Neuro-Fuzzy Computing
1st Problem Set

Ioannis Roumpos 2980
Konstantinos Vermisoglou 2988
Nikos Gkagkosis 3079

December 22, 2023

Contents

1 Problem - 01

$f(x, y) = x^2 + 4xy + y^2$

Calculating the gradient $\nabla f(x)$

$$\nabla f(x) = \begin{pmatrix} 2x + 4y \\ 2y + 4x \end{pmatrix}$$

The first step is to calculate the partial producers. At the point where it produces zero is the critical point at which there will be either a maximum or a minimum or a saddle point 2x + 4y=0 and 2y + 4x=0 $\rightarrow$ x = y = 0. So the critical point is the x , y = (0,0).After that the next step is to calculate the Hessian.

Calculating the Hessian $\nabla^2 f(x)$

$$\nabla^2 f(x) = \begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix}$$

The calculation of the eigenvalues follows.

$$P(\lambda) = det(A - I\lambda) = \begin{pmatrix} 2 - \lambda & 4 \\ 4 & 2 - \lambda \end{pmatrix} = 4 - 2\lambda + \lambda^2 - 16$$

P(λ) = 0 $\Rightarrow$ λ_1 = 2.6 , λ_2 = -4.6.

We can observe that λ_1 is positive but λ_2 is negative so the Hessian matrix is indefinite and there is a saddle point in (x,y) = (0,0).

Quadratic form:

$$f(x, y) = \frac{1}{2}\, x^T Ax + d^T x + c$$

$$A = Hessian = \begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix}$$

$$\nabla f(x) = Ax + d = \begin{pmatrix} 2 \\ 2 \end{pmatrix} [x, y] + \begin{pmatrix} 4y \\ 4x \end{pmatrix}$$

$$d = \begin{pmatrix} 4y \\ 4x \end{pmatrix}$$

$$f(x, y) = \frac{1}{2}\ x^T \begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix} x + \begin{pmatrix} 4y \\ 4x \end{pmatrix} x + c$$

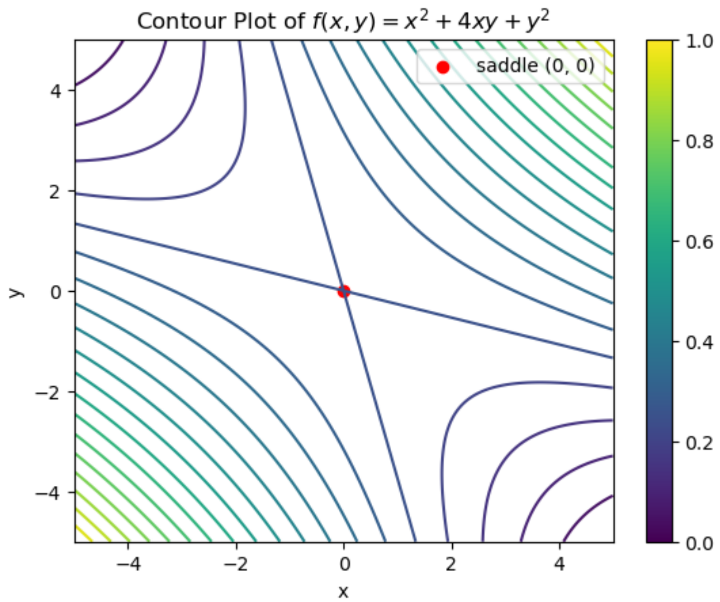


Figure 1: Saddle point.

2 Problem - 02

$f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad x_0 = (-9.5, 9.5)$

Calculating the gradient $\nabla f(x_0)$

$$\nabla f(x) = \begin{pmatrix} 2(x_1 + 2x_2 - 7) + 4(2x_1 + x_2 - 5) \\ 4(x_1 + 2x_2 - 7) + 2(2x_1 + x_2 - 5) \end{pmatrix}$$

$\frac{\partial f(x_0)}{\partial x_1} = -53$

$\frac{\partial f(x_0)}{\partial x_2} = -19$

$$\nabla f(x_0) = \begin{pmatrix} -53 \\ -19 \end{pmatrix}$$

Calculating the direction of descending $s_0 = -\frac{\nabla f(x_0)}{||\nabla f(x_0)||}$

$||\nabla f(x_0)|| = \sqrt{\nabla^T f(x) \nabla f(x)} = \sqrt{\begin{pmatrix} -53 & -19 \end{pmatrix} \begin{pmatrix} -53 \\ -19 \end{pmatrix}} = \sqrt{3170} = 56.3$

$s_0 = -\frac{\nabla f(x_0)}{||\nabla f(x_0)||} = -\frac{1}{56.3} \begin{pmatrix} -53 \\ -19 \end{pmatrix} = \begin{pmatrix} 0.941 \\ 0.337 \end{pmatrix}$

Next point x_1:

$x_1 = (-9.5 - \lambda_0 * 0.941, 9.5 - \lambda_0 * 0.337)$

Replacing the new point in the original function f:

$f(x_1) = (-9.5 - \lambda_0 * 0.941 + 2(9.5 - \lambda_0 * 0.337) - 7)^2 + (2(-9.5 - \lambda_0 * 0.941) + 9.5 - \lambda_0 * 0.337 - 5)^2$

$\frac{\partial f(x_1)}{\partial \lambda_0} = 2[-9.5 - \lambda_0 * 0.941 + 2(9.5 - \lambda_0 * 0.337) - 7](-0.941 - 2 * 0.337) + 2[2(-9.5 - \lambda_0 * 0.941) + 9.5 - \lambda_0 * 0.337 - 5](-2 * 0.941 - 0.337)$

$\frac{\partial f(x_1)}{\partial \lambda_0} = 0 \Rightarrow \lambda_0 = -3.73$

Replacing λ_0 in x_1 point: $x_1 = (-5.99, 10.75)$

Starting the second iteration

Calculating the gradient $\nabla f(x_1)$

$\nabla f(x_1) = \begin{pmatrix} -7.9 \\ 21.58 \end{pmatrix}$

Calculating the direction of descending $s_1 = -\frac{\nabla f(x_1)}{||\nabla f(x_1)||}$

$||\nabla f(x_1)|| = \sqrt{\nabla^T f(x) \nabla f(x)} = \sqrt{\begin{pmatrix} -7.9 & 21.58 \end{pmatrix} \begin{pmatrix} -7.9 \\ 21.58 \end{pmatrix}} = \sqrt{528.106} = 22.98$

$s_1 = -\frac{\nabla f(x_1)}{||\nabla f(x_1)||} = -\frac{1}{22.98} \begin{pmatrix} -7.9 \\ 21.58 \end{pmatrix} = \begin{pmatrix} 0.344 \\ -0.939 \end{pmatrix}$

Next point x_2:

$x_2 = (-5.99 - \lambda_1 * 0.344, 10.75 + \lambda_1 * 0.939)$

Replacing the new point in the original function f:

$$f(x_2) = (-5.99 - \lambda_1 * 0.344 + 2(10.75 + \lambda_1 * 0.939) - 7)^2 + (2(-5.99 - \lambda_1 * 0.344) + 10.75 + \lambda_1 * 0.939 - 5)^2$$

$$\frac{\partial f(x_2)}{\partial \lambda_1} = 2[-5.99 - \lambda_1 * 0.344 + 2(10.75 + \lambda_1 * 0.939) - 7](-0.344 + 2 * 0.939) + 2[2(-5.99 - \lambda_1 * 0.344) + 10.75 + \lambda_1 * 0.939 - 5](-2 * 0.344 + 0.939)$$

$$\frac{\partial f(x_2)}{\partial \lambda_1} = 0 \Rightarrow \lambda_1 = -4.755$$

Replacing λ_1 in x_2 point: $x_1 = (-4.35, 6.28)$

k	x_k^T	λ_k	$\nabla^T f(x_k)$	f(x_k)
0	(-9.5,9.5)	-3.73	(-53,19)	216.5
1	(-5.99,10.75)	-4.755	(-7.9,21.58)	111.23
2	(-4.35,6.28)		(-27.26,-10)	56.52

3 Problem - 03

In the dynamical system called Henon map, depending control or gain parameters, we can observe the transition of it in an aperiodic (chaos) state or a periodic one. For initial values *(a,b) = (0.3,0.4)* we plot the trajectories of the system for different initial values $x_0 = 0$ and $x_0 = 0.000001$. In the figures below, we can see that the trajectories are **periodic**.

