PRACOWNIA SPECJALISTYCZNA

Sztuczne sieci neuronowe i systemy ekspertowe

Ćwiczenie nr 1.

KOD PRZEDMIOTU: MYAR2S22003M

Autor: Ostaszewicz Dawid

Kierunek: Automatyka i Robotyka, II stopień

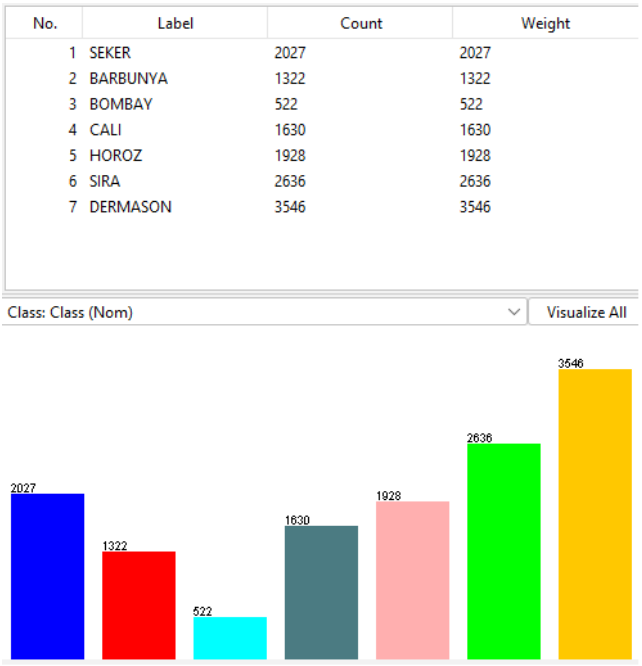
Prowadzący: dr inż. Marcin Derlatka

Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z praktycznym zastosowaniem oprogramowania Weka w treningu sieci neuronowych, o radialnych funkcjach bazowych. Oprogramowanie umożliwia implementację i podział danych na: treningowe, testowe i walidacyjne. Gotowe biblioteki pozwalają na trening sieci neuronowych, przy czym należy zwrócić uwagę na źródło i ich pochodzenie, ponieważ skuteczność dostępnego oprogramowania nie zawsze jest związana z jego poprawnym uzasadnieniem. Jednak w przypadku programu Weka, jego skuteczność jest potwierdzona w licznych artykułach. Ostatecznie celem zajęć było zaprojektowanie optymalnego klasyfikatora RBF, dla przesłanego zbioru danych, zawierającego zdjęcia 7 różnych rodzajów wysuszonych ziaren fasoli.

Dane

Otrzymany zbiór danych obejmuje następujące parametry: powierzchnię, obwód, długość mniejszej i większej osi, proporcje, mimośród, obszar wypukły oraz inne, co łącznie stanowi 16 cech. Zbiór zawiera również informacje o siedmiu gatunkach fasoli: seker, barbunya, bombay, cali, horoz, sira, dermason. W całym zestawie danych znajduje się łącznie 13 611 ziaren, przy czym należy zwrócić uwagę na nierównomierny rozkład liczebności pomiędzy poszczególnymi gatunkami (Rysunek 1).



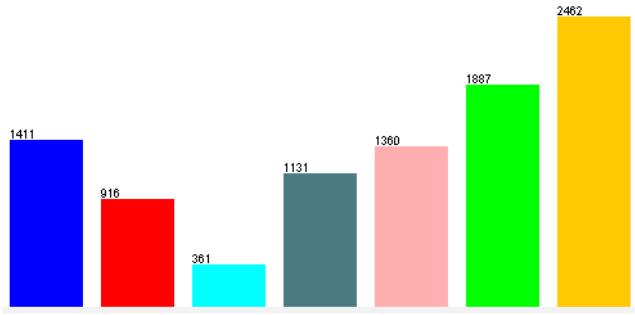
No.	Label	Count	Weight
1	SEKER	2027	2027
2	BARBUNYA	1322	1322
3	BOMBAY	522	522
4	CALI	1630	1630
5	HOROZ	1928	1928
6	SIRA	2636	2636
7	DERMASON	3546	3546

Rysunek 1: Gatunki fasoli - pełny zestaw danych

Kolejnym etapem analizy danych jest weryfikacja podziału zbioru na dane treningowe i testowe. W przypadku braku zachowania proporcji poszczególnych gatunków fasoli w obu ze-

