



УДК 004.056

*Ляхов Александр Федорович,
Тришин Илья Михайлович*

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ИГРОКА В ИНТЕЛЛЕКТУАЛЬНОЙ КАРТОЧНОЙ ИГРЕ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ

Аннотация

В работе создана и описана искусственная нейронная сеть для выбора решения в неопределённой игровой ситуации, в классической карточной игре в «Дурака».

Проведённые исследования показывают, что принимаемые во время игры решения человеком, основанные на чувственном уровне понимания игровой ситуации, могут быть просчитаны количественно. Гибкость структуры нейронной сети позволяет в процессе обучения изменять структуру сети, то есть удалять избыточные нейроны или добавлять недостающие. Созданная программа может служить примером развития нового рефлексивно-ориентированного программирования, являющегося функциональным расширением парадигмы объектно-ориентированного программирования.

Ключевые слова: нейронная сеть, теория игр, нечёткая логика, обучение нейронной сети.

Одна из основных проблем XXI века – проблема создания искусственного интеллекта, то есть создание автоматических устройств, способных принимать решения в задачах с большой степенью неопределённости. В середине прошлого века исследователи обратились к математическому и программному моделированию процессов высшей нервной деятельности живых организмов. В это время была создана модель искусственного нейрона и нейронной сети.

Искусственная нейронная сеть представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Каждый процессор подобной сети имеет дело с сиг-

налами, которые он периодически получает, и сигналами, которые он посылает другим процессорам. Будучи соединёнными, в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры способны выполнять сложные задачи. Нейронные сети по своей сути, это программы с неопределёнными параметрами, которые определяются в ходе обучения. Технически это заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть учится выявлять сложные зависимости между эталонными входными данными и выходными, а также выполнять обобщение. В случае успешного обучения, сеть может получать верный результат на основании данных, которые отсутствовали в обучении.

© Ляхов А.Ф., Тришин И.М., 2013

В настоящее время с помощью нейронных сетей осуществляется автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем, организация ассоциативной памяти.

Создание различных видов нейронных сетей, исследование их работы позволяет моделировать интеллектуальную деятельность человека. Кроме того, нейронные сети являются достаточно гибким инструментом и легко поддаются различным модификациям и изменениям. Это позволяет создавать такие программы, структура которых может изменяться в процессе работы.

В данной работе создана и описана искусственная нейронная сеть для выбора решения в неопределённой игровой ситуации, в классической карточной игре в «Дурака»¹. Для обучения нейронной сети и исследования её свойств были созданы две программы, играющие по простым алгоритмам.

Проведённый анализ показал, что решения, принимаемые нейронной сетью во время игры, допускают общепринятые качественные логические обоснования.

МОДЕЛЬ ИГРЫ

В качестве модели игры взята классическая игра в «дурака». Основные правила игры такие: в игре участвует колода из 36 карт, игрокам раздается по 6 карт, одна карта открывается и она определяет козырную масть. Игра состоит из локальных партий (один игрок ходит, другой отвечает). После розыгрыша локальной партии игроки дополняют свои наборы карт до шести карт из колоды. Часть карт побитых игроком выходит из игры в отбой. Побеждает тот игрок, у которого в конце последней локальной партии не остается карт. Элемент случайности этой игры связан с раскладом карт и ошибками игроков.

В работе процесс развития игры рассматривается как последовательность локальных партий. Для количественного анализа игры вводится шкала оценки карт:

	Цена	
	для карт простой масти	для карт козырной масти
Шестерка	1	10
Семерка	2	11
Восьмерка	3	12
Девятка	4	13
Десятка	5	14
Валет	6	15
Дама	7	16
Король	8	17
Туз	9	18

и понятие средней стоимости одной карты на руках игрока:

$$m = \frac{\sum_{i=1}^k r_i}{k},$$

где k – число карт у игрока, r_i – стоимость карты в соответствии с введенной шкалой.

На рис. 1 приведена диаграмма изменения средней стоимости одной карты выигравшего игрока в течение игры. Можно видеть, что при игре, приводящей к выигрышу, средняя стоимость одной карты возрастает.

ЦЕЛЕВЫЕ ФУНКЦИИ ИГРЫ

Сама игра и процесс её развития определяется целевой функцией игры. Для игры в «Дурака» может быть предложен следующий вариант целевой функции игры: *каждый игрок стремится к тому, чтобы у него карты на руках закончились, а у противника остались.*

Игра состоит из последовательности локальных партий, но формализовать их связь при условии большого многообразия различных наборов карт не удаётся. Поэтому на данном этапе моделирования этой связью между партиями пренебрегаем.

В работе рассматривались два вида целевых функций локальных партий.

Первая целевая функция для локальной партии выглядит так: *отдать как можно больше карт к концу локальной партии.*

Для этой целевой функции был разработан следующий алгоритм.

