רשת נוירונים
ANN

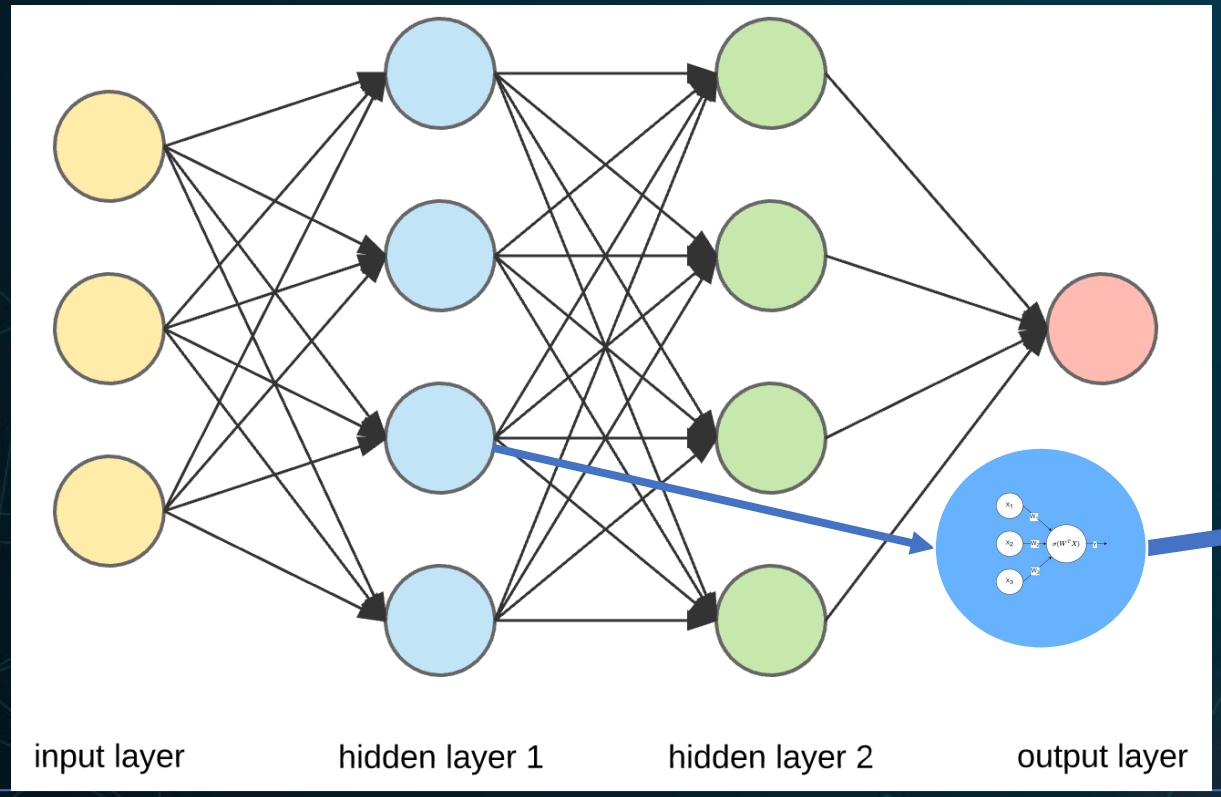
גלעד מרקמן

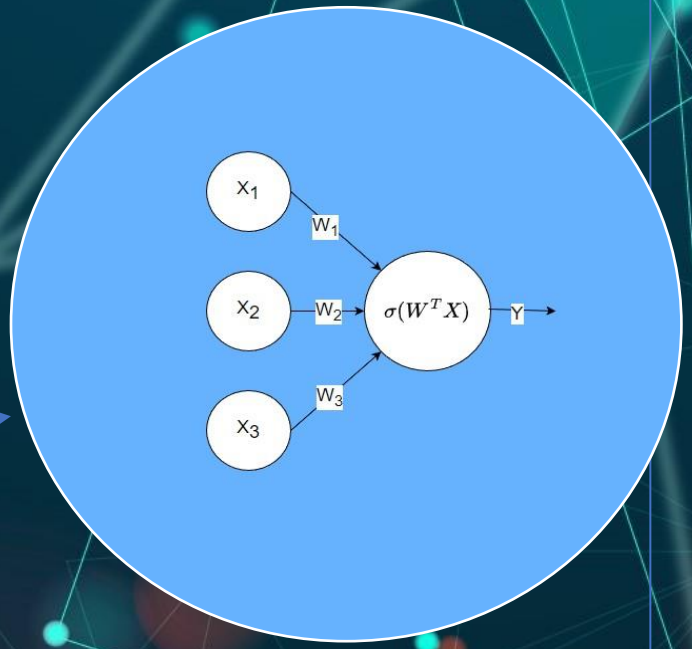
קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

