No.

Date

HW1 整理

Topic 1: Why learning can't be too big / lr & σ

Simpliest Formulation: $\sigma w = y$ (scalar equation)

Loss: $L(w) = (y - \sigma w)^2$, η is learning rate

$\therefore$ GD: $w_{t+1} = w_t + 2\eta\sigma(y - \sigma w_t)$

$= (1 - 2\eta\sigma^2) w_t + 2\eta\sigma y$

Important thought! ⇑

We know $w^* = y/\sigma$, and want $|w_t - w^*|$ converge!

$\therefore w_{t+1} - \frac{y}{\sigma} = (1 - 2\eta\sigma^2)(w_t - \frac{y}{\sigma})$

$w_{t+1} = \frac{y}{\sigma} + (1 - 2\eta\sigma^2)(w_0 - \frac{y}{\sigma})$

We need $|1 - 2\eta\sigma^2| < 1$, $\therefore \eta < \frac{1}{\sigma^2}$

If we want: $|w_t - w^*| < \varepsilon |w^*|$, and assume $w_0 = 0$

Then $|(1 - 2\eta\sigma^2)^t| < \varepsilon$ $\quad t > \frac{\ln \varepsilon}{\ln(|1 - 2\eta\sigma^2|)}$

The above is scalar problem, what if: $\begin{bmatrix} \sigma_1 & \\ & \sigma_2 \end{bmatrix} \begin{bmatrix} w[1] \\ w[2] \end{bmatrix} = \begin{bmatrix} y[1] \\ y[2] \end{bmatrix}$

(And assume: $\sigma_1 > \sigma_2$)

Assume this is $\Sigma \vec{w} = \vec{y}$, then $L(w) = \|y - \Sigma w\|_2^2$, GD:

$\vec{w_{t+1}} = (I - 2\eta\Sigma^2)\vec{w_t} + 2\eta\Sigma\vec{y}$, similar to convergence analysis above:

$\begin{cases} |1 - 2\eta\sigma_1^2| < 1 \\ |1 - 2\eta\sigma_2^2| < 1 \end{cases}$ $\therefore \eta < \min\{\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}\} = \frac{1}{\sigma_1^2}$

If we want fastest convergence speed? $|1 - 2\eta\sigma_1^2| = |1 - 2\eta\sigma_2^2|$

$\eta' = \frac{1}{\sigma_1^2 + \sigma_2^2}$

If $\Sigma \in \mathbb{R}^{k \times k}$, diagnal ($\sigma_1 \sigma_2 \dots \sigma_k$), then $\eta < \min\{\frac{1}{\sigma_k^2}\}$,

$\eta' = \frac{1}{\sigma_1^2 + \sigma_2^2}$? Not exactly, hard to tell

Finally, for trivial $Xw = y$, we can SVD: $U \Sigma V^T w = y$ (orthogonal, i.e., transpose is inverse)

We can: $\Sigma(V^T w) = U^T y = \tilde{y}$, let $V^T w = \hat{w}$, i.e., $w = V\hat{w}$

$\therefore \Sigma \hat{w} = \tilde{y}$, and it connect to the previous analysis

KOKUYO

No.

Date

Topic2 : Why SGD can work ?

there are solutions

Suppose : $Xw = y$, $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$, and X has full rank, $d > n$, $x_i \in \mathbb{R}^d$

Assumn $w_0 = 0$, we know $w^* = X^T(XX^T)^{-1}y$, rewrite $w' = w - w^*$

$\therefore$ Goal is $Xw' = 0$, $w_0' = -w^*$

Let's Use SVD : $\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \vec{0} = [\Sigma_1, \vec{0}]$, then :

$Xw' = U\Sigma V^T w' = U[\Sigma_1 \ \vec{0}_{n\times(d-n)}] V^T w' = [\tilde{X}, \vec{0}] V^T w' = [\tilde{X}, \vec{0}] w''$

$\therefore w_0'' = -V^T X^T (XX^T)^{-1} y = \dots = -\begin{bmatrix} \Sigma_1^{-1} U^T y \\ \vec{0}_{d-n} \end{bmatrix} \cdot = \begin{bmatrix} a \\ \vec{0}_{d-n} \end{bmatrix}$ ← all zeros

system

Now we know : $[\tilde{X}, \vec{0}] w'' = 0$ is Goal. and $(d-n)$ colsk rows will not work in this

$\therefore \tilde{X}$ is : $U\Sigma_1$, $\tilde{X}$ is also $[\tilde{x}_i^T]_{i\in[1,n]}$, we have: $\tilde{x}_i^T \tilde{w} = 0$

Solving $\tilde{X}\tilde{w} = 0$ is exactly calculating the original problem, since $\tilde{X}\tilde{w}$ has linear relationship with X and w. Now: $\tilde{X}\tilde{w} = 0$, $\mathcal{L}(\tilde{w}) = (\tilde{X}\tilde{w})^T \tilde{X}\tilde{w}$

$\therefore$ SGD : $\tilde{w}_{t+1} = \tilde{w}_t - 2\eta \tilde{x}_{I_t} \tilde{x}_{I_t}^T \tilde{w}_t$, $I_t \in [1, n]$, one sample

Try to decompose : $\mathcal{L}(\tilde{w}_{t+1}) = \mathcal{L}(\tilde{w}_t) + A + B$. , $\mathcal{L}(\tilde{w}) = (\tilde{X}\tilde{w})^T \tilde{X}\tilde{w}$

Bring in, and $\begin{cases} A = 2\tilde{w}_t^T \tilde{X}^T \tilde{X} (\tilde{w}_{t+1} - \tilde{w}_t) \\ B = (\tilde{w}_{t+1} - \tilde{w}_t)^T \tilde{X}^T \tilde{X} (\tilde{w}_{t+1} - \tilde{w}_t) \end{cases}$ A. B are in form of $(\tilde{w}_{t+1} - \tilde{w}_t)$ & $\tilde{w}_t$

Insight : $E(A|\tilde{w}_t)$, bring in A, $\tilde{w}_{t+1} = \tilde{w}_t - 2\eta \tilde{x}_{I_t} \tilde{x}_{I_t}^T \tilde{w}_t$

$E(A|\tilde{w}_t) = 2\tilde{w}_t^T \tilde{X}^T \tilde{X} E[-2\eta \tilde{x}_{I_t} x_{I_t}^T \tilde{w}_t | \tilde{w}_t] = -\frac{4}{n}\eta \tilde{w}_t^T \tilde{X}^T \tilde{X} \tilde{X}^T \tilde{X} \tilde{w}_t$

$\leq -4/n \ \eta \sigma_{min}^2 \tilde{w}_t^T \tilde{X}^T \tilde{X} \tilde{w}_t = -\frac{4}{n}\eta \sigma_{min}^2 \mathcal{L}(\tilde{w}_t) = -C_1 \eta \mathcal{L}(\tilde{w}_t)$, $C_1 > 0$

Similarly $E(B|\tilde{w}_t) \leq \frac{4}{n}\eta^2 \sigma_{max}^2 \rho^2 \mathcal{L}(\tilde{w}_t) = C_2 \eta^2 \mathcal{L}(\tilde{w}_t)$, $C_2 > 0$

↳ largest norm of $\tilde{x}_i$

$\therefore E(\mathcal{L}(\tilde{w}_{t+1})|\tilde{w}_t) = E(\mathcal{L}(\tilde{w}_t)) + E(A|\tilde{w}_t) + E(B|\tilde{w}_t)$

$\leq (1 - C_1\eta + C_2\eta^2)\mathcal{L}(\tilde{w}_t)$

So, we can set η s.t. $1 - C_1\eta + C_2\eta^2 < 1$ if $\eta < \frac{C_1}{C_2}$

To achieve fastest convergence, $\eta^* = \frac{C_1}{2C_2}$

No.

