Use of Quantified Uncertainty in Integrated Gradient Attribution Baselines

David Drakard

Abstract

As the power and complexity of deep neural networks increases, interpretability is becoming a more important consideration. Predictions may not be usable in real world contexts if they cannot be sufficiently justified. Recent research has developed numerous methods for attributing predictions to features. Many of these methods require a baseline, an input value which must in some sense represent missingness. The choice of baseline affects the attribution result, and optimal baseline choice is an area of active research. Baselines have previously been drawn from the true input domain, and therefore cannot truly represent missing data. We introduce artificial certainty parameters to the data, expanding the input space. This expanded space contains a maximal uncertainty value which is a natural baseline.

1 Introduction

Attribution, the identification of the input features most salient to a model prediction, is an increasingly important requirement for neural networks. It is a core part of model interpretability, which is valuable as a research tool and design aid, but also increasingly as an output requirement in its own right.

Requires citations

European union right to explanation [4], explainable machine learning for loan applications [1]

Gradients of predictions with respect to model inputs can be calculated using backpropagation. These gradients have been used for feature attribution[7]. The gradients indicate which features are most sensitive to a perturbation, causing the largest change in prediction outcome. Empirically, this method does often highlight areas salient to prediction. It does not always identify the most important areas, due to saturation of the gradients. For example, a dark area may be the most distinguishing characteristic of an object. If the variation between black and dark grey is unimportant, gradients may not indicate this dark area when it is black. Sensitivity is not identical to saliency.

The method of *integrated gradients*[10] develops a method of attribution based on several theoretical justifications, and overcomes gradient saturation. The method integrates the gradients as the input varies linearly between a *baseline* and the input of interest:

$$\text{IntegratedGradient}(x,i) = \underbrace{(x_i - x_i')}_{\text{distance}} \underbrace{\int_{\alpha=0}^{1} \frac{\partial F\left(x' + \alpha\left(x - x'\right)\right)}{\partial x_i} d\alpha}_{\text{mean gradient}} \tag{1}$$

where F is the prediction of the neural network, x is the input vector of interest, x' is the baseline input vector, and i is the feature of interest. A numerical approximation is normally used to calculate the integral in practice. This method is a valuable advance in attribution, but unlike simple gradients it requires a choice of baseline.

2 Choice of baseline

Theoretical concerns of baselines, asymmetry

Practical concerns of baselines, matching areas of image missing. Can mention "natural" baseline choices such as black.

Effective choice of baselines in practice is a topic of ongoing research. A recent review of four state-of-the-art models for tabular data did not find a best performing choice[5]. As this paper noted, the concept of missingness within an arbitrary space is domain-specific. Therefore applying attribution techniques to a new domain requires this concept to be determined, if the baseline is to be taken from within the input space. Another review came to similar conclusions in the context of image classification attribution[8].

2.1 Previously investigated baselines

3 Artificial uncertainty baselines

Introduce a concept of uncertainty. The uncertainty baseline will be the input vector having maximum uncertainty

To implement aritificial uncertainty we must define an *augmented input* space $\mathcal{A}$ based upon the *real input* space $\mathcal{R}$, where the real input space is simply the usual space of input vectors from the original distribution. To form $\mathcal{A}$ associate a certainty value in the range $[0, 1]$ with each 'independent' component of the real input vector. 'Independence' in this context can sometimes be chosen using domain specific knowledge, but there is a general choice which is always suitable: the basis components of the real input vector space. Note that this independence merely describes an implementation choice of aritifical uncerainty, it does not pertain to statistical independence.

For example, an image with height h, width w, and three colour channels represented by a vector of dimension $h \times w \times 3$ would under the default augmentation process have a certainty associated with each colour channel of each pixel, to produce an augmented vector of dimension $h \times w \times 3 \times (1+1)$. If the model designer decides instead the three colour channels should be considered 'dependent', they may associate a certainty value with each whole pixel, producing an augmented vector of dimension $h \times w \times (3+1)$. In either case, the augmented input space will be suitable to produce an uncertainty baseline, and an attribution value will be calculated for each independent component.

After defining the space, uncertainty semantics must be applied. This is done during model training. The model is trained on *damaged* input vectors:

$$x^{\mathcal{D}} = D(x^{\mathcal{R}}) : x^{\mathcal{D}} \in \mathcal{A}, x^{\mathcal{R}} \in \mathcal{R} \tag{2}$$

where D is a *damage function*.

