



מעבדה ב-VLSI ספרתי - 0450111

חוברת הדרכה
רשת נוירונים

<http://www.ee.technion.ac.il/vlsi/>

[הערות נא לשלוח ל-goel@ee](mailto:goel@ee)

כל הערה תתקבל בברכה!

עדכון אחרון - 11:06 19/09/2024

מסמך זה כתוב בלשון זכר ע"מ להקל על הכתיבה אך מתייחס לנשים ולגברים כאחד. עמכם הסליחה.

תיאור רשת נוירונים פשוטה

בתחילת כל אחד משני המפגשים תדרשו להגיש את דו"ח ההכנה הרלוונטי לאותו מפגש כשהוא מכיל פתרונות לכל המטלות השונות – תשובות לשאלות, דיאגרמות וקטעי קוד. את קטעי הקוד יש להכין בנוסף בקבצים נפרדים לפי שמות המודולים השונים ולהעלות אותם ל- `labadmin` יחד עם דו"ח המכין בקובץ `.zip`.

הערות כלליות :

- בכל `testbench` שמגדיר אות שעון, יש לאתחל את אות השעון בזמן 0 ואת יתר הסיגנלים בזמן 1.
- הקבצים שתתבקשו ליצור הם קבצי טקסט רגילים אשר הסיימת שלהם היא `.sv`. במקום `.txt`. על הדו"ח המכין להכיל בתוכו את כל קטעי הקוד שבקבצים.
- להכנת הקבצים בסעיף זה, רצוי להשתמש ב- `notepad` (או אפילו טוב יותר ב- `notepad++`) ולא ב- `Word`. לעתים קבצי `Word` מכילים תווים לא רצויים ב- `linux`.
- שמות הקבצים שעליך להכין מוגדרים בהמשך המסמך.
- יש לצרף את הקוד לדוח בצורה מסודרת ו"קריאה" או כקובצי טקסט נפרדים
- אין צורך להכין קבצים נוספים. כל יתר הקבצים יינתנו במהלך הניסוי.
- נא להקפיד להגיש קוד מלא ולא רק חתיכות קוד.
- יש להגיש את הדו"ח בפורמט `pdf` ועם שם כמו `part1_shimshon_yovav.pdf`.

רקע כללי

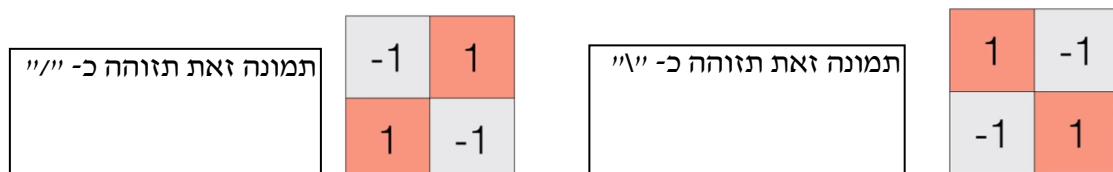
מטרת הניסוי היא תכנון חומרה (מאיץ ייעודי) למערכת לומדת. מערכת לומדת עובדת בשתי פאזות, פאזת הלימוד ופאזת הפעולה (במקרה שלנו נקרא לשלב זה שלב הסיווג). במהלך הניסוי נעסוק בעיקר בשלב הסיווג. אנו נניח ששלב הלימוד התבצע מראש בצורה זו או אחרת. ראשית נתחיל בתיאור מערכת לומדת פשוטה. יחד עם זאת, מומלץ **מאד מאד** לצפות בסרטון :

A friendly introduction to Convolutional Neural Networks and Image Recognition

<https://www.youtube.com/watch?v=2-017ZB0MmU&t=160s>

המטרה של תיאור זה היא להבין את האלגוריתם של המערכת ובדיוק איזה חישוב עלינו לבצע וכל זה על מנת שנוכל לתכנן חומרה שתבצע את החישוב ביעילות מירבית.

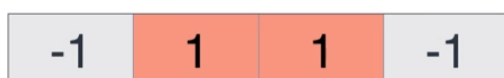
בשלב ראשון נדרש לתכנן מערכת שתהיה מסוגלת לזהות שני תווים `"/` ו- `"\` (2X2 פיקסלים).



איור מס' 3

נערוך את התמונות בצורה קצת יותר נוחה, כלומר כמערך חד מימדי. נשתמש במערך מסוג זה כאשר נגיע למימוש המערכת.

