

Concours BraTS (MLRF)

Nicolas Boutry¹

✉ nicolas.boutry@lrde.epita.fr

¹ Laboratoire de Recherche et Développement de l'EPITA (LRDE), France

Juin 2019



Outline

- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Outline

- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Plan du cours I

- Présentation et objectif du concours BraTS (Brain Tumor Segmentation),
- Acquisition des différentes modalités (T1,T1ce,T2,FLAIR),
- Données fournies par les organisateurs (modalités + segmentations),
- Méthodologie de segmentation (pipeline)
 - mise en forme / sélection des données (pre-processing),
 - séparation des données en un set d'apprentissage et un set de validation,
 - entraînement du classifieur sur les données d'apprentissage,
 - mesures de qualité du classifieur (vrais/faux positifs/négatifs, recall, precision, F-Score),
 - reparamétrage du réseau pour atteindre une forme d'optimalité (batchsize, époques, optimiseur, ...),
 - une fois l'optimal atteint pour ce modèle, repenser le modèle (données d'entrée locales ou régionales ? changer de classifieur ? ...),
 - recherche de l'optimal pour le nouveau modèle,
 - comparaison des résultats entre les deux modèles précédents et déduction d'un nouveau modèle,
 - etc.
- Différentes approches que l'on va tester,
 - deux modèles étudiés ici (SVM's vs. ANN's),
 - features locales vs. régionales ($K \in \{0, 1, \dots\}$) et caractère trop ou trop peu discriminant → trouver une forme d'optimalité,
 - nombre de cerveaux nécessaires pour l'apprentissage (dû à l'hétérogénéité des tumeurs en intensité et en formes): le plus est-il le mieux? contraintes de mémoire et de temps ?

Outline

- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Anatomie I

Une peu d'anatomie d'un cerveau (coupe axiale):

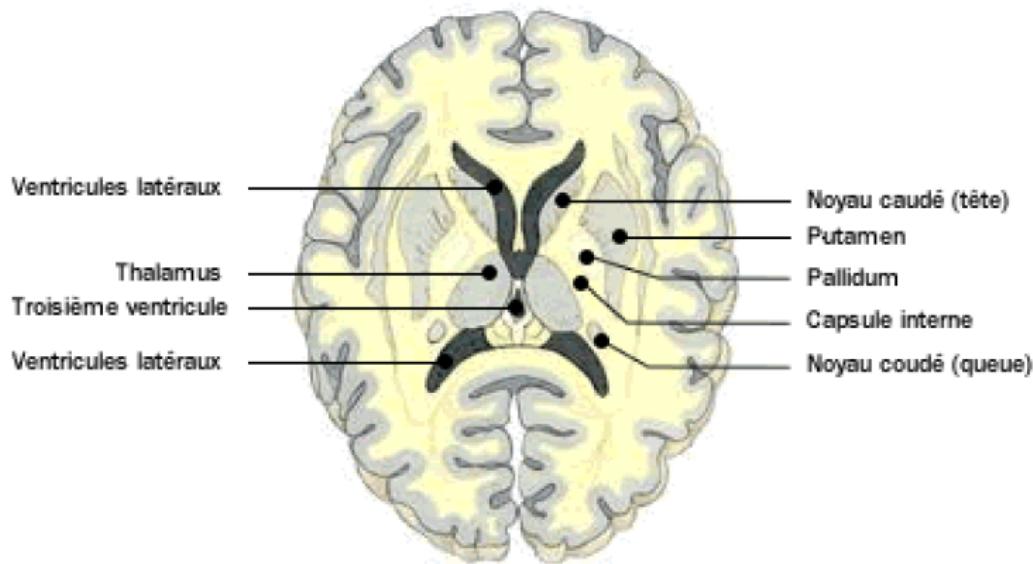
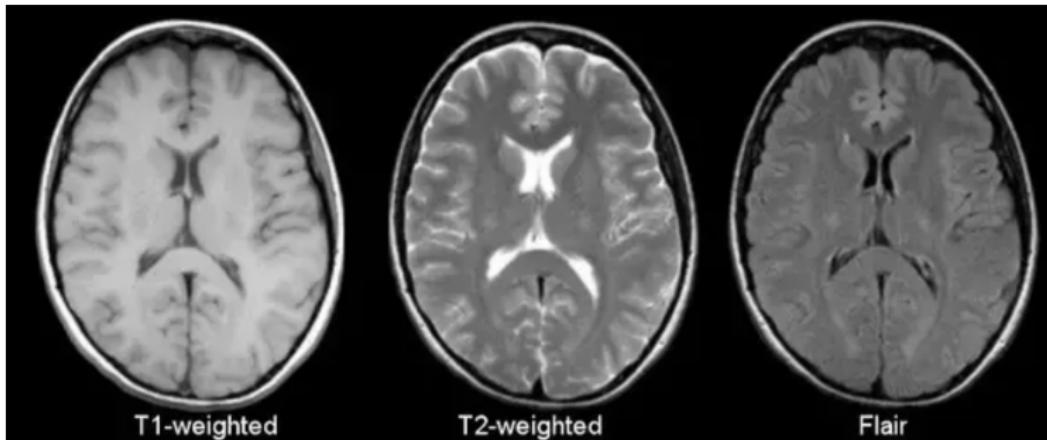


