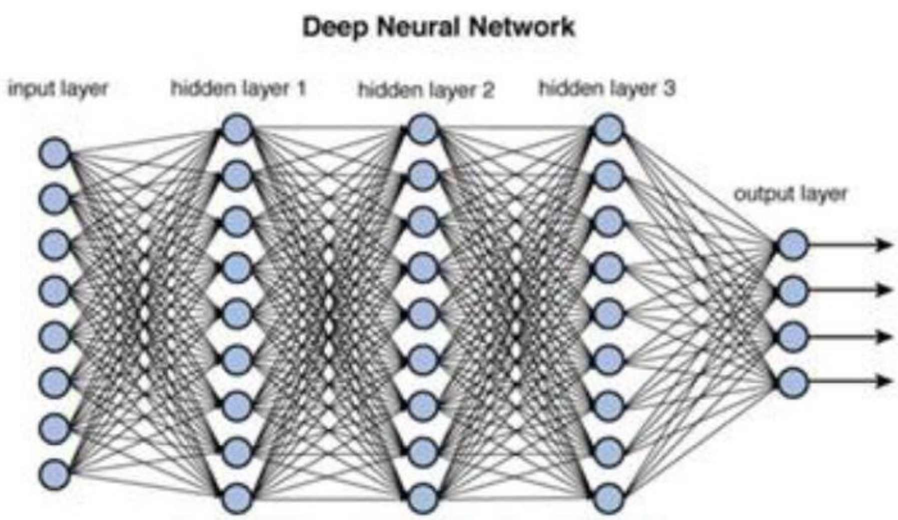
<u>רשת נוירונים עמוקה</u>

רשת נוירונים הינה מערכת למידה מפוקחת הבנויה ממספר גדול של אלמנטים פשוטים, המכונים נוירונים כאשר כל נירון מסוגל לבצע בחירה של החלטה פשוטה ואת ההחלטה הזו הוא מעביר קדימה לנוירונים הבאים, כאשר הנוירונים מסודרים בצורה של שכבות אשר מחוברות זו לזו .

באמצעות הרשת, יהיה ניתן להביע כמעט כל פונקציה, ולכן ניתן לענות על משימות כמו מיון, סיווג וחישובים מסוגים שונים.

על מנת למטב את ביצועי הרשת נדאג לספק סט אימון רחב הכולל דוגמאות שונות של אותה בעיה אשר אנו מנסים לפתור.

המחשה של רשת נוירונים עמוקה המורכבת משכבות של נוירונים, כניסה ויציאה ושכבות נסתרות ונוירונים:



<u>השכבה השמאלית (input layer)</u> - מתארת את שכבת הקלט.

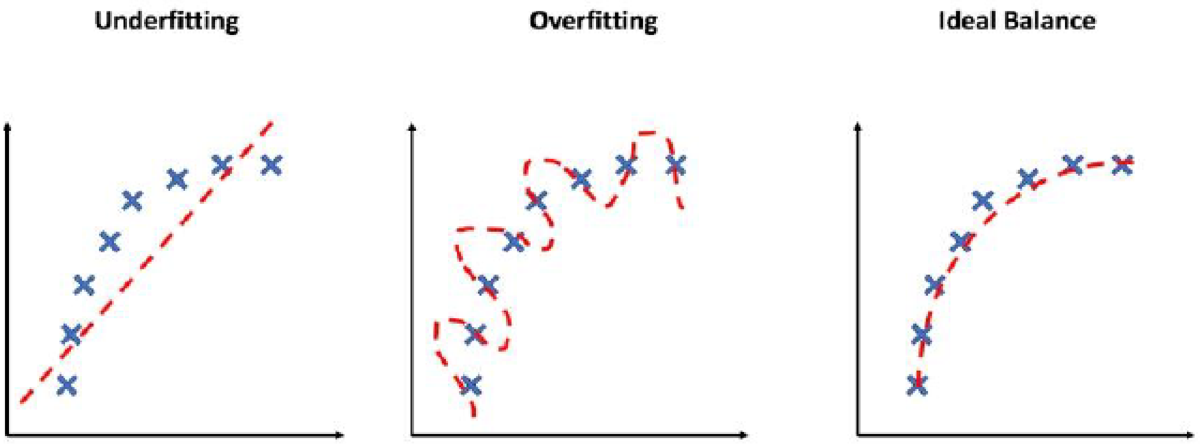
<u>השכבות האמצעיות(hidden layers)</u> - אלו הן שכבות הביניים, כאשר מספר שכבות אלו איננו מוגבל.

