Datacraft Seminar

FLINT: a Framework to learn with Interpretability

Jayneel Parekh*
(Joint work with Pavlo Mozharovskyi* and Florence d'Alché-Buc*)

*LTCI, Télécom Paris, IP Paris

May 10, 2022

Introduction

- Interpretability is the ability to provide human-understandable insights on the decision process of an AI system

Introduction

- Interpretability is the ability to provide human-understandable insights on the decision process of an AI system

Regarding data-driven AI systems (aka Machine Learning), two primary problem settings for interpretability in literature:

1. Post-hoc approaches
2. Interpretability by design

Introduction

- Interpretability is the ability to provide human-understandable insights on the decision process of an AI system

Regarding data-driven AI systems (aka Machine Learning), two primary problem settings for interpretability in literature:

1. Post-hoc approaches
2. Interpretability by design

We propose a novel framework FLINT – primarily designed to jointly learn a pair of networks (predictor, interpreter), it can be specialized to enable post-hoc interpretability, when a (trained) prediction network is available.

FLINT and related works

Key aspects of FLINT

- *Means of interpretation*: high-level features/concepts.
- *Scope of interpretation*: Local AND Global.

Immediate related works to FLINT

1. Jointly learning predictor & interpreter: GAME – Lee et al (Local interpreter for each sample)
2. Using concepts for interpretation: SENN (Alvarez-Melis & Jaakkola), TCAV-based approaches
3. Applicability to both *by-design & post-hoc* problems: None

Supervised Learning with Interpretation (SLI)

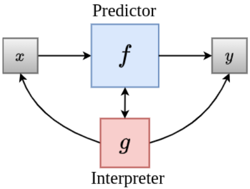
- Generic task SLI: Considers prediction and interpretation as separate tasks with dedicated models f and g.

Supervised Learning with Interpretation (SLI)

- Generic task SLI: Considers prediction and interpretation as separate tasks with dedicated models f and g.

- Optimization problem:

$$\arg \min_{f \in \mathcal{F}, g \in \mathcal{G}_f} \mathcal{L}_{pred}(f, \mathcal{S}) + \mathcal{L}_{int}(f, g, \mathcal{S})$$

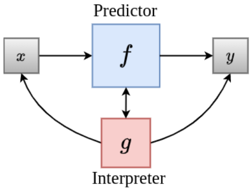


- $\mathcal{F}$ is the space of predictive models. $\mathcal{G}_f$ is family of interpreter models dependent on f.

Supervised Learning with Interpretation (SLI)

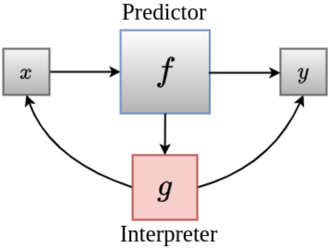
- Generic task SLI: Considers prediction and interpretation as separate tasks with dedicated models f and g.
- Optimization problem:

$$\arg \min_{f \in \mathcal{F}, g \in \mathcal{G}_f} \mathcal{L}_{pred}(f, \mathcal{S}) + \mathcal{L}_{int}(f, g, \mathcal{S})$$



- $\mathcal{F}$ is the space of predictive models. $\mathcal{G}_f$ is family of interpreter models dependent on f.
- Our goal is to address SLI when $\mathcal{F}$ instantiated with deep neural networks and task is multi-class classification.

Specializing SLI: Post-hoc interpretation



- A special case with $f = \hat{f}$ is fixed and we only learn g.
- Optimization problem:

$$\arg\min_{g \in \mathcal{G}_f} \mathcal{L}_{int}(f, g, \mathcal{S}),$$

(No gradients are backpropagated to f.)

Design of FLINT

FLINT: Framework to Learn INTerpretable networks

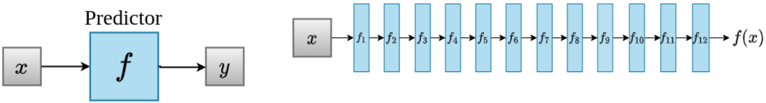
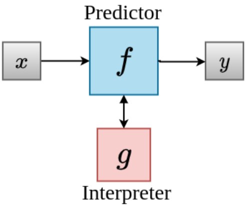


Figure: System Overview

Design of FLINT

FLINT: Framework to Learn INTerpretable networks



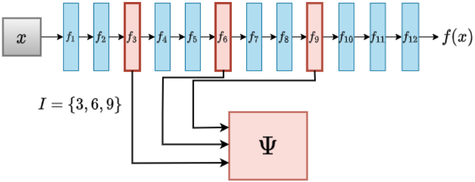


Figure: System Overview

- **Interpreter** $g(x) = h \circ \Psi \circ f_{\mathcal{I}}(x) = h \circ \Phi(x) := \text{softmax}(W^T \Phi(x))$. Computes composition of attribute functions $\Phi(x)$ and interpretable function h characterized by weight matrix W.

