

ДОРОГИЙ Я.Ю.,
ЦУРКАН В.В.,
ДОРОГА-ІВАНЮК О.О.,
ФЕРЕНС Д.А.

ЗАСТОСУВАННЯ АЛГОРИТМУ СТРУКТУРНОЇ ОПТИМІЗАЦІЇ НЕЙРОННОЇ МЕРЕЖІ В ЗАДАЧАХ КЛАСИФІКАЦІЇ ДАНИХ

В статті розглянуте питання архітектурної оптимізації нейронної мережі та її застосування для задач класифікації даних.

In the article the question of neural network architecture optimization and its application for classification tasks were considered.

Введення в проблему

На сьогоднішній день апарат нейронних мереж широко використовується для розв'язання різноманітних задач. В зв'язку з цим, питання розробки алгоритму навчання, що мав би змогу динамічно оптимізувати структуру нейронної мережі, є дуже актуальним. Наявність такого методу дала б можливість досліднику швидше отримати структуру нейронної мережі, яка б найкраще відповідала предметній області та наявним вхідним даним.

Аналіз існуючих рішень

Для побудови оптимальної структури нейронної мережі використовується досить широке коло алгоритмів. Першим з таких алгоритмів є алгоритм черепичної побудови [1]. Ідея алгоритму полягає в додаванні нових шарів нейронів таким чином, щоб вхідні навчальні вектори, які мають різні відповідні вихідні значення, мали б в ньому різне внутрішнє представлення. Ще одним яскравим представником є алгоритм швидкої надбудови [2]. Нові нейрони за цим алгоритмом додаються між вихідними шарами. Роль цих нейронів – корегування помилки вихідних нейронів. В загальному вигляді нейронна мережа, що побудована за таким алгоритмом, має форму бінарного дерева. Широко відомими є алгоритми Monoplan, NetLines та NetSphere [3], метод редукції [4]. Але всі ці алгоритми мають достатньо широкий перелік недоліків, тому було вирішено розробити свій алгоритм. В роботі [6] запропоновано алгоритм, застосування якого для задач класифікації і було перевірено в цій роботі.

Мета роботи

Метою даної роботи є аналіз алгоритму структурної оптимізації нейронної мережі в ході її навчання для задач розпізнавання образів.

Алгоритм структурної оптимізації при навчанні

Алгоритм структурного навчання використовується на багатошарових мережах прямого поширення та має ітеративний характер: на кожній ітерації виконується пошук структури мережі кращої за попередню. Пошук мережі виконується шляхом перебору усіх можливих мутацій мережі, вибору та комбінації кращих (селекція та схрещення).

Розглянемо основні параметри алгоритму.

Параметри навчання:

- швидкість навчання, η ;
- коефіцієнт інерції, μ ;
- коефіцієнт затухання ваг, ϵ ;
- вірогідність активації нейрону прихованого шару, p_h ;
- вірогідність активації нейрону вхідного шару, p_i .

Параметри структурного навчання:

- початкова кількість нейронів прихованого шару;
- функція активації прихованого шару;
- функція активації та функція ціни вхідного шару;
- максимальна кількість мутацій при схрещенні;
- кількість епох навчання початкової мережі;

- кількість епох навчання у ітерації;
- види допустимих мутацій;
- частина навчальної вибірки, що використовується для навчання.

Елементарні структурні операції над нейронною мережею

Згідно з [5] введено наступні елементарні структурні операції над мережею:

- додавання синапсу між двома випадково вибраними незв'язаними вузлами або нейронами мережі – операція Syn_{ADD} ;
- видалення синапсу між двома випадково вибраними незв'язаними вузлами або нейронами мережі – операція Syn_{DEL} ;
- переміщення синапсу між двома випадково вибраними незв'язаними вузлами або нейронами мережі – операція Syn_{MOD} ;
- зміна функції активації нейрона для випадково вибраного нейрона – операція A_{MOD} ;
- серіалізація вузла або нейрона – операції – Ser_{NODE} і Ser_{NR} ;
- паралелізація вузла або нейрона – операції Par_{NODE} і Par_{NR} ;
- додавання вузла або нейрона – операції Add_{NODE} і Add_{NR} ;
- створення нового шару – операція L_{ADD} ;
- видалення шару НМ – операція L_{DEL} .

Використання чи невикористання наведених структурних операцій залежить від складності поставленої задачі.