As $x^{\mathcal{D}}$ are members of $\mathcal{A}$ the model's input layer size must accommodate this. D is evaluated in two steps, any specific D being determined by the combination.

A *certainty* $x^{\mathcal{D}}_{i,c}$ is generated for each independent component $x^{\mathcal{D}}_i$. A simple option is to sample from $U(0,1)$:

$$x^{\mathcal{D}}_{i,c} \leftarrow U(0,1) \tag{3}$$

Then the *value* $x^{\mathcal{D}}_{i,v}$ for each $x^{\mathcal{D}}_i$ is determined. Generate a random number r_i from $U(0,1)$. If $r_i \leq x^{\mathcal{D}}_{i,c}$, $x^{\mathcal{D}}_{i,v}$ is just the component of the real input vector $x^{\mathcal{R}}_i$. If $r_i > x^{\mathcal{D}}_{i,c}$, choose $x^{\mathcal{D}}_{i,v}$ from an alternative distribution, statistically independent of $x^{\mathcal{R}}_i$. For a simple case where the independent components are the basis components of $\mathcal{R}$ and the alterantive distribution is $U(0,1)$:

$$\begin{aligned} r_i &\leftarrow U(0,1) \\ x^{\mathcal{D}}_{i,v} &\leftarrow \begin{cases} x^{\mathcal{R}}_i & r_i \leq x^{\mathcal{D}}_{i,c} \\ U(0,1) & r_i > x^{\mathcal{D}}_{i,c} \end{cases} \end{aligned} \tag{4}$$

A visualisation of this process is shown in figure 1. It shows an image from the MNIST database[6] in false colour, before and after damage.

The features $x^{\mathcal{D}}_i$ of an input prepared in this way have a varying chance, quantified by the certainty $x^{\mathcal{D}}_{i,c}$, to give information about the original input $x^{\mathcal{R}}$, and therefore the label y. During training the model learns to ignore features unrelated to y, discounting features with lower certainty, and develops the required uncertainty semantics.

In practice the simple damage function described above may not influence a model to learn uncertainly semantics sufficiently strongly.

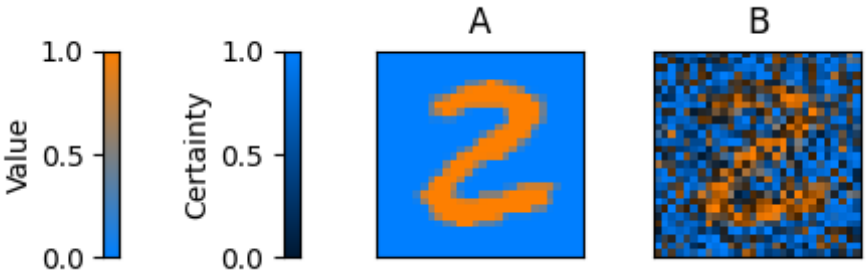

Figure 1: Visualisation of a damage function D applied to a greyscale image in the MNIST database. A: real input $x^{\mathcal{R}}$. B: damaged input $x^{\mathcal{D}}$.

Learning rate and accurancy can be improved by using an alterantive, such as using other samples from the training dataset as adversarial alternative distributions. We found training to be more effective using certainty $\sim U\{0, 1\}$ and adversarial alternative distributions.

$\mathcal{A}$ contains a natural baseline choice for any $x^{\mathcal{R}}$: $x^{\mathcal{B}} \in \mathcal{B}$:

$$x^{\mathcal{B}}_{i,v} = x^{\mathcal{R}}_i \tag{5}$$

$$x^{\mathcal{B}}_{i,c} = 0 \tag{6}$$

The better the model is trained, the less it will demonstrate a relationship between any $x^{\mathcal{B}}_{i,v}$ and y:

$$\frac{\partial F(x^{\mathcal{B}})}{\partial x^{\mathcal{B}}_{i,v}} \approx 0 \tag{7}$$

4 Integrated Certainty Gradients

Let $\mathcal{R}' \subset \mathcal{A}$ be the set of maximally certain input vectors and

$$x^{\mathcal{R}'}_{i,v} = x^{\mathcal{R}}_i \tag{8}$$

$$x^{\mathcal{R}'}_{i,c} = 1 \tag{9}$$

Integrated Certainty Gradients is then the integrated gradient from $x^{\mathcal{B}}$ to $x^{\mathcal{R}'}$.

$$\text{I.C.G.}(x, i) = \int_{\alpha=0}^{1} \frac{\partial F\left(x^{\mathcal{B}}_i + \alpha\left(x^{\mathcal{R}'}_i - x^{\mathcal{B}}_i\right)\right)}{\partial x_{i,c}} d\alpha \tag{10}$$

Only the mean gradient term is retained from Equation 1 because the distance term $|x_i - x'_i| = |x^{\mathcal{R}'}_i - x^{\mathcal{B}}_i| = 1$.

