Case Study:
Explaining Diabetic Retinopathy Detection Deep CNNs via Integrated Gradients

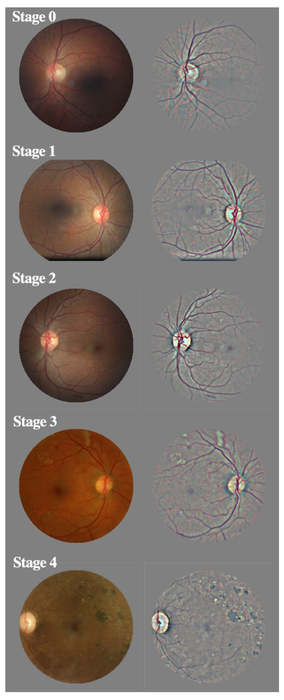
Linyi Li Matt Fredrikson Shayak Sen Klas Leino Anupam Datta

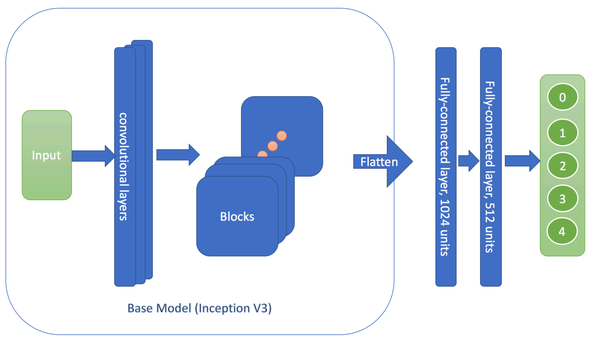
Carnegie Mellon University
Carnegie Mellon University CyLab Security and Privacy Institute

Abstract

The model Diabetic retinopathy is an eye disease caused by diabetes. We trained a model for detecting the stage of diabetic retinopathy using digital fundus photographs as input.
The method The integrated gradient is an attribution method which measures the contributions of input to the quantity of interest.
In this work, we applied integrated gradients to explaining a neural network for diabetic retinopathy detection.
We explored some new ways for applying this method. The visualization results extend the use of diabetic retinopathy detection model from merely predicting to assisting finding potential lesions.

Model





Preprocessing

Images are preprocessed before feeding to the classifier.
1. Detect the actual retina diameter by counting non-black pixels of middle horizontal line.
2. Calculate Gaussian blurred image to filter out high-frequency noises of the image.
3. Use the diameter to scale down the retina to 256x256 size - the input shape of our network.
4. Calculate a new highly Gaussian blurred image, subtract blurred image from original image.
5. Crop outer 10% of retina to eliminate border noises, fill in the blank area with half-gray color.

- Varied sizes and tones are now in the same style.
- Features like lesions, vessels and bulges are highlighted.

	Baseline (Harry et al., 2016)	Our Model
Accuracy	73.76%	75.1540%
Sensitivity	95.05%	96.1500%
Specificity	29.99%	31.1423%
κ	0.4398	0.5107

Training

- Use Google Inception V3 architecture trained for ImageNet Challenge as the base model.
- Remove the top output layer of the base model
- Add two fully-connected layers over the base model for transfer learning.
- Use a five-unit softmax classification layer as the output, and each of unit represents the predict probability for each stage.
- The training data is highly unbalanced. → For each mini-batch, we guarantee that
  - 50% is from stage 0, 6.25% is from stage 1,
  - 12.5% is from stage 2, 6.25% is from stage 3
  - and 25% is from stage 4.
- Learning rate:
  - 0.001 for first 550 mini-batches
  - 0.0002 for next 350 mini-batches
  - 0.00005 for last 495 mini-batches

Attribution Method

The definition of integrated gradients:

$$IntegratedGrads_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

- x' - the baseline } Both are vector or matrix of input shape
- x - current input } Subscript i stands for the i^{th} dimension
- Function $F(\cdot)$ is the quantity of interest.
- The baseline should be a void input:
  - Contains no useful information
  - Totally neutral for the quantity of interest.

- The integrated gradients method attributes the outcome value $F(x)$ to the input.
- It is a significant indicator telling us the contributions of each components of the input to the quantity.

Numerical Approximation

- Uniformly pick out points on the straight line from the baseline to the input and calculate the gradient of $F(x)$ on these points.

$$IntegratedGrads_i(x) \approx (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial F\left(x + \frac{k}{m} \times (x - x')\right)}{\partial x_i} \times \frac{1}{m}$$

- Full half-gray image is chosen as baseline, which conveys no information, rather than the frequently used all-zero baseline.

