Explaining Neural Network Decisions Is Hard

Jan Macdonald [1] **Stephan Wäldchen** [1] **Sascha Hauch** [1] **Gitta Kutyniok** [1 2]

Abstract

We connect the widespread idea of interpreting classifier decisions to probabilistic prime implicants. A set of input features is deemed relevant for a classification decision if the classifier score remains nearly constant when randomising the remaining features. This introduces a rate-distortion trade-off between the set size and the deviation of the score. We explain how relevance maps can be interpreted as a greedy strategy to calculate the rate-distortion function. For neural networks we show that approximating this function even in a single point up to any non-trivial approximation factor is NP-hard. Thus, no algorithm will provably find small relevant sets of input features even if they exist. Finally, as a numerical comparison we express a Boolean function, for which the prime implicant sets are known, as a neural network and investigate which relevance mapping methods are able to highlight them.

1. Introduction

Traditional machine learning models such as linear regression, decision trees, or k-nearest neighbours allow for a straight-forward human interpretation of the model prediction. In contrast, the reasoning of highly nonlinear and parameter-rich neural networks remains generally inaccessible. An important first step to solve this problem is deciding which input features are important for a specific classification.

Recent years have seen progress on this front with the introduction of multiple explanation models for deep neural networks (Bach et al., 2015; Lundberg & Lee, 2017; Ribeiro et al., 2016; Shrikumar et al., 2017; Simonyan et al., 2013; Zeiler & Fergus, 2014). These models provide additional information to a prediction in form of a map that assigns importance values to individual input features.

Most commonly, these maps rely on heuristic arguments that motivate the algorithms that produce them. There is yet no formal agreed upon common notion of relevance. There is no specific question that these maps try to answer and they are mostly compared to human intuition about what part of the input variables should be of importance. Notable exceptions are Shapley values (Shapley, 1953) that are required to satisfy certain game theoretic properties.

However, relevance maps have been compared numerically using, e.g., pixel-flipping (Samek et al., 2017) and input perturbation (Fong & Vedaldi, 2017). This points us towards what practitioners understand as relevance and what information they expect relevance maps to provide. The common criterion for relevance that we identified can be summarised by the following question.

> *Q1:* Is there a small part of the input that determines the classification with high probability?

A more quantitative version of the question is the following.

> *Q2:* What is the smallest part of the input that determines the output with high probability?

A reasonable explanation method should be able to answer these questions.

We will see that answering these questions is closely related to a probabilistic version of prime implicants and results in evaluating a rate-distortion function. We demonstrate how relevance maps can be understood as visualisations of greedy approximations of such a function. This allows us to give a direct meaning to the actual values in a relevance map.

Furthermore, we show that answering these questions is generally a hard computational problem:

> Any efficient algorithm cannot reliably answer *Q1* or approximate *Q2* within any non-trivial approximation factor[1].

[1]unless $\mathsf{P} = \mathsf{NP}$

[1]Institut für Mathematik, Technische Universität Berlin, Berlin, Germany [2]Department of Physics and Technology, University of Tromsø, Tromsø, Norway. Correspondence to: Jan Macdonald <macdonald@math.tu-berlin.de>, Stephan Wäldchen <stephanw@math.tu-berlin.de>.

Presented at the XXAI Workshop, 37^{th} *International Conference on Machine Learning (ICML)*, 2020.

We see this result as an indication that explanation algorithms will have to continue to rely on heuristic motivation and thorough numerical comparison.

Notation Throughout, $d \in \mathbb{N}$ is the dimension of the signal domain. We set $[d] = \{1, \dots, d\}$ and for a binary signal $\mathbf{x} \in \{0,1\}^d$ or continuous signal $\mathbf{x} \in [0,1]^d$ and a subset $S \subseteq [d]$ we denote by $\mathbf{x}_S$ the restriction of $\mathbf{x}$ to components indexed by S. The uniform distributions on $\{0,1\}^d$ and $[0,1]^d$ are $\mathcal{U}(\{0,1\}^d)$ and $\mathcal{U}([0,1]^d)$ respectively and $\mathbf{1}_d \in \mathbb{R}^d$ is a vector of ones.

2. From Prime Implicants to Rate-Distortion

Prime implicants are a concept from Boolean logic that has been extended for abductive reasoning in first order logics (Marquis, 1991; 2000) and explanation of classifier decisions (Shih et al., 2018). In a nutshell, an implicant explanation is a subset of the input variables that is sufficient for the decision. In other words, keeping the implicant variables fixed will lead to the same classification for all possible completions of the remaining variables. A prime implicant explanation is a minimal implicant with respect to set inclusion and thus can not be reduced further.

However, the deterministic requirement to produce the same classification for *all* completions is often to strict, especially for high-dimensional problems as commonly found in modern machine learning. Let us illustrate this with the task of image classification as an example. Here, often small regions of the input image can be manipulated in a way that changes a classifier prediction, e.g. through adversarial patches (Brown et al., 2017; Liu et al., 2018). Thus, prime implicant explanations will have to cover large portions of the input image in order to exclude all adversarial patches, independent of the size of the actual object in the image that led to the original classifier prediction. This is often undesirable and not very useful to uncover the underlying reasoning of the classifier.

