

ROB LAB 5-6 – Adrian Wiśniewski

Klasyfikator tworzę wykorzystując wielowarstwową sieć neuronową uczoną algorytmem wstecznej propagacji błędów. Po kilku próbach stworzyłem czterowarstwową sieć. Warstwa wejściowa liczy 40 neuronów, i podaje oryginalne dane po transformacji PCA. Kolejne dwie warstwy ukryte liczą po 60 neuronów każda. Warstwa wyjściowa zawiera 10 neuronów odpowiadających kolejnym etykiety klasyfikatora. Każdy neuron posiada sigmoidalną funkcję wyjścia i dodatkowe wejście sygnału stałego "1". Wagi początkowe neuronów są losowe z przedziału $[-0.5, 0.5]$. Sieć była uczona przez 10 epok, ze stałą uczenia równą 0.005 i bez ograniczenia na próg błędów. W każdej epoce przykłady podawane na wejście sieci były losowo permutowane. Klasyfikator podejmuje decyzję odpowiadającą maksymalnie wzbudzonemu neuronowi z warstwy wyjściowej, pod warunkiem że wartość na jego wyjściu jest dodatnia i o co najmniej 0.5 większa niż druga największa wartość. Gdy któryś z tych warunków nie jest spełniony, klasyfikator zwraca decyzję wymijającą.

Ufność jest obliczana jako:

$$\max(\text{signal}(\text{sidx}(1))^2 - \sum(\text{signal}(\text{signal} > 0)) + \text{signal}(\text{sidx}(1)), 0)$$

Gdzie signal to wektor wyjść sieci, a $\text{signal}(\text{sidx}(1))$ to wyjście neuronu o największej wartości wyjścia. Funkcja ta od kwadratu największego wyjścia odejmuje sumę pozostałych dodatnich wyjść i odcina wartości ujemne. Wartość funkcji rośnie wraz z siłą wyjścia neuronu odpowiadającego decyzji i maleje wraz z siłą pozostałych neuronów reagujących pozytywnie na próbkę.

Wyniki

Wyniki ogólne dla przestrzeni 40-wymiarowej:

Skuteczność	Błędy	Decyzje wymijające
0.944400	0.016500	0.039100

Macierz pomyłek:

961	0	0	0	0	0	1	2	0	0	16
0	1114	2	0	0	0	3	0	3	0	13
6	0	962	0	1	0	2	7	5	1	48
0	0	2	939	0	1	0	6	3	2	57
1	0	1	0	939	0	3	1	1	8	28
3	0	3	9	0	812	4	1	2	4	54
3	1	1	1	1	4	906	0	5	0	36
0	3	9	0	0	0	0	967	0	5	44
3	0	3	6	3	2	1	2	912	0	42
3	2	0	2	7	0	1	4	5	932	53

Ufność:

	Przykłady poprawnie klasyfikowane	Przykłady źle klasyfikowane
Średnia	0.87767	0.43861
Wariancja	0.060958	0.080359

Implementacja

- [ann labels confidence signals confusionMatrix errors] = annLab(tvec, tlab, tstv, tstl) – funkcja fasadowa wykonująca całe laboratorium.
 - ann – zbudowana sieć neuronowa
 - labes – etykiety przykładów testowych
 - confidence – ufność
 - signals – wyjścia neuronów ostatniej warstwy
 - confusionMatrix – macierz pomyłek
 - errors – błędy klasyfikatora
- ann = annBuild(tvec, hiddenLayerNeurons) – przygotowanie sieci neuronowej o zadanej ilości warst i neuronów w każdej warstwie.
- ann = annTrain(ann_in, tvec, tlab, epochs) – funkcja trenująca sieć neuronową przez zadaną liczbę epok. Może być wołana wiele razy na tej samej sieci.
- [labels confidence signals] = annTest(ann, tstv) – funkcja klasyfikująca dane testowe z użyciem stworzonej wcześniej sieci.
- [confusionMatrix] = utilCreateConfusionMatrix(lab, tstl) – funkcja pomocnicza obliczająca macierz pomyłek na podstawie etykiet uzyskanych klasyfikacji i zbioru testowego.
- [errors] = utilCompErrors(confmx) – funkcja pomocnicza obliczająca ilość błędów na podstawie macierzy pomyłek.