

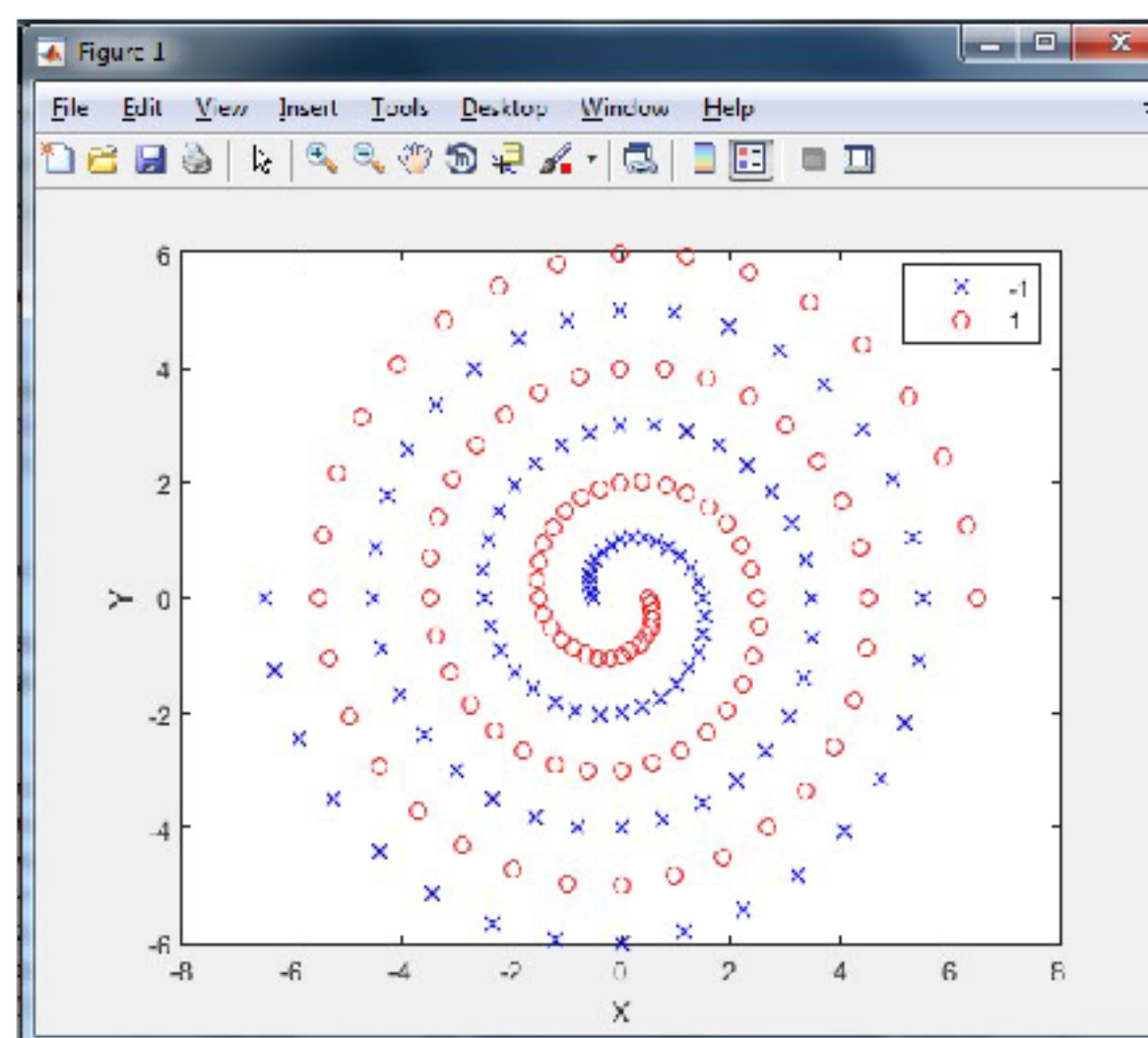
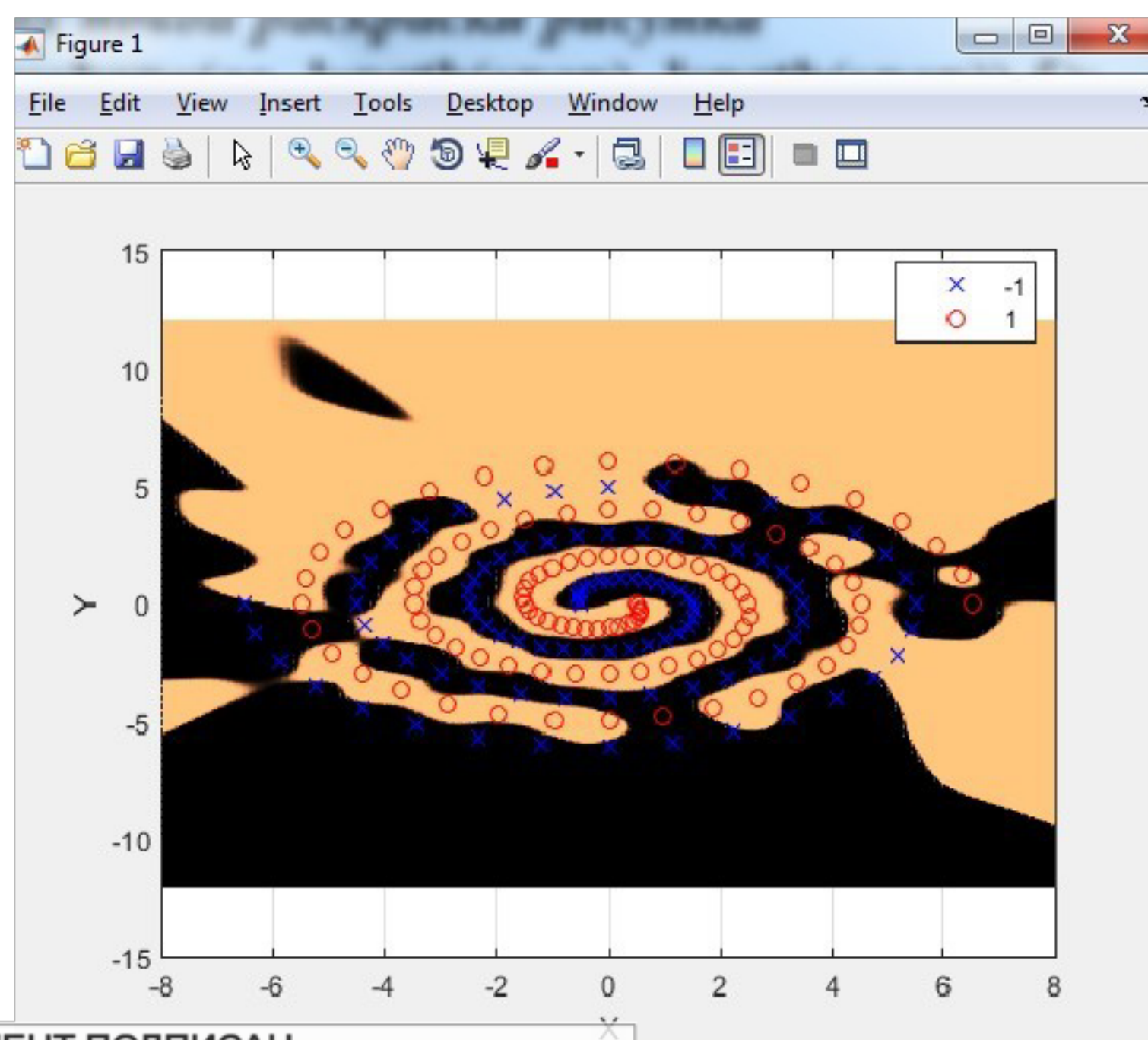


Рисунок 5.14 – Вывод точек спиралей для двух классов

Далее создаём сеть с помощью функции **newff(initSet, T, [60], {'tansig' 'tansig'}, 'trainlm')**. Мы задаём количество скрытых нейронов, функции активации и способ обучения. С помощью **net.trainParam.min_grad = 1e-9** мы устанавливаем минимальное значение нормы вектора градиента, достигнув которого, процесс обучения остановится. Дело в том, что алгоритм Левенберга-Марквардта (trainlm) быстро достигнет локального минимума, поэтому это значение можно и уменьшить, но тогда сеть может переобучиться, растеряв часть своих навыков.

Результат закрашивания квадратной области, в пределах которой находятся точки спирали, показан на рисунке 5.15.



Видно, что не все представители классов двух спиралей попали на свой цвет, что и не удивительно, ведь в процессе обучения использовалась не вся спираль, а только её часть (напомним, что задачу о спиральях часто используют для тестирования алгоритмов обучения: насколько хорошо будет построена в процессе достижения локального или глобального минимума на поверхности ошибок искомая спиральная поверхность отклика. Бить на обучающее и тестовое множество не обязательно, задача и так сложна).

Поведение сети на тесте и на обучающем множестве показано на рисунке 5.16.

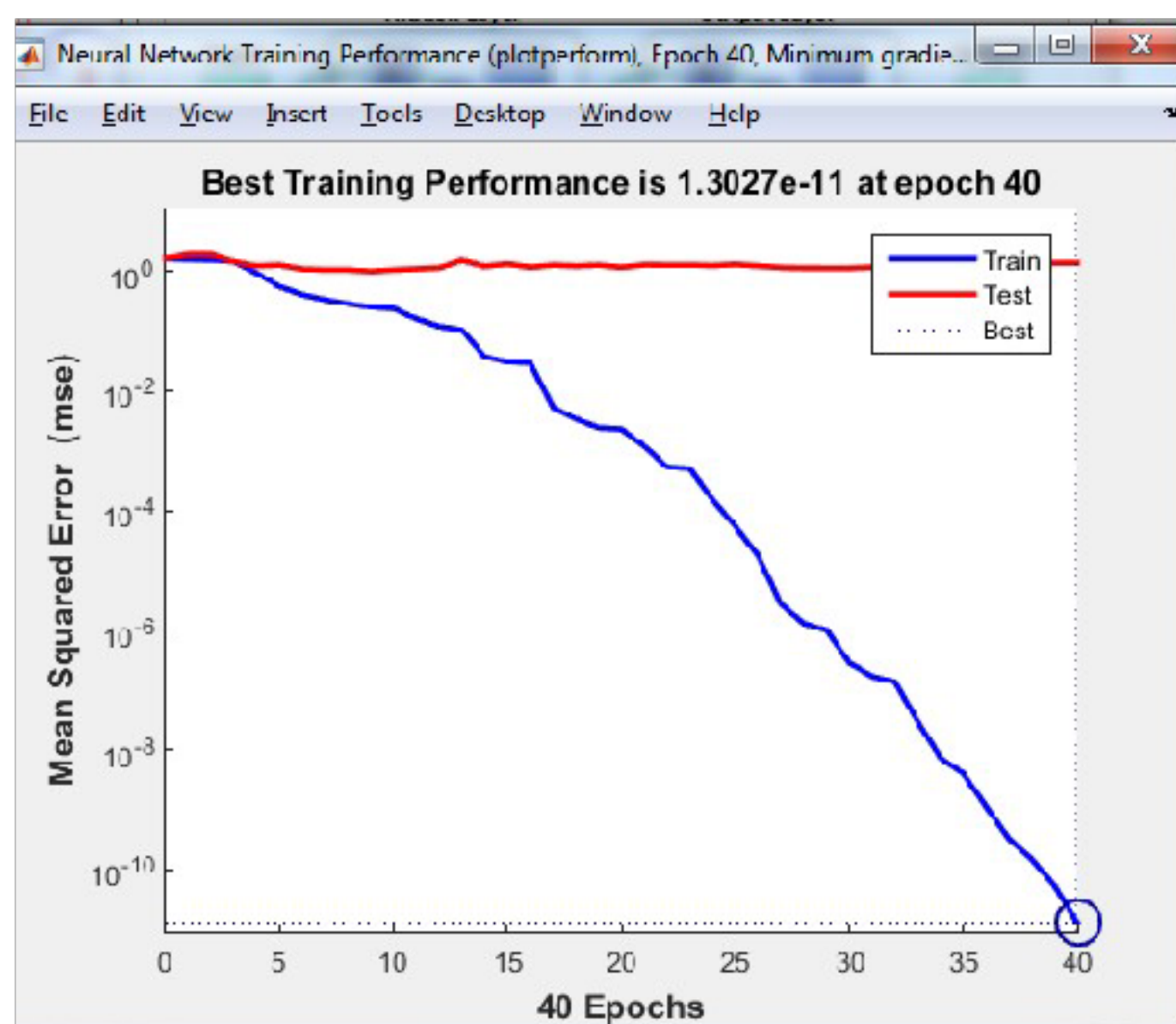


Рисунок 5.16 – Ошибка обучения и обобщения для 40 эпох

Из графиков видно, что снижать норму вектора градиента для продолжения обучения опасно, т.к. красный график (ошибка обобщения) потихоньку начинает ползти вверх.

В данной задаче тестовое множество специально не выделялось в коде, но индексы (29 штук), сгенерированные средой хранятся в **tr.testInd**, поэтому для получения конкретных ответов сети на тестовом множестве, можно написать **Q = sim(net, initSet(tr.testInd))**. Можно сравнить эти значения с ответами учителя: **T(tr.testInd)**. Из результатов сравнения становится видно, что даже несмотря на малую ошибку для обучающего множества сеть с большим трудом моделирует данную спиральную зависимость для тестового множества. Добиться для данной задачи необходимой поверхности отклика значительно проще, чем нужного ответа для тестового множества. С чем это связано?

В реальных задачах экземпляры классов обычно располагаются кучно друг по

отношению друг к другу, образуя кластеры в пространстве признаков. Связано это с тем, что за каждым классом стоит некий инвариант, который в той или иной степени

проявляется в каждом экземпляре. Поэтому, если обучающая и тестовая выборка составлены репрезентативно, то низкая ошибка на обучающем множестве в целом должна приводить к низкой ошибке на тестовом множестве, т.к. гиперплоскости должны быть настроены и отделять одни кластеры от других в процессе обучения. В данной же задаче поверхность кластеров не выпуклая, более того, кластер одного класса глубоко заходит в кластер другого класса. В таких условиях угадать какому кластеру принадлежит экземпляр очень сложно. Этот пример демонстрирует ту особенность, что если зависимость сложная или в данных много хаоса (не шума), то хорошая ошибка обучения вообще ничего не скажет о том, как поведёт себя сеть на тестовом множестве. Вот почему очень сложно строить модели для предсказаний на финансовых рынках или модели для задачи классификации вроде рассмотренной.

Далее сравним различные алгоритмы обучения на примере задачи о спирали. Разбивать на тестовую и обучающую выборки уже не будем. Задача упрощается: смоделировать поверхность отклика для сети (рисунок 5.7), соответственно критерием качества будет моделируемая поверхность, а также количество эпох, которое потребовалось на это.

Подробную справку по алгоритмам обучения можно получить через команду **help nntrain**.

Рассмотрим следующие алгоритмы: **trainlm** (алгоритм Левенберга-Марквардта), **trainbfg** (метод BFGS), **traingd** (обычный алгоритм обратного распространения), **traingdm** (алгоритм обратного распространения с моментом), **trainrp** (алгоритм RPROP).

Для более подробного ознакомления рекомендуется [4–6].

Обучения ИНС – это поиск минимума на поверхности ошибок. Все методы обучения можно разбить на два класса: локальные и глобальные. Глобальные пытаются найти глобальный минимум, самый известный класс из таких методов – это генетические алгоритмы. Локальные методы ориентированы на пошаговый спуск из некоторой точки на поверхности ошибок к локальному минимуму. Локальные методы можно разбить на три класса в зависимости от того какая информация ими используется для спуска: методы нулевого порядка (не используют производные), методы первого порядка (используют первые производные, - обычно для получения вектора градиента), методы второго порядка (используют информацию от вторых производных (матрица Гесса, матрица Якоби)). В целом зависимость такая: чем выше порядок метода, тем он считается эффективнее для

спуска, но требует больше вычислений и памяти для хранения дополнительных структур (матриц). Если сети большие, то придётся отказаться от методов второго порядка, т.к. потребуется слишком много памяти и времени на обучение.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Глобальные методы часто используются для преднастройки сети перед использованием локальных методов.

Алгоритмы **trainlm** и **trainbfg** – алгоритмы второго порядка, все остальные – первого порядка. Между рассматриваемыми алгоритмами можно выделить следующую взаимосвязь, таблица 5.3.

Таблица 5.3 – Взаимосвязь между рассматриваемыми алгоритмами

№	Тип алгоритма	Скорость, эффективность	Память
1	Trainlm	Самый быстрый, во многих функциях стоит по умолчанию	Расходует больше всех памяти
2	Trainbfg	Более медленный, чем Trainlm	Расходует меньше памяти, чем Trainlm
3	Trainpr	Более медленный, чем Trainbfg	Расходует меньше памяти, чем Trainbfg
4	Traingdm	Очень медленный	Расходует меньше памяти, чем Trainpr
5	Traingd	Очень медленный	Расходует меньше памяти, чем Trainpr, отличий от Traingdm практически нет

Листинг по применению алгоритма **trainlm** приведён ниже. Стоит обратить внимание, что здесь использована практически полная форма вызова функции **newff()**, а также установлен параметр **net.trainParam.epochs = 60**. Дело в том, что если не устанавливать этот параметр, то алгоритм остановит свою работу, когда модуль вектора градиента станет близким к нулю, но, если при этом изучить график ошибки обучения (синий цвет), то окажется, что при использовании этого алгоритма минимум достигается очень быстро (в среднем 50-60 эпох), а остальные силы (порядка 150 эпох) идут на «топтанье на месте» (приближение к асимптоте). В результате этого процесса сеть всё больше становится похожа на память и всё больше теряет обобщающие способности, поэтому имеет смысл остановить этот процесс. Получившаяся поверхность отклика представлена на рисунке 5.17.

Переменная **pref** содержит результирующую ошибку обучения (разницу между метками учителя и реальными ответами сети).


```

студентов\Лабы\5\DataForSpiral.xlsx', 1, 'D:D');
gscatter(initSet(:, 1), initSet(:, 2), T, 'br', 'xo');
xlabel('X');
ylabel('Y');
initSet = initSet';
T = T';
% Вместо learngd можно применить learnngdm, вместо mse
% msereg – с регуляризацией, crossentropy – кросс-энтропия
net = newff(initSet, T, [60], {'tansig' 'tansig'}, 'trainlm', 'learnngd', 'mse', {}, {}, 'dividerand');
net.divideParam.trainRatio = 1;
net.divideParam.testRatio = 0;
net.divideParam.valRatio = 0;
net.trainParam.epochs = 60;
net = init(net);
[net, tr] = train(net, initSet, T);
% один из способов получения ответов сети
y = net(initSet);
% высчитываем разницу между ожидаемыми и реальными ответами
% PREF ≈ 0.0014
perf = perform(net, T, y);
span = -12:.05:12;
[P1, P2] = meshgrid(span, span);
pp = [P1(:) P2(:)]';
aa = net(pp);
figure(1)
hold on;
grid on;
mesh(P1, P2, reshape(aa, length(span), length(span))-5);
colormap copper

```

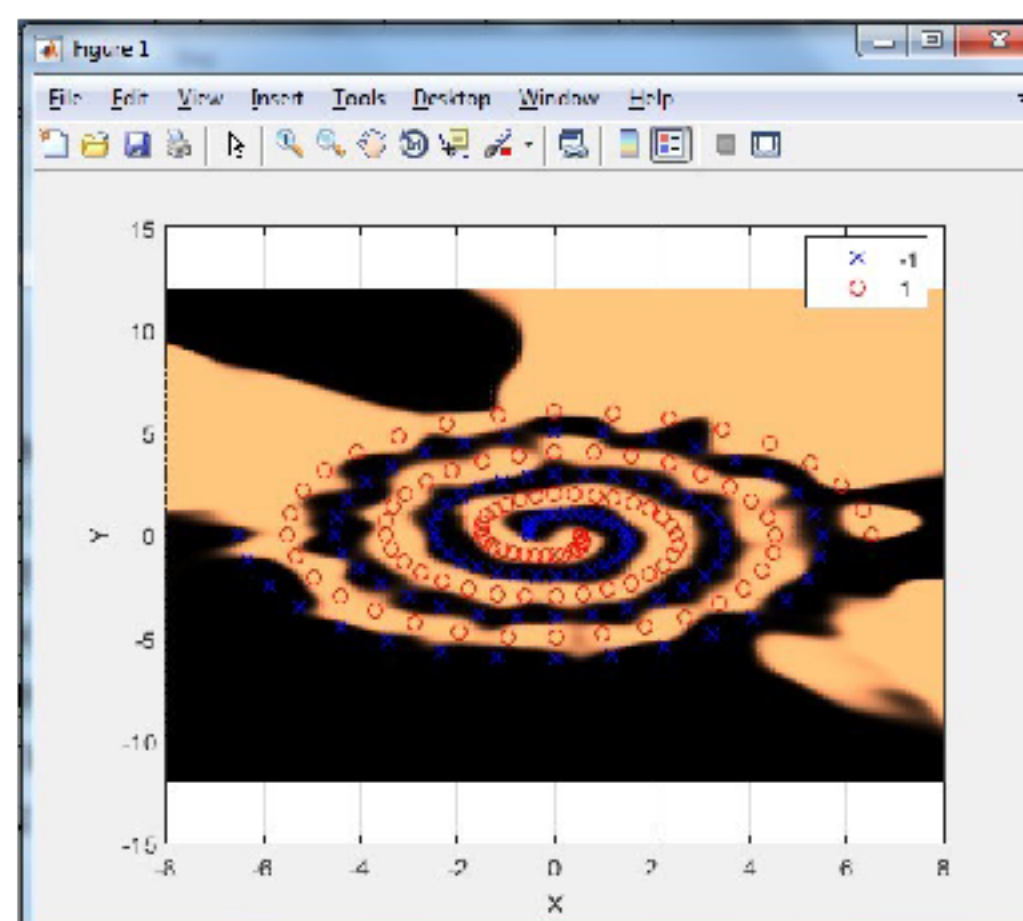


Рисунок 5.17– Поверхность отклика после применения алгоритма trainlm

Ниже приведён листинг для использования алгоритм **trainbfg**. Как видно, было изменено количество эпох, которые примерно потребуются для нахождения решения. Это количество было снижено до 50. Количество нейронов в скрытом слое также было снижено до 50.

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Найденное решение показано на рисунке 5.18. Изменение ошибки обучения

показано на рисунке 5.19.

```
initSet = xlsread('C:\Users\Computer  
студентов\Лабы\5\DataForSpiral.xlsx', 1, 'B:C');  
T = xlsread('C:\Users\Computer  
студентов\Лабы\5\DataForSpiral.xlsx', 1, 'D:D');  
gscatter(initSet(:, 1), initSet(:, 2), T, 'br', 'xo');  
xlabel('X');  
ylabel('Y');  
initSet = initSet';  
T = T';  
net = newff(initSet, T, [50], {'tansig' 'tansig'}, 'trainbfg', 'learnngdm', 'mse', {}, {},  
'dividerand');  
net.divideParam.trainRatio = 1;  
net.divideParam.testRatio = 0;  
net.divideParam.valRatio = 0;  
net.trainParam.epochs = 250;  
net = init(net);  
[net, tr] = train(net, initSet, T);  
y = net(initSet);  
% PREF == 7.4596e-08;  
perf = perform(net, T, y);  
span = -12:.05:12;  
[P1, P2] = meshgrid(span, span);  
pp = [P1(:) P2(:)]';  
aa = net(pp);  
figure(1)  
hold on;  
grid on;  
mesh(P1, P2, reshape(aa, length(span), length(span))-5);  
colormap copper
```

Grand\Desktop\Для
Grand\Desktop\Для

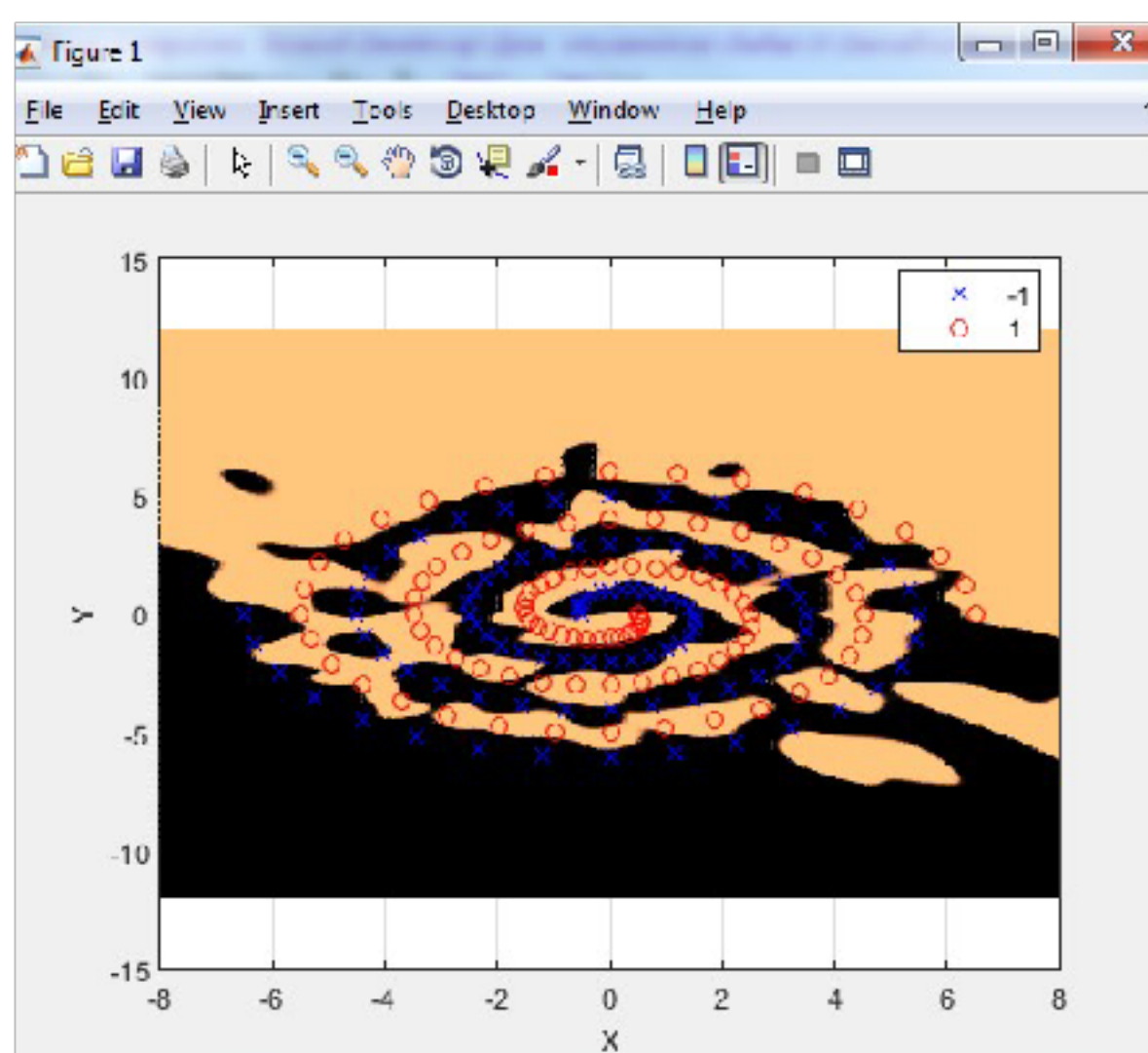


Рисунок 5.18 – Найденное решение с использованием алгоритм trainbfg

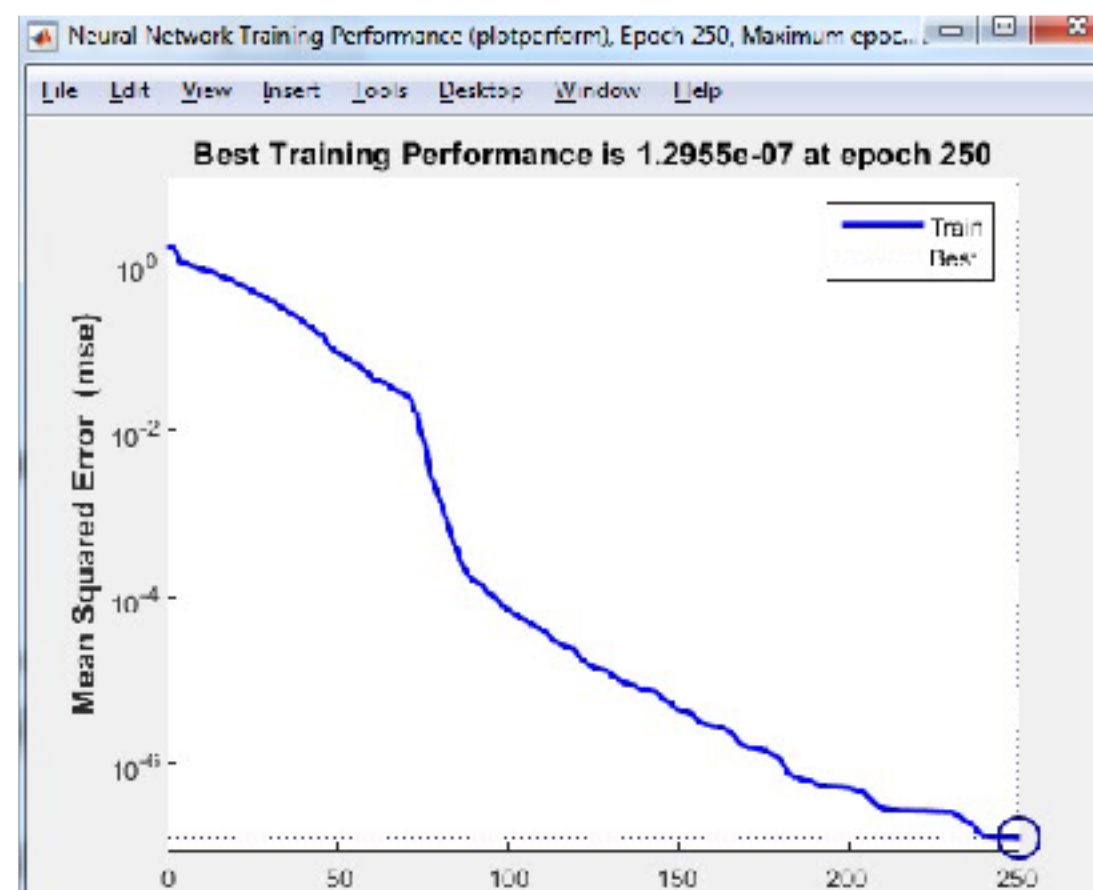


Рисунок 5.19 – Изменение величины ошибки обучения за 250 эпох

Далее рассмотрим применение алгоритма **trainrp**, но вместо функции **newff()** будем использовать **feedforwardnet()**. В пакете matlab очень часто существует иерархия функций, которые выполняют, по сути, одно и то же, но с разной степенью детализации для пользователя. Для создания сетей прямого распространения можно выделить условно следующую иерархию: **network()** → **newff()** → **feedforwardnet()** → **patternnet()**.

