Νευρο-Ασαφής Υπολογιστική
Neuro-Fuzzy Computing

ΗΥ418

Διδάσκων –
Δημήτριος Κατσαρός

@ Τμ. ΗΜΜΥ
Πανεπιστήμιο Θεσσαλίας

Διάλεξη 16η

Practice on Backpropagation with momentum

Exercise-12

In ADALINE lecture we proved that the LMS algorithm, (whose performance index is a quadratic function), is stable if the learning rate is less than 2 divided by the maximum eigenvalue of the input correlation matrix **R**

That result is identical to what it holds for the steepest descent algorithm, when applied to a quadratic function; steepest descent is stable if the learning rate is less than 2 divided by the maximum eigenvalue of the Hessian matrix

Show that if a momentum term is added to the steepest descent algorithm there will always be a momentum coefficient that will make the algorithm stable, regardless of the learning rate. Solution in class

Exercise-12: background

Let's consider a quadratic function (the performance index):

$$F(\mathbf{x}) = 1/2\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{d}^T\mathbf{x} + c$$

The gradient of this quadratic function is:

$$\nabla F(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{d}$$

If we now insert this expression into our expression for the steepest descent algorithm (assuming a constant learning rate), we obtain:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\mathbf{g}_k = \mathbf{x}_k - \alpha(\mathbf{A}\mathbf{x}_k + \mathbf{d}) \rightarrow$$

$$\rightarrow \mathbf{x}_{k+1} = [I - \alpha\mathbf{A}]\mathbf{x}_k - \alpha\mathbf{d}$$

This is a linear dynamic system, which will be stable if the eigenvalues of the matrix $[I-\alpha\mathbf{A}]$ are less than one in magnitude

Exercise-12: background

We can express the eigenvalues of this matrix in terms of the eigenvalues of the Hessian matrix **A**. Let $\{\lambda_1,\lambda_2,\ldots,\lambda_n\}$ and $\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_n\}$ be the eigenvalues and eigenvectors of the Hessian matrix. Then:

$$[I-\alpha\mathbf{A}]\mathbf{z}_i = \mathbf{z}_i - \alpha\mathbf{A}\mathbf{z}_i = \mathbf{z}_i - \alpha\lambda_i\mathbf{z}_i = (1-\alpha\lambda_i)\mathbf{z}_i$$

Therefore the eigenvectors of $[I-\alpha\mathbf{A}]$ are the same as the eigenvectors of **A**, and the eigenvalues of $[I-\alpha\mathbf{A}]$ are $(1-\alpha\lambda_i)$

Our condition for the stability of the steepest descent algorithm is then: $|(1-\alpha\lambda_i)| < 1$

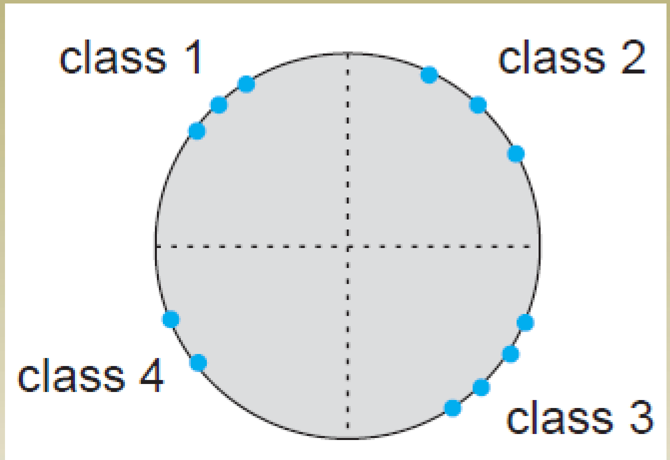
If we assume that the quadratic function has a strong minimum point, then its eigenvalues must be positive numbers. Thus, this reduces to: $\alpha < 2/\lambda_i$

Since this must be true for all the eigenvalues of the Hessian matrix, we get: $\alpha < 2/\lambda_{max}$

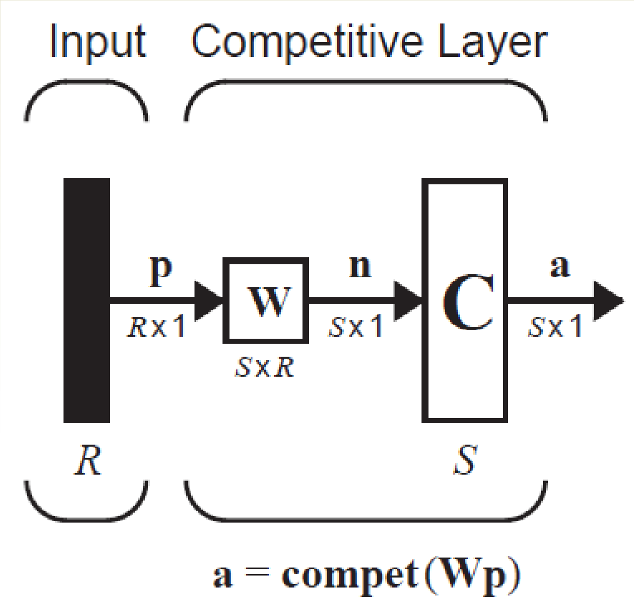
Practice on competitive learning, and Kohonen' learning

Exercise-13

The figure on the right shows several clusters of normalized vectors



Design the weights of the competitive network shown at the right, so that it classifies the vectors according to the classes indicated in the diagram and with the minimum number of neurons



