מחלקה: מדעי המחשב
שם קורס: למידה עמוקה יישומית
מספר: 0-202-1-5031
מרצה: עומרי אזנקוט
סמסטר:
תאריך: 8.9.2024
משך בחינה: 2 שעות
הגשה: דף תשובות בלבד.
חומר עזר:

# למידה עמוקה יישומית
# Applied Deep Learning

## הערכה חלופית

**חלק א' (20 נק'): שאלות כלליות**. התשובה לכל שאלה צריכה להיות באורך של שורה אחת או שתיים לכל היותר.

1. ברשתות עמוקות, יש לעיתים נפוצות שימוש בפונקציות לא גזירות כגון sign. איך ניתן להשתמש בפונקציות כאלו במסגרת רשת שלומדת עם backpropagation?

_____

_____

2. כמה פרמטרים (weights) נוצרים עבור `RNNcell` שה-`input_size` שלו הוא 64 וה-`hidden_size` שלו הוא 256 ויש לו `bias` (לדוגמא ע"י `nn.GRUcell(64, 256)`)?

_____

_____

3. מנו חסרון אחד עיקרי לשיטה gradient clipping.

_____

_____

4. האם עדיף שלמודל יהיו הרבה hyper-parameters או מעט?

_____

_____

5. באילו מקרים רשת עמוקה תרוץ יותר מהר על CPU מאשר על GPU?

_____

_____

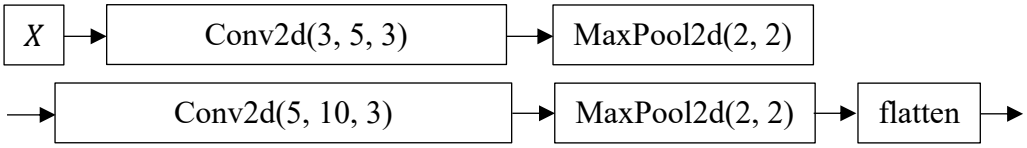
**חלק ב' (20 נק'): שאלות כלליות**. התשובה לכל שאלה צריכה להיות באורך של עד 5 שורות.

1. מה ההבדל בין `KL(p,q)` ל- `KL(q,p)`, כאשר KL זה ה-Kullback—Liebler divergence. הסבירו ותנו דוגמא. נזכיר ש- $KL(p,q) = \sum_x p(x)\log\left(p(x)/q(x)\right)$.

2. נניח ומימשתם רשת פשוטה מאוד כדי לפתור בעיה כלשהי. לצערכם, התוצאות ממש גרועות. מה כדאי לעשות כדי לשפר את הביצועים? האם תמיד כדאי להפוך את הרשת למאוד עמוקה ובעלת הרבה פרמטרים?

3. בהינתן רשת הקונבולוציה שמופיעה בדיאגרמה, פרטו את גדלי הטנזורים המתקבלים מהפעלת כל אחת משכבות הרשת עבור קלט בגודל $X \in \mathbb{R}^{b\times3\times28\times28}$, כאשר b הוא מספר הדוגמאות ב-batch. נזכיר ש-`Conv2d(3, 5, 3)` מבצעת קונבולוציה ומקבלת 3 ערוצים, מחזירה 5 ערוצים, גודל הקרנל (kernel) הוא 3. בנוסף, `MaxPool2d(2,2)` מחזירה את הערך המקסימלי מ-patch בגודל 2x2 עם stride בגודל 2. נזכיר את הנוסחא לחישוב גודל הפלט מ-Conv2d, כאשר $z_{in}, z_{out}$ הם רוחב הקלט והפלט בהתאמה או גובה הקלט והפלט בהתאמה של התמונה:

$$z_{out} = \left\lfloor \frac{z_{in} + 2 \times \text{padding} - \text{kernel}}{\text{stride}} + 1 \right\rfloor$$




4. תארו בפירוט את פרוטוקול בחירת המודל של Machine Learning. כלומר, תארו איך מאומן, נבחר ונבדק מודל ML בעל מספר hyper-parameters.

**חלק ג' (20 נק'): RNN.** לאחרונה, התפרסם מאמר המתאר שיטה רקורסיבית חדשה שנקראת Lipschitz RNN שהמשוואות שלה רשומות למטה. אנא ממשו מודול ב-pyTorch עבור המשוואות האלו. כלומר, עליכם לייצר class ולממש עבורו את הפונקציות `__init__` ו-`forward` וכל פונקציה אחרת שאתם רואים לנכון. פונקצית ה-`__init__` מקבלת את ה-`input_size` ו-`hidden_size` ומאתחלת את המודול. פונקצית ה-`forward` מקבלת את הקלט הנוכחי ואת ה-`hidden` הקודם ומחזירה את ה-`hidden` הבא. אם אינכם זוכרים את ה-syntax של pyTorch, מספיק לרשום פסאודו-קוד עם הסברים בגוף הקוד במידת הצורך.

משוואות Lipschitz RNN:

$$h_{t+1} = Ah_t + \tanh(Wh_t + Ux_{t+1} + b)$$

כאשר, המטריצות $A, W, U$ והוקטור $b$ הם הפרמטרים של המודל.

**חלק ד' (20 נק'): CNN.** תארו בפירוט את המאפיינים של רשתות CNN כפי שנלמדו בכיתה. נסו לקבוע עבור כל מאפיין האם הוא מהווה יתרון או חסרון, ואילו מאפיינים הם משמעותיים להתנהגות של הרשת.