

# Лабораторная работа №4

## Рекуррентная нейронная сеть

Цель работы: изучить обучение и функционирование РНС/МРНС при решении задач прогнозирования.

### 1. Теория

Весь необходимый теоретический материал можно найти в лекционных материалах или в [данной книге](#) в главе 4.

### 2. Использование линейной нейронной сети для прогнозирования

Изучение способности к прогнозированию линейной нейронной сети будет производиться на периодический функциях. Весь процесс работы должен быть разделен на следующие этапы:

1. Подготовка исходных данных
2. Разделение исходных данных на обучающую и тестовую выборки
3. Подготовка данных для линейной нейронной сети согласно Вашему варианту
4. Обучение и тестирование линейной нейронной сети
5. Вывод результатов настроенной модели

#### 2.1 Подготовка исходных данных

Для функции из Вашего варианта сперва вычислите период. Затем необходимо выполнить табуляцию данных таким образом, чтобы в неё попал минимум 1 период. Можете ограничиться 45-ю значениями. В итоге у вас должен получиться исходный набор данных, который будет использоваться для обработки Вашей нейросетевой моделью.

Примечание 1: для входа нейросетевой модели эти данные имеют обозначение  $x_i$ , где  $i$  – порядковый номер образа.

Примечание 2: для вычисления ошибки сети эти данные имеют обозначение  $e_1^k$ , где  $k$  – порядковый номер образа.

Примечание 3: для вычисления весовых коэффициентов нейронной сети эти данные имеют обозначение  $x_i^k$ , где  $i$  – порядковый номер нейрона,  $k$  – порядковый номер образа.

#### 2.2 Разделение исходных данных на обучающую и тестовую выборки

Для полученных исходных данных из пункта 2.1 выполните разделение по следующему принципу: 2/3 данных выделите для обучения, 1/3 для проверки, т.е. тестирования. Если в результате табуляции у вас получилось 45 исходных значений функции, тогда значения с 1 по 30-е будут использоваться для обучения, а с 31-45-е для тестирования. С учетом задачи прогнозирования, для тестирования будут использоваться данные с 28-45-е значения, т.к. Вашей задачей прогнозирования будет продолжение поиска значений из обучающего набора, а ввиду того что вам необходимо отправить на вход 3 значения, чтобы получить 1 выходное, вы будете подавать на вход 3 последних значения из обучающего набора, чтобы получить 1-е значение тестового набора.

#### 2.3 Подготовка данных для линейной нейронной сети согласно Вашему варианту

Для того, чтобы Ваша модель могла обрабатывать данные согласно Вашему варианту, Вам необходимо выполнить подготовку данных для Вашей модели. Для этого необходимо знать число входных и выходных нейронов. Также, Вам необходимо знать, какие из данных будут отправлены на вход нейронной сети, а какие данные будут использоваться для сравнения полученных результатов от Вашей модели.

Если у вас есть 3 распределительных (входных) нейрона и 1 выходной нейрон (в данной работе у всех вариантов будет использоваться 1 выходной нейрон), тогда распределение будет выполнено следующим образом:

Набор данных для обучения:

$$(y_1, y_2, y_3)^1 \rightarrow y_4^1 - 1\text{-й входной набор}$$

$$(y_2, y_3, y_4)^2 \rightarrow y_5^2 - 2\text{-й входной набор}$$

...

$$(y_{57}, y_{58}, y_{59})^{57} \rightarrow y_{60}^{57} - 57\text{-й входной набор}$$

Набор данных для тестирования:

С учетом особенностей прогнозирования, в которой вы знаете только начальные исходные данные и не знаете последующие, необходимо учесть это при построении. Соответственно, заранее можно подготовить только 1-й входной набор и значения для проверки результатов прогнозирования. Начиная со 2-го набора процесс подачи входных образов для тестирования необходимо делать динамически, относительно результатов прогнозирования для предыдущего набора.

$$(y_{58}, y_{59}, y_{60})^1 \rightarrow y_{61}^1 - 1\text{-й входной набор}$$

$$(y_{59}, y_{60}, y_{61}')^2 \rightarrow y_{62}^2 - 2\text{-й входной набор, где } y_{61}' - \text{предсказанное значение в 1-м входном наборе}$$

...

$$(y_{87}', y_{88}', y_{89}') \rightarrow y_{90}^{30} - 30\text{-й входной набор}$$

#### 4. Обучение и тестирование линейной нейронной сети

Цикл, в котором происходит обработка всех значений из набора называется эпохой. В течении эпохи, вычисляется выходное значение для каждого из наборов (из пункта 2.3), сравнивается с эталонным значением по формуле 1.3. В итоге, в конце эпохи должна быть вычислена итоговая среднеквадратичная ошибка для обучения и итоговая среднеквадратичная ошибка для тестирования (формула 1.2). Последняя сравнивается с заданной минимальной ошибкой для проверки остановки обучения.

