

GEN_Olaf_Werner_291139_sprawozdanie

Olaf Werner

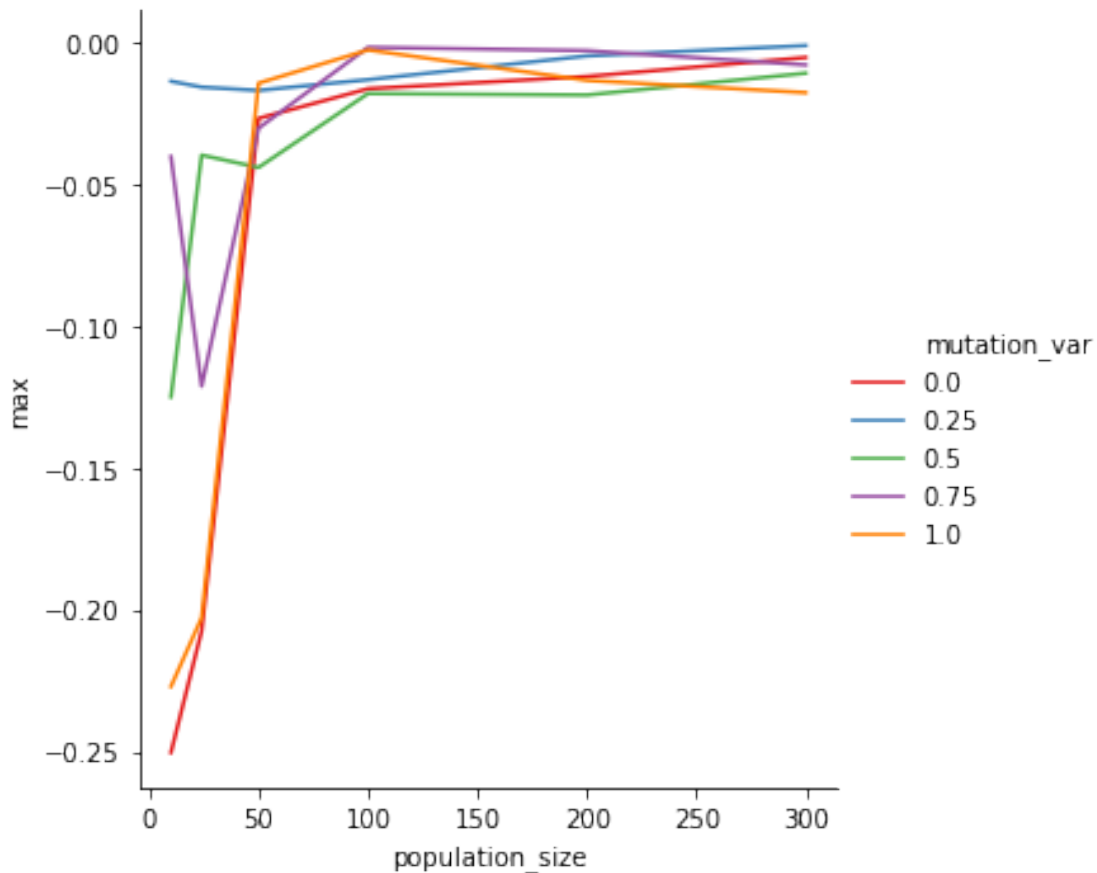
12 czerwca 2020

- 1 Potwierdzam samodzielność powyższej pracy oraz niekorzystanie przeze mnie z niedozwolonych źródeł. Olaf Werner¶¶**
- 2 Zadanie 1 - Minimalizacja funkcji z zastosowaniem algorytmu genetycznego**
 - 2.1 Pierwsza funkcja do dopasowania**
 - 2.2 Druga funkcja do dopasowania**
 - 2.3 domyślnie maksymalizujemy funkcje dopasowanie, dlatego powyższe funkcje są pomnożone przez -1**
 - 2.4 algorytm genetyczny z mutacją gaussowską i krzyżowaniem jednopunktowym**

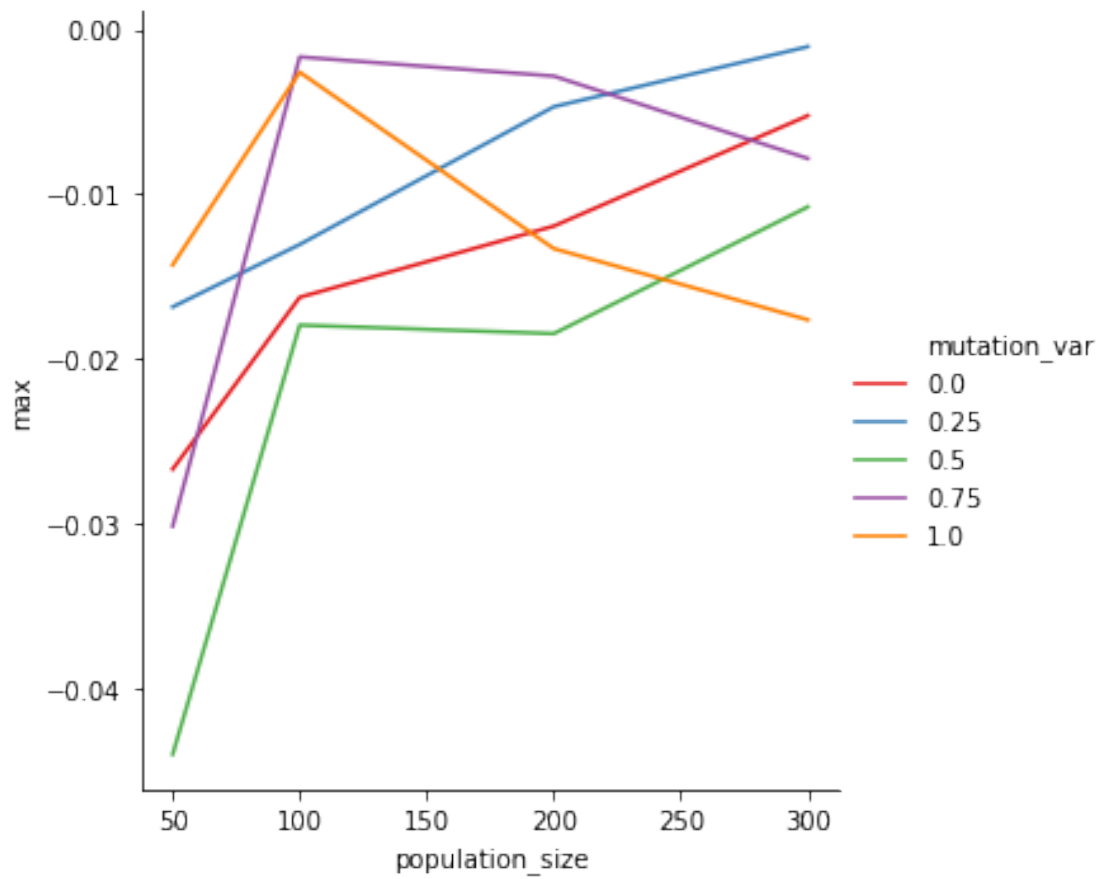
Prawdopodobieństwo rodzicielstwa zależy od softmaxu funkcji dostosowania populacji. Będziemy mieli 100 generacji. I testowaliśmy dla rozmiarów populacji ze zbioru [10,24,50,100,200,300] i wariancji mutacji ze zbioru [0,0.25,0.5,0.75,1]. Rozmiar populacji w każdej generacji jest stały. Początkowy genotyp jest losowany jednostajnie z przedziału [-1,1].

3 $\text{fit1 } x^2 + y^2 + 2z^2$

- 3.1 generujemy statystyki podsumowujące każdą generację w zależności od wariacji mutowania gaussowskiego i rozmiaru populacji
- 3.2 dostosowanie najlepszego osobnika w zależności od rozmiaru populacji i wariacji mutowania w ostatniej(100) generacji



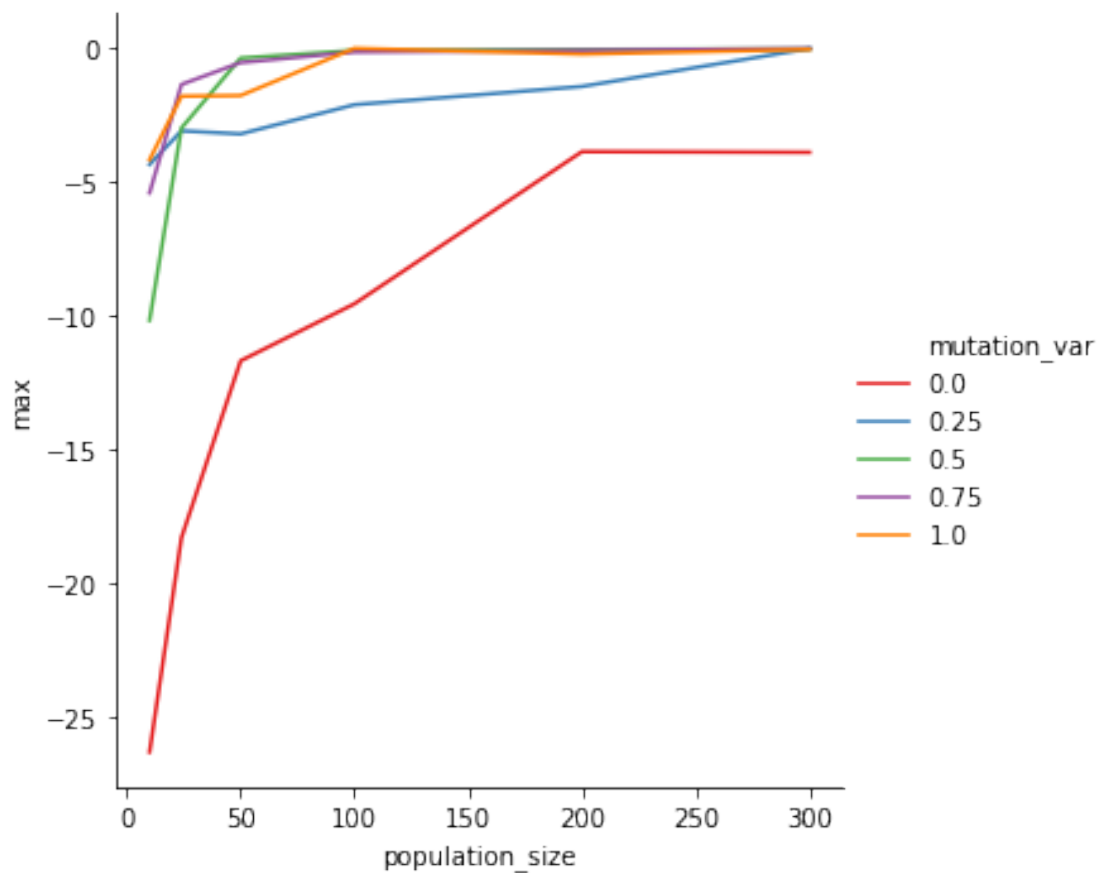
Jak widzimy dla małego rozmiar populacji wysoka wariancja daje bardzo złe rezultaty i samym mechanizmem krzyżowania też się nie da wiele osiągnąć, dla populacji większej od 24 rezultaty stają się bardziej sensowne



