

# Deep Learning Final Project

---

**BBC NEWS - CATEGORY CLASSIFICATION**

NOA HONIGSTEIN & MATANEL LEVAVI

## הגדרת הבעיה והיעד

---

- **המשימה:** סיווג אוטומטי של כתבות חדשות ל-5 קטגוריות: עסקים, טכנולוגיה, פוליטיקה, ספורט ובידור.
- **מאגר הנתונים:** אוסף BBC News הכולל 2,225 מסמכי טקסט.
- **מטרה:** השוואה בין אסטרטגיות סיווג שונות – החל מניחוש סטטיסטי ועד לרשתות עצביות עמוקות.

# מודלי בסיס (BASELINES) - קביעת רף תחתון

---

## 1. MAJORITY CLASS BASELINE:

ניבוי הקטגוריה השכיחה ביותר ("ספורט") עבור כל כתבה.

**תוצאות הדיוק (ACCURACY): 24.27%**

## 2. DICTIONARY-BASED BASELINE:

בניית "מילון חכם" לכל קטגוריה בעזרת 20 מילות המפתח המובילות בTFIDF.

**תוצאות הדיוק (ACCURACY): 88.31%**

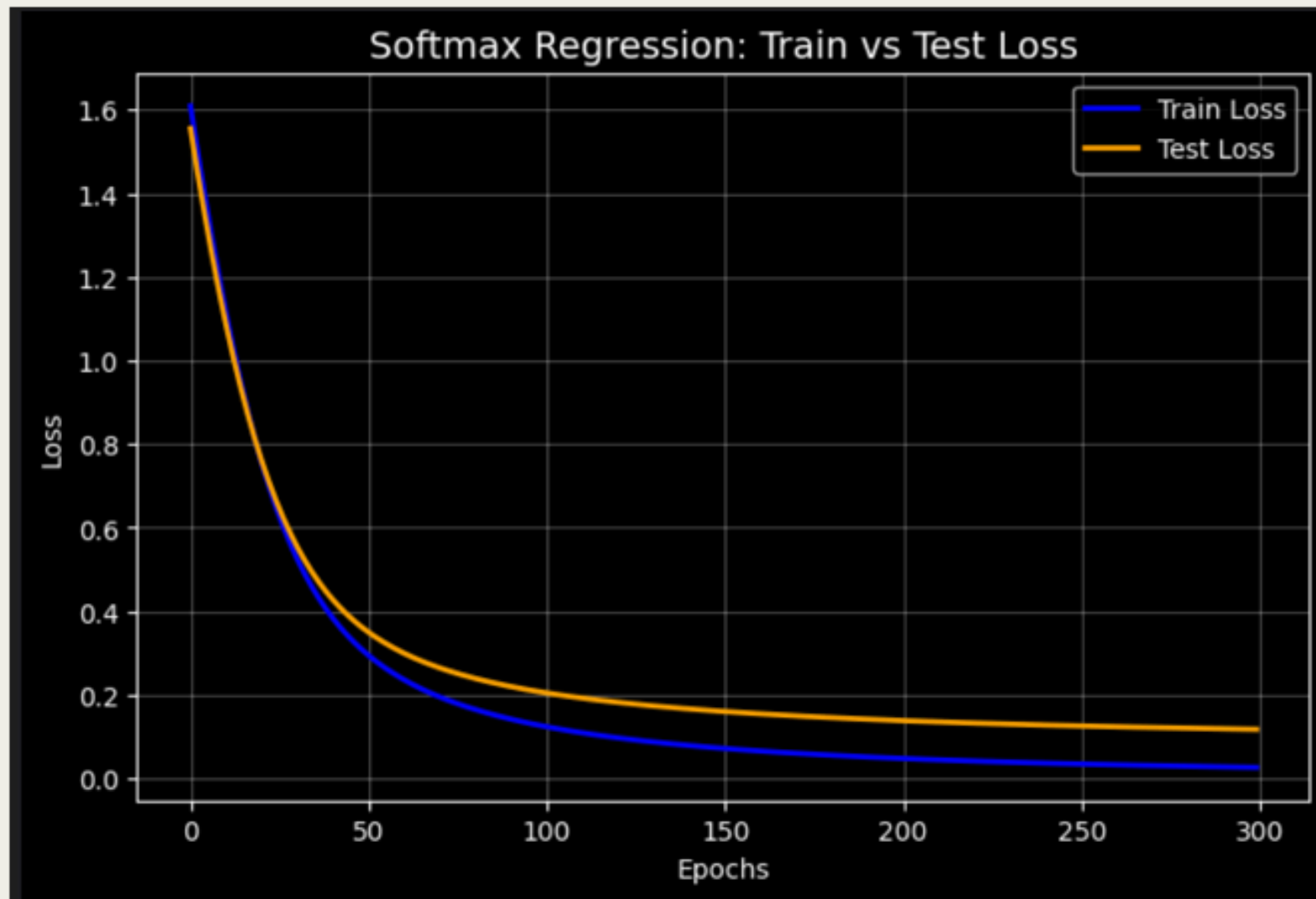
```
CLASS SOFTMAXCLASSIFIER(NN.MODULE):
    DEF __INIT__(SELF, INPUT_DIM, OUTPUT_DIM):
        SUPER(SOFTMAXCLASSIFIER, SELF).__INIT__()
        # SINGLE LINEAR LAYER (SOFTMAX/LOGISTIC REGRESSION)
        SELF.LINEAR = NN.LINEAR(INPUT_DIM, OUTPUT_DIM)

    DEF FORWARD(SELF, X):
        # WE RETURN LOGITS (RAW SCORES).
        # NOTE: CROSSENTROPYLOSS IN PYTORCH APPLIES SOFTMAX
AUTOMATICALLY.
        RETURN SELF.LINEAR(X)
```

