

Deep Learning

Computer vision

Part II

מונע בעיית זיגזוג בדרך למינימום
המקומי.

יעיל כאשר חלק מהצירים לומדים
בצורה יציבה וחלק לא.

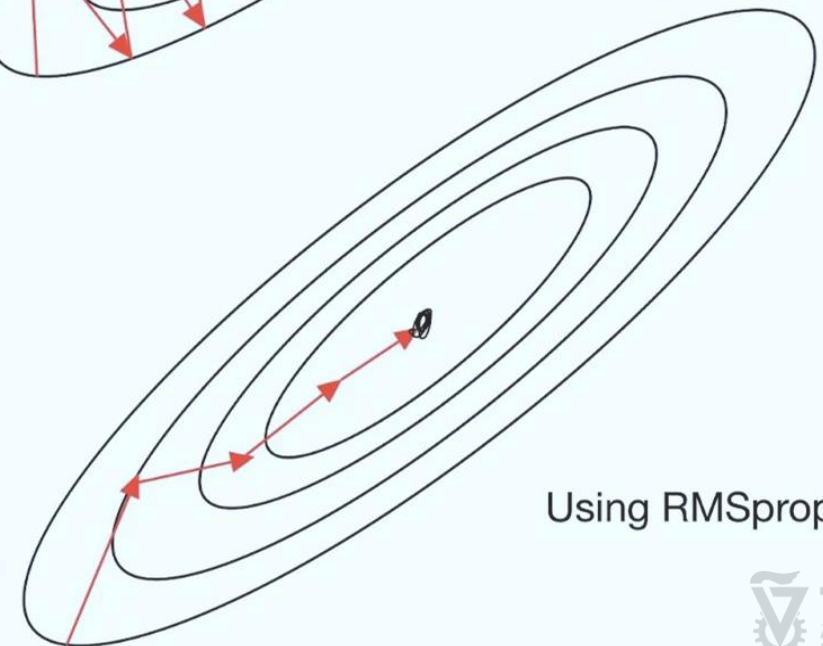
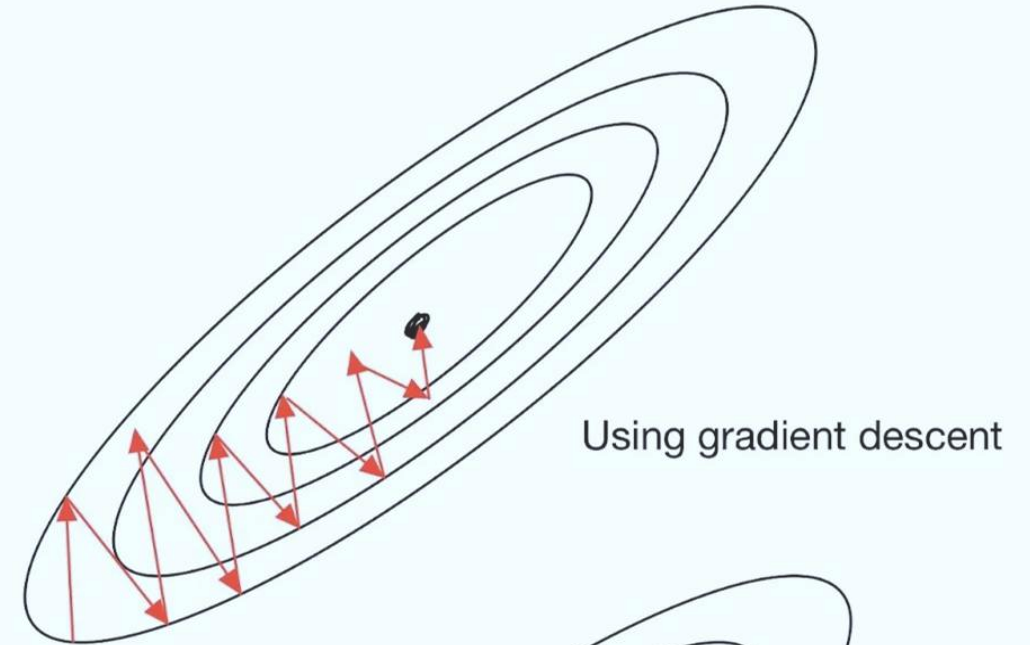
רלונטי ב CNN הנוטה לתנודתיות,

משום שהפילטרים נוטים לחפש דפוסים

חדשים (בעיקר בתחילת האימון)

שורה תחתונה:

בוחר קצב לימוד שונה לכל פרמטר.



