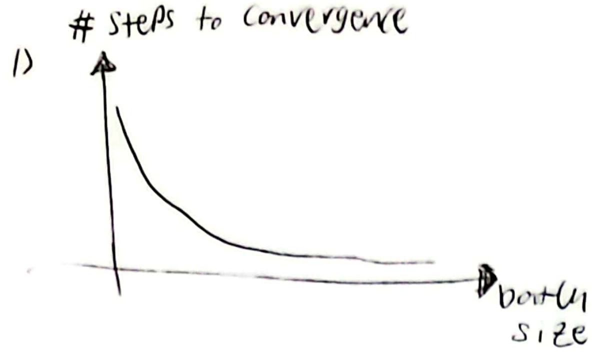
به نام خدا

متین آرمان مهر (401208442)

پاسخ تمرینات تئوری یادگیری عمیق سری دوم

1)

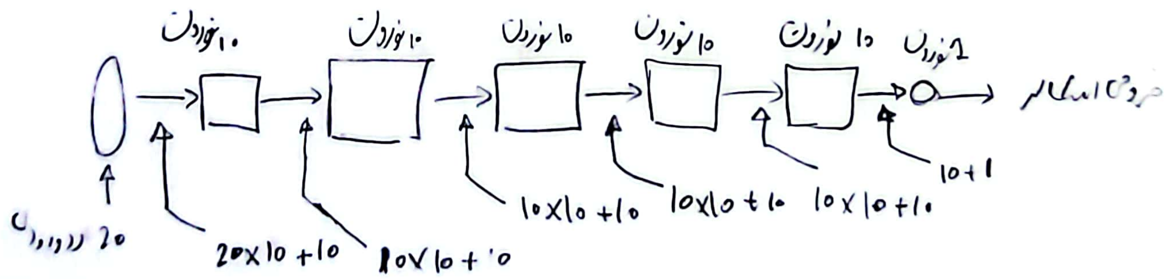


الف) با افزایش batch size، تعداد قدم هایی که به سمت ناحیه همگرایی لازم داریم کاهش می‌یابد. زیرا با در نظر گرفتن نرخ یادگیری ثابت، در batch size های بالا قدم های دقیق‌تری برمی‌داریم اما در batch size های پایین نمودار خطای آموزش پرش زیادی دارد و با تعداد قدم های زیادی converge می‌کند. از طرفی دیگر در صورتی که batch size افزایش یابد، سرعت همگرایی به دلیل محاسبات بیشتر بار کاهش می‌یابد.

ب) با توجه به اینکه در لایه batch normalization محاسبه میانگین و واریانس در یک mini batch انجام می‌شود، لذا تخمین دقیقی از میانگین و واریانس واقعی حاصل نمی‌شود. همچنین با توجه به اینکه در این لایه میانگین و واریانس داده های ورودی را توسط دو پارامتر تغییر می‌دهیم پس انگار یک نویزی به داده اصلی اضافه می‌کنیم.

ج) در صورتی که در یک شبکه FC با توابع فعالیت سیگموید وزن های اولیه یک عدد مثبت بزرگ باشد، شبکه در ناحیه اشباع قرار می‌گیرد و وزن ها به سختی تغییر می‌کنند. زیرا در تابع سیگموید در مقادیر بسیار بزرگ مشتق آن نزدیک به صفر می‌شود.

د)



تعداد کل پارامتر ها: $\underbrace{20\times10}_{\text{لایه مخفی اول}} + \underbrace{4\times10\times10}_{\text{4 لایه مخفی}} + \underbrace{5\times10}_{\text{bias های لایه ها مخفی}} + \underbrace{10}_{\text{وزن های لایه آخر}} + \underbrace{1}_{\text{بایاس لایه آخر}} = 661$

