

Perceptron

תיאור

מודל המשתמש בנוירון אחד של Mcp, על כן יכול ללמוד רק פונקציות הניתנות להפרדה לינארית. במידה והפונקציה אותה אנחנו מנסים ללמוד אכן ניתנת להפרדה לינארית תהליך הלמידה יסתיים תוך מספר סופי של איטרציות. במידה ואיננה ניתנת להפרדה לינארית הפרספטרון לא יצליח למצוא הפרדה ומסיבה זו לא יעצור. במידה והאלגוריתם הוגדר מראש לבצע מספר סופי של איטרציות אזי האלגוריתם יחזיר הפרדה שאינה מייצגת את הפונקציה.

מאפיינים

- Single-layer.
- עבור בעיות הניתנות להפרדה לינארית בלבד.
- מבצע סיווג בינארי (קלסיפיקציה).
- לומד באופן איטרטיבי.
- שימוש בפונקציית אקטיבציה (threshold)
- זיהוי צורות \ זיהוי תבניות \ ניבוי \ סיווג \ דחיסת מידע \ זיהוי קול וכו'.

אלגוריתם

- וקטור המשקלים: $w = (w_1, \dots, w_m)$
- נאתחל אותו לאפסים, או מספרים רנדומלים נמוכים.
- וקטור קלט: $x = (x_1, \dots, x_m)$
- Output target: y_{target}

בכל איטרציה נבצע שני שלבים:

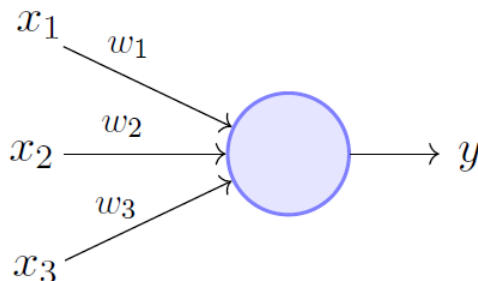
שלב 1: נבצע חישוב

$$y_{output} = \sum_{j=1}^m x_j w_j$$

שלב 2: נעדכן את המשקולות אם ה- y_{target} שונה מה- y_{output} שקיבלנו לאחר החישוב.

$$w_{new} = w_{old} + \alpha * (y_{target} - y_{output}) * x$$

ארכיטקטורה



Perceptron Model (Minsky-Papert in 1969)

Adaline	
מאפיינים	<ul style="list-style-type: none">Single-layer.עבור בעיות הניתנות להפרדה לינארית בלבד.מבצע סיווג בינארי (קלסיפיקציה).לומד באופן איטרטיבי.שימוש בפונקציית אקטיבציה (threshold)זיהוי צורות \ זיהוי תבניות \ ניבוי \ סיווג \ דחיסת מידע \ זיהוי קול וכו'.
אלגוריתם	
הערות	<ul style="list-style-type: none">נרצה להגיע לנק' מינימום של השגיאה, אבל מכיוון שקשה לנו לגזור ולהשוות ל-0, בצורה איטרטיבית מנסים ללכת לכיוון בו הגרדיאנט קטן.לכן, משתמש באלגוריתם BP לירידה בגרדיאנט.אפשר להיתקע במינימום מקומי.תמיד מתכנס אבל עם שגיאה.

Feed Forward Network and Back Propagation

תיאור

רשת נוירונים יודעת ללמוד כל פונקציה בוליאנית וכל פונקציה רציפה כאשר לרשת לפחות 3 רמות. הרשת בזמן תהליך הלימוד מסוגלת לסדר את עצמה ע"י עדכון המשקלים. מכיוון שאנחנו משתמשים במספר רב של נוירונים סיבוכיות החישוב עולה ועל מנת לצמצם אותה נשתמש בחישוב של "ירידה בגרדיאנט". חישובים אלה דורשים שהפונקציה בה אנו משתמשים בנוירונים תהיה גזירה. הפונקציה שמוגדרת במבנה הבסיסי של Mcp אינה גזירה (מכיוון שאינה רציפה) ועל כן נשנה את המבנה של הנוירון ונחליף את הפונקציה בפונקציית הסיגמוייד אשר פחות דומה למבנה של הנוירון במח.

