מבוא ללמידה עמוקה – דו"ח מטלה 1

1. הסברים על פונקציות עזר

create_batches(x, y, batch_size) – מקבלת את הנתונים שעתידים להכנס לרשת כקלט (שהם כבר לאחר עיבוד מקדים), ומחלקת אותם לקבוצות בגודל batch_size, למעט קבוצה אחת כאשר הנתונים לא מתחלקים ללא שארית.

preprocess_data(x, y) – לכאן אנו מכניסים את הנתונים הגולמיים שקיבלנו מהפונקציה של Keras, על מנת לבצע עליהם עיבוד מקדים שכולל reshape למטריצות במימדים שהרשת מצפה לעבוד איתם, וחלוקת הפיקסלים ב-255 כדי שתהליך הלמידה ישתפר.

evaluate(x, y, parameters, use_batchnorm) – פונקציה שעושה את המעבר קדימה (forward propagation) כדי לחשב את ההפסד עבור x, y נתונים, אותו היא מחזירה. להבדיל מ-()Predict, פה לא מחשבים את הדיוק, אלא את ה-cost.

acc_metric(Y, probas) – פונקציית עזר לחישוב הדיוק ביחס לתוצאות ה-softmax במטריצת ה-probas. פונ' זו מקבלת כבר את הפלט של המודל, בניגוד ל-()Predict שמקבלת את הקלט ועושה forward propagation.

evaluate_validation(x_val, y_val, parameters, use_batchnorm, val_loss, params_container, min_epochs, curr_epoch)

זוהי פונקציית עזר שמטפלת בכל הנוגע לחישוב ה-validation loss לאחר כל training iteration, ומנהלת את ההיסטוריה של ערכי פונקציית ה-loss ואת המערך ששומר את הפרמטרים של הרשת הכי טובים עד כה (כלומר מטפלת בכל מה שקשור למנגנון ה-early stopping).

2. תוצאות הרצת המודל

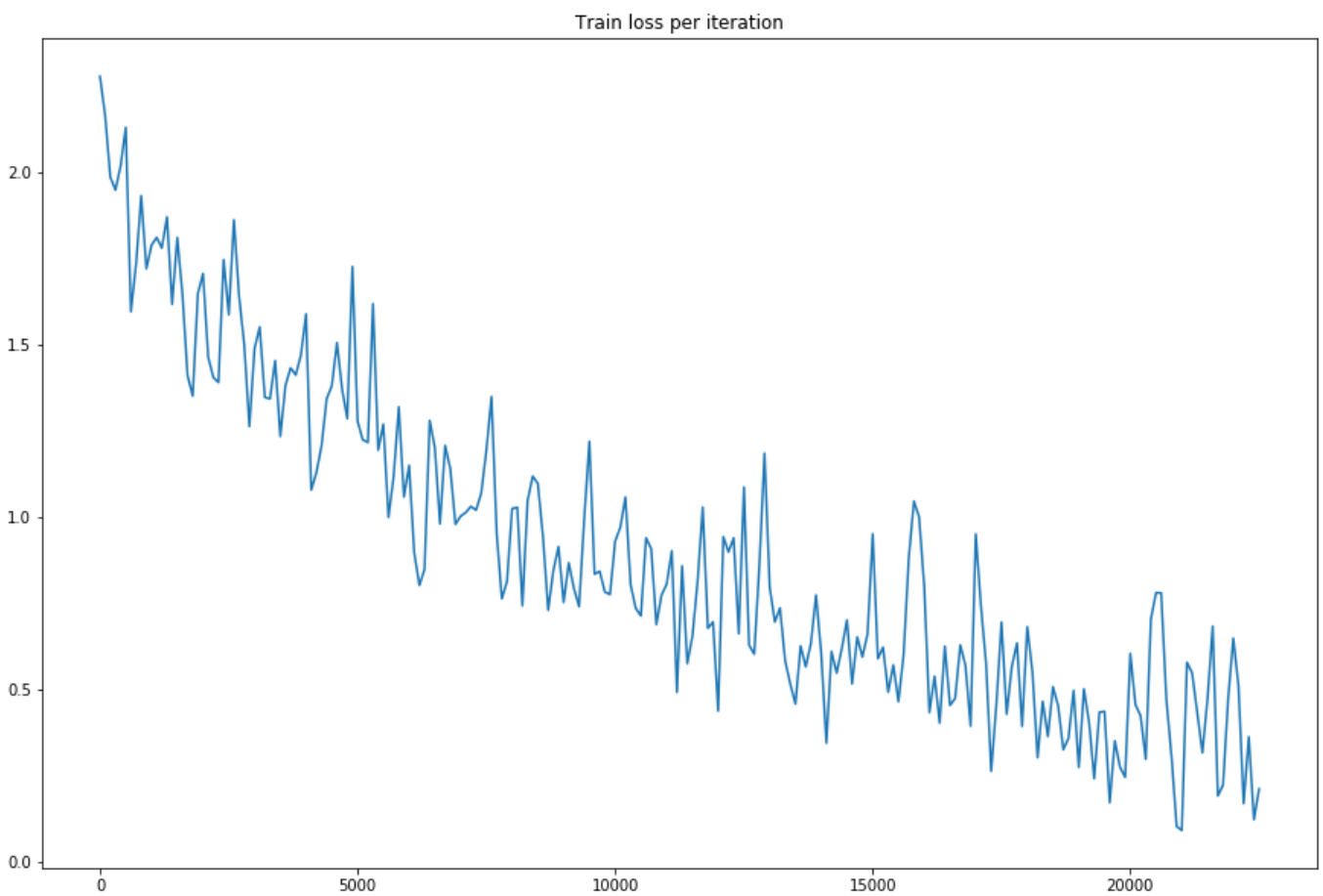
הרצה ראשונה עם ההיפר-פרמטרים הבאים:

```
use_batchnorm=False
learning_rate=0.009
early_stopping=100
batch_size=32
dropout_threshold=0.
min_epochs = 15
```