Ниже приведён листинг для обучения с помощью алгоритма **trainrp** (RPROP). Стоит обратить внимание, что количество эпох уже исчисляется тысячами. Сеть создавали с помощью функции **feedforwardnet()**, которая получает в качестве входных параметров количество нейронов в скрытом слое и алгоритм обучения. У данного алгоритма обычно настраивают такие параметры обучения как **net.trainParam.delt_inc**, **net.trainParam.delt_dec**, **net.trainParam.delta0**, **net.trainParam.deltamax**, **net.trainParam.lr**, однако, учитывая, что для осмысленного регулирования этих параметров необходимо знание тонкостей работы алгоритмы, мы их оставили настроенными по умолчанию. Для ознакомления с этими параметрами можно использовать команду **help trainrp**. Полученное решение показано на рисунке 5.20, а изменение значения ошибки обучения на рисунке 5.21.

```
initSet = xlsread('C:\Users\Computer
студентов\Лабы\5\DataForSpiral.xlsx', 1, 'B:C');
T = xlsread('C:\Users\Computer
студентов\Лабы\5\DataForSpiral.xlsx', 1, 'D:D');
gscatter(initSet(:, 1), initSet(:, 2), T, 'br', 'xo');
xlabel('X');
ylabel('Y');
initSet = initSet';
T = T';
```

Grand\Desktop\Для

Grand\Desktop\Для

net = feedforwardnet(60, 'trainrp');
net = configureNet(net, initSet, T);

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

net.divideParam.trainRatio = 1;

Действителен: с 20.08.2021 по 20.08.2022

```

net.divideParam.testRatio = 0;
net.divideParam.valRatio = 0;
net.trainParam.epochs = 6000;
net = init(net);
[net, tr] = train(net, initSet, T);
y = net(initSet);
perf = perform(net, T, y);
% PREF == 1.7276e-05.
span = -12:.05:12;
[P1, P2] = meshgrid(span, span);
pp = [P1(:) P2(:)]';
aa = net(pp);
figure(1)
hold on;
grid on;
mesh(P1, P2, reshape(aa, length(span), length(span))-5);
colormap copper

```

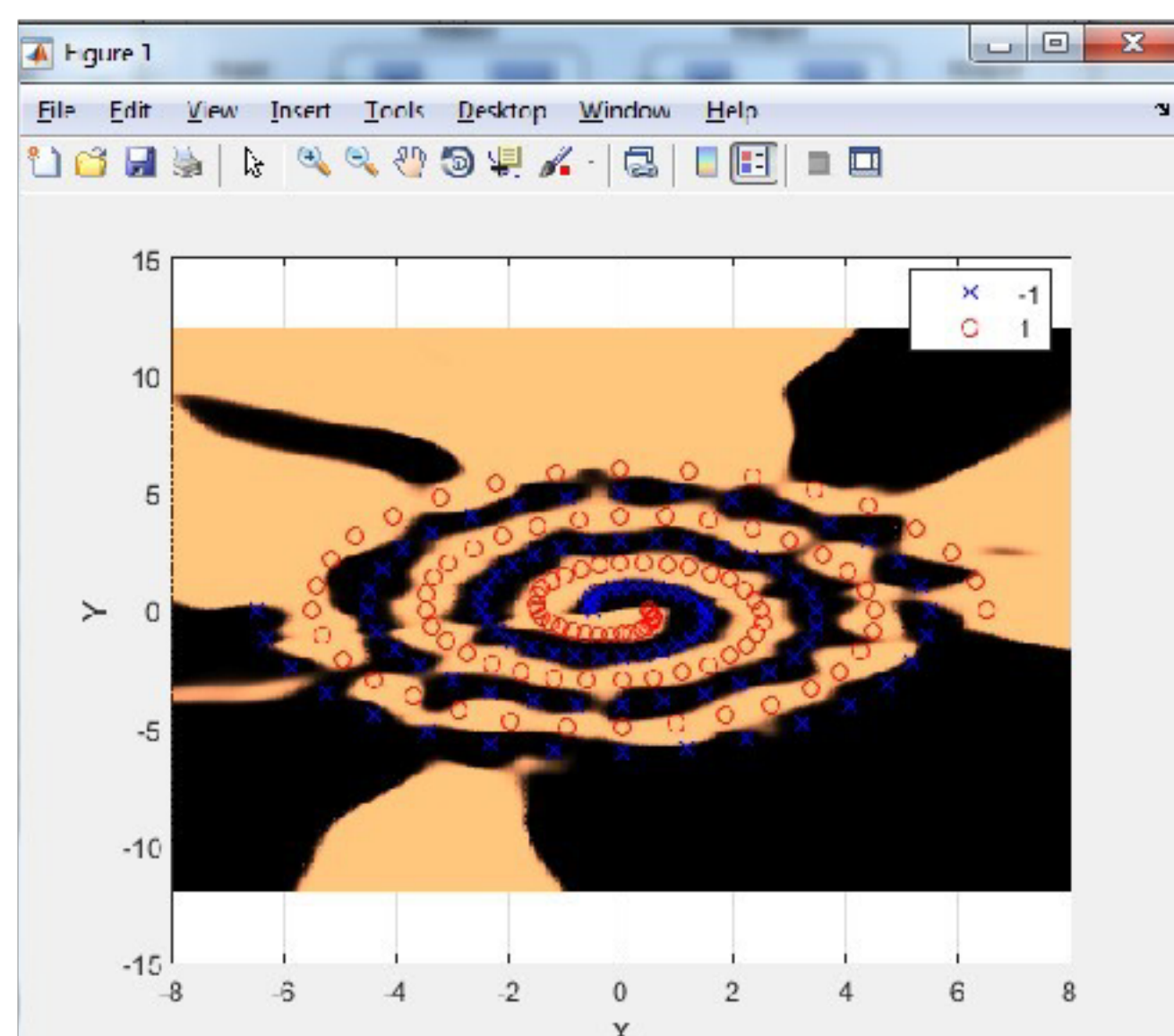
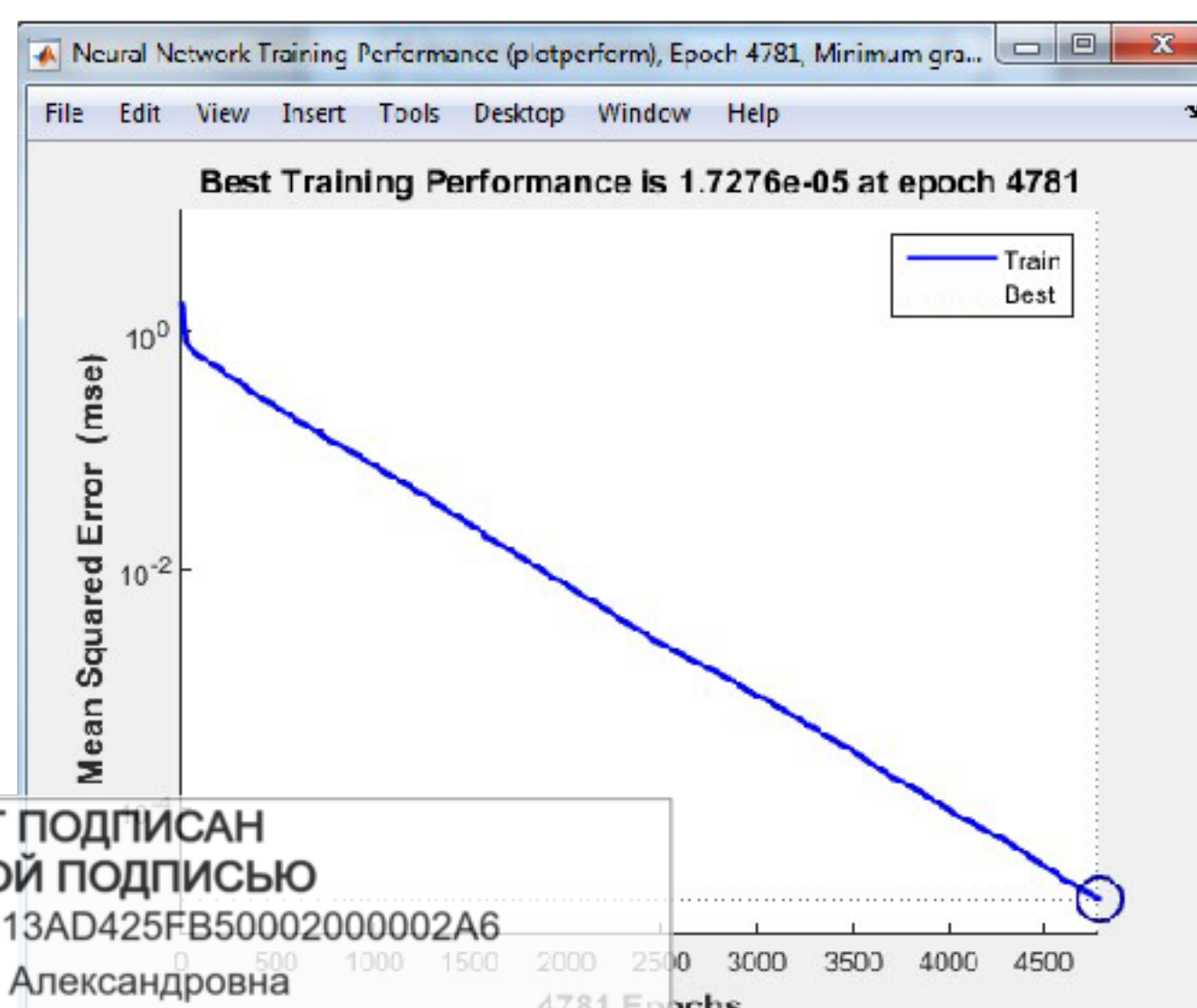


Рисунок 5.20 – Найденное решение с использованием алгоритма trainrp



ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

Рисунок 5.21 – Изменение величины ошибки обучения за 5000 эпох

Далее рассмотрим сеть с двумя скрытыми слоями для решения задачи о спиралях.

Формулы (5.2) и (5.3) тут также не подойдут для выбора размеров скрытых слоёв.

Ориентируясь на различные источники выберем структуру 2-10-10-1.

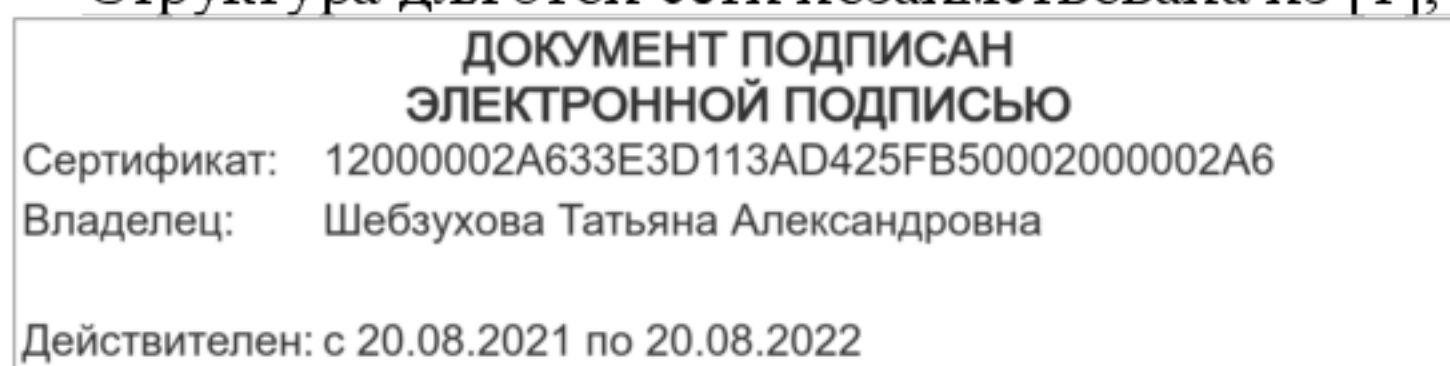
Листинг обучения такой сети приведён ниже.

```
initSet = xlsread('C:\Users\Computer
студентов\Лабы\5\DataForSpiral.xlsx', 1, 'B:C');
T = xlsread('C:\Users\Computer
студентов\Лабы\5\DataForSpiral.xlsx', 1, 'D:D');
gscatter(initSet(:, 1), initSet(:, 2), T, 'br', 'xo');
xlabel('X');
ylabel('Y');
initSet = initSet';
T = T';
net = feedforwardnet([10 10], 'trainlm');
net = configure(net, initSet, T);
net.layers{3}.transferFcn = 'tansig';
net.divideParam.trainRatio = 0.85;
net.divideParam.testRatio = 0.15;
net.divideParam.valRatio = 0;
net.trainParam.epochs = 6000;
net.trainParam.min_grad = 1e-10;
net = init(net);
[net, tr] = train(net, initSet, T);
y = net(initSet);
perf = perform(net, T, y);
span = -12:.05:12;
[P1, P2] = meshgrid(span, span);
pp = [P1(:) P2(:)]';
aa = net(pp);
figure(1)
hold on;
grid on;
mesh(P1, P2, reshape(aa, length(span), length(span))-5);
colormap copper
Q = sim(net, initSet(tr.testInd))
T(tr.testInd)
```

Подсчитав количество совпадений между векторами Q и T найдём, что качество работы сети выросло на неизвестных примерах с 28% до 50%.

Теперь создадим пользовательскую сеть, которая решает задачу о спирали.

Структура для этой сети позаимствована из [1], рисунок 5.22.



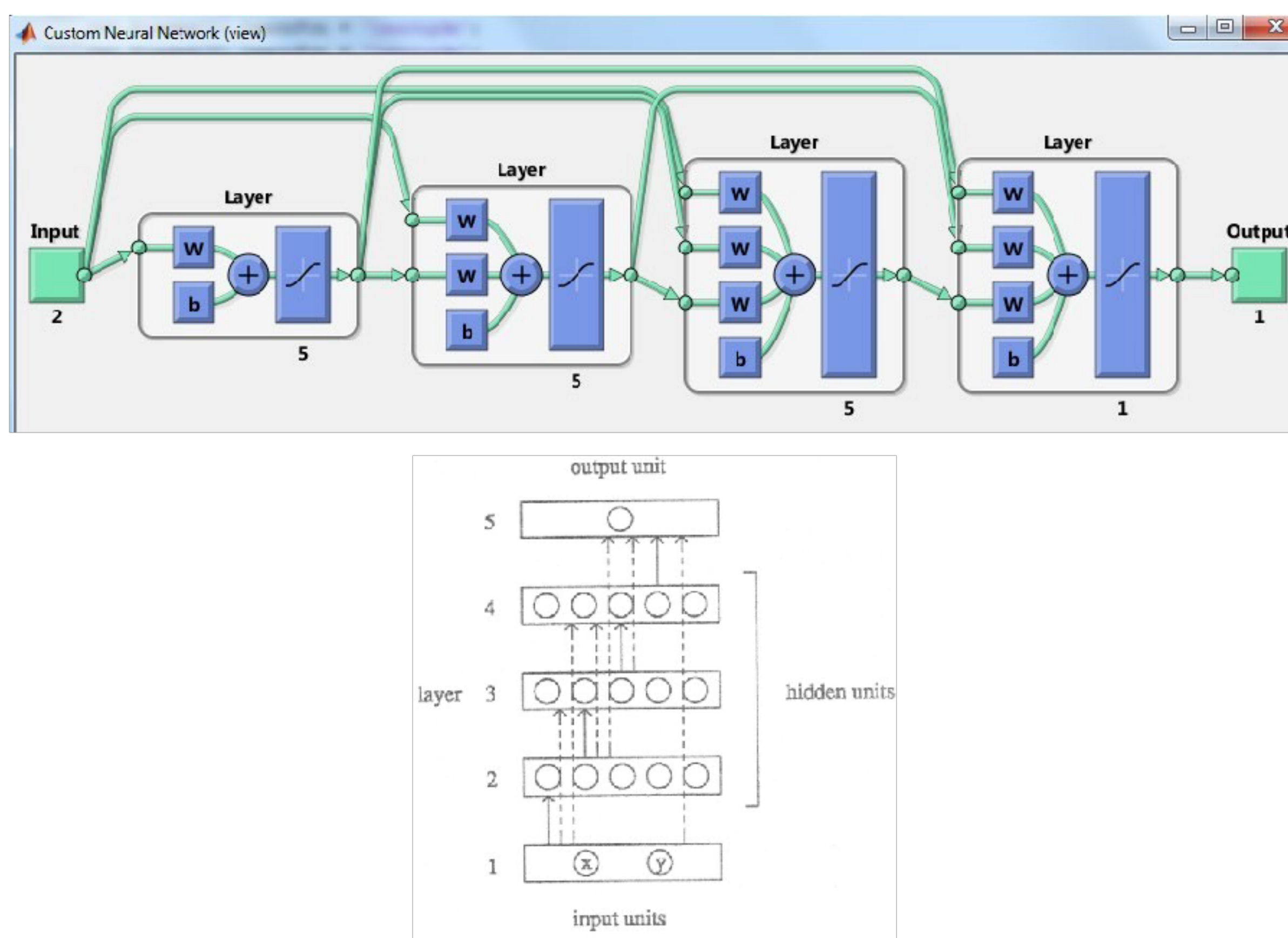


Рисунок 5.22 – Сверху структура предлагаемой сети в Matlab, снизу она же в виде обобщённой схемы

Как пишут авторы статьи, такая структура была выбрана потому, что они допустили, что каждый синапс-вес может с лёгкостью запомнить-хранить 1.5 бит информации (судя по всему предполагается, что вещественное число может в частности принять три значения: два концевых, допустим, 0 и 1, и одно промежуточное, допустим, 0.5. Таким образом, это не два бита информации, но и не один). Всего в выборке 194 паттерна и для их хранения им потребуется $194 / 1.5 = 130$ весов. Далее они подбирали сеть с несколькими слоями, чтобы она содержала примерно 130 весов. Сеть на рисунке 5.22 имеет 136 параметров. Можно это число получить сразу (после создания сети набрать **net.numWeightElements**) или подсчитать вручную: первый слой имеет два нейрона и связан с тремя другими слоями $((5+5)*3)$, второй слой связан с тремя слоями, два из них по 5 нейронов и один содержит один нейрон, сам же этот первый скрытый слой имеет тоже 5 нейронов $(25+25+5)$, третий слой связан с двумя слоями $(25+5)$, и для четвёртого слоя – 5. Также учитываем количество смещений для всех скрытых слоёв и выходного слоя – 16. Получаем $120+16=136$. Что касается связей через слой, то они нужны, чтобы частично скомпенсировать затухание

– вынужденная мера. Стоит отметить, что вообще с затуханием градиента борются через введение общих весов (сверточные сети).

Листинг программы с подробными комментариями приведен ниже. Стоит обратить внимание на применение функции **network()**. В лабораторной работе 4 объяснялись её параметры.

```
initSet = xlsread('C:\Users\Computer\студентов\Лабы\5\DataForSpiral.xlsx', 1, 'B:C');
T = xlsread('C:\Users\Computer\студентов\Лабы\5\DataForSpiral.xlsx', 1, 'D:D');
gscatter(initSet(:, 1), initSet(:, 2), T, 'br', 'xo');
xlabel('X');
ylabel('Y');
initSet = initSet';
T = T';
% создание пользовательской сети
net = network(1, 4, [1; 1; 1; 1], [1; 1; 1; 0], [0 0 0 0; 1 0 0 0; 1 1 0 0; 1 1 1 0], [0 0 0 1]);
% размеры скрытых слоёв
net.layers{1}.size = 5;
net.layers{2}.size = 5;
net.layers{3}.size = 5;
% функции активации на скрытых слоях
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.layers{3}.transferFcn = 'tansig';
% функция активации на выходном слое
net.layers{4}.transferFcn = 'tansig';
% способ разделения обучающего множества: случайно
net.divideFcn = 'dividerand';
net.divideParam.trainRatio = 1;
net.divideParam.testRatio = 0;
net.divideParam.valRatio = 0;
% алгоритм обучения RPROP, нужно учесть, что чем сложнее
% структура пользовательской сети, тем более осторожно
% нужно применять мощные алгоритмы, т.к. они могут не сработать
% в силу отсутствия реализации некоторых тонкостей настройки
% этих алгоритмов к пользовательским архитектурам.
% trainrp, traingd, traingdm – как раз простые алгоритмы, которые
% подойдут практически ко всему
net.trainFcn = 'trainrp';
% функция ошибки – mse, ставить, допустим, crossentropy удастся
% не всегда, т.к. многие алгоритмы рассчитаны на mse или msereg
net.performFcn = 'mse';
% максимальное количество эпох, ставим побольше
net.trainParam.epochs = 35000;
net.trainParam.goal = 0;
% начальная скорость обучения
net.trainParam.lr = 0.01;
% настройки RPROP
net.trainParam.delta = 1.2;
```

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

```

net.trainParam.delt_dec = 0.5;
net.trainParam.delta0 = 0.07;
net.trainParam.deltamax = 50.0;
% минимальный размер модуля градиента для остановки алгоритма
% для более мощных алгоритмов можно установить 1e-10, для слабых
% от 1e-5 до 1e-7
net.trainParam.min_grad = 1e-7;
% подключаем график изменения ошибки обучения
net.plotFcns = {'plotperform'};
% чтобы на пользовательской архитектуре работал алгоритм
% обратного распространения ошибки, нужно вручную для слоёв
% установить инициализацию весов и правило обновления-обучения
% установка правила обучения для весов
net.adaptFcn = 'adaptwb';
% net.layerWeights{i, j} – это матрица 4x4, показывающая какой слой
% с каким связан, в ней заполнены все элементы ниже главной
% диагонали, кроме элемента {4, 4}, поэтому для них для всех нужно
% установить правило обучения
net.layerWeights{1}.learnFcn = 'learngdm';
net.layerWeights{2}.learnFcn = 'learngdm';
net.layerWeights{3}.learnFcn = 'learngdm';
net.layerWeights{3,2}.learnFcn = 'learngdm';
net.layerWeights{4}.learnFcn = 'learngdm';
net.layerWeights{4,2}.learnFcn = 'learngdm';
net.layerWeights{4,3}.learnFcn = 'learngdm';
% тоже, но для смещений
net.biases{1}.learnFcn = 'learngdm';
net.biases{2}.learnFcn = 'learngdm';
net.biases{3}.learnFcn = 'learngdm';
net.biases{4}.learnFcn = 'learngdm';
% для входного слоя
net.inputWeights{1}.learnFcn = 'learngdm';
% этот параметр сообщает, что при применении configure(), веса
% будут инициализированы с помощью выбранного нами параметра,
% мы выбрали initnw, т.е. инициализация по методу Нгуена-Видроу
net.initFcn = 'initlay';
net.layers{1}.initFcn = 'initnw';
net.layers{2}.initFcn = 'initnw';
net.layers{3}.initFcn = 'initnw';
net.layers{4}.initFcn = 'initnw';
net.biases{1}.initFcn = 'initnw';
net.biases{2}.initFcn = 'initnw';
net.biases{3}.initFcn = 'initnw';
net.biases{4}.initFcn = 'initnw';
% устанавливаем величину момента
net.trainParam.mc = 0.5;
% конфигурируем сеть, т.е. применяем к ней все наши настройки
net = configure(net, initSet, T);

```

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шибзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022


```

perf = perform(net, T, y);
span = -12:.05:12;
[P1, P2] = meshgrid(span, span);
pp = [P1(:) P2(:)]';
aa = net(pp);
figure(1)
hold on;
grid on;
% выводим полученное решение
mesh(P1, P2, reshape(aa, length(span), length(span))-5);
colormap copper

```

На рисунках 5.23–5.25 приведены результаты обучения этой сети.

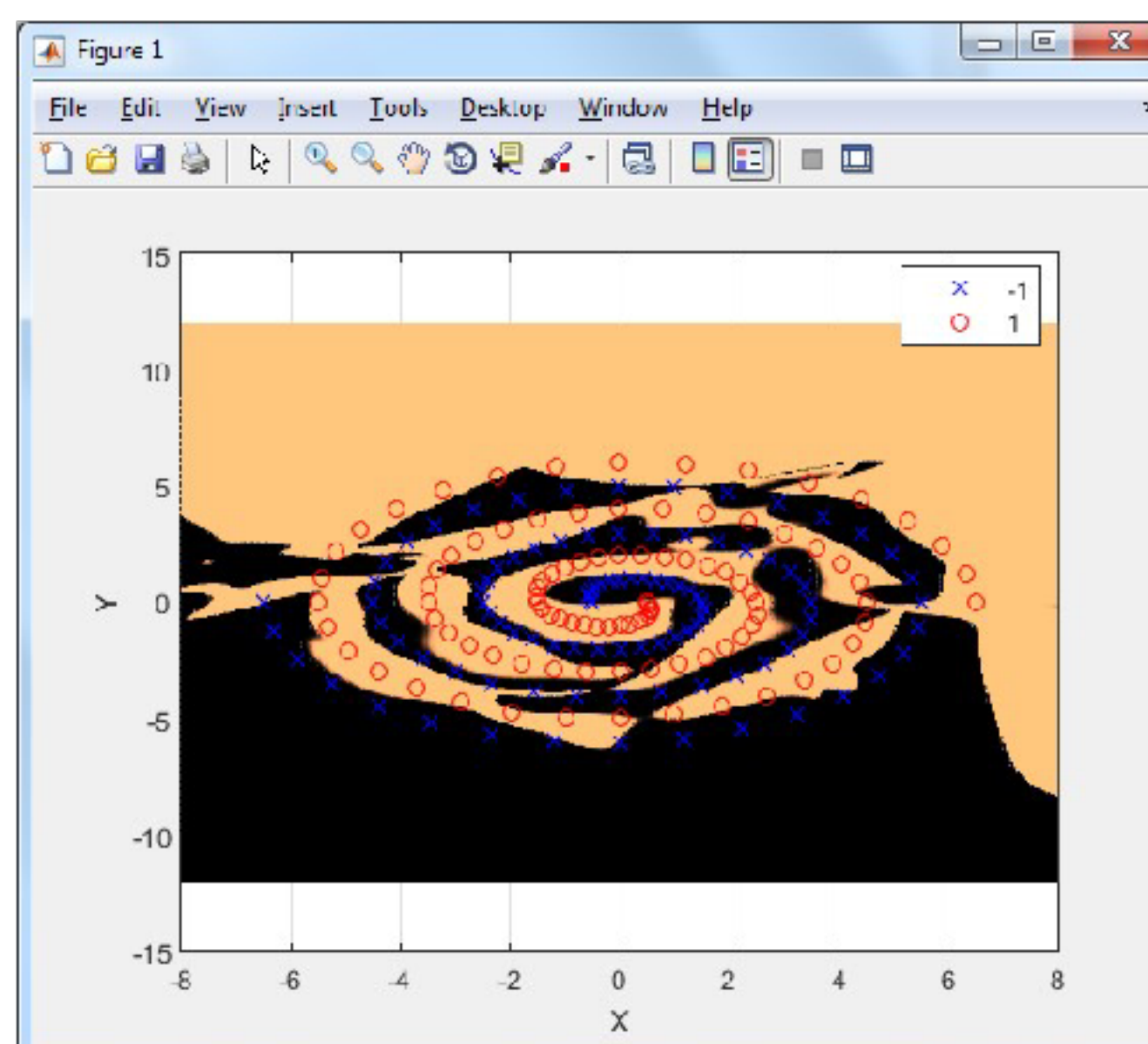


Рисунок 5.23 – Полученная поверхность отклика для пользовательской сети

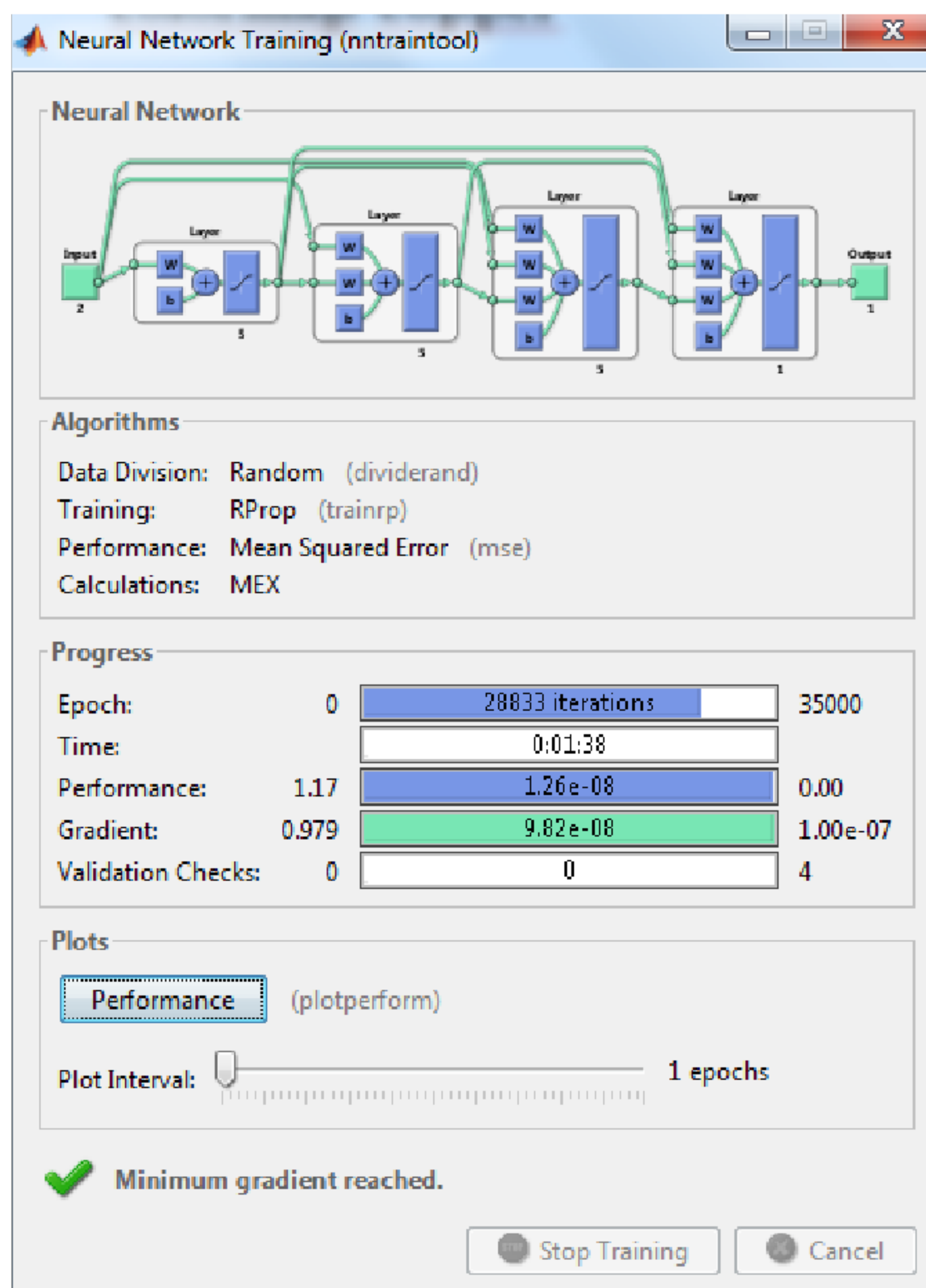
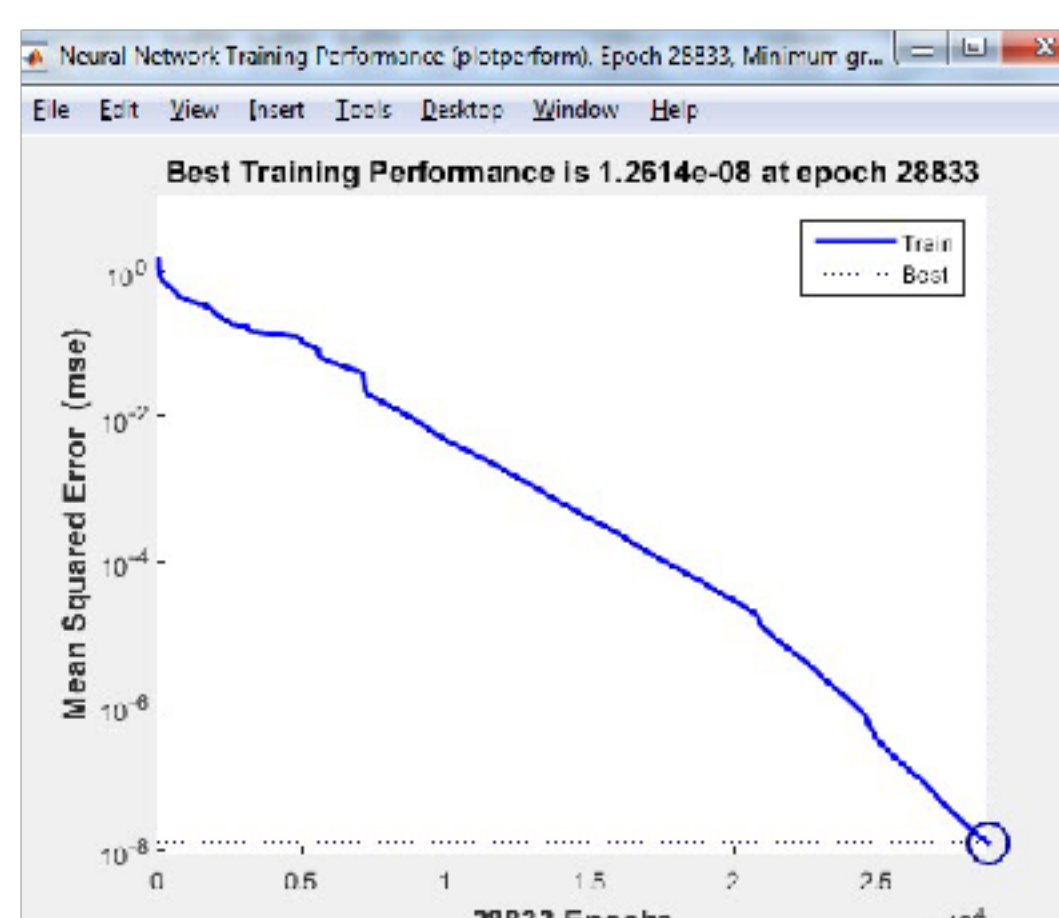


Рисунок 5.24 – Результирующие параметры обучения



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Изменив `net.divideParam.trainRatio = 0.85`, `net.divideParam.testRatio = 0.15` можно

Действителен: с 20.08.2021 по 20.08.2022

ещё раз обучить сеть, подсчитать число совпадений между векторами Q и T (как в первом листинге). Получилось порядка 65%.

Теперь можно сделать общие выводы по всем трём сетям. Будем оценивать ресурсы, обобщающую способность, аппроксимационную способность. Под ресурсами понимается количество синаптических весов, т.е. количество настраиваемых параметров. Так как мы проводили эксперименты с разным количеством нейронов для первой сети с одним скрытым слоем, то для определённости пусть будет 50 нейронов в скрытом слое. Под аппроксимационной способностью подразумевается способность построить необходимую поверхность отклика. Результаты приведены в таблице 5.4.

Таблица 5.4 – Сравнение трёх сетей с различной структурой

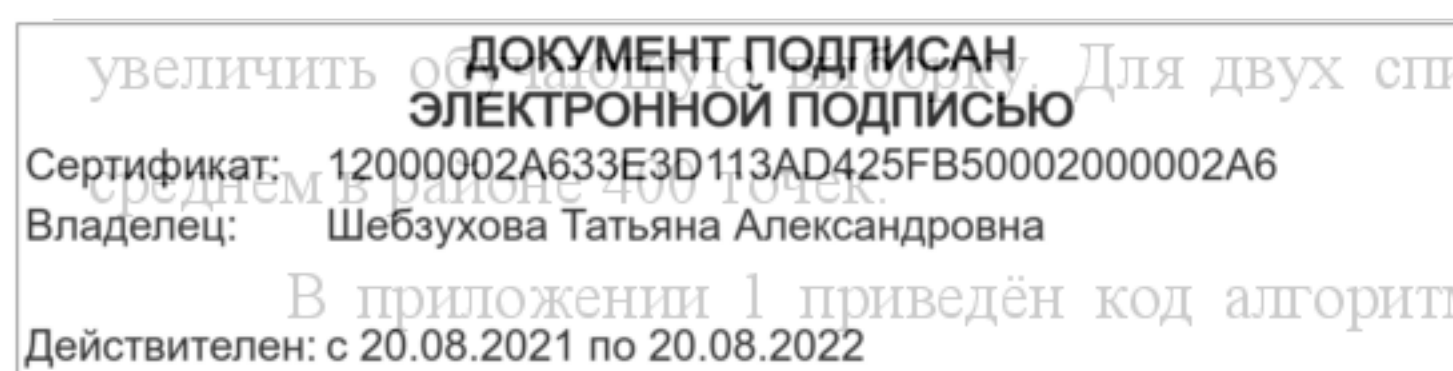
Структура сети	Ресурсы net.numWeightElements	Справилась с задачей	Обобщающая способность
2-50-1	201	да	30%
2-10-10-1	151	да	50%
2-5-5-5-1	136	да	65%

Обобщающую способность можно было бы оценить более точно, если провести не один эксперимент, а целую серию и усреднить результаты, подсчитав, допустим, доверительные интервалы. Что видно из этих результатов?

Во-первых, введение промежуточных слоёв позволяет уменьшить количество настраиваемых параметров.

Во-вторых, все три сети без проблем справились с постройки правильной поверхности отклика, т.е. без проблем раскрасили спирали.

