GDT: White-box transformers

Claim: l'architecture assez spécifique des transformers découle "naturellement" de certains choix de modélisation sur la manière d'encoder des données/token.

→ meilleure interprétabilité (≠ black box)

→ ouvre la porte à des architectures plus adaptées à un pb.

I Rappels sur l'architecture des transformers

1) Architecture à haut niveau

Un transformer manipule des tokens $x \in \mathbb{R}^d$, obtenus après tokenisation de données (ici d est la dimension du modèle $\sim 10^4$)

Plus précisément on traite une suite de tokens, de longueur T

$$X = \left[x_1 \middle| \cdots \middle| x_T \right] \in \mathbb{R}^{d \times T}$$

Rq: Ici les vecteurs sont des colonnes, contrairement à l'exposé précédent.

Un transformer est une fonction $f : \mathbb{R}^{d \times T} \longrightarrow \mathbb{R}^{d \times T}$ qui est la composée de L couches / blocs de transformer $f = f_L \circ \cdots \circ f_1$ $[L \sim 10^2]$

Chaque bloc est lui même composé de deux blocs:

$$f_i = \text{FFN} \circ \text{ATT}$$

↳ bloc d'attention : essentiel

FFN ← petit réseau de neurones

2) Bloc d'attention

- Couche d'attention classique $SATT : \mathbb{R}^{d \times T} \longrightarrow \mathbb{R}^{d \times T}$

$$SATT(X) = (VX)\ \text{softmax}\left[(KX)^T(QX)\right]$$

où $V, K, Q \in \mathbb{R}^{d \times d}$ sont les paramètres de value, key, query.

- Couche d'attention multi-tête $MHATT : \mathbb{R}^{d \times T} \longrightarrow \mathbb{R}^{d \times T}$

$$MHATT(X) = A \begin{bmatrix} SATT_1(X) \\ \vdots \\ SATT_H(X) \end{bmatrix}$$

où $SATT_h(X) = (V_h X)\ \text{softmax}\left[(K_h X)^T (Q_h X)\right]$

avec $V_h, K_h, Q_h \in \mathbb{R}^{p \times d}$, $p \ll d$ (typiquement $p = \frac{d}{H}$)

et $A \in \mathbb{R}^{d \times Hp}$ une matrice d'agglomération qui combine les infos de toutes les heads.

- En general la couche d'attention ne se résume pas à MHATT. Il y a également :

  * la skip connection / residual connection, qui consiste à ajouter l'id :

$$X \longmapsto X + MHATT(X)$$

  * une couche de normalisation

$$X \longmapsto \gamma \frac{X}{|X|} \quad (=: RMS_\gamma(X))$$

En conclusion le bloc d'attention s'écrit :

$$\boxed{X \longmapsto RMS_\gamma\left(X + MHATT(X)\right)}$$

3) FFN

la couche Fully connected Feedforward Network la plus simple est le Multi Layer Perceptron (MLP) [Désuet]

$$\text{MLP} : \mathbb{R}^{d \times T} \longrightarrow \mathbb{R}^{d \times T}$$
$$X \longmapsto (\varphi(x_1), \dots, \varphi(x_T))$$

où
$$\varphi : \mathbb{R}^d \longrightarrow \mathbb{R}^d$$
$$x \longmapsto A_2\, \sigma(A_1 x)$$

avec $A_1 \in \mathbb{R}^{d' \times d}$ et $A_2 \in \mathbb{R}^{d \times d'}$

II Un paradigme white box (et de belles promesses)

1) Idée générale

On veut trouver/apprendre une fonction $f : \mathbb{R}^{d \times T} \longrightarrow \mathbb{R}^{d \times T}$ qui soit capable de transformer des données d'entrée X en Y, de telle façon que même si X soit une distribution non linéaire et multimodale, alors Y est essentiellement linéaire (par morceaux) et compacte.

Rq : Ici $Y = (y_1, \dots, y_T)$. Il est vraisemblable que la loi jointe sur Y soit compliquée, mais on s'attend à ce que les marginales soient simples !

Ce genre d'hypothèse peut se modéliser par le fait que f doive minimiser un certain critère, càd qu'elle soit solution

d'un problème $\min_{f \in \mathcal{F}} \mathbb{E}_Y \Phi(Y, \text{param})$
$Y = f(X)$

les auteurs proposent alors l'idée suivante : une bonne fonction candidate est la transformation $Y \longmapsto \text{algo}(Y)$ où "algo" est une étape d'un algo de minimisation par la fonction $Y \mapsto \Phi(Y, \text{param})$

Rq : Pourquoi passe-t-on de la minimisation sur f à une minimisation sur Y ???

L'avantage de cette approche est que l'on peut voir la succession de couches comme le fait que les données passent successivement les itérés de cet algo d'optimisation.