מהלך האימון, אחוזי הדיוק הסופיים עבור כל סט נתונים וגרף המתאר את ערכי פונקציית ההפסד בכל איטרציה:

```
Epoch #1 in: 43s | train_loss: 1.8853 - val_loss: 1.6729 | train_acc: 25.43 - val_acc: 31.08
Epoch #2 in: 44s | train_loss: 1.5505 - val_loss: 1.4773 | train_acc: 37.86 - val_acc: 45.87
Epoch #3 in: 41s | train_loss: 1.4023 - val_loss: 1.3546 | train_acc: 47.45 - val_acc: 52.29
Epoch #4 in: 46s | train_loss: 1.2411 - val_loss: 1.1722 | train_acc: 57.46 - val_acc: 59.59
Epoch #5 in: 44s | train_loss: 1.077 - val_loss: 1.0786 | train_acc: 62.7 - val_acc: 61.9
Epoch #6 in: 43s | train_loss: 0.9745 - val_loss: 0.9537 | train_acc: 66.8 - val_acc: 67.61
Epoch #7 in: 45s | train_loss: 0.8973 - val_loss: 0.89 | train_acc: 69.41 - val_acc: 70.59
Epoch #8 in: 45s | train_loss: 0.8239 - val_loss: 0.8127 | train_acc: 72.23 - val_acc: 73.6
Epoch #9 in: 40s | train_loss: 0.749 - val_loss: 0.7386 | train_acc: 76.21 - val_acc: 77.65
Epoch #10 in: 39s | train_loss: 0.6799 - val_loss: 0.6711 | train_acc: 79.27 - val_acc: 79.65
Epoch #11 in: 39s | train_loss: 0.6224 - val_loss: 0.6192 | train_acc: 81.69 - val_acc: 85.21
Epoch #12 in: 39s | train_loss: 0.5461 - val_loss: 0.5257 | train_acc: 87.6 - val_acc: 88.58
Epoch #13 in: 41s | train_loss: 0.4651 - val_loss: 0.4924 | train_acc: 89.77 - val_acc: 88.8
Epoch #14 in: 46s | train_loss: 0.4142 - val_loss: 0.4395 | train_acc: 90.6 - val_acc: 90.36
Epoch #15 in: 41s | train_loss: 0.3788 - val_loss: 0.3933 | train_acc: 91.23 - val_acc: 90.82
Epoch #16 in: 3s | train_loss: 0.3768 - val_loss: 0.3943 | train_acc: 91.51 - val_acc: 91.2
Early stopping...
Training process has completed after 22602 iterations and 16 epochs.
```

```
Train Accuracy: 92.11% - Validation Accuracy: 91.2% - Test Accuracy: 91.4%
```

