

PERCEPTRON

z.1

theta	avg epochs	x weight	y weight
0.05	2	0.029	0.030
0.2	2	0.109	0.110
0.4	3	0.218	0.219
0.6	4	0.327	0.329
0.8	5	0.437	0.438
0.9	5	0.487	0.489
1.0	7	0.545	0.548
1.2	8	0.655	0.657

Uczenie jest tym szybsze im mniejsza jest wartość theta. Dla większych theta, model musi wypracować nadmiernie większe wagi, co powoduje wydłużenie uczenia.

z.2

weight range	avg epochs	x weight	y weight	bias weight
-1.0..1.0	11.0	0.260	0.260	-0.423
-0.8..0.8	11.4	0.139	0.191	-0.289
-0.5..0.5	7.7	0.072	0.102	-0.132
-0.2..0.2	4.9	0.044	0.053	-0.080
-0.1..0.1	3.6	0.027	0.025	-0.044
-0.05..0.05	2.8	0.015	0.016	-0.024
-0.01..0.01	3.0	0.013	0.019	-0.025

Uczenie jest najszybsze dla wag w granicy -0.05..0.05. Ogólnie można powiedzieć, że im początkowe wagi są bliższe zeru, tym uczenie jest szybsze.

z.3

learn factor	avg epochs	x weight	y weight	bias weight
0.001	14.9	0.023	0.014	-0.034
0.01	3.4	0.022	0.037	-0.049
0.1	2.5	0.123	0.154	-0.213
0.2	3.1	0.268	0.396	-0.516
0.5	3.1	0.708	0.994	-1.260
0.8	3.3	1.247	1.692	-2.261
1	3.6	1.805	2.334	-3.216
2	4.8	4.198	5.379	-7.586

Uczenie jest najszybsze przy alfie 0.1. Prawdopodobnie przy zbyt wysokiej alfie skoki są za duże i program nie może się wyuczyć, a przy zbyt małych alfach nauka jest zbyt wolna.

z.4

type	learn factor	avg epochs	x weight	y weight	bias weight
unipolar	0.01	3.9	0.018	0.028	-0.038
bipolar	0.01	3.7	0.026	0.024	-0.042

Nauka jest minimalnie szybsza dla funkcji aktywacji progowej bipolarnej. Prawdopodobnie ponieważ użycie bipolarnej funkcji progowej wywołuje większy kontrast w wynikach (i uczeniu) co przyspiesza ten proces.

ADALINE

z.1

weight range	avg epochs	reached error	x weight	y weight	bias weight
-1.0..1.0	7.4	0.201	0.726	0.514	-0.479
-0.8..0.8	8.8	0.202	0.712	0.489	-0.460
-0.5..0.5	8.7	0.203	0.723	0.463	-0.441
-0.2..0.2	9.0	0.204	0.725	0.458	-0.435
-0.1..0.1	8.8	0.204	0.723	0.459	-0.436
-0.05..0.05	9.0	0.203	0.725	0.460	-0.437
-0.01..0.01	9.0	0.202	0.726	0.462	-0.439

Przy bipolarnych danych wagi docelowe wynoszą około 0.7 i 0.45, więc też dla większych przedziałów wagowych nauka jest szybsza.

z.2

learn factor	avg epochs	reached error	x weight	y weight	bias weight
0.001	70.6	0.219	0.683	0.407	-0.365
0.002	36.1	0.218	0.688	0.410	-0.369
0.01	8.0	0.213	0.710	0.437	-0.411
0.02	4.7	0.205	0.736	0.479	-0.468
0.03	3.4	0.208	0.748	0.497	-0.497
0.05	2.9	0.197	0.788	0.577	-0.593
0.1	2.0	0.212	0.828	0.672	-0.684

Uczenie jest tym szybsze im współczynnik nauki jest większy, jednak przy większych współczynnikach nauki minimalny osiągnięty błąd jest większy, tj. algorytm uczy się mniej dokładnie.

z.3

error boundary	avg epochs	reached error	x weight	y weight	bias weight
0.30	3.0	0.276	0.661	0.370	-0.347
0.25	4.0	0.225	0.712	0.438	-0.423
0.22	4.8	0.205	0.739	0.478	-0.468
0.21	5.0	0.201	0.743	0.486	-0.478
0.20	5.8	0.192	0.758	0.516	-0.510
0.19	6.6	0.186	0.768	0.539	-0.535
0.18	11.0	0.179	0.789	0.603	-0.602

Im granica błędu jest niższa tym dłużej zajmuje wyuczenie modelu. Tempo przyrostu czasu uczenia przypomina wzrost potęgowy (przy małych zmianach granicy błędu pod koniec, ilość epok uczenia podwaja się)

z.4

Przy odpowiednich parametrach, uczenie Adaline może być szybsze niż dla Perceptronu. Jednak potencjalnie tracimy 100% skuteczność algorytmu - tu akurat algorytm i tak w pełni się wyuczy ale nie widać tego po samej granicy błędu. Oba algorytmy mogą rozwiązać jedynie problemy rozdzielalne liniowo.