

מגישים: אור שחר, 209493709, יהונתן שאקי 204920367.

המשימות שמצאנו וקשות לרשת:

- מספרים ראשוניים. הגרלנו מספרים בטווח מסוים והצגנו אותם בתצוגה בינארית. המילה אמורה להתקבל אם היא מספר ראשוני ואמורה להידחות אם אינה. על מנת לעשות את המשימה קשה, בחרנו לצד השלילי רק מספרים אי זוגיים.
הרשת בכלל לא הצליחה במשימה, כלומר דיוק של 50% על הטסט. כשהשתמשנו ב-dataset גדול (10000 דוגמאות בסך הכל) היא לא הצליחה גם על ה-train. עבור datasets קטנים היא הצליחה על ה-train ולא על הטסט. זה היה עקבי לאורך הרבה איטרציות אימון ומספר ארכיטקטורות שניסו. ככל הנראה המשימה הייתה קשה בגלל הפעולות האריתמטיות שנדרשות כדי לקיים אותה; לא נראה שרשתות נוירונים סטנדרטיות מצוידות במה שנדרש על מנת לבדוק חילוקים, מודולו, שלא לדבר על ביצוע איטרציות בשביל לבדוק האם קיים מחלק.
- בדיקת פולינדרום. הדוגמאות החיוביות היו פולינדרומים, והדוגמאות השליליות היו "פולינדרומים משובשים" עבורם שינינו תו אחד כדי להרוס את הפולינדרום. הרשת הצליחה לא רע על אורכים קטנים (10-20), אבל ככל שהאורכים גדלו היא הצליחה פחות. עבור אורכים עד 1000 הרשת הצליחה טוב על האימון אבל די גרוע על הטסט, או די גרוע על האימון אם היו מספיק דוגמאות. בכל מקרה, על הטסט הרשת לא הצליחה להגיע למשהו משמעותי מעל 0.5.
(מעניין לציין שהוא כן הצליח להבדיל בין פולינדרומים למחרוזות אקראיות לגמרי)
- בדיקת תו התחלה ותו סיום זהים במחרוזת. יצרנו רצף תווים ארוך (1000) רנדומלי (מעל {0,1}), ועבור דוגמאות שנמצאות ב-data הוספנו את אותו תו בשני הצדדים, אחרת תווים שונים בהכרח. גם כאן, עבור כמות גדולה של דוגמאות, הרשת לא הצליחה מעבר לניחוש אקראי. זה ככל הנראה קרה בגלל שקשה לרשת להבין את החוקיות – הגרדיאנט צריך לפעפע 1000 צעדים אחורה בשביל זה. יתכן שיש כאן סוג של vanishing gradient (שאמור דווקא לא לקרות יותר מידי ב-LSTM) או שסתם קשה לגרדיאנט דסנט "לחבר" בין הסוף להתחלה באופן הזה.

את כל היוצרים לדוגמאות ואת הדוגמאות ניתן לראות בתקיה data/part_two.