

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 13:00:36
Уникальный программный ключ:
65ab2aa0d384efe84406d330e11a

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

Кафедра программной инженерии



Утверждаю:

Директор по учебной работе

Локтионова О.Г.

2021г.

**АПРОКСИМАЦИЯ ФУНКЦИЙ С ПОМОЩЬЮ
НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ**

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

Курск 2021

УДК 004.932

Составитель: Р.А. Томакова

Рецензент

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

Аппроксимация функций с помощью нейронной сети прямого распространения: методические указания для проведения лабораторных работ и выполнения самостоятельной внеаудиторной работы по дисциплине «Теория нейрокомпьютерных систем» для студентов всех форм обучения направления подготовки 09.03.04 «Программная инженерия» / Юго-Зап. гос. ун-т; сост. Р.А. Томакова. Курск, 2021. –17с.

Рассмотрена методика аппроксимации функций с помощью нейронной сети прямого распространения, 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

Формат 60×84 1/16.

Усл. печ. л. 1,1. Уч.- изд. л. 1 . Тираж 25 экз. Заказ. 884 Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

АППРОКСИМАЦИЯ ФУНКЦИЙ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ

Цель работы: изучение возможности аппроксимации с помощью нейронной сети прямого распространения.

1. 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.

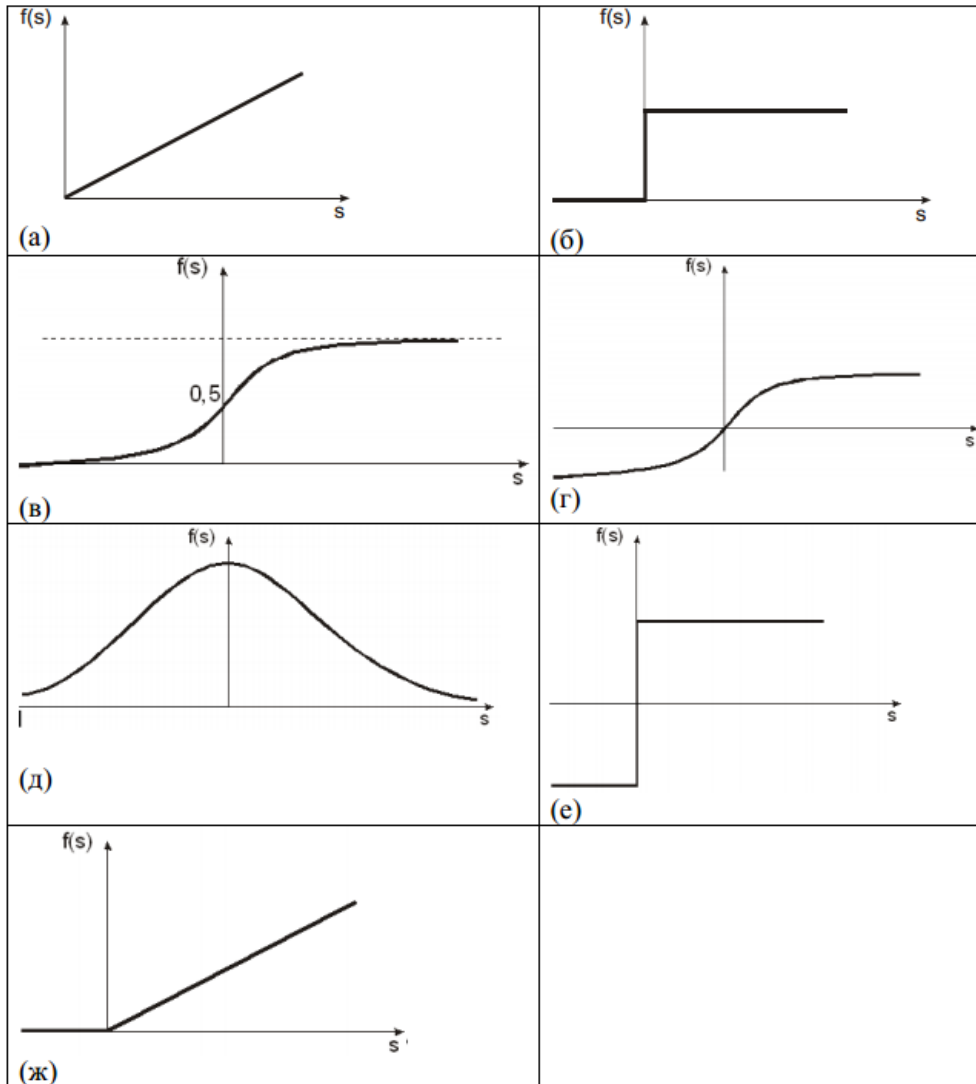


Рисунок 1. Виды функций активации нейронов:

- 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.