Figure: Anatomie d'une coupe transversale de cerveau humain

Les différentes modalités I

What is the difference between MRI scans T1, T1c, T2, Flair ? Is there another type of scan?

MR images can be acquired with several different techniques (pulse sequences) and acquisition parameters (called e.g. echo time, TE, repetition time TR etc.) resulting in different image contrast. The 4 image types that you mentioned belong to the most common MR acquisitions:



Les différentes modalités II

- T1: T1-weighted MRI: image contrast is based predominantly on the T1 (longitudinal) relaxation time of tissue; tissue with short T1 relaxation time appears brighter (hyperintense),
- T2: T2-weighted MRI: image contrast is based predominantly on the T2 (transverse) relaxation time of tissue; tissue with long T2 relaxation time appears brighter (hyperintense),
- T1C: T1-weighted MRI after administration of underlinecontrast media: many tumors show signal enhancement after administration of contrast agent,
- FLAIR: fluid-attenuated inversion-recovery MRI: bright signal of the CSF (cerebrospinal fluid) is suppressed which allows a better detection of small hyperintense lesions.

[...] there are dozens (or more) of other scan types; generally, many of them are available (sometimes with varying names) at practically all MRI systems.

https://www.researchgate.net/post/What_is_the_difference_between_MRI_scans_T1_T1c_T2_FlaIR_Is_there_another_type_of_scan

Outline

- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Pre-processing et extraction des données I

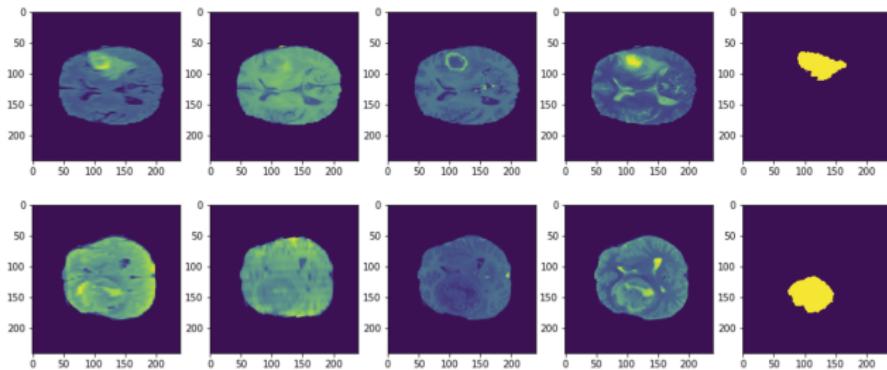


Figure: Exemples d'inputs (flair/t1/t1ce/t2)

Données fournies : 4 modalités de 285 cerveaux définies par $240 \times 240 \times 155$ voxels de valeurs entre 0 et environ 2000, plus les segmentations associées (de valeur dans $\{0, 1, 2, 4\}$).

Pour info : All the imaging datasets have been segmented manually, by one to four raters, following the same annotation protocol, and their annotations were approved by experienced neuro-radiologists. Annotations comprise the **GD-enhancing tumor (label 4)**, the **peritumoral edema (label 2)**, and the **necrotic and non-enhancing tumor core (label 1)**.

Dans notre cas, on cherchera à localiser toutes les zones de la tumeur en même temps.

