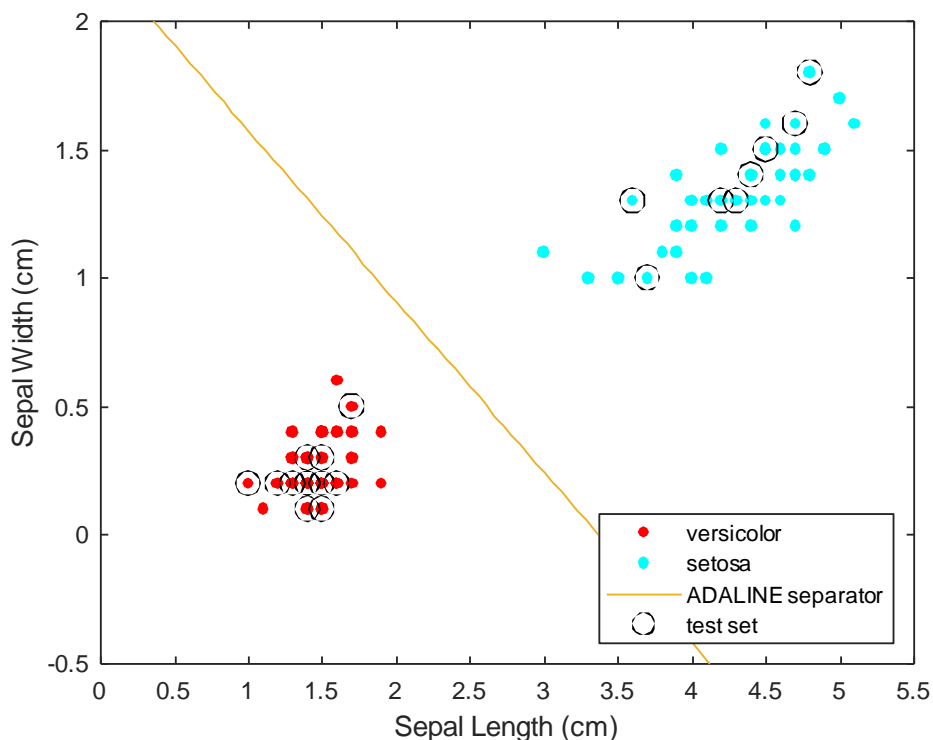


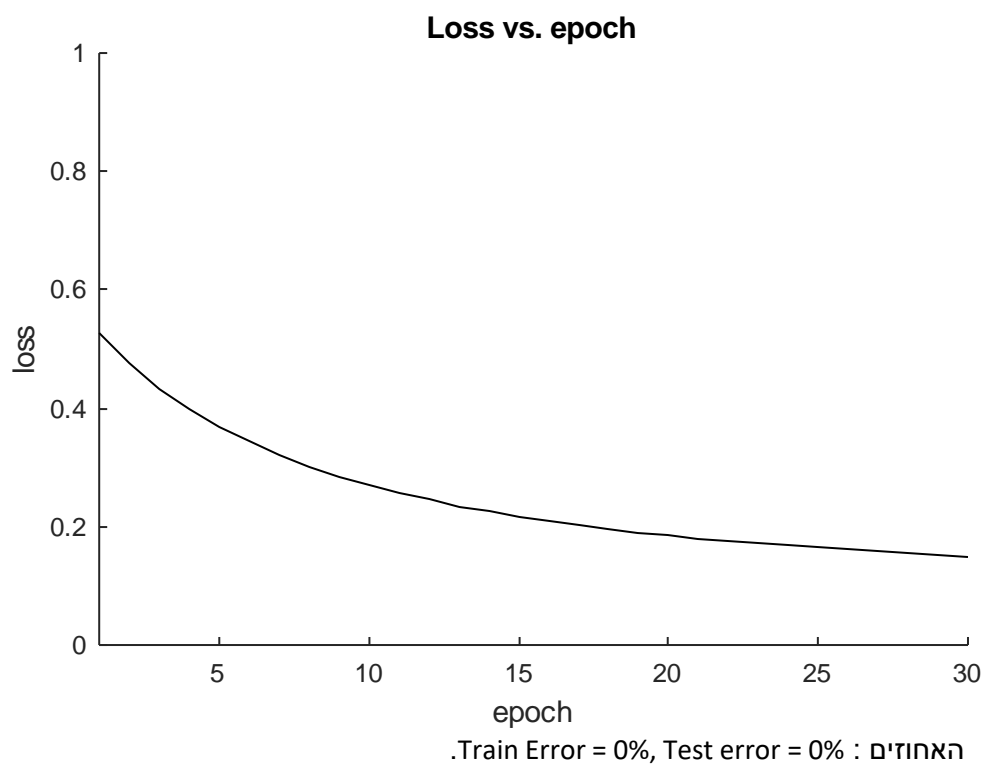
למידה עמוקה – מעבדה – ניסוי 1

אלמוג אדטו, ליאור וובצ'וק

חלק א'

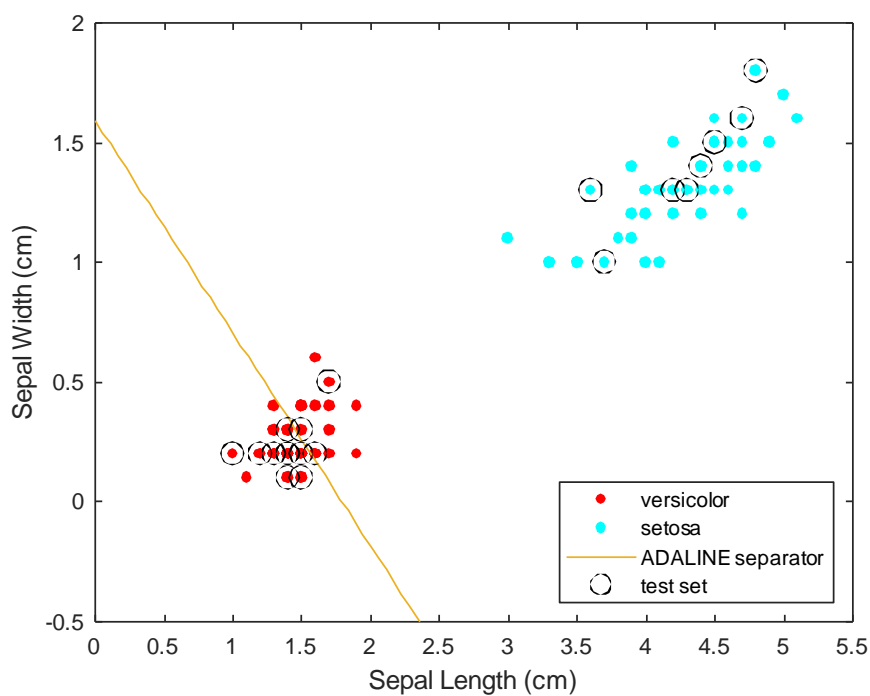
1. עבור שכבה לינארית נקבל ווקטור באורך N (שכן מחשבים $W^T A$ כאשר A היא מטריצה $n \times m$ כפי שנתון).
עבור שכבת סיגמואיד נקבל עדיין ווקטור באורך N (שכן פונק' הסיגמואיד מופעלת על כל איבר).
עבור שכבת NLL נקבל סקלר (שכן פונ' ה-NLL סוכמת את הביטויים עם ה-Log).
2. ממדי הגרדיאנטים הם מימדי מטריצה $n \times m$, כאשר m הוא גודל הווקטור עליו מפעילים גרדיאנט, ו- n הוא גודל הווקטור לפיו מתבצעת פעולת הגזירה.
כאשר מסתכלים על כל שכבה בנפרד: עבור שכבת ה-NLL נקבל ווקטור $n \times 1$. עבור שכבת הסיגמואיד נקבל סקלר. עבור השכבה הלינארית נקבל $n \times m$.
3. בקוד.
4. יש להפעיל במעבר הקדמי את השכבות לפי הסדר הבא: שכבה לינארית, סיגמואיד, NLL.
במעבר האחורי יש להפעיל בסדר ההפוך: NLL, סיגמואיד, לינארית.
5. נצפה לקו לינארי המפריד בין 2 הקבוצות.
6. בקוד.
7. קצב התכנסות 0.02 :