#### 5. Вывод результатов настроенной модели

Осуществляется для последней эпохи, в которой значение общей среднеквадратичной ошибки тестирования было меньше заданного значения. Необходимо выводить значения аналогично значению наборов из пункта 2.3 для обучения и тестирования, чтобы были явно обозначены значения, которые Вы подаете, значения, которые Вы получаете в результате работы нейронной сети и эталонные значения для данного набора. Сделать для только для тестового наборов. Пример:

$$(y_{58}, y_{59}, y_{60})^1 \rightarrow y_{61}'(y_{61}^1)$$

### Задание

Написать на любом ЯВУ программу моделирования прогнозирующей линейной ИНС. Для решения использовать архитектуру нейронной сети Элмана, Джордана или мультирекуррентную НС из главы №4. Настройки сети выбираются студентом самостоятельно, в т.ч. число входов ИНС.

Варианты заданий приведены в следующей таблице:

№ варианта	Функция
1	$f(x) = \sin(x) + \cos(2x) + \sin(3x)$
2	$f(x) = \cos(x) + \frac{\sin(4x)}{2} - \cos(3x)$
3	$f(x) = \sin(x) * \cos(\frac{x}{2}) + \cos(\frac{3x}{2})$
4	$f(x) = \sin(x) + \cos(2x) * \sin(3x)$
5	$f(x) = \sin(2x) + \cos(x)^2 * \sin(x)$
6	$f(x) = \cos(3x) * \sin(x) + \sin(2x)$
7	$f(x) = \sin(5x) * \cos(4x) + \cos(x)$
8	$f(x) = \cos(2x) + \sin(x)^3 * \cos(3x)$
9	$f(x) = \sin(x)^2 * \cos(x) + \sin(2x)$
10	$f(x) = \cos(x) + \sin(2x) * \cos(4x)$
11	$f(x) = \cos(x) * \sin(\frac{3x}{2}) + \sin(\frac{x}{2})$

Минимальное число точек для обучения/тестирования: 60 и 30 соответственно. Функция табулируется произвольным шагом исходя из числа выбранных точек. Шаг обучения выбирается самостоятельно, для чего моделирование проводится несколько раз для разных  $\alpha$ . Результаты оцениваются по двум критериям - шагу обучения и минимальной достигнутой ошибке. Необходимо заметить, что эти критерии в общем случае являются взаимоисключающими, и оптимальные значения для каждого критерия достигаются при разных  $\alpha$ .

Образец вывода программы:

Stage 1: data preparing

```
x1 = .. ; y1 = ..
x2 = .. ; y2 = ..
...
x90 = .. ; y3 = ..
```

Stage 2: split data on train/test data

```
train_data:
x1 = .. ; y1 = ..
x2 = .. ; y2 = ..
...
x60 = ..
```

```
test_data:
x61 = .. ; y1 = ..
x62 = .. ; y2 = ..
...
x90 = ...
```

Stage 3: prepare train/test data for NN

for example for 3 input neurons:

train\_data:

y1, y2, y3 -> y4

y2, y3, y4 -> y5

...

y57, y58, y59 -> y60

test\_data (y60 -> original value, y'61 -> model output value):

y58, y59, y60 -> y'61

Stage 4: train & test model

Epoch #1

train\_loss: 0,15 \t test\_loss: 0,23

test\_loss > 1e-4 -> continue training

...

Epoch #5

train\_loss: 4.3e-5 \t test\_loss: 8.6e-5

test\_loss < 1e-4 -> stop training

Stage 5: print full model outputs for best epoch

y58, y59, y60 -> y'61 (y61)

y59, y60, y'61 -> y'62 (y62)

...

y'87, y'88, y'89 -> y'90 (y90)

Результаты представить в виде отчета содержащего:

1. Титульный лист
2. Цель работы
3. Задание
4. Вывод согласно предложенному образцу
5. Выводы по лабораторной работе.

Примечание: если в выводе будет отсутствовать информация по каким-либо этапам из образца вывода программы – работа засчитана не будет.