To calculate the integral in Equation 1 and Equation 10 a numerical approximation is needed. For Integrated Certainty Gradients variants of the Riemann Sum approximation are used:

$$\int_{\alpha=0}^{1} F(\alpha)d\alpha \approx \frac{1}{m}\sum_{\alpha=1}^{m} F(\alpha) \text{ for large } m \tag{11}$$

Using this with Equation 10 gives:

$$\text{I.C.G.}(x,i) \approx \sum_{\alpha=1}^{m} \frac{\partial F\left(x_i^{\mathcal{B}} + \alpha\left(x_i^{\mathcal{R}'} - x_i^{\mathcal{B}}\right)\right)}{\partial x_{i,c}} \text{ for large } m \tag{12}$$

Integrated Certainty Gradients is an Integrated Gradients method, which are themselves part of *Shapely values* attribution method family. Shapely values methods uniquely have a set of mathematically proven properties desirable for attribution, listed in Appendix A. Integrated Gradients also do not suffer from gradient saturation, unlike some other gradient methods. Integrated Certainty Gradients inherits these desirable properties.

Add reference

The Integrated Certainty Gradients method was evaluated on a text classification task using the MNIST dataset[6]. A convolutional neural network was trained for this purpose, using damaged inputs as described above. Its architecture is given in Appendix B.

An example attribution using Integrated Certainty Gradients on this model is shown in Figure 2. Image A shows the input and image D shows the calculated attribution. The model correctly recognized the the input's class ("2"). Basic features of attribution are exhibited. The central region of the image, which is subject to variation in the input dataset, and corresponding to 'the area where the number is' in human recognition, is indicated. Peripheral regions, which do not vary in the input dataset, and cannot distinguish the image from others, are not indicated. The second highest predicted class for this image is "3". The areas on negative attribution visually correspond to those matching a "3". This type of subjectively reasonable result is characteristic of attributions across images in the dataset.

Measure effectiveness by ablation: allocate a number of pixels in proportion to calculated attribution.

Computational performance of the technique is good. The integral approximation is calculated using multiple interpolations. In standard Integrated Gradients, manyinterpolations are recommended, which increases the likelihood of averaging gradients fairly in the case of a highly nonlinear relationship between input and output. The gradients do vary as across the uncertainty range in Integrated Certainty Gradients, but this variation appears to be rather smooth in the cases

If possible test on another model (larger resolution with richer contents) where attribution quality is more obvious

cite how many

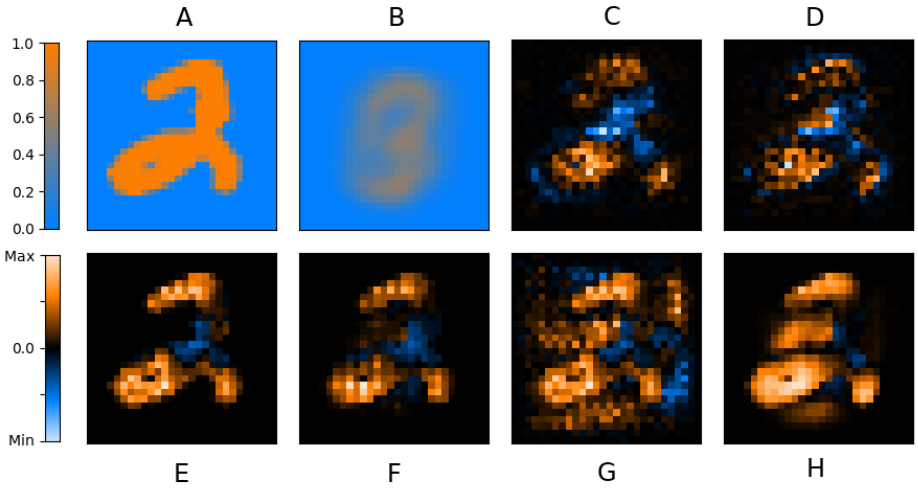

Figure 2: Attribution results for an example image. Values are scaled separately for each attribution method for greatest visual clarity. A: The input image. B: Mean value of the input dataset. C: Probability-value gradients. D: Integrated Certainty Gradients. E: Zero baseline Integrated Gradients. F: Mean baseline Integrated Gradients. G: Double sided Integrated Gradients. H: Expected Gradients.

