

Polytech Nice-Sophia Antipolis

4^{ème} année d'école d'ingénieur – Mathématiques Appliquées et Modélisation

Méthode de classification hiérarchique d'images de planctons par apprentissage profond

Auteur : Baptiste POUTHIER

Année universitaire : 2017 / 2018

Tuteur de stage : Éric DEBREUVE

Professeur responsable : Frédéric PRECIOSO

Organismes d'accueil : Laboratoires I3S / INRIA – Equipe mixte MORPHEME

Remerciements :

Je tiens à remercier vivement mon tuteur, Monsieur Éric DEBREUVE, pour m'avoir aidé pendant toute la durée de ce stage. Ses enseignements et ses conseils m'ont été très précieux et j'ai beaucoup appris à ses côtés.

J'adresse également mes remerciements à Monsieur Frédéric PRECIOSO, mon professeur responsable, grâce à qui j'ai trouvé ce stage en adéquation avec toutes mes attentes et qui m'a également accompagné tout au long du projet.

Aussi, je veux remercier toute l'équipe mixte INRIA-I3S « MORPHEME » pour son accueil et en particulier Mademoiselle Sarah LAROUÏ, Mademoiselle Anca Grapa, Monsieur Kévin Giulietti et Monsieur Raphaël Pagès. Votre aide et votre bonne humeur ont fait de ce stage une expérience aussi enrichissante qu'agréable.

Sommaire

1. Introduction et présentation de l'entreprise	3
a. Présentation de l'organisme d'accueil	3
b. Contexte du projet	3
c. Sujet de stage	3
2. Travail proposé	4
a. Problèmes posés	4
b. Présentation du projet	4
c. Sujet associé	5
3. Travail réalisé	5
a. Résultats	5
b. Difficultés rencontrés	7
c. Changement apportés aux objectifs initiaux	8
4. Conclusion	9
5. Bibliographie	10
6. Annexes	11

1. Introduction et présentation de l'entreprise

Dans le cadre de la quatrième année du parcours ingénieur à Polytech Nice-Sophia Antipolis en spécialité « Mathématiques Appliquées et Modélisation », j'ai réalisé un stage au sein d'INRIA¹ Sophia Antipolis et du laboratoire I3S². Ce rapport détaille le projet effectué durant cette expérience ainsi que son contexte et ses enjeux.

Dans ce dossier, tous les termes marqués d'un astérisque sont définis en annexe. Les mots suivis d'un numéro entre parenthèse font référence à la bibliographie tandis que les sigles et acronymes sont définis en fin de page.

a. Présentation de l'organisme d'accueil

Du 18 juin 2018 au 09 septembre 2018, j'ai effectué un stage d'assistant ingénieur au sein de l'équipe mixte INRIA-I3S MORPHEME à Sophia Antipolis, dirigée par Monsieur Xavier Descombes (directeur de recherche INRIA). Les axes de recherche principaux de cette équipe sont la caractérisation et la modélisation des propriétés morphologiques de structures biologiques. Le développement de solutions d'analyse d'images et de modèles de prédiction est prépondérant à leur expertise.

b. Contexte du projet

Aujourd'hui en vogue, l'apprentissage automatique* constitue un champ d'étude ou d'application dans de très nombreux secteurs tels que le sport, la finance ou l'automobile. Et le secteur de la biologie n'est pas en reste : la classification d'images de structures biologiques constitue, par exemple, un réel enjeu dans ce domaine. C'est dans ce contexte que le Laboratoire d'Océanographie de Villefranche-sur-Mer (LOV) cherche à automatiser la classification d'images de planctons. Ces organismes, omniprésents au sein de la biosphère, sont à la fois le premier maillon de la chaîne alimentaire marine et l'une des sources principales d'oxygène de la planète. Par conséquent, leur étude est clé en biologie et en écologie.

c. Sujet du stage

L'objectif est double :

- 1) développer un modèle prédictif permettant d'automatiser la classification d'images de planctons en améliorant les scores des étudiants ayant travaillé sur le sujet précédemment (1).
- 2) expérimenter des structures hiérarchiques d'un réseau neuronal convolutif* (CNN) ; en effet, les recherches récentes concernant l'apprentissage profond* ont mis en évidence que les structures hiérarchiques améliorent les résultats des algorithmes existants (2)(3).

Ainsi, le but du stage est de développer une méthode de classification hiérarchique d'images de planctons par apprentissage profond.

¹ Institut National de Recherche en Informatique et en Automatique

² Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis

2. Travail proposé

a. Problèmes posés

Le LOV a déjà développé des méthodes de classification fondées sur l'utilisation de forêts d'arbres décisionnels* (Random Forest). Toutefois, les performances de ce procédé ne sont pas à la hauteur des attentes des biologistes de l'environnement. Afin d'extraire des images les linéaments (features) utiles aux algorithmes décisionnels, ils ont essayé de combiner leur procédé à un réseau convolutif mais n'ont pas eu des résultats satisfaisants.

