Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Apprentissage de sous-espaces de préfixes

Louis Falissard[1, 2] Vincent Guigue[3] Laure Soulier[1, 4]
(1) Sorbonne Université, CNRS, ISIR, 75005 Paris, France
(2) Bibliothèque nationale de France, 75013 Paris, France
(3) AgroParisTech, UMR MIA-PS, 91120 Palaiseau, France
(4) Université Paris Saclay, CNRS, LISN, 91400 Orsay, France
louis.falissard@gmail.com, vincent.guigue@isir.upmc.fr,
laure.soulier@isir.upmc.fr

RÉSUMÉ

Cet article propose une nouvelle façon d'ajuster des modèles de langue en "Few-shot learning" se basant sur une méthode d'optimisation récemment introduite en vision informatique, l'apprentissage de sous-espaces de modèles. Cette méthode, permettant de trouver non pas un point minimum local de la fonction coût dans l'espace des paramètres du modèle, mais tout un simplexe associé à des valeurs basses, présente typiquement des capacités de généralisation supérieures aux solutions obtenues par ajustement traditionnel. L'adaptation de cette méthode aux gros modèles de langue n'est pas triviale mais son application aux méthodes d'ajustement dites "Parameter Efficient" est quant à elle relativement naturelle. On propose de plus une façon innovante d'utiliser le simplexe de solution étudié afin de revisiter la notion de guidage de l'ajustement d'un modèle par l'inférence d'une métrique de validation, problématique d'actualité en "few-shot learning". On montre finalement que ces différentes contributions centrées autour de l'ajustement de sous-espaces de modèles est empiriquement associée à un gain considérable en performances de généralisation sur les tâches de compréhension du langage du benchmark GLUE, dans un contexte de "few-shot learning".

ABSTRACT

Learning prefix subspaces

This paper proposes a new way of fitting language models in few-shot learning based on a recently introduced optimization method in computer vision, model subspace learning. This method, allowing to find not a local minimum point of the cost function in the parameter space of the model, but a whole simplex associated with low values, typically presents higher generalization capabilities than solutions obtained by traditional fitting. The adaptation of this method to large language models is not trivial, but we observe that its application to so-called "Parameter Efficient" fitting methods is relatively natural. We also propose an innovative way to use the studied solution simplex in order to revisit the notion of guiding the adjustment of a model by inferring a validation metric, a current problem in "few-shot learning". We finally show that these different contributions centered around the adjustment of model subspaces is empirically associated with a considerable gain in generalization performance on the GLUE benchmark language understanding tasks, in a "few-shot learning" context.

MOTS-CLÉS : Modèles de langues, apprentissages sur petits échantillons, apprentissage de sous-espaces, classification de texte.

KEYWORDS: Large language models, few-shot learning, subspace learning, text classification.

1 Introduction

L'avènement au cours des dernières années des gros modèles de langues (Devlin *et al.*, 2019; Radford *et al.*, 2019; Raffel *et al.*, 2019) a été la source d'une évolution considérable des applications de méthodes d'apprentissage profond en traitement automatique des langues. Ces modèles, pré-entraînés de manière non-supervisée sur des corpus de données textuelles massifs, permettent notamment l'ajustement de puissant modèles neuronaux à partir de quelques milliers, voire centaines d'observations, avec des performances de généralisation qui demandaient encore il y a quelques années plusieurs millions d'observations. Plus récemment encore, l'augmentation en dimensionnalité de ces modèles, couplée à des corpus de pré-entraînement encore plus massifs, permet de les utiliser comme base pour l'implémentation de puissant algorithme de classification de texte, ceci avec une poignée d'observations, notamment par leur utilisation en conjonction à des prompts d'instruction discrets (Radford *et al.*, 2019).

L'extension de ces méthodes discrètes à l'apprentissage de prompts différentiables, qui vient s'inscrire au moins conceptuellement dans le cadre des méthodes dites de "Parameter Efficient Fine-Tuning" (PEFT) (Houlsby *et al.*, 2019; Bapna & Firat, 2019), pose cependant quelques problèmes dans un cadre de few-shot learning, dont notamment celui du guidage par la métrique de validation de l'ajustement du modèle pendant l'optimisation par descente. En effet, il est coutume, dans le processus d'ajustement d'un modèle neuronal, d'exclure au préalable environ un tiers des observations du jeux de données d'entraînement afin de créer un jeu dit de validation (ou de développement), consacré à l'inférence d'une estimation non biaisée des performances du modèle. Cette métrique est utilisée autant pendant ajustement du modèle (pour estimer la convergence de l'algorithme de descente, ou pour informer une heuristique d'arrêt prématuré) qu'en aval pour guider la recherche d'hyperparamètres typiquement utilisée dans le cadre d'ajustement de gros modèles de langues. Le bien fondé de cette approche repose en revanche sur la garantie que la distribution du jeu de validation est un minimum représentative de celle du phénomène réel observé. Cette garantie peut très vite perdre de son sens dans un cadre de "few-shot learning", où une dizaine d'observations tout au plus est disponibles à l'estimation de la métrique de validation. Cette notion est de nos jours devenue suffisamment problématique pour qu'une partie de la littérature académique en apprentissage continu sur petits jeux de données présente des résultats d'expériences utilisant des jeux de validations autant irréaliste qu'artificiels, comportant jusqu'à plusieurs ordres de grandeurs plus d'observations que le jeu d'entraînement utilisé pour l'ajustement même du modèle (Wortsman *et al.*, 2021).

Récemment introduit en vision informatique, le concept d'apprentissage de sous-espace de paramètres de modèles (qu'on appellera par soucis de concision "méthode des sous-espaces" (Wortsman *et al.*, 2021) est une technique d'optimisation permettant de trouver non pas minimum local de la fonction coût dans l'espace des paramètre du modèle, mais tout un simplexe associé à de faibles valeurs de cet objectif, comme illustré en figure 1. Les modèles ajustés par le biais de cette méthode présentent notamment des capacités de généralisation supérieures aux solutions obtenues par ajustement traditionnel. Ce phénomène, expliqués empiriquement par les propriétés enviables des minimums locaux qu'elles permettent d'identifier, revêt un intérêt tout particulier lorsqu'on l'observe à travers le prisme d'une problématique de *few-shot learning*, où la capacité du modèle à généraliser une classe de concept à partir d'un nombre réduit d'exemple est clé. Son application directe à l'ajustement de gros modèles de langue, en revanche, n'est pas trivial. En effet, cette méthode de sous-espace propose

d'obtenir ce simplexe de solutions (dans l'espace des paramètres du modèle étudié) par un unique processus de descente de la manière suivante :

— Un simplexe de modèles est initialisé aléatoirement (par initialisation de chacun de ses sommets via une méthode classique d'initialisation non déterministe de modèles neuronaux).
— Un modèle est construit par échantillonnage uniforme sur le simplexe à chaque itération de descente, et utilisé pour l'inférence, le calcul de la fonction objectif et du gradient, de manière à ajuster les sommets du simplexe.

