Hw2

Konstantios Konstantinidis 2546
Nikolaos Stavrinos 2631

December 2020

1 .

In order to find the minimum of the function $f(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}, \qquad \mathbf{x} \in \mathbf{R}^n$. we need to find what value sets its derivative to zero,that's where we will use the Conjugate Gradient (Fletcher-Reeves)

$\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} = 0$.

$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$

For x1=1.5 and x2=-0.75=>

$$\mathbf{r}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1.5 \\ -0.75 \end{bmatrix} = \begin{bmatrix} -3.75 \\ 0 \end{bmatrix} = \mathbf{p}_0.$$

$$\alpha_0 = \frac{\mathbf{r}_0^\mathsf{T}\mathbf{r}_0}{\mathbf{p}_0^\mathsf{T}\mathbf{A}\mathbf{p}_0} = \frac{\begin{bmatrix} -3.75 & 0 \end{bmatrix} \begin{bmatrix} -3.75 \\ 0 \end{bmatrix}}{\begin{bmatrix} -3.75 & 0 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -3.75 \\ 0 \end{bmatrix}} = \frac{1}{3}.$$

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0\mathbf{p}_0 = \begin{bmatrix} 1.5 \\ -0.75 \end{bmatrix} + \frac{1}{3} \begin{bmatrix} -3.75 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.25 \\ -0.75 \end{bmatrix}.$$

Second iteration:

$$\mathbf{r}_1 = \mathbf{r}_0 - \alpha_0\mathbf{A}\mathbf{p}_0 = \begin{bmatrix} -3.75 \\ 0 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -3.75 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}.$$

$$\beta_0 = \frac{\mathbf{r}_1^\mathsf{T}\mathbf{r}_1}{\mathbf{r}_0^\mathsf{T}\mathbf{r}_0} = \frac{\begin{bmatrix} 0 & 1.25 \end{bmatrix} \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}}{\begin{bmatrix} -3.75 & 0 \end{bmatrix} \begin{bmatrix} -3.75 \\ 0 \end{bmatrix}} = \frac{1}{9}.$$

$$\mathbf{p}_1 = \mathbf{r}_1 + \beta_0\mathbf{p}_0 = \begin{bmatrix} 0 \\ 1.25 \end{bmatrix} + \frac{1}{9} \begin{bmatrix} -3.75 \\ 0 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} -1.25 \\ 3.75 \end{bmatrix}.$$

$$\alpha_1 = \frac{\mathbf{r}_1^\mathsf{T}\mathbf{r}_1}{\mathbf{p}_1^\mathsf{T}\mathbf{A}\mathbf{p}_1} = 9 * \frac{\begin{bmatrix} 0 & 1.25 \end{bmatrix} \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}}{\begin{bmatrix} -1.25 & 3.75 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -1.25 \\ 3.75 \end{bmatrix}} = 0.6.$$

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha_1\mathbf{p}_1 = \begin{bmatrix} 0.25 \\ -0.75 \end{bmatrix} + 0.2 \begin{bmatrix} -1.25 \\ 3.75 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

In order to find the minimum of the function $f(\mathbf{x}) = 3w_1^2 + 2w_2^2 + 2w_1w_2$with gradient descent
$\nabla f(\mathbf{x}) = \left[\begin{array}{c} 6w_1 + 2w_2 \\ 2w_1 + 4w_2 \end{array} \right] = 0\,.$
$(w_1(1), w_2(1)) = (w_1(0), w_2(0)) - t_0\nabla f((w_1(0), w_2(0))$
To find t0 we need to find the minimum of the function $\theta(t) = f(w_1(0), w_2(0)) - t_0\nabla f((w_1(0), w_2(0))$
$\theta'(t) = -\nabla f((w_1(0), w_2(0)) - t\nabla f(w_1(0), w_2(0)))^T \nabla f(w_1(0), w_2(0)) = f(w_1(0), w_2(0)) - t_0\nabla f(1.5 - 0.75t, -0.75 + 3t)^T * \nabla f(7.5, -3) = -38.25 + 283.5t$
That has to equal to zero,so t=0.135
So $(w_1(1), w_2(1)) = (1.5, -0.75) - 0.135 * (7.5, -3) = (1.5 - 1.012, -0.75 + 0.405) = (0.488, -0.345)$
Second iteration:
$\theta'(t) == f(w_1(1), w_2(1)) - t_0\nabla f(1.5 - 0.75t, -0.75 + 3t)^T * \nabla f(7.5, -3) = -3.762 + 13.98 * t$
That has to equal to zero,so t=0.269
So $(w_1(2), w_2(2)) = (0.488, -0.345) - 0.269 * (1.608, -0.404) = (0.055, -0.236)$
which will also converge to (0,0) as well.

2 .

Problem 2

$$F(w) = 100(w_2 - w_1^2)^2 + (1 - w_1)^2$$

$x_0 = [-0{,}5 \quad 0{,}5]^T$

Newton's method and fixed step $\lambda_k = 1$

$$F(w) = 100w_1^4 + w_1^2 - 200w_2w_1^2 - 2w_1 + 100w_2^2 + 1$$

$$\nabla F(w) = \begin{pmatrix} 400w_1^3 + 2w_1 - 400w_2w_1 - 2 \\ -200w_1^2 + 200w_2 \end{pmatrix}$$

$$H = \begin{bmatrix} 1200w_1^2 - 400w_2 + 2 & -400w_1 \\ -400w_1 & +200 \end{bmatrix}$$

