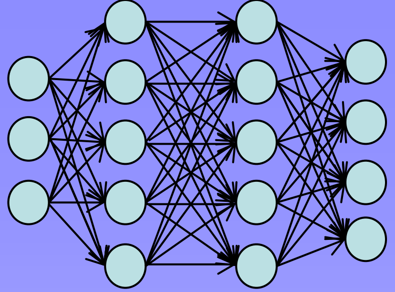


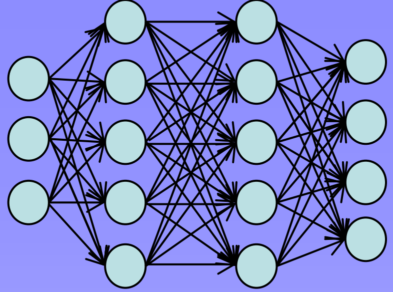
פונקציית המחיר עבור רשת עצבית

פונקציית המחיר עבור הרגרסיה הלוגיסטית:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log((h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)})) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^{\,2}$$

(איבר הרגולריזציה
ללא θ_0)

עבור רשת עצבית נבצע הכללה של פונקציית המחיר.



פונקציית המחיר עבור רשת עצבית

- עבור רשת עצבית נבצע הכללה של פונקציית המחיר:

$$h_\Theta(x) \in R^K \quad (h_\Theta(x))_i = i^{th}\ output$$

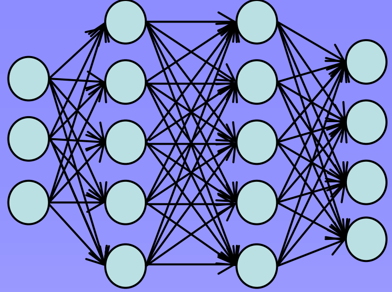
$$J(\theta) = -\frac{1}{m}\left(\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log((h_\theta(x^{(i)}))_k + (1-y_k^{(i)})\log(1-h_\theta(x^{(i)})_k) \right)$$

$$+\underbrace{\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{S_l}\sum_{j=1}^{n}(\Theta_{ji}^{\ l})^2}_{\text{איבר הרגולריזציה}}$$

עבור כל יחידות הפלט

עבור כל הדוגמאות

בהמשך נבצע אופטימיזציה עבור פונקציית המחיר הנ״ל



אלגוריתם ה- Backpropagation

נדבר על אלגוריתם למיזעור פונקציית המחיר: $\min_{\Theta} J(\Theta)$

כדי למזער את פונקציית המחיר נצטרך לחשב את: $J(\Theta)$

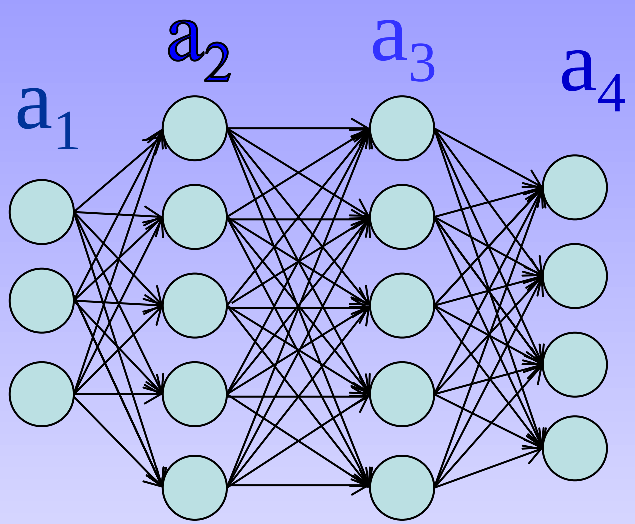
$$\frac{\partial}{\partial \Theta_{ij}^{\,l}} J(\Theta), \quad \Theta_{ij}^{\,l} \in R$$

איך נחשב את הנגזרת החלקית לפי $\Theta_{ij}^{\,l}$?

אלגוריתם ה- Backpropagation

נניח כי יש רק דוגמת אימון אחת: $\{x, y\}$

ראשית נבצע Forward propagation:



$$a^{(1)} = x$$

$$z^{(2)} = \Theta^{(1)} a^{(1)}$$

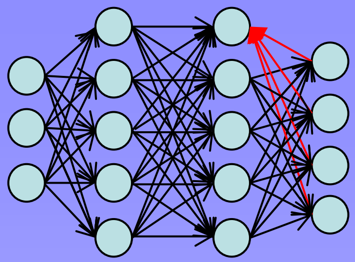
$$a^{(2)} = g(z^{(2)}) \ \ (add\ a_0^{(2)})$$

$$z^{(3)} = \Theta^{(2)} a^{(2)}$$

$$a^{(3)} = g(z^{(3)}) \ \ (add\ a_0^{(3)})$$

$$z^{(4)} = \Theta^{(3)} a^{(3)}$$

$$a^{(4)} = h_\Theta(x) = g(z^{(4)})$$



אלגוריתם ה- Backpropagation

כדי למזער את פונקציית המחיר $J(\Theta)$ על-ידי שינוי הפרמטרים (המשקלות) הסינפטיים נחשב עבור כל יחידה, עבור כל פרמטר (משקל) את הנגזרת החלקית הבאה:

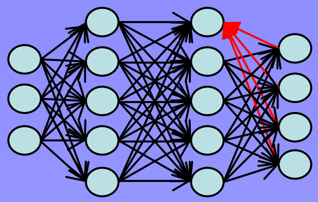
$$\frac{\partial}{\partial \theta_{ij}^{\;l-1}} J(\Theta)$$

ברור ש- $J(\Theta)$ מושפע על-ידי $\theta_{ij}^{\;l-1}$ דרך המשתנה $z_i^{(l)}$, סכום הקלטים ליחידה ה- i:

$$z_i^{(l)} = \sum_{j=0}^{S_{l-1}} \theta_{ij}^{\;l-1} a_j^{(l-1)}$$

ולכן לפי כלל השרשרת:

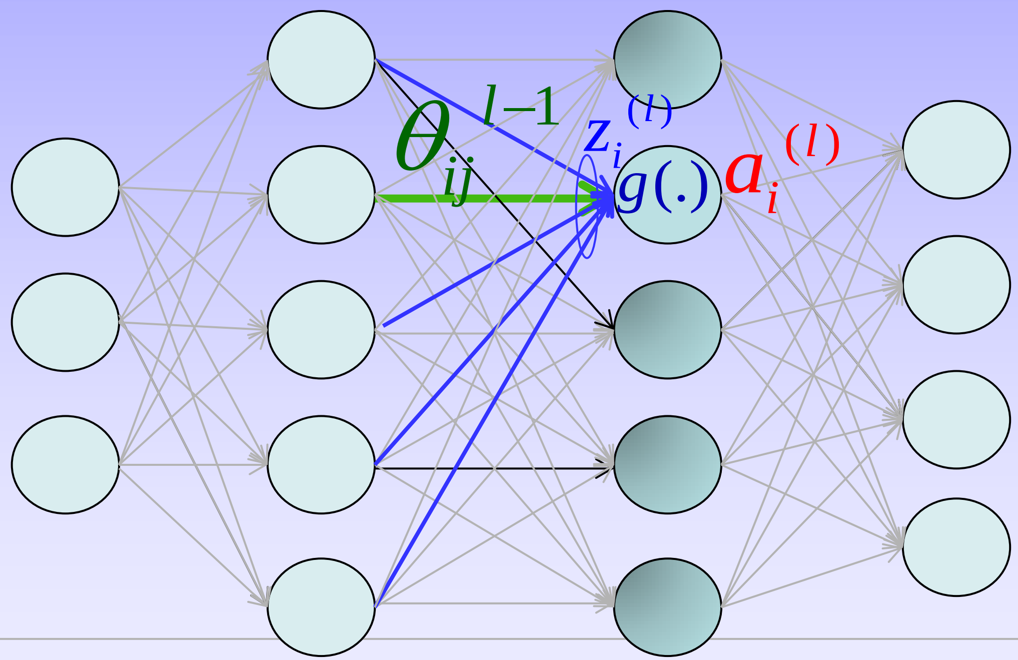
$$\frac{\partial}{\partial \theta_{ij}^{\;l-1}} J(\Theta) = \frac{\partial J(\Theta)}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial \theta_{ij}^{\;l-1}}$$



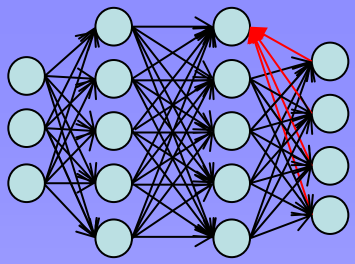
אלגוריתם ה- Backpropagation

ברור ש- $J(\Theta)$ מושפע על-ידי $\partial\theta_{ij}^{\,l-1}$ דרך המשתנה $z_i^{(l)}$, סכום הקלטים ליחידה ה- i :

$$z_i^{(l)} = \sum_{r=0}^{S_{l-1}} \theta_{ir}^{\,l-1} a_r^{(l-1)}$$



$$\frac{\partial}{\partial\theta_{ij}^{\,l-1}} J(\Theta) = \frac{\partial J(\Theta)}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial\theta_{ij}^{\,l-1}}$$



אלגוריתם ה- Backpropagation

אם כן נגדיר:

$$\delta_i^{(l)} \equiv \frac{\partial J(\Theta)}{\partial z_i^{l}}$$

בדרך כלל ה- δ -ות נקראות השגיאות מסיבות אותן נראה בהמשך.

לפיכך נרשום:

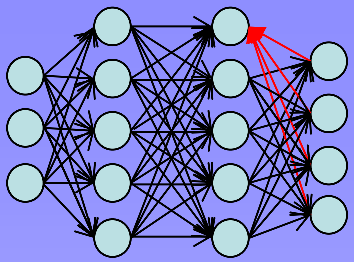
$$\frac{\partial}{\partial \theta_{ij}^{l-1}} J(\Theta) = \delta_i^{(l)} \cdot \frac{\partial z_i^{(l)}}{\partial \theta_{ij}^{l-1}}$$

אבל:

$$\frac{\partial z_i^{(l)}}{\partial \theta_{ij}^{l-1}} = \frac{\partial}{\partial \theta_{ij}^{l-1}} \sum_{r=0}^{S_{l-1}} \theta_{ir}^{l-1} a_r^{(l-1)} = a_j^{(l-1)}$$

ולכן:

$$\frac{\partial}{\partial \theta_{ij}^{l-1}} J(\Theta) = \delta_i^{(l)} \cdot a_j^{(l-1)}$$

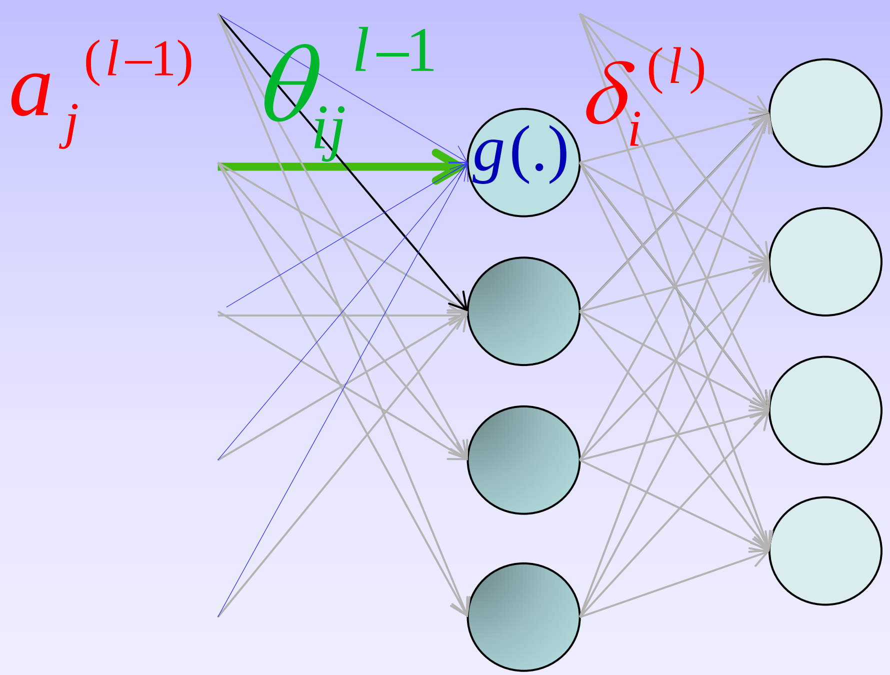


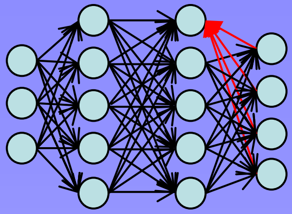
אלגוריתם ה- Backpropagation

כפי שהראינו:

$$\frac{\partial}{\partial \theta_{ij}^{\,l-1}} J(\Theta) = \delta_i^{(l)} \cdot a_j^{(l-1)}$$

כלומר הנגזרת הדרושה היא מכפלה של ה- δ של היחידה בקצה הפלט של הפרמטר (או המשקל הסינפטי) בערך של a של היחידה בחלק הקלט של הפרמטר אותו מעדכנים.





