כוונון פרמטרים היפר-פרמטריים לרשת מחוברת מלאה על MNIST

1. מבוא

כוונון היפר-פרמטרים הוא שלב קריטי בבניית רשת נוירונים אפקטיבית. בחירת שילוב אופטימלי של היפר-פרמטרים (כגון שיעור הלמידה, שיעור ה-dropout, ועוד) יכולה לשפר באופן משמעותי את הדיוק וההכללה של המודל. מסמך זה מתאר את כוונון ה-Multi-Layer Perceptron על מערכת נתונים MNIST, שבה:

- MNIST מורכב מ-60,000 תמונות אימון ו-10,000 תמונות מבחן של ספרות כתובות ביד (0–9), כל אחת בגודל 28×28 פיקסלים.
- נעשה שימוש בארכיטקטורה מחוברת מלאה, עם ארבע שכבות Dense. מספר הנוירונים בכל שכבה נגזר מזהות ספציפית, והתוצאה היא המידות [82, 4, 76, 12].
- נעשה שימוש ב-Keras Tuner כדי לחפש באופן שיטתי את ערכי ההיפר-פרמטרים האופטימליים.

2. ארכיטקטורת המודל

קונפיגורציה בסיסית

1. שכבת קלט: מפשטת את תמונות ה-28×28 פיקסלים לווקטור בגודל 784.
2. ארבע שכבות Dense:
   - שכבה 1: 82 נוירונים
   - שכבה 2: 4 נוירונים
   - שכבה 3: 76 נוירונים
   - שכבה 4: 12 נוירונים
3. Batch Normalization ו-ReLU: כל שכבה מלווה ב-BatchNormalization, ולאחר מכן הפעלת ReLU.
4. Dropout: חלה הסתברות dropout מסוימת לאחר כל הפעלת activation כדי להפחית overfitting.
5. שכבת פלט: שכבת Dense אחרונה עם 10 יחידות (אחת לכל קטגוריית ספרה), תוך שימוש בהפעלה מסוג softmax.

- Batch Normalization מסייעת לייצב ולהאיץ את האימון על ידי הפחתת שינויים פנימיים בקורלציות (internal covariate shift).
- Dropout פועל כצורת רגולריזציה על ידי "כיבוי" רנדומלי של חלק מהנוירונים בכל מעגל, ובכך מונעת תלות יתר בנוירון בודד.
- הפעלת ReLU היא בחירה פופולרית וסטנדרטית לרשתות מחוברות מלאה בזכות פשטותה ויעילותה.

3. כוונון היפר-פרמטרים

3.1 סקירה של Keras Tuner

השתמשנו ב-Keras Tuner, ספריה שמאוטומטת את תהליך חיפוש ההיפר-פרמטרים. זה כולל:

1. הגדרת מרחב החיפוש לכל היפר-פרמטר.
2. ניסיון קומבינציות שונות (כלומר, "ניסויים") ומדידת ביצועים על קבוצת אימות.
3. בחירת הקומבינציה עם הדיוק הגבוה ביותר בקבוצת האימות כסט ההיפר-פרמטרים הסופי.

3.2 הגדרת מרחב החיפוש

במהלך הכוונון הגדרנו את ההיפר-פרמטרים והטווחים הבאים:

1. שיעור הלמידה (η\etaη):
   - נבחר באופן לוגריתמי בטווח של 10^{-4} עד 10^{-2}.
   - כך נתפסים גם שיעורי למידה נמוכים (שמרניים) וגם גבוהים (אגרסיביים).
2. שיעור ה-dropout לכל שכבה (סה"כ 4 שכבות):
   - עבור כל שכבה, שיעור ה-dropout נבחר מתוך {0.0,0.1,0.2,0.3,0.4,0.5}
   - ה-Keras Tuner מחפש באופן עצמאי את שיעור ה-dropout האופטימלי לכל שכבה.

3.3 תהליך הכוונון

1. אלגוריתם החיפוש: השתמשנו ב-RandomSearch. אלגוריתם זה מדגם קומבינציות רנדומליות מתוך מרחב החיפוש המוגדר עבור מספר ניסויים.
2. מספר הניסויים: הגדרנו `max_trials = 5` בניסוי. (הגדלת מספר הניסויים יכולה לשפר את התוצאות, במחיר של זמן חישוב ארוך יותר.)
3. ביצוע לכל ניסוי: כל קונפיגורציה של היפר-פרמטרים אומנה פעם אחת (`executions_per_trial = 1`).
4. הגדרות אימון:
   - אפיקים: 10
   - גודל קבוצת נתונים: 128
   - פיצול אימות: 20% מתוך נתוני האימון נשמרו לאימות במהלך הכוונון.
   - מניעת אופטימיזציה: השתמשנו ב-Adam בכל ניסוי, אך שיעור הלמידה היה חלק מהחיפוש.

