

CNN - 11 הרצאה

PDF

Classical) Gradient Descent)

צעד העדכון ב gradient descent נתון על ידי:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{(t)})$$

:ERM ב .1

$$\arg \min_{\boldsymbol{\theta}} \underbrace{\frac{1}{N} \sum_{i=1}^N l(h(\mathbf{x}^{(i)}; \boldsymbol{\theta}), y^{(i)})}_{g(\boldsymbol{\theta}; \mathcal{D})}$$

:MLE .2

$$\arg \min_{\boldsymbol{\theta}} \underbrace{- \sum_{i=1}^N \log(p_{y|x}(y^{(i)} | \mathbf{x}^{(i)}; \boldsymbol{\theta}))}_{g(\boldsymbol{\theta}; \mathcal{D})}$$

Classical) Gradient Descent)

ב ERM:

$$\arg \min_{\theta} \underbrace{\frac{1}{N} \sum_{i=1}^N l(h(\mathbf{x}^{(i)}; \theta), y^{(i)})}_{g(\theta; \mathcal{D})}$$

ב MLE:

$$\arg \min_{\theta} \underbrace{- \sum_{i=1}^N \log(p_{y;x}(y^{(i)} | \mathbf{x}^{(i)}; \theta))}_{g(\theta; \mathcal{D})}$$

• הגרדיאנט מכיל סכום על כל המדגם אשר יכול להיות בעייתי כאשר המדגם גדול.

• נרצה להשתמש בחישוב אשר משתמש בכל צעד רק בחלק מן המדגם.

Stochastic Gradient Descent

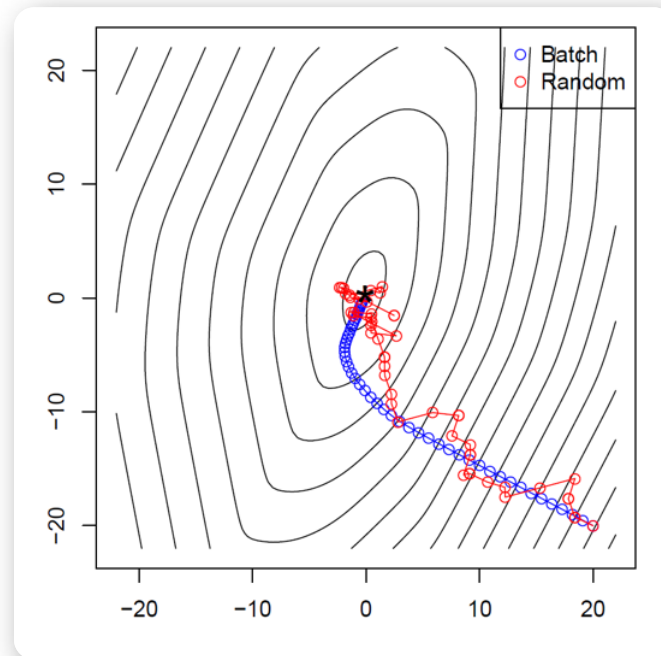
- מחשב בכל פעם את הנגזרת על פי **דגימה בודדת** מתוך המדגם, כאשר בכל צעד נשתמש בדגימה אחרת.
- שתי אופציות לבחירה של הדגימה בכל צעד הינן:
 1. להגריל דגימה אקראית אחרת בכל צעד.
 2. לעבור על הדגימות במדגם בצורה סידרתית.
- כל אחת מהדגימות תצביע לכיוון שונה מהנגזרת של הסכום אבל בממוצע הכיוון הכללי יהיה זהה לכיוון של הסכום.
- החישוב הוא מאד מהיר אבל הגרדיאנט מאד "רועש".

Stochastic Gradient Descent

יתרונות:

1. מחיר איטרציה לא תלוי במדגם

2. חיטכון בזיכרון



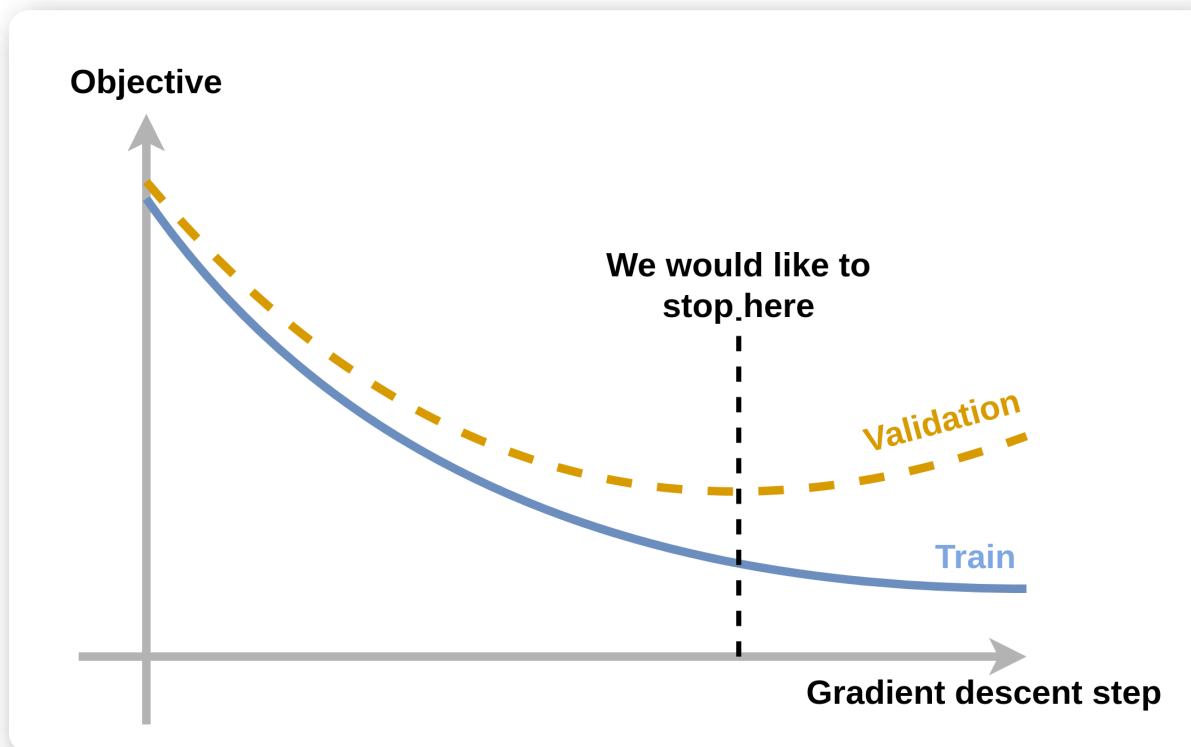
Mini-Batch Gradient Descent

- פתרון ביניים.
- בשיטה זו נשתמש בקבוצת דגימות מתוך המדגם המכונה **mini-batch**. בכל צעד אנו נחליף את ה **mini-batch**.
- השיטה הנפוצה ביותר לאימון של רשתות נוירונים.
- גדלים אופייניים של ה **mini-batch** הינם 32-256 דגימות.

- **Epoch**: מעבר שלם על המדגם.
- מתייחסים ל **mini-batch** בשם **batch**.
- בחבילות רבות האלגוריתם **gradient descent** מופיע תחת השם **stochastic gradient descent**.

עצירה מוקדמת של gradient descent

- דרך מוצלחת נוספת למנוע התאמת-יתר הינה לעצור את אלגוריתם הגרדיאנט לפני שהוא מתכנס.
- זה נעשה על ידי חישוב ה objective על ה validation set ובחירת הפרמטרים שממזערים את ה objective.



Convolutional Neural Networks (CNN

- ב MLP ניתן להגדיל את יכולת הייצוג על ידי הגדלת הרשת (מספר השכבות והרוחב שלהם).
- כפי שקורה בכל מודל פרמטרי, הגדלה של יכולת הייצוג תגדיל גם את ה overfitting.
- רשת בעלת ארכיטקטורה טובה היא דווקא רשת בעלת יכולת ייצוג נמוכה אשר עדיין מסוגלת לקרב בצורה טובה את הפונקציה שאותה היא מנסה למדל.
- במקרים מסויימים ארכיטקטורה בשם convolutional neural network (CNN) עונה בדיוק על דרישות אלו. המוטיבציה המקורית שלה הגיע מהתחום של עיבוד תמונה.

Convolutional Neural Networks (CNN

עד עכשיו הנחנו שהמאפיינים נתונים לנו. מה קורה כאשר הקלט הוא אות טבעי - תמונה, אודיו וכו'?

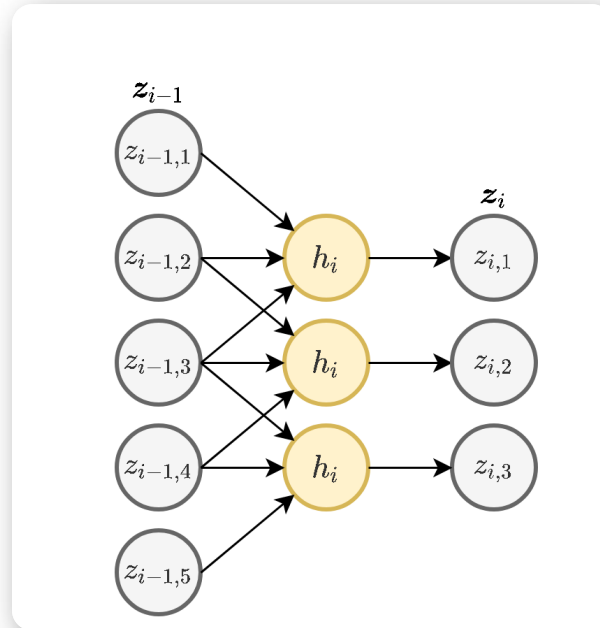


Convolutional Neural Networks (CNN)

מוטיבציה להכנסת מבנה לרשתות עצביות העוסקות בתמונות

- בתמונות טבעיות יש קורלציה גבוהה בין פיקסלים קרובים שדועכת עם המרחק
 - לתמונות יש מבנה היררכי של מאפיינים מקומיים (קצוות, צורות, אובייקטים)
 - זיהוי אובייקטים אינווריאנטי למיקום
 - אובייקטים מובחנים בתמונה ע"י קווים בכיוונים שונים
 - שינויי הארה קטנים ורעש לא אמורים להשפיע על זיהוי
- שאלה:** האם ניתן להטמיע אלמנטים אלה ברשת עצבית?

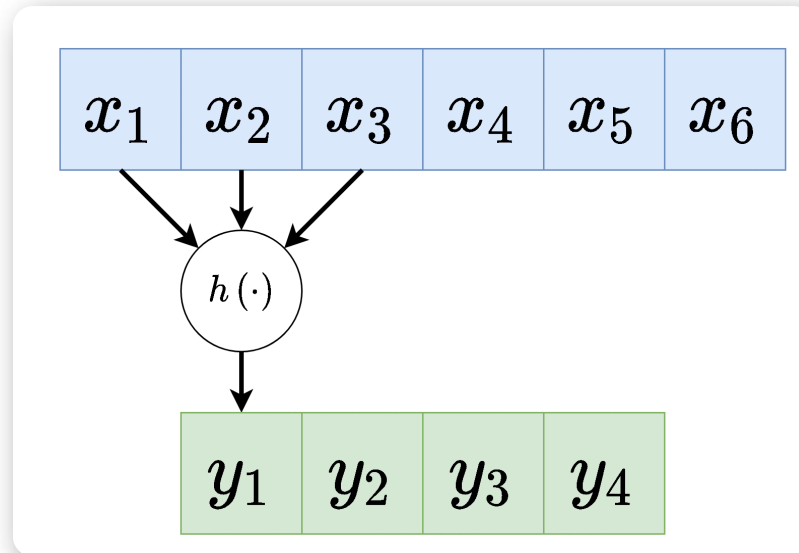
נדגים זאת עבור קלט חד-ממדי

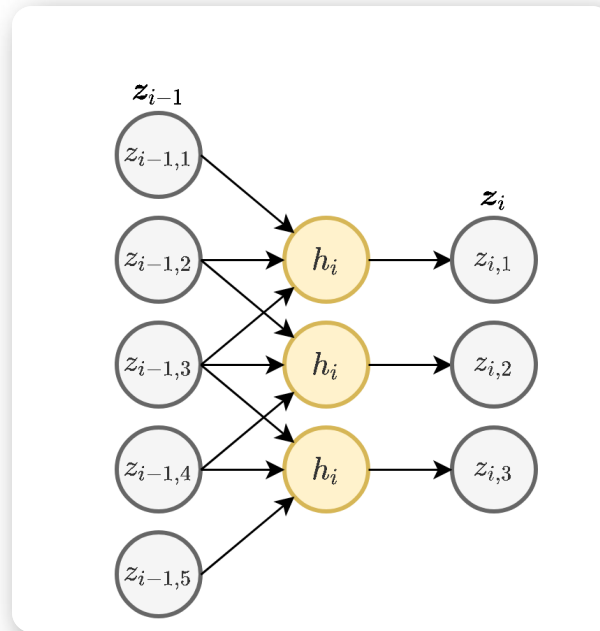


1. כל נוירון בשכבה זו מוזן רק מכמות מוגבלת של ערכים הנמצאים בסביבתו הקרובה.

2. כל הנוירונים בשכבה מסוימת זהים (**weight sharing**).

ניתן להסתכל על הפעולה של שכבת הקונבולוציה באופן הבא:



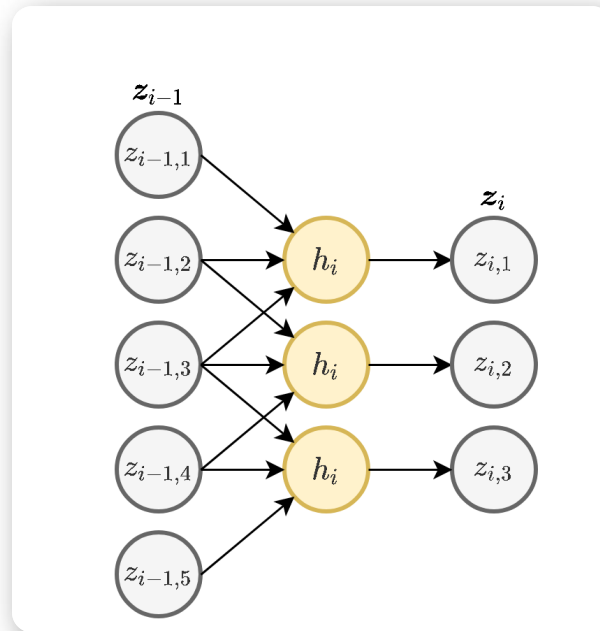


מתמטית השיכבה מבצעת את שלוש הפעולות הבאות:

1. פעולת קרוס-קורלציה (ולא קונבולוציה) בין וקטור הכניסה x ווקטור משקולות w באורך K .

2. הוספת הסט b (אופציונלי).

3. הפעלה של פונקציית הפעלה על וקטור המוצא איבר איבר.

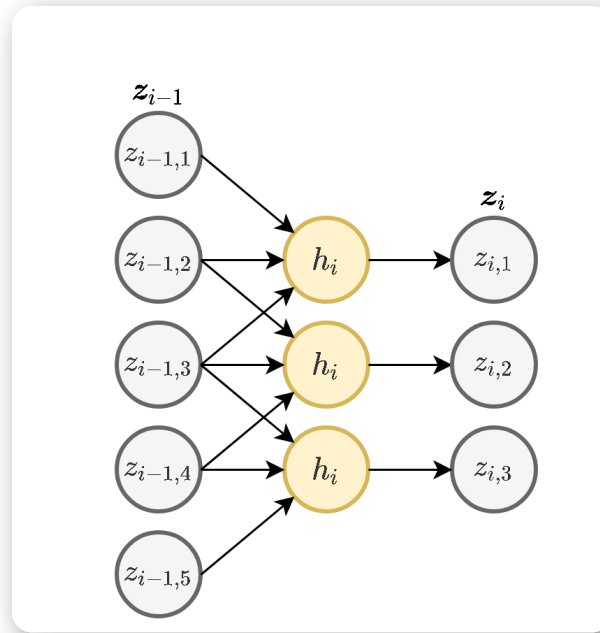


פעולת הקרוס-קורלציה מוגדרת באופן הבא:

$$y_i = \sum_{m=1}^K x_{i+m-1} w_m$$

וקטור המשקולות של שכבת הקונבולוציה w נקרא **גרעין הקונבולוציה (convolution kernel)**.

שכבת קונבולוציה



- גודל המוצא של שכבת הקונבולוציה הוא קטן יותר מהכניסה והוא נתון על ידי $D_{out} = D_{in} - K + 1$.
- בשכבת FC קיימות $D_{in} \times D_{out}$ משקולות ועוד D_{out} איברי היסט.
- בשכבת קונבולוציה יש K משקולות ואיבר היסט בודד.

שכבת קונבולוציה

$$y_i = \sum_{m=1}^K x_{i+m-1} w_m$$

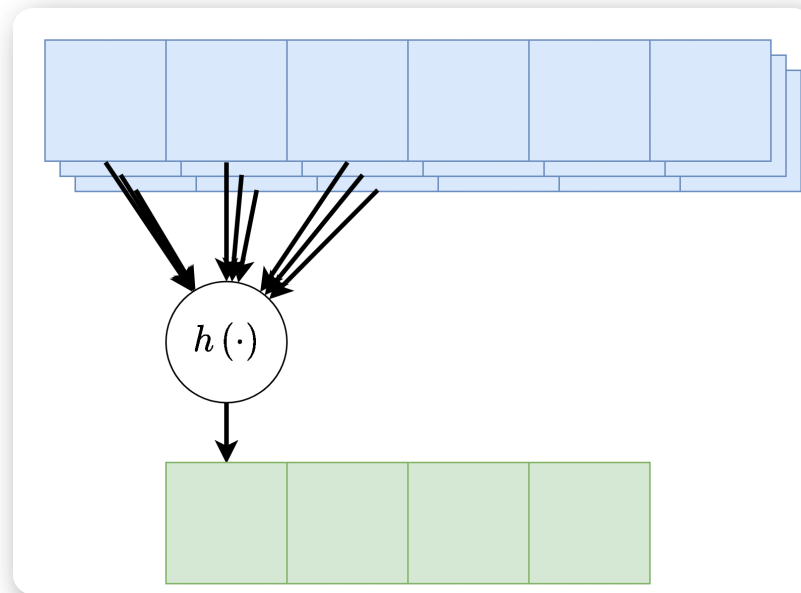


$$K = 4$$

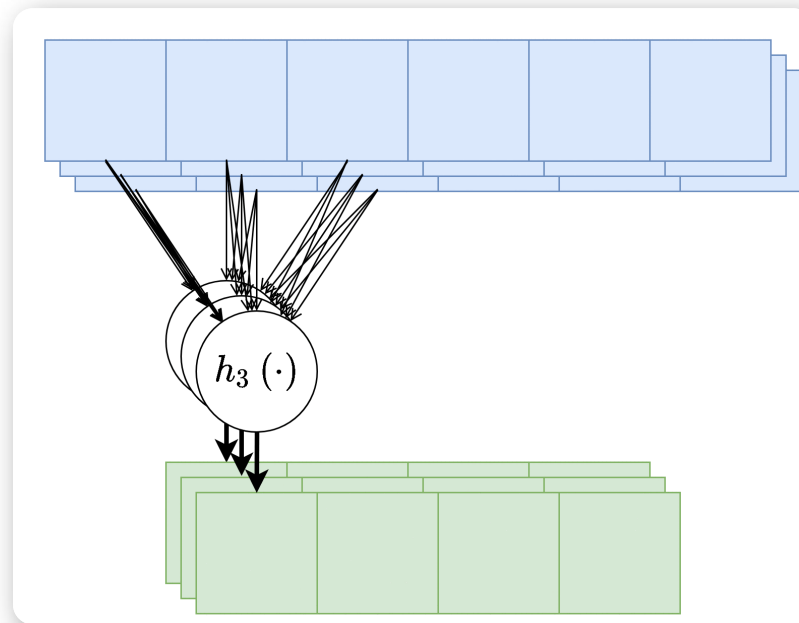
$$D_{out} = D_{in} - K + 1 = 9$$

קלט רב-ערוצי

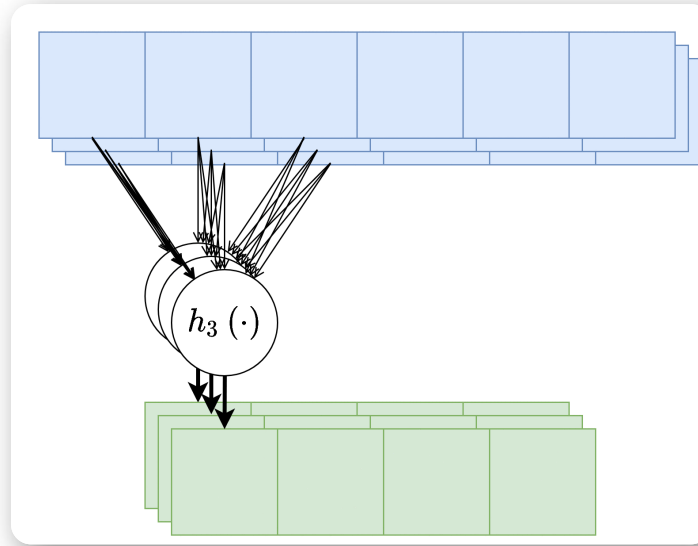
במקרים רבים נרצה ששכבת הקונבולציה תקבל קלט רב ממדי, לדוגמא, תמונה בעלת שלושה ערוצי צבע או קלט שמע ממספר ערוצי הקלטה.



נרצה לרוב להשתמש ביותר מגרעין קונבולוציה אחד, במקרים אלו נייצר מספר ערוצים ביציאה עבור כל אחד מגרעיני הקונבולוציה.



אין שיתוף של משקולות בין ערוצי הפלט השונים.



• C_{in} - מספר ערוצי קלט.

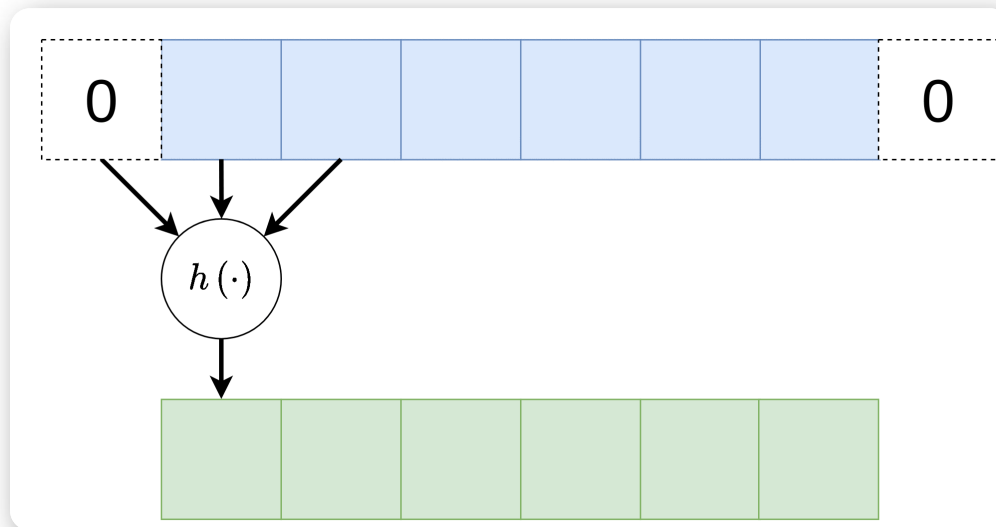
• C_{out} - מספר ערוצי פלט.

• K - גודל הגרעין.

מספר הפרמטרים בשכבה: $\underbrace{C_{in} \times C_{out} \times K}_{\text{the weights}} + \underbrace{C_{out}}_{\text{the bias}}$

Padding - ריפוד

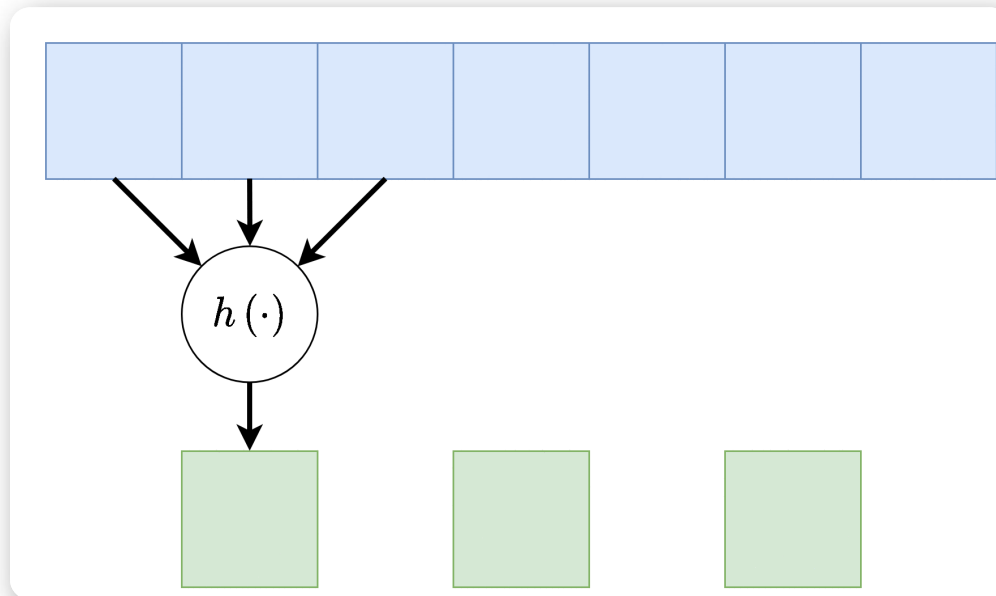
במידה ונרצה לשמור על גודל הוקטור במוצא של שכבת הקונבולוציה, ניתן לרפד את וקטור הכניסה באפסים. לדוגמא:



מאפשר הזזה של הגרעין לאורך התמונה.

Stride - גודל צעד

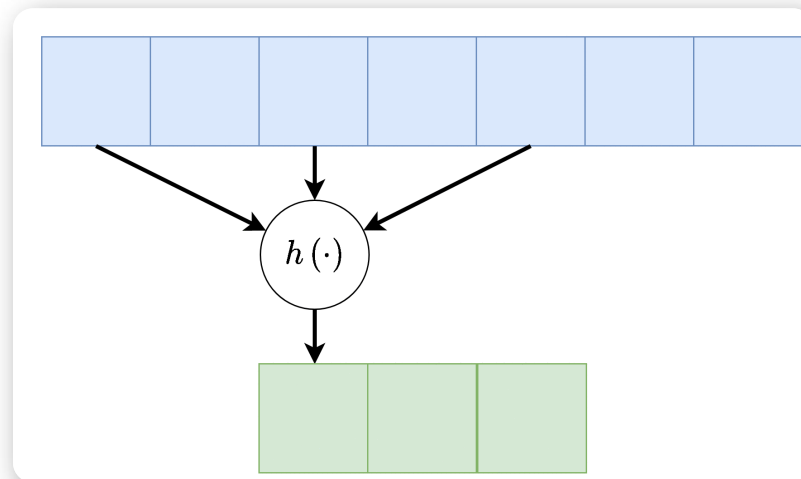
לעיתים נרצה דווקא להקטין את גודל הוקטור במוצא בפקטור מסויים. דרך אחת לעשות זאת היא על ידי דילול המוצא. בפועל אין צורך לחשב את הערכים במוצא שנזרקים ולכן למעשה ניתן לחשב את הקונבולוציה בקפיצות מסויימות המכונות **stride**.



מצמצם עלות חישובית ע"י ביצוע **downsampling**.

Dilation - התרחבות

במקרים אחרים נרצה להגדיל את האיזור שממנו אוסף נירון מסויים את הקלט שלו מבלי להגדיל את מספר הפרמטרים ואת הסיבוכיות החישובית. לשם כך ניתן לדלל את הדרך בה נדגם הקלט על מנת להרחיב את איזור הקלט. אלא אם רשום אחרת, ה dilation של שכבה (הצפיפות בה הכניסה נדגמת) הוא 1.

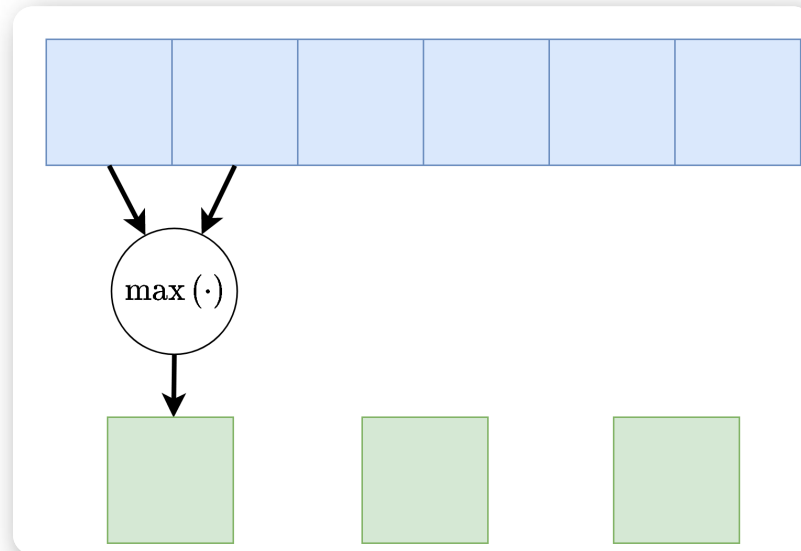


Max / Average Pooling

מוטיבציה: הקטנת הרזולוציה המרחבית, לצורך זיהוי אובייקטים למשל.

שכבות נוספות אשר מופיעות במקרים רבים ברשתות CNN הם שכבות מסוג pooling. שתי שכבות pooling נפוצות הן \max pooling ו average pooling , שכבה זו לוקחת את הממוצע או המקסימום של ערכי הכניסה.

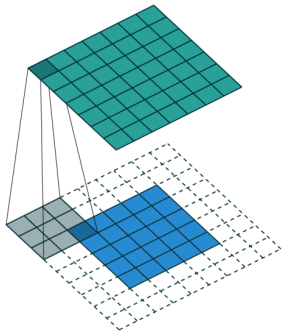
דוגמא זו מציגה \max pooling בגודל 2 עם גודל צעד (stride) גם כן של 2:



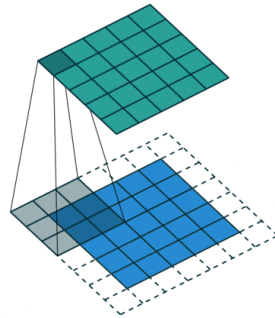
בשכבה זאת אין פרמטרים נלמדים.

2D Convolutional Layer

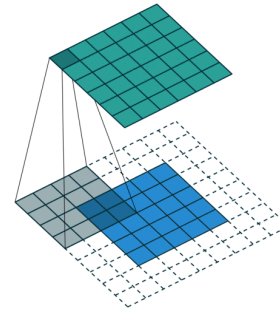
kernel size=3
padding=2
stride=1
dilation=1
(Full padding)



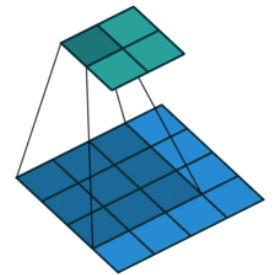
kernel size=3
padding=1
stride=1
dilation=1
(Half padding)



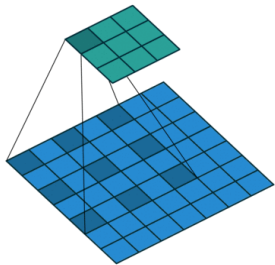
kernel size=4
padding=2
stride=1
dilation=1



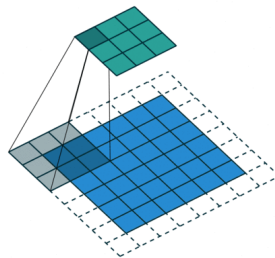
kernel size=3
padding=0
stride=1
dilation=1



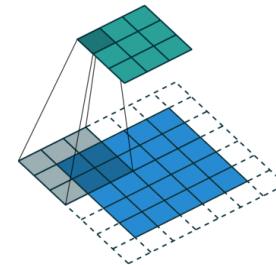
kernel size=3
padding=0
stride=1
dilation=2



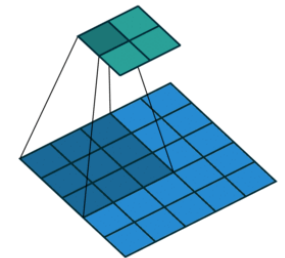
kernel size=3
padding=1
stride=2
dilation=1



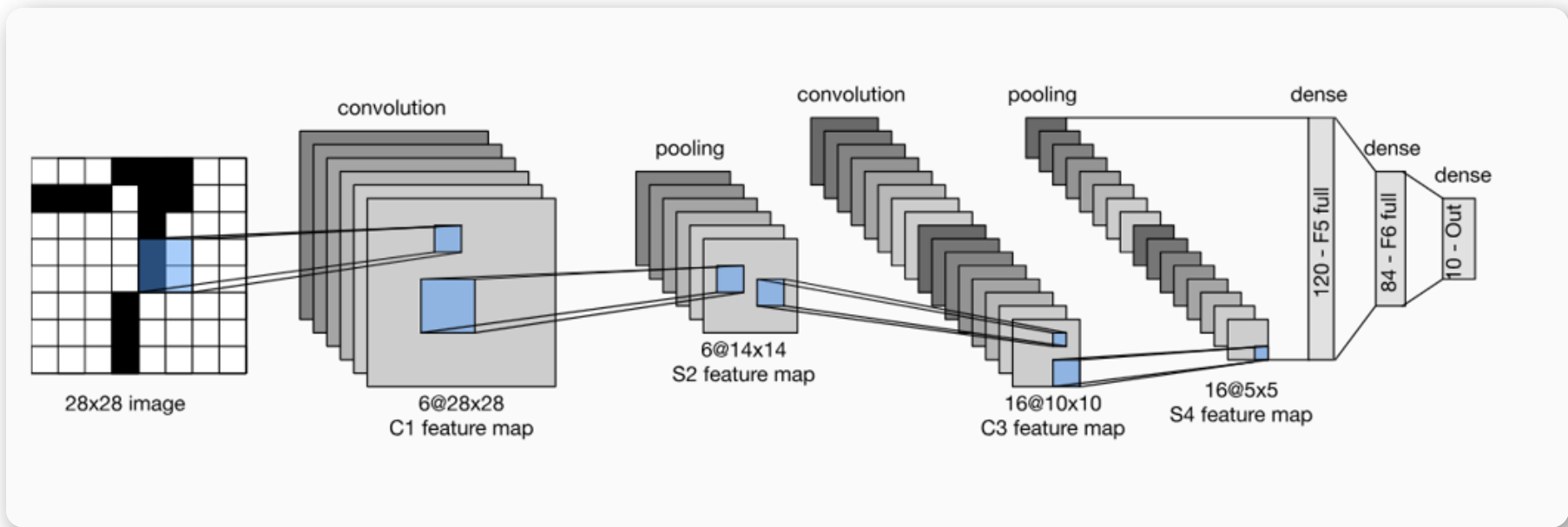
kernel size=3
padding=1
stride=2
dilation=1



kernel size=3
padding=0
stride=2
dilation=1

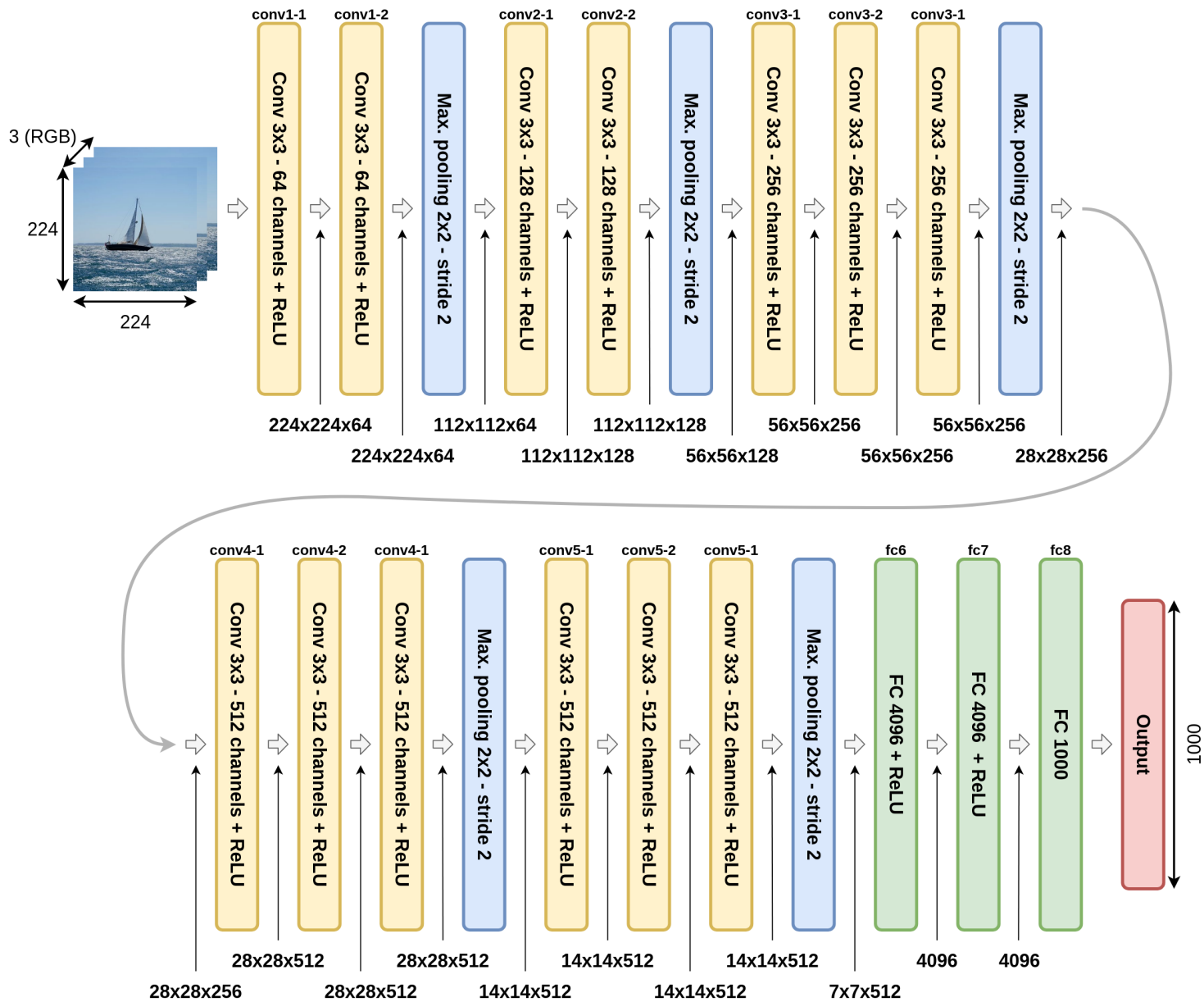


מבנה רשת CNN הראשונה שהוצגה בשנת 1989.

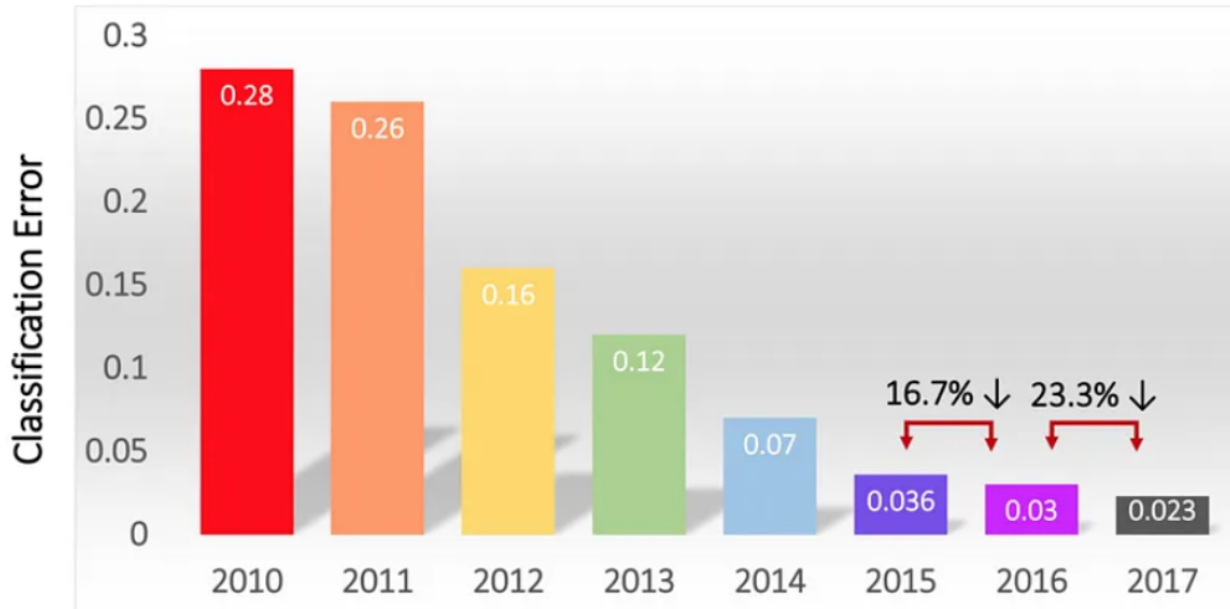


From https://d2l.ai/chapter_convolutional-neural-networks/lenet.html

מבנה רשת CNN



Classification Results (CLS) ImageNet



AlexNet

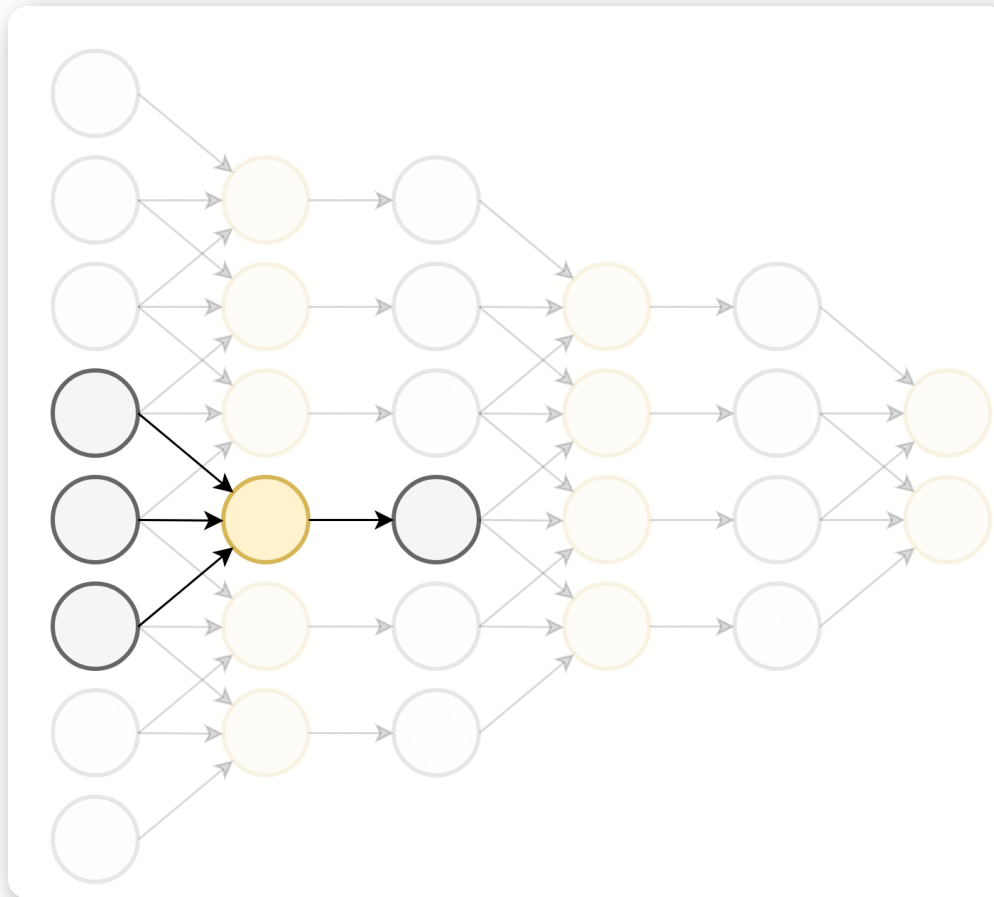
Human 0.05

למה CNN כל כך טובים לבעיות מסוימות?

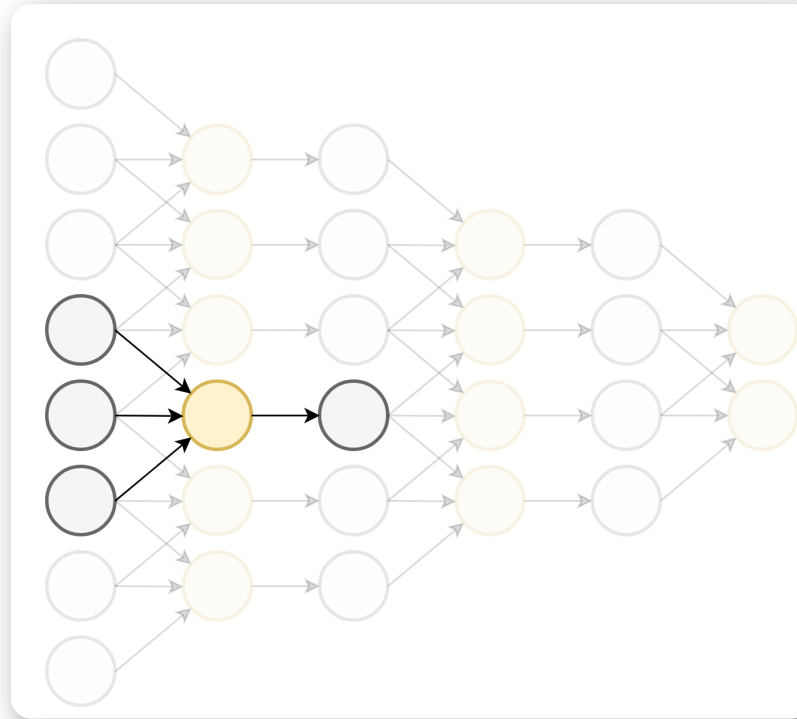
- CNNs מאד טובים בסיווג של תמונות לפי התוכן שלהם.
- הסיבה שבגללה CNNs מתאימים לפתרון של בעיה זו היא בין היתר בגלל ששתי התכונות, שמבדילות שכבת קונבולוציה משכבות FC, מתאימות לייצוג של הפתרון.
- נתייחס לכל אחת משתי התכונות בנפרד.

תלות של כל נוירון רק בסביבה המיידית שלו

כל נוירון רואה רק את הסביבה המיידית שלו ולכן על הרשת
יהיה לנסות לנתח את התמונה בצורה היררכית:



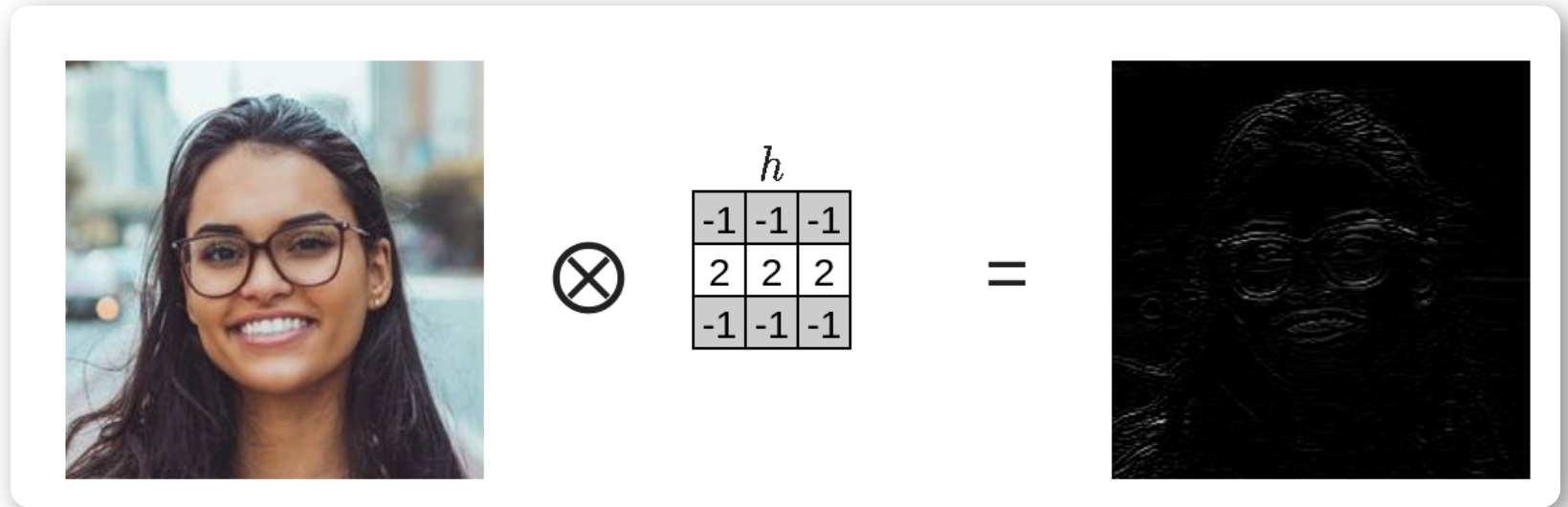
Receptive Field



- הגודל של האיזור שממנו מושפע נירון בשכבה מסוימת נקרא ה **receptive field** שלו.
- לדוגמא, ה **receptive field** של נירון בשכבה השלישית הוא 7.

חילוץ מאפיינים מהתמונה

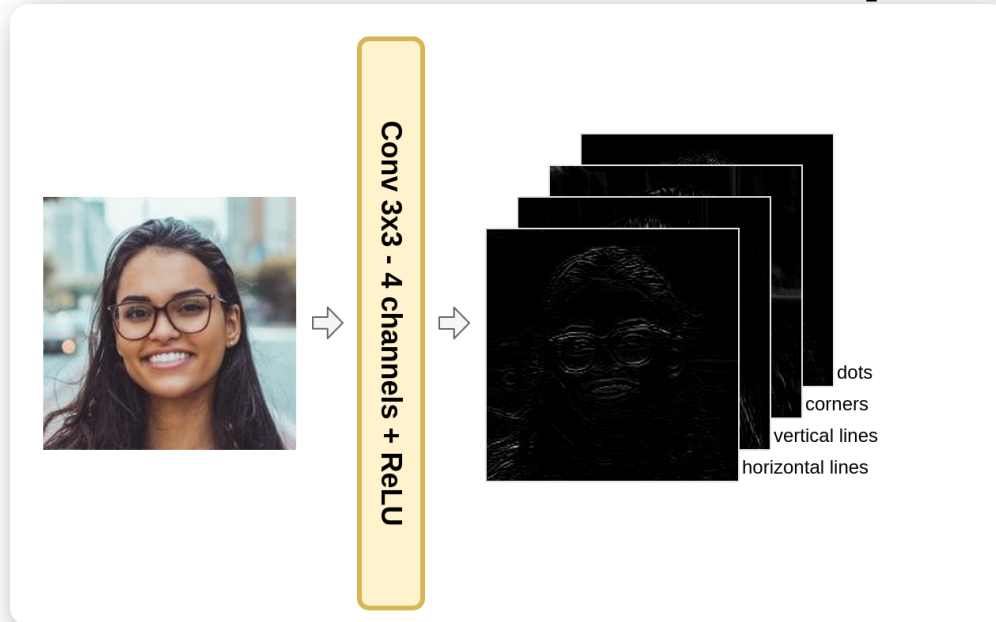
נדגים את הפעולה שמבצעת השכבה הראשונה ברשת אשר מנסה לזהות האם בתמונה מסויימת מופיע פרצוף.



גרעיני הקונבולוציה של השכבות הראשונות יעברו על התמונה ויחפשו תופעות בסיסיות כמו פסים אנכיים, פסים אופקיים, פינות, נקודות קטנות וכו'.

חילוץ מאפיינים מהתמונה

כל גרעין ייצר ערוץ אשר מתאים לתופעה שאותה הוא מחפש:



- השכבות הבאות ברשת יחפשו אובייקטים אשר מורכבים מהתופעות שמצאו השכבות הראשונות.
- לדוגמא נחפש איזורים שמכילים הרבה פסים אנכיים בכדי לזהות שיער, או שני פסים אופקיים סמוכים שעשויים להכיל שפתיים. סוג זה של עיבוד מידע דומה למה שמבצעים במערכת הראייה של יונקים.

Weight sharing

**התכונה הנוספת של שכבת הקונבולוציה הינה שהמשקולות של כל הנוירונים משותפים בין כל הנוירונים באותה השכבה + ערוץ.
למה זה לא מגביל את הרשת:**

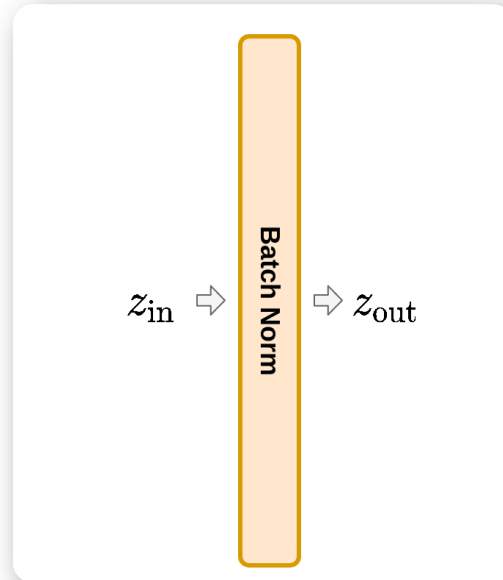
- 1. הסיווג של התמונה לא אמור להיות מושפע אם מזיזים את האובייקט בתמונה מעט לצדדים.**
- 2. הפעולות שהשכבות הראשונות מבצעות, כגון חיפוש קווים אופקיים ואנכיים משותף לכל האיזורים בתמונה.**

סיכום - יתרונות גישת ה CNN

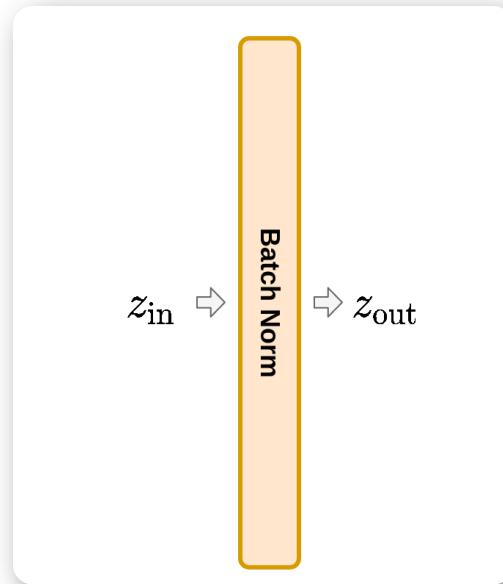
- עובד ישירות על הקלט המקורי - תמונה
- היסטורית - הצלחה משמעותית ראשונה בשיפור ביצועים משמעותי (2012)
- מאפיינים מקומיים תופסים היטב תכונות של תמונות ובשילובם מאפשרים שילוב היררכי
- מאפיינים רלבנטיים נלמדים אוטומטית (באופן היררכי) - רזולוציה משתנה
- אינוריאנטיות להזזות וחסונות בפני שינויים בנתונים
- שיתוף פרמטרים - הקטנה משמעותית ומניעת התאמת יתר, חיסכון בחישוב וזיכרון
- התאמות והרחבות ליישומים אחרים (אודיו, וידאו)

Batch Normalization ## (לא למבחן)

- אחת הבעיות בעבודה עם רשתות עמוקות הינה מצב שבו הערכים במוצאים של השכבות הם מסדר גודל שונה.
- הדבר משפיע על הגרדיאנטים ומקשה על הבחירה של גודל הצעד.
- דרך אחת לנסות ולהבטיח כי המוצאים יהיו בערך מאותו סדר גודל הינה על ידי הוספה של שכבה בשם `batch normalization`.

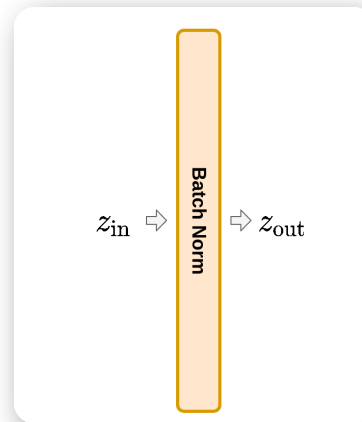


Batch Normalization (לא למבחן)



- מנסה לנרמל את הערכים אשר עוברים דרכה (מביאה את התוחלת של הערכים ל 0 ואת הסטיית תקן ל 1).
- עושה זאת על ידי חישוב התוחלת וסטיית התקן האמפירית של הערכים על פני ה batch.

(לא למבחן) Batch Normalization



$$\mu = \frac{1}{M} \sum_{i=1}^M z_{in}^{(i)}$$

$$\sigma^2 = \frac{1}{M} \sum_{i=1}^M (z_{in}^{(i)} - \mu)^2$$

המוצא של השכבה יהיה:

$$z_{out} = \frac{z_{in} - \mu}{\sigma + \epsilon}$$

Batch Normalization (לא למבחן)

לרוב השכבה תכיל גם טרנספורמציה לינארית נלמדת עם פרמטרים γ ו β :

$$z_{\text{out}} = \frac{z_{\text{in}} - \mu}{\sigma + \epsilon} \cdot \gamma + \beta$$

כאשר γ ו β הוא וקטורים באורך של z והמכפלה עם γ היא איבר איבר.

במהלך הלימוד מחזיקים ממוצע נע (**exponential moving average**) של הערכים μ ו σ ובסוף שלב הלימוד מקבעים את הערכים שלהם ואלו הערכים שבהם הרשת תשתמש לאחר שלב האימון.