УДК 004.02

ПРИМЕНЕНИЕ РАСТУЩЕЙ НЕЙРОННОЙ СЕТИ ДЛЯ РЕШЕНИЯ КВАДРАТИЧНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

И. Л. Каширина

Воронежский государственный университет

В статье предлагается и исследуется метод решения квадратичной задачи о назначениях с помощью растущей нейронной сети. В качестве базовой постановки используется задача о размещении производственных объектов.

1. ВВЕДЕНИЕ

Квадратичная задача о назначениях остается одной из самых больших проблем в комбинаторной оптимизации. До сих пор считается нетривиальной задачей отыскать оптимальное назначение при размерности более 30×30 переменных, тогда как для близкой по формальной постановке задачи коммивояжера этот порог уже приблизительно равен 1000×1000. Так как в реальных постановках размерность квадратичной задачи о назначениях существенно превышает число 30, на практике используются приближенные алгоритмы, среди которых главенствующее положение занимают генетические методы. Однако для задачи коммивояжера хорошо себя зарекомендовал подход, основанный на оптимизации с применением растущих нейросетей [1]. В данной статье предлагается и исследуется развитие этого метода на случай квадратичной задачи о назначениях.

2. РАСТУЩИЕ НЕЙРОННЫЕ СЕТИ

Эффективное практическое применение нейронных сетей для оптимизации возможно, если вычислительные затраты у соответствующей модели сети растут не слишком быстро с ростом размерности задачи. Так, для задачи коммивояжера размерности n временные затраты при эмуляции сети Хопфилда растут как n^6, т.к. каждый из n^2 нейронов имеет порядка n^4 синаптических весов. Фритцке и Вильке предложили нейросетевую систему, в которой с ростом размерности задачи затраты растут лишь линейно[2]. Предложенная ими модель относится к классу растущих нейронных сетей. Такие сети настраивают свою структуру в соответствии с требованиями решаемой задачи. Они стартуют с очень простых и небольших структур, которые разрастаются и усложняются по мере необходимости.

Растущие нейронные сети поддерживают все функции, присущие обычным нейронным сетям и интеллектуальным системам. Вместе с тем эти сети приобретает повышенную семантическую ясность, так как здесь имеет место не просто построение сети путем размещения смысловых структур в выбранной архитектуре нейронов, а, собственно, создание самой этой среды, как эквивалента среды памяти.

3. РАСТУЩАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ЗАДАЧИ КОМИВОЯЖЕРА

В большинстве нейросетевых методов решения задачи коммивояжера, требуется задание декартовых координат каждого города на плоскости (такая постановка называется геометрической). Исходя из этих координат, можно составить матрицу расстояний (в то время как далеко не из каждой матрицы расстояний можно вычислить относительные координаты городов). Растущая клеточная структура для геометрической задачи коммивояжера изначально представляет собой кольцо из трех нейронов[3]. Каждый нейрон характеризуется двумерным вектором $\mathbf{w}_i$, определяющим его положение на плоскости. Каждому нейрону в кольце приписывается также своя величина погрешности ε_i, которая вначале полагается равной нулю. Дальнейшая последовательность действий включает две следующие основные элементарные операции: смещение и добавление нового нейрона.

Смещение происходит следующим образом: выбирается случайный город $\boldsymbol{c}$; определяется нейрон-победитель $\mathbf{w}_*$, ближайший к этому городу; положение нейрона $\mathbf{w}_*$ и его двух ближайших в кольце соседей смещается в сторону города $\boldsymbol{c}$ на определенную долю расстояния до него.

Добавление нового нейрона осуществляется

после нескольких циклов смещений. К этому времени накапливается информация, на основании которой принимается решение о месте, в котором должен быть добавлен новый нейрон. Каждый раз, когда для случайно выбранного города $\mathbf{c}$ определяется ближайший к нему нейрон $\mathbf{w}_*$, локальная ошибка для последнего ε_* получает приращение $\|\mathbf{w}_* - \mathbf{c}\|$. Большое значение этой ошибки служит указанием на то, что соответствующий нейрон лежит в области, где отношение "число нейронов/число городов" (коэффициент связности сети) невелико. Именно в таких областях следует добавлять новые нейроны, поскольку для получения правильного осмысленного маршрута около каждого города должен находиться свой ближайший нейрон. Маршрут определяется путем перехода вдоль кольца к нейрону, являющимся ближайшим к некоторому городу. Очевидно, что решение задачи может быть найдено не ранее того, как число нейронов в кольце достигнет числа городов N.