investigated, and attributions with as few 5 interpolations closely approximate those with 100 approximations.

4.1 Comparison with other methods

refer back and compare with section Previously investigated baselines

The value of Integrated Certainty Gradients is assessed relative to other gradient based attribution methods: simple gradients, Integrated Gradients with several baselines, and Expected Gradients. Except when performing Integrated Certainty Gradients, certainty values $x_{i,c}$ were kept at 1.0 at all times.

Simple gradient attribution is shown in image C of Figure 2. A relatively close correlation with Integrated Certainty Gradients is seen in this case. Good correlation is typical throughout the dataset although this case is particularly close. A notable difference is seen at the center: the simple gradients are uniformly negative, whereas I.C.G. are positive in the dark regions and negative in the light regions. This shows I.C.G. is indicating attribution: dark regions where darkness is "desired" (negative gradient) indicate positively, light regions where

darkness is desired indicate negatively. Unlike all other methods that will be discussed including I.C.G., simple gradients is not an Integrated Gradients nor a Shapely values attribution method, and does not share the theoretical strengths of those methods.

Standard Integrated Gradients with 3 choices of baseline are now considered.

An example of Integrated Gradients using a zero baseline is shown in image E of Figure 2. Over the high value areas of the input (image A) the result generally correlates with simple gradients and I.C.G. This example notably exhibits the weakness of baseline dependent methods: the zero-value areas of the input image exhibit no attribution because they do not differ from the baseline. The variants of standard Integrated Gradients that follow intend to mitigate this problem. Due to this limitation and better alternatives, use of a simple zero baseline will not be considered again.

A simple alternative to a constant baseline is use of a double-sided baseline: performing Integrated Gradients both with the minimal and maximal (constant 0.0 and 1.0 in this case) baselines and taking the sum of the result. This treats features more fairly in a sense because the total integrated distance ($|x_i - x'^{+}_i| + |x_i - x'^{-}_i|$) is constant for all x_i. This technique has not to our knowledge been used before. The result of this method is shown in image G of Figure 2. Attribution is no longer restricted to the high-value area of the input. This method gives the most disbursed allocation of attribution, which will be discussed further in section 4.2.

Expected gradients [2]

Note: not deterministic.

4.2 Out-of-distribution attribution

Determine ideal terminology

Interventional vs Observational conditional expectation. From paper "True to the Model or True to the Data" -¿ cited in paper "SHAPLEY EXPLAINABILITY ON THE DATA MANIFOLD"

4.3 Applicability

5 Other applications of artificial uncerainty

[3] Discussion of on and off manifold. Is on/off manifold or in/out of distribution preferred terminology?

off manifold data question: we can't know the underlying manifold so all we can answer is in test data or out of test data but for neural network to have any predictive power it must venture outside