Pre-processing et extraction des données II

Méthodologie :

- normalisation statistique des données de modalité (on considère si possible le cerveau en entier),

$$\forall i \in \{1, \dots, 4\}, X_{norm}^i := \frac{X^i - \mu(X^i)}{\sigma(X^i)},$$

- changement des valeurs des données de segmentation: 0 reste à 0 mais toutes les valeurs dans $\{1, 2, 4\}$ passent à 1 car correspondent à de la tumeur,
- extraction des coupes transversales 2D ($z = cte$) des modalités et des segmentations,
- on crée les données d'entrée/sortie pour notre futur modèle d'apprentissage:
 - extraction des pixels (x, y) étant à des valeurs non nulles simultanément dans les 4 modalités X^i ,
 - extraction des valeurs $\{(X_{norm}^i(x, y))\}_i$,
 - extraction de la segmentation correspondante $seg(x, y)$ à chacun de ces pixels (vérité terrain).
 - on obtient des associations:

$$\{(X_{norm}^i(x, y))\}_i \rightarrow seg(x, y)$$

On obtient finalement un tableau *input* de dimension $[N, 4]$ où N représente le nombre de pixels donnés en entrée, et 4 correspond aux modalités, et un tableau *output* de dimension $[N, 1]$ correspondant aux segmentations par les spécialistes (vérité terrain).

Outline

- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Pre-processing et extraction des données I

Set de données d'apprentissage : associations inputs/outputs permettant à notre classifieur de se calibrer : il prédit la solution, calcule l'erreur de prédiction (par une fonction de coût) avec la vérité terrain, et s'autocorrigie.

Set de données de validation : mêmes associations mais qui permettent d'évaluer la fonction de coût séparément des données d'apprentissage. Autrement dit, cela nous donne une première idée de sa capacité à segmenter correctement des données qui n'auront pas été utilisées pour l'apprentissage.

Code Python pour séparer les données préparées précédemment en 2 sets disjoints:

```
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test = train_test_split(x_all, y_all, test_size = 0.1, random_state = 0)
```

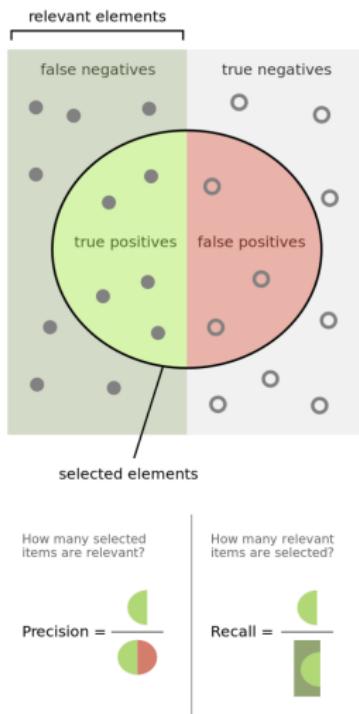
Ici, 10% des données seront choisies pour le set de validation.

Outline

- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Calculs des scores de prédition

Il existe (au moins) 3 façons de calculer la qualité d'un estimateur:



- La **précision** est le nombre de VP sur le nombre de P (nombre de pixels de tumeurs bien détectés sur le nombre de pixels indiqués comme étant de la tumeur),
- Le **rappel** est le nombre de VP sur le nombre de FN plus VP (nombre de pixels de tumeur bien détectés sur le nombre réel de pixels de la tumeur),
- Le **F-score** est la moyenne (harmonique) des deux mesures précédentes:

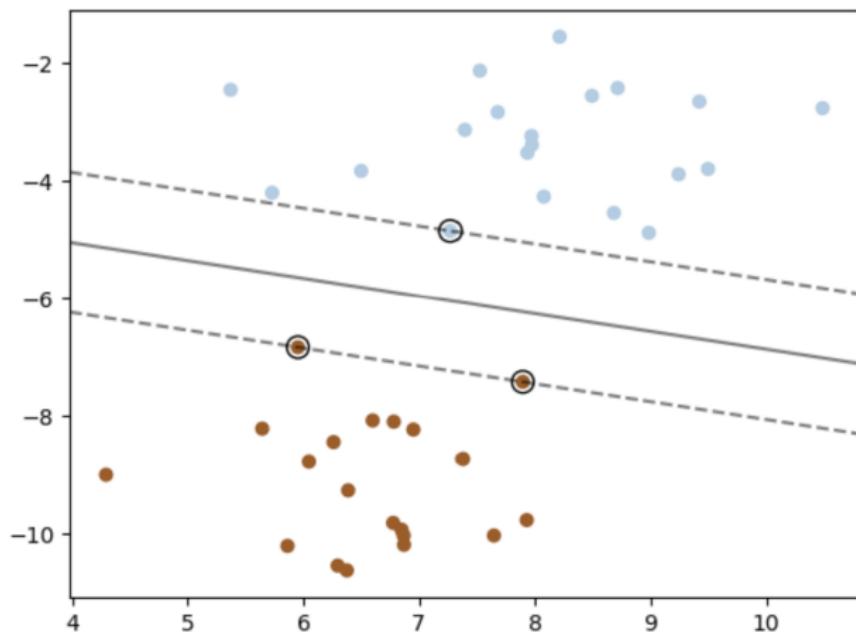
$$\text{Fscore} := 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

