Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Sztuczne Sieci Neuronowe

Klasyfikacja cukrzycy

Sobolewski Konrad, Walczak Paweł

Warszawa, 2017

Spis treści

1. Podział danych i implementacja sieci

Celem projektu było zaimplementowanie sieci neuronowej do klasyfikacji występowania cukrzycy.

Korzystaliśmy ze zbioru danych pochodzącego ze strony

$$https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes$$

Zbiór składa się z 768 próbek. Liczba atrybutów to 8, zaś wyjściem jest binarna klasyfikacja (1 - cukrzyca występuje, 0 - brak cukrzycy). Przed podaniem danych na wejście sieci neuronowej, dokonaliśmy szeregu modyfikacji. Po pierwsze wydzielamy, atrybuty ($features$) od danych o klasyfikacji ($labels$). Następnie normalizujemy dane. Brak normalizacji danych, charakteryzuje się tym, że pewne z atrybutów (ze względu na zakres wartości jakie przyjmuje, czyt. większe niż inne atrybuty) dominują nad innymi, co powoduje że sieć neuronowa daje błędne wyniki. Następnie dokonaliśmy podziału danych przy użyciu pakietu *sklearn* na dwa zbiory: uczący (80%) oraz testowy (20%). W tym celu została wykorzystana funkcja *train_test_split*. Po uzyskaniu danych stosujemy metodę *one hot encoding* dla danych o klasyfikacji (wyjść), aby pozyskane dane przetransformować do formatu, tak aby zgadzały się z formą wymaganą przez metody tensorflow. W dalszej części sprawozdania ukażemy również wpływ PCA (*Principal Component Analysis*) na skuteczności klasyfikacji. Uzyskane dane wspomnianymi krokami są gotowe do przepuszczenia przez sieć neuronową. Dla naliczania kosztu, stosujemy metodę $softmax\ cross\ entropy$. Możemy ją stosować, dzięki uprzedniemu zastosowaniu one hot encoding dla wyjść.

2. Badania

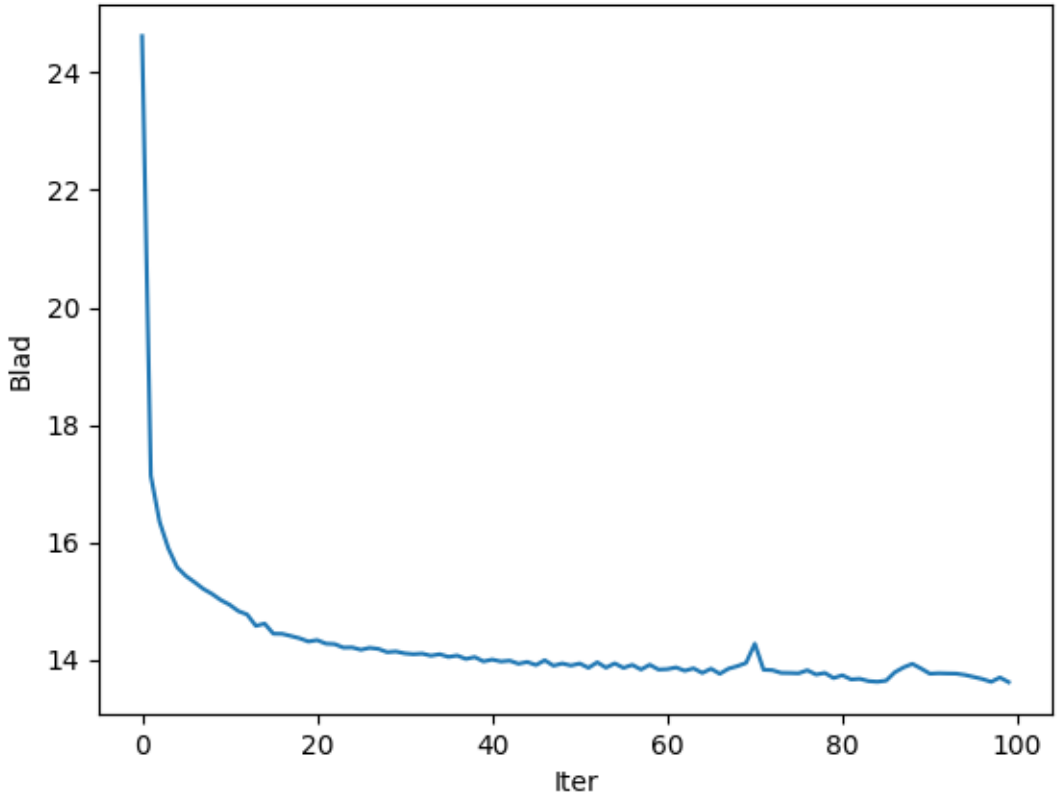
Należy uwzględnić fakt, że ze względu na losowe wybieranie wartości początkowych wag, osiągane są wyniki różnej jakości. Z tego względu, dla każdej konfiguracji, dokonywaliśmy kilku (4-5) generacji po czym wybieraliśmy najlepszy wynik.

