

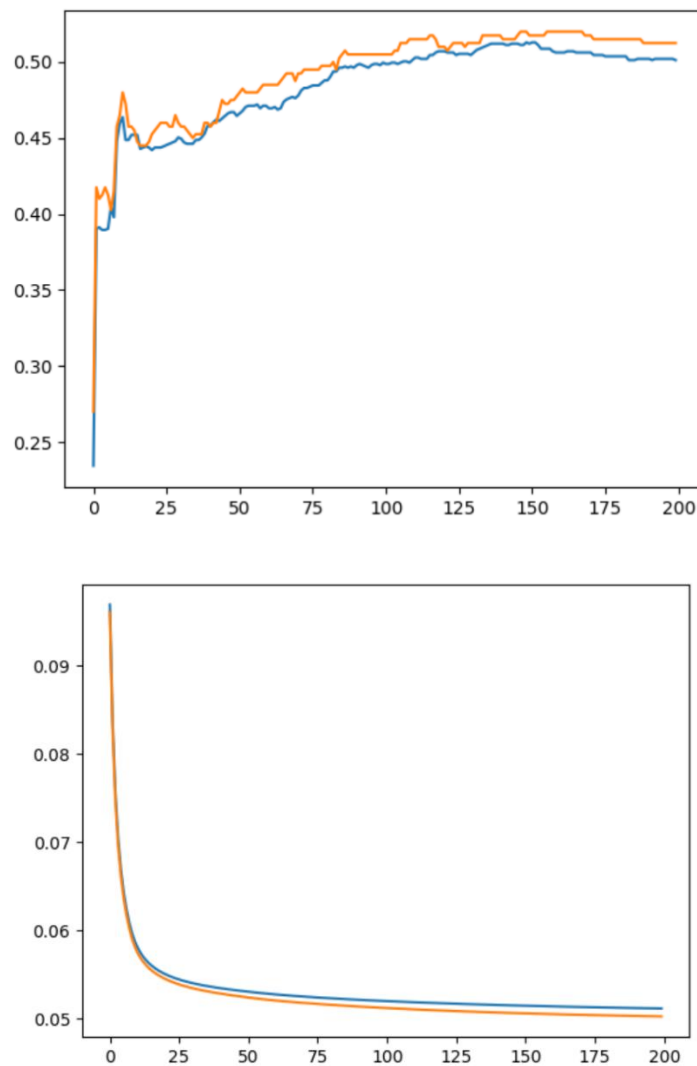
Laboratorium 5 – WSI – Sztuczne sieci neuronowe

Treść zadania:

Zaimplementować sztuczną sieć neuronową z warstwą ukrytą. Implementacja powinna być elastyczna na tyle, żeby można było zdefiniować liczbę neuronów wejściowych, ukrytych i wyjściowych. Wykorzystać sigmoidalną funkcję aktywacji i do trenowania użyć wstecznej propagacji błędów z użyciem metody stochastycznego najszybszego spadku. Sieć nauczyć rozpoznawać jakość wina jak w ćwiczeniu 4. i porównać wyniki z otrzymanymi poprzednio. Na wykresie pokazać jak zmieniał się błąd uczącej sieci w kolejnych epokach. Poeksperymentować ze współczynnikiem uczenia oraz liczbą epok.

Otrzymane wyniki:

Na Rysunku 1 przedstawiono skuteczność działania sieci oraz jej błąd w kolejnych epokach – kolorem niebieskim dla zbioru treningowego, kolorem pomarańczowym dla zbioru testowego (wpuszczonego do sieci po uprzednim trenowaniu jej na danych ze zbioru treningowego). Na Rysunku 2 zamieszczone są parametry, dla jakich sporządzony został wykres.



Rys. 1. Skuteczność działania sieci i błąd sieci w kolejnych epokach.

```

N_INPUT_NEURONS = 11
N_OUTPUT_NEURONS = 6
N_HIDDEN_NEURONS = 50
lr = 0.0001 # learning rate
N_EPOCHS = 200
PROP = 0.75 # size of training set