Для задач розпізнавання, що будуть описані в цій статті, використані наступні операції (мутації):

- пуста (змін в структурі не відбувається) (рис. 1);

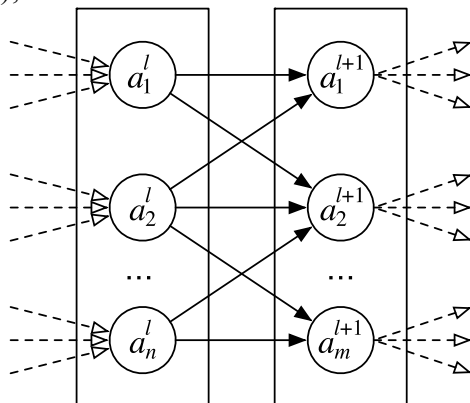


Рис.1. Пуста мутація

- видалення синапсу між випадково вибраними нейронами (рис. 2);

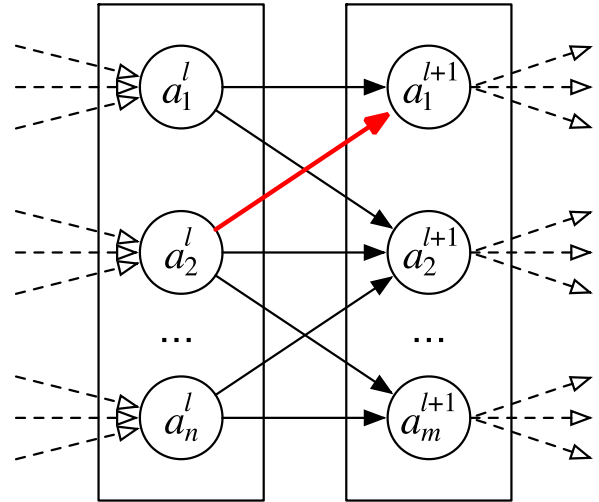


Рис.2. Видалення синапсу

- додавання синапсу між випадково вибраними нейронами (рис. 3);

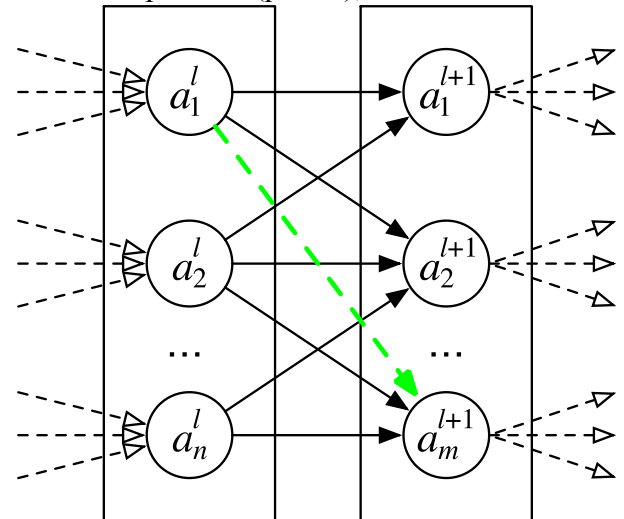


Рис.3. Додавання синапсу

- видалення випадково вибраного нейрону прихованого шару та усіх його синапсів (рис. 4);

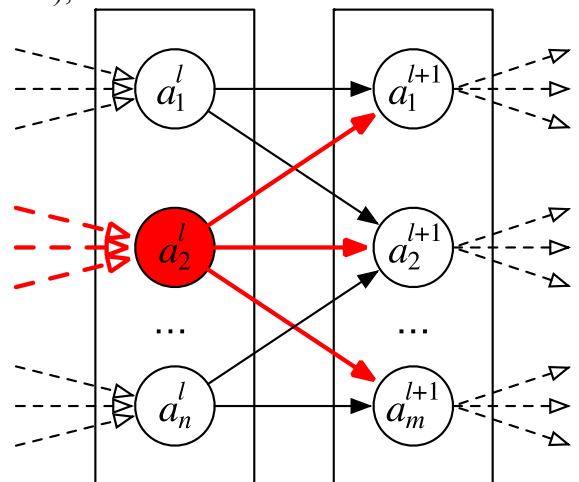


Рис.4. Видалення нейрону

- додавання нового нейрону у вихідному шарі (рис. 5);

