פעפוע לאחור דרך הזמן

בעת הפעלת אלגוריתם אופטימיזציה מבוסס גרדיאנט על רשת נשנית, לא ניכר על פניו שניתקל בבעיה: כל פעולות החשבון בהן אנו משתמשים לחישוב המצב החבוי האחרון הן גזירות, ולכן ניתן לפעפע דרכן את גרדיאנט פונקציית השגיאה. כך באמת עושה מערכת ה-Autograd של PyTorch. יחד עם זאת, המבנה המסויים של רשת נשנית, בו אנו חוזרים ומשתמשים באותם פרמטרים שוב ושוב בשלבים שונים בחישוב מוביל לבעיות יציבות נומרית, בהן נדון כעת.

זכרו כי נוסחת עדכון המצב החבוי של תא נשנה פשוט היא

$$h_{t+1} = \tanh\left(W_{input}X_t + b_{input} + W_{hidden}h_t + b_{hidden}\right)$$

אך כעת חשבו על כך שעבור משפט קלט באורך T טוקנים אנו מפעילים נוסחה זו T פעמים עד ליצירת המצב החבוי הסופי, כפי שהופיע באיור הרשת הפרוסה בפרק הקודם.

נתחיל את הדיון במקרה הפשוט בו המצב החבוי הוא בעל מימד יחיד: h_t הוא סקלר, ובהתאם W_{hidden}, מטריצת משקלי השכבה הליניארית החבויה מתנוונת לסקלר גם היא. נסמן אם כך $W_{hidden} = w$. עתה, נרצה לחשב את נגזרת פונקציית המחיר ביחס לפרמטר זה, $\frac{\partial C}{\partial w}$, עבור צעד העדכון של SGD. w משפיע על החישוב המבוצע ברשת בפעם האחרונה בעת חישוב המצב החבוי האחרון, h_T, ולכן אם נניח כי הנגזרת $\frac{\partial C}{\partial h_T}$ כבר חושבה, נקבל מכלל השרשרת:

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial h_T}\frac{\partial h_T}{\partial w}.$$

בבואנו לחשב את

$$\frac{\partial h_T}{\partial w} = \frac{\partial}{\partial w}\tanh\left(W_{input}X_t + b_{input} + wh_{T-1} + b_{hidden}\right)$$

ניתקל באתגר אותו יוצר המבנה הרקורסיבי של הרשת: מצד אחד, w מופיע באופן מפורש בנוסחת המעבר, ומצד שני גם h_{T-1} עצמו חושב בעזרת אותם פרמטרים וחישוב זה משפיע גם הוא על הגרדיאנט. על כן, יש:

1. לחשב את הנגזרת המיידית $\frac{\partial h_T}{\partial w}$ כאילו h_{T-1} קבוע, אותה נסמן ב-$\left[\frac{\partial h_T}{\partial w}\right]$ לצורך מניעת בלבול.
2. לפעפע אחורה לזמן הקודם את הנגזרת, כלומר לחשב את $\frac{\partial h_T}{\partial h_{T-1}}$.
3. לחשב את $\frac{\partial h_{T-1}}{\partial w}$.
4. לחבר את התוצאות בעזרת כלל השרשרת.

נקבל לבסוף,

$$\frac{\partial h_T}{\partial w} = \left[\frac{\partial h_T}{\partial w}\right] + \frac{\partial h_T}{\partial h_{T-1}}\frac{\partial h_{T-1}}{\partial w}.$$

את $\left[\frac{\partial h_T}{\partial w}\right]$ ו-$\frac{\partial h_T}{\partial h_{T-1}}$ נוכל לחשב בנקל, ונעשה זאת בהמשך. לעומת זאת, בבואנו לחשב את $\frac{\partial h_{T-1}}{\partial w}$ ניתקל באותה צרה כמקודם שכן גם h_{T-1} תלוי ב-w באותן שתי דרכים: w מופיע בנוסחת המעבר

מ- h_{T-2} אל h_{T-1} וכן גם h_{T-2} עצמו חושב בעזרת w. על כן יש להמשיך לפעפע את השגיאה אל h_{T-2} בדומה לחישוב שביצענו לעיל:

$$\frac{\partial h_{T-1}}{\partial w}=\left[\frac{\partial h_{T-1}}{\partial w}\right]+\frac{\partial h_{T-1}}{\partial h_{T-2}}\frac{\partial h_{T-2}}{\partial w}.$$

