Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2
по курсу «Методы решения задач в интеллектуальных системах»
на тему: «Предсказание числовых последовательностей нейросетевыми методами»

Выполнил студент группы 421702: Бруцкий Д.С.

Проверил: Ивашенко В.П.

Минск, 2016

Цель

Ознакомиться, проанализировать и получить навыки реализации модели нейронной сети для задачи предсказания числовых последовательностей. Реализовать модель нейронной сети Джордана с функцией активации гиперболический тангенс.

Используемые обозначения

N — размер окна
L — размер обучающей выборки
sequence — числовая последовательность
step — коэффициент обучения
iterations — количество итераций обучения
predicted — количество предсказанных элементов последовательности

Описание модели

Обучающая выборка формируется методом скользящего окна с шагом в единицу.

Для реализации обратной связи нейронная сеть модифицирует входной вектор сигналов. Поэтому входной вектор обучающей выборки создается размером N + 1.

Гиперболический тангенс — ограниченная функция, с областью значений (-1,1). Поэтому для предсказания последовательностей с более широкой областью значений к ней применяется линейное преобразование:

```
public double normalize(double value){
    return 2 * (value - min) / (max - min) - 1;
}
```

Обратное преобразование:

```
public double restore(double value) {
    return min + (value + 1) * (max - min) / 2;
}
```

Реализация

Модель реализована на языке программирование java. Для проведения операций с матрицами используется библиотека jama.

Прямое распространение сигнала

```
public Matrix straightPropagation(Matrix vector) {
    if (vector.getColumnDimension() != inputSize + contextSize + 1) {
        throw new IllegalArgumentException("{{<vector><place to threshold><place
to context signals>}}, " +
                "expected size " + (inputSize + contextSize) + ", result is " +
vector.getColumnDimension());
    }
    vector.set(0, inputSize, 1.);
    vector.setMatrix(0, 0, inputSize + 1, inputSize + 1 + contextSize - 1,
context);
    input = vector.copy();
    firstSynapticOutput = input.times(firstW);
    firstActivationOutput = applyFunction(firstSynapticOutput);
    firstActivationOutput.set(0, hiddenLayerSize, 1.);
    secondSynapticOutput = firstActivationOutput.times(secondW);
    secondActivationOutput = applyFunction(secondSynapticOutput);
    if (!contextFreeze) context = secondActivationOutput.copy();
    return secondActivationOutput;
}
```

Корректировка весов

```
public void backPropagate(Matrix outputDifference, double teachingStep) {
    Matrix secondActivationDerivation = new Matrix(1, outputLayerSize, 1.)
            .minus(secondActivationOutput.arrayTimes(secondActivationOutput));
    Matrix hiddenLayerDifference = outputDifference
            .arrayTimesEquals(secondActivationDerivation)
            .times(secondW.transpose());
    Matrix firstActivationDerivation = new Matrix(1, hiddenLayerSize + 1, 1.)
            .minus(firstActivationOutput.arrayTimes(firstActivationOutput));
    Matrix firstWeightsCorrection = input
            .transpose()
            .times(hiddenLayerDifference.arrayTimes(firstActivationDerivation))
            .times(teachingStep);
    firstW = firstW.minusEquals(firstWeightsCorrection);
    Matrix secondWeightsCorrection = firstActivationOutput
            .transpose()
            .times(outputDifference.arrayTimes(secondActivationDerivation))
            .times(teachingStep);
    secondW = secondW.minusEquals(secondWeightsCorrection);
}
```

