



MODEL UNDERSTANDING FOR PYTORCH AND RECOMMENDER SYSTEMS

NARINE KOKHLIKYAN

07.24.2020

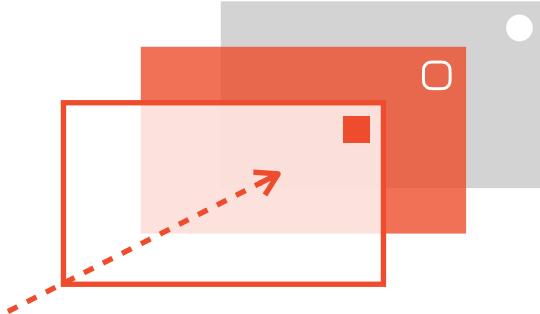
~TEAM CAPTUM~

KDD 2020

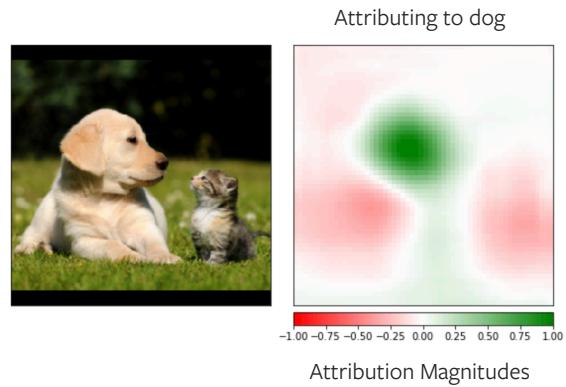


MODEL INTERPRETABILITY

INCREASED
TRANSPARENCY



BETTER
UNDERSTANDING



DEBUGGING

CAPTUM INSIGHTS

Instance Attribution Direct Target Export

Filter by Classes Filter by Instances Integrated Gradients

Animal and 2 other classes are selected: [Edit](#) Prediction: [Incorrect](#) Approximation steps: [50](#) [Fetch](#)

Predicted	Label	Contribution	Movie Reviews (Text)
Positive (0.869)	Negative	Movie Reviews	this picture was released in may of 1979 starring playboy playmate susan cunk as honey shayne , playboy playmate lisa lorden as chara and playboy playmate pamela jean brooks as terry lynn . in one of the most delicious sex comedies in drive - in history a bevy of bouncing young lovelies all come together in a tale of battling slunks sorority sisters who will stop at nothing to bare everything . so what does cunk really stand for ? you 're going to have to watch the movie to get the answer . you see , there are two other problems with this movie . you can tell that that make them out to be nothing but sex crazed bimbos . therefore , the girls set out to discredit the society girls no matter what they have to do to get the job done . in addition , cut up in this text is the dean of the college who wants to dismantle the group of girls before they grow out of control . i loved this movie especially lisa lorden . i thought her acting was fantastic and i 'm disappointed that she did n't get other acting jobs . based on the three playmates alone i give this movie 10 weasel stars .

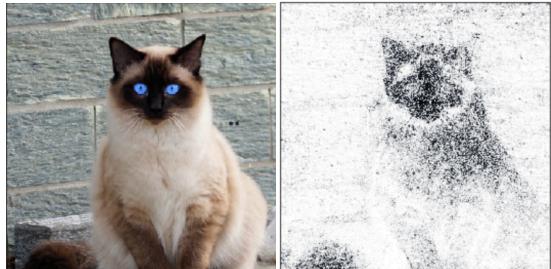


HOW CAN WE MAKE
INTERPRETABILITY ALGORITHMS ACCESSIBLE
TO ALL PYTORCH MODEL DEVELOPERS?



MODEL INTERPRETABILITY LIBRARY FOR PYTORCH

MULTIMODAL



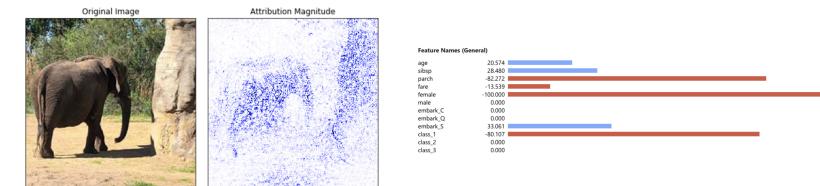
What color are the cats eyes?
Predicted
Blue (0.517)

EXTENSIBLE

```
class MyAttribution(Attribution):  
  
    def attribute(self, input, ...):  
        attributions = self._compute_attrs(input, ... )  
        # <Add any logic necessary for attribution>  
        return attributions
```

EASY TO USE

visualize_image_attr(attr.algo.attribute(input), ...)



this movie is awful . just awful . someone bought it for me as a christmas present because they knew i liked a good horror flick . i do n't think they understood the "good" part . all i can say is next year this person is getting slipper socks from me . avoid this movie - it makes you bitter . peace.



WHAT DOES THE CAPTUM LIBRARY OFFER?

Attribution algorithms to interpret:

- + Output predictions with respect to inputs
- + Output predictions with respect to all neurons in the layers
- + Neurons with respect to inputs

Currently we supports gradient and perturbation based algorithms



ATTRIBUTION ALGORITHMS

Attribute model output (or internal neurons) to input features

SHAP Methods

GradientSHAP

DeepLiftSHAP

DeepLift

Input * Gradient

GuidedGradCam

Integrated Gradients

Saliency

Occlusion

Shapely Value Sampling

FeatureAblation /
FeaturePermutation

GuidedBackprop /
Deconvolution

Attribute model output to the layers of the model

SHAP Methods

LayerGradientSHAP

LayerDeepLiftSHAP

LayerDeepLift

LayerFeatureAblation

LayerIntegratedGradients

InternalInfluence

GradCam

LayerActivation

LayerGradientXActivation

LayerConductance

NoiseTunnel (Smoothgrad, Vargrad, Smoothgrad Square)

- Gradient
- Perturbation
- Other



ATTRIBUTION ALGORITHMS

* Simple baseline approaches based on gradients, activations and inputs

Attribute model output (or internal neurons) to input features

SHAP Methods

GradientSHAP

DeepLiftSHAP

DeepLift

Input * Gradient

GuidedGradCam

Integrated Gradients

Saliency

Occlusion

Shapely Value Sampling

FeatureAblation /
FeaturePermutation

GuidedBackprop /
Deconvolution

Attribute model output to the layers of the model

SHAP Methods

LayerGradientSHAP

LayerDeepLiftSHAP

LayerDeepLift

LayerFeatureAblation

LayerIntegratedGradients

InternalInfluence

GradCam

LayerActivation

LayerGradientXActivation

LayerConductance

NoiseTunnel (Smoothgrad, Vargrad, Smoothgrad Square)

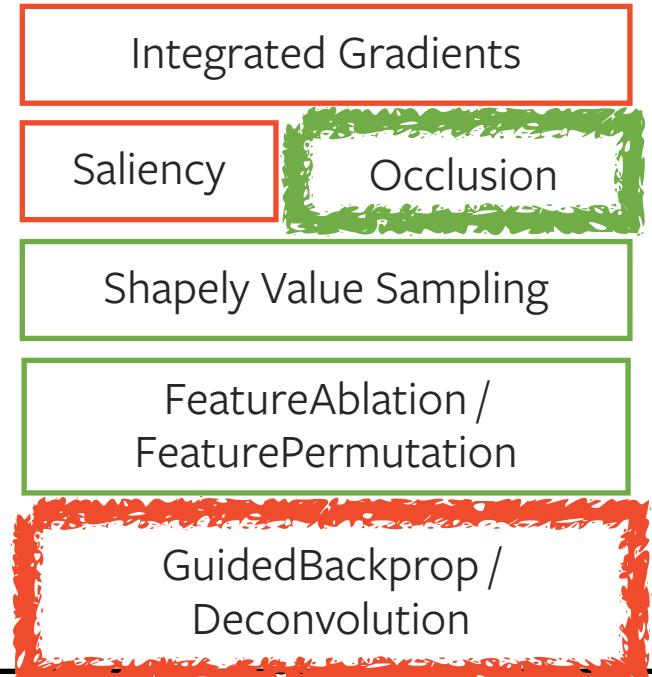
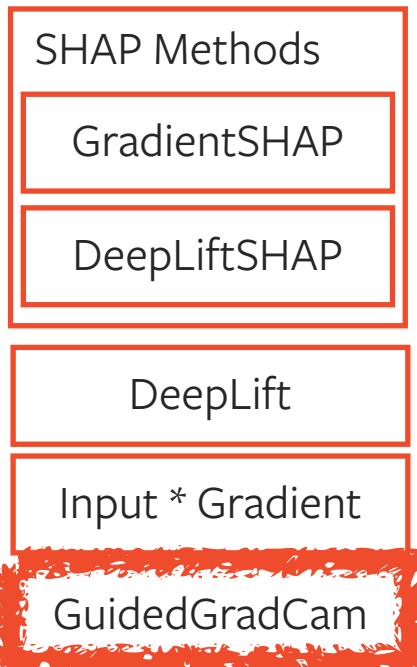
- Gradient
- Perturbation
- Other



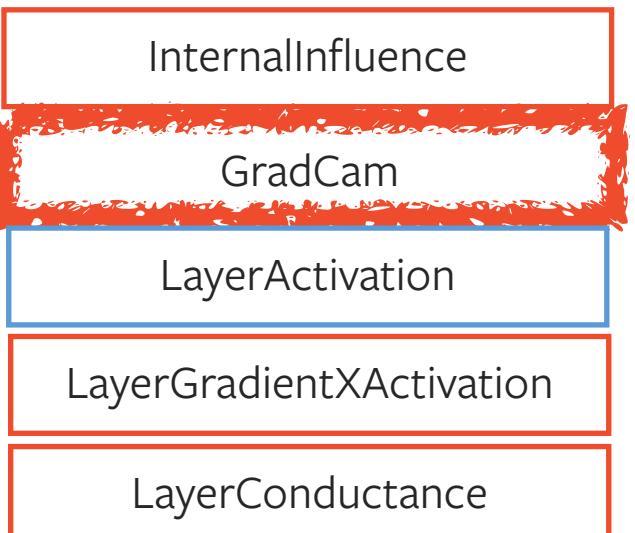
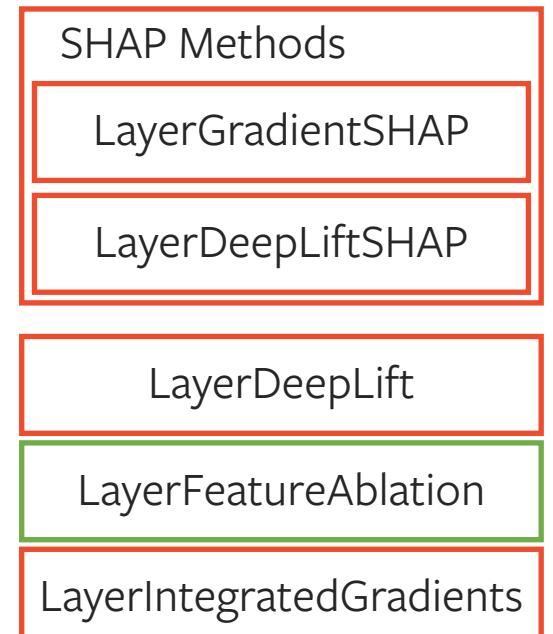
ATTRIBUTION ALGORITHMS

* Often used for computer vision models

Attribute model output (or internal neurons) to input features



Attribute model output to the layers of the model



NoiseTunnel (Smoothgrad, Vargrad, Smoothgrad Square)

- Gradient
- Perturbation
- Other



ATTRIBUTION ALGORITHMS

* Algorithms requiring baseline / reference / background

Attribute model output (or internal neurons) to input features

SHAP Methods

GradientSHAP

DeepLiftSHAP

DeepLift

Input * Gradient

GuidedGradCam

Integrated Gradients

Saliency

Occlusion

Shapely Value Sampling

FeatureAblation /
FeaturePermutation

GuidedBackprop /
Deconvolution

Attribute model output to the layers of the model

SHAP Methods

LayerGradientSHAP

LayerDeepLiftSHAP

LayerDeepLift

LayerFeatureAblation

LayerIntegratedGradients

InternalInfluence

GradCam

LayerActivation

LayerGradientXActivation

LayerConductance

NoiseTunnel (Smoothgrad, Vargrad, Smoothgrad Square)

- Gradient
- Perturbation
- Other

•

WHAT IS BASELINE ?



