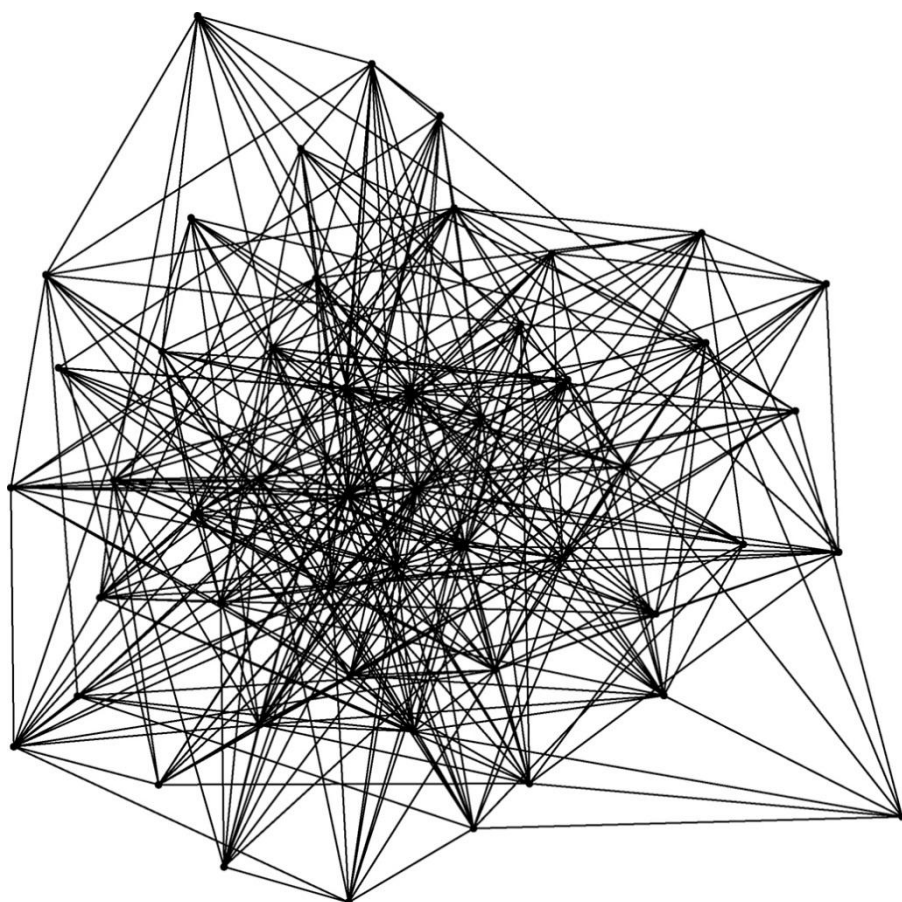


רשתות נוירונים ונהיגה אוטונומית

גיא בנויש

ת.ז 304870223

מרץ 2015



תוכן עניינים

3	מבוא
4	אינטליגנציה מלאכותית
4	רקע
5	למידה חישובית
6	רקע לרשתות נוירונים
6	הגדרת רשת נוירונים
7	יתרונות של רשתות נוירונים
9	המוח האנושי
14	רשתות נוירונים מלאכותיות
14	ייצוג
16	סוגים של פונקציות הפעלה
18	ארכיטקטורות רשת
20	ייצוג חלופי
22	בעיית XOR
25	סוגי בעיות
26	למידה
27	פונקציית העלות
27	אלגוריתם back-propagation
36	נהיגה אוטונומית
36	הקדמה
37	ארכיטקטורת הרשת
38	אימון הרשת
38	בעיות פוטנציאליות
38	טרנספורמציות תמונה
42	טרנספורמציות היגוי
43	פרטי האימון
45	ביבליוגרפיה

מבוא

רשתות נירונים, או רשתות נירונים מלאכותיות ליתר דיוק, מייצגות ענף טכנולוגי בעל שורשים הנוגעים במספר תחומים שונים כגון נירולוגיה, מתמטיקה, סטטיסטיקה, פיזיקה, מדעי המחשב והנדסה. השימוש ברשתות נירונים לפתרון בעיות הוא מגוון ורחב, והוא כולל בתוכו מידול, ניתוח זמנים, זיהוי תבניות, עיבוד אותות ואפילו נהיגה אוטונומית. רשתות נירונים מסתמכות על תכונה חשובה ועוצמתית אשר איננה קיימת במערכות אחרות: היכולת ללמוד מהקלט ועל ידי כך להסתגל לסביבה בה היא נמצאת.

אינטליגנציה מלאכותית הינה ענף במדע העוסק בחקר ישויות תבונתיות. אחת המוטיבציות לחקר בנושא היא שאנו נוכל ללמוד עוד על עצמנו, בהיותנו ישויות תבונתיות. בניגוד לפסיכולוגיה ולפילוסופיה, ענפים העוסקים בעיקר בהבנת התבונה, אינטליגנציה מלאכותית עוסקת גם בעניין חיקוי התבונה בצורה מלאכותית. סיבה נוספת למחקר הרב הנעשה בנושא היא שהישויות המלאכותיות הללו מעניינות ושימושיות בפני עצמן. ענף האינטליגנציה המלאכותית הפיק תוצאות משמעותיות ומרשימות אפילו שאיננו קיים הרבה מאוד זמן כמו ענפים אחרים במדע. אין לדעת מה בדיוק צופן העתיד, אך זה ברור שלמכונות בעלות רמת אינטליגנציה המשתווה (ואולי אפילו עולה על) לרמת האינטליגנציה של בני אדם תהיינה השפעה ענקית על חיי היום-יום ועל התקדמות הציביליזציה.

לאורך השנים הוצעו מספר הגדרות לאינטליגנציה מלאכותית ע"י אישים בתחומים שונים כגון מדעי המחשב, פילוסופיה ופסיכולוגיה. להלן 4 הגישות הפופולריות להגדרת אינטליגנציה מלאכותית [9]:

1. התנהגות אנושית: גישת מבחן טיורינג – מבחן טיורינג, כפי שהוצע ע"י אלן טיורינג ב-1950, עוצב כדי לספק הגדרה מספקת לאינטליגנציה. טיורינג הגדיר התנהגות אינטליגנטית בתור היכולת להשיג ביצועים כמו-אנושיים בכל המטלות הקוגניטיביות, ברמה מספיק גבוהה כדי להתל באדם החוקר את המכונה. המבחן שטיורינג הציע הוא שמכונה נחקרת ע"י אדם אשר איננו נמצא בקרבת המכונה. המכונה נחשבת כעוברת את המבחן אם החוקר אינו יכול לדעת האם יש אדם או מכונה בצד השני. כדי לעבור את המבחן, המכונה תצטרך לבצע את הדברים הבאים:
 - **עיבוד שפה טבעית** כדי לאפשר לה לתקשר עם בן אדם בשפה האנגלית או שפה כלשהי אחרת.
 - **ייצוג ידע** כדי לאחסן מידע אשר מגיע למכונה לפני או תוך כדי החקירה.
 - **היגיון** כדי להשתמש במידע שיש לה על מנת לענות לשאלות ולהסיק מסקנות חדשות.
 - **למידה חישובית** כדי להסתגל לנסיבות חדשות ולזהות ולנתח תבניות.
2. מחשבה אנושית: גישת המידול הקוגניטיבי – אם ברצוננו לקבוע האם מכונה חושבת כמו בן אדם, חייבת להיות בידינו דרך כלשהי כדי לקבוע כיצד בני אדם חושבים. צריך לחקור לעומק את תהליך החשיבה של המוח האנושי. ישנן שתי דרכים לעשות זאת: בחינה עצמית (introspection) – לנסות להבין את מחשבותינו בעוד הן מתרחשות – או באמצעות ניסויים פסיכולוגיים. רק ברגע שתהיה לנו תאוריה מספיק מדויקת על תהליך החשיבה, יהיה זה אפשרי לכתוב תוכנה אשר תממש תאוריה זו. השדה הבין-תחומי מדע קוגניטיבי משלב מודלים ממוחשבים מענף האינטליגנציה המלאכותית ביחד עם טכניקות ניסיוניות מתחום הפסיכולוגיה כדי לבנות תאוריות מדויקות וברות-בחינה על תהליך המחשבה האנושי.
3. מחשבה רציונלית: גישת חוקי המחשבה – הפילוסוף היווני אריסטו היה אחד האנשים הראשונים שניסו לאגד חוקים המגדירים "חשיבה נכונה" – תהליכי הצדקה שלא ניתן להפריכם. ניסיונות אלה הובילו להתפתחות ענף הלוגיקה במדע. התפתחות הלוגיקה הפורמלית בסוף המאה ה-19 ובתחילת המאה ה-20 סיפקו מינוח מדויק להצהרות על כל מיני דברים בעולם ומערכות היחסים ביניהם. עד 1965 כבר היו קיימות תוכנות אשר יכולות למצוא פתרון (בהינתן מספיק זיכרון וזמן) לפסוק לוגי במידה והוא ספיק. הגישה הלוגיסטית של אינטליגנציה מלאכותית מטרתה לבנות תוכנות יעילות כאלו כדי ליצור מערכות אינטליגנטיות.

4. התנהגות רציונלית: גישת הסוכן הרציונלי – התנהגות רציונלית משמעותה לפעול כדי להשיג מטרות מסוימות, תחת אמונות וחוקים מסוימים. "סוכן" הוא משהו שתופס את הסביבה ומבצע פעולות מסוימות. בגישה זו, אינטליגנציה מלאכותית היא המחקר שמטרתו בניית סוכנים רציונליים. בגישת "חוקי המחשבה" הדגש מונח על הסקה נכונה. הסקה נכונה היא רק חלק מתכונותיו של סוכן רציונלי, שכן לא ניתן להתנהג בצורה רציונלית ללא הסקת מסקנות רציונלית. מצד שני, הסקה נכונה של מסקנות איננה הרציונליות כולה, שכן לפעמים ישנן סיטואציות בהן אין משהו לעשות שניתן להוכיחו כנכון, אך בכל זאת משהו חייב להיעשות.

למידה חישובית

למידה חישובית הינה תחום במדע החוקר את הבנייה והמחקר על אלגוריתמים שיכולים ללמוד מתוך אוסף נתון של מידע. למידה חישובית היא כלי חשוב בשדה אינטליגנציה המלאכותית שכן ללא יכולות למידה, אף מכונה לא תעבור את מבחן טיורינג ולא תוכל להיות מוגדרת כמכונה אינטליגנטית. אלגוריתמים אלו בונים מודל מבוסס קלט אשר לפיו הם מבצעים תחזיות או החלטות. תכונה זו שונה מאלגוריתמים חסרי יכולת למידה אשר עובדים ע"י ביצוע רצף הוראות מפורשות ידועות מראש. בד"כ לאלגוריתם של למידה חישובית יש שני שלבים: שלב האימון, בו מזינים לאלגוריתם מידע מתוך סט אימון כלשהו, ושלב העבודה, בו האלגוריתם צריך להסיק מסקנות מתוך מידע בסט הבחינה על סמך הידע שצבר בשלב האימון.

ישנם שני סוגי למידה עיקריים:

- למידה מפוקחת (supervised learning) – תהליך למידה שמטרתו הסקת מסקנות ממידע בעל תוויות (לדוגמה, זוג בעל שני אובייקטים כאשר האובייקט הראשון הוא הקלט, בדרך כלל בצורת וקטור ערכים, והאובייקט השני הוא הפלט המבוקש ביחס לווקטור הקלט). לדוגמה ניקח את מטלת חיזוי מחירי בתים: סט האימון יורכב מזוגות אובייקטים כאשר הראשון הוא וקטור של מאפייני בית (שטח, כמות חדרים, האם יש חצר וכו') ואובייקט השני יהיה המחיר בו נמכר הבית. בשלב העבודה נזין לאלגוריתם מאפייני בית שעומד למכירה, והאלגוריתם יחזה את המחיר בו הוא יימכר על סמך ניסיון העבר עם סט האימון.
- למידה לא מפוקחת (unsupervised learning) – תהליך למידה שמטרתו למצוא מבניות כלשהו בסט של מידע כאשר למידע אין תוויות. בניגוד ללמידה מפוקחת, קשה להעריך את טיבו של אלגוריתם הלומד בצורה לא מפוקחת: כיוון שאין פלט מבוקש (תוויות) אז אין דרך טובה להעריך את הפתרון הפוטנציאלי.

רקע לרשתות נירונים

הגדרת רשת נירונים

מחקר תיאורטי ומעשי על רשתות נירונים מלאכותיות נבע ממש מראשיתו מההכרה שהמוח האנושי מבצע חישובים בצורה השונה ביסודה מהדרך בה מתבצעים חישובים ע"י מחשב דיגיטלי קונבנציונלי. המוח הוא יחידת חישוב מורכבת, אי-ליניארית ומקבילית. יש למוח את היכולת לארגן את המבנים-המעבדים שלו, הנירונים, בצורה ייחודית המאפשרת לאותו מקבץ נירונים לבצע פעולות מסוימות (זיהוי תבניות, שליפה מזיכרון, יכולות מוטוריות וכד') במהירות גדולה בכמה סדרי גדול מהמחשב המהיר ביותר כיום.

אחת המערכות הנורולוגיות החשובות ביותר בחיי היום-יום, הראייה, הינה מערכת עיבוד מידע. תפקידה של מערכת הראייה הוא לספק ייצוג של הסביבה המקיפה אותנו, וחשוב מזאת, את המידע הדרוש לנו כדי לתקשר עם הסביבה. המוח מבצע מטלות של זיהוי תפיסתי של העולם (כגון זיהוי אובייקטים מוכרים) בצורה תדירה ובמהירות גבוהה מאוד, בעוד שאפילו המחשב החזק בעולם יתקשה בהבחנה חזותית בין שני אובייקטים דומים.

כיצד המוח עושה זאת? בזמן הלידה המוח מכיל מעט מאוד מידע לגבי הסביבה (בעיקר אינסטינקטים מולדים) אך יש לו את היכולות לבנות לעצמו כללים חדשים באמצעות ניסיון וחוויות. כידוע, ניסיון וחוויות הם דברים אשר צוברים במהלך החיים, אך ההתפתחות הדרמטית ביותר – היצירה של החיווט הפיזי במוח – מתבצעת בשנתיים הראשונות לחיים [7]. התפתחויות נוספות של המוח, כגון שינוי הקשרים הפיזיים, קורות לאורך כל החיים.

למוח יש את תכונת הגמישות: היכולת לפתח את מערכת עיבוד המידע בעקבות שינוי בסביבה ותוך כדי התאמה אליה. המוח מסתגל, ותכונה זו חיונית להתפתחות ולייעול אופן החשיבה. כשם שתכונה זו חיונית למוח הביולוגי, כך היא חיונית גם לרשתות הנירונים המלאכותיות. בצורתה הכללית ביותר, רשת נירונים מלאכותית היא מכונה אשר מעוצבת כדי לשמש מודל לדרך בה המוח מבצע חישוב מסוים. רשת כזו בד"כ ממושמת על גבי רכיבים אלקטרוניים באמצעות סימולציה תוכניתית על מחשב דיגיטלי.

רשתות נירונים מלאכותיות בנויות כך שהן יכולות "ללמוד" מניסיון העבר: בכל רשת יש הרבה מאוד קשרים בין נירונים מלאכותיים (הנקראים גם יחידות עיבוד) אשר יכולים להשתנות לאורך חיי הרשת כדי להפיק ביצועים טובים וחישוב אופטימלי.

כעת אנו יכולים להציע את ההגדרה הבאה לרשת נירונים מלאכותית: רשת נירונים הינה יחידת עיבוד מקבילית ומבוזרת אשר מורכבת מכמה תת-יחידות עיבוד פשוטות אשר להן היכולת לאחסן מידע ולהעביר מידע לתת-יחידות עיבוד נוספות.

רשת נירונים מלאכותית דומה למוח בשני אופנים עיקריים: רכישת ידע באמצעות תהליך למידה ואחסון מידע באמצעות חיבורים בין נירונים, הידועים גם בתור משקלים סינפטיים.

התהליך באמצעותו תהליך הלמידה מתבצע נקרא אלגוריתם למידה. באמצעות אלגוריתם זה משתנים המשקלים הסינפטיים של הרשת בצורה מוסדרת והיררכית כדי להשיג את מטרת העיצוב הנדרש מהרשת.

שינוי המשקלים הסינפטיים הוא השיטה המקובלת כדי להשיג את הפונקציונליות הדרושה. הקשרים עצמם קבועים אך המשקל של כל קשר, או החוזק שלו, משתנה (תהליך הדומה למוח של בוגר). ישנן גם רשתות אשר יש להן את היכולות לשנות את הטופולוגיה של עצמן ע"י יצירת קשרים חדשים (תהליך הדומה למוח מתפתח).

יתרונות של רשתות נוירונים

כוח החישוב של רשתות נוירונים נובע, בראש ובראשונה, מהמבנה המבוזר והמקבילי שלהן. מעבר לכך, לרשתות נוירונים יש את היכולות ללמוד ולכן, להכליל. הכללה ברשתות נוירונים פירושה הפקת פלט סביר עבור קלט שלא הופיע בשלב האימון (למידה). שתי יכולות עיבוד המידע שהוזכרו מאפשרות לרשתות נוירונים לפתור בעיות מורכבות וקשות לפתירה. בפועל, רשתות נוירונים אינן יכולות לספק פתרון כאשר המערכת כולה מורכבת רק מהרשת עצמה. גישת עיצוב המערכת צריכה לכלול רשתות נוירונים כחלק אינטגרלי עוד בשלב בעיצוב. בהינתן בעיה מורכבת, יש לפרקה לבעיות יותר קטנות ופשוטות ולהקצות רשת נוירונים לכל אחת ואחת מהן, כאשר כל רשת נוירונים בנויה לטפל בבעיה הספציפית שהוקצתה לה. למרות הדמיון לדרך הפעולה של המוח עצמו, יש לומר שישנה עוד דרך ארוכה עד אשר, ואם בכלל, נצליח לבנות מכונה אשר מחקה את המוח האנושי.