Данный алгоритм можно модифицировать следующим способом: если некоторый нейрон несколько раз оказывается ближайшим для данного города, то этот нейрон "приклеивается" к данному пункту маршрута. Это означает, что он совмещается со своим городом и больше уже не двигается. Город же удаляется из списка городов, разыгрываемых на шаге *смещения*. Когда этот список становится пустым, процесс поиска маршрута заканчивается.

4. ПОСТАНОВКА КВАДРАТИЧНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Квадратичная задача о назначениях как математическая модель имеет весьма широкие применения. Одним из самых ее распространенных приложений является задача о размещении производственных объектов, которая формулируется следующим образом.

Пусть имеется n зданий, где могут размещаться цеха, и n производственных цехов. Обозначим: c_{kl} — расстояние между зданиями k и l, b_{ij} — объем продукции, транспортируемый из цеха i в цех j, где $i,j,k,l = 1, 2, ..., n$. Требуется найти размещение цехов по зданиям, при котором суммарный объем перевозок продукции был бы минимальным.

Переменные задачи вводятся следующим образом:

$$x_{ik} = \begin{cases} 1, & \text{если цех } i \text{ размещается в здании } k, \\ 0, & \text{в противном случае.} \end{cases}$$

Математическая постановка задачи имеет вид:

$$F(x_{11},\ldots x_{nn}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} b_{ij}c_{kl}x_{ik}x_{jl} \to \min$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i = 1,2,\ldots,n),$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (j = 1,2,\ldots,n),$$

$$x_{ij} \in \{0,1\}, \quad (i.j = 1,2,\ldots,n),$$

Для квадратичной задачи о назначениях известно не так много алгоритмов решения, как для задачи коммивояжера. Все существующие на данный момент точные методы близки по своей форме к полному перебору решений, и пока не могут применяться для задач размерности больше 30 переменных. Поэтому на сегодняшний день наиболее используемыми являются генетические алгоритмы, которые лучше всего себя зарекомендовали для решения квадратичных задач о назначениях, но, разумеется, они не могут гарантировать нахождение оптимального решения.

5. АЛГОРИТМ РЕШЕНИЯ КЗН С ПОМОЩЬЮ РАСТУЩЕЙ НЕЙРОННОЙ СЕТИ

Используем идеологию метода растущей нейронной сети для решения квадратичной задачи о назначениях. Предположим, что КЗН сформулирована как задача о размещении производственных объектов и представлена в геометрической форме, т.е. для всех зданий заранее известны их координаты на плоскости. В растущей сети для квадратичной задачи о назначениях нейроны будут обозначать цеха, каждый из которых нужно будет соотнести с одним из имеющихся зданий.

Растущая нейронная сеть для задачи коммивояжера имела кольцевую архитектуру, и каждый нейрон в сети имел связи ровно с двумя другими нейронами — соседними звеньями цепи. Сеть для решения квадратичной задачи о назначениях будет представлять собой связный граф, в котором вершины (нейроны) i и j будут связаны, если предусматриваются транспортировки продукции между i-м и j-м цехами (то есть элемент b_{ij} матрицы объемов перевозок B

не равен нулю). Соответственно, изменятся и законы перемещения нейронов: теперь притягиваться к случайно выбранному зданию будет ближайший нейрон-победитель, к которому, в свою очередь, будут притягиваться все нейроны, напрямую с ним связанные, но сила этого притяжения будет пропорциональна объему транспортировок продукции между их цехом и цехом нейрона-победителя. Этим достигается близкое взаимное расположение цехов с большим объемом транспортировок продукции.

С учетом этих особенностей алгоритм решения квадратичной задачи имеет следующий вид:

1. Создаем на плоскости с заранее расставленными зданиями три нейрона (в случайных координатах), обозначающих три случайных цеха.

2. Присваиваем каждому нейрону нулевую ошибку: $\varepsilon_i = 0$, $i = 1..3$.

3. По числу нейронов на плоскости повторяем процедуру ***смещения***:

1) Выбираем случайное здание.

2) Определяем ближайший к этому зданию нейрон.

3) Значение локальной ошибки данного нейрона ε_i увеличиваем на величину, пропорциональную расстоянию от него до выбранного здания.

4) Смещаем этот нейрон в сторону здания, а все связанные с ним нейроны — в сторону выбранного нейрона, пропорционально объемам транспортировок продукции между ними.

4. Если число нейронов достигло числа зданий, переходим на шаг 5. Иначе запускаем процедуру ***добавления*** нейрона:

1) Определяем «наихудшее» звено в сети, связывающее два нейрона i и j, для которых суммарная ошибка $\varepsilon_i + \varepsilon_j$ максимальна.