¹ Данная работа является развитием работ [1, 2].

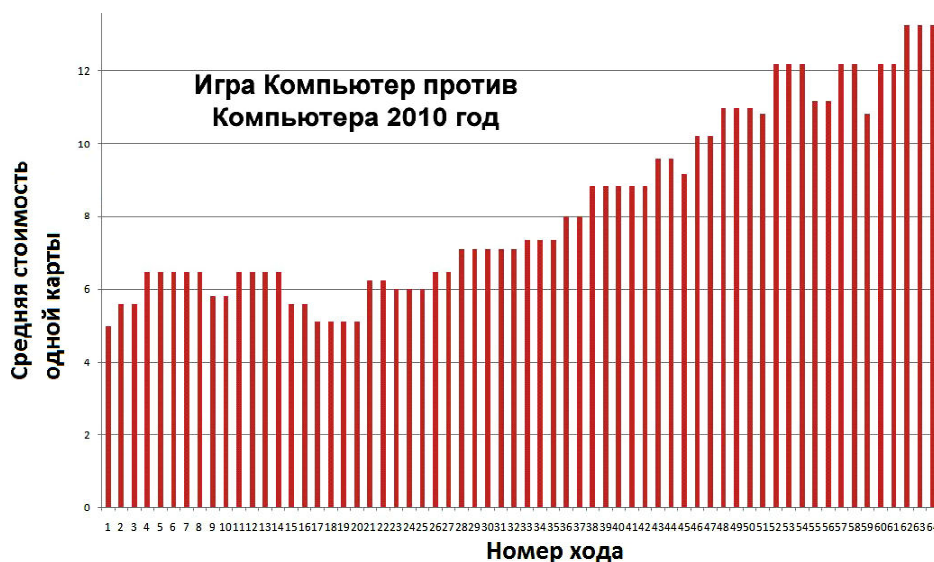


Рис. 1

Простой алгоритм принятия решения

- Во время розыгрыша локальной партии компьютер делает ход наименьшей картой.
- Если противник отбивается и есть возможность добавить, то компьютер добавляет.
- Если компьютер имеет возможность покрыть карту, он делает это наименьшей возможной картой.

Вторая целевая функция для локальной партии может быть сформулирована в следующем виде: *требуется получить наибольшую среднюю стоимость одной карты расклада к концу локальной партии.*

Усложненный алгоритм принятия игроком решения

Усложнение алгоритма осуществляется введением функции для определения средней стоимости одной карты среди своих карт, средней стоимости карты из колоды, а также средней стоимости одной карты противника. Все оценки средней стоимости выполняются с учетом известных своих карт, карт вышедших из игры (находящихся в отбое) и карт на руках у противника. В усложненном алгоритме компьютер для принятия решения проводит оценку изменения средней стоимости одной карты в своем раскладе при ходе и принимает решения таким

образом, чтобы эта средняя стоимость одной карты была максимальной.

Для реализации алгоритма были использованы следующие правила:

- Во время розыгрыша локальной партии компьютер делает ход наименьшей картой.
- При возможности добавления карты на стол учитывается изменение средней стоимости одной карты расклада на руках и делается выбор (добавлять или не добавлять) таким образом, чтобы средняя стоимость одной карты своего расклада была наибольшей. Карта добавляется, если её стоимость меньше средней стоимости карт на руках
- При покрытии карт делается выбор (крыть или не крыть) таким образом, чтобы получить наибольшее значение средней стоимости одной карты своего расклада после розыгрыша локальной партии

В программе *fool2.exe* были реализованы простой и усложненный алгоритмы моделирования поведения игроков. Вид её интерфейса показан на рис. 2. Описание программы и её интерфейса приведены в приложении.

Программа *fool2.exe* позволяет провести сравнения «простого» и «усложненного» алгоритмов. В ходе тестирования и розыгрышей большого количества партий было выяснено, что компьютер, принимающий решения по «усложненному» алгоритму вы-

игрывает в среднем на 40 % больше партий, чем компьютер, играющий по «простому алгоритму». Например, из 15676 сыгранных партий «усложненный» компьютер выиграл 69,18 %.

Реализованные алгоритмы использовались для обучения и исследования свойств нейронной сети, которая была включена в данную программу.

НЕЙРОННАЯ СЕТЬ

Исследуемая искусственная нейронная сеть была создана по принципу перцептрона Розенблата. Её обучение осуществлялось без учителя с Альфа-системой подкрепления [3].

Альфа-системой подкрепления называется система подкрепления, при которой веса всех активных связей сети cI , которые ведут к элементу rI , изменяются на одинаковую величину r , а веса неактивных связей не изменяются.

Все данные о картах, подаваемые на вход нейронной сети во время игры, были разбиты на четыре группы. Причём объём каждой группы был максимально возможным.

