Documentazione della Pipeline Immagini
`xai_img`

June 27, 2025

1. Contesto

è stata implementata una pipeline per analizzare il *Disagreement Problem* su spiegazioni visuali: si confrontano due mappe di salienza generate dallo stesso modello (o da modelli diversi) su una medesima immagine, quantificando la loro divergenza con quattro metriche.

2. Tecnologie e librerie

È stato utilizzato Python 3.10 in ambiente `conda (xai)`, con:

- **PyTorch** e **torchvision** per il caricamento di una rete pre-addestrata (ResNet-18);
- **Captum** per i metodi di interpretazione Integrated Gradients e Saliency;
- **scikit-image** per il caricamento e la segmentazione SLIC in super-pixel;
- **NumPy** per la manipolazione delle mappe di salienza;
- **Matplotlib** per la visualizzazione finale delle heat-map.

3. Metodi di interpretazione

3.1 Integrated Gradients (IG)

Integrated Gradients è un explainer che stima l'importanza di ciascun pixel (o feature) calcolando l'integrale del gradiente lungo un percorso che va da un *baseline* (un'immagine nera) all'input reale. In pratica:

1. si definisce un *baseline* x' (vettore di zeri);
2. si suddivide il segmento $[x', x]$ in m passi e si calcolano i gradienti in punti intermedi;
3. si sommano i gradienti e si moltiplica per la variazione $x - x'$ per ottenere un valore di attributo per ciascun pixel.

Aumentando m (`n_steps`) l'approssimazione diventa più fedele all'integrale continuo ma aumenta il costo computazionale. (se non viene passato `n_steps`, Captum userà `n_steps = 50` di default)

3.2 Saliency (gradiente puro)

Il metodo *Saliency* è la forma più semplice di explainability basata sul gradiente: si calcola il gradiente dell'output di interesse rispetto all'input, pixel per pixel. Il valore assoluto di questo gradiente indica quanto la predizione cambierebbe modificando lievemente quel pixel. Rispetto a IG, Saliency:

- è istantaneo (un solo backward pass);
- tende ad essere più "rumoroso";
- non richiede un *baseline* né parametri aggiuntivi.

4. Flusso di lavoro

La pipeline si articola in due fasi: generazione delle mappe di salienza e confronto quantitativo e visivo.

4.1 Generazione delle mappe (`make_maps.py`)

1. Caricamento di ResNet-18 con pesi ImageNet: `models.resnet18(weights=ResNet18_Weights.DEFAULT)`.
2. Preprocessing dell'immagine (`cat.jpg`): resize a 224×224, conversione in tensore, normalizzazione.
3. Predizione della classe di riferimento: `pred = model(x).argmax()`.
4. Baseline: vettore di zeri di forma $(1, 3, 224, 224)$ per IG.
5. Calcolo di:
   - `IntegratedGradients.attribute(x, baselines=x*0, target=pred)` → mappa IG
   - `Saliency.attribute(x, target=pred)` → mappa gradiente puro
6. Salvataggio in `.npy`: `saliency_A.npy`, `saliency_B.npy`.

4.2 Confronto delle mappe (`compare_saliency.py`)

Ridimensionamento L'immagine originale viene riportata alle dimensioni delle mappe caricate.

Super-pixel Si applica SLIC con $N_{\text{segments}} = 200$ per aggregare pixel in blocchi omogenei, ottenendo n_{sp} super-pixel.

Vettori di salienza Per ogni super-pixel si calcola la media della salienza:

$$v_i = \frac{1}{|S_i|} \sum_{(x,y)\in S_i} \text{saliency}(x, y),$$

da cui si ottengono due vettori $\mathbf{v}^{(A)}, \mathbf{v}^{(B)} \in \mathbb{R}^{n_{\text{sp}}}$.

Metriche di disaccordo Si applicano:

- *Feature Disagreement*: $1 - \frac{|\mathrm{Top}_k(\mathbf{v}^{(A)}) \cap \mathrm{Top}_k(\mathbf{v}^{(B)})|}{k}$;
- *Sign Disagreement*: penalizza differenze di segno;
- *Euclidean*: distanza L_2 tra vettori normalizzati;
- *Euclidean-abs*: distanza L_2 tra moduli normalizzati.

Impostata $K_{\mathrm{FRAC}} = 0.05 \implies k = 0.05\, n_{\mathrm{sp}}$.

5. Risultati

Eseguendo:

```
python make_maps.py
python compare_saliency.py
```

si ottiene:

$$\begin{aligned} \text{FeatureDisagreement} &= 1.000, \\ \text{SignDisagreement} &= 1.000, \\ \text{Euclidean} &= 1.417, \\ \text{Euclidean-abs} &= 0.856. \end{aligned}$$

Questi valori indicano disaccordo massimo sulle top-5% di super-pixel e grande distanza globale, confermando che le due mappe (IG vs Saliency) evidenziano aree significativamente diverse.

5.1 Interpretazione quantitativa

- **FeatureDisagreement = 1.000**: fra i super-pixel "top-k" scelti da saliency A e quelli scelti da saliency B, non c'è alcuna sovrapposizione (100% di disaccordo).
- **SignDisagreement = 1.000**: anche considerando il segno dell'attribuzione (positivo vs negativo), non c'è alcuna corrispondenza nelle stesse regioni.
- **Euclidean $\approx$ 1.417**: misura "globale" di distanza L_2 fra le due distribuzioni di saliency (vettori normalizzati): quindi le due mappe sono molto diverse.
- **Euclidean-abs $\approx$ 0.856**: stessa distanza ma ignorando il segno: resta elevata, il che conferma che anche le intensità "in assoluto" sono molto differenti.

6. Le tre immagini

6.1 Immagine originale

La foto originale (`cat.jpg`), ridimensionata a 224×224 pixel per coincidere con le mappe.

6.2 Saliency A

Overlay della mappa generata con Integrated Gradients. I colori (scala dal verde al rosso, mappati con `plt.cm.jet`) mostrano le aree che IG ritiene importanti: più una zona è "rossastra", più pesa sulla decisione del modello. Qui emerge una colorazione diffusa, con pochissime aree calde concentrate.

Figure 1: Da sinistra: immagine originale, heat-map IG (saliency A), heat-map Saliency (saliency B).

6.3 Saliency B

Overlay della mappa generata con il metodo Saliency (gradiente puro). Qui il colore prevalente è bluastro, con pochi puntini verde-chiaro che indicano dove il gradiente è massimo.

7. Discussione sul Disagreement Problem

Nessuna sovrapposizione nelle aree top-k (FeatureDisagreement = 1) → IG e gradiente puro indicano regioni completamente diverse come "più importanti".

Anche le intensità complessive non sono vicine (Euclidean = 1.417) → l'intero pattern di saliency è distante.

Visivamente, IG ha evidenziato una zona ampia e sfumata (al centro del manto), mentre Saliency ha punti sparsi in tutta l'immagine.

Due explainers, applicati allo stesso modello e alla stessa immagine, raccontano storie molto diverse su quali pixel abbiano spinto la predizione.