No.	Label	Count	Weight
1	SEKER	1411	1411
2	BARBUNYA	916	916
3	BOMBAY	361	361
4	CALI	1131	1131
5	HOROZ	1360	1360
6	SIRA	1887	1887
7	DERMASON	2462	2462

Class: Class (Nom) Visualize All

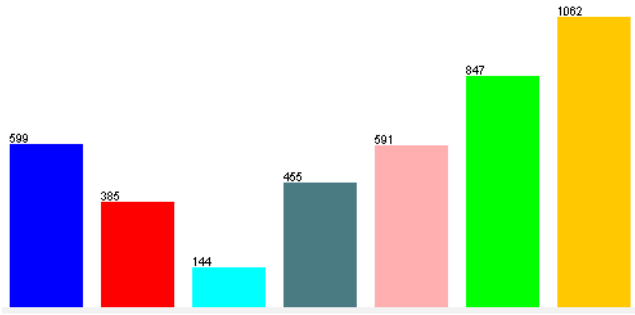


Rysunek 2: Gatunki fasoli - dane treningowe

stawach, istnieje ryzyko, że model zostanie nadmiernie wytrenowany na jednym z nich; co może obniżyć jego zdolność do poprawnej klasyfikacji pozostałych gatunków.

No.	Label	Count	Weight
1	SEKER	599	599
2	BARBUNYA	385	385
3	BOMBAY	144	144
4	CALI	455	455
5	HOROZ	591	591
6	SIRA	847	847
7	DERMASON	1062	1062

Class: Class (Nom) Visualize All

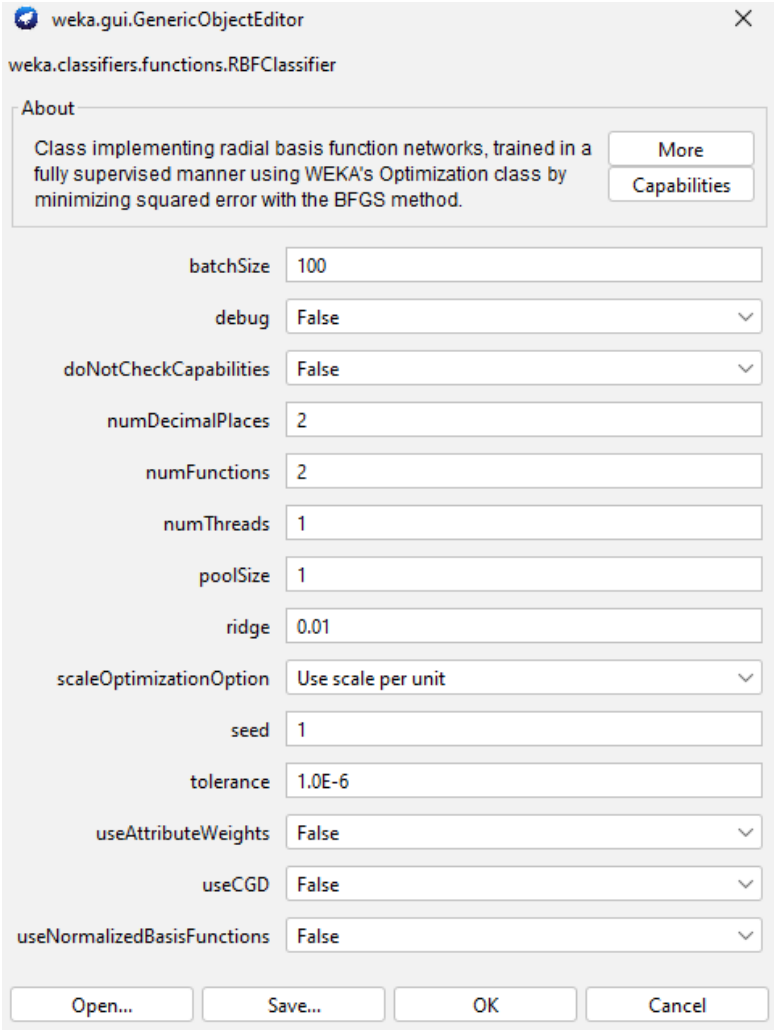


Rysunek 3: Gatunki fasoli - dane testowe

Dane po podziale zachowują pierwotne proporcje gatunków, co wskazuje na poprawność przeprowadzonego procesu. Zbiór treningowy obejmuje 9528 ziaren, a testowy 4083 ziarna, co oznacza, że dane treningowe stanowią 70% całkowitego zbioru. Na tej podstawie można uznać, że podział danych został prawidłowo przeprowadzony.

