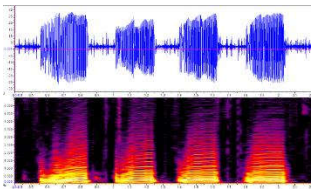
SL5- Interpretable Models

Luis Souto Maior, BSc

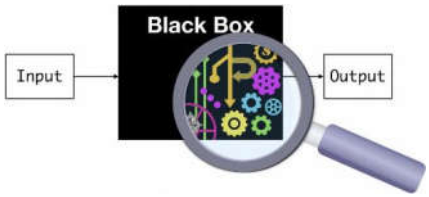
`Interpretable Machine Learning

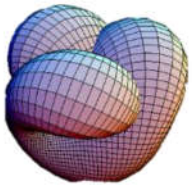
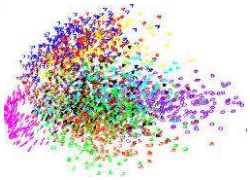
What is interpretable?

Interpretable information consists of pieces of data which humans can understand. These types of data include, for example: images, sounds, words, letters.

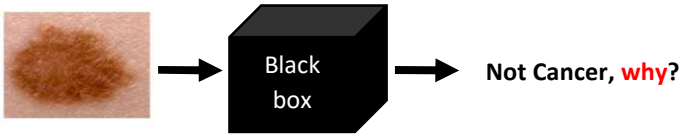
With deep learning, however, these **black-box models** learn to translate these interpretable pieces of data, into complex representations which are not interpretable for us humans. For example, high dimensional manifolds and nonparametric multimodal distributions.



Motivation

In high risk areas, **such as medicine**, it is of outmost importance to understand the rationale of predictive models, otherwise it could lead to life threatening situations. Knowing the **"why" of given prediction** can enable us to understand more of the problem in hands.



Definitions

Things to think about

- **What kind and amount** of explanation is needed? How much do I have to **modify the model** to enable interpretability? How much of an **impact** will there be on the model **performance**?

Interpretability *versus* Explainability

Interpretability corresponds to understanding what the model has learned. For example, a model trained to predict disease versus healthy should learn known features of the disease. Knowing this enables us to assert future behaviour, allowing it to be more predicable in real world scenario.

Explainability, on the other hand, represents the ability to explain a single prediction. Essentially describing what features were used to achieve a given prediction, and how changes in these features modify the prediction outcome, enabling, in practice, an exact rationale for that specific inference.

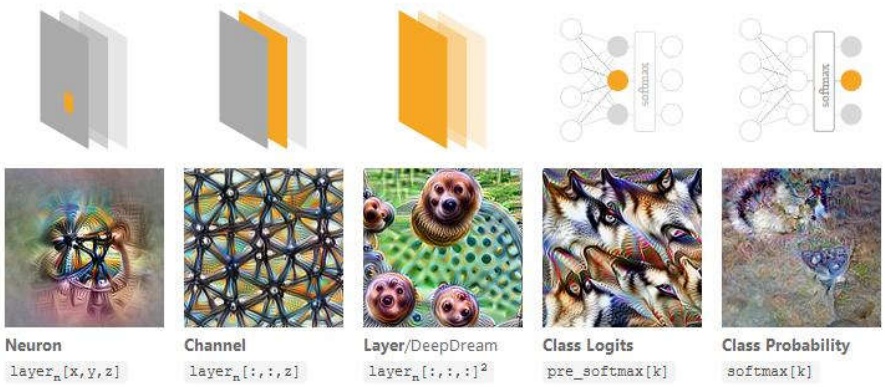
Model-agnostic *versus* Model-specific Methods

Interpretability methods can be differentiated on whether they work for every type of inference model, also known as model-agnostic methods, or if the model itself needs to be designed or modified in a specific way to explain its prediction.

Activation-based methods

Feature Visualization via Activation Maximization

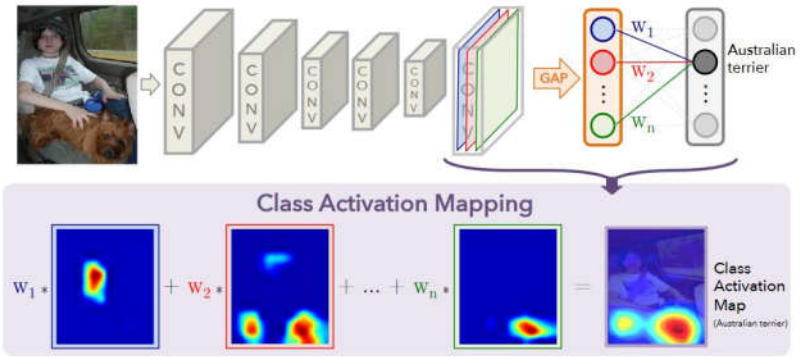
After training a deep learning model, it is possible to find out the exact contribution of a node inside of the model by simply optimizing an input image so that it maximizes the activation in the output of that node. Depending on the type of optimization objective, i.e., what to optimize for (single neuron output, channel output, full layer output, class logits or probability), a different insight can be obtained.



