הגדרת המודל ופונקציית המחיר

הגדרת המודל

עלינו לתכנן רשת המסוגלת לסווג את התמונות הנתונות לעשר המחלקות השונות של פריטי הלבוש, לצורך זה נרחיב את מודל הנוירון היחיד משני צידיו:
ראשית, הקלט שלנו כעת הוא טנזור מגודל 28X28, על כן **נשטח** אותו: נמיר אותו לטנזור חד מימדי מגודל 784 על ידי שרשור השורות אחת אחרי השניה כמו בדוגמה הבאה.

```
A = torch.arange(2*5).reshape(2,5)
print(A, A.size())
A = A.flatten()
print(A, A.size())
```

פלט:

```
tensor([[0, 1, 2, 3, 4],
        [5, 6, 7, 8, 9]]) torch.Size([2, 5])
tensor([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]) torch.Size([10])
```

הטנזור המשוטח יהווה את הקלט החדש לרשת.

שנית, הנתונים מתחלקים לעשר מחלקות ובהתאם יהיו במודל עשרה נוירוני פלט – אחד לכל מחלקה. את הפונקציה הלוגיסטית תחליף המקבילה הרב-מימדית שלה, פונקציית ה-softmax:

$$\text{softmax}\left(z_0, z_1, \ldots, z_k\right) = \frac{1}{\sum_{n=0}^{k} e^{z_k}} \begin{pmatrix} e^{z_0} \\ e^{z_1} \\ \vdots \\ e^{z_k} \end{pmatrix}.$$

שימו לב שאיברי וקטור הפלט הם מספרים חיוביים, אשר סכומם אחד, ולכן הפלט הוא **וקטור הסתברויות**. פונקציה זו מקבלת את שמה עקב התכונה הבאה: אם אחד מהערכים בוקטור הקלט $\left(z_0, z_1, \ldots, z_k\right)$ גדול מהאחרים, למשל $z_1 > z_m$ לכל m, אז פעולת האקספוננט תקצין פער זה, ויתקבל $e^{z_1} >> e^{z_m}$. לאחר הנרמול, הפלט יהיה קרוב ל-$\left(0,1,0,..,0\right)$: 1 היכן שהיה הערך המקסימלי בקלט, ו-0 בשאר הערכים, זהו למעשה ה-argmax של הקלט. ככל ש z_1 גדול יותר משאר ערכי הקלט, כך הפלט יהיה קרוב לוקטור $\left(0,1,0,..,0\right)$. תוצאה זו, יחד עם העובדה שה-softmax היא פונקציה גזירה, מסבירה את נוכחות המילה soft בשמה.
כעת בידנו כל הדרוש להגדרת מודל הסיווג כרשת נוירונים:

1. הרשת תקבל כקלט טנזור חד מימדי, $\left(x_0, x_1, \ldots, x_{783}\right)$.
2. על הקלט יופעלו עשר פונקציות ליניאריות שונות, אחת לכל מחלקה:

$$z_0 = w_{0,0}x_0 + w_{0,1}x_1 + \ldots + w_{0,783}x_{783} + b_0$$
$$z_1 = w_{1,0}x_0 + w_{1,1}x_1 + \ldots + w_{1,783}x_{783} + b_1$$
$$\vdots$$
$$z_9 = w_{9,0}x_0 + w_{9,1}x_1 + \ldots + w_{9,783}x_{783} + b_9$$

או בכתיב וקטורי,

$$Z = W \cdot X + b,$$

כאשר

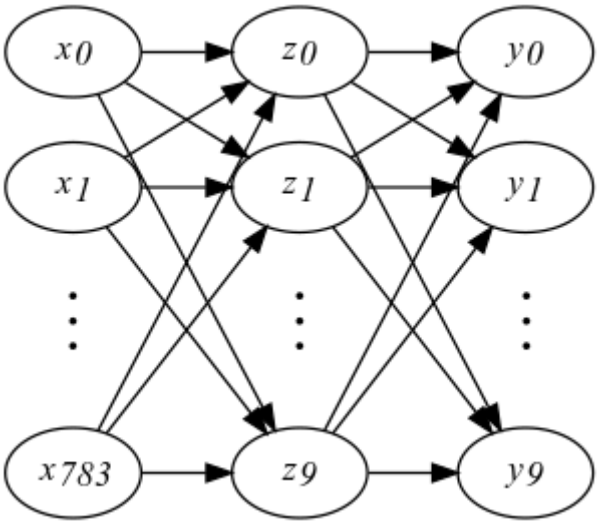
$$Z = \begin{pmatrix} z_0 \\ \vdots \\ z_9 \end{pmatrix}, X = \begin{pmatrix} x_0 \\ \vdots \\ x_9 \end{pmatrix}, b = \begin{pmatrix} b_0 \\ \vdots \\ b_9 \end{pmatrix}, W = \begin{pmatrix} w_{0,0} & \cdots & w_{0,783} \\ \vdots & \ddots & \vdots \\ w_{9,783} & \cdots & w_{9,783} \end{pmatrix}.$$

3. תוצאת חישוב זה תועבר לפונקציית ה-softmax,

$$Y=\begin{pmatrix} y_0 \\ \vdots \\ y_9 \end{pmatrix}=\text{softmax}\left(Z\right)=\frac{1}{\sum_{n=0}^{9} e^{z_n}}\begin{pmatrix} e^{z_0} \\ \vdots \\ e^{z_9} \end{pmatrix}.$$

4. פלט הרשת, Y, הוא וקטור אשר כל איבר בו הוא ההסתברות שהקלט הנתון שייך למחלקה המתאימה: y_k היא הסתברות השייכות למחלקה ה- k.

