Лабораторная работа №2

Исследование активационных функций нейронных элементов

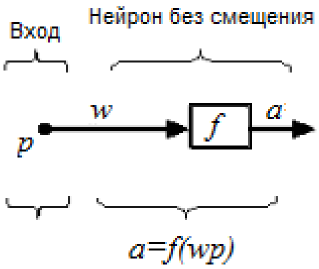
1 Цель работы

Углубление теоретических знаний в области архитектуры нейронных сетей, исследование свойств активационных функций нейронных элементов, приобретение практических навыков моделирования простейшей нейронной сети прямого распространения.

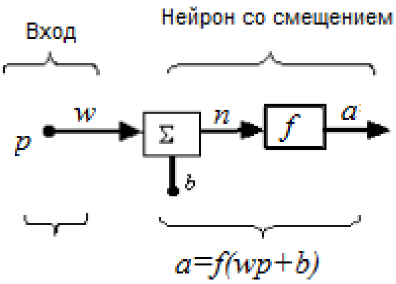
2 Основные теоретические положения

2.1 Структура нейрона с одним входом

Элементарной ячейкой нейронной сети является простейший нейрон. Структурная схема простейшего нейрона с единственным скалярным входом показана на рисунке 2.1*а*. На рисунке приняты обозначения, используемые в книге [3].



а)



б)

Рисунок 2.1 – Структура нейрона с единственным входом

Скалярный входной сигнал *p* умножается на скалярный весовой коэффициент *w*, и результирующий взвешенный вход w*p является аргументом функции активации нейрона *f*, которая порождает скалярный выход *a*.

Нейрон, показанный на рисунке 2.1,*б*, дополнен скалярным смещением *b*. Смещение суммируется со взвешенным входом w*p и приводит к сдвигу аргумента функции *f* на величину *b*. Действие смещения можно свести к схеме взвешивания, если представить, что нейрон имеет второй входной сигнал со значением равным 1, а весовой коэффициент равен *b*. Вход *n* функции активации нейрона по-прежнему остается скалярным и равным сумме взвешенного входа и смещения *b*. Эта сумма является аргументом функции активации *f*; выходом функции активации является сигнал *a*. Переменные *w* и *b* являются скалярными параметрами нейрона. Основной принцип работы нейронной сети состоит в настройке параметров нейрона таким образом, чтобы поведение сети соответствовало некоторому желаемому критерию. Регулируя веса и смещение, можно обучить сеть выполнять определенную функцию; возможно также, что сеть сама будет корректировать свои параметры, чтобы достичь требуемого результата.

Уравнение простейшего нейронного элемента со смещением можно записывать в виде

$$a = f(w * p + b * 1)\,. \tag{2.1}$$

Здесь константа 1 рассматривается как входное значение и может быть учтена в линейной комбинации *расширенного вектора весов* **x**=[*w b*] и *расширенного вектора входа* **z**=[*p 1*]$^{\mathbf{T}}$ **:**

$$a = f([w \quad b]\begin{bmatrix} p \\ 1 \end{bmatrix}) = f(\mathbf{x}\,\mathbf{z}). \tag{2.2}$$

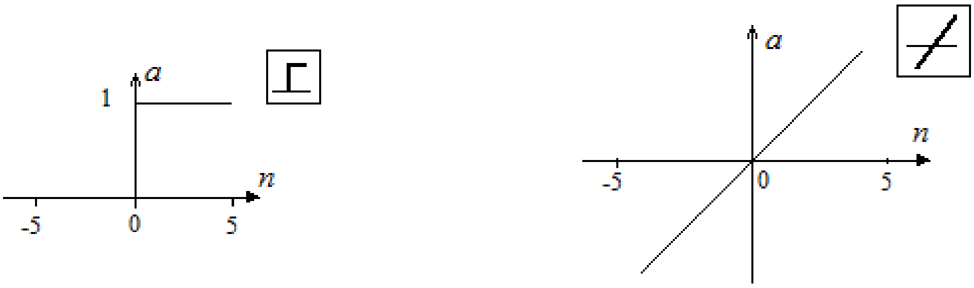
2.2 Активационные функции

Рассмотрим основные функции активации, используемые в нейронных сетях и реализованные в модуле NeuralNetworks 2.0 пакета Scilab.

