

Ляхов Александр Федорович,
Февральских Сергей Владимирович

САМООБУЧАЮЩАЯСЯ ПРОГРАММА УПРАВЛЕНИЯ КОМПЬЮТЕРОМ В КАРТОЧНЫХ ИГРАХ, СОЗДАННАЯ НА ОСНОВЕ ТЕОРИИ НЕЧЁТКИХ МНОЖЕСТВ

Проблема создания систем управления на основе искусственного интеллекта возникла в эпоху появления ЭВМ первого поколения. Описанию различных методов и подходов к её решению посвящено большое количество публикаций¹.

В пятидесятых годах прошлого века для исследования динамики сложных не поддающихся математическому описанию систем были созданы модели, имитирующие нервную деятельность животных, вызываемую различными раздражителями, при этом нервная система представлялась в виде сети

нейронов. Процесс принятия решения в этой модели представляется следующим образом: на вход в нейронную сеть подаётся сложный набор сигналов, а на выходе принимается некоторый отклик. Требуется так подобрать весовые коэффициенты входных сигналов, чтобы выходной сигнал удовлетворял заданным требованиям. С помощью таких моделей осуществляется прогнозирование и управление случайными процессами, например, прогнозируется изменение стоимости ценных бумаг на торговых биржах.

В это же время возникло понятие обучения и самообучения «нейронной сети». Под самообучением понимается автоматизированный процесс изменения весовых коэффициентов входных сигналов «сети» для получения необходимых выходных сигналов. В конце двадцатого века в связи с появлением качественно новой вычислительной техники появилась возможность создавать многомерные, нелинейные «нейронные сети» и исследовать их.

В данной работе приводится пример создания самообучающейся управляющей программы, которая имитирует обучение игрока в карты.



Обучения заканчивается, когда обучающийся игрок выигрывает 70% партий ...

¹ В настоящее время в этом направлении достигнуты большие успехи, примером которых может служить победа программы Deep Fritz над чемпионом мира по шахматам Владимиром Крамником.

В журнале «Компьютерные инструменты в образовании» (№ 2, 2006) была опубликована статья А.Ф. Ляхова, И.М. Тришина «Алгоритмы и программы управления компьютером в азартных играх, созданные на основе теории нечётких множеств», в которой описана программа, осуществляющая управление игрой компьютера в известной карточной игре «дурак».

Проведенные в этой работе исследования позволили создать программу, имитирующую процесс обучения игрока. В процессе обучения компьютер разыгрывает большое число партий, играя за обоих противников. Вводится понятие периода обучения – некоторое количество партий, при разыгрывании которых параметры игроков остаются постоянными. После разыгрывания партий первого периода обучения компьютер вносит изменение в оценку игровых карт одного из игроков, при этом происходит изменение лингвистических термов, и процесс разыгрывания повторяется. Если число побед обучающегося игрока превышает число побед второго игрока за второй период обучения, то считается, что процесс обучения идёт успешно, и он может быть продолжен. Если изменения привели к ухудшению результата, то происходит отказ от них, вносятся новые изменения, после чего процесс обучения продолжается. Обучение заканчивается, когда обучающийся игрок выиграет 70% партий за период обучения.

Программа, реализующая процесс обучения, написана на языке Pascal в оболочке

Delphi. Интерфейс программы позволяет наблюдать весь процесс обучения компьютера. Для тестирования программы была включена возможность просмотра отдельных партий и реализована возможность просмотра и сохранения промежуточных результатов.

Созданная программа позволяет проводить исследования устойчивости и эффективности процесса самообучения компьютера.

Опишем основные идеи, положенные в основу управляющих программ игроков [1].

Правила классической игра в «дурака» хорошо известны. Игра состоит из локальных партий (один игрок ходит, другой отвечает). После розыгрыша локальной партии игроки дополняют свои наборы карт до шести карт из колоды. Побеждает тот игрок, у которого в конце последней локальной партии не остается карт.