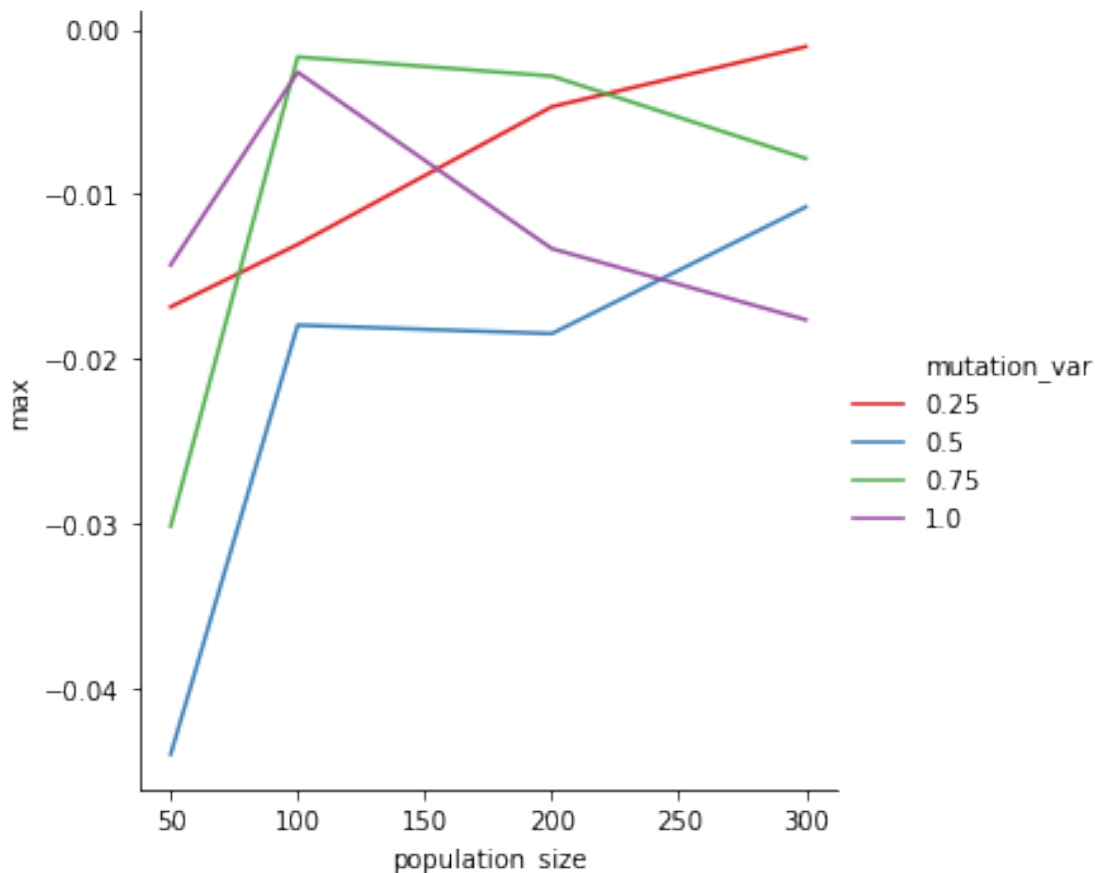
najlepiej okazało się dać małą mutacyjność i dużą populację, co ciekawe nawet używając tylko mechanizmu krzyżowania można osiągnąć dobre rezultaty.

4 fit2 Rastringa

- 4.1 generujemy statystyki podsumowujące każdą generację w zależności od wariacji mutowania gaussowskiego i rozmiaru populacji
- 4.2 dostosowanie najlepszego osobnika w zależności od rozmiaru populacji i wariacji mutowania w ostatniej(100) generacji



Bez mechanizmu mutacji nie da się osiągnąć dobrych rezultatów i też trzeba większej populacji



Ponownie najlepsza okazała się mała mutacyjność i duża populacja, trzeba jednak zaznaczyć że duża mutacyjność może na 3 krotnie mniejszej populacji osiągnąć zbliżone rezultaty

5 Wnioski

Dobór mutacyjności i rozmiaru populacji jest kluczowy. By osiągnąć najdokładniejsze rezultaty najlepiej dać dużą populację i małą mutacyjność. Za mała populacja daje złe rezultaty dla każdej mutacyjności. Duża mutacyjność pozwala osiągnąć lepsze rezultaty przy mniejszej populacji niż w przypadku niskiej mutacyjności. Za duża populacja może spowalniać uczenie.

6 Zadanie 2 - Wypełnianie koła prostokątami

7 algorytm

Prawdopodobieństwo rodzicielstwa zależy to stosunek dostosowania do sumy dostosowania całej populacji. Będziemy mieli 70 generacji. I testowaliśmy dla rozmiarów populacji 100 i wariancji

mutacji równej pierwiastkowi z promienia. Rozmiar populacji w każdej generacji jest stały. Dokonujemy mutacji w ilości $1/4$ wszystkich prostokątów dla danego osobnika. Pojedynczy gen może być mutowany wielokrotnie. Początkowy genotyp jest losowany jednostajnie z przedziału $[-R, R]$ gdzie R to rozmiar koła. Genotyp koduje położenie środka każdego prostokąta, jego aktywność i rodzaj prostokąta. Jeśli prostokąt nie jest aktywny to nie liczy się do funkcji celu. Prostokąt może być aktywny tylko wtedy gdy nie pokrywa się z innymi aktywnymi prostokątami, ani nie wychodzi poza koło.

Ponieważ prostokąt może być wybrany dowolną ilość razy to po prostu genotyp będzie miał zmienną długość i przechowywał informację o rodzaju wybranego prostokąta. W każdej generacji dodajemy nowy prostokąt. Krzyżowanie polega na wzięciu prostokątów o współrzędnej x/y większej od jednego rodzica, mniejszej od drugiego rodzica i wymieszaniu.

Jak prostokąt wychodzi poza koło to się go daje z powrotem do koła ("odbija się"). Jak prostokąt aktywny zderzy się z innym prostokątem aktywnym to ustawia się na styk do prostokąta z którym kolidował i prostokąt o mniejszym stosunku wartości do pola staje się nieaktywny. Legalność aktywności i sprawdzanie czy coś może stać się aktywne jest sprawdzane w losowej kolejności. Nieaktywne prostokąty są rozpychane w następujący sposób w losowej kolejności wybiera się nieaktywne prostokąty po czy od wybranego prostokąta odsuwa się prostokąty sąsiadujące z nim.

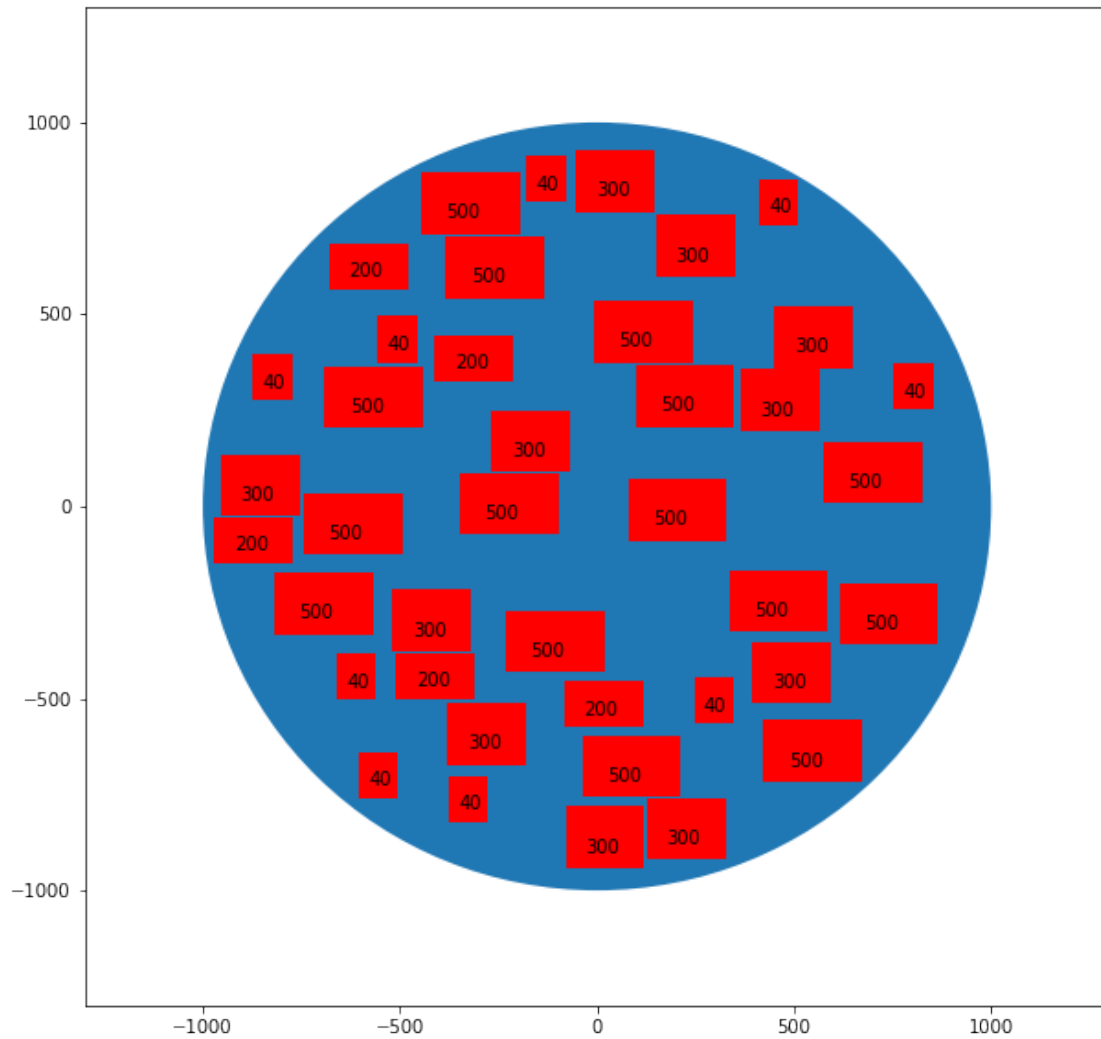
7.1 Początkowe ustawienia: można zwiększać POP i GEN, by osiągać lepsze rezultaty