3.4 תוצאות

לאחר 5 ניסויים, נמצא הסט האופטימלי של פרמטרים:

- שיעור הלמידה: 0.00694
- שיעור Dropout (שכבה 1): 0.1
- שיעור Dropout (שכבה 2): 0.1
- שיעור Dropout (שכבה 3): 0.1
- שיעור Dropout (שכבה 4): 0.2

הגדרה זו השיגה דיוק אימות של כ-96.2%. כאשר המודל אומן מחדש על כל מערכת האימון (באמצעות אותו פיצול), הושג דיוק מבחן של כ-95.9%.

4. מסקנה

ההיפר-פרמטרים האופטימליים (מהכוונון):

- שיעור הלמידה: 0.00694
- Dropout שכבה 1: 0.1
- Dropout שכבה 2: 0.1
- Dropout שכבה 3: 0.1
- Dropout שכבה 4: 0.2

לסיכום, תהליך הכוונון חקר בצורה שיטתית את שיעורי הלמידה מ-10−4{-4}^10−410 ועד 10−2{-2}^10−210 ושיעורי ה-dropout מ-0.0 ועד 0.5 לכל שכבה, וזיהה קונפיגורציה ששילבה למידה מהירה (שיעור למידה גבוה יחסית) עם מספיק dropout כדי להקל על overfitting. למרות שניתן להשיג דיוק גבוה יותר על MNIST עם ארכיטקטורות אחרות (כגון רשתות קונבולוציה), פרמטרים אלו מייצגים בחירה אופטימלית בתוך מגבלות המודל המחובר המלא ומרחב ההיפר-פרמטרים שהגדרנו.

על ידי יישום פרמטרים אלו בהדרכה הסופית של המודל, ניתן לצפות להצלחה גבוהה בסביבות 95-96% דיוק על MNIST עם הרשת המחוברת המלאה בעלת ארבע השכבות.

תהליך החיפוש עם Hyperband

אז איך Hyperband באמת עובד? מדובר באלגוריתם שיודע לבדוק מלא תצורות של היפר-פרמטרים בצורה יעילה. הוא בוחן כמה שיותר שילובים, אבל לא מבזבז זמן על כאלה שלא נראים מבטיחים. במקום לתת לכל תצורה לרוץ עד הסוף, הוא מפסיק את אלה שפחות מצליחות כבר בשלבים הראשונים.

כמה ניסיונות זה לקח?

בפרויקט הזה, Hyperband בדק 30 ניסיונות בסך הכול. כל ניסיון ייצג שילוב אחר של שיעורי דרופאאוט לשכבות ושיעור למידה. התהליך כולו לקח 11 דקות. כן, זה כל מה שהוא היה צריך כדי למצוא את התצורה הכי טובה! אלגוריתם כזה ממש חוסך זמן כי הוא לא מתעכב על תצורות שלא עובדות טוב.

מה הוא מצא?

- שיעורי דרופאאוט:
  - שכבה :1 0.1
  - שכבה :2 0.1
  - שכבה :3 0.1
  - שכבה :4 0.2
- שיעור למידה: 0.01

התוצאות של המודל הכי טוב

אחרי ש-Hyperband מצא את ההיפר-פרמטרים הכי טובים, המודל אומן מחדש, וזה מה שהוא השיג:

- דיוק באימות: 0.95
- דיוק בבדיקה: 0.94

למה Hyperband כל כך יעיל?

היופי ב-Hyperband זה שהוא יודע לעצור ניסיונות שלא הולכים לשום מקום מוקדם, ולתת יותר זמן ומשאבים לתצורות שנראות מבטיחות. זה אומר שאתה מקבל תוצאות מצוינות בזמן קצר מאוד. זה בדיוק מה שעשינו פה – תוך 30 ניסיונות בלבד, קיבלנו מודל עם ביצועים מרשימים על MNIST.

```
Best val_accuracy So Far: 0.9528999924659729
Total elapsed time: 00h 11m 01s
Best dropout rates: [0.1, 0.1, 0.4, 0.1]
Best learning rate: 0.001
```