Subject:

Year Month Date ( )

سؤال ۲ . الف)

$$\frac{\partial \hat{y}}{\partial z_i} = \begin{bmatrix} \frac{\partial \hat{y}_1}{\partial z_i} \\ \vdots \\ \frac{\partial \hat{y}_i}{\partial z_i} \\ \vdots \\ \frac{\partial \hat{y}_c}{\partial z_i} \end{bmatrix} \quad 1) \Rightarrow \frac{\partial \hat{y}_i}{\partial z_i} = \frac{\partial}{\partial z_i}\left(\frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} \right)$$

$$= \frac{e^{z_i}\left(\sum_{j=1}^{c} e^{z_j}\right) - e^{z_i} \times e^{z_i}}{\left(\sum_{j=1}^{c} e^{z_j}\right)^2}$$

$$= \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} \times \frac{\left(\sum_{j=1}^{c} e^{z_j}\right) - e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}$$

$$= \operatorname{softmax}(z_i) \times \left(1 - \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}\right)$$

$$\Rightarrow \frac{\partial \hat{y}_i}{\partial z_i} = \operatorname{softmax}(z_i) \times (1 - \operatorname{softmax}(z_i))$$

$$2) \frac{\partial \hat{y}_j}{\partial z_i},\ j \neq i \Rightarrow = \frac{0 - e^{z_i} e^{z_j}}{\left(\sum_{j=1}^{c} e^{z_j}\right)^2} = -1 \times \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} \times \frac{e^{z_j}}{\sum e^{z_j}}$$

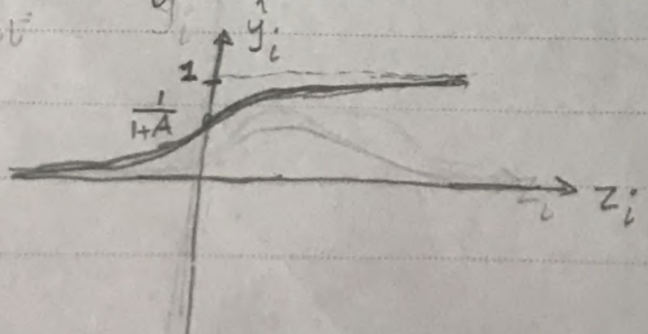
$$\Rightarrow \frac{\partial \hat{y}_j}{\partial z_i} = -\operatorname{softmax}(z_i) \times \operatorname{softmax}(z_j), \quad j \neq i$$

رسم نمودار: یک نمودار به ازای $\hat{y}_i - z_i$ و یک نمودار به ازای $\hat{y}_j - z_i$ ($j \neq i$) رسم می‌کنیم. z_j ($j \neq i$) را ثابت فرض می‌کنیم. داریم:

$$\operatorname{softmax}(z_i) = \frac{e^{z_i}}{\sum e^{z_j}}$$

$$\Rightarrow \hat{y}_i = \frac{e^{z_i}}{e^{z_i} + \sum_{j \neq i} e^{z_j}} = \frac{e^{z_i}}{e^{z_i} + A} \rightarrow A \text{ ثابت}$$

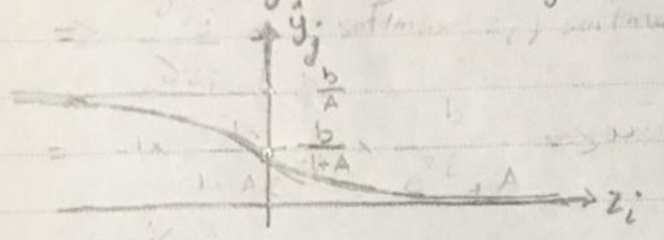
A: ثابت



Vazik

Subject:
Year Month Date ( )

$$\hat{y}_j = \frac{e^{z_j}}{\sum_j e^{z_j}} = \frac{e^{z_j}}{e^{z_i} + \sum_{j\neq i} e^{z_j}} = \frac{b}{e^{z_i} + A}$$



ب) تابع sigmoid به فرم $\frac{1}{1+e^{-x}}$ است. با توجه به آنکه گفته شده است y

one hot است، پس مسئله classification است. در این مسئله می‌خواهیم خروجی

به صورت توزیع احتمال مشخص شده باشد (تعلق به دسته‌ها)، تابع softmax تنها در صورتی که دو کلاس

جواب

داشته باشیم می‌تواند یک توزیع احتمال بسازد اما Cross Entropy به ازای هر تعداد کلاس

یک توزیع احتمال برای احتمال تعلق به هر دسته می‌سازد که مزیت بزرگی نسبت به sigmoid

محسوب می‌شود. یک موضوع دیگر که از دو نمودار قسمت الف نتیجه می‌شود این است که cross entropy،

بسیار بهتر می‌تواند خروجی one-hot در صورتی که ایجاب می‌کند، ایجاد کند. در دو نمودار الف می‌بینیم

که هر چقدر z_i بیشتر باشد، $\hat{y}_i$ متناظر به 1 میل می‌کند و $\hat{y}_j$ به صفر میل می‌کند یعنی یک

logit می‌تواند تاثیر خود را روی تمام دسته‌ها بگذارد که ویژگی مطلوبی است. اما در sigmoid

هر خروجی نهایی تنها به logit متناظر خود وابسته است، و بالا بردن logit یک دسته

Vazik

تاثیری در صفر شدن احتمال دسته دیگر نمی‌شود

Subject:
Year Month Date ( )

سوال ۲. ب)

$$\Rightarrow \frac{e^{(z_i+\varepsilon)}}{\sum_{j=1}^{c} e^{(z_j+\varepsilon)}} = \frac{e^{\varepsilon} e^{z_i}}{\sum e^{z_j} . e^{\varepsilon}}$$

$$= \frac{e^{\varepsilon} e^{z_i}}{e^{\varepsilon} \sum e^{z_j}} = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}$$

ت) اگر $\varepsilon = \max_{j=1,\dots,c} z_j$ ، باعث می‌شود که logitها یا همان z_jها کوچکتر از صفر شوند و در نتیجه: $z_j \leqslant 0 \rightarrow e^{z_j} \leqslant 1$

که باعث می‌شود مقادیر محاسبه شده در تابع نمایی محدود شود و overflow رخ ندهد

ث) تابع likelihood : $P(y^P, x \mid \Theta) = P(y \mid x, \Theta)\, \underbrace{P(x)}$

چون تابعی از پارامترها نیست، ثابت است و در حل ~~مسئله~~ بهینه‌سازی تاثیری ندارد

$$\Rightarrow P(y \mid x, \Theta) = \prod_{i=1}^{c} \hat{y}_i^{\,y_i} \rightarrow$$ که c تعداد کلاس‌ها است و y_i بردار به صورت one-hot درآمده y است

$$\Rightarrow \text{log likelihood}: \log\left(\prod_{i=1}^{c} \hat{y}_i^{\,y_i}\right) = \sum_{i=1}^{c} y_i \log(\hat{y}_i)$$

تابع Cross-Entropy ، منفی log likelihood است پس مینیمم کردن آن معادل ماکسیمم کردن log likelihood است

Vazik

سوال دو ج)

$$L_{CE}(y,\hat{y}) = -\sum_{i=1}^{c} y_i \log(\hat{y}_i)$$

$$= -\sum_{i=1}^{c} y_i \log \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} = -\sum_{i=1}^{c} y_i \left(\log e^{z_i} - \log \sum e^{z_j}\right) = -\sum_{i=1}^{c} y_i \left(z_i - \log \sum e^{z_j}\right)$$

$$L_{MSE}(y,\hat{y}) = \frac{1}{n} \sum_{i=1}^{c} (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^{c} \left(y_i - \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}\right)^2$$