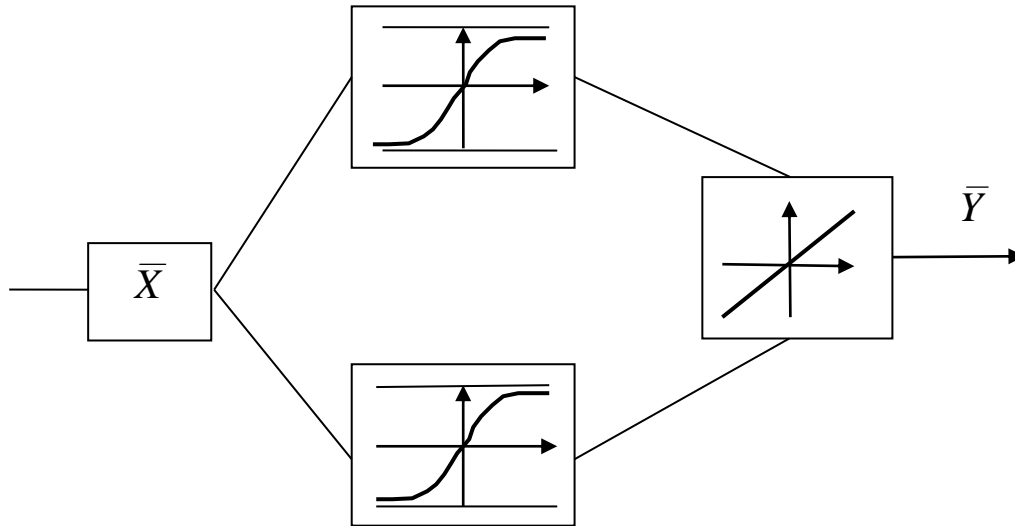


Рисунок 2. Структура нейронной сети

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

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

$$\begin{aligned}
 f'(S) &= \frac{\partial}{\partial S_j} \left(\frac{1}{1 + e^{-S_j}} \right) = \frac{1}{(1 + e^{-S_j})^2} \cdot (-e^{-S_j}) = \\
 &= \frac{1}{1 + e^{-S_j}} \cdot \frac{-e^{-S_j}}{1 + e^{-S_j}} = y_j \cdot (1 - y_j).
 \end{aligned} \tag{11}$$

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

Рассмотрим третий сомножитель

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

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

Можно преобразовать, что

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

Обозначим

$$\delta_j = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_j}{\partial S_j}. \tag{16}$$

Для внутреннего нейрона получим:

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

Для внешнего нейрона будем иметь:

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

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

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

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 **nnaprox**.
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).

