Application of ‘ADALINE’ Artificial Neural Network for Decision Making of Classroom Type in Schools at Elementary Level

Amit Kumar Roy,

Department of Computer Science, United College of Engineering and Research, Naini, Prayagraj

Abstract. In this era of Technologies almost none area is untouched .In this view if we talk about the primary education, it can be said that the technology is not so much applied here. In this research paper artificial neural network under deep learning Named as ADALINE (Adaptive Linear Neural Element Network) will be applied for classification of students as ‘regular’ or ‘special’ in schools at elementary level. For the decision of classification reading test and writing test will be performed. The student securing less than the minimum standard score will enter in special class otherwise regular class will be allotted to students. ADALINE makes use of supervised learning where it uses training algorithm and then find the target output.

Keywords: ADALINE, Weight update, Decision making, supervised learning, intelligent system.

Introduction

Technologies like online examination system, online entrance system; online attendance system etc. has been implemented in education field at almost all level. But Artificial intelligence system can be further integrated in existing technology so that the outcome may be more precise, quick and relevant. Although Artificial Intelligence has been also applied in education field but it is also true that it has not been applied at all level specifically at elementary level. As we all know that the elementary level decides the future of a student so we can understand the importance of need of such type of optimized intelligence system to fulfill the requirement.

The decision of classifying the class for a student either under regular or special is still being done manually which obviously have human error. It can be overcome by using an efficient artificial intelligent system definitely.

ADALINE Network

The Adaptive Linear Neural Element Network framed by Bernard Widrow of Stanford University.

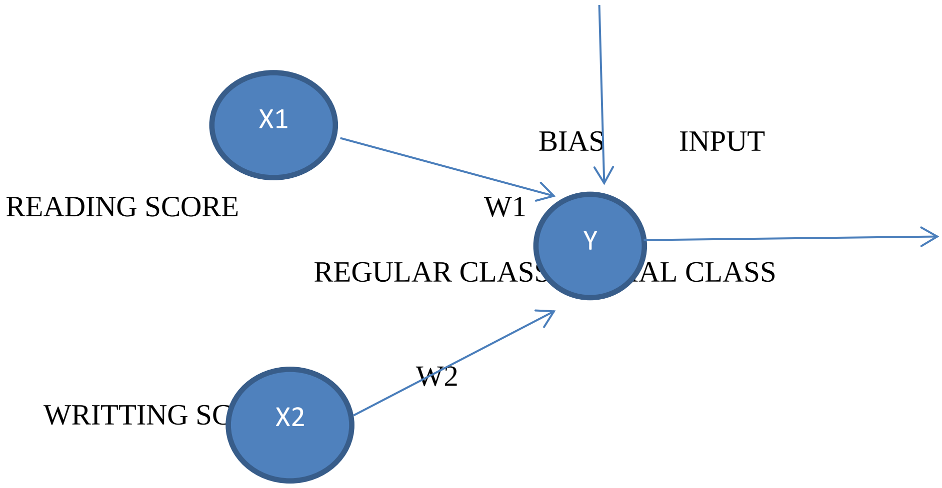

Figure 1. 'ADALINE' Architecture Network.

ADALINE network uses delta rule for weight updation. This rule is also called Widrow – Hoff's delta rule. The rule is given by,
$w_i(new)=w_i(old) + \alpha (t-y)x_i$

In this Artificial intelligent system, two input data in the form of test reading value (X1) and write test value (X2) has been used to produce one output data in the form of decision of suitable class (regular class or special class). The range of input and output data is [1, -1] because the Adaline uses bipolar activation function.

Methodology

The development of this class decision system based on Deep learning starts from collection of input data in the form of reading test and written test score of 25 students from elementary schools. The target output will be known for these input data. After collection of input data the following ADALINE Training algorithm has been applied to gain the target output.

ADALINE Training Algorithm for determining class

Step 1: Weights (w1 and w2) are initialized as a random value (0.1) and bias input (b) as 0.01. Also set learning rate
Parameter (α) = 0.1.
Step 2: Perform steps 3 **– 6** till stopping condition is false.
Step 3: Perform steps **4 – 6** for each bipolar training set.
S : t
Step 4: Find normalized inputs x1 and x2 for all sample units using the min-max normalization:

$$x' = [(u-l)*(x - x_{min)} / (x_{max} - x_{min)}] + l$$

where

- x is original data
- x' is normalized data
- x_{max} and x_{min} are respectively maximum and minimum values of original vectors
- u and l are respectively the upper and lower values of the new range for normalized data. As here (u,l) = (-1,+1)

Step 5: Calculate net input to the output unit.