ج)

$$L_{CE}(y,\hat{y}) = -\sum_{i=1}^{c} y_i \log \frac{e^{z_i}}{\sum e^{z_j}} = -\sum_{i=1}^{c} y_i \left(z_i - \log \sum_j e^{z_j}\right)$$

$$= -z_y + \log \sum_j e^{z_j}$$

y نشان دهنده شماره برچسب درست است چرا که به ازای بقیه i صفر می‌شود

(چون y_i متناظر باید یک باشد چون $i=y$)

$$\Rightarrow \frac{\partial L}{\partial z} = \begin{bmatrix} \frac{\partial L}{\partial z_1} \\ \vdots \\ \frac{\partial L}{\partial z_i} \\ \vdots \\ \frac{\partial L}{\partial z_c} \end{bmatrix}^T$$

$$\Rightarrow \frac{\partial L}{\partial z_i},\ (i=y) \Rightarrow \frac{\partial L}{\partial z_i} = -1 + \frac{e^{z_i}}{\sum_j e^{z_j}} = -1 + \hat{y}_i = -y_i + \hat{y}_i$$

$$\Rightarrow \text{"},\ (i \neq y) \Rightarrow \frac{\partial L}{\partial z_i} = 0 + \frac{e^{z_j}}{\sum_j e^{z_j}} = 0 + \hat{y}_i = -y_i + \hat{y}_i$$

y_i متناظر صفر است چرا که $i \neq y$

$\Leftarrow$ با توجه به آنکه در هر دو حالت به فرم $\hat{y}_i - y_i$ در آمده، به صورت برداری داریم

$$\frac{\partial L}{\partial z} = \begin{bmatrix} \frac{\partial L}{\partial z_1} \\ \vdots \\ \frac{\partial L}{\partial z_i} \\ \vdots \end{bmatrix}^T = \begin{bmatrix} \hat{y}_1 - y_1 \\ \vdots \\ \hat{y}_i - y_i \\ \vdots \end{bmatrix}^T = \Bigg(\underbrace{\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_i \\ \vdots \end{bmatrix}}_{\hat{y}} - \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \end{bmatrix}}_{y} \Bigg)^T = (\hat{y} - y)^T$$

پس می‌توان نوشت

Subject:

Year Month Date ( )

$$L_{MSE}(y,\hat{y}) = \frac{1}{n}\sum_{i}\left(y_i - \frac{e^{z_i}}{\sum_j e^{z_j}}\right)^2$$

کلاس n

$$= \frac{1}{n}\left[\left(y_1 - \frac{e^{z_1}}{\sum e^{z_j}}\right)^2 + \cdots + \left(y_i - \frac{e^{z_i}}{\sum e^{z_j}}\right)^2 + \cdots + \left(y_n - \frac{e^{z_n}}{\sum e^{z_j}}\right)^2\right]$$

$$\Rightarrow \frac{\partial L}{\partial z_i} = \frac{\sum\left(2\left(y_m - \frac{e^{z_m}}{\sum e^{z_j}}\right) e^{z_m} e^{z_i}\right) - e^{z_i}\sum e^{z_j}}{\left(\sum e^{z_j}\right)^2}$$

$$= \sum_m\left(2(y_m - \hat{y}_m)\cdot\frac{e^{z_m}}{\sum e^{z_j}}\cdot\frac{e^{z_i}}{\sum e^{z_j}}\right) - \frac{e^{z_i}}{\sum e^{z_j}}$$

$$= \sum_m\left(2(y_m - \hat{y}_m)\,\hat{y}_m \cdot \hat{y}_i\right) - \hat{y}_i$$

$$= \hat{y}_i \times 2\sum_m\left((y_m - \hat{y}_m)\,\hat{y}_m\right) - \hat{y}_i$$

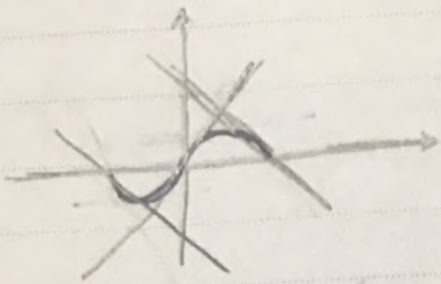
$$= 2\hat{y}_i\,(y-\hat{y})^T\hat{y} - \hat{y}_i = \hat{y}_i\left(2(y-\hat{y})^T\hat{y} - 1\right)$$

$$\Rightarrow \frac{\partial L}{\partial z} = \begin{bmatrix} \hat{y}_1\left(2(y-\hat{y})^T\hat{y}-1\right) \\ \vdots \\ \hat{y}_i\left(2(y-\hat{y})^T\hat{y}-1\right) \end{bmatrix}$$

$$= \left(2(y-\hat{y})^T\hat{y}-1\right)\underbrace{\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_i \\ \vdots \end{bmatrix}}_{\hat{y}} = \left(2(y-\hat{y})^T\hat{y}-1\right)\hat{y}$$

... نورون آخر است تا گرادیان را از طریق آن

به عقب انتشار دهیم.

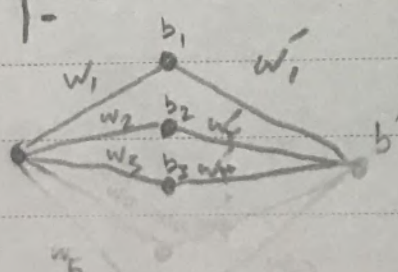


سوال ۳. روش هایی وجود دارند که با استفاده از تابع sigmoid، تابع مورد نظر را با قطعات متعدد خطوط به شکل step تخمین می‌زنند، اما با توجه به آنکه در صورت سوال کمترین تعداد خط خواسته شده است، یک مدل مناسب می‌توان تقریب با ۳ خط بالا باشد. به طور کلی عبارت ریاضی این تابع عبارت است:

$$f(x) = \begin{cases} a_1 x + b_1 & -1 \le x \le m \\ a_2 x + b_2 & m \le x \le n \\ a_3 x + b_3 & n \le x \le k \end{cases}$$

برای آنکه خط‌ها را پیدا کنیم می‌بایست خطای MSE را میان خروجی مدل که در بازه‌های مختلف یک خط است را نسبت به تابع sin کمینه کنیم.

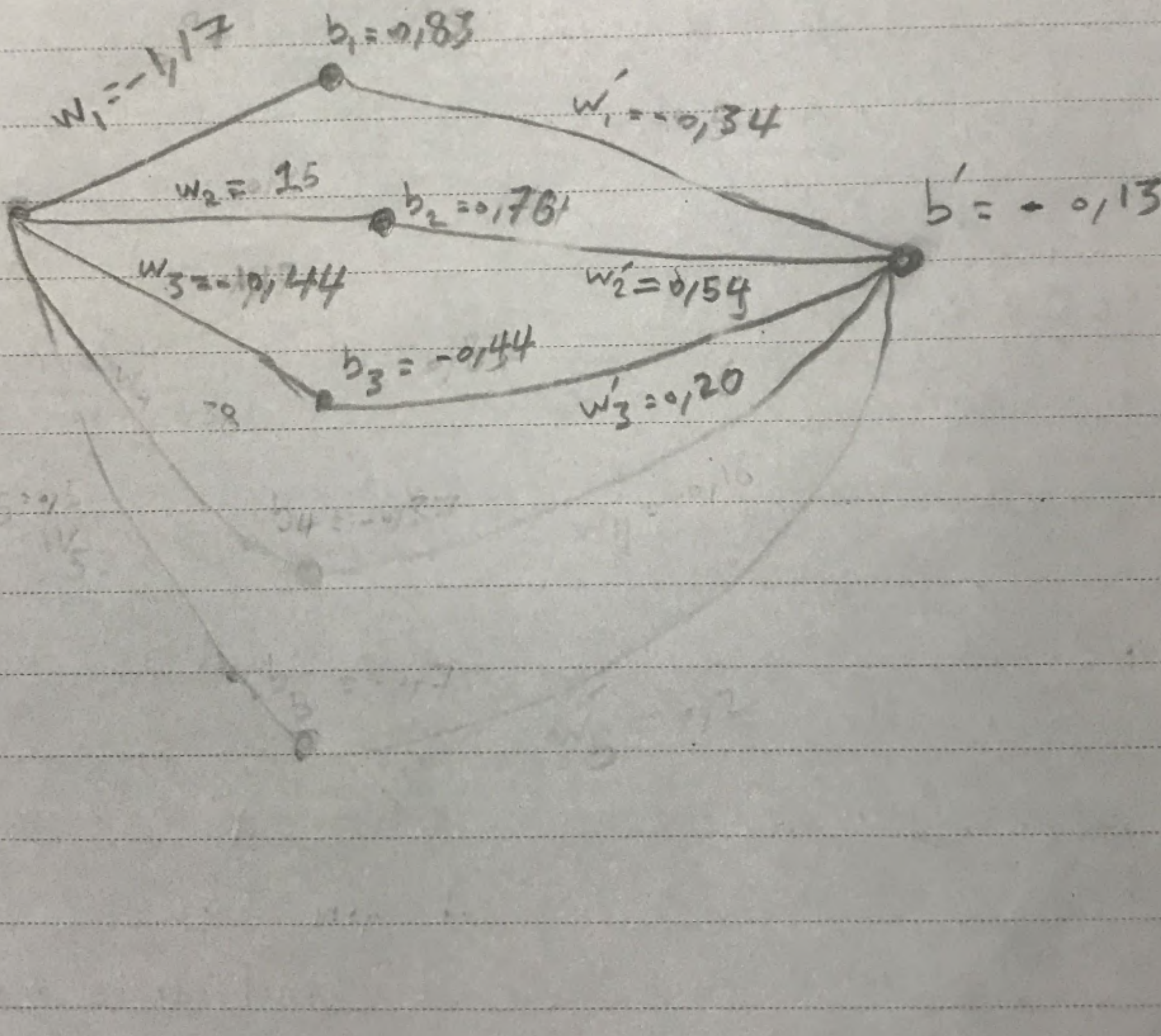
برای ساختن مدل از ۳ نورون مخفی با تابع فعالساز relu استفاده می‌کنیم. حال وزن‌ها باید به گونه‌ای باشد که در ابتدا در بازه $-1 \le x \le m$ نورون 1 روشن باشد (فعال). یعنی $w_1 x + b_1 \ge 0$