נציב ביטוי זה בנוסחה עבור $\frac{\partial h_T}{\partial w}$ ונקבל:

$$\frac{\partial h_T}{\partial w}=\left[\frac{\partial h_T}{\partial w}\right]+\frac{\partial h_T}{\partial h_{T-1}}\left(\left[\frac{\partial h_{T-1}}{\partial w}\right]+\frac{\partial h_{T-1}}{\partial h_{T-2}}\frac{\partial h_{T-2}}{\partial w}\right)=$$
$$=\left[\frac{\partial h_T}{\partial w}\right]+\frac{\partial h_T}{\partial h_{T-1}}\left[\frac{\partial h_{T-1}}{\partial w}\right]+\frac{\partial h_T}{\partial h_{T-1}}\frac{\partial h_{T-1}}{\partial h_{T-2}}\frac{\partial h_{T-2}}{\partial w}.$$

השיקולים הנ"ל תקפים עבור כל אחד מהמצבים החבויים הקודמים ולכן יהיה צורך להמשיך לפעפע את השגיאה לאחור דרך הזמן עד שנגיע ל- h_1, כך שלבסוף, הנוסחה לפיה יש לחשב את $\frac{\partial h_T}{\partial w}$ היא:

$$\begin{aligned}\frac{\partial h_T}{\partial w}&=\left[\frac{\partial h_T}{\partial w}\right]+\\&+\frac{\partial h_T}{\partial h_{T-1}}\left[\frac{\partial h_{T-1}}{\partial w}\right]+\\&+\frac{\partial h_T}{\partial h_{T-1}}\frac{\partial h_{T-1}}{\partial h_{T-2}}\left[\frac{\partial h_{T-2}}{\partial w}\right]+\\&\vdots\\&+\frac{\partial h_T}{\partial h_{T-1}}\frac{\partial h_{T-1}}{\partial h_{T-2}}\cdots\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial w}\end{aligned}$$

בנוסחה זו מוקדשת שורה עבור כל אחת מאיטרציות הלולאה המופיעה בחישוב המעבר קדימה ברשת אלמן. השורה הראשונה היא התרומה של האיטרציה האחרונה, הקולטת את הטוקן האחרון במשפט, בעוד שהשורה האחרונה – התרומה של האיטרציה הראשונה. הבעייתיות החישובית בנוסחה זו, מעבר לאורכה, נגלית לעין כאשר שמים לב ש-

$$\frac{\partial h_t}{\partial h_{t-1}}=\left(1-h_t^2\right)\left[\frac{\partial}{\partial w}\left(W_{input}X_t+b_{input}+wh_{t-1}+b_{hidden}\right)\right]=\left(1-h_t^2\right)w.$$

שכן $\tanh'(x)=1-\tanh^2(x)$. בהצבת תוצאה זו בנוסחה לעיל נקבל

$$\begin{aligned}\frac{\partial h_T}{\partial w}&=\left[\frac{\partial h_T}{\partial w}\right]+\\&+w\left(1-h_T^2\right)\left[\frac{\partial h_{T-1}}{\partial w}\right]+\\&+w^2\left(1-h_T^2\right)\left(1-h_{T-1}^2\right)\left[\frac{\partial h_{T-2}}{\partial w}\right]+\\&\vdots\\&+w^{T-1}\left(1-h_T^2\right)\left(1-h_{T-1}^2\right)\cdots\left(1-h_2^2\right)\frac{\partial h_1}{\partial w}\end{aligned}$$

כעת ניכרים הגורמים הבעייתיים בנוסחה: הם מהצורה w^t. אם ערך הפרמטר הנוכחי גדול מ-1 (בערכו המוחלט), גורמים אלו שואפים לאינסוף מהר מאוד כך שעבור משפט קלט מספיק ארוך, הנגזרת $\frac{\partial h_T}{\partial w}$ תהיה גדולה מאוד בערכה המוחלט, דבר אשר ימנע מאלגוריתם האופטימיזציה להתכנס – זוהי תופעת הגרדיאנט המתפוצץ (exploding gradient).