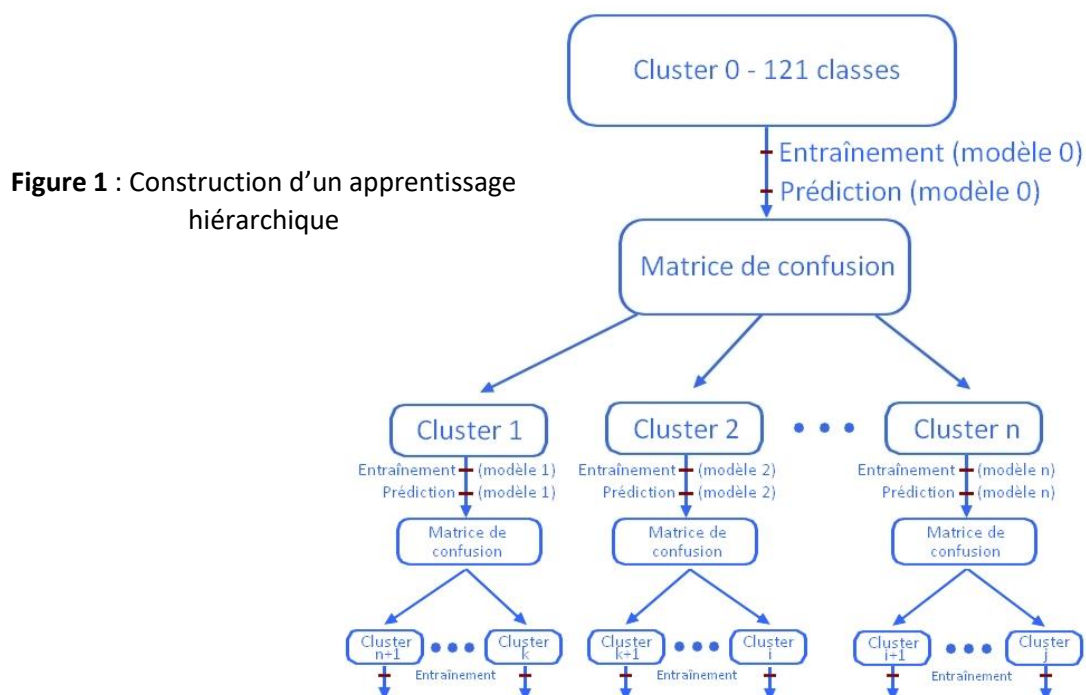
Les travaux réalisés précédemment (1) ont montré que l'utilisation d'un réseau convolutif optimisé permettait d'obtenir des résultats corrects mais cependant insuffisants pour que le LOV puisse utiliser l'algorithme aveuglement. Il est donc nécessaire de développer un nouvel algorithme tirant profit des bons résultats du CNN tout en améliorant ses résultats.

Des images de planctons, issues de 121 classes, sont fournies par le LOV. Cela constitue la collection de données utile à l'apprentissage d'un réseau convolutif. Ces images nécessitent un traitement particulier en amont (redimensionnement sans déformation, data augmentation*) pour permettre un apprentissage efficace.

b. Présentation du projet

L'idée du projet est d'ajouter une dimension hiérarchique au réseau convolutif existant. Pour cela, nous avons utilisé des matrices de confusion* pour déterminer quelles espèces sont fortement confondues dans les prédictions du CNN. Ces espèces sont alors regroupées et forment des clusters de forte confusion. Chacun des groupes ainsi créés va être réutilisé pour apprendre un réseau convolutif spécifique. La procédure s'arrête lorsque tous les groupes sont composés de deux classes ou si un groupe ne s'affine pas après un entraînement du CNN.

De cette manière, chaque cluster possède un modèle prédictif entraîné. A l'issue de cet entraînement, un algorithme de prédiction doit prédire l'espèce de chaque image à l'aide des modèles prédictifs entraînés la concernant.



c. Sujet associé

Il est possible d'envisager d'autres types d'architectures hiérarchiques telles que des architectures en arbres suivant la taxonomie du plancton ou des architectures utilisant des linéaments propres à des groupes « grossiers » conjointement avec les linéaments relatifs aux groupes finement triés (2)(3). Seulement, l'emploi de ces architectures nécessite un dataset présentant plusieurs degrés de labélisation (espèce, sous-espèce, ...). Actuellement, les informations et la collection de données fournies par le LOV sont insuffisantes pour réaliser ces hiérarchies.