Date

Topic 3. Relationship between Noise and Regulation

Consider $(x_i, y_i)_{i=1}^m$, $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, assume we add noise:

$\tilde{x}_i = x_i + N_i$, $N_i \sim \mathcal{N}(\vec{0}, \sigma^2 I_n)$

For noisy data matrix, we want $\arg\min_w \mathbb{E}[\|\tilde{X}w - y\|_2^2]$

$\mathbb{E}[\|\tilde{X}w - y\|_2^2\|] = \mathbb{E}[\sum_{i=1}^m [(x_i + N_i)^T w - y_i]^2]$ $\quad \mathbb{E}: \sigma^2 I_n$.

$= \sum_{i=1}^m \mathbb{E}[(x_i^T w - y_i)^2] - 2\mathbb{E}[(N_i^T w)](x_i^T w - y_i) + \mathbb{E}[w^T N_i N_i^T w]$

$= \|Xw - y\|_2^2 + m\sigma^2 \|w\|_2^2$

If in SGD t-th step, we have N_t to x_i ← sample in t-th step, then:

$\frac{\partial L}{\partial w} = (w\tilde{x} - y)\tilde{x} = w(x^2 + 2xN_t + N_t^2) - y(x + N_t)$

$W_{t+1} = W_t - \eta \nabla_w L(W_t) = W_t(1 - \eta(x^2 + 2xN_t + N_t^2)) + \eta y(x + N_t)$

$\mathbb{E}(W_{t+1}) = \mathbb{E}(W_t)(1 - \eta(x^2 + \sigma^2)) + \eta y x$

* $\mathrm{Var}(N_t) = \mathbb{E}(N_t^2) - [\mathbb{E}(N_t)]^2 = \mathbb{E}(N_t^2) = \sigma^2$

$\therefore \eta$ should satisfy: $|1 - \eta(x^2 + \sigma^2)| < 1$, $0 < \eta < \frac{2}{x^2 + \sigma^2}$

Topic 4. Another perspective of: $(X^T X + \lambda I)^{-1} X^T y = X^T (XX^T + \lambda I)^{-1} y$

We want build r.v. Z, s.t., $Z = W - AY$, but Z, Y are not correlated:

$\mathrm{Cov}(Z, Y) = \cdots = \mathbb{E}[(W - \mu_W)(Y - \mu_Y)^T] - A \cdot \mathbb{E}[(Y - \mu_Y)(Y - \mu_Y)^T]$

$= \Sigma_{WY} - A \cdot \Sigma_{YY}$ $\quad \therefore A = \Sigma_{WY} \cdot \Sigma_{YY}^{-1}$

$\therefore W = Z + AY$, $\mathbb{E}[W|Y=y] = \mathbb{E}[Z|Y=y]^* + Ay$, bring in $Z = W - Ay$

$= \cdots = \mu_W + \Sigma_{WY}\Sigma_{YY}^{-1}(y - \mu_y)$ $\quad Z, Y$: $\sim \mathcal{N}$, so uncorrelation

Assume: $\mu_W = \mu_y = 0$, i.e., W, y has zero mean ⇒ independent

$\therefore \mathbb{E}[W|Y=y] = \Sigma_{WY}\Sigma_{YY}^{-1} y$, assume *) $Y = XW + \sqrt{\lambda}N$. then $W \sim \mathcal{N}(0, I)$

$\Sigma_{WY} = \mathbb{E}(W(XW + \sqrt{\lambda}N)^T] = X^T$

$\Sigma_{YY} = \mathbb{E}((XW + N)(XW + N)^T] = XX^T + \lambda I$

$\therefore \hat{W} \overset{*}{=} \mathbb{E}(W|Y=y) = X^T(XX^T + \lambda I)^{-1} y$

Constrained: As in ridge regression, $\|W\|_2^2$

* MAP version to explain ridge: prior $W \sim \mathcal{N}(0, I)$, view random Y as $Y = X^T W + \sqrt{\lambda}N$, N is i.i.d. $\sim \mathcal{N}(0,1)$ I should refer to HW0 整理

* Why mean? A normal r.v. has its density maximized at its mean.

KOKUYO

No.

Date

HW0. HW2 整理

Topic 1. MAP perspective of Ridge Regression. We can think $W \sim N(0,1)$, and view Y as being generated using $Y = x^T W + \sqrt{\lambda} N$, where $N \sim N(0,1)$

Vector version: $Y = XW + \sqrt{\lambda} N$

$$\therefore MAP(w|Y=y) = \arg\max_w f(w|Y=y) = \arg\max_w \frac{f(w,y)}{f(y)}$$

$$\Rightarrow \arg\max_w f(w) \cdot f(y|w) = \arg\max_w f(w) \cdot \prod_{i=1}^{n} f(y_i|w)$$

For $N(0,1)$, PDF: $f_Z(z) = \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}$ $\therefore \arg\max_w \frac{e^{-\frac{\|w\|_2^2}{2}}}{\sqrt{2\pi}} \prod_{i=1}^{n} \frac{e^{-\left(\frac{y_i - x_i^T w}{\sqrt{\lambda}}\right)^2/2}}{\sqrt{2\pi}}$

$$\overset{*}{=} \arg\min_w \|w\|_2^2 + \frac{1}{\lambda}\|y - Xw\|_2^2 \equiv \arg\min_w \lambda\|w\|_2^2 + \|Xw - y\|_2^2$$

*: negation, argmax → argmin, take log

Topic 2. Originally in optimizer: $\arg\min_{\|\Delta w\| < \eta} \langle \nabla_w \mathcal{L}(w), \Delta w \rangle$

We set constraint as: $\|\Delta w\| < \eta$

What if we take it as an optimization problem term?

