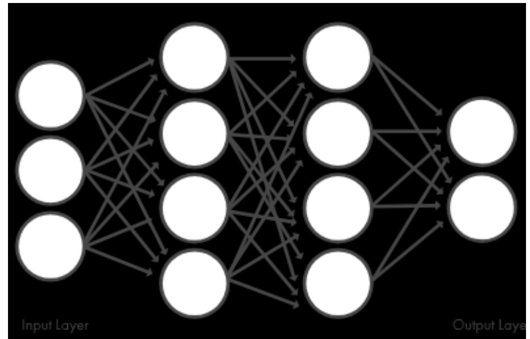
Layered Explanations:
Interpreting Neural Networks with Numerical Influence Measures

Ho Xuan Vinh

NUS National University of Singapore | School of Computing

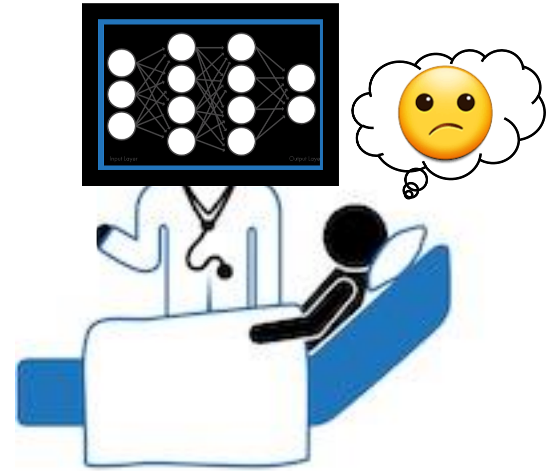
Algorithmic Transparency: Motivation



Neural Network
Deep Learning

- Manufactoring quality control
- Market trading
- Determining loan eligibility
- Disease diagnosis
- Traffic control
- Streaming and course allocation
- Credit Scores
- News & Social Media

Algorithmic Transparency: Motivation



Trustworthy

- Doctor-patient relationship
- Doctor is a specialist
- Doctor has a personal stake
- Reinforced reliability through time

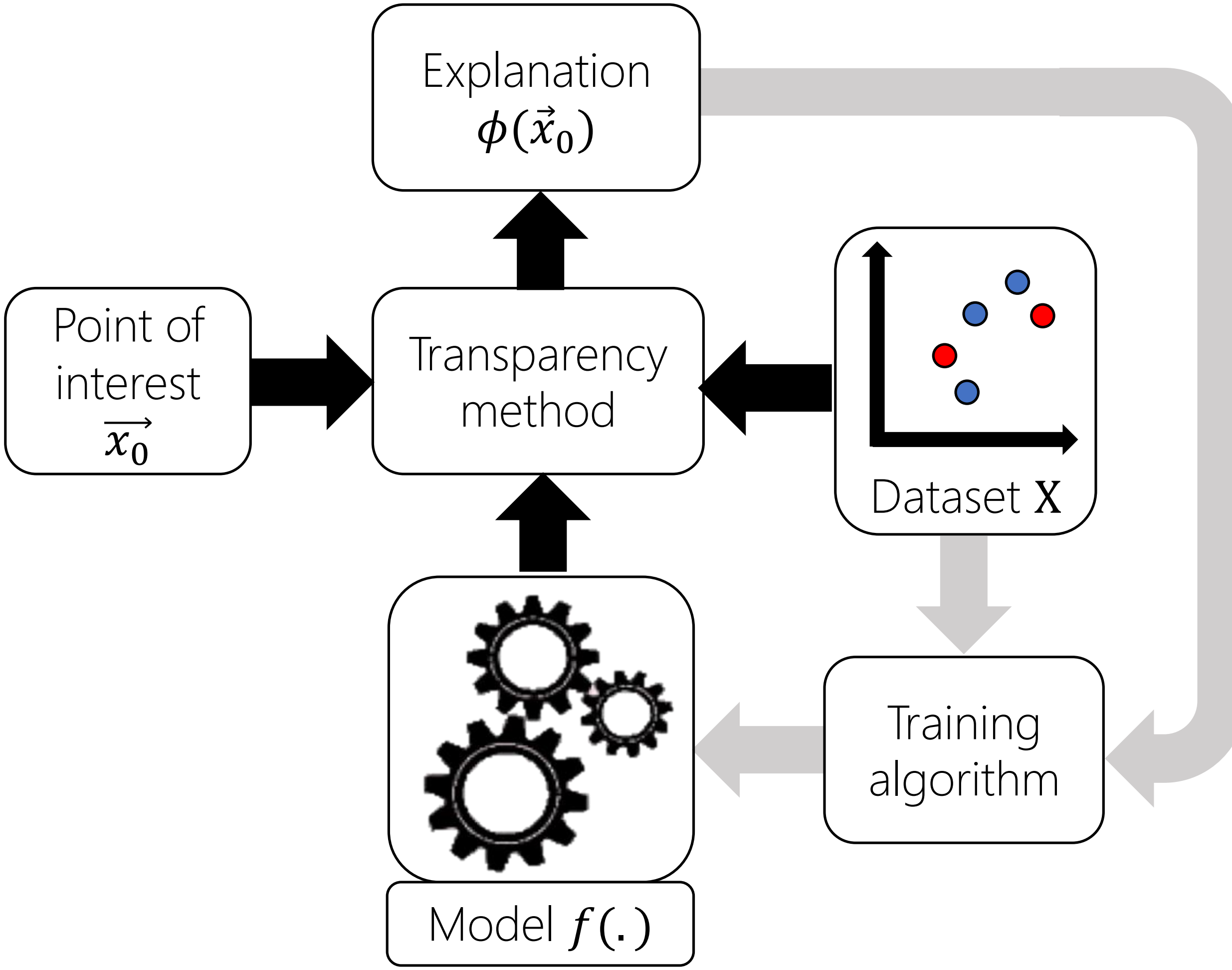
Untrustworthy

- How knowledgeable is NN?
- NN faces no consequence
- Unreliable historical records

Algorithmic Transparency: Interpretable Explanation

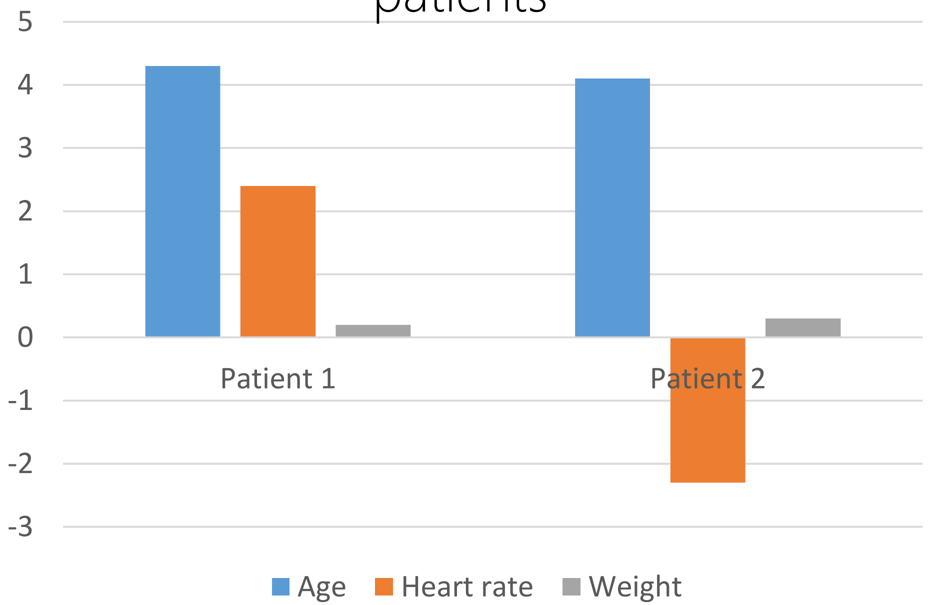
Given $\vec{x} \in \mathbb{R}^n$, why was $\vec{x}$ labeled $f(\vec{x})$?