Периодическая функция

Числовая последовательность генерируется функцией

$$y = \cos(0.5 * x), x \in N \cup 0$$

N = 10, L = 38, step = 0.001, iterations = 1,000,000, predicted = 20

Ожидаемая последовательность	Предсказанная последовательность	Линейная ошибка	Относительная ошибка, %
-0.061905294	-0.0663200298	-0.0044147358	0.0713143507
0.4241790073	0.4237230526	-0.0004559547	0.0010749112
0.8064094939	0.812636166	0.0062266721	0.0077214766
0.9912028119	0.9742364938	-0.016966318	0.0171168986
0.9333151121	0.9319739126	-0.0013411994	0.0014370274
0.6469193223	0.6430558097	-0.0038635126	0.0059721706
0.2021351204	0.2029216389	0.0007865185	0.0038910531
-0.2921388087	-0.3064015365	-0.0142627278	0.0488217496
-0.7148869688	-0.7016733822	0.0132135866	0.0184834627
-0.9626058663	-0.9586530887	0.0039527777	0.0041063303
-0.9746452757	-0.9595055658	0.0151397099	0.015533559
-0.7480575297	-0.7493419646	-0.0012844349	0.0017170269
-0.338319211	-0.3206099053	0.0177093056	0.0523449602
0.1542514499	0.1404144944	-0.0138369555	0.0897038924
0.6090559761	0.5973062963	-0.0117496798	0.0192916254
0.9147423578	0.9247070267	0.0099646689	0.0108934158
0.9964679076	0.9715931391	-0.0248747684	0.0249629398
0.8342233605	0.8397729955	0.005549635	0.006652457
0.4677318402	0.4793893292	0.011657489	0.0249234454
-0.0132767472	0.0170386864	0.0303154336	2.2833479557
-0.4910347239	-0.4917463027	-0.0007115788	0.0014491415

