

Ex4 – Report

מבנה הרשת:

- הרשת בנויה מ 8 שכבות – 3 שכבות קונבולוציה ו 5 שכבות fully connected.
- שכבה 1: שכבת קונבולוציה, 32 kernels כל אחד בגודל 9, בלי padding וגודל צעד 1. לאחר מכן ReLU, batchNorm, ו maxPooling בגודל 2X2 וצעד 2.
- שכבה 2: שכבת קונבולוציה, 64 kernels כל אחד בגודל 5, padding=2 וגודל צעד 1. לאחר מכן ReLU, batchNorm.
- שכבה 3: שכבת קונבולוציה, 100 kernels כל אחד בגודל 3, padding=1 וגודל צעד 1. לאחר מכן ReLU, batchNorm.
- שכבה 4: שכבת fully connected, מגודל 16,000 לגודל 8,000. לאחר מכן ReLU, batchNorm ו dropout בהסתברות 0.5.
- שכבה 5: שכבת fully connected, מגודל 8,000 לגודל 4,000. לאחר מכן ReLU, batchNorm ו dropout בהסתברות 0.5.
- שכבה 6: שכבת fully connected, מגודל 4,000 לגודל 2,000. לאחר מכן ReLU, batchNorm ו dropout בהסתברות 0.5.
- שכבה 7: שכבת fully connected, מגודל 2,000 לגודל 1,000. לאחר מכן ReLU, batchNorm ו dropout בהסתברות 0.5.
- שכבה 8: שכבת fully connected, מגודל 1,000 לגודל 30. לאחר מכן הפעלנו softmax.

ניסינו מספר רשתות עם כמות שכבות שונות (4 שכבות, 6 שכבות) ומשחק בין מספר שכבות הקונבולוציה למספר השכבות ה- fully connected. ניסינו גם לשנות את הערכים שבשכבות עצמן. בסופו של דבר ראינו שזה מבנה הרשת המוצלח ביותר. ניסינו לשלב RNN בחלק מהרשתות שיצרנו ולא ראינו שיפור.

בחירת היפר פרמטרים:

- Learning rate – בחרנו את מקדם הלמידה להיות 0.01 כי ראינו שבריצות אחרות עם מקדם למידה שונה התוצאות היו פחות טובות.
- Loss – ניסינו להשוות בין Cross entropy loss לבין Negative Log Loss, סוג ה Loss שבחרנו הוא Cross Entropy loss שכן הוא נתן תוצאות טובות יותר.
- כמות ה epochs – חיפשנו כמות epochs שתיתן אחוז loss נמוך עבור ה train set ועבור ה validation set, בנוסף לאחוזי דיוק גבוהים. ביצענו עבור הרשת מספר הרצות כדי לראות שהתוצאות עקביות ואכן הייתה עקביות. ניתן לראות בתמונה הבאה כי התוצאה הטובה ביותר התקבלה עבור ה epoch ה 8.

```
Epoch [1/15], Train loss: 0.0169, Validation loss: 1.4546, Accuracy : 58.25%
Epoch [2/15], Train loss: 0.0069, Validation loss: 0.6319, Accuracy : 88.64%
Epoch [3/15], Train loss: 0.0046, Validation loss: 0.6385, Accuracy : 81.01%
Epoch [4/15], Train loss: 0.0038, Validation loss: 0.4737, Accuracy : 86.80%
Epoch [5/15], Train loss: 0.0030, Validation loss: 0.4187, Accuracy : 88.39%
Epoch [6/15], Train loss: 0.0026, Validation loss: 0.4382, Accuracy : 88.32%
Epoch [7/15], Train loss: 0.0023, Validation loss: 0.4336, Accuracy : 88.58%
Epoch [8/15], Train loss: 0.0019, Validation loss: 0.3994, Accuracy : 89.03%
Epoch [9/15], Train loss: 0.0017, Validation loss: 0.4597, Accuracy : 88.58%
Epoch [10/15], Train loss: 0.0015, Validation loss: 0.4617, Accuracy : 88.82%
Epoch [11/15], Train loss: 0.0015, Validation loss: 0.4619, Accuracy : 89.10%
Epoch [12/15], Train loss: 0.0013, Validation loss: 0.4925, Accuracy : 89.10%
Epoch [13/15], Train loss: 0.0012, Validation loss: 0.4880, Accuracy : 89.45%
Epoch [14/15], Train loss: 0.0011, Validation loss: 0.5289, Accuracy : 88.72%
Epoch [15/15], Train loss: 0.0011, Validation loss: 0.4581, Accuracy : 90.07%
```

- גודל הפילטרים – ניסינו גדלים שונים של פילטרים בשביל למזער את הדאטא ובשביל להפוך את הפרמטרים מכלליים לספציפיים. לכן בכל פעם הקטנו את הפילטרים החל מגודל 9X9 ועד ל 3X3 כאשר הפילטרים הגדולים מופעלים על התמונות הגדולות והקטנים על התמונות הקטנות.
- Max Pooling – על מנת למזער את גודל המידע שמועבר ברשת ובנוסף להמשיך רק עם הפיצ'רים הבולטים ביותר בסבב הראשון השתמשנו ב max pooling בגודל 2 ובקפיצות של 2. כתוצאה מכך גודל המידע קטן בסבב הקונבולוציה הראשון פי 4.
- Optimization – ערכנו השוואות בין סוגים שונים של אופטימיזציות, ביניהן Adam, SGD, Adagrad. לאחר ההשוואות ניתן היה לראות שהאופטימיזציה שנתנה את התוצאות הטובות ביותר היא Adam.