https://en.wikipedia.org/wiki/F1_score

Outline

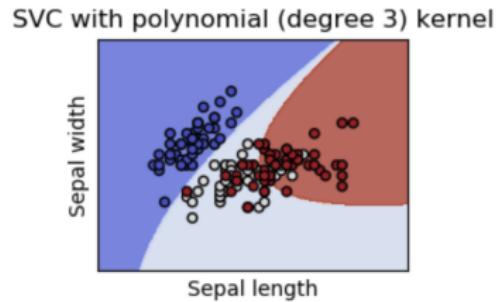
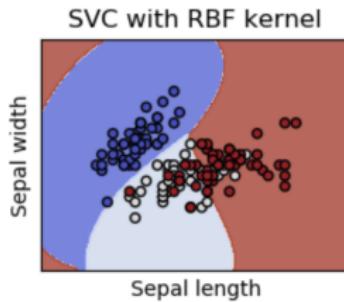
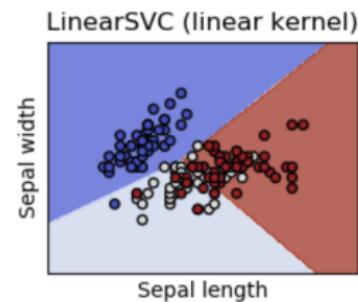
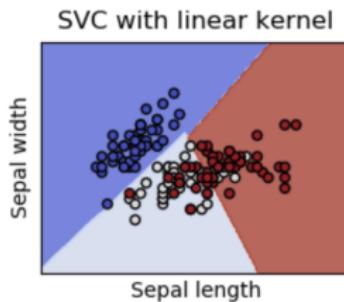
- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Support Vector Machines (SVM's) I



Support Vector Machines (SVM's) II

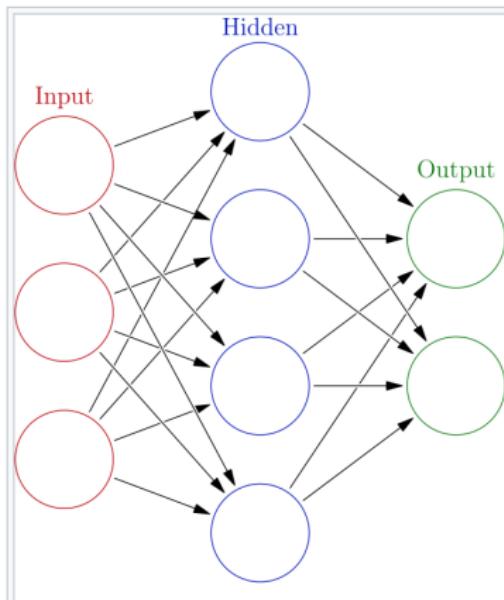
Exemples de classifications avec différents kernels :



Outline

- 1 Introduction
- 2 Anatomie cérébrales et IRM's
- 3 Prétraitement des données
- 4 Séparation des données
- 5 Scores
- 6 Rappel rapide sur les SVM's
- 7 Rappel rapide sur les ANN's
- 8 Expérimentations
- 9 Améliorations possibles

Artificial Neural Networks (ANN's)



An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

Dans notre cas, nous aurons 4 entrées si on travaille en local et $4 * (2 * K + 1)^2$ entrées si on

Outline

1 Introduction

2 Anatomie cérébrales et IRM's

3 Prétraitement des données

4 Séparation des données

5 Scores

6 Rappel rapide sur les SVM's

7 Rappel rapide sur les ANN's

8 Expérimentations

9 Améliorations possibles

Apprentissage (pixelwise) par SVM I

Commençons par utiliser une SVM.

```
from sklearn.svm import SVC  
svclassifier = SVC(kernel='linear',max_iter=10000)  
svclassifier.fit(x_train, y_train.ravel())
```

Le nombre maximal d'itérations est ici de 10.000.