שימוש בחישוב של "ירידה בגרדיאנט" יוצר לנו בעיה העלולה להביא אותנו למינימום לוקאלי ולהיתקע בו, בכך לא נגיע למינימום הגלובאלי שהוא הפיתרון האופטימאלי.

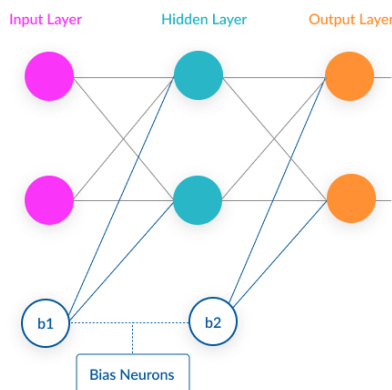
מאפיינים

- Feed Forward – לימוד הקלט במערכת.
- Back Propagation – חישוב טעות הלמידה בכל אחת מהשכבות על ידי חלחול אחורה.
- זיהוי צורות \ זיהוי תבניות \ ניבוי \ סיווג \ דחיסת מידע \ זיהוי קול וכו'.

פרמטרים

1. α – קצב לימוד (ניתן לשינוי במהלך הריצה)
2. תנאי עצירה – כאשר הטעות קטנה.
3. Bipolar – נשתמש בביפולאר (-1 \ 1) כי אם נקבל 0 לא נלמד כלום.
4. פונקציית אקטיבציה – נשתמש בסיגמוייד מכיוון שהיא קלה לגזירה: $f(x) = \frac{1}{1+e^{-x}}$
5. מומנט – מזרח התכנסות ומאפשר מניעה של בעיית המינימום הלוקאלי.

ארכיטקטורה



Hopfield

מאפיינים

- זיהוי אסוציאטיבי.
- רשת שכולם מחוברים לכולם.
- מודל אסינכרוני.
- משתמש בפונקציית אנרגיה.

אלגוריתם

$$E(x) = -\frac{1}{2} \sum_i \sum_j x_i w_{ij} x_j$$

פונקציית האנרגיה:

הוכחת הופילד

טענה: פונקציית האנרגיה תמיד יורדת וחסומה מלמטה.

הוכחה: מכיוון שרק ה- x_i משתנים, נוציא גורם משותף ולכן

$$E^{new} - E^{old} = -(x_i^{new} - x_i^{old}) \left(\sum_{j=1}^n w_{ij} x_j - \theta_i \right)$$

נחלק למקרים –

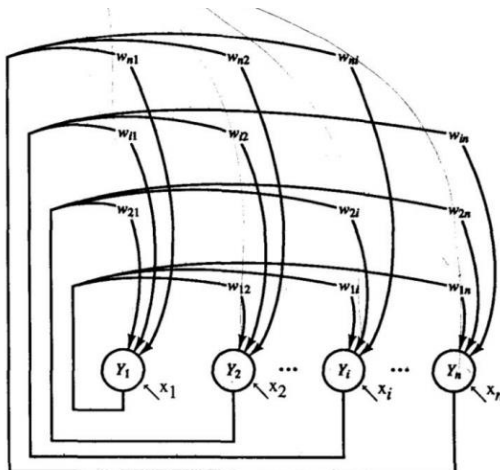
$$1. \quad \text{אם } x_i^{new} > x_i^{old}, \quad \text{אז } \sum_{j=1}^n w_{ij} x_j > \theta_i$$

$$\text{לכן } E^{new} - E^{old} = (-)(+)(+) = E^{new} < E^{old}$$

$$2. \quad \text{אם } x_i^{new} < x_i^{old}, \quad \text{אז } \sum_{j=1}^n w_{ij} x_j < \theta_i$$

$$\text{לכן } E^{new} - E^{old} = (-)(-)(-) = E^{new} < E^{old}$$

ארכיטקטורה



Hamming Net

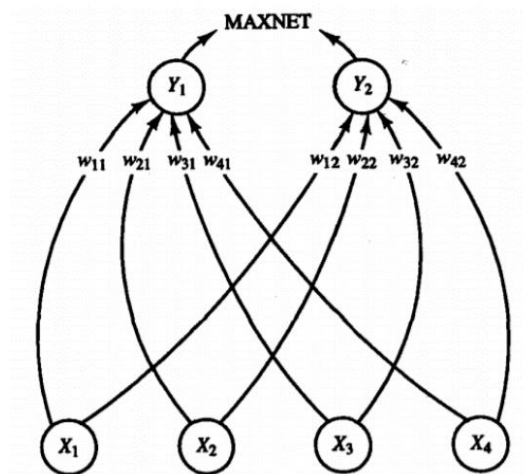
מאפיינים

- סיווג של אובייקטים למחלקות (Clustering)
- מכיל אבטיפוסים.
- משתמש ב-Hamming distance כדי לחשב מרחק.
- משתמש ב-Max Net כדי למצוא מקסימום התאמה.