2.1. Główne Badania

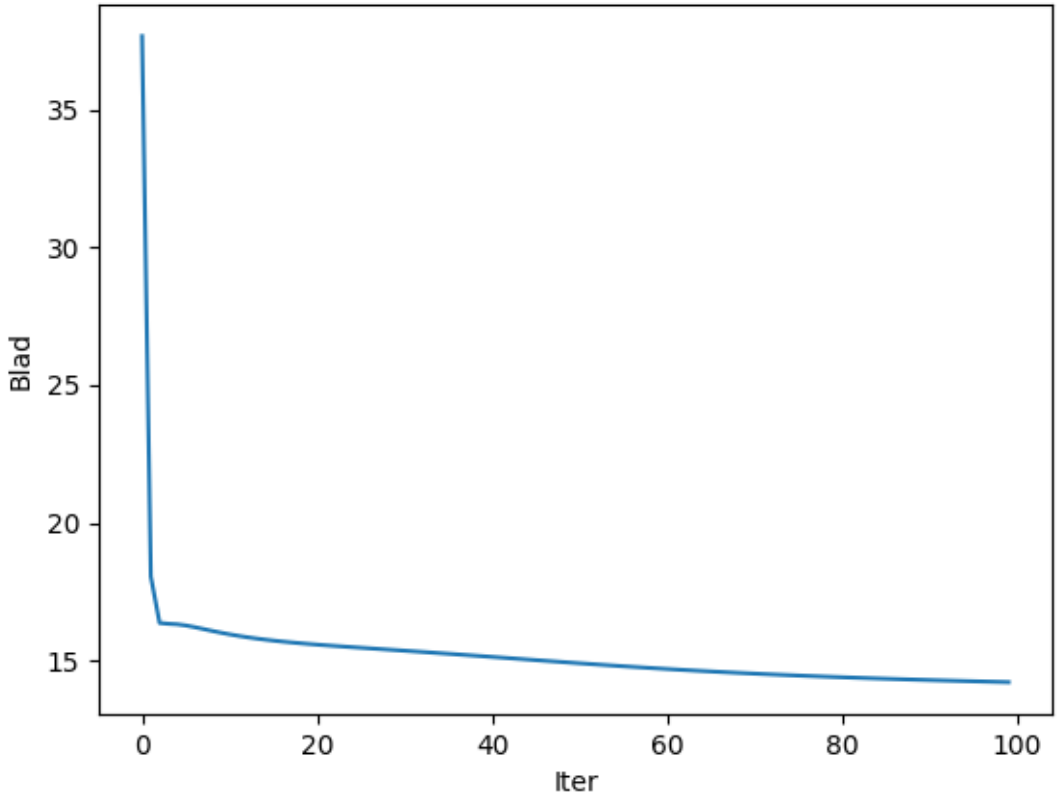
Nadmieniamy, że badania wykonujemy z uwzględnieniem tzw. Batch Size. Jest to sposób nauczania sieci neuronowych, gdzie aktualizację wag wykonujemy dopiero po uwzględnieniu danej ilości próbek. Dla systemów komputerowych z GPU nauka sieci neuronowych z wykorzystaniem wspomnianych batch size, jest czasowo dużo bardziej efektywna. Powoduje to również inny sposób naliczania błędów (naliczanie dla każdego "batcha" osobno i późniejsze sumowanie).