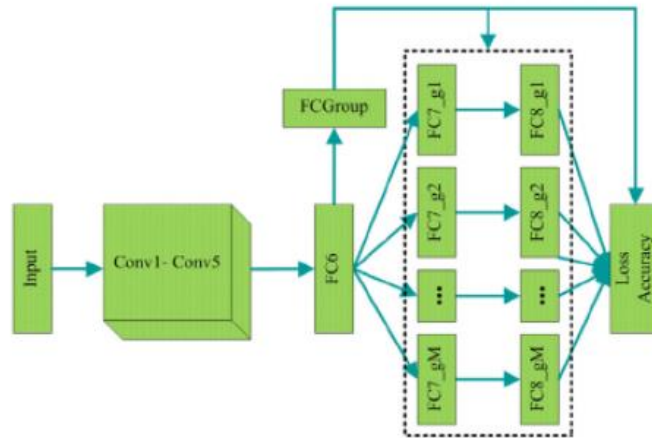


Figure 2 : CNN avec structure hiérarchique
Source : (2)

3. Travail réalisé

a. Résultats

- Entraînement :

Chaque cluster est entraîné via un réseau de neurones profond.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 95, 95, 32)	320
leaky_re_lu_1 (LeakyReLU)	(None, 95, 95, 32)	0
conv2d_2 (Conv2D)	(None, 93, 93, 16)	4624
leaky_re_lu_2 (LeakyReLU)	(None, 93, 93, 16)	0
max_pooling2d_1 (MaxPooling2)	(None, 46, 46, 16)	0
dropout_1 (Dropout)	(None, 46, 46, 16)	0
conv2d_3 (Conv2D)	(None, 46, 46, 64)	9280
leaky_re_lu_3 (LeakyReLU)	(None, 46, 46, 64)	0
conv2d_4 (Conv2D)	(None, 44, 44, 32)	18464
leaky_re_lu_4 (LeakyReLU)	(None, 44, 44, 32)	0
max_pooling2d_2 (MaxPooling2)	(None, 21, 21, 32)	0
dropout_2 (Dropout)	(None, 21, 21, 32)	0
conv2d_5 (Conv2D)	(None, 21, 21, 128)	36992
leaky_re_lu_5 (LeakyReLU)	(None, 21, 21, 128)	0
conv2d_6 (Conv2D)	(None, 19, 19, 128)	147584
leaky_re_lu_6 (LeakyReLU)	(None, 19, 19, 128)	0
conv2d_7 (Conv2D)	(None, 17, 17, 64)	73792
leaky_re_lu_7 (LeakyReLU)	(None, 17, 17, 64)	0
max_pooling2d_3 (MaxPooling2)	(None, 8, 8, 64)	0
dropout_3 (Dropout)	(None, 8, 8, 64)	0
conv2d_8 (Conv2D)	(None, 8, 8, 256)	147712
leaky_re_lu_8 (LeakyReLU)	(None, 8, 8, 256)	0
conv2d_9 (Conv2D)	(None, 6, 6, 256)	590080
conv2d_10 (Conv2D)	(None, 4, 4, 128)	295040
leaky_re_lu_9 (LeakyReLU)	(None, 4, 4, 128)	0
max_pooling2d_4 (MaxPooling2)	(None, 1, 1, 128)	0
dropout_4 (Dropout)	(None, 1, 1, 128)	0
flatten_1 (Flatten)	(None, 128)	0
dense_1 (Dense)	(None, 512)	66048
leaky_re_lu_10 (LeakyReLU)	(None, 512)	0
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 121)	62073
Total params: 1,452,009		
Trainable params: 1,452,009		
Non-trainable params: 0		

Figure 3 : Couches du réseau de neurone convolutif (architecture proposée par Keras)

Les couches ci-dessus sont : des couches de convolution (conv2d), des fonctions d'activation* (leaky_re_lu), des couches de pooling* (max_pooling2d), de dropout*, de mise à plat* (flatten) ou des couches « fully connected »* (dense).

- **Analyse :**

A l'issue des entraînements de chaque groupe, on construit des matrices de confusion. Une fois normalisées (par ligne), celles-ci indiquent les pourcentages des éléments de la classe i étant classifiés j . Ainsi, on remarque sur la diagonale les éléments bien classés. On extrait les classes fortement mélangées en prenant les hauts pourcentages, comparés à un seuil, présents dans la matrice en dehors de la diagonale.

Ces classes composent les clusters qui seront à leur tour entraînés et dont on va extraire de nouvelles matrices de confusion.