ובאיור סכמטי,



שימו לב שפרמטרי המודל הם המטריצה W והוקטור b אשר קובעים את **פונקציית האגרגציה** (aggregation function) הליניארית $Z=W\cdot X+b$.

פונקציית המחיר

בעבר עסקנו בסיווג נקודות שחורות ולבנות ובחרנו בפונקציית מחיר אשר מענישה מודל שמסווג לא נכונה נקודה שחורה נתונה באופן פרופורציונלי ל- $\log\left(y\right)$, כאשר y דאז הייתה ההסתברות החזויה על ידי המודל שהנקודה היא שחורה. בדומה הקנס על כל נקודה לבנה היה פרופורציונלי ל- $\log\left(1-y\right)$, ונשים לב ש- $1-y$ היא למעשה ההסתברות החזויה שהנקודה היא **לבנה**. עתה נכליל רעיון זה למקרה הרב מימדי. ראשית, לכל נקודת דגימה נתונה, $X=\left(x_0,x_1,...,x_{783}\right)$ נתון גם הסיווג לאחת המחלקות 0-9. נקודד סיווג זה בצורת one-hot ונקבל וקטור $Y_t=\left(y_{t0},y_{t1},...,y_{t9}\right)$ אשר רק אחד מאיבריו הוא 1 ושאר האיברים הם 0: $y_{tk}=1$ אם ורק אם סיווג הנקודה הנתון הוא למחלקה ה- k. ראו למשל,

```
imgs, labels = next(iter(train_dataloader))
print(labels)
torch.nn.functional.one_hot(labels,num_classes=10)
```

פלט:

```
tensor([0, 3, 1, 9])
tensor([[1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]])
```

כעת, נגדיר את **התרומה למחיר של כל דגימה** בצורה הבאה,

$$H\left(X,Y_t\right)=-\sum_{n=0}^{9}y_{tn}\log\left(y_n\right),$$

כאשר הוקטור $Y=\left(y_0,..,y_9\right)$ הוא פלט הרשת עבור הקלט X. לפי הגדרת Y_t, כל האיברים בביטוי זה, למעט אחד, מתאפסים. נקבל ש-$H\left(X,Y_t\right)=-\log\left(y_k\right)$, כאשר k היא המחלקה אליה נקודת דגימה זו שייכת, בדיוק כמו במקרה הדו-מימדי. בכדי לחשב את פונקציית המחיר הכללית, נשאר רק לחשב ממוצע של פונקציה זו על כל נקודות הדגימה. התוצאה המתקבלת היא **האנטרופיה הצולבת** עבור בעיית סיווג למספר רב של מחלקות:

$$C\left(W,b\right)=\frac{1}{\#Data}\sum_{\left(X,Y_t\right)\in Data}H\left(X,Y_t\right)=-\frac{1}{\#Data}\sum_{\left(X,Y_t\right)\in Data}\sum_{n}y_{tn}\log\left(y_n\right)$$

כאן יש לשים לב שפונקציית המחיר תלויה בפרמטרים, W,b דרך הסתברויות הסיווג $\left(y_0,...,y_9\right)$ אשר בתורן תלויות בפרמטרים דרך האגרגציה Z ולכן כלל השרשרת שוב יהיה שימושי לצורך אימון המודל בעזרת אלגוריתם מורד הגרדיאנט.

שאלות לתרגול

1. ספרו במדויק כמה פרמטרים קיימים במודל הסיווג הנ"ל.
2. חשבו את $\nabla H=\left(\frac{\partial H}{\partial W},\frac{\partial H}{\partial b}\right)$.

הנחיות:

א. עבור אחד מנוירוני הפלט, חשבו את $\frac{\partial H}{\partial y_k}$.

ב. עבור אותו נוירון, חשבו את הנגזרת לפי אחד מרכיבי האגרגציה, $\frac{\partial y_k}{\partial z_m}$.

ג. עבור אגרגציה זו חשבו את הנגזרת לפי אחד מהפרמטרים, $\frac{\partial z_m}{\partial w_{p,q}}$.

- הפרידו את החישוב למקרים, $p=q$ או $p\neq q$.

ד. חברו את כל התוצאות בעזרת כלל השרשרת.

- היעזרו באיור הרשת המופיע לעיל.

3. האם תוכלו לכתוב ביטוי פשוט עבור $\frac{\partial Z}{\partial W}$? ועבור $\frac{\partial Z}{\partial b}$?
4. הגדירו את מודל הרשת בשתי שורות קוד. **רמז:** היעזרו בספרייה `torch.nn`.
5. הזינו לתוך הרשת שהגדרתם בשאלה הקודמת את סט האימון של אוסף הנתונים MNIST-Fashion במלואו וחשבו את פונקציית המחיר על התוצאה.

הנחיות:

- קודם להזנת הנתונים, עליכם להמירם לפורמט נתונים מתאים, float. השתמשו לצורך זה בפונקציה `torchvision.transforms.ConvertImageDtype` כבר בשלב טעינת הנתונים.
- אחרי כן, עליכם לשטח את התמונות, השתמשו במתודה `flatten()`.