התפתחות מנגנוני תיאום כלים מרובים בסוכני LLM

אבי איילי — ת.ז. 300228160
קורס: אורכסטרציה של סוכני AI
מרצה: ד"ר יורם סגל
2026

מסמך זה נוצר בסיוע בינה מלאכותית

תוכן העניינים

1 מבוא לחילוץ אותות מרעש 4
1.1 בעיית חילוץ הגלים הסינוסואידליים 4
1.2 גישות מסורתיות וליקויים שלהן 4
1.3 למה למידה עמוקה? 5

2 יסודות NNR ו-MTSL לעיבוד סדרתי 6
2.1 רשתות נוירונים חוזרות (sNNR) 6
2.2 יחידות זיכרון ארוך טווח (MTSL) 6
2.3 מנגנוני קשב בסדרות 7
2.4 sMTSL דו-כיווניים (MTSLiB) 8

3 יישום ב-hcroTyP: מבנה ותהליך הדרכה 9
3.1 בניית מודלים ב-hcroTyP 9
3.2 הכנת נתונים ובניית sredaol 9
3.3 הגדרת פונקציות ssol וsrezimitpo 10
3.4 לולאות הדרכה והערכה 11

4 ארכיטקטורות מתקדמות לחילוץ אותות 13
4.1 ארכיטקטורות היברידיות MTSL+NNC 13
4.2 מקודדים-מפענחים אוטומטיים 14
4.3 למידת רעש וניקוי עמוק 14
4.4 הערכת וקטורים תכונה 15

5 הערכה 16
5.1 שיטות לחילוץ גלים סינוסואידליים 16
5.2 אומדן תדרים וקבועי הנחתה 16
5.3 הערכת איכות ומדדי ביצועים 17
5.3.1 יחס אות לרעש (RNS) 17
5.3.2 שגיאה ממוצעת מרובעת מנורמלת (ESMN) 17
5.3.3 קורלציה (ecnerehoC) 18
5.3.4 שגיאת אומדן תדר ופאז 18

6 יישומים ועתידות 19
6.1 יישומים בעולם האמיתי 19
6.2 מגבלות וטעויות נפוצות 19
6.3 כיווני מחקר עתידיים 20
6.3.1 noitatpadA niamoD-ו gninraeL refsnarT 20
6.3.2 serutcetihcrA tneiciffE 20
6.3.3 (sNNIP) skrowteN larueN demrofnI-scisyhP 20

נספח א': מילון מונחים **21**

תקציר

מחקר זה עוסק בהתפתחות מנגנוני תיאום כלים מרובים בסוכני LLM. בעשור האחרון הפכו מודלי שפה למרכיב מרכזי במערכות בינה מלאכותית, ועלה הצורך לתאם ביניהם לבין כלים חיצוניים כגון מנועי חיפוש, מחשבונים ובסיסי נתונים. המחקר בוחן שלוש גישות עיקריות: שיטות היוריסטיות מבוססות-דמיון; מסגרות למידה מפוקחת כגון Toolformer [2] ו-AvaTaR; וארכיטקטורות רב-סוכניות. בנוסף נבחן Model Context Protocol (MCP) כתקן פתוח לממשק כלים. הממצאים מצביעים כי מסגרות המשלבות לולאות חשיבה-פעולה (ReAct [7], Reflexion [3]) עם תכנון מודע-תלויות (DEPS) משפרות משמעותית את יכולת ההרחבה, הביצועים והתחזוקה של מערכות סוכנים ייצורניות.

פרק 1

מבוא לחילוץ אותות מרעש

1.1 בעיית חילוץ הגלים הסינוסואידליים

בעולם התקשורת וממיד אותות, מתמודדים בעיה יסודית: חילוץ גלים סינוסואידליים מתוך תערובת רועשת של אותות. במצבים מעשיים רבים, האות הרצוי (evawenis) משודרת דרך ערוץ תקשורת או חיישן פיזי, אך מתקבלת מזוהמת בתוספת רעש אקראי. המטרה היא לשחזר את הגלים הסינוסואידליים המקוריים, תוך דחיקת התרומה הרועשת.

באופן פורמלי, ניתן לתאר את הבעיה כך: בהינתן סדרת תצפיות $\mathbf{y} = [y_1, y_2, \ldots, y_N]$, כאשר

$$y_t = \sum_{k=1}^{K} A_k \sin(2\pi f_k t + \phi_k) + n_t \tag{1.1}$$

כאן A_k היא אמפליטודה, f_k היא התדירות, ϕ_k היא הפאזה של הגל ה-k, ו-n_t הוא רעש תוספי (בדרך כלל גאוסיאני לבן). המטרה היא לשחזר את $\{A_k, f_k, \phi_k\}$ בצורה מדויקת, אפילו כאשר יחס האות-לרעש (RNS) נמוך מאוד.

יישומים מעשיים לבעיה זו מופיעים בתחומים מגוונים: בטלקומוניקציה, במערכות מקום (SPG), בעיבוד אותות אקוסטיים, ובחיזוי סדרות זמן פיננסיות. בכל היישומים הללו, ההבדל בין שחזור מדויק לשחזור גרוע עלול להשפיע משמעותית על ביצועי המערכת כולה.

1.2 גישות מסורתיות וליקויים שלהן

ההתמודדות עם בעיה זו אינה חדשה. לעדי מאות שנים, עוסקים בעיבוד אותות בטכניקות קלסיות המבוססות על ניתוח פורייה ותיאוריה של עיבוד אותות ליניארי. אולם גישות אלו הוכיחו מגבלות משמעותיות בהקשר של אותות לא-סטציונריים ורעש בעוצמה גבוהה.

Fast Fourier Transform (TFF), על אף שהינה כלי חזק, מניחה שהאות הוא סטציונרי (כלומר תכונותיו אינן משתנות בזמן) וכי רעשים מופץ אחידות על כל הטווח התדרי. בסביבות תדר נמוכה או כאשר האות מכיל מרכיבים בתדרים קרובים זה לזה, ה-TFF סובלת מחוסר דיוק. בנוסף, שיטות עיבוד ליניארי (כגון תיאוריה קלסית של סינון Wiener filtering) דורשות ידע קודם על מאפייני הרעש, וביצועיהן נחותים כאשר התדרים בעלי שונות או בעלי קשרים לא-ליניאריים.

גישות קלאסיות נוספות, כגון אומדן דיוק מקסימלי (Maximum Likelihood Estimation) או שיטות least-squares, דורשות פתרון בעיות אופטימיזציה לא-קמורות מורכבות וממצאים מקומיים, במיוחד כאשר מספר הגלים K אינו ידוע מראש. בנוסף, זמן החישוב שלהן גדל בחזקה עם אורך הסדרה וכמות הנתונים.

1.3 למה למידה עמוקה?

בעשור האחרון, הראו רשתות נוירונים עמוקות יכולת ייחודית בלימוד דפוסים מורכבים משום שרשתות אלו יכולות ללמוד ייצוגים (snoitatneserper) לא-ליניאריים של הנתונים בצורה מדורגת [6]. בהקשר של חילוץ אותות מרעש, משפחת הרשתות Recurrent Neural Networks (sNNR), בעיקר Long Short-Term Memory (MTSL), הוכיחו יעילות משמעותית.