Данные, которые подавались на вход нейронной сети:

1. Карты на руках игрока (36 входов).
2. Карты, вышедшие из игры в отбой (36 входов).
3. Карты на столе (36 входов).
4. Карта, которой можно покрыть (36 входов).

Данные о каждой карте в процессе игры подаются по отдельному каналу. Если карта находится в группе, то по соответствующему каналу подаётся единица, если карта отсутствует в группе, то подаётся ноль.

Схема входов для каждой группы из 36-ти пунктов одинакова. На рис. 3 представлена схема для одной группы нейронной сети:

В карточной игре «Дурак» есть 9 номиналов карт (от шестерки до туза) и 4 масти, одна из которых в начале игры объявляется козырной. Поэтому на схеме для каждой карты некозырной масти учитывается не только её наличие, но и количество таких карт. При наличии той или иной карты в одном из 4-х мест, соответствующий элемент возбуждается. Элементы cI учитывают количество карт каждого номинала. Это значит, что $c1$ выдает +1, если сумма пришедших на него сигналов ≥ 1 (одна шестёрка), $c2$ – если сумма сигналов на нем ≥ 2 (две шестёрки), $c3$ – если сигналов на нем ≥ 3 (три шестерки),

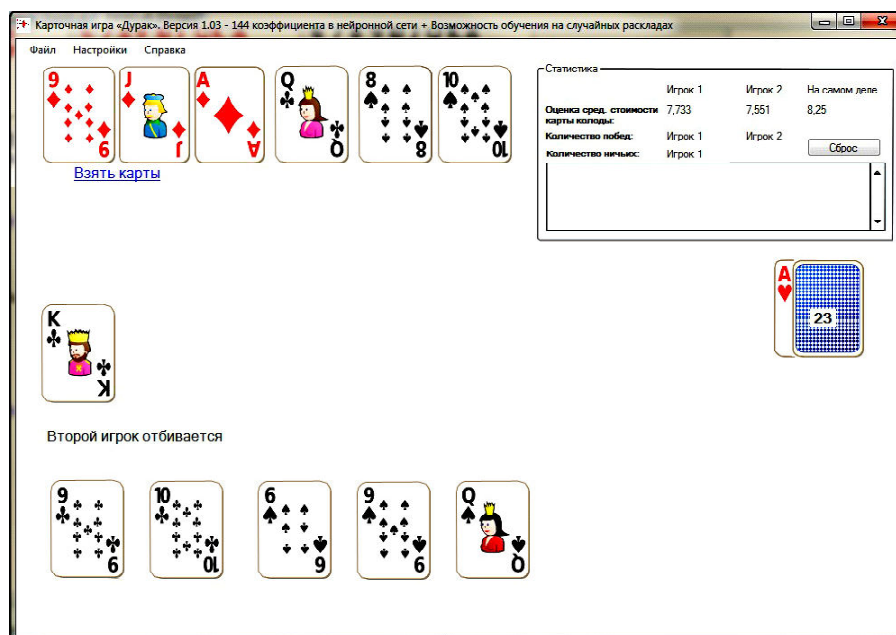


Рис. 2

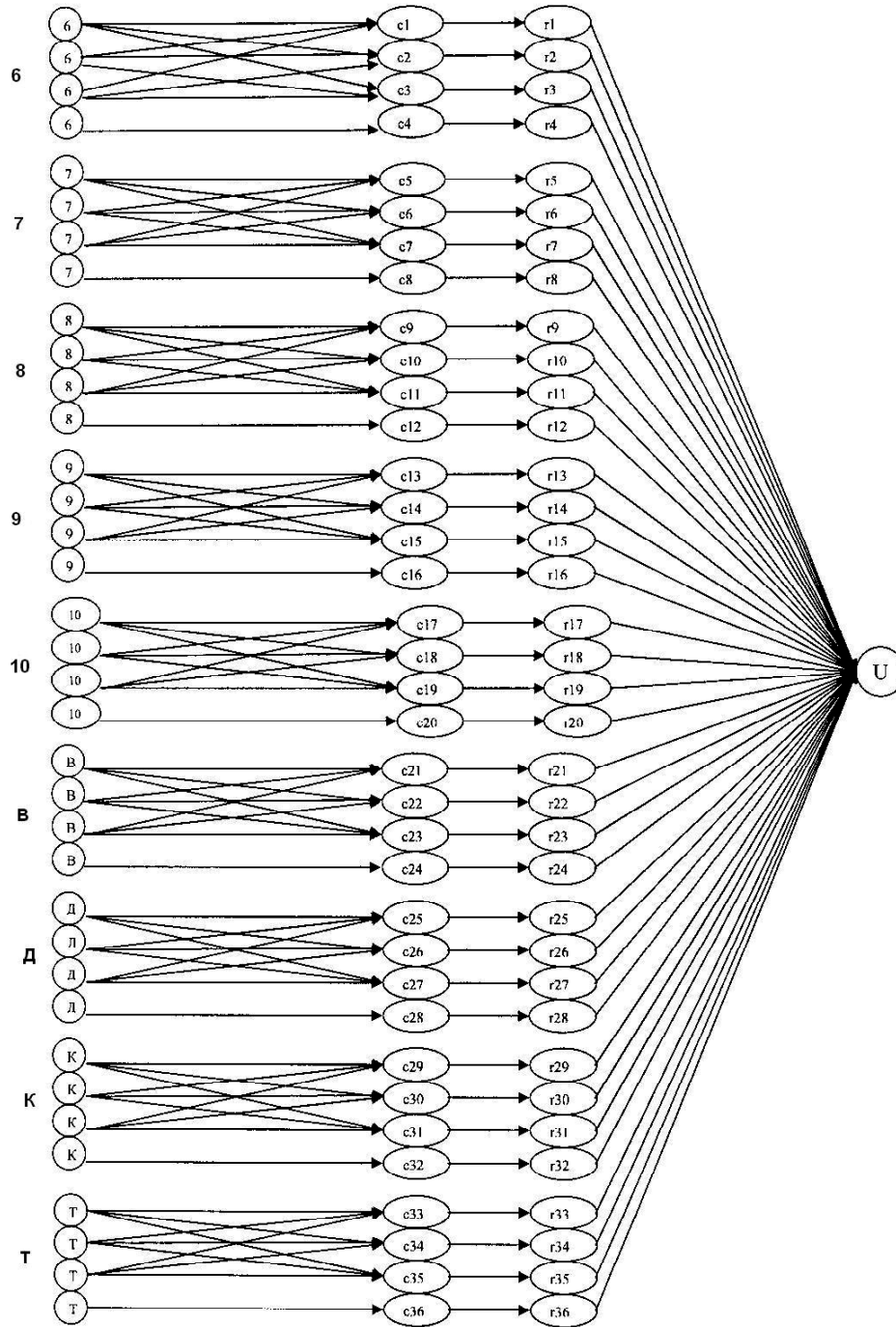


Рис. 3

$c4$ – если сигнал ≥ 1 (козырная шестёрка),
