МОДЕЛЬ АГЕНТА-БРОКЕРА*

О. П. Мосалов
Московский физико-технический институт

В настоящей работе предлагается модель искусственного организма (агента), который может 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 представляет собой двухслойный перцептрон, 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 агента – максимизировать прирост ресурса, получаемый за достаточно длительный период времени.

Общие предположения модели

1) Есть агент, который располагает некоторым количеством жизненного ресурса (виртуальных денег) R 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 ресурс, 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 ресурса *mindeltaR*. При совершении сделки затраты составляют DR , кроме того, ресурс агента меняется в соответствии с изменением количества и стоимости его акций.

Схема принятия решения агентом при игре на бирже

В каждый момент 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 3 нейрона, соответствующих трем возможным действиям агента.

Значения на выходах сети рассматриваются как оценки ожидаемой прибыли (в долгосрочной перспективе) при соответствующих действиях агента. Агент с вероятностью 1-ε выбирает то действие, по которому предсказана наибольшая прибыль, и с вероятностью ε – случайным образом любое из действий.

Рассмотрим изменение ресурса агента $R(t)$ и количества его акций $N_A(t)$ для всех три вариантов действий.

1) Ожидание:

$$\Delta R(t) = - mindeltaR + N_A(t)*delta\_Rpacket, \qquad (1)$$

где *delta_Rpacket* - изменение стоимости пакета акций: $delta\_Rpacket = X(t) - X(t\text{-}1)$, t - текущий момент времени, N_A не изменяется.

* Работа выполнена при финансовой поддержке РФФИ (проект № 02-07-90197)

2) Покупка:

$\Delta R(t) = -\ mindeltaR - DR + N_A\ (t)*delta\_Rpacket$,

$N_A\ (t+1) = N_A\ (t) + 1,$ (2)

где DR - затраты на сделку.

3) Продажа:

$\Delta R(t) = -\ mindeltaR - DR + N_A\ (t)*delta\_Rpacket$,

$N_A\ (t+1) = N_A\ (t) - 1.$ (3)

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

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:

$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \alpha\delta(t)\mathbf{e}(t),$ (4)

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

$\delta(t) = \Delta R(t+1) + \gamma Q(\mathbf{S}(t+1), a(t+1)) - Q(\mathbf{S}(t), a(t)),$ (5)

$\mathbf{e}(t) = \gamma\lambda\mathbf{e}(t-1) + \mathrm{grad}_{\boldsymbol{\theta}}(Q(\mathbf{S}(t), a(t))),$ (6)

$\mathbf{e}(0) = \mathbf{0}.$

Здесь $\Delta R(t)$ – изменение ресурса в момент времени t, γ – дисконтный фактор, $0 < \gamma < 1$, $0 < \lambda < 1$ Формула (5) вытекает из требования максимизации суммарной награды $\Sigma_l\ \gamma^l\ \Delta R(t+l+1)$ с учетом дисконтного фактора, выражение (6) 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:

$Q_k = f(\sum_j W_{jk}\ Y_j),\ Y_j = f(\sum_i V_{ij}\ X_i),$ (7)

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]):

$$\partial Q_k/\partial W_{jk} = Y_j\, Q_k\, (1 - Q_k), \qquad (8)$$

$$\partial Q_k/\partial V_{ij} = W_{jk}\, X_i\, Y_j\, (1 - Y_j)\, Q_k\, (1 - Q_k). \qquad (9)$$

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

$$\Delta V_{ij}(t) = \alpha\delta(t) e_{Vij}(t),\ e_{Vij}(t) = \gamma\lambda e_{Vij}(t\text{-}1) + \partial Q_k/\partial V_{ij}, \qquad (10)$$

$$\Delta W_{jk}(t) = \alpha\delta(t) e_{Wjk}(t),\ e_{Wjk}(t) = \gamma\lambda e_{Wjk}(t\text{-}1) + \partial Q_k/\partial W_{jk}. \qquad (11)$$

Формулы (10), (11) совместно с (5) и (8), (9) определяют процедуру обучения нейронной сети агента.

Обсуждение модели

Данная модель является развитием предыдущей модели агентов, играющих на бирже [3]. В отличие от [3], где предполагалось, что агент делает прогноз временного ряда, а затем на основе прогноза и простых эвристических правил принимает решение о покупке-продаже, здесь схема обучения агента построена более естественно: агенту не «навязываются» эвристические правила, он самостоятельно принимает решение только лишь на основе входной информации. Более того, следует подчеркнуть, что данная схема обучения является достаточно универсальной. Она применима для всех тех случаев, когда есть входной вектор $\mathbf{S}(t)$, характеризующий внешнюю среду, каждый такт времени нужно выбрать одно из альтернативных действий и стремиться максимизировать суммарную награду, получаемую за длительный интервал времени.

Литература:

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. PP. 533-536.
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.