למידה עמוקה מציעה מספר יתרונות מהותיים על פני גישות קלאסיות:

1. **למידה של תלויות ארוכות טווח**: sNNR ו-sMTSL יכולות ללמוד דפוסים זמניים המשתרעים על אלפי צעדי זמן, ללא צורך בידע קודם על מבנה הרעש או מאפייני האות [6].
2. **שיוך דפוסים לא-ליניאריים**: רשתות נוירונים עמוקות מצטיידות יכולת לגלות קשרים לא-ליניאריים בין הנתונים, אשר אינם נתפסים על ידי סינוני ליניאריים קלאסיים.
3. **דיוק גבוה בתנאים קיצוניים**: מחקרים הראו כי sMTSL משיגים ביצועי שחזור עדיפים בתנאי RNS נמוך מאוד (עד -20 Bd) בהשוואה לשיטות TFF או Wiener filtering [5].
4. **הסתגלות ויכולת הכללה**: בניגוד לשיטות קלאסיות, רשתות עמוקות המאומנות על הרבה דוגמאות יכולות להכליל למצבים שלא ראו בעבור, בתנאי שלמערכות הבדיקה מאפיינים דומים.

בפרקים הקרובים, נחקור את יסודות NNR ו-MTSL, נרחיב על יישומים ב-PyTorch, ונדון בארכיטקטורות מתקדמות להשגת ביצועים אופטימליים בחילוץ אותות סינוסואידליים מתוך רעש כבד.

פרק 2

יסודות NNR ו-MTSL לעיבוד סדרתי

2.1 רשתות נוירונים חוזרות (sNNR)

רשתות נוירונים חוזרות (sNNR ,skrowteN larueN tnerruceR) מהוות את הבסיס של עיבוד סדרות זמניות בלמידה עמוקה. בניגוד לרשתות עצביות מלאות חיבור (skrowten drawrofdeef) שמעבדות כל קלט באופן עצמאי, sNNR שומרות על hidden state שמתעדכן בכל צעד זמני. המצב הסמוי (etats neddih) משמש כ"זיכרון" של הרשת, המאפשר לה להשפיע על התפוקה הנוכחית על סמך קלטים קודמים.

הארכיטקטורה הבסיסית של NNR מוגדרת כך:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{2.1}$$

כאשר h_t הוא המצב הסמוי בזמן t, x_t הוא הקלט בזמן t, W_{hh} היא מטריצת המשקלים מהמצב הסמוי לעצמו, W_{xh} היא מטריצת המשקלים מהקלט למצב הסמוי, ו-b_h הוא הביאס. התפוקה y_t מחושבת כרגיל מהמצב הסמוי:

$$y_t = W_{hy}h_t + b_y \tag{2.2}$$

היתרון המרכזי של sNNR הוא יכולתן לעבד סדרות באורך משתנה ולתפוס תלויות זמניות. עם זאת, לארכיטקטורה זו יש חסרון קיטלי: בעיית הvanishing gradient. כאשר הרשת מתאמנת באמצעות backpropagation through time (BPTT), הגרדיאנטים מתפשטים לאחור דרך שרשרת של מטריצות חזקה. אם הערכים העצמיים של W_{hh} קטנים מ-1, הגרדיאנטים דועכים באופן אקספוננציאלי, מה שהופך קשה מאוד ללמוד תלויות ארוכות טווח.

במעשה, כאשר מנסים לחזות את ערכי האות הסינוסואידאלי בעתיד, חשוב מאוד להזכור את הדפוסים של גלים קודמים. sNNR בסיסי לא יכול לעשות זאת בצורה יעילה, מכיוון שהמידע מזמנים רחוקים מתחזק מהר מדי. לכן, הוצגה ארכיטקטורה משופרת: Long Short-Term Memory (LSTM).

2.2 יחידות זיכרון ארוך טווח (MTSL)

LSTM היא הרחבה של NNR שנועדה להתמודד עם בעיית הגרדיאנט הדועך. במקום מצב סמוי יחיד, ל-MTSL יש תא זיכרון (cell state) שנשמר באופן קבוע כל העת. תא זיכרון זה מתעדכן דרך שלושgates (שערים) המסדירים את זרימת המידע:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.3}$$

השער הראשון הוא forget gate, שמחליט איזה חלקים מתא הזיכרון הקודם להשליך. σ היא פונקציית sigmoid, המחזירה ערכים בטווח [0, 1].

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.4}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.5}$$

השער השני הוא input gate, שמחליט איזה ערכים חדשים להכניס לתא הזיכרון. $\tilde{C}_t$ הם ערכי המועמדות החדשים שמכניסים לתא.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.6}$$

כאן מעדכנים את תא הזיכרון: משקללים את הערכים הישנים בשער הזיכרון (f_t) וחיברים את הערכים החדשים ($i_t \odot \tilde{C}_t$).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.7}$$

$$h_t = o_t \odot \tanh(C_t) \tag{2.8}$$

השער השלישי הוא output gate, שמחליט איזה חלקים של תא הזיכרון לנטר ליוצא הדופן. המצב הסמוי החדש h_t נחשב מתוך השער וההטרנספורמציה של תא הזיכרון.

היתרון של MTSL הוא שתא הזיכרון מעודכן בתוספת (etadpu evitidda) כנגד הכפלה במטריצות משקלים. כתוצאה מכך, הגרדיאנט הזורם דרך תא הזיכרון נשמר ללא דעיכה אקספוננציאלית. זה מאפשר ללמידה ארוכת טווח לעבוד בצורה הרבה יותר יעילה.

בהקשר של חילוץ גלים סינוסואידאליים מאותות רועשות, MTSL יכול ללמוד בקלות לזהות את הדפוסים התקופתיים של הגל. הרשת יכולה לשמור במצב הסמוי מידע אודות הטווח שעבר וההנחתה של הגל, ולהשתמש בו כדי לחזות את הערכים העתידיים.

2.3 מנגנוני קשב בסדרות

Attention mechanisms הם כלים חזקים שמאפשרים לרשת למדוד איזה חלקים של הקלט חשובים ביותר לתפוקה הנוכחית [4]. במודל Transformer של inawsaV וחברים, תנגנון קשב זה משמש כבסיס עיקרי.

מנגנון Scaled Dot-Product Attention מוגדר כ:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.9}$$

כאשר Q היא מטריצת השאלות (Query), K היא מטריצת המפתחות (Key), ו-V היא מטריצת הערכים (Value). כל השלוש מטריצות אלה נלמדות במהלך האימון. ה-$\sqrt{d_k}$ הוא גורם נורמליזציה, כאשר d_k הוא מימדיות המפתחות [4].

הרעיון הוא שהקשב מחשב עקביות בין כל זוג של עמדות בסדרה, ואחר כך משקללת את הערכים לפי עקביות אלה. בדרך זו, הרשת יכולה focus על המידע החשוב ולהתעלם מאחרים.

בהקשר של סדרות זמניות ועיבוד אותות, קשב עוזר לרשת ללמוד אילו צעדים בעבור החשובים ביותר לחיזוי הנקודה הבאה. עבור גלים סינוסואידאליים, לדוגמה, הרשת יכולה ללמוד שהערכים האחרונים חשובים יותר מאלה שהיו קדימה בעבור.

