

Exploration interne de la structure conceptuelle des données

Une image contenant texte

Description générée automatiquementMaxime Haurel | L3 MIASHS parcours Sciences Cognitives | Mai - Juillet 2022

Sous la direction de : Mathieu D’Aquin

# Table des matières

[Table des matières 1](#_Toc109133966)

[Introduction 2](#_Toc109133967)

[Présentation de l’entreprise 3](#_Toc109133968)

[Présentation du projet global 5](#_Toc109133969)

[Prérequis 5](#_Toc109133970)

[Enjeux de l’interprétabilité 5](#_Toc109133971)

[Description du projet 5](#_Toc109133972)

[Heatmaps 6](#_Toc109133973)

[Similarité 6](#_Toc109133974)

[Travaux existants 7](#_Toc109133975)

[Présentation du travail réalisé 9](#_Toc109133976)

[Science des données 9](#_Toc109133977)

[Récolte des données 9](#_Toc109133978)

[Traitement des données 10](#_Toc109133979)

[Exploration des activations du modele 11](#_Toc109133980)

[logiciel de visualisation 11](#_Toc109133981)

[Conclusion 11](#_Toc109133982)

[competences developpees 11](#_Toc109133983)

[Glossaire 12](#_Toc109133984)

[Annexes 13](#_Toc109133985)

[Remerciements 14](#_Toc109133986)

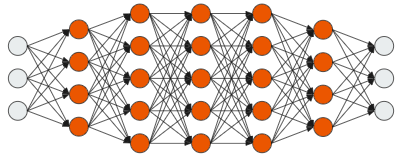
[Quatrième de couverture (changer titre) 15](#_Toc109133987)

# Introduction

Le domaine de l’intelligence artificielle (IA) est en plein essor et n’a presque jamais connu d’hiver. Le nombre de recherches en IA n’a jamais été aussi conséquent et les applications de l’IA touchent tous les domaines, de la santé à l’agriculture en passant par l’éducation.

Alors que les systèmes experts utilisant des règles bien définies étaient les systèmes rois il y a 20 ans, aujourd’hui c’est les modèles statistiques qui sont les plus utilisés dans l’industrie et les plus connus sont les réseaux de neurones artificiels. Ces modèles fonctionnent très bien pour un grand nombre de tâches et c’est la principale tendance en IA connue sous le nom de Deep Learning.

Bien que ces réseaux de neurones soient largement utilisés de partout, ils ont aussi un ensemble de désavantages dont un est **l’inexplicabilité**. En effet, les réseaux de neurones permettent d’abstraire les entrées reçues dans les couches cachées qui composent le réseau.



Ainsi, comme dans la psychologie behavioriste, on ne s’intéresse qu’aux entrées et aux sorties et les couches cachées du réseau (en orange ci-dessus) sont la boîte noire dont, à l’heure d’aujourd’hui, nous ne sommes pas en mesure de comprendre les rouages.

Ces réseaux peuvent être utilisés en production pour prédire le comportement d’un utilisateur. Mais bien que ce cas soit relativement basique, pour une utilisation dans le domaine de la santé, on aimerait savoir en détail pourquoi et comment le réseau a qualifié une radiographie pulmonaire comme laissant présager un cancer proche par exemple.

C’est donc presque dans cet intérêt que le stage s’inscrit. La vision de mon maître de stage est portée sur le comparaison des activations d’une couche cachée d’un réseau de neurones à une connaissance bien définie.  
Dans cette optique, il est possible de penser plus loin en essayant par exemple de corriger le réseau sur des individus qu’il a mal classifié.

# Présentation de l’entreprise

L’entité dans laquelle j’ai fait mon stage est le Laboratoire Lorrain de Recherche en Informatique et ses Applications, abrégé en [LORIA](https://www.loria.fr/fr/) et qui a été créé en 1997.  
Le laboratoire se situe à Villers-lès-Nancy et représente un des plus grands laboratoires en recherche informatique sur le territoire français et un des plus grands laboratoires toutes thématiques confondues dans la région lorraine.