: Max Net

- מודל סינכרוני למציאת ערך מקסימום.
- רשת שבה כולם מחוברים לכולם.

ארכיטקטורה



Kohonen

מאפיינים

- Zero-class
- מיפוי טופולוגי.
- סיווג של אובייקטים למחלקות (Clustering)
- זיכרון אסוציאטיבי (?)

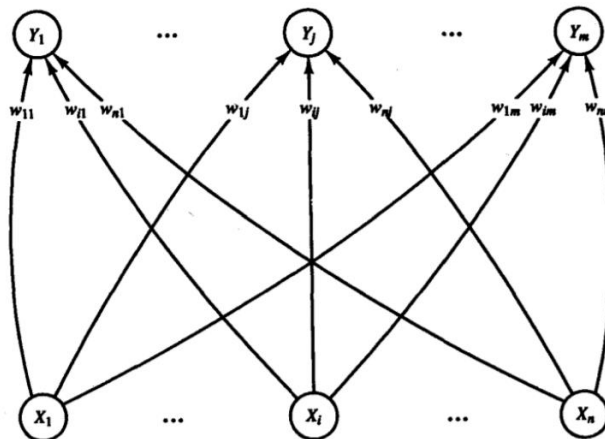
אלגוריתם

1. מתחילים במשקלים רנדומלים קטנים בין 0 ל- 1.
2. מקבלים קלט ומוצאים את הנירון הכי קרוב אליו.
3. נבחר את ה- w הכי קרוב ל- x ונעדכן את w .
4. נעדכן גם את הסביבה של ה- w הנבחר.

בעיות והערות

1. נרצה לוודא שלא רק אזור אחד יזוז בכל פעם ויהיו ניירונים בחלק אחר שישארו במקום. מקרה זה עלול לקרות כאשר הקלט שמתקבל תמיד מוצא נירון קרוב אליו באותו חלק מסוים. איך נמנע זאת? על ידי 'מצפון'. בעזרת מצפון זה נסמן את אותו נירון שנבחר, ונוכל לקבוע שלאחר מספר בחירות מסוים ניתן לו "לנוח" והוא לא יבחר יותר, כך ייבחר הנירון שאחריו.

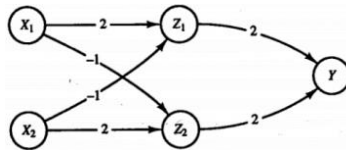
ארכיטקטורה



1. הראה כי בעזרת Perceptron ניתן לייצג כל פונקציה בוליאנית.

ניתן לייצג AND, OR, NOT בעזרת נורון אחד.
בעזרת DNF ניתן לייצג כל פונקציה בוליאנית.
לכן ניתן לממש XOR שמשוואתו היא:

$$x_1 XOR x_2 \leftrightarrow (x_1 AND NOT x_2) OR (x_2 AND NOT x_1)$$



2. מה ההבדלים בין פרספטרון לנוירון McCulloch-Pitts?

- המשקלים והסף לא חייבים להיות זהים.
- המשקולות יכולות להיות חיוביות או שליליות.
- פונקציית התוצאה נעה בין $[-1, 1]$ ולא $[0, 1]$.
- יש כלל למידה.

3. מי מבין הרשתות מסוגלת לבצע זיהוי צורות (זיהוי תבניות)?

- Perceptron
- Adaline
- Feed Forward

4. מי מבין הרשתות משוייכת לזיהוי אסוציאטיבי?

- Hopfield
- BAM

5. מי מבין הרשתות מסוגלת לבצע Clustering?

- Hamming Net
- Kohonen

6. מה ההבדל העיקרי (הפונקציונלי) בין Hamming Net ובין Kohonen?

