

הרשת הניורונית שמימשתי מורכבת משכבת קלט, שכבת hidden layer ושכבת פלט. על שכבת hidden layer הפעלתי פונקציית sigmoid ועל שכבת הפלט הפעלתי את softmax כפי שצוין בתרגיל.

האתחלתי את **הפרמטרים** באופן הבא:

```
W1 = np.random.rand(100, 784) - 0.5
b1 = np.zeros([100, 1], dtype=float)
W2 = np.random.rand(10, 100) - 0.5
b2 = np.zeros([10, 1], dtype=float)
```

ערכי המשקל W1 וW2 מאותחלים בצורה אקראית ועשיתי 0.5 - כדי שיהיו גם משקלים שליליים ולא רק חיוביים, כלומר משקלים נמצאים בטווח בין -0.5 ל-0.5. **ערכי הbias** האתחלתי באפסים.

לגבי **כמות הניורונים** – ניסיתי כמה אופציות, התחלתי מ-10 שגם היה עובד אבל היה לומד לאט מידי, הייתי צריכה בערך 400 epochs כדי להגיע לאחוזים גבוהים (מעל 90%), ואז הגדלתי את כמות הניורונים (בדוגמא זה 100 ניורונים) ורשת התחיל ללמוד מאוד מהר, הספיק לי **20 epochs** כדי להגיע לאחוזי דיוק ~95%. כלומר הקטנתי את זמן הריצה של הרשת ל-20 epochs על ידי הגדלת כמות הניורונים בשכבה בלי שאיבדתי את האיכות הלמידה, אלא אפילו האיכות עלתה. עצרתי ב-100 ניורונים מכיוון שכל הערכים שגדולים יותר מ-100 כבר לא מפשרים משמעותית את הלמידה.

נורמליזציה:

```
norm = 0.99 / 255
norm_train_x = train_x * norm + 0.01
norm_test_x = test_x * norm + 0.01
```

עשיתי נירמול בצורה כזאת כדי להימנע מערכי ה-0 מכיוון שזה יכול לגרום לזה שהמשקלים לא יתעדכנו.

:Learning rate

בחרתי בערך 0.01 מכיוון שערכים יותר קטנים היו מעכבים בקצת את תהליך הלמידה והערכים יותר גדולים היו גורמים ללמידה של 100% וכביכול סיכוי ל-overfitting.