# Subject: An advanced scalable hardware accelerator for mini batch gradient descent, targets deep-learning applications.

**תוכן העניינים**

1. מבוא ....................................................................................................2
2. רקע תיאורטי ......................................................................................3-7
   1. למידת מכונות - Machine Learning
      1. למידה מבוקרת- Supervised learning
      2. למידה בלתי מבוקרת- Unsupervised learning
   2. למידה עמוקה – Deep Learning
   3. Gradient Descent
      1. Mini Batch Gradient Descent
   4. פרוטוקול - AMBA APB
   5. המאמר

**פרק 1: מבוא**

הלמידה הינה תופעת רבת פנים. תהליך הלמידה כולל רכישת ידע חדש, פיתוח מיומנויות מוטוריות וקוגניטיביות באמצעות הדרכה או תרגול, ארגון של ידע חדש לכלל, הצגה אפקטיבית, וגילוי של עובדות חדשות ותאוריות מתוך תצפית וניסויים.

מאז תחילתו של עידן המחשב, החוקרים כבר חתרו להשתיל יכולות כאלה במחשבים. פתרון בעיה זאת הייתה ועדיין נשארה מטרה לטווח הרחוק המאתגרת ביותר והמרתקת בבינה מלאכותית. המחקר ומודל המחשב של תהליכי הלמידה מהווה את הנושא של למידה ממחושבת.

רשתות עצביות עמוקות (רשתות נוירונים) הוא מודל מתמטי שפותח בהשראת תהליכים מוחיים או קוגניטיביים המתרחשים ברשת עצבית טבעית ומשמש במסגרת למידת מכונה.

רשתות עצביות עמוקות, נמצאות בשימוש נרחב במספר רב של יישומים לניתוח וחילוץ מידע שימושי מתוך כמות גדולה של נתונים אשר שמופקת בכל יום. היכרות והכשרה הם שני מצבי הפעולה העיקריים של רשת עצבית. תהליך אימון (שעליו נפרט בהמשך) הוא משימה מאוד מאתגרת מבחינה חישובית, שכן הוא כרוך בפתרון בעיית אופטימיזציה בקנה מידה גדול, בדרך כלל באמצעות אלגוריתם התפשטות לאחור, backprop, אשר מבוסס על שיטת gradient descent. שיטה זו, היא שיטת אופטימיזציה איטרטיבית למציאת מינימום מקומי של פונקציה.

במאמר זה מוצעת סכמה לניצול הזיכרון המוטבע ולהתגברות על "צוואר הבקבוק" של מערכת הזיכרון המגבילה את יכולת הגמישות המקבילית של SGD. הדבר אפשרי ע"י יצירת מערכת זיכרון המותאמת אישית אל מערך הנתונים הספציפי ואל תבניות הגישה המיוחדות לזיכרון בשלבים השונים של האלגוריתם gradient descent. בזכות זה ניתן לצמצם לחלוטין את הצורך בתקשורת פנימית במעבד באלגוריתם descent mini-batch gradient ובכך לשפר את גמישות היישום.

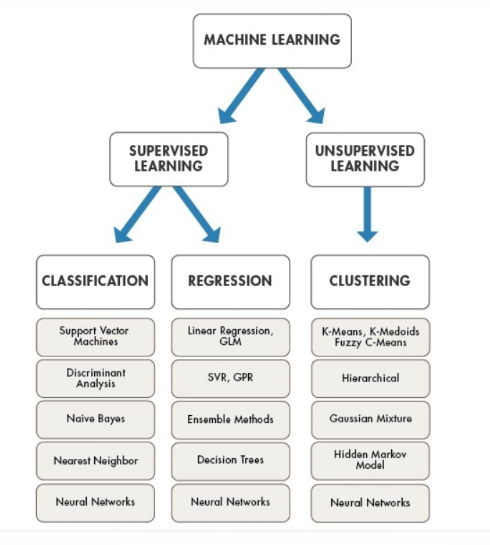
פרויקט זה השואב השראה רבה מן המאמר, מציע תכנון ובניית IP האצה –accelerator, אשר מבצע ביעילות חישובי RAMל-RAM בצורת צינור, בכך מאפשר הורדת העומס של יישומי תוכנה ללמידת מכונות.