Wpływ parametrów modelu na jakość klasyfikacji

Po załadowaniu modelu uzyskano parametry podstawowe (Rysunek 4). Celem pierwszego treningu była analiza zachowania modelu operującego na domyślnych parametrach. W tej konfiguracji model ma dwa neurony, co jest minimalną wymaganą liczbą dla sieci bazującej na radialnych funkcjach bazowych (RBF). Zastosowanie klasyfikatora RBF pozwala na dostosowanie parametrów sieci tak, aby minimalizować błąd, przy użyciu algorytmu BFGS (Broyden-Fletcher-Goldfarb-Shanno). Wszystkie parametry zostały znormalizowane do zakresu [0, 1]. Inicjalizacja położenia centrów funkcji bazowych przeprowadzana jest z użyciem algorytmu SimpleKMeans zaimplementowanego w WEKA, natomiast początkowe wartości parametru sigma określono jako maksymalną odległość między każdym centrum a jego najbliższym sąsiadem.



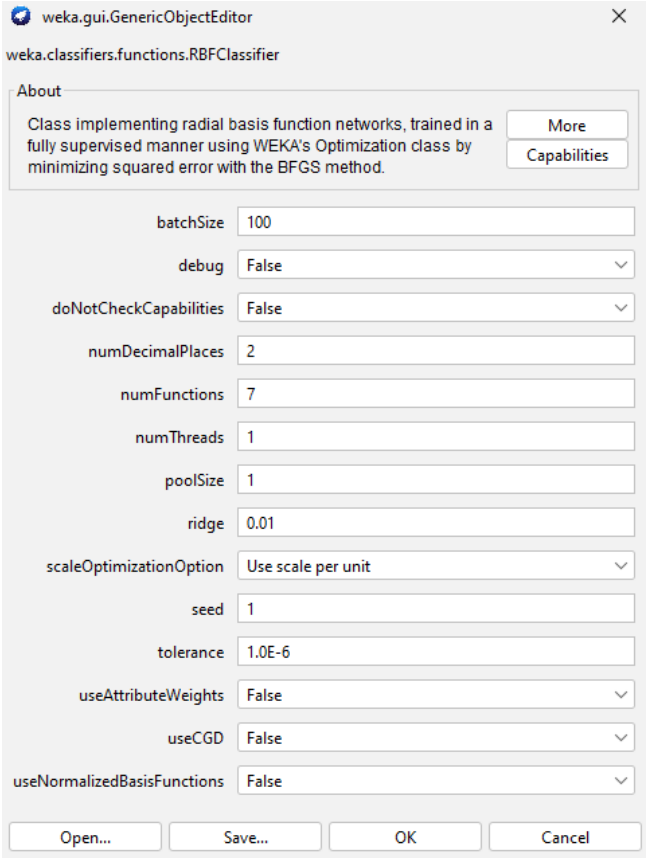
Rysunek 4: Podstawowe parametry modelu

Model o parametrach początkowych błędnie zaklasyfikował 23% ziaren fasoli. Natomiast początkowy czas treningu wynosił około 11.5 s.

```
Incorrectly Classified Instances         936               22.9243 %
```

```
Time taken to build model: 11.49 seconds
```

Poprawę można osiągnąć przez dostosowanie liczby neuronów do liczby podprzestrzeni umożliwiających właściwy podział przestrzeni problemowej. W przypadku klasyfikacji gatunków fasoli konieczne jest przyporządkowanie siedmiu gatunków, co wymaga zastosowania siedmiu neuronów, aby uzyskać zadowalające wyniki klasyfikacji. Dlatego ustawiono liczbę neuronów na 7 w parametrze *numFunctions* (Rysunek 5).