$\mathbf{y_{in} = b} + \Sigma x_i w_i$ for all sample units

Calculate output y by applying activation function over net input.

Step 6: Update weights and bias as,

For **all sample input units**

$w_i(new)=w_i(old) + \boldsymbol{\alpha}(t-y)x_i$

$b(new)= b(old) + \alpha (t- y)$

Also compute

Least mean square error (E) as

$E= (t-y)^2$

Step 7: Set a condition for total least mean square error, which will be the stopping condition to gain the target output i.e. in each epoch the total least mean square error will be minimized.

The above algorithm is implemented for all input samples and the error is calculated. One EPOCH is completed when all inputs are presented. Summing up all the errors obtained for each input during on EPOCH will give total mean square error of that EPOCH.

Result and Discussion

Column 'X_1' is result of normalization from reading score, Column' X_2' is result of normalization from writing score and target is final result of classroom determination on data learn. Normalization is a process to change the value into the range [1,-1]. Target output is '1' if students enter the regular class, and target output is '-1' if students enter the special class.

Table 1. Data learn classroom determination in elementary school.

Student_Id	Reading Score	Writing Score	Total Score	Target Output (Expected) Class
1	40	55	47.5	special
2	50	50	50	regular
3	40	80	60	regular
4	80	78	79	regular
5	84	10	47	special
6	95	20	57.5	regular
7	80	50	65	regular
8	85	35	60	regular
9	45	70	57.5	regular
10	62	15	38.5	special
11	76	20	48	special
12	25	85	55	regular
13	05	80	42.5	special
14	55	70	62.5	regular
15	55	65	60	regular
16	85	12	48.5	special
17	40	90	65	regular
18	27	28	27.5	special
19	50	60	55	regular
20	92	49	70.5	regular
21	58	34	46	special
22	62	68	65	regular
23	55	57	56	regular
24	30	65	47.5	special
25	60	45	52.5	regular

Table 2. Normalized input pairs calculation for 25 data.

Student_Id	X1	X2	Target
1	-0.2222	0.1111	-1
2	0	0	1
3	-0.2222	0.6666	1
4	0.6666	0.6222	1
5	0.7556	-0.8888	-1
6	1	-0.5556	1
7	0.6666	0	1
8	0.7778	-0.3333	1
9	-0.1111	0.4444	1
10	0.2666	-0.7778	-1
11	0.5778	-0.6666	-1
12	-0.5555	0.7778	1
13	-1	0.6666	-1
14	0.1111	0.4444	1
15	0.1111	0.3333	1
16	0.7778	-0.8444	-1
17	0.2222	0.8478	1
18	-0.5111	-0.4889	-1
19	0	0.2222	1
20	0.9333	-0.0222	1
21	0.1778	-0.3555	-1
22	0.2667	0.40	1
23	0.1111	0.1555	1
24	-0.4444	0.3333	-1
25	0.2222	-0.1111	1

