



Tables rondes : Deep Learning et spectroscopie

15/06/2023

Sommaire

- I. Introduction générale
- II. Introduction aux Transformers
- III. Exemple sur une base de données fourrages
- IV. Conclusions
- V. Perspectives



I. Introduction générale

Deep learning et chimiométrie

Aujourd'hui, le Deep learning est de plus en plus utilisé, notamment en industrie pour plusieurs raisons :

- Capable de capturer des informations complexes
- Réduit les difficultés de passage à l'échelle
- Progrès matériels et logiciels (keras, tensorflow, pytorch)
- Adaptabilité à différentes tâches
- Augmentation de la taille des bases de données

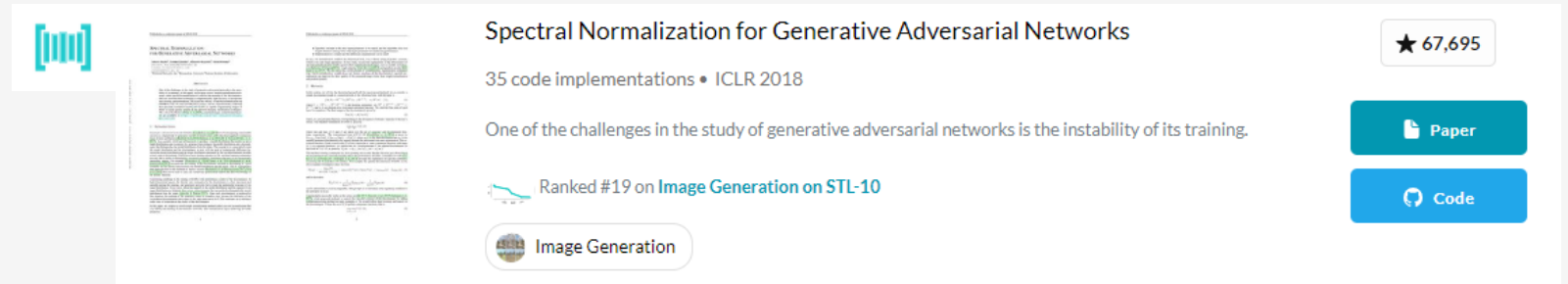
En chimiométrie, aujourd'hui on retrouve principalement une méthode de DL : le CNN (convolutional neural network)

I. Introduction générale

Le Deep Learning pour l'industrie

Architecture des modèles et codes associés disponibles librement :

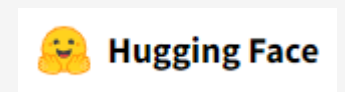
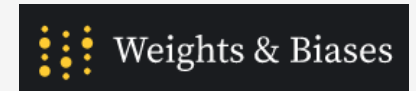
Ex : papers with code
openreview.net



The screenshot shows a page for the paper "Spectral Normalization for Generative Adversarial Networks" on the Papers with Code platform. It features a star rating of 67,695, 35 code implementations, and a mention of ICLR 2018. A description states: "One of the challenges in the study of generative adversarial networks is the instability of its training." It also shows a ranking of #19 on "Image Generation on STL-10" and buttons for "Paper" and "Code".

Outils à disposition :

