НАУЧНОЕ ИЗДАНИЕ МГТУ ИМ. Н. Э. БАУМАНА

НАУКА и ОБРАЗОВАНИЕ

Эл № ФС77 - 48211. Государственная регистрация №0421200025. ISSN 1994-0408

электронный научно-технический журнал

Иерархическое представление компьютерной сети на основе нейронной сети Хопфилда

 # 09, сентябрь 2013

DOI: 10.7463/0913.0630141

Басараб М. А., Вельц С. В.

УДК 004.056

Россия, МГТУ им. Н.Э. Баумана

bmic@mail.ru

svelts@rambler.ru

Введение

В данной работе рассматривается задача создания иерархического представления компьютерной сети, пригодного для проведения многомасштабного анализа методами машинного обучения. Решение данной задачи позволит эффективнее управлять сетями для обеспечения качества обслуживания и обнаруживать новые типы аномалии для обеспечения информационной безопасности. Под многомасштабностью понимается применение выбранного метода машинного обучения к определённому уровню представления, что даст возможность выбирать степень обобщения информации, обнаруживать взаимосвязи между частями системы на определённом масштабе, снижать вычислительную сложность задачи.

В качестве инструментов многомасштабного анализа могут выступать графовые нейронные сети (GNN – Graph Neural Network) [3,4] или иерархическая темпоральная память (HTM) [2]. И тот, и другой методы могут обрабатывать 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 встаёт 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.

В данной работе рассматривается задача о покрытии множества (SCP), к которой сводится задача о вершинном покрытии. Поскольку данная задача является NP-полной [9] и размер задач, которые требуется решать, достаточно велик, нахождение точного решения вычислительно сложно. Это обуславливает необходимость применения приближённых эвристических алгоритмов. Среди эти алгоритмов можно отметить методы лагранжевой релаксации [13–15], генетические алгоритмы [11, 16–18], поиск с запретами [19], алгоритмы муравьиной колонии [20], нейронные сети [5, 7]. В статье предлагается нейросетевой метод для решения задачи о покрытии множества. Отличие предлагаемого 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 могут применяться тоже 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, что позволит реализовать алгоритм для параллельных вычислительных архитектур.

Далее в разделе 1 формулируется решаемая задача как комбинаторная задача оптимизации с ограничениями. В разделе 2 описан нейросетевой метод решения. В разделе 3 приведены результаты вычислительного эксперимента на стандартных задачах из OR Library. Наконец, в разделе 4, приводится алгоритм получения иерархического представления сети на основе SCP.

1 Постановка задачи

Перед тем как поставить задачу, неформально опишем требуемый результат.

Конечная цель заключается в последовательном формировании уровней иерархического представления сети. Для этого на каждом шаге надо выбрать подмножество элементов сети так, чтобы это подмножество 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 ведёт к увеличению длины вектора признаков для алгоритмов машинного обучения и, соответственно, к экспоненциальному увеличению требуемого размера обучающей выборки.

Перейдём к формальной постановке задачи в виде SCP.

Пусть дано множество M, $|M|=n$, и множество S подмножеств M:

$$S=\{S_1,\ldots, S_m\}.$$

Зададим матрицу A размера $n \times m$, $a_{ij}=1$, если $a_i \in S_j$, иначе $a_{ij}=0$.

Обозначим через

a_i – i-ю строку матрицы А;

x – вектор-столбец, кодирующий решение, $x=(x_{1,}\ldots,x_m)$, где $x_i=1$, если S_i входит в решение и $x_i=0$ иначе;

c – вектор-столбец цен подмножеств;

s – вектор-столбец размеров подмножеств, $s=(S_{1,}\ldots,S_m)$.

