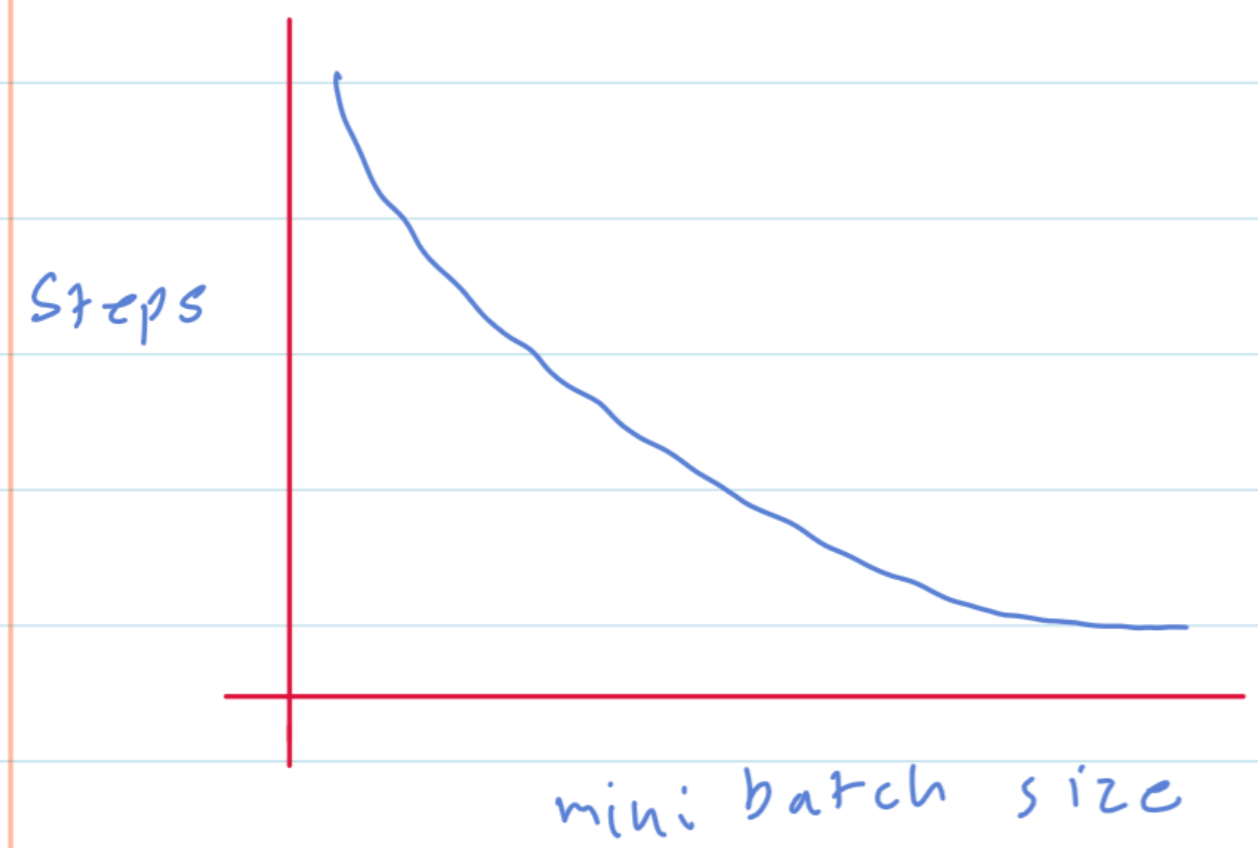
تمرین سری 2 یادگیری عمیق — حسام حسنی - 400101034



سوال ۱- آ) با افزایش سایز mini-batch در هر step محاسبه گرادیان دیتای بیشتری استفاده می‌شود که باعث می‌شود واریانس تخمین گر گرادیان کم شود (n در مخرج ظاهر می‌شود) پس تخمین گر گرادیان به مقدار واقعی خود نزدیک تر می‌شود و در جهتی که درست تر است حرکت می‌کنیم البته این اتفاق باعث می‌شود که زمان لازم برای محاسبه گرادیان در هر مرحله زیاد شود همچنین حافظه بیشتری نیاز داریم و همچنین چون تخمین گر ما نسبت به دیتای train دقیق تر می‌شود احتمال یادگرفتن نویز دیتا و در نتیجه overfitting بیشتر می‌شود