ه)

$$\hat{y} = softmax(W_S x + b_S) \rightarrow$$

خروجی هر نورون قبل از تابع softmax:

$$\begin{cases} z_1 \triangleq W_{s_1} x + b_{s_1} \\ z_2 \triangleq W_{s_2} x + b_{s_2} \end{cases} \rightarrow \begin{cases} \hat{y}_1 = softmax(z_1) \\ \hat{y}_2 = softmax(z_2) \end{cases}$$

$$\hat{y}_1 \underset{1}{\overset{0}{\gtrless}} y_2 \rightarrow \frac{e^{z_1}}{e^{z_1}+e^{z_2}} \underset{1}{\overset{0}{\gtrless}} \frac{e^{z_2}}{e^{z_1}+e^{z_2}} \rightarrow \frac{e^{z_1}-e^{z_2}}{e^{z_1}+e^{z_2}} \underset{1}{\overset{0}{\gtrless}} 0 \rightarrow \frac{1-e^{z_2-z_1}}{1+e^{z_2-z_1}}$$

$$\xrightarrow{z_1+z_2 \overset{\Delta}{=} t} \frac{1-e^{-t}}{1+e^{-t}} \underset{1}{\overset{0}{\gtrless}} 0 \rightarrow \frac{2-(1+e^{-t})}{1+e^{-t}} \underset{1}{\overset{0}{\gtrless}} 0 \rightarrow \frac{2}{1+e^{-t}} - 1 \underset{1}{\overset{0}{\gtrless}} 0 \rightarrow \sigma(t) \underset{1}{\overset{0}{\gtrless}} 0.5 \quad t = z_1 - z_2$$

بنابراین اگر دو خروجی طبقه‌بند softmax را به صورت $z_1 - z_2$ تعریف کنیم رویکرد ما مشابه طبقه‌بند اول خواهد بود پس افزایش تعداد پارامترها تاثیری در توانایی طبقه‌بند در حل مسائل پیچیده نداشته است یعنی:

$$W_{s_1}x + b_1 - W_{s_2}x + b_2 = W_l x + b_l$$

---

2) الف) در این مقاله کمیته به سیاست یا رویکردی گفته می‌شود که n طبقه‌بند را با یکدیگر ترکیب کرده و یک نتیجه با دقت بالاتر ارائه دهد. بنابراین در خروجی نتایج طبقه‌بندهای k کلاسه (تابع) یک لایه مخفی کمیته قرار می‌دهیم تا ترکیب خطی خروجی‌های n طبقه‌بند را محاسبه کرده و وزن‌های بهینه را یاد بگیرد.

$$y^k_{com}(x) = \sum_{n=1}^{N} w_{nk}\, y_{nk}(x)$$

خروجی‌های لایه کمیته (k کلاس)

وزن اعمال شونده خروجی کلاس kام طبقه‌بند nام در لایه کمیته

خروجی طبقه‌بند nام در کلاس k

در هر طبقه‌بند از رویکرد متفاوتی در پیش‌پردازش داده‌ها استفاده شده است به گونه‌ای که خطای هر طبقه‌بند با بقیه ناهمبسته باشد.

ب) در پیش‌پردازش داده‌ها از روش‌های data augmentation استفاده شده است:

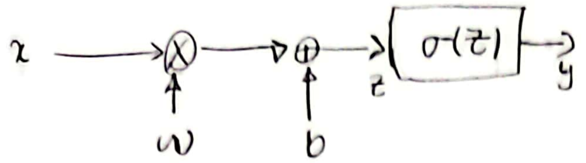
1) ضریب distortion که وضوح تصویر را تغییر می‌دهد. (σ و α)

2) چرخش تصادفی عکس‌ها با پارامتر β

3) scale کردن داده‌ها در محورهای افقی یا عمودی با پارامترهای γ_x , γ_y

۳) الف)

$$z = wx + b \ , \ y = \sigma(z) \ , \ L = \frac{1}{2}(y-t)^2 \ , \ R = \frac{1}{2}w^2 \ , \ L_{reg} = L + \lambda R$$



$$\frac{\partial L_{reg}}{\partial w} = \underbrace{\frac{\partial L}{\partial w}}_{} + \lambda \underbrace{\frac{\partial R}{\partial w}}_{=w} = \sigma^2(wx+b)(1-\sigma(wx+b))(y-t)x$$

$$\frac{\partial L}{\partial w} = \underbrace{\frac{\partial L}{\partial y}}_{1} \times \underbrace{\frac{\partial y}{\partial z}}_{2} \times \underbrace{\frac{\partial z}{\partial w}}_{3}$$

