פרק 6: רשתות עצביות

6.1 מבוא

6.1.1 רשתות עצביות מלאכותיות

רשתות עצביות מלאכותיות (Artificial Neural Networks) הן כלי בסיסי בתחום הלמידה הממוחשבת. למעשה, מדובר במשפחה רחבה למדי של רשתות בעלות מבנים שונים. בקורס נדון בסוג הנפוץ ביותר, הנקרא פרספטרון רב שכבתי (Multilayer Perception), או רשת היזון-קדמי (Feedforward NN).

עקרונית, רשת עצבית הינה צירוף של רכיבים חישוביים *פשוטים* ("נוירונים"), בעלי אופי אנלוגי, אשר צירופם יוצר *מיפוי לא לינארי* בין משתני הכניסה למשתני היציאה. תכונות בסיסיות של רשתות אלו הינן:

(1) מבנה מודולרי ויכולת גידול, המאפשרים יצירת מיפויים מסובכים כנדרש.

(2) מקביליות גבוהה, המתבטאת בזמן חישוב מהיר.

(3) יכולת לימוד.

יכולת הלימוד מסתמכת על האפשרות לכיוונון פרמטרי הרשת על מנת לקרב פונקציות לא-לינאריות שונות. בעזרת אלגוריתמים מתאימים, הרשת מסוגלת לבצע כיוונון זה באופן אוטומטי, בהסתמך על "דוגמאות" של המיפוי הנדרש.

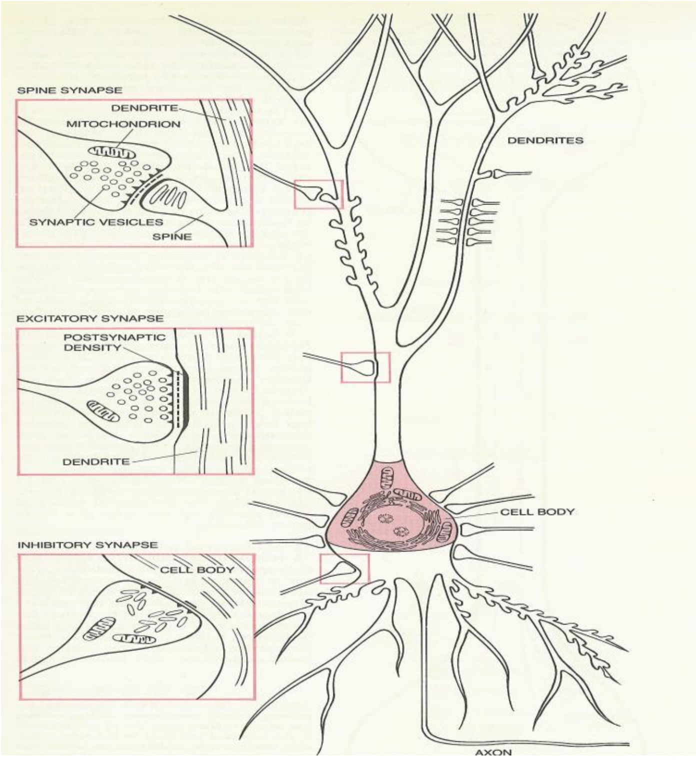
6.1.2 רקע ביולוגי

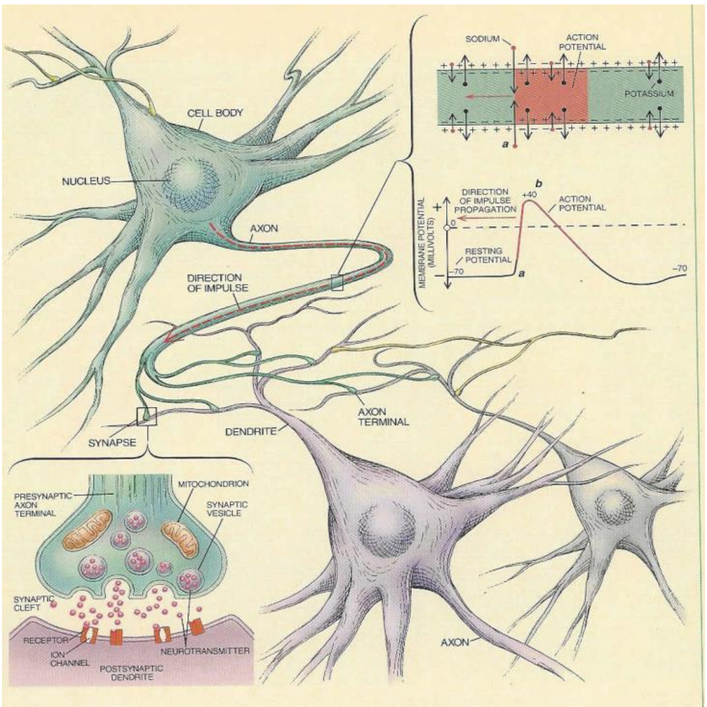
התפתחותן של רשתות הנוירונים המלאכותיות ככלי חישובי הושפעה במידה מרובה מההיבט הביולוגי. המוח ומערכת העצבים המרכזית בנויים מתאי-עצב, הנוירונים, בעלי קישוריות גבוהה. מבנה של נוירון אופייני נראה בציור 1.

ניתן להבחין במרכיבים הבאים:

1. Dendrites : "ערוצי הכניסה" מנוירונים אחרים.
2. גוף התא: המרכיב ה"חישובי".
3. אקסונים (Axons) וחיבורים סינפטיים: ערוצי היציאה לנוירונים אחרים.

נציין כי אורכו הכולל של נוירון מסוג זה הינו כ- 0.1 מ"מ.





ציור 1 : נוירון ביולוגי

יחידה בסיסית	תא עצב	שער לוגי
מספר	10^{11}	10^{7}
מספר הקשרים	10^{15}	10^{8}
יחידות זמן	10^{-3}	10^{-10}
איחסון כולל	10^{16}	10^{12}
פעולות חישוביות (כולל)	10^{16}/sec	10^{12}/sec

טבלת השוואה (מקורבת) : המוח לעומת המחשב

מרבית הנוירונים מקודדים את יציאתם על ידי סדרה של פולסי מתח (action potential) הנוצרים כאשר הסכום המצטבר של הכניסות לנוירון עובר סף מסוים. פולסים אלה נוצרים בגוף הנוירון, ומתקדמים לאורך האקסונים באמצעות תהליכים אלקטרו-כימיים.

המידע הטמון במערכת הביולוגית מתבטא בקשרים בין הנוירונים ובחוזקם. הדעה הרווחת היא כי למידה מתבטאת בעיצוב קשרים אלה.

מספר נתונים כמותיים לגבי המוח האנושי :

- במוח האדם כ- 10^{11} נוירונים.
- נוירון ממוצע מחובר ל- 10^{4} אחרים.
- זמן תגובה של נוירון הינו כ- 10^{-3} שניות.

זמן התגובה של נוירון בודד הינו איטי באופן מפתיע. ניתן להשוותו, למשל, לזמן תגובה של טרנזיסטור מהיר שהוא כ- 10^{-10} שניות (לעת עתה ...). ברור לפיכך כי כוחו החישובי של המוח נובע מאופיו המקבילי והקישוריות בין רכיביו, ולא ממספר רב של חישובים טוריים. זאת בניגוד מוחלט לאופיו הטורי של המחשב הספרתי! אבחנה זו הניעה חוקרים להתבונן במבנים חישוביים בעלי מאפיינים דומים – רשתות הנוירונים המלאכותיות.

<u>לימוד הביאני (Hebian Learning)</u>

מנגנון הלימוד הנוירו-פסיכולוגי עדיין אינו מובן לגמרי. עקרון לימוד חשוב הוצע בהקשר זה על ידי הנוירו-פסיכולוגי Hebb ב- 1949. עקרון זה הינו :

- באם שני נוירונים הקשורים ביניהם מופעלים בסמיכות זמנים, הקשר ביניהם מתחזק.