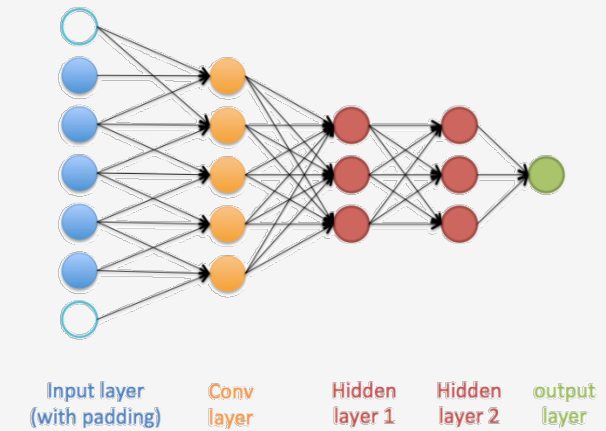
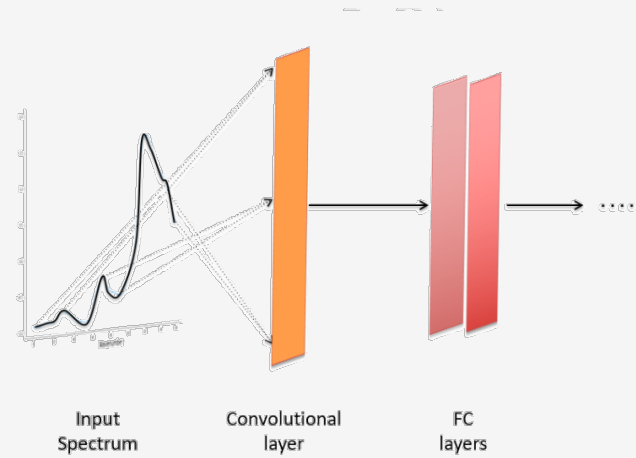
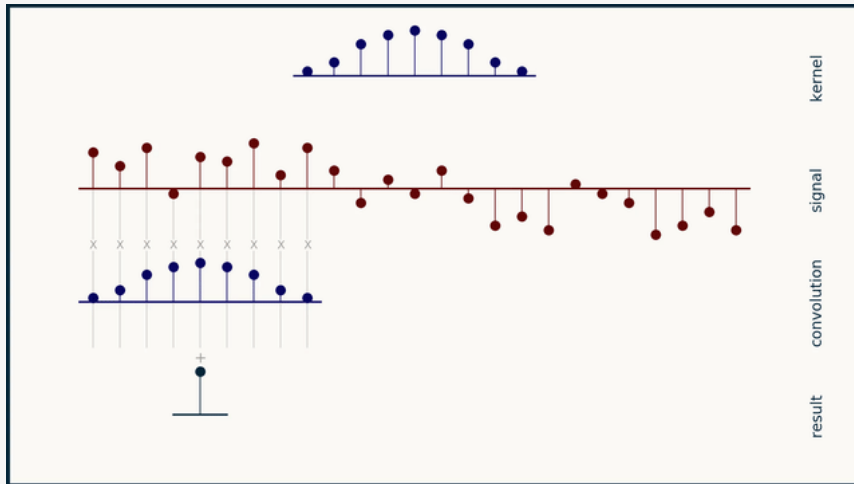
- DataOps : gestion des données (prétraitements, annotation, versionning...)
- MLOps : gestion des modèles (hyperparamètres, métriques, versionning, CI/CD, ...)



Certains domaines plus appropriés car communauté très importante :

