МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования «Южно-Уральский государственный университет»  
(национальный исследовательский университет)

Высшая школа электроники и компьютерных наук

Кафедра «Информационно-измерительная техника»

Задача классификации (распознавания) образов с помощью нейронных сетей.

ОТЧЁТ

по практической работе № 2

по дисциплине «Интеллектуальные средства измерений»

Выполнил:

студент группы КЭ–413

/ С.С. Ильин /

(подпись)

« » 2022 г.

Проверил: доцент

/ А.С. Волосников /

(подпись)

« » 2022 г.

ОГЛАВЛЕНИЕ

[1. ЗАДАНИЕ 3](#_Toc99910648)

[2. ОПИСАНИЕ МЕТОДОВ 3](#_Toc99910649)

[3. КОД ПРОГРАММЫ 5](#_Toc99910650)

[4. ХОД РАБОТЫ 7](#_Toc99910651)

[5. ВЫВОД 11](#_Toc99910652)

# ЗАДАНИЕ

1) Создать черно-белые графические образы цифр от 0 до 9 в формате .bmp невысокого разрешения (до 15х15 пикс.);

2) обучить нейронную сеть (персептрон - *perceptron* или вероятностную сеть - *newpnn*)  классифицировать эти образы;

3) провести тестирование сети на зашумленных (функция - *imnoise*) образах (типы шумов: гауссовый - '*gaussian*' и бинарный - '*salt & pepper*');

4) с помощью автокодировщика (*trainAutoencoder*) сжать обучающие образы вполовину от исходного разрешения и повторить пункты 2 и 3 задания с применением автокодировщика.

# ОПИСАНИЕ МЕТОДОВ

Решение задачи классификации является одним из важнейших применений нейронных сетей.

Задача классификации представляет собой задачу отнесения образца к одному из нескольких попарно не пересекающихся множеств. Примером таких задач может быть, например, задача определения кредитоспособности клиента банка, медицинские задачи, в которых необходимо определить, например, исход заболевания, решение задач управления портфелем ценных бумаг (продать купить или "придержать" акции в зависимости от ситуации на рынке), задача определения жизнеспособных и склонных к банкротству фирм.

**Персептрон (Perceptron)** – простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов.

Первыми в работу включаются сенсорные элементы. Они могут находиться либо в состоянии покоя (сигнал равен 0), либо в состоянии возбуждения (сигнал равен 1). Далее сигналы от сенсорных элементов передаются ассоциативным элементам по так называемым сенсорно- ассоциативным связям. Эти связи могут иметь веса, равные только -1, 0 или 1.

Затем сигналы от сенсорных элементов, прошедших по сенсорно- ассоциативным связям попадают в ассоциативные элементы. Стоит заметить, что одному ассоциативному элементу может соответствовать несколько сенсорных элементов. Если сигналы, поступившие на ассоциативный элемент, в совокупности превышают некоторый его порог θ, то этот ассоциативный элемент возбуждается и выдает сигнал, равный 1. В противном случае (сигнал от сенсорных элементов не превысил порога ассоциативного элемента), генерируется нулевой сигнал.

Далее сигналы, которые произвели возбужденные ассоциативные элементы, направляются к сумматору (реагирующий элемент), действие которого вам уже известно. Однако, чтобы добраться до реагирующего элемента, они проходят по ассоциативно-реагирующим связям, у которых тоже есть веса. Однако здесь они уже могут принимать любые значения.

Реагирующий элемент складывает друг с другом взвешенные сигналы от ассоциативных элементов и, если превышен определенный порог, генерирует выходной сигнал, равный 1. Это означает, например, что в общем потоке информации от глаз мы распознали лицо человека.

Если порог не превышен, то выход персептрона равен -1. То есть мы не выделили лицо из общего потока информации.

