פונקציית המחיר ואלגוריתם האימון של הנוירון

פונקציית מחיר

בפרק הקודם ראינו כיצד אפשר להשתמש בנוירון יחיד לסיווג אוסף נתונים לשתי מחלקות, וכעת נלמד כיצד לעשות זאת באופן אוטומטי. ראשית, נבחר **פונקציית מחיר** (cost function) מתאימה, היא תבטא את מידת ההתאמה של המודל הנבחר אל הנתונים בהם נשתמש לאימון המודל. כאשר המודל יתאים לנתונים, קרי יסווגם נכון, המחיר יהיה נמוך וכאשר לא – המחיר יהיה גבוה.

נזכיר שהמודל הנידון לסיווג נקודות שחורות ולבנות מקבל כקלט קואורדינטות של נקודות במרחב דו מימדי, ומחזיר כפלט את הפונקציה

$$y=\frac{1}{1+e^{-(w_0x_0+w_1x_1+b)}}.$$

בהתאם, הפרמטרים של המודל, אותם באפשרותנו לשנות בכדי להתאימו לנתונים הם w_0,w_1,b ולכן פונקציית המחיר תהיה תלויה אך ורק בהם. בבעיות של סיווג נהוג לבחור בפונקציית המחיר של **האנטרופיה הצולבת** (Cross Entropy), אשר עבור הסיווג לשתי מחלקות מוגדרת כך:

$$C(w_0,w_1,b)=-\frac{1}{\#Data}\sum_{(x_0,x_1,y_t)\in Data} y_t\log\left(y(x_0,x_1)\right)+(1-y_t)\log\left(1-y(x_0,x_1)\right).$$

יש לשים לב למספר פרטים בנוסחה זו:

1. אוסף הנתונים שלנו מסומן ב-$Data$, ומספר הדגימות בו ב-$\#Data$.
2. הסכום רץ על כל הדגימות באוסף הנתונים, המיוצגות בתור שלשות מהצורה (x_0,x_1,y_t).
3. y_t הוא הסיווג האמיתי של הדגימה. $y_t=1$ כאשר מדובר בנקודה שחורה, ו-$y_t=0$ כאשר מדובר בנקודה לבנה.
4. $y(x_0,x_1)$ היא ההסתברות שמקנה המודל לכך שהנקודה (x_0,x_1) היא נקודה לבנה. ערך זה הוא פונקציה של פרמטרי המודל וכאן באה לידי ביטוי התלות של פונקציית המחיר בפרמטרים.
5. כל נקודת דגימה מופיעה רק פעם אחת בסכום, ועל כן אפשר לכתוב את הנוסחה בצורה הבאה,

$$C(w_0,w_1,b)=\frac{1}{\#Data}\sum_{(x_0,x_1,y_t)\in Data} H(x_0,x_1,y_t),$$

כאשר

$$H(x_0,x_1,y_t)=-\left[y_t\log\left(y(x_0,x_1)\right)+(1-y_t)\log\left(1-y(x_0,x_1)\right)\right]$$

היא **התרומה למחיר** של נקודת הדגימה (x_0,x_1,y_t).

מטרתנו כעת היא למצוא נקודת מינימום של פונקציית המחיר – ערכי פרמטרים המביאים למינימום את C. המודל שיתקבל עבור ערכים אלו יהיה מוצלח במיוחד, מהסיבה הבאה: בביטוי לתרומה למחיר של אחת מנקודות הדגימה יש סכום של שני איברים, אך אחד מהם תמיד מתאפס – בהתאם לסיווג שלה כלבנה או שחורה. למשל, אם הנקודה היא לבנה, אז $y_t=0$ ולכן הביטוי מצטמצם ל

$$H(x_0,x_1,y_t)=-\log\left(1-y(x_0,x_1)\right).$$

כעת, על הנוירון לסווג נקודה זו כלבנה, ובהתאם ברצוננו לקבל ב-$y(x_0,x_1)$ הסתברות הקרובה ל-0. לו הנוירון היה מסווג נקודה זו בצורה לא נכונה, הביטוי $\log\left(1-y(x_0,x_1)\right)$ היה שלילי וגדול בערכו המוחלט, ולבסוף תרומתה של הנקודה למחיר הייתה חיובית וגדולה. על כן, מודל שיסווג נקודה זו נכונה, ייקנס במחיר נמוך יותר, ומודל מוצלח במיוחד יימנע מסיווגים שגויים מאוד עבור כל הנקודות.

