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

Linyi Li 1 2  Matt Fredrikson 1  Shayak Sen 1  Anupam Datta 1

Abstract
In this report, we applied integrated gradients to explaining a neural network for diabetic retinopathy detection. The integrated gradient is an attribution method which measures the contributions of input to the quantity of interest. We explored some new ways for applying this method such as explaining intermediate layers, filtering out unimportant units by their attribution value and generating contrary samples. Moreover, the visualization results extend the use of diabetic retinopathy detection model from merely predicting to assisting finding potential lesions.

1. Introduction
Diabetic retinopathy (DR), also known as diabetic eye disease, is an eye disease caused by diabetes mellitus (DM). It is a leading cause of blindness in the United States, often affecting working-aged adults. Among an estimated 10.2 million US adults 40 years and older known to have DM, the estimated crude prevalence rates for retinopathy and vision-threatening retinopathy were 40.3% and 8.2%, respectively. With proper treatment and monitoring of the eyes, at least 90% of new cases could be reduced.

Currently, detecting diabetic retinopathy is difficult, time-consuming and arduous. The manual process requires a trained clinician to examine and evaluate digital color fundus photographs of the retina. The micro aneurysms (microscopic blood-filled bulges in the artery walls), abnormal new blood vessels which leave a few specks of blood, and leaky macula blood vessels are major signs for detection. Therefore, the automated method for detection has long been recognized and previous efforts have made good progress. Along with the development of deep convolutional neuron networks, the previous model has achieved an accuracy of 73.76%.

We propose a network model which has an accuracy of 75.15%. After that, we apply integrated gradients method to the network in several different ways, which attributes the prediction of the model to input pixels. We visualize the attribution results by highlights and heat maps. The results significantly show the lesions of DR samples and reflect the model behavior. We also modify the examples according to their attribution weights. The modification serves as a sample generating process and the generated samples are more authentic than adversarial examples generated by fast gradient examples. Finally, we visualize the attribution value of intermediate layers for adversarial examples. The examples generated from fast gradient sign method cause great disparity of attribution among intermediate layer neurons. The disparity has an obvious increasing tendency along with the coefficient used for generating adversarial examples, which may be an evidence for highly linear nature of neural network model.

The remainder of this report is organized as follows. Section 2 presents an overview of related work, Section 3 introduces our model along with the training process in detail, Section 4 is a brief introduction of the attribution method we used – integrated gradients, Section 5 proposes the experiment data and visualization results, Section 6 is a general analysis of the experiment results and conclusion, and Section 7 issues the future work.

2. Related Work
The convolutional neural network (CNN) has recently developed as an important and useful method in image recognition, analysis and classification. Comparing to traditional classification models like regression, SVMs and random forests, CNN is able to extract high-level abstract features. It is relatively more independent from human effort and specific domain knowledge.

Some works have been made to apply CNN to diabetic retinopathy detection. A study by Stanford University (Alex et al., 2016) used transfer learning techniques based...
Explaining Diabetic Retinopathy Detection Deep CNNs via Integrated Gradients

on Inception V3 model. It reaches 72.96% accuracy on detecting referable DR (Stage 2, 3 or 4 DR) and 92.59% on detecting stage 4 DR. Harry Pratt, Frans Coenen and Deborah M Broadbent et al (Harry et al., 2016) designed a CNN network for five class DR detection task which enables the classifier to predict the exact DR stage of the sample. The classifier is more specific than other generalized ones. They achieved 73.76% accuracy, 29.99% sensitivity and 95.05% specificity.

In the field of machine learning system explaining and attributing, Mukund S, Abkur T, Qiqi Y et al (Sundararajan et al., 2017) proposed the method of integrated gradients. It combines the Implementation Invariance of Gradients along with the Sensitivity of techniques like LRP or DeepLift. Quantitative Input Influence proposed by Anupam Datta, Shayak Sen and Yair Zick (Datta et al., 2016) provides another method to measure the degree of influence of inputs on outputs of systems. Some case studies are included in their papers.

