EPITA - SCIA/DNN

Nikoloz CHADUNELI/Alexis JULIEN/Paul MESSEANT/Paul RENOUX/Matthieu SCHLIENGER

DNN

Contents

1 Introduction

Dans ce rapport, nous présentons l'implémentation de la méthode des Integrated Gradients pour l'attribution de l'importance des features dans un modèle de Deep Learning. Les Integrated Gradients sont une technique d'attribution d'importance qui consiste à mesurer l'impact de chaque feature d'une entrée sur la sortie du modèle. Cette méthode a été proposée pour la première fois par Sundararajan, Mukherjee, et Taly dans leur papier de 2017 intitulé "Axiomatic Attribution for Deep Networks" que nous avons étudié.

Les Integrated Gradients sont particulièrement utiles pour comprendre les décisions prises par les modèles de Deep Learning qui sont souvent considérés comme étant des boîtes noires en raison de leur complexité. Cette méthode permet d'affecter une contribution quantifiée à chaque feature d'une entrée en se basant sur la perturbation de chaque feature et la différence de sortie produite par cette perturbation.

Nous avons donc implémenté cette méthode en utilisant PyTorch. Pour ensuite l'appliquer à trois modèles différents: Resnet18, notre propre modèle de classification et un modèle de détection. Cela nous a permis de comprendre la décision de nos modèles.En effet, les résultats obtenus montrent que les Integrated Gradients sont capables de mettre en évidence les features les plus importantes pour la décision prise par le modèle.

2 Définition

Integrated Gradients Les integrated gradients sont une méthode d'attribution d'importance pour les réseaux de neurones. Le but de cette méthode est de déterminer l'importance de chaque entrée dans la prédiction finale d'un modèle. Les integrated gradients sont particulièrement utiles pour comprendre comment les réseaux de neurones fonctionnent et comment ils prennent leurs décisions.
La définition mathématique des integrated gradients est la suivante :

$$IntegratedGrads_i(x) ::= \left(x_i - x_i'\right) \times \int_{\alpha=0}^{1} \frac{\partial F\left(x' + \alpha \times (x - x')\right)}{\partial x_i} d\alpha$$

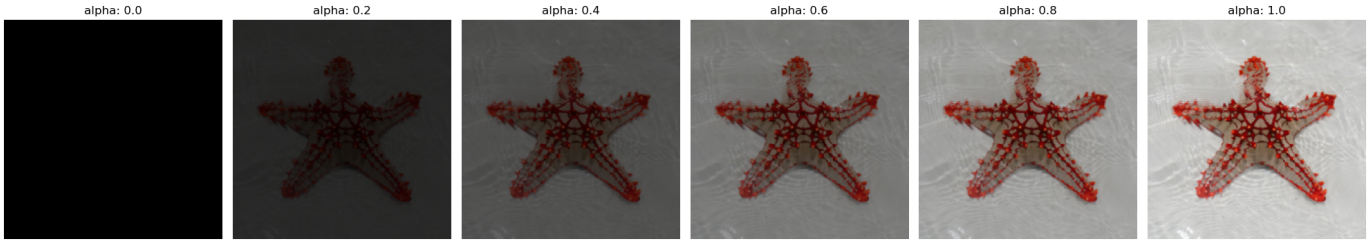
Où $F(x)$ représente la fonction du modèle, x_i représente la i-ème entrée du modèle, et x' représente la valeur de référence ou baseline. Le terme $(x_i - x_i')$ mesure la différence entre la valeur réelle de x_i et sa valeur de référence. L'intégrale termine le long de la droite reliant la valeur de référence à la valeur réelle de x_i et mesure le gradient total de la fonction de modèle le long de cette droite pour x_i.
Dans la pratique, cette intégrale est souvent approximée en utilisant une somme finie pour calculer la moyenne des gradients sur une série d'interpolations entre la valeur de référence et la valeur réelle de x_i. Pour ce faire, on définit un nombre m de points d'échantillonnage, et pour chaque point k, on calcule la dérivée partielle du modèle F par rapport à la feature x_i sur le chemin interpolé $x' + \frac{k}{m} \times (x - x')$. Ensuite, on calcule la somme de ces dérivées partielles et on la multiplie par la différence entre la feature actuelle x_i et la feature de référence x_i'. La valeur finale est alors la moyenne de ces sommes. Ce qui donne:

$$\text{IntegratedGrads}_i^{approx}(x) ::= \left(x_i - x_i'\right) \times \sum_{k=1}^{m} \frac{\partial F\left(x' + \frac{k}{m} \times (x - x')\right)\big)}{\partial x_i} \times \frac{1}{m}$$

3 Implémentation

Pour l'implémentation de ce papier nous avons choisi d'appliquer cette méthode à un modèle de deep learning de classification, mais en premier lieu nous avons implémenté l'algorithme permettant de calculer les integrated gradients. Les étapes clés sont les suivantes :

- Choix d'un point de base (baseline) : Le point de base est une image qui est considérée comme une image neutre et qui sera utilisée comme référence pour calculer les contributions de chaque pixel. Il est généralement choisi en utilisant une image noire ou une image remplie de valeurs de zéro. Dans notre cas, nous avons choisi de prendre une image noire.
- Interpolation de l'image : pour calculer les contributions, nous devons trouver une série d'images intermédiaires entre notre baseline et l'image originale. Cela se fait en linéarisant les deux images, puis en ajoutant les différences entre les pixels pour obtenir une image intermédiaire. (Voir ci-dessous)