Rysunek 5: Zmiana liczby neuronów - numFunctions = 7

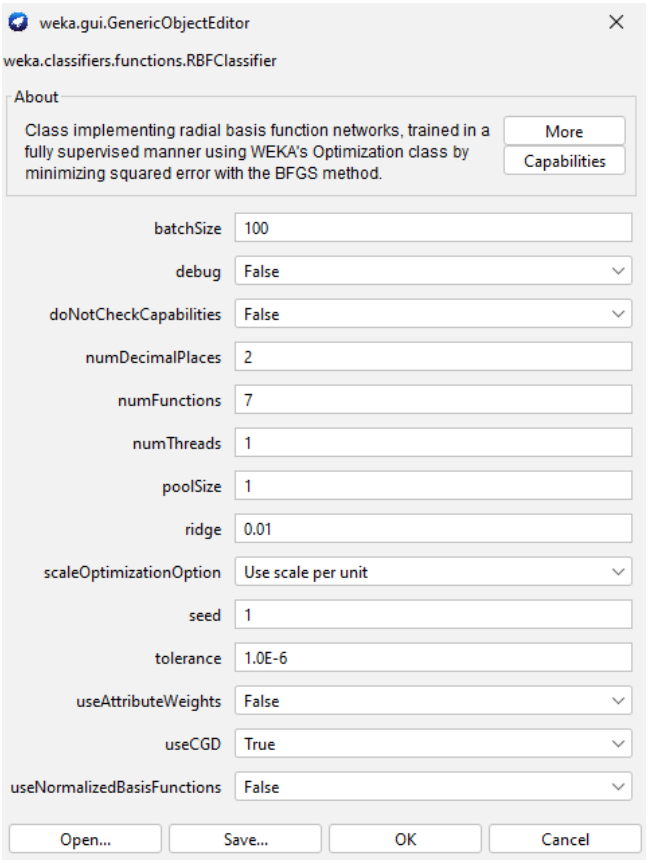
Rezultatem tej konfiguracji jest duży spadek niepoprawnych klasyfikacji, do około 11%

```
Incorrectly Classified Instances       445               10.8988 %
Time taken to build model: 65.73 seconds
```

Kolejnym istotnym parametrem jest useCGD, który aktywuje algorytm gradientu sprzężonego zamiast BFGS. Choć jego zastosowanie spowalnia proces uczenia sieci (Rysunek 6), w tym przypadku przyczyniło się również do zmniejszenia liczby błędnych klasyfikacji.

```
Incorrectly Classified Instances       305                7.47   %
Time taken to build model: 123.21 seconds
```

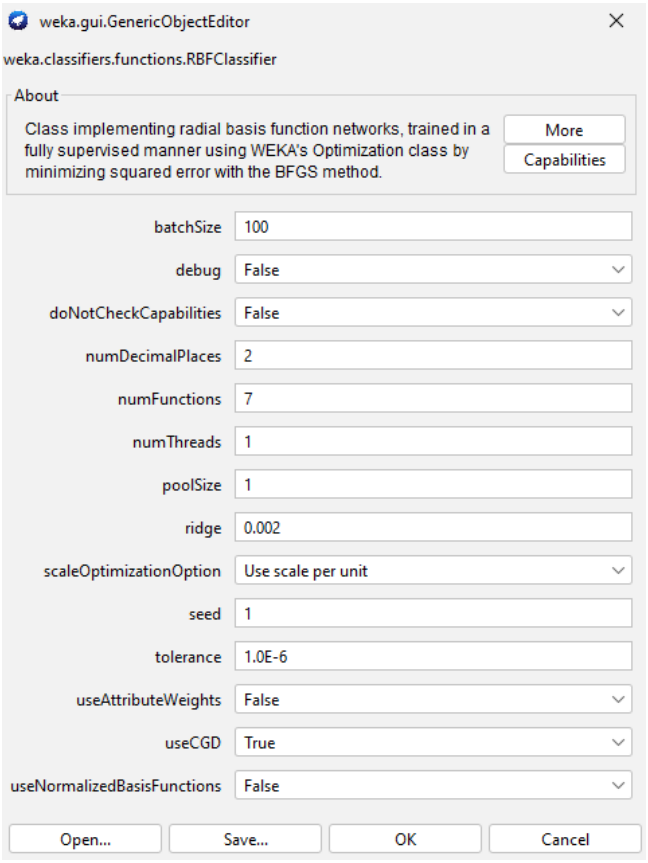
Przy tej samej liczbie neuronów, zmiana metody treningu na gradient sprzężony, spowodowała niemal dwukrotne wydłużenie czasu treningu. Kolejnym parametrem poddanym analizie był



