УДК 004.056

Нейросетевой подход к иерархическому представлению компьютерной сети в задачах информационной безопасности

М.А. Басараб [1], С.В. Вельц [1]

[1] МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Рассмотрена задача создания иерархического представления компьютерной сети. Задача сведена к задаче о покрытии множества; предложен нейросетевой подход на основе сетей Хопфилда для ее решения.

E-mail: bmic@mail.ru

Ключевые слова: *задача о покрытии множества, нейросетевая оптимизация, сеть Хопфилда, машина Больцмана.*

Рассматриваемая задача создания иерархического представления компьютерной сети, пригодного для проведения многомасштабного анализа методами машинного обучения, является актуальной. Ее решение позволит эффективнее управлять сетями, повышая качество обслуживания, и обнаруживать новые типы аномалии для обеспечения информационной безопасности. Под многомасштабностью понимают применение выбранного метода машинного обучения к определенному уровню представления, что дает возможность выбирать степень обобщения информации, обнаруживать взаимосвязи между частями системы в определенном масштабе, снижать вычислительную сложность задачи.

В качестве инструментов многомасштабного анализа могут выступать графовые нейронные сети (Graph Neural Network — GNN) [1, 2] или иерархическая темпоральная память (HTM) [3]. И теми, и другими методами обрабатывают информацию, представленную в виде графа. Однако нейроны в GNN получают только локальную информацию от своих соседей, что не позволяет обобщать и находить взаимосвязи между разными частями графа. В случае HTM необходимо сформировать структуру связей между узлами/нейронами в разных слоях. В обоих случаях встает задача выбора подмножества вершин графа для формирования следующего слоя. При этом любая вершина графа должна либо входить в выбранное подмножество, либо быть смежной хотя бы с одной вершиной из выбранного подмножества. Это приводит к задаче о вершинном покрытии, которая сводится к задаче о покрытии множества (SCP).

Поскольку задача о покрытии множества является NP-полной [4] и размер задач, требующих решения, достаточно велик, найти точное решение с помощью вычислений сложно. Это обусловливает необходимость применения приближенных эвристических алгоритмов, сре-

ди которых можно отметить методы лагранжевой релаксации [5 — 7], генетические алгоритмы [8 — 11], поиск с запретами [12], алгоритмы муравьиной колонии [13], нейронные сети [14, 15]. Авторами данной работы предлагается нейросетевой метод для решения задачи о покрытии множества. Отличие этого метода заключается в использовании машины Больцмана совместно с методом Монте-Карло для борьбы с локальными минимумами. Также в данной работе предложена эвристика для оценки метапараметров модели.

Нейронные сети выбраны из следующих соображений. Во-первых, в качестве метода машинного обучения можно применять и нейронные сети (GNN или HTM). Использование нейросетевого алгоритма и на этапе построения представления позволит получить комплексное решение проблемы в рамках парадигмы нейрокомпьютинга. Во-вторых, нейросетевые архитектуры по своей сути обладают параллелизмом, благодаря чему возможна реализация алгоритма для параллельных вычислительных архитектур.

Постановка задачи. Перед тем как поставить задачу, неформально опишем требуемый результат.

Целью является последовательное формирование уровней иерархического представления сети. Для этого на каждом шаге необходимо выбрать подмножество элементов сети так, чтобы оно вместе со смежными элементами покрывало всю сеть. Желательно, чтобы размер этого подмножества был минимален и количество смежных элементов, которые покрываются выбранным элементом (степень вершины), было небольшим. Последнее требование связано с тем, что возрастание степени вершины ведет к увеличению длины вектора признаков для алгоритмов машинного обучения и соответственно к экспоненциальному увеличению требуемого размера обучающей выборки.

Перейдем к формальной постановке задачи в виде SCP.

Пусть даны множество M, $|M| = n$ и множество S подмножеств M $S = \{S_1, ..., S_m\}$. Пусть задана матрица A размерности $n \times m$, $a_{ij} = 1$, если $a_i \in S_j$, иначе $a_{ij} = 0$.

Обозначим: a_i — i-ю строку матрицы A; x — вектор-столбец, кодирующий решение, $x = (x_1, ..., x_m)$, где $x_i = 1$, если S_i входит в решение, и 0 — если иначе; c — вектор-столбец цен подмножеств; s — вектор-столбец размеров подмножеств, $s = (S_1, ..., S_m)$.

