Deep Approximate Shapley Propagation
Supplementary Material

A Proof of Proposition 1

Proposition Occlusion, Gradient × Input, Integrated Gradients and DeepLIFT produce exact Shapley values when applied to a linear model and a zero baseline is used.

Proof The proof follows directly from the observation that all the aforementioned methods are equivalent for a linear model [1]. In this case, we can write the model function as $\mathbf{f}(\mathbf{x}) = \sum_i x_i w_i + b$, where w_i is a fixed weight associated with each input x_i, and b is a constant. The marginal effect of each feature i is $(x_i w_i)$ which is the attribution produced by Gradient × Input. As the marginal effect does not depend on the chosen coalition, all elements in the sum of the Shapley values definition are equal $(=(x_i w_i))$, meaning the Shapley value for unit i is also $(x_i w_i)$. Given the equivalence of the methods, they all compute Shapley values.

B Proof of Proposition 2

Proposition Shapley values is the only possible attribution method that satisfies Axioms 1-5.

Proof Integrated Gradients and Shapley values are the only attribution methods that always satisfy Axioms 1-4, as shown in a previous work [12]. We can show that Integrated Gradients does not satisfy Continuity by taking the function $f(x_1, x_2) = min(x_1, x_2)$ and evaluating attributions generated with the two nearly identical inputs $\mathbf{x}_1 = (2, 2 + \epsilon)$ and $\mathbf{x}_1 = (2, 2 - \epsilon)$, using $(0, 0)$ as baseline in both cases. Notice that f is a continuous function and that the output with the two inputs only differs by a factor ϵ. The attributions produced by Integrated Gradients concentrate on the minimum of the two values resulting in $R^{\mathbf{x}_1}_{intgrad} = (2, 0)$, $R^{\mathbf{x}_2}_{intgrad} = (0, 2 - \epsilon)$, thus violating Continuity. Conversely, Shapley values satisfies Continuity because, by definition, the values are a weighted sum of several evaluations of a continuous function.

The counterexample illustrated above highlights the importance of Continuity. By following the gradient, Integrated Gradients propagates the relevance only to the minimum between x_1 and x_2, ignoring the fact that, if the second value were zero, the output would have also been zero. On the other hand, Shapley values distribute the relevance to the two inputs equally by considering all possible coalitions, resulting in $R^{\mathbf{x}_1}_{shapley} = (1, 1) \approx (1 - 0.5\epsilon, 1 - 0.5\epsilon) = R^{\mathbf{x}_2}_{shapley}$

C About the need for a baseline

A feature with an attribution value different than zero is expected to play some role in determining the model outcome. This also implies that *without* such feature the outcome would be different. As pointed out by [12], humans also assign blame to a cause by comparing the outcomes of a process when including that cause, with when not including it. However, this requires the ability to test a process with and without a specific feature, which is problematic with current neural network architectures that do not allow us to explicitly remove them without retraining.

The usual approach to *simulate* the absence of a feature consists of defining a baseline x', for example the black image or the zero input, that will represent the absence of information. On some domains, it is also possible to marginalize over the features to be removed in order to simulate their absence. For example, local coherence of images can be exploited to marginalize over image patches [16]. Unfortunately, this approach is extremely slow. What is more, it can only be applied to images, where contiguous features have a strong correlation, or to other domains where some prior knowledge exists.

In other previous works, the baseline value is sampled from the training set or a prior distribution [11, 2, 8]. This approach can be applied to any dataset but the number of necessary samples (i.e. model evaluations) increases.

Instead, most literature on attribution methods for DNNs suggests the use a fixed baseline value. In this case, zero is the canonical choice [12, 15, 9]. Notice that Gradient $\times$ Input and LRP can also be interpreted as using a zero baseline implicitly. One possible justification relies on the observation that in network that implements a chain of operations of the form $x_j^{(1)} = \sigma(\sum_i (w_{ij} x_i) + b_j)$, the all-zero input is somehow neutral to the output (ie. $\forall c \in C : R_c(\mathbf{0}) \approx 0$). In fact, if all additive biases b_j in the network are zero and we only allow nonlinearities that cross the origin (e.g. ReLU or Tanh), the output for a zero input is exactly zero for all classes. Empirically, the output is often near zero even when biases have different values, which makes the choice of zero for the baseline reasonable, although arbitrary.