ממוצע נע של ריבוע הגרדיאנט (Exponential Moving Average of Squared Gradients) הוא טכניקה המשמשת באופטימיזציה, כמו ב-RMSprop ו-Adam כדי להתאים את קצב הלמידה באופן דינמי.

- איך זה עובד?

מחשבים ממוצע משוקלל של הריבוע של הגרדיאנט בכל צעד.

מדד זה עוזר לזהות אם הגרדיאנט **תנודתי** או **עקבי**.

- אם השיפוע גדול לאורך זמן \leftarrow המודל יקטין את קצב הלמידה.

אם השיפוע קטן או יציב \leftarrow המודל ישמור על קצב למידה גבוה יותר.

התוצאה: יציבות טובה יותר במהלך האימון ומניעת קפיצות חדות בגרדיאנט.

$$W_{new} = W_{old} - \alpha \cdot \frac{dw}{\sqrt{v_{dw} + \epsilon}}$$

W_{new} → משקל חדש
 W_{old} → משקל ישן
 α → מספר מאוד קטן 10^{-8}
 dw → נגזרת
 v_{dw} → ממוצע נע של ריבוע הגרדיאנט
 ϵ → מספר מאוד קטן 10^{-8}

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2$$

β → מקדם בד"כ 0.9999
 v_{dw} → ממוצע קודם
 dw^2 → נגזרת

Batch Normalization

Batch Normalization היא שיטה לייצוב האימון של רשתות נוירונים על ידי נרמול הערכים בכל שכבה. היא מפחיתה את התנודתיות של הנתונים בין השכבות, מה שמוביל לאימון יציב ומהיר יותר.

איך זה עובד?

1. בכל מיני-באטץ' במהלך האימון, מחושבים הממוצע והסטיית התקן של האקטיבציות.
2. הערכים מנורמלים כך שתהיה להם התפלגות בעלת ממוצע 0 ושונות 1.
3. הרשת לומדת שני פרמטרים נוספים **Scale**, **Shift** – המאפשרים למודל לשמור על ייצוגים רלוונטיים.

Batch Normalization

יתרונות:

- מאיץ את קצב הלמידה ומקטין את התלות באתחול המשקלים.
- מאפשר להשתמש בערכים גדולים יותר של קצב הלמידה.
- מפחית את הצורך בשכבות כמו Dropout למניעת Overfitting.
- עוזר לרשתות עמוקות להתאמן בצורה יציבה יותר.

Batch Normalization יכול להיות משולב לאחר שכבות Conv2D או Dense, ומומלץ למקם אותו לפני פונקציות האקטיבציה כמו ReLU.



Pretrained Models

מודלים מאומנים מראש (Pretrained Models) הם רשתות נוירונים שנלמדו על סטים גדולים של נתונים, כמו **ImageNet**, ומספקים יכולות מוכנות לזיהוי דפוסים.

יתרונות:

- חוסכים זמן ומשאבים בכך שמתחילים מאימון קודם במקום מאפס.
 - מאפשרים שימוש בידע כללי שנרכש מנתונים גדולים, גם עבור בעיות חדשות.
 - משמשים לביצוע **Transfer Learning** שבו מותאמים המודלים למשימה חדשה בעזרת Fine-Tuning או Feature Extraction.
- ## דוגמאות נפוצות:

- **ResNet, VGG, EfficientNet** – לזיהוי תמונות.
- **BERT, GPT** – לעיבוד שפה טבעית.
- **YOLO, Faster R-CNN** – לזיהוי עצמים בתמונות.

מודלים מאומנים מראש נפוצים בעיקר ב Computer Vision ו NLP – והם משמשים רבות בתעשייה כדי להשיג ביצועים טובים במהירות וביעילות.



Pretrained Models

```
model_url = 'some_url...'
feature_extractor_layer = hub.KerasLayer(model_url, trainable=False, input_shape=(224, 224, 3))
num_classes = 10

resnet_model = tf.keras.Sequential([
    feature_extractor_layer,
    tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

→ קוד קצר

Layer (type)	Output Shape	Param #
keras_layer (KerasLayer)	(None, 2048)	23561152
dense (Dense)	(None, 10)	20490

=====
Total params: 23581642 (89.96 MB)

Trainable params: 20490 (80.04 KB)

Non-trainable params: 23561152 (89.88 MB)

→ פרמטרים מאומנים מראש