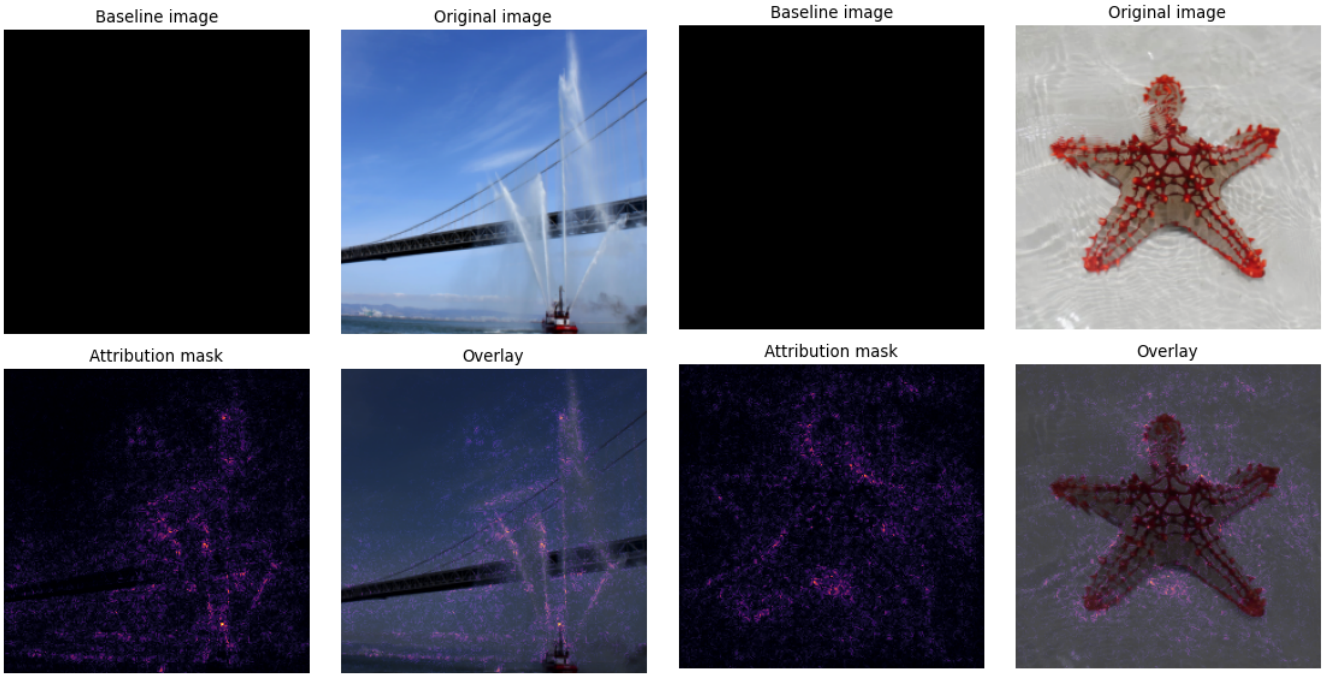
- Calcul des gradients : Pour chaque image interpolée, dans notre cas 51 images donc, nous avons calculé les gradients par rapport à chaque pixel, de manière à obtenir le gradient pour chaque pixel de chaque image le long du chemin d'interpolation. On peut alors savoir, pour chaque pixel, l'impact qu'il a sur la prédiction.
- Intégration des gradients : Les gradients calculés pour chaque image intermédiaire sont ensuite intégrés pour donner les Integrated Gradients. Cela peut être accompli en utilisant une méthode d'intégration telle que la méthode de trapezoidal. Cette méthode consiste à prendre la moyenne des gradients de deux points consécutifs.
- Attribution : Enfin, les Integrated Gradients sont utilisés pour attribuer les contributions à chaque pixel dans l'image originale. On additionne les valeurs absolues des integrated gradients sur les canaux de couleur pour produire un masque d'attribution. Les valeurs de contribution peuvent être utilisées pour visualiser les régions les plus importantes dans l'image pour une prédiction donnée.

Nous avons choisi d'utiliser le modèle ResNet18 pour classifier des images et donc voire comment celui-ci se servait des images en input pour établir sa prédiction. Le modèle ResNet est un modèle de reconnaissance d'images populaires qui utilise la conception de bloc de récupération pour améliorer la profondeur et la performance du réseau de neurones. C'est un modèle de référence dans la reconnaissance d'images et est souvent utilisé pour l'illustration des algorithmes d'explication d'images tels que les Intégrated Gradients.
Nous avons aussi essayé de créer notre propre modèle pour pouvoir comparer les résultats avec ResNet. Cela nous aurait permis de nous rendre compte que d'un modèle à l'autre les pixels ne sont pas forcément utilisés de la même façon. Malheureusement, par manque de temps nous n'avons pas pu finir cette implémentation car nous avons rencontré des problèmes. Notre calcul de gradient renvoyait un gradient nul mais nous n'avons pas réussi à trouver la cause de ce problème.

De même, nous avons essayé d'implémenter un modèle de détection et nous avons eu le même problème ainsi qu'un problème de gestion de mémoire : nos machines plantaient au beau milieu du calcul des integrated gradients.

4 Résultats Obtenus

Grâce à la méthode des integrated gradients nous pouvons observer l'importance de chaque pixel d'une image lors de sa classification. Cela nous permet de mieux comprendre comment fonctionne un réseau de neurones convolutionels.
Nous avons pu essayer cela sur quelques images et voici les résultats que nous avons obtenus.



Sur ces images, on peut voir en violet sur les attribution masks les pixels qui ont le plus servi pour réaliser la prédiction. On peut clairement distinguer les formes des objets importants présents sur les images.