Lesion Detection

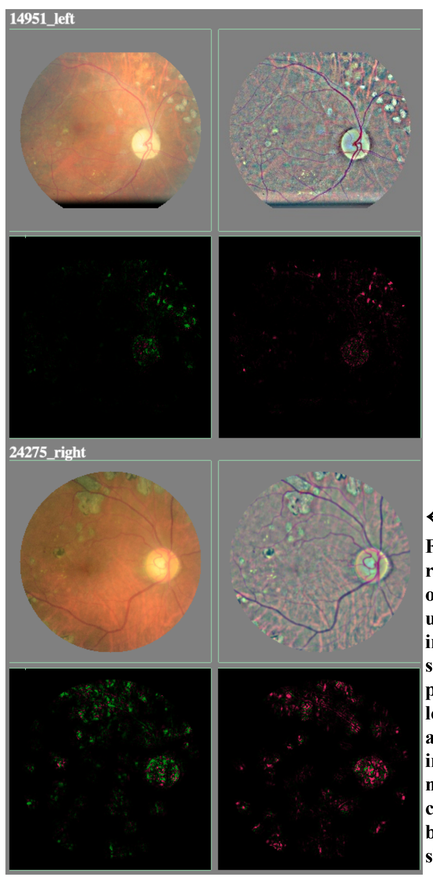
- Directly attribute the predict outcome to input pixel → Results are not as ideal as we expected.

Attributing Outcome to Intermediate Layers

- Use the half-gray baseline to calculate the baseline activations of the inspected layer neurons. These activations are stored as the baseline of this layer.
- The instance is inputted and instance activations are figured out.
- Use the integrated gradients expression to calculate the attribution values for each neuron.
- Use the integrated gradients expression again to figure out what pixels are responsible for the intermediate layer neuron.
  - Quantity of Interest: activation of the intermediate layer neuron
  - Variables: Input layer pixels
- Sort neurons by their attribution value, and pick out top most positive ones and most negative ones for computing.
- Visualize the influential neurons by
  - their active region, or
  - their resultant active region, i.e., the sum of active regions weighted by the neuron attribution value.

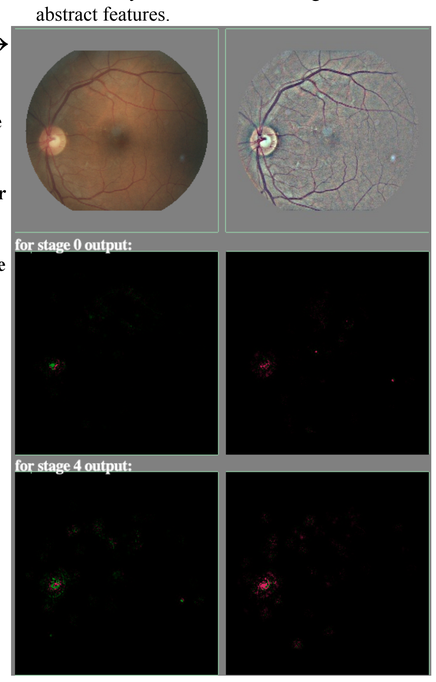
Observations

- For the lower convolutional layer, the sum of active regions significantly reveal the lesions which contributes to the classification results.
  Example - correctly predicted stage 4 samples
  - The quantity of interest: the predict outcome for stage 4.
  - Except for the optic disc, most of other highlighted regions of the heat map match the lesions. In general, the attribution method is capable of finding potential lesions.
  - See **Fig(a)** (left figure below), left heat map is the active regions of most positive neurons, and the right heat map is the active regions of most negative neurons.
  - The left one is dominated by green color: regions responsible for the stage 4 prediction.
  - The right one is dominated by red color: regions opposed for the stage 4 prediction.
- When inspecting a single neuron of these layers, the sensitive region is quite small and concentrate, which reflects the locality of convolutional layer neurons.
- Also calculate the sum heat map for different quantities of examples.
  Example – correctly predicted stage 2 samples
  - Quantity of Interest: Predict outcome for stage 0 & stage 2
  - See **Fig(b)** (right figure below), it's easy to find the symmetry – the region opposed the stage 0 outcome is similar to the region in favor of stage 4 and vice versa.
- For higher fully-connected layers, the active regions become more scattered, revealing the fact that they concentrate on more global and abstract features.