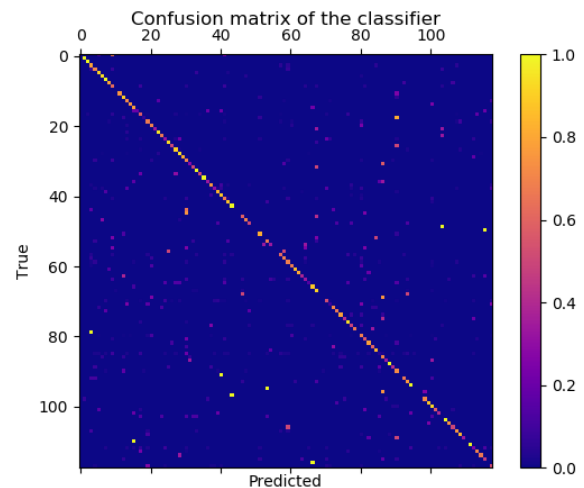


Figure 4 : Matrice de confusion pour 121 classes

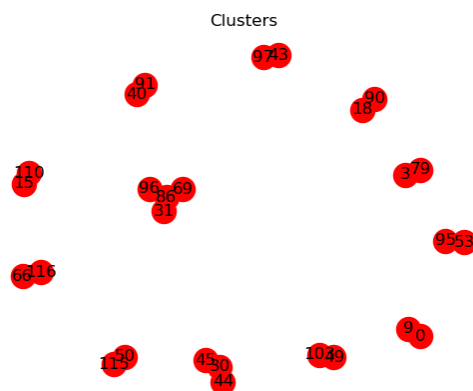


Figure 5 : Clusters sous forme de graphes

La formation des groupes est effectuée en utilisant les graphes : des liaisons vont être créées entre les classes (représentées par des nœuds) fortement confuses ; chaque sous-graphe ainsi créé formera un groupe.

Note : Les graphes sont très utilisés en science des données pour créer des groupes d'objets tout en ayant une représentation graphique claire et organisée.

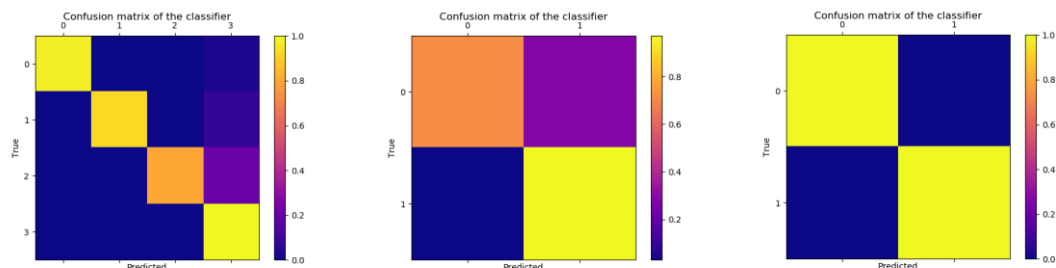


Figure 6 : Matrice de confusion de clusters (composés de respectivement 4, 2 et 2 classes)

Dans le cas précédent, le groupe formant la matrice de taille 4 sur la figure 6 va être re-entraîné pour affiner le résultat si une valeur non diagonale est trop importante. Pour suivre l'architecture hiérarchique et l'évolution des clusters il est possible d'utiliser les graphes une nouvelle fois.