Введем целевой функционал:

$$Q_0(x) = K_1 cx + K_2 \sum_{i=1}^{n} x_i \left|S_i\right| = \left(K_1 c + K_2 \left|S_i\right|\right) x. \qquad (1)$$

Первое слагаемое задает цену решения. Второе задает штраф за выбор подмножеств с большой степенью. Коэффициенты K_1, K_2 определяют относительное влияние штрафов на общее решение.

Задача формулируется как задача дискретной оптимизации с ограничениями:

$$\Pi_{\min} \quad Q_0(x). \tag{2}$$
$$x,s.t.\text{решение корректное}$$

Решение X называется корректным, если оно покрывает все элементы множества M, т. е. $M = \sum_{S_i \in X} S_i$.

Нейросетевой метод решения. Задача (1) — эквивалентная задача о покрытии множества. Подробный обзор методов решения приведен в [16]. Можно выделить точные алгоритмы, жадные алгоритмы (Хватал), генетические, нейросетевые. Здесь подробнее остановимся на последних.

Исторически применение нейронных сетей для решения задач комбинаторной оптимизации было направлено в основном на задачу коммивояжера [17]. Подробный обзор нейросетевых методов комбинаторной оптимизации дан в [15].

Рассмотрим дискретную нейронную сеть Хопфилда из n {–1,+1}-нейронов.

Состояния нейронов кодируются {–1, +1}-вектором $y = (y_1, ..., y_n)$. Решение кодируется {0, 1}-вектором $x = (y+1)/2$. Нейроны обрабатываются асинхронно и в случайном порядке (условие отсутствия осцилляций в динамике сети). Начальное состояние нейронов выбирается случайным образом.

Учет ограничений в нейронных сетях при комбинаторной оптимизации заключается в модификации выражения энергии сети. Общий подход разработан в [18] для двухслойной модели Extended Hopfield Model (EHM). Также может быть модифицирован механизм активации нейронов [19].

Для получения корректного решения (т. е. покрывающего все элементы M) модифицируем целевой функционал Q_0 так, чтобы он включал дополнительный штраф за непокрытые вершины:

$$Q(x) = (K_1 c + K_2 s)x + K_3 \sum_{i=1}^{n} h(a_i x), \tag{3}$$

где $h(u)$ — функция штрафа за покрытие элемента u раз, например: $h(u) = c_0[u = 0]$ или $h(u) = c_0(u-1)^2$. Далее рассмотрим эти варианты подробнее.

Для определения значений коэффициентов используется следующая эвристика: в результате решения задачи алгоритмом (1) стоимость выбранного подмножества должна быть равна выигрышу от покрытия вершин. Коэффициенты K_1 и K_2 приравнивают 1 и 0 соответственно и определяется K_3.

Несмотря на ***пост-обработку решения***, некорректные решения появляются часто [15]. Для того чтобы решить эту проблему, приходится применять дополнительные шаги пост-обработки на основе жадного алгоритма Хватала [20, 21]

Алгоритм 1. Жадный алгоритм пост-обработки решения:
Вход: M, S
Выход: X — подмножество S
while есть непокрытые элементы:
$U := M \cdot \sum_{S_i \in X} S_i$ — множество непокрытых элементов
$x := \text{argmax}_{S_i} \left\{ \frac{|S_i \cap U|}{c_i} \right\}$ — оптимальное множество
$X := X \cup x$
endwhile
while истина:
$T = \{x \in X | X \cdot x$ — корректное решение$\}$
if $|T| = 0$ **then** прервать цикл **endif**
$x := \text{argmax}_{v_i \in T} c_i$
$X := X \cdot x$
endwhile
return X.

Энергия сети и функция Ляпунова. Энергию нейронной сети Хопфилда задают стандартным выражением:

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} I_i x_i. \qquad (4)$$

В [22] показано, что этот функционал является функцией Ляпунова, т. е. $E(\hat{x}) = 0$ (достаточно $E(\hat{x}) = \min_{x \in B} E(x)$), B — некоторая локальная область притяжения), $E(x) > 0$, $x \neq \hat{x}$ (знакоположительность) и $dE/dt < 0$ (диссипативность системы), где $\hat{x}$ — устройчивое состояние (точечный аттрактор). Из этого следует, что система стремится к устойчивому состоянию.

Замечание 1. Функция (4) является квадратичной формой, но несмотря на это, у нее может быть несколько локальных минимумов, так как модель дискретная (в непрерывном случае есть еще интеграл) и значения x ограниченны.

