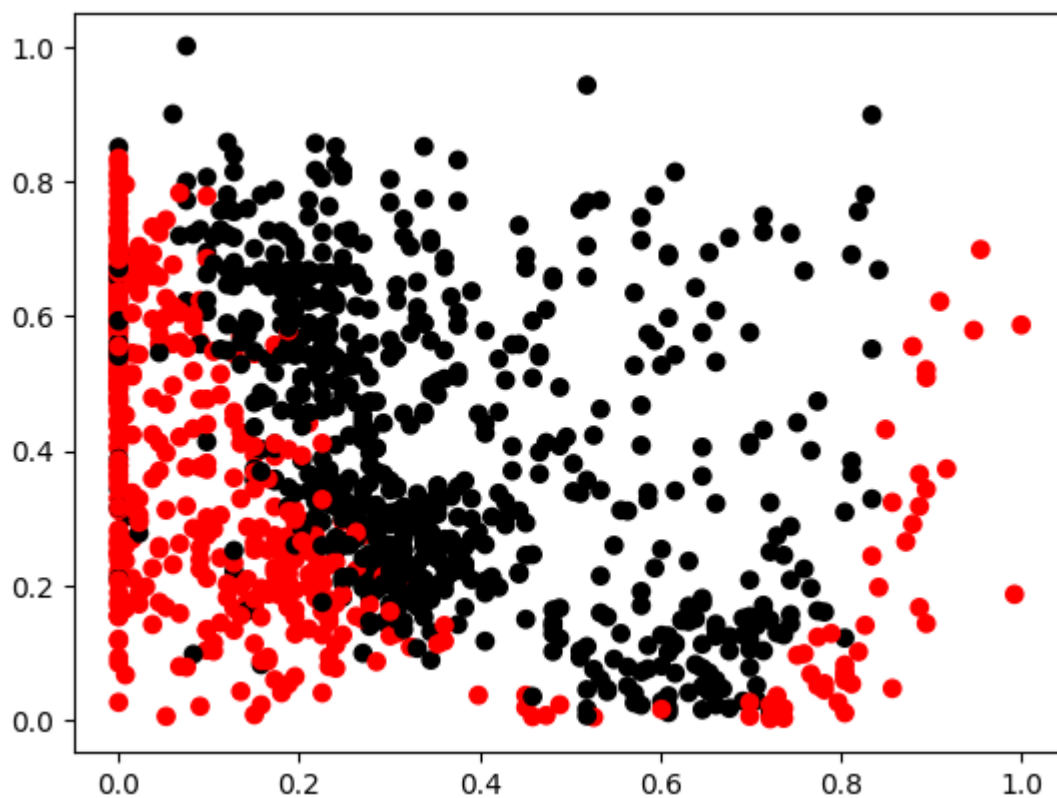


Metody Inteligencji Obliczeniowej
Laboratorium 3
Sieci wielowarstwowe

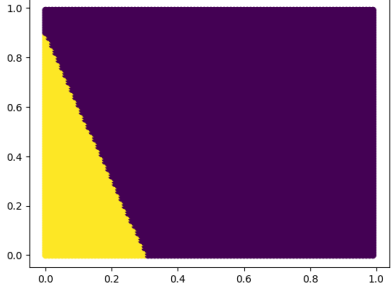
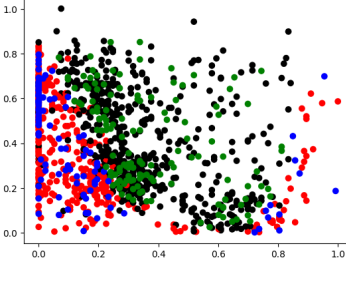
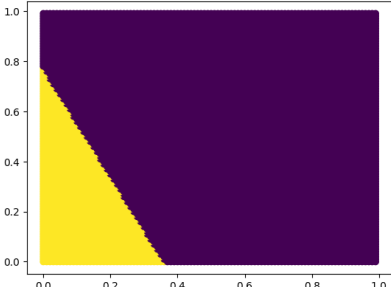
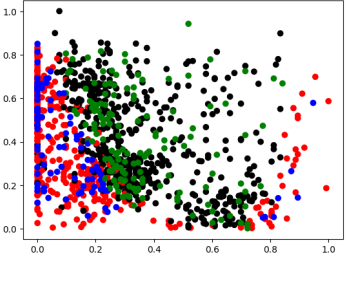
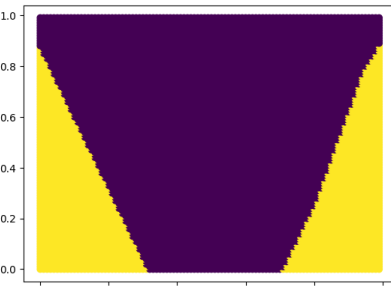
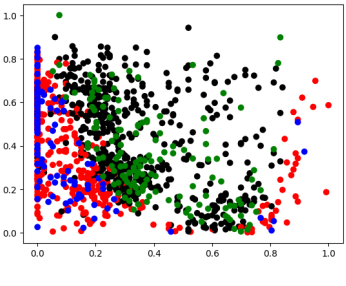