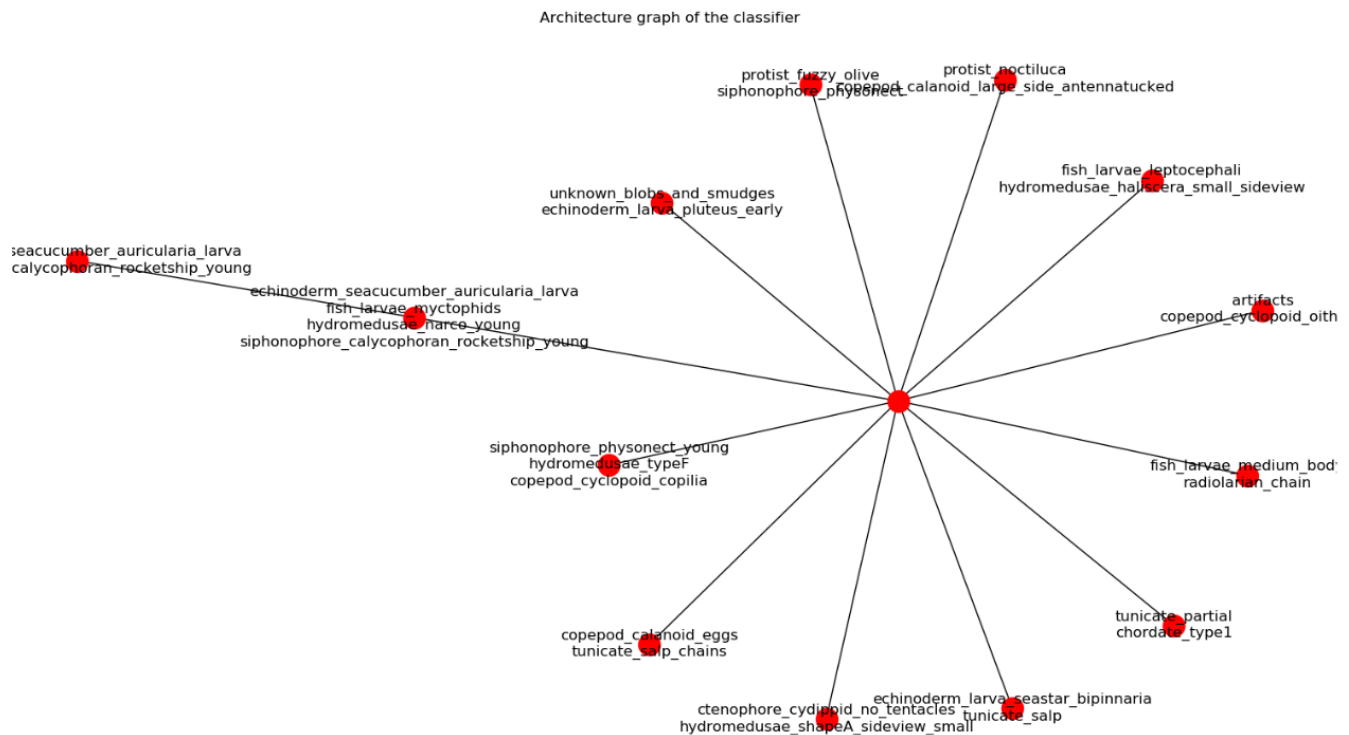


Figure 7 : Architecture des clusters créés

- **Prédictions et scores**

Afin de prédire les classes des images et ainsi tester le modèle, il faut prédire successivement chaque image avec tous les modèles qui composent sa hiérarchie. A l'issue de mon stage, les scores obtenues avec le classifieur hiérarchique égalent les scores du classifieur CNN seul (70%). Ce résultat est encourageant car il a été obtenu sans optimiser le seuil des matrices de confusion ni les paramètres propres au CNN ; il appuie donc l'idée que les architectures hiérarchiques peuvent améliorer les réseaux classiques.

b. Difficultés rencontrés

La principale difficulté rencontrée concerne le temps de procédure : l'entraînement des différents réseaux composant la hiérarchie prend beaucoup de temps. J'ai eu la chance d'avoir un accès invité à NEF, le serveur de calcul de l'INRIA. Cela permet de réduire drastiquement le temps de calcul : avec cette solution, il faut environ 19h pour un entraînement complet de tous les modèles de la structure hiérarchique.

Seulement, mon statut d'invité sur la plateforme NEF implique qu'un utilisateur permanent peut, si besoin, arrêter sans sauvegarde un de mes calculs. Lors d'un calcul long, cela est courant et il devient difficile de mener une procédure à terme.

c. Changement apportés aux objectifs initiaux

Pour vérifier le bon fonctionnement du programme et plus précisément celui des matrices de confusion, j'ai remplacé le classifieur CNN par un classifieur SVM* « one versus all ». En effet, au cours de mon stage, j'ai obtenu des matrices de confusion erronées et il fallait trouver si cela provenait du classifieur ou d'une mauvaise implémentation.

J'ai d'abord vérifié que le programme fonctionnait correctement avec un nuage de points.

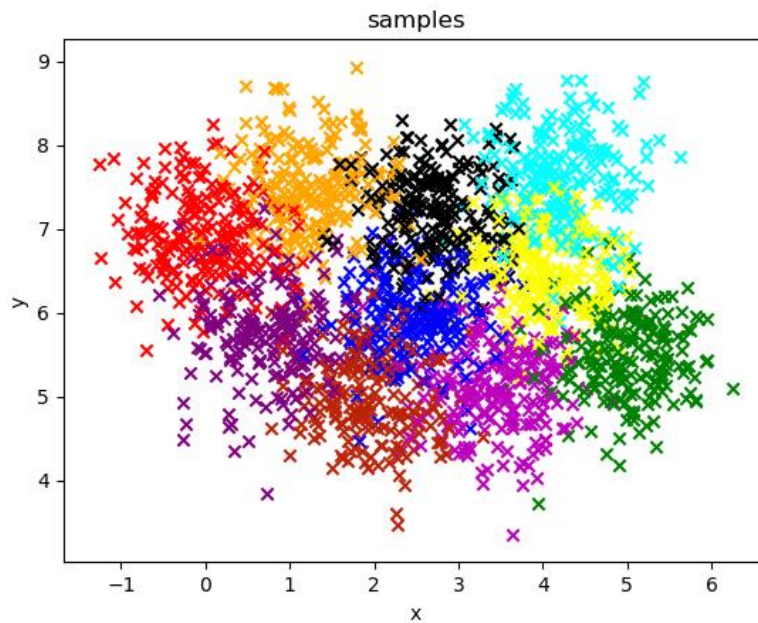


Figure 8 : Nuage de points (10 classes)

En travaillant dans un cadre plus simple, il est plus facile de vérifier le bon fonctionnement des méthodes implémentées ; les matrices de confusion ainsi que le graphe d'architecture ci-dessous montrent que les résultats sont cohérents et que les matrices de confusion sont effectives.

