

Uniwersytet Łódzki
Wydział Fizyki i Informatyki Stosowanej

SPRAWOZDANIE Z ĆWICZEŃ
Podstawy sztucznej inteligencji AI

Klasyfikacja zbioru IRIS
Sieć jednowarstwowa i MLP
Wpływ liczby neuronów w warstwie ukrytej

Autor: Katarzyna Stańczyk
Kierunek studiów: Informatyka
Data wykonania: 23 stycznia 2026
Rok akademicki: 2025/2026

Łódź, 2025

1 Wstęp

Sieci neuronowe są jednym z podstawowych narzędzi wykorzystywanych do rozpoznawania wzorców i klasyfikacji danych. W zadaniach **uczenia nadzorowanego** model otrzymuje przykłady danych wejściowych wraz z poprawnymi etykietami klas, a następnie uczy się reguły (funkcji decyzyjnej), która pozwala przypisywać nowe, nieznanne dane do odpowiednich kategorii.

Celem niniejszego ćwiczenia jest porównanie możliwości:

- **sieci jednowarstwowej** (klasyfikatora liniowego) oraz
- **sieci wielowarstwowej MLP** (z jedną warstwą ukrytą),

w zadaniu klasyfikacji próbek ze zbioru **IRIS**. Dodatkowo zbadano wpływ liczby neuronów w warstwie ukrytej H na jakość klasyfikacji.

Zbiór IRIS zawiera 150 próbek, opisanych czterema cechami liczbowymi (długość i szerokość działki kielicha oraz płatk) oraz należących do trzech klas: *setosa*, *versicolor*, *virginica* [?]. W ramach eksperymentu przeprowadzono wielokrotne uczenie z różnymi losowaniami podziału danych (różne ziarna losowości), aby ocenić stabilność wyników i uniknąć wnioskowania na podstawie pojedynczego, przypadkowego podziału.

Źródło danych: UCI Machine Learning Repository (IRIS): <https://archive.ics.uci.edu/dataset/53/iris>.

2 Rozwinięcie

2.1 Opis problemu i danych IRIS

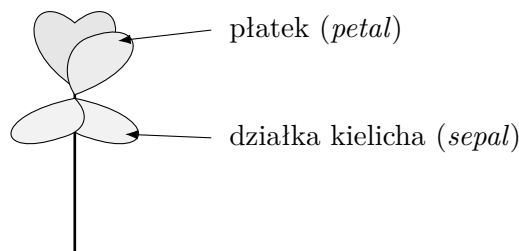
W zadaniu klasyfikacji każda próbka danych jest opisana zestawem **cech** (ang. *features*), czyli liczb, które charakteryzują obiekt. W przypadku IRIS są to cztery wartości pomiarów kwiatu:

- długość i szerokość działki kielicha (*sepal length*, *sepal width*),
- długość i szerokość płatk (*petal length*, *petal width*).

Dla każdej próbki znana jest również **etykieta** (ang. *label*), czyli poprawna odpowiedź: jedna z trzech klas gatunków irysa. Celem modelu jest nauczenie się zależności:

$$x \in \mathbb{R}^4 \longrightarrow \{0, 1, 2\}, \quad (1)$$

czyli przypisania czterowymiarowego wektora cech do jednej z trzech kategorii.



Rysunek 1: Schemat kwiatu irysa i elementów, których wymiary tworzą cechy w zbiorze IRIS (petal/sepal).

2.2 Przygotowanie danych

Podział na zbiory uczący i testowy. Dane zostały podzielone na zbiór treningowy (80%) oraz testowy (20%). Wykorzystano podział ze **stratyfikacją**, co oznacza zachowanie podobnych proporcji klas w obu częściach zbioru. Jest to istotne, aby wyniki testu były reprezentatywne i nie zależały od przypadkowego “przekoszenia” klas.

Standaryzacja cech. Cechy wejściowe mogą mieć różne skale i rozrzuty, co utrudnia proces uczenia (gradienty mogą być zdominowane przez cechy o większych wartościach liczbowych). Dlatego zastosowano **standaryzację**:

$$x' = \frac{x - \mu}{\sigma}, \quad (2)$$

gdzie μ i σ są średnią i odchyleniem standardowym obliczonymi na zbiorze treningowym. Ważne jest, że parametry standaryzacji wyznacza się **wyłącznie na treningu**, a następnie stosuje do danych testowych (dzięki temu model nie “podgląda” informacji z testu podczas przygotowania danych).

