МIНIСТЕРСТВО ОСВIТИ I НАУКИ УКРАЇНИ

НАЦIОНАЛЬНИЙ ТЕХНIЧНИЙ УНIВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛIТЕХНIЧНИЙ IНСТИТУТ»

Кафедра прикладної математики

Звіт

із лабораторної роботи №2

з дисципліни «Методи штучного інтелекту»

на тему:

«PNN»

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-63 | *Терейковська Л. О.* |
| *Вовченко І.В* |  |

Київ — 2019

# **ЗМІСТ**

[**ЗМІСТ** 1](#_Toc25666656)

[**ТЕОРЕТИЧНІ ВІДОМОСТІ** 2](#_Toc25666657)

[**ДОДАТКИ** 3](#_Toc25666658)

[**Додаток А (Код програми)** 3](#_Toc25666659)

[**Додаток Б (Скріншот)** 5](#_Toc25666660)

# **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Дана нейрона мережа Складається з 4 шарів

* Вхідний шар
* Шар образів
* Шар додавання
* Вихідний шар

Вхідний шар

Лінійна функція активації.

Шар образів

Функція активації - Гауса

Шар додавання

Рахуємо середню ймовірність вхідних сигналів

Вихідний шар

Знаходимо максимальну ймовірність і виводимо і відповідну назву класу.

Функція активації (лінійна)

Функція активації (Гауса)

# **ДОДАТКИ**

# **Додаток А (Код програми)**

# from math import exp

# input\_w = [[1, 1, 2], [1, 2, 1], [2, 2, 1], [5, 5, 5]]

# input\_data = [4, 4, 4]

# class Input\_L:

# def activation(self, x):

# self.y = x \* 1

# return self.y

# class Image\_L:

# def \_\_init\_\_(self, w):

# self.w = w

# def activation(self, x):

# sum = 0

# for i in range(len(self.w)):

# sum += exp(-(self.w[i] - x[i])\*\*2/0.3\*\*2)

# return sum

# class Add\_L:

# def activation(self, x):

# sum = 0

# for i in x:

# sum += i

# res = sum / len(x)

# return res

# class Output\_L:

# def activation(self, x):

# clas = "A"

# val = x[clas]

# for i in x.items():

# if i[1] > val:

# clas = i[0]

# return clas

# input\_1 = Input\_L()

# input\_2 = Input\_L()

# input\_3 = Input\_L()

# image\_1 = Image\_L(input\_w[0])

# image\_2 = Image\_L(input\_w[1])

# image\_3 = Image\_L(input\_w[2])

# image\_4 = Image\_L(input\_w[3])

# class\_A = Add\_L()

# class\_B = Add\_L()

# res = Output\_L()

# y\_input = []

# y\_image = []

# y\_add = {}

# y\_input.append(input\_1.activation(input\_data[0]))

# y\_input.append(input\_1.activation(input\_data[1]))

# y\_input.append(input\_1.activation(input\_data[2]))

# y\_image.append([])

# y\_image.append([])

# y\_image[0].append(image\_1.activation(y\_input))

# y\_image[0].append(image\_2.activation(y\_input))

# y\_image[0].append(image\_3.activation(y\_input))

# y\_image[1].append(image\_4.activation(y\_input))

# y\_add.update({"A": class\_A.activation(y\_image[0])})

# y\_add.update({"B": class\_B.activation(y\_image[1])})

# res = res.activation(y\_add)

# print("Класс",res)

# **Додаток Б (Скріншот)**

