Soutenance du Stage de deuxième année F4
Détermination d'un algorithme améliorant l'apprentissage d'un réseau de neurones

Julien Feuillas

ISIMA

23 Mai 2019

Objectifs du groupe MMIV

- MMIV (Mohn Medical Imaging and Visualization Centre)
  - Centre de recherche en termes d'imagerie médicale
  - Basé en Norvège
  - Dépendant de l'Université de Bergen
  - Représentante : Mme Renate Grüner
- Objectifs
  - Mettre en place de nouvelles techniques d'apprentisssage automatique

Travaux réalisés dans le cadre du projet

- Recherche effectuée par le laboratoire
- Récupération de données
  - Jeu de données d'IRM de cerveaux
- Mise en place d'une première solution
  - Acquisition de données
  - Algorithme modifiant le paramètre "Learning Rate" au cours de l'entraînement

Cadre du Stage

- Nos Objectifs
  - Déterminer l'impact du paramètre Learning Rate sur l'apprentissage d'un réseau de neurones
  - Améliorer si possible cet apprentissage
  - Étude de différentes solutions

- Cadre d'étude
  - Optimisation de fonction
  - Deep Learning
    - Segmentation
    - Learning rate

Problématique

Est-il possible d'améliorer l'apprentissage d'une méthode de Deep Learning en modifiant le "taux d'apprentissage" au cours eu cours de l'entraînement ?

Plan

Plan

Installation

- Choix du matériel
  - CPU
  - GPU
- Installation Anaconda
- Installation Tensorflow
- Installation NiftyNet

Fichier de Configuration

```
[image]
path_to_search=data/images
filename_contains=IXI, orig
interp_order=3
axcodes=L,P,S
spatial_window_size=80, 80, 80

[label]
path_to_search=data/labels
filename_contains=IXI, brain
interp_order=0
axcodes=L,P,S
spatial_window_size=80, 80, 80

[SYSTEM]
cuda_devices=0
num_threads=10
num_gpus=1

[NETWORK]
name=highres3dnet
activation_function=prelu
batch_size=1
reg_type=L2
decay=1e-5
queue_length=20
```

```
[TRAINING]
optimiser=adam
sample_per_volume=80
lr=1e-3
loss_type=Dice
starting_iter=0
save_every_n=2500
max_iter=20000
max_checkpoints=1000

[INFERENCE]

[EVALUATION]

[SEGMENTATION]
image=image
label=label
output_prob=False
num_classes=2
label_normalisation=True
```