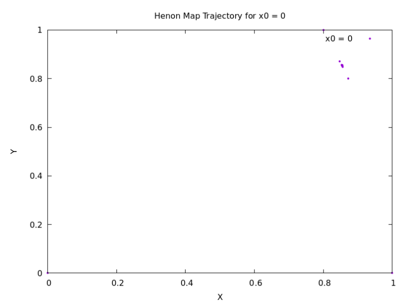


Figure 2: Trajectory with $x_0 = 0.0$

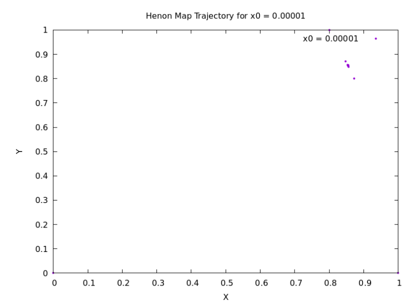


Figure 3: Trajectory with $x_0 = 0.000001$

A. For different values of a in the following figures it can be observed that the trajectories are **fixed points** and the system is not transitioning to chaos. It is more close to a periodic one.

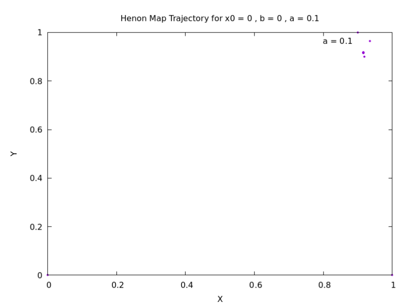


Figure 4: Trajectory with a = 0.1

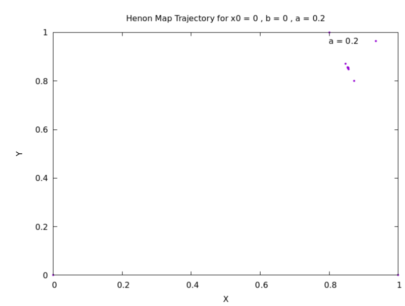


Figure 5: Trajectory with a = 0.2

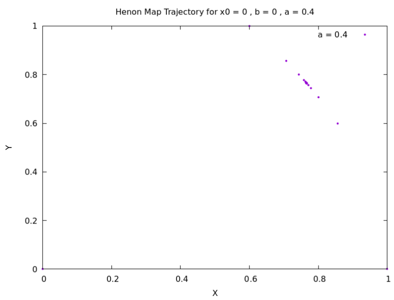


Figure 6: Trajectory with a = 0.4

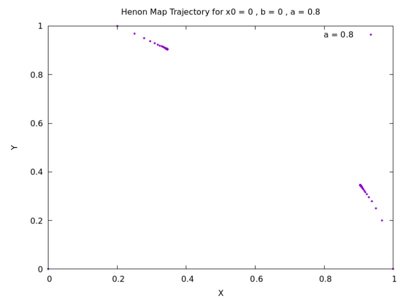


Figure 7: Trajectory with a = 0.8

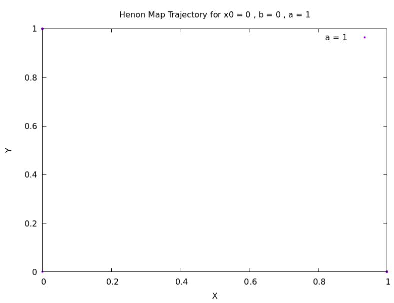


Figure 8: Trajectory with a = 1

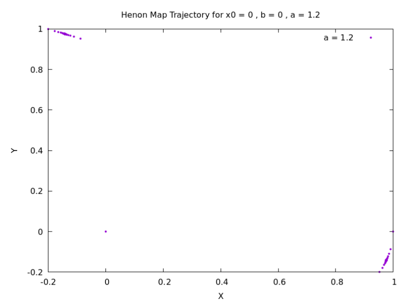


Figure 9: Trajectory with a = 1.2

B. For the next system conditions we assume that $a = 0.25$ and we assign different values for b. The system, as the following figures dictate, while the b increases we observe a transition from a periodic condition to fixed points but not ending up in a chaos like position.

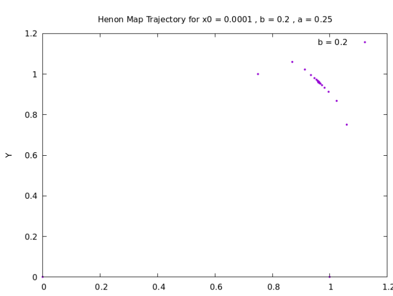


Figure 10: Trajectory with b = 0.2

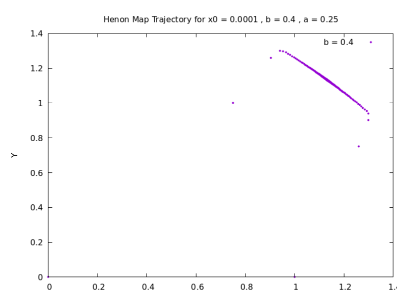


Figure 11: Trajectory with b = 0.4

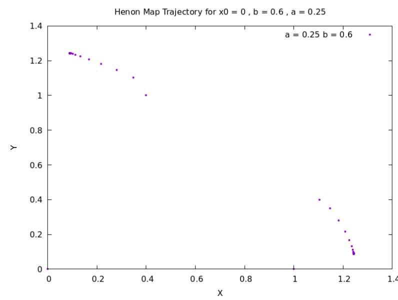


Figure 12: Trajectory with b = 0.6

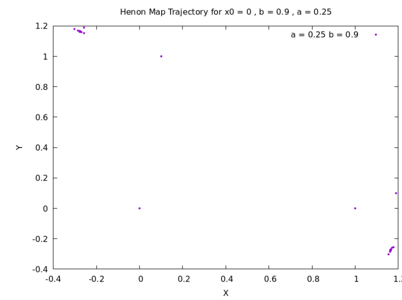


Figure 13: Trajectory with b = 0.9

C. The combination *(a,b) = (0.3675,0.3)* seems to create a system with many **fixed points** and the transition to chaos is not the case for this control and gain parameters.

