

На первом этапе нужно создать объект ридера, указав ему ссылку на книгу (в этом случае все происходит в цикле, который проходит по папке с книгами, где находится десять тысяч книг в формате EPUB на английском языке, книги совершенно неструктурированы, разных годов, авторов и изданий.):

```
Book book = new EpubReader().  
    readEpub(new FileInputStream("path/to/folder"  
    + bookList[i].getName()));
```

После этого в переменную book помещается следующая книга и начинается ее разбор. Книга разбивается на несколько полей: ISBN книги, ФИО автора, название, издатель, краткая аннотация, год издания. В конечном итоге из книги достается весь контент. Контент представляет из себя html страницы, которые так же подвергаются разбору. Для каждой страницы генерируется ID, достается ее порядковый номер и сам контент страницы. Пример извлечения автора книги:

```
document.setField("author",  
    String.valueOf(book.getMetadata()  
    .getAuthors().get(0)));
```

Через написанный парсер было разобрано примерно десять тысяч книг. Однако, не все они соответствуют желаемой структуре. Поэтому основной задачей является разработка такой структуры, при которой из книг будет извлекаться максимально полезной информации и потери её при фильтрации будут минимальны. К примеру, не каждая книга содержит краткое описание. Если в книге оно не содержится, то будет предпринята попытка поиска его по контенту с помощью регулярного выражения. Если и таким способом описание не будет найдено, то поле остается пустым.

После проделанных операций все документы были загружены в базу данных и далее проиндексированы средствами платформы полнотекстового поиска Solr.

**В.А. Прохоренко** (УО «ГГУ имени Ф. Скорины», Гомель)  
Науч. рук. **В.С. Смородин**, д-р техн. наук, профессор

## **ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МЕТОДОВ В ЗАДАЧЕ ПОИСКА ПУТИ**

Целью работы было исследование возможности применения нейросетевых архитектур для решения задач поиска пути. В качестве примера задачи поиска пути рассмотрена задача поиска выхода из лабиринта. В процессе работы были поставлены и решены следующие задачи:

- разработка действующей модели нейронной сети, а также программная реализация алгоритмов обучения;
- формализация задачи и приведение входных данных к виду, в котором они могут быть обработаны нейронной сетью;
- подготовка обучающего множества и обучение нейросети.

Модели нейронных сетей реализованы на языке C++ в среде Visual Studio. Задача поиска пути представлена как последовательное перемещение управляемого объекта по полю, на котором имеются непроходимые участки. Перемещение происходит до тех пор, пока не будет достигнут выход.

Нейронная сеть принимает в качестве входных данных контекстное окно, состоящее из соседних с управляемым объектом клеток, а также текущее направление движения. Выходные сигналы нейронной сети определяют направление следующего движения объекта.

Все нейронные сети обучались с использованием метода сопряжённых градиентов. Производные вычислялись с помощью обратного распространения ошибки. Коррекции настраиваемых параметров осуществлялись после предъявления в произвольном порядке всех 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.

В рассматриваемой задаче были протестированы две нейросетевые архитектуры: однонаправленная полносвязная нейронная сеть типа многослойный персептрон и рекуррентная нейронная сеть на базе многослойного персептрона. Схемы нейронных сетей приведены на рисунок 1.