Замечание 2. Функция (4) для дискретной сети, в непрерывном случае еще есть интеграл $\int_0^{x_j} \phi_j^{-1}(x)dx$ (см. [22]). Однако в статьях часто встречается непрерывная модель, например, функции активации $g(u) = \frac{1}{2}\left(1 + \text{th}(ku)\right)$ или $g(u) = \frac{1}{1 + \exp(-ku)}$, при этом выражение (4) используется в качестве энергии сети и функции Ляпунова.

Определение параметров модели. Рассмотрим, как, имея целевой функционал, определить параметры модели (веса и смещения нейронов).

1. *Квадратичная функция штрафа*.

Пусть $h(x) = C_p(x-1)^2$, где C_p — коэффициент штрафа.

Замечание 3. Это неудачная функция, так как штрафуем двукратное покрытие так же, как и непокрытие.

Замечание 4. Квадратичная функция широко используется в статьях, так как позволяет получить удобные аналитические выражения.

Раскрывая (3) и приравнивая к (4), получаем веса и смещения:

$$w_{jk} = -\frac{C_p}{2}\sum_{i=1}^{n} a_{ij}a_{ik}; \tag{5}$$

$$I_j = -\frac{1}{2}\left(c_j - 2C_p\sum_{k=1}^{n} a_{jk} + C_p\sum_{k=1}^{n}\sum_{i=1}^{n} a_{ik}a_{ij} \right). \tag{6}$$

Отметим, что $w_{ij} = w_{ji}$, но $w_{ii} \neq 0$, это является необходимым условием устойчивости работы сети [20] (в случае $w_{ii} \neq 0$ возможна осцилляция сети). Для того чтобы это компенсировать, можно изменить условие остановки процесса оптимизации с учетом возможных осцилляций.

2. *Разрывная функция штрафа*.

Пусть теперь $h(x) = C_p\left[x = 0\right]$, где C_p — коэффициент штрафа. Проблема с данным функционалом заключается в том, что он разрывен и напрямую получить выражение для параметров модели не удается.

Эту проблему можно решить двумя способами:

1) использовать непрерывную аппроксимацию, например $\hat{h}(x) = \dfrac{C_p}{1+\exp(\alpha x)}$, а для получения выражений параметров сети раскладывать получившийся целевой функционал в ряд Тейлора в текущей точке. Поскольку текущая точка постоянно изменяется, разложение потребуется постоянно пересчитывать, что негативно влияет на быстродействие. К тому же разложение корректно в небольшой окрестности текущей точки, а движение осуществляется по вершинам гиперкуба;

2) модифицировать механизм активации нейронов — использовать идею, похожую на машину Больцмана. В нейроне будем принимать решение не пороговым правилом на основе локального потенциала $\sum_{i=1}^{n} w_{ij}x_i - \theta_j$, а считать целевые функционалы для разных состояний

рассматриваемого нейрона и разность между ними $\Delta Q_i = Q(x_i = 1) - Q(x_i = 0)$ с вероятностью $P[x_i = 1] = \frac{1}{1 + \exp(-\Delta Q_i / T)}$, T — «температура» сети. Будем переводить нейрон в возбужденное состояние.

Отметим, что возможность выбирать состояния, в которых целевой функционал возрастает, позволяет сети выходить из локальных минимумов.

Результаты вычислительного эксперимента. Описанный выше алгоритм был реализован на языке Scala (объектно-функциональное расширение Java). Оценка эффективности предложенного алгоритма проведена на тестовых задачах из набора OR Library. Набор включает в себя 65 задач различной размерности.

Поскольку результат применения алгоритма зависит от случайного начального состояния нейронной сети, для каждой задачи проводили пять запусков и результаты усредняли. Результаты расчетов приведены в табл. 1.

Для задач классов E—H оптимальные решения неизвестны из-за большой размерности задач. Для оценки были использованы предполагаемые решения, указанные в [8]. Результаты расчетов приведены в табл. 2.

В 57 задачах из 65 предложенный нейросетевой алгоритм показал результат лучше, чем алгоритм Хватала. Средняя цена решения алгоритмом Хватала — 207,6, предложенным алгоритмом — 204,3. Средняя предполагаемая оптимальная цена — 196,6. Среднее время расчетов — 14 с на задачу.

Построение иерархического представления. Используя решение SCP, можно создать алгоритм построения иерархического представления. Принцип работы заключается в итерационном решении SCP.

Алгоритм 2. Построение иерархического представления на основе SCP.