Design of FLINT

FLINT: Framework to Learn INTerpretable networks

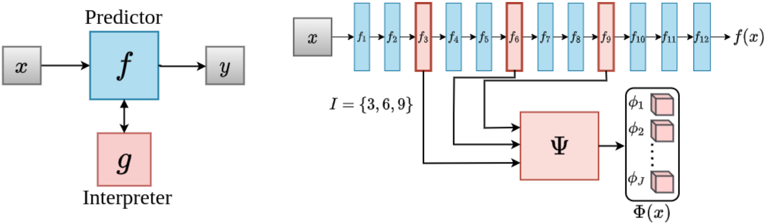


Figure: System Overview

- **Interpreter** $g(x) = h \circ \Psi \circ f_{\mathcal{I}}(x) = h \circ \Phi(x) := \mathrm{softmax}(W^T \Phi(x))$. Computes composition of attribute functions $\Phi(x)$ and interpretable function h characterized by weight matrix W.

Design of FLINT

FLINT: Framework to Learn INTerpretable networks

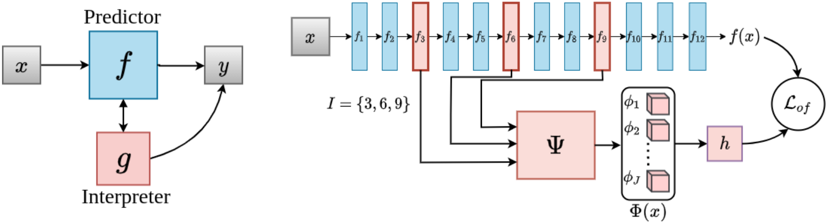


Figure: System Overview

- **Interpreter** $g(x) = h \circ \Psi \circ f_{\mathcal{I}}(x) = h \circ \Phi(x) := \text{softmax}(W^T \Phi(x))$. Computes composition of attribute functions $\Phi(x)$ and interpretable function h characterized by weight matrix W.
- **Attribute dictionary**: functions $\phi_j : \mathcal{X} \to \mathbb{R}^+, j = 1, \ldots J$. $\phi_j(x)$ is activation of some high level attribute, i.e. a "concept" over $\mathcal{X}$.

Design of FLINT

FLINT: Framework to Learn INTerpretable networks

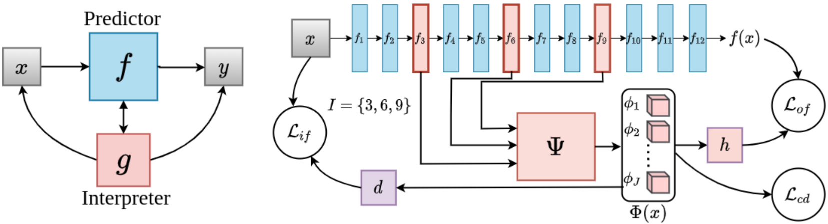


Figure: System Overview

- **Interpreter** $g(x) = h \circ \Psi \circ f_{\mathcal{I}}(x) = h \circ \Phi(x) := \text{softmax}(W^T \Phi(x))$. Computes composition of attribute functions $\Phi(x)$ and interpretable function h characterized by weight matrix W.

- **Attribute dictionary**: functions $\phi_j : \mathcal{X} \to \mathbb{R}^+, j = 1, \ldots J$. $\phi_j(x)$ is activation of some high level attribute, i.e. a "concept" over $\mathcal{X}$.

Losses for Interpretability

- Fidelity to output: $\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} h(\Phi(x))^T log(f(x))$

Losses for Interpretability

- Fidelity to output: $\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} h(\Phi(x))^T log(f(x))$
- *Conciseness and Diversity*: Only a small number of attributes should activate (Conciseness).

Losses for Interpretability

- Fidelity to output: $\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} h(\Phi(x))^T log(f(x))$
- *Conciseness and Diversity*: Only a small number of attributes should activate (Conciseness). However, multiple attributes should be utilized across multiple samples (Diversity). Use of entropy (Jain et al)

Losses for Interpretability

- Fidelity to output: $\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} h(\Phi(x))^T log(f(x))$
- *Conciseness and Diversity*: Only a small number of attributes should activate (Conciseness). However, multiple attributes should be utilized across multiple samples (Diversity). Use of entropy (Jain et al)

$$\bar{\Phi}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \Phi(x)$$

$$\mathcal{L}_{cd}(\Phi, \mathcal{S}) = -\mathcal{E}(\bar{\Phi}_{\mathcal{S}}) + \sum_{x \in \mathcal{S}} \mathcal{E}(\Phi(x)) + \sum_{x \in \mathcal{S}} \eta \|\Phi(x)\|_1$$

- Fidelity to input (via d). To promote encoding high-level patterns, relevant to input, use of autoencoder. (Melis & Jaakkola, 2018):

Losses for Interpretability

- Fidelity to output: $\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} h(\Phi(x))^T log(f(x))$
- *Conciseness and Diversity*: Only a small number of attributes should activate (Conciseness). However, multiple attributes should be utilized across multiple samples (Diversity). Use of entropy (Jain et al)

$$\bar{\Phi}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \Phi(x)$$

$$\mathcal{L}_{cd}(\Phi, \mathcal{S}) = -\mathcal{E}(\bar{\Phi}_{\mathcal{S}}) + \sum_{x \in \mathcal{S}} \mathcal{E}(\Phi(x)) + \sum_{x \in \mathcal{S}} \eta \|\Phi(x)\|_1$$