Inputs						Weight change			Initial Weights w1=0.1,w2 = 0.1, b = 0.01			
x1	**x_2**	**b**	**Target (t)**	**Net Input (y_{in})**	**($t - y_{in}$)**	$\Delta w_1 = \alpha.(t - y_{in}).x_1$	$\Delta w_2 = \alpha.(t - y_{in}).x_2$	$\Delta b = \alpha.(t - y_{in})$	$w_1(new) = w_1(old) + \Delta w1$	$w_2(new) = w_2(old) + \Delta w2$	**b (new) = b(old) + Δb**	**Error $(t - y_{in})^2$**
EPOCH-1												
-0.2222	0.1111	0.01	-1	-0.0011	-0.9989	0.0222	-0.0111	-0.0998	0.1222	0.0889	-0.0898	0.9978
0	0	-0.0898	1	-0.0898	1.0898	0	0	0.1089	0.1222	0.0889	0.0191	1.1877
-0.2222	0.6666	0.0191	1	0.0512	0.9488	-0.0211	0.0632	0.0948	0.0678	0.1521	0.1139	0.9002
0.6666	0.6222	0.1139	1	0.2537	0.7463	0.0497	0.0464	0.0746	0.1175	0.1985	0.1885	0.5569
0.7556	-0.8888	0.1885	-1	0.1008	-1.1008	-0.0832	0.0978	-0.1101	0.0343	0.2963	0.0784	1.2117
1	-0.5556	0.0784	1	-0.0519	1.0519	0.1052	-0.0583	0.1051	0.1395	0.2380	0.1835	1.1064
0.6666	0	0.1835	1	0.2764	0.7236	0.0482	0	0.0723	0.1877	0.2380	0.2558	0.5236
0.7778	-0.3333	0.2558	1	0.3225	0.6775	0.0526	-0.0225	0.0677	0.2403	0.2155	0.3235	0.4590
-0.1111	0.4444	0.3235	1	0.3926	0.6074	-0.0067	0.0269	0.0607	0.2336	0.2424	0.3842	0.3689
0.2666	-7778	0.3842	-1	0.2579	-1.2579	-0.0335	0.0978	-0.1258	0.2001	0.3402	0.2584	1.5823
0.5778	-0.6666	0.2584	-1	0.1472	-1.1472	-0.0663	0.0765	-0.1147	0.1338	0.4167	0.1437	1.3161
-0.5555	0.7778	0.1437	1	0.3935	0.6065	-0.0336	0.0471	0.0606	0.1002	0.4608	0.2043	0.3678
-1	0.6666	0.3678	-1	0.4113	-1.4113	0.1411	-0.0941	-0.1411	0.2413	0.3667	0.0632	1.9917
0.1111	0.4444	0.0632	1	0.2529	0.7471	0.0083	0.0332	0.0747	0.2496	0.3999	0.1379	0.5581
0.1111	0.3333	0.1379	1	0.2989	0.7011	0.0078	0.0233	0.0701	0.2574	0.4232	0.2080	0.4915
0.7778	-0.8444	0.2080	-1	0.0508	-1.0508	-0.0817	0.0887	-0.1051	0.1757	0.5119	0.1029	1.1041
0.2222	0.8478	0.1029	1	0.5759	0.4241	0.0094	0.0359	0.0424	0.1851	0.5478	0.1453	0.1798
-0.5111	-0.4889	0.1453	-1	-0.2171	-0.7829	0.0400	0.0383	-0.0783	0.2251	0.5861	0.0670	0.6129
0	0.2222	0.0670	1	0.1972	0.8028	0	0.0178	0.0803	0.2251	0.6039	0.1473	0.6445

0.9333	-0.0222	0.1473	1	0.3439	0.6561	0.0612	-0.0015	0.0656	0.2863	0.6024	0.2129	0.4304
0.1778	-0.3555	0.2129	-1	0.0496	-1.0496	-0.0186	0.0373	-0.1049	0.2677	0.6397	0.1080	1.1016
0.2667	0.40	0.108	1	0.4353	0.5647	0.0151	0.0226	0.0565	0.2828	0.6623	0.1645	0.3189
0.1111	0.1555	0.1645	1	0.2989	0.7011	0.0078	0.0109	0.0701	0.2906	0.6732	0.2346	0.4915
-0.4444	0.3333	0.2346	-1	0.1394	-1.1394	0.0506	-0.0379	-0.1139	0.3412	0.6353	0.1207	1.2982
0.2222	-0.1111	0.1207	1	0.1259	0.8741	0.0194	-0.0097	0.0874	0.3606	0.6256	0.2081	0.7641