Entraînement du modèle

```
INFO:niftynet: training iter 2725, loss=0.5583992600440979 (0.868634s)
INFO:niftynet: training iter 2726, loss=0.6007713079452515 (0.876589s)
INFO:niftynet: training iter 2727, loss=0.6028639078140259 (0.863837s)
INFO:niftynet: training iter 2728, loss=0.7271522879600525 (0.868486s)
INFO:niftynet: training iter 2729, loss=0.5072353482246399 (0.882997s)
INFO:niftynet: training iter 2730, loss=0.5244722962379456 (0.850369s)
INFO:niftynet:     validation iter 2730, loss=0.5917767882347107 (0.354249s)
INFO:niftynet: training iter 2731, loss=0.5584901571273804 (0.855727s)
INFO:niftynet: training iter 2732, loss=0.5593559145927429 (0.836951s)
INFO:niftynet: training iter 2733, loss=0.2506962716579437 (0.829555s)
INFO:niftynet: training iter 2734, loss=0.5547387599945068 (0.832222s)
INFO:niftynet: training iter 2735, loss=0.7395037412643433 (0.877460s)
INFO:niftynet: training iter 2736, loss=0.5985323190689087 (0.877698s)
INFO:niftynet: training iter 2737, loss=0.5332014560699463 (0.887804s)
INFO:niftynet: training iter 2738, loss=0.5504813194274902 (0.879380s)
INFO:niftynet: training iter 2739, loss=0.646152138710022 (0.865054s)
INFO:niftynet: training iter 2740, loss=0.6777690052986145 (0.918569s)
INFO:niftynet:     validation iter 2740, loss=0.5845479965209961 (0.371661s)
INFO:niftynet: training iter 2741, loss=0.5483676791191101 (0.866762s)
INFO:niftynet: training iter 2742, loss=0.597277045249939 (0.867351s)
INFO:niftynet: training iter 2743, loss=0.5173805952072144 (0.834379s)
INFO:niftynet: training iter 2744, loss=0.6961113214492798 (0.873546s)
INFO:niftynet: training iter 2745, loss=0.5568277835845947 (0.837432s)
INFO:niftynet: training iter 2746, loss=0.5968761444091797 (0.859206s)
INFO:niftynet: training iter 2747, loss=0.5399388074874878 (0.873980s)
INFO:niftynet: training iter 2748, loss=0.5923281908035278 (0.844980s)
INFO:niftynet: training iter 2749, loss=0.5360341668128967 (0.869045s)
INFO:niftynet: training iter 2750, loss=0.0 (0.819159s)
INFO:niftynet:     validation iter 2750, loss=0.5554222464561462 (0.361879s)
INFO:niftynet: training iter 2751, loss=0.5011717081069946 (0.878481s)
INFO:niftynet: training iter 2752, loss=0.6217421293258667 (0.879448s)
INFO:niftynet: training iter 2753, loss=0.5196523666381836 (0.882210s)
INFO:niftynet: training iter 2754, loss=0.2939032316207886 (0.842363s)
INFO:niftynet: training iter 2755, loss=0.6358374953269958 (0.876634s)
INFO:niftynet: training iter 2756, loss=0.578665018081665 (0.872801s)
INFO:niftynet: training iter 2757, loss=0.5292662978172302 (0.870333s)
INFO:niftynet: training iter 2758, loss=0.537699818611145 (0.893807s)
INFO:niftynet: training iter 2759, loss=0.577454149723053 (0.893881s)
INFO:niftynet: training iter 2760, loss=0.5138360857963562 (0.873233s)
INFO:niftynet:     validation iter 2760, loss=0.5427689552307129 (0.366395s)
INFO:niftynet: training iter 2761, loss=0.7498124837875366 (0.907313s)
INFO:niftynet: training iter 2762, loss=0.5954206585884094 (0.884163s)
INFO:niftynet: training iter 2763, loss=0.5001587867736816 (0.847311s)
```