2) Sparse Rate Reduction

a) Taux d'encodage

Les auteurs se tournent vers des outils de la théorie de l'information pour obtenir une fonction $R(\cdot)$ qui mesure à quel point des données sont bien "compressées", représentées sous forme "compacte".

Rq : Je ne suis pas sûr mais je crois que pour des distributions gaussiennes on a

$R_\varepsilon(Y) = \min I(Y, \hat{Y})$ ← information mutuelle
$\mathbb{E}\|Y - \hat{Y}\|^2 \leq \varepsilon$ ← taux d'erreur admis lors de la quantisation cf Cover Thomas chap 10

En tout cas c'est vrai pour les Gaussiennes.

b) ~~Gain d'information~~ Mixtures de Gaussiennes

* On va faire l'hypothèse que les tokens y_i suivent une distribut° qui est une mixture de Gaussiennes de basse dimension

↳ hypothèse de modélisat° centrale du papier

Plus précisément on va parler de K gaussiennes, centrées ($\mu_k = 0$) avec ~~ess~~ covariance $\Sigma_k \in \mathbb{R}^{d \times d}$ de faible rang, rang $\Sigma_k = r$. ← p dans le papier.
On note $U_k \in \mathbb{R}^{d \times r}$ la famille orthonormale qui engendre le support.

c) Gain d'information

Par mesure qu'une distribution Y est bien ~~ess~~proche d'un mixture de Gaussienne, on va vouloir que

- le taux d'encodage de Y vu comme mixture est bas (= bon)
- " " " " autre chose (Ex: $N(0, \Sigma)$ est haut

On va donc vouloir minimiser $R_{mix}(Y)$ $[-R_{gauss}(Y)]$ ← - taux reduction - information gain

Lem: 1) $R_{gauss}(Y) \approx \frac{1}{2} \log\det\left[I + \frac{d}{T\varepsilon^2} Y^T Y\right]$

2) $R_{mix}(Y) \approx \sum_{k=1}^{K} R_{gauss}(U_k^T Y)$ ← ~~ess~~ cf a). Pas discuté ds le papier.

Dem 2): $U_k^T Y$ projette Y sur le ~~ess~~ support de Σ_k, ce qui donne une borne supérieure naïve de R_{mix}

Minimiser cette fct va promouvoir des Y qui sont des mix Gauss.
↳ $R_{mix}(Y)$

d) Sparse encoding

La fonction $R_{mix}[-R_{gauss}]$ est invariante par changement de base car c'est une mesure "intrinsèque" de la représentation.

Du coup le pb est mal posé et va par encore spécifier des caractéristiques pr notre distribut°. (*)

↳ cf + tard

~~Tant qu'à faire on va demander que la représentation Y soit sparse.~~

~~Bac on va aussi rajouter un terme $+\lambda\|Y\|_1$ au pb. Ahah $Y \geq 0$ aussi car ça arrange...~~

Rq: Argument un peu confus, par moi sparse voudrait dire jouer sur la taille du rang r..?

e) Le problème final

possiblement remplacé ?! oui

$$\min_{\substack{Y \geq 0 \\ Y = f(x)}} \mathbb{E}_Y \underbrace{\lambda\|Y\|_1 + R_{mix}(Y) - R_{gauss}(Y)}_{\Phi(Y)}$$

D^*

ici les paramètres sont $U_K \in \mathbb{R}^{d \times r}$, $\lambda, \varepsilon \in \mathbb{R}$

3) Dén algo ou transformer

On va minimiser $\Phi(Y)$ en deux étapes:

- descente de gradient (GD) sur $R_{mix}(Y)$
- ~~prox-gradient / forward-backward~~ (PGD) sur ~~$\lambda\|Y\|_1 + \delta_{Y \geq 0} - R_{gauss}(Y)$~~ $\lambda\|D^r Y\|_1^+$

Ainsi on aura une pipeline:

$$Y_\ell \xrightarrow{\text{GD}} Y_{\ell+1/2} \xrightarrow{\text{prox}} Y_{\ell+1}$$

$\simeq$ ATT $\quad$ $\simeq$ FFN

a) GD sur R_{mix}

On veut pour $Y_{\ell+\frac{1}{2}} = Y_\ell - \gamma \nabla R_{mix}(Y_\ell)$

Pro: $Y - \gamma \nabla R_{mix}(Y) \approx (1-\hat{\gamma})Y + \hat{\gamma}\,\widehat{MHATT}(Y)$

où $\widehat{MHATT}(Y) = A \begin{bmatrix} \widehat{SATT}_1(Y) \\ \vdots \\ \widehat{SATT}_K(Y) \end{bmatrix}$

avec $\widehat{SATT}_K(Y) = (U_K^T Y)\,\text{softmax}\left[(U_K^T Y)^T (U_K^T Y)\right] \in \mathbb{R}^{r \times T}$

et $A = [U_1 \dots U_K] \in \mathbb{R}^{d \times Kr}$

On voit que

- les Q_k, K_k, V_k sont tous égaux à U_k^T → vague justific° bio
- la matrice d'agglomération A est aussi paramétrée par les U_k
- nb de gaussiennes K = nb de têtes H
- ~~[illegible]~~ rang des gaussiennes r = projet° des têtes p
- on retrouve à peu près la skip connection
  - ↳ $\hat{\gamma}$ peut être absorbé par A
  - ↳ $1-\hat{\gamma}$ dans les U_k via chgmt var ?