- Fidelity to input (via d). To promote encoding high-level patterns, relevant to input, use of autoencoder. (Melis & Jaakkola, 2018):

$$\mathcal{L}_{if}(d, \Phi, f, \mathcal{S}) = \sum_{x \in \mathcal{S}} (d(\Phi(x)) - x)^2$$

Losses for Interpretability

- Fidelity to output: $\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} h(\Phi(x))^T log(f(x))$
- *Conciseness and Diversity*: Only a small number of attributes should activate (Conciseness). However, multiple attributes should be utilized across multiple samples (Diversity). Use of entropy (Jain et al)

$$\bar{\Phi}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \Phi(x)$$

$$\mathcal{L}_{cd}(\Phi, \mathcal{S}) = -\mathcal{E}(\bar{\Phi}_{\mathcal{S}}) + \sum_{x \in \mathcal{S}} \mathcal{E}(\Phi(x)) + \sum_{x \in \mathcal{S}} \eta \|\Phi(x)\|_1$$

- Fidelity to input (via d). To promote encoding high-level patterns, relevant to input, use of autoencoder. (Melis & Jaakkola, 2018):

$$\mathcal{L}_{if}(d, \Phi, f, \mathcal{S}) = \sum_{x \in \mathcal{S}} (d(\Phi(x)) - x)^2$$

- Complete interpretability loss term:

$$\mathcal{L}_{int}(f, \Phi, h, d, \mathcal{S}) = \beta \mathcal{L}_{of}(f, \Phi, h, \mathcal{S}) + \gamma \mathcal{L}_{if}(\Phi, h, d, \mathcal{S}) + \delta \mathcal{L}_{cd}(\Phi, \mathcal{S})$$

Generating Interpretations

How do we get local and global interpretability from our trained model?

Generating Interpretations

How do we get local and global interpretability from our trained model?

1. Importance of attribute in prediction of a sample ($r_{j,x}$): Obtain this via attribute activation $\phi_j(x)$ and weight for that attribute $w_{j,\hat{y}}$.

$$r_{j,x} = \frac{\alpha_{j,\hat{y},x}}{\max_i |\alpha_{i,\hat{y},x}|}, \alpha_{j,\hat{y},x} = \phi_j(x).w_{j,\hat{y}}$$

Generating Interpretations

How do we get local and global interpretability from our trained model?

1. Importance of attribute in prediction of a sample ($r_{j,x}$): Obtain this via attribute activation $\phi_j(x)$ and weight for that attribute $w_{j,\hat{y}}$.

$$r_{j,x} = \frac{\alpha_{j,\hat{y},x}}{\max_i |\alpha_{i,\hat{y},x}|}, \alpha_{j,\hat{y},x} = \phi_j(x).w_{j,\hat{y}}$$

2. Average out $r_{j,x}$ for many samples with same predicted class to get a global picture of class-attribute relationships $r_{j,c}$.

$$r_{j,c} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} r_{j,x}, \mathcal{S}_c = \{x \in \mathcal{S} | \hat{y} = c\}$$

Generating Interpretations

How do we get local and global interpretability from our trained model?

1. Importance of attribute in prediction of a sample ($r_{j,x}$): Obtain this via attribute activation $\phi_j(x)$ and weight for that attribute $w_{j,\hat{y}}$.

$$r_{j,x} = \frac{\alpha_{j,\hat{y},x}}{\max_i |\alpha_{i,\hat{y},x}|}, \alpha_{j,\hat{y},x} = \phi_j(x).w_{j,\hat{y}}$$

2. Average out $r_{j,x}$ for many samples with same predicted class to get a global picture of class-attribute relationships $r_{j,c}$.

$$r_{j,c} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} r_{j,x}, \mathcal{S}_c = \{x \in \mathcal{S} | \hat{y} = c\}$$

3. Understanding concept encoded by an attribute.

Generating Interpretations

How do we get local and global interpretability from our trained model?

1. Importance of attribute in prediction of a sample ($r_{j,x}$): Obtain this via attribute activation $\phi_j(x)$ and weight for that attribute $w_{j,\hat{y}}$.

$$r_{j,x} = \frac{\alpha_{j,\hat{y},x}}{\max_i |\alpha_{i,\hat{y},x}|}, \alpha_{j,\hat{y},x} = \phi_j(x).w_{j,\hat{y}}$$

2. Average out $r_{j,x}$ for many samples with same predicted class to get a global picture of class-attribute relationships $r_{j,c}$.

$$r_{j,c} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} r_{j,x}, \mathcal{S}_c = \{x \in \mathcal{S} | \hat{y} = c\}$$

3. Understanding concept encoded by an attribute.

$$1 + 3 \longrightarrow \text{local interpretability}$$

$$2 + 3 \longrightarrow \text{global interpretability}$$

Generating Interpretations

Last piece: How do we understand concept encoded by an attribute ϕ_j?