Rysunek 6: Zmiana useCGD

współczynnik kary, ridge, ograniczający nadmierny wzrost wartości wag w warstwie wyjściowej sieci. Zmiana tego parametru nie poprawiła znacząco jakości klasyfikacji, powodując jedynie minimalny spadek liczby błędnych klasyfikacji, przy jednoczesnym wydłużeniu czasu treningu. Czas ten nie stanowi jednak kluczowego kryterium, gdyż, zależnie od potrzeb, możliwe jest zastosowanie wydajniejszego sprzętu, który przyspieszy proces. Wytrenowaną sieć można wielokrotnie wykorzystywać do klasyfikacji. Podsumowując, jakość klasyfikacji jest ważniejsza niż czas treningu. W związku z brakiem istotnej poprawy przy zmianie parametru ridge, pozostawiono jego wartość domyślną. Kolejnym krokiem będzie analiza jakości klasyfikacji przy użyciu macierzy omyłek dla aktualnej konfiguracji sieci.

```
Time taken to build model: 164.85 seconds
Incorrectly Classified Instances       298               7.2986 %
```

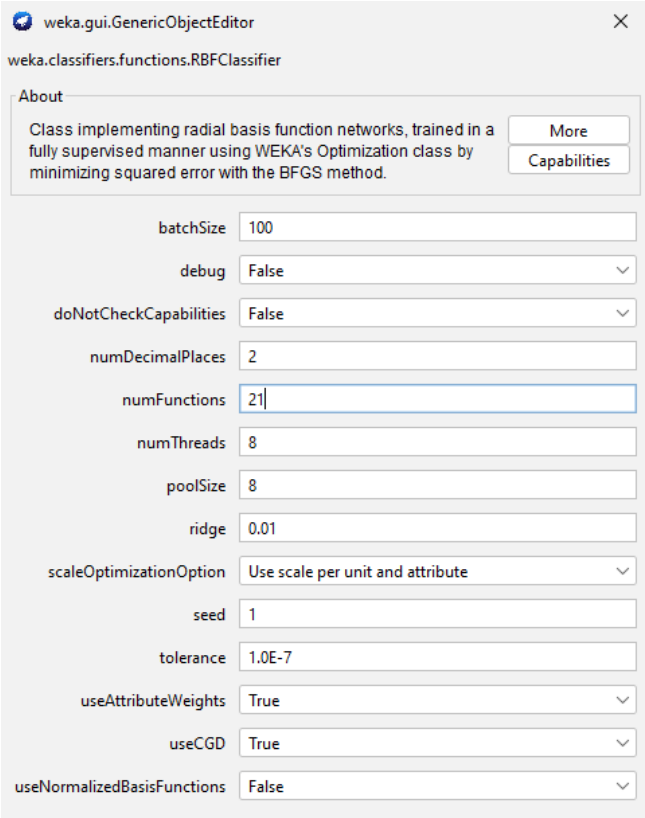


Rysunek 7: Zmiana Ridge

Macierz omyłek wskazuje, że gatunek bombay jest zawsze poprawnie rozpoznawany, co świadczy o wyraźnie odmiennych cechach w porównaniu z pozostałymi gatunkami. Najwięcej błędów klasyfikacyjnych dotyczy natomiast gatunków dermason, sira oraz barbunya, co sugeruje pewne nakładanie się ich cech, uniemożliwiające jednoznaczne przyporządkowanie do właściwych klas. Rozwiązaniem częściowego zmniejszenia liczby pomyłek jest podział przestrzeni na dodatkowe podprzestrzenie wewnątrz klas, co można osiągnąć przez zwiększenie liczby neuronów.

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 565   5   1   0   0  16  12 |   a = SEKER
   3 351   0  20   3   8   0 |   b = BARBUNYA
   0   0 144   0   0   0   0 |   c = BOMBAY
   0  23   0 421   6   5   0 |   d = CALI
   0   2   0   5 569  10   5 |   e = HOROZ
   3   1   0   3  10 762  68 |   f = SIRA
  21   1   0   0   2  65 973 |   g = DERMASON
```



Rysunek 8: Zmiana parametrów modelu

Rysunek 8 przedstawia parametry nowego modelu, w którym liczba neuronów została zwiększona trzykrotnie, z 7 do 21, umożliwiając podział przestrzeni każdej klasy na trzy dodatkowe fragmenty. Taki zabieg powinien poprawić zdolność modelu do klasyfikacji ziaren, które wcześniej sprawiały trudności. Aby przyspieszyć proces, zmodyfikowano również parametry numThreads i poolSize, ustawiając ich wartości na 8. Mimo że sprzęt nie dysponował taką liczbą wątków, pozwoliło to na pełne wykorzystanie zasobów, przyspieszając trening. Dla lepszej jakości klasyfikacji opcję ScaleOptimizationOption ustawiono na „Use scale per unit and attribute", co pozwala na skalowanie każdego neuronu i atrybutu indywidualnie według sigmy. Z tego względu aktywowano również useAttributeWeights w celu uruchomienia wag dla atrybutów. Parametr tolerance, wpływający jedynie na szybkość obliczeń, pozostawiono bez zmian, ponieważ wykorzystano już wszystkie dostępne zasoby.