Figure – Entraînement du système

Loss Function

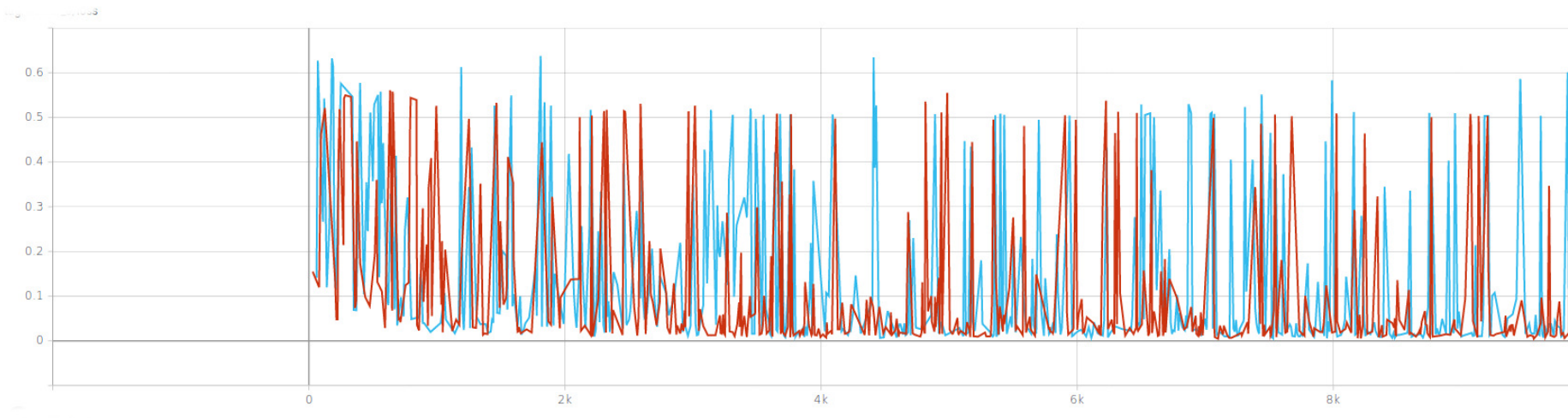

Figure – Courbe réelle

Loss Function

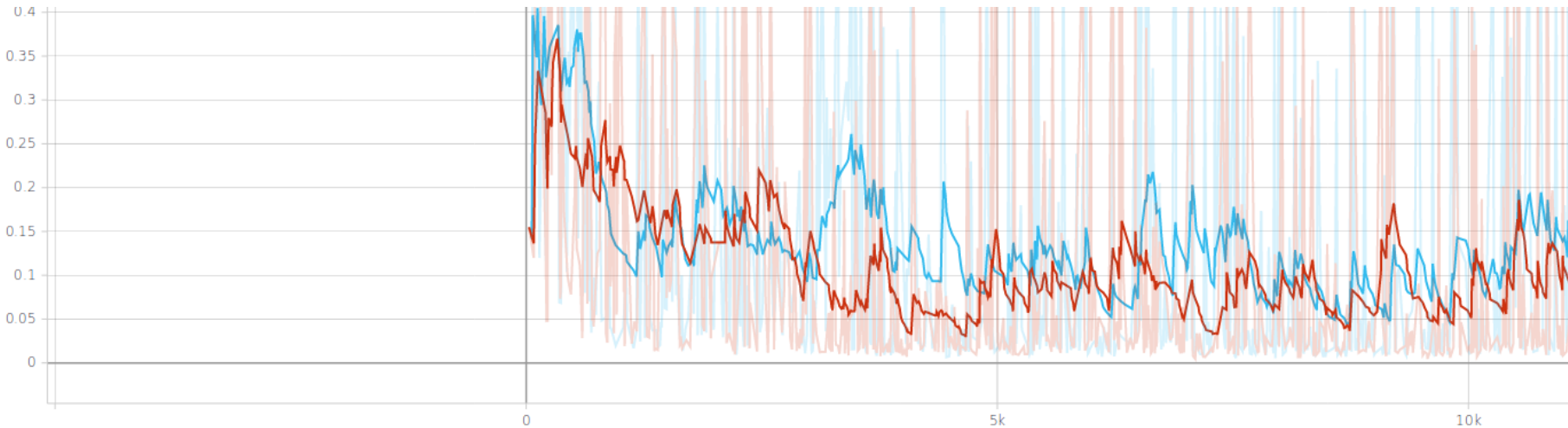

Figure – Courbe amortie avec un coefficient 0.9

Loss Function

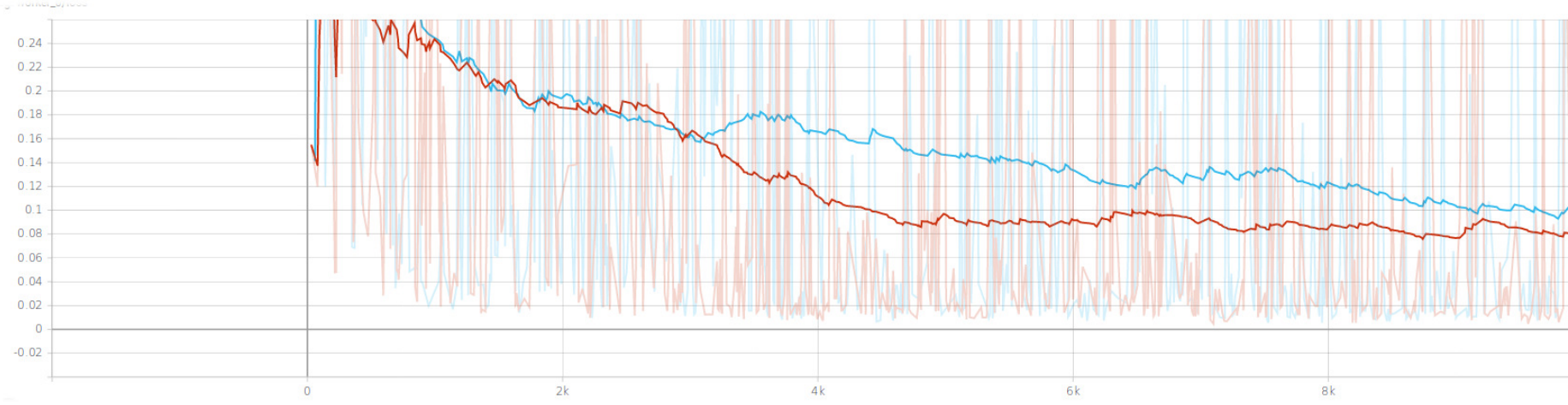

Figure – Courbe amortie avec un coefficient 0.99

Loss Function

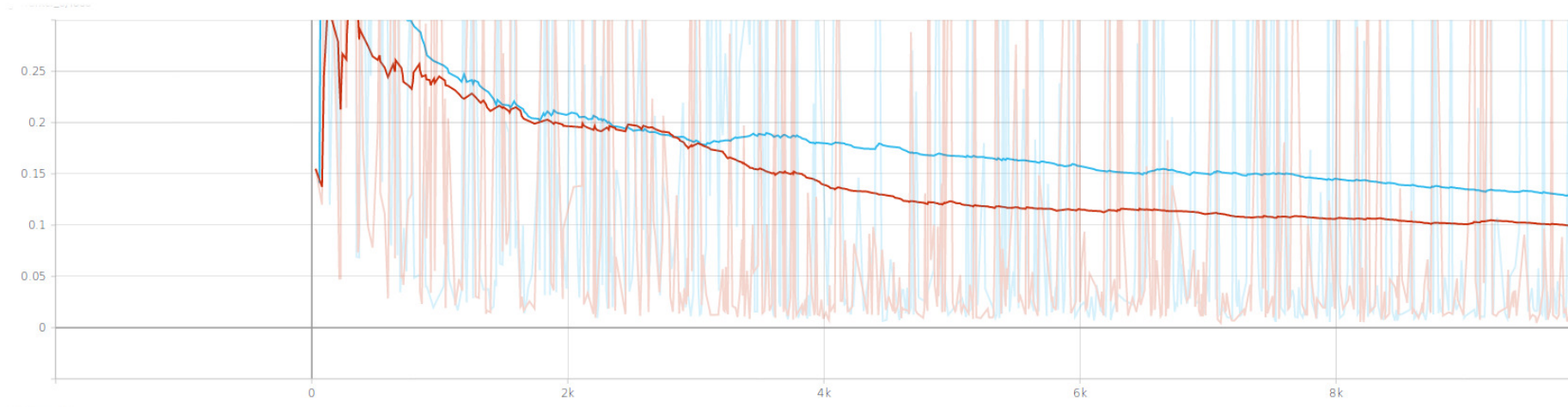

Figure – Courbe amortie avec un coefficient 0.999

Résultats

Plan

Première modification de lr

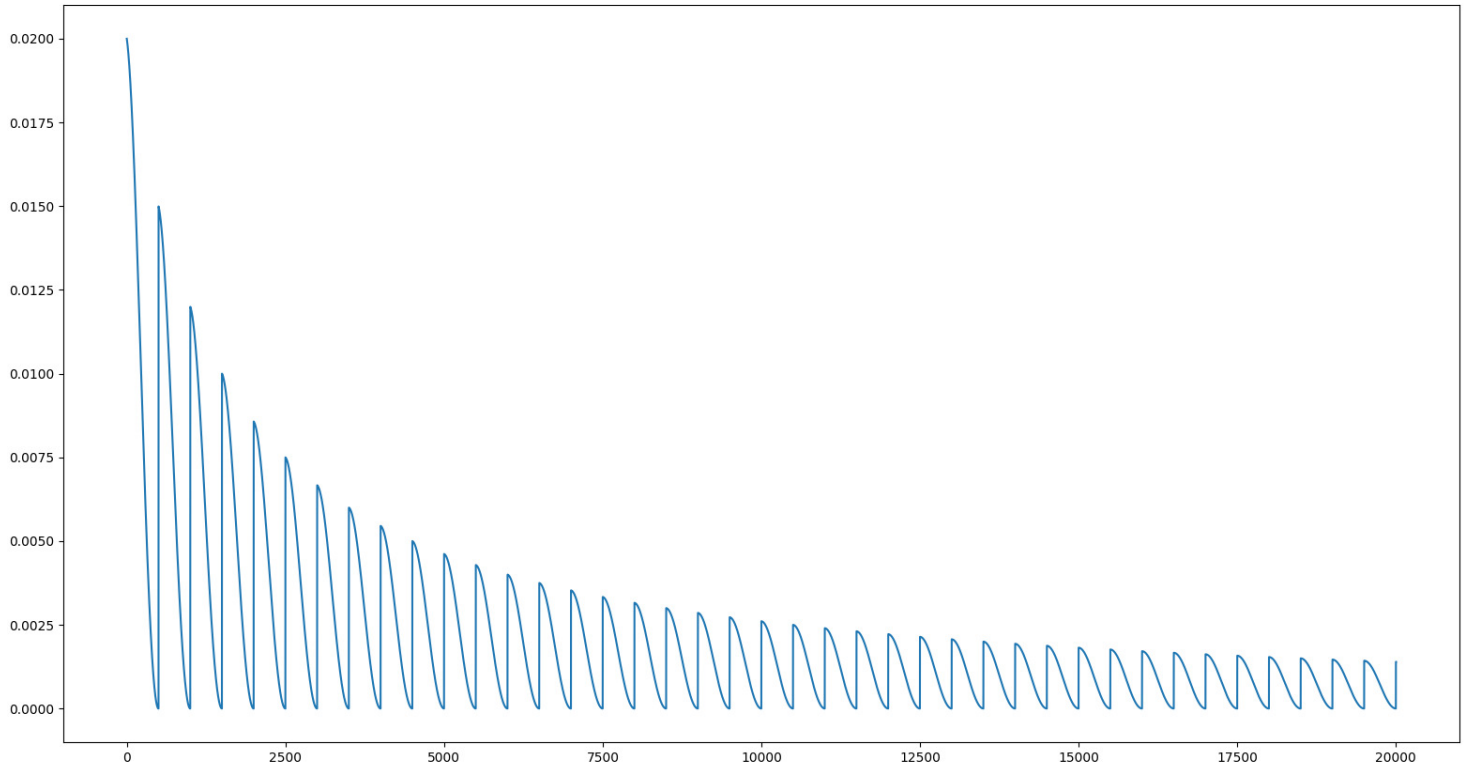

Figure – Modification du “taux d’apprentissage” à chaque itration

Plan

Acquisition des données

- Découpage de la pharmacie en différentes zones
- Méthode d'acquisition des données :
  - Téléphone posé dans une zone