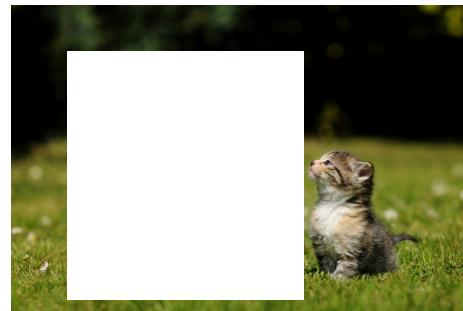
WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**



WHAT IS BASELINE ?

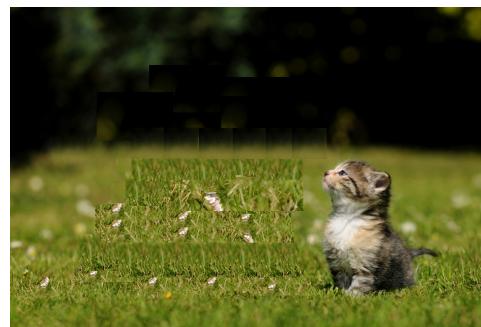
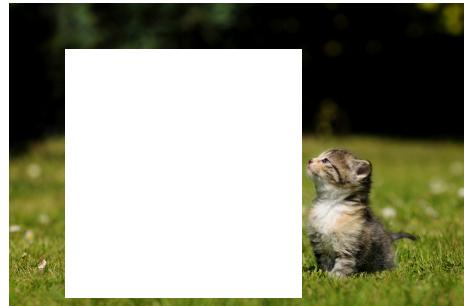
Comparing **input** with the **reference / baseline / background**





WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**





WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**

"Those who know, do.
Those that
understand, teach."
~Aristotle

<PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD>

"Those who know, do.
Those that
understand, teach."
~Aristotle

"Those who know, do.
Those that <PAD>,
teach." ~Aristotle



WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**

feat_1	feat_2	feat_3
12.3	33.3	0.0
3.9	45.8	3.0
21.0	0.0	5.0

feat_1	feat_2	feat_3
12.3	33.3	0.0
3.9	45.8	3.0
21.0	0.0	5.0

feat_1	feat_2	feat_3
12.3	33.3	0.0
3.9	45.8	3.0
21.0	0.0	5.0

feat_1	feat_2	feat_3
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

feat_1	feat_2	feat_3
12.3	0.0	0.0
3.9	0.0	3.0
21.0	0.0	5.0

Ablation

feat_1	feat_2	feat_3
3.9	33.3	0.0
21.0	45.8	3.0
12.3	0.0	5.0

Permutation



WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**

feat_1	feat_2	feat_3
12.3	33.3	0.0
3.9	45.8	3.0
21.0	0.0	5.0

feat_1	feat_2	feat_3
12.3	33.3	0.0
3.9	45.8	3.0
21.0	0.0	5.0

feat_1	feat_2	feat_3
12.3	33.3	0.0
3.9	45.8	3.0
21.0	0.0	5.0

feat_1	feat_2	feat_3
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

feat_1	feat_2	feat_3
12.3	0.0	0.0
3.9	0.0	3.0
21.0	0.0	5.0

Ablation

feat_1	feat_2	feat_3
3.9	33.3	0.0
21.0	45.8	3.0
12.3	0.0	5.0

Permutation

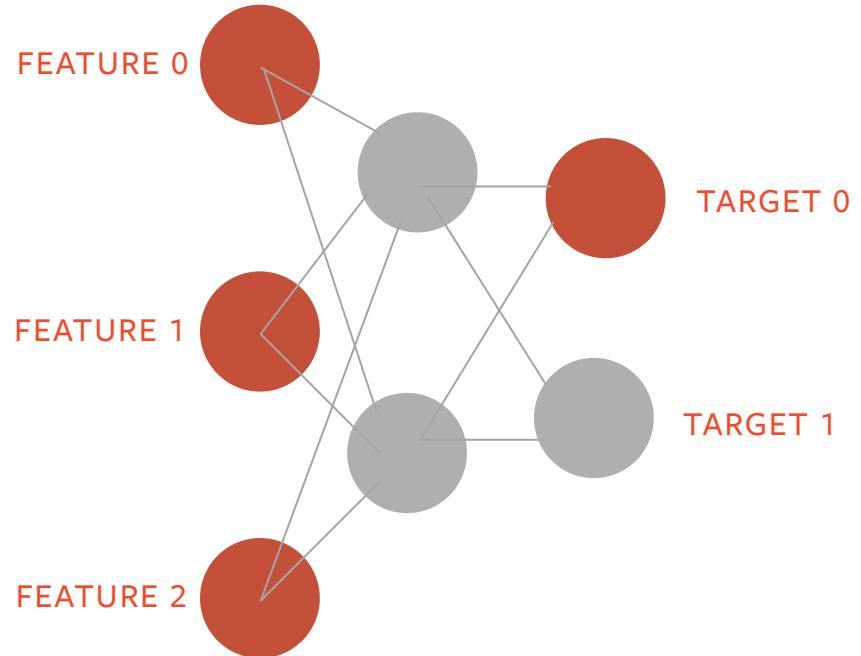
The choice of the reference/baseline/background is challenging!



How can we use Captum library ?



PRIMARY ATTRIBUTION



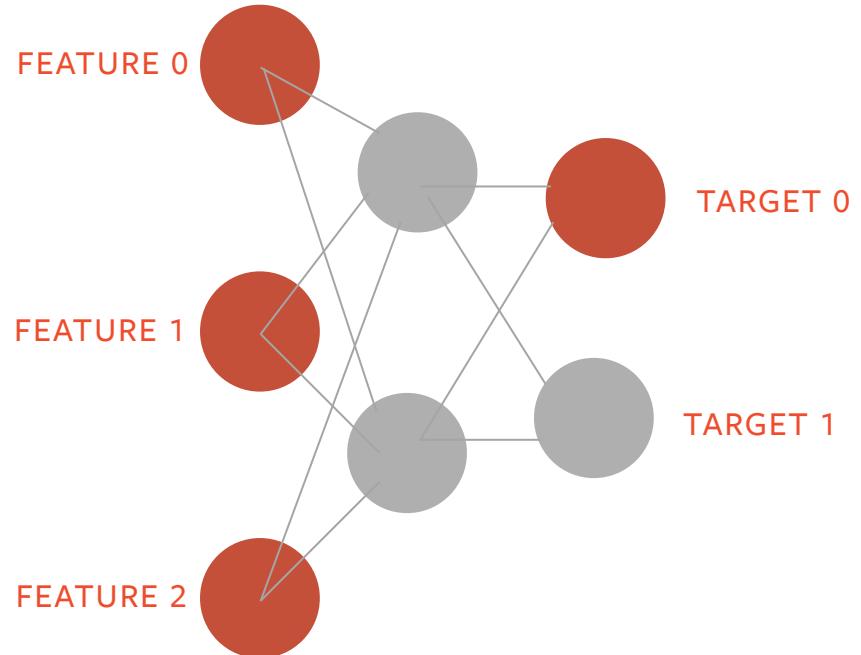
```
from captum.attr import IntegratedGradients  
  
attr_algo = IntegratedGradients(model)  
  
input = torch.rand(1, 3)  
  
attribution = attr_algo.attribute(input, target=0)
```

Results

```
attribution: [[0.1, -0.3, 0.0]]
```



PRIMARY ATTRIBUTION



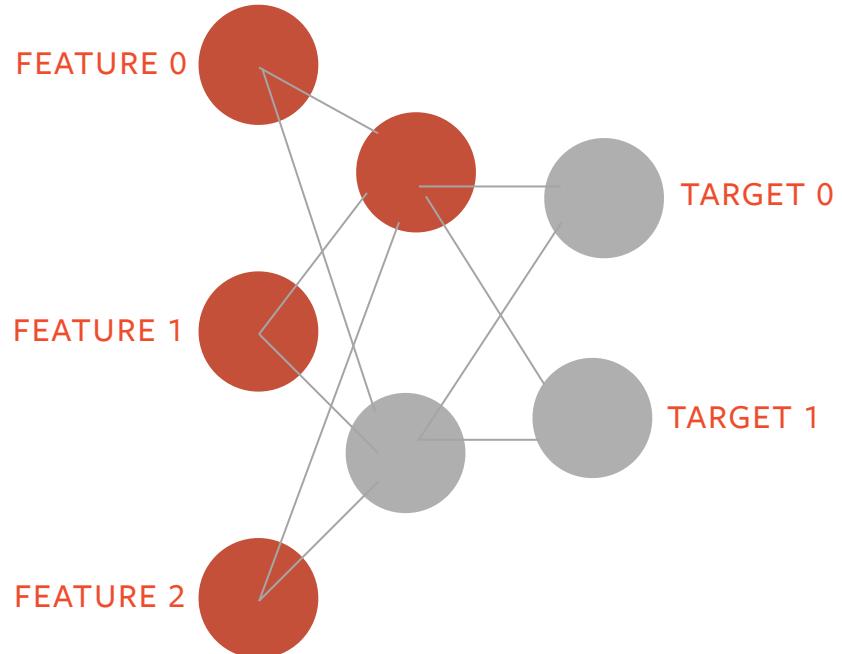
```
from captum.attr import IntegratedGradients  
  
attr_algo = IntegratedGradients(model)  
  
input = torch.rand(1, 3)  
  
baseline = torch.zeros(1, 3)  
  
attributions = attr_algo.attribute(input,  
                                    baselines=baseline  
                                    target=0)
```

Results

```
attributions: [[0.1, -0.3, 0.0]]
```



NEURON ATTRIBUTION



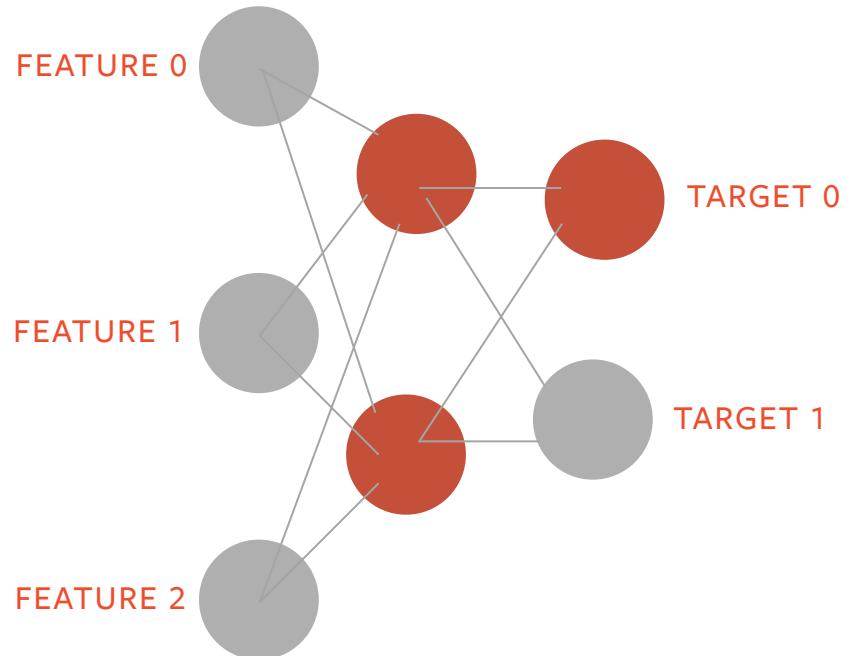
```
from captum.attr import NeuronConductance  
  
attr_algo = NeuronConductance(model, model.lin)  
  
input = torch.rand(1, 3)  
  
attribution = attr_algo.attribute(input,  
                                  neuron_ind = 0)
```

Results

```
attribution: [[0.0, -0.03, -0.01]]
```



LAYER ATTRIBUTION



```
from captum.attr import LayerConductance  
  
attr_algo = LayerConductance(model, model.lin)  
  
input = torch.rand(1, 3)  
  
attributions = attr_algo.attribute(input,  
target = 0)
```

Results

```
attributions: [[0.2, 0.0]]
```



```
attributions = Attribution(forward_func, ...).attribute(input, ...)*
```

* Check out our Getting Started docs and API:
<https://github.com/pytorch/captum>
<https://captum.ai/api/>



SCALING ALGORITHMS

- + PyTorch DataParallel models are supported across all algorithms
 - + Including distributed execution of forward / backward hooks
- + Optional internal slicing of large input tensors
 - + Computing forward/backward passes on smaller slices of inputs and aggregating outputs
- + Perturbing multiple features together on the basis of batches

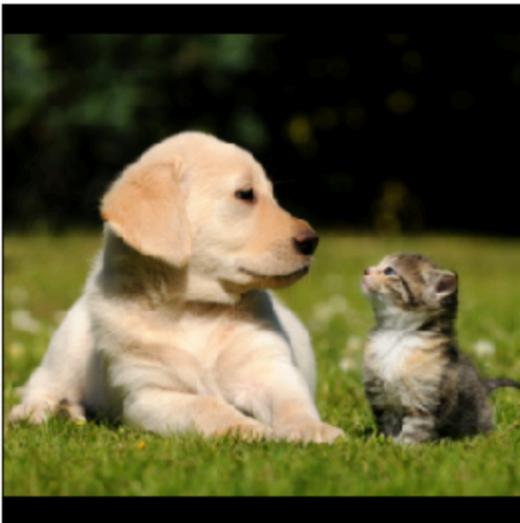


HOW CAN WE VISUALIZE
THE ATTRIBUTIONS
OF COMPLEX MODELS?

