

למידה עמוקה תשפ"ג - תרגיל 3

רשתות נוירונים קונבולוציוניות CNNs עיבוד טקסט ורציפים

שימו לב: משימות להגשה מודגשות בצבע חום.

מבנה שכבות ב CNN

הסתכלו במחברת **5.1-introduction-to-convnets.ipynb**,
ובפרק 5.1 בספר של Francois Chollet: **5.1. introduction to convnets.**

המודל במחברת נבנה בעזרת הקוד הזה

```
model = models.Sequential()  
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))  
model.add(layers.MaxPooling2D((2, 2)))  
model.add(layers.Conv2D(64, (3, 3), activation='relu'))  
model.add(layers.MaxPooling2D((2, 2)))  
model.add(layers.Conv2D(64, (3, 3), activation='relu'))  
model.add(layers.Flatten())  
model.add(layers.Dense(64, activation='relu'))  
model.add(layers.Dense(10, activation='softmax'))
```

התקציר של המודל (`model.summary()`) נראה כך

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_2 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
conv2d_3 (Conv2D)	(None, 3, 3, 64)	36928
flatten_1 (Flatten)	(None, 576)	0
dense_1 (Dense)	(None, 64)	36928
dense_2 (Dense)	(None, 10)	650
Total params: 93,322		
Trainable params: 93,322		
Non-trainable params: 0		

ממד הקלט

כמו במחברת הראשונה שראינו **2.1-a-first-look-at-a-neural-network.ipynb** גם במחברת הנוכחית משתמשים במודל mnist, אך הפעם מתבצעת פעולת reshape שונה לאחר קריאת הנתונים

```
train_images = train_images.reshape((60000, 28, 28, 1))
```

להגשה:

1. מה ה shape של train_images לפני ה reshape?

2. מה ה shape של הקלט לשכבה הראשונה של הרשת?

3. לכל אחת משכבות הרשת הסבירו: מה הקלט לשכבה, מה הפעולה שהשכבה עושה, מה הפלט, מה מספר הפרמטרים וכיצד מחשבים אותו לשכבה זו. השתמשו בנוסח הזה (השלימו כמובן את הפרטים שהחסרתי):

השכבה: conv2d_1

הקלט: תמונה המיוצגת במערך בגודל (הכניסו כאן את ה shape המתאים).
פעולה: קונבולוציה של הקלט עם פילטר (kernel) בגודל (השלימו). הפלט של הקונבולוציה לתמונה בודדת הוא מערך בגודל (השלימו). אחרי ביצוע קונבולוציה מופעלת האקטיבציה relu על התוצאה. יש 32 פילטרים, על כן מקבלים 32 תוצאות.

הפלט: מערך של תוצאות הקונבולוציות בגודל (השלימו).
מספר הפרמטרים: 320. המספר מחושב כך: 32 פילטרים כפול 3x3 (גודל כל פילטר) ועוד 32 ערכי bias של פעולת ה relu המתבצעת אחרי הקונבולוציה.

כדי להבין את הפרטים ולהסבירם כראוי עליכם לקרוא את תת הפרק בספר.

קראו גם את הפוסטים של *Jason Brownlee*:

How Do Convolutional Layers Work in Deep Learning Neural Networks?

<https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>

Crash Course in Convolutional Neural Networks for Machine Learning

<https://machinelearningmastery.com/crash-course-convolutional-neural-networks/>

הרחבה:

להסבר מפורט על הפרמטרים השונים של שכבת Conv2D קראו כאן

Keras Conv2D and Convolutional Layers

<https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/>

שאלות לבירור על עיבוד טקסט ורצפים מתוך פרק 6

4. המרה של טקסט לייצוג מספרי

א. מה היתרונות והחסרונות של one-hot word vectors לעומת Word embedding?

ב. מתי עדיף להשתמש בכל שיטה?

ג. השימוש ב word embedding: אפשר ללמוד את ה embedding תוך כדי המשימה (שהרשת שלנו מבצעת), או לטעון ולהשתמש ב word embedding שנלמד קודם לכן למשימה \ נתונים אחרים. מה היתרונות והחסרונות לכל אפשרות?

