

- how large were the training and test sets,
- did your network succeed in distinguishing the two languages (it should)?
- how long did it take (both wall-clock time (i.e., number of seconds), and number of iterations)? *
- did it succeed only on the train and not on the test?
- what you did in order to make it work, etc.

תחילה נציג את הפרמטרים שבחרנו לתרגיל :

- גודל שכבה נסתרת – 35
- גודל שכבת האינפוט – 30
- אוצר המילים שלנו בתרגיל הוא : ['a', 'b', 'c', 'd', '1', '2', '3', '4', '5', '6', '7', '8', '9']
- גודל שכבת ה-output הנו 2 (1 או 0 – מייצג good 0 מייצג רע)
- מספר האפוקים – 4
- גודל סט האימון שבחרתי הינו : 4000
- גודל הטסט שבחרתי הינו : 200
- השתמשתי בספריית dynet וב-adamTrainer כיוון שזה היה הרבה יותר מהיר מ-SGD לאחר שניסיון ובדקנו.

כל משפט בסט אימון פירקנו לתווים שהוא מורכב מהן, ולכל תו נתנו אינדקס מתאים, המייצג בעצם embedding vector.

בכלל שלב כל מילה בתורה נכנסה לתוך הרשת ה-LSTM ולבסוף הוקטור שמתקבל משכבת האאוטפוט הוכנס לתוך MLP, אשר החזיר לנו וקטור עם שני ערכים – 1 מציין good ו-0 מציין bad

לסיכום ניתן לומר שהרשת הצליחה להפריד בין הדוגמאות הטובות לרעות, הסיבה לכך שהצלחנו היא שעשינו שימוש ברשת מסוג RNN שמתחשבת במה שקרה בעבר – הרשת מקבלת את הקלט בשלבים, תו אחר תו, ובכל שלב היא מקבלת גם את הפלט של השכבה הקודמת, ככה שבעצם הרשת מקבלת מידע מהעבר ומסתמכת עליו, לכן אם היה רצף של b בעבר הרשת תבחין בכך כיוון שזה יבוא לידי ביטוי במידע שהיא תקבל מהשכבות הקודמות.

- סט האימון הגיע לדיוק של 100% כבר באיטרציה השנייה - לאחר 73 שניות
- סט האימון הגיע ל-loss בשווי 0 באיטרציה החמישית – לאחר 181 שניות
- סט הטסט הגיע ל-100% דיוק באיטרציה הראשונה – לאחר 37 שניות
- סט הטסט הגיע ל-loss בשווי 0 באיטרציה הרביעית – לאחר 145 שניות