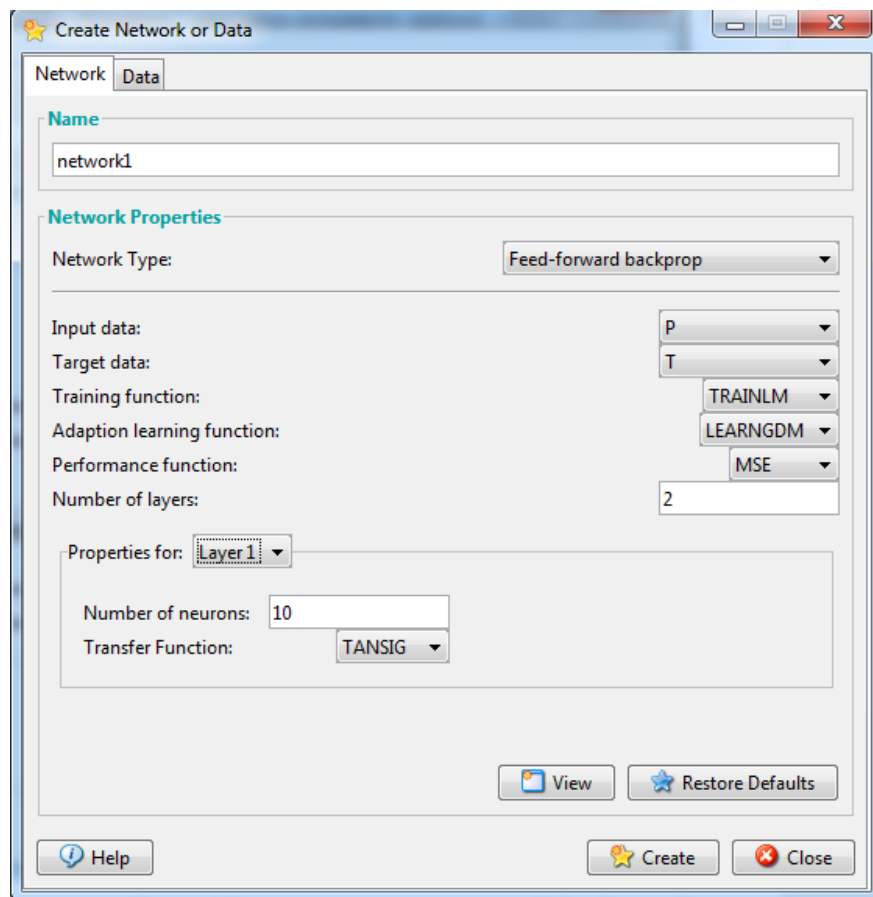


Рисунок 3. Выбор позиции “**Layer 1**” в выпадающем списке “**Numbers of neurons**” в окне создания нейронной сети графического инструмента пользователя **NNTOOL**

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).