בשכבות אלו מתבצע עיבוד ביניים לטובת חישוב השערוך, כל שכבה מכילה משקולות אשר מחושבות באמצעות גרדיאנטים, הנוירונים עוברים דרך פונקציית אקטיבציה לפני המעבר לשכבה הבאה.

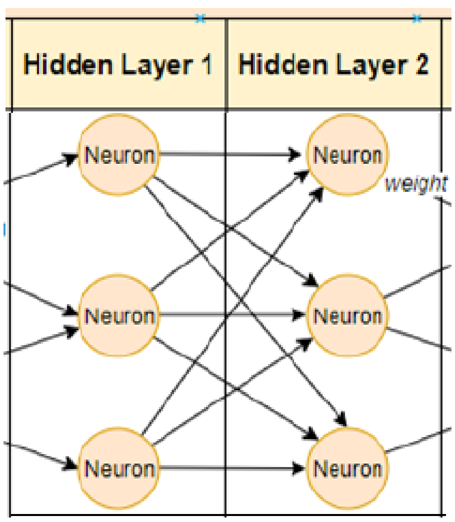
<u>השכבה הימנית (output layer)</u> - שכבת המוצא, אשר מקבלת את המידע שהועבר אליה דרך השכבות הנסתרות ומציגה חיזוי סופי.

<u>שיטות תכנון רשת נוירונים</u>

הבעיה המרכזית ביותר בתכנון מערכות אלו היא בחירה של ההיפר פרמטרים, אלו הפרמטרים שאנו קובעים אשר עבורם הרשת תיתן את הביצועים הטובים ביותר .
על מנת לבחור את ההיפר-פרמטרים, משתמשים בשיטה של קרוס ולידציה, כאשר מחלקים את הסט של האימון לסט של אימון וגם סט של קרוס-ולידציה (בדרך כלל נבחר את האחוז הדומיננטי של הדאטה סט לטובת האימון), את סט האימון נאמן דרך חישוב הגרדיאנטים ואת סט הקרוס ולידציה נעביר דרך הרשת ללא חישוב גרדיאנטים אך נשתמש בחישוב של השגיאה וכך נדע האם המערכת מצליחה לשערך מידע שהוא זר מהסט אימון, במילים אחרות אנחנו פותרים כך את בעיית ה overfitting שעלולה להגרם לרשת, במצב של overfitting אנחנו נראה ביצועים טובים של הרשת כלפי סט האימון אך לא כלפי מידע זר.



המחשה של overfitting, במצב זה ניתן לראות כי הקו עובר דרך הנקודות של סט האימון אך בין הנקודות קיימת אוסילציה.