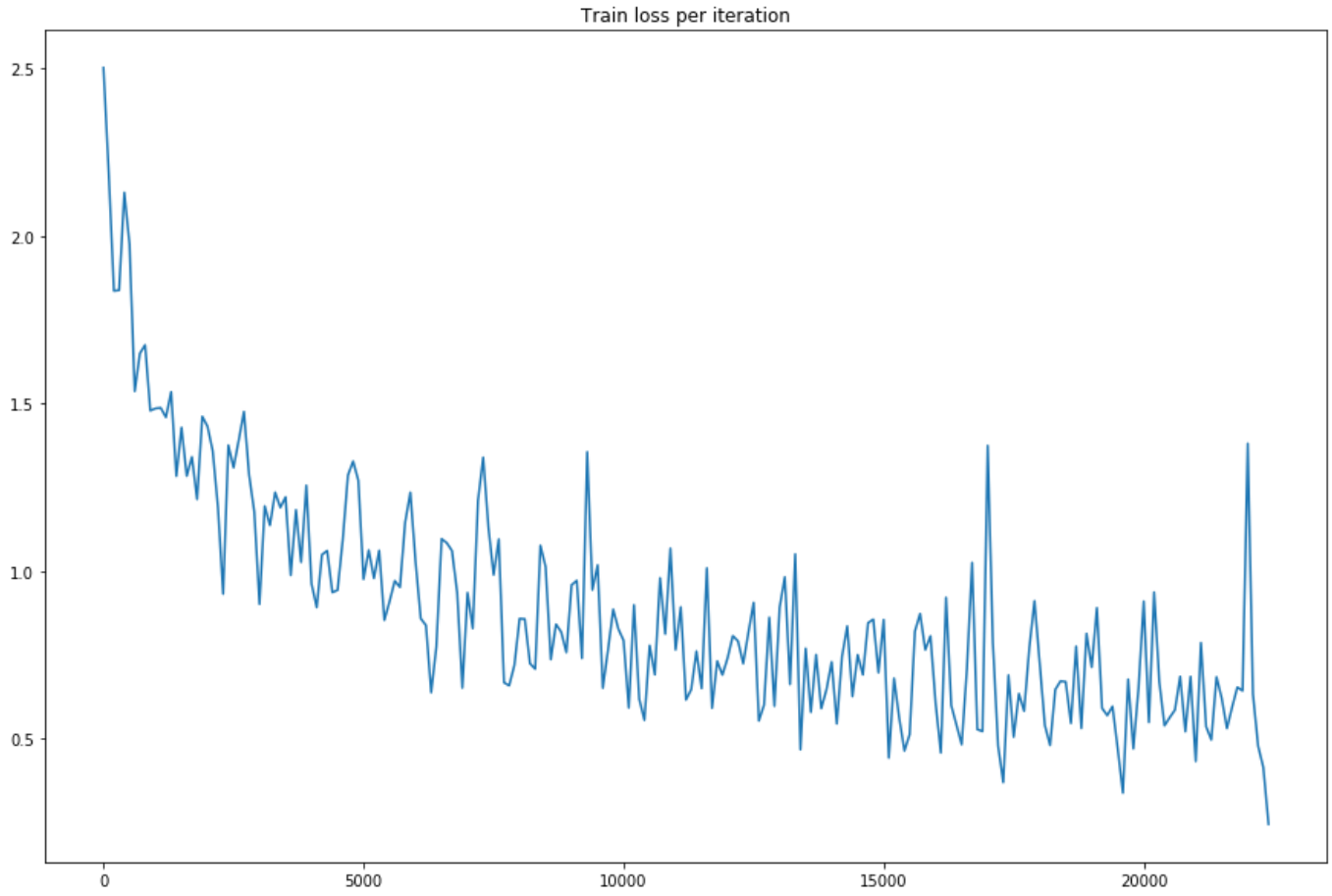


הרצה שניה עם אותם היפר-פרמטרים מההרצה הקודמת, רק עם batch normalization בכל שכבה למעט האחרונה. נציין שכפי שנאמר בדיון עם גלעד, לא מומש ה-back propagation המלא, כלומר עם למידה של הפרמטרים של ה-batchnorm למען פשטות העבודה, לכן ביצועי המודל הסופי לא אופטימליים.

```
Epoch #1 in: 41s | train_loss: 1.7804 - val_loss: 1.4401 | train_acc: 37.95 - val_acc: 48.8
Epoch #2 in: 43s | train_loss: 1.3451 - val_loss: 1.1853 | train_acc: 51.5 - val_acc: 58.61
Epoch #3 in: 43s | train_loss: 1.1688 - val_loss: 1.0233 | train_acc: 59.47 - val_acc: 67.03
Epoch #4 in: 44s | train_loss: 1.0475 - val_loss: 0.9107 | train_acc: 65.08 - val_acc: 71.92
Epoch #5 in: 46s | train_loss: 0.9524 - val_loss: 0.8163 | train_acc: 70.08 - val_acc: 75.68
Epoch #6 in: 47s | train_loss: 0.8735 - val_loss: 0.7398 | train_acc: 73.18 - val_acc: 78.63
Epoch #7 in: 47s | train_loss: 0.8106 - val_loss: 0.6873 | train_acc: 75.75 - val_acc: 80.48
Epoch #8 in: 46s | train_loss: 0.7626 - val_loss: 0.6335 | train_acc: 77.63 - val_acc: 82.76
Epoch #9 in: 47s | train_loss: 0.7295 - val_loss: 0.6103 | train_acc: 78.65 - val_acc: 83.29
Epoch #10 in: 62s | train_loss: 0.702 - val_loss: 0.5829 | train_acc: 79.49 - val_acc: 84.32
