



# Attributions Axiomatiques pour les Réseaux Profonds

## Introduction

Ce travail a été effectué dans le cadre du cours de **DNN (Deep neural Network)** à l'[Epita](#). Il a été produit par les étudiants :

- **Adrien Merat** <[adrien.merat@epita.fr](mailto:adrien.merat@epita.fr)>
- **Corentin Duchêne** <[corentin.duchene@epita.fr](mailto:corentin.duchene@epita.fr)>
- **Hao Ye** <[hao.ye@epita.fr](mailto:hao.ye@epita.fr)>
- **Henri Jamet** <[henri.jamet@epita.fr](mailto:henri.jamet@epita.fr)>
- **Theo Perinet** <[theo.perinet@epita.fr](mailto:theo.perinet@epita.fr)>



### Ressources

- Le repository contenant notre travail peut -être trouvé ici : <https://gitfront.io/r/user-5856462/PMez3XNpXAJC/Epita-S9-DNN/>
- Si vous lisez ce document depuis le **PDF** exporté dans le repository, nous vous invitons à le consulter en ligne pour plus de praticité : <https://henri-jamet.notion.site/Attributions-Axiomatiques-pour-les-R-seaux-Profonds-146fdf9f541b48738a7c60a2ad669ed8>

L'objectif de ce projet est la *compréhension* et la *réimplémentation* d'un article de recherche portant sur les **Réseaux de Neurones Profonds** dans le cadre du **traitement de l'image**.

L'article que nous avons traité s'intitule [Axiomatic Attribution for Deep Networks](#). Il a été publié en Mars 2017 par [Mukund Sundararajan](#), [Ankur Taly](#) & [Qiqi Yan](#) et traite de l'**attribution axiomatique** pour les réseaux de neurones, une technique d'explicabilité de l'IA visant à déterminer l'importance de chaque feature pour la prédiction d'un réseau de neurones.

## Plan de ce document

Ce document s'organise de la manière suivante :

[Introduction](#)

[Plan de ce document](#)

[Méthodes des Gradients Intégrés](#)

[Axiomes](#)

[Sensitivité](#)

[Invariance de l'Implémentation](#)

[Sensitivité totale](#)

[Linéarité](#)

[Préservation de la Symétrie](#)

[Construction de la méthode](#)

[Choix de la Baseline](#)

[Baseline floue](#)

[Baseline Opposée](#)

[Baseline Aléatoire](#)

[Moyenne sur de multiples Baselines](#)

[Articles liés](#)

[Reinforced Integrated Gradients](#)

[Attention Is All You Need](#)

[Conclusion](#)

[Références](#)

[Papiers](#)

[Autres ressources](#)

1. Dans un premier temps, nous expliquerons l'idée présentée dans l'article original.
2. Puis, nous discuterons plus avant des choix de **Baselines** possibles et des idées derrière ces derniers.
3. Finalement, nous ferons le lien avec deux autres articles scientifiques connexes que nous pensons pertinent de souligner.

## Méthodes des Gradients Intégrés

Inspirée par un article publié en 2016 : [Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization](#), la méthode des **Gradients Intégrés** présentée dans l'article que nous avons étudié a vocation à *expliquer* les prédictions faites par des modèles *Boîte Noire* comme les réseaux de neurones profonds. Pour ce faire, elle décrit une fonction  $A_F : \mathcal{D}_F^2 \longrightarrow (a_1, \dots, a_n)$  qui associe

- Pour un modèle donné dont le comportement est décrit par la fonction  $F$
- Avec une **Baseline** choisie  $x' \in \mathcal{D}_F$

à toute entrée  $x \in \mathcal{D}_F$  les attributions  $(a_1, \dots, a_n)$ , ie l'**importance de chacune des features de  $x$  dans la prédiction finale  $F(x)$** .


## Axiomes

Pour éviter la difficulté fondamentale de la définition d'une **méthode d'attribution**, à savoir l'impossibilité d'identifier la cause première d'une attribution peu révélatrice du comportement du réseau entre les faiblesses du réseau lui-même ou la non-pertinence de la méthode d'attribution, les auteurs ont identifié un certain nombre d'Axiomes de plus ou moins grande importance à partir desquels ils ont imaginé la méthode des **Gradients Intégrés** pour pallier les faiblesses des méthodes antérieures.


- Deux Axiomes Fondamentaux :
  - Sensitivité
  - Invariance de l'Implémentation
- Deux Axiomes Désirables :
  - Sensitivité totale
  - Linéarité
- Un Axiome Désirable & Canonique :
  - Préservation de la Symétrie

Définissons ces Axiomes :

Sensitivité

 Une méthode d'attribution respecte la Sensitivité ssi  
→ toute feature dont le réseau ne dépend pas mathématiquement a une attribution **nulle**


Invariance de l'Implémentation

 Une méthode d'attribution respecte l'Invariance de l'Implémentation ssi  
→ deux réseaux *fonctionnellement équivalents* (ie effectuant toujours des prédictions identiques pour toute entrée  $x$ ) ont toujours les mêmes attributions, *même lorsque leurs architectures sont différentes*.

Sensitivité totale

 Une méthode d'attribution respecte la Sensitivité totale ssi  
→ toute feature dont le réseau ne dépend pas mathématiquement a une attribution **nulle**.


Linéarité

 Une méthode d'attribution respecte la Linéarité ssi  
→ toute composition linéaire de deux réseaux  $F_1$  &  $F_2$

$$a \times F_1 + b \times F_2$$

voit ses attributions également composées linéairement en fonction de  $a$  &  $b$ .

Préservation de la Symétrie

 Une méthode d'attribution est dite Préservant la Symétrie ssi  
→ Une méthode d'attribution **préserve la symétrie** ssi pour deux features symétriques (ie telles que  $F(x, y) = F(y, x)$ , les attributions sont identiques.

On montre que les seules méthodes d'attributions respectant à la fois les deux **Axiomes Fondamentaux** & les deux **Axiomes Désirables** appartiennent à une catégorie de méthodes appelée **Path Methods**, qui se caractérisent par la génération d'images interpolées le long d'un "chemin" dans l'espace vectoriel formé par l'entrée du réseau  $F$ .

Au sein de ces méthodes, on en note une **canonique** : La méthode des **Gradients Intégrés** qui se caractérise par un **Axiome Désirable** supplémentaire dont est la seule à disposer : la **Préservation de la Symétrie**.

Construction de la méthode

Avec :

- $x_i$  la  $i_{ème}$  feature de l'**Entrée**,
- $x'_i$  la  $i_{ème}$  feature de la **Baseline**
- $F$  la fonction décrivant notre modèle,
- $x, x'$  respectivement **Entrée & Baseline**,

La méthode est décrite par la formule exacte suivante :

$$\text{Gradients Intégrés}_{i(x)} = (x_i - x'_i) \times \int_{\alpha=0}^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

Et par la formule approximée par une somme de Riemann suivante :

$$\text{Gradients Intégrés}_{i(x)} \simeq (x_i - x'_i) \times \sum_{k=1}^n \frac{\partial F(x' + \frac{k}{n} \times (x - x'))}{\partial x_i} \times \frac{1}{n}$$

On peut décomposer cette dernière formule en 4 étapes qui s'implémente relativement aisément :

- Générer  $n$  images interpolées entre la **Baseline** et l'**Input** :

$$x' + \frac{k}{n} \times (x - x') \quad \text{pour } k \in [[0, n]]$$

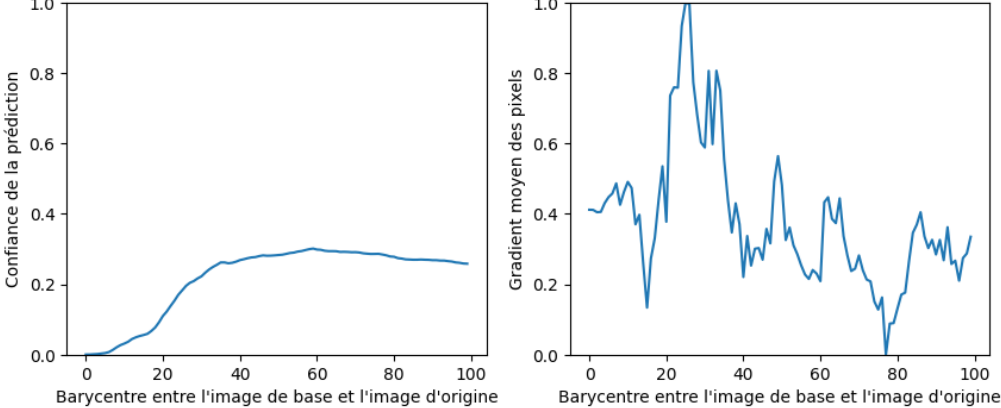


- Calculer les *Gradients* entre les prédictions de sortie du modèle  $F$  par rapport aux caractéristiques d'entrée :

$$\frac{\partial F(\text{Image Interpolée})}{\partial x_i}$$

Pour l'image Pillows, en fonction du barycentre entre l'image de base et l'image d'origine

Probabilité de prédiction de la classe la plus probable. Moyenne normalisée des gradients des pixels.

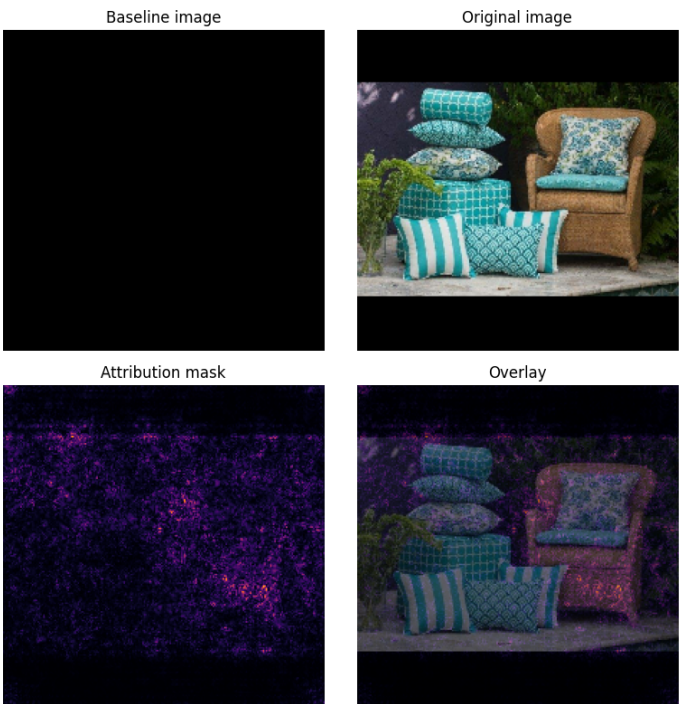


- Cumuler les *Gradients* ainsi calculés (On approxime l'intégrale au moyen de la **méthode des Trapèzes** :

$$\sum_{k=1}^n \text{Gradients} \times \frac{1}{n}$$

- Mettre les *Gradients Intégrés* à l'échelle par rapport à l'image d'origine :

$$(x_i - x_{i'}) \times \text{Gradients Intégrés}$$



### Choix de la Baseline

On remarque que le choix de la **Baseline** a été assez peu débattu dans ce rapport. Elle constitue pourtant un choix complexe affectant considérablement la qualité des résultats de notre méthode d'attribution. L'objectif d'une **Baseline** étant de définir une entrée  $x'$  suffisamment

différente de  $x$  pour servir de “référentiel” pour notre calcul.

Une approche naïve consiste à effectuer ce que nous avons fait dans notre implémentation détaillée avec le notebook [tensorflow.ipynb](#), c’est à dire choisir une image uniforme (dans notre cas, uniformément noire).

Cependant, nombre de nos [Références](#) font état de Baselines & d’idées d’optimisation astucieuses que nous allons tenter de détailler ici :

## Baseline floue

Comme ce que nous cherchons finalement à remarquer lorsque nous lisons une carte d’attribution sont surtout les *contours* des éléments sur lesquels le réseau porte son attention, un article intitulé [Interpretable Explanations of Black Boxes by Meaningful Perturbation](#) paru en 2017 propose l’utilisation d’une variation de l’entrée floutée comme Baseline. De cette manière, nous forçons notre méthode d’attribution à mettre en exergue l’absence de certains éléments, ce qui permet de faire davantage sortir les contours.

## Baseline Opposée

Étant donné que la **Baseline** sert avant tout de référentiel pour noter la différence entre une image neutre & une image dont nous cherchons à obtenir les attributions, on pourrait imaginer une Baseline calculée comme le parfait “opposé” de notre image d’entrée. De manière plus rigoureuse, on peut définir une **Baseline**  $x'$  définie comme l’image la plus éloignée de  $x$  en norme  $L1$  dont la valeur des pixels reste possible. Cependant, il s’avère que cette méthode donne de médiocres résultats car la **Baseline** reste trop semblable à l’image d’origine pour que la différence ( $x - x'$ ) soit porteuse de sens.

## Baseline Aléatoire

Finalement, une solution alternative intéressante peut simplement consister en la génération d’une **Baseline** comme une image constituée de pixels de couleur aléatoire. En effet, cette stratégie permet d’éviter l’important biais inhérent à une **Baseline uniforme** dont la couleur risque de faire négliger à la méthode d’attribution l’importance des pixels de la même couleur qu’elle (*la différence ( $x - x'$ ) devenant quasiment nulle*). L’idée consistant à utiliser une **Baseline Aléatoire** permet d’espérer que si un pixel de la **Baseline** s’avère être par malheur de la même couleur que le pixel correspondant de l’image entrée, son voisin échappe quant à lui à ce coup du sort.

## Moyenne sur de multiples Baselines

Finalement, suivant l’idée présentée dans un article paru en 2019 : [Learning Explainable Models Using Attribution Priors](#), nous avons décidé d’utiliser  $n$  **Baselines** aléatoires dont nous moyennons finalement les attributions dans notre notebook [pytorch.ipynb](#). Il eut été intéressant de tester une moyenne sur des **Baselines** autres que **Aléatoires**, mais nous n’avons malheureusement pas eu le temps de mener à bien ces expérimentations complémentaires.

## Articles liés

Nous finirons ce rapport en évoquant deux autres articles scientifiques qu’il nous a semblé pertinent de confronter avec l’article [Axiomatic Attribution for Deep Networks](#) étudié :

## Reinforced Integrated Gradients

Présenté très récemment à la conférence [EGC2023](#) à laquelle certains des membres de notre groupe ont eu la chance de pouvoir assister, cet article cherche à améliorer la pertinence des **cartes d’attributions** générés au moyen de la méthode des **Gradients Intégrés** en entrainer  $n$  modèles légèrement différents (*une forme de Bagging*), pour lesquels sont ensuite calculés les attributions d’une même image. Les attributions obtenus sont finalement moyennées pour obtenir une **carte d’attribution** nettement plus pertinente dans la mesure où elle permet de mettre en lumière les feature *généralement utilisés par un modèle du type choisi* dans sa prédiction, ce qui permet de simplifier nettement l’interprétation du résultat.

On note cependant que cette technique a surtout de l’intérêt si la **méthode d’attribution** n’a pas vocation à étudier le modèle mais d’avantage le problème auquel il est confronté.

Hélas, l’article ayant paru extrêmement récemment, nous ne sommes pas parvenu à retrouver le papier original, ce qui nous empêche de le citer dans nos [Références](#). Cependant, l’idée nous a semblé suffisamment intéressante pour qu’il soit pertinent de la mentionner ici.

## Attention Is All You Need

Paru peu après l’article [Axiomatic Attribution for Deep Networks](#), la même année, cet article fondateur qui a introduit l’architecture des [Transformers](#) basée sur le [mécanisme d’Attention](#) paraît extrêmement corrélé à l’article que nous étudions. En effet, la notion d’Attention a l’avantage d’être facilement explicable dans la mesure où elle est par définition construite sur une forme de mécanisme d’attribution. Plusieurs résultats présentés dans l’article des **Gradients Intégrés** font énormément penser à la projection de la matrice d’Attention d’un modèle en disposant et on peut supposer que bien que l’idée d’**Attention** soit antérieure à 2017 ([Neural Machine Translation by Jointly Learning to Align and Translate](#), 2014), l’idée de la méthode d’Attribution des **Gradients Intégrés** a du contribuer à la naissance des mécanismes d’**Attention**.

## Conclusion

En conclusion, nous avons d’abord résumé l’idée des **Gradients Intégrés** introduite dans l’article [Axiomatic Attribution for Deep Networks](#). Nous avons ensuite constaté que le choix d’une **Baseline** est complexe et peut avoir une grande influence sur la qualité des résultats obtenus par nos méthodes d’attribution. Un choix de **Baseline** uniforme ou aléatoire peut faire la différence entre des résultats clairs et pertinents et des résultats trop biaisés pour être interprétables. Les articles scientifiques dont nous avons parlé nous ont également permis d’en apprendre davantage sur les implications possibles de la méthode des **Gradients Intégrés**, notamment en ce qui concerne les mécanismes d’**Attention**. Nous avons pu constater que les **Gradients Intégrés** peuvent être une méthode très puissante pour obtenir des attributions qui peuvent nous aider à comprendre le fonctionnement des modèles profonds.

Comme expliqué sur notre [Repository](#), nous avons cherché à reproduire au mieux les résultats présentés dans l’article original au moyen de deux notebooks :

- [tensorflow.ipynb](#) : Qui suit un tutoriel détaillé en ligne tout en revisitant complètement le code proposé & en testant de nouvelles idées,
- [pytorch.ipynb](#) : Qui présente une implémentation libre & optimisée des **Gradients Intégrés**, en utilisant notamment l’idée du moyennage sur de nombreuses Baselines aléatoires.


Références


| Par ordre de publication,

Papiers

**Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization**


We propose a technique for producing "visual explanations" for decisions from a large class of CNN-based models, making them more transparent. Our approach - Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept, flowing into the final convolutional layer to produce a coarse localization map highlighting important regions in


 <https://arxiv.org/abs/1610.02391>



**Axiomatic Attribution for Deep Networks**


We study the problem of attributing the prediction of a deep network to its input features, a problem previously studied by several other works. We identify two fundamental axioms---Sensitivity and Implementation Invariance that attribution methods ought to satisfy. We show that they are not satisfied by most known attribution methods, which we consider to be a


 <https://arxiv.org/abs/1703.01365>



**Interpretable Explanations of Black Boxes by Meaningful Perturbation**


As machine learning algorithms are increasingly applied to high impact yet high risk tasks, such as medical diagnosis or autonomous driving, it is critical that researchers can explain how such algorithms arrived at their predictions. In recent years, a number of image saliency methods have been developed to summarize where highly complex neural networks "look" in an


 <https://arxiv.org/abs/1704.03296>



**Neural Machine Translation by Jointly Learning to Align and Translate**


Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance.


 <https://arxiv.org/abs/1409.0473>



**Attention Is All You Need**


The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder-decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms,


 <https://arxiv.org/abs/1706.03762>




**Learning Explainable Models Using Attribution Priors**

Code: <https://www.dropbox.com/sh/xvt3vqv8xb5nwh/AACgt-0OxieflmjVXX5UJSuaa?dl=0> Keywords: Deep Learning, Interpretability, Attributions, Explanations, Biology, Health, Computational Biology TL;DR: A method for encouraging axiomatic feature attributions of a deep model to match human intuition. Abstract: Two important topics in deep learning both involve

 <https://openreview.net/forum?id=rygPm64tDH>





 **Reinforced Integrated Gradients (non encore disponible en ligne, encadré par Nicolas Boutry)**

Autres ressources

**Gradients intégrés | TensorFlow Core**


Ce didacticiel montre comment implémenter les gradients intégrés (IG) , une technique d' IA explicable introduite dans l'article Attribution axiomatique pour les réseaux profonds . IG vise à expliquer la relation entre les prédictions d'un modèle en fonction de ses caractéristiques.

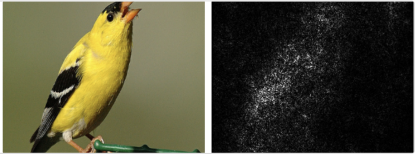
 [https://www.tensorflow.org/tutorials/interpretability/integrated\\_gradients?hl=fr](https://www.tensorflow.org/tutorials/interpretability/integrated_gradients?hl=fr)

  
**TensorFlow**


**Visualizing the Impact of Feature Attribution Baselines**


Path attribution methods are a gradient-based way of explaining deep models. These methods require choosing a hyperparameter known as the baseline input. What does this hyperparameter mean, and how important is it? In this article, we investigate these questions using image classification networks as a case study.

 <https://distill.pub/2020/attribution-baselines/>




► Grab Your FREE MLOps Monitoring Whitepaper here: <https://hubs.ly/H0F8w230> ► Request Your FREE Fiddler Demo here: <https://hubs.ly/H0FZrK20> ► Grab Your FREE 2020 Market Research here: <https://hubs.ly/H0F8zcP0> === Follow us! ► LinkedIn: <https://www.linkedin.com/company/fiddler-labs/> ► Twitter: <https://twitter.com/fiddlerlabs> === The second in AI

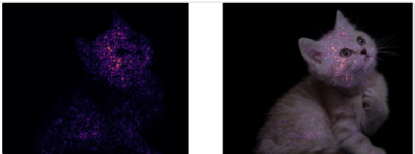
 <https://youtu.be/9AaDc35YiI>



**Understanding Deep Learning Models with Integrated Gradients**


This post will help you to understand the two basic axioms of Integrated Gradients and how to implement Integrated Gradient using TensorFlow using a transfer learned model. What is Integrated Gradient?

 <https://medium.com/towards-data-science/understanding-deep-learning-models-with-integrated-gradients-24ddce643dbf>



**Explainable AI: Integrated Gradients for Deep Neural Network Predictions**

Integrated Gradients make it possible to examine the inputs of a deep learning model on their importance for the output. A major criticism of deep neural networks is their lack of interpretability, as we know it from linear regression, for example.

 <https://medium.com/codex/explainable-ai-integrated-gradients-for-deep-neural-network-predictions-eb4f96248afb>

