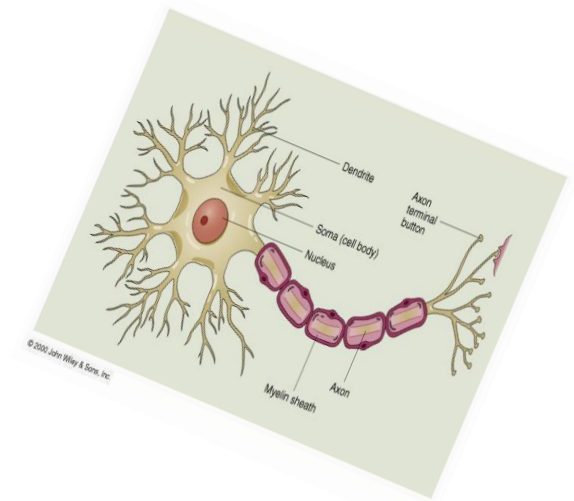
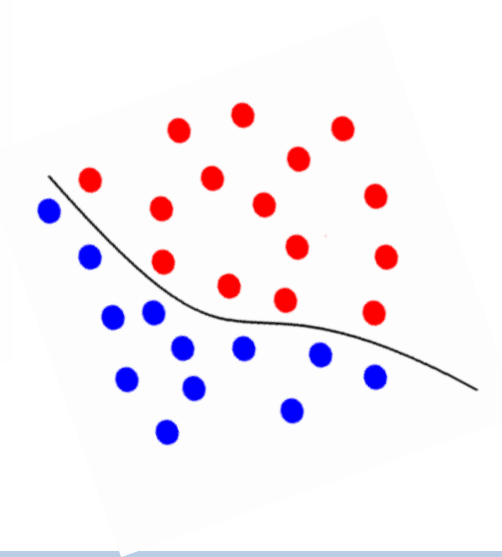
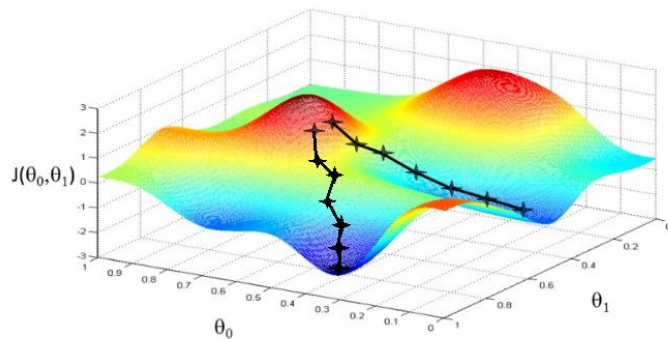




לא לשכוח להפעיל הקלטה!

קלסיפיקציה ברשתות נוירונים



מה אנחנו צריכים כבר לדעת?

- למידה מפוקחת
- רגרסיה לוגיסטית ופונקציית מעבר
- מבנה הרשת

✓ ייצוג של הערכים בקלט ובפלט

✓ אתחול מטריצות המשקולות - 3 שיטות:

□ אתחול לאפס

□ אתחול לערכים רנדומליים

□ He Initialization (מתאים לרשתות בהם קיימת ReLU כפונקציית אקטיבציה)

✓ פונקציות אקטיבציה

✓ היפר פרמטרים של רשת נוירונים

✓ תהליך אימון הרשת

✓ Overfit and underfit

✓ שיפור המודל

• רגרסיה – הערכת הפונקציה בין הקלט לפלט.

– הפלט הוא מספר ממשי (או כמה מספרים).

• קלסיפיקציה (סיווג) – ניבוי לאיזו קבוצה שייכת כל דוגמא.



– הפלט הוא התוית של הקבוצה.

ייצוג של הערכים בקלט ובפלט

• נתונים סטטיסטיים

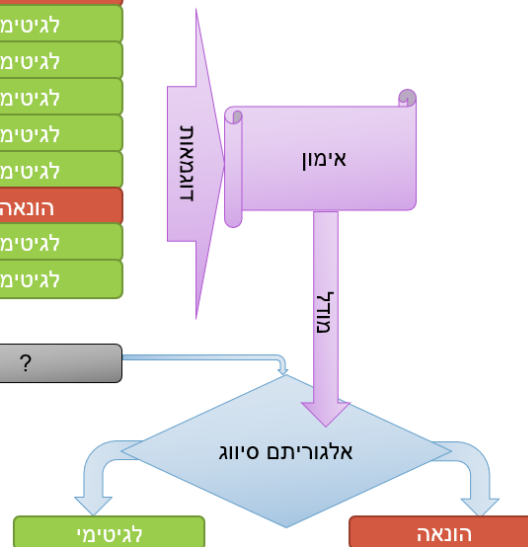
input: הפיצ'רים הינם ערכים מספריים

output: ערכים שמבטאים את הסוג. ניתן להבדיל בין שני סוגים ויותר.