```

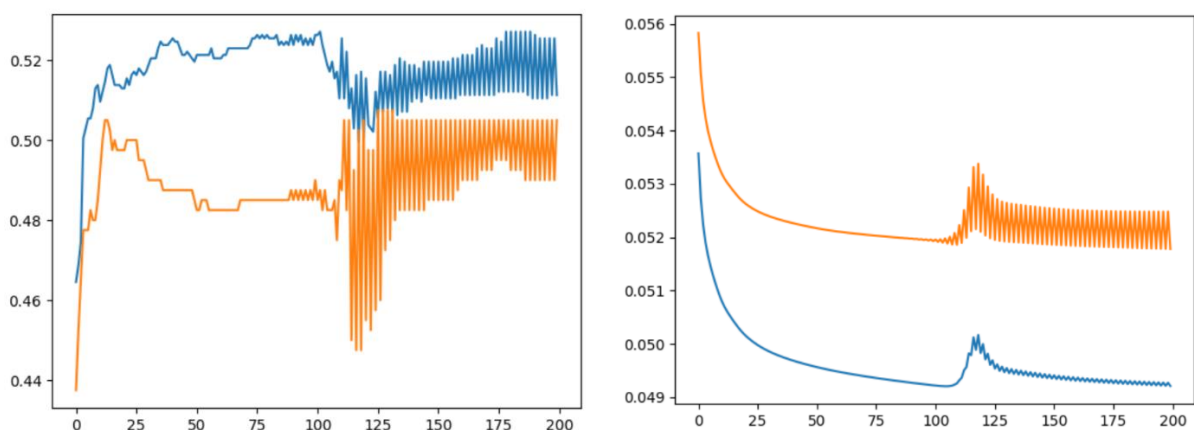
Rys. 2. Parametry programu.

Widać, że skuteczność rozpoznawania jakości wina rośnie, a błąd maleje w miarę przemijania kolejnych epok, co zgodne jest z założeniem uczenia się algorytmu. Widać też, że skuteczność stabilizuje się w okolicach 50-kilku procent.

Obserwacje:

Wpływ współczynnika uczenia oraz liczby epok:

- współczynnik uczenia – został dobrany do sieci metodą inżynierską tak, aby dawał miarodajne wyniki. Wartość zbyt duża powodowała „przestrzeliwanie” szukanego minimum i oscylację naokoło niego (tzw. *overfitting*), zobrazowane na Rysunku 3. Wartość zbyt mała powodowała zbyt wolne uczenie się algorytmu,
- liczba epok – zbyt mała liczba epok nie dawała algorytmowi możliwości nauczenia się do momentu ustabilizowania się jakości sieci, zbyt duża liczba nie prowadziła do znaczących różnic w wynikach (ponieważ sieć stabilizowała się wcześniej), a niepotrzebnie zabierała czas i zasoby komputera (większa liczba epok – dłuższy czas działania programu, więcej wykorzystanych zasobów – ze względu na większą liczbę wykonywanych obliczeń).



Rys. 3. *Overfitting* zobrazowany kolejno na skuteczności oraz błędzie sieci ze zbyt dużym współczynnikiem uczenia.

Wnioski:

Wynik otrzymany z zaimplementowanej sieci neuronowej podobny jest tego otrzymanego w ramach poprzedniego laboratorium (tam: ~55%). Wpływ na to mogą mieć następujące czynniki:

- specyficzny zbiór danych – po poprzednim laboratorium wiemy, że jakość implementowanego klasyfikatora Bayesa nie była zadowalająca, co spowodowane było faktem, że klasy win nie były separowalne (nachodziły na siebie). Ma to wpływ również na działanie sieci neuronowej, ze względu na możliwą niedopasowaną strukturę sieci.
- niedopasowana struktura sieci – przy implementowaniu sieci neuronowej zmieniają się granice obszarów decyzyjnych w stosunku do tych występujących przy klasyfikatorze Bayesa. W przypadku sieci, przestrzeń dzielona jest na obszary decyzyjne o kształtach sympleksów, odpowiadających tu: kolejnym klasom jakości wina. W przypadku klasyfikatora Bayesa obszary mogły być nieliniowe, a ich kształt związany był z największym prawdopodobieństwem osiąganym przez daną klasę. W przypadku sieci dodanie większej ilości warstw ukrytych powoduje podzielenie przestrzeni na większą ilość sympleksów, a co może za tym iść zwiększenie dokładności działania sieci (dzięki możliwości podzielenia przestrzeni na mniejsze, bardziej dokładne obszary). Możliwe, że w przypadku zbioru danych dot. jakości wina zaimplementowanie większej ilości warstw ukrytych przyczyniłoby się do lepszych wyników i wyższej skuteczności sieci.
- zbyt duża ilość klas do przewidzenia – ten fakt łączy się bezpośrednio z poprzednimi punktami. Z obserwacji wynika, że przy podawaniu do sieci zbioru o mniejszej ilości klas do przewidzenia, skuteczność modelu osiąga wyższe wartości. Najprawdopodobniej spowodowane to jest nieseparowalnością klas w zbiorze (na podstawie danych wartości kolejnych cech wina) i zbyt małymi różnicami pomiędzy kolejnymi klasami jakości – jest to przyczyną problemu ze zbudowaniem odpowiedniego modelu przy 1 warstwie ukrytej.