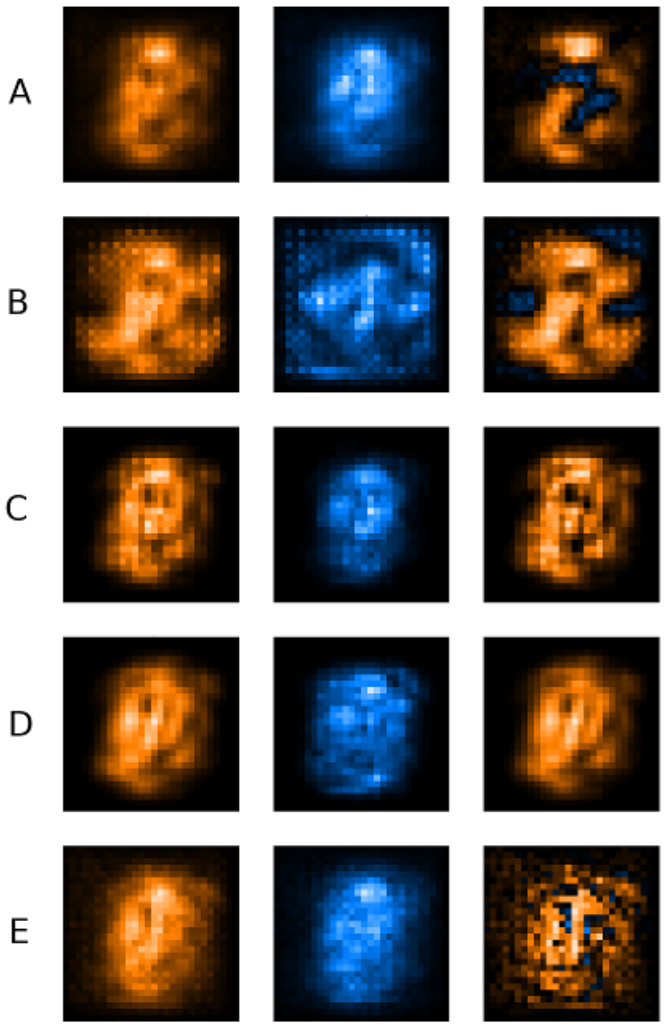

Figure 3: Mean attribution over randomly sampled images. Left column: mean over positive values. Center column: mean over negative values. Right column: mean over all values. Row A: probability-value gradient over 1000 images. Row B: Double sided Integrated Gradients over 600 images, 100 interpolations. Row C: Integrated Gradients with distribution-mean baseline over 1000 images, 1000 baseline images, 100 interpolations. Row D: Expected Gradients over 600 images, 600 interpolations/baselines. Row E: Integrated Certainty Gradients over 600 images, 100 interpolations. Values are scaled independently per image for maximum contrast, for comparison of attribution distributions within each image, not values between images.

6 Conclusion

References

[1] Umang Bhatt et al. *Explainable Machine Learning in Deployment*. 2020. arXiv: 1909.06342 [cs.LG].

[2] Gabriel Erion et al. *Improving performance of deep learning models with axiomatic attribution priors and expected gradients*. 2020. arXiv: 1906.10670 [cs.LG].

[3] Christopher Frye et al. *Shapley explainability on the data manifold*. 2020. arXiv: 2006.01272 [cs.LG].

[4] Bryce Goodman and Seth Flaxman. "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"". In: *AI Magazine* 38.3 (Oct. 2017), pp. 50–57. ISSN: 0738-4602. DOI: 10.1609/aimag.v38i3.2741. URL: http://dx.doi.org/10.1609/aimag.v38i3.2741.

[5] Johannes Haug et al. *On Baselines for Local Feature Attributions*. 2021. arXiv: 2101.00905 [cs.LG].

[6] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[7] K Simonyan, A Vedaldi, and A Zisserman. "Deep inside convolutional networks: visualising image classification models and saliency maps". In: ICLR, 2014, pp. 1–8.

[8] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. "Visualizing the Impact of Feature Attribution Baselines". In: *Distill* (2020). https://distill.pub/2020/attribution-baselines. DOI: 10.23915/distill.00022.

[9] Mukund Sundararajan and Amir Najmi. "The Many Shapley Values for Model Explanation". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 9269–9278. URL: http://proceedings.mlr.press/v119/sundararajan20b.html.

[10] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *CoRR* abs/1703.01365 (2017). arXiv: 1703.01365. URL: http://arxiv.org/abs/1703.01365.

Appendix A Mathematical properties of Shapely attribution methods

Say shapely methods are the only ones satisfying all these properties.

List the shapely properties, and for each give a motivating example

Appendix B Network architecture

The MNIST classification model used was a feed forward convolutional network, with the following layers:

- Input
  $28 \times 28 \times 2 = 1568$ dimensional
- 2D convolution, 3×3 kernel, stride 1, 32 channel output, ReLu activation
  $26 \times 26 \times 32$
- 2D convolution, 3×3 kernel, stride 1, 64 channel output, ReLu activation
  $24 \times 24 \times 64$
- 2D max pooling, 2×2 kernel, stride 2
  $12 \times 12 \times 64$
- Dropout layer (training only), 25% drop rate
- Dense layer, ReLu activation
  128
- Dropout layer (training only), 50% drop rate
- Dense layer, Softmax activation
  10