עבור $|w|<1$ המצב אינו טוב בהרבה: גורמים אלו ישאפו לאפס מהר מאוד, ויחד איתם גם ההשפעה של הטוקנים המופיעים בתחילת המשפט על $\frac{\partial h_T}{\partial w}$. כתוצאה מכך ערך נגזרת זו יחושב בעיקרו על סמך התרומות של הטוקנים האחרונים שהוזנו לרשת והרשת לא תוכל ללמוד תלויות ארוכות טווח בתוך המשפט.

מכיוון שקשיים אלו נובעים מהפעפוע לאחור דרך המצבים החבויים, הם קיימים גם עבור שאר פרמטרי הרשת, כל עוד יש לפעפע את הנגזרות דרך התא הנשנה בשביל לחשב את גרדיאנט המחיר לפיהם. קשיים אלו קיימים גם כאשר המצב החבוי הוא רב מימדי, אז את הביטויים w^t מחליפות חזקות של המטריצה W_{hidden}, אשר להן התנהגות אסימפטוטית דומה.

קיימים מספר כלים להתמודד עם הקושי המובנה שפעפוע הגרדיאנט דרך רשת נשנית יוצר. חלקם מנסים לפתור את הבעיה על ידי שינוי אלגוריתם האופטימיזציה, למשל בעזרת הקטנת קצב הלמידה כאשר הגרדיאנט גדול מדי, בדומה לאלגוריתמים עם קצב למידה אדפטיביים עליהם למדנו ביחידה 3.

הכלים המוצלחים יותר פותרים את הבעיה על ידי שינוי ארכיטקטורת הרשת: במקום תא אלמן פשוט, הרשת תשתמש בתאים נשנים מתקדמים, המכילים רכיבים דומים לחיבורי הדילוג של רשתות שיוריות. הפופולרי שבהם הוא תא LSTM (Long Short-Term Memory), אשר מכיל גם רכיב בקרת זרימה, המאפשר לתא ללמוד מתי לפתוח ומתי לסגור את חיבור הדילוג, וכן רכיב נלמד המאפשר לתא לאפס את המצב החבוי, למשל לאחר שהוא זיהה נתק סמנטי בין שני חלקי משפט.

רשתות מבוססות LSTM מומשו ב-PyTorch, והפלט שלהן זהה במבנהו לזה של רשת RNN פשוטה. על כן, נוכל להשתמש בהן בנקל, כלהלן:

```
class LSTMclassifier(FasterDeepRNNClassifier):
    def __init__(self, embed_dim, hidden_dim, RNNlayers):
      super().__init__(embed_dim, hidden_dim, RNNlayers)
      self.rnn_stack = nn.LSTM(embed_dim,hidden_dim,RNNlayers)
```

ראו כיצד רשת הסיווג החדשה יורשת מהרשת האחרונה שהגדרנו, בה `rnn_stack` הוא אובייקט מסוג `nn.RNN`. השינוי היחיד שיש לעשות הוא להגדירו מחדש כ-LSTM.

לסיום דיון זה, נציין כי בעוד שרשתות LSTM (או בעלות תאים נשנים דומים) הפגינו את הביצועים הטובים ביותר במשימות עיבוד שפה טבעית עד לשנת 2017, רשתות מבוססות Transformer מפגינות מאז את הביצועים הטובים ביותר. ארכיטקטורת ה-Transformer, בה נדון ביחידת הלימוד הבאה, אינה נשנית ולכן מתחמקת משני האתגרים המרכזיים של שימוש ב-RNN: איטיות החישוב בטור ובעיות הגרדיאנט המתפוצץ/הנעלם.

שאלות לתרגול

1. חשבו את $\left[\frac{\partial h_t}{\partial w}\right]$ במפורש, והראו כי ביטוי זה חסום.

2. הראו כי קיים חסם יחיד המתאים לכל הביטויים מהצורה
$\left(1-h_T^2\right)\left(1-h_{T-1}^2\right)\cdots\left(1-h_t^2\right)\left[\frac{\partial h_{t-1}}{\partial w}\right]$.

3. חשבו את $\frac{\partial h_1}{\partial w}$ בהנחה ש- h_0 הוא וקטור האפס, כנהוג באתחול תא נשנה. האם תוכלו להסביר מדוע לא מופיע ביטוי מהצורה $\left[\frac{\partial h_1}{\partial w}\right]$ לעיל?