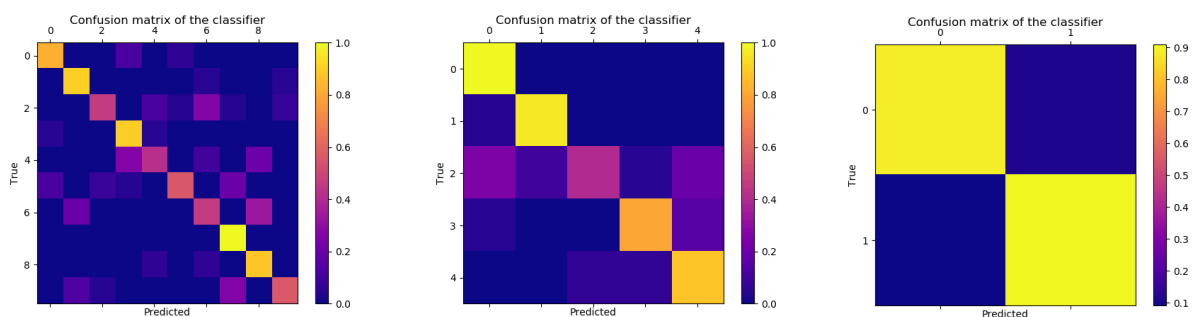


Figure 9 : Matrices de confusion pour un nuage de points (10 classes)

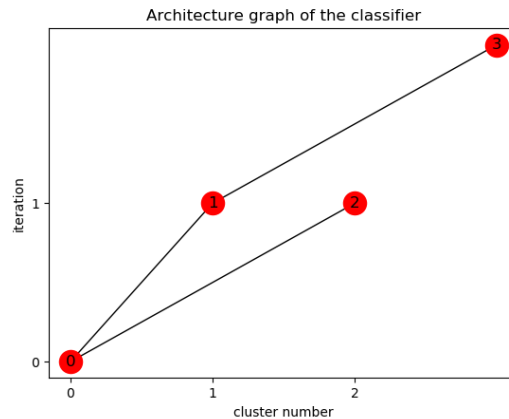


Figure 10 : Architecture des clusters créés

Le problème ne venant pas de l'analyse, elle venait probablement du classifieur. En utilisant le « transfer learning », les résultats des matrices de confusion ont été grandement améliorés : au lieu d'entraîner les réseaux à partir d'une initialisation aléatoire, on leur transmet les poids des modèles relatifs aux clusters pères.

J'ai également mis à l'épreuve le classifieur SVM avec les images de plancton. Pour ce faire, on extrait en amont les features « SIFT »* des images. Les résultats ont été décevants (30%) et cela appuie l'intérêt des CNN dans des problèmes comme celui-ci (avec beaucoup de classes et des données sous formes d'images). Les expérimentations ont démontré que les SVM étaient cependant performants pour classifier des groupes avec peu de classe. Au contraire, les CNN sont peu adaptés lorsque le dataset est petit. Ainsi, les SVM sont utilisés à l'instar du réseau neuronal lorsqu'un groupe possède peu d'images.

4. Conclusion

Ce stage a été pour moi l'opportunité de gagner en compétence en apprentissage automatique et plus spécifiquement en apprentissage profond. Aussi, j'ai beaucoup appris en programmation Python et plus particulièrement au niveau de l'organisation du code. Ayant utilisé NEF, j'ai compris comment utiliser un serveur de calcul OAR.

J'ai également eu la chance de découvrir l'univers de la recherche et les travaux réalisés par les membres de l'équipe MORPHEME. Ce milieu m'a beaucoup enthousiasmé, d'autant plus que le sujet de mon stage m'a vraiment intéressé. J'ai eu la chance d'assister à plusieurs séminaires, à chaque fois très intéressantes et j'ai trouvé ces événements très positifs et motivants.

Cette expérience m'a conforté dans l'idée de continuer à découvrir le domaine de l'intelligence artificielle et plus précisément l'apprentissage automatique. Dans cette optique, je pense m'orienter une nouvelle fois dans le machine learning via un prochain stage.

5. Bibliographie

- (1) Cédric Bailly, Thomas Mahiout, Thomas Jalabert – Classification d’images de plancton par Deep Learning ou apprentissage hybride Deep Features/Random Forest – PFE 2016/2017
- (2) Zhenzhong Kuang, Jun Yu, Zongmin Li, Baopeng Zhang, Jianping Fan – Integrating multi-level deep learning and concept ontology for large-scale visual recognition – Pattern Recognition – ScienceDirect – 2018 – N° 78
- (3) Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, Yizhou Yu – HD-CNN : Hierarchical Deep Convolutional Neural Network for Large Scale Visual Recognition – arXiv.org – Article ID 410.0736v4 – 16 mai 2015