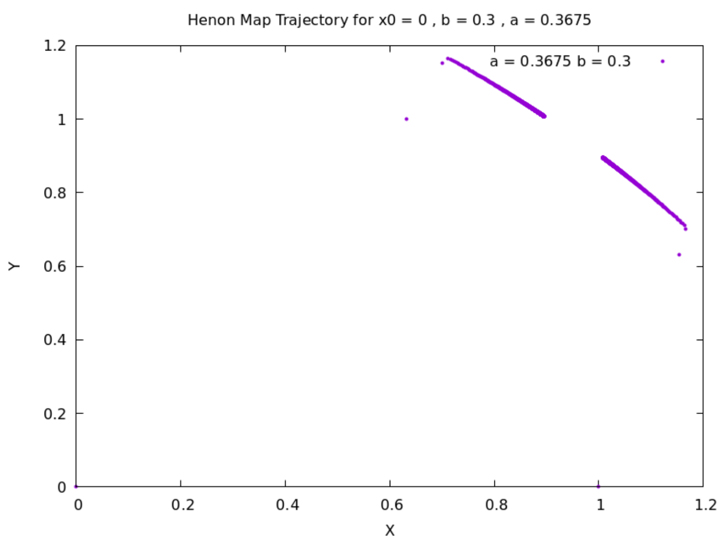


Figure 14: Trajectory with a = 0.3675, b = 0.3

D. As the control parameter a increases we observe a more **chaotic behavior** from the dynamical system. From fixed points, the systems transitions to a seemingly periodic one and then to a chaotic.

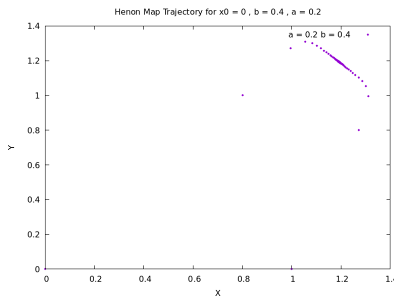


Figure 15: Plot with a = 0.2, b = 0.4

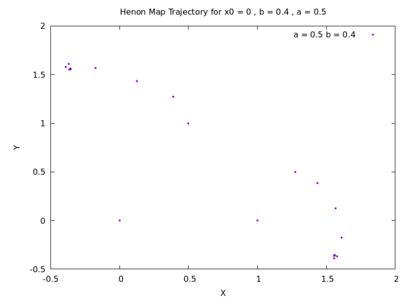


Figure 16: Plot with a = 0.5, b = 0.4

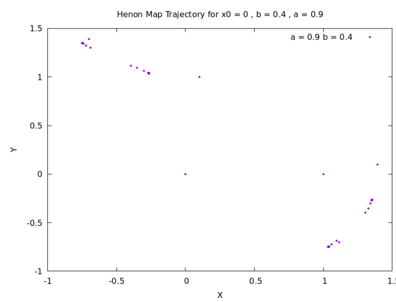


Figure 17: Plot with a = 0.9, b = 0.4

4 Problem - 04

Expressing the derivative of an activation function in terms of the original is an important modification. This turns out to be a convenient form for efficiently calculating gradients used in neural networks: if one keeps in memory the feed-forward activations of the function for a given layer, the gradients for that layer can be evaluated using simple multiplication and subtraction rather than performing any re-evaluating the function itself, which requires extra exponentiation.

LogSig

$S_A = \frac{1}{1+e^{-x}}$
$S'_A = (\frac{1}{1+e^{-x}})' = \frac{e^x}{(e^{-x}+1)^2} = \frac{1}{1+e^{-x}}(1 - \frac{1}{1+e^{-x}}) = S_A(1 - S_A)$
$\boxed{S'_A = S_A(1 - S_A)}$

TanSig

$S_B = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
$S'_B = \frac{(e^x+e^{-x})(e^x+e^{-x})-(e^x-e^{-x})(e^x-e^{-x})}{(e^x+e^{-x})^2}$
$= \frac{(e^x+e^{-x})^2-(e^x-e^{-x})^2}{(e^x+e^{-x})^2}$
$= 1 - S_B^2$
$\boxed{S'_B = 1 - S_B^2}$

Google's Swish

$S_C = \frac{x}{e^x+e^{-x}}$
$S'_C = \frac{(1+e^{-x}+xe^{-x}}{(1+e^{-x})^2}$
$= \frac{(1+e^{-x}+xe^{-x}+x-x}{(1+e^{-x})^2}$
$= \frac{x}{1+e^{-x}} + \frac{1}{1+e^{-x}} - \frac{x}{(1+e^{-x})^2}$
$= \frac{x}{1+e^{-x}}(1 - \frac{1}{1+e^{-x}}) + \frac{1}{1+e^{-x}}$
$= S_C(1 - S_A) + S_A$
$\boxed{S'_C = S_C(1 - S_A) + S_A}$

Mish[1]

$S_D = x * tanh(ln(1 + e^x))$
$S'_D = tanh(ln(1 + e^x)) + xsech^2 ln(1 + e^x) \frac{e^x}{1+e^x}$
$= \frac{S_D}{x} + sech^2 ln(1 + e^x) * S_C$
$N(x) = sech^2 ln(1 + e^x)$
$\boxed{S'_D = \frac{S_D}{x} + N(x) * S_C}$

[1] D. Misra, Mish: A self regularized non-monotonic neural activation function, arXiv preprint arXiv:1908.08681 (2019).

5 Problem - 05

In the figures below, we observe how the outputs a_1, a_2, a change for different values of a pattern p. In more detail, due to the nature of the activation function LogSig the output ranges in $(-1, 1)$ in comparison with the Swish activation function which has no upper bound. Hence, the values of Swish are higher. In addition, the slope of Swish is more steep than the LogSig one, that can be observed for the negative p values.

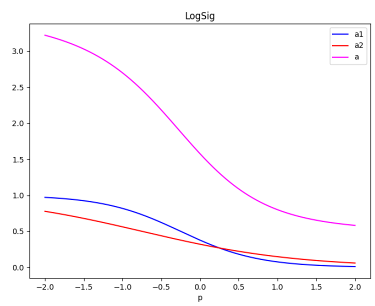


Figure 18: Logsig

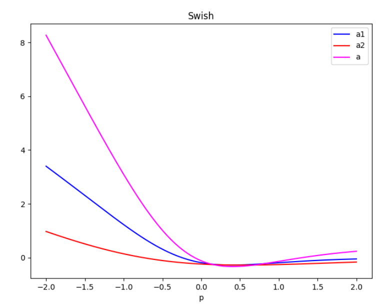


Figure 19: Swish

6 Problem - 06

When the batch size is 1 ($n_b = 1$), it represents the stochastic gradient descent, where each iteration considers only one sample to update the parameters. This randomness introduces noise, but it can help escape local minima and explore the search space better. In contrast, when the batch size is equal to the total number of samples ($n_b = n$), it's standard gradient descent, which computes the gradient on the entire dataset before updating the parameters. While this method provides a more accurate estimate of the gradient, it might converge slower and potentially get stuck in local minima.
Increasing the batch size from 1 to $n/10$ enhances convergence because it strikes a balance between the stochastic nature of SGD and the accuracy of GD. A smaller batch size introduces noise that can aid in escaping local minima, while a larger size provides a more stable estimate of the gradient. The observed improvement in convergence with $n/10$ as the batch size can be attributed to reduced noise compared to $n_b = 1$ and a less computation-intensive process

compared to $n_b = n$. This intermediate batch size allows for faster convergence by leveraging some advantages of both extremes.
As the batch size further increases from $n/10$ to n, convergence slows down. This occurs because a larger batch size leads to a more accurate but less frequent update of the model parameters. It also reduces the stochasticity, potentially causing the optimization process to get stuck in regions of the parameter space. Moreover, larger batch sizes demand more computational resources and memory, making each iteration more computationally expensive, hence slowing down the convergence rate.