) pas discuté dans la ppio
- on n'a pas la norm layer. (purement technique ?)

Dem: $R_{mix}(Y) = \sum R(U_k^T Y)$ où $\nabla R(Y) = \alpha\, Y (I + \alpha Y^T Y)^{-1}$, $\alpha = \frac{d}{T\varepsilon^2}$

Donc $\nabla R_{mix}(Y) = \alpha \sum U_k U_k^T Y \left(I + \alpha (U_k^T Y)^T (U_k^T Y)\right)^{-1}$

Or $(I+A)^{-1} \approx I - A$

↳ $\approx \alpha \sum U U^T Y (I - \alpha (U^T Y)^T (U^T Y))$

et $(U^T Y)$ = proj° de Y sur U donc $(U^T Y)^T (U^T Y) \approx$ ☹ $\text{softmax}\left((U^T Y)^T (U^T Y)\right)$

Après c'est un jeu de réécriture.

b) PGD sur $\|\cdot\|_1$ - R_{gauss}

On veut poser $Y_{l+1} = prox_{\gamma\phi}\left(Y_{l+\frac{1}{2}} + \gamma \nabla R_{gauss}(Y_{l+\frac{1}{2}})\right)$

$\lambda\|\cdot\|_1 + \delta_{\geq 0}$

Pro: En fait c'est complètement con, donc on va sauter sa partie fin.

(*) 2.d la suite

Rq: Ici j'improvise en m'inspirant du papier afin d'avoir un truc qui tienne la route.

Minimiser R_{mix} est invariant par transformée orthogonale, donc on peut rajouter un prior. Ici les auteurs suggèrent que l'output soit, si possible, sparse dans un dictionnaire.

On se donne donc $D = [d_1 | \dots | d_{d_{atoms}}]$ qu'on suppose orthogonale (les auteurs disent que avec famille incohérente $D^*D \simeq I$ ça passe) et on voudrait que le $Y = [y_1 | \dots | y_T]$ en sortie soit tel que $y_t \simeq d_{i_t}$ càd $y_t \underset{\simeq}{\in} D$. Ceci équivaut à $y_t = De_{i_t} \Leftrightarrow D^* y_t = e_{i_t}$ $\Leftrightarrow D^* y_t$ sparse $\forall t \Rightarrow D^* Y$ sparse $d_{atoms} \geq 0$

Rq: Bon ici c'est pas top car la sparsité de Y n'équivaut pas à la sparsité des y_t. Il faudrait une notion de sparsité par groupes...

Donc on va rajouter un terme de pénalisation $+\lambda \|D^* Y\|_1$ où $\|\ \|_1 \sim \|\ \|_0$ (standard). Et aussi $D^* Y \geq 0$ donc $+\lambda \|D^* Y\|_1^+$

où $\|x\|_1^+ = \begin{cases} \|x\|_1 & \text{si } x \geq 0 \\ +\infty & \text{sinon} \end{cases}$ càd $\|x\|_1 + \delta_{\mathbb{R}_+^{atoms}}(x)$.

b) Prox de $\|D^* Y\|_1^+$

On veut pour $Y_{t+1} = \text{prox}_{\gamma\lambda\|D^*\cdot\|_1^+}(Y_{t+\frac{1}{2}})$

Lem : $\text{prox}_{\gamma\lambda\|D^*\cdot\|_1^+}(Y) = D\sigma(D^*Y - \gamma\lambda \mathbb{1})$

↑ ReLU ⇐ dès lors $D^*Y \geq 0$ ... BoF

↑ matrice que des 1

On voit que

- On retrouve une architecture MLP avec $A_1 = D$ $A_2 = D^*$
- Il y a un terme de biais (normal ? non ? pas discuté dans le papier)
- La ReLU est un peu forcée selon moi.

4) Bonus : ce que les auteurs font vraiment

* Dans un premier temps les auteurs proposent de minimiser

$$R_{mix}(Y) - R_{gauss}(Y) + \lambda\|Y\|_1^+$$

en disant qu'ils veulent Y sparse (pr pas..) et que la différence $R_{mix} - R_{gauss}$ a du sens (information gain pr discriminer mix vs gauss).