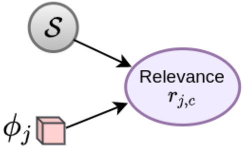


Figure: Flow to understand encoded concept by attribute ϕ_j

- Compute global relevance $r_{j,c}$ (for each class c)

Generating Interpretations

Last piece: How do we understand concept encoded by an attribute ϕ_j?

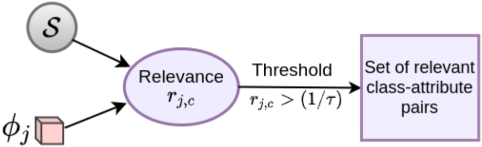


Figure: Flow to understand encoded concept by attribute ϕ_j

- Compute global relevance $r_{j,c}$ (for each class c)
- Select relevant class-attribute pairs by thresholding $r_{j,c}$

Generating Interpretations

Last piece: How do we understand concept encoded by an attribute ϕ_j?

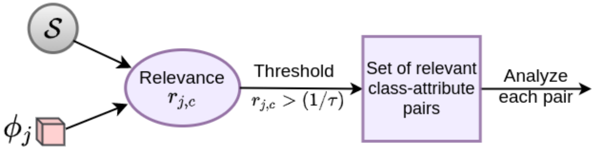


Figure: Flow to understand encoded concept by attribute ϕ_j

- Compute global relevance $r_{j,c}$ (for each class c)
- Select relevant class-attribute pairs by thresholding $r_{j,c}$
- Analyze each pair by repeating this:

Generating Interpretations

Last piece: How do we understand concept encoded by an attribute ϕ_j?

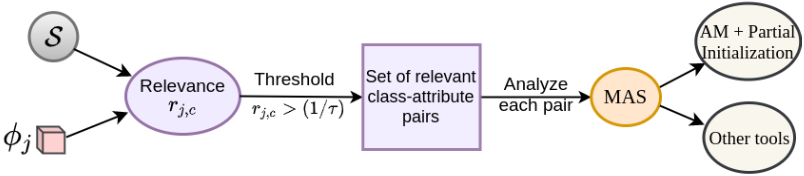


Figure: Flow to understand encoded concept by attribute ϕ_j

- Compute global relevance $r_{j,c}$ (for each class c)
- Select relevant class-attribute pairs by thresholding $r_{j,c}$
- Analyze each pair by repeating this:
  - Select samples of class c maximally activating ϕ_j (MAS)
  - Use Activation Maximization w/ Partial Initialization (AM+PI) as tool – *optimizes* weakly initialized input to maximally activate ϕ_j

Generating Interpretations

Last piece: How do we understand concept encoded by an attribute ϕ_j?

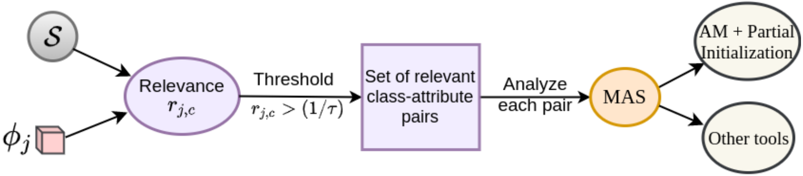
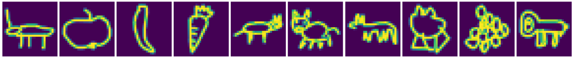


Figure: Flow to understand encoded concept by attribute ϕ_j

- Compute global relevance $r_{j,c}$ (for each class c)
- Select relevant class-attribute pairs by thresholding $r_{j,c}$
- Analyze each pair by repeating this:
  - Select samples of class c maximally activating ϕ_j (MAS)
  - Use Activation Maximization w/ Partial Initialization (AM+PI) as tool – *optimizes* weakly initialized input to maximally activate ϕ_j
- Can use AM+PI to analyze any sample for local interpretations.

Datasets & Networks

- MNIST – LeNet
- FashionMNIST – LeNet
- CIFAR10 – ResNet
- QuickDraw (Hand sketch recognition) – ResNet
  - 10000 random images for 10 classes: 'Ant', 'Apple', 'Banana', 'Carrot', 'Cat', 'Cow', 'Dog', 'Frog', 'Grapes', 'Lion'.



  - 8000 images for training, 2000 for testing.
- Additional results on CIFAR100, CUB-200 (ResNet18)

Quantitative Evaluation

- **Accuracy**: Two goals regarding this
  - Comparison to other related interpretable NN architectures
  - Training f & g jointly does not negatively affect performance.

- **Fidelity of interpreter**: Fraction of samples where prediction of g is same as f.