Au sein du laboratoire, on compte 400 personnes dont la plupart sont des ingénieurs et des chercheurs faisant partie de l’Université de Lorraine, de l’INRIA et du CNRS et les recherches sont divisées en 5 départements :

* Département 1 : Algorithmique, calcul, image et géométrie
* Département 2 : Méthodes formelles
* Département 3 : Réseaux, systèmes et services
* Département 4 : Traitement automatique des langues et des connaissances
* Département 5 : Systèmes complexes, intelligence artificielle et robotique

Quant à ma position dans le laboratoire, je me place en tant que stagiaire dans une des équipes les plus récentes, [l’équipe K](https://k.loria.fr/), dont le chef d’équipe est Mathieu d’Aquin.  
L’équipe K est donc composée de :

* Mathieu D’Aquin
* Aurélie Bannay
* Jean Lieber
* Nicolas Jay
* Emmanuel Nauer
* Nicolas Lasolle (doctorant)

Cette équipe s’intéresse à l’intelligence artificielle symbolique et notamment au systèmes basés sur les connaissances dont les principaux problèmes sont de savoir comment :

* Concevoir des moteurs d’inférence
* Construire des bases de connaissances utilisées par les moteurs d’inférence

L’équipe se centre donc sur la connaissance qui donne lieu à 3 domaines de recherche principaux :

* La science des données (ainsi que la découverte de connaissances)
* L’ingénierie des connaissances
* Le raisonnement (hypothétique et déductif)

Dans le cadre de leurs recherches, les membres utilisent majoritairement le langage Python qui est le langage par référence pour traiter des données.

Mon intégration à l’équipe s’est faite naturellement puisque les centres d’intérêts de l’équipe et les miens se rejoignent. J’ai pu fréquenter d’autres stagiaires de l’équipe qui travaillaient sur des sujets assez différents du mien.

Par ailleurs, je dirais que le LORIA a été très bénéfique dans ce sens car j’ai aussi pu rencontrer des stagiaires issus d’autres équipes, souvent de thématiques très éloignées, qui m’ont donné une ouverture d’esprit sur leurs domaines de compétences en ayant échangé avec eux.

# Présentation du projet global

L’intuition scientifique de M. Mathieu d’Aquin a conduit à la création de ce stage. Cette intuition de pouvoir révéler des concepts au sein des réseaux de neurones artificiels à partir des activations des couches cachées est très inspirante et pourrait faire avancer de nombreux domaines de recherche.

## Prérequis

Dans le but de pouvoir concrétiser cette intuition, certains prérequis étaient nécessaires, le principal étant de savoir programmer dans le langage Python.  
Heureusement, je connaissais déjà le langage mais les librairies que j’ai du utiliser m’étaient souvent inconnues.  
De plus, malgré mon appétence pour le domaine de la recherche et les sciences dures de manière générale, lire des articles de recherche n’était pas aisé lorsque j’ai commencé ce stage.

## Enjeux de l’interprétabilité

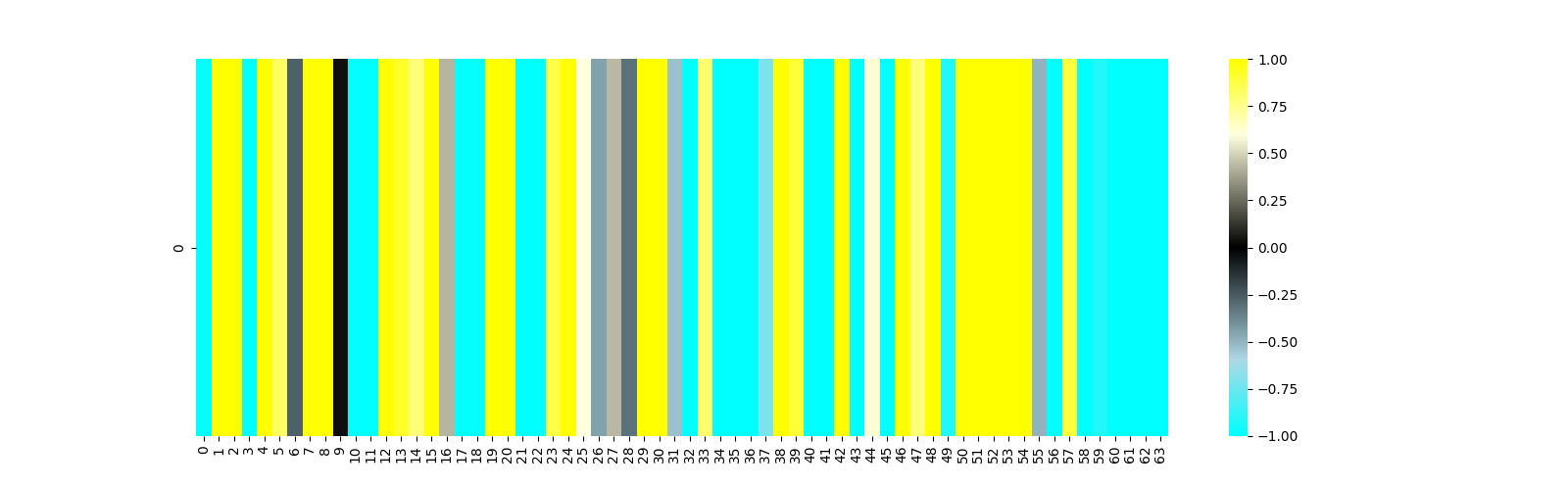
## Description du projet

Il n’est pas encore possible aujourd’hui d’expliquer le pourquoi du comment d’une décision faite par un réseau de neurone, ce qui est frustrant lorsque l’on veut par exemple savoir la raison pour laquelle la radiographie A a été classifiée comme montrant une tumeur alors que la radio B ne montre pas de signe de tumeur selon le réseau.

Ce que je viens de décrire par un exemple est un vrai problème. On aimerait savoir ce qu’il se passe à l’intérieur d’un modèle de ce genre, ne serait-ce que pour expliquer au patient que l’IA est légitime de lui donner une tumeur.

Je prends volontairement un cas extrême pour donner une image forte et concrète du problème mais on peut aussi appliquer cela dans le monde du droit.  
Imaginez-vous étant jugé pour un crime que vous n’avez pas commis. Il serait injuste qu’une IA vous punisse pour quelque chose que vous n’avez pas fait n’est-ce pas ?  
Dans ce cas, on aimerait bien inspecter les profondeurs du système intelligent afin de vérifier si l’IA n’a pas fait de jugement hâtif en prenant un raccourci dans son raisonnement. Et bien c’est ce que les chercheurs en explicabilité/interprétabilité cherchent à montrer.

Pour le cas de mon stage, on aimerait savoir comment des concepts existants ou qui regroupent des éléments sont représentés au sein des réseaux de neurones.

Un nouvel exemple serait de voir si un modèle ayant été entraîné à prédire les revenus d’un film en se basant sur sa description textuelle – par exemple « Ratatouille is a 2007 American computer-animated comedy film produced by Pixar […] later voted one of the 100 greatest motion pictures of the 21st century by a 2016 poll of international critics conducted by the BBC. » - contiendrait, parmi ses activations, une représentation du concept de pays de provenance du film, ici « United States ».

J’ai donc construit et entraîné un modèle de Deep Learning, soit un réseau de neurones artificiels qui à partir de la description d’un film prédit sa classe de revenus.

Grâce à ce réseau, il nous est possible d’explorer en détail chaque couche cachée et de comparer les activations pour un individu avec les activations moyennes pour un ensemble d’individus appartenant à une catégorie.

Cependant, comme notre but est de comparer des valeurs numériques entre elles, nous avons le besoin de standardiser toutes les valeurs d’activations. Ainsi, ils fera sens d’analyser ces valeurs selon une norme.

Il m’a été utile d’utiliser et de calculer des paramètres statistiques tels que :

* La moyenne des prédictions
* L’écart-type des prédictions
* La moyenne des valeurs vraies
* L’écart-type des valeurs vraies
* L’erreur absolue moyenne

Grâce à ces paramètres, il devient plus facile de se rendre compte quelles sont les catégories qui ont tendance à prédire mieux ou moins bien. On se rend aussi compte que pour certaines catégories, l’écart-type est très important et c’est surement ce qui rend la prédiction plus difficile donc moins fiable.

### Heatmaps

Bien que ces paramètres aident dans la compréhension de la décision faite par le réseau, on veut ici s’intéresser aux activations des couches cachées du réseau.  
Les valeurs étant comprises entre -1 et 1, on peut les représenter sous la forme de heatmaps afin de distinguer les activations fortes des activations faibles.

[parler de la diff heatmaps ?]

[Mettre photo heatmaps ?]