**Вероятностные нейронные сети (PNN-сети)** – нейронные сети, в которых плотность вероятности принадлежности классам оценивается посредством ядерной аппроксимации. Вероятностные нейронные сети имеют слоистую структуру. В простейшем случае есть три слоя - входной, радиальный и выходной. Каждому обучающему примеру соответствует один элемент радиального слоя. Каждому классу соответствует один выходной элемент, который соединен только с радиальными элементами, относящимися к его классу. Выходной элемент суммирует сигналы всех радиальных элементов, принадлежащих к его классу. Нормированные значения выходных сигналов позволяют оценить вероятности принадлежности классам.

Вероятностная сеть имеет три слоя: входной, радиальный и выходной. Радиальные элементы берутся по одному на каждый пример. Каждый из них имеет гауссовскую функцию с центром в этом примере. Каждому классу отвечает один выходной элемент. Выходной элемент соединен лишь с радиальными элементами, относящихся к его классу и подытоживает выходы всех элементов, принадлежащих к его классу. Значения выходных сигналов получаются пропорционально ядерным оценкам вероятности принадлежности соответствующим классам.

# КОД

clear; clc; close all;

%======================

X = zeros(10,10,10);

for i = 1:10

filename = strcat('dva\',num2str(i-1), '.bmp');

X(:,:,i) = imread(filename);

subplot(1,10,i); imshow(X(:,:,i));

end

P = reshape(X,10\*10,10); % pATTERN

T = eye(10); % Target

%===========================

net = feedforwardnet (5);

net = train(net,P,T);

n = 0;

V = P;

Y = sim(net, V)

%===========================

net = perceptron;

net = train(net,P,T);

n = 0;

V = P;

d =0.04; N = imnoise(X, 'salt & pepper', d);

for i = 1:10

subplot(1,10,i); imshow( N(:,:,i));

end

d =0.002; N = imnoise(X, 'gaussian', d);

for i = 1:10

subplot(1,10,i); imshow( N(:,:,i));

end

V = reshape(N,10\*10,10);

Y = sim(net, V)

%===========================

spread = 1;

Tc = 1:10;

T = ind2vec(Tc)

net = newpnn(P,T,spread);

d =0.5; N = imnoise(X, 'salt & pepper', d);

for i = 1:10

subplot(1,10,i); imshow( N(:,:,i));

end

d =0.5; N = imnoise(X, 'gaussian', d);

for i = 1:10

subplot(1,10,i); imshow( N(:,:,i));

end

V = reshape(N,10\*10,10);

Yc = sim(net,V);

Y = vec2ind(Yc)-1

%===========================

autoenc = trainAutoencoder(P, 5\*5);

R = predict(autoenc, P);

S = reshape(R,10,10,10);

Z = encode(autoenc, P);

E = reshape(Z,5,5,10);

figure;

for i = 1:10

subplot(1,10,i); imshow( S(:,:,i));

end

figure;

for i = 1:10

subplot(1,10,i); imshow( E(:,:,i));

end

spread = 1;

Tc = 1:10;

T = ind2vec(Tc)

net = newpnn(R,T,spread);

d =0.4; N = imnoise(X, 'salt & pepper', d);

for i = 1:10

subplot(1,10,i); imshow( N(:,:,i));

end

d =0.4; N = imnoise(X, 'gaussian', d);

for i = 1:10

subplot(1,10,i); imshow( N(:,:,i));

end

V = reshape(N,10\*10,10);

Yc = sim(net,V);

Y = vec2ind(Yc)-1

# ХОД РАБОТЫ

Создали черно-белые графические образы цифр от 0 до 9 в формате .bmp невысокого разрешения (10х10 пикс.) и вывели их на рисунке 1.

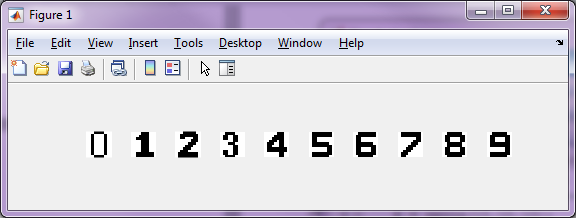


Рисунок 1 – Графические образы цифр

Обучили нейронную сеть функцией персептрона – **perceptron** и провели тестирование сети на зашумленных (функция - **imnoise**) образах (типы шумов: гауссовый - '**gaussian'** и бинарный - **'salt & pepper'**). Зашумление **'salt & pepper**' и распознавание этих цифр представлены на рисунках 2 и 3, при *d = 0.04*, так как при d больше 0.04 сеть не распознаёт всех цифр.