On voudra alors estimer la qualité de notre estimateur par des scores ou par des observations sur les données fournies ...

Scores de la SVM I

Scores de la SVM

```
Entrée [15]: y_pred = svclassifier.predict(x_test)
```

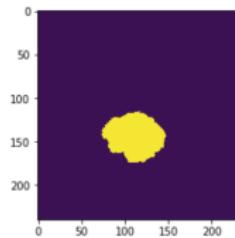
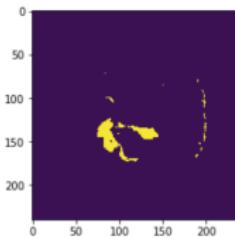
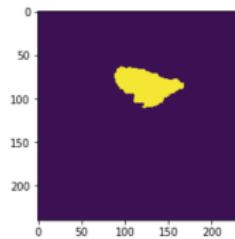
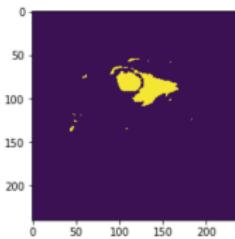
```
Entrée [16]: from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[2976  22]
 [ 273 255]]
```

	precision	recall	f1-score	support
0.0	0.92	0.99	0.95	2998
1.0	0.92	0.48	0.63	528
accuracy			0.92	3526
macro avg	0.92	0.74	0.79	3526
weighted avg	0.92	0.92	0.90	3526

Observation des résultats de segmentation I

On observe la prédiction de notre SVM sur les données d'apprentissage dans un premier temps.



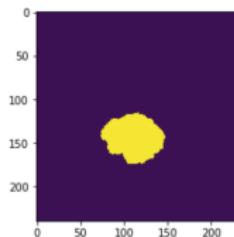
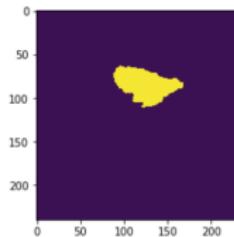
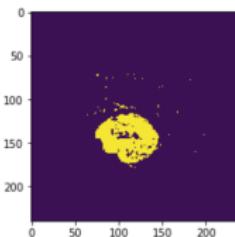
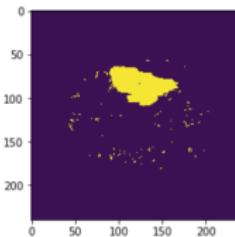
Question : la linéarité du kernel de la SVM est-elle trop contraignante ? On va essayer les RBF (Radial Basis Functions) pour courber les séparatrices.

Et avec une SVM à base de RBF's ? I

Bon point : plusieurs paramètres sont alors possibles (C la constante, γ la puissance, etc.).

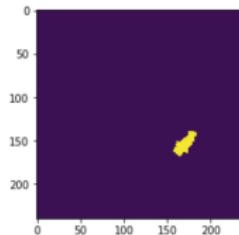
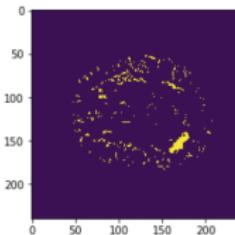
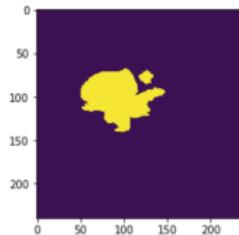
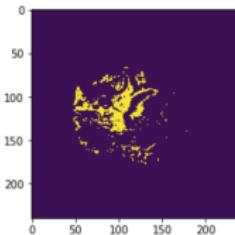
Mauvais point: il faut rechercher la combinaison de paramètres optimale pour notre cas.