שני המודלים מבחינה פונקציונלית מחפשים את המשקל הכי קרוב, אך:

- קוהונן עושה מיון טופולוגי שבעזרתו הוא מבצע Clustering עבור קבוצה של ערכים.
- רשת האמינג פשוט מחזירה את האבטיפוס.

7. עבור Back Propagation – מתי הוא לא יתכנס בצורה אופטימלית? מדוע?

- במידה ונקבל קלט שיש לו כמה מינימומים בנוסף למינימום הגלובאלי, במהלך הלמידה אנו עלולים להיתקל במינימום לוקאלי ובכך נקבל שגיאה.
- המינימום הגלובאלי הוא הפיתרון האופטימלי.

הסיבה לכך שלא נקבל פיתרון אופטימלי היא מכיוון ש-BP מתקן את הטעויות שלו בעזרת ירידה בגרדיינט. הרשת מגרילה משקלים רנדומליים והקלט נכנס באופן רנדומלי, יכול להיות מצב שנרד בגרדיינט ונתכנס למינימום לוקאלי.

8. איך ניתן להימנע מבעיית המינימום הלוקאלי של BP?

- הרצה כמה פעמים של המודל ובחירת המודל הכי טוב.
- הוספת מומנט למודל, בהתחלה נתחיל באלפא גדולה ובעזרת המומנט ברגע שהיה שינוי נקטין את הצעד עד שיתכנס. בנוסף לאורך הזמן נקטין את אלפא.
- הרצה של המודל עם פרמטרים שונים בצורה רנדומלית שונה.

9. הסבר מהו האלפא, ה-Step size והמומנט?

- אלפא – קובע את קצב הלימוד.
- Step size – קובע את הצעד שנצטרך לקחת באמצעות הכפלת האלפא בגרדיאנט כפול הטעות שצריך לתקן.
- מומנט – עוזר לנו בהתחלה להתחיל עם קצב לימוד מאוד גדול וכך במעט איטרציות להתכנס למינימום, ברגע שסימן הגרדיאנט משתנה אנחנו מקטינים את אלפא וכך נתכנס למינימום. המומנט עוזר לנו להתכנס במעט איטרציות ולהימנע ממינימום לוקאלי.

10. מהם ה-Hidden Layers ואיך הם תורמים?

-

11. אילו מבין הרשתות הבאות יצליחו ללמוד XOR כך שתהיה התכנסות BP, Adaline, Perceptron?

נשים לב ש-XOR לא ניתנת להפרדה לינארית.

- Perceptron – לא ניתן ללמוד את הפונקציה XOR מכיוון שלא ניתן להפריד אותה לינארית.
- Adaline – ניתן לייצג כל פונקציה עם Adaline אך עם טעות, מכיוון שהאלגוריתם תמיד מתכנס. בעזרת השימוש בירידה בגרדיאנט הוא עלול להגיע למינימום לוקאלי, כך מתכנס אך לא מגיע לפיתרון האופטימלי.
- BP – ניתן לייצג בצורה טובה מכיוון שניתן להחליט את מספר הפרמטרים וכך נוכל לייצג כל פונקציה שנרצה.

12. מהו המונח Equiprobable? עבור איזו רשת הוא רלוונטי?

- עבור רשת קוהון.
- שימוש ב-Equiprobable דואג לכך שפריסת הנירונים במרחב תתאים לפריסת הנתונים במרחב.

13. מהו מצפון? עבור איזו רשת המצפון רלוונטי?

- רשת קוהון משתמשת במצפון.
 - מצפון נועד לפתור את הבעיה שתוארה בשאלה הקודמת (11) תחת המונח Equiprobable.
 - קוהון משתמש בשני פרמטרים נוספים שמשתנים: רדיוס השכנים וקצב הלמידה.
- מכיוון שהקלט בקוהון הוא רנדומלי, עלולה להיווצר בעיה בה הנירונים שיזוזו יהיו באיזור מסוים, כך שאיזור אחר מוזנח. מצפון נועד למנוע את הבעיה בכך שלאחר שנירונ יבחר מספר פעמים (אנחנו נחליט כמה) הוא לא יבחר בפעם הבאה ובכך תהיה התפלגות שווה.

Lior Daniel

Lior7daniel@gmail.com

[illegible]