

למידת מכונה תרגיל 3 - report

רוני ויגנסקי 211545892

רשת נוירונים מורכבת ממספר שכבות. בתרגיל בדקתי מה מספר השכבות שיוביל אותי לביצועים טובים, כמו גם מה מספר הנוירונים הרצוי בכל שכבה, כמה epochs לבצע ומהו קצב הלמידה (RATE). בהתאם לבדיקות שאפרט בהמשך הגעתי לבחירות הבאות:

גודל שכבת הקלט: 28 על 28 כגודל התמונה, כלומר 784 נוירונים.

שכבה חבויה אחת שגודלה: 100 נוירונים.

גודל שכבת הפלט: 10 נוירונים כמספר ה classes שיש לסווג אליהם.

מספר epochs: 12 RATE: 0.3 . (בדקתי כמה זמן המודל פועל במחשב שלי, והוא רץ 4 דק וכמה שניות.)

בחרתי SEEDS טובים עבור ה shuffle ההתחלתי וכן עבור ה shuffle בתוך כל epoch. הגרלתי אותם רנדומית וקיבעתי את המודל שלי עם SEEDS שנתנו לי אחוזי accuracy גבוהים. ה SEEDS הנבחרים:

FIRST_SEED = 967

SEEDS_ARRAY = [722, 766, 54, 185, 927, 36, 604, 909, 108, 782, 885, 128]

אופן התהליך בו עבדתי: נרמלתי את המידע שקיבלתי, כלומר חילקתי ב-255, משמע, נרמלתי כל פיקסל בתמונה, כדי למנוע תנודתיות גבוהה מידי בכניסה לרשת. אתחלתי את המשקולות W_i ו b_i בפעם הראשונה באופן רנדומי בין 0 ל 1, ואז שלחתי לאימון. תהליך האימון מורכב מקריאה ל fprop ול bprop שהם עיקר האלגוריתם במציאת \hat{y} ושינוי המשקולות כך שנקבל פרמטרים אופטימליים לקבלת loss מינימלי.

את fprop ביצעתי בדומה למה שלמדנו בשיעור, בחרתי את פונקציית האקטיבציה להיות פונקציית ה sigmoid ובשכבה האחרונה הפעלתי את פונקציית האקטיבציה softmax, שאיתה פועלים במקרה של multi class. בשלב זה נרמלתי גם את הוקטורים בשלב שלפני הכנסתם לפונקציית האקטיבציה, חילקתי כל וקטור במספר הנוירונים שבשכבה.

את bprop גם כן ביצעתי בדומה למה שראינו בשיעור, מצאתי את פונקציית ה loss שלנו שהינה $-\log(h_2)$, כך ש h_2 זהו הווקטור לאחר שהפעלתי את פונקציית ה softmax שצינתי קודם.

גזרתי לפי W_i ולפי b_i בצורה הבאה:

$$\begin{array}{llll} \text{W2:} & \text{W1:} & \text{B2:} & \text{B1:} \\ \frac{dL}{dz_2} = \frac{d(-\log(h_2))}{dz_2} = (h_2 - y) & \frac{dL}{dz_1} = \frac{dL}{dz_2} \frac{dz_2}{dh_1} \frac{dh_1}{dz_1} = ((h_2 - y) \cdot W_2) * (sig(z_1) * (1 - sig(z_1))) & \frac{dL}{dB2} = (h_2 - y) = \frac{dL}{dz_2} & \frac{dL}{dB1} = \frac{dL}{dz_1} \frac{dz_1}{dB1} = \frac{dL}{dz_1} \cdot 1 = \frac{dL}{dz_1} \\ \frac{dL}{dW2} = (h_2 - y) \cdot h_1^T & \frac{dL}{dW1} = \frac{dL}{dz_1} \frac{dz_1}{dW1} = \frac{dL}{dz_1} \cdot x & & \end{array}$$

לפי זה עדכנתי את b_i ו W_i בהתאמה, לפי SGD. כל תהליך האימון קורה בלולאה של מספר epochs. בתוך כל epoch עשיתי shuffle למידע כדי שלא ילמד את הסדר של המידע. הרצתי זאת על מספר שונה של epochs כדי למצוא את המספר האופטימלי, וקבעתי על סמך ההרצות שעשיתי תוך השוואת הפלטים לפי אחוזי ה accuracy. בשביל למצוא מה מספר הנוירונים בכל שכבה ומספר השכבות בכללי, הרצתי את התהליך הנ"ל על מודל עם שכבה אחת חבויה ועל מודל עם 2 שכבות חבויות. עבור המודל עם השכבה האחת יצאו ביצועים טובים יותר, כמו כן, המודל עם השכבה החבויה האחת רץ מהר יותר ולכן לא רציתי להכביד עם יותר שכבות שיגמרו לזמן ריצה גדול יותר ולא בהכרח ישפרו את הביצועים. את כל מה שתיארתי לעיל, ביצעתי על חלוקה ל train ול validation כדי לראות מה הערכים הטובים שיוצאים לי עבור data רחב יותר ולא data ספציפי. כלומר לאחר האימון שלחתי ל test על validation בו בדקתי את אחוזי ההצלחה. בעזרת בדיקות אלה קבעתי את כל הפרמטרים שצינתי לעיל.

בשביל אימון train כולו לצורך הגשת התרגיל, אימנתי כפי שפרטתי לעיל (כמובן שבלי חלוקות ל train ול validation). לאחר האימון, שלחתי את ה data לפונקציית test שלוקחת את המשקולות W_i ו b_i שעדכנתי בתהליך האימון ואת המידע החדש אותו רוצה לבחון, וביצעתי fprop כדי להגיע ל \hat{y} , אותם כתבתי לתוך הקובץ test_y כנדרש.