Лабораторная работа № 3.
Многослойные сети. Алгоритм обратного распространения ошибки

Целью работы является исследование свойств многослойной нейронной сети прямого распространения и алгоритмов ее обучения, применение сети в задачах классификации и аппроксимации функции.

Основные этапы работы:

1. Использовать многослойную нейронную сеть для классификации точек в случае, когда классы не являются линейно разделимыми.

2. Использовать многослойную нейронную сеть для аппроксимации функции. Произвести обучение с помощью одного из методов первого порядка.

3. Использовать многослойную нейронную сеть для аппроксимации функции. Произвести обучение с помощью одного из методов второго порядка.

Для обучения многослойных нейронных сетей прямого распространения используются методы поиска экстремума функций многих переменных.

Методы первого порядка:

- Метод градиентного спуска (*traingd*). Для работы алгоритма необходимо задать скорость обучения: *net.trainParam.lr = 0.05*.
- Метод градиентного спуска с моментом (*traingdm*). Для работы алгоритма необходимо задать скорость обучения и величину момента: *net.trainParam.lr = 0.05*, *net.trainParam.mc = 0.9*.
- Метод градиентного спуска с адаптивным шагом (*traingda*). Для работы алгоритма необходимо задать скорость обучения и коэффициент увеличения скорости настройки: *net.trainParam.lr = 0.05*, *net.trainParam.lr_inc = 1.05*.
- Метод градиентного спуска с адаптивным шагом и моментом (*traingdx*). Для работы алгоритма необходимо задать скорость обучения, коэффициент увеличения скорости настройки и величину момента: *net.trainParam.lr = 0.05*, *net.trainParam.lr_inc = 1.05*, *net.trainParam.mc = 0.9*.
- Метод гибкого распространения (*trainrp*). Другое название RProp (Resilient Backpropagation).
- Методы сопряженных градиентов: метод Флетчера-Ривса (*traincgf*), метод Полака-Рибейры (*traincgp*), метод Пауэлла-Биеле (*traincgb*), метод Моллера (*trainscg*). Для группы методов сопряженных градиентов рекомендуется задать число нейронов в скрытом слое равным 15.

Методы второго порядка:

- Квазиньютоновский метод, предложенный Бройденом, Флетчером, Гольдфарбом и Шанно (*trainbfg*).
- Метод Левенберга-Марквардта (*trainlm*).
- Одношаговый метод секущих (*trainoss*).

Сценарий работы:

Этап 1

1. Заданы 3 линейно неразделимых класса. Точки, принадлежащие одному классу, лежат на алгебраической линии. Построить и обучить многослойную сеть прямого распространения, которая будет классифицировать точки заданной области.

Обучающий набор $\{x_i, y_i\},\ i = 1, .., N$, число классов $K = 3$. Сеть реализует отображение вида:

$$f(x_i, y_i) = \{(z_k)_{k=1}^{K} = (0, .., 1, ..0) |\ z_{k=K^*} = 1 \text{ при } (x_i, y_i) \in K^*\}$$

1.1 В соответствии с вариантом задания для каждой линии сгенерировать множество точек. Далее для первого класса выбрать из исходного множества случайным образом 60 точек. Для второго и третьего классов 100 и 120 точек соответственно. Для выбора точек рекомендуется использовать функцию $randperm$, с помощью которой получить псевдослучайную последовательность индексов вектора.

1.2 Множество точек, принадлежащее каждому классу, разделить на обучающее, контрольное, и тестовое подмножества с помощью функции $dividerand$ в отношении 70%-20%-10%.

1.3 Отобразить с помощью функции $plot$ исходные множества точек для каждого из классов. Задать параметр $LineWidth$ равным 2, подписать линии, задать сетку. С помощью $axis$ задать границы для входного множества. Параметры отображения для классов:

Класс 1 Исходное множество: $-r$. Обучающее подмножество: or, $MarkerEdgeColor = k$, $MarkerFaceColor = r$, $MarkerSize = 7$. Контрольное подмножество: rV, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.
Тестовое подмножество: rs, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.

Класс 2 Исходное множество: $-g$. Обучающее подмножество: og, $MarkerEdgeColor = k$, $MarkerFaceColor = g$, $MarkerSize = 7$. Контрольное подмножество: gV, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.
Тестовое подмножество: gs, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.

Класс 3 Исходное множество: $-b$. Обучающее подмножество: ob, $MarkerEdgeColor = k$, $MarkerFaceColor = b$, $MarkerSize = 7$. Контрольное подмножество: bV, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.
Тестовое подмножество: bs, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.

1.4 Соответствующие подмножества точек каждого класса объединить в обучающее, контрольное, и тестовое подмножества обучающей выборки. Обучающая выборка состоит из последовательного объединения полученных обучающего, контрольного, и тестового подмножеств.

