**סיכום פגישות חן וברק**

עד התאריך ה-5.12:

כתבנו קוד ראשוני שמייצר לוח ומוחק ממנו מספרים כך שיהיה פתרון יחיד לסודוקו. הקובץ שומר שומר את הלוח המקורי ואת הלוח עם המספרים המחוקים בקובץ py.(זה מציג בג'יבריש כאשר עושים בקובץ py אולי נשנה ונעזר ב-numpy) נעזרנו באתר:

<https://www.101computing.net/sudoku-generator-algorithm/>

קראנו לקובץ: createGrid

הצלחנו לשנות את לולאת ה-while כך שיקח מהר יותר (שלא יחפש באופן אקראי מספר מהלוח).

ניסינו לעשות את הקוד כללי ללוח לא בהכרח 9 על 9 אבל עדין לא הצלחנו.

כדאי אולי לשנות את הקוד שיהיה יותר יפה וגם שיעבוד על לוחות שהם לא 9 על 9.

בנוסף, כתבנו קוד שפותר את הלוח באמצעות backtracking . נעזרנו באתר:

<https://www.geeksforgeeks.org/sudoku-backtracking-7/>

קראנו לקובץ: solveBacktracking

כתבנו דוח אפיון שבו התייחסנו לסקר הספרות שאותו נבצע.

<https://www.tandfonline.com/doi/full/10.1080/10586458.2013.870056>

McGuire, Gary, Bastian Tugemann, and Gilles Civario. "There is no 16-clue Sudoku: solving the Sudoku minimum number of clues problem via hitting set enumeration." Experimental Mathematics 23.2 (2014): 190-217.

מאמר שמסיר שיש מינימום 17 מספרים בסודוקו ואומר כמה אפשרויות יש ללוחות סודוקו.

התחלנו לכתוב סקר ספרות ודוח אפיון:

אתרים באינטרנט שמצאנו:

<https://www.tandfonline.com/doi/full/10.1080/10586458.2013.870056>

McGuire, Gary, Bastian Tugemann, and Gilles Civario. "There is no 16-clue Sudoku: solving the Sudoku minimum number of clues problem via hitting set enumeration." Experimental Mathematics 23.2 (2014): 190-217.‏

<http://micsymposium.org/mics2018/proceedings/MICS_2018_paper_17.pdf>

<https://www.mathworks.com/company/newsletters/articles/solving-sudoku-with-matlab.html>

<https://www.mathworks.com/help/optim/examples/solve-sudoku-puzzles-via-integer-programming.html>

<https://www.kaggle.com/bryanpark/sudoku>

<https://ieeexplore.ieee.org/abstract/document/7333884>

<https://ieeexplore.ieee.org/abstract/document/7007986>

קוד מחולק לפי רמות שהגיע ל-86% דיוק (כאן יש חלוקה לרמות קושי):

<https://github.com/Kyubyong/sudoku>

פתרון שהגיע ל-99.7% ללא קונובלוציה רק fully connected:

<https://www.kaggle.com/dithyrambe/neural-nets-as-sudoku-solvers>

פתרון שהגיע ל-99 אחוז:

<https://towardsdatascience.com/solving-sudoku-with-convolution-neural-network-keras-655ba4be3b11>

<https://github.com/shivaverma/Sudoku-Solver-CNN/blob/master/sudoku.ipynb?source=post_page-----655ba4be3b11---------------------->

10.12.19

נפגשנו עם רון ותום וסיפרנו להם את הפתרונות שמצאנו.

דברים מרכזיים שעלו מהפגישה:

* להתחיל לכתוב קוד.
* לבדוק אם ה-loss הוא עבור המשבצת החסרה לבד בכל פעם או יותר.
* לבדוק את התוצאות של הפתרון שראינו אולי על לוחות קשים יותר (יותר מ-50 משבצות חסרות)
* לנסות לעשות קונבולוציה במקום הפתרון שהגיע ל-99.7% עם fully connected בלבד.
* לעשות One hot עם 9X9X9 או 9X9X10 (ספרה נוספת עבור האפס)
* להתחיל מהפתרון שראינו באינטרנט
* לא מנרמלים one hot
* לא בהכרח חייבים להשתמש במליון לוחות מהדאטה. להשתמש במספר לוחות שיביא לנו תוצאה טובה.
* אפשר להשתמש ב-keras או ב-pytorch או ב-tensorflow (מה שבא לנו).
* כשנקבל רשת שלמדה - דרך כלשהי להבין מה הרשת למדה ולנתח את התוצאות:

אפשר לבדוק איזה מספר הוא בעל הסתברות מקסימלית לצאת. כלומר אם ההסתברות הכוללת של 4 למשל יותר גבוהה מההסתברות הכוללת של 9 עבור קלט מסוים (עם מעט 4 למשל).

12.12.19

סיימנו לכתוב דוח אפיון ושלחנו אותו. הכנסנו 2 אתרים מהאתרים שמצאנו והכנסנו מאמר אחד לסקר ספרות.