

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таныгин Максим Олегович
Должность: и.о. декана факультета фундаментальной и прикладной информатики
Дата подписания: 21.09.2023 13:01:30
Уникальный программный ключ:
65ab2aa0d384efe849fed330e11a

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет» (ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ:
проректор по учебной работе
Локтионова О.Г.
2022 г.



**ОТЫСКИВАНИЕ ПЕРИОДИЧЕСКИХ ДВИЖЕНИЙ
НЕАВТНОМНЫХ ДИНАМИЧЕСКИХ СИСТЕМ**

Методические указания для выполнения лабораторной работы по
дисциплине «Теория динамических систем» для студентов
направления подготовки 09.03.04 «Программная инженерия»

Курск 2022

УДК 519.71

Составитель: Р.А. Томакова

Рецензент

Кандидат технических наук, к.т.н., доцент А.В. Малышев

Отыскание периодических движений неавтономных динамических систем: методические указания для проведения лабораторных работ и выполнения самостоятельной внеаудиторной работы по дисциплине «Теория динамических систем» для студентов всех форм обучения направления подготовки 09.03.04 «Программная инженерия» / Юго-Зап. гос. ун-т; сост. Р.А. Томакова. Курск, 2022. –18с.

Рассмотрена методика отыскания периодических движений неавтономных динамических систем, произведено обучение сети на основе алгоритма обратного распространения ошибки, выполнено исследование модели ИНС в системе **MATLAB** с использованием графического инструмента пользователя **NNTOOL** из пакета **NeuralNetworkToolbox**.

Методические указания соответствуют требованиям рабочей программы дисциплины «Теория нейрокомпьютерных систем».

Предназначены для студентов всех форм обучения направления подготовки 09.03.04 «Программная инженерия».

Текст печатается в авторской редакции

Подписано в печать 2022 Формат 60×84 1/16.
Усл. печ. л. 1,1. Уч.- изд. л. 1 . Тираж 25 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

ОТЫСКАНИЕ ПЕРИОДИЧЕСКИХ ДВИЖЕНИЙ НЕАВТОНОМНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

Цель и задачи лабораторного занятия (лабораторной работы): изучить методы нахождения периодических движений неавтономных динамических систем; овладеть навыками обучения сети на основе алгоритма обратного распространения ошибки; реализовать исследование модели ИНС в системе **MATLAB** с использованием графического инструмента пользователя **NNTOOL** из пакета **NeuralNetworkToolbox**.

Планируемые результаты обучения (формируемые знания, умения, навыки и компетенции):

Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной:
ПК-7.4 - Вносит изменения в процедуры сборки модулей и компонент программного обеспечения, развертывания программного обеспечения, миграции и преобразования данных

Необходимые материально-техническое оборудование и материалы:

1. Класс ПЭВМ - Athlon 64 X2-2.4; Cel 2.4, Cel 2.6, Cel 800.
2. Мультимедиа центр: ноутбук ASUS X50VL PMD T2330/14"/1024Mb/ 160Gb/ сум-ка/проектор inFocus IN24+ .
3. Экран мобильный Draper Diplomat 60x60
4. Доступ в сеть Интернет.

Шкала оценивания и критерии оценивания выполненной лабораторной работы:

Форма контроля	Минимальный балл		Максимальный балл	
	балл	примечание	балл	примечание
Лабораторная работа № 1 (Работа с файлами различных типов изображений в среде MATLAB)	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»

План проведения лабораторного занятия (лабораторной работы)

1. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

Для заданной функции построить ее табличные значения (количество значений должно быть достаточным для того, чтобы аппроксимированная функция визуально совпадала с табличными значениями). Использовать нейронную сеть для аппроксимации этих табличных значений и нахождения аппроксимированной функции. Построить графики таблично и аналитически заданной функции. Произвести аппроксимацию с использованием любого численного метода. Рассчитать абсолютную и относительную погрешности.

Таблица 1.

Варианты индивидуальных заданий

Вариант	Вид функции	Промежуток нахождения решения
1	$(1.85-x) \cdot \cos(3.5x-0.5)$	$x \in [-10, 10]$
2	$\cos(\exp(x)) / \sin(\ln(x))$	$x \in [2, 4]$
3	$\sin(x) / x^2$	$x \in [3.1, 20]$
4	$\sin(2x) / x^2$	$x \in [-20, -3.1]$
5	$\cos(2x) / x^2$	$x \in [-20, -2.3]$
6	$(x-1) \cos(3x-15)$	$x \in [-10, 10]$
7	$\ln(x) \cos(3x-15)$	$x \in [1, 10]$
8	$\cos(3x-15) / \text{abs}(x) = 0$	$x \in [-10, -0.3), (0.3, 10]$
9	$\cos(3x-15) \cdot x$	$x \in [-9.6, 9.1]$
10	$\sin(x) / (1 + \exp(-x))$	$x \in [0.5, 10]$
11	$\cos(x) / (1 + \exp(-x))$	$x \in [0.5, 10]$
12	$(\exp(x) - \exp(-x)) \cos(x) /$	$x \in [-5, 5]$
13	$(\exp(-x) - \exp(x)) \cos(x) /$	$x \in [-5, 5]$
14	$\cos(x-0.5) / \text{abs}(x)$	$x \in [-10, 0), (0, 10]$
15	$\cos(2x) / \text{abs}(x-2)$	$x \in [-10, 2), (2, 10]$

2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Аппроксимация представляет процесс приближения одних математических объектов другими, близкими к исходным. Эта замена позволяет свести задачу к изучению более простых или более удобных объектов и производить исследования их числовых характеристик и качественных свойств.