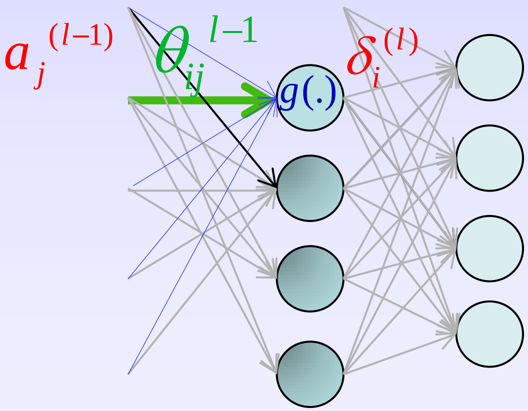
אלגוריתם ה- Backpropagation

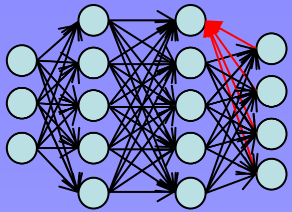
לפיכך, כדי לחשב את הנגזרות, צריך לחשב את הערך של $\delta_i^{(l)}$
לכל יחידת hidden ברשת, ואז להפעיל את:

$$\frac{\partial}{\partial \theta_{ij}^{\,l-1}} J(\Theta) = \delta_i^{(l)} \cdot a_j^{\,(l-1)}$$

נניח שעבור היחידות בשכבת הפלט (output): $\delta_k^{(L)} = a_k^{(L)} - y_k$

או בצורה וקטורית: $\delta^{(L)} = a^{(L)} - y$





אלגוריתם ה- Backpropagation

נחשב את הערך של ה- δ עבור יחידת הפלט, היחידה ה- L:

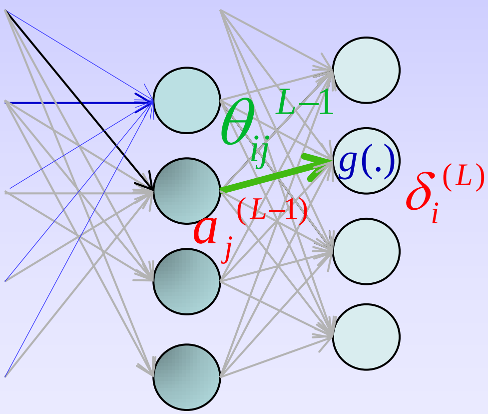
$$\delta_i^{(L)}$$

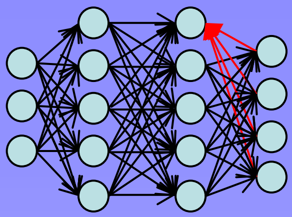
עבור כל יחידת פלט:

$$\frac{\partial}{\partial \theta_{ij}^{L-1}} J(\Theta) = \delta_i^{(L)} \cdot a_j^{(L-1)}$$

כאשר:

$$\delta_i^{(L)} \equiv \frac{\partial J(\Theta)}{\partial z_i^{L}}$$





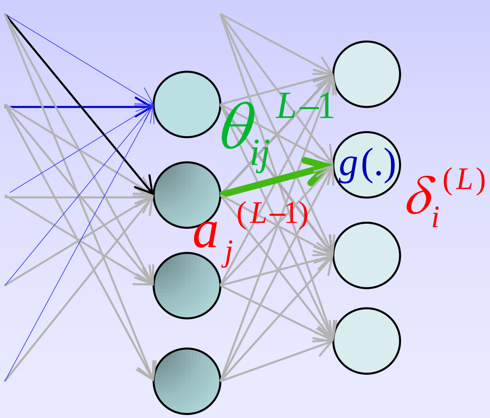
אלגוריתם ה- Backpropagation

נוכיח כי עבור היחידות בשכבת הפלט (output):

$$\delta_i^{(L)} = a_i^{(L)} - y_i$$

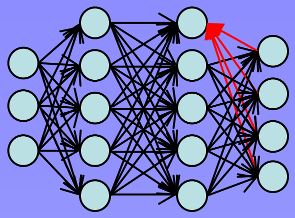
או בצורה וקטורית:

$$\delta^{(L)} = a^{(L)} - y$$



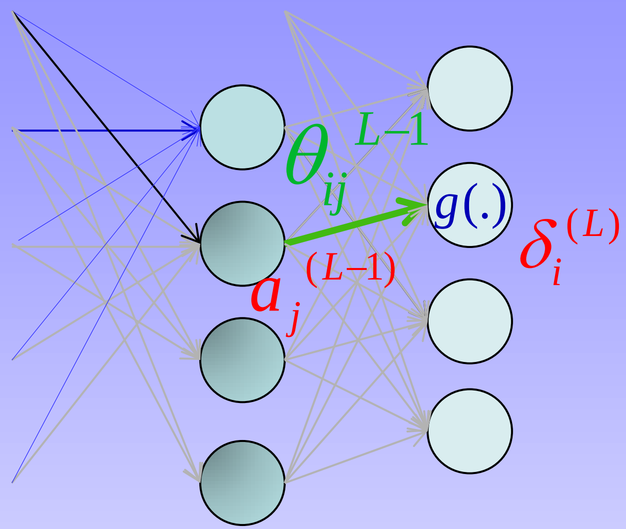
כאשר:

$$\delta_i^{(L)} \equiv \frac{\partial J(\Theta)}{\partial z_i^{L}}$$



אלגוריתם ה- Backpropagation

נבחן את ה- δ ביחידת הפלט:

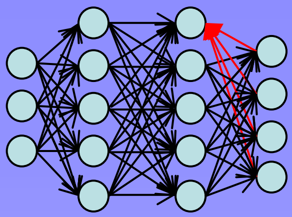


נוכיח כי עבור היחידות בשכבת הפלט (output):

$$\delta_i^{(L)} = a_i^{(L)} - y_i$$

על-ידי גזירה לפי ההגדרה:

$$\delta_i^{(L)} \equiv \frac{\partial J(\Theta)}{\partial z_i^{L}}$$



אלגוריתם ה- Backpropagation

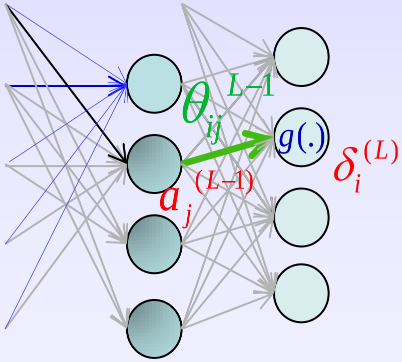
נגזור את פונקציית המחיר לפי z_i :

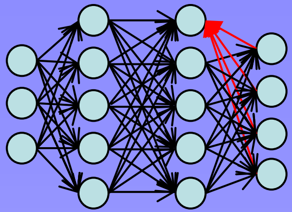
כאשר:

$$\delta_i^{(L)} \equiv \frac{\partial J(\Theta)}{\partial z_i^{\;L}}$$

$$h_\Theta(x) \in R^K \quad (h_\Theta(x))_i = i^{th}\ output$$

$$J(\theta) = \sum_{k=1}^{K}\left(y_k^{(i)} \log((h_\theta(x^{(i)}))_k + (1-y_k^{(i)})\log(1-h_\theta(x^{(i)})_k)\ \right)$$





אלגוריתם ה- Backpropagation

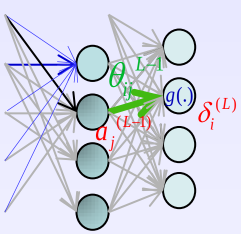
ונקבל (תרגיל):

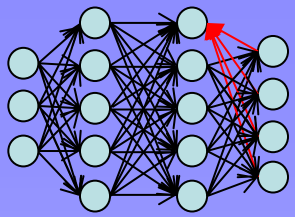
$$\delta_i^{(L)} \equiv \frac{\partial J(\Theta)}{\partial \mathbf{z}_i^{\,L}} =$$

$$\frac{\partial}{\partial \mathbf{z}_i^{\,L}}(-1)\left(\sum_{k=1}^{K}\left(y_k \log((h_\theta(x))_k + (1-y_k)\log(1-h_\theta(x)_k)\ \right)\right) =$$

$$\frac{\partial}{\partial \mathbf{z}_i^{\,L}}(-1)\left(\sum_{k=1}^{K}\left(y_k \log(a_k^{\,(L)}) + (1-y_k)\log(1-a_k^{\,(L)}\ \right)\right) =$$

$$\frac{\partial}{\partial \mathbf{z}_i^{\,L}}(-1)\left(\sum_{k=1}^{K}\left(y_k \log\left(g\left(z_k^{\,(L)}\right)\right) + (1-y_k)\log\left(1-g\left(z_k^{\,(L)}\right)\right)\right)\right)$$





אלגוריתם ה- Backpropagation

ונקבל (בדקו):

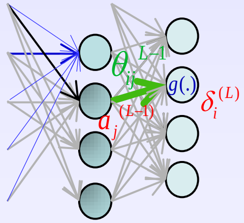
$$= -y_i \cdot \frac{1}{g\left(z_i^{(L)}\right)} g\left(z_i^{(L)}\right)\left(\left(1 - g\left(z_i^{(L)}\right)\right)\right)$$

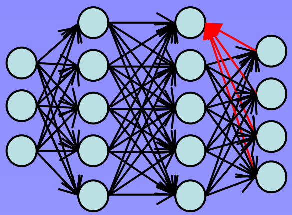
$$-\left(1 - y_i\right) \cdot \frac{-1}{\left(1 - g\left(z_i^{(L)}\right)\right)} \left(g\left(z_i^{(L)}\right)\right)\left(1 - g\left(z_i^{(L)}\right)\right)$$

$$= -y_i + y_i g\left(z_i^{(L)}\right) + g\left(z_i^{(L)}\right) - y_i g\left(z_i^{(L)}\right)$$

$$= g\left(z_i^{(L)}\right) - y_i$$

$$= a_i^{(L)} - y_i$$





אלגוריתם ה- Backpropagation

ונקבל :

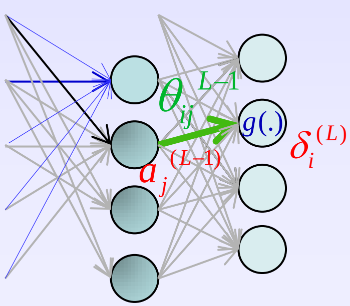
$$\delta_i^{(L)} \equiv \frac{\partial J(\Theta)}{\partial z_i^{L}} =$$

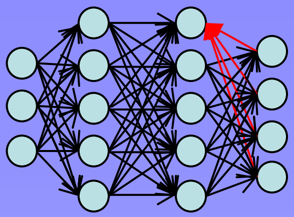
$$\frac{\partial}{\partial z_i^{L}} \sum_{k=1}^{K} \Big(y_k \log((h_\theta(x))_k + (1-y_k)\log(1-h_\theta(x)_k) \Big) =$$

$$= g\left(z_i^{(L)}\right) - y_i = a_i^{(L)} - y_i$$

כלומר:

$$\delta_i^{(L)} = a_i^{(L)} - y_i$$



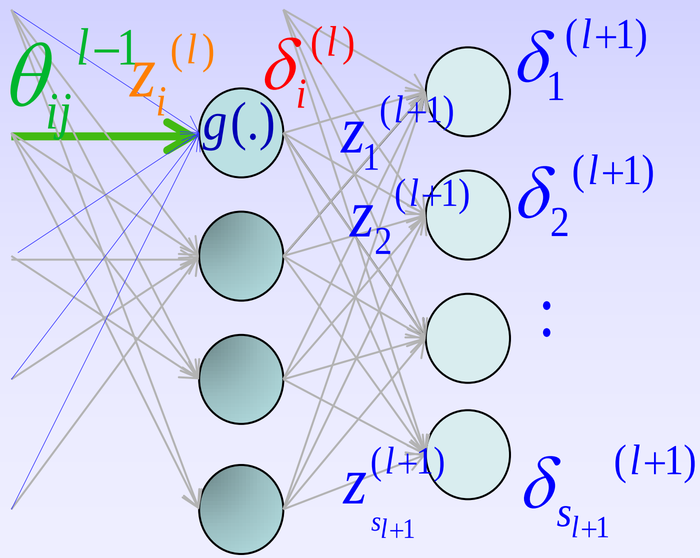


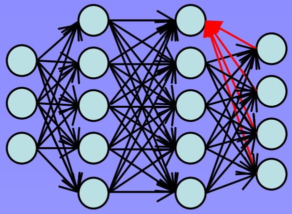
אלגוריתם ה- Backpropagation

כדי לחשב את הערך של ה- δ-ות ביחידות ה- hidden נשתמש בכלל השרשרת עבור נגזרות חלקיות:

$$\delta_i^{(l)} = \frac{\partial J(\Theta)}{\partial z_i^{(l)}} = \sum_{k=1}^{s_{l+1}} \frac{\partial J(\Theta)}{\partial z_k^{(l+1)}} \cdot \frac{\partial z_k^{(l+1)}}{\partial z_i^{(l)}} \quad l = L-1, L-2, \ldots$$

כאשר הסכום רץ על כל היחידות k בשכבת הפלט (output) להן היחידה i מחוברת.



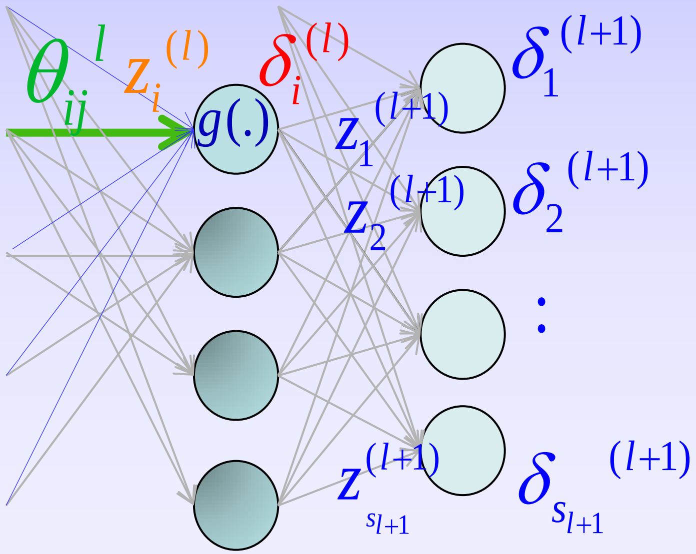


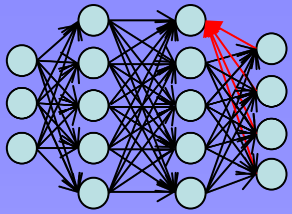
אלגוריתם ה- Backpropagation

היחידות המסומנות ב- k יכולות להיות יחידות hidden אחרות או יחידות פלט. בכתיבת המשוואה:

$$\delta_i^{(l)} = \frac{\partial J(\Theta)}{\partial z_i^{(l)}} = \sum_{k=1}^{s_{l+1}} \frac{\partial J(\Theta)}{\partial z_k^{(l+1)}} \cdot \frac{\partial z_k^{(l+1)}}{\partial z_i^{(l)}} \qquad l = L-1, L-2, \ldots$$

משתמשים בעובדה שהשינויים ב- $z_i^{(l)}$ גורמים לשינויים בפונקציית המחיר J רק דרך שינויים במשתנים $z_k^{(l+1)}$





אלגוריתם ה- Backpropagation

נשתמש בעובדות הבאות:

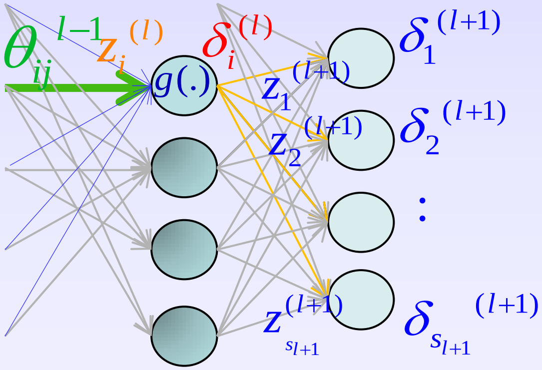
$$z_i^{(l)} = \sum_{j=0}^{S_{l-1}} \theta_{ij}^{\ l} a_j^{(l-1)}$$

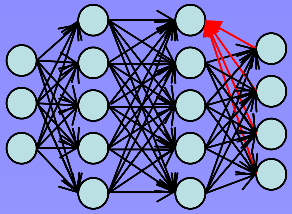
$$a_i^{(l)} = g(z_i^{(l)})$$

כדי לקבל:

$$\frac{\partial z_k^{(l+1)}}{\partial z_i^{(l)}} = \frac{\partial}{\partial z_i^{(l)}} \sum_{j=0}^{S_{l-1}} \theta_{kj}^{\ l+1} a_j^{(l)} = \frac{\partial}{\partial z_i^{(l)}} \sum_{j=0}^{S_{l-1}} \theta_{kj}^{\ l} g(z_j^{(l)})$$