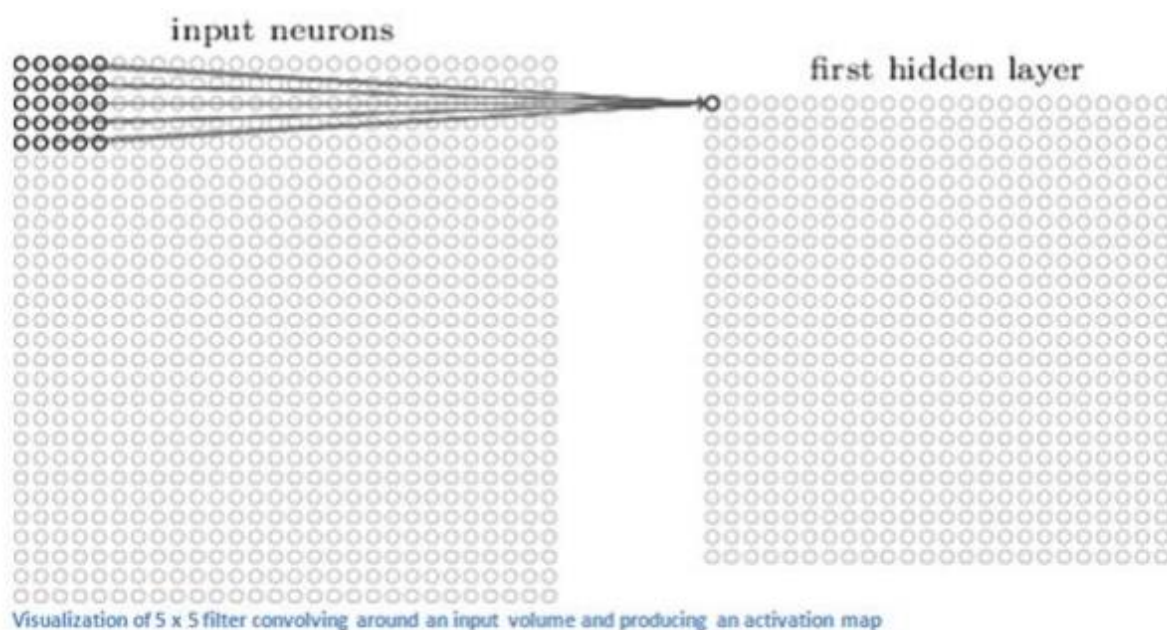
6. Annexes

Dépôt du projet : <https://github.com/baptistepouthier/plancton>

Apprentissage automatique : Le « Machine Learning » fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.

Source : <http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bonastre/ApprentissageAutoIntroductionI.pdf>

Réseaux neuronal convolutif : Inspirés des neurones biologiques, les réseaux de neurones à convolution sont composés de différentes couches (input layer, hidden layer*, output layer) dont certaines sont des couches de convolution. En utilisant des filtres* qui parcourent la matrice d'entrée (les valeurs en niveau de gris des pixels de l'image), un produit terme à terme est effectué entre les valeurs du filtre et celles d'entrées. Pour chaque filtre, les produits obtenus sont sommés. Chaque filtre produit une « feature map » qui représente sa réponse à chaque position de l'image. En superposant ces filtres, on obtient la matrice de sortie.



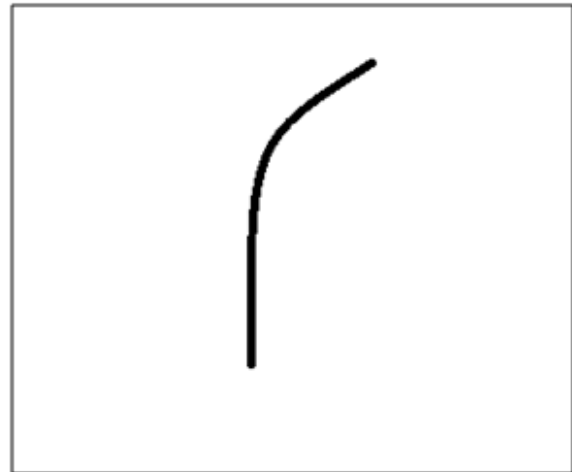
Source : <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>

Hidden Layer (couche cachée) : le rôle d'une couche cachée est d'appliquer une transformation aux données permettant à la couche de sortie de les utiliser.

Filtre : Un filtre est une matrice de pixels représentant une forme.

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

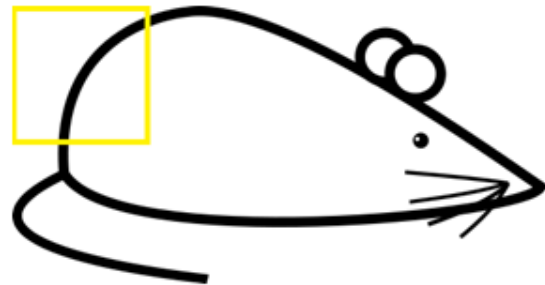
Pixel representation of filter



Visualization of a curve detector filter



Original image



Visualization of the filter on the image



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$ (A large number!)

