

תרגיל 1 - אלגוריתמים ביו אינטליגנטים

מגישים

אריאל וצלר 207458688

הדס אשל 206775074

קישור למחברת ה-colab של גוגל בה מימשנו את הקוד -

https://colab.research.google.com/drive/1wpX7xCWEBtZiAvgTcCfNBE4S1qzfDQj2?usp=s_haring

למחברת מצורפים הסברים, נתאר את הארכיטקטורה של הרשת, את התוצאות שלה וכן את הניסויים שערכנו.

MLP

תיאור הרשת

הרשת מורכבת מ-5 שכבות:

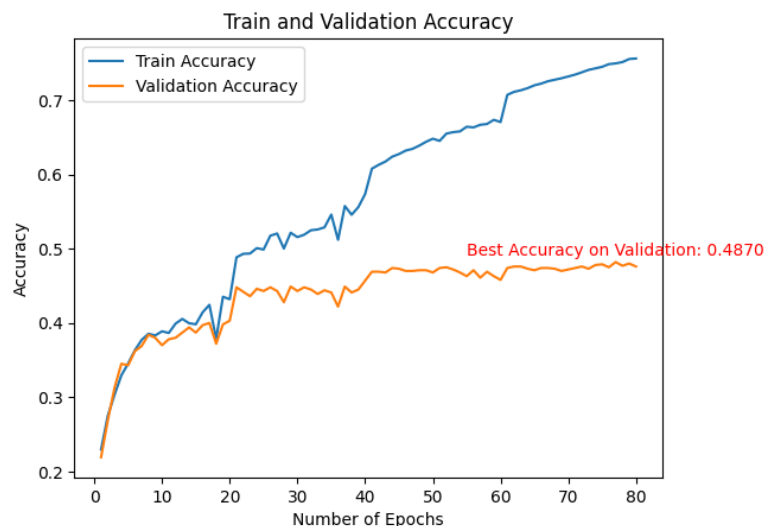
- (1) 3072 - קלט.
- (2) hidden layer. 550
- (3) hidden layer. 292
- (4) 10 - פלט.

בין כל זוג שכבות ישנה שכבת Relu.

batch size - גודל ה-batch הינו 128.

learning rate - תדירות הלמידה היא 0.001.

תוצאות אחוזי דיוק על הולידציה



הגענו ל-0.487 אחוזי דיוק על הולידציה.

במידה ותריץ את התא הראשון במחברת ואת התא השלישי תוכל לצפות בהדפסה של אחוזי הדיוק.

הוראות לביצוע inference

יש להריץ את התא הראשון (Getting data from google drive) במחברת ואת התא הרביעי (The model prediction for the test set) כדי לבצע inference על סט המבחן. מצורף הסבר במחברת.

הוראות לביצוע train

יש להריץ את התא הראשון (Getting data from google drive) במחברת ואת התא השלישי (The realization of the model, and the training of the model in the main function) כדי לבצע inference על סט המבחן. מצורף הסבר במחברת.

תוצאות על ה-test set

מצורף בקובץ output.txt. בנוסף, מופיע ב-section 4 במחברת ב-colab, ניתן להורדה.

ניסויים

אופטימוזציה על ערך ה-dropout

בחנו אפשרות להוסיף dropout לרשת. ביצענו מספר הרצות עם ערכים שונים ל-dropout ובכך הערכנו מה מוביל לביצועים הטובים ביותר.

נתאר את תצורת הרשת שעבורה ביצענו את הניסויים הללו.

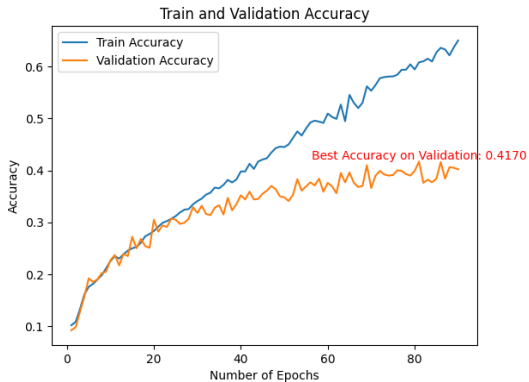
הרשת מורכבת מ-5 שכבות:

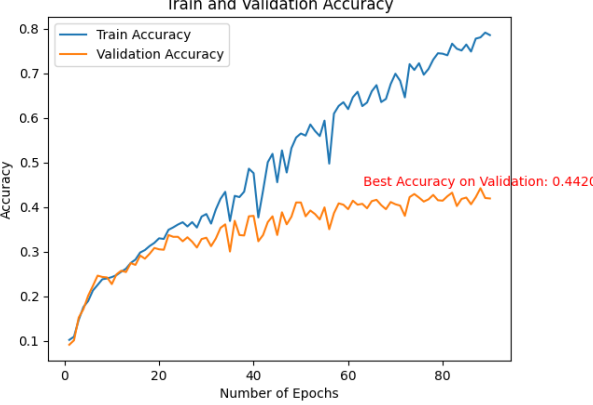
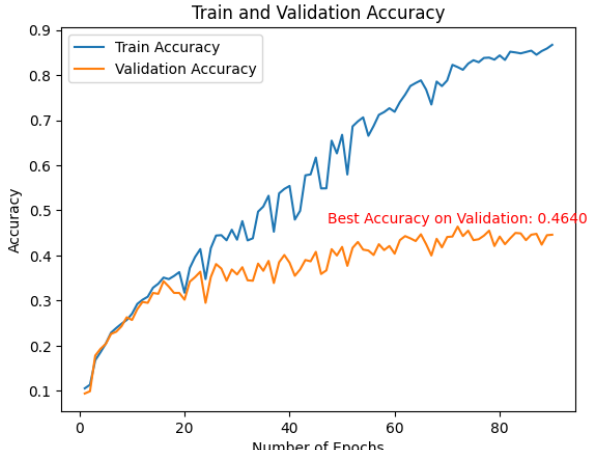
- (1) 3072 - קלט.
- (2) 1024 - hidden layer.
- (3) 200 - hidden layer.
- (4) 100 - hidden layer.
- (5) 10 - פלט.

בין כל זוג שכבות ישנה שכבת Relu. בנוסף, הכפלנו את סט האימון על ידי שימוש באוגמנטציה. הפעלנו אוגמנטציה כבדה: הוספת רעש רנדומי לקלט, הזזת התמונה, סיבוב התמונה וכיבוי רנדומי של תאים בקלט.

אימנו את הרשת על פני סט האימון עם תוספת האוגמנטציה על סט האימון.

קיבלנו את התוצאות הבאות:

ערך ה-dropout	תוצאות	הערות
0.5	 <p>Train and Validation Accuracy</p> <p>Best Accuracy on Validation: 0.4170</p>	הדיוק על ה-training set מגיע ל-65% לכל היותר. הדיוק על ה-validation set מגיע ל-40% באפוק 69, ואינו יציב שם, נראה שמתייצב החל מהאפוק ה-88. הדיוק על ה-validation set הוא לכל היותר 41%. <u>זוה אינו אחוז דיוק שמרצה אותנו.</u>

<p>הדיוק על ה-training set מגיע ל-79% לכל היותר. הדיוק על ה-validation set הוא לכל היותר 44%, אך זהו שיא חריג, הדיוק נראה יותר יציב סביב ה-41%. <u>נזת</u> <u>אינו אחוז דיוק שמרצה אותנו.</u></p>		0.4
<p>הדיוק על ה-training set מגיע ל-85% לכל היותר. הדיוק על ה-validation set הוא לכל היותר 46.4%. ונראה יציב סביב ה-44%.</p>		0.3

המסקנה המתבקשת היא שהערך האופטימלי לתדירות ה-dropout הוא 0.3.

אופטימיזציה על האגומנטציה

בחנו אפשרות להוסיף אוגמנטציה על הקלט כחלק מתהליך האימון. ביצענו מספר הרצות עם הגדרות שונות לאוגמנטציה ובכך הערכנו מה מוביל לביצועים הטובים ביותר.

נתאר את תצורת הרשת שעבורה ביצענו את הניסויים הללו.

הרשת מורכבת מ-5 שכבות:

6) 3072 - קלט.

