О.П. МОСАЛОВ
Московский физико-технический институт
olegmos_@mail.ru

МОДЕЛЬ ЭВОЛЮЦИИ СИСТЕМЫ АГЕНТОВ-БРОКЕРОВ*

Аннотация

Предлагается модель Интернет-системы, осуществляющей принятие решений при игре на бирже. Система состоит из нейросетевых агентов, образующих подпопуляции на узлах Интернета. Оптимизация многоагентной системы осуществляется в результате: 1) индивидуального обучения агентов путем настройки нейронной сети методом градиентного спуска, 2) эволюции популяции агентов, 3) коммуникации между агентами, 4) перелетов агентов между узлами Интернета.

O.P. MOSALOV
Moscow Institute of Physics and Technologies
olegmos_@mail.ru

THE MODEL OF EVOLUTION OF A SYSTEM OF AGENTS-BROKERS

Abstract

A model of Internet-system making decision while speculate on the stock exchange has been proposed. The system consists of neuron-net agents composing sub-populations on Internet lobes. Optimization of the multiagent system is put into effect as a result of: 1) individual learning of agents by means of adjusting of neuron network using gradient-descent method, 2) evolution of the agent population, 3) communications between agents, 4) flights of agents between Internet lobes.

Введение

Одно из перспективных направлений развития Интернет-систем – создание интеллектуальных систем принятия решений на базе информации, полученной в Интернете. Так как объемы информации в Интернете так велики, что их практически невозможно обработать

* Работа выполнена при финансовой поддержке РФФИ (проект № 02-07-90197) и ОИТВС РАН (проект ОИТВС № 2.1).

человеку, то необходимы компьютерные системы интеллектуальной обработки информации, обеспечивающие поддержку принятия решений. Одно из направлений создания интеллектуальных Интернет-систем – разработка многоагентных систем, позволяющих производить распределенную обработку информации.

В настоящей работе предлагается модель системы искусственных организмов (агентов), которые могут принимать решения о покупке-продаже акций, играя на бирже. Принятие решение осуществляется с помощью метода обучения с подкреплением [1], при этом оценка функции качества действия (action value function), которая используется в этом методе, производится аппроксимирующей нейронной сетью. Нейронная сеть двухслойная, в процессе обучения веса синапсов нейронной сети оптимизируются градиентным методом, аналогично тому, как это делается в обычном методе обратного распространения ошибки [2]. Агент имеет свой жизненный ресурс, который увеличивается либо уменьшается в зависимости от игры на бирже. Цель каждого агента – максимизировать прирост ресурса, получаемый за достаточно длительный период времени.

Общие предположения модели

1) Есть агент, который располагает некоторым количеством ресурсов двух типов: виртуальными деньгами M и некоторым числом акций N_A.

2) Внешняя среда определяется временным рядом $X(t)$, $t = 0,1,2,...$, который задает курс акций на бирже (строим модель в дискретном времени).

3) Агент стремится увеличить свой суммарный ресурс $R(t) = M(t) + N_A(t) X(t)$, продавая и покупая акции. Агент имеет нейронную сеть, которую он использует для выбора действия (покупка, продажа, ожидание более выгодной ситуации).

4) Каждый такт агент тратит на поддержание существования небольшое количество денег ΔM_L.

5) Агент действует по схеме день-вечер. Сначала (днем) он играет на бирже, выбирая при помощи нейронной сети одно из трех действий (ожидание, покупка, продажа). При совершении сделки затраты составляют ΔM_J, кроме того, ресурс агента меняется в соответствии с изменением количества и стоимости его акций. Затем (вечером) агент, в случае, если у него осталось мало акций или денег, совершает дополнительную бесплатную сделку (затраты на нее равны 0), для конвертирования одного вида ресурса в другой.

