

פרויקט גמר 5 יחידות לימוד

התמחות – תכנון ותכנות מערכות Deep Learning

Clothing Classification with Fashion-MNIST dataset using deep learning

מגישה: הילה דניאל

כיתה: יב-4



בית ספר: מקיף י"א ראשונים ראשון לציון

מנחה: דינה קראוס, רכזת מגמת הנדסת תכנה ומורה במגמה

תאריך הגשה: 20.06.2021



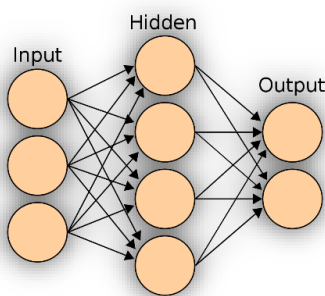
תוכן עניינים:

מבוא.....	3
מדריך למשתמש.....	5
מדריך למפתח.....	14
מסקנות הרצת המודל.....	20
רפלקציה / סיכום אישי.....	23
ביבליוגרפיה.....	24
נספחים.....	25

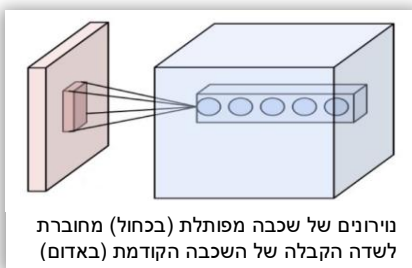
מבוא

נושא ההתמחות שלנו בבית הספר במגמת הנדסת תכנה היה השנה "למידה עמוקה" – Deep Learning. "למידה עמוקה" נחשבת תת נושא בתוך תחום הבינה המלאכות (Artificial Intelligence). DL עוסק בלמידת מכונה באמצעות רשתות נוירונים מלאכותיות המנסות לחקות את פעילות המוח, פעולת הלמידה האנושית. מוח האדם בנוי כרשת של נוירונים המקושרים ביניהם, כך שלרשת כולה מתפתחות יכולות חשיבה ולמידה המאפשרות לפתור בעיות מורכבות. תחום זה חולל מהפכה ואפשר לפתור בעיות הנדסיות בדרכים יעילות. כיום ניתן לראות שלל פיתוחים והמצאות מבוססים Deep Learning במגוון תחומים: רפואה, תחבורה, כלכלה ועוד.

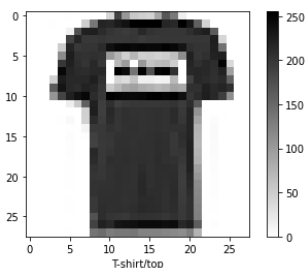
התכנה שפיתחתי עוסקת ב- multi-class classification ומטרתה היא לסווג פריטי לבוש. הלמידה נעשית באמצעות deep learning בעזרת מודל המבוסס על רשתות עצביות מפותלות (CNN).



רשת עצבית (NN) היא מודל מתמטי חישובי שפותח בהשראת תהליכים מוחיים או קוגניטיביים המתרחשים ברשת עצבית טבעית ומשמש במסגרת למידת מכונה. רשת מסוג זה מכילה בדרך כלל מספר רב של יחידות מידע (קלט ופלט) המקושרות זו לזו, קשרים שלעיתים קרובות עוברים דרך יחידות מידע "חבויות" (Hidden Layer). צורת הקישור בין היחידות, המכילה מידע על חוזק הקשר, מדמה את אופן חיבור הנוירונים במוח.



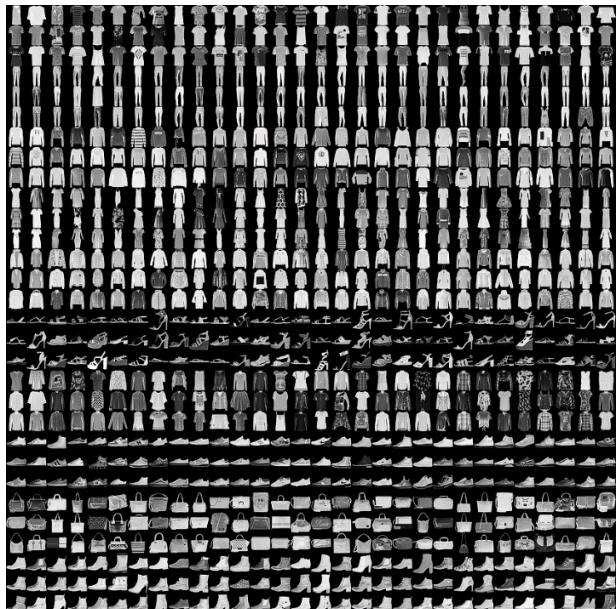
רשת עצבית מפותלת (Convolutional Neural Networks) היא סוג של רשת עצבית, המשמשת בעיקר לניתוח דימויים חזותיים, כמו תמונות וסרטונים. אלה מקבלות כרכים תלת ממדיים (רוחב, גובה ועומק) של נוירונים. כאשר כל נוירון בתוך שכבה מפותלת מחובר רק לאזור קטן בשכבה שלפניו, שנקרא שדה קבלה.



תמונה מן הדטאסט להמחשה עם המאפיינים שלה כמתואר לעיל.

בפרויקט אני משתמשת במערך נתונים [Fashion-MNIST Dataset](#) המכיל 70,000 תמונות מרובעות בגווי אפור של פריטי לבוש המסווגים ל-10 קטגוריות, 7,000 תמונות בכל קטגוריה. 60,000 תמונות משמשות לאימון רשת הנוירונים (train), ו-10,000 תמונות משמשות להערכת מידת הדיוק של הרשת לסיווג התמונות של פריטי הלבוש (כלומר ל- test). התמונות הן מערכי NumPy בגודל 28X28, עם ערכי פיקסלים שנעים בין 0 ל-255 / 0 –לבן, 255 – שחור. כל תמונה ב- training set וב- testing set משויכת לתווית – label. התוויות הן מערך של מספרים שלמים שנעים בין 0 ל-9, כאשר כל מספר מייצג את ה- class של פריט הלבוש בתמונה.

*Fashion-MNIST dataset מהווה מעין תחליף למערך הנתונים המקורי MNIST שמכיל תמונות של ספרות בכתב יד ומשמש לאלגוריתמים של למידת מכונה. שניהם חולקים את אותו גודל תמונה, פורמט נתונים ומבנה של חלוקת הדתא ל- train ו- test.



תמונה הממחישה איך נראה הדתא.

בתמונה ניתן לראות בגדי לבוש בודדים ברזולוציה 28x28 פיקסלים (במסמך זה התמונה מוקטנת).

Class	Label
T-shirt/top	0
Trouser	1
Pullover	2
Dress	3
Coat	4
Sandal	5
Shirt	6
Sneaker	7
Bag	8
Ankle boot	9

קיימות מספר דרכים להשיג את Fashion-MNIST dataset. אני בחרתי לייבא ולטעון את הדתא ישירות מספריית TensorFlow הכוללת את Fashion-MNIST כמערכת נתונים מובנית, באמצעות שימוש ב-`tf.keras`.