Computer Vision ou Natural Language Processing par exemple

I. Introduction générale



Descripteurs

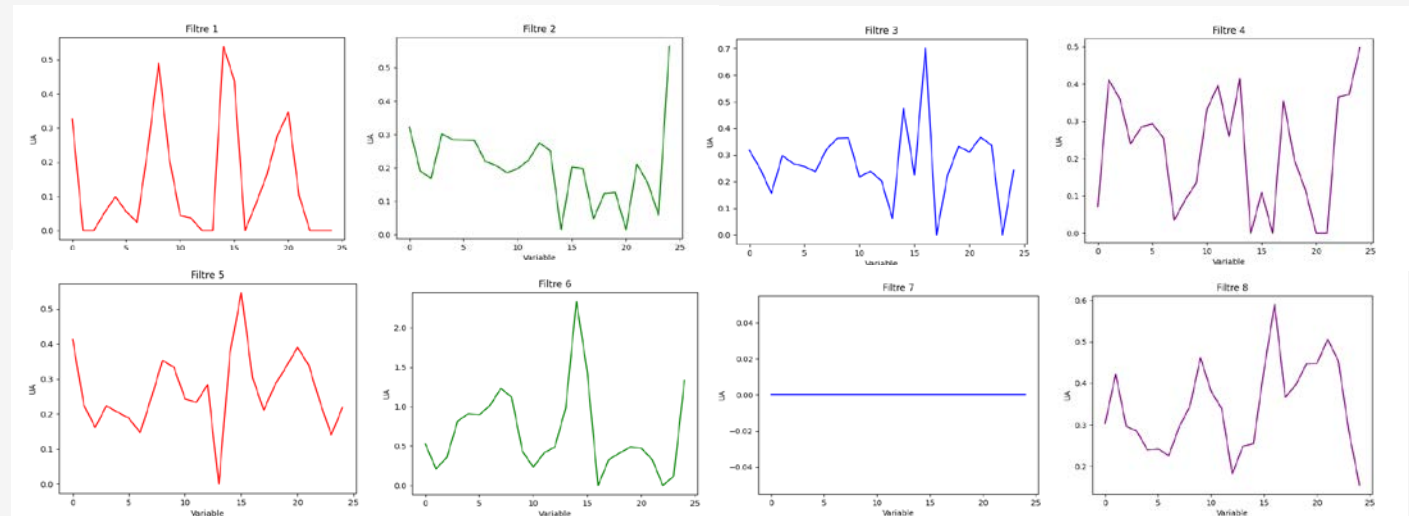
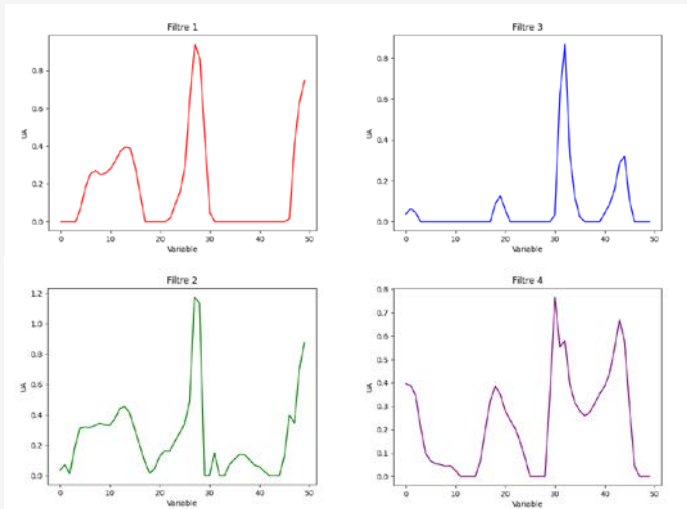
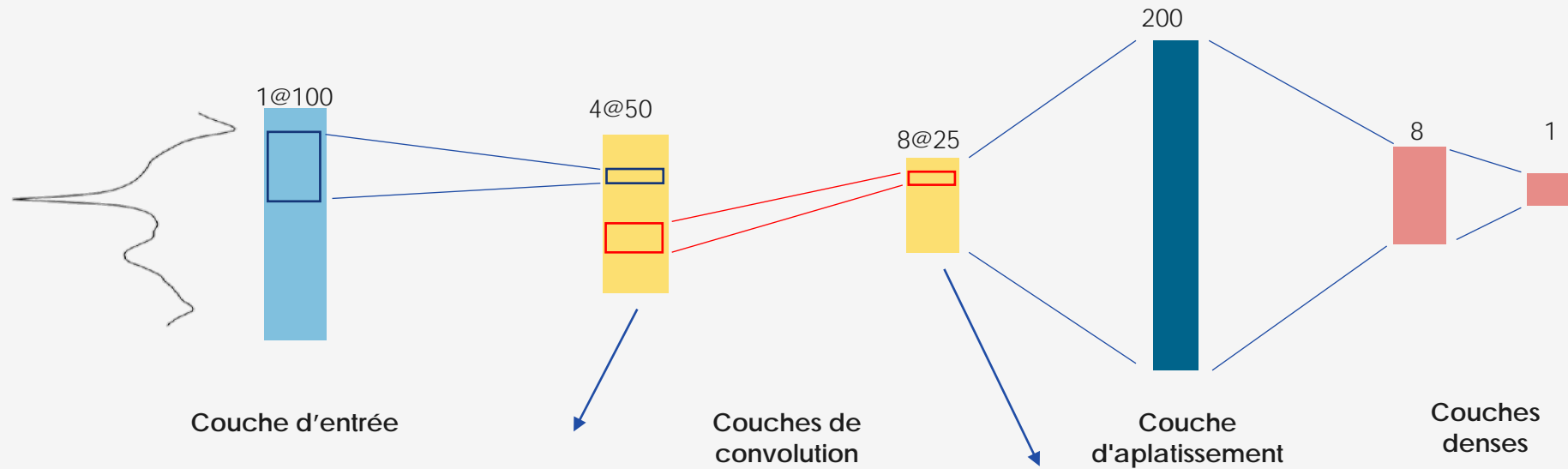
Régresseur/
Classifieur