להלן רשימת יתרונות ותכונות רצויות של רשתות נוירונים [7]:

- אי-ליניאריות – רשת נוירונים יכולה להיות ליניארית או אי-ליניארית. רשת נוירונים המורכבת מקשרים בין נוירונים אי-ליניאריים (הכוונה היא לפונקציית העיבוד שמחשב כל נוירון מלאכותי) היא בעצמה אי-ליניארית. אי-הליניאריות הפוטנציאלית של רשת נוירונים מיוחד במובן שהוא מבוזר לאורך כל הנוירונים המלאכותיים. תכונה זו חשובה מאוד שכן יכול להיות שאות הקלט הוא אי-ליניארי באופן אינהרנטי (קול, לדוגמה).
- מיפוי קלט-פלט – הפרדיגמה הפופולרית של למידה מפוקחת מסתמכת על כך שהמשקלים הסינפטיים של הרשת משתנים בעקבות תהליך האימון בעזרת סט האימון. כל דוגמה בסט האימון מכילה אות קלט ואת התגובה הרצויה. מגישים לרשת הנוירונים דוגמה שנבחרה באקראי מסט האימון ומשנים את המשקלים הסינפטיים כדי לצמצם את הפער בין התגובה האמתית של הרשת לתגובה הרצויה. תהליך זה חוזר על עצמו הרבה פעמים (ככל שיש יותר דוגמאות בסט האימון כך הרשת תהיה מנוסה יותר ובעלת שיעור שגיאה קטן יותר) עד אשר כבר אין שינוי משמעותי במשקלים הסינפטיים והרשת הגיעה למצב יציב. ניתן גם להשתמש באותן דוגמאות מסט האימון רק בסדר שונה כדי שתהליך האימון יהיה אפקטיבי. רשת הנוירונים לומדת מהדוגמאות ע"י בניית מיפוי קלט-פלט לבעיה הנתונה. גישה זו מזכירה הסקה סטטיסטית א-פרמטרית: ענף בסטטיסטיקה אשר עוסק בהערכה חסרת מודל קובע מראש. א-פרמטרי פירושו שאין הנחות מוקדמות על המודל הסטטיסטי של אות הקלט. זאת בניגוד לסטטיסטיקה פרמטרית אשר מסתמכת על כך שהמודל הסטטיסטי (התפלגות, לדוגמה) ידוע מראש. ניקח לדוגמה מטלה של סיווג תבניות: הדרישה היא לבצע התאמה בין אות קלט המייצג אובייקט לאחת מבין כמה קטגוריות ידועות מראש. גישה א-פרמטרית לבעיה זו תהיה הערכה של גבולות החלטה שרירותיים במרחב אותות הקלט לבעיה באמצעות שימוש בסט של דוגמאות, ולעשות זאת ללא שימוש במודל התפלגות הסתברותי כלשהו. תכונה זו מצביעה על אנלוגיה חזקה בין מיפוי קלט-פלט לבין הסקה סטטיסטית א-פרמטרית.

- הסתגלות – לרשתות נוירונים יש את היכולת להתאים את המשקלים סינפטיים שלהם בהתאם לשינויים בסביבה. כמו כן, ניתן לאמן מחדש בקלות רשת נוירונים, אשר כבר אומנה בעבר למטרה ספציפית, כדי שתהיה יותר אפקטיבית בהינתן שסביבת העבודה השתנתה מעט. מעבר לכך, כאשר סביבה העבודה אינה קבועה (סביבה בה הסטטיסטיקות משתנות ללא הרף), רשת נוירונים יכולה להיות מעוצבת כך שהמשקלים הסינפטיים משתנים תוך כדי עבודה (בד"כ המשקלים הסינפטיים משתנים בזמן האימון אך נשארים קבועים בזמן העבודה). באופן כללי ניתן לומר שככל שמערכת יותר סתגלנית, תוך כדי שמירה על מערכת יציבה בזמן עבודה, כך ביצועיה יותר חסינים (robust) כאשר הסביבה אינה קבועה. אמירה זו מלווה בהסתייגות קטנה: סתגלניות לא תמיד מובילה לחסינות. לפעמים אפילו ההפך הוא הנכון. לדוגמה, אם מערכת מסתגלת בקבועי זמן קטנים מדי היא תשתנה מהר מדי ועלולה לנטות להגיב להפרעות רנדומליות בקלט כך שביצועי המערכת יהיו גרועים. כדי להשתמש בצורה נכונה בתכונת הסתגלניות יש לבחור קבועי זמן לא קצרים מדי – כדי למנוע פלט לא מדויק עקב הפרעות בקלט – ולא ארוכים מדי – כדי לאפשר הגבה לשינויים משמעותיים בסביבה.
- תגובה מבוססת ראיות – בהקשר של זיהוי תבניות, רשת נוירונים יכולה להיות מעוצבת כך שהפלט איננו מורכב אך ורק מהקטגוריה אליו שייך הפלט, אלא גם מרמת הביטחון של הרשת לגבי החלטתה. מידע זה יכול להיות שימושי כדי לפסול החלטות רב-משמעותיות וע"י כך לשפר את ביצועי הרשת.
- עמידות בכשלים – לרשת נוירונים הממומשת בחומרה או בקושחה יש הפוטנציאל להיות בעלת עמידות גבוהה בכשלים במובן שביצועיה ידרדרו במעט בלבד בתנאי פעולה לא אופטימליים. לדוגמה, אם נוירון מלאכותי אחד או קשריו ניזוק, הזיכרון של תבנית שאוחסנה בעבר גם הוא נפגע. למרות זאת, עקב האופי המבוזר של רשת נוירונים, הנזק לנוירונים המלאכותיים או לקשרים צריך להיות מקיף על מנת שהביצועים ידרדרו בצורה רצינית. לכן, תחת תנאים פיזיים לא אופטימליים, רשת נוירונים תחווה רק ירידה קטנה בביצועים ולא כשל קטסטרופלי.
- מימוש על גבי מעגלים משולבים – המקביליות המסיבית של רשתות נוירונים מאפשרת לבצע חישובים מסוימים בצורה מהירה מאוד. בשל מאפיין זה, רשתות נוירונים הינן מעומדות טובות מאוד למימוש על גבי מעגלים משולבים בקנה מדיה גדול (VLSI). מאפיין חשוב של VLSI שהוא מאפשר לממש התנהגות מורכבת מאוד בצורה היררכית.
- אחידות אנליטית ועיצובית – רשתות נוירונים הינן מעבדי מידע אוניברסליים במובן שמשתמשים באותם הסימונים והמאפיינים בכל התחומים בהן מעורבות רשתות נוירונים: נוירונים מלאכותיים, בצורה כזו או אחרת, הם מרכיב משותף לכל רשתות הנוירונים. שיתופיות זו מאפשרת לחלוק תאוריות ואלגוריתמי למידה בין רשתות שונות. ניתן לבנות רשתות מודולריות באמצעות אינטגרציה של רשתות שונות כמעט ללא מאמץ.

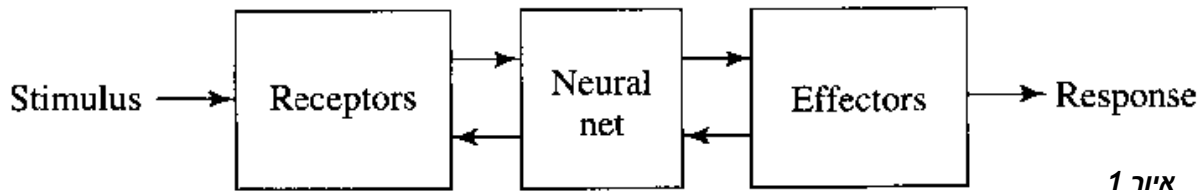
בשל ההשראה שנתן המוח לרעיון רשתות הנוירונים המלאכותיות, יהיה זה ראוי להציג רקע ביולוגי על מבנה המוח, ולזה מוקדש הפרק הבא.

המוח האנושי

בצורה פשטנית, ניתן להסתכל על מערכת העצבים האנושית כעל מערכת בעלת 3 שלבים (איור 1). מרכז המערכת הוא המוח, מיוצג על ידי רשת של תאי עצב, אשר עוסק באופן שוטף בקבלת מידע, תפיסת המהות שלו, וקבלת החלטות מתאימות. ישנם שני סטים נפרדים של חצים באיור. החצים שכיוונם משמאל לימין מייצגים העברת אות נושא "קדימה" במערכת. החצים שכיוונם מימין לשמאל מייצגים נוכח "משוב" (feedback). הקולטנים הופכים גירויים חיצוניים (צליל לדוגמה) או פנימיים (כאב לדוגמה) לאותות חשמליים אשר מועברים למוח. האפקטורים הופכים אותות חשמליים שמקורם במוח לתגובה הרצויה (בהתאם למידע הנישא על גבי האותות החשמליים) בתור פלט המערכת [7].

הנוירונים (תאי עצב), הינם יחידות העיבוד הבסיסיות אשר מרכיבות את המוח. נוירונים איטיים יותר בסדר גודל של בין 5 ל-6 סדרי גודל מאשר שערים לוגיים על שבב סיליקון. עיבוד אות בשער לוגי קורה בסדר גודל של ננו-שנייה ואילו עיבוד אות בנוירון קורה בסדר גודל של מילי-שנייה. למרות זאת, המוח מפצה על האיטיות היחסית של יחידות העיבוד שלו: במוח יש מספר עצום של נוירונים ומספר עוד יותר עצום של קשרים ביניהם. מוערך כי יש במוח אנושי טיפוס 10 ביליון נוירונים ו-60 טריליון קשרים בין נוירונים [11].

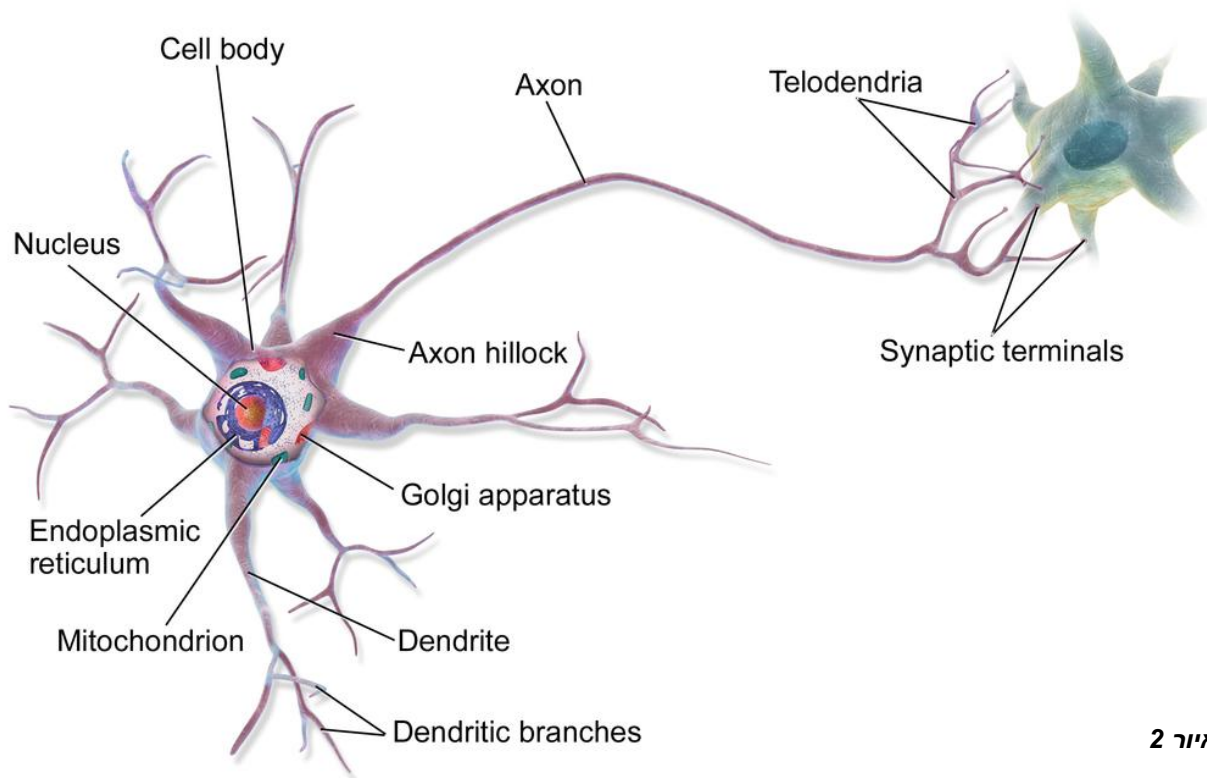
התוצאה היא שהמוח הוא מבנה יעיל מאוד מבחינת כוח עיבוד וגם מבחינת צריכת האנרגיה שלו: מוערך כי העלות האנרגטית של חישוב לשנייה במוח היא 10^{-16} ג'אול ואילו הערך המקביל במחשבים כיום הוא בערך 10^{-6} ג'אול.



איור 1

נוירון טיפוס מורכב ממספר חלקים (איור 2):

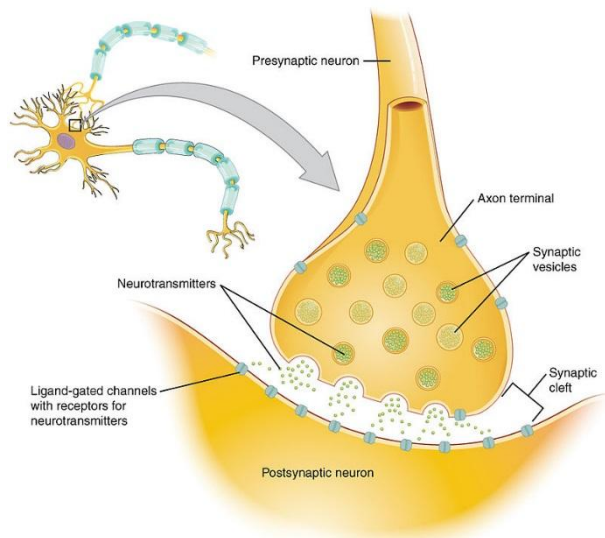
- גוף התא (soma - סומה) – מכיל את גרעין התא. באזור זה בתא מתרחשת רוב פעילות התא כגון נשימה תאית וסינתוז חלבונים.
- כניסון (דנדריט - dendrite) – שלוחה קצרה ומסועפת (dendros פירושה "עץ" ביוונית) אשר מהווה מקור הקלט העיקרי של הנוירון. ענפי הדנדריט מחוברים לקצה של אקסון של תא עצב אחר על מנת לקבל מידע בצורת אותות כימיים והעברתם לגוף התא בצורת אותות חשמליים.
- יציאון (אקסון - axon) – שלוחה ארוכה בצורת סיב אשר על גביה עוברים אותו חשמליים מגוף התא ומתורגמים לאותות כימיים בקצה האקסון – הטרמינלים הסינפטיים. האותות הכימיים הללו מהווים את מקור הפלט העיקרי של הנוירון ויכולים להגיע לנוירון אחר או לכל תא בעל סינפסות.



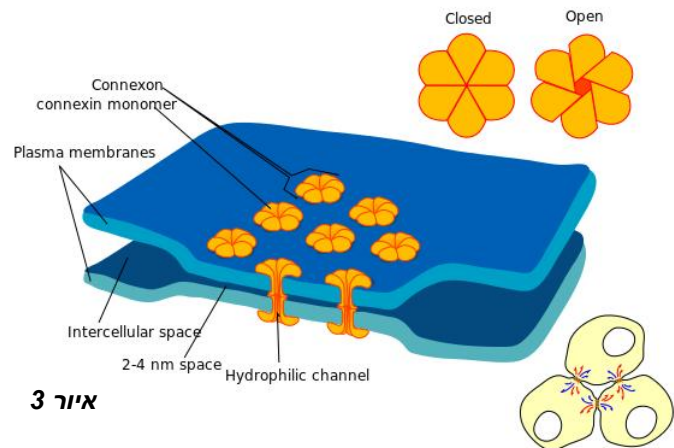
איור 2

הסינפסות (synapses) הם המבנים האלמנטריים אשר מאפשרים חיבוריות בין נוירונים. ניתן לחלק את הסינפסות לשני סוגים עיקריים:

- סינפסות חשמליות – קרום התא אשר מעביר מידע וקרום התא אשר מקבל מידע מצוידים ב gap junctions: ערוץ מיוחד אשר מסוגל להעביר זרם חשמלי מתא אחד לאחר (איור 3). יתרון משמעותי לסוג הסינפסות הזה הוא תקשורת מהירה בין תאים.
- סינפסות כימיות – זהו הסוג הנפוץ ביותר של סינפסות בגוף האנושי. תהליך חשמלי פרה-סינפטי (presynaptic) משחרר חומר נושא מידע (מוליך עצבי - neurotransmitter) אשר עובר בדיפוזיה לקולטנים של נוירון אחר (תהליך פוסט-סינפטי). בצורה זו סינפסה הופכת אות חשמלי פרה-סינפטי לאות כימי אשר הופך בחזרה לאות חשמלי פוסט-סינפטי בנוירון היעד. תרגום האותות החשמליים נעשה ע"י מנגנון מיוחד בקצה האקסון הנקרא תעלת יונים ממותגת מתח. זוהי תעלה אשר מאפשר מעבר של יונים (אלו הם המוליכים העצביים) באמצעות הפעלת מתח (מתח זה הינו האות החשמלי העובר דרך האקסון) (איור 4)



איור 4



איור 3

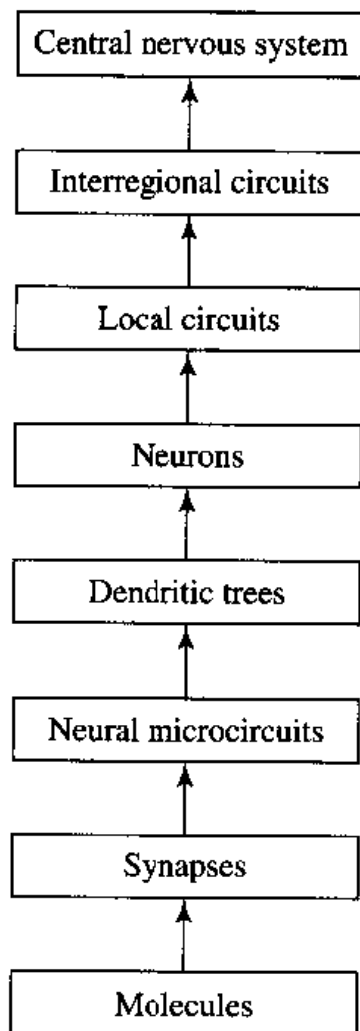
קיימים מאות מוליכים עצביים. להלן רשימה של מוליכים עצביים עיקריים:

1. אצטילכולין - מוליך הקשור לזיכרון ולמידה. על האזורים שבהם עובר האצטילכולין נמנה ההיפוקמפוס, מעבד הזיכרונות, וכנראה בשל כך אין חולה במחלת אלצהיימר, שההיפוקמפוס שלו נפגע, מסוגל לאחסן זיכרונות לטווח קצר.
2. קטקולאמיניים – בקבוצה זו נכללים הנוראדרנלין (מקור האדרנלין בגוף) והדופמין. הדופמין קשור לשני מסלולים מרכזיים במוח. הראשון הוא מסלול מוטורי, ונראה שמיעוט של מולקולות דופמין במוח קשור למחלת פרקינסון ולשיתוק. השני הוא מסלול הקשור למערכת הלימבית ולרגשות, שם עודף בדופמין מוביל להזיות ולסכיזופרניה. היעדר נוראדרנלין מקושר עם דיכאון. הדור החדש של התרופות האנטי דיכאוניות (SNRI) פוגע בתהליך הטבעי שבו תא המוצא שואב לתוכו שוב את רוב הנוראדרנלין שהוציא. תרופות אלו מאפשרות על ידי כך לכמות גדולה יותר של נוראדרנלין לצאת ולעבור בתאים.
3. אינדולמיניים – כאן נמצאים המלטונין והסרטונין. המלטונין אחראי על וויסות השעון הביולוגי ומקושר עם תהליך השינה. בדומה לנוראדרנלין, גם הסרטונין חשוב בטפול בדיכאון, והתרופות הפסיכיאטריות החדשות (SSRI ו-SNRI) מנסות לעכב את תהליך הספיגה החוזרת שלו חזרה לתא המוצא.
4. חומצות אמינו - מוליכים עצביים רבים מבוססים על אבני הבניין של חלבונים, החומצות האמיניות: GABA, גלוטמט, גליצין, היסטידין. היסטידין הופך בגוף להיסטמין. ההיסטמין קשור לוויסות מצבי עוררות ושינה. ככל שהוא מתמעט הגוף נע למצב של הרדמות. כמות מוגברת שלו גורמת לנדודי שינה, ואילו לקיחת תרופות אנטי-היסטמיניות גורמת לתופעה של נמנמת והרדמות.
5. נויורופפטידים - הנוירופפטידים הם מולקולות הקצרות מכדי להיקרא חלבון. עליהם נמנה חומר P, הממלא תפקיד חשוב בהעברת הכאב בעמוד השדרה, וכן האנדורפינים, השייכים לקבוצות החומרים שמפיק הגוף ודומים לאופיום. בשל כך, יש להם השפעה מרסנת על הכאב.