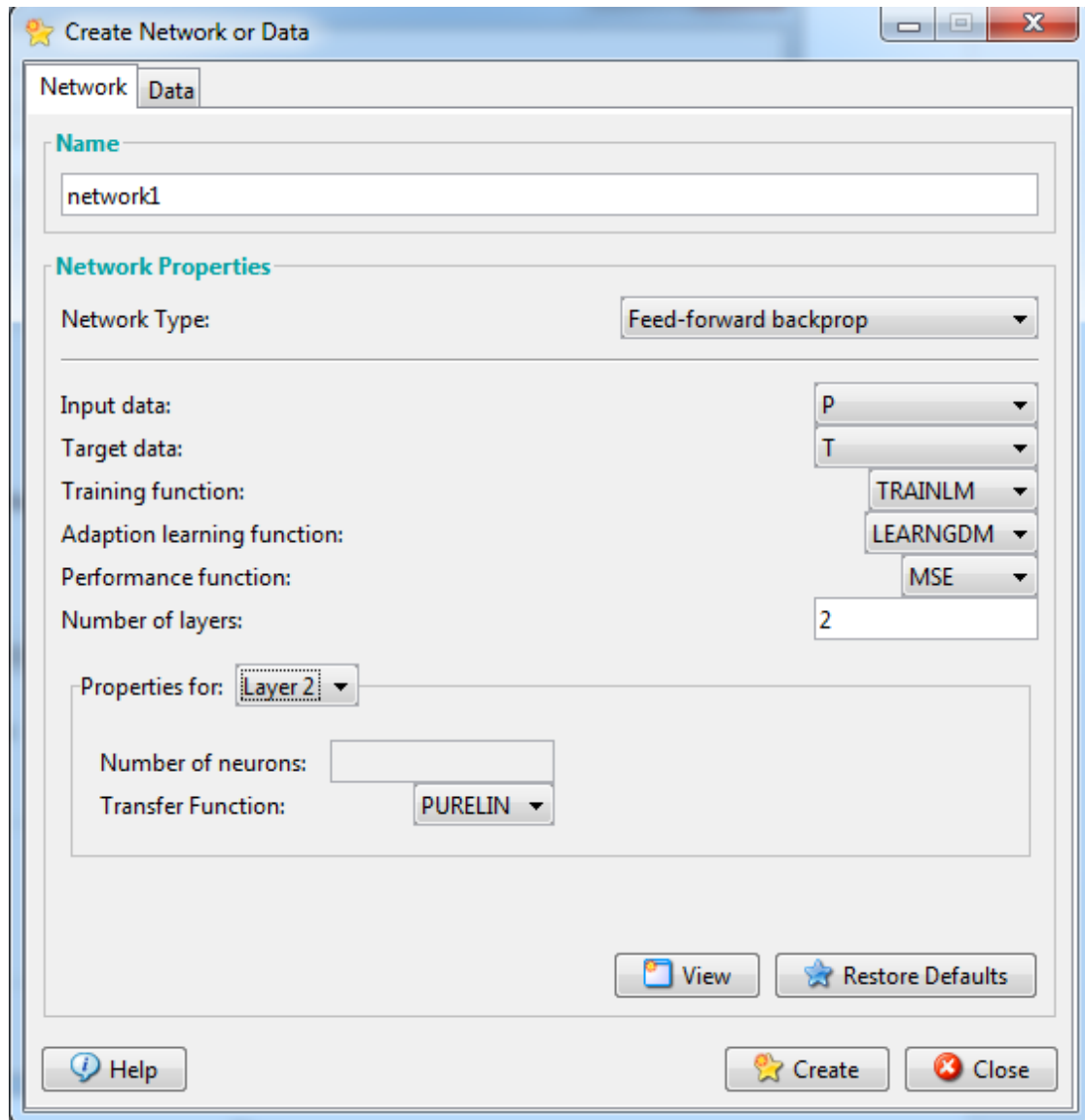


Рисунок 4. Выбор позиции “**PURELIN**” в выпадающем списке “**Transfer Function**” окне создания новой нейронной сети графического инструмента пользователя **NNTOOL**

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.

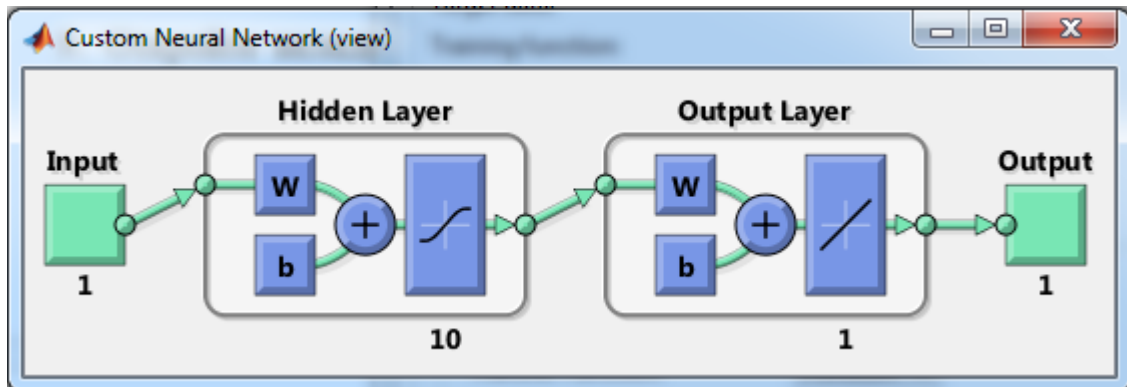


Рисунок 5. Отображение структуры нейронной сети

11. Нажмите кнопку **“Create”**.

12. Нажмите кнопку **“Close”**.

