Towards Interpretable AI: Visualization of Machine Learning Models

Harish Guruprasad Ramaswamy
IIT Madras and RBCDSAI

Outline

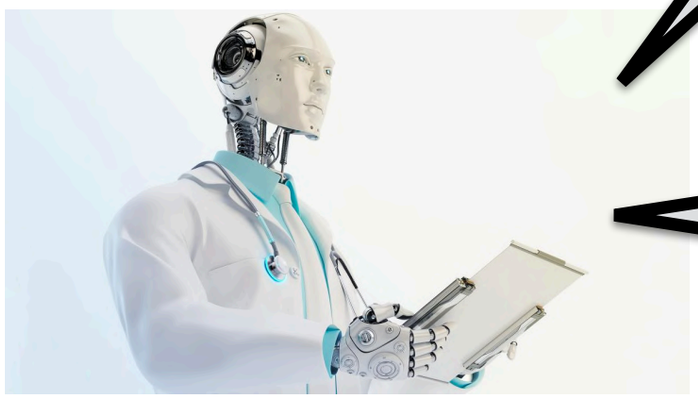
1. Need for interpretability in AI

2. Neural Models
   - Linear
   - Trees
   - Multi-layer Perceptrons
   - CNNs
   - Attention networks
   - RNNs

3. Interpretability Paradigms
   - Intrinsic
   - Reverse Engg.
   - Feature visualization
   - Attribution maps
   - Influence functions
   - Other

4. Caveat Emptor

A Visit to A Robot Doctor

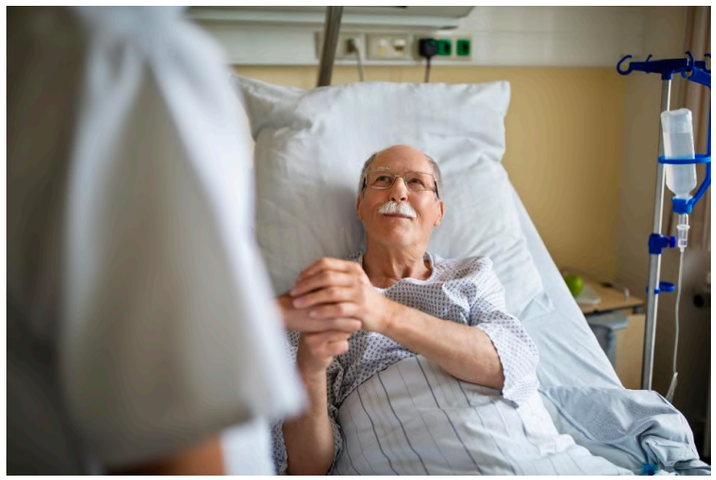


You are likely to have a heart attack within a month

What?? Why?

Because
“BP*top_right_X_ray+
(Sugar-sin(BP^2)) > 45”

?!*#!!!



A Racist Surveillance Bot

A Data Scientist Stretched Thin

Training data

Healthy

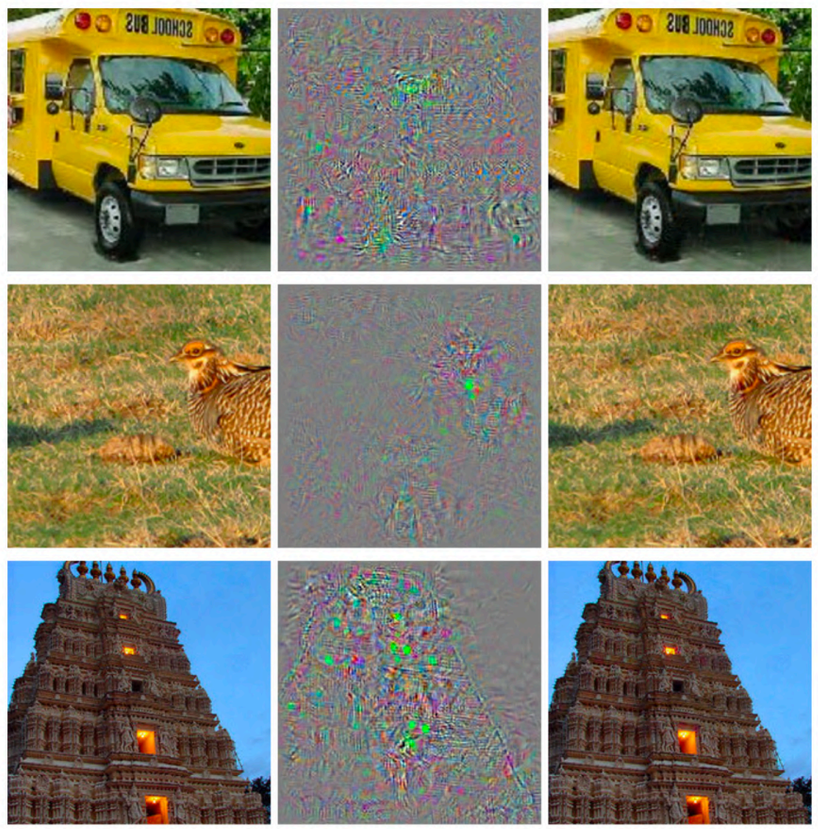
Healthy

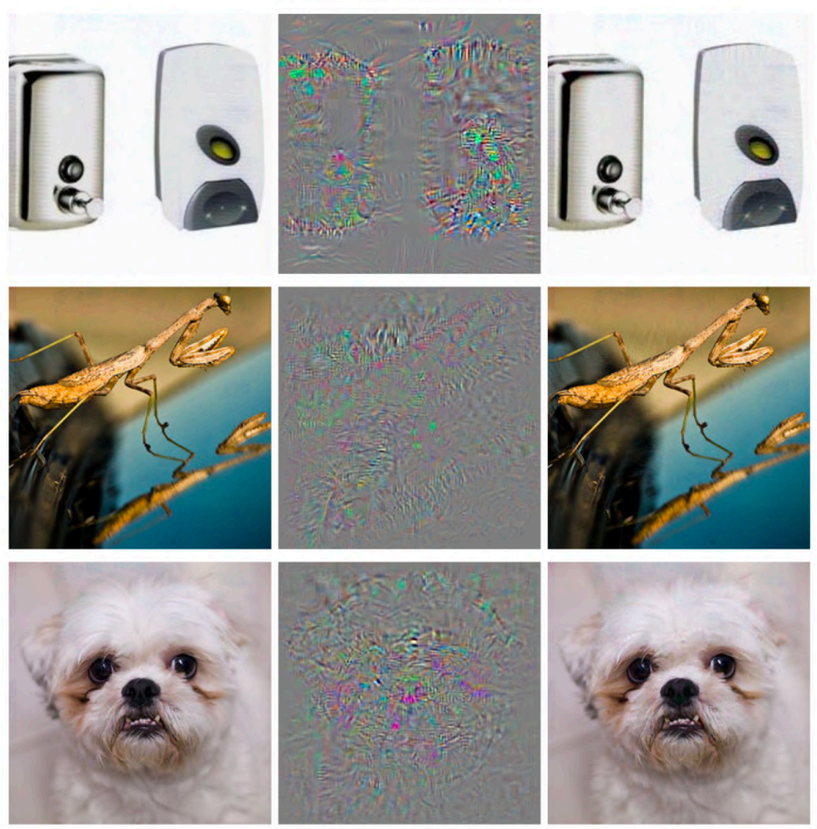
Diseased

Diseased

Testing data

Adversarial Model Attacks



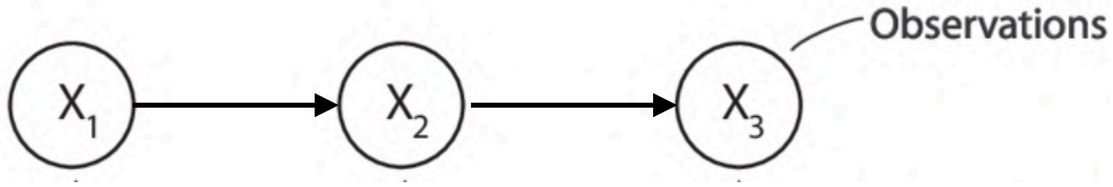


Interpretability Goals

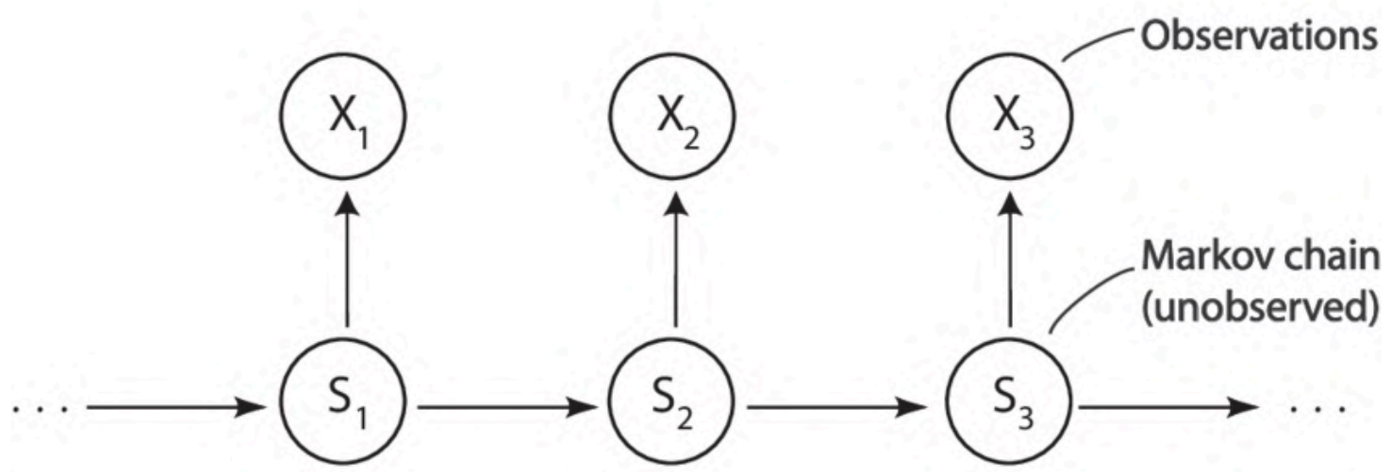
- Accuracy - Interpretability tradeoff
- General philosophy of the model — Global methods
- Precise explanations of a prediction —Local Methods

Latent Variable Models: Its Power and Curse

Simple fully observed models are often weak, e.g. Markov Chain models for speech.



Latent/Hidden variable models improve performance significantly. e.g. Hidden Markov models.



Outline

1. Need for interpretability in AI

2. Machine Learning Models
   - Linear
   - Trees
   - Multi-layer Perceptrons
   - CNNs
   - Attention networks
   - RNNs

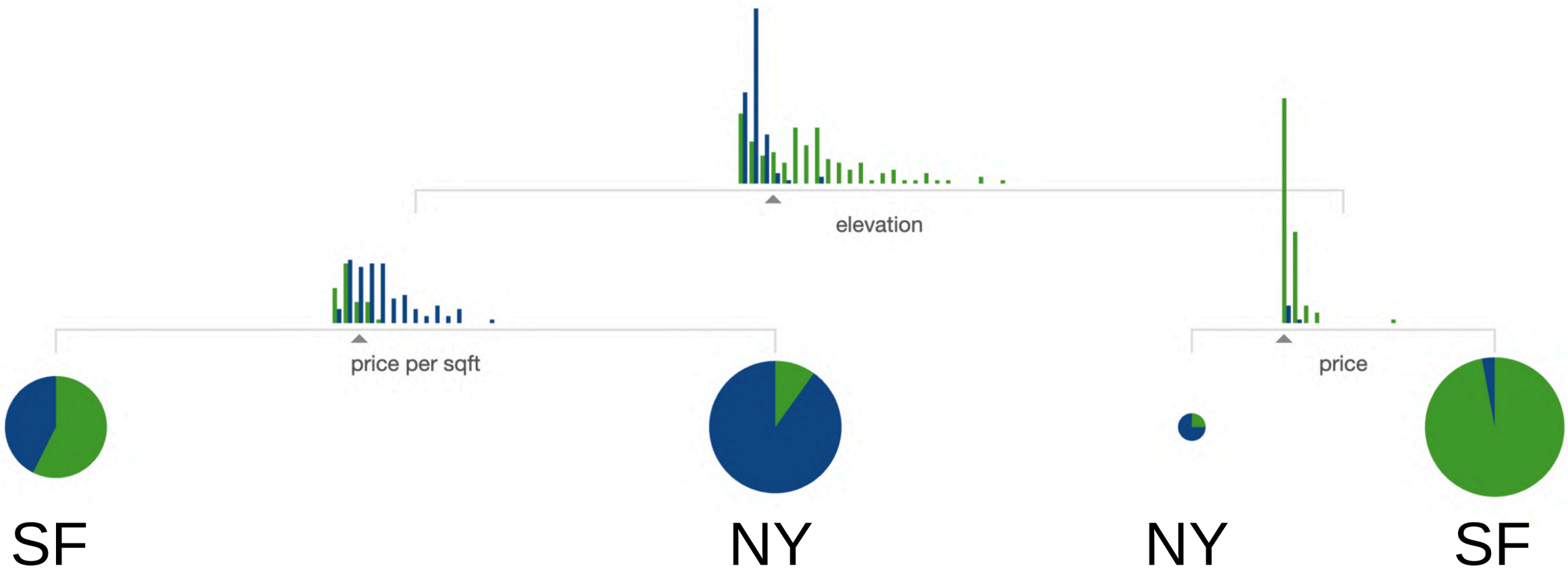
3. Interpretability Paradigms
   - Intrinsic
   - Reverse Engg.
   - Feature visualization
   - Attribution maps
   - Influence functions
   - Other

4. Caveat Emptor

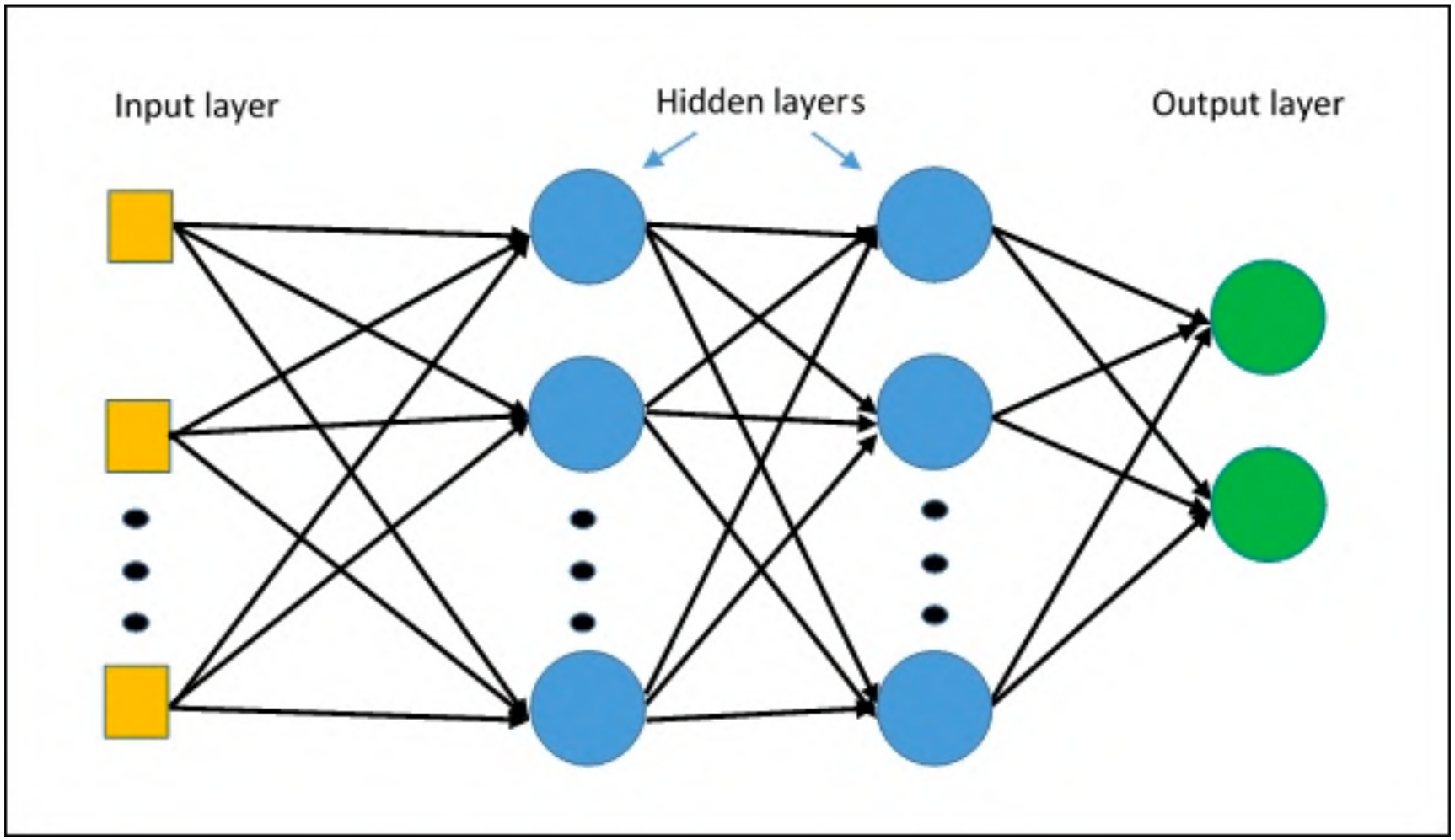
Linear Models

- Output is a linear function of input.
- e.g. Heart Risk = (BP-120)+(Sugar-100)+10(Cholesterol)
- Ultimate interpretability

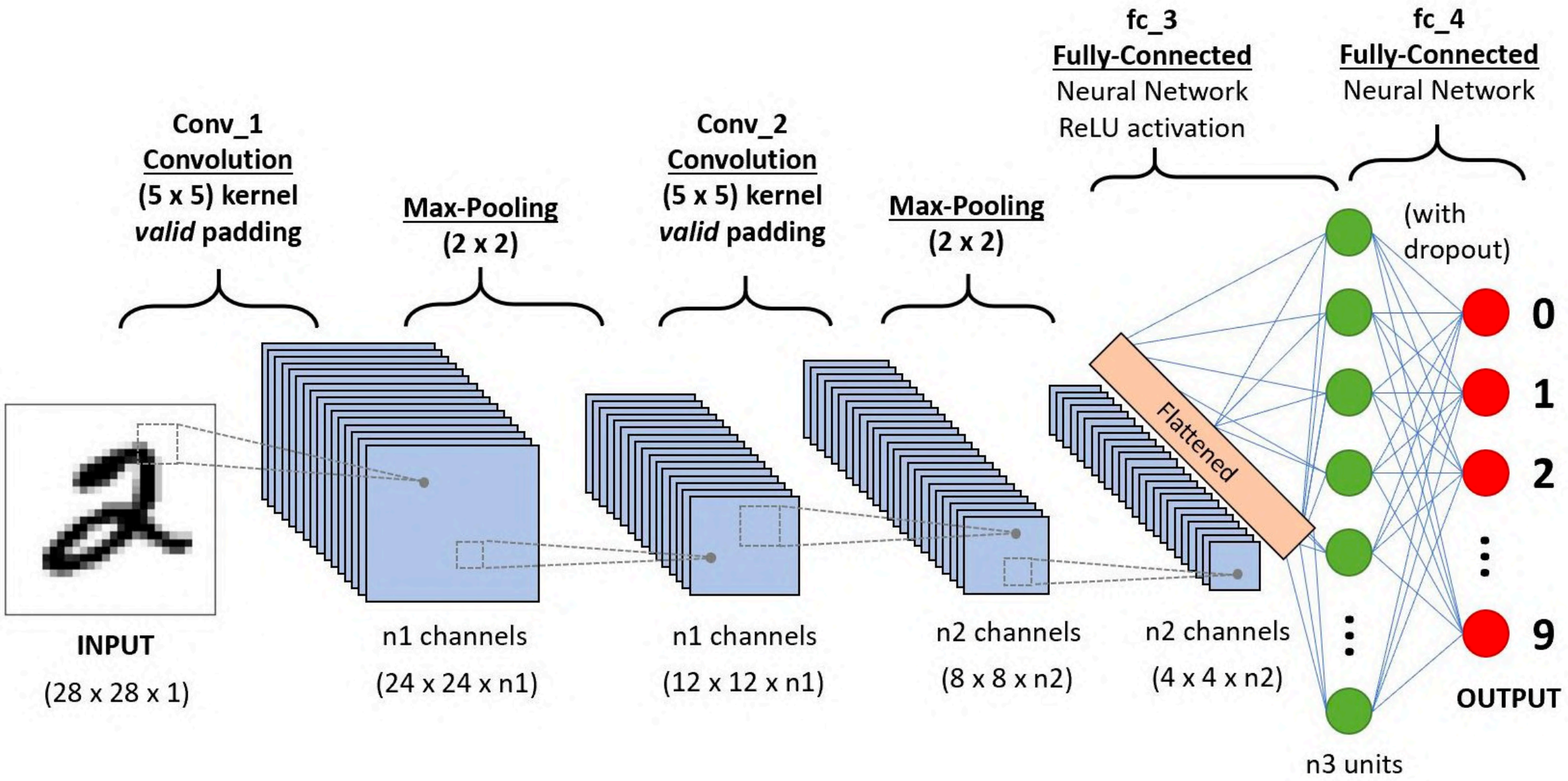
Trees



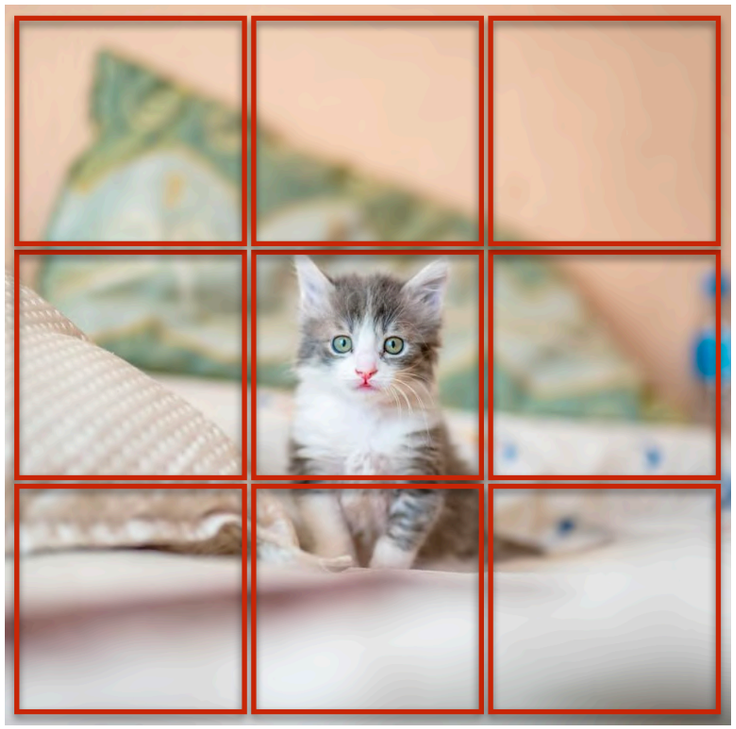
Multilayer Perceptrons



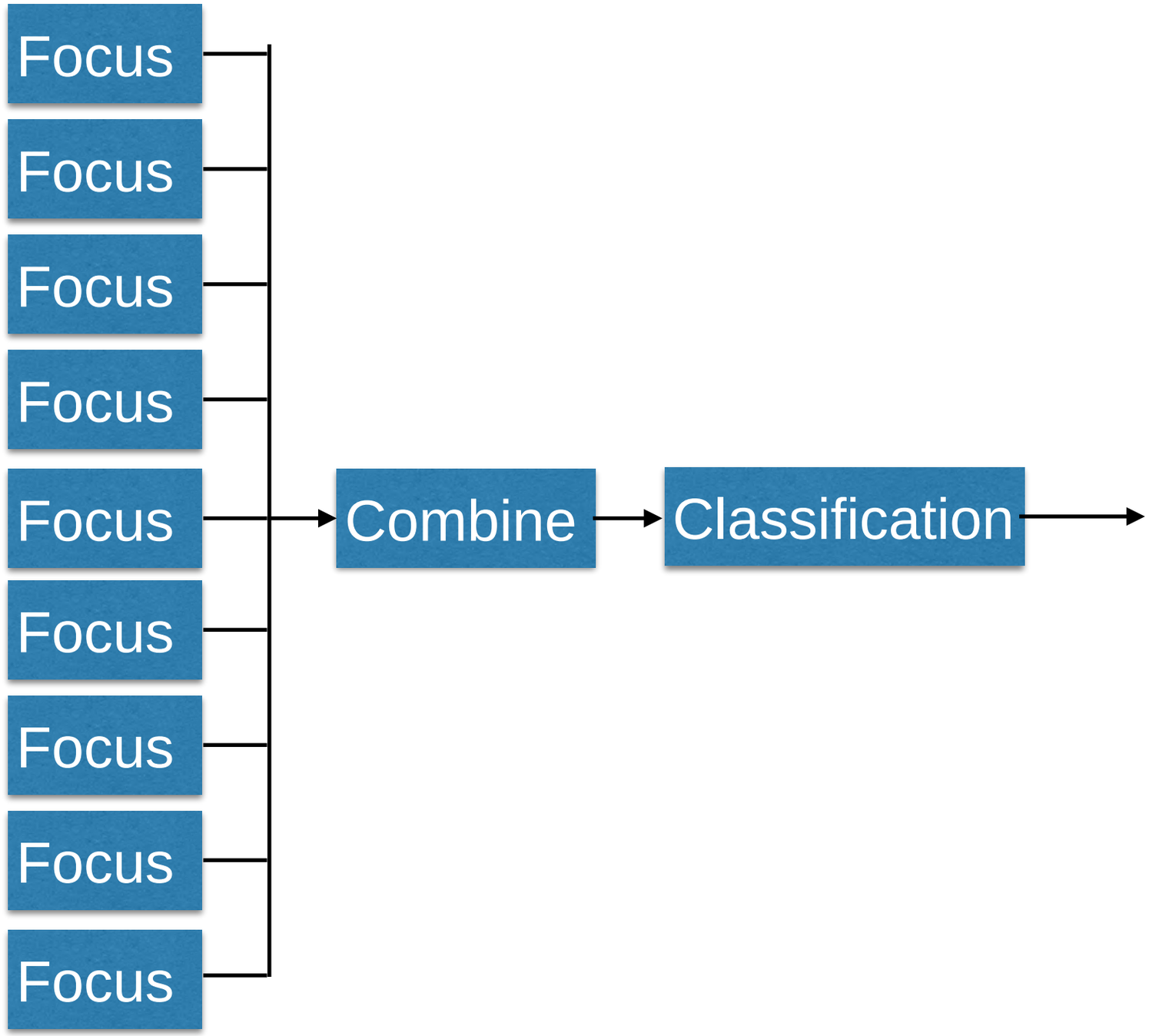
CNNs


Conv_1
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
Conv_2
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
fc_3
Fully-Connected
Neural Network
ReLU activation
fc_4
Fully-Connected
Neural Network
(with
dropout)
Flattened
0
1
2
9
OUTPUT
INPUT
(28 x 28 x 1)
n1 channels
(24 x 24 x n1)
n1 channels
(12 x 12 x n1)
n2 channels
(8 x 8 x n2)
n2 channels
(4 x 4 x n2)
n3 units

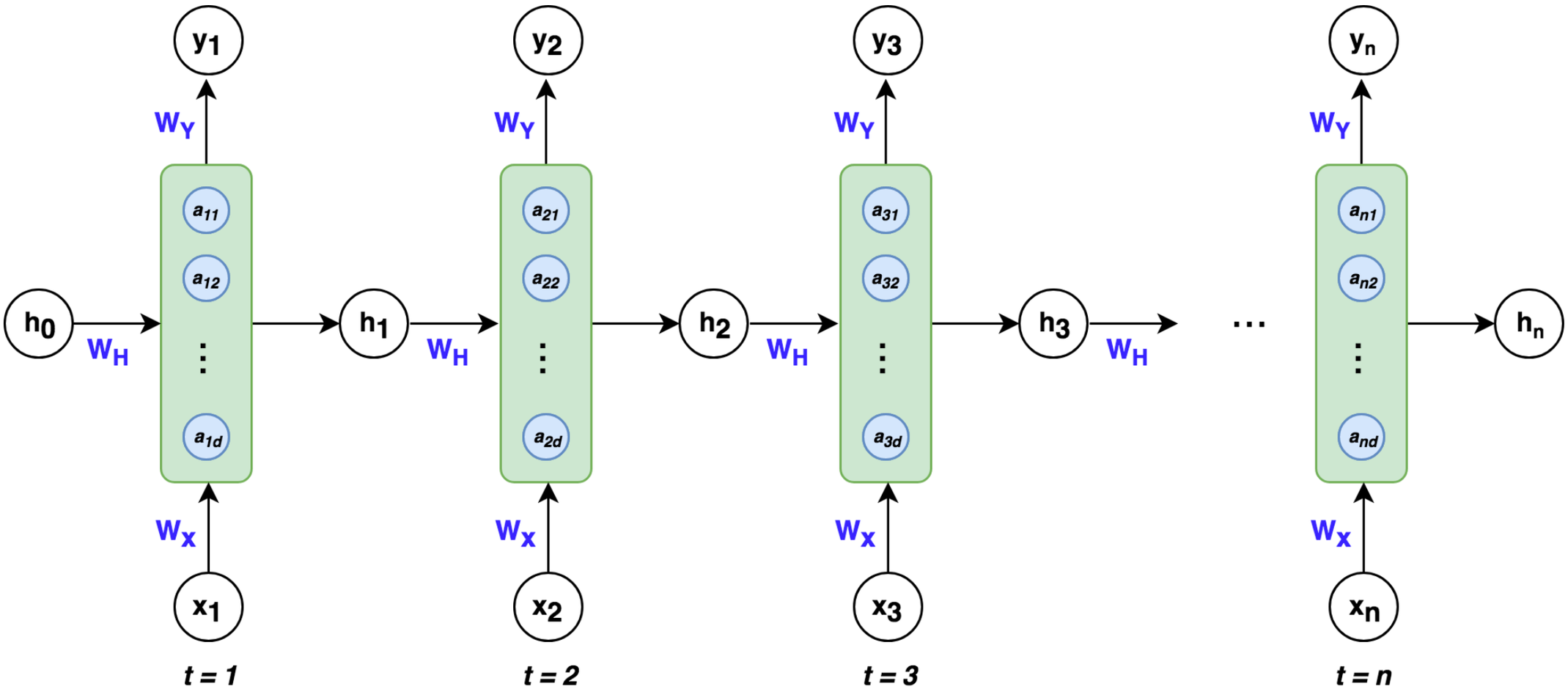
Attention Models



Cat



RNNs


y1
y2
y3
yn
WY
a11
a12
a1d
a21
a22
a2d
a31
a32
a3d
an1
an2
and
h0
h1
h2
h3
hn
WH
...
WX
x1
x2
x3
xn
t = 1
t = 2
t = 3
t = n

Outline

1. Need for interpretability in AI

2. Neural Models
   - Linear
   - Trees
   - Multi-layer Perceptrons
   - CNNs
   - Attention networks
   - RNNs

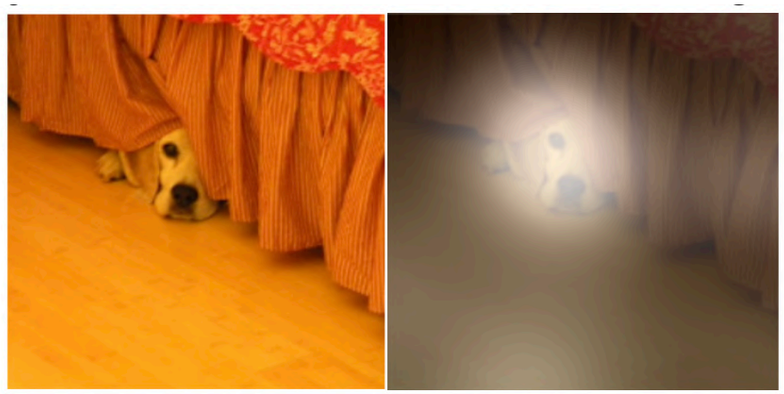
3. Interpretability Paradigms
   - Intrinsic
   - Reverse Engg.
   - Feature visualization
   - Attribution maps
   - Influence functions
   - Training for Interpretability
   - Other

4. Caveat Emptor

Intrinsic Interpretability

- Linear and small depth tree models are intrinsically interpretable.
- Simple attention models are also interpretable to an extent.

A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.

A woman holding a clock in her hand.

Xu et al. (2015)

Reverse Engineering: RNNs

Figuring out the meaning of each element of the state vector:

Cell sensitive to position in line:

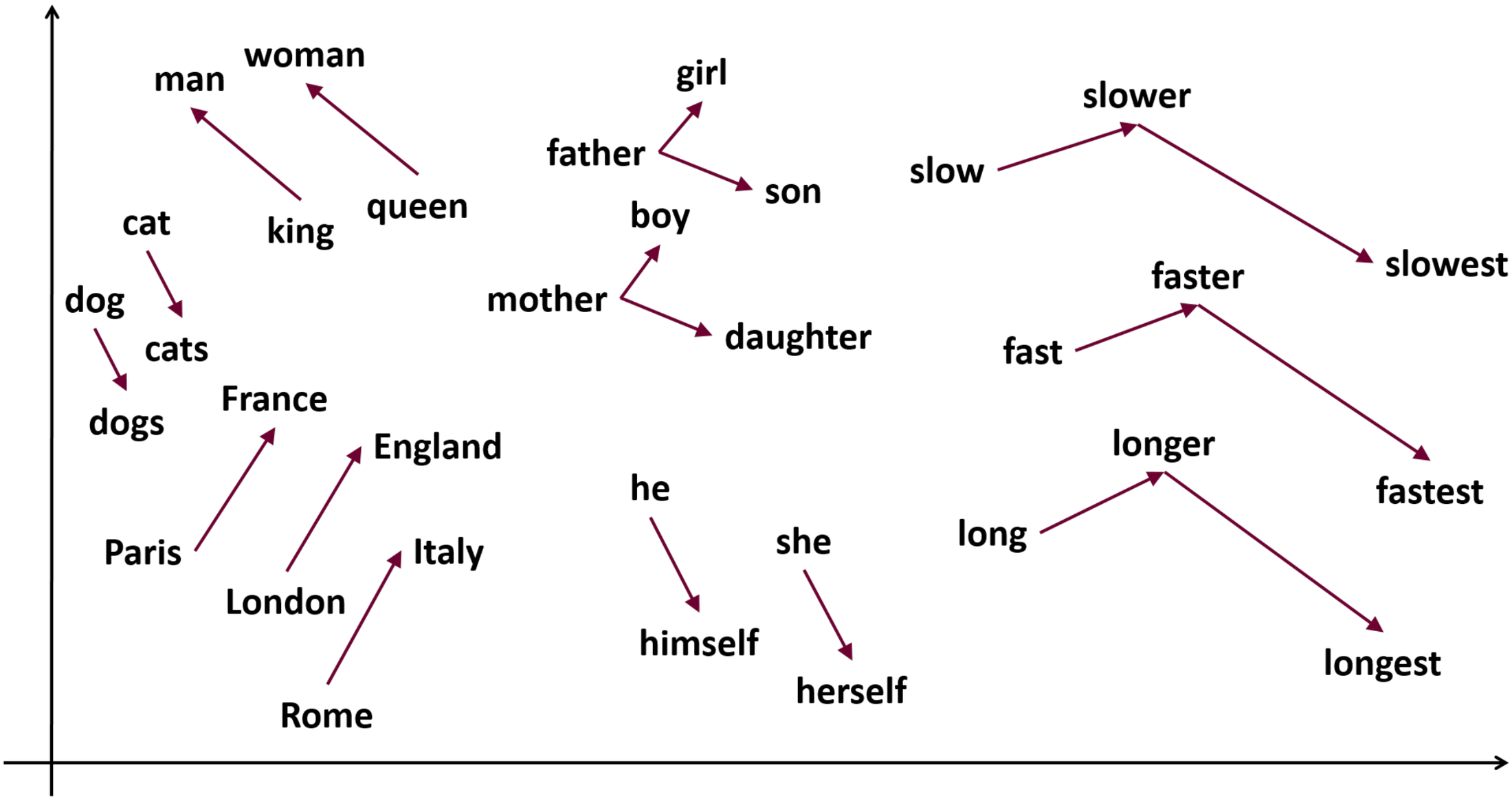
```
The sole importance of the crossing of the Berezina lies in the fact
that it plainly and indubitably proved the fallacy of all the plans for
cutting off the enemy's retreat and the soundness of the only possible
line of action--the one Kutuzov and the general mass of the army
demanded--namely, simply to follow the enemy up. The French crowd fled
at a continually increasing speed and all its energy was directed to
reaching its goal. It fled like a wounded animal and it was impossible
to block its path. This was shown not so much by the arrangements it
made for crossing as by what took place at the bridges. When the bridges
broke down, unarmed soldiers, people from Moscow and women with children
who were with the French transport, all--carried on by vis inertiae--
pressed forward into boats and into the ice-covered water and did not,
surrender.
```

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
        siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

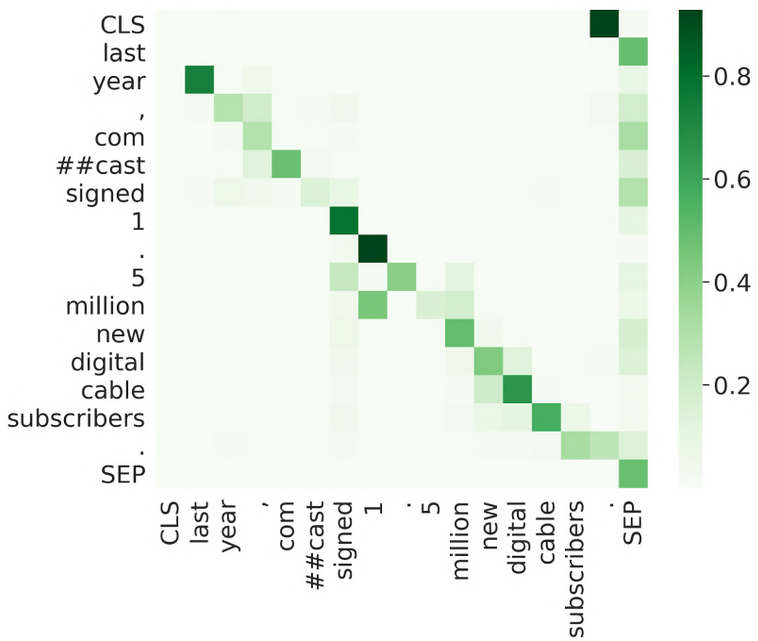
Karpathy et al. 2016

Reverse Engineering: Word2Vec

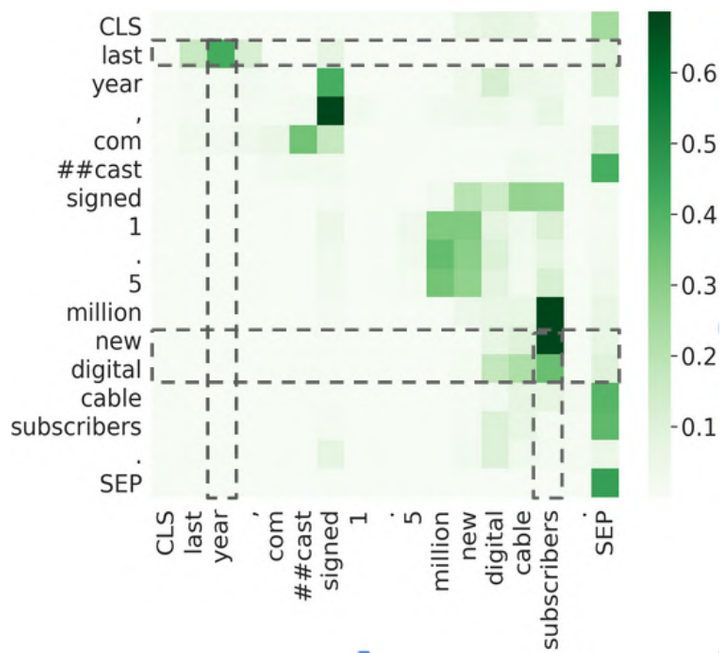


Reverse Engineering: The Heads of BERT

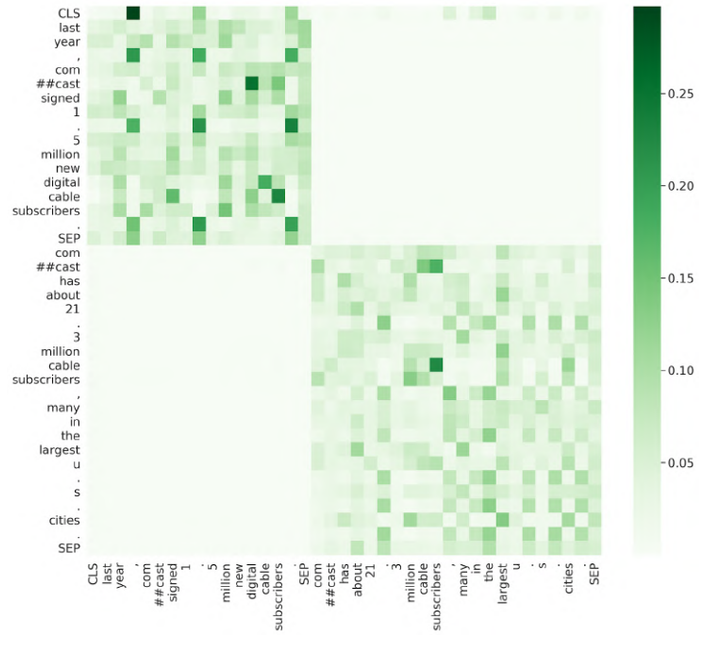
Local



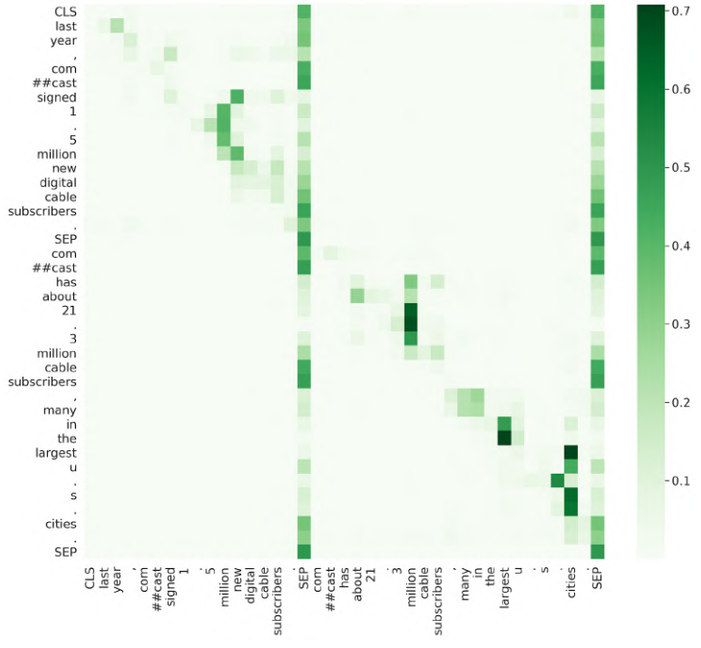
Syntactic



Block

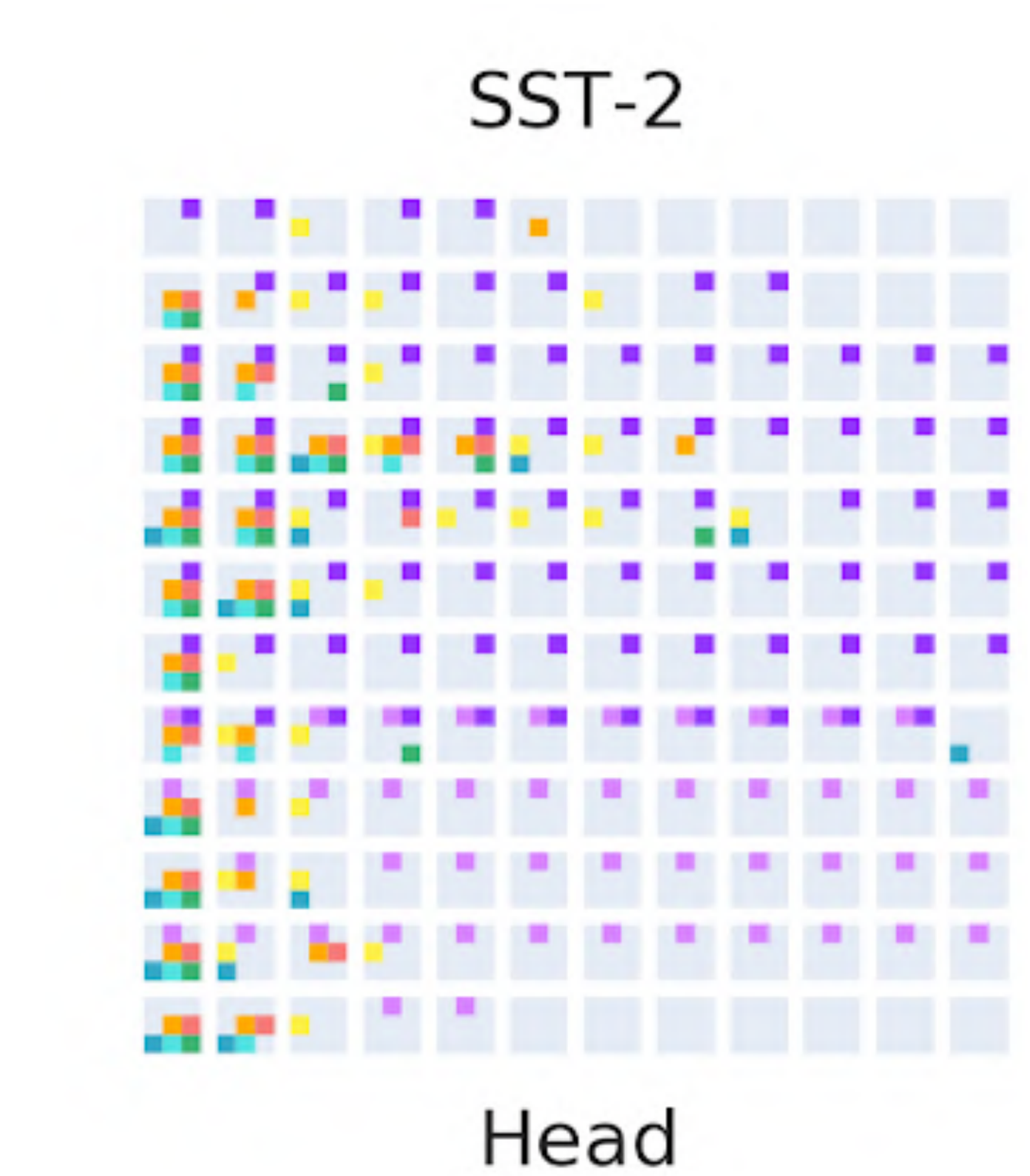


Delimiter



Pande et al. 2020

Reverse Engineering: The Heads of BERT



Block	CLS	SEP
dobj	Amod	advm
Local	Syntax	Nsubj

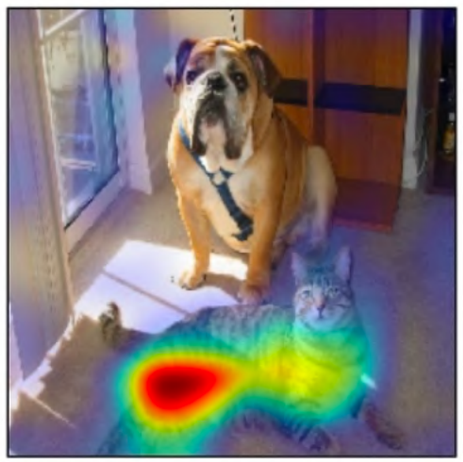
Pande et al. 2020

CNN Visualisation:

1. Black Box Approaches.
    1. Saliency
    2. Occlusion
    3. Class Activation Maps.
2. Optimising the Model
    1. Feature visualisation
    2. Other

Attribution Maps: CAM/GradCAM

Original image

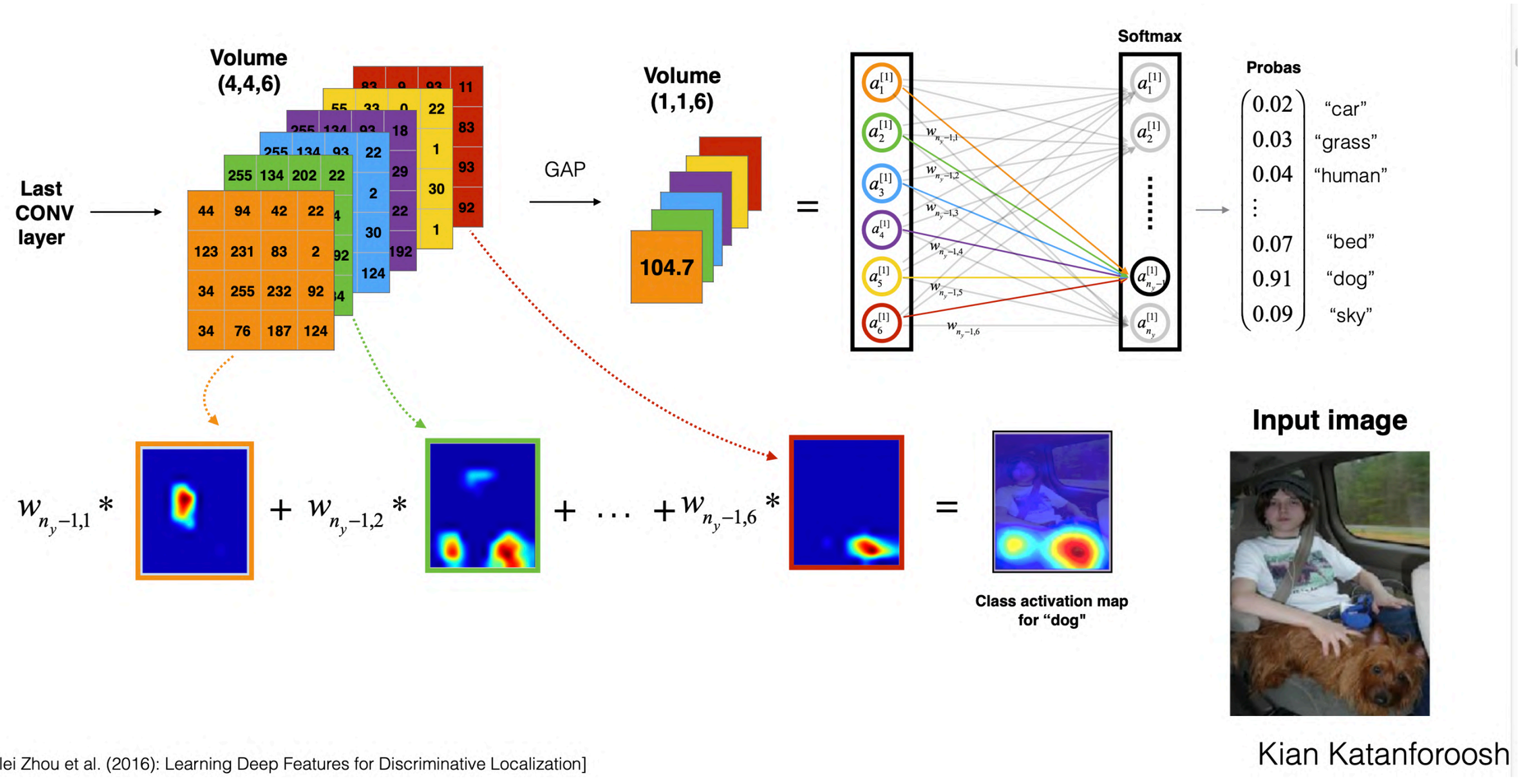


GradCAM: cat

GradCAM: dog

Selvaraju et al. (2017)

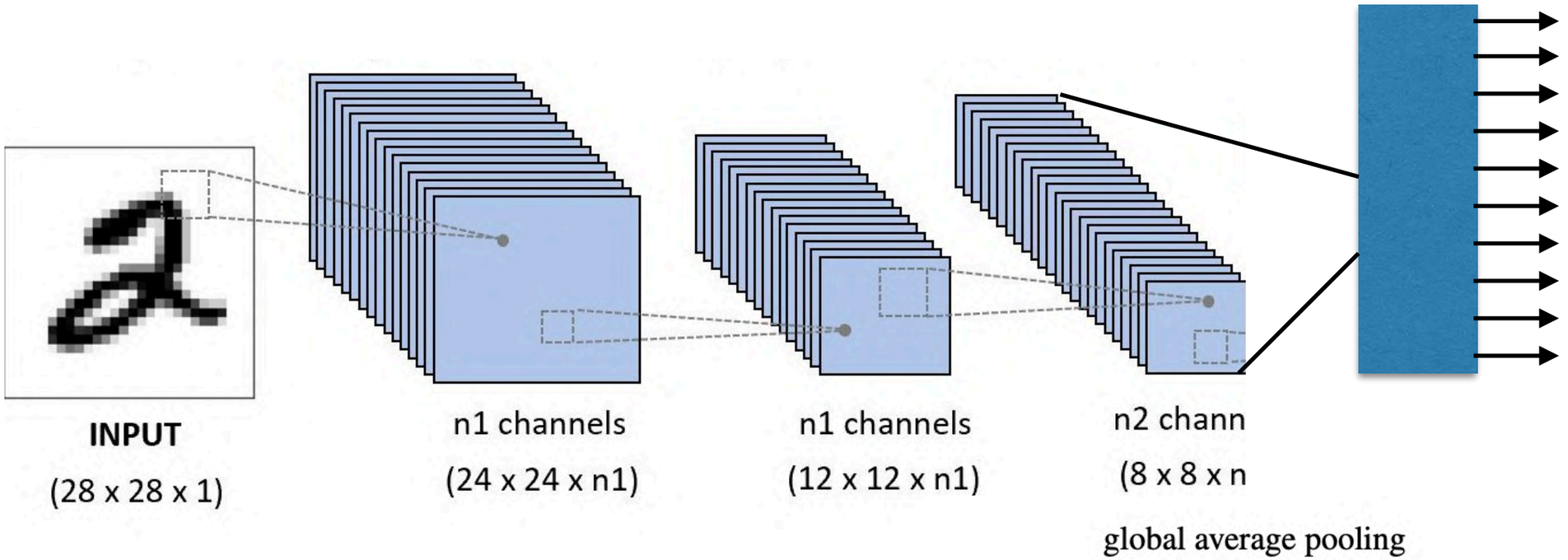
Attribution Maps: CAM/GradCAM



[Bolei Zhou et al. (2016): Learning Deep Features for Discriminative Localization]

Kian Katanforoosh

Attribution Maps : Grad CAM

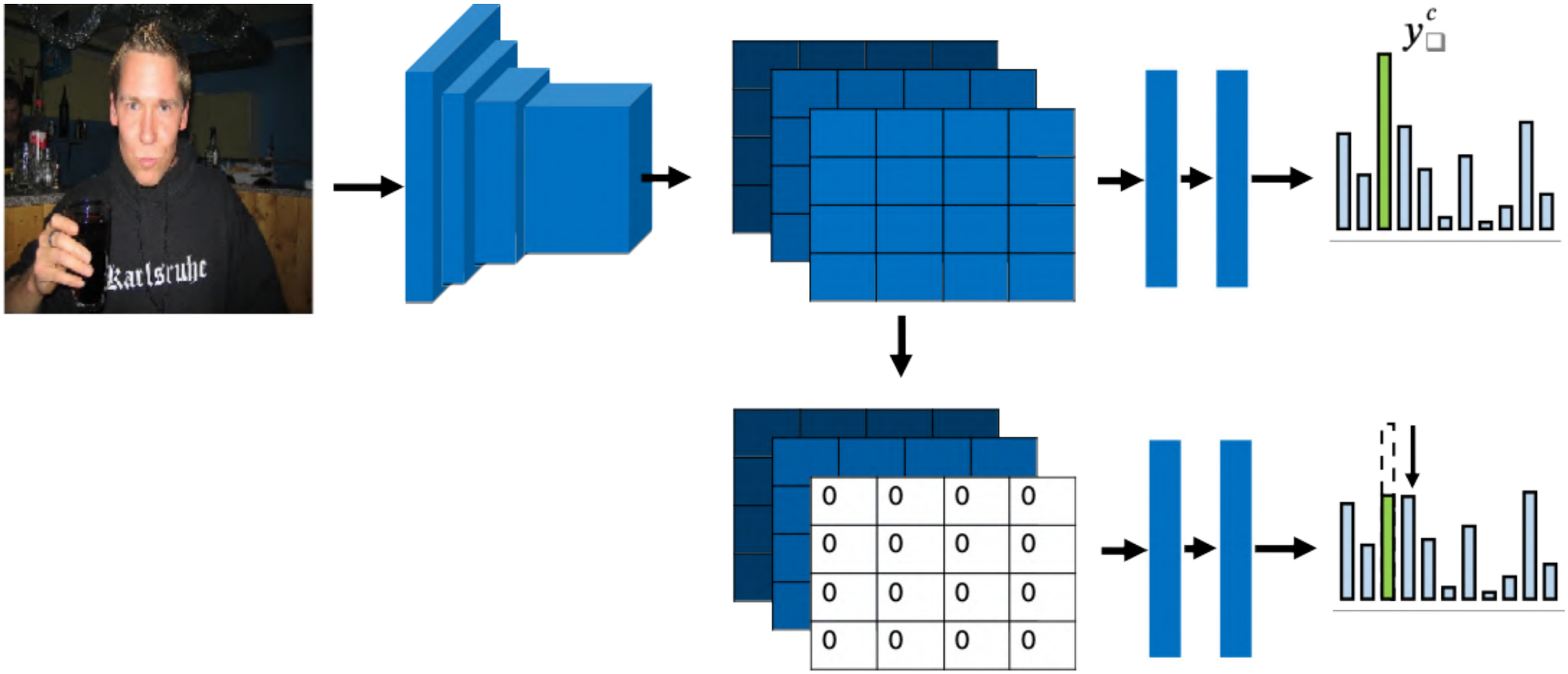


Importance of channel ‘k’ for class ‘c’ = $\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Selvaraju et al. (2017)

Attribution Maps : Ablation CAM

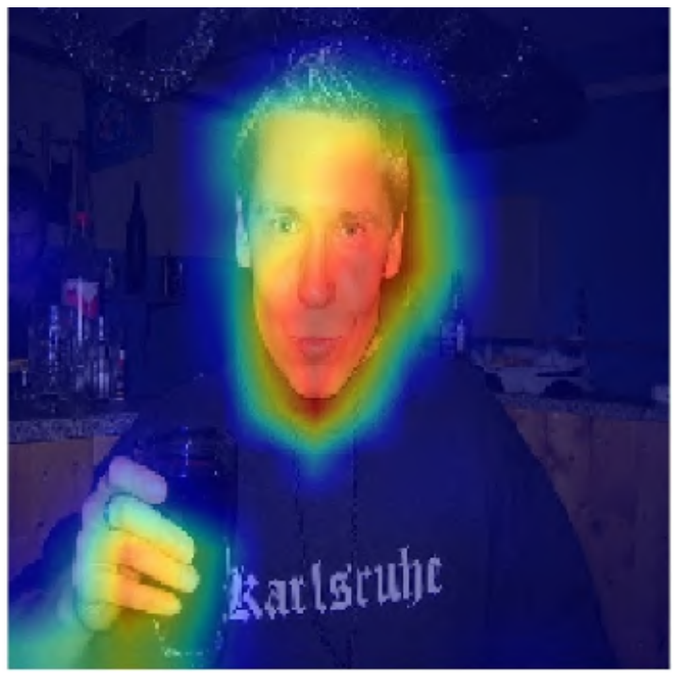


Desai et al. (2020)

Attribution Maps: AblationCAM

Image of person

Grad-CAM for “person”



Ablation-CAM for “person”

Desai et al. (2020)

Feature Viz in CNNs

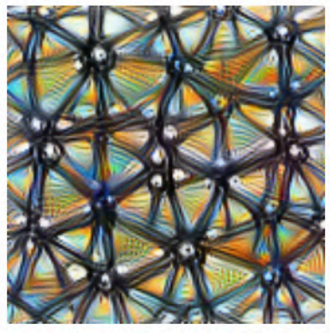
Visualize a learned filter by finding an artificial image that triggers it.

Different **optimization objectives** show what different parts of a network are looking for.

- `n` layer index
- `x,y` spatial position
- `z` channel index
- `k` class index

Neuron

`layer`$_n$`[x,y,z]`



Channel

`layer`$_n$`[:,:,z]`

Layer/DeepDream

`layer`$_n$`[:,:,:]`^{2}

Class Logits

`pre_softmax[k]`

Class Probability

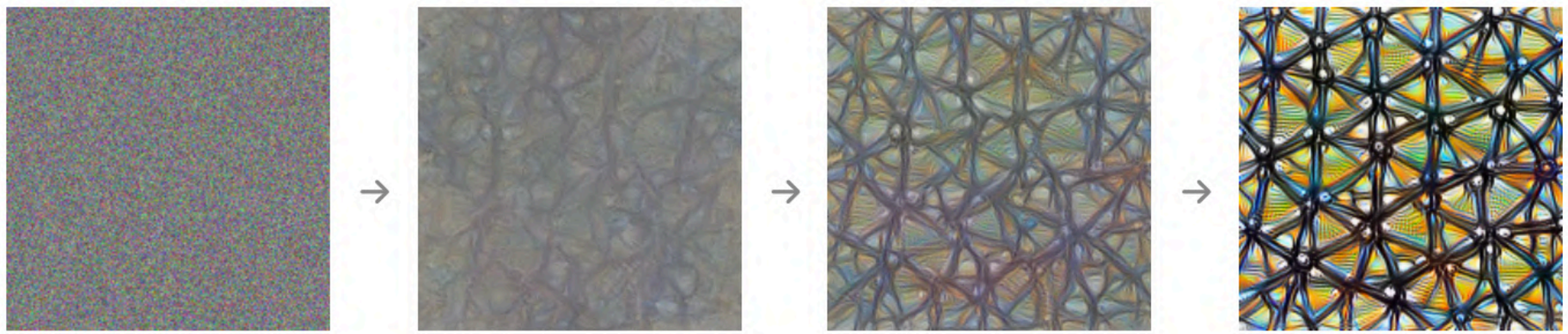
`softmax[k]`

Olah et al. 2017

Feature Viz in CNNs

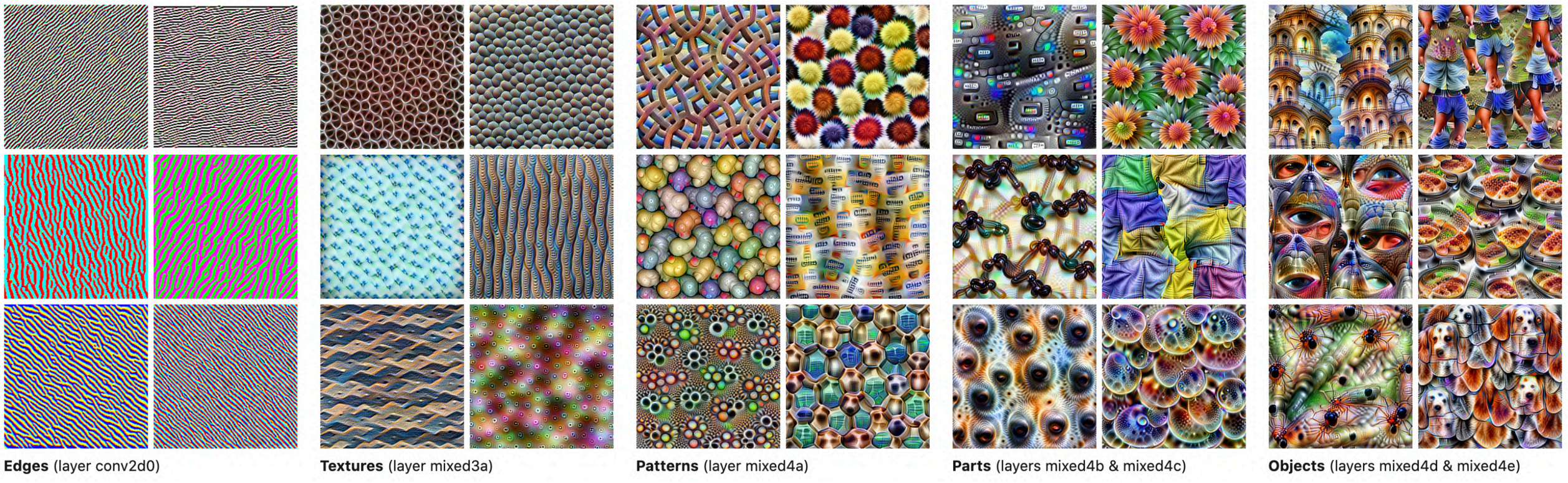
Visualize a learned filter by finding an artificial image that triggers it.

Starting from random noise, we optimize an image to activate a particular neuron (layer mixed4a, unit 11).



Olah et al. 2017

Feature Viz in CNNs



Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

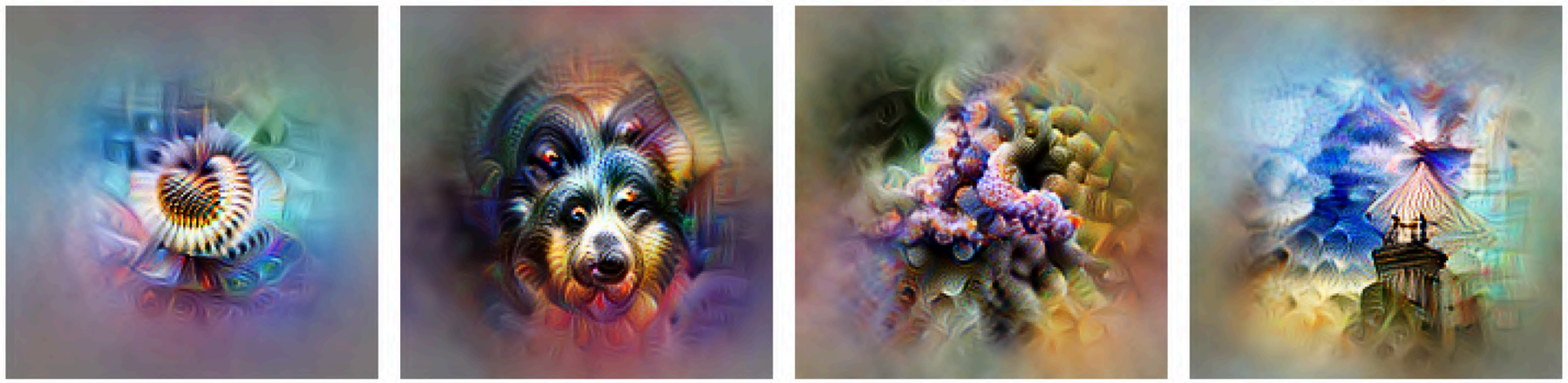
Olah et al. 2017

Feature Viz in CNNs

Why optimize over hallucinated images?

Dataset Examples show us what neurons respond to in practice

Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.



Baseball—or stripes?
mixed4a, Unit 6

Animal faces—or snouts?
mixed4a, Unit 240

Clouds—or fluffiness?
mixed4a, Unit 453

Buildings—or sky?
mixed4a, Unit 492

Olah et al. 2017

Interpretability for CNNs

https://distill.pub/2018/building-blocks/

Olah et al. 2018

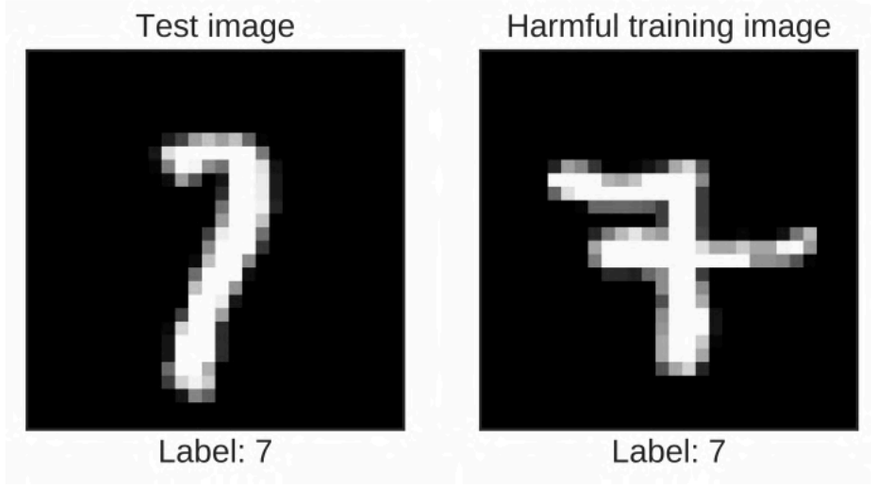
Influence Functions

- Can we explain predictions using training data?
- “ How would the model’s predictions change on a given test point, if we did not have a given training point?”
- Remove/Perturb/Repeat a sample and retrain!
- Influence functions: A more efficient approach for the same.

Koh and Liang (2017)

Influence Functions: Applications

Most harmful training image for a wrong prediction



Most useful training image for a right prediction

Koh and Liang (2017)

Surrogate Model Learning

1. Select a dataset X. This can be the same dataset that was used for training the black box model or a new dataset from the same distribution. You could even select a subset of the data or a grid of points, depending on your application.

2. For the selected dataset X, get the predictions of the black box model.

3. Select an interpretable model type (linear model, decision tree, …).

4. Train the interpretable model on the dataset X and its predictions.

5. Congratulations! You now have a surrogate model.

6. Measure how well the surrogate model replicates the predictions of the black box model.

7. Interpret the surrogate model.

Molnar (2020)

Surrogate Model Learning

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_*^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^{n}(\hat{y}^{(i)} - \bar{\hat{y}})^2}$$

$\hat{y}_*^{(i)}$ — Simple model prediction

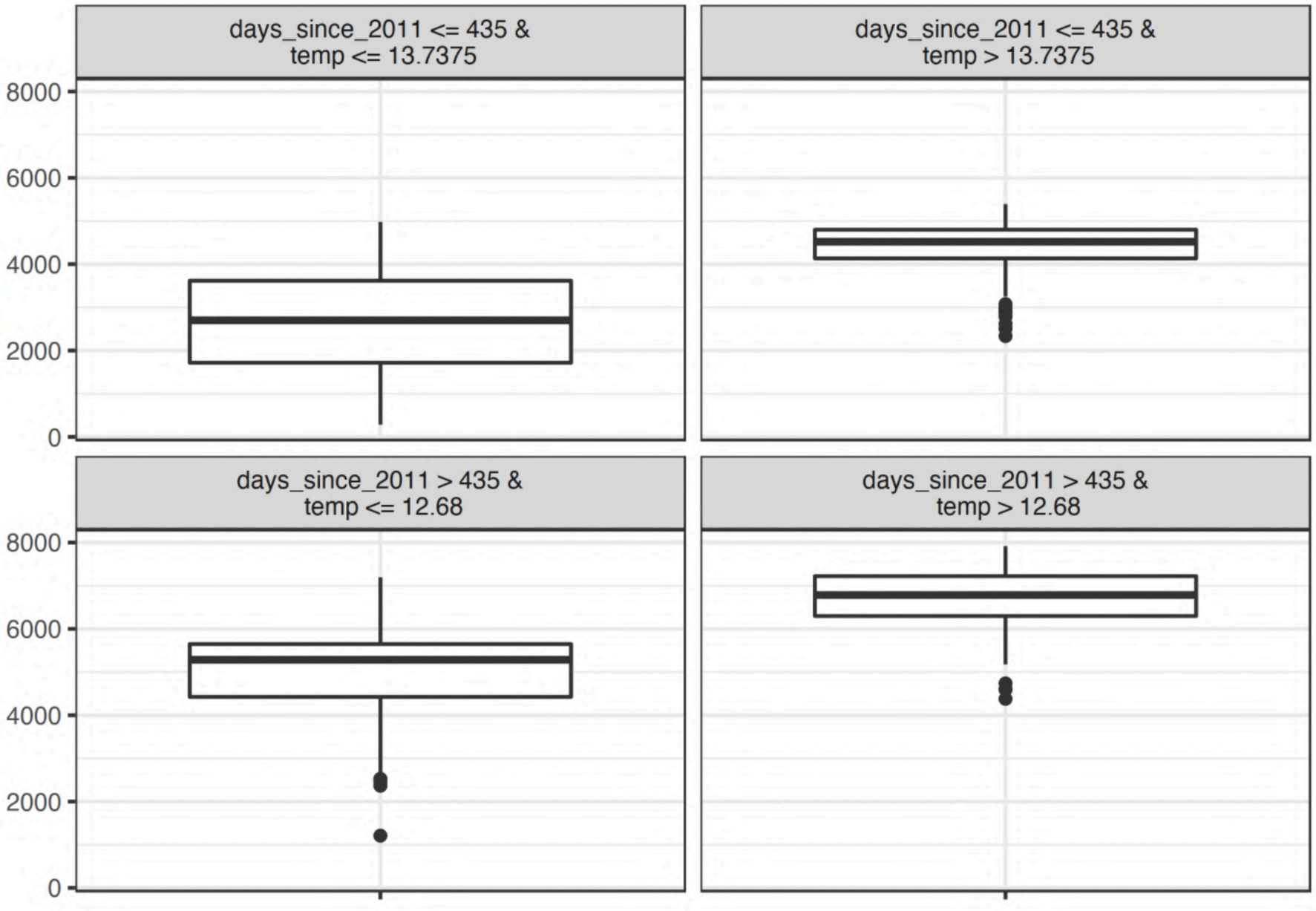
$\hat{y}^{(i)}$ — Complex model prediction

R^2 captures how much better the simple model is at explaining the complex model, when compared to a constant.

Molnar (2020)

Surrogate Model Learning

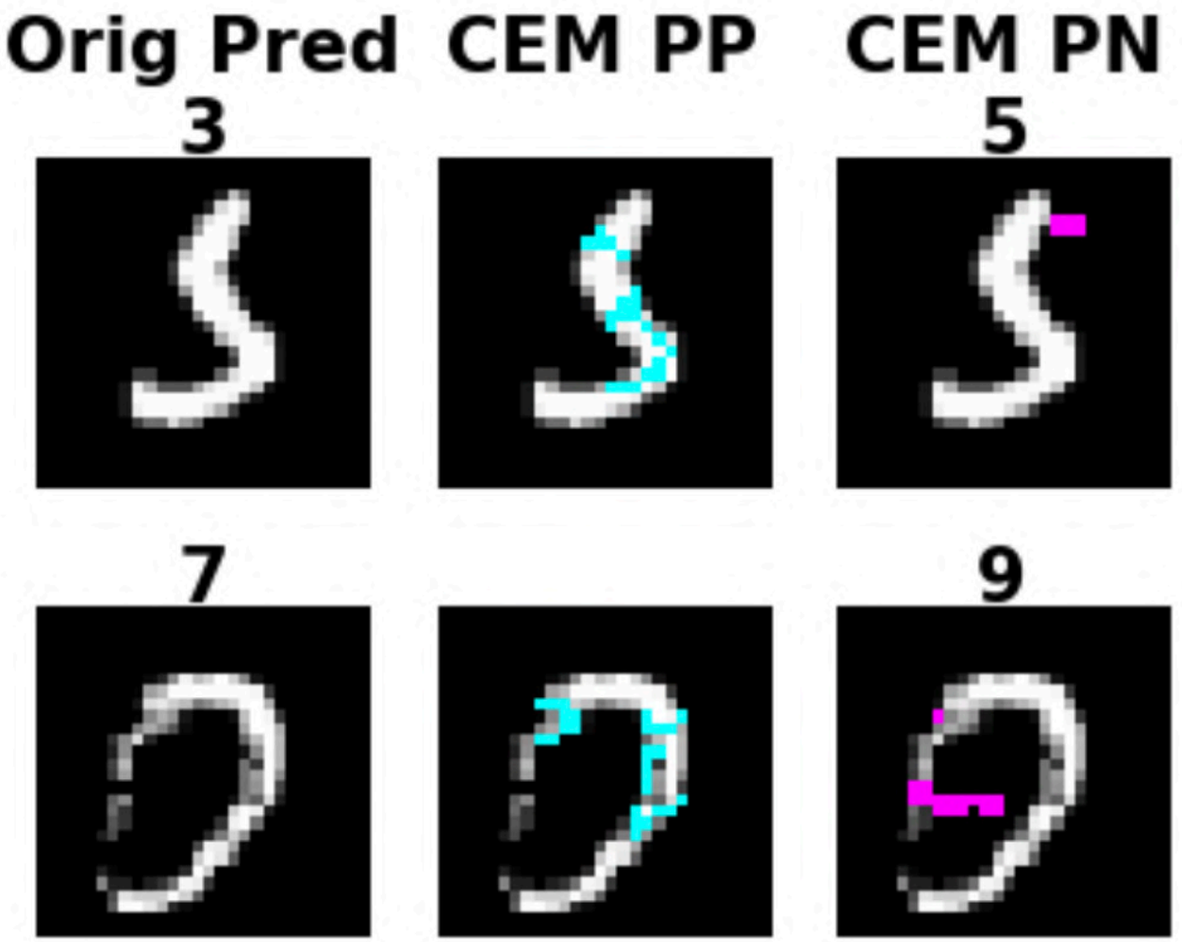
Example: Explain a complex SVM model for predicting daily number of rented bikes using a regression tree.



R^2=0.77

Molnar (2020)

Pertinent Positives and Negatives



Dhurandhar et al (2018)

Pertinent Positives and Negatives

Pertinent negatives:

$$\min_{\boldsymbol{\delta} \in \mathcal{X}/\mathbf{x}_0} \quad c \cdot f_\kappa^{\text{neg}}(\mathbf{x}_0, \boldsymbol{\delta}) + \beta \|\boldsymbol{\delta}\|_1 \,.$$

Pertinent positive:

$$\min_{\boldsymbol{\delta} \in \mathcal{X} \cap \mathbf{x}_0} \quad c \cdot f_\kappa^{\text{pos}}(\mathbf{x}_0, \boldsymbol{\delta}) + \beta \|\boldsymbol{\delta}\|_1$$

Dhurandhar et al (2018)

Outline

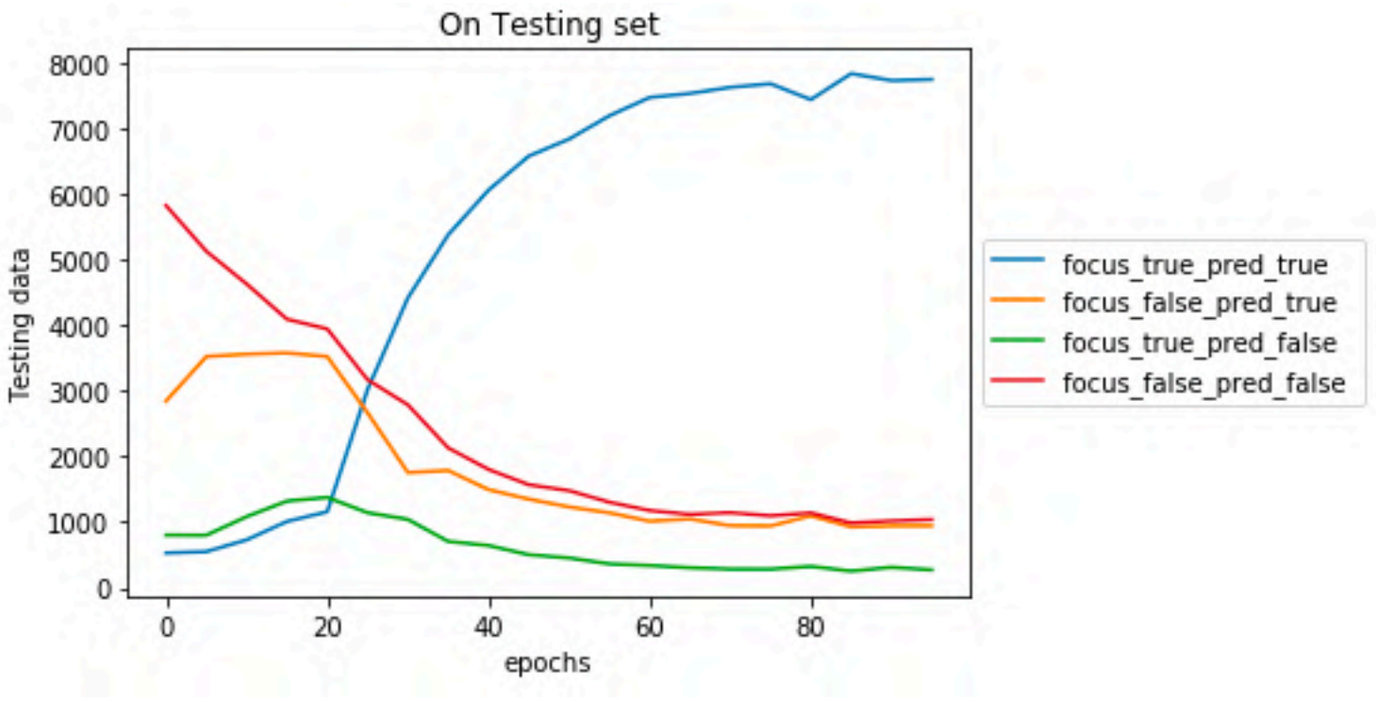
1. Need for interpretability in AI

2. Neural Models
   - Linear
   - Trees
   - Multi-layer Perceptrons
   - CNNs
   - Attention networks
   - RNNs

3. Interpretability Paradigms
   - Intrinsic
   - Reverse Engg.
   - Feature visualization
   - Attribution maps
   - Influence functions
   - Other

4. Caveat Emptor

Phantoms in the Mind



References:

- Chris Olah’s blog and Distill posts: colah.github.io
- Christoph Molnar. *Interpretable Machine Learning.*
- Karpathy et al. *Visualizing and understanding RNNs.* ICLR 2016.
- Kian Katanfaroosh. Stanford Interpretability Lecture.
- Xu et al. *Show, Attend and Tell.* ICML 2015.
- Koh and Liang. *Understanding Black-box Predictions via Influence Functions.* ICML 2017.
- Selvaraju et al. GradCAM. ICCV 2017.