On cherche alors la combinaison de C et γ qui maximise le F-score, et on obtient alors les résultats suivants:



Et avec une SVM à base de RBF's ? II

Les données d'apprentissage sont beaucoup mieux segmentées, mais qu'en est-il sur des nouvelles données ? (c'est-à-dire sur des cerveaux encore jamais vus mais dont on a les vérités terrains)



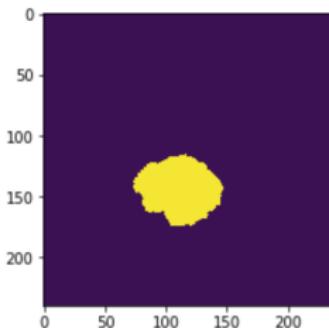
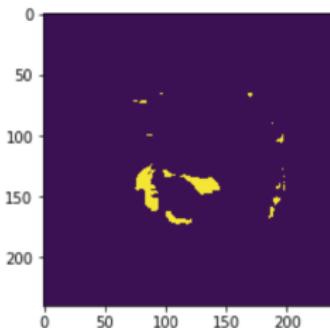
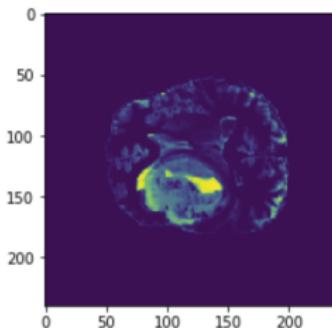
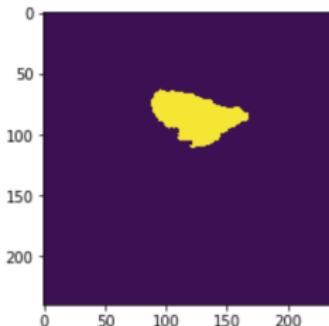
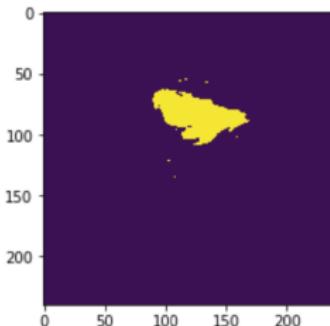
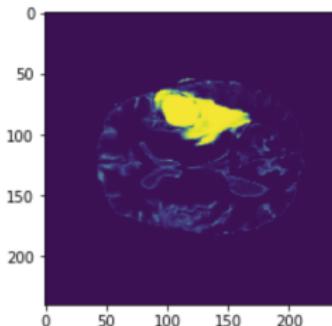
La prédiction sur des nouveaux cas est très mauvaise, mais comment peut-on alors modifier le modèle ou les entrées pour être plus efficace ?

Approche à base de réseau de neurones I

Code python d'un ANN:

```
ANN = models.Sequential()
regul = 1e-6
ANN.add(Dense(16, activation='relu', kernel_regularizer=l1(regul), bias_regularizer=l1(regul), kernel_initializer='uniform'))
ANN.add(Dense(4, activation='relu', kernel_regularizer=l1(regul), bias_regularizer=l1(regul), kernel_initializer='uniform'))
ANN.add(Dense(1, activation='sigmoid', kernel_regularizer=l1(regul), bias_regularizer=l1(regul), kernel_initializer='uniform'))
ANN.compile(optimizer='Adam',
            loss='binary_crossentropy',
            metrics=['accuracy'])
)
ANN.fit(x_train, y_train,
        batch_size=256, #petits réseaux => 8/16.../256 en général
        epochs=10,
        verbose=1,
        validation_split=0.1
)
```

Approche à base de réseau de neurones II



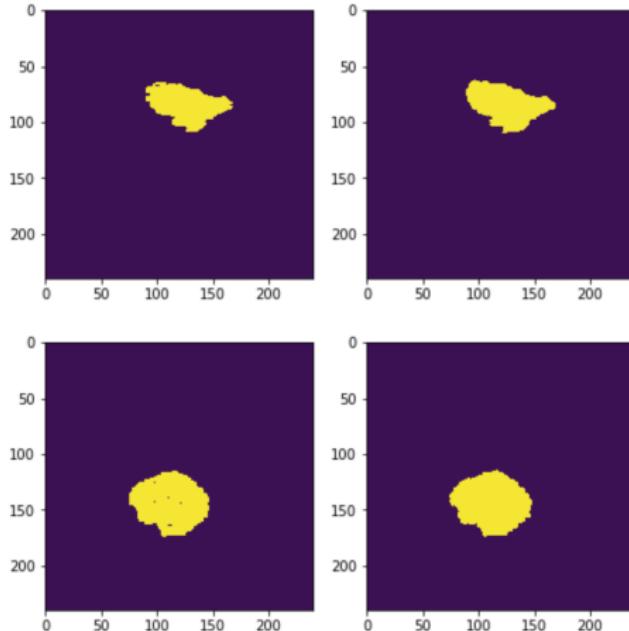
On aura seuillé à 0.5 les valeurs en sortie de l'ANN pour obtenir soit 0 soit 1 en sortie.