6) При обращении хотя бы одного из двух типов ресурса в ноль агент погибает.
7) Если количество виртуальных денег агента превысит определенный предел M_{mate}, то такой агент становится половозрелым.
8) Два половозрелых агента на одном узле рождают нового агента.
9) Структура и начальные (полученные при рождении) веса синапсов нейронной сети составляют геном агента.
10) При рождении нового агента его геном формируется из геномов родителей в результате рекомбинации и мутаций генов.
11) Половозрелые агенты с определенной вероятностью посылают сигналы на другие узлы Интернета о своей готовности дать потомство. Если агент половозрелый и получает достаточно много таких сигналов из других узлов, то он перелетает в какой-либо из узлов.

Схема действий агента днем (при игре на бирже)

Агент имеет $M(t)$ денег и $N_A(t)$ акций, где t - номер временного такта. Общий ресурс агента равен

$$R(t) = M(t) + A(t), \qquad (1)$$

где $A(t) = N_A(t)\, X(t)$.

Цель агента – максимизация общего ресурса. Изменение общего ресурса равно:

$$\Delta R(t) = \Delta M(t) + \Delta A(t). \qquad (2)$$

Изменение $\Delta A(t)$ равно:

$$\Delta A(t) = N_A(t)\, X(t) - N_A(t\text{-}1)\, X(t\text{-}1) =$$

$$N_A(t\text{-}1)\, \Delta X(t) + \Delta N_A(t)\, X(t), \qquad (3)$$

где $\Delta N_A(t) = N_A(t) - N_A(t\text{-}1)$, $\Delta X(t) = X(t) - X(t\text{-}1)$.

Затраты агента на жизнь равны ΔM_L, затраты на сделку равны ΔM_J.

В каждый такт времени агент выбирает одно из трех действий: ожидание, покупка или продажа акций. Выбор действия осуществляется при помощи двухслойной нейронной сети. На входы нейронной сети

подаются значения временного ряда $X(t)$ за последние N тактов; в выходном слое расположено три нейрона, соответствующих трем возможным действиям агента.

Значения на выходах сети рассматриваются как оценки ожидаемой прибыли (в долгосрочной перспективе) при соответствующих действиях агента.

Агент придерживается «ε-жадной» политики [1]: с вероятностью 1-ε он выбирает то действие, по которому предсказана наибольшая прибыль, и с вероятностью ε – случайным образом любое из действий.

Рассмотрим изменение общего ресурса агента $R(t)$, а также количества его денег $M(t)$ и числа акций $N_A(t)$ для всех три вариантов действий.

1) Ожидание:

$$\Delta M(t) = -\Delta M_L;$$
$$\Delta N_A(t) = 0;$$
$$\Delta A(t) = N_A(t\text{-}1)\,\Delta X(t).$$

$$\Delta R(t) = -\Delta M_L + N_A(t\text{-}1)\,\Delta X(t). \qquad (4)$$

2) Покупка:

$$\Delta M(t) = -\Delta M_L - \Delta M_J - X(t);$$
$$\Delta N_A(t) = 1;$$
$$\Delta A(t) = N_A(t\text{-}1)\,\Delta X(t) + X(t).$$

$$\Delta R(t) = -\Delta M_L - \Delta M_J + N_A(t\text{-}1)\,\Delta X(t). \qquad (5)$$

3) Продажа:

$$\Delta M(t) = -\Delta M_L - \Delta M_J + X(t);$$
$$\Delta N_A(t) = -1;$$
$$\Delta A(t) = N_A(t\text{-}1)\,\Delta X(t) - X(t).$$

$$\Delta R(t) = -\Delta M_L - \Delta M_J + N_A(t\text{-}1)\,\Delta X(t). \qquad (6)$$

Естественно считать, что при $\Delta R(t) > 0$ агент получает поощрение за свои действия, а при $\Delta R(t) < 0$ – наказание.

Схема действий агента вечером (принудительное конвертирование)

Так как агент оценивает свое состояние по общему ресурсу $R(t)$, то при малом количестве одного из ресурсов (денег либо акций), и достаточно большом количестве второго, естественно конвертировать часть большого ресурса в малый. Схема конвертирования излагается ниже.

