

Н.О. Матвеева, І.В. Гаркуша

## АЛГОРИТМИ ОПТИМІЗАЦІЇ НАВЧАННЯ ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ ДЕФЕКТІВ

*Анотація. Представлені результати дослідження якості навчання (залежність суми квадратів помилок від кількості епох) для розв'язання задачі класифікації дефектів. Моделювання багатосарового перцептрона виконувалось в середовищі MATLAB.*

*Ключові слова: нейронні мережі, багатосаровий перцептрон, алгоритми оптимізації.*

**Вступ.** Проблема класифікації є однією з найбільш часто виникаючих і розв'язуваних задач як при наукових дослідженнях, так і на практиці. Розв'язання задачі класифікації полягає в розробці й формуванні таких правил і закономірностей, які дозволили б розпізнати певні явища або об'єкти та обумовити їх належність до деяких класів або груп. Задача значно ускладнюється в реальних умовах: доводиться враховувати велику кількість різних факторів і явищ [1,2].

При проведенні неруйнівного контролю композитних матеріалів слід брати до уваги їх складний рельєф поверхні. Технологія виготовлення волокнистих композитів звичайно не передбачає механічну обробку, що ускладнює процес сканування поверхні та додає різні види шумів. Виникає задача – аналізуючи оброблювані сигнали, необхідно отримати інформацію про наявність та розміри дефектів. Для розв'язання таких задач використовуються нейронні мережі [3, 4], які активно розвиваються останнім часом, володіють універсальними та адаптивними властивостями й забезпечують високу ефективність розпізнавання. Але, у свою чергу, характеризуються такими недоліками, як тривалість і складність процесу навчання, складність підбору параметрів нейронної мережі для впевненого розв'язання задачі.

**Метою роботи** є порівняльний аналіз використання градієнтних та сполучених алгоритмів навчання при вирішенні задачі класифікації сигналів дефектоскопії.

**Основна частина.** Кожна штучна нейронна мережа являє собою множиную простих елементів – нейронів, які сполучені певним чином. Задача нейронної мережі полягає у перетворенні інформації. Конкретний вигляд виконуваної мережею перетворення даних обумовлюється не тільки характеристиками нейронів, які входять до її структури, але і особливостями її архітектури, а самотопологією міжнейронних зв'язків, напрямом і способами передачі інформації між нейронами, а також засобами навчання мережі.

В роботі запропоновано використовувати багатошаровий перцептрон (multilayer perceptron - MLP) засобами бібліотеки Neural Networks Tool середовища Matlab.

Алгоритми оптимізації навчання є стратегіями, заснованими на реалізації ідеї ітеративного спуску, які забезпечують мінімізацію функціонала навчання. У процесі роботи алгоритмів, як правило, виникає задача одновимірного пошуку мінімуму уздовж заданого напрямку. Це можуть бути антиградієнти або сполучені напрямки [4].

Градiєнтні алгоритми навчання є специфічною реалізацією градієнтного спуску в просторі вагових коефіцієнтів і зміщень MLP і забезпечують рух по поверхні функціоналу помилки в напрямку, протилежному вектору градієнта. До них відносяться алгоритми: градієнтного спуску з параметром швидкості навчання (налаштування) – GD (М-функція *traingd*); градієнтного спуску з адаптацією параметра швидкості навчання – GDA (*traingda*). Якщо до алгоритмів GD і GDA додати ще один параметр – збурення, то отримаємо ще два алгоритми – GDM (*traingdm*) і GDX (*traingdx*), відповідно [4]. Формально всі алгоритми градієнтного спуску можна записати в наступному вигляді:

$$w_{k+1} = w_k - \alpha_k g_k \quad (1)$$

де  $w_k$  – вектор вагових коефіцієнтів;  $\alpha_k$  – параметр швидкості навчання;  $g_k$  – вектор градієнта функціонала помилки. Переходячи від  $k$ -ї ітерації до  $k+1$ -ї, алгоритм виконує корекцію вагових коефіцієнтів:

$$\Delta w_k = w_{k+1} - w_k = -\alpha_k g_k. \quad (2)$$

З виразу (2) витікає, що метод градієнтного спуску працює на основі лінійної апроксимації функції вартості біля поточної точки  $w_k$ , при якій єдиним джерелом інформації щодо поверхні помилок є градієнт  $g_k$ . Таке обмеження забезпечує значну перевагу – простоту реалізації, але отримуємо низьку швидкість збіжності. Крім того, на