**פרק 2: רקע תיאורטי**

2.1 למידת מכונה - Machine Learning

Machine Learning היא ענף של בינה מלאכותית (Al) העוסק בבניית מערכות שיכולות ללמוד ולקבל החלטות על סמך נתונים. במקום לתכנת במפורש את המכונה לפתור בעיה ספציפית, ניתן להראות לה כיצד בעיה דומה נפתרה בערב ובכך לתת למכונה את המידע הנדרש כדי שתוכל לתכנן את הצעדים העיקריים לפתרון אותה משימה באופן עצמאי. באופן זה המכונה "לומדת" מדוגמאות שמספקים לה.

אלגוריתמים של למידה ממוחשבת מתחלקים לשתי קטגוריות- למידה מבוקרת ולמידה בלתי מבוקרת (איור 2.1).



**איור 2.1- חלוקת נושא ה- Machine Learning לקטגוריות**

כל אחת מהקטגוריות מורכבת ממספר שיטות של ייצוג המידע וסיווגו לקבוצות מדגם המשמשות לפתרון בעיות. ניתן לראות כי מערכות-נוירונים (בתחתית האיור) חולשות על שתי הקטגוריות, ונותנות מענה למרבית הבעיות הנפוצות בעולם הלמידה הממוחשבת.

2.1.1 למידה מבוקרת – Supervised Learning

באופן מעשי, רוב הלמידה הממוחשבת נעשית ע"י למידה מבוקרת. למידה מבוקרת מתבססת על נתוני קלט ונתוני פלט מתאימים, ונעשה שימוש באלגוריתם כדי ללמד את פונקציה המיפוי מהפלט לקלט. המטרה היא להעריך בצורה מדויקת את פונקצית המיפוי כך שתוכל לחזור את נתוני הפלט באופן עצמאי.

בעיות בלמידה מבוקרת ניתן לחלק לשתי קבוצות- סיווג (כאשר מדובר בקטגוריה מסוימת) ורגרסיה (עבור משתנים רציפים).

2.1.2 למידה לא מבוקרת – Unsupervised Learning

למידה לא מבוקרת מתבססת על מידע שאינו מסווג או מתויג ובכך מאפשר לאלגוריתם לפעול על מידע זה ללא בקרה ו"הדרכה". בלמידה מסוג זה, המערכת עשויה לקבץ מידע עפ"י קווי דמיון והבדלים, למרות שלא מגדירים לה את כללי ה"סיווג" כמו במקרה של למידה מבוקרת.

אלגוריתמי למידה לא מבוקרת יכולים לבצע משימות עיבוד מורכבות יותר ממערכות למידה מבוקרות. עם זאת, בלמידה לא מבוקרת לא תמיד ניתן לצפות את התוצאות מראש ועל כן ניתן לקבל תוצאות בלתי צפויות.

שתי שיטות עיקריות בלמידה לא מבוקרת הן:

* קיבוץ - בעיות בהן המטרה למצוא קבוצות מובנות בנתונים.
* אסוציאציה – בעיות בהן המטרה למצוא מכנה משותף בין קבוצת הקלט לקבוצת הפלט ( כלומר תכונות המתארות כמות גדולה של נתונים).