- $\phi(\vec{x}) \in \mathbb{R}^n$ - an explanation. May use the dataset $\mathbf{X}$ and additional domain knowledge.
- *Linear explanation* is mainly focused



Algorithmic Transparency: Interpretable Explanation

Transparency report $\phi(\vec{x})$ of released patients



Challenges

- Features often do not have intrinsic meaning
- Individual measured effect is smalll

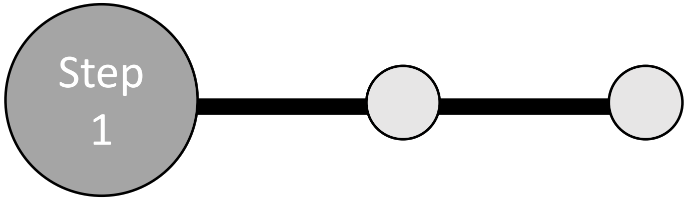
Our approach

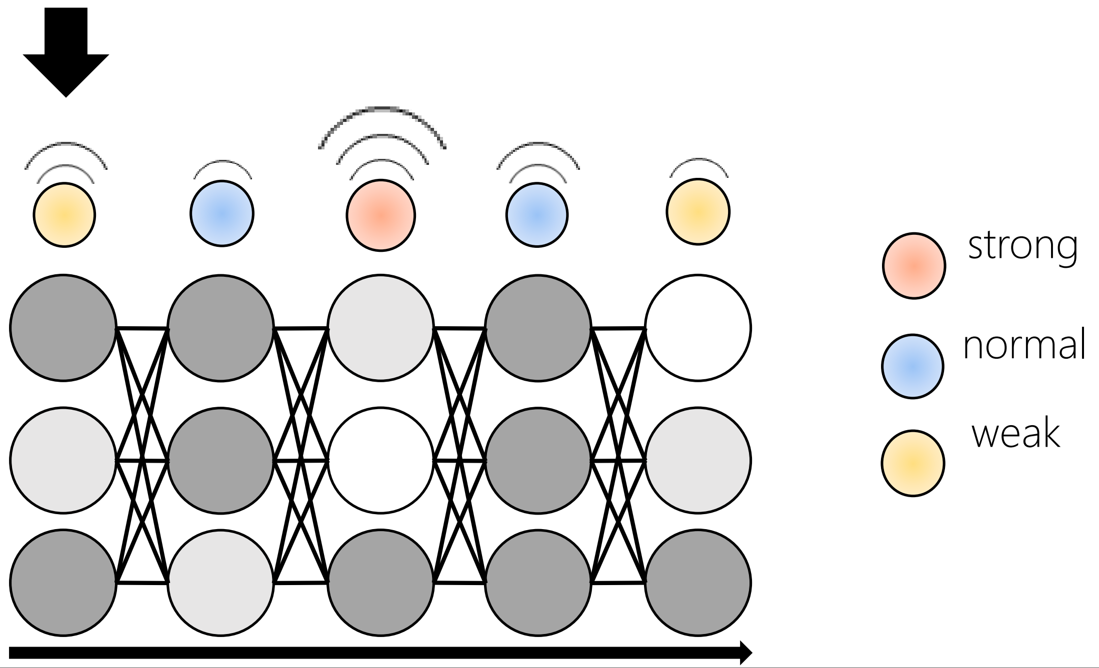
- Measure influence in Neural Network's internal layers