Powtarzanie eksperymentu dla różnych ziaren losowości. Ponieważ wynik zależy od losowego podziału danych i inicjalizacji wag, uczenie powtórzono dla kilku ziaren (seedów), a wyniki przedstawiono jako średnią oraz rozrzut (np. na wykresie pudełkowym). Pozwala to ocenić, czy dany wariant modelu jest stabilny i czy różnice nie wynikają jedynie ze szczęśliwego/pechowego losowania.

2.3 Modele porównywane w eksperymencie

2.3.1 Sieć jednowarstwowa (klasyfikator liniowy)

Model jednowarstwowy realizuje liniowe mapowanie wektora cech do trzech wartości wyjściowych:

$$z = Wx + b, \quad (3)$$

gdzie $x \in \mathbb{R}^4$ jest wektorem cech, W i b są parametrami uczonymi, a $z \in \mathbb{R}^3$ to tzw. **logity** (surowe wyniki dla każdej klasy). Taki model odpowiada **wieloklasowej regresji logistycznej**:

prawdopodobieństwa klas uzyskuje się przez funkcję softmax, a w praktyce w PyTorch jest ona uwzględniona w funkcji straty `CrossEntropyLoss`. Model liniowy tworzy *proste granice decyzyjne* (hiperpłaszczyzny) w przestrzeni cech.

2.3.2 Sieć wielowarstwowa MLP z jedną warstwą ukrytą

Model MLP (Multi-Layer Perceptron) rozszerza klasyfikator liniowy o **warstwę ukrytą** z nieliniową funkcją aktywacji ReLU:

$$h = \text{ReLU}(W_1x + b_1), \quad z = W_2h + b_2, \quad (4)$$

gdzie $h \in \mathbb{R}^H$ jest wektorem aktywacji w warstwie ukrytej, a H oznacza liczbę neuronów. Dzięki nieliniowości ReLU model może tworzyć bardziej złożone granice decyzyjne niż klasyfikator liniowy. W eksperymencie badano wpływ parametru H na jakość klasyfikacji, testując wartości:

$$H \in \{1, 2, 4, 8, 16, 32, 64\}. \quad (5)$$

2.4 Algorytm uczenia (intuicyjnie)

Uczenie polega na dopasowaniu wag W i biasów b tak, aby model przypisywał poprawne klasy. Dla każdej próbki model zwraca logity z . Następnie funkcja softmax zamienia logity na rozkład prawdopodobieństw klas, a **funkcja straty** (Cross-Entropy) porównuje przewidywania z prawdziwą etykietą. Im większy błąd, tym większa wartość straty.

Parametry modelu są aktualizowane metodą **wstecznej propagacji błędów** (*backpropagation*), czyli obliczaniem pochodnych funkcji straty względem wag. Następnie optymalizator wykonuje krok aktualizacji parametrów, zmniejszając stratę. W pracy użyto optymalizatora **Adam**, który automatycznie dobiera efektywne kroki uczenia dla parametrów.

Pojęcia praktyczne.

- **Epoka** – jedno pełne przejście przez zbiór treningowy.
- **Batch** – liczba próbek przetwarzanych jednocześnie; uczenie na batchach stabilizuje i przyspiesza obliczenia.
- **Learning rate (lr)** – rozmiar kroku aktualizacji parametrów; zbyt duży może destabilizować uczenie, zbyt mały je spowalnia.

2.5 Parametry eksperymentu

Wszystkie porównywane modele uczono z tą samą konfiguracją, aby różnice w wynikach wynikały z architektury (linear vs MLP i H), a nie z innych ustawień. Wykorzystano:

- podział danych: 80% trening / 20% test,
- liczba epok: 250,
- optymalizator: Adam,
- funkcja straty: Cross-Entropy (`CrossEntropyLoss`),

- learning rate: $lr = 0.03$,
- batch size: 16,
- kilka ziaren losowości (seedy) w celu oceny stabilności wyników.

2.6 Sposób oceny jakości i przygotowane wizualizacje

Wyniki oceniano kilkoma uzupełniającymi się metodami:

- **Accuracy** – odsetek poprawnie sklasyfikowanych próbek na zbiorze testowym.
- **Macierz pomyłek (confusion matrix)** – pokazuje, które klasy są mylone i w jakim stopniu.
- **Krzywe uczenia (loss train vs test)** – przebieg straty w kolejnych epokach; pozwala ocenić, czy model się uczy oraz czy nie dochodzi do przeuczenia (np. spadek straty na treningu przy wzroście na teście).
- **Wykres accuracy w funkcji H** – pokazuje wpływ liczby neuronów w warstwie ukrytej na jakość klasyfikacji.
- **Boxplot accuracy vs H** – przedstawia rozrzut wyników dla różnych seedów, czyli stabilność danego wariantu.

