Récapitulatif sur l’intelligence artificielle employée pendant le stage

Victor Dubus-Chanson

Avec l’aide de Jean-Luc Feugeas et Morad Ben Tayeb

Table des matières

IA initiale

Préparation des données

La base de données est séparée en deux parties : les profils 1D (profils sur une dimension) et les scalaires associés aux profils. 80% des données seront utilisées pour l'entraînement du modèle, et 20% pour la validation de cet entraînement. Cette validation permettra d'éviter un surapprentissage et donc une meilleure généralisation.

Les données sont réduites par un facteur identique avant d'être utilisées. Un filtrage est appliqué aux scalaires : ceux ne respectant pas les conditions imposées sont éliminés de la base de données, tout comme leur profil associé.

Architecture du modèle

Le modèle est séparé en deux modules : l'auto-encodeur variationnel ou VAE, et le perceptron multicouche ou MLP.

Le VAE sert à reconstruire les profils, en passant par une espace latent de dimension réduite suivant une distribution normale, permettant de conserver une cohérence entre les points vus par le modèle, et des points non observés. Cela permettra de générer de nouvelles données suivant les mêmes propriétés que les données d'entraînement. L'encodeur et le décodeur sont des CNN (réseaux de neurones convolutifs).

Le MLP sert à prédire, à partir de l'espace latent, les scalaires associés.

Entraînement

L'entraînement se fait par descente de gradient stochastique par batch : les données sont séparés en batch aléatoirement formés à chaque epoch (ou cycle de l'entraînement), puis la descente de gradient se fait pour chaque batch à la suite.

Le VAE est entraîné à l'aide de la combinaison pondérée des coût de reconstruction et coût de Kullback-Leibler pour le suivi de la distribution. Cette pondération donne en général un poids plus élevé à la reconstruction qu'au suivi de la distribution, car on veut que les données générées ressemblent aux données d'entraînement. Le MLP est entraîné à l'aide de la fonction MSE (erreur quadratique moyenne).

Le MLP est entraîné après que l'entraînement du VAE soit complété.

Modifications de l'IA

Préparation des données

La base de données n'est pas séparée en plusieurs parties, chaque vecteur est la concaténation de plusieurs profils 1D (par exemple, le pas et le rayon pour des cibles hélicoïdales) et des scalaires associés à ces profils. Chaque élément de ce vecteur a une position précise connue, ainsi dans notre code, il est possible de récupérer chaque morceau des vecteurs au choix.

Chaque partie de chaque vecteur est standardisée. Pour chaque profil et chaque scalaire, la moyenne et la déviation standard sont calculées (pour un profil, les moyenne et déviation standard sont uniques pour l'ensemble des points), puis l'on applique la formule suivante à chaque point concerné :

$$x_{std} = \frac{x - E}{\sigma}$$

où E et σ sont respectivement la moyenne et la déviation standard liées au point x.

Les données sont toujours séparées en une partie destinée à l'entraînement d'un modèle et une partie servant à la validation de l'entraînement, en proportion 80/20.

Un filtrage est appliqué aux scalaires : ceux ne respectant pas les conditions imposées sont éliminés de la base de données, tout comme leur profil associé.

Architectures des modèles

Plusieurs variations du modèle de base ont menés à l'architecture finale :

1. **Incorporation des scalaires dans l'espace latent :** les scalaires ne sont plus prédits à partir de l'espace latent déjà formé, mais sont reconstruits, comme les profils, dans le VAE. D'abord en passant uniquement dans l'espace latent, puis dans les encodeur et décodeur, puis avec un couple encodeur-décodeur dédié. Pour ce faire, le vecteur associé aux scalaires est concaténé avec le vecteur associé aux profils avant le passage dans l'espace latent. Le module de l'espace latent est composé d'une couche probabiliste créant la distribution des données, ainsi que d'une couche de neurones dense avant et après cette couche probabiliste.