Единичная функция активации с жестким ограничением (hardlim) соответствует функции Хевисайда. Она равна 0, если n<0, и 1, если $n \geq 0$. Пример вызова функции в пакете Scilab:

```
n = [-5:0.1:5];
a = ann_hardlim_activ(n); // вызов функции hardlim
plot(n,a,'.');
```

В результате получим график функции hardlim в диапазоне значений входа от –5 до + 5 (рисунок 2.2*а*).



а) *б*)

Рисунок 2.2 – Функции активации нейронных элементов

Линейная функция активации (purelin**)**. Линейная функция описывается выражением a=n. График функции изображен на рисунке 2.2*б*. Пример вызова функции в пакете Scilab: a = ann_purelin_activ(n).

Логистическая (сигмовидная) функция активации (logsig). Эта функция описывается соотношением $a = 1/(1 + \exp(-n))$. График функции изображен на рисунке 2.3. Аргумент функции может принимать любое значение в диапазоне от –∞ до +∞, а выход изменяется в диапазоне от 0 до 1. Пример вызова в пакете

Scilab: a = ann_logsig_activ(n). Благодаря свойству дифференцируемости эта функция часто используется в сетях с обучением на основе градиентных методов оптимизации.

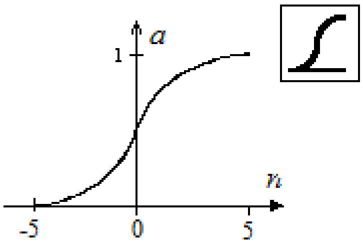
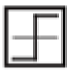
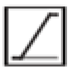
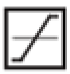
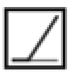
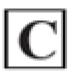


Рисунок 2.3 – График сигмовидной функции активации

Тангенциальная (сигмовидная) функция активации (tansig). Эта функция описывается соотношением $a = (exp(n)\text{-}(exp(\text{-}n))/(exp(n) + exp(-n))$. График функции подобен графику функции logsig. Её аргумент также может принимать любое значение в диапазоне от $-\infty$ до $+\infty$, но значения функции изменяются в диапазоне от -1 до 1. Пример вызова функции в пакете Scilab: a = ann_tansig_activ(n).

В таблице 2.1 представлены дополнительные функции активации, часто используемые в моделях ИНС. При необходимости они легко могут быть запрограммированы самостоятельно.

Таблица 2.1 – Дополнительные функции активации

Симметричная hardlim	$a = -1 \quad n < 0$ $a = +1 \quad n \ge 0$		hardlims
Линейная с насыщением	$a = 0 \quad n < 0$ $a = n \quad 0 \le n \le 1$ $a = 1 \quad n > 1$		satlin
Симметричная линейная с насыщением	$a = -1 \quad n < -1$ $a = n \quad -1 \le n \le 1$ $a = 1 \quad n > 1$		satlins
Положительная линейная (ReLU)	$a = 0 \quad n < 0$ $a = n \quad 0 \le n$		poslin
Состязательная	$a = 1$ для нейрона с max n $a = 0$ для всех других		compet

Символы, изображенные в квадрате в правом верхнем углу графиков на рисунках 2.2–2.3, а также в 3-ем столбце таблицы 2.1, представляют условное обозначение функций активации. Эти обозначения используются на структурных схемах нейронных сетей.

2.3 Нейрон с векторным входом и сеть прямого распространения

Нейрон с одним вектором входа **p**, состоящим из R элементов p_1, p_2, …, p_R, изображен на рисунке 2.4. Здесь каждый элемент вектора p умножается на веса w_{11}, w_{12}, … , w_{1R} . Взвешенные значения вектора входа поступают на сумматор. Их сумма равна скалярному произведению вектора-строки **W** на вектор входа **p**.

