שניר קוסקא – 208144477
דור ארביב - 206556318

למידה עמוקה – תרגיל בית 1

חלק תיאורטי

Ex1 -

1)

a. given: input size: 10, batch size: m

$\rightarrow$ Shape(X) = $m \times 10$.

number of input samples — dimension/number of features of each sample

b. given: hidden layer output of size 50.

$\rightarrow$ shape(W_h) = 10×50

shape(b_h) = $m \times 50$

$\underset{m\times 50}{(Y_h} = \mathrm{ReLU}(\underset{m\times 10}{X} \cdot \underset{10\times 50}{W_h} + \underset{m\times 50}{b_h})$

c. given: output layer of size 3.

$\rightarrow$ shape(W_o) = 50×3

shape(b_o) = $m \times 3$

$\underset{m\times 3}{(Y} = \mathrm{ReLU}(\underset{m\times 50}{Y_h} \cdot \underset{50\times 3}{W_o} + \underset{m\times 3}{b_o})$

d. shape(Y) = $m \times 3$. $\leftarrow$ מסתכלים קדימה

e. $Y = \mathrm{ReLU}(Y_h \cdot W_o + b_o) \Leftrightarrow$

$\Leftrightarrow \mathrm{ReLU}(\mathrm{ReLU}(X \cdot W_h + b_h) \cdot W_o + b_o)$

שניר קוסקא – 208144477
דור ארביב - 206556318

2) CNN - 3 conv layers
3×3 kernels

input: 200 × 300 × 3
for RGB - 3 channels.

layer 1: 3x3x3 kernel ⇒ 27 + 1 = 28 weights
bias
stride 2.
Input size: 200 × 300 × 3
output feature maps: 100
→ 100 filters of 3x3x3 are needed.
→ (3×3×3+1) · 100 = 2800 parames/weights

layer 2: 3x3x100 kernel → 900+1 = 901 weights,
900 for each feature map + 1 bias term.
200 output feature maps
→ 200 filters of 3x3x100 are required
→ 901 × 200 = 180,200 parameters/weights.

layer 3: 3x3x200 kernel → 1800+1 = 1801 weights
1800 for each feature map + 1 bias term,
400 output feature maps.
→ 400 filters of 3x3x200 are required
→ 1801 × 400 = 720,400 parames.

→ in total 903,400 parameters are required in the CNN.

$$(903{,}400 = \underbrace{2800}_{l_1} + \underbrace{180{,}200}_{l_2} + \underbrace{720{,}400}_{l_3})$$

שניר קוסקא – 208144477
דור ארביב - 206556318

$y = \sum_{i=1}^{m} y_i$

3) a. $\frac{\partial f}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \frac{\partial y_i}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \hat{x}_i$

$y_i = \gamma \hat{x}_i + \beta$

b. $\frac{\partial f}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \frac{\partial y_i}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot 1$

c. $\frac{\partial f}{\partial \hat{x}_i} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \frac{\partial y_i}{\partial \hat{x}_i} = \frac{\partial f}{\partial y_i} \cdot \gamma$

d. $\frac{\partial f}{\partial \sigma^2} = \sum_{i=1}^{m} \underbrace{\frac{\partial f}{\partial \hat{x}_i}}_{\text{נגזרת c}} \cdot \frac{\partial \hat{x}_i}{\partial \sigma^2} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \gamma \cdot \left(-\frac{1}{2}\right) \cdot \frac{(x_i - \mu)}{(\sigma^2 + \varepsilon)^{\frac{3}{2}}} =$

$= -\frac{1}{2}\gamma \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \frac{(x_i - \mu)}{(\sigma^2 + \varepsilon)^{\frac{3}{2}}}$

e. $\frac{\partial f}{\partial \mu} = \sum_{i=1}^{m} \frac{\partial f}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial \mu} + \frac{\partial f}{\partial \sigma^2} \cdot \frac{d\sigma^2}{d\mu} =$

$\sigma^2 - 1 \quad \hat{x}_i$
רק פונ' של μ

$= \sum_{i=1}^{m} \underbrace{\frac{\partial f}{\partial \hat{x}_i}}_{\frac{\partial f}{\partial \hat{x}_i}} \cdot \gamma \cdot \left(-\frac{1}{\sqrt{\sigma^2 + \varepsilon}}\right) - \frac{\partial f}{\partial \sigma^2} \cdot \frac{2}{m} \sum_{i=1}^{m} (x_i - \mu)$

$= \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \gamma \left(-\frac{1}{\sqrt{\sigma^2 + \varepsilon}}\right) - \frac{\partial f}{\partial \sigma^2} \cdot \left[2 \cdot \mu - \frac{2 \cdot m \cdot \mu}{m}\right] =$

