Quelques méthodes XAI, cours XAI, semestre 2 MALIA, 2024, 1–11
doi: objet numérique sans identifiant (préparation pédagogique)
Date de soumission : février 2025
Enseignement de : Pr Stephane Chretien

Présentation Et Tests De méthodes D'Explainability

Mehdi Mansour[1,*]

[1]Departement Informatique, ICOM, 5 avenue Pierre Mendès France, 69500 Bron, France

[*]Corresponding author. mehdi.mansour@univ-lyon2.fr

Abstract

Dans ce travail nous nous approprions et testons une implémentation pratique de six méthodes d'Explainability: SHAP (SHapley Additive exPlanations), DiCE (Diverse Counterfactual Explanations), Grad-CAM (Gradient-weighted Class Activation Mapping), La librairie Captum avec des calculs d'attributions basés sur le gradient pour l'interprétabilité des modèles de langage, et deux variantes de Layer-wise Relevance Propagation (LRP) - pour les réseaux de neurones entièrement connectés et pour les réseaux de neurones sur graphes (GNN-LRP).

Key words: XAI, SHAP,DiCE, Grad-CAM, LRP, Captum library

Introduction

Alors que les systèmes d'intelligence artificielle deviennent de plus en plus complexes et omniprésents dans les processus décisionnels critiques, la nécessité d'une IA explicable (XAI) est devenue primordiale. La capacité à comprendre et interpréter les décisions des modèles n'est plus simplement une quête académique mais une nécessité pratique, particulièrement dans des domaines comme la santé, la finance et les systèmes juridiques où les décisions doivent être transparentes et responsables. Le défi de l'explicabilité en IA découle de la tension fondamentale entre la complexité des modèles et leur interprétabilité. Bien que les modèles plus complexes atteignent souvent des performances supérieures, ils tendent à être moins transparents dans leurs processus décisionnels. Cela a conduit au développement de diverses méthodes d'explication post-hoc, chacune conçue pour éclairer différents aspects du comportement et des processus décisionnels des modèles. Dans ce travail, nous explorons six méthodes XAI. Chaque méthode représente une approche différente de l'interprétation des modèles. Notre travail se concentre, pour chaque méthode, sur l'identification de son fondement théorique et ensuite sur l'application simple et pédagogique d'un cas d'usage.

Valeurs de Shapley (SHAP)

Fondements théoriques

Contexte

Les valeurs de Shapley, initialement introduites par Lloyd Shapley en 1953 dans le contexte de la théorie des jeux coopératifs, ont trouvé une nouvelle application majeure en apprentissage automatique grâce aux travaux de Lundberg et Lee en 2017 avec l'introduction de SHAP (SHapley Additive exPlanations) [3].

Intuition physique

L'idée fondamentale de SHAP est d'expliquer la prédiction d'un modèle pour une instance particulière en considérant chaque caractéristique comme un "joueur" dans un jeu coopératif où la "récompense" est la différence entre la prédiction actuelle et la prédiction moyenne du modèle.

Dans le contexte de notre étude sur le cancer du sein, pour comprendre pourquoi le modèle a fait une prédiction particulière, SHAP calcule la contribution de chaque caractéristique en observant comment la prédiction change lorsque cette caractéristique est présente ou absente, en considérant toutes les combinaisons possibles avec les autres caractéristiques.

Fondement mathématique

Définition formelle

Pour un modèle f et une instance x à expliquer, la valeur de Shapley ϕ_i pour la caractéristique i est définie par :

$$\phi_i(f,x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|F|-|S|-1)!|S|!}{|F|!}[f_x(S \cup \{i\}) - f_x(S)] \quad (1)$$

où :

- F est l'ensemble de toutes les caractéristiques
- S est un sous-ensemble de caractéristiques
- $f_x(S)$ est la prédiction du modèle pour l'instance x en utilisant uniquement les caractéristiques dans S

Propriétés fondamentales

Les valeurs SHAP satisfont quatre propriétés essentielles :

1. **Efficacité locale** :

$$\sum_i \phi_i = f(x) - E[f(x)] \quad (2)$$

© The Author 2024. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

La somme des contributions égale la différence entre la prédiction et la moyenne des prédictions.

2. **Cohérence** : Si un modèle change de telle sorte que la contribution marginale d'une caractéristique augmente ou reste constante, sa valeur de Shapley ne diminue pas.
3. **Équité** : Si deux caractéristiques contribuent toujours de manière identique pour toutes les coalitions possibles, leurs valeurs de Shapley sont égales.
4. **Additivité** : Pour deux modèles f et g, $\phi_i(f+g) = \phi_i(f) + \phi_i(g)$

Calcul pratique

En pratique, le calcul exact des valeurs de Shapley est #P-hard car il nécessite d'évaluer toutes les combinaisons possibles de caractéristiques. SHAP propose plusieurs approximations :

- **KernelSHAP** : Utilise une régression linéaire pondérée pour approximer les valeurs de Shapley
- **TreeSHAP** : Algorithme spécifique pour les modèles basés sur les arbres (comme dans notre cas avec XGBoost)
- **DeepSHAP** : Adaptation pour les réseaux de neurones profonds

L'équation fondamentale de SHAP pour une explication est :

$$g(z') = \phi_0 + \sum_i \phi_i z'_i \tag{3}$$

où :

- g est la fonction d'explication
- z' est le vecteur de présence/absence des caractéristiques
- ϕ_0 est la valeur moyenne des prédictions
- ϕ_i sont les valeurs de Shapley

Cette formulation additive permet une interprétation intuitive: chaque terme $\phi_i z'_i$ représente la contribution de la caractéristique i à la prédiction.

Application et Résultats

Description des données utilisées

Nous avons utilisé le jeu de données breast cancer de scikit-learn, qui est un ensemble de données de référence pour la détection du cancer du sein. Ce dataset contient des mesures extraites d'images numériques de de noyeaux de cellules issues de prélèvements à l'aiguille fine (FNA) de tumeurs mammaires. Pour chaque image, 30 caractéristiques quantitatives du nucleus cellulaire sont calculées , décrivant des propriétés comme :

- Le rayon (moyenne des distances du centre aux points du périmètre)
- La texture (écart-type des valeurs en niveaux de gris)
- Le périmètre
- L'aire
- La smoothness (variation locale des longueurs des rayons)
- La compacité (périmètre^2 / aire - 1.0)
- La concavité (sévérité des portions concaves du contour)
- Les points concaves (nombre de portions concaves du contour)
- La symétrie
- La dimension fractale

Pour chaque caractéristique, trois valeurs sont calculées : la moyenne, l'erreur standard, et la "pire" valeur (moyenne des trois plus grandes valeurs), donnant ainsi les 30 caractéristiques.