Аппроксимация может сводиться к нахождению функциональной зависимости по имеющемуся набору данных. В нашем случае, функциональная зависимость будет представлена в нейросетевом базисе, т.е. через комбинацию активационных функций нейронных сетей.

Согласно теореме, доказанной в 1989г. Л. Фунахаши, *произвольно сложная функция может быть аппроксимирована двухслойной нейронной сетью, имеющей произвольные активационные функции.*

Поэтому часто для решения задачи аппроксимации используют нейронную сеть (НС), представляющую собой многослойный персептрон.

В данной лабораторной работе для решения задачи аппроксимации предлагается использование *трехслойной нейронной сети, содержащей входной, один скрытый и выходной слою.*

Первый (входной) слой содержит число нейронов, равное количеству входных факторов.

Выходной слой содержит один нейрон, выдающий выходное значение чаще всего с линейной функцией активации.

Второй (скрытый) слой содержит завышенное число нейронов равное, например, половине суммы числа обучающих данных, равное:

$$n^{(2)} = \frac{\sum P \cdot k_p}{2} \quad (1)$$

где P – число входных факторов, k_p – объем выборки по p -ому фактору.

Затем число нейронов скрытого слоя уменьшается до тех пор, пока ошибка аппроксимации не начнет возрастать.

В качестве функции активации обычно используются функции:

1) линейная $f(S) = k \cdot S$;

2) пороговая $f(S) = \begin{cases} 1, & \text{если } S > 0, \\ 0, & \text{если } S \leq 0 \end{cases};$

3) логистическая (сигмоидальная) $f(S) = \frac{1}{1 + e^{-\alpha S}};$

4) гиперболический тангенс $f(S) = \frac{e^{\alpha S} - e^{-\alpha S}}{e^{\alpha S} + e^{-\alpha S}};$

5) радиального базиса $f(S) = e^{-S^2};$

6) знаковая $f(S) = \begin{cases} 1, & \text{если } S > 0, \\ -1, & \text{если } S \leq 0 \end{cases};$

7) полулинейная $f(S) = \begin{cases} k \cdot S, & \text{если } S > 0, \\ 0, & \text{если } S \leq 0 \end{cases};$

8) экспоненциальная $f(S) = e^{\alpha S}.$

Графики этих функций представлены на рисунке 1.