7 Problem - 07

Sigmoid

From the graphic representation of the sigmoid we notice that the function approaches the value 1 for an input value close to 5 and correspondingly the value 0 for an input value close to -5. So our initial thought is to set L=5. so as to create these two intervals (-∞ ,-5) and (5,∞) in which the function converges. If we proceed to assign the value of k=0, then our function also converges to 1 for a value above 10 and converges to 0 for a value below -5. So we are left with the investigation of the interval between (-5,5). In this interval, a characteristic point is that of the intersection of the vertical axis at point 0.5. Having set L=5 and k=0 the function gives us output with 0.5 for input 0 for value m=2.

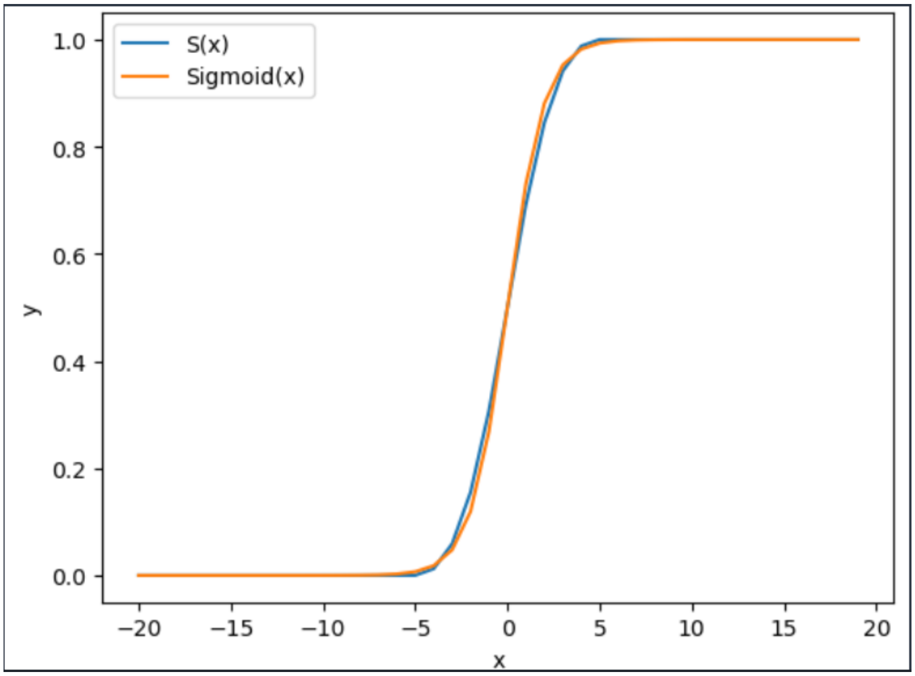


Figure : S(X) vs Sigmoid

Swish

From the graphical representation of swish we notice that the function approaches the form y=x for an input value greater than 5 and correspondingly the value 0 for an input value smaller than -5. So our initial thought is to set L=5. so as to create these two intervals (-∞,-5) and (5,∞) in which the function converges. If we proceed to assign the value of k=1, then our function approaches the form y=x for values above 5 and converges to 0 for values below -5. So we are left with the investigation of the interval between (-5,5). We notice that if we leave the value m=2 as in the sigmoid example, your function approaches the swish to a very satisfactory degree.

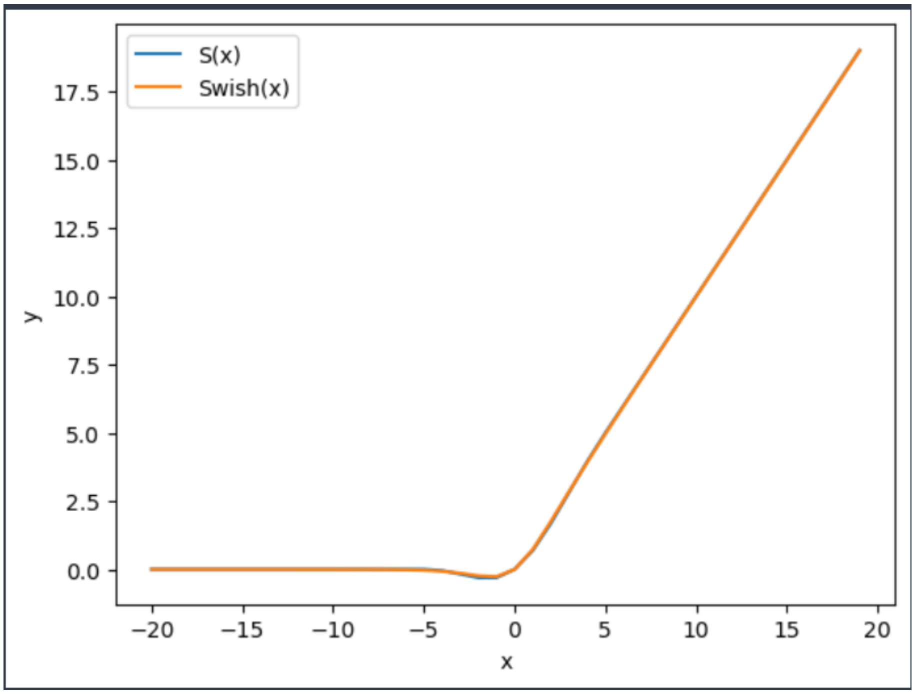


Figure : S(X) vs Swish

ReLU

From the graphical representation of ReLU we notice that the function approaches the form y=x for an input value greater than 0. For an input value smaller than 0 the function has an output of 0. So our initial thought is to set L=0. so as to create these two intervals (-∞,0) and (0,∞). If we proceed to assign the value of k=1 then our function approaches the form y=x for values above 0 and equals 0 for a value below 0. Finally to set a value for m=0 then the output equals with 0 to enter with 0.

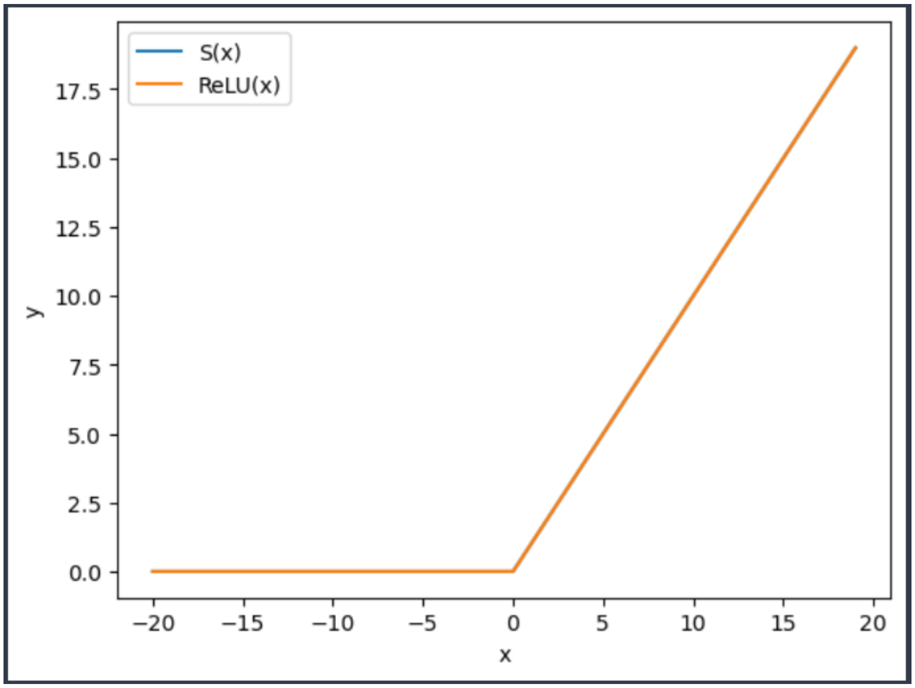


Figure : S(X) vs ReLU

B. For the given function $S(x)$ we have to compute the derivatives in terms of x, k, L, m

$$S(x) = \begin{cases} x^k & , x > L \\ x^k * \frac{(L+x)^m}{(L+x)^m+(L-x)^m} & , |x| \leq L \\ 0 & , x < -L \end{cases}$$