רשת נוירונים - ANN

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- ANN – Artificial Neural Network הינה רשת של Perceptron's המחוברים אחד לשני. הכניסה לכל perceptron היא תוצאת החישוב של perceptron אחר.

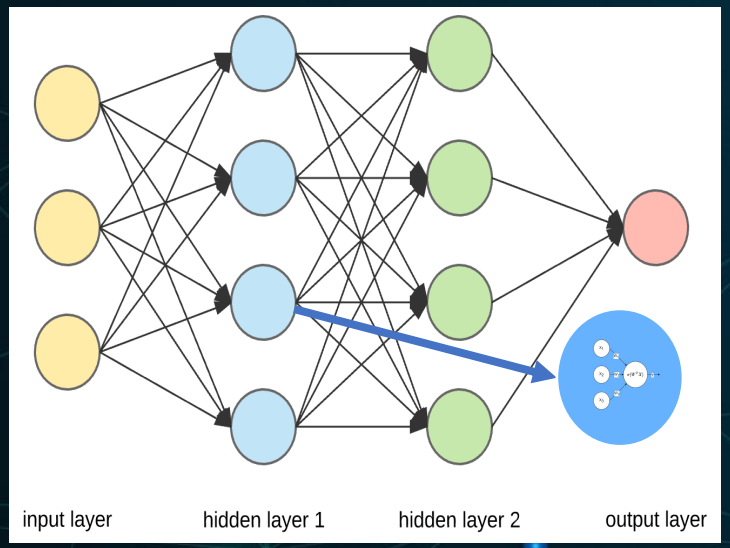




רשת נוירונים ANN

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- רשת נוירונים שהיא Fully Connected:
  - תוצאת החישוב של כל נוירון "משוכפלת" כמספר הנוירונים בשכבה הבאה.
  - כלל תוצאות החישוב מהוות את הקלט של הנוירון בשכבה הבאה.
- מחלקים את הרשת לשכבות עיקריות הבאות:
  - שכבת הקלט input.
  - שכבות נסתרות Hidden layer.
  - שכבת הפלט – output.

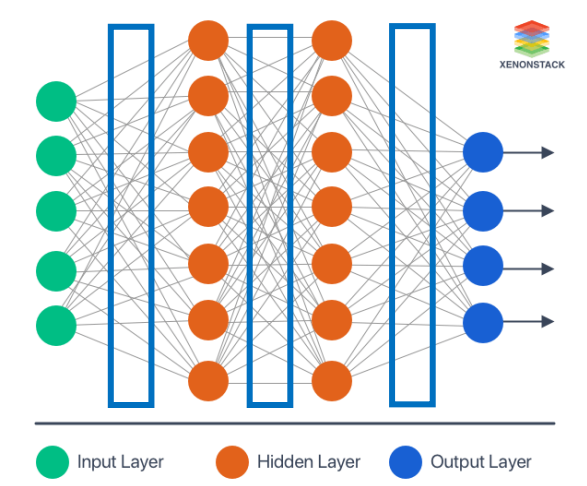


- שכבת הפלט נקבעת לפי השאלה אם למשל מבקשים תשובה שהיא בוליאנית (כן / לא) הפלט יכלול צומת אחת.