1.5 Создать сеть с помощью функции $feedforwardnet$. Сконфигурировать сеть ($configure$), указав диапазоны изменения для входного множества и эталонных выходов сети. Точки входного и выходного множеств лежат на отрезках $[-1.2, 1.2]$ и $[0, 1]$ по каждой из координат соответственно.

Число нейронов скрытого слоя задать равным 20. Использовать активационные функцию $tansig$ для скрытого и выходного слоев. Задать RProp в качестве алгоритма обучения.

1.6 Для разделения обучающего множества на подмножества использовать $net.divideFcn =' divideind'$. Также задать параметры:

$$net.divideParam.trainInd = 1 : trnInd;$$
$$net.divideParam.valInd = trnInd + 1 : tstInd;$$
$$net.divideParam.testInd = tstInd + 1 : proInd;$$

где $trnInd$, $tstInd$, $proInd$ задают количество примеров в обучающем, контрольном, и тестовом подмножествах.

1.7 Инициализировать ($init$) весовые коэффициенты и смещения сети с помощью функции, заданной по умолчанию.

1.8 Задать параметры обучения: число эпох обучения ($net.trainParam.epochs$) и число эпох, в течение которых может расти ошибка на контрольном подмножестве ($net.trainParam.max\_fail$), равными 1500, предельное значение критерия обучения ($net.trainParam.goal$) равным 10^{-5}.

1.9 Выполнить обучение сети с помощью функции $train$. Для обучения использовать обучающую выборку. Занести в отчет содержимое Performance и Neural Network Training.

1.10 Отразить структуру сети и проведенное обучение в отчете, заполнив таблицу 1.

1.11 Рассчитать выход сети (sim) для обучающего подмножества. Преобразовать значения по правилу

$$o_{ij} = \begin{cases} 1, & a_{ij} \geqslant 0.5; \\ 0, & a_{ij} < 0.5; \end{cases}$$

Занести в отчет количество правильно классифицированных точек.

1.12 Провести аналогичные расчеты для контрольного и тестового подмножеств.

1.13 Произвести классификацию точек области $[-1.2, 1.2] \times [-1.2, 1.2]$. Для этого задать сетку для указанной области с шагом $h = 0.025$. Рассчитать выход сети для всех узлов сетки.

1.14 Выход сети для каждой точки задает ее принадлежность к трем классам. Закодировать принадлежности к классам различными цветами и занести полученное изображение в отчет.

Для этого использовать функции $image$ и $colormap$. $image$ отображает матрицу, каждый элемент которой содержит ссылку на таблицу цветов, которая задается с помощью $colormap$. Элементы в матрице цветов должны находиться в диапазоне $[0, 1]$. Каждая компонента выходного вектора задает интенсивность одного из цветов в модели RGB. Например, $(1, 0, 0)$ — красный цвет, $(0, 1, 0)$ — зеленый, $(0, 0, 1)$ — голубой, $(1, 1, 0)$ — желтый цвет.

Сначала нужно сформировать таблицу цветов: округлить компоненты выходных векторов до десятых ($floor$ или $round$) и удалить повторяющиеся вектора с помощью функции $unique('rows')$. Затем каждый из выходных векторов заменить на номер строки из таблицы цветов. Для перехода от пакетного (batch) представления к матрице ссылок использовать функцию $reshape$ и операцию транспонирования.

Варианты заданий:

Номер варианта соответствует номеру в списке группы. Для генерации точек использовать параметрическое уравнение линии в канонической системе координат.

$$t = 0 : 0.025 : 2\pi$$
$$x = f(t)$$
$$y = g(t)$$

Константы a и b задают большую и малую полуоси эллипса, p — параметр параболы. Параметры преобразования прямоугольной системы координат на плоскости: угол поворота (α) и координаты параллельного переноса (x_0, y_0).

№	**Алгебраические линии**
1.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$

№	Алгебраические линии
2.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
3.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
4.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
5.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = -0.1$, $y_0 = 0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
6.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
7.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = 0.25$, $y_0 = -0.25$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
8.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = -0.25$, $y_0 = 0.25$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
9.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = -0.2$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
10.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = 0.2$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
11.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/3$, $x_0 = -0.2$, $y_0 = -0.18$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = -0.2$, $y_0 = -0.18$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
12.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0.4$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0.2$, $y_0 = 0.18$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$