Layered Explanations Framework

Layered Explanations Framework

Provide explanation of learning model w.r.t. an instance

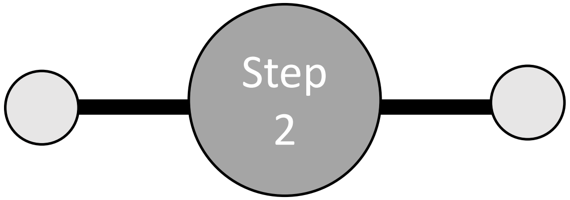


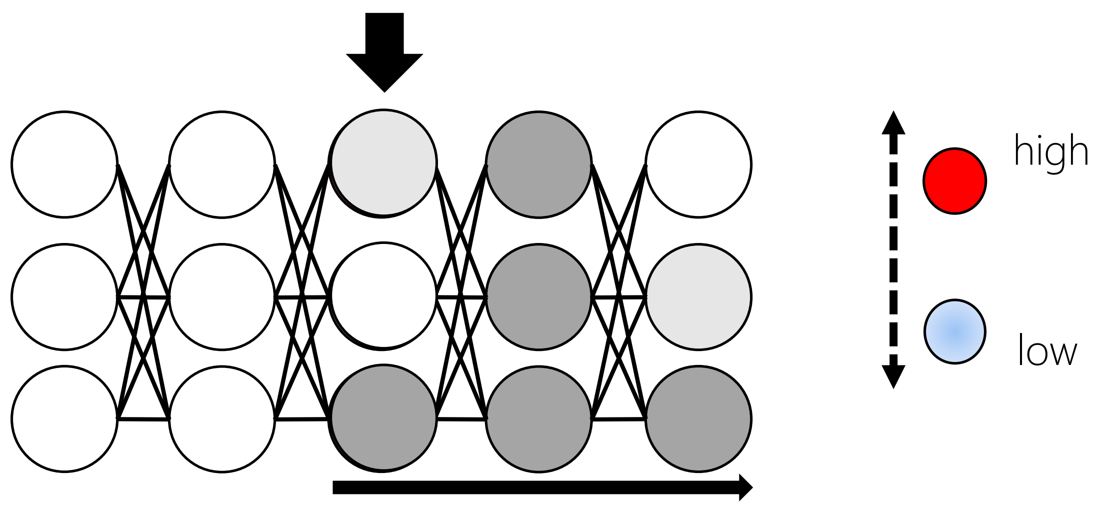


Automatically identify influential layer

Layered Explanations Framework

Provide explanation of learning model w.r.t. an instance

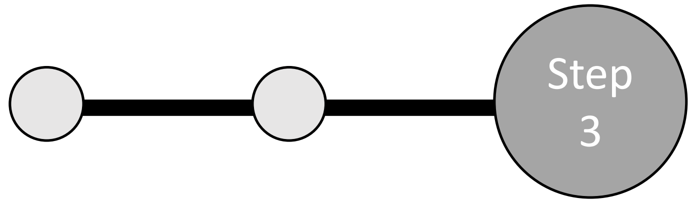


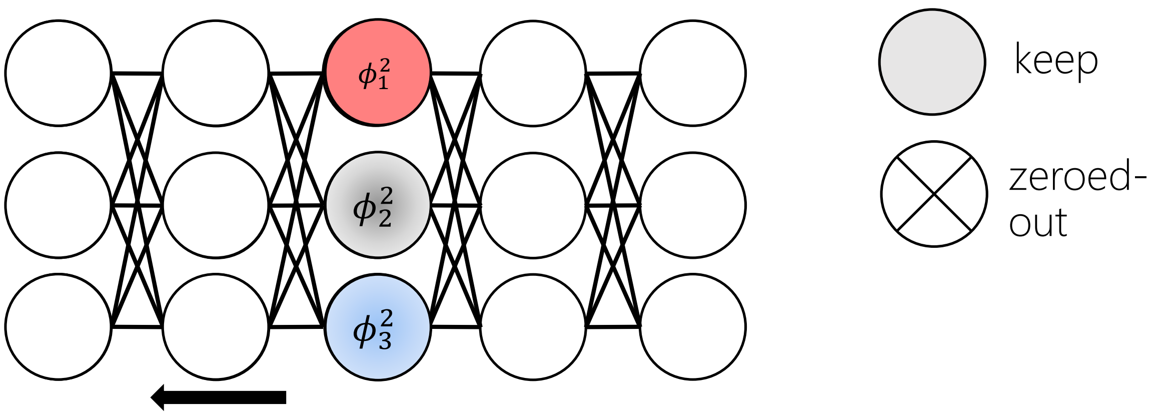


Apply Numerical influence measures

Layered Explanations Framework

Provide explanation of learning model w.r.t. an instance

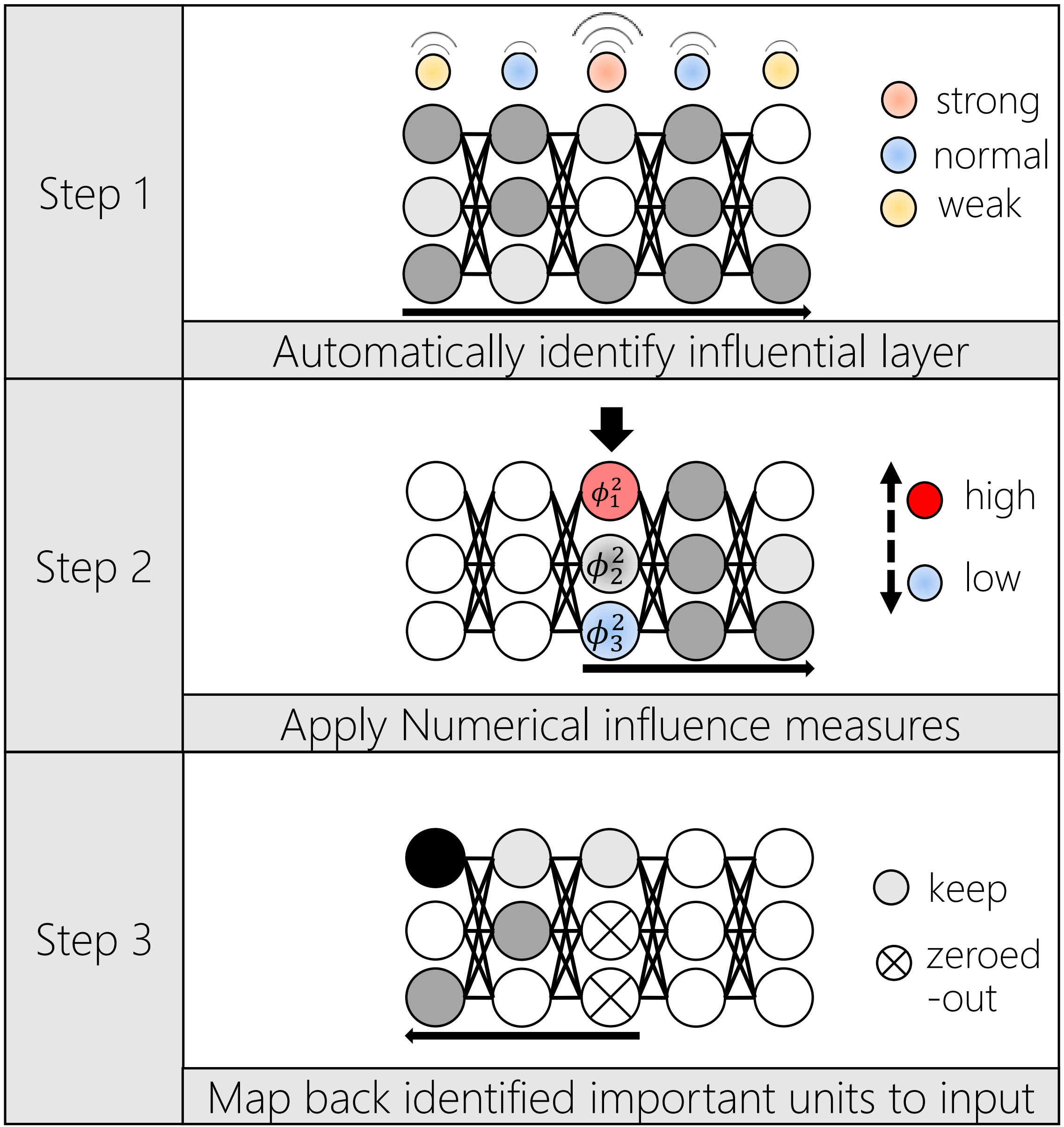




Map back identified important units to input

Layered Explanations Framework

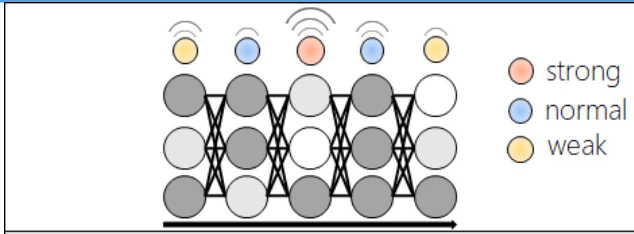
Provide explanation of learning model w.r.t. an instance



Step 1 – Identify Influential Layer

Motivation:

- Potentially capture high-level feature
- Receptive field are bigger



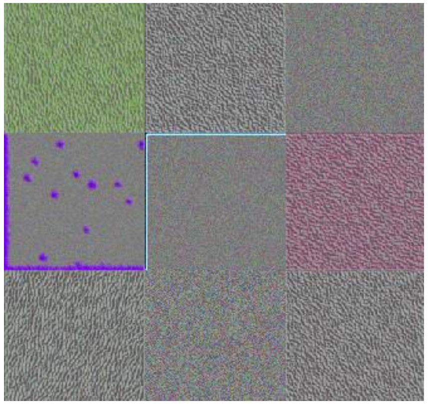
Automatically identify influential layer

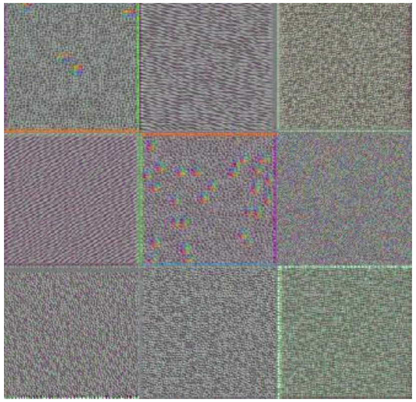
Input:

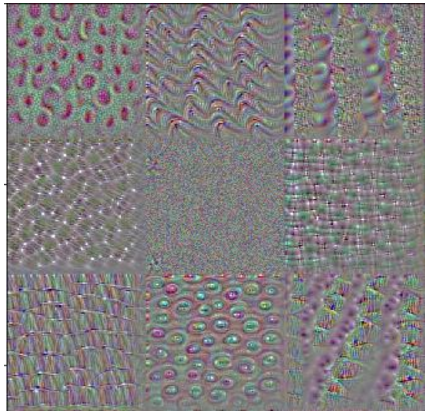
- Model f

Output

- Chosen layer L_k





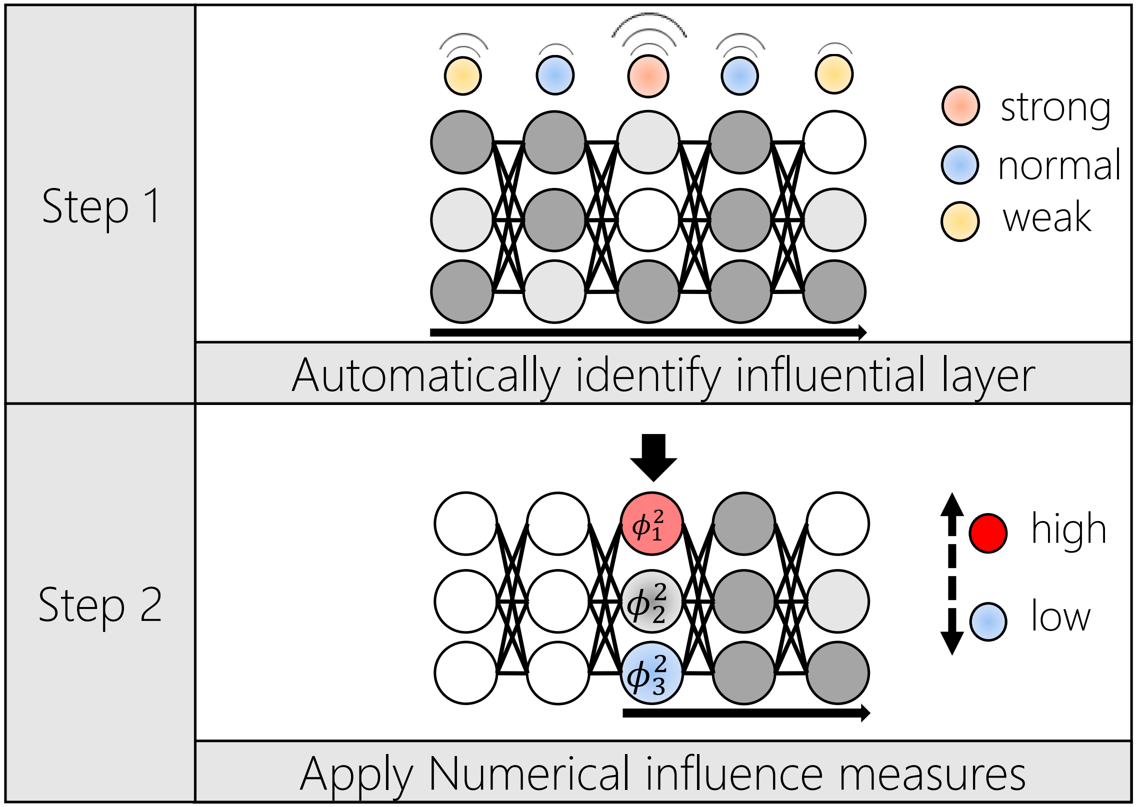


Method:

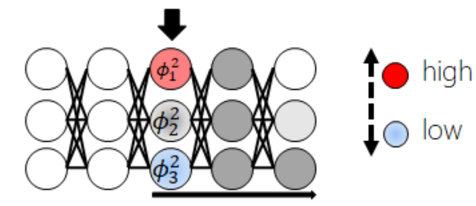
- Influence measures
- Linear probe
  - Most robust layer
  - Most discriminative layer

Step 2 – Numerical Influence Measure

Provides explanation of learning model w.r.t. an instance



Step 2 – Numerical Influence Measure



Apply Numerical influence measures

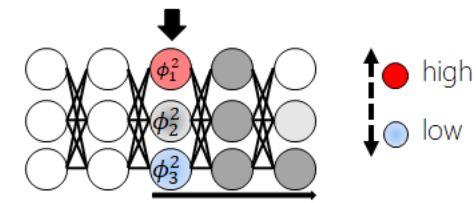
Given $\vec{x}$, why was $\vec{x}$ labeled $f(\vec{x})$?

- $\phi(\vec{x})$ - an explanation. May use the dataset $\mathbf{X}$ and additional domain knowledge.

- *This talk*: numerical explanations
  - $\phi(x) \in \mathbb{R}^n$, where $\phi_i(\vec{x})$ is how important was feature i in determining $f(\vec{x})$?
  - MIM, QII, LIME, Parzen, DeepLIFT…

- Our method can use any numerical influence measure. We focus on MIM
  - Computationaly efficient
  - Axiomatically justified

Step 2 – Numerical Influence Measure



Apply Numerical influence measures

Input:

- Datapoint $\vec{x} = \{x_1, x_2, \dots, x_n\}$
- Dataset X
- Model f

Output:

- Influence score $\phi^i(\vec{x})$ of each feature/unit i in $\vec{x}$

Method:

Quantitative Input Influence (QII)	Monotone Influence Measure (MIM)

Step 2 – Numerical Influence Measure

Quantitative Input Influence | Monotone Influence Measure

Intuition

The obtained influence measure $\phi_{QII}(\vec{x}, \mathrm{X})$ tells how much each feature *marginally contributes* to model's outcome $Q(\vec{x}, \mathrm{N})$ w.r.t. instance $\vec{x}$.

Marginal contribution $\phi_{QII}^{i}(\vec{x}, \mathrm{X})$

$$\phi_{QII}^{i}(\vec{x}, \mathrm{X}) = \frac{1}{n!} \underbrace{\sum_{\pi \in \Pi}}_{} [Q(\vec{x}, S_\pi(i) \cup \{i\}) - Q(\vec{x}\ S_\pi(i))]$$

Over the permutation set Π of n features

Step 2 – Numerical Influence Measure

Quantitative Input Influence | Monotone Influence Measure

Intuition

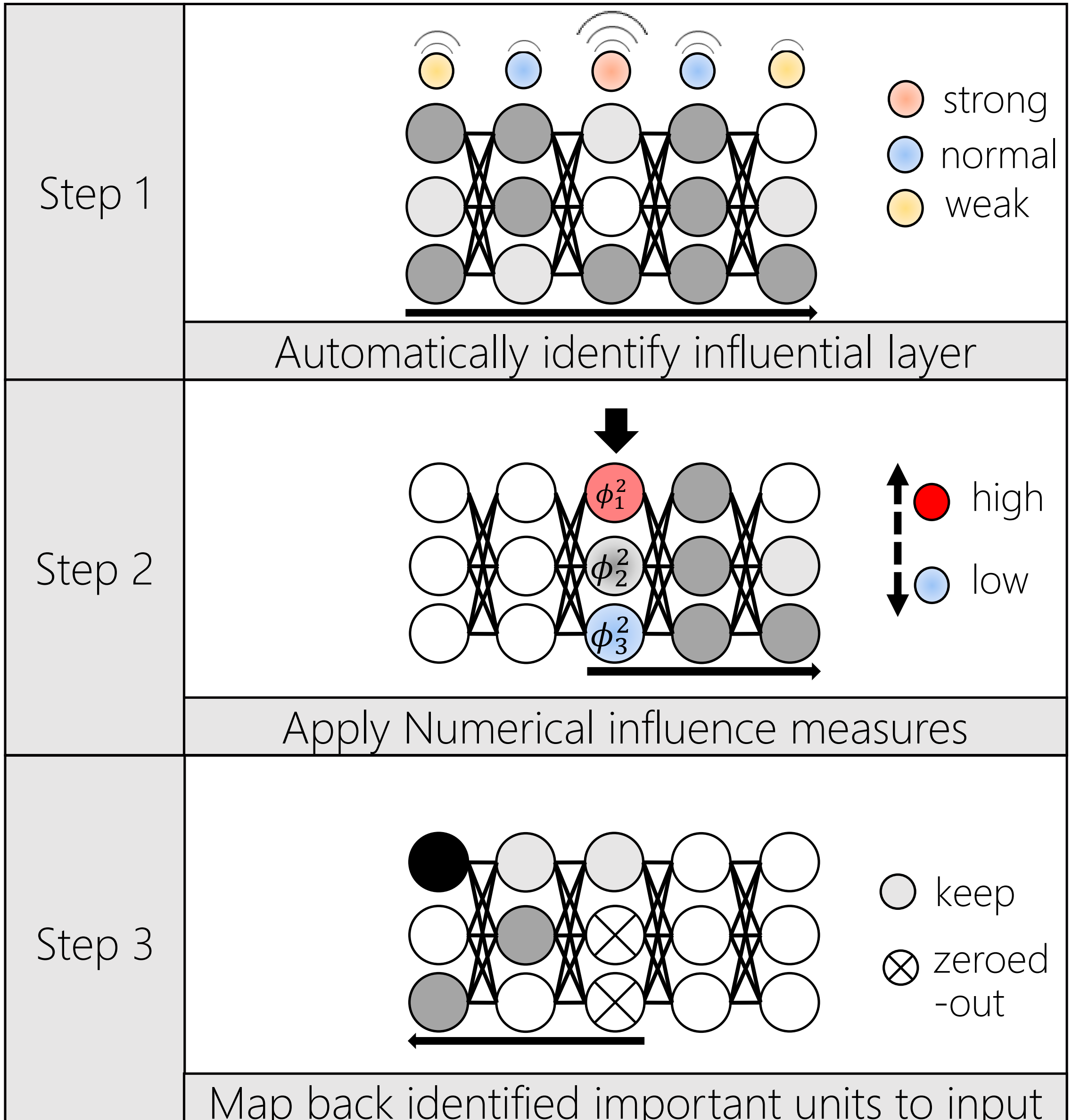
The obtained influence measure $\phi_{MIM}(\vec{x}, \mathrm{X})$ indicates global direction that *<u>strengthens</u>* the current label of instance $\vec{x}$.