швидкість збіжності впливає значення параметра  $\alpha$ , вибір якого викликає певні труднощі. В алгоритмі GD значення параметра  $\alpha$  задається тільки на початку навчання, залежить від конкретної задачі і отримується дослідним шляхом. Алгоритм GDA використовує евристичну стратегію зміни параметра  $\alpha$  у процесі навчання. На кожному циклі навчання обчислюються значення параметрів, що настроюються, і нові значення виходів і похибок. Нові значення порівнюються із значеннями, отриманими на попередньому кроці. Якщо нова похибка менша за попередню, то параметр швидкості настройки збільшується, і навпаки. Алгоритм GDM модифікує (2) з урахуванням параметра збурення  $mc$  і реалізує наступне співвідношення для збільшення вектора параметрів, що настроюються

$$\Delta w_k = mc\Delta w_{k-1} + (1 - mc)\alpha g_k. \quad (3)$$

Таким чином, алгоритми градієнтного спуску коригують параметри, які настроюються, в напрямку антиградієнту, однак такий напрямок далеко не завжди є самим сприятливим (за можливо мале число кроків забезпечити збіжність до мінімуму функціонала якості).

Сполучені напрямки і, відповідно, метод оптимізації – *метод сполучених градієнтів* – дозволяють визначити необхідний мінімум набагато швидше. Розглянемо наступні алгоритми методу спряжених градієнтів: Флетчера-Рівса CGF (М-функція - *traincgf*); Полака-Рібейри CGP (*traincgp*); Пауелла-Біель CGB (*traincgb*); Моллера SCG (*trainscg*). Всі алгоритми методу сполучених градієнтів на першій ітерації починають пошук у напрямку антиградієнту:

$$p_0 = -g_0. \quad (4)$$

Для визначення розміру кроку уздовж сполученого напрямку виконуються спеціальні одномірні процедури пошуку мінімуму. Коли вибрано напрямок спуску, потрібно визначити оптимальну відстань (крок пошуку), на величину якого слід змінити настроюванні параметри:

$$w_{k+1} = w_k + \alpha_k p_k. \quad (5)$$

Потім визначається наступний напрямок пошуку як лінійна комбінація нового напрямку найшвидшого спуску і вектора руху:

$$p_k = -g_k + \beta_k p_{k-1}, \quad (6)$$

Різні алгоритми методу сполученого градієнта розрізняються способом обчислення константи  $\beta_k$ . Алгоритм CGF використовує такий вираз:

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}. \quad (7)$$

Алгоритм Полака-Рібейри CGP обчислює константу  $\beta_k$ :

$$\beta_k = \frac{\Delta \mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}. \quad (8)$$

Для всіх алгоритмів методу сполучених градієнтів напрямком пошуку періодично перевстановлюється заново на напрямок антиградієнту (рестарт). Це відбувається, коли виникають проблеми зі збіжністю. Одна з таких стратегій рестарту реалізована в алгоритмі CGB. Відповідно до цієї стратегії рестарт виконується, якщо поточний та попередній напрямки градієнтів слабкоортогональні, і ця умова визначається наступним чином:

$$\|\mathbf{g}_k^T \mathbf{g}_k\| \geq 0.2 \|\mathbf{g}_k\|^2. \quad (9)$$

Всі алгоритми, засновані на методі сполучених градієнтів, реалізують на кожній ітерації процедуру одновимірного пошуку. Ця коштотна в обчислювальному відношенні процедура вимагає на кожній ітерації кілька разів обчислювати реакцію мережі.

**Експериментальні дослідження.** Для моделювання процесу класифікації сигналів використовувалось обчислювальне середовище MATLAB R2010b. У якості навчальної множини для нейронної мережі запропоновані значення функцій, які відповідають модельним сигналам, одержаним при скануванні поверхні композитів [5], у точках  $x = -2, -1.9, \dots, 2$ :

$$y(x) = \exp(-1,5x^2) - k \cdot \exp(-3x^2) \quad (10)$$

де  $k$  змінюється від 0 до 1. Вираз (10) при зміні значень  $k$  описує різні форми сигналів дефектоскопії: *вузький унімодальний сигнал*, який характеризує довгі тріщини, довжина яких перебільшує зону контролю; *положистий унімодальний сигнал* – для тріщин меншої розмірності; *бімодальний сигнал* мають маленькі тріщини (при  $k = 1$  – точковий дефект) [6].