5. RNN

א. הסתכלו על ה block הבסיסי של RNN באיור 6.9. כמה פעמים מופעל ה block הזה ב forward pass של RNN ?

ב. בפסקה 6.2.1 (סוף עמוד 198 ותחילת עמוד 199) מתוארת רשת ולה סיכום זה

Layer (type)	Output Shape	Param #
embedding_22 (Embedding)	(None, None, 32)	320000
simplernn_10 (SimpleRNN)	(None, 32)	2080
Total params: 322,080		
Trainable params: 322,080		
Non-trainable params: 0		

הסבירו את חישוב מספר הפרמטרים של כל שכבה.

6. LSTM

א. על פי ההסבר בספר, מה תפקיד של הפעולות המתבצעות כאן

Listing 6.25 Pseudocode details of the LSTM architecture (1/2)

```
output_t = activation(dot(state_t, Uo) + dot(input_t, Wo) + dot(C_t, Vo) + bo)
i_t = activation(dot(state_t, Ui) + dot(input_t, Wi) + bi)
f_t = activation(dot(state_t, Uf) + dot(input_t, Wf) + bf)
k_t = activation(dot(state_t, Uk) + dot(input_t, Wk) + bk)
```

You obtain the new carry state (the next c_t) by combining i_t , f_t , and k_t .

Listing 6.26 Pseudocode details of the LSTM architecture (2/2)

```
c_{t+1} = i_t * k_t + c_t * f_t
```

ב. בהשוואה בין simple RNN ל LSTM בהרצות על IMDB, איזו ארכיטקטורה משיגה תוצאות טובות יותר?

7. 1D convnet

א. מה היתרונות ומה החסרונות בהפעלת 1D convnet לעיבוד תבניות עיתיות (temporal patterns) ?

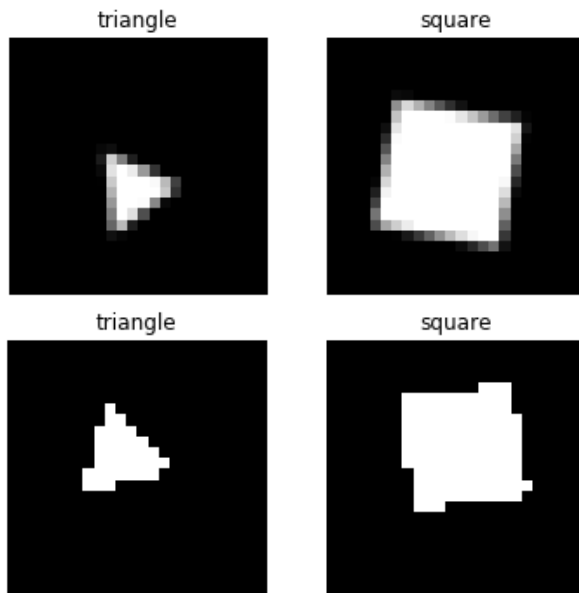
הפעלה של CNN

כעת נבדוק את יכולות CNN קטנה לבצע הפרדה בין צורות, בתנאים שונים. נתחיל ממשימת סיווג פשוטה ונעלה במורכבות של המשימה בצורה מבוקרת.

הכינו CNN לאבחנה בין משולשים לריבועים בתמונות ערכי אפור (gray scale), כלומר תמונות שבהן יש הצבעים נעים בין שחור ללבן. על פי ההנחיות הבאות:

- הכנת הנתונים.

הקלט לרשת יהיו תמונות ערכי אפור בגודל 25×25 .
בכל תמונה תופיע צורה סגורה אחת. הרקע יהיה מורכב מפיקסלים שערכם 0.
הצורה תורכב מפיקסלים שערכם 1 (או פחות מזה, כפי שיוסבר בהמשך).
הצורות יהיו משולשים (שווי צלעות) או ריבועים.
אורך צלע יוגרל באקראי בין רבע לחצי רוחב התמונה.
מיקום הצורה יוגרל באקראי כך שכל הצורה תופיע בתמונה.
סיבוב הצורה יוגרל באקראי.