Une fois l'ajustement par descente de gradient terminé, le simplexe peut être utilisé soit dans le cadre de méthodes d'ensemble, soit en choisissant un modèle unique dans le simplexe (généralement son centroïde).

On comprend clairement en se basant sur cette observation pourquoi cette méthode n'a (du moins à notre connaissance) jamais été appliquée en traitement automatique des langues (ou du moins à l'ajustement de modèles de langue). En effet, cette méthode se base fondamentalement sur une initialisation *aléatoire* de l'algorithme de descente, ceci afin de construire un simplexe de modèles initial. En revanche, les modèles de langues pré-entraînés sont par essence initialisés de manière *déterministe*. L'intégralité de leurs capacités de transfert reposent d'ailleurs sur les représentations de données textuelles que ces modèles incorporent dans leur vecteur de paramètre durant le pré-entraînement. De plus, la méthode des sous-espace nécessite de garder en mémoire durant l'ajustement non pas un modèle, mais tous les sommets du simplexes de modèles étudiés. Cette contrainte additionnelle en complexité mémoire est probablement tout à fait gérable dans le cadre de l'ajustement d'un réseau à convolution en vision informatique. Les modèles de langue, en revanche, sont connus pour leur taille considérable pouvant très bien avoisiner la centaine de milliard de paramètres, à tel point que l'ajustement traditionnel de l'un d'entre eux constitue déjà un challenge technique considérable pour la plupart des infrastructures de calculs spécialisées. L'idée d'en ajuster non pas un, mais jusqu'à six simultanément (nombre de sommets de simplexes typiquement utilisé dans la méthode des sous-espaces), semble donc peu envisageable. En revanche, les méthodes d'ajustement par prompts continus et, par extension, les méthodes PEFT, proposent non pas d'ajuster ces modèles de langue directement, mais au contraire d'y introduire de nouveaux paramètres apprenables (à l'instar des embeddings des tokens virtuels en apprentissage de prompts continus), et d'ajuster ceux-ci tout en figeant les paramètres pré-entraînés du modèle de langue. L'avantage principal de cette approche réside dans la capacité de ces modèles "adaptés" à répliquer (voire améliorer dans des contextes associés à des tailles d'échantillons faibles) les performances des modèles de langue tout en réduisant leur nombre de paramètres apprenables de plusieurs ordres de grandeur. Ces approches nécessitent de plus typiquement une initialisation aléatoire des paramètres additionnels qu'elles introduisent dans le modèle, en faisant ainsi des candidats naturels particulièrement prometteurs pour l'adaptation de la méthode des sous-espaces aux gros modèles de langage.

Les contributions de cet article sont les suivantes. Tout d'abord, nous introduisons la première adaptation de la méthode des sous-espace aux gros modèles de langage, via d'ajustement de sous-espace de préfixes (une méthodes PEFT similaire à l'ajustement de prompts continus parmi les plus performantes dans la littérature académique actuelles). Ensuite, cet article propose d'exploiter certains avantages naturels que la méthode des sous-espaces offre afin de revisiter la notion de guidage d'ajustement d'un modèle par la métrique de validation. On montrera empiriquement que la combinaison de ces deux idées amène un gain conséquent en termes de prédiction moyenne sur les tâches de compréhension du langage naturel que propose le benchmark GLUE (Wang *et al.*, 2018).

Finalement, une étude d'ablation sera présentée pour fournir quelques éléments d'explication quant aux mécanismes permettant ce gain en termes de prédiction.

2 Méthode des sous-espaces

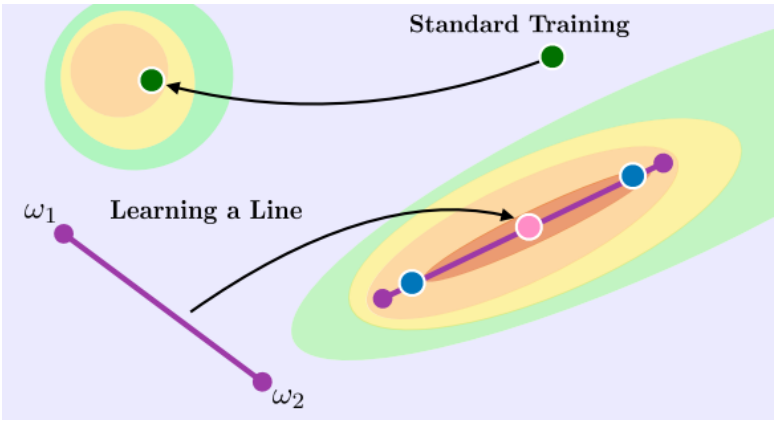

FIGURE 1 – Illustration de la méthode d'apprentissage de sous-espaces. On cherche à obtenir non pas un unique modèle, ici représenté comme un point sur la surface de la fonction objectif, mais une droite entière associée à de faibles valeurs de cette surface. Illustration tirée de (Wortsman *et al.*, 2021)

La méthode d'apprentissage de sous-espaces de réseaux neuronaux, illustrée en figure 1, permet d'obtenir en un seul processus d'ajustement une région connexe de l'espace des paramètres composée de modèles autant divers que tous associés à des performances. On choisi de définir ce domaine de modèles Λ comme un simplexe, caractérisé par ses m sommets $\{\omega_i\}_{i=1}^m$ comme l'ensemble des barycentres de ces derniers :

$$P(\alpha, \{\omega_i\}_{i=1}^m) = \Sigma_{i=1}^m \alpha_i \omega_i \; avec \; \{\alpha \in \mathbb{R}^m : \Sigma_i \alpha_i = 1, \alpha_i > 0\} \tag{1}$$

L'objectif est donc de minimiser la fonction coût choisie l pour tout paramétrage de modèle appartenant à Λ. Autrement dit, on cherche à minimiser l'espérance de l mesurée sur la distribution des données D pour tout modèle f échantillonné uniformément du simplexe Λ paramétré par α :

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim D}[\mathbb{E}_{\alpha\sim U(\Lambda)}[l(f(\mathbf{x}, P(\alpha, \{\omega_i\}_{i=1}^m)), \mathbf{y})]] \tag{2}$$

Avec :

- D la distribution des données,
- $U(\Lambda)$ la distribution uniforme sur le domaine Λ,
- $(\mathbf{x}, \mathbf{y})$ les variables d'entrées et la variable à expliquer, respectivement,
- f un modèle prédictif paramétré par un élément quelconque de D.