Спочатку будемо використовувати градієнтні методи: *traingd*, *traingdm* і *traingda*, *traingdx*. Для вирішення поставленого завдання сформована чотирьохшарова мережа зворотного поширення, яка включає 41 нейрон у вхідному шарі (за кількістю компонент вхідного

вектора) з передатною функцією *logsig*, 20 нейронів у другому шарі з гіперболічним тангенсом *tansig*, 10 нейронів у третьому шарі з логістичною сигмоїдальною функцією *logsig* і 3 нейрони у вихідному шарі (за кількістю компонент вихідного вектора) з лінійною функцією *purelin* (рис. 1). Для обчислення функціоналу якості навчання, який залежить від помилок мережі  $e$ , в роботі використовується сума квадратів помилок:

$$SSE = \sum_{i=1}^N e_i^2.$$

Спочатку проводилось навчання мережі на модельних сигналах до досягнення  $SSE=0.01$ . Потім здійснювалось навчання при наявності шуму, припустиму похибку збільшували до 0.06.

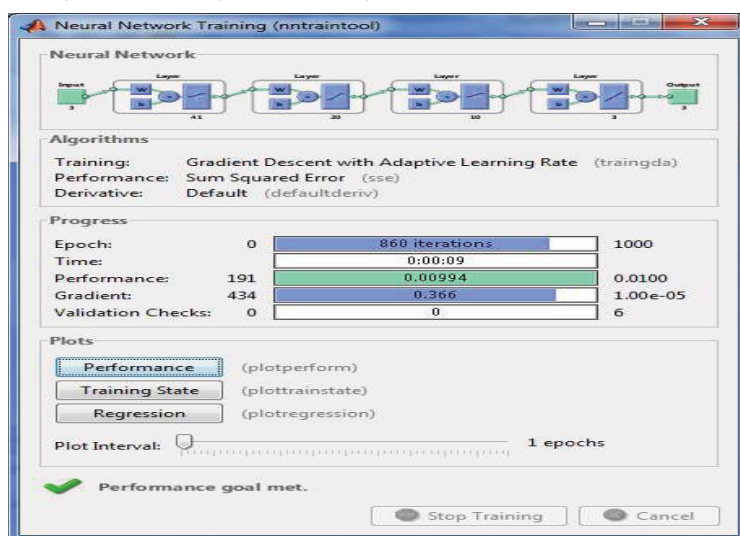


Рисунок 1 - Моделювання мережі в Neural Networks Toolbox

Далі в якості алгоритмів навчання використовувались методи сполучених градієнтів: *traincgf*, *trainscg*, *traincgp*, *traincgb* (рис. 2). Знов проводили навчання мережі на модельних сигналах до досягнення допустимої суми квадратів помилок, яка дорівнює 0.01. Потім проводилось навчання при наявності шуму із середніми значеннями 0.05 та 0.1. При навчанні з шумом припустиму похибку залишили  $SSE=0.01$ .