حال در بازه بعدی یعنی $m \le x \le n$ ، نورون یک که قاعدتا باید روشن باشد، وزن ها را به صورتی می‌چینیم که نورون دو هم روشن باشد و باقی خاموش. و به همین ترتیب.

(پس توجه می‌کنیم که $a_j x + b_j$ در $f(x)$ معادل $w_j x + b_j$ نیست)

برای آنکه این شرایط فراهم باشد مدل با پارامترهای زیر پیشنهاد می‌شود (با استفاده از کد یادگرفته شد)



Subject:
Year Month Date ( )

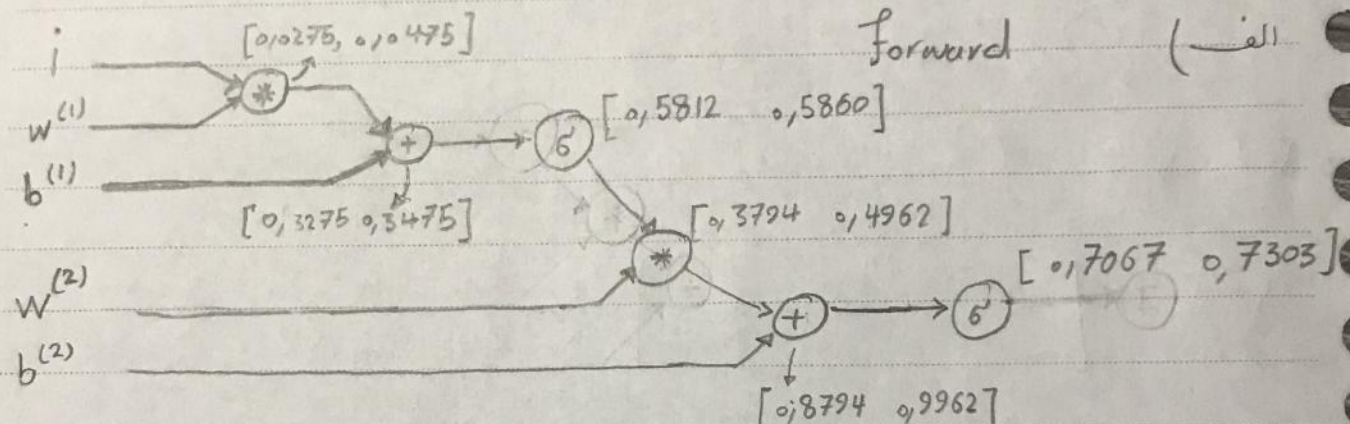
مسئله‌ی ۴: ماتریس وزن‌های میان لایه مخفی و خروجی : $W^{(2)}$, ماتریس وزن‌های میان ورودی و لایه مخفی : $W^{(1)}$

$W^{(1)} = \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} = \begin{bmatrix} 0,1 & 0,2 \\ 0,15 & 0,25 \end{bmatrix}$, $W^{(2)} = \begin{bmatrix} w_5 & w_7 \\ w_6 & w_8 \end{bmatrix} = \begin{bmatrix} 0,3 & 0,4 \\ 0,35 & 0,45 \end{bmatrix}$

$i = [i_1 \;\; i_2] = [0,05 ,\; 0,15]$, $b^{(1)} = [0,3 \;\; 0,3]$, $b^{(2)} = [0,5 ,\; 0,5]$

$z^{(L)}$: مقدار preactivation در لایه L $\quad$ $a^{(L)}$: اکتیویشن در لایه L

الف) Forward



خروجی شبکه : $\sigma(\sigma(i \times w^{(1)} + b^{(1)}) \times w^{(2)} + b^{(2)})$

(روی شکل مرحله به مرحله نشان داده شده است)

به صورت درایه به درایه عمل می‌کند

محاسبات: ۱) $i \times w^{(1)} = [0,05 \;\; 0,15] \begin{bmatrix} 0,1 & 0,2 \\ 0,15 & 0,25 \end{bmatrix} = [0,0275 \;\; 0,0475]$

۲) $z^{(1)} = i \times w^{(1)} + b^{(1)} = [0,0275 ,\; 0,0475] + [0,3 \;\; 0,3] = [0,3275 \;\; 0,3475]$

۳) $a^{(1)} = \sigma(i \times w^{(1)} + b^{(1)}) = \left[\frac{1}{1+e^{-0,3275}} \;\; \frac{1}{1+e^{-0,3475}} \right] = [0,5812 \;\; 0,5860]$

element-wise

۴) $a^{(1)} \times w^{(2)} = [0,5812 \;\; 0,5860] \begin{bmatrix} 0,3 & 0,4 \\ 0,35 & 0,45 \end{bmatrix} = [0,3794 \;\; 0,4962]$

5) $z^{(2)} = a^{(1)} + w^{(2)} + b^{(2)} = [0,3794 \;\; 0,4962] + [0,5 \;\; 0,5] = [0,8794 \;\; 0,9962]$

6) $a^{(2)} = \sigma(z^{(2)}) = [\underbrace{0,7067}_{o_1} \;\; \underbrace{0,7303}_{o_2}]$

Subject:
Year Month Date ( )

۴. ب)

$$E = \frac{1}{2}\left[(0,1 - 0,7067)^2 + (0,9 - 0,7303)^2 \right] =$$

$$= \frac{1}{2}(0,3681 + 0,0288) = 0,1984$$

الگوریتم Back propagation :

$$\nabla_o E = \left[\frac{\partial E}{\partial o_1} \quad \frac{\partial E}{\partial o_2} \right] = \nabla_{a^{(2)}} E$$

$$\Rightarrow \frac{\partial E}{\partial o_j} = \frac{\partial}{\partial o_j}\left[\frac{1}{2}\sum_{k=1}^{2} (o_k - t_k)^2 \right] = \frac{1}{2}\frac{\partial (o_j^2 - 2o_j t_j + t_j^2)}{\partial o_j}$$

$$= \frac{1}{2}(2o_j - 2t_j) = o_j - t_j$$

$$\Rightarrow \nabla_{a^{(2)}} E = \nabla_o E = [\, o_1 - t_1 \quad o_2 - t_2 \,] = [\, 0,6067 \quad -0,1697 \,]$$

$$\nabla_{z^{(2)}} E = \left[\frac{\partial E}{\partial z_1^{(2)}} \quad \frac{\partial E}{\partial z_2^{(2)}} \right] = \left[\sum_k \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial z_1^{(2)}} \quad \sum_k \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial z_2^{(2)}} \right]$$