 $c5$ – если ≥ 1 (одна семёрка) и т. д.

Предыдущую схему в более сжатом виде можно изобразить следующим образом (см. рис. 4).

Первые элементы схемы – это S-элементы, которые при наличии соответствующей карты возбуждаются. Далее сигналы от них

идут на A-элементы, которые при выполнении условий также возбуждаются и передают сигнал, умноженный на соответствующий вес rI дальше на ячейку нейрона, где вычисляется

$$Z = \sum_{I=1}^{144} cI \cdot rI .$$

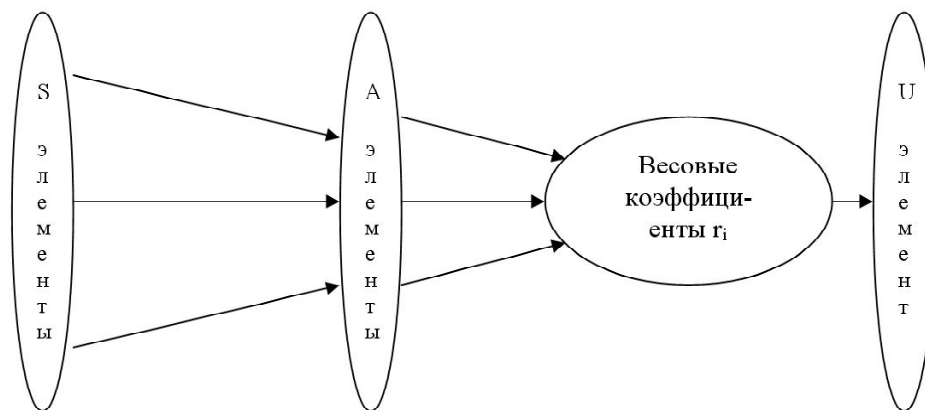


Рис. 4

В конце цепи стоит U-элемент, который выдает +1, если сумма пришедших на него сигналов положительна и -1, если отрицательна

$$U = f(Z) = \begin{cases} 1, & Z \geq 0 \\ -1, & Z < 0 \end{cases}$$

В случае +1 карту надо закрыть, в случае -1 карту со стола надо взять.

Описанная выше нейронная сеть была реализована в программе *fool2.exe*.

Сеть, рассматривая каждую локальную партию, в которой требуется закрыть карту, определяет: стоит ли закрыть или лучше взять все карты со стола, так чтобы средняя стоимость одной карты расклада была наибольшей. Проанализировав исходные данные, она должна разделить все карты в зависимости от их местонахождения на 2 класса:

1. Те, которые увеличивают среднюю стоимость одной карты.

2. Те, которые ее уменьшают.