Therefore, a relaxation of this notion, called δ-relevance, was recently introduced for Boolean classifiers. It can be seen as a probabilistic version of prime implicant explanations, which only requires that the classifier prediction remains unchanged with high probability.

Definition 2.1 (Wäldchen et al., 2019)**.** Let $\delta \in [0,1]$, $\Psi\colon \{0,1\}^d \to \{0,1\}$, and $\mathbf{x} \in \{0,1\}^d$. A set $S \subseteq [d]$ is called *δ-relevant* for Ψ and $\mathbf{x}$, if

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S] \geq \delta.$$

For the special case $\delta = 1$ this is the same as prime implicant explanations. Deciding whether there exists a small δ-relevant set was shown to be $\mathsf{NP}^{\mathsf{PP}}$-hard for $\delta \in (0,1)$ by Wäldchen et al. (2019). For the binary case, this exactly amounts to answering *Q1*, while finding the smallest set S that achieves relevance of a given δ corresponds to answering *Q2*.

The formulation of δ-relevance introduces a trade-off between the probability threshold δ and the minimal set size $|S|$ that can achieve it.

Here we want to take this idea further and extend it from the binary to the continuous setting. Given S and $\mathbf{x}$ we write $\mathbf{y} \sim \mathcal{U}_S$ when $\mathbf{y}_S = \mathbf{x}_S$ and $\mathbf{y}_{S^c} \sim \mathcal{U}\big(\{0,1\}^{d-|S|}\big)$. Then, we can rewrite the δ-relevance condition as

$$\begin{aligned} & \mathbb{P}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S] \geq \delta \\ \iff \quad & \mathbb{E}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}[|\Psi(\mathbf{y}) - \Psi(\mathbf{x})| \,|\, \mathbf{y}_S = \mathbf{x}_S] \leq 1 - \delta \\ \iff \quad & \mathbb{E}_{\mathbf{y}\sim\mathcal{U}_S}[|\Psi(\mathbf{y}) - \Psi(\mathbf{x})|] \leq 1 - \delta. \end{aligned}$$

The right hand side $1-\delta$ can be seen as a distortion measure bounding the expected change in the classifier prediction. This formulation through an expectation is well suited to be generalised to a continuous setting. Also, other probability distributions as well as other distance measures than the absolute difference might be of interest.

Let now $\Phi\colon [0,1]^d \to [0,1]$ be a classifier function, $\mathcal{V}$ be a probability distribution on $[0,1]^d$, and $\mathbf{n} \sim \mathcal{V}$ a random vector. We define the *obfuscation* of a signal $\mathbf{x} \in [0,1]^d$ with respect to $S \subseteq [d]$ and $\mathcal{V}$ as a random vector $\mathbf{y}$ that is deterministically defined on S as $\mathbf{y}_S = \mathbf{x}_S$ and distributed on the complement according to $\mathbf{y}_{S^c} = \mathbf{n}_{S^c}$. As above, we write $\mathcal{V}_S$ for the resulting distribution of $\mathbf{y}$. This enables us to define the distortion of S with respect to $\Phi, \mathbf{x}$ and $\mathcal{V}$ as

$$D(S, \Phi, \mathbf{x}, \mathcal{V}) = \mathbb{E}_{\mathbf{y}\sim\mathcal{V}_S}\left[\mathrm{dist}(\Phi(\mathbf{x}), \Phi(\mathbf{y}))\right],$$

where $\mathrm{dist}(\cdot)$ is a distance measure, e.g. absolute difference or squared difference. We will use the abbreviated notation $D(S)$, whenever Φ, $\mathbf{x}$, and $\mathcal{V}$ are clear from context.

The relationship between set size and distortion is described by the rate-distortion function, defined as

$$R(\epsilon, \Phi, \mathbf{x}, \mathcal{V}) = \min\{\, |S| \,:\, S \subseteq [d],\, D(S, \Phi, \mathbf{x}, \mathcal{V}) \leq \epsilon \,\}. \tag{1}$$

The distortion limit ϵ takes the role of $1-\delta$ from before. Again, we use the abbreviation $R(\epsilon)$ if the context is clear.

The idea of formulating relevance from a rate-distortion viewpoint can be derived from the hypothetical setup illustrated in Figure 1. The terminology is borrowed from information theory where rate-distortion is used to analyse lossy data compression. In that sense, the set of relevant components can be thought of as a compressed description of the signal with the expected deviation from the classification score being a measure for the reconstruction error.

This framework is used to state a clearly defined objective that relevance maps should fulfil: Given a distortion limit ϵ,

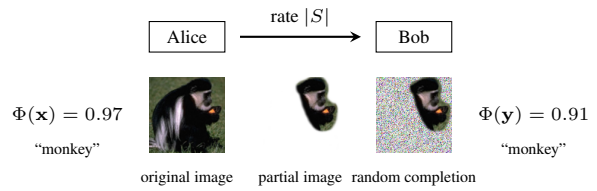

Figure 1. We motivate the rate-distortion viewpoint from the following hypothetical scenario: two people, Alice and Bob, have access to the same neural network classifier. Alice classified an image as a "monkey" and wants to convey this to Bob. She is only allowed to send a limited number of pixels to Bob, who will complete the image with random values. Alice's best chance of convincing Bob is to transmit those pixels that are most relevant for the class "monkey" and ensure a small difference between their classification scores in expectation.

the goal is to find a