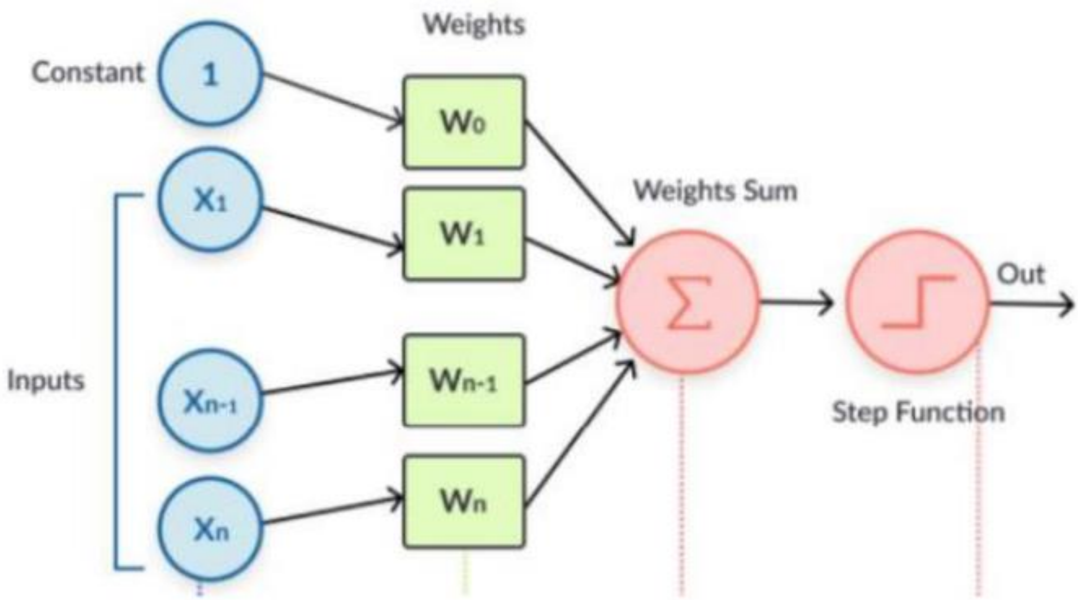


המחשה של מבנה השכבות הנסתרות והחיבורים ביניהן

<u>אלגוריתם העברת החישובים ברשת(Forward Passing):</u>

נתאר את התהליך שהמידע עובר בתור הרשת ודרך הנוירונים .

1. הכניסות המוזנות לרשת מוכפלות במשקולות.
2. סכימה של הכניסות המוכפלות במשקולות.
3. הוספה של קבוע bias על מנת לבצע הזזה לכיוון השערוך.
4. העברה של הסכום דרך פונקציית אקטיבציה .
5. העברת התוצאה הכוללת ככניסה של הנוירון הבא ברשת.



המחשה של אופן חישוב המידע ברשת נוירונים

שכבות הביניים מורכבות משרשור של שכבות נוירונים וביניהן פונקציות שיכולות להיות לא ליניאריות בשם פונקציות אקטיבציה.

הנוירון בכל שכבה, הינו צירוף ליניארי של כל הערכים של הנוירונים בשכבה הקודמת ועליו מופעלת פונקציית האקטיבציה.

מטרת פונקציות האקטיבציה הינה מיפוי הסכום המשוקלל שאותו מקבל נוירון לערך בין 0 ל1 כמובן באפליקציות מורכבות יותר ניתן להישתמש בפונקציות אקטיבציה לא לינאריות וכך תחום הפונקציות אותן הרשת יכולה להביע גדל.

פונקציות אקטיבציה נפוצות:

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

פונקציית סיגמונד:

פונקציה זו מקבלת כך ערך ומנרמלת אותו לטווח מספרים בין 0 ל 1.

פונקציה זו איננה נמצאת בדרך כלל בשכבות הביניים, השימוש הנפוץ של פונקציה זו הוא בבעיות מסוג סיווג, לדוגמא, במערכת שתפקידה להחליט האם קלט של תמונה כלשהי מכיל חתול, סיגמוייד יציג לנו שערוך סופי בטווח של 0 עד 1 וכך נדע את הסיכוי של המצאות החתול בתוך התמונה (למשל, עבור ערך יותר קרוב ל- 1, כנראה שמודבר בחתול).

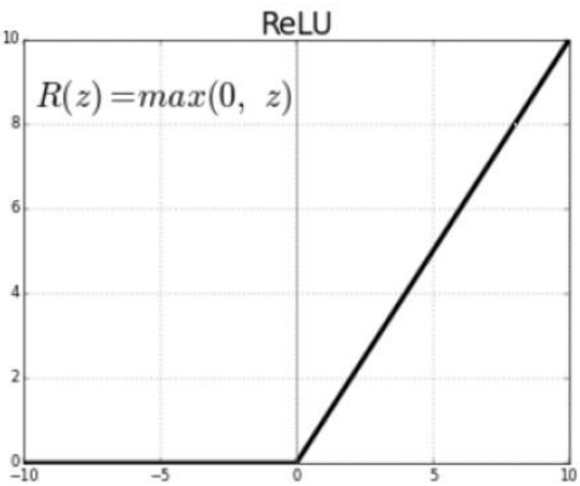
פונקציית ReLU

$$f(x) = x^{+} = \max(0, x)$$

הפונקציה מאפסת כל ערך שלילי שהיא מקבלת ומעבירה הלאה כל ערך חיובי שהיא מקבלת.