populacja startowa: 100

ilość generacji: 70

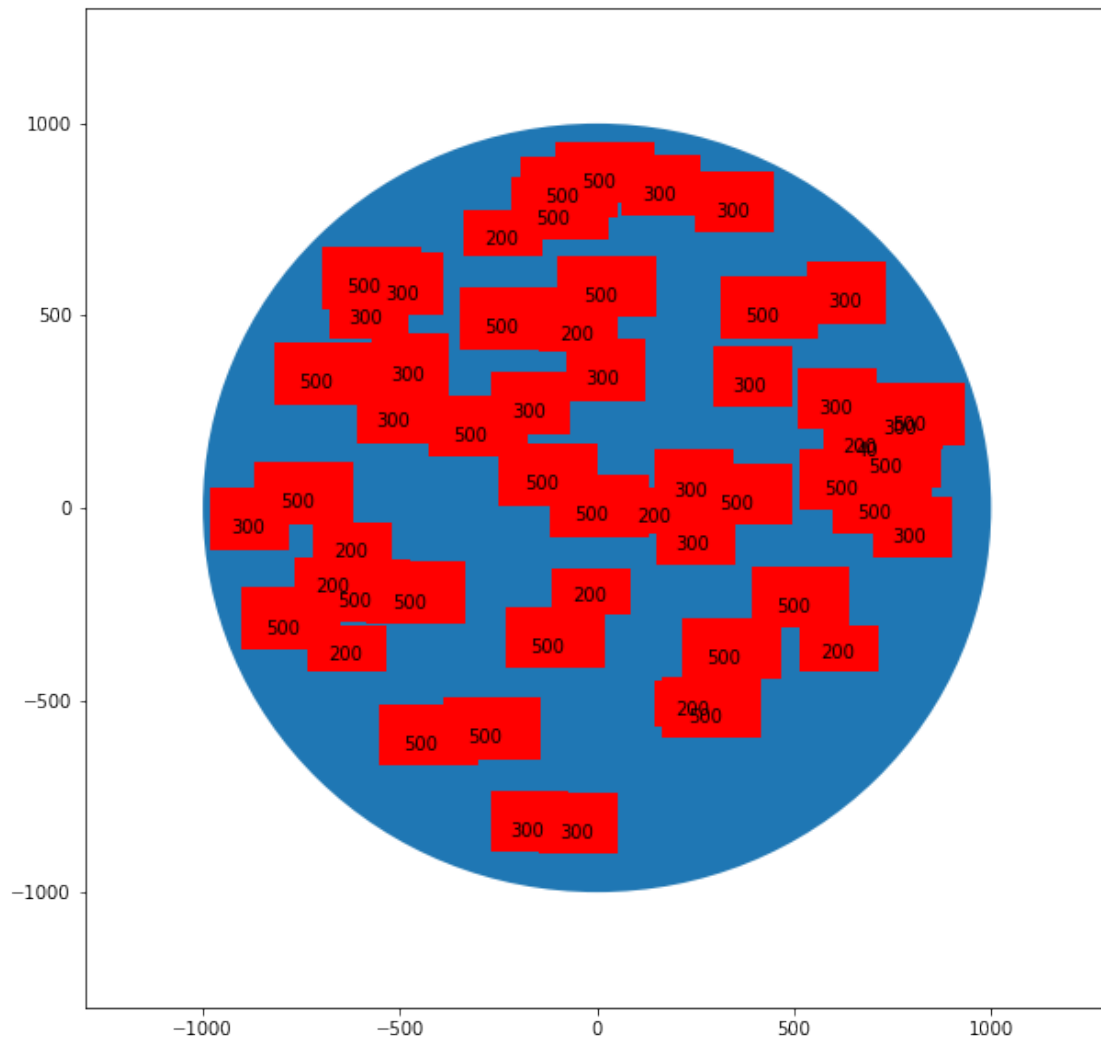
8 r1000

9 Active



Rozwiązanie nie jest doskonałe. Mamy dziury, ale nie widać jakiegoś szczególnie pustego obszaru. Widzimy że prostokąty z dużymi wartościami są preferowane.

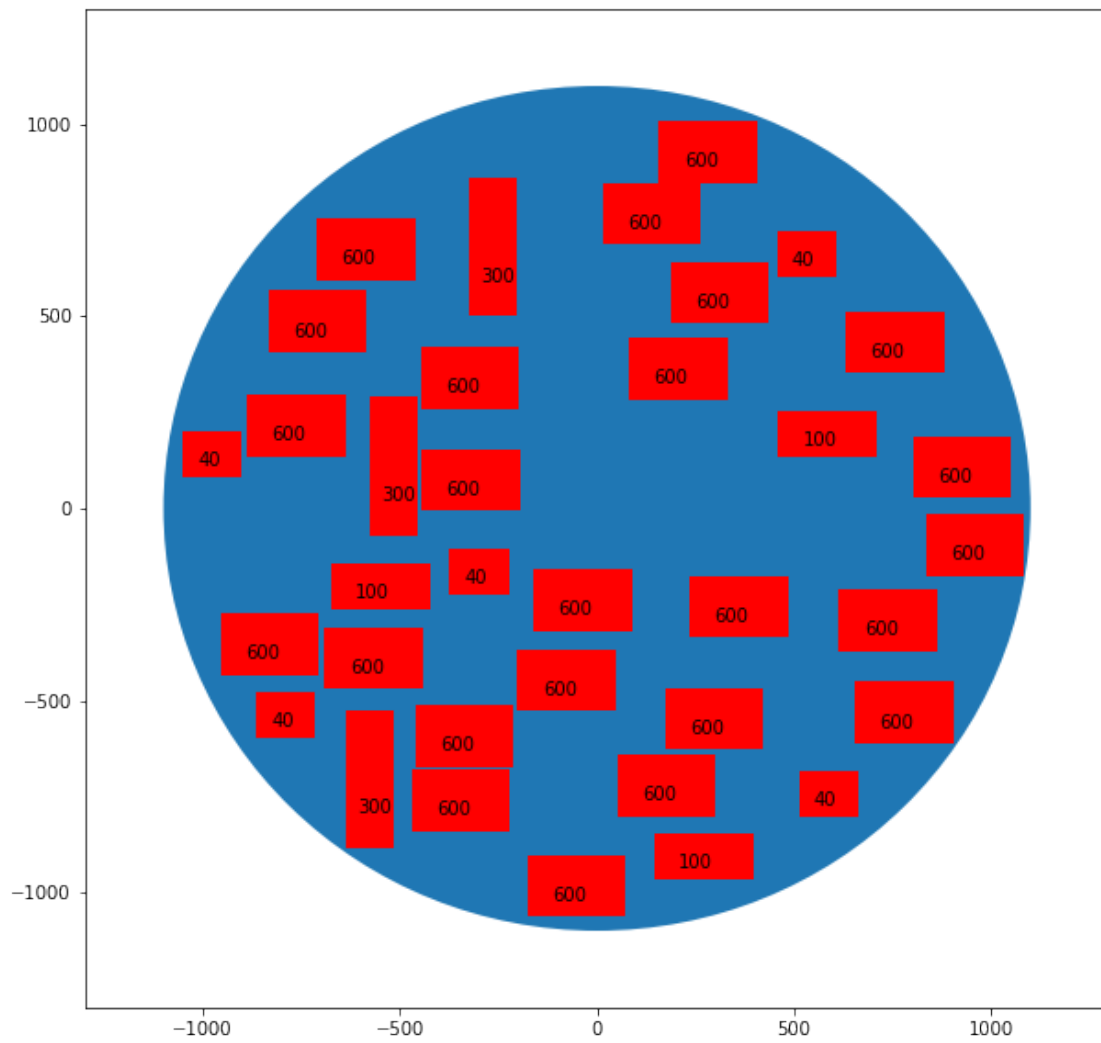
10 Not Active



Widzimy że nieaktywne prostokąty nie są równomiernie rozłożone po kole i stąd wynikają dziury w wykresie poprzednim wykresie.

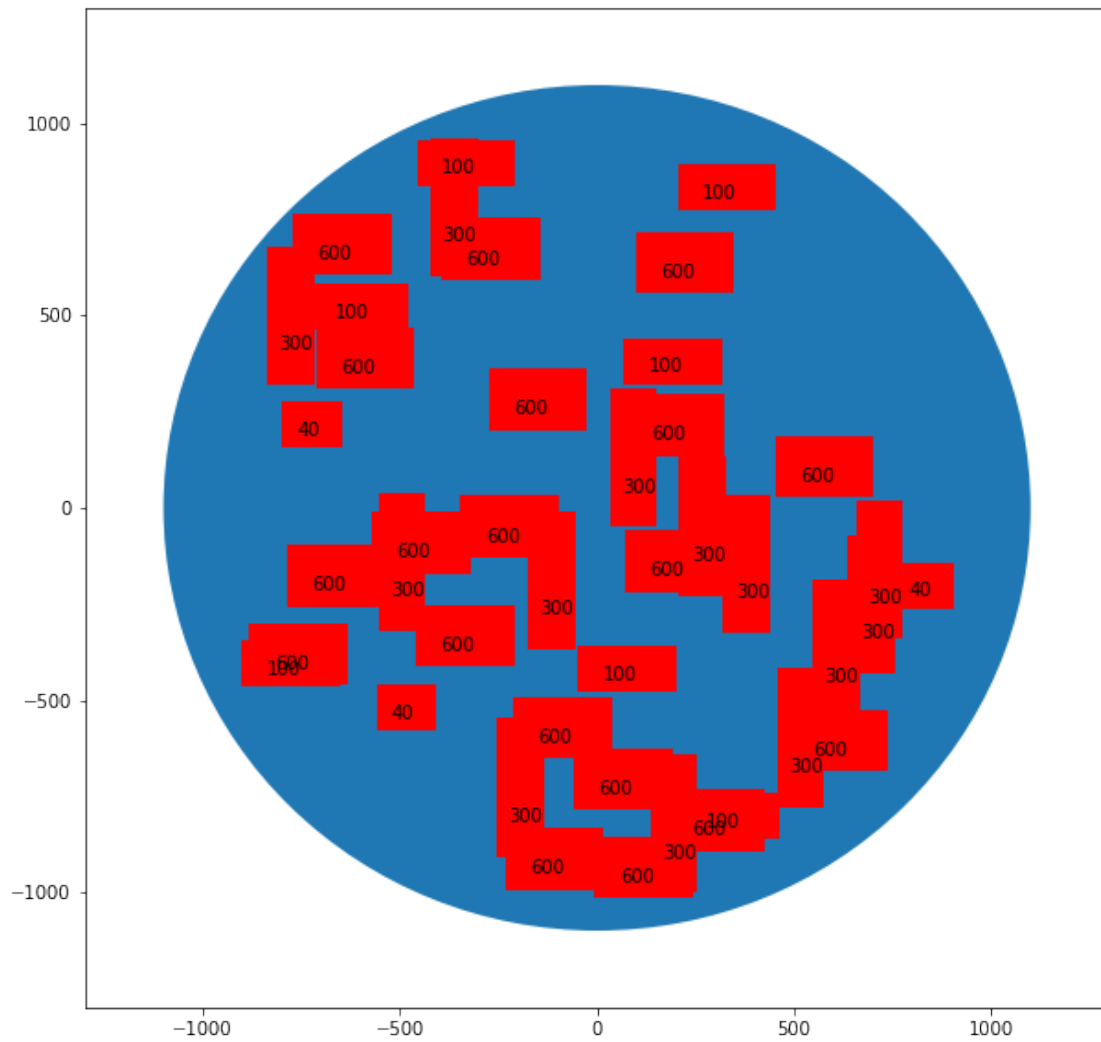
11 r1100

12 Active



Też mamy dziurę.

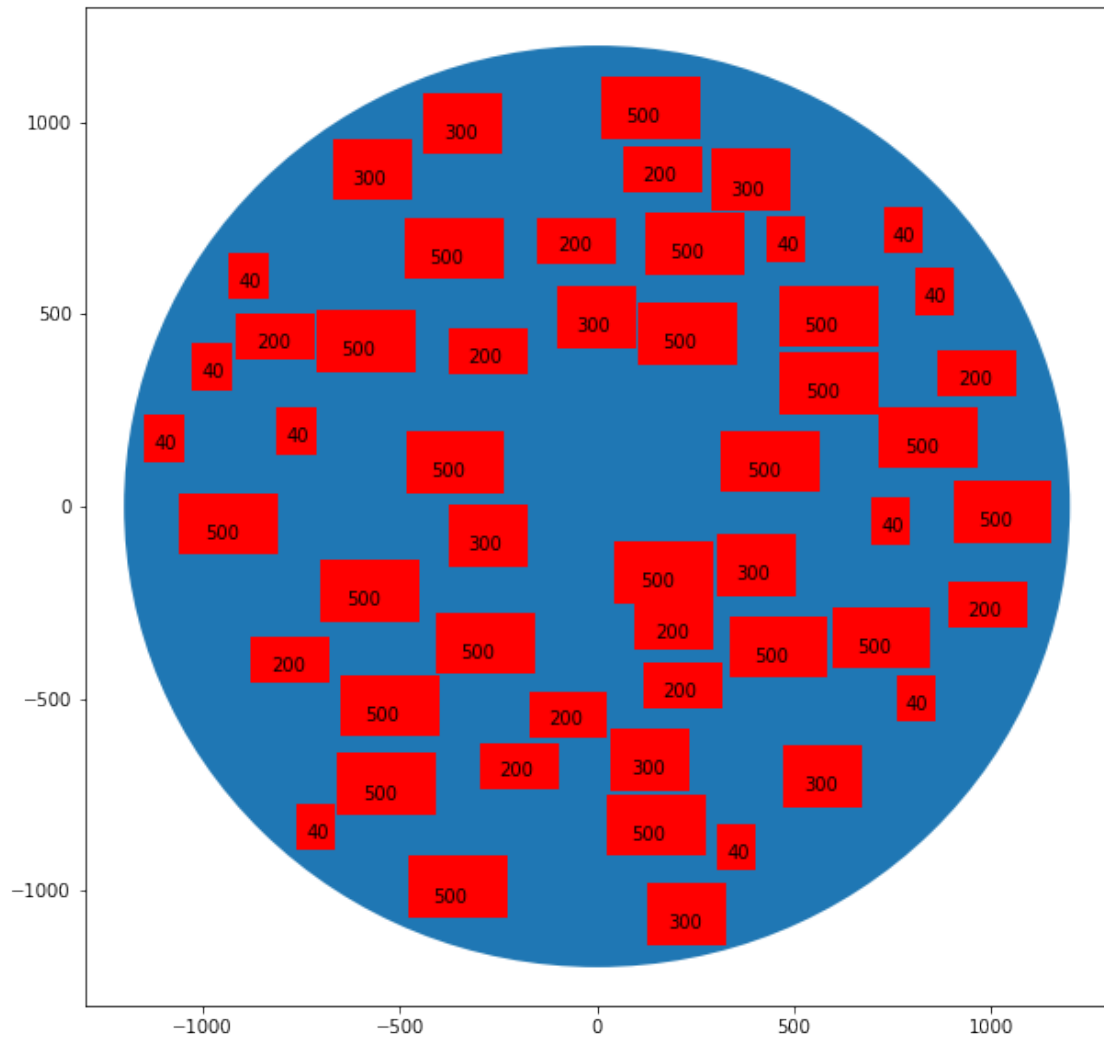
13 Not Active



Podobnie jak wcześniej nieaktywne prstokąty są zgrupowane w "klastry".