The derivative of $S(x)$ in terms of x

$$\frac{dS(x)}{dx} = \begin{cases} k * x^{k-1} & , x > L \\ k * x^{k-1} * \frac{(L+x)^m}{(L+x)^m+(L-x)^m} + \frac{m*x^k*(L+x)^{m-1}}{(L+x)^m+(L-x)^m} - \frac{x^k*(L+x)^m*(m(L+x)^{m-1}-m(L-x)^{m-1})}{((L+x)^m+(L-x)^m)^2} & , |x| \leq L \\ 0 & , x < -L \end{cases}$$

The derivative of $S(x)$ in terms of k

$$\frac{dS(x)}{dk} = \begin{cases} x^k * log(k) & , x > L \\ x^k * log(k) * \frac{(L+x)^m}{(L+x)^m+(L-x)^m} & , |x| \leq L \\ 0 & , x < -L \end{cases}$$

The derivative of $S(x)$ in terms of L

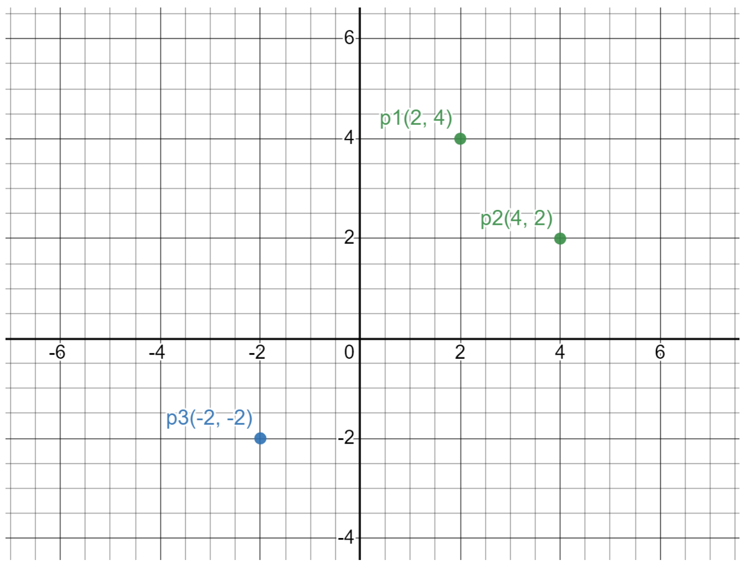
$$\frac{dS(x)}{dk} = \begin{cases} 0 & , x > L \\ \frac{2m*x^{k+1}*(L+x)^{m-1}*(L-x)^{m-1}}{((L+x)^m+(L-x)^m)} & , |x| \leq L \\ 0 & , x < -L \end{cases}$$

The derivative of $S(x)$ in terms of m

$$\frac{dS(x)}{dk} = \begin{cases} 0 & , x > L \\ \frac{x^k*(L-x)^m(L+x)^m(log(L+x)-log(L-x)))}{((L+x)^m+(L-x)^m)^2} & , |x| \leq L \\ 0 & , x < -L \end{cases}$$

8 Problem - 08

A.

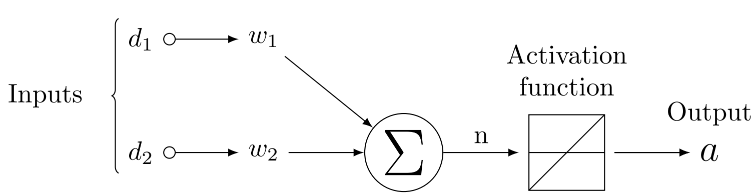


- class 1 : t1 = $\begin{bmatrix} 26 \end{bmatrix}$
- class 2 : t2 = $\begin{bmatrix} -26 \end{bmatrix}$

Patterns p1, p2 belong to class 1 and pattern p3 belong to class 2.
These patterns are linearly seperable so one line can split the patterns into 2 classes.
Hence, the network diagram ADALINE will consist of:

- 2 inputs since patterns are 2-D
- 1 neuron which represent the line that will be placed correctly using the optimal weights

ADALINE network diagram



B. To scetch the contour plot of the ADALINE's Mean Squared Error we need to find the terms of quadratic form:

$$F(x) = c - 2 \cdot x^T \cdot h + x^T \cdot R \cdot x \tag{1}$$

- h : cross-correlation vector between input and target
- R : input correlation matrix
- x = $\begin{bmatrix} w_{1,1} \\ w_{1,2} \end{bmatrix}$
- c: constant

Calculation of c

$$\begin{aligned} \mathrm{c} &= E[t^2] \\ &= p(t_1) \cdot (t_1)^2 + p(t_2) \cdot (t_2)^2 + p(t_3) \cdot (t_3)^2 \\ &= 0.2 \cdot (26)^2 + 0.7 \cdot (26)^2 + 0.1 \cdot (-26)^2 \\ \\ \mathrm{c} &= 676 \end{aligned}$$

Calculation of h

$$\begin{aligned} \mathrm{h} &= E[tz] \\ &= p(t_1) \cdot t_1 \cdot p_1 + p(t_2) \cdot t_2 \cdot p_2 + p(t_3) \cdot t_3 \cdot p_3 \\ &= 0.2 \cdot 26 \cdot \begin{bmatrix} 2 \\ 4 \end{bmatrix} + 0.7 \cdot 26 \cdot \begin{bmatrix} 4 \\ 2 \end{bmatrix} + 0.1 \cdot (-26) \cdot \begin{bmatrix} -2 \\ -2 \end{bmatrix} \\ \mathrm{h} &= \begin{bmatrix} 88.4 \\ 62.4 \end{bmatrix} \end{aligned}$$

Calculation of R

$$\begin{aligned} \mathrm{R} &= E[zz^T] \\ &= p(p_1) \cdot p_1 \cdot p_1^T + p(p_2) \cdot p_2 \cdot p_2^T + p(p_3) \cdot p_3 \cdot p_3^T \\ &= 0.2 \cdot \begin{bmatrix} 2 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 2 & 4 \end{bmatrix} + 0.7 \cdot \begin{bmatrix} 4 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 4 & 2 \end{bmatrix} + 0.1 \cdot \begin{bmatrix} -2 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} -2 & -2 \end{bmatrix} \\ \mathrm{R} &= \begin{bmatrix} 12.4 & 7.6 \\ 7.6 & 6.4 \end{bmatrix} \end{aligned}$$