$$1: \frac{\partial L}{\partial y} = \frac{1}{2} \times 2 \times (y-t) \qquad 2: \frac{\partial y}{\partial z} = \frac{\partial}{\partial z}\left(\frac{1}{1+e^{-z}}\right) = \frac{e^{-z}+1-1}{(1+e^{-z})^2} = \sigma(z) - \sigma^2(z)$$

$$3: \frac{\partial z}{\partial w} = x$$

$$\Rightarrow \frac{\partial L}{\partial w} = (\sigma(wx+b))x(\sigma(wx+b) - \sigma^2(wx+b)) \times x = \sigma^2(wx+b)(1-\sigma(wx+b))(y-t)x$$

$$\frac{\partial L_{reg}}{\partial x} = \underbrace{\frac{\partial L}{\partial y}}_{1} \times \underbrace{\frac{\partial y}{\partial z}}_{2} \times \underbrace{\frac{\partial z}{\partial x}}_{=w} + \lambda \frac{\partial R}{\partial x} = 0 \ = \sigma^2(wx+b)(1-\sigma(wx+b))w(y-t)$$

$$\frac{\partial L_{reg}}{\partial b} = \underbrace{\frac{\partial L}{\partial y}}_{1} \times \underbrace{\frac{\partial y}{\partial z}}_{2} \times \underbrace{\frac{\partial z}{\partial b}}_{=1} + \lambda \underbrace{\frac{\partial R}{\partial b}}_{=0} = \sigma^2(wx+b)(1-\sigma(wx+b))(y-t)$$

$$\frac{\partial L_{reg}}{\partial \lambda} = R = \frac{1}{2}w^2 \ , \ \frac{\partial L_{reg}}{\partial t} = \frac{\partial L}{\partial t} = t-y = t-\sigma(wx+b)$$

ب) * اگر همه نورون‌ها یکسان وزن (۰) شوند، همگی خروجی یکسانی تولید می‌کنند و شبکه نمی‌تواند مسائل متنوع را حل کند و ممکن است تابع هزینه در یک local minimum یا saddle point گیر کند. سپس وزن‌دهی به صورت تصادفی انجام می‌شود.

* در صورتی که از وزن‌های با مقادیر اولیه بالا استفاده کنیم باعث افزایش حجم محاسبات می‌شود و ممکن است ایجاد gradient exploding شود.

ج) [illegible]:

$$\left|\begin{array}{l} w^{(0)} = 0.1 \\ \mu = 0.1 \\ x^{(1)} = 2,\ x^{(2)} = 1,\ x^{(3)} = 2,\ x^{(4)} = 1 \\ b^{(0)} = 0 \\ t^{(1)} = 0,\ t^{(2)} = 1,\ t^{(3)} = 0,\ t^{(4)} = 1 \end{array}\right. \rightarrow$$

$$w^{(1)} = w^{(0)} - \mu \frac{\partial L_{reg}}{\partial w^{(0)}} = 0.1 - 0.1\sigma^2(0.2)(1-\sigma(0.2))2\times(\sigma(0.2)-0) - 0 = 0.1372$$

$$b^{(1)} = b^{(0)} - \frac{\partial L_{reg}}{\partial b^{(0)}} = 0 - 0.1\sigma^2(0.2)(1-\sigma(0.2))0.1\times(\sigma(0.2)-0) = 1$$

$$w^{(2)} = 0.1372 - 0.1\sigma^2(0.1386)(1-\sigma(0.1386))\times(\sigma(0.1386)-1) - 0.1372 = 0.0014 = 0.1393$$

$$b^{(2)} = 0.014 - 0.1\sigma^2(0.1386)(1-\sigma(0.1386))\times(\sigma(0.1386)-1)\times 0.1372 \simeq -1.9\times10^{-4}$$

$$w^{(3)} = 0.1811,\quad b^{(3)} = 0.0018$$

$$w^{(4)} = 0.1879,\quad b^{(4)} = -0.00024$$

---

4) الف) الگوریتم ADAM به منظور بکارگیری نرخ یادگیری متغیر در شبکه های عصبی استفاده می شود:

* $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ → در این خط گرادیان تابع هزینه نسبت به ضرایب θ از زمان $t-1$ محاسبه شده و در متغیر g_t ذخیره می شود.