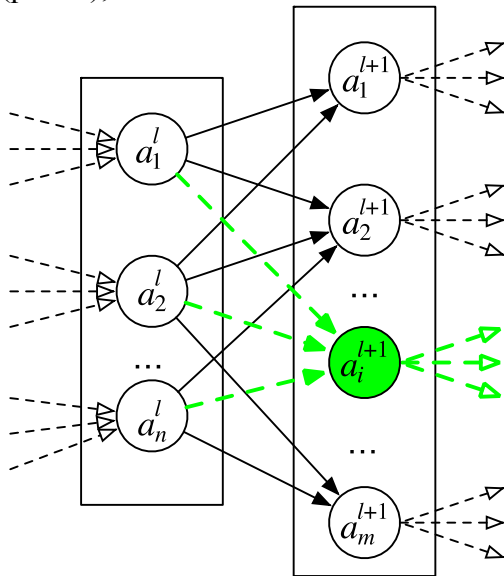


Рис.5. Додавання нейрону

- додавання нового прихованого шару між випадково вибраними сусідніми шарами (рис. 6).

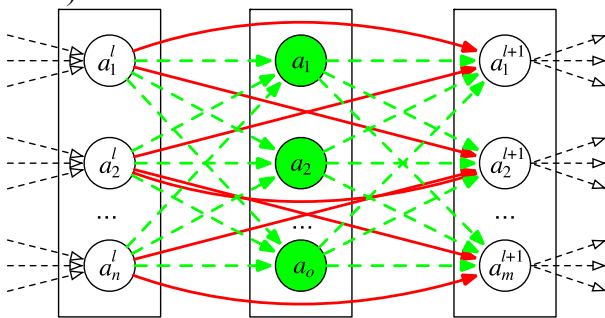


Рис.6. Додавання нового прихованого шару

Під час додавання нового прихованого шару руйнуються всі поточні зв'язки між сусідніми шарами.

Розглянемо алгоритм структурної оптимізації при навчанні. В якості базового алгоритму навчання візьмемо алгоритм зворотного поширення помилки, запропонований в [5, 6].

На початку роботи алгоритму створюється мережа із одним прихованим шаром, заданою кількістю нейронів та функцією активації. Оскільки початкові значення синаптичних ваг вибираються довільним чином, початкова мережа проходить задану кількість навчальних епох.

У кожній ітерації алгоритму виконується пошук усіх можливих мутацій поточної мережі: видалення кожного синапсу, видалення кожного нейрону усіх прихованих шарів і т. д. Далі,

мережі, отримані внаслідок виконання кожної мутації, навчаються протягом заданої кількості епох незалежно одна від іншої.

Після навчання мереж усіх можливих мутацій виконується вибір та схрещення кращих з них для отримання комбінованих мутацій. Як критерій для порівняння мереж використовується значення ціни на тренувальній вибірці. Для створення комбінованих мутацій використовуються лише ті мережі, значення цін яких є меншими за ціну пустої мутації.

Після отримання комбінованих мутацій вони порівнюються та вибирається найкраща. Ця мережа і є результатом роботи ітерації.

Алгоритм припиняє роботу якщо отримана мережа є гіршою (має більше значення ціни) за мережу в попередній ітерації.

Якщо мережа є кращою, відбувається перехід до наступної ітерації.

Експериментальні дослідження

Задача MONK's. Задачі MONK's [7] були одними із перших що були використані для порівняння алгоритмів класифікації. Кожен навчальний приклад вибірки містить 7 атрибутів, де останній атрибут – номер класу до якого необхідно віднести приклад:

1. $a_1 \in \{1, 2, 3\}$
2. $a_2 \in \{1, 2, 3\}$
3. $a_3 \in \{1, 2\}$
4. $a_4 \in \{1, 2, 3\}$
5. $a_5 \in \{1, 2, 3, 4\}$
6. $a_6 \in \{1, 2\}$
7. $a_7 \in \{0, 1\}$

Задачі визначаються наступним чином:

- Проблема M_1 : $(a_1 = a_2) \vee (a_5 = 1)$
- Проблема M_2 :
хоча б 2 з $\{a_1 = 1, a_2 = 1, a_3 = 1, a_4 = 1, a_5 = 1, a_6 = 1\}$
- Проблема M_3 :
 $((a_5 = 3) \wedge (a_4 = 1)) \vee ((a_5 \neq 4) \wedge (a_2 \neq 3))$

Нейронні мережі легко вирішують проблеми M_1 та M_2 та досягають 100% точності класифікації на тестовій вибірці. Навчальна вибірка для проблеми M_3 містить шум у вигляді 5% неправильно класифікованих прикладів, саме ця проблема буде використана для дослідження.