Influence measure $\phi_{MIM}^{i}(\vec{x}\,, \mathrm{X})$

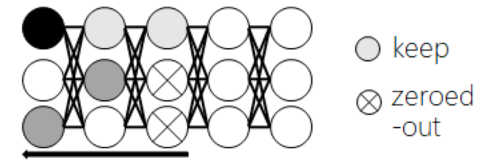
$$\phi_{MIM}^{i}(\vec{x}, \mathrm{X}) = \underbrace{\sum_{\vec{y} \in \mathrm{X} \setminus \vec{\mathrm{x}}}}_{\text{Over all data points except for } x} (\vec{y} - \vec{x}) \overbrace{\alpha(\| \vec{y}, \vec{x} \|)}^{\text{Function of Distance}} \underbrace{\mathbb{I}(f(\vec{x}) = f(\vec{y}))}_{\{-1,1\}\text{-valued Indicator function}}$$

Step 3 - Map back identified important units to Input

Provides explanation of learning model w.r.t. an instance



Step 3 - Map back identified important units to input



Map back identified important units to input

Input:

- Datapoint $\vec{x}$ and its value in each layer L_i, $\forall i \in \{1, \ldots, K\}$
- Influence scores at chosen layer L_k
- Neural network f

Output:

- Back-propagated value at input space L_1

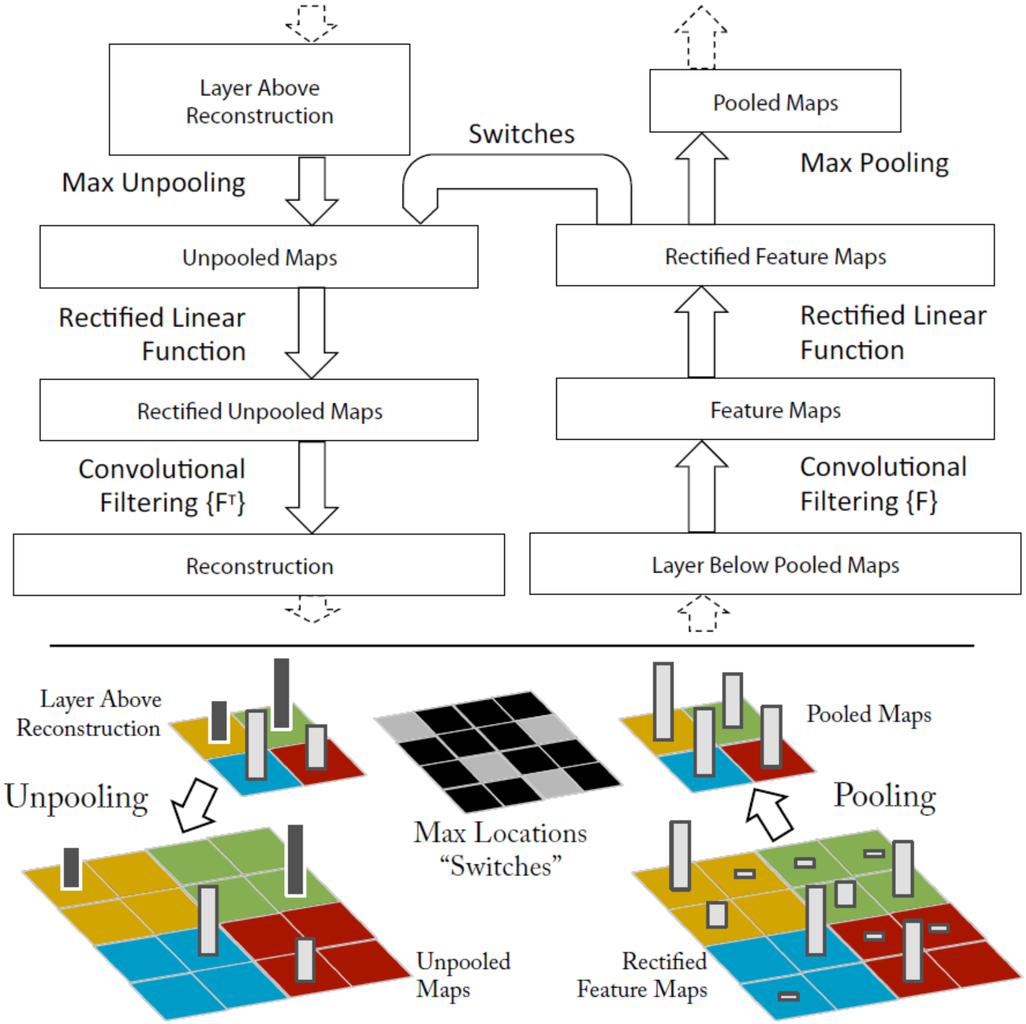
Method:

Deconvolutional Neural Net (DeconvNet) | Guided Backpropagation (GuidedBackprop)

Step 3 – Backpropagation methods

Deconvolutional Neural Net

Guided Backpropagation



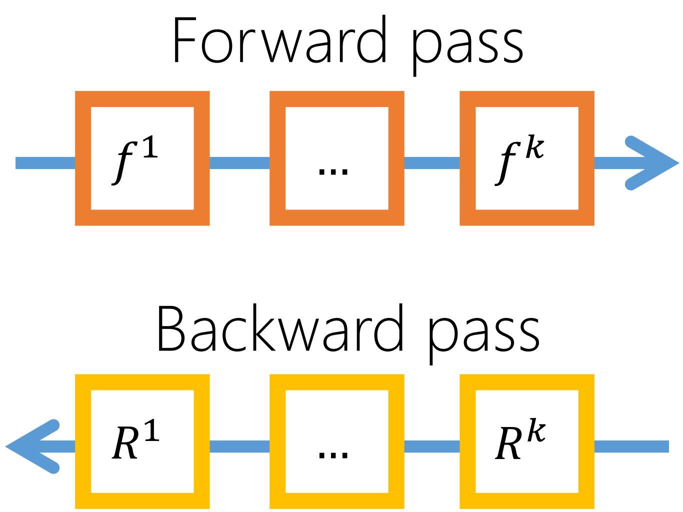
- Unpooling: switch
- Deconvolutional layer
- ReLU in backward phase

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In European conference on computer vision, pages 818–833. Springer.