Replace the terms c, h, R in $F(x)$ and plot the contour

$$
\begin{aligned}
F(x) &= c - 2 \cdot x^T \cdot h + x^T \cdot R \cdot x \\
&= 676 - 2 \cdot \begin{bmatrix} w_{1,1} & w_{1,2} \end{bmatrix} \cdot \begin{bmatrix} 88.4 \\ 62.4 \end{bmatrix} + \begin{bmatrix} w_{1,1} & w_{1,2} \end{bmatrix} \cdot \begin{bmatrix} 12.4 & 7.6 \\ 7.6 & 6.4 \end{bmatrix} \cdot \begin{bmatrix} w_{1,1} \\ w_{1,2} \end{bmatrix}
\end{aligned}
$$

$$
F(x) = 676 - 176.8 \cdot w_{1,1} - 124.8 \cdot w_{1,2} + 12.4 \cdot w_{1,1}^2 + 6.4 \cdot w_{1,2}^2 + 15.2 \cdot w_{1,1} \cdot w_{1,2}
$$

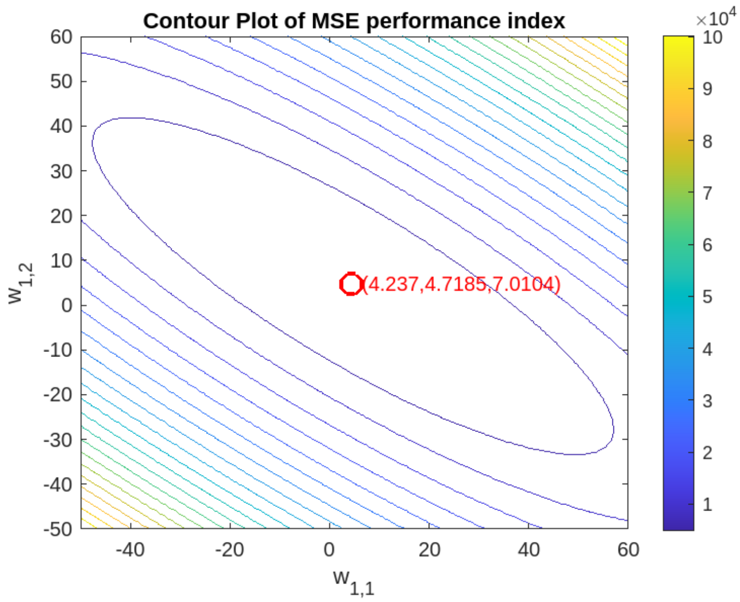


Figure 20: Contour plot of MSE

As we observe from the contour plot Figure 20, F contours consists of ellipses.
This means that F has one unique global minimum and Hessian matrix A is positive definite.
Hence, eigenvalues of A must be positive:

Eigenvalues of $A = 2R$

$$
\begin{aligned}
A &= 2R \\
A &= \begin{bmatrix} 24.8 & 15.2 \\ 15.2 & 12.8 \end{bmatrix}
\end{aligned}
$$

$$A \cdot v = \lambda \cdot v$$
$$(A - \lambda \cdot I) \cdot v = 0$$

$$\det(A - \lambda \cdot I) = 0$$
$$\det \begin{bmatrix} 24.8 - \lambda & 15.2 \\ 15.2 & 12.8 - \lambda \end{bmatrix} = 0$$
$$\lambda^2 - 27.6 \cdot \lambda + 86.4 = 0$$

$$\lambda_1 = 2.45$$
$$\lambda_2 = 35.14$$

C.

The optimal decision boundary is found when the weights $w_{1,1}$ and $w_{1,2}$ minimize the MSE performance index.
So we need to solve:

$$\dot{x} = R^{-1} \cdot h$$
$$\dot{x} = \begin{bmatrix} 12.4 & 7.6 \\ 7.6 & 6.4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 88.4 \\ 62.4 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} 4.24 \\ 4.72 \end{bmatrix}$$

The decision boundary is determined by the net input that satisfy:

$$n = w_{1,1} \cdot d_1 + w_{1,2} \cdot d_2 = 0$$
$$n = 4.24 \cdot d_1 + 4.72 \cdot d_2 = 0$$

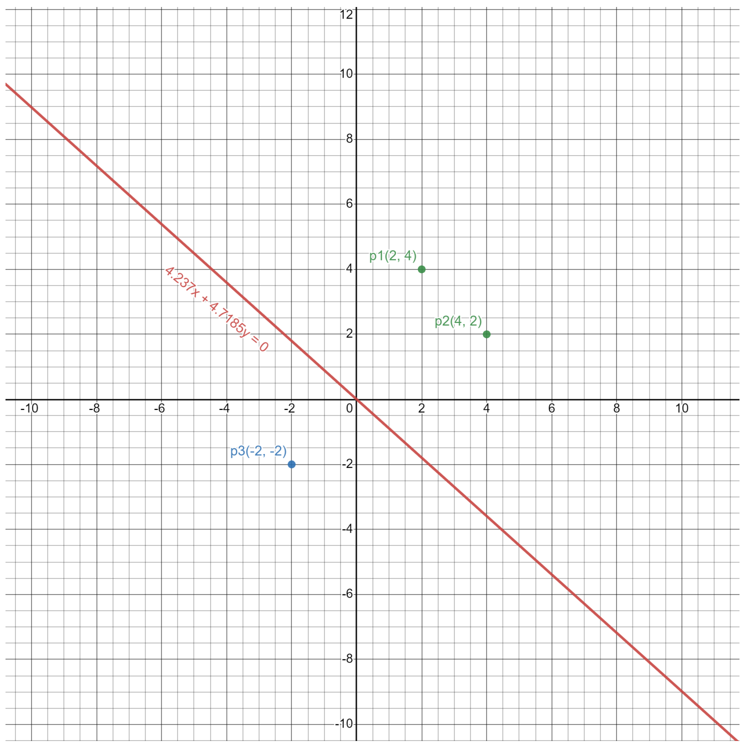


Figure 21: Line boundary

The line from the Figure 21 splits the patterns into 2 classes successfully and it can be shown analytically:

Inputs	Outputs
p_1	27.35
p_2	26.38
p_3	-17.91

D.

Maximum stable learning rate must satisfy the condition:

$$0 < \alpha < \frac{1}{\lambda_{\max}}$$

where $\lambda_{\max}$ is the maximum eigenvalue of R. Eigenvalues of $R = 0.5 \cdot A$ are:

$$\lambda_1^{'} = \frac{\lambda_1}{2} => \lambda_1^{'} = 1.23$$
$$\lambda_2^{'} = \frac{\lambda_2}{2} => \lambda_2^{'} = 17.57$$

$$\alpha_{max} = \frac{1}{\lambda_2^{'}}$$

$$\alpha_{max} = 0.057$$

Changing target values won't affect α_{max} because it is determined from the eigenvalues of $R = E[zz^T]$, where z are the input vectors

E.

One iteration of LMS algorithm

$$\begin{aligned} n(0) &= w(0) \cdot p_1 \\ &= \begin{bmatrix} w_{1,1} & w_{1,2} \end{bmatrix} \cdot \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \end{bmatrix} \end{aligned}$$

$$n(0) = 0$$

$$a(0) = purelin[n(0)]$$
$$a(0) = 0$$

$$e(0) = t_1 - a(0) = 26 - 0$$
$$e(0) = 26$$

Delta rule

$$w(1) = w(0) + 2 \cdot a(0) \cdot e(0) \cdot p_1^T$$

$$w(1) = \begin{bmatrix} 5.2 & 10.4 \end{bmatrix}$$

9 Problem - 09

A.