כפי שנאמר, למוח יש את תכונת הגמישות: הוא יכול לשנות את מבנהו הפיזי בעקבות גירויים חיצוניים או פנימיים. תכונה זו חיונית בהתפתחות המוח שכן בשלביו הראשונים בעולם פועל המוח ללא הרף כדי ליצור נוירונים חדשים וקשרים ביניהם. תכונת הגמישות במבוגרים באה לידי ביטוי בשני מגננים: יצירת קשרים סינפטיים בין נוירונים, ושינוי של קשרים סינפטיים קיימים.

אקסונים, קווי התמסורת בנוירון, ודנדריטים, התקני הקלט של הנוירון, הינם של מבנים דמויי חוט אשר שונים זה מזה ברמה המורפולוגית: לאקסון יש פני שטח חלקים יותר, מעט ענפים והוא ארוך יותר מדנדריט. לדנדריט יש פני שטח מחוספסים ורמת הסתעפות גבוהה.

במוח ישנם ארגונים אנטומיים בקנה מידה גדול ובקנה מידה קטן, ופונקציות שונות מתחרשות ברמות שונות של הארגון. באיור 5 ניתן לראות היררכיה של רמות הארגון במוח, אשר נובעת ממחקרים נרחבים על אזורים שונים במוח [11]. הסינפסות נמצאות ברמה הכי נמוכה, שכן פעולותיהן תלויות במולקולות וביונים. ברמות הבאות ישנם מיקרו-מעגלים נוירוניים (neural microcircuits), עצים דנדריטים (dendrite trees) ולבסוף נוירונים, בהתאמה. מיקרו-מעגל נוירוני הינו אוסף של סינפסות המאורגנות בדפוס מסוים של חיבוריות שמטרתה ביצוע פונקציה מסוימת על הקלט. ניתן להשוות מיקרו-מעגל נוירוני לשבב סיליקון הבנוי מאוסף של טרנזיסטורים. גודלו המינימלי של מיקרו-מעגל כזה הוא נמדד במיקרומטרים וזמן הביצוע המהיר ביותר של החישוב נמדד במילישניות. קבוצה של מספר מיקרו-מעגלים נוירונים נקראת תת-יחידה דנדריטית (dendrite subunit) אשר מהן מורכבים העצים הדנדריטים. הנוירון כולו, אשר גודלו 100 מיקרומטר בערך, מכיל מספר תת-יחידות דנדריטיות. ברמה הבאה של הארגון נמצאים המעגלים המקומיים (local circuits) אשר גודל אחד מהם בערך 1 מילימטר. המעגלים המקומיים מורכבים מנוירונים בעלי תכונות דומות או שונות. ההרכבים הנוירוניים הללו מבצעים חישובים בעלי בהתאם לאופיו של האזור הלוקלי במוח בו הם נמצאים. אחר כך מופיעים המעגלים הבין-אזוריים (interregional circuits) אשר מורכבים ממסלולים וממפות טופוגרפיות. המעגלים הבין-אזוריים מקשרים בין מספר אזורים במוח אשר אחראים על פעולות שונות. מפות טופוגרפיות מטרתן הגבה למידע אשר מגיע מהחושים. הן בד"כ בנויות בצורה של יריעות אחת על גבי השנייה כך שכל מפה מגיבה לגירויים המגיעים מחושים שונים. ברמה האחרונה נמצאת מערכת העצבים המרכזית (central nervous system), המוח, הלכה למעשה. המוח מורכב ממפות טופוגרפיות ומעגלים בין-אזוריים. בצורה זו בנוי המוח בצורה מאורגנת מאוד, כאשר לכל אזור פיזי יש את התפקיד שלו.



איור 6

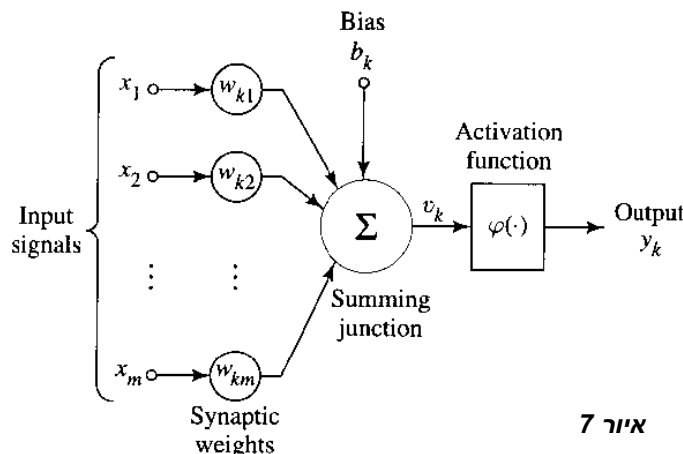
חשוב להכיר בעובדה שהמבנים הארגונים (מפות טופוגרפיות ומעגלים בין-אזוריים) המתוארים לעיל הינן מאפיין ייחודי של המוח. לא ניתן למצוא אותך במחשבים דיגיטליים ועדיין אין ביכולתנו לדמות אותן בצורה מושלמת באמצעות רשתות נוירונים מלאכותיות. למרות זאת, אנחנו מתקדמים בצעדי ענק לעבר רמות חישוביות הדומות להיררכיה המתוארת באיור 6. הנוירונים המלאכותיים בהם אנו משתמשים לבניית רשתות נוירונים מלאכותיות הם אכן מאוד פרימיטיביים בהשוואה לנוירונים ביולוגיים הנמצאים במוח. פרימיטיביים באותה מידה הן הרשתות המלאכותיות בהשוואה למעגלים המקומיים והבין-אזוריים במוח. למרות זאת, עבודה מספקת היא ההתקדמות הרבה שנעשתה בנושא בעשורים האחרונים. עם אנלוגיה נוירו-ביולוגית כמקור השראה ועושר טכנולוגי ותיאורטי רב אשר פועלים יד ביד, זה כמעט ודאי שבעשורים הקרובים ההבנה שלנו לגבי רשתות נוירונים מלאכותיות תהיה רבה ומתחכמת יותר מאשר היא כיום.

רשתות נוירונים מלאכותיות

ייצוג

הנוירון הוא יחידת עיבוד המידע הבסיסית הנחוצה לתפקוד רשת הנוירונים. באיור 7 ניתן לראות מודל של נוירון, היוצר את הבסיס לעיצוב רשת נוירונים מלאכותית. ניתן לזהות 3 אלמנטים בסיסיים במודל המוצג:

1. סט של סינפסות (או קישורים), אשר כל אחת מהן מאופיינת במשקל (או חוזק) משלה. הלכה למעשה, אות כלשהו x_j , המגיע בתור קלט לסינפסה j המחוברת לנוירון k , מוכפל במשקל הסינפטי w_{kj} .
2. מחבר (adder) של אותות הקלט, לאחר הפעלת הפרמוטציה באמצעות המשקלים הסינפטיים. עד כה (הכפלת האותות בקבועים וסכימתם) יצרנו משלב ליניארי (linear combiner).
3. פונקציית הפעלה (activation function) להגבלת האמפליטודה של אות הפלט. כפי שנראה בהמשך, פונקציית הפעלה היא הגורם לכך שפונקציית הפלט של הנוירון יכולה להיות אי-ליניארית.



איור 7

בדרך כלל, הפלט בצורתו המנורמלת (קרי, לאחר הפעלת פונקציית הפעלה) שוכן בטווח $[0,1]$ אך לעיתים גם בטווח $[-1,1]$.

כפי שניתן לראות באיור, ישנו עוד אות אשר מעובד ע"י הנוירון והוא איננו אות קלט. אות זה נקרא **אות הטיה** (bias) והוא מסומן בסימון b_k . מטרת אות זה היא הגברה או הנמכה של האות המסומן טרם כניסתו לפונקציית הפעלה.

במינוח מתמטי ניתן לתאר נוירון k ע"י זוג המשוואות הבאות:

$$(1.1) u_k = \sum_{j=1}^m w_{kj} x_j$$

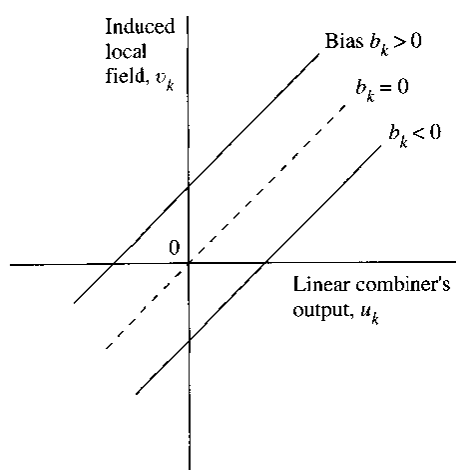
וגם:

$$(1.2) y_k = \varphi(u_k + b_k)$$

כאשר x_1, x_2, \dots, x_m הינם אותות הקלט הנקראים גם מאפיינים (features), $w_{k1}, w_{k2}, \dots, w_{km}$ הינם המשקלים הסינפטיים של הנוירון, u_k הוא הפלט של המשלב הליניארי על אותו הקלט, b_k הוא אות ההטיה, $\varphi(\cdot)$ היא פונקציית ההפעלה ו- y_k הוא הפלט של הנוירון. השימוש באות ההטיה נותן אפקט של טרנספורמציה אפינית (affine transformation) על הפלט של המשלב הליניארי כמתואר בנוסחה:

$$(1.3) v_k = u_k + b_k$$

הקשר בין הפלט של המשלב הליניארי u_k לבין פוטנציאל ההפעלה (activation potential or induced local field) מושפע מהחיוביות או השליליות של אות ההטיה כמתואר באיור 8.



איור 8

כיוון שאות ההטיה הוא אות קלט (אמנם חיצוני אך עדיין אות קלט) ניתן לשנות את המשוואות (1.2) ו(1.3) לצורה:

$$(1.4) v_k = \sum_{j=0}^m w_{kj} x_j$$

וגם:

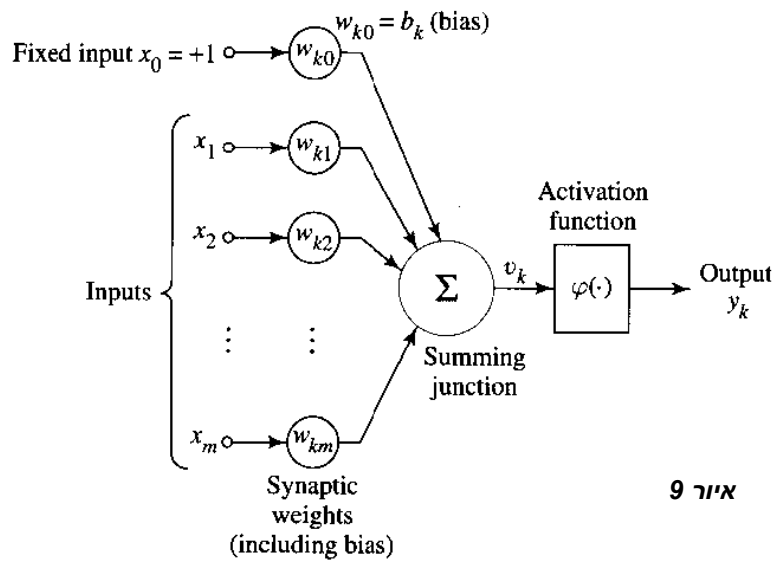
$$(1.5) y_k = \varphi(v_k)$$

כך שהוספנו אות קלט חדש (הוא אות ההטיה) וגם סינפסה חדשה אשר דרכה עובר האות. ניתן למצוא תיאור סכמתי של המתואר באיור 9. ערך האות החדש הוא:

$$(1.6) x_0 = +1$$

והמשקל הסינפטי הוא:

$$(1.7) w_{k0} = b_k$$



סוגים של פונקציות הפעלה

פונקציית ההפעלה מגדירה את הפלט של הנירון כתלות בפוטנציאל ההפעלה שלו v . להלן 3 סוגים בסיסיים של פונקציות הפעלה:

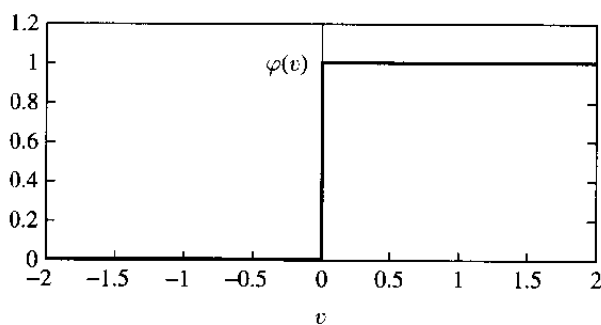
1. פונקציית סף (threshold function) – פונקציה זו מוגדרת ע"י:

$$(1.8) \varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

כתוצאה מכך מתקיים:

$$(1.9) y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases}$$

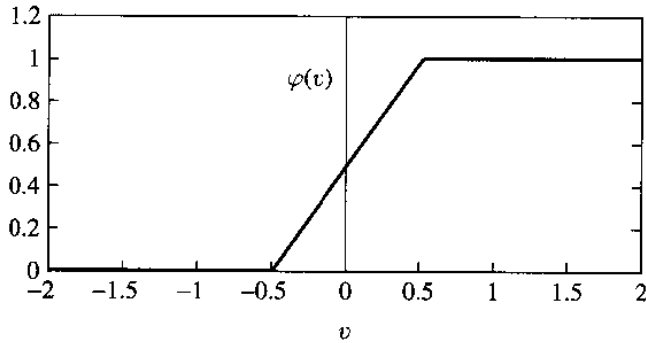
תיאור סכמתי:



2. פונקציית מקטעים ליניארית (piecewise linear function) – פונקציה זו מוגדרת ע"י:

$$(1.10) \varphi(v) = \begin{cases} 1 & \text{if } v \geq \frac{1}{2} \\ v & \text{if } \frac{1}{2} > v > -\frac{1}{2} \\ 0 & \text{if } v \leq -\frac{1}{2} \end{cases}$$

תיאור סכמתי:



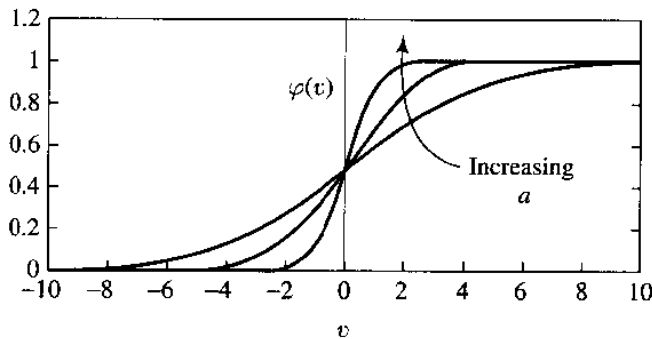
איור 11

3. פונקציית סיגמויד (sigmoid function) – פונקציה זו, אשר הגרף שלה דומה לאות האנגלית S, הינה אחת הפונקציות הנפוצות ביותר בבניית רשתות נוירונים מלאכותיות. היא מוגדרת כפונקציה עולה-ממש ובעלת איזון חינוני בין ליניאריות לאי-ליניאריות. פונקציית סיגמויד נפוצה הינה הפונקציה הלוגיסטית (מסומנת לעיתים באות g) אשר מוגדרת ע"י:

$$(1.11) \varphi(v) = g(v) = \frac{1}{1 + \exp(-av)}$$

כאשר a הוא פרמטר השיפוע של פונקציית סיגמויד זו. ע"י שינוי הערך של a ניתן לשלוט ברמת ה"התלילות" של הפונקציה. ניתן לראות באיור 12 גרף של הפונקציה הלוגיסטית עם ערכים שונים של הפרמטר a. כאשר a שואף לאינסוף, פונקציית סיגמויד הופכת להיות פונקציית סף פשוטה. בעוד תמונת פונקציית סף מורכבת מהערכים הבדידים 0 ו-1, תמונת פונקציית סיגמויד הינה הטווח הרציף בין 0 ל-1. כמו כן, פונקציית סיגמויד הינה פונקציה גזירה בעוד פונקציית סף איננה.