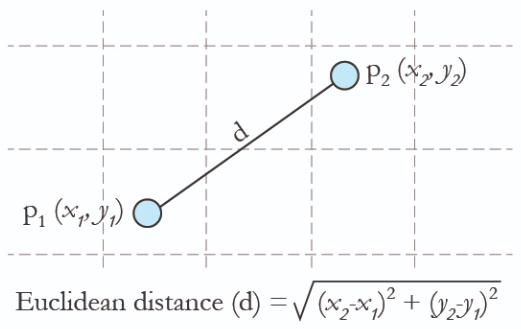
### Significativité

### Similarité

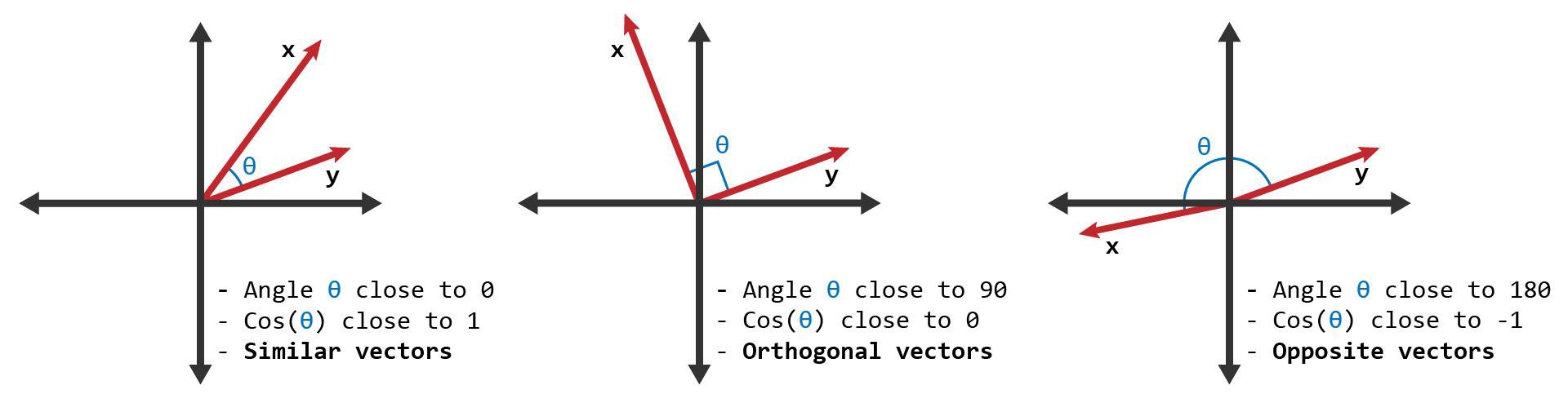
On peut dorénavant comparer à l’œil les activations entre catégories et entre individus. Ce qui fait sens ici est de voir si un individu appartenant à une catégorie – disons « France » - est réellement proche de cette catégorie en terme d’activations. On pourrait simplement regarder les deux heatmaps puisque c’est une très bonne visualisation si l’on s’intéresse au différences. Néanmoins, on aimerait une mesure plus fiable et c’est pour cela que j’introduis la similarité dans la comparaison.

Les deux similarités qui me sont venues en tête sont :

* la distance euclidienne



* la similarité cosinus



Ces deux mesures sont totalement raisonnables puisque les activations des couches que nous explorons sont une liste à 1 ou plusieurs dimensions donc totalement représentables sous la forme d’un vecteur dont on pourrait calculer la distance entre les points (distance euclidienne) ou l’angle séparant les deux projections vectorielles (similarité cosinus).  
Les valeurs de similarité nous donneront donc une mesure concrète d’à quel point deux catégories se ressemblent ou à quel point un individu s’apparente à une catégorie.

|

## Travaux existants

L’interprétabilité en IA est un domaine assez à la mode dans la communauté IA. Beaucoup d’articles sont publiés sur différentes techniques mais les articles qui nous intéressent ici sont ceux traitant de concepts de haut niveau et des activations des couches dans les systèmes d’IA numérique/stochastique.

# Présentation du travail réalisé

## Science des données

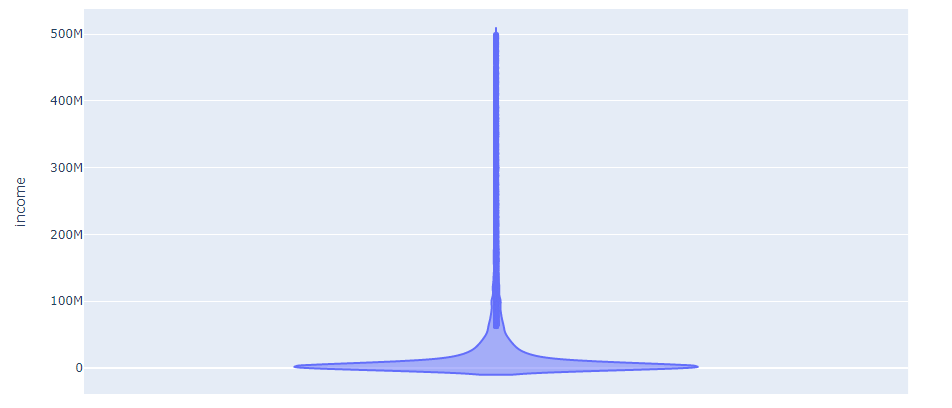
### Récolte des données

La première partie de mon stage consistait en la récolte de données. L’idée était de pouvoir récupérer des données publiques qui conviendrait au travail futur, c’est-à-dire un jeu de données sur lequel nous pourrions faire des prédictions à partir d’un attribut d’entrée ainsi qu’un autre attribut représentant des connaissances sur chaque individu.

Ainsi, le jeu de données que j’ai récolté comportait des films et pour chacun des films, sa description textuelle, ses revenus et la/les catégorie(s) à laquelle il appartient.  
Ce jeu de données nous permettait bien de pouvoir prédire à partir de la description d’un film, son revenu séparé en 3 classes :

* medium-low
* medium-high
* exceptional

Ce besoin de séparer les revenus en 3 classes distinctes relève du fait que nous ne pouvons pas prédire une valeur bien définie pour chaque film, c’est-à-dire faire une régression puisque les valeurs sont trop disparates.



Ce jeu de données a été récolté par le biais de DBpedia qui est une base de données indexée sur Wikipédia donc avec beaucoup de données accessibles et ce, en faisant une requête dans le langage SPARQL dont voici la requête :

select distinct ?film ?income ?cat ?desc where {  
?film a <http://dbpedia.org/ontology/Film> .  
?film <http://dbpedia.org/ontology/gross> ?income .  
?film <http://dbpedia.org/ontology/abstract> ?desc .  
 {  
 ?film <http://purl.org/dc/terms/subject> ?cat .   
 } UNION {  
 ?film <http://purl.org/dc/terms/subject> ?scat .   
 ?scat <http://www.w3.org/2004/02/skos/core#broader> ?cat.  
 }   
 filter (lang(?desc) = "en")  
 filter (lang(?film) = "en")  
} group by ?film ?cat ?desc LIMIT 3 OFFSET 100000

### Traitement des données

Les données étant récupérées sur DBpedia, elles ne sont pas prêtes à l’emploi. Elles ont besoin d’être traitées de manière réfléchie pour que le modèle les interprète correctement.

C’est notamment le cas des descriptions qui sont au format texte. On ne peut pas donner de texte à un réseau de neurones ou à tout autre modèle de Machine Learning puisque ces modèles sont avant tout des modèles statistiques donc qui fonctionnent avec des entrées numériques. Alors, afin de nettoyer le texte, j’ai appliqué des méthodes courantes de traitement automatique des langues comme :

* Transformer chaque lettre en minuscule
* Enlever la ponctuation, les liens, les entités numériques, ainsi que les mots très fréquents (stopwords) du genre *the, a, been, ...*.

Précision : le jeu de données étant en anglais, les mots très fréquents sont évidemment les mots fréquents en langue anglaise.

Le texte a aussi besoin d’être tokenizé, c’est-à-dire d’être représenté sous forme de liste. Chaque description a donc la forme d’une liste qui contient comme élément chaque mot qui compose la description qui a déjà commencée à être traitée.

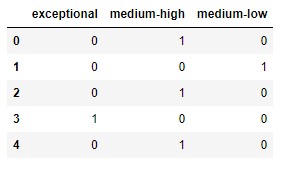
Le réseau n’accepte uniquement des entrées numériques. Or, nos descriptions sont encore au format texte. On utilise encore une fois une méthode provenant de keras afin de transcrire ce texte en séquences de nombres.

Il nous faut aussi ajouter du padding dans cette liste de séquences puisque toutes les descriptions ne font pas la même taille et le réseau n’acceptera pas plusieurs entrées de tailles différentes. Ici, le padding est représenté par des zéros qui viennent s’ajouter au début de la liste jusqu’à ce que la liste fasse la bonne taille.

Ensuite, j’ai dû remanier les revenus pour créer les classes de revenus que j’ai évoqué plus tôt. Puisque le modèle ne fonctionne qu’avec des valeurs numériques, les classes de revenus *(medium-low, medium-high et exceptional)* ont été encodé de la manière suivante :

* On crée une colonne dans le jeu de données pour chaque classe de revenus
* Pour chaque film et chaque colonne des classes de revenus on place un 0 si le film ne fait pas partie de cette classe, ou un 1 si justement le film fait partie de cette classe de revenus

On a alors un tableau présenté tel que :



Ce tableau représente la « cible » du réseau. Durant son entraînement, il va essayer en fonction de ses entrées, d’ajuster ses poids qui module les neurones au niveau des couches cachées afin d’améliorer sa précision à être fidèle à la « cible ».

Un modèle apprend sur un jeu de données et on évalue sa performance via une précision sur un jeu de validation. La différence est que le modèle ne connait pas les valeurs vraies (« cible ») des individus du jeu de validation et cela est logique puisque ce serait insensé de l’évaluer sur ce qu’il connait déjà.  
On découpe alors le jeu de données traité en un jeu de données d’entraînement qui prendra 80% du jeu de données originel et donc les 20% restants du jeu de données originel vont pour le jeu de validation.