2.1.1. Warstwy

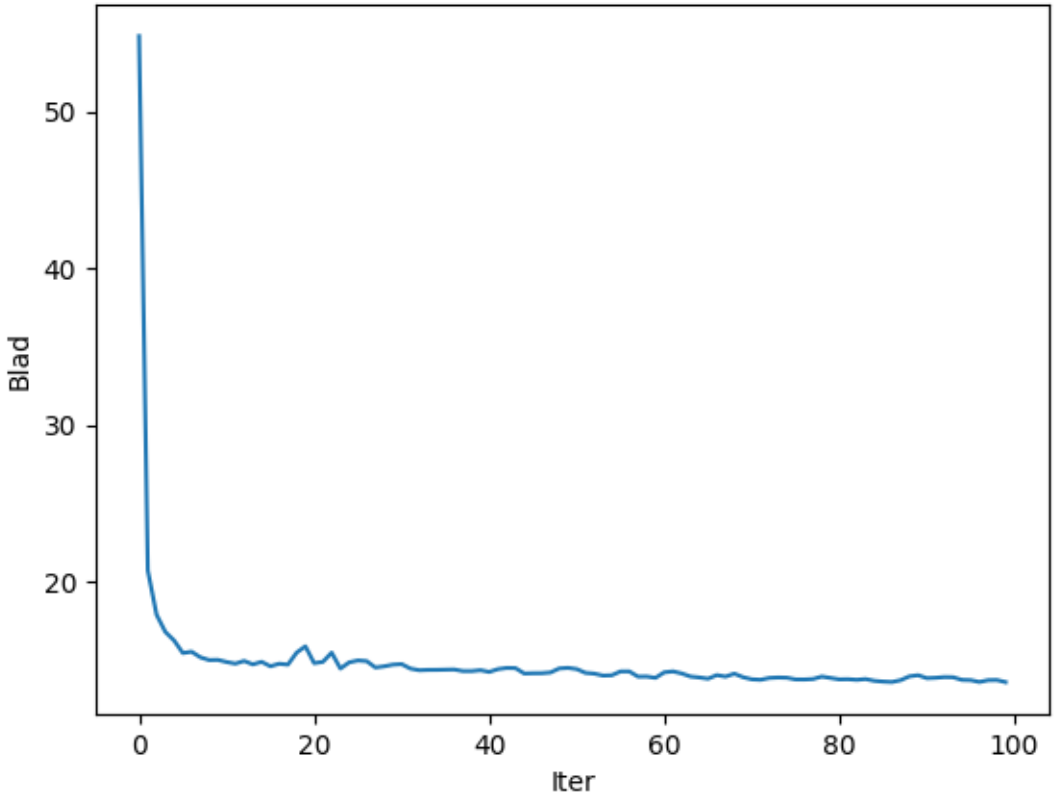
Warstwy(neurony)	Wynik
1 (10)	Uczący: 0.791531, Testowy: 0.785702
1 (50)	Uczący: 0.775244, Testowy: 0.805195
2 (50,10)	Uczący: 0.832388, Testowy: 0.815191
2 (100,50)	Uczący: 0.806843, Testowy: 0.785937
3 (50,10,10)	Uczący: 0.770847, Testowy: 0.795195
3 (500,500,500)	Uczący: 0.660637, Testowy: 0.710585