At $x_0 = [-0{,}5 , 0{,}5]^T$, we get: $\nabla F(x_0) = [47 , 50]^T$ and

$$H = \begin{bmatrix} 102 & 200 \\ 200 & 200 \end{bmatrix}$$

$$H^{-1} = \begin{bmatrix} -0{,}01 & 0{,}01 \\ 0{,}01 & -0{,}005 \end{bmatrix}$$

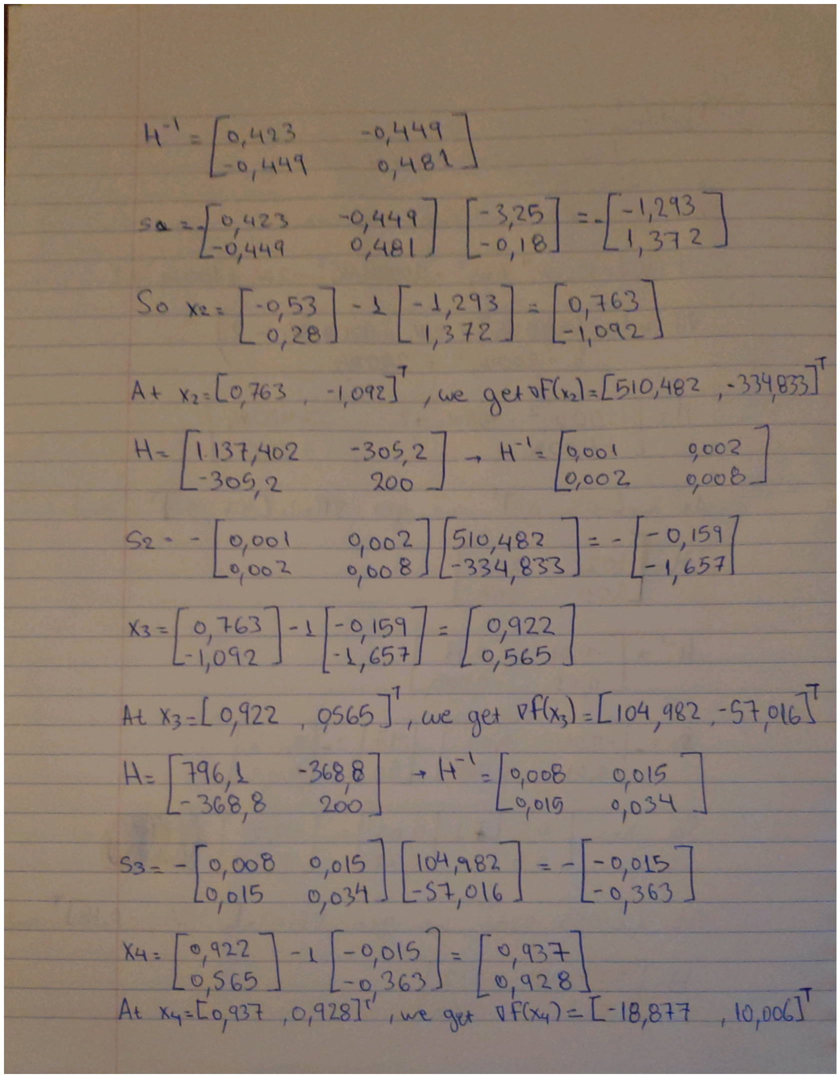
$$S_0 = -\begin{bmatrix} -0{,}01 & 0{,}01 \\ 0{,}01 & -0{,}005 \end{bmatrix} \begin{bmatrix} 47 \\ 50 \end{bmatrix} = -\begin{bmatrix} 0{,}03 \\ 0{,}22 \end{bmatrix}$$

So $x_1 = \begin{bmatrix} -0{,}5 \\ 0{,}5 \end{bmatrix} + 1 \begin{bmatrix} 0{,}03 \\ 0{,}22 \end{bmatrix} = \begin{bmatrix} -0{,}53 \\ 0{,}28 \end{bmatrix}$

At $x_1 = [-0{,}53 , 0{,}28]^T$, we get $\nabla f(x_1) = [-3{,}25 , -0{,}18]^T$ and

$$H = \begin{bmatrix} 227{,}08 & 212 \\ 212 & 200 \end{bmatrix}$$

Figure 1:



$$H^{-1} = \begin{bmatrix} 0{,}423 & -0{,}449 \\ -0{,}449 & 0{,}481 \end{bmatrix}$$

$$s_2 = -\begin{bmatrix} 0{,}423 & -0{,}449 \\ -0{,}449 & 0{,}481 \end{bmatrix} \begin{bmatrix} -3{,}25 \\ -0{,}18 \end{bmatrix} = -\begin{bmatrix} -1{,}293 \\ 1{,}372 \end{bmatrix}$$

So $x_2 = \begin{bmatrix} -0{,}53 \\ 0{,}28 \end{bmatrix} - 1 \begin{bmatrix} -1{,}293 \\ 1{,}372 \end{bmatrix} = \begin{bmatrix} 0{,}763 \\ -1{,}092 \end{bmatrix}$

At $x_2 = [0{,}763 \;, -1{,}092]^T$, we get $\nabla f(x_2) = [510{,}482 \;, -334{,}833]^T$

$$H = \begin{bmatrix} 1.137{,}402 & -305{,}2 \\ -305{,}2 & 200 \end{bmatrix} \rightarrow H^{-1} = \begin{bmatrix} 0{,}001 & 0{,}002 \\ 0{,}002 & 0{,}008 \end{bmatrix}$$

$$S_2 = -\begin{bmatrix} 0{,}001 & 0{,}002 \\ 0{,}002 & 0{,}008 \end{bmatrix} \begin{bmatrix} 510{,}482 \\ -334{,}833 \end{bmatrix} = -\begin{bmatrix} -0{,}159 \\ -1{,}657 \end{bmatrix}$$

$$x_3 = \begin{bmatrix} 0{,}763 \\ -1{,}092 \end{bmatrix} - 1 \begin{bmatrix} -0{,}159 \\ -1{,}657 \end{bmatrix} = \begin{bmatrix} 0{,}922 \\ 0{,}565 \end{bmatrix}$$