### Construction & Entraînement du Modèle

Grâce à la librairie Keras venant de Tensorflow, j’ai pu construire un modèle comprenant les couches

[illustration modèle]

Malheureusement il n’existe pas de meilleure méthode précise à suivre pour entraîner un réseau de neurone, il existe seulement des bonnes pratiques mais chaque jeu de données peu comporter des biais différents et je devais donc tester avec des paramètres différents à chaque entraînement pour essayer d’obtenir la meilleure précision possible.

J’ai donc du jouer sur le taux d’apprentissage, la fonction de coût, la taille du lot d’individu avant que le modèle se mette à jour et le nombre d’itérations d’entraînement du modèle.

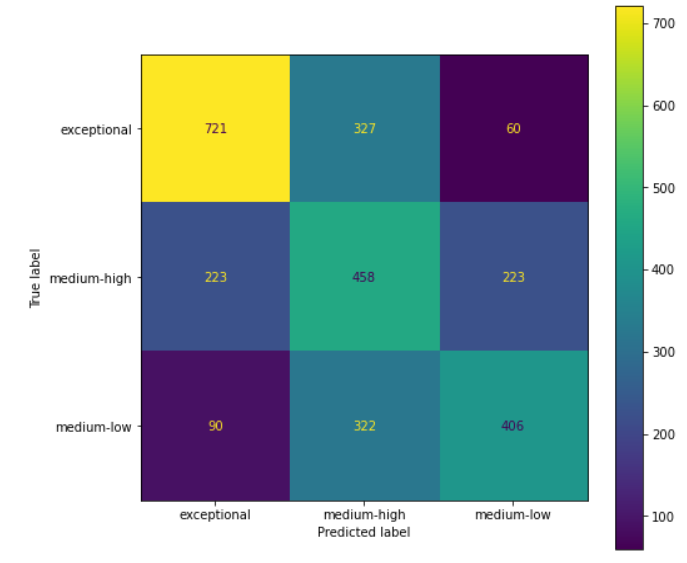
Voici les paramètres qui m’ont donné la meilleure précision :

* Taux d’apprentissage : 0.0001
* Fonction de coût : « categorical\_crossentropy »
* Taille du lot : 64
* Itérations d’entraînement : 30

A la fin de l’entraînement, on peut accéder à la précision que le modèle a eu sur le jeu de validation, ici nous avons une précision 56%.

La précision de notre modèle ne semble pas très satisfaisante a première vue. Néanmoins, il faut rappeler que la classification n’est pas binaire mais multiple. Le modèle arrive quand même plutôt bien a prédire les classes de revenus malgré qu’il existe 3 classes et que les classes *medium-low* et *medium-high* sont très ressemblantes.

Pour montrer visuellement ce que le réseau a prédit, il est judicieux de calculer et d’afficher la matrice de confusion.



Ce qui nous intéresse vraiment sur une matrice de confusion est la diagonale partant d’en haut à gauche puisque plus jaune elle sera, plus le modèle aura été précis dans sa classification.

A partir de ceci, on peut assumer que le réseau est assez performant pour avoir en ses activations, une représentation d’un concept qui serait par exemple, le genre cinématographique du film ou le pays de provenance du film.

## Exploration des activations du modele

La direction de la recherche de la présence de concepts au sein même du réseau de neurones a été orientée par l’intuition que les activations d’une couche cachée d’un réseau pouvaient être porteuses d’un ou de plusieurs concept(s).

Maintenant que nous avons un modèle entraîné à notre disposition, on va pouvoir analyser ses activations. Mais avant cela, il nous faut les récupérer.

### Récupération des activations

Il n’existe pas de fonction préconstruite pour afficher les activations d’une couche d’un modèle au sein des librairies Tensorflow et Keras. Et pour cause, il suffit de reconstruire le réseau jusqu’à la couche que l’on veut explorer et on re-exécute des prédictions avec les entrées originales pour obtenir les activations.

De cette manière il semble possible d’automatiser très facilement ce processus d’obtention des activations pour chaque couche et ce peu importe le modèle tant que l’on a accès aux données qui ont servies à son entraînement.

### Traitement des activations

On va vouloir comparer des activations entre elles par la suite. Il nous faut alors les standardiser pour avoir une des données distribuée de la même manière.

[illustration ?]