Dodatkowo wykonano wizualizację **granic decyzyjnych w przestrzeni 2D po redukcji wymiaru PCA**. Ponieważ dane IRIS mają 4 cechy, zastosowano PCA do rzutowania danych na dwie składowe główne, aby możliwe było zwizualizowanie obszarów decyzyjnych modelu. Należy pamiętać, że jest to przybliżenie: wykres pokazuje zachowanie modelu w przestrzeni zredukowanej, a nie w oryginalnej przestrzeni 4D.

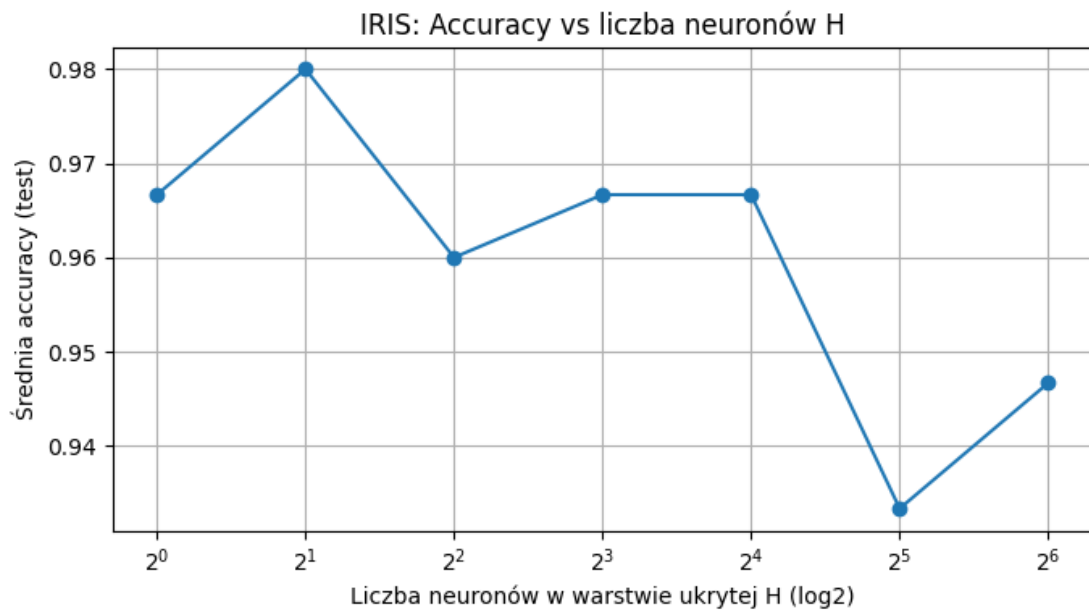
3 Wyniki

3.1 Zestawienie jakości klasyfikacji

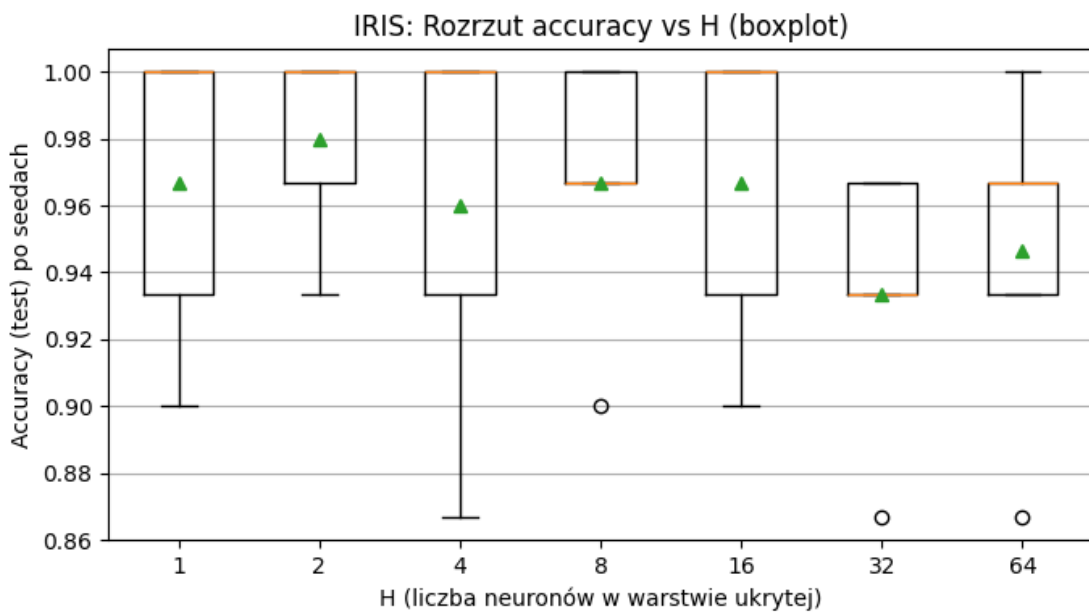
Tabela 1: Accuracy na zbiorze testowym (średnia i odchylenie standardowe po seedach).