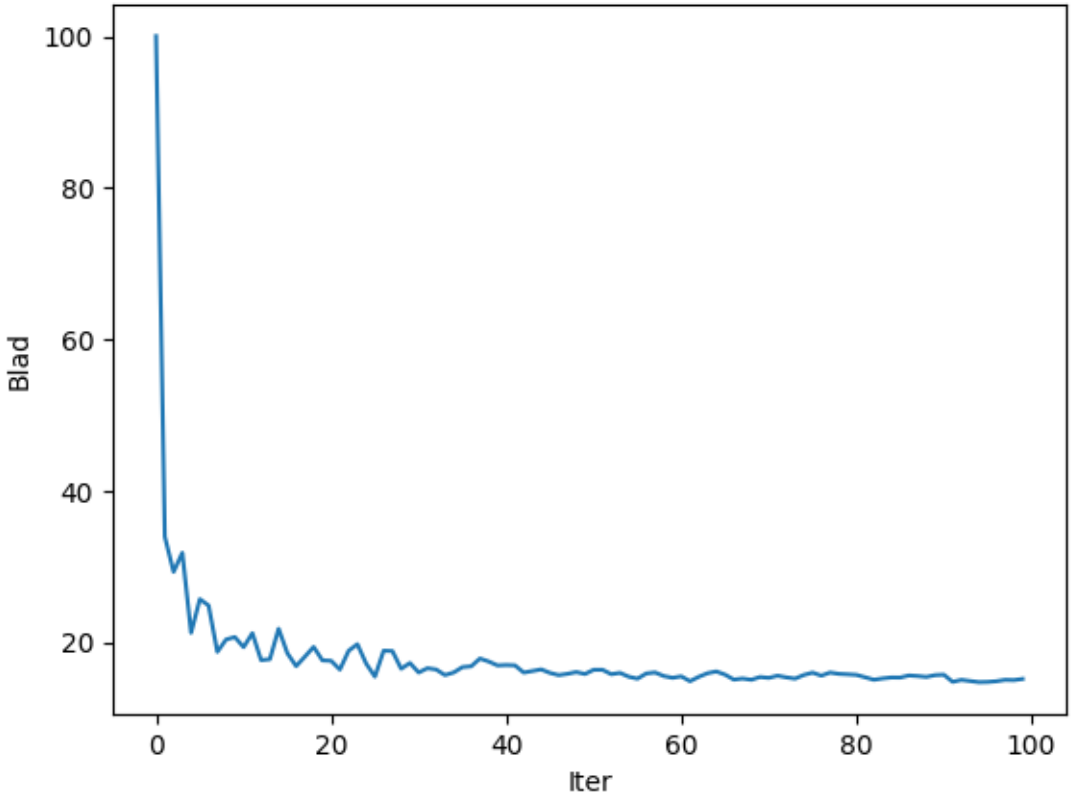
Rys. 2.1. Błąd , 1 warstwa 10 neuronów



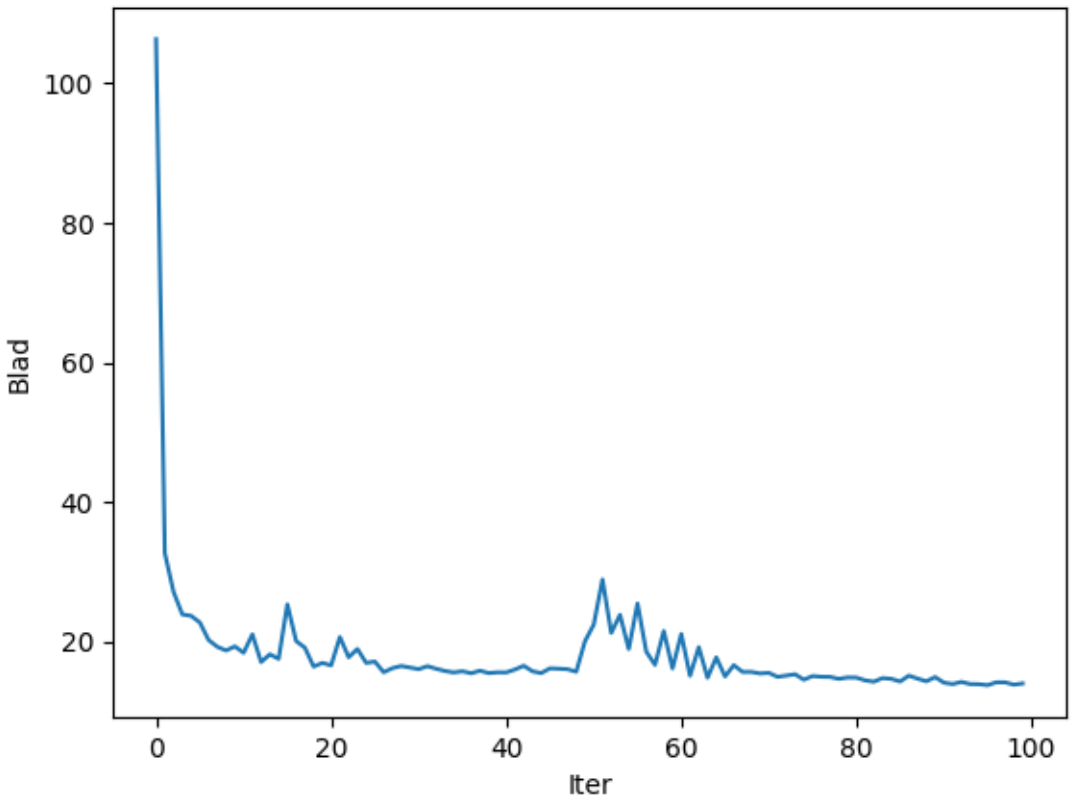