הגדרת רשת נוירונים ב PyTorch

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- ניתן להגדיר מודל של רשת נוירונים באמצעות המחלקה nn.Sequential.
  - כל שכבה מורכבת מפונקציה לינארית + פונקציית אקטיבציה.
  - מספר המשתנים ביציאה משכבה מסויימת שווה למספר המשתנים בכניסה לשכבה הבאה אחריה.
  - פונקציית המחיר נקבעת לפי פונקציית האקטיבציה בשכבת הפלט (האחרונה).
- אין שינוי באימון המודל לעומת המודלים הקודמים שלנו.




```
Model = nn.Sequential(
    nn.Linear(input_size,hidden_size_1,device='cuda'),
    nn.ReLU(),
    nn.Linear(hidden_size_1,hidden_size_2,device='cuda'),
    nn.ReLU(),
    nn.Linear(hidden_size_2, 4, device='cuda'),
    nn.Sigmoid()
)
```

מבנה רשת נוירונים

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

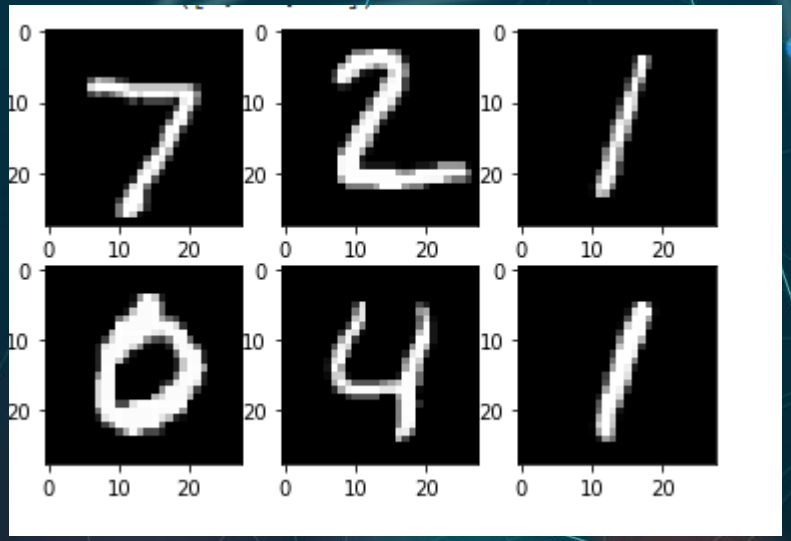
- אין כללים באשר למספר השכבות ומספר הצמתים בכל שכבה. המדובר בניסוי וטעיה.
- כלל האצבע הוא כי ככל והשאלה סבוכה יותר וכוללת יותר פרמטרים נצטרך להשתמש ביותר נוירונים ושכבות רבות יותר.
- פונקציית האקטיבציה המקובלת בתוך הרשת בין השכבות הפנימיות הינה ReLU או LeakyRelu.
- פונקציית האקטיבציה בשכבת הפלט תלויה בשאלה אותה המודל מתבקש לפתור:
  - שאלה בינארית (כן / לא): sigmoid או tanh, ופונקציית מחיר של BCE
  - שאלה עם מספר תשובות בדידות: softmax, ופונקציית מחיר של Cross Entropy.
  - שאלה עם תשובה מספרית נשתמש ב: sigmoid, tanh ואולי אף בפונקציה לינארית ללא פונקציית אקטיבציה. פונקציית המחיר היא MSE.

דוגמה - זיהוי ספרה בודדת MNIST

- נבנה רשת נוירונים אשר תזהה ספרה הכתובה בכתב יד, באמצעות בסיס נתונים ידוע המכונה MNIST וכולל 70,000 דוגמאות של ספרות בכתב יד.
- התמונות הינן בגודל 28*28 פיקסלים בשחור לבן, ולכל תמונה מוצמדת תווית עם המספר.
- בשלב ראשון נבחר ספרה מסויימת מראש – והמטרה לבנות מודל שיזהה האם זוהי הספרה הנבחרת אם לאו – מודל בינארי.

