למידה עמוקה בפייתון

מגישים: רודין חאתוקאי-207064353

מוסא תחאוך-311590707

תיאור הפרוייקט:

המטרה של הפרוייקט היא להמחיש לנו את היכולות של pytorch בתחום הלמידה העמוקה ובנוסף לכך להמחיש לנו איך עושים את זה, משמע להראות את היכולת של המשאבים הקיימים שאנו יכולים להשתמש בהם ללימוד מודלים של בינה מלאכותית מבלי להכנס לליבה של איך לבנות את הבינה מלאכותית עצמה ובכך יש לנו יותר מרחב להתעסק עם איך ללמד (האלגוריתם לימוד) את המודלים האלה בצורה יעילה.

ספריות בשימוש:

OS:

ספרייה שנותנת ממשק לפעולות של מערכת הפעלה לדגומה פתיחת קבצים

TORCH:

ממשק לעבודה נוחה עם מערכים עם ממדים גדולים עבור למידה חישובית, מספקת אלגוריתמים ללמידה עמוקקה

TORCHVISION:

ספריית ראיית מחשב מספקת ממשק לצורך מניצפולציה בתמונות לעיבוד תמונות

PANDAS:

ספרייה התומעת בעיבוד נתונים וניתוח מציעה מבני נתונים (ניתן להשתמש בפנדה כמו אקסאל בתוך פייתון)

TORCH.NN:

ספרייה לבניית רשת ניורנית

TQDM:

ספרייה לתצוגת טעינה בתוך notebook

MATLABLIB.PYPLOT:

ממשק לפונקציות שגורמות לעבוד כמו matlab, בעיקרון משתמשים להציג תמונות וגרפים.

PYTORCH.NN.FUNCTIONAL:

ספרייה המספקת פונקציות לחישובים מתמטיים וסטטיסטיים

TORCH.UTILS.DATA:

ממשק לבניית מבנה נתונים

TORCHVISION.MODELS:

מספקת מודל של resntes

TORCHVISION.UTILS:

ממשק לעיצוב תמונות

TORCHVISION.TRANSFORMS:

ממשק למניפולציות של תמונות שינוי גדלים ובחירת קטעים מהתמונות ועוד

TORCHVISION.DATASETS.FOLDER:

טעינת תיקייה של תמונות לתוך מערך

שלבים במימוש:

ביידנו תיקייה בה תמונות פרחים וקובץ שמכיל זהויות של פרחים המתאימים לתמונות, בעזרת Dataset מהספרייה של torch.utils.data, אשר משמשת לבנייה ופעולות על אובייקטים באופן חד משמעי ונוח (במקרה זה, אובייקט בו יש תיקיית תמונות וקובץ .csv בו מספר זיהוי ושם שתואם לתמונות ובנוסף לכך אופציה לטרנספורמציה על התמונה בכל אופן שנרצה, אם נרצה) הגישה לתמונות מותאמת כך שהאינדקס יקשר בין התמונה והזיהוי שלהו בקובץ הזהויות.

המטרה של הבנייה הזו היא לשימוש בחינת המודל (הערכת המודל במקרי אמת), יש לנו עוד שני מקרים אליהם אנחנו צריכים להכין את המערכת והן הם לאימות(למען הערכת המודל בזמן האימון) ואימון המערכת.

למען המשתנה של האימון אין צורך לבניית טיפוס dataset מכיוון שהפרחים מחולקים לתיקיות לפי סוגם (שמם) כנ"ל גם למשתנה של תמונות האימות, נציין שגם השמות של התמונות עצמם בתיקיית train ו-val מתואמים עם קובץ ה-flowers_IDx, ובכך השימוש הוא במשתנים\אובייקטים מהספריות torch -ו torchvision המכילים הכרה ופעולות tensor ופעולות טרנספורמציה של תמונות שיכולות לייעל את האימון.

בשלב זה יש טעינה למשתנה DataLoader מספריית torch.utils.data העוזר לחלק את המאגר שלנו לקבוצות ולתת גישה לקטעי מידע מהם כדי לא להעמיס קריאה וכתיבה חוזרת או איטית על התוכנית.

הגדרת המודל:

בתחילת הדבר אנו בונים מחלקה שתחשב את פונקצית מחיר - הפסד ובכך נוכל לאמן ולמדוד את רמת הדיוק של המודל.

בדיקת רמת הדיוק בסיסית למדי, ממוצע העצמים שחוזו נכון חלקי מספר עצמים שיש לחזות, משמע אחוז מספר העצמים שחזינו נכון.

חישוב פונקצית מחיר - הפסד שלנו נעשת על ידי פונקציה הנקראת cross_entropy מתוך ספריית torch.nn.functional כאשר nn היא ספרייה מתוך torch המיועדת לבניית רשתות ניורוניות.

שלב למידה:

בשלב זה אנו בוחרים ב-resnets כמודל שלנו למערכת ונותנים לה את ההגדרות שכתבנו לעיל.

שיטת העבודה של resnets היא דגימת מקטעי 3x3 של התמונה וסכימה שלהם לערך יחיד שכופלת בסוף עם ערכי התמונה, היתרון שנאמר לנו שיש ל-resnets הוא שהיא יכולה לעבוד ולהתאים עבודה על תמונות קלט בצורות שונות.

את המודל resnet אנו מאתחילים במחקלה שלנו של flower_resnet, נשים לב לפונקציות freeze ו-unfreeze

שאמורות להקפיא את רוב השכבות מעודף אימון כי הרי המודל שבחרנו מאומן כבר לזיהוי תמונות ואנו רוצים רק שיזה את סוג הפרח עכשיו.

