

למידת מכונה תרגיל 4

בתרגיל זה למדנו לעבוד עם חבילות ה-pyTorch, דבר אשר עזר לנו לייעל את התרגיל, שכן חלק מהחבילות מוכנות וניתן להשתמש בהן, ובעיקר כי ה-pyTorch יודע לגזור את תהליך ה-back-propagation בעצמו בהתאם ל-forawrd שכתבנו.

הכנות מקדימות טרם ריצת המודל:

- טעינת הדאטה – ע"י שימוש בפונקציה הניתנה לנו בפיאצה GCommandLoader, ולאחר מכן בעזרת פונקציית ה-DataLoader של pyTorch, אליה הכנסנו attributes נוספים (גודל batch, האם לעשות shuffle וכו').
- שימוש ב-shuffle – השתמשנו ב-shuffle על קבצי ה-train וה-valid בלבד. על קובץ ה-test אין צורך לעשות shuffle (שכן לא מתבצעת שם למידה כלל, אלא רק בחינה). כמו בתרגילים הקודמים ביצענו shuffle וכן ביצענו את תהליך ה-cross validation בכדי לוודא שאחוז השגיאה דומה בין שניהם (כלומר, ווידאנו שהאלגוריתם שלנו לא למד "לשנן" את ה-training set - אלא באמת למד לסווג כל data שהוא יקבל).
- איתחול המודל – האתחול התבצע בבנאי של המחלקה ConvolutionModel שבנינו. מחלקה זו מתארת את המודל שלנו. המודל מורכב מ-3 שכבות – את השכבות הגדרנו כך שמימדיהן בחזקות של 2, כיוון שכפי שלמדנו זו אחת הדרכים האופטימליות לייעל כפל מטריצות, ובכך המודל שלנו נעשה יעיל יותר. כמו כן עשינו שימוש בפונקציית אקטיבציה relu וכן שימוש ב-pooling וב-drop-out על מנת למקסם את יכולות הלמידה של המודל שלנו.
- פונקצית ה-loss – לפי עקרון הניסוי והטעייה "שיחקנו" עם המודל וכל פעם ניסינו פונקציות loss שונות (למשל nll_loss), ולבסוף השתמשנו בפונקציה CrossEntropyLoss.
- בחירת optimizer – בדומה לבחירת ה-loss גם פה בחרנו לפי ניסוי וטעייה. ניסינו SGD, RMS-Prop וכו' ולבסוף בחרנו להשתמש ב-Adam כיוון שהוא משלב את היתרונות של מומנטום ו-RMS-Prop ביחד.

בחירת היפר-פרמטרים:

- eta – קצב הלמידה של המודל, ה-learning rate. תחילה ניסינו לשים את הערך 0.1 וראינו שהמודל מגיע לתוצאות לא טובות מספיק. לאחר מכן ע"י שיטת ניסוי וטעייה ניסינו להגדיל ולהקטין ולבסוף ניסינו את קצב הלמידה שערכו 0.001 והוא זה שנתן את התוצאות הטובות ביותר למודל שלנו.
- epoch – מס' האיטרציות שהמודל רץ. התחלנו עם 10 פעמים אך הגענו ל-overfitting – המודל שינן היטב את ה-train (הגיע אף ל-100% הצלחה) אך על test התוצאות לא היו טובות. קיצצנו בחצי וירדנו ל-5 אך גילינו כי התוצאות לא טובות מספיק וכשבחרנו אותו להיות 7, הגענו למצב בו האלגוריתם גם יגיע לתוצאות הטובות ביותר שהוא מסוגל, וגם זמן הריצה יהיה סביר ולא ארוך מדי.