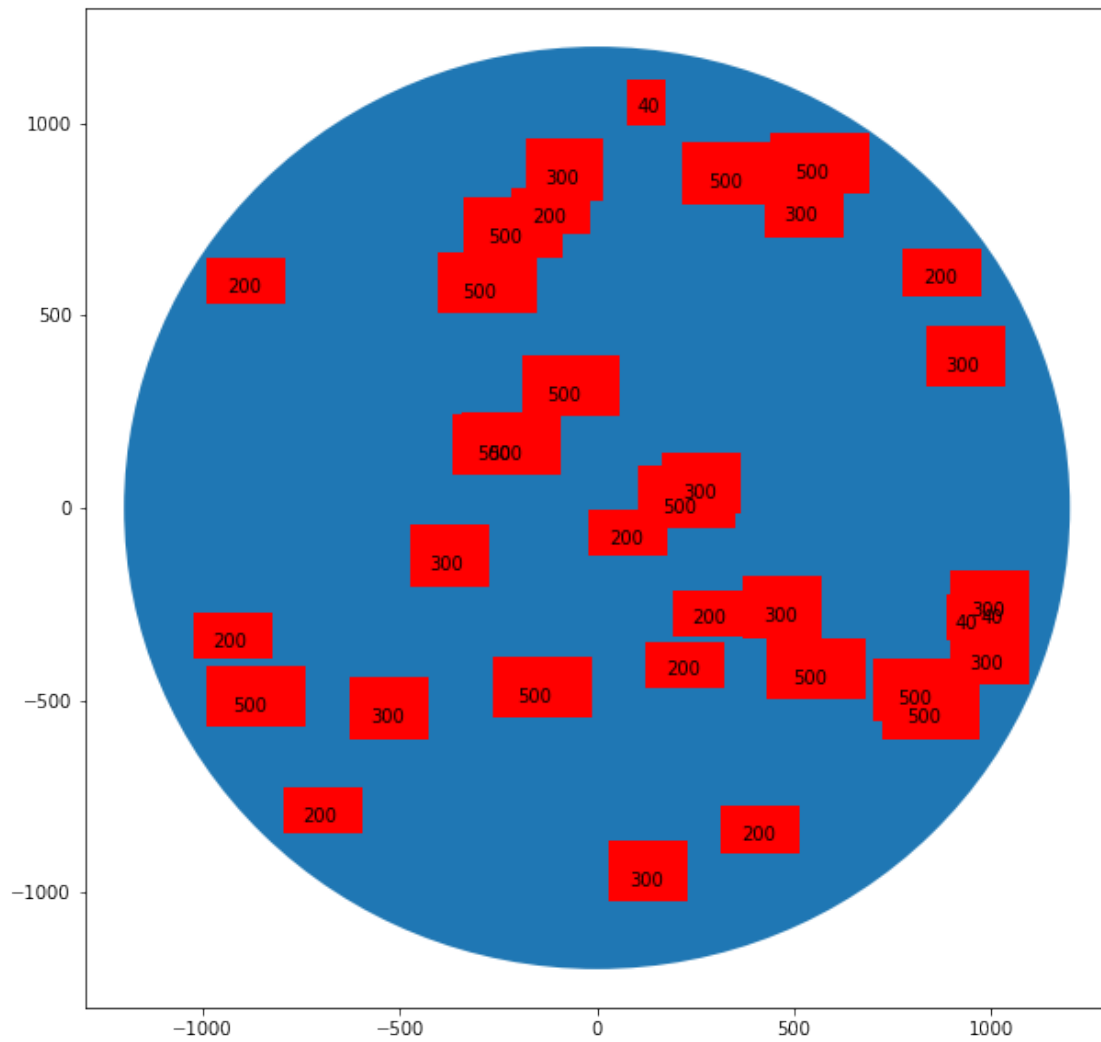
14 r1200

15 Active



Widzimy dziurę na środku.

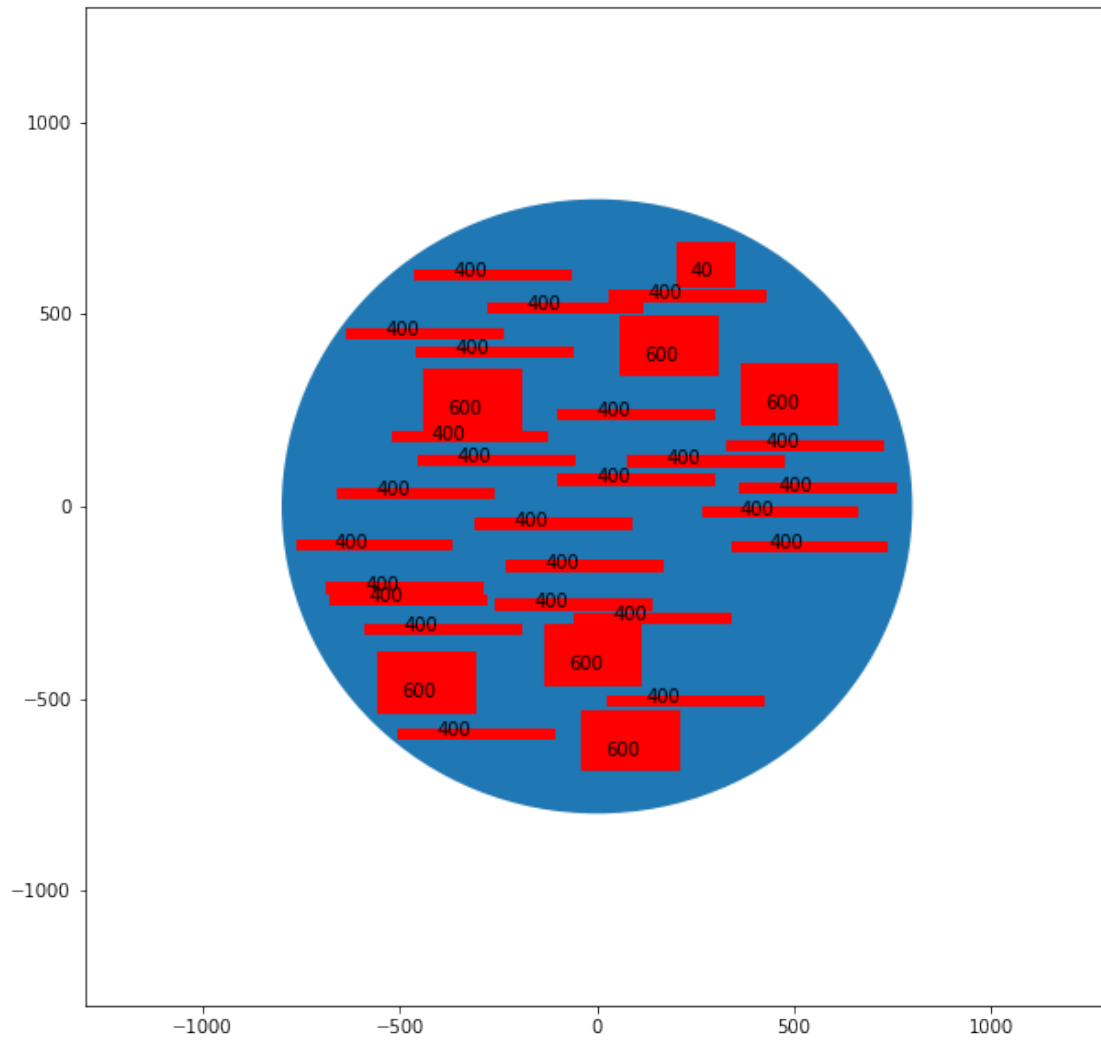
16 Not Active



Też są klastry, ale ogólnie prostokątów jest mało.

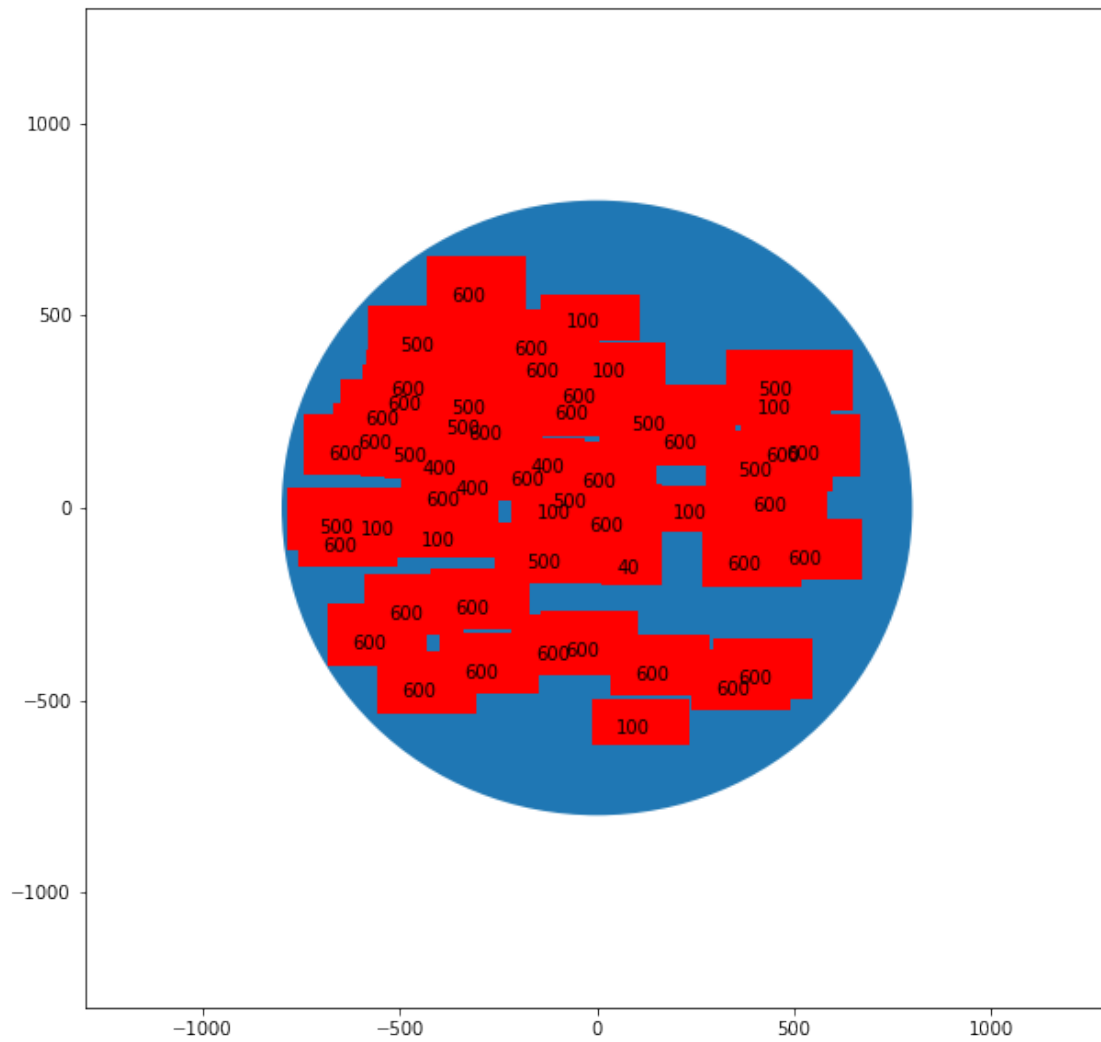
17 r800

18 Active



Wielokrotnie użyto prostokątów o wartości 400 bo łatwiej się mieściły i mają dobry stosunek wartości do pola.