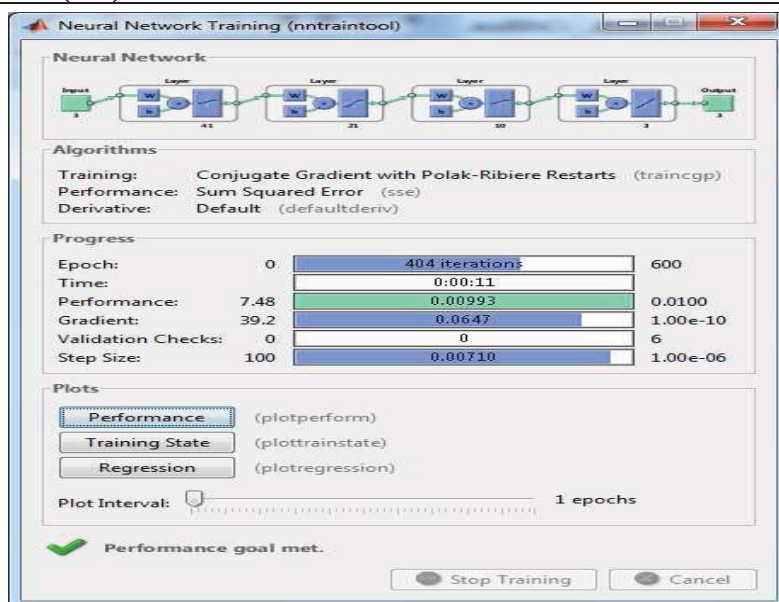


Рисунок 2 - Моделювання мережі в Neural Networks Toolbox

**Порівняльний аналіз результатів експерименту.** Для отримання усереднених параметрів було виконано понад 50 експериментів моделювання для кожного з методів оптимізації з різними параметрами (табл. 1).

Таблиця 1

Показники функцій оцінки якості навчання

Функції активації	Кількість епох (циклів) без шуму	Кількість епох (циклів) з шумом
GD	5000 ÷ 6000	3500 ÷ 5400
GDM	1900 ÷ 3500	1500 ÷ 3000
GDA	840 ÷ 1900	300 ÷ 1200
GDX	400 ÷ 500	200 ÷ 4300
Rprop	30 ÷ 50	50 ÷ 200
CGF	60 ÷ 65	100 ÷ 240
SCG	60 ÷ 85	100 ÷ 400
CGP	45 ÷ 70	400 ÷ 550
CGB	50 ÷ 65	170 ÷ 500

У навчальних алгоритмах градієнтного спуску управління збіжністю здійснюється за допомогою параметра швидкості настроювання, а в алгоритмах методу сполучених градієнтів розмір кроку коригується на кожній ітерації і тому ці алгоритми значно перебільшують за швидкістю навчання градієнтні алгоритми (табл. 1), однак для їх реалізації необхідний значний обсяг пам'яті. Порівняльний аналіз

алгоритмів GD, GDM і GDA, GDX для задачі класифікації сигналів показав, що швидкість збіжності і значення функцій помилок краще всього для алгоритму GDX (він містить в собі параметри адаптації та збурення). Але найкращим з методів градієнтних алгоритмів є Rprop, котрий має найвищу швидкість навчання.

Одним з найбільш інформативних способів перевірки якості алгоритмів є побудова кривих навчання. Крива навчання є графіком зміни суми квадратів помилок (SSE) залежно від кількості ітерацій (рис. 3 та рис. 4).