- **Conciseness of interpretations**: Average number of attributes "important" to interpretations.

$$\mathrm{CNS}_{g,x} = |\{j : |r_{j,x}| > 1/\tau\}|$$

Results – Quantitative I

	BASE-f	SENN	PrototypeDNN	FLINT-f	FLINT-g
MNIST	98.9±0.1	98.4±0.1	**99.2**	98.9±0.2	98.3±0.2
FashionMNIST	90.4±0.1	84.2±0.3	90.0	**90.5±0.2**	86.8±0.4
CIFAR10	84.7±0.3	77.8±0.7	–	**84.5±0.2**	84.0±0.4
QuickDraw	85.3±0.2	85.5±0.4	–	**85.7±0.3**	85.4±0.1

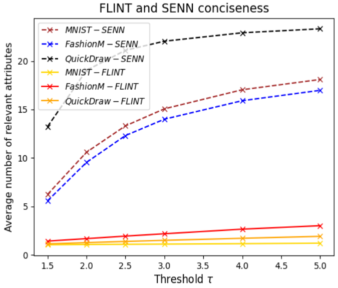
Table: Accuracy (in %) on different datasets. BASE-f is system trained with just accuracy loss. FLINT-f, FLINT-g denote the predictor and interpreter trained in our framework.

Dataset	LIME	VIBI	FLINT-g
MNIST	95.6±0.4	96.6±0.7	**98.7±0.1**
FashionMNIST	67.3±1.3	88.4±0.3	**91.5±0.1**
CIFAR-10	31.5±0.9	65.5±0.3	**93.2±0.2**
QuickDraw	76.3±0.1	78.6±0.4	**90.8±0.4**

Table: Results for fidelity to FLINT-f (in %)

Results – Quantitative II

- Evaluate conciseness by measuring the average number of *important* concepts/attributes in generated interpretations.
- Conciseness for a given sample x, $\mathrm{CNS}_{g,x}, = |\{j : |r_{j,x}| > 1/\tau\}|$.
- For different thresholds $1/\tau$, compute mean of $\mathrm{CNS}_{g,x}$ over test data



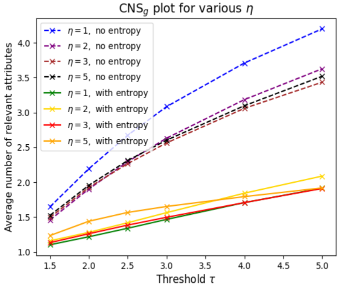
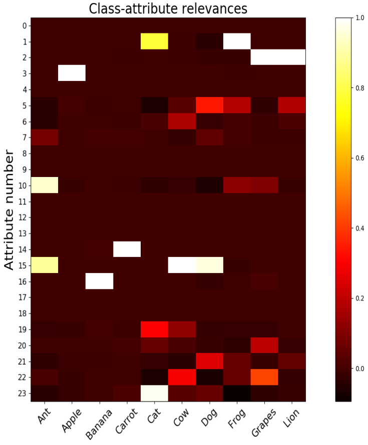
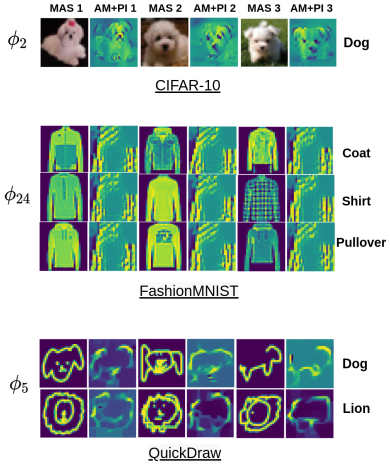


Figure: (Left) Conciseness comparison with SENN. (Right) Effect of entropy and different ℓ_1 regularization strength on conciseness on QuickDraw