To scetch the contour plot of the ADALINE's Mean Squared Error we need to find the terms of quadratic form:

$$F(x) = c - 2 \cdot x^T \cdot h + x^T \cdot R \cdot x \tag{2}$$

Calculation of c

$$\begin{aligned} \text{c} &= E[t^2] \\ &= p(t_1) \cdot (t_1)^2 + p(t_2) \cdot (t_2)^2 \\ &= 0.5 \cdot (-1)^2 + 0.5 \cdot (1)^2 \\ \text{c} &= 1 \end{aligned}$$

Calculation of h

$$\begin{aligned} \text{h} &= E[tz] \\ &= p(t_1) \cdot t_1 \cdot p_1 + p(t_2) \cdot t_2 \cdot p_2 \\ &= 0.5 \cdot (-1) \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 0.5 \cdot 1 \cdot \begin{bmatrix} -2 \\ 1 \end{bmatrix} \\ \text{h} &= \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix} \end{aligned}$$

Calculation of R

$$\begin{aligned} \text{R} &= E[zz^T] \\ &= p(p_1) \cdot p_1 \cdot p_1^T + p(p_2) \cdot p_2 \cdot p_2^T \\ &= 0.5 \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \end{bmatrix} + 0.5 \cdot \begin{bmatrix} -2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -2 & 1 \end{bmatrix} \\ \text{R} &= \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \end{bmatrix} \end{aligned}$$

Replace the terms c, h, R in $F(x)$ and plot the contour

$$F(x) = c - 2 \cdot x^T \cdot h + x^T \cdot R \cdot x$$

$$= 1 - 2 \cdot \begin{bmatrix} w_{1,1} & w_{1,2} \end{bmatrix} \cdot \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix} + \begin{bmatrix} w_{1,1} & w_{1,2} \end{bmatrix} \cdot \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \end{bmatrix} \cdot \begin{bmatrix} w_{1,1} \\ w_{1,2} \end{bmatrix}$$

$$F(x) = 1 + 3 \cdot w_{1,1} + w_{1,2} + 2.5 \cdot w_{1,1}^2 + 2.5 \cdot w_{1,2}^2$$

The following graph shows the contour plot of the mean square error.

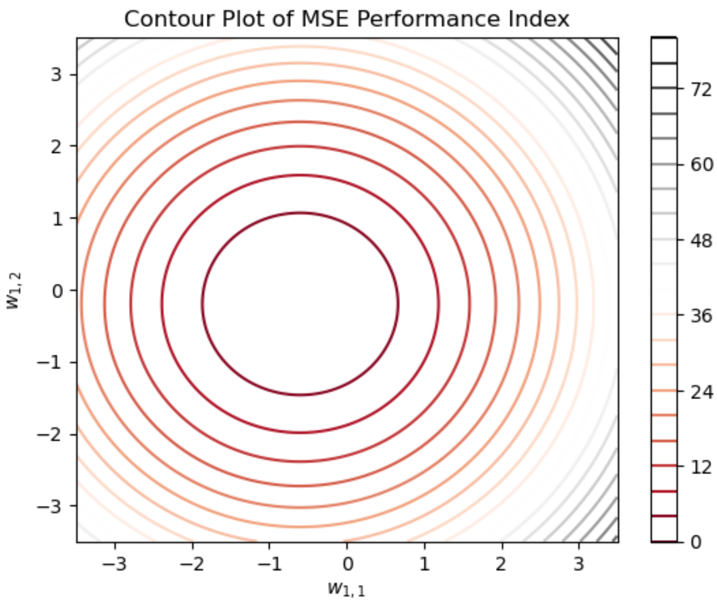


Figure 22: Contour plot

B.

Optimal decision boundary

The optimal decision boundary is found when the weights $w_{1,1}$ and $w_{1,2}$ minimize the MSE performance index

$$\dot{x} = R^{-1} \cdot h$$

$$\dot{x} = \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \end{bmatrix}^{-1} \cdot \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} -0.6 \\ -0.2 \end{bmatrix}$$

The decision boundary is determined by the net input that satisfy:

$$n = w_{1,1} \cdot d_1 + w_{1,2} \cdot d_2 = 0$$
$$n = -0.6 \cdot d_1 - 0.2 \cdot d_2 = 0$$

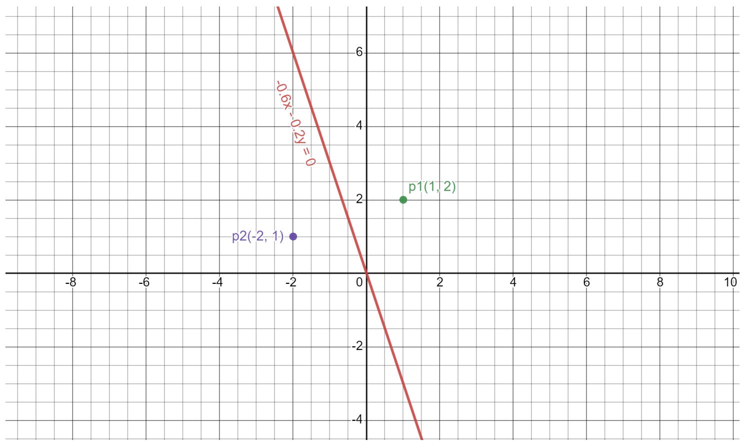


Figure 23: Line boundary

The line from the Figure 23 splits the patterns into 2 classes successfully and it can be shown analytically:

Inputs	Outputs
p_1	-1
p_2	1

C.

Trajectory of the LMS algorithm

After applying the LMS algorithm in $F(x)$, the trajectory from the beginning till the convergence of the algorithm is shown in the following figure

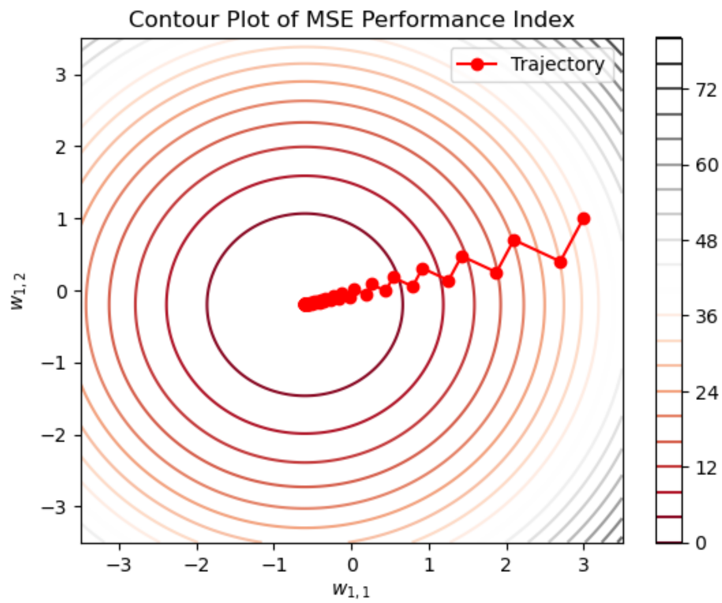


Figure 24: Trajectory of LMS

10 Problem - 10

A. In the following figure the patters are drawn in a 2-D diagram and the line $y = x - 2$ is additionally drawn to show that **the patterns are linearly separable**.