Агент после игры на бирже оценивает свое состояние. Если ресурсов обоих видов достаточно ($M(t) > X(t)$ и $N_A(t) > 0$), то агент ничего не делает. Если же одного из ресурсов осталось мало или не осталось вообще, происходит принудительное конвертирование (покупка либо продажа одной акции) без затрат на саму сделку.

В случае, когда у агента нет акций, но есть достаточно большая сумма денег ($M(t) > X(t)$ и $N_A(t) = 0$), он конвертирует деньги в акции. При этом

$$\Delta M(t) = - X(t);$$
$$\Delta N_A(t) = 1. \qquad (7)$$

В случае, когда у агента денег меньше, чем текущий курс акций, но есть акции ($M(t) < X(t)$ и $N_A(t) > 0$), он конвертирует акции в деньги. При этом

$$\Delta M(t) = X(t);$$
$$\Delta N_A(t) = -1. \qquad (8)$$

Если по итогам временного такта (дня и вечера) количество денег или число акций стало равным или меньшим нуля, то такой агент погибает.

Схема обучения агента

Для построения схемы обучения используем подход работы [1]. Пусть в момент времени t агент совершает действие $a(t)$. Прогноз прибыли, оцениваемый с помощью аппроксимирующей нейронной сети для этого действия, обозначим $Q(\mathbf{S}(t), a(t))$. Вектор $\mathbf{S}(t)$ характеризует входную ситуацию, в нашем случае это значения временного ряда $X(t)$ за последние N тактов времени.

Обучение нейронной сети производится методом градиентного спуска. Согласно [1] при аппроксимации значений $Q(\mathbf{S}(t), a(t))$ с помощью вектора параметров $\boldsymbol{\theta}(t)$ этот вектор изменяется на каждом такте в соответствии с формулой:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \alpha\delta(t)\mathbf{e}(t), \qquad (9)$$

где $\delta(t)$ – ошибка временной разности, вычисляемая следующим образом (обоснование см. в [1]):

$$\delta(t) = \Delta R(t) + \gamma Q(\mathbf{S}(t+1), a(t+1)) - Q(\mathbf{S}(t), a(t)). \qquad (10)$$

$$\mathbf{e}(t) = \gamma\lambda\mathbf{e}(t-1) + \mathrm{grad}_\theta(Q_t(\mathbf{S}(t), a(t))), \qquad (11)$$
$$\mathbf{e}(0) = \mathbf{0}.$$

Здесь $\Delta R(t)$ – изменение ресурса в момент времени t, γ – дисконтный фактор, $0 < \gamma < 1$, $0 < \lambda < 1$. Формула (10) вытекает из требования максимизации суммарной награды $\Sigma_l \gamma^l \Delta R(t+l+1)$ с учетом дисконтного фактора, выражение (11) учитывает «след градиента» от предыдущих моментов времени.

В нашем случае компонентами вектора $\boldsymbol{\theta}(t)$ являются веса синапсов нейронной сети V_{ij} и W_{jk} скрытого и выходного слоя соответственно.

Значение выходов нейронной сети определяется следующим образом:

$$Q_k = f(\sum_j W_{jk} Y_j),\ Y_j = f(\sum_i V_{ij} X_i), \qquad (12)$$

где $f(u) = 1/(1 + \exp(-u))$ – функция активации нейрона. Далее считаем, что k – номер действия, выбираемого в данный момент времени.

Вычисляя частные производные Q_k по весам как производные сложной функции с учетом равенства $\partial f(u)/\partial u = f(1 - f)$, имеем (расчет сходен с расчетом производных ошибки нейронной сети по весам в методе обратного распространения ошибки [2]):

$$\partial Q_k/\partial W_{jk} = Y_j\, Q_k\, (1 - Q_k), \qquad (13)$$

$$\partial Q_k/\partial V_{ij} = W_{jk}\, X_i\, Y_j\, (1 - Y_j)\, Q_k\, (1 - Q_k). \qquad (14)$$