Були використані наступні значення параметрів навчання та параметри структурної оптимізації:

- швидкість навчання: $\eta = 0.001$
- коефіцієнт інерції: $\mu = 0$;
- коефіцієнт затухання ваг: $\varepsilon = 0.5$;
- максимальна кількість мутацій при схрещенні: $M = 10$;
- кількість епох навчання початкової мережі: $T_0 = 100$;
- кількість епох навчання у ітерації: $T_i = 20$;
- види допустимих мутацій: додавання та видалення ваг;

Отримані залежності значення ціни та точності класифікації наведені на рис. 7 та 8.

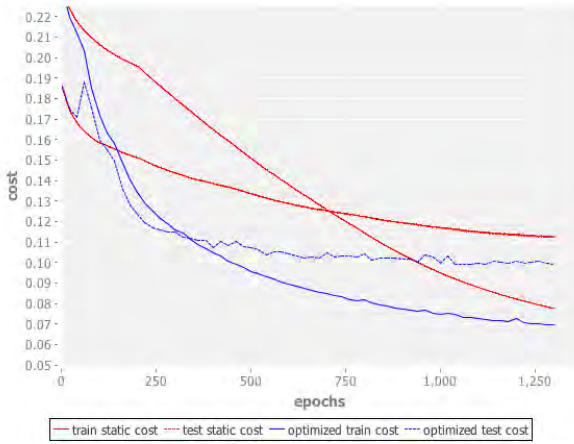


Рис.7. Значення ціни звичайної та оптимізованої мереж

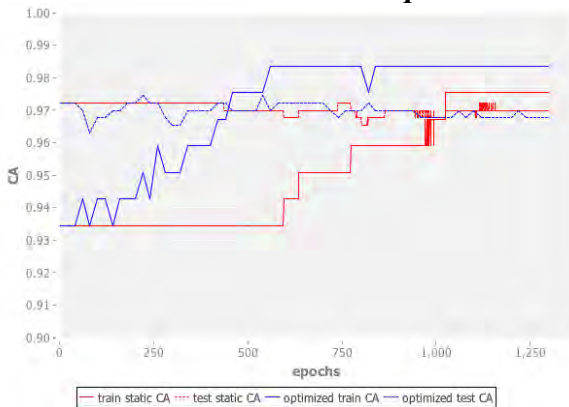


Рис.8. Точність класифікації звичайної та оптимізованої мереж

Отримана точність класифікації наведена в табл. 1.

Табл.1. Результуюча точність класифікації для MONK's проблеми

HM	Навчання, %	Тестування, %
Звичайна	97.54	96.99
Оптимізована	98.36	96.75

Хоча значного збільшення точності класифікації не відбулося, з наведених залежностей можна зробити висновок, що в результаті оптимізації структури під час навчання мережа не зупинялась у точках локальних мінімумів і навчалась удвічі швидше.

Задача TwoSpirals. Ця проблема є досить складною задачею класифікації та узагальненням для багатьох алгоритмів розпізнавання, була запропонована у [8]. Вибірка складається із набору точок що утворюють дві двовимірні спіралі. Необхідно правильно класифікувати точки, що не входять до навчальної вибірки.

Формування вибірки. Кожен кортеж тренувальної вибірки складається із трьох елементів: координати x та y у діапазоні $0..1$ та номер кривої, якій він відповідає. У графічному вигляді вибірка зображена на рис. 9.

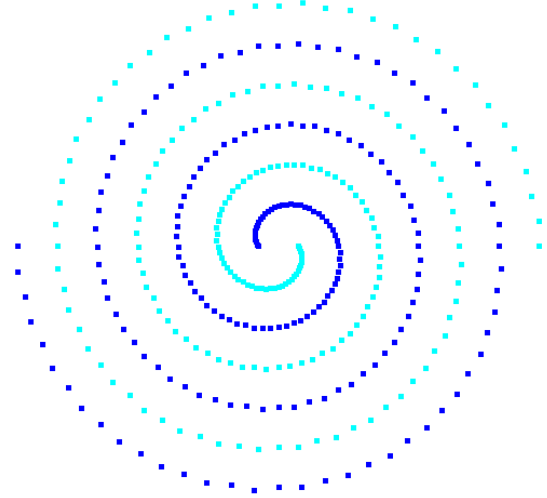


Рис.9. Графічне представлення вибірки TwoSpirals

Архітектура мережі. Для вирішення задачі в якості початкової мережі була вибрана 2-шарова мережа з одним прихованим шаром, що містить 10 нейронів з випрямленою лінійною функцією активації.