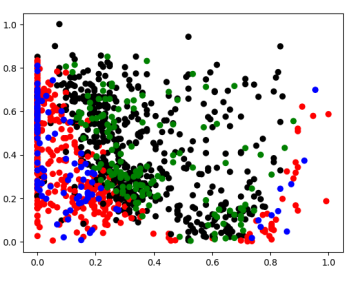
Kamil Pyla

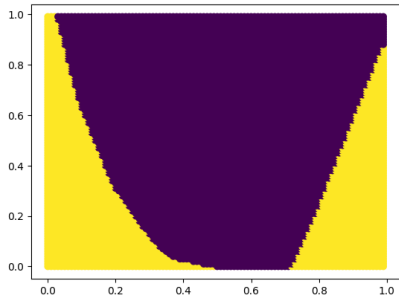
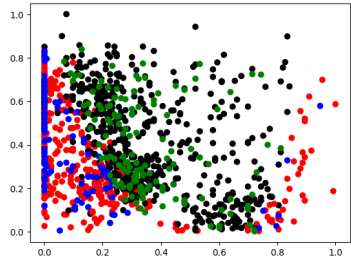
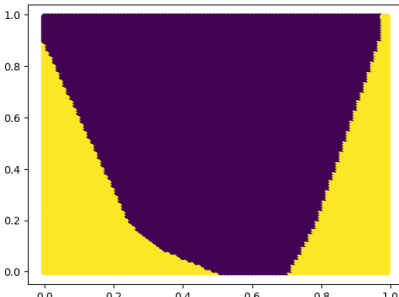
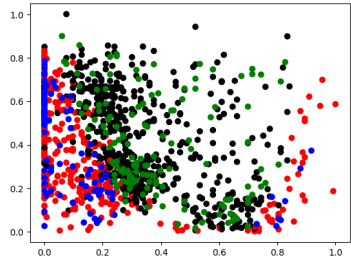
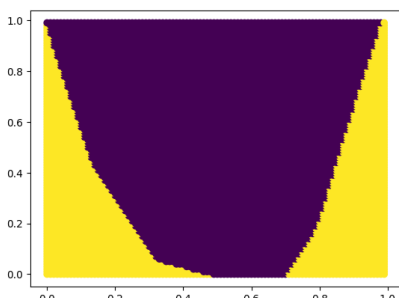
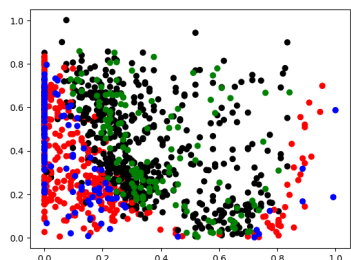
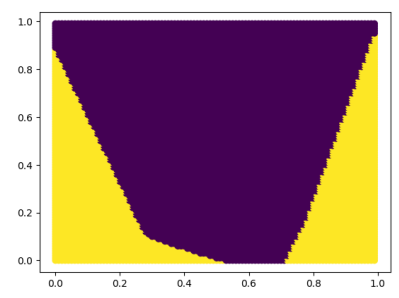
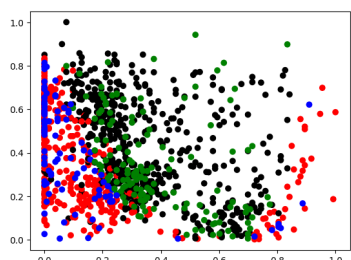


