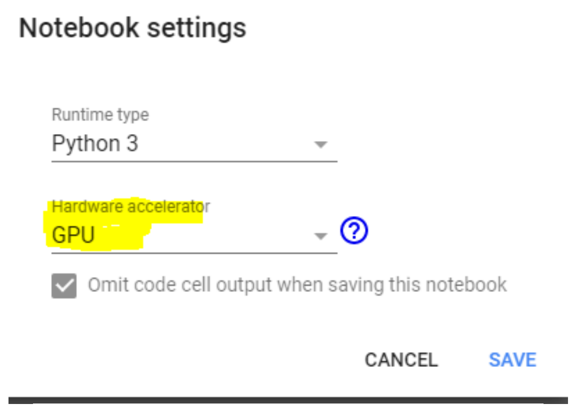
Explanation techniques for neural networks

Laura **Ri**eger (lauri@dtu.dk)

DTU Compute
Department of Applied Mathematics and Computer Science

Colab notebook

- Colab notebook:
  - **bit.ly/2PdScxx**
  - File -> Save a Copy in Drive
  - Edit -> Notebook settings -> GPU



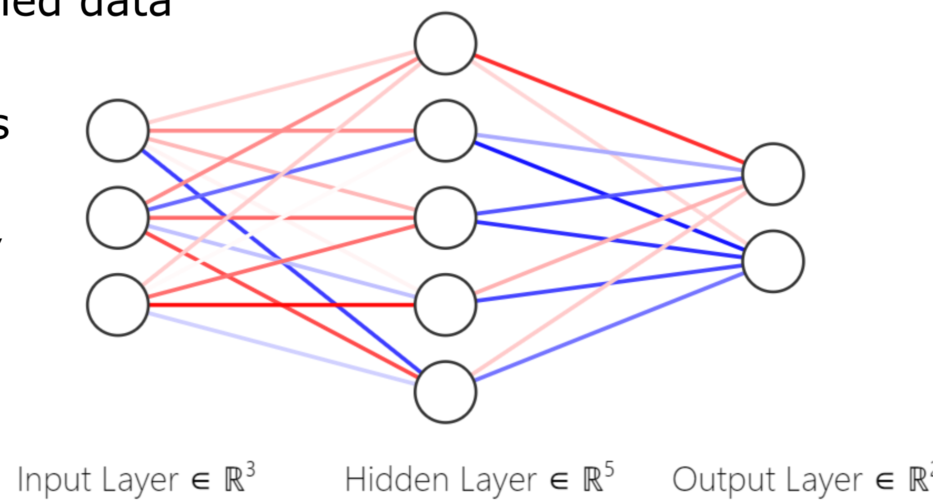
  - Tools -> Preferences -> Misc ->

Intro

- Assumptions
  - Complicated non-linear task
  - Can **not** be solved by an intuitively explainable model
- Complete understanding is not possible
  - Single decision explanations
  - Region-based explanations
- Approaches
  - Backpropagation
  - Local approximation with simple model

Neural Networks

- Complex function composed of smaller functions
- Trained by large amounts of labelled data
- Training:
  - Differentiate loss over weights
  - Update weights
- Differentiation done automatically



$$\mathcal{L}_{(}\theta, X, y) = f_\theta(X) * \log(y) + (1 - f_\theta(X)) * \log(1 - y)$$