```
Time taken to build model: 505.1 seconds
Incorrectly Classified Instances       235               5.7556 %
```

Obecne ustawienia spowodowały wzrost jakość klasyfikacji; jedynie 5.7556 % niepoprawnych klasyfikacji. Patrząc na macierz omyłek można stwierdzić, że istnieją jeszcze ziarna, które dzielą większą liczbę cech do siebie zbliżonych. Lepszą jakość klasyfikacji można osiągnąć przez zwiększenie liczby neuronów. Górną granicą jest wyłącznie przeuczenie sieci RBF.

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 580   0   0   0   1  11   7 |   a = SEKER
   2 362   0  14   3   4   0 |   b = BARBUNYA
   0   0 144   0   0   0   0 |   c = BOMBAY
   1  15   0 431   6   2   0 |   d = CALI
   0   2   0   6 570   9   4 |   e = HOROZ
   5   2   0   2   8 765  65 |   f = SIRA
  15   0   0   0   0  51 996 |   g = DERMASON
```

W kolejnym etapie liczba neuronów została zwiększona do 63. Dotychczasowy wynik programu wskazuje na niepoprawną klasyfikację na poziomie 4,6534%. Macierz omyłek pokazuje, że jedynie nieliczne ziarna są błędnie klasyfikowane. Na tym etapie można zakończyć trening, ponieważ dalsze zwiększanie jakości klasyfikacji w przypadku ziaren nie wydaje się uzasadnione. Większość ziaren jest poprawnie klasyfikowana.

```
Incorrectly Classified Instances       190               4.6534 %

=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 582   0   0   0   0   7  10 |   a = SEKER
   1 373   0   6   2   3   0 |   b = BARBUNYA
   0   0 144   0   0   0   0 |   c = BOMBAY
   0   6   0 444   2   3   0 |   d = CALI
   0   2   0   4 572  10   3 |   e = HOROZ
   6   1   0   1   6 778  55 |   f = SIRA
  14   0   0   0   0  48 1000 |    g = DERMASON
```

Wnioski

Sieci RBF klasyfikują dane w przestrzeni wokół centrum funkcji bazowej, więc jakość klasyfikacji zależy w dużym stopniu od specyfiki danych. Jeśli dane są zorganizowane wokół konkretnych centrów, RBF dobrze sobie z nimi poradzi; w przypadku bardziej jednolitego rozkładu danych lepszym wyborem będzie MLP. Gdy dane cech klasyfikowanych obiektów są bardziej przemieszane, może się zdarzyć, że ziarno jednego gatunku wykazuje więcej cech innego gatunku, np. z

powodu jakości zdjęć, błędów przy odczytywaniu cech lub pomyłek podczas wstępnej klasyfikacji. Jeśli jedno ziarno przypomina inny gatunek, warto rozważyć, czy nie jest to faktycznie inny gatunek. W tym przypadku uzyskano satysfakcjonujący poziom błędu klasyfikacji na poziomie 4,6534%, co jest wystarczające przy klasyfikacji ziaren fasoli. W zastosowaniach wymagających większej precyzji, jak diagnostyka chorób, należałoby jednak dążyć do wyższej jakości klasyfikacji. Podsumowując, jakość klasyfikacji powinna być zgodna z potrzebami projektu; jeśli założenia nie są spełnione, kluczowa jest dalsza analiza danych i dostosowywanie modelu do ich specyfiki. Jeśli chodzi o czas treningu sieci; nie ma znaczenia jak długo sieć będzie trenowana, ponieważ jest to sytuacja jednokrotna, tzn. można wynająć lepszy sprzęt do treningu, a następnie wystarczy powielać gotowy model.