В-третьих, то, что они хорошо строят поверхность отклика, не говорит ничего про их обобщающую способность! Первые две сети не справились с заданием вообще, т.к. 50% - это метод «тыка» при двух классах. Это говорит о том, что для сложных задач сложная система нелинейности (много слоёв) подходит лучше, чем простая, т.е. такие сети могут лучше справляться с задачами классификации. Более подробно про доказательства этого результата можно почитать в [7], где рассматриваются неглубокие архитектуры (shallow architecture) и глубокие архитектуры (deep architecture). Стоит также отметить, что для того, чтобы ИНС качественно предсказывали какой спирали может принадлежать точка, нужно



Для двух спиралей обучающая выборка должна быть в

ма обратного распространения ошибки для

построения разделяющей поверхности к данным, представленным на рисунке 5.26. Это две полуспирали.

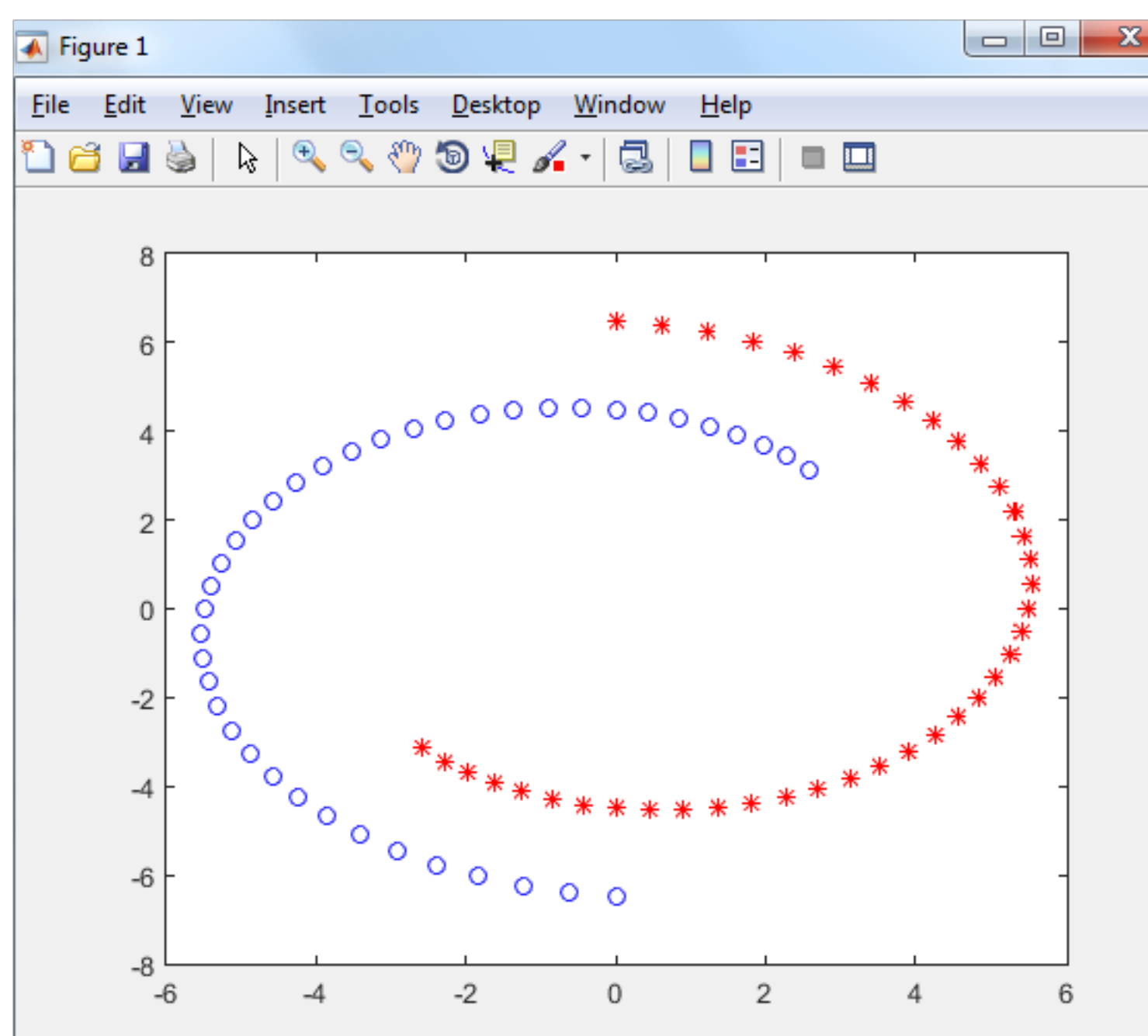


Рисунок 5.26 – Две полуспирали, которые необходимо разделить

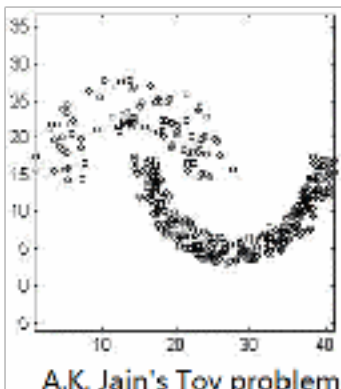
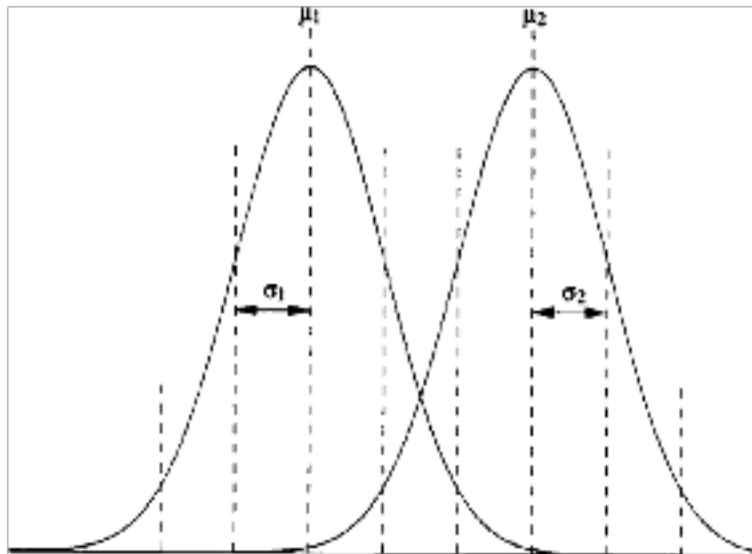
В данном лабораторном практикуме везде используются высокоуровневые функции Matlab для обучения и создания нейронных сетей. В приложении 1 показано, как самому запрограммировать обучение сети.

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

некоторых задачах её можно было бы сделать), приемлемая точность на тестовой выборке везде отмечена как 70%, что в целом не много. Достижение более большой точности распознавания оценивается в дополнительных баллах.

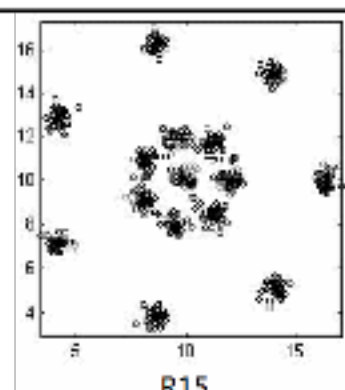
Таблица 5.5 – Варианты заданий

№	Условие задачи
1	Синтетические данные на классификацию. Постоянный адрес задачи: https://cs.joensuu.fi/sipu/datasets/.  Необходимо перевести выборку в формат Excel. Затем решить задачу классификации: создать нейронную сеть, которая принимая координаты точки, относит её к тому или иному классу. Разбитие на тестовую и обучающую выборку осуществляет сам студент, исходя из задачи. Можно ориентироваться на 80%-85% для обучающей выборки и 15%-20% для тестовой. В выборках обязательно учесть степень представленности классов. Если один класс представлен 20 паттернами, а другой 5, то очевидно, что в обучающей и тестовой выборке окажутся разное количество паттернов из этих классов. Постараться получить классификатор с качеством правильных ответов не ниже 70% для тестовой выборки. Правильным ответом считается ответ сети, при котором евклидово расстояние между идеальным требуемым ответом (учитель) и ответом сети не превышает по модулю 0.1.
2	Задача о двух многомерных перекрывающихся нормальных распределениях.  Даны два двадцатимерных нормальных распределения (7400 паттернов). Одно кодируется классом 0, другое – 1. Класс 1 имеет среднее $(a, a, a, \dots, a)$, класс 2 – $(-a, -a, -a, \dots, -a)$, где $a = 2 / 20$. Необходимо на тестовом множестве (можно использовать 300 паттернов) добиться правильной классификации в 70%. Ошибка примерно 25% - нормальна.
3	Дана синтетическая выборка, в которой паттерны принадлежат двум кластерам в форме банана. Каждый паттерн задаётся двумя точками и типом класса: -1 или 1. Требуется разбить выборку на два множества: обучающее и тестовое и добиться правильной классификации на тестовом множестве более 70%. В этой задаче полезно для студента вывести данные на канву и оценить степень репрезентативной представленности каждого класса.
4	Синтетические данные на классификацию. Постоянный адрес задачи: http://cs.joensuu.fi/sipu/datasets/.

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

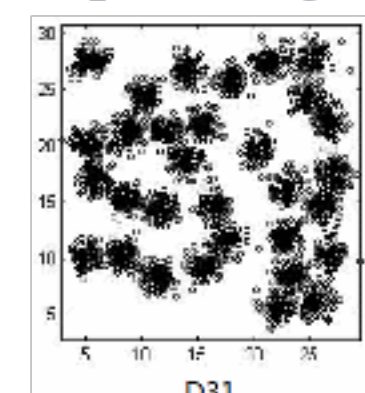


Комментарий к варианту 1.

5

Задача про фонемы. Даны два класса: один описывает назальный звук (класс 0), другой – оральный (класс 1). Класс 0 содержит 3818 паттернов, а класс 1 – 1586 паттернов. Каждый класс описывается 5 фонемами: Aa, Ao, Dcl, Iy, Sh. Фонемы изменяются в следующих вещественных диапазонах: Aa [-1.7, 4.107], Ao [-1.327, 4.378], Dcl [-1.823, 3.199], Iy [-1.581, 2.826], Sh [-1.284, 2.719]. В физический смысл этих фонем и диапазонов вникать не нужно. Необходимо произвести классификацию и добиться на тестовом множестве качества распознавания не ниже 70%.

6

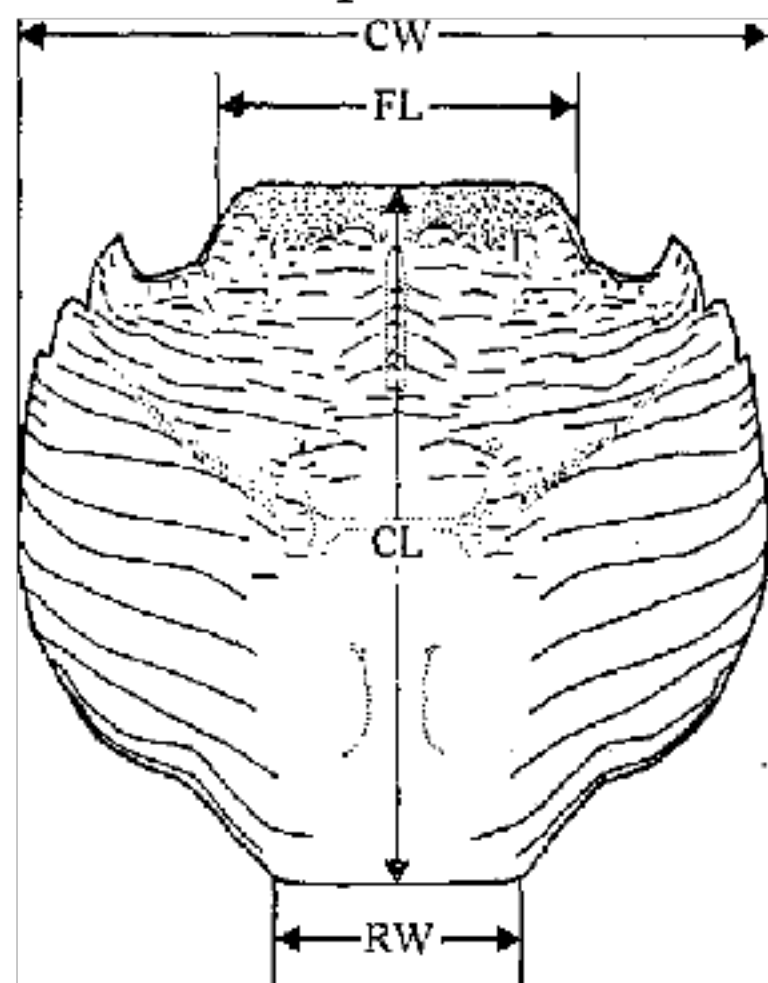


Комментарий к варианту 1.

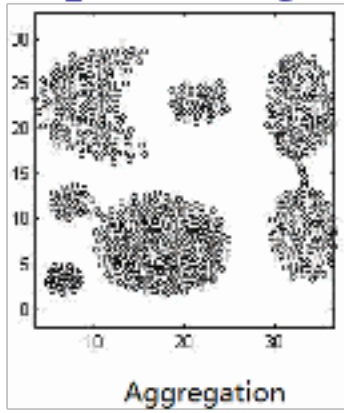
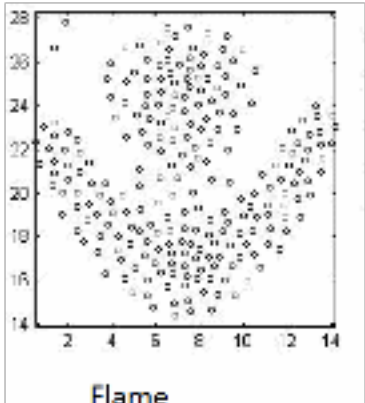
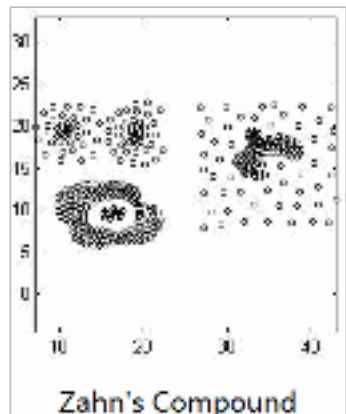
7

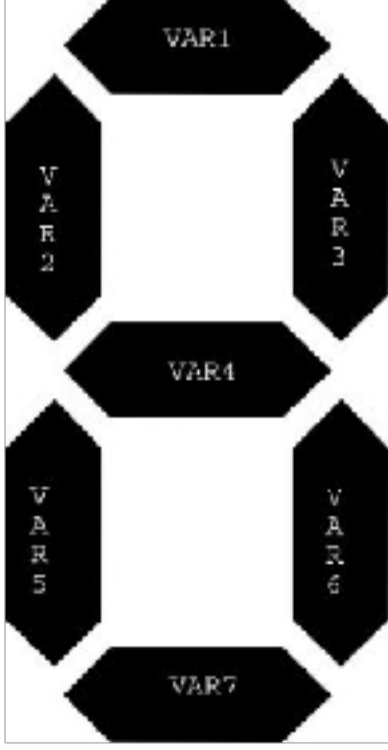
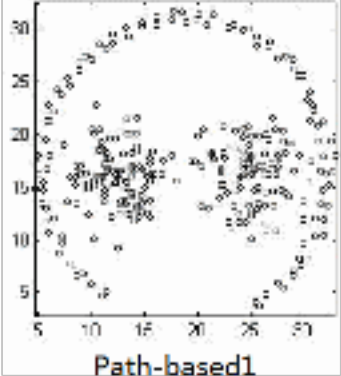
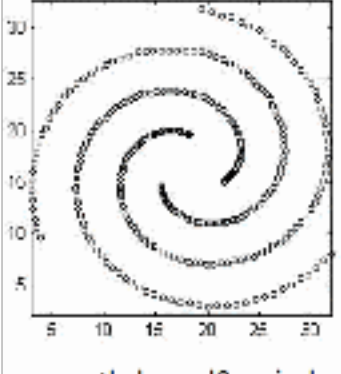
Задача о крабах genus Leptograpsus.

Имеется выборка по крабам двух видов. Каждый вид представлен 50 самцами и 50 самками, т.е. вся выборка 200 измерений. Первая колонка – пол. Вторая колонка – индекс от 1 до 50, далее 5 колонок – это измерения тела краба (как в задачах об ирисах Фишера) в мм: FL, RW, CL, CW, BD. Схема измерений представлена ниже. В [8] можно получить подробную информацию об этих животных, а также о смысле эксперимента.



Последняя колонка это вид краба. Выборка бьётся на тестовую и обучающую. Для тестовой выборки подойдёт по 5 экземпляров из каждого класса: 5 самцов типа 1, 5 самок типа 1, 5 самцов типа 2, 5 самок типа 2. Всё остальное – обучающая выборка, хотя студент может предложить своё разбиение. На вход сети подаётся 6 параметров (индексы, разумеется, подавать не надо, они нужны для разбиения на обучающую и тестовую выборку, а также для анализа ответов). Требуется создать и обучить нейронную сеть, которая правильно определяет тип краба на тестовой выборке с вероятностью не менее 70% (чем больше, тем лучше). Что такое правильный ответ написано в варианте №1, а также разобрано в задаче об ирисах Фишера.

8	Синтетические данные на классификацию. Постоянный адрес задачи: https://cs.joensuu.fi/sipu/datasets/.  Комментарий к варианту 1.
9	Синтетические данные на классификацию. Постоянный адрес задачи: https://cs.joensuu.fi/sipu/datasets/.  Комментарий к варианту 1.
10	Задача о классификации стекла. Имеется выборка из 214 записей. Состоит из 10 колонок (+ одна колонка индексов): RI – индекс преломления стекла, далее 8 колонок, которые обозначают содержание в процентах того или иного элемента (Na, Mg, Al, Si, K, Ca, Ba, Fe). Последняя колонка – тип стекла: 1 – window float glass, 2 – window non-float glass, 3 – vehicle glass, 4 – vehicle non-float glass (данный класс отсутствует в данной выборке), 5 – containers, 6 – tableware, 7 – vehicle headlamp glass. Из-за отсутствующего одного класса получается 6 типов стекла. Выборку необходимо разбить на две части: обучающую и тестовую. В данной выборке классы представлены различным количеством элементов, поэтому для тестовой каждый класс вносит не одинаковое количество паттернов: 1 – 10 паттернов, 2 – 10, 3 – 3, 5 – 3, 6 – 2, 7 – 4. Всё остальное идёт в обучающую выборку. Добиться классификации с точностью на тестовой выборке более 70%. Дополнительную информацию вместе со статьями, в которых использовалась эта выборка, можно получить по адресу: https://archive.ics.uci.edu/ml/datasets/Glass+Identification/. В Matlab можно набрать команду nprtool, нажать Next, далее Load Example Dataset и выбрать Types of Glass и почитать информацию об этой задаче. Однако, стоит учесть, что в Matlab интегрирована упрощенная задача, где на выходе имеем всего 2 класса.
11	Синтетические данные на классификацию. Постоянный адрес задачи: https://cs.joensuu.fi/sipu/datasets/.  Комментарий к варианту 1.
12	Задача о классификации.

	 Цифра на калькуляторе кодируется переменными VAR1..VAR7, расположенными так, как показано выше. Переменная может принимать значение 0 (ZERO), если часть табло, которая связана с этой переменной не работает, или 1 (ONE), если часть табло работает. В выборке всего 500 записей, каждая запись состоит из 7 значений переменных и класса цифры. Не все переменные правильно кодируют свой класс, т.к. табло неисправно. Необходимо составить классификатор, который на тестовом множестве покажет выше 70% правильных ответов. Можно использовать следующую структуру сети: 7-5-10, количество эпох 500, начальная скорость обучения 0.01. Разбитие на обучающую и тестовую выборки можно производить исходя из соотношений 85%-15%, степень равномерной представленности каждого класса должен сделать сам студент.
13	Синтетические данные на классификацию. Постоянный адрес задачи: https://cs.joensuu.fi/sipu/datasets/.  Комментарий к варианту 1.
14	Синтетические данные на классификацию. Постоянный адрес задачи: https://cs.joensuu.fi/sipu/datasets/.  Комментарий к варианту 1.

Содержание отчета и его форма

Отчёт по лабораторной работе должен содержать следующую информацию:

1. Название лабораторной работы и её номер.
2. ФИО студента и группу.
3. Формулировка индивидуального задания.

4. Документ, подписанный в диалоговых окнах по шагам для своего варианта по ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

количестве классов, количестве паттернов в каждом классе, о размерах тестовой и

обучающей выборки, о представленности каждого класса в тестовой и обучающей выборке, об используемых алгоритмах обучения нейронных сетей, а также результирующие графики и процент правильно распознанных паттернов на тестовой выборке. Необходимо также указать структуру сети и обосновать почему выбрана именно такая сеть.

5. Ответы на контрольные вопросы.

Вопросы для защиты работы

- 1) Какие алгоритмы обучения нейронных сетей вы знаете, чем они отличаются?
- 2) Для чего используются в сетях прямого распространения связи через слой?
- 3) В каких случаях для задачи классификации выгоднее использовать рбф-нейрон, чем сигмоидальный нейрон?
- 4) Какой ответ сети считать правильным?
- 5) Чем нужно руководствоваться, разбивая выборку на тестовую и обучающую?
- 6) Какие функции в Matlab можно использовать для создания сети и в чём их отличие?
- 7) Какая структура данных содержит информацию об процессе обучения сети и как её получить?

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

Лабораторная работа № 6

Пример создания и обучения нейронных сетей для задач классификации в среде Matlab. Часть 2

Цель и содержание работы: создать и обучить несколько нейронных сетей, решающих задачи классификации, рассмотреть предобработку данных.

Задачи:

- рассмотреть различные способы нормировки входных данных;
- научиться кодировать категориальные переменные;
- научиться работать с вещественными переменными разных диапазонов.

Теоретическое обоснование

Рассмотрим 2 задачи, уделив особое внимание предобработки данных. Под предобработкой будет пониматься работа с категориальными, ординальными (порядковыми) и обычными числовыми переменными, и нормализация данных.

Первая задача. Оценка влияния социальных факторов на выживание пассажиров потерпевшего аварию судна «Титаник».

В интернете есть различные варианты данной выборки. Одна из наиболее старых и больших, это выборка с Delve Project [1]. Выборка состоит всего из 2201 наблюдений. Каждое наблюдение состоит из трёх параметров и результата. Первая колонка – класс, которым плыл человек на «Титанике» (0 – член команды, 1 – первый класс, 2 – второй класс, 3 – третий класс), вторая колонка – возраст (1 – взрослый, 0 - ребёнок), третья колонка – пол (1 – мужчина, 0 - женщина), четвёртая колонка – результат, по сути, - класс (1 – выжил после крушения, 0 – нет). Фрагмент выборки приведён в таблице 6.1.

Таблица 6.1 – Фрагмент выборки для задачи о «Титанике»

№	Class	Age	Sex	Survived
1	2	1	0	1
2	1	0	1	1
3	1	1	1	0
4	0	1	1	0
5	0	1	0	1
6	0	1	1	1

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Подойдёт нам. Дело в том, что входной вектор
Действителен: с 20.08.2021 по 20.08.2022

Titanic_1.xlsx». Однако данная выборка не
в этом варианте выборки содержит только

категориальные и ординальные переменные и их слишком мало: Пол, Класс, Возраст (в виде категории). К чему это приводит? К тому, что данные становятся противоречивыми: слишком много одинаковых входных векторов будут иметь разные выходные классы (выжил, не выжил). Если бы мы решали данную задачу не с помощью нейронных сетей, где требуется получить один классификатор, который и принимает решение, а с помощью, допустим, random forest, то проблемы бы не было. Нам же требуется «разбавить» эти отобранные главные компоненты более второстепенной информацией. Поэтому за основу была взята выборка, содержащаяся в файле «DataForTitanic_2.xlsx», она была преобразована к финальной выборке «DataForTitanic.xlsx». Нужно учесть, что несмотря на то, что теперь данная задача стала адекватной для нейронной сети, однако серьёзных результатов ждать не приходится, т.к. на выживаемость больше всего влияют именно класс, пол и возраст, поэтому то, что мы «разбавили» входную информацию, ещё не означает, что мы сняли логические противоречия. В замаскированном виде они все равно присутствуют в выборке, т.е. данная задача неудобна для нейронной сети.

Теперь сеть имеет отличающиеся входные вектора там, где раньше они были похожи для двух разных классов, но самые влияющие на выживаемость компоненты этих векторов всё равно одинаковые.

Были оставлены такие столбцы как class (каким классом плыл), survived (выжил, не выжил), sex (пол), age (возраст в виде числа, а не в виде категории), sibsp (количество родственников второго порядка: муж, жена, братья), parch (количество родственников первого порядка: мать, отец, дети), fare (цена билета), embarked (порт посадки: 0 – Cherbourg, 1 – Queenstown, 2 – Southampton). Колонка Age содержала 263 пропуска, они были заполнены медианой по всему столбцу от этой выборки (28 лет). Колонка Fare содержала один пропуск, он также был заполнен медианой (14.45). Колонка Embarked содержала два пропуска, они были заполнены значениями «2». Именно на этой выборке будет обучаться нейронная сеть. Выборка содержит 1309 паттернов.

Вторая задача. Классификация грибов.

Имеется выборка Шлиммера в которой описаны 23 вида грибов из семейств Agaricus и Lepiota. Всего в выборке 8124 записей, каждый гриб описывается 22 параметрами (однако, один из параметров: stalk-root, может содержать пропущенные значения, поэтому он исключается из оригинальной выборки Шлиммера и остаётся 21 параметр). Необходимо классифицировать гриб как съедобный или не съедобный. Ниже приведены номера

столбцов и их значения (русские термины переведены приближённо).

Документ подписан ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
convex (выпуклая) = x, flat (плоская) = f, knobbed (шарообразная) = k, sunken (впадая) = s).
Действителен: с 20.08.2021 по 20.08.2022

2. Поверхность шляпки (cap-surface: fibrous (жилистая) = f, grooves (с выемками) = g, scaly (чешуйчатая) = y, smooth (гладкая) = s).

3. Цвет шляпки (cap-color: brown (коричневый) = n, buff (цвета буйволового кожи) = b, cinnamon (цвета корицы) = c, gray (серый) = g, green (зелёный) = r, pink (розовый) = p, purple (пурпурный) = u, red (красный) = e, white (белый) = w, yellow (жёлтый) = y).

4. Наличие вмятин (bruises?: bruises = t, no = f).

5. Запах (odor: almond (миндаль) = a, anise (анис) = l, creosote (креозот) = c, fishy (рыбный) = y, foul (отталкивающий) = f, musty (запах плесени) = m, none (нет запаха) = n, pungent (острый) = p, spicy (запах пряностей) = s).

6. Наличие гимениальной пластинки, т.е. пластинки у гриба на нижней стороне шляпки (gill-attachment: attached (прикреплена) = a, descending (нисходящая) = d, free (свободно размещена) = f, notched (пилообразна) = n).

7. Тип разбивки на гимениальной пластинке (gill-spacing: close (близкая) = c, crowded (тесная) = w, distant (на некоторой дистанции) = d).

8. Размер гимениальной пластинки (gill-size: broad (широкий) = b, narrow (узкий) = n).

9. Цвет гимениальной пластинки (gill-color: black (черная) = k, brown (коричневая) = n, buff (цвета буйволового кожи) = b, chocolate (шоколадная) = h, gray (коричневая) = g, green (зелёная) = r, orange (оранжевая) = o, pink (розовая) = p, purple (пурпурная) = u, red (красная) = e, white (белая) = w, yellow (жёлтая) = y).

10. Форма ножки гриба (stalk-shape: enlarging (расширяющаяся) = e, tapering (суживающаяся) = t).

11. Поверхность ножки гриба выше кольца (stalk-surface-above-ring: fibrous () = f, scaly (чешуйчатая) = y, silky (шелковистая) = k, smooth (гладкая) = s).

12. Поверхность ножки гриба ниже кольца (stalk-surface-below-ring: fibrous = f, scaly = y, silky = k, smooth = s).

13. Цвет ножки гриба выше кольца (stalk-color-above-ring: brown = n, buff = b, cinnamon (корицы) = c, gray = g, orange = o, pink = p, red = e, white = w, yellow = y).

14. Цвет ножки гриба ниже кольца (stalk-color-below-ring: brown = n, buff = b, cinnamon = c, gray = g, orange = o, pink = p, red = e, white = w, yellow = y).

15. Тип раскраски (veil-type: partial (частичный) = p, universal (всеобщий) = u).

16. Цвет раскраски (veil-color: brown = n, orange (оранжевый) = o, white = w, yellow = y).

17. Количество колец (ring-number: none (нет) = n, one (одно) = o, two (два) = t).

18. Типы (обволакивающее) = s, zone (зональное) = z).

19. Цвет спор (spore-print-color: black = k, brown = n, buff = b, chocolate = h, green = r, orange = o, purple = u, white = w, yellow = y).

20. Популяция (population: abundant (изобильная) = a, clustered (пучочная, растущая пучками) = c, numerous (равномерная) = n, scattered (редкая, рваная) = s, several (малая) = v, solitary (одионочная) = y).

21. Место распространения (habitat: grasses (тpава) = g, leaves (листва) = l, meadows (луга) = m, paths (тропы) = p, urban (город) = u, waste (кустарник) = w, woods (леса) = d).

Класс для съедобного гриба edible = e, класс для несъедобного гриба poisonous = p.

Данная выборка находится в файле «DataForMushrooms.xlsx». Скачать выборку можно с сайта DELVE Project. Более подробная информация по выборке приведена в [2].

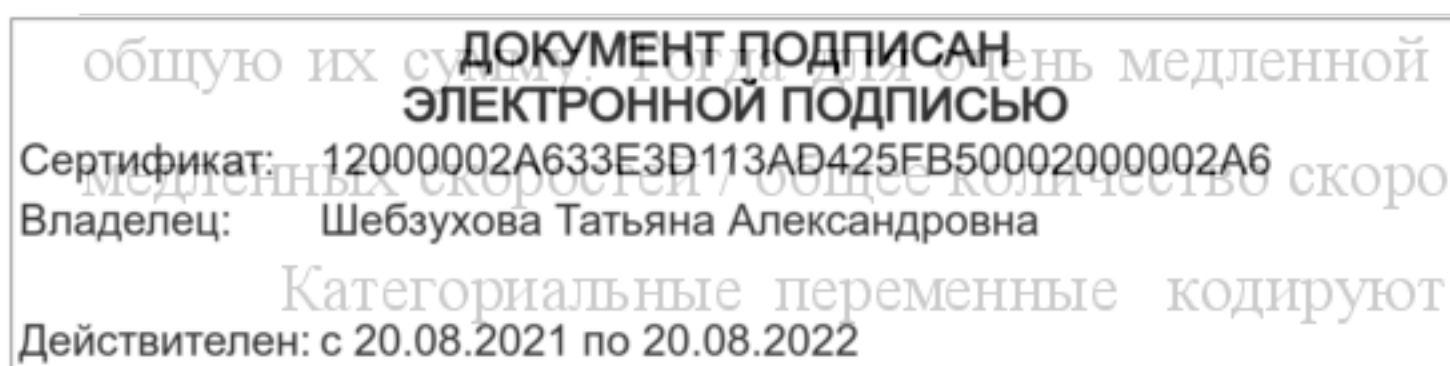
Для начала приведём краткую полезную информацию по предобработке данных: по кодированию входных переменных, а также по нормировке. Более сложные методы, такие как выбеливание, анализ главных компонент и т.д. не рассматриваются, т.к. эти методы больше относятся к большим объёмам данным (направление data mining) и для тренировочных задач избыточны.

Нейронные сети могут работать только с числами, поэтому вся входная информация должна быть представлена в числовом формате. Если параметр нечисловой, то обычно его можно отнести к одному из двух классов: категориальная переменная, ординальная переменная. Ординальные переменные отличаются от категориальных только тем, что их можно ранжировать, допустим, плохо – хорошо – отлично. Пример категориальных переменных красный, синий, зелёный и т.д.

Ординальные переменные более близки числовому ряду (т.к. есть порядок), поэтому их можно закодировать числами, допустим, очень медленно ($A = 0$) – медленно ($A = 0.25$) – средние ($A = 0.5$) – быстро ($A = 0.75$) – очень быстро ($A = 1$). Однако, нужно учитывать, что это не всегда разумное решение, т.к. тогда мы вводим отношения, которых изначально не было, допустим, при таких A получается, что «быстро» в 3 раза больше, чем «медленно». 0.75 – конкретное число, но ведь «быстро» может быть больше и в 4 раза, чем «медленно», это ведь абстрактные понятия, поэтому чтобы не вводить лишнюю информацию, лучше кодировать эти переменные также как и категориальные.

Примечание. В рассмотренном примере со скоростями значение A выбиралось произвольно человеком, более общий способ это подсчитать количества очень медленных, медленных, средних, быстрых, очень быстрых скоростей, и делить это количество на общую их сумму, тогда для очень медленной скорости получится ($A = \text{количество очень медленных скоростей} / \text{общее количество скоростей}$) и т.д.

Категориальные переменные кодируют через задание численных эквивалентов



классов (двоичное кодирование). Обычно используют два возможных варианта: код $n \rightarrow n$ и код $n \rightarrow m$ ($m < n$).

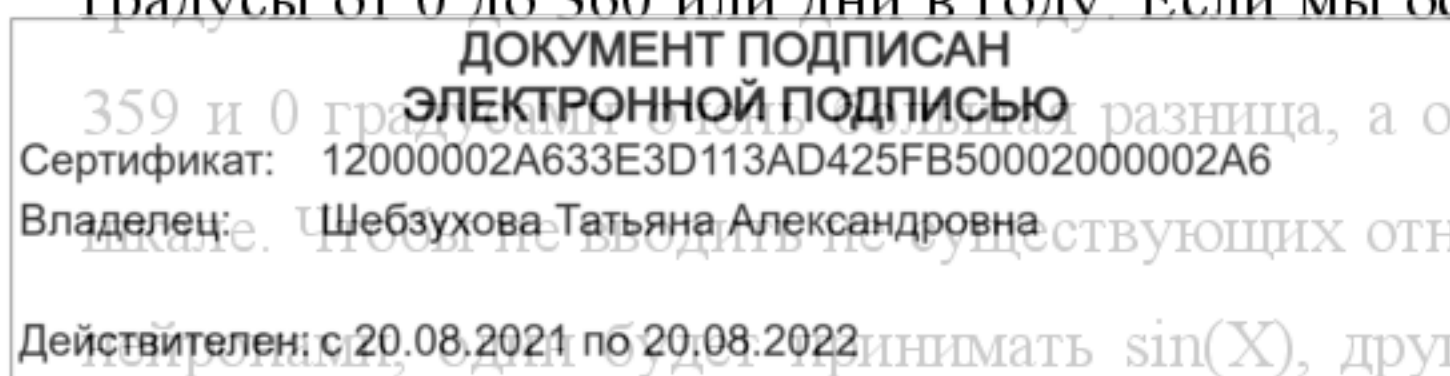
$N \rightarrow n$ означает, что если у нас 5 состояний, которые может принимать переменная, то используется 5 регистров-нейронов для её кодирования. Очень медленно (10000), медленно (01000), средне (00100), быстро (00010), очень быстро (00001).

