

УДК 519.21

*Леонід Тимченко
Василь Самойлов*

РОЗПОДІЛЕНА МАРШРУТИЗАЦІЯ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ

Різні дослідники показали, що методики машинного навчання в стані конкурувати із традиційними алгоритмами на деяких проблемах. У статті демонструється здатність простої однорівневої нейронної мережі знайти самі короткі шляхи від кожної вершини графа, працюючи розподілено. Ціль даної статті – показати, що розподілена нейронна мережа може виконувати маршрутизацію принаймні так само, як традиційно використовуваний алгоритм Дейкстри.

Вступ

Методики машинного навчання можуть успішно застосовуватися при вирішенні завдань, що виникають у телекомунікаційній галузі. Методика знаходження мережної топології, запропонована Хопфилдом і Тенком [1], використовується Тсенгом при розробці методу DIANNER [2]. У даній статті дослідження прагне відтворити дослідження Тсенга [2], щоб краще зрозуміти деталі реалізації.

Принцип роботи

Кожний маршрутизатор у комунікаційній мережі складається з однорівневої нейронної мережі. Деталізований опис використовуваних нейронних мереж лежить за рамками даної статті, з ними можна ознайомитися в [5]; досить сказати, що в кожній нейронній мережі є вхід для кожної можливої адреси в мережі й вихід для кожного порту на виходах маршрутизатора. Коли пакет прибуває, адресат подається на вхід нейронної мережі, і оцінка вартості шляху від поточного маршрутизатора до адресата виводиться на кожному відповідному порті. Інакше кажучи, вихід нейронної мережі для порта x буде оцінкою того, як довго пакет буде передаватися від цього маршрутизатора до адресата через порт x . У результаті пакет відправляється в порт, що відповідає самому маленькому рівню на виході.

Навчання

Перш ніж нейронна мережа маршрутизатора може видати значущі виводи, вона повинна спочатку "вивчити" витрати для кожного шляху до адресата. Початкові виводи нейронної мережі будуть повністю випадкові. Щоб поліпшити оцінки витрати на шляху, маршрутизатор відсилає "навчальні" пакети на кожному порті кожному адресатові в мережі. Ці пакети можуть бути маршрутизовані, використовуючи початкову випадкову конфігурацію мережі, або з використанням

© Тимченко Л.І., Самойлов В.М., 2007

протоколів широкомовної адресації. Якщо використовувати широкомовний метод, то пакети повинні мати такий параметр, як час життя та час відгуку на запит, тому що випадкова початкова конфігурація, швидше за все, буде містити петлі маршрутизації. У даній реалізації використовувалося обмеження за часом на маршрутизаторі, що відсилає, і поточна оцінна величина змінювалась, якщо не був отриманий відгук за певний період часу. Якщо використовувати протоколи широкомовної адресації при відсиланні пакетів, то збільшиться ймовірність перевантаження мережі, але в той же час збільшиться ймовірність прибуття пакетів до місця призначення.

Час проходження пакета в даній реалізації оцінюється в такий спосіб: навчальні пакети, відправлені до адреси призначення, підсумовують кількість часу, необхідного для проходження кожного зв'язку між маршрутизаторами при передачі по мережі. Іншими можливими схемами оцінки вартості передачі пакета можуть часові мітки або оцінка розмірів черги пакетів. Як тільки прибуває навчальний пакет, маршрутизатор адресата відсилає назад пакет підтвердження, що містить оцінку реальної вартості (відсилання можливо реалізувати, використовуючи широкомовну передачу, щоб можна було бути впевненими в доставці пакета з підтвердженням). Коли відсилаючий маршрутизатор одержує підтверджувальний пакет, він обновляє поточну оцінку через відповідний порт, використовуючи алгоритм навчання нейронної мережі[5].

Процес, що відбувається в комунікаційній мережі, у якій для маршрутизації використовується нейронна мережа, виглядає так:

1. Кожний маршрутизатор ініціалізується з випадковими оцінками
2. Кожний маршрутизатор відсилає навчальні пакети будь-якому маршрутизатору через кожний порт і обновляє поточну оцінку, коли вертаються пакети підтвердження.
3. Кожний маршрутизатор повторює крок 2, поки всі поточні оцінки не дорівнюють оцінкам у підтверджувальних пакетах.
4. Кожний маршрутизатор регулярно може повторювати кроки 2 і 3, щоб підтримувати шляхи в актуальному стані.

У навчальному циклі є один параметр, що можна налаштувати. Він називається нормою навчання. Для даних, представлених у цій статті, використовувалася норма навчання, яка дорівнювала 1. Норма навчання визначає, як швидко мережа виправляє свої оцінки, та ніколи не повинна бути більше 1. Наприклад, при значенні, рівному 1, оцінка змінюється таким чином, щоб точно відповідати даним, прийнятим у пакеті підтвердження. При значенні 0,5 поточна оцінка буде змінена на величину, рівну половині різниці між поточним значенням і оцінкою, отриманою в пакеті підтвердження. Швидше всього будуть побудовані маршрути в мережі при нормі навчання 1, виходячи із припущення, що оцінки, які приходять у пакетах підтвердження, точні. Якщо оцінки не зовсім точні або якщо бажано мати мережу, що повільно "приспосовується" до умов, що змінюються, необхідно використовувати більш низьку норму навчання.