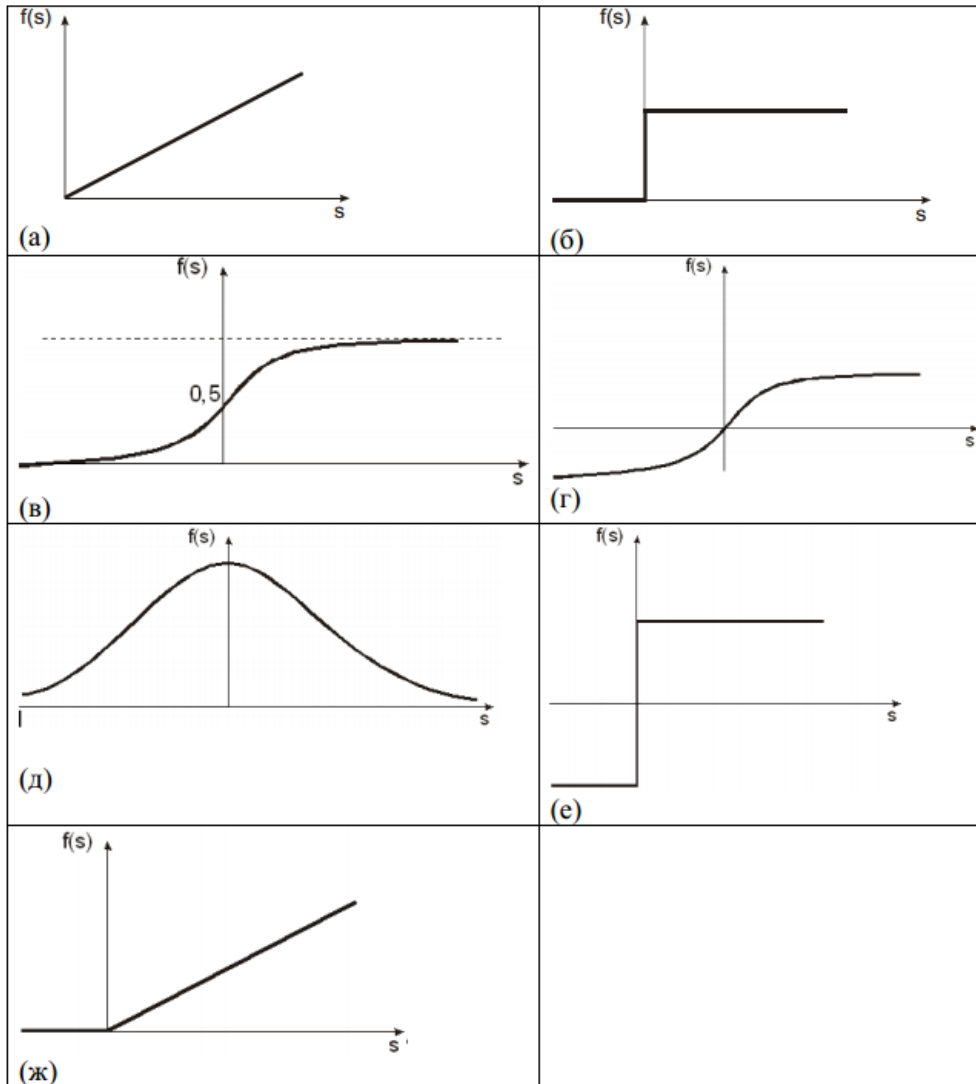


Рисунок 1. Виды функций активации нейронов:

- а) – линейная; б) – пороговая; в) – логистическая (сигмоидальная);
 г) – гиперболический тангенс; д) – функция радиального базиса;
 е) – знаковая; ж) – полулинейная

Чаще всего выбирается гиперболический тангенс. Предпочтительность использования сигмоидального тангенса в качестве функции активации связана с тем, что эта функция является гладкой, дифференцируемой, а производная выражается через саму функцию, что позволяет снизить скорость обучения нейронной сети.

Для простейшего случая аппроксимации одномерной функции $y = f(x)$ нейронная сеть выполняет функцию нахождения преобразования $f(x)$ через функции активации. Число входов в случае одномерной функции равно 1 (x), число выходов – 1 (y).

Нейронная сеть в таком случае может быть представлена рисунке 2.

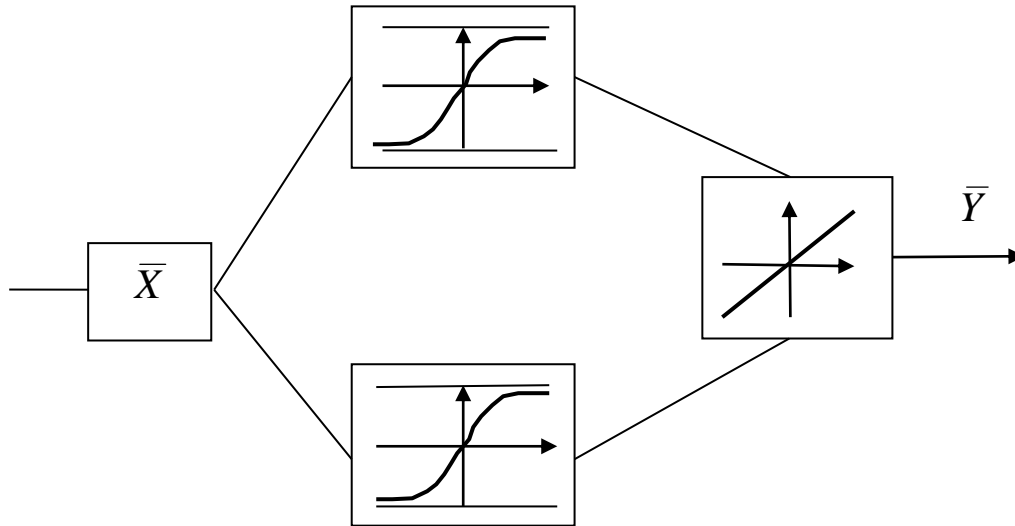


Рисунок 2. Структура нейронной сети

Обучение многослойного персептрона чаще всего выполняется с помощью алгоритма обратного распространения ошибки или его модификации.

1.1 Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки представляет собой системный метод обучения многослойных нейронных сетей (МНС). На этапе обучения происходит вычисление синаптических коэффициентов W .

При этом в основе лежат методы обучения по образцам с помощью примеров, сгруппированных на обучаемом множестве. Для каждого образа из обучающей выборки считается известным требуемое значение выхода нейронной сети (*обучение с учителем*).

Этот процесс можно рассматривать как решение оптимизационной задачи. Ее целью является *минимизация функции ошибки или невязки E на обучающем множестве путем выбора значений синаптических коэффициентов W* .

$$E = \frac{1}{2} \sum_i^P (d_j - y_i)^2 \quad (2)$$

где d_j – требуемое (желаемое) выхода на j -ом образе выборки; y_i – реальное значение выхода; P – число образов обучающей выборки.