Rys. 2.2. Błąd , 1 warstwa 50 neuronów



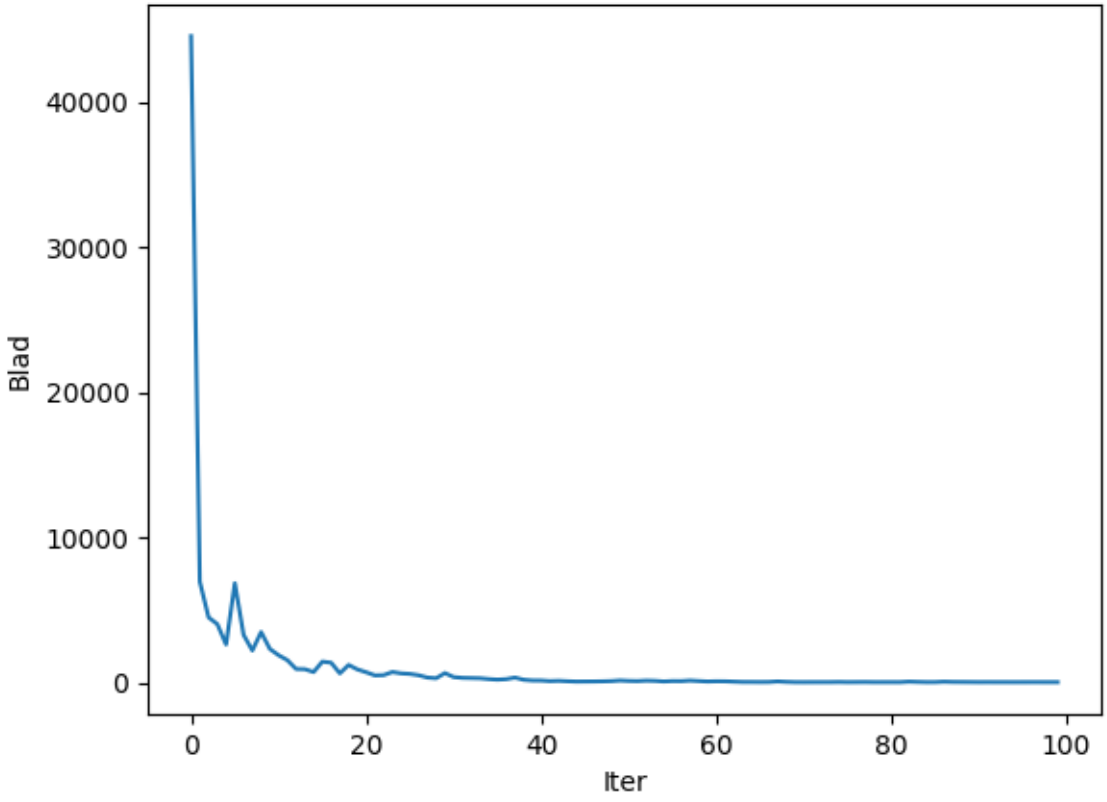
Rys. 2.3. Błąd , 2 warstwy 50,10 neuronów



