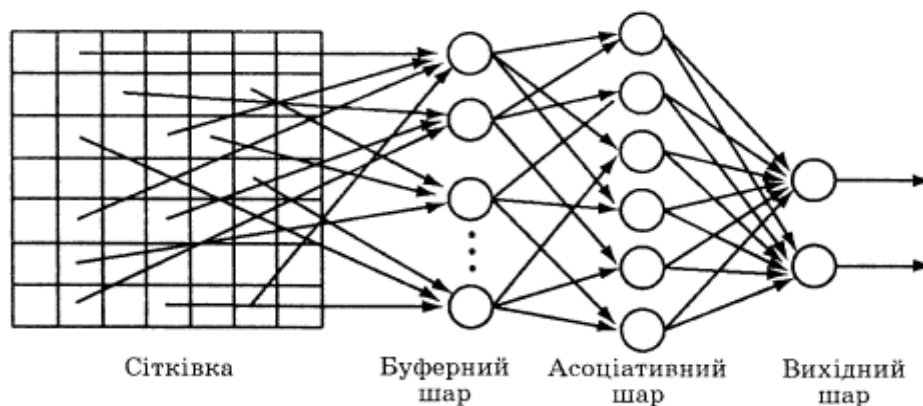


Рис. 8.2. Перцептрон Розенблатта

Попередня обробка образу не залежить від його виду. Однак результат цієї обробки має забезпечити можливість розпізнавання образів на основі аналізу їхніх характеристик. Нарешті, вихідний шар (класифікатор) аналізує характеристики знову пропонованого образу й установлює його відповідність одному з раніше поданих.

Сигнали першого шару, сітківки, подані у двійковій формі, надходять на асоціативний шар, причому в загальному випадку не всі нейрони першого шару пов'язані з усіма нейронами другого шару. При встановленні цих зв'язків виникає можливість структуризації вхідних даних, тобто виділення й об'єднання в так звані рецептивні поля найбільш важливих ознак (областей).

У зв'язку з цим під рецептивним полем розуміють множину всіх нейронів вхідного шару, пов'язаних з одним нейроном асоціативного шару. Зв'язки між нейронами асоціативного й вихідного шарів варіабельні й можуть модифікуватися шляхом зміни вагових коефіцієнтів.

Нейрони асоціативного шару мають лінійні активаційні функції, тому під час надходження із сітківки вхідних сигналів вони посилають імпульси на вихідний шар, де й відбувається додавання зважених імпульсів. У вихідному шарі використовуються уні- або біполярна активаційні функції. Якщо сума зважених імпульсів перевищує деяке задане порогове значення, виробляється одиничний вихідний сигнал, якщо не перевищує — нульовий (для уніполярної) або -1 (для біполярної функції активації). У зв'язку з цим перцептрон може розглядатися як двошарова ШНМ прямого поширення.

8.5.2. Навчання перцептрона

Існують різні шляхи реалізації процесу навчання перцептрона, однак у їхній основі лежить таке правило: вагові коефіцієнти перцептрона змінюються тільки тоді,

коли виникає розбіжність між його фактичною й бажаною реакціями.

Схему навчання перцептрона наведено на рис. 8.3.

Передбачається, що активаційна функція є пороговою. Процес навчання полягає в послідовному поданні множини пар, що навчають (\mathbf{x}_p, y_p^*) , $p = \overline{1, P}$, де \mathbf{x}_p — N -вимірний вхідний вектор, y_p^* — бажаний вихідний сигнал p -ї пари, що навчає, відповідно, за допомогою яких визначається необхідний вектор вагових коефіцієнтів \mathbf{w}^* такий, що

$$y_p = \text{sgn}(\mathbf{w}^{*T} \mathbf{x}_p) = y_p^*, \quad p = \overline{1, P}.$$

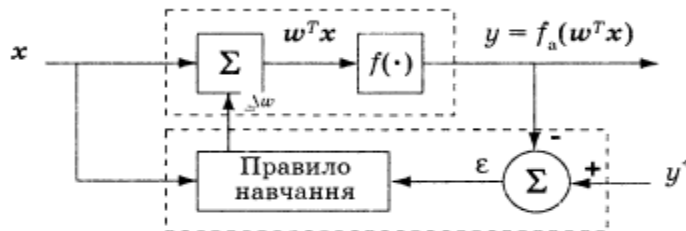


Рис. 8.3. Схема навчання перцептрона

У цьому випадку вектор \mathbf{w}^* забезпечить правильну класифікацію перцептроном усіх пар, що навчають, із поданої множини (завершується цикл навчання).