Source : <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>

Apprentissage profond : Le « Deep Learning » est une branche de l'apprentissage automatique basée sur les réseaux de neurones profonds ; les réseaux de neurones convolutifs sont un type d'architecture de réseaux de neurones profonds.

Forêt d'arbres décisionnels : C'est une technique d'apprentissage automatique qui effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

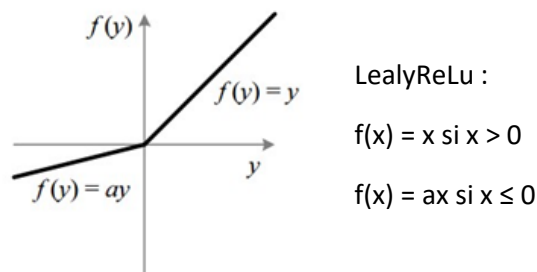
Source : https://fr.wikipedia.org/wiki/For%C3%AAt_d%27arbres_d%C3%A9cisionnels

Data Augmentation : L'augmentation de données est une technique d'ajout de données à un dataset en transformant de diverses manières (rotation, ajout de bruits, ...) les données de base.

Matrice de confusion : Outil servant à mesurer la qualité d'un système de classification. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle.

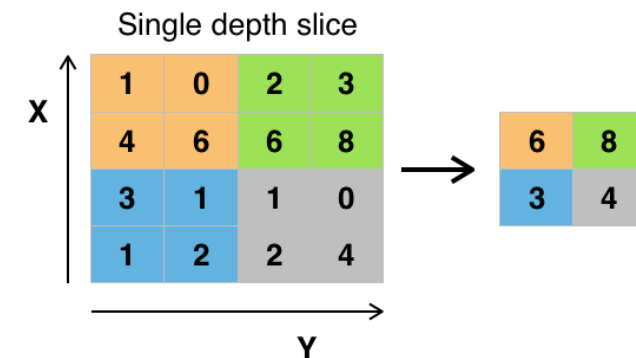
Source : https://fr.wikipedia.org/wiki/Matrice_de_confusion

Fonction d'activation : Fonction appliquée en sortie d'un neurone artificiel (ici LeakyReLU). Son rôle est essentiellement d'assurer la non-linéarité des neurones et de garantir la flexibilité et l'expressivité des CNN.



Source : https://ml4a.github.io/ml4a/fr/neural_networks/

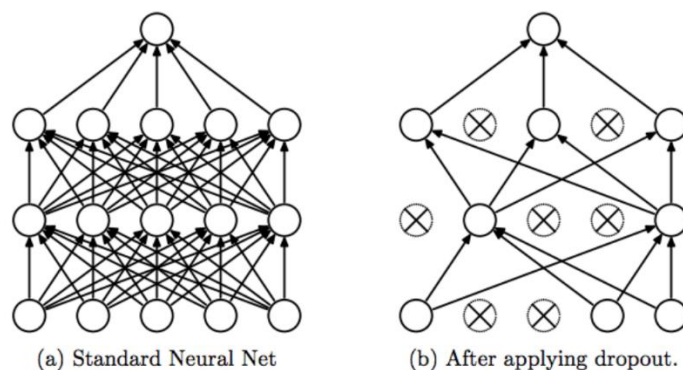
Pooling : Le pooling sert à réduire la dimension de la matrice mise en entrée, tout en conservant l'invariance en translation et les informations importantes de la matrice. Pour cela, la méthode de pooling la plus largement utilisée est le « Max Pooling » qui extrait la valeur la plus grande de chaque subdivision de la matrice d'entrée.



Example of Maxpool with a 2x2 filter and a stride of 2

Source : <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>

Dropout : Supprime un ensemble de neurones aléatoirement choisis. Evite le surentraînement et réduit le nombre de contraintes sur les neurones pour assurer le bon fonctionnement du réseau.



Source : <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>

Mise à plat (flatten) : transforme la dimension de la matrice pour la transformer en un vecteur.

Fully connected : couche entièrement connectée à la précédente d'une taille préalablement définie.

SVM : Les machines à vecteurs de support (Support Vector Machine) sont un ensemble de technique d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression.

Source : https://fr.wikipedia.org/wiki/Machine_%C3%A0_vecteurs_de_support

SIFT : SIFT signifie « Scale-Invariant Feature Transform », que l'on peut traduire par « transformation de caractéristiques visuelles invariante à l'échelle ». C'est un algorithme utilisé dans le domaine de la vision par ordinateur pour détecter et identifier les éléments similaires entre différentes images numériques. L'étape fondamentale consiste à calculer ce qu'on appelle les « descripteurs SIFT ».

Source : https://fr.wikipedia.org/wiki/Scale-invariant_feature_transform