ب) مراحل Batch normalization :

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 = \frac{1}{m}\sum (x_i - \mu_B)^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

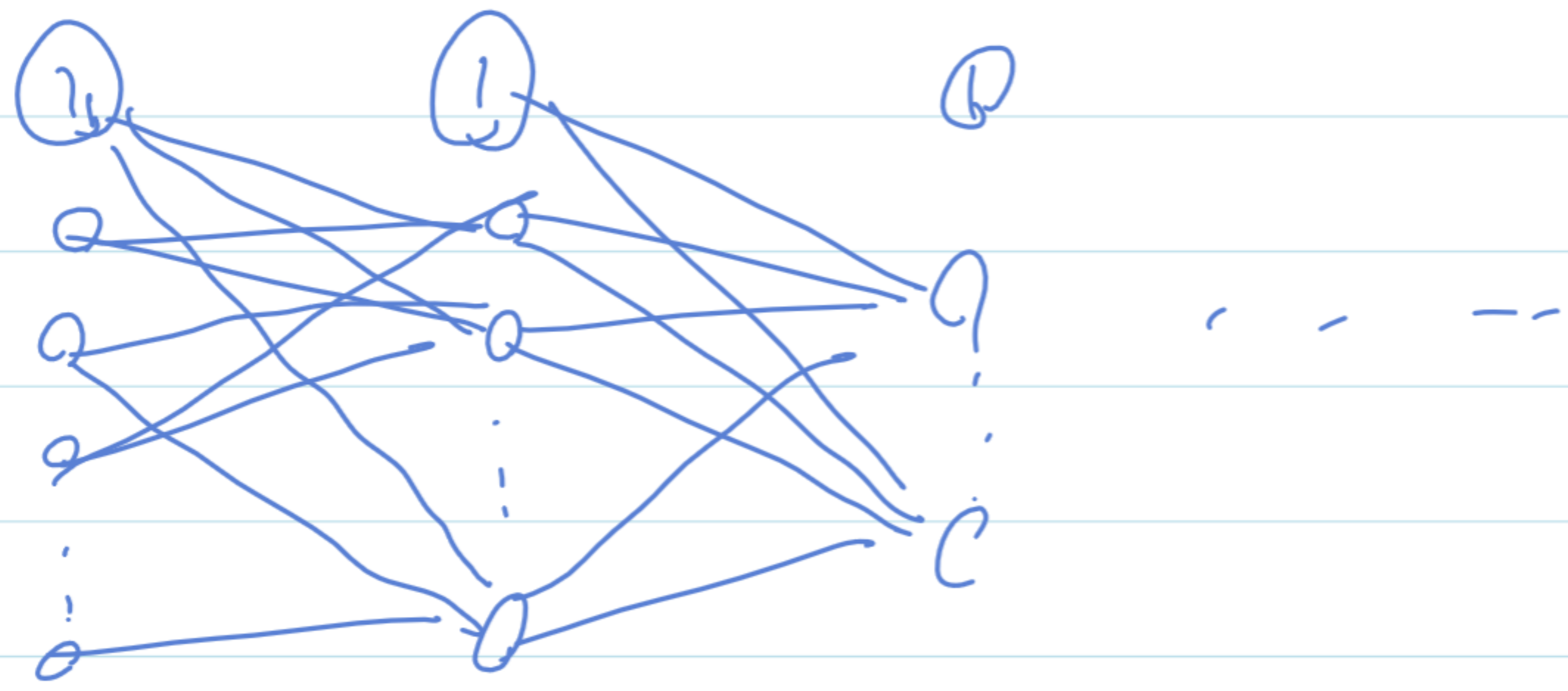
$$y_i = \gamma x_i + \beta \equiv BN_{\gamma,\beta}(x_i)$$

بله:

μ_B و σ_B تخمینی از μ و σ واقعی هستند که واضحا شامل نویزی نسبت به مقادیر واقعی است و در نهایت باعث می‌شود که update وزن ها با توام مقداری نویز انجام شود که می‌تواند از overfitting جلوگیری کند

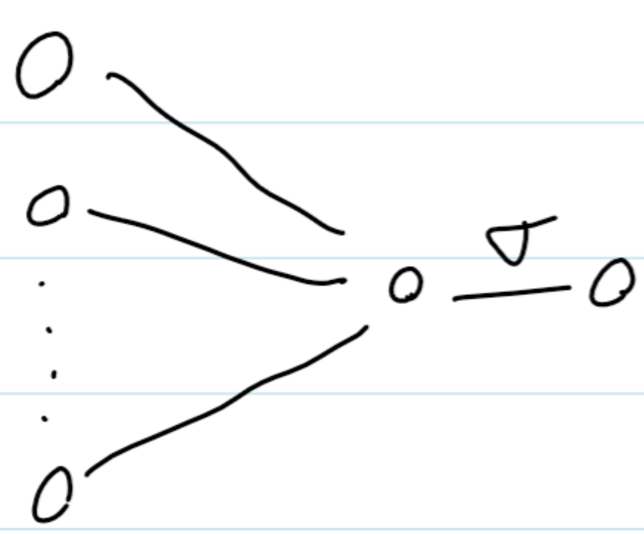
ج) خیر چون وقتی وزن ها بزرگ باشند احتمال بزرگ شدن ورودی لایه activation زیاد می‌شود که در این حالت به بخش اشباع تابع سیگموید می‌رویم که در این ناحیه گرادیان به سمت صفر می‌رود و آپدیت کردن وزن ها خیلی کند می‌شود و همچنین با افزایش تعداد لایه‌ها گرادیان ونیشینگ رخ می‌دهد که train کردن شبکه را تقریبا غیر ممکن می‌کند

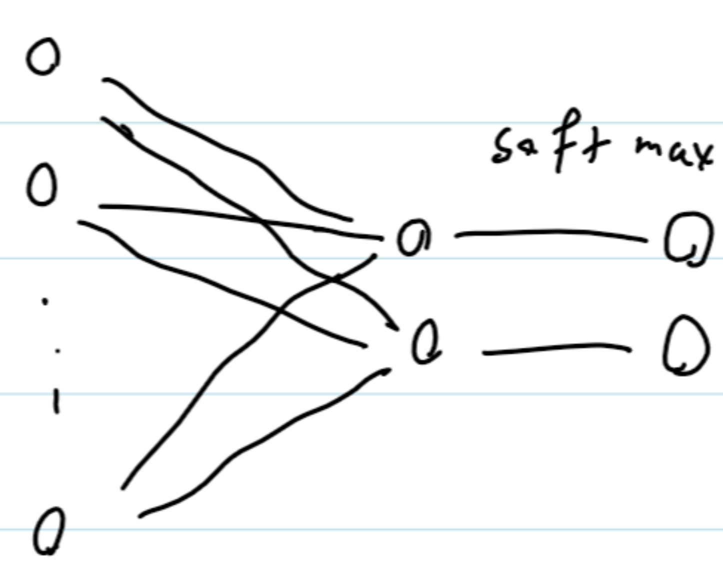
د) برای راحتی bias ها شبکه را جزو وزن ها در نظر میگیریم و ورودی ۱ را برای این وزن در نظر میگیریم



$$(20+1)\times 10 + \big((10+1)\times 10\big)\times 4 + 10+1 = 661$$

ه)





مشهوداً اینگونه میتوان استدلال نمود که چون $\hat{y} = \hat{y}_1 = 1-\hat{y}_2$ دانستن یکی از این دو برای ما کافی است و نیازی به یادگیری هردو نیست