7) hidden layer - 1024

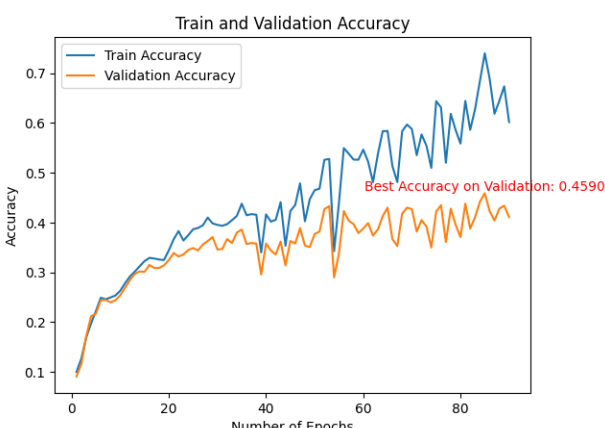
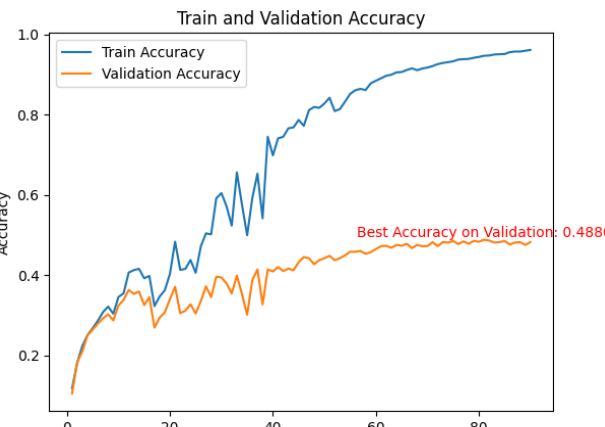
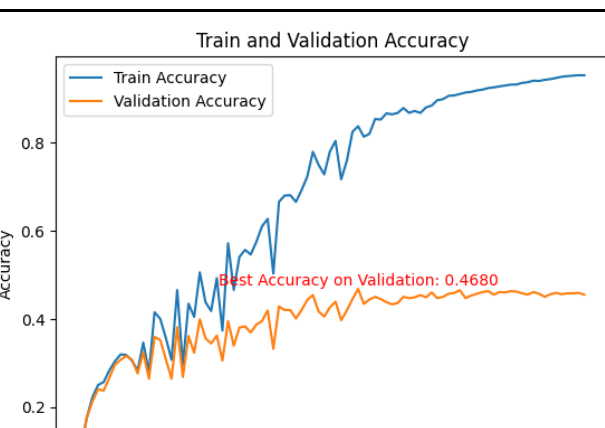
8) hidden layer - 200

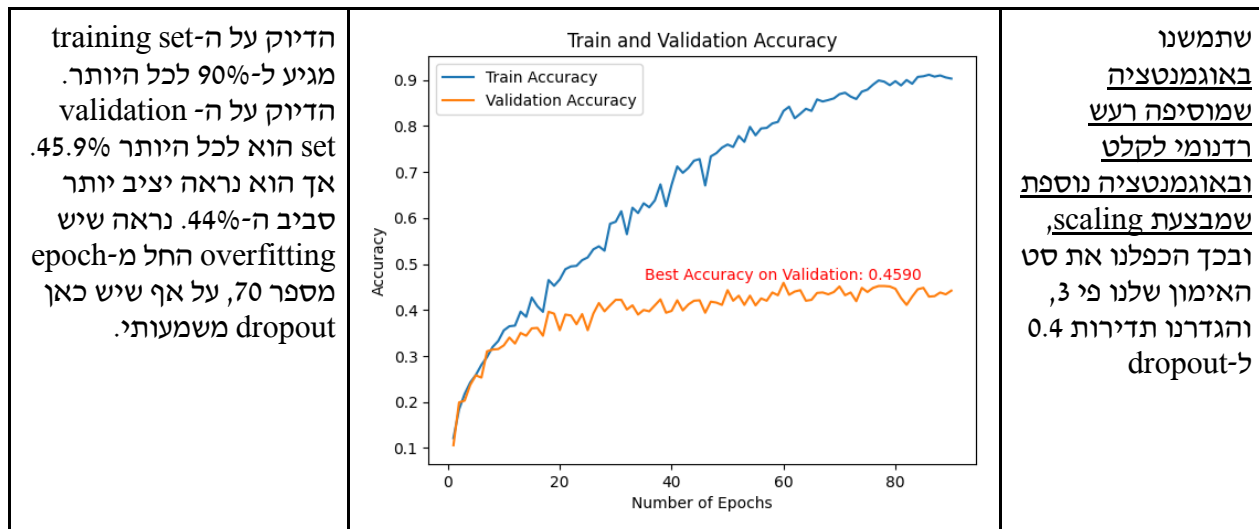
9) hidden layer - 100

10) 10 - פלט.