В работе при обучении рассматривались только те варианты локальных партий, когда карта была покрыта. Для каждой партии вес активной связи изменялся на одинаковую величину: $\delta_i = 0.001$.

ПРОЦЕСС ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

Цель обучения нейронной сети состоит в определении её весовых коэффициентов. Обучение сети велось посредством анализа локальных партий, разыгрываемых с компьютером, играющим по усложнённому алго-

ритму. Для обучения с помощью программы *fool2.exe* в сеть автоматически было подано большое количество (порядка 10^5) сыгранных локальных партий и их результатов. В результате обучения нейронной сети были получены следующие значения коэффициентов связей, которые показаны на диаграммах (рис. 5).

Эти результаты были получены за сутки непрерывной работы программы *fool2.exe* на компьютере со следующей конфигурацией: *Intel® Xeon™ CPU 3.60GHz, 3.60GHz, 2.00 GB of RAM.*, при этом было сыграно 11573 партии.

АНАЛИЗ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

На первой диаграмме показано значение коэффициента связи, учитывающего влияние наличия той или иной карты на руках игрока на решение нейронной сети: закрыть карту или взять все карты со стола. Цифры 1, 5, 9, 13, 17, 21, 25, 29, 33 показывают наличие одной карты соответствующего значения 6, 7, 8, 9, 10, валет, дама, король, туз. Цифры 2, 6, 10, ..., 34, наличие 2-х карт. Цифры 3, 7, 11, ..., 35 – трех карт. Цифры 4, 8, 12, ..., 36 – наличие козырной карты. Заметим, что при наличии двух карт, например двух шестерок, активными будут и первая и вторая связи, то есть наличие двух шестерок сильнее влияет на решение, чем наличие одной шестерки.

Поскольку решение нейронной сети зависит от знака суммы всех значений активных связей, то наличие любой козырной кар-

ты добавляет отрицательное значение в сумму, то есть крыться козырными картами не выгодно. Если рассмотреть предельный случай, когда все карты козырные, то решение нейронной сети будет – взять карту со стола, поскольку вероятность получения козырной карты из колоды мала.

На второй диаграмме показано влияние карт вышедших из игры в отбой. Из нее следует, что если среди вышедших карт в отбой есть одна девятка, десятка, валет, дама или король, то карту со стола следует взять, во всех остальных случаях (например, из игры в отбой вышло 2 девятки и т. д.) карту стоит крыть. В данном случае это можно объяснить с помощью следующих логических рассуждений: если в отбой ушла только одна карта какого-то номинала или вообще не ушла, то при покрытии этой карты могут добавить ещё одну карту.

Анализ третьей диаграммы показывает зависимость решения нейронной сети от

карт, которые лежат на столе и которые будут взяты в случае отрицательного решения. По диаграмме можно сделать такой вывод: если на столе есть шестерки, семерки или восьмерки, то карту надо крыть, наличие других карт на столе оказывает меньшее влияние на результат или вообще его не оказывает.

Анализ четвёртой диаграммы показывает влияние на вывод нейронной сети от карт, которыми можно покрыть. Из диаграммы видно, что крыть надо некозырными семерками, восьмерками, девятками, десятками, валетами и дамами, а козырными не надо.

Учитывая количественное соотношение коэффициентов (коэффициенты первой диаграммы больше коэффициентов остальных диаграмм в 2–4 раза) из полученных диаграмм можно сделать следующие выводы. Наибольшее влияние на конечное решение нейронной сети, взять или побить карту, оказывают карты на руках, следующее