$N \rightarrow m$ означает, что количество регистров-нейронов будет меньше, чем количество состояний. Обычно используются два варианта: «температурное» кодирование и классический двоичный код. Рассмотрим «температурное» кодирование. Очень медленно (0000), медленно (1000), средне (1100), быстро (1110), очень быстро (1111). Видно, что состояний 5, а регистров-нейронов уже 4. Второй тип кодирования – это классическое бинарное. У нас 5 состояний, тогда нам нужно m регистров-нейронов, чтобы $2^m \geq 5$, т.е. $m = 3$. Очень медленно (001), медленно (010), средне (100), быстро (101), очень быстро (110).

В чём особенность каждого типа и когда оно применяется? Кодирование $n \rightarrow n$ применяется тогда, когда количество значений каждого типа примерно равно. Т.е. пусть P_i – это количество значений переменной по скорости (A), $i = 1..5$, где P_1 – количество «очень медленных» скоростей, а P_5 – количество «очень быстрых» скоростей. Если $P_1 \approx P_2 \approx P_3 \approx P_4 \approx P_5$, то можно применять кодирование из $n \rightarrow n$, если это условие не выполняется хотя бы для одного P_i , то лучше применить из $n \rightarrow m$. Почему? Предположим, что P_1 значительно больше, чем $P_2..P_5$. Тогда, те связи, которые выходят из этого нейрона (первого из пяти) чаще активируются в процессе обучения, тогда как остальные – простаивают. Получается ситуация, что связи есть, а их обучения – нет.

Также стоит упомянуть ещё один момент, что изменения в процессе обучения весов первого слоя зависят от входных значений. И если во входном слое очень много нулей, то обучения просто не будет, процесс растянется. Поэтому во многих случаях лучше кодировать не 0 и 1, а ± 1 .

Теперь рассмотрим числовые входы, где значения только числа. В таком случае размер этих входов обычно не меняется. Если, к примеру, у нас есть 10 входных нейронов, из них 7 нейронов получают числовые входы, а не категории, то эти 7 нейронов так и останутся 7-ю нейронами, тогда, как оставшиеся 3, скорее всего, будут преобразованы в $q > 3$ нейронов по какому-нибудь рассмотренному выше способу кодирования. Однако, бывают и исключения, допустим, круговое кодирование, когда некий нейрон должен получать градусы от 0 до 360 или дни в году. Если мы оставим такой вход, то получится, что между 359 и 0 градусах есть большая разница, а они, тем не менее, стоят рядом на круговой шкале. Поэтому не вводить не существующих отношений можно этот нейрон заменить двумя нейронами, один будет принимать $\sin(X)$, другой $\cos(X)$. Но обычно числовые входы по



количеству не меняются. Главная проблема числовых входов – различные диапазоны значений. Один нейрон может принимать числа от 0 до 1, а другой от 10^3 до 10^4 . Можно такие входы и оставить, но тогда сеть будет вынуждена для больших входов подбирать меньшие значения весов, чтобы скомпенсировать разницу, т.е. появляется дополнительная сложность в обучении, которая может вообще затормозить весь процесс. Поэтому все числовые входы должны быть отнормированы к некоторому стандартному диапазону.

Существуют множество типов нормировок. Тут будут рассмотрены три типа нормировки: линейная, использующая среднеквадратическое отклонение, и сигмоидальная.

Для начала рассмотрим формулы этих нормировок. (6.1) – линейная нормировка, (6.2) – нормировка, использующая среднеквадратическое отклонение, (6.3) – сигмоидальная нормировка.

$$X_{new} = \left(\frac{X - X_{min}}{X_{max} - X_{min}} \right) * (b - a) + a, \quad (6.1)$$

где X_{new} – новое значение входа, X – старое значение входа, X_{min} – минимальное значение из всех значений входа, X_{max} – максимальное значение из всех значений входа, b – максимальное значение нового диапазона в который переводим старое значение, a – минимальное значение нового диапазона.

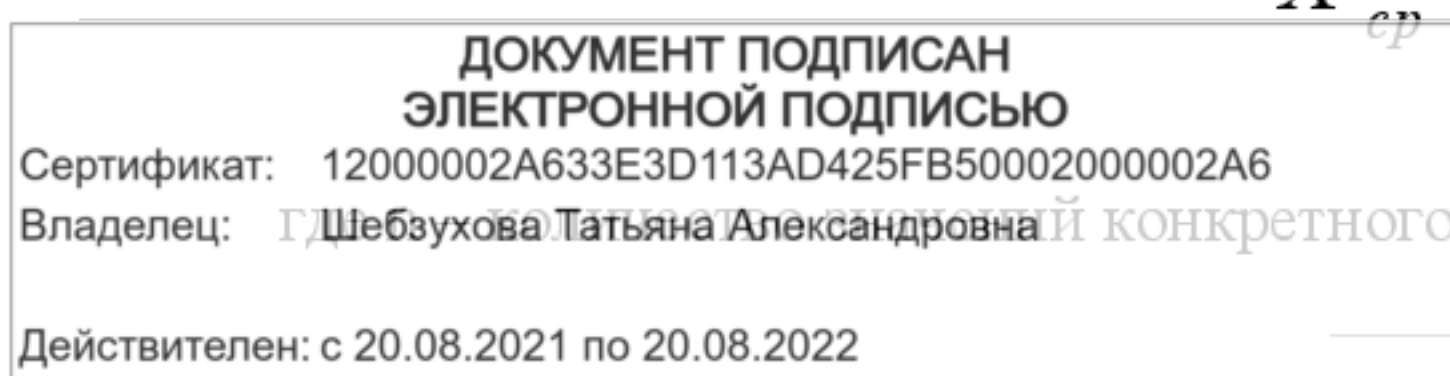
$$X_{new} = \frac{X - X_{cp}}{\sigma}, \quad (6.2)$$

где X_{cp} – среднеарифметическое значение (6.5) всех значений конкретного входа, σ – среднеквадратическое отклонение (6.6).

$$X_{new} = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}}, \quad (6.3)$$

$$\alpha = \frac{X - X_{cp}}{\sigma}, \quad (6.4)$$

$$X_{cp} = \frac{1}{n} * \sum_{i=1}^n x_i, \quad (6.5)$$



$$\sigma = \sqrt{\frac{1}{n} * \sum_{i=1}^n (x_i - X_{cp})^2}, \quad (6.6)$$

входа, x_i – конкретное значение входа.

Исходя уже из одних формул можно сделать вывод, что все нормировки используют

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

какие-нибудь статистики, заранее извлечённые из нужных диапазонов. Линейная использует минимальное и максимальное значения, другие – среднеквадратическое отклонение и т.д. А это означает, что не все нормировки применимы к режиму реального времени. Ведь подразумевается, что обучающая выборка не изменится в размерах, поэтому можно заранее в процессе предобработки извлечь нужные статистики и потом их всегда использовать. Если же обучающая выборка может пополняться в процессе обучения, а это очень вероятно для реальных задач, то нужно либо отказаться от сложных и качественных нормировок (за исключением линейной), либо использовать ускорительные техники для быстрого пересчёта значений и статистик, что не всегда возможно.

Рассмотрим код, демонстрирующий все три типа нормировки, а затем кратко сравним их.

```
% Двести значений
x=5:0.005:6;
% получаем некий числовой ряд с помощью тригонометрических
% функций
y1=exp(x)+25*sin(100*x)+43*randn(1,201)+36*cos(10*x);
% делаем в этом ряде два резких выброса, т.е. нетипичных значения
y1(50)=y1(50)+986;
y1(150)=y1(150)+1286;
% высчитываем статистики
% среднее арифметическое
mean1 = sum(y1)/length(y1);
% среднеквадратическое отклонение
std1 =sqrt((norm(y1)^2)/length(y1)-mean1);
% альфу
alpha = (y1 - mean1)/std1;
% минимум и максимум
min1=min(y1);
max1=max(y1);
% задаём новый диапазон для линейной нормировки
min2=-1; max2=1;
% линейно нормируем
y2=((y1-min1) / (max1-min1)) * (max2-min2)+min2;
% нормируем с помощью среднеквадратического отклонения
y3=(y1-mean1)/std1;
% сигмоидальная нормировка
y4 = (1-exp(-alpha))./(1+exp(-alpha));
% Выводим 4 графика, при желании их можно
% увеличить с помощью zoom+
figure(4)
subplot (221), plot(y1(1:200)), xlabel('Y1');
subplot (222), plot(y2(1:200)), xlabel('Y2. Linear normalization');
subplot (223), plot(y3(1:200)), xlabel('Y3. Zscore normalization');
subplot (224), plot(y4(1:200)), xlabel('Y4. Sigmoidal normalization');
```

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

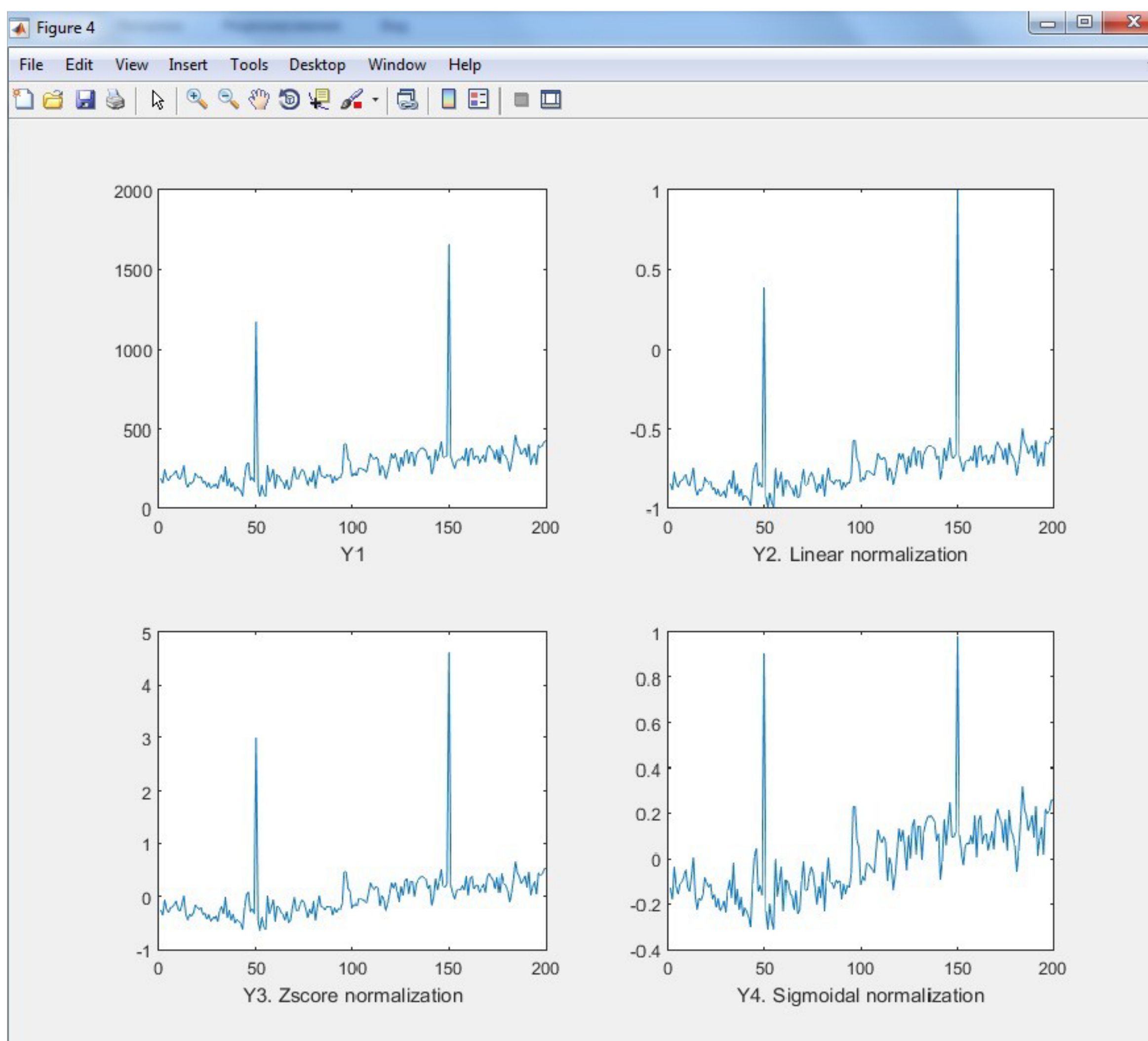


Рисунок 6.1 – Исходные значения (Y1) и три типа нормировки: Y2 – линейная, Y3 – с использованием среднеквадратического отклонения, Y4 - сигмоидальная

Исходя из рисунка 6.1 можно сделать следующие выводы: линейная нормировка совершенно не справляется с размазыванием значений между положительными и отрицательными (а это очень важно, чтобы отрицательных и положительных значений было примерно поровну в отнормированных данных. Это ускоряет процесс обучения), зато мы сами можем задать интересующий нас диапазон; нормировка с использованием среднеквадратического отклонения справляется с размазыванием лучше, однако нет гарантии интервала от -1 до +1; наконец сигмоидальная нормировка лучше остальных размазывает значения между положительными и отрицательными величинами, а также неплохо борется с выбросами данных, помещая их в единичный диапазон.

Конечный выбор нормировки остаётся всегда за разработчиком.

Рассмотрим первую задачу о «Титанике».

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022



Рисунок 6.2 – Крушение «Титаника»

Прежде чем решать задачу необходимо провести предварительный анализ входных данных путём ответов на ряд основных вопросов. Тогда понимание задачи значительно вырастет. При практическом решении задачи ряд пунктов предварительного анализа могут быть пересмотрены.

Вопрос 1. Если задача классификации, то сколько есть классов?

Ответ очевиден, в данной задаче 2 класса: выжил–не выжил.

Вопрос 2. Сколькими паттернами представлен каждый класс (степень равномерной представленности классов)?

Для ответа на этот вопрос нужно провести небольшой анализ. Удобно его провести либо силами Excel (фильтры, функции), либо написать небольшой код в Matlab. В данной лабораторной работе будет использован Excel. В файле «DataForTitanic.xlsx» можно в свободной ячейки написать «=СЧЁТЕСЛИ(H2:H1310; 1)» или «=СЧЁТЕСЛИ(H2:H1310; 0)», что вернёт количество классов выжил и не выжил, соответственно. Получится первый класс (1) представлен 500 паттернами, второй класс (0) представлен 809 паттернами, т.е. классы представлены неравномерно (погибших больше, чем выживших).

Вопрос 3. Сколькими элементами описывается паттерн каждого класса?

В данной задаче каждый класс до предобработки описывается вектором из семи элементов.

Вопрос 4. Какой тип каждой переменной-элемента (категориальная, ординальная, числовая) и её возможный числовой диапазон?

1-ая переменная (Class) – ординальная переменная: 1 – первый класс, 2 – второй класс, 3 – третий класс.

2-ая переменная (Sex) – категориальная переменная: 0 – женщина, 1 – мужчина.

3-ая переменная (Age) – обычная числовая из диапазона [0..80].

4-ая переменная (SibSp) – обычная числовая из диапазона [0..8].

5-ая переменная (Parch) – обычная числовая из диапазона [0..9].

6-ая переменная (Fare) – обычная числовая из диапазона [0..512.33].

7-ая переменная (Embarked) – категориальная: 0 – Cherbourg, 1 – Queenstown, 2 – Southampton.

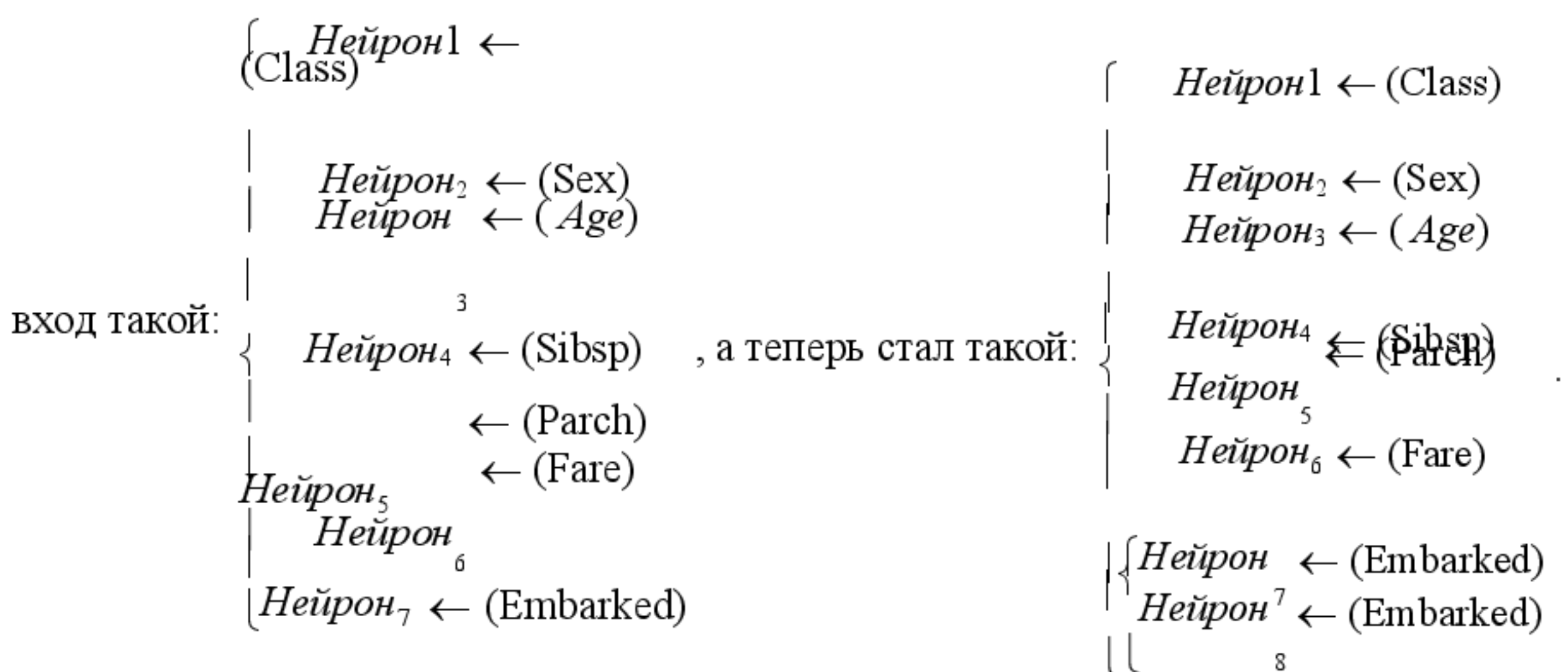
Ответ: 0 – не выжил, 1 – выжил.

Вопрос 5. Требуется ли изменять размерность входного вектора?

Есть категориальные и ординальные переменные, поэтому размерность придётся менять.

Переменную Class мы можем как перекодировать, так и нет, т.к. она ординальная и если мы оставим значения 1, 2, 3, то мы не введём не существующих отношений. Переменную Sex тоже менять не будем.

Перекодируем категориальную переменную Embarked. Для того, чтобы узнать, как это сделать, необходимо сначала проанализировать степень равномерной представленности значений этого компонента входного вектора. Анализ осуществлялся с помощью Excel («=СЧЁТЕСЛИ(B2:B2202; 0)», «=СЧЁТЕСЛИ(G2:G1310; 0)», «=СЧЁТЕСЛИ(G2:G1310; 1)», «=СЧЁТЕСЛИ(G2:G1310; 2)»). Получились следующие значения: 0 – 272 штук, 1 – 123 штук, 2 – 914 штук. Пусть P_i – количество соответствующих значений для данного значения-переменной входного вектора, где $i = 0..2$, тогда получается, что $P_2 \gg P_0 > P_1$, а, следовательно, нужно использовать классическое бинарное кодирование. $2^m \geq 3$ откуда $m = 2$. Т.е. вместо седьмого входного нейрона необходимо сделать два входных нейрона. Был



Ответ на пятый вопрос будет не полным, если не определиться с кодировкой входных значений. Если мы выберем кодировку 0/1, тогда Embarked закодируется, допустим, так 0 – (1, 1), 1 – (1, 0), 2 – (0, 1) (Учитывая расположение нейронов, далее везде в лабораторной работе предполагается, что 0 – (1, 1) это на самом деле $0 - (1, 1)^T$). Это приемлемо, но в целом не стоит забывать, что обучение первого слоя нейронной сети зависит от входных значений, и если тот или иной нейрон будет иметь слишком много нулей в качестве своих входных значений, то обучение его связей с первым скрытым слоем затормозится. Процесс обучения ускорится при использовании кодировки ± 1 .

Приходим к следующему результату: Embarked ($0 \rightarrow (-1, 1)$, $1 \rightarrow (1, -1)$, $2 \rightarrow (1, 1)$).

Количество портов посадки с индексом 0 и 1 меньше отличается друг от друга, чем с индексом 2, поэтому им назначаем симметричные коды.

Вопрос 6. Как кодировать класс, который нужно определить?

Мы уже определились, что у нас два класса, которые нужно классифицировать. Это простейший случай. Для двух классов достаточно одного нейрона в выходном слое с сигмоидальной или линейной функцией активации. Соответственно, для одного класса нейрон должен активироваться в зоне «плюс», для другого класса – в зоне «минус». Однако, более правильно сделать два нейрона. В задачах классификации обычно количество выходных нейронов соответствует количеству классов, которые нужно распознать. Такой подход упрощает процесс обучения, т.к. веса каждого такого нейрона решают свою локальную задачу классификации (отделения своего класса от всех остальных), а не пытаются решить две прямо противоположные задачи для разных классов (активировать нейрон в симметричных зонах). Исходя из этого, имеет смысл перекодировать класс Survived: $0 \rightarrow (1, -1)$, $1 \rightarrow (-1, 1)$.

Вопрос 7. К какому диапазону приводить остальные значения?

Мы перекодировали вход Embarked и выходной класс. Есть ещё одна категориальная переменная Sex, её также необходимо перекодировать в ± 1 : $0 \rightarrow (-1)$, $1 \rightarrow (1)$. Теперь остались обычные числовые значения и один ординарный вход. Их всех необходимо как можно более равномерно «размазать» между положительными и отрицательными значениями. Можно либо писать свой код для нормировки того или иного входа нейронной сети, как в листинге выше, а можно воспользоваться стандартными средствами Matlab, прежде всего, **mapminmax()** и **mapstd()**. **Mapminmax()** реализует нормировку, которая соответствует формуле (6.1), **mapstd()** – формуле (6.2). Можно также воспользоваться **fixunknowns()** для работы с пропущенными значениями. Более подробное применение этих функций приводится в листинге ниже.

Вопрос 8. Какая пропорция берётся для обучающей и тестовой выборки?

В тех задачах, где выборка заранее разбита этот вопрос не имеет смысла, если же дана одна искомая неразбитая выборка, то пользователь сам должен решать вопрос о разбитии. В данной задаче мы возьмём 80% (1047 паттернов) для обучающей выборки и 20% (262) для тестовой. Также популярны разбития 70%/30%. В целом, чем сложнее задача, тем больше паттернов должно идти в обучающую выборку и тем грамотнее они должны отбираться.

Вопрос 9. Как разделить обучающую и тестовую выборки?

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
каждый класс был равномерно представлен,
Действителен: с 20.08.2021 по 20.08.2022

Этот вопрос тесно связан с ответом на второй вопрос. Если бы в нашей выборке было достаточно было бы просто случайным

образом отобрать нужное количество паттернов для теста и для обучения. Но тут ситуация такая, что выжило 500, а погибло 809 человек, и если мы при таком раскладе начнём отбирать случайно, то может получиться, что, допустим, в обучающей выборке будет слишком мало паттернов класса «Выжил», что неправильно. Тогда перед нами стоит задача отобрать в тестовую и обучающую выборки количество паттернов каждого класса, учитывая общую пропорцию классов «Выжил» и «Погиб». Найдём сначала, сколько процентов занимает класс «Выжил» от «Погиб».

$$\left\{ \begin{array}{l} 809 = 100\% \\ 500 = x\% \end{array} \right. \Rightarrow x = \frac{500 * 100}{809} = 62\%, \text{ теперь зная это, найдём количество паттернов}$$

классов «Выжил» и «Погиб» для обучающей выборки:

$$\left\{ \begin{array}{l} 1047 = t_1 + t_2 \\ 0.62 * t_1 = t_2 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} 1047 = t_1 + 0.62 * t_1 \\ 0.62 * t_1 = t_2 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} t_1 = 646 \\ t_2 = 401 \end{array} \right., \text{ где } t_1 - \text{ количество паттернов класса}$$

«Погиб» для обучающей выборки, t_2 – количество паттернов класса «Выжил» для обучающей выборки, учитывая общее соотношение этих классов в первоначальной выборке.

Решим аналогичную систему для тестовой выборки:

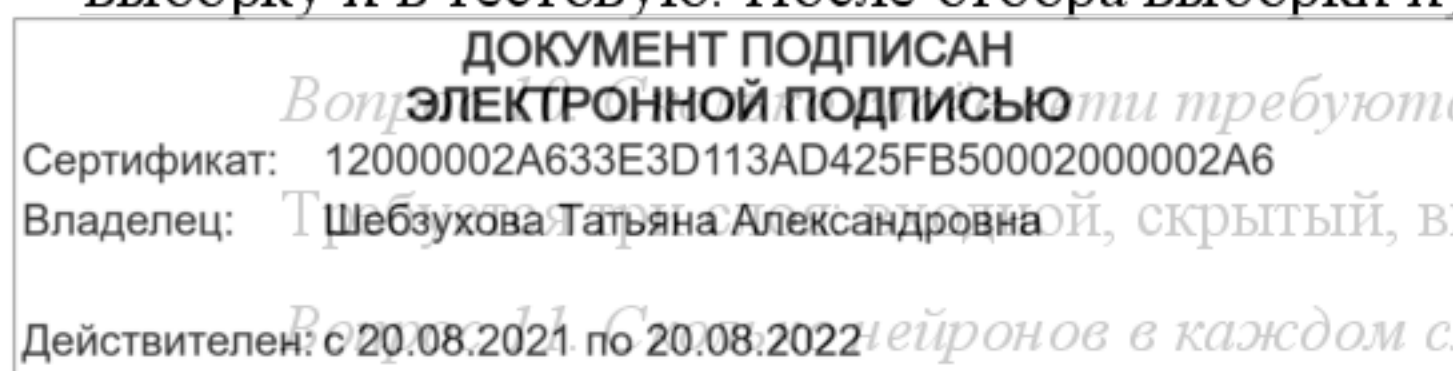
$$\left\{ \begin{array}{l} 262 = z_1 + z_2 \\ 0.62 * z_1 = z_2 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} 262 = 1.62 * z_1 \\ z_1 = 162 \\ z_2 = 100 \end{array} \right., \text{ где } z_1 \text{ и } z_2 - \text{ количество паттернов класса}$$

«Погиб» и «Выжил» для тестовой выборке соответственно. Однако, не нужно забывать, что решая эти две системы мы округляли проценты до 0.62, поэтому возможны небольшие неточности. Проверим результат. Общее количество выживших людей $z_2 + t_2 = 100 + 401 = 501$, общее количество погибших $z_1 + t_1 = 162 + 646 = 808$. Ответ же на второй вопрос при анализе выборки показал, что количество выживших людей составило 500, а погибших 809 людей, т.е. есть небольшая неточность, хотя по суммам всё сходится ($808 + 501 = 809 + 500$). Получается отличие в два человека. Вручную подкорректировав результат, отняв от одного уравнения и прибавив одного человека к другому уравнению в рамках одной из двух

систем, окончательно получим

$$\left\{ \begin{array}{l} t_1 = 647 \\ t_2 = 400 \end{array} \right. \wedge \left\{ \begin{array}{l} z_1 = 162 \\ z_2 = 100 \end{array} \right.$$

Теперь мы точно знаем, сколько паттернов и какого типа отбирать в обучающую выборку и в тестовую. После отбора выборки нужно перемешать.



Вопрос: *Вопросы требуются для решения данной задачи?*

Вопрос: *Вопросы требуются для решения данной задачи?*

Вопрос: *Вопросы требуются для решения данной задачи?*

Как уже неоднократно отмечалось на этапе предварительного анализа точно

ответить на этот вопрос невозможно, нужна практика.

Для входного слоя 8 нейрона (после расширения), для выходного – 2 нейрона. Для скрытого $q = \sqrt{8} = 4$ нейрона (Лабораторная работа №5). Как было замечено выше в этой задаче главная проблема – противоречие в данных, поэтому повысить качество через два скрытых слоя или серьёзное увеличение нейронов в скрытом слое не удастся всё равно.

Вопрос 12. Какие связи между слоями?

Сеть полносвязанная: каждый нейрон слоя связан со всеми нейронами последующего слоя. В большинстве задач, однако, полносвязанность почти всегда избыточное условие, которое накладывается в силу того, что мало что известно о внутренней структуре «черного ящика» (ИНС). Поэтому почти всегда удаётся улучшить результат, вводя дополнительную информацию в структуру сети, как это обычно происходит в сверточных сетях.

Вопрос 13. Какие функции активации?

В скрытом слое нейроны с гиперболическим тангенсом, значения которого изменяются от -1 до +1 (т.к. вход закодирован в ± 1), нейроны выходного слоя могут быть также такими же или линейными. Если выбирается гиперболический тангенс, то можно ставить функции активации на выходном слое вида $A \cdot \tanh(B \cdot x)$, где A и B – специальные константы. Это делается потому, что $\tanh(x)$ имеет ± 1 в зонах насыщения, этих значений тяжело достичь, а при константе A функция вытягивается вдоль оси OY и ± 1 оказываются в линейной части.

Вопрос 14. Какая функция ошибки?

Среднеквадратическая ошибка (MSE).

Приведённый анализ должен помочь пользователю лучше понять задачу. Ряд других вопросов лучше решать на практике: какой алгоритм использовать при обучении, какие результаты и как отображать, и т.д.

% Загружаем выборку

```
initSet = xlsread('C:\Users\Computer Grand\Desktop\МАТЕРИАЛЫ\К лабе  
6\DataForTitanic.xls', 1, 'A:H');
```

% Транспонируем выборку

```
initSet = initSet';
```

```
% Находим индексы тех элементов, где в последней строчке стоит 0,
```

```
% т.е., где человек утонул
```

```
ind_zero = find(initSet(8, :)==0);
```

```
% находим индексы 1
```

```
ind_one = find(initSet(8, :)==1);
```

```
% Формируем вектор с нужным балансом 0 и 1
```

```
ind_for_training = [ind_zero(1:647) ind_one(1:400)];
```

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

```
% Аналогично для тестового вектора
```

```
ind_for_test = [ind_zero(648:809) ind_one(401:500)];
```

Действителен с 20.08.2021 по 20.08.2022

```

% Но теперь они не перемешаны, индексы нужно перемешать
IndForTest = ind_for_test(randperm(length(ind_for_test)));
IndForTraining = ind_for_training(randperm(length(ind_for_training)));
% Очищаем оперативную память от громоздких матричных данных,
% которые уже не нужны
clear ind_for_test, clear ind_for_training, clear ind_one, clear ind_zero;
% Создаём заготовку для всей выборке, куда будут помещаться уже
% перемешанные данные
initSetPerm = [repmat([0 0 0 0 0 0 0 0]', 1, length(initSet))];
% Заполняем нашу заготовку, теперь она и есть наша выборка
initSetPerm(:, 1:1047) = initSet(:, IndForTraining(:));
initSetPerm(:, 1048:1309) = initSet(:, IndForTest(:));
clear IndForTest, clear IndForTraining, clear initSet;
% Создаём ещё одну заготовку, но уже для выборки с изменённым размером
% входов теперь станет 8, а не 7 и выход с одного нейрона расширится до 2-х
ExtendedInitSet = [repmat([0 0 0 0 0 0 0 0 0]', 1, length(initSetPerm))];
% осуществляем все необходимые замены
proba = find(initSetPerm(8, :)==0);
for i=1:length(proba)
    ExtendedInitSet(9:10, proba(i)) = [1 -1]';
end
proba = find(initSetPerm(8, :)==1);
for i=1:length(proba)
    ExtendedInitSet(9:10, proba(i)) = [-1 1]';
end
proba = find(initSetPerm(7, :)==0);
for i=1:length(proba)
    ExtendedInitSet(7:8, proba(i)) = [-1 1]';
end
proba = find(initSetPerm(7, :)==1);
for i=1:length(proba)
    ExtendedInitSet(7:8, proba(i)) = [1 -1]';
end
proba = find(initSetPerm(7, :)==2);
for i=1:length(proba)
    ExtendedInitSet(7:8, proba(i)) = [1 1]';
end
proba = find(initSetPerm(2, :)==0);
for i=1:length(proba)
    ExtendedInitSet(2, proba(i)) = -1;
end
proba = find(initSetPerm(2, :)==1);
for i=1:length(proba)
    ExtendedInitSet(2, proba(i)) = 1;
end
% копируем данные, которые не требуют на этом этапе нормировки,
% они будут отнормированы на этапе обучения сети
% ExtendedInitSet(1, :) = initSetPerm(1, :);

```



```

% однако их можно и в принудительном порядке отнормировать, как
% показано ниже
ExtendedInitSet(1, :) = mapminmax(initSetPerm(1, :));
ExtendedInitSet(3, :) = mapstd(initSetPerm(3, :));
ExtendedInitSet(4, :) = mapstd(initSetPerm(4, :));
ExtendedInitSet(5, :) = mapstd(initSetPerm(5, :));
ExtendedInitSet(6, :) = mapstd(initSetPerm(6, :));

clear i, clear initSetPerm, clear proba;
% Бъём выборку на входы и метки
input_data = ExtendedInitSet(1:8, :);
output_data = ExtendedInitSet(9:10, :);

% Создаём сеть для решения задачи, использовали 8 нейронов в скрытом слое,
% можно попробовать сеть с двумя скрытыми слоями, но результат не улучшится
net = newff(input_data, output_data, [8], {'tansig', 'tansig'}, 'trainlm', 'learngd', 'mse', {}, {},
'divideind');
% Отмечаем диапазон для обучения и валидационной выборки (теста)
net.divideParam.trainInd = 1:1047;
net.divideParam.valInd = 1048:1309;
net.trainParam.max_fail = 2;
% Устанавливаем тип нормировки
% net.inputs{1}.processFcns = {'mapstd'};
net = init(net);
[net, tr] = train(net, input_data, output_data);
% выделяем тест для проверки
testSet = input_data(1:8, 1048:1309);
testT = output_data(1:2, 1048:1309);
Y = sim(net, testSet);
col = 0;
for i = 1:length(testSet)
% сравниваем ответ сети с идеальным
    if norm(testT(i)-Y(i)) < 0.3
        col = col + 1;
    end
end
% выводим процент правильных ответов на тесте
percent = 100 * col / length(testT)

```

В результате работы этого кода, можно добиться точности распознавания до 65%, что довольно мало. Какой вывод следует из решения данной задачи? Если нейронная сеть получает противоречивые данные или данные, где противоречие находится не в явном виде, то никакая архитектура нейронной сети не сможет качественно решить задачу. Что тогда можно сделать? Можно, во-первых, постараться изменить обучающую выборку, что и было сделано. Во-вторых, нужно на обучающее множество отвести как можно больше

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шибзухова Татьяна Александровна
 Действителен с 20.08.2021 по 20.08.2022

паттернов. В-четвёртых, можно попробовать сеть с несколькими скрытыми слоями. В-пятых, можно попробовать использовать другие функции активации. Так в коде правильный ответ считается тот, чей вектор отличается от идеального на 0.3 или меньше. Но последняя техника возможна не

всегда. Альтернатива всем этим действиям – выделить нейронной сети много настраиваемых параметров и превратить её в память, но тогда потеряются все свойства обобщения.