classification

אברהם	מקרר	3000	8:00	לגיטימי
אברהם	מסטיק	5	9:45	הונאה
שרה	חנייה	40	10:02	לגיטימי
שרה	חשבון חשמל	3000	15:33	לגיטימי
שרה	ספה	3000	9:01	לגיטימי
יצחק	מכולת	25	18:45	לגיטימי
יצחק	שעון יד	300	23:23	לגיטימי
יצחק	מקרר	3000	8:00	הונאה
רבקה	טלוויזיה	6600	17:02	לגיטימי
רבקה	מזון לחתולים	30	02:30	לגיטימי

יעקב	קפה ומאפה	50	06:30	?
------	-----------	----	-------	---



קלסיפיקציה ברשתות נוירונים – ייצוג הפלט

- יהיו לנו בשכבת האאוטפוט מספר נוירונים כמספר המחלקות.
- נגדיר את ה-target כך שכל נוירון יגיב ב-1 עבור מחלקה אחת, וב-0 עבור כל המחלקות האחרות.

	1	1	1	0	0	0	0	0	0
Target	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	1	1	1
Input									



ברשת שאומנה לזהות צבעים: אדום, כחול
תכלת. כאשר נזין תמונה בצבע אדום נצפה
לקבל בפלט את הערכים הבאים:

	1	1	1	0	0	0	0	0	0
Target	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	1	1	1
Input									

.4	3.	2.	1.
1	1	0	0
	0	1	0
	0	0	0



ברשת שאומנה לזהות צבעים: אדום, כחול
תכלת. כאשר נזין תמונה בצבע ירוק נצפה
לקבל בפלט את הערכים הבאים:

	1	1	1	0	0	0	0	0	0
Target	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	1	1	1
Input									

.4	3.	2.	1.
0	1	0	0
	0	1	0
	0	0	0

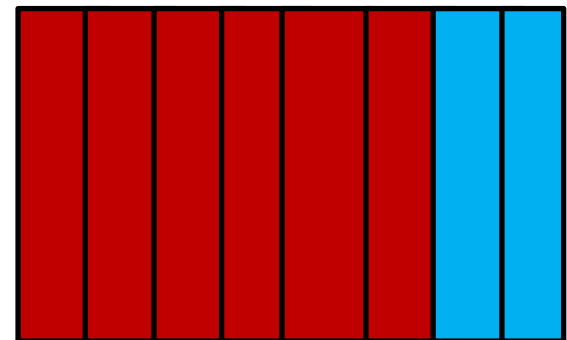
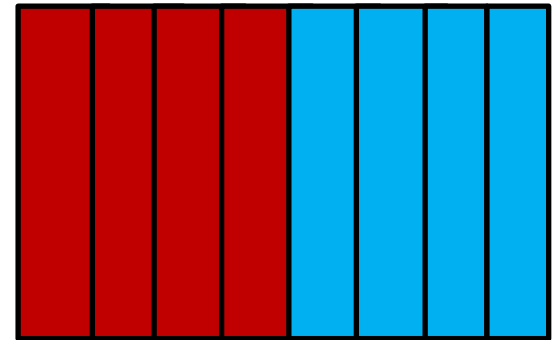


ברשת שאומנה לדהות צבעים: אדום, כחול
תכלת. כאשר נזין תמונה בצבע ירוק נצפה
לקבל בפלט את הערכים הבאים:

	1	1	1	0	0	0	0	0	0
Target	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	1	1	1
Input									

.4	3.	2.	1.
0	1	0	0
	0	1	0
	0	0	0

מספר הדוגמאות עבור כל מחלקה יהיה זהה.



קלסיפיקציה ברשתות נוירונים – ייצוג הפלט – שיטה 2

□ ה – output יהיה וקטור עם המימדים: $(m,)$.

m – מספר הדגימות

כל מחלקה תיוצג ע"י ערך שאופייני לה:

$$0 = \text{Iris-setosa} \circ$$
$$1 = \text{Iris-versicolor} \circ$$






2 = Iris-virginica ○

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
```

פונקציות מעבר

ייצוג הפלט בבעיות קלסיפיקציה – פונקציות מעבר

בשביל לפתור בעיות לא ליניאריות, בשכבות הביניים יהיו פונקציות מעבר לא ליניאריות (פונקציה לוגיסטית logsig או טנגנס היפרבולי tansig).

Name	Plot	Function, $f(x)$
Identity		x מתאים בשכבה אחרונה לבעיות רגרסיה
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$
Logistic, sigmoid, or soft step		$\sigma(x) = \frac{1}{1 + e^{-x}}$ מתאים בשכבה אחרונה לבעיות קלסיפיקציה
Hyperbolic tangent (tanh)		$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ מתאים בשכבה אחרונה לבעיות קלסיפיקציה +/-1
Rectified linear unit (ReLU) ^[7]		$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max\{0, x\} = x1_{x>0}$ מתאים בשכבות חבויות או בשכבה אחרונה בבעיות רגרסיה של ערכים חיוביים

https://en.wikipedia.org/wiki/Activation_function


פונקציות מעבר – פונקצית softmax

- אחד הפירושים של הפלט של נוירוני שכבת הפלט הוא ההסתברות של כל מחלקה.
- אולם אם נשתמש בפונקצית מעבר סיגמואידית, שום דבר לא מגביל את סכום ההסתברויות ל-1.
- דרך להתמודד עם בעיה זו היא להשתמש בפונקצית המעבר softmax:

$$y_i = \frac{e^{z_i}}{\sum_{j \in \text{group}} e^{z_j}}$$

- פונקצית השגיאה נקראת cross-entropy:

$$C = - \sum_j t_j \log y_j$$

target value

היפר פרמטרים

layers

hidden units

learning rates

activation functions

תרגיל כיתה 1 – כיוון היפר פרמטרים

הכנסו לאתר <https://playground.tensorflow.org/>

אמנו את הרשת ל-4 סוגי הבעיות, רשמו את ההיפר פרמטרים הטובים ביותר:

PROBLEM TYPE	מספר התכונות FEATURES	מספר השכבות	Learning rate	Activation
				
				
				
				

תהליך אימון הרשת

✓ ייבוא נתונים

✓ הכנת נתונים

▪ נורמליזציה

▪ חלוקה לסט אימון/ולידציה/בדיקה

✓ יצירת רשת מסוג קלסיפיקציה

✓ אימון הרשת עם נתוני הדגימות שלנו

✓ הערכת ביצועים על סט הבחינה:

▪ חישוב פונקציית השגיאה על סט הבחינה.

▪ חישוב אחוז ההצלחה על סט הבחינה.

✓ Troubleshooting - פתרון בעיות

אימון - Train

בדיקה - Test

1. Import Data

מבוסס על פרויקט לדוגמא:
7-2-NN-CL-1.ipynb

```
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target
```

```
[3] print(X.shape, y.shape)
```

```
(150, 4) (150,)
```

```
[4] print(y)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

2. הכנת הנתונים

נירמול הנתונים

```
scaler = StandardScaler()  
X_scaled = scaler.fit_transform(X)
```

חלוקת סט הנתונים ל-3 חלקים

```
# train - 70%, temp - 30%  
X_train, X_temp, y_train, y_temp = train_test_split(X_scaled, y, test_size=0.3, random_state=42)  
# takes the temporary set and splits it equally into validation and test sets  
# train - 70%, val 15%, test 15%  
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
```