Рассмотрим игру двух игроков. В этом случае игру можно классифицировать как антагонистическую игру с нулевой суммой, с неполной информацией.

Введём количественную шкалу оценки карт и лингвистические термы (см. табл. 1).

Определим следующие лингвистические термы.

Определение 1.

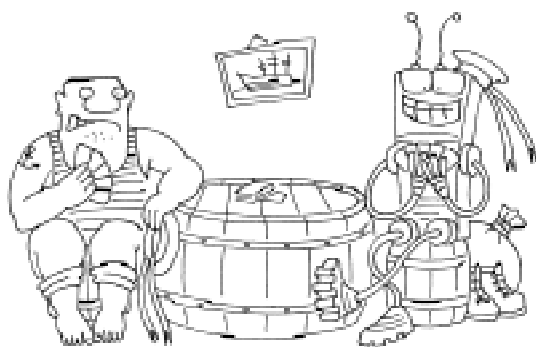
Обозначим термом «плохие» карты – карты 6, 7, 8, 9, 10 некозырных мастей.

Определение 2.

Обозначим термом «средние» карты – валет, дама, король, туз некозырной масти.

Таблица 1

Карта	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



Рассмотрим лингвистическую переменную «Качество расклада».

Определение 3.

Обозначим термом «хорошие» карты – все карты козырной масти.

Количественный диапазон изменения «плохих» карт [1–5], «средних» карт [6–9], и «хороших» карт [10–18]. Таким образом, любое подмножество карт будет разделено на три подмножества.

Рассмотрим множество различных пар карт. Пара карт «плохая», если каждая карта «плохая», карты «средние», если они состоят из одной «хорошей» и одной «плохой» карты, и карты «хорошие», если обе карты «хорошие».

Введём определение лингвистических термов для любого подмножества карт.

Определение 4.

Подмножество назовём «хорошим», если количество «хороших» карт в нём строго больше $2/3$ общего количества карт.

Определение 5.

Подмножество назовём «плохим», если количество «плохих» карт в нём строго больше $2/3$ общего количества карт.

Определение 6.

Подмножество назовём «средним», если не выполняются условия определения 4 и определения 5.

Рассмотрим лингвистическую переменную «Качество расклада». Для реализации лингвистической переменной необходимо определить значения термов этой переменной. В соответствии с выше приведенными определениями, зададим её термами «Плохой», «Средний» и «Хороший».

Значение переменной «Качество расклада» определяется средним значением карт расклада. Степень принадлежности расклада к тому или иному терму лингвистической переменной «Качество расклада» определяется так называемой функцией принадлежности.

Описав карты игрока, ожидаемые карты из колоды и предполагаемые карты противника в лингвистических термах можно построить управляющие лингвистические выражения. Например: «если у игрока А «хорошие карты» и карта, которая придет из колоды, «хорошая», и если у игрока В вероятней всего «плохие карты», то игрок А должен ходить «хорошей» картой».

При создании программы, управляющей действиями компьютера, моделирующего одного из игроков, были созданы таблицы лингвистических термов. Приведём одну из управляющих таблиц.

Таблица управляющего действия при выборе взятия карты со стола или из колоды (табл. 2). После хода игрока В игроком А (компьютером) принимается решение покрыть карту (отдать одну свою карту и взять одну из колоды) или взять (взять карту со стола).

Все используемые в программе управляющие действием компьютера таблицы описаны в работе [1].

Эта программа позволила поставить вопрос: а что будет, если компьютер будет играть с компьютером, то есть компьютер будет моделировать обоих игроков? Можно

Таблица 2

		Расклад после взятия карты из колоды		
		Плохой	Средний	Хороший
Расклад после взятия карты со стола	Плохой	Из колоды	Из колоды	Из колоды
	Средний	Со стола	Из колоды	Из колоды
	Хороший	Со стола	Со стола	Со стола

предположить, что в среднем количество выигрышей каждого игрока будет равно количеству проигранных.