Hebb הציע עקרון זה בהקשר של זכרון אסוציאטיבי, כאשר ה"המטרה" היא לחזק קשרים מוצלחים. כיום קיימת עדות פיסיולוגית לכך שבאזורים מסוימים במוח אכן מתקיימים תהליכי למידה מסוג זה.

מנגנוני הלמידה שנפתח בהמשך הפרק לא יהיו מסוג זה, אם כי ניתן למצוא הקבלה מסוימת.

6.2 הפרספטרון

נעבור עתה לדיון ברשתות ניירונים מלאכותיות מהסוג הנפוץ ביותר, המכונה לעתים בשם פרספטרון רב-שכבתי. בסעיף זה נציג את הרכיב הבסיסי ברשת כזו, דהיינו הפרספטרון הבודד.

הערה לגבי הסימון : בפרק זה סדרת הלימוד תסומן ע"י $\{x_k, d_k\}_{k=1}^{n}$, דהיינו התויות הרצויות יסומנו על ידי d_k (ולא y_k). הסימון y יישמר למוצא הפרספטרון או רשת הניירונים.

6.2.1 הגדרת הפרספטרון

נוירון בודד מסוג פרספטרון הינו רכיב חישובי המממש את הקשר הבא בין כניסותיו ליציאתו (ציור 2) :

$$y = \varphi\left(\sum_{i=1}^{d} w_i\, x_i + w_0\right) \qquad (6.1)$$

כאשר :

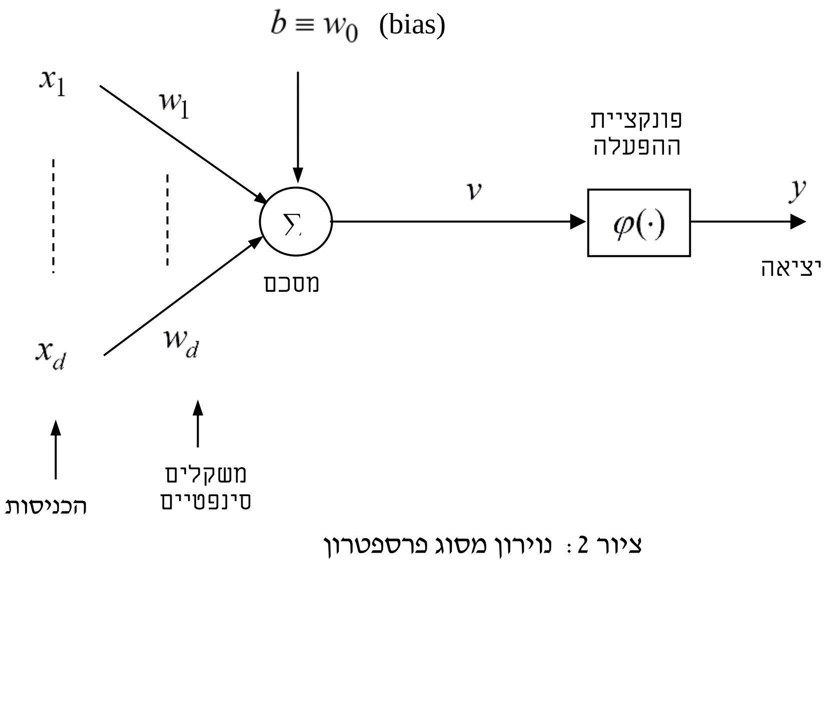
$x_1, \ldots, x_d$ - משתני הכניסה.

y - היציאה

$w_1, \ldots, w_d$ - פרמטרים הנקראים משקלים סינפטיים (synaptic weights).

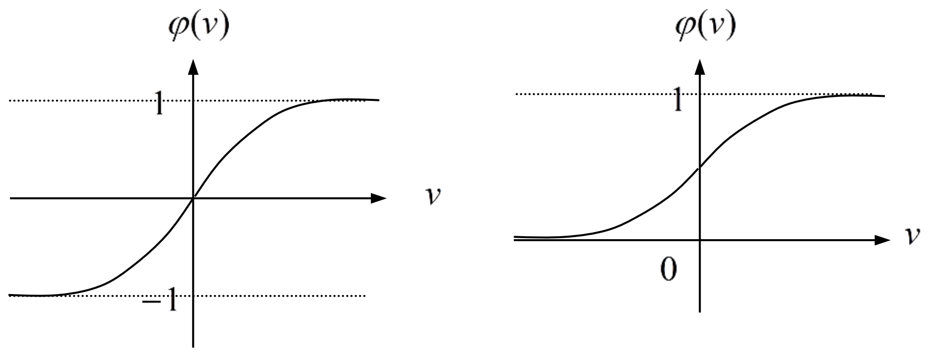
w_0 - פרמטר הטיה (bias). לעיתים מסומן ע"י b.

φ - פונקציה לא לינארית, הנקראת פונקצית ההפעלה (Activation function).



ציור 2 : נוירון מסוג פרספטרון

צורה אופיינית לפונקצית ההפעלה מודגמת בציור 3. פונקציה זו מאופיינת בהיותה לא לינארית, מונוטונית עולה, ובעלת טווח חסום. פונקציה בעלת צורה כזו נקראת לעתים פונקצית Sigmoid , או פונקציית-S. מקרה פרטי אחר הוא פונקציה לינארית: $\varphi(v)=\alpha v$. בהמשך נראה כי לאי-לינאריות של φ חשיבות מרובה ביותר.



$$\varphi(v)=\tanh\left(\frac{av}{2}\right)=\frac{1-e^{-av}}{1+e^{-av}} \qquad \varphi(v)=\frac{1}{1+e^{-av}}$$

ציור 3 : פונקציות הפעלה אופייניות – "פונקציה הלוגיסטית" וטנגנס היפרבולי.

וקטור הפרמטרים של הנוירון הינו, אם כן,

$$w=\left(w_0, w_1, \ldots, w_d\right)^T .$$

שינוי פרמטרים אלה משנה את הפונקציה שמייצג הנוירון.

נוח לרשום את היחס (6.1) באופן מקוצר, על ידי רישום וקטורי. לשם כך נגדיר וקטור כניסה קבוע (1) בקואורדינטה הראשונה:

$$x=(1, x_1, \ldots, x_d)$$

ונקבל

$$y=\varphi(v)=\varphi\left(w^T x\right)$$

<u>כח הייצוג של הפרספטרון הבודד</u>: לפי הגדרתו, הפרספטרון הבודד מוגבל לפונקציה לינארית של הכניסות, עד כדי העיוות הלא-לינארי של פונקצית ההפעלה ביציאה. בפרט, בהקשר של סיווג, משטח ההחלטה המתקבל על ידי $\mathrm{sign}\{y\}$ הוא על-מישור כאשר φ היא פונקציית הזהות.

6.2.2 אלגוריתם הגרדיאנט לכיוונון פרמטרי הפרספטרון

אלגוריתם הלימוד הנפוץ ביותר לכיוונון פרמטרי הפרספטרון מבוסס על על פונקצית מחיר ריבועית ואלגוריתם הגרדיאנט (Gradient Descent).

כרגיל, נניח כי נתונה סדרת לימוד של דוגמאות מתויגות $\{x_k, d_k\}_{k=1}^n$, כאשר $x_k \in \mathbb{R}^d$ ואילו d_k תג סקלרי (דיסקרטי או רציף). נגדיר את פונקצית המחיר הריבועית:

$$E(w) = \frac{1}{2}\sum_{k=1}^{n} e_k^2$$

כאשר

$$e_k = d_k - y_k = d_k - \varphi\left(w^T x_k\right).$$

מטרתנו למצוא וקטור פרמטרים w שמביא למינימום את המחיר $E(w)$, ומשיג בכך התאמה מיטבית לדוגמאות.