0

$= -\sum_{i=1}^{m} \frac{\partial f}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma^2 + \varepsilon}}$

שניר קוסקא – 208144477
דור ארביב - 206556318

$$\frac{\partial f}{\partial x_i} = \frac{\partial f}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial x_i} + \frac{\partial f}{\partial \mu} \cdot \frac{\partial \mu}{\partial x_i} + \frac{\partial f}{\partial \sigma^2} \cdot \frac{\partial \sigma^2}{\partial x_i} \quad (*)$$

$\sigma^2, \mu, \hat{x}_i$

תלויים ב x_i

$$(*) = \frac{\partial f}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma^2+\varepsilon}} - \frac{1}{m} \cdot \left(\sum_{i=1}^{m} \frac{\partial f}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma^2+\varepsilon}} \right)$$

$$- \frac{1}{2} \sum_{j=1}^{m} \frac{\partial f}{\partial \hat{x}_j} \cdot \frac{(x_j-\mu)}{(\sigma^2+\varepsilon)^{\frac{3}{2}}} \cdot \frac{2(x_i-\mu)}{m} \quad (*)$$

$$(*) = \frac{\partial f}{\partial \hat{x}_i} \frac{1}{\sqrt{\sigma^2+\varepsilon}} - \frac{1}{m} \cdot \frac{1}{\sqrt{\sigma^2+\varepsilon}} \cdot \sum_{i=1}^{m} \frac{\partial f}{\partial \hat{x}_i}$$

$$- \frac{(x_i-\mu)}{m} \frac{1}{(\sigma^2+\varepsilon)^{\frac{3}{2}}} \cdot \sum_{j=1}^{m} \frac{\partial f}{\partial \hat{x}_j} \cdot (x_j-\mu) \quad (*)$$

$$(*) = \frac{1}{m\sqrt{\sigma^2+\varepsilon}} \cdot \left[m \frac{\partial f}{\partial \hat{x}_i} - \sum_{j=1}^{m} \frac{\partial f}{\partial \hat{x}_j} - \hat{x}_i \sum_{j=1}^{m} \frac{\partial f}{\partial \hat{x}_j} \cdot \hat{x}_j \right]$$

שניר קוסקא – 208144477
דור ארביב - 206556318

חלק פרקטי

בתיקייה המשותפת ישנו קובץ readme עם הנחיות הרצה.

בחלק זה יושם מודל רשת LeNet5 על פני הסט Fashion MNIST.

ארכיטקטורת הרשת נלקחה מהקישור הבא:

https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/

ומכילה את השכבות הבאות:

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	
Conv 3	120	5 * 5	1	120	tanh
Fully Connected 1	-	-	-	84	tanh
Fully Connected 2	-	-	-	10	Softmax

(נלקחה פונקציית אקטיבציה לא לינארית של ReLU במקום tanh המצוינת עקב קבלת תוצאות טובות יותר עבור פונקציה זו).

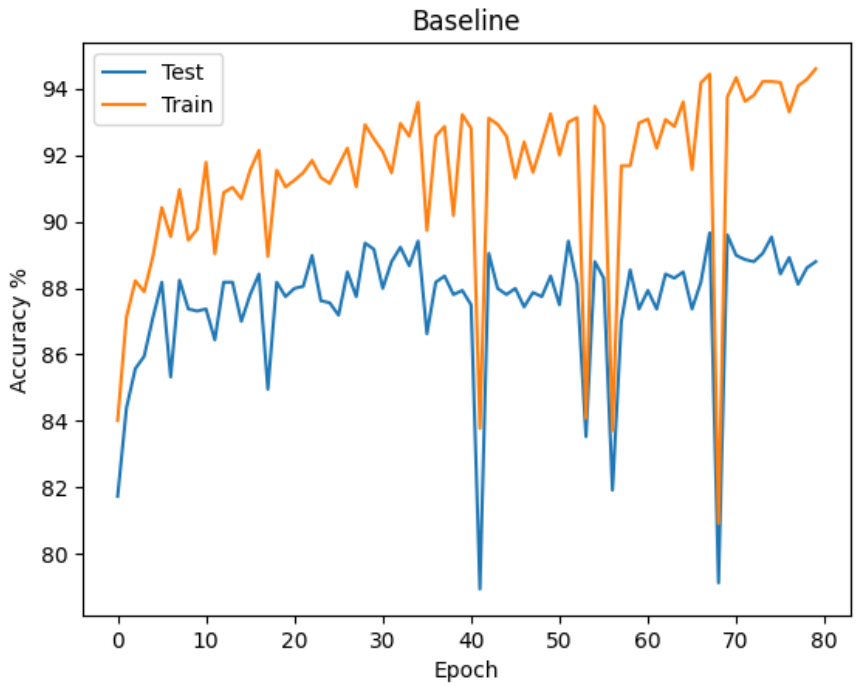
לאחר מספר איטרציות אימון, נבחרו הפרמטרים הבאים בעת אימון הרשת עבור כל אחת מהטכניקות הדרושות בתרגיל:

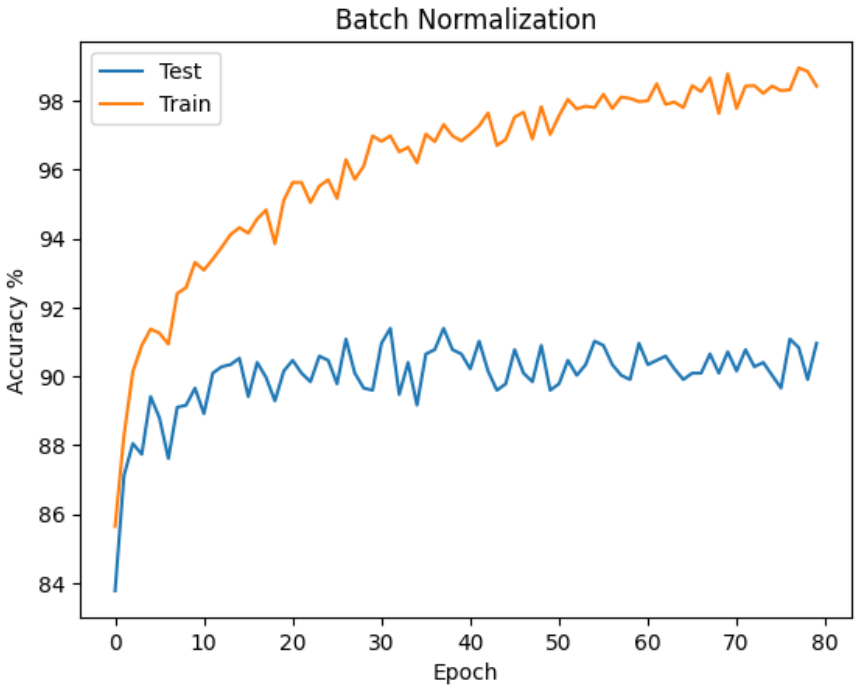
- Dropout: 20% (probability to drop each neuron, during training only, when calculating accuracy, model set to eval mode and dropout is shut off).
- Batch normalization: when activated, performed after each convolutional layer, according to the number of output channels.
- Weight decay (l2 penalty) parameter: 0.001.
- Optimizer: Adam (with learning rate of 0.01).
- Batch size: 64.
- Percent of training data used as validation set: 20% (to choose a model for each technique according to smallest achieved validation loss).
- Input images sized 28X28 transformed into size 32X32 to be given as input to the Lenet5 architecture.
- Number of epochs during training: 80.

כפי שניתן לראות במחברת המשותפת בקולב, לאורך האימון מודפס לאחר כל epoch הloss על סט הוולידציה, כמו גם ה-accuracy על סט האימון והמבחן. לאחר הרצת כל ה-epochs מודפס גם הloss על סט המבחן עבור המודל הנבחר.
בסיום כל אימון (עבור כל טכניקה) נשמר המודל הנבחר בקובץ pt. בתיקייה המשותפת (בה קיימים 4 מודלים, אחד לכל שיטה שאומנה).