Возникает следующий вопрос: как надо изменять управляющую программу одного из игроков, чтобы он получил большее число выигрышей?

Изменение программы может осуществляться в автоматическом режиме, то есть можно создать самообучающуюся систему.

ОПИСАНИЕ САМООБУЧАЮЩЕЙСЯ ПРОГРАММЫ FOOLS

В программе смоделированы два игрока, играющие друг с другом. Программа написана на языке Pascal в оболочке Delphi 5.0.

ПРИНЦИПАЛЬНЫЙ АЛГОРИТМ ПРОГРАММЫ FOOLS

В начале игры обоим игрокам указывается стоимость каждой карты, в соответствии с таблицей, приведенной выше. После этого компьютер разыгрывает заданное число партий (период развития) и определяет процент выигрышей первого игрока. Если процент выигрышей первого игрока меньше 70%, то начинается процесс обучения. Компьютер случайным образом выбирает карту, такую, чтобы одна из соседних карт, то есть карт, стоящих рядом по таблице стоимости, имела качественную характеристику, отличную от характеристики данной карты, и изменяет её стоимость. К стоимости карты добавляется случайно сгенерированное число из интервала ограниченного ценной соседних карт. Затем вновь разыгрывается определённое число партий и проверяется процентное соотношение выигрышей. Если оно уменьшилось, то компьютер возвращает предыдущую стоимость карт и вновь выбирает случайно карту и меняет её качественную ценность, если же процентное соотношение увеличилось, то он изменяет качественную характеристику другой карты, выбирая её по тому же принципу. Этот процесс продолжается до тех пор, пока количество выигрышей первого игрока не достигнет 70%.

ТЕСТИРОВАНИЕ ПРОГРАММЫ FOOLS

Тестирование программы проводилось многократным разыгрыванием при разных периодах развития, а также многократным просмотром тренировочных партий.

ИНТЕРФЕЙС ПРОГРАММЫ

Меню «Файл» содержит несколько управляющих команд.

Команда «Новая игра» показывает результаты однократного разыгрывания партии. Эта команда совместно с выполнением опции «проверочная партия» позволяет наблюдать пошаговый процесс розыгрыша партии (см. рис. 2).

Команда «Начать развитие» открывает основную форму программы и позволяет проводить исследование процесса обучения.

Команда «Закончить развитие» прерывает процесс обучения до его завершения.

Команда «Закончить игру» прерывает процесс игры одиночной партии.

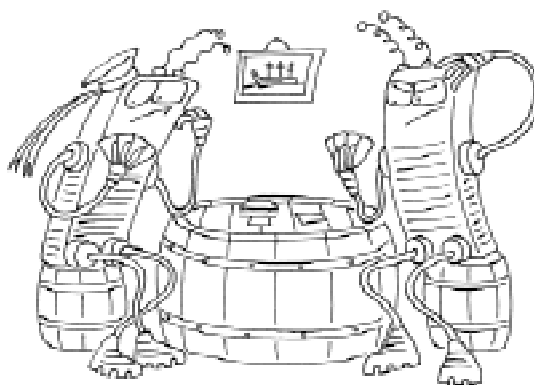
Команда «Сохранить результат» сохраняет результат обучения в указанный файл.

Команда «Выход» осуществляет стандартный выход из программы.

В меню «Опции» содержатся следующие команды

Команда «Файл» → «Новая игра» → «Проверочная партия» позволяет проводить пошаговый просмотр отдельной партии.

Команда «Показывать статистику» показывает окно, в котором отображаются количественные характеристики карт на



...а что будет, если компьютер будет играть с компьютером...