I. Introduction générale

Les avantages du CNN :

- Exploitation de motifs locaux
- Apprentissage de caractéristiques hiérarchiques
- Invariances au translations

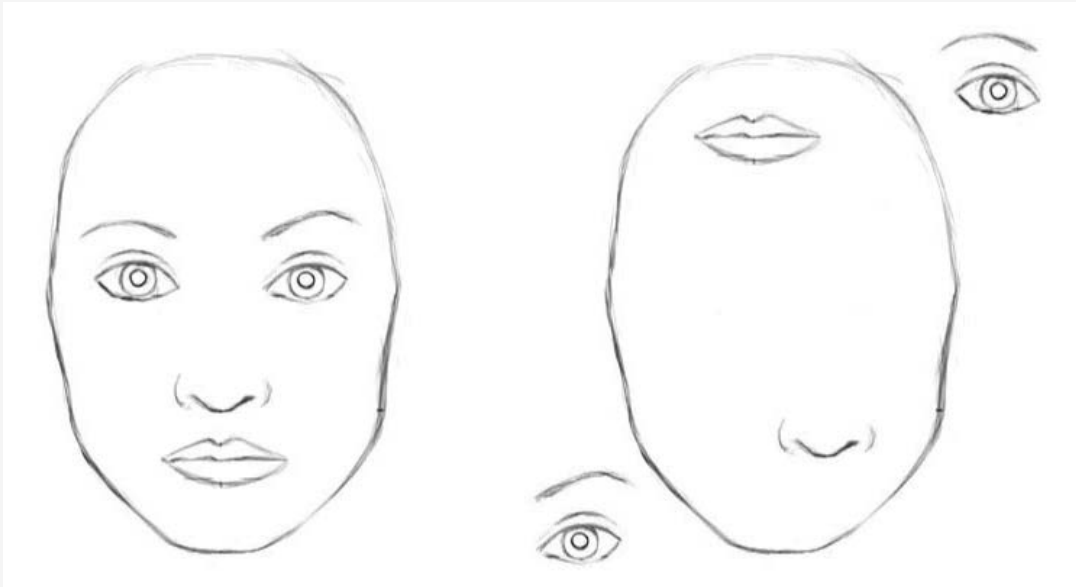
I. Introduction générale



I. Introduction générale

Quels est la principale limite des CNNs pour le traitement des données spectrales ?

- **Manque de contexte** : des Zones spectrales très éloignées peuvent être corrélées