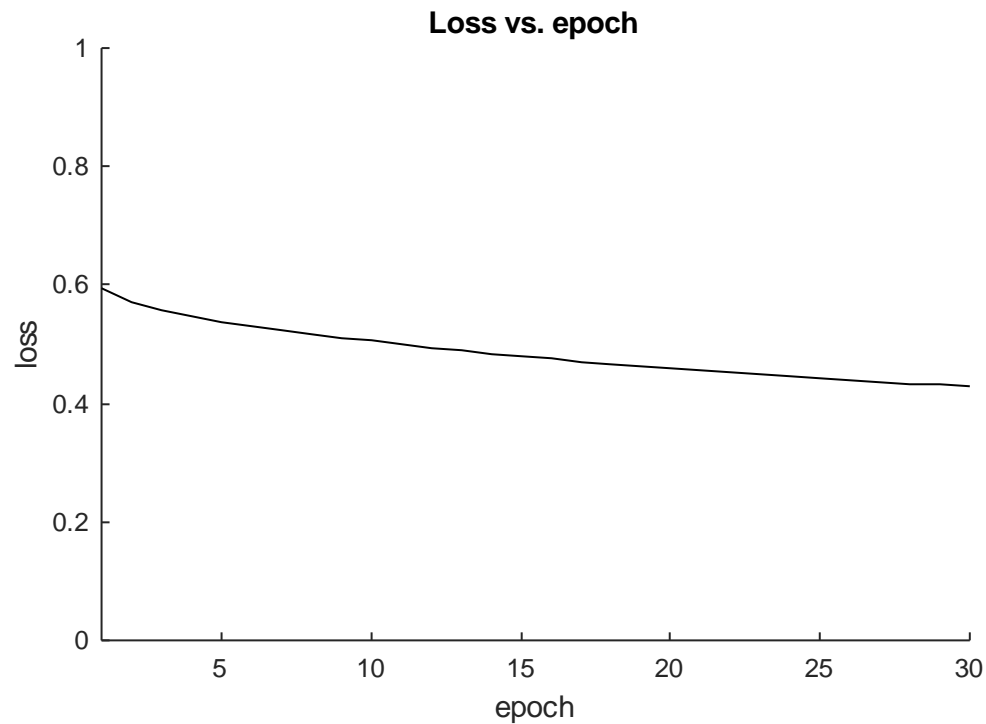




הסבר: ניתן לראות שקצב ההתכנסות הוא טוב, השגיאה יורדת ממש, והקו הלינארי מתכנס לערך נכון.

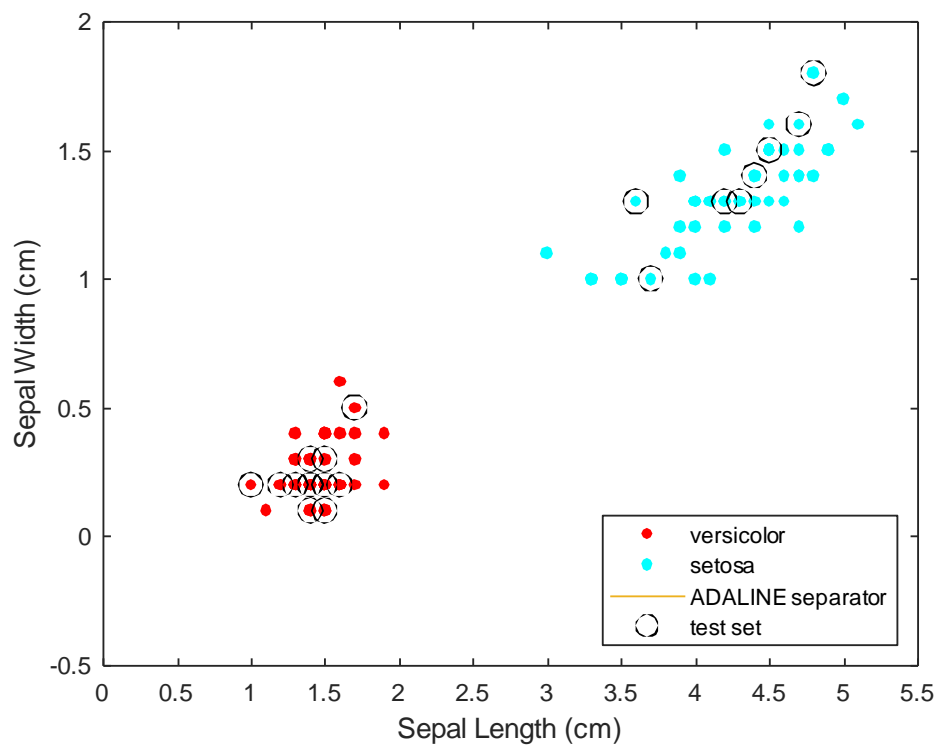
קצב התכנסות 0.03 :