Таким образом, получаем формулы для изменения весов нейронной сети V_{ij} и W_{jk}:

$$\Delta V_{ij}(t) = \alpha\delta(t) e_{Vij}(t),\ e_{Vij}(t) = \gamma\lambda e_{Vij}(t-1) + \partial Q_k/\partial V_{ij}, \qquad (15)$$

$$\Delta W_{jk}(t) = \alpha\delta(t)e_{Wjk}(t),\ e_{Wjk}(t) = \gamma\lambda e_{Wjk}(t\text{-}1) + \partial Q_k/\partial W_{jk}. \qquad (16)$$

Формулы (15), (16) совместно с (10) и (13), (14) определяют процедуру обучения нейронной сети агента.

Общая схема управления, скрещивания, коммуникации и перелеты

Общая схема управления агентом представлена на рис. 1.

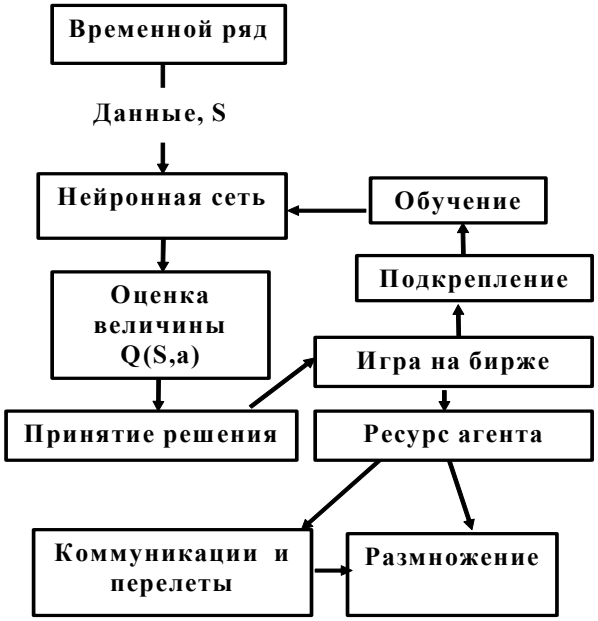

Рис. 1. Схема управления агентом.

Механизмы скрещивания, коммуникаций и перелетов агентов аналогичны описанным в [3].

Геном агента кодирует структуру и начальные веса синапсов нейронной сети. При скрещивании происходят рекомбинации хромосом, входящих в геномы родителей, мутации и формирование генома потомка.

Предполагаем, что каждый половозрелый агент в каждый такт времени t с вероятностью P_{com} подает на всю популяцию сигнал о том, что он

готов к скрещиванию. Другие половозрелые агенты оценивают общий уровень L таких сигналов, пришедших за последнее время, и в соответствии с этим уровнем могут «принимать решение» о перелете на другие узлы рассматриваемого модельного мира.

Эволюция всей популяции, связь с пользователем

Все агенты действуют синхронно. Т.е. время t для всех узлов одинаково. Если агент в данный такт времени скрещивается или перелетает, то он в этот такт времени не участвует в сделках.

Агенты играют на биржах виртуально, т.е. сами они не совершают сделок. Но на каждом узле пользователь может принимать решения в соответствии с рекомендациями агентов. Предполагаем, что пользователь следует рекомендациям того агента, у которого ресурс возрастает с максимальной скоростью.

Список литературы

1. Sutton R. and Barto A. *Reinforcement Learning: An Introduction*. – Cambridge: MIT Press, 1998.
   See also: http://www-anw.cs.umass.edu/~rich/book/the-book.html
2. Rumelhart D.E., Hinton G.E., Williams R.G. Learning representation by back-propagating error // *Nature*. 1986. V.323. N.6088.
3. Мосалов О.П., Бурцев М.С., Митин Н.А., Редько В.Г. Модель многоагентной интернет-системы, предназначенной для предсказания временных рядов // *V Всероссийская научно-техническая конференция "Нейроинформатика-2003"*. Сборник научных трудов. М.: МИФИ, 2003. Часть 1. С.177-183.