Wyniki zadania pierwszego:



Rys 1. Poprawny podział zbioru

ilość neuronów	ilość ukrytych warstw	dokładność klasyfikacji	podział obszaru klasyfikacji dokonanej przez sieć	Klasyfikacja danych
5	5	85.4 %		
8	8	87.8 %		
9	9	90.7 %		
10	10	91.6 %		

13	13	90.3 %		
15	15	93.7 %		
22	22	93.7 %		
50	50	92 %		

Wnioski: im więcej neuronów w sieci tym łamana którą możemy dokonać podziału zbioru, może być bardziej złożona, jednak użycie zbyt dużej ilości neuronów nie poprawia już dokładności klasyfikacji.

Wyniki zadania 2:

konfiguracja sieci hidden_layer_sizes	dokładność
--	------------

(5, 5)	85.5 %
(8, 8)	93.6 %
(13, 13)	95.5 %

Poprawność klasyfikacji na poziomie 95,5% jest zadowalającą poprawnością

Wyniki zadania 3:

solver: sgd

activation: logistic

learning_rate: adaptive

score: 0.9083333333333333

confusion_matrix: [[141 0 0 0 0 0 0 0 0 0 1]

[0 144 0 0 0 0 0 0 1 0]

[0 2 136 1 0 0 0 0 3 0]

[0 0 0 141 0 4 0 1 0 0]

[0 0 0 0 145 0 0 0 0 0]

[0 0 0 4 0 134 0 0 0 8]

[3 0 0 0 1 0 141 0 0 0]

[0 1 1 0 1 0 0 140 0 0]

[0 8 0 1 0 1 2 0 127 0]

[0 0 0 1 0 2 0 1 1 139]]

solver: sgd

activation: tanh

learning_rate: constant

score: 0.9444444444444444

confusion_matrix: [[140 0 0 0 0 0 2 0 0 0 0]

[0 144 0 0 0 0 0 0 1 0]

[0 0 141 1 0 0 0 0 0 0]

[1 0 0 145 0 0 0 0 0 0]

[0 0 0 0 143 0 0 1 1 0]

[0 0 0 0 0 141 0 0 3 2]

[0 0 0 0 0 1 144 0 0 0]

[0 0 0 0 0 0 0 143 0 0]

[0 1 0 0 0 1 0 0 137 0]

[0 1 0 0 0 0 0 0 1 142]]

solver: sgd
activation: tanh
learning_rate: invscaling
score: 0.08611111111111111
confusion_matrix: [[0 11 11 12 0 0 0 1 107 0]
[0 0 4 4 13 0 0 4 120 0]
[6 0 28 3 27 4 1 26 47 0]
[1 12 25 1 0 0 5 5 97 0]
[0 3 0 10 1 3 0 7 121 0]
[0 5 20 1 23 1 0 5 91 0]
[0 5 5 47 4 15 0 5 64 0]
[0 0 6 2 4 0 0 14 117 0]
[0 10 5 5 3 2 0 9 105 0]
[0 9 4 0 0 0 1 2 128 0]]

solver: sgd
activation: tanh
learning_rate: adaptive
score: 0.9472222222222222
confusion_matrix: [[140 0 0 0 0 0 1 1 0 0 0]
[0 145 0 0 0 0 0 0 0 0 0]
[0 1 141 0 0 0 0 0 0 0 0]
[0 0 0 146 0 0 0 0 0 0 0]
[0 0 0 0 144 0 1 0 0 0 0]
[0 0 0 1 0 144 0 0 0 0 1]
[0 0 0 0 0 0 145 0 0 0 0]
[0 0 0 0 0 0 0 143 0 0 0]
[0 2 0 1 0 0 0 0 136 0 0]
[0 0 0 0 0 0 0 0 1 143]]

solver: sgd
activation: relu
learning_rate: constant
score: 0.95
confusion_matrix: [[142 0 0 0 0 0 0 0 0 0 0]
[0 145 0 0 0 0 0 0 0 0 0]
[0 0 142 0 0 0 0 0 0 0 0]
[0 0 0 146 0 0 0 0 0 0 0]
[0 0 0 0 145 0 0 0 0 0 0]

```
[ 0 0 0 0 0 0 144 0 0 0 2]
[ 0 0 0 0 0 0 0 145 0 0 0]
[ 0 0 0 0 0 0 0 0 143 0 0]
[ 0 0 0 0 0 0 0 0 0 139 0]
[ 0 0 0 0 0 0 0 0 0 0 144]]
```

solver: sgd

activation: relu

learning_rate: invscaling

score: 0.25555555555555554

```
confusion_matrix: [[ 0  4  0  0 79  8 35  0  0 16]
 [ 2 91  0  7 11  3  7 23  0  1]
 [ 0 30  3 21  3  1  4 80  0  0]
 [ 8 39  3 39  5  7  4 38  0  3]
 [ 0 18  0  0 76 29 21  1  0  0]
 [ 0 36  0  0 41 36 19 14  0  0]
 [ 0 12  2  1 38 67 21  3  0  1]
 [ 2 18  0  4  0 13  2 101  0  3]
 [ 1 69  0 14 19  7 10 17  0  2]
 [ 0 66  0  6 29 11  5  9  0 18]]
```

solver: sgd

activation: relu

learning_rate: adaptive

score: 0.9555555555555556

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0  0]
 [ 0 145  0  0  0  0  0  0  0  0  0]
 [ 0  0 142  0  0  0  0  0  0  0  0]
 [ 0  0  0 146  0  0  0  0  0  0  0]
 [ 0  0  0  0 145  0  0  0  0  0  0]
 [ 0  0  0  0  0 146  0  0  0  0  0]
 [ 0  0  0  0  0  0 145  0  0  0  0]
 [ 0  0  0  0  0  0  0 143  0  0  0]
 [ 0  1  0  0  0  0  0  0 138  0  0]
 [ 0  0  0  0  0  0  0  0  1 143]]
```

solver: lbfgs

activation: identity

learning_rate: constant

score: 0.9444444444444444

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0  0]
 [ 0 145  0  0  0  0  0  0  0  0  0]
 [ 0  0 142  0  0  0  0  0  0  0  0]
 [ 0  0  0 146  0  0  0  0  0  0  0]
 [ 0  0  0  0 145  0  0  0  0  0  0]
 [ 0  0  0  0  0 146  0  0  0  0  0]
 [ 0  0  0  0  0  0 145  0  0  0  0]
 [ 0  0  0  0  0  0  0 143  0  0  0]
 [ 0  0  0  0  0  0  0  0 139  0  0]
 [ 0  0  0  0  0  0  0  0  0 144  0]]
```