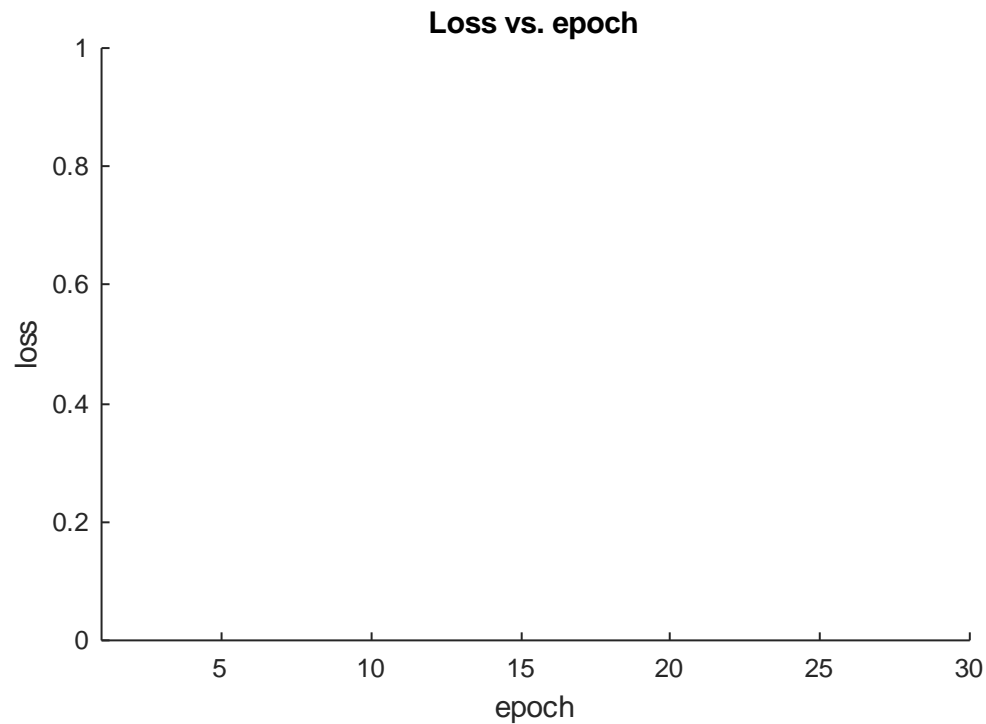




האחוזים : Train Error = 17.5%, Test error = 15% .
 הסבר: ניתן לראות שקצה ההתכנסות איטי מדי, ולכן ערך השגיאה לא מספיק להתכנס ולהגיע לערך נמוך מספיק.

קצב התכנסות 2 :





האחוזים אינם מחושבים.

הסבר: ניתן לראות שבגלל שערך ההתכנסות גבוה מדי יש התבדרות ולא ניתן לחשב בכלל את הפלט הרצוי.

חלק ב'

1. שלב 1: $w_1^T x + b_1$
- שלב 2: $\sigma(w_1^T x + b_1)$
- שלב 3: $w_2^T \sigma(w_1^T x + b_1) + b_2$
- שלב 4: $\text{softmax}(w_2^T \sigma(w_1^T x + b_1) + b_2)$
- שלב 5: $l(\text{softmax}(w_2^T \sigma(w_1^T x + b_1) + b_2), y)$

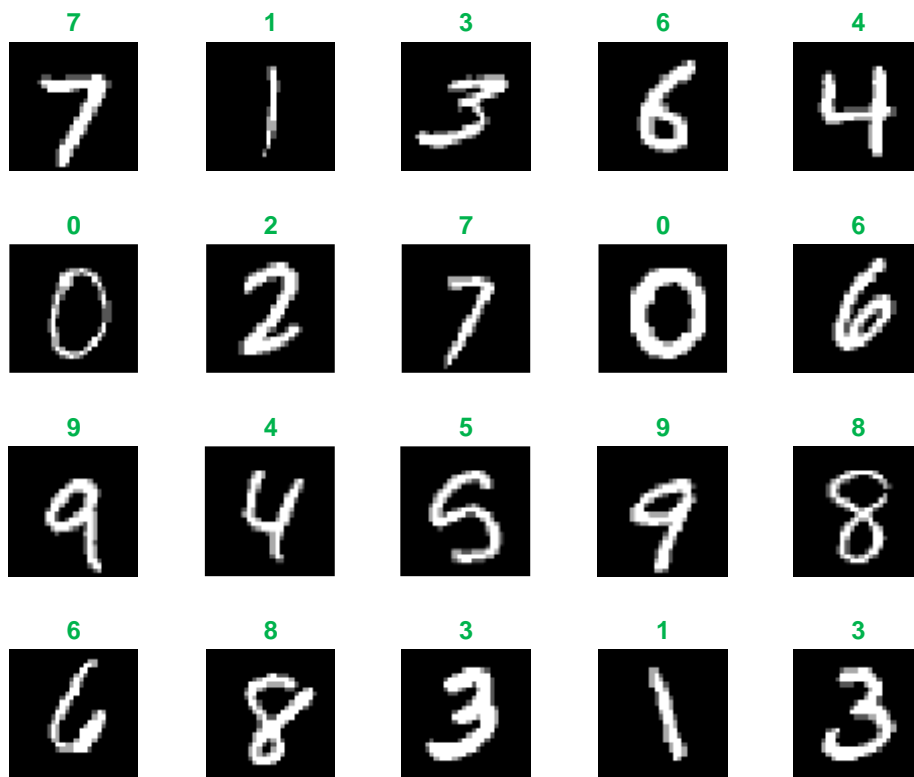
2. בקוד.

3. בקוד.

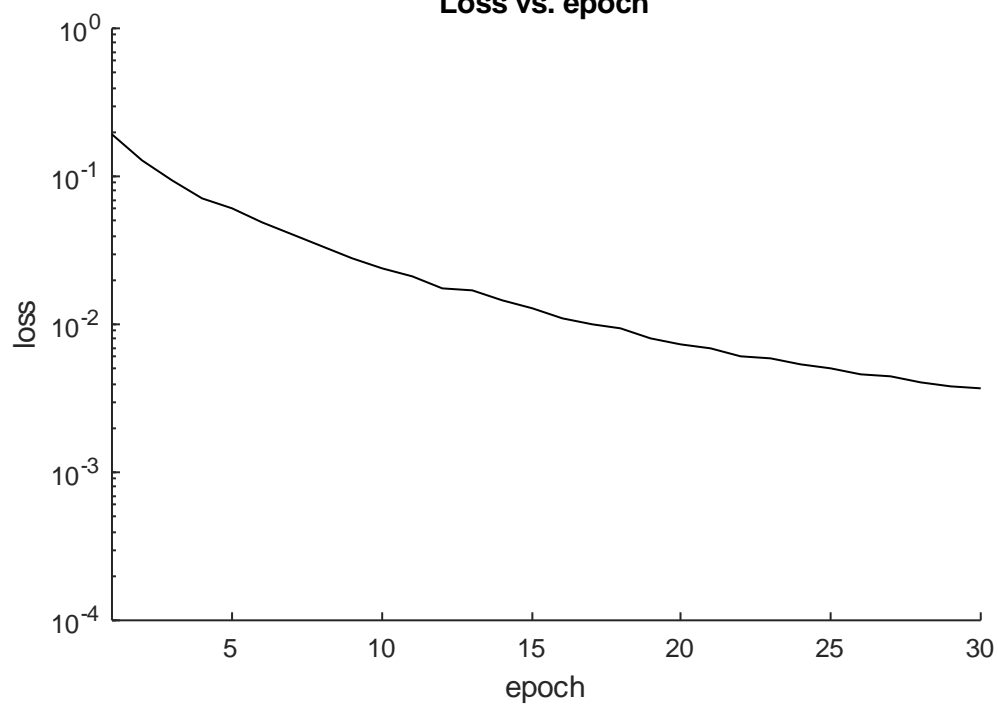
4. בקוד.