לצורך חישוב הנגזרות בהמשך, נדרש מעתה כי פונקצית ההפעלה $\varphi(\cdot)$ הינה **גזירה**. אלגוריתם הגרדיאנט מוגדר, באופן כללי, באופן הבא:

אלגוריתם הגרדיאנט (גרסת Batch):

איתחול: וקטור משקלים w_0.

איטרציה: עבור $t = 0,1,2,\ldots$:

$$w_{t+1} = w_t - \eta_t \left.\frac{\partial E(w)}{\partial w}\right|_{w = w_t}$$

או בקיצור:

$$w := w - \eta_t \frac{\partial E(w)}{\partial w} \qquad (*)$$

את הגרדיאנט $\frac{\partial E(w)}{\partial w}$ ניתן לחשב בקלות כלהלן:

$$\frac{\partial E(w)}{\partial w} = \sum_{k=1}^{n} \frac{\partial e_k}{\partial w} e_k$$

כאשר

$$\frac{\partial e_k}{\partial w} = -\frac{\partial \hat{y}_k}{\partial w} = -\varphi'\left(w^T x_k\right) x_k$$

מכאן:

$$\frac{\partial E(w)}{\partial w} = -\sum_{k=1}^{n} \varphi'\left(w^T x_k\right) x_k e_k$$

משוואת האלגוריתם (*) הינה, לפיכך :

$$\boxed{w := w + \eta_t \sum_{k=1}^{n} \varphi'\left(w^T x_k\right) \; e_k \, x_k}$$

<u>התכנסות:</u> פונקצית המחיר $E(w)$ אינה קמורה במקרה הכללי. כתוצאה מכך אלגוריתם הגרדיאנט מבטיח רק התכנסות למינימום מקומי. התכנסות למינימום הגלובלי מובטחת במקרה הפרטי שבו φ לינארית (כיוון שאז פונקציית השגיאה הינה פונקציה ריבועית, ולכן קמורה, של המשקלים) וגודל הצעד קטן דיו.

<u>* 6.2.3 אלגוריתם הלימוד של רוזנבלט (Rosenblatt, 1958)</u>

אלגוריתם הלימוד הבסיסי המקובל כיום לרשתות ניירונים הינו אלגוריתם מבוסס-גרדיאנט, ואותו נתאר בהמשך. מבחינה היסטורית, אלגוריתם הלימוד הראשון שעורר עניין רב בפרספטרון כרכיב מסווג הוצע עוד ב-1958, ונקרא "אלגוריתם אימון הפרסטרון" (Perceptron Training Rule). אלגוריתם זה היה הראשון שעבורו ניתנה הוכחת התכנסות (בתנאים שנזכיר בהמשך). למען השלמות נתאר אותו בקצרה.

נניח כי נתון אוסף דוגמאות מסווגות $\{x_k, d_k\}_{k=1}^{n}$, כאשר $d_k \in \{-1,+1\}$. מטרתנו לסווג דוגמאות אלו באופן נכון (כלומר, בהתאם לתויות) באמצעות פרספטרון מהצורה :

$$y = \varphi_{HL}\left(w^T x\right) \cdot \begin{cases} -1: & w^T x < 0 \\ +1: & w^T x \geq 0 \end{cases}$$

פרספטרון זה משתמש בפונקצית הפעלה מסוג Hard Limiter. (כפי שציינו בסעיף הקודם, אין יתרון בשימוש בפוקצית הפעלה (מונוטונית) אחרת לצורך סיווג בינארי עם פרספטורן יחיד.) מטרתנו לקבוע את וקטור המשקלים w.

<u>תהליך הלימוד:</u>

איתחול : וקטור הפרמטרים מאותחל בערך w_0 כלשהו.

בכל שלב $t = 1, 2, \ldots$:

- בחרו דוגמא אחת x_t ממאגר הדוגמאות $\{x_k\}_{k=1}^{n}$.
- חשבו את יציאת הפרספטרון עבור דוגמא זו, עם וקטור המשקלים הנוכחי w_t :

$$y_t = \varphi_{HL}\left(w_t^T x_t\right)$$

- עדכנו את וקטור המשקלים :

$$w_{t+1} = w_t + \eta\left(d_t - y_t\right)x_t$$

הערות :

1. η הינו קבוע חיובי, הנקרא "קצב הלימוד", "ההגבר" או "גודל הצעד".

2. ההפרש $d_t - y_t$ הינו השגיאה המניעה את האלגוריתם. כאשר $d_t - y_t = 0$ ("הרצוי והמצוי מתלכדים") אין שינוי בוקטור המשקלים. אלגוריתמים כגון זה, שאינם מעדכנים את הפרמטרים כאשר אין שגיאה או כאשר יש ניבוי מוצלח, מכונים פסיביים או קונסרבטיביים.

<u>משפט התכנסות הפרסספטרון:</u> *נניח כי אוסף הדוגמאות $\{x_k, d_k\}_{k=1}^n$ <u>ניתן להפרדה לינארית</u>. נניח גם כי כל אחת מהדוגמאות נבחרת מספר בלתי-חסום של פעמים (כלומר, אינסוף חזרות על סט הדוגמאות, אם כי הסדר אינו חייב להיות קבוע). אזי אלגוריתם לימוד הפרספטרון המתואר לעיל מתכנס בתוך <u>מספר סופי של צעדים</u> לוקטור משקלים w^* שמסווג נכונה את כל הדוגמאות.*

החיסרון המרכזי בתוצאה זו הינו כמובן ההגבלה לסט דוגמאות הניתן להפרדה לינארית. כמו כן אין כל אבטחה לגבי זמן ההתכנסות, או לביצועי האלגוריתם כאשר לא ניתן לבצע הפרדה לינארית מושלמת.

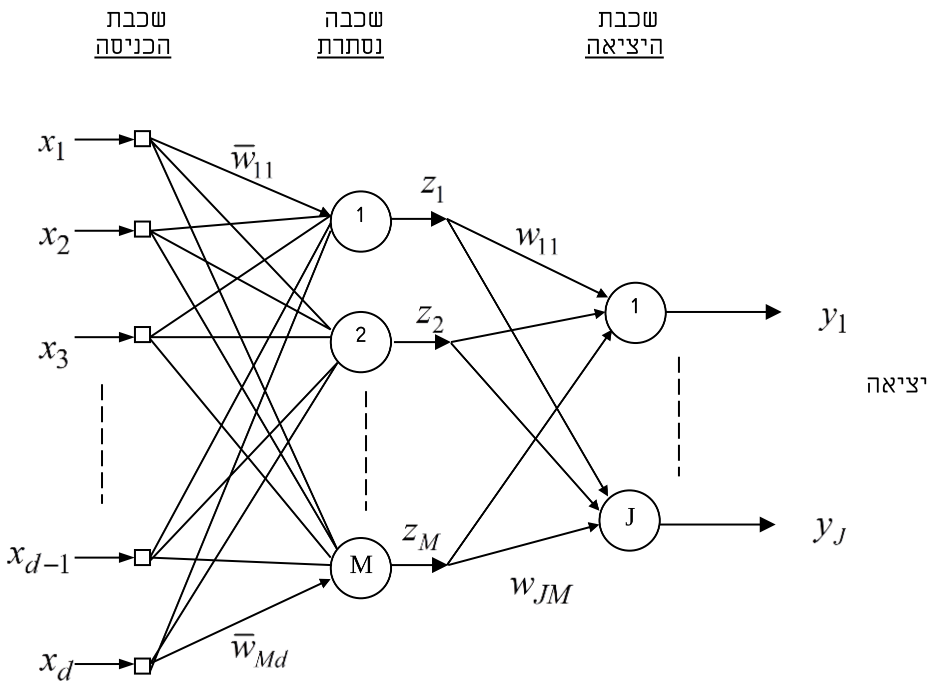
<u>6.3 פרספטרון רב-שכבתי</u>

נעבור עתה לדון ברשתות עצביות מורכבות יותר, מסוג Feedforward Networks (רשתות היזון-קדמי). רשתות אלו כוללות מספר "שכבות" של נוירונים מסוג פרספטרון, כאשר כל שכבה מזינה את השכבה הבאה אחריה. בפרט, אין היזון חוזר לשכבות הקודמות. רשתות אלו נקראות גם "פרספטרון רב-שכבתי" (Multi-layer Perceptron).