הנה דוגמה ל 2 תמונות עם צורות:

שימו לב לשימוש בפיקסלים ברמות שונות של אפור בהיקפי הצורות.

להשוואה הסתכלו על 2 תמונות אחרות בהן רק ערכי שחור ולבן:

השימוש בערכי פיקסלים שונים כדי לקרב טוב יותר צורות אלכסוניות נקרא ¹anti-aliasing, ואנו נבחר בגרסה הזו בה הצורות נראות טוב יותר לעין האנושית.

הנתונים יוכנו על ידי קוד על פי השלבים הבאים:

- יצירת תמונה בגודל גדול פי 10 (250×250) על ידי הספרייה pil. סוג הנתונים של התמונה 8-bit pixel, black and white.
- ציור הצורה על גבי התמונה בעזרת הפונקציה polygon של pil. הקודקודים של הפוליגון שיועבר לפונקציה יחושבו על ידי קוד שלכם ויתארו ריבוע או משולש כנדרש.
- הקטנת התמונה לגודל הנדרש (25×25) על ידי resize של pil בשיטת דגמה bicubic.
- הפיכת התמונה למערך numpy מסוג float32 ונרמול הערכים בין 0 ל 1.

¹ https://en.wikipedia.org/wiki/Spatial_anti-aliasing

- שמירת התמונה במקום המתאים במערך התמונות.
- שמירת התיוג של התמונה במקום המתאים במערך התיוגים.

הקוד הזה יקרא כדי לייצר סט אימון, סט ולידציה וסט מבחן.

לאחר ייצור הסטים השונים כדאי לשמור אותם לדיסק, וכך לקרא אותם שוב לאחר שהוכנו. זה יחסוך לכם זמן בשלבים הבאים.

הסתכלו בקוד המצורף לתרגיל, בו מוכנים כבר חלקים רבים. השלימו את החלקים החסרים.

לפניכם כמה משימות קלסיפיקציה. בכל משימה עליכם לייצר את הנתונים המתאימים לה, ולמצוא את הרשת המינימלית (בעלת מספר הפרמטרים הקטן ביותר) שמבצעת את המשימה ומשיגה דיוק של לפחות 0.995 על סט הוולידציה. בדקו שהרשת שלכם עמידה מספיק – הריצו לפחות 5 אימונים וראו שמתקבלות תוצאות דומות בכל פעם. אם אתם מקבלים תוצאות לא אמינות, יתכן ושינוי קצב הלמידה יעזור.

להגשה בכל משימה:

הציגו עבור הרשת שמצאתם את הנתונים האלה:

- הפלט של `model.summary()`.
- מספר ה `epochs`, וגודל ה `batch` שבחרתם.
- גרף ה `Loss` כפונקציה של מספר ה `epoch`. כאן יש להציג `training loss`, `validation loss`.
- גרף `Accuracy` כפונקציה של מספר ה `epoch`. כאן יש להציג `training accuracy`, `validation accuracy`.
- ערך ה `test accuracy` של הרשת.
- התמונות שהרשת סיווגה לא נכון (בהנחה שהיו כאלה). אם יש יותר מ 10 תמונות שסווגו לא נכון, אז יש לציין את מספרן מכל מחלקה ולהראות 10 מתוכן (לשתי המחלקות אם יש כאלה).
- האקטיבציות של שכבות הקונבולוציה (אם בחרתם CNN).
- **חישוב** תאורטי של מספר האפשרויות לנתונים עבור המשימה (כמה תמונות שונות יש על פי תנאי המשימה). הסבירו את החישוב.

אין צורך להגיש את הקוד למשימות 8-13.

יש להגיש את הקוד למשימות 14-16.

המשימות:

8. אבחנה בין ריבוע אחד למשולש אחד

- גודל הצורות: שתיהן בעלות צלע באורך 15.
- מיקום הצורות: מרכז הצורה במרכז התמונה.
- כיוון הצורות: הפאה התחתונה מקבילה לציר האופקי.

- נתונים: 2 תמונות – אחת של ריבוע ואחת של משולש.
על כן סט האימון זהה לסטים האחרים.
batch=1.