5. התוצאות:

MNIST validation results



Loss vs. epoch



6. מקום וזמן – השכבה הנסתרת תגדיל את סיבוכיות המקום והזמן של החישוב.

דיוק – מצד אחד, הגדלת השכבה הנסתרת גורמת להגדלת הדיוק שבסיווג. מצד שני – דבר זה יכול להיות מזיק שכן עלול לגרום ל-overfitting, ולכן השגיאה של סט הולידציה תגדל.

7. היתרון הוא כמובן למידה יותר מהירה.

החסרון הוא שעקב כך שיש פחות שלבים בתהליך הלמידה, עדכון המשקלים מתבצע על סמך פחות דוגמאות ולכן דיוק התוצאות מעט יורד.

חלק ג'

1. בקוד.

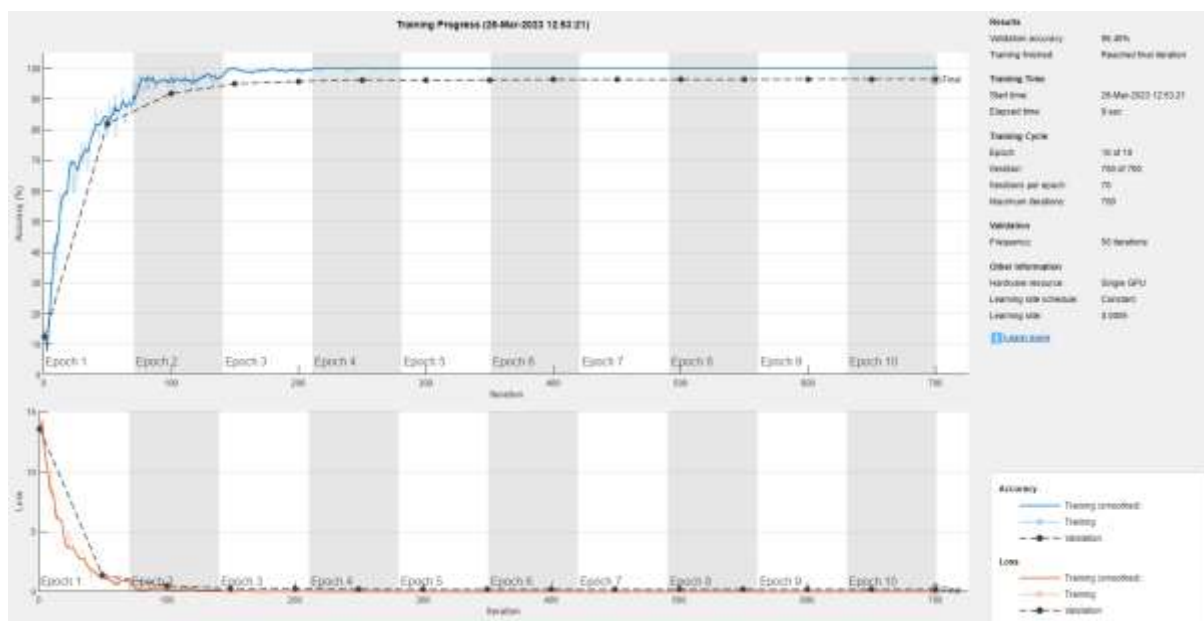
2. Optimizer = 'sdgm' : סוג האופטימיזציה לתהליך האימון – במקרה זה מדובר ב-stochastic gradient descent with momentum – כלומר מזעור פונקציית השגיאה על ידי gradient descent עם מומנטום.

LearnRate : גודל הצעד במהלך תהליך האימון.

Epoches : מספר האפוקים (חזרות) שלאחריהן יעצור תהליך האימון.

3. תוצאות ההרצה :

Learn Rate = 5e-4:



Training Progress:

=====						
Epoch	Iteration	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	
=====						
1.37	1.42	81.80%	78.13%	50	1	
0.49	0.01	91.70%	99.22%	100	2	
0.27	0.01	94.90%	99.22%	150	3	
0.25	0.08	95.70%	98.44%	200	3	
0.20	0.00	96.20%	100.00%	250	4	
0.19	0.00	96.00%	100.00%	300	5	
0.18	0.00	96.20%	100.00%	350	5	
0.18	0.00	96.30%	100.00%	400	6	
0.18	0.00	96.30%	100.00%	450	7	
0.18	0.00	96.30%	100.00%	500	8	
0.18	0.00	96.30%	100.00%	550	8	
0.18	0.00	96.40%	100.00%	600	9	
0.18	0.00	96.40%	100.00%	650	10	
0.18	0.00	96.40%	100.00%	700	10	
0.18	0.00	96.40%	100.00%	700	10	
=====						

Validation error: 3.60%

Learn Rate = 8e-4:

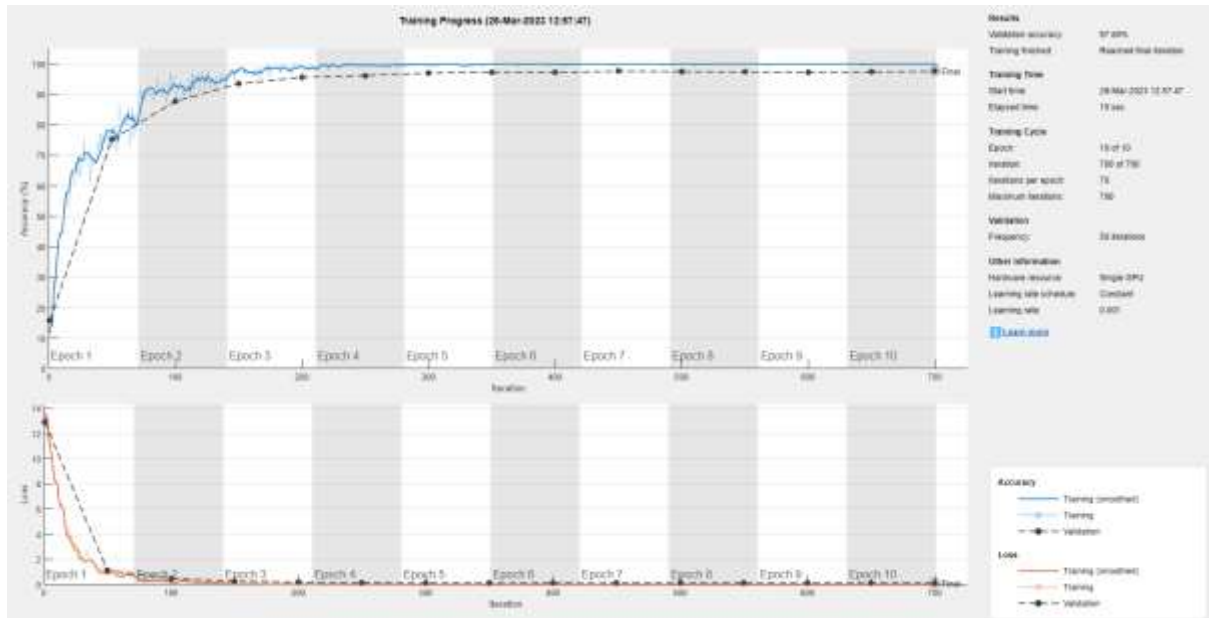


Training Progress:

=====						
Epoch	Iteration	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	
=====						
1	0	1.08	1.06	73.20%	70.31%	50 1
1	100	0.48	0.24	87.20%	91.41%	100 2
1	200	0.34	0.13	91.40%	94.53%	150 3
1	300	0.31	0.10	92.10%	98.44%	200 3
1	400	0.29	0.04	94.30%	99.22%	250 4
1	500	0.26	0.03	94.60%	99.22%	300 5
1	600	0.26	0.01	94.50%	100.00%	350 5
1	700	0.26	0.00	94.40%	100.00%	400 6
1	800	0.26	0.01	94.50%	100.00%	450 7
1	900	0.26	0.00	94.60%	100.00%	500 8
1	1000	0.25	0.00	94.70%	100.00%	550 8
1	1100	0.25	0.00	95.10%	100.00%	600 9
1	1200	0.25	0.00	95.20%	100.00%	650 10
1	1300	0.25	0.00	95.20%	100.00%	700 10
1	1400	0.25	0.00	95.20%	100.00%	700 10

Validation error: 4.80%

Learn Rate = 1e-3:



Training Progress:

[Epoch | Iteration | Training | Validation | Training | Validation]

| | | Accuracy | Accuracy | Loss | Loss |

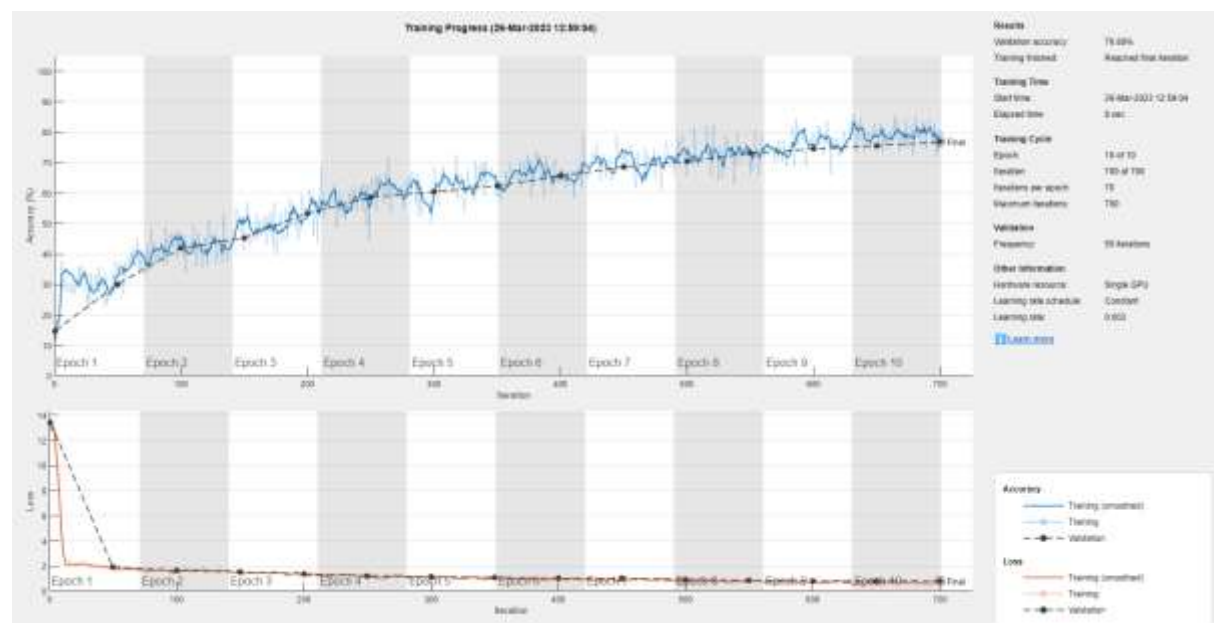
	1.14		0.79		75.20%		78.91%		50		1	
	0.49		0.18		87.70%		92.19%		100		2	
	0.28		0.17		93.50%		96.88%		150		3	
	0.19		0.07		95.60%		98.44%		200		3	
	0.16		0.03		96.20%		99.22%		250		4	
	0.14		0.01		97.00%		100.00%		300		5	
	0.14		0.01		97.30%		100.00%		350		5	
	0.14		0.00		97.30%		100.00%		400		6	
	0.14		0.00		97.60%		100.00%		450		7	
	0.14		0.00		97.50%		100.00%		500		8	
	0.14		0.00		97.40%		100.00%		550		8	
	0.14		0.00		97.30%		100.00%		600		9	

	0.14		0.00		97.40%		100.00%		650		10	
	0.14		0.00		97.60%		100.00%		700		10	
	0.14		0.00		97.60%		100.00%		700		10	

=====

Validation error: 2.40%

Learn Rate = 3e-3:



Training Progress:

=====

	Epoch		Iteration		Training		Validation		Training		Validation	
--	-------	--	-----------	--	----------	--	------------	--	----------	--	------------	--

			Accuracy		Accuracy		Loss		Loss	
--	--	--	----------	--	----------	--	------	--	------	--

=====

	1.92		1.88		30.00%		29.69%		50		1	
	1.64		1.66		41.90%		46.88%		100		2	
	1.54		1.60		45.30%		41.41%		150		3	
	1.38		1.34		53.40%		53.13%		200		3	
	1.23		1.09		58.60%		62.50%		250		4	
	1.14		0.90		60.40%		67.97%		300		5	
	1.09		1.19		62.40%		58.59%		350		5	
	1.02		0.94		65.60%		67.97%		400		6	

	1.02		1.11		68.60%		68.75%		450		7	
	0.95		0.93		70.40%		65.63%		500		8	
	0.85		0.89		73.10%		66.41%		550		8	
	0.80		0.61		74.50%		79.69%		600		9	
	0.82		0.59		75.50%		79.69%		650		10	
	0.81		0.61		76.80%		79.69%		700		10	
	0.81		0.61		76.80%		79.69%		700		10	