## SOFTMAX REGRESSION

# כוונון מאפיינים - 500 לעומת 2,000 פיצ'רים

MAX\_FEATURE = 2000



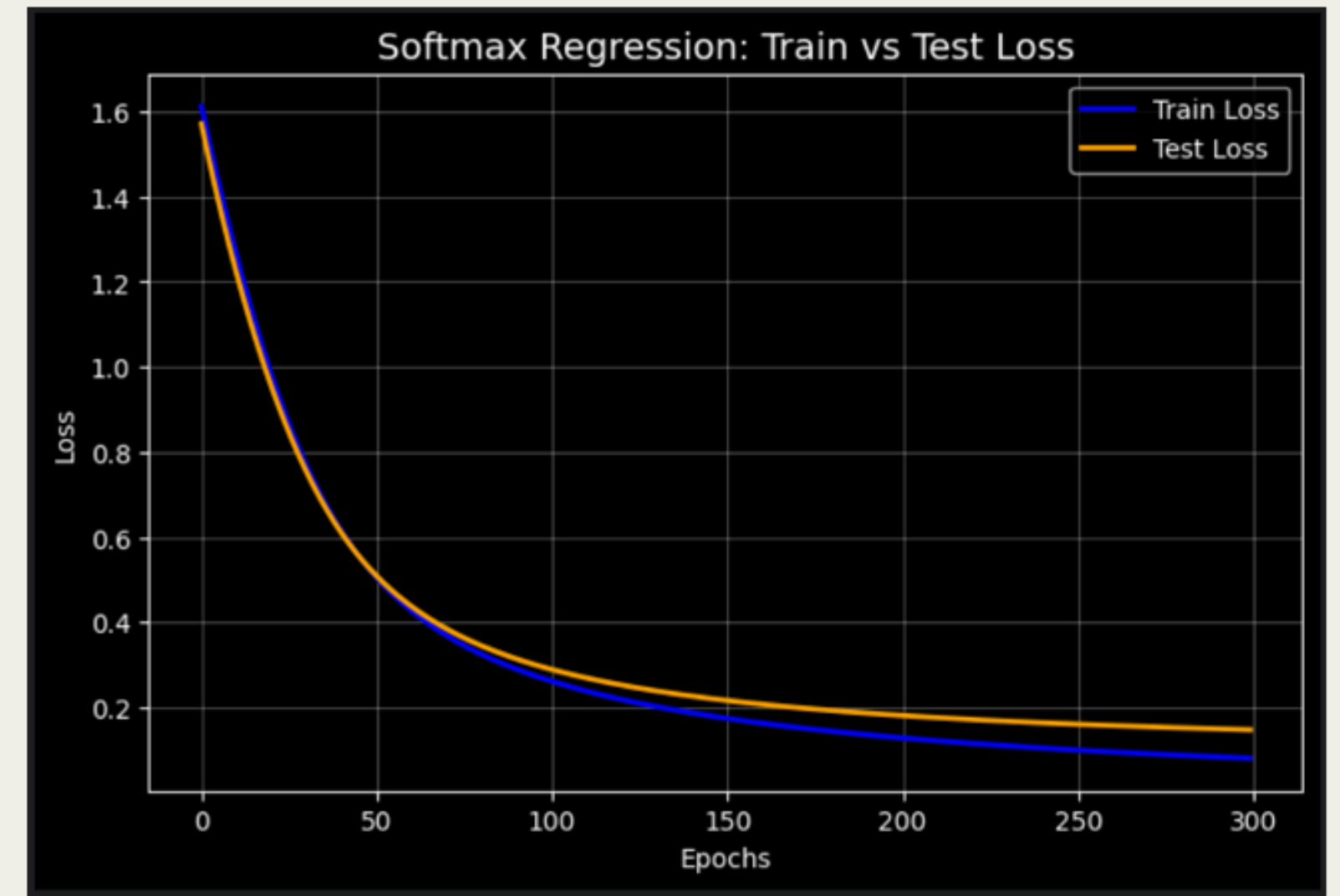
FINAL MODEL RESULTS:

Accuracy: 0.9775 (97.75%)

Precision: 0.9776

Recall: 0.9775

MAX\_FEATURE = 500



FINAL MODEL RESULTS:

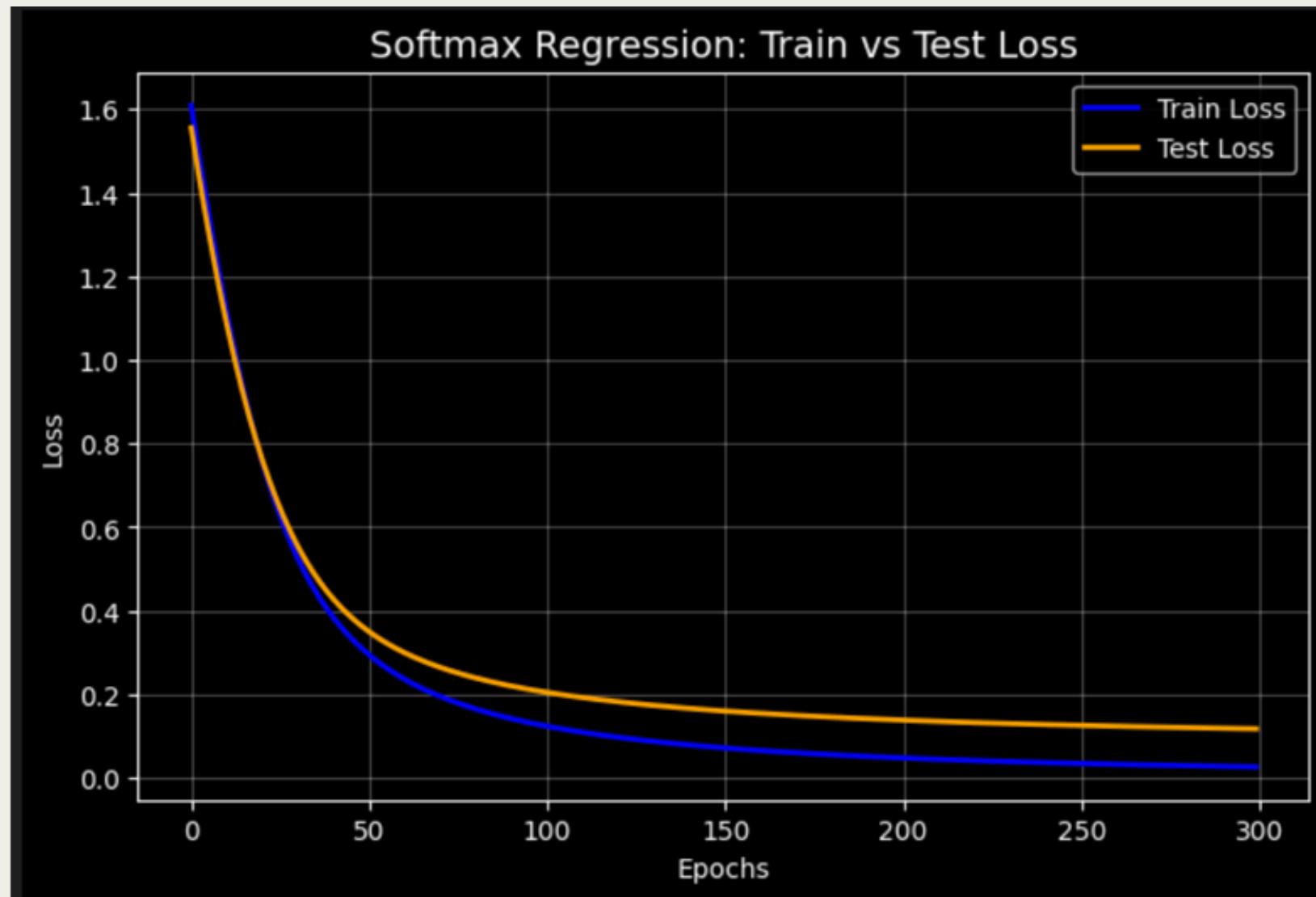
Accuracy: 0.9640 (96.40%)