Model	H	Accuracy (mean)	Accuracy (std)
linear	–	0.9733	0.0327
MLP	1	0.9667	0.0422
MLP	2	0.9800	0.0267
MLP	4	0.9600	0.0533
MLP	8	0.9667	0.0365
MLP	16	0.9667	0.0422
MLP	32	0.9333	0.0365
MLP	64	0.9467	0.0452

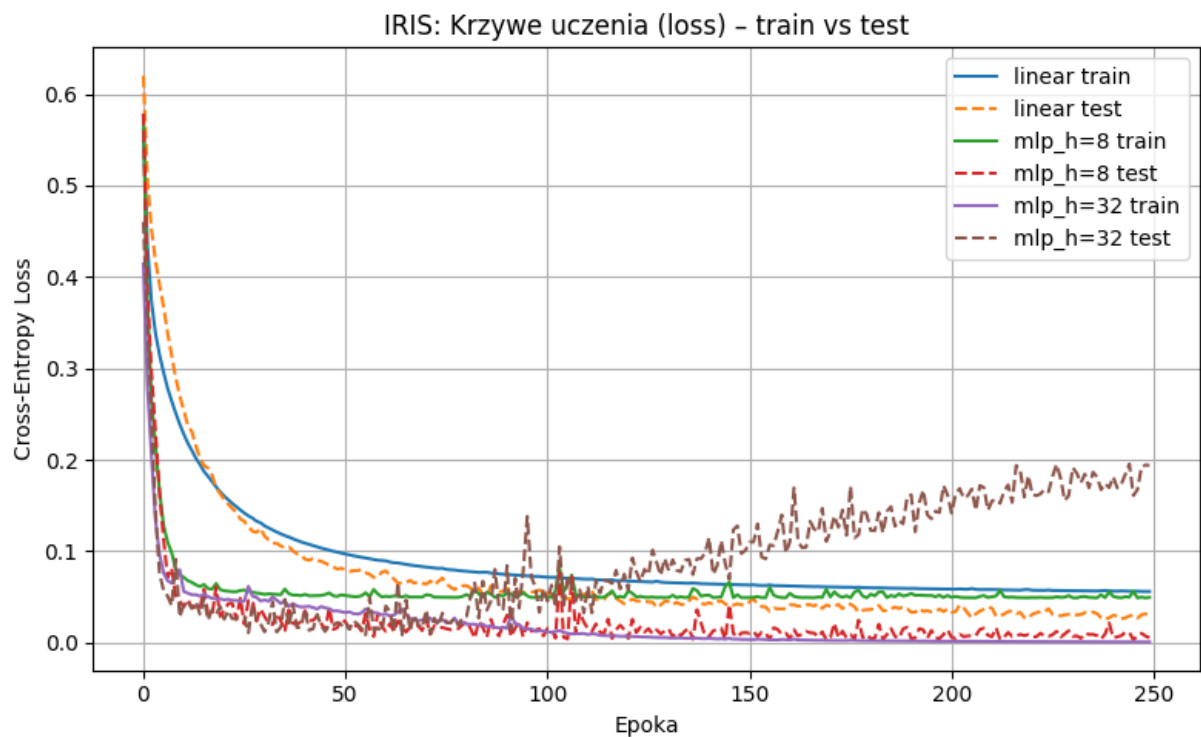
3.2 Wykresy



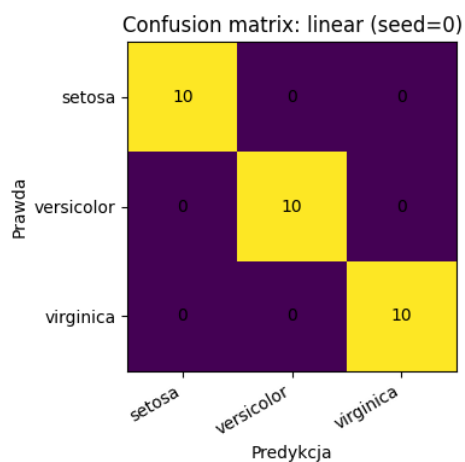
Rysunek 2: Średnia accuracy na zbiorze testowym w zależności od liczby neuronów w warstwie ukrytej H (MLP).