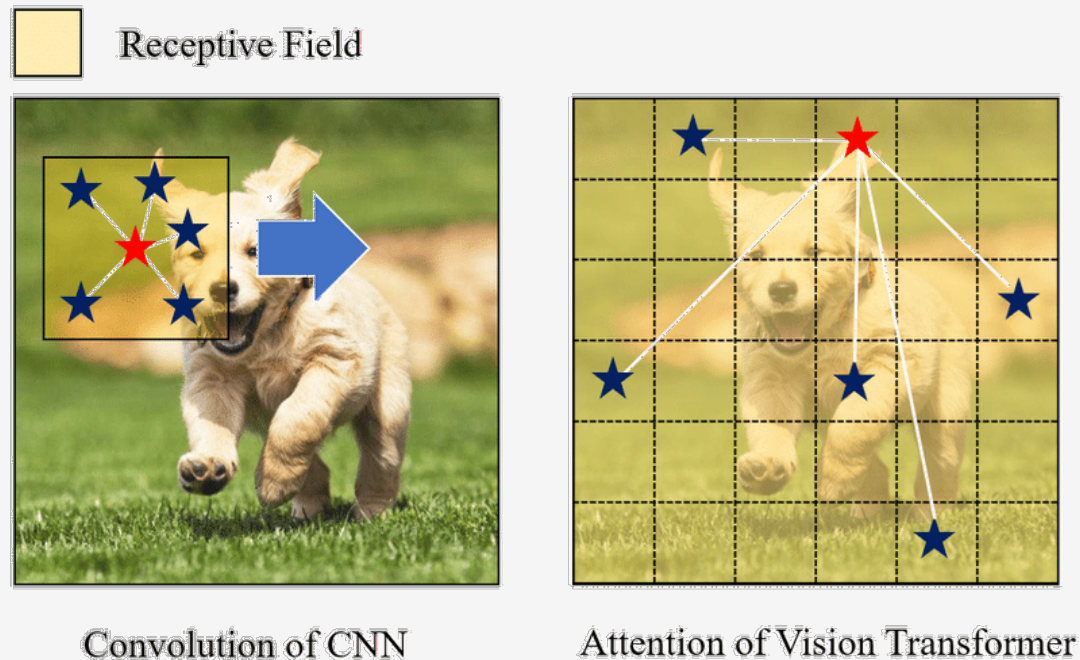


Pour un CNN ces deux images peuvent être les mêmes

II. Introduction aux Transformers

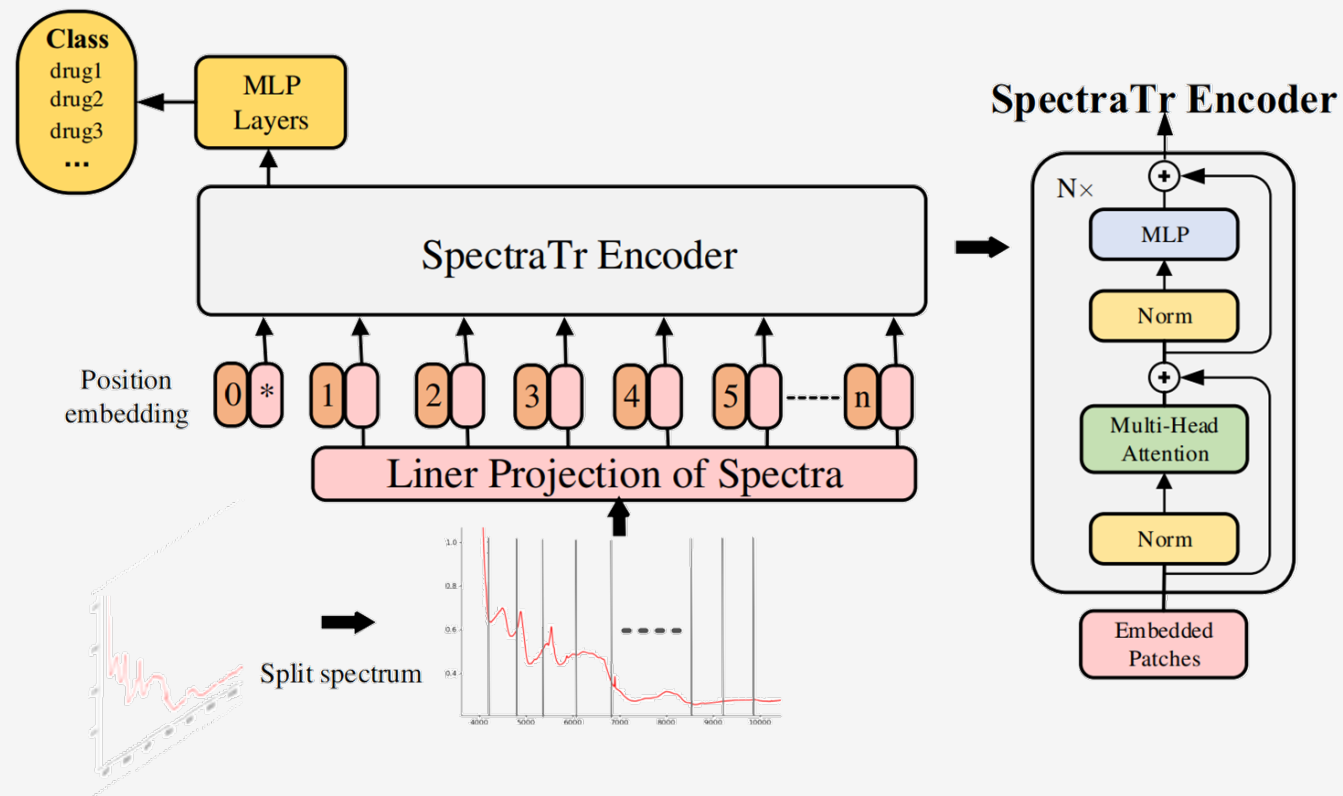
Les transformers sont des architectures qui ont été initialement proposées pour les tâches de traitement du langage naturel mais qui peuvent être appliquées aux données spectrales.