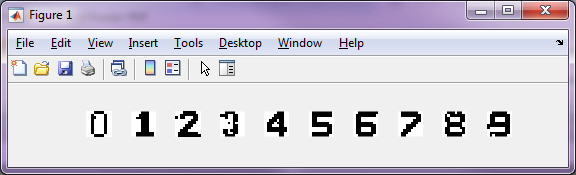


Рисунок 2 – Графические образы цифр при зашумлении '*salt & pepper*'.

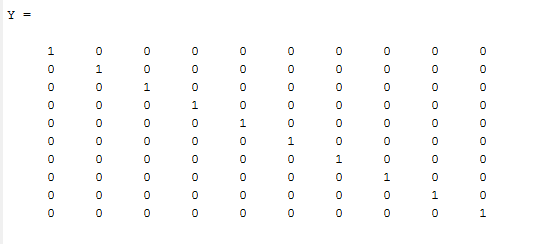


Рисунок 3 – Распознавание нейронной сетью

Зашумление '***gaussian***' и распознавание этих цифр представлены на рисунках 4 и 5, при *d = 0.002*, так как при d больше 0.002 сеть не распознаёт всех цифр.

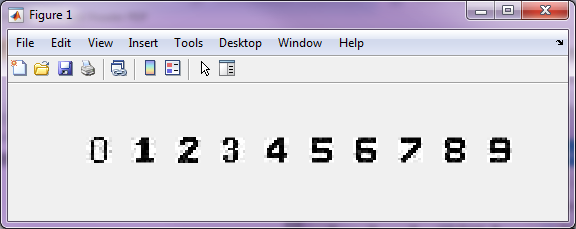


Рисунок 4 – Графические образы цифр при зашумлении '*gaussian*'

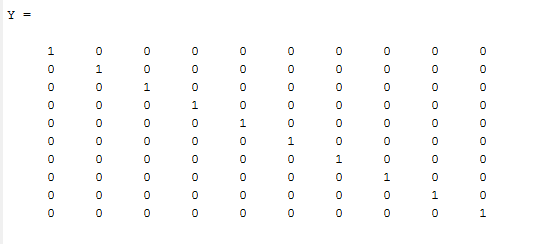


Рисунок 5 – Распознавание нейронной сетью

Обучили сеть функцией**newpnn** и провели тестирование сети на зашумленных (функция - **imnoise**) образах (типы шумов: гауссовый - **'gaussian'** и бинарный – **'salt & pepper'**). Зашумление **'salt & pepper'** и распознавание этих цифр представлены на рисунках 6 и 7, при *d = 0.05*, так как при d больше 0.05 сеть не распознаёт всех цифр.

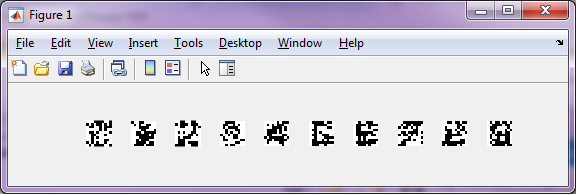


Рисунок 6 – Графические образы цифр при зашумлении '*salt & pepper*'.

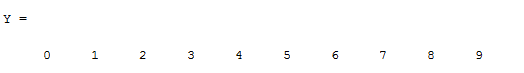


Рисунок 7 – Распознавание сетью

Зашумление **'gaussian'** и распознавание этих цифр представлены на рисунках 8 и 9, при *d = 0.05*, так как при d больше 0.05 сеть не распознаёт всех цифр.