2.2 למידה עמוקה – Deep Learning

Deep Learning היא היבט של בינה מלאכותית ( Al) העוסקת ב"חיקוי" צורת הלמידה בה נוהגים להשתמש בני אדם על מנת לרכוש ידע. בעוד האלגוריתמים המסורתיים של [למידת מכונות](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&ie=UTF8&prev=_t&rurl=translate.google.co.il&sl=en&sp=nmt4&tl=iw&u=https://searchenterpriseai.techtarget.com/definition/machine-learning-ML&xid=17259,15700019,15700124,15700149,15700186,15700190,15700201&usg=ALkJrhiLJHumI-ceKpX2fpRvT5VULRM8UA) הם לינאריים, [אלגוריתמי](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&ie=UTF8&prev=_t&rurl=translate.google.co.il&sl=en&sp=nmt4&tl=iw&u=https://whatis.techtarget.com/definition/algorithm&xid=17259,15700019,15700124,15700149,15700186,15700190,15700201&usg=ALkJrhhAZc6ZU8L6mRXsK5VFRRct_tiUOg) למידה עמוקה נערמים [בהיררכיה](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&ie=UTF8&prev=_t&rurl=translate.google.co.il&sl=en&sp=nmt4&tl=iw&u=https://whatis.techtarget.com/definition/hierarchy&xid=17259,15700019,15700124,15700149,15700186,15700190,15700201&usg=ALkJrhjWuWOSWQXxZnu4cRULaaoOPygtTA) של מורכבות והפשטה הולכים וגדלים.

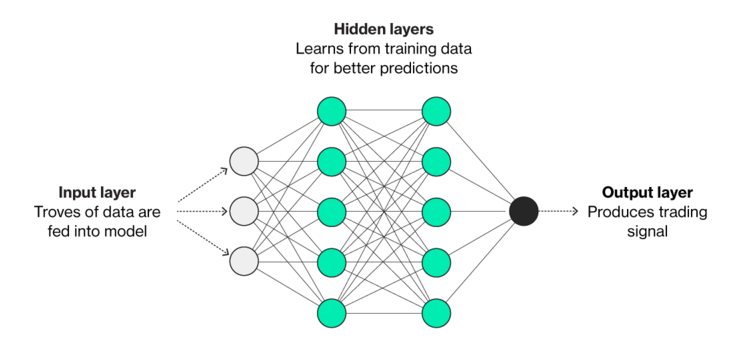
כל אלגוריתם בהיררכיה משתמש בטרנספורמציה לא ליניארית על הקלט שלו ומשתמש במה שהוא "לומד" כדי ליצור מודל סטטיסטי כפלט. [האיטרציות](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&ie=UTF8&prev=_t&rurl=translate.google.co.il&sl=en&sp=nmt4&tl=iw&u=https://searchsoftwarequality.techtarget.com/definition/iterative&xid=17259,15700019,15700124,15700149,15700186,15700190,15700201&usg=ALkJrhhBChCRVvQ712jfSRVUpGa9AOlXrg) ממשיכות (בשכבות) עד שהפלט הגיע לרמת דיוק קבילה.

מכיוון שתהליך זה מחקה מערכת אנושית של נוירונים, למידה עמוקה מכונה גם רשת נוירונים ( או רשת עצבית). אופן הבנייה והקישוריות בין הנוירונים המלאכותיים תלוי בצורה חזקה באלגוריתם הלמידה בו משתמשים כדי לאמן את הרשת.

קיימות מספר ארכיטקטורות בסיסיות המייצגות רשת נוירונים:

1. רשתות חד-שכבתיות ללא משוב.
2. רשתות רב-שכבתיות ללא משוב.
3. רשתות חוזרות בעלות לולאת משוב.

אחת החוזקות של הלמידה העמוקה היא השימוש החוזר באותן ארכיטקטורות לבעיות שונות (בדומה למוח האנושי). במילים אחרות, אותה רשת יכולה לעבור אימון לבעיות שונות.

נציג דוגמא לרשת רב שכבתית:

**איור 2.2- תיאור מודל השכבות של רשת נוירונים עבור ארכיטקטורה רב שכבתית**

בכל שכבת חישוב, לכל עיגול (נוירון - יחידת עיבוד המידע הבסיסית הנחוצה לתפקוד רשת הנוירונים) יש משקל והרשת עושה חישוב מסוים בהתבסס על המשקלים הנתונים. המידע מועבר משכבה לשכבה בשכבות המוסתרות (Hidden layer) כמתואר באיור 2.2.