Вход: граф $G = (V, E)$, цены вершин c.

Выход: $N = ((G_1, A_1), (G_2, A_2), \ldots, (G_L, A_L))$, где A_i — данные о принадлежности вершин.

$N := ()$ — сначала слоев нет

$V_c := V$

$E_c := E$

while $|V_c| > 1$: — формируем новые слои, пока не получим одну вершину в слое.

$S :=$ SolveSCP(V_c, E_c) — S является подмножеством V_c

$A := \{\}$

$V_c := \{\}$

$E_c := \{\}$

head : = {} — отображение множества эквивалентных вершин в представителя

Таблица 1

Результаты вычислительного эксперимента по сравнению алгоритмов

Номер задачи	Решение			Номер задачи	Решение		
	оптимальное	жадным алгоритмом	нейронной сетью		оптимальное	жадным алгоритмом	нейронной сетью
4.1	429	438	437	6.4	131	137	136
4.2	512	551	551	6.5	161	178	178
4.3	516	546	539	A.1	253	271	262
4.4	494	510	506	A.2	252	267	261
4.5	512	519	520	A.3	232	244	239
4.6	560	594	586	A.4	234	246	242
4.7	430	449	447	A.5	236	247	240
4.8	492	502	502	B.1	69	73	71
4.9	641	672	658	B.2	76	78	78
4.10	514	521	517	B.3	80	85	81
5.1	253	271	268	B.4	79	85	84
5.2	302	329	328	B.5	72	76	74
5.3	226	232	230	C.1	227	246	239
5.4	242	253	251	C.2	219	231	229
5.5	211	220	216	C.3	243	256	256
5.6	213	234	228	C.4	219	239	236
5.7	293	311	306	C.5	215	228	220
5.8	288	308	301	D.1	60	68	65
5.9	279	290	290	D.2	66	70	67
5.10	265	274	273	D.3	72	78	75
6.1	138	147	145	D.4	62	65	63
6.2	146	160	153	D.5	61	71	63
6.3	145	152	149				

Таблица 2

Результаты вычислительного эксперимента для задач большой размерности

Номер задачи	Решение			Номер задачи	Решение		
	оптимальное	жадным алгоритмом	нейронной сетью		оптимальное	жадным алгоритмом	нейронной сетью
E.1	29	31	29	G.1	179	194	186
E.2	30	34	32	G.2	158	165	163
E.3	27	32	28	G.3	169	179	173
E.4	28	32	29	G.4	172	184	178
E.5	28	31	28	G.5	168	181	179
F.1	14	17	15	H.1	64	71	68
F.2	15	16	15	H.2	64	69	68
F.3	14	16	15	H.3	60	65	64
F.4	14	15	15	H.4	59	66	63
F.5	14	15	14	H.5	55	62	59

for v **in** S:

$A_i := v \cup$ {смежные с v вершины}

$A := A \cup A_i$

$V_c := V_c \cup v$

head $A_i := v$

endfor

for a **in** A, **for** b **in** $A \setminus a$: — создаем ребра

if $\exists v_i \in a \wedge v_j \in b : (v_i, v_j) \in E$ **then**:

$E_c := E_c \; v_i, v_j \cup$ (head a, head b)

endif

endfor

$N := N \cup ((V_c, E_c), A)$ — добавляем слой

endwhile

return N

конец

Пример работы алгоритма 2 представлен на рисунке.

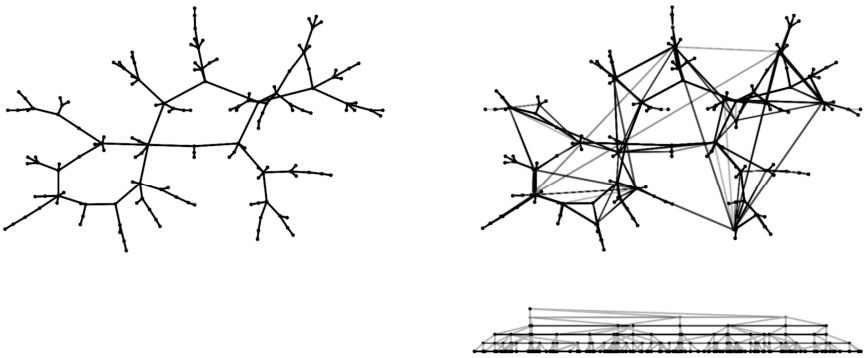

Схема работы алгоритма 2

В данной работе рассмотрено решение задачи о покрытии множества в рамках нейросетевой парадигмы и на его основе предложен алгоритм построения иерархического представления компьютерной сети. Описанные результаты могут найти применение при построении систем обнаружения аномалий в сетях и прогнозирования поведения трафика.