Les transformers peuvent capturer les dépendances à longue distance pour chaque élément de la séquence d'entrée.



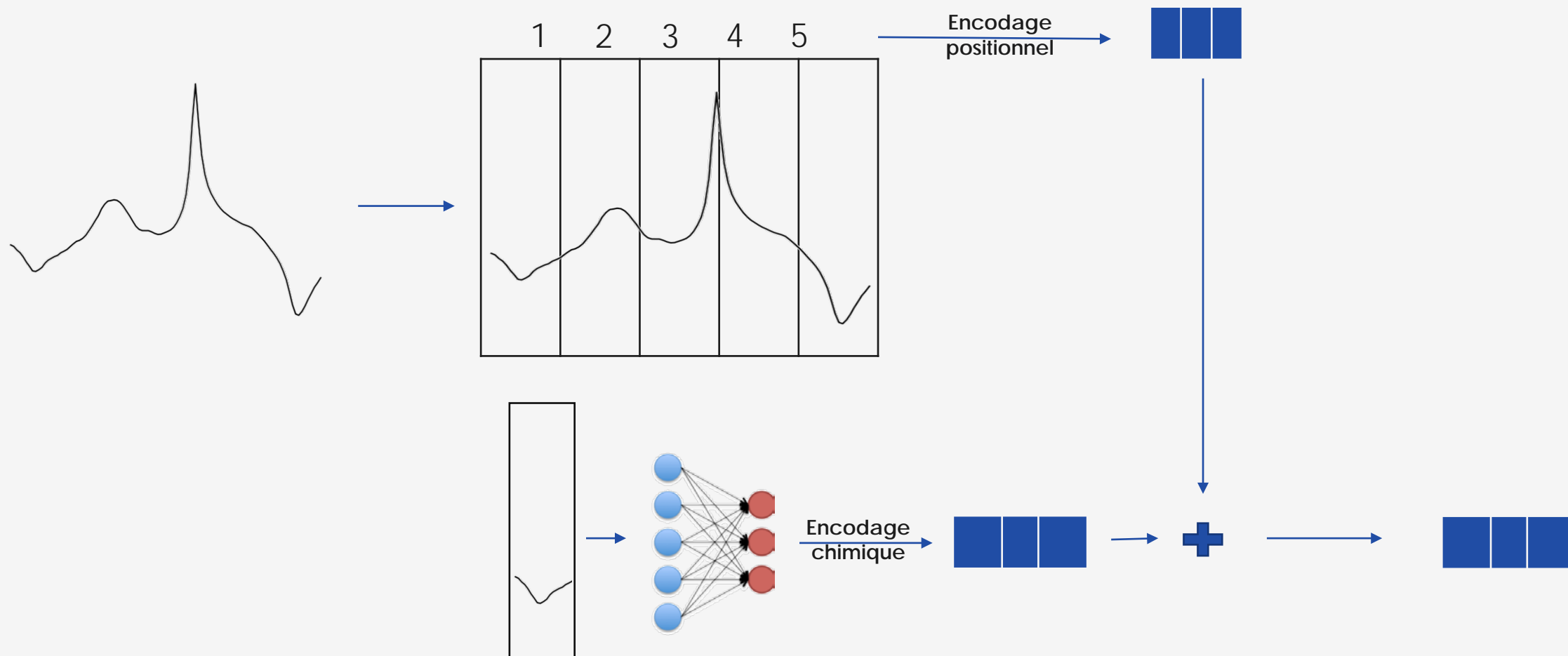
II. Introduction aux Transformers

Exemple de SpectraTr (copié sur ViT) :