3. Dataset, Model and Training

3.1. Dataset and Preprocessing

We use a dataset of retina images from a recent Kaggle competition. The dataset is composed by colorized retina images taken from different cameras and have different size. There are about 35,000 images in training set and 55,000 images in test set. Images are labelled by an integer between 0 and 4, indicating the corresponding DR stage of the image. The image filename indicates whether it belongs to left or right eye. However, the difference is only the position of eyeball, which has been trained to ignore by image random rotation and flipping in data augmentation.

Before training, we preprocess the image. The preprocessing contains steps as follows: Firstly, we detect the actual retina diameter by counting the non-black pixels of the middle horizontal line. Secondly, calculating the Gaussian blurred image to filter out high-frequency noises of the image and make preparations for scaling down. Thirdly, use the diameter to scale down the retina to 256x256 size, which is the input shape of our network. Fourthly, calculate a new highly Gaussian blurred image and use the difference between the original image and blurred one as new one. Finally, crop out 10% of retina by radius to eliminate border noises and fill in the blank area with half-gray color. After these steps, images taken by different cameras which have various sizes and tones are now in the same style. The key features like lesions, vessels and bulges are highlighted. Some ideas of the preprocessing are from solution reports of leader teams of Kaggle diabetic retinopathy contest.

| Stage 0 |
| --- |
| Stage 1 |
| Stage 2 |
| Stage 3 |
| Stage 4 |

Figure 1. The raw image(scaled, left column) and preprocessed image(right column).

3.2. Model Structure

We use Google Inception V3 architecture trained for ImageNet Challenge as the base model. Then we remove the top output layer of the base model and add two full-connected layers over the base model for transfer learning. The first fully-connected layer has 1024 units, and the sec-

---

1Diabetic retinopathy develops in stages. Stage 0 means not having DR. Stage 1 is background retinopathy. Stage 2 is pre-proliferative retinopathy. Stage 3 is proliferative retinopathy. Stage 4 is diabetic maculopathy.

2Sensitivity = True, Predict DR / True DR; Specificity = True Not, Predict Not DR / True Not DR.

3https://www.kaggle.com/c/diabetic-retinopathy-detection
ond one has 512 units. They both use rectified linear unit as activation function and have a dropout rate of 0.25. For the output, we use a five-unit softmax classification layer, and each of unit represents the predict probability for each stage. The predict outcome is the stage number with maximum probability.

3.3. Training

For training, normal Adam optimizer is applied and categorical cross entropy is the loss function. The learning rate is initially set to 0.001. The training data is highly unbalanced, as about 71.4% samples are labelled as stage 0, comparing to that only about 2.02% samples are labelled as stage 4. Therefore, if trained with batch data directly drawn from training set, the classifier would be too biased toward stage 0. So, we divide the training set by the label and randomly select samples from each label set with fixed weights. 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. The weights are more balanced comparing to the raw dataset, but still far from full balance. It can prevent too much difference between dataset distribution and actual training distribution, and reflect a trade-off between balance and accuracy.

We also use data augmentation for better fitting ability. The image is rotated by a random angle and randomly flipped horizontally and vertically. Because left eye and right eye images are horizontal symmetry, the augmentation eliminates the difference and generalizes the model.

Because the base model has been trained ideally, and the low-level features of image are transferable, we fix the weights of lower 8 blocks as well as bottom convolutional layers. The trainable parameters are in top 2 blocks of base model and newly added fully-connected layers.

The length of training process is not measured by epochs but by mini-batches because the adjustment by weight makes the training data be picked out randomly, not sequentially. The mini-batch size is 256 samples. At the beginning of each training phase, we pick out 20% of training data as validation set. The first phase trains 550 mini-batches and the learning rate is 0.001. The accuracy on training set reaches about 75% but only 65% on validation set. The second phase trains 350 mini-batches and the learning rate is 0.0002. The accuracy on training set is about 77.5% and 72.5% on validation set. The third phase trains 495 mini-batches with learning rate 0.00005. The training set accuracy is about 80% and validation set accuracy is about 75%. The whole training process takes about 1 day on NVIDIA Titan X(Pascal), Core i7-7700K. We use Keras library with Theano as backend.