En pratique, l'ajustement du simplexe de modèle ne se fait pas par optimisation directe de cette entité,

mais d'une approximation stochastique échantillonnée à la fois au niveau des données (comme en descente stochastique traditionnelle) et de $U(D)$. En d'autres termes, l'initialisation du simplexe est tout d'abord typiquement réalisée via l'initialisation indépendante de chacun de ses sommets, par le biais de méthodes traditionnelles d'initialisation aléatoire de modèles neuronaux.

Par la suite, et ce pour chaque itération de descente de gradient, un modèle paramétré en θ est choisi par échantillonnage uniforme sur le simplexe, et utilisé dans le cadre d'une descente de gradient stochastique traditionnelle. Le gradient de la fonction objectif peut ensuite être propagé aux sommets du simplexe en écrivant :

$$\frac{\partial l}{\partial \omega_i} = \frac{\partial l}{\partial \theta} \frac{\partial P(\alpha, \{\omega_i\}_{i=1}^m)}{\partial \theta} \tag{3}$$

Dans le but de garantir une certaine diversité fonctionnelle au sein du simplexe de modèles, un terme de régularisation est ajouté à l'objectif, encourageant la dissimilarité (au sens de la similitude cosine) entre les différents sommets du simplexe :

$$\beta.\mathbb{E}_{j\neq k}[cos^2(\omega_j, \omega_k)] = \beta.\mathbb{E}_{j\neq k}[\frac{\langle \omega_j, \omega_k \rangle^2}{\|\omega_j\|_2^2 \|\omega_k\|_2^2}] \tag{4}$$

L'intensité β de ce terme de régularisation constitue un hyperparamètre au modèle, qu'on fixe à la valeur recommandée par défaut de $\beta = 1$ (Wortsman *et al.*, 2021).

On dispose donc, après descente de gradient, d'un simplexe entier de modèles que l'on peut échantillonner à volonté pour inférence. Le centroïde du simplexe, en particulier, présente typiquement des capacités de généralisation supérieures à celles de modèles obtenus par ajustement traditionnel.

Une possible justification de cette propriété, visualisée en figure 2, réside dans l'idée qu'un modèle obtenu par ajustement traditionnel serait localisé à la périphérie d'un minimum local de la fonction objectif, typiquement plus sensible à des erreurs de généralisation (Izmailov *et al.*, 2018; Dziugaite & Roy, 2018). Parcourir le sous-espace permet au contraire de "traverser" le minimum local, afin d'obtenir un modèle associé à une zone plus stable de la fonction objectif.

Jusqu'ici, la définition de la procédure d'ajustement ne dépend aucunement de l'aspect connectionniste des réseaux neuronaux, et considère simplement un modèle comme un vecteur de paramètres apprenables. L'immense majorité des modèles d'apprentissage profond, en revanche, sont définis par une succession de transformations non linéaires. Il semble donc naturel d'incorporer, d'une manière ou d'une autre, cette structure séquentielle des modèles neuronaux dans la définition de la procédure d'ajustement. Pour ce faire, il est conseillé d'échantillonner les paramètres de chaque couche du modèle indépendamment. Cette variante dite "couche par couche" de la méthode est typiquement associée à de meilleures performances prédictives.

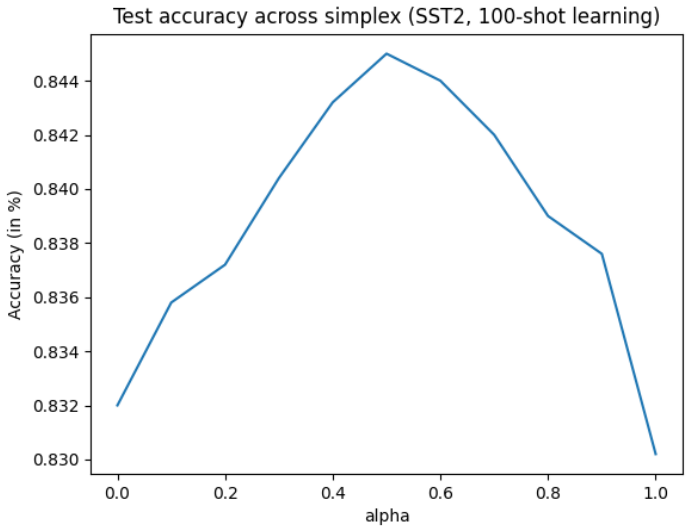

FIGURE 2 – Évolution des performances prédictives d'un modèle de langue ajusté sur une droite de préfixes. Les performances de généralisation suivent une courbe similaire à une parabole maximale en son centre

3 Méthode proposée

3.1 Apprentissage de sous-espaces de préfixes

Comme mentionné en introduction, la méthode des sous-espaces est en pratique difficilement applicable dans le cadre de l'ajustement classique de modèles de langue. Les méthodes de prompts, en revanche, reposent sur l'apprentissage non pas du modèle lui-même, mais sur l'ajustement de n vecteurs d'embeddings $\{E_i\}_{i=1}^{n}$ concaténés typiquement au début de la séquence d'embeddings d'entrée du modèle de langue LM_Φ paramétré en Φ. En d'autres termes, pour une séquence d'entrées de L tokens, $\{I_i\}_{i=1}^{l}$ on construit un modèle prédictif à partir non pas de la sortie du modèle de langage :

$$LM_\Phi(\{I_i\}_{i=1}^{l}) \tag{5}$$

mais de

$$LM_\Phi(concat(\{E_i\}_{i=1}^{n}, \{I_i\}_{i=1}^{l})) \tag{6}$$

L'ajustement du modèle prédictif se fait uniquement par ajustement des tokens virtuels $(E_i)_{i=1}^{n}$, tout en figeant les paramètres du modèle de langue Φ.