СПИСОК ЛИТЕРАТУРЫ

1. Scarselli F. A short description of the Graph Neural Network toolbox. 2011. URL: http://www.dii.unisi.it/~franco/Research/GNN_DOC.pdf
2. The graph neural network model / F. Scarselli, M. Gori, A.C. Tsoi, G. Monfardini // IEEE Transactions on Neural Networks. 2009. Vol. 20. No 1. P. 61–80.
3. Dileep G., Bobby Jaros. The HTM learning algorithms // Numenta Inc., 2007. URL: http://www.numenta.com
4. Karp R.M. Reducibility among combinatorial problems: Complexity of computer computations. Proc. of a Symp. on the Complexity of Computer Computations / R.E. Miller, J.W. Thatcher, eds. [The IBM Research Symposia Series]. N.-Y.: Plenum Press, 1972. P. 85–103.
5. Balas E., Carrera M.C. A dynamic subgradient-based branch and bound procedure for set covering // Oper. Res. 1996. Vol. 44. No 6. P. 875–890.
6. Beasley J.E. A Lagrangian heuristic for set-covering problems // Naval Res. Logist. 1990. Vol. 37. No 1. P. 151–164.
7. Caprara A., Fischetti M., Toth P. Algorithms for the set covering problem // DEIS — Operations Research Group. 1998. Technical Rep. No OR-98-3.
8. Нгуен Минь Ханг. Применение генетического алгоритма для задачи нахождения покрытия множества // Тр. ИСА РАН. 2008. № 33. С. 206–219.
9. Еремеев А.В. Генетический алгоритм для задачи о покрытии // Дискретный анализ и исследование операций. Сер. 2. 2000. Т. 7. № 1. С. 47–60.
10. Back Th., Schiiltz M., Khuri S. A comparative study of a penalty function, a repair heuristic, and stochastic operators with the set-covering

problem // Artificial Evolution. Proc. Berlin: Springer, 1996. P. 3–20. (Lecture Notes in Comput. Sci.; Vol. 1063.)
11. Beasley J.E., Chu P.C. A genetic algorithm for the set covering problem // European J. Oper. Res. 1996. Vol. 94. No. 2. P. 394–404.
12. Ramalhinho H., Pinto R., Portugal R. Metaheuristics for the bus-driver scheduling problem // Univ. Pompeu Fabra. Economic Working Papers Series. Technical Rep. 1998. No 304.
13. Alexandrov D., Kochetov Yu. Behavior of the ant colony algorithm for the set covering problem // Operations Research Proc. 1999 (Magdeburg, 1999). Berlin: Springer, 2000. P. 255–260.
14. Grossman T., Wool A. Computational experience with approximation algorithms for the set covering problem // European J. Oper. Res. 1997. Vol. 101. No 1. P. 81–92.
15. Smith K.A. Neural networks for combinatorial optimization: A review of more than a decade of research // Informs journal on Computing. 1999. Vol. 11. No 1. P. 15–34.
16. Еремеев А.В., Заозерская Л.А., Колоколов А.А. Задача о покрытии множества: сложность, алгоритмы, экспериментальные исследования // Дискретный анализ и исследование операций. 2000. Т. 7. С. 22–46.
17. Hopfield J.J., Tank D.W. “Neural” computation of decisions in optimization problems // Biological Cybernetics. 1985. No 52. P. 141–152.
18. Abe, Kawakami, Hirasawa. Solving inequality constrained combinatorial optimization problems by the Hopfield neural network // Neural Networks. 1992. Vol. 5. No 4. P. 663–670.
19. Le Gall, Zissimopoulos. Extended hopfield models for combinatorial optimization // IEEE Transactions on Neural Networks. 1999. Vol. 10. No 1. P. 72–80.
20. Cohen, Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks // IEEE transactions on systems, man, and cybernetics. 1983. No 5. P. 815–826.
21. Chvatal V.A. Greedy heuristic for the set-covering problem // Mathematics of Operations Research. 1979. Vol. 4. No 3. P. 233–235.
22. Хайкин С. Нейронные сети: полный курс: пер. с англ. 2-е изд. М.: Вильямс, 2006. 1104 с.

Статья поступила в редакцию 25.10.2012