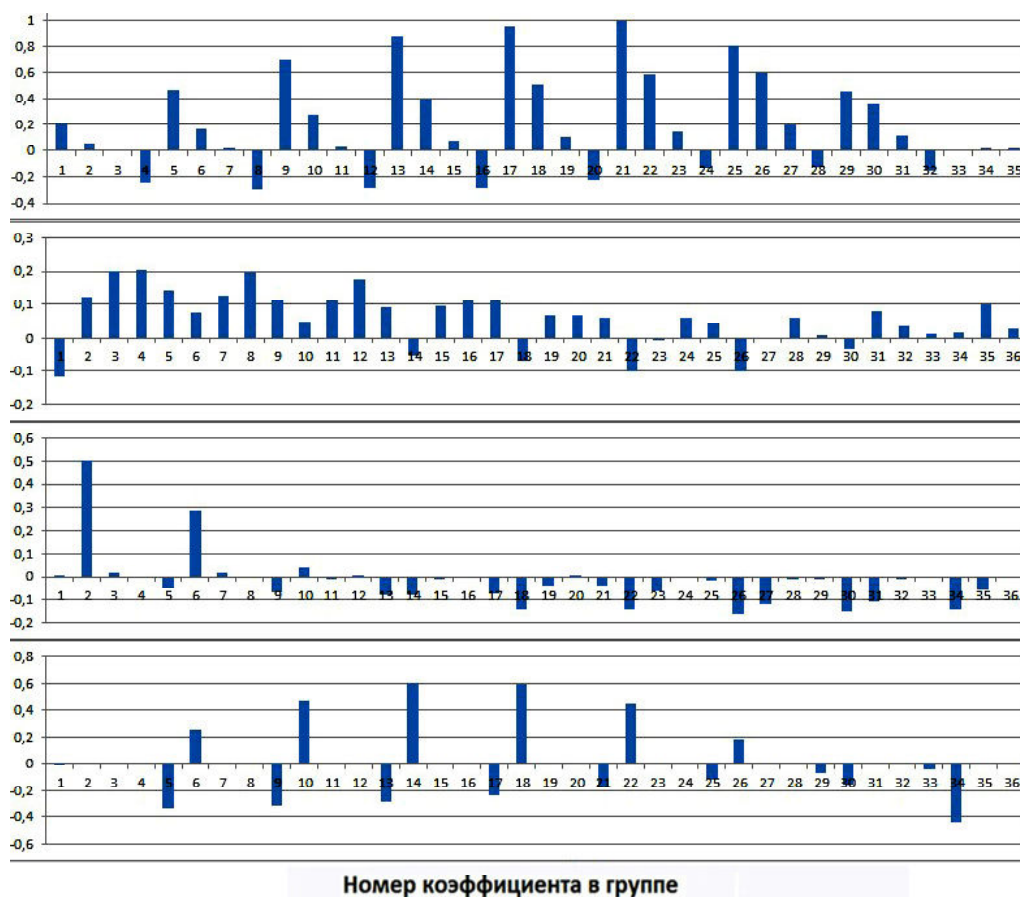


Рис. 5

по величине влияние – карта, которой можно покрыть. На третьем месте по важности – карты, лежащие на столе, которые можно взять или отправить в отбой, покрыв последнюю. И наименьшее влияние на результат оказывают карты, вышедшие из игры в отбой.

ТЕСТИРОВАНИЕ ПРОГРАММЫ НЕЙРОННОЙ СЕТИ

Тестирование программы нейронной сети показало следующее.

Слабый компьютер против обученной нейронной сети:

Всего сыграно партий: 16984.

Побед слабого компьютера: 7046 (41,49 %).

Побед нейронной сети: 9713 (57,19 %).

Ничьих: 225 (1,32 %).

Сильный компьютер против обученной нейронной сети:

Всего сыграно партий: 12509.

Побед сильного компьютера: 5813 (46,47 %).

Побед нейронной сети: 5404 (43,20 %).

Ничьих: 1292 (10,33 %).

Можно видеть, что нейронная сеть, обученная на основе игры сильного компьютера, играет лучше, чем слабый компьютер, и немного уступает сильному.

Анализ изменения средней стоимости карт для различных алгоритмов показал, что в случае простого алгоритма средняя стоимость карт практически не меняется (8,54), для усложнённого алгоритма она нарастает в течение игры (9,75), а для нейронной сети имеет место рост средней стоимости карт, но он несколько меньше роста в случае усложненного алгоритма (9,25).

Дисперсия средней стоимости карт для простого алгоритма меняется мало (5,39), для усложнённого алгоритма она даже несколько убывает в процессе игры (4,37), а для нейронной сети, она больше чем, в обоих случаях (5,70).

На рис. 6 приведены характерные графики изменения средней стоимости карт и дисперсии в процессе игры для нейронной сети.

Созданная нейронная сеть является достаточно гибкой структурой и допускает раз-

личные модификации. Известно, что игрок, как правило, пытается оценить неизвестные карты противника и карты в колоде. В отличие от остальных данных, использовавшихся для обучения, эти величины не могут быть точно вычислены во время игры. Для их определения используются элементы нечёткой логики.

В начале игры средняя стоимость неизвестных карт равна 7,25. После удаления одной карты эта стоимость будет лежать в пределах от 6,94 до 7,43 и т. д. Максимально возможный диапазон изменения стоимости карты от 1 до 18.

Пределы изменения средней стоимости неизвестных карт лежат в верхней области, показанной на рисунке рис. 7.

Весь диапазон изменения стоимости карт был разбит на пять интервалов, и были введены лингвистические переменные:

[1; 4,4] – очень плохие карты,

(4,4; 7,8] – плохие карты,

(7,8; 11,2] – средние карты,

(11,2; 14,6] – хорошие карты,

(14,6; 18] – очень хорошие карты.

В соответствии с этим разбиением в сеть было добавлено 10 дополнительных входов, пять – для карт противника и пять – для колоды. В зависимости от оценки средней стоимости неизвестных карт на соответствующий вход подавалась единица или ноль.