D Stochastic input distributions

As a first step to apply DASP, input distributions to the units of the first hidden layers have to be estimated. We assume a univariate Gaussian distribution and estimate mean and variance of these random variables using sample theory. Fig. 1 shows a numerical comparison between empirical and estimated distributions, computed using the MNIST dataset. Even though MNIST input pixels are not normally distributed, the Gaussian approximation for the distribution of a random coalition after the multiplication with the first layer weights seems reasonable, especially for values of k far from the extremes.

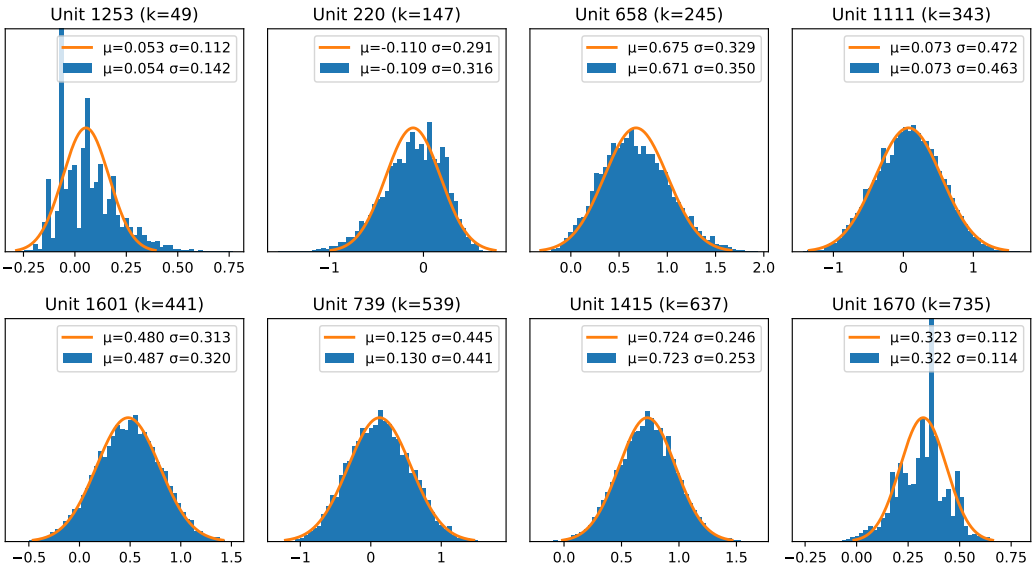

Figure 1: Comparison of numerically estimated distributions (blue) and their approximation using Gaussian distributions (orange) over random hidden units and for different coalition sizes k. We report the mean and standard deviation for both. For each value of k, empirical distributions are computed sampling 10000 random coalitions of the corresponding size from input pixel of a random MNIST image. Moments of the approximate distribution are computed using the method described in Section 4 of the paper.

E Distribution filtering

In this section, we report mean and variance of the filtered distribution of some common DNN operations.

E.1 ReLU activation

The output of a ReLU activation that receives a Gaussian input $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ is a *rectified Gaussian distribution* [10] with mean and variance [3]:

$$\boldsymbol{\mu}_r = \boldsymbol{\mu}\Phi\left(\frac{\boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2}}\right) + \sqrt{\boldsymbol{\sigma}^2}\phi\left(\frac{\boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2}}\right) \tag{1a}$$

$$\boldsymbol{\sigma}^2{}_r = (\boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2)\Phi\left(\frac{\boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2}}\right) + \boldsymbol{\mu}\sqrt{\boldsymbol{\sigma}^2}\phi\left(\frac{\boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2}}\right) - \boldsymbol{\mu}_r^2, \tag{1b}$$

where Φ and ϕ are the cumulative distribution function (CFD) and the PDF of the standard normal distribution, respectively.