Рассмотрим вторую задачу о грибах.



Рисунок 6.3 – Гриб из семейства Agaricus

Также проведём предварительный анализ данной задачи.

Вопрос 1. Если задача классификации, то сколько есть классов?

Два класса: съедобен (edible), ядовит (poisons).

Вопрос 2. Сколькими паттернами представлен каждый класс (степень равномерной представленности классов)?

Съедобен – 4208 (51.8%), ядовит – 3916 (48.2%). Фактически это равномерная представленность классов.

Вопрос 3. Сколькими элементами описывается паттерн каждого класса?

Каждый паттерн описывается вектором из 21 элементов-параметров.

Вопрос 4. Какой тип каждой переменной-элемента (категориальная, ординальная, числовая) и её возможный числовой диапазон?

Все переменные категориальные.

Вопрос 5. Требуется ли изменять размерность входного вектора?

Прежде, чем отвечать на этот вопрос в учебных целях проведём анализ входных переменных. На основе этого анализа сделаем ряд предварительных выводов. Результаты анализа представлены в таблице 6.2.

Таблица 6.2 – Анализ значений входных переменных

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ		Входная переменная (количество)	Вывод
№	Входная переменная	Входная переменная (количество)	Вывод
Сертификат: 12000002A633E3D113AD425FB50002000002A6	Владелец: Шебзухова Татьяна Александровна	Входная переменная (количество)	Неравномерное распределение
Действителен: с 20.08.2021 по 20.08.2022		Входная переменная (количество)	Вывод

			значений, кодирование бинарное
2	Cap-surface	f(2320), g(4), y(3244), s(2556)	Неравномерное распределение значений, кодирование бинарное
3	Cap-colour	n(2284), b(168), c(44), g(1840), r(16), p(144), u(16), e(1500), w(1040), y(1072)	Неравномерное распределение значений, кодирование бинарное
4	Bruises?	t(3376), f(4748)	Равномерное распределение, но всего два значения, поэтому остаётся один нейрон
5	Odor	a(400), l(400), c(192), y(576), f(2160), m(36), n(3528), p(256), s(576)	Неравномерное распределение значений, кодирование бинарное
6	Gill-attachment	a(210), d(0), f(7914), n(0)	Неравномерное распределение значений, кодирование бинарное, два значения в реальной выборке отсутствуют, поэтому значений не 4, а 2
7	Gill-spacing	c(6812), w(1312), d(0)	Неравномерное распределение значений, кодирование бинарное, значений не 3, а 2
8	Gill-size	b(5612), n(2512)	Неравномерное распределение значений, всего два значения, поэтому остаётся один нейрон
9	Gill-colour	k(408), n(1048), b(1728), h(732), g(752), r(24), o(64), p(1492), u(492), e(96), w(1202), v(86)	Неравномерное распределение значений, кодирование бинарное
			Равномерное распределение

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Щербатых Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

			значений, но их всего два, поэтому остаётся один нейрон
11	Stalk-surface-above-ring	f(552), y(24), k(2372), s(5176)	Неравномерное распределение значений, кодирование бинарное
12	Stalk-surface-below-ring	f(600), y(284), k(2304), s(4936)	Неравномерное распределение значений, кодирование бинарное
13	Stalk-colour-above-ring	n(448), b(432), c(36), g(576), o(192), p(1872), e(96), w(4464), y(8)	Неравномерное распределение значений, кодирование бинарное
14	Stalk-colour-below-ring	n(512), b(432), c(36), g(576), o(192), p(1872), e(96), w(4384), y(24)	Неравномерное распределение значений, кодирование бинарное
15	Veil-type	p(8124), u(0)	Данный вход по факту есть константа, достаточно одного нейрона
16	Veil-colour	n(96), o(96), w(7924), y(8)	Неравномерное распределение значений, кодирование бинарное
17	Ring-number	n(36), o(7488), t(600)	Неравномерное распределение значений, кодирование бинарное
18	Ring-type	c(0), e(2776), f(48), l(1296), n(36), p(3968), s(0), z(0)	Неравномерное распределение значений, кодирование бинарное, значений не 8, а 5
19	Spore-print-colour	k(1872), n(1968), b(48), h(1632), r(72), o(48), u(48), w(2388), y(48)	Неравномерное распределение значений, кодирование бинарное
ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 12000002A633E3D113AD425FB50002000002A6 Владелец: Шебзухова Татьяна Александровна Действителен: с 20.08.2021 по 20.08.2022			Неравномерное распределение значений, кодирование бинарное
		a(364), c(340), n(400), s(1248), v(4040), y(1712)	Неравномерное распределение значений, кодирование бинарное

			значений, кодирование бинарное
21	Habitat	g(2148), l(832), m(292), p(1144), u(368), w(192), d(3148)	Неравномерное распределение значений, кодирование бинарное

Как видно из таблицы анализ помог выявить, что часть значений у некоторых переменных вообще не встречается, хотя они описаны в условии задачи, а также почти везде был выбран самый компактный тип кодирования категориальных переменных: классическое бинарное кодирование.

В отличие от первой задачи тут размерность входа достаточно большая (21 переменная), поэтому не нужно опасаться, что входной вектор будет близок к нулевому во многих случаях, он будет просто разряженным. Тут уместно оставить кодировку 0/1, но можно использовать и ± 1 . Выберем ± 1 из-за скорости обучения.

Прежде, чем кодировать входные значения необходимо учесть, что может быть такая ситуация, что, допустим, некоторая переменная принимает 5 значений, тогда для того, чтобы закодировать все значения бинарной кодировкой нужно 3 нейрона, но три нейрона могут максимум закодировать 8 значений (2^3). Получается, что у нас есть 3 лишних кода, тогда встаёт вопрос о том, какие значения бинарной кодировки использовать для кодирования? Лучше использовать такие значения из возможных 8, чтобы сохранялась симметрия между 0 и 1 или ± 1 . Т.е., если мы закодируем просто 5 значений кодом без учёта симметрии, то получим что-то вроде следующего: $0 \rightarrow (000)$, $1 \rightarrow (001)$, $2 \rightarrow (010)$, $3 \rightarrow (011)$, $4 \rightarrow (100)$. Видно, что нулей в таком коде больше единиц, а с учётом симметрии будет так: $0 \rightarrow (000)$, $1 \rightarrow (111)$, $2 \rightarrow (001)$, $3 \rightarrow (110)$, $4 \rightarrow (100)$. Учтём этот момент при кодировке. Результат кодировки представлен в таблице 6.3.

Таблица 6.3 – Кодировка входных значений

№	Переменная	Кодировка значений
1	Cap-shape	$b \rightarrow (-1, -1, -1)$ $c \rightarrow (1, 1, 1)$ $x \rightarrow (-1, -1, 1)$ $f \rightarrow (1, 1, -1)$ $k \rightarrow (-1, 1, -1)$

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

2	Cap-surface	$\left\{ \begin{array}{l} f \rightarrow (-1, -1) \\ g \rightarrow (-1, 1) \\ y \rightarrow (1, -1) \\ s \rightarrow (1, 1) \end{array} \right.$
3	Cap-colour	$\left\{ \begin{array}{l} n \rightarrow (-1, -1, -1, -1) \\ b \rightarrow (1, 1, 1, 1) \\ c \rightarrow (-1, -1, -1, 1) \\ g \rightarrow (1, 1, 1, -1) \\ r \rightarrow (-1, -1, 1, -1) \\ p \rightarrow (1, 1, -1, 1) \\ u \rightarrow (-1, -1, 1, 1) \\ e \rightarrow (1, 1, -1, -1) \\ w \rightarrow (-1, 1, -1, -1) \\ y \rightarrow (1, -1, 1, 1) \end{array} \right.$
4	Bruises?	$\left\{ \begin{array}{l} t \rightarrow (1) \\ f \rightarrow (-1) \end{array} \right.$
5	Odor	$\left\{ \begin{array}{l} a \rightarrow (-1, -1, -1, -1) \\ l \rightarrow (1, 1, 1, 1) \\ c \rightarrow (-1, -1, -1, 1) \\ y \rightarrow (1, 1, 1, -1) \\ f \rightarrow (-1, -1, 1, -1) \\ m \rightarrow (1, 1, -1, 1) \\ n \rightarrow (-1, -1, 1, 1) \\ p \rightarrow (1, 1, -1, -1) \\ s \rightarrow (-1, 1, 1, -1) \end{array} \right.$
6	Gill-attachment	$\left\{ \begin{array}{l} a \rightarrow (1) \\ f \rightarrow (-1) \end{array} \right.$
7	Gill-spacing	$\left\{ \begin{array}{l} c \rightarrow (-1) \\ w \rightarrow (1) \end{array} \right.$
8	Gill-size	$\left\{ \begin{array}{l} b \rightarrow (1) \\ n \rightarrow (-1) \end{array} \right.$

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

9	Gill-colour	$\left\{ \begin{array}{l} k \rightarrow (-1, -1, -1, -1) \\ n \rightarrow (1, 1, 1, 1) \\ b \rightarrow (-1, -1, -1, 1) \\ h \rightarrow (1, 1, 1, -1) \\ g \rightarrow (-1, -1, 1, -1) \\ r \rightarrow (1, 1, -1, 1) \\ o \rightarrow (-1, -1, 1, 1) \\ p \rightarrow (1, 1, -1, -1) \\ u \rightarrow (-1, 1, -1, -1) \\ e \rightarrow (1, -1, 1, 1) \\ w \rightarrow (-1, 1, -1, 1) \\ y \rightarrow (1, -1, 1, -1) \end{array} \right.$
10	Stalk-shape	$\left\{ \begin{array}{l} e \rightarrow (-1) \\ t \rightarrow (1) \end{array} \right.$
11	Stalk-surface-above-ring	$\left\{ \begin{array}{l} f \rightarrow (-1, -1) \\ y \rightarrow (1, 1) \\ k \rightarrow (1, -1) \\ s \rightarrow (-1, 1) \end{array} \right.$
12	Stalk-surface-below-ring	$\left\{ \begin{array}{l} f \rightarrow (-1, -1) \\ y \rightarrow (1, 1) \\ k \rightarrow (1, -1) \\ s \rightarrow (-1, 1) \end{array} \right.$
13	Stalk-colour-above-ring	$\left\{ \begin{array}{l} n \rightarrow (-1, -1, -1, -1) \\ b \rightarrow (1, 1, 1, 1) \\ c \rightarrow (-1, -1, -1, 1) \\ g \rightarrow (1, 1, 1, -1) \\ o \rightarrow (-1, -1, 1, -1) \\ p \rightarrow (1, 1, -1, 1) \\ e \rightarrow (-1, -1, 1, 1) \\ w \rightarrow (1, 1, -1, -1) \\ y \rightarrow (-1, 1, 1, -1) \end{array} \right.$

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

14	Stalk-colour- below-ring	$\left\{ \begin{array}{l} n \rightarrow (-1, -1, -1, -1) \\ b \rightarrow (1, 1, 1, 1) \\ c \rightarrow (-1, -1, -1, 1) \\ g \rightarrow (1, 1, 1, -1) \\ o \rightarrow (-1, -1, 1, -1) \\ p \rightarrow (1, 1, -1, 1) \\ e \rightarrow (-1, -1, 1, 1) \\ w \rightarrow (1, 1, -1, -1) \end{array} \right\}$ $\left \begin{array}{l} y \rightarrow (-1, 1, 1, -1) \end{array} \right.$
15	Veil-type	$p \rightarrow 1$
16	Veil-colour	$\left\{ \begin{array}{l} n \rightarrow (-1, -1) \\ o \rightarrow (1, 1) \\ w \rightarrow (1, -1) \end{array} \right\}$ $\left \begin{array}{l} y \rightarrow (-1, 1) \end{array} \right.$
17	Ring-number	$\left\{ \begin{array}{l} n \rightarrow (-1, -1) \\ o \rightarrow (1, 1) \\ t \rightarrow (-1, 1) \end{array} \right\}$
18	Ring-type	$\left\{ \begin{array}{l} e \rightarrow (-1, -1, -1) \\ f \rightarrow (1, 1, 1) \\ l \rightarrow (-1, -1, 1) \\ n \rightarrow (1, 1, -1) \end{array} \right\}$ $\left \begin{array}{l} p \rightarrow (-1, 1, 1) \end{array} \right.$
19	Spore-print- colour	$\left\{ \begin{array}{l} k \rightarrow (-1, -1, -1, -1) \\ n \rightarrow (1, 1, 1, 1) \\ b \rightarrow (-1, -1, -1, 1) \\ h \rightarrow (1, 1, 1, -1) \\ r \rightarrow (-1, -1, 1, -1) \\ o \rightarrow (1, 1, -1, 1) \\ u \rightarrow (-1, -1, 1, 1) \\ w \rightarrow (1, 1, -1, -1) \end{array} \right\}$ $\left \begin{array}{l} y \rightarrow (-1, 1, 1, -1) \end{array} \right.$
20	Population	$\left\{ \begin{array}{l} a \rightarrow (-1, -1, -1) \\ c \rightarrow (1, 1, 1) \\ n \rightarrow (-1, -1, 1) \\ s \rightarrow (1, 1, -1) \\ v \rightarrow (-1, 1, -1) \end{array} \right\}$

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

21	Habitat	$\left\{ \begin{array}{l} g \rightarrow (-1, -1, -1) \\ l \rightarrow (1, 1, 1) \\ m \rightarrow (-1, -1, 1) \\ p \rightarrow (1, 1, -1) \\ u \rightarrow (-1, 1, -1) \\ w \rightarrow (1, -1, 1) \\ d \rightarrow (-1, 1, 1) \end{array} \right.$
----	---------	--

Таким образом, размерность входного вектора сильно изменилась от первоначальных 21 переменных к 52. И теперь возвращаемся к вопросу 5. Мы видим, что входной вектор сильно вырос. По возможности, нужно стараться не расширять вектор входных значений, а сужать его, т.к. существует такая проблема, как «проклятие размерности», когда с ростом входа количество требуемых данных для обучения растёт экспоненциально. Конечно, в данном случае рост входа компенсируется тем, что добавленные нейроны принимают только два значения. Также учтём, что все входные переменные категориальные, а это означает, что в этой задаче мы можем и не расширять входной вектор.

Вопрос 6. Как кодировать класс, который нужно определить?

Так как мы приняли решение не расширять входной вектор, то выход можно просто отнормировать к диапазону $-1..+1$ вместо 0 и 1.

Если всё же расширять вход, то учитывая больший размер входного паттерна, ещё более необходимо каждый класс представить своим нейроном, а не использовать один нейрон на два класса: $e \rightarrow (-1, 1)$, $p \rightarrow (1, -1)$.

Вопрос 7. К какому диапазону приводить остальные значения?

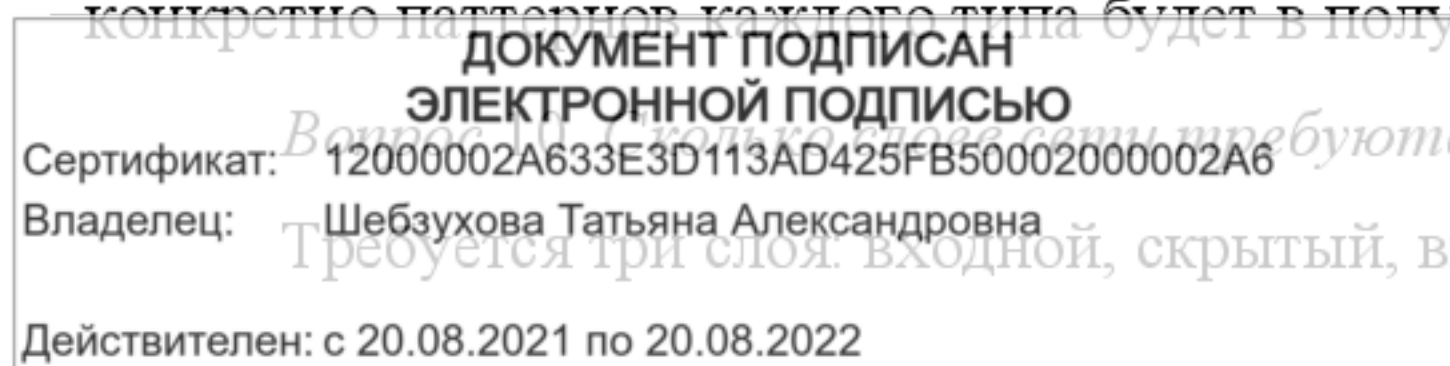
Все 21 входных значений можно отнормировать к диапазону $-1..+1$ для увеличения скорости обучения.

Вопрос 8. Какая пропорция берётся для обучающей и тестовой выборки?

90% - обучающая выборка, 10% - тестовая выборка, 10% - валидационная выборка.

Вопрос 9. Как формировать обучающую и тестовую выборки?

Классы в искомой выборке представлены равномерно (51.8% и 48.2%, что почти одно и то же). Тогда можно случайным образом (средствами Matlab) сформировать обучающую и тестовую выборки соответствующих размеров, не заботясь сколько конкретно паттернов каждого типа будет в получившихся выборках.



Вопрос 11. Сколько нейронов в каждом слое?

Во входном слое 21 нейрон, в выходном 1 нейрона, в скрытом можно взять 10 нейронов.

Вопрос 12. Какие связи между слоями?

Сеть полносвязанная.

Вопрос 13. Какие функции активации?

На скрытом слое – гиперболический тангенс, на выходном слое – аналогично или линейная функция активации.

Вопрос 14. Какая функция ошибки?

Среднеквадратическая ошибка (MSE).

Листинг программы приведён ниже. Использовалась выборка «DataForMushrooms_num.xls», в которой символы были заменены числами от 0 до количества переменных в системе из таблицы 6.3 для соответствующей строки.

```
initSet = xlsread('C:\Users\Computer Grand\Desktop\МАТЕРИАЛЫК лабе  
6\DataForMushrooms_num.xls', 1, 'B:W');  
initSet = initSet;  
% Приводим выходы сети к диапазону -1..+1, ниже в коде есть альтернатива  
initSet(22, :) = mapminmax(initSet(22, :));  
input_data = initSet(1:21,:);  
output_data = initSet(22,:);  
net = newff(input_data, output_data, [10], {'tansig', 'tansig'}, 'trainlm', 'learngd', 'mse', {},  
{}, 'dividerand');  
% Ограничиваем обучение по количеству эпох  
net.trainParam.epochs = 100;  
net.trainParam.max_fail = 2;  
% Разбиваем выборку  
net.divideParam.trainRatio = 0.8;  
net.divideParam.testRatio = 0.1;  
net.divideParam.valRatio = 0.1;  
% Устанавливаем номерку одну для всех входов  
net.inputs{1}.processFcns = {'mapminmax'};  
% Можно альтернативно отнормировать выходы сети  
% net.outputs{2}.processFcns = {'mapminmax'};  
% инициализируем сеть нашими параметрами  
net = init(net);  
[net, tr] = train(net, input_data, output_data);  
testSet = input_data(:, tr.testInd(:));  
testT = output_data(:, tr.testInd(:));  
Y = sim(net, testSet);  
col = 0;  
for i = 1:length(testSet)
```

```
if norm(testT(i) - Y(i)) < 0.1
```

```
col = col + 1;
```

```
end
```

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

% выводим процент правильных ответов на тесте

percent = 100 * col / length(testT)

Как видно из рисунка 6.4 обучение осуществляется почти идеально. Если мы не отнормируем выходные значения, оставив их в диапазоне 0 и 1, а входы переведём в диапазон -1..+1, то процесс обучения сильно затормозится, как показано на рисунке 6.5. Нам потребуется не 12, а 100 эпох.

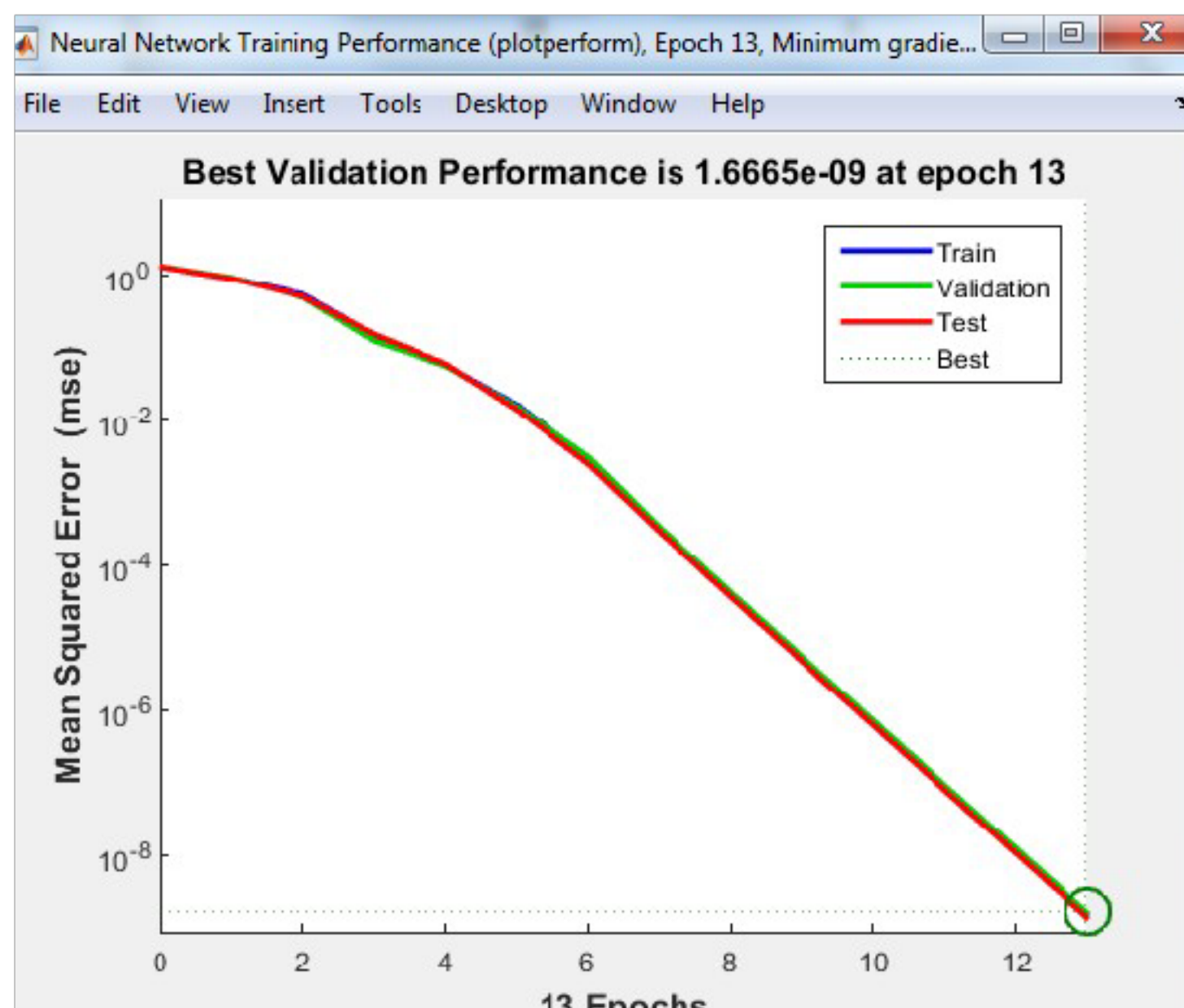


Рисунок 6.4 – Обучение сети со всеми отнормированными данными к диапазону -1..+1

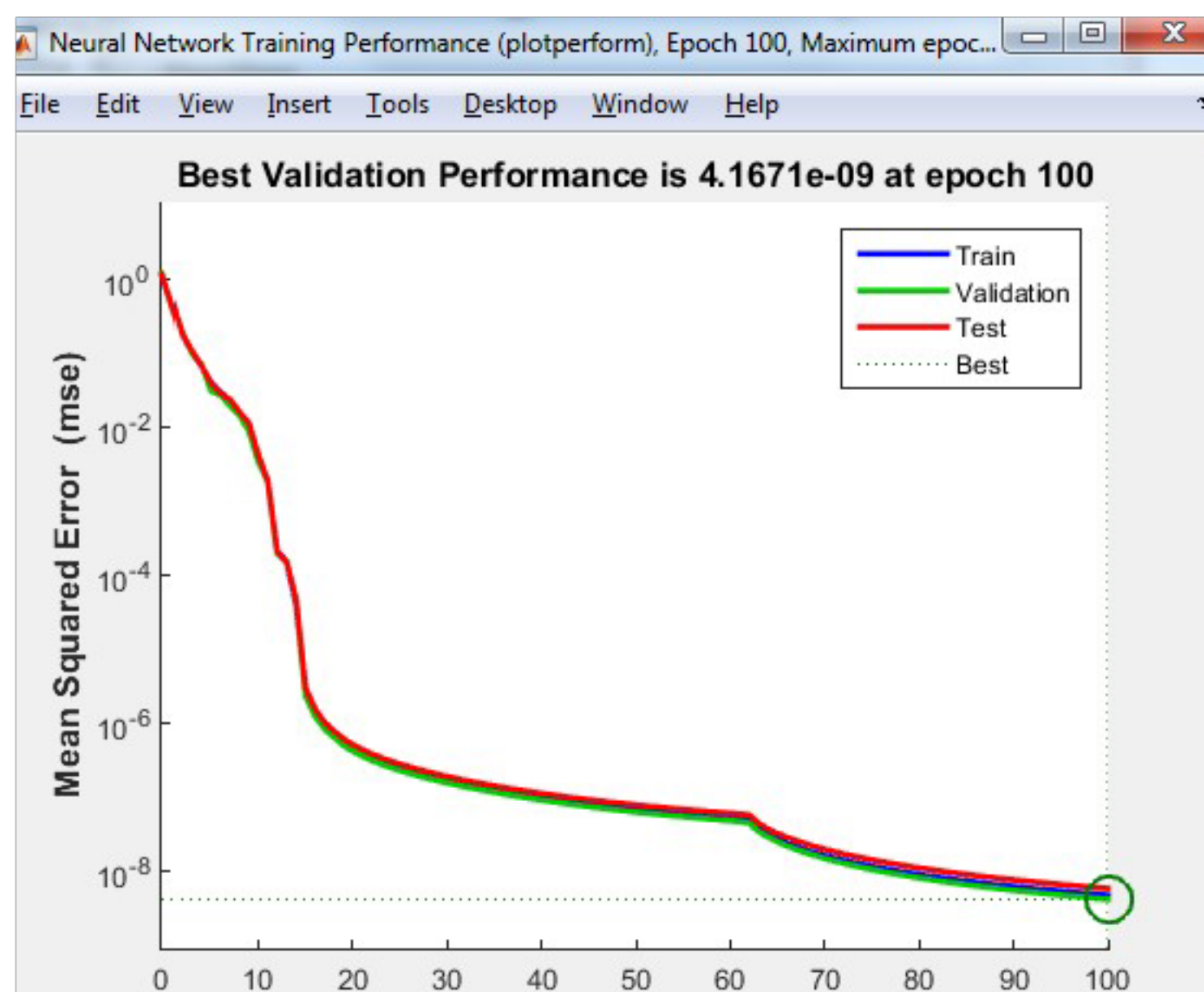


Рисунок 6.5 – Обучение сети, где отнормированы только входные данные

На тестовом множестве мы распознаём все паттерны. Студент может попробовать самостоятельно решить данную задачу, изменив ошибку обучения с MSE на CrossEntropy,

тогда количество использованных итераций алгоритма trainrp (RPROP) составит в

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Щербанова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

В индивидуальном варианте (таблица 6.4) дано условие задачи на классификацию. Приемлемая точность на тестовой выборке везде отмечена как 70%, что в целом не много. Достижение более большой точности распознавания оценивается в дополнительных баллах.

Таблица 6.4 – Варианты заданий

№	Условие задачи
1	Решить задачу о грибах с расширением входного вектора и выходного так, как это описано в разборе второй задачи. Сравнить результаты с обучением на рисунке 6.4.
2	В США в Аризоне существует племя индейцев Пима (https://ru.wikipedia.org/wiki/Пима). Особенность этого племени в том, что у его представителей самый высокий в мире процент заболеваемости сахарным диабетом 2-го типа. Среди женщин этого племени было проведено исследование (спонсируемое Всемирной Организацией Здравоохранения, речь идёт об исследовании 1988 г., т.к. были и другие). Одним из результатов данного исследования было то, что была составлена выборка, где отмечали наличие или отсутствие диабета в зависимости от разных факторов (адрес выборки: https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes/). Выборка состоит из 768 записей. Всего 9 столбцов, 8 из них – показатели, а 1 – флаг (наличие или отсутствие диабета). Классы представлены не равномерно: женщины у которых нет диабета 500 штук, у которых есть – 268 штук. Первый столбец – количество беременностей, второй – концентрация глюкозы в плазме в течении двух часов при использовании орального теста на концентрацию глюкозы, третий – кровяное давление, четвёртый – толщина сгибающей трёхглавой мышцы (Triceps skin fold thickness), пятый - 2-Hour serum insulin, шестой – индекс массы тела, седьмой - Diabetes pedigree function, восьмой – возраст, девятый – обозначение класса (0 – нет диабета, 1 – есть диабет). Для нейронной сети (многослойный персептрон) достигали результата в 76.4%

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Бушубина Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

3. Вспомогательные материалы: английские заглавные буквы, каждая буква была записана 20 различными шрифтами и к каждой такой записи применялись ещё случайные

	искажения. Всего было 20000 записей английских заглавных букв, все они располагались в рамке определённого размера. Потом через эту рамку проводилось ряд прямых, подсчитывалось количество пересечений этих прямых с буквой и результат записывался в вектор, также в вектор записывалась базовая информация о размерах рамки, количестве пикселей и т.д. Таким образом, удалось перейти от пиксельного представления буквы к её признаковому представлению (осуществить сжатие) в виде вектора размером в 16 позиций. Т.е. каждая английская буква определённого шрифта и с определённым искажением описывается таким вектором. Требуется по входному вектору классифицировать букву. Подробную информацию о сети и выборке можно получить из статьи Hussein Salim Qasim "Letter Recognition Data Using Neural Network" (https://www.researchgate.net/publication/289129248_Letter_Recognition_Data_Using_Neural_Network) или найти информацию в интернете по запросу «letter dataset» . На примере этой задачи видно, как сильно изменились нейросетевые технологии классификации. Раньше отдельным этапом осуществляли предобработку входных данных, причём часто этот этап был весьма нетривиальным. Теперь стандарт – подача на вход сети пиксельного изображения. Похожая задача рассматривается в варианте 10.
4	Классификация типов узлов скрещивания в цепочках ДНК (http://www.cs.toronto.edu/~dave/data/splice/spliceDetail.html). В данной задаче рассматриваются цепочки ДНК, состоящих из последовательностей аминокислот (А, С, Т и G) длиной 60 каждая. В каждом примере середина цепочки является кандидатом на узел скрещивания (рассматриваются 3 класса - 0 - переход типа "интрон-эксон", 0.5 - незначимый переход, 1 - переход "эксон-интрон"). Аминокислоты закодированы целыми числами А - 0, С - 1, Т - 2, G - 3. Требуется построить нейросетевую систему классификации. Данные (3175 примеров) содержатся в файле. 2500 или 2000 первых примеров используются для обучения, остальные - для тестирования. Требуется построить нейросетевую классификационную систему с наименьшей ошибкой тестирования. 60 входных значений обычно переводятся в двоичный код, и вход расширяется либо до 180, либо до 240 значений. Возможные распределения для тестовой и обучающей выборки: Обучение, 1-(464, 23.20%), 2-(485, 24.25%), 3-(1051, 52.55%); Тест, 1-(303, 25.55%), 2-(280, 23.61%), 3-(603, 50.84%).
5	Задача об оценке эффективности преподавания учителя на основе данных обучения 3 регулярных семестров и двух летних в статистическом отделении университета Висконсин-Мэдисон. Выходной класс – оценка (1 – низкая, 2 – средняя, 3 - высокая). Входной вектор: Native (английский родной – 1, не родной - 2), Instructor (25 категорий для инструктора курса), Course (26 категорий), Semester (1 – летний, 2 - регулярный), Size (числовой вход – размер класса). Всего имеется 151 запись, пропущенных значений нет.
6	Из переписи данных США 1994 взята небольшая выборка о взрослых людях. Она содержит 14 признаков (более подробно они описаны в файле к заданию). Требуется предсказать получает ли этот человек в год более 50000 долларов или нет. Выборка имеет пропущенные данные.
7	Шкала баланса. Имеются весы с двумя плечами, однако длины левого и правого плеча могут меняться. К каждому плечу подвешивался груз разной массы. От ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 12000002A633E3D113AD425FB50002000002A6 Владелец: Шебзухова Татьяна Александровна есть ли весы сбалансированы или отклонятся его длина. Есть три выходных класса: В – баланс (когда произведение левого плеча

	на левый груз равно произведению правого плеча на правый груз), R – отклонение вправо, L – отклонение влево. Пропущенных данных нет. Более подробная информация в файле задания.
8	Нарушение печени. Имеются 5 входных параметров, которые анализируют работу печени. Требуется определить является ли человек алкоголиком или нет. Более подробная информация в файле задания.
9	Набор данных о выживании Хабермана. Этот набор данных содержит случаи из исследования, которое проводилось между 1958 и 1970 годами в больнице Биллингса Университета Чикаго по выживанию пациентов, перенесших операцию по удалению рака из молочной железы. Задача состоит в том, чтобы определить, выжил ли пациент в течении последующих 5 лет или более (положительный результат), или нет (отрицательный результат). Пропущенных данных нет. Более подробная информация в файле задания.
10	Из силуэта транспортного средства извлекли ряд признаков. По ним требуется классифицировать тип транспортного средства. Задача похожа на задачу из варианта 3. Более подробная информация в файле задания.
11	Имеется ряд параметров, измеряющих работу щитовидной железы. Требуется установить является ли человек нормальным (1) или страдает гипертиреозом (2), или гипотиреозом (3). Более подробная информация в файле задания.
12	Набор данных дрожжей. Эта база данных содержит информацию о наборе дрожжевых клеток. Задача состоит в том, чтобы определить локализацию каждой клетки среди 10 возможных вариантов. Более подробная информация в файле задания.