چون sigmoid درایه به درایه بود

$$= \left[\frac{\partial E}{\partial o_1}\frac{\partial o_1}{\partial z_1^{(2)}} \quad \frac{\partial E}{\partial o_2}\frac{\partial o_2}{\partial z_2^{(2)}} \right]$$

$$\Rightarrow \frac{\partial o_j}{\partial z_j^{(2)}} = \frac{\partial}{\partial z_j^{(2)}}\left[\frac{1}{1+e^{-z_j^{(2)}}} \right] \Rightarrow$$

مشتق تابع sigmoid : $y = \frac{1}{1+e^{-x}} \rightarrow y' = \frac{0+e^{-x}}{(1+e^{-x})^2}$

$$\Rightarrow y' = \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} = \sigma'(x)\left(1 - \frac{1}{1+e^{-x}}\right) = \sigma'(x)(1 - \sigma'(x))$$

$$\Rightarrow \frac{\partial o_j}{\partial z_j^{(2)}} = \sigma'(z_j^{(2)})(1 - \sigma'(z_j^{(2)}))$$

$$\Rightarrow \nabla_{z^{(2)}} E = \left[0,6097 \times \sigma'(0,8794) \times 1 - \sigma'(0,8794) \,,\; -0,1697 \times \sigma'(0,9962) \times 1 - \sigma'(0,9962) \right]$$

$$= [\, 0,1263 \quad -0,0334 \,] \xRightarrow{\text{chain rule}} \nabla_{b^{(2)}} E = \nabla_{z^{(2)}} E \; \underbrace{\nabla_{b^{(2)}} z^{(2)}}_{I} = \nabla_{z^{(2)}} E$$

$$\Rightarrow \nabla_{b^{(2)}} E = [\, 0,1263 \quad -0,0334 \,]$$

Subject:
Year Month Date ( )

$$\nabla_{W^{(2)}} E = \begin{bmatrix} \frac{\partial E}{\partial w_5} & \frac{\partial E}{w_7} \\ \frac{\partial E}{\partial w_6} & \frac{\partial E}{\partial w_8} \end{bmatrix} \overset{\text{chain}}{\underset{\text{rule}}{=}} \begin{bmatrix} \nabla_{z^{(2)}} E \, \nabla_{w_5} z^{(2)} & \nabla_{z^{(2)}} E \, \nabla_{w_7} z^{(2)} \\ \nabla_{z^{(2)}} E \, \nabla_{w_6} z^{(2)} & \nabla_{z^{(2)}} E \, \nabla_{w_8} z^{(2)} \end{bmatrix}$$

$$\nabla_{w_5} z^{(2)} = \begin{bmatrix} \frac{\partial z_1^{(2)}}{\partial w_5} \\ \frac{\partial z_2^{(2)}}{\partial w_5} \end{bmatrix} = \begin{bmatrix} a_1^{(1)} \\ 0 \end{bmatrix}, \quad \nabla_{w_6} z^{(2)} = \begin{bmatrix} a_2^{(1)} \\ 0 \end{bmatrix}$$

$$\nabla_{w_7} z^{(2)} = \begin{bmatrix} 0 \\ a_1^{(1)} \end{bmatrix}, \quad \nabla_{w_8} z^{(2)} = \begin{bmatrix} 0 \\ a_2^{(1)} \end{bmatrix}$$

$$\Rightarrow \nabla_{W^{(2)}} E = \begin{bmatrix} \frac{\partial E}{\partial z_1^{(2)}} a_1^{(1)} & \frac{\partial E}{\partial z_2^{(2)}} a_1^{(1)} \\ \frac{\partial E}{\partial z_1^{(2)}} a_2^{(1)} & \frac{\partial E}{\partial z_2^{(2)}} a_2^{(2)} \end{bmatrix} = \begin{bmatrix} a_1^{(1)} \\ a_2^{(1)} \end{bmatrix} \begin{bmatrix} \frac{\partial E}{\partial z_1^{(2)}} & \frac{\partial E}{\partial z_2^{(2)}} \end{bmatrix}$$

$$\Rightarrow \nabla_{W^{(2)}} E = (a^{(1)})^T \times \nabla_{z^{(2)}} E = \begin{bmatrix} 0{,}5812 \\ 0{,}5860 \end{bmatrix} \begin{bmatrix} 0{,}1263 & -0{,}0334 \end{bmatrix}$$

$$= \begin{bmatrix} 0{,}0734 & -0{,}0194 \\ 0{,}0740 & -0{,}0195 \end{bmatrix}$$

برای $\nabla_{a^{(1)}} E$ به طور مشابه با بالا می‌توان نشان داد: $\nabla_{a^{(1)}} = \nabla_{z^{(2)}} E \times (W^{(2)})^T$

$$\Rightarrow \nabla_{a^{(1)}} E = \begin{bmatrix} 0{,}1265 & -0{,}0334 \end{bmatrix} \begin{bmatrix} 0{,}3 & 0{,}35 \\ 0{,}4 & 0{,}45 \end{bmatrix} = \begin{bmatrix} 0{,}0245 & 0{,}0291 \end{bmatrix}$$

$\nabla_{z^{(1)}} E \Longrightarrow$ sigmoid نشان داده شد $\Rightarrow \nabla_{z^{(1)}} E = [\, 0{,}0245 \times \sigma(0{,}3275) \times 1 - \sigma(0{,}3275)$

$0{,}0291 \times \sigma(0{,}3475) \times (1 - \sigma(0{,}3475) \,]$

Vazik

Subject:

Year Month Date ( )

$$\Rightarrow \nabla_{z^{(1)}} E = [0.0064 \quad 0.0086] \xRightarrow[\text{نشان دادیم}]{\text{همانطور که}} \nabla_{b^{(1)}} E = [0.0059 \quad 0.0070]$$

$$\nabla_{w^{(1)}} E \xlongequal[\text{برای } w^{(2)} \text{ نشان دادیم}]{\text{همانطور که}} i^T \nabla_{z^{(1)}} E = \begin{bmatrix} 0.05 \\ 0.15 \end{bmatrix} [0.0059 \quad 0.0070]$$

$$\Rightarrow \nabla_{w^{(1)}} E = \begin{bmatrix} 0.0003 & 0.0004 \\ 0.0009 & 0.0011 \end{bmatrix}$$

$$\text{Update: } w^{(1)} = w^{(1)} - 0.4 \frac{\partial E}{\partial w^{(1)}} = \begin{bmatrix} 0.1 & 0.2 \\ 0.15 & 0.25 \end{bmatrix} - 0.4 \begin{bmatrix} 0.0003 & 0.0004 \\ 0.0009 & 0.0011 \end{bmatrix}$$

$$\Rightarrow w^{(1)} = \begin{bmatrix} 0.0999 & 0.1999 \\ 0.1496 & 0.2496 \end{bmatrix}$$

$$b^{(1)} = [0.3 \quad 0.3] - 0.4 [0.0059 \quad 0.0070] = [0.2976 \quad 0.2972]$$

$$w^{(2)} = \begin{bmatrix} 0.3 & 0.4 \\ 0.35 & 0.45 \end{bmatrix} - 0.4 \begin{bmatrix} 0.0734 & -0.0194 \\ 0.0740 & -0.0195 \end{bmatrix}$$

$$= \begin{bmatrix} 0.2706 & 0.4078 \\ 0.3204 & 0.4578 \end{bmatrix}$$

$$b^{(2)} = [0.5 \quad 0.5] - 0.4 [0.1263 \quad -0.0334] = [0.4495 \quad 0.5134]$$