Epoch #11 in: 49s | train_loss: 0.6942 - val_loss: 0.5667 | train_acc: 79.89 - val_acc: 84.8
Epoch #12 in: 46s | train_loss: 0.6761 - val_loss: 0.5493 | train_acc: 80.16 - val_acc: 85.28
Epoch #13 in: 46s | train_loss: 0.6576 - val_loss: 0.5357 | train_acc: 80.7 - val_acc: 85.45
Epoch #14 in: 46s | train_loss: 0.6503 - val_loss: 0.5226 | train_acc: 80.86 - val_acc: 85.88
Epoch #15 in: 47s | train_loss: 0.6356 - val_loss: 0.516 | train_acc: 81.34 - val_acc: 86.07
Epoch #16 in: 1s | train_loss: 0.6363 - val_loss: 0.5153 | train_acc: 80.29 - val_acc: 85.98
Early stopping...
Training process has completed after 22513 iterations and 16 epochs.
```

ניתן לראות שזמן הריצה של כל epoch הוא מעט גדול יותר מההרצה הקודמת, וזה הגיוני בהתחשב שהתווספה פה פונקציה בכל שכבה לגרף החישוב של הרשת. ניכר גם שהרשת עם BN מתכנסת יותר מהר, לדוגמא כבר ב-epoch הראשון הרשת הגיעה ל-48.8% ו-38% דיוק על סט הולדיציה והאימון בהתאמה, לעומת רק 31% ו-25% ברשת הראשונה ללא BN. עם זאת, כנראה בגלל שאנחנו לא מאמנים את הפרמטרים של BN, כלומר גמא ובטא, ניתן לראות שהתוצאה הסופית ב-epoch ה-16 היא לטובת הרשת ללא ה-BN, כי כנראה שלאורך זמן ה-BN לא מועילה במיוחד. תנאי ה-early stopping עבד פחות או יותר לאחר אותו מספר של איטרציות בשתי הרשתות.

```
Train Accuracy: 86.16% - Validation Accuracy: 85.98% - Test Accuracy: 85.76%
```



3. בונוס Dropout:

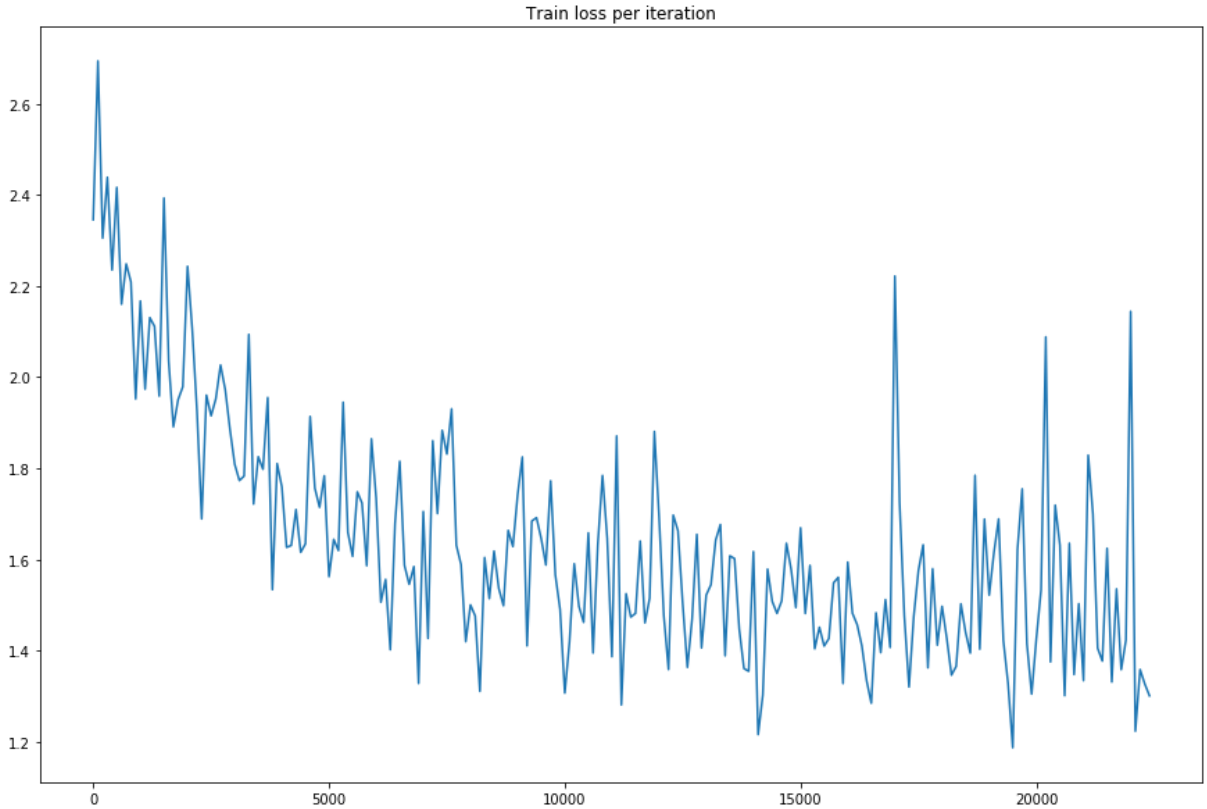
מימשנו את הרעיון בכך שהוספנו בפונקציה של linear_activation_forward לאחר האקטיבציה ולאחר ה-batchnorm (אם יש), את קטע הקוד הבא:

```
scale = 1./(1.-dropout_threshold)
dropout_matrix = np.random.binomial(1, 1.-dropout_threshold, size=A.shape)
A *= dropout_matrix
A *= scale
linear_cache = linear_cache + (dropout_matrix, ) # appends to tuple
```

ב-Dropout אנחנו רוצים "לבטל" אחוז רנדומלי מסוים מהנוירונים (dropout_threshold), ולכן ניצור מטריצה של 1 ו-0 לפי התפלגות בינומית שתהווה לנו מעין mask למטריצת A לאחר האקטיבציה. בנוסף, מכיוון שביטלנו חלק מהנוירונים, נרצה לבצע scale up לשאר הנוירונים כך שהשפעתם תעלה (זהו ה-scale). לאחר התהליך, נרצה לשמור את מטריצת ה-mask כדי לדעת איזה נוירונים "התאפסו" בהתליך ה-forward, ובכך לדעת לא לעדכן את המשקולות שלהם ב-backprop, לכן המטריצה התווספה ל-linear_cache של הרשת.