שימו לב: על מנת לחלק את הנתונים ל 3 חלקים יש להריץ את פקודת החלוקה

פעמיים

- חלוקה train+val, test
- ואז חלוקה של train+val ל train, val

2. הכנת הנתונים – חלוקת הנתונים ל 3 חלקים

Training Dataset

מדגם הנתונים
המשמש לאימון
המודל

תוצאה שלב זה:
סט משקלות
וערכי הטייה

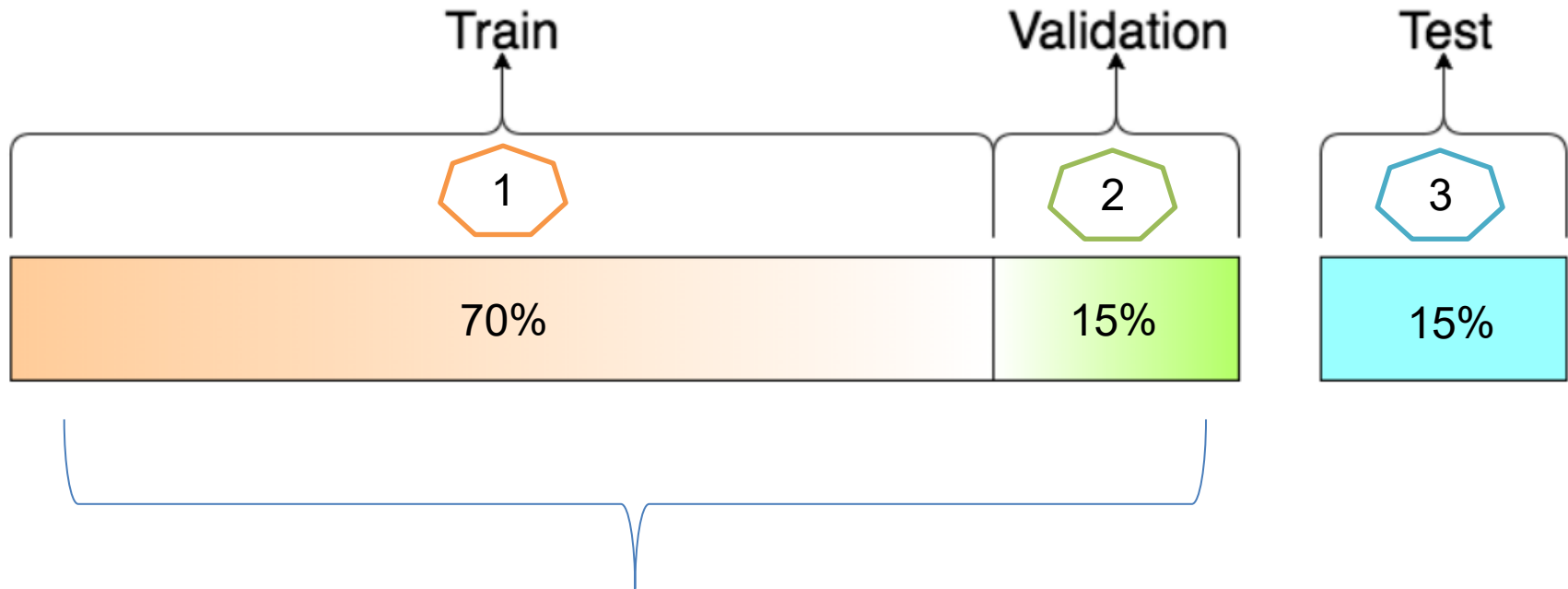
Validation Dataset

מדגם הנתונים
משמש לבחינת
המודל המאומן
ובמידת הצורך
שינוי המודל ע"י:
הוספת שכבות,
הוספת ניורונים
בכל שכבה, שינוי
מספר ה
features

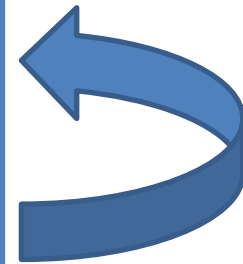
Test Dataset

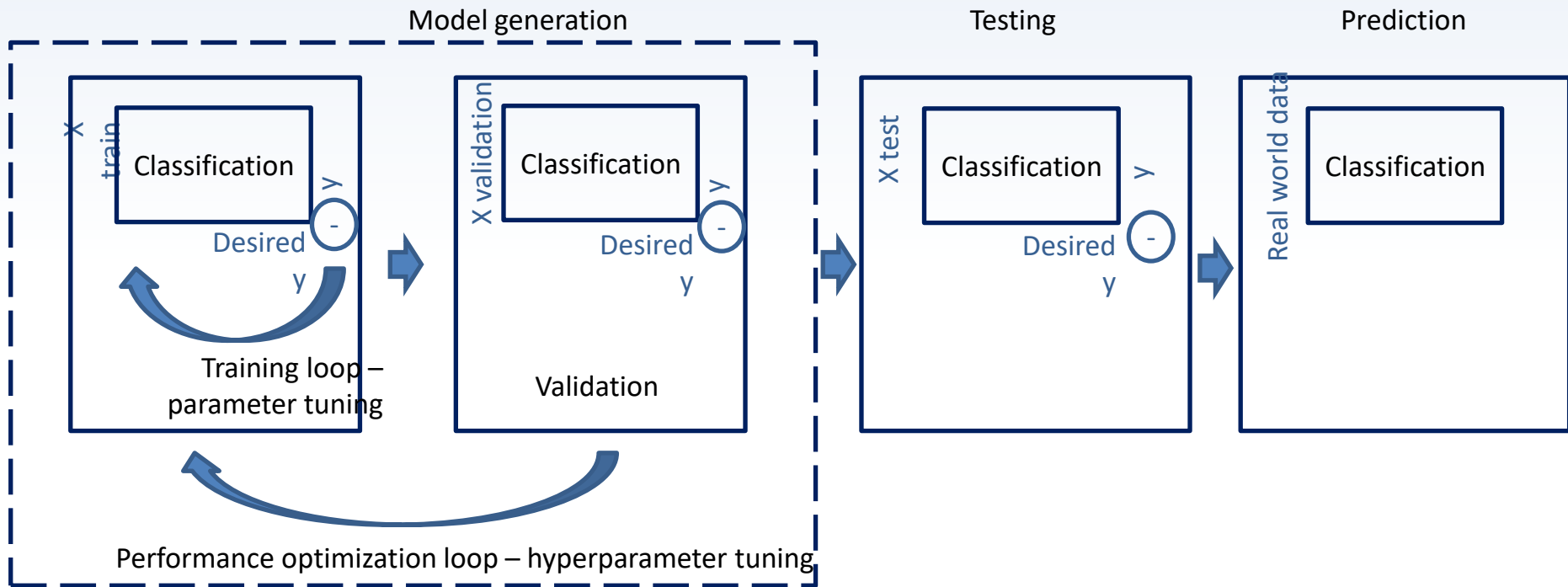
מדגם נתונים
שמספק את
האישור הסופי
לבחירת המודל
המאומן לאחר
שאומת עם סט
הולידציה

4. הכנת הנתונים – חלוקת הנתונים ל 3 חלקים



1. אימון באמצעות סט האימון לקבלת המודל
2. אימות המודל באמצעות סט הולידציה
 - 2.1 אם תקין – ממשיכים לשלב הבא
 - 2.2 אם לא תקין – כיוון היפר-פרמטרים וחוזרים לשלב 1
3. בחינת המודל הסופי באמצעות סט הבחינה