עבודה על GPU:

בשלבים הבאים אנחנו פשוט מעבירים את העבודה של התוכנית ל-GPU שמתאימה יותר לעבודה מקבילית של מודלים

קבוצת הפונקציות הבאות משתמשות בפונקציות מהגדרת המודל כדי לבצע את שלב הלמידה יחד עם תכונות ה- hyper paramters

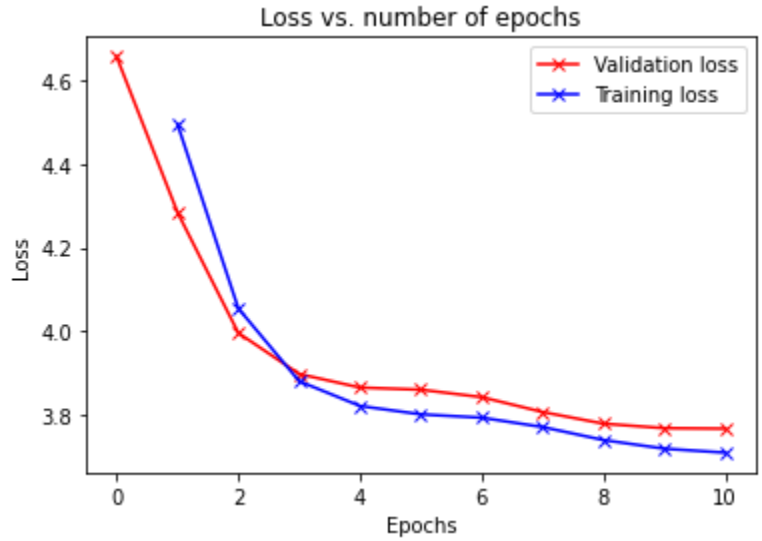
שזה אומר לבנות פונקציה עבור אימות המודל הקיים ברגע נתון, אימון בפרקי זמן נתונים ואופטימזציה של המודל עם משתני ההייפר, בכל פרק זמן אנחנו יכולים לראות הדפס של תוצאות הדיוק וערכי פונקצית ההפסד של האימון, ולהדפיס גם כן את ערכי ההפסד של המודל באופן כללי בעזרת פונקציית ההערכה

תוצאות:

את המודל אימנו 10 פרקי זמן, 5 מהם אימנו שכבה אחת של בלבד, בו קיבלנו שהדיוק של המודל הוא בערך 86% (גם באימון 14 פרקי זמן הערך היה זהה אפילו פחות מהריצה הראשונית עם 10).

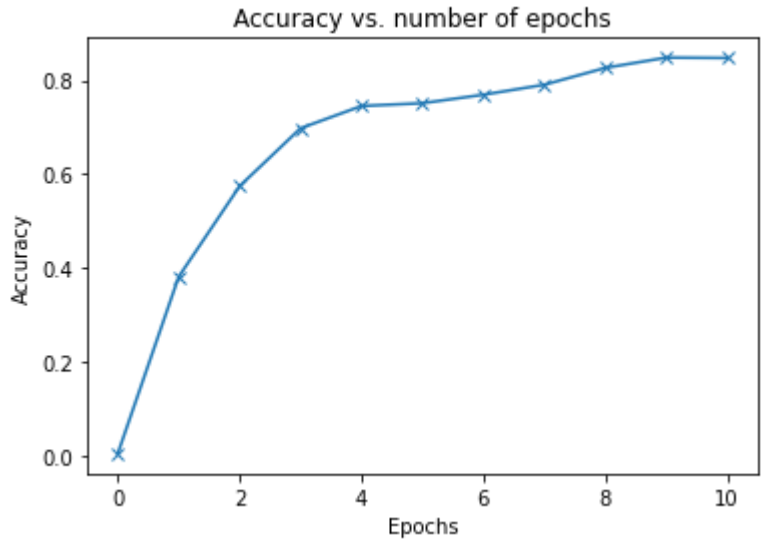
בגרף הבא אנחנו רואים את ערך פונקצית מחיר - הפסד כתלות בפרקי זמן אימון שכולל בתוכו גם אימות (סה"כ 10 בהם ב-5 מהם אימנו שכבה אחת בלבד), הvalidation loss משער ו"מתקן" את הערך שה-training loss קיבל ולכן בכל שלב אנחנו נראה שהערכי loss שלנו בזמן אימון יקטנו לפני שהם קטנים בשלב האימות בו אנחנו משערכים מחדש את מדד ההערכה הנכונה של המודל (מספר הקביעות הנכונות בזמן הנתון)

מהרצה נוספת עם epochs גדולים יותר, ניתן לראות שהפונקציות מתיישרות (כבר ב-9 epochs אפשר לראות) ואין ירידה משמעותית בערכי ה-Loss של שני הגרפים.



בהמשך לעיל אנו רואים שהדיוק של המודל התייעל לאחר כל הרצת פרק זמן, אך נראה ש(לאחר בדיקה הרצה של פרקי זמן גדולים מ-10 גם כן) אחוז הדיוק לא משתפר אחרי 8 epochs כל כך והוא מתחיל להתיישר (leveling out) בלי עליה משמעותית, המשמעות היא שיש מקום לשיפור בערכי ה hyper parameters למען כיוון הלמידה משמע הערכים הבאים:

max_lr,grad_clip,weight_decay,opt_func ,Epochs



שיפורים אפשריים הם שיפור ערכי hyper parameters, שימוש במודל אחר, למשל resnet50, שימוש בפונקצית מחיר - הפסד שונה, מאגר תמונות גדול יותר.

נוסיף שנראה ש-86 או אפילו 83 אחוז דיוק נחשב למספק אך עושה רושם שיש מקום לשיפור

הגרפים הבאים הם בדיקה של שינויים בערכי הייפר שונים.

Epochs = 7

Weight_decay = 1e-5

Val_acc = 87