Class Activation Mapping (CAM)

Algorithm:

1. Feed input and store feature maps from last layer
2. Apply global average pooling to find weights $Fk = \sum_{x,y} f_k(x,y)$
3. Compute softmax for a class c with $Sc = \sum_k w_k^c F_k$ where w_k^c are learnable



Gradient-based methods

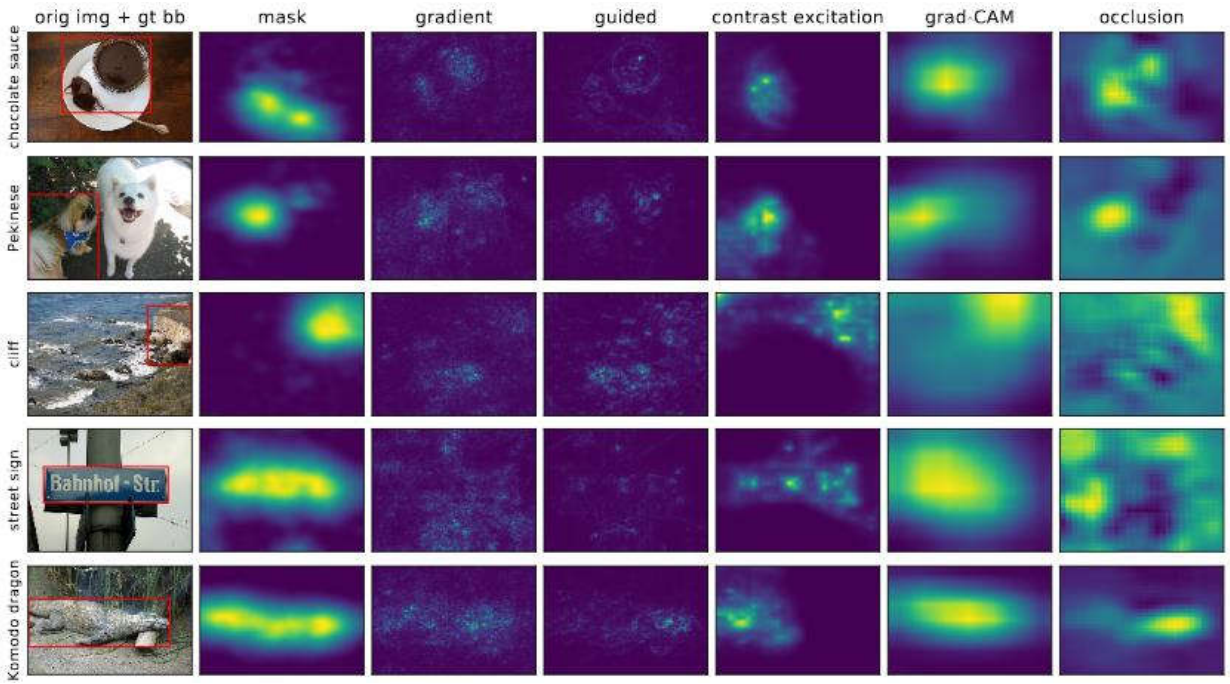
Sensitivity Analysis

Uses gradients of the output with respect to input to attribute importance to features on the input.

Figure 2: **Image-specific class saliency maps for the top-1 predicted class in ILSVRC-2013 test images.** The maps were extracted using a single back-propagation pass through a classification ConvNet. No additional annotation (except for the image labels) was used in training.

Guided Backpropagation and Gradient Class Activation Mapping (Grad-CAM)

Both of these methods are variations of sensitivity analysis. Guided backprop essentially zeroes out all non-positive gradients during backpropagation for attribution. The rationale is that we are looking for features on the input that are positively impacting the prediction. Grad-CAM follows up on CAM by measuring the weights as the global average of the gradients at each layer during backprop. This enables model-agnostic feature attribution and explanation.



Uncertainty-based methods

Data dependent uncertainty and model dependent uncertainty

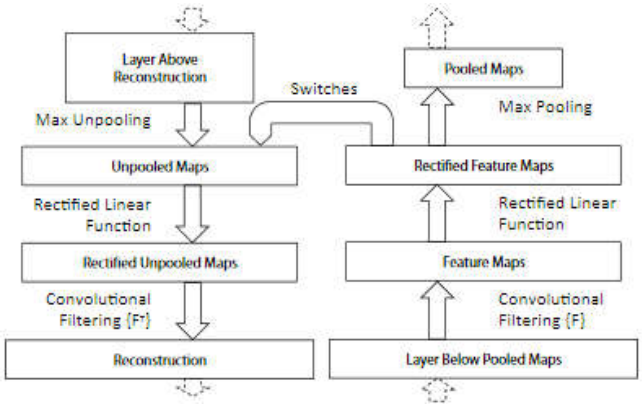
These techniques consist of trying to infer not only the output of a prediction but the confidence via predicting the standard deviation of the output. This enables modelling of data-related uncertainty on the prediction. With methods such as Monte Carlo Dropout, it is also possible to measure model dependant uncertainty via turning on an off of all the nodes in the model randomly and reporting the average output that is predicted. This enables us to stochastically rule out the effects of the nodes in the model, essentially leaving out only the error associated with the model.

Explainer-based methods

DeConvNet

Algorithm:

1. Attach a deconv layers to conv layers
2. Force deconv layers to predict feature maps of regular conv layers
3. DeConvNet will learn to predict inputs corresponding to a class



Latent Space Decoding

Rationale: train a decoder from latent space to input space to translate from non-interpretable to interpretable space, then interpolate through principal components of latent space to explain features learned.