הגדרת מבנה הרשת: מספר השכבות וכמה ניורונים בכל שכבה

הגדרת פונקציית האקטיבציה בכל אחת מהשכבות

```
model1 = keras.Sequential([  
    layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),  
    layers.Dense(64, activation='relu'),  
    layers.Dense(3, activation='softmax')  
])
```

מה מבנה הרשת:

- א. כמה נוירונים בשכבת הקלט
- ב. כמה שכבות חביות?
- ג. כמה נוירונים בכל שכבה חבויה
- ד. כמה נוירונים בשכבת הפלט

מה מבנה הרשת:

- א. כמה נוירונים בשכבת הקלט - 4
- ב. כמה שכבות חביות? - 2
- ג. כמה נוירונים בכל שכבה חבויה - 64 ו 64
- ד. כמה נוירונים בשכבת הפלט - 3

4. אימון הרשת עם נתוני הדגימות שלנו

```
model1.compile(optimizer='adam',  
               loss='sparse_categorical_crossentropy',  
               metrics=['accuracy'])  
  
history1 = model1.fit(X_train, y_train, epochs=20, validation_data=(X_val, y_val))
```



4. אימון הרשת עם נתוני הדגימות שלנו – מאפיינים לאחר ההרצה

```
i=0
for layer in model1.layers:
    i=i+1
    weights, biases = layer.get_weights()
    print("w", str(i), weights.shape, ";b", str(i), biases.shape)
```

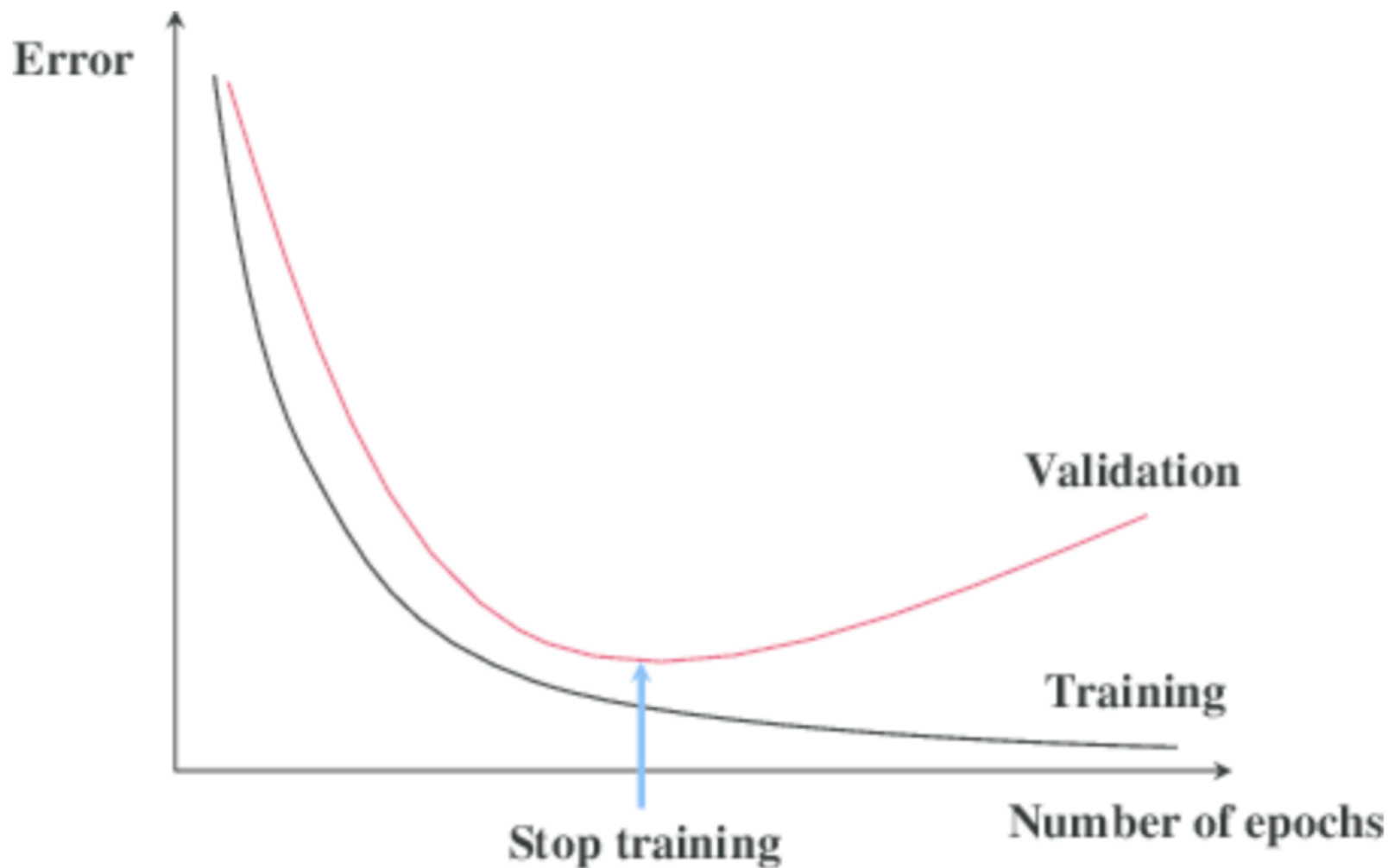
משקולות וערכי הספים בכל השכבות:

```
w 1 (4, 64) ;b 1 (64,)
w 2 (64, 64) ;b 2 (64,)
w 3 (64, 3) ;b 3 (3,)
```