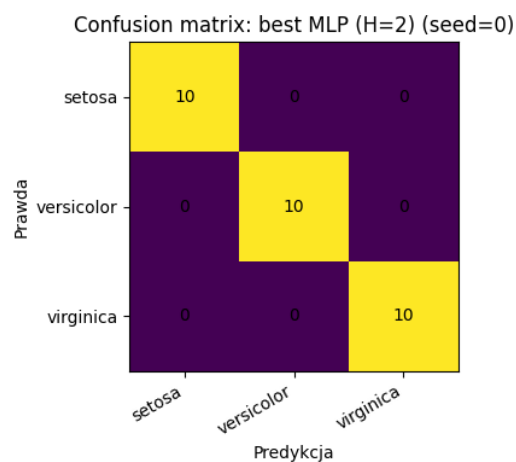
Rysunek 3: Rozrzut accuracy po seedach dla różnych wartości H (MLP). Zielony znacznik przedstawia średnią.



Rysunek 4: Krzywe uczenia (loss) dla modeli: linear oraz MLP ($H=8$, $H=32$). Linie przerywane odpowiadają zbiorowi testowemu.

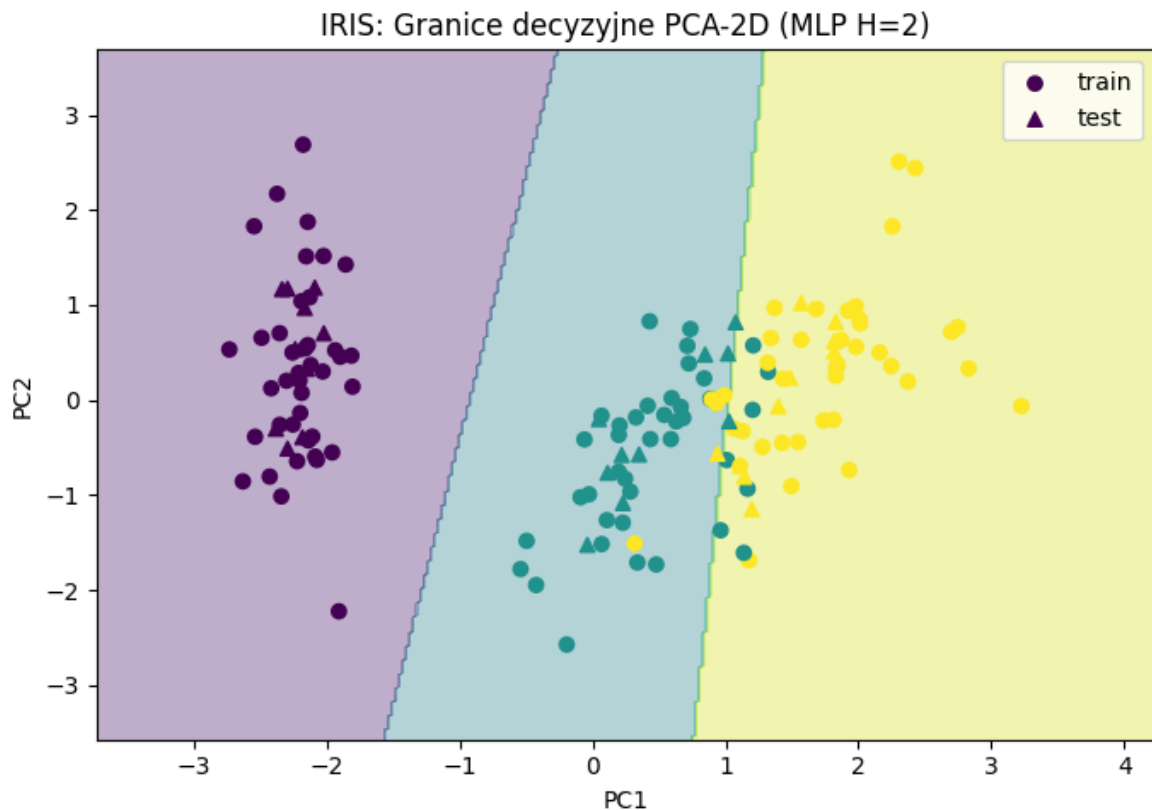


(a) Model linear (seed=0).