Введём целевой функционал:

$$Q_0(x)=K_1cx+K_2\sum_{i=1}^{n}x_i|S_i|=(K_1c+K_2|S_i|)x. \tag{1}$$

Первое слагаемое задаёт цену решения. Второе задаёт 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:

$$Q_0^*=\min_x Q_0(x). \tag{2}$$

Решение X называется корректным, если оно покрывает все элементы множества М, т.е.

$$M=\sum_{s_i\in X}S_i.$$

2 Нейросетевой метод решения

Задача (1) эквивалентна задаче о покрытии множества. Подробный обзор методов ее решения приведён в [10]. Среди алгоритмов можно выделить точные, жадные (Хватал), генетические, нейросетевые. Подробнее остановимся на последних.

Исторически применение нейронных сетей для решения задач комбинаторной оптимизации было направлено, в основном, на решение задачи коммивояжера (TSP) [6]. Подробный обзор нейросетевых методов комбинаторной оптимизации дан в [7].

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

Состояния нейронов кодируются {-1,+1}-вектором $y=(y_{1,}\ldots,y_n)$. 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.

Учёт ограничений. Учёт ограничений в нейронных сетях при комбинаторной оптимизации заключается в модификации выражения энергии сети. Общий подход разработан в [1] для двухслойной модели Extended Hopfield Model (EHM). Также может модифицироваться механизм активации нейронов [8].

Для получения корректного решения (т. е. покрывающего все элементы *M*) модифицируем целевой функционал Q_0, 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 мы рассмотрим этим 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. K_1 и K_2 приравниваются 1 и 0 соответственно и определяется K_3.

Пост-обработка решения. Несмотря на вышесказанное, некорректные решения появляются достаточно часто [7]. Для решения этой проблемы приходится применять дополнительных шаг пост-обработки на основе жадного алгоритма Хватала [22, 23]:

Алгоритм 1: <u>Жадный алгоритм пост-обработки решения:</u>

Вход: *M, S*

Выход: *X* — подмножество *S*

while есть непокрытые элементы:

$U := M \sum_{S_i \in X} S_i$ – множество непокрытых элементов,

$x := \arg\max_{S_i} \left\{ \frac{|S_i \cap U|}{c_i} \right\}$ – оптимальное множество,

$X := X \cup x$.

endwhile

while истина:

$T = \{x \in X \mid X \cdot x - корректное\ решение\}$

if |T| = 0 **then** прервать цикл **endif**

$x := \arg\max_{v_i \in T} c_i$

$X := X \cdot x$

endwhile

return *X*.

Энергия сети и функция Ляпунова. Энергия нейронной сети Хопфилда задаётся стандартным выражением:

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

В [12] 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 $\hat{x}$ – устойчивое состояние (точечный аттрактор). Из этого следует, что система стремится к устойчивому состоянию.

Функция (4) является квадратичной формой, но, несмотря на это, у неё может быть несколько локальных минимумов, т. к. модель дискретная (в непрерывном случае есть ещё интеграл) и значения x ограничены.

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

Квадратичная функция штрафа. Пусть $h(x) = C_p (x-1)^2$, где C_p – коэффициент штрафа. Это неудачная функция, т. к. мы штрафуем двукратное покрытие так же, как и непокрытие, однако, квадратичная функция широко используется в статьях, т. к. позволяет получить удобные аналитические выражения.

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

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

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

Заметим, что $w_{ij} = w_{ji}$, но $w_{ii} \neq 0$, а это является необходимым условием устойчивости работы сети [21] (в случае $w_{ii} \neq 0$ возможна осцилляция сети). Чтобы это компенсировать, можно изменить условие остановки процесса оптимизации с учётом возможных осцилляций.

Разрывная функция штрафа. Пусть теперь $h(x) = C_p[x = 0]$, где C_p – коэффициент штрафа.

Проблема с данным функционалом заключается в том, что он разрывен и напрямую получить выражение для параметров модели не удаётся.