At $x_3 = [0{,}922 \;, 0565]^T$, we get $\nabla f(x_3) = [104{,}982 \;, -57{,}016]^T$

$$H = \begin{bmatrix} 796{,}1 & -368{,}8 \\ -368{,}8 & 200 \end{bmatrix} \rightarrow H^{-1} = \begin{bmatrix} 0{,}008 & 0{,}015 \\ 0{,}015 & 0{,}034 \end{bmatrix}$$

$$S_3 = -\begin{bmatrix} 0{,}008 & 0{,}015 \\ 0{,}015 & 0{,}034 \end{bmatrix} \begin{bmatrix} 104{,}982 \\ -57{,}016 \end{bmatrix} = -\begin{bmatrix} -0{,}015 \\ -0{,}363 \end{bmatrix}$$

$$x_4 = \begin{bmatrix} 0{,}922 \\ 0{,}565 \end{bmatrix} - 1 \begin{bmatrix} -0{,}015 \\ -0{,}363 \end{bmatrix} = \begin{bmatrix} 0{,}937 \\ 0{,}928 \end{bmatrix}$$

At $x_4 = [0{,}937 \;, 0{,}928]^T$, we get $\nabla f(x_4) = [-18{,}877 \;, 10{,}006]^T$

Figure 2:

$$H = \begin{bmatrix} 684{,}362 & -374{,}8 \\ -374{,}8 & 200 \end{bmatrix} \rightarrow H^{-1} = \begin{bmatrix} -0{,}055 & -0{,}104 \\ -0{,}104 & -0{,}189 \end{bmatrix}$$

$$S_4 = -\begin{bmatrix} -0{,}055 & -0{,}104 \\ -0{,}104 & -0{,}189 \end{bmatrix} \begin{bmatrix} -18{,}877 \\ 10{,}006 \end{bmatrix} = -\begin{bmatrix} -0{,}002 \\ 0{,}072 \end{bmatrix}$$

$$x_5 = \begin{bmatrix} 0{,}937 \\ 0{,}928 \end{bmatrix} - 1\begin{bmatrix} -0{,}002 \\ 0{,}072 \end{bmatrix} = \begin{bmatrix} 0{,}939 \\ 0{,}856 \end{bmatrix}$$

At $x_5 = [0{,}939 \;,\; 0{,}856]^T$, we get $\nabla f(x_5) = [9{,}538 \;,\; -5{,}144]^T$

$$H = \begin{bmatrix} 717{,}665 & -375{,}6 \\ -375{,}6 & 200 \end{bmatrix} \rightarrow H^{-1} = \begin{bmatrix} 0{,}081 & 0{,}152 \\ 0{,}152 & 0{,}292 \end{bmatrix}$$

$$S_5 = -\begin{bmatrix} 0{,}081 & 0{,}152 \\ 0{,}152 & 0{,}292 \end{bmatrix} \begin{bmatrix} 9{,}538 \\ -5{,}144 \end{bmatrix} = -\begin{bmatrix} -0{,}009 \\ -0{,}052 \end{bmatrix}$$

$$x_6 = \begin{bmatrix} 0{,}939 \\ 0{,}856 \end{bmatrix} - 1\begin{bmatrix} -0{,}009 \\ -0{,}052 \end{bmatrix} = \begin{bmatrix} 0{,}948 \\ 0{,}908 \end{bmatrix}$$

At $x_6 = [0{,}948 \;,\; 0{,}908]^T$, we get $\nabla f(x_6) = [-3{,}629, 1{,}859]^T$

As we can see we get closer to the answer, which is $x = [1 \; 1]^T \rightarrow \nabla f(x) = 0$. With some more iterations we would have gotten the correct answer.

Figure 3:

3 .

3. Feed-Forward Phase:
   Compute Error:
   a1 = (-2 * 1 + 1 * 1)3 = -1 a2 = (1 * 1 -(2 * 1))2 = 1 (2)
   Backpropagation: Sensitivities:
   error = 0 - output = 0 - 1 =- 1 (3)
   s2 = -2Ft2'(n2)(t - a) = -4(-1) (e) = -4
s1 = F'1(n1)(W 2)T s2 = 3(n1)2* 1 *(-4) = -12 (4)
   Update weights and biases:
   W 2(1) = W 2(0) - a * s2(a1)T = 1 - 1 * (-4) * 1 = -3
b2(1) = b2(0) - a * s2 = -2 - 1 * (-4) = 6
W 1(1) = W 1(0) - a * s1(a0)T = -2 - 1 * (-12) = 10
b1(1) = b1(0) - a * s1 = 1 - 1 * (-12) = 13

4 .

$H1(1) = 0.1 * 1 + 0.3 * 4 + 0.5 * 5 + 0.5 = 4.3$
$H2(1) = 0.2 * 1 + 0.4 * 4 + 0.6 * 5 + 0.5 = 5.3$

$Sigmoid(H1(1)) = 1/(1 + e^{-4.3}) = 0.986$
$Sigmoid(H2(1)) = 1/(1 + e^{-5.3}) = 0.995$

$O1 = 0.7 * 0.986 + 0.9 * 0.995 + 0.5 = 2.085$
$O2 = 0.8 * 0.986 + 0.1 * 0.995 + 0.5 = 1.338$

$Sigmoid(O1(1)) = 0.889$
$Sigmoid(O2(1)) = 0.792$

$E1 = t1 - 0.889 = 0.1 - 0.889 = -0.789$
$E2 = t2 - 0.889 = 0.05 - 0.792 = -0.749$
$Sigmoid(x) = 1/(1 + e^{-x}) => Sigmoid'(x) = \frac{d}{dx}(1/(1 + e^{-x}) = e^{-x}/((1 + e^{-x}))^2 = (1 - 1/(1 + e^{-x}) * (1/(1 + e^{-x}) = (1 - a^1) * (a^1)$