$$\Delta\theta = \lambda * \frac{\delta\mathcal{L}_{(}\theta, X, y)}{\theta}$$

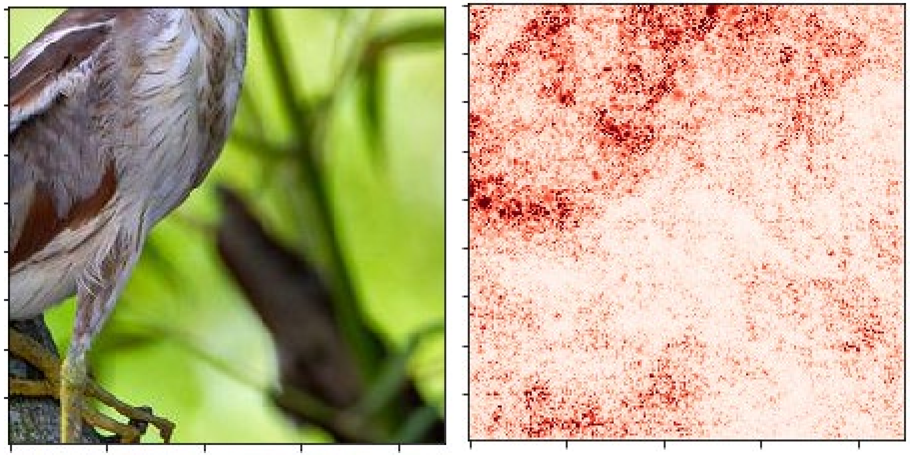
Saliency [1]

- Want to know what parts of the input responsible for the output
- Areas with a high gradient
    - Output changes fast here
        - Probably important

$$\frac{\delta y_c}{\delta X}$$

- Output is fully differentiable
- Basic explainability method
- Super noisy
    - High redundancy
    - Noisy function
- Popular variant:
  Guided Backprop [9]

$$\frac{\partial' y_c}{\partial' x_{i,j}} = \mathrm{ReLU}\left(\frac{\partial y_c}{\partial l_{-1}}\right)\dots\mathrm{ReLU}\left(\frac{\partial l_1}{\partial x_{i,j}}\right)$$

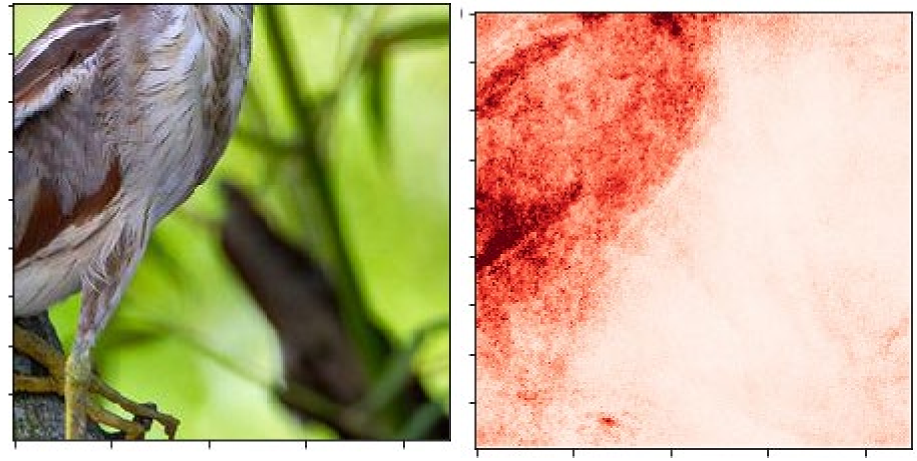


SmoothGrad [15]

- How do we remove noise? With more noise!

$$\sum_N \frac{\delta y_c}{\delta(X + \sigma_n)}$$

- Average over number of input with noise added
- Clears up image
- Applicable to all basic explanation techniques



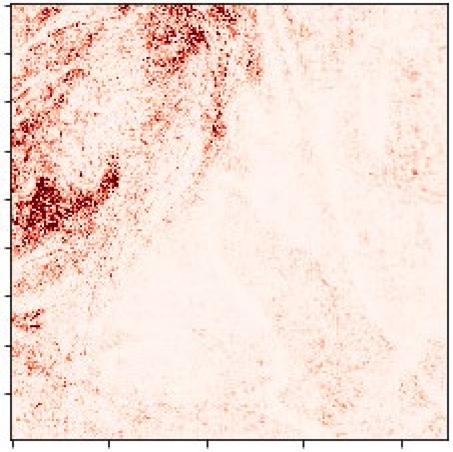
Integrated Gradients [16]

- How do we remove noise? With more (different) noise!

$$\frac{1}{N}\sum_{N}\frac{\delta y_c}{\delta(\mathbf{0}+o_n*(X-\mathbf{0}))}*(X-\mathbf{0}), o_n \sim U(0,1)$$