$$= g'(z_i^{(l)}) \cdot \theta_{ki}^{\ l}$$



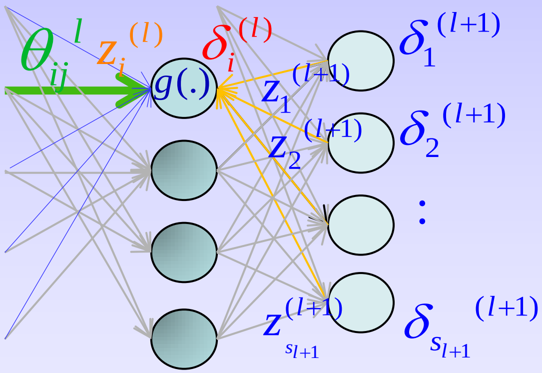


אלגוריתם ה- Backpropagation

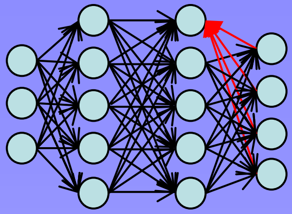
ולכן:

$$\delta_i^{(l)} = \sum_{k=1}^{S_{l+1}} \theta_{ki}^{(l)} \delta_k^{(l+1)} g'(z_i^{(l)})$$

$$= g'(z_i^{(l)}) \sum_{k=1}^{S_{l+1}} \theta_{ki}^{(l)} \delta_k^{(l+1)}$$



נוסחה זו נקראת **נוסחת ה- backpropagation**

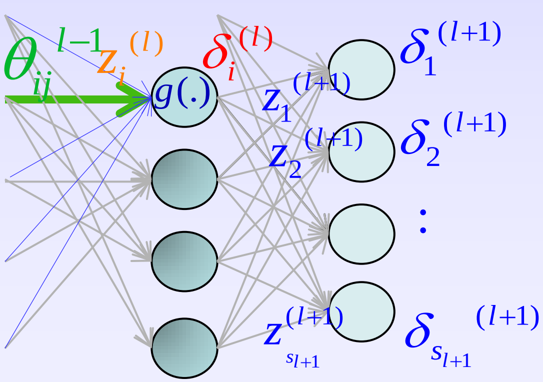


אלגוריתם ה- Backpropagation

כלומר כדי לקבל את ה- δ -ות של יחידת hidden צריך להזרים אחורנית (backward propagate) את ה- δ -ות של יחידות גבוהות יותר ברשת כפי שמתואר בציור.

נבחין שבזרימה אחורנית, אינדקס הסכימה של ה- δ -ות הוא ראשון בעוד שבזרימה קדימה (forward propagation) אינדקס הסכימה הוא שני.

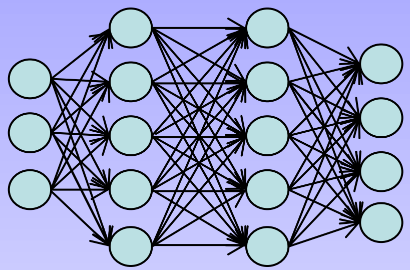
מאחר ואנו כבר יודעים את ה- δ -ות של היחידות בשכבת הפלט, אפשר לדעת את הפלט עבור כל יחידות ה- hidden על-ידי הפעלה רקורסיבית של נוסחת ה- backpropagation, ללא תלות בטופולוגיה של הרשת.



חישוב הגרדיאנט: אלגוריתם ה- Backpropagation

אינטואיציה: $\delta_j^{(l)}$ - ה"שגיאה" של היחידה ה- j בשכבה ה- l.

כלומר ה- $\delta_j^{(l)}$ תייצג באיזשהו אופן את השגיאה עבור היחידה הנ"ל.



$a_j^{(l)}$ - האקטיבציה של היחידה ה- j בשכבה ה- l.

עבור כל יחידה ביציאה (בשכבת הפלט, layer L=4):

$$\delta_j^{(4)} = a_j^{(4)} - y_j = (h_\Theta(x))_j - y_j$$

באופן וקטורי:

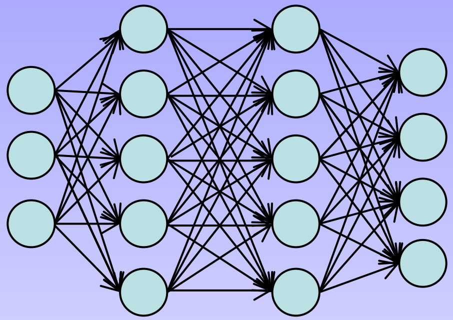
$$\delta_j^{(4)} = a^{(4)} - y$$

וקטורים שהממד שלהם הוא מספר היחידות בשכבת הפלט

חישוב הגרדיאנט: אלגוריתם ה- Backpropagation

אינטואיציה: $\delta_j^{(l)}$ - ה"שגיאה" של היחידה ה- j בשכבה ה- l.

כלומר ה- $\delta_j^{(l)}$ תייצג באיזשהו אופן את השגיאה עבור היחידה הנ"ל.



$a_j^{(l)}$ - האקטיבציה של היחידה ה- j בשכבה ה- l.

עבור כל יחידה בשכבות ה- hidden, (layer l=2,3):

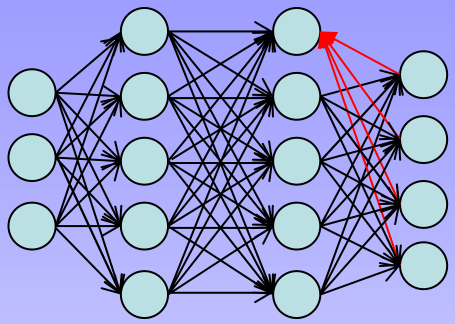
$$\delta^{(3)} = (\Theta^{(3)})^T \delta^{(4)} .* g'(z^{(3)})$$

$$where: g'(z^{(3)}) = a^{(3)} .* (1 - a^{(3)})$$

$$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} .* g'(z^{(2)})$$

$$where: g'(z^{(2)}) = a^{(2)} .* (1 - a^{(2)})$$

חישוב הגרדיאנט: אלגוריתם ה- Backpropagation



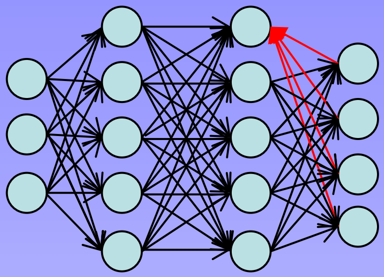
הערה: אין ביטוי ל $\delta^{(1)}$ - זוהי הכניסה או הקלט.

מזרימים את השגיאה אחורנית מהיציאה לכניסה דרך השכבות השונות.