גוון הפיקסל

```
7 torch.Size([1, 28, 28])
2 torch.Size([1, 28, 28])
1 torch.Size([1, 28, 28])
0 torch.Size([1, 28, 28])
4 torch.Size([1, 28, 28])
1 torch.Size([1, 28, 28])
```



כיצד נזין תמונה למודל

- רשת נוירונים כוללת שכבה (וקטור) של קלט. לעומת זאת, כל תמונה בנויה מטנסור בעל שלושה מימדים (1,28,28). כיצד נפתור את הבעיה?

- הפתרון פשוט – "נשטח" את התמונה לוקטור אחד באורך 28*28 תאים, סה"כ 784 פרמטרים בכל שורה של דוגמה.

- הטבלה שתכנס אל המודל (רשת הנוירונים) תהיה טבלה של 50 שורות (batch_size = 50), כשבכל שורה יש 784 פרמטרים (נתונים) כמספר הפיקסלים.

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

Torchvision & DataLoader

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- ספריית pyTorch כוללת את הספריה Torchvision הכוללת בסיסי נתונים של תמונות לתרגול לימוד מכונה, וביניהם בסיס הנתונים MNIST.
- המחלקה torchvision.dataset.MNIST – טוענת את בסיס הנתונים למחשב. הנתונים מורדים לדיסק של המחשב אם אינם קיימים בו, ונטענים לזכרון של המחשב מן הדיסק.
- המחלקה torch.utils.data.DataLoader – מסייעת לנו בעבודה עם בסיסי נתונים גדולים, ומאפשרת לנו חלוקה לקבוצות batches.

חלוקה לקבוצות Batches

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- אימון רשת נוירונים כוללת שלושה שלבים:
  - חלחול קדימה וחישוב הערכים הצפויים.
  - חישוב ההפסד וחלחול לאחור למציאת גרדיאנים.
  - עדכון פרמטרים.
- ראינו שאנחנו יכולים לבצע חישוב על מספר דוגמאות (טבלה) בבת אחת.
- אולם, כשיש כמות גדולה של נתונים אין זה יעיל לבצע את האימון בכל איטרציה על כל הנתונים, ולהמתין עם עדכון הפרמטרים רק לאחר חישוב ההפסד של כל הדוגמאות. עדיף לחלק את הנתונים לקבוצות (batches), ולבצע את האימון על כל קבוצה בנפרד, ובסיום כל קבוצה לעדכן את הפרמטרים.
- בסיום ניתן לחזור ולבצע שוב את האימון על כל הקבוצות.
- לצורך אימון בקבוצות נעזר במחלקה DataLoader.

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

בניית מודל לזיהוי ספרה בכתב יד

MNIST

יבוא ספריות והגדרת Device

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- נייבא את הספריות הנדרשות.
- כמו כן, נגדיר Device אשר יאפשר לנו להשתמש ב GPU לחישובים מהירים על טנסורים.

```
[ ]  import torch
     import torch.nn as nn
     import torch.nn.functional as F
     import numpy as np
     import matplotlib.pyplot as plt
     import torchvision
     import torchvision.transforms as transforms
```

Device Configuration

```
[ ]  if torch.cuda.is_available:
       device = torch.device('cuda')
     else:
       device = torch.device('cpu')
```

- cuda – שימוש ב GPU
- Cpu – שימוש במעבד הרגיל.