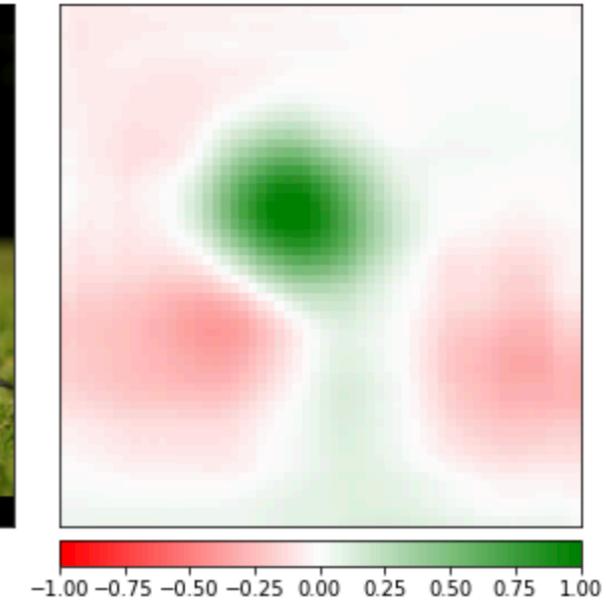


VISUALIZATIONS USING RESNET152 MODEL

ORIGINAL IMAGE



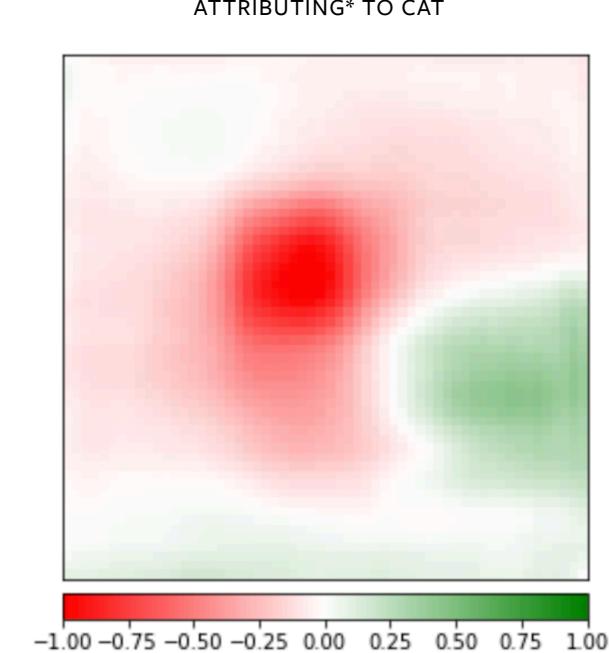
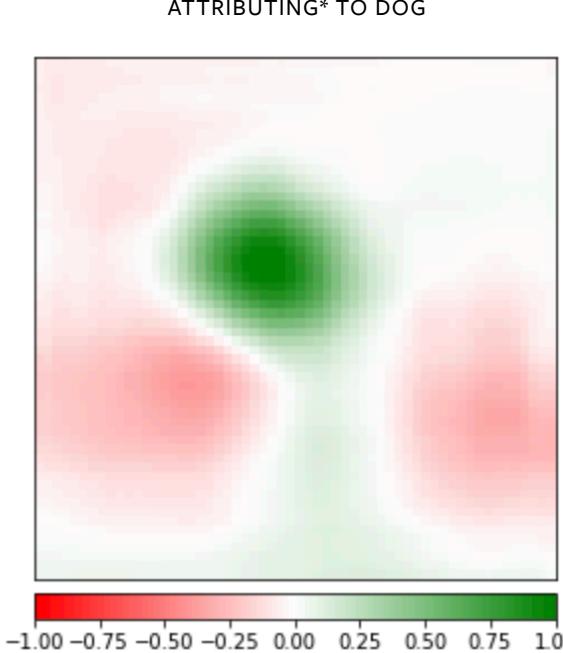
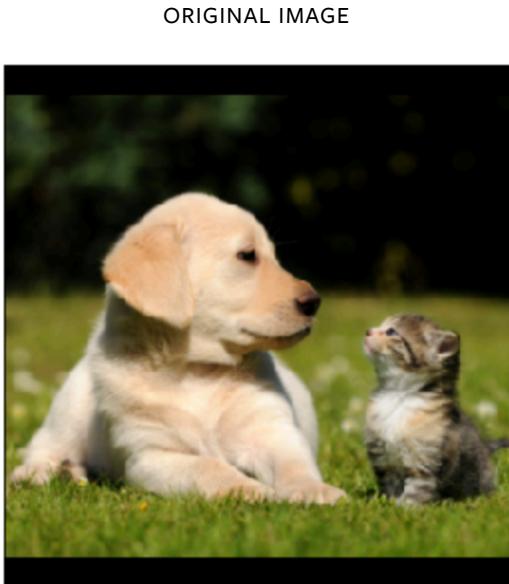
ATTRIBUTING TO DOG*



* MATTHEW D ZEILER, ROB FERGUS, OCCLUSION: VISUALIZING AND UNDERSTANDING CONVOLUTIONAL NETWORKS, IN SPRINGER INTERNATIONAL PUBLISHING SWITZERLAND, 2014



VISUALIZATIONS USING RESNET152 MODEL



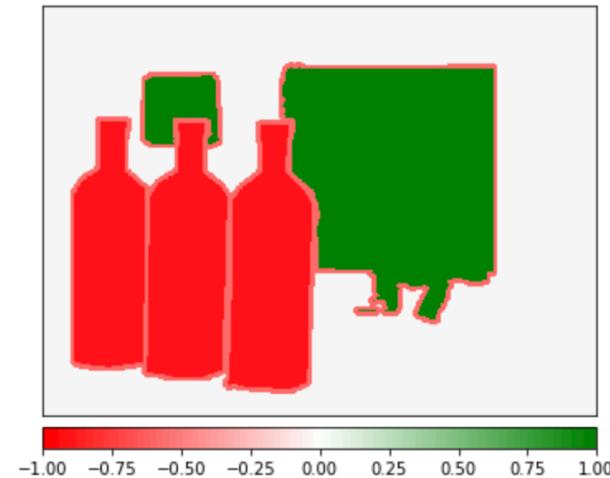


VISUALIZATIONS USING RESNET18 MODEL AND VOC SEGMENTATION

FEATURE ABALATION BASED ON IMAGE SEGMENTATION



ATTRIBUTING TO MONITOR





An interactive visualization tool for

+ debugging and understanding model predictions

Supports different types of PyTorch models and input features



VISUALIZING EXPLANATIONS OF RESNET-50 MODEL WITH CAPTUM INSIGHTS

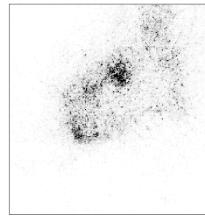
CAPTUM INSIGHTS

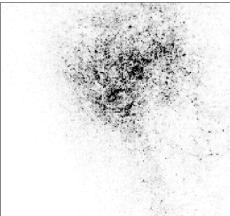
Instance Attribution Direct Target Export

Filter by Classes Animal and 2 other classes are selected. [Edit](#)

Filter by Instances Prediction: All

Integrated Gradients Approximation steps: 50

Predicted	Label	Contribution	Photo (Image)
wood_rabbit (0.962)	wood_rabbit	Photo	 Original
hare (0.037)			 Attribution Magnitude
wallaby (0.000)			
fox_squirrel (0.000)			

Predicted	Label	Contribution	Photo (Image)
lynx (0.530)	lion	Photo	 Original
lion (0.351)			 Attribution Magnitude
cheetah (0.084)			
cougar (0.020)			



VISUALIZING EXPLANATIONS OF A TEXT CLASSIFICATION MODEL USING IMDB DATASET

WITH
CAPTUM INSIGHTS

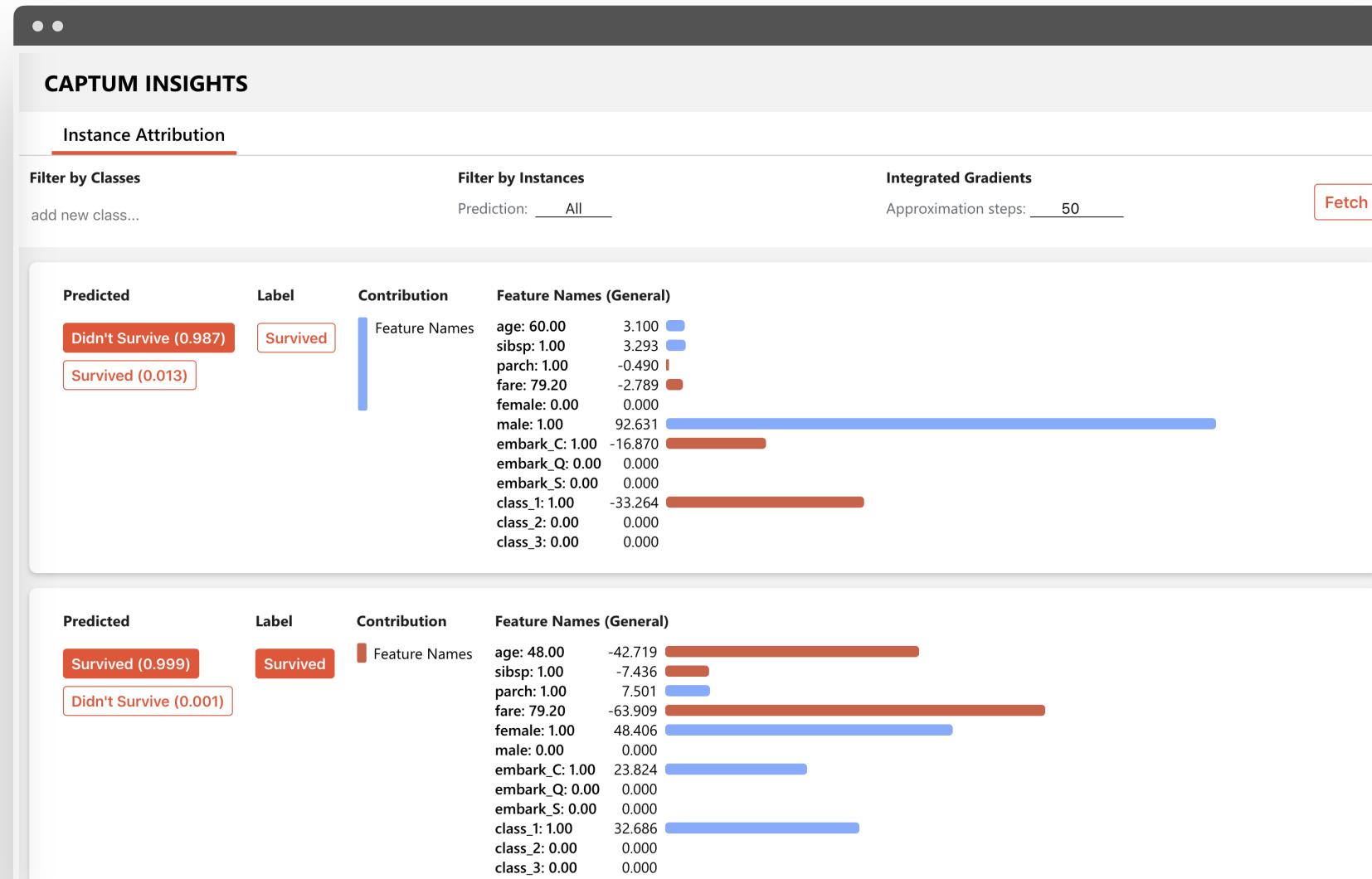
The screenshot shows the Captum Insights interface with the title "CAPTUM INSIGHTS". The top navigation bar includes "Instance Attribution" (which is selected), "Direct Target", and "Export". Below the navigation are three sections: "Filter by Classes" (Animal and 2 other classes are selected, Edit), "Filter by Instances" (Prediction: All), and "Integrated Gradients" (Approximation steps: 50). The main content area displays two rows of explanations for movie reviews. Each row has columns for "Predicted" (Positive or Negative), "Label" (Negative), "Contribution" (Movie Reviews), and "Movie Reviews (Text)". The first review is predicted as Positive (0.874) and labeled as Negative. The second review is predicted as Negative (0.022) and labeled as Negative. The "Contribution" column for both reviews is "Movie Reviews". The "Movie Reviews (Text)" column contains the movie reviews with tokens highlighted in blue and red, indicating their contribution to the model's prediction.