E.2 Max Pooling

Max pooling can be seen as returning the maximum response of n random variables $Z_1, ...Z_n$. For two independent inputs $A \sim \mathcal{N}(\mu_A, \sigma_A^2)$, $B \sim \mathcal{N}(\mu_B, \sigma_B^2)$, the maximum is not normally

distributed anymore. Nevertheless, it has been shown that the univariate normal is an effective approximation [4] and the first and second moments can be derived analytically [5]:

$$\mu_{max} = \sqrt{\sigma_A^2 + \sigma_B^2} \cdot \phi(\alpha) + (\mu_A - \mu_B) \cdot \Phi(\alpha) + \mu_B \tag{2}$$

$$\begin{aligned} \sigma_{max}^2 = (\mu_A + \mu_b)\sqrt{\sigma_A^2 + \sigma_B^2} \cdot \phi(\alpha) + (\mu_A^2 + \sigma_A^2) \cdot \Phi(\alpha) \\ + (\mu_B^2 + \sigma_B^2) \cdot (1 - \Phi(\alpha)) - \mu_{max}^2, \end{aligned} \tag{3}$$

where $\alpha = (\mu_A - \mu_B)/\sqrt{\sigma_A^2 + \sigma_B^2}$. When the pooling occurs with more than two inputs, we apply this filtering recursively. The recursion order does not affect the resulting performance significantly [4].

F Experimental setup

We report here the details of the architectures used in our experiments.

As a general remark, we always consider the network output *before* the last layer non-linearity, if any. This is to avoid cross-influence of different output units (in case of a *softmax*) or output shrinking (in case of *sigmoid* or *tanh*). We also use a zero baseline in all experiments and for all attribution methods.

F.1 Parkinsons disability assessment

We trained a multilayer perceptron on the Parkinsons Telemonitoring Data Set [13]. Each input dimension was first normalized in the range $[-1; 1]$. We used Adam [6] and early stopping to train the network, achieving a 4.0 test error (MSE). The following is the architecture.

MNIST CNN
Dense (128)
Activation (ReLU)
Dropout (0.2)
Dense (64)
Activation (ReLU)
Dropout (0.2)
Dense (1)

F.2 Classifying regulatory DNA sequences

The details for this architecture and its training can be found on the original paper [9].

F.3 Digit classification (MNIST)

The MNIST dataset [7] was pre-processed to normalize the input images between -1 (background) and 1 (digit stroke). We trained a convolutional neural network, using Adadelta [14], obtaining a 98.7% test accuracy. The list of layers is listed below.

MNIST CNN
Conv 2D (5x5, 6 kernels)
Activation (ReLU)
Max-pooling (2x2)
Conv 2D (5x5, 32 kernels)
Activation (ReLU)
Max-pooling (2x2)
Dense (120)
Activation (ReLU)
Dense (84)
Activation (ReLU)
Dense (10)
Activation (Softmax)

References

[1] M. Ancona, E. Ceolini, C. Oztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations (ICLR 2018)*, 2018.

[2] A. Datta, S. Sen, and Y. Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *IEEE Symposium on Security and Privacy (SP)*, pages 598–617. IEEE, 2016.

[3] B. J. Frey and G. E. Hinton. Variational learning in nonlinear gaussian belief networks. *Neural Comput.*, 11(1):193–213, Jan. 1999.

[4] J. Gast and S. Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3369–3378, 2018.

[5] E. T. A. F. Jacobs and M. R. C. M. Berkelaar. Gate sizing using a statistical delay model. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000 (Cat. No. PR00537)*, pages 283–290, March 2000.

[6] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998.

[8] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[9] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[10] N. D. Socci, D. D. Lee, and H. S. Seung. The rectified gaussian distribution. In *Advances in neural information processing systems*, pages 350–356, 1998.

[11] E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[12] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[13] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, April 2010.

[14] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[15] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[16] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. 2017.