רשתות רב-שכבתיות פותרות את בעיית כוח הייצוג המוגבל של פרספטרון חד-שכבתי.

<u>6.3.1 מבנה וסימון פרספטרון רב-שכבתי</u>

רשת עצבית אופיינית מסוג פרספטרון רב-שכבתי מתוארת בציור הבא. לרשת זו שלוש שכבות :



ציור 4 : רשת עצבית תלת-שכבתית.

- שכבת הכניסה איננה אלא נקודת פיצול של הכניסות לרשת.
- שכבת הביניים כוללת פרספטרונים המוזנים ישירות על ידי הכניסות $(x_1,\ldots,x_d)$. יציאת פרספטרון m בשכבת הביניים נתונה על ידי:

$$Z_m = \varphi_m(\bar{w}_{m0} + \sum_{i=1}^{d} \bar{w}_{mi} x_i) \equiv \varphi_m(\bar{w}_m^T X)$$

כאשר $\bar{w}_m = (\bar{w}_{m0},\ldots,\bar{w}_{md})^T$ הינו וקטור המשקלים עבור ניורון זה, ואילו $X = (1, x_1,\ldots,x_d)^T$. נציין כי פונקציית האקטיבציה φ_m עשויה להיות שונה מניורון לניורון, אולם לרוב היא זהה לכל הניורונים בשכבה נתונה.

- שכבת היציאה מוזנת על ידי יציאות הפרספטרונים בשכבת הביניים. יציאת פרספטרון j בשכבת היציאה נתונה על ידי

$$y_j = \varphi_j(w_{j0} + \sum_{m=1}^{M} w_{jm} z_m) \equiv \varphi_j(w_j^T Z)$$

כאשר $w_j = (w_{j0},\ldots,w_{jM})^T$ הינו וקטור המשקלים עבור ניורון זה, ואילו $Z = (1, z_1,\ldots,z_M)^T$.

נציין כי רשת כללית אינה מוגבלת לשלוש שכבות, ובמקרים רבים קיימת שכבה נסתרת נוספת. לשם פשטות נתמקד פה במקרה של שלוש שכבות, וההכללה למספר כלשהו תהיה ברורה. ברשת המצוירת קישוריות מלאה בין הנוירונים בשכבות העוקבות (כלומר קיים קשר בין כל ניירון לכל הניירונים בשכבה העוקבת), אולם קיים עניין גם ברשתות בעלות קישוריות חלקית.

נציין כי רשת ניירונים מסוג זה עשויה לשמש הן לרגרסיה (קרוב פונקציה בעלת טווח יציאה רציף) והן לסיווג. מספר הדגשים ספציפיים לכל מקרה:

<u>רגרסיה:</u>

- לקרוב פונקציה חד-מימדית ($y=f(x)\in\mathbb{R}$) מספיק כמובן ניורון יציאה אחד. עבור פונקציה רב מימדית ($y=f(x)\in\mathbb{R}^N$) יידרשו כמובן N ניירונים ביציאה.
- יש להקפיד כי פונקצית ההפעלה בשכבת היציאה תהיה בעלת טווח תואם לטווח הנדרש עבור משתני היציאה. בחירה מקובלת הינה פונקצית ההפעלה לינארית (ביציאה בלבד!).

<u>סיווג:</u>

- סיווג ל- K מחלקות מקובל לבצע על ידי $J=K$ ניירוני יציאה, כאשר כל ניירון משויך לאחת המחלקות, והסיווג מתבצע לפי הניירון בעל הערך המכסימלי (winner takes all). חוק החלטה זה מקביל לסיווג לפי "פונקציות דיסקרימינציה".
- אימון הרשת במקרה כזה יתבצע עם <u>ערכי ייחוס של 1 ביציאה המתאימה למחלקה הנכונה, ו-0 (או 1-) בשאר היציאות</u>. טווח פונקצית האקטיבציה ביציאה צריך לפיכך לכלול את האינטרוול [0,1]. בחירה אפשרית הינה פונקציה אקטיבציה לינארית (ביציאה בלבד!). אפשרות אחרת הינה לנרמל את היציאות באופן הבא, כך שיתאימו לפילוג הסתברות על פני המחלקות:

$$y_j \equiv \varphi_j(Z) = \frac{\exp(w_j^T Z)}{\sum_{j=1}^{J}\exp(w_j^T Z)} .$$

<u>דוגמא לרשת נוירונים המשמשת לזיהוי כתב-יד</u>

הרשת הבאה היתה אחת מהראשונות ששימשו בהצלחה ביישום מעשי – במקרה זה זיהוי ספרות אוטומטי:

548 LeCun, Boser, Denker, Henderson, Howard, Hubbard, and Jackel

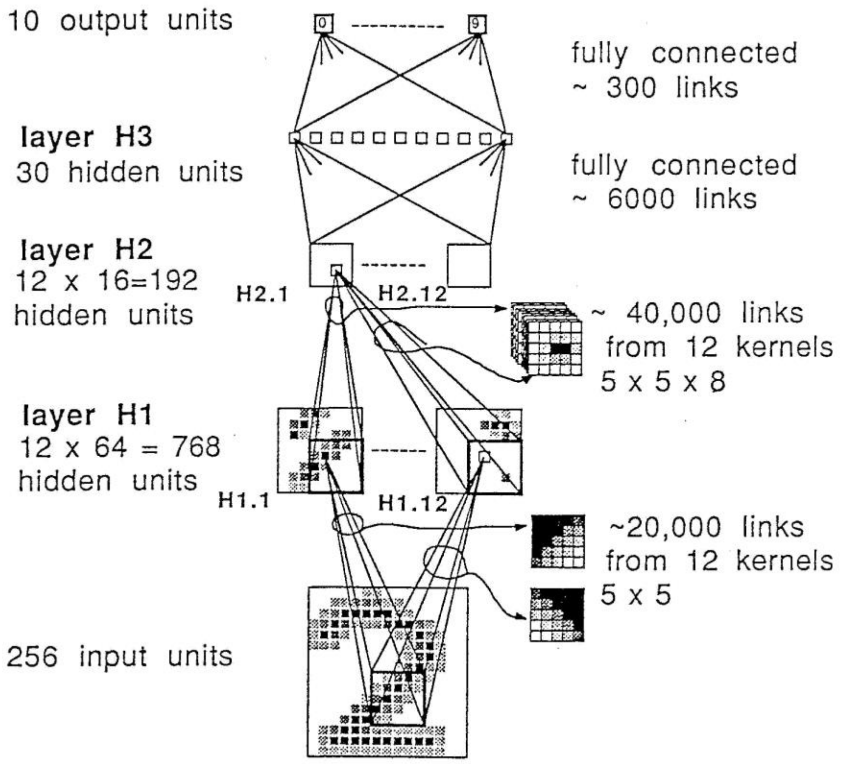


Figure 3: Log mean squared error (MSE) (top) and raw error rate (bottom) versus number of training passes.