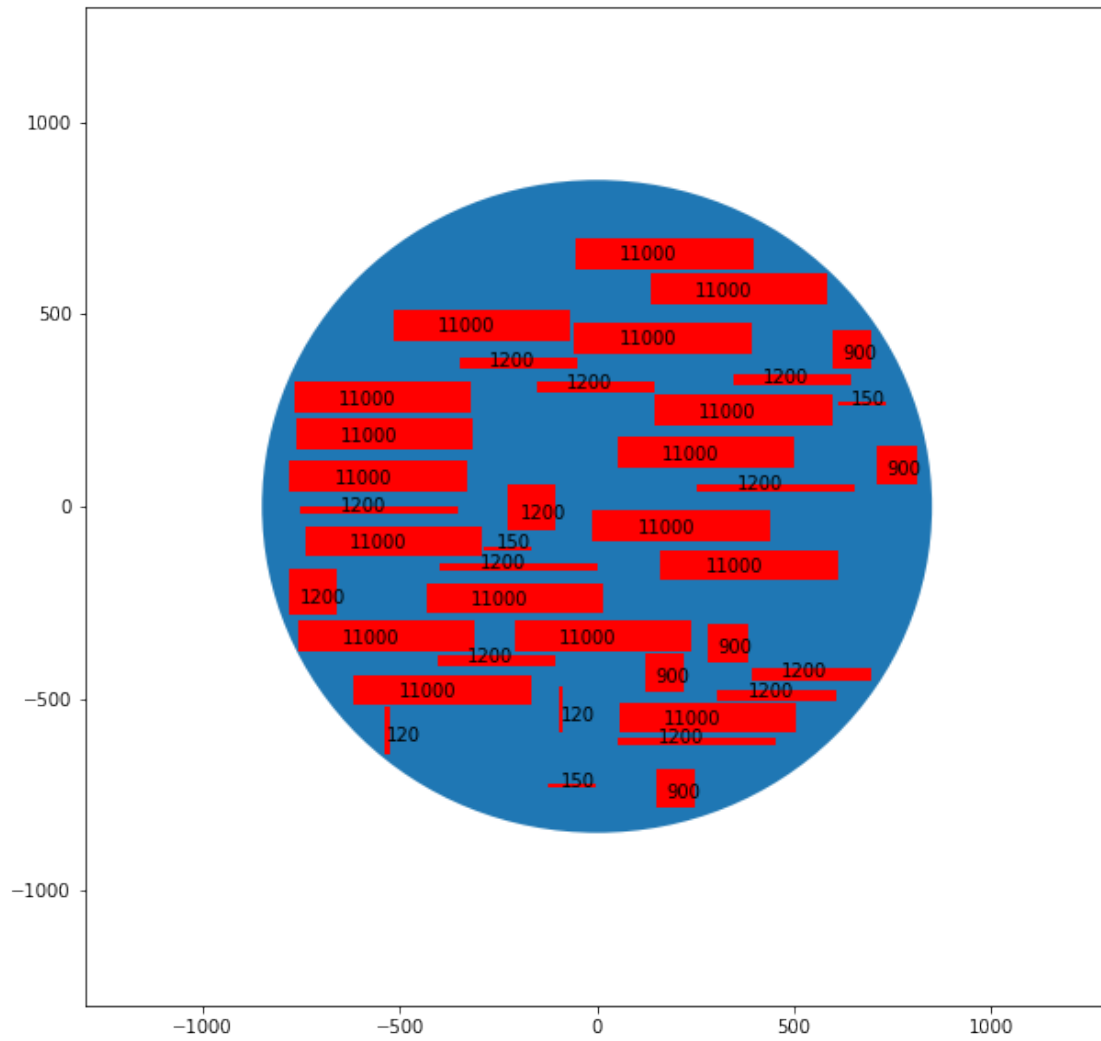
19 Not Active



Widzimy że algorytm próbował dodawać prostokąty o dużych wartościach.

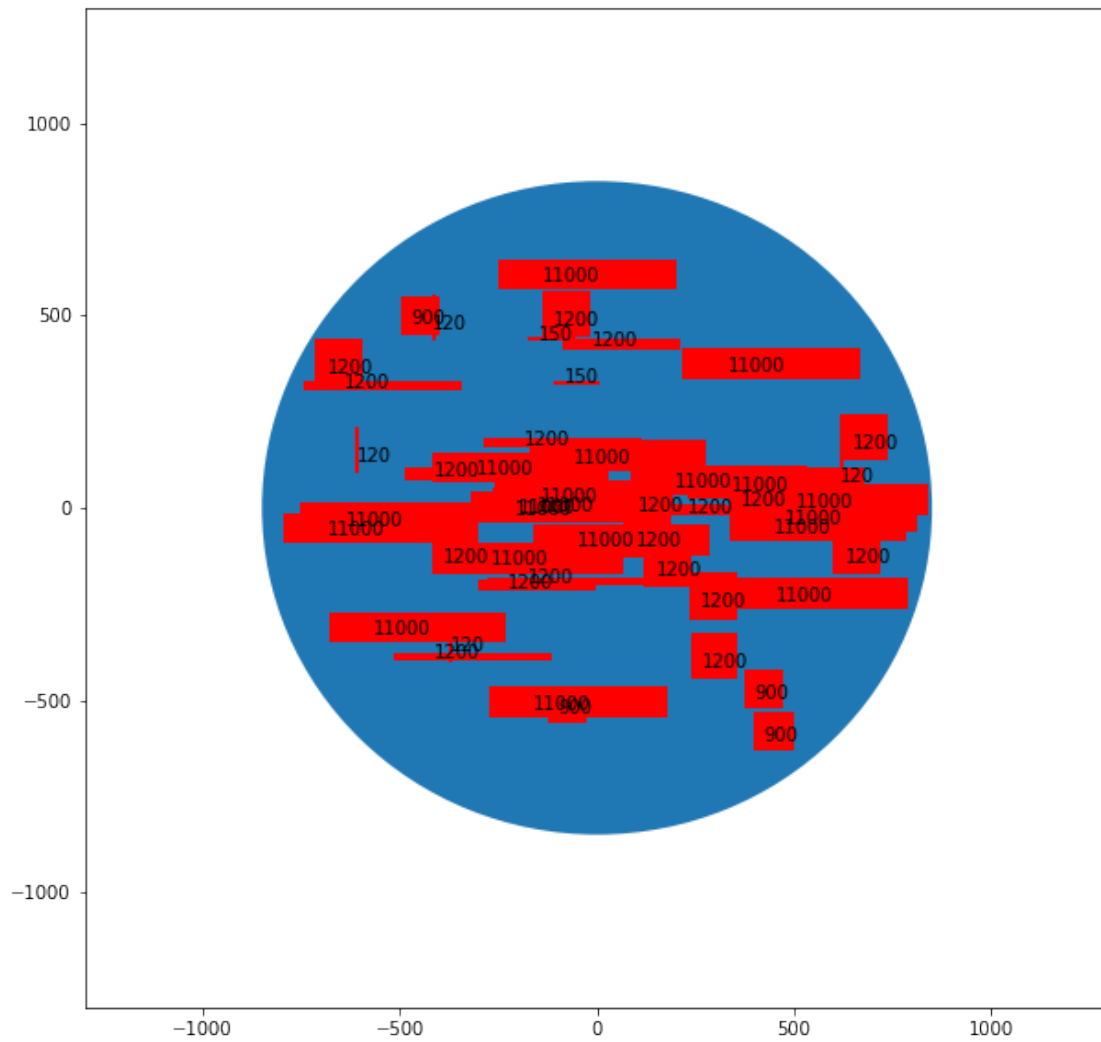
20 r850

21 Active



Tutaj widzimy że duże prostokąty są preferowane.

22 Not Active



Ponieważ prostokąty są szerokie, ale nie wysokie to są słabiej rozpychane do góry i dołu.

23 Podsumowanie

Widzimy że nasz podstawowy algorytm dla populacji 100 i 70 generacjach nie radzi sobie dobrze. Możliwe że to dlatego że po prostu parametry modeli były za małe. Po za tym są problemy z dokładnym przesunięciem prostokątów w puste miejsce. Możliwym usprawnieniem jest dopracowanie rozpychania prostokątów i wykrywanie pustych miejsc.

24 Zadanie 3 NEAT

24.1 maksymalizujemy fit więc mnożymy przez -1

25 algorytm

Parametry algorytmu zostały wzięte z oryginalnego artykułu dotyczącego NEAT. Za populację startową będziemy mieli 150 osobników i będziemy mieli 30 generacji. Po 15 generacjach bez poprawy gatunek wymiera, ustala się gatunek na podstawie podobieństwa genomów $\sigma = \frac{c_1 D}{N} + \frac{c_2 E}{N} + c_3 \hat{W}$ gdzie $c_1 = 1, c_2 = 1, c_3 = 0.4$. Gatunki są różne gdy: $\sigma_t < 3$. Jest 80% szansy na zmianę wagi połączenia: na 90% jest to perturbacja normalna, a na 10% jest to przydzielenie losowej jednostajnie wagi z $[-1,1]$. Dla 25% nie używamy krzyżowania. Szansa na zmutowanie nowego połączenia to 5%, a nowego wierzchołka 3%. W gatunkach o rozmiarze większym niż 5 elita przechodzi automatycznie.