התחלת הרצת הפרויקט (הרצת הסקריפט `project.py`): בכניסה לתוכנה יש למשתמש 5 אפשרויות: להכיר את Fashion-MNIST dataset, לטעון מודל קיים, לאמן את המודל, לבחון את המודל ולצאת מהתכנית. הפלט הבא מתקבל בהתאם לבחירת המשתמש.

אופן פעולת התוכנה: התוכנה לומדת לזהות תמונות של פריטי הלבוש ממאגר התמונות Fashion-MNIST dataset.

סיום ההרצה: ההרצה מסתיימת כאשר המשתמש בוחר את האופציה החמישית – יציאה `/exit`.

דרישות להרצה:

- Python 3.8
- מערכת הפעלה windows
- הספריות בהן משתמשים הן: `numpy`, `sklearn`, `TensorFlow`, `Keras`, `matplotlib`, `graphviz`, `pydot`.

מדריך למשתמש

את הפרויקט כתבתי בהפצה הנקראת אנקונדה בסביבת עבודה Spyder (שכלולה כבר בהתקנה של אנקונדה).

ניתן להוריד את אנקונדה בלינק הבא: <https://www.anaconda.com/products/individual>, או לחילופין להשתמש בכל IDE אחר.

מדריך לעבודה עם אנקונדה: <https://conda.io/projects/conda/en/latest/user-guide/index.html>

ספריות קוד הדרושות להרצת הפרויקט:

שם הספרייה	הוראת התקנה שיש להקליד בשורת הפקודה של אנקונדה	גרסה של הספרייה שבה הפרויקט רץ	למה משמשת הספרייה בפרויקט
numpy	pip install numpy	1.20.1	משמשת לביצוע פעולות מתמטיות וחישוביות על מערכים המוגדרים בפרויקט.
TensorFlow	pip install tensorflow	2.4.0	משמשת לייבוא ולטעינה של ה- Fashion-MNIST dataset.
Keras	pip install keras	2.4.3	משמשת לבניית המודל.
Sklearn	pip install scikit-learn		משמשת להערכת המודל (כמה המודל לומד לסווג את פריטי הלבוש).
matplotlib	pip install matplotlib	3.3.4	משמשת להצגת תמונות לדוגמה של הדתאסט ולהצגת תרשימים על תהליך הלמידה של המודל.
pydot	pip install pydot	1.4.2	משמשת להצגת התרשים הוויזואלי של מבנה המודל.
graphviz	pip install graphviz	0.16	משמשת להצגת התרשים הוויזואלי של מבנה המודל.

*הספריות numpy, TensorFlow, Keras ו-matplotlib, graphviz ו-pydot כלולות ברשימת הספריות שאנקונדה מציעה לשימוש ולכן ניתן להורידן ישירות מאנקונדה בדרך הזאת:

פתח את Anaconda Navigator -> לחץ בתפריט בצד שמאל של המסך על Environments -> הקלד בשורת החיפוש את שם הספרייה -> אם אתה נמצא בסביבת ה-base של אנקונדה, ייתכן שהספרייה מותקנת מראש (למשל numpy), אולם ייתכן גם שתצטרך להתקין אותה ידנית (למשל Keras) בדרך הזאת: סמן את הספרייה שחיפשת ותרצה להתקין (הסימון יופיע בוי ירוק) -> לחץ apply.

הרצת התכנית:

הרצת התכנית תתבצע בשורת הפקודה/Command Line של סביבת העבודה Anaconda (רצוי בסביבת ה-base של אנקונדה).

ב-CMD זה יש לכתוב python ולאחר מכן להעתיק את הכתובת/מיקום של קובץ הפיתוח project.py. הרצת קובץ זה תריץ בפני המשתמש את כל התכנית.

הרצת הקובץ תדפיס בפני המשתמש תפריט עם 6 אפשרויות כמצולם בתמונה המצורפת והמשתמש ישאל מה הוא מעוניין לעשות.

```
Anaconda Prompt (anaconda3) - python C:\Users\Daniel\proj_hila\project.py

(base) C:\Users\Daniel>python C:\Users\Daniel\proj_hila\project.py

Wellcome!
This is a deep learning project of Clothes Classifier using Fashion MNIST Dataset.

What would you like to do?
1. Get to know the dataset
2. Train the model
3. Load the model
4. Test the model
5. Predict an image
6. Exit
Enter your choice:
```

*עבור קלט לא תקין תודפס ההודעה:

```
Anaconda Prompt (anaconda3) - python C:\Users\Daniel\proj_hila\project.py

(base) C:\Users\Daniel>python C:\Users\Daniel\proj_hila\project.py

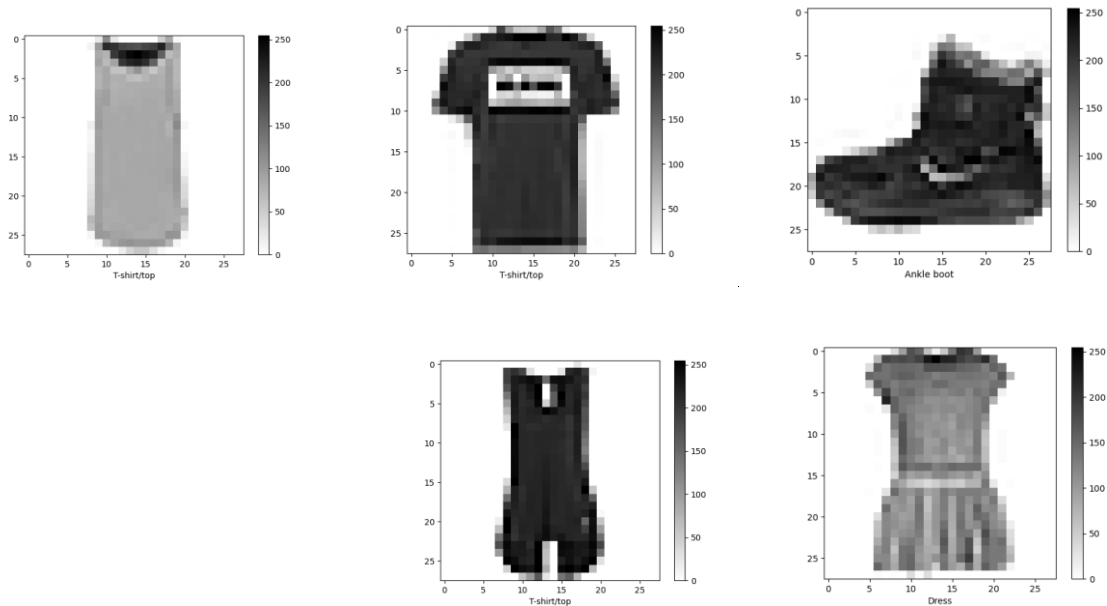
Wellcome!
This is a deep learning project of Clothes Classifier using Fashion MNIST Dataset.

What would you like to do?
1. Get to know the dataset
2. Train the model
3. Load the model
4. Test the model
5. Predict an image
6. Exit
Enter your choice: 7
Sorry, this is not a valid input.
Please enter your choice again:
```

בחירה באופציה הראשונה תדפיס בפני המשתמש מידע על המערכים של הדתא- training set וה- testing set. בפלט זה ניתן לראות את מספר האלמנטים בכל מימד במערך התמונות של ה- train ושל ה- test, לראות את מספר התמונות המשמשות ל- train (60,000) ומספר התמונות המשמשות ל- test (10,000) ולראות שהמערכים של התוויות (הן של ה- train והן של ה- test) הם מערכים חד ממדיים של מספרים שלמים מ-0 עד 9 כאמור.

```
Please enter your choice again: 1
Train dataset shape: (60000, 28, 28)
Train labels shape: [9 0 0 ... 3 0 5]
Sum of training images (also sum of train-labels): 60000
Test dataset shape: (10000, 28, 28)
Test labels shape: [9 2 1 ... 8 1 5]
Sum of testing images (also sum of test-labels): 10000
```

בנוסף, בחירה באופציה 1 מציגה את חמש התמונות הראשונות של ה-train, כאשר כל תמונה בגודל 28X28 פיקסלים שנעים בין 0 ל-255 / לבן לשחור.