№	Алгебраические линии
13.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
14.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
15.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
16.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
17.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = -0.8$, $y_0 = 0$
18.	Эллипс: $a = 0.4$, $b = 0.4$, $\alpha = 0$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
19.	Эллипс: $a = 0.4$, $b = 0.4$, $\alpha = 0$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = -1$, $\alpha = 0$, $x_0 = 0.8$, $y_0 = 0$
20.	Эллипс: $a = 0.4$, $b = 0.4$, $\alpha = 0$, $x_0 = -0.1$, $y_0 = 0.15$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = -1$, $\alpha = 0$, $x_0 = 0.8$, $y_0 = 0$
21.	Эллипс: $a = 0.5$, $b = 0.2$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0.08$, $y_0 = 0.05$ Парабола: $p = -1$, $\alpha = -\pi/2$, $x_0 = 0$, $y_0 = -0.8$
22.	Эллипс: $a = 0.5$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.8$, $b = 0.8$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
23.	Эллипс: $a = 0.4$, $b = 0.5$, $\alpha = 0$, $x_0 = 0.05$, $y_0 = 0$ Эллипс: $a = 0.6$, $b = 0.6$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.8$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$

№	Алгебраические линии
24.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.35$ Эллипс: $a = 0.5$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.35$ Эллипс: $a = 0.8$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
25.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.2$ Эллипс: $a = 0.5$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.2$ Эллипс: $a = 0.8$, $b = 1$, $\alpha = 0$, $x_0 = -0.1$, $y_0 = -0.1$
26.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 0.5$, $\alpha = 0$, $x_0 = -0.5$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
27.	Парабола: $p = 0.3$, $\alpha = 0$, $x_0 = 0.$, $y_0 = 0$ Парабола: $p = 0.5$, $\alpha = 0$, $x_0 = -0.5$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
28.	Парабола: $p = -0.3$, $\alpha = 0$, $x_0 = 0.$, $y_0 = 0$ Парабола: $p = -0.4$, $\alpha = 0$, $x_0 = 0.4$, $y_0 = 0$ Парабола: $p = -0.5$, $\alpha = 0$, $x_0 = 0.65$, $y_0 = 0$
29.	Эллипс: $a = 0.3$, $b = 0.15$, $\alpha = -\pi/6$, $x_0 = -0.05$, $y_0 = -0.05$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
30.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 0.7$, $\alpha = -\pi/3$, $x_0 = -0.2$, $y_0 = 0.4$ Парабола: $p = 1$, $\alpha = -\pi/3$, $x_0 = -0.4$, $y_0 = 0.6$

Этапы 2 и 3

2. Задан обучающий набор $\{x(i), y(i)\}$. Построить и обучить двухслойную нейронную сеть прямого распространения, которая будет выполнять аппроксимацию функции вида

$$\hat{y}(i) = f[x(i)]$$

Для обучения использовать алгоритм, реализующий метод поиска экстремума функции многих переменных первого порядка. Функция и метод обучения определяются вариантом задания.

2.1 Создать сеть с помощью функции *feedforwardnet*. Сконфигурировать сеть под обучающее множество с помощью функции *configure*. Число нейронов скрытого слоя задать равным 10. Использовать активационные функции, заданные по умолчанию (*tansig, purelin*). Алгоритм обучения определяется вариантом задания.

2.2 Для разделения обучающией выборки на обучающее, контрольное, и тестовое подмножества использовать функцию *divideind*. Выделить с конца временной последовательности 10% отсчетов на контрольное подмножество. Тестовое подмножество оставить пустым.

$$net.divideParam.testInd = []$$

2.3 Инициализировать сеть (*init*) с помощью функции, заданной по умолчанию.

2.4 Задать параметры обучения: значения параметров для некоторых методов обучения описаны выше, число эпох обучения (*net.trainParam.epochs*) и число эпох, в течение которых может расти ошибка на контрольном подмножестве (*net.trainParam.max_fail*), равными 600, предельное значение критерия обучения (*net.trainParam.goal*) равным 10^{-8},

2.5 Выполнить обучение сети с помощью функции *train*. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные или наблюдается переобучение, то изменить число нейронов в функции *feedforwardnet*, увеличить число эпох обучения или уменьшить предельное значение критерия обучения. Занести в отчет весовые коэффициенты и смещения для двух слоев. Занести в отчет окна Performance и Neural Network Training, если это возможно для данного метода обучения.

2.6 Отразить структуру сети и проведенное обучение в отчете, заполнив таблицу 1.

2.7 Рассчитать выход сети (*sim*) для обучающего подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения. Графики занести в отчет.

2.8 Проделать тоже самое для контрольного подмножества.

3. Построить и обучить двухслойную нейронную сеть прямого распространения, которая будет выполнять аппроксимацию функции. Для обучения использовать алгоритм, реализующий метод оптимизации функций многих переменных второго порядка. Функция и метод обучения определяются вариантом задания.