(b) MLP ($H = 2$) (seed=0).

Rysunek 5: Porównanie macierzy pomyłek dla modelu linear oraz najlepszego wariantu MLP.



Rysunek 6: Granice decyzyjne w przestrzeni PCA-2D dla MLP ($H = 2$). Punkty oznaczają dane treningowe (koła) i testowe (trójkąty).

4 Komentarz do wyników

Porównanie modeli: linear vs MLP. Na podstawie wyników przedstawionych w tabeli oraz na rys. 2 można zauważyć, że zbiór IRIS jest klasyfikowany z wysoką skutecznością zarówno przez model jednowarstwowy, jak i przez MLP. Średnia accuracy dla klasyfikatora liniowego wyniosła 0.9733, natomiast najlepszy wynik średni uzyskano dla MLP z $H = 2$ (0.9800). Różnice pomiędzy wariantami są niewielkie, co sugeruje, że dla tego zbioru danych model liniowy jest często wystarczający.

Wpływ liczby neuronów H i stabilność wyników. Wykres rozrzutu (rys. 3) pokazuje, że dla części konfiguracji MLP pojawiają się większe wahania jakości pomiędzy seedami (np. $H = 4$ i $H = 64$), co wynika z losowego podziału danych oraz inicjalizacji wag. Dla większej liczby neuronów ($H = 32$) zaobserwowano spadek średniej accuracy oraz większą podatność na błędy w pojedynczych uruchomieniach.

Krzywe uczenia i przeuczenie (overfitting). Analiza krzywych uczenia (rys. 4) wskazuje, że model linear uczy się stabilnie i osiąga plateau. Dla MLP z $H = 8$ również obserwowany jest stabilny spadek straty. Natomiast dla MLP z $H = 32$ w pewnym momencie loss na zbiorze testowym zaczyna rosnąć, co jest typowym objawem **przeuczenia** (overfitting) przy zbyt dużej pojemności modelu względem rozmiaru danych.

Macierze pomyłek (confusion matrix). Macierze pomyłek (rys. 5a i 5b) dla seeda 0 są idealne (brak błędów klasyfikacji). Oznacza to, że przy tym konkretnym podziale danych oba modele potrafią w pełni rozdzielić klasy. Różnice pomiędzy wariantami ujawniają się dopiero po uśrednieniu wyników dla różnych seedów.

Wizualizacja granic decyzyjnych w PCA-2D. Dodatkowa wizualizacja w przestrzeni PCA-2D (rys. 6) pokazuje, że klasy są w dużej mierze separowalne już w dwóch wymiarach, co tłumaczy wysoką skuteczność nawet prostych modeli. W szczególności klasa *setosa* tworzy wyraźnie odseparowane skupisko, natomiast granica pomiędzy *versicolor* i *virginica* jest mniej jednoznaczna.

5 Zakończenie

W ramach ćwiczenia porównano sieć jednowarstwową oraz wielowarstwową MLP w zadaniu klasyfikacji próbek ze zbioru IRIS. Zbadano wpływ liczby neuronów w warstwie ukrytej na jakość klasyfikacji, wykonując eksperymenty dla kilku wartości H oraz kilku ziaren losowych.

Wyniki pokazują, że zbiór IRIS jest stosunkowo prosty z punktu widzenia klasyfikacji: model liniowy osiąga wysoką skuteczność, a dodanie warstwy ukrytej nie gwarantuje istotnej poprawy. Najlepszą jakość średnią uzyskano dla niewielkiej liczby neuronów ($H = 2$), natomiast zbyt duża warstwa ukryta może prowadzić do przeuczenia i pogorszenia generalizacji (przykład $H = 32$).

Uzyskane rezultaty są zgodne z intuicją: przy małej liczbie próbek (150) oraz niewielkiej liczbie cech (4) zbyt złożone modele nie zawsze są korzystne, a kluczowe znaczenie ma stabilna procedura oceny (uśrednianie po seedach oraz analiza rozrzutu wyników).