EPOCH-2												
-0.2222	0.1111	0.2081	-1	0.1974	-1.1974	0.0266	-0.0133	-0.1197	0.3872	0.6123	0.0884	1.4337
0	0	0.0884	1	0.0884	0.9116	0	0	0.0912	0.3872	0.6123	0.1796	0.8310
-0.2222	0.6666	0.1796	1	0.5017	0.4983	-0.0111	0.0332	0.0498	0.3761	0.6455	0.2294	0.2483
0.6666	0.6222	0.2294	1	0.8817	0.1183	0.0079	0.0073	0.0118	0.3840	0.6528	0.2412	0.0139
0.7556	-0.8888	0.2412	-1	-0.0488	-0.9512	-0.0718	0.0845	-0.0951	0.3122	0.7373	0.1461	0.9047
1	-0.5556	0.1461	1	0.0486	0.9514	0.0951	-0.0528	0.0951	0.4073	0.6845	0.2412	0.9051
0.6666	0	0.2412	1	0.5127	0.4873	0.0325	0	0.0487	0.4398	0.6845	0.2899	0.2374
0.7778	-0.3333	0.2899	1	0.4038	0.5962	0.0464	-0.0199	0.0596	0.4862	0.6646	0.3495	0.3554
-0.1111	0.4444	0.3495	1	0.5908	0.4092	-0.0045	0.0182	0.0409	0.4817	0.6828	0.3904	0.1674
0.2666	-0.7778	0.3904	-1	-0.0123	-0.9877	-0.0263	0.0768	-0.0988	0.4554	0.7596	0.2916	0.9755
0.5778	-0.6666	0.2916	-1	0.0484	-1.0484	-0.0605	0.0698	-0.1048	0.3949	0.8294	0.1868	1.0991
-0.5555	0.7778	0.1868	1	0.6125	0.3875	-0.0215	0.0301	0.0387	0.3734	0.8595	0.2255	0.1501
-1	0.6666	0.2255	-1	0.4250	-1.4250	0.1425	-0.0945	-0.1425	0.5159	0.7650	0.0830	2.0306
0.1111	0.4444	0.083	1	0.4803	0.5197	0.0058	0.0231	0.0519	0.5217	0.7881	0.1349	0.2701
0.1111	0.3333	0.1349	1	0.4555	0.5445	0.0060	0.0181	0.0544	0.5277	0.8062	0.1893	0.2965
0.7778	-0.8444	0.1893	-1	-0.0810	-0.919	-0.0715	0.0776	-0.0919	0.4562	0.8838	0.0974	0.8445
0.2222	0.8478	0.0974	1	0.9481	0.0519	0.0011	0.0044	0.0052	0.4573	0.8882	0.1026	0.0027
-0.5111	-0.4889	0.1026	-1	-0.5654	-0.4346	0.0222	0.0213	-0.0435	0.4795	0.9095	0.0591	0.1888
0	0.2222	0.0591	1	0.2612	0.7388	0	0.0164	0.0739	0.4795	0.9259	0.1330	0.5458
0.9333	-0.0222	0.1330	1	0.3748	0.6252	0.0583	-0.0014	0.0625	0.5378	0.9245	0.1955	0.3908

		b	t	yin	t-yin	>w1	>w2	>b	W1	W2	B	
0.1778	-0.3555	0.1955	-1	-0.0375	-0.9625	-0.0171	0.0342	0.0962	0.5207	0.9587	0.2917	0.9264
0.2667	0.40	0.2917	1	0.8140	0.1860	0.0049	0.0074	0.0186	0.5256	0.9661	0.3103	0.0346
0.1111	0.1555	0.3103	1	0.5189	0.4811	0.0053	0.0075	0.0481	0.5309	0.9736	0.3584	0.2314
-0.4444	0.3333	0.3584	-1	0.4469	-1.4469	0.0643	-0.0482	-0.1447	0.5952	0.9254	0.2137	2.0935
0.2222	-0.1111	0.2137	1	0.2431	0.7569	0.0168	-0.0084	0.0757	0.6120	0.9170	0.2894	0.5729