Эту проблему можно решить двумя способами:

1. Использовать непрерывную аппроксимацию, например,

$$\hat{h}(x) = \frac{C_p}{1 + \exp(\alpha x)},$$

а для получения выражений параметров сети раскладывать получившийся целевой функционал в ряд Тейлора в текущей точке. Поскольку текущая точка постоянно меняется, разложение требуется постоянно пересчитывать, что негативно влияет на быстродействие. К тому же, разложение корректно в небольшой окрестности текущей точки, а мы движемся 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 расчитывать целевые функционалы для разных состояний рассматриваемого нейрона и разность между ними $\varDelta Q_i = Q(x_i = 1) - Q(x_i = 0)$. С вероятностью

$$P[x_i = 1] = \frac{1}{1 + \exp(-\varDelta Q_i / T)},$$

где T – так называемая «температура» сети, будем переводить нейрон в возбуждённое состояние. Закон изменения температуры может быть различным, например, монотонно убывающим по гиперболическому закону, либо пилообразно убывающим с чередованием участков монотонного убывания и резкого возрастания.

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

3 Результаты вычислительного эксперимента

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

Так как результат применения алгоритма зависит от случайного начального состояния нейронной сети, для каждой задачи проводилось 5 запусков и результаты усреднялись. Результаты расчётов приведены в таблице 1.

Таблица 1. Результаты вычислительного эксперимента для сравнения алгоритмов

Номер задачи	Оптимальное решение	Решение жадным алгоритмом	Решение нейронной сетью	Номер задачи	Оптимальное решение	Решение жадным алгоритмом	Решение 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

Номер задачи	Оптимальное решение	Решение жадным алгоритмом	Решение нейронной сетью	Номер задачи	Оптимальное решение	Решение жадным алгоритмом	Решение нейронной сетью
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				

Для задач классов E-H оптимальные решения неизвестны из-за большой размерности задач. Для оценки были использованы предполагаемые решения, указанные в [11]. Результаты расчётов приведены в таблице 2.

Таблица 2. Результаты вычислительного эксперимента для задач большой размерности

Номер задачи	Предполагаемое решение	Решение жадным алгоритмом	Решение нейронной сетью	Номер задачи	Предполагаемое решение	Решение жадным алгоритмом	Решение 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

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 расчётов – 14 секунд на задачу.

4 Построение иерархического представления

Используя решение SCP можно построить 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 слоёв нет

$V_c: = V$

$E_c := E$

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

S := SolveSCP(V_c, E_c) – S является подмножеством V_c

$A := \{\}$

$\mathrm{V_c} := \{\}$

$\mathrm{E_c} := \{\}$

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

for v **in** S:

$A_i := v$ U {смежные с v вершины}

$A := A$ U A_i

$V_c := V_c$ U v

head $A_i := v$

endfor

for a **in** A, **for** b **in** A \ a: – создаём рёбра

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

$E_c := E_c$ U (head a ,head b)

endif

endfor

$N := N$ U ((V_c, E_c), A) – добавляем слой

endwhile

return N

конец

Пример работы алгоритма 2 представлен на рис. 1.