Минимизация ошибки E обычно осуществляется с помощью градиентных методов. При этом изменение весов происходит в направлении обратном направлению наибольшей крутизны функции ошибки.

$$W(t+1) = W(t) - \eta \frac{\partial E}{\partial W}, \quad 0 < \eta \leq 1. \quad (3)$$

Существует два подхода к обучению. В первом из них веса w пересчитываются после подачи всего обучающего множества, и ошибка имеет вид:

$$E = \frac{1}{2} \sum_i^P (d_j - y_i)^2. \quad (4)$$

Во втором подходе ошибка пересчитывается после предъявления каждого образца.

$$E_i = \frac{1}{2} (d_i - y_i)^2. \quad (5)$$

$$E(W) = \frac{1}{2} \sum_j (d_{jp} - y_{jp}). \quad (6)$$

Пусть

$$\Delta W_{ij}(t) = -\eta \frac{\partial E}{\partial W_{ij}}. \quad (7)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t). \quad (8)$$

Тогда

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial y_{ij}} \cdot \frac{\partial y_{ij}}{\partial S_i} \cdot \frac{\partial S_j}{\partial W_{ij}}, \quad (9)$$

где $y_j(S_j)$ - функция активации нейронов, например, для функции

$$y = f(S) = \frac{1}{1 + e^{-Sj}}. \quad (10)$$

$$f'(S) = \frac{\partial}{\partial S_j} \left(\frac{1}{1 + e^{-S_j}} \right) = \frac{1}{(1 + e^{-S_j})^2} \cdot (-e^{-S_j}) =$$

$$= \frac{1}{1 + e^{-S_j}} \cdot \frac{-e^{-S_j}}{1 + e^{-S_j}} = y_j \cdot (1 - y_j). \quad (11)$$

$$\frac{\partial y_j}{\partial S_j} = (1 - y_j)^2. \quad (12)$$

Рассмотрим третий сомножитель

$$S_j = \sum W_{ij} \cdot y_i^{(n-1)}. \quad (13)$$

$$\frac{\partial S_j}{\partial W_{ij}} = y_i^{(n-1)}. \quad (14)$$

Можно преобразовать, что

$$\frac{\partial E}{\partial y_j} = \sum \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial S_k} \cdot \frac{\partial S_k}{\partial y_j} = \sum \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial S_k} \cdot W_{jk}^{(n+1)}. \quad (15)$$

Обозначим

$$\delta_j = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_j}{\partial S_j}. \quad (16)$$

Для внутреннего нейрона получим:

$$\delta_j^{(n-1)} = \left[\sum_k \delta_k^n W_{jk}^n \right] \cdot \frac{\partial y_j}{\partial S_j}. \quad (17)$$

Для внешнего нейрона будем иметь:

$$\delta_j^n = (d_j^n - y_j^n) \frac{\partial y_j}{\partial S_j}. \quad (18)$$

$$\Delta W_{ij}^n = -\eta \cdot \delta_j^n \cdot y_i^{(n-1)}. \quad (19)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}. \quad (20)$$

Обучение нейронной сети с помощью алгоритма обратного распространения ошибки имеет вид:

1. Подать на входы сети один из возможных образцов и в режиме обычного функционирования НС, когда сигналы распространяются от входа к выходам рассчитать значения выходов

всех нейронов (обычно начальное значение веса составляют малые значения).

2. Рассчитать значения δ_j^n для нейронов выходного слоя по формуле для внешнего нейрона (18).

3. Рассчитать значения δ_j^n для всех внутренних нейронов по формуле (17).

4. С помощью формулы (19) для всех связей найти приращения весовых коэффициентов W_{ij} .

5. Скорректировать синаптические веса по формуле 20.

6. Повторить шаги 1-5 для каждого образа обучающей выборки пока ошибка E не станет достаточно маленькой.

1.2 Пример выполнения лабораторной работы

1. В окне “**CommandWindow**” введите команду **nntool**. Окно графического инструмента пользователя **nntool**.
2. Нажмите кнопку “**New**”.
3. Задайте имя сети, например **nnapprox**.
4. Введите количество слоев сети равное двум (выходной слой и один скрытый слой) в текстовом поле ввода “**Number of layers**”и нажмите клавишу “**Enter**”.
5. В выпадающем списке “**Propertiesfor**” выберите позицию “**Layer 1**” (рис. 3).