مسئله ۵ : الف)

$\frac{\partial E}{\partial w(f', c', h', w')} \Rightarrow$ از قاعده chain rule استفاده می‌کنیم و مشتق را نسبت به خروجی عملیات convolution باز می‌کنیم

$$= \sum_{a=0}^{N-1} \sum_{b=0}^{F-1} \sum_{i=0}^{o_h-1} \sum_{j=0}^{o_w-1} \frac{\partial E}{\partial o(a,b,i,j)} \cdot \frac{\partial o(a,b,i,j)}{\partial w(f', c', h', w')}$$

حال طبق صورت سوال می‌توان $o(a,b,i,j)$ را به صورت زیر نوشت:

Subject:
Year Month Date ( )

$$\Rightarrow O(a,b,i,j) = \sum_{r=0}^{C-1}\sum_{k=0}^{f_h-1}\sum_{f=0}^{f_w-1} W(b,r,k,f)\,X(a,r,s\times i+k,s\times j+f)$$

با توجه به آنکه مشتق نسبت به پارامتر $W(f',c',h',w')$ گرفته می‌شود،

تنها زمانی که داریم: $\text{I}\begin{cases} b=f' \\ r=c' \\ k=h' \\ f=w' \end{cases}$ مشتق غیرصفر است و مقدار مشتق آن

$$\frac{\partial O(a,b,i,j)}{\partial W(f',c',h',w')}$$

برابر است با: II

$$= X(a,r,s\times i+k,s\times j+f) \overset{\text{I}}{=} X(a,c',s\times i+h',s\times j+w')$$

و در باقی موارد صفر است. در نتیجه در نهایت داریم:

$$\frac{\partial E}{\partial W(f',c',h',w')} = \sum_{a=0}^{N-1}\sum_{b=0}^{F-1}\sum_{i=0}^{O_h-1}\sum_{j=0}^{O_w-1}\frac{\partial E}{\partial O(a,b,i,j)}\cdot\frac{\partial O(a,b,i,j)}{\partial W(f',c',h',w')}$$

با توجه به آنکه در I پیدا کردیم که $b=f'$، در نتیجه جمع روی b برداشته می‌شود
چرا که در $b\neq f'$ مشتق صفر است.

$$\Rightarrow = \sum_{a=0}^{N-1}\sum_{i=0}^{O_h-1}\sum_{j=0}^{O_w-1}\frac{\partial E}{\partial O(a,f',i,j)}\cdot\frac{\partial O(a,f',i,j)}{\partial W(f',c',h',w')}$$

$$\overset{\text{II}}{\Longrightarrow} = \sum_{a=0}^{N-1}\sum_{i=0}^{O_h-1}\sum_{j=0}^{O_w-1}\frac{\partial E}{\partial O(a,f',i,j)}\cdot X(a,c',s\times i+h',s\times j+w')$$

ب) همانند بخش الف داریم:

$$\frac{\partial E}{\partial X(n',c',h',w')} = \sum_{a=0}^{N-1}\sum_{b=0}^{F-1}\sum_{i=0}^{O_h-1}\sum_{j=0}^{O_w-1}\frac{\partial E}{\partial O(a,b,i,j)}\cdot\frac{\partial O(a,b,i,j)}{\partial X(n',c',h',w')}$$

$$O(a,b,i,j) = \sum_{r=0}^{C-1}\sum_{k=0}^{f_h-1}\sum_{f=0}^{f_w-1} W(b,r,k,f)\,X(a,r,s\times i+k,s\times j+f)$$

طبق استدلال مشابه با الف، چون مشتق نسبت به $X(n',c',h',w')$ است داریم:

$$\begin{cases} a = n' \\ r = c' \\ s\times i + k = h' \rightarrow k = h' - s\times i \\ s\times j + l = w' \rightarrow l = w' - s\times j \end{cases}$$

$$\Rightarrow \frac{\partial O(a,b,i,j)}{\partial X(n',c',h',w')} = W(b,c',h'-s\times i, w'-s\times j)$$

و چون $a=n'$، مجموع روی a از مشتق برداشته می‌شود چون در حالت‌های $a\neq n'$ مشتق صفر است:

$$\frac{\partial E}{\partial X(n',c',h',w')} = \sum_{b=0}^{F-1}\sum_{i=0}^{O_h-1}\sum_{j=0}^{O_w-1} \frac{\partial E}{\partial O(n',b,i,j)} \cdot \frac{\partial O(a,b,i,j)}{\partial X(n',c',h',w')}$$

$$= \sum_{b=0}^{F-1}\sum_{i=0}^{O_h-1}\sum_{j=0}^{O_w-1} \frac{\partial E}{\partial O(n',b,i,j)} \cdot W(b,c',h'-s\times i, w'-s\times j)$$

یک مورد دیگر هم در بخش relu اشاره خواهد شد

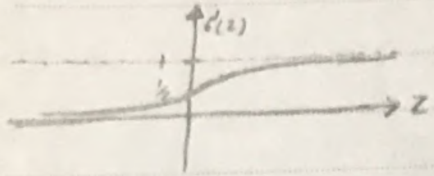
Subject:
Year Month Date ( )

مسئله‌ی ۶. الف) ۱- تابع sigmoid : از لحاظ مقدار تابع: $\sigma(z) = \frac{1}{1+e^{-z}}$

عیب:

$0 < \sigma(z) < 1$

مقدار تابع sigmoid مثبت است و zero-centered نیست که می‌تواند منجر به مشکل زیر شود:



* مشکل zero-centered نبودن تابع activation : ورودی لایه مخفی k ام $a^{(k-1)}$ یا همان a را در نظر می‌گیریم که مقادیر آن همگی مثبت هستند (به عنوان مثال اگر تابع activation در لایه k-1 ام، تابع sigmoid باشد این اتفاق قطعاً رخ می‌دهد). حال نورون دلخواه j را در لایه k در نظر می‌گیریم. در این نورون داریم:

$$z_j^{(k)} = \sum_{i=1}^{N^{(k-1)}} W_{ij}^{(k)} a_i^{(k-1)} + b^{(k)}$$

که $N^{(k-1)}$ تعداد نورون‌ها در لایه مخفی (k-1) را نشان می‌دهد. و سپس تابع activation اعمال می‌شود که فرض می‌کنیم تابع sigmoid است. در Backpropagation داریم:

$$\frac{dL}{dw_{ij}^{(k)}} = \frac{dL}{dz_j^{(k)}} \cdot \frac{dz_j^{(k)}}{dw_{ij}^{(k)}}$$

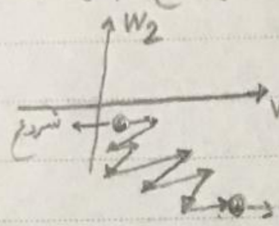
$$\Longrightarrow \frac{dz_j^{(k)}}{dw_{ij}^{(k)}} = a_i^{(k-1)} > 0 \longrightarrow$$ چون از سیگموید لایه قبل آمده است

← در نتیجه علامت گرادیان $W_{ij}^{(k)}$ به ازای هر i، هم علامت با $\frac{dL}{dz_j^{(k)}}$ است که این موضوع می‌تواند در بهینه‌سازی و رسیدن به نقطه optimal مشکل و محدودیت ایجاد کند

Subject:
Year Month Date ( )

به عنوان مثال فرض می‌کنیم تنها دو وزن ورودی وجود داشته باشد. با توجه به آنکه گرادیان هر دو هم علامت است، پس بردار update تنها می‌تواند به دو شکل کلی ↗ یا ↙ باشد، یعنی یا هر دو زیاد شوند یا هر دو کم. واضح است که این نوع آپدیت می‌تواند محدودیت در رسیدن به نقطه بهینه ایجاد کند چرا که فرض کنید داشته باشیم

