HOW CAN I EXPLAIN THIS TO YOU ?

Dhrubajit Basumatary[1], Pranjal Sharma[2], and Ujwal Kumar[3]

[1]Department of Mechanical Engineering [2]Department of Economics [3]Department of Chemical Engineering

1 Abstract

Deep visual models are fast surpassing human-level performance for various vision tasks, including image classification. However, state of the art deep classification models networks have two major problems: (1) Deep neural networks hardly offer any explanation of their decisions, hence considered "black-boxes" and (2) They've also been proven to be subject to adversarial examples, which are meant to cause models to predict incorrectly.

Neural networks, consistently misclassify adversarial examples—inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset, such that the perturbed input results in the model outputting an incorrect answer with high confidence. We used *FGSM* for creating adversarial attacks. Another issue is that, while deep neural networks are increasingly being utilised to automate data processing and decision-making, their decision-making process is still mostly unknown and difficult to explain to end users. In this project, we used the *GradCAM* method to solve the challenge of Explainable AI for deep image classification networks in a white-box manner.

2 Image Classification

The Image classification task deals with predicting the categories for a novel set of test images, given a set of images that are all labeled with a single category and measure the accuracy of the predictions. We build two famous classification models namely, VGG19 using Keras and Resnet using PyTorch.

2.1 VGG

VGG [1] is an innovative object-recognition model that supports up to 19 layers. Built as a deep CNN, VGG also outperforms baselines on many tasks and datasets outside of ImageNet when it was published. VGG addresses very important aspect of CNNs that previous derivates hasn't: depth, supports up to 19 layers. VGG takes in a 224x224 pixel RGB image. The VGG model architecture consists of 5 convolution blocks with 5 max pooling layers. The convolutional layers in VGG use a very small receptive field which is followed by a ReLU unit and 3 FC layers at last each uses ReLU.

2.2 ResNet

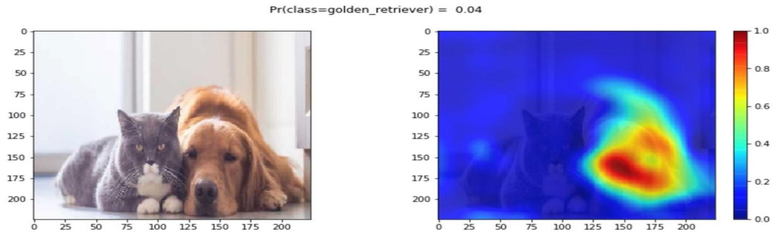
One of the problems ResNets [2] solve is the famous known vanishing gradient. This is because when the network is too deep, the gradients from where the loss function is calculated easily shrink to zero after several applications of the chain rule. The authors address the degradation problem by introducing a deep residual learning framework with a hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping.. With ResNets, the gradients can flow directly through the skip connections backwards from later layers to initial filters.

The implementation of the ResNet50 model is carried out using Pytorch and VGG with Tensorflow. The models are trained over the CIFAR-10 dataset. Mini-batch gradient descent, with a batch size of 128 was used for training. Accuracy obtained over the test dataset for ResNet model is close to 80% and for Vgg model close to 60%.

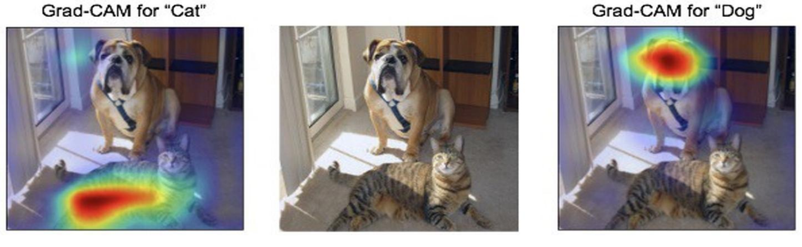
3 Explainable AI

In order to build trust in intelligent systems and move towards their meaningful integration into our everyday lives, it is clear that we must build 'transparent' models that have the ability to explain why they predict what they predict. We used an approach called GradCAM [3] that generates an importance map indicating how salient each pixel is for the model's prediction using internals of the base model, such as the gradients of the output with respect to the input and intermediate feature maps.

Gradient-weighted Class Activation Mapping (Grad-CAM), uses the class-specific gradient information flowing into the final convolutional layer of a CNN to produce a coarse localization map of the important regions in the image. Grad-CAM requires no re-training and is broadly applicable to any CNN-based architectures. A number of previous works have asserted that deeper representations in a CNN capture higher-level visual constructs. Furthermore, convolutional layers naturally retain spatial information which is lost in fully-connected layers, so with the expectation that the last convolutional layers to have the best compromise between high-level semantics and detailed spatial information, Grad-CAM uses the gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest. These gradients flowing back are global-average-pooled for weighting the activation maps. The results below are for pre-trained VGG and ResNet.



In the above example, The first image is a sample image, and the second image is the heatmap imposed on the sample image for the target class of Labrador retriever.



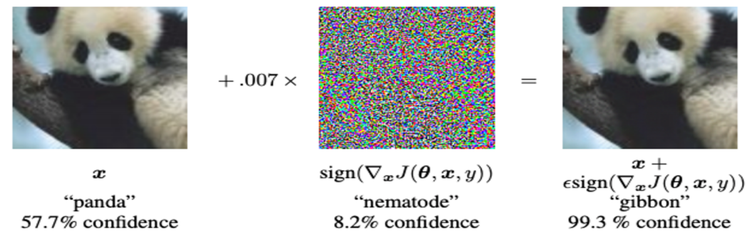
In this example, both the labels are used, one for cat and one for dog and the heatmap clearly shows the correct expected region for both the categories.

4 Adversarial AI

As we seek to deploy machine learning systems not only on virtual domains, but also in real systems, it becomes critical that we examine not only whether the systems don't simply work "most of the time", but which are truly robust and reliable. However, several state-of-the-art neural networks, are vulnerable to adversarial examples. An adversarial example is an instance with small, intentional feature perturbations that cause a machine learning model to make a false prediction. [4] invented the *Fast Gradient Sign Method(FGSM)* for generating adversarial images. The gradient sign method uses the gradient of the underlying model to find adversarial examples. The original image x is manipulated by adding or subtracting a small error ε to each pixel. Whether we add or subtract ε depends on whether the sign of the gradient for a pixel is positive or negative. Adding errors in the direction of the gradient means that the image is intentionally altered so that the model classification fails. This vulnerability occurs when a neural network treats a relationship between an input pixel intensity and the class score linearly. The following formula describes the core of the fast gradient sign method:

$$x' = x + \varepsilon.sign(\nabla_x J(\theta, x, y))$$

where $\nabla_x J$ is the gradient of the models loss function with respect to the original input pixel vector x, y is the true label vector for x and θ is the model parameter vector.



In the example above, the pretrained vgg model was predicting the image correctly with 57.7% confidence. After that, an adversarial image is created by adding perturbations with $\varepsilon = 0.007$. Now the model completely misinterprets the class and predicts the image as a gibbon with a very high confidence.

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019.

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.