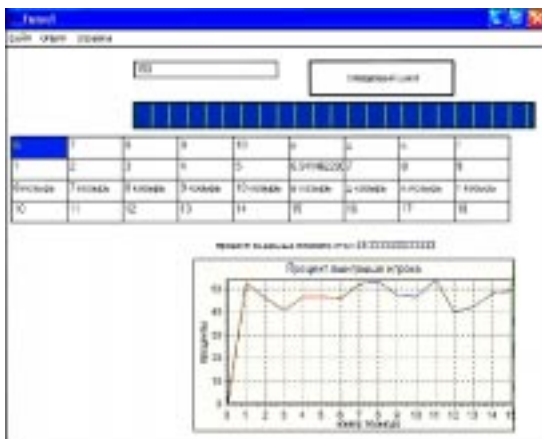


Рис. 1

столе и игроков, а также отображается процесс принятия решения компьютером на основе теории нечётких множеств.

Команда «Игра с использованием результатов обучения» позволяет загрузить сохранённые ранее результаты обучения и использовать его при разыгрывании одиночной партии.

Команда «Развитие без обучения» позволяет во время процесса обучения разыграть такое же число партий, как период развития, при текущих характеристиках карт. Результаты разыгрывания этого цикла будут отображены на графике синим цветом.

Команда «Изменить цену карт» позволяет в ручном режиме задать произвольную начальную стоимость карт.

Кнопка «Справка» позволяет получить доступ к справке по работе с программой.

Основное окно работающей программы выглядит следующим образом.

В окне «Период развития» пользователь указывает количество партий разыгрывае-



Рис. 2

мых с данной ценовой шкалой карт. Показана шкала стоимости карт, которая изменяется при каждом цикле разыгрывания. На графике приводится процент выигрышей за период развития.

Синяя линейка отображает вспомогательную информацию о реальном времени процесса розыгрыша одного периода развития.

Розыгрыш каждого периода развития осуществляется после щелчка по кнопке «Следующий цикл» (см. рис. 1).

Второе окно программы (рис. 2) позволяет проводить тестирование программы и наблюдать процесс обучения в одиночной партии. Управление осуществляется щелчком по кнопке «Следующий ход».

КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

Приведём несколько графиков показывающих процесс обучения компьютера с различными периодами развития.

На рис. 3 изображён график результата обучения при периоде 20 партий. Обучение было прервано на 68 периоде при достиже-

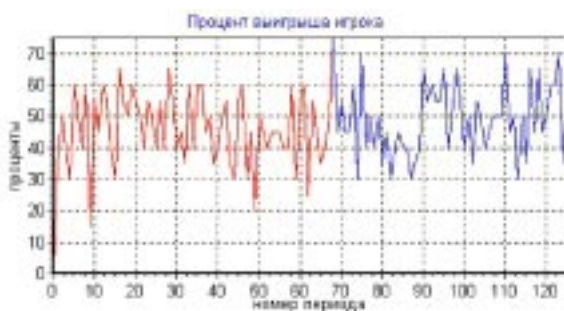


Рис. 3



Рис. 4

Таблица 3

период	1	5	7	8	9	10	11	12	13	15	20	25	30
кол. период	1,5	8,8	5	5	5,8	7,5	7,9	14,8	16,1	17,8	19,9	44	95
среднеквадр. откл.	1,02	4,7	2,9	3,4	5,2	7,9	6,1	7,5	6,7	12	11,7	21	38
отн. среднеквадрат. откл	0,68	0,53	0,58	0,68	0,89	1,05	0,78	0,51	0,42	0,67	0,59	0,48	0,4

нии 70% выигрыша. Далее процесс розыгрышей происходил без обучения с фиксированными значениями стоимости карт. На интервале [68, 125] периодов 70% уровень достигался четыре раза, т.е. частоты выигрыша $p_{обуч} = 0,075$.

На рис. 4 изображен график розыгрыша периодов по двадцать партий без обучения. Можно видеть, что 70 % уровень достигался пять раз, и он равномерно распределён по всему интервалу эксперимента (~ 125 периодов), соответственно частота события $p_n = 0,04$. Полученная частота достижения порогового значения выигрышей практически совпадает с теоретической вероятностью этого события.