استدلال ریاضی :

$$\hat{y}_1 = \frac{e^{w_{s_1}x+b_{s_1}}}{e^{w_{s_1}x+b_{s_1}}+e^{w_{s_2}x+b_{s_2}}} = \frac{1}{1+e^{-((w_{s_1}-w_{s_2})x+b_{s_1}-b_{s_2})}} = \sigma\big((w_{s_1}-w_{s_2})x+b_{s_1}-b_{s_2}\big)$$

$$\hat{y}_2 = \frac{e^{w_{s_2}x+b_{s_2}}}{e^{w_{s_2}x+b_{s_2}}+e^{w_{s_1}x+b_{s_1}}} = \frac{1}{1+e^{-((w_{s_2}-w_{s_1})x+b_{s_2}-b_{s_1})}} = \sigma\big((w_{s_2}-w_{s_1})x+b_{s_2}-b_{s_1}\big)$$

اگر $\hat{y} = y_1$ بگیریم

$$\hat{y} = \sigma(w_e x + b_e) \Longrightarrow w_e = w_{s_1} - w_{s_2} \text{ و } b_e = b_{s_1} - b_{s_2}$$

پس داشتن دو ماتریس وزن w_{s_1} و w_{s_2} ضروری نیست و میتوان با یک w_e دقیقا همان کار را انجام داد

سوال ۲- الف) یک کمیته به مجموعه ای از شبکه های عصبی ساده چندگانه اشاره می‌کند برای اینکه کمیته به بهترین شکل کار کنند error ها نباید correlated باشند، که این می‌تواند با استفاده از طبقه بند های مختلف و دادگان آموزشی مختلف انجام بپذیرد. دلیل استفاده از کمیته کاهش واریانس در پیش‌بینی انجام شده توسط هر مدل و بهبود دقت نهایی کمیته با انجام majority voting یا majority voting وزن دار است.

یک طبقه بند ← $y^{k}_{com}(x) = \sum_{n=1}^{N} w_{nk}\, y_{nk}(x)$

ب) اگر دادگان آموزشی به طور مداوم تغییر شکل دهد از تکنیک های Data Augmentation استفاده می‌کنیم: تغییر شکل های الاستیک، تغییر مقیاس (افقی و عمودی) و چرخش، تغییر شکل های affine که با پارامتر های زیر انجام می‌شود

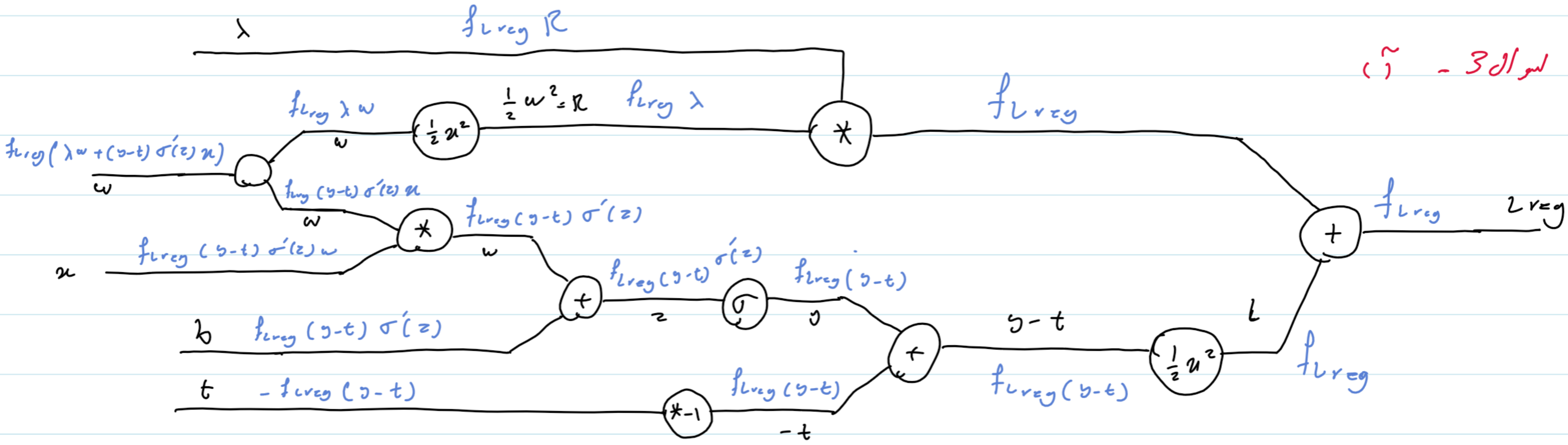
- σ و α برای اعوجاج الاستیک که از نوسانات عضلات دست ناشی می‌شود
- β یک زاویه تصادفی بین $[-\beta, \beta]$ که یا چرخش یا برش افقی را توصیف می‌کند و در صورت برش $\tan\beta$ نسبت جابجایی افقی و ارتفاع تصویر می‌باشد
- γ_x و γ_y که برای scaling افقی و عمودی هستند که به صورت رندوم از $[1-\gamma/100,\ 1+\gamma/100]$ برداشته می‌شوند