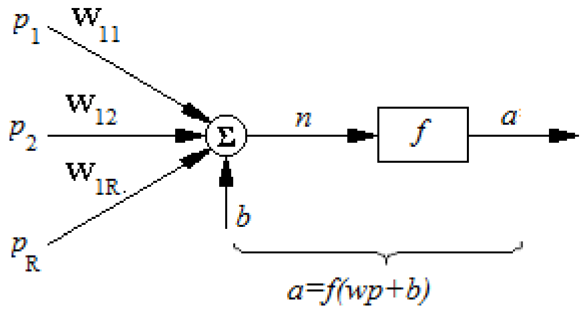


Рисунок 2.4 – Схема простейшего нейрона с векторонным входом

Смещение b суммируется со взвешенной суммой входов. Результирующая сумма n, называемая *сетевым значением*, равна

$$n = w_{11}\, p_1 + w_{12}\, p_2 + \ldots + w_{1R}\, p_R + b \qquad (2.3)$$

Сетевое значение n служит аргументом функции активации f. Выражение (2.3) удобно записывать в векторно-матричной форме:

$$n = \mathbf{Wp} + b. \qquad (2.4)$$

При рассмотрении нейросетей, содержащих нейроны с векторными входами, часто используется обобщенная векторно-матричная схема нейрона, изображенная на рисунке 2.5 и которая структурно соответствует выражению (2.4).

Вход нейрона изображается в виде темного прямоугольника, под которым указывается количество элементов R входного вектора **p**. Размер вектора входа **p** указывается ниже символа **p** и равен R×1. Вектор входа умножается на вектор-строку **W** длины R. Как и прежде, константа 1 рассматривается как вход, который умножается на скалярное смещение b.

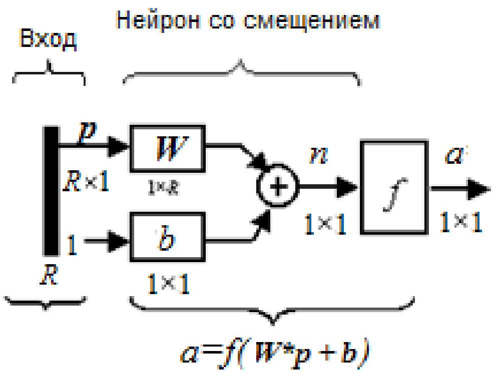


Рисунок 2.5 – Обобщенная векторно-матричная схема нейронного элемента

Входом *n* функции активации нейрона служит сумма смещения *b* и произведения **W*p**. Эта сумма преобразуется функцией активации *f* в выходное значение нейрона *a*, которое в данном случае является скалярной величиной. Структурная схема, изображенная на рисунке 2.5, соответствует одному *слою сети.* Если слой содержит *S* нейронов, то он характеризуется *матрицей весов* **W** размером *SxR*, и вектором смещений **b** размером *S*x1.

Каждый раз, когда используется обобщенное обозначение сети, размерность матриц указывается под именами векторно-матричных переменных. Эта система обозначений поясняет структуру сети и связанную с ней матричную математику.

С помощью схемы однослойной сети, изображенной на рисунке 2.5, можно построить многослойную сеть. В качестве примера на рисунке 2.6 изображена трехслойная сеть прямого распространения (FF – feed forward). Для этой сети выход предыдущего слоя является входом следующего слоя. Входом сети является вход первого слоя, т.е. вектор **p**, а выходом – выход последнего слоя, т.е. вектор $\mathbf{a}^3$. Соответственно, прямое распространение входного вектора по сети опишется уравнением:

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3) \quad . \qquad (2.5)$$

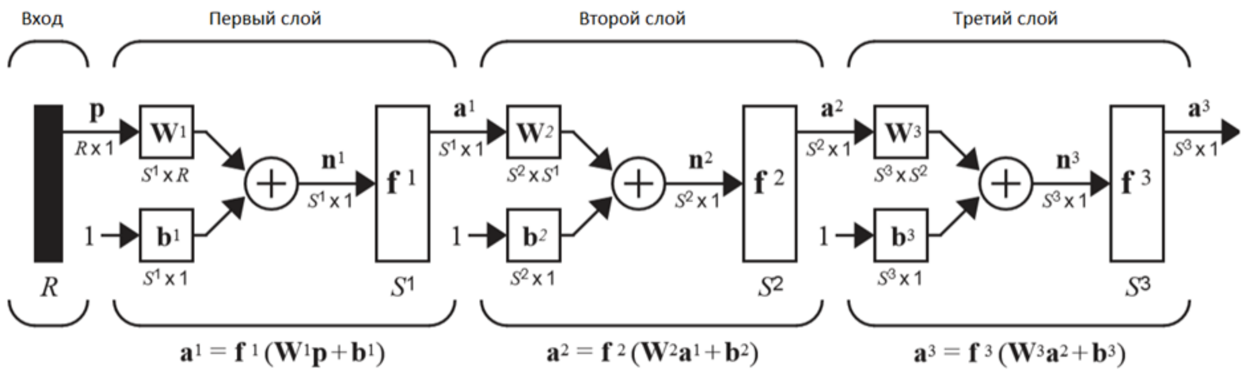


Рисунок 2.6 – Трехслойная сеть прямого распространения

Внутренние слои сети называются *скрытыми слоями* (hidden layers). Вход сети, представленный вектором **p**, называют *входным слоем*. Сокращенно структуру многослойной сети обозначают, указывая последовательно размерность входного слоя и количество нейронов в последующих слоях. Например, *структура сети*, изображенная на рисунке 2.6, обозначается в виде: R-S^1-S^2-S^3.

2.4. Производные активационных функций и инициализация весов нейросети

В ходе обучения нейронной сети выполняют минимизацию некоторой функции потерь (целевой функции) сети, которая определяется целью функционирования сети. Поиск минимума функции потерь сопряжен с её дифференцированием, что приводит к необходимости вычисления производных и распространению их значений в обратном направлении для определения чувствительности функции потерь к настраиваемым параметрам сети. Важно, чтобы при обратном распространении значения производных не становились нулевыми из-за зон насыщения функций активации, т.к. это приводит к потере эффективности рассматриваемых далее градиентных алгоритмов обучения.

Производные рассмотренных функций активации, базирующиеся на линейной зависимости, равны 0 в зоне насыщения и равны 1 на линейном участке (satlin, satlins, poslin, purelin). Производные сигмовидных функций вычисляются на основе следующих выражений [3]:

- производная логистической сигмовидной функции $\sigma(n)$ (logsig):

$$\frac{d\sigma(n)}{dn} = \sigma(n)(1 - \sigma(n)) \; ; \qquad (2.6)$$

- производная тангенциальной сигмовидной функции (гиперболический тангенс) $th(n)$ (tansig):

$$\frac{dth(n)}{dn} = (1 - th^2(n)). \qquad (2.7)$$

Под инициализацией нейронной сети подразумевается задание начальных значений матрицы весов **W**. Обычно начальным весам сети присваивают небольшие случайные значения, а смещениям – нули. Для этого могут непосредственно использоваться генераторы случайных чисел (rand, grand), рассмотренные в лабораторной работе 1. В модуле NeuralNetworks 2.0 пакета Scilab имеется встроенная функция ann_ffbp_init для инициализации многослойной сети с прямыми связями. Вызов этой функции осуществляется следующим образом:

W = ann_ffbp_init(N,r),

где N – вектор, задающий количество нейронов в каждом слое сети, включая входной и выходной слои; r– вектор, задающий желаемый диапазон изменения значений весов; W – список, элементами которого являются расширенные матрицы весов W(i) слоя i. Функция инициализирует веса сети путём вызова генератора случайных чисел grand с равномерным законом распределения. При этом векторы смещений b(i) каждого слоя получают нулевые значения и включаются в матрицу W(i) в виде её последнего столбца. Пример вызова функции инициализации для сети со структурой 3-3-2-1:

```
N = [3 3 2 1];
r = [-1 1];
W = ann_ffbp_init(N, r)
W  =
     W(1)
         0.9297771   0.4516779   0.9143339   0.
         0.9353899   0.9411856  -0.7802765   0.
        -0.6847738   0.9622194  -0.0292487   0.
    W(2)
```

```
      0.5962117  -0.4059411  -0.990433    0.
      0.6005609  -0.7162273  -0.1564774   0.
 W(3)
     -0.775071   0.831471   0.
```

3 Варианты заданий и программа работы

3.1. Повторить теоретический материал по источникам [1, 2, 5].

3.2. Выбрать вариант в соответствии с таблицей 2.2. Правила определения номера варианта изложены в лабораторной работе 1.

Таблица 2.2 – Варианты заданий

Вариант	Структура сети	Активационная функция 1	Активационная функция 2	Входной сигнал для i-го входа сети
1	100-2-1	logsig	purelin	sin(2*%pi*0.01*i*t)
2	100-4-1	logsig	satlin	sin(2*%pi*0.02*i*t)
3	100-6-1	logsig	satlins	sin(2*%pi*0.03*i*t)
4	100-8-1	logsig	poslin	sin(2*%pi*0.04*i*t)
5	100-10-1	tansig	purelin	sin(2*%pi*0.01*i*t)
6	100-12-1	tansig	satlin	sin(2*%pi*0.02*i*t)
7	100-14-1	tansig	satlins	sin(2*%pi*0.03*i*t)
8	100-16-1	tansig	poslin	sin(2*%pi*0.04*i*t)
9	50-4-1	logsig	purelin	sin(2*%pi*0.01*i*t)
10	50-8-1	logsig	satlin	sin(2*%pi*0.02*i*t)
11	50-16-1	logsig	satlins	sin(2*%pi*0.03*i*t)
12	50-32-1	logsig	poslin	sin(2*%pi*0.04*i*t)
13	50-16-1	tansig	purelin	sin(2*%pi*0.01*i*t)
14	50-8-1	tansig	satlin	sin(2*%pi*0.02*i*t)
15	50-4-1	tansig	satlins	sin(2*%pi*0.03*i*t)

3.3. Построить графики активационных функций и их производных в соответствии с вариантом задания, используя соответствующие встроенные функции модуля NeuralNetworks 2.0 пакета Scilab. В случае отсутствия указанных встроенных функции определить их самостоятельно на основе выражений, приведенных в таблице 2.1.

3.4. Реализовать две сети прямого распространения в соответствии с заданной структурой, запрограммировав вычисления в соответствии с выражением (2.5) для двух видов активационных функций, заданных по варианту.

3.5. Выполнить моделирование двух нейросетей, сгенерировав входные сигналы в соответствии с вариантом задания на интервале времени от 0 до $2/F$, где F – минимальная частота входного гармонического сигнала (например, для варианта 1 F=0.01Гц). При этом инициализацию сетей выполнить двумя способами:

1) случайными значениями из диапазона от -10 до +10;

2) случайными значениями из диапазона от -1/sqrt(R) до +1/sqrt(R), где R число входов нейрона.

3.6. Вычислить выходные значения всех нейронов сети. Построить графики активностей одного из нейронов скрытого слоя и нейрона выходного слоя, а также гистограмму общей активности всех нейронов скрытого слоя и отдельно гистограмму активности выходного нейрона для двух способов инициализации для каждой из двух реализованных сетей. Вычислить значение средней активности и стандартного отклонения на выходах нейронов скрытого и выходного слоёв.