תיאור סכמתי:



איור 12

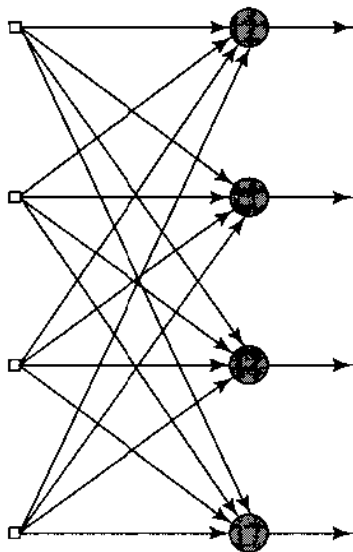
ארכיטקטורות רשת

אופן הבנייה והקישוריות בין הניורונים המלאכותיים תלוי בצורה חזקה באלגוריתם הלמידה בו משתמשים כדי לאמן את הרשת. אלגוריתמי למידה שונים יוצגו בהמשך. בתת-פרק זה אנו מתמקדים בארכיטקטורות הרשת (מבנה הרשת).

באופן כללי, ישנם 3 סוגים בסיסיים של ארכיטקטורות רשת:

1. רשתות חד-שכבתיות ללא משוב

כאשר מדברים על רשת בעלת שכבות אנו מדברים על רשת שבה הניורונים מאורגנים בצורת שכבות אשר קשורות ביניהן. בצורתה הפשוטה ביותר של רשת בעלת שכבות היא מכילה שכבת-קלט של צמתי מקור (צמתים עליהם נישא אות הקלט) אשר מעבירה מידע לשכבת-פלט של ניורונים (צמתי חישוב), אך מידע לא יכול לזרום בכיוון ההפוך. במילים אחרות, זוהי רשת א-ציקלית או חסרת-משוב (feedforward). באיור 13 מוצגת רשת בעלת 4 צמתי-מקור ו-4 צמתי חישוב (ניורונים). רשת כזו נקראת "חד-שכבתית" שכן יש לה רק שכבת חישוב אחת – שכבת הפלט.



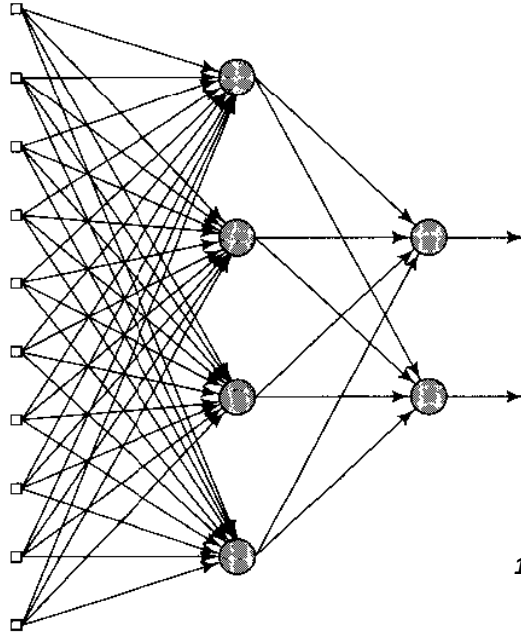
איור 13

2. רשתות רב-שכבתיות ללא משוב

הגרסה השנייה של רשתות ללא משוב נבדלת מהראשונה בכך שהיא מכילה שכבות-חישוב נוספות הנקראות שכבות חבויות (hidden layers). בהתאם לכך, הניורונים בכל שכבה כזו נקראים ניורונים חבויים (hidden neurons) או יחידות חבויות (hidden units). תפקיד השכבות החבויות הוא "להתערב" בין שכבת הקלט ושכבת הפלט באיזשהו אופן שימושי. ע"י הוספת שכבות חבויות הרשת מסוגלת להפיק סטטיסטיקות מסדר גבוה יותר. תכונה זו חשובה במיוחד כאשר שכבת הקלט מכילה הרבה צמתים.

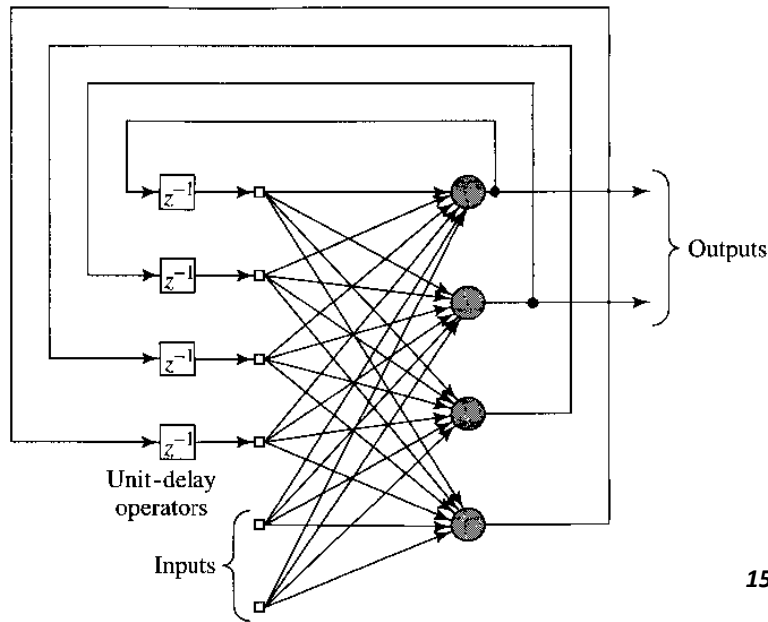
צמתי המקור בשכבת הקלט מספקים את האלמנטים של דפוס ההפעלה (וקטור הקלט) אשר מרכיבים את אותות הקלט לניורונים בשכבה השנייה (השכבה החבויה הראשונה). אותות הפלט של הניורונים בשכבה השנייה הופכים לאותות הפלט של השכבה השלישית, וכך הלאה. בדרך כלל, הניורונים בכל שכבה מקבלים את אותות הקלט שלהם רק מהשכבה הקודמת להם (ולכן

זוהי רשת ללא משוב). אותות הפלט של השכבה האחרונה (שכבת הפלט) מהווים את התגובה הכוללת של הרשת לאותות הקלט אשר התקבלו בעזרת השכבה הראשונה. איור 14 מדגים את הפריסה של רשת רב-שכבתית בעלת שכבה חבויה אחת. ניתן לתאר את הרשת הזו ע"י צירוף המספרים 10-4-2 שכן יש לה 10 צמתי מקור, 4 נוירונים חבויים ו-2 נוירוני פלט. כמו כן, רשת זו מחוברת בצורה מלאה: כל נוירון מחובר לכל אחד מהנוירונים בשכבה הבאה. אם לרשת אין את התכונה הזו, היא נקראת רשת מחוברת-חלקית.



איור 14

3. רשתות חוזרות (recurrent) רשות חוזרות נבדלות מהרשתות חסרות המשוב בכך שיש להן לולאות משוב (feedback loops). באיור 15 ניתן לראות רשת חוזרת רב-שכבתית. ברשת זו יש משוב-עצמי: הפלט של כל נוירון מועבר בתור קלט לעצמו. ניתן גם לתכנן רשת ללא משוב עצמי בה הפלט של כל נוירון מועבר לנוירון אחר באותה שכבה. לולאות משוב משפיעות מאוד על יכולות הלמידה של הרשת ועל ביצועיה. בלולאות משוב ניתן למצוא ענפים (קשרים סינפטיים) מיוחדים המכילים יחידות מעכבות מידע אשר מטרתן לעכב את האות לפני שהוא מגיע לנוירון הבא (מסומנות ב- z^{-1}). יחידות אלה תורמות לאופי האי-ליניארי של רשתות הנוירונים.

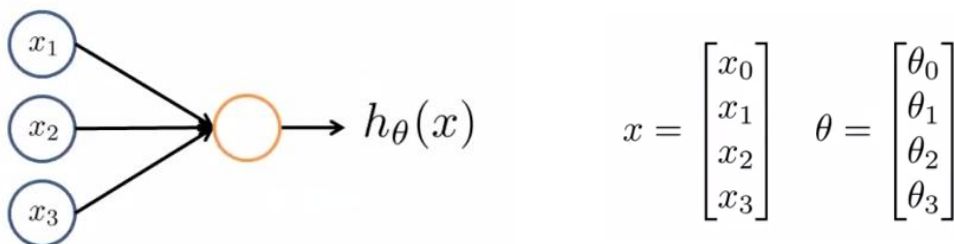


איור 15

ייצוג חלופי

כדי לבצע חישובים מתמטיים על רשת נירונים (חישובים מעין אלו יפורטו בהמשך) נדגים ייצוג חלופי (אך עם זאת מאוד דומה) לייצוג שכבר פורט לעיל [10].

באיור 16 ניתן לראות ייצוג של נירון (ע"י מעגל אדום). יש לו 3 אותות קלט ואות פלט אחד (הפלט של פונקציית ההפעלה)



איור 16

הוקטור x מייצג את אותות הקלט ואילו הוקטור θ מייצג את משקלי הקשתות (המשקלים הסינפטיים). אם, לצורך הדוגמה, נניח שפונקציית ההפעלה שלנו היא הפונקציה הלוגיסטית, אז אות הפלט של הנירון יחושב ע"י:

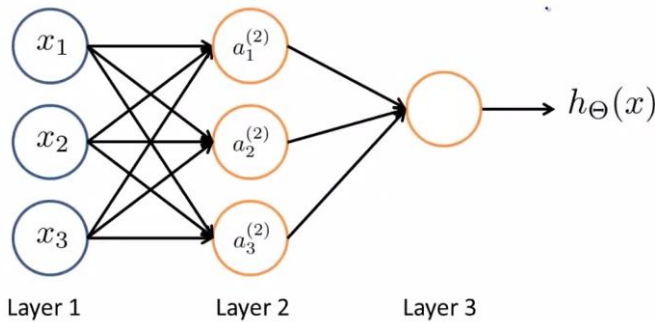
$$(1.12) \quad h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}$$

הבדל בולט בין הייצוג החלופי לייצוג המקורי הוא שכעת פעולות הסכימה והחיבור מתבצעות "בתוך" פונקציית ההפעלה (כפי שניתן לראות, הקלט שלה הוא וקטור המאפיינים ולא מספר ממשי):

$$(1.13) \theta^T x = \sum_{j=0}^m \theta_j x_j$$

כאשר כמות אותות הקלט היא m .

באיור 17 ניתן לראות רשת נוירונים בעלת שכבה חבויה אחת.



איור 17

כפי שנאמר, נוירון יחיד מבצע את חישוביו ע"י שימוש באותות הקלט ובמשקלים הסינפטיים (וקטור המשקלים הסינפטיים). אם ברצוננו לייצג חישובים של שכבה אחת של נוירונים אנו זקוקים לאותות הקלט (נזכור כי הם משותפים בין כל הנוירונים באותה שכבה) ולמטריצה של משקלים סינפטיים המורכבת מכל הוקטורים של המשקלים הסינפטיים של כל נוירון בשכבה.

הסבר לסימונים באיור 17:

$a_i^{(j)}$ – יישום פונקציית ההפעלה של נוירון i בשכבה j . אם $i = 0$ אז זוהי יחידת הטיה (bias) הקיימת תמיד (אפילו אם לא מצוין במפורש) אשר פלטה הוא 1.

$\theta^{(j)}$ – מטריצת המשקלים הסינפטיים בפונקציית המיפוי משכבה j לשכבה $(j+1)$.

הפלט של הרשת באיור 17 יחושב ע"י אוסף המשוואות:

$$a_1^{(2)} = g(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3)$$

$$h_\theta(x) = a_1^{(3)} = g(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)})$$

יש לשים לב שאם לרשת נירונים יש s_j יחידות עיבוד בשכבה j ויחידות בשכבה $j+1$ אז המטריצה $\theta^{(j)}$ תהיה בגודל $(s_{j+1} \times (s_j + 1))$.

על מנת להקל על חישוב הפלט הסופי של הרשת כולה נוסיף סימונים נוספים:

$$a^{(1)} - \text{וקטור הקלט } x.$$

$z_i^{(j)}$ - הקלט של פונקציית ההפעלה g של נירון i בשכבה j . זהו מספר ממשי המחושב ע"י מכפלת הוקטורים $\theta^T x$ של נירון i .

כעת המשוואות לחישוב הפלט נראות כך:

$$a_1^{(2)} = g(z_1^{(2)})$$

$$a_2^{(2)} = g(z_2^{(2)})$$

$$a_3^{(2)} = g(z_3^{(2)})$$

$$h_\theta(x) = a_1^{(3)} = g(z_1^{(3)})$$

בהינתן המטריצות $\theta^{(j)}$ עבור כל שכבה j , חישוב הפלט מתבצע ע"פ הפעולות הסדורות הבאות:

$$z^{(2)} = \theta^{(1)} a^{(1)} = \theta^{(1)} x$$

$$a^{(2)} = g(z^{(2)})$$

$$z^{(3)} = \theta^{(2)} a^{(2)}$$

$$a^{(3)} = g(z^{(3)}) = h_\theta(x)$$

תהליך זה נקרא forward-propagation: הפלט של שכבה מסוימת משמש כקלט לשכבה הבאה ולכן בצורה מטאפורית המידע "מתפשט קדימה".

בעיית XOR

במקרה של רשת נירונים חד-שכבתית אין שכבות חבויות. כתוצאה מכך, רשת כזו לא יכולה לסווג אותות קלט שאינם ניתנים להפרדה ע"י פונקציה ליניארית. תבניות שלא ניתן להפרידן באמצעות פונקציה ליניארית הן נפוצות ורבות. לדוגמה, זהו המצב של בעיית XOR: יש לסווג את 4 הפינות של ריבוע היחידה (ריבוע שקודקודיו הם הנקודות $(0,0), (0,1), (1,1), (1,0)$) כך שזוג קודקודים נגדיים יהיו באותה קטגוריה (class). הזוגות $(0,0)$ ו $(1,1)$ צריכים להיות בקטגוריה 0, כפי שניתן לראות במשוואות:

$$1 \oplus 1 = 0$$

$$0 \oplus 0 = 0$$

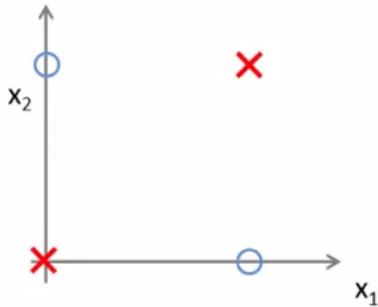
כאשר \oplus הוא אופרטור XOR הבוליאני.

הנקודות (0,0) ו (1,1) נמצאות בפינות נגדיות של הריבוע אך מפיקות את אותו הפלט, 0. הנקודות (0,1) ו (1,0) גם הן בפינות נגדיות ומפיקות את הפלט 1:

$$1 \oplus 0 = 1$$

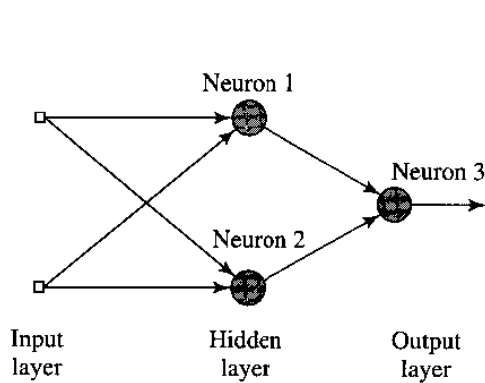
$$0 \oplus 1 = 1$$

נשים לב ששימוש בנירון אחד עם 2 אותות קלט יניב קו ישר בתור גבול ההחלטה (decision boundary). עבור כל הנקודות מצד מסוים של גבול ההחלטה הנירון יוציא 1 כפלט ואילו עבור שאר הנקודות הנירון יוציא 0. המיקום והשיפוע של גבול ההחלטה תלויים במשקלים הסינפטיים של הנירון המחובר לצמתי הקלט והמשקל הסינפטי של צומת הבias. כאשר אותות הקלט (0,0) ו (1,1) נמצאים בפינות נגדיות של ריבוע היחידה וכך גם (0,1) ו (1,0), ברור שלא ניתן לבנות קו ישר אשר יפריד בין 4 הנקודות בצורה הדרושה.

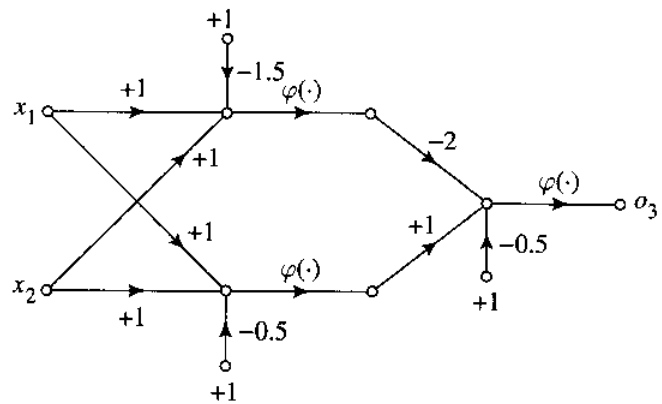


איור 18

ניתן לפתור את הבעיה באמצעות רשת בעלת שכבה חבויה אחת המכילה שני נירונים (איור 19 מציג תיאור סכמתי ואיור 20 מציג גרף של זרימת האותות). כדי לפתור את הבעיה נשתמש בפונקציית הפעלה מסוג פונקציית סף.



איור 19



איור 20

להלן וקטור המשקלים של הנירון העליון (בעל התווית 'Neuron 1') ופונקציית הפלט:

$$\theta_1 = \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix} \quad h_{\theta_1}(x) = \varphi(-1.5 + x_1 + x_2)$$

כאשר φ היא פונקציית הסף.