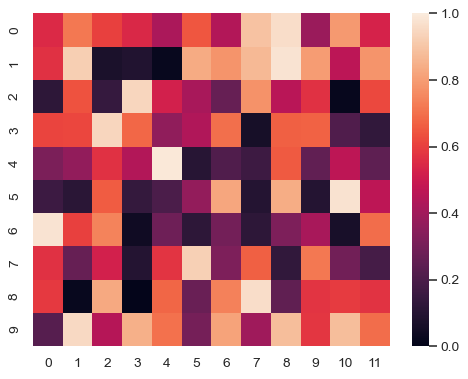
Lorsque l’on voudra visualiser les activations d’une catégorie, il nous faudra prendre la moyenne de chaque neurone des activations associées à cette catégorie et on répète ce processus pour chaque couche.

C’est au niveau de la couche « Embedding » que ce processus de moyenne des activations est différent. En effet, cette couche nous retourne un vecteur de dimension 2 pour chaque individu. J’ai donc du remettre chaque vecteur à plat afin d’ensuite appliquer la moyenne […]

Pour ce qui est des individus, il nous suffira de récupérer les activations puis de les afficher.

### Heatmaps

Une heatmap est une technique de visualisation qui permet de visualiser plus efficacement les différences au sein d’une collection via l’utilisation de couleurs représentant l’intensité d’une valeur pour chaque valeur dans la heatmap.



Voici une heatmap basique qui représente des données quelconques.

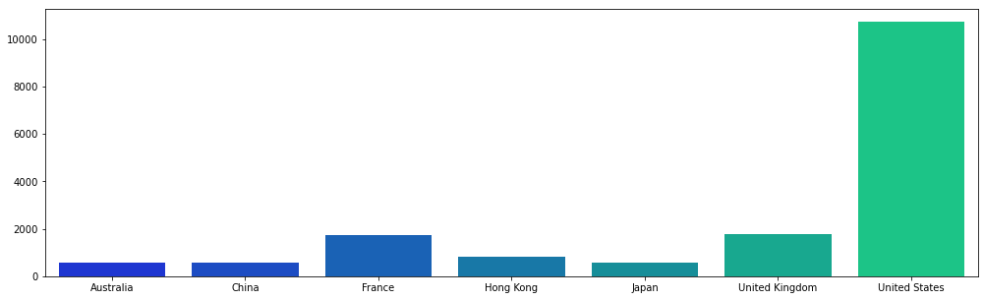
On voit grâce à la légende présente sur le côté droit que plus une case est blanche, plus la valeur y étant représentée est proche de 1.0.

Ce qui fonctionne plutôt bien dans notre cas puisque l’on espère visualiser des différences entre les neurones d’une catégorie/individu mais aussi de voir à l’œil nu les similarités entre un individu et la catégorie qui lui est associée par exemple.

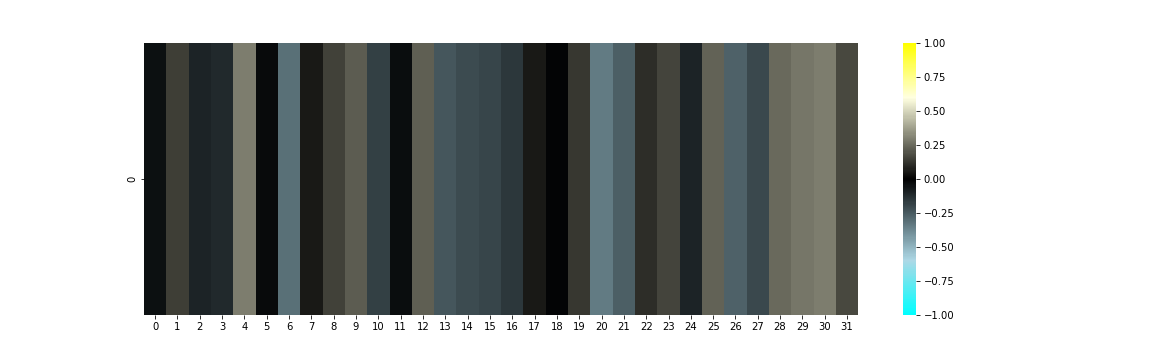
Les heatmaps sont générées par le biais des librairies matplotlib et seaborn qui fonctionnent ensemble et offrent une multitude d’outils de visualisation très pratiques pour la science des données.

Etant donnée la multitude de genres cinématographiques différents, j’ai choisi de ne prendre que les plus populaires et donc d’exclure les genres étant sous un certain seuil, seuil qui correspond au nombre de films inclus dans le concept auquel on s’intéresse, ici le genre cinématographique.

Rapidement, en essayant plusieurs seuils, 500 semble être une bonne valeur de référence puisque l’on se retrouve avec assez de concepts différents pour faire des comparaisons intéressantes.

