

ML – EX4 – SPEECH RECOGNITION

USING PyTorch

נפרט את תהליך קיבוע כלל הפרמטרים של המודל:

אלגוריתם האופטימיזציה:

אלג' האופטימיזציה נקבע בתחילת התהליך להיות Adam, וזאת בהתבסס על החומר שהועבר בתרגולים. תכונה חשובה של אלג' Adam היא שהוא משמר 'מומנטום' לכיוון המינימום הלוקאלי, כלומר, בעת זיהוי שיפור בערכי הגרדיאנט עבור כיוון מסוים, אלג' Adam ייתן לכיוון הנ"ל משקל גדול יותר במעמד העדכון. תכונה יעילה נוספת היא שהוא שומר על איזון בקצב הלמידה (בהשוואה ל-Adagrad לדוגמה) כך שהרשת לא תסיים ללמוד בשלב מוקדם של התהליך.

ארכיטקטורת המודל:

תחילה, קיבענו את ארכיטקטורת הבסיס של המודל להיות LeNet (כלומר, מן הצורה: CONV-POOL-CONV-POOL-FC-FC), ולאחר מכן ביצענו שינויים בהתאם לתוצאות על סט הולידציה. מן הבדיקות על סט הולידציה (לצורך אחידות, כלל הארכיטקטורות ביצעו 20 איפוקים של אימון) זיהינו שהמודל המתאים ביותר היה מן הצורה:

CONV-POOL-CONV-POOL-CONV-POOL-FC-FC-FC

כאשר לאחר כל שכבה (בין אם שכבת קונבולוציה ובין אם לינארית) הופעלה פונקציית אקטיבציה מסוג ReLU וכן נרמול מסוג Batch-Norm. לאחר הפעלת שכבת ה-FC האחרונה, החזרנו את הפלט ללא הפעלת SoftMax וזאת משום שהגדרנו את פונקציית העלות של המודל להיות Cross-Entropy, ומבדיקת מימושה ראינו כי מתבצע בה SoftMax.

נציין כי לכל אורך התהליך הקפדנו על מספר נקודות (מקור: [Stanford-CS231](#), סופק ע"י המרצה – ד"ר יוסי קשת):

1. לכל אורך בניית המודל, הקפדנו כי הקטנת ממדי הדגימה תתבצע באופן בלעדי ע"י הפעלת POOL-MAX (וכמובן, גם ע"י השכבות הלינאריות). הרווח מכך הינו אי-הצורך לעקוב אחרי שינויי הממדים בשכבות הקונבולוציה ולכן החופש בבחירת צורתם ומספרם עזר לנו לבחון מס' רב יותר של ארכיטקטורות. הגענו למטרה זו ע"י ריפוד נכון של קלטי שכבות הקונבולוציה, כך שהפלט יהיה זהה בממדיו לקלט.
2. הריפוד של שכבות הקונבולוציה, בנוסף לשמירת ממדי הדוגמה, תרם גם לכך שפחות 'מידע' מדפנות הדוגמה (העמודות והשורות בעלות אינדקס מס' 0) נאבד – שכן כתוצאה מכך הפילטר מתחיל לנוע מעמודות/שורות הריפוד, ולכן "עובר" על דפנות הדוגמה.
3. פרט להפעלה השלישית והאחרונה של POOL-MAX, בה השתמשנו בפילטר בגודל 5X5 ובעל Stride=5 (כדי להכין את הדוגמה לקראת שכבות ה-FC), הקפדנו על פילטרים ו-Strides קטנים (3 לכל היותר עבור ממדי הפילטר, 2 עבור ה-Strides) וזאת בכדי להימנע מאיבוד מידע מיותר של הדוגמה.

קצב הלמידה עבור אלגוריתם האופטימיזציה Adam:

קובע לאחר בניית המודל שתואר לעיל. ראשית, בדקנו קצבי למידה שונים כפרמטר ל-Adam כאשר הרשת מבצעת 20 איפוקים, ובכל batch (שהינו בגודל 100) הדפסנו את ערך ה-loss המתאים לו. $(10^{-5} - 10^{-1})$, הקצב המתאים ביותר לרשת היה 0.01.

מס' ה-epochs:

כאשר אימנו את הרשת במס' איפוקים הגבוה מ-20 (הערכים שנבדקו היו 25 ו-30) שמנו לב כי פונקציית העלות אמנם הייתה במגמת ירידה (per Batch), אך אחוז הדיוק על סט הולידציה היה זהה – 88% ~ 84%. כלומר, מס' איפוקים ≤ 20 לא היה אפקטיבי ואף תפס זמן GPU יקר בו היינו יכולים לחפש עבור היפר-פרמטרים שונים ערך אופטימלי יותר.

כעת, נציג את מבנה הרשת הכולל:

* אם לא צוין אחרת, מתקיים Stride=1.

Layer Kind	Hyper-Parameters	Input & Output shapes
CONV	In channels = 1 Out channels =6 Kernel=3X3 Padding=1	In = [1,161,101] Out=[6,161,101]
POOL-MAX	Kernel=3X3 Stride=2	In = [6,161,101] Out = [6,80,50]
CONV	In channels = 6 Out channels = 12 Kernel=3X3 Padding=1	In = [6,80,50] Out = [12,80,50]
POOL-MAX	Kernel=2X2 Stride=2	In = [12,80,50] Out = [12,40,25]
CONV	In channels = 12 Out channels = 16 Kernel = 3X3 Padding = 1	In = [12,40,25] Out = [16,40,25]
POOL-MAX	Kernel = 5X5 Stride = 5	In = [16,40,25] Out = [16,8,5]
FULLY-CONN.	-	In = [16X8X5] Out = [120]
FULLY-CONN.	-	In = [120] Out = [60]
FULLY-CONN.	-	In = [60] Out = [30]