* $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$ → در این مرحله متغیر momentum مرتبه اول از روی مقادیر قبلی آن و گرادیان فعلی با توجه به وزن β_1 آپدیت می شود.

* $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ → در این مرحله متغیر مرتبه دوم که به آن velocity هم گفته می شود از روی مقادیر قبلی خودش و مجذور متغیر گرادیان مرحله فعلی آپدیت می شود (با وزن β_2)

* $\hat{m}_t \leftarrow \frac{m_1}{1-\beta_1^t}$, $\hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$ این دو term در واقع بایاس مربوط به مقادیر m_t و v_t را در مراحل اولیه آپدیت آن ها از بین می برد.

* $\theta_t \leftarrow \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$ → در نهایت با توجه به مقادیر بدست آمده از $\hat{m}_t$ و $\hat{v}_t$ وزن های شبکه با نرخ ثابت α و ضریب متغیر $\frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$ آپدیت می شوند. دلیل وجود متغیر ϵ، جلوگیری از صفر شدن مخرج کسر است.

ب) با توجه به اینکه $\beta_1, \beta_2 = (0,1)$ هستند و مقادیر اولیه m_0 و v_0 برابر صفر در نظر گرفته می شوند لذا در مراحل اولیه آپدیت شدن m_t و v_t، آن ها به سمت صفر بایاس دارند و سرعت همگرایی به دلیل کوچک شدن ضرایب گرادیان ها پایین است. لذا عبارات $\hat{m}_t$ و $\hat{v}_t$ می توانند از این اتفاق جلوگیری کرده و با تقسیم $\frac{1}{1-\beta_1^t}$ و $\frac{1}{1-\beta_2^t}$ سرعت همگرایی را در زمان های اولیه افزایش دهند. رفته رفته با افزایش t، $\hat{m}_t \simeq m_t$ و $\hat{v}_t = v_t$ خواهد شد.

5) الف)

$$\underline{w}^{(t+1)} = \underline{w}^{(t)} - \varepsilon \underbrace{\nabla f(\underline{w})}_{=2H\underline{w}^{(t)}} \Rightarrow w^{(t+1)} = w^{(t)} - 2\varepsilon H w^{(t)} = w^{(t)}(I - 2\varepsilon H)$$

ب)

$$w^{(1)} = w^{(0)}(I - 2\varepsilon H) \rightarrow w^{(2)} = w^{(1)}(I - 2\varepsilon H) = w^{0}(I - 2\varepsilon H)^{2} \rightarrow \cdots$$

$$\rightarrow w^{(t)} = w^{(0)}(I - 2\varepsilon H)^{t}$$

ج)

$$\underline{w}^{(t+1)} = \underline{w}^{(t)}(I - 2\varepsilon H) \rightarrow \underline{w}^{(t+1)} = \underline{w}^{(t)}(I - 2\varepsilon Q \Lambda Q^{T}) \xrightarrow{\text{ضرب Q از راست}} \underline{w}^{(t+1)} Q = \underline{w}^{(t)}(Q - 2\varepsilon Q \Lambda \underbrace{Q^{T} Q}_{=I})$$

$$\xrightarrow{\text{ضرب } Q^T \text{ از چپ}} Q^{T} \underline{w}^{(t+1)} Q = \underline{w}^{(t)}(\underbrace{Q^{T} Q}_{=I} - 2\varepsilon \underbrace{Q^{T} Q}_{=I} \Lambda) \rightarrow Q^{T} \underline{w}^{(t)} Q = \underline{w}^{(0)}(I - 2\varepsilon \Lambda)^{t}$$

شرط همگرایی: $|I - 2\varepsilon \Lambda| < I$:

$$\rightarrow |1 - 2\varepsilon \lambda_i| < 1 \quad \forall i = 1, \dots, \underset{\text{تعداد بعد داده}}{n} \rightarrow |1 - 2\varepsilon \lambda_{max}| < 1 \rightarrow \boxed{0 < \varepsilon < \frac{1}{2\lambda_{max}}}$$