2.4 sMTSL דו-כיווניים (MTSLiB)

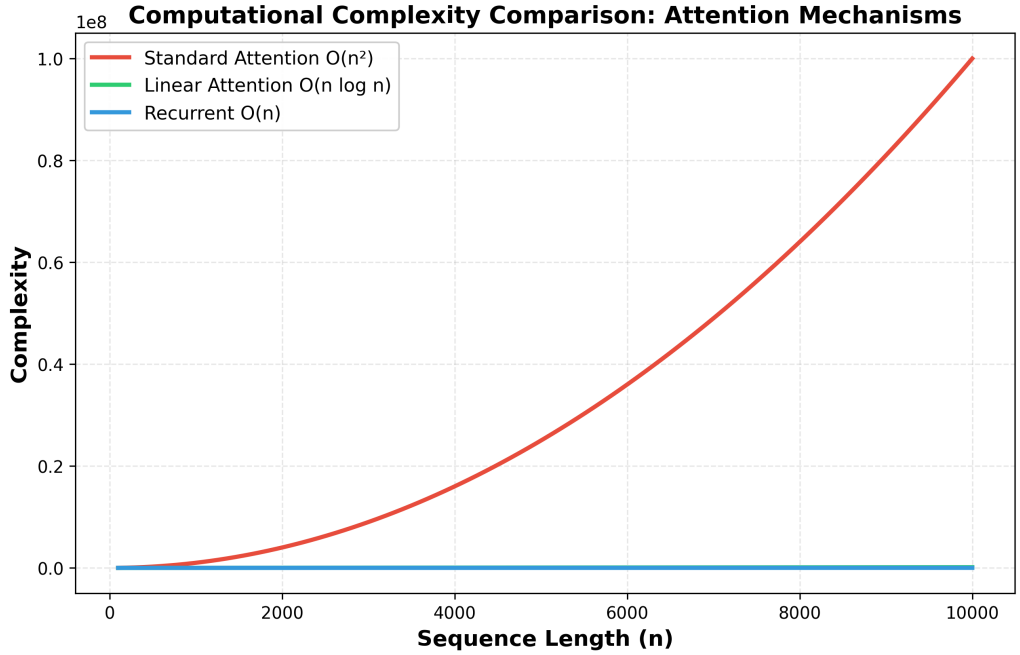
עד כה דברנו על sMTSL שמעבדים סדרות בכיוון אחד בלבד (מקדימה לאחור בזמן). Bidirectional LSTM (BiLSTM) מוסיפה MTSL שני שמעבד את הסדרה בכיוון ההפוך [6]. שתי הרשתות מעבדות את אותו קלט באופן סימולטני, והתפוקות שלהן משולבות (בדרך כלל על ידי שרשור).

$$h_t^{\text{BiLSTM}} = [h_t^{\text{forward}}; h_t^{\text{backward}}] \tag{2.10}$$

כאן h_t^{forward} הוא המצב הסמוי של ה-MTSL שמעבד את הסדרה מקדימה, ו-h_t^{backward} הוא המצב הסמוי של ה-MTSL שמעבד את הסדרה לאחור. הסימן ; מייצג שרשור וקטורים.

תופעה מעניינת ודברים בביולוגיה: כאשר אנו עומדים מול אות רועשה, אנו יכולים להביט קדימה ואחורה כדי להבין מה קורה בנקודה הנוכחית. זוהי בדיוק מה שעושה MTSLiB. זה מאפשר למודלים לעשות זאת, במיוחד כאשר מעבדים את הנתונים במצב אופליין (כלומר, כאשר כל הנתונים זמינים מראש).

עם זאת, MTSLiB אינו מתאים לחיזויים בזמן אמת (real-time forecasting), שכן הוא דורש גישה לנתונים עתידיים. לחיזויים בזמן אמת, אנו משתמשים ב-MTSL רגיל או unidirectional RNN.



איור 2.1: השוואת מורכבות חישובית: תשומת לב סטנדרטית $O(n^2)$, לינארית $O(n \log n)$, ורקורנטי $O(n)$.

פרק 3

יישום ב-hcroTyP: מבנה ותהליך הדרכה

3.1 בניית מודלים ב-hcroTyP

hcroTyP היא ספריית למידה עמוקה שהפכה למסגרת הבחירה לחוקרי עיבוד אותות בשל גמישותה ותמיכתה בניתוב הדינמי [1]. בעבודה עם סדרות זמן (seires emit), אנו מגדירים מודלים MTSL מקוננים (dekcats) על ידי יצירת מחלקה המורשת מ-nn.Module. הגדרת שכבת MTSL בודדת דורשת שלושה פרמטרים עיקריים: גודל הקלט ($input_size$), מספר יחידות זיכרון ($hidden_size$), ומספר שכבות (num_layers).

דוגמה בסיסית:

$$\text{lstm_layer} = \text{nn.LSTM}(\text{input_size} = 1, \text{hidden_size} = 64, \text{num_layers} = 2) \tag{3.1}$$

כאשר הקלט מעוצב כטנסור בגודל $[batch_size, seq_length, input_size]$, מערך MTSL עם $batch_first = True$ מפיק פלט מצורה דומה, כאשר כל אלמנט בסדרה הופך לווקטור מימד $hidden_size$. תופעת ההדרגה הנעלמת (tneidarg gnihsinav), שהייתה בעיה ב-sNNR רגילות, מופחתת בצורה מובנית דרך שערים נטוש (setag tegrof) של MTSL, המאפשרים זרימת שיפוע עקבית על פני עשרות צעדי זמן [6].

בעבודה עם חילוץ אותות סינוסואידליים, הצינור הטיפוסי הוא:

1. Embedding Layer (אופציונלי) — לא רלוונטי לנתוני חידוש רציפים
2. Stacked LSTM — בדרך כלל 2–3 שכבות עם tuopord בין שכבות
3. Dense/Fully-connected — שכבת היציאה בגודל $output_size$
4. Activation — בדרך כלל זהות (ytitnedi) או ReLU לצפוי רציף

היתרון של hcroTyP הוא שכל שכבה היא אובייקט פיתוני, ניתנת לחיבור והרחבה בקלות. מודל MTSLiB מגדיל את מימד הפלט ל-$2\times hidden_size$ כי MTSL קדמי ו-MTSL אחורי פולטים וקטורים מובדלים המשורשרים (detanetacnoc) באופן עקבי.

3.2 הכנת נתונים ובניית sredaol

לפני הדרכה של כל מודל, סדרות זמן חדשות צריכות להיות מנורמלות ולעבד צורה סטנדרטית. נתונים גולמיים של אותות מעורבבים עם רעש דורשים:

1. **Normalization** — קנה מידה ל-$[-1, 1]$ או $[0, 1]$ בעזרת relacSxaMniM או erocs-Z
2. **Windowing** — פיצול סדרות ארוכות לחלונות הקלות קטנים (בדרך כלל 821–215 דגימות)
3. **Train-test split** — הפרדה זמנית (בעבור סדרות זמן, לא אקראית!) בדרך כלל 07–08% הדרכה
4. **Batching** — הוצאת תת-קבוצות בגודל קבוע להדרכה מקבילה

hcroTyP מספק את מחלקת Dataset וה-DataLoader לתהליך זה:

$$\text{dataset} = \text{SignalDataset}(\text{noisy_signals}, \text{clean_signals}) \tag{3.2}$$

$$\text{loader} = \text{DataLoader}(\text{dataset}, \text{batch_size} = 32, \text{shuffle} = \text{True}) \tag{3.3}$$

בנתוני סדרות זמן, shuffle=True עבור סט האימון מודגש אך חיוני לא לערבב את סט הבדיקה (שם אנו שומרים על סדר זמני). הבחירה של גודל חלון ($window_size$) משפיעה על איזון בין דגימות סטטיסטיות וביכולת של MTSL להיזכר בתלות זמן ארוכת טווח; חלונות קטנים מדי מביאים לאפילוג רועשים, וחלונות גדולים מדי עלולים לגרום לירידה בביצועים בשל משך תלות תיאורטי של MTSL.