3.7. Вычислить значения производных нелинейностей для всех нейронов в точках, соответствующих значениям входных сигналов этих нейронов. Построить графики вычисленных значений производных для одного из нейронов скрытого слоя и нейрона выходного слоя, а также гистограмму значений производных функции активации скрытых нейронов и выходного нейрона (для двух способов инициализации и для каждой из двух реализованных сетей). Вычислить среднее значение и стандартное отклонение производных для нейронов скрытого и выходного слоёв.

3.8. Выполнить анализ полученных результатов, обратив внимание на характер гистограмм. Сделать выводы о характере распределения значений активностей и значений производных для заданных активационных функций и каждого из способов инициализации. Сравнить результаты для каждой из двух реализованных сетей.

3.9. Подготовить и защитить отчет по работе.

4 Методические рекомендации по выполнению работы

4.1. Модуль NeuralNetwork 2.0 пакета Scilab имеет ограниченный набор встроенных функций активации и их производных:

a = ann_hardlim_activ(n) – единичная с жестким ограничением;
a = ann_logsig_activ(n) – логистическая, однополярная сигмоидальная;
a = ann_purelin_activ(n) – линейная;
a = ann_tansig_activ(n) – биполярная тангенциальная сигмоидальная;
d_a= ann_d_hardlim_activ(y) – производная единичной функции;
d_a = ann_d_logsig_activ(y) – производная логистической функции;
d_a = ann_d_purelin_activ(y) – производная линейной функции;
d_a = ann_d_tansig_activ(y) – производная биполярной сигмоидальной функции.

Остальные функции активации и их производные следует определять самостоятельно на основе выражений, указанных в таблице 2.1.

Обратите внимание на то, что функции ann_d_logsig_activ(y) и ann_d_tansig_activ(y) вычисляют производные на основе выражений (2.6) и (2.7). При этом входные аргументы этих функций должны представлять собой предварительно вычисленные значения однополярной сигмовидной функции ann_logsig_activ(n) или тангенциальной биполярной сигмовидной функции ann_tansig_activ(n).

Указанные выше встроенные функции допускают обработку входных аргументов, заданных векторами либо матрицами. В этом случае они применяются к векторам и матрицам поэлементно. Однако в силу специфики определения встроенной функции a = ann_logsig_activ(n) она не позволяет обрабатывать аргумент n, заданный прямоугольной матрицей. Поэтому рекомендуется переопределить эту функцию самостоятельно, воспользовавшись следующим выражением:

a=1 ./ (1+exp(-n)).

4.2. При выполнении задания 3.4 для реализации выражения (2.5) удобно входные векторы представлять матрицей **P**, в которой каждый столбец соответствует очередному входному вектору. В этом случае произведение **WP** будет матрицей, что потребует преобразования вектора смещений **b** в матрицу для обеспечения выполнения операции сложения при вычислении **n**. Преобразование **b** матрицу можно выполнить с помощью встроенной функции repmat пакета Scilab. Пример вызова этой функции см. ниже в п.4.3.

4.3. Для генерации матрицы **P** рекомендуется воспользоваться нижеследующим примером, в котором также показано как выполнить моделирование сети (задание 3.5) на примере одного слоя, построить графики активностей и производных, вычислить необходимые статистики (задание 3.6 и 3.7):

```
//задание исходных данных
F=0.01 //минимальная частота входного сигнала
R=5; //число входов слоя
S=3; // число нейронов слоя

//формирование матрицы p входных сигналов
t=0:0.1:2/F;
fi=(2*%pi*F).*t;
p=[sin(fi)];
for i=2:R
   p=[p; sin(fi*i)];
end

//инициализация весов и смещений
scale=1/sqrt(R);
w=grand(S,R,'unf',-scale,scale)
b=zeros(S,1);

//моделирование слоя с tansig нелинейностью
n=w*p+repmat(b,1,size(p,2))
a=ann_tansig_activ(n);

//вычисление производных tansig нелинейности
d_a=ann_d_tansig_activ(n);

//вычисление статистик
mean_a=mean(a)
```

```
stdev_a=stdev(a)
mean_d_a=mean(d_a)
stdev_d_a=stdev(d_a)

//построение графиков активностей, производных и гистограмм
clf(1);
figure(1);
subplot(2,2,1)
plot(t,a(1,:),t,a(2,:), t,a(3,:));
title('Активность нейронов слоя')
subplot(2,2,2)
plot(t,d_a(1,:),t,d_a(2,:), t,d_a(3,:));
title('Производные функции активации  слоя')
subplot(2,2,3)
histplot(20,a);
title('Гистограмма активности нейронов слоя')
subplot(2,2,4)
histplot(20,d_a);
title('Гистограмма производных функции активации')
```