למעשה זו משימה שאינה מתאימה כל כך ללמידה באמצעות רשת נוירונים, ואילו הנדסת מאפיינים ידנית פותרת אותה בקלות: אנחנו יכולים למצוא בקלות פיקסל יחיד המבחין בין התמונות ולהשתמש בו לסיווג. הניסיון שתנסו כאן יראה מה צריך כדי בכל אופן ללמוד סט פשוט זה.

9. אבחנה בין שתי הצורות (ריבוע ומשולש) בסיבוב אקראי

- גודל הצורות: שתיהן בעלות צלע באורך 10.
- מיקום הצורות: מרכז הצורה במרכז התמונה.
- כיוון הצורות: אקראי.
- נתונים:
- סט אימון – 600 תמונות (מחצית לכל מחלקה $\frac{1}{2}, \frac{1}{2}$)
- סט וולידציה – 200 תמונות $(\frac{1}{2}, \frac{1}{2})$
- סט מבחן – 200 תמונות $(\frac{1}{2}, \frac{1}{2})$

10. אבחנה בין שתי הצורות (ריבוע ומשולש) במיקום אקראי

- גודל הצורות: שתיהן בעלות צלע באורך 10.
- מיקום הצורות: אקראי בתנאי שלא נוגעות בקצה.
- כיוון הצורות: הפאה התחתונה מקבילה לציר האופקי.
- נתונים:
- סט אימון – 4000 תמונות $(\frac{1}{2}, \frac{1}{2})$
- סט וולידציה – 1000 תמונות $(\frac{1}{2}, \frac{1}{2})$
- סט מבחן – 1000 תמונות $(\frac{1}{2}, \frac{1}{2})$

11. אבחנה בין שתי הצורות (ריבוע ומשולש) במיקום אקראי וסיבוב אקראי

- גודל הצורות: שתיהן בעלות צלע באורך 10.
- מיקום הצורות: אקראי בתנאי שלא נוגעות בקצה.
- כיוון הצורות: אקראי.
- נתונים:
- סט אימון – 4000 תמונות $(\frac{1}{2}, \frac{1}{2})$
- סט וולידציה – 1000 תמונות $(\frac{1}{2}, \frac{1}{2})$
- סט מבחן – 1000 תמונות $(\frac{1}{2}, \frac{1}{2})$

12. אבחנה בין שתי הצורות (ריבוע ומשולש) בגודל אקראי ומיקום אקראי

- גודל הצורות: שתיהן בעלות צלע בגודל אקראי בין 6 ל 12
- מיקום הצורות: אקראי בתנאי שלא נוגעות בקצה.
- כיוון הצורות: הפאה התחתונה מקבילה לציר האופקי.
- נתונים:
- סט אימון – 4000 תמונות $(\frac{1}{2}, \frac{1}{2})$
- סט וולידציה – 1000 תמונות $(\frac{1}{2}, \frac{1}{2})$

סט מבחן – 1000 תמונות ($\frac{1}{2}, \frac{1}{2}$)

13. אבחנה בין שתי הצורות (ריבוע ומשולש) בגודל אקראי, מיקום אקראי, סיבוב

אקראי

- גודל הצורות: שתיהן בעלות צלע בגודל אקראי בין 6 ל 12
- מיקום הצורות: אקראי בתנאי שלא נוגעות בקצה.
- כיוון הצורות: אקראי.
- נתונים:

סט אימון – 20000 תמונות ($\frac{1}{2}, \frac{1}{2}$)

סט וולידציה – 2000 תמונות ($\frac{1}{2}, \frac{1}{2}$)

סט מבחן – 2000 תמונות ($\frac{1}{2}, \frac{1}{2}$)

14. אבחנה בין 3 מחלקות: ריבוע, משולש, שניהם (מופיעים בתמונה אחת עם חפיפה מקסימלית של מחצית שטח הצורה הקטנה יותר), כל הצורות בגודל, מיקום וסיבוב אקראיים (כמו ב 13). כאן יש להגיע לדיוק של 0.96 לפחות (בנוס על דיוק 0.98 לפחות).

