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

Gradients of predictions with respect to model inputs can be calculated using backpropagation. These gradients have been used for feature attribution[4]. 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*[6] 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:

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

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.

In practice the mean gradient is calculated using a discrete approximation. There are numerous methods of numerically approximating integrals. Using the Reimann sum method to calculate the mean gradient:

$$\text{I.G.}^{approx}(x,i) = (x_i - x_i')\frac{1}{m}\sum_{\alpha=1}^{m} \frac{\partial F\left(x' + \frac{\alpha}{m}\left(x - x'\right)\right)}{\partial x_i} \tag{2}$$

In this work we used both Riemann sum and trapezium rule methods.

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[2]. 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[5].

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. 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:

$$x^{\mathcal{D}} = D(x^{\mathcal{R}}) : x^{\mathcal{D}} \in \mathcal{A}, x^{\mathcal{R}} \in \mathcal{R} \tag{3}$$

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 pair.

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

$$x^{\mathcal{D}}_{i,c} \leftarrow U(0, 1) \tag{4}$$

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, 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 $U(0, 1)$:

$$\begin{gathered} r_i \leftarrow U(0, 1) \\ x^{\mathcal{D}}_{i,v} \leftarrow \begin{cases} x^{\mathcal{R}}_i & r_i \leq x^{\mathcal{D}}_{i,c} \\ U(0, 1) & r_i > x^{\mathcal{D}}_{i,c} \end{cases} \end{gathered} \tag{5}$$

A visualisation of this process is shown in figure 1. It shows an image from the MNIST database[3] 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 developing the required uncertainty semantics.

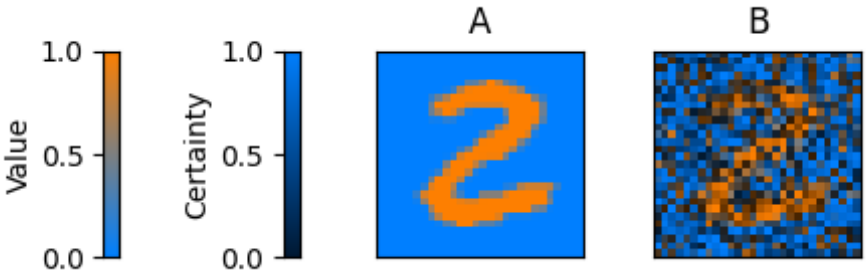

Figure 1: Visualisation of a damage function D applied to a greyscale image in the MNIST database. A: real input $x^{\mathcal{R}}$. B: damaged input $x^{\mathcal{D}}$.

In practice the simple damage function described above may not influence a model to learn uncertainly semantics sufficiently strongly. Learning rate and accurancy can be improved by using an alterantive, such as using other input samples 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 $x^{\mathcal{R}}$: $x^{\mathcal{B}} \in \mathcal{B}$:

$$x^{\mathcal{B}}_{i,v} = x^{\mathcal{R}}_{i} \tag{6}$$

$$x^{\mathcal{B}}_{i,c} = 0 \tag{7}$$

If the model is trained well, it will demonstrate negligable relationship between any $x^{\mathcal{B}}_{i,v}$ and y:

$$\frac{\partial F(x^{\mathcal{B}})}{\partial x^{\mathcal{B}}_{i,v}} \approx 0 \tag{8}$$

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

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

$$x^{\mathcal{R}'}_{i,c} = 1 \tag{10}$$

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

$$\text{I.C.G.}(x, i) = \int_{\alpha=0}^{1} \frac{\partial F\left(x^{\mathcal{B}}_{i} + \alpha\left(x^{\mathcal{R}'}_{i} - x^{\mathcal{B}}_{i}\right)\right)}{\partial x_{i,c}} \partial\alpha \tag{11}$$

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

4 Results

4.1 Data manifold considerations

[1] Discussion of on and off manifold

5 Applications

5.1 Applications beyond attribution

6 Conclusion

References

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

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

[3] 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.

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

[5] 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.

[6] 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.