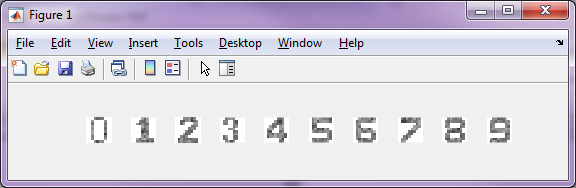


Рисунок 8 – Графические образы цифр при зашумлении '**gaussian**'



Рисунок 9 – Распознавание сетью

С помощью автокодировщика (**trainAutoencoder**) сжать обучающие образы вполовину от исходного разрешения на (5x5). На рисунке 10 показано сжатие и на рисунке 11 процесс востоновления.

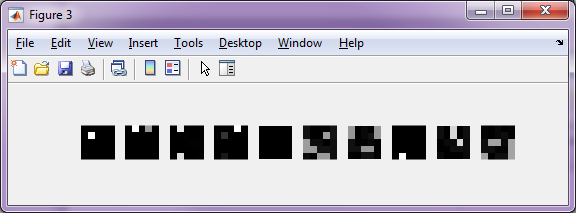


Рисунок 10 – Сжатые рисунки цифр.

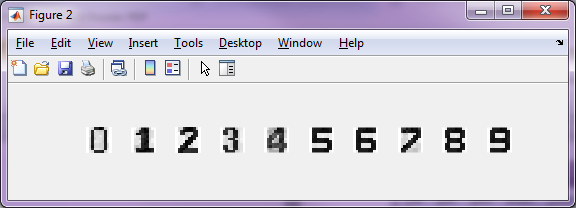


Рисунок 11 – Восстановленные рисунки цифр.

С помощью автокодировщика (**trainAutoencoder**) повторим функцию **newpnn***,* так как она показала себя лучше, чем **perceptron***.* Провели тестирование сети на зашумленных (функция - **imnoise**) образах (типы шумов: гауссовый - **'gaussian'** и бинарный - **'salt & pepper'**). Зашумление **'salt & pepper'** и распознавание этих цифр представлены на рисунках 12 и 13, при *d = 0.04*, так как при d больше 0.04 сеть не распознаёт всех цифр.

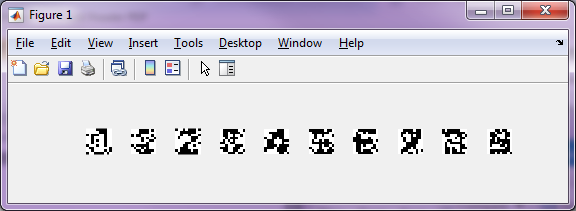


Рисунок 12 – Графические образы цифр при зашумлении '*salt & pepper*'.

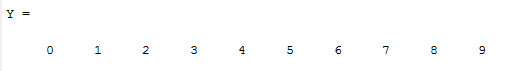


Рисунок 13 – Распознавание сетью

Зашумление **'gaussian'** и распознавание этих цифр представлены на рисунках 14 и 15, при *d = 0.04*, так как при d больше 0.04 сеть не распознаёт всех цифр.

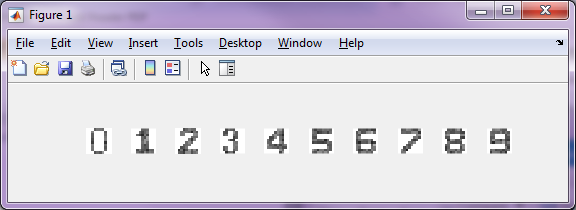


Рисунок 14 – Графические образы цифр при зашумлении '*gaussian*'



Рисунок 15 – Распознавание сетью

# ВЫВОД

Выполнили все поставленные задачи, сравнивая функции *perceptron* и *newpnn, newpnn* распознает цифры лучше прикоэффициенте *d = 0.05*, при '*gaussian*' и '*salt & pepper*', когда *perceptron d = 0.04* при'*salt & pepper*' и *d = 0.002*  при '*gaussian*'.

С помощью автокодировщика (*trainAutoencoder*) сжали обучающие образы вполовину от исходного разрешения (5х5) и повторили функцию *newpnn* с применением автокодировщика, коэффициент *d = 0.04* в обоях случаях зашумления, то есть при восстановлении рисунков результат получается немного хуже.