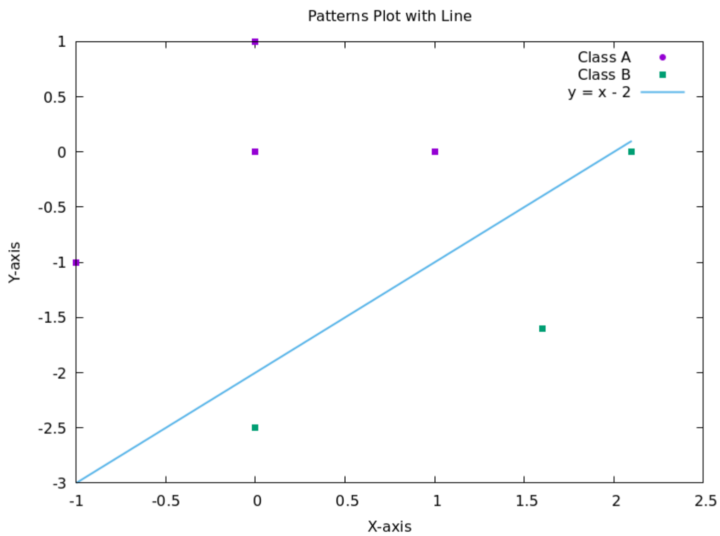
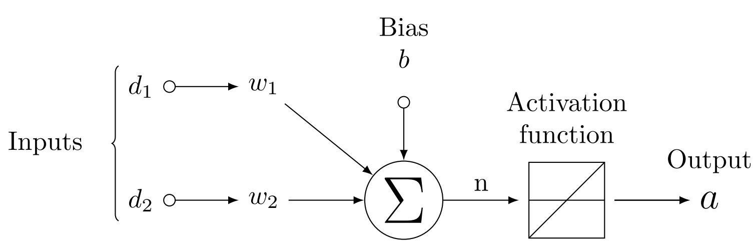


Figure 25: Patterns with y = x - 2 line

B.
The following picture shows the architecture of the adaline neural network. In our case, **only one neuron** is needed to separate the classes. In addition, we have the dimension of the inputs noted as *d1,d2*.



C.
For the first pattern we use target output $t_1 = 1$ and for the second one we use $t_2 = -1$. The first two steps of iterations will look like this:

$$\begin{cases} a(0) = W(0) * p(0) + b(0) \\ e(0) = t(0) - a(0) \\ W(1) = W(0) + 2 * lrate * e(0) * p^T(0) \\ b(1) = b(0) + 2 * a(0) * e(0) \end{cases} \tag{3}$$

and the second iteration

$$\begin{cases} a(1) = W(1) * p(1) + b(1) \\ e(1) = t(1) - a(1) \\ W(2) = W(1) + 2 * lrate * e(1) * p^T(1) \\ b(2) = b(1) + 2 * a(1) * e(1) \end{cases} \tag{4}$$

When the weight converges the weight vector is **W** = [-0.66, 0.61] and the **bias** = 0.83.

11 Problem - 11

In fuzzy set theory, a fuzzy set A is considered a fuzzy subset of another fuzzy set B if and only if the membership value of every element x in the universe set X is less than or equal to the membership value of x in B. This is denoted by the expression $A \subseteq B$.
More formally, $A \subseteq B$ if and only if $\mu_a \leq \mu_b \; \forall x \in X$.

A.
Let "S" be a fuzzy set. Then "Very S" is a fuzzy subset of "S".

A ("S"): μ_a = x

B ("Very S"): $\mu_b = x^2$

$\mu_b \leq \mu_a \; \forall x \in$ [0,1], so B is a subset of A and the sentence is **True.**

B.
Let "S" be a fuzzy set. Then "S" is a fuzzy subset of "more or less S."

A ("S"): μ_A = x

B ("Very S"): $\mu_b = \sqrt{x}$
$\mu_b \geq \mu_A \; \forall x \in$ [0,1] ,so A is a subset of B and the sentence is **True**

C.
A ("Not Very S"): μ_a = 1-x^2

B ("more or less"): μ_b =$\sqrt{x}$

For $\forall x \in$ [0,1] ,we observe the function f(x)=1-x^2 -$\sqrt{x}$.

For $\forall x \in$ [0,0.549] f(x)$\geq$ 0. For $\forall x \in$ [0.549,1] f(x) $\leq$ 0. **So we cant determine.**

D.
A ("Not more or less"): μ_A = 1-$\sqrt{x}$
B ("Very S"): μ_b =x^2
For $\forall x \in$ [0,1] ,we observe the function f(x)=1-x^2 -$\sqrt{x}$.
For $\forall x \in$ [0,0.549] f(x)$\geq$ 0. For $\forall x \in$ [0.549,1] f(x) $\leq$ 0. **So we cant determine.**

12 Problem - 12

$$A(x) = \begin{cases} 1 & ,x \leq 2 \\ 1 - \frac{x-2}{3} & ,2 < x < 5 \\ 0 & ,x \geq 5 \end{cases}$$

$$B(x) = \begin{cases} 0 & ,x \leq 3 \\ \frac{x-3}{4} & ,3 < x < 7 \\ 1 & ,x \geq 7 \end{cases}$$

Let F be a function such that $F \equiv (A(x)\ OR\ B(x)) = max(A(x), B(x))$

- $x \leq 2$

  $F = max(1, 0) = 1$

- $2 < x \leq 3$

  $F = max(1 - \frac{x-2}{3}, 0) = 1 - \frac{x-2}{3}$

- $3 < x < 5$

  $\frac{x-3}{4} < 1 - \frac{x-2}{3} \Rightarrow 7x < 29 \Rightarrow x < 4.14$

- $3 < x \leq 4.14$

  $F = max(\frac{x-3}{4}, 1 - \frac{x-2}{3}) = 1 - \frac{x-2}{3}$

- $4.14 < x < 5$

  $F = max(\frac{x-3}{4}, 1 - \frac{x-2}{3}) = \frac{x-3}{4}$

- $5 \leq x < 7$

  $F = max(\frac{x-3}{4}, 0)$

  $\frac{x-3}{4} > 0 \Rightarrow x > 3$, so $F = \frac{x-3}{4}$

- $x \geq 7$
  $F = max(1, 0) = 1$

Let H be a function such that $H \equiv \neg F = \neg(A(x) \cup B(x))$.

$$H = \begin{cases} 0 & , x \leq 2 \\ \frac{x-2}{3} & , 2 < x \leq 4.14 \\ 1 - \frac{x-3}{4} & , 4.14 < x < 7 \\ 0 & , x \geq 7 \end{cases}$$

Having the function now we can imply that the maximum value of x is **x = 4.14** and that happens at **H(4.14) = 0.713** as we can see at the graph below.

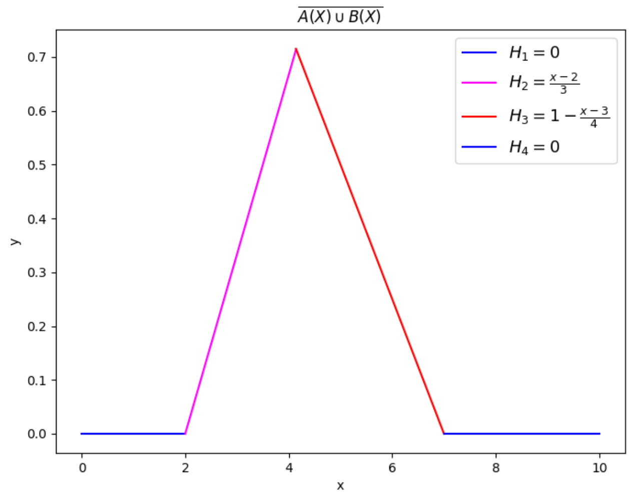


Figure 26: Plot of Multi-branch function H