Rys. 2.4. Błąd , 2 warstwy 100,50 neuronów



Rys. 2.5. Błąd , 3 warstwa 50,10,10 neuronów

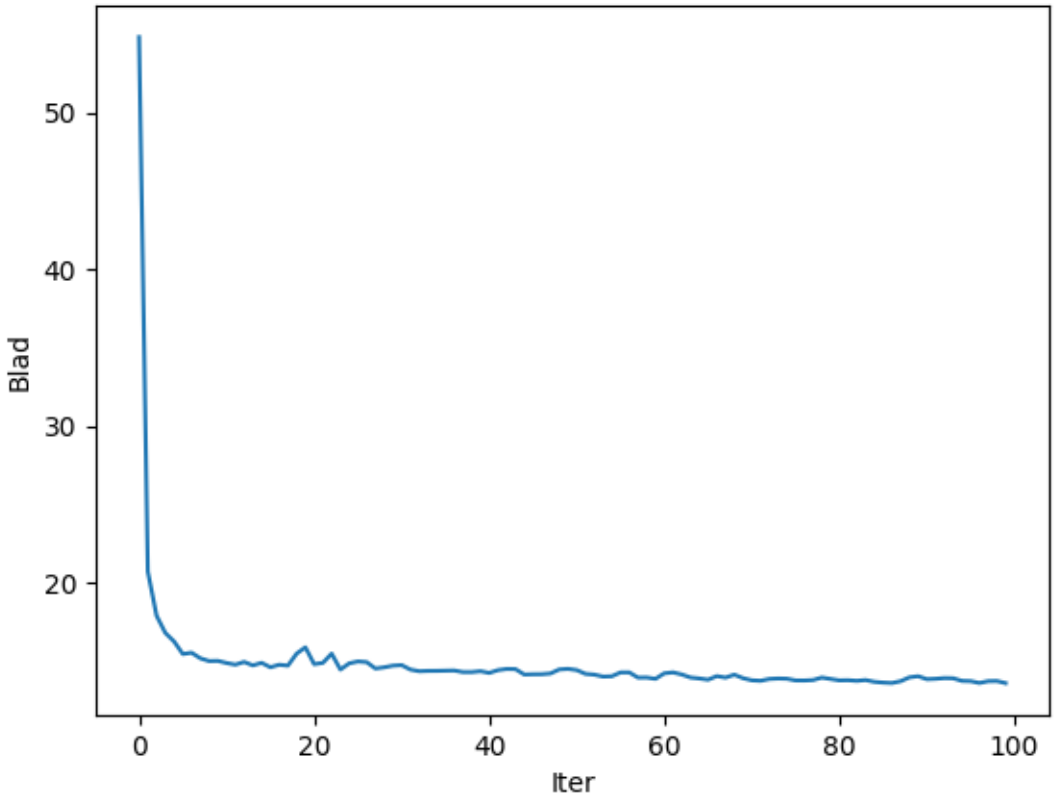


Rys. 2.6. Błąd , 3 warstwa 500,500,500 neuronów

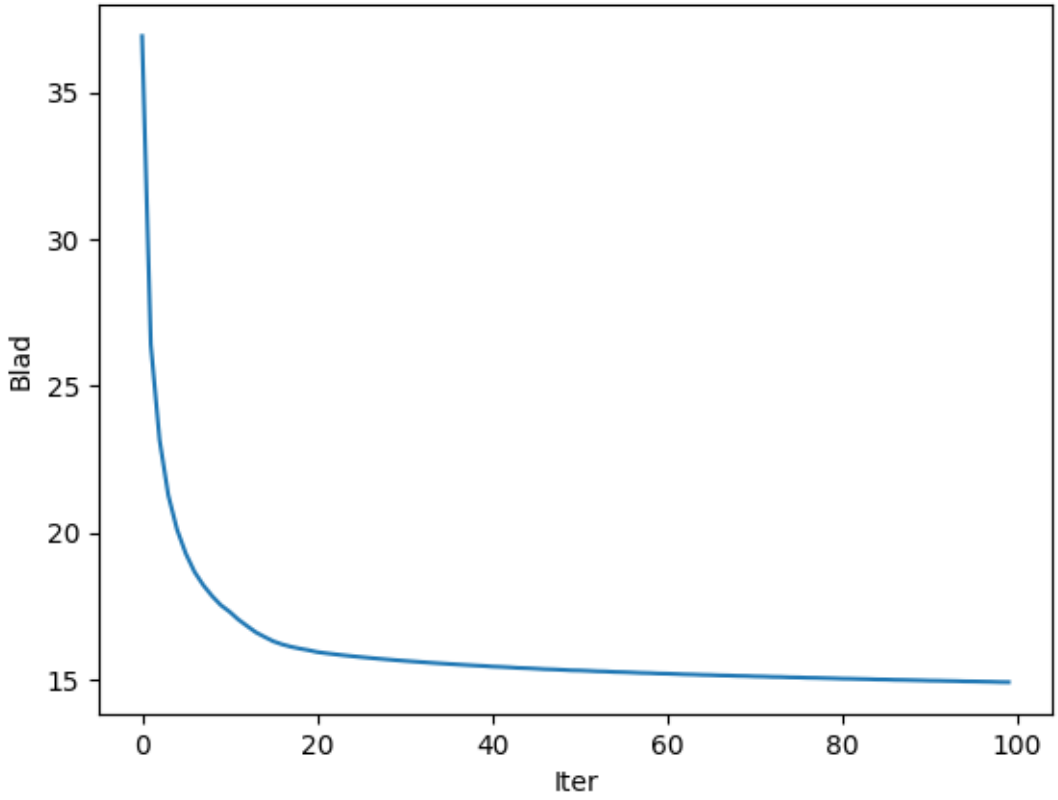
Oceniając wykresy, zauważamy, że w pierwszych kilku/kilkunastu iteracjach, następuje szybki spadek wartości błędu. W dalszych iteracjach jakość rozwiązania nie poprawia się. Dla niektórych konfiguracji następują pewne oscylacje w fazie stabilizacji, jednakże szybko również i one gasną. Zauważamy, że liczba neuronów w jednej warstwie ukrytej nie wpływa znacząco na jakość rozwiązań.Wyniki dla dwóch warstw ukrytych są nieco lepsze (precyzja dla obu zbiorów o 2 % większa). Wyniki są podobne do poprzednich konfiguracji (nieco gorsze od 2 warstw ukrytych). Jednakże, zauważamy że czas obliczeń jest już znaczny. Na podstawie powyższych wyników jednoznacznie widać przewagę sieci z dwoma warstwami. Zostanie ona wykorzystana do dalszych badań.

2.1.2. Metoda Optymalizacji

Metoda	Wynik
Adam	Uczący: 0.833876, Testowy: 0.818182
Stochastic Gradient Descent	Uczący: 0.767101, Testowy: 0.785714



