

АДАПТИВНЫЙ ОПЕРАТОР МУТАЦИИ ДЛЯ НЕЙРОЭВОЛЮЦИОННОГО АЛГОРИТМА

Цой Ю.Р., Спицын В.Г.

Томский политехнический университет, кафедра

Вычислительной техники

qai@mail.ru, spitsyn@ce.cctpu.edu.ru

Аннотация

Рассматривается применение эволюционного подхода для настройки искусственной нейронной сети (ИНС). Описан полностью адаптивный оператор мутации для генетического алгоритма (ГА) для одновременного поиска топологии и обучения ИНС. Представлены результаты работы разработанного алгоритма (NEvA) для проблемы Исключающего ИЛИ и задачи адаптивного нейрорегулирования. Проводится сравнение полученных данных с результатами других алгоритмов.

Введение

Использование нейронных сетей подразумевает, помимо всего прочего, этап выбора структуры сети и ее обучение (настройка значений весов межнейронных связей) [1]. Для решения этих подзадач удобно использовать эволюционные принципы, отличительной чертой которых является робастность и эффективность работы в сложном многоэкстремальном пространстве поиска. При этом подзадачи можно решать как отдельно, последовательно [2, 3], так и одновременно [4-9]. Одним из видов эволюционных вычислений являются генетические алгоритмы [10].

В настоящий момент разработано немало алгоритмов [4, 6, 8, 9], позволяющих производить поиск топологии ИНС одновременно с настройкой значений весов связей. При этом могут накладываться ограничения на число скрытых нейронов [5, 7], количество входных связей нейрона [8], направление «роста» сетей (усложнение / упрощение) [6]. Генетическое представление ИНС может быть как прямым, в виде вектора атрибутов сети [5, 6, 9], так и с использованием специальных грамматик [4, 8].

В данной работе рассматривается эволюционный алгоритм для одновременной настройки структуры и обучения ИНС, использующий адаптивный оператор мутации, в котором тип мутации и ее вероятность определяются для каждой особи отдельно, в зависимости от характеристик сети, соответствующей этой особи.

Алгоритм NEvA

Подробное описание алгоритма приведено в [9]. Здесь будут описаны только основные характеристики. В предлагаемом алгоритме рассматривается популяция нейронных сетей. Генетическая информация каждой особи содержит информацию о связях сети и их весах.

Использовалась селекция усечением [11] с порогом 50%. В процессе репродукции получаются два потомка, которые заимствуют общие нейроны

и связи обоих родителей, при этом веса родительских особей рекомбинируют с использованием 2-точечного бинарного кроссовера. Различные нейроны и связи «разыгрываются» между потомками случайным образом.

В начальной популяции все особи представляют сети без скрытых нейронов, все входные нейроны которых соединены с выходными. Веса сетей являются случайными величинами из диапазона $[-0,5; 0,5]$. Использовались нейроны без смещений с лог-сигмоидной функцией активации.

Адаптивный оператор мутации

Механизм мутации предусматривает добавление и удаление нейронов и связей, а также изменение случайной связи на произвольную величину. Выбор типа мутации производится в соответствии со следующими эвристиками:

$$f_c = \frac{N_c}{2^{FB-1} [N_N(N_N - 1) - N_I(N_I - 1) - (1 - FB)N_O(N_O - 1)]}, \quad (1)$$

$$f_N = \frac{N_I + N_O}{N_N}, \quad (2)$$

где N_c – количество связей в сети, N_I , N_O , N_N – соответственно количество входных, выходных нейронов и общее число нейронов в сети, FB – флаг, обозначающий, разрешено появление обратных связей ($FB=1$) или нет ($FB=0$). Необходимо отметить, что связи от скрытых нейронов к выходным могут появляться в любом случае. Коэффициент f_c необходим для оценивания степени «связанности» нейронов в сети, т.е. чем большее число возможных связей реализовано, тем больше значение коэффициента. Использование второго коэффициента f_N основано на эвристическом предположении, что чем больше количество входных и выходных сигналов нейронной сети, тем, возможно, более сложная структура необходима для решения задачи.

Для выбора вида мутации использовались коэффициенты, вычисленные по формулам (1) и (2) и возведенные в квадрат, обозначим их как F_c и F_N . Схематично алгоритм выбора вида мутации можно представить следующим образом:

- 0 Начало
- 1 Если $CB > F_c$, то 2, иначе 3
- 2 Если $CB > F_N$, то 2.1, иначе 2.3
 - 2.1 Если $N_H > 0$, то 2.2, иначе 2.3
 - 2.2 Удалить случайный нейрон. Перейти на 5.
 - 2.3 Добавить связь. Перейти на 5.
- 3 $F_H = F_N * F_c$
- 4 Если $CB > F_H$, то 4.1, иначе 4.4
 - 4.1 Если $(CB > F_c \text{ и } N_H > 0)$, то 4.2, иначе 4.3
 - 4.2 Удалить случайный нейрон. Перейти на 5.
 - 4.3 Удалить случайную связь. Перейти на 5.
 - 4.4 Добавить нейрон.
- 5 Конец

Здесь CB – случайная нормализованная величина, N_H – количество скрытых нейронов в мутующей сети.

Условно весь алгоритм можно разделить на две «ветви» по первому условному переходу:

1. Ветвь увеличения f_C (усложнение структуры сети).
2. Ветвь уменьшения f_C (упрощение структуры сети).