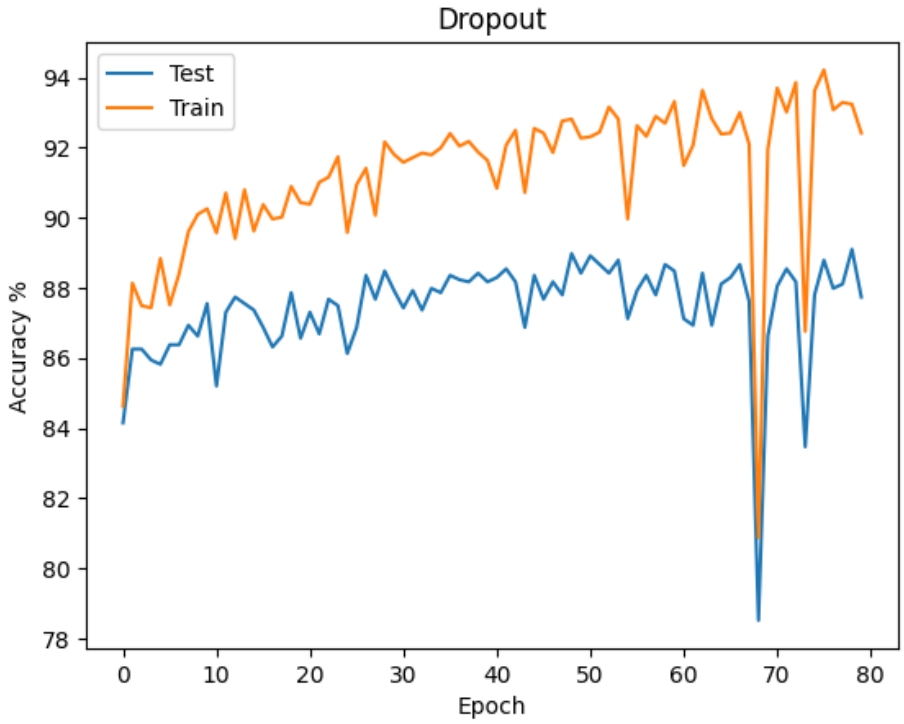
להלן הפלטים שהתקבלו עבור גרפי ההתכנסות בכל אחת מהשיטות:

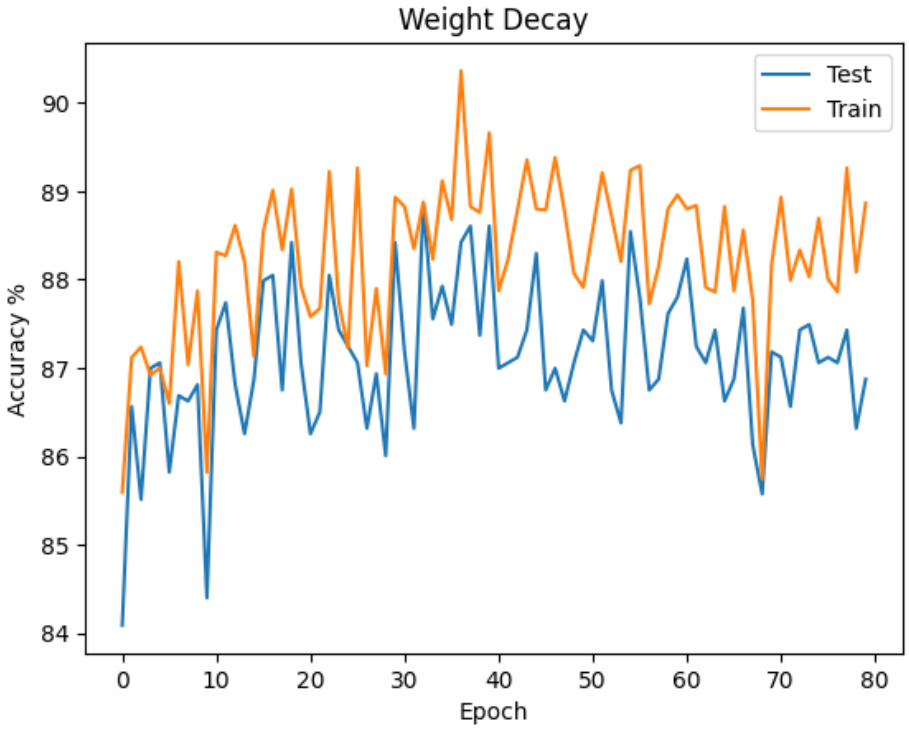
שניר קוסקא – 208144477
דור ארביב - 206556318





שניר קוסקא – 208144477
דור ארביב - 206556318





שניר קוסקא – 208144477
דור ארביב - 206556318

להלן תוצאות הדיוק בסיום האימון על סט המבחן עבור המודל הנבחר בכל אחת מהשיטות :

	Test Accuracy %	Train Accuracy %
Baseline	88.54489164086688	89.62226640159045
Batch Normalization	87.55417956656346	90.28495692511598
Dropout	89.6594427244582	94.03578528827038
Weight Decay	88.17337461300309	90.77534791252485

סה״כ, התקבלו תוצאות דיוק דומות עבור כלל השיטות, כאשר הדיוק המקסימלי התקבל בשיטת ה-Dropout. שיטת רגולריזציה זו מנסה למנוע את התלות בין נוירונים ברשת (co-adaptation) ע״י התעלמות רנדומלית מחלק מהנוירונים בעת האימון, ובכך למנוע ovefit ולאפשר לרשת לבצע הכללה טוב יותר. עבור המודל הנבחן, שיטה זו עבדה מעט טוב יותר מהגבלת המשקלות בשיטת weight decay או בשיטת נרמול מוצא השכבות ברשת (batch normalization).