ניתן לראות כי שתי השכבות הראשונות בעלות מבנה מוגדר, ומשמשות לחישוב מאפיינים מקומיים על פני התמונה. כל ניירון בשכבה H1, למשל, מחובר לאזור בגודל 5×5 בתמונת הכניסה. לכל הניירונים בקבוצה H1,j יש משקלים זהים בכניסה, אשר מהווים למעשה גרעין קורלציה (מסנן).

שתי השכבות האחרונות, לעומת זאת, הינן בעלות חיבוריות מלאה, עם מקדמים הניתנים לכיוונון באופן בלתי תלוי. בסה"כ יש ברשת 1256 יחידות (ניירונים), 64660 קישורים, ו- 9760 פרמטרים בלתי תלויים.

6.3.2 כוח הייצוג של רשת רב-שכבתית

כפי שראינו, פרספטרון חד-שכבתית מהצורה $y=\varphi\left(\sum w_i x_i\right)$ מוגבל ביכולת הייצוג והקרוב שלו : יציאתו קבועה בכיוון המאונך ל- w , ובבעיות סיווג הוא משרה משטחי הפרדה לינאריים. לעומת זאת, רשת רב-שכבתית מאפשרת לקרב פונקציה רציפה כלשהי (כמודגם בציור להלן).

קיימות מספר תוצאות מתמטיות בנושא זה, פה רק נציין (ללא הוכחה) אחת מתוצאות אלו :

טענה ("משפט הקרוב האוניברסלי"): נניח כי פונקצית ההפעלה φ היא פונקציה **לא-פולינומיאלית**. תהי $f_0(x)$ פונקציה רציפה על קוביית היחידה :

$$f_0:[0,1]^d\to\Re,\qquad x=\left(x_1,\ldots,x_d\right)$$

אזי ניתן לקרב את f_0 בקרוב טוב כרצוננו על ידי הביטוי :

$$f(x)=\sum_{m=1}^{M} w_m\ \varphi\left(\sum_{i=1}^{d}\ \bar{w}_{mi}\,x_i+\bar{w}_{m0}\right)\qquad(*)$$

כלומר : לכל $\varepsilon>0$, ניתן למצוא קבועים M , $\left(\bar{w}_{mi},w_m\right)$ כך שיתקיים :

$\left|f(x)-f_0(x)\right|\le\varepsilon$, לכל $x\in[0,1]^d$.

נשים לב כי הביטוי (*) מתאר רשת עצבית בעלת שכבה נסתרת אחת, ושכבת יציאת בעלת ניירון לינארי (בודד). תוצאה זו מראה כי רשת עצבית בעלת שכבה נסתרת אחת היא מקרב אוניברסלי (לפונקציות רציפות). בפרט, רשת עצבית כזו מאפשרת מימוש תחומי החלטה (רציפים) כלשהם על ידי חוק החלטה מהצורה $y\equiv f(x)\lessgtr 0$.

הערות נוספות:

- הפונקציה $\varphi(\cdot)$ ניתנת לבחירה כרצוננו כל עוד היא מקיימת את דרישות המשפט. נציין כי :

  (1) $\varphi(v)=\dfrac{1}{1+e^{-v}}$ מקיימת הדרישה.

  (2) $\varphi(v)=\operatorname{sgn}(v)$ מקיימת הדרישה.

  (3) $\varphi(v)=v^3$ אינה מקיימת.

- קיימים בספרות חסמים שונים על גודל הרשת הנדרשת, התלויים ב"קצב השינוי" של הפונקציה f_0 . קצב שינוי זה ניתן לכימות באחת משתי דרכים :
  - חסמים על נגזרות f_0 .

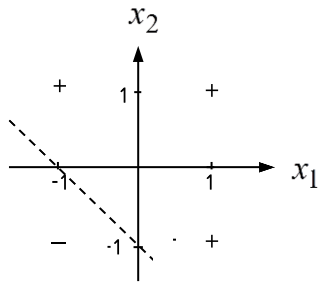
- חסמים בתחום התדר, על תכולת התדרים הגבוהים של f_0 .

- שכבה נסתרת אחת אינה בהכרח הבחירה האופטימלית מבחינת מספר הנוירונים הנדרשים ויכולת הלימוד.

אימון רשתות: נעיר שקיימים מספר אלגוריתמי אימון לרשתות עצביות. האלגוריתם הבסיסי נקרא Back Propagation (פעפוע אחורי) ומבוסס על העברת השגיאה לאחר בין השכבות השונות. האלגוריתם הוא אלגוריתם גרדיאנט ולא מובטחת התכנסות אלא למינימום לוקאלי.

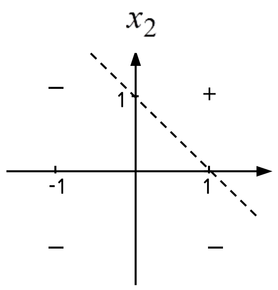
* <u>6.3.3 הפרספטרון כרכיב לוגי</u>

כאשר כניסות הפרספטרון הן בינאריות (נניח 1+ עבור '1' לוגי, 1- עבור '0' לוגי), ניתן לראות את הפרספטרון כרכיב לוגי. יציאה בינארית ניתן להגדיר באמצעות $\varphi(v)=\operatorname{sgn}(v)$. המימוש של פונקציות AND, OR, ו- NAND עבור כניסה דו-מימדית מודגם בציור הבא.



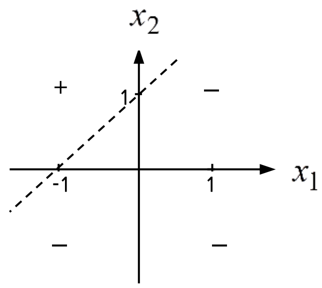
$y = x_1 \vee x_2 \quad \text{(OR)}$

$w_1 = w_2 = 1, \; w_0 = 1$



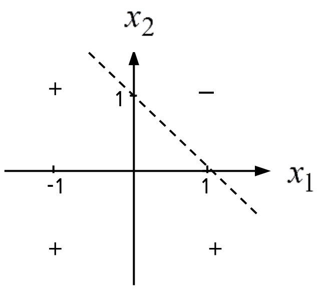
$y = x_1 \wedge x_2 \quad \text{(AND)}$

$w_1 = w_2 = 1, \; w_0 = -1$



$y = \overline{x_1} \wedge x_2$

$w_1 = -1, w_2 = 1, w_0 = -1$



$y = \overline{x_1 \wedge x_2} \quad \text{(NAND)}$

$w_1 = w_2 = -1, \; w_0 = 1$

מימוש פונקציות לוגיות באמצעות פרספטרון.

חשוב לציין כי פונקצית XOR אינה ניתנת למימוש על ידי פרספטרון יחיד.

<u>לכניסה רב מימדית</u>:

- ניתן לממש בצורה דומה פונקצית AND: $y = x_1 \wedge x_2 \wedge \ldots x_m = \overset{m}{\underset{i=1}{\Lambda}} x_i$
  באמצעות הנוסחה: $\mathrm{AND}(x) = sign(\sum_{j=1}^{m} x_j - (m-1))$.
- כן ניתן לממש את פונקצית OR: $y = x_1 \vee \ldots \vee x_m = \overset{m}{\underset{i=1}{\vee}} x_i$
  באמצעות הנוסחה $\mathrm{OR}(x) = sign(\sum_{j=1}^{m} x_j + (m-1))$.
- בנוסף ניתן להפוך לוגית (NOT) כל אחת מהכניסות x_j בנפרד, ע"י הפיכת סימן w_j.
- באופן כללי יותר, ניתן לממש בעזרת פרספטרון בודד כל פונקציה לוגית מהצורה "k מתוך m", כלומר $y=1$ כאשר <u>לפחות</u> k מתוך m הכניסות הן 1+. הפונקציות AND, OR הן מקרים פרטיים (עבור איזה k?).