Последовательности шагов для выполнения 2 и 3 этапов работы совпадают.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Функция	Методы обучения
1.	$x = \sin\left(t^2\right), \quad t \in [0, 4],\ h = 0.02$	*trainrp, trainoss*
2.	$x = \sin\left(t^2 - 2t + 3\right), \quad t \in [0, 6],\ h = 0.025$	*traincgb, trainlm*
3.	$x = \cos\left(2.5t^2 - 5t\right), \quad t \in [0, 2.2],\ h = 0.01$	*trainscg, trainbfg*
4.	$x = \sin\left(\sin(t)t^2 + 5t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingd, trainbfg*
5.	$x = \cos\left(-3t^2 + 5t + 10\right), \quad t \in [0, 2.5],\ h = 0.01$	*traingdm, trainoss*
6.	$x = \sin\left(0.5t^2 - 5t\right), \quad t \in [0, 2],\ h = 0.01$	*traincgp, trainlm*
7.	$x = \sin\left(\sin(t)t^2 - t\right), \quad t \in [1, 4.5],\ h = 0.01$	*traincgp, trainbfg*
8.	$x = \sin\left(-2t^2 + 7t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingdx, trainoss*
9.	$x = \sin\left(t^2 - 2t + 5\right), \quad t \in [0, 5],\ h = 0.025$	*traingd, trainbfg*

№	T	trainFcn
10.	$x = \sin\left(t^2 - 7t\right), \quad t \in [0, 5],\ h = 0.025$	*trainscg, trainoss*
11.	$x = \cos\left(t^2\right), \quad t \in [0, 4],\ h = 0.02$	*traingda, trainbfg*
12.	$x = \cos\left(-\cos(t)t^2 + t\right), \quad t \in [0.5, 4],\ h = 0.01$	*traincgf, trainlm*
13.	$x = \cos\left(t^2 - 2t + 3\right), \quad t \in [0, 5],\ h = 0.02$	*traingdx, trainbfg*
14.	$x = \cos\left(t^2 - 10t + 3\right), \quad t \in [1, 6],\ h = 0.025$	*trainrp, trainlm*
15.	$x = \cos\left(-2t^2 + 7t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingda, trainoss*
16.	$x = \sin\left(\sin(t)t^2 + 3t - 10\right), \quad t \in [2.5, 5],\ h = 0.01$	*traincgb, trainbfg*
17.	$x = \cos\left(-5t^2 + 10t - 5\right), \quad t \in [0, 2.5],\ h = 0.01$	*trainscg, trainoss*
18.	$x = \cos\left(\cos(t)t^2 - t\right), \quad t \in [1, 4.5],\ h = 0.01$	*traincgf, trainlm*
19.	$x = \sin\left(-5t^2 + 10t - 5\right), \quad t \in [0, 2.5],\ h = 0.01$	*traingda, trainoss*
20.	$x = \cos\left(\cos(t)t^2 + 5t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingdx, trainlm*
21.	$x = \sin\left(t^2 - 10t + 3\right), \quad t \in [1, 6],\ h = 0.025$	*traingdx, trainoss*
22.	$x = \sin\left(-\sin(t)t^2 + t\right), \quad t \in [0.5, 4],\ h = 0.01$	*traincgb, trainlm*
23.	$x = \sin\left(t^2 - 6t + 3\right), \quad t \in [0, 5],\ h = 0.025$	*traingd, trainbfg*
24.	$x = \sin\left(0.66\pi t\right), \quad t \in [0, 5],\ h = 0.025$	*traingdx, trainoss*
25.	$x = \sin\left(t^2 - 6t + 3\right), \quad t \in [0, 6],\ h = 0.025$	*traincgp, trainlm*
26.	$x = \sin\left(\sin(t)t^2\right), \quad t \in [0, 3.5],\ h = 0.01$	*trainrp, trainbfg*
27.	$x = \sin\left(t^2 - 5t + 6\right), \quad t \in [0, 6],\ h = 0.02$	*traingda, trainoss*
28.	$x = \sin\left(2.5t^2 - 5t\right), \quad t \in [0, 2.2],\ h = 0.01$	*traincgf, trainbfg*

№	T	trainFcn
29.	$x = \sin\bigl(2t^2 - 6t + 3\bigr), \quad t \in [0, 5],\ h = 0.02$	*traincgp, trainlm*
30.	$x = \sin\bigl(-3t^2 + 5t + 10\bigr), \quad t \in [0, 2.5],\ h = 0.01$	*trainscg, trainlm*

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. – –pp. 2-2–2-32.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 47–101.

3. *Hagan M., Demuth H.* Neural Network Design. 1996. – Chapter 11 and 12. –pp. 11-1–12-52.

4. *Бортаковский А. С., Пантелеев А. В.* Аналитическая геометрия в примерах и задачах: Учеб. пособие. – М.: Высш. шк., 2005. – с. 135–137, 268-289.