همچنین با Bounding-Box های مختلف با عرض 8 تا 20 و step size = 2

دیتا نورمالایز می‌شود که باعث می‌شود 7 دیتاست مختلف داشته باشیم

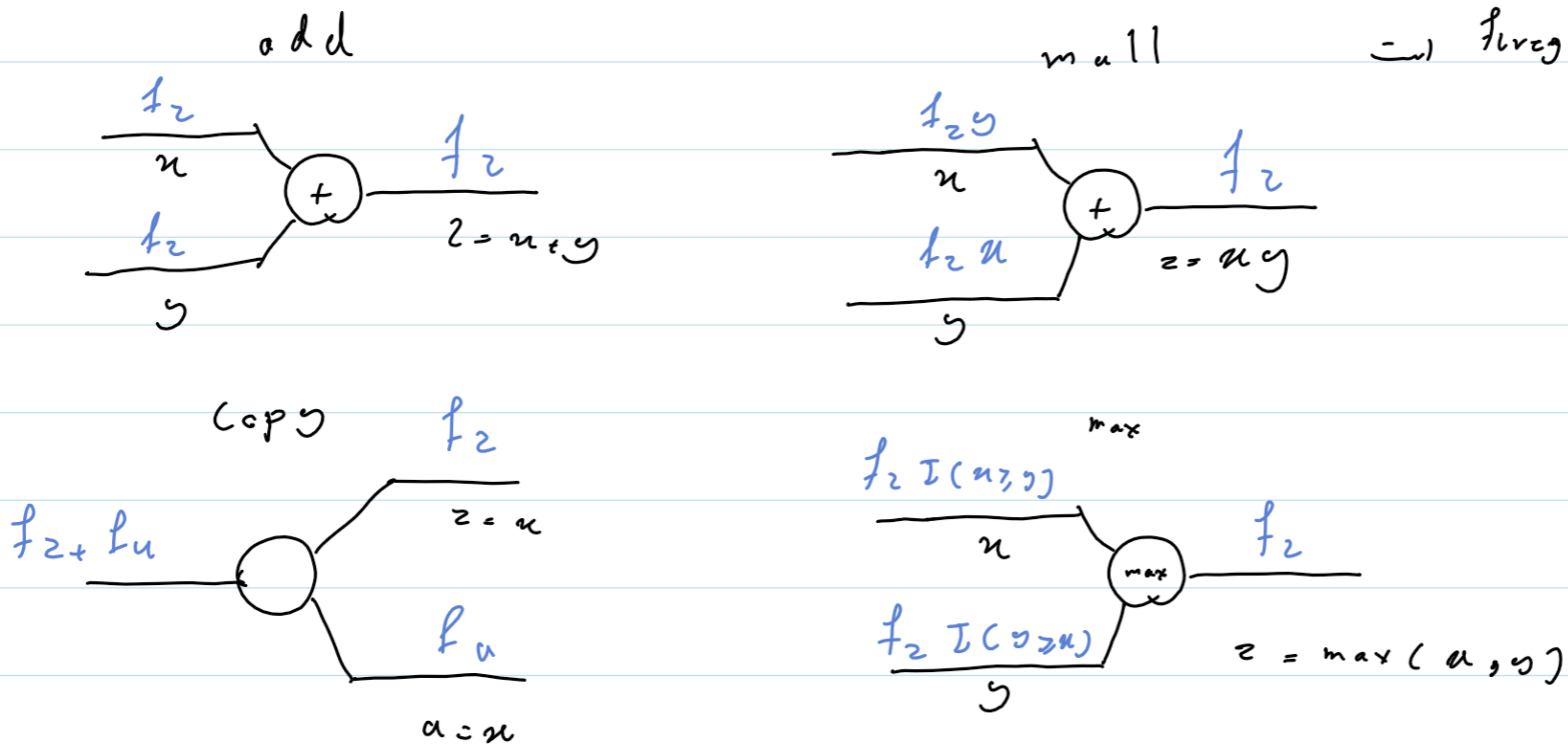
در نهایت همه این کار های بالا باعث می‌شود که تنوع چیزهایی که یاد می‌گیرد بالا رود و منجر به خطا های غیر همبسته شود و در نهایت خطای کل را کاهش دهد