להלן תוצאות האימון על רשת שכוללת Dropout של 0.3, בנוסף ל-batch normalization:

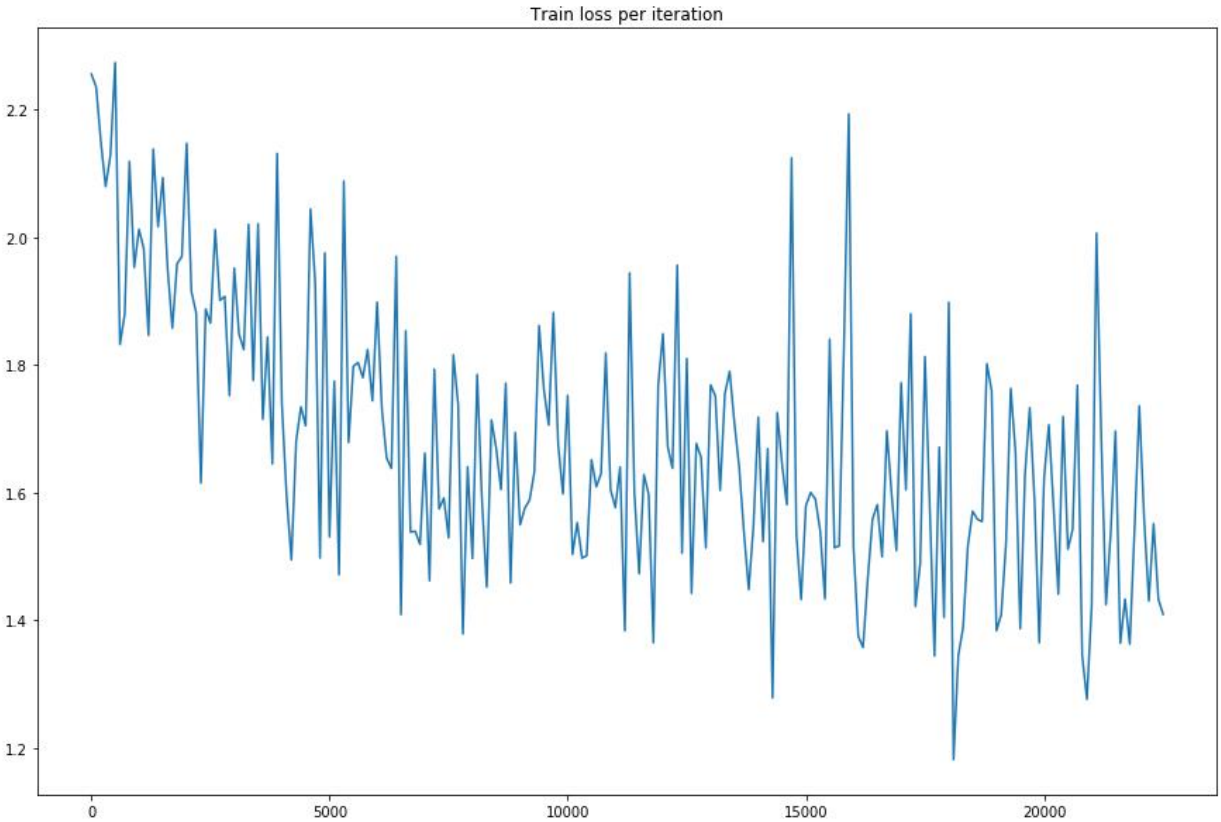
```
Epoch #1 in: 42s | train_loss: 2.27 - val_loss: 1.83 | train_acc: 19.73 - val_acc: 37.53
Epoch #2 in: 41s | train_loss: 1.9638 - val_loss: 1.6319 | train_acc: 27.75 - val_acc: 42.67
Epoch #3 in: 41s | train_loss: 1.8201 - val_loss: 1.4862 | train_acc: 31.13 - val_acc: 46.48
Epoch #4 in: 42s | train_loss: 1.7125 - val_loss: 1.388 | train_acc: 32.64 - val_acc: 46.78
Epoch #5 in: 42s | train_loss: 1.634 - val_loss: 1.3395 | train_acc: 33.83 - val_acc: 45.17
Epoch #6 in: 43s | train_loss: 1.5853 - val_loss: 1.309 | train_acc: 34.31 - val_acc: 47.56
Epoch #7 in: 45s | train_loss: 1.5574 - val_loss: 1.293 | train_acc: 34.51 - val_acc: 44.85
Epoch #8 in: 46s | train_loss: 1.5349 - val_loss: 1.2637 | train_acc: 35.15 - val_acc: 46.12
Epoch #9 in: 46s | train_loss: 1.5153 - val_loss: 1.2513 | train_acc: 36.04 - val_acc: 46.36
Epoch #10 in: 45s | train_loss: 1.5011 - val_loss: 1.2242 | train_acc: 36.32 - val_acc: 48.96
Epoch #11 in: 45s | train_loss: 1.5049 - val_loss: 1.2189 | train_acc: 36.48 - val_acc: 48.82
Epoch #12 in: 46s | train_loss: 1.4898 - val_loss: 1.1992 | train_acc: 37.09 - val_acc: 53.19
Epoch #13 in: 45s | train_loss: 1.4779 - val_loss: 1.1839 | train_acc: 38.11 - val_acc: 55.36
Epoch #14 in: 46s | train_loss: 1.4735 - val_loss: 1.1661 | train_acc: 37.94 - val_acc: 57.04
Epoch #15 in: 46s | train_loss: 1.4577 - val_loss: 1.1482 | train_acc: 39.21 - val_acc: 59.21
Epoch #16 in: 0s | train_loss: 1.4577 - val_loss: 1.1484 | train_acc: 43.75 - val_acc: 58.86
Early stopping...
Training process has completed after 22501 iterations and 16 epochs.

Train Accuracy: 60.81999999999999% - Validation Accuracy: 58.86% - Test Accuracy: 60.22%
```