II. Introduction aux Transformers

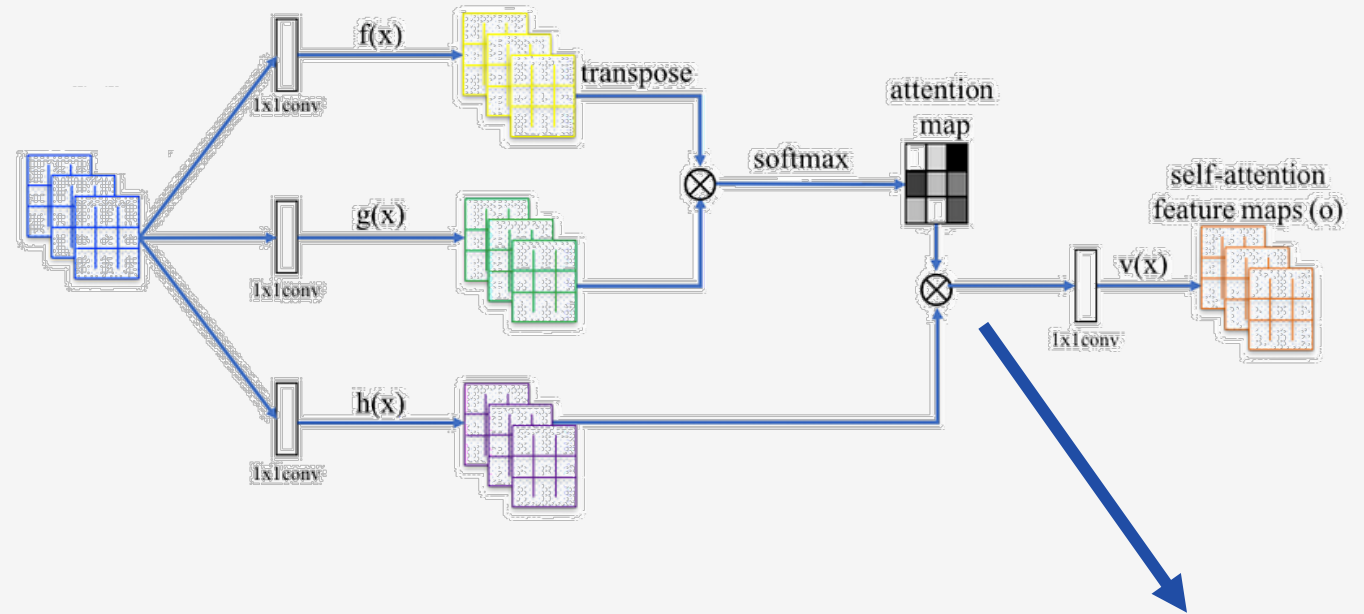
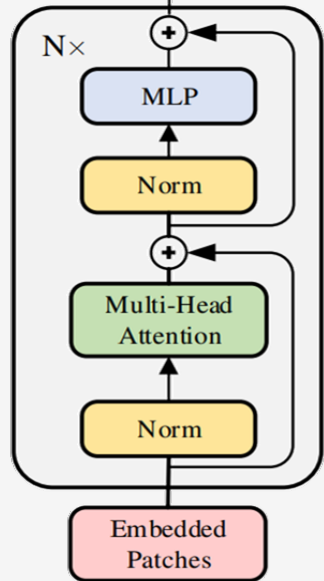
Embeddings (Encodage) : Compression de l'information chimique et positionnelle



II. Introduction aux Transformers

Encodeur :

SpectraTr Encoder



Combinaisons
linéaires des
morceaux de
départs

Performance sur une base de données fourrages

Origine des données :

Données fourrages (pré-traitées et totalement anonymisées) de 700 longueurs d'onde différentes
Cirad-UMR Selmet - Plateforme @DoPredict
Préparation des données : L. Bonnal, M. Lesnoff

Utilisation :

Les ensembles de données NIR pour les 6 mesures de références.
Il y a approximativement 12 000 observations pour chaque mesure.
2000 observations sont utilisées comme ensemble de test.

III. Exemple sur une base de données fourrages

Modèles testés :

- CNN
- SpectraTr
- PLS-locale

- CNN

Layer (type)	Output Shape	Param #
reshape_1 (Reshape)	(None, 700, 1)	0
conv1d_8 (Conv1D)	(None, 350, 4)	132
conv1d_9 (Conv1D)	(None, 175, 8)	200
conv1d_10 (Conv1D)	(None, 88, 16)	784
flatten_7 (Flatten)	(None, 1408)	0
dense_38 (Dense)	(None, 64)	90176
dense_39 (Dense)	(None, 32)	2080
dense_40 (Dense)	(None, 1)	33