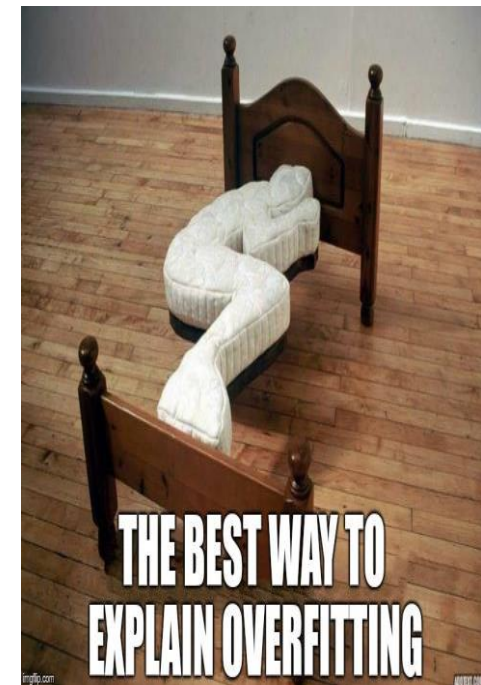
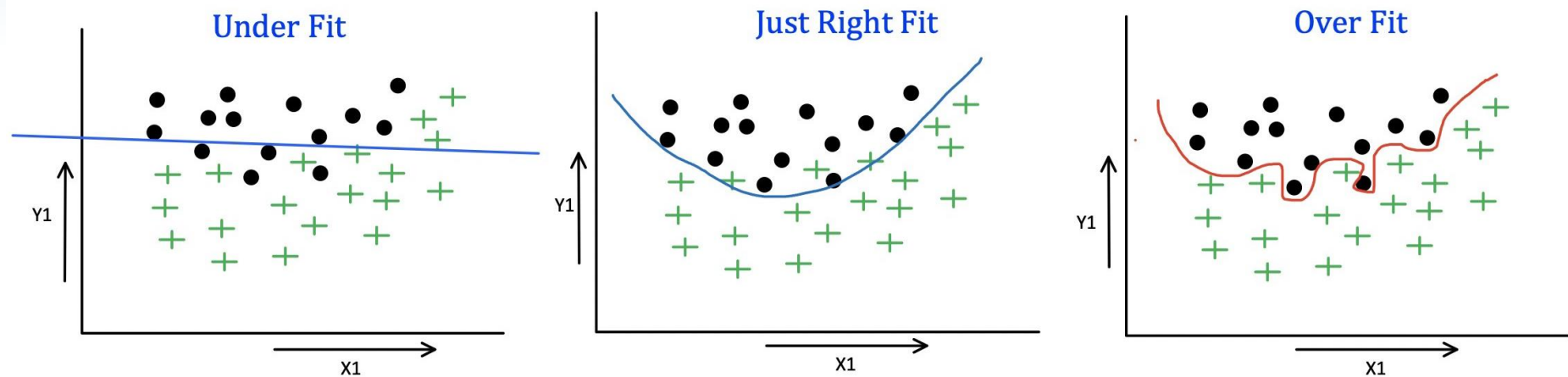
דיוק סט האימון בכל האיטרציות: `history1.history['accuracy']`