לפונקציה זו שימוש נפוץ בעולם של רשתות נוירונים מאחר והיא זולה לשימוש כתוצאה מכך שישנם טווח ערכים שהיא מאפסת, ולכן חלק מהנוירונים לא יעבדו בו זמנית וכך נחסוך זמני חישוב .

מ2017 זוהי הפונקציה שזוכה לשימוש הנפוץ ביותר בקרב אלגוריתמים של ראייה ממוחשבת ועוד.



סרטוט של פונקציית הReLU.

כיוון שנעסוק ברשתות בינאריות פונקציית הReLU תיראה למשל כך:

$$\sigma(x) = max(0, min(1, (x+1)/2))$$

<u>אלגוריתם למידה והעברה של מידע אחורה(Back Propogation):</u>

מאחר ואנחנו מאתחלים את רשת הנוירונים עם משקולות התחלתיות שניתן להגיד שהן לא אופטימליות, החיזוי הראשוני שמתקבל בתהליך ה Forward Passing(שכלול של הכניסות עם המשקולות) הוא לא החיזוי הטוב ביותר .

שימוש נפוץ למזעור השגיאה ברשת נוירונים הוא בעזרת Backpropagation .

המושג backpropagation כולל בתוכו משפחה של אלגוריתמים, שהרעיון המרכזי שלהם הוא למזער את השגיאה ולהביא את המשקולות של שכבות הביניים למצב אופטימלי בצורה מהירה יחסית, אפילו עבור רשת עם כמות גדולה של נוירונים.

<u>רשת נוירונים בינארית:</u>

רשת נוירונים בינארית היא רשת נוירונים זהה במבנה למה שהוצג עד כה פרט לכך שהמשקלים שמשויכים לכלל הנוירונים הינם 1 או -1 שנשמר בזכרון כ- '1' או '0' בהתאמה, דבר אשר מאיץ משמעותית את כלל החישובים במערכת על מנת להגיע להחלטה.

<u>מימוש בחומרה:</u>

בסופו של דבר אנו מנסים לחשב את החישוב הבא:

$$a_i^l = \sigma\left(a_1^{l-1}w_1 + a_2^{l-2}w_2 + \cdots a_{n-1}^{l-n}w_{n-1} + b\right)$$

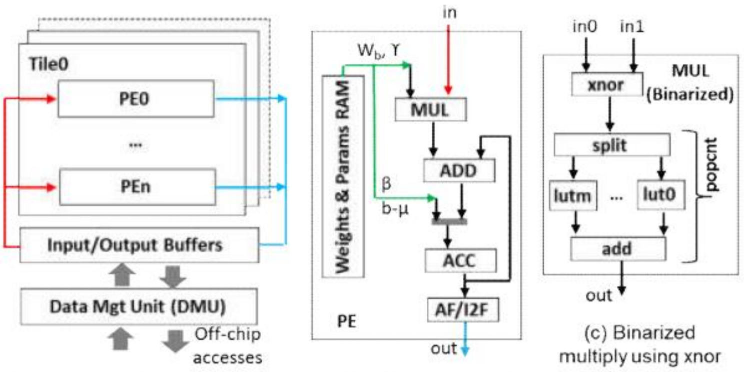
כאשר w_i הינו המשקולת, a_i הינו הנוירון במיקום ה-i, b הוא ערך מספרי משתנה בהתאם לנוירון וקובע את רמת הפעילות.

לכן בסך הכול זה הינו כפל המטריצות הבא:

$$\begin{bmatrix} w_{0,0} & \cdots & w_0, n \\ \vdots & \ddots & \vdots \\ w_{k,0} & \cdots & w_{k,n} \end{bmatrix} \cdot \begin{bmatrix} a_0^0 \\ \vdots \\ a_n^0 \end{bmatrix} + \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix}$$

לכן מבחינת חומרה נצטרך full adder ומכפל.

לכן דוגמא למימוש חומרתי יהיה:



(a) BNN Hardware Accelerator (b) Processing Element (c) Binarized multiply using xnor and popcount

לו"ז לפרוייקט:

ינואר-מרץ

השמת מערכת מבוססת רשת ניורונים בינארית לזיהוי מספרים

ניתוח וייעול פונקציית סיגמא

ריכוז תוצאות ומסקנות