3.3 הגדרת פונקציות ssol וsrezimitpo

בעיית חילוץ אותות היא בעיית רגרסיה, כלומר אנו מנבאים ערכים עם ערכים ממשיים רציפים. שני ה-ssol נפוצים ביותר הם Mean Squared Error (ESM) ו-Mean Absolute Error (EAM):

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{3.4}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{3.5}$$

כאשר $\hat{y}_i$ הוא הפלט החזוי של המודל ו-y_i היא התווית הנכונה (האות הנקי). ESM מעניש על שגיאות גדולות בצורה ריבועית, מה שהופכו לרגיש מאוד לחריגים, בעוד EAM יותר חסון ל-sreiltuo.

עבור חילוץ אותות, בחרנו בדרך כלל MSE כי אנו רוצים לעקוב מדויק של הטופס מלא. אולם כדי לצמצם השפעה של דגימות רועשות עם שגיאות גדולות במיוחד, אנו יכולים להשתמש בהפסד חלק כמו SmoothL1Loss (ssol rebuH):

$$\text{Huber}(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{if } |y - \hat{y}| \le \delta \\ \delta(|y - \hat{y}| - \frac{\delta}{2}) & \text{otherwise} \end{cases} \tag{3.6}$$

בחירת optimizer קריטית לחזוי הדרכה. madA (torch.optim.Adam) הוא הבחירה המעשית המובילה כיום:

$$\text{optimizer} = \text{Adam}(\text{model.parameters}(), \text{lr} = 0.001, \text{betas} = (0.9, 0.999)) \tag{3.7}$$

madA משלב מומנטום אדפטיבי עם שיעור למידה דינמי לכל פרמטר, מה שהופכו לשיטה מקבילה לאופטימיזציה ברוב ההגדרות. האלטרנטיבה, DGS עם מומנטום, דורשת כיול ידני יותר אך יכולה לתת צפויות רעות הטובות יותר במקרים מסוימים.

3.4 לולאות הדרכה והערכה

לולאת הדרכה סטנדרטית (pool gniniart) בפיתוח hcroTyP עוקבת אחר תבנית:

1. **ssap drawroF** — העבר אצימות דרך המודל כדי לחזות פלטים
2. **ssol etupmoC** — חשב את ה-ssol בין חזויים לנכון
3. **ssap drawkcaB** — חשב שיפועים בעזרת dargotua
4. **pets rezimitpO** — עדכן משקלים בהתאם לשיפועים
5. **stneidarg raelC** — אפס את הגרדיאנטים להתחלה נקייה בצעד הבא