Predicted	Label	Contribution	Movie Reviews (Text)
Positive (0.874)	Negative	Movie Reviews	carla <unk> literally melts the screen in this crime caper . her sex appeal , ample assets and sexy southern accent more t <unk> . but the film has so many " other " things to make it almost a modern classic . simon baker 's performance for o of a great actor . til schweiger is so perfectly cast it makes me wonder why he 's not a huge " crossover " star . gil bell <unk> crew and literally steals this show . other <unk> 's cleavage easily the best performance in the movie . emma th rickman are so utterly brilliant ... i can not even describe it . great chemistry. the music is so powerfully eclectic character in and of itself . wonderous to behold. people may say it 's tarantino influenced ... i disagree . it 's clever in it 's own right . sebastian <unk> does a wonderful job at directing this tightly woven film and underplaying a lot me reeling . the ending is perfectly sinister and you never see it coming . all in all a lovely film that , to me , is the very good movie should be .
Negative (0.022)	Negative	Movie Reviews	this movie had great production values , good lighting , costumes , set , cinematography and acting . but someone , some script , and replaced all the dialogue with grade - school level barely literate writing . i felt my iq dropping points any t <unk> /> did they do this on purpose ? was this just an accident of brain dead studio executives ? at this point , /> all i know is , this movie was one great mistake from beginning to end . we do n't even get to see how the s so instead of any character development , we get what feels like a bad tv - movie leftover from the 60 's . or <unk> /> screenwriters , beat them with a sock full of quarters . everyone else , nice work , but read your scripts next time.



VISUALIZING EXPLANATIONS OF A 3-LAYER MLP MODEL USING TITANIC DATASET

WITH
CAPTUM INSIGHTS





VISUALIZING EXPLANATIONS OF MULTIMODAL VQA MODELS

CAPTUM INSIGHTS

Instance Attribution Direct Target Export

Filter by Classes
Animal and 2 other classes are selected. [Edit](#)

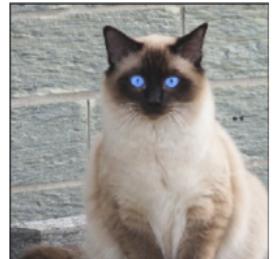
Filter by Instances
Instance Type: All

Integrated Gradients
Approximation steps: 50

Fetch

Predicted	Label	Contribution	Question (Text)
Blue (0.517)	Blue	Question	What color are the cats eyes?
Green (0.176)			
Yellow (0.128)		Image	

Image



Original Attribution Magnitude



CASE STUDY FOR DEEP LEARNING RECOMMENDER MODEL (DLRM)



CASE STUDY OVERVIEW

1. DLRM model and dataset
2. Dense and Sparse Feature importances
3. Feature importances in interaction layer
4. Neuron Importances for the last FC layer
5. Model pruning & performance