Hint:
Green for positive attribution.
Red for negative attribution.

Fig(a): Two examples for resultant attributing. For each one, upper left is origin image, upper right is preprocessed image, lower left is weighted sum of attribution for most positive influential neurons, lower right is weighted sum of attribution for most negative influential neurons. The neurons are from the last convolutional layer before blocks. The heat maps significantly catch vital lesions.



Fig(b): The sample '27790_right' is a correctly classified stage 2 sample. We visualize the attribution towards output for stage 0 and for stage 4, and the color of captured lesions are complementary and symmetry. It conforms to the meaning of attribution.

Generating Counterfactuals

- Because the integrated gradients indicate the contribution of each pixel, it's a useful way to generate counterfactuals.

Method

- For true label 1, predicted label 0 samples, calculate the attribution value of each pixel for stage 1 outcome.
  For positive pixel, move it away from the half-gray baseline.
  For negative pixel, move it close to the half-gray baseline.

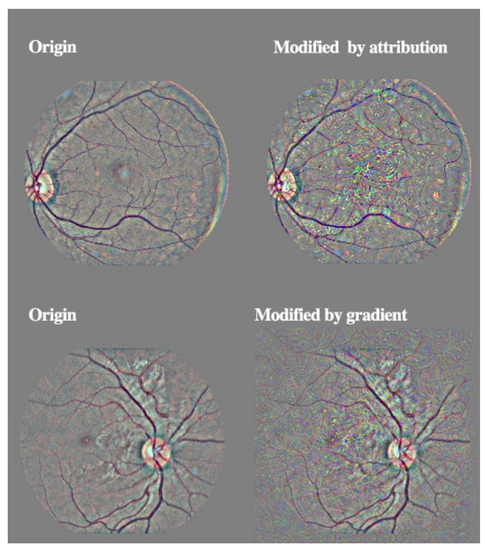
$$x'_{i,raw} = x_i + c \times sgn(x_i - baseline) \times IntegratedGrads_i(x),$$

$$x'_i = \max\left(\min(x'_{i,raw}, 255), 0\right),$$

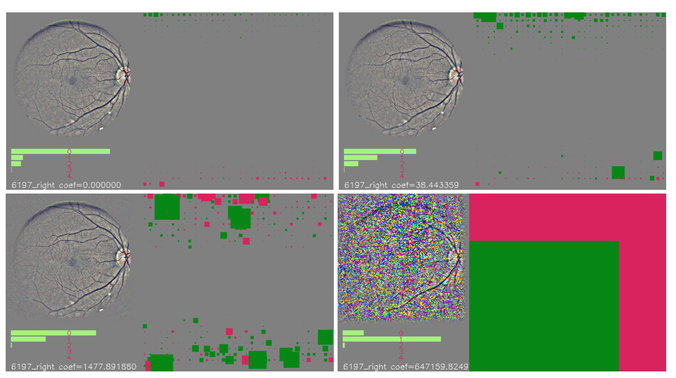
  - $sgn(\cdot)$ is the sign function,
  - c is a positive parameter to tune the degree of the modification.
  - Raw modified input is adjusted to legal range by the second equation.

Comparison

- Comparing to other counterfactual generating techniques such as adversarial learning, the sample generated by integrated gradients are closer to reality.
- The attribution technique doesn't only take gradient into consideration but also its value and intermediate points along the line, therefore the generated samples are smoother.



Linearity of the Model



- Size of square: the magnitude of the unit attribution
- Color: the sign
  Green for positive attribution. **Red** for negative attribution.

We use gradient to generate adversarial examples and visualize the attributions of intermediate layer units.

$$x'_{i,raw} = x_i + c \times \frac{\partial F(x)}{\partial x_i},$$

$$x'_i = \max\left(\min(x'_{i,raw}, 255), 0\right).$$

- When exponentially increased c becomes really large, the attributions of these units become tremendous large too.
- Considering the integrated gradient equation, the linearity between c and attribution value could only come from the difference term $(x_i - x'_i)$.

→ Average of gradients remains almost unchanged.
→ We can conclude the linearity between fully-connected layer units and the predict outcome.

Future Work

- There remains some special units whose attribution value change in different direction or remain constant.

There may be some interesting results if inspecting these units individually.

- Current work highlights all lesions.

It would be more powerful if new ways can explain individual neuron and distinguish different types of lesions.

- Develop more powerful model.

Explaining performance relies on the model performance. And we need model which predict stage 1 samples.