

абстрагирование, анализ, синтез, сравнение, моделирование, что способствует развитию их творческих способностей.

Литература

1 Наговицына, А.А. Педагогические условия развития творческих способностей школьников во внеурочной деятельности [Электронный ресурс] / А. А. Наговицына, П. А. Тутолмин-Линдт. – Режим доступа: <https://www.scienceforum.ru/2016/1763/20244>.

2 Синельникова, О. Н. Развитие творческих способностей учащихся – одна из форм профессиональной самореализации педагога [Электронный ресурс] / О. Н. Синельникова. – Режим доступа : <https://www.prodlenka.org/opyt-i-problemy>.

УДК 004.8

В. А. Прохоренко

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧЕ ПОИСКА ПУТИ

Целью работы было исследование возможности применения подходов управления на основе нейронных сетей для решения задач поиска пути. В качестве примера задачи поиска пути рассмотрена задача поиска управляемым объектом выхода из лабиринта. Для построения нейроконтроллеров используются нейронные сети типа многослойный персептрон и рекуррентная нейронная сеть на базе многослойного персептрона, сравнивается эффективность этих контроллеров.

Введение

Искусственные нейронные сети успешно применяются для решения задач классификации, прогнозирования, аппроксимации, сжатия данных и управления. Несмотря на сложность прикладных задач, которые могут быть решены с применением нейронных сетей, последние представляют собой достаточно простую и удобную модель. Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения – одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение.

Рекуррентная сеть на базе многослойного персептрона представляет собой архитектуру, которая характеризуется наличием связей между слоями в форме элементов запаздывания. Это позволяет сети накапливать память о своих предыдущих состояниях и воспроизводить последовательности реакций.

Особенности рекуррентных нейронных сетей потенциально предоставляют множество возможностей для моделирования биологических нейронных сетей. Однако большинство возможностей на данный момент плохо изучены в связи с возможностью построения разнообразных архитектур и сложностью их анализа.

Нейросетевое моделирование контроллера применяется в тех случаях, когда существует качественный контроллер управляемой системы. Нейронная сеть выступает в роли аппроксиматора его функции и обучается таким образом, чтобы моделировать его воздействия на управляющую систему. Использование полученного таким образом контроллера в некоторых ситуациях может быть более целесообразным, чем использование оригинального. Например, в силу общих свойств нейронных сетей, которые включают в себя способность к обобщению и обработке зашумленных данных, отказоустойчивость в силу параллельной архитектуры.

В процессе работы была исследована возможность применения подходов управления на основе нейронных сетей для решения задач поиска пути. В качестве примера задачи поиска пути рассмотрена задача поиска управляемым объектом выхода из лабиринта. Нейронная сеть в данной задаче использована в качестве контроллера управляемого объекта, определяющего его ход на каждом шаге. В процессе работы были поставлены и решены следующие задачи:

- формализация задачи;
- разработка действующих программных моделей нейронных сетей, реализация алгоритмов их обучения;
- подготовка обучающих множеств для нейронных сетей и их обучение для последующего использования в качестве контроллера.

Формализация задачи и архитектуры нейронных сетей

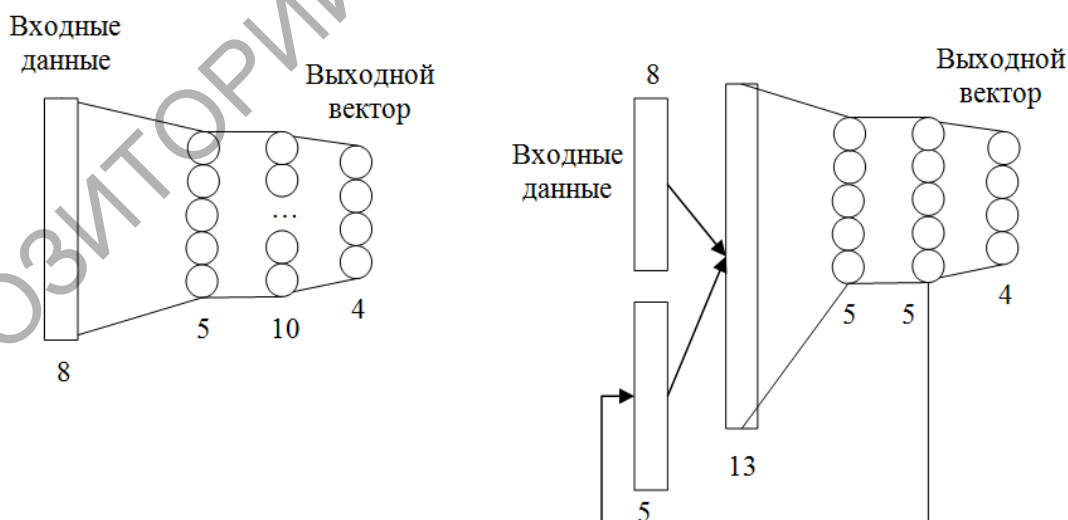
Для рассматриваемой задачи предполагается, что в лабиринте нет замкнутых маршрутов. В таком случае контроллер для управляемого объекта, проходящего лабиринт, несложно реализовать алгоритмически. В качестве входных данных контроллера выступает вектор, описывающий соседние с текущей позицией клетки, а также вектор, показывающий текущее направление движения. Клетке с препятствием соответствует координата с единицей, свободной клетке – координата с нулем. Результатом работы контроллера является вектор из четырёх элементов, максимальная координата которого определяет направление следующего движения объекта.