26 NeuroEvolution of Augmenting Topologies NEAT

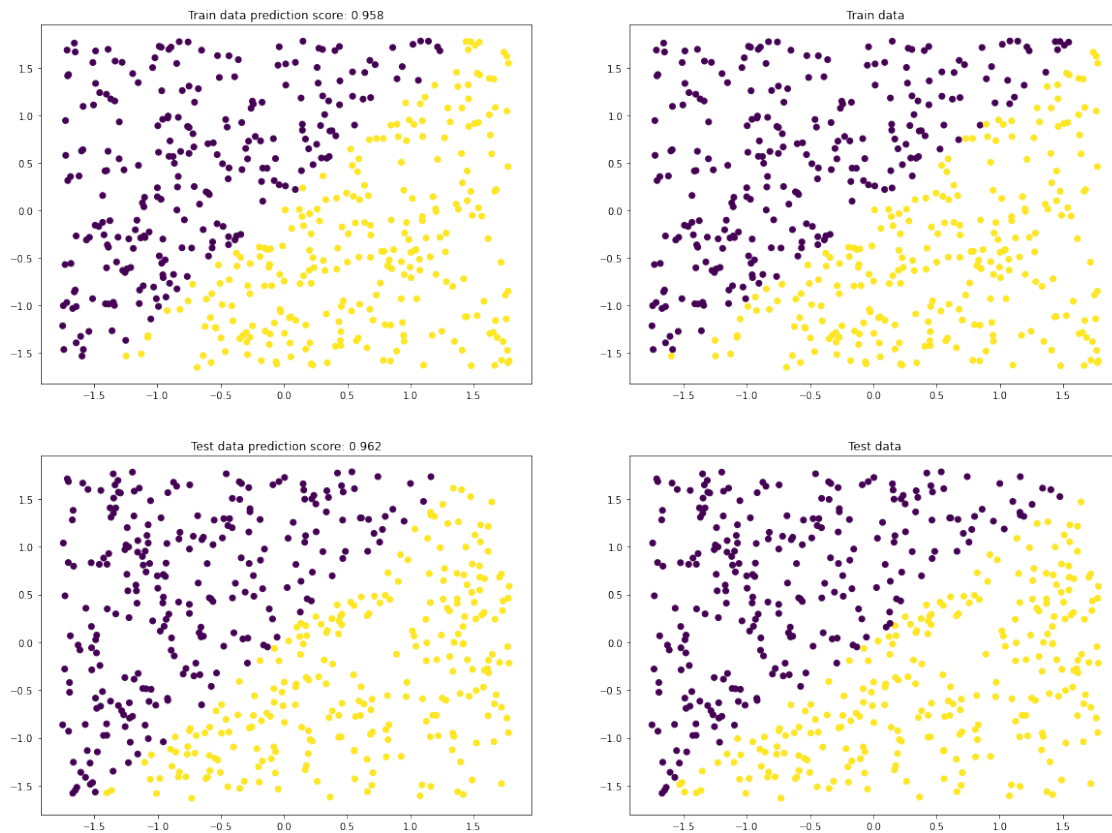
26.1 Początkowe ustawienia: można zwiększać POP i GEN, by osiągać lepsze rezultaty

populacja startowa: 150

ilość generacji: 30

27 Test na easy

28 Rezultaty

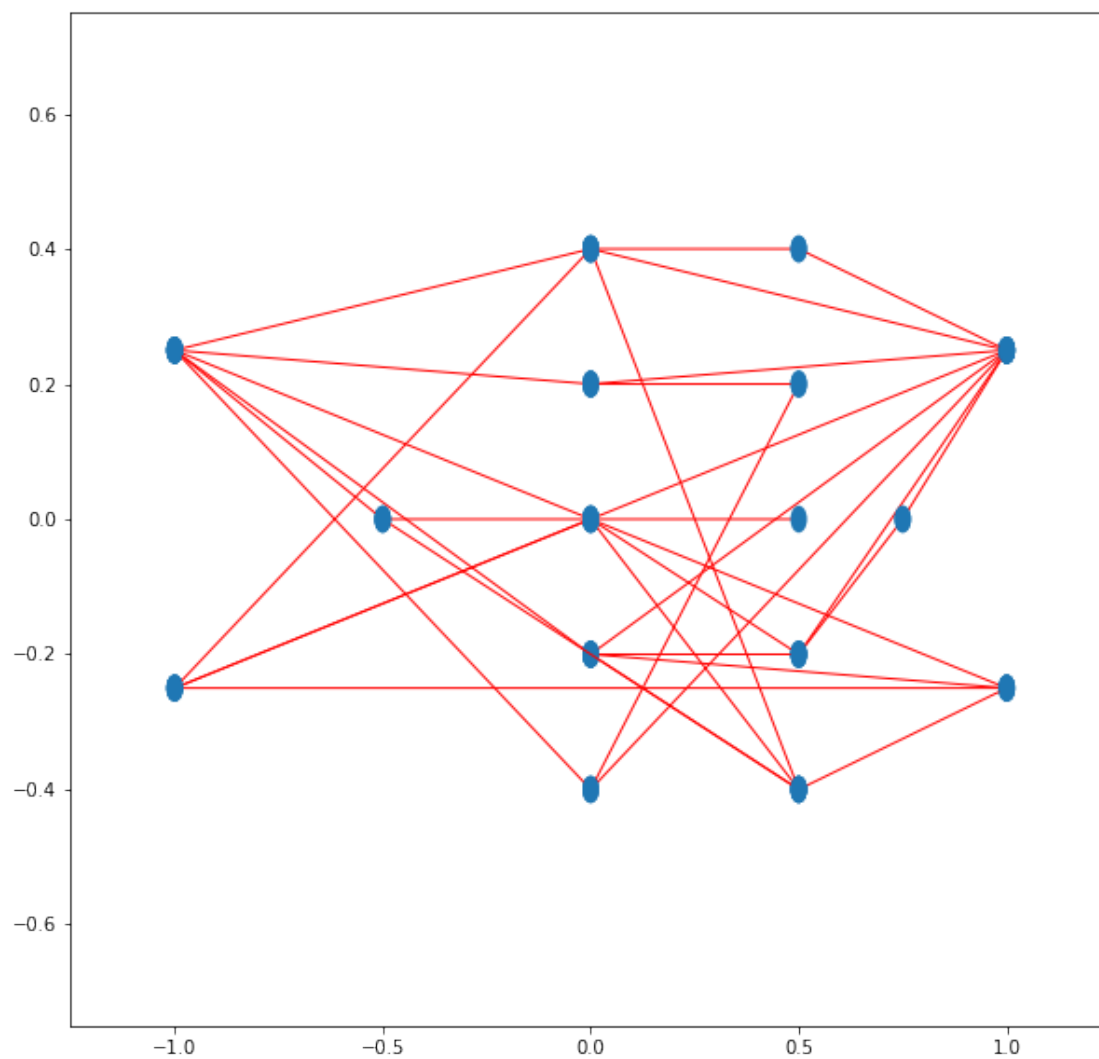


28.1 Podsumowanie

Otrzymano 96% na zbiorze testowym. Algorytm poradził sobie bez problemów. Otrzymano też dość małą sieć.

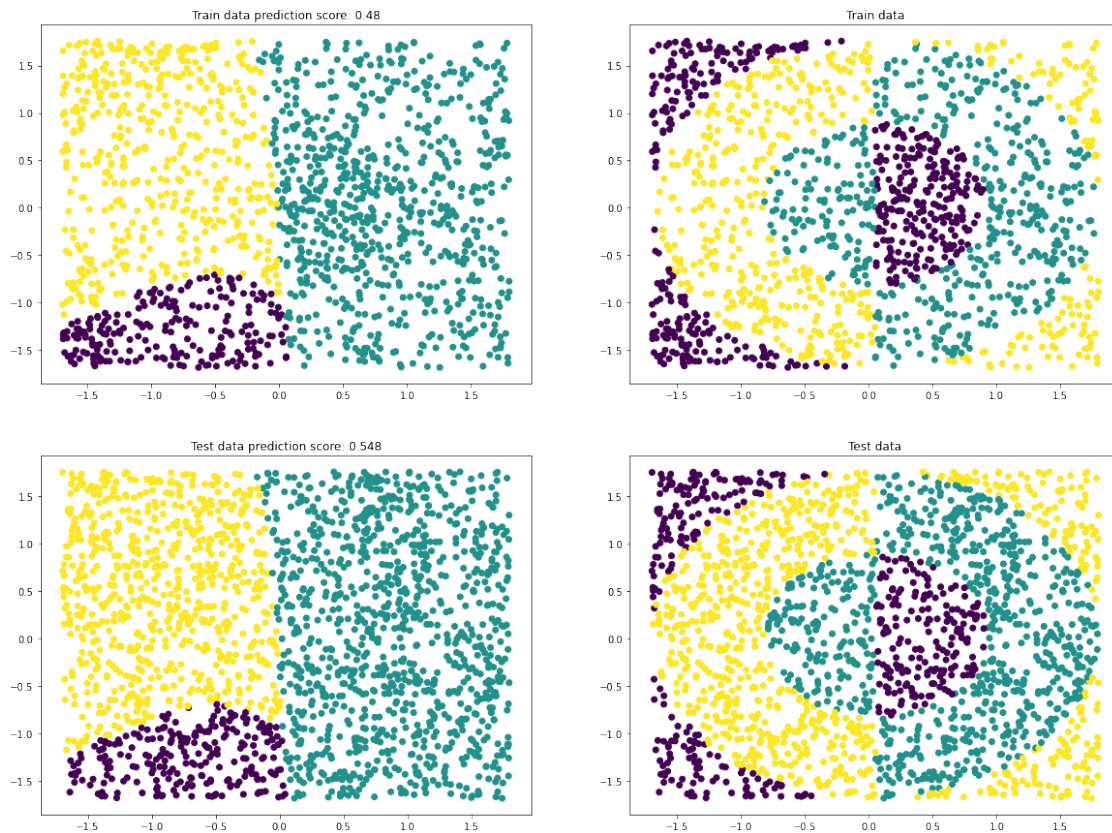
28.2 parametry najlepszej sieci (aktywne połączenia): 31

28.3 parametry najlepszej sieci (neurony):16



29 Test na rings-3-regular

30 Rezultaty

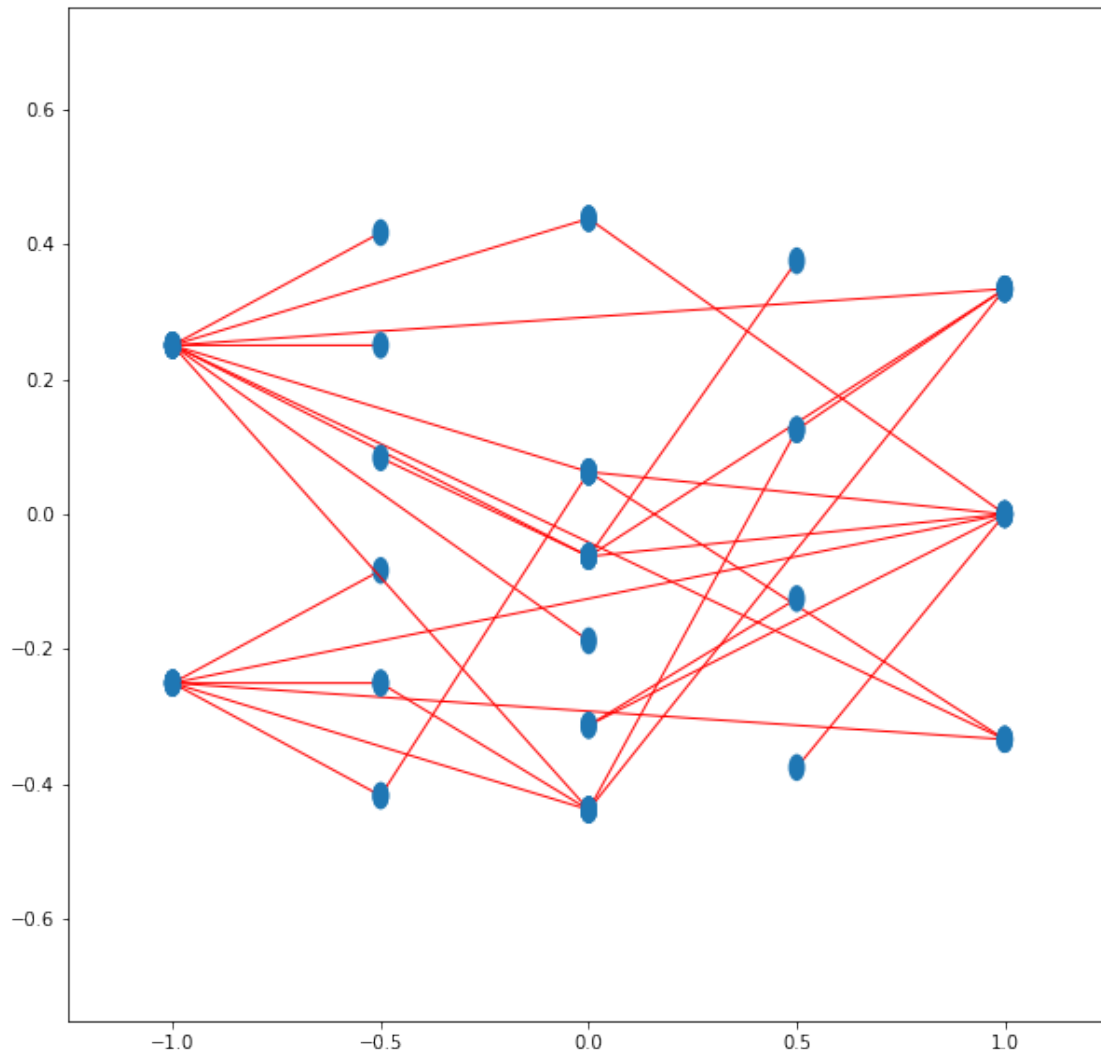


30.1 Podsumowanie

Otrzymano 55% na zbiorze testowym. Algorytm próbował separować dane liniowo. Sieć jest mała i niektóre neurony są bezużyteczne (nie prowadzą do ostatniej warstwy)