Global Interpretations I



(a) Global relevances ($r_{j,c}$) for all class-attribute pairs for QuickDraw



(b) Sample class-attribute pairs with high relevance

Global Interpretations II

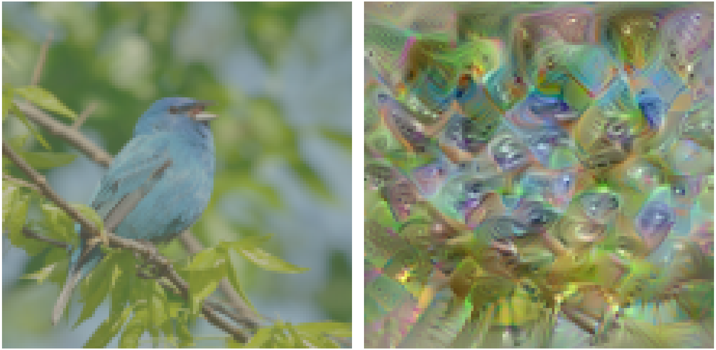


Figure: Example attribute ϕ_{120} on CUB-200, detecting blue faced birds

Global Interpretations II

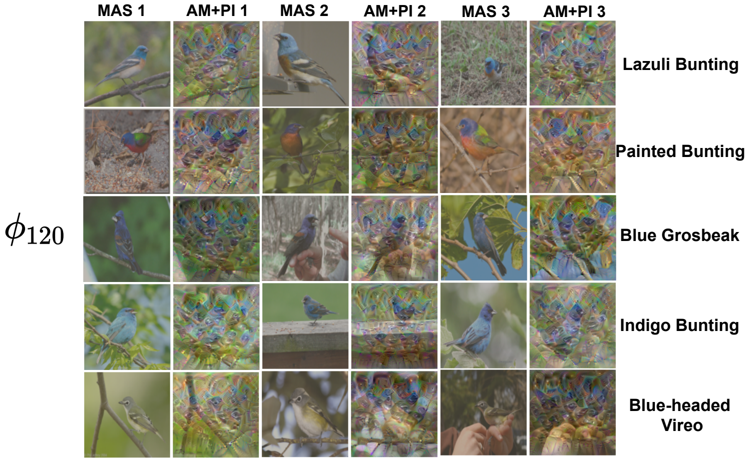


Figure: Example attribute ϕ_{120} on CUB-200, detecting blue faced birds

Local Interpretations

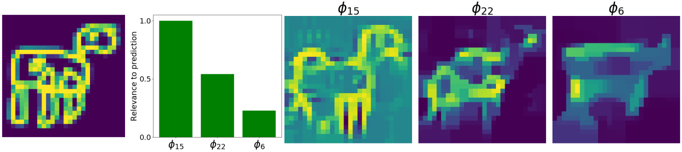


Figure: Local interpretation example. True label 'Cow'

Local Interpretations

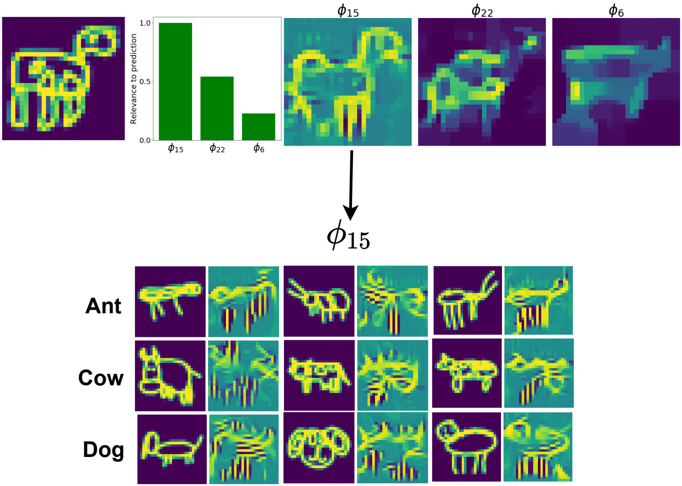


Figure: Local interpretation example. True label 'Cow'

Local Interpretations

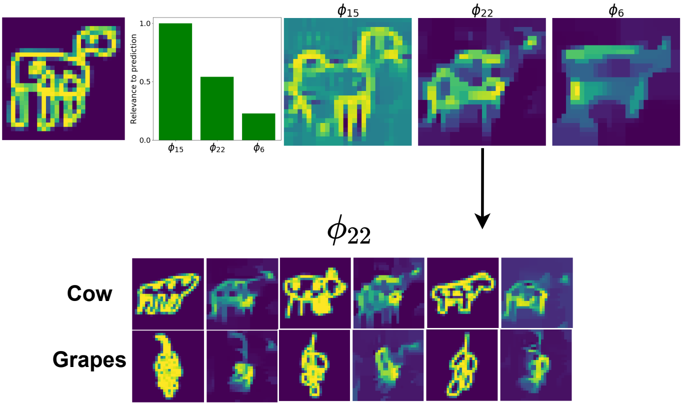


Figure: Local interpretation example. True label 'Cow'

Local Interpretations

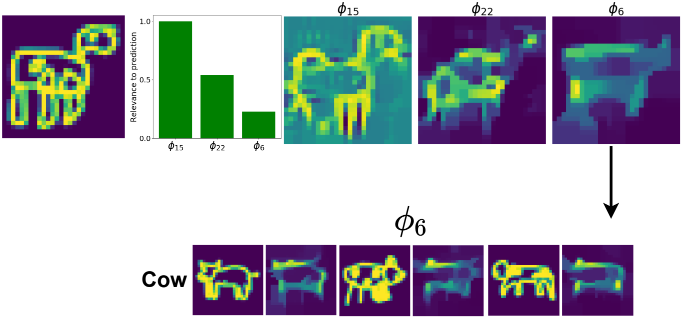


Figure: Local interpretation example. True label 'Cow'

Meaningfulness of Learnt Attributes

- We further conducted **Subjective evaluation** with 20 participants to evaluate meaningfulness of attributes and their visualizations.

Meaningfulness of Learnt Attributes

- We further conducted **Subjective evaluation** with 20 participants to evaluate meaningfulness of attributes and their visualizations.
- Each participant shown visualizations of 10 attributes (covering 17 class-attribute pairs) from QuickDraw dataset and a textual description.

Meaningfulness of Learnt Attributes

- We further conducted **Subjective evaluation** with 20 participants to evaluate meaningfulness of attributes and their visualizations.

- Each participant shown visualizations of 10 attributes (covering 17 class-attribute pairs) from QuickDraw dataset and a textual description.