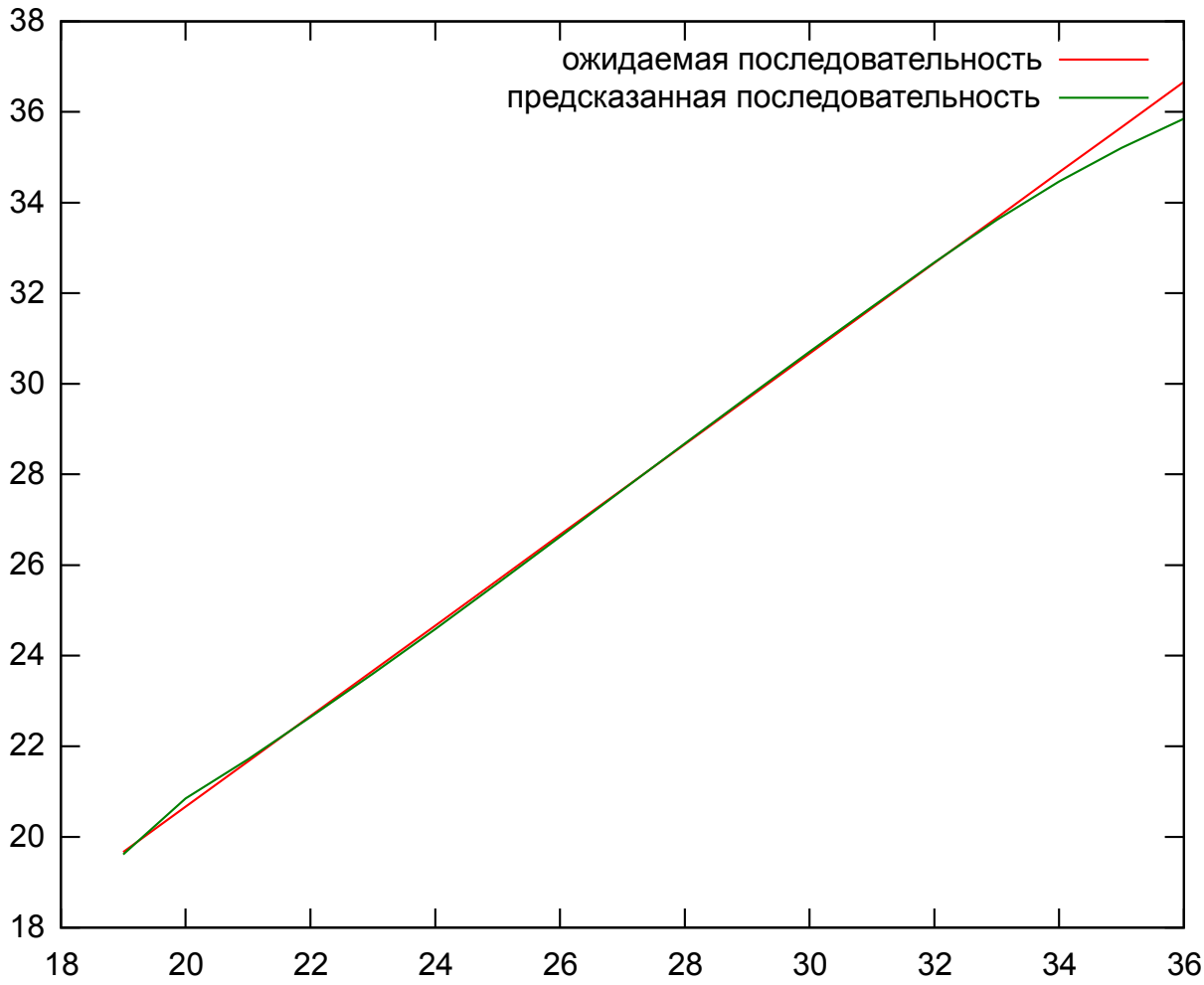
Линейная функция

Числовая последовательность генерируется функцией

$$y = x, x \in N \cup 0$$

N = 3, L = 20, step = 0.001, iterations = 1,000,000, predicted = 5

#	Ожидаемая последовательность	Предсказанная последовательность	Линейная ошибка	Относительная ошибка, %
32	32.6666666667	32.6848364677	0.018169801	0.0005562184
33	33.6666666667	33.6205208004	-0.0461458663	0.0013706693
34	34.6666666667	34.4646577454	-0.2020089213	0.0058271804
35	35.6666666667	35.2039211759	-0.4627454908	0.0129741726
36	36.6666666667	35.8522934606	-0.8143732061	0.0222101783