=====
Total params: 93,405
Trainable params: 93,405
Non-trainable params: 0
=====

III. Exemple sur une base de données fourrages

Modèles testés :

- CNN
- SpectraTr
- PLS-locale

- SpectraTr

```
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 700, 1)]	0	[]
gaussian_noise (GaussianNoise)	(None, 700, 1)	0	['input_1[0][0]']
conv1d_86 (Conv1D)	(None, 140, 140)	840	['gaussian_noise[0][0]']
patch_encoder (PatchEncoder)	(None, 140, 64)	17984	['conv1d_86[0][0]']
multi_head_attention (MultiHeadAttention)	(None, 140, 64)	33216	['patch_encoder[0][0]', 'patch_encoder[0][0]']
add (Add)	(None, 140, 64)	0	['multi_head_attention[0][0]', 'patch_encoder[0][0]']
layer_normalization (LayerNormalization)	(None, 140, 64)	128	['add[0][0]']
dense_90 (Dense)	(None, 140, 64)	4160	['layer_normalization[0][0]']
dense_91 (Dense)	(None, 140, 32)	2080	['dense_90[0][0]']
flatten_37 (Flatten)	(None, 4480)	0	['dense_91[0][0]']
dense_92 (Dense)	(None, 512)	2294272	['flatten_37[0][0]']
dropout (Dropout)	(None, 512)	0	['dense_92[0][0]']
dense_93 (Dense)	(None, 64)	32832	['dropout[0][0]']
dense_94 (Dense)	(None, 8)	520	['dense_93[0][0]']
dense_95 (Dense)	(None, 1)	9	['dense_94[0][0]']

```

=====
Total params: 2,386,041
Trainable params: 2,386,041
Non-trainable params: 0
=====

```

III. Exemple sur une base de données fourrages

Résultats :

Code Y/R2	PLS-locale	SpectraTr	CNN
Y1	0,91	0,90	0,89
Y2	0,94	0,95	0,93
Y5	0,95	0,95	0,94
Y6	0,88	0,87	0,84
Y7	0,94	0,93	0,92
Y8	0,94	0,94	0,92

- Les résultats sont équivalents entre PLS-locale et SpectraTr et CNN sauf pour y6
- Le CNN performe globalement moins bien que la PLS-locale et SpectraTr
- Pour les approches CNN/transformers seul le prétraitement apporte une connaissance métier (ni les architectures proposées, ni la stratégie d'entraînement)

VI. Conclusions

Pour des relations faiblement non-linéaires, ces modèles ne sont pas obligatoirement nécessaires (mais ils permettent des temps d'inférence intéressants)

Sans connaissances métier, le modèle basé sur le transformer performe aussi bien qu'un modèle PLS-local.

Le modèle basé sur les transformers est souvent très polyvalent et permet de traiter des données très diverses

Les apports du DL se trouvent sur des tâches plus complexes (identification de pics)

Traitement d'images Hyper/multi-spectrales

V. Perspectives pour l'industrie

Adapter les mécanismes d'attention pour les images multispectrales / hyperspectrales :

- Les transformers actuels exploitent beaucoup plus le contexte spatial car ils ont été élaborés sur des images couleurs (3 canaux colorimétriques) => comment exploiter le contexte dans la dimension spectrale?

Tirer profit de l'accès facilité aux données pour entraîner des modèles en continu :

- Modèle auto-supervisé alimenté par des données acquises en continu par des process de production sans besoin d'annotation (centre de tri de déchets par exemple)

Evaluer l'intérêt des mécanismes d'attention pour les données multimodales :

- Architectures intégrant des données de spectroscopie proche infrarouge, visible, colorimétrie pour la résolution spatiale fine et 3D par exemple

V. Perspectives pour la recherche en chimiométrie

Intégrer de la connaissance métier dans les approches de DL :

- Augmentations de données ~ prétraitements (covariant vs invariant)
- Utilisation de PINNS (Physics-Informed Neural Networks)

Utiliser des approches modernes de Deep learning :

- Développement de nouvelles stratégies d'entraînement (SSL, AL, WSL)
- Utiliser des approches génératives pour prédire des mélanges spectraux ? Une meilleure simulation de mélange

Mettre en place des modèles génériques/ et base de données communes pour offrir à toute la communauté des modèles polyvalents et performants