בחירה באופציה 2 תוביל לשאלה ממוקדת יותר עבור המשתמש:

האם ברצונו:

לאמן את המודל ולהעריך את ביצועיו תוך כדי בעזרת validation

או-

לאמן את המודל בלבד

```
What would you like to do?
1. Get to know the dataset
2. Train the model
3. Load the model
4. Test the model
5. Predict an image
6. Exit
Enter your choice: 2
What do you prefer?
1. Doing training and validation.
2. Doing training only.
Enter your choice:
```

בחירה באופציה 2.1 תאמן את המודל ותעריך את ביצועיו 5 פעמים, וגם תשמור אותו. כל ערכת validation תהיה 20% ממערך ה-train, כלומר כ- 12,000 תמונות. הפלטים הבאים יוצגו:

I. מודפסים ערכי ה- accuracy בכל k-fold cross-validation:
(משתנים מהרצה להרצה, להלן דוגמה לערכים שקיבלתי באחת ההרצות)

```
What do you prefer?
1. Doing training and validation.
2. Doing training only.
Enter your choice: 1

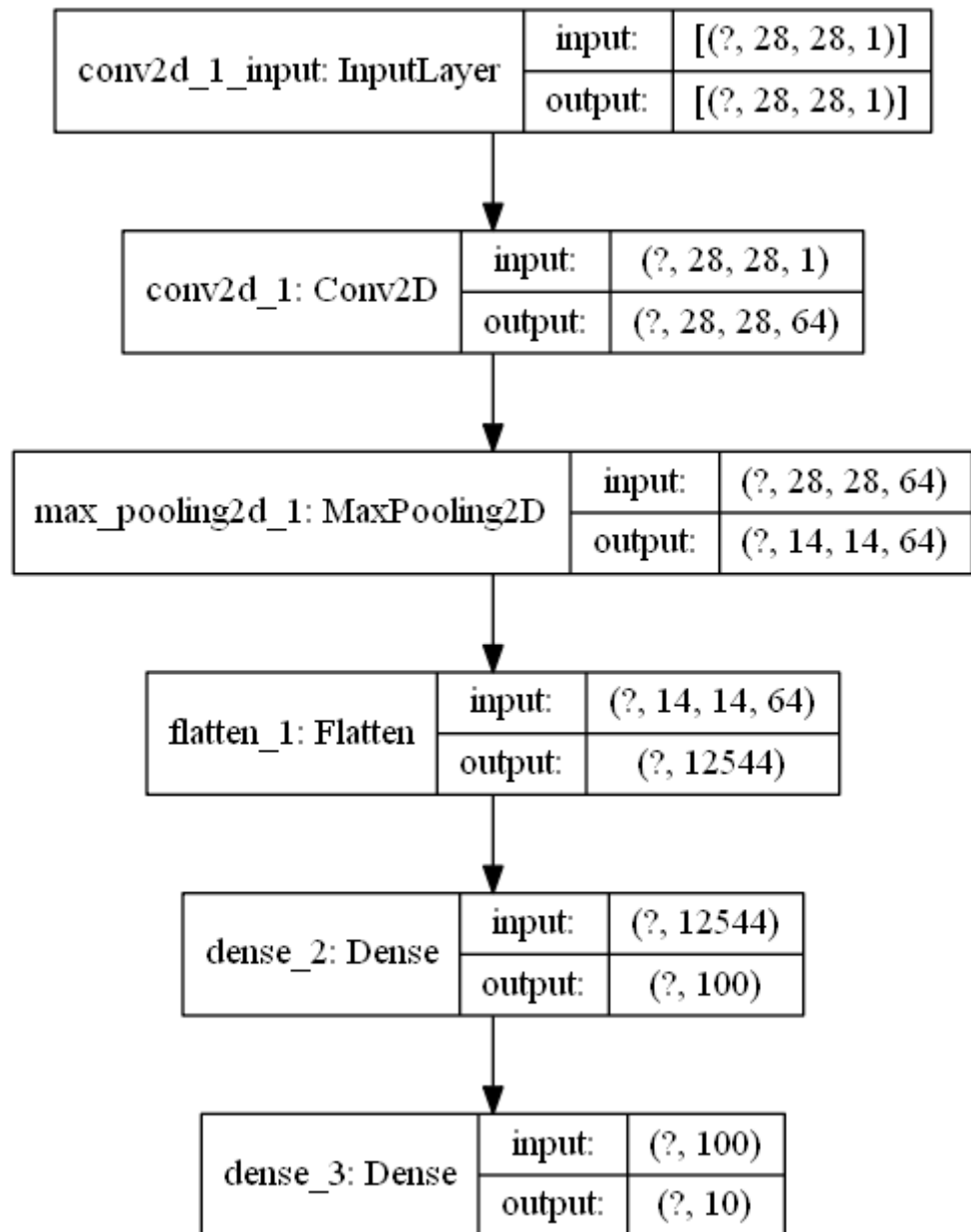
Classification accuracy for each fold of the cross-validation process:
2021-07-03 11:33:47.962715: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with
oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
> 91.317
> 91.742
> 91.633
> 90.967
> 91.117
```

II. מודפס סיכום מילולי של המודל הכולל מידע על: שכבות המודל וסדרן, צורת הפלט של כל שכבה, מספר הפרמטרים בכל שכבה ומספר הפרמטרים הכולל במודל.

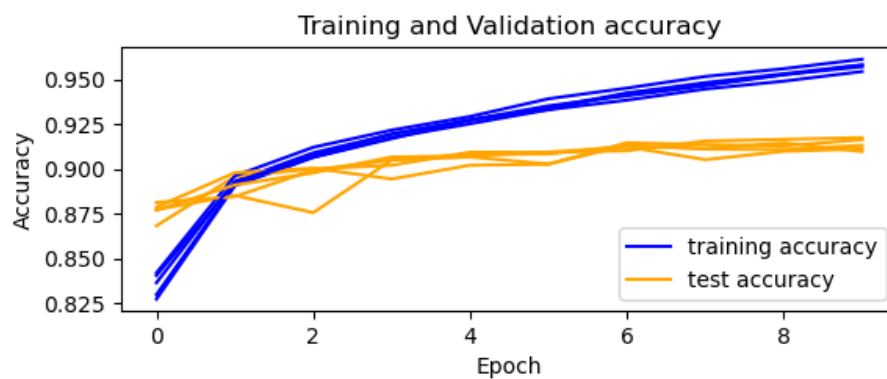
```
Model: "sequential_4"
Layer (type)                Output Shape                 Param #
-----
conv2d_4 (Conv2D)           (None, 28, 28, 64)          640
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 64)          0
flatten_4 (Flatten)         (None, 12544)                0
dense_8 (Dense)             (None, 100)                  1254500
dense_9 (Dense)             (None, 10)                   1010
-----
Total params: 1,256,150
Trainable params: 1,256,150
Non-trainable params: 0
None
```