בין כל זוג שכבות ישנה שכבת Relu.

קיבלנו את התוצאות הבאות:

הערות	תוצאות	ערך ה-dropout
<p>הדיוק על ה-training set מגיע ל-60% לכל היותר.</p> <p>הדיוק על ה-validation set הוא לכל היותר 45.9%.</p> <p>ונראה יציב סביב ה-41%.</p>		<p>ריצת הרשת ללא אוגמנטציה</p>
<p>הדיוק על ה-training set מגיע ל-96% לכל היותר.</p> <p>הדיוק על ה-validation set הוא לכל היותר 48.8%.</p> <p>ונראה יציב סביב ה-48%.</p>		<p>הכפלנו את סט האימון על ידי שימוש באוגמנטציה אוגמנטציה שמוסיפה רעש רנדומי לקלט</p>
<p>כאשר הדיוק על ה-training set מגיע ל-95% לכל היותר. הדיוק על ה-validation set הוא לכל היותר 46.8%, הדיוק נראה יותר יציב סביב ה-46%.</p> <p>אך החלק מ-epoch מספר 66 נראה שיש overfitting.</p>		<p>הכפלנו את סט האימון על ידי שימוש באוגמנטציה. הפעלנו אוגמנטציה כבדה: הוספת רעש רנדומי לקלט, הזזת התמונה, סיבוב התמונה וכיבוי רנדומי של תאים בקלט</p>



המסקנות שעולות מן ההרצות היא שמיותר להכפיל את הקלט פי 3, זה לא מביא לתוצאות יותר טובות על סט הולידציה. עדיף להשתמש באוגמנטציה שמוסיפה רעש רנדומי לקלט על פני אוגמנטציה כבדה שמבצעת הוספת רעש רנדומי לקלט, הזזת התמונה, סיבוב התמונה וכיבוי רנדומי של תאים בקלט. וגם אז, נדרש להשתמש ב-dropout משום שאחוז שניכר כי ישנו overfitting על ה-training set.

אופטימיזציה על אתחול המשקולות

לפני שקיבענו את שיטת אתחול המשקולות ביצענו מספר הרצות שונות כאשר בכל אחת מהריצות אתחלנו בצורה שונה את המשקולות.

1) אתחול Xavier (ידוע גם בשם Glorot Initialization), מימשנו אותו כך:

```
limit = np.sqrt(6 / (output_size + input_size))
self.weights = np.random.uniform(low=-limit, high=limit,
size=(output_size, input_size))
self.biases = np.zeros((output_size, 1))
```

האתחול הזה לא הוביל את המודל ללמידה, וזאת משום שטכניקת האתחול הזו יותר מתאימה לפונקציות אקטיבזציה שהן סימטריות סביב ה-0 כמו sigmoid ואנו ביצענו שימוש ב-relu. ווארינט של אתחול Xavier שמוותאם יותר לפונקציית אקטיבזציה relu:

```
limit = np.sqrt(2 / input_size)
self.weights = np.random.randn(output_size, input_size) * limit
self.biases = np.zeros((output_size, 1))
```

האתחול הזה מייצר התפוצצות של הגרדאינטים ו-overflow בחישובים. ולכן לא השתמשנו באתחול זה.

3) אתחול גאוסני, מימשנו כך:

```
self.weights = np.random.randn(output_size, input_size) * 0.01
self.biases = np.zeros((output_size, 1))
```

האתחול הזה הוביל ללמידה, ובו השתמשנו בכל הריצות של הניסויים.

אופטימיזציה על פונקציות אקטיבזציה

לפני שקיבענו את פונקציית האקטיבזציה ביצענו מספר הרצות שונות כאשר בכל אחת מהריצות השתמשנו באקטיבזציה שונה.

1) Relu

2) Sigmoid

שימוש ב-Relu נתן תוצאות טובות יותר, ולכן השתמשנו בו.

אופטימיזציה על הקטנת ה-learning rate

לפני שקיבענו את שיטת הקטנת פרמטר הלמידה ביצענו מספר הרצות שונות כאשר הרצנו עם פרמטרים שונים בכל ריצה. הגענו למסקנה שהקטנה של ה-learning rate אחת ל-20 אפוקים מביא לתוצאות טובות.