$s2 = -2F'^2(n2)(t - a) = -2 * \begin{bmatrix} a_1 * (1 - a_1) & 0 \\ 0 & a_2(1 - a_2) \end{bmatrix} \begin{bmatrix} -0.789 \\ -0.749 \end{bmatrix} = -2 * \begin{bmatrix} 0.889 * (1 - 0.889) & 0 \\ 0 & 0.792(1 - 0.792) \end{bmatrix} \begin{bmatrix} -0.789 \\ -0.75 \end{bmatrix} = \begin{bmatrix} 0.155 \\ 0.24 \end{bmatrix}$
$s1 = F'^1(n1)'(W^2)^T s2 = \begin{bmatrix} 0.013 & 0 \\ 0 & 0.05 \end{bmatrix} \begin{bmatrix} 0.7 & 0.9 \\ 0.8 & 0.1 \end{bmatrix} \begin{bmatrix} 0.155 \\ 0.24 \end{bmatrix} = \begin{bmatrix} 0.0042 \\ 0.0007 \end{bmatrix}$
$W2(1) = W2(0) - a * s2(a1)T = \begin{bmatrix} 0.7 & 0.9 \\ 0.8 & 0.1 \end{bmatrix} - 0.01 * \begin{bmatrix} 0.155 \\ 0.24 \end{bmatrix} \begin{bmatrix} 0.986 \\ 0.995 \end{bmatrix}^T \approx \begin{bmatrix} 0.698 & 0.898 \\ 0.797 & 0.017 \end{bmatrix}$

b2(1) = b2(0) - a * s2 = -2 - 1 * (-4) = 6= $\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - 0.01 \begin{bmatrix} 0.155 \\ 0.24 \end{bmatrix} = \begin{bmatrix} 0.4845 \\ 0.476 \end{bmatrix}$
W 1(1) = W 1(0) - a * s1(a0)T = $\begin{bmatrix} 0.1 & 0.3 & 0.5 \\ 0.2 & 0.4 & 0.6 \end{bmatrix} - 0.1 * \begin{bmatrix} 0.0042 \\ 0.0007 \end{bmatrix} * \begin{bmatrix} 1 & 4 & 5 \end{bmatrix} =$
$\begin{bmatrix} 0.1 & 0.2998 & 0.499 \\ 0.2 & 0.4 & 0.6 \end{bmatrix}$
b1(1) = b1(0) - a * s1 = $\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - 0.01 * \begin{bmatrix} 0.0042 \\ 0.0007 \end{bmatrix} \approx \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$,the difference is too small to be taken into account.

5 .

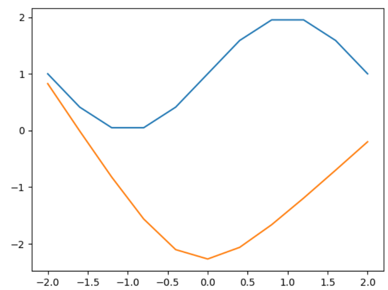
Suppose that:
$x_0 = a * in + b$
$x_1 = sigmoid(x_0)$
$x_2 = a * x_1 + b$
.
.
.
$x_n = a * x_{n-1} + b$

Knowing that S'(x)=Sigmoid'(x)=S(S-1) the derivative of xn with regards to in can be written like this: $\frac{d}{din}(x_n) = a * x_{n-1} * (x_{n-1} - 1) * \frac{d}{dx}(x_{n-2})$...
which will eventually look something like this:
$a^{n/2+1} \prod_{i=1}^{n} S_{2n-1} * (S_{2n-1} - 1)$

In terms of the vanishing gradient problem ,it's a common case with sigmoid activation functions.When the sigmoid function value is either too high or too low, the derivative becomes very small i.e. << 1. This causes vanishing gradients and poor learning for deep networks. This can occur when the weights of our networks are initialized poorly – with too-large negative and positive values. These too-large values saturate the input to the sigmoid and pushes the derivatives into the small valued regions.

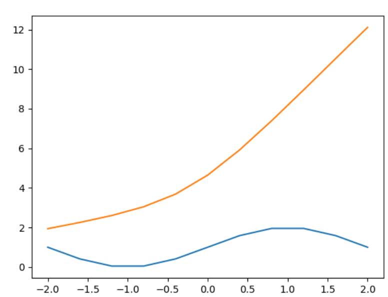
6 .

Learning rate = 0.001 and S=3
Final weight1: [0.030909503133541627, -3.1942548621059763, -3.1069772191111356]



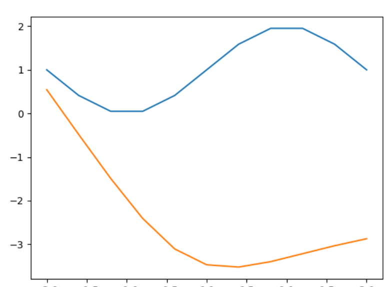
and bias1: [0.5112930381588776, -0.0038236442505651244, -0.03493906158817512]
Final weight2: [0.7868737753116977, -0.6733869741641436, -0.6539490598958959]
and bias2: 1.1681376688809941
P0 is 0.8263888898491442 and is supposed to be 0.9999999999999999
P1 is -0.009524055853615154 and is supposed to be 0.41221474770752675
P2 is -0.8216509447115238 and is supposed to be 0.04894348370484636
P3 is -1.5610607586741283 and is supposed to be 0.04894348370484647
P4 is -2.1011380032554157 and is supposed to be 0.41221474770752686
P5 is -2.2665685056520224 and is supposed to be 1.0
P6 is -2.0612501510493635 and is supposed to be 1.5877852522924731
P7 is -1.6618523455024368 and is supposed to be 1.9510565162951536
P8 is -1.1922061417108392 and is supposed to be 1.9510565162951536
P9 is -0.6996341191567705 and is supposed to be 1.5877852522924734
P10 is -0.19883030363382093 and is supposed to be 1.0000000000000002

