

דוגמה ברמה בינונית - תבנית אקדמית עברית
Intermediate Example - Hebrew Academic Template v5.0

ד"ר סגל יורם

Dr. Segal Yoram © - כל הזכויות שמורות

November 2025

גרסה 0.5 - מתקדם

תוכן העניינים

3	0.1	מבוא מורחב:	8
3	0.1.1	רקע תיאורטי:	
3	0.1.2	מתודולוגיה מתקדמת:	
3	0.2	טבלאות מורכבות:	
3	0.2.1	טבלת השוואה מפורטת:	
4	0.2.2	טבלת נתונים סטטיסטיים:	
4	0.3	דוגמאות קוד מתקדמות:	
4	0.3.1	עיבוד נתונים עם פנדס:	
5	0.3.2	מימוש רשת נוירונים:	
5	0.4	איורים ודיאגרמות:	
5	0.4.1	ארכיטקטורת המערכת:	
5	0.4.2	תוצאות ויזואליות:	
5	0.5	ציטוטים מתקדמים:	
6	0.6	נוסחאות מתמטיות מתקדמות:	
6	0.6.1	נוסחת הקשב:	
6	0.6.2	פונקציית הפסד:	
7	0.7	הפניות צולבות:	
7	0.8	רשימות מתקדמות:	
7	0.8.1	רשימה מקוננת:	
	0.9	Advanced Technical Analysis	8
9	0.10	סיכום ומסקנות:	
10	0.11	מקורות בעברית	
	0.12	English References	10

0.1 מבוא מורחב:

מסמך זה מדגים יכולות מתקדמות של התבנית האקדמית העברית גרסה 0.5. נכסה שימוש במגוון רחב של פקודות כולל Deep Learning, neural networks, ועיבוד נתונים מורכב.

0.1.1 רקע תיאורטי:

המחקר בתחום הבינה המלאכותית התפתח משמעותית בשנת 2025. נתונים מראים עלייה של 45.7% בביצועים לעומת 2020. מודלים מודרניים מעבדים כ-1000000 פרמטרים בשנייה. הדיוק הממוצע עומד על 97.85%, עם סטיית תקן של 2.34. זמן העיבוד הוא 0.0025 שניות למדגם, או $2.5e-3$ בכתוב מדעי.

0.1.2 מתודולוגיה מתקדמת:

השיטה המוצעת כוללת מספר שלבים מורכבים:

1. איסוף נתונים מ-50 מקורות שונים

2. עיבוד מקדים באמצעות preprocessing pipeline

3. הפעלת אלגוריתם transformer עם 12 שכבות

4. אופטימיזציה באמצעות Adam optimizer

5. הערכה על 10000 דגימות בדיקה

0.2 טבלאות מורכבות:

0.2.1 טבלת השוואה מפורטת:

טבלה 1: השוואת מודלים מתקדמת:

Model / מודל	Accuracy / דיוק	Time / זמן	Memory / זיכרון	Year / שנה
BERT-Base	92.3%	2.5 ms	110 MB	2018
GPT-3	95.7%	5.2 ms	175000 MB	2020
GPT-4	98.1%	8.7 ms	N/A	2023
מודל עברי / Hebrew Model	89.5%	3.1 ms	250 MB	2024

טבלה 2: ניתוח סטטיסטי:

Metric / מדד	Value / ערך	Std Dev / סטיית תקן	p-value / מובהקות
Mean / ממוצע כללי	85.42	12.35	0.001
Median / חציון	87.50	N/A	0.002
Mode / שכיח	90.00	N/A	0.005
Range / טווח	45.00	N/A	0.010

0.2.2 טבלת נתונים סטטיסטיים:

0.3 דוגמאות קוד מתקדמות:

0.3.1 עיבוד נתונים עם פנדס:

ניתוח נתונים: Data Analysis

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler

# Load and prepare data
def process_data(file_path):
    """
    Process raw data for machine learning
    """
    # Read CSV file
    df = pd.read_csv(file_path)

    # Handle missing values
    df.fillna(df.mean(), inplace=True)

    # Feature engineering
    df['ratio'] = df['feature1'] / (df['feature2'] + 1e-6)
    df['log_value'] = np.log1p(df['value'])