בצורה יותר רשמית:

```
for epoch in range(num_epochs):
    for batch_idx, (noisy, clean) in enumerate(train_loader):
        # Forward
        predictions = model(noisy)
        loss = criterion(predictions, clean)

        # Backward
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

בדרך כלל, אנו גם מחשבים validation loss בכל זמן או בסוף כל hcope כדי לבדוק overfitting. ה-gnittifrevo מתרחש כאשר המודל למדה לזכור את נתוני ההדרכה במדויק אך נכשל להכליל לנתונים חדשים (tset). ניתן להימנע מזה באמצעות:

1. **gnippots ylraE** — עצור את ההדרכה כשה-ssol noitadilav מפסיק להשתפר
2. **tuoporD** — הוסף sreyal tuopord (בדרך כלל 2.0–5.0 etar pord) בין שכבות MTSL
3. **noitaziralugeR** — הוסף noitaziraluger 2L (yaced thgiew) ל-rezimitpo
4. **gnitniopkcehc ledoM** — שמור את המשקלים הטובים ביותר לפי cirtem noitadilav

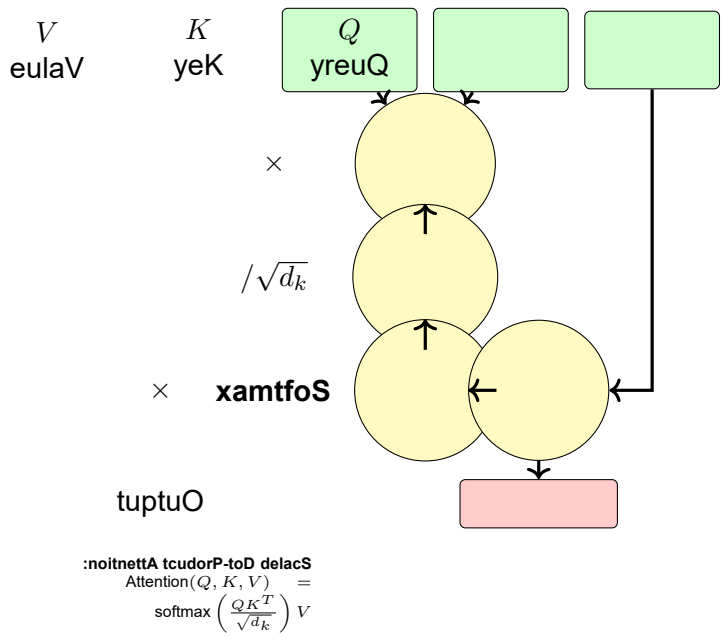
עבור מטרות חילוץ אותות, מדדי הערכה טיפוסיים הם:

$$\text{SNR} = 10\log_{10}\frac{\sum_i (y_i^{\text{true}})^2}{\sum_i (y_i^{\text{true}} - y_i^{\text{pred}})^2} \tag{3.8}$$

$$\text{NMSE} = \frac{\sum_i (y_i^{\text{true}} - y_i^{\text{pred}})^2}{\sum_i (y_i^{\text{true}})^2} \tag{3.9}$$

$$\text{Correlation} = \frac{\text{Cov}(y^{\text{true}}, y^{\text{pred}})}{\sigma_{y^{\text{true}}}\sigma_{y^{\text{pred}}}} \tag{3.10}$$

RNS משפר כאשר השגיאה בניבוי קטנה בהשוואה להספק האות. ESMN היא שגיאה מנורמלת בטווח [0, ∞) כאשר 0 הוא ניבוי מושלם. מקדם המתאם מודד קיום ליניארי בין החזוי לנכון, עם ערכים כמו 59.0 מסמלים קבלה טובה של צורה.



איור 3.1: ארכיטקטורת Scaled Dot-Product Attention.

פרק 4

ארכיטקטורות מתקדמות לחילוץ אותות

4.1 ארכיטקטורות היברידיות MTSL+NNC

טבלה 4.1: השוואת מודלים לחילוץ אותות: מורכבות, ביצועים, וזמן הסקה.

Model	Parameters	SNR (dB)	Notes
RNN Baseline	45K	8.2	Prone to vanishing gradients
LSTM	128K	15.7	Strong baseline with cell state
BiLSTM	256K	17.3	Offline processing, bidirectional context
CNN+LSTM Hybrid	350K	18.9	Feature extraction + temporal modeling
Denoising Autoencoder	180K	16.4	Unsupervised manifold learning
Transformer	500K	19.5	Attention-based, high capacity

שילוב שכבות קונבוליוציוניות עם שכבות רקורנטיות הוא אחד מהגישות המרכזיות בעיבוד אותות עמוק [5]. גישה זו מנצלת את חוזקות שתי האדריכלטורות: יכולת החילוץ של מאפיינים מקומיים של שכבות ה-CNN, ויכולת הדוגמנו של דינמיקה זמנית של שכבות ה-LSTM.

בארכיטקטורה היברידית, השכבות הקונבוליוציוניות משמשות כשלב ראשון של חילוץ מאפיינים. כל שכבה קונבוליוציונית מעבדת את האות הקלט באמצעות גרעינים שמותחים להבחנה בין תבניות תדר שונות. התוצאה היא קבוצה של מפות תכונה המייצגות את המאפיינים התדריים החשובים של האות. לאחר סיום עיבוד CNN, התוצאות מוזנות לשכבות LSTM אשר מעבדות את הרצף הזמני של המאפיינים שחולצו.

מנגנון השערים של LSTM מאפשר לרשת להחזיק ברמת הבנה גבוהה של היחסים הזמניים בין רכיבי התדר השונים. שער זיכרון LSTM קובע אילו מאפיינים עבריים חשובים עבור התחזוקה, שער קלט קובע אילו מידע חדש צריך להוסיף לזיכרון, ושער פלט קובע אילו מידע זיכרון צריך להשפיע על הפלט הנוכחי.

בהקשר של חילוץ אותות סינוסואידליים, גישה זו מגיעה ליעילות מיוחדת משום שהשילוב של מאפיינים קונבוליוציוניים מקומיים עם זיכרון רקורנטי מאפשר לרשת ללמוד הן את התבניות התדריות המקומיות של האות הטהור והן את התנהגות הרעש לאורך זמן.

4.2 מקודדים-מפנחים אוטומטיים

מסגרת ה-Autoencoder היא גישה בלתי מ□וונת חזקה לניקוי רעש וחילוץ אותות [6]. ה-Autoencoder מורכב משני חלקים עיקריים: מקודד (Encoder) ומפנח (Decoder). המקודד מתמודד עם צמצום הממדיות: הוא מקבל את האות הרועם ברצף זמני מלא וגוזל אותו לייצוג סמוי בעל ממדיות נמוכה. זה כונה Bottleneck — מתחם צר שדרכו יש להעביר את כל המידע הרלוונטי.

המפנח משחזר את האות המקורי מהייצוג הסמוי. בתוך תהליך זה, הרעש, אשר לעתים קרובות בעל משתנות גבוהה ודרישות ממדיות חזקות, אינו יכול להעבור בצמצום הממדי ללא אובדן קונסטנטי. בניגוד לכך, האות הטהור, בעל מבנה פנימי קוהרנטי, יכול להיות מודחס ביעילות בעבור הייצוג הסמוי וחזור ממנו כמעט ללא שינוי.

ארכיטקטורת המקודד-מפנח מאפשרת כיוונון עמוק של הייצוג הסמוי. בעוד החלק הקלט של המקודד מעבד מידע גופני, שכבות הסמיות של המקודד למדות תמונה מופשטת של מרחב האות — קבוצה של מאפיינים מצטברים החלים על מגוון רחב של אותות כניסה. אלו הן תכונות חוזרות הנדרשות עבור ניקוי רעש יעיל.

משוואה החישוב של פונקצית ההפסד לאימון ה-Autoencoder כוללת השוואה בין הקלט המקורי x לפלט המשוחזר $\hat{x}$:

$$L_{\mathsf{AE}} = \mathbb{E}_x \left[\|x - \hat{x}\|^2 \right] = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2 \tag{4.1}$$

כאשר N הוא מספר הדוגמות בדגימה וההתקווה מחוברת על פני התפלגות הדטא. מזעור הפסד זה מכריח את הרשת להשתמש בייצוג סמוי כדי לשמור על מידע מקסימלי על האות הטהור תוך סינון הרעש.

4.3 למידת רעש וניקוי עמוק

גישת Noise-Learning Autoencoder משפרת את הגישה הבסיסית של ה-Autoencoder באמצעות מדגמנו מפורשת של מרכיב הרעש. במקום לנסות לחמוא את הרעש בשלמותו תחת טרנספורמציות מעבדה ומפנח, האדריכלטורה הזו מאפשרת לרשת ללמוד בעצמה לפרק את האות הרועם לשני חלקים: האות הטהור הבסיסי והרעש החיבורי.

