סיכום שיעור 28: סיווג XOR באמצעות פרספטרון

Lesson 28: XOR Classification with Perceptron

ד"ר יורם סגל

January 2026

גרסה 1.0

מילות מפתח: פרספטרון, נוירון, סיווג לינארי, XOR, פונקציית אקטיבציה, Gradient Descent, משקלים, Sigmoid, רשתות נוירונים, Deep Learning

---

1 מבוא

שיעור זה עוסק במעבר מהתיאוריה לפרקטיקה בתחום הלמידה העמוקה (Deep Learning). המטרה העיקרית היא להבין את הפרספטרון כיחידה הבסיסית של רשתות נוירונים, ולהבין מדוע בעיית ה-XOR הייתה מכשול משמעותי בהתפתחות התחום.

2 הפרספטרון – האטום של הלמידה העמוקה

2.1 מהנוירון לפרספטרון

הפרספטרון הוא היחידה האטומית הבסיסית של כל רשת נוירונים. ההבדל בין נוירון לפרספטרון:

- **נוירון:** מבצע חיבור לינארי בלבד – מכפלה משוקללת (Scalar Product). הוא לינארי לחלוטין ומוגבל ביכולת הפתרון שלו.
- **פרספטרון:** מוסיף פונקציית אקטיבציה (Activation Function) שמכניסה אי-לינאריות ומאפשרת קבלת החלטה בינארית.

2.2 המשוואה המתמטית

משוואת הנוירון הבסיסית:

$$z = \sum_{i=0}^{n} w_i \cdot x_i = w_0 \cdot x_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots \tag{1}$$

כאשר:

- $x_0 = 1$ תמיד (ה-Bias)
- $w_0 = b$ – ה-Bias שמזיז את קו ההפרדה
- w_i – המשקלים (חשיבות כל תכונה)
- x_i – הפיצ'רים (התכונות)

הייצוג הווקטורי: מכפלה סקלארית של וקטור משקלים כפול וקטור פיצ'רים.

2.3 פונקציית האקטיבציה

פונקציית האקטיבציה (Activation Function) מחליטה האם הנקודה נמצאת מעל הקו או מתחתיו:

$$f(z) = \begin{cases} 1 & \text{אם} z > 0 \\ 0 & \text{אם} z \leq 0 \end{cases} \tag{2}$$

3 סיווג לינארי ובעיית ה-ROX

3.1 עקרון הסיווג הלינארי

המטרה: להעביר קו (או היפרפליין במימדים גבוהים) שמפריד בין שתי קבוצות נתונים.

נקודה חשובה: אם יודעים להפריד בין 2 קבוצות, יודעים להפריד גם בין 100 קבוצות – פשוט מפרידים 99 מול 1, ואז ממשיכים.

3.2 משמעות המשקלים

המשקלים מייצגים את **חשיבות** כל תכונה. לדוגמה, אם רופא בודק לחץ דם, גובה ומשקל:

- לחץ דם – משקל גבוה (קריטי להחלטה)
- גובה – משקל נמוך (פחות משמעותי)

4 ההוכחה שה-ROX אינו ניתן לפתרון בפרספטרון בודד

4.1 טבלת האמת של XOR

טבלה 1: טבלת האמת של שער XOR

x_1	x_2	**XOR**
0	0	0
0	1	1
1	0	1
1	1	0

4.2 ההוכחה בשלילה

נניח שקיים פרספטרון שמצליח לסווג את ה-XOR. נציב את התנאים:

1. עבור $(0,0)$: נדרש $b < 0$ (כדי לקבל 0)
2. עבור $(0,1)$: נדרש $w_2 + b > 0$ (כדי לקבל 1)

3. עבור $(1,0)$: נדרש $w_1 + b > 0$ (כדי לקבל 1)
4. עבור $(1,1)$: נדרש $w_1 + w_2 + b < 0$ (כדי לקבל 0)

נחבר את אי-שוויונים 2 ו-3:

$$w_1 + w_2 + 2b > 0$$

אבל מתנאי 1 אנחנו יודעים ש-$b < 0$, ומתנאי 4 אנחנו יודעים ש-$w_1 + w_2 + b < 0$.
זוהי **סתירה** – ולכן לא קיים פרספטרון בודד שפותר את ה-XOR.

5 הפתרון: רשת רב-שכבתית

הפתרון לבעיית ה-XOR הוא הוספת **שכבה נוספת**. כל שכבה:

1. מוצאת קו הפרדה
2. שואלת: "האם מעל הקו או מתחת?"
3. מעבירה את התוצאה לשכבה הבאה

תובנה מרכזית: התגובות של שכבה אחת הופכות לפיצ'רים של השכבה הבאה!

6 Gradient Descent – אלגוריתם האופטימיזציה

6.1 העיקרון

- מטרה: למצוא את המשקלים שממזערים את פונקציית השגיאה
- שיטה: תנועה בכיוון ההפוך לנגזרת (ירידה במורד ההר)

6.2 נוסחת העדכון

$$w_{new} = w_{old} - \alpha \cdot \nabla E(w) \tag{3}$$

כאשר:

- α – קצב הלמידה (Learning Rate)
- $\nabla E(w)$ – הגרדיאנט של פונקציית השגיאה

6.3 פונקציית הסיגמואיד

בגלל שפונקציית הסימן אינה רציפה ולא ניתנת לגזירה, משתמשים בקירוב רציף – **פונקציית הסיגמואיד**:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

תכונות:

- כאשר $x = 0$: $\sigma(0) = 0.5$
- כאשר $x \to +\infty$: $\sigma(x) \to 1$
- כאשר $x \to -\infty$: $\sigma(x) \to 0$

7 מושגים חשובים נוספים

7.1 One-Hot Encoding

כאשר יש קטגוריות (למשל: אדום, צהוב, כחול), במקום לתת להן מספרים (1, 2, 3), יוצרים עמודה נפרדת לכל קטגוריה עם ערכים 0 או 1.

הסיבה: לשמור על מספרים בטווח $[0, 1]$ ולמנוע התבדרות של החישובים.

7.2 Epoch ו-Batch

- **Epoch:** מעבר אחד על כל הדוגמאות ב-Dataset
- **Batch:** קבוצת דוגמאות שמעובדות ביחד לפני עדכון המשקלים

7.3 שיטת Adam

אלגוריתם אופטימיזציה שמתחיל עם צעדים קטנים ומקטין אותם ככל שמתקדמים – למניעת "קפיצה" מעל המינימום.

8 תובנות מרכזיות

נקודות חשובות לזכור:

1. **הדבר החשוב ביותר ב-Deep Learning הוא פונקציית השגיאה** – היא המצפן שמנחה את הלמידה.
2. **תהליך אקראי אינו ניתן לחיזוי** – אם הנתונים אקראיים לחלוטין, לא ניתן לבנות מודל.
3. **דיוק של 50% הוא הגרוע ביותר** – זה שווה לזריקת מטבע.
4. **דיוק של 0% מעיד על היפוך התוויות** – פשוט הופכים את התשובות.
5. **נוירון = ייצוג גרפי של משוואה מתמטית**

9 רקע היסטורי

- בשנות ה-60: הפרספטרון פותח והציג תקווה גדולה
- 1969: מינסקי ופייפרט הוכיחו שפרספטרון בודד לא פותר XOR
- "חורף ה-AI": תקופה ארוכה שבה המחקר קפא
- שנות ה-2000: פריצת דרך עם GPU ועיבוד מקבילי
- כינוי "Deep Learning" הופיע כדי לעקוף את הסטיגמה של "רשתות נוירונים"