- For each attribute, asked to indicate their agreement/disagreement if description meaningfully associates to visualizations (Choices: Strongly Agree (SA), Agree (A), Disagree (D), Strongly Disagree (SD), Don't Know (DK)). 40% incorrect descriptions were manually added.

Meaningfulness of Learnt Attributes

- We further conducted **Subjective evaluation** with 20 participants to evaluate meaningfulness of attributes and their visualizations.
- Each participant shown visualizations of 10 attributes (covering 17 class-attribute pairs) from QuickDraw dataset and a textual description.
- For each attribute, asked to indicate their agreement/disagreement if description meaningfully associates to visualizations (Choices: Strongly Agree (SA), Agree (A), Disagree (D), Strongly Disagree (SD), Don't Know (DK)). 40% incorrect descriptions were manually added.
- **Results**: For correct descriptions: 77.5% – SA/A, 10.0% – DK, 12.5% – D/SD. For incorrect descriptions: 83.7% – D/SD, 7.5% – DK, 8.8% – SA/A.

Post-hoc interpretations

Interpreting the BASE-f model (trained only for accuracy).

Dataset	VIBI	FLINT-g
MNIST	95.8$\pm$0.2	**98.6$\pm$0.2**
FashionMNIST	88.4$\pm$0.2	**92.8$\pm$0.3**
CIFAR10	64.2$\pm$0.3	**89.1$\pm$0.5**
QuickDraw	78.0$\pm$0.4	**90.5$\pm$0.3**

Table: Fidelity for post-hoc interpretations of BASE-f (in %)

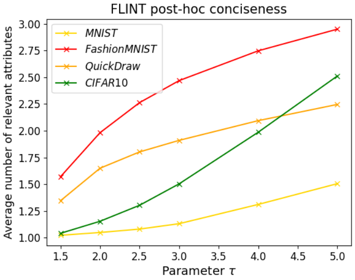


Figure: Conciseness plots for post-hoc interpretations

Perspectives I

Summary: FLINT is a novel framework to jointly learn predictor and interpreter network. The interpreter provides local and global interpretability in terms of high-level attributes.
Different usages of FLINT:

- **by-design**: Learning a pair (predictor, interpreter) of networks, provide global interpretation of classes, provide local interpretation when predictor and interpreter agree
- **Post-hoc**: interpret a known network

Promising use of FLINT Retaining only the interpreter as the final prediction model: fully-faithful and reduced complexity.
Important: the so-called prediction network is useful to provide proper data representation.

Perspectives II

Future Directions

- **Additional constraints**: To enforce properties on attributes such as stability, adversarial robustness, invariance to transformations etc.
- **Faithfulness** of g to f – Studying its evaluation, enforcement.
- **Evaluation strategies**: To compare between methods using different means of explanations.

Perspectives III

There is also a possibility to apply/modify the framework for application to other input modalities, models or tasks

- **Input modality**: Eg. **audio**, *video, text, graphs*.
- **Models/Tasks**: graph-CNNs, structured prediction energy networks (SPEN) or more generally tasks like regression, structured prediction, reinforcement learning etc.
- The key modification here is to redesign method to generate interpretations: That is designing high-level units of interpretation suitable to the task, revising constraints and method to understand them accordingly.

The End

THANK YOU!

Most of the presentation based on A Framework to Learn with Interpretation. arXiv preprint arXiv:2010.09345 (Presented at NeurIPS 2021)

References

1. Lee et al. Functional trans-parency for structured data: a game-theoretic approach. In ICML 2019
2. Alvarez-Melis & Jaakkola Towards robust interpretability with self-explaining neural networks. In NeurIPS 2018
3. Jain et al. Subic: A supervised, structured binary code for image search/ In ICCV 2017
4. Li et al. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In AAAI 2018
5. Ribeiro et al. Why should i trust you?: Explaining the predictions of any classifier. In ACM SIGKDD 2016
6. Bang et al. Explaining a black-box using deep variationalinformation bottleneck approach. In AAAI 2021

Appendix 1: Effect of autoencoder

Training without autoencoder affects the attributes. The learnt attributes are more inconsistent in detected patterns. This makes it hard to understand them.

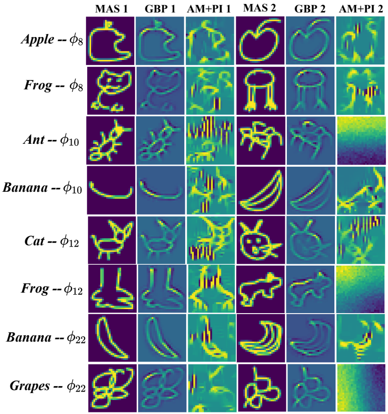


Figure: Sample attribute interpretations without any autoencoder. GBP stands for Guided Backpropagation

Appendix 2: Effect of J - Quantitative

	$\mathcal{L}_{if}$ (train)	$\mathcal{L}_{of}$ (train)	Fidelity (test) (%)
$J = 4$	0.058	0.57	87.4
$J = 8$	0.053	0.23	97.5
$J = 25$	0.029	0.16	98.8

Table: Effect of J for MNIST with LeNet.