Exercise-13: Bac

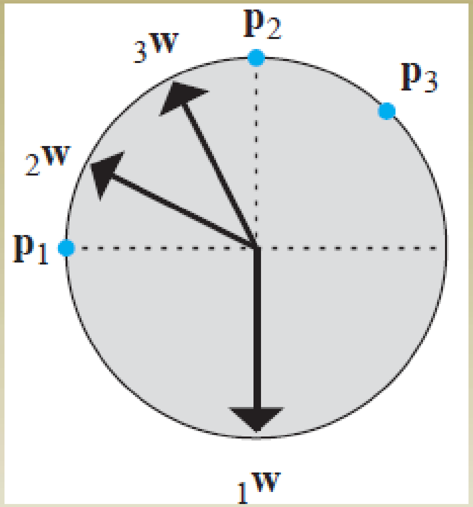
ΠΙΝΑΚΑΣ ΤΡΙΓΩΝΟΜΕΤΡΙΚΩΝ ΑΡΙΘΜΩΝ

Γωνία	ημω	συνω	εφω
0°	0,000	1,000	0,000
1°	0,017	0,999	0,017
2°	0,035	0,999	0,035
3°	0,052	0,999	0,052
4°	0,070	0,998	0,070
5°	0,087	0,996	0,087
6°	0,105	0,395	0,105
7°	0,122	0,993	0,123
8°	0,139	0,990	0,141
9°	0,156	0,988	0,158
10°	0,174	0,985	0,176
11°	0,191	0,982	0,194
12°	0,208	0,978	0,213
13°	0,225	0,974	0,231
14°	0,242	0,970	0,249
15°	0,259	0,966	0,268
16°	0,276	0,961	0,287
17°	0,292	0,956	0,306
18°	0,309	0,951	0,325
19°	0,326	0,946	0,344
20°	0,342	0,940	0,364
21°	0,358	0,934	0,348
22°	0,375	0,927	0,404
23°	0,391	0,921	0,424
24°	0,407	0,914	0,445
25°	0,423	0,906	0,466
28°	0,438	0,899	0,488
27°	0,454	0,891	0,510
28°	0,469	0,883	0,532
29°	0.485	0,875	0,554
30°	0,500	0,866	0,577
31°	0,515	0,857	0,601
32°	0,530	0,848	0,625
33°	0,545	0,839	0,649
34°	0,559	0,829	0,675
35°	0,574	0,819	0,700
38°	0,588	0,809	0,727
37°	0,602	0,799	0,754
38°	0,616	0,788	0,781
39°	0,629	0,777	0,810
40°	0,643	0,766	0,839
41°	0,656	0,755	0,869
42°	0,669	0,743	0,900
43°	0,682	0,731	0,933
44°	0,695	0,719	0,966
45°	0,707	0,707	1,000
46°	0,720	0,695	1,036
47°	0,731	0,682	1,072
48°	0,743	0,669	1,111
49°	0,755	0,656	1,150
50°	0,766	0,643	1,192
51°	0,777	0,629	1,235
52°	0,788	0,616	1.280
53°	0,799	0,602	1,327
54°	0,809	0,588	1,376
55°	0,819	0,574	1,428
56°	0,829	0,559	1,483
57°	0,839	0,545	1,540
58°	0,848	0,530	1,600
59°	0,857	0,515	1,664
60°	0,866	0,500	1,732
61°	0,875	0,485	1,804
62°	0,883	0,470	1,881
63°	0,891	0,454	1,963
64°	0,899	0,438	2,050
65°	0,906	0.423	2,145
66°	0,914	0,407	2,246
67°	0,921	0,391	2,356
68°	0,927	0,375	2,475
69°	0,934	0,358	2,605
70°	0,940	0,342	2,748
71°	0,946	0,326	2,904
72°	0,951	0,309	3,078
73°	0,956	0,292	3,271
74°	0,961	0,276	3,487
75°	0,966	0,259	3,732
76°	0,970	0,242	4,011
77°	0,974	0.225	4,333
78°	0,978	0,203	4,705
79°	0,982	0,191	5,145
80°	0,985	0,174	5,671
81°	0,988	0.156	6,314
82°	0,990	0,139	7,115
83°	0,993	0,122	8,144
84°	0,995	0.105	9,514
85°	0,996	0,087	11,430
86°	0,998	0,070	14,301
87°	0,999	0,052	19,081
88°	0,999	0,035	28,636
89°	0,999	0,018	57,290
90°	1,000	0,000	

σε μοίρες	σε rad	ημω	συνω	εφω	σφω
0°	0	0	1	0	Δεν ορίζεται
30°	$\frac{\pi}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{3}$	$\sqrt{3}$
45°	$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	1	1
60°	$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\sqrt{3}$	$\frac{\sqrt{3}}{3}$
90°	$\frac{\pi}{2}$	1	0	Δεν ορίζεται	0

Exercise-14

The figure shows three input vectors and three initial weight vectors for a three-neuron competitive layer



The values of the input vectors:

$$\mathbf{p}_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{p}_3 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

The initial values of the three weight vectors are:

$$_1\mathbf{w} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, {}_2\mathbf{w} = \begin{bmatrix} -2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}, {}_3\mathbf{w} = \begin{bmatrix} -1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}$$

Calculate the resulting weights found after training the competitive layer with the Kohonen rule and a learning rate of α=0.5, on the following series of inputs: $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$

Exercise-15

Consider the configuration of input vectors and initial weights shown in the figure

Train a competitive network to cluster these vectors using the Kohonen rule with learning rate α=0.5

Find graphically the position of the weights after all of the input vectors (in the order shown, i.e., $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$) have been presented once