$\Rightarrow u = \arg\min_{\Delta\theta} g^T \Delta\theta + \frac{1}{\alpha} d(\Delta\theta)$, where $g = \nabla f(\theta)$, d is $\mathbb{R}^{\dim(\theta)} \to \mathbb{R}^+$

Then: $\nabla_{\Delta\theta} u = g + \frac{2}{\alpha}\Delta\theta = 0$ if d is the euclidean distance (L_2-Norm)

$\therefore \Delta\theta^* = -\frac{\alpha}{2} g$

let $M = \max_i |\Delta\theta_i|$

But if d is ∞ norm? $u = \arg\min_{\Delta\theta} g^T\Delta\theta + \frac{1}{\alpha}\max|\Delta\theta_i|$

$\therefore u = \arg\min_{\Delta\theta, M} g^T\Delta\theta + \frac{1}{\alpha}M^2$, $|\Delta\theta_i| \le M$

Since $|\Delta\theta_i| \le M$, we want $\sum_{i=1}^{\dim(\Delta\theta)} g_i^T \Delta\theta_i$ as negatively small as possible, $|\Delta\theta_i| = M$ for $\forall i$

$\therefore \Delta\theta_i = -M \cdot \text{sign}(g_i)$, $u = -M\|g\|_1 + \frac{1}{\alpha}M^2$

$\Rightarrow \nabla_M u = -\|g\|_1 + \frac{2}{\alpha}M = 0$, $M = \frac{\alpha}{2}\|g\|_1$

$\therefore u = -\frac{\alpha}{2}\|g\|_1 \cdot \text{sign}(g)$, $\Delta\theta^* = \{\Delta\theta_i = -M\,\text{sign}(g_i)\}_{i=1}^{\dim(\Delta\theta_i)}$

Date

HW3 整理

Topic 1. Maximal Update Parameterization

(a). Consider dense layer $x \in \mathbb{R}^{d_1}$, $W \in \mathbb{R}^{d_2 \times d_1}$, W is initialized with i.i.d standard Gaussian entries, and entries of X are sampled i.i.d from unit Gaussian.

Then: $y = Wx$, its squared RMS norm? consider $W_i \in \mathbb{R}^{1 \times d_1}$, $y_i = W_i x$

Then $y_i = \sum_{j=1}^{d_1} w_{ij} x_j$, $w_{ij}, x_j \sim \mathcal{N}(0,1)$.

Regard $\{w_{ij} x_j\}$, $\mathrm{Var}(w_{ij} x_j) = \mathrm{Var}(z) = E(z^2) - [E(z)]^2$

$E(z) = E(w_{ij}) E(x_j)$ (since w_{ij} and x_j is individually independent)

$E(z^2) = E(w_{ij}^2 x_j^2) = E(w_{ij}^2) E(x_j^2) = 1 \cdot 1 = 1$

$\therefore \mathrm{Var}(z) = 1 \Rightarrow \mathrm{Var}(y_i) = d_1 \cdot \mathrm{Var}(z) \overset{*}{=} d_1$

* If $S = \sum_{i=1}^{n} X_n$, then $\mathrm{Var}(S) = \mathrm{Var}(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathrm{Var}(X_i)$ if X_n i.i.d

$\therefore \|y\|_{RMS}^2 = \frac{1}{d_1} \sum_{i=1}^{d_1} y_i^2 = \frac{1}{d_1} \sum_{i=1}^{d_1} [y_i - \underbrace{E(y_i)}_{=0}]^2 = \frac{1}{d_1} \cdot d_1 \cdot \mathrm{Var}(y_i) = d_1$

$\therefore E(\|y\|_{RMS}^2) = O(d_1) \|x\|_{RMS}^2$, so W should be $\frac{1}{\sqrt{d_1}} W_{previous}$

△ $\mathrm{Var}(z) = E(x_j^2) E(w_{ij}^2) = 1/d_1$, $\mathrm{Var}(w_{ij}) = \frac{1}{d_1}$, $w_{ij} \overset{\text{☆}}{\sim} \mathcal{N}(0, 1/\sqrt{d_1})$

☆ If $S = tX \Rightarrow \mathrm{Var}(S) = t^2 \mathrm{Var}(X)$.

→ These two seems to contradict, but they are both true

(b). Consider using RMS norm in weight update: $W_{t+1} \leftarrow W_t + \eta\, \mathrm{sign}(g_i x_i^T)$

$y' = \mathrm{sign}(g_i x_i^T) x_i$, what is $E(\|y\|_{RMS}^2)$? $g_i \in \mathbb{R}^{d_2}$, $x_i \in \mathbb{R}^{d_1}$

↑ Δy, cauze $\Delta y = \Delta W x_i$

Dimension analysis: $\mathrm{sign}(g_i x_i^T) \in \mathbb{R}^{d_2 \times d_1}$, $x_i \in \mathbb{R}^{d_1}$, $y' \in \mathbb{R}^{d_2}$

$\therefore y'_j = \eta\, \mathrm{sign}(g_{ij} x) x$, $E(y_j'^2) = E\left(\eta^2 \left[\sum_{i=1}^{din} \mathrm{sign}(g_{ij} x_i) x_i\right]^2\right) \overset{*}{=} \eta^2 \left[E\left(\sum_{i=1}^{din} |x_i|\right)\right]^2$

$\therefore E(\|y'\|_{RMS}^2) = \frac{1}{d_2} \sum_{j=1}^{d_2} \underbrace{E(y_j'^2)}_{\text{i.i.d}} = \frac{1}{d_2} \cdot \underbrace{d_2}_{\text{i.i.d}} \cdot \eta^2 \cdot \left[E\left(\sum_{i=1}^{din} |x_i|\right)\right]^2 = \eta^2 din^2$

$\therefore$ For Δy, y', scale is din !! Hence need to multiply Δy by: $\frac{1}{din}$

~~* sign will force all x_i to face toward the same +/- direction with g_{ij},~~

and square will erase +/- $\Rightarrow$ 1 !!

KOKUYO

No.

Date

Topic 2: Maximal Update Parameterization Research

Consider $\Delta W_l \in \mathbb{R}^{n_l \times n_{l-1}}$, $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$, if:

$$\|W_l\|_2 = \Theta\left(\sqrt{\frac{n_l}{n_{l-1}}}\right) \quad \|\Delta W_l\|_2 = \Theta\left(\sqrt{\frac{n_l}{n_{l-1}}}\right)$$

then, suppose $h_l(x) \in \mathbb{R}^{n_l}$ as the feature of input x at layer l, $\Delta h_l(x) \in \mathbb{R}^{n_l}$ denote the change, then:

$$\|h_l\|_2 = \Theta(\sqrt{n_l}) \quad \|\Delta h_l\|_2 = \Theta(\sqrt{n_l}), \text{ i.e., their RMS norm is } \Theta(1)$$

$\Rightarrow$ Each entry of hidden vectors moves and remains by $\Theta(1)$

Proof: If condition is satisfied: then $\|W\|_{RMS \to RMS} =^{*} \Theta(1)$.

*: $\|W_l\|_2 = \sqrt{\frac{n_l}{n_{l-1}}}$ $\|W\|_{RMS \to RMS} = \Theta\left(\sqrt{\frac{n_l}{n_{l-1}}}\right)$

$$\therefore \|h_l(x)\|_2 = \|W_l h_{l-1}(x)\|_2 \le \|W_l\|_2 \|h_{l-1}(x)\|_2 = \Theta\left(\sqrt{\frac{n_l}{n_{l-1}}}\right)\Theta(\sqrt{n_{l-1}}) = \Theta(\sqrt{n_l})$$

$\Rightarrow$ if $h_0(x)$ is initialized properly, above induction holds true

$$\|\Delta h_l(x)\|_2 = \|W_l \Delta h_{l-1}(x) + \Delta W_l h_{l-1}(x) + \Delta W_l \Delta h_{l-1}(x)\|_2$$

$$\le \Theta\left(\sqrt{\frac{n_l}{n_{l-1}}}\right)\Theta(\sqrt{n_{l-1}}) + \Theta\left(\sqrt{\frac{n_l}{n_{l-1}}}\right)\Theta(\sqrt{n_{l-1}}) + \Theta\left(\sqrt{\frac{n_l}{n_{l-1}}}\right)\Theta(\sqrt{n_{l-1}}) = \Theta(\sqrt{n_l})$$

Topic 3: Policy Gradient & Reparameterization Gradient Estimator.

Consider $F(\theta) = \mathbb{E}_{x \sim p_\theta}[f(x)]$, its derivative to θ is a challenge!

($x \sim p_\theta$: how x is sampled. f: sort 'eval' func)

Suppose x is k-D multivariate Gaussian r.v., $\theta = \mu$, then.

$$p_\mu(x) = (2\pi\sigma)^{-n/2} \exp\left(-\|x-\mu\|_2^2/(2\sigma^2)\right)$$

$$\nabla_\mu F(\mu) = \nabla_\mu \int_x (2\pi\sigma)^{-n/2} \exp\left(-\|x-\mu\|_2^2/(2\sigma^2)\right) f(x)\,dx$$

$$= \int_x (2\pi\sigma)^{-n/2} \exp\left(-\|x-\mu\|_2^2/(2\sigma^2)\right) \frac{x-\mu}{\sigma^2} f(x)\,dx = \mathbb{E}_{x \sim p_\mu(x)}\left[\frac{x-\mu}{\sigma^2} f(x)\right]$$

For any p_θ: $\nabla_\theta F(\theta) = \int_x \nabla_\theta p_\theta(x) f(x)\,dx = \int_x p_\theta(x) \frac{\nabla_\theta p_\theta(x)}{p_\theta(x)} f(x)\,dx$ ($\nabla_\theta \log p_\theta(x)$)

Else, decouple: $x = g(z, \theta)$ (distribution unaffected by θ) $= \mathbb{E}_{x \sim p_\theta}[\nabla_\theta \log p_\theta(x) f(x)]$

Then $\nabla_\theta F(\theta) = \mathbb{E}_{z \sim p(z)}[\nabla_\theta g(z,\theta)^T \nabla_x f|_{x=g(z,\theta)}]$ (Chain Rule).

Date

HW4 整理

Topic 1: Newton-Schulz Runtime

In muon, we use NS to Find:

$G_t = U\Sigma V^T$, $P(G_t) = U P(\Sigma) V^T$, $[P^n(G_t) = U P^n(\Sigma) V^T = UV^T]$

to replace finding SVD of G_t, so as to save time!

(Recall in Muon: $W_{t+1} = W_t - \eta UV^T$, i.e., discard Σ)

How can NS save runtime? Consider:

$$p(W) = \frac{1}{2}(3I_{dout} - WW^T)W$$

Assume 1 operation takes one unit of time. Then:

WW^T: $W \in \mathbb{R}^{d_{out} \times d_{in}}$, takes $2d_{out}^2 d_{in}$ *

*: For $A \times B$, $A \in \mathbb{R}^{p \times m}$, $B \in \mathbb{R}^{m \times n}$, time: $2mpn$

$p \times n$ elements, each need m multiplication & $m-1 \approx m$ summation

$(3I_{dout} - WW^T)^T$ takes $2cd_{out}^2 d_{in}$, c is a constant

$\therefore$ Total: $\Theta(d_{out}^2 d_{in})$.

But, if $d_{out} >> d_{in}$, can we be faster?

re-write: $p(W) = \frac{1}{2}(3I_{dout} - WW^T)W = \frac{1}{2}W(3I_{din} - W^TW)$

Then total: $\Theta(d_{in} d_{out})$.

Topic 2. MuP at unit scale.

If wanting to train on fp8, its number of representable values is small.

So for Xavier Init, if $\sim N(0, 1/d)$, can be a bug: rounding issue.

So it makes sense to: $\sim N(0,1)$

However, to avoid exploding, we can assign a constant float scalar:

$$y = cWx$$

So we can set $c = 1/\sqrt{d_{in}}$ to set variace $\frac{1}{d_{in}}$ scaled.

Now. $\Delta y = c\Delta W x$, if $\|x\|_{RMS} = 1$, and we want $\|y\|_{RMS} \le 1$, then:

$$\|\Delta y\|_{RMS} \le \frac{1}{\sqrt{d_{in}}} \cdot \frac{\sqrt{d_{in}}}{\sqrt{d_{out}}} \|\Delta W\|_2 \cdot 1 \le 1 \quad \therefore \|\Delta W\|_2 \le \sqrt{d_{out}}$$

*: $\|A \cdot B\| \le \|A\| \|B\|$, $\|A\|_{RMS} = \sqrt{\frac{d_{in}}{d_{out}}} \|A\|_2$, $A \in \mathbb{R}^{d_{out} \times d_{in}}$

We want to abide by this constraint!

If using SignSGD: $\Delta W = \alpha \cdot sign(\nabla_W \mathcal{L})$

So $\|\Delta W\|_2 \le \alpha \|sign(g)\|_1 \|sign(x)^T\|_2 = \alpha \sqrt{d_{in} \times d_{out}} = \sqrt{d_{out}} \therefore \alpha = \frac{1}{\sqrt{d_{in}}}$

where $g = \nabla_y \mathcal{L}(y)$ $(y = Wx + b)$.

KOKUYO

Date

But if we use Muon-style method:

Consider: $U, \Sigma, V^T = SVD(\nabla_W \mathcal{L})$, $\Delta W = \alpha UV^T$, we use NS to get UV^T

Then $\|W\|_2 \le \alpha \|U\|_2 \|V^T\|_2 = \alpha \le \sqrt{d_{out}} \Rightarrow \alpha \le \sqrt{d_{out}}$

✡ SignGD, Adam and Muon: global scale of raw gradients doesn't effect final update direction.

Topic3 Feature Dimensions of CNN

Suppose input: $W \times H \times C$, Kernel size, padding, number and stride are K, P, S, F. Then:

# of weights: FK^2C # of bias: F

$$W' = \left\lfloor \frac{W-K+2P}{S} \right\rfloor + 1, \quad H' = \left\lfloor \frac{H-K+2P}{S} \right\rfloor + 1, \quad C' = F$$

And a max pooling layer has stride S and size K. Then:

$$W' = \frac{W-K}{S} + 1, \quad H' = \frac{H-K}{S} + 1, \quad C' = C$$

Suppose L layers with kernel size of K, and stride is 1. The final outputs receptive field size is $L \times (K-1) + 1 = O(LK)$

Topic4: Weights and Gradients in a CNN

Input: $X = [x_{i,j}]$, $i,j \in [1,n]$; $W = [w_{i,j}]$, $i,j \in [1,k]$; Output $Y = [y_{i,j}]$, $i,j \in [1,m]$

Then: $y_{i,j} = \sum_{h=1}^{k} \sum_{l=1}^{k} x_{i+h-1, j+l-1} w_{h,l}$ (assume no padding, stride is 1, channel is 1)

Denote as $Y = X * w$. We have $\mathcal{L}$ and $dY = [dy_{i,j}]$, where:

$dy_{i,j} = \frac{\partial \mathcal{L}}{\partial y_{i,j}}$ So we're curious about dW.

We have $\frac{\partial \mathcal{L}}{\partial w_{h,l}} = \sum_{i,j} \frac{\partial \mathcal{L}}{\partial y_{i,j}} \cdot \frac{\partial y_{i,j}}{\partial w_{h,l}}$, $\frac{\partial y_{i,j}}{\partial w_{h,l}} = x_{i+h-1, j+l-1}$, $i,j \in [1,m]$

$\therefore dw_{h,l} = \sum_{i=1}^{m} \sum_{j=1}^{m} x_{i+h-1, j+l-1} dy_{i,j} \Rightarrow dW = X * dY$

Suppose $x_{i,j}, dy_{i,j}$ are r.v. with 0 mean, σ_x^2 σ_y^2 variance. then:

✡ $\mathbb{E}(dw_{i,j}) = 0$, $Var(dw_{i,j}) = m^2 \sigma_x^2 \sigma_y^2$, where $m = n-k+1$

$\therefore Std(dw_{i,j}) = \Theta(n)$. Finally, for pooling:

If max: only one of input pixel has gradient of magnitude, and the rest are zeros

If average: distribute gradient evenly across each input blocks

No.

Date

HW5 整理

Topic 1. BN.

For x_i: $\frac{\partial L}{\partial x_i} = \sum_{j=1}^{n} \frac{\partial L}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_i}$, and $y_i = \gamma(x_i - \frac{1}{n}\sum_{j=1}^{m} x_j) + \beta$

$\therefore \frac{\partial y_j}{\partial x_i} = \begin{cases} \gamma(1-\frac{1}{n}), & \text{if } i=j \\ \gamma(-\frac{1}{n}), & \text{if } i \neq j \end{cases}$ $\therefore \frac{\partial L}{\partial x_i} = \left[\sum_{i=1}^{n} -\frac{\gamma}{n}\frac{\partial L}{\partial y_i}\right] + \left[\gamma \frac{\partial L}{\partial y_i}\right]$

$= \gamma\left[\frac{\partial L}{\partial y_i} - \frac{1}{n}\sum_{j=1}^{n} \frac{\partial L}{\partial y_i}\right]$

*: CNN: BN is applied to every value in a channel & in a batch

i.e., average over : $BS \times H \times W$ entries and scale & bias

Topic 2: Regularization and Dropout

Recall: $L(w) = \|y - Wx\|_2^2$. One way of dropout during SGD on $X \in \mathbb{R}^d$: keep each feature $\sim$ i.i.d Bernoulli(p), and zero out if not kept.

It turns out, our goal now is:

$L(\check{w}) = \mathbb{E}_{R\sim \text{Bernoulli}(p)}[\|y - (R \odot X)\check{w}\|_2^2]$, while $R_{i,j} \sim$ i.i.d Bernoulli(p) and $\odot$ is the element-wise product.

Consider Generalized Ridge-Regression involving:

$L(w) = \|y - Xw\|_2^2 + \|\Gamma w\|_2^2$.

It turns out *we can rewrite and get: $L(\hat{w}) = \|y - pX\check{w}\|_2^2 + p(1-p)\|\check{\Gamma}\check{w}\|_2^2$ with $\check{\Gamma}$ being a diagonal matrix whose j-th diagonal entry [is the norm of the j-th column of the X.]

* Long Proof and we ignore

Then we can tell: $\Gamma = \sqrt{\frac{1-p}{p}}\check{\Gamma}$. If we want to finally write to:

$L(\tilde{w}) = \|y - \tilde{X}\tilde{w}\|_2^2 + \lambda\|\tilde{w}\|_2^2$

Then: $w = \Gamma^{-1}\tilde{w}$, $\tilde{X} = X\Gamma^{-1}$, with the definition of $\Gamma = \sqrt{\frac{1-p}{p}}\check{\Gamma}$

we conclude that: dropout is actually BN (even include rescale!).

HW6 整理

Topic1: Connection Between graph dynamics & GNN.

Let's assume: ①For a network, underlying graph is adjacency matrix A.

② Has n vertices in each layer, corresponding to the n vertices of underlying graph

③ Each vertex has n channels

④ The input to each node in the 0-th layer is a one-hot encoding of own identity. That is, the node i in the graph has input $(0, \cdots, \underset{\uparrow}{1}, 0, \cdots)$ i-th entry

⑤ Weight connecting node i in layer k to node j in layer k+1: $A_{i,j}$

⑥ At each layer, operation at each node: weighted sum up

(a): j-th node at layer k: A^k_j. Since $G^{t+1} = AG^t$, $G^0 = I$

(b): So the i-th output of j-th node at layer k, equals to the number of path from i to j who total distance is k.

(c): In this simple setting, update for v_j is: $v_j = \sum_{i \in \underset{\uparrow}{V_{(j)}}} v_i$ j's neighbor

(d): If replace 'sum' func with 'max'. Then the ith output of node j at layer k represents: if there's a path

(e). Analogy of CNN to GNN: (CNN vs GNN).

Semantic Segmentation ↔ Node-level prediction

Color Jitter ↔ Jittering node value

Dropout ↔ In GNN layers, set some node or edge to 0

Blur ↔ Average with Neighbors ResNet ↔ $v_i^{t+1} = v_i^t + \text{update}(v_i^t)$

Topic 2: GNN

For GNN node update, a function which satisfies permutation invariance is valid. For loss, we usually use cross-entropy loss:

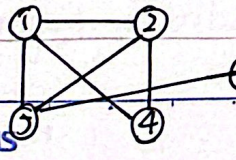
$$\frac{1}{n} \sum_x \left(y(x) \log \frac{1}{\hat{y}(x)} + (1 - y(x)) \log \left(\frac{1}{1 - \hat{y}(x)} \right) \right)$$

$y(x) = 1$ if true else 0; $\hat{y}(x)$ is the prob models predict that $y(x)$ is 1

Example Update: $S_i^l = S_i^{(l-1)} + W_1 \frac{\sum_{j=1}^{n_i} \tanh(W_2 m_{i,j}^{(l-1)})}{n_i}$

This is permutation invariant and using connection of residual.

If $s \in \mathbb{R}^d$, W_2 k rows, then $W_1 \in \mathbb{R}^{d \times k}$ $+ \tanh(W_2 S_5^{(l-1)})]$



For graph like left

e.g: $S_2^l = S_2^{(l-1)} + \frac{W_1}{3}[\tanh(W_2 S_1^{(l-1)}) + \tanh(W_2 S_4^{(l-1)})$

$S_3^l = S_3^{(l-1)} + \frac{W_1}{1}(\tanh(W_2 S_5^{(l-1)}))$.