Afin d'augmenter l'expressivité de cette méthode (particulièrement limitée en terme de nombre de paramètres apprenables), la méthode d'apprentissage de préfixes (Li & Liang, 2021), choisie dans cet article comme candidat à l'application de la méthode des sous-espaces, propose de concaténer ces tokens virtuels non pas à la séquence d'entrée du modèle, mais aux séquences Key $\{K_i\}_{i=1}^{l}$ et Values $\{V_i\}_{i=1}^{l}$ utilisées en entrée des modules d'attention multiplicative présents dans chaque couche du modèle de langue. En d'autre terme, la sortie $\{H_i\}_{i=1}^{l}$ d'un module d'attention Att d'une couche de modèle de langage, initialement définie comme :

$$\{H_i\}_{i=1}^{l} = Att_{\Psi}(\{K_i\}_{i=1}^{l}; \{Q_i\}_{i=1}^{l}; \{V_i\}_{i=1}^{l}) \tag{7}$$

est définie dans le cadre de l'apprentissage de préfixes comme :

$$\{H_i\}_{i=1}^{l} = Att_{\Psi}(concat(\{E_i^k\}_{i=1}^{n}, \{K_i\}_{i=1}^{l}); \{Q_i\}_{i=1}^{l}; concat(\{E_i^v\}_{i=1}^{n}, \{V_i\}_{i=1}^{l})) \tag{8}$$

Avec :

- $\{H_i\}_{i=1}^{l}$ la sortie du module d'attention
- Att le module d'attention multiplicative d'une couche de modèle de langage
- $\{K_i\}_{i=1}^{l}$ la séquences Keys d'entrées au module d'attention (obtenue par transformation linéaire de la séquence d'entrée de la couche du modèle de langue)
- $\{V_i\}_{i=1}^{l}$ la séquences Values d'entrées au module d'attention (obtenue par transformation linéaire de la séquence d'entrée de la couche du modèle de langue)
- $\{K_i\}_{i=1}^{l}$ la séquences Queries d'entrées au module d'attention (obtenue par transformation linéaire de la séquence d'entrée de la couche du modèle de langue)
- $\{E_i^k\}_{i=1}^{n}$ les vecteurs d'embeddings apprenables concaténé à la séquence Keys
- $\{E_i^v\}_{i=1}^{n}$ les vecteurs d'embeddings apprenables concaténé à la séquence Values

Dans une approche similaire à la méthode de prompts, l'ajustement de préfixes se fait uniquement par ajustement (via descente de gradient) des tokens virtuels $\{E_i^k\}_{i=1}^{n}$ et $\{E_i^v\}_{i=1}^{n}$, tout en laissant les paramètres du modèle de langage lui même figés. Apprendre directement ces embeddings s'avère en revanche particulièrement instable. Aussi, il est coutume de pas les ajuster directement, mais d'utiliser une astuce de reparamétrisation, consistant à concaténer aux séquences Keys et Values non pas les séquences de préfixes directement, mais une transformation de ces derniers, paramétrée par un perceptron sous complets à deux couches, comme illustré en figure 3-1.

L'adaptation de la méthode des sous-espaces à l'ajustement de préfixes peut donc se faire par deux approches distinctes :

1. Application aux paramètres apprenables du modèle eux même, et donc à l'embedding initial et au perceptron sous-complet de reparamétrisation
2. Application de la méthode aux préfixes eux même, et donc à la sortie du module de reparamétrisation

On propose dans cet article d'étudier l'option 2 la méthode des sous-espaces, et donc d'appliquer la méthode de sous-espaces directement aux préfixes $\mathbf{E}^k$ et $\mathbf{E}^v$, comme illustré en figure 3.2.

Il convient également de s'intéresser à l'adaptation de la variante "couche par couche" de la méthode. En effet, l'ajustement de préfixe ne repose pas sur l'introduction dans le modèle de langue d'une structure classique de perceptron, mais d'une modification de l'opération du module d'attention multitête de ce dernier. Nous proposons dans cet article d'étendre cette variante "couche par couche" à $\mathbf{E}^k$ et $\mathbf{E}^v$. Ainsi, à chaque itération de descente durant l'ajustement, les $\mathbf{E}^k$ et $\mathbf{E}^v$ de chaque couches seront tous échantillonnés indépendamment. De plus, cet échantillonnage sera effectué indépendamment pour toutes les observations, contrairement à l'approche traditionnelle qui préfère créer un unique modèle par itération de descente.

Additionnellement, la tête de prédiction du modèle est typiquement initialisée aléatoirement. On choisit donc de lui appliquer également la méthode des sous-espaces, comme décrite en partie 2. Par

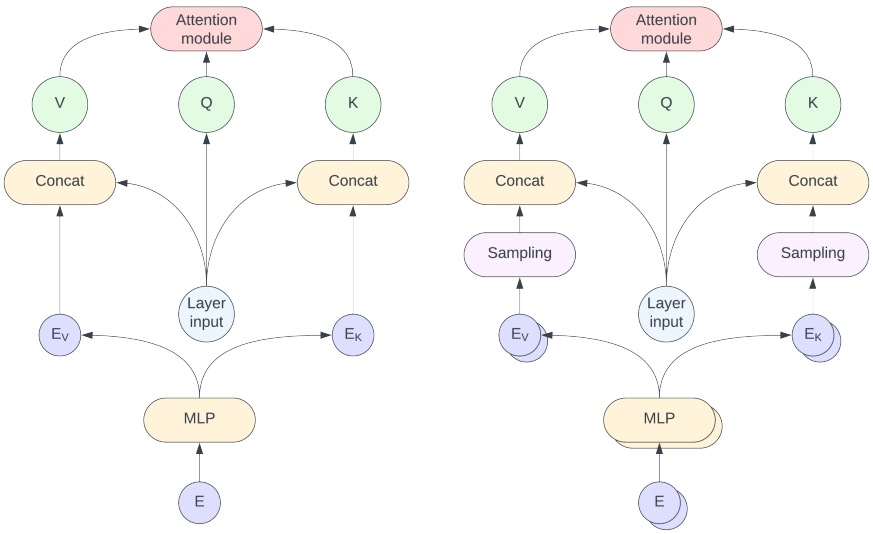

FIGURE 3 – 3.1 (gauche) : Méthode d'apprentissage de préfixe. Un premier embedding est utilisée comme entrée d'un perceptron à deux couches, dont la sortie est concaténée aux séquences de Keys et Values des modules d'attention multi-tête du modèle de langue. Les paramètre apprenables du modèles sont l'embedding initial et le perceptron de reparamétrisation. 3.2 (droite) : Proposition d'extension à l'apprentissage de sous-espaces de préfixes. Chaque sommet du simplexe est calculé indépendamment, et l'échantillonnage aléatoire est effectué dans le simplexe de préfixes (par opposition au simplexe de l'embedding et du perceptron de reparamétrisation)