עבור הנירון התחתון (בעל התווית 'Neuron 2'):

$$\theta_2 = \begin{bmatrix} -0.5 \\ 1 \\ 1 \end{bmatrix} \quad h_{\theta_2}(x) = \varphi(-0.5 + x_1 + x_2)$$

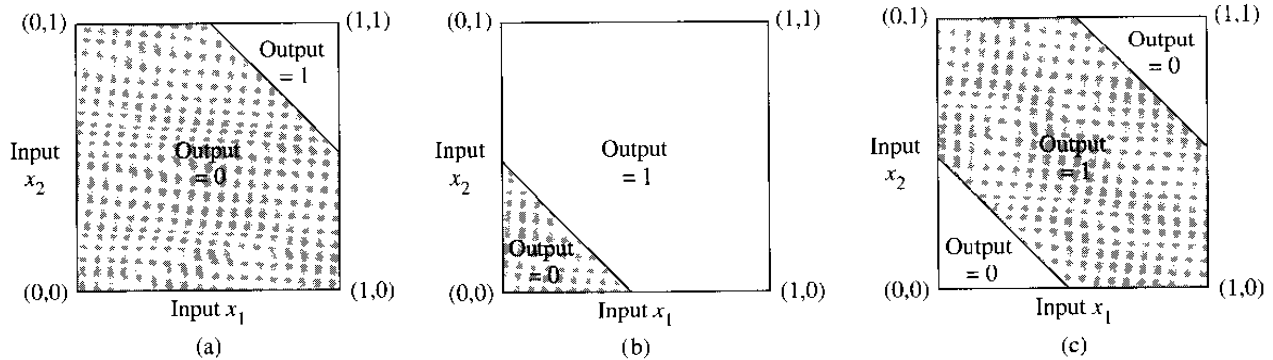
נשים לב שהנירון העליון נותן 1 כפלט רק עבור הנקודה (1,1) ו0 עבור כל שאר הנקודות ואילו הנירון התחתון נותן 0 כפלט רק עבור הנקודה (0,0) ו1 עבור כל שאר הנקודות.

עבור נירון הפלט (בעל התווית 'Neuron 3'):

$$\theta_3 = \begin{bmatrix} -0.5 \\ -2 \\ 1 \end{bmatrix} \quad h_{\theta_3}(x) = \varphi(-0.5 - 2x_1 + x_2)$$

יש לזכור שבמקרה של נירון הפלט, x_1 הוא הפלט של הנירון העליון ו x_2 הוא הפלט של הנירון התחתון.

תפקידו של נירון הפלט הוא ליצור שילוב ליניארי של גבולות ההחלטה אשר נוצרו ע"י שני הנירונים החבויים. אם נקביל את הרשתות המלאכותיים לרשתות ביולוגיות, ניתן לומר שלנירון התחתון יש קשר מגרה (excitatory) לנירון הפלט, ואילו לנירון העליון יש קשר מעכב (inhibitory) לנירון הפלט. כאשר אות הקלט הוא (0,0) שני הנירונים החבויים כבויים (כיוון שהפלט של ניורונים המשתמשים בפונקציית סף הוא 0 או 1, ניתן לכנות אותם "כבויים" או "דלוקים", בהתאמה) וגם נירון הפלט כבוי. כאשר אות הקלט הוא (1,1) שני הנירונים החבויים דלוקים ונירון הפלט כבוי כיוון ש'הכוח' (המקדם שצמוד ל x_1) המעכב של הנירון העליון גדול מ'הכוח' המגרה של הנירון התחתון. כאשר אות הפלט הוא (0,1) או (1,0) אז הנירון העליון כבוי והנירון התחתון דלוק. כתוצאה מכך גם נירון הפלט דלוק. זהו הפתרון של בעיית XOR. באיור 21 ניתן למצוא תיאור סכמתי של גבולות ההחלטה של הנירון העליון (a), הנירון התחתון (b) ונירון הפלט (c).



איור 21

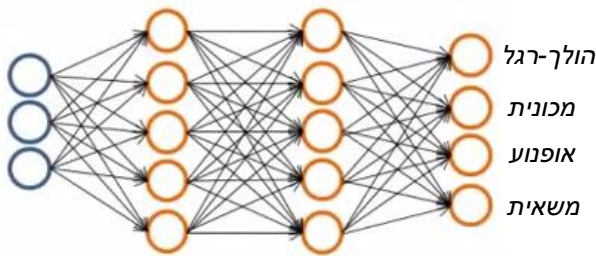
סוגי בעיות

רשתות נוירונים הן כלי מצוין למגוון בעיות [5]. ביניהן:

בעיות התאמה: בעיות שבהן יש להחליט, בהינתן הקלט, מהו הערך המספרי אשר הכי "מתאים" לקלט (המשמעות של ערך זה כמובן תלויה בבעיה המדוברת). לדוגמה, בעיית מחירי הבתים: נתונים מאפיינים של בית (שטח, מספר חדרים, גודל חצר וכדומה) והפלט הוא המחיר שבו יימכר הבית הזה. במקרה זה $y \in R$. (נזכיר כי y הוא הפלט של הרשת)

בעיות סיווג: בעיות שבהן יש להחליט האם הקלט נופל תחת קטגוריה (מחלקה) מסוימת או לא. לדוגמה, בעיית זיהוי תמונה: נתונה תמונה ויש לקבוע האם התמונה הינה תמונה של מכונית (לצורך העניין) או לא. במקרה זה $y \in \{0,1\}$.

בעיות סיווג מרובה-מחלקות: בעיות שבהן יש להחליט לאיזו קטגוריה (מחלקה), מבין הקיימות, שייך הקלט. לדוגמה, גרסה קצת יותר מסובכת של בעיית זיהוי תמונה: נתונה תמונה ויש לקבוע האם התמונה הינה תמונה של מכונית, של הולך-רגל, של אופנוע או של משאית. להלן רשת פוטנציאלית לפתרון הבעיה הזו:



איור 22

ברשת זו ישנם 4 נוירוני-פלט אשר פלטם הוא 0 או 1 (שכן זוהי בעיית סיווג). ליד כל נוירון פלט מצוינת המחלקה אותה "מייצג" הנוירון. במקרה זה $y \in \{0,1\}^4$.

סט האימון יורכב מזוגות (x, y) כאשר x הוא תמונה (מטריצת פיקסלים) ו- y הוא וקטור בינארי בגודל 4, כאשר:

$$\begin{matrix} \text{מייצג משאית} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \text{מייצג אופנוע} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \text{מייצג מכונית} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \text{מייצג הולך-רגל} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{matrix}$$

ללא קשר לסוג הבעיה, תהליך הפתרון שאמצעות רשת נוירונים הוא זהה: יש למצוא כמות (גדולה כמה שיותר) של זוגות (פלט, קלט) אשר ישמשו כסט אימון. ככל שיהיו יותר זוגות בסט האימון כך הרשת תוכל לפתור את הבעיה בצורה טובה יותר. לאחר מכן יש להעריך את רמת הדיוק (ולתקן במידת הנדרש) של הרשת האמצעות זוגות (פלט, קלט) אשר יהוו את סט הבדיקה. לאחר מכן ניתן להשתמש ברשת על הבעיה האמתית שברצוננו לפתור.

כוחן הגדול של רשתות נוירונים מקורו ביכולת הלמידה וההסתגלות שלהן לסביבה. בפרק זה נבחן את הדרך בה רשתות נוירונים יכולות לעשות זו.

הפלט של רשת הנוירון תלוי (מלבד בקלט כמובן) במשקלים הסינפטיים שלה (המטריצות $\theta^{(j)}$). הרעיון הכללי הוא להזין לרשת כמה שיותר דוגמאות קונקרטיות (דוגמאות אלה מרכיבות את סט האימון) כאשר בכל פעם שנותנים לרשת "ללמוד" דוגמה, המשקלים הסינפטיים משתנים על מנת שהפלט של הרשת יתאים כמה שיותר לכל הדוגמאות שנלמדו עד עכשיו. הדרך לעשות זו הינה להגדיר "פונקציית עלות" אשר נמצאת ביחס ישר עם השגיאה של הרשת (קרי, כמה הרשת עובדת "רע") ולמצוא את המינימום שלה (וע"י כך לגרום לרשת לעבוד "טוב" במידת האפשר).

פרק זה הינו טכני ומתמטי, ועל כן, כדי להקל על הקורא, להלן רוב המושגים אשר ישמשו אותנו במהלך הפרק:

- האינדקסים i, j, k הינם אינדקסים של נוירונים שונים ברשת כאשר האותות מתפשטים משמאל לימין: נוירון j נמצא בשכבה מימין לשכבה שבה נמצא i ונוירון k נמצא בשכבה מימין לשכבה בה נמצא נוירון j .
- כאשר אנו מדברים על איטרציה n הכוונה היא שהדוגמה ה- n מסט האימון כרגע "נלמדת" ע"י הרשת.
- הסימון $J(n)$ מסמן את סכום ריבועי-השגיאות (error squares) או אנרגיית השגיאה באיטרציה n . הממוצע של $J(n)$ על גבי כל הערכים של n (כל סט האימון) מניב את אנרגיית השגיאה הממוצעת $J_{av}(n)$.
- הסימון $e_j(n)$ מסמן את אות השגיאה (גודל השגיאה) של נוירון j באיטרציה n .
- הסימון $d_j(n)$ מסמן את התגובה הרצויה של נוירון j באיטרציה n . בעזרתו מחשבים את $e_j(n)$.
- הסימון $y_j(n)$ מסמן את הפלט של נוירון j באיטרציה n .
- הסימון $w_{ji}(n)$ מסמן את המשקל הסינפטי בין הפלט של נוירון i לקלט של נוירון j באיטרציה n . התיקון שיש ליישם במשקל זה באיטרציה n מסומן ב- $\Delta w_{ji}(n)$.
- הסימון $v_j(n)$ מסמן את פוטנציאל ההפעלה (סכום המכפלות של אותות הקלט במשקלים הסינפטיים, כולל אות ה-bias) של נוירון j באיטרציה n . הערך הזה הינו הקלט של פונקציית ההפעלה של נוירון j .
- הסימון $\varphi_j(\cdot)$ מסמן את פונקציית ההפעלה של נוירון j .
- הסימון b_j מסמן את גודל אות ההטיה (bias) של נוירון j . השפעתו על הפלט נובעת מסינפסה בעלת משקל $w_{j0} = b_j$ המחוברת לקלט קבוע שערכו $+1$.
- הסימון $x_i(n)$ מסמן את האלמנט ה- i של וקטור הקלט באיטרציה n .
- הסימון $o_k(n)$ מסמן את האלמנט ה- k של וקטור הפלט באיטרציה n .
- הסימון η מסמן את פרמטר קצב הלמידה של הרשת.
- הסימון m_l מסמן את הגודל (כמות הצמתים) של שכבה l ברשת. $l = 0, 1, \dots, L$ כאשר L הינו "עומק" הרשת (כמות השכבות). m_0 הינו גודל שכבת הקלט (גודל וקטור הקלט x), m_1 הינו הגודל של השכבה החבויה הראשונה ו- m_L הינו הגודל של שכבת הפלט. לעיתים נשתמש בסימון $m_L = M$.

פונקציית העלות

אות השגיאה של הפלט של נירון j באיטרציה n מוגדר ע"י הנוסחה:

$$(2.1) e_j(n) = d_j(n) - y_j(n)$$

אנו מגדירים את ערך אנרגיית השגיאה של נירון j ע"י הביטוי $\frac{1}{2}e_j^2(n)$. בהתאם לכך, הערך $J(n)$ (סכום אנרגיית השגיאה באיטרציה n) מחושב ע"י סכימת $\frac{1}{2}e_j^2(n)$ על פני כל נירוני הפלט. אלה הם הניורונים ה"נראים" (לא חבויים) ולכן ניתן לחשב את גודל שגיאתם בצורה ישירה:

$$(2.2) J(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

כאשר C מסמן את קבוצת כל הניורונים בשכבת הפלט. נסמן ב- N את מספר כל הדוגמאות בסט האימון (גודל סט האימון). אנרגיית ריבועי השגיאות הממוצעת מחושבת ע"י סכימת כל $J(n)$ ונרמול ביחס ל- N :

$$(2.3) J_{av} = \frac{1}{N} \sum_{n=1}^N J(n)$$

הערך $J(n)$, וכך גם הערך J_{av} , הוא פונקציה של הפרמטרים החופשיים (המשקלים הסינפטיים ומשקלי bias) של הרשת. עבור סט אימון מסוים, J_{av} מייצג את פונקציית העלות כאמת מידה לביצועי הרשת. המטרה של תהליך הלמידה היא לכוון את הפרמטרים החופשיים על מנת להגיע למינימום של הערך J_{av} .

אלגוריתם back-propagation

כדי להביא את J_{av} למינימום משתמשים באלגוריתם הנקרא back-propagation. הרעיון הוא לעדכן את המשקלים הסינפטיים, דוגמה אחר דוגמה, עד סוף תקופה מסוימת (epoch). תקופה זו מייצגת את מספר המעברים הנדרש להגיע להצגה מלאה של כל סט האימון לרשת הניורונים. עדכון המשקלים נעשה ביחס לשגיאות המחושבות עבור כל אחת מהדוגמאות בסט האימון.

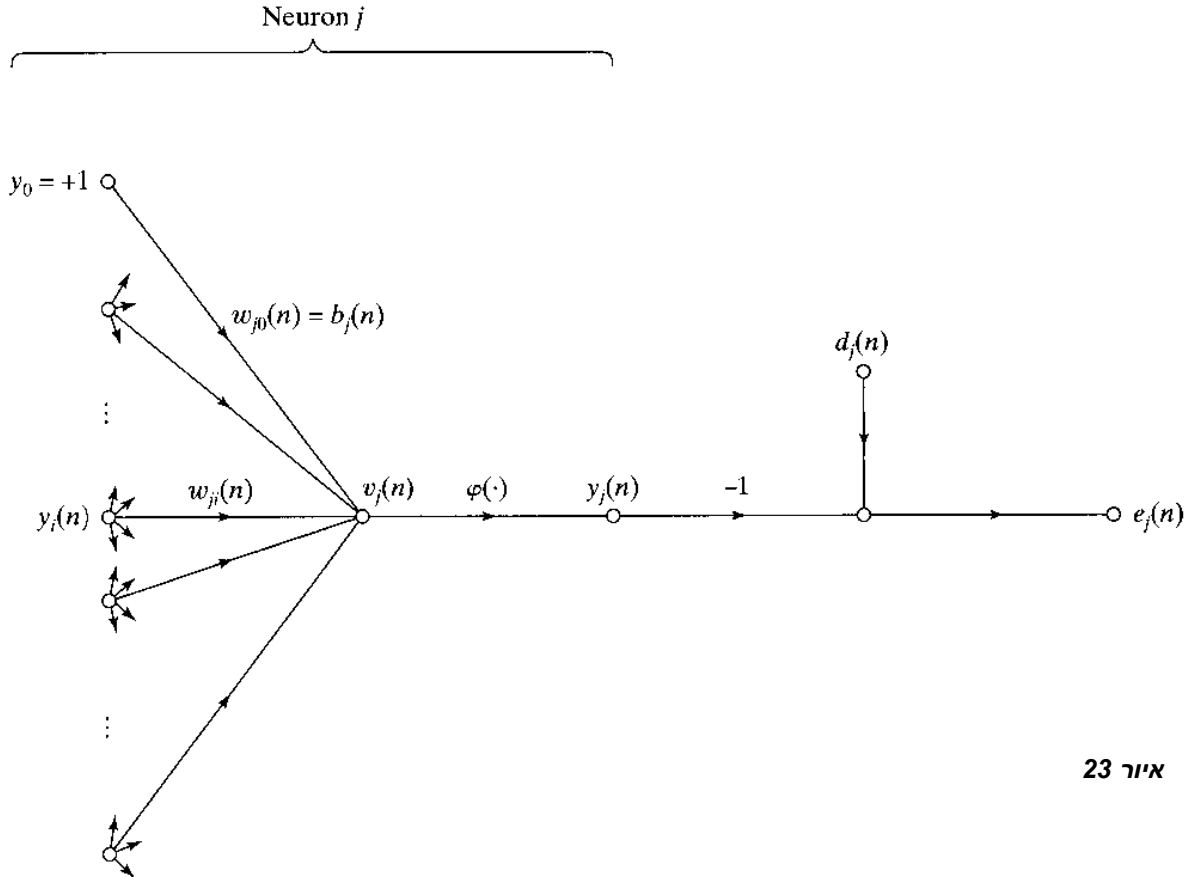
הממוצע החשבוני של שינויי המשקל הללו תוך כדי מעבר על סט האימון הוא הערכה של השינוי האמתי שיקרה כתוצאה משינוי משקלים המבוסס על הבאה למינימום של הפונקציה J_{av} .

באיור 23 ניתן לראות נירון j אשר מוזנים אליו הפלטים של כל הניורונים המחוברים אליו משמאל. פוטנציאל הפעלה, הקלט לפונקציית הפעלה, של נירון j הוא:

$$(2.4) v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n)$$

כאשר m מסמן את כמות הקלטים (לא כולל bias) של נוירון j . הפלט של נוירון j הוא:

$$(2.5) y_j(n) = \varphi_j(v_j(n))$$



איור 23

אלגוריתם back-propagation מיישם תיקון $\Delta w_{ji}(n)$ על המשקל $w_{ji}(n)$. התיקון הוא פרופורציונאלי לנגזרת החלקית $\partial J(n) / \partial w_{ji}(n)$. ע"פ כלל השרשרת של הגזירה ניתן להביע את השיפוע (gradient) בתור:

$$(2.6) \frac{\partial J(n)}{\partial w_{ji}(n)} = \frac{\partial J(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

הנגזרת החלקית $\partial J(n) / \partial w_{ji}(n)$ מייצגת גורם רגישות, הקובע באיזה כיוון יש לחפש את המשקל הסינפטי $w_{ji}(n)$ במרחב כל המשקלים כדי להביא למינימום את J_{av} . גזירה חלקית של שני הצדדים של משוואה (2.2) ביחס ל $e_j(n)$ תניב:

$$(2.7) \frac{\partial J(n)}{\partial e_j(n)} = e_j(n)$$

גזירה חלקית של שני הצדדים של משוואה (2.1) ביחס ל $y_j(n)$ תניב:

$$(2.8) \frac{\partial e_j(n)}{\partial y_j(n)} = -1$$

גזירה חלקית של שני הצדדים של משוואה (2.5) ביחס ל $v_j(n)$ תניב:

$$(2.9) \frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n))$$

ולבסוף, גזירה חלקית של שני הצדדים של משוואה (2.4) ביחס ל $w_{ji}(n)$ תניב:

$$(2.10) \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

הצבה של משוואות (2.7) עד (2.10) במשוואה (2.6) מניבה:

$$(2.11) \frac{\partial J(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi_j'(v_j(n)) y_i(n)$$

התיקון $\Delta w_{ji}(n)$ על המשקל $w_{ji}(n)$ מוגדר ע"י כלל הדלתא:

$$(2.12) \Delta w_{ji}(n) = -\eta \frac{\partial J(n)}{\partial w_{ji}(n)}$$

כאשר η הוא פרמטר קצב הלמידה של האלגוריתם. משתמשים בסימן מינוס במשוואה (2.12) כדי ליצור שיפוע יורד (gradient descent) במרחב המשקלים (מחפשים כיוון לשינוי במשקל שיפחית את הערך של J_{av}). לפיכך, שימוש במשוואות (2.11) ו(2.12) מניב:

$$(2.13) \Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

כאשר השיפוע המקומי (local gradient) $\delta_j(n)$ מוגדר כך:

$$(2.14) \delta_j(n) = -\frac{\partial J(n)}{\partial v_j(n)} = -\frac{\partial J(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \varphi_j'(v_j(n))$$

השיפוע המקומי מצביע לשינויים הנדרשים במשקלים הסינפטיים. לפי משוואה (2.14), השיפוע המקומי $\delta_j(n)$ של נוירון j הוא המכפלה בין אות השגיאה $e_j(n)$ (כמה j "טועה") לבין הנגזרת $\varphi_j'(v_j(n))$ של פונקציית הפעלה של j (שיפוע פונקציית הפלט של j).