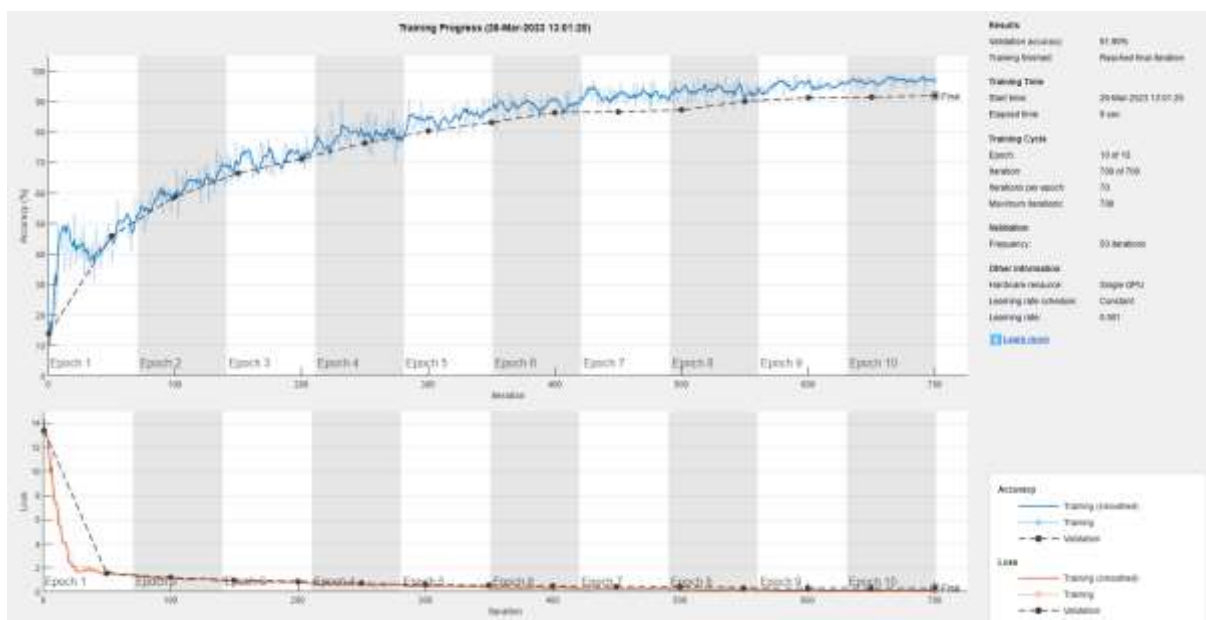
|=====|

Validation error: 23.20%

ניתן לראות כי עבור שני קצבי הלמידה הראשונים מדובר בקצבים איטיים יחסית, אשר משיגים תוצאות סבירות עבור שגיאת הולידציה. בהעלאת קצב הלמידה, גודל הצעד בתהליך ה-SGD גדל, וניתן להתקרב יותר לנק' המינימום של פונקציית ההפסד. לכן, עבור הקצב השלישי, מדובר בקצב שלדעתינו הוא הטוב ביותר, ואכן משיג התאמה טובה ושגיאת ולידציה נמוכה ביותר. הקצב הרביעי הוא גבוה מדי, מפני שישנה בעיה של חוסר התכנסות לנק' המינימום הגלובלי (האלגוריתם "מפספס" אותה). ואכן - אין התכנסות לנקודת המינימום של ה-Loss function, ושגיאת הוולידציה גבוהה.

4. כעת נשנה את גדלי השכבה הנסתרת, עבור קצב למידה אידיאלי $1e-3$:

:200

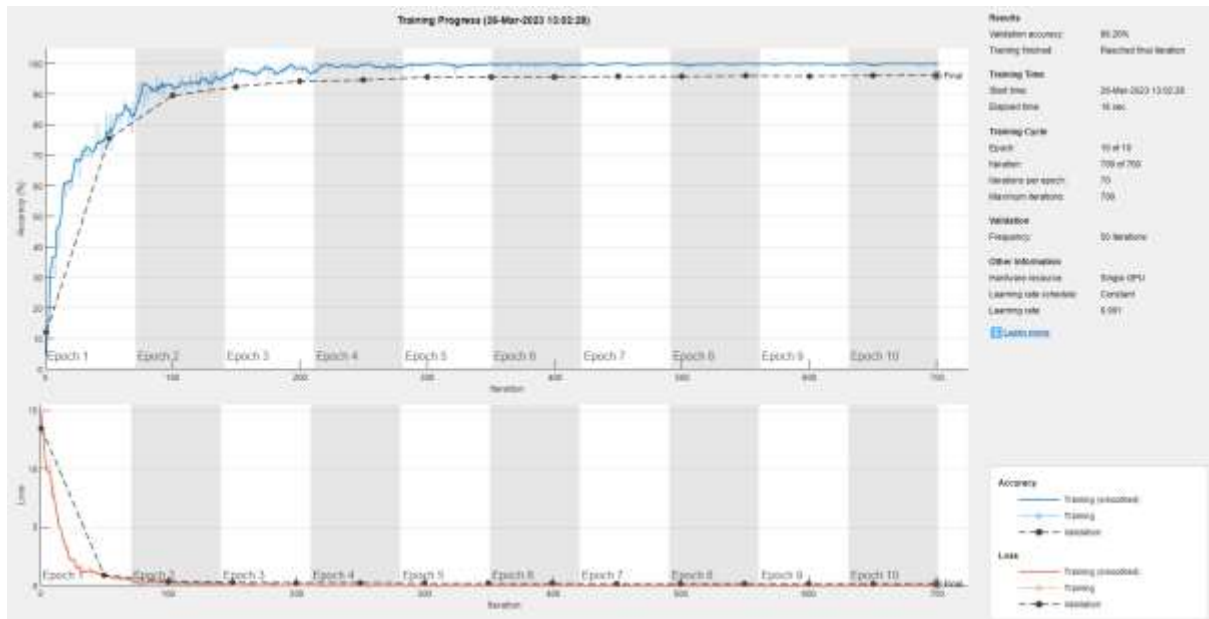


Training Progress:

=====						
Epoch	Iteration	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	
=====						
1.55	1.55	45.80%	42.97%	50	1	
1.19	1.18	58.80%	58.59%	100	2	
0.99	1.14	66.50%	64.06%	150	3	
0.87	0.85	71.20%	71.09%	200	3	
0.71	0.71	76.40%	75.78%	250	4	
0.60	0.47	80.30%	85.16%	300	5	
0.53	0.34	83.20%	85.94%	350	5	
0.49	0.26	86.40%	90.63%	400	6	
0.45	0.17	86.70%	93.75%	450	7	
0.43	0.13	87.20%	94.53%	500	8	
0.35	0.35	90.10%	87.50%	550	8	
0.34	0.07	91.20%	98.44%	600	9	
0.33	0.08	91.40%	96.09%	650	10	
0.32	0.09	91.90%	96.88%	700	10	
0.32	0.09	91.90%	96.88%	700	10	
=====						

Validation error: 8.10%

500:



Training Progress:

=====						
Epoch	Iteration	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	
=====						
0.89	0.90	75.40%	76.56%	50	1	
0.36	0.18	89.60%	92.97%	100	2	
0.28	0.07	92.50%	97.66%	150	3	
0.23	0.04	94.30%	98.44%	200	3	
0.23	0.04	94.60%	99.22%	250	4	
0.20	0.01	95.60%	100.00%	300	5	
0.19	0.01	95.60%	100.00%	350	5	
0.19	0.00	95.70%	100.00%	400	6	
0.18	0.00	95.80%	100.00%	450	7	
0.18	0.00	95.80%	100.00%	500	8	
0.17	0.00	96.00%	100.00%	550	8	
0.17	0.00	95.90%	100.00%	600	9	
0.17	0.02	96.10%	99.22%	650	10	
0.17	0.00	96.20%	100.00%	700	10	
0.17	0.00	96.20%	100.00%	700	10	

Validation error: 3.80%

700:



Training Progress:

[Epoch | Iteration | Training | Validation | Training | Validation]

| | | Accuracy | Accuracy | Loss | Loss |

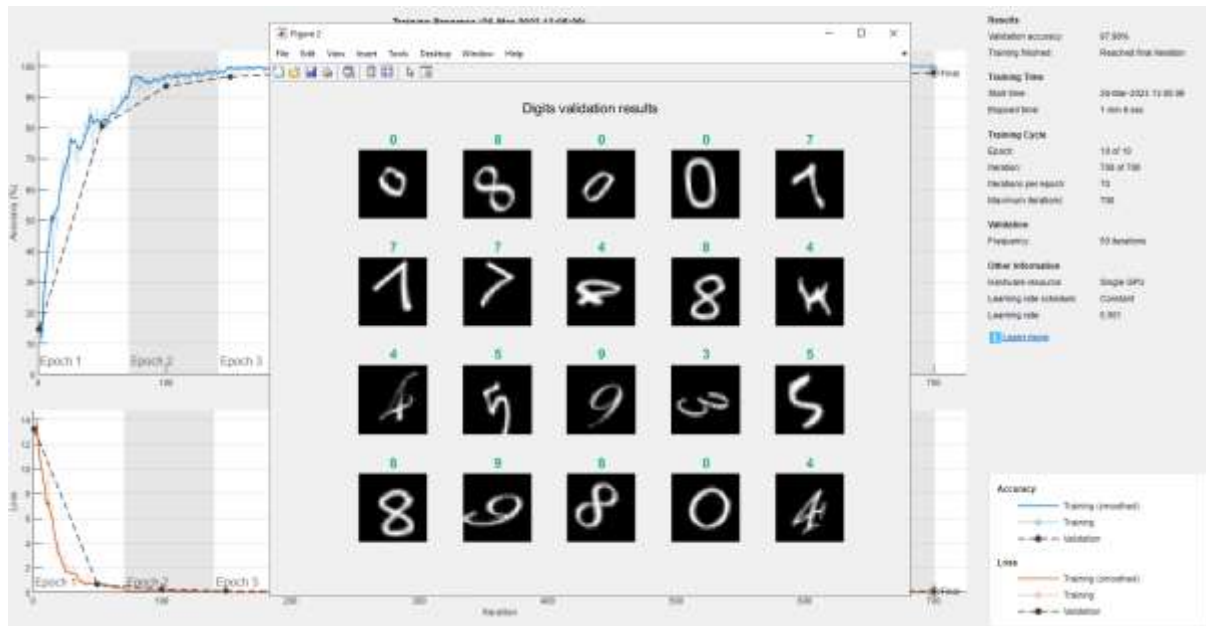
	0.68		0.65		86.40%		87.50%		50		1	
	0.28		0.08		94.40%		98.44%		100		2	
	0.17		0.06		95.50%		99.22%		150		3	
	0.14		0.00		97.10%		100.00%		200		3	
	0.13		0.00		97.40%		100.00%		250		4	
	0.12		0.00		97.60%		100.00%		300		5	
	0.12		0.00		97.80%		100.00%		350		5	
	0.12		0.00		97.90%		100.00%		400		6	
	0.12		0.00		97.80%		100.00%		450		7	
	0.12		0.00		97.90%		100.00%		500		8	
	0.12		0.00		97.90%		100.00%		550		8	
	0.12		0.00		97.90%		100.00%		600		9	

	0.12		0.00		97.90%		100.00%		650		10	
	0.12		0.00		98.00%		100.00%		700		10	
	0.12		0.00		98.00%		100.00%		700		10	

=====

Validation error: 2.00%

1000:



Training Progress:

=====

[Epoch	Iteration	Training	Validation	Training	Validation]
--------	-----------	----------	------------	----------	-------------

			Accuracy	Accuracy	Loss	Loss]
--	--	--	----------	----------	------	-------

=====

	0.68		0.63		80.50%		85.16%		50		1	
	0.25		0.12		93.50%		96.88%		100		2	
	0.13		0.04		96.60%		99.22%		150		3	
	0.11		0.01		97.80%		100.00%		200		3	
	0.10		0.00		97.80%		100.00%		250		4	
	0.10		0.00		97.80%		100.00%		300		5	
	0.10		0.00		97.80%		100.00%		350		5	
	0.10		0.00		97.90%		100.00%		400		6	
	0.10		0.00		97.80%		100.00%		450		7	

	0.10		0.00		97.90%		100.00%		500		8	
	0.10		0.00		97.80%		100.00%		550		8	
	0.10		0.00		97.80%		100.00%		600		9	
	0.10		0.00		97.80%		100.00%		650		10	
	0.10		0.00		97.90%		100.00%		700		10	
	0.10		0.00		97.90%		100.00%		700		10	

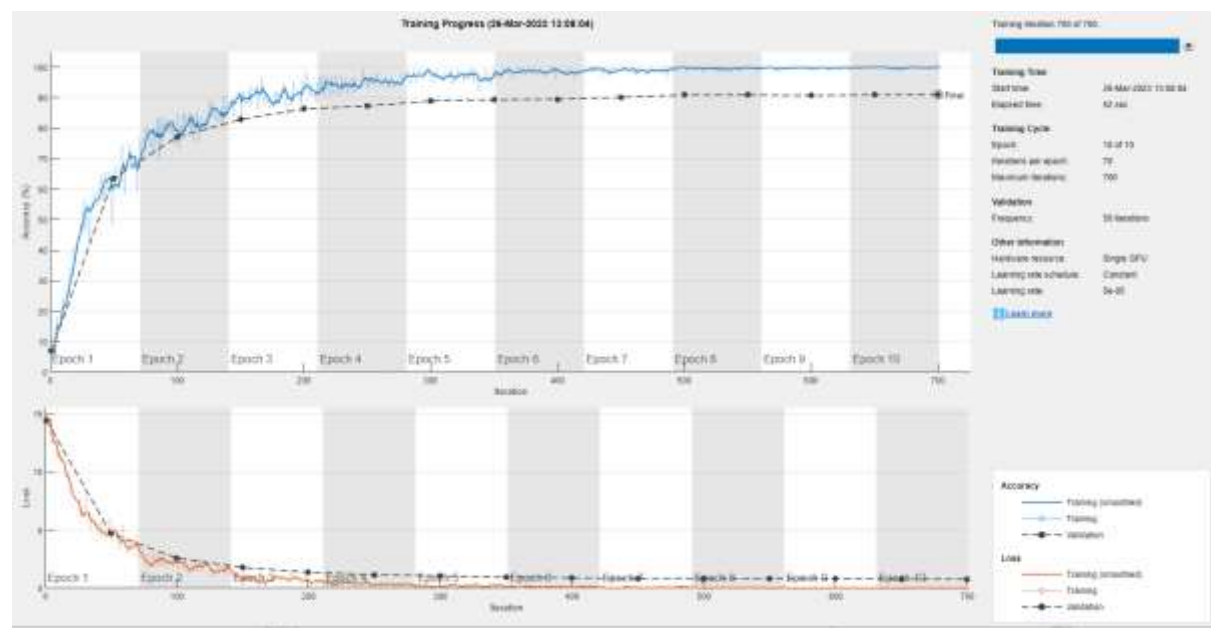
|=====|

Validation error: 2.10%

ניתן לראות שעם העלאת גודל השכבה הנסתרת עבור 3 הגדלים הראשונים – שגיאת הולידציה קטנה. הסיבה לכך היא שכעת ישנם יותר פרמטרים נלמדים ורזולוציית הלמידה גדלה – כלומר ניתן לסווג את התמונות בצורה טובה יותר. עם זאת, עבור העלאה נוספת של גודל השכבה הנסתרת מ-700 ל-1000 ניתן לראות כי שגיאת הולידציה גדלה שוב. הסיבה לכך היא Overfitting – כלומר התאמה יתרה לסט האימון על חשבון סט הולידציה, עקב גודל גדול מדי של השכבה הנסתרת. לכן, הגודל הטוב ביותר הוא 700.