ניתן להוכיח מתימטית כי בהתעלם מביטוי הרגולריזציה

$$\frac{\partial}{\partial \Theta_{ij}^{\ l}} J(\Theta) = a_j^{(l)} \delta_i^{(l+1)} \quad if \quad \lambda = 0$$

חישוב הגרדיאנט: אלגוריתם ה- Backpropagation



קבוצת האימון: $\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),...(x^{(m)},y^{(m)})\}$
סה"כ m דוגמאות.

Set $\Delta_{ij}^{(l)}=0 \;\; (\,for\ all\ l,i,j\ use\ to\ compute\ \frac{\partial}{\partial\theta_{ij}^{(l)}}J(\theta))$

For i=1:m

set $a^{(1)}=x^{(i)}$

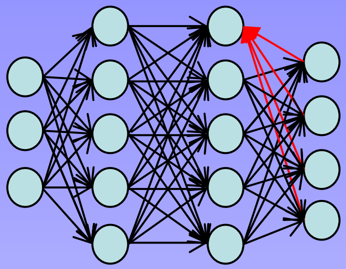
perform forward propagation to compute $a^{(l)}$ for l=2,3,…,L

using $y^{(i)}$ compute $\delta^{(L)}=a^{(L)}-y^{(i)}$

compute $\delta^{(L-1)},\delta^{(L-2)},\ldots,\delta^{(2)}$

$\Delta_{ij}^{(l)}=\Delta_{ij}^{(l)}+a_j^{(l)}\delta_i^{(l+1)}$ or in matrix form $\Delta^{(l)}:=\Delta^{(l)}+\delta^{(l+1)}(a^{(l)})^T$

חישוב הגרדיאנט: אלגוריתם ה- Backpropagation



קבוצת האימון: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(m)}, y^{(m)})\}$

סה"כ m דוגמאות.

לבסוף מחשבים את

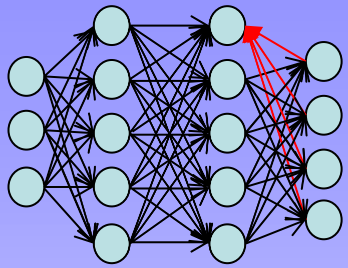
$$D_{ij}^{(l)} = \frac{1}{m}\Delta_{ij}^{(l)} + \lambda\theta_{ij}^{(l)} \; if \;\; j \neq 0$$

$$D_{ij}^{(l)} = \frac{1}{m}\Delta_{ij}^{(l)} \; if \;\; j = 0$$

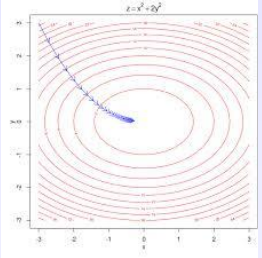
אפשר להוכיח כי

$$\frac{\partial}{\partial\theta_{ij}^{(l)}} J(\theta)) = D_{ij}^{(l)}$$

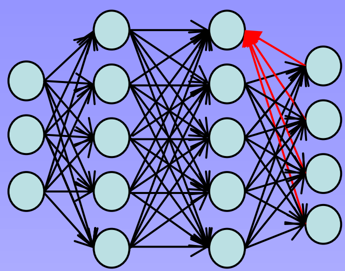
חישוב הגרדיאנט: אלגוריתם ה- Backpropagation



- כדי לאמן את הרשת צריך לחשב את ערכי הפרמטרים אותם נכנה הפרמטרים הסינפטיים.
- הפרמטרים מחושבים כך שפונקציית המחיר $J(\theta)$ ממוזערת. (פונקציית המחיר $J(\theta)$ תלוייה בערכי y – הערך הרצוי בפלט, ובערכי $h_\theta(x)$ – ערכי שכבת הפלט).
- ברור שהפונקציה J תלויה בפרמטרים או במשקלות הסינפטיים, ושזוהי תלות לא ליניארית בשל אופי הרשת.
- לפיכך המזעור יושג בשיטות איטרטיביות.
- נאמץ את שיטת ה- gradient descent (זוהי השיטה הנפוצה ביותר, קיימות שיטות נוספות)



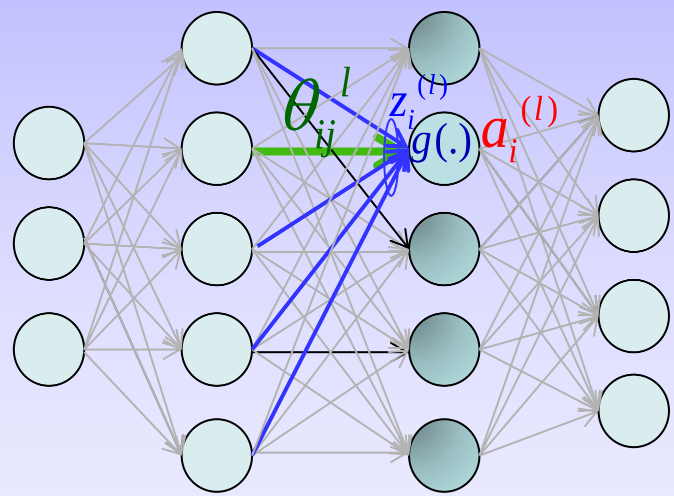
חישוב הגרדיאנט: אלגוריתם ה- Backpropagation



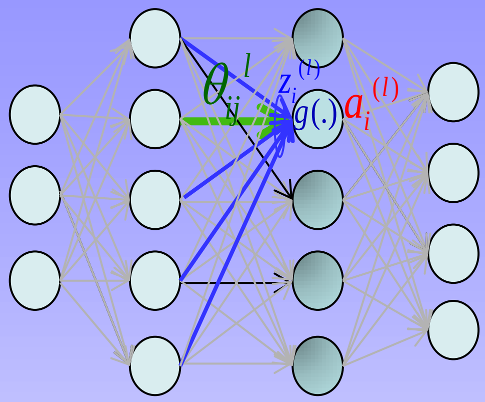
- נסמן: $\theta_i^{(l)}$ - וקטור הפרמטרים של היחידה ה- i –ית בשכבה ה- l (כולל יחידת ה- bias).