○ נתונים:

סט אימון – 30000 תמונות (שליש לכל מחלקה)

סט וולידציה – 3000 תמונות (שליש לכל מחלקה)

סט מבחן – 3000 תמונות (שליש לכל מחלקה)

15. הצגת התמונות שגורמות למקסימום תגובה לפילטרים שהרשת (משאלה 14) מצאה.

קראו בספר בפסקה 5.4.2 על השיטה לעשות זאת.

16. הצגת מפת האקטיבציה למחלקה שהרשת (משאלה 14) מנבאה על פני התמונה המקורית.

קראו בספר בפסקה 5.4.3 על השיטה לעשות זאת.

בנוס (עד 10 נקודות!) יינתן על רשתות קטנות במיוחד, ניסויים של ארכיטקטורות שונות, הסברים יוצאי דופן ומסקנות מעניינות.

סיכום:

התרגיל הזה בודק יכולות של CNNs קטנות לביצוע משימות סיווג וויזואליות. התרגיל מאפשר לכם לקבל אינטואיציה באשר לקושי של משימות סיווג פשוטות, והסיבוך העולה בהתאם לסוגים שונים של invariances.

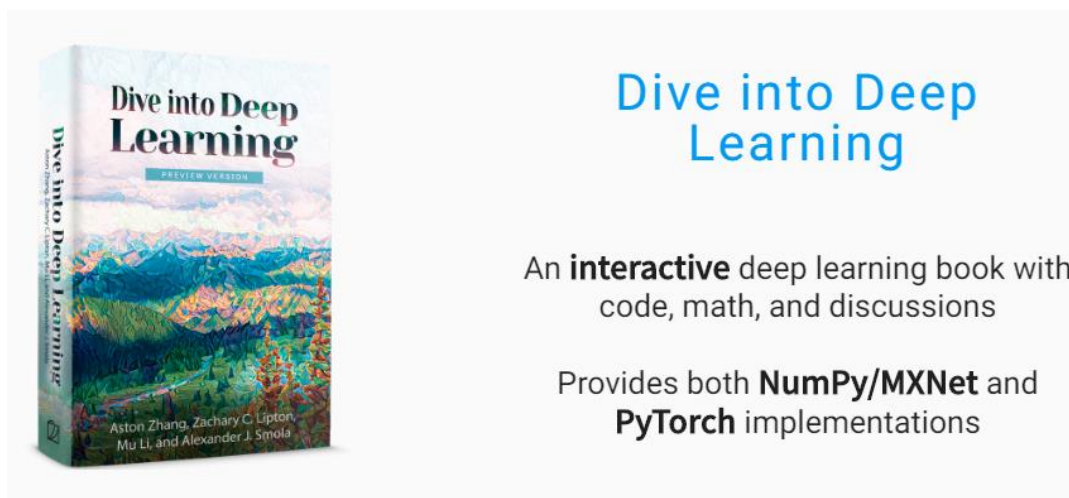
משימות קשורות הן זיהוי מיקום (של עצמים בתמונות), מניה, עקיבה, סגמנטציה. חישבו:

כיצד אפשר להרחיב את הקוד שבתרגיל כך שימנה כמה עצמים נמצאים בתמונה? כיצד אפשר להרחיב את הקוד כך שיחזיר את מיקום העצמים בתמונה? את התיבה התוחמת אותם?

המשימות שבוצעו בתרגיל זה הן משימות ויזואליות פשוטות. ברור כי למשימות פשוטות אלה אפשר לכתוב קוד יעיל שיפתור אותן מבלי להזדקק ל CNN. אבל כאשר מספר המחלקות עולה, והתמונות מפגינות שונות רבה יותר (כפי שיש בתמונות טבעיות) אזי מתגלה ההבדל בין הגישות. קשה מאוד לכתוב קוד מספיק כללי הפותר בעיות סיווג וזיהוי. לעומת זאת ארכיטקטורות CNN מודרניות מטפלות בהצלחה רבה מאוד בבעיות כאלה.