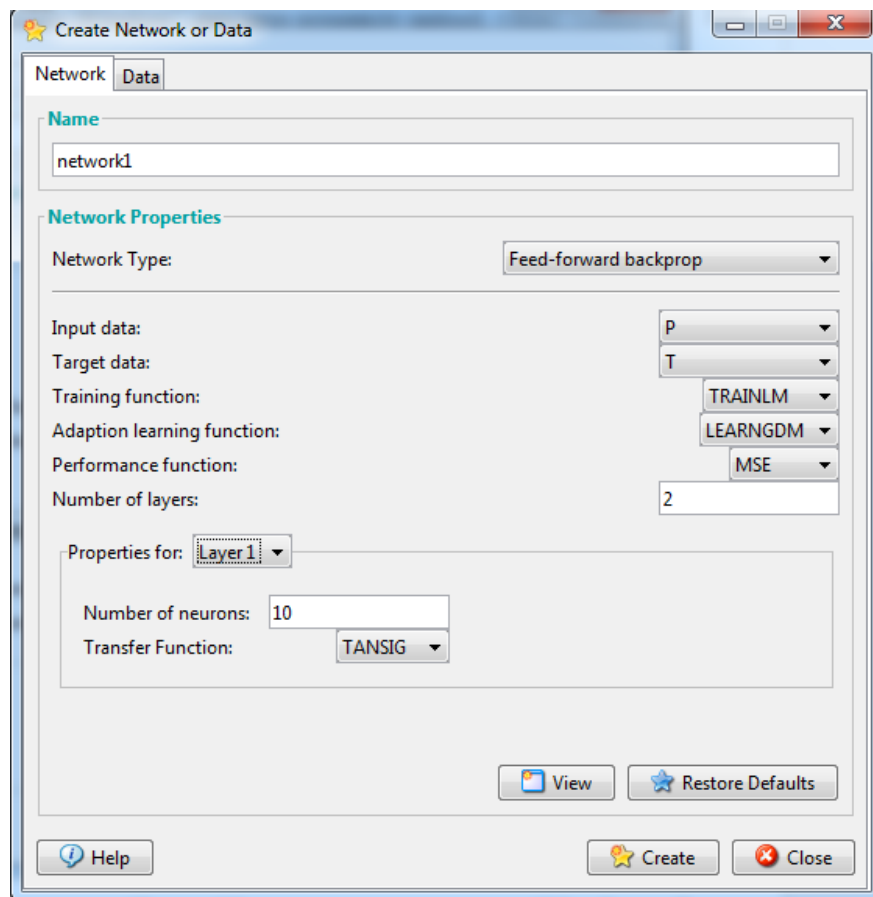


Рисунок 3. Выбор позиции “**Layer 1**” в выпадающем списке “**Numbers of neurons**” в окне создания нейронной сети графического инструмента пользователя **NNTOOL**

В текстовом поле ввода “**Numbers of neurons**” введите количество нейронов в скрытом слое, например 10.

6. В выпадающем списке “**Propertiesfor**” выберите позицию “**Layer 2**”. В выпадающем списке “**Transferfunction**” выберите позицию “**PURELIN**” (рис. 4).

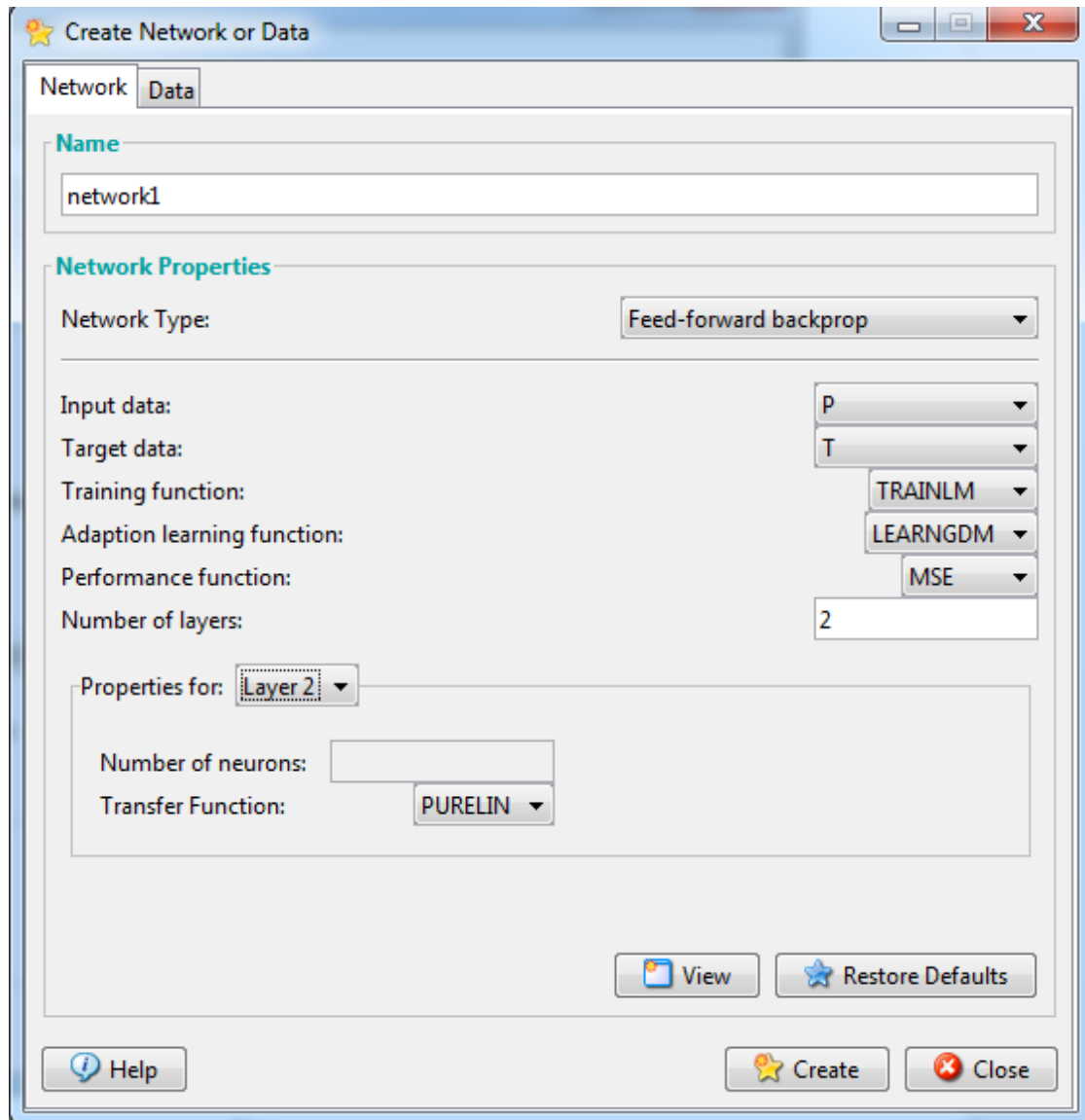


Рисунок 4. Выбор позиции “**PURELIN**” в выпадающем списке “**Transfer Function**” окне создания новой нейронной сети графического инструмента пользователя **NNTOOL**