Было рассмотрено три типа контроллеров на основе нейронных сетей:

- контроллер на основе многослойного персептрона;
- контроллер на основе рекуррентной нейросети на базе многослойного персептрона;
- аддитивный контроллер, комбинирующий сигналы двух предыдущих.

Многослойный персептрон состоит из трёх полносвязных слоёв. В первом слое 5 нейронов, во втором – 10 нейронов, в третьем – 4 нейрона. Рекуррентная нейросеть также имеет три полносвязных слоя, в которых содержится 5, 5 и 4 нейрона, при этом второй скрытый слой имеет обратные связи с первым слоем через элементы единичной задержки. Схемы нейронных сетей приведены на рисунке 1.

В рассматриваемых нейронных сетях в качестве функции активации нейронов используется гиперболический тангенс.



Слева – многослойный персептрон;
справа – рекуррентная сеть на базе многослойного персептрона

Рисунок 1 – схемы использованных нейронных сетей.

Обучение нейронных сетей и результаты

Процесс обучения нейронной сети заключается в поиске таких значений настраиваемых параметров сети (весовых коэффициентов связей между нейронами и уровней активации нейронов), чтобы она осуществляла правильное отображение. Обучение можно рассматривать как нелинейную оптимизационную задачу минимизации некоторой функции ошибки, заданной на обучающем множестве, относительно всех настраиваемых параметров сети.

Обе нейронные сети обучались с использованием метода сопряжённых градиентов. Данный метод позволил достичь быстрой и монотонной сходимости без заикливания в точках локального минимума. Производные вычислялись с помощью обратного распространения ошибки. Коррекции настраиваемых параметров осуществлялись после предъявления в произвольном порядке всех пар или последовательностей из обучающего множества. Обучающее множество представляет собой пары входных и требуемых выходных векторов в случае многослойного персептрона и последовательности пар векторов в случае рекуррентной нейросети. В качестве минимизируемой ошибки использовалась среднеквадратичная ошибка по обучающему множеству.

Обучающие множества сгенерированы путём применения существующего контроллера к различным ситуациям, возникающим при решении задачи. Для обучения рекуррентной нейронной сети использованы последовательности выборов из одного, двух, трёх и четырёх ходов. Всего в процессе обучения использовано 100 сгенерированных последовательностей. Для тестирования обученных нейросетей использовано множество из 25 примеров.

Многослойный персептрон не имеет внутреннего состояния и в рамках рассматриваемой формализации задачи не учитывает входные и выходные данные в предыдущие моменты времени. По этой причине контроллер на основе нейронной сети такого типа может быть успешно применён в рассматриваемой задаче в очень ограниченном наборе ситуаций и плохо подходит для её решения. Решение задачи было найдено только в 28 % использованных для тестирования примерах.

Рекуррентная нейронная сеть после соответствующего обучения продемонстрировала способность воспроизводить действия алгоритмического контроллера в большинстве случаев. Это позволяет найти путь с помощью контроллера на основе рекуррентной нейронной сети в большинстве (84 %) примеров.