זהו וקטור עם $s_{l-1}+1$ רכיבים המוגדר על-ידי:

$$\theta_i^{(l)} = \begin{pmatrix} \theta_{i0}^{(l)} \\ \theta_{i1}^{(l)} \\ \theta_{i2}^{(l)} \\ \vdots \\ \theta_{is_{l-1}}^{(l)} \end{pmatrix}$$



חישוב הגרדיאנט: אלגוריתם ה- Backpropagation



- צעד האיטרציה יהיה מהצורה:

$$\theta_i^{(l)} = \theta_i^{(l)} + \Delta\theta_i^{(l)}$$

new *old*

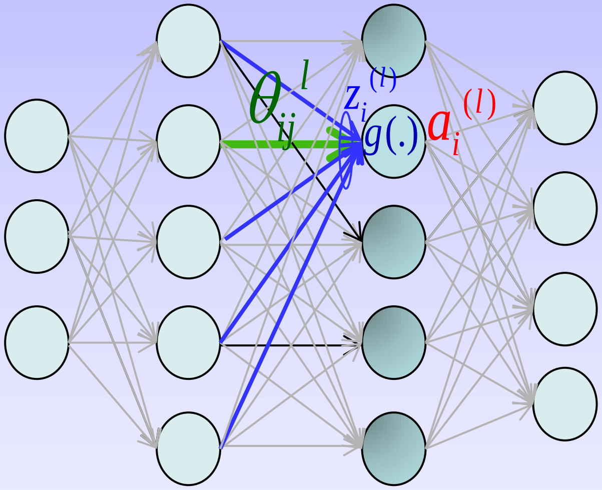
- כאשר:

$$\Delta\theta_i^{(l)} = -\alpha \cdot \frac{\partial J(\theta)}{\partial \theta_i^{(l)}}$$

- כאשר $\Delta\theta_i^{(l)}$ זהו התיקון עבור $\theta_i^{(l)}$ (שניהם וקטורים).

חישוב הגרדיאנט: אלגוריתם ה- Backpropagation

- הנחה: בכל השכבות פונקציית האקטיבציה היא סיגמואידית.
- נסמן $z_i^{(l)}$ - סכום הקלטים המשוקללים ליחידה ה- i בשכבה ה- l.
- נסמן $a_i^{(l)}$ - היציאה של היחידה ה- i בשכבה ה- l לאחר הפעלת פונקציית האקטיבציה.
- בהמשך נתרכז בפונקציות מחיר מהצורה



$$J = \sum_{k=1}^{m} \varepsilon\left(k\right)$$

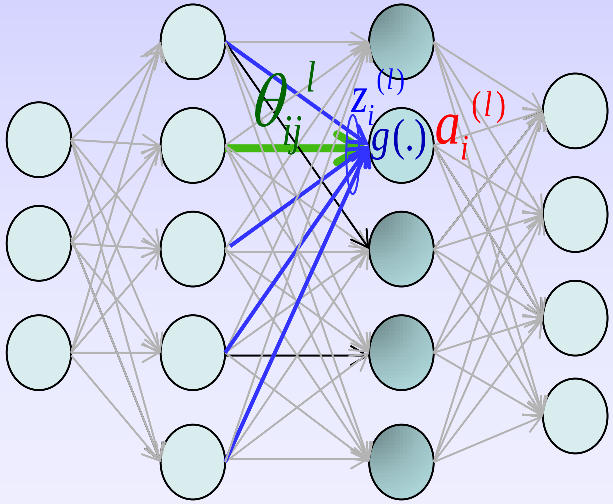
- כאשר ε היא פונקציה המוגדרת בהתאם,

התלוייה בפלט ובתיוג המתאים של דוגמאות האימון $\left(x^{(k)}, y^{(k)}\right)$

חישוב הגרדיאנט: אלגוריתם ה- Backpropagation

- לדוגמא, אפשר לבחור פונקציית ε כסכום ריבועי השגיאות ביחידות הפלט:

$$\varepsilon(k)=\frac{1}{2}\sum_{r=1}^{S_L} e_r(k)=\frac{1}{2}\sum_{r=1}^{S_L}\left(h_\theta(x)_r-y_r(k)\right)^2 \quad k=1,2,\ldots,m$$

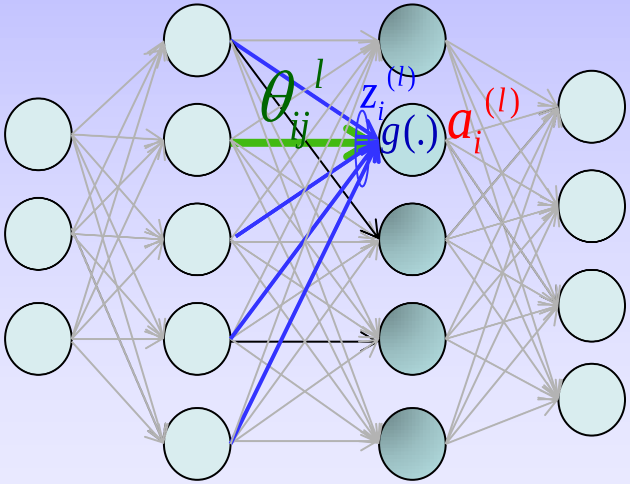


חישוב הגרדיאנטים

- הנחה: (מטעמי נוחות חישובית): קיימת דוגמת אימון אחת (x, y)
- יהי $a_j^{(l-1)}$ היציאה של היחידה ה-j בשכבה ה- *l-1* , כאשר $j=1,2,\ldots,s_{l-1}$
- נסמן: $\theta_{ij}{}^{l}$ - הפרמטר או המשקל הסינפטי בין היחידה ה- j –ית בשכבה ה- *l-1* ליחידה ה- i –ית בשכבה ה- l, כאשר $i=1,2,\ldots,s_l$
- לפיכך, הקלט עבור פונקציית האקטיבציה של היחידה הנ"ל הוא:

$$z_i^{(l)} = \sum_{j=1}^{s_{l-1}} \theta_{ij}{}^{(l)} a_j{}^{(l-1)} + \theta_{i0}{}^{(l)} =$$

$$\sum_{j=0}^{s_{l-1}} \theta_{ij}{}^{(l)} a_j{}^{(l-1)}$$



כאשר לפי ההגדרה $a_0^{(l-1)}=1$, כלומר לכל l כך שה- bias נכלל במשקלות.