5. נציג את התוצאות:

5e-5:



Training Progress:

|=====|

|Epoch | Iteration | Training | Validation | Training | Validation|

| | |Accuracy | Accuracy | Loss | Loss|

|=====|

| 4.71 | 5.49 | 63.50% | 58.59% | 50 | 1 |

| 2.58 | 2.42 | 77.10% | 75.78% | 100 | 2 |

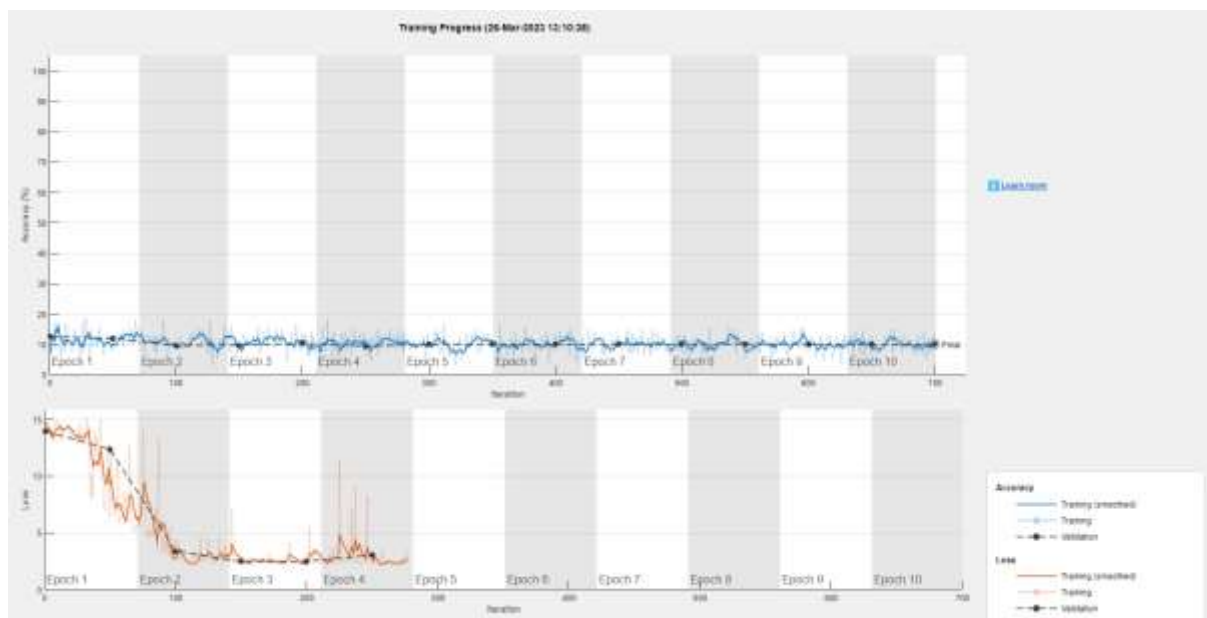
	1.77		0.70		82.80%		92.19%		150		3	
	1.35		0.77		86.30%		91.41%		200		3	
	1.15		0.37		87.30%		92.97%		250		4	
	1.05		0.12		88.90%		96.88%		300		5	
	0.93		0.01		89.30%		99.22%		350		5	
	0.89		0.06		89.50%		98.44%		400		6	
	0.83		0.01		90.10%		99.22%		450		7	
	0.82		0.01		90.90%		100.00%		500		8	
	0.82		0.01		90.90%		99.22%		550		8	
	0.80		0.00		90.70%		100.00%		600		9	
	0.77		0.01		90.90%		100.00%		650		10	
	0.77		0.00		91.00%		100.00%		700		10	
	0.77		0.00		91.00%		100.00%		700		10	

|=====|

Validation error: 9.00%

מדובר בגודל צעד קטן מדי בצורה קיצונית, מספר האפוקים אינו מספיק להגעה לתוצאה טובה עם גודל צעד כזה. לכן – גם שגיאת הולידציה גדולה. שגיאת האימון מגיעה ל-0% רק לקראת סוף התהליך, בניגוד לשאר הניסיונות בהם מספר אפוקים בודדים הספיקו להגעה ל-0% בשגיאת האימון.

6. התוצאות:



Training Progress:

|=====|

|Epoch | Iteration | Training | Validation | Training | Validation|

		Accuracy	Accuracy	Loss	Loss		
=====							
	12.37		6.44		12.00%		13.28%
	50		1				
	3.40		4.44		9.60%		9.38%
	100		2				
	2.48		2.82		9.70%		10.16%
	150		3				
	2.50		2.37		10.30%		13.28%
	200		3				
	3.05		2.42		9.40%		11.72%
	250		4				
	10.00%		9.38%		300		5
	NaN		NaN				
	10.00%		11.72%		350		5
	NaN		NaN				
	10.00%		13.28%		400		6
	NaN		NaN				
	10.00%		9.38%		450		7
	NaN		NaN				
	10.00%		7.03%		500		8
	NaN		NaN				
	10.00%		5.47%		550		8
	NaN		NaN				
	10.00%		7.81%		600		9
	NaN		NaN				
	10.00%		3.91%		650		10
	NaN		NaN				
	10.00%		10.16%		700		10
	NaN		NaN				
	10.00%		10.16%		700		10
	NaN		NaN				
=====							

Validation error: 100.00%

ניתן לראות כי עבור ערך צעד גדול בהגזמה, ה-SGD מתבדר – ותהליך הלמידה כלל אינו מגיע לסיומו.

7. הגרף העליון מציג את אחוזי הדיוק בסיווג – כלומר כמה דוגמאות אכן סווגו בצורה נכונה מתוך סך הדוגמאות הכולל. הגרף התחתון מציג את פונקציית ההפסד/המחיר (Loss function) – המחושב ע"י שכבת קלסיפיקציה שלאחריה חישוב פונקציית המחיר (למשל NLL). ניתן להתייחס לפונקציית המחיר כמדד לכמה רחוק הקלט המעובד מסיווג של תמונות כלשהן. לכן – מזעורו יביא באופן ישיר להגדלת אחוזי הדיוק בסיווג.