Date

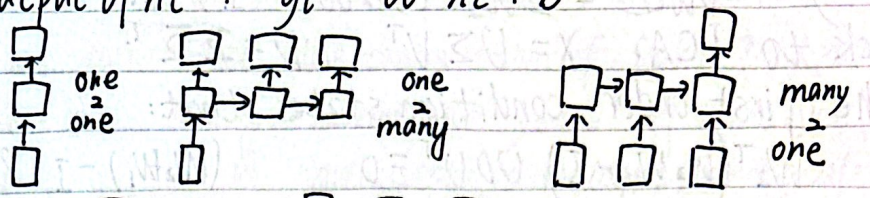
HW7 整理 & Discussion 8 整理

Topic 1: Implementing RNN

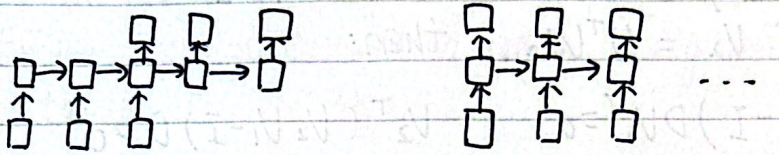
In RNN, the update of $h_t \to h_{t+1}$: $h_t = \sigma(W^h h_{t-1} + W^x x_t + b)$

and the output of h_t : $\hat{y}_t = W^T h_t + b^T$

RNN can be used for many prediction:



many 2 many:



Consider a many 2 one at timestep T. Then:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial h_T} \cdot \frac{\partial h_T}{\partial W} + \frac{\partial L}{\partial h_{T-1}} \cdot \frac{\partial h_{T-1}}{\partial W} + \cdots + \frac{\partial L}{\partial h_1} \cdot \frac{\partial h_1}{\partial W}$$

The later terms in the sum often end up small or large → Vanishing & Exploding

If nonlinearity is not considered and use MSE:

$\frac{\partial L}{\partial h_t} = 2(\hat{y} - y) W^T (W^h)^{T-t}$, so if the magnitude of the largest eigenvalue of W^h is much larger than 1, gradient explode; vice versa.

So if $t \ll k$, $\partial L_k / \partial h_t$ will be very small ⇒ hard to learn long-range dependency

Topic 2: PCA & AutoEncoders

Assume data $x_i \in \mathbb{R}^m$ and encoder $W_1 \in \mathbb{R}^{k \times m}$ ($k < m$) & decoder $W_2 \in \mathbb{R}^{m \times k}$

We have dataset $X \in \mathbb{R}^{m \times n}$, the Loss is:

$$L(W_1, W_2; X) = \|X - W_2 W_1 X\|_F$$

We assume $\sigma_1^2 > \cdots > \sigma_k^2 > 0$ are top-k eigenvalues of XX^T

So set $S = \mathrm{diag}(\sigma_1, \cdots, \sigma_k)$ with corresponding eigenvectors are the columns of $U_k \in \mathbb{R}^{m \times k}$. Then, first want to find out first order optimality conditions that L satisfy.

$\|A\|_F^2 = tr(AA^T)$, $tr(ABC) = tr(BCA) = tr(CAB)$, $tr(A) = tr(A^T)$

$\nabla_A tr(AB) = B^T$ $\quad$ $\nabla_B tr(AB) = A^T$

$\therefore \nabla_A tr(ABBA) = \nabla_A tr(ABBC)|_{C=A} + \nabla_A tr(CBBA)|_{C=A}$

$= (BBA)^T + (ABB)^T$

KOKUYO

No.

Date

$\therefore \nabla_{W_2} \mathcal{L} = \nabla_{W_2} tr(XX^T + W_2W_1XX^TW_1^TW_2^T - 2W_2W_1XX^T)$

$= 2(W_1XX^TW_1^TX_2^T)^T - 2(W_1XX^T)^T$

$= 2(W_2W_1 - I)\, XX^TW_1^T$

Similarly: $\nabla_W \mathcal{L} = 2W_2^T(W_2W_1 - I)XX^T$

Back to PCA: $X = U\Sigma V^T$, $D = \frac{1}{n}\Sigma\Sigma^T$.

The first order condition states that:

$W_2^T(W_2W_1 - I)UDU^T = 0 \qquad (W_2W_1 - I)UDU^TW_1^T = 0$

Let $V_1 = W_1U$, $V_2 = U^TW_2$, then:

$(V_2V_1 - I)DV_1^T = 0 \qquad V_2^T(V_2V_1 - I)D = 0$

If $W_2 = W_1^T$ are the first k columns of U, then

$V_1 = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & 0 \end{bmatrix}$ $V_2 = V_1^T \Rightarrow W_1 = U_k^T$, $W_2 = U_k$

Topic3: SSM as kernel

Consider $y_t = \sum_{i=0}^{T} K_{t-i}\, u_i$. Then: $x_1 = Bu_0$, $x_2 = ABu_0 + Bu_1$

$x_3 = A^2Bu_0 + ABu_1 + Bu_2 \cdots \qquad y_i = Cx_i$

$\therefore K = (CB,\ CAB,\ CA^2B,\ \cdots)$.

Topic4: AutoEncoder

① If train two different autoencoder on the same dataset, the one which produces lower validation loss not necessarily perform better on downstreaming task! representation.

E.g.: Add noise or bottleneck smaller, loss ↑, but can produce more useful

② Using representations by a learned autoencoder is most useful when only having a few data samples for downstream task. (Can also be useful if plenty, of course)

③ For training autoencoder with noise or masking, when inferencing for the downstreaming task, no augmentation for input is needed. (Or: uncommon!)

Topic5: $\mathcal{L}_\lambda(W_1, W_2; X) = \frac{1}{n}\|X - W_2W_1X\|_F^2 + \lambda\|W_1\|_F^2 + \lambda\|W_2\|_F^2$

represent inductive bias of: Forcing W_1 W_2 to be orthogonal.

Think of $W_1 = U_1\Sigma_1V_1^T$ $W_2 = U_2\Sigma_2V_2^T$. W_1W_2 should be 'pseudo inverse'

Regularization: $\lambda\sum_{i=1}^{k}(\sigma_i^2 + \frac{1}{\sigma_i^2}) \Rightarrow$ forcing $\sigma_i \approx 1$

$\Rightarrow W_1W_1^T = I \quad W_2W_2^T = I$

Campus

Date

HW8 整理

Topic 1: SSM as convolution kernel.