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

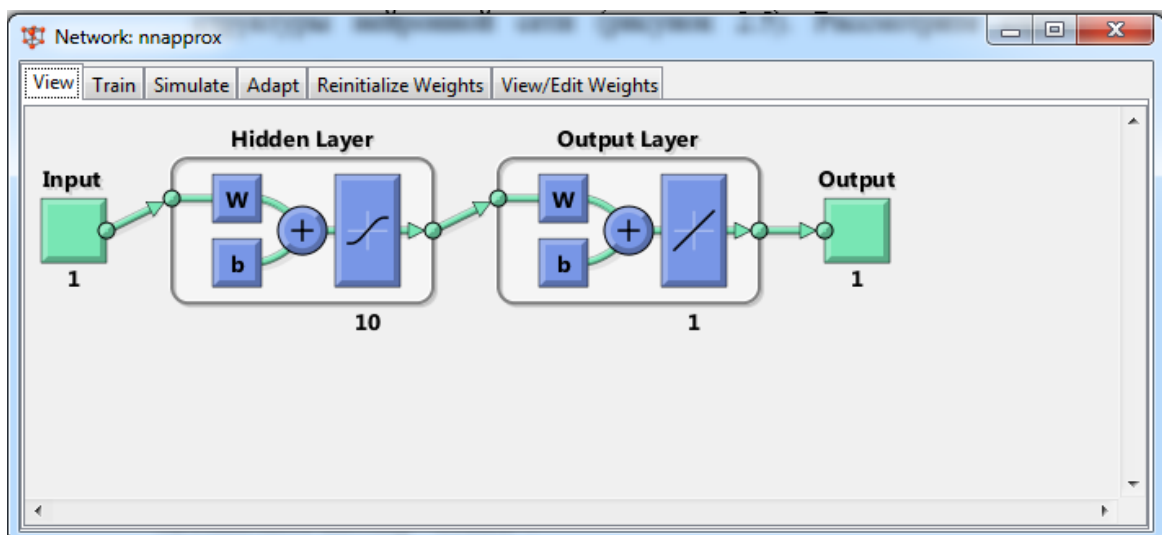


Рисунок 6. Окно инициализации и обучения сети

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

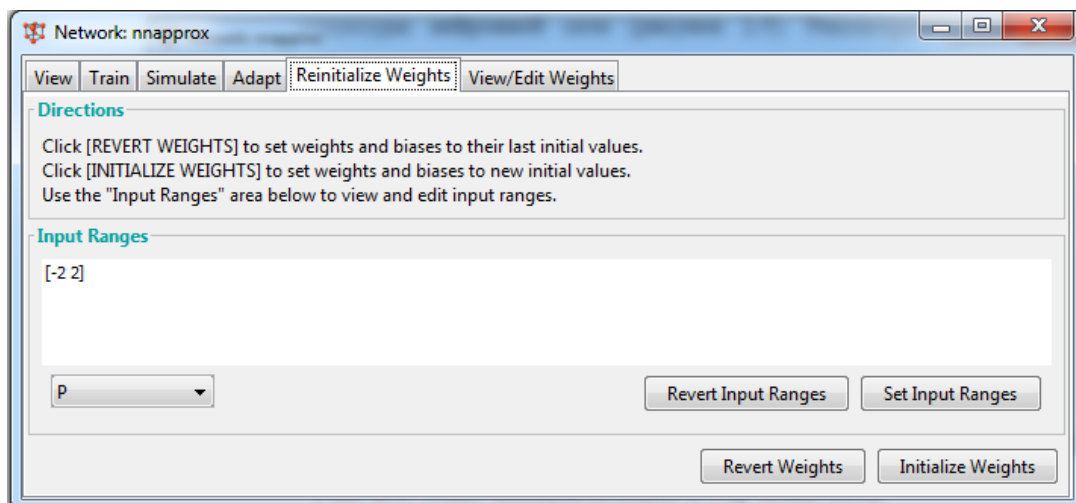


Рисунок 7. Вкладка “Reinitialize Weights”

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

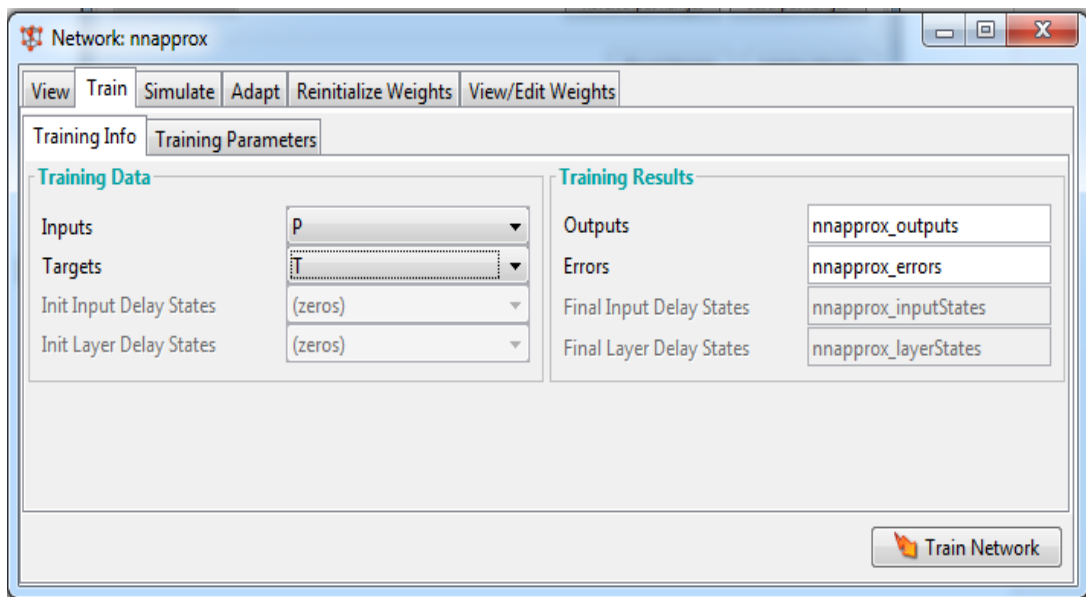


Рисунок 8. Вкладка “Train”

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**.