3.4. Model Performance

The performance of model is measured on the whole test set of Kaggle contest. It contains 53574 images. Preprocessing is also applied before fed to the trained model. We measure the performance by accuracy, sensitivity, specificity and kappa score. Accuracy is defined as the percentage of correctly classified samples in all samples. Sensitivity is defined as the percentage of samples which correctly predicted as diabetic retinopathy (belonging to stage 1, 2, 3 or 4) among all samples which has diabetic retinopathy. And Specificity is defined as the percentage of correctly predicted health samples among all health samples (belonging to stage 0).

The definition of kappa score is from Kaggle competition. It is a type of quadratic weighted kappa. It is calculated as
Finally, the kappa score is the normalized histogram of ratings. Thirdly, define 

\[ w_{i,j} = \frac{(i - j)^2}{(N - 1)^2} (N = 5) \]  

(1)

Finally, the kappa score is

\[ \kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \]  

(2)

Our confusion matrix is below 1.

And the indexes for the baseline from the paper (Harry et al., 2016) and our model are as follows 2.

| Baseline (Harry et al., 2016) | Our Model |
|------------------------------|-----------|
| **ACCURACY**                 |           |
| 73.76%                       | 75.1540%  |
| **SENSITIVITY**              |           |
| 95.05%                       | 96.1500%  |
| **SPECIFICITY**              |           |
| 29.99%                       | 31.1423%  |
| **\( \kappa \)**            |           |
| 0.4398                       | 0.5107    |

Table 2. Some Indexes of the baseline model and our model.

The integrated gradients method attributes the outcome value \( F(x) \) to the input. It is a significant indicator telling us the contributions of each component of the input to the quantity. Informally, if the attribution value for a component is positive, the difference between the component and baseline is of the same sign as the gradient integration. Therefore, its actual value goes along the direction which increases the quantity. On the contrary, if the attribution value is negative, the actual value goes opposite to the direction which increases the quantity. Thus, more positive the attribution value is, the more it contributes to the increase of the quantity. And the more negative the attribution value is, the more it contributes to the decrease of the quantity.

Integrated gradients method is an attribution method for black box system which is fit for learning systems. It maintains several axioms and properties. For example, it satisfies completeness: if the quantity of interest \( F(\cdot) \) is differentiable almost everywhere, then

\[ \sum_{i=1}^{m} \text{IntegratedGrads}_i(x) = F(x) - F(x') \]  

(4)

Moreover, the method aggregates the gradients along the straight line between the baseline and the input and is a type of path methods. Here, the path is the simplest straight line. Comparing to other path methods, the integrated gradients not only satisfies some axioms of path methods such as sensitivity and linearity, but also stands out by its unique symmetry-preserving property.

4.2. Utilizing Integrated Gradients

The integral of integrated gradients can be numerically approximated efficiently. We uniformly pick out points on the straight line from the baseline to the input and calculate the gradient of \( F(x) \) on these points. Then,

\[ \text{IntegratedGrads}_i(x) \approx (x_i - x'_i) \times \frac{\int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \, d\alpha}{m} \]  

(5)

We set \( m \) to 50 during calculation as its sufficient for satisfactory precision in practice.

Due to the preprocessing process, the images are expected to be symmetric about the half-gray color. The background

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

(3)
color outside the retina is also half-gray. Therefore, full half-gray image is chosen as baseline, which conveys no information, rather than the frequently used all-zero baseline.

The paper (Sundararajan et al., 2017) already shows some applications of integrated gradients, including those on images recognition task and those on diabetic retinopathy classifier. We explore and harness its advantages in diabetic retinopathy task in detail.