30.2 parametry najlepszej sieci (aktywne połączenia): 30

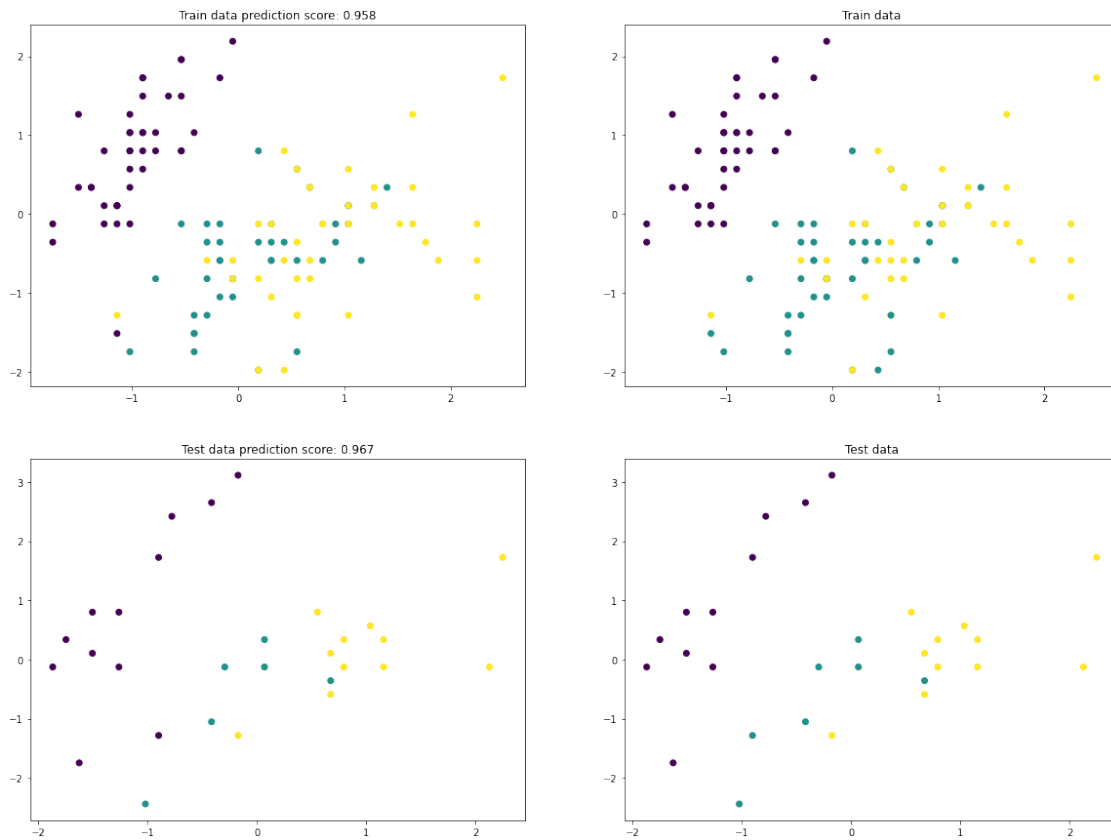
30.3 parametry najlepszej sieci (neurony): 23



31 Test na Iris

Podzielono dane na zbiór treningowy i testowy w stosunku 80:20

32 Rezultaty

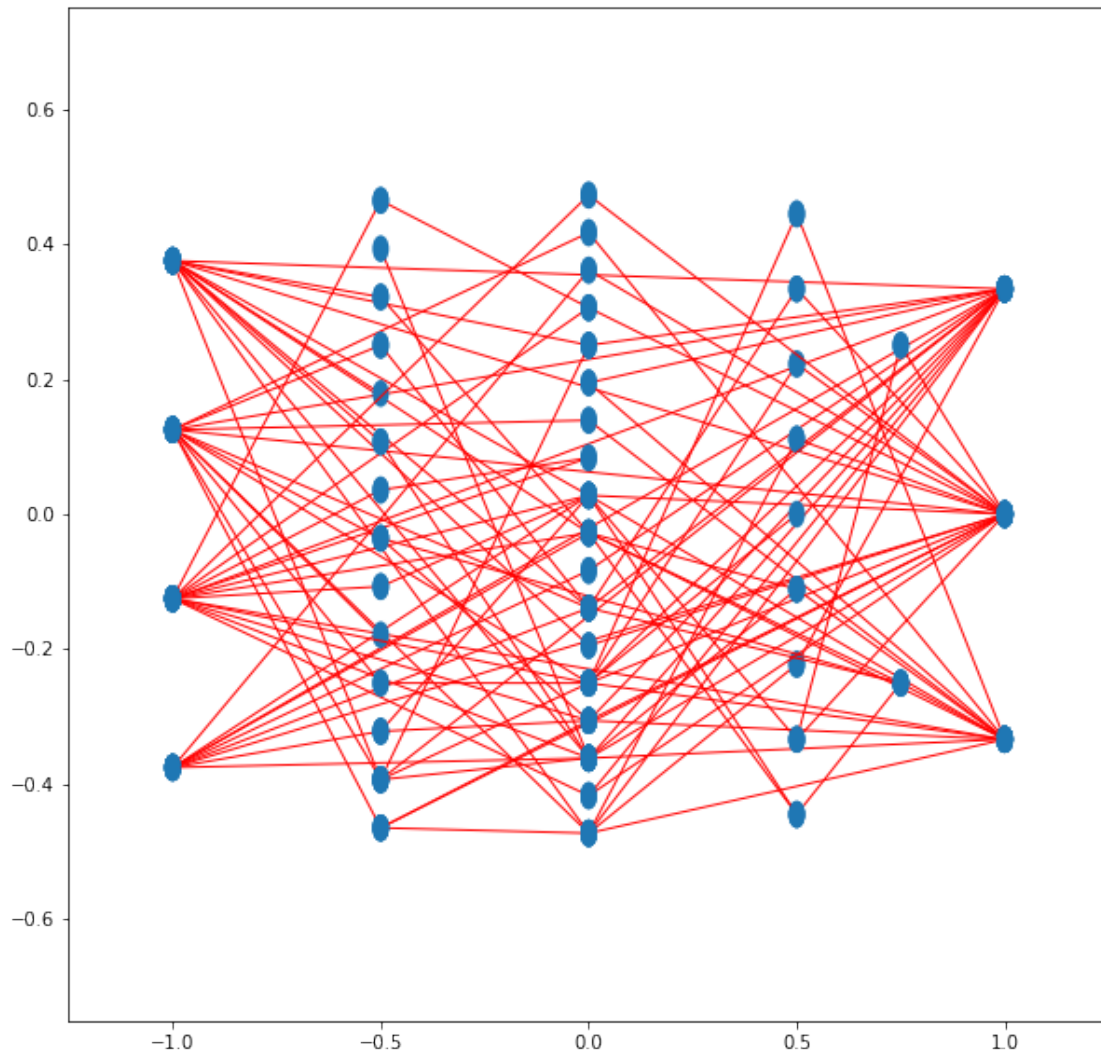


32.1 Podsumowanie

Otrzymaliśmy 97% na zbiorze treningowym. Sieć ma 3 rozwinięte warstwy.

32.2 parametry najlepszej sieci (aktywne połączenia): 119

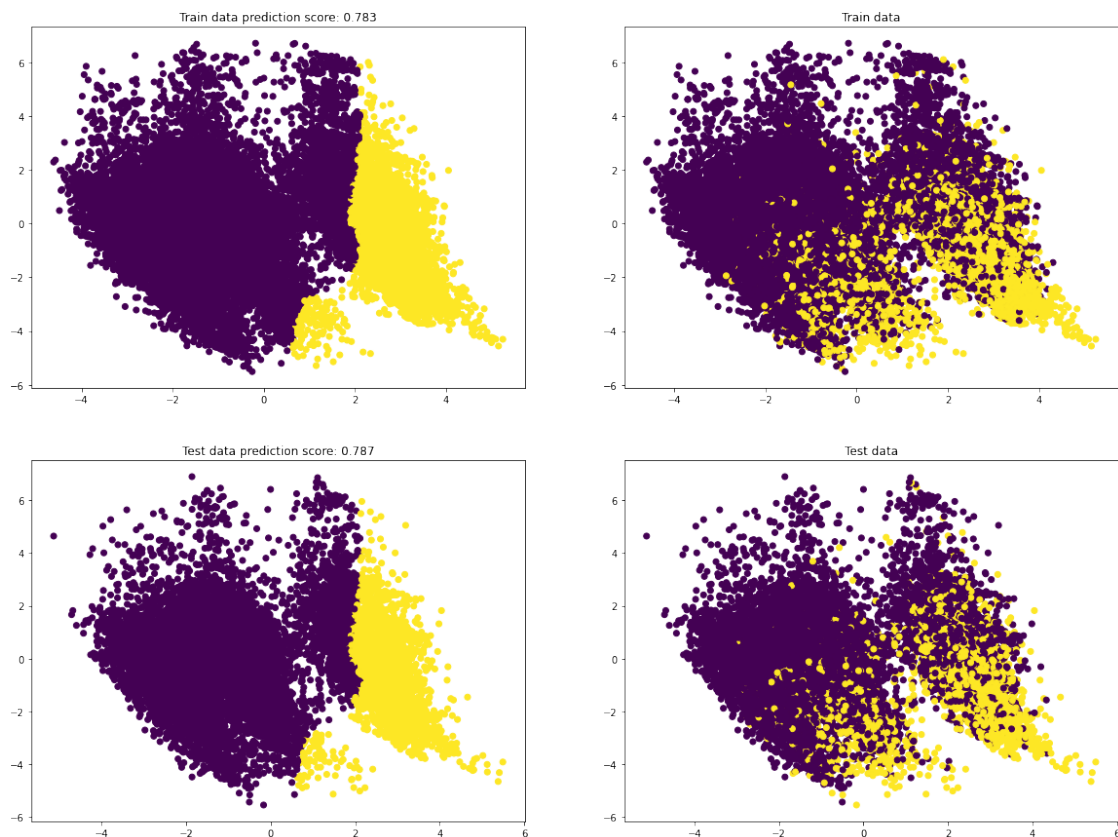
32.3 parametry najlepszej sieci (neurony): 50



33 Test na adult

By ograniczyć czas obliczeń i ułatwić zadanie sieci użyliśmy PCA i wzięliśmy 82 komponenty.