הטעות סט האימון בכל האיטרציות: `history1.history['loss']`

4. אימון הרשת עם נתוני הדגימות שלנו – Over fitting / Under fitting



4. אימון הרשת עם נתוני הדגימות שלנו – Over fitting / Under fitting



מעל"ה - מדע חישובי פיזיקה

4. אימון הרשת עם נתוני הדגימות שלנו – Under fitting /Over fitting

Over fitting התאמת יתר

Under fitting תת התאמה

תיאור הבעיה

המודל שלנו מתאים מצוין לנתוני האימון אך מתקשה לסווג נתונים חדשים
כישלון בהכללה

המודל שלנו אינו מתאים לנתוני האימון וגם מתקשה לסווג נתונים חדשים

מאפייני ערך הטעות

טעות סט האימון נמוכה
טעות סט הולידציה גבוהה

טעות סט האימון גבוהה
טעות סט הולידציה גבוהה

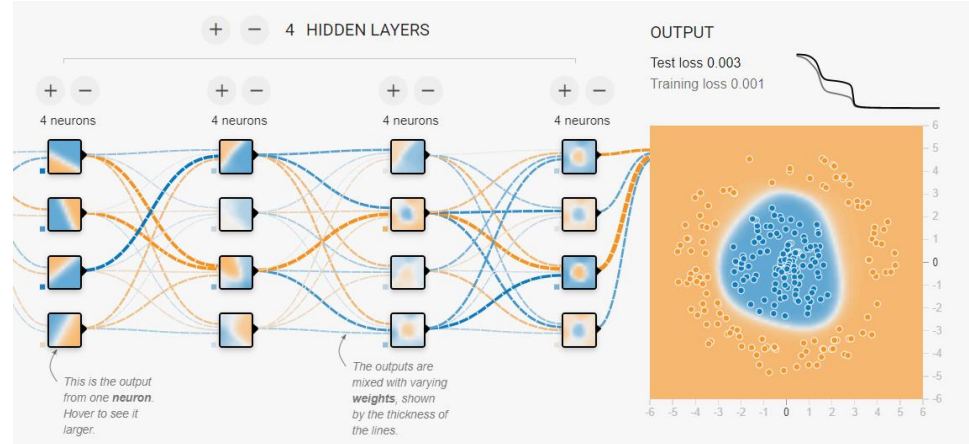
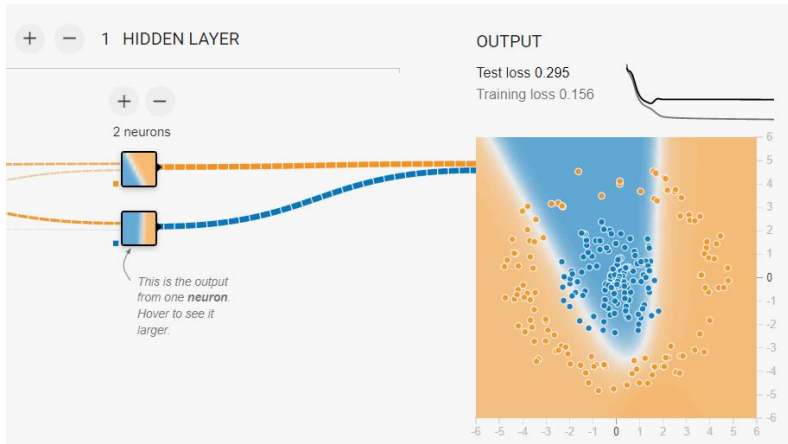
איך אני נמנע מהבעיה הזו?

1. יש צורך להגדיל מספר דוגמאות
2. צמצום פיצ'רים (תכונות)
3. צמצום מורכבות המודל (להקטין את מספר הניורונים).
4. עצירה מוקדמת בשלב האימון **Early stopping** (ברירת המחדל בtoolbox)
5. Regularization

1. אין צורך להגדיל מספר דוגמאות
2. הוספת פיצ'רים (תכונות)
2. העלאת מורכבות המודל (להגדיל את מספר הניורונים).

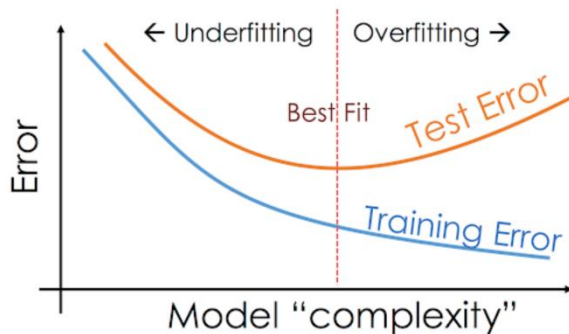
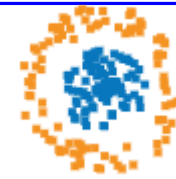
תרגיל כיתה 2 – כיוון היפר פרמטרים Under fitting /Over fitting

Generalization Vs. Complexity



<https://playground.tensorflow.org/> הכנסו לאתר

עבור הבעיה , מה הם היפר-פרמטרים במצב של overfitting ו-underfitting.



5. הערכת ביצועי הרשת – מדדי הערכה

מציגה את הסיווג הנכון והלא נכון בצורה גרפית וכמטריצה

```
y_pred = np.argmax(model1.predict(X_test), axis=1)
conf_matrix = confusion_matrix(y_test, y_pred)
print(conf_matrix)
```

1/1 [=====] - 0s 230ms/step

[[6 0 0]
[0 5 5]
[0 0 7]]

real

Predicted

מתוך 6 הדוגמאות של פרח האירוס מסוג 1 כמה זוהו כפרח

אירוס מסוג 1?

real

6	0	0
0	5	5
0	0	7

Predicted

א. 5

ב. 10

ג. 6

ד. 7

מתוך 6 הדוגמאות של פרח האירוס מסוג 1 כמה זוהו כפרח

אירוס מסוג 1?

real

6	0	0
0	5	5
0	0	7

Predicted

א. 5

ב. 10

ג. 6

ד. 7

מתוך 10 הדוגמאות של פרח האירוס מסוג 2 כמה זוהו כפרח

אירוס מסוג 2?

real

[6	0	0]
[0	5	5]
[0	0	7]]

Predicted

א. 5

ב. 10

ג. 6

ד. 7

מתוך 10 הדוגמאות של פרח האירוס מסוג 2 כמה זוהו כפרח

אירוס מסוג 2?