Learning rate = 0.001 and s=15
Final weight1: [0.44248942488399756, 2.192577697977375, -2.9334139200837996,



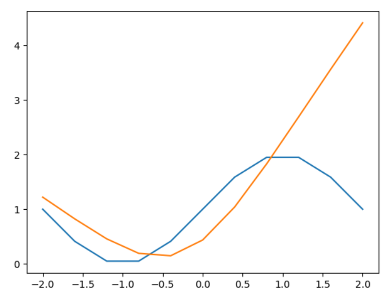
0.023921515983921348, 0.1655817134382285, 0.08042321033838204, -0.25129977565460937, -1.7874009468671817, -0.9457353201443084, -0.05928499625125257, 1.2991168623567368, -0.4632036069582728, 1.0495255718224887, 0.13163840730061563, 0.20572166694019547]
and bias1: [-0.11734813295953414, 0.11293384881877229, 0.04977567352124146, 0.18966767919722374, -0.15091223854548963, 0.4146491855413997, 0.4510993079333351, 0.01584086676386032, 0.15350363148328103, -0.208116011985671, -0.02202720588175104, 0.28334630953654366, 0.19555675069896544, 0.3271045368598385, 0.3408128065616652]
Final weight2: [-0.047223304497623776, 0.22556669924561606, -0.8016135689844859, -0.02377623811570163, -0.2646619467263139, 0.003107406186166055, 0.5277092052373649, -0.25001679839177476, 0.15481301876079062, 0.49869553562195834, 0.3767325473095455, 0.5162848333788569, 0.17864500535177347, 0.08628723979248827, 0.46686824344210454]
and bias2: 0.04223447251335259
P0 is 1.9366610083957623 and is supposed to be 0.9999999999999999
P1 is 2.251975615154788 and is supposed to be 0.41221474770752675
P2 is 2.6024578046802644 and is supposed to be 0.04894348370484636
P3 is 3.0456421915182776 and is supposed to be 0.04894348370484647
P4 is 3.686633485414877 and is supposed to be 0.41221474770752686
P5 is 4.644236623691551 and is supposed to be 1.0
P6 is 5.920127967409252 and is supposed to be 1.5877852522924731
P7 is 7.392649070245966 and is supposed to be 1.9510565162951536
P8 is 8.950781625121916 and is supposed to be 1.9510565162951536
P9 is 10.533635746706944 and is supposed to be 1.5877852522924734
P10 is 12.111946940528163 and is supposed to be 1.0000000000000002

Learning rate = 0.005 and S=3
Final weight1: [-0.8419387318381734, -2.469628232488532, -0.6625817894012812]



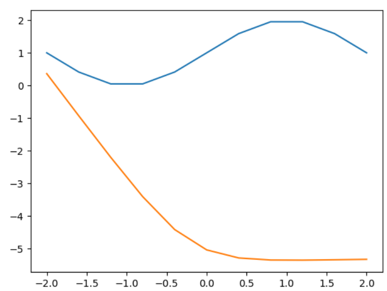
and bias1: [0.05687613055865347, -0.12860465010128191, 0.29883535110650133]
Final weight2: [0.21118944892587627, -1.6425329669736575, 0.5158409399669326]
and bias2: 1.3970208646859419
P0 is 0.5445437204361546 and is supposed to be 0.9999999999999999
P1 is -0.4828967327193102 and is supposed to be 0.41221474770752675
P2 is -1.4907611273684096 and is supposed to be 0.04894348370484636
P3 is -2.4079381121750125 and is supposed to be 0.04894348370484647

P4 is -3.1069389266087093 and is supposed to be 0.41221474770752686
P5 is -3.4706978148065857 and is supposed to be 1.0
P6 is -3.52266845128195 and is supposed to be 1.5877852522924731
P7 is -3.3989140165481033 and is supposed to be 1.9510565162951536
P8 is -3.216297011566445 and is supposed to be 1.9510565162951536
P9 is -3.033802588854829 and is supposed to be 1.5877852522924734
P10 is -2.8733159094187606 and is supposed to be 1.0000000000000002
Learning rate = 0.005 and S=15
Final weight1: [1.0483386583475927, -0.027818322420441088, 0.20730636410898232,