ОБУЧЕНИЕ НОВОЙ СЕТИ

После изменения структуры сеть была снова обучена. При этом соотношения пер-



Рис. 6

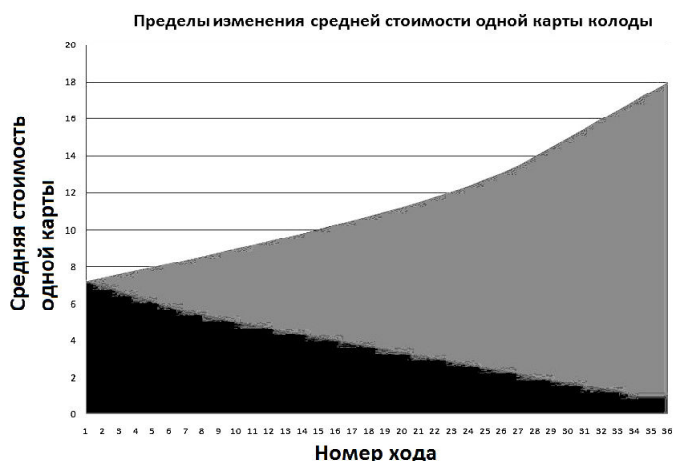


Рис. 7

вых 144 коэффициентов качественно не изменились. Диаграммы десяти коэффициентов, характеризующих оценку карт противника и карт колоды, показаны на рис. 8.

Дополнительные коэффициенты оказывают сильное влияние на решение сети покрыть или брать карту. Это можно объяснить тем, что два из десяти коэффициентов всегда активны, а значит, в ходе обучения на каждом шаге обучения меняются их значения. Можно также заметить, что первый, пятый, шестой и десятый коэффициенты почти равны нулю, это значит, что в ходе игры средняя стоимость одной карты очень редко принимает значения ниже 4,4 или выше 14,6. Из первых пяти коэффициентов следует, что если из колоды ожидается плохая карта, то покрывать не надо, а если средняя или хорошая, то карту следует покрывать. В отношении карт противника из анализа коэффициентов следует, что если у противника плохие карты на руках, то карту покрывать не стоит,

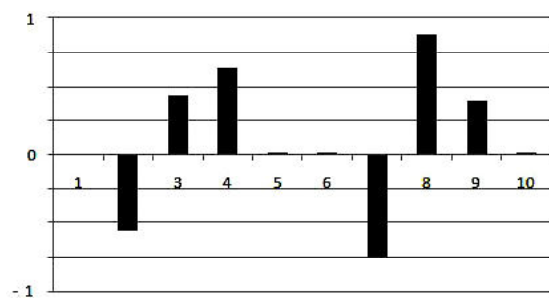


Рис. 8

а если у противника средние или хорошие карты, то карту следует покрывать.

После ввода дополнительных коэффициентов также было проведено сравнение новой сети с простым и сильным компьютером.

Слабый компьютер против обученной нейронной сети с дополнительными коэффициентами:

Всего сыграно партий: 17165.
 Побед слабого компьютера: 7256 (42,27 %).
 Побед нейронной сети: 9673 (56,35 %).
 Ничьих: 236 (1,37 %).

Сильный компьютер против обученной нейронной сети с дополнительными коэффициентами:

Всего сыграно партий: 16802.
 Побед сильного компьютера: 8640 (51,42 %).
 Побед нейронной сети: 7904 (47,04 %).
 Ничьих: 258 (1,54 %).

Из этих данных можно сделать вывод, что после добавления дополнительных коэффициентов решения уменьшилось количество ничьих.

На рис. 9 приведено изменение средней стоимости карт и дисперсии для модифицированной нейронной сети.

Математическое ожидание средней стоимости карты растёт, однако не так сильно как у предыдущей сети, при этом значительно выросла дисперсия, особенно на последних ходах, когда оценка карт колоды и противника точна. В этом случае игра просчи-



Рис. 9

тывается точно, а нейронная сеть продолжает играть с коэффициентами, которые были получены при усреднении.

Проведённые исследования показывают, что принимаемые во время игры решения человеком, основанные на как бы чувственном уровне понимания игровой ситуации, могут быть просчитаны количественно. Гибкость структуры нейронной сети позволяет создавать программы, которые в процессе обучения могут изменять свою структуру, то

есть удалять избыточные элементы или добавлять недостающие. Всё это позволяет говорить о возможности развития нового рефлексивно-ориентированного программирования, которое является функциональным расширением парадигмы объектно-ориентированного программирования.

Желающие получить программу *fool2.exe* обращайтесь по электронному адресу: *ALF19545@ramble.ru*.

Литература

1. Ляхов А.Ф. Тришин И. М. Алгоритмы и программы управления компьютером в азартных играх, созданные на основе теории нечётких множеств // Компьютерные инструменты в образовании, 2006. № 2. С. 65–71.