Echelle globale: Analyse de l'importance des features

Le summary plot sous forme de beeswarm montre l'impact de chaque caractéristique sur les prédictions du modèle. En effet chaque point correspond à la valeur shapley d'une caractéristique d'une observation. La couleur désigne l'intensité de la mesure de la caractéristique et non de la valeur shapley. (Fig. 1). Plusieurs observations clés émergent :

- La caractéristique "mean concave points" apparait comme la caractéristique la plus influente (en moyenne les valeurs shapley les plus extrêmes).
- La caractéristique "area error" montre une distribution bimodale intéressante
- La caractéristique "worst area" présente une corrélation positive forte avec la probabilité de cancer
- Les "worst concave points" montrent une distribution asymétrique des impacts

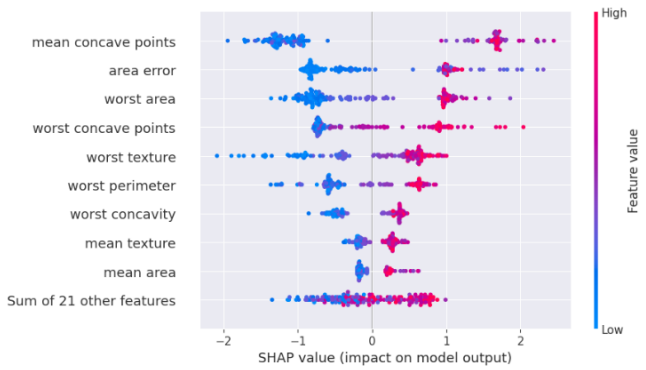

Fig. 1: Distribution des valeurs SHAP pour chaque caractéristique.

Echelle de la caractéristique: Analyse spécifique des "worst concave points"

L'analyse détaillée des "worst concave points" révèle une relation non-linéaire sophistiquée (Fig. 2) :

- Un seuil critique est visible autour de 0.15
- Pour les valeurs inférieures à 0.10, l'impact SHAP est stable et négatif (-0.0)
- Au-delà de 0.15, on observe une augmentation rapide de l'impact SHAP positif sur le diagnostic d une tumeur maligne
- La dispersion des points indique des interactions complexes

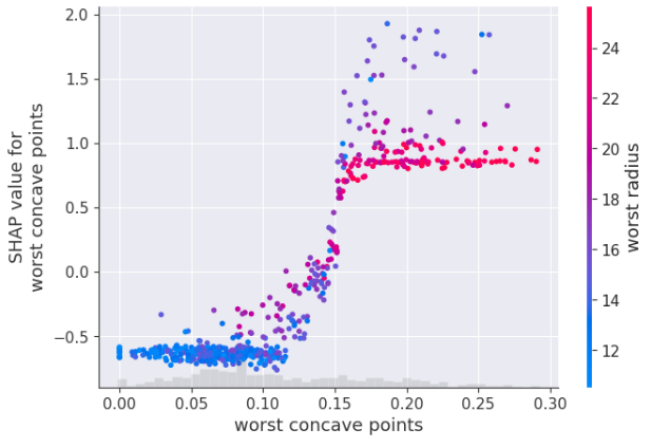

Fig. 2: Relation entre la valeur des "worst concave points" (abscisses) et leur impact SHAP (ordonnées). La couleur indique la valeur du "worst radius".

Echelle d'une observation: Décomposition d'une prédiction individuelle

Le waterfall plot offre une décomposition détaillée d'une prédiction spécifique (Fig. 3) :

- La valeur de base (E[f(X)] = -0.911) représente les valeur de ϕ_0
- "Area error" montre la plus forte contribution positive (+1.19)
- "Mean texture" est la seule caractéristique avec une contribution négative (-0.22)
- L'explication shapley finale g(x) = 5.775 résulte de l'accumulation des contributions, qui est dans ce cas très supérieure à la valeur de base.

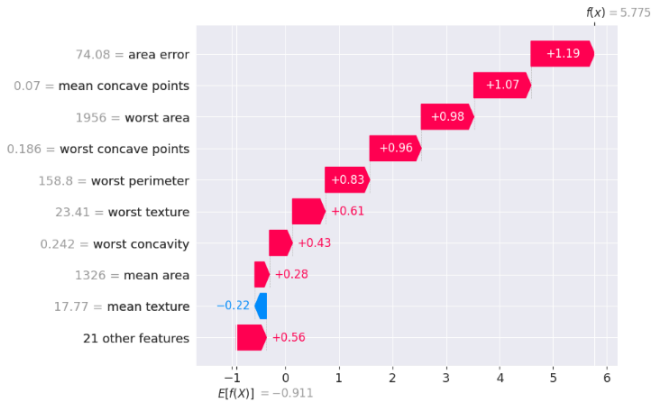

Fig. 3: Décomposition d'une prédiction montrant la contribution de chaque caractéristique.

Visualisation des interactions en terme de forces

Le force plot illustre la dynamique des features dans la prédiction (Fig. 4) :

- La progression va de la valeur de base (-0.9106) à la prédiction finale (5.78)
- Les caractéristiques en rouge poussent vers une prédiction de cancer
- L'accumulation progressive montre la construction de la décision
- La longueur des segments reflète l'importance des caractéristiques dans le résultat

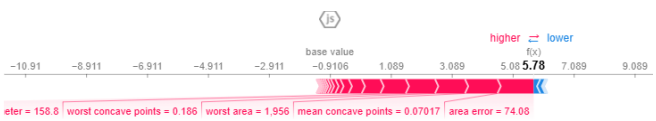

Fig. 4: Forces des contributions des caractéristiques pour déplacer de la Base Value.