DATASET

- + Criteo's traffic over a period of 7 days*

 - + train: ~ 39M samples

 - + test: ~ 3.2M samples

- + Highly unbalanced dataset

 - + clicks: 26%

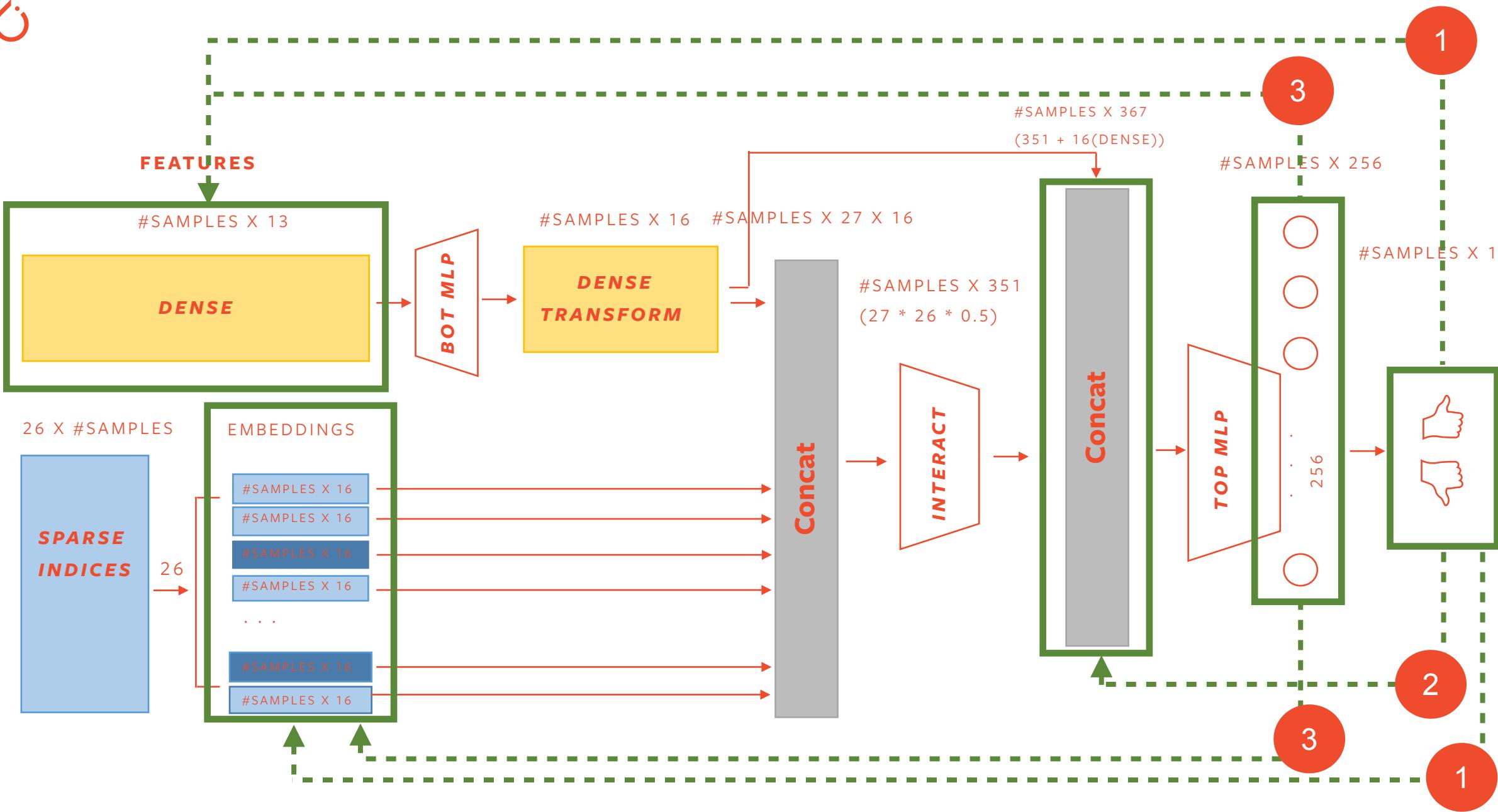
- + Features

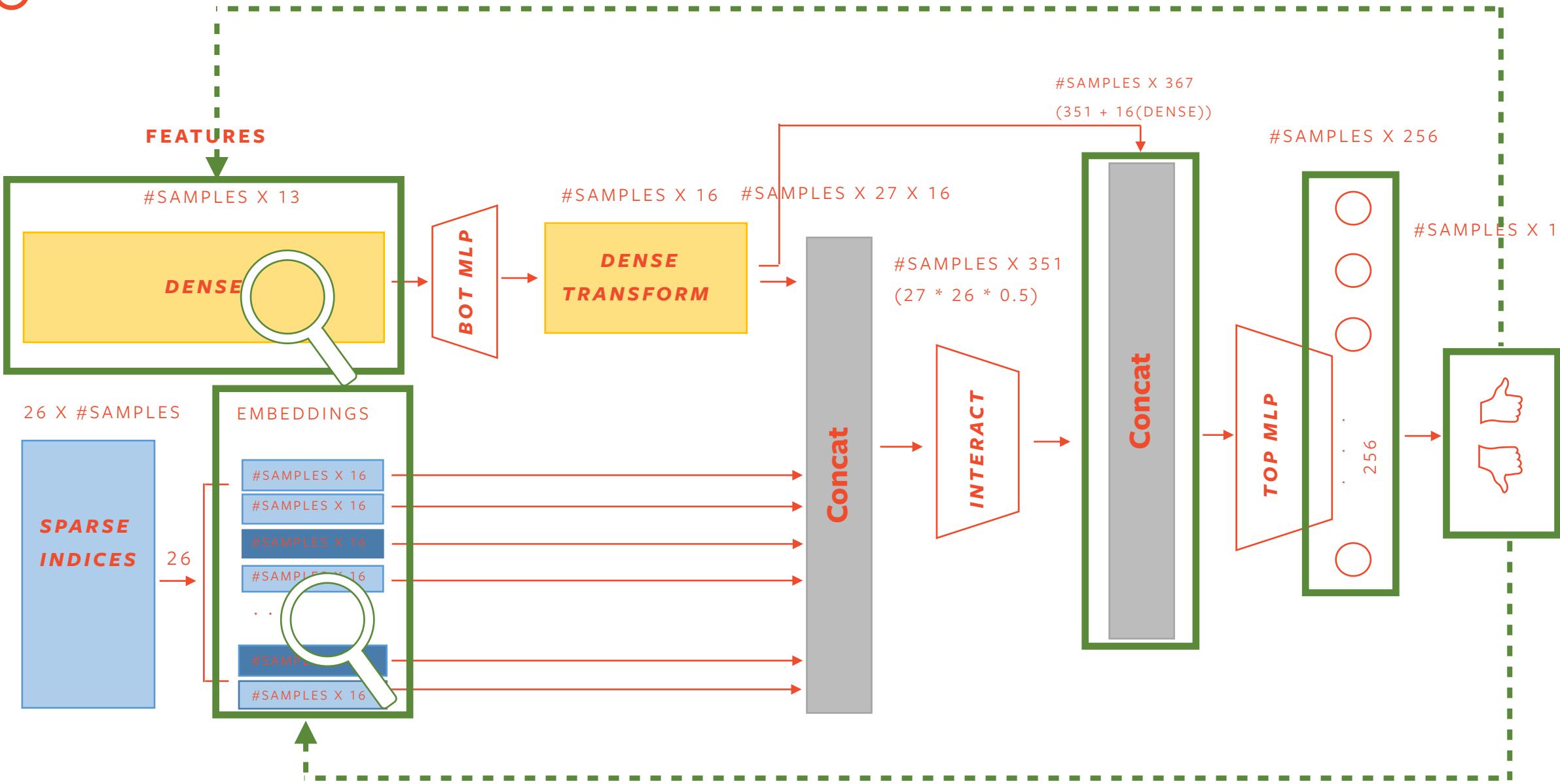
 - + sparse: 26

 - + dense: 13

*<https://www.kaggle.com/c/criteo-display-ad-challenge/data>

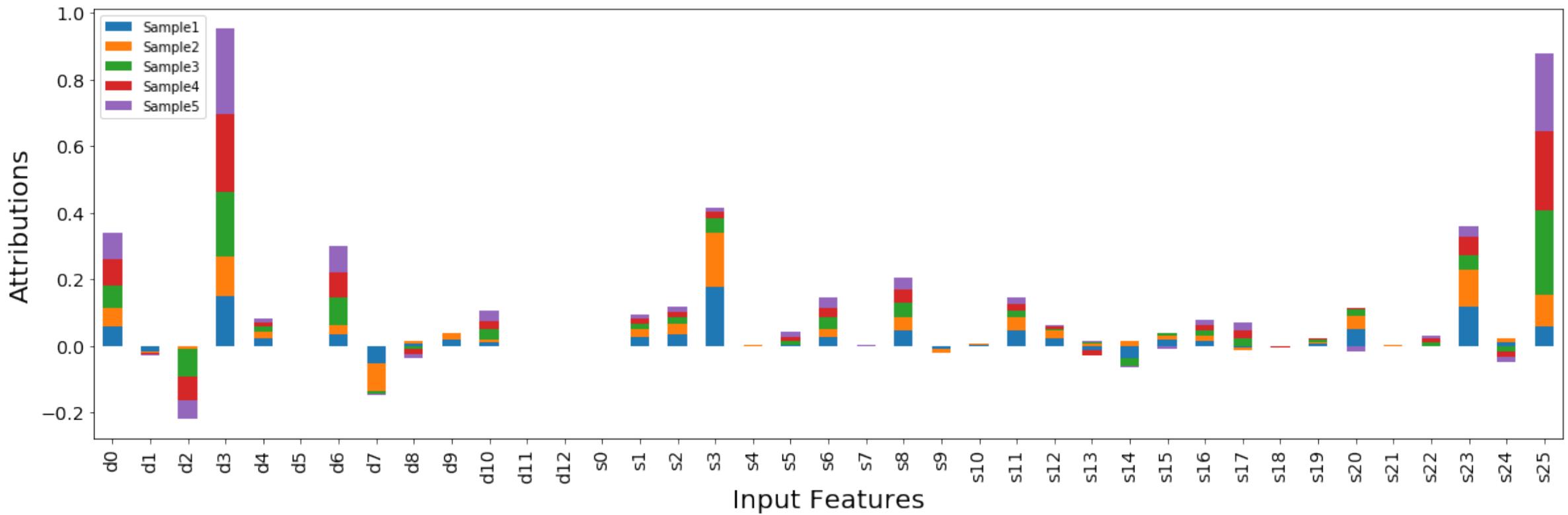
O





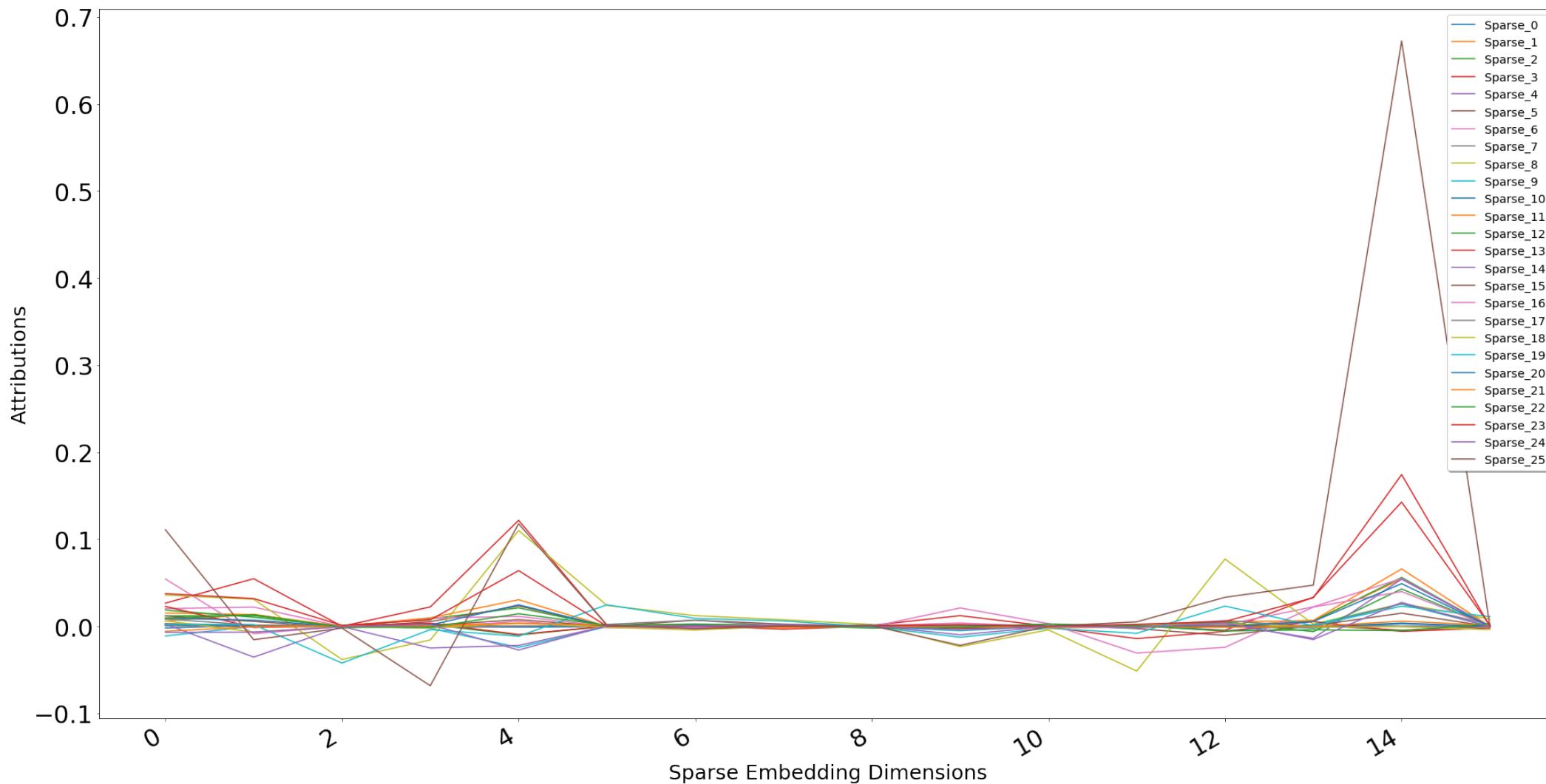


FEATURE IMPORTANCES FOR 5 SAMPLES WITH PREDICTION SCORE > 0.999





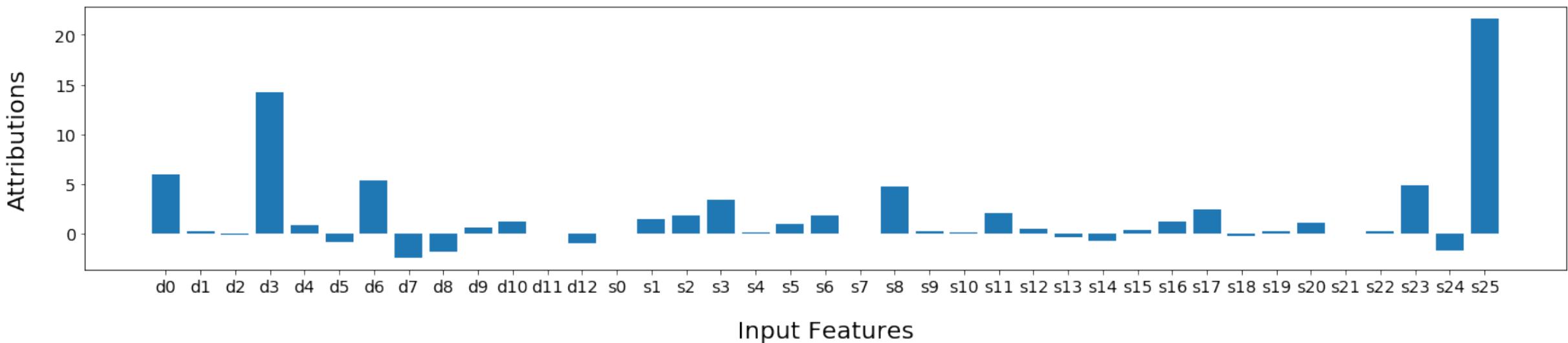
SPARSE FEATURE IMPORTANCES FOR 85 SAMPLES WITH PREDICTION SCORE > 0.999