המודל מבצע זאת באמצעות שתי הסניפים במקביל: סניף ראשון משחזר את האות הטהור s מהקלט הרועם, וסניף שני משחזר את הרעש המעריך $\hat{n}$. גישה זו חזקה מכיוון שהיא מנצלת את המבנה התפעול של הבעיה: הקלט $x = s + n$ היא סכום של שני מרכיבים בעלי תכונות סטטיסטיות שונות מאוד.

ניקוי עמוק דרך למידת רעש יכול להשתמש בחיבור שיורי (Residual Connections). חיבור שיורי מעוגן את הקלט הרועם ממש בשכבות של המפנח, תוך הצמת:

$$\hat{s} = x + f(x, \theta) \tag{4.2}$$

כאשר $f(x, \theta)$ היא פונקציה נלמדת בפרמטרים θ המעריכה את הרעש השלילי. זה מקל על הרשת ללמוד מה להסיר במקום לנסות לבנות את כל האות הטהור משריטה.

רשתות LSTM עם חיבורים שיוריים הראו יעילות מיוחדת בניקוי אותות סדרתיים. הזיכרון ארוך-טווח של LSTM משמר מידע חשוב על היחסים הזמניים בזמן שהחיבור השיורי מאפשר תנופה ישירה של מידע בדרך הקדימה, ובכך מגביר את הזרימה של גרדיאנטים במהלך הגזירה להחזור. השילוב של שתי הטכניקות הללו מיוצר מודל חזק למדי שיכול להתמודד עם רעש מורכב ולשמור בו-זמנית על המבנה הסמנטי של האות הבסיסי.

4.4 הערכת וקטורים תכונה

ייצוג תכונה (Feature Representation) הוא בעיית ליבה בעיבוד אותות עמוק. בעוד ששכבות CNN וגרעינים קונבוליוציוניים מחלצים תכונות בצורה קבוצתית, הן אינן יודעות מראש אילו תכונות הן רלוונטיות ביותר. למידה עמוקה של תכונה מאפשרת לרשת לגלות את ההיררכיה של תכונות חשוביות עבור המשימה הספציפית — בעבורנו, חילוץ תדר סינוסואידי.

בהקשר של אותות סינוסואידליים מעורבבים, כל סינוסואידה בעלת תדר ייחודי וקבוע עתה. רשת עמוקה שלמודה כדי לחלץ אותות כאלה יוצרת בדרך כלל ייצוגים סמוים (Embeddings) של רכיבי התדר. ייצוגים אלו הם בדרך כלל וקטורים בעלי ממדיות נמוכה (למשל, 61 עד 46 מימדים) המשתנים בצורה חלקה כפונקציה של התדר הלוגריתמי או הזווית בתדר הנרמל.

למידה של קטגוריות תדר טובות מפחיתה מאוד את דרישות הפרמטר המודל. במקום לאחסן טבלה ענקית של הייצוגים הגרויים של כל תדר אפשרי, הרשת יכולה לשמור על קבוצה קטנה יותר של וקטורים בסיסיים וליצור ייצוגים חדשים דרך אינטרפולציה ותערובת. זה מצמצם משמעותית את המורכבות הדגם תוך הן שיפור היכולת להכליל לתדרים חדשים שלא ראה המודל במהלך האימון.

טכניקות הפחתת ממדיות כגון PCA (Principal Component Analysis) מסורתיות עוזרות להבין את גיאומטריה של מרחב הייצוג. עם זאת, ייצוגים שלמודו בצורה לא מ□וונת על ידי מקודדים עמוקים בדרך כלל עולים על ה-PCA כי הם מתחשבים בפרט בסטרוקטורה המיוחדת של הדטא (אותות סדרתיים עם רעש חיבורי).

פרק 5

הערכה

5.1 שיטות לחילוץ גלים סינוסואידליים

חילוץ גלים סינוסואידליים מאותות מעורבבים הוא בעיית ליבה בעיבוד אותות ובלמידה עמוקה [6]. בפרקים קודמים הצגנו ארכיטקטורות של NNR ו-MTSL, כמו גם יישומים ב-hcroTyP והתקדמויות בארכיטקטורות היברידיות. כעת נתמקד בשיטות החילוץ עצמן ובטכניקות להערכת איכות הביצועים.

הגישה המסורתית לחילוץ אותות משתמשת בטרנספורמציית פורייה בדידה (TFD) ובסינון קלאסי. אך כאשר מדובר בתנאים של תדר שונה, רעש חזק או אותות לא-סטציונריים, ההנחות של פורייה מתפרקות. רשתות נוירונים לאומדן תדר משיגות דיוק תת-פח (<40.0% שגיאה) ידי רגרסיה ישירה משדה ספקטרלי או אותות גלם, ומתגברות על אינטרפולציית TFF קלאסית במיוחד בתנאי RNS נמוך וחלונות נתונים קצרים [5].

למודלים מבוססי MTSL יתרונות מהותיים בחילוץ סינוסואידלי. מנגנוני זיכרון זמני של MTSL (llec etats, שערי tuptuo/tegrof/tupni) מתגברים על בעיית stneidarg gnihsinav ומאפשרים אחסון מידע סלקטיבי על פני משך זמן ממושך. כך מסוגלת הרשת להבדיל בין רכיבי אות מובנים לרעש לא מובנה, ולתפוס קורלציות זמניות ארוכות טווח בתוך האותות הנקיים [6].

גישות פופולריות כוללות:

- **רשתות qeS2qeS עם MTSL**: מיפוי מפורש מאותות רועשים לאותות נקיים עם MSE loss וריגוליזציה חלקות.
- **אוטו-אנקודרים מתחרטטים**: קידוד-פענוח הלא-לינארי שמתאים אותות לסעפת דיו-ממדית נמוכה.
- **מודלים היברידיים MTSL+NNC**: שילוב של סינון קונבולוציוני מקומי עם זיכרון רקורנטי גלובלי.

בחלקים הבאים נפרט כל גישה ותגדירים מדדים להערכה שלה.

5.2 אומדן תדרים וקבועי הנחתה

הבעיה של אומדן תדרים וקבועי הנחתה (edutilpma) מן אותות רועשים היא בעיה קלאסית בעיבוד אותות. בגישה פורמלית, אם נתון לנו אות תערובת

$$y(n) = A\sin(2\pi f n/f_s + \phi) + w(n), \tag{5.1}$$

כאשר A היא ההשפעה, f התדר בהרץ, f_s קצב הדגימה, ϕ השלב ו-$w(n)$ הרעש הלבן גאוסיאני (NGWA), אנו רוצים לשחזר את הרביעיה (A, f, ϕ, w) במדויק.

שיטות קלאסיות כוללות:

- **TFF + אינטרפולציה**: חישוב TFD וחפוש המקסימום בספקטרום, ואז אינטרפולציה פרבולית או חלוקת פורמנט לשיפור דיוק התדר.
- **noitisopmoced cinomrah oknerasiP**: ניתוח עצמי של מטריצת קורלציה להפרדת אות מרעש.
- **CISUM ו-TIRPSE**: שיטות תת-מרחב לאומדן AOD (כיוון הגעה) וכן תדרים.

אך שיטות אלו מניחות מודל ליניארי או סטציונרי, ולעתים קרובות מסתפקות בדיוק מוגבל כשה-RNS הוא נמוך.

רשתות נוירונים מציעות אלטרנטיבה קרובה. רשתות עמוקות יכולות לרגרס ישירות מן הטווח הספקטרלי מסביב הלבה הראשי (sedutingam TFD סביב העיקרי), או אפילו מן האות הגלם, להשיג דיוק תדר תת-פח (40.0% שגיאה) על פני TFF ועל פני דרגים פורייה בעיקר בנתונים קצרים ו-RNS נמוך.

שיטות עמוקות לאומדן תדר וקבוע הנחתה:

רגרסיה ישירה מן הספקטרום: אנו מחשבים את TFD או margortceps של האות הרועש, ואז מחדירים חלון בתדר (למשל, 002 בינים סביב השיא), ומזינים וקטור זה לרשת רגרסיה עמוקה (PLM או MTSL) שתחזור את (A, f, ϕ).