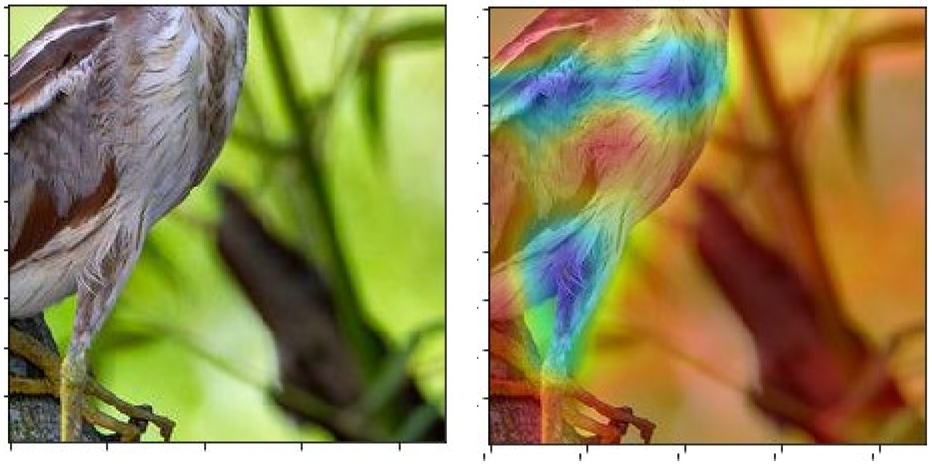
- Average over samples on the pathway between baseline
- Dependent on choice of baseline (all black, average, …)

- Recent extension: use samples as baseline



GradCAM [13]

- Gradient-weighted Class Activation Mapping
- Way less noisy
- No finegrained input
- Procedure:
- Calculate gradient to last convolutional layer
- Average gradient for each channel
- Multiply average gradient with all activations
- Average over all channels
- Upsample

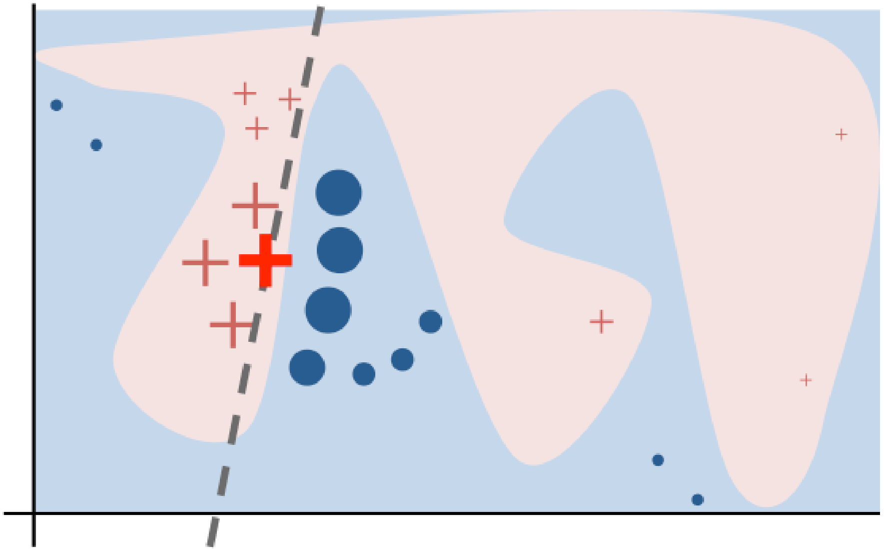


Other methods based on backpropagation

- LRP [12]
- Expected Gradients [17]
- CAM [2]
- GuidedBackprop [9]

Local approximation with interpretable model – LIME [4]