Содержание отчета и его форма

Отчёт по лабораторной работе должен содержать следующую информацию:

1. Название лабораторной работы и её номер.
2. ФИО студента и группу.
3. Формулировка индивидуального задания.
4. Документ отчёта с Print Pritscr диалоговых окон по шагам для своего варианта по подобию того, что описано в теоретической части. Должна присутствовать информация о количестве классов, количестве паттернов в каждом классе, о размерах тестовой и обучающей выборки, о проделанной предобработке, о представленности каждого класса в тестовой и обучающей выборке, об используемых алгоритмах обучения нейронных сетей, а также результирующие графики и процент правильно распознанных паттернов на тестовой выборке. Необходимо также указать структуру сети и обосновать почему выбрана именная такая сеть.
5. Ответы на контрольные вопросы.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

вопросы для защиты работы

1) С какими типами переменных может работать нейронная сеть?

- 2) Какими способами можно предобработать входные и выходные данные для нейронной сети?
- 3) Чем ординарные данные отличаются от категориальных?
- 4) На какие ключевые вопросы нужно дать ответы для этапа предобработки данных?
- 5) Как нужно разбить выборку для теста и обучения если имеются два неравномерно представленных класса?
- 6) Как нужно разбить выборку для теста и обучения если имеются два равномерно представленных класса?
- 7) Какие типы нормировок вы знаете?

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

Лабораторная работа № 7

Пример создания и обучения нейронных сетей для задач регрессии в среде Matlab.

Цель и содержание работы: создать и обучить нейронную сеть для задачи регрессии.

Задачи:

- понять каким образом нейронные сети решают задачу регрессии;
- рассмотреть постобработку данных;
- создать и обучить нейронную сеть для задачи регрессии.

Теоретическое обоснование

В предыдущих лабораторных работах рассматривалось, как нейронные сети могут осуществлять классификацию входных паттернов, однако нейронные сети можно использовать и для осуществления предсказаний значений (задача регрессии). Предсказание отличается от задачи классификации тем, что на выходе сеть должна получить некоторое значение, а не некоторую фиксированную метку класса. Тем не менее, процесс обучения сети остаётся тем же. Каждый входной паттерн и выходную метку можно рассматривать как некоторую координату в многомерном пространстве, тогда задача обучения сводится к тому, чтобы аппроксимировать эти данные некоторой нелинейной моделью, так чтобы ещё появились интерполирующие свойства. Сеть добивается для каждой такой точки получать наиболее близкий ответ к учителю, и за счёт этого, по факту, сама сеть превращается в некоторую аппроксимирующую модель, на рисунке 7.1 – это кривая, проходящая через точки обучающего набора. Однако через набор таких точек можно провести бесконечно много кривых, на рисунке 7.1 вверху и внизу проходят разные кривые. Но нетрудно заметить, что нижняя кривая не имеет предсказательной способности (говорят, что сеть переобучилась), а верхняя кривая вполне подходит для предсказания ответа на тестовых данных (кружок на рисунке 7.1).

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

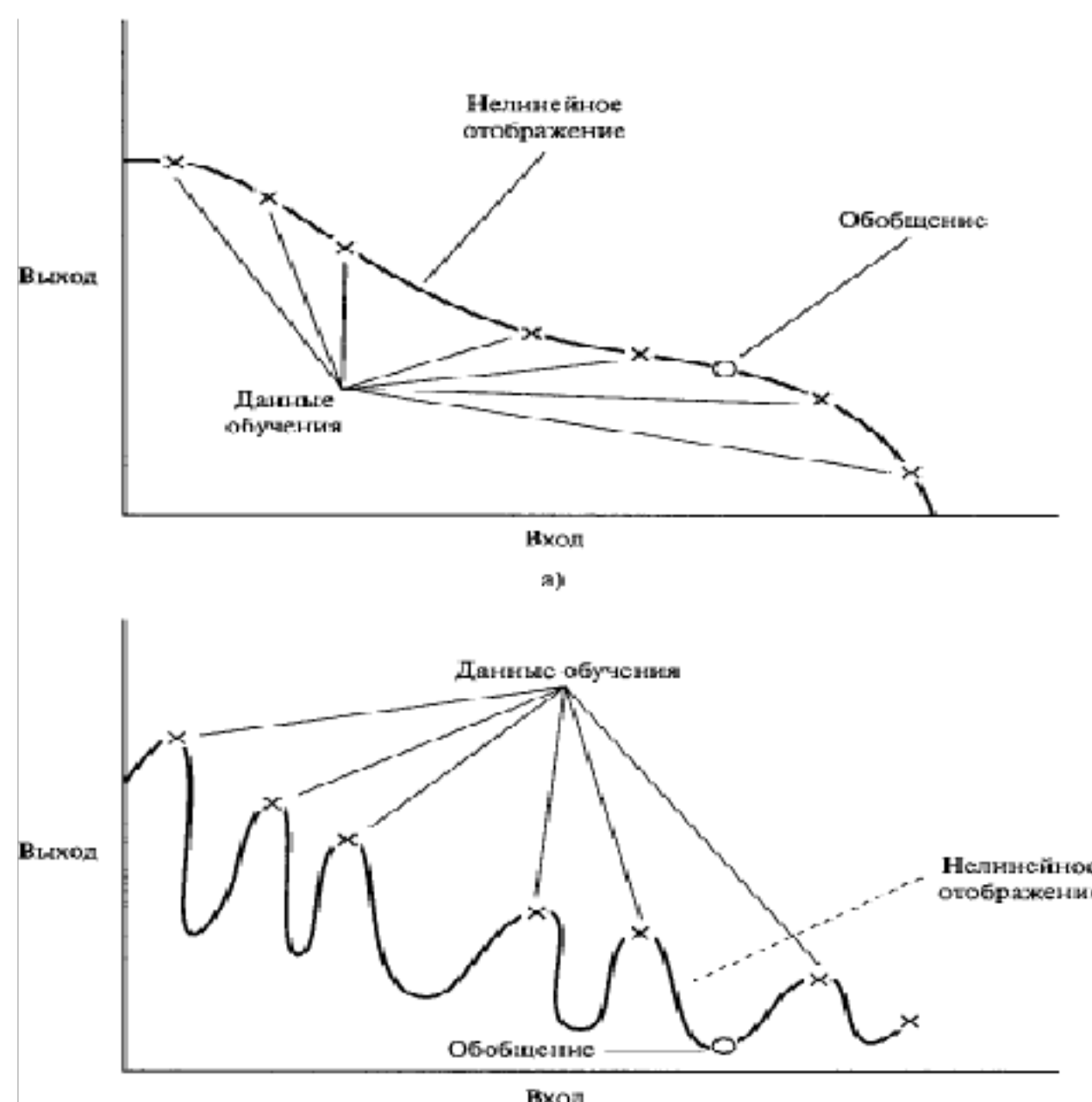


Рисунок 7.1 – Две кривые аппроксимирующие один и тот же набор данных

Интерполяционная способность достигается ростом объема обучающей выборки до момента описания выборкой всех статистических свойств генеральной совокупности и контролем размера и свойств сети. Хорошая аппроксимация ещё ничего не говорит про ошибку обобщения.

Нейронные сети, как уже отмечалось выше, строят нелинейные модели. Самая же простая модель для предсказания – это линейная регрессия. В файле «DataForRegression.xls» на листе «Пример» в столбцах А и В приведена простая зависимость количества предметов от месяца. Если ось ОХ будет обозначать номер месяца, а ось ОУ будет обозначать количество предметов, то легко можно построить график (синий цвет на рисунке 7.2).

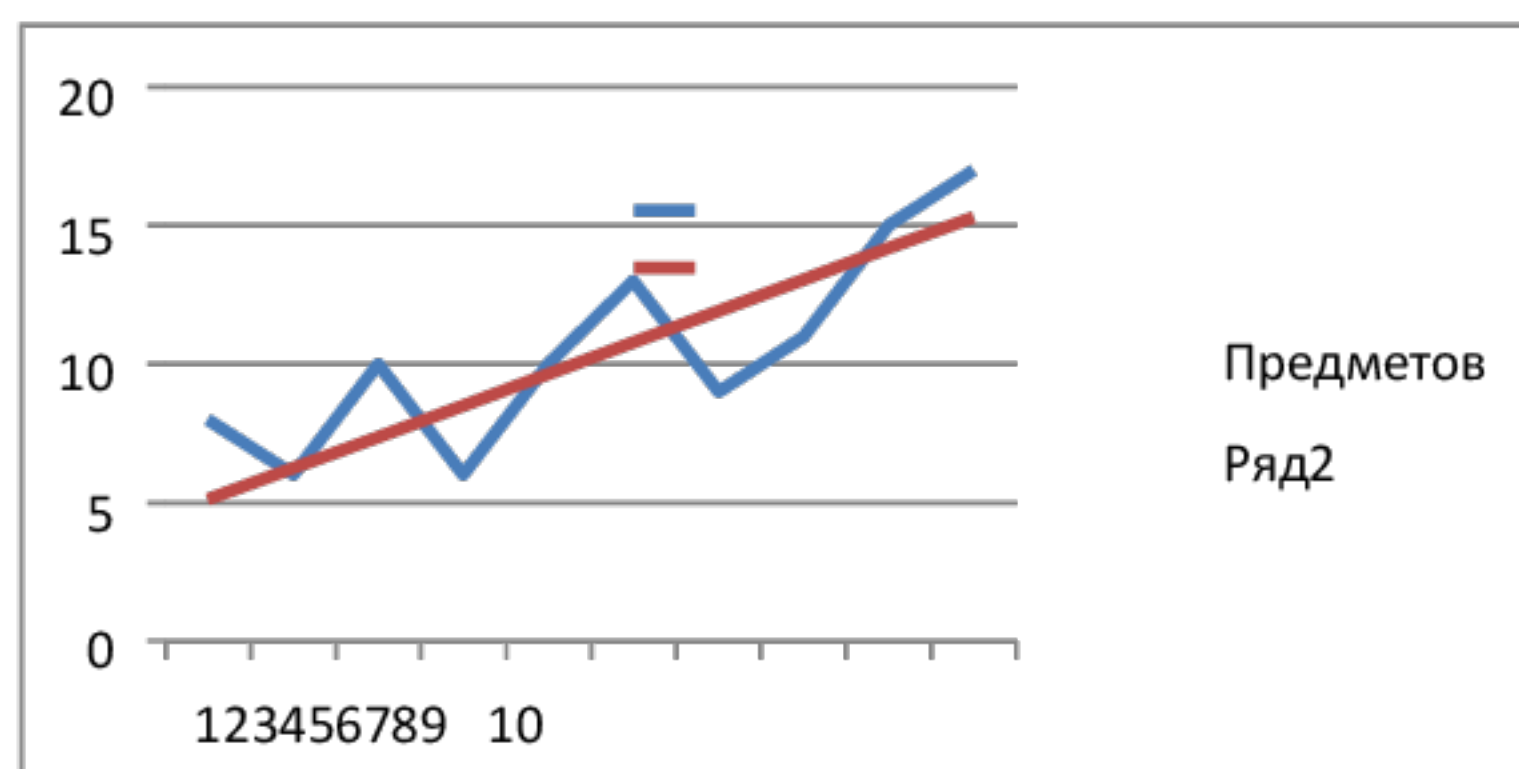


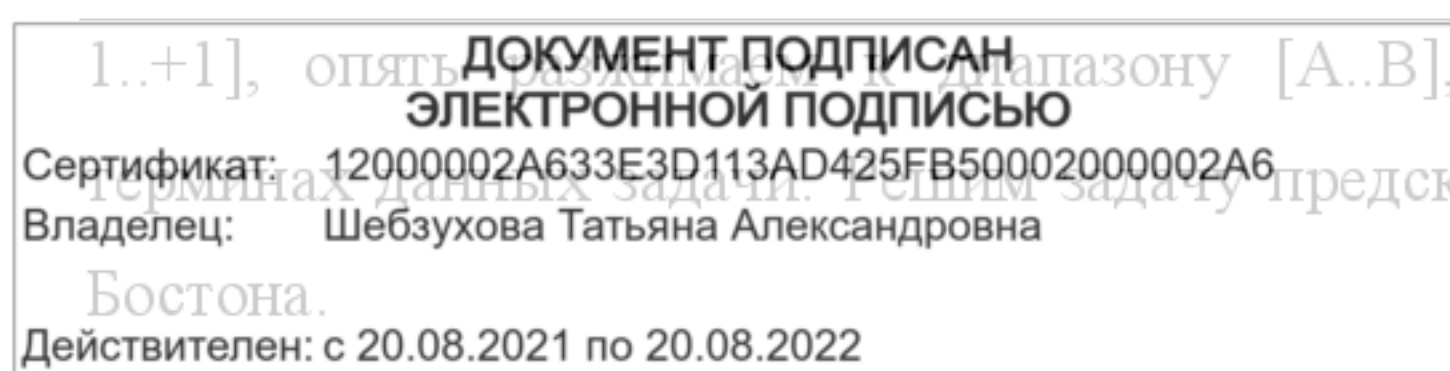
Рисунок 7.2 – Исходные данные и регрессионная модель

так бы проходила около или через эти точки, что давала бы наиболее близкие ответы к исходным данным. Коэффициенты этой прямой выделены жёлтым цветом в столбце F. Получилось следующее уравнение прямой: $y=1.13*x+3.97$. Это и есть красная прямая на рисунке 7.2. Значения, предсказанные этой прямой для соответствующих месяцев, приведены в столбце A и B. В данном примере мы можем легко визуальным образом построить модель, но в реальности, если входных значений будет много, то мы не сможем визуальным образом отобразить нашу многомерную прямую. Так на листе «Задача» в столбцах A-L даются данные о стоимости домов в одном из округов США в 2014-2015 годах. Price – цена дома в долларах, train – метка для обучающей или тестовой выборки (1 – паттерн для обучающей выборки, 0 – для тестовой, они выделены красным цветом), bathrooms – количество ванных комнат, sqft_living – жилая площадь, waterfront – близко ли дом к воде, возможные значения [0, 1], view – характеристика вида возле дома в некоторой шкале возможных значений [0, 1, 2, 3, 4], condition – состояние дома в некоторой шкале возможных значений [1, 2, 3, 4, 5], grade – общая оценка дома по некоторой шкале [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13], yr_built – год постройки, zipcode – индекс из адреса дома, lat – широта из координат дома на карте, long – долгота из координат дома на карте. Соответственно требуется с помощью линейной регрессии предсказать цену дома. В столбцах P-Y представлены результаты подбора коэффициентов для данной регрессионной задачи. Получается уравнение 10-мерной прямой. Коэффициенты для этой прямой выделены жёлтым цветом, а в столбце N представлены ответы данной модели на входные данные, т.е. предсказанная цена.

Так вот, нейронная сеть также строит некоторую, но уже нелинейную модель, отсюда её предсказательные способности ещё больше, но получить готовую модель сложнее из-за риска переобучения. Переобучение наступает тогда, когда сеть имеет избыточные ресурсы (много настраиваемых параметров по отношению к обучающим данным), тогда сеть моделирует слишком сложную зависимость вместо более простой.

При использовании нейронной сети для предсказаний возникает задача обработать не только входные, но и выходные данные. Обычно выходные данные также приводятся к диапазону 0..1 или -1..1, а когда сеть тестируют на тестовой выборке, то выходные значения вновь расширяют к искомому диапазону. Если, допустим, изначально выходные значения были в диапазоне [A..B], то для обучения мы сжимаем их по формулам нормировки (лабораторная работа 6) к диапазону [-1..+1], а при тесте ответ сети, данные из диапазона [-

1..+1], опять сжимаем к диапазону [A..B], чтобы получился осмысленный ответ в терминах данных задачи. Решим задачу предсказания стоимости недвижимости в районах



Прогноз стоимости недвижимости в районах Бостона.

Исходные данные содержатся в файле «DataForBoston.doc», однако, учитывая, что это одна из задач-примеров в Matlab, исходные данные можно загрузить и через команду.

Требуется построить прогностическую систему для оценки стоимости жилья в Бостоне, основываясь на следующих 13 параметрах, в скобках приведены диапазоны, которые может принимать параметр:

1. CRIM – уровень преступности на душу населения [0.0063, 88.98).
2. ZN – доля жилых массивов на площади 25.000 футов² [0, 100].
3. NDUS – доля предприятий, не связанных с розничной торговлей [0.46, 27.74].
4. CHAS – близость к реке Charles River (1 – если район граничит с рекой, 0 – в противном случае).
5. NOX – концентрация окисей азота (в долях 1/10 миллионов) [0.385, 0.871).
6. RM – среднее число комнат в жилище [3.561, 8.78].
7. AGE – доля частных владений, построенных до 1940 г [2.9, 100].
8. DIS – взвешенное расстояние до 5 центров сосредоточения работы в Бостоне [1.1296, 12.1265].
9. RAD – индекс доступности радиальных автострад [1..24].
10. TAX – полный налог на недвижимость в расчёте на \$10.000 [187, 711].
11. PTRATIO – доля школьных учителей в районе [12.6, 22].
12. B – коэффициент $10^3 \cdot (B_k - 0.63)^2$, где B_k – доля чернокожего населения [0.32, 396.9].
13. LSTAT – процент населения ниже черты бедности [1.73, 37.97].

MEDV – параметр, который нужно предсказать: средняя цена частного жилья в \$1000 [5, 50].

Всего выборка содержит 506 записей.

Выборку можно также скачать с сайта DELVE Project или с <http://mlearn.ics.uci.edu/MLRepository.html>. Более подробную информацию по выборке можно посмотреть в [1].

Приведём листинг решения с комментариями.

```
% Загрузка искомой выборки
load house_dataset;
% Входные значения
x = houseInputs;
```

```
% Выходные значения
t = houseTargets;
% Алгоритм обучения сети
trainFcn = 'trainlm';
% Коэффициент потерь в промежуточном слое
```

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

```

hiddenLayerSize = 15;
% Создание сети для предсказаний
net = fitnet(hiddenLayerSize,trainFcn);
% Обработать возможные пропуски во входных данных
% Используя линейную нормировку сжать все данные к диапазону [-1..+1]
net.input.processFcns = {'removeconstantrows','mapminmax'};
% Аналогично предобработать выходы
net.output.processFcns = {'removeconstantrows','mapminmax'};
% Случайно разбить выборку на обучающую и валидационную
net.divideFcn = 'dividerand';
net.divideMode = 'sample';
% В обучающей 85% всех паттернов
net.divideParam.trainRatio = 85/100;
% В валидационной 15%
net.divideParam.valRatio = 15/100;
% В тестовой ничего нет
net.divideParam.testRatio = 0/100;
net.performFcn = 'mse';
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', 'plotregression', 'plotfit'};
% Обучить сеть
[net,tr] = train(net,x,t);
% Получить ответы сети для валидационного множества
y1 = net(x(tr.valInd(1,:)));
% Получить идеальные ответы
t1=t(tr.valInd(1,:));
% Вычислить их разность
e1 = gsubtract(t1, y1);
% Посчитать количество таких разностей по модулю, которые
% меньше 10, выразить это число в процентах
col = length(find(abs(e1(1, :))<=10))/length(e1)*100;
% Получившейся ответ для задачи col = 59.21;

```

Процесс обучения и структура сети представлены на рисунке 7.3, видно, что на выходном слое находится линейная функция активации. Увеличивать количество нейронов в скрытом слое особого смысла не имеет, т.к. тогда при возрастании качества предсказания на обучающей выборки, будет падать качество предсказания на тестовой выборки. На рисунке 7.4 показан процесс изменения ошибки обучения и обобщения. На рисунке 7.5 и 7.6 представлены дополнительные графики, которые дают возможность более точно оценить степень предсказательной способности сети.

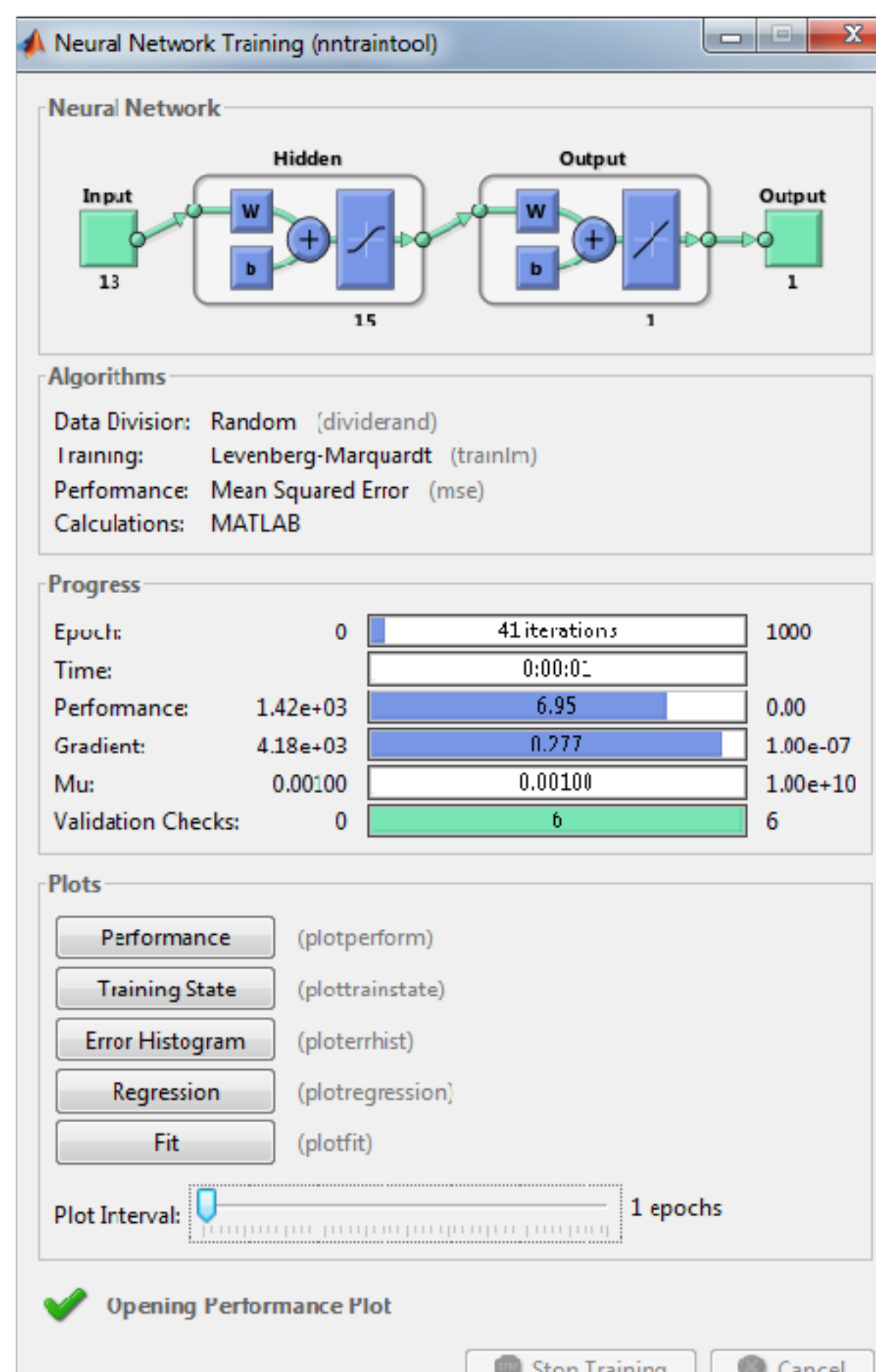


Рисунок 7.3 – Процесс обучения и структура сети для предсказания

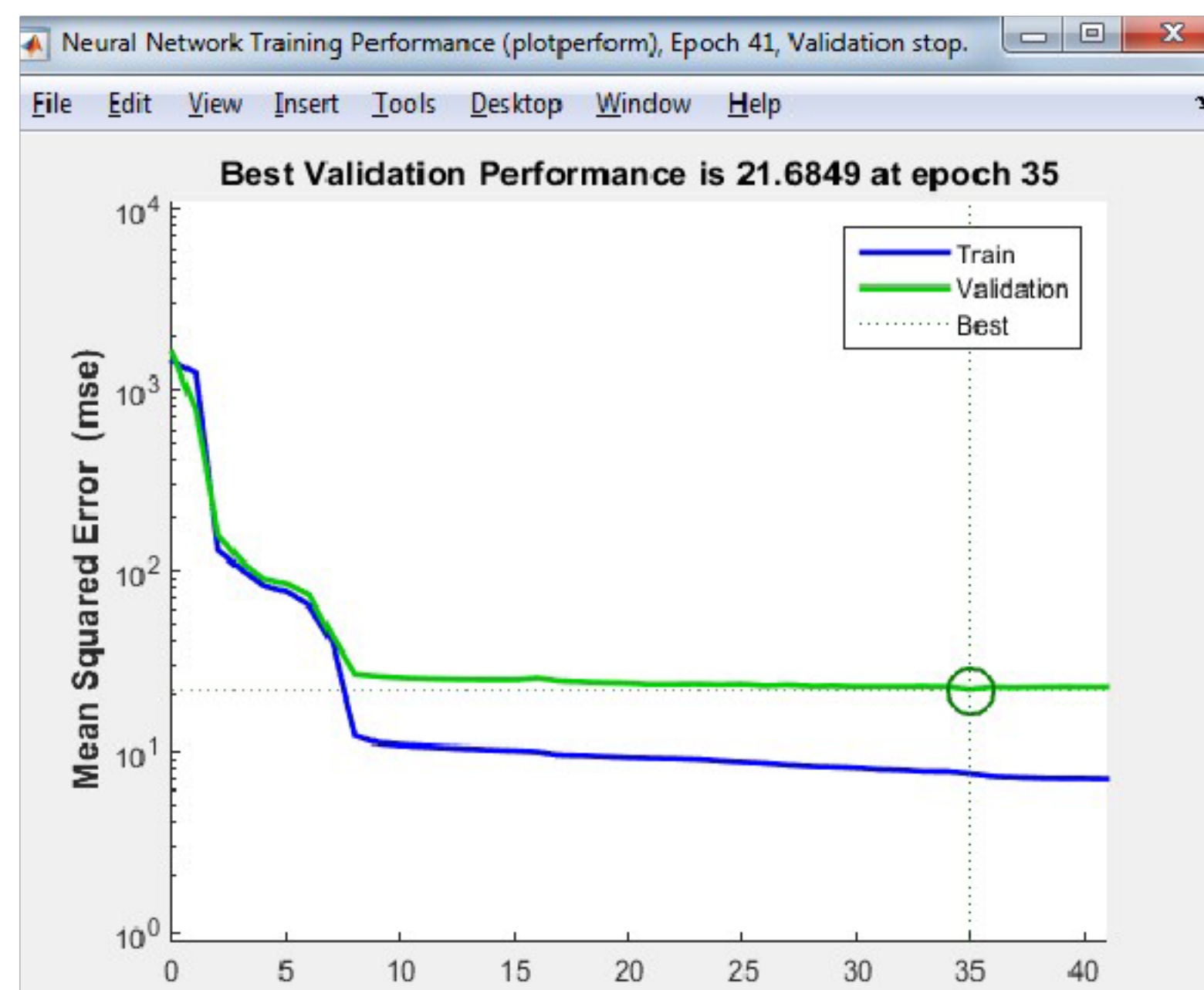


Рисунок 7.4 – Изменения ошибки обучения и ошибки обобщения

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

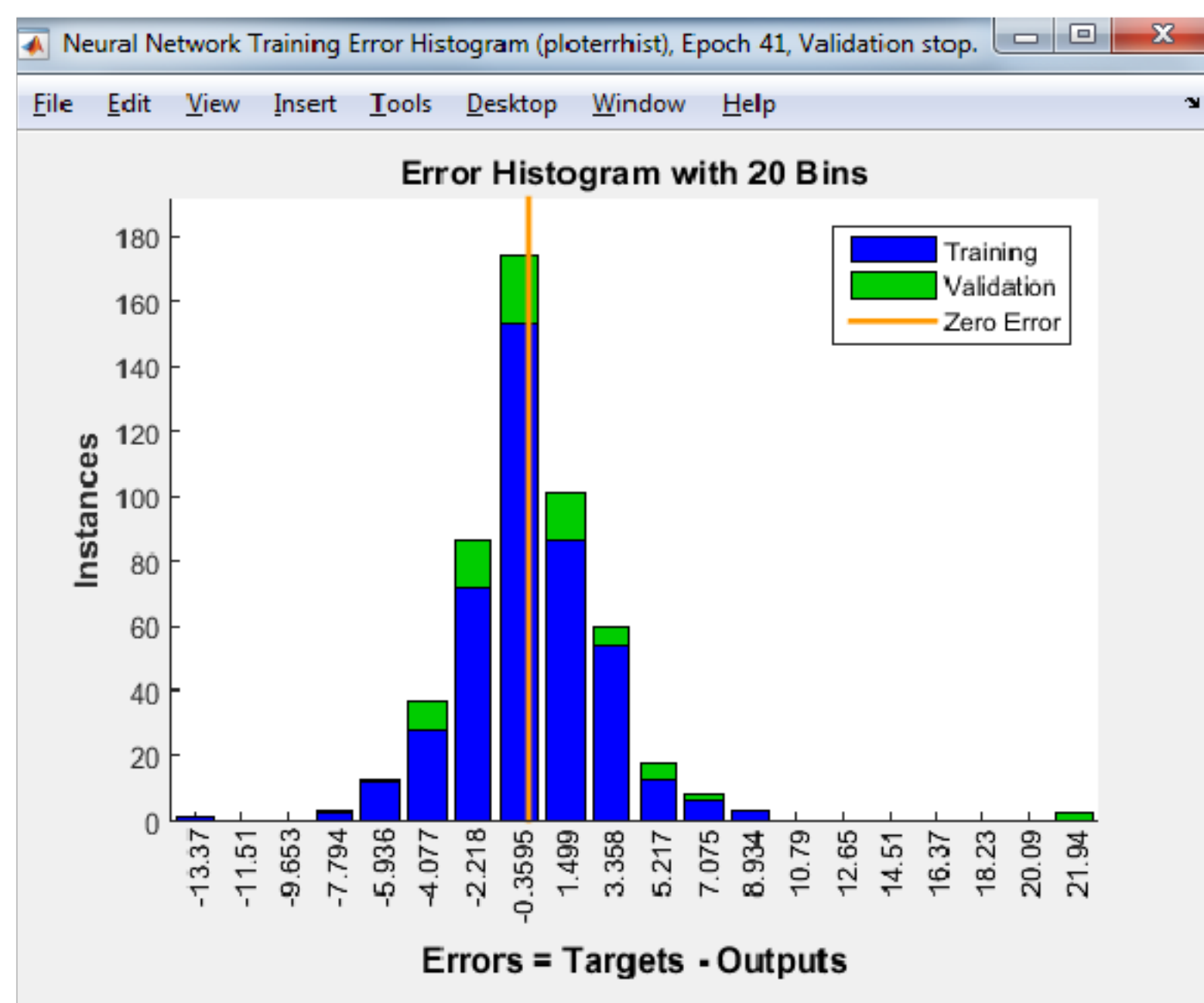


Рисунок 7.5 – Диаграмма распределения фактической ошибки при предсказании

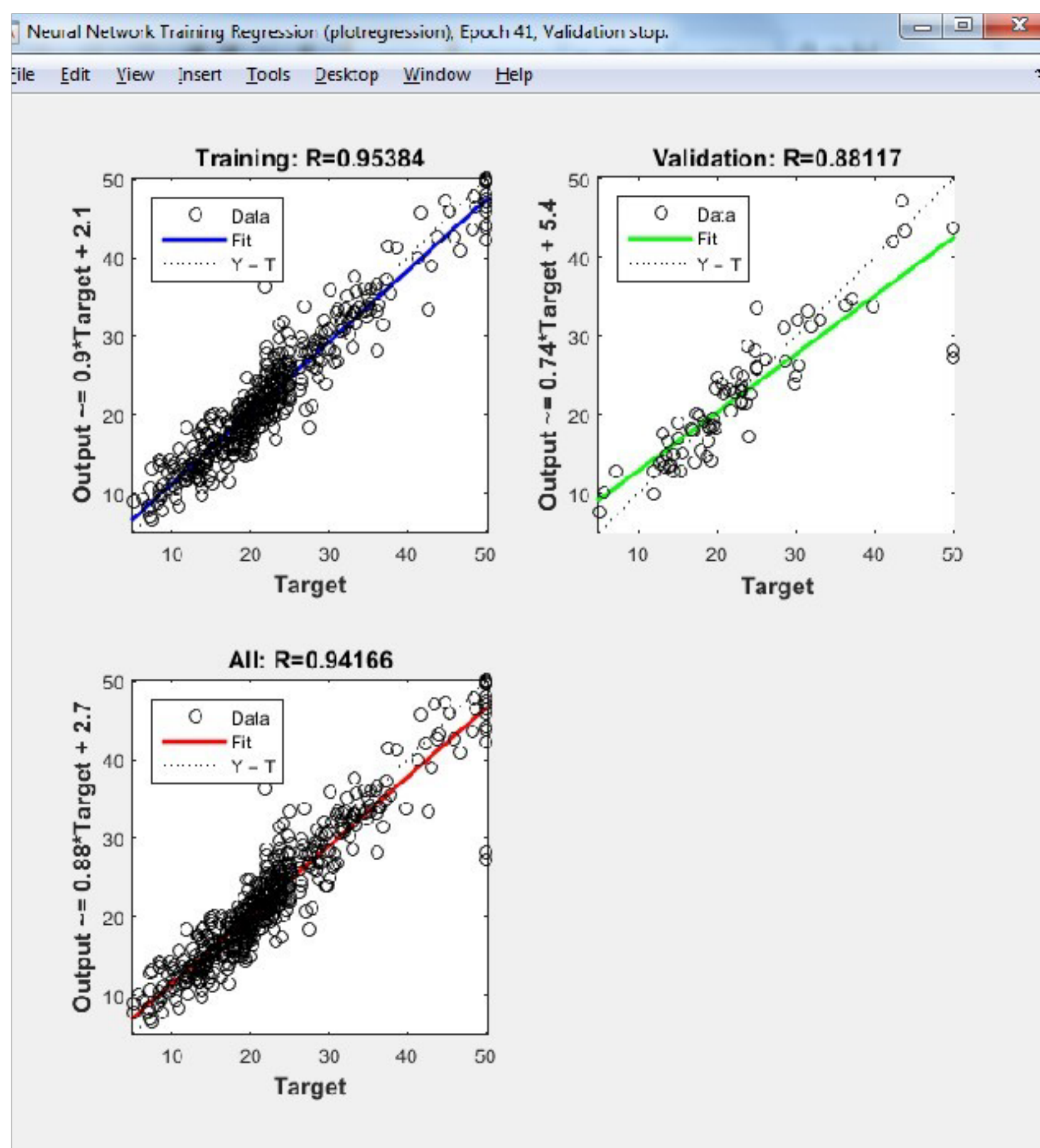


Рисунок 7.6 – Отношение данных (кружки) к получившимся моделям (линии)

В результате модели предсказали 60% значений, где предсказанная цена на 10 единиц отличается от идеальной требуемой.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

В индивидуальном варианте (таблица 7.1) дано условие задачи на регрессию. Решить задачу с точностью предсказания на тестовой выборке более 60%.

