

פרוייקט סיום – מימוש רשת עמוקה מבוססת TRANSFER LEARNING

הנחיות הגשה

1. הגשה חובה בקבוצות של שלושה.
2. בנוסף להגשת הקוד שכתבתם, יש להגיש גם דוח בפורמט PDF שבו:
 - פירוט של כל התוצאות שהתקבלו בכל משימה.
 - מענה על השאלות בכל משימה.

מבוא

בפרוייקט זה נממש רשת לסיווג תמונות צבע של פרחים משבעה עשר סוגים שונים, תוך שימוש בטכניקת **TRANSFER LEARNING**:

1. נטען מקובץ רשת שתוכננה לסיווג 1.2 מיליון תמונות מ-1000 סוגים (תחרות ILSVRC). הרשת פותחה ע"י GOOGLE והיא נתונה בקובץ GoogLeNet_v2_nn.t7. קובץ זה מכיל את מבנה הרשת ואת כל הפרמטרים שאומנו בתהליך נפרד.
2. ניצור רשת חדשה שתכלול את 10 השכבות הראשונות של הרשת GoogLeNet_v2_nn.t7 (כולל כל הפרמטרים). נוסיף לרשת החדשה מספר שכבות לצורך סיווג תמונות הפרחים.
3. נאמן את הרשת החדשה תוך הקפאת הפרמטרים של 10 השכבות הראשונות. כלומר, רק הפרמטרים של השכבות החדשות יאומנו, באמצעות סט אימון תמונות הנתונות בקובץ flowers.t7.
4. נבדוק את ביצועי הרשת שלמדנו על סט הבדיקה של תמונות הפרחים.

משימה מס' 1 – יצירת סט תמונות האימון והבדיקה

בקובץ transfer_learning_project המצורף מבוצעת טעינה של כל התמונות ויצירת TENSOR הכולל 17 קבוצות פרחים, ובכל קבוצה 80 תמונות צבע בגודל 128 X 128 פיקסלים. בנוסף, נוצרים סט אימון (הכולל 85% מסך כל התמונות) וסט בדיקה (הכולל 15% מסך כל התמונות), כל אחד עם 17 קבוצות.

שנו את הקוד כך שיתווסף משתנה בשם NUM_CLASSES אשר יכיל מספר בין 2 ל-17. כעת יש ליצור סט אימון וסט בדיקה הכוללים NUM_CLASSES קבוצות פרחים, ובכל קבוצה 80 תמונות כ"א. מומלץ להשתמש בפקודה NARROW לצורך ביצוע משימה זאת.

משימה מס' 2 – בניית הרשת החדשה

בקובץ transfer_learning_project המצורף מבוצעת טעינה של GoogLeNet_v2_nn.t7, ויצירת רשת חדשה שכוללת את 10 השכבות הראשונות של הרשת GoogLeNet. בנוסף, הקוד מממש את הקפאת הפרמטרים של שכבות אלה.

הוסיפו לרשת החדשה את הרכיבים הבאים:

1. שכבת קונבולוציה עם מימד כניסה 320, מימד יציאה 16, גודל המסננים 3X3, Stride=1, ללא PADDING.
2. RELU
3. MAXPOOLING על חלונות בגודל 4X4 עם Stride=4 בכל מימד.
4. המרת ה- FEATURE MAPS לוקטור עמודה (באמצעות nn.View).
5. DROPOUT עם הסתברות 50% (באמצעות nn.Dropout).
6. שכבה FULLY CONNECTED (באמצעות nn.Linear, מספר היציאות הינו NUM_CLASSES).
7. LOGSOFTMAX

משימה מס' 3 – אימון הרשת

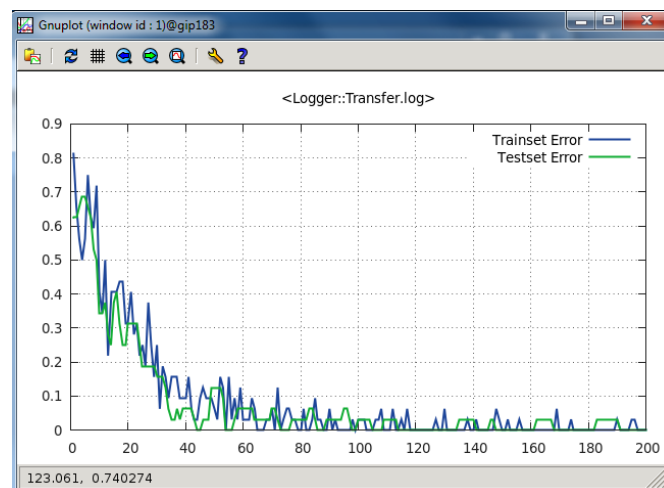
קבעו NUM_CLASSES=4, והוסיפו את כל הקוד הנדרש לאימון הרשת (בדומה למבנה הקוד שניתן בתרגיל בית 2).

הוסיפו תצוגה של סיכוי הטעות (אימון ובדיקה) באמצעות `CONFUSION MATRIX` ו-`LOGGER`. קבעו את גודל `MiniBatch` להיות 32. שימו לב כי בכל צעד אימון יבחר `MiniBatch` אחד בלבד מתוך כל סט האימון (אחרת ההרצה תהיה מאד איטית). אל תשכחו לערבב את סדר דוגמאות האימון לאחר כל צעד אימון.

הריצו את האימון באמצעות אלגוריתם ADAM ע"י שימוש בפקודה: `optim.adam(feval,w,optimState)` וקבעו בתוך `optimState`: `learningrate = 0.1`.

קבעו את מספר צעדי האימון המקסימלי להיות 200. ממשו תנאי לעצירת האימון אם סיכוי הטעות של סט הבדיקה נמוך מ-10% (`EARLY STOPPING`).

להלן גרף אימון לדוגמא (ללא `EARLY STOPPING`):



הציגו את גרף האימון שקיבלתם. בנוסף, בסיום האימון הציגו 10 תמונות כלשהן מסט האימון, ולכל תמונה הציגו את וקטור ההסתברויות שחישבה הרשת עבורה (המירו את מוצא שכבת `LOGSOFTMAX` להסתברויות). האם סיכוי הטעות (במוצע על 10 התמונות שבחרתם) תואם את הגרף?

משימה מס' 4 – אימון הרשת למספר משתנה של קבוצות

חזרו על משימה 3 עבור `NUM_CLASSES=8,12,16`. ממשו את התנאי לעצירת האימון אם סיכוי הטעות של סט הבדיקה נמוך מ-15% עבור 8 קבוצות ו-20% עבור 12,16 קבוצות. להלן גרף אימון לדוגמא עבור 12 קבוצות (ללא `EARLY STOPPING`):