solver: lbfgs

activation: identity

learning_rate: invscaling

score: 0.9527777777777777

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0  0]
 [ 0 145  0  0  0  0  0  0  0  0  0]
 [ 0  0 142  0  0  0  0  0  0  0  0]
 [ 0  0  0 146  0  0  0  0  0  0  0]
 [ 0  0  0  0 145  0  0  0  0  0  0]
 [ 0  0  0  0  0 146  0  0  0  0  0]
 [ 0  0  0  0  0  0 145  0  0  0  0]
 [ 0  0  0  0  0  0  0 143  0  0  0]
 [ 0  0  0  0  0  0  0  0 139  0  0]
 [ 0  0  0  0  0  0  0  0  0 144  0]]
```

solver: lbfgs

activation: identity

learning_rate: adaptive

score: 0.9444444444444444

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0  0]
 [ 0 145  0  0  0  0  0  0  0  0  0]
 [ 0  0 142  0  0  0  0  0  0  0  0]
 [ 0  0  0 146  0  0  0  0  0  0  0]
 [ 0  0  0  0 145  0  0  0  0  0  0]
 [ 0  0  0  0  0 146  0  0  0  0  0]
 [ 0  0  0  0  0  0 145  0  0  0  0]
 [ 0  0  0  0  0  0  0 143  0  0  0]
 [ 0  0  0  0  0  0  0  0 139  0  0]]
```

[0 0 0 0 0 0 0 0 0 0 0 144]]

solver: adam

activation: identity

learning_rate: constant

score: 0.9611111111111111

confusion_matrix: [[142 0 0 0 0 0 0 0 0 0 0]

[0 146 0 0 0 0 0 0 0 0 0]

[0 0 142 0 0 0 0 0 0 0 0]

[0 0 0 146 0 0 0 0 0 0 0]

[0 0 0 0 145 0 0 0 0 0 0]

[0 0 0 0 0 145 0 0 0 0 0]

[0 0 0 0 0 0 145 0 0 0 0]

[0 0 0 0 0 0 0 143 0 0 0]

[0 0 0 0 0 0 0 0 139 0 0]

[0 0 0 0 0 0 0 0 0 144]]

solver: adam

activation: identity

learning_rate: invscaling

score: 0.95

confusion_matrix: [[142 0 0 0 0 0 0 0 0 0 0]

[0 146 0 0 0 0 0 0 0 0 0]

[0 0 142 0 0 0 0 0 0 0 0]

[0 0 0 146 0 0 0 0 0 0 0]

[0 0 0 0 145 0 0 0 0 0 0]

[0 0 0 0 0 145 0 0 0 0 0]

[0 0 0 0 0 0 145 0 0 0 0]

[0 0 0 0 0 0 0 143 0 0 0]

[0 0 0 0 0 0 0 0 139 0 0]

[0 0 0 0 0 0 0 0 0 144]]

solver: adam

activation: identity

learning_rate: adaptive

score: 0.9611111111111111

confusion_matrix: [[142 0 0 0 0 0 0 0 0 0 0]

[0 146 0 0 0 0 0 0 0 0 0]

[0 0 142 0 0 0 0 0 0 0 0]

[0 0 0 146 0 0 0 0 0 0 0]


```
[ 0 0 0 0 145 0 0 0 0 0]
[ 0 0 0 0 0 145 0 0 0 0]
[ 0 0 0 0 0 0 145 0 0 0]
[ 0 0 0 0 0 0 0 143 0 0]
[ 0 0 0 0 0 0 0 0 139 0]
[ 0 0 0 0 0 0 0 0 0 144]]
```

solver: adam

activation: logistic

learning_rate: constant

score: 0.9555555555555556

```
confusion_matrix: [[142 0 0 0 0 0 0 0 0 0]
[ 0 146 0 0 0 0 0 0 0 0]
[ 0 0 142 0 0 0 0 0 0 0]
[ 0 0 0 145 0 1 0 0 0 0]
[ 0 0 0 0 145 0 0 0 0 0]
[ 0 0 0 0 0 145 0 0 0 0]
[ 0 0 0 0 1 0 144 0 0 0]
[ 0 0 0 0 0 0 0 143 0 0]
[ 0 0 0 0 0 0 0 0 139 0]
[ 0 0 0 1 0 0 0 0 0 143]]
```