ממשוואות (2.13) ו(2.14) אנו למדים שחלק גדול בחישוב התיקון $\Delta w_{ji}(n)$ הוא אות השגיאה $e_j(n)$ בפלט של נוירון j . בהקשר זה, אנו יכולים לזהות שני מקרים נפרדים כתלות במיקומו של הנוירון j ברשת. במקרה 1, הנוירון j הוא נוירון פלט. המקרה הזה ניתן לטיפול בפשטות יחסית שכן אנו יודעים בדיוק מה התגובה שהרצויה של כל אחת מיחידות הפלט (שכן אנו סיפקנו את סט האימון, הכולל קלטים ופלטים רצויים) ולכן ניתן לחשב בצורה ישירה את $e_j(n)$. במקרה 2, הנוירון j הוא נוירון חבוי. למרות שהנוירונים החבויים אינם נגישים בצורה ישירה, הם חולקים את האחריות של כל שגיאה הנעשית בפלט הרשת

כולה. כעת רק נותרה המטלה של חלוקת עונשים או מתנות (בצורה מטפורית) על חלקם של הניורונים החבויים באחריות זו. סוגיה זו נפתרת בצורה אלגנטית ע"י גרימה לכל אותות השגיאה משכבות מימין להתפשט אחורה (back-propagate) לשכבות משמאל.

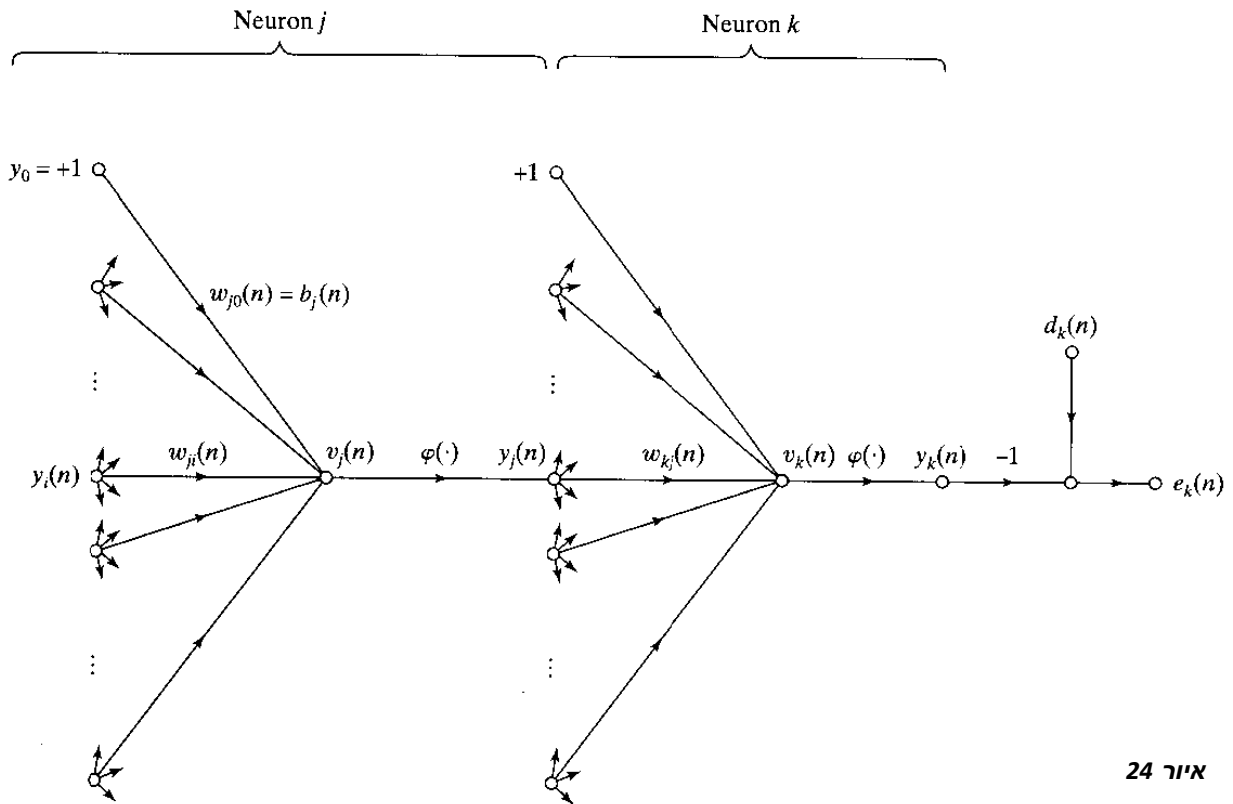
מקרה 1: נירון j הוא נירון פלט

כאשר נירון j ממוקם בשכבת הפלט של הרשת, הפלט הרצוי ידוע מראש. ניתן להשתמש במשוואה (2.1) כדי לחשב את אות השגיאה $e_j(n)$ של נירון זה. לאחר שאות השגיאה נקבע, זה עניין חישובי פשוט למצוא את השיפוע המקומי $\delta_j(n)$ בעזרת משוואה (2.14).

מקרה 2: נירון j הוא נירון חבוי

כאשר נירון j ממוקם בשכבה חבויה, אין פלט רצוי כלשהו שניתן להצביע עליו. לפיכך, אות השגיאה של נירון חבוי צריך להיקבע בצורה רקורסיבית כתלות באותות השגיאה של הניורונים אליהם נירון j מחובר בצורה ישירה. כאן הדברים מסתבכים. ראו לדוגמה את המצב המוצג באיור 24: נירון j הוא נירון חבוי ברשת. לפי משוואה (2.14) רשאים אנו להגדיר את השיפוע המקומי $\delta_j(n)$ עבור הניורון החבוי j כך:

$$(2.15) \delta_j(n) = -\frac{\partial J(n)}{\partial v_j(n)} = -\frac{\partial J(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial J(n)}{\partial y_j(n)} \varphi_j'(v_j(n))$$



איור 24

כדי לחשב את הנגזרת החלקית $\partial J(n)/\partial y_j(n)$, יכולים אנו לנסות את הדרך הבאה:
לפי איור 24 ניתן לראות כי:

$$(2.16) J(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n)$$

משוואה זו זהה למשוואה (2.2) למעט החלפת האינדקס j באינדקס k . החלפה זו נעשתה כדי למנוע בלבול עם השימוש באינדקס j שמתייחס לנירון חבוי במקרה 2 (במקרה 1 נירון j היה נירון פלט). גזירה חלקית של משוואה (2.16) ביחס לאות הפלט $y_j(n)$ תניב:

$$(2.17) \frac{\partial J(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)}$$

נעת נשתמש בכלל השרשרת כדי לחשב את הנגזרת החלקית $\partial e_k(n)/\partial y_j(n)$ ונכתוב את משוואה (2.17) בצורה השקולה:

$$(2.18) \frac{\partial J(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)}$$

לפי איור 24 ניתן לראות כי:

$$(2.19) e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n))$$

ולכן:

$$(2.20) \frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi_k'(v_k(n))$$

נזכיר כי פוטנציאל ההפעלה של נירון k הוא:

$$(2.21) v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n)$$

כאשר m מסמן את כמות הקלטות (לא כולל bias) של נירון k . גזירה חלקית של משוואה (2.21) ביחס ל $y_j(n)$ תניב:

$$(2.22) \frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

ע"י הצבה של משוואות (2.20) ו(2.22) במשוואה (2.18) נקבל את הנגזרת החלקית הדרושה:

$$(2.23) \frac{\partial J(n)}{\partial y_j(n)} = - \sum_k e_k(n) \varphi_k'(v_k(n)) w_{kj}(n) = - \sum_k \delta_k(n) w_{kj}(n)$$

כאשר במעבר האחרון השתמשנו בהגדרה של השיפוע המקומי $\delta_k(n)$ אשר הובאה במשוואה (2.14) כאשר האינדקס k מחליף את האינדקס j מהמשוואה המקורית.

בעזרת הצבת משוואה (2.23) במשוואה (2.15), נקבל את הנוסחה הרקורסיבית, הידועה בשם *נוסחת ההתפשטות לאחור* (*back-propagation formula*), עבור השיפוע המקומי $\delta_j(n)$ של נירון חבוי j :

$$(2.24) \quad \delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

הגורם $\varphi_j'(v_j(n))$, המעורב בחישוב במשוואה (2.24), תלוי אך ורק בפונקציית ההפעלה של הנירון j . הגורם האחר המעורב בחישוב זה, הסכימה על k , תלוי בשני סטים של גורמים. סט הגורמים הראשון, $\delta_k(n)$, דורש לדעת את אותות השגיאה $e_k(n)$, עבור כל הנירונים הנמצאים בשכבה הימנית המידית של הנירון j , ומחוברים בצורה ישירה לנירון j . סט הגורמים השני, $w_{kj}(n)$, מורכב מהמשקלים הסינפטיים של אותם חיבורים.

כעת נסכם בקצרה את השיקולים בפיתוח האלגוריתם back-propagation. ראשית, התיקון $\Delta w_{ji}(n)$ אשר מיושם למשקל הסינפטי של החיבור בין נירון i לנירון j מוגדר ע"פ כלל הדלתא:

$$(2.25) \quad \begin{pmatrix} \text{Weight} \\ \text{correction} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{learning-} \\ \text{rate parameter} \\ \eta \end{pmatrix} \cdot \begin{pmatrix} \text{local} \\ \text{gradient} \\ \delta_j(n) \end{pmatrix} \cdot \begin{pmatrix} \text{input signal} \\ \text{of neuron } j \\ y_i(n) \end{pmatrix}$$

שנית, השיפוע המקומי $\delta_j(n)$ תלוי במיקום הנירון j ברשת (האם הוא נירון פלט או נירון חבוי):

1. אם נירון j הוא נירון פלט, $\delta_j(n)$ שווה למכפלה בין הנגזרת $\varphi_j'(v_j(n))$ לאות השגיאה $e_j(n)$. שני הגורמים קשורים אך ורק לנירון j עצמו. ראה משוואה (2.14).

2. אם נירון j הוא נירון חבוי, $\delta_j(n)$ שווה למכפלה בין הנגזרת $\varphi_j'(v_j(n))$ לסכום המשוקלל של כל ה- δ של הנירונים ברמה הבאה מימין שמחוברים ישירות לנירון j . ראה משוואה (2.24).

שני מעברי חישוב

הפעלה הלכה למעשה של אלגוריתם back-propagation טומנת בחובה שני מעברי חישוב שונים. המעבר הראשון נקרא מעבר-קדימה והמעבר השני נקרא מעבר-אחורה.

במעבר-קדימה המשקלים הסינפטיים נשארים ללא שינוי לאורך כל הרשת ואותות הפלט של הנירונים מחושבים נירון-אחר-נירון. אות הפלט של נירון j מחושב כך:

$$(2.26) \quad y_j(n) = \varphi_j(v_j(n))$$

כאשר $v_j(n)$ הוא פוטנציאל ההפעלה של נירון j ומוגדר כך:

$$(2.27) v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n)$$

כאשר m מסמן את כמות הקלטים (לא כולל הביאס) של נירון j ו- $w_{ji}(n)$ הוא המשקל הסינפטי של החיבור בין נירון i לנירון j . $y_i(n)$ הוא אות הקלט של נירון j (וגם אות הפלט של נירון i) בחיבור בין i ל- j . אם j נמצא בשכבה החבויה הראשונה, אז $m = m_0$ והאינדקס i מתייחס לאלמנט במיקום i של וקטור הקלט x :

$$(2.28) y_i(n) = x_i(n)$$

מצד שני, אם נירון j נמצא בשכבת הפלט, אז $m = m_L$ והאינדקס j מתייחס לאלמנט במיקום j של וקטור הפלט y :

$$(2.29) y_j(n) = o_j(n)$$

ההפרש בין הפלט לתגובה הרצויה $d_j(n)$ מגדיר את אות השגיאה $e_j(n)$ של נירון הפלט במיקום j . המעבר-קדימה מתחיל בשכבה החבויה הראשונה, ע"י הזנת וקטור הקלט לנירונים בה, ומסתיים בשכבת הפלט ע"י חישוב אות השגיאה של כל נירון בשכבה זו.

בניגוד לכך, המעבר-אחורה מתחיל בשכבת הפלט ע"י העברת אותות השגיאה לכיוון שמאל לאורך כל הרשת, שכבה אחר שכבה, ומחשב באופן רקורסיבי את ה- δ (השיפוע המקומי) של כל נירון. המעבר הרקורסיבי עלול לגרום לשינויים במשקלים הסינפטיים של הרשת כתוצאה מהפעלת כלל הדלתא במשוואה (2.25). עבור נירון בשכבת הפלט, השיפוע המקומי הוא פשוט המכפלה של אות השגיאה שלו והנגזרת הראשונה של פונקציית ההפעלה שלו. כעת, אנו יכולים להשתמש במשוואה (2.25) כדי לחשב את השינויים במשקלים של כל החיבורים שמסתיימים בשכבת הפלט. כיוון שאנו יודעים את הערך של השיפועים המקומיים של כל הנירונים בשכבת הפלט, אנו יכולים להשתמש במשוואה (2.24) כדי לחשב את כל השיפועים המקומיים של השכבה המידית משמאל וכך גם את כל השינויים במשקלים של השכבה הזו. החישוב הרקורסיבי ממשיך, שכבה אחר שכבה, ע"י התפשטות אחורנית של השינויים לכל המשקלים הסינפטיים ברשת.

יש לציין שעבור הצגה של דוגמה אחת מסט האימון, הקלט נשאר קבוע ("נעול") לאורך שני המעברים.

פונקציית ההפעלה

חישוב השיפוע המקומי של כל נירון דורש לדעת את הנגזרת של פונקציית ההפעלה $\varphi(\cdot)$ של אותו נירון. כדי שנגזרת זו תהיה קיימת אנו דורשים שפונקציית ההפעלה תהיה רציפה. דוגמה לסוג פונקציה רציפה אי-ליניארית הנמצא בשימוש רחב בתחום רשתות הנירונים היא אי-ליניאריות סיגמוידיאלית (*sigmoidal nonlinearity*). להלן תיאור של 2 פונקציות כאלה:

1. הפונקציה הלוגיסטית (logistic function) – הצורה הכללית של אי-ליניאריות סיגמוידיאלית זו מוגדרת ע"י:

$$(2.30) \varphi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))} \quad a > 0; -\infty < v_j(n) < \infty$$

ע"פ אי-ליניאריות זו, האמפילטודה של אות הפלט שוכן בטווח $0 \leq y_j \leq 1$. גזירה של (2.30) ביחס ל $v_j(n)$ תניב:

$$(2.31) \varphi_j'(v_j(n)) = \frac{a \exp(-av_j(n))}{(1 + \exp(-av_j(n)))^2}$$

ניתן לפשט את משוואה (2.31) באמצעות משוואות (2.26) ו(2.30):

$$(2.32) \varphi_j'(v_j(n)) = ay_j(n)[1 - y_j(n)]$$

עבור נירון j הממוקם בשכבת הפלט, $y_j(n) = o_j(n)$ ולכן ניתן להביע את השיפוע המקומי שלו כך:

$$(2.33) \delta_j(n) = e_j(n)\varphi_j'(v_j(n)) = a[d_j(n) - o_j(n)]o_j(n)[1 - o_j(n)]$$

עבור נירון j שהינו נירון חבוי ניתן להביע את השיפוע המקומי כך:

$$(2.34) \delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n)w_{kj}(n) = ay_j(n)[1 - y_j(n)] \sum_k \delta_k(n)w_{kj}(n)$$

ניתוח בסיסי של משוואה (2.32) מגלה שהמקסימום שלה מתקבל כאשר $y_j(n) = 0.5$, והמינימום מתקבל כאשר $y_j(n) = 1$ או $y_j(n) = 0$. כיוון שגודל השינוי במשקלים הסינפטיים פרופורציונאלי לנגזרת של פונקציית ההפעלה, ניתן להסיק שעבור פונקציות סיגמויד רוב השינוי במשקלים הסינפטיים יקרה היכן שאותות הפלט הם באמצע הטווח.