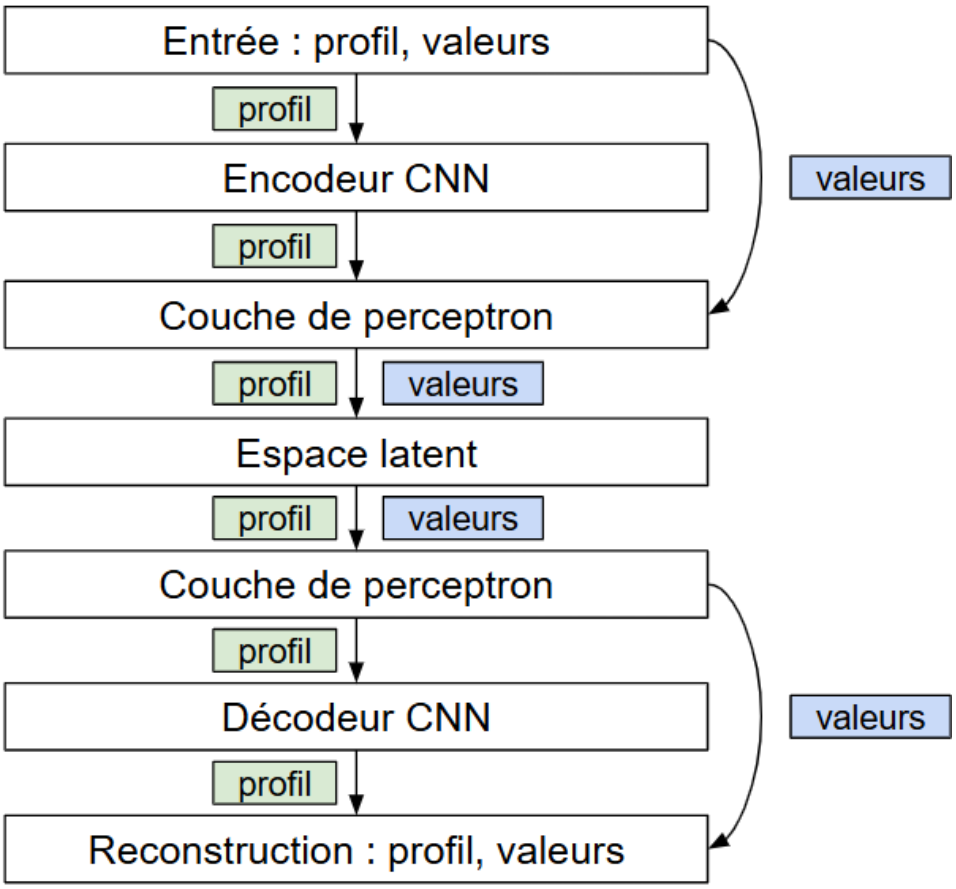

Figure 1 : structure du VAE avec incorporation des scalaires (valeurs) dans la reconstruction.

2. **Double encodeur, double décodeur :** afin de séparer les données présentant une cohérence spatiale (les profils) de celles n'en ayant pas (les scalaires), deux encodeurs distincts, ainsi que deux décodeurs sont utilisés. Un couple encodeur-décodeur est utilisé pour reconstruire les profils, ce sont des CNN. Le second est utilisé pour reconstruire les scalaires, ce sont des MLP.

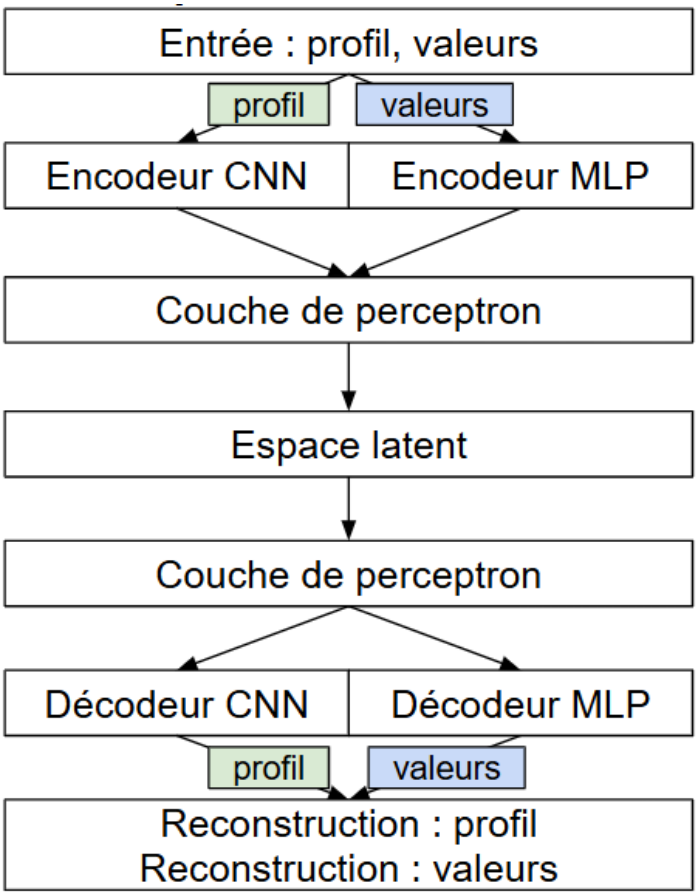

Figure 2 : structure du VAE avec séparation des encodeurs et décodeurs pour les profils et les scalaires.

3. **Double espace latent :** les structures précédentes menaient souvent à donner trop d'importance dans l'espace latent aux scalaires. Pour pallier cela, et focaliser le modèle sur les profils, l'on rajoute un espace latent intermédiaire après l'encodeur des profils. Ce sera sa sortie, concaténée à la sortie de l'encodeur des scalaires qui sera passé dans l'espace latent principal. Le but est de former une première représentation de l'espace sur les profils, puis d'y rajouter l'impact des scalaires. Cette utilisation de plusieurs espaces latents dans un seul modèle est déjà apparu dans divers projets de recherches sous une forme hiérarchique, proche de celle utilisée ici [1] voir plus complexe [2].

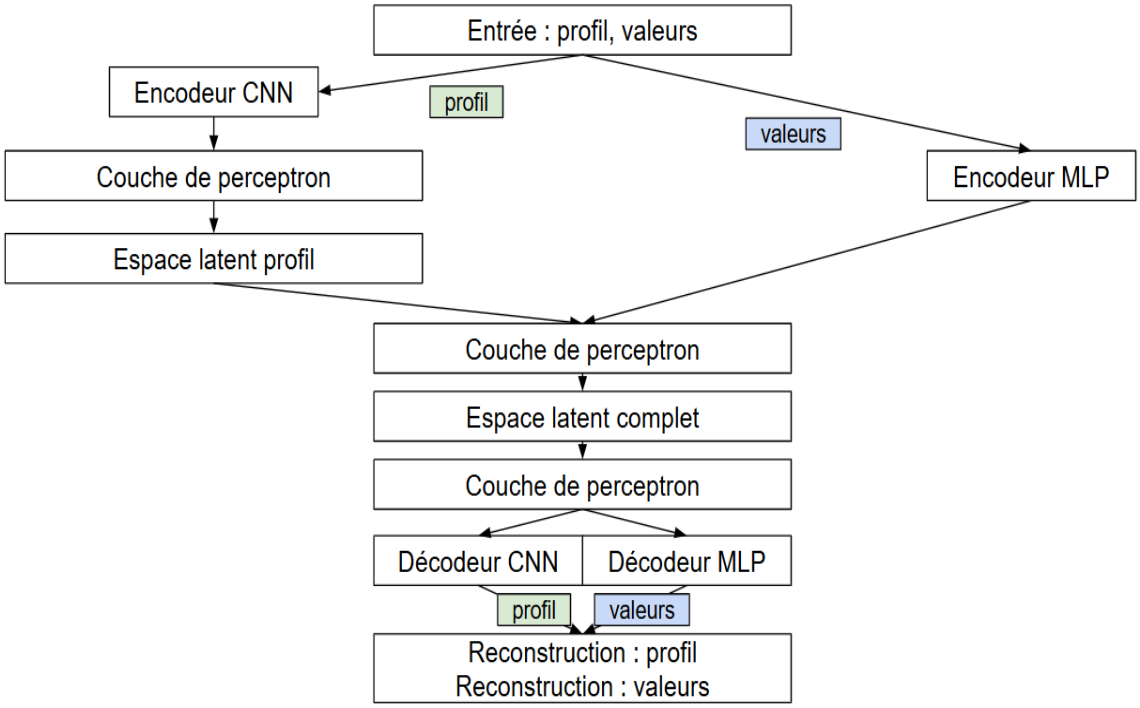

Figure 3 : structure du VAE avec ajout d'un espace latent intermédiaire dédié à la distribution des profils.

Entraînement

L'entraînement se fait par descente de gradient stochastique par batch : les données sont séparés en batch aléatoirement formés à chaque epoch, puis la descente de gradient se fait pour chaque batch à la suite.