Гіперплощина $\mathbf{w}^{*T} \mathbf{x}_p = 0$ ділить вхідний простір на два підпростори. Для $y_p^* = 1$ має виконуватися умова $\mathbf{w}^{*T} \mathbf{x}_p > 0$, а для $y_p^* = -1$ — умова $\mathbf{w}^{*T} \mathbf{x}_p < 0$.

Алгоритм навчання може бути записаний у такий спосіб:

$$\mathbf{w}_{p+1} = \mathbf{w}_p + \gamma e_p \mathbf{x}_p,$$

де $e_p = y_p^* - y_p$ — помилка класифікації; γ — параметр, що впливає на швидкість збіжності алгоритму (тривалість процесу навчання). Корекція ваг відповідно може відбуватися в режимах online й offline.

У режимі online корекція відбувається при поданні кожної навчальної пари (\mathbf{x}_p, y_p^*) , $p = \overline{1, P}$.

У режимі offline у M -му циклі (епосі) навчання подаються всі пари (\mathbf{x}_p, y_p^*) й обчислюється середнє значення помилки класифікації

$$\bar{e}_m = \frac{1}{P} \sum_{p=1}^P (y_{p,m}^* - y_{p,m}),$$

що й використовується в алгоритмі навчання. Тут $y_{p,m}^*$, $y_{p,m}$ — бажані й реальні вихідні сигнали при пред'явленні p -ї пари, що навчає, в M -му циклі навчання відповідно.

Теорема збіжності для перцептрона, доведена Розенблаттом, стверджує, що *перцептрон може навчитися правильно класифікувати подані йому образи на скінченному числі навчальних пар*.

8.6. Алгоритм зворотного поширення помилки навчання багат шарових нейромереж прямого поширення

Алгоритм зворотного поширення помилки описаний у [11]. Він реалізує градієнтний метод мінімізації опуклого (звичайного квадратичного) функціонала

помилки в багатошарових мережах прямого поширення, що використовують моделі нейронів з диференціальними функціями активації. Застосування сигмоїдальних функцій активації, які мають відмінні від нуля похідні на всій області визначення, забезпечує правильне навчання й функціонування мережі. Процес навчання полягає у послідовному поданні мережі пар $(\mathbf{x}(i), \mathbf{y}^*(i))$ $i = \overline{1, P}$, що навчають, де $\mathbf{x}(i)$ і $\mathbf{y}^*(i)$ — вектор вхідних і бажаних вихідних сигналів мережі відповідно, вивчені реакції на них мережі й корекції відповідно до реакції вагових параметрів (елементів вагової матриці).

Перед початком навчання всім вагам присвоюють невеликі випадкові значення (якщо задати всі значення однакові, а для правильного функціонування мережі знадобляться нерівні значення, мережа не навчатиметься).

Для реалізації алгоритму зворотного поширення необхідно:

1. Вибрати із заданої навчальної множини чергову навчальну пару $(\mathbf{x}(i), \mathbf{y}^*(i))$, $i = \overline{1, P}$ та подати на вхід мережі вхідний сигнал $\mathbf{x}(i)$.
2. Обчислити реакцію мережі $\mathbf{y}(i)$.
3. Визначити помилку $\mathbf{y}^*(i) - \mathbf{y}(i)$.
4. Скорегувати ваги так, щоб помилка була мінімальною.
5. Кроки 1-4 повторити для всієї множини пар $(\mathbf{x}(i), \mathbf{y}^*(i))$ $i = \overline{1, P}$ доти, поки на заданій множині помилка не досягне необхідної величини.