Step 3 – Backpropagation methods

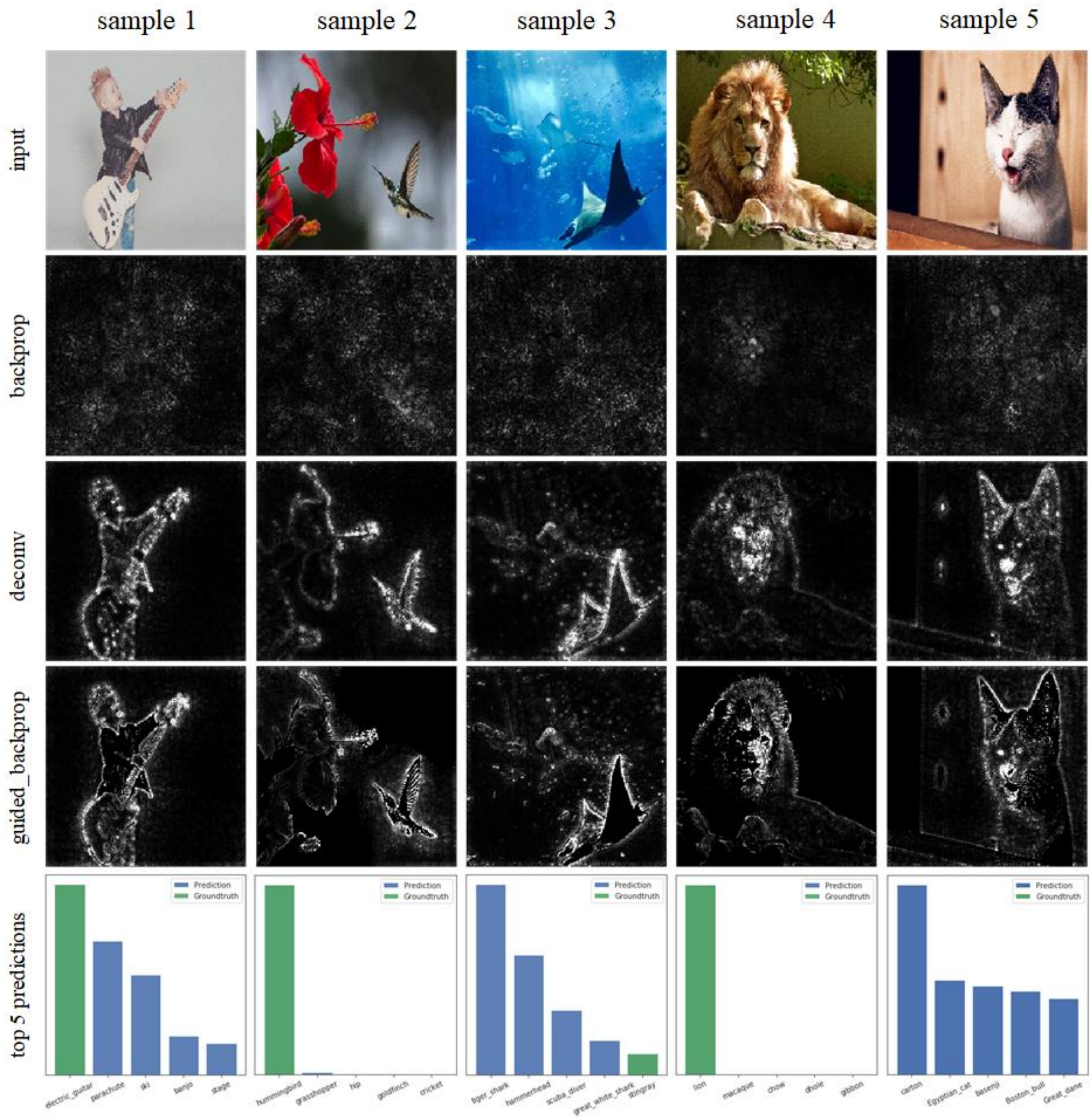
Deconvolutional Neural Net

Guided Backpropagation



$$\mathrm{R}^i = (f^i > 0).(R^{i+1} > 0).R^{i+1}$$

where $R^{i+1} = \frac{\partial f^{out}}{\partial f^{i+1}}$


sample 1
sample 2
sample 3
sample 4
sample 5
input
backprop
deconv
guided_backprop
top 5 predictions
Prediction
Groundtruth
electric_guitar
parachute
ski
banjo
stage
hummingbird
grasshopper
hip
goldfinch
cricket
tiger_shark
hammerhead
scuba_diver
great_white_shark
stingray
lion
macaque
chow
dhole
gibbon
carton
Egyptian_cat
basenji
Boston_bull
Great_dane

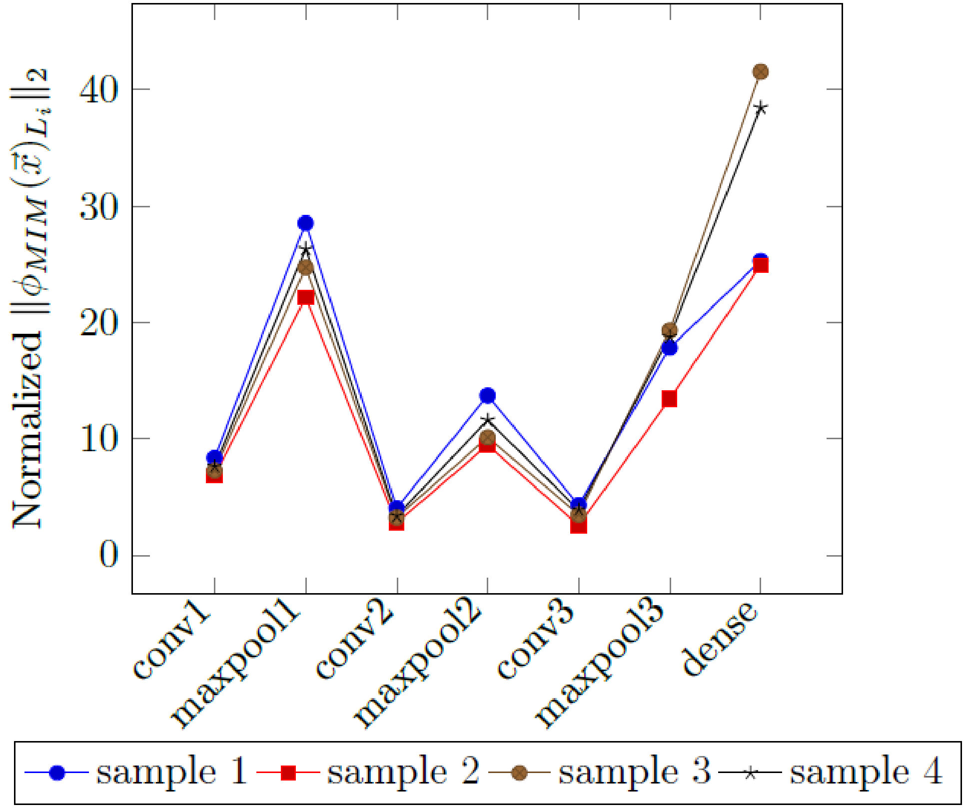
Experiment

- Dataset & classifier:
  - MNIST (digit pair 6-9) + 3-convolutional-layer CNN
  - Dog-Fish (extracted from ImageNet) + VGG16 net

- Comparison methods:
  - MIM on input layer
  - Guided-Activation:
    - Masking $M = \mathbb{I}\,(|\vec{x}_{L_i}| \geq \delta)$
  - Guided-MIM: MIM + GuidedBackprop
    - Masking $M = \mathbb{I}\,(|\phi_{MIM}(\vec{x}_{L_i})| \geq \delta)$

- Parameters:
  - We set δ such that only top 1% influential hidden units are chosen.

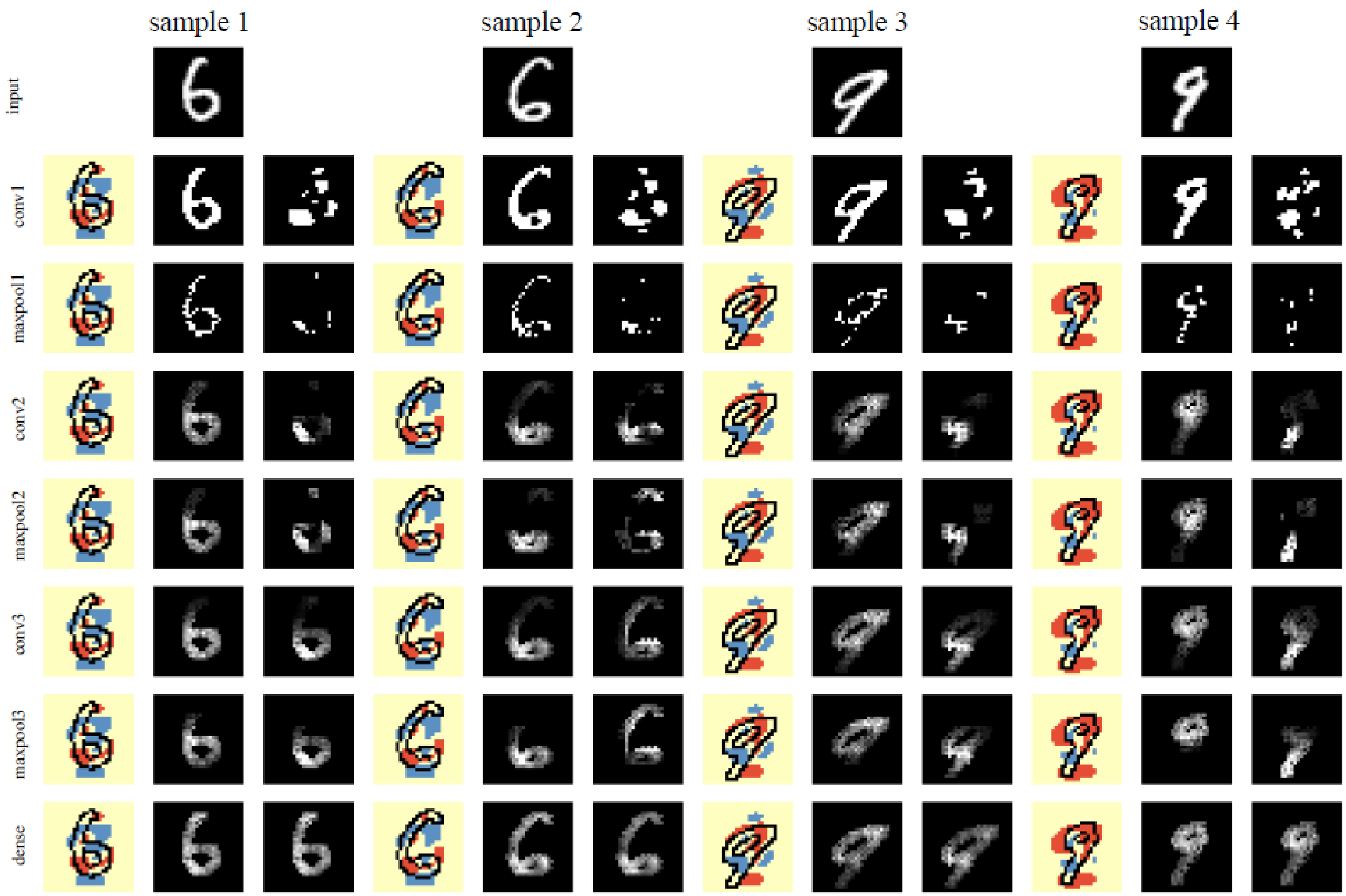
Experiment: MNIST



Normalized value of $\left\|\phi_{MIM}^{i}(\vec{x}, \mathrm{X})\right\|$ of four samples.

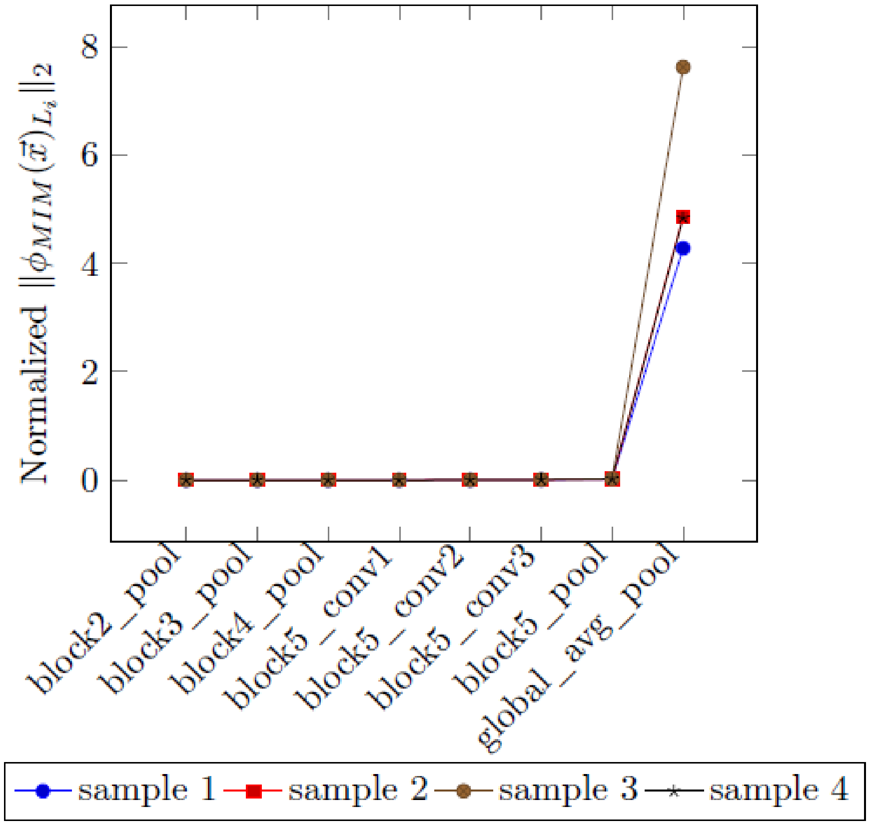
$$L^* = \underset{L_i \in \{L_2, \ldots, L_{K-1}\}}{\operatorname{argmax}} \frac{\left\|\phi_{MIM}^{i}(\vec{x}, \mathrm{X})\right\|_2}{\max_{\vec{y} \in \mathrm{X} \setminus \vec{\mathrm{x}}} \left\| \vec{y}_{L_i} - \vec{x}_{L_i} \right\|_2}$$

Experiment: MNIST



Heatmaps of three methods: MIM, Guided-Activation, and Guided-MIM on four samples of the MNIST dataset, only 1% of total units in each layer backpropagated.

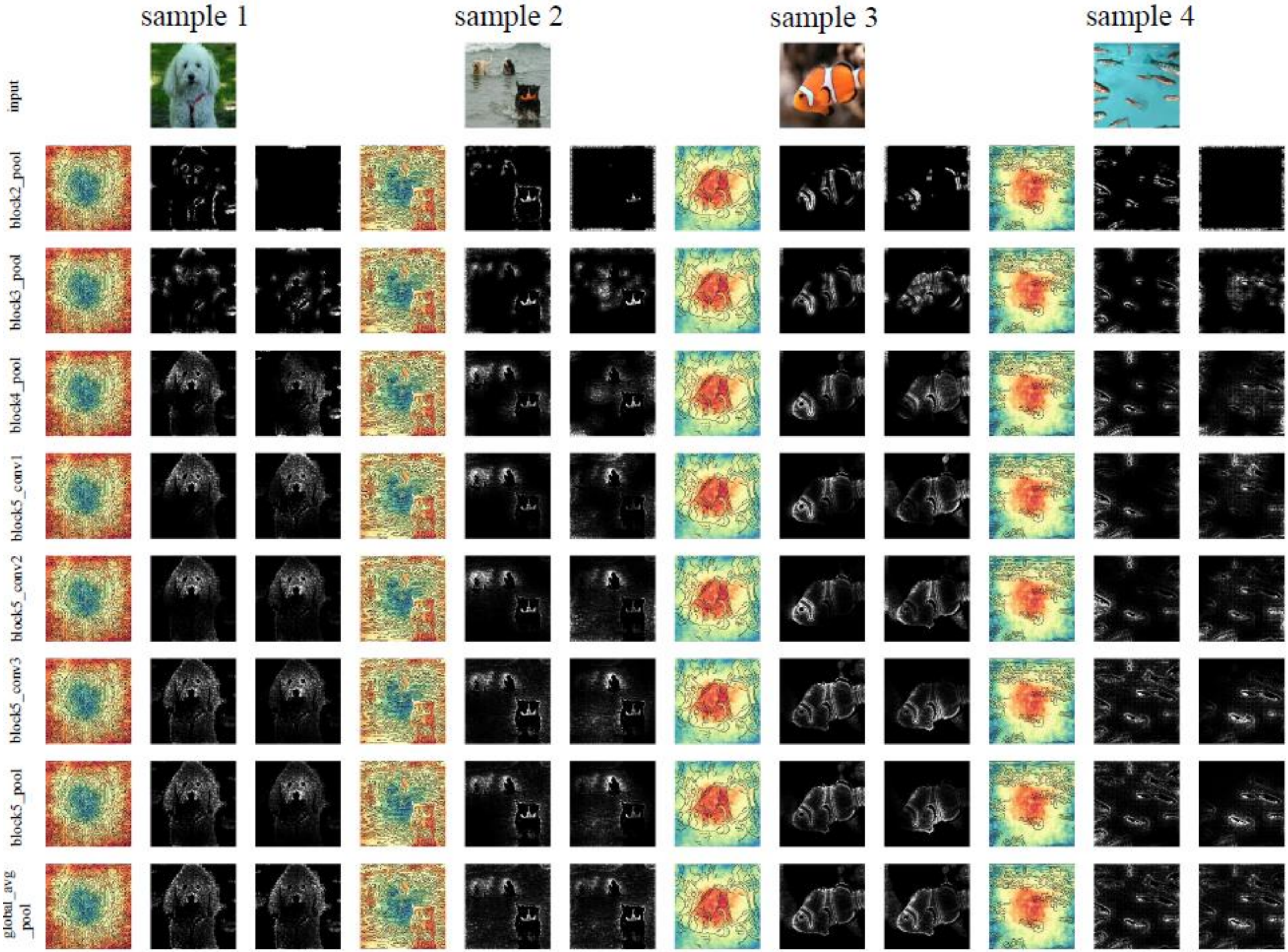
Experiment: Dog-fish



Normalized value of $\left\|\phi_{MIM}^{i}(\vec{x},\mathrm{X})\right\|$ of four samples.

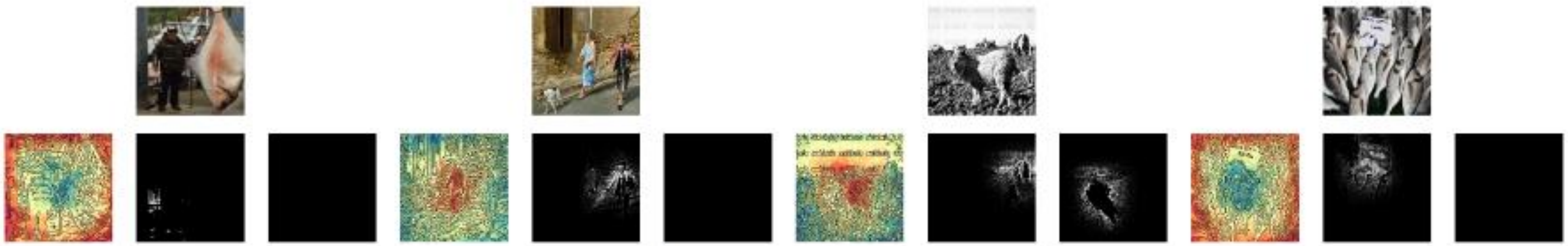
$$L^* = \operatorname*{argmax}_{L_i \in \{L_2,\dots,L_{K-1}\}} \frac{\left\|\phi_{MIM}^{i}(\vec{x},\mathrm{X})\right\|_2}{\max_{\vec{y}\in \mathrm{X}\setminus\vec{\mathrm{x}}} \left\| \vec{y}_{L_i} - \vec{x}_{L_i}\right\|_2}$$

Experiment: Dog-fish

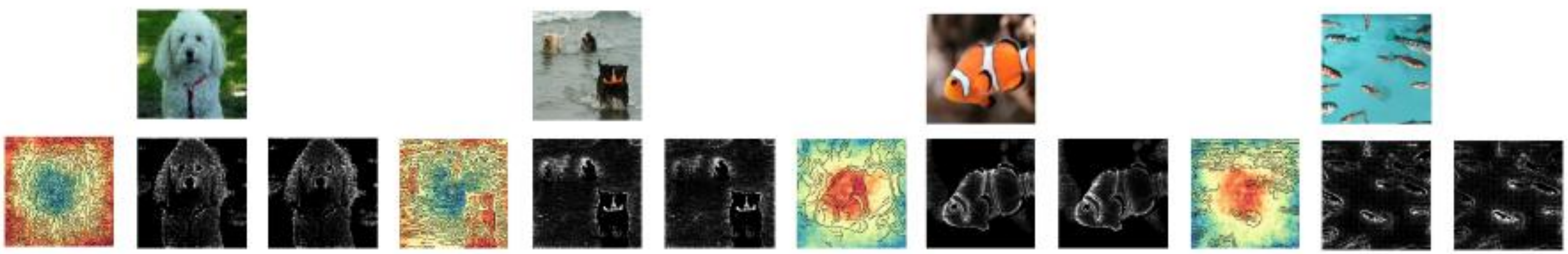


Experiment: Dog-fish

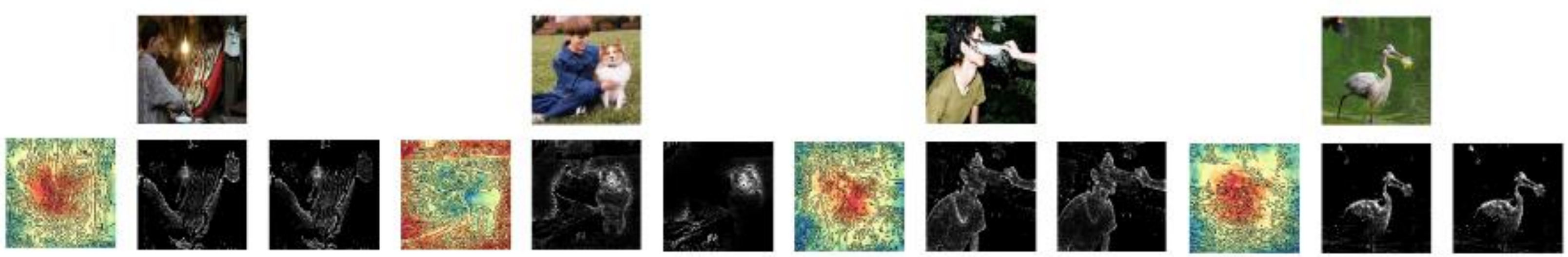
a) Low predicted probability



b) Incorrect prediction



c) Clear background



d) Noisy background

Experiment: Incremental examples

Dataset	Label t	$f(\vec{x}) = t$	$f(\vec{x} + \phi_{MIM}(\vec{x})) = t$	$f_P(\vec{x} + \phi_{MIM}(\vec{x})) > f_P(\vec{x})$
MNIST	Digit 6	1008	1008	262
	Digit 9	959	959	875
Dog-Fish	Dog	298	298	138
	Fish	302	302	0

Number of samples increases their predicted probabilities $f_P(.)$ by shifting toward the vector $\phi_{MIM}(\vec{x})$ on MNIST and Dog-fish dataset.

Discussion

- Guided-MIM is a preliminary step toward a better model.
  - Guided-Activation and Guided-MIM can outline the full shape of target object.
- Possible extensions:
  - Step 1: A NN slice marks transition from general to specific-class feature.
  - Step 2: Other numerical influence methods.
  - Step 3: Backpropate influence scores and redistribute to input layer.
  - Combination of different influence measure – backpropagation pair requires different interpretation.
  - Quantitative evaluation
- Assumption
  - pretrained model has acceptable accuracy

Conclusion

- Proposes Layered Explanations Framework
  1. Identify the greatest-explanatory layer and
  2. Influential units using numerical influence measures, then we
  3. Reconstruct relevant input regions responsible for activating these influential units.
- Experiment on MNIST & Dog-fish dataset:
  - combination of MIM and GuidedBackprop
  - able to outline target object, but no clear signal of identifying influential components.
- Possible extensions
  - applicable methods on step 2 and 3
  - possible intepretation of selected method pairs
  - quantitative evaluation

References

- **DeconvNet**: Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In European conference on computer vision, pages 818–833. Springer.
- **Guided Backpropagation**: Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806
- **QII**: Datta, A., Sen, S., and Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In Security and Privacy (SP), 2016 IEEE Symposium on, pages 598–617. IEEE.
- **MIM**: Sliwinski, J., Strobel, M., & Zick, Y. (2017). A Characterization of Monotone Influence Measures for Data Classification. arXiv preprint arXiv:1708.02153.