Rys. 2.7. Adam Optymalization

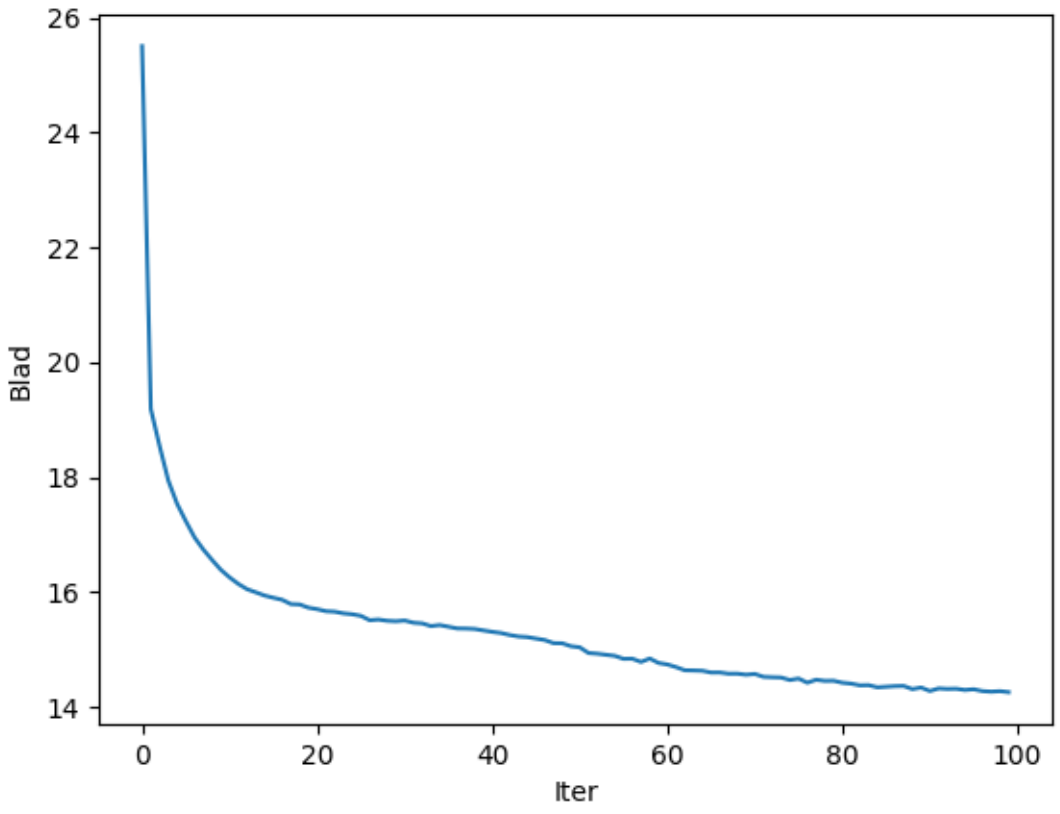


Rys. 2.8. Stochastic Gradient Descent Optymalization

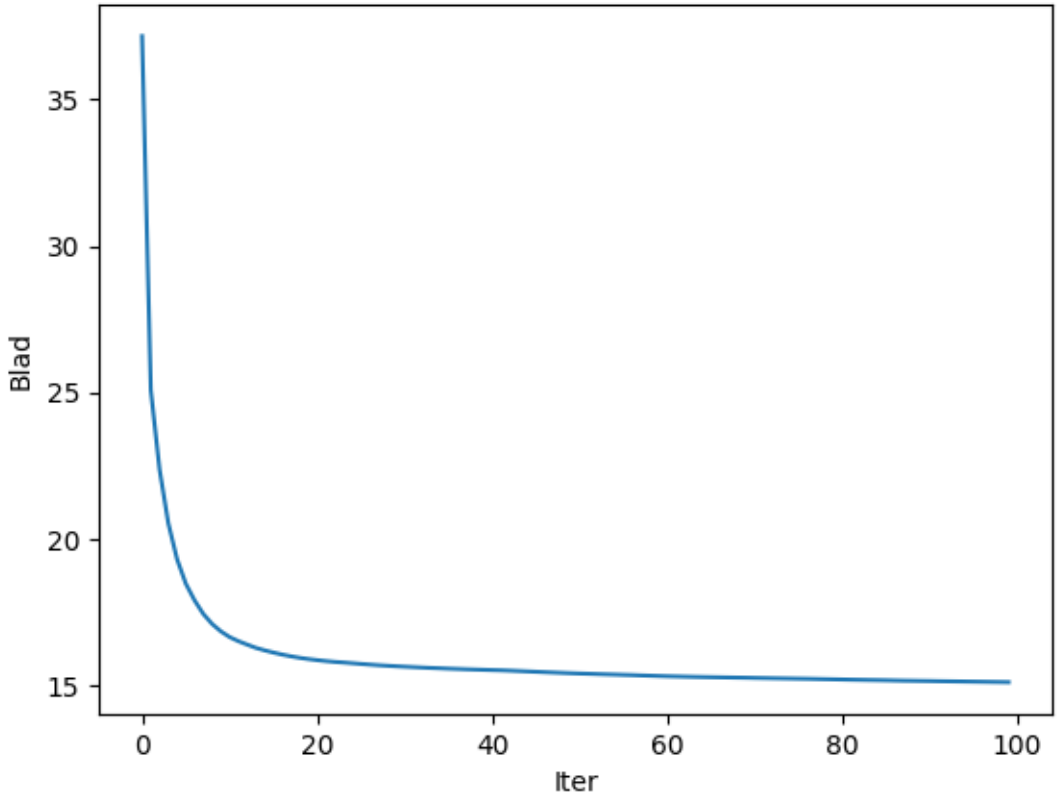
Lepszą metodą optymalizacji jest metoda Adam. Osiąga ona lepsze skuteczności na obu zbiorach, jak i szybciej znajduje optymalne wartości wag.