souci de cohérence, la variante d'échantillonnage de paramètres à l'échelle de l'observation, et non de la batch, lui sera en revanche également appliquée.

En résumé, les paramètres ajustables du modèle prédictif dans un cadre d'apprentissage de préfixe sont les suivant :

En résumé, nous proposons d'ajuster un simplexe à m sommets de préfixes de la manière suivante :

- Initialisation indépendante de m systèmes de reparamétrisation
- Calcul, pour chaque itération de descente, des m sommets du simplexe
- Construction des préfixes utilisés pour l'inférence de la fonction coût et le calcul de son gradient par échantillonnage uniforme indépendamment pour chaque observation, chaque couche, ainsi que pour les préfixes des séquences Keys et Values

3.2 Méthode des sous-espaces et inférence stochastique de métriques de validation

L'ajustement d'un gros modèle de langue est typiquement guidé par l'estimation d'une métrique de performance sur un jeu de validation, ceci autant durant la recherche d'hyperparamètres, que durant le processus de descente même, où le meilleur modèle au sens de cette valeur scalaire est retenue comme modèle final. La question de l'estimation de cette métrique dans un cadre d'apprentissage de sous-espace soulève des questions. En effet, l'ajustement non pas d'un seul modèle, mais tout un

simplexe, résulte en autant de métriques de validation potentiellement estimables.

Puisque l'on se limite dans le cadre de cet article à l'utilisation de cette méthode afin d'en extraire le centroïde -associé à de meilleures performances de généralisation- il serait naturel de l'estimer vis-à-vis de ce dernier. Cependant, l'existence pour l'inférence non pas d'un unique modèle, mais de ce simplexe, et notamment des informations supplémentaires qu'il apporte quant à la nature du minimum local obtenu, peut s'avérer intéressante. C'est notamment le cas dans un contexte de *few-shot learning*. En effet, comme mentionné précédemment, pour des jeux de données de validation associés à des tailles d'échantillon faibles (< 100 typiquement), l'estimation de cette métrique peut devenir déraisonnablement bruité. Cette propriété a deux effets indésirables sur le processus d'élaboration du modèle :

— La recherche d'hyperparamètres perd sensiblement de son intérêt, puisque la valeur guidant la sélection de modèles n'est plus informative

— Plus important encore, il devient particulièrement difficile durant l'ajustement d'identifier le surajustement de modèle. Les méthodes d'arrêt précoces, pourtant particulièrement importante dans un cadre de *few-shot learning* (par exemple pour limiter le surajustement), perdent donc également en efficacité

On propose donc d'utiliser le simplexe dans son ensemble pour "augmenter" l'estimation de la métrique de validation. Celle-ci se fera donc non pas en utilisant le centroïde du simplexe, mais avec *plusieurs* modèles échantillonnés aléatoirement pour chaque individu du jeu de validation. En d'autre termes, on présentera à chaque estimation de métrique le jeu de validation n fois, et le processus d'inférence se fera dans les mêmes conditions que pendant l'ajustement, c'est-à-dire avec échantillonnage aléatoire du modèle utilisé pour chaque observation. On fixe le nombre n de répétitions du jeu de données de validation à 10 dans toutes les expériences présentées dans cet article qui utilise cette approche d'inférence stochastique.

On sélectionne tout de même en guise de modèle final le centroïde du simplexe. En effet, le déterminisme d'un modèle reste une propriété désirable dans des situations de production.

4 Expériences

Toutes les expériences décrites dans cet article afin d'évaluer les performances prédictives de la méthode proposée sont réalisées avec BERT-base-cased sur des jeux de données construits à partir du benchmark GLUE (Wang *et al.*, 2018), un corpus de 8 tâches de compréhension du langage anglais, toutes formulées comme des problèmes de classification. Cependant, de part l'inaccessibilité des jeu de test sur ce benchmark ainsi que leur taille d'échantillon sensiblement trop importante pour être pertinente dans un cadre de "few-shot learning", nous n'utilisons pas directement ces jeux de données, mais en construisons de nouveaux plus adaptés à notre problématique suivant une méthodologie similaire à celle présentée dans (Mao *et al.*, 2022). Nous construisons par échantillonnage aléatoire des séries de jeux de données de tailles d'échantillon variable (50, 100, 200 et 500 observations). Notre méthode de construction de ces jeux de données diffère cependant de la leur sur quelques points clés.

Premièrement, les auteurs ont choisi de construire des jeux de validation de 1 000 d'observations pour tous leurs jeux de données d'apprentissage, ce qui, de notre avis, n'est pas réaliste dans un contexte de "few-shot learning" (un jeu de validation ne contient généralement pas dix fois plus d'exemple que son jeu d'entraînement). Deuxièmement, ils utilisent les jeux de validation du benchmark GLUE comme

jeux de test. Toutefois, certains de jeux validation ont des tailles d'échantillon faibles (277 pour RTE, 408 pour MRPC), pouvant potentiellement bruiter l'estimation des métriques de performances. Par conséquent, pour chaque jeu de données de référence, un ensemble de données de taille d'échantillon K est construit comme suit :

— Les jeux d'entraînement et de validation sont concaténés en un unique jeu
— La moitié des observations (plafonnée à 5000 observations) sont exclues de ce jeu de données pour construire un jeu de données de test, commun à toutes les expériences
— K observations sont ensuites sélectionnées par échantillonnage uniforme, et répartie dans un jeu d'entraînement et de validation suivant des proportions 70/30

Pour chaque tâche du benchmark, et pour chacune des tailles d'échantillon retenues, 10 jeux de données sont construits en suivant cette méthodologie, afin de permettre de répliquer les expériences sur différents jeux de données, de permettre l'estimation de performances moyennes, et de tester la significativité au seuil des 5% des différences obtenues (via bootstrap).