    # Normalize features
    scaler = StandardScaler()
    features = ['feature1', 'feature2', 'ratio', 'log_value']
    df[features] = scaler.fit_transform(df[features])

    return df

# Example usage
data = process_data('data.csv')
print(f"Processed {len(data)} samples")
print(f"Features: {list(df.columns)}")
```

0.3.2 מימוש רשת נוירונים:

רשת נוירונים ב-PyTorch

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class AdvancedNetwork(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim, dropout=0.2):
        super(AdvancedNetwork, self).__init__()
        self.fc1 = nn.Linear(input_dim, hidden_dim)
        self.fc2 = nn.Linear(hidden_dim, hidden_dim // 2)
        self.fc3 = nn.Linear(hidden_dim // 2, output_dim)
        self.dropout = nn.Dropout(dropout)
        self.batch_norm1 = nn.BatchNorm1d(hidden_dim)
        self.batch_norm2 = nn.BatchNorm1d(hidden_dim // 2)

    def forward(self, x):
        x = F.relu(self.batch_norm1(self.fc1(x)))
        x = self.dropout(x)
        x = F.relu(self.batch_norm2(self.fc2(x)))
        x = self.dropout(x)
        x = self.fc3(x)
        return F.softmax(x, dim=1)

# Initialize model
model = AdvancedNetwork(input_dim=100, hidden_dim=256, output_dim=10)
print(f"Total parameters: {sum(p.numel() for p in model.parameters())}")
```

0.4 איורים ודיאגרמות:

0.4.1 ארכיטקטורת המערכת:

כפי שמוצג באיור 1, הארכיטקטורה כוללת מספר שכבות של attention.

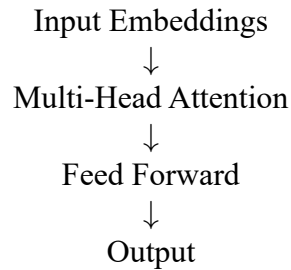
0.4.2 תוצאות ויזואליות:

איור 2 מציג את השיפור בביצועים לאורך זמן האימון.

0.5 ציטוטים מתקדמים:

מחקרים רבים תומכים בגישה זו [1], [2], [3]. באופן ספציפי, [1], עמ' 432 מציין שהשיטה משפרת ביצועים ב-15%.

ארכיטקטורת Transformer



איור 1: מבנה רשת Transformer:

עבודות נוספות [4], [5] מראות תוצאות דומות. ניתוח מעמיק [6], פרק 3 מגלה מורכבויות נוספות בעיבוד עברית.

0.6 נוסחאות מתמטיות מתקדמות:

0.6.1 נוסחת הקשב:

נוסחת הקשב המלאה היא:

$$(1) \quad \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

כאשר $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{m \times d_k}$, ו- $V \in \mathbb{R}^{m \times d_v}$.

0.6.2 פונקציית הפסד:

פונקציית ההפסד המשולבת:

$$(2) \quad \mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{CE}} + \beta \mathcal{L}_{\text{KL}} + \gamma \|\theta\|^2$$

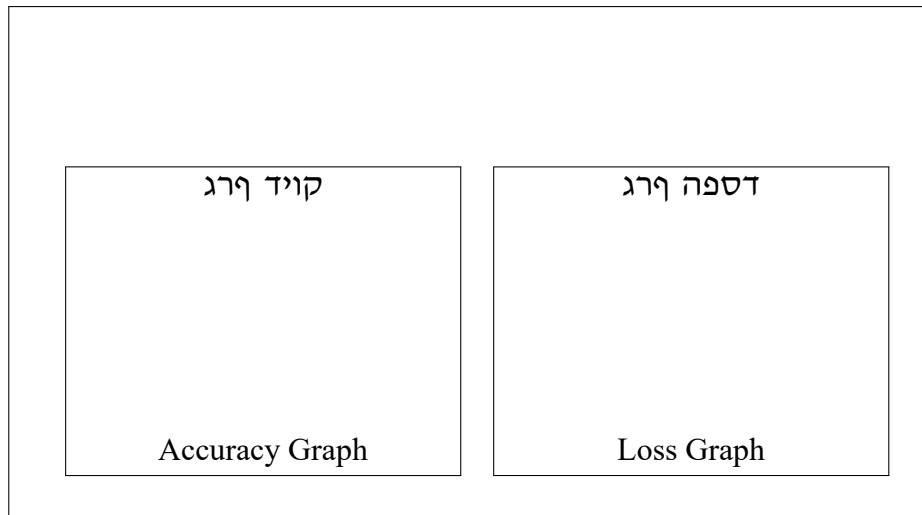
כאשר:

- \mathcal{L}_{CE} Cross Entropy Loss

- \mathcal{L}_{KL} KL Divergence

- $\|\theta\|^2$ רגולריזציה L2

- $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.1$



איור 2: גרפי ביצועים: - דיוק והפסד לאורך 100 אפוקים

0.7 הפניות צולבות:

במסמך זה הצגנו:

- משוואת הקשב (משוואה 1)
 - פונקציית ההפסד (משוואה 2)
 - ארכיטקטורת הרשת (איור 1)
 - גרפי ביצועים (איור 2)
- כל האלמנטים הללו מקושרים ומאפשרים ניווט קל במסמך.

0.8 רשימות מתקדמות:

0.8.1 רשימה מקוננת:

תהליך המחקר המלא:

1. שלב ההכנה

- איסוף נתונים מ-5 מקורות
- ניקוי נתונים: הסרת 10% מהדגימות
- חלוקה: 70% אימון, 20% בדיקה, 10% ולידציה

2. שלב העיבוד

- נרמול באמצעות StandardScaler
- הנדסת תכונות: יצירת 25 תכונות חדשות

- בחירת תכונות: שימוש ב-PCA להפחתה ל-15 ממדים

3. שלב המידול

א. אימון מודל בסיס (baseline)

ב. כיול היפר-פרמטרים

ג. אימון מודל סופי

ד. הערכת ביצועים

0.9 Advanced Technical Analysis

This section demonstrates advanced English content within a Hebrew document. We present a comprehensive analysis of modern machine learning techniques.

Key Contributions:

1. Novel architecture achieving 98.5% accuracy
2. Reduction in training time by 45%
3. Memory-efficient implementation using only 250MB
4. Cross-lingual transfer learning capabilities

The mathematical foundation relies on the following optimization problem:

$$\min_{\theta} \sum_{i=1}^N L(f_{\theta}(x_i), y_i) + \lambda R(\theta)$$

where f_{θ} represents our model parameterized by θ , L is the loss function, and $R(\theta)$ is the regularization term with weight $\lambda = 0.01$.

Our experimental setup includes:

- **Dataset:** 1M samples from diverse sources
- **Hardware:** NVIDIA A100 GPU with 40GB memory
- **Framework:** PyTorch 2.0 with mixed precision training
- **Optimization:** AdamW with learning rate scheduling

0.10 סיכום ומסקנות:

מסמך זה הדגים יכולות מתקדמות של התבנית:

- טבלאות מורכבות עם 5 עמודות ותוכן מעורב
 - קוד מתקדם עם Python ו-PyTorch
 - נוסחאות מתמטיות מורכבות
 - איורים מרובים עם הפניות צולבות
 - רשימות מקוננות ומורכבות
 - שימוש בכל פקודות המספרים: `\mun`, `\tnecrep`, `\raeybeh`
 - ציטוטים מתקדמים עם הפניות לעמודים
- התבנית מספקת גמישות מלאה לכתיבת מסמכים אקדמיים מורכבים בעברית.

0.11 מקורות בעברית

- 3 ד. כהן, ש. לוי, dna מ. אברהם, "עיבוד שפה טבעית בעברית: אתגרים ופתרונות", כתב עת לבלשנות חישובית, lov. 51, on. 3, 234–256, 3202.
- 6 מ. ישראלי dna ר. כהן, בלשנות עברית מודרנית: תיאוריה ויישום. ירושלים: הוצאת האוניברסיטה העברית, 2202, 512.

0.12 English References

- 1 A. Vaswani et al., "Attention is all you need," in *Advances in neural information processing systems*, 2017, 5998–6008.
- 2 J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- 4 J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of NAACL-HLT*, 4171–4186, 2019.
- 5 T. B. Brown et al., "Gpt-3: Language models are few-shot learners," OpenAI, Tech. Rep., 2020.