FEATURE IMPORTANCES

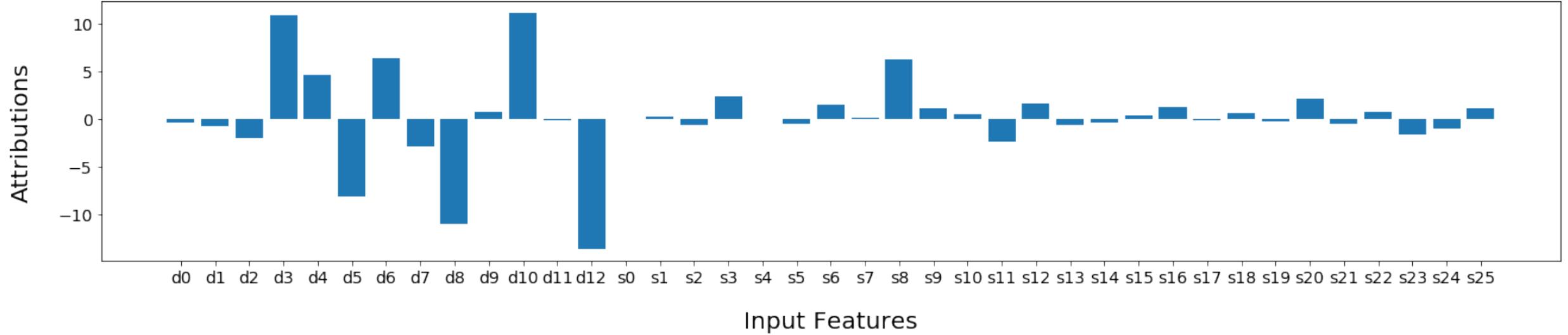
Top 85 samples with pred > 0.99





FEATURE IMPORTANCES

Aggregated across 85 samples

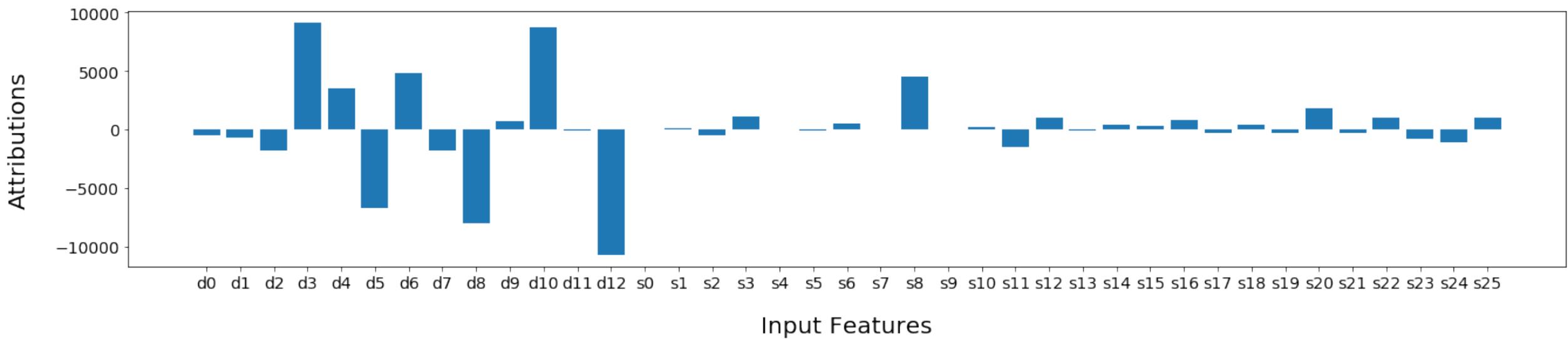


* ONLY 13% OF ALL SAMPLES ARE CLICKS



FEATURE IMPORTANCES

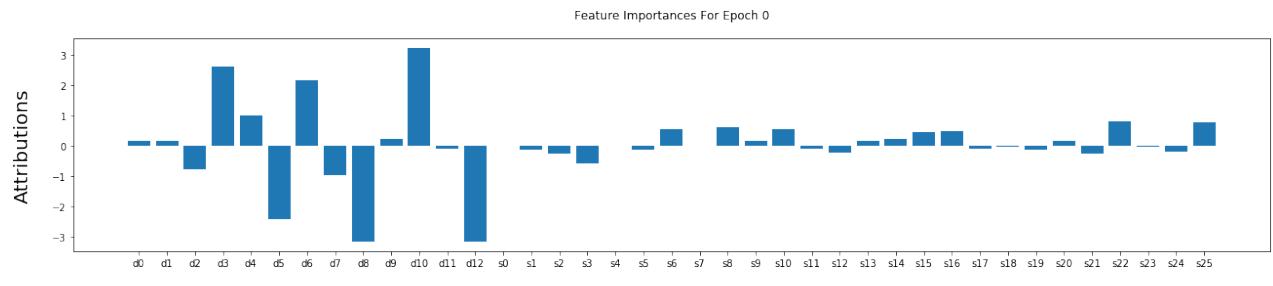
Aggregated across 135k samples



* ONLY 13% OF ALL SAMPLES ARE CLICKS

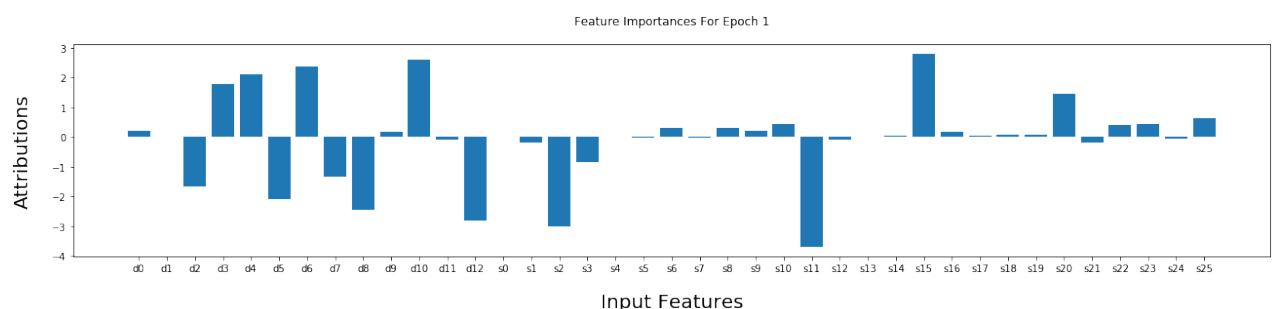


COMPARING FEATURE IMPORTANCES ACROSS DIFFERENT EPOCHS



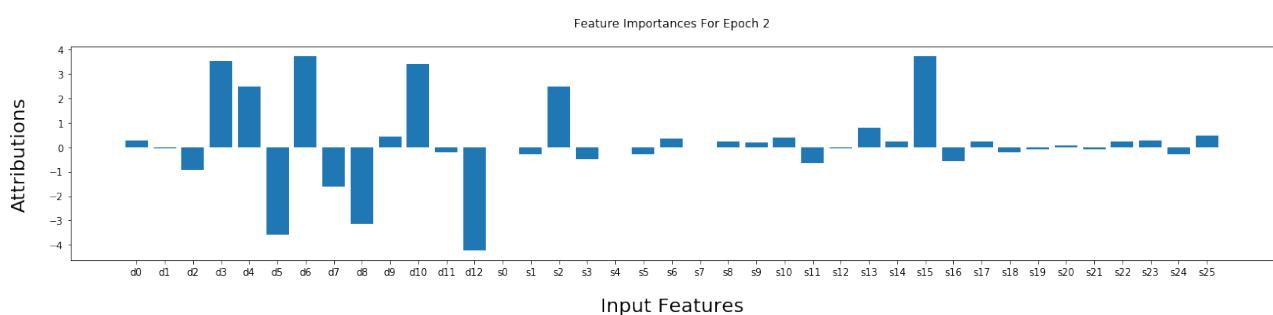
Epoch 0

Accuracy: 78.80%
ROC AUC: 0.8
F1: 0.51



Epoch 1

Accuracy: 78.34%
ROC AUC: 0.79
F1: 0.47