Pour tous ces jeux de données, nous comparons notre méthode à 5 modèles d'ajustement de référence, dont 4 méthodes PEFT. Ces méthodes, de manière similaires à la méthode d'apprentissage de préfixes, se basent sur l'idée de figer les paramètres du modèle de langue, et d'introduire une fraction de nouveaux paramètres ajustables (typiquement d'une cardinalité inférieure de plusieurs ordres de grandeur à celle du modèle lui même), mais elles diffèrent dans la manière dont elles introduisent ces nouveaux paramètres dans le modèle :

— Par ajustement traditionnel (Devlin *et al.*, 2019), où l'intégralité des paramètres du modèle de langue sont ajustés par descente de gradient. Cette méthode reste parmi les plus usitées en traitement automatique des langues, et représente donc une référence essentielle à laquelle comparer la méthode que nous proposons
— Par adapteur standard (Houlsby *et al.*, 2019), qui proposent typiquement d'introduire un ou plusieurs perceptron à deux couches "bottleneck" à différents étages d'une couche de Transformer, première méthode PEFT a avoir été introduite, et la plus reconnue
— Par Low Rank Adaption (LORA) (Hu *et al.*, 2021), qui reparamétrise les matrices de projections des Values et Queries précédent le module d'attention multiplicative multitête par via deux perceptrons sous complets linéaires à deux couches, première méthode PEFT à proposer différentes transformations pour différents éléments du module d'attention
— Par méthode UniPELT (Mao *et al.*, 2022) méthode de fusion combinant adapteurs, LoRA et préfixes afin de bénéficier des avantages de chacunes (sans souffrir de leurs potentiels inconvénients respectifs)
— Par apprentissage de préfixes classique, méthode de référence cruciale afin d'estimer la part des performances de la méthode proposée qui lui est réellement attribuable

Pour toutes les méthodes, nous suivons la même procédure d'apprentissage et de recherche d'hyperparamètres que proposée par (Mao *et al.*, 2022). Tous les modèles ont été ajustés pendant 50 epochs en utilisant les réglages d'usine du Trainer Huggingface, ainsi qu'un mécanisme d'arrêt anticipé munie d'une patience de 10 epochs. La taille de batch est fixée à 16 pour toutes les expériences, et les recherches d'hyperparamètres sont effectuées par recherche exhaustive dans les valeurs suivantes :

— Ajustement traditionnel : Taux de descente parmi $[1e-5, 2e-5]$
— Adapteur standard : Taux de descente de $1e-4$ et taux de réduction parmi $[3, 6, 12]$
— LoRA : Rang et valeur alpha fixés à 8, taux de descente parmi $[1e-4, 5e-4]$
— UniPELT : Longueur de préfixe fixée à 10, adapteur à taux de réduction fixé à 16, et LoRA

avec range et valeur alpha fixés à 8. Taux de descente parmi $[2e-4, 5e-4]$

— Apprentissage de préfixe : Longueur de préfixe fixée à 50, taux de descente parmi $[1e-4, 2e-4, 5e-4]$

Pour garantir une comparabilité optimale, le choix des hyperparamètres de la méthode proposée seront choisis de manière à correspondre exactement à ceux de la baseline d'apprentissage par préfixe, eux aussi déterminés pour la première fois dans un cadre de classification de texte par (Mao *et al.*, 2022). Les sous-espaces ajustés dans les expériences sont tous des simplexes à 6 sommets.

5 Résultats

Method	MNLI	QNLI	SST-2	QQP	CoLA	STS-B	MRPC	RTE	Avg.
[*K = 50]*									
Ajustement classique	35.5	65.9	57.57$_*$	45.6$_*$	**3.5**	45.1$_*$	81.1	50.6	48.1
Adapteur	35.6	62.6$_*$	64.7$_*$	35.3$_*$	0.0	59.7	80.2	**53.1***	48.9
LoRA	35.7	63.9$_*$	68.4$_*$	47$_*$	1.0	56.5$_*$	**81.4**	52.8	50.8
UniPELT	35.3	62.4$_*$	73.1$_*$	42.3$_*$	1.1	**64.3***	80.7*	51.8	51.4
Préfixes	**37.8**	63.5$_*$	74.9$_*$	53.1	1.8	59.2$_*$	80.4*	52.6	52.9
Préfixes (sous-espace)	36.6	**66.6**	**80.1**	**54.3**	0.8	61.1	80.0	52.2	**54.0**
[*K = 100]*									
Ajustement classique	35.5$_*$	68.9$_*$	73.9$_*$	52.6$_*$	3.0$_*$	64.1$_*$	81.3	52.1	53.9
Adapteur	36.3$_*$	66.7$_*$	72.8$_*$	54.0$_*$	7.2	63.8$_*$	80.5	53.0	54.3
LoRA	37.3	64.9$_*$	73.2$_*$	54.2$_*$	7.3	60.4$_*$	81.3	52.9	53.9
UniPELT	37.7	66.9$_*$	79.1$_*$	53.6$_*$	5.1	68.4	79.7$_*$	52.0$_*$	55.3
Préfixes	38.3	69.4$_*$	80.8$_*$	57.2	**8.1**	66.6$_*$	81.1	**54.2**	57.0
Préfixes (sous-espace)	**38.5**	**70.8**	**82.5**	**59.6**	7.8	**68.3**	**81.5**	54.1	**57.9**
[*K = 200]*									
Ajustement classique	42.3	**71.9**	80.8$_*$	63.0	20.2	69.0$_*$	80.8	54.6	60.3
Adapteur	42.7	69.1$_*$	83.1$_*$	59.5$_*$	**26.5***	70.3$_*$	80.7	**56.2**	61.0
LoRA	41.0	67.1$_*$	82.2$_*$	61.2$_*$	19.8	67.8$_*$	80.1	54.5	59.2
UniPELT	41.6	70.2	82.8$_*$	58.7$_*$	16.4	**72.8**	**81.7**	54.9	59.9
Préfixes	**44.9**	71.4	**84.2**	63.0$_*$	22.2	71.3	79.6$_*$	56.0	61.6
Préfixes (sous-espace)	44.7	71.2	84.1	**64.4**	21.1	72.3	81.6	**55.9**	**61.9**
[*K = 500]*									
Ajustement classique	52.7$_*$	74.3$_*$	85.4$_*$	66.8	32.2$_*$	78.0	82.5	59.8	66.5
Adapteur	51.1$_*$	72.4$_*$	85.4$_*$	65.7$_*$	**38.9***	76.1$_*$	81.9$_*$	59.8	66.4
LoRA	50.1$_*$	73.6$_*$	84.6$_*$	66.5	35.3	75.6$_*$	82.3$_*$	58.3$_*$	65.8
UniPELT	50.7$_*$	74.2$_*$	85.4$_*$	63.4$_*$	34.2	77.2	82.1	57.8$_*$	65.6
Préfixes	54.0$_*$	74.7$_*$	85.6$_*$	66.2	35.7	77.8	82$_*$	60	67.0
Préfixes (sous-espace)	**55.7**	**75.4**	**86.1**	**67.2**	36.0	**78.1**	**83.1**	**60.8**	**67.8**