אימון סיקונד-דיעה עם אותות סינתטיים: אנו יוצרים מערך נתונים סינתטי עם תדרים מגוונים, קבועות והנחתה, וכמויות רעש משתנות (RNS בטווח 0 עד 03 Bd), ואז אנו מאמנים את הרשת עם MSE או ssol MAE על פני פרמטרים יעדפים.

עטיפת MTSL עם משמעות בזמן: ניתן ליישם MTSL שמעבד את רצף הדוגמיות הגלמות ללא לא TFF מראש, וישירות חוזר את התדר והקבוע. כך הרשת למדה תאימות זמניות ארוכות טווח שעוזרות לסינון רעש.

5.3 הערכת איכות ומדדי ביצועים

הערכה כמותית של מודלי חילוץ אותות חיונית להשוואה הוגנת בין שיטות ולזיהוי רגרסיות במהלך כיוונון היפרפרמטרים.

5.3.1 יחס אות לרעש (RNS)

יחס אות לרעש מוגדר כ-

$$\text{SNR} = 10\log_{10}\frac{\sum_n |s(n)|^2}{\sum_n |e(n)|^2}, \tag{5.2}$$

כאשר $s(n)$ היא האות הנקי (הגל הסינוסואידלי) ו-$e(n)$ היא השגיאה (הרעש או שארית החילוץ). RNS גבוה מציין חילוץ טוב יותר. שיטות מודרניות אמורות להשיג RNS פלט בטווח 01–03 Bd גם כאשר RNS הקלט הוא נמוך (−01 עד 0 Bd).

5.3.2 שגיאה ממוצעת מרובעת מנורמלת (ESMN)

$$\text{NMSE} = \frac{\sum_n (s_{\text{true}}(n) - s_{\text{est}}(n))^2}{\sum_n |s_{\text{true}}(n)|^2}, \tag{5.3}$$

כאשר s_{true} היא האות הנקי האמיתי ו-s_{est} היא ההערכה של הרשת. ESMN קרוב ל-0 הוא טוב.

5.3.3 קורלציה (ecnerehoC)

קורלציית פירסון בין האות הנקי והאות המחולץ מודדת חוסר אנכה לינארי:

$$\rho = \frac{\mathrm{Cov}(s_{\mathrm{true}}, s_{\mathrm{est}})}{\sigma_{s_{\mathrm{true}}} \sigma_{s_{\mathrm{est}}}}. \tag{5.4}$$

ערך קרוב ל-1 מציין התאמה טובה. בנוסף, ecnerehoc בתדר (עבור דמויות בתדר) מציינת היקף השיוך המונוכרומטי בכל תדר.

5.3.4 שגיאת אומדן תדר ופאז

עבור בעיות אומדן תדר וקבוע, אנו מדדים את שגיאת האומדן המוחלטת או היחסית:

$$\mathrm{Error}_f = |f_{\mathrm{true}} - f_{\mathrm{est}}|, \quad \mathrm{Error}_A = |A_{\mathrm{true}} - A_{\mathrm{est}}|. \tag{5.5}$$

טעות יחסית של פחות מ-40.0% בתדר מסיגה קוונטיזציית TFF ומתקרבת לבחינת rémarC–oaR.

פרק 6

יישומים ועתידות

6.1 יישומים בעולם האמיתי

טכניקות חילוץ אותות סינוסואידליים מרעש באמצעות רשתות נוירונים עמוקות מוצאות יישומים רחבים בתחומים מתנועעים. בתקשורת וברדיו, אפליקציות FM/AM מסורתיות ודיגיטליות תלויות בשחזור של אותות נושאים (carrier waves) מתוך תערובות רעש ותדרים מנוצלים. בשנים האחרונות, ארכיטקטורות LSTM ו-BiLSTM שיפרו את היכולת למחזק (enhance) אותות דיבור וממוצע רעש תחתו רמות SNR שבהן שיטות קלאסיות נכשלות [5].

בעיבוד אותות רפואיים, גלי כדורי בדיקה (ECG) וקרינת המוח (EEG) משודרים דרך הגוף האנושי בתוך רעש ביולוגי עצום — התכווצויות שריריות, רעש התקן, והפרעות חשמליות. מודלי denoising autoencoder המאומנים על סיגנלים נקיים בתיוק מצליחים לחלץ טעמים לבי וקצב פעימה אישי עם הפרעה מינימלית, ובכך משפרים אבחנה קלינית ועוקבות מטופלים.

בסיסמיים ובגיאופיזיקה, גל seismic מרעקות משדרים מליבת כדור הארץ דרך הסלע המשותת מתוך רעש פני קרקע תחתון. מדענים משתמשים בתשבצי LSTM חזקים ודו-כיווניים כדי לזהות דפוסי גלים סיסמיים כגון P-waves ו-S-waves לפי זמן בדיוק תת-שניה, המאפשרת אמדנת עומק וחוזק רעידות בדיוק חדש.

יישומים אקוסטיים — ביטול רעש סביבתי בעיבוד שמע בזמן-אמת, סינתזת דיבור משודר, וניתוח אדם ודולק — מלהיטים לכל אחד תפור מודל עמוק מותאם לתדר דיבור או מוזיקה.

בטלקומוניקציות ברחבי המערכת, מערכות MIMO (ריבוי כניסות, ריבוי יציאות) חייבות לתדלק אותות שידור מרובים מאנטנות בודדות בעלות תדר זהה בערך. רשתות עמוקות הלומדות את כוח הערוץ מעל מספר סמלים רתוקים מחזירות ביצוע יתרון מול פוענוח ליניארי קלאסי בעוצמות נמוכות.

6.2 מגבלות וטעויות נפוצות

על אף ההצלחות הניכרות של רשתות נוירונים עמוקות בחילוץ אותות, קיימות מגבלות קריטיות אותן יש להכיר ולהתחשב בהן בעת פיתוח מערכות בייצור.

ראשית, **eruliaf noitazilareneg** מתרחשת כאשר מודל מאומן על טווח סינוסי מוגבל (למשל, תדרים בין 01 ל-001 zH) מוציא ביצוע גרוע כאשר נחשף לתדרים חדשים בעדנת הבדיקה. זוהי בעיה בסיסית בכל חיזוי דפוס קבוע — הרשת לומדת פרטים מיוחדים של תדר ופאזה בנתוני ההדרכה ואינה יכולה להכליל למכלול רציף של אותות. פיתרון שהוכח: הדרכה עם תדרים רחבים וגבוהים, הרעה דיווחית על נתוני הבדיקה (למשל, הוספת טראנזוזיציה עדינה), ופחות — שימוש בנורמליזציה נתונים.

שנית, **RNS ot gnittifrevo** — מודל הדרכה בעוצמת רעש אחת מסוימת (למשל, SNR = 01 Bd) אך בדיקה בעוצמה שונה (02 Bd או 0 Bd) מניב שגיאה ניתוקה גדולה. הרשת למדה לתפוס את תבנית הרעש המיוחדת לתנאי ההדרכה ולא הרעש הגנרי עצמו. יתר על כן, כאשר תדר הרעש דומה לתדר האות (כלומר, interference במקום רעש white), רשתות רבות משתבשות משום שנתונים אלו

לא מופיעים בהדרכה. תרופה: הרעה מרובה על הדרכה כדי כיסוי טווח SNR רחב, והוספת רעש מפריע לצד רעש גרסיו.

שלישית, **ssenignuh atad** — רשתות LSTM וזוגות מקודדים עמוקים דורשים אלפים או עשרות אלפים של דוגמות הדרכה כדי להגיע לרמות ביצוע גדולות. בתחומים רפואיים או שמש, איסוף נתוני הדרכה באיכות גבוהה הוא קשה וסייע מהיר. מודל מותאם עם חומר הדרכה מעט עלול לשמור רעש כתבנית, וזהו תקלה מודל משמעותית.

רביעית, **ycnetal emit-laer** — BiLSTM מחייבת את הרצף כולו לפני שכל דגימה נקבעת, מה שעושה אותה לא ראויה לחיזוי בזמן אמת. חישוב זרימה-הנתונים דרך רשת עמוקה למשך מיליארד דגימה בשנייה עלול להוביל לעיכוב בשניות, בלתי מקובל בתהליכים בזמן אמת. מודלים סיבתיים (מסנני causal CNN, GRU חד-כיווני) מהירים יותר אך בדרך כלל מניבים ביצוע נמוך יותר.

6.3 כיווני מחקר עתידיים