Previously, we've known: $\{y\} = K * \{u\}$, i.e., $y_k = \sum_{l=0}^{L} K_l u_{k-l}$
($k \in [1, L]$, any $u \le 0$ is set to 0) and $K = (CB, CAB, CA^2B, \cdots)$

Then: how much can we parallelize the computation of $\{y_k\}$? ($u_k \in \mathbb{R}^n$)

If unroll in a recursive way, we have to compute $y_{L-1}, \cdots$ if want y_L.
then critical path is of $O(L)$ length, and depth is $O(\log n)$, so total: $O(L \log n)$

But if using FFT, depth of computation graph: $O(\log L)$; each with depth $O(\log n)$; so total: $O(\log L \log n)$

Also, given A. B. C, how can we compute K efficiently? We can tell that the bottleneck is $A^?$ $\Rightarrow (I, A, \cdots, A^{L-1}) = X^L$

We can use divide and conquer for this:

$$X^L = (X^{L/2}, A^{L/2} X^{L/2}), \quad A^L = A^{L/2} A^{L/2}$$

So for each $L = 2^l$, we can compute X^L: l times, each with depth of $O(\log n)$. Therefore, total: $O(\log n \cdot \log L)$

But If we add structure to A? Suppose A is diagonal, then we can avoid matrix multiplication and degrade complexity to $O(\log L)$

However A can't always be that perfect. Suppose $A = I_n + pp^T$, $p \in \mathbb{R}^n$

Now: $A^L = (I_n + pp^T)^L = I_n + L pp^T + \binom{L}{2} pp^T pp^T + \cdots + (pp^T)^L$

and $(pp^T)^L = p(p^T p)^{L-1} p^T = pp^T \cdot v^{L-1}$, where $v = \|p\|_2^2$

Then $A^L = I_n + pp^T \underbrace{\sum_{l=1}^{L} \left[\binom{L}{l} v^{l-1} \right]}$, and computing this can be done in $O(\log L)$ using divide and conquer (No matrix multiplication!)

Topic 2: Self-Supervised Linear Purification

Consider non-trivial purification scenario: $\mathcal{L}(W; X, \lambda) = \|X - WX\|_F^2 + \lambda \|W\|_F^2$

Where: $W \in \mathbb{R}^{m \times m}$, $X \in \mathbb{R}^{m \times n}$.

Assume $\sigma_1 > \cdots > \sigma_m \ge 0$, $X = U \Sigma V^T$. Then we can actually work out the optimal $\hat{W}$:

$$\hat{W} = U \cdot \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2 + \lambda} & & \\ & \frac{\sigma_2^2}{\sigma_2^2 + \lambda} & \\ & & \ddots \\ & & & \frac{\sigma_m^2}{\sigma_m^2 + \lambda} \end{bmatrix} U^T$$

KOKUYO

No.

Date

Proof: $\mathcal{L}(W; X, \lambda) = \|X - WX\|_F^2 + \lambda \|W\|_F^2 = \sum_{i=1}^{m} \|X_i - W_i X\|_2^2 + \lambda \|W_i\|_2^2$ ($X_i \in \mathbb{R}^{1\times n}$, $W_i \in \mathbb{R}^{1\times m}$)

$= \sum_{i=1}^{m} \|X_i^T - X^T W_i^T\|_2^2 + \lambda \|W_i^T\|_2^2$

We're familiar with this: $\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \Rightarrow \hat{\beta} = (X^TX + \lambda I)^{-1} X^T y$

$\therefore \hat{W}_i^T = (XX^T + \lambda I)^{-1} XX_i^T \Leftrightarrow \hat{W} = XX^T(XX^T + \lambda I)^{-1}$

$= \dots = U\Sigma\Sigma^T(\Sigma\Sigma^T + \lambda I)^{-1} U^T$

Topic3. Ridge Attention

Firstly: updating simple averaging can be efficient! $m = \frac{1}{n}\sum_{i=1}^{n} x_i$,

then if x_{n+1} is looped in, $m' = \frac{mn + x_{n+1}}{n+1}$

Consider Ridge Regression. $A = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$ $y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ $x_i \in \mathbb{R}^d$, y_i: scalar

so $w^* = (A^TA + \lambda I)^{-1} A^T y$.

Now want to rewrite $(A^TA + \lambda I)$ and A^Ty as sums involving x_i and y_i

① $A^TA + \lambda I = \lambda I + \sum_{i=1}^{n} x_i x_i^T$ ② $A^Ty = \sum_{i=1}^{n} x_i y_i$

Now suppose we want to do ridge-self-attention*, with a context length of n and d-dim q.k.v. *: n queries is applied to the same pool of (k.v) pairs

k: A matrix; q_n: x; y: v, so actually we want W^*, and inference: $W^{*T} q_n$

For $W^{*T} q_n$: $W^* \in \mathbb{R}^{d\times d}$ $q_n \in \mathbb{R}^{d\times n}$ $\Rightarrow O(nd^2)$.

For computing $W^* = (\lambda I + A^TA)^{-1} A^TB$, $A = \begin{bmatrix} k_1^T \\ \vdots \\ k_n^T \end{bmatrix}$ and $B = \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$

$O(nd^2)$, inverse $O(d^3)$. $O(nd^2)$

$\therefore$ Complexity: $O(nd^2) + O(d^3)$; if $n > d$, then $O(nd^2)$

But for inverse: a nice trick: Sherman-Morrison Formula.

$$(M + UV^T)^{-1} = M^{-1} - \frac{1}{1 + V^TM^{-1}U}(M^{-1}U)(V^TM^{-1})$$

if u, v are two d-dimension vector and $M \in \mathbb{R}^{d\times d}$, $O(d^2)$

Date

HW9 整理 & Discussion Q, 10

* q_i of $q_h \sim \mu_i, 6$

Topic 1. Justifying Scaled-Dot Product Attention

Suppose $q, k \in \mathbb{R}^d \sim N(\mu, \sigma^2 I)$* Then: ($\mu \in \mathbb{R}^d$)

$E[q_h^T k] = E[\sum_{i=1}^d q_i k_i] = \sum_{i=1}^d E[q_i]E[k_i] = \sum_{i=1}^d \mu_i^2 = \|\mu\|_2^2$

If $\mu=0, \sigma=1$, then $Var(q_h^T k) = E[(q_h^T k)^2] - E[q_h^T k]^2 \to d \times \sigma^4$

$E[(q_h^T k)^2] = E[k^T q_h q_h^T k]^2 = \cdots$?

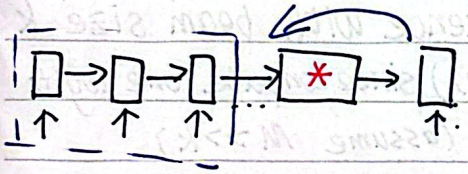
So $Var(q_h^T k) = Var(\sum_{i=1}^d (q_i k_i)) = d\,Var(q_i k_i)$

$=^* d(E[q_1]^2 Var(k_1) + E[k_1]^2 Var(q_1) + Var(q_1)Var(k_1))$

$= d\sigma^4 = d$ *: $Var(XY) = Var(X)Var(Y) + Var(X)E[Y]^2 + Var(Y)E[X]^2$

So now, we want $E[\frac{q_h^T k}{s}] = 0$, and $Var[\frac{q_h^T k}{s}] = \frac{d}{s^2} = 1 \Rightarrow s = \sqrt{d}$

Topic 2: Attention Mechanism to Mitigate RNN bottleneck



In RNN, many to many, the bottleneck is that * need to store all information about the input.

But now, we allow the hidden states from decoder as queries and hidden states from encoder as keys and values.

So decoder hidden state can compute similarity with input's hidden state, 'look at them' $\Rightarrow \hat{h}_i = \sum_{j=0}^{L} a_{j,i}$ henc, attention score.