solver: adam

activation: logistic

learning_rate: invscaling

score: 0.9305555555555556

```
confusion_matrix: [[142 0 0 0 0 0 0 0 0 0]
[ 0 146 0 0 0 0 0 0 0 0]
[ 0 0 142 0 0 0 0 0 0 0]
[ 0 0 0 146 0 0 0 0 0 0]
[ 0 0 0 0 145 0 0 0 0 0]
[ 0 0 0 1 0 144 0 0 0 0]
[ 0 0 0 0 0 0 145 0 0 0]
[ 0 0 0 0 0 0 0 143 0 0]
[ 0 0 0 0 0 0 0 0 139 0]
[ 0 0 0 1 0 0 0 0 0 143]]
```

solver: adam

activation: logistic

learning_rate: adaptive

score: 0.9388888888888889

confusion_matrix: [[142 0 0 0 0 0 0 0 0 0]
[0 146 0 0 0 0 0 0 0 0]
[0 0 142 0 0 0 0 0 0 0]
[0 0 0 146 0 0 0 0 0 0]
[0 0 0 0 145 0 0 0 0 0]
[0 0 0 0 0 145 0 0 0 0]
[0 0 0 0 0 0 145 0 0 0]
[0 0 0 0 0 0 0 143 0 0]
[0 0 0 0 0 0 0 0 139 0]
[0 0 0 0 0 0 0 0 0 144]]

solver: adam

activation: tanh

learning_rate: constant

score: 0.9416666666666667

confusion_matrix: [[142 0 0 0 0 0 0 0 0 0]
[0 146 0 0 0 0 0 0 0 0]
[0 0 142 0 0 0 0 0 0 0]
[0 0 0 146 0 0 0 0 0 0]
[0 0 0 0 145 0 0 0 0 0]
[0 0 0 0 0 145 0 0 0 0]
[0 0 0 0 0 0 145 0 0 0]
[0 0 0 0 0 0 0 143 0 0]
[0 1 0 0 0 0 0 0 138 0]
[0 0 0 0 0 0 0 0 0 144]]

solver: adam

activation: tanh

learning_rate: invscaling

score: 0.9138888888888889

confusion_matrix: [[139 0 0 0 0 3 0 0 0 0]
[0 145 0 0 0 0 0 0 1 0]
[0 0 142 0 0 0 0 0 0 0]
[0 0 0 146 0 0 0 0 0 0]
[0 0 0 0 145 0 0 0 0 0]
[0 0 0 0 0 144 0 0 0 1]
[0 0 0 0 0 0 145 0 0 0]
[0 0 0 0 0 0 0 143 0 0]

```
[ 0 1 0 0 0 0 0 0 0 138 0]
[ 0 0 0 0 2 0 0 0 0 1 141]]
```

solver: adam

activation: tanh

learning_rate: adaptive

score: 0.9361111111111111

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0  0]
 [ 0 146  0  0  0  0  0  0  0  0  0]
 [ 0  0 142  0  0  0  0  0  0  0  0]
 [ 0  0  0 146  0  0  0  0  0  0  0]
 [ 0  0  0  0 145  0  0  0  0  0  0]
 [ 0  0  0  0  0 145  0  0  0  0  0]
 [ 0  0  0  0  0  0 145  0  0  0  0]
 [ 0  0  0  0  0  0  0 143  0  0  0]
 [ 0  0  0  0  0  0  0  0 139  0  0]
 [ 0  0  0  0  0  0  0  0  0 144  0]]
```

solver: adam

activation: relu

learning_rate: constant

score: 0.9361111111111111

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0  0]
 [ 0 146  0  0  0  0  0  0  0  0  0]
 [ 0  0 142  0  0  0  0  0  0  0  0]
 [ 0  0  0 146  0  0  0  0  0  0  0]
 [ 0  0  0  0 145  0  0  0  0  0  0]
 [ 0  0  0  0  0 145  0  0  0  0  0]
 [ 0  0  0  0  0  0 145  0  0  0  0]
 [ 0  0  0  0  0  0  0 143  0  0  0]
 [ 0  0  0  0  0  0  0  0 139  0  0]
 [ 0  0  0  0  0  0  0  0  0 144  0]]
```

