# שכבות

אבן הבניין הבסיסית של רשת נוירונים היא השכבה. רוב הלמידה העמוקה מורכבת משרשור שכבות.

### tf.keras.layers.Flatten

הופך את הקלט ממערך רב ממדי למערך חד ממדי. לשכבה זו אין פרמטרים ללמוד, אלא רק מעצבת מחדש את הנתונים.

### tf.keras.layers.Dense

מייצגת שכבת fully connected. מקבלת את מספר הנוירונים בפלט של שכבה זו, פונקציית אקטיבציה ופרמטרים נוספים.

מודל

כדי ליצור מודל מאומן ב-Keras שניתן להריץ עליו אובייקטים ולקבל חיזוי, יש לדעת לבצע חמש פעולות:

1. יצירה – הגדרת רצף השכבות במודל.
2. קמפול – הגדרת פונקציית loss, אופטימייזר ומה רוצים למקסם.
3. אימון – נאמן המודל באמצעות הדאטה.
4. הערכה – באמצעות השוואה לקבוצת ה-test.
5. חיזוי – נתינת אובייקט חדש וקבלת חיזוי.

## יצירת מודל

כדי ליצור מודל שהוא שרשור של שכבות נשתמש בפונקציה tf.keras.Sequential המקבלת מערך של שכבות ומחזירה מודל.

model = tf.keras.Sequential([

tf.keras.layers.Flatten(input\_shape=(28, 28)),

tf.keras.layers.Dense(128, activation='relu'),

tf.keras.layers.Dense(10)

])

## קמפול

באמצעות הפעלת הפונקציה compile על המודל ניתן להגדיר למודל את הרכיבים הבאים:

* פונקציית loss - מודד עד כמה המודל מדויק במהלך האימון. אנו רוצים מינימום של הפונקציה הזו כדי לנוו" את המודל בכיוון הנכון.
* אופטימייזר - באיזה שיטה לעדכן את הפרמטרים של המודל על סמך הנתונים שהוא רואה פונקציית ה-loss.
* מדדים - רשימת מדדים להערכת המודל במהלך אימון ובדיקות. בדרך accuracy.

model.compile(optimizer='adam',

loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

metrics=['accuracy'])

## אימון

באמצעות הפעלת הפונקציה fit נאמן את המודל. פונקציה זו צריכה לקבל את:

x – קבוצת המידע עליו המודל מתאמן.

y – התיוג של קבוצת האימון.

batch\_size - מספר שלם או None. מייצג את מספר הדגימות בכל עדכון של הגרדיאנט. אם לא צוין, ברירת המחדל 32.

epochs – מספר האיטרציות שעוברים על כל הדאטה כדי לאמן את במודל.

model.fit(train\_images, train\_labels, epochs=10)

## הערכה

באמצעות הפונקציה evaluate על המודל נוכל להעריך אותו. הפונקציה מקבלת את קבוצת ה-test ואת התיוג שלה, ומחזירה את הערך הסופי של הפונקציית loss ואת המדדים שהכנסנו למודל במהלך הקמפול.

test\_loss, test\_acc = model.evaluate(test\_images, test\_labels)

## חיזוי

נפעיל את הפונקציה predict על המודל כדי לקבל חיזוי עבור קבוצת אובייקטים. הפונקציה מקבלת מערך של אובייקטים x ומחזירה מערך numpy של החיזויים לכל הקלטים.

במקרה שהמודל צריך לסווג אובייקט ליותר משני מחלקות הוא מחזיר logits לכל מחלקה, ולכן יש להפעיל שכבת softmax כדי להמיר את ה-logits להסתברויות.

probability\_model = tf.keras.Sequential([model,

tf.keras.layers.Softmax()])

predictions = probability\_model.predict(test\_images)

np.argmax(predictions[0])