Approche à base de réseau de neurones III

Cette approche là ne marche pas non plus, peut-être manque-t-on d'informations pour classifier correctement ? On va donc se servir des pixels sur un voisinage (données plus complètes mais plus lourdes à traiter, on va donc rester avec les ANN's, plus rapides).

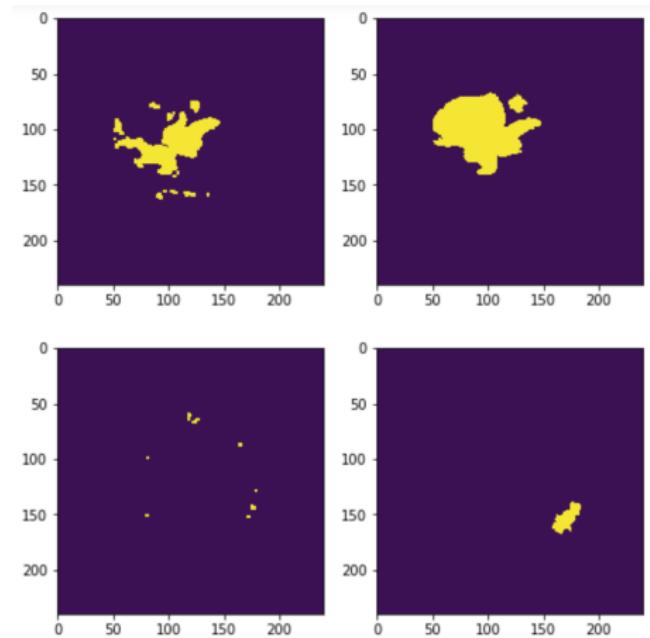
On va maintenant rentrer comme données pour (x, y) de X_{norm}^i l'ensemble $[x - 1 : x + 2, y - 1 : y + 2]$ ($K = 1$), ce qui fait $9 * 4$ données en entrée par pixel. On obtient alors sur les données d'apprentissage :

Approche à base de réseau de neurones IV



Approche à base de réseau de neurones V

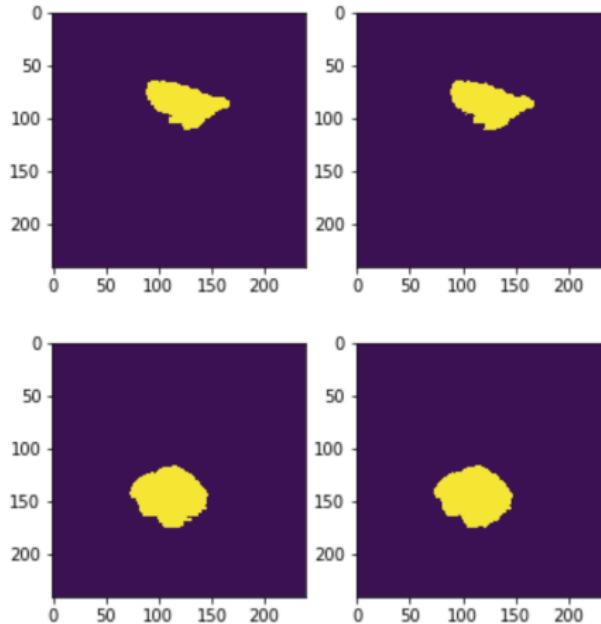
Maintenant que l'on obtient de bons résultats sur les données d'apprentissage, on regarde à nouveau les résultats sur les données tests:



Approche à base de réseau de neurones VI

On voit alors que les résultats à base de voisinage donnent des résultats moins bons sur la deuxième ligne (cas non appris), on est donc plus discriminants qu'avant (ce qui est a priori un bon point en soi: les données sont plus séparées dans l'espace des features), mais en même temps on a fait de l'overfitting: le système ne sait plus que réciter par coeur ce qu'il a appris, mais ne sait plus prédire. On va donc essayer de régulariser les poids de notre réseau (minimisation L^1). On obtient alors sur les données d'apprentissage :