یعنی مسیر رسیدن

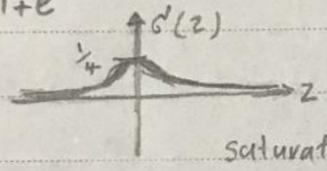


به نقطه مورد نظر زیگزاگی می‌شود و نمی‌توان به صورت مستقیم به آن دست یافت

⟸ در نتیجه تابع sigmoid چون مقادیر مثبت (غیر zero-centered) دارد منجر به این شکل می‌شود

الف) ۱- تابع sigmoid : از لحاظ مشتق: عیب:

$$\sigma(z) = \frac{1}{1+e^{-z}} \rightarrow \sigma'(z) = \sigma(z)\,(1-\sigma(z))$$

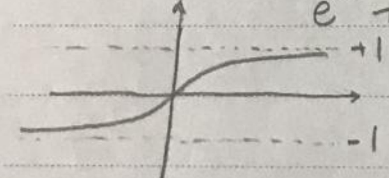


اگر مقدار (اندازه) preactivation یا همان z بزرگ باشد باعث می‌شود گرادیان صفر شود و به اصطلاح اشباع یا saturate شود. با صفر شدن گرادیان، در backprop دیگر گرادیان منتشر شده به لایه‌های ابتدایی صفر می‌شود و هیچ آپدیتی رخ نمی‌دهد. به این مشکل vanishing gradiant می‌گویند

(یک مورد دیگر در بخش relu)

الف) ۲- تابع tanh : از لحاظ مقدار تابع : مزیت:

$$\tanh(z) = \frac{e^{2x}-1}{e^{2x}+1}$$



همانطور که مشخص است مقادیر خروجی این تابع میان ۱+ و ۱- است که یعنی این تابع zero-centered است و از این لحاظ مطلوب است

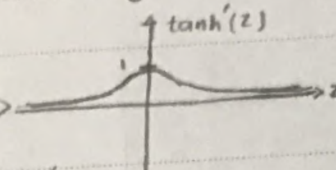
- همچنین به طور کلی می‌توان نشان داد که تابع tanh را می‌توان با تابع سیگموید ساخت و بالعکس

$$\tanh(z) = \frac{e^{2z}-1}{e^{2z}+1} = 1 - \frac{2}{e^{2z}+1} = 1 - 2\sigma(-2z)$$

الف) ۲- تابع tanh: از لحاظ مشتق:

$$\tanh(z) = \frac{e^{2z}-1}{e^{2z}+1}$$

$$\rightarrow \tanh'(z) = 1 - \tanh^2(z) \Rightarrow$$



عیب: مشتق این تابع نیز به ازای مقادیر بزرگ z (preactivation) صفر می‌شود
که منجر به صفر شدن انتشار گرادیان یعنی vanishing gradient می‌شود

مزیت نسبت به sigmoid: مشتق در tanh نسبت به sigmoid دیرتر صفر می‌شود
یعنی مقادیر z هر چقدر بزرگ‌تر شوند، گرادیان در tanh دیرتر شروع به صفر شدن می‌کند (نسبت به sigmoid) و در نتیجه کمی قدرت عمل بیشتری دارد

← با این تفاسیر می‌توان گفت tanh همواره به sigmoid ترجیح داده می‌شود (البته این حرف کاملا دقیق نیست. از نظر بیولوژیکی تابع sigmoid مدل بهتری برای fire کردن یک نورون است چرا که مقدار پتانسیل خروجی یک نورون در هنگامی که فعال نیست صفر است)

الف) ۳) relu: از لحاظ مقدار تابع: عیب:

$$f(x) = \begin{cases} x & x > 0 \\ 0 & o.w \end{cases}$$

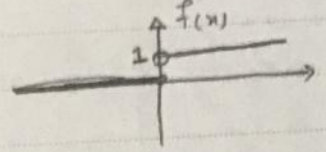
مقدار این تابع می‌تواند به صورت نامحدود بزرگ باشد و با توجه به آنکه مقدار آن در محاسبه گرادیان ماتریس وزن‌های لایه بعد تاثیرگذار است، اگر learning rate مناسب نباشد، می‌تواند باعث exploding gradient شود (مقادیر بسیار زیاد شوند که overflow کنند)

اما در sigmoid و tanh چون مقادیر خروجی آن‌ها محدود است این اتفاق رخ نمی‌دهد

الف) ۳) relu از لحاظ مشتق: مزیت بزرگ:

$$f(x) = \begin{cases} x & x>0 \\ 0 & o.w \end{cases}$$

$$\rightarrow f'(x) = \begin{cases} 1 & x>0 \\ 0 & o.w \end{cases}$$



مزیت بزرگ relu این است که مشتق آن برخلاف tanh و sigmoid باعث vanishing gradient نمی‌شود چرا که مشتق آن در ناحیه فعال همواره یک است و به خوبی باعث انتشار گرادیان به لایه‌های اولیه می‌شود

الف) ۳) relu از لحاظ مشتق: عیب: relu دارای مشکل dying relu است. فرض کنید در یک نورون بردار w و b متناظر با آن آپدیتی دریافت کند که به ازای بردار جدید، مقدار W^Tx+b برای هر x ای کوچکتر از صفر شود (به عنوان مثال اگر مقدار $\nabla_W L$ یا $\nabla_b L$ بزرگ باشد، چون داریم: $W := W - \lambda \nabla_W L$ ، $b := b - \lambda \nabla_b L$ مقادیر جدید b و W می‌تواند بسیار منفی شود که فارغ از اینکه ورودی چیست، preactivation z منفی شود). در صورتی که این اتفاق بیفتد، گرادیان relu صفر می‌شود و در نتیجه هیچ آپدیتی روی W و b رخ نمی‌دهد و در نتیجه W^Tx+b به ازای هر x منفی باقی می‌ماند و تا ابد به همین شکل باقی خواهد ماند

در این حالت به ~~[illegible]~~ اصطلاح می‌گویند که نورون مرده است

Subject:
Year Month Date ( )

دقت می‌کنیم که این مشکل با vanishing gradient متفاوت است

مسئله‌ی ۶ ب) هزینه محاسباتی relu بسیار کمتر از دو تابع دیگر است

چرا که در همان مقدار ورودی یا مقدار صفر را خروجی می‌دهد و در نتیجه سرعت یادگیری را افزایش

می‌دهد. دو تابع tanh و sigmoid به علت محاسبه exp از لحاظ محاسبات

غیربهینه هستند. از بین آن دو هم با توجه به اینکه $\tanh(z) = \frac{e^{2z}-1}{e^{2z}+1}$ و $\sigma(z) = \frac{1}{1+e^{-z}}$

می‌توان گفت هزینه محاسباتی tanh از sigmoid بیشتر است

مسئله‌ی ۶ پ) توضیحات در بخش الف داده شد. tanh و sigmoid مشکل

vanishing gradient دارند اما relu این مشکل را ندارد

Vazik

Subject:
Year Month Date ( )

مسئله‌ی ۷. الف) در مقاله‌ای که dropout در آن معرفی شده است، علت این امر این اشاره شده است که dropout می‌تواند از co-adaptation در شبکه‌های عصبی جلوگیری کند. تعریف co-adaptation: در یک شبکه عصبی، گرادیانی که هر پارامتر دریافت می‌کند باعث می‌شود که تابع هزینه کاهش یابد و در عین حال از وضعیت بقیه نورون‌ها آگاه است در نتیجه یک نورون می‌تواند به نوعی اشتباهات بقیه نورون‌ها را جبران کند. این بدان معنا است که یک سری از نورون‌ها بسیار خوب آموزش می‌بینند و فیچرهای خوبی استخراج می‌کنند و تاثیر بسزایی در خروجی می‌گذارند. اما یک سری دیگر از نورون‌های دیگر، فیچر خاصی استخراج نمی‌کنند (مثلاً همان ورودی لایه را خروجی می‌دهند) چرا که در همکاری با آن نورون‌های غنی در مجموع عملکرد خوبی را روی داده train از خود نشان می‌دهند و به نوعی نورون‌های غنی جور آنها را می‌کشد و به اصطلاح co-adaptation بین آنها رخ داده است.