سوال 3 - آ)



نوشته های سیاه مقادیر ارزش نقطه گراف محاسباتی هستند و نوشته های آبی مقادیر گرادیان نسبت به Loss هستند به شرطی که $f_{reg} = \frac{\partial loss}{\partial L_{reg}}$

گرادیان Lreg به loss باشد $f_{reg} = 1$ است



توضیح گیت ها:

مشتقات نسبت به Lreg:

$$\frac{\partial L_{reg}}{\partial w} = \lambda w + (y-t)\,\sigma(z)\,w \qquad \frac{\partial L_{reg}}{\partial L_{reg}} = R$$

$$\frac{\partial L_{reg}}{\partial x} = (y-t)\,\sigma'(z)\,w \qquad \frac{\partial L_{reg}}{\partial t} = -(y-t)$$

$$\frac{\partial L_{reg}}{\partial b} = (y-t)\,\sigma'(z) \qquad \sigma'(z) = \frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1-\sigma(z))$$

گرادیان تنها w و b پارامترهای قابل یادگیری هستند و در هر step به اندازه منفی گرادیان با مقدار خود جمع می‌شوند

$$w_{t+1} = w_t - \frac{\partial L_{reg}}{\partial w}$$

$$b_{t+1} = b_t - \frac{\partial L_{reg}}{\partial b}$$

ب) همان طور که در سوال ۱ بخش ج توضیح داده شده بزرگ شدن وزن ها ممکن است باعث شود به نواحی اشباع توابعی مانند سیگموید یا tanh برویم که دارای ناحیه گرادیان تقریبا صفر است و خارج شدن از این ناحیه سخت می باشد همچنین در صورت استفاده از ترم رگولاریزیشن مانند L2 باعث می شود loss خیلی بزرگ شده و converge کردن سخت شود و مشکلاتی مانند exploding graidient رخ دهد

همچنین اگر همه مقادیر وزن کم باشد ممکن است مشکل کم شدن متغیرهای شبکه رخ دهد که به اصطلاح به آن symmetry problem می گویند که واقعا ظرفیت شبکه را کاهش می دهد که میتوانیم در یک local minimum گیر کرده ایم که فاصله اش از گلوبال minimum زیاد است

و یا ممکن است نسبت به ویژگی های بایاس شویم و شبکه قابلیت جنرالیزیشن خود را از دست بدهد

ج)

initial valeu : $\alpha = 0.1$ $w_0 = 0.5$ $b = 0.5$ $\lambda = 0.1$

$$\Rightarrow z_0 = 0.5x + 0.5 \quad y_0 = \sigma(0.5x + 0.5) \quad \sigma'(z_0) = y_0(1-y_0)$$

$$\Rightarrow \frac{\partial L_{reg}}{\partial w} = 0.1 \times 0.5 + x(y_0 - t)y_0(1-y_0)$$

$$\Rightarrow \frac{\partial L_{reg}}{\partial b} = (y_0 - t)y_0(1-y_0)$$

$$\Rightarrow w_1 = 0.5 - 0.1\left(0.05 + x(y_0 - t)y_0(1-y_0)\right)$$

$$b = 0.5 - 0.1\left((y_0 - t)y_0(1-y_0)\right)$$

سوال 4 _ a)

$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$: محاسبه گرادیان loss-fn و ذخیره آن در g_t