Таблица 7.1 – Варианты заданий

№	Условие задачи
1	Используя данные из задачи о стоимости жилья в США составить нелинейную регрессионную модель с помощью нейронной сети. Данные содержатся в листе Data файла «DataForRegression.xls». Сравнить ошибку MSE для предсказанных данных с помощью линейной регрессионной модели и предсказанных данных нейронной сетью.
2	Предсказать решение верховного суда. Описание входов и выхода смотрите в файле задания.
3	Предсказать расход топлива по городу. Имеется один дискретный вход и 5 непрерывных. Описание входов и выходов смотрите в файле задания.
4	Предсказать расход топлива по городу. Имеется 3 дискретных входа и 5 непрерывных. Описание входов и выходов смотрите в файле задания.
5	Предсказать среднюю стоимость жилья в Калифорнии. Задача аналогичная двум рассматриваемым в теоретической части. Описание входов и выходов смотрите в файле задания.
6	Предсказать силу сопротивления сжатию для цемента. Описание входов и выходов смотрите в файле задания.
7	Предсказать угол отклонения элерона для самолёта F16. Описание входов и выходов смотрите в файле задания.
8	Предсказать уровень С-пептидов в сыворотке крови детей больных диабетом. Описание входов и выходов смотрите в файле задания.
9	Требуется предсказать недвижимость домов в США в зависимости от демографических данных района и рынка недвижимости. Описание входов и выходов смотрите в файле задания.
10	Предсказать зависимость Y от $X_1..X_{10}$. Описание входов и выходов смотрите в файле задания.
11	Предсказать давление пластика при определённой силе и температуре. Описание входов и выходов смотрите в файле задания.

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

12	Предсказать угловое ускорение для одной из степени свободы механической руки Puma 560. Описание входов и выходов смотрите в файле задания. Информацию по механическому манипулятору можно найти в сети Интернет.
----	--

Содержание отчета и его форма

Отчёт по лабораторной работе должен содержать следующую информацию:

1. Название лабораторной работы и её номер.
2. ФИО студента и группу.
3. Формулировка индивидуального задания.
4. Документ отчёта с Print Prtscr диалоговых окон по шагам для своего варианта по подобию того, что описано в теоретической части. Должна присутствовать информация о размерах тестовой и обучающей выборки, о проделанной предобработке и постобработке, об используемых алгоритмах обучения нейронных сетей, а также результирующие графики и процент правильно распознанных паттернов на тестовой выборке. Необходимо также указать структуру сети и обосновать, почему выбрана именно такая сеть.
5. Ответы на контрольные вопросы.

Вопросы для защиты работы

- 1) С какими типами переменных может работать нейронная сеть?
- 2) Как осуществляется предсказание значений нейронной сетью?
- 3) В чем заключаются аппроксимирующие и интерполирующие свойства нейронных сетей?
- 4) Как осуществлять постобработку и для чего она нужна?
- 5) Нейроны с какими функциями активации можно ставить на выходной слой?
- 6) Чем нейронная сеть отличается от линейной регрессионной модели?
- 7) Что такое переобучение сети и как с ним бороться?

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат:	12000002A633E3D113AD425FB50002000002A6
Владелец:	Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022	

Приложение 1

```
function [alpha, b, w, evals, stp, glob] = SMO2(X, Y, kernel, kpar1, kpar2, C, tol, steps, eps, method)
```

```
dbstop if error
dbg = 2;
```

```
if (nargin < 10)
    method = 1;
end
```

```
if (nargin < 9)
    eps = 0.0001;
end
```

```
if (nargin < 8)
    steps = 10000;
end
```

```
if (nargin < 7)
    tol = 0.001;
end
```

```
if (nargin < 6)
    C = inf;
end
```

```
if (nargin < 5)
    kpar2 = 0;
end
```

```
if (nargin < 4)
    kpar1 = 0;
end
```

```
if (nargin < 3)
    kernel = 0;
end
```

```
if (nargin < 2)
    error('Error: At least two arguments (training points and class values) must be supplied');
else
    [n, D] = size(X);
    [n1, D1] = size(Y);
end
```

```
if (1 ~= D1)
    error('Error: Class values cannot be vectors but real numbers');
end
```

```
if (n ~= n1)
    error('Error: Number of rows of X and Y must be the same (one class value for each sample)');
end
```

```
if (method == 1)
    [alpha, b, w, evals, stp, glob] = SMO_Keerthi_modif1(X, Y, kernel, kpar1, kpar2, C, tol, steps, eps);
elseif (method == 2)
    [alpha, b, w, evals, stp, glob] = SMO_Keerthi_modif2(X, Y, kernel, kpar1, kpar2, C, tol, steps, eps);
else
    [alpha, b, w, evals, stp, glob] = SMO_Platt(X, Y, kernel, kpar1, kpar2, C, tol, steps, eps);
end
```

```
if (method == 1) || (method == 2)
    flag = (glob.b_up < glob.b_low - 2 * tol) || (stp >= steps);
else
    flag = (stp >= steps);
end
```

```
if (flag == 1)
```

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

```
function [alpha, b, w, evals, stp, glob] = SMO_Platt(X, Y, kernel, kpar1, kpar2, C, tol, steps, eps)
```

```

[n, D] = size(X);
%initialize alpha array to all zero
alpha = zeros(n,1);
w = zeros(1,D);
b = 0;
evals = 0;

for i=1:n
    K(:,i)=CalcKernel(X,X(i,:), kernel, kpar1, kpar2);
end

%initialize struct for temporary variables that must be global
glob = struct('ecache',[],'v_1',[],'v_2',[],'I_0',[],'ecache_f',[]);
%%initialize fcache array to all zero and its size to n
glob.ecache = zeros(n,1);
glob.ecache_f = zeros(n,1); %0->ecache value not-OK, 1->value OK
glob.v_1 = find(Y==1);
glob.v_2 = find(Y==1);

stp = 0;
numChanged = 0;
examineAll = 1;
while ((numChanged > 0 || examineAll == 1) && stp <= steps)
    numChanged = 0;
    if (examineAll == 1)
        for i = 1 : n
            stp = stp + 1;
            if (stp > steps) break; end
            [retval, alpha, w, b, stp, evals, glob] =...
                examineExampleP(i, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, steps, stp, evals, eps,K);
            numChanged = numChanged + retval;
        end
    else
        glob.I_0 = find(alpha > 0 & alpha < C);
        k = length(glob.I_0);
        for i = 1 : k
            stp = stp + 1;
            if (stp > steps) break; end
            if (i > length(glob.I_0)) break; end %glob.I_0 changes inside loop (in examineExampleP)
            [retval, alpha, w, b, stp, evals, glob] =...
                examineExampleP(glob.I_0(i), glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, steps, stp, evals, eps,K);
            numChanged = numChanged + retval;
        end
    end
    end
    if (examineAll == 1)
        examineAll = 0;
    elseif (numChanged == 0)
        examineAll = 1;
    end
end
end

```

```

function [retval, alpha, w, b, evals, glob] =...
    takeStepP(i1, i2, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, evals, eps,K)

```

```

%drawnow; %%used to give Matlab the opportunity to examine any pending ctrl+C (while in deep loops)
if (get(0,'PointerLocation')==[1 1])
    %disp('Press <ctrl+c> to stop or any key to interrupt execution temporarily ...');
    %pause;
    %disp('Type "return" to carry on');
    keyboard;
end
end

```

```

[n D] = size(X);
if (i1 == i2)
    retval = 0;
    return;
end
alph1 = alpha(i1);
y1 = Y(i1);
alph2 = alpha(i2);

```

```

y2 = Y(i2);
s = y1 * y2;
%Compare s to 0
if (y1 == y2)
    L = max([0, alph2 - alph1]);
    H = min([C, C + alph2 - alph1]);
end
%Drawnow

```

Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен с: 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ


```

    L = max([0, alph1 + alph2 - C]);
    H = min([C, alph1 + alph2]);
end
if (L == H)
    retval = 0;
    return;
end
%calculate E1 = SVM output in X[i1] - y1 (check in error cache)
if (glob.ecache_f(i1) == 0)
    k11 = K(:,i1);
    evals = evals + n;
    E1 = -y1 + (k11' * (Y .* alpha)) - b;
    glob.ecache(i1) = E1;
    glob.ecache_f(i1) = 1;
else
    E1 = glob.ecache(i1);
end
%calculate E2 = SVM output in X[i2] - y2 (check in error cache)
if (glob.ecache_f(i2) == 0)
    k12 = K(:,i2);
    evals = evals + n;
    E2 = -y2 + (k12' * (Y .* alpha)) - b;
    glob.ecache(i2) = E2;
    glob.ecache_f(i2) = 1;
else
    E2 = glob.ecache(i2);
end
%%computation of the derivative eta
k11 = K(i1,i1);
k12 = K(i2,i1);
k22 = K(i2,i2);
evals = evals + 3;
eta = -(2 * k12) + k11 + k22;
%%computation of new alpha(i2)
if (eta > 0)
    a2 = alph2 + (y2 * (E1 - E2) / eta);
    if (a2 < L)
        a2 = L;
    elseif (a2 > H)
        a2 = H;
    end
else %%the derivative is 0 => we have to make optimization by other means
    %%Lobj = objective function at a2=L (according to Platt)
    %%Hobj = objective function at a2=H (according to Platt)
    L1 = alph1 + (s * (alph2 - L));
    H1 = alph1 + (s * (alph2 - H));
    f1 = y1 * (E1 + b) - (alph1 * k11) - (s * alph2 * k12);
    f2 = y2 * (E2 + b) - (alph2 * k22) - (s * alph1 * k12);
    Lobj = (L1 * f1) + (L * f2) + (0.5 * k11 * L^2) + (0.5 * k22 * L^2) + (s * k12 * L * L1);
    Hobj = (H1 * f1) + (H * f2) + (0.5 * k11 * H^2) + (0.5 * k22 * H^2) + (s * k12 * H * H1);
    if (Lobj < Hobj - eps)
        a2 = L;
    elseif (Lobj > Hobj + eps)
        a2 = H;
    else
        a2 = alph2;
    end
end
end
if (abs(a2 - alph2) < (eps * (a2 + alph2 + eps)))
    retval = 0;
    return
end
% computation on new alpha1(a1)
a1 = alph1 + (s * (alph2 - a2));
%Update threshold to reflect change in Lagrange multipliers
b_old = b;
if (a1 > L && a1 < H)
    b = E1 + (y1 * (a1 - alph1) * k11) + (y2 * (a2 - alph2) * k12) + b;
elseif (a2 > L && a2 < H)
    b = E2 + (y1 * (a1 - alph1) * k12) + (y2 * (a2 - alph2) * k22) + b;
else
    b1 = E1 + (y1 * (a1 - alph1) * k11) + (y2 * (a2 - alph2) * k12) + b;
    b2 = E2 + (y2 * (a2 - alph2) * k22) + b;
    b = (b1 + b2) / 2;
end
%Update weight vector to reflect change in a1 & a2, if linear SVM
if (strcmp(kernel, 'linear') == 1)
    w = w + (y1 * (a1 - alph1) * X(i1,:)) + (y2 * (a2 - alph2) * X(i2,:));
end

```

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

```

%Update ecache[i] using new Lagrange multipliers
v = find(glob.ecache_f==1);
for i = 1 : length(v)
    ki1 = K(v(i),i1);
    ki2 = K(v(i),i2);
    evals = evals + 2;
    %glob.ecache(v(i)) = glob.ecache(v(i)) + b - b_old - (y1 * (a1 - alph1) * ki1) - (y2 * (a2 - alph2) * ki2);
    glob.ecache(v(i)) = glob.ecache(v(i)) + b_old - b + (y1 * (a1 - alph1) * ki1) + (y2 * (a2 - alph2) * ki2);
end
%Store a1 and a2 in the alpha array
alpha(i1) = a1;
alpha(i2) = a2;

%%Compute updated E values for i1 and i2
%glob.ecache(i1) = E1 + b - b_old - (y1 * (a1 - alph1) * k11) - (y2 * (a2 - alph2) * k12);
glob.ecache(i1) = E1 + b_old - b + (y1 * (a1 - alph1) * k11) + (y2 * (a2 - alph2) * k12);
glob.ecache_f(i1) = 1;
%glob.ecache(i2) = E2 + b - b_old - (y1 * (a1 - alph1) * k12) - (y2 * (a2 - alph2) * k22);
glob.ecache(i2) = E2 + b_old - b + (y1 * (a1 - alph1) * k12) + (y2 * (a2 - alph2) * k22);
glob.ecache_f(i2) = 1;
% Update I_0
glob.I_0 = find(alpha > 0 & alpha < C);

retval = 1;
return

function [retval, alpha, w, b, stp, evals, glob] =...
    examineExampleP(i2, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, steps, stp, evals, eps,K)

%drawnow; %%used to give Matlab the opportunity to examine any pending ctrl+C (while in deep loops)
if (get(0,'PointerLocation')==[1 1])
    %disp('Press <ctrl+c> to stop or any key to interrupt execution temporarily ...');
    %pause;
    %disp('Type "return" to carry on');
    keyboard;
end

retval = 0;
[n D] = size(X);
y2 = Y(i2);
alph2 = alpha(i2);
if (glob.ecache_f(i2) == 1)
    E2 = glob.ecache(i2);
else
    ki2 = K(:,i2);
    evals = evals + n;
    E2 = -y2 + (ki2' * (Y .* alpha)) - b;
    glob.ecache(i2) = E2;
    glob.ecache_f(i2) = 1;
end
r2 = E2 * y2;
if ((r2 < -tol && alph2 < C) || (r2 > tol && alph2 > 0))
    if (length(glob.I_0) > 1)
        %i1 = result of second choice heuristic
        v = find(glob.ecache_f==1);
        k = length(v);
        Emax = 0;
        for i = 1 : k
            tmp = abs(glob.ecache(v(i)) - E2);
            if (tmp > Emax)
                Emax = tmp;
                i1 = v(i);
            end
        end
        stp = stp + 1;
        [retval, alpha, w, b, evals, glob] =...
            takeStepP(i1, i2, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, evals, eps,K);
        if (retval == 1)
            return;
        end
    end
end

```

%loop over i1, starting at a random point

k = length(v);

randi(k,1,1);

r = floor(k * rand);

for i = 1 : k

i1 = mod(r + i, k) + 1;

stp = stp + 1;

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен с 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ


```

        [retval, alpha, w, b, evals, glob] =...
            takeStepP(i1, i2, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, evals, eps,K);
        if (retval == 1)
            return;
        end
    end
end
%loop over all possible i1, starting at a random point
k = n;
r = floor(k * rand);
for i = 1 : k
    i1 = mod(r + i, k) + 1;
    stp = stp + 1;
    [retval, alpha, w, b, evals, glob] =...
        takeStepP(i1, i2, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, evals, eps,K);
    if (retval == 1)
        return;
    end
end
end
return

function [alpha, b, w, evals, stp, glob] = SMO_Keerthi_modif1(X, Y, kernel, kpar1, kpar2, C, tol, steps, eps)
[n, D] = size(X);
%initialize alpha array to all zero
%-----
alpha = zeros(n,1);
w = zeros(1,D);
b = 0;
evals = 0;
%-----

for i=1:n
    K(:,i)=CalcKernel(X,X(i,:), kernel, kpar1, kpar2);
end

%initialize struct for temporary variables that must be global
%-----

glob = struct('fcache',[],'b_up',0,'b_low',0,'i_up',0,'i_low',0,'v_1',[],'v_2',[],...
    'I_0',[],'I_1',[],'I_2',[],'I_3',[],'I_4',[]);
%%initialize fcache array to all zero and its size to n
glob.fcache = zeros(n,1);
%initialize b_up = -1, i_up to any one index of class 1
glob.b_up = -1;
glob.v_1 = find( Y==1 );
glob.i_up = glob.v_1(1);
%initialize b_low = 1, i_low to any one index of class 2
glob.b_low = 1;
glob.v_2 = find( Y== -1 );
glob.i_low = glob.v_2(1);
%set fcache[i_low] = 1 and fcache[i_up] = -1
glob.fcache(glob.i_low) = 1;
glob.fcache(glob.i_up) = -1;

%Initialize the I_* sets
glob.I_0 = find(alpha > 0 & alpha < C);
glob.I_1 = find(alpha(glob.v_1) == 0);
glob.I_1 = glob.v_1(glob.I_1);
glob.I_2 = find(alpha(glob.v_2) == C);
glob.I_2 = glob.v_2(glob.I_2);
glob.I_3 = find(alpha(glob.v_1) == C);
glob.I_3 = glob.v_1(glob.I_3);
glob.I_4 = find(alpha(glob.v_2) == 0);
glob.I_4 = glob.v_2(glob.I_4);
%-----

stp = 0;
numChanged = 0;
examineAll = 1;
while ( ((numChanged > 0) || (examineAll==1))&&(stp <= steps))
    numChanged = 0;
    if (examineAll==1)
        for i=1:n
            stp = stp + 1;
            if (stp>steps) break;
        end
        [retval, alpha, w, b, stp, evals,glob] =...
            examineExample(X, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, steps, stp, evals, eps,K);
    end
end

```

Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ

```

        numChanged = numChanged + retval;
    end
else
    k = length(glob.I_0);
    for i = 1 : k
        stp = stp + 1;
        if (stp > steps) break; end
        if (i > length(glob.I_0)) break; end %glob.I_0 changes inside loop (in examineExampleK)
        [retval, alpha, w, b, stp, evals, glob] = ...
            examineExampleK(glob.I_0(i), glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, steps, stp, evals, eps, K);
        numChanged = numChanged + retval;
        %it is easy to check if optimality on I_0 is attained...
        if ( (glob.b_up) > ( glob.b_low - (2*tol) ) )
            %exit the loop after setting numChanged = 0
            numChanged = 0;
            break;    %EDW TA PRAGMATA ALLAZOUN AN XRHSIMOPOIHSOUME THN RETURN. SYSKEKRIMENA
MERIKES FORES ENW EXEI SYMBEI
            %TO b_up>b_low POY EINAI TO ZHTOUMENO, ME THN BREAK BGAINOUME MONO APO TO ESWTERIKO
FOR (TO PSAKSIMO STO I_0)
            %ENW ISWS (DEN EIMAI SIGOUROS) THA EPREPE NA TERMATIZEI O ALGORITHMOS. ME THN BREAK TO
PROGRAMMA SYNEXIZEI
            %KAI ALLAZOUN TA b_up KAI b_low KAI SXHMATIKA FAINETAI KALYTEROS O TAKSIMOMHTHS.
            %ME THN RETURN OMWS EINAI PIO SYNTOMOS KAI PALI FAINETAI SWSTOS. NA TO PSAKSOYME.
        end
    end
end
end
if (examineAll == 1)
    examineAll = 0;
elseif (numChanged == 0)
    examineAll = 1;
end
b=(glob.b_up+glob.b_low)/2;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function [alpha, b, w, evals, stp, glob] = SMO_Keerthi_modif2(X, Y, kernel, kpar1, kpar2, C, tol, steps, eps)
[n, D] = size(X);
%initialize alpha array to all zero
%-----
alpha = zeros(n,1);
% if (strcmpi(kernel, 'linear') == 1)
w = zeros(1,D);
% else
%   w = [];
% end
b = 0;
evals = 0;
%-----

```

```

for i=1:n
    K(:,i)=CalcKernel(X,X(i,:), kernel, kpar1, kpar2);
end
%initialize struct for temporary variables that must be global
%-----

glob = struct('fcache',[], 'b_up',0, 'b_low',0, 'i_up',0, 'i_low',0, 'v_1',[], 'v_2',[], ...
    'I_0',[], 'I_1',[], 'I_2',[], 'I_3',[], 'I_4',[]);
%%initialize fcache array to all zero and its size to n
glob.fcache = zeros(n,1);
%initialize b_up = -1, i_up to any one index of class 1
glob.b_up = -1;
glob.v_1 = find( Y==1 );
glob.i_up = glob.v_1(1);
%initialize b_low = 1, i_low to any one index of class 2
glob.b_low = 1;

```

```

glob.v_2 = find(
glob.i_low =
%set fcache f_low
glob.fcache(glob.i_low) = 1;
glob.fcache(glob.i_up) = -1;

```

Сертификат: 12000002A633E3D113AD425FB50602000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН

ЭЛЕКТРОННОЙ ПОДПИСЬЮ


```

stp = 0;
numChanged = 0;
examineAll = 1;
while ((numChanged > 0 || examineAll == 1) && stp <= steps)
    numChanged = 0;
    if (examineAll == 1)
        for i = 1 : n
            stp = stp + 1;
            if (stp > steps) break; end
            [retval, alpha, w, b, stp, evals, glob] =...
                examineExampleK(i, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, steps, stp, evals, eps,K);
            numChanged = numChanged + retval;
        end
    else
        %the following loop is the only difference between the two SMO
        %modifications. Whereas, in modification 1, the inner loop selects
        %i2 from I_0 sequentially, here i2 is always set to the current
        %i_low and i1 is set to the current i_up; clearly, this corresponds
        %to choosing the worst violating pair using members of I_0 and some
        %other indices.
        inner_loop_success = 1;
        while ( ((glob.b_up)<(glob.b_low-(2*tol)))&&(inner_loop_success~=0))
            i2 = glob.i_low;
            y2 = Y(i2);
            alph2 = alpha(i2);
            F2 = glob.fcache(i2);
            i1 = glob.i_up;
            stp = stp + 1;
            if (stp > steps) break; end
            stp = stp + 1;
            [inner_loop_success, alpha, w, b, evals, glob] =...
                takeStepK(glob.i_up, glob.i_low, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, evals, eps,K);
            numChanged = numChanged + inner_loop_success;
        end
        num_changed = 0;
    end
    if (examineAll == 1)
        examineAll = 0;
    elseif (numChanged == 0)
        examineAll = 1;
    end
end
b=(glob.b_up+glob.b_low)/2;
return

```

```
function [retval, alpha, w, b, evals, glob] =...
    takeStepK(i1, i2, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, evals, eps, K)
%Much of this procedure is same as that in Platt's SMO pseudo-code.
```

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
12000002A633E3D113AD425FB50002000002A6
Шебзухова Татьяна Александровна
с 20.08.2021 по 20.08.2022

```

y1 = Y(i1);
F1 = glob.fcache(i1);
alph2 = alpha(i2);
y2 = Y(i2);
F2 = glob.fcache(i2);
s = y1 * y2;

% Compute L, H - If L=H return 0
%-----
if (y1 ~= y2)
    L = max([0, alph2 - alph1]);
    H = min([C, C + alph2 - alph1]);
else % y1 == y2
    L = max([0, alph1 + alph2 - C]);
    H = min([C, alph1 + alph2]);
end
if (L == H)
    retval = 0;
    return;
end
%-----

%% computation of the derivative eta
%-----
k11 = K(i1,i1);
k12 = K(i1,i2);
k22 = K(i2,i2);
evals = evals + 3;
eta = (2*k12) - k11 - k22;
%% computation of new alpha(i2)
if (eta<0)
    a2=alph2-(y2*(F1-F2)/eta); %% HERE is different from Platt
    if (a2 < L)
        a2 = L;
    elseif (a2 > H)
        a2 = H;
    end
else %% the derivative is 0 => we have to make optimization by other means
    %% Lobj = objective function at a2=L (according to Platt)
    %% Hobj = objective function at a2=H (according to Platt)
    L1 = alph1 + (s*(alph2-L));
    H1 = alph1 + (s*(alph2-H));
    f1 = (-y1 * F1) + (alph1 * k11) + (s * alph2 * k12);
    f2 = (-y2 * F2) + (alph2 * k22) + (s * alph1 * k12);
    Lobj = (L1 * f1) + (L * f2) - (0.5 * k11 * L^2) - (0.5 * k22 * L^2) - (s * k12 * L * L1);
    Hobj = (H1 * f1) + (H * f2) - (0.5 * k11 * H^2) - (0.5 * k22 * H^2) - (s * k12 * H * H1);
    if (Lobj > Hobj + eps)
        a2 = L;
    elseif (Lobj < Hobj - eps)
        a2 = H;
    else
        a2 = alph2;
    end
end
end
%-----

% Calculate the change in a - if very small return 0
%% NOTE!! if eps not small enough it may stop the algorithm early!!!!
%-----
if (abs(a2-alph2)<eps*(a2+alph2+eps))
    retval=0;
    return
end
%-----

% computation on new alpha1(a1)
%-----
a1 = alph1 + (s*(alph2-a2));
%-----

```

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

```

% Update weight vector to reflect change in a1 & a2, if linear SVM
if (strcmpi(kernel, 'linear') == 1)
    new_y1 = (a1 * alph1) * X(i1,:)) + (y2 * (a2 - alph2) * X(i2,:));

```



```

end
%-----

%Store a1 and a2 in the alpha array
%-----
alpha(i1) = a1;
alpha(i2) = a2;
%-----

%Update fcache[i] for i in I_0 using new Lagrange multipliers
%-----
k = length(glob.I_0);
for i = 1 : k
    ki1 = K(glob.I_0(i),i1);
    ki2 = K(glob.I_0(i),i2);
    evals = evals + 2;
    glob.fcache(glob.I_0(i)) = glob.fcache(glob.I_0(i)) + (y1 * (a1 - alph1) * ki1) + (y2 * (a2 - alph2) * ki2);
end
%-----

%The update below is simply achieved by keeping and updating information
%about alpha_i being at 0, C or in between them. Using this together with
%target[i] gives information as to which index set i belongs.
% Update I_0, I_1, I_2, I_3, I_4
%-----
glob.I_0 = find(alpha > 0 & alpha < C);
glob.I_1 = find(alpha(glob.v_1) == 0);
glob.I_1 = glob.v_1(glob.I_1);
glob.I_2 = find(alpha(glob.v_2) == C);
glob.I_2 = glob.v_2(glob.I_2);
glob.I_3 = find(alpha(glob.v_1) == C);
glob.I_3 = glob.v_1(glob.I_3);
glob.I_4 = find(alpha(glob.v_2) == 0);
glob.I_4 = glob.v_2(glob.I_4);
%-----

%Compute updated F values for i1 and i2
%-----
glob.fcache(i1) = F1 + (y1 * (a1 - alph1) * k11) + (y2 * (a2 - alph2) * k12);
glob.fcache(i2) = F2 + (y1 * (a1 - alph1) * k12) + (y2 * (a2 - alph2) * k22);
%-----

%Compute (i_low, b_low) and (i_up, b_up),
%using only i1, i2 and indices in I_0

%-----
%--GIA TO i1 -----
v = find(glob.I_1==i1);
i1_in_I_1 = length(v);
v = find(glob.I_2==i1);
i1_in_I_2 = length(v);
v = find(glob.I_3==i1);
i1_in_I_3 = length(v);
v = find(glob.I_4==i1);
i1_in_I_4 = length(v);

% %-----
%--GIA TO i2 -----
v = find(glob.I_1==i2);
i2_in_I_1 = length(v);
v = find(glob.I_2==i2);
i2_in_I_2 = length(v);
v = find(glob.I_3==i2);
i2_in_I_3 = length(v);
v = find(glob.I_4==i2);
i2_in_I_4 = length(v);

```

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

if size(glob.I_0)==0 % Trying to run the smo mod1 for the adult datasets I discovered that for small values of C there was this problem!