2) Определяем среди незадействованных цехов такой, для которого суммарный объем перевозок продукции к цехам i и j максимален.

3) Вставляем между нейронами i и j новый нейрон k, соответствующий выбранному цеху. Его ошибка полагается равной $\varepsilon_k = (\varepsilon_i + \varepsilon_j)/3$, при этом значения ошибок для нейронов i и j уменьшаются на 1/3, и суммарная ошибка остается неизменной. Возвращаемся на шаг 3.

5. Если каждому зданию соответствует единственный ближайший нейрон, то получено решение. Иначе повторяем процедуру смещения до тех пор, пока это соответствие не будет достигнуто.

Замечание. Сила притяжения нейронов к зданиям на протяжении всей работы алгоритма остается постоянной (5—10 % расстояния), в то время как сила взаимного притяжения цехов медленно уменьшается до нуля. При этом цеха распределяются по зданиям, а их взаимное расположение стремится к наиболее оптимальному.

Практические испытания выявили следующий недочет разработанного метода: иногда возникают ситуации, когда некоторые нейроны остаются в стороне и не выбираются ни одним из зданий, в то время как другие одновременно оказываются ближайшими сразу к целой группе зданий, в результате чего алгоритм не сходится, так как каждому зданию не удается поставить в соответствие один единственный нейрон. Для избежания таких ситуаций модифицируем представленный алгоритм. Для каждого нейрона введем еще четыре новых входных характеристики:

1) g_i — номер здания, которое последним выбрало данный нейрон как ближайший;

2) k_i — количество выборов данного нейрона зданием g_i подряд;

3) l_i — количество выборов данного нейрона предыдущим зданием;

4) n_i — количество итераций, прошедших с момента, когда данный нейрон был выбран и сдвинут.

На основании значений этих переменных введем следующую классификацию нейронов-цехов: нейрон — *актуальный*, если $n_i < n$; то есть актуальный нейрон — тот, который выбирался в течении последних n итераций; нейрон — *определившийся*, если $l_i > 4$ либо $k_i > 4$. Если нейрон является ближайшим сразу к нескольким зданиям, то он будет выбираться каждым из этих зданий, и эти переменные будут постоянно обнуляться. Определившиеся нейроны "приклеиваются" к своим зданиям, и эти здания удаляются из списка разыгрываемых в процедуре смещения. Таким образом, нейрон, который является ближайшим сразу к нескольким зданиям, будет актуальным, но неопределившимся. В ходе работы алгоритма обычные итерации с некоторой периодичностью будут заменяться проверками на наличие неактуальных нейронов, которые при обнаружении будут сдвигаться в сторону не определившихся нейронов.

Модифицированный таким образом алгоритм сходится в 100 % случаев.

6. ТЕСТИРОВАНИЕ АЛГОРИТМА

Программа, решающая квадратичную задачу о назначениях, написана в среде Delphi 7. Тестирование программы проводилось следующим образом. Для задач с размерностью не более 25-ти методом полного перебора отыскивалось точное решение, которое сравнивалось с результатом работы растущей нейронной сети. Среднее отношение найденного результата к оптимальному составило 1,18 (на 50 случайных примерах). Для задач большой размерности (от 26 до 100), ввиду невозможности получения точного решения, результат сравнивался с лучшим из N случайно выбранных назначений (N — размерность задачи) . В данном случае среднее отношение результатов составило 0,518 (как правило, решение, найденное методом растущей нейронной сети было существенно лучше)

7. ЗАКЛЮЧЕНИЕ

Результатом данного исследования явилась разработка нового метода решения квадратичной задачи о назначениях, способного конкурировать по эффективности с генетическими алгоритмами, традиционно признающимися лучшим приближенным решением оптимизационных задач большой размерности.

ЛИТЕРАТУРА

1. *Ященко В.А.* Рецепторно-эффекторные нейроподобные растущие сети эффективное средство моделирования интеллекта. I, II // Кибернетика и сист. анализ, № 4, 1995. С. 54—62, № 5, 1995. С. 94—102.

2. *Рабинович З.Л., Ященко В.А.* Подход к моделированию мыслительных процессов на основе нейроподобных растущих сетей // Кибернетика и сист. анализ № 5, 1996. С. 3—20.

3. *Fritzke B., Wilke P.* Flexmap — A Neural Network For The Traveling Salesman Problem With Linear Time And Space Complexity, Proc. of IJCNN, Singapore (1991)

Статья принята к опубликованию 25 декабря 2006 г.