Topic 3: RoPE

PE is used to ensure that word position is known. Because attention is applied symmetrically to all input vectors. For RoPE: $RoPE(x_t^{(1)}, x_t^{(2)}, t)$

$= \begin{pmatrix} \cos tw & -\sin tw \\ \sin tw & \cos tw \end{pmatrix} \begin{pmatrix} x_t^{(1)} \\ x_t^{(2)} \end{pmatrix}$ for 2-dim vectors Moreover:

if $z_t = x_t^{(1)} + j\, x_t^{(2)}$, then $RoPE(z_t, t) = e^{jtw} z_t$, hence $RoPE(x,m)^T RoPE(y,n)$

$= RoPE(x, m+k)^T RoPE(y, n+k)$

So we can extend to 2n-dim vector:

$RoPE((x^{(1)}, x^{(2)}, \cdots, x^{(m)}), t) = \begin{pmatrix} x^{(1)}\cos tw^{(1)} - x^{(2)}\sin tw^{(1)} \\ x^{(2)}\cos tw^{(1)} + x^{(1)}\sin tw^{(1)} \\ \vdots \end{pmatrix}$ d: dim of input vector

RoPE doesn't add any learnable parameter! w is fixed! $w^{(i)} = 10000^{-2\cdot\frac{i-1}{d}}$

△: Note that in some model (specific. like GPT), it might be possible for attention layer itself to learn to encode positional infomation.

KOKUYO

No.

Date

HW10 & HW11 整理

Topic 1: Kernelized Linear Attention

$O(N^2)$ complexity in memory and computation is bad. Let $x \in \mathbb{R}^{N \times F}$ be the sequence. The transformer is a function $T: \mathbb{R}^{N \times F} \to \mathbb{R}^{N \times F}$:

$$T_l(x) = f_l(A_l(x) + x)$$

Formally: $W_Q \in \mathbb{R}^{F \times D}$, $W_K \in \mathbb{R}^{F \times D}$, $W_V \in \mathbb{R}^{F \times M}$, then:

$$\begin{cases} Q = xW_Q, \quad K = xW_K, \quad V = xW_V \\ A_l(x) = V' = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V \end{cases}$$

$\text{sim}(q_i, k) = \exp\left(\frac{q_i^T k}{\sqrt{D}}\right)$

But generically: $V_i' = \dfrac{\sum_{j=1}^{N} \text{sim}(Q_{i\cdot}, K_{j\cdot})\, V_j}{\sum_{j=1}^{N} \text{sim}(Q_{i\cdot}, K_{j\cdot})}$

One potential attention variant is 'polynomial kernel attention'. E.g., degree is 2: $\text{sim}(q_i, k) = (q_i^T k + 1)^2 = \phi(q_i)^T \phi(k)$

Then one benefit of using kernalized attention is that we can represent a kernel using a feature map: $\phi(\cdot)$. In this case:

$$\phi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \cdots, \sqrt{2}x_D, x_1^2, \cdots, x_D^2, \sqrt{2}x_1x_2, \cdots \sqrt{2}x_{D-1}x_D]^T$$

Then, kernel can be: $K(q_i, k) = (\phi(q_i)^T \phi(k))$, then:

$$V_i' = \frac{\sum_{j=1}^{N} \phi(Q_{i\cdot})\phi(K_{j\cdot})^T V_j}{\sum_{j=1}^{N} \phi(Q_{i\cdot})\phi(K_{j\cdot})^T} = \frac{\phi(Q_{i\cdot})\sum_{j=1}^{N} \phi(K_{j\cdot})^T V_j}{\phi(Q_{i\cdot})\sum_{j=1}^{N} \phi(K_{j\cdot})^T}$$

What's better: j is irrelavant to $\phi(Q_{i\cdot})$:

For quadratic kernel, space complexity is $O(ND + D^3)$

If $N >> D^2$, then lot's of memory can be saved.

Topic 2: LoRA

A common strategy to do pretrained model adapting is to update only a subset of its parameters and keep the rest frozen.

In this topic, focus on $W \in \mathbb{R}^{m \times l}$. W_0 is the pretrained value.

In LoRA: $W = W_0 + AB$, $A \in \mathbb{R}^{m \times k}$, $B \in \mathbb{R}^{k \times l}$, $k \ll \min(m, l)$

(a) Suppose using LoRA to adapt to a new task and observe underfitting

① Try $k \uparrow$ to allow more expressive power

② Change initialization to leverage SVD of either the original

KOKUYO

matrix or matrix-valued gradient for a large amount of training data. By aligning initializations to natural gradient directions, we are likely to be able to have more significant impacts faster.

③ Change learning rate or other hyperparameter

(b). If A, B are initialized to all zero ⇒ Zero gradient for A&B. No!
But if both Xavier initialized, $AB \neq 0$, and feature of pretrained model is not used.

Topic 3: A brief introduction of Transformer Interpretability.

A core concept is residual stream:

$$X^{l+1} = X^l + f^{(l)}(X^l) \Rightarrow X^{final} = X^0 + \sum_{l=0}^{L-1} f^{(l)}(X^l)$$

Consider attention head with H heads. In 'concatenation' view, each head's output are stacked together vertically, followed by a W_O^H. So $H = W_O^H \cdot R^H$ $W_O^H \in \mathbb{R}^{d_{model} \times (d_{head} \cdot H)}$

Consider $W_O^h \in \mathbb{R}^{d_{model} \times d_{head}}$, head output $r^h \in \mathbb{R}^{d_{model} \times n_{context}}$

So H is also: $H = \sum_{h=1}^{H} W_O^h r^h$, these two are equivalent.

$$H = W_O^H R^H = [W_O^1, W_O^2, \cdots, W_O^H] \begin{bmatrix} r^1 \\ \vdots \\ r^H \end{bmatrix} = \sum_{h=1}^{H} W_O^h r^h$$

It means that we can analyze the contributions of each head independently.

The 'concat' view can be efficient, but 'additive' view reveal independency

'Circuit Theory' treats Transformer as a collection of interacting circuits built from attention and MLP.

QK circuit: Determining Attention Patterns ↓ information to move

OV circuit

An attn head can be split into two operations: where to look and what

Pre-softmax score: $S_{ij} = Q_i^T K_j$, $Q_i = W_Q X_i$, $K_j = W_K X_j \Rightarrow S_{ij} = X_i^T (W_Q^T W_K) X_j$, W_{QK}

If $W_{QK} = I$ ⇒ 内积; if $W_{QK} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, then this head only cares about first dim.

There can be 负数 in W_{QK} which encourage dissimilarity. These mean that W_{QK} can be highly robust and obtain powerful expressivity.

Attn pattern from QK circuit

OV Circuit: $h(X^0) = W_O^h W_V^h X^0 (A^h)^T = W_{OV}^h X^0 (A^h)^T$ (post-softmax)

$X^0 (A^h)^T$: attention-weighted average. ; Then: $X^{final} = X^0 + \sum_{i=1}^{H} h_i(X^0)$

$= X^0 + \sum_{i=1}^{H} W_{OV}^i X^0 (A^i)^T$ We can notice that $h_i(X^0)_t$ (t-th column) $\in Col(W_{OV}^i)$

Campus ⇒ Head can only write to a low-dim subspace of the d_{model}-dim residual stream