Donc l'étape (b) consiste à minimiser $\lambda\|Y\|_1^+ - R_{gauss}(Y)$

* Ils font le calcul exact en annexe, avec un pas prox-gradient :

$$Y_{t+1} = \text{prox}_{\gamma\lambda\|Y\|_1^+}(Y_{t+\frac{1}{2}} + \gamma \nabla R_{gauss}(Y_{t+\frac{1}{2}}))$$

où le prox va donner un ReLU et

$$\nabla R(Y) = \alpha Y (I + \alpha Y^T Y)^{-1}, \quad \alpha = \frac{d}{T\varepsilon^2}.$$

Là ils font l'hypothèse que $Y_{t+\frac{1}{2}}$ est orthogonale et donc que $\nabla R(Y) = \frac{\alpha}{1+\alpha} Y$

Déf une update $Y_{t+1} = \text{prox}\left(Y + \frac{\gamma d}{1+\alpha} Y\right) = \text{prox}\left(\left(1+\frac{\gamma d}{1+\alpha}\right) Y\right)$

$$= \sigma\left[\left(1+\frac{\gamma d}{1+\alpha}\right) Y - \gamma\lambda \mathbb{1}\right]$$ (en vrai ils oublient le γ une ligne sur deux...)

Bon là ils sortent du chapeau que oui R est invariant par transformation orthogonale, et avec un changement de variable illégal fait apparaître D :

$$Y_{t+1} = \sigma\left[\left(1+\frac{\gamma\lambda}{1+\alpha}\right) D^T Y_{t+\frac{1}{2}} - \gamma\lambda \mathbb{1}\right]$$

* Dans le corps du papier ils disent que $\nabla R_{\text{gauss}}(Y)$ trop cher à calculer en général et donc laissent tomber leur analyse.

Ils sortent ad hoc le fait qu'on a $Y_{t+\frac{1}{2}}$ et qu'il faut le sparsifier, à savoir trouver Y_{t+1} sparse tq $DY_{t+1} = Y_{t+\frac{1}{2}}$, donc il faut maintenant résoudre un nouveau pb $\Rightarrow Y_{t+1} = \underset{Y \geq 0}{\text{argmin}}\ \lambda\|Y\|_1 + \frac{1}{2}\|DY - Y_{t+\frac{1}{2}}\|_F^2$

Ceci donne $$Y_{t+1} = \sigma\left[Y + \gamma D^T(Y - DY) - \gamma\lambda \mathbb{1}\right]$$

- Dans les deux cas ils veulent vraiment output les paramètres sparse et pas les représentations ... (et donc perd le A_1)
- Casse totalement l'histoire du papier
- Ils disent vaguement que c'est de l'optim alternée, mais

$$\min_{Y,Z}\ \lambda\|Y\|_1 + \frac{1}{2}\|DY - Z\|_F^2 + R_{\text{mix}}(Z)$$

fera que la 1ère étape devrait dériver ici

III Quelques points positifs

1) XPs

- Les auteurs vérifiat que l'architecture fait bien ce qu'elle prétend (pas A mais appris)
  - → compression ($R_{mix} \searrow$)
  - → sparsité ($|| ||_1 \searrow$)
- Ils ne testent pas si les transformers standard le font ?
- À nb de paramètres $\approx$, $\hat{m}$ perf que transformer standard (ViT)
  - → possible avantage du partage des param entre VICRA
  - → ne testent "que" sur ImNet ? CIFAR finetunings ? pas de texte ?
- Bonnes lois de scaling par les hyperparamètres.

2) Un dernier nugget théorique

Les auteurs ont une autre histoire pour justifier le softmax dans le transformer.

Supposons $N=1$

On veut que encoder $x \mapsto y$ avec y mixture de Gaussiennes

On va faire comme si x était une moins bonne approximation de y, càd $x = z + \sigma\omega$ où $\omega \sim \mathcal{N}(0, I)$.

Retrouver z à partir de x peut se faire en calculant

$$z_{\ell+1} = \mathbb{E}[z \mid z_\ell] = \underset{f}{\operatorname{argmin}}\ \mathbb{E}_{z,\omega} \| f(z + \sigma^\ell \omega) - z \|_2^2$$

Pro (Tweedie) $z_{\ell+1} = z_\ell + \sigma_\ell^2 \nabla \log q_\ell(z_\ell)$ où q_ℓ densité de z^ℓ

Thm: Sous hypothèses,

$$z_{\ell+1} \approx A \begin{bmatrix} \widehat{SATT}_1(z_\ell) \\ \vdots \\ \widehat{SATT}_K(z_\ell) \end{bmatrix}, \quad \widehat{SATT}(z) = \text{softmax}\left(\frac{1}{2\sigma_\ell^2} (U_k^T z)^T (U_k^T z) \right)$$

↑ pas de V! ni skip connexion

$$A = [U_1 \cdots U_K]$$