2.1.3. Szybkość nauki

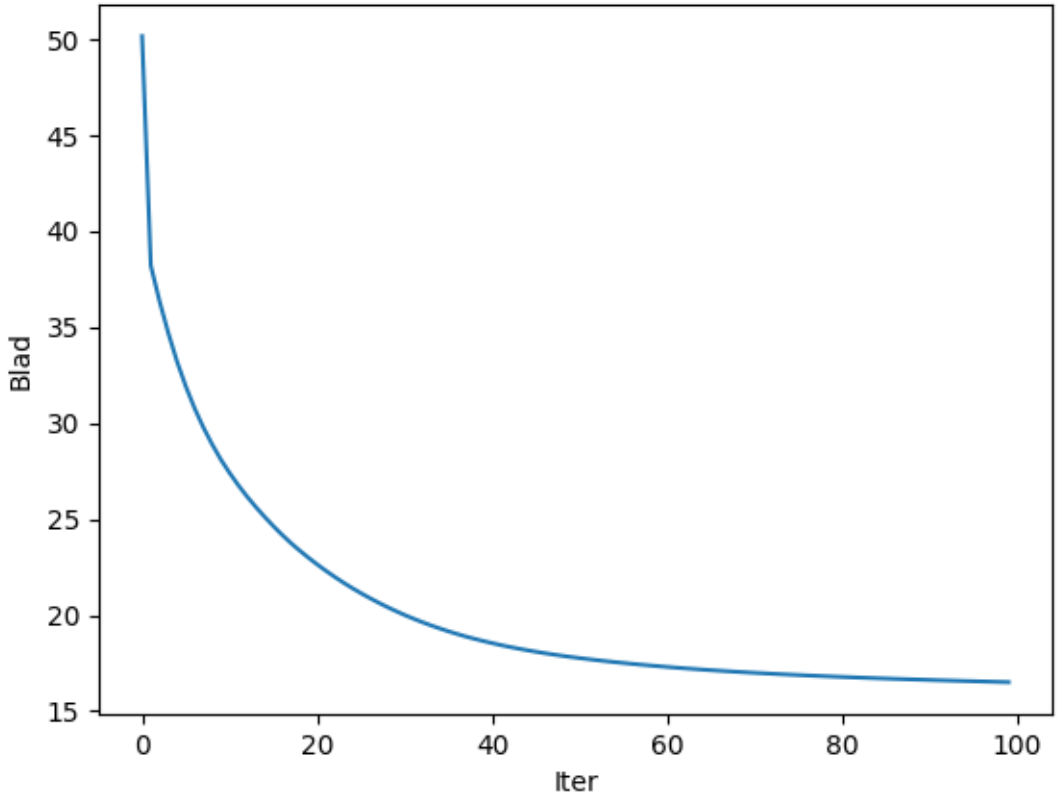
Szybkość nauki	Wynik
0.1	Uczący: 0.778063, Testowy: 0.780105
0.01	Uczący: 0.779221, Testowy: 0.778502
0.001	Uczący: 0.753348, Testowy: 0.753929



Rys. 2.9. Szybkość nauki - 0.1



Rys. 2.10. Szybkość nauki - 0.01



Rys. 2.11. Szybkość nauki - 0.001

Oceniając kształty wykresów dochodzimy do wniosku, że najlepiej sieć neuronowa uczy się dla współczynnika nauki o wartości 0.01.

2.2. Badania dodatkowe

2.2.1. Batch size

Batch size	Wynik
1	Uczący: 0.794332, Testowy: 0.772727
5	Uczący: 0.762704, Testowy: 0.759221
10	Uczący: 0.782476, Testowy: 0.778182
20	Uczący: 0.832388, Testowy: 0.815191
100	Uczący: 0.811075, Testowy: 0.811688

Najlepsze wyniki osiągamy dla batch size o wartości 20. Wybranie batch size o bardzo małych wartościach powoduje bardzo długie liczenie, dla tego kompromisem między szybkością a jakością wyników jest własnie 20.

2.2.2. Iteracje

Iteracje	Wynik
10	Uczący: 0.785016, Testowy: 0.811688
20	Uczący: 0.79316 , Testowy: 0.811688
50	Uczący: 0.802932, Testowy: 0.811688
100	Uczący: 0.833876, Testowy: 0.818182
200	Uczący: 0.869707, Testowy: 0.805195
500	Uczący: 0.941368, Testowy: 0.75974

Najlepsze wyniki wg nas osiągamy dla liczby iteracji (a właściwie epok, uwzględniając batch size) 100. Dla większych wartości iteracji, dokładnie widzimy zjawisko przeuczania, tzn. osiągamy bardzo dobre wyniki dla zbioru uczącego, jednakże złe wyniki dla zbioru testowego.

2.2.3. PCA

Liczba atrybutów	Wynik
3	Uczący: 0.756083, Testowy: 0.770483
4	Uczący: 0.761274, Testowy: 0.782353
5	Uczący: 0.779273, Testowy: 0.726237
6	Uczący: 0.833876, Testowy: 0.818182
7	Uczący: 0.804182, Testowy: 0.741462

Bardzo ciekawe wyniki otrzymujemy dla wykorzystania PCA. Widzimy, że po zredukowaniu wymiarowości do 6 atrybutów, otrzymujemy bardzo dobre wyniki. Nawet po zredukowaniu o jeszcze więcej atrybutów wyniki nadal są stosunkowo przyzwoite. Dowodzi to, że niektóre z atrybutów ze zbioru danych nie niosą ze sobą użytecznej informacji.

2.3. Wnioski końcowe

Po przeprowadzeniu badań, dochodzimy do wniosku, że najlepszą konfiguracją sieci neuronowej dla zadania klasyfikacji cukrzycy jest:
— struktura sieci: 2 warstwy ukryte 50 , 10 neuronów,
— 100 iteracji,
— szybkość uczenia - 0.01,
— batch size - 20 - szybsza nauka oraz dobre wyniki,

— PCA - 6 - nieuwzględnienie części atrybutów powoduje uzyskanie lepszych wyników. Wpływa to również na szybsze nauczanie sieci, ze względu na mniejszy nakład obliczeń.