המשקלים נקבעים עפ"י הרשת ומוצבים בחישוב של פונקציית מחיר (cost function). המטרה של האלגוריתם היא לקבוע את המשקלים כך שיקטינו ככל האפשר את ה- cost function. הקטנת פונקציית ה"מחיר" נעשית ע"י שיטת ה- Gradient Descent שעליה נפרט בסעיף הבא.

2.3 Gradient Descent

Gradient Descent הוא אלגוריתם אופטימיזציה המשמש לעיתים קרובות למציאת המשקלים או המקדמים של אלגוריתמי למידה כגון רשתות עצביות.

המודל עושה תחזיות על סמך נתוני "האימון" ומשתמש בפונקצית שגיאה התלויה בתחזיות על מנת לעדכן את המודל בצורה כזו שיקטין את השגיאה.

מטרת האלגוריתם היא למצוא את פרמטרי המודל (למשל מקדמים או משקולות), אשר מקטינים את שגיאת המודל במערך נתוני "האימון".  האלגוריתם עושה זאת על ידי ביצוע שינויים במודל "שמזיזים" אותו לאורך שיפוע כלומר, לאורך השיפוע של השגיאה כלפי מטה לעבר ערך השגיאה המינימלית.

אלגוריתמי ה- Gradient Descent נבדלים במובנים של מספר תבניות "האימון" בהן עושים שימוש על מנת לחשב את השגיאה או באיזה שלב מעדכנים את המודל (על סמך חישוב השגיאה). לפיכך קיימים שלושה מודלים עיקריים: Stochastic, Batch, Mini Batch.

Stochastic Gradient Descent או בקיצור SGD הוא וריאציה של האלגוריתם GD המחשבת את השגיאה ומעדכנת את המודל עבור כל דוגמה במערך האימונים. החיסרון העיקרי של השיטה הזו הוא התדירות הגבוהה של העדכונים במודל אשר גורמת לכך שלוקח זמן רב יותר לאמן מודלים של מערכי נתונים גדולים.

המודל השני, Batch GD, מחשבת את השגיאה עבור כל דוגמה במערך האימונים, אך מעדכנת את המודל רק לאחר שכל הדוגמאות האימון נערכו. החיסרון בשיטה זו הוא שההתכנסות היציבה של פונקציה השגיאה יכולה לגרום לבחירת ערכים שאינם אופטימליים עבור המערכת.

המודל השלישי, Mini Batch GD, מפצל את מערך האימון לקבוצות קטנות המשמשות המשמשות לחישוב שגיאת המודל ואת המקדמים המתאימים שישמשו לעדכון המודל בהתאם לשגיאה. בצורה זו, מודל זה מתגבר על שני החסרונות של המודלים הקודמים המופרטים לעיל.

2.3.1 Mini Batch Gradient Descent

אלגוריתם ה- Mini Batch Gradient Descent הוא ואריאציה של האלגוריתם Gradient Descent המפצל את מערך האימון לקבוצות קטנות המשמשות לחישוב שגיאת המודל ואת המקדמים המתאימים שישמשו לעדכון המודל בהתאם לשגיאה.

בין היישומים האפשריים, ניתן לבחור בסכימת הגרדיאנטים של הקבוצות הקטנות (Mini Batch ) או לחשב את ממוצע הגרדיאנטים אשר מקטין עוד יותר את השונות של השיפוע.

אלגוריתם זה מהווה איזון בין הרובסטיות של Stochastic Gradient Descent לבין היעילות של Batch Gradient Descent ועל כן זה היישום הנפוץ ביותר בתחום הלמידה העמוקה.

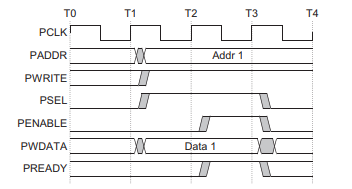
2.4 פרוטוקול - AMBA APB

ה- APB הוא חלק ממשפחת הפרוטוקולים של AMBA 3 המיועד לתקשורת בקצבים נמוכים יחסית, ונפוץ מאוד בגישות לקריאה וכתיבה של פרמטרי התקנים. הפרוטוקול מספק ממשק "בעלות נמוכה" משום שהפרוטוקול מותאם במיוחד לצריכת הספק מינימלית ומורכבות ממשק מופחתת.