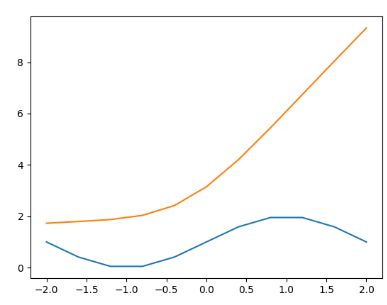
-2.641580661484995, -0.32305620552702596, 0.03227842637125662, -0.6575980949914932, -0.1550287283973066, 1.4421031210482036, -0.09104542725601783, 0.20837479838020448, -0.1008441735787519, -0.21982411464533727, -2.019067621503454, 0.001870705741622904] and bias1: [-0.3061649971615442, 0.30186652366327255, -0.12069880594199672, -0.09491564068117862, 0.10004960663062432, -0.08161690398191108, 0.22812032856039874, 0.4319525950320058, 0.2483794180191233, 0.4125995970135096, -0.23510101808461378, 0.26996947956350886, -0.31118732117359915, -0.04376261069444169, -0.25878865930926376]
Final weight2: [-0.28748440801038627, 0.45695770488490806, 0.34104022389889865, -1.025286634976018, 0.3678096786267325, -0.3604502273783705, 0.2399336097050874, 0.01190148666249783, 0.3105063161071186, -0.13555167548617963, 0.04821312624796542, 0.03129651439459619, 0.5224922001833565, -0.4965958108634263, 0.47139208311967684]
and bias2: 0.7317846751523781
P0 is 1.220706150009886 and is supposed to be 0.9999999999999999
P1 is 0.8234000579309669 and is supposed to be 0.41221474770752675
P2 is 0.4569809513973019 and is supposed to be 0.04894348370484636
P3 is 0.19149662825713823 and is supposed to be 0.04894348370484647
P4 is 0.14686202679453045 and is supposed to be 0.41221474770752686
P5 is 0.43592815152071773 and is supposed to be 1.0
P6 is 1.0405847184060888 and is supposed to be 1.5877852522924731
P7 is 1.8311735605021464 and is supposed to be 1.9510565162951536
P8 is 2.692391976954826 and is supposed to be 1.9510565162951536
P9 is 3.5625317272403487 and is supposed to be 1.5877852522924734
P10 is 4.4158928630509475 and is supposed to be 1.0000000000000002
Learning rate = 0.01 and S=3

Final weight1: [-3.2747628057401945, -0.6556448833569682, -0.6655380952383398]



and bias1: [-0.04502160721259928, 0.22332333021315504, -0.4540506384021959]
Final weight2: [-1.3340811051148649, 0.08634553188202508, -0.047309571489522126]
and bias2: 1.6291113423259438
P0 is 0.36190769225761305 and is supposed to be 0.9999999999999999
P1 is -0.9280289346499493 and is supposed to be 0.41221474770752675
P2 is -2.199637366917776 and is supposed to be 0.04894348370484636
P3 is -3.405271893522561 and is supposed to be 0.04894348370484647
P4 is -4.413662097507572 and is supposed to be 0.41221474770752686
P5 is -5.036090943009504 and is supposed to be 1.0
P6 is -5.282819361806901 and is supposed to be 1.5877852522924731
P7 is -5.345772146802508 and is supposed to be 1.9510565162951536
P8 is -5.349561788926244 and is supposed to be 1.9510565162951536
P9 is -5.3384909568011505 and is supposed to be 1.5877852522924734
P10 is -5.3253510574372855 and is supposed to be 1.0000000000000002
Learning rate = 0.01 and S=15

Final weight1: [-2.0237520231170962, -0.09893143399059656, -0.21210133740179446,



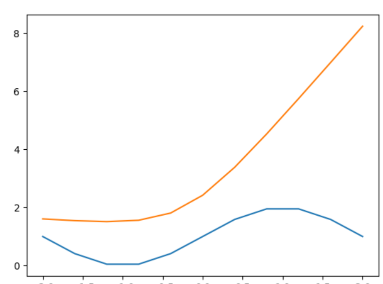
0.016536964024686667, -2.4601126708409984, -0.024688676945908556, 0.4158409485327284, 1.6603391162739862, 1.1596126677694492, 0.7278911198560799, 2.615943128110069,

-0.03841717876716053, -0.022914155563302018, 0.2468067593753753, 0.23828711324676885]
and bias1: [-0.3259279390668322, 0.41086447512186985, 0.010480276465058091,
-0.26393996984709106, -0.10419349348505465, 0.28243426993717424, -0.27078446052594146,
0.15880785853913773, 0.34905474581804186, -0.1631155521688279, 0.14511728897520065,
0.3697126022661095, 0.24599762409379913, -0.4097029510828328, 0.4311591216569938]
Final weight2: [-0.2534172608951247, 0.5990518993465422, 0.680319420729582,
-0.40295521635078824, -0.7576281623173399, 0.32792856917174046, -0.10189424164139978,
-0.029459129632529314, -0.39030433536374115, -0.16110986952858639, 0.7892000849571653,
0.322811868474445, -0.11903385661336525, 0.3255190160450032, 0.2775181251259697]
and bias2: 0.2679078651681681
P0 is 1.7296084544589791 and is supposed to be 0.9999999999999999
P1 is 1.7957283265393613 and is supposed to be 0.41221474770752675
P2 is 1.872737787166379 and is supposed to be 0.04894348370484636
P3 is 2.034560178740187 and is supposed to be 0.04894348370484647
P4 is 2.4173562964327773 and is supposed to be 0.41221474770752686
P5 is 3.1486176744669923 and is supposed to be 1.0
P6 is 4.204385448102281 and is supposed to be 1.5877852522924731
P7 is 5.441429307396668 and is supposed to be 1.9510565162951536
P8 is 6.74042640905029 and is supposed to be 1.9510565162951536
P9 is 8.04340414023412 and is supposed to be 1.5877852522924734
P10 is 9.32952197475456 and is supposed to be 1.0000000000000002

By the results from the learning rates we used,the value of the learning rate seems to be more reliable between the values of 0.1 and 0.001 because the g(p) function changes periodically and its' results are between 0 and 2.Lower learning rates help the training by changing the direction of the graph with more ease.

7 .