ителен: с 20.08.2021 г.

```

i2_in_I_1 = length(v);
v = find(glob.I_2==i2);
i2_in_I_2 = length(v);
v = find(glob.I_3==i2);
i2_in_I_3 = length(v);
v = find(glob.I_4==i2);
i2_in_I_4 = length(v);

if ((i2_in_I_1 + i2_in_I_2 > 0) && (F2 < glob.b_up))
    glob.b_up = F2;
    glob.i_up = i2;
elseif ((i2_in_I_3 + i2_in_I_4 > 0) && (F2 > glob.b_low))
    glob.b_low = F2;
    glob.i_low = i2;
end
%-----

%Chech optimality using current b_low and b_up and, if
%violated, find an index i1 to do joint optimization with i2
%-----
optimality = 1;
if ((i2_in_I_0 + i2_in_I_1 + i2_in_I_2) > 0)
    if ((glob.b_low - F2) > (2 * tol) )
        optimality = 0;
        i1 = glob.i_low;
    end
end
if ((i2_in_I_0 + i2_in_I_3 + i2_in_I_4) > 0)
    if ((F2 - glob.b_up) > (2 * tol))
        optimality = 0;
        i1 = glob.i_up;
    end
end
if (optimality == 1)
    retval = 0;
    return;
end
%-----

%For i2 in I_0 choose the better i1
%-----

if (i2_in_I_0 > 0)
    if ((glob.b_low - F2) > (F2 - glob.b_up))
        i1 = glob.i_low;
    else
        i1 = glob.i_up;
    end
end
%-----

stp = stp + 1;

[retval, alpha, w, b, evals, glob] =...
    takeStepK(i1, i2, glob, alpha, w, b, X, Y, kernel, kpar1, kpar2, C, tol, evals, eps,K);

```

Функция svcplot_book

```
function svcplot_book(X,Y,ker,kpar1,kpar2,alpha,bias,aspect,mag,xaxis,yaxis,input)
```

global figt4

color_shade = 1;

gridcellsX = 50;

gridcellsY = 50;

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ


```

help svcp lot
else
    epsilon = 1e-5;
    if (margin < 12) input = zeros(1,size(X,2));, end
    if (margin < 11) yaxis = 2;; end
    if (margin < 10) xaxis = 1;; end
    if (margin < 9) mag = 0.1;; end
    if (margin < 8) aspect = 0;; end

    xmin = min(X(:,xaxis));, xmax = max(X(:,xaxis));
    ymin = min(X(:,yaxis));, ymax = max(X(:,yaxis));
    xa = (xmax - xmin);, ya = (ymax - ymin);
    if (~aspect)
        if (0.75*abs(xa) < abs(ya))
            offadd = marg*(ya*4/3 - xa);,
            xmin = xmin - offadd - mag*marg*ya;, xmax = xmax + offadd + mag*marg*ya;
            ymin = ymin - mag*marg*ya;, ymax = ymax + mag*marg*ya;
        else
            offadd = marg*(xa*3/4 - ya);,
            xmin = xmin - mag*marg*xa;, xmax = xmax + mag*marg*xa;
            ymin = ymin - offadd - mag*marg*xa;, ymax = ymax + offadd + mag*marg*xa;
        end
    else
        xmin = xmin - mag*marg*xa;, xmax = xmax + mag*marg*xa;
        ymin = ymin - mag*marg*ya;, ymax = ymax + mag*marg*ya;
    end

    alpha_min=min(alpha);
    alpha_max=max(alpha);
    alpha_threshold = (alpha_max - alpha_min) * 0.01;
    alpha_threshold = alpha_threshold + alpha_min;

    [x,y] = meshgrid(xmin: (xmax-xmin)/gridcellsX:xmax,ymin: (ymax-ymin)/gridcellsY:ymax);
    z = bias*ones(size(x));
    wh = waitbar(0,'Plotting...');
    for x1 = 1 : size(x,1)
        for y1 = 1 : size(x,2)
            input(xaxis) = x(x1,y1);, input(yaxis) = y(x1,y1);
            for i = 1 : length(Y)
                if (abs(alpha(i)) >= 0)
                    z(x1,y1) = z(x1,y1) + Y(i)*alpha(i)*CalcKernel(input,X(i,:),ker,kpar1,kpar2);
                end
            end
        end
    end

    waitbar((x1)/size(x,1)) ;
    drawnow
end

```

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен с 20.08.2021 по 20.08.2022


```

fh = gcf;

set(gca,'XLim',[xmin xmax],'YLim',[ymin ymax]);
set(gca,'TickDir','in');
set(gca,'XTick',[floor(xmin):ceil(xmax)]);
%      set(gca,'XTickLabel', []); %Null list => Does not print Tick labels
set(gca,'YTick',[floor(ymin):ceil(ymax)]);
%      set(gca,'YTickLabel', []); %Null list => Does not print Tick labels
set(gca,'Box','on');
set(gca,'DataAspectRatio',[1 1 1]);
%eliminate borders of figure
old_gca_units = get(gca,'Units');
set(gca,'Units','Normalized');
set(gca,'Position',...
    [0.0 0.0 1.0 1.0]);
set(gca,'Units',old_gca_units);

l = (-min(min(z)) + max(max(z)))/2.0;
if (color_shade == 1)
    sp = pcolor(x,y,z);
    shading interp %has bug and does not interpolate correctly last column
    %shading flat
    set(sp,'LineStyle','none');
    set(gca,'Clim',[-1, 1])
    set(gca,'Position',[0 0 1 1])
    %      axis off
    load cmap
    colormap(cmap)
    %colormap(gray)
else
    whitebg('w')
end
hold on
for i = 1:size(Y)
    if (Y(i) == 1)
        if (color_shade == 1)
            plot(X(i,xaxis),X(i,yaxis),'rx','LineWidth',2) % Class A
        else
            plot(X(i,xaxis),X(i,yaxis),'x','LineWidth',1, 'MarkerSize', 4, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'none') % Class A
        end
    else
        if (color_shade == 1)
            plot(X(i,xaxis),X(i,yaxis),'bx','LineWidth',2) % Class B
        else
            %plot(X(i,xaxis),X(i,yaxis),'*', 'LineWidth',1, 'MarkerSize', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k') % Class B
            plot(X(i,xaxis),X(i,yaxis),'x','LineWidth',1, 'MarkerSize', 3, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'none') % Class B
        end
    end
end
end

```

Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 6) % Support Vector

```

        plot(X(i,xaxis),X(i,yaxis),'x','LineWidth',1, 'MarkerSize', 5, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'none') % Support
Vector
    end
end
end
% Plot Boundary contour

hold on
if (color_shade == 1)
    contour(x,y,z,[0 0],'k')
    %    contour(x,y,z,[-0.5 -0.5],'b')
    contour(x,y,z,[-1 -1],'b-')
    contour(x,y,z,[1 1],'r-')
    %    contour(x,y,z,[0.5 0.5],'b')
else
    zones = 1; %how many zones to be present in [0,1]
    steps = [0 : 1/zones : 1];
    for j = 1 : zones + 1
        if ((mod(j,2))==1),
            clsp = 'k-';
        elseif ((mod(j,2))==0),
            clsp = 'k: ';
        end
        if (steps(j) == 0)
            contour(x,y,z,[steps(j) steps(j)],clsp, 'LineWidth', 2)
        else
            contour(x,y,z,[steps(j) steps(j)],clsp, 'LineWidth', 1)
            contour(x,y,z,[-steps(j) -steps(j)],clsp, 'LineWidth', 1)
        end
    end
end
end
hold off
end

```

Функция CalcKernel

```
function k = CalcKernel(u, v, ker, kpar1, kpar2)
```

```

if (nargin < 3)
    error('CalcKernel needs at least 3 arguments')
end
if (nargin < 5)
    kpar2 = 0;
end
if (nargin < 4)
    kpar1 = 0;
end

```

[r1 c1] = size(u);
 [r2 c2] = size(v);
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 error('CalcKernel expect u=column of row-vectors and v a row-vector')


```

if (c1 ~= c2)
    error('CalcKernel needs both x1 and x2 to have same num of columns')
end

switch lower(ker)
case 'linear'
    k = u*v';
case 'poly'
    k = (u*v' + kpar1).^kpar2;
case 'rbf'
    k = zeros(r1,1);
    for i = 1 : r1
        k(i) = exp(-(u(i,:)-v)*(u(i,:)-v)'/(2*kpar1^2));
    end
case 'erbf'
    k = zeros(r1,1);
    for i = 1 : r1
        k(i) = exp(-sqrt((u(i,:)-v)*(u(i,:)-v)'/(2*kpar1^2)));
    end
case 'sigmoid'
    k = zeros(r1,1);
    for i = 1 : r1
        k(i) = tanh(kpar1*u(i,:)*v'/length(u(i,:)) + kpar2);
    end
case 'fourier'
    k = zeros(r1,1);
    for i = 1 : r1
        z = sin(kpar1 + 1/2)*2*ones(length(u(i,:)),1);
        j = find(u(i,:)-v);
        z(j) = sin(kpar1 + 1/2)*(u(i,j)-v(j))./sin((u(i,j)-v(j))/2);
        k(i) = prod(z);
    end
case 'spline'
    k = zeros(r1,1);
    for i = 1 : r1
        z = 1 + u(i,:).*v + u(i,:).*v.*min(u(i,:),v) - ((u(i,:)+v)/2).*(min(u(i,:),v)).^2 + (1/3)*(min(u(i,:),v)).^3;
        k(i) = prod(z);
    end
case {'curvspline', 'anova'}
    k = zeros(r1,1);
    for i = 1 : r1
        z = 1 + u(i,:).*v + (1/2)*u(i,:).*v.*min(u(i,:),v) - (1/6)*(min(u(i,:),v)).^3;
        k(i) = prod(z);
    end

    % - sum(u.*v) - 1;
    %     z = 1 + u.*v + (1/2)*u.*v.*min(u,v) - (1/6)*(min(u,v)).^3;
    %     k = prod(z);
    %     z = 1 + u.*v + (1/2)*u.*v.*min(u,v) - (1/6)*(min(u,v)).^3;
    %     k = prod(z);

```

ДОКУМЕНТ ПОДПИСАН
 ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022


```

for i = 1 : r1
    z = 0;
    for r = 0: 2*(kpar1+1)
        z = z + (-1)^r*binomial(2*(kpar1+1),r)*(max(0,u(i,:)-v + kpar1+1 - r)).^(2*kpar1 + 1);
    end
    k(i) = prod(z);
end
case 'anovaspline1'
    k = zeros(r1,1);
    for i = 1 : r1
        z = 1 + u(i,:).*v + u(i,:).*v.*min(u(i,:),v) - ((u(i,:)+v)/2).*(min(u(i,:),v)).^2 + (1/3)*(min(u(i,:),v)).^3;
        k(i) = prod(z);
    end
case 'anovaspline2'
    k = zeros(r1,1);
    for i = 1 : r1
        z = 1 + u(i,:).*v + (u(i,:).*v).^2 + (u(i,:).*v).^2.*min(u(i,:),v) - u(i,:).*v.*(u(i,:)+v).*(min(u(i,:),v)).^2 + (1/3)*(u(i,:).^2 +
4*u(i,:).*v + v.^2).*(min(u(i,:),v)).^3 - (1/2)*(u(i,:)+v).*(min(u(i,:),v)).^4 + (1/5)*(min(u(i,:),v)).^5;
        k(i) = prod(z);
    end
case 'anovaspline3'
    k = zeros(r1,1);
    for i = 1 : r1
        z = 1 + u(i,:).*v + (u(i,:).*v).^2 + (u(i,:).*v).^3 + (u(i,:).*v).^3.*min(u(i,:),v) - (3/2)*(u(i,:).*v).^2.*(u(i,:)+v).*(min(u(i,:),v)).^2 +
u(i,:).*v.*(u(i,:).^2 + 3*u(i,:).*v + v.^2).*(min(u(i,:),v)).^3 - (1/4)*(u(i,:).^3 + 9*u(i,:).^2.*v + 9*u(i,:).*v.^2 + v.^3).*(min(u(i,:),v)).^4 +
(3/5)*(u(i,:).^2 + 3*u(i,:).*v + v.^2).*(min(u(i,:),v)).^5 - (1/2)*(u(i,:)+v).*(min(u(i,:),v)).^6 + (1/7)*(min(u(i,:),v)).^7;
        k(i) = prod(z);
    end
case 'anovabspline'
    k = zeros(r1,1);
    for i = 1 : r1
        z = 0;
        for r = 0: 2*(kpar1+1)
            z = z + (-1)^r*binomial(2*(kpar1+1),r)*(max(0,u(i,:)-v + kpar1+1 - r)).^(2*kpar1 + 1);
        end
        k(i) = prod(1 + z);
    end
otherwise
    %k = u*v'; % linear (identity kernel)
    fprintf('CalcKernel: wrong kernel ''%s''\n',ker);
end

```

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

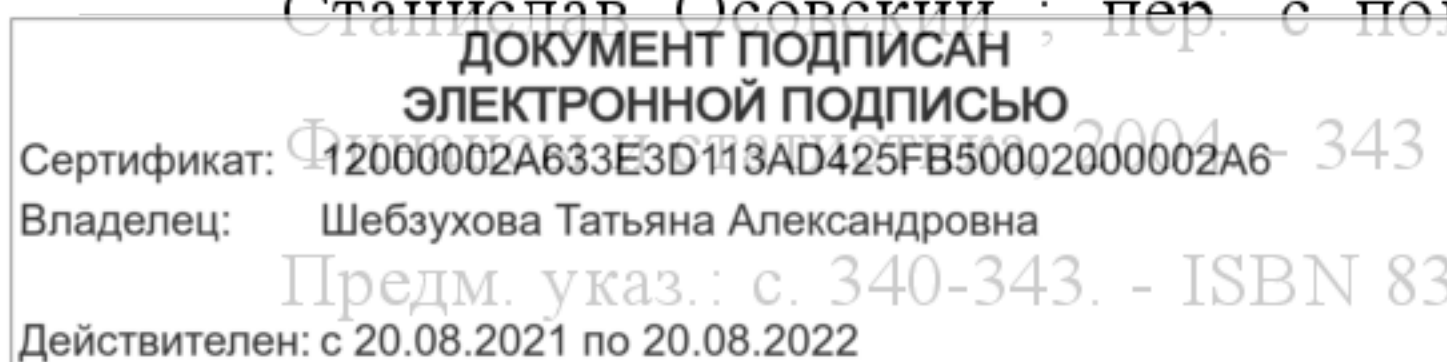
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основная литература

1. Курейчик, В. В. Теория эволюционных вычислений / Курейчик В. В. - Москва : ФИЗМАТЛИТ, 2012. - 260 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9221-1390-8
2. Курейчик, В. М. Основы теории эволюционных вычислений : Научная монография / Курейчик В. М. - Ростов-на-Дону : Южный федеральный университет, 2010. - 224 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9275-0799-3
3. Курейчик, В. М. Поисковая адаптация. Теория и практика / Курейчик В. М. - Москва : ФИЗМАТЛИТ, 2006. - 270 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9221-0749-6

Дополнительная литература

1. Методические указания по организации самостоятельной работы по дисциплине «Нечёткая логика и нейронные сети» : для студентов направления 38.03.05 «Бизнес-информатика» (профиль «Электронный бизнес», «Информационная бизнес-аналитика»). Учебный план 2012 г. / сост. И. Ю. Глазкова ; ФГАОУ ВПО Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 20 с.
2. Барский, А. Б. Нейронные сети: распознавание, управление, принятие решений : [учеб. пособие] / А.Б. Барский. - М. : Финансы и статистика, 2004. - 176 с. : ил. - (Прикладные информационные технологии). - Библиогр.: с. 170-173. - ISBN 5-279-02757-X
3. Осовский, С. Нейронные сети для обработки информации / Станислав Осовский ; пер. с польск. И. Д. Рудинского. - М. : 343 с. : ил. - Библиогр.: с. 3330-339. - Предм. указ.: с. 340-343. - ISBN 83-7207-187-X. - ISBN 5-279-



02567-4

- Осовский, С. Нейронные сети для обработки информации : [учеб.-справ. изд.] / С. Осовский ; пер. с польск. И.Д. Рудинского. - М. : Финансы и статистика, 2004. - 344 с. : ил. - Библиогр.: с. 330-339. - ISBN 5-279-02567-4

Методическая литература

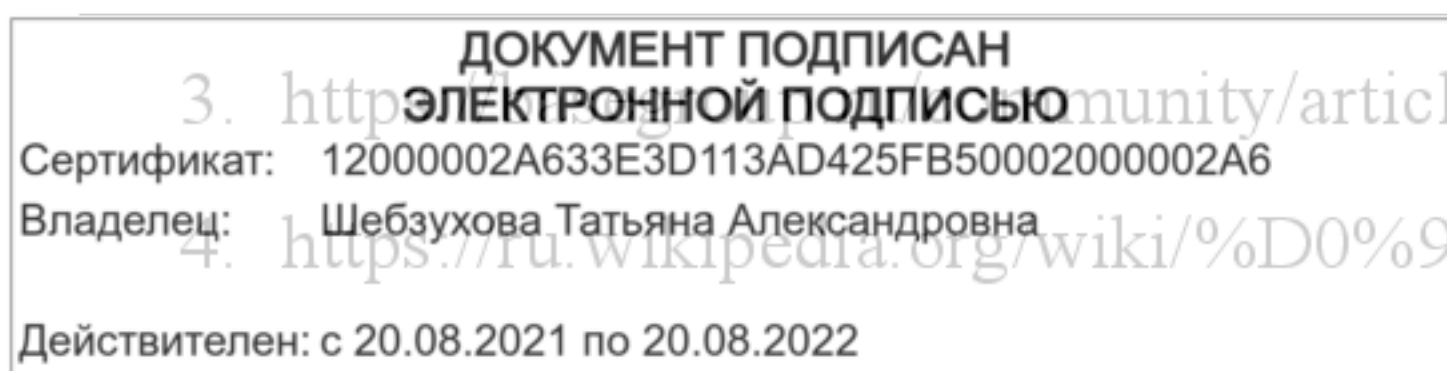
- Методические рекомендации по организации самостоятельной работы студентов по дисциплине "Нейронные сети" : Направление подготовки 01.04.02 - Прикладная математика и информатика. Профиль подготовки «Математическое моделирование». Квалификация выпускника - магистр. Очная форма обучения. Изучается в 3 семестре. Учебный план 2015 г. - Ставрополь : СКФУ, 2015. - 16 с.
- Комашинский В.И. Нейронные сети и их применение в системах управления и связи / Д.А. Смирнов. - М: Горячая линия-Телеком, 2002. - 94 с. - с 88
- Нейронные сети: история развития теории : учеб. пособие для вузов / под ред. А. И. Галушкина, Я. З. Цыпкина. - М. : ИПРЖР, 2001. - 840 с. - (Нейрокомпьютеры и их применение, Кн. 5). - Гриф: Рек. МО. - Прил.: с. 826-834. - ISBN 5-93108-007-4
- Ширяев, В. И. Финансовые рынки. Нейронные сети, хаос и нелинейная динамика : [учеб. пособие] / В.И. Ширяев. - 3-е изд. - М. : КРАСАНД, 2010. - 232 с. : ил. - На учебнике гриф: Доп. УМО. - Библиогр.: с. 210-221. - ISBN 978-5-396-00273-9, экземпляров 1

Интернет-ресурсы

- <http://algolist.manual.ru/ai/ga/ga1.php>
- <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>

3. <http://community/articles/ga-math>

4. <https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%>



D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D 0%B9_
%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D
1%82%D0%BC

5. <http://www.prodav.narod.ru/dsp/index.html> - Давыдов А.В. Цифровая обработка сигналов. Тематические лекции: Учебное пособие в электронной форме. – Екатеринбург, УГТУ

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт информационных технологий и телекоммуникаций



**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
К САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ**

по дисциплине

«Основы распознавания образов»

направления

09.03.02 «Информационные системы и технологии»

профиль подготовки «Прикладное программирование в информационных системах»

Бакалавриат

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна **Ставрополь, 2019 г.**

Действителен: с 20.08.2021 по 20.08.2022

1. Цель и задачи освоения дисциплины

Цель освоения дисциплины «Основы распознавания образов» - формирование профессиональных компетенций будущего магистра по направлению подготовки 09.03.02 – Информационные системы и технологии и получение знаний, практических навыков и умений, используемых при разработке сложных систем и управлении их жизненным циклом. Задачей дисциплины является не только знакомство студентов с основными понятиями системной инженерии, международными стандартами и практиками, но и формирование системного подхода к созданию сложных систем. Целью курса является также рассмотрение перспектив развития системного подхода к созданию сложных проектов в нашей стране.

2. Место дисциплины в структуре ОП магистратуры

Дисциплина относится к вариативной части Блока 1(Б1.В.ДВ.04.02). Ее освоение происходит в 8 семестре.

3. Связь с предшествующими дисциплинами

Для успешного освоения дисциплины Б1.В.ДВ.04.02 «Основы распознавания образов» студент должен знать основы математики и информатики.

4. Связь с последующими дисциплинами

Дисциплина «Основы распознавания образов» включена в учебный процесс для формирования у магистрантов системного подхода к стратегии проектирования сложных информационных систем, к определению целей проектирования, к выявлению заинтересованных сторон и их требований. Дисциплина способствует формированию у бакалавров комплексного представления о современных требованиях к созданию крупных проектов и тенденциях развития информационных систем. Изучение данной дисциплины является необходимой основой для изучения дисциплин «Информационные системы в науке и производстве», «Промышленный интернет», «Анализ, моделирование и оптимизация бизнес-процессов», а также при подготовке к государственной итоговой аттестации.

5. Компетенции обучающегося, формируемые в результате освоения дисциплины

5.1. Наименование компетенций

Код	Формулировка
ПК-2	Способен использовать современные инструментальные средства и технологии программирования при разработке прикладного программного обеспечения вычислительных средств и систем различного функционального назначения

6. Структура и содержание дисциплины (модуля)

	Акад. часов	3.е
Объем занятий	108.00	3.00
В том числе аудиторных		
Сертификат: 12000002A633E3D113AD425FB50002000002A6	40.00	
Владелец: Шебзухова Татьяна Александровна		
Действителен: с 20.08.2021 по 20.08.2022	20.00	
Лабораторных работ	20.00	

7. Организационно-методические рекомендации по освоению дисциплины

Самостоятельная работа студентов является важнейшим условием формирования научного способа познания. Она проводится накануне каждого лабораторного занятия и включает изучение необходимого для выполнения лабораторной работы теоретического материала, а также подготовку отчета по выполненным на лабораторном и практическом занятии заданиям.

Подготовленный материал оформляется в виде отчета. Самостоятельные занятия (СЗ) являются одной из активных форм обучения.

Самостоятельные занятия по дисциплине «Основы распознавания образов» имеют целью:

- закрепить и углубить знания, полученные студентами на лекциях и в процессе лабораторных занятий;
- привить практические навыки при проведении анализа и разработки сложных систем.

Предлагаемые методические рекомендации содержат информацию для студентов, необходимую при подготовке и проведении лабораторных и практических занятий по дисциплине «Основы распознавания образов».

8. Методические указания по выполнению самостоятельной работы студентов

Методические указания должны включать следующие разделы:

- цель работы;
- задание, которое должно быть выполнено студентом в результате проведения самостоятельной работы;
- варианты индивидуальных заданий;
- основные теоретические положения, необходимые для выполнения задания, они должны быть краткими и содержать ссылки на литературу, в которой эти положения изложены в объеме, достаточном для выполнения самостоятельной работы;
- этапы выполнения задания с указанием конкретных сроков выполнения каждого из этапов и всего задания в целом;
- требования к оформлению графической и текстовой части самостоятельной работы;
- пример выполнения одного из вариантов задания и оформления отчета;
- библиографический список использованных источников.

9. Содержание самостоятельной работы

Наименование разделов и тем дисциплины, их краткое содержание; вид самостоятельной работы представлено в таблице 2.

Таблица 2 – Наименование разделов и тем для самостоятельной работы

№	Раздел (тема) дисциплины	Реализуемые компетенции	Контактная работа обучающихся с преподавателем, часов				Самостоятельная работа, часов
			Лекции	Практические занятия	Лабораторные работы	Групповые консультации	
8 семестр							
	ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ		2.00		2.00		
			4.00		2.00		

1 Основы классификации

2 Классификаторы, основанные на оценке функций оптимизации

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

3	Создание и обучение нейронной сети на языке высокого уровня среды Matlab		4.00		4.00	
4	Классификаторы, основанные на оценке функции оптимизации. Машины опорных векторов и нейронные сети прямого распространения		4.00		2.00	
5	Пример создания и обучения нейронных сетей для задач классификации в среде Matlab		4.00		6.00	
6	Пример создания и обучения нейронных сетей для задач регрессии в среде Matlab		2.00		4.00	
ИТОГО за 8 семестр			20.00		20.00	68.00
ИТОГО			20.00		20.00	68.00

Наименование тем дисциплины, цель, форма контроля, задания, требования к оформлению, перечень литературных источников представлен в таблице 3.

Таблица 3 – Наименование тем дисциплины, цель, форма контроля, задания и требования к оформлению самостоятельной работы

Название темы	Цель	Форма контроля СРС	Задания для СРС	Требования к оформлению результатов СРС	Рекомендуемая литература
Подготовка к лабораторным занятиям	Изучить UML	Индивидуальное собеседование	Сформулировать ответы на контрольные вопросы к лабораторным работам и практическим занятиям	Письменно ответить на контрольные вопросы к лабораторным работам	Основная литература - Курейчик, В. В. Теория эволюционных вычислений / Курейчик В. В. - Москва : ФИЗМАТЛИТ, 2012. - 260 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9221-1390-8 - Курейчик, В. М. Основы теории эволюционных вычислений : Научная монография / Курейчик В. М. - Ростов-на-Дону : Южный федеральный университет, 2010. - 224 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9275-0799-3 - Курейчик, В. М. Поисковая
Изучение средств Matlab для работы с ГА	Изучить подходы, применяемые в системной инженерии для управления жизненными		Сформулировать и письменно ответить на вопросы для контроля владения	Краткий конспект теоретического материала и ответы на вопросы для контроля владения компетенциями	

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**
Сертификат: 12000002A633E3D113AD425FB5000200002A6
Владелец: Шебзухова Татьяна Александровна

**КОМПЕТЕНЦИИ
данного
раздела**

Действителен: с 20.08.2021 по 20.08.2022

Изучение библиотек С++ для работы с ГА	Скращивание, селекция, ранжирование, создание популяций		программы	Краткий конспект теоретического материала и ответы на вопросы для контроля владения компетенциями	адаптация. Теория и практика / Курейчик В. М. - Москва : ФИЗМАТЛИТ, 2016. - 270 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9221-0749-6 Дополнительная литература 4. Методические указания по организации самостоятельной работы по дисциплине «Нечёткая логика и нейронные сети» : для студентов направления 38.03.05 «Бизнес-информатика» (профиль «Электронный бизнес», «Информационная бизнес-аналитика»). Учебный план 2012 г. / сост. И. Ю. Глазкова ; ФГАОУ ВПО Сев.-Кав. федер. ун-та. - Ставрополь : СКФУ, 2015.- 30 с. 5. Барский, А. Б. Нейронные сети: распознавание, управление, принятие решений : [учеб. пособие] / А.Б. Барский. - М. : Финансы и статистика, 2004. - 176 с. : ил. - (Прикладные информационные технологии). - Библиогр.: с. 170-173. - ISBN 5-279-02757 6. Осовский, С. Нейронные сети для обработки информации / Станислав Осовский ; пер. с польск. И. Д. Рудинского. - М. : Финансы и статистика, 2004. - 343 с. : ил. - Библиогр.: с. 3330-3339. - Предм. указ.: с. 340-343. - ISBN 83-7207-187-X. - ISBN 5-279-02567-4 7. Осовский, С. Нейронные сети для обработки информации : [учеб.-справ. изд.] / С. Осовский ; пер. с польск. И.Д. Рудинского. - М. : Финансы и статистика, 2004. - 344 с. : ил. - Библиогр.: с. 330-339. - ISBN 5-279-02567-4
Методы проектирования информационных систем	Изучить методы проектирования ИС				
Стандартизация, качество и надежность программно-обеспечения (ПО)	Изучить стандарты, определяющие качество и надежность ПО				

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

				Методическая литература 8. Методические рекомендации по организации самостоятельной работы студентов по дисциплине "Нейронные сети" : Направление подготовки 01.04.02 - Прикладная математика и информатика. Профиль подготовки «Математическое моделирование». Квалификация выпускника - магистр. Очная форма обучения. Изучается в 3 семестре. Учебный план 2015 г. 9. Комашинский В.И. Нейронные сети и их применение в системах управления и связи/ Д.А.Смирнов. - М:Горячая линия-Телеком,2002. - 94с. – с. 88 10. Нейронные сети: история развития теории : учеб. пособие для вузов / под ред. А. И. Галушкина, Я. З. Цыпкина. - М. : ИПРЖР, 2001. - 840 с. - (Нейрокомпьютеры и их применение, Кн. 5). - Гриф: Рек. МО. - Прил.: с. 826-834. – ISBN 5-93108-007-4 11. Ширяев, В. И. Финансовые рынки. Нейронные сети, хаос и нелинейная динамика : [учеб. пособие] / В.И. Ширяев. - 3-е изд. - М. : КРАСАНД, 2010. - 232 с. : ил. - На учебнике гриф: Доп.УМО. - Библиогр.: с. 210-221. - ISBN 978-5-396-00273-9, экземпляров 1 Интернет-ресурсы 12. http://algolist.manual.ru/ai/ga/ga1.php 13. http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005 14. https://basegroup.ru/community/articles/ga-math
--	--	--	--	--

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

9.1. Примерная тематика заданий для самостоятельной работы студентов

Примерная тематика заданий:

1. Классификация практик системной инженерии. Основные группы процессов. Стандарты по разработке программного обеспечения, по управлению качеством и надежностью.
2. Принципы моделирования деятельности организации. Основные типы методологий проектирования информационных систем. Методология SADT. Методологии серии IDEF. Другие методологии.
3. Документирование программных продуктов.
4. Разработка технического задания.

10. План-график выполнения СРС

№№	Название темы	Срок сдачи результатов, неделя
8 семестр		
1	Различные методы распознавания образов	3
2	Экспорт кода Matlab в приложение, написанное на языке высокого уровня.	13

11. Организация контроля знаний студентов

11.1 Формы контроля знаний студентов

Контроль и оценка знаний, умений и навыков студентов осуществляется на лабораторных и практических занятиях, консультациях, при сдаче экзамена. В ходе контроля знаний преподаватель оценивает понимание студентом содержания дисциплины «Основы распознавания образов», его способность анализировать состояние информационных систем и процессов.

Контроль знаний студентов может осуществляться в следующих формах:

- текущий контроль знаний;
- итоговый контроль знаний. Текущий контроль знаний студентов имеет целью: дать оценку работы каждого студента по усвоению им учебного материала, выявить недостатки в его подготовке и оказать практическую помощь в их устранении.

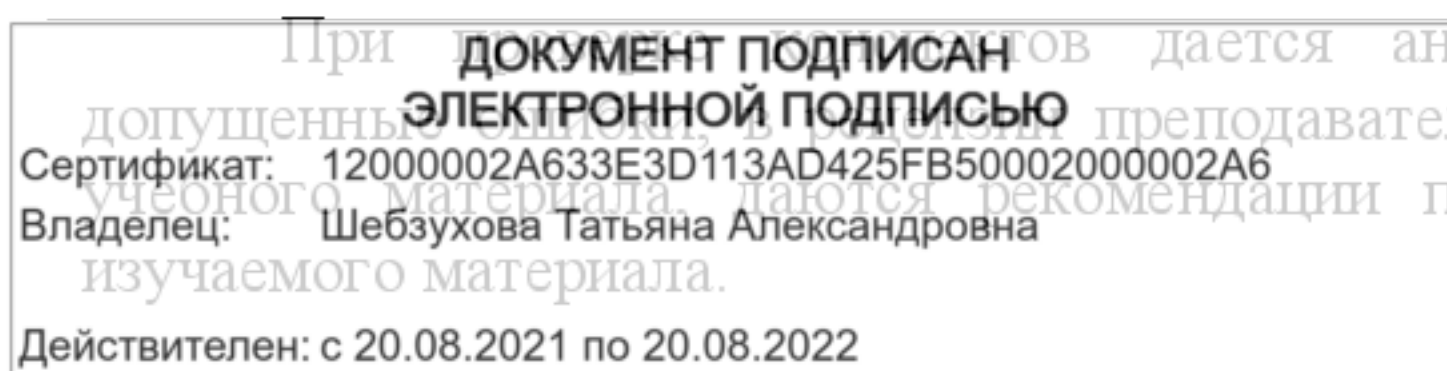
Основными формами текущего контроля знаний студентов являются:

- устный контрольный опрос;
- защита лабораторной работы;
- проверка конспектов лекций;
- проверка конспектов по теме, вынесенной на самостоятельное изучение.

Устный контрольный опрос студентов проводится на лекциях (и лабораторных занятиях). По его результатам преподаватель оценивает качество подготовки студента к занятию.

На лабораторных занятиях знания и практические навыки студентов оцениваются по 5-балльной системе. Полученные оценки выставляются в журнале.

При проведении практических занятий дается анализ качества их ведения. Отмечаются допущенные ошибки, даются рекомендации преподавателя по улучшению качества конспектирования учебного материала, даются рекомендации по улучшению качества конспектирования изучаемого материала.



11.2. Рекомендации по подготовке к экзамену Подготовка к экзамену начинается с начала изучения дисциплины. Необходимо посещать все виды занятий. Экзамен, как итоговый контроль знаний студентов имеет целью проверить и оценить учебную работу студентов, уровень полученных знаний и практических навыков.

Экзамен проводится во 2 семестре после защиты всех лабораторных работ и выполнения заданий к практическим занятиям в объеме учебной программы.

12. Рекомендации по работе с литературой и источниками

Изучение литературы и источников необходимо начинать с прочтения соответствующих глав учебных изданий, учебных пособий или литературы, рекомендованной в качестве основной или дополнительной по дисциплине «Основы распознавания образов», которые прямо или косвенно относятся к изучаемой теме.

При изучении литературы и источников студенту рекомендуется вести краткий конспект. Однако не следует переписывать все содержание изучаемой темы, нужно выписывать лишь основные идеи и главные мысли. В отдельных случаях, когда встречаются важные определения, понятия, необходимый фактический материал и примеры, статистическая информация, имеющие отношение к изучаемой теме, необходимо выписать их в виде цитат с полным указанием библиографических источников.

Конспектирование рекомендуемой литературы и источников необходимо вести с распределением собранных материалов по отдельным главам и параграфам согласно учебно-тематическому плану. Необходимо выписывать все выходные данные по используемой литературе и источникам.

Основой технологии интенсификации обучения на платформе цифровых образовательных технологий являются учебно-иллюстрационные материалы (опорный конспект) по дисциплине «Системной инженерии».

Работа с учебно-иллюстрационными материалами имеет следующие этапы.

1. Изучение теоретических основ учебного материала в аудитории: изложение преподавателем изучаемого материала студентам с объяснением по опорному конспекту;
 2. Самостоятельная работа: индивидуальная работа студентов по опорному конспекту; фронтальное закрепление по блокам опорного конспекта.
 3. Первое повторение - воспроизведение содержания заданной темы опорного конспекта по памяти.
 4. Устное проговаривание материала опорного конспекта – необходимый этап внешне речевой деятельности при усвоении учебного материала.
 5. Второе повторение – взаимопрос и взаимопомощь студентов друг другу.
- Применение учебно-иллюстрационных материалов позволяет обобщить сложный по содержанию материал, активизировать мыслительную деятельность студентов.

Необходимо помнить, что главное для студента в самостоятельной работе с рекомендуемой литературой и источниками - это формирование своего индивидуального стиля, который может стать основой в будущей профессиональной деятельности.

13. Перечень рекомендуемой литературы

Основная литература

1. Курейчик, В. В. Теория эволюционных вычислений / Курейчик В. В. - Москва : ФИЗМАТЛИТ, 2012. - 260 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9221-1390-8
2. Курейчик, В. М. Основы теории эволюционных вычислений : Научная монография / Курейчик В. М. - Ростов-на-Дону : Южный федеральный университет, 2006. - 270 с. - Книга находится в базовой версии ЭБС IPRbooks.

Сертификат: - 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

3. Курейчик, В. М. Поисковая адаптация. Теория и практика / Курейчик В. М. - Москва : ФИЗМАТЛИТ, 2006. - 270 с. - Книга находится в базовой версии ЭБС IPRbooks.

IPRbooks. - ISBN 978-5-9221-0749-6

Дополнительная литература

4. Методические указания по организации самостоятельной работы по дисциплине «Нечёткая логика и нейронные сети» : для студентов направления 38.03.05 «Бизнес-информатика» (профиль «Электронный бизнес», «Информационная бизнес-аналитика»). Учебный план 2012 г. / сост. И. Ю. Глазкова ; ФГАОУ ВПО Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 20 с.
5. Барский, А. Б. Нейронные сети: распознавание, управление, принятие решений : [учеб. пособие] / А.Б. Барский. - М. : Финансы и статистика, 2004. - 176 с. : ил. - (Прикладные информационные технологии). - Библиогр.: с. 170-173. - ISBN 5-279-02757-X
6. Осовский, С. Нейронные сети для обработки информации / Станислав Осовский ; пер. с польск. И. Д. Рудинского. - М. : Финансы и статистика, 2004. - 343 с. : ил. - Библиогр.: с. 3330-339. - Предм. указ.: с. 340-343. - ISBN 83-7207-187-X. - ISBN 5-279-02567-4
7. Осовский, С. Нейронные сети для обработки информации : [учеб.-справ. изд.] / С. Осовский ; пер. с польск. И.Д. Рудинского. - М. : Финансы и статистика, 2004. - 344 с. : ил. - Библиогр.: с. 330-339. - ISBN 5-279-02567-4

Методическая литература

8. Методические рекомендации по организации самостоятельной работы студентов по дисциплине "Нейронные сети" : Направление подготовки 01.04.02 - Прикладная математика и информатика. Профиль подготовки «Математическое моделирование». Квалификация выпускника - магистр. Очная форма обучения. Изучается в 3 семестре. Учебный план 2015 г. - Ставрополь : СКФУ, 2015. - 16 с.
9. Комашинский В.И. Нейронные сети и их применение в системах управления и связи/ Д.А.Смирнов. - М:Горячая линия-Телеком,2002. - 94с. - с88
10. Нейронные сети: история развития теории : учеб. пособие для вузов / под ред. А. И. Галушкина, Я. З. Цыпкина. - М. : ИПРЖР, 2001. - 840 с. - (Нейрокомпьютеры и их применение, Кн. 5). - Гриф: Рек. МО. - Прил.: с. 826-834. - ISBN 5-93108-007-4
11. Ширяев, В. И. Финансовые рынки. Нейронные сети, хаос и нелинейная динамика : [учеб. пособие] / В.И. Ширяев. - 3-е изд. - М. : КРАСАНД, 2010. - 232 с. : ил. - На учебнике гриф: Доп.УМО. - Библиогр.: с. 210-221. - ISBN 978-5-396-00273-9, экземпляров 1

Интернет-ресурсы

12. <http://algotlist.manual.ru/ai/ga/ga1.php>
13. <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>
14. <https://basegroup.ru/community/articles/ga-math>
15. https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC
16. <http://www.prodav.narod.ru/dsp/index.html> - Давыдов А.В. Цифровая обработка сигналов. Тематические лекции: Учебное пособие в электронной форме. – Екатеринбург, УГГУ

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022