חדי העין ישימו לב שהמודל אותו אנו רוצים לאמן הוא למעשה רגרסיה לוגיסטית, ופונקציית המחיר הנ"ל קשורה באופן הדוק לפונקציית הנראות (Likelihood) של בעיית רגרסיה זו. מציאת נק'

המינימום של C שקולה למציאת אומד הנראות המקסימלית בבעיית הרגרסיה, ונקודת מבט זו מספקת צידוק נוסף לשימוש בה.

בעוד שאת בעיית הרגרסיה הלוגיסטית אפשר לפתור באופן אנליטי, למצוא אומד נראות מקסימלית במדויק ולהוכיח שקיים רק אחד כזה, לרוב המצב עבור רשתות נוירונים הפוך: קיימים מספר ערכי פרמטרים שונים הממזערים את פונקציית המחיר, ואין דרך אנליטית ישירה למציאתם. על כן נרבה להשתמש באלגוריתמים לקירוב נקודת המינימום ולכן יש ערך ללמוד שיטה זו כבר כעת.

אלגוריתם אימון: מורד הגרדיאנט

מורד הגרדיאנט (Gradient Descent) הוא אלגוריתם קלאסי ופשוט למציאת מינימום של פונקציה. בהמשך הקורס נלמד גרסאות מתקדמות שלו עם שכלולים שונים, אשר נועדו להאיץ את פעולתו ולהתחמק מבעיות נפוצות הצצות בעת אימון רשתות נוירונים. בפרק זה, נשתמש בגרסתו הבסיסית.

פעולתו של האלגוריתם מבוססת על כך שבכל נקודה במרחב הפרמטרים, תנועה קטנה בכיוון השלילי של הגרדיאנט ∇C תוביל לירידה בערך פונקציית המחיר, C, ולכן סביר שסדרה של צעדים קטנים כאלו תוביל לנק' מינימום.

על כן, תהליך מציאת הפרמטרים המתאימים ביותר לבעיית הסיווג שלנו יתבסס על שני צעדים, עליהם יש לחזור עד למציאת פתרון סביר. הצעדים הם:

1. חישוב הגרדיאנט עבור ערכי הפרמטרים הנוכחיים $\nabla C\left(w_0, w_1, b\right)$.
2. עדכון ערכי הפרמטרים הנוכחיים: $\left(w_0, w_1, b\right) = \left(w_0, w_1, b\right) - \alpha \nabla C\left(w_0, w_1, b\right)$.

שימו לב ש-α, גודל הצעד בכל שלב הוא פרמטר של האלגוריתם אשר יש לקבוע לפני ריצתו. גודל צעד קטן מדי יוביל לכך שהאלגוריתם לא מתכנס גם לאחר מספר רב של חישובים בעוד שגודל צעד גדול מדי יוביל לאי יציבות בתהליך: ייתכן שבעדכון הפרמטרים "נדלג מעל" נקודת המינימום של C, וכך ערך הפונקציה לא ירד. בהמשך הקורס נקדיש תשומת לב רבה לפרמטר זה.

ובכן, בכדי לממש אלגוריתם זה בקוד, עלינו לדעת כיצד לחשב הגרדיאנט. בעתיד נשתמש בכלי הגזירה האוטומטית של PyTorch, מבלי להידרש לעיסוק בפרטים הקטנים, אך עבור דוגמה פשוטה זו נעשה זאת בעצמנו לצורך תרגול.

ראשית, ניזכר שפונקציית המחיר C היא ממוצע התרומות למחיר של כל אחת מהנקודות הנתונות:

$$C = \frac{1}{\#Data} \sum_{(x_0, x_1, y_t) \in Data} H\left(x_0, x_1, y_t\right).$$