7. Откройте вкладку “**Data**” и задайте входные данные **P** для обучения нейронной сети, например [-2 -1 0 1 2]. Задайте целевые данные **T** для обучения нейронной сети, например [2 1 0 -1 -2].

8. Откройте вкладку “**Network**” и в выпадающих списках “**Input data**” и “**Target data**” выберите позиции **P** и **T** соответственно (рис. 4).

9. Выберите позиции P и T в выпадающих списках **“Input data”** и **“Output data”** (рис. 4).

10. Нажмите кнопку **“View”**. Откроется окно, содержащее изображение структуры нейронной сети (рис.5). Рассмотрите структуру нейронной сети и закройте это окно.

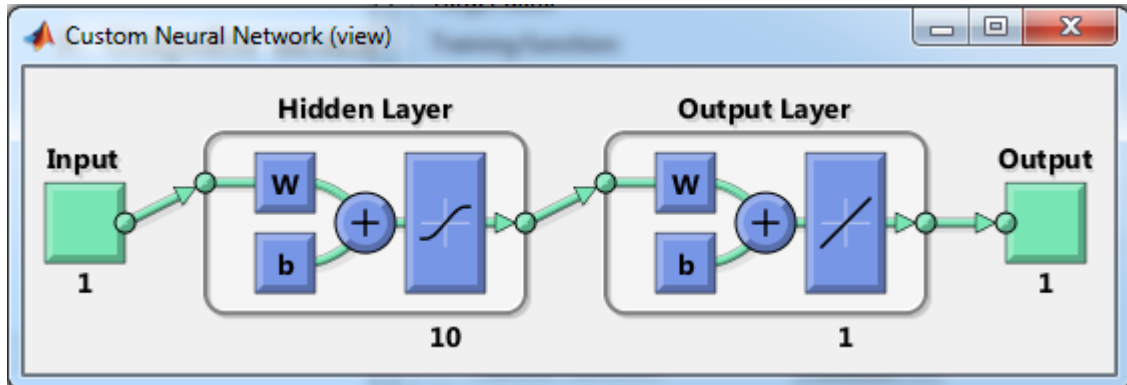


Рисунок 5. Отображение структуры нейронной сети

11. Нажмите кнопку **“Create”**.

12. Нажмите кнопку **“Close”**.

13. Дважды кликните мышкой по позиции **“nnapprox”**. Откроется окно инициализации и обучения сети (рис. 6).

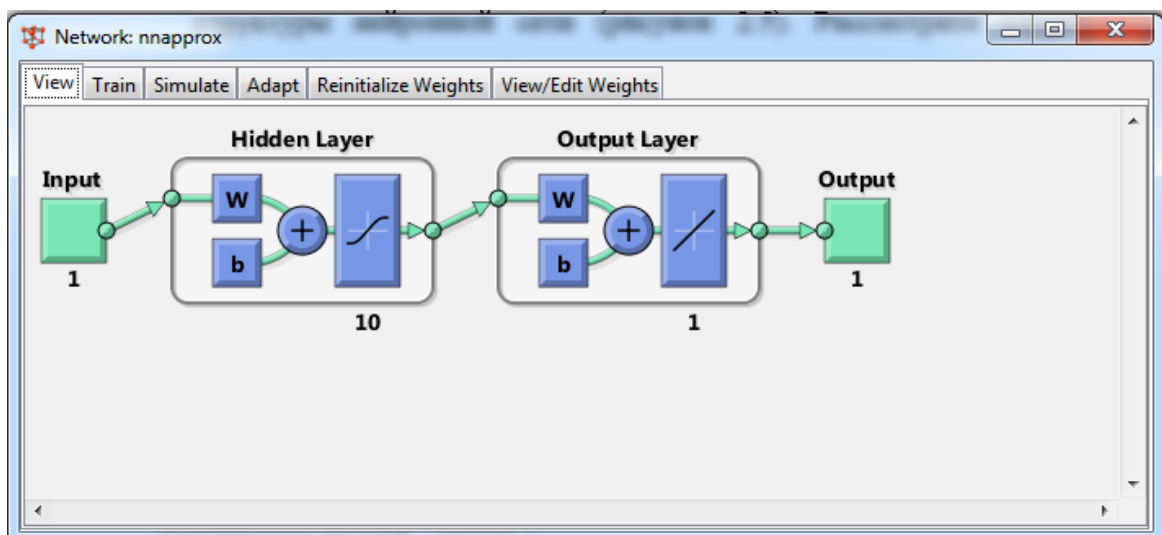


Рисунок 6. Окно инициализации и обучения сети

14. Откройте на вкладку **“ReinitializeWeights”** (рис.7) и нажмите кнопку **“InitializeWeights”**.