תמונת ה- `"/`

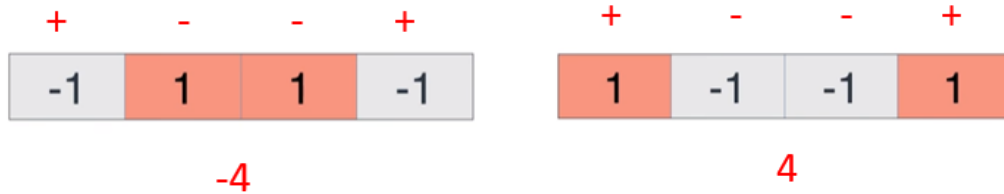


תמונת ה- `"\`



איור מס' 4

כעת צריך למצוא דרך שתאפשר למחשב להבחין בין שתי הצורות. אם למשל ננקוט בגישה של לחבר או להכפיל יחד את כל אברי המערך, לא נצליח להבדיל ביניהם כי בשני המקרים נקבל את אותה התוצאה. לעומת זאת, אם נחבר את שני האיברים הקיצוניים ונחסיר את שני האמצעיים נקבל את התוצאה הבא :



איור מס' 5

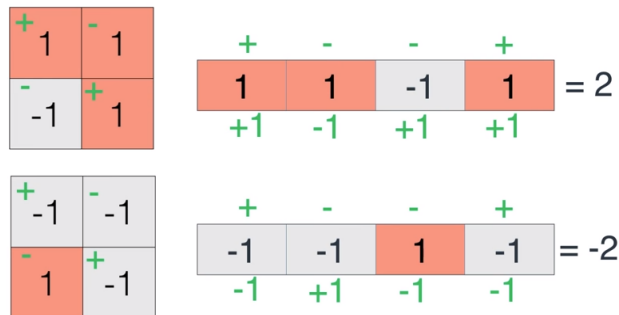
פעולה מסוג זה תאפשר למחשב להבחין בין הצורות. במילים אחרות, אם פעולה זאת מניבה מספר חיובי אז יש לזהות את הצורה כ- "/" אחרת פעולה זאת לא מזהה דבר. לתבנית הספציפית הזאת של '+' (פלוסים) ושל '-' (מינוסים) נקרא "מסנן" (במקרה זה מסנן א2). קל לראות שניתן בקלות לבנות תבנית/מסנן של '+' (פלוסים) ושל '-' (מינוסים) שבאמצעותה ניתן לזהות "\". (יקרא בהמשך מסנן א1). מילה אחרת למסננים שנשתמש בה בהמשך התיאור היא kernels.

מה לגבי הצורות :



איור מס' 6

אם נפעיל על שתי הצורות את המסנן לזיהוי "\", נקבל את התוצאה הבא :



איור מס' 7

במקרה זה, המחשב יזהה את הצורה השמאלית באיור 6 כ- "\ " והמסנן לזיהוי "/" יזהה את הצורה הימנית באיור 6 כ- "/" . ניחוש לא רע בכלל! כלומר שיטה זאת בהחלט עושה רושם טוב. אבל כיצד מחשב יכול להגיע לשיטה המתוארת לבד? כלומר איך מוצאים את המסננים עבור כל מקרה ומקרה?

דרך אחת היא פשוט "ללמוד". כלומר אפשר לתכנת את המחשב לבדוק את כל האפשרויות ולספק למחשב משוב חיובי או שלילי לגבי כל אפשרות. בסוף התהליך המחשב בוחר במסננים הטובים ביותר. זה למעשה תהליך הלמידה. כאמור, במהלך הניסוי, אנו נתכנן מערכת רק לשלב הסיווג.

✗	✗	✗	✗	✗	✗	✗	✓	מסנן א1	
-	-	+	-	-	-	+	+		+
-	-	-	-	+	-	-	-	+	-
✓	✗	✗	✗	✗	✗	✗	✗		
-	+	-	+	+	+	+	-	-	+
+	-	-	+	-	+	+	+	+	+

מסנן א2

איור מס' 8 – כל המסננים האפשריים עבור 4 פיקסלים

בדוגמא פשוטה זאת, נדרשת בדיקה של 16 אפשרויות (2 בחזקת 4). קל לראות שעבור מערכות אמיתיות, מספר האפשרויות הוא עצום ומכאן הצורך במאיצים ממומשים בחומרה ייעודית. כמובן שבנוסף לזה קיימות שיטות יעילות יותר למציאת המסווג (ראה סרטון). נחזור ונאמר כי ניסוי זה אינו עוסק בשלב הלמידה ומניח כי הלמידה הנדרשת כבר נעשתה מראש והמשקלים האופטימליים לכל מסנן ידועים, אבל עדיין תידרשנה פעולות כדי לטעון את המשקלים הידועים למערכת החומרה כדי שזו תוכל לבצע את פעולותיה.

