**Sieci neuronowe**

**Projekt nr.1**

**Prowadzący:**

Dr. Inż. Krzysztof Halawa

**Wykonali:**

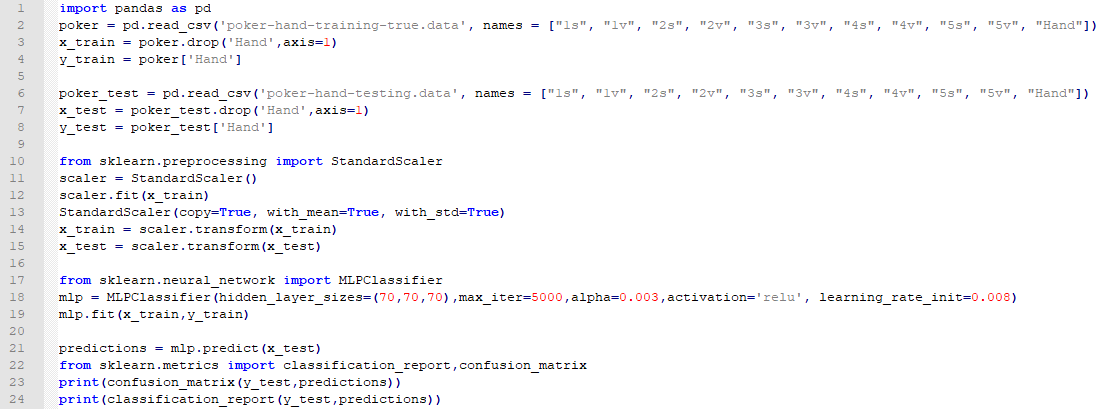
Armand Piecyk 235293

.....

1.Wstęp

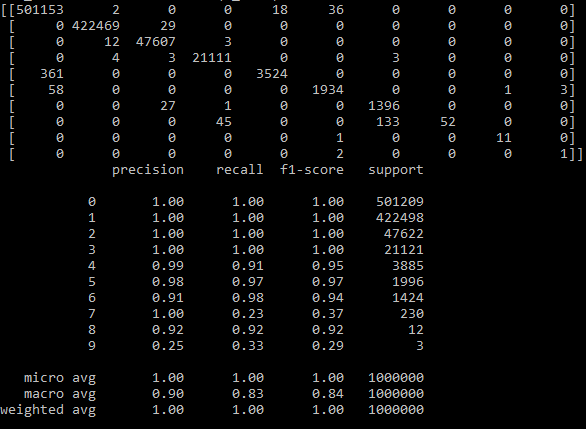
Za zadanie mieliśmy stworzyć program, który poprzez nauke na danych wejściowych będzie w stanie sam sklasyfikować kolejne dane. Nasze dane wzieliśmy ze strony <https://archive.ics.uci.edu/ml/index.php> i nazywają się Poker Hand. Jedna grupa danych treningowych składa się z 11 zmiennych. Każdą z 5 kart opisują 2 zmienne: pierwsza to kolor,a druga to figura. Daje nam to w sumie 10 zmiennych. Jedenasta zmienna opisuje nam wartość (0-9) zestawu 5 kart, gdzie 0-nie znaleziono żadnej pary....9-Poker królewski.

2.Program i testy



*1.Kod programu w języku Python*

Głównym celem naszych testów było uzyskanie zadowalającej precyzji ocenianych danych. W tym momencie wyniki wachają się na granicy 80-90 % poprawnej klasyfikacji. Wyniki byłyby o wiele większe, bo pomimo prawie 100% poprawności oceny rąk dla wartości (0-8) to znikoma ilość pokerów królewskich, które najczęściej dają nam 0% precyzji znacznie zaniżają nam średnią. Na poniższym obrazku proszę zwrócić uwagę na kolumnę precision oraz wartość macro avg.



*2.Wyniki działania programu.*

Testy przeprowadzaliśmy poprzez zmianę lub dodawanie dodatkowych argumentów w funkcji MLPClassifier.

Zmiana argumentu :

1. **hidden\_layer\_sizes**

dla 2 warstw wyniki są w przedziale 60-80%

dla 3/4 warstw wyniki są takie same(80-90 %) dla wartości 40-80, zwiększanie nie powoduje zmian

1. **max\_iter**

dla wartości w granicach 250-15000 nie ma różnicy, z powodów technicznych nie mogę sprawdzić większych wartości.

1. **Alpha**

Dla alpha w przedziale 0.0001-0.03 wyniki są bardzo zbliżone w granicach 85% precyzji.

Dla wartości 0.04 i wyższych otrzymujemy błąd obliczeń.

1. **Learning\_rate\_init**

Wartości w przedziale 0.001-0.01 dają zbliżone wyniki w granicach 80-90%.

Po uruchomieniu programu z wartościa 0.0001 oraz 0.05 otrzymaliśmy 20% precyzji.

1. **Activation**

Sprawdzałem wszystkie możliwości, czyli identity, logistic, tanh i relu z czego ostatni dawał najlepsze wyniki.

1. **Shuffle**

Domyślnie true. Zmiana wartości na false nie zmienia wyników.

1. **Epsilon**

Domyślnie 1e-8. Wartości w przedziale 1e-3 -- 1e-11 nie powodują widocznych zmian precyzji. Dla wartości 1e-1 otrzymaliśmy 20% precyzji, dla 2e-1 40% ,a dla 2e-8 86%.

3. Wnioski

Przetestowałem także argumenty tol,beta\_1 oraz **n\_iter\_no\_change i nie zauważyłem żadnych znaczących różnic precyzji. Uważam, że precyzja na poziomie 80% < jest zadowalająca biorąc pod uwagę, że z większą ilością danych treningowych(chodzi głównie o rękę poker królewski) wynik byłby znacznie wyższy.**