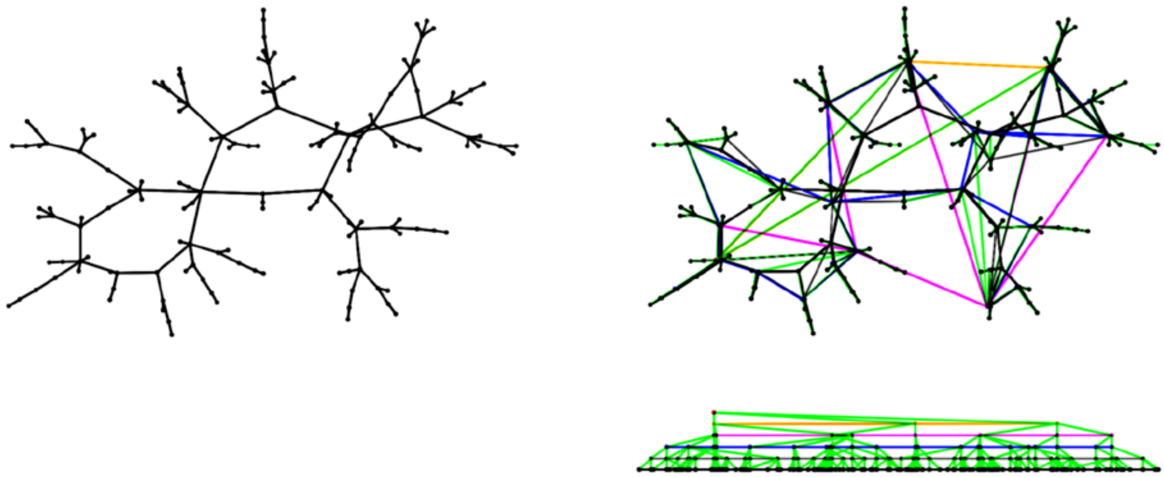

Рисунок 1. Схема работы алгоритма 2

Выводы

В данной работе рассмотрено решение задачи о покрытия 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. Abe S., Kawakami J., Hirasawa K. Solving inequality constrained combinatorial optimization problems by the Hopfield neural network // Neural Networks. 1992. Vol. 5, no. 4. P. 663-670.
2. George D., Jaros B. The HTM learning algorithms. Technical report. Numenta Inc., 2007.
3. Scarselli F. A short description of the graph neural network toolbox. 2011. Available at: http://www.dii.unisi.it/~franco/Research/GNN_DOC.pdf , accessed 01.08.2013.
4. Scarselli F., Gori M., Ah Chung Tsoi, Hagenbuchner M., Monfardini G. The graph neural Network model // IEEE Transactions on Neural Networks. 2009. Vol. 20, iss. 1. P. 61-80. DOI: 10.1109/TNN.2008.2005605
5. 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.
6. Hopfield J., Tank D. “Neural” computation of decisions in optimization problems // Biological Cybernetics. 1985. Vol. 52. P. 141-152.
7. Smith K. 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.
8. Le Gall A., Zissimopoulos V. Extended Hopfield models for combinatorial optimization // IEEE Transactions on Neural Networks. 1999. Vol. 10, iss. 1. P. 72-80. DOI: 10.1109/72.737495

9. Karp R. Reducibility among combinatorial problems // Complexity of Computer Computations, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y. 1972 / R. E. Miller, J. W. Thatcher, eds. New York: Plenum, 1972. P. 85-103.
10. Еремеев А.В., Заозерская Л.А., Колоколов А.А. Задача о покрытии множества: сложность, алгоритмы, экспериментальные исследования // Дискретный анализ и исследование операций. Сер. 2. 2000. Т. 7, № 2. С. 22-46.
11. Нгуен Минь Ханг. Применение генетического алгоритма для задачи нахождения покрытия множества // Труды ИСА РАН. 2008. № 33. С. 206-219.
12. Хайкин С. Нейронные сети: полный курс : пер. с англ. М.: Издательский дом «Вильямс», 2006. 1104 с.
13. Balas E., Carrera M. A dynamic subgradient-based branch and bound procedure for set covering // Oper. Res. 1996. Vol. 44, no. 6. P. 875-890.
14. Beasley J. A Lagrangian heuristic for set-covering problems // Naval Res. Logist. 1990. Vol. 37, no. 1. P. 151-164.
15. Caprara A., Fischetti M., Toth P. Algorithms for the set covering problem. DEIS — Operations Research Group. Technical Rep. No. OR-98-3. 1998.
16. Еремеев А.В. Генетический алгоритм для задачи о покрытии // Дискретный анализ и исследование операций. Сер. 2. 2000. Т. 7, № 1. С. 47-60.
17. 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. Berlin: Springer, 1996, pp. 3-20. (Ser. Lecture Notes in Comput. Sci.; vol. 1063). DOI: 10.1007/3-540-61108-8_47
18. Beasley J., Chu P. A genetic algorithm for the set covering problem // European J. Oper. Res. 1996. Vol. 94, no. 2. P. 394-404.
19. Ramalhinho H., Pinto J., Portugal R. Metaheuristics for the bus-driver scheduling problem. Barcelona, Spain, Univ. Pompeu Fabra. 1998. (Economic Working Papers Series; no. 304). Available at: http://www.econ.upf.edu/docs/papers/downloads/304.pdf , accessed 01.08.2013.
20. Alexandrov D., Kochetov Yu. Behavior of the ant colony algorithm for the set covering problem // Operations Research Proceedings 1999. Berlin: Springer, 2000. P. 255-260. (Ser. Operations Research Proceedings; vol. 1999). DOI: 10.1007/978-3-642-58300-1_38 .
21. Cohen M., Grossberg S. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks // IEEE Transactions on Systems, Man, and Cybernetics. 1983. Vol. SMC-13, iss. 5. P. 815-826. DOI: 10.1109/TSMC.1983.6313075
22. Chvatal V. Greedy heuristic for the set-covering problem // Mathematics of Operations Research. 1979. Vol. 4, no. 3. P. 233-235.
23. Пролубников А.В. Оценка приближенного решения алгоритмом Хватала задачи о покрытии // Математические структуры и моделирование: сб. науч. тр. Вып. 2. Омск: Изд-во ОмГУ, 2002. С. 58-61.