Із цією задачею найкраще справляються мережі типу 2-10-10-2, 2-5-10-2, що використовують непарні функції активації (біполярну сигмоїду або гіперболічний тангенс). Натомість, цікавим було дослідити можливість вирішення задачі лише шляхом оптимізації структури, використовуючи більш уживані топології мереж.

Дослідження роботи алгоритму. Були використані наступні значення параметрів навчання та параметри структурної оптимізації:

- швидкість навчання: $\eta = 0.005$;
- коефіцієнт інерції: $\mu = 0$;

- коефіцієнт затування ваг: $\varepsilon = 0.1$;
- вірогідність активації нейрону прихованого шару: $p_h = 1$;
- вірогідність активації нейрону вхідного шару: $p_i = 1$.
- максимальна кількість мутацій при схрещенні: $M = 20$;
- кількість епох навчання початкової мережі: $T_0 = 50$;
- кількість епох навчання у ітерації: $T_i = 150$;
- види допустимих мутацій: усі;

На рис. 10 та 11 зображені залежності значення ціни та точності класифікації від кількості виконаних епох навчання.

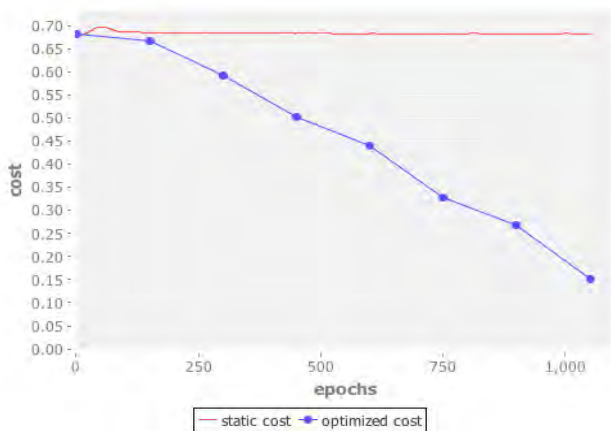


Рис.10. Значення ціни звичайної



Рис.11. Точність класифікації звичайної та оптимізованої мереж під час навчання

Протягом 7-ми ітерацій алгоритму було отримано [2-9-9-7-7-2] мережу з точністю класифікації 92.7%.

Висновки

В ході експериментальних досліджень проведений ряд дослідів запропонованого в [6] алгоритму, які доводять можливість використання його в задачах розпізнавання та класифікації даних.

Список посилань

1. Mezard M., and Nadal J.P. Learning in feedforward layered networks: The Tiling algorithm // Journal of Physics. – 1989. – V. A22. – P. 2191 - 2203,
2. Frean M. The Upstart Algorithm: A Method for Constructing and Training Feed-Forward Neural Networks // Tech. Rep. 89/469, Edinburgh Univ., 1989.
3. Ash T. Dynamic Node Creation in Back-Propagation Networks // Connection Science. – 1989. – V. 1.
4. Mozer M.C., Smolensky P. Skeletonization: a technique for trimming the fat from a network via relevance assessment // Advances in Neural Information Processing Systems. – 1989. – V. 1. – P. 107 - 115.
5. Дорогий Я.Ю. Ускоренний алгоритм обучения сверточных нейронных сетей / Я.Ю. Дорогий // Вісник НТУУ «КПІ», «Інформатика, управління та обчислювальна техніка», №57. – 2012. – С. 150-154.
6. Дорогий Я.Ю. Алгоритм структурної оптимізації нейронної мережі / Я.Ю.Дорогий // Вісник НТУУ «КПІ», «Інформатика, управління та обчислювальна техніка», №61. – 2014. – С. 169-173.
7. S. Thrun et al. The monk's problems: A performance comparison of diferent learning algorithms // Technical Report CMU-CS-91-197, Carnegie Mellon University, 1991.
8. Kevin J. Lang: Learning to Tell Two Spirals Apart / Kevin J. Lang, Michael J. // Proceedings of the 1988, Connectionist Models Summer School.