Час навчання здебільшого залежить від числа ітерацій, помноженого на число маршрутизаторів і на середнє число портів на кожному маршрутизаторі. Число ітерацій при навчанні, загалом, є постійним (і досить низьким). Середнє число портів на маршрутизаторі – також невелика константа. Тому якщо допустити, що число ітерацій не пов'язане із числом маршрутизаторів, складність навчання лінійно залежить від числа вузлів у мережі, що простіше, ніж квадратична

залежність алгоритму Дейкстри, що використовується для рішення тих же завдань.

Експеримент

Методика, наведена вище, була застосована до мережі з 9 вузлами (рис. 1).

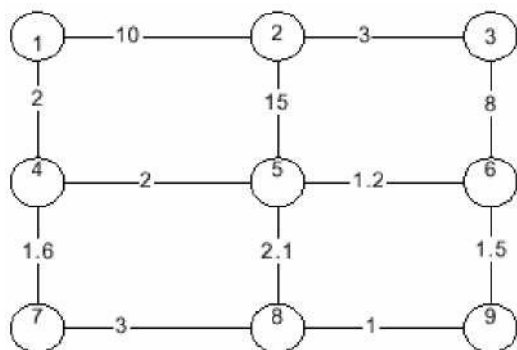


Рис. 1. Приклад мережі

Кожна вершина графа являє собою маршрутизатор з нейронною мережею, і мітки на дугах є ціною передачі між маршрутизаторами.

Після розрахунку маршрутів у даній мережі було змодельоване перевантаження, що змінило ціну зв'язку на такі значення, після чого знову було запущено перерахування маршрутів:

- дуга 2 – 5 – зросло від 15 до 20;
- дуга 4 – 5 – зросло від 2 до 21;
- дуга 6 – 5 – зросло від 1,2 до 10;
- дуга 8 – 5 – зросло від 2,1 до 11;

Результати

При використанні норми навчання, що дорівнює 1, кожний маршрутизатор навчився оптимальним шляхам до будь-якого іншого маршрутизатора у вищезгаданій мережі після 6 ітерацій навчання. Після того як було додане "перевантаження", мережа вивчила нові оптимальні шляхи після додаткових 4 ітерацій. В [2] використовували норму навчання 0,5 і одержували оптимальні шляхи після 9 ітерацій і після додаткових 11 ітерацій при "перевантаженні".

Висновок

Результати підтримують гіпотезу про те, що розподілена система маршрутизації на основі нейронної мережі може так само знаходити мережеві маршрути, як сучасні традиційні методи (OSPF). Якщо число ітерацій, необхідних для побудови маршрутизації залишається відносно постійними для великих комунікаційних мереж, цей новий метод маршрутизації дає кращі показники, чим квадратична залежність для мереж більших за 18 вузлів. Ця оцінка ігнорує константи для загальних квадратичних методів (як алгоритм Дейкстри), а це може означати, що цей алгоритм може бути ефективний при застосуванні в менших мережах.

Подальша робота повинна бути проведена у двох напрямках. По-перше, необхідно зробити додаткові перевірки. У цей процес повинне бути включене виконання того ж самого експерименту на декількох різних топологіях та з декількома різними конфігураціями. Потім перевірити на симуляторі з дискретним часом і, нарешті, в реальних мережах. Одним з найважливіших параметрів, який необхідно визначити, є такий: чи буде число ітерацій збільшуватися разом зі збільшенням мережі (або зі збільшенням середнього числа портів у маршрутизаторі)? Якщо ці числа залишаться відносно постійними, то складність завдання залишиться лінійною.

По-друге, майбутні дослідження можуть бути спрямовані на розширення можливостей алгоритму для вирішення більш цікавої проблеми маршрутизації, такої, як вибір шляху на основі безлічі критеріїв, особливо якщо це об'єднє

скорочення складності обчислень. У випадку позитивних результатів можна одержати багато випадків, коли при застосуванні методів, що навчаються, спрощується рішення проблем, які виникають при застосуванні більш традиційних алгоритмів.

ЛІТЕРАТУРА

1. *Hopfield J.J., Tank D.W.* Neural Computation of Decisions in Optimization Problems // *Biological Cybernetics* 52 (1985), p.141.
2. *Chiu-Che Tseng, Max Garzon.* Hybrid Distributed Adaptive Neural Router // *Proceeding of the ANNIE 1998.*
3. *Boyan, J.A., Littman M. L.* *Advances in Neural Information Processing Systems* // Morgan Kaufmann Publishers, Inc., 1994, p. 671-678.
4. *In Cowan J. D., Tesauro G., Alspector J.* eds. *Advances In Neural Information Processing Systems 6* // Morgan Kaufmann Publishers, 1994.
5. *Lee S.L., Chang S.* Neural Networks for routing of Communication Networks with Unreliable Components // *IEEE Transactions on Neural Networks* 4:5, 1993, pp.854-863.
6. *Mohamad H. Hassoun.* *Fundamentals of Artificial Neural Network* // MIT Press, Cambridge, MA, 1995.

Надійшла 27 вересня 2007 р.