SCIENTIFIC PERIODICAL OF THE BAUMAN MSTU

SCIENCE and EDUCATION

EL № FS77 - 48211. №0421200025. ISSN 1994-0408

electronic scientific and technical journal

Hierarchical representation of a computer network on the basis of a Hopfield neural network

 # 09, September 2013

DOI: 10.7463/0913.0630141

Basarab M.A., Vel'c S.V.

Bauman Moscow State Technical University, 105005, Moscow, Russian Federation

bmic@mail.ru

svelts@rambler.ru

The problem of hierarchical representation of a computer network for carrying out its multiscale analysis was considered. An original problem was reduced to the known set cover problem (SCP). Since this problem is NP-difficult, approximate heuristic techniques were useful for its solving. We proposed an algorithm based on the recurrent Hopfield neural network and its combination with the simulated annealing algorithm (so-called Boltzmann machine). A greedy algorithm was used for a construction of hierarchy levels. This approach was evaluated with a use of the test problems from OR-Library. Proposed approach could be used in the systems of traffic analysis and anomaly detection in computer networks.

Publications with keywords: set cover problem, hierarchical representation, computer network, neural optimization, Hopfield network, Boltzmann machine

Publications with words: set cover problem, hierarchical representation, computer network, neural optimization, Hopfield network, Boltzmann machine

References

1. Abe S., Kawakami J., Hirasawa K. Solving inequality constrained combinatorial optimization problems by the Hopfield neural network. *Neural Networks*, 1992, vol. 5, no. 4, pp. 663-670.
2. George D., Jaros B. *The HTM learning algorithms.* Technical report. Numenta Inc., 2007.
3. Scarselli F. *A short description of the graph neural network toolbox*. 2011. Available at: http://www.dii.unisi.it/~franco/Research/GNN_DOC.pdf , accessed 01.08.2013.
4. Scarselli F., Gori M., Ah Chung Tsoi, Hagenbuchner M., Monfardini G. The graph neural Network model. *IEEE Transactions on Neural Networks*, 2009, vol. 20, no. 1, pp. 61-80. DOI: 10.1109/TNN.2008.2005605