2. Ляхов А.Ф. Февральских С.М. Самообучающаяся программа управления компьютером в карточных играх, созданная на основе теории нечётких множеств // Компьютерные инструменты в образовании, 2007. № 5. С. 76–82.

3. Хайкин С. Нейронные сети. Полный курс. М.–СПб.–Киев, 2006.

Приложение

ОПИСАНИЕ ПРОГРАММЫ FOOL.EXE

В программе fool2.exe реализованы простой и сложный алгоритмы поведения игроков и нейронная сеть. Интерфейс программы показан на рис. 1. Заметим, что, управление программы достаточно простое и не требует специального описания управляющих кнопок.

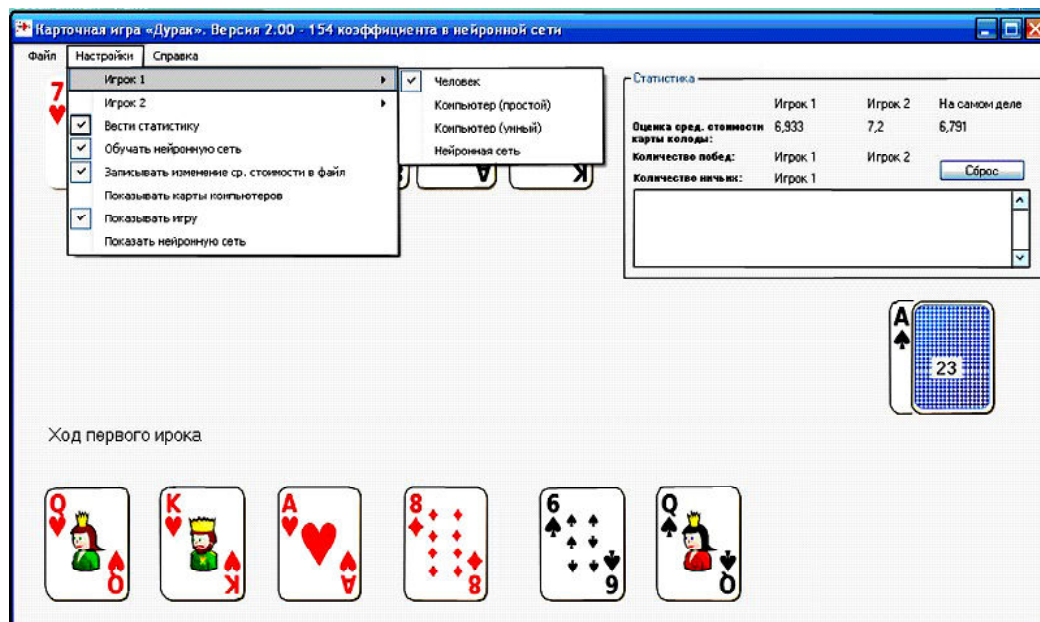


Рис. 1

На панели отображается окно, в котором описывается основание принятия решения первым и вторым игроком. Приводится оценка средней стоимости карт в колоде каждым игроком и реальная стоимость, приводится количество выигранных партий первым игроком. В программе имеется возможность видеть карты каждого игрока. Заметим, что при игре компьютера с компьютером и при обучении нейронной сети игра происходит очень быстро и поэтому карты можно убрать со стола.

Программа позволяет выбрать различные виды игры:

1. Человек с компьютером, причём компьютер может играть по первому или второму алгоритму, или использовать нейронную сетью.

2. Компьютер играет с компьютером. Этот режим может быть использован для исследования эффективности работы алгоритмов, обучения нейронной сети и её сравнения с реализованными игровыми алгоритмами.

Программа позволяет вычислить различные статистические характеристики процесса игры и обучения нейронной сети (среднюю стоимость карт игроков, количество побед каждого из игроков изменение коэффициентов нейронной сети) и визуализировать.

COMPUTER SIMULATION OF GAMER'S BEHAVIOR IN INTELLECTUAL CARD GAME "FOOL" USING NEURAL NETWORK

Abstract

In the paper artificial neural network is described and created for decision-making in the indefinite gaming situation of classic card game "Fool". The research shows human-made decisions based on perceptual level of comprehension during the game can be quantitatively calculated. Flexible structure of the neural network can morph during the process of training i.e. excessive neurons can be removed and missing ones can be added. The software created along the research is an example of new reflex-oriented programming paradigm being functional extension of well-known object-oriented programming paradigm.

Keywords: neural network, game theory, fuzzy logic, training of neural network.

*Ляхов Александр Федорович,
кандидат физико-математических
наук, доцент кафедры
теоретической механики механико-
математического факультета
НГУ им. Н.И. Лобачевского
(Национального исследовательского
университета),*

Lyakhov@mm.unn.ru,

*Тришин Илья Михайлович,
инженер-программист
«Интел Телнолджис»,
admin@top22.ru.*



Наши авторы, 2013.

Our authors, 2013.