Maintenant, chaque module du VAE est entraîné en même temps, mais sur des fonctions de coût séparées, adaptées aux données que les modules observent et à leur rôle respectif. Sont calculés : le coût de reconstruction des profils, le coût de reconstruction des scalaires, le coût de Kullback-Leibler associé à l'espace latent des profils, le coût de Kullback-Leibler associé à l'espace latent global, et le coût de lissage (permettant d'obtenir des profils géométriques sans pics, souvent employé avec une pondération nulle ou faible). L'espace latent global est entraîné sur la somme pondérée de tous les coûts, l'encodeur des profils et l'espace latent des profils sur la somme pondérée du coût de reconstruction des profils et des coûts de Kullback-Leibler, le décodeur des profils sur la somme pondérée du coût de reconstruction des profils et du coût de Kullback-Leibler de l'espace latent global, et les encodeur et décodeur des scalaires sur la somme pondérée du coût de reconstruction des scalaires et du coût de Kullback-Leibler de l'espace latent global.

La pondération des coûts de Kullback-Leibler peut varier au cours du temps en appliquant du recuit simulé. Cela consiste à commencer par une pondération faible de ces coûts permettant de favoriser la reconstruction dans un premier temps, puis d'augmenter leur part pour faire suivre la distribution voulue à la représentation des données sans que la reconstruction prenne le plus d'ampleur à chaque epoch. Cela peut se faire de manière simple (recuit sur 100 epochs sur 1000 par exemple), ou de manière plus complexe, cyclique notamment (recuit sur 100 epochs sur 1000 répété toutes les 200 epochs par exemple). Cette méthode est introduite en 2019 [3].

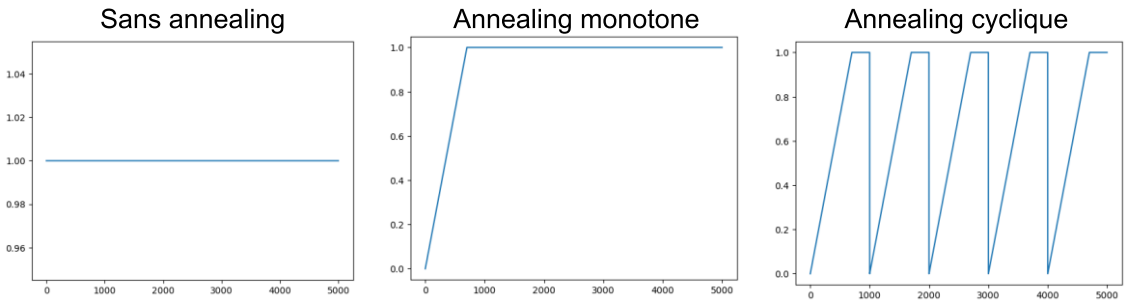

Figure 4 : représentation graphique de différents types de recuit simulé, avec la valeur du coefficient de pondération en ordonnée, et l'epoch en abscisse.

L'optimiseur utilisé est l'optimiseur AdamW, incorporant le moment dans la descente de gradient (les gradients calculés au cours des epochs précédentes influencent les nouveaux gradients) et la décroissance du taux d'apprentissage au cours du temps, de manière mathématiquement correcte, comparée à Adam [4]. La décroissance du taux d'apprentissage est lente et se fait de manière exponentielle. Dans notre cas, nous entraînons souvent sur 5000 epochs notre modèle, et la valeur finale du taux d'apprentissage (λ initialement) est à peu près de 0,9λ.

Bibliographie

[1] Jiaman Li, Ruben Villegas, Duygu Ceylan, Jimei Yang, Zhengfei Kuang, Hao Li, Yajie Zhao. *Task-Generic Hierarchical Human Motion Prior using VAEs*. arXiv:2106.04004v1 [cs.CV] 7 Jun 2021.

[2] Rujikorn Charakorn, Yuttapong Thawornwattana, Sirawaj Itthipuripat, Nick Pawlowski, Poramate Manoonpong, Nat Dilokthanakul. *An Explicit Local and Global Representation Disentanglement Framework with Applications in Deep Clustering and Unsupervised Object Detection*. arXiv:2001.08957v2 [cs.CV] 24 Feb 2020.

[3] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, Lawrence Carin. *Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing*. arXiv:1903.10145v3 [cs.LG] 10 Jun 2019.

[4] Yufeng Hao, Zhengdao Tang, Yixiao Tian, Yijie Zhang, Zheng Zhou. *AdamW*. Cornell University, Automne 2024.

Lexique

CNN : réseau de neurones convolutifs ou convolutionnal neural network

IA : intelligence artificielle

MLP : perceptron multicouche ou multi-layer perceptron

Perceptron : réseau de neurones à composée d'une seule couche de neurones complètement connectés à chaque sortie et chaque entrée.

VAE : autoencodeur variationnel ou variationnal autoencoder