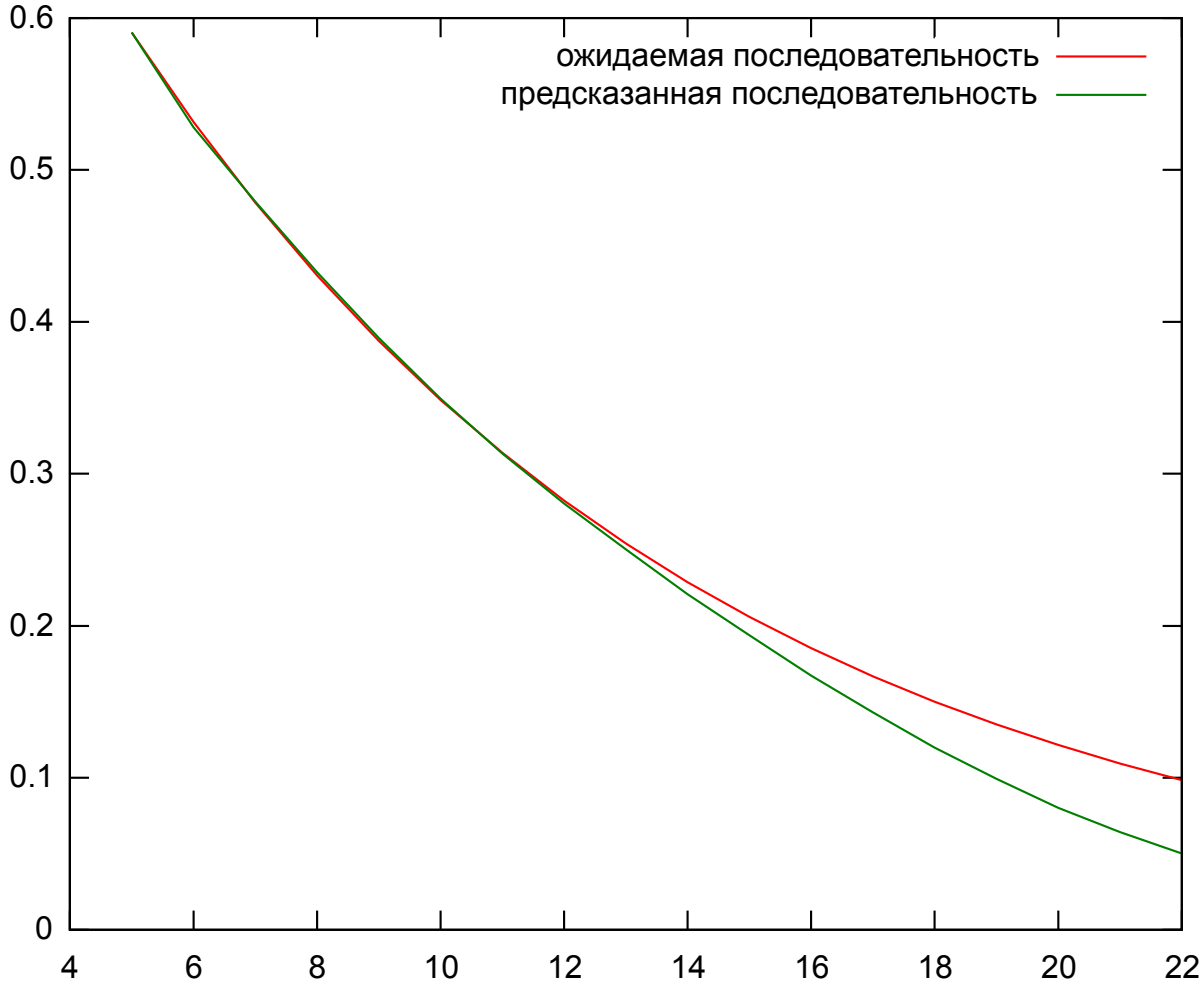
Степенная функция

Числовая последовательность генерируется функцией

$y = 0.9^x, x \in N \cup 0$

N = 5, L = 13, step = 0.0002, iterations = 1,000,000, predicted = 10

#	Ожидаемая последовательность	Предсказанная последовательность	Линейная ошибка	Относительная ошибка, %
13	0.2541865828	0.2503286758	-0.0038579071	0.0151774614
14	0.2287679245	0.2208471111	-0.0079208135	0.0346237939
15	0.2058911321	0.1937206004	-0.0121705317	0.0591114907
16	0.1853020189	0.1672754887	-0.0180265302	0.0972818876
17	0.166771817	0.1430484564	-0.0237233606	0.1422504175
18	0.1500946353	0.1198656419	-0.0302289934	0.2013995594
19	0.1350851718	0.0992269984	-0.0358581734	0.2654486269
20	0.1215766546	0.0801660677	-0.0414105869	0.3406129824
21	0.1094189891	0.0641524717	-0.0452665174	0.4136989179
22	0.0984770902	0.0501260914	-0.0483509988	0.4909872814


ожидаемая последовательность
предсказанная последовательность
0.6
0.5
0.4
0.3
0.2
0.1
0
4
6
8
10
12
14
16
18
20
22

Вывод

В данной работе реализована модель сети Джордана с функцией активации гиперболический тангенс.

Для предсказания функций с областью значений большей (-1, 1), к последовательности нужно применить преобразование описанное выше.

Сеть хорошо предсказывает периодические функции. Также сеть может предсказать убывающие последовательности. Последовательности возрастающие быстрее O(n^2) предсказываются только на очень малом интервале.