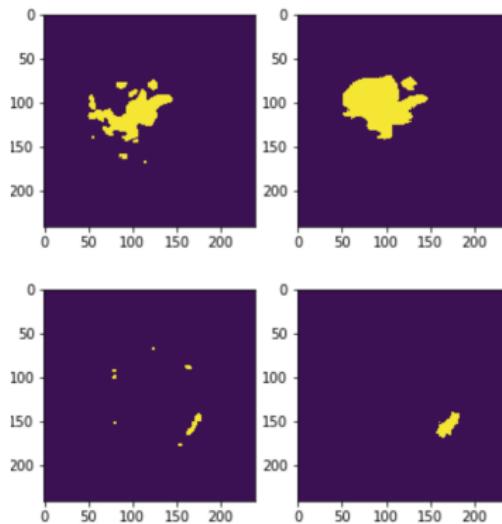
Approche à base de réseau de neurones VII



Le résultat reste correct.

Approche à base de réseau de neurones VIII

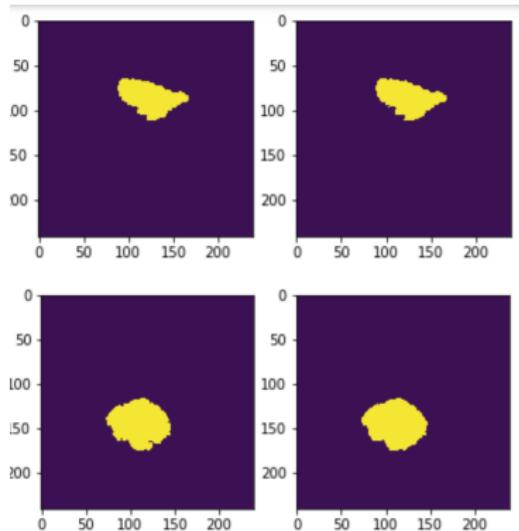
On regarde à nouveau le résultats sur les données tests:



On retrouve une bonne segmentation à la deuxième ligne, ce qui est bon signe, mais la première ligne montre que cette régularisation n'a pas été suffisante.

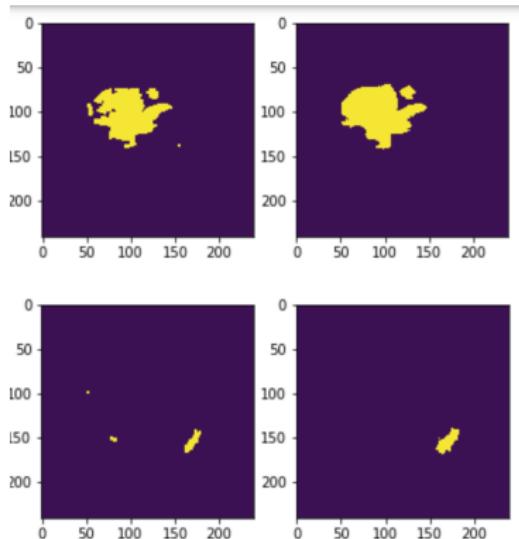
Approche à base de réseau de neurones IX

Il reste une solution: augmenter le set de données. On s'est servi de 10 cerveaux pour l'apprentissage pour le moment, on va maintenant en utiliser 30, cela afin de compenser l'hétérogénéité des tumeurs dans le cerveau (au niveau de leurs valeurs de modalités). On obtient alors sur le set d'apprentissage:



Approche à base de réseau de neurones X

Et sur les données tests on obtient:



On obtient enfin un résultat intéressant !

Outline

1

Introduction

2

Anatomie cérébrales et IRM's

3

Prétraitement des données

4

Séparation des données

5

Scores

6

Rappel rapide sur les SVM's

7

Rappel rapide sur les ANN's

8

Expérimentations

9

Améliorations possibles

Possibilités pour améliorer le modèle :

- augmenter le nombre de cerveaux d'apprentissage,
- augmenter la taille du voisinage pour augmenter le nombre de features d'entrées ? (attention, ne deviendra-t-on pas trop discriminant ?),
- améliorer la structure de l'ANN en fonction du contexte (nombre de couches, nombre de neurones par couche),
- changement de l'espace de représentation (transformées de Fourier, transformées en ondelettes, représentations morphologiques, etc.),
- sachant que les composantes des tumeurs sont simplement connexes en pratique, pourquoi ne pas utiliser un opérateur qui "bouche" les cavités ?
- utiliser d'autres classifieurs (UNet, RandomForests, etc.)

A vous de jouer !