Precision: 0.9641

Recall: 0.9640

# השפעת קצב הלמידה (LEARNING RATE)

**LR=0.01**



**LR=0.001**



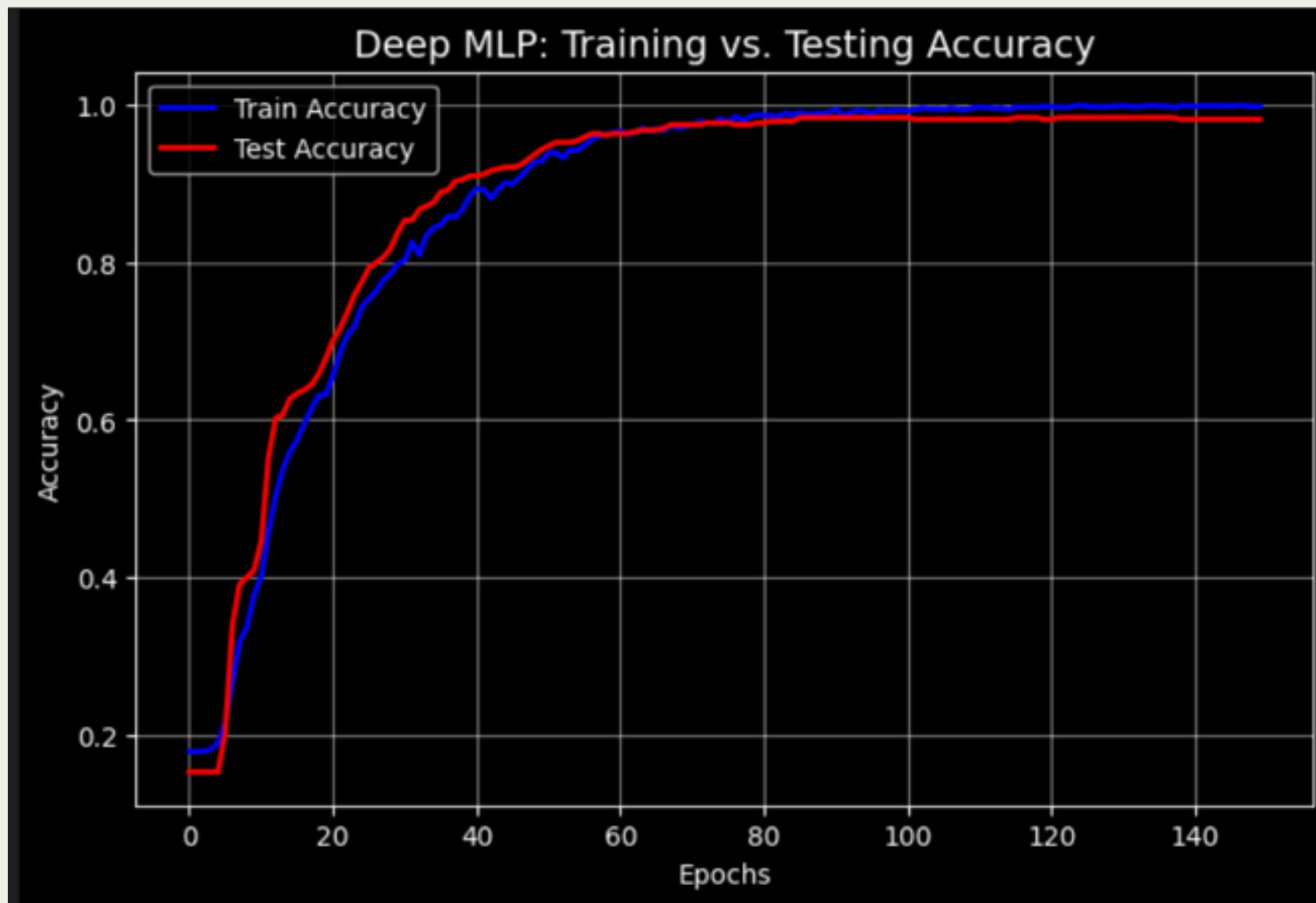
```
CLASS MLPCLASSIFIER(NN.MODULE):
    DEF __INIT__(SELF, INPUT_DIM, OUTPUT_DIM):
        SUPER(MLPCLASSIFIER, SELF).__INIT__()
        # FIRST HIDDEN LAYER: INPUT -> 100 NEURONS
        SELF.HIDDEN1 = NN.LINEAR(INPUT_DIM, 100)
        # SECOND HIDDEN LAYER: 100 -> 50 NEURONS