Результаты вычислений представлены на рисунке 2.7.

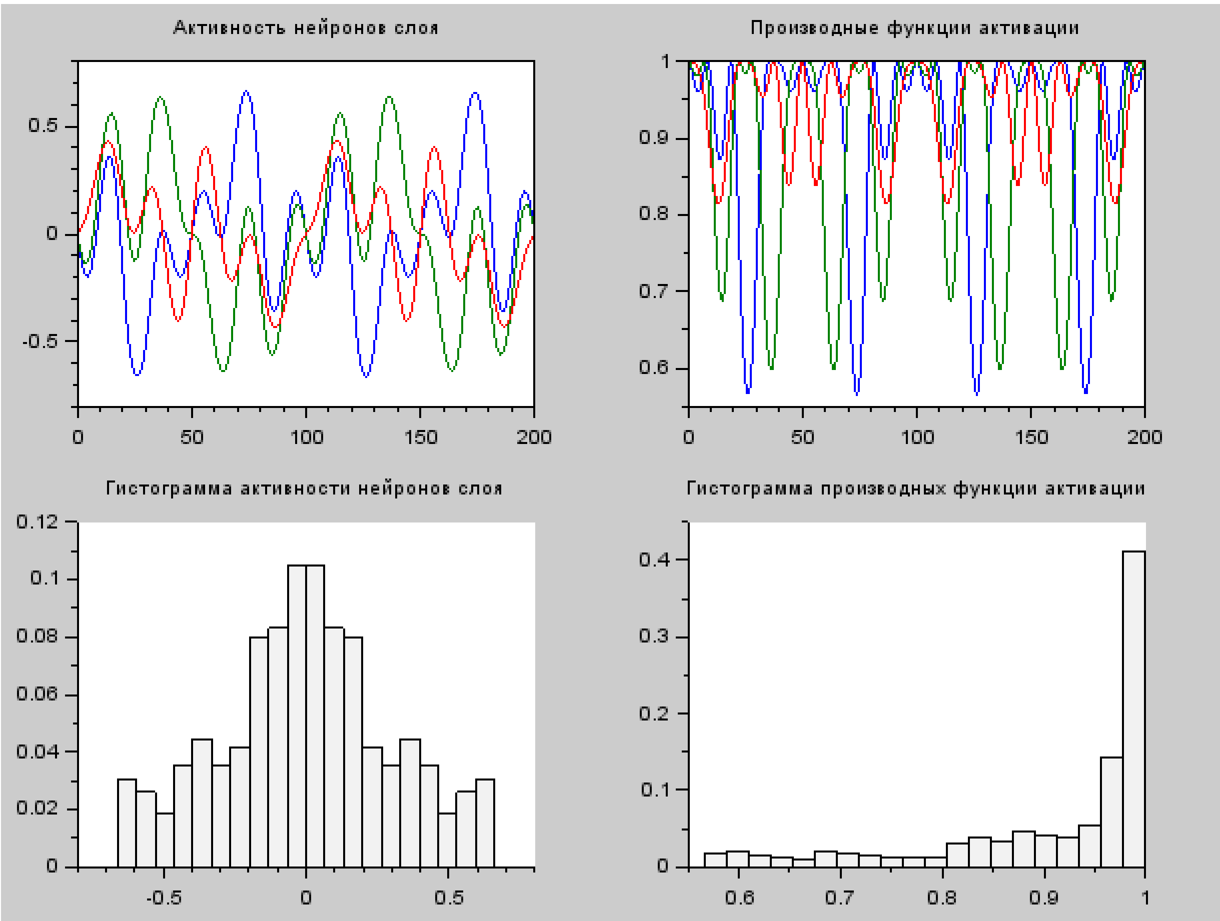


Рисунок 2.7 – Графики активности нейронов и производных функции активации, гистограммы