Поскольку удаление нейрона может привести как к уменьшению, так и к увеличению f_C , в зависимости от количества связей нейрона, то этот вариант присутствует в обеих ветвях. Таким образом, основным фактором для регуляции структуры получаемых ИНС является степень их «связанности».

Такой подход, с одной стороны, слабо ограничивает «сверху» количество скрытых нейронов, а с другой – препятствует быстрому увеличению числа нейронов в сети, т.к. добавление каждого нового нейрона в сеть будет происходить с меньшей вероятностью.

Не только тип мутации выбирается адаптивно, но и вероятность мутации. Величину вероятности мутации принято выбирать небольшой, т.к. в противном случае, при P_m больше некоторого значения (так называемый порог ошибки [12]), эволюция становится «похожа» на случайный поиск. В силу этого вероятность мутации равная $1/L$, где L – длина хромосомы (в данном случае число связей), представляется достаточно разумной, т.к. предполагает в среднем одну мутацию для каждой особи. Однако на более поздних этапах эволюции P_m лучше уменьшить, чтобы снизить вероятность разрушения «хороших» особей. Поэтому было решено разыгрывать вероятность проведения оператора мутации $N_I \cdot N_O$ раз, где N_I и N_O – соответственно число входных и выходных нейронов в сети. Таким образом, с учетом того, что используется концепция растущих нейросетей, можно сказать, что эффект от мутации уменьшается с течением времени. Т.е. значение L в ходе эволюции увеличивается, и вероятность однократной мутации, равная $\frac{N_I \cdot N_O}{L}$, становится меньше.

Результаты работы

Для проверки алгоритма были выбраны задачи классификации (Исключающее ИЛИ) и нейрорегуляции (проблема перевернутого маятника с полной информацией [13]).

Результаты экспериментов показали высокую эффективность разработанного алгоритма. В проблеме Исключающего ИЛИ предлагаемый алгоритм «проиграл» градиентным методам, однако стоит отметить изначально большую сложность задачи, а также двукратный выигрыш у простого ГА [10] и преимущество перед NEAT [6, 14]. Для перевернутого маятника с 1 шестом алгоритм незначительно уступил только алгоритму ESP [15], значительно опередив SANE [5] и GENITOR [16]. В случае с двумя шестами результаты NEvA оказались лучше результатов всех остальных алгоритмов [6, 15, 17].

Источники

1. Круглов В.В., Борисов В.В. Искусственные нейронные сети: Теория и практика. – 2-е изд. – М.: Горячая линия-Телеком, 2002. – 382 с.: ил.
2. Fiszlelew A., Britos P., Perichisky G., Garsna-Martinez, R. Automatic Generation of Neural Networks based on Genetic Algorithms. // Information Systems Electronic Review. – №1(2) – 2003.
3. Хомич А. В., Жуков Л. А. Эволюционный метод оптимизации структуры нейронной сети с учителем. // Труды "Нейроинформатика-2005". – Москва, 2005. В печати.
4. Gruau, F. Genetic synthesis of modular neural networks. // Proceedings of the ICGA-93. – San Francisco, California. – Morgan Kaufmann, 1993.
5. Moriarty D.E., Miikkulainen R. Efficient Reinforcement Learning through Symbiotic Evolution. // Machine Learning. – 1996. – № 22. – P. 11-33.
6. Stanley K., Miikkulainen R. Evolving Neural Topologies through Augmenting Topologies // Evol. Comp. The MIT Press. – 2002 – №10(2). – P. 99-127.
7. Igel C., Kreutz M. Operator adaptation in evolutionary computation and its application to structure optimization of neural networks. // Neurocomputing. Elsevier Science B.V. – 2002.
8. Ferreira C. Designing Neural Networks Using Gene Expression Programming. // Proceedings of 9th Online World Conference on Soft Computing in Industrial Applications. – 2004.
9. Цой Ю.Р., Спицын В.Г. Применение генетического алгоритма для решения задачи адаптивного нейрорегуляции. // Труды "Нейроинформатика-2005". – Москва, 2005. В печати.
10. Holland J. Adaptation in Natural and Artificial Systems. – The University of Michigan Press, Ann Arbor, 1975.
11. Bickel T., Thiele L. A Comparison of Selection Schemes used in Genetic Algorithms (2nd Edition). TIK-Report No.11. – Zurich, Switzerland, 1995.
12. Nowak M., Schuster P. Error thresholds of replication in finite populations: Mutation frequencies and the onset of Muller's ratchet. // Theoretical Biology. – 1989. – №137 – P.375-395.
13. Wieland, A. Evolving neural network controllers for unstable systems. // Proceedings of the International Joint Conference on Neural Networks. – Piscataway, New Jersey, 1991. – P. 667-673.
14. James D., Tucker P. A Comparative Analysis of Simplification and Complexification in the Evolution of Neural Network Topologies. // Proceedings of GECCO 2004. – Washington, USA, 2004.
15. Gomez F., Miikkulainen R. Incremental evolution of complex general behavior. // Adaptive Behavior. –1997. – №5. – P. 317-342.
16. Whitley D., A genetic algorithm tutorial. // Statistics and Computing. – 1994. – №4 – P.65-85.
17. Saravanan N., Fogel D. B. Evolving neural control systems. // IEEE Expert. – 1995. – P. 23-27.