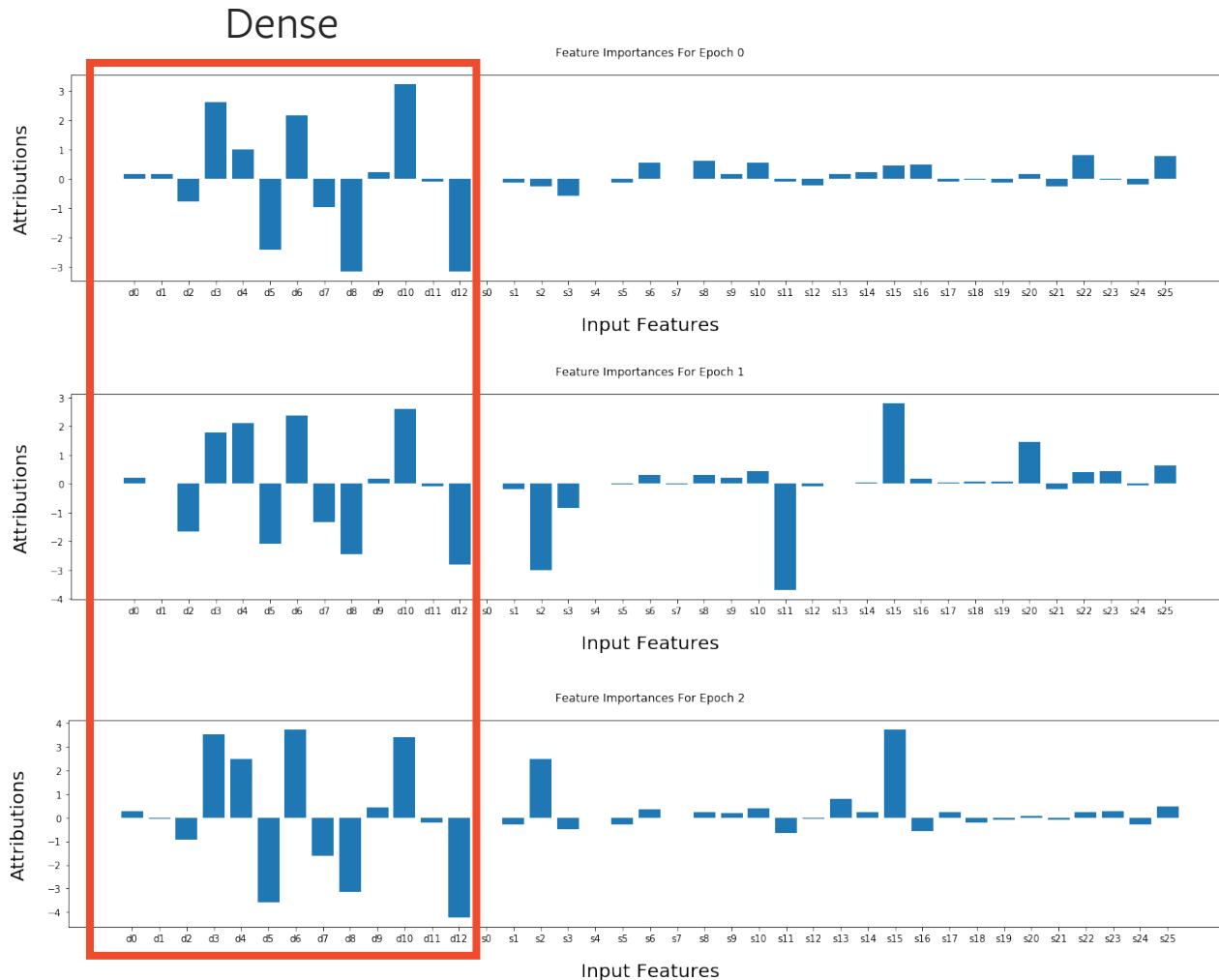


Epoch 2

Accuracy: 78.52%
ROC AUC: 0.75
F1: 0.50



COMPARING FEATURE IMPORTANCES ACROSS DIFFERENT EPOCHS



Epoch 0

Accuracy: 78.80%
ROC AUC: 0.8
F1: 0.51

Epoch 1

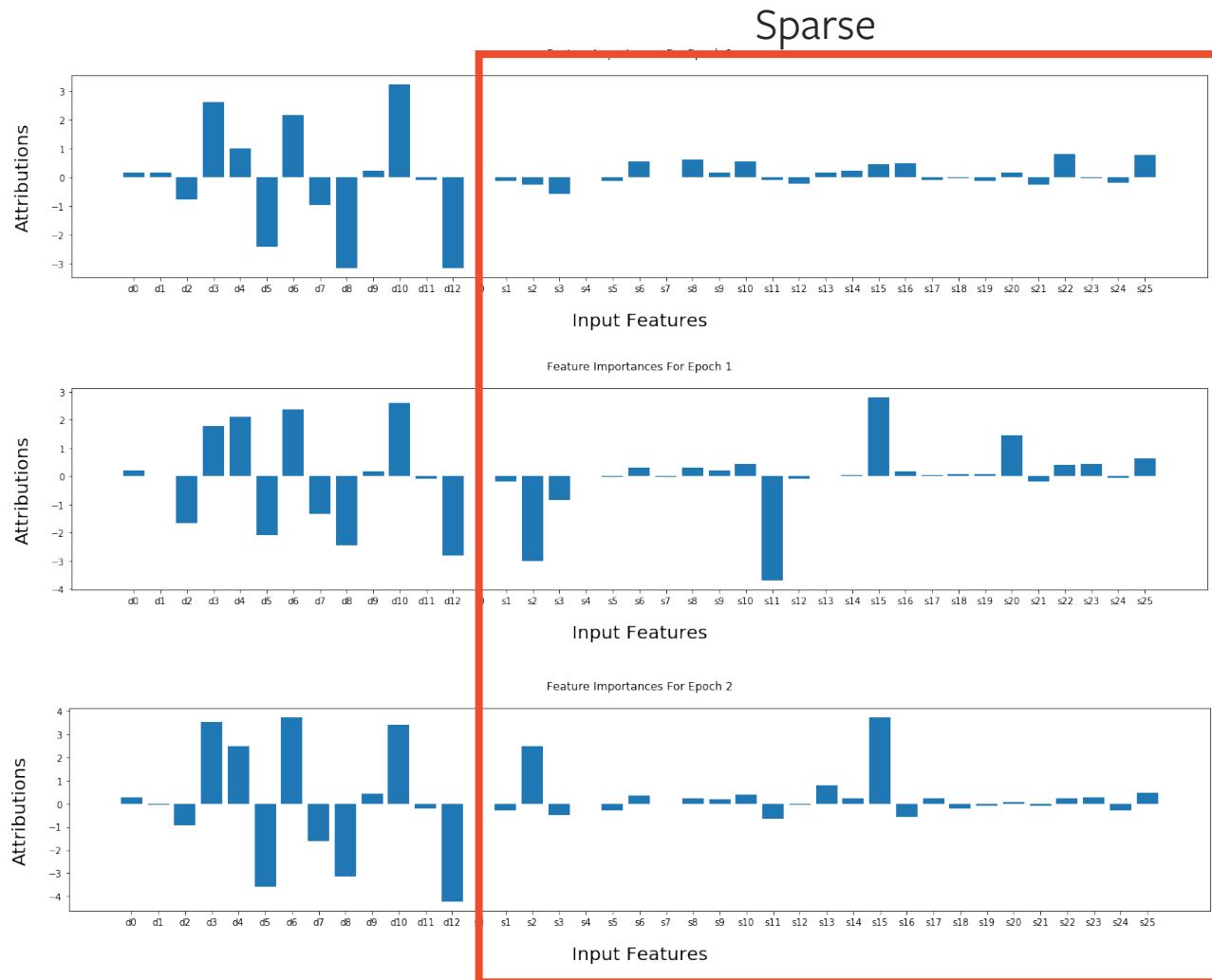
Accuracy: 78.34%
ROC AUC: 0.79
F1: 0.47

Epoch 2

Accuracy: 78.52%
ROC AUC: 0.75
F1: 0.50



COMPARING FEATURE IMPORTANCES ACROSS DIFFERENT EPOCHS



Epoch 0

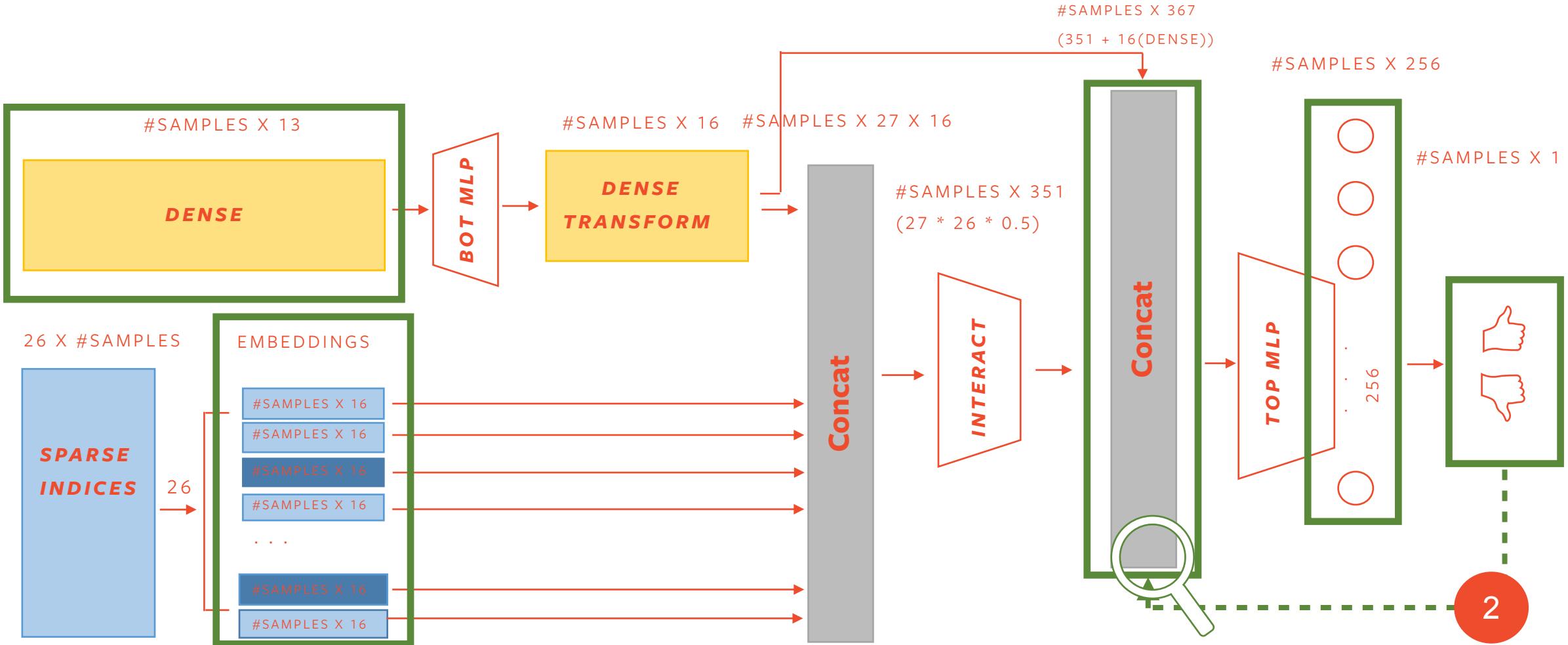
Accuracy: 78.80%
ROC AUC: 0.8
F1: 0.51

Epoch 1

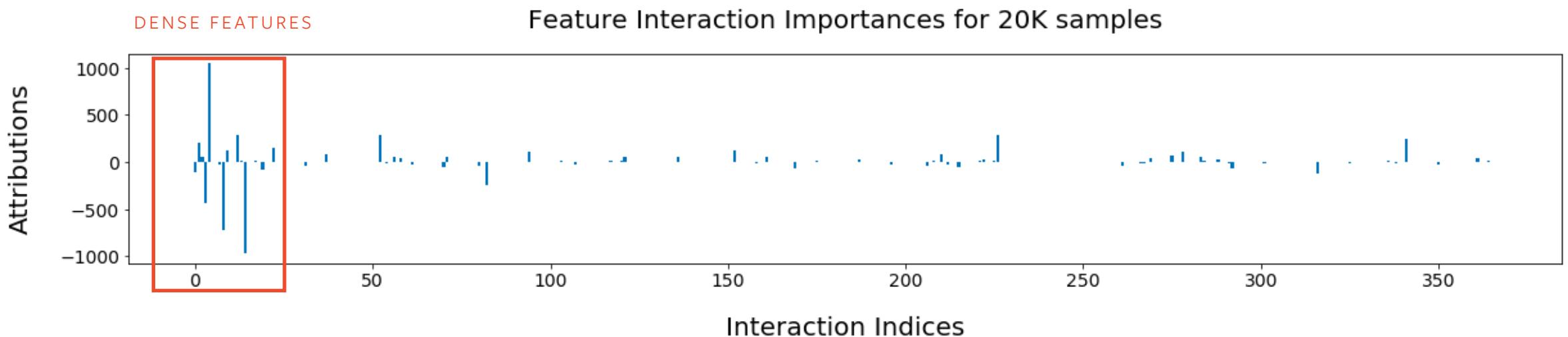
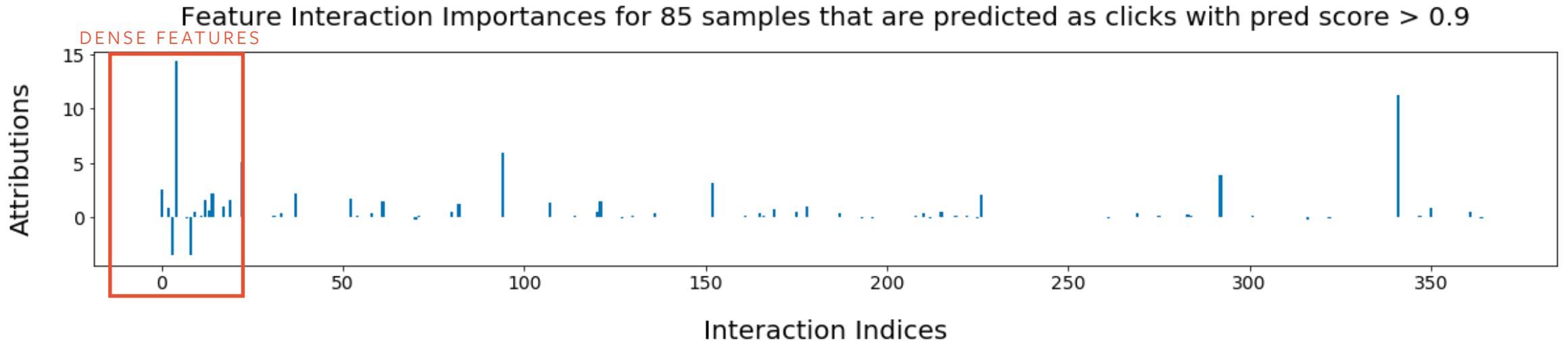
Accuracy: 78.34%
ROC AUC: 0.79
F1: 0.47

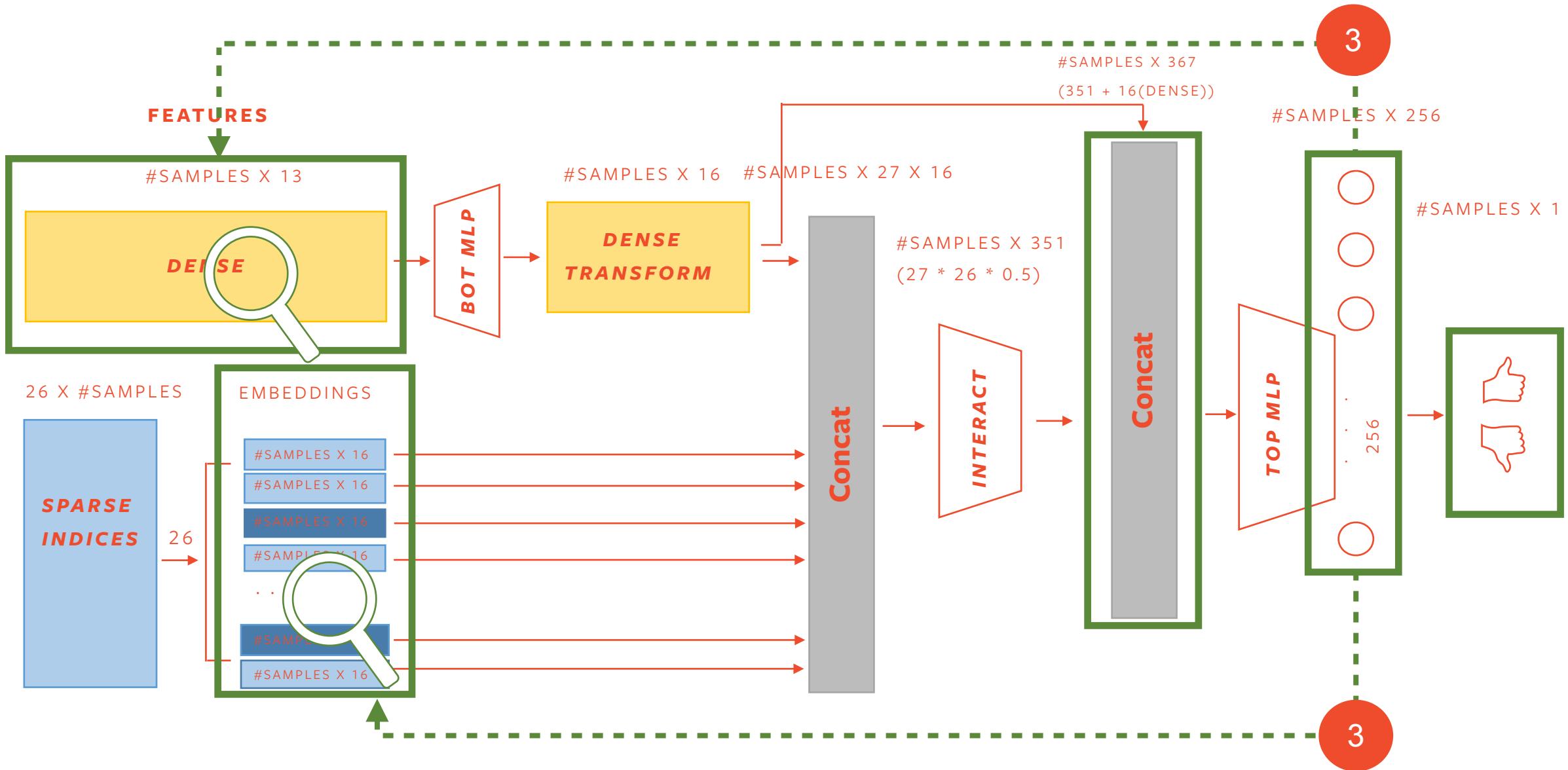
Epoch 2

Accuracy: 78.52%
ROC AUC: 0.79
F1: 0.50



FEATURE IMPORTANCES FOR 86 SAMPLES WITH PREDICTION SCORE > 0.9

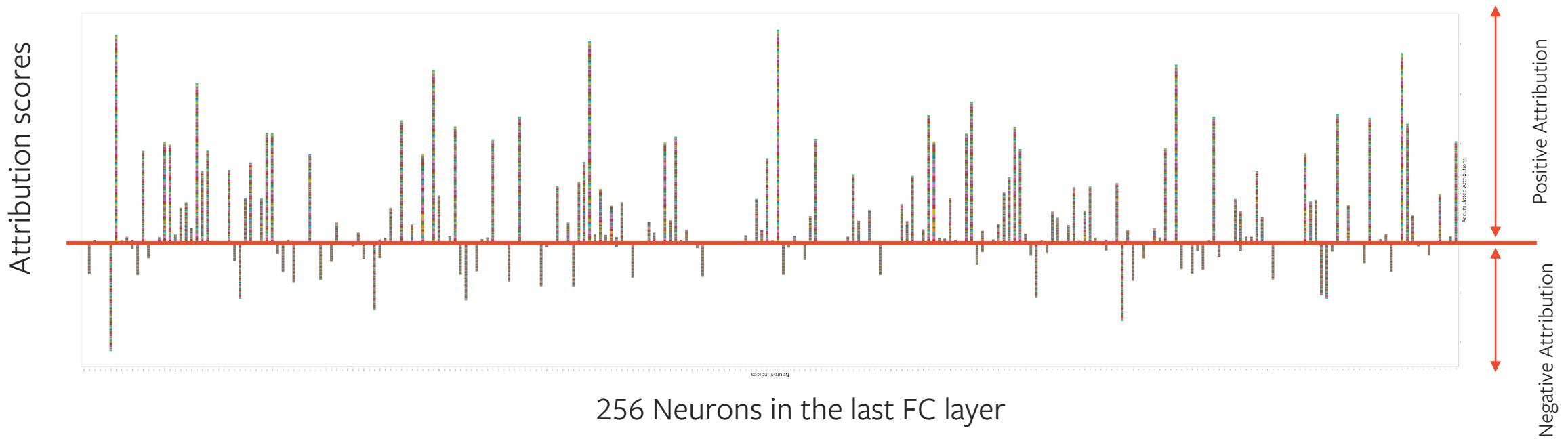






NEURON IMPORTANCE

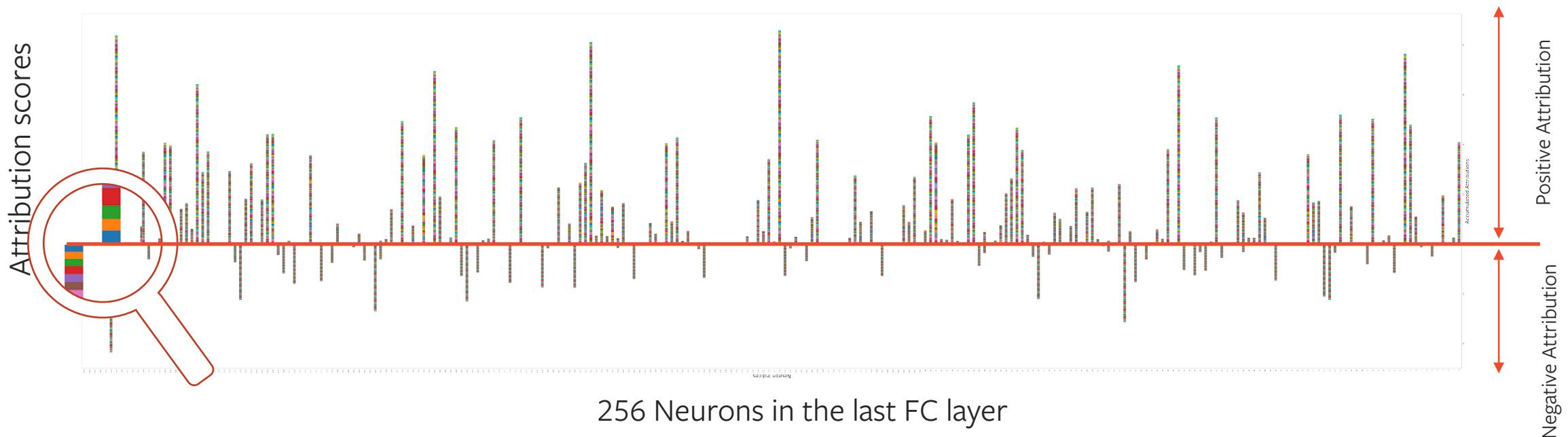
NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9





NEURON IMPORTANCE

NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9

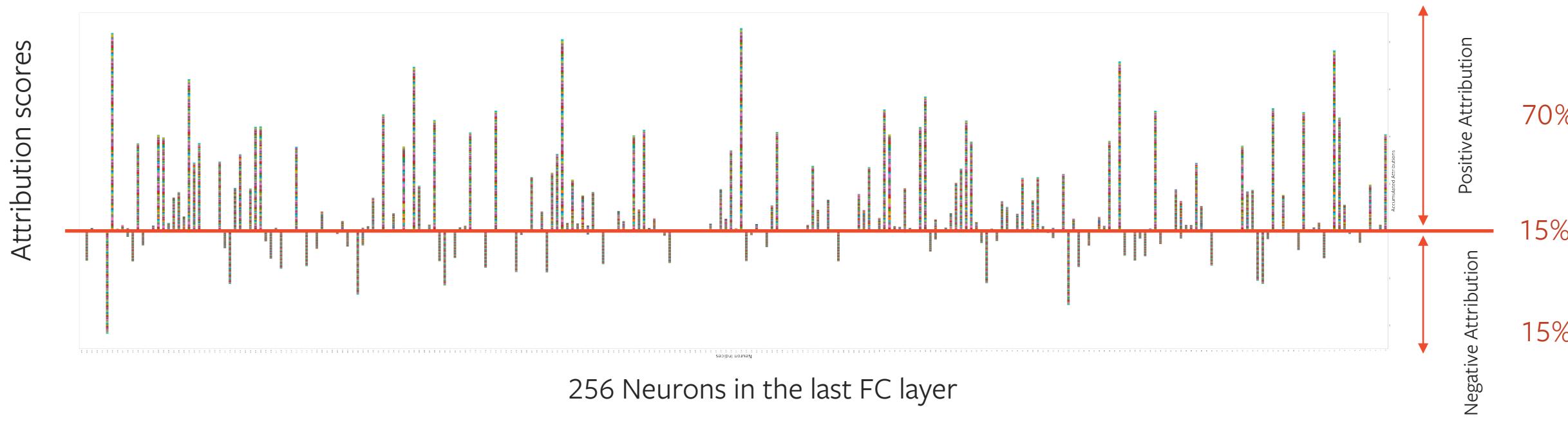


* Each input example has a different color coding



NEURON IMPORTANCE

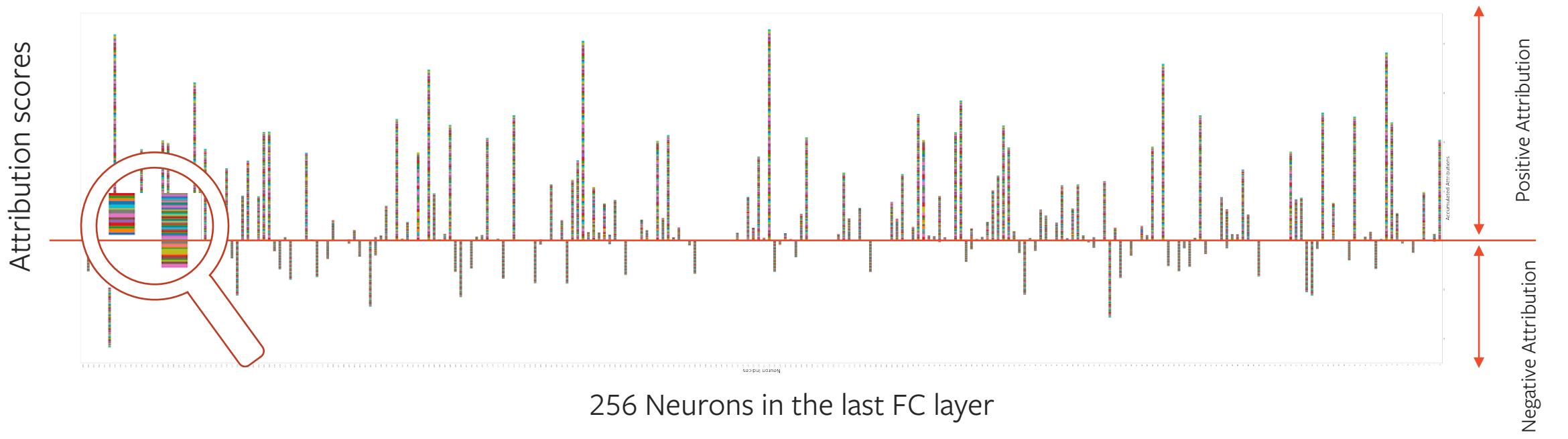
NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9





NEURON IMPORTANCE

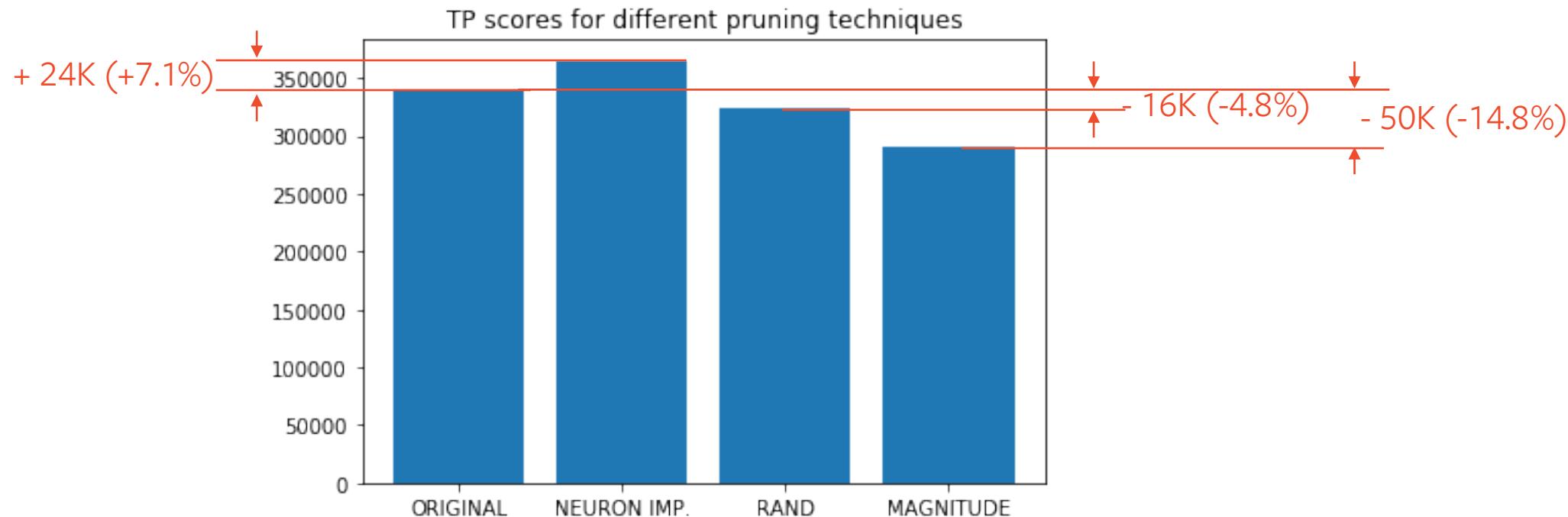
NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9





MODEL PRUNING

PRUNED ~15% OF ALL 256 NEURONS IN THE LAST FC LAYER





MODEL PRUNING

PRUNED ~15% OF ALL 256 NEURONS IN THE LAST FC LAYER

Performance Measures	Original model	Pruned with Neuron Importance	Randomly Pruned	Pruned with Weight Magnitude
Precision	66%	64%	67%	69%
Recall	40%	43%	38%	34%
F1	50%	51%	48%	45%
ROC AUC	80%	80%	80%	80%
Accuracy	79%	79%	79%	79%



CASE STUDY SUMMARY

- + Sparse features primarily contribute to clicks predictions
- + Sparse Feature importance patterns significantly fluctuate across epochs
- + Feature interactions results to features that primarily contribute to clicks or have no affects on the prediction
- + Neuron importance based pruning can help us to increase TP, F1 and Recall scores and reduce FN
- + Magnitude based pruning can help us to reduce FP score and increase Precision.



THE LIMITATIONS OF ATTRIBUTIONS

- + Attributions do not capture feature correlations and interactions
- + Finding good baselines is challenging
- + They are difficult to evaluate
- + Attributions do not explain the model globally



FUTURE DIRECTIONS

+ **captum.robust**

- + adversarial robustness and attacks
- + studying the connections between model robustness and interpretability

+ **captum.metrics**

- + model interpretability, sensitivity, trust, infidelity and robustness related metrics

+ **captum.concept**

- + Testing with Concept Activation Vectors (TCAV)

+ **captum.optim**

- + optimization-based visualizations

...