בהתאם לפרוטוקול זה נגדיר את התנאים המתאים עבור פעולות כתיבה וקריאה, וכפי שנסביר בהמשך, כל פעולה מתבצעת לפחות בשני מחזורי שעון.

נתאר בקצרה את פעולת הכתיבה והקריאה עפ"י פרוטוקול זה:

* פעולת כתיבה ( Write transfer ): כמתואר באיור 2.4.1-



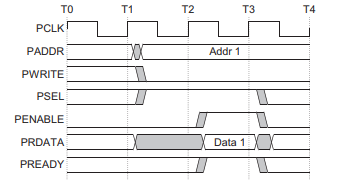
**איור 2.4.1 – תיאור Write transfer עפ"י פרוטוקול APB**

פעולת הכתיבה מתחילה עם הכתובת, המידע אותו רוצים לכתוב , סיגנל הכתיבה וסיגנל הבחירה. כלומר בעלית השעון הראשונה או קצת אחרי עליית השעון אם מתייחסים לזמני השהייה (במחזור הראשון – מסומן באדום באיור 2.4.1), מעדכנים בסינגל PADDR את כתובת הרגיסטר אליו רוצים לכתוב וב-PWDATA את המידע אותו רוצים לכתוב לרגיסטר ומעלים את הסיגנלים PWRITE, PSEL.

בעלית השעון השנייה או כאמור קצת אחרי עליית השעון אם מתייחסים לזמני השהייה (במחזור השני – מסומן בירוק באיור 2.4.1), מעלים את הסיגנלים PENABLE. כל הסיגנלים המפרטים עד כה נשלטים ע"י ה-master אשר יוזם את פעולת הכתיבה. מכיוון ש- PENABLE גבוהה ופעולת הכתיבה אכן הולכת להתבצע, ה-slave מעלה את סיגנל ה-PREADY ובך מאשר כי פעולת הכתיבה התבצעה. כל שאר הסיגנלים נשארים בתוקף גם במחזור הזה.

בעלית השעון השלישית (במחזור השלישי – מסומן בכחול באיור 2.4.1), מאפסים את כל הסיגנלים ובכך מסיימים את פעולת הכתיבה.

* פעולת קריאה ( Read transfer ): כמתואר באיור 2.4.2-



**איור 2.4.2 – תיאור Read transfer עפ"י פרוטוקול APB**

פעולת הקריאה מתחילה עם הכתובת, סיגנל הכתיבה וסיגנל הבחירה. כלומר בעלית השעון הראשונה או קצת אחרי עליית השעון אם מתייחסים לזמני השהייה (במחזור הראשון – מסומן באדום באיור 2.4.2), מעדכנים ב- PADDR את כתובת הרגיסטר ממנו רוצים לקרוא את המידע, מורידים את הסיגנלים PWRITE, ומעלים את סיגנל PSEL.

בעלית השעון השנייה (במחזור השני – מסומן בירוק באיור 2.4.2), מעלים את הסיגנלים PENABLE. כל שאר הסיגנלים נשארים בתוקף גם במחזור הזה וניתן לראות (באיור 2.4.2) שלאחר זמן השהייה מסוים, המידע אותו קראנו מהרגיסטר נמצא ב- PRDATA ובנוסף ה-slave מעלה את הסיגנל PREADY וכל עוד הסיגנל PREADY גבוה המידע שנמצא על ב-PDATA הוא תקף.

בעלית השעון השלישית (במחזור השלישי – מסומן בכחול באיור 2.4.2), מאפסים את כל הסיגנלים ובכך מסיימים את פעולת הקריאה.

2.5 המאמר

נשלים בהמשך

האם יש מה להוסיף או לשנות?

Machine learning

<https://whatis.techtarget.com/definition/unsupervised-learning>

<https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>

NN

<https://www.youtube.com/watch?v=IHZwWFHWa-w>

GD

<https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/>