5 Содержание отчета

5.1. Цель работы.
5.2. Вариант задания.
5.3. Схема нейронной сети, формулы нелинейностей и их производных в соответствии с вариантом задания.
5.4. Листинги программ с комментариями.
5.5. Графики активности на выходе нейрона скрытого и выходного слоёв. Гистограммы, значения основных статистик активности.
5.6. Графики значений производных нелинейности на выходе нейронов скрытого и выходного слоёв. Гистограммы, значения основных статистик производных
5.7. Выводы по результатам исследований.

6 **Контрольные вопросы**

6.1. Нарисуйте схему нейронного элемента с векторным входом и запишите выражение для вычисления выходного значения.
6.2. Приведите выражение функции активации с жестким ограничением (функция Хевисайда) и ее графическое представление.
6.3. Приведите выражения для линейных функций активации (satlin, satlins) с насыщением и их графическое представление.
6.4. Приведите выражение для знаковой функции активации и ее графическое представление.
6.5. Приведите выражения для униполярной и биполярной сигмовидных функций активации и их графическое представление.
6.6. Приведите выражение для выходных сигналов НЭ с квадратической радиальной функцией.
6.7. Нарисуйте схему и приведите аналитическое выражение функции преобразования для линейного порогового элемента.
6.8. Продемонстрируйте на компьютере, каким образом проводились исследования функций активации?
6.9. Нарисуйте структурные схемы однослойной и многослойной искусственных нейронных сетей с прямыми связями.
6.10. Поясните понятия входного, скрытого и выходного слоев. Как сокращенно обозначают структуру сетей прямого распространения?
6.11. Приведите выражения для вычисления производных активационных функций, указанных в таблице 2.1.
6.12. Используя правила дифференцирования, выведите выражения для вычисления производных логистической и тагенциальной сигмовидных активационных функций.
6.13. Что такое иницализация нейронной сети и как её выполняют? Приведите примеры кода на языке Scilab.
6.14. Нарисуйте структурную схему 3-х слойной сети прямого распространения и запишите на языке Scilab фрагмент кода для вычисления её выходного значения, если последовательность входных векторов p представляется в виде матрицы.

6.15. Какой желательный характер распределений должны иметь выходные значения нейронов нейросети и почему? Как этого добиваются?

6.16. Почему нежелательно, чтобы производные активационных функций принимали нулевые значения?

Список рекомендованной литературы

1. Бондарев В.Н. Искусственный интеллект: Учеб. пособие для студентов вузов / В. Н. Бондарев, Ф. Г. Аде. — Севастополь: Изд-во СевНТУ, 2002. — 613 с.
2. Ерин С.В. Scilab – примеры и задачи: практическое пособие / С.В. Ерин – М.: Лаборатория «Знания будущего», 2017. – 154 с.
3. Медведев, В.С. Нейронные сети. MATLAB 6 / В.С. Медведев, В.Г. Потемкин; под общ. ред. В.Г. Потемкина. — М.: ДИАЛОГ-МИФИ, 2002. — 496 с.
4. Хайкин С. Нейронные сети: Полный курс. Пер. С англ. / С. Хайкин. — М.: Изд. «Вильямс», 2006. — 1104 с.
5. Hagan M.T. Neural Network Design. The 2nd edition [Электронный ресурс] /M.T.Hagan, H.B.Demuth, M.H.Beale, O.D. Jesus. . — Frisco, Texas, 2014 . — 1012 p. Режим доступа: https://www.hagan.okstate.edu/NNDesign.pdf. —Последний доступ: 14.01.2019. —Название с экрана.