$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$: محاسبه m_t که ضریبی از m_t های قبلی است (β_1) و همچنین مقدار جدید گرادیان

را به اندازه $(1-\beta_1)$ تاثیر می‌دهد که میانگین نمایی از گرادیان های قبلی است

در واقع این ترم برای این است که از خواص momentum update استفاده کنیم

$$m_t = (1-\beta_1) g_t + (1-\beta_1)\beta_1 g_{t} + (1-\beta_1)\beta_1^2 g_{t-2} \cdots + (1-\beta_1)\beta_1^t m_0$$

$$\frac{1}{1-\beta_1} = 1 + \beta_1 + \beta_1^2 + \cdots +$$

← مجموع ضرایب برابر ۱ می‌شود و ضرایب به صورت نمایی کاهش می‌یابد

$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$: مشابه RMS prop این ترم برای این است مقدار حرکت کردن به سمت جهت‌های

مختلف Balence شود به این صورت که در زمانی که در سطح تخت هستیم learning rate بزرگ شود و در دره ها که ممکن

است از مقادیر بیشینه optimal learning rate کوچک شود β_2 همان decay rate است و می‌گوید

چقدر به مقادیر قبلی اهمیت بدهد و در نهایت v_t میانگین نمایی از مقادیر قبلی خود به علاوه مقدار جدید g_t است

$\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$ برای حل مشکل Bias m_t استفاده می‌شود که در بخش بعدی توضیح می‌دهم

$\hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$ برای حل مشکل Bias v_t استفاده می‌شود

$\theta_t \leftarrow \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$ همان‌گونه که توضیح داده شد با استفاده از $\hat{m}_t$ و $\hat{v}_t$ گرادیان نورمالایز شده را

استفاده می‌کنیم و α همان learning rate اصلی است و همچنین در مخرج ϵ داریم که برای

این است که تقسیم به صفر رخ نده یا در هنگام کوچک بودن $\hat{v}_t$ مقدار زیادی حرکت نکنیم

ب) طبق رابطه ای که در قسمت قبل برای m_t نوشتیم این مقدار در t های اولیه به سمت m_0 بایاس

شده است برای درست کردن بایاس اینگونه عمل می‌کنیم

adam paper

$$E(m_t) = E\left[(1-\beta_1)\sum \beta_1^{t-i} g_i\right] = E(g_t)(1-\beta_1^t) + \zeta$$

ζ برابر صفر خواهد شد اگر $E(g_t)$ stationary باشد و در غیر این صورت هم کوچک خواهد بود

با انتخاب درست β

با تقسیم بر $\frac{1}{1-\beta_1^t}$ $\hat{m}_t$ تخمین unbiased می‌شود

in $t=1$:

$$m_1 = \beta_1 m_0 + (1-\beta_1) g_t \Rightarrow \hat{m}_1 = \frac{m_1 - \beta m_0}{1-\beta_1} = \frac{m_1}{1-\beta}$$

$m_0 = 0$

سوال 5 - آ)

$$\frac{\partial L}{\partial w} = \frac{\partial w^T H w}{\partial w} \overset{H^T=H}{=} 2Hw$$

$$\Rightarrow w_{t+1} = w_t - \alpha(2Hw_t) = (I - 2\alpha H) w_t = Q(I - 2\alpha\lambda) Q^T w_t$$

ب) با توجه به جواب قسمت آ در هر step مقدار w در $Q(I-2\alpha\lambda)Q^T$ ضرب می‌شود

$$\Rightarrow w_t = [Q(I-2\alpha\lambda)Q^T]^t w_0 = Q \underbrace{(I-2\alpha\lambda)^t}_{\text{ماتریس قطری تنها درایه‌های قطر اصلی به توان t می‌رسد}} Q^T w_0$$

ج) شرط همگرایی این است که درایه‌های ماتریس قطری $(I-2\alpha\lambda)^t$ همگرا شود

$$|1 - 2\alpha\lambda_i| \leq 1 \quad \forall i \in \{1, \dots, n\}$$

$$\Rightarrow -1 \leq 1 - 2\alpha\lambda_i \leq 1 \Rightarrow \underset{psd}{\lambda_i \geq 0} \quad \text{و} \quad \alpha\lambda_i \leq 1 \Rightarrow \alpha \leq \frac{1}{\lambda_i}$$