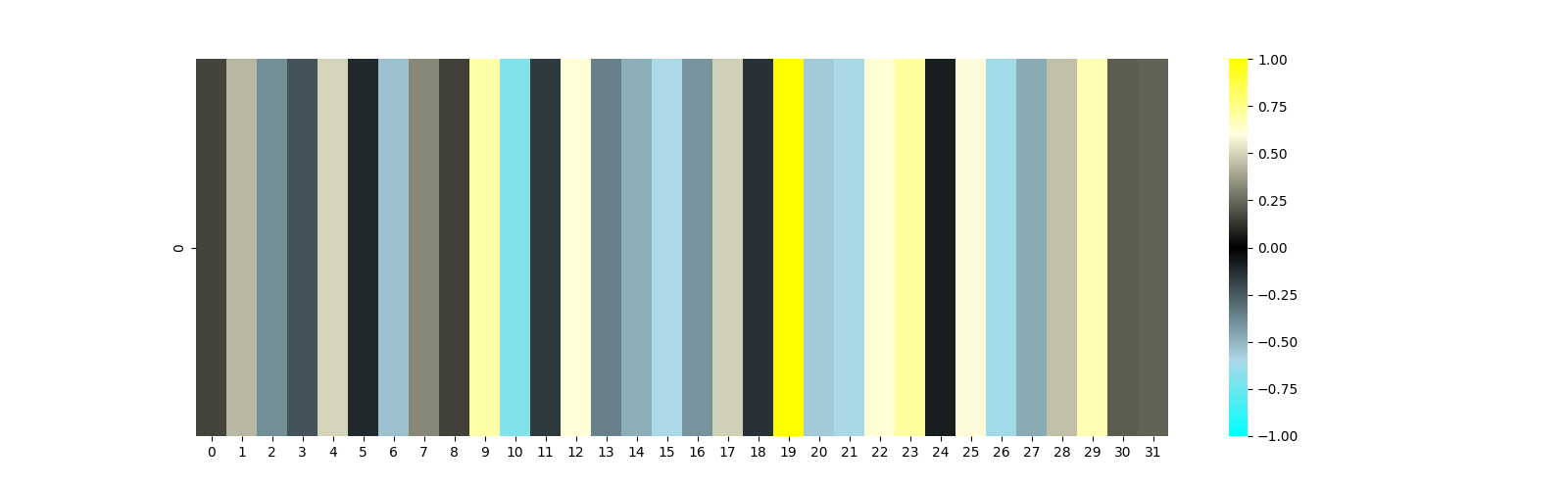


On peut donc visualiser la heatmap des activations pour un concept et pour une couche spécifique du réseau. Prenons par exemple le cas de « France » comme pays de provenance en tant que concept et la couche « LSTM » de notre réseau :



On voit donc que les activations sur la heatmap sont assez ternes. Ce qui veut dire qu’en moyenne, les activations des films dont le pays de provenance est la France tendent à se situer dans l’intervalle [-0.5 ; 0.5] environ. Ce qui est logique puisque dans les films français, certaines activations doivent tendre vers 1 et d’autres vers -1, et on se retrouve donc avec des activations autour du milieu de notre palette de couleurs comme on peut le voir sur la légende à droite.

Ainsi, si on regarde les activations pour un seul individu appartenant au concept « provenant de France », on devrait voir une heatmap avec des couleurs bien moins ternes :



Ici, on voit la heatmap représentant les activations au niveau de la couche « LSTM » pour le film « Without Leaving an Address ».  
On voit clairement que les activations sont plus prononcées et on remarque surtout que certains neurones comme par exemples les numéros 6, 20, 21, 26, 27 et d’autres sont activés de la même manière. Ceci pourrait être un indicateur de la présence d’un concept au sein même des activations du réseau de neurones.

|

## logiciel de visualisation

[intro

### Apprentissage de Qt

[bla bla]

### Fonctionnalités

[bla bla]

### Boost des activations

[bla bla]

# Conclusion

## competences developpees

# Glossaire

**Intelligence artificielle symbolique** : sorte d’IA se basant sur des règles / connaissances en imitant le raisonnement humain dans le but de produire des décisions.

**Système à base de connaissance** : programme utilisant une base de connaissances afin de résoudre des problèmes. Ce genre de système est en général composé d’au moins un moteur d’inférence et au moins une base de connaissances.

# Annexes

[DESCRIPTION DE CHAQUE PAGE]

# Remerciements

Je tiens à remercier chaleureusement M. Mathieu d’Aquin qui m’a permis de réaliser ma première expérience de recherche dans une entité que j’affectionne tout particulièrement.

# Quatrième de couverture (changer titre)

* Mathieu D’Aquin : [mathieu.daquin@loria.fr](mailto:mathieu.daquin@loria.fr)