נעיר כי ניתן לעבור מהתחום $x_j \in \{-1,1\}$ לתחום $\tilde{x}_j \in \{0,1\}$ באמצעות ההצבה $x_j = 2\tilde{x}_j - 1$.

<u>פרספטרון רב-שכבתי</u>: למרות המגבלה על הפרספטרון הבודד, הרי שעל ידי צרוף מספר פרספטרונים בינריים במספר שכבות ניתן לממש פונקציה לוגית כלשהי. למעשה, נדרשות שתי שכבות באחת מהצורות הקנוניות:

א) DNF (Disjunctive Normal Form):

$$y = T_1 \vee T_2 \vee \ldots \vee T_n$$

כאשר T_k מהצורה $T_k = \overset{m}{\underset{i=1}{\wedge}} z_i$, ואילו $x_i = z_i$ או $\bar{x}_i$.

ב) CNF (Conjunctive Normal Form):

$$y = T_1 \wedge T_2 \wedge \ldots \wedge T_n$$

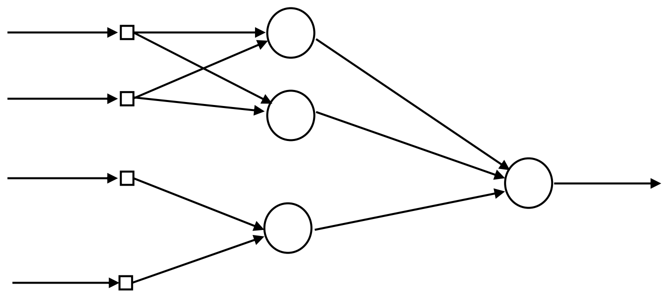
כאשר T_k מהצורה $T_k = \overset{m}{\underset{i=1}{\vee}} z_i$, ו- z_i כמו קודם.

6.3.4 הערות לגבי מבנה רשת הנוירונים ותהליך הלימוד

בכל יישום לא-טריוויאלי של רשל ניוריונים נדרשת התאמה מושכלת של רשת הנוירונים ליישום, מבחינת סוג הרשת, גודלה (מספר הנוירונים), מבנה וארגון (מספר שכבות וכו'), תהליך הלימוד, מספר הדוגמאות וארגונן, ובחירת הכניסות. נזכיר פה "על קצה המזלג" מספר שיקולים ותופעות בהם יש להתחשב. נתחיל בשיקולים לגבי מבנה הרשת.

<u>(1) גודל הרשת:</u> "יש לבחור את הרשת גדולה מספיק, אך לא גדולה מדי". רשת קטנה מדי לא תוכל לקרב בדיוק מספיק את המיפוי הנדרש, ואילו רשת גדולה מדי תמנע לימוד יעיל (זהו ה-Bias-Variance Tradeoff). רשתות רב-שכבתיות אופיניות הן בעלות שכבה נסתרת אחת או שתיים, ומספר הנוירונים בשכבות הנסתרות במקרים רבים אינו עולה על 10 (אלא אם כן קיימת חלוקה מודולרית למספר תת-רשתות, ראו להלן). מספר הניורונים בשכבות הכניסה והיציאה נקבע על ידי מספר הכניסות והיציאות. כיוונון מספר הנוירונים ומבנה הרשת יעשה לרוב אמפירית, תוך שימוש בסדרת ביקורת (Validation) או באמצעות אימות-צולב (Cross-Validation). כלל אצבע הוא כי המספר הכולל של הפרמטרים לא יעלה על עשירית ממספר הדוגמאות בסדרת הלימוד.

<u>(2) ארכיטקטורת הרשת וקישוריות פנימית</u>: כאשר הבעיה ניתנת לחלוקה רצוי לפצל את הרשת למספר תת-רשתות, או לבחור רשת בעלת קישוריות חלקית. לדוגמא, ברשת שבציור הכניסות מחולקות לשתי קבוצות, המטופלות על ידי נוירונים שונים, וממוזגות באחרון. הורדת קישורים לא נחוצים תאיץ את תהליך הלימוד, ותמנע כניסה למינימום מקומיים שאינם מתאימים למבנה הבעיה (ראו דוגמא ברשת זיהוי הכתב).



ציור 7 : קישוריות חלקית

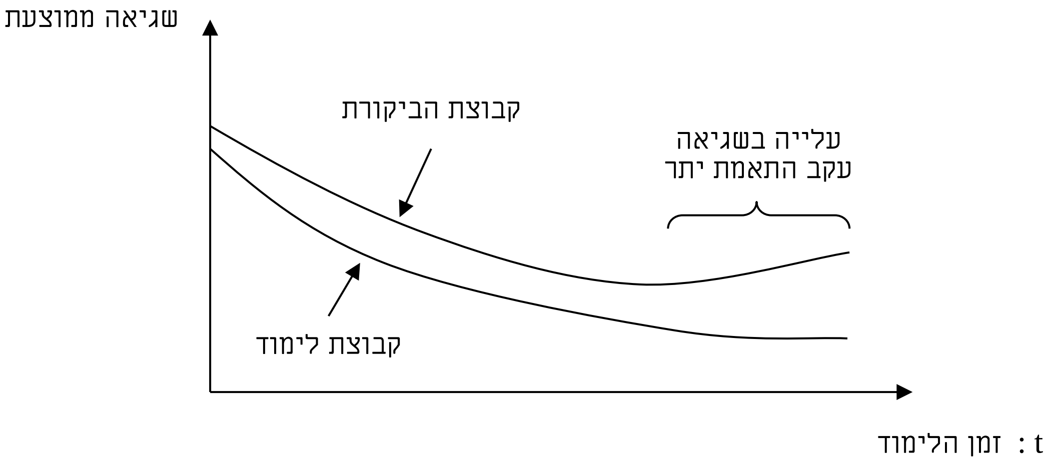
<u>(3) בחירת הכניסות ומאפינים</u>: יש לזכור את החשיבות הרבה בהגדרת מאפינים מתאימים על סמך הכניסות הגולמיות והזנתם לרשת במקום או בנוסף לכניסות הגולמיות.
מציאת המאפיינים המתאימים היא במידה רבה תפקיד המתכנן, על בסיס הבנת הבעיה, בתמיכה של שיטות חישוביות שונות, שבחלקן ניגע בהמשך.

<u>(4) יציאות השכבה הנסתרת כמאפיינים:</u> ברשת רב שכבתית, ניתן לראות ביציאת השכבות הנסתרות "מאפינים" של הכניסה, כלומר ייצוגה במימד נמוך יותר. זאת כאשר מספר הנוירונים בשכבה הנסתרת, קטן יותר ממספר הכניסות. ניתן לשכלל אבחנה זו לצורך מיצוי מאפיינים באמצעות רשת עצבית.

נתייחס עתה בקצרה למספר נקודות הקשורות לתהליך הלימוד.

(5) התאמת יתר: מכיוון שהתאמה מדויקת של גודל הרשת ("סדר המודל") היא קשה לביצוע, לרוב הרשת הנבחרת גדולה יחסית וקיימת סכנה של התאמת יתר, אותה יש למנוע בתהליך הלימוד.

כפי שכבר הזכרנו, דרך מקובלת למנוע התאמת יתר היא להשתמש בקבוצת הביקורת כדי לעצור את תהליך הלימוד כאשר השגיאה האמפירית (ביחס לקבוצת הביקורת) מגיעה למינימום. גודל מקובל לקבוצת הבקורת הינו כ- 1/3 כלל הדוגמאות (לימוד+ביקורת).
נציין כי בעית התאמת-היתר פחות משמעותית כאשר מספר הדגימות השונות (N) גדול מאד. כלל אצבע ל"גדול מאד": $N > 30W$, כאשר W מספר פרמטרי הרשת (עבור לימוד batch).