	$\mathcal{L}_{if}$ (train)	$\mathcal{L}_{of}$ (train)	Fidelity (test) (%)
$J = 4$	0.094	2.08	19.5
$J = 8$	0.079	1.48	57.6
$J = 24$	0.069	0.34	90.8

Table: Effect of J for QuickDraw with ResNet.

Appendix 3: Effect of J - Qualitative

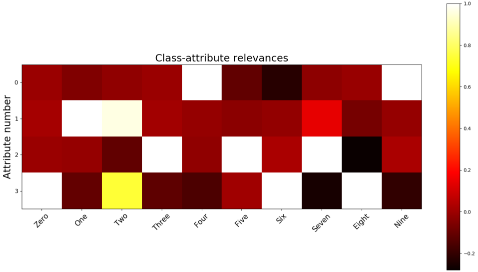


Figure: Global class attribute relevances for model with $J = 4$ on MNIST.

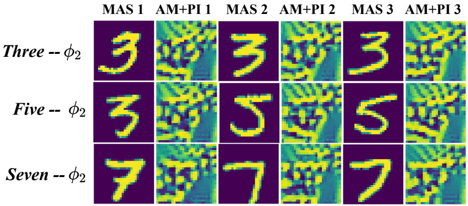


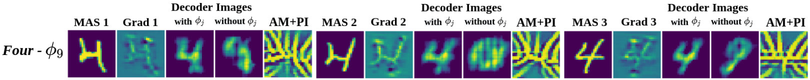
Figure: Interpretation for attribute ϕ_2 for model learn on MNIST with $J = 4$.

Appendix 4: Relevant class-attribute pairs

- For a sample x with predicted class $\hat{c}$ (by the interpreter), we define the total contribution of attribute j as $\alpha_{j,\hat{c},x} = \phi_j(x^{\mathcal{I}}).w_{j,\hat{c}}$, where $w_{j,\hat{c}}$ are weights of linear classifier h.

- The importance of attribute j, for predicting class $\hat{c}$, for sample x is, $r_{j,\hat{c},x} = \frac{\alpha_{j,\hat{c},x}}{max_i|\alpha_{i,\hat{c},x}|}$. To estimate $r_{j,c}$, compute mean of $r_{j,\hat{c},x}$ for samples x where predicted class $\hat{c} = c$. That is, $r_{j,c} = \sum_{\{x \in \mathcal{S}_{rnd} | \hat{c}=c\}} r_{j,\hat{c},x}$ ($\mathcal{S}_{rnd}$ is random subset of the training set).

- To select relevant class-attribute pairs, we simply threshold $r_{j,c}$ for each (j, c). For each such selected pair we analyze the attribute's maximum activating samples (MAS) from the class.



Appendix 5: How to use other tools

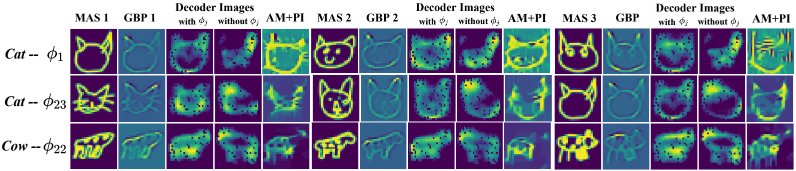


Figure: Examples of class-attribute pairs for decoder assistance

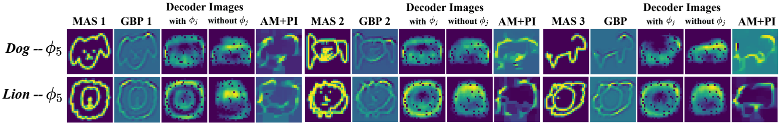


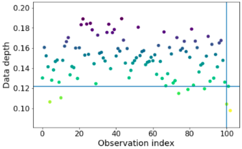
Figure: Examples of class-attribute pairs for input attribution assistance

Appendix 6: Disagreement analysis

What if the predictor and interpreter disagree in their outputs?

- if the class predicted by f is among the top predicted classes of g, the disagreement is acceptable to some extent as the attributes can still potentially interpret the prediction of f.
- The worse kind of samples – where prediction of f is not among top predictions of g, and even worse are where, in addition to this, f predicts the true label.
- Measure top-k fidelity. For QuickDraw: top-2 – 94.7%, top-3 – 96.9%, and top-4 – 98.2%

Figure 13: The three 'Apple' class samples classified correctly by f but not by g.



Appendix 7: Importance of Attributes

- To test how crucial the learnt attributes are to predictions of FLINT-g and SENN, we shuffle the attribute values $\Phi(x)$ for each sample x and calculate the drop in prediction accuracy.
- Extreme test, therefore a significant drop in accuracy is expected

Dataset	SENN	FLINT-g
MNIST	0.5	87.6
FashionMNIST	10.9	76.6
CIFAR-10	17.5	74.4
QuickDraw	0.3	74.9

Table: FLINT and SENN accuracy drop for shuffled attributes (in %)