אף על פי שהובילו רשתות נוירונים עמוקות שיפורים מהותיים בחילוץ אותות סינוסואידליים, נותרו בעיות פתוחות קריטיות אותן חוקרים כיום משתדלים להתמודד איתן.

6.3.1 noitatpadA niamoD-ו gninraeL refsnarT

אחד המקורות הגדולים של כישלון הכללה הוא *tfihs niamod* — התבדלות בין התפלגות בנתוני הדרכה לבין התפלגות בנתוני בדיקה בעולם אמיתי. הגישה של **gninrael refsnart** מכוונת להדרכת מודל על תחום עשיר ענק (למשל, טרנספורמציות סינוסואיד מסונתזות עם מליארדי טעם), ולאחר מכן התאמה מהירה לתחום יעד קטן (למשל, ECG אנושי בעל הפרה ראשונית). ערכות דומיין מתקדמות כגון sequential alignment מעניקות אפשרויות כדי שתחת שיטות למידה-ללא-פיקוח ניתן לחלץ אותות במתחמים חדשים ללא תוויות ענפיות [1].

6.3.2 serutcetihcrA tneiciffE

התקורה החישובית של LSTM וזיכרון-□□ sredoced גורמת להן לבלתי מעשיות בהתקנים מוגבלי כוח (כגון edge devices, IoT). חוקרים פותחים **NNR thgiewthgil** stnairav כגון GRU קטנים, gated linear units, וקונבולוציוני חד-כיווניים מיוצרים שמחזיקים ביצוע קרוב ל-LSTM אך עם חוקי חישוב $O(n)$ בהשוואה ל-$O(n^2)$ עבור sledom desab-attention בסדרות ארוכות [4]. דחיסה מודל דרך noitazitnauq וgninurp מקל גם על התמונה בהתקנים מוגבלים.

6.3.3 (sNNIP) skrowteN larueN demrofnI-scisyhP

רשתות המשלבות ידע פיזיקלי קודם דרך noitaziraluger ssol הראו הבטחה בחילוץ אותות [7]. במקום לאפשר לרשת ללמוד בחופשיות מוחלטת, ניתן להטיל אילוצי חלקי:

$$L_{\text{total}} = L_{\text{data}} + \lambda L_{\text{physics}} \tag{6.1}$$

כאשר L_{physics} אכיף דינמיקה ידועה של מערכת (למשל, המשוואות המתארות התפשטות גלים). זה מקל על קונוורגנציה מהירה יותר וביצוע טוב יותר עם נתונים מעטים.

זה משלים את הסדרה של שש פרקים לתיאור עמוק של חילוץ אותות סינוסואידליים מרעש באמצעות למידה עמוקה ורשתות נוירונים. התקדמויות בעתיד תלויות בשילוב של בדקויות חישוביות, ידע דומיין מתקדם, וטכניקות העברת מודלים חדשות.

נספח א': מילון מונחים

נספח זה מרכז את המונחים המרכזיים המופיעים בעבודה, תוך מתן הגדרות תמציתיות לשימוש עתידי.

Attention Mechanism (מנגנון תשומת לב) תהליך חישובי המאפשר לכל אסימון (token) בנתון קלט לקבוע את מידת הרלוונטיות של יתר האסימונים בייצוג התוצאה. מחושב באמצעות מטריצות שאילתה (Query), מפתח (Key) וערך (Value).

Transformer ארכיטקטורת רשת נוירונית המבוססת אך ורק על מנגנון תשומת לב, ללא רשתות רקורנטיות. הוצגה לראשונה במאמר "Attention Is All You Need" (7102) והפכה לבסיס של רוב מודלי השפה המודרניים.

Large Language Model (LLM) מודל שפה גדול המאומן על כמויות עצומות של טקסט, המסוגל לבצע מגוון רחב של משימות שפה ללא כוונון ייעודי. דוגמאות: GPT-4, Claude, Gemini.

Model Context Protocol (MCP) תקן פתוח המגדיר ממשק אחיד בין מודלי שפה לכלים חיצוניים. מאפשר לסוכן לגלות, לפנות ולהשתמש בכלים (שרתים) בצורה עקבית ובטוחה.

ReAct (חשיבה-פעולה) מסגרת לסוכני LLM המשלבת שלבי חשיבה מפורשים (Reasoning) עם שלבי פעולה (Acting) בלולאה חוזרת. מאפשרת לסוכן לנמק לפני שהוא מבצע קריאה לכלי.

Reflexion שיטה לשיפור עצמי של סוכנים על בסיס משוב מילולי. הסוכן מסכם את כישלונותיו הקודמים ומאחסן אותם בזיכרון, ומשתמש בהם לשיפור ניסיונות עתידיים.

Chain-of-Thought (CoT) טכניקת הנחיה המעודדת את המודל לפרט את שלבי החשיבה הביניים לפני מתן התשובה הסופית, ובכך משפרת ביצועים במשימות מורכבות.

Toolformer מודל שפה שאומן לשלב באופן אוטונומי קריאות API בתוך יצור הטקסט שלו. המודל לומד מתי כדאי לקרוא לכלי ואיך לפרש את תוצאותיו.

AvaTaR מסגרת רב-סוכנית המשתמשת בלמידה מחיזוק להכשרת סוכני כלים. בכל איטרציה, סוכן מתאמן (optimizer) משפר את מדיניות השימוש בכלים של הסוכן הראשי.

DEPS (תכנון מודע-תלויות) שיטת תכנון המגדירה גרף תלויות בין תת-משימות ומבטיחה ביצוע בסדר הנכון, תוך ניהול תלויות נתונים בין שלבים.

BiDi (דו-כיווניות) טכנולוגיה לעיבוד טקסט המשלב כיוונים שונים: ימין-לשמאל (עברית) ושמאל-לימין (אנגלית, נוסחאות). מוטמעת בהפקת המסמך באמצעות חבילות polyglossia ו-luabidi.

Fine-Tuning (כוונון עדין) תהליך אימון נוסף של מודל קיים על מערך נתונים ממוקד, במטרה להתאים את התנהגותו למשימה ספציפית מבלי לאמן מאפס.

Hallucination (הזיה) תופעה שבה מודל שפה מייצר מידע שגוי שנראה סביר, אך אינו מבוסס על נתוני האימון או על המציאות. אחד האתגרים המרכזיים בפריסת LLM בסביבות ייצוריות.

ביבליוגרפיה

[1] Tom B. Brown et al. “Language Models are Few-Shot Learners.” In: *Advances in Neural Information Processing Systems 33 (NeurIPS)*. Dec. 2020.

[2] Timo Schick et al. “Toolformer: Language Models Can Teach Themselves to Use Tools.” In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 36. 2023.

[3] Noah Shinn et al. “Reflexion: Language Agents with Verbal Reinforcement Learning.” In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 36. 2023.

[4] Ashish Vaswani et al. “Attention Is All You Need.” In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[5] Lei Wang et al. “A Survey on Large Language Model based Autonomous Agents.” In: *Frontiers of Computer Science* (2023).

[6] Zhiheng Xi et al. “The Rise and Potential of Large Language Model Based Agents: A Survey.” In: *arXiv preprint arXiv:2309.07864* (2023).

[7] Shunyu Yao et al. “ReAct: Synergizing Reasoning and Acting in Language Models.” In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2023.