        SELF.HIDDEN2 = NN.LINEAR(100, 50)

        SELF.DROPOUT = NN.DROPOUT(0.3)
        # OUTPUT LAYER: 50 -> CATEGORIES (5)
        SELF.LINEAR = NN.LINEAR(50, OUTPUT_DIM)

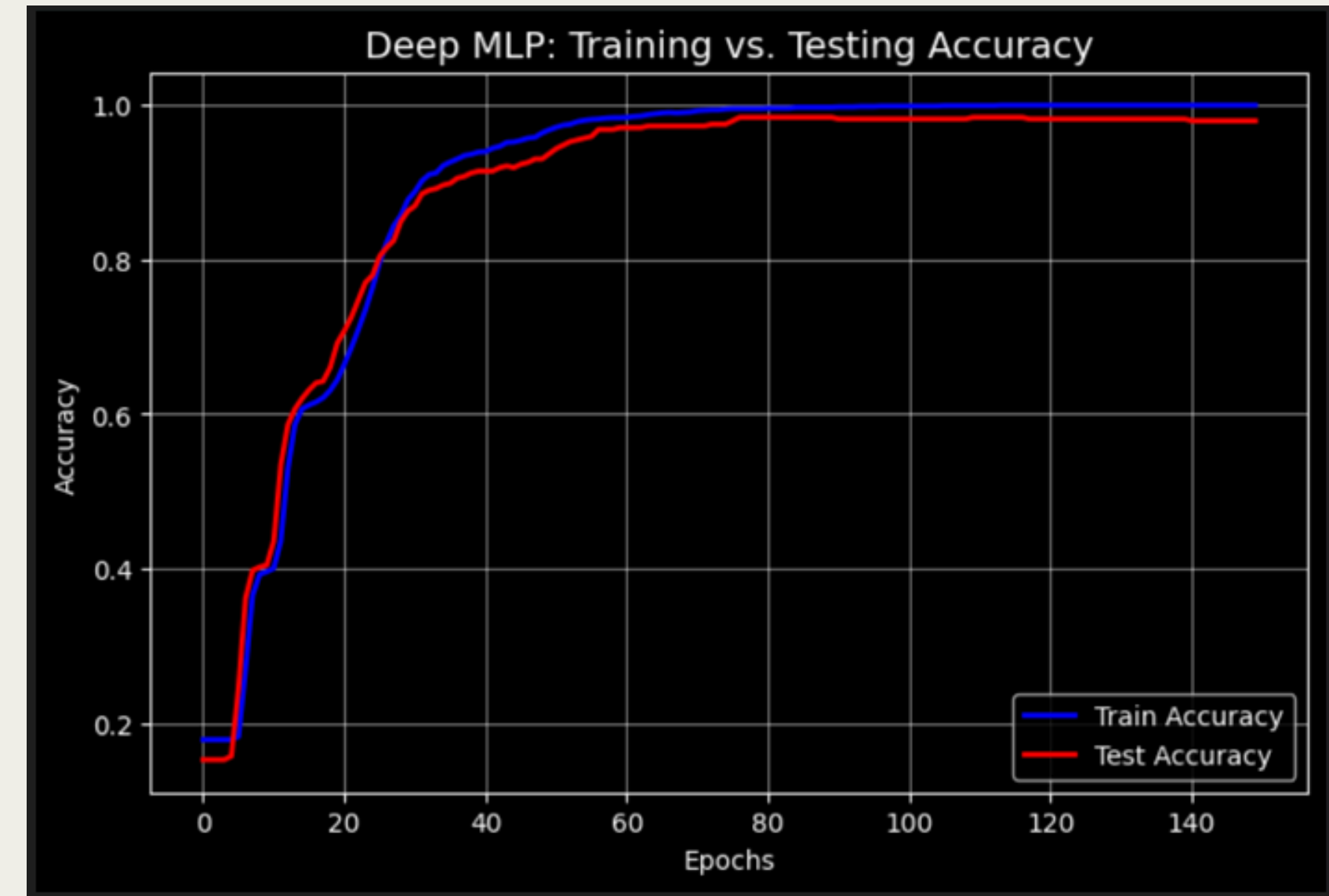
    DEF FORWARD(SELF, X):
        X = TORCH.RELU(SELF.HIDDEN1(X))
        X = SELF.DROPOUT(X)
        X = TORCH.RELU(SELF.HIDDEN2(X))
        X = SELF.DROPOUT(X)
        RETURN SELF.LINEAR(X)
```