Таким чином, у процесі навчання мережі подача вхідного сигналу й обчислення реакції відповідає прямому проходу сигналу від вхідного шару до вихідного, а обчислення помилки й корекція вихідних параметрів — зворотному, коли сигнал помилки поширюється по мережі від її виходу до входу. При зворотному проході здійснюється пошарова корекція ваг, починаючи з вихідного шару.

Алгоритм зворотного поширення застосовують також щодо мереж з будь-якою кількістю шарів: як мереж прямого поширення, так і таких, що містять зворотні зв'язки. Обмежимося розглядом випадку навчання двох шарів мережі прямого поширення. Фрагмент такої мережі зображено на рис. 8.4.

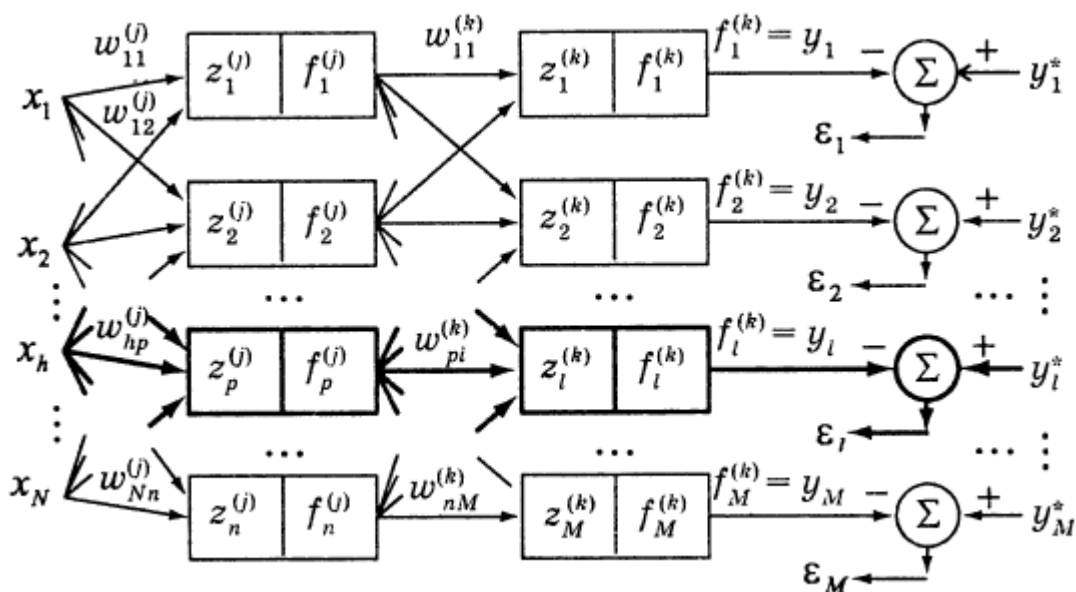


Рис. 8.4. Фрагмент мережі прямого поширення

На рисунку позначено: $w_{pl}^{(k)}$ — вага зв'язку p -го нейрона попереднього шару (j -го) з l -м нейроном наступного (k -го) шару; $f_p^{(j)}$ — активаційна функція p -го нейрона j -го шару; $z_p^{(j)}$ — зважена сума вихідних сигналів попереднього (i -го) шару, що надходять на вхід p -го нейрона наступного (j -го) шару.

Обчислення ваг нейронів вихідного шару

Розглянемо l -й нейрон вихідного (k -го) шару. Вихід даного нейрона є виходом мережі, тому його сигнал $y_l = f_l^{(k)}$ порівнюється з необхідним y_l^* й обчислюється помилка