این اتفاق منجر به overfitting می‌شود چرا که این جنس همکاری‌ها لزوماً قابل تعمیم به داده‌گان تست نیست و در نتیجه چون تعداد قابل توجهی از نورون‌ها فیچر خاصی استخراج نکرده‌اند عملکرد ضعیفی هنگام تست خواهیم داشت.

Vazik

dropout، با حذف کردن تصادفی نورون ها کاری می‌کند که نورون ها نتوانند به حضور نورون دیگری وابسته شود و در نتیجه نمی‌تواند جبران اشتباهات را به نورون دیگری واگذار کند و مجبور می‌شود ویژگی مطلوبی استخراج کند.

در مقاله Understanding Dropout, Baldi et. al یک بررسی تئوری روی یک حالت ساده انجام شده است یک نورون با تابع activation خطی را در نظر می‌گیریم و تابع MSE را به عنوان loss برای آن در نظر می‌گیریم. تابع خطا را برای مدل با dropout و مدل ensemble (همان چیزی که در حالت تست dropout استفاده می‌شود) می‌نویسیم:

$$E_{ENS} = \frac{1}{2}\left(t - \sum_{i=1}^{n} p_i w_i x_i\right)^2$$

$$E_D = \frac{1}{2}\left(t - \sum_{i=1}^{N} \delta_i w_i x_i\right)^2 \quad , \quad \delta_i \sim \text{Bernouli}(p_i)$$

حال گرادیان ها را محاسبه می‌کنیم:

$$\frac{\partial E_{ENS}}{\partial w_i} = -t p_i x_i + w_i p_i^2 x_i^2 + \sum_{j=1, j\neq i}^{n} p_j w_j x_j p_i x_i$$

$$\frac{\partial E_D}{\partial w_i} = -t \delta_i x_i + w_i \delta_i^2 x_i^2 + \sum_{j=1, j\neq i}^{n} \delta_i x_i \delta_j w_j x_j$$

حال امید ریاضی $\frac{\partial E_D}{\partial w_i}$ را حساب می‌کنیم (از آنجایی که dropout یک randomness به مدل اضافه می‌کند پس خطا را می‌توان یک متغیر تصادفی در نظر گرفت که مشتق آن هم متغیر تصادفی است)

Subject:
Date

$$\text{Expected}\left[\frac{\partial E_D}{\partial w_i}\right] \xrightarrow[\text{استقلال } \delta_i, \delta_j]{E[\delta_i]=p_i} -t p_i x_i + w_i p_i^2 x_i^2 + w_i \mathrm{Var}(\delta_i) x_i^2 + \sum_{j=1, j\neq i}^{n} p_i x_i p_j w_j x_j$$

$$= \frac{\partial E_{ENS}}{\partial w_i} + w_i \mathrm{Var}(\delta_i) x_i^2$$

$$= \frac{\partial E_{ENS}}{\partial w_i} + w_i p_i (1-p_i) x_i^2$$

← همانطور که مشخص است گرادیان در حالت dropout، گرادیان نورون بدون dropout به همراه یک regularizer است. پس در واقع تابع هزینه به صورت زیر قابل بازنویسی است:

$$= \underbrace{\frac{1}{2}\left(t - \sum_{i=1}^{N} p_i w_i x_i\right)^2}_{\text{loss}} + \underbrace{\frac{1}{2}\sum_{i=1}^{n} p_i (1-p_i) w_i^2 x_i^2}_{\text{regularization}}$$

۷-ب) در dropout در حالت training، هر یک از نورون‌ها یا حاضرش است یا روشن. اگر M نورون در شبکه داشته باشیم در مجموع می‌توان 2^M شبکه با ساختار مختلف متصور شد. در dropout در هر iteration یک نمونه از این 2^M شبکه گرفته می‌شود و یک بار Backpropagation روی آن زیرشبکه روی یک mini Batch از داده‌ها انجام می‌شود و آن زیرشبکه را می‌توان یک مدل در نظر گرفت. در زمان test از شبکه کامل استفاده می‌کنیم که در واقع شامل تمام زیرشبکه‌هایی است که

Subject:
Date

که در زمان training آموزش دیده‌اند. در نتیجه در زمان test انگار به نوعی
از نتایج تمام آن زیرشبکه‌ها استفاده می‌کنیم. این به نوعی همان کاری است
که در Ensemble Learning رخ می‌دهد که نتایج چندین مدل استفاده می‌شود. البته دقت
می‌کنیم که در dropout وزن‌ها میان انواع زیرشبکه‌ها مشترک است
و در نتیجه مدل‌ها استقلال از هم ندارند که در Ensemble learning بدین صورت نیست

پ) در مرحله train در هر iteration یک سری از نورون‌ها را خاموش می‌کنیم اما در مرحله test
از تمام نورون‌ها برای آنکه از کل ظرفیت استفاده کنیم و به نوعی Ensemble
کنیم استفاده می‌کنیم. در نتیجه از این جنبه عملکرد dropout در مرحله train, test
متفاوت است. اما می‌خواهیم امید ریاضی اکتیویشن نورون‌ها در مراحل train, test
یکسان باشد در نتیجه با ضرب کردن احتمال خاموش کردن نورون‌ها در activation
به صورت تقریبی این کار را انجام می‌دهند و در نتیجه در مراحل train, test
در dropout، امید ریاضی activation ها یکسان است.

اما به طور کلی همانطور که در ابتدا گفتیم dropout در مراحل train, test کارهای
مختلفی انجام می‌دهد و به عنوان مثال در هنگام نوشتن کد یک شبکه
که از dropout استفاده می‌کند باید به این توجه کرد که خاموش نورون را در تست غیرفعال کنیم

Subject:
Date

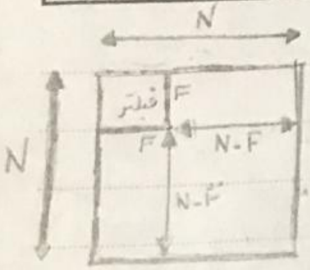
ت) در regularizer هایی مانند dropout ، data augmentation ، ... یک الگوی کلی‌تری دیده می‌شود که آن این است که در مرحله training به نوعی نویز یا randomness به آن اضافه کنیم. چنین کاری باعث می‌شود که مدل به این سمت نرود که داده‌ها را حفظ کند و در برابر تغییرات ورودی به نوعی مقاوم می‌شود و در نتیجه واریانس مدل کاهش یابد. پس وجود نوعی نویز می‌تواند در جلوگیری از overfitting و generalization کمک کند. (در مرحله تست سعی می‌کنیم خروجی را نسبت به randomness، marginalize کنیم)

در Batch Normalization هم این نویز به طور کوچک و جالبی وجود دارد. در این روش میانگین و انحراف از معیار هر minibatch محاسبه می‌شود و روی داده‌ها یا خروجی‌ها تاثیر می‌گذارد. حال با توجه به آنکه minibatch ها به صورت تصادفی انتخاب می‌شوند، اگر یک داده خاص را در نظر بگیریم، بسته به آنکه آن داده در کدام batch قرار می‌گیرد، نحوه normalize شدن آن در Batch Norm تفاوت می‌شود و در نتیجه به نوعی randomness به مدل اضافه شده است. در مرحله تست، با توجه به آنکه میانگین و انحراف از معیار را از کل داده train محاسبه کرده‌ایم، انگار marginalize کرده‌ایم

PAPCO

Subject:
Date

ش) بله. در هنگام Backpropagation نیاز داریم بدانیم که خروجی حاصل از کدام نورون آمده است تا گرادیان را از طریق آن به عقب انتشار دهیم



مسئله‌ی ۸ . الف) هنگامی که فیلتر در اولین جایگاه خود قرار می‌گیرد، می‌بایست N-F خانه را به سمت راست حرکت کند تا ضلع راست آن بر ضلع راست تصویر منطبق شود. حال اگر اندازه گام‌هایی که فیلتر بر می‌دارد را S در نظر بگیریم، فیلتر با $\frac{N-F}{S}$ حرکت می‌تواند به گوشه تصویر برسد. یک مقدار هم برای جایگاه اول فیلتر باید حساب کنیم. پس در نهایت $\frac{N-F}{S}+1$ جایگاه برای فیلتر وجود دارد. با استدلال مشابه حالت عمودی نیز به همین شکل است. پس سایز خروجی $(\frac{N-F}{S}+1)$ و $(\frac{N-F}{S}+1)$ خواهد بود.