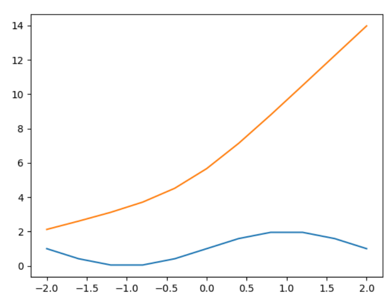
Dropout probability = 0.10
Final weight1: [-2.59910509 0.11120802 0.04913053 -0.01710935 -0.40061256 -



0.17218889 0. -0.10881888 0.05121299 -0.07120143 -0.21352516 -1.13821909 0.
2.19343773 3.28428206]

and bias1: [0.02125911 0.09966034 -0.13790664 0.25835404 -0.35902373 -0.40731497 0. 0.08744258 0.06730358 0.39121647 0.35541087 -0.21745325 0. 0.04988636 0.07497568]
Final weight2: [-0.09375145 -0.26012935 -0.33176239 0.3031241 -0.06749578 -0.07396368 -0.40910055 0.4595959 0.50765136 0. -0.06594695 0. -0.14486669 0.31212735 0.94427507]
and bias2: 0.3888552378385044
P0 is 1.6063565932441717 and is supposed to be 0.9999999999999999
P1 is 1.5465368957975516 and is supposed to be 0.41221474770752675
P2 is 1.5135645553075916 and is supposed to be 0.04894348370484636
P3 is 1.5597572126274655 and is supposed to be 0.04894348370484647
P4 is 1.8085356608153844 and is supposed to be 0.41221474770752686
P5 is 2.420116840610599 and is supposed to be 1.0
P6 is 3.381586799543923 and is supposed to be 1.5877852522924731
P7 is 4.5285037980884715 and is supposed to be 1.9510565162951536
P8 is 5.746222955758188 and is supposed to be 1.9510565162951536
P9 is 6.98740151229165 and is supposed to be 1.5877852522924734
P10 is 8.235041498629453 and is supposed to be 1.0000000000000002

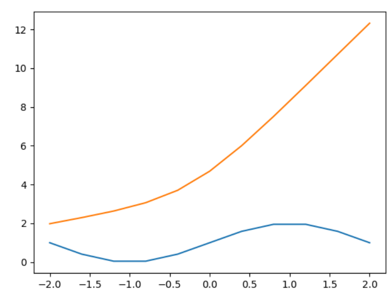
Dropout probability = 0.25
Final weight1: [8.58672372e-01 0.00000000e+00 3.39537126e-02 0.00000000e+00



3.25577600e-01 2.07978851e+00 0.00000000e+00 -1.64843112e-03 -2.23555589e+00 2.71709630e+00 1.31452509e+00 -1.19593827e+00 2.10880056e-01 -7.51945555e-02 0.00000000e+00]
and bias1: [-0.1690654 0. 0.27909064 0. -0.18602904 0.03264184 0. 0.01307758 -0.10866247 0.08776054 -0.09998297 -0.03050814 -0.15209358 0.09353 0.]
Final weight2: [-0.45340106 0.55331627 0. -0.1090509 0.34532304 0.08985712 0. 0.31457253 -0. 1.41793166 -0.08810215 0. 0.2317704 0.08018592 -0.15165098]
and bias2: -0.13774154918341364
P0 is 2.1203701555272065 and is supposed to be 0.9999999999999999
P1 is 2.606043771664603 and is supposed to be 0.41221474770752675
P2 is 3.116716283593564 and is supposed to be 0.04894348370484636
P3 is 3.7118037127905557 and is supposed to be 0.04894348370484647

P4 is 4.515802689577027 and is supposed to be 0.41221474770752686
P5 is 5.662412330515885 and is supposed to be 1.0
P6 is 7.134933289133754 and is supposed to be 1.5877852522924731
P7 is 8.788521787398686 and is supposed to be 1.9510565162951536
P8 is 10.507708563296589 and is supposed to be 1.9510565162951536
P9 is 12.241272140607023 and is supposed to be 1.5877852522924734
P10 is 13.973416628346884 and is supposed to be 1.0000000000000002

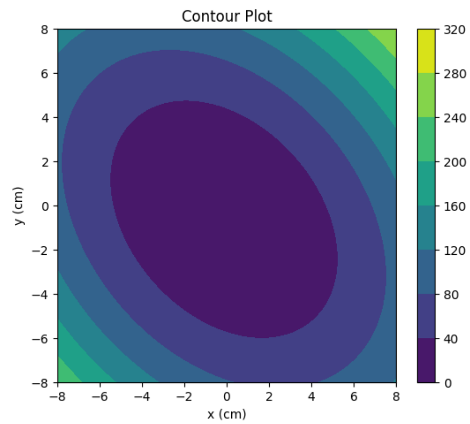
Dropout probability = 0.50
Final weight1: [1.89135572 0. 0. 0. 0. 2.37380589 0. -1.61275657 0. 0.25219742



-0.35666342 -3.05603642 0. 2.71251341 0.]
and bias1: [-0.09784551 0. 0. 0. 0. 0.13734658 0. -0.10566741 0. 0.11077245 0.04867221 0.01692556 0. 0.05744437 0.]
Final weight2: [-0.00000000e+00 -1.16140799e-03 -0.00000000e+00 -0.00000000e+00 -2.08932055e-02 -0.00000000e+00 3.63927823e-01 -1.80894902e-01 -0.00000000e+00 -0.00000000e+00 6.00020116e-01 -0.00000000e+00 -0.00000000e+00 1.33952651e+00 -1.95809416e-01]
and bias2: 0.020152768895894554
P0 is 1.980419392484462 and is supposed to be 0.9999999999999999
P1 is 2.294898766470035 and is supposed to be 0.41221474770752675
P2 is 2.634825090047943 and is supposed to be 0.04894348370484636
P3 is 3.063495983069266 and is supposed to be 0.04894348370484647
P4 is 3.703926313288953 and is supposed to be 0.41221474770752686
P5 is 4.687589719848333 and is supposed to be 1.0
P6 is 6.0040720809547565 and is supposed to be 1.5877852522924731
P7 is 7.515050843947688 and is supposed to be 1.9510565162951536
P8 is 9.103515137911428 and is supposed to be 1.9510565162951536
P9 is 10.711685202576104 and is supposed to be 1.5877852522924734
P10 is 12.316275328055655 and is supposed to be 1.0000000000000002

In addition to the previous exercise,the dropout technique gives more stability to the system results by reducing the nuber of weights.