## MULTI-LAYER PERCEPTRON

# מודל עם 2 שכבות נסתרות : עם רגולריזציה VS בלי רגולריזציה



עם DROPOUT



בלי DROPOUT



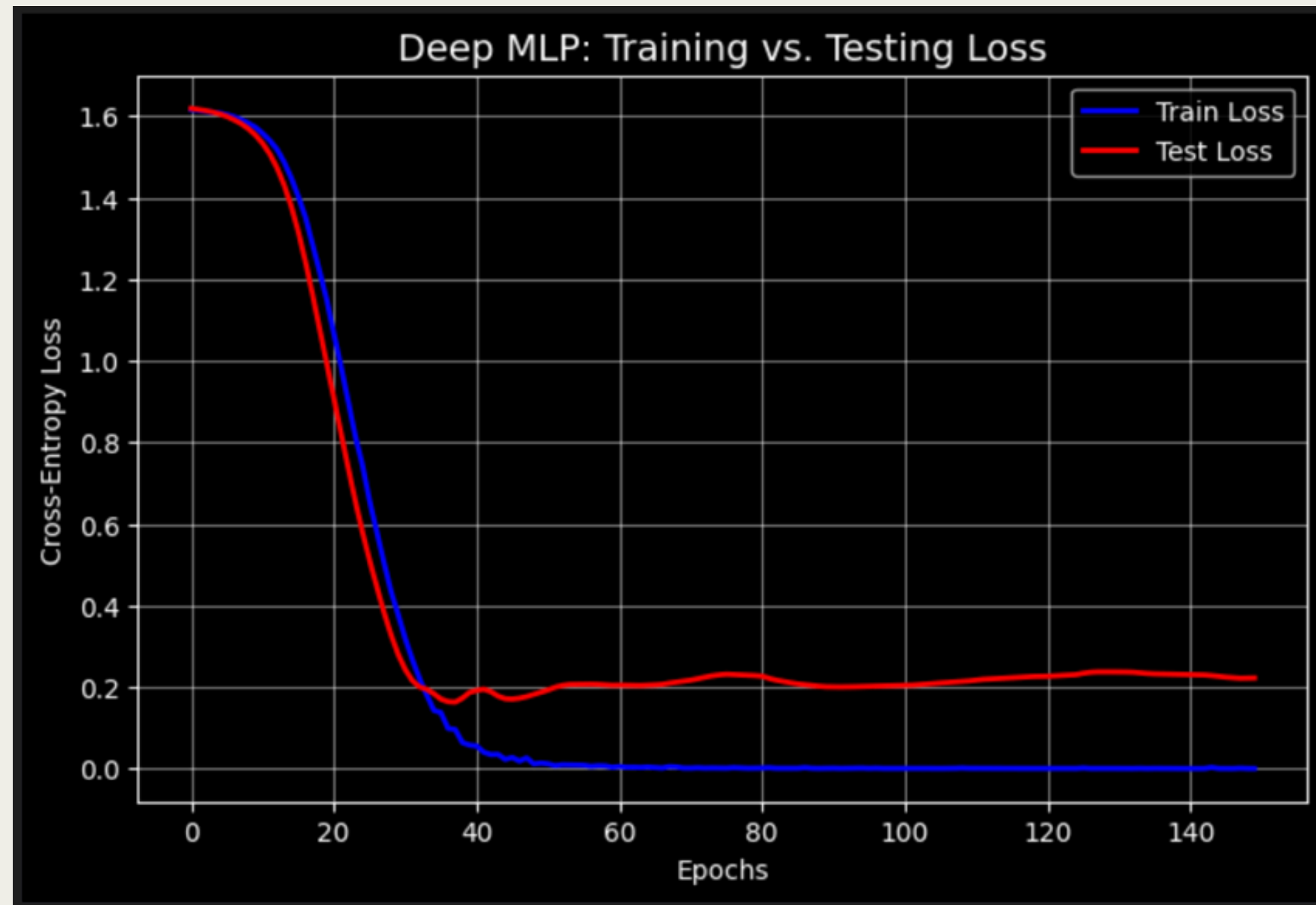
# DROPOUT עם



# DROPOUT בלי



# מחיר המורכבות - 4 שכבות נסתרות



Num of hidden layers	Dropout(0.3)	Accuracy	Precision	Recall
2	with	0.9820	0.9821	0.9820
4	with	0.9640	0.9641	0.9640
2	without	0.9798	0.9799	0.9798
4	without	0.9596	0.9598	0.9596

השוואה סופית וסיכום ביצועים

Model	Accuracy	Precision	Recall
Simple Baseline (Majority)	24.27%	0.0589	0.2427
Rule-Based (Dictionary)	88.3%	0.8862	0.8831
Softmax Regression	97.75%	0.9776	0.9775
Deep MLP (Best Model)	98.20%	0.9821	0.9820