כעת נסבך את הבעיה קצת. הפעם נרצה לזהות 4 צורות שונות ("X", "O", "\", "/") בתמונה של 3x3 פיקסלים כאשר כל אחת מהצורות האלו מורכבת מ-"/" ו-"\" בלבד:

1	-1	-1
-1	1	-1
-1	-1	1

\

1	-1	1
-1	1	-1
1	-1	1

X

-1	-1	1
-1	1	-1
1	-1	-1

/

-1	1	-1
1	-1	1
-1	1	-1

O

איור מס' 9

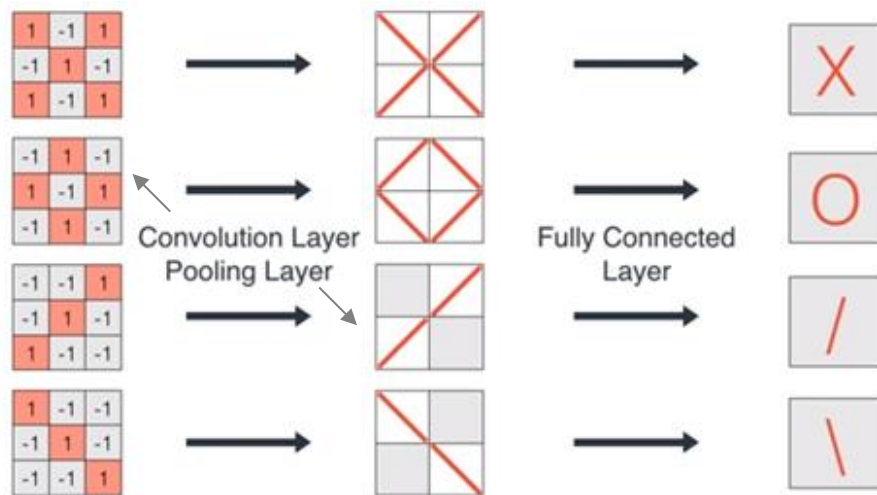
כיצד ניתן לבצע זאת בצורה יעילה? אפשרות אחת כמובן היא להשתמש בשיטה הקודמת, כלומר לנסות לבנות מסנן לתמונה השלמה:

+	1	-1	+	1	
-	-1	+	1	-	-1
+	1	-	-1	+	1

איור מס' 10

מסתבר ששיטה זאת מאד לא יעילה ובין השאר מכיוון שקיימים לה 2048 מסננים (2 בחזקת 9). שיטה טובה בהרבה היא השיטה שמנסה לפרק את התמונה לתתי תמונות (של 2x2 פיקסלים) ולנסות לזהות איזו צורה קיימת בכל תת תמונה ולאחר מכן להגיע למסקנה לגבי התמונה השלמה.

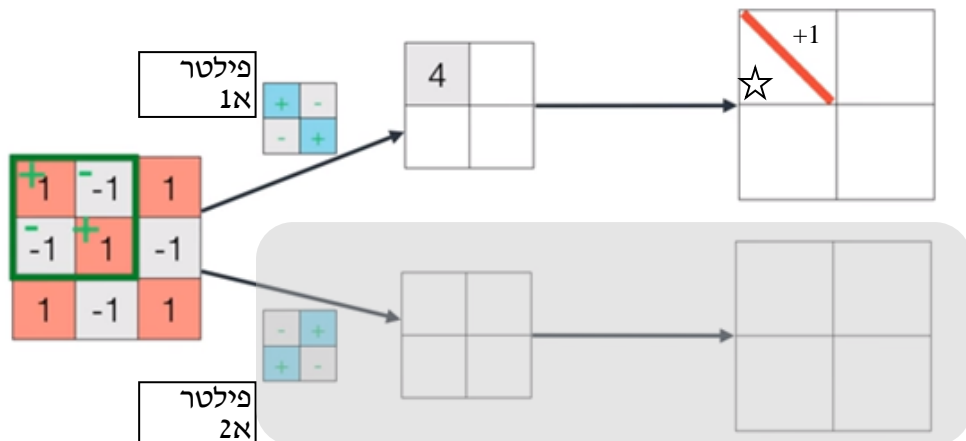
השיטה עובדת בצורה הבאה. כפי שמתואר באיור הבא, בשלב ראשון מנסים לזהות איזו צורה (" \ " , " / " או כלום) קיימת בכל "אזור" (2X2) של התמונה הגדולה. בשלב שני מנסים לזהות האם בתמונה הגדולה מצויר " \ " , " / " , " \ " או " X " .



איור מס' 11

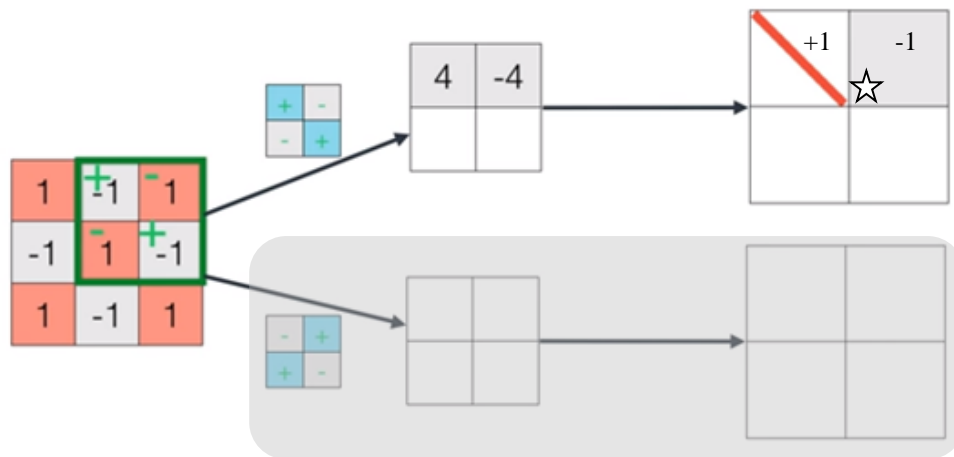
נתמקד בחלק הראשון המורכב משני שלבים, הראשון נקרא שלב הקונבולוציה והשני שלב **pooling** (קיצוץ נתוני ביניים לפי השגת ערך סף של זיהוי תת צורה בתת אזור בתמונה).

לזיהוי שתי צורות " \ " ו- " / " דרושים שני פילטרים (מסננים). נתחיל עם הפילטר הראשון (1א). ממקמים את הפילטר באתר הראשון כפי שמופיע באיור מס' 12 במסגרת הירוקה. מבצעים את החישוב. התוצאה המתקבלת היא 4. מכיוון שזה מספר "גדול" (עבר ערך סף מסוים שקבענו) מחליטים שבאתר זה נמצא " \ " .



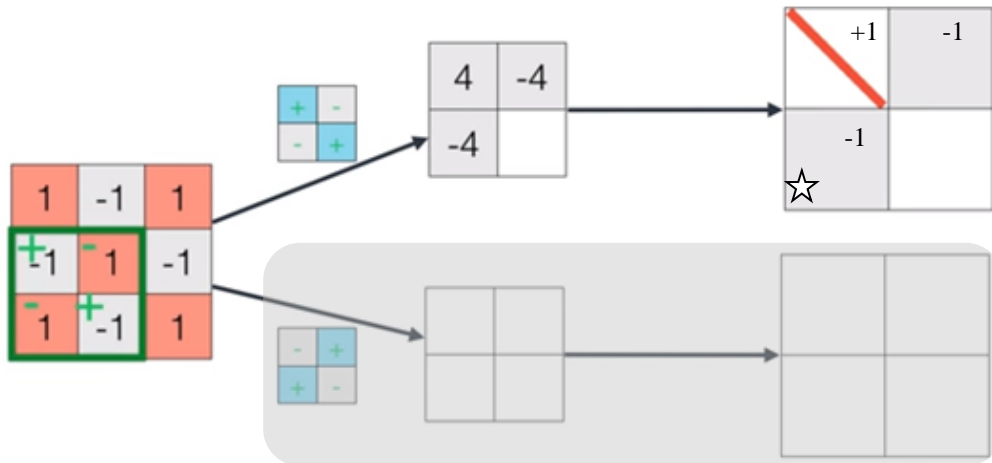
איור מס' 12

עוברים לאתר הבא. הפעם מתקבלת תוצאה -4 מכיוון שזה מספר "קטן" מחליטים שלא נמצא " \ " במקום הזה :



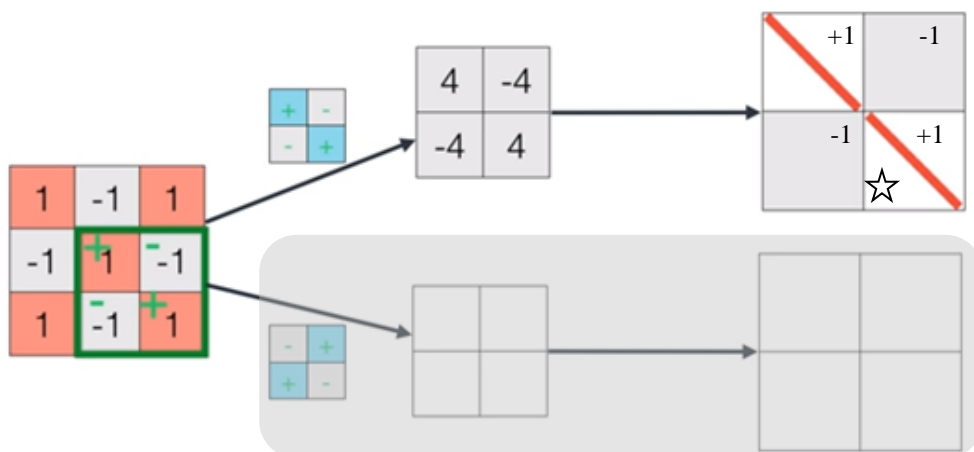
איור מס' 13

עוברים לאתר הבא. גם הפעם מתקבלת תוצאה -4 לכן מחליטים שוב שאין כלום במקום הזה :



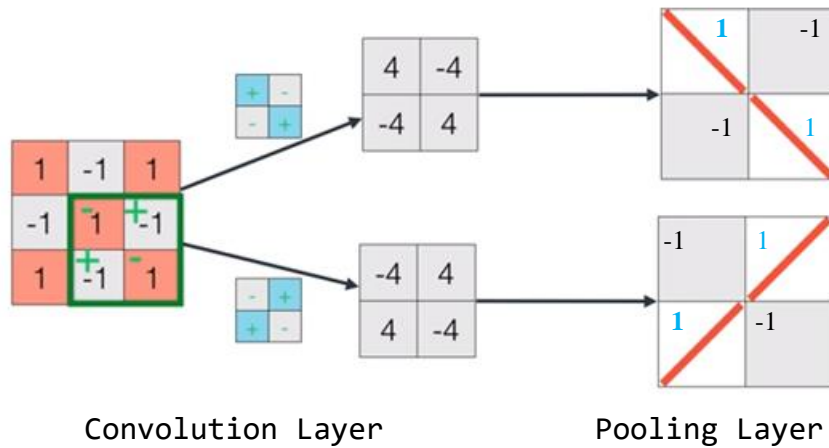
איור מס' 14

באתר האחרון שוב מתקבלת התוצאה 4 ולכן מחליטים שבאתר זה נמצא "\".



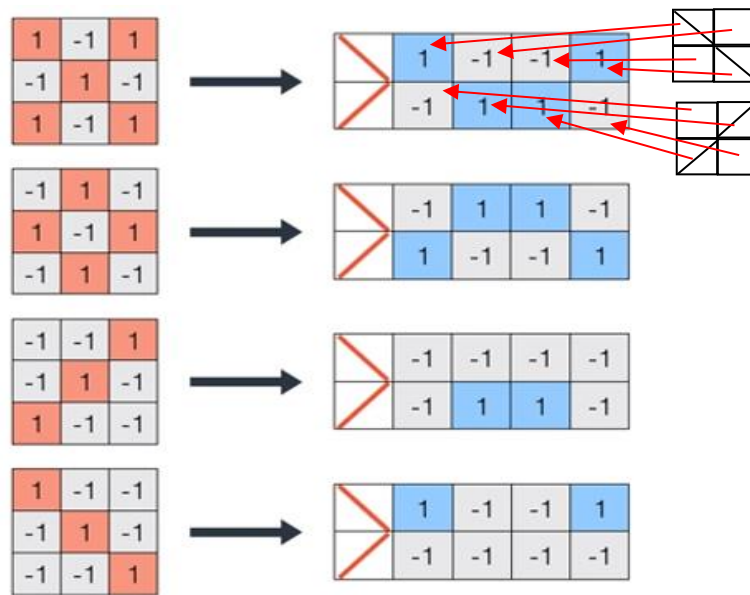
איור מס' 15

כעת חוזרים על כל התהליך עם המסנן השני (א2). מתקבלת התוצאה הבא:



איור מס' 16

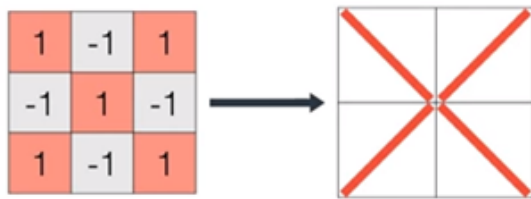
למעשה, לכל צורה ("\" /" - ו-), השלב הראשון מניב שתי תוצאות (או מטריצות) – אחת לכל מסנן שבאמצעותן ניתן להסיק לגבי הצורה שבכניסה. להלן סיכום שכל התוצאות שיכולות להתקבל עבור הצורות השונות כאשר שתי המטריצות (2x2 כל אחת) נפרסות לשתי שורות (1x4 כל אחת):



איור מס' 17

איור 17 מסכם מציאת "/" או "\" בכל אתר של התמונה הגדולה עבור כניסות שונות (השורה הראשונה מעידה על מציאת (1) או אי מציאת (-1) צורה "\" בכל אחד מארבעת האתרים ואילו השורה השניה מעידה על מציאת (1) או אי מציאת (-1) צורה "/" בכל אחד מארבעת האתרים). זוהי למעשה תוצאת ה- **pooling layer**.

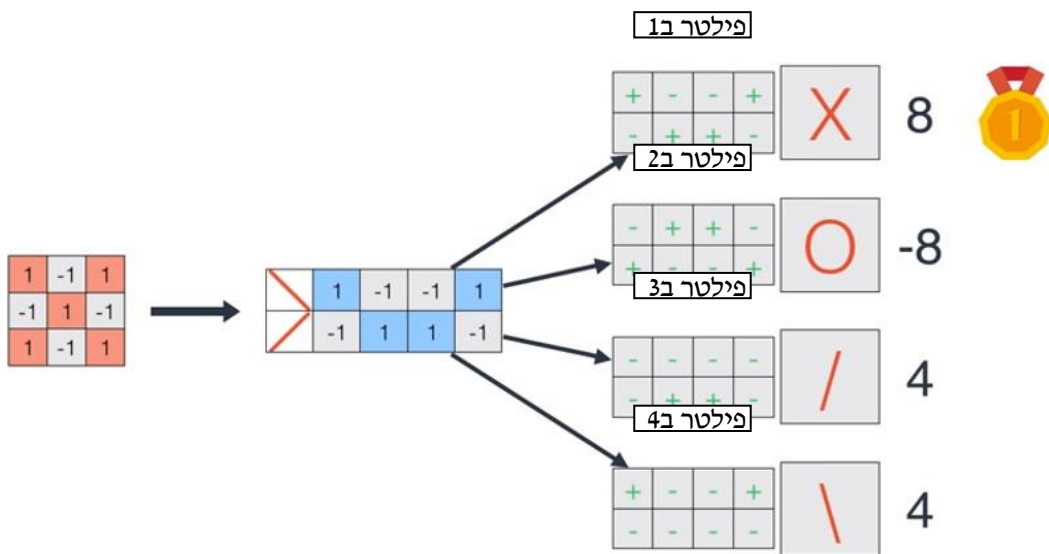
בשלב השני (**Fully Connected Layer**) המחשב משלב את שתי התוצאות הנייל על מנת להגיע למסקנה סופית לגבי הצורה שבתמונת הכניסה. בדוגמא הקודמת הצורה שבתמונה הכניסה היא "X". כיצד זה מתבצע ?



איור מס' 18

זהו התפקיד של מסננים (ארבעה) מסננים של ה- Fully Connected Layer שמכיל אוסף נוסף של שבאמצעותם ניתן להחליט על איזו צורה מדובר. מסננים

תוצאת ה- pooling



איור מס' 19

כל תוצאה שמתקבלת משלב ה- pooling "נבחנת" על ידי ארבעה מסננים כפי שמתואר באיור 19. החישוב שמתבצע הוא סכום האיברים בתוצאת ה- pooling לפי הסימנים שבמסנן. מכיוון שהמסנן של ה- "X" מניב את התוצאה הגבוהה ביותר, מזהים את הכניסה כ- "X". גם כאן, תכנון המסננים נעשה בשלב הלימוד שלא נעסוק בו בניסוי זה (למעט הצורך בטעינת ערכי המסננים למערכת).