د) روش نیوتن:

$$\nabla f(w) = 2Hw \rightarrow \nabla^2 f(\underline{w}) = 2H \rightarrow (\nabla^2 f(\underline{w}))^{-1} = \frac{1}{2} H^{-1}$$

$$w^{(t+1)} = w^{(t)} - \varepsilon \Big(\underbrace{(\nabla^2 f(\underline{w}))^{-1}}_{=\frac{1}{2}H^{-1}} \underbrace{\nabla f(\underline{w})}_{=2H\underline{w}^{(t)}} \Big) = w^{(t)} - \varepsilon w^{(t)} = w^{(t)}(1 - \varepsilon)$$

$$\rightarrow w^{(t)} = w^{(0)}(1 - \varepsilon)^{t} \xrightarrow{\hspace{3em}} \text{در صورتی که } \varepsilon = 1 \rightarrow w^{(t)} = 0 \rightarrow \text{در یک گام همگرا می‌شویم!}$$

شرط همگرایی: $0 < \varepsilon < 1$

ه) در مسائلی که با تابع هزینه پیچیده روبرو هستیم، بدست آوردن ماتریس هسین که از مشتق مرتبه 2 بدست می‌آید و معکوس گرفتن از آن ~ که تا محاسبات بسیار پیچیده می‌شود. از طرفی برای ذخیره مقادیر ماتریس هسین با محدودیت حافظه روبرو هستیم. همچنین در صورتی که در یک Saddle Point قرار بگیریم، معکوس ماتریس هسین مقادیر بزرگی به خود می‌گیرد و محاسبات را پیچیده‌تر می‌کند.

الف) (6

$$J_1 = \frac{1}{2}\left(y_d - \sum_{k=1}^{n}\left((w_k - \delta_k) x_k\right)\right)^2$$

$$= \frac{1}{2}\left(y_d - \sum_{k=1}^{n} w_k x_k + \sum_{k=1}^{n} \delta_k x_k\right)^2 = \underbrace{\frac{1}{2}\left(y_d - \sum_{k=1}^{n} w_k x_k\right)^2}_{\triangleq J_0(w)} - \left(\sum_{k=1}^{n} \delta_k x_k\right)\left(y_d - \sum_{k=1}^{n} w_k x_k\right) + \left(\sum_{k=1}^{n} \delta_k x_k\right)^2 \times \frac{1}{2}$$

$$E\left\{\frac{\partial J_1}{\partial w_i}\right\} = E\left\{\frac{\partial J_0(w)}{\partial w_i}\right\} - \underbrace{\frac{\partial}{\partial w_i}\left(\left(\sum_{k=1}^{n} \underbrace{E\{\delta_k\}}_{=0} x_k\right)\left(y_d - \sum_{k=1}^{n} w_k x_k\right)\right)}_{=0} + \frac{\partial}{\partial w_i}\left(\sum_{k_1=1}^{n} \sum_{k_2}^{n} E\{\delta_{k_1}\delta_{k_2}\} x_{k_1} x_{k_2}\right)$$

$$= E\left\{\frac{\partial J_0(w)}{\partial w_i}\right\} + \frac{\partial}{\partial w_i}\left(\sum_{k=1}^{n} \underbrace{E\{\delta_k^2\}}_{=\alpha w_k^2} x_k^2 + \sum_{k_1=1}^{n} \sum_{\substack{k_2=1 \\ k_2 \neq k_1}}^{n} \underbrace{E\{\delta_{k_1}\delta_{k_2}\}}_{=\underbrace{E\{\delta_{k_1}\}}_{=0}\underbrace{E\{\delta_{k_2}\}}_{=0}=0} x_{k_1} x_{k_2}\right)$$

$$\rightarrow E\left\{\frac{\partial J_1}{\partial w_i}\right\} = E\left\{\frac{\partial J_0(w)}{\partial w_i}\right\} + 2\alpha w_i x_i^2$$

ب) با توجه به اضافه شدن عبارت $2\alpha w_i x_i^2$ به مشتق متوسط تابع هزینه، نشان‌دهنده اعمال شدن regularization به شبکه است.
که توسط dropout انجام می‌شود.