پس شرط همگرایی این است که اولا ماتریس λ psd باشد و ثانیا نرخ یادگیری کوچک‌تر از $\frac{1}{\lambda_{max}}$ باشد

د) برای استفاده از روش نیوتن نیازمند معکوس ماتریس هسین هستیم که به نوعی مشتق دوم loss function است

$$\mathcal{H} = \left[\frac{\partial \nabla L}{\partial w}\right]^T = \left[\frac{\partial 2Hw}{\partial w}\right]^T = 2H \Rightarrow \mathcal{H}^{-1} = \frac{H^{-1}}{2}$$

$$w_{t+1} = w_t - \frac{H^{-1}}{2}(2Hw_t) = w_t - w_t = 0 \rightarrow \text{مقدار مینیموم}$$

پس توانستیم در یک مرحله همگرا شویم البته چون تابع هدف درجه 2 بود این اتفاق افتاد به طور کلی همگرایی روش نیوتن از اردر $O(\frac{1}{t^2})$ است و در مقابل روش گرادیان دیسنت که $O(\frac{1}{t})$ است همگرایی سریع‌تری دارد

ذ) زیرا محاسبه ماتریس هسین هم به حافظه زیادی $O(n^2)$ هم مقدار محاسبات زیادی می‌طلبد همچنین نیازمند محاسبه معکوس آن نیز هستیم که در نهایت اردر محاسباتی $O(n^3)$ دارد و با توجه به تعداد پارامترهای شبکه‌های عمیق این کار عملا ممکن نیست

البته روش‌هایی برای تخمین این ماتریس هسین وجود دارد که اردرهای محاسباتی کمتری دارند

سوال ۶- الف)

$$E\left[\frac{\partial J_1}{\partial w_k}\right] = \frac{\partial E[J_1]}{\partial w_k}$$

$$E[J_1] = E\left[\frac{1}{2}\Big(y_d - \overbrace{\sum_{k=1}^{n}(w_k+\delta_k)}^{\tilde{w}} x_k\Big)^2\right] = E\left[\frac{1}{2}(y_d - \tilde{w}X)^2\right]$$

$$= \frac{1}{2}E\left[y_d^2 - 2y_d\tilde{w}^T X + X^T\tilde{w}\tilde{w}^T X\right] \qquad \text{if } \operatorname{cov}(\delta_i, \delta_j) = 0 \quad i \neq j$$

$$E[\tilde{w}] = w \quad \Leftarrow$$

$$E[(\tilde{w}w^T)_{ij}] = E\left[(w_i w_j + \delta_i\delta_j + \delta_i w_j + \delta_j w_i)\right] = w_i w_j + E(\delta_i, \delta_j)$$

$$\Rightarrow E(\tilde{w}w^T) = ww^T + \alpha A \qquad A_{ij} = \begin{bmatrix} w_1^2 & & 0 \\ & w_2^2 & \\ 0 & \cdots & w_n^2 \end{bmatrix}$$

$$\Rightarrow E(J_1) = \frac{1}{2}\left(y_d^2 + 2y_d w^T X + (w^T x)^2 + \alpha x^T A x\right)$$

$$= \frac{1}{2}\Big(y_d - \sum_{i=1}^{n} w_i x_i\Big)^2 + \frac{1}{2}\alpha\sum_{i=1}^{n} w_i^2 x_i^2 = J_0 + \frac{1}{2}\alpha\sum_{i=1}^{n} w_i^2 x_i^2$$

$$\Rightarrow E\left(\frac{\partial J_1}{\partial w_k}\right) = -\nabla J_0 + \alpha w_k x_k^2$$

$$= -\Big(y_d - \sum_{i=1}^{n} w_i x_i\Big)x_k + \alpha w_k x_k^2$$

ب) با توجه به مقدار بدست آمده برای $E(J_1)$ به طور میانگین مانند رگولاریزیشن L_2 عمل می‌کند و تنها فرق آن این است که به اندازه x_k به آن اهمیت می‌دهیم