  - Émission de signaux pendant une certaine durée
- Stockage des données :
  `x;y;date;c1;c2;c3;c4`

Découpage de la pharmacie

Machine Learning

- Vocabulaire :
  - Feature ($c1$,$c2$,$c3$,$c4$)
  - Target (x,y)
- Algorithmes paramétrés
- Deux étapes :
  - Entraînement
  - Test

Classification et Régression

- Classification
  - Regroupement en classes
  - Classes – Targets

- Régression
  - Forme de la fonction : features $\mapsto$ targets
  - Minimisation de l'erreur
  - Méthodes linéaires :

$$\begin{cases} t_1 = \alpha_{1,0} + \alpha_{1,1} f_1 + \dots + \alpha_{1,n} f_n \\ \vdots \\ t_m = \alpha_{m,0} + \alpha_{m,1} f_1 + \dots + \alpha_{m,n} f_n \end{cases}$$

Outils utilisés

- Python
- Bibliothèques Python :
  - Numpy
  - Scikit-Learn et Pandas
  - Matplotlib et Seaborn

Déroulement du projet

- Rendez-vous hebdomadaires
- Objectifs définis au cours du projet

Plan

Découpage du jeu de données

Méthodes de Classification

- K Nearest Neighbors
  - Recherche des voisins
  - Détermination de la classe
- Support Vector Machine (SVM)
  - Choix d'une méthode
  - Itération

Métriques pour la Classification

- Précision $= \frac{\text{Nombre de prédictions correctes}}{\text{Nombre de données}}$
- Matrice de confusion :

	A	**B**
A	2345	0
B	213	2143

Régressions linéaire et polynomiale

- Régression linéaire
  - Méthode des moindres carrés
  - Erreur : $\|.\|_2^2$

- Régression polynomiale
  - Paramètre : degré
  - Exemple :
    $t = a_0 + a_1 f_1 + a_2 f_2 + a_3 f_1^2 + a_4 f_2^2 + a_5 f_1 f_2$
  - Méthode linéaire ?

