

**Министерство образования Республики Беларусь**

**Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»**

---

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ**

**Кафедра интеллектуальных информационных технологий**

**Отчёт по лабораторной работе №5  
по курсу «МРЗвИС» на тему:  
«Предсказание числовых последовательностей  
нейросетевыми методами»**

Выполнил студент группы 921703:

**Кравцов Михаил Сергеевич**

Проверил:

**Бруцкий Дмитрий Сергеевич**

**Минск 2021**

**Цель:** Ознакомиться, проанализировать и получить навыки реализации модели нейронной сети для задачи предсказания числовых последовательностей.

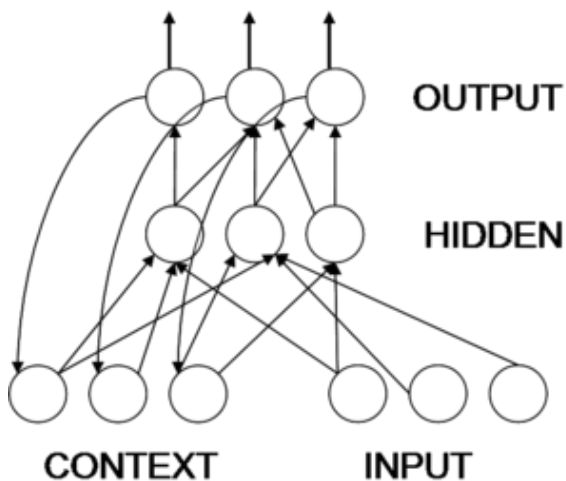
### Вариант: 11

**Задача:** Реализовать модель сети Джордана с недовыпрямленной линейной функцией активации (Leaky ReLU).

### Справочная информация

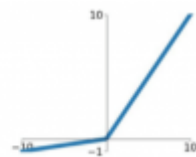
Сеть Джордана — вид нейронных сетей, который получается из многослойного перцептрона, если на его вход подать, помимо входного вектора, выходной с задержкой на один или несколько тактов.

Сигнал поступает на группу рецепторов соединенных с внешним миром (INPUT) и проходит в скрытый слой (HIDDEN). Преобразованный скрытым слоем сигнал пойдет на выходной слой (OUTPUT) и выйдет из сети, а его копия попадет на задержку. Далее в сеть, на рецепторы, воспринимающие внешние сигналы, поступает второй образ, а на контекстную группу рецепторов (CONTEXT) — выходной образ с задержки. Далее со всех рецепторов сигнал пойдет в скрытый слой, затем на выходной.



В качестве линейной функции активации использовалась Leaky ReLU.

**Leaky ReLU**  
 $\max(0.1x, x)$



### Описание задачи

Задана числовая последовательность:  $(x_0, \dots, x_q)$ , где  $x_i = f(t+i \cdot h)$ . Реализованная модель должна после обучения на выборке из  $L = q-p$  образов  $(x_k, \dots, x_{k+p})$ , где  $p < q$  и  $k = 0, \dots, q-p-1$ , эталонными значениями для которых являются  $x_{k+p+1}$ , должна обеспечивать прогнозирование  $p+1$ -го значения ( $i > 1$ ), для произвольной последовательности из  $p+1$  значений.

**N** – количество прогнозируемых значений;  
**I** – максимальное число итераций;  
**E** – максимальная допустимая среднеквадратическая ошибка;  
**M** – размер окна;  
 **$\alpha$**  – коэффициент обучения

### Результаты выполнения

- Кол-во нейронов скрытого слоя – 2
- Размер окна – 4
- Шаг обучения – 0,0005
- Максимальная допустимая ошибка – 0,000005
- Максимальное кол-во итераций – 1000000

Последовательность {1, 2, 3, 5, 8, 13, 21}

```
input -> [1.0, 2.0, 3.0, 5.0, 8.0, 13.0, 21.0]

W1 ->
0.02394813536899259    -0.05879287095120023
0.2701910760138529    0.5818477172840577
0.4667230121952157    0.20161695062320167
0.37729291073524734    0.6807217884428467
0.42509874568751416    0.5674245735510569
-0.7683902866580602    0.33590555008259665

W2 ->
0.8282650321839856
0.7550482203057072
0.2600256703983299

result -> [1.0, 2.0, 3.0, 5.0, 8.0, 13.0, 21.0,
34.122787883107925, 55.338384609199586, 89.8484113145515, 145.8466614062646, 236.8343219713177]
```

Последовательность {1, 4, 9, 16, 25, 36, 49, 64}

```
input -> [1.0, 4.0, 9.0, 16.0, 25.0, 36.0, 49.0, 64.0]

W1 ->
-2.6511574408467573    2.27051060906979
0.07615904363852283    2.4830539829711467
5.7208299554350175    -2.827314287971944
12.204932472880401    -12.295422910339777
-7.296705217892129    -1.1175510907958044
1.8974660534509116    -2.0589238808584542

W2 ->
-0.14665753121565836
-1.814938174440113
2.3244264010510056

result -> [1.0, 4.0, 9.0, 16.0, 25.0, 36.0, 49.0, 64.0,
80.76017316774129, 99.01289530271225, 118.33239710845369, 138.14690095753588, 157.75181994041333]
```

Последовательность {1, 2, 4, 8, 16, 32, 64}

```
input -> [1.0, 2.0, 4.0, 8.0, 16.0, 32.0, 64.0]

W1 ->
-0.025492384948162836   0.5066151795018292
0.23780024770089295    -1.6572706005746665
1.4595379626254852     -1.4916215961472932
3.179295795244216      -4.44582486262931
0.3323929692914114     -0.3886895705446309
-0.5580035494315634    -0.42714043188331613

W2 ->
0.9274867255031981
0.46555565953320865
0.5369777536571139

result -> [1.0, 2.0, 4.0, 8.0, 16.0, 32.0, 64.0,
128.00048315270973, 256.00155538332933, 512.005055272191, 1024.013359225055, 2048.0342671194408]
```

### Алгоритм обучения рекуррентной нейронной сети

1. В начальный момент времени  $t=1$  все контекстные нейроны устанавливаются в нулевое состояние, т.е. их выходные значения равняются нулю. Инициализация весовых коэффициентов значениями равномерно распределённой (псевдо) случайной величины. Получение максимально допустимого значения среднеквадратической ошибки. Получение обучающей выборки.
2. Входной образ подаётся на вход и происходит прямое распространение его в сети
3. В соответствии с алгоритмом обратного распространения ошибки производится модификация весовых коэффициентов и пороговых значений нейронных элементов.
4. Устанавливается  $t=t+1$  и осуществляется переход к шагу 2.

### Вывод

В ходе лабораторной работы была реализована модель сети Джордана с недовыпрямленной линейной функцией активации (Leaky ReLU). Практически были предсказаны неизвестные элементы таких числовых последовательностей, как «числа Фибоначчи», «числа Белла» и т.д. Рекуррентные нейронные сети имеют синаптические соединения между выходами нейронных элементов последующих слоев с нейронами предшествующих слоёв. Такие сети применяются для обработки динамических данных и временных образов, прогнозирования и идентификации систем, распознавания речи, видео, обработки естественного языка.