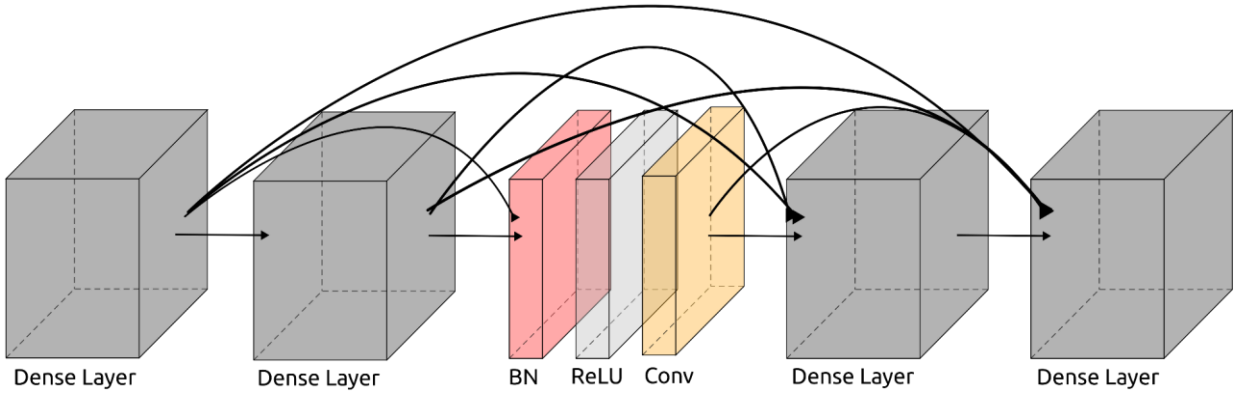
DenseNet Canonization

As we saw in a previous post, some challenges caused in explaining neural network decisions can be overcome via canonization. However, models with complex topology can pose other challenges in the canonization process. In this case, a focused approach on the network structure needs to be taken. In this post, we will demonstrate the canonization of DenseNets.

DenseNet Architecture

For example, in the basic DenseNet architecture, the neural network is defined as a sequence of *Dense* blocks. Each Dense block includes Dense layers, each of which is composed of a sequence of Batch Norm (BN), ReLU and convolutional layers. The following figure illustrates the connectivity pattern in each dense block:



This complex connectivity pattern doesn't allow us to straightforwardly merge batch normalization layers to the convolutional layers before them, because the initial batch normalization in a dense block receives the concatenated outputs of many linear layers.

Canonization Steps

Since batch normalization can be seen as a linear layer, we can merge it to the convolutional layer if we can find a way to switch the order of the ReLU activation and the Batch Normalization.

This is possible by defining the new activation function

$$ReLU_{Thresh}(x) = \begin{cases} x \text{ if } (w_{BN} > 0 \text{ and } x > z) \text{ or } (w_{BN} < 0 \text{ and } x < -z) \\ 0 \text{ otherwise} \end{cases}$$

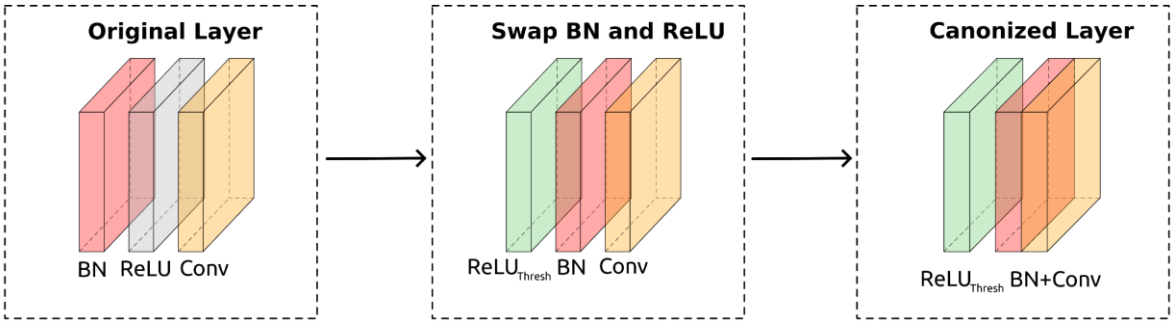
With this definition, it is easy to see that

$$ReLU(BN(x)) = BN(ReLU_{Thresh}(x))$$

This equation means that we can turn the $ReLU$ activation in each dense layer to $ReLU_{Thresh}$, using the parameters of the batch normalization. We can then merge the batch normalization parameters with the convolutional layer using the following equations:

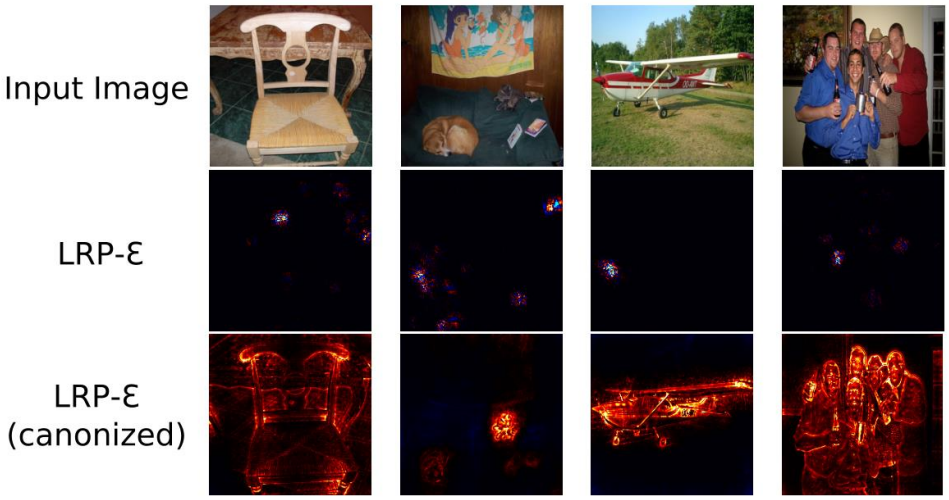
$$w_{new} = \frac{w_{BN}}{\sqrt{\sigma+\epsilon}} w_L \qquad \text{and} \qquad b_{new} = (b_{BN} - \frac{w_{BN}\mu}{\sqrt{\sigma+\epsilon}}) w_L + b_L$$

This concludes the canonization: we can now discard the batch normalization layer and use backpropagation methods to explain network decisions. The canonization steps are summarized in the following figure.



Conclusions

We can define a custom activation function that allows us to change components in the network in a way that allows Batch Normalization layers to be merged into linear layers. As the following figure shows, this solves issues caused by the Batch Normalization in the explanations:



Relevance to IToBoS

In iToBoS, many different AI systems will be trained for specific tasks, which in combination will culminate in an “AI Cognitive Assistant”. All those systems will need to be explained with suitable XAI approaches to elucidate all possible and required aspects of the systems’ decision making. Throughout the iToBoS project, innovative state-of-the-art model architectures will be deployed, for which model canonization will be required to optimize the quality of explanations.

Authors

Galip Ümit Yolcu, Fraunhofer Heinrich-Hertz-Institute

Frederik Pahde, Fraunhofer Heinrich-Hertz-Institute

Sebastian Lapuschkin, Fraunhofer Heinrich-Hertz-Institute