Действительно, поскольку оба игрока при отсутствии обучения играют по одной и той же программе, то результат игры определяется случайным раскладом карт. Из симметрии игры можно предположить, что в этом случае вероятность выигрыша в одной партии $p = \frac{1}{2}$.

Вероятность выиграть k партий из n разыгрываемых партий соответственно равна $P = C_n^k p^k (1 - p)^{n-k}$ (формула Бернулли). В нашем случае период обучения $n = 20$, количество выигранных партий $k = 14$ и вероятность этого события будет равна $P = 0,037$.

Следовательно, обучение привело к увеличению частоты достижения порогового значения выигрышей $p_{обуч} > p_n$.

В таблице 3 приведено среднее число периодов обучения и среднеквадратичное отклонение наблюдаемых экспериментальных значений от среднего значения

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (P_{cp} - P_i)^2}{N}}$$

и относительное значение среднеквадратичного отклонения. Обучение с каждым периодом проводилось десять раз.

На рис. 5 приведены графики полученных зависимостей.

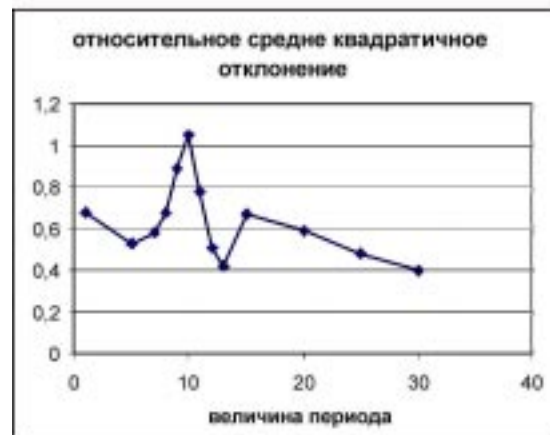
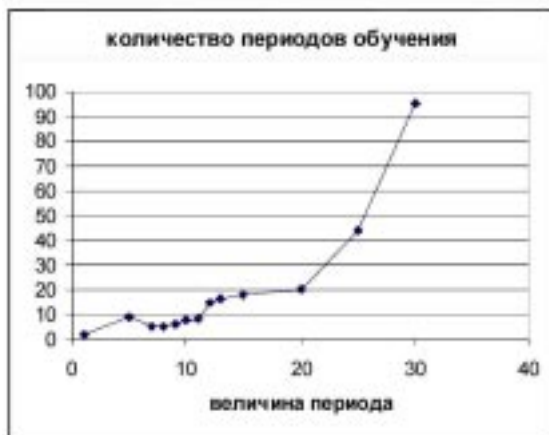


Рис. 5



...при небольшом периоде развития процесс обучения заканчивается быстро, но оно имеет неустойчивый характер...

Экспериментальные исследования показали, что при небольшом периоде развития процесс обучения заканчивается быстро, но оно имеет неустойчивый характер, определяемый случайным раскладом карт.

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. № 2, 2006.

При большом периоде развития процесс обучения носит длительный характер и иногда не достигается за реальное большое время эксперимента (~ 50 и более).

Относительное среднее квадратичное отклонение при больших периодах обучения мало и имеет тенденцию к уменьшению. Это означает, что предложенная в программе стратегия, статистически устойчива к малым возмущениям. Малые изменения стоимости карт мало влияют на результат при больших периодах обучения.

Большие изменения ценностной шкалы одного из игроков переводят процесс игры в другое устойчивое состояние, то есть один из игроков постоянно выигрывает.

В заключение заметим, что построенная модель самообучающегося игрока может быть описана в терминах теории «нейронных сетей».

Ляхов Александр Федорович,
кандидат физико-математических наук, доцент кафедры теоретической механики механико-математического факультета Нижегородского государственного университета им. Н.И. Лобачевского,

Февральских Сергей Владимирович,
студент 1 курса механико-математического факультета Нижегородского государственного университета им. Н.И. Лобачевского.



Наши авторы, 2007
Our authors, 2007