אלעד בעל צדקה 312531973
חן לארי 209192798

# דו"ח ביולוגיה חישובית

## הוראות הרצה:

לתרגיל מצורף קובץ requirements.txt עם כל הספריות שיש לייבא לתרגיל. כדי להתקין אותם יש להריץ:

```
pip install -r requirements.txt
```

מצורפים קבצי exe שאחראיים לאימון ובחינת הרשת.

הקובץ buildnet אחראי ללמידת המידע.
הוא מצפה לקבל שני ארגומנטים: שם הקובץ עם הנתונים ללמידה, ושם הקובץ שאליו נכתוב את נתוני הרשת.
כדי להפעילו עבור nn0 יש להריץ:

```
./buildnet0 nn0.txt wnet0.txt
```

כדי להפעילו עבור nn1 יש להריץ:

```
./buildnet1 nn1.txt wnet1.txt
```

בנוסף מצורף הקובץ runnet שאחראי למבחן הרשת.
הוא מצפה לקבל שני ארגומנטים: שם הקובץ עם נתוני הרשת, ושם הקובץ עם הנתונים למבחן.
כדי להפעילו עבור nn0 יש להריץ:

```
./runnet0 wnet0.txt test0.txt
```

כדי להפעילו עבור nn1 יש להריץ:

```
./runnet1 wnet1.txt test1.txt
```

## ייצוג המידע:

אנחנו קיבלנו את המידע בתור string באורך 16 תווים בינאריים.
אנחנו המרנו כל מחרוזת בעזרת one hot encoder למערך של 16 תווים מסוג int 0/1.

```
def one_hot_encode(string):
    string_array = list(string)
    string_array_len = len(string_array)
    input_array = []
    for i in range(string_array_len):
        input_array.append(int(string_array[i]))
    return input_array
```

## מבנה הרשת:

ברשת ה- neural network שבנינו יש שכבה אחת בגודל 8.

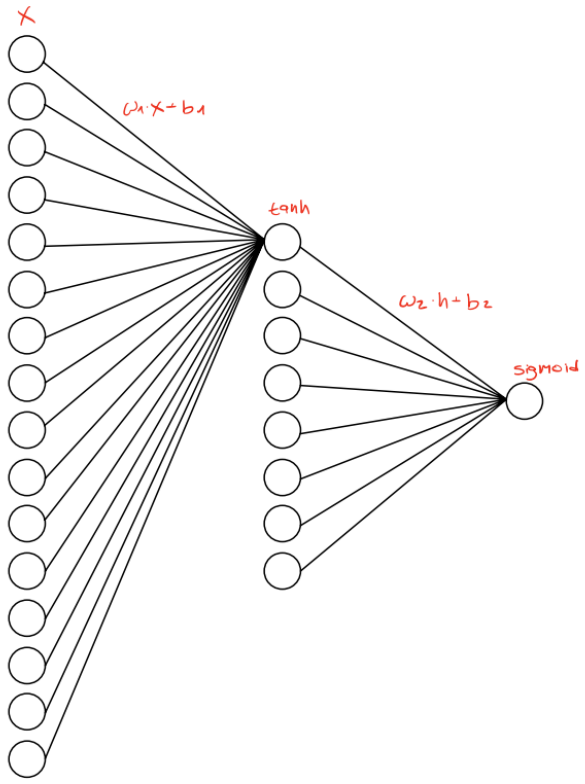
שכבת ה-input מקבלת מערך x בגודל 16.
שכבת ה-hidden בגודל 8 משתמשת בפונקציית האקטיבציה tanh.
שכבת ה-output בגודל 1 משתמשת בפונקציית sigmoid על מנת לנבא את התיוג.
לבסוף כדי לחשב את התיוג אנו בודקים אם התוצאה ב-output גדולה מ-0.5 ואז התיוג הוא 1 אחרת התיוג הוא 0.

אלעד בעל צדקה 312531973
חן לארי 209192798

Input x

$$h1 = \tanh(w1 \cdot x + b1)$$
$$output = sigmoid(w2 \cdot h1 + b2)$$
$$y = output > 0.5$$

<u>ניתן לראות זאת בציור:</u>




הערה: הרשת שבנינו היא Fully Connected, כלומר כל קודקוד מחובר לכל קודקוד בשכבה הבאה, למרות שבציור מתואר רק קשר בין קודקוד יחיד לקודקוד יחיד. לא ציירנו את כל הקשרים לצורך ההבנה של הציור.

<u>ובקוד:</u>

```
x = input.reshape(1, input.shape[0])
z1 = np.dot(x, self.w1) + self.b1
h1 = self._tanh(z1)
z2 = np.dot(h1, self.w2) + self.b2
h2 = self._sigmoid(z2)
y_hat = (h2 > 0.5).astype(int)
```

## מהי פונקציית ההערכה?

פונקציית ה-fitness שלנו מחשבת הצלחה לפי ממוצע הצלחות החיזויים על המידע.
הסבר:
אם חזינו 1 והתיוג האמיתי הוא 1, או שחזינו 0 והתיוג האמיתי הוא 0 – זו הצלחה.
אם חזינו 1 והתיוג האמיתי הוא 0, או שחזינו 0 והתיוג האמיתי הוא 1 – זו טעות.
פונקציית ה-fitness סופרת כמה הצלחות קיבלנו בהעברת המידע (forward) ברשת ומחשבת ממוצע לפי כמות הדוגמאות.