III. נשמר תרשים ויזואלי של מבנה המודל כתמונה בשם model_plot בתיקייה שבה שמור הקובץ project.

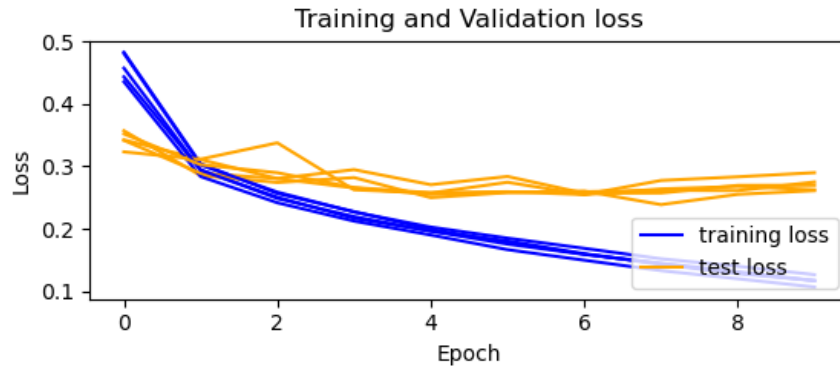
A plot of the neural network model is waiting for you in proj_hila folder!



IV. נפתח תרשים המתאר את התנהגות המודל על ה- training set ועל ה- validation set במהלך כל fold של ה- k-fold cross-validation לפי accuracy:



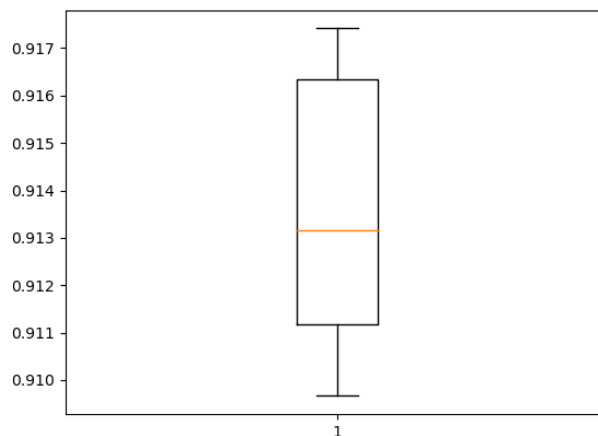
V. נפתח תרשים המתאר את התנהגות המודל על ה- training set ועל ה- validation במהלך כל fold של ה- k-fold cross-validation לפי loss :



VI. מודפסים במסך שלושה ערכים הקשורים לדיוק הסיווג-accuracy :
הערך הממוצע של ערכי ה- accuracy שנאספו במהלך כל fold של ה- k-fold cross-validation,
סטיית התקן - standard deviation,
וכמה ערכים של accuracy יש (שזה למעשה מספר ה- folds).

Accuracy: mean=91.355 std=0.295, n=5

VII. נפתח תרשים box plot נקרא גם (whisker plot) המציג סיכום ויזואלי של ערכי ה- accuracy : מינימום, רבעון ראשון, חציון, רבע שלישי ומקסימום.



בחירה באופציה 2.2 תאמן את המודל פעם אחת על כל ה- training set, תשמור אותו ותדפיס :

- I. סיכום מילולי של המודל כפי שהודפס באפשרות 2.1
- II. סיכום ויזואלי של המודל כפי שהודפס באפשרות 2.1
- III. הודעה שהמודל אומן ונשמר בהצלחה. : The model is successfully trained and saved.

בחירה באופציה 3 תטען את המודל שנשמר מראש ותדפיס :

```
Done. Model is loaded.
```

בחירה באופציה 4 מאפשרת למשתמש לבחון את המודל הקיים על ה- testing data.

- אם אופציה זו נבחרת בפעם הראשונה שבה מתבקש המשתמש לבחור אופציה מהתפריט, תודפס ההודעה :

```
What would you like to do?
1. Get to know the dataset
2. Train the model
3. Load the model
4. Test the model
5. Predict an image
6. Exit
Enter your choice: 4
Please load or train the model first.
```

כלומר, צריך תחילה לטעון מודל קיים או לאמן מודל חדש.

- אם אופציה זו נבחרת אחרי שנבחרה קודם אופציה 2 או אופציה 3.1 או אופציה 3.2, אזי תודפס מידת הדיוק של המודל הנוכחי לסיווג פריטי הלבוש בתמונות המופיעות ב- testing data כלומר test-accuracy, וכן יודפס ה- test-loss.

למשל, אם המשתמש יבחר את אופציה 2.1 (כלומר ב- train ו- validation), ואחריה את אופציה 4, יודפס :

```
Test accuracy: 0.8970999717712402
Test loss: 0.3130113482475281
```

ואם המשתמש יבחר באפשרות 2.2 (כלומר באימון המודל על כל ה- training data), ואחריה את אפשרות 4, יודפס :

```
What would you like to do?
1. Get to know the dataset
2. Train the model
3. Load the model
4. Test the model
5. Predict an image
6. Exit
Enter your choice: 4

Test accuracy: 0.911300003528595
Test loss: 0.2895078957080841
```

בחירה באופציה 5 מאפשרת למשתמש לעשות חיזוי של המודל לפריט לבוש המופיע בתמונה שלקוחה מה-
testing data.

- אם אופציה זו נבחרת בפעם הראשונה שבה מתבקש המשתמש לבחור אופציה מהתפריט, תודפס
ההודעה:

```
Wellcome!  
This is a deep learning project of Clothes Classifier using Fashion MNIST Dataset.  
  
What would you like to do?  
1. Get to know the dataset  
2. Train the model  
3. Load the model  
4. Test the model  
5. Predict an image  
6. Exit  
Enter your choice: 5  
Please load or train the model first.
```

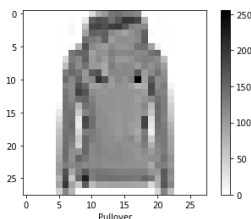
כלומר, צריך תחילה לטעון מודל קיים או לאמן מודל חדש.

- אם אופציה זו נבחרת אחרי שנבחרה קודם אופציה 2 או אופציה 3.1 או אופציה 3.2, אזי ניתן
לחזות תמונה. במקרה כזה המשתמש יתבקש להקליד מספר מ-1 עד 10,000 (מספר זה פחות
מייצג את האינדקס של התמונה הנבחרת לחיזוי). לאחר מכן תוצג התמונה של פריט הלבוש אותה
כביכול בחר המשתמש. לבסוף, יודפס החיזוי של המחשב לפריט הלבוש המופיע בתמונה. אם
החיזוי נכון תודפס ההודעה 😊 Prediction is true, אחרת- 😞 Prediction is wrong.