2. פונקציית טנגנס היפרבולית (hyperbolic tangent function) – זוהי עוד דוגמה לצורת אי-ליניאריות סיגמוידיאלית אשר נמצאת בשימוש נרחב. היא מוגדרת כך:

$$(2.35) \varphi_j(v_j(n)) = a \tanh(bv_j(n)) \quad a, b > 0$$

הלכה למעשה, פונקציית הטנגנס ההיפרבולית היא פשוט הפונקציה הלוגיסטית לאחר שינוי קנה מידה. נגזרתה ביחס ל $v_j(n)$ היא:

$$(2.36) \varphi_j'(v_j(n)) = ab \operatorname{sech}^2(bv_j(n)) = ab(1 - \tanh^2(bv_j(n))) \\ = \frac{b}{a}[a - y_j(n)][a + y_j(n)]$$

עבור נירון j בשכבת הפלט, השיפוע המקומי הוא:

$$(2.37) \delta_j(n) = e_j(n)\varphi_j'(v_j(n)) = \frac{b}{a}[d_j(n) - o_j(n)][a - o_j(n)][a + o_j(n)]$$

עבור נירון חבוי j הנגזרת המקומית היא:

$$(2.38) \delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n)w_{kj}(n) = \frac{b}{a}[a - y_j(n)][a + y_j(n)] \sum_k \delta_k(n)w_{kj}(n)$$

קריטריון עצירה

באופן כללי, לא ניתן להראות שאלגוריתם back-propagation מתכנס ואין קריטריון מוגדר-היטב לעצירתו. עם זאת, ישנם כמה קריטריונים סבירים שניתן להשתמש בהם כדי לעצור את עדכון המשקלים. כדי לנסח כזה קריטריון, הגיוני לחשוב על התכונות הייחודיות של מינימום גלובלי או לוקאלי של פני שטח השגיאה. נסמן את וקטור המשקלים בו מתקבל מינימום (גלובלי או לוקאלי) ב w^*

תכונה ייחודית של נקודת מינימום היא שפונקציית גודל השגיאה $J_{av}(w)$ נשארת ללא שינוי (לאורך הרצת האלגוריתם) כאשר $w = w^*$. אם כך, ניתן להציע את קריטריון העצירה:

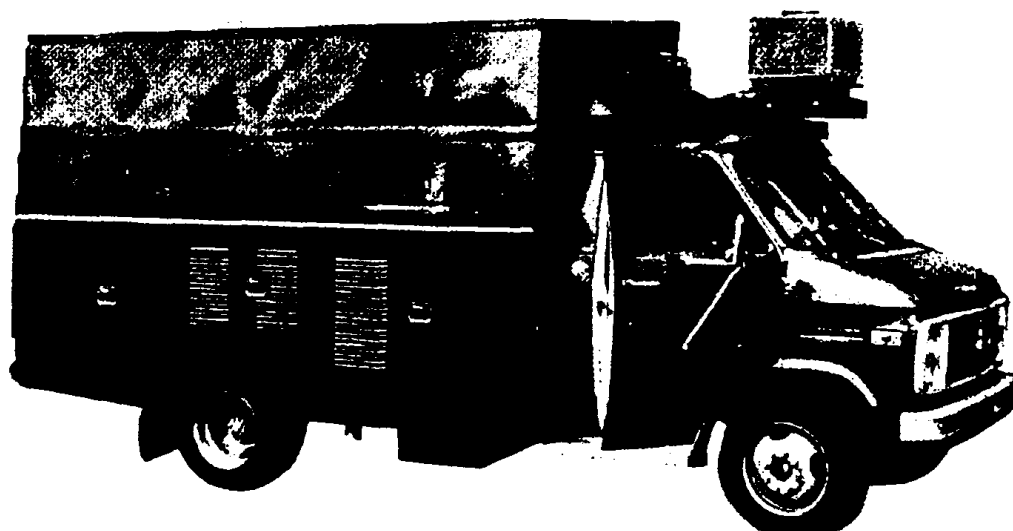
אלגוריתם back-propagation נגיע להתכנסות אם קצב השינוי של ממוצע ריבועי השגיאות הוא קטן בצורה מספקת.

קצב השינוי בממוצע ריבועי השגיאות בדרך כלל נחשב כקטן מספיק אם הוא שוכן בטווח שבין 0.1 ל 1 אחוז. לעיתים משתמשים בערך קטן עד כדי 0.01 אחוז. למרבה הצער, קריטריון זה לפעמים עלול להניב סיום מוקדם מדי של תהליך הלמידה.

הקדמה

ניווט אוטונומי הינו מטלה קשה למערכות ראייה (vision) רובוטיות מסורתיות. הסיבה העיקרית לכך היא חוסר היציבות, הרעש והשונות של העולם האמתי. מערכות ניווט אוטונומי מבוססות על עיבוד תמונה מסורתי עובדות היטב תחת תנאים מאוד מסוימים וחוות בעיות בתנאים אחרים. חלק מהקושי נובע מהעובדה שהעיבוד שמבוצע ע"י מערכות אלו הוא קבוע בסביבות שונות.

רשתות נוירונים מלאכותיות מראות ביצועים וגמישות מרשימים בסביבות בעלות תנאים בעלי רעש ושונות גבוהה, כגון זיהוי כתב-יד [1], זיהוי דיבור וזיהוי פנים [4]. המערכת (ALVINN Autonomous Land Vehicle In a Neural Network) מביאה את הגמישות הדרושה למטלת הניווט האוטונומי [2]. מערכת זו היא מערכת מבוססת רשתות נוירונים שמטרתה לשלוט ברכב המכונה Navlab (איור 25).

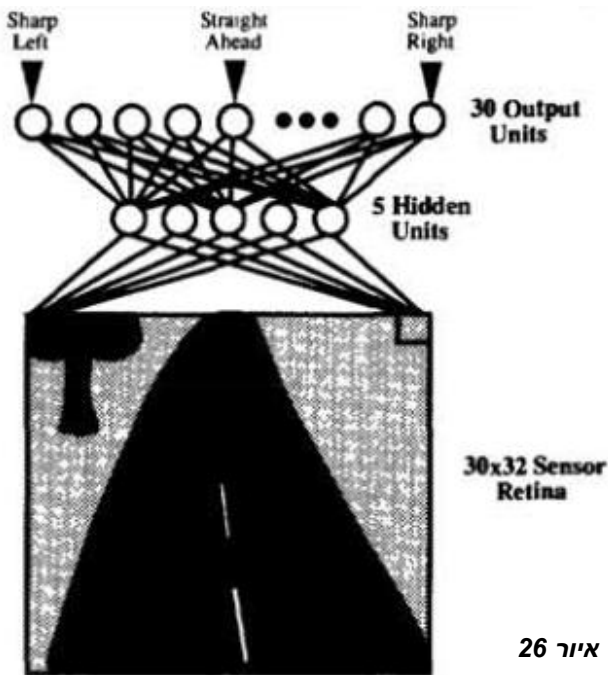


איור 25

בהמשך נתאר את הארכיטקטורה, האימון והביצועים של ALVINN. המערכת מדגימה כיצד רשת פשוטה יחסית יכולה ללמוד בצורה מדויקת כיצד לשלוט ברכב במגוון סיטואציות כאשר היא מאומנת כהלכה. המערכת אומנה באמצעות טכניקות אימון אשר אפשרו לALVINN לנהוג בצורה עצמאית לאחר 5 דקות למידה בלבד בהן המערכת אומנה ע"י התבוננות בנהג אנושי ובתגובותיו לסיטואציות חדשות בכביש. תוך שימוש בטכניקות אלה, ALVINN אומן לנהוג במגוון תנאים הכוללים כביש בעל נתיב אחד עם וללא מדרכה, כבישים בעלי כמה נתיבים עם ובלי קווי הפרדה וגם התחמקות ממכשולים ונהיגת שטח, במהירות של עד 55mph [8].

ארכיטקטורת הרשת

מבין הארכיטקטורות שתוארו בפרק הקודם, הארכיטקטורה הבסיסית של ALVINN שייכת לסוג 2 – רשת רב-שכבתית ללא משוב, כאשר יש שימוש בשכבה חבויה אחת (איור 26). שכבת הקלט מורכבת מ"רשתית" יחידה ברזולוציה של 30X32 פיקסלים אליה מוקרנת תמונה שמקורה במצלמת וידאו או במד טווח לייזר (laser rangefinder). כל אחת מ-960 (30X32) יחידות הקלט מחוברת בצורה מלאה לשכבה החבויה המכילה 5 נירונים (בלבד) וזאתי מחוברת בצורה מלאה לשכבת הפלט. שכבת הפלט מכילה 30 יחידות והיא מהווה ייצוג ליניארי של כיוון ההיגוי הנוכחי כך שהרכב יישאר בנתיב או כדי למנוע ממנו להתנגש במכשולים. היחידה המרכזית מייצגת את התנאי "סע ישר" בעוד שהיחידות משמאל ומימין מייצג פניות שמאלה וימינה חדות יותר ויותר. היחידות הקיצוניות ביותר מימין ומשמאל מייצג פניות בעלות רדיוס של 20 מטר.



איור 26

כדי לנהוג בNavlab, תמונה מהחיישן המתאים עוברת שינוי רזולוציה (image reduction) כדי להתאים של לרזולוציה של 30X32 ואז מוזנת לשכבת הקלט. לאחר התפשטות האות במערכת, אות הפלט מתורגם לפקודת היגוי. מתייחסים לכיוון ההיגוי שהרשת מכתובה בתור מרכז המסה של "גבעה" שמקיפה את יחידת הפלט עם רמת ההפעלה (אות הפלט) החזקה ביותר. שימוש במרכז מסת ההפעלה במקום ביחידת הפלט הפעילה ביותר כדי לקבוע את כיוון ההיגוי נותן תיקוני היגוי "עדינים" יותר וכך משפר את דיוק הנהיגה. לדוגמה, ניתן לחשוב על מעבר נתיב: ניתן להזיז את ההגה בבת אחת לזווית הדרושה כדי להגיע לנתיב הסמוך (חוויית נהיגה לא טובה) וניתן להעלות את זווית ההיגוי בהדרגה לכיוון הנתיב הסמוך ובכך ליצור חוויית נהיגה טובה יותר.

אימון הרשת

הרשת אומנה כדי להפיק את כיוון ההיגוי הדרוש באמצעות אלגוריתם back-propagation. כפי שנאמר, אות הקלט ראשית מתפשט קדימה (מעבר 1) ואז תגובת הרשת, הפלט, מתפשטת אחורה (מעבר 2) תוך כדי השוואת התגובה המצויה לתגובה הרצויה ועדכון המשקלים הסינפטיים.

לנהיגה אוטונומית יש הפוטנציאל להיחשב בתור תחום אידיאלי לאלגוריתם למידה מפוקחת כגון back-propagation כיוון שקיימת "תגובה נכונה" בצורה של כיוון ההיגוי הנכון של נהג אנושי. זה אפשרי תיאורטית (וגם מעשית) ללמד את הרשת לחקות נהג אנושי באמצעות תמונת חיישן בתור קלט וכיוון היגוי הנכון של הנהג בתור פלט בזמן אמיתי. טכניקה זו נקראת אימון "on-the-fly".

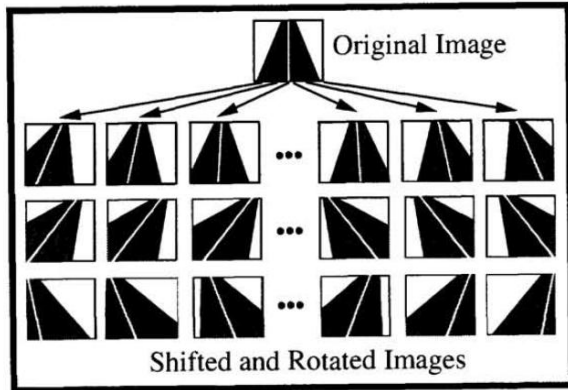
בעיות פוטנציאליות

ישנן 2 בעיות פוטנציאליות הקשורות באימון הרשת באמצעות תמונות חיישן בזמן אמת תוך כדי שנהג אנושי מנווט את הרכב. ראשית, כיוון שהאדם נוהג ברכב לאורך מרכז הדרך בזמן אימון, הרשת לעולם לא תקבל בתור קלט סיטואציות בהן היא צריכה להתאושש משגיאות מסוג סטייה מהנתיב (misalignment errors). כאשר הרשת בשליטה על הרכב, הרכב עשוי לעיתים לסטות מעט ממרכז הדרך ולכן המערכת צריכה לדעת להתאושש ע"י היגוי חזרה למרכז. הבעיה השנייה היא שאימון נאיבי של הרשת רק באמצעות תמונות הווידאו הנוכחיות וכיוון ההגה עלול לגרום ללימוד יתר (overlearn) של הקלטים האחרונים. אם נהג אנושי נוהג בNavlab בקטע דרך ישרה לקראת סוף שלב האימון, הרשת תקבל כדוגמאות רצף ארוך של תמונות דומות. חוסר הגיון הזה עלול לגרום לרשת "לשכוח" את מה שהיא למדה על נהיגה בדרכים עקלקלות, למשל.

שתי הבעיות הללו עם לימוד on-the-fly נובעות מהצורך של אלגוריתם back-propagation בסט אימון שמייצג את כל ההיבטים של המטלה שיש לבצע.

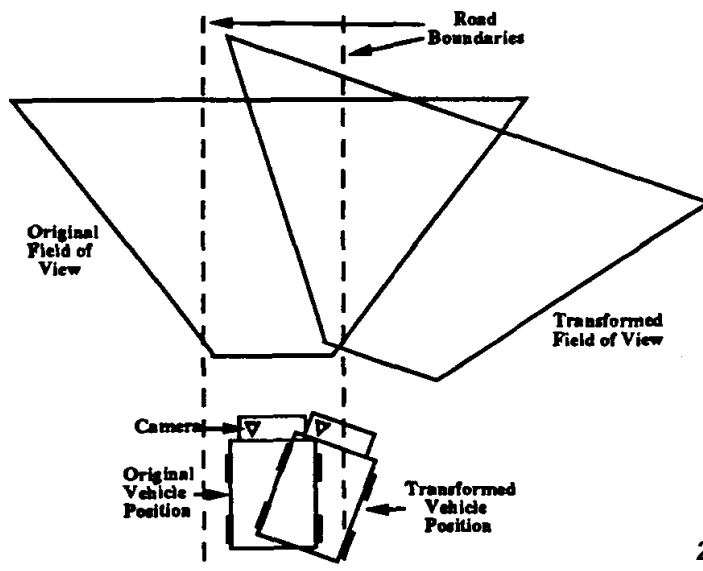
טרנספורמציות תמונה

כדי להשיג רמה מספקת של גיוון בתמונות חיישן אמיתיות, צוות הפרויקט ALVINN פיתח טכניקה של שינוי (transformation) התמונות המגיעות מהחיישנים כדי ליצור דוגמאות נוספות בסט האימון. במקום להציג לרשת רק את התמונה הנוכחית המגיעה מהחיישן (כקלט) ואת כיוון ההגה (כפלט), כל תמונה מוזזת ומסובבת (shift and rotate) ע"י תוכנה כדי ליצור תמונות נוספות בהן נראה כאילו הרכב נמצא במיקום שונה ביחס לסביבה (איור 27). המיקום והאוריינטציה ביחס לקרקע של החיישן ידועים, לכן ניתן ליצור טרנספורמציות מדויקות באמצעות גאומטריה תפיסתית (perceptive geometry).



איור 27

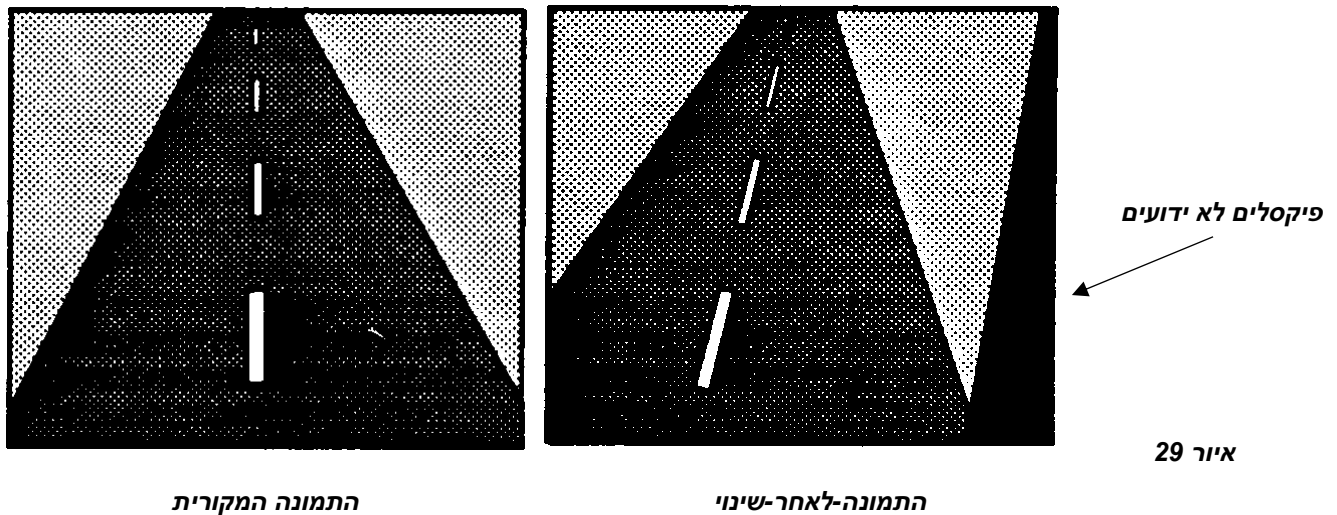
הטרנספורמציות נוצרות באמצעות קביעה של שטח הקרקע אשר נראה בתמונה המקורית, והשטח שאמור להיראות בתמונה-לאחר-שינוי. השטחים הללו יוצרים שני טרפזים חופפים כפי שניתן לראות באיור 28. כדי לקבוע את הערך המתאים לפיקסל בתמונה-לאחר-שינוי, הפיקסל המדובר מוטל (projected) על שטח הקרקע ואז מוטל חזרה על התמונה המקורית. משתמשים בערך של הפיקסל בתמונה המקורית בתור הערך של הפיקסל בתמונה-לאחר-שינוי. חשוב להבין שהמיפוי פיקסל-לפיקסל שמשמש לטרנספורמציה אחת הוא קבוע. במילים אחרות, אם מניחים עולם שטוח, הפיקסלים שיש לדגום בתמונה המקורית על מנת להשיג הזזה ספציפית של תמונה ותרגומם לתמונה-לאחר-שינוי תמיד יהיו אותם הפיקסלים. ALVINN מנצל עובדה זו ומחשב מראש את הפיקסלים שיש לדגום על מנת ליצור את כל הטרנספורמציות הדרושות. כתוצאה מכך, שינוי התמונה המקורית (כדי לדמות שינוי מיקום הרכב) דורש אך ורק שינוי של דפוס דגימת הפיקסלים בזמן השלב הקדם-עיבודי (preprocessing phase) של שינוי הרזולוציה של התמונה. לכן, יצירת הטרנספורמציות ברזולוציה נמוכה לא לוקחת יותר מזמן מהזמן שצריך (ממילא) כדי לשנות את הרזולוציה של התמונה המקורית. ההנחה הדרושה של עולם שטוח כמובן איננה נכונה, אך שינויי הגובה בכביש כתוצאה ממהמורות ומבורות הם קטנים מספיק כדי להיחשב זניחים.