טעינת בסיס הנתונים בקבוצות של 50 תמונות

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- מחלקים את הדוגמאות לשתי קבוצות: קבוצה לאימון המודל (train_dataset) וקבוצה לבדיקת המודל (test_dataset).
- כל קבוצה מחולקת ל batches של 50 תמונות.
- החלוקה של קבוצת האימון הינה ראנדומלית shuffle = True.
- תוך כדי הטעינה מבצעים המרה ל Tensors.

```
[ ]  batch_size = 50

     train_dataset = torchvision.datasets.MNIST(root='./data',
                                                train=True,
                                                transform=transforms.ToTensor(),
                                                download=True)

     test_dataset = torchvision.datasets.MNIST(root='./data',
                                               train=False,
                                               transform=transforms.ToTensor())

     # Data loader
     train_loader = torch.utils.data.DataLoader(dataset=train_dataset,
                                                batch_size=batch_size,
                                                shuffle=True)

     test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                               batch_size=batch_size,
                                               shuffle=False)
```

```
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz to ./data/MNIST/raw/train-images-idx3-ubyte.gz
100% 9912422/9912422 [00:00<00:00, 191661148.17it/s]
Extracting ./data/MNIST/raw/train-images-idx3-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz to ./data/MNIST/raw/train-labels-idx1-ubyte.gz
100% 28881/28881 [00:00<00:00, 1295070.28it/s]
Extracting ./data/MNIST/raw/train-labels-idx1-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz to ./data/MNIST/raw/t10k-images-idx3-ubyte.gz
100% 1648877/1648877 [00:00<00:00, 49020019.40it/s]
Extracting ./data/MNIST/raw/t10k-images-idx3-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz to ./data/MNIST/raw/t10k-labels-idx1-ubyte.gz
100% 4542/4542 [00:00<00:00, 104241.38it/s]
Extracting ./data/MNIST/raw/t10k-labels-idx1-ubyte.gz to ./data/MNIST/raw
```

מבנה הנתונים

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

- מבנה הנתונים שלנו כולל 50 קבוצות (batches) שכל קבוצה כוללת 50 דוגמאות (תמונות + תווית של התמונה).

- train_loader כולל אם כך 50*1200 = 60,000 דוגמאות.
- test_loader כולל 50*200 = 10,000 דוגמאות.

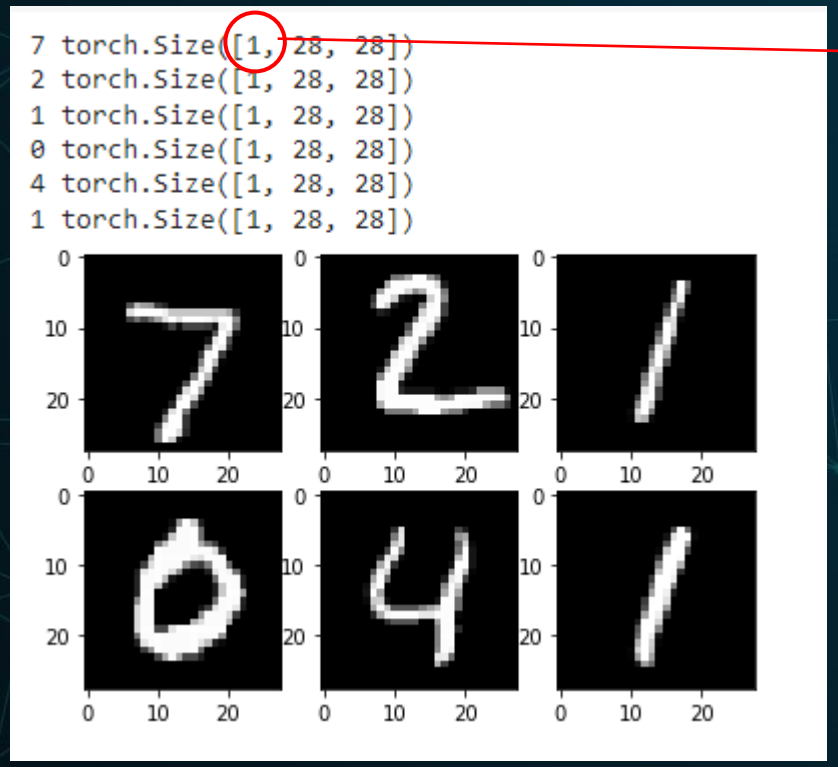
```
print(len(train_loader))
print(len(test_loader))
```

```
1200
200
```

- למבנה כל דוגמה ראה בשקף הבא.

הדפסת דוגמה

- ניתן לראות שכל דוגמה כוללת:
  - Example_Data – התמונה עצמה. מטריצה של 28*28 פיקסלים שכל פיקסל כולל מספר בין 0-1 המתאר את הגוון של האפור.
  - Example_target – הספרה שבתמונה.
- required_label – זוהי הספרה אותה אנחנו מבקשים לזהות (במקרה זה היא 7).



