

Министерство образования Республики Беларусь Учреждение  
образования

«Белорусский государственный университет информатики и  
радиоэлектроники»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе

№3 по курсу:

«Модели решения задач в интеллектуальных системах»

Вариант №14

Выполнил студент группы 021702:

Латышев А.Т.

Проверил:

Жук А.А

МИНСК 2022

## Цель

Ознакомиться, проанализировать и получить навыки реализации модели рекуррентной нейронной сети.

## Постановка задачи

Реализовать модель сети Джордана с линейной функцией активации.

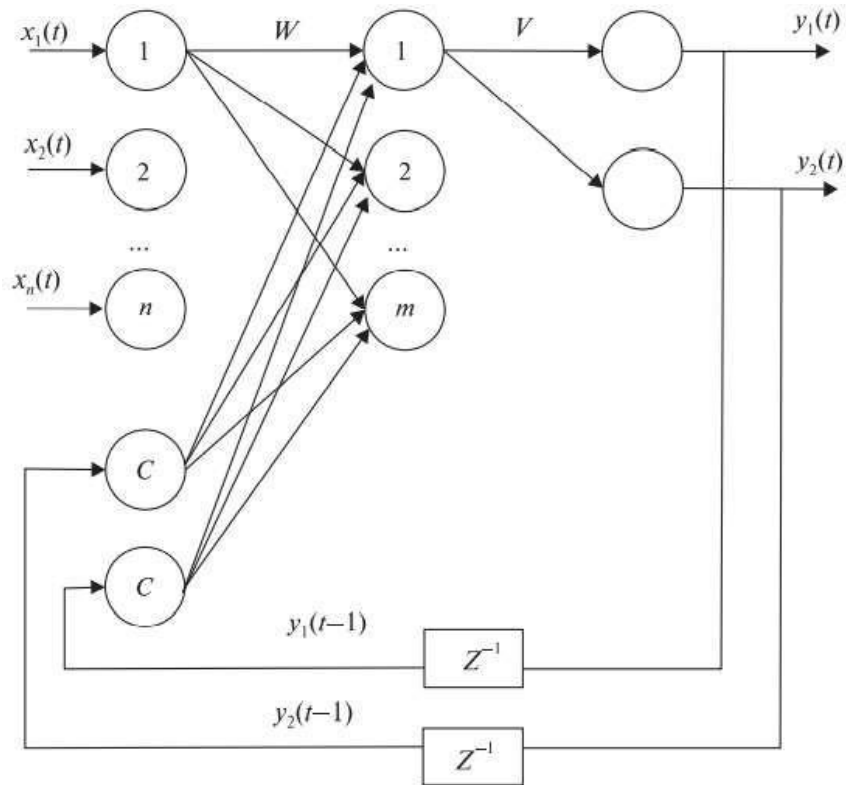
## ОПИСАНИЕ МОДЕЛИ

Данные:

sequence – исходная последовательность;  
resSequence – выходная последовательность;  
expSequence – ожидаемая последовательность;  
input\_size – размер входного вектора;  
in\_size – размер полного входного вектора;  
hidden\_size – размер скрытого слоя;  
e – максимально допустимая ошибка;  
alpha – коэффициент обучения;  
I – количество итераций;  
size- размер предсказываемой последовательности;

input – входной вектор;  
hidden - вектор скрытого слоя;  
output – выходной вектор;  
context – контекстный слой;  
in – полный входной вектор (объединение input и context);  
W1 – матрица весов на скрытом слое in\_size x hidden\_size;  
W2 - матрица весов на выходном слое hidden\_size x 1;

Сеть Джордана - рекуррентная сеть в которой выходы нейронных элементов последнего слоя посредством специальных входных нейронов соединены с нейронами промежуточного слоя



Входные нейроны, которые используются для организации обратных связей, называются *контекстными* (context units). Они распределяют выходные данные нейронной сети на нейронные элементы промежуточного слоя (рис. 4.2).

Количество контекстных нейронов равно числу выходных нейронных элементов рекуррентной сети. В качестве выходного слоя таких сетей можно использовать нейроны с линейной функцией активации. Выходное значение  $j$ -го нейронного элемента последнего слоя в этом случае вычисляется как

$$y_j(t) = \sum_{i=1}^m v_{ij} p_i(t) - T_j, \quad (4.2)$$

где  $v_{ij}$  – весовой коэффициент между  $i$ -м нейроном промежуточного и  $j$ -м нейроном выходного слоя;  $p_i(t)$  – выходное значение  $i$ -го нейрона промежуточного слоя;  $T_j$  – пороговое значение  $j$ -го нейрона выходного слоя.

Взвешенная сумма  $i$ -го нейрона промежуточного слоя определяется следующим выражением:

$$S_i(t) = \sum_{k=1}^n w_{ki} x_k(t) + \sum_{j=1}^p w_{ji} y_j(t-1) - T_i, \quad (4.3)$$

где  $w_{ki}$  – весовой коэффициент между  $k$ -м нейроном входного и  $i$ -м нейроном скрытого слоя;  $T_i$  – пороговое значение  $i$ -го нейрона скрытого слоя,  $n$  – размерность входного вектора;  $p$  – количество нейронов выходного слоя;  $w_{ji}$  – весовой коэффициент между  $j$ -м контекстным нейроном и  $i$ -м нейроном скрытого слоя.

Тогда выходное значение  $i$ -го нейрона скрытого слоя равно

$$p_i(t) = F(S_i(t)). \quad (4.4)$$

В качестве функции активации использовалась линейная функция активации

## 1. Результаты

Тестирование производилось на трех последовательностях, но так же можно вести свою :

```
1) 1 1 2 3 5 8 13 21 34 55
2) 1 -1 1 -1 1 -1 1 -1 1 -1
3) 1 0 -1 0 1 0 -1 0 1 0 -1
4) other
sequence_type: |
```

После выбора последовательности, вводится размер исходной последовательности. Предлагается выбор стандартных параметров, или ввести самому, а также ввести число предсказываемых значений.

Фибоначчи:

```
task(LEARN, TEST, SEQUENCE, EXIT): S
size: 100
sequence: [1.0, 1.0, 2.0, 2.9998927, 4.999769, 7.9994307, 12.998843, 20.997686,
33.99559, 54.991753, 88.98488, 143.97263, 232.95103, 376.91324, 609.84735, 986.7
333, 1596.5364, 2583.1978, 4179.618, 6762.628, 10941.942, 17704.078, 28645.227,
46348.02, 74991.17, 121335.83, 196321.56, 317648.56, 513955.88, 831581.5, 134550
0.1, 2177021.2, 3522423.5, 5699287.0, 9221455.0, 1.4920328E7, 2.4141112E7, 3.906
0356E7, 6.319972E7, 1.02257248E8, 1.65452384E8, 2.67702208E8, 4.3314256E8, 7.008
2534E8, 1.13393651E9, 1.83471104E9, 2.96856525E9, 4.8031432E9, 7.7714924E9, 1.25
742868E10, 2.0345215E10, 3.29185874E10, 5.3262328E10, 8.6178529E10, 1.39436982E1
1, 2.25609253E11, 3.6503611E11, 5.9062891E11, 9.5563848E11, 1.54622453E12, 2.501
79381E12, 4.04790628E12, 6.5495189E12, 1.05971314E13, 1.71461732E13, 2.77425359E
13, 4.4887467E13, 7.2627998E13, 1.17512201E14, 1.90134931E14, 3.07638575E14, 4.9
7759732E14, 8.0537603E14, 1.30309965E15, 2.10841716E15, 3.41142219E15, 5.5196863
E15, 8.9308616E15, 1.44501469E16, 2.3380361E16, 3.7829462E16, 6.120812E16, 9.903
4833E16, 1.60238495E17, 2.5926613E17, 4.1949296E17, 6.7874021E17, 1.09820273E18,
1.77689367E18, 2.87501675E18, 4.6517814E18, 7.5265903E18, 1.2178034E19, 1.97040
77E19, 3.1881224E19, 5.1583865E19, 8.3462775E19, 1.3504289E20, 2.1849963E20, 3.5
35327E20, 5.7201647E20]
```

Сеть обучилась за 105 итераций.

Полученные результаты близки к эталонным значениям (имеют погрешность), для обучения сети потребовалось небольшое количество итераций для достижения заданной ошибки.

Последовательность 1 -1 1 -1 ...:

```
task(LEARN, TEST, SEQUENCE, EXIT): S
size: 100
sequence: [1.0, -1.0, 1.0, -0.999634, 0.99977297, -0.9992155, 0.99961865, -0.998
6968, 0.9996028, -0.9979874, 0.9998501, -0.9969151, 1.0005983, -0.9951521, 1.002
2997, -0.9920746, 1.0058144, -0.9864957, 1.0127802, -0.9761565, 1.0263126, -0.95
675886, 1.052341, -0.9201238, 1.1021482, -0.85068697, 1.1972046, -0.7188307, 1.3
783668, -0.46819443, 1.723382, 0.0084713325, 2.3801975, 0.9152549, 3.6303468, 2.
640522, 6.0095663, 5.923303, 10.537325, 12.16992, 19.153597, 24.056503, 35.55001
, 46.67553, 66.75147, 89.71762, 126.12587, 171.62332, 239.11127, 327.48364, 454.
1146, 624.0742, 863.2505, 1188.4641, 1641.8069, 2262.4565, 3123.3425, 4306.186,
5942.598, 8195.254, 11307.437, 15595.869, 21516.332, 29678.7, 40943.12, 56477.29
7, 77910.88, 107473.086, 148257.84, 204514.34, 282123.0, 389176.8, 536858.75, 74
0576.1, 1021602.25, 1409263.5, 1944033.9, 2681726.8, 3699354.2, 5103132.0, 70396
01.0, 9710890.0, 1.3395847E7, 1.8479118E7, 2.549132E7, 3.5164416E7, 4.8508124E7,
6.6915328E7, 9.230744E7, 1.27335E8, 1.75654368E8, 2.42309232E8, 3.34257472E8, 4
.61096896E8, 6.3606765E8, 8.7743392E8, 1.21039027E9, 1.66969293E9, 2.30328474E9,
3.17730432E9, 4.3829837E9]
```

Сеть обучилась за 175 итераций.

Из полученных результатов только 20 спрогнозированное значение близко к эталонному оставшиеся значения постепенно накапливают ошибку.

Последовательность 1 0 -1 0 1 0 -1 0 ...:

```
size: 100
sequence: [1.0, 0.0, -1.0, 2.5324523E-5, 0.9999896, -5.069375E-5, -0.99997926, 7.607788E-5, 0.9999688, -1.01424754E-4, -0.99995846, 1.2675673E-4, 0.9999481, -1.5206635E-4, -0.9999377, 1.7736107E-4, 0.9999273, -2.0270795E-4, -0.999917, 2.2812188E-4, 0.9999066, -2.533719E-4, -0.9998962, 2.7862936E-4, 0.9998858, -3.0397624E-4, -0.9998753, 3.29338E-4, 0.99986494, -3.546551E-4, -0.99985456, 3.7998706E-4, 0.9998442, -4.0534884E-4, -0.9998338, 4.305914E-4, 0.99982345, -4.5588613E-4, -0.9998131, 4.811883E-4, 0.9998027, -5.0646067E-4, -0.9997923, 5.317405E-4, 0.9997819, -5.5699795E-4, -0.9997715, 5.823076E-4, 0.9997611, -6.0759485E-4, -0.99975073, 6.329566E-4, 0.99974036, -6.5828115E-4, -0.99973, 6.8356097E-4, 0.9997196, -7.0878863E-4, -0.9997092, 7.3409826E-4, 0.99969876, -7.593408E-4, -0.99968845, 7.846728E-4, 0.999678, -8.099526E-4, -0.9996675, 8.352399E-4, 0.99965703, -8.6060166E-4, -0.9996467, 8.8590384E-4, 0.99963635, -9.1119856E-4, -0.999626, 9.364262E-4, 0.99961567, -9.61788E-4, -0.9996053, 9.870902E-4, 0.9995949, -0.0010123774, -0.9995845, 0.0010376424, 0.999574, -0.0010629371, -0.9995635, 0.0010881945, 0.9995531, -0.0011135414, -0.99954265, 0.0011388883, 0.99953216, -0.0011642203, -0.9995218, 0.0011895746, 0.9995114, -0.0012148917, -0.999501, 0.001240179, 0.99949056]
```

Сеть обучилась за 2133 итераций.

Полученные результаты близки к эталонным значениям.

Вывод :

В ходе выполнения лабораторной работы была реализована модель сети Джордана. Было установлено на основе экспериментальных данных, что для различных числовых последовательностей варьируется необходимое количество шагов обучения нейронной сети для достижения максимально-допустимой ошибки. Также было установлено, что в последовательностях сложных для предсказания выход сети отличается от эталонного значения на большую величину, чем в более простых последовательностях.