8 .



This is the contour plot of the F(x) function,we used python's matplotlib library in the file ask8.py that we added in the code folder.

9 .

Adam with $\beta_1 = 0, \beta_2$ really close to 1 and a replacement of a by an annealed version $a_t = at^{-1/2}$,which is

$$\theta_t - \alpha \cdot t^{-1/2} \cdot \widehat{m}_t / \sqrt{\lim_{\beta_2 \to 1} \widehat{v}_t} = \theta_t - \alpha \cdot t^{-1/2} \cdot g_t / \sqrt{t^{-1} \cdot \sum_{i=1}^{t} g_t^2} =$$

$$\theta_t - \alpha \cdot g_t / \sqrt{\sum_{i=1}^{t} g_t^2}.$$

corresponds to Adagrad with β_2 converging to 1 from from bellow,and so

$$\lim_{\beta_2 \to 1} \widehat{v}_t = t^{-1} \cdot \sum_{i=1}^{t} g_t^2$$

10 .

Problem 10:

2)

#	name	size	neurons
0	input	1x28x28	0
1	conv2d1	32x24x24	24x24x32 = 18.432
2	maxpool1	32x12x12	12x12x32 = 4.608
3	conv2d2	32x10x10	10x10x32 = 3.200
4	maxpool2	32x5x5	5x5x32 = 800
5	dense	256	256
6	output	10	10

So in the network, we have a total of
18.432 + 4.608 + 3.200 + 800 + 256 + 10 = 27306

3)

#	name	size	connections to neurons
0	input	1x28x28	0
1	conv2d1	32x24x24	24x24x32x(5x5 +1) = 344.448
2	maxpool1	32x12x12	12x12x32x(2x2) = 18.432
3	conv2d2	32x10x10	10x10x32x(3x3+1) = 32.000
4	maxpool2	32x5x5	5x5x32(2x2) = 3.200
5	dense	256	801x 256 = 205.056
6	output	10	257x 10 = 2.570

So in the network, we have a total of

344.448 + 18.432 + 32.000 + 3.200 + 205.056 + 2.570 = 605.706

1)

#	name	size	weights + bias
0	input	1x28x28	0
1	conv2d1	32x24x24	5*5*1 + 1 = 26
2	maxpool1	32x12x12	0
3	conv2d2	32x10x10	3*3*32 + 1 = 289
4	maxpool2	32x5x5	0
5	dense	256	32*5*5 + 1 = 801
6	output	10	256 + 1 = 257

I assume that each pooling layer hasn't parameters

4)

#	name	size	independent parameters
0	input	1x28x28	0
1	conv2d1	32x24x24	(5*5*1 + 1)*32 = 832
2	maxpool1	32x12x12	0
3	conv2d2	32x10x10	(3*3*32+1)*32 = 9.248
4	maxpool2	32x5x5	0
5	dense	256	(32*5*5 + 1)*256 = 205.056
6	output	10	(256+1)*10 = 2.570

So in the network, we have a total of 832 + 9.248 + 205.056 + 2.570 = 217.706 independent learnable parameters.

11 .

Input1:

0	0	0	0	1	1	1	1	0	0	0	0

Convolution with:

-1	0	1
1	0	-1

Equals:

0	0	1	1	0	0	-1	-1	0	0
0	0	-1	-1	0	0	1	1	0	0

Max pooling with stride=2 and filter size=2:

0	1	0	-1	0
0	-1	0	1	0

Convolution with:

-1	0	1
1	0	-1

Equals:

0	-4	0

Fully connected layer:

4	-4

Sigmoid activation($1/(1+e^{-x})$)=

0.99	0.1

Input2:

1	1	1	1	0	0	0	0	1	1	1	1

Convolution with:

-1	0	1
1	0	-1

Equals:

0	0	-1	-1	0	0	1	1	0	0
0	0	1	1	0	0	-1	-1	0	0

Max pooling with stride=2 and filter size=2:

0	-1	0	1	0
0	1	0	-1	0

Convolution with:

-1	0	1
1	0	-1

Equals:

0	4	0

Fully connected layer:

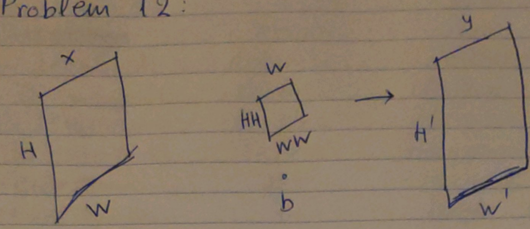
-4	4

Sigmoid activation($1/(1+e^{-x})$)=

0.1	0.99

12 .

Problem 12:



stride = S   with no padding

$$y_{ij} = \left(\sum_{k=1}^{HH} \sum_{l=1}^{WW} w_{kl} \, x_{si+k-1,\, sj+l-1} \right) + b$$

$$\forall (i,j) \in [1, H'] \times [1, W']$$

So the gradient of convolution layer when convolution stride is greater than 1 is:

$$dw_{si+k-1,\, sj+l-1} = \sum_{i=1}^{H'} \sum_{j=1}^{W'} dy_{ij} \cdot x_{si+k-1,\, sj+l-1}$$

$$\forall (k,l) \in [1, HH] \times [1, WW]$$

13 .

When the gradient includes overlapping regions with the same parameters, the gradients values from the source can be summed together, because the operations in the forward phase are also additions which means that $\frac{d}{dx}(y + z) = \frac{dy}{dx} + \frac{dz}{dx}$.This is exactly the case with max pooling as well.