למשל, אם המשתמש יבחר את אופציה 2.1 (כלומר ב- train ו- validation), ואחריה את אופציה
4, יודפס:

Choose a number between 1 to 10,000: 55

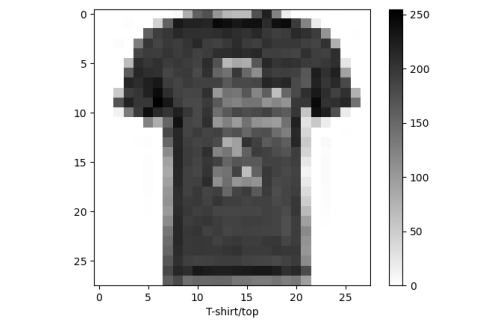
You choose to predict Pullover



Prediction is: Pullover
Prediction is true :)

ואם המשתמש יבחר באפשרות 2.2 (כלומר באימון המודל על כל ה- training data), ואחריה את
אפשרות 4, יודפס:

```
What would you like to do?
1. Get to know the dataset
2. Train the model
3. Load the model
4. Test the model
5. Predict an image
6. Exit
Enter your choice: 5
Choose a number between 1 to 10,000: 756
You choose to predict T-shirt/top
Prediction is: T-shirt/top
Prediction is true :)
```



בחירה באופציה 6 תביא לסיום ההרצה.

```
What would you like to do?
1. Get to know the dataset
2. Train the model
3. Load the model
4. Test the model
5. Predict an image
6. Exit
Enter your choice: 6

(base) C:\Users\Daniel>
```

מדריך למפתח

הפרויקט שלי מחולק לשלושה קבצי קוד אשר כל אחד מהם אחראי על תחום אחר בפרויקט. בחרתי לחלק את התכנית למספר קבצים של קוד מפני שחלוקה כזו מאפשרת ניווט קל ונוח בקוד, מקלה על קריאתו ותורמת לארגון ולסדר שלו לפי תחומי אחריות שונים.

להלן שמות הקבצים השונים ותפקידם:

שם הקובץ	מטרת הקובץ	האם מזמן קובץ פייתון? אם כן, איזה?	על ידי איזה קובץ פייתון הוא מזומן?
project.py	הקובץ הראשי שמטרתו לנהל את התכנית לפי בחירות המשתמש ולקרוא לפונקציות המתאימות.	כן, מזמן את קבצי הקוד manage_dataset.py ו- model.py	אינו מזומן על ידי קובץ קוד אחרי מפני שקובץ זה משמש כקובץ הראשי.
manage_dataset.py	לרכז את הטיפול בדתאסט- הקובץ כולל ייבוא וטעינה של הדתאסט, הכנסת הדתא לתוך משתנים מתאימים, הדפסה מידע על הדתאסט והכנתו לאימון הרשת.	לא.	מזומן על ידי הקובץ הראשי project.py.
model.py	להגדיר את מודל ה- cnn שישמש ללמידת הבעיה.	לא.	מזומן על ידי הקובץ הראשי project.py.

קובץ ראשי project.py:

*בקובץ זה מוגדר משתנה גלובלי בשם class_names מסוג רשימה. זו רשימה של מחרוזות, שה- len שלה הוא 10. כל איבר ברשימה הוא מחרוזת שהיא אחת מ-10 הקטגוריות של פריטי הלבוש (כלומר class). מיקום המחרוזות הוא בהתאמה ל- label של אותו class.

```
class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

כותרת הפעולה	תיאור הפעולה
main()	הפעולה הראשית המנהלת את התכנית. היא מזמנת את הפעולות של הקובץ manage_dataset.py כדי לטעון את הדתא ולאחר מכן היא בודקת את ההחזרה מ- showMenu ופועלת בהתאם לדרישות המשתמש. לשם כך היא מזמנת בהתאמה את הפעולות:

option_1, option_2, option_3, option_4 ו- option_5.	
פעולה זו מציגה את תפריט המשתמש, קולטת קלט בתור בחירת המשתמש, בודקת את תקינותו ומחזירה את בחירת המשתמש ל-main, נקראת בפעולה ה-main.	print_menu()
הפעולה מקבלת את ה-training set ואת מספר ה-folds עבור cross-validation. הפעולה מגדירה את המודל באמצעות זימון הפעולה define_model, מאמנת אותו ומעריכה את ביצועיו 5 פעמים. כל ערכת validation מהווה 20% ממערך ה-train, או כ- 12,000 דגימות. הפעולה מחזירה רשימה של ערכי ה-accuracy, רשימה של היסטוריית האימונים של המודל ואת המודל עצמו. פעולה זו נקראת בפעולה option_2.	training_and_validation(data_x, data_y, n_folds=5)
הפעולה מקבלת את רשימת היסטוריית האימונים של המודל שנאספו בפעולה הקודמת. הפעולה מציגה את ביצועי המודל ב-train וב-test עבור כל k-fold cross-validation בתרשים של Training and Validation accuracy. קווים כחולים מציינים את ביצועי המודל על ה- training set וקווים כתומים מציינים את ביצועי המודל על ה-validation. פעולה זו נקראת בפעולה option_2.	plot_accuracy(histories)
הפעולה מקבלת את רשימת היסטוריית האימונים של המודל שנאספו בפעולה def training_and_validation ומציגה את ביצועי המודל ב-train וב-test עבור כל k-fold cross-validation, בתרשים של Training and Validation loss. קווים כחולים מציינים את ביצועי המודל על ה- training set וקווים כתומים מציינים את ביצועי המודל על ה-validation. פעולה זו נקראת בפעולה option_2.	plot_loss(histories)
הפעולה מקבלת את רשימת ערכי ה-accuracy שנאספו במהלך כל fold בפעולה הקודמת. הפעולה מסכמת ערכים אלו על ידי חישוב סטיית התקן והממוצע ובכך מספקת אומדן של הביצועים	plot_accuracy_scores(scores)

<p>הצפויים הממוצעים של המודל שהוכשר במערך נתונים זה. לבסוף הפעולה מציגה את התפלגות הערכים של scores באמצעות box and whisker plot. פעולה זו נקראת בפעולה option_2.</p>	
<p>הפעולה אחראית על ביצוע האפשרות הראשונה בתפריט – להכיר את הדתאסט. הפעולה מקבלת את ארבעת מערכי ה-dataset המקוריים, כלומר אלו שלא עברו שינוי גודל/שינוי מבנה/נרמול (בפעולות prepare_dataset ו-explore_Fashion_MNIST_data). הפעולה נקראת ב-main().</p>	<p>option_1(train_images_orl, train_labels_orl, test_images_orl, test_labels_orl)</p>
<p>הפעולה אחראית על ביצוע האפשרות השנייה בתפריט – לאמן מודל חדש. הפעולה מקבלת את המערכים של ה-training data ופועלת בהתאם לבחירת המשתמש – לאמן מודל עם validation או לאמן מודל על כל ה-training data ללא validation. הפעולה מזמנת את הפעולות הבאות:</p> <ul style="list-style-type: none"> - training_and_validation (אם המשתמש בחר באפשרות הראשונה), - define_model (אם המשתמש בחר באפשרות השנייה), - summarize_model - visualize_model <p>הפעולה מציגה מידע על המודל, ושומרת אותו ומחזירה שני משתנים: הראשון- ערך בוליאני – אמת, המסמן כי נשמר מודל והשני – את המודל. הפעולה נקראת ב-main().</p>	<p>option_2(train_images, train_labels)</p>
<p>הפעולה אחראית על ביצוע האפשרות השלישית בתפריט – לטעון מודל קיים. הפעולה טוענת את המודל האחרון שנשמר בשם final_model.h5 בתיקייה שבה שמור הקובץ</p>	<p>option_3()</p>

<p>project.py ומדפיסה הודעה שהטעינה בוצעה בהצלחה.</p> <p>הפעולה מחזירה ערך בוליאני-אמת המסמן כי נטען המודל ואת המודל שנטען.</p> <p>הפעולה נקראת ב-main().</p>	
<p>הפעולה אחראית על ביצוע האפשרות הרביעית בתפריט – לבחון את המודל.</p> <p>הפעולה מקבלת את המערכים של ה-testing data ואת המודל ובוחנת את המודל עליהם.</p> <p>הפעולה מדפיסה את ה-Test accuracy ו-Test loss.</p> <p>הפעולה נקראת ב-main().</p>	<p>option_4(test_images, test_labels, model)</p>
<p>הפעולה אחראית על ביצוע האפשרות החמישית בתפריט – לחזות תמונה.</p> <p>הפעולה מקבלת את המערכים המקוריים של ה-testing data כלומר אלו שלא עברו שינוי גודל/שינוי מבנה/נרמול (בפעולות prepare_dataset ו-scale_pixel_values) ואת המודל.</p> <p>הפעולה קולטת מהמשתמש מספר בין 1 ל-10,000 המייצג את התמונה שרוצים לחזות, מדפיסה את הקטגוריה אליך שייך פריט הלבוש בתמונה ומזמנת את הפעולה show_Image כדי להציג את התמונה הנבחרת.</p> <p>הפעולה חווה את פריט הלבוש המופיע בתמונה ומדפיסה את ה-prediction, אם החיזוי נכון מודפסת ההודעה : Prediction is true ואם החיזוי שגוי מודפסת ההודעה Prediction is wrong): wrong.</p> <p>הפעולה נקראת ב-main().</p>	<p>option_5(test_images_orl, test_labels_orl, model)</p>

קובץ הגדרת מבנה המודל ושכבותיו, והצגת מידע על המודל - model.py :