מכך נסיק שמספיק לנו להתרכז בחישוב הגרדיאנט ∇H לכל דגימה, ולבסוף לחשב ממוצע, וזאת מפני שלפי כלל הנגזרת של סכום:

$$\nabla C = \frac{1}{\#Data} \sum_{(x_0, x_1, y_t) \in Data} \nabla H.$$

כעת, עלינו לחשב את הגרדיאנט $\nabla H = \left(\frac{\partial H}{\partial w_0}, \frac{\partial H}{\partial w_1}, \frac{\partial H}{\partial b}\right)$ אך הפונקציה H תלויה בפרמטרי המודל דרך האקטיבציה $y = \frac{1}{1+e^{-z}}$ אשר בתורה תלויה בהם דרך האגרגציה $z = w_0 x_0 + w_1 x_1 + b$, דהיינו H היא הרכבה של פונקציות:

$$H\left(w_0, w_1, b\right) = H\left(y\left(z\left(\left(w_0, w_1, b\right)\right)\right)\right).$$

במקרה זה כלל השרשרת יקל עלינו את החישוב, שכן:

$$\frac{\partial H}{\partial w_0}=\frac{\partial H}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w_0}$$
$$\frac{\partial H}{\partial w_1}=\frac{\partial H}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w_1}$$
$$\frac{\partial H}{\partial b}=\frac{\partial H}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial b}$$

שימו לב שבנוסף לפירוק החישוב האנליטי של נגזרות אלו לשלבים פשוטים, בשימוש בכלל השרשרת יש גם יתרון חישובי: נוכל לחשב את הנגזרות $\frac{\partial H}{\partial y},\frac{\partial y}{\partial z}$ פעם אחת בלבד בכל איטרציה של האלגוריתם, ולכפול אותן בשלוש הנגזרות של z בכדי לקבל את ∇H.

על ידי חישוב ישיר נסיק כי:

$$\frac{\partial H}{\partial y}=-\left(\frac{y_t}{y}-\frac{1-y_t}{1-y}\right)=-\frac{y_t-y}{y-y^2}$$
$$\frac{\partial y}{\partial z}=\frac{e^{-z}}{\left(1+e^{-z}\right)^2}=\frac{1}{1+e^{-z}}\frac{e^{-z}}{1+e^{-z}}=y\left(1-y\right)$$
$$.\frac{\partial z}{\partial w_0}=x_0,\quad \frac{\partial z}{\partial w_1}=x_1,\quad \frac{\partial z}{\partial b}=1$$

לאחר חישוב זה יש בידינו את כל הדרוש בכדי לממש את האלגוריתם בקוד.

שאלות לתרגול

1. המשיכו את חישוב הגרדיאנט ∇H ומצאו ביטוי פשוט עבור כל אחד מרכיביו.
2. פונקציית הנראות של מודל הסתברותי הנועד לחיזוי משתנה תלוי מבטאת כמה "סביר" המודל בהנתן הנתונים – כמה הפרמטרים הנבחרים מתאימים לנתונים. בבעיית סיווג בינארי, אשר בה יש לחזות את ערך המחלקה $y=0$ או $y=1$, פונקציית נראות פופולרית במקרה זה נתונה על ידי הביטוי

$$.L\left(Model\right)=\prod_{Data}y^{y_t}\left(1-y\right)^{1-y_t}$$

כאשר הכפל מתבצע על כל הנקודות הנתונות, y_t הוא הסיווג האמיתי של נק' נתונה אחת ו y הוא הסיווג שהמודל חוזה עבור נקודה זו.

א. הסבירו מהן ההנחות שיש להניח על אוסף הנתונים ודרך איסופם בכדי לקבל פונקציית נראות שכזו. זכרו שהצורה הכללית של הנראות היא

$$,L\left(Model\right)=P\left(Data\,|\,Model\right)$$

קרי – ההסתברות לדגום את אוסף הנתונים מהמודל.

רמזים:

1. זכרו שכאשר שני מאורעות הם בלתי תלויים מתקיים $.P\left(A\ and\ B\right)=P\left(A\right)P\left(B\right)$
2. שימו לב ש $y^{y_t}\left(1-y\right)^{1-y_t}=\begin{cases}y & y_t=1\\ 1-y & y_t=0\end{cases}$, זכרו ש- y היא ההסתברות החזויה על פי המודל שהנקודה הנתונה היא שחורה ו- $y_t=1$ כאשר הנקודה היא באמת שחורה.

ב. הוכיחו שהמקסימום של פונקציית נראות זו מתקבל באותה נקודה בה מתקבל המינימום של פונקציית המחיר האנטרופיה הצולבת.
רמז: הפעילו את פונקציית הלוגריתם על הנראות.