При использовании аддитивного контроллера, выходные сигналы обеих нейросетей суммировались. На рассматриваемом множестве примеров, использованном для тестирования, применение такого контроллера позволило незначительно улучшить результат (88 %) по сравнению с использованием рекуррентной нейросети.

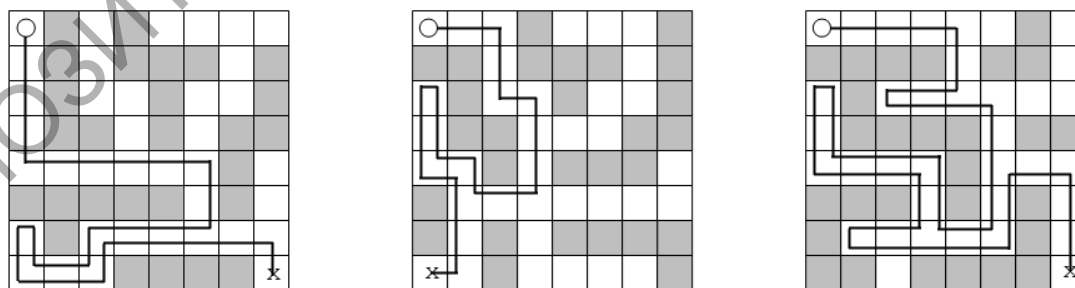


Рисунок 2 – Примеры нахождения пути с использованием нейроконтроллеров

Заключение

В ходе работы поставленные задачи были решены в полной мере. Проведенные эксперименты показали, что рекуррентная нейронная сеть может достаточно успешно

применяться в рассматриваемой задаче в качестве контроллера, в то время как применение многослойного персептрона показало себя неэффективным при рассматриваемой формализации задачи. Применение аддитивного контроллера из нескольких обученных отдельно нейронных сетей требует дальнейшего изучения.

Литература

- 1 Уоссермен, Ф. Нейрокомпьютерная техника: теория и практика / Ф. Уоссермен. – М. : Мир, 1992. – 240 с.
- 2 Заенцев, И. В. Нейронные сети: основные модели / И. В. Заенцев. – Воронеж : ВГУ, 1999. – 76 с.
- 3 Осовский, С. Нейронные сети для обработки информации / С. Осовский. – Москва : Финансы и статистика, 2002. – 345 с.
- 4 Omidvar, O. Neural Systems for Control / O. Omidvar, D. L. Elliott. – New York : Academic Press, 1997. – 358 p.
- 5 Widrow, B. Pattern-recognizing control systems / B. Widrow, F. W. Smith // Proceedings of Computer and Information Sciences. – 1964. – Vol. 12. – P. 288–317.

УДК 004.7

А. Д. Разгуляев, Н. Б. Осипенко

КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ ДЛЯ ТОРГОВО-ПРОИЗВОДСТВЕННОГО ПРЕДПРИЯТИЯ

Статья посвящена описанию разработанного интернет-магазина торгового-производственного предприятия дверных и оконных конструкций. Разработанное клиент-серверное приложение построено с применением паттерна программирования MVC, среды разработки JetBrains PhpStorm, базы данных MySQL, языка программирования PHP, языка сценариев JavaScript. Оно может быть использовано в качестве типового примера реализации любого производственно-торгового предприятия.

В настоящее время есть определенная обратная корреляция между качеством программного обеспечения и объемом кода, написанным для него, т. е. чем больше программный продукт, тем больше плотность ошибок на 1000 строк кода в нем [1]. Ошибки можно условно разделить на 2 вида: опечатки и ошибки в проектировании. Причём если первые можно «отловить» с помощью стандартных средств разработки и тестирования, то вторые представляют наибольшую трудность при выявлении. Кроме этого, с увеличением программного продукта возрастает сложность внесения изменений и корректировок, что соответственно увеличивает время на разработку и стоимость готового продукта.

Все эти и другие проблемы требуют решения, которое на практике реализовывается наличием стойкой и унифицированной архитектуры приложения. По этой причине возникли паттерны программирования, один из которых – MVC [2]. Он использован в разработке описываемого клиент-серверного приложения для торгового-производственного предприятия.

Паттерн MVC (model-view-controller) предполагает разделение приложения на 3 уровня: данных приложения (model); обработки данных (controller); визуализации обработанных данных (view). Каждый из этих уровней можно модифицировать независимо от другого, что позволяет облегчить внесение правок и ускорить разработку.