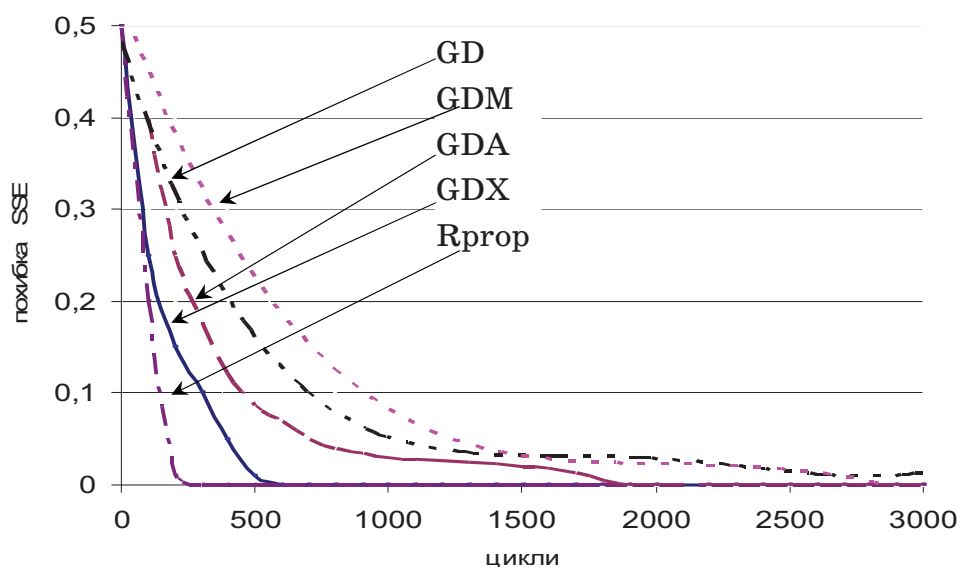


Рисунок 3 - Усереднені криві навчання для градієнтних методів

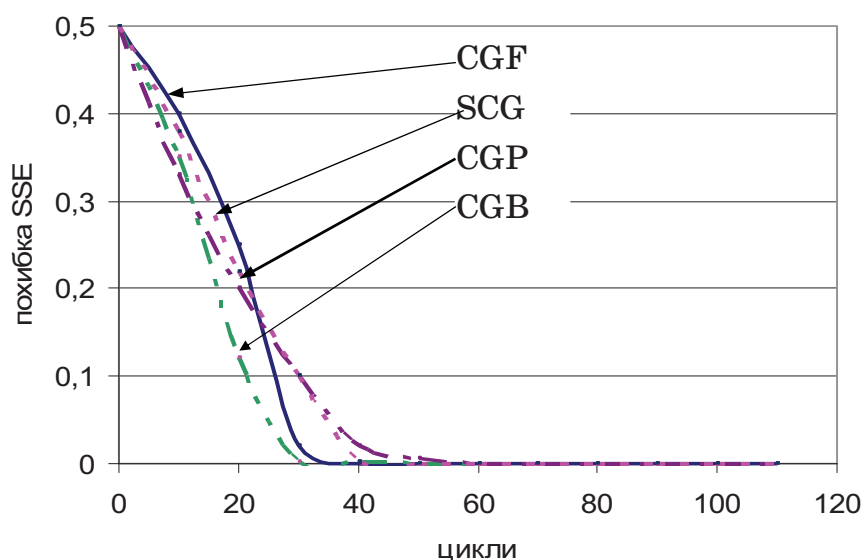


Рисунок 4 - Усереднені криві навчання для методів сполучених градієнтів

Криві навчання (рис. 3) починаються з досить великих значень помилок, обумовлених початковими умовами, і потім їх значення зменшуються з деякою швидкістю, залежно від типу алгоритму, і сходяться у межах деякого сталого граничного значення помилки, яка мало відрізняється для різних алгоритмів. Це значить, що поверхня помилок у досліджуваній задачі досить гладка.

Всі досліджувані алгоритми метода сполучених градієнтів показали практично однаково високу швидкість навчання (рис. 2). Якщо для порівняння алгоритмів використовувати не тільки необхідну кількість циклів навчання, але і кількість додаткових параметрів, то вимоги до розміру оперативної пам'яті для алгоритмів CGP та CGB трохи більші, оскільки треба запам'ятати 4 та 6 векторів, відповідно, в той же час для алгоритму CGF – тільки 3. Вимоги до пам'яті для алгоритму SCG приблизно такі, як і для метода CGF.

**Висновки.** В роботі проведені дослідження алгоритмів оптимізації навчання при розв'язанні задачі класифікації сигналів, які отримуються при проведенні неруйнівного контролю композитів. Найкращі результати показали методи сполучених градієнтів. Алгоритм Rprop показав практично таку ж швидкість навчання, однак він не потребує використання процедури одномірного пошуку, характеризується високою швидкістю й немає вимог до об'єму пам'яті.

## ЛІТЕРАТУРА

1. Горелик А.Л., Скрипкин В.А. Методы распознавания. -М.: Высшая школа, 1989.-232 с
2. Аксенов С.В. Организация и использование нейронных сетей (методы и технологии) / С.В. Аксенов, В.Б. Новосельцев – Томск: Изд-во НТЛ, 2006. – 128 с.
3. Хайкин Саймон. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006
4. Медведев В.С. Нейронные сети. MATLAB 6 / В.С. Медведев, В.Г. Потемкин – М.: ДИАЛОГ-МИФИ, 2002. – 496 с
5. Хандецкий В.С. Спектральная идентификация сигналов в дефектоскопии композитов с использованием теории статистических испытаний / Хандецкий В.С., Герасимов В.В. //Вісник ДНУ: Фізика. Радіоелектроніка. – Дніпропетровськ: – 2003. № 10. – С. 128 – 132
6. Матвеева Н.А. Моделирование нейросети для решения задачи классификации в дефектоскопии // Системні технології. Регіон. міжвуз. зб. наук. праць. - Дніпропетровськ: ДНВП «Системні технології», 2011. -Вип. 1(72). - С. 37-44.