EPOCH-3												
-0.2222	0.1111	0.2894	-1	0.2553	0.7447	-0.0165	0.0083	0.0745	0.5955	0.9253	0.3639	0.5546
0	0	0.3639	1	0.3639	0.6361	0	0	0.0636	0.5955	0.9253	0.4275	0.4046
-0.2222	0.6666	0.4275	1	0.9119	0.0881	-0.0019	0.0058	0.0088	0.5936	0.9331	0.4363	0.0077
0.6666	0.6222	0.4363	1	1.4125	-0.4125	-0.0275	-0.0256	-0.0412	0.5661	0.9075	0.3951	0.1701
0.7556	-0.8888	0.3951	-1	0.0162	-1.0162	-0.0767	0.0903	-0.1016	0.4894	0.9978	0.2935	1.0326
1	-0.5556	0.2935	1	0.2285	0.7715	0.0771	-0.0428	0.0771	0.5665	0.9550	0.3706	0.5952
0.6666	0	0.3706	1	0.7482	0.2518	0.0168	0	0.0252	0.5833	0.9550	0.3958	0.0634
0.7778	-0.3333	0.3958	1	0.5312	0.4688	0.0364	-0.0156	0.0468	0.6197	0.9394	0.4426	0.2197
-0.1111	0.4444	0.4426	1	0.7912	0.2088	-0.0023	0.0092	0.0208	0.6174	0.9486	0.4634	0.0436
0.2666	-0.7778	0.4634	-1	-0.1098	-0.8902	-0.0237	0.0692	-0.0890	0.5937	1.0178	0.3744	0.7924
0.5778	-0.6666	0.3744	-1	0.0389	-1.0389	-0.0600	0.0692	-0.1039	0.5337	1.0870	0.2705	1.0793
-0.5555	0.7778	0.2705	1	0.8195	0.1805	-0.0099	0.0140	0.0180	0.5238	1.1010	0.2885	0.0326
-1	0.6666	0.2885	-1	0.4986	-1.4986	0.1498	-0.0998	-0.1498	0.6736	1.0012	0.1387	2.2458
0.1111	0.4444	0.1387	1	0.6585	0.3415	0.0038	0.0151	0.0341	0.6774	1.0163	0.1728	0.1166
0.1111	0.3333	0.1728	1	0.5868	0.4132	0.0046	0.0138	0.0413	0.6820	1.0301	0.2141	0.1707
0.7778	-0.8444	0.2141	-1	-0.1252	-0.8748	-0.0680	0.0738	-0.0875	0.6140	1.1039	0.1266	0.7653
0.2222	0.8478	0.1266	1	1.1989	-0.1989	-0.0044	-0.0169	-0.0199	0.6096	1.0870	0.1067	0.0395
-0.5111	-0.4889	0.1067	-1	-0.7363	-0.2637	0.0135	0.0129	-0.0264	0.6231	1.0999	0.0803	0.0695
0	0.2222	0.0803	1	0.3247	0.6753	0	0.0149	0.0675	0.6231	1.1148	0.1478	0.4560
0.9333	-0.0222	0.1478	1	0.7046	0.2954	0.0275	-0.0007	0.0295	0.6506	1.1141	0.1773	0.0873

0.1778	-0.3555	0.1773	-1	-0.1031	-0.8969	-0.0159	0.0319	-0.0897	0.6347	1.1460	0.0876	0.8044
0.2667	0.40	0.0876	1	0.7153	0.2847	0.0076	0.0114	0.0285	0.6423	1.1574	0.1161	0.0810
0.1111	0.1555	0.1161	1	0.3674	0.6326	0.0070	0.0098	0.0633	0.6493	1.1672	0.1794	0.4002
-0.4444	0.3333	0.1794	-1	0.2799	-1.2799	0.0568	-0.0426	-0.1279	0.7061	1.1246	0.0515	1.6381
0.2222	-0.1111	0.0515	1	0.0834	0.9166	0.0204	-0.0102	0.0917	0.7265	1.1144	0.1432	0.8401

Conclusion

The results and the discussion concludes that, ADALINE based intelligent system can be used in the elementary schools for classification of special class and regular class for students by entering the value of read data and write data, which can facilitate the school in determining the type of class more quickly and precisely. In this system, the calculation begins with determining the value of x1 and x2 as the result of normalizing the input data to convert the value into the range [1, -1], then the value will be the basis for searching change of weights, new weight and final weight similarly for change in bias input and getting final bias input. According to the calculations done in Epochs 1, 2, and 3, different weight results using the same x1 and x2 values have been obtained. The epoch calculation will stop when total mean square error minimizes epoch wise. Based on 25 data learn, the following result has been computed:

EPOCH	TOTAL MEAN SQUARE ERROR
1	20.56
2	15.75
3	12.71

So it can be clearly seen that every time the total mean square error is minimizing,which is the indication of reducing the gap between the required output and computed output, fulfilling the main objective of this research study.

References

[1] Handayani A N 2009 Perbandingan Model PERCEPTRON dan ADALINE pada Fungsi Logika AND fan OR menggunakan Artificial Neural Network Seminar Nasional Elecctrical, Informatics and Its Education

[2] Jong Jek Siang 2009 Jaringan Syaraf Tiruan & Pemrograman Menggunakan MATLAB (Yogyakarta, Indonesia: ANDI)

[3] Kusumoputro B 2009 Metode Pembelajaran Computational Intelligence1-20

[4] Handayani, A. N., A. D. Aindra, D. F. Wahyulis, S. Pathmantara, and R. A. Asmara. "Application of adaline artificial neural network for classroom determination in elementary school." In IOP Conference Series: Materials Science and Engineering, vol. 434, no. 1, p. 012030. IOP Publishing, 2018.