ציור 8 : התאמת-היתר (Over-fitting)

(6) רגולריזציה: שיטה אפשרית למניעת התאמת יתר גם כאשר הרשת גדולה הינה באמצעות רגולריזציה, כלומר יצירת העדפה לפרמטרים הנותנים פונקציה פשוטה (חלקה). הגישה הפשוטה ביותר הינה להגדיר פונקציה שגיאה (או מחיר) חדשה על ידי הוספת איבר ריבועי, כלהלן:

$$E_\lambda(\theta) = E(\theta) + \frac{\lambda}{2} \| \theta \|^2$$

$E(\theta)$ היא השגיאה האמפירית הרגילה, ואילו האיבר השני (איבר הרגולריציה) מטיל "קנס" על גודל המקדמים, ולפיכך גורם להעדפה של מקדמים קטנים. (ברשת ניורונים הקטנת המקדמים אכן מובילה לפונקציה חלקה יותר – עד כדי פונקציה לינארית למקדמים מספיק קטנים). λ הינו קבוע חיובי השולט על המשקל היחסי של איבר הרגולריציה (ביחס לאיבר השגיאה המקורית). הגרדיאנט המתקבל עתה הינו:

$$\frac{\partial E_\lambda(\theta)}{\partial \theta} = \frac{\partial E(\theta)}{\partial \theta} + \lambda \theta$$

לפיכך אלגוריתם הגרדיאנט ביחס למחיר $E_\lambda(\theta)$ הינו:

$$\theta := \theta - \eta \frac{\partial E(\theta)}{\partial \theta} - \eta \lambda \theta$$

כיוונון פרמטר הרגולריזציה λ נעשה לרוב באמצעות סדרת Validation או Cross-Validation.

<u>(7) התכנסות למינימום מקומי:</u> עקב אי-הלינאריות של רשת הניירונים, לפונקצית השגיאה יהיו בדרך כלל נקודות מינימום מקומי רבות, ואלגוריתם הגרדיאנט יתכנס לאחת מהן. אין ערובה מראש כי נקודה זו קרובה לאופטימום, אולם הניסיון מלמד כי האלגוריתם מתכנס לרוב לנקודה סבירה. בכל מקרה, רצוי להריץ את אלגוריתם הלימוד מספר פעמים עם תנאי התחלה שונים, ולבחור את התוצאה הטובה ביותר. ניתן גם להכניס שיפורים שונים באלגוריתם הגרדיאנט – למשל איבר אנרציה המונע עצירה במינימום מקומי "רדוד" (ראה פרק האופטימיזציה).

<u>(8) אלגוריתמי אופטימיזציה משופרים</u>: אלגוריתם הגרדיאנט, ואלגוריתם ה- BP הנגזר ממנו, הוא אלגוריתם האופטימיזציה הפשוט ביותר. קיימים בספרות אלגוריתמים משופרים עבור רשתות עצביות, אשר מאפשרים התכנסות מהירה יותר. בפרט, אלגוריתמים סטוכסטיים מונעים לעיתים היתקעות במינימום מקומי.

נתאר עתה מספר *"כללי אצבע"* שעשויים לשפר ולהנחות את תהליך הלימוד.

<u>(9) עיבוד מקדים:</u> רצוי לבצע עיבוד מקדים על אוסף הדוגמאות $\{x_k\}$ כך שנקבל:

א. מירכוז: $$\bar{x}_i \equiv \frac{1}{n}\sum_{k=1}^{n} x_{ki} = 0$$

זאת נקבל ע"י הורדת הממוצע מכל דוגמת קלט: $x_{ki} := x_{ki} - \bar{x}_i$.

א. נירמול: שונות (ווריאנס) דומה בכל אחת מהכניסות (למשל 1) – ע"י נרמול הכניסות (הממורכזות) בסטית התקן האמפירית: $\hat{\sigma}_i^2 = \frac{1}{n}\sum_{k=1}^{n}\left(x_{ki} - \bar{x}_i\right)^2, \quad x_{ki} := \frac{x_{ki}}{\hat{\sigma}_i}$.

ב. (אופציה) חוסר-קורלציה בין הכניסות השונות:

$$\frac{1}{n}\sum_{k=1}^{n}\left(x_{ki} - \bar{x}_i\right)\left(x_{kj} - \bar{x}_j\right) = 0$$ (עבור $i \neq j$)

זאת ניתן לבצע בעזרת אלגוריתמי PCA שנלמד בהמשך.

<u>(10) תנאי התחלה</u>: מקובלת בחירה אקראית של המשקלים ההתחלתיים (על מנת לקבל גיוון בין הרצות שונות), תוך הקפדה על כך שיציאות הניורונים יהיו בתחום הלינארי שלהם (ניורון הנמצא ברוויה ילמד לאט). בפרט, נבחן נוירון בעל N כניסות, עם פונקצית ההפעלה בעלת "טווח דינמי" a. נניח כי כל כניסה x_j לניורון היא בעלת וריאנס σ_x^2 (למשל 1, לאחר ביצוע נרמול וממוצע 0). במקרה זה יש לבחור כל משקל w_j בסדר גודל של $\sigma_w = \sqrt{a/N\sigma_x^2}$ – למשל לפי פילוג גאוסי עם ממוצע 0 ווריאנס σ_w^2, או לפי פילוג אחיד בתחום $[-\sigma_w, \sigma_w]$.

הסבר : עבור N כניסות בלתי תלויות בעלות שונות σ_x^2 ומשקלים קבועים $w_j = \sigma_j$, הסכום $\sum_j w_j x_j$ יהיה בעל שונות $\sigma_w^2 \cdot N\sigma_x^2 = a^2$, כלומר בסדר גודל של a.

נציין כי ערך מקובל להגבר (גודל צעד) אלגוריתם הגרדיאנט הינו $\eta \approx 0.1$.

(11) מספר הדוגמאות: לבעיות סיווג, קיים הקשר המקורב הבא לגבי מספר הדוגמאות (n) הנחוץ על מנת לקבל שגיאת הכללה ε :

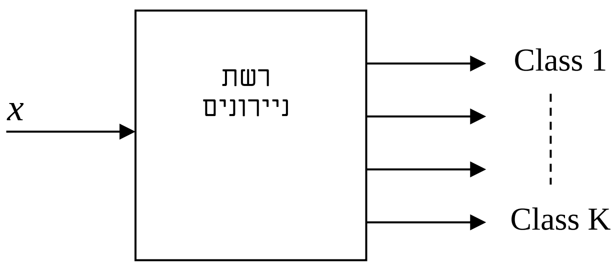
$$N = O\left(\frac{W}{\varepsilon}\right)^{\alpha}, \ \alpha \in (1,2)$$

כאשר W מספר פרמטרי הרשת. כלומר: n יחסי למספר הפרמטרים (גודל הרשת) וביחס הפוך לאחוז השגיאה הנדרש.

נציין לסכום כי רשתות עצביות רב-שכבתיות מהוות כלי שימושי ורב-תכליתי ללימוד מדוגמאות. עם זאת, תהליך הלימוד ואימון הרשת עשוי לדרוש זמן רב.

* 6.4 שימושים

(1) סיווג (Classification): יישום זה כבר נדון בהרחבה כמובן. כפי שצוין, בסיווג ל-K מחלקות מקובל הוא להקצות יציאה אחת לכל אחת מהמחלקות, כאשר היציאה הגבוהה ביותר "זוכה".



