Table of content

• Introduction to explainability and Xplique

• What’s new?

• Two demonstrations

• Feed-back from an user
Explainability and Xplique
The black-box problem

- Training Dataset
- Learning Algorithm
- Inputs
- Predictions
Why do we need explainability?

- Build **trust** in the model predictions
Why do we need explainability?

• Build **trust** in the model predictions

• **Satisfy regulatory requirements** and Certification process

Image by rawpixel.com on Freepik
Why do we need explainability?

• Build **trust** in the model predictions

• **Satisfy** regulatory requirements and Certification process

• **Reveal bias** or other unintended effects learned by a model

• Understand to **intervene on models**

• ...

From Ribeiro et al.: "Why Should I Trust You?"
A technical challenge

Model
Feature Viz,
Concept Activation Vector
Explanation ‘by design’
...

© DEEL. All rights reserved to IVADO, IID, IRT Saint Exupéry, CRIAQ and ANITI. Confidential and proprietary document

Iguana
Loggerhead
Green snake
Tiger shark
Tench
Boa

16/04/2024
A technical challenge

Predictions
Feature Attribution
Feature Inversion
...

<table>
<thead>
<tr>
<th>Saliency</th>
<th>Gradient-Input</th>
<th>SmoothGrad</th>
<th>VarGrad</th>
<th>Integrated-Gradients</th>
<th>Occlusion</th>
<th>Grad-CAM</th>
<th>Grad-CAM++</th>
<th>RISE</th>
<th>Greedy-AS</th>
<th>EVA (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
</tr>
</tbody>
</table>
A technical challenge
A technical challenge

- Model
  - Feature Viz,
  - Concept Activation Vector
  - Explanation ‘by design’
  -...

- Predictions
  - Feature Attribution
  - Feature Inversion
  -...

- Data
  - Nearest Neighbourhood
  - Influence Function
  -...

16/04/2024
Xplique: A DL Explainability Toolbox

**Attribution Methods** more than 14 black-box / white-box methods*

- Savity
- Smoothgrad
- Occlusion
- GradCAM
- RSE
- Subbl

```python
from xplique.attributions import GradCAM
explainer = GradCAM(model)
explanations = explainer(x, y)
```

* SOTA

**Feature Visualization**

- Neurons
- Channels
- Directions

Visualize Neurons, Channels, Vectors in activation space (e.g. CAV) or a mix of them!

**Metrics** more than 6 attributions metrics each supporting baselines

- Deletion (low AUC = better faithfulness)
- Insertion* (high AUC = better faithfulness)

```python
from xplique.metrics import Deletion
from xplique.attributions import GradCAM
metric = Deletion(model, x, y)
explanations = GradCAM(model)(x, y)
score = metric(explanations)
```

* SOTA

**Concept based** concept activation vector, CRAFT (new!)

Easily extract and test CAVs:

```python
from xplique.concepts import CAV
extractor = CAV(model, neuron)
concept_vector = extractor(positive_samples, random_samples)
```
What’s new?
List of new features

- PyTorch wrapper
- Operators, they allow to treat new tasks:
  - Object detection
  - Semantic segmentation
- New state-of-the-art attribution methods:
  - Sobol, HSIC, and FORGrad
- Automatic concepts extraction
  - CRAFT
- Last feature visualization methods:
  - MaCO
PyTorch wrapper and operators API

Initial API

```python
from xplique.attributions import saliency
from xplique.metrics import deletion
from xplique.wrappers import TorchWrapper

# load images, targets, and model
# ...

# initialize the explainer with the model and method parameters
explainer = Saliency(model)

# call the explainer on the sample to explain
explanations = explainer(inputs, targets)

# compute explanation metrics
metric = Deletion(model, inputs, targets)
deletion_score = metric(explanation)
```

PyTorch Wrapper

```python
import torch

from xplique.attributions import saliency
from xplique.metrics import deletion
from xplique.wrappers import TorchWrapper

# load images, targets, and torch_model
# ...

# device = 'cuda' if torch.cuda.is_available() else 'cpu'
model = TorchWrapper(torch_model, device)

# initialize the explainer with the model and method parameters
explainer = Saliency(model)

# call the explainer on the sample to explain
explanations = explainer(inputs, targets)

# compute explanation metrics
metric = Deletion(model, inputs, targets)
deletion_score = metric(explanation)
```

Operators

```python
from xplique.attributions import saliency
from xplique.metrics import deletion

# load images, targets, and model
# ...

# initialize the explainer with the model and method parameters
explainer = Saliency(model, operator=xplique.Tasks.SEMANTIC_SEGMENTATION)

# call the explainer on the sample to explain
explanations = explainer(inputs, targets)

# compute explanation metrics
metric = Deletion(model, inputs, targets, operator=xplique.Tasks.SEMANTIC_SEGMENTATION)
deletion_score = metric(explanation)
```
New methods

HSIC

CRAFT
Demonstrations
Demonstrations with Xplique tutorials

- PyTorch wrapper
- Operators, they allow to treat new tasks:
  - Object detection
  - Semantic segmentation
- New state-of-the-art attribution methods:
  - Sobol and HSIC
- Automatic concepts extraction
  - CRAFT
- Last feature visualization methods:
  - MaCO

HSIC on a PyTorch model for semantic segmentation

CRAFT and MaCO
Future Works

Research
- Development of new methods (example-based)
- Extend to new Use Cases
- Development of new metrics
- ....

Tools
- Enhance our library efficiency
- Extend to new Use Cases
- CI/CD, include the newest methods
- ....

Benchmarking
- Provide a thorough User Guide
- Multi-criteria evaluation
- Challenge on more Use Cases
- ....
Feed-back from an user
Want to know more?

- www.deel.ai
- github/deel-ai
- linkedin/showcase/deel-ai

Want to join DEEL?

- gregory dot flandin at irt-saintexupery dot com
- Bâtiment B612
  3 Rue Tarfaya, 31400 Toulouse

Do not forget to leave a star