TABLE 1 – Résultats de l'expérience. Des mesure F1 sont reportées pour QQP et MRPC. Une corrélation de Spearman est reportée pour STS-B. Une corrélation de Matthews pour CoLA. Des mesures d'accuracy sont reportées pour le reste des tâches. Les résultats en gras et soulignés correspondent aux premières et secondes meilleures performances, respectivement. Les résultats suivis d'une astérisque en indice ou exposant correspondent à des résultats significativement supérieurs ou inférieurs à ceux de la méthode proposée, respectivement

Les performances de toutes les méthodes PEFT sélectionnées ainsi que de l'approche proposée sont

présentées dans la Table 1, ceci pour toutes les différentes tâches du benchmark GLUE, et pour les différentes tailles d'échantillons retenues. Dans l'ensemble, l'apprentissage de sous-espace de préfixes surpasse toutes les autres méthodes de base en moyenne, ceci pour toutes les tailles d'échantillons. La méthode présente notamment un gain d'un point en comparaison à l'apprentissage de préfixes classique, pour des tailles d'échantillons de 50 et 100 observations. Ce gain diminue mais reste présent lorsque la taille des jeux de données échantillonnés augmente, ce qui n'est pas nécessairement surprenant, l'apprentissage de sous-espace améliorant principalement la capacité de généralisation du modèle final.

La comparaison entre la méthode par apprentissage de sous-espace de préfixes surpasse l'apprentissage de préfixes traditionnel est particulièrement intéressante. En effet, les deux approches reposent essentiellement sur le même formalisme. En terme de significativité statistique, la méthode proposée surpasse son équivalent classique 12 fois :

— Sur QNLI, SST-2, et STS-B pour $K = 50$ et $K = 100$
— Sur MRPC et QQP pour $K = 200$
— Sur MNLI, QNLI, MRPC et STS-B pour $K = 500$

Elle est en revanche surpassée statistiquement une unique fois, sur MRPC pour $K = 50$, ce qui est d'autant plus surprenant quand on remarque que la différence entre les deux méthodes sur cette expérience est de 0.4%. De plus, la méthode proposée redevient significativement supérieure sur cette tâche une fois que la taille d'échantillon augmente jusqu'à 500 observations.

Plus largement, la méthode proposée n'est surpassé significativement que 6 fois sur toutes les expériences :

— Sur MRPC par les méthodes de préfixe et LoRA pour $K = 50$
— Sur RTE par la méthode d'Adapteur pour $K = 50$
— Sur STS-B par la méthode UniPELT pour $K = 50$
— Sur CoLA par la méthode d'Adapteur pour $K = 200$ et $K = 500$

On remarquera notamment que tous ces événements s'observent pour $K = 50$ (et donc de jeux de validations de 15 observations), où l'ajustement de modèles devient particulièrement complexes.

La méthode proposée surpasse en revanche significativement l'une des autres méthodes de base à travers les expériences effectuées un total de 80 fois, montrant un clair avantage en termes de pouvoir prédictif.

On remarque notamment que la majorité des expériences où la méthode proposée surpasse les méthodes de référence principalement sur 3 jeux de données, à savoir QNLI, SST-2 et QQP. De plus, la capacité de la méthode proposée à surpasser significativement les méthodes de référence sur ces tâches ne semble pas dépendre de la taille d'échantillon des jeux de données.

Il est en revanche difficile d'identifier ce qui différencie ces jeux de données de ceux où la méthode proposée reste comparable aux méthodes de références. En effet, les deux camps présentent autant de tâches similaires, et autant de jeux de données déséquilibrés.

6 Étude d'ablation

De manière à mieux identifier l'impact des différents aspects de la méthode proposées, une étude d'ablation est également rapportée en table 2, avec les variantes suivantes :

— Même méthode avec des simplexes à 2 sommets (ie une ligne)

— Même méthode sans l'inférence de validation stochastique
— Même méthode sans sous-espaces des têtes de prédiction
— Même méthode sans sous-espace de préfixe (donc uniquement sur les têtes de prédictions)

Method	$K = 50$	$K = 100$	$K = 200$	$K = 500$
Méthode proposée	54.0	57.9	61.9	67.8
Simplexe à 2 sommets	53.6	57.9	62.1	67.6
Validation déterministe	49.5	56.0	61.4	67.8
Sans sous-espace de tête	53.5	56.8	61.3	67.2
Sans sous-espace de préfixes	52.6	57.7	62.2	67.6

TABLE 2 – Résultats de l'étude d'ablation. Les scores rapportés correspondent à la moyenne des performances prédictives sur toutes les tâche du benchmark GLUE

Les résultats de cette étude d'ablation peuvent être résumés comme suit :

1. L'utilisation de simplexes à deux sommets montrent des performances légèrement inférieures à la méthode proposées pour $K = 50$, puis des performances similaires par la suite
2. L'utilisation de la méthode de sous-espace guidée par une métrique de validation estimée de manière déterministe s'effondre pour $K = 50$, $K = 100$ et $K = 200$ (cas pour lesquels les performances sont d'ailleurs inférieures à la méthode d'ajustement de préfixes classique), puis finissent par devenir équivalents à la méthode proposée
3. L'utilisation de sous-espaces de préfixes, sans sous-espace de tête, est surpassée avec consistance par la méthode proposée
4. L'utilisation de sous-espace de tête de prédiction, couplée à des préfixes classique, est surpassée de manière considérable pour $K = 50$ (cas où cette approche est moins performante que la méthode d'ajustement de préfixes classique), et similaire à la méthode proposée lorsque la taille d'échantillon augmente