We find the use of attribution method should not limited to input layer only. It can also be a good way to evaluate the contributions for each unit of intermediate layers. In this way, we use the method to help us understand intermediate layers and internal structure of the inspected model. We get some interesting results and some previous conclusions and presumptions are verified. Section 5 shows the detail.

5. Experiments and Results

5.1. Highlight Lesions

The lesions of micro aneurysms and leaky or blocked blood vessels are decisive symbols for diabetic retinopathy detection. In the retina image, these lesions are apparent local features. Therefore, they are expected to have large contributions to the predict outcome of severe diabetic retinopathy stages. The attribution value of these lesion pixels should be a relatively large positive value which distinguish them from other normal pixels. We visualize the attribution value by heat map. The positive pixels are colored red and negative ones are colored green. The intensity stands for the magnitude and is normalized per example.

We firstly attribute the predict outcome directly to input pixels, which is as same as previous applications. However, the results are not as ideal as we expected. The regions are scattered. And the key lesions are largely mixed with subtle turbulences, which makes it difficult to capture decisive features. Moreover, the sign of attribution value appears unstable, i.e., they vary among neighboring pixels. We split the positive and negative pixels to different pixels and find that they are of similar distribution and largely coincide with each other. Without loss of generality, the examples from true label stage 4 and predicted label stage 4 reflect these observations.

The reason for the scattered distribution could be the gradients of single pixel are too local and the effect of variation of single pixel could not reflect the global contribution of the lesion the pixel lies in. Therefore, the attribution value can reveal the sensitivity of the pixel to the quantity of interest in some degree, but the local noise interferes its ability for detecting vital lesions.

Therefore, we turn to explain intermediate layers. Similarly, the integrated gradient for each neuron reflects its contribution to the quantity of interest. We firstly 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 like half-gray baseline of input layer. After that, the instance is inputted and instance activations are figured out. Then we use the same equation to calculate the attribution values for each neuron.

We also use integrated gradients to figure out what pixels are responsible for the intermediate layer neuron. We set the activation of the intermediate layer neuron we are interested in as the quantity of interest, and calculate the input layer attribution value relative to it.

We assume that the lesions are effectively captured by the most important intermediate neurons, so we sort neurons by their attribution value, and pick out top several most

![Figure 4. Two examples for direct attributing. For each one, upper left is origin image, upper right is preprocessed image, lower left is positive pixels, lower right is negative pixels. The distribution is scattered and disperse.](image-url)
positive ones at head and top several most negative ones at negative for computing. The selecting ratio is 10% respectively for fully-connected layers, and becomes much lower for convolutional layers to guarantee no much than 1,000 neurons are selected because computing time grows linearly with the number of selected neurons. We try to visualize the influential neurons by their active region as well as their resultant effects by computing the sum of active regions weighted by the neuron attribution value.

For the lower convolutional layer, the sum of active regions significantly reveal the lesions which contributes to the classification results. For example, for correctly predicted stage 4 samples, except for the optic disc, most of other highlighted regions of the heat map match the lesions. The quantity of interest is the predict outcome for stage 4. See Fig. 5, the 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, which reveals that these neurons and these regions are responsible for the stage 4 prediction. The right one is dominated by blue color, which in the same way tells what opposes the stage 4 prediction. The left heat map and right heat map both capture potential lesions, but the ones captured by right heat map actually oppose the stage 4 prediction. One possible explanation for this could be that these lesions are thought to be indicators of stage 1, 2 or 3 by the classifier. Another explanation could be the fitting strength shortage of the classifier itself. In general, the attribution method is capable of finding potential lesions.

Moreover, when inspecting a single neuron of these layers, the sensitive region which activates them is quite small and concentrate, which reflects the locality of convolutional layer neurons. See Fig. 6.

We also calculate the sum heat map for different quantity of examples. The picture is heat maps for correctly predicted stage 2 examples with respect to the predict outcome of stage 0 and stage 4. Comparing the results for the two quantities, its 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. See Fig. 7.