solver: adam

activation: relu

learning_rate: invscaling

score: 0.9555555555555556

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0  0]
 [ 0 146  0  0  0  0  0  0  0  0  0]
 [ 0  0 142  0  0  0  0  0  0  0  0]]
```

```
[ 0 0 0 146 0 0 0 0 0 0]
[ 0 0 0 0 145 0 0 0 0 0]
[ 0 0 0 0 0 145 0 0 0 0]
[ 0 0 0 0 0 0 145 0 0 0]
[ 0 0 0 0 0 0 0 143 0 0]
[ 0 0 0 0 0 0 0 0 139 0]
[ 0 0 0 0 0 0 0 0 0 144]]
```

```
confusion_matrix: [[142  0  0  0  0  0  0  0  0  0]
[ 0 146  0  0  0  0  0  0  0  0]
[ 0  0 142  0  0  0  0  0  0  0]
[ 0  0  0 146  0  0  0  0  0  0]
[ 0  0  0  0 145  0  0  0  0  0]
[ 0  0  0  0  0 145  0  0  0  0]
[ 0  0  0  0  0  0 145  0  0  0]
[ 0  0  0  0  0  0  0 143  0  0]
[ 0  0  0  0  0  0  0  0 139  0]
[ 0  0  0  0  0  0  0  0  0 144]]
```

Wnioski: najgorsze wyniki osiągnięto dla solvera SDG, activation: tanh,
learning_rate: invscaling, oraz solver: sgd, activation: relu, learning_rate:
invscaling, score: 0.25555555555555554, najlepsze dla konfiguracji solver:
adam
activation: relu
learning_rate: adaptive
score: 0.9638888888888889

Wyniki zadania 4:

wszystkie szczegółowe wyniki znajdują się w plikach w repozytorium
najlepszy wynik:
solver: adam
activation: tanh
learning_rate: adaptive
score: 0.6397306397306397

```
confusion_matrix train: [[236 27 98 0 1 2 5 0 1 0]
[ 53 112 17 1 0 6 6 0 0 0]
[ 98 20 212 0 0 1 12 0 0 0]
[ 0 2 0 27 5 0 1 0 0 0]
[ 1 2 0 3 19 3 0 0 0 0]
[ 7 1 2 5 4 16 6 0 0 0]
[ 7 5 8 1 0 0 110 0 0 0]
[ 11 1 4 0 2 1 5 0 0 0]
[ 4 1 0 1 0 1 0 0 9 0]
[ 0 0 0 0 0 0 0 0 0 4]]
confusion_matrix test: [[55 8 30 0 0 0 0 0 0 0]
[ 6 34 4 1 0 0 2 0 2 0]
[23 5 55 0 0 1 2 0 0 0]
[ 0 0 0 5 2 2 0 0 0 0]
[ 2 0 2 0 3 0 0 0 0 0]
[ 0 2 1 2 2 3 0 0 0 0]
[ 0 0 5 0 0 0 27 0 0 0]
[ 3 1 0 0 0 0 2 0 0 0]
[ 2 0 0 0 0 0 0 0 2 0]
[ 0 0 0 0 0 0 0 0 0 1]]
```

Wnioski dokładność na poziomie 64 % przy 8 klasach jest bardzo dobrą dokładnością.

Link do repozytorium z kodem:

https://github.com/KamilPyla/MIO_2023/tree/master/lab_02