ומכיוון שהתוצאות לא גבוהות במיוחד, ניסינו להריץ את הרשת עם Dropout ובלי BN. להלן התוצאות המעט יותר טובות:

```
Epoch #1 in: 35s | train_loss: 2.0954 - val_loss: 1.807 | train_acc: 20.56 - val_acc: 45.18
Epoch #2 in: 38s | train_loss: 1.8895 - val_loss: 1.5671 | train_acc: 30.0 - val_acc: 56.02
Epoch #3 in: 38s | train_loss: 1.7918 - val_loss: 1.4291 | train_acc: 31.66 - val_acc: 62.22
Epoch #4 in: 38s | train_loss: 1.749 - val_loss: 1.3468 | train_acc: 32.46 - val_acc: 67.31
Epoch #5 in: 38s | train_loss: 1.706 - val_loss: 1.2854 | train_acc: 33.39 - val_acc: 68.77
Epoch #6 in: 38s | train_loss: 1.6731 - val_loss: 1.2446 | train_acc: 35.52 - val_acc: 69.12
Epoch #7 in: 38s | train_loss: 1.6521 - val_loss: 1.2029 | train_acc: 37.5 - val_acc: 69.58
Epoch #8 in: 38s | train_loss: 1.6305 - val_loss: 1.1871 | train_acc: 38.79 - val_acc: 69.17
Epoch #9 in: 38s | train_loss: 1.6116 - val_loss: 1.1611 | train_acc: 39.48 - val_acc: 69.66
Epoch #10 in: 39s | train_loss: 1.6085 - val_loss: 1.159 | train_acc: 39.5 - val_acc: 69.72
Epoch #11 in: 38s | train_loss: 1.5974 - val_loss: 1.1375 | train_acc: 39.96 - val_acc: 68.37
Epoch #12 in: 38s | train_loss: 1.5882 - val_loss: 1.1154 | train_acc: 40.27 - val_acc: 70.23
Epoch #13 in: 38s | train_loss: 1.5806 - val_loss: 1.1132 | train_acc: 40.69 - val_acc: 68.92
Epoch #14 in: 42s | train_loss: 1.5718 - val_loss: 1.0945 | train_acc: 41.05 - val_acc: 69.85
Epoch #15 in: 42s | train_loss: 1.561 - val_loss: 1.0845 | train_acc: 41.03 - val_acc: 71.31
Epoch #16 in: 3s | train_loss: 1.5616 - val_loss: 1.0943 | train_acc: 39.24 - val_acc: 71.07
Early stopping...
Training process has completed after 22604 iterations and 16 epochs.
```

```
Train Accuracy: 71.81% - Validation Accuracy: 71.07% - Test Accuracy: 71.03%
```



התוצאות שכוללות Dropout בשני המקרים אינן טובות במיוחד בהשוואה לרשתות שאימנו ללא Dropout, כנראה מהסיבה שזו דרך להתמודדות עם overfitting וכפי שניתן היה לראות ברשתות הקודמות לא היה overfitting, ואולי זה רק פגע בביצועים. יתכן שאחוז אחר של dropout או שילוב אחר של היפר פרמטרים היה טוב יותר לאימון הרשת, אך העדפנו לשמור עליהם אחידים בכל האימונים.