For higher fully-connected layers, the locality of single unit decreases. The active regions become more scattered, which reveals the fact that they concentrate on more global and abstract features.

5.2. Generate Contrary Samples

Because the integrated gradients indicate the contribution of each pixel, its a useful way to generate contrary samples. Our experiment starts with wrongly classified examples which have true label stage 1 but misclassified as stage 0. We calculate the contribution 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. Formally,

\[ x'_{i, raw} = x_i + c \times \text{sgn}(x_i, \text{baseline}) \times \text{IntegratedGrads}_i(x), \]

\[ x'_i = \max(\min(x'_{i, raw}, 255), 0), \]

where \( \text{sgn}(\cdot) \) is the sign function, and \( c \) is a positive parameter used for tuning the degree of the modification. The raw modified input is adjusted to the legal range. According to the meaning of integrated gradients, the quantity for the new instance tends to get larger along with the increase of \( c \).

We expect the modification could give us some stage 1 predicted examples because the original model doesn’t classify any example as stage 1. The outcome of modified samples shifts directly from stage 0 to stage 2, 3 or 4. This may be caused by the unbalanced training data which let other stage outputs become more sensitive to lesions. Even though, the generated samples are still capable of serving as contrary samples. Comparing to other counter sample generating techniques such as adversarial learning, the sample generated by integrated gradients are closer to reality, which can be revealed in the figure. 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. See Fig. 8.

We adjust the parameter \( c \) and plot the prediction outcome. Most of involved samples show the same tendency – at first stage 0 outcome dominant as it is originally predicted as, then stage 2 or 3 outcome rises and takes over the position, then followed by stage 4, finally the noises of gradient numeric result are amplified when \( c \) gets too big and the random noise sample is predicted as stage 0.

5.3. Verify Linearity of Model

The paper (Goodfellow et al., 2014) proposes an efficient way to generate adversarial examples. The method computes the gradient and creates slight perturbation which maximize the confidence for another label. Concluded by the paper, this method works because of the linearity of neural networks.

Inspired by the conclusion, we use gradient to generate our adversarial examples and visualize the attributions of intermediate layer units. Formally, the examples are generated as follows:
\[ x_{i,raw}' = x_i + c \times \frac{\partial F(x)}{\partial x_i}, \quad (8) \]
\[ x_i' = \max \left( \min \left( x_{i,raw}' , 255 \right), 0 \right) . \quad (9) \]

Here, \( c \) is a positive parameter and \( x_{i,raw}' \) is adjusted to the normal range \(-x_i'\).

In this way, we randomly select some stage 1 samples which are classified as stage 0, and successfully let most of samples be predicted as stage 1 only through tiny perturbation. We then continue increasing parameter \( c \), and find that the attribution of a few intermediate layer units grows linearly along with \( c \). When exponentially increased \( c \) becomes really large, the attributions of these units become exaggeratedly large too. This is revealed in the figure, where the size of square stands for the magnitude of the unit attribution and color stands for the sign. Considering the integrated gradient equation 3, the linearity of \( c \) and attribution value could only come from the difference term \( (x_i - x_i') \), which means the average of gradients remains almost unchanged. Thus, the linearity between fully-connected layer units and the predict outcome is verified. See Fig. 9.

6. Conclusions and Discussions

In this report, we explored the applications of integrated gradients on diabetic retinopathy detection. We trained a neural network using transfer learning to classify retina images to different stages of diabetic retinopathy. Then we tried several different ways to using integrated gradients for detecting the lesions. Directly attributing output quantity to input pixels is not as good as expected, because the highlights are too dispersed and vital lesions are hard to be distinguished from minor features. However, attributing output quantity to intermediate layer works better, especially for convolutional layer. Only most influential neurons are selected and the vital lesions are obvious when attributing only these neurons. Currently, the learning system has not been strong enough to be totally replied on. Even though, our results show a possible way for doctors and clinicians to use it as an important aid for diagnosis which prevents them from neglecting potential lesions.