real

[6	0	0]
[0	5	5]
[0	0	7]]

Predicted

א. 5

ב. 10

ג. 6

ד. 7

5. הערכת ביצועי הרשת – מדדי הערכה

חישוב מדדי הערכה: precision, recall, f1_score

```
TP_0 = conf_matrix[0,0]
TN_0 = conf_matrix[1,1] + conf_matrix[2,2] + conf_matrix[1,2] + conf_matrix[2,1]
FP_0 = conf_matrix[1,0] + conf_matrix[2,0]
FN_0 = conf_matrix[0,1] + conf_matrix[0,2]
```

```
#Calculate Precision, Recall, F1 Score for each category
Precision = np.zeros(3)
Recall = np.zeros(3)
F1_Score = np.zeros(3)

# YOUR CODE HERE - START
# -- 0 --
Precision[0] = TP_0/(TP_0+FP_0)
Recall[0] = TP_0/(TP_0+FN_0)
F1_Score[0] = 2*(Precision[0]*Recall[0]) / (Precision[0]+Recall[0])
```

חישוב המדדים
לקטגוריה 0

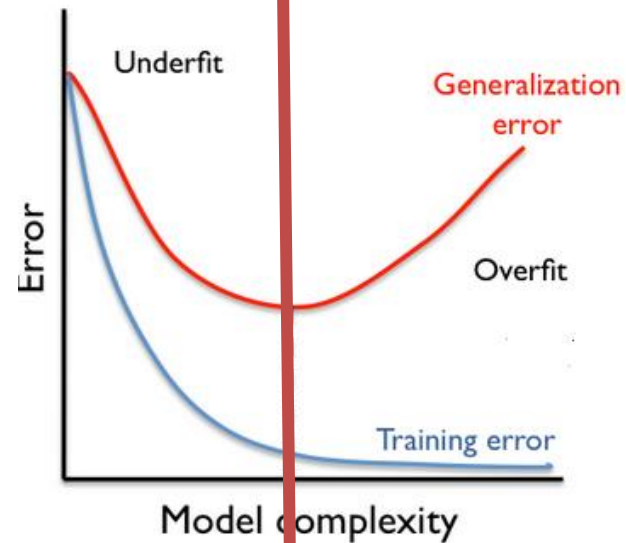
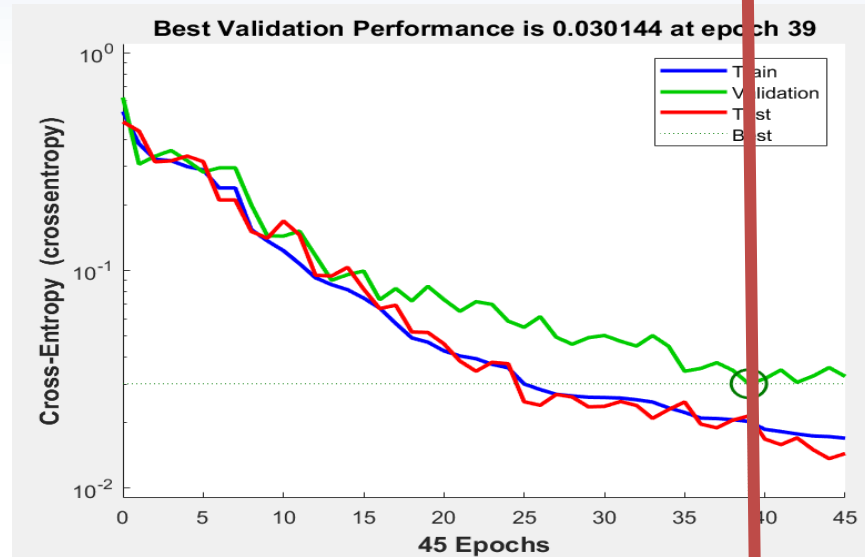
שינויים שניתן לעשות לשיפור המודל

- נתונים נוספים: לפעמים, המודל צריך עוד דוגמאות ללמוד מהן.
- איזון בנתונים: ייצוג שווה של דגימות מכל סוג
- שינוי ארכיטקטורת הרשת: הוספה או הפחתת שכבות / נוירונים בכל שכבה.
- שינוי קצב למידה, סוג האופטימיזציה, גודל ה batch, פונקציות אקטיבציה שונות.
- שינוי מספר האיטרציות וטכניקת early-stopping.

עצירה מוקדמת בשלב האימון

- הנתונים מחולקים ל 3 חלקים:
 - Training set
 - Validation set
 - Test set
- האימון יעשה רק על סט האימון, והבחינה הסופית תעשה רק על סט הבחינה.
- בתהליך האימון, נבדוק לא רק את שגיאת האימון, אלא גם את השגיאה על סט הולידציה.
- אם השגיאה על סט הולידציה עלתה \times פעמים רצוף, נעצור את אימון הרשת, ונשמור את הנתונים (את המשקלות והסיפים) שהיו לפני העלייה.

6. שיפור המודל – Early stopping



תרגיל – הוספת מנגנון early-stopping