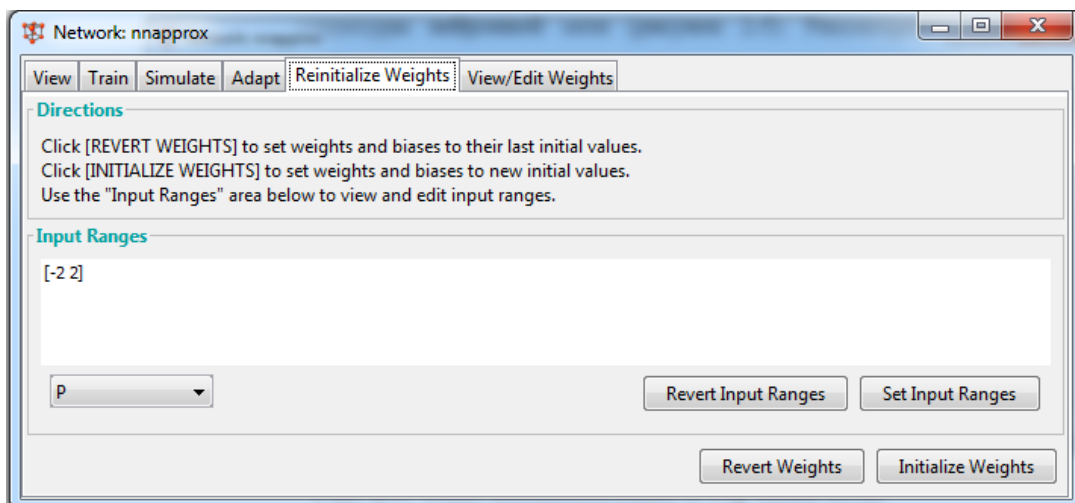


Рисунок 7. Вкладка “Reinitialize Weights”

15. Перейдите на вкладку “Train” и выпадающих списках “Inputs” и “Targets” выберите позиции **P** и **T** соответственно (рис.8).

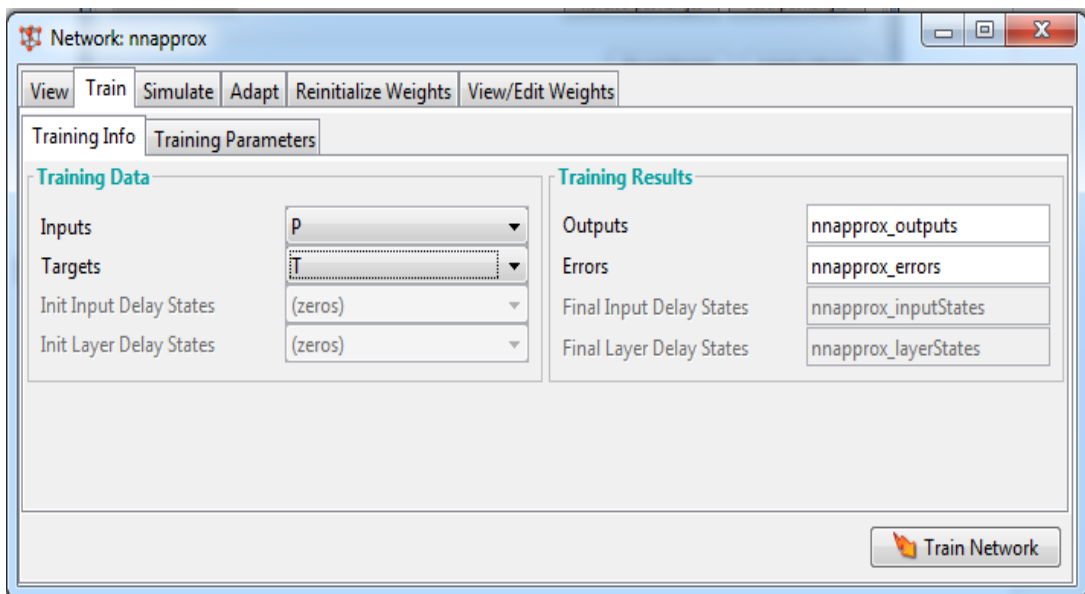


Рисунок 8. Вкладка “Train”

16. Нажмите кнопку “Train Network”. Откроется окно обучения сети. Дождитесь окончания обучения. После окончания обучения закройте окно обучения сети.

17. Перейдите на вкладку “Simulate” (рис. 9). В выпадающем списке “Inputs” выберите позицию **P**.