Ces observations, prises dans leur ensemble, amènent notamment plusieurs éléments de preuves quant à la pertinence de l'utilisation en "few-shot learning" de la notion que nous proposons d'inférence stochastique de métrique de validation. L'observation 3, en particulier, montre que l'ajustement de sous-espace de préfixes avec estimation classique de la métrique de validation n'est associé aux mêmes gains de performances que la méthode proposée *qu'à partir* de $K = 500$. Que les performances de cette variante soient de plus inférieures à celles obtenues par ajustement de préfixes classiques vient encore appuyer l'importance de la méthode proposée d'inférence stochastique de métrique de validation.

Les observation 1, 2, 3 permettent par la suite d'amener des arguments légèrement plus faibles sur l'importance de la taille du simplexe dans le cadre de cette estimation stochastique. En effet, bien que la taille du simplexe ne semble pas avoir d'effet pour $K > 50$ (indiquant fortement qu'il est préférable d'apprendre des lignes pour ces tailles d'échantillons, considérablement plus économes en terme de complexité mémoire), elle semble en avoir un pour des tailles d'échantillon très faible. Cette observation pourrait s'expliquer par la richesse de l'information extraites du jeu de données de validation par estimation stochastique, dûe à un simplexe plus important. Les résultats présentés dans cet article sont en revanche insuffisants pour confirmer (ou infirmer) cette hypothèse.

Similairement, les observations 2 et 3 montrent tout particulièrement l'importance d'ajuster par sous-espace l'intégralité des paramètres apprenables du modèle dans le cas où $K = 50$. Ceci pourrait également s'expliquer en avançant l'idée que l'on perd en capacité à caractériser le minimum local obtenu en limitant l'estimation stochastique de la métrique de validation à une sous partie des paramètres apprenables du modèle.

7 Conclusion

On a dans cet article introduit deux idées novatrices. La première, une adaptation de la méthode des sous-espaces à l'ajustement de gros modèles de langues par le biais de méthode PEFT, est à notre connaissance le premier exemple d'utilisation de cette méthode dans la littérature académique portant sur le traitement automatique des langues. La seconde, proposant une manière alternative d'estimation des métriques de validation, constitue une application originale de la méthode des sous-espaces et n'est en aucun cas spécifique à des problématiques rencontrées en analyse de données textuelles. L'utilisation jointe de ces deux méthodes donne lieu à une augmentation considérables des performances de modèles de langues communs comme BERT, sur les tâches de compréhension du langage proposées par le benchmark GLUE reformulées dans un contexte de "few-shot learning". L'étude d'ablation présentée en fin d'article permet en outre de poser des hypothèses quant à l'impact de ces deux contributions. Le gain de performances observés sur très petits jeux de données (≤ 100) semble être en effet principalement expliqué par l'information plus fine extraite du jeu de validation via la méthode d'estimation stochastique de métrique. Ce gain semble en revanche se dissiper pour des tailles d'échantillon plus élevées, ou la méthode des sous-espaces appliquées à l'apprentissage de préfixes semble se suffire à elle-même pour permettre un gain de performance sur les méthodes PEFT, ainsi que sur l'ajustement de modèle classique.

L'application de la méthode des sous-espaces aux méthodes PEFT permet en outre l'ajustement de puissants modèles prédictifs tout en réduisant considérablement les ressources machines typiquement nécessaires à l'ajustement des gros modèles de langues, ne dénaturant pas ainsi la volonté fondamentale d'*efficience* de ces méthodes. L'apprentissage de sous-espaces de préfixes reste ainsi accessible même dans des situations où les ressources, autant en données qu'en puissance de calcul, sont limitées.

Remerciements

Nous tenons à remercier le Sorbonne Center for Artificial Intelligence pour le financement du contrat post-doctoral de Louis Falissard au sein du laboratoire MLIA de l'Institut des Systèmes Intelligents et de Robotique.

Références

BAPNA A. & FIRAT O. (2019). Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), p. 1538–1548, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1165.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DZIUGAITE G. K. & ROY D. (2018). Entropy-SGD optimizes the prior of a PAC-Bayes bound : Generalization properties of entropy-SGD and data-dependent priors. In J. DY & A. KRAUSE, Éds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 de *Proceedings of Machine Learning Research*, p. 1377–1386 : PMLR.

HOULSBY N., GIURGIU A., JASTRZEBSKI S., MORRONE B., DE LAROUSSILHE Q., GESMUNDO A., ATTARIYAN M. & GELLY S. (2019). Parameter-efficient transfer learning for NLP. In K. CHAUDHURI & R. SALAKHUTDINOV, Éds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 de *Proceedings of Machine Learning Research*, p. 2790–2799 : PMLR.

HU E. J., SHEN Y., WALLIS P., ALLEN-ZHU Z., LI Y., WANG S., WANG L. & CHEN W. (2021). Lora : Low-rank adaptation of large language models. DOI : 10.48550/ARXIV.2106.09685.

IZMAILOV P., PODOPRIKHIN D., GARIPOV T., VETROV D. & WILSON A. G. (2018). Averaging weights leads to wider optima and better generalization. DOI : 10.48550/ARXIV.1803.05407.

LI X. L. & LIANG P. (2021). Prefix-tuning : Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 4582–4597, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.353.

MAO Y., MATHIAS L., HOU R., ALMAHAIRI A., MA H., HAN J., YIH S. & KHABSA M. (2022). UniPELT : A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 6253–6264, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.433.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language Models are Unsupervised Multitask Learners.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. DOI : 10.48550/ARXIV.1910.10683.

WANG A., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. (2018). GLUE : A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 353–355, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5446.

WORTSMAN M., HORTON M. C., GUESTRIN C., FARHADI A. & RASTEGARI M. (2021). Learning neural network subspaces. In M. MEILA & T. ZHANG, Éds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 de *Proceedings of Machine Learning Research*, p. 11217–11227 : PMLR.