ب) grayscale

ورودی: X : (۱, ۲۲۴, ۲۲۴) ⟶ ابعاد

grayscale

Network 1: 1) 16 x conv(5×5) : تعداد پارامترها = $(\overbrace{5\times 5\times 1}^{\text{فیلتر}} + \overbrace{1}^{bias}) \times \overbrace{16}^{\text{تعداد فیلتر}}$

به ازای هر فیلتر یک پارامتر بایاس داریم

⇒ تعداد پارامترها = 416 ⟶ ابعاد خروجی : $\frac{224-5}{1}+1=220$ ⟶ (16, 220, 220)

2) 32 x conv(5×5) : تعداد پارامترها = $(5\times5\times16 + 1)\times 32 = 12832$

→ ابعاد خروجی : $\frac{220-5}{1}+1=216$ ⟶ (32, 216, 216)

3) maxpooling(5×5, stride=(2,2)) : لایه maxpooling پارامتری ندارد

→ ابعاد خروجی : $\underbrace{\frac{216-5}{2}}_{105.5}+1$ ⟶ با توجه به آنکه حاشیه می‌تواند دور ریخته شود ⟶ $105+1=106$ ⟶ (32, 106, 106)

4) 32 x conv(5×5) : تعداد پارامترها = $(5\times5\times32 + 1)\times 32 = 25632$

→ ابعاد خروجی : $\frac{106-5}{1}+1=102$ ⟶ (32, 102, 102)

Subject:
Year Month Date ( )

5) 64 x Conv(5,5) : تعداد پارامترها = $(5\times5\times32+1)\times64 = 51264$

→ ابعاد خروجی : $\frac{102-5}{1}+1 = 98 \rightarrow (64, 98, 98)$

6) maxpooling(5x5, stride=(2,2)) پارامتر ندارد

→ ابعاد خروجی : $\frac{98-5}{2}+1 = 46.5+1$ → حاشیه دور ریخته می‌شود → $46+1=47 \rightarrow (64, 47, 47)$

7) 64 x conv(5x5) : تعداد پارامترها = $(5\times5\times64+1)\times64 = 102464$

→ ابعاد خروجی : $\frac{47-5}{1}+1 = 43 \rightarrow (64, 43, 43)$

8) 128 x conv(5x5) : تعداد پارامترها = $(5\times5\times64+1)\times128 = 204928$

→ ابعاد خروجی : $\frac{43-5}{1}+1 = 39 \rightarrow (128, 39, 39)$

9) maxpooling(5x5, stride=(2,2)) : پارامتر ندارد

ابعاد خروجی : $\frac{39-5}{2}+1 = 18 \rightarrow (128, 18, 18)$

10) FC(1024) → می‌بایست در ابتدا خروجی لایه قبل را که به صورت تنسور است را به بردار تبدیل کنیم. سپس هر نورون به 1024 نورون به صورت dense وصل است

$\Rightarrow 128\times18\times18 = 41472 \Rightarrow$ تعداد پارامترها $= 1024(41472+1)$ bias

$= 42468552$

→ ابعاد خروجی : (1024)

11) FC(100) ⇒ تعداد پارامترها = $100(1024+1) = 102500$

→ ابعاد خروجی : (100)

⇐ بیشترین تعداد پارامترها در اولین لایه FC اتفاق می‌افتد یعنی بخش 10

Network 2 : 1) 128 x conv(5x5) → تعداد پارامترها = $(5\times5\times1+1)\times128 = 3328$

→ ابعاد خروجی : $\frac{224-5}{1}+1 = 220 \rightarrow (128, 220, 220)$

2) 64 x conv(5x5) : تعداد پارامترها = $(5\times5\times128+1)\times64 = 204864$

→ ابعاد خروجی : $\frac{220-5}{1}+1 = 216 \rightarrow (64, 216, 216)$

Subject:

Year Month Date ( )

3) maxpooling(5×5, stride=(2,2)) → پارامتر ندارد

ابعاد خروجی : $\frac{216-5}{2} + 1 = 105{,}5 + 1$ حاشیه دور ریخته میشود → $105+1 = 106$

⟹ (64, 106, 106)

4) 64 × conv(5×5) ; تعداد پارامترها = $(5\times5\times64 + 1)\times64 = 102464$

→ ابعاد خروجی : $\frac{106-5}{1} + 1 = 102$ → (64, 102, 102)

5) 32 × Conv(5×5) ; تعداد پارامترها = $(5\times5\times64 + 1)\times32 = 51232$

→ ابعاد خروجی ; $\frac{102-5}{1} + 1 = 98$ → (32, 98, 98)

6) maxpooling : پارامتر ندارد / ابعاد خروجی : $\underbrace{\frac{98-5}{2}}_{46{,}5} + 1 \overset{\text{ریختن حاشیه دور}}{=} 47$

→ ابعاد : (32, 47, 47)

7) 32×conv(5,5) : تعداد پارامتر = $(5\times5\times32 + 1)\times32 = 25632$

→ ابعاد خروجی : $\frac{47-5}{1} + 1 = 43$ → (32, 43, 43)

8) 16× Conv(5,5) : تعداد پارامتر = $(5\times5\times32 + 1)\times16 = 12816$

→ ابعاد خروجی : $\frac{43-5}{1} + 1 = 39$ → (16, 39, 39)

9) maxpooling : پارامتر ندارد / ابعاد خروجی : $\frac{39-5}{2} + 1 = 18$ → (16, 18, 18)

10) FC(1024) → خروجی را به تبل flatten می کنیم → $16\times18\times18 = 5184$

که هر یک از این نورون ها به هم 1024 نورون وصل هستند

⟹ تعداد پارامترها = $(5184+1)\times1024 = 5309440$

→ ابعاد خروجی : (1024)

Subject:
Year Month Date ( )

11) FC(100) : تعداد پارامترها : $(1024+1) \times 100 = 102500$

→ ابعاد خروجی : (100)

← لایه اول FC بیشترین پارامتر را تحمیل می‌کند یعنی لایه شماره 10

← لایه‌های FC تعداد پارامترهای زیادی را به شبکه تحمیل می‌کنند

ت) در قسمت ب هنگام بررسی هر لایه انجام شد

ث) با توجه به آنکه سایز تصاویر (دقت می‌کنیم چون در صورت سوال

در لایه‌های مختلف در حافظه قرار می‌گیرند، برای فقط سایز تصاویر مورد بحث است تنها آن را در نظر گرفتیم و از در نظر گرفتن

هر دو مدل حافظه مورد نیاز را حساب می‌کنیم پارامترها خودداری کردیم)

تعداد کل ورودی‌ها در لایه‌های مختلف در مدل اول

$$= 224^2 \times 1 + 220^2 \times 16 + 216^2 \times 32 + 106^2 \times 32 + 102^2 \times 32 + 98^2 \times 64 + 47^2 \times 64 + 43^2 \times 64 + 39^2 \times 128 + 18^2 \times 128 + 1024 + 100$$

تعداد کل ورودی‌ها در لایه‌های مختلف در مدل دوم

$$= 224^2 \times 1 + 220^2 \times 128 + 216^2 \times 64 + 106^2 \times 64 + 102^2 \times 64 + 98^2 \times 32 + 47^2 \times 32 + 43^2 \times 32 + 39^2 \times 16 + 18^2 \times 16 + 1024 + 100$$

تعداد در مدل دوم > تعداد در مدل اول

⇐ مدل اول حافظه کمتری را مورد استفاده قرار می‌دهد. در نتیجه مناسب‌تر است

Vazik