$$\zeta_l = y_l^* - f_l^{(k)}.$$

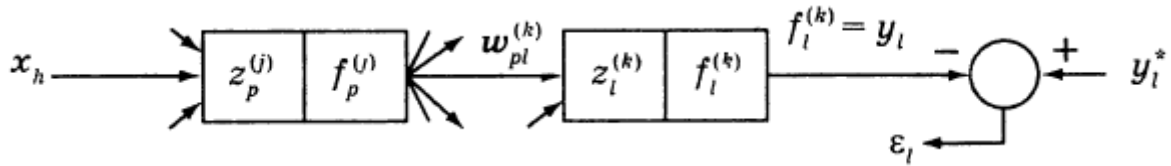


Рис. 8.5. Фрагмент багатошарової мережі

Використання квадратичного критерію якості навчання

$$\zeta_l^2 = (y_l^* - f_l^{(k)})^2 \quad (8.1)$$

дозволяє отримати градієнтний алгоритм корекції ваг, що у цьому випадку набуває вигляду

$$\Delta w_{pl}^{(k)} = -\gamma_{pl} \frac{\partial \zeta_l^2}{\partial w_{pl}^{(k)}},$$

де γ_{pl} — коефіцієнт, що впливає на швидкість навчання. Обчислення складної частинної похідної здійснюється за правилом

$$\frac{\partial \zeta_l^2}{\partial w_{pl}^{(k)}} = \frac{\partial \zeta_l^2}{\partial f_l^{(k)}} \frac{\partial f_l^{(k)}}{\partial z_l^{(k)}} \frac{\partial z_l^{(k)}}{\partial w_{pl}^{(k)}}. \quad (8.2)$$

З урахуванням (8.1), виду активаційних функцій і того, що $z_l^{(k)} = \sum_{p=1}^n w_{pl}^{(k)} f_p^{(j)}$, отримуємо

$$\frac{\partial \zeta_l^2}{\partial f_l^{(k)}} = -2(y_l^* - y_l) = -2\zeta_l; \quad (8.3)$$

$$\frac{\partial f_l^{(k)}}{\partial z_l^{(k)}} = \alpha f_l^{(k)} (1 - f_l^{(k)}); \quad (8.4)$$

$$\frac{\partial z_l^{(k)}}{\partial w_{pl}^{(k)}} = f_p^{(j)}. \quad (8.5)$$

Підстановка (8.3)-(8.5) у (8.2) дає

$$\Delta w_{pl}^{(k)} = 2\alpha\gamma_{pl}\zeta_l f_l^{(k)}(1 - f_l^{(k)})f_p^{(k)} \quad (8.6)$$

або

$$w_{pl}^{(k)}(i+1) = w_{pl}^{(k)}(i) + \gamma_{pl}\delta_{pl}^{(k)}f_p^{(k)}, \quad (8.7)$$

де $\delta_{pl}^{(k)} = 2\alpha\zeta_l f_l^{(k)}(1 - f_l^{(k)})$, i — номер ітерації.

Аналогічно обчислюються ваги інших нейронів вихідного шару. Всі величини, що входять у (8.7), є відомими. Присутня в (8.6), (8.7) помилка ζ_l відмінна від нуля внаслідок того, що нейрони вихідного шару виробляють помилкові вихідні сигнали, тому що, по-перше, їм присвоєнні випадкові вагові коефіцієнти й, по-друге, нейрони прихованих шарів також виробляють помилкові сигнали. Помилка ζ_l поширюється назад на попередні приховані шари й використовується для корекції ваг цих шарів.

Обчислення ваг нейронів прихованого шару

Фрагмент мережі для обчислення ваг нейронів прихованого шару наведено на рис. 8.6.

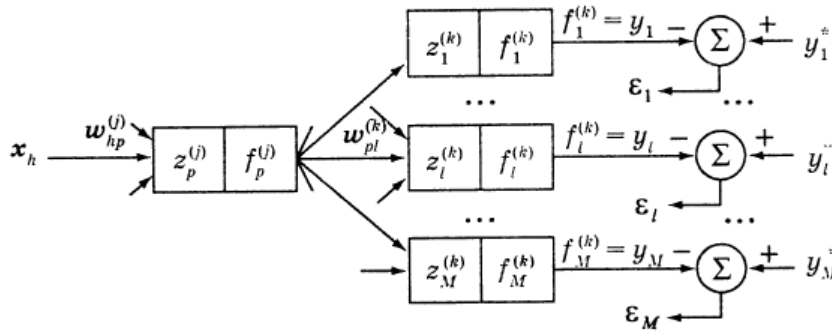


Рис. 8.6. Фрагмент мережі

Розглянемо p -й нейрон прихованого (j -го) шару. Оскільки вихідний сигнал цього нейрона надходить на виходи всіх нейронів вхідного шару, алгоритм настроювання записується в такий спосіб:

$$\Delta w_{hp}^{(j)} = -\gamma_{hp} \frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} = -\gamma_{hp} \sum_{l=1}^M \frac{\partial \zeta_l^{(2)}}{\partial w_{hp}^{(j)}}. \quad (8.8)$$