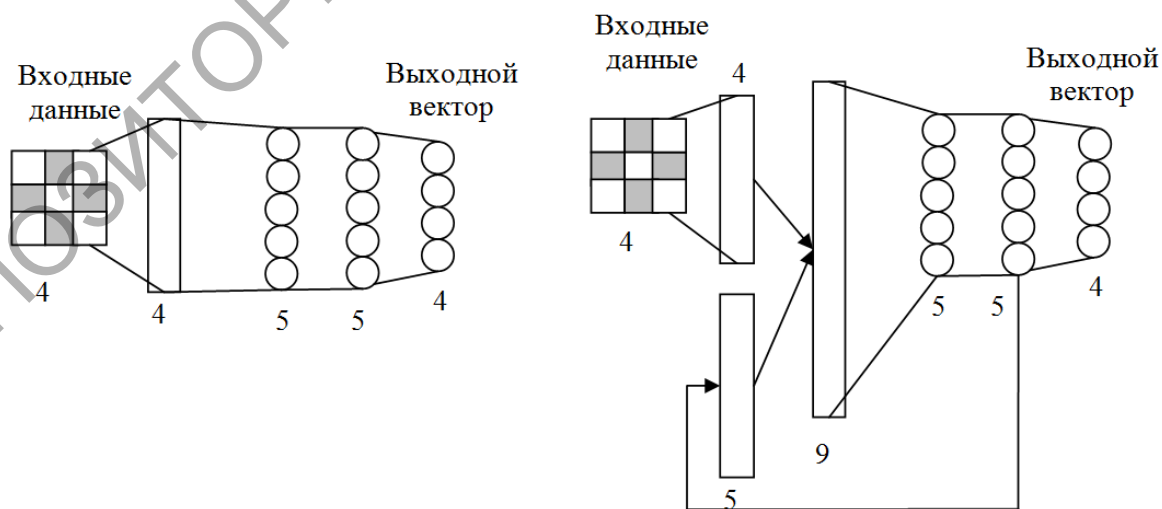


Рисунок 1 – схемы использованных нейронных сетей.

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

В силу своей структуры, многослойный персептрон не может принимать во внимание входные и выходные данные в предыдущие моменты времени. По этой причине нейронная сеть такого типа может быть успешно применена в рассматриваемой задаче в очень ограниченном наборе ситуаций и плохо подходит для её решения.

Рекуррентная сеть характеризуется наличием связей между слоями в форме элементов запаздывания. Это позволяет сети накапливать память о своих предыдущих состояниях и воспроизводить последовательности реакций. Для обучения рекуррентной нейронной сети использованы последовательности из 2 и 3 ходов, что позволило успешно находить путь с её помощью в гораздо более широком кругу задач.

Примеры нахождения пути с помощью рекуррентной нейросети по описанной методике приведены на рисунке 2.

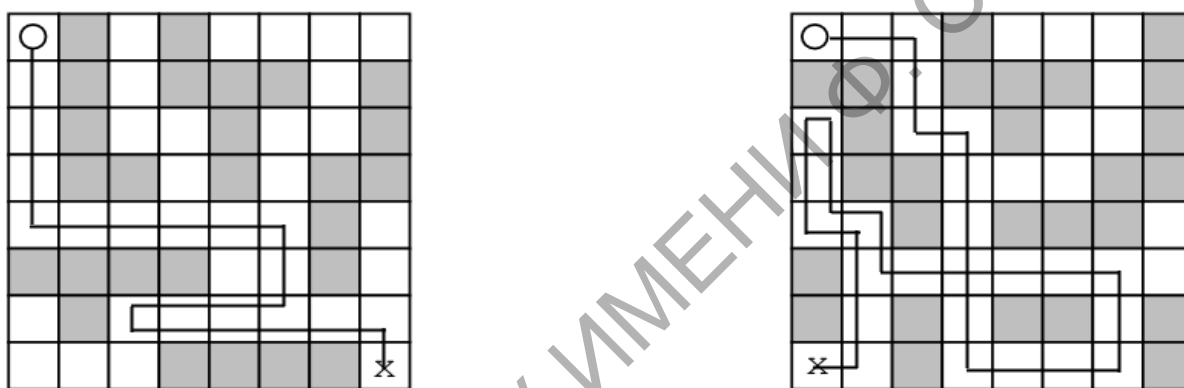


Рисунок 2 – Примеры нахождения пути с использованием рекуррентной нейросети

Проведенные эксперименты показали, что рекуррентная нейронная сеть может успешно применяться в задачах нахождения пути. Эффективность применения этого метода может быть повышена, если включить в обучающее множество последовательности из большего количества ходов.

**С.О. Рачкова** (УО «ГГУ имени Ф. Скорины», Гомель)

Науч. рук. **А. И. Кучеров**, старший преподаватель

## **РАСЧЕТ МУЛЬТИСЕРВИСНОЙ СЕТИ СПРОЕКТИРОВАННОЙ, ПО ТЕХНОЛОГИИ PON**

Расчет сети PON с известной топологией и заданными параметрами компонентов достаточно прост: чем больше запас системы, тем больше и дольше можно подключать абонентов без серьезных изменений в сети. Когда же запас будет исчерпан, следует пересчитать параметры разветвителей с целью минимизации разброса. Именно поэтому