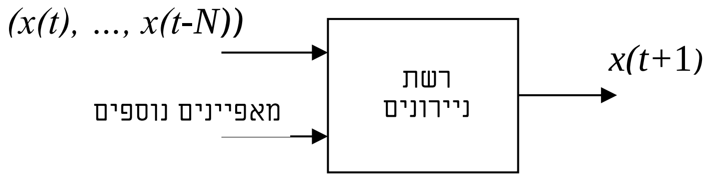
ציור 9 : סיווג

<u>(2) מיפוי מידע לפעולה:</u>

ציור 10 : בחירת פעולה

רשת הנוירונים מאפשרת כזכור מימוש של מיפוי לא-לינארי בין הכניסה ליציאה. הכניסה עשויה להיות מידע (גולמי או מעובד) מחיישן, למשל תמונת דרך, והיציאה פקודה כלשהי – למשל זווית ההגה במכונית. ביישום זה, הדוגמאות התקבלו על ידי הקלטת פקודות הנהיגה של נהג אנושי.
רשת הנוירונים מהווה פה תחליף (או משלים) לניתוח פרוצדורלי של המידע לצורך חישוב הפעולה הנדרשת.

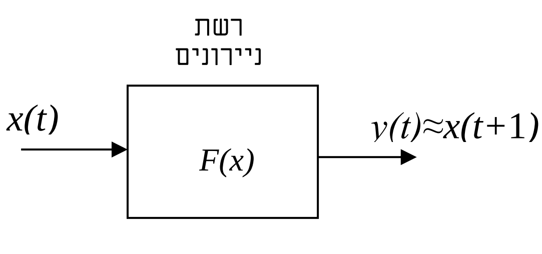
<u>(3) חיזוי סדרות עיתיות (זמניות):</u> המטרה פה לחזות ערכים עתידיים של סדרה זמנית על סמך ערכי העבר. שימוש אפייני הינו חיזוי שערי הבורסה. הדוגמאות מתקבלות מתוך ההיסטוריה של התהליך. באותה מידה ניתן כמובן לנסות ולאמוד גודל אחר, $y(t)$, מתוך הסדרה $\{x(t)\}$.



ציור 11 : חיזוי.

<u>(4) זיהוי מערכות דינמיות:</u>

מערכות דינמיות רבות מאופינות על ידי מודל לא-לינארי. מודל מקובל הינו מודל המצב:
$x_{t+1} = f(x_t), \quad x_t \in \Re^n$. זיהוי המערכת במקרה זה פרושו מציאת הפונקציה f .

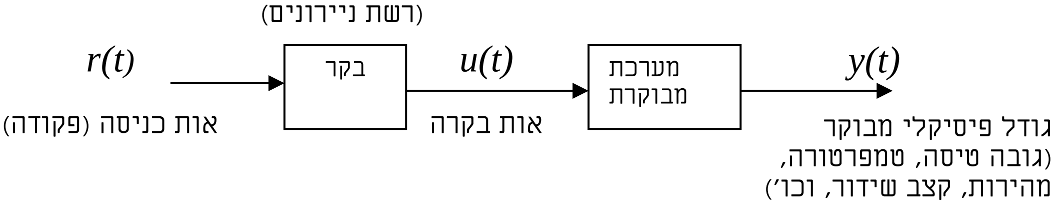


ציור 12 : זיהוי.

הלימוד מתבצע פה מתוך ההיסטוריה של התהליך, בעזרת "דוגמאות" מהצורה $\{x_t, d_t = x_{t+1}\}$.

(5) בקרה אדפטיבית (בחוג פתוח או סגור)

לרשתות עצביות שימוש כחלק מבקר לא לינארי של מערכות דינמיות. בעיית הבקרה בחוג פתוח הינה כבציור:

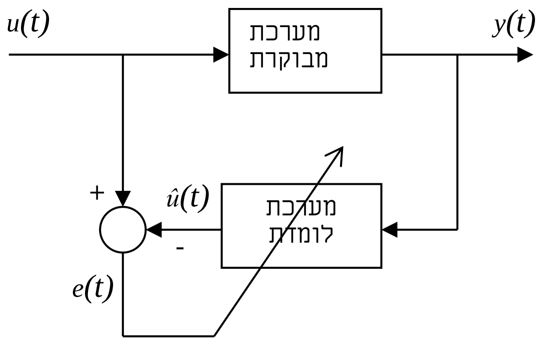


ציור 13 : בקרה בחוג פתוח

בבעיית עקיבה, המטרה הנדרשת הינה כי ערך הגודל הפיסיקלי המבוקר (יציאת המערכת) יהיה זהה לערך אות הפקודה. ניתן להכיל בבקר רשת עצבית, המממשת מיפוי לא-לינארי (סטטי או דינמי) בין הכניסה ליציאה.

קיימת בעיה מסוימת במציאת הדוגמאות לצורך כיוונון הבקר. לשם כך יש שתי גישות :

א. <u>לימוד מודל הפכי</u> : לצורך עקיבה מושלמת הבקר נדרש לבצע מעין "הפיכה" של המערכת הדינמית. ניתן לחבר את הבקר המיועד באופן הבא :

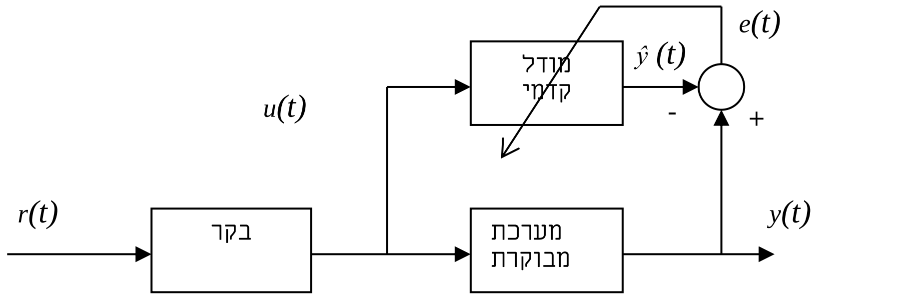


ציור 14 : לימוד מודל הפכי.

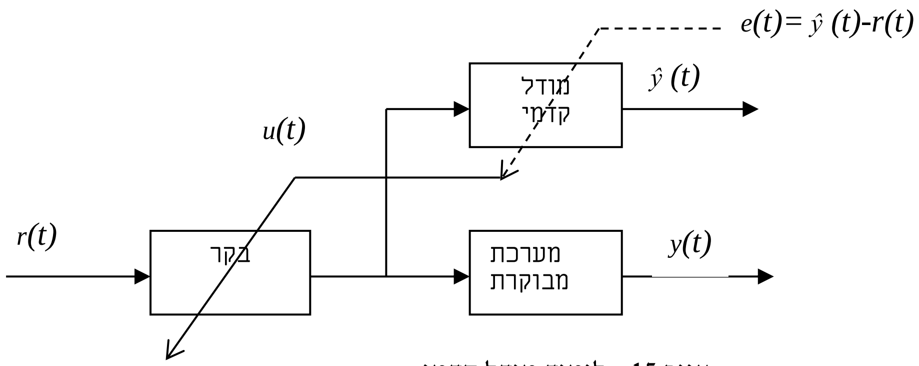
באופן זה הבקר ילמד מיפוי הפוך : $u(t) = f\left(y(t)\right)$ בגמר שלב הלימוד, הבקר יחובר באופן הרגיל. שיטה זו מותנית בקיומו של מיפוי הפכי (במובן מתאים למערכת דינמית).

ב. <u>לימוד מודל קדמי :</u> שיטה זו מבוצעת בשני שלבים. ראשית נלמד מודל קדמי של המערכת.

שלב א:



שלב ב:



ציור 15 : לימוד מודל קדמי.

בגמר הלימוד, או לאחר שהושג דיוק סביר במודל הקדמי, נוסף לימוד הבקר. לימוד זה מבוצע באמצעות Back-Propagation של שגיאת היציאה $e(t)=\hat{y}(t)-r(t)$ דרך שכבות המודל הקדמי, אל הנוירונים של הבקר.

ניתן להשתמש בשיטות דומות לצורך בעיית הבקרה עם משוב (בקרה בחוג סגור), שהיא הנפוצה והשימושית יותר.

ציור 16 : מערכת בקרה בחוג סגור