Dans cette application au diagnostic du cancer du sein, la méthode SHAP a démontré sa capacité à fournir des explications pertinentes à différentes échelles d'analyse. À l'échelle globale, le beeswarm plot a efficacement hiérarchisé l'importance des caractéristiques tout en révélant des patterns significatifs comme la distribution bimodale de "area error". À l'échelle des caractéristiques individuelles, l'analyse des "worst concave points" a mis en évidence un seuil critique de 0.15, potentiellement pertinent d'un point de vue médical. Enfin, à l'échelle d'une observation individuelle, les visualisations waterfall et force plot ont permis de décomposer de manière transparente le processus de décision du modèle, montrant comment chaque caractéristique contribue à la prédiction finale.

Diverse Counterfactual Explanations (DiCE)

Fondements théoriques

Contexte et motivation

Les explications contrefactuelles (counterfactual explanations), introduites notamment par Wachter et al. en 2017 [7], représentent une approche novatrice qui permet d'expliquer les décisions automatiques sans avoir à ouvrir la "boîte noire" des algorithmes. Ces explications prennent typiquement la forme : "Vous avez été refusé pour un prêt car votre revenu annuel était de 30 000€. Si votre revenu avait été de 45 000€, vous auriez obtenu le prêt." Cette méthode est particulièrement pertinente car elle fournit une explication actionnable sans nécessiter la compréhension des mécanismes internes complexes de l'algorithme.

Principe fondamental

DiCE (Diverse Counterfactual Explanations) étend le concept d'explications contrefactuelles en cherchant à générer un ensemble de k exemples contrefactuels $\{c_1, c_2, ..., c_k\}$ qui modifient la décision du modèle f pour une instance donnée x. La méthode DiCE se distingue par sa prise en compte simultanée de trois aspects clés :

- **Proximité** : Les exemples générés doivent minimiser la distance avec l'instance originale pour garantir des explications réalistes et actionnables

- **Diversité** : Les exemples doivent explorer différentes possibilités de changement pour offrir plusieurs options à l'utilisateur
- **Sparsité** : Les changements doivent impliquer un nombre minimal de caractéristiques pour faciliter la compréhension

La diversité est mesurée de manière rigoureuse via un processus ponctuel déterminantal (DPP) qui encourage la génération d'exemples contrefactuels différents les uns des autres :

$$dpp\_diversity = det(K) \tag{4}$$

où $K_{i,j} = \frac{1}{1+dist(c_i,c_j)}$ est une matrice de noyau construite à partir d'une métrique de distance *dist* entre les exemples contrefactuels.

Formalisation mathématique

DiCE formalise la génération d'exemples contrefactuels comme un problème d'optimisation multi-objectif. Pour une instance donnée x, la méthode minimise la fonction $C(x)$ suivante:

$$C(x) = \arg \min_{c_1,c_2,\ldots,c_k} [L_{pred} + L_{prox} - L_{div}] \tag{5}$$

où les trois composantes sont définies comme suit:

$$L_{pred} = \frac{1}{k} \sum_{i=1}^{k} yloss(f(c_i), y) \tag{6}$$

$$L_{prox} = \frac{\lambda_1}{k} \sum_{i=1}^{k} dist(c_i, x) \tag{7}$$

$$L_{div} = \lambda_2 \cdot dpp\_diversity(c_1, c_2, ..., c_k) \tag{8}$$

avec :

- L_{pred} : perte de prédiction mesurant l'écart entre la prédiction $f(c_i)$ et la classe désirée y
- L_{prox} : terme de proximité quantifiant la distance entre les exemples contrefactuels et l'instance originale
- L_{div} : terme de diversité basé sur le déterminant de la matrice de noyau
- λ_1, λ_2 : hyperparamètres contrôlant l'importance relative des objectifs

Cette formulation permet de générer des exemples contrefactuels qui sont à la fois proches de l'instance originale (facilitant l'actionnabilité), divers (offrant plusieurs options), et impliquant des changements minimaux (améliorant l'interprétabilité).

où :

- $yloss(f(c_i), y)$ mesure l'écart entre la prédiction et le résultat souhaité
- $dist(c_i, x)$ quantifie la distance avec l'instance originale
- *dpp_diversity* évalue la diversité entre les exemples contrefactuels
- λ_1, λ_2 sont des hyperparamètres de pondération

Mesure de la diversité

La diversité est mesurée via un processus ponctuel déterminantal (DPP) :

$$dpp\_diversity = det(K) \tag{9}$$

où $K_{i,j} = \frac{1}{1+dist(c_i,c_j)}$, avec $dist(c_i, c_j)$ une métrique de distance.

Application et Résultats

Configuration expérimentale

Ici nous utiliserons un sous ensemble du dataset Breast cancer, ou l'on gardera les 9 features les plus impactantes détectées avec Shapley, avec un modèle Random Forest comme classifieur de base. Les paramètres clés du modèle sont :

- Nombre d'arbres : 50
- Profondeur maximale : 8

Le modèle montre une accuracy de 96% équivalente à l'accuracy issue de l'utilisation de toutes les features. Ce qui vient confirmer les résultats de la section précédente.

Pour construire les exemples contrefactuels, nous choisissons à tour de rôle une prédiction positive que l'on cherche à modifier en prédiction négative et vice versa.

Analyse des exemples passant de prédiction à forte confiance positive à négative

Prenons le cas d'une observation initialement classée comme maligne (classe 1) avec une forte confiance :

Table 1. Instance originale avec prédiction maligne

Caractéristique	Valeur
mean concave points	0.08465
area error	224.100006
worst area	3143.0
worst concave points	0.182
worst texture	26.440001
worst perimeter	199.5
worst concavity	0.2861
mean texture	17.25
mean area	1546.0
Prédiction	1

DiCE génère les exemples contrefactuels suivants qui conduiraient à une classification bénigne :

Table 2. Contrefactuels générés pour obtenir une prédiction bénigne

Caractéristique	Original	CF1	CF2	CF3
worst concave points	0.182	0.0	0.0	0.0
worst concavity	0.2861	0.0	0.1	0.2
worst perimeter	199.5	199.5	199.5	76.1
mean texture	17.25	10.46	11.60	17.25
Prédiction	1	0	0	0

L'analyse de ces exemples révèle plusieurs points intéressants :