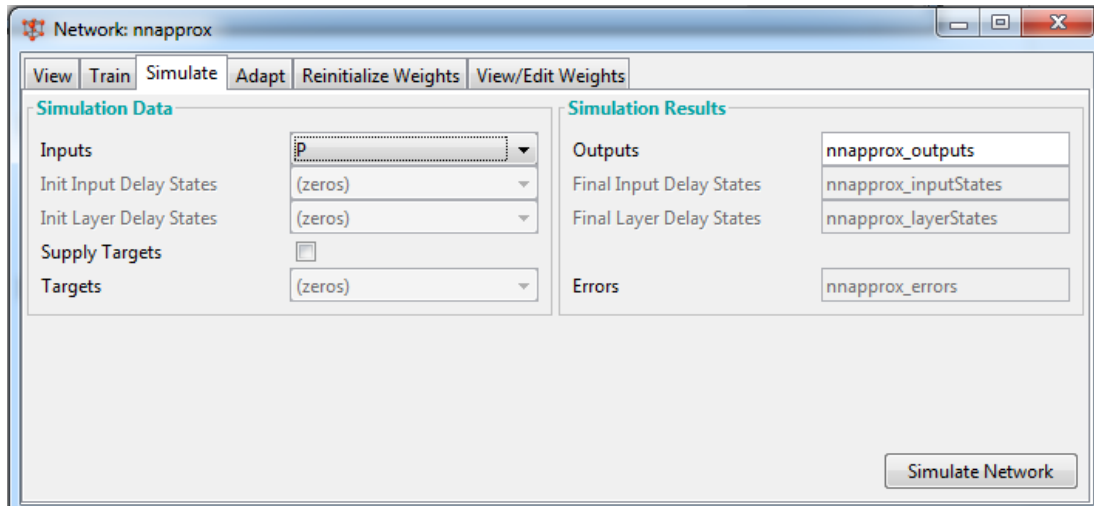


Рисунок 9. Вкладка “Simulate”

18. Нажмите кнопку “**Simulate Network**”.
19. Сравните результат симуляции **nnapprox_outputs** и целевыми данными **T**.
20. Произведите аппроксимацию любым численным методом (например, методом наименьших квадратов).
21. Рассчитайте абсолютную и относительную погрешности.

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Приведите основные типы функций активации нейронов.
2. Какой вид имеют графики активационных функций?
3. Сформулируйте постановку задачи аппроксимации
4. В чем заключается алгоритм обратного распространения ошибки?
6. Как выполняется обучение нейронной сети с учителем?
5. Какие способы нахождения погрешности результата вы знаете?
6. Предоставить отчет. Содержание отчета:
 - титульный лист;
 - тема и цель работы;
 - формальная постановка задачи классификации;
 - основные теоретические сведения;
 - задание, учитывая свой вариант;
 - ход создания, обучения и тестирования персептронной однослойной ИНС с использованием графического инструмента пользователя **nntool**;
 - выводы.
7. Расчет относительной и абсолютной погрешностей.
8. Выводы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Коробова, Л. А. Теория динамических систем (теория и практика) : учебное пособие / Л. А. Коробова, Ю. А. Сафонова ; науч. ред. Л. А. Коробова ; Воронежский государственный университет инженерных технологий. – Воронеж : Воронежский государственный университет инженерных технологий, 2017. – 100 с. – URL: <https://biblioclub.ru/index.php?page=book&id=482071> (дата обращения: 22.03.2022). – Режим доступа: по подписке. – Текст : электронный.
2. Зубарев, А. А. Имитационное моделирование динамических систем в среде AnyLogic : учебное пособие : [16+] / А. А. Зубарев ; Омский государственный технический университет. – Омск : Омский государственный технический университет (ОмГТУ), 2020. – 82 с. –

URL: <https://biblioclub.ru/index.php?page=book&id=682373> (дата обращения: 22.03.2022). – Режим доступа: по подписке.– Текст : электронный.

3. Вагин, Д. В. Численное моделирование динамических систем, описываемых обыкновенными дифференциальными уравнениями : учебное пособие / Д. В. Вагин ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2019. – 63 с. URL: <https://biblioclub.ru/index.php?page=book&id=573956> (дата обращения: 22.03.2022). – Режим доступа: по подписке. – Текст : электронный.

4. Чубич, В. М. Активная идентификация стохастических динамических систем: планирование эксперимента для моделей непрерывно-дискретных систем : учебное пособие / В. М. Чубич, Е. В. Филиппова ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2017. – 96 с. – URL: <https://biblioclub.ru/index.php?page=book&id=574667> (дата обращения: 22.03.2022). – Режим доступа: по подписке.– Текст : электронный.

5. Шорников, Ю. В. Компьютерное моделирование динамических систем : учебное пособие / Ю. В. Шорников, Д. Н. Достовалов ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2017. – 68 с. – URL: <https://biblioclub.ru/index.php?page=book&id=575038> (дата обращения: 22.03.2022). – Режим доступа: по подписке. – Текст : электронный.

6. Ушаков, А.В., Хабалов В.В., Дударенко Н.А. Математические основы теории систем: Элементы теории и практикум [Текст]: учебное пособие / А. В. Ушаков, В. В. Хабалов, Н. А. Дударенко. - СПб.: СПбГУ ИТМО, 2007. - 174 с. – Режим доступа : <http://window.edu.ru/resource/589/41589>.

7. Громов, Ю.Ю. Специальные разделы теории управления. Оптимальное управление динамическими системами [Текст]: учебное пособие/Ю.Ю. Громов, Н.А. Земский, А.В. Лагутин [и др.] - Тамбов: Издательство ТГТУ, 2007. - 108 с. – Режим доступа : <http://window.edu.ru/resource/880/56880>.