Régressions linéaires paramétrées

- Régression Ridge
  - Redondance d'information entre les individus
  - Coefficient de pénalité α
- Régression LASSO
  - Redondance d'information dans les features
  - Coefficient de pénalité α
- Régression Elastic-Net
  - Combinaison des méthodes précédentes
  - Deux Coefficients :
    - α : Coefficient de pénalité
    - $\rho \in [0, 1]$: Contrôle de la combinaision

Métriques pour la régression

- Erreur de "distance" :
  - Erreur quadratique :
    - $MSE(Y, Y_{pred}) = \frac{1}{N_{test}} \sum_{l=1}^{N_{test}} (Y_l - Y_{pred,l})^2$
  - Erreur moyenne absolue
    - $MAE(Y, Y_{pred}) = \frac{1}{N_{test}} \sum_{l=1}^{N_{test}} |Y_l - Y_{pred,l}|$

- Capacité de prédiction du modèle
  - Coefficient de détermination
    - $R^2(Y, Y_{pred}) = 1 - \frac{\sum_{l=1}^{N_{test}} (Y_l - Y_{pred,l})^2}{\sum_{l=1}^{N_{test}} (Y_l - \bar{Y})^2}$
  - Score de variance expliquée
    - $EVS(Y, Y_{pred}) = 1 - \frac{Var(Y - Y_{pred,l})}{Var(Y)}$

Efficacité des méthodes de régression

- Résultats dépendants du découpage initial
- Écriture d'une fonction Python :
  - `apply_regressions`
  - Plusieurs applications du même modèle de régression
  - Renvoie la valeur des métriques pour chaque application
  - Variables d'entrée :
    - Nombre de régressions
    - Le modèle à appliquer
    - Conservation des lignes dupliquées ?
- Utilisable pour la détermination de paramètres

Plan

K Nearest Neighbors

- Précision = 98%

Support Vector Classification

- Précision = 68%

Détermination des paramètres : polynomiale

Détermination des paramètres : Ridge

Détermination des paramètres : LASSO

Détermination des paramètres : Elastic-Net

- Méthode précédente non utilisable
- Utilisation d'un objet déjà implémenté :
  - `MultiTaskElasticNetCV` de `sklearn.linear_model`
  - Validation croisée + méthode de régression
- Paramètre de construction :
  - Nombre de plis à effectuer
  - Un tableau de valeurs possibles pour α er ρ

Tableau de résultats

	Duplications ?	MSE	MAE	R2	EVS
Linéaire	oui	1.862	1.097	0.642	0.643
	non	2.061	1.158	0.602	0.606
Polynomiale	oui	0.900	0.666	0.824	0.825
	non	1.299	0.773	0.717	0.720
Elastic-Net	oui	1.864	1.098	0.642	0.642
	non	2.064	1.160	0.601	0.604

Comparaison entre classification et régression

- Efficacité de la régression :
  - Au sens des métriques de régression
  - Différentes des métriques de classification
- Passage régression $\rightarrow$ classification
  - Pour la régression polynomiale

Passage de la régression à la classification

Résultats de la régression polynomiale pour la classification

- Résultats :
  - Précision : 54%
  - Termes extra-diagonaux de la matrice de confusion
- Résultats peu précis
  - Zones trop grandes
  - Nécessité d'un nouveau jeu de données

Graphique de densité

Résumé du travail réalisé

- Méthodes de régression implémentées
  - Méthodes linéaires
  - Méthodes probabilistes ? Régressions à noyau ?
- Résultats obtenus
  - Non concluants
  - Méthodes employées réutilisables
- Ce qu'il reste à faire
  - Mise en place de nouvelles régressions
  - Étude d'un nouveau jeu de données

Remerciements

Nous vous remercions pour votre attention