גוון הפיקסל

```
examples = iter(test_loader)
example_data, example_targets = next(examples)

for i in range(6):
    plt.subplot(2,3,i+1)
    plt.imshow(example_data[i][0], cmap='gray')
    print(example_targets[i].item(),end = " ")
plt.show()

required_label = 7
```

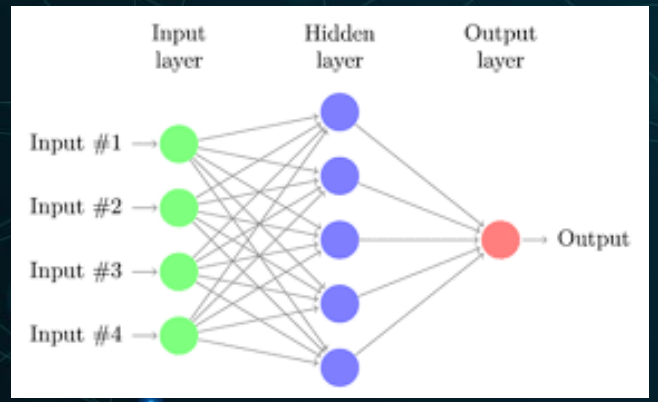
הגדרת מודל פרמטרים

Hyper-parameters

```
[23] input_size = 784 # 28x28
     hidden_size = 100
     num_classes = 10
     epochs = 2
     learning_rate = 0.01
```

Model

```
[24] Model = nn.Sequential(
         nn.Linear(input_size,hidden_size,device=device),
         nn.ReLU(),
         nn.Linear(hidden_size, 1, device=device),
         nn.Sigmoid()
     )
```



- שכבת ה input כוללת 784 צמתים, כמספר המאפיינים (העמודות) בטבלה.
- כל נוירון מקבל 784 ערכים.
- סה"כ יש 100 נוירונים בשכבה הראשונה.
- בסוף יש נוירון אחד שמוציא את התשובה (כן / לא המספר 7).
- פונקציית האקטיבציה היא Relu בין השכבה הראשונה לשכבת הפלט.
- פונקציית האקטיבציה של שכבת הפלט היא sigmoid, כיוון שהתשובה שלנו היא לוגית.

פונקציית המחיר והעדכון

- פונקציית המחיר היא Binary Cross Entropy המתאימה לתוצאות לוגיות (אמת או שקר).
- אם התוצאה תהיה גדולה מ- 0.5 התשובה היא אמת (הספרה 7), אחרת התשובה היא שקר (הספרה אינה 7).

loss and optimizer

```
Loss = nn.BCELoss()

# init optimizer

optim = torch.optim.Adam(Model.parameters(), lr=learning_rate)
```

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

לולאת האימון

```
n_total_steps = len(train_loader)


for epoch in range(epochs):
  for i, (images, lables) in enumerate(train_loader):

    # origin shape: [50, 1, 28, 28]
    # resized: [50, 784]
    images = images.reshape(-1, 28*28).to(device)
    lables = lables.reshape(-1,1).to(device)
    lables = (torch.eq(lables, required_label)).float()


    # forward
    Y_predict = Model(images)

    # backward
    loss = Loss(Y_predict, lables)
    loss.backward()

    # update wights
    optim.step()

    if (i + epoch * n_total_steps) % 100 == 0:
      print(f"epoch= {epoch} i= {i} num= {i+epoch * n_total_st
      # print (Y_predict.T)
      # print (lables.T)

    # zero wights
    optim.zero_grad()
```

- קבענו כי נבצע epoch פעמים מעבר על כל הדוגמאות.
- בכל פעם אנחנו מבצעים 1200 איטרציות של 50 תמונות (batch_size = 50).
- הוספנו את "שיטוח" התמונות, ושינוי מימדי הטנסור.
- המבנה המקורי של כל batch הוא: [50,1,28,28]
- אנחנו המרנו אותו ל: [50,784].
- כמו כן, שינינו את התוצאות (התוויות) של התמונות כך שאם יהיה required_label = 7, התווית תהא 1, אחרת יהיה 0).