34 Rezultaty

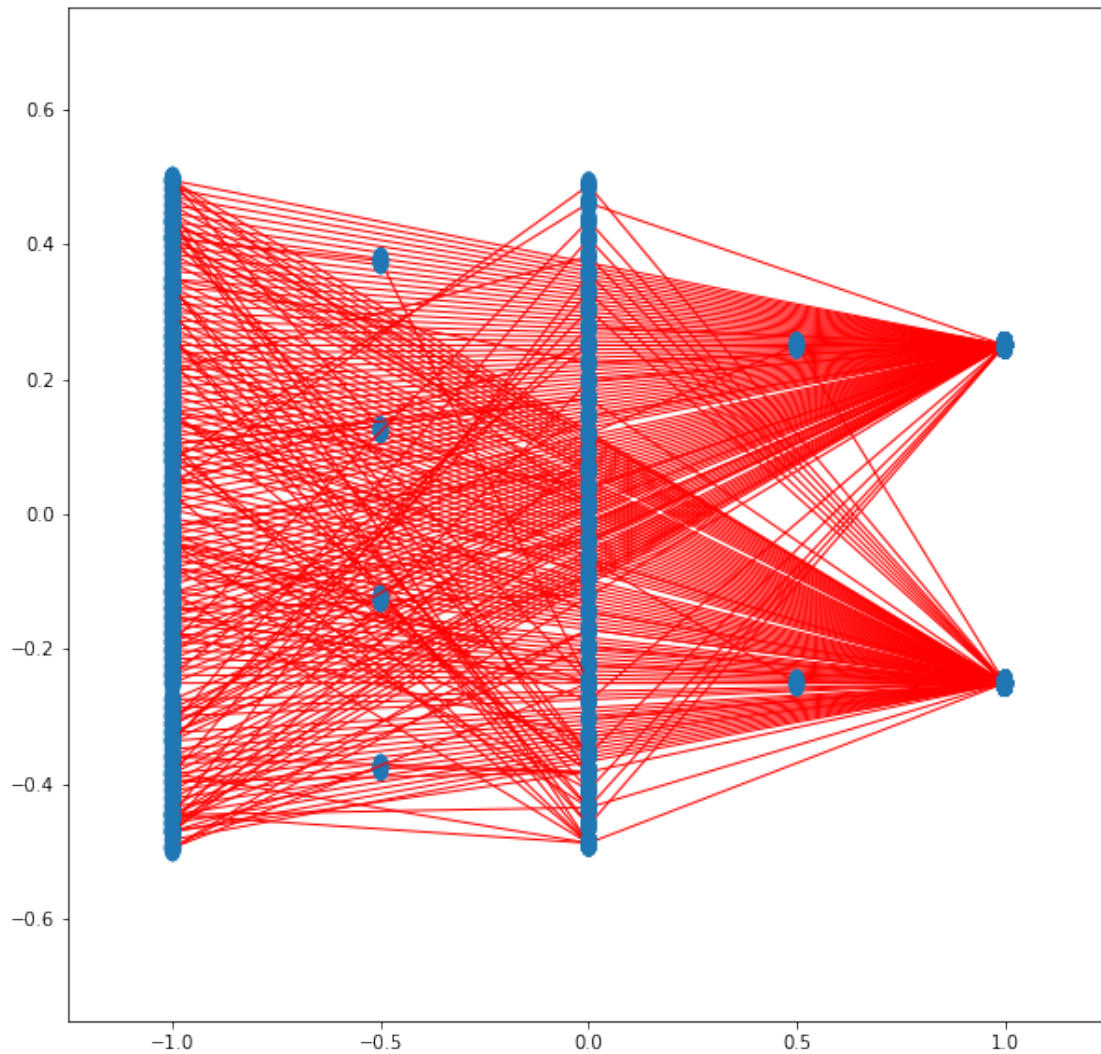


34.1 Podsumowanie

Otrzymano 79% na zbiorze testowym, ale klasy nie są zbalansowane. Są w stosunku 76:24, więc nasz algorytm jest lepszy niż losowy, ale nie satysfakcjonujący. Nasza sieć składa się z dwóch jednej rozwiniętej warstwy, a większość wierzchołków warstwy wyjściowej łączy się z wejściową.

34.2 parametry najlepszej sieci (aktywne połączenia): 266

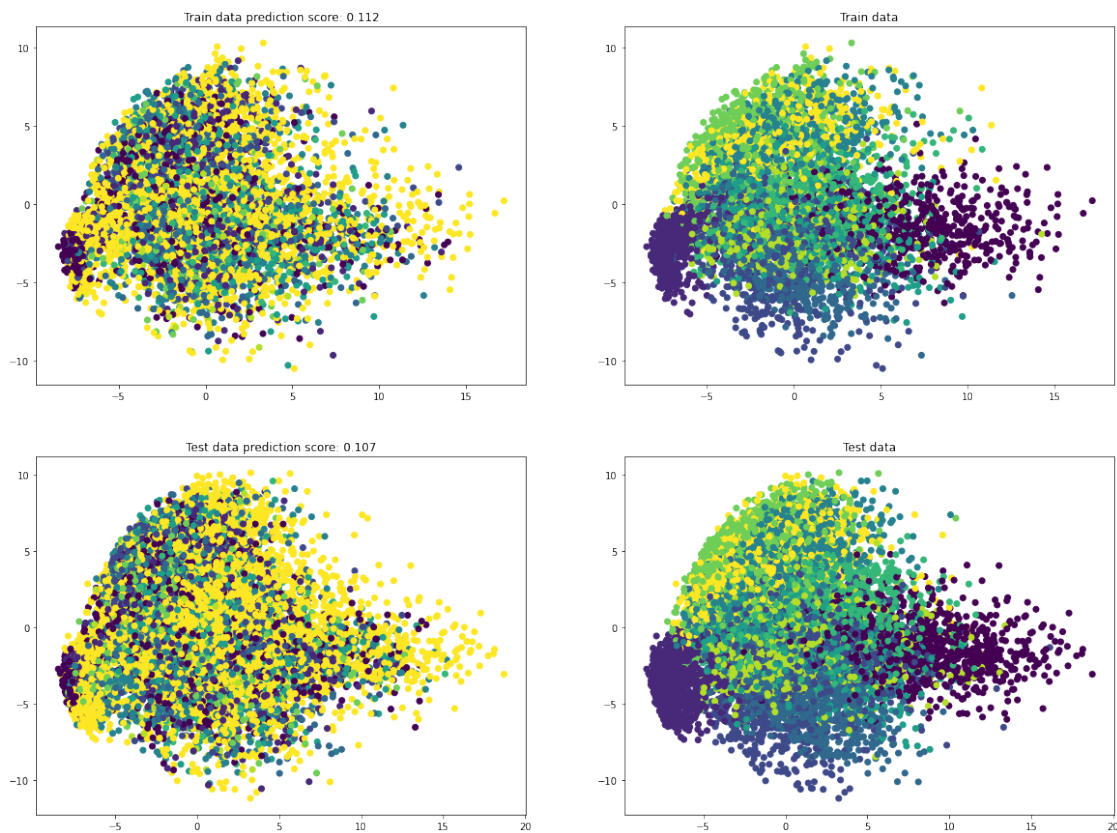
34.3 parametry najlepszej sieci (neurony): 128



35 Test na MNIST

By przyspieszyć obliczenia zastosowaliśmy PCA i wzięliśmy 87 komponentów. Następnie wzięliśmy co 10 wiersz ze zbioru treningowego.

36 Rezultaty

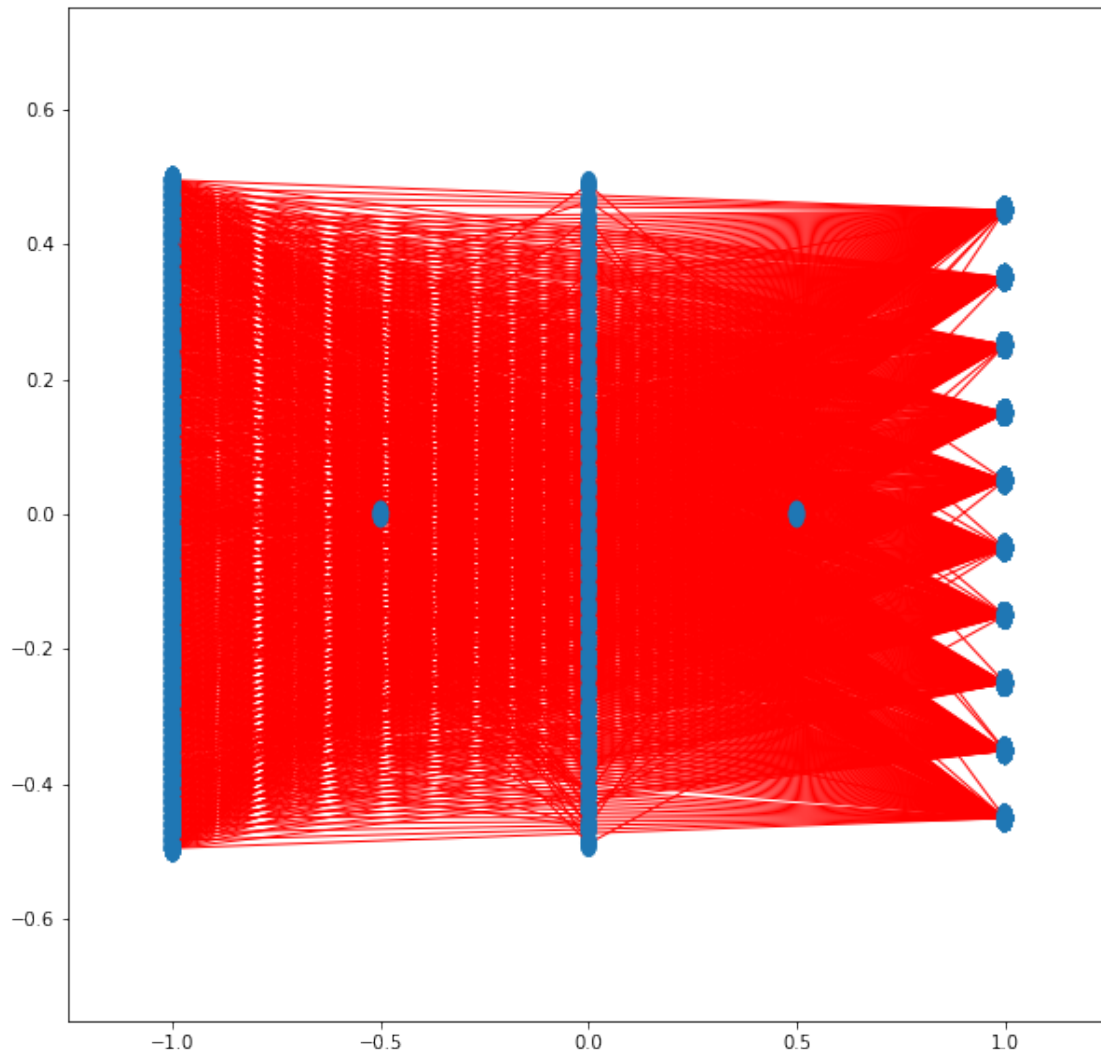


36.1 Podsumowanie

Otrzymaliśmy 10.7% na zbiorze testowym jest to wynik nie wiele lepszy od losowego wybierania. Najwyraźniej parametry modelu były za małe. Sieć wytworzyła dopiero 1 rozwiniętą warstwę. Jest to zbyt prosta sieć na taki problem.

36.2 parametry najlepszej sieci (aktywne połączenia): 965

36.3 parametry najlepszej sieci (neurony): 139



37 Podsumowanie

Algorytm sprawnie i szybko rozwiązuje proste problemy, ale dla skomplikowanych nie daje rady. Prawdopodobnie przyczyną są po prostu zbyt mała ilość generacji i populacji by sieć mogła się nauczyć. Niestety z powodu za słabej mocy obliczeniowej dostępnego sprzętu nie mogę przeprowadzić pełnego uczenia.