Тут ζ_l — помилка на i -му виході; γ_{hp} — параметр, що виконує ту саму роль, що й γ_{pl} у (8.7).

Оскільки квадратичний критерій якості в цьому випадку приймає вигляд

$$\zeta = \sum_{l=1}^M [y_l^* - f_l^{(k)}]^2,$$

частинна похідна в (8.8), обчислюється так:

$$\frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} = \sum_{l=1}^M \frac{\partial \zeta_l^{(2)}}{\partial f_l^{(k)}} \frac{\partial f_l^{(k)}}{\partial z_l^{(k)}} \frac{\partial z_l^{(k)}}{\partial f_p^{(j)}} \frac{\partial f_p^{(j)}}{\partial z_p^{(j)}} \frac{\partial z_p^{(j)}}{\partial w_{hp}^{(j)}}.$$

Для розглянутого випадку за аналогією з вищевикладеним отримуємо

$$\frac{\partial \zeta_l^2}{\partial f_l^{(k)}} = -2 \left[y_l^* - f_l^{(k)} \right] = -2 \zeta_l;$$

$$\frac{\partial f_i^{(k)}}{\partial z_i^{(k)}} = \alpha f_i^{(k)} (1 - f_i^{(k)});$$

$$\frac{\partial z_l^{(k)}}{\partial f_p^j} = w_{pl}^{(k)};$$

$$\frac{\partial f_p^{(j)}}{\partial z_p^{(j)}} = \alpha f_p^{(j)} (1 - f_p^{(j)});$$

$$\frac{\partial z_p^{(j)}}{\partial w_{hp}^{(j)}} = x_h.$$

Тут враховано, що $z_l^{(k)} = \sum_{p=1}^m w_{pl}^{(k)} f_p^{(j)}$; $z_l^{(j)} = \sum_{h=1}^N w_{hp}^{(j)} x_h$, а функції активації є сигмоїдальними. Таким чином,

$$\begin{aligned} \frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} &= \sum_{i=1}^m (-2) \alpha \zeta_l \left[f_l^{(k)} (1 - f_l^{(k)}) \right] w_{pl}^{(k)} \alpha \left[f_p^{(j)} (1 - f_p^{(j)}) \right] x_h = \\ &= - \sum_{l=1}^M \delta_{pl}^{(k)} w_{pl}^{(k)} \frac{\partial f_p^{(j)}}{\partial z_p^{(j)}} x_h. \end{aligned}$$

Позначимо $\delta_{hp}^{(j)} = \delta_{pl}^{(k)} w_{pl}^{(k)} \frac{\partial f_p^{(j)}}{\partial z_p^{(j)}}$.

Тоді

$$\frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} = - \sum_{l=1}^M \delta_{hp}^{(j)} x_h$$

і алгоритм корекції ваг приймає вигляд

$$\Delta w_{hp}^{(j)} = -\gamma_{hp} x_h \sum_{l=1}^M \delta_{hp}^{(j)},$$

або

$$w_{hp}^{(j)}(i+1) = \Delta w_{hp}^{(j)}(i) + \gamma_{hp} x_h \sum_{l=1}^M \delta_{hp}^{(j)}$$

Слід зазначити, що використання випадкових початкових значень вагових параметрів призводить до того, що при повторному моделюванні з іншими початковими значеннями форма поверхонь може бути іншою.

Опис бібліотек, у яких реалізовано різні модифікації методів навчання штучних нейромереж, наведено у [12, 13].