אתם מוזמנים לקרא בספר הנפלא הזה (בגרסת אתר או PDF), שבו הסברים ומימוש בקוד

Dive into Deep Learning <https://d2l.ai/index.html>



כדאי לעבור על כל הספר, אך הפרקים הקשורים לנושא הנוכחי הם:

- 6. Convolutional Neural Networks
- 7. Modern Convolutional Neural Networks
- 13. Computer Vision

הסברים נוספים ליצירת הקוד הדרוש בתרגיל:

א. מצורף קובץ

ex3_base_to_complete.py

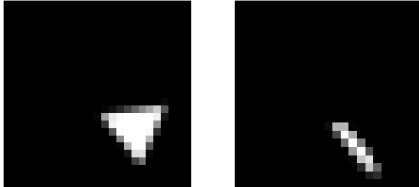
ובו תשתית קוד אותה עליכם להשלים.

ב. הסתכלות על הפעולות שהרשת עושה אחרי סיום האימון

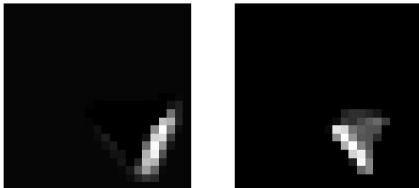
שכבות הקונבולוציה מבצעות טרנספורמציה על הקלט. נוכל לראות את תוצאות הטרנספורמציה בהצגת אקטיבציות של שכבות הקונבולוציה. העזרו בפרק 5 סעיף 5.4.1.

דוגמאות:

התמונה השמאלית העליונה הוזנה לרשת כקלט



התמונות האחרות הן תוצאות הפעולה של שלושה גרעיני קונבולוציה מן השכבה הראשונה שהרשת למדה

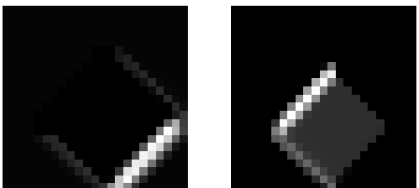
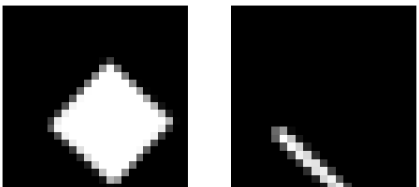


בשכבת הקונבולוציה השנייה היו 2 גרעינים ואלה תוצאות הפעלתם

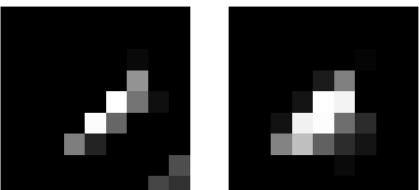
(שימו לב שבתוצאות השכבה השנייה יש פחות פיקסלים, זו כמובן תוצאה של שכבת maxpolling).



והנה הפעלת השכבה הראשונה על תמונה של ריבוע



ותוצאות הפעלת השכבה השנייה



הגשה

- א. תאריך הגשה: עד יום שני, 26.6.23, בשעת חצות הלילה.
 - ב. יש להגיש בזוגות. אסור לעבוד בקבוצות גדולות יותר. **הגשה ב Moodle**.
 - ג. יש לכתוב **שם \ שמות + ת"ז** בראשית המסמכים המוגשים.
 - ד. כל מגיש צריך לדעת להסביר כל מה שנעשה בפתרון המוגש. חלק מן המגשים ידרשו להסביר את הפתרון שלהם למרצה.
 - ה. יש להגיש מסמך **Word** המכיל את כל התשובות לתרגיל. הקפידו שמספור סעיפי התשובות שלכם יהיה **זהה** למספור סעיפי השאלות.
 - ו. שם מסמך זה יהיה מורכב משני שמות המשפחה של המגשים:
EX3Family1Family2.docx.
 - ז. יש להגיש את הקוד הפותר את שאלות 14-16 בקובץ **ex3.py**
- שאלות בקשר לתרגיל נא לשלוח למרצה באימייל: yoramye@hac.ac.il
- שאלות שרלוונטיות לכלל הכיתה יועלו לפורום הקורס על ידי המרצה.