- **Modifications systématiques** :
  - Le worst concave points est réduit de 0.182 à 0.0 dans tous les cas
  - La worst concavity est significativement réduite
- **Modifications variables** :
  - Le worst perimeter reste soit inchangé (199.5), soit est fortement réduit (76.1)
  - La mean texture varie entre 10.46 et 17.25

Analyse des exemples passant de prédiction à faible confiance négative à positive

Examinons maintenant le cas d'une observation initialement classée comme bénigne (classe 0) :

Table 3. Instance originale avec prédiction bénigne

Caractéristique	Valeur
mean concave points	0.07857
area error	49.849998
worst area	380.5
worst concave points	0.1571
worst texture	19.49
worst perimeter	71.040001
worst concavity	0.8216
mean texture	15.34
mean area	300.200012
Prédiction	0

DiCE génère les exemples contrefactuels suivants pour obtenir une classification maligne :

Table 4. Contrefactuels générés pour obtenir une prédiction maligne

Caractéristique	Original	CF1	CF2	CF3
worst perimeter	71.04	245.40	174.90	172.00
worst area	380.5	380.5	380.5	380.5
worst concave points	0.1571	0.1571	0.1571	0.2000
mean texture	15.34	15.34	29.07	15.34
Prédiction	0	1	1	1

L'analyse des contrefactuels révèle différentes stratégies pour obtenir une classification maligne :

- **Modifications isolées majeures** :
  - Une forte augmentation du worst perimeter (jusqu'à 245.40)
  - Une augmentation significative de mean texture (jusqu'à 29.07)
  - Une augmentation du worst concave points (jusqu'à 0.2000)
- **Combinaisons de modifications** :
  - Chemin 1 : Augmentation importante du worst perimeter seul
  - Chemin 2 : Augmentation modérée du worst perimeter avec changement de texture
  - Chemin 3 : Augmentation modérée du worst perimeter avec augmentation des points concaves

En effet, avec une prédiction initiale à faible confiance, les voies permettant un changement de prédiction semblent être plus fins et variés.

Dans les deux cas, positif vers négatif et négatif vers positif, les modifications nécéssaires au changement de label restent cohérentes avec la nature de la caractéristique et sa contribution dans la classification.

Apport méthodologique de DiCE en contexte clinique

L'utilisation de DiCE dans le contexte du diagnostic du cancer du sein permet de transformer un modèle "boîte noire" en outil d'analyse exploitable cliniquement. La méthode présente plusieurs avantages méthodologiques significatifs :

- **Quantification des seuils décisionnels**
- **Identification des voies alternatives**
- **Caractérisation des zones de sensibilité**

Cette approche permet ainsi d'extraire, à partir d'un modèle complexe, des règles décisionnelles quantitatives directement utilisables dans un cadre clinique, sans nécessiter la compréhension des mécanismes internes du modèle.

Néanmoins, Les contrefactuels générés sont intrinsèquement liés à la qualité du modèle de classification utilisé. Un modèle biaisé ou mal calibré produira des contrefactuels potentiellement trompeurs.

- **Réalisme biologique** : Les combinaisons de modifications proposées, bien que mathématiquement valides, ne correspondent pas nécessairement à des évolutions morphologiques biologiquement plausibles. Par exemple, certains contrefactuels suggèrent des modifications de paramètres qui sont physiologiquement incompatibles.
- **Stabilité des contrefactuels** : La génération des contrefactuels peut être sensible aux paramètres d'initialisation et produire des résultats variables pour une même instance, ce qui nécessite une validation croisée approfondie.

Cette approche, bien que prometteuse pour l'interprétabilité des modèles en contexte clinique, doit donc être utilisée comme un outil complémentaire d'aide à l'analyse, en gardant à l'esprit ses limitations techniques et conceptuelles.

Gradient-weighted Class Activation Mapping (Grad-CAM)

Fondements théoriques

Contexte historique

Grad-CAM est une évolution des méthodes de visualisation des réseaux de neurones convolutifs (CNN), développée pour améliorer l'interprétabilité des décisions des modèles de vision par ordinateur. Cette méthode généralise les Class Activation Mapping (CAM) en utilisant les gradients pour identifier les régions importantes d'une image pour une prédiction spécifique[5].

Intuition physique

L'idée fondamentale de Grad-CAM est de suivre le flux des gradients d'une classe cible à travers le réseau convolutif pour localiser et mettre en évidence les régions de l'image qui ont le plus influencé la décision du modèle. Cette approche permet de générer des cartes de chaleur qui visualisent les zones d'intérêt sans nécessiter de modification de l'architecture du réseau ou de réentraînement.

Fondement mathématique

Définition formelle

Pour une classe c et une couche de convolution donnée avec des cartes de caractéristiques A^k, Grad-CAM calcule une carte de localisation $L^c_{Grad-CAM}$ selon les étapes suivantes :

1. Calcul des gradients de la classe par rapport aux feature maps:

$$\frac{\partial y^c}{\partial A_{ij}^k} \tag{10}$$

2. Pooling global des gradients par feature map :

$$\alpha_k^c = \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{11}$$

3. Combinaison pondérée des feature maps :

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \tag{12}$$

où :

- y^c est le score pour la classe c
- A^k est la k-ième carte de caractéristiques
- α_k^c représente l'importance de la k-ième feature map pour la classe c

Calcul pratique
L'implémentation de Grad-CAM peut être décomposée en plusieurs variantes adaptées à différents besoins d'interprétabilité :

- **Grad-CAM standard** :

$$L_{Grad-CAM}^c(x,y) = ReLU\left(\sum_k \alpha_k^c A^k(x,y)\right) \tag{13}$$

où $A^k(x,y)$ représente l'activation à la position (x,y) de la k-ième feature map.

- **Guided Grad-CAM** :

$$L_{Guided-Grad-CAM}^c = L_{Grad-CAM}^c \odot Guided\_Backprop \tag{14}$$

Cette version combine la localisation grossière de Grad-CAM avec les détails fins du guided backpropagation.

- **Counterfactual Grad-CAM** :

$$\alpha_k^c = -\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{15}$$

En inversant le signe des gradients, cette variante identifie les régions qui contrediraient la prédiction.

Pour les tâches de localisation faiblement supervisées, on applique généralement un seuil τ à la carte de chaleur :

$$L_{binary}^c = \mathbb{1}[L_{Grad-CAM}^c > \tau \cdot \max(L_{Grad-CAM}^c)] \tag{16}$$

Cette binarisation permet de générer des masques de segmentation ou des boîtes englobantes.

Application et Résultats

Configuration expérimentale

Pour cette démonstration de Grad-CAM, nous avons utilisé une image test représentant une interaction entre une enfant et un âne. L'implémentation a été réalisée avec Tensorflow/Keras, suivant l'algorithme standard de Grad-CAM.

L'image originale montre une scène naturelle avec une interaction claire entre les sujets, ce qui en fait un bon cas test pour évaluer la capacité du modèle à identifier les éléments pertinents de la scène.

Fig. 5: Image originale utilisée pour l'analyse

Le prédiction sur l'image est faite avec le modèle Xception, qui renvoie la classe la plus probable parmi les mille classes du dataset Imagenet sur lequel il est entrainé.

Génération de la heatmap

La première étape de visualisation consiste à générer heatmap des zones de la dernière couche de convolution ayant le plus contribué à la prédiction (les gradients les plus élevés) :

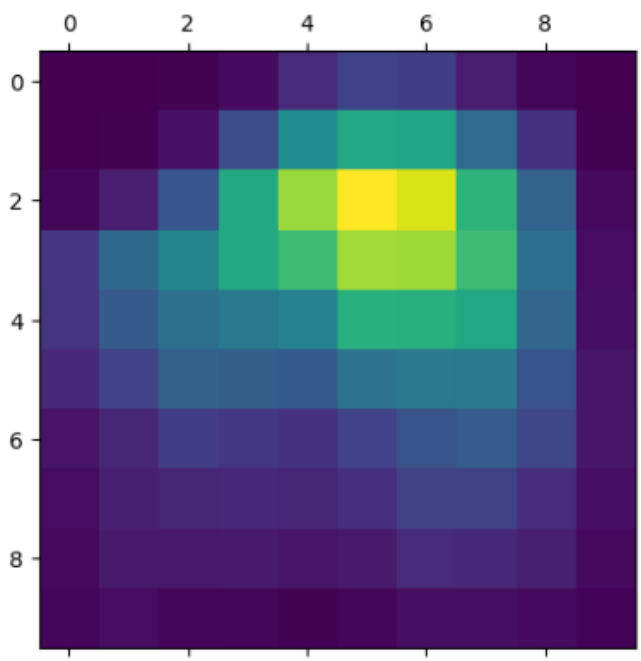

Fig. 6: Carte de chaleur Grad-CAM brute

Cette heatmap montre une zone signifiante dans le haut droit de l'image. Néanmoins la taille de la dernière couche de convolution est de 10*10, afin de visualiser la relation entre pixels de l'image et les gradients associés, on superpose les deux.

Superposition et visualisation finale

Pour une interprétation plus intuitive, la carte de chaleur est superposée à l'image originale :

Fig. 7: Superposition de la carte de chaleur sur l'image originale

Cette visualisation finale permet plusieurs observations :

- **Zone haute activation (jaune)** : Concentrée sur la tête de l'âne, particulièrement autour des traits distinctifs comme les oreilles et les yeux
- **Zone moyenne activation (vert)** : Couvre la zone d'interaction entre l'enfant et l'animal
- **Zone faible activation (bleu foncé)** : Correspond au fond et aux éléments non pertinents de la scène

Le modèle se concentre principalement sur les yeux et des zones pileuses du haut de la tête de l'animal.

La zone d'interaction entre l'enfant et l'animal reçoit une attention modérée

Le fond et les éléments de contexte sont largement ignorés par le modèle

Cette visualisation intuitive et claire doit néanmoins prendre en compte d'autres éléments comme:

- **La prédiction** : Malgré la facilité de l'exemple et la finesse du modèle, la classe prédite dans notre cas est un "Lama".

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
Classe prédite : llama
Score : 0.6058315
```

Fig. 8: Classe prédite par le modèle

- **Granularité** : la taille réduite de la dernière couche de convolution rend l'identification des détails signifiants de l'image difficile.

Interprétabilité des modèles de language avec la librairie Captum

Fondements théoriques

Contexte

Captum est une Librairie d'interprétabilité qui propose différentes méthodes d'attribution pour comprendre les décisions des modèles de deep learning. Dans le cas des modèles de traitement du langage comme BERT, ces méthodes permettent d'identifier les éléments du texte qui influencent le plus la prédiction [6].

Intuition physique

L'idée fondamentale est de quantifier la contribution de chaque composante de l'entrée (mots, tokens, embeddings) à la décision finale du modèle. Pour ce faire, Captum utilise principalement l'approche des gradients intégrés, qui consiste à mesurer comment la prédiction évolue en passant d'une référence neutre (baseline) à l'entrée réelle.

Fondement mathématique

La méthode des gradients intégrés

Pour un modèle f et une entrée x, les gradients intégrés sont définis par :

$$IG_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha \tag{17}$$

où :

- x' est la baseline (référence neutre)
- $\frac{\partial f}{\partial x_i}$ est le gradient de la sortie par rapport à l'entrée i
- α paramètre l'interpolation linéaire entre la baseline et l'entrée réelle

Le terme $(x_i - x'_i)$ pondère l'intégrale par l'écart entre l'entrée et la baseline, reflétant ainsi l'impact effectif de ce changement sur la prédiction.

Implémentation dans Captum

Trois approches principales sont utilisées :

- **LayerIntegratedGradients** : Calcule les attributions pour une couche spécifique.
- **LayerConductance** : Analyse la contribution de chaque couche du réseau.
- **Multi-Embedding Attribution** : Décompose l'attribution entre les différents types d'embeddings.

Application et Résultats

Configuration expérimentale

Pour ce test, nous avons utilisé :

- **Modèle** : bert-large-uncased-whole-word-masking-finetuned-squad de HuggingFace
- **Exemple analysé** :
  - Question : "Who was the King of France in 1661?"
  - Contexte : "In 1661, Louis XIV personally took power after the death of Mazarin. He then became the absolute monarch and established his rule over France."
  - Réponse attendue : "Louis XIV"

Analyse des attributions par embedding global

Première approche utilisant LayerIntegratedGradients sur la couche d'embedding complète :

Les résultats montrent :

- Pour la position de début (prédiction de "Louis") : forte attribution sur "Who" et "1661"
- Pour la position de fin (prédiction de "XIV") : concentration sur "who", "1661", "louis" et un désintérêt pour "France"

Décomposition des attributions par embedding (word, token type, position)

	Word(Index), Attribution	Token Type(Index), Attribution	Position(Index), Attribution
0	personally (16), 0.36	[SEP] (10), 0.36	xiv (15), 0.43
1	after (19), 0.35	[SEP] (40), 0.35	personally (16), 0.32
2	in (11), 0.31	xiv (15), 0.31	after (19), 0.31
3	rule (36), 0.28	in (11), 0.28	who (1), 0.28
4	france (38), 0.27	personally (16), 0.27	in (11), 0.25

Fig. 9: Les 5 token avec la plus forte attribution par type de prédiction, pour le premier token de la réponse

Cette décomposition des attributions par embedding révèle la complexité de la prédiction faite par le modèle, et que les tokens les plus importants pour un type d'embedding ne le sont pas forcément pour les autres types.

Cette analyse tri-dimensionnelle montre que le modèle localise le début de la réponse en combinant des indices lexicaux (mots clés), structurels (type de tokens) et positionnels, avec une attention particulière aux éléments de contexte temporel et à la structure question-réponse.

Analyse de la propagation des attributions à travers les couches

L'utilisation de LayerConductance permet d'observer l'évolution des attributions à travers les couches du modèle :

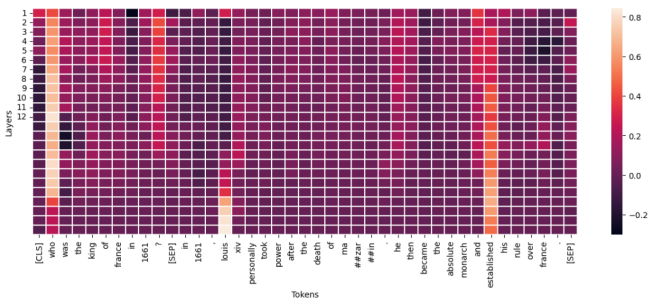

Fig. 10: L'évolution des attributions à travers les couches du modèle par mot d'entrée, pour le premier mot de la prédiction

- **Couches initiales (1-4)** :
  - Focus sur le mot d'interrogation "who"
  - Attribution dispersée sur plusieurs tokens
- **Couches intermédiaires (5-8)** :
  - Concentration progressive sur les tokens pertinents
  - Renforcement des attributions sur "Louis XIV"
- **Couches finales (9-dernière couche)** :
  - Convergence vers les mots le plus pertinent pour la réponse, avec une distribution d'attributions égale aux autres tokens.
  - Forte attribution sur le nom royal "Louis" et son contexte

Limites inhérentes au modèle et à la méthode

L'interprétation des résultats doit être nuancée par plusieurs facteurs liés à l'architecture du modèle et à la méthode d'attribution :

- **Complexité des interactions** :
  - Le modèle BERT, de par son architecture basée sur l'attention, produit des représentations contextuelles où chaque token influence et est influencé par les autres
  - Notre analyse par attribution individuelle ne capture qu'imparfaitement ces interdépendances complexes
  - Par exemple, l'attribution élevée sur "personally" pourrait en réalité provenir de son interaction avec "took power" plutôt que de sa signification isolée
- **Impact du pré-entraînement** :
  - Le modèle BERT-large utilisé a été pré-entraîné sur un large corpus avant son fine-tuning sur SQUAD
  - Les patterns d'attribution observés peuvent refléter des biais acquis durant le pré-entraînement plutôt que des relations causales réelles
  - La généralisation de nos observations est donc limitée par l'histoire d'apprentissage spécifique du modèle
- **Choix méthodologiques** :
  - La méthode des gradients intégrés nécessite une baseline qui influence significativement les résultats
  - Notre utilisation de tokens de padding comme baseline est un choix arbitraire qui pourrait masquer certaines attributions importantes
  - D'autres choix de baseline pourraient produire des attributions sensiblement différentes tout en restant mathématiquement valides

Ces limitations soulignent la nécessité d'interpréter nos résultats comme une perspective partielle sur le fonctionnement du modèle, plutôt qu'une explication exhaustive de son processus de décision.

Layer-wise Relevance Propagation pour réseaux entièrement connectés

Fondements théoriques

Contexte

Layer-wise Relevance Propagation (LRP) est une technique d'explicabilité qui opère en rétropropageant la prédiction à travers le réseau de neurones, au moyen de règles de propagation spécifiquement conçues. LRP exploite la structure en couches du réseau pour calculer rapidement et de manière fiable des explications [1].

Dans le cas des réseaux entièrement connectés (FC), LRP présente un intérêt particulier car la propagation peut être formalisée de manière exacte, grâce à la simplicité de l'architecture.

Principe fondamental

LRP repose sur un principe de conservation, selon lequel la quantité de pertinence reçue par un neurone doit être redistribuée intégralement à la couche inférieure. Cette conservation rappelle, par analogie, les lois de Kirchhoff dans les circuits électriques. Pour un neurone j de la couche inférieure et un neurone k de la couche supérieure, la propagation des scores de pertinence R_k vers la couche inférieure s'exprime par la règle :

$$R_j = \sum_k \frac{z_{jk}}{\sum_{j'} z_{j'k}} R_k \tag{18}$$

où z_{jk} quantifie l'influence du neurone j sur l'activation du neurone k (par exemple, $z_{jk} = x_j\, w_{jk}$). Le dénominateur, qui somme sur tous les neurones j' contribuant à k, garantit la propriété de conservation de la pertinence.

Fondement mathématique

Pour un réseau FC utilisant une fonction d'activation ReLU, la propagation de la pertinence s'effectue en trois étapes principales :

1. Propagation au niveau de la couche de sortie

Pour un neurone de sortie k, la pertinence R_k est directement égale à sa sortie. Dans le cas d'un pooling par somme, on a :

$$R_k = x_k = \sum_j x_j \tag{19}$$

2. Propagation dans la couche cachée

Pour un neurone j de la couche cachée, la sortie est calculée via l'opération :

$$x_j = \max\left(0, \sum_i x_i\, w_{ij} + b_j\right) \tag{20}$$

Dans cette implémentation, la pertinence R_j est directement associée à l'activation x_j du neurone.

3. Propagation vers la couche d'entrée

Pour un neurone d'entrée i, la pertinence R_i est répartie à partir des pertinences de la couche cachée selon la règle :

$$R_i = \sum_j \frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2} R_j \tag{21}$$

Cette formulation, qui utilise le carré des poids, permet de tenir compte de l'importance relative des connexions, qu'elles soient positives ou négatives, tout en garantissant la non-négativité des pertinences.

Application et Résultats

Configuration expérimentale

Pour démontrer l'application de LRP sur un réseau FC, nous avons implémenté un réseau simple avec :

- Une couche d'entrée comportant 3 neurones,
- Une couche cachée comportant 5 neurones avec activation ReLU,
- Une couche de sortie composée d'un unique neurone utilisant un pooling par somme.

Les poids du réseau ont été initialisés selon une distribution normale $\mathcal{N}(0, 0.05)$ afin d'assurer une dispersion adéquate des valeurs.

Vérification des propriétés fondamentales

L'implémentation a été validée en vérifiant deux propriétés essentielles de LRP :

1. **Positivité** : Pour chaque entrée $\mathbf{x}$ et pour tout neurone p, la pertinence doit être positive :

$$\forall \mathbf{x}, \forall p : \quad R_p(\mathbf{x}) \geq 0 \tag{22}$$

2. **Conservation** : La somme des pertinences est conservée d'une couche à l'autre, c'est-à-dire :

$$\sum_i R_i = \sum_j R_j = \sum_k R_k \tag{23}$$

Ces propriétés ont été vérifiées empiriquement au moyen d'un ensemble de tests unitaires systématiques.

Analyse des résultats

L'application de LRP sur notre réseau FC a permis de faire les observations suivantes :

- Les pertinences se propagent de manière cohérente à travers les couches, respectant ainsi la conservation de la somme totale.
- L'impact des poids négatifs et positifs est correctement pris en compte grâce à l'utilisation du carré des poids dans la répartition de la pertinence.
- La non-linéarité introduite par la fonction ReLU est bien intégrée dans le calcul des pertinences de la couche cachée.

Limitations et considérations pratiques

Bien que LRP soit particulièrement adaptée aux réseaux FC, certaines limitations doivent être prises en compte :

- **Sensibilité à l'initialisation** : Les résultats peuvent varier significativement selon l'initialisation des poids.
- **Impact du biais** : La présence de termes de biais peut compliquer l'interprétation des pertinences.
- **Échelle des pertinences** : Les valeurs absolues des pertinences peuvent être difficiles à interpréter sans une normalisation adéquate.

En pratique, il est plus judicieux de :

- Normaliser les entrées afin d'obtenir des échelles comparables,
- Vérifier la stabilité des explications en testant différentes initialisations,
- Comparer les pertinences relatives entre les neurones plutôt que leurs valeurs absolues.

GNN-LRP : Explicabilité des Graph Neural Networks via la Propagation de Pertinence

Fondements théoriques

Contexte

Les Graph Neural Networks (GNN) sont des modèles de deep learning capables de traiter des données structurées sous forme

de graphes. Afin d'interpréter les prédictions de ces modèles, la méthode **Graph Neural Network Layer-wise Relevance Propagation (GNN-LRP)** a été proposée pour fournir une explication fine du processus décisionnel en identifiant les composantes du graphe qui influencent le plus la prédiction. Contrairement aux approches classiques d'explicabilité, GNN-LRP exploite explicitement la structure du graphe et la propagation des informations entre les nœuds pour quantifier la contribution (ou pertinence) de différentes régions du graphe[4, 2].

Principe fondamental

La méthode GNN-LRP repose sur le même principe de conservation que la LRP classique : la somme des scores de pertinence est préservée à travers les différentes étapes de la propagation, de la sortie vers les entrées. Dans le cas d'un GNN, cette rétropropagation tient compte non seulement des connexions entre nœuds mais également des interactions via des chemins (ou "walks") dans le graphe.

Plus précisément, pour une prédiction issue d'un GNN, la pertinence globale (associée à la classe cible) est décomposée en contributions provenant de différents chemins dans le graphe. Chaque chemin, composé d'une séquence de nœuds et d'arêtes, reçoit une attribution de pertinence qui est ensuite visualisée sous forme de courbes colorées, permettant ainsi d'identifier les parties du graphe ayant une influence positive (en rouge) ou négative (en bleu) sur la décision du modèle.

Fondement mathématique

Le calcul de la pertinence dans GNN-LRP s'effectue en plusieurs étapes correspondant aux différentes couches du GNN. Soit un GNN composé d'une architecture de type `GraphNet` dont les poids sont notés U, W_1, W_2 et V. Le passage avant (forward pass) est réalisé en appliquant successivement des opérations linéaires suivies d'une activation de type ReLU, ce qui garantit la non-négativité des activations intermédiaires. Pour la rétropropagation de la pertinence, la méthode procède comme suit :

1. **Première couche (entrée vers couche cachée)** : À partir des activations initiales (typiquement l'identité pour simuler l'entrée brute), la pertinence est propagée via une opération de type
$$Q_1 = \left(\frac{P_1}{P_1^+ + \varepsilon} \right) \odot P_1^+,$$
où $P_1 = A^\top H_0 W_1$ et P_1^+ représente une version positive de P_1, ajustée par un hyperparamètre γ (i.e., $W_1^+ = W_1 + \gamma \cdot \max(0, W_1)$). La division se fait de manière élément-wise, assurant la conservation de la pertinence.
2. **Deuxième couche (couche cachée intermédiaire)** : La même stratégie est appliquée pour la seconde couche avec les poids W_2. On calcule alors
$$Q_2 = \left(\frac{P_2}{P_2^+ + \varepsilon} \right) \odot P_2^+,$$
où $P_2 = A^\top H_1 W_2$ et P_2^+ est la version positive de P_2 obtenue via $W_2^+ = W_2 + \gamma \cdot \max(0, W_2)$.
3. **Troisième couche (sortie)** : La dernière étape consiste à propager la pertinence via le poids de sortie V (avec $V^+ = V + \gamma \cdot \max(0, V)$) afin d'obtenir une attribution finale, qui est ensuite moyennée pour extraire la pertinence associée à la classe cible.

L'opération globale de rétropropagation de la pertinence suit ainsi le schéma :

$$\text{Pertinence} \quad R = f_{\text{LRP}}(A, \{U, W_1, W_2, V\}, \gamma),$$

avec A représentant la matrice d'adjacence (ou son laplacien) et γ un hyperparamètre permettant d'ajuster l'effet des parties positives des poids.

Par ailleurs, l'approche intègre une méthode de visualisation basée sur la construction de courbes de pertinence. Ces courbes, dont les coordonnées sont calculées en fonction des positions des nœuds (issue d'un layout, tel que Kamada-Kawai), permettent de représenter graphiquement l'influence d'un chemin (walk) dans le graphe sur la décision du modèle.

Application et Résultats

Configuration expérimentale

Notre étude s'est concentrée sur l'analyse de graphes scale-free générés selon le modèle de Barabási-Albert avec les paramètres suivants :

- Nombre de nœuds : 10
- Facteur de croissance : 1, 2
- Taille de couche cachée : 64
- Paramètre de LRP : 0.1

Le modèle GNN a été entraîné à classifier le facteur de croissance des graphes scale-free, atteignant une précision de test supérieure à 90

Analyse des patterns d'explication

L'application de GNN-LRP a produit des visualisations distinctes pour les différents facteurs de croissance, comme illustré dans la Figure 11. L'analyse des relevances révèle des patterns distinctifs:

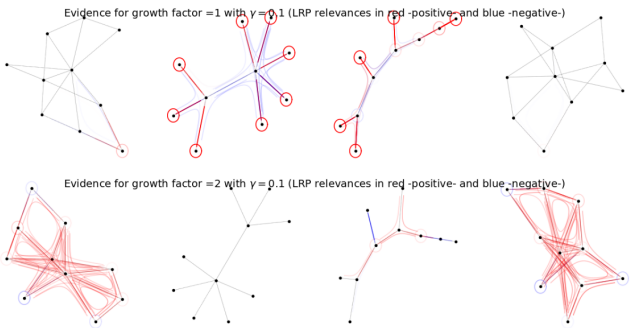

Fig. 11: Visualisation des relevances LRP pour différents graphes scale-free. Les chemins rouges indiquent une contribution positive à la prédiction du facteur de croissance, les bleus une contribution négative.

Pour le facteur de croissance 1:

- Les chemins pertinents (en rouge) sont principalement des chemins terminaux, partant des nœuds centraux vers les feuilles du graphe

- La distribution des relevances est plus éparse, avec des chemins bien définis et peu de chevauchement
- Les chemins négatifs (en bleu) sont rares, suggérant une identification claire des caractéristiques du facteur 1

Pour le facteur de croissance 2 :

- Les relevances positives forment des motifs denses et entrecroisés, particulièrement autour des nœuds à forte centralité
- Les chemins se chevauchent fréquemment, créant des zones de haute intensité de relevance
- La structure plus dense des relevances reflète la connectivité accrue caractéristique du facteur de croissance 2

Interprétation des patterns

Les visualisations mettent en évidence comment le GNN distingue les facteurs de croissance :

- **Structure topologique** : Le modèle identifie des motifs topologiques spécifiques à chaque facteur de croissance, visibles dans la distribution spatiale des relevances Copy
- **Centralité et connectivité** : Les nœuds centraux jouent un rôle plus important dans l'identification du facteur 2, comme en témoigne la concentration des relevances positives
- **Densité des explications** : La densité des chemins pertinents augmente avec le facteur de croissance, reflétant la structure plus interconnectée des graphes à facteur de croissance plus élevé

Ces résultats démontrent la capacité de GNN-LRP à capturer et visualiser les caractéristiques structurelles que le GNN utilise pour sa classification.

Limitations et considérations pratiques

L'application de LRP aux GNN présente certaines limitations spécifiques :

- **Complexité computationnelle** : Le calcul des relevances pour tous les chemins de longueur 3 peut devenir coûteux pour les grands graphes.
- **Sensibilité au paramètre** γ : Le choix de γ influence significativement la distribution des relevances et donc l'interprétation des résultats.
- **Interprétation des chemins** : La visualisation peut devenir confuse lorsque de nombreux chemins se chevauchent.

Conclusion

Ce travail nous a permis d'explorer et de mettre en pratique six méthodes d'explicabilité de l'IA à travers différents projets. L'implémentation de ces méthodes sur des cas concrets nous a permis de mieux comprendre leurs principes de fonctionnement et leurs applications spécifiques.

Les valeurs de Shapley (SHAP) ont montré leur efficacité pour quantifier l'importance des caractéristiques dans le diagnostic du cancer du sein, offrant une interprétation à la fois globale et locale des prédictions du modèle.

La méthode DiCE a fourni des explications contrefactuelles pertinentes, permettant d'identifier les modifications nécessaires pour changer une prédiction, bien que la plausibilité biologique des suggestions reste un point d'attention.

L'application de Grad-CAM à l'analyse d'images nous a permis de visualiser les zones d'intérêt du modèle, même si la prédiction finale (confusion entre âne et lama) souligne l'importance d'une interprétation critique des résultats.

L'utilisation de Captum pour l'analyse de BERT a révélé la complexité des mécanismes de prédiction dans les modèles de langage, montrant comment différents tokens d'entrée et de contexte contribuent à la prédiction finale.

Enfin, l'exploration de LRP, tant dans sa version classique que dans son application aux réseaux de neurones sur graphes, a démontré l'importance de la propagation de la pertinence pour comprendre le cheminement des décisions à travers les couches du réseau.

References

1. Sebastian Bach, Alexander Binder, Gr'egoire Montavon, Frederick Klauschen, Klaus-Robert M"uller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.
2. Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.
3. Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.
4. Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10772–10781, 2019.
5. Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
6. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017.
7. Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31:841, 2017.