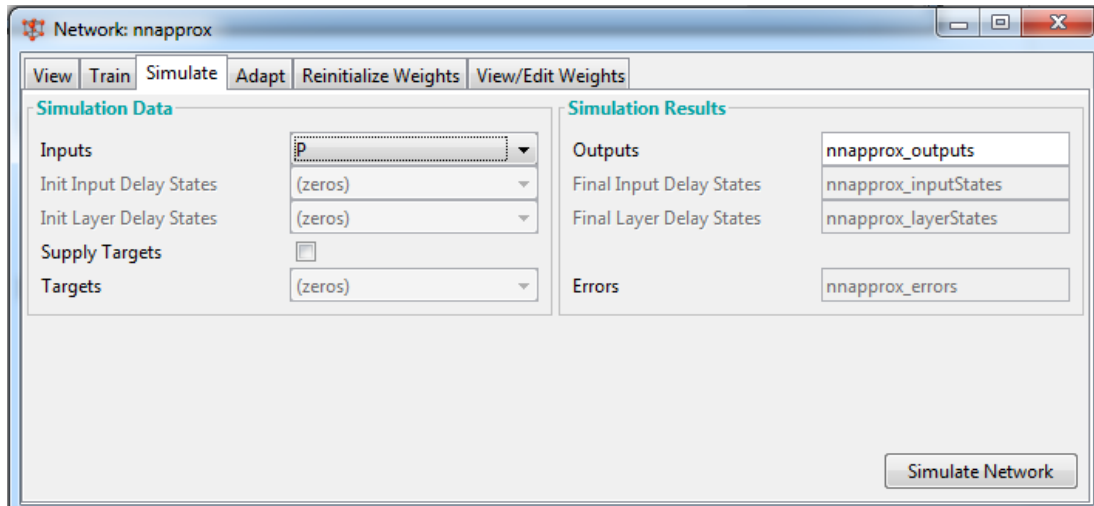


Рисунок 9. Вкладка “Simulate”

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.

2. 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	$\cos(2x) / \text{abs}(x-2)$	$x \in [-10, 2), (2, 10]$

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 перцептронной однослойной ИНС с использованием графического инструмента пользователя **nntool**;
 - выводы.
7. Расчет относительной и абсолютной погрешностей.
8. Выводы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Борисовский, С.А. Нейросетевые модели с иерархическим пространством информативных признаков для сегментации плохоструктурированных изображений/ С.А. Борисовский, А.Н. Брежнева, Р.А. Томакова// Биомедицинская радиоэлектроника, 2010, - № 2. – С. 49-53.
2. Дж Ту, Гонсалес, Р. Принципы распознавания образов / Р. Гонсалес, Дж Ту – М.: Мир, 1978. – 411 с.
3. Корневский, Н.А. Нейронные сети с макрослоями для классификации и прогнозирования патологий сетчатки глаза/ Н.А. Корневский, Р.А. Томакова, С.П. Серегин, А.Ф. Рыбочкин // Медицинская техника, 2013. - № 4. – С. 16-18.
4. Томакова Р.А. Теоретические основы и методы обработки и анализа микроскопических изображений биоматериалов: монография

/ Р.А. Томакова, С.Г. Емельянов, С.А. Филист. – Курск, Юго-Зап. гос. ун-т, 2011. – 202с.

5. Томакова Р.А. Интеллектуальные технологии сегментации и классификации биомедицинских изображений: монография / Р.А. Томакова, С.Г. Емельянов, С.А. Филист. – Курск, Юго-Зап. гос. ун-т, 2012. – 222с.

6. Томакова, Р.А. Структурно-функциональные решения нечетких нейронных сетей для интеллектуальных систем анализа разнотипных признаков/ Р.А. Томакова, С.А. Филист, В.В. Жилин, С.А. Горбатенко//Фундаментальные и прикладные проблемы техники и технологии, 2011. - № 1. –С. 85-91.

7. Томакова, Р.А. Универсальные сетевые модели для задач классификации биомедицинских данных/ Р.А. Томакова, С.А. Филист, Яа Зар До// Известия Юго-Западного государственного университета, 2012. - № 4-2(43). – С.44-50.

8. Томакова, Р.А. Метод обработки и анализа сложноструктурируемых изображений на основе встроенных функций среды MATLAB / Р.А. Томакова, С.А. Филист //Вестник Читинского государственного университета. – 2012. – № 1 (80). – С. 3-9.

9. Томакова, Р.А. Нейросетевые модели принятия решений для диагностики заболеваний легких на основе анализа флюорограмм грудной клетки/ Р.А. Томакова, М.В. Дюдин, М.В. Томаков // Биомедицинская радиоэлектроника. –2014. – №9. – С. 12-15.

10. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: ООО «И.Д. Вильямс», 2006. – 1104с.