5. Grossman T., Wool A. Computational experience with approximation algorithms for the set covering problem. *European J. Oper. Res*., 1997, vol. 101, no. 1, pp. 81-92.
6. Hopfield J., Tank D. “Neural” computation of decisions in optimization problems. *Biological Cybernetics*, 1985, vol. 52, pp. 141-152.
7. Smith K. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing*, 1999, vol. 11, no. 1, pp. 15-34.
8. Le Gall A., Zissimopoulos V. Extended Hopfield models for combinatorial optimization. *IEEE Transactions on Neural Networks*, 1999, vol. 10, iss. 1, pp. 72-80. DOI: 10.1109/72.737495
9. Karp R. Reducibility among combinatorial problems. In: Miller R.E, Thatcher *J.M., eds. Complexity of Computer Computations, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972*. New York, Plenum, 1972, pp. 85-103.
10. Eremeev A.V., Zaozerskaya L.A., Kolokolov A.A. Zadacha o pokrytii mnozhestva: slozhnost', algoritmy, eksperimental'nye issledovaniya [Set covering problem: complexity, algorithms, and experimental studies]. *Diskretnyy analiz i issledovanie operatsiy. Ser. 2*, 2000, vol. 7, no 2, pp. 22-46.
11. Nguen Min' Khang. Primenenie geneticheskogo algoritma dlya zadachi nakhozhdeniya pokrytiya mnozhestva [Application of genetic algorithm to the problem of finding set covering]. *Trudy ISA RAN*, 2008, no. 33, pp. 206-219.
12. Haykin S. *Neural Networks: A Comprehensive Foundation*. 2nd ed. Prentice Hall, 1999. 823 p. (Russ. ed.: Haykin S. *Neironnye seti: polnyi kurs*. Moscow, Publishing House “Vil'iams”, 2006. 1104 p.).
13. Balas E., Carrera M. A dynamic subgradient-based branch and bound procedure for set covering. *Oper. Res*., 1996, vol. 44, no. 6, pp. 875-890.
14. Beasley J. A Lagrangian heuristic for set-covering problems. *Naval Res. Logist*., 1990, vol. 37, no. 1, pp. 151-164.
15. Caprara A., Fischetti M., Toth P. *Algorithms for the set covering problem. DEIS — Operations Research Group.* Technical Rep. No. OR-98-3, 1998.
16. Eremeev A.V. Geneticheskiy algoritm dlya zadachi o pokrytii [Genetic algorithm for covering problem]. *Diskretnyy analiz i issledovanie operatsiy. Ser. 2*, 2000, vol. 7, no. 1, pp. 47-60.
17. Back Th., Schiiltz M., Khuri S. A comparative study of a penalty function, a repair heuristic, and stochastic operators with the set-covering problem. In: *Artificial Evolution*. Berlin, Springer, 1996, pp. 3-20. (Ser. *Lecture Notes in Comput. Sci*.; vol. 1063). DOI: 10.1007/3-540-61108-8_47
18. Beasley J., Chu P. A genetic algorithm for the set covering problem. *European J. Oper. Res*., 1996, vol. 94, no. 2, pp. 394-404.

19. Ramalhinho H., Pinto J., Portugal R. *Metaheuristics for the bus-driver scheduling problem*. Barcelona, Spain, Univ. Pompeu Fabra. 1998. (*Economic Working Papers Series*; no. 304). Available at: http://www.econ.upf.edu/docs/papers/downloads/304.pdf , accessed 01.08.2013.

20. Alexandrov D., Kochetov Yu. Behavior of the ant colony algorithm for the set covering problem. In: *Operations Research Proceedings 1999*. Berlin, Springer, 2000, pp. 255-260. (Ser. *Operations Research Proceedings*; vol. 1999). DOI: 10.1007/978-3-642-58300-1_38

21. Cohen M., Grossberg S. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 1983, vol. SMC-13, iss. 5, pp. 815-826. DOI: 10.1109/TSMC.1983.6313075

22. Chvatal V. Greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 1979, vol. 4, no. 3, pp. 233-235.

23. Prolubnikov A.V. Otsenka priblizhennogo resheniya algoritmom Khvatala zadachi o pokrytii [Estimation of approximative solution of covering problems with the use of Chvátal algorithm]. *Matematicheskie struktury i modelirovanie: sb. nauch. tr.* [Mathematical structures and modeling: collection of scientific papers]. Iss. 2. Omsk, OmSU Publ., 2002, pp. 58-61.