ריצת המודל

```
epoch= 0 i= 0 num= 0 loss=0.6477
epoch= 0 i= 100 num= 100 loss=0.1028
epoch= 0 i= 200 num= 200 loss=0.0293
epoch= 0 i= 300 num= 300 loss=0.0310
epoch= 0 i= 400 num= 400 loss=0.0296
epoch= 0 i= 500 num= 500 loss=0.0034
epoch= 0 i= 600 num= 600 loss=0.0089
epoch= 0 i= 700 num= 700 loss=0.0870
epoch= 0 i= 800 num= 800 loss=0.0549
epoch= 0 i= 900 num= 900 loss=0.0050
epoch= 0 i= 1000 num= 1000 loss=0.0041
epoch= 0 i= 1100 num= 1100 loss=0.0452
epoch= 1 i= 0 num= 1200 loss=0.0254
epoch= 1 i= 100 num= 1300 loss=0.0094
epoch= 1 i= 200 num= 1400 loss=0.0024
epoch= 1 i= 300 num= 1500 loss=0.0302
epoch= 1 i= 400 num= 1600 loss=0.0104
epoch= 1 i= 500 num= 1700 loss=0.0245
epoch= 1 i= 600 num= 1800 loss=0.0053
epoch= 1 i= 700 num= 1900 loss=0.0387
epoch= 1 i= 800 num= 2000 loss=0.0426
epoch= 1 i= 900 num= 2100 loss=0.0048
epoch= 1 i= 1000 num= 2200 loss=0.0024
epoch= 1 i= 1100 num= 2300 loss=0.0494
```

- ניתן לראות שריצת המודל הביאה לדיוק (הפסד) של 0.0494 לאחר 2400 איטרציות.

- ביצענו שני איטרציות (epoch = 2), כאשר בכל אחת ביצענו 1200 איטרציות (כאשר בכל פעם העברנו לתוך המודל 50 תמונות בבת אחת).

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

בדיקת המודל

- הרצנו את המודל על ה test_data. ובדקנו שבאמת המודל מזהה את הספרה 7. קיבלנו דיוק של 99.27% !!!!

```
with torch.no_grad():
    n_correct = 0
    n_samples = 0
    for images, lables in test_loader:
        images = images.reshape(-1, 28*28).to(device)
        lables = lables.reshape(-1,1).to(device)
        lables = (torch.eq(lables, required_label)).float()
        y_predict = Model(images).round()
        # print(y_predict.T)
        # print(lables.T)

        n_samples += lables.size(0)
        n_correct += (y_predict == lables).sum().item()

    acc = 100 * n_correct / n_samples
    print(f'Accuracy of the network on the {n_samples} test images: {acc} %')
```

```
Accuracy of the network on the 10000 test images: 99.27 %
```

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

בדיקה נוספת

- בדקנו את ה Batch האחרון של ה tets_data והדפסנו את התוצאה של המודל שלנו לעומת התוצאה האמיתית (התווית).

- ניתן לראות כי כמעט בכל המקרים המודל שלנו נותן תשובה נכונה, כלומר המודל נותן 1 כאשר התווית היא 7, ו- 0 במקרים האחרים.

```
200
10000
torch.Size([50, 1])
1.0 7
0.0 2
0.0 1
0.0 0
0.0 4
0.0 1
0.0 4
0.0 9
0.0 5
0.0 9
0.0 0
0.0 6
0.0 9
0.0 0
0.0 1
0.0 5
0.0 9
1.0 7
0.0 3
0.0 4
0.0 9
0.0 6
0.0 6
0.0 5
0.0 4
0.0 0
1.0 7
0.0 4
0.0 0
0.0 1
0.0 3
0.0 1
0.0 3
0.0 4
1.0 7
0.0 2
1.0 7
```

```python
print(len(test_loader))
print (n_samples)
print(lables.shape)
examples = iter(test_loader)
example_data, example_targets = next(examples)
example_data = example_data.to(device)
example_targets = example_targets.to(device)
example_predict = Model(example_data.reshape(-1, 28*28)).T.round()
example_predict.shape
for i in range(len(example_data)):
  print (example_predict[0,i].item(), example_targets[i].item() )
```

קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות