Advanced Visualization Methods in Deep Learning

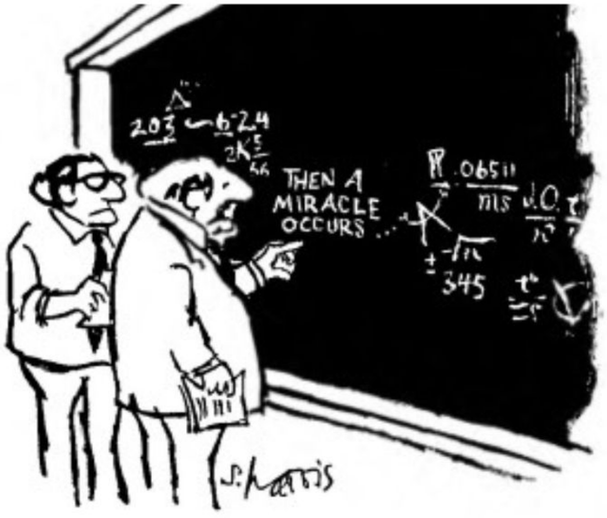
Ester Hlav

Columbia University

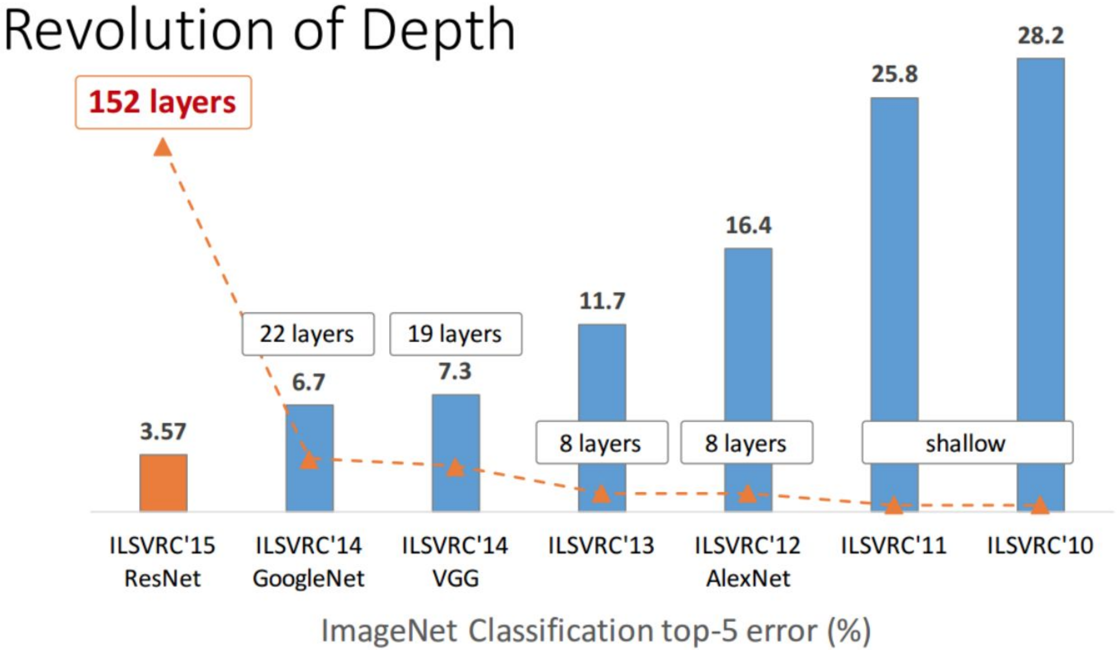
E6040 Deep Learning Research

February 2018

Tradeoff Complexity vs Interpretability in NNs

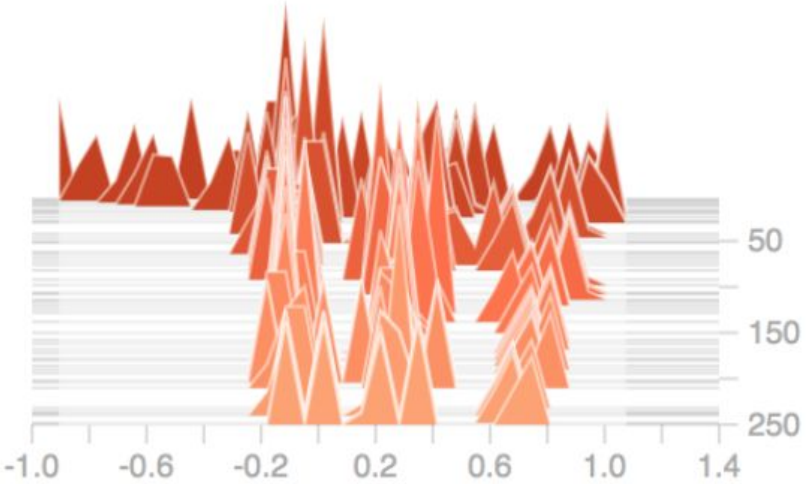


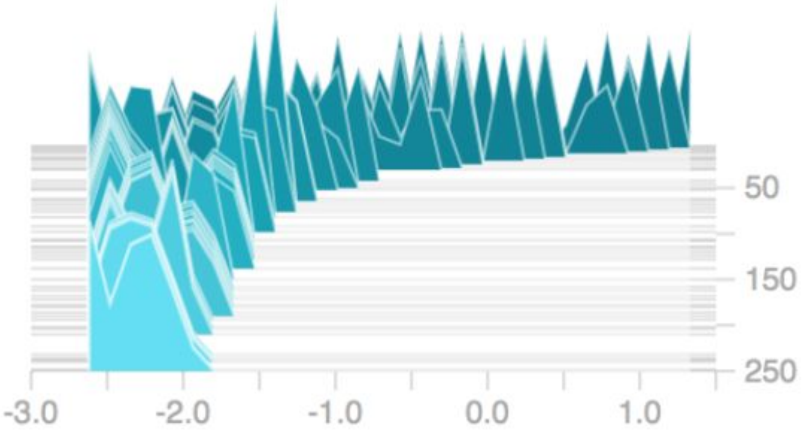
"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

TensorBoard





Outline

I. **CNN Visualization** - Computer Vision (image)
   - **Sanity Checks for Saliency Maps** by Julius Adebayo , Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim, 2018

II. **RNN Visualization** - Natural Language Processing (text)
   - **LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks** by Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Sasha Rush, 2018

III. **Explainability** for Neural Networks (generic features)
   - **A Unified Approach to Interpreting Model Predictions** by Scott Lundberg and Su-In Lee, 2017

I. CNN Visualization Methods for Images

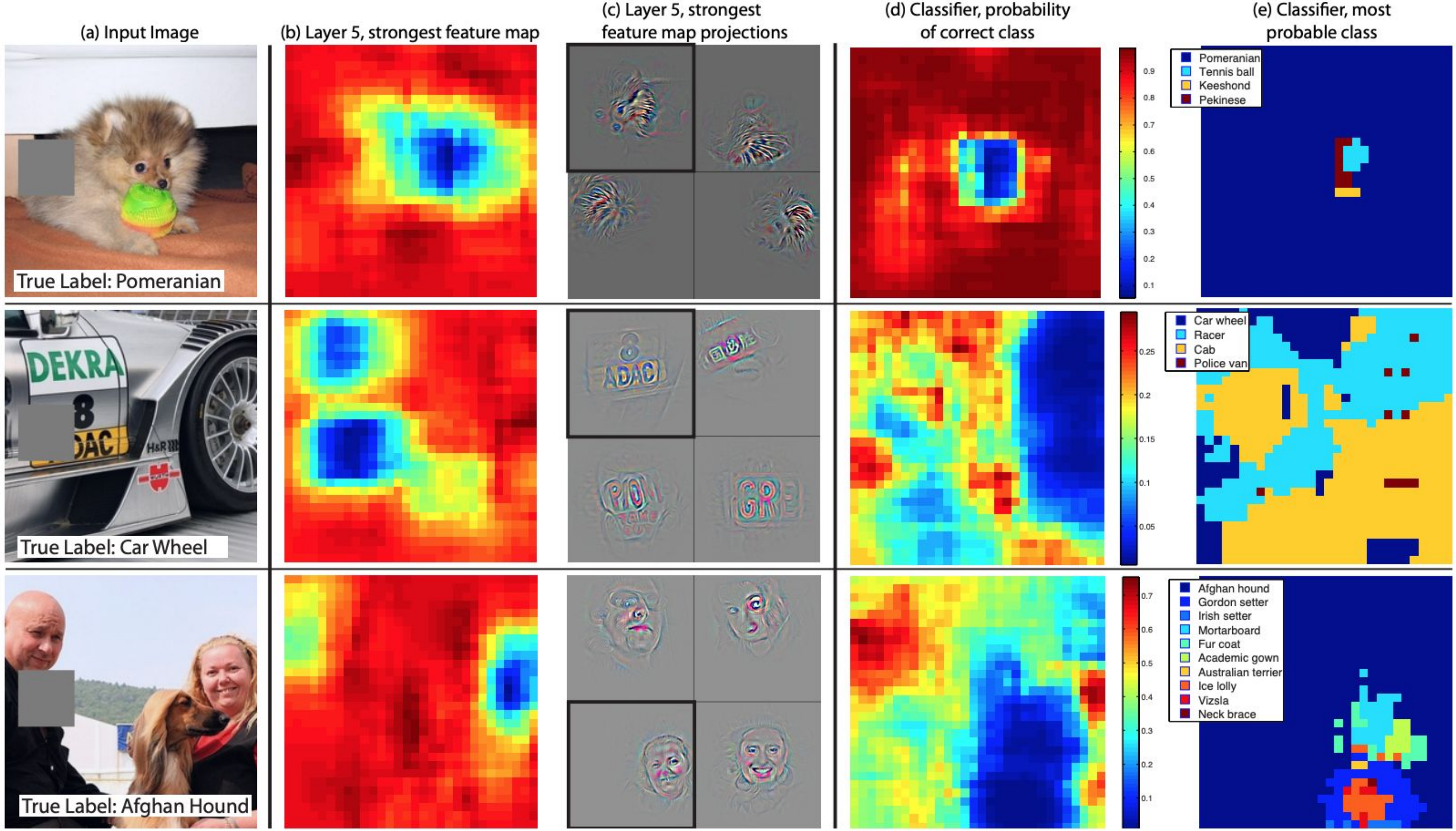
A. Filter (kernel) Visualization

- *Paper*: Visualizing and Understanding Convolutional Networks, 2013
- *Method*: **plot kernels as images**

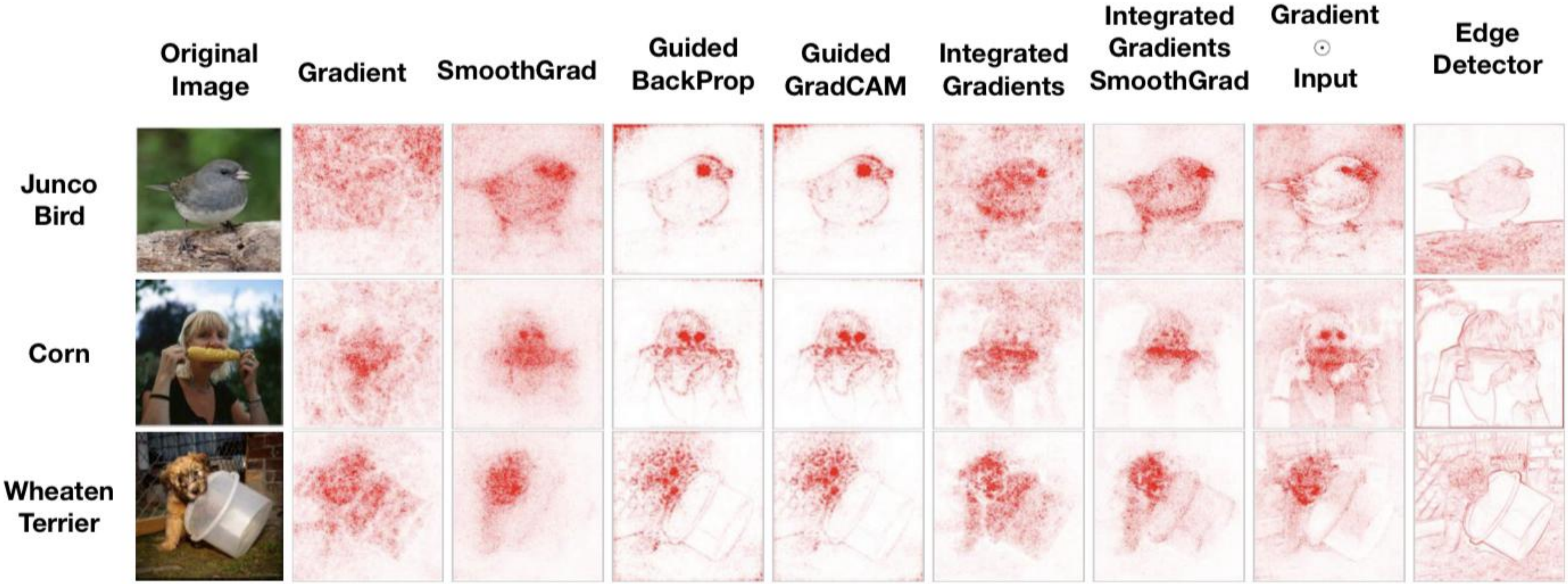
B. Occlusion Patches

- *Paper*: Visualizing and Understanding Convolutional Networks, 2013
- *Method*: **monitors the output of the classifier by systematically occluding different portions of the input image with a grey square (zeros)**
  - occluding the true object within the scene → probability of the correct class drops

Occlusion Patches



Comparison for Sanity Checks - Saliency Maps generated by Visualization Methods



J. Adebayo, B. Kim, I. Goodfellow, J. Gilmer, and M. Hardt. Sanity checks for saliency maps. In Proceedings of Advances in Neural Information Processing Systems, 2018.

I. CNN Visualization Methods for Images

C. Saliency Map (Gradient)

- *Paper*: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, 2014
- *Method*: **computes local sensitivity of the output class with regard to each pixel** $E_{\text{grad}}(x) = \frac{\partial S}{\partial x}$

D. SmoothGrad

- *Paper*: SmoothGrad: Removing Noise by Adding Noise, 2017
- *Method*: **take an image of interest, sample similar images by adding noise to the image, then take the average of the resulting sensitivity maps for each sampled image**
  - $E_{\text{sg}}(x) = \frac{1}{N} \sum_{i=1}^{N} E_{\text{grad}}(x + g_i)$ where $g_i \sim \mathcal{N}(0, \sigma^2)$
  - tends to reduce visual noise

E. Gradient ⊙ Input

- *Method*: **elementwise multiplication of the gradient by input** $x \odot \frac{\partial S}{\partial x}$

I. CNN Visualization Methods for Images

F. Guided BackProp

- *Paper*: Striving for Simplicity: The All Convolutional Net, 2015
- *Method*: **computes truncated gradients by backpropagating only positive gradients (i.e. guided)**

G. Integrated Gradient

- *Paper:* Axiomatic Attribution for Deep Networks, 2017
- *Method*: **integrates gradients between a reference input and the considered input**

$$\text{IntegratedGrads}_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \, d\alpha$$

$$\approx \; (x_i - x'_i) \times \Sigma_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x')))}{\partial x_i} \times \frac{1}{m}$$

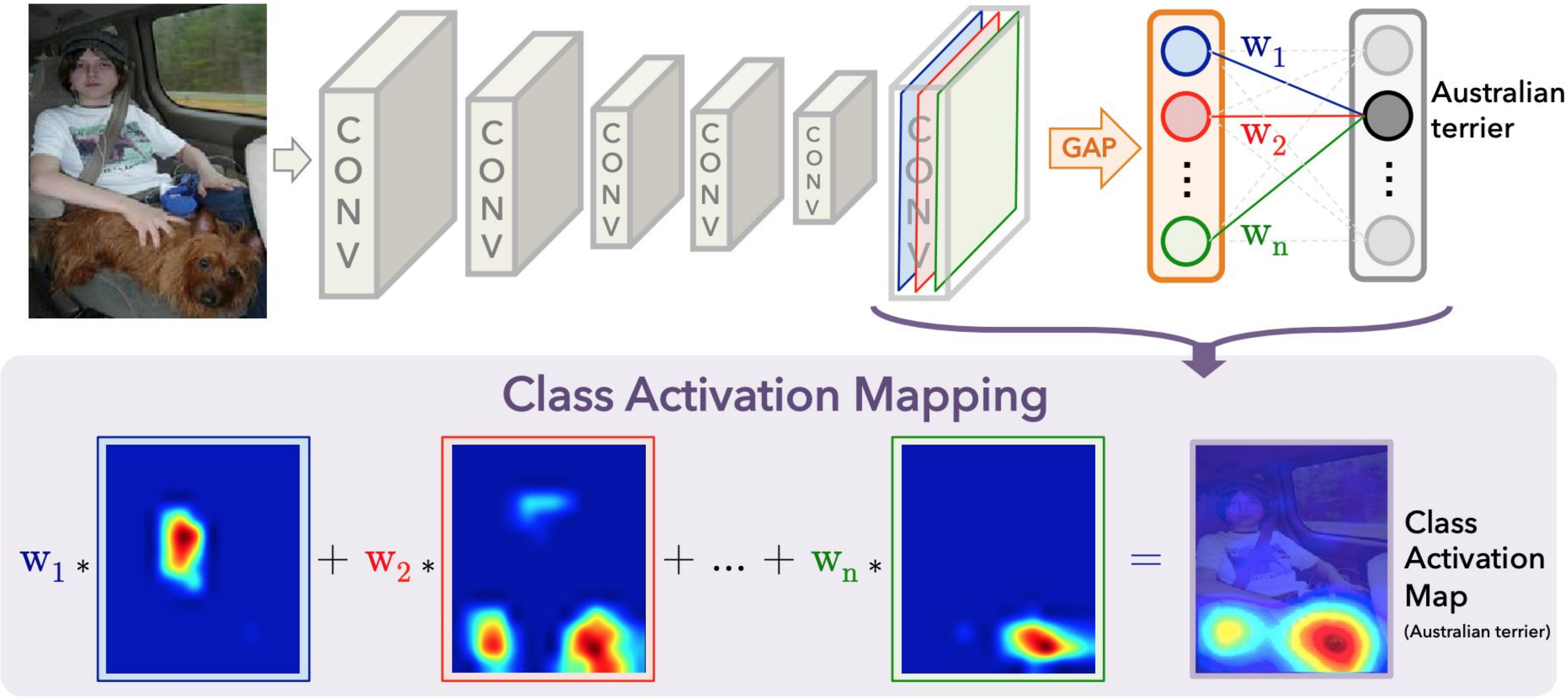
I. CNN Visualization Methods for Images

H. Class Activation Maps (CAM)

- *Paper*: Learning Deep Features for Discriminative Localization, 2016
- *Method*: **weighted average of filters of last conv layer by weights of logits post global average pooling (GAP) of output class**

J. Grad-CAM

- *Paper*: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, 2017
- *Method*: **gradients of logits of output class with regard to feature map of last conv layer**



B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In CVPR, 2016

Sanity Checks for Saliency Maps - Methods

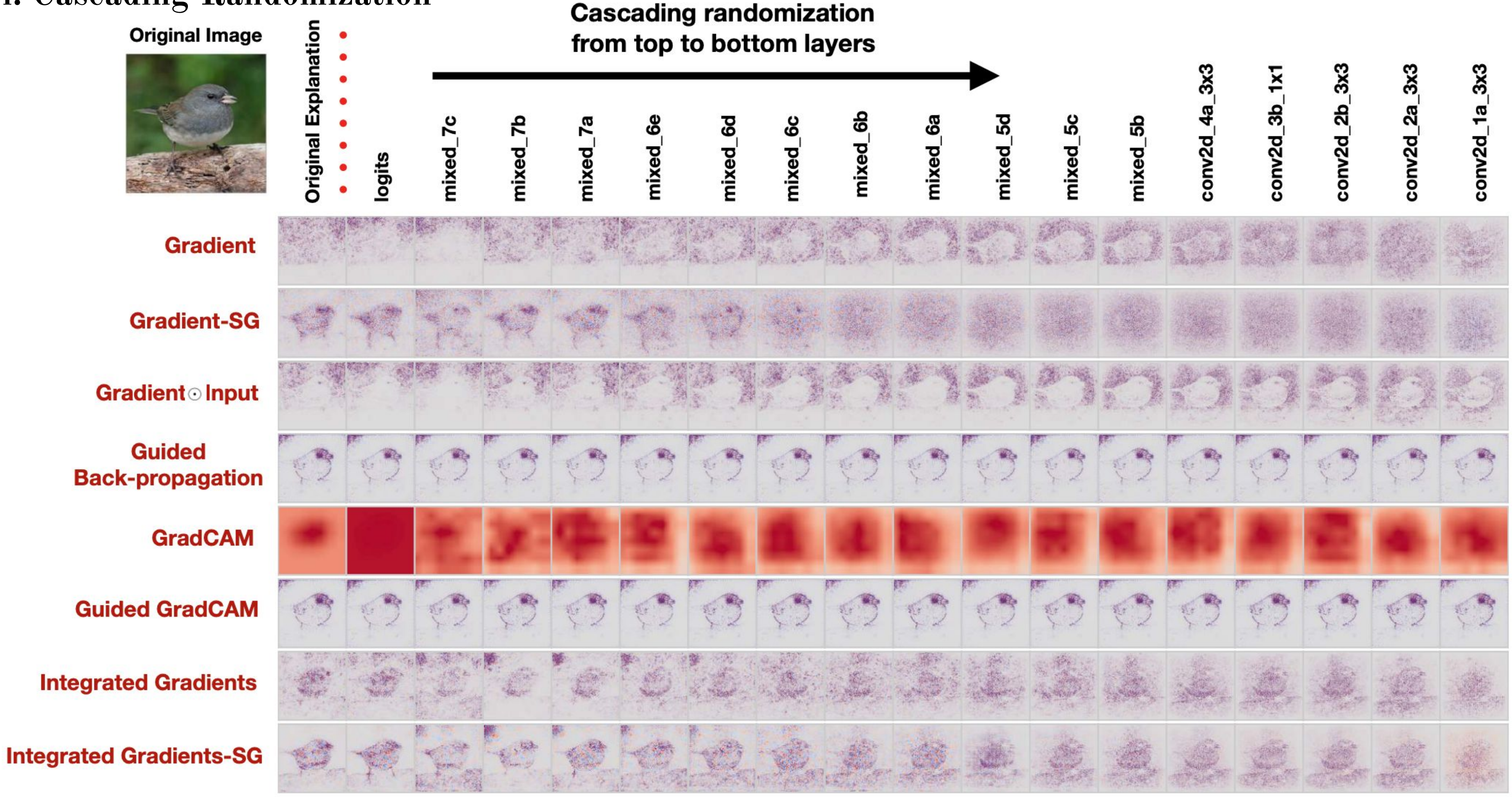
Paper: Sanity Checks for Saliency Maps, 2018

→ spot methods that are independent of the *model* and/or the *data generating process*

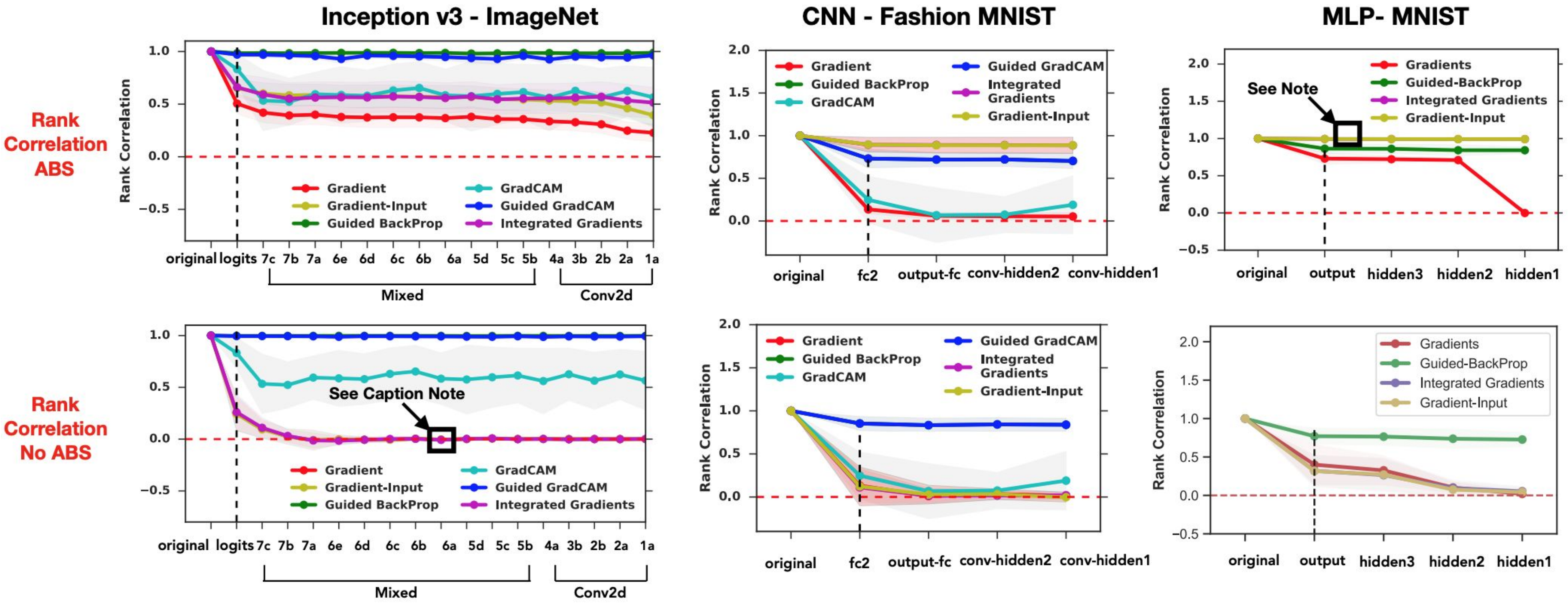
i. **Cascading Randomization** (model parameters)
   - → from top to bottom layers inject successively destructive noise; observe changes in map
     - variation: single layer perturbation

ii. **Data Randomization** (targets)
   - → randomly permute the labels and retrain the same model on the randomized data
     - model is forced to memorize the randomize level (senseless learning), no relationship between x and y

i. Cascading Randomization



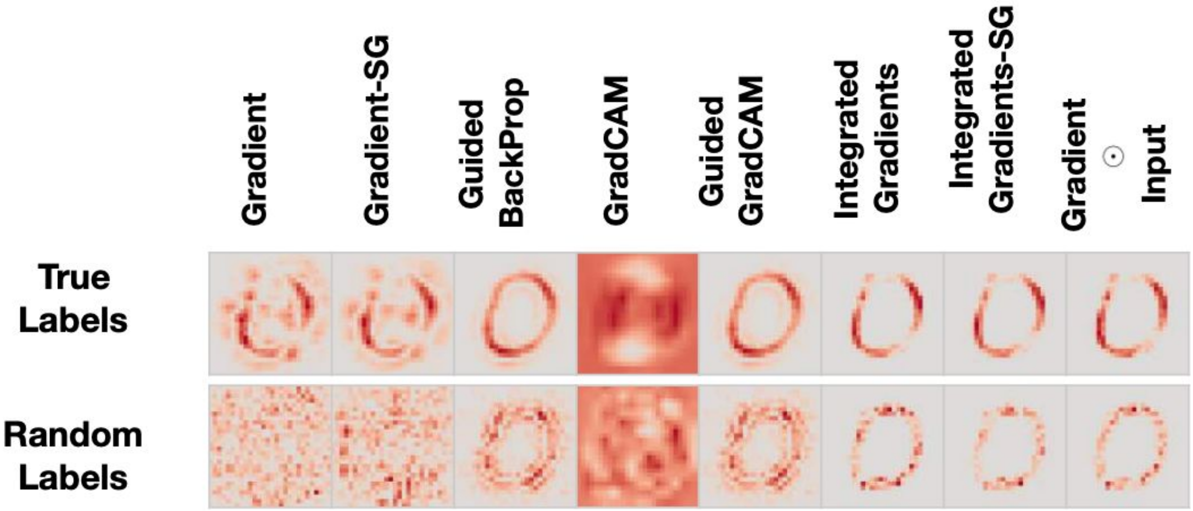
i. Cascading Randomization - Correlation Graph

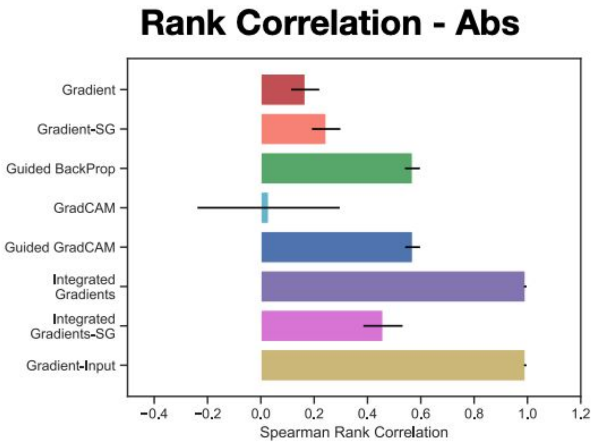


ii. Data Randomization

Absolute-Value Visualization

CNN - MNIST





Sanity Checks for Saliency Maps - Results

Some widely deployed saliency methods are independent of both the data the model was trained on, and the model parameters.

i. **Cascading Randomization:** Gradients & GradCAM show sensitivity while guided methods are invariant to model change.
ii. **Data Randomization:** Gradients & GradCAM are sensitive while guided methods are relatively invariant to label randomization.

Conclusion:

- Gradients & GradCAM pass the sanity checks;
- Guided BackProp & Guided GradCAM fail
- Visual inspection should be followed by similarity metrics assessment

PyTorch Implementation - CNN Visualization

https://github.com/utkuozbulak/pytorch-cnn-visualizations

II. RNN Visualization for Text

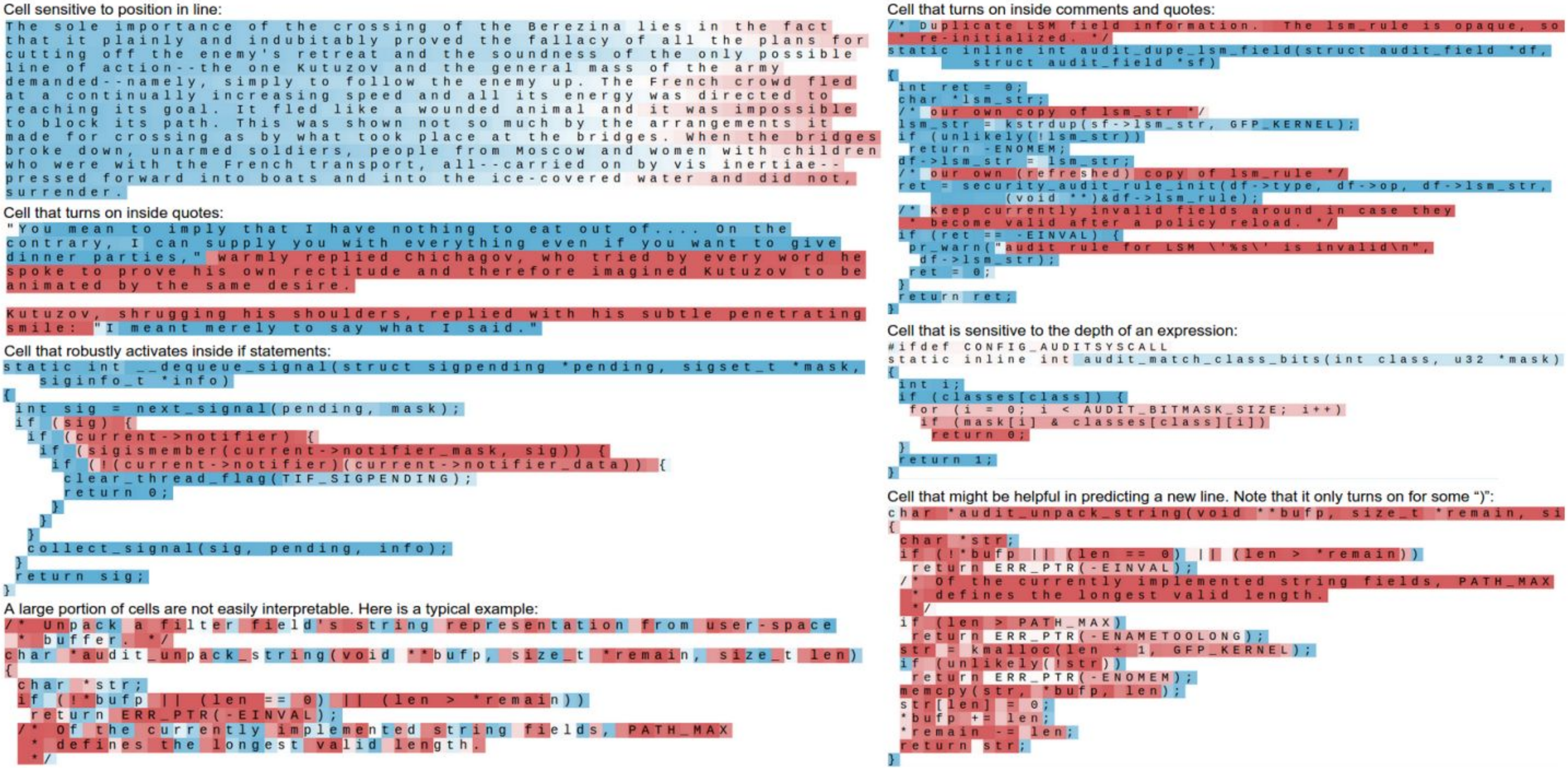
Paper: Visualizing and Understanding Recurrent Networks, 2016

→ Experiments reveal the existence of *interpretable cells* that keep track of long-range dependencies such as line lengths, quotes and brackets

→ Goal: track hidden-state-dynamics to identify interpretable cells

- **Hidden-state-dynamics** - changes in learned hidden representations over time produced by the model
- **Interpretable cells** - cells that activate related to some structure encountered in the language
- Complexity of interpretable cells differs depending on RNN/GRU/LSTM networks (cell state, hidden state, gates)

LSTM Cell Visualization



II. RNN Visualization for Text

LSTMVis tool

Paper: LSTMVis: Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks, 2017

- Provides interactive visualization to facilitate data analysis of RNN hidden states
- Helps to uncover what recurrent networks are able to capture
- Allows exploration and formation of hypotheses about RNN hidden state dynamics
- Combines a time-series based *select* interface with an interactive *match* tool to search for similar hidden state patterns in a large dataset (live system+source code)
- Requires only time-series of hidden states (no model requirement)
- Easy to adopt for visual analyses of different data sets, models and tasks (language modelling, translation,..)

http://lstm.seas.harvard.edu/

III. Interpretability for Neural Networks - Methods

1. **Expected Saliency** (Gradients)
   - *Paper*: Sensitivity based Neural Networks Explanations, 2018
   - *Method*: **computes aggregated gradients over dataset to aim for global explainability in neural networks by leveraging backpropagation**

2. **LIME** (Local Interpretable Model-Agnostic Explanations)
   - *Paper*: "Why Should I Trust You?" Explaining the Predictions of Any Classifier, 2016
   - *Method*: **local explanation by locally performing a weighted linear regression of slightly perturbed samples**

3. **DeepLIFT** (Deep Learning Important FeaTures)
   - *Paper*: Learning Important Features Through Propagating Activation Differences, 2017
   - *Method*: **attributes to each input x_i a value that represents the effect of that input being set to a reference value as opposed to its original value**

III. Interpretability for Neural Networks - Methods

4. **SHAP** (SHapley Additive exPlanations)

- *Paper*: A Unified Approach to Interpreting Model Predictions, 2017
- *Method*: **derives a comprehensive framework that unifies methods like LIME and DeepLIFT by leveraging cooperative game theory to compute explanations of model predictions**
- *Formula*:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

- *Pros*: enjoys the advantage of being the only values that satisfies three desirable axioms within the class of additive feature attribution methods - local accuracy, missingness and consistency
- *Cons*: requires 2^F passes for F features, which is very expensive for a Neural Network
  - find approximation methods (that the axioms conserve as much as possible) to accelerate computation depending on data structure, model architecture, ... - e.g. Kernel SHAP, LinearSHAP, DeepSHAP, MaxSHAP...

Definition 1 Additive feature attribution methods *have an explanation model that is a linear function of binary variables:*

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i, \tag{1}$$

→ methods that give additive attribution to features

where $z' \in \{0,1\}^M$, M *is the number of simplified input features, and* $\phi_i \in \mathbb{R}$.

Property 1 (Local accuracy)

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

→ explanations are truthfully explaining the model

The explanation model $g(x')$ *matches the original model* $f(x)$ *when* $x = h_x(x')$.

Property 2 (Missingness)

$$x'_i = 0 \implies \phi_i = 0$$

Missingness constrains features where $x'_i = 0$ *to have no attributed impact.*

→ absent features have no attributed impact

Property 3 (Consistency) *Let* $f_x(z') = f(h_x(z'))$ *and* $z' \setminus i$ *denote setting* $z'_i = 0$. *For any two models* f *and* f', *if*

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \tag{7}$$

for all inputs $z' \in \{0,1\}^M$, *then* $\phi_i(f', x) \geq \phi_i(f, x)$.

→ if input contribution is same or higher, then attribution to the feature should not decrease.

Theorem 1 *Only one possible explanation model g follows Definition 1 and satisfies Properties 1, 2, and 3:*

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \tag{8}$$

where $|z'|$ *is the number of non-zero entries in* z', *and* $z' \subseteq x'$ *represents all* z' *vectors where the non-zero entries are a subset of the non-zero entries in* x'.

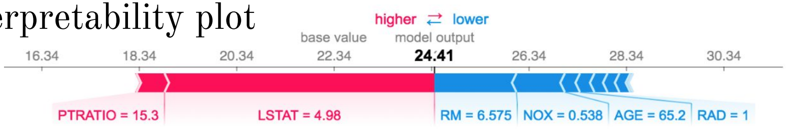
→ Shapley values are the only values satisfying these 3 axioms in the class of Additive feature attribution methods

III. Interpretability for Neural Networks - Methods

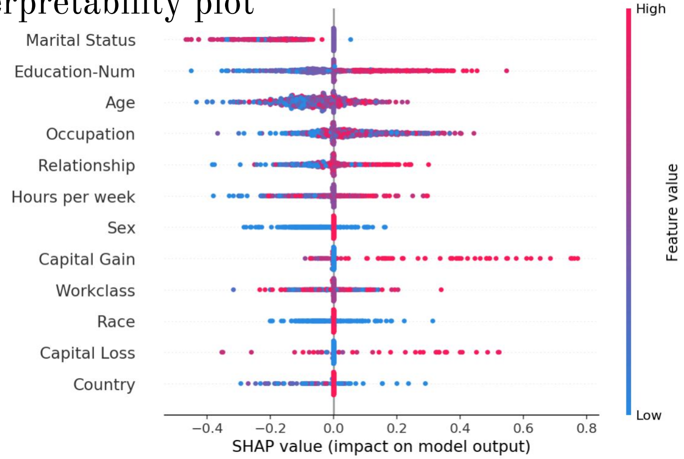
4. SHAP toolbox

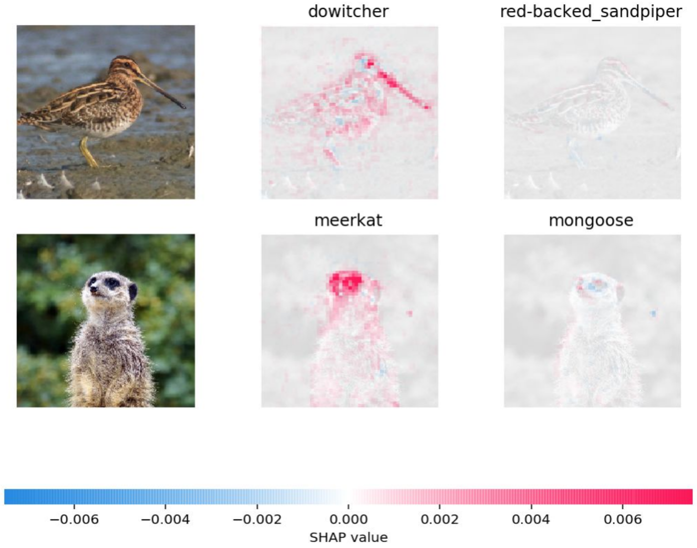
- *Link:* https://github.com/slundberg/shap

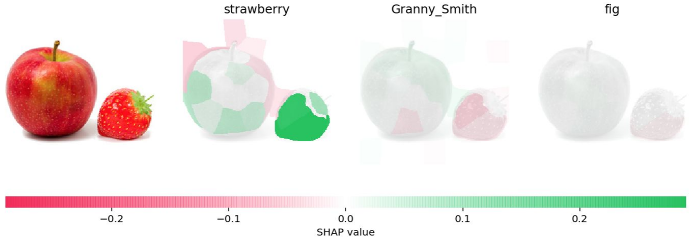
Local interpretability plot



Global interpretability plot







References: CNN Visualization

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In Advances in Neural Information Processing Systems, pages 9525–9536, 2018.

[2] Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. CoRR, abs/1311.2901, 2013. Published in Proc. ECCV, 2014.

[3] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034, 2013.

[4] Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524v5, 2014. Published in Proc. CVPR, 2014.

[5] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825, 2017.

References: CNN Visualization

[6] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806, 2014.

[7] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 3319–3328, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[8] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In CVPR, 2016.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In arXiv:1610.02391v3, 2017.

[10] PyTorch library “pytorch-cnn-visualizations”: github.com/utkuozbulak/pytorch-cnn-visualizations

References: RNN Visualization

[1] Andrej Karpathy, Justin Johnson, Li Fei-Fei, Visualizing and Understanding Recurrent Networks, arXiv preprint arXiv:1506.02078, 2015

[2] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, Alexander M. Rush, LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks, arXiv preprint arXiv:1606.07461, 2016

[3] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In Proceedings of NAACL

References: Explainability

[1] Enguerrand Horel, Virgile Mison, Tao Xiong, Kay Giesecke and Lidia Mangu, Sensitivity based Neural Networks Explanations, arXiv preprint arXiv:1812.01029, NeurIPS 2018

[2] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, arXiv preprint arXiv:1602.04938, 2016

[3] Avanti Shrikumar, Peyton Greenside, Anshul Kundaje, Learning Important Features Through Propagating Activation Differences, arXiv preprint arXiv:1704.02685, 2017

[4] Scott Lundberg, Su-In Lee, A Unified Approach to Interpreting Model Predictions, arXiv preprint arXiv:1705.07874, , NeurIPS 2017

Other Interesting Resources

- Visualizing and Understanding: CNNs for Computer Vision (slides)
- Visualizing and Understanding Neural Models in NLP (2016) (paper)
- Understanding Neural Networks Through Deep Visualization (article)
- How Sensitive are Sensitivity-Based Explanations? (2019) (paper)
- Using CAM for Visualizing CNNs (code)