תיאור הפעולה	כותרת הפעולה
<p>הפעולה מגדירה מודל מסוג Convolutional Neural Networks (CNN). הגדרת רשת נוירונים מפותלת נעשית כך :</p> <p>1. קונבולוציה (Convolution): Convolutional Layer היא השכבה הראשונה במודל המשמשת לחילוף התכונות השונות מתמונות הקלט. בשכבה זו נוצרות</p>	<p>define_model()</p>

<p>מפות של מאפיינים (Features) מתוך נתוני הקלט שלנו. לאחר מכן מופעלת פונקציה לסינון המפות.</p> <p>2. Max-Pooling : פעולה המאפשרת לרשת הנוירונים המפותלת לזהות תמונה כשהיא מוצגת עם שינוי. המטרה העיקרית של שכבה זו היא להקטין את גודל מפת התכונות המפותלת כדי להפחית את עלויות החישוב.</p> <p>3. השטחה (Flattening) : השטחת הנתונים לכדי מערך נתונים על מנת שרשת הנוירונים המפותלת תוכל לקרוא אותו.</p> <p>4. Dense : במודל זה מוגדרות 2 שכבות dense לצורך הסיווג.</p> <p>* כל השכבות משתמשות ב-ReLU activation function. * הואיל ומדובר ב-multi-class classification, נשתמש גם ב-softmax activation function.</p> <p>5. שכבת פלט.</p> <p>6. שלב ה-complie : לפני שהמודל מוכן לאימון, יש להגדיר מספר הגדרות נוספות : Loss function - מחשבת את פונקציית ההפסד עבור המודל, או במילים אחרות מודדת עד כמה המודל מדויק במהלך האימון. Optimizer - כך מתעדכן המודל על סמך הנתונים שהוא מקבל ולפי ה-Loss function. Metrics - משמש לבדיקת תהליך הלמידה של המודל. במודל המוגדר בפעולה זו השתמשתי ב-accuracy המעידה על שבר התמונות שמסווגות כהלכה.</p> <p>הפעולה מזומנת בקובץ project.py על ידי הפעולות training_and_validation ו-option_2.</p>	
<p>הפעולה מקבלת כפרמטר את המודל ומדפיסה סיכום טקסטואלי עליו הכולל מידע על : שכבות המודל וסדרן, צורת הפלט של כל שכבה, מספר הפרמטרים בכל שכבה ומספר הפרמטרים הכולל במודל.</p> <p>הפעולה מזומנת בקובץ project.py על ידי הפעולה option_2.</p>	<p>summarize_model(model)</p>
<p>הפעולה מקבלת כפרמטר את המודל ויוצרת תרשים ויזואלי של מבנה המודל. התרשים נשמר כתמונה model_plot.png בתיקייה שבה נמצא קובץ הפייתון project.py.</p> <p>הפעולה מזומנת בקובץ project.py על ידי הפעולה option_2.</p>	<p>visualize_model(model)</p>

קובץ טיפול בדתאסט - `manage_dataset.py`:

*גם בקובץ זה מוגדר משתנה גלובלי בשם `class_names` זהה למשתנה הגלובלי בקובץ `project.py`.

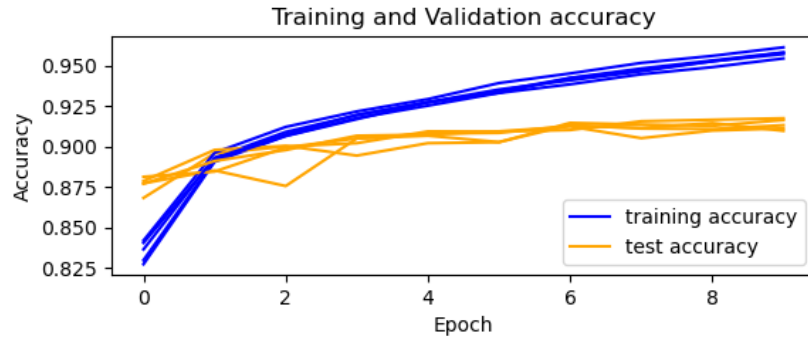
כותרת הפעולה	תיאור הפעולה
<code>load_Fashion_MNIST_data()</code>	<p>הפעולה טוענת את Fashion-MNIST dataset מספריית TensorFlow בעזרת ממשק API של Keras. טעינת הדתאסט מחזירה ארבעה מערכי numpy אותם הפעולה מגדירה לתוך משתנים מתאימים ומחזירה אותם. המערכים הם:</p> <p>Training set: <code>train_images</code>, <code>train_labels</code>.</p> <p>המודל משתמש בהם לצורך למידת הבעיה.</p> <p>Testing set: <code>test_images</code>, <code>test_labels</code>.</p> <p>המודל נבחן בעזרת מערכים אלו עד כמה הוא מסווג נכון את פריטי הלבוש. הפעולה נקראת בקובץ <code>project.py</code> בפעולה הראשית.</p>
<code>explore_Fashion_MNIST_data(train_images, train_labels, test_images, test_labels)</code>	<p>הפעולה מקבלת את ארבעת המערכים המייצגים את הדתאסט ומדפיסה מידע רלוונטי עליהם. הפעולה גם מציגה את חמש התמונות הראשונות של ה-training set באמצעות זימון הפעולה <code>.show_Image(data, class_i)</code>. הפעולה מזומנת בקובץ <code>project.py</code> על ידי הפעולה <code>option_1</code>.</p>
<code>show_Image(data, class_i)</code>	<p>הפעולה מקבלת תמונה ואת התווית המתאימה לה (המיוצגת בעזרת אינקס שמסמן את הכיתה אליה שייך פריט הלבוש בתמונה). הפעולה מציגה את התמונה בגודלה (28X28) בגווני אפור לצד סקלה של ערכי הפיקסלים (שנעים בין 0 ל-255 כאמור) ועם כותרת של פריט הלבוש המופיע בתמונה. הפעולה נקראת ב-<code>explore_Fashion_MNIST_data</code> וגם</p>

מזומנת בקובץ project.py על ידי הפעולה option_5.	
<p>הפעולה מקבלת את ארבעת המערכים המייצגים את הדתסאט ומעצבת מחדש את מערך הנתונים של התמונות כך שיכילו ערוץ צבע יחיד. בנוסף, הפעולה משתמשת בטכניקת one hot encoding עבור הכיתה של כל פריט, והופכת את המספר השלם לוקטור בינארי באמצעות הפונקציה to_categorical(). הפעולה מחזירה את ארבעת המערכים המעודכנים.</p> <p>הפעולה נקראת בקובץ project.py בפעולה הראשית.</p>	<pre>prepare_dataset(train_images, train_labels, test_images, test_labels)</pre>
<p>הפעולה מקבלת את מערך התמונות של ה-training set ואת מערך התמונות של ה-testing set ומנרמלת את ערכי הפיקסלים של התמונות בשני שלבים: תחילה הפעולה ממירה את סוג הנתונים ממספרים שלמים לא מכוונים לממשיים, ואז מחלקת אותם בערך הפיקסלים המרבי (כלומר ב-255).</p> <p>הפעולה נקראת בקובץ project.py בפעולה הראשית.</p>	<pre>scale_pixel_values(train_images, test_images)</pre>

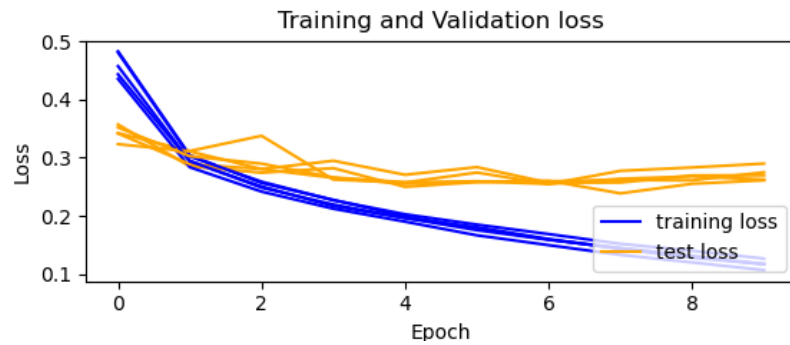
מסקנות הרצת המודל

מסקנות מהרצת המודל על ה- training data ושימוש ב- validation :

Learning curve of classification accuracy:



Learning curve of cross-entropy loss

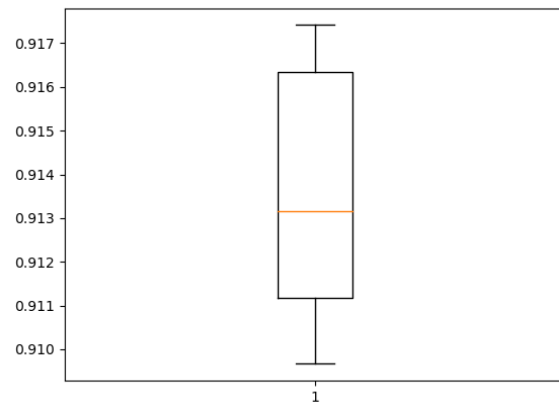


ערכי ה- accuracy עבור כל

k-fold cross-validation