אלעד בעל צדקה 312531973
חן לארי 209192798

```
def fitness(self, network):
    predictions = network.forward(self.inputs)
    accuracy = np.mean(predictions == self.labels)
    return accuracy
```

## פעולת ה- crossover בין הפתרונות:

פעולת ה cross-over שלנו מחשבת ממוצע בין המשקולות וה-bias של 2 ההורים.

```
def crossover(self, parent1, parent2):
    child = NeuralNetwork(input_size=parent1.input_size, hidden_size=parent1.hidden_size)
    # calculate weights and biases for child
    w1 = (parent1.w1 + parent2.w1) / 2
    w2 = (parent1.w2 + parent2.w2) / 2
    b1 = (parent1.b1 + parent2.b1) / 2
    b2 = (parent1.b2 + parent2.b2) / 2
    # change for child
    child.w1 = w1
    child.w2= w2
    child.b1 = b1
    child.b2 = b2
    return child
```

## כיצד מומשו מוטציות?

באלגוריתם שלנו כל האוכלוסייה עוברת מוטציה.
הגרלנו מספרים בין 1- ל-1 והוספנו אותם למשקולות אחרי הכפלה ב mutation rate.
כלומר לכל רשת באוכלוסייה מתבצעת תזוזה של עד mutation rate במשקולות שלה.

```
mutated_network = copy.deepcopy(next_generation[i])
mutated_network.w1 += np.random.randn(*mutated_network.w1.shape) * mutation_rate
mutated_network.w2 += np.random.randn(*mutated_network.w2.shape) * mutation_rate
mutated_network.b1 += np.random.randn(*mutated_network.b1.shape) * mutation_rate
mutated_network.b2 += np.random.randn(*mutated_network.b2.shape) * mutation_rate
next_generation[i] = mutated_network
```

## האם/איך התייחסתם לבעיית ההתכנסות המוקדמת?

כדי להימנע (כמה שאפשר) מהתכנסות מוקדמת נקטנו כמה צעדים:

1. הגדלנו את כמות האוכלוסייה.
2. אפשרנו מוטציות בכל האוכלוסייה.

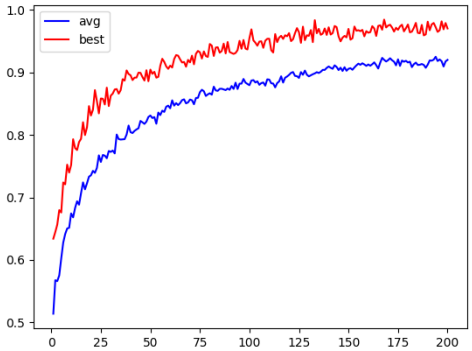
## ביצועי התוכנית:

חילקנו את המידע ל-80% קבוצת למידה ו-20% קבוצת מבחן.
אחרי שהאלגוריתם למד ומצא רשת טובה ביותר עבור קבוצת הלמידה, בחנו את הרשת בעזרת קבוצת המבחן.

עבור nn0.txt:

אלעד בעל צדקה 312531973
חן לארי 209192798

את המציאה של הרשת עבור קבוצת הלמידה ניתן לראות בגרף

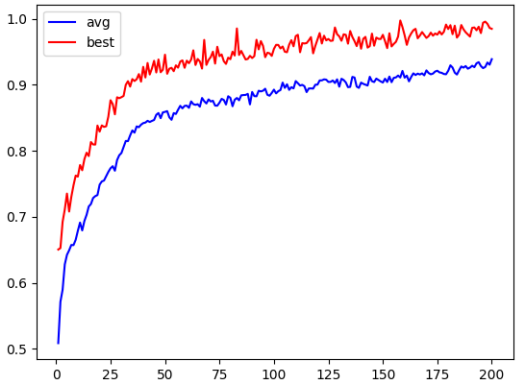



שמתאר את אחוז הדיוק של הרשת על קבוצת הלמידה לפי מספר איטרציות.
ניתן לראות שהאלגוריתם מוצא במהירות רשת שמגיעה ליותר מ-90% דיוק על קבוצת הלמידה, לאחר 100 איטרציות מוצא רשת עם דיוק של כ-95%, אך כאשר מגיע לאחוזים גבוהים עולה בצורה איטית.

לבסוף אחוז הדיוק על קבוצת המבחן הוא כ-98%.

עבור nn1.txt:

את המציאה של הרשת עבור קבוצת הלמידה ניתן לראות בגרף




שמתאר גם הוא את אחוז הדיוק על קבוצת הלמידה לפי מספר איטרציות.
גם כאן ניתן לראות שהדיוק עולה מהר עד להגעה ל-100 איטרציות, מגיע לדיוק של כ-96%, ואז קצת יותר קשה לו לעלות בדיוק.

לבסוף אחוז הדיוק על קבוצת המבחן הוא כ-99%.

## החוקיות מאחורי שתי התבניות:

עבור nn0.txt:

לדעתנו, החוקיות היא שבמחרוזת יופיע התו 1 לפחות 8 פעמים (כולל).

עבור nn1.txt:

לדעתנו, החוקיות היא שבמחרוזת יופיע התו 1 לכל היותר 8 פעמים (לא כולל).