Besides, we use the attribution method to generate contrary examples and trace the changes of neurons. In this process, we find attribution method a good way to generate nearly authentic contrary examples. Moreover, the increase of attribution value when generating adversarial examples show the linearity of the model.

There are still many unanswered questions we need to figure out. The input image has three channels and the attribution results for each pixel also have three channels.

The value for each channel are usually quite different and even different in symbols. Thus, an intuitive explanation and visualization for attribution value is not easy. Simple methods such as maximum, minimum and average all suffer from information loss. Another problem is the coincidence of positive regions and negative regions. Empirically this should not happen because the lesions are responsible for disease stages and normal vessels are responsible for the health stage, which could not coincide in normal cases.

7. Future Work

Besides the questions above, we are interested in applying integrated gradients to some special units. When generating examples by several ways, we found 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.

Although integrated gradients can significantly capture vital lesions, there are still some vital lesions missed. And it takes much time to calculating the attribution for a large number of intermediate layers. This process mainly consumes CPU resource and each sample needs 4 hours of computing on Core i7-7700K. Therefore, there remains room for improvement.

Acknowledgements

Thanks to CyLab, Carnegie Mellon University for offering me such a nice place for summer internship. Thanks to Chuang Shi Jie program of Tsinghua University for the funding.

References

Alex, T., Iain, U., and Chala, F. Deep CNNs for diabetic retinopathy detection. CS229 Project Report, 2016, Stanford, 2016.

Datta, Anupam, Sen, Shayak, and Zick, Yair. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In IEEE Symposium on Security and Privacy, pp. 598–617. IEEE Computer Society, 2016. ISBN 978-1-5090-0824-7. URL http://dblp.uni-trier.de/db/conf/sp/sp2016.html#DattaSZ16.

Goodfellow, Ian J., Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. CoRR, abs/1412.6572, 2014. URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#GoodfellowSS14.

Harry, P., Frans, C., Deborah, M. B., Simon, P. H., and Yalin, Z. Convolutional neural networks for diabetic...
Explaining Diabetic Retinopathy Detection Deep CNNs via Integrated Gradients

Kempem, J. H., O’Colmain, B. J., Leske, M. C., Haffner, S. M., Klein, R., Moss, S. E., Taylor, H. R., and Hamman, R. F. The prevalence of diabetic retinopathy among adults in the United States. *Archives of Ophthalmology*, 122(4):552–563, 2004.

Sundararajan, Mukund, Taly, Ankur, and Yan, Qiqi. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017. URL http://dblp.uni-trier.de/db/journals/corr/corr1703.html#SundararajanTY17.

Tapp, R. J., Shaw, J. E., Harper, C. A., de Courten, M. P., Balkau, B., McCarty, D. J., Taylor, H. R., Welborn, T. A., and Zimmet, P. Z. The prevalence of and factors associated with diabetic retinopathy in the Australian population. *Diabetes Care*, 26(6):1731–1737, 2003.

Zhiguang, W. and Jianbo, Y. Diabetic retinopathy detection via deep convolutional networks for discriminative localization and visual explanation. arXiv preprint arxiv: 1703.10757, 2017, 2017.

Figure 5. 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 (see Fig. 2). The heat maps significantly catch vital lesions.
Explaining Diabetic Retinopathy Detection Deep CNNs via Integrated Gradients

Figure 6. This is ‘17099_left’, a correctly predicted stage 4 sample. The influential units all focus on their own fixed small regions because they lie in a lower convolutional layer.

Figure 7. 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.

Figure 8. Sample generated from attribution vs. generated from adversarial attack(gradient). Notice: the degree of modifying is decided by coefficient in both case, but the meaning of coefficient is different and incomparable between them.
Figure 9. The size of small square in the right part resembles the attribution value. At first, the attributions are relatively even. However, as we modify the image using adversarial gradient (Goodfellow et al., 2014), some squares grow magically, then even occupy the whole part of images. It reveals the linearity of the model.