```
> 91.317
> 91.742
> 91.633
> 90.967
> 91.117
```

```
Accuracy: mean=91.355 std=0.295, n=5
```



מסקנות מהרצת מודל שאומן על 80% מה- training data (ה-20% הנותרים שימשו ל- validation):

```
Test accuracy: 0.8970999717712402
Test loss: 0.3130113482475281
```

מסקנות מהרצת מודל שאומן על כל ה- training data :

```
Test accuracy: 0.911300003528595
Test loss: 0.2895078957080841
```

ניתוח התוצאות והסקת מסקנות :

- המודל פועל ביעילות של 91%, המשמעות היא שב-9% מהמקרים פריטי הלבוש לא יסווגו כראוי.
- כפי שניתן לראות בתרשים של classification accuracy, ה-accuracy עולה במהירות ב-epoch הראשון, דבר המצביע על כך שהרשת לומדת מהר. אולם העקומה ממשיכה לעלות ב-epochs הבאים לכן ה-epochs האלה נדרשים גם כדי להכשיר את המודל.
- ניתן לראות בתרשים model-loss שהעקומה של ה-training loss ממשיכה לרדת עד לסיום האימון ואינה משתטחת. הדבר רומז כי ייתכן שתהליך האימון נפסק בטרם עת והמודל מסוגל להמשיך ללמוד ולהשתפר.
- ניתן לראות בתרשים model-loss שהעקומה של ה-validation loss יורדת עד שהיא משתטחת. ייתכן כי אם היינו מגדילים את מספר ה-epochs, הפער בין ערכי ה-loss היה קטן יותר והיינו מגיעים להתאמה טובה יותר של המודל לפתרון הבעיה (נקרא good fitting).
- בנוסף, בהתחשב בכך ש-Test accuracy באלגוריתמי DL לסיווג פרטי לבוש נע לרוב בין 90% ל-95%, ניתן להגדיר את תוצאות הפרויקט כמוצלחות.

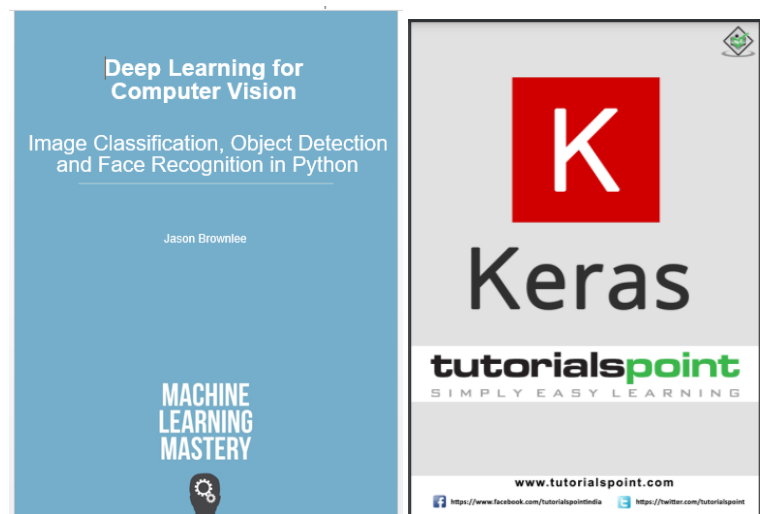
רפלקציה/סיכום אישי

תהליך העבודה היה ארוך, מורכב ומאתגר. במסגרת ההתמחות בנושא "למידה עמוקה" נחשפתי לראשונה לתחום הבינה המלאכותית בכלל ולתת התחום deep learning בפרט ובתחילת דרכי התקשתי להבין כיצד מכונה, דבר שנתפס בעיני כאוטומטי ותלתי במפתחים, מסוגלת לחקות חשיבה אנושית. השיעורים עם המורה דינה והמדריכים שקראתי באינטרנט עזרו לי להכיר את עקרונות התחום deep learning, ולהבין איך מלמדים מכונה בעזרת רשתות נוירונים מלאכותיות. כשהתבקשנו לבחור נושא לפרויקט המבוסס על DL חיפשתי נושא מעניין שיפתיע אותי שלא רק אדם מסוגל ללמוד אותו אלא גם מכונה. שני רעיונות שמצאתי היו חיזוי מסלול הוריקן וסיווג של סרטן השד. אולם נתקלתי במספר קשיים כגון ההמרה של הדתאסט ובניית המודל, הואיל והנושא היה חדש לי ונדרשתי לחקור ולהבין לבד את התחום. לכן החלטתי לגשת לבעיית סיווג של פריטי לבוש שבה הצלחתי להבין את הפתרון לבעיה ואת אופן מימושו ב-DL. במהלך כתיבת הפרויקט, בדקתי שני מודלים ב-machine learning ועוד מודל ב-deep learning לצורך בניית רשת המודל, אך לבסוף פסלתי אותם בשל accuracy נמוך ובעיית Overfitting קלה. המודל שמוצג בספר הזה זהו המודל הסופי שהניב את התוצאות הטובות ביותר. תהליך כתיבת התכנית היה לא פשוט אך גם מלמד ומעשיר. Deep Learning זהו תחום יוצא דופן ופורץ דרך בתחום ה-AI. בפרויקט זה הייתה לי ההזדמנות להתנסות בתחום ולהיווכח שמכונה אכן מסוגלת לדמות את מערכת הנוירונים במוח וללמוד לזהות פריטי לבוש שונים.

ביבליוגרפיה

נעזרתי באתרים/מדריכים/ספרים הבאים :

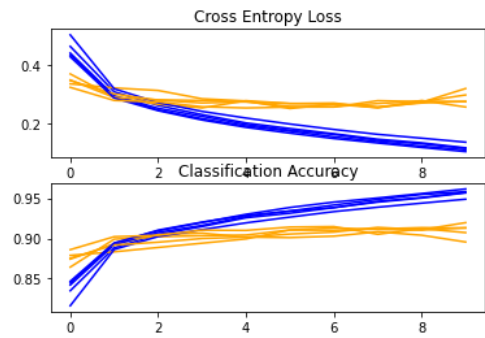
- [/https://www.upgrad.com/blog/basic-cnn-architecture](https://www.upgrad.com/blog/basic-cnn-architecture)
- <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>
- [/https://cs231n.github.io/convolutional-networks](https://cs231n.github.io/convolutional-networks)
- https://www.tensorflow.org/datasets/catalog/fashion_mnist
- <https://medium.com/analytics-vidhya/deep-learning-regularization-techniques-to-reduce-overfitting-e623c5900d97>



נספחים

91.292 ערכי ה training and
91.967 : validation accuracy
91.333
89.558
90.725

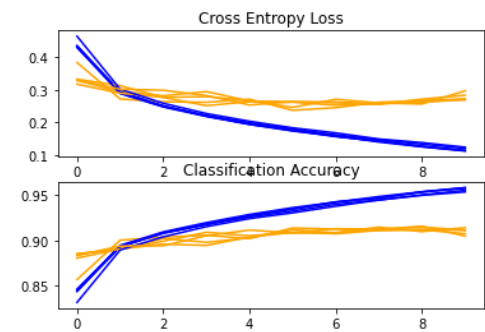
Accuracy: mean=90.975 std=0.810, n=5



1 Figure

> 90.742 ערכי ה training and
> 91.117 : validation accuracy
> 90.475
> 91.142
> 91.408

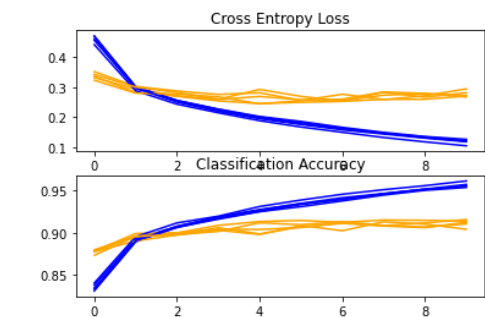
Accuracy: mean=90.977 std=0.329, n=5



2 Figure

> 91.258 ערכי ה training and
> 91.392 : validation accuracy
> 90.433
> 91.058
> 91.575

Accuracy: mean=91.143 std=0.393, n=5



3 Figure

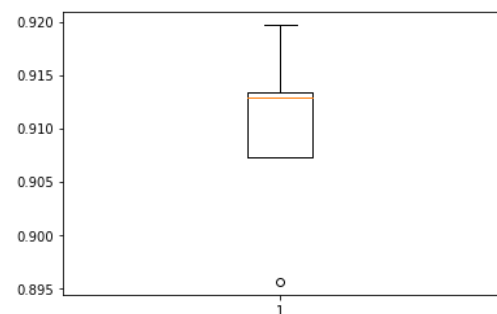


Figure 1

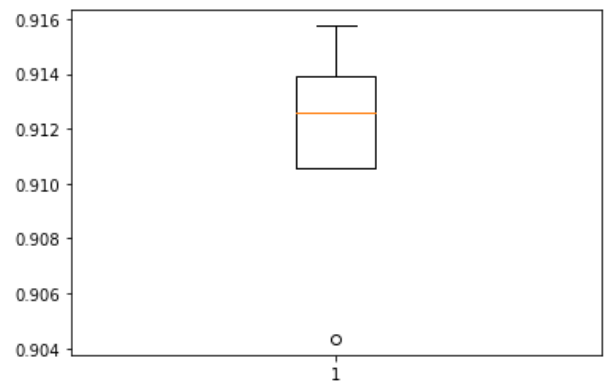


Figure 2

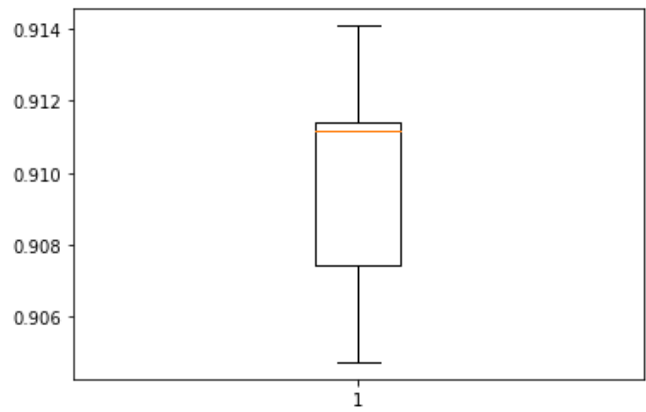


Figure 3