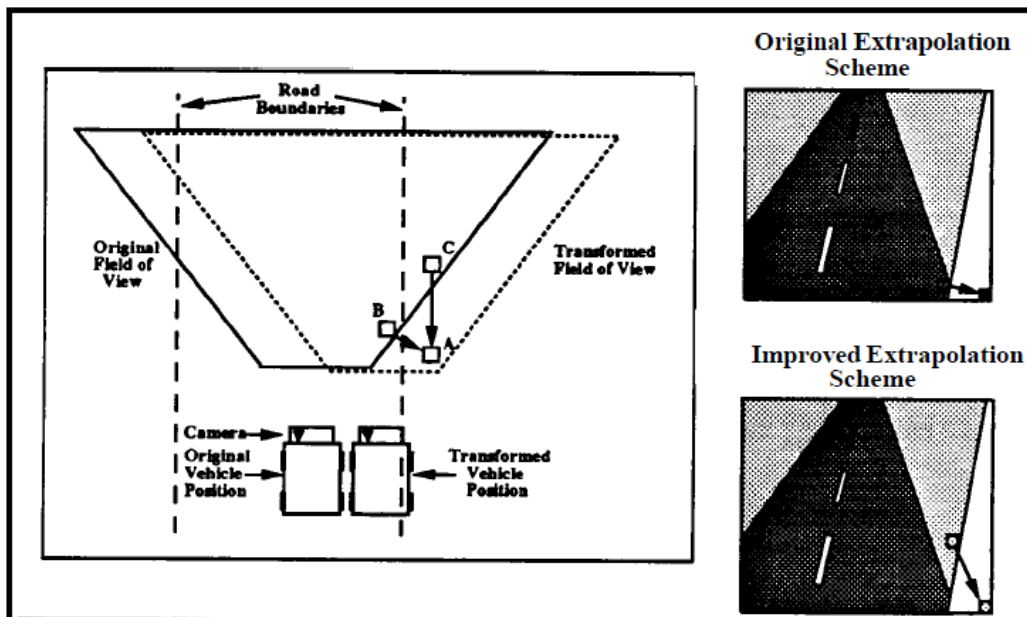
איור 28

הסקת הפיקסלים החסרים

כפי שניתן לראות באיור 28, חפיפת הטרפזים רחוקה משלמה. יש צורך בשלב נוסף ביצירת טרנספורמציות התמונות. בשלב זה יש לקבוע את ערכם של הפיקסלים בתמונה-לאחר-שינוי אשר להם אין פיקסלים מתאימים (לפי המיפוי פיקסל-לפיקסל שתואר מעלה) בתמונה המקורית. ראו לדוגמה את איור 29. כדי לגרום למיקום המדומה של הרכב להיות מטר אחד ימינה מהמיקום האמיתי שלו, לא רק שצריך להזיז את הפיקסלים בתמונה המקורית שמאלה, אלא גם צריך למלא את הפיקסלים הלא-ידועים בקצה הימני. שימו לב שכמות הפיקסלים לשורה שיש למלא גדולה יותר בתחתית התמונה מאשר בחלקה העליון. זאת מכיוון שמטר אחד של שטח קרקע מימין לגבול של התמונה המקורית מכסה יותר פיקסלים בתחתית מאשר בחלק העליון (כי העצמים בחלק העליון רחוקים יותר מהמצלמה מאשר עצמים בחלק התחתון). צוות ALVINN השתמש בשתי טכניקות כדי להסיק את הפיקסלים החסרים.



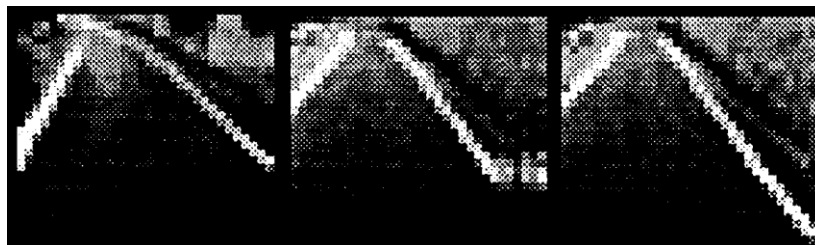
בטכניקה הראשונה, כדי לקבוע את הערך של הפיקסל שמוטל על משטח הקרקע בנקודה A (איור 30) בתמונה-לאחר-שינוי, יש למצוא את הנקודה הקרובה ביותר ל A בטרפז המקורי (נקודה B). נקודה זו מוטלת חזרה על התמונה המקורית כדי למצוא את הפיקסל הנכון לדגום. בתמונה בפינה הימנית העליונה של איור 30 מדגימה שיטה זו. הבעיה בטכניקה זו היא "מריחה" (smearing) של הפיקסלים המוסקים לאורך שורות התמונה (התמונה האמצעית באיור 31). התמונה הימנית ביותר באיור 31 היא התמונה המקורית של כביש דו-נתיבי כפי שצולמה במצלמת הווידאו. שימו לב לקווים המציינים את הגבולות של הנתיב הימני. התמונה האמצעית היא טרנספורמציה של התמונה המקורית שמדמה תזוזה של הרכב מטר אחד ימינה שבוצעה באמצעות הטכניקה שתוארה לעיל. הקו בצד ימין של הכביש נראה מרוח מימין בנקודת המפגש עם הגבול של התמונה המקורית. בגלל שאורך המריחה קשור לכיוון ההיגוי הנוכחי, הרשת לומדת על התלות בין גודל המריחה לתחזית כיוון ההיגוי הנכון. כאשר הרשת בשליטה על הרכב, אין מריחה של התמונה ולכן הרשת לא מתפקדת כראוי (כי איננה יודעת איך להתמודד עם מצב שבו הרכב באמת נמצא מטר אחד ימינה והתמונה שהיא צריכה לנתח איננה מרוחה).



איור 30

כדי להעלים את התוצאה הלא רצויה הזו, צוות ALVINN השתמש בטכניקת הסקה המסתמכת על ההנחה הזו: מאפיינים מעניינים (כגון קצוות הכביש וסימוני נתיבים) בדרך כלל מקבילים לכביש, ולכן גם מקבילים לכיוון הנסיעה הנוכחי של הרכב. כאשר מניחים הנחה זו, כדי להסיק את הערך של הפיקסל בנקודה A, הנקודה המתאימה ביותר לדגימה בטרפז המקורי איננה הנקודה הקרובה ביותר (נקודה B) אלא הנקודה הקרובה ביותר בטרפז המקורי לאורך הקו היוצא מ A ומקביל לכיוון הנסיעה המקורי של הרכב (נקודה C).

תיאור סכמתי של שיטה זו נמצא בפינה הימנית התחתונה של איור 30. התמונה הימנית ביותר באיור 31 נוצרה באמצעות שיטה זו (כמו מקודם, המיקום המדומה של הרכב הוא מטר אחד ימינה ביחס למיקום המקורי). המריחה אשר הייתה קיימת בשיטה הקודמת כבר איננה נראית לעין.



איור 31

אמנם חשוב לבצע טרנספורמציות תמונה כדי ליצור גיוון בסט האימון, אך יש גם לבצע טרנספורמציות על כיוון ההגה כדי שיתאים לתמונות-לאחר-השינוי. כיוון ההיגוי הנכון, כפי שנקבע ע"י נהג ע"פ התמונה המקורית חייב להשתנות עבור כל אחד ממיקומי הרכב המדומים. ניתן לעשות זו באמצעות שיטה הנקראת pure pursuit steering [8]. ע"פ שיטה זו כיוון ההיגוי ה"נכון" הוא זה שיביא את הרכב למיקום כלשהו (בדרך כלל מרכז הדרך) הנמצא במרחק קבוע בהמשך הדרך.

באיור 32 מוצג הרעיון מאחורי pure pursuit steering. אם הרכב נמצא במיקום A, נהיגה לאורך מרחק ידוע מראש לאורך קשת ההיגוי הנוכחית תביא את הרכב למיקום מטרה T, אשר נקבע להיות מרכז הדרך.

אחרי ביצוע טרנספורמציה של הזזה אופקית s וסיבוב θ כדי לדמות שהרכב נמצא בנקודה B, כיוון ההיגוי הנכון ע"פ מודל pure pursuit steering גם יביא את הרכב לנקודה T. להלן הנוסחה לחישוב רדיוס ההיגוי שיביא את הרכב מנקודה B לנקודה T:

$$(3.1) r = \frac{l^2 + d^2}{2d}$$

כאשר r הוא רדיוס ההיגוי, l הוא המרחק האנכי מהנקודה T (lookahead distance) d הוא המרחק בין נקודה T למיקום בו הרכב יימצא אם ימשיך ישר מנקודה B לאורך מרחק l . ניתן לחשב את d ע"י:

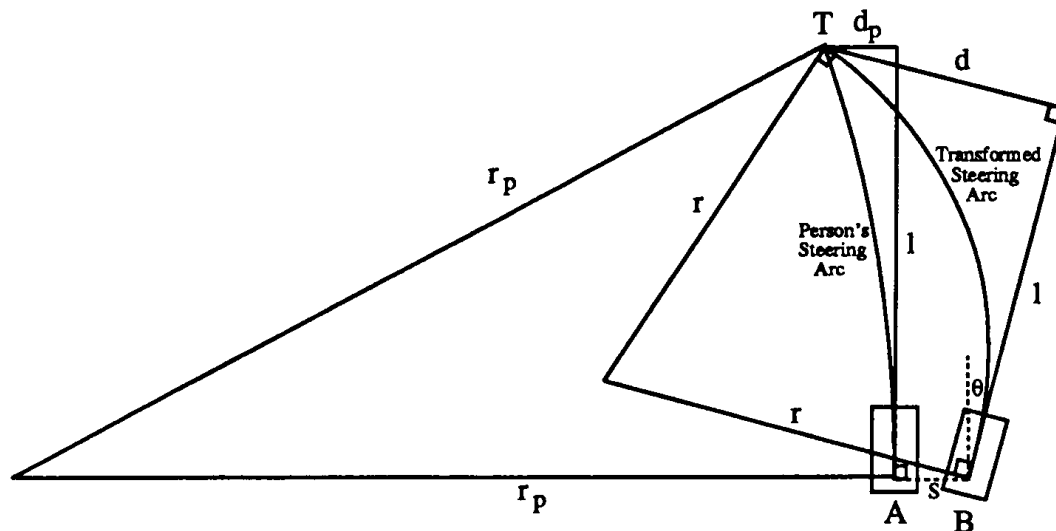
$$(3.2) d = (d_p + s + l \tan \theta) \cos \theta$$

כאשר d_p הוא המרחק בין נקודה T למיקום בו הרכב יימצא אם ימשיך ישר מנקודה A לאורך מרחק l . s הוא מרחק האופקי בין A ל B ו θ היא זווית סיבוב הרכב מ A ל B. ניתן לחשב את d_p באמצעות:

$$(3.2) d_p = r_p - \sqrt{r_p^2 - l^2}$$

כאשר r_p הוא הרדיוס של קשת ההיגוי בזמן שצולמה התמונה.

הפרמטר האחרון שלא נקבע עדיין הוא l , המרחק האנכי אל נקודת היעד T. צוות ALVINN בדק ומצא, באופן אמפירי, שבטווח מהירות שנע בין 5 ל 55 מיילים לשעה, על מנת להשיג שליטה יציבה ומדויקת ברכב ניתן לקבוע את l כך: המרחק אותו הרכב יעבור במהירות הנוכחית תוך 2-3 שניות. כלל אמפירי זה מפיק תוצאות טובות המדמות את הדרך בה נהג אנושי היה מבצע פנייה.



איור 32

כמו בשיטת טרנספורמצית התמונות, טרנספורמצית כיוון ההיגוי משתמשת במודל פשוט כדי לקבוע איך שינוי במיקום ובאוריינטציית הרכב ישפיע על הסיטואציה. במקרה של טרנספורמצית תמונות, הנחת עולם שטוח וכללי ההטלה עזרו לקבוע כיצד שינוי במיקום ובאוריינטציית הרכב ישפיעו על תמונת החיישן של המתרחש בשדה הראייה של הנהג. במקרה של טרנספורמצית כיוון ההיגוי, מודל של איך אנשים נוהגים עזר לקבוע כיצד טרנספורמצית רכב ספציפית תשנה את כיוון ההיגוי הנכון. בשני המקרים, טכניקות הטרנספורמציה אינן תלויות בסיטואציית הנהיגה. אדם יכול לנהוג בדרך עפר בעלת נתיב אחד או בכביש מהיר בעל מספר נתיבים: טכניקות הטרנספורמציה לא ישתנו.

אנתרופומורפית, שינוי תמונות החיישן כדי ליצור יותר דוגמאות בסט האימון שקול ללומר לרשת "אני לא יודע אילו מאפיינים בתמונה חשובים כדי לקבוע את כיוון ההיגוי הנכון, אך יהיו אשר יהיו, הנה עוד מיקומים ואוריינטציות שבהם מאפיינים אלו אולי קיימים". באופן דומה, שינוי כיוון ההיגוי עבור כל אחת מהתמונות החדשות שקול ללומר לרשת "לא משנה מהם המאפיינים החשובים, אם הם נמצאים במיקומים ובאוריינטציות החדשים, ככה עליך להגיב". בגלל שהמערכת לא מסתמכת על מודל חזק של איך מאפיינים חשובים אמורים להיראות, אלא רוכשת את הידע הזה באמצעות אימון, המערכת יכולה לפעול במגוון רחב של סביבות.

2 סוגי הטרנספורמציות שתוארו לעיל מספקים פתרון ל-2 הבעיות הפוטנציאליות: הרשת לומדת איך להתאושש משגיאות נהיגה שלא הייתה יכולה להתמודד איתם בעזרת אימון נאיבי (כמו סטייה קלה מהנתיב). כמו כן, אימון-יתר עם תמונות דומות הוא כבר לא בעיה, שכן התמונות שעברו טרנספורמציה דואגות לגיוון בסט האימון.

פרטי האימון

הפרטים האחרונים הדרושים כדי לאפיין את תהליך הלמידה on-the-fly הם כמות וסדר הגודל של הטרנספורמציות בשימוש לאימון הרשת. הכמויות הבאות נקבעו בצורה אמפירית כדי לשמור על רמת גיוון מספיק גבוהה כדי לאפשר לרשת ללמוד לנהוג במגוון סיטואציות. התמונה מקורית מוזזת ומסובבת

14 פעמים באמצעות הטכניקות שתוארו מעלה כדי ליצור 14 דוגמאות חדשות על כל דוגמה אמיתית. גודל ההזזה עבור כל טרנספורמציה נקבע באופן אקראי מהטווח -0.6 עד $+0.6$ מטר. זווית הסיבוב נבחרה באופן אקראי מהטווח -6.0 עד $+6.0$ מעלות. שדה הראייה האופקי של המצלמה של Navlab הוא בן 42 מעלות. תמונה עם הזזה מקסימלית של 0.6 מטר יוצרת תמונה בה הדרך זזה בערך שלישי מהאורך של התמונה המקורית בתחתית.

לפני שמבצעים את ההזזה והסיבוב של התמונה המקורית, מחשבים את כיוון ההיגוי החדש (כאילו השינוי כבר התבצע). אם כיוון ההיגוי הוא חד יותר מהפנייה החדה ביותר שפלט הרשת מסוגל להביע (פנייה בעלת רדיוס של בערך 20 מטר), אז מדלגים על הטרנספורמציה הזו ובוחרים באקראי טרנספורמציה אחרת. ע"י מניעה של תנאי נהיגה קיצוניים ולא-סבירים (למשל כאלו בהם הרכב הוזז הרבה מאוד ימינה ונקודת היעד היא בשמאל הקיצוני), הרשת יכולה להקדיש הרבה יותר מיכולות ייצוג הסביבה שלה למצבים סבירים.

15 הדוגמאות החדשות, 14 טרנספורמציות יחד עם הדוגמה המקורית, נכנסות לתוך חוצץ בעל 200 תאים אשר "מעורבב" בכל הכנסת דוגמה (כדי ליצור גיוון בסדר האימון) – 15 דוגמאות יוצאות. כעת מזינים את כל 200 הדוגמאות בחוצץ לרשת בתור סט אימון כאשר לאחר כל דוגמה מתבצעת הרצה של אלגוריתם back-propagation, בעל פרמטר קצב למידה שערכו 0.01 . חוזרים על התהליך עבור כל תמונה חדשה. כל הרצה כזו לוקחת בערך 2.5 שניות באמצעות 3 מעבדים על גבי הרכב. הראשון אחראי לקבלת התמונות מהחיישנים וביצוע הטרנספורמציות, השני מממש את סימולציית רשת הנוירונית והשלישי מתקשר עם בקרי הרכב כדי לגרום להם לפעול ע"פ פלט הרשת ולהציג מידע לצופה אנושי. הרשת צריכה בערך 100 איטרציות שלמות (100 תמונות אמיתיות ו- 1400 טרנספורמציות) כדי לדעת לנהוג בשטח בו אומנה. כיוון שלכל איטרציה לוקח 2.5 שניות להסתיים, כל האימון לוקח קצת יותר מ- 4 דקות של נהיגה אנושית ברכב. בזמן האימון, הנהג האנושי נוהג בערך במהירות בה הרשת הולכת להבחן, בטווח בין 5 ל- 55 מיילים לשעה.

- [1] Le Cun Y., Boser B., Denker J. S., Henderson D., Howard R.E., Hubbard W., and Jackel L.D. (1990). Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in neural information processing systems*, volume II, pages 396-404. Morgan Kaufmann Publishers Inc. San Francisco, CA.
- [2] Pomerleau, D.A. (1990) Rapidly adapting artificial neural networks for autonomous navigation. In *Advances in Neural Information Processing Systems 3*
- [3] Hadsell R., Sermanet P., Scoffier M., Erkan A., Kavackuoglu K., Muller U., and LeCun Y. (2009) Learning Long-Range Vision for Autonomous Off-Road Driving. *Journal of Field Robotics*, 26(2) pages 120-144.
- [4] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012) ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*
- [5] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) pages 2278–2324.
- [6] Le Cun Y., Boser B., Denker J. S., Henderson D., Howard R.E., Hubbard W., and Jackel L.D. (1990). Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in neural information processing systems*, volume II, pages 396-404. Morgan Kaufmann Publishers Inc. San Francisco, CA.
- [7] Haykin S. (1998) *Neural Networks: A Comprehensive Foundation*. Prentice Hall New Jersey, NY.
- [8] Pomerleau, D.A. (1993) Knowledge-based training of artificial neural networks for autonomous robot driving. In *J.H. Connell & S. Mahadevan (Eds.), Robot learning* pages 19–43. Kluwer Academic New York, NY.
- [9] Russell, S.J., and Norvig P. (1995) *Artificial Intelligence: A Modern Approach*. Prentice Hall New Jersey, NY.
- [10] Ng, A. (2011) Machine Learning. <https://www.coursera.org/course/ml>.
- [11] Wikipedia, The free encyclopedia. http://en.wikipedia.org/wiki/Main_Page.