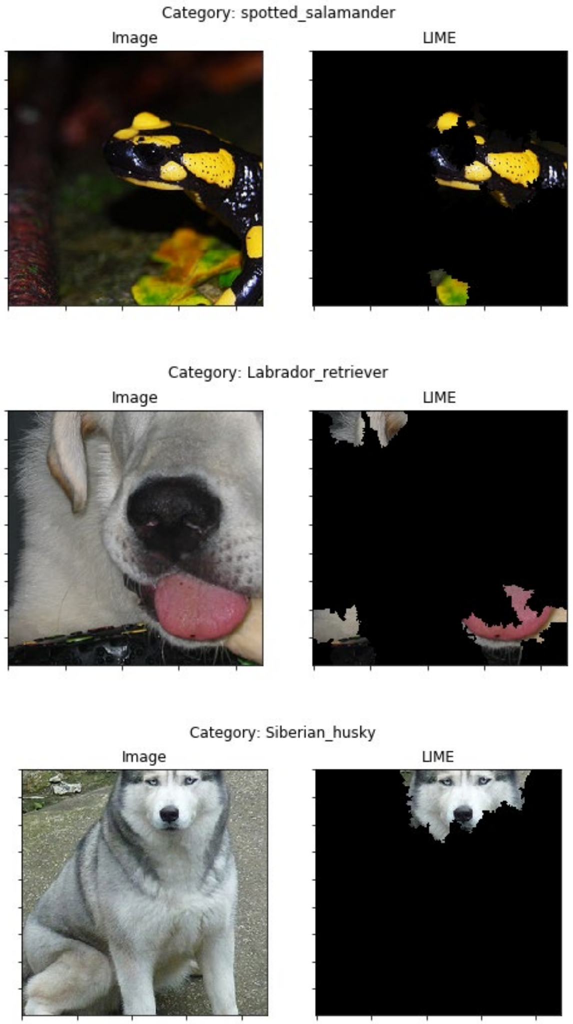
- Intuition:
  - Sample around **x**
  - Weigh samples according to distance
  - Train linear classifier
  - Obtain explanation
- Low-dimensional representation necessary
  - For images: segment into super-pixels
  - For text: bag of words



From https://github.com/marcotcr/lime

Local approximation with interpretable model – LIME [4]

- Intuition:
  - Sample around **x**
  - Weigh samples according to distance
  - Train linear classifier
  - Obtain explanation
- Low-dimensional representation necessary
  - For images: segment into super-pixels
  - For text: bag of words



Images obtained with LIME library from pretrained VGG16

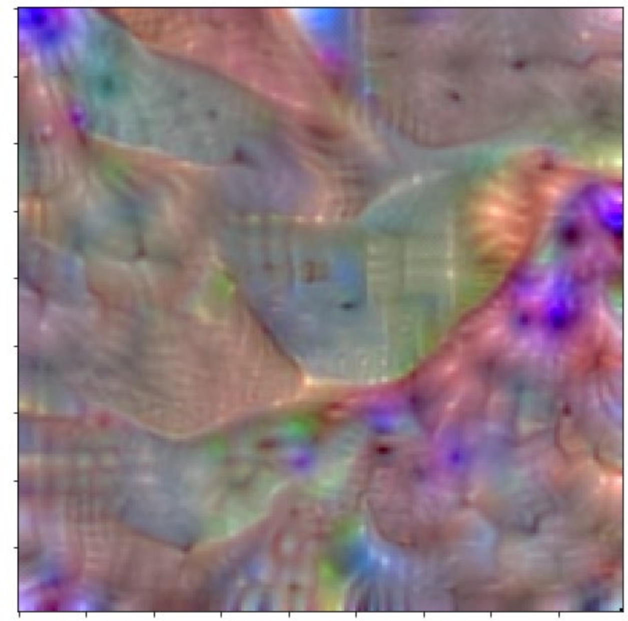
Higher-level

- Network level explanations
- Requires domain knowledge
- Interesting for risk and fairness analysis
- Two approaches presented
  - Analyzing specific network parts
  - Analyzing specific aspects

Probing the network

- "Understanding Neural Networks Through Deep Visualization" [5]
  - Idea: iteratively optimize activation of neurons with backpropagation
  - Regularize to encourage realisism
  - For output or intermediate layers
- Alternatives
  - Bau, David, et al. "Network Dissection: Quantifying Interpretability of Deep Visual Representations." *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.
  - Alain, Guillaume, and Yoshua Bengio. "Understanding intermediate layers using linear classifier probes." (2016).

Category: hen



Obtained with kerasvis
from pretrained VGG16

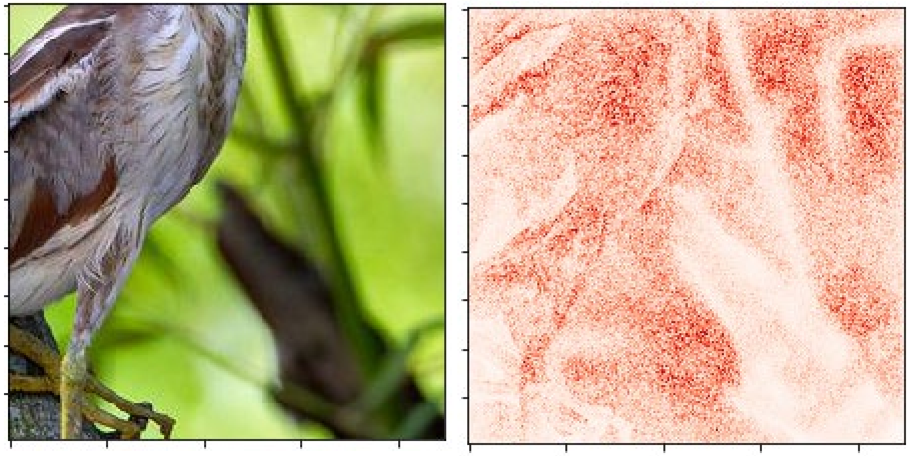
Testing with Concept Activation Vectors CAV

- Proposed by Kim et al [8]
- Requires high domain knowledge
- Idea:
  - assemble dataset {P,N}
  - Train linear classifier on representation of given layer
  - Obtain weights
- Useful for
  - Evaluating given input
  - Identifying a known bias in dataset and model

Kim, Been, et al.
"TCAV: Relative concept importance testing with Linear Concept Activation Vectors." (2018).

What should I use?

- How do we evaluate interpretation methods?
  - Humans are really bad evaluators
- Theoretical properties
- Sanity checks
- Human evaluation
  - Fooled by clearness
- Remove input consecutively – check output degradation



Integrated Gradients on random weights

Take-away

- We have to make a trade-off when obtaining explanations from NNs
- Different approaches have different pros and contras
- Task in question needs to be considered

	Fidelity	Understandability	Sufficiency	Low construction overhead	Efficiency
Backprop	+	-	o	+	+
Local	o	+	o	+	-
High-level	+	+	-	-	o

References

1. Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." *arXiv preprint arXiv:1312.6034* (2013).
2. Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
3. Zhang, Quanshi, Ying Nian Wu, and Song-Chun Zhu. "Interpretable Convolutional Neural Networks."
4. Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why should i trust you?: Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2016.
5. Yosinski, Jason, et al. "Understanding neural networks through deep visualization." In ICML Workshop on Deep Learning.
6. Bau, David, et al. "Network dissection: Quantifying interpretability of deep visual representations." arXiv preprint arXiv:1704.05796 (2017).
7. Alain, Guillaume, and Yoshua Bengio. "Understanding intermediate layers using linear classifier probes." arXiv preprint arXiv:1610.01644 (2016).
8. Kim, Been, et al. "TCAV: Relative concept importance testing with Linear Concept Activation Vectors." (2018).
9. Springenberg, Jost Tobias, et al. "Striving for simplicity: The all convolutional net." arXiv preprint arXiv:1412.6806 (2014).
10. 1Zhang Q, Wu YN, Zhu S-C. Interpretable Convolutional Neural Networks. https://arxiv.org/pdf/1710.00935.pdf. Accessed February 20, 2018.
11. Smilkov, Daniel, et al. "Smoothgrad: removing noise by adding noise." arXiv preprint arXiv:1706.03825 (2017).
12. Bach, Sebastian, et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." PloS one 10.7 (2015): e0130140.
13. Selvaraju, Ramprasaath R., et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." ICCV. 2017.
14. Swartout, William R., and Johanna D. Moore. "Explanation in second generation expert systems." Second generation expert systems. Springer, Berlin, Heidelberg, 1993. 543-585.
15. Smilkov, Daniel, et al. "Smoothgrad: removing noise by adding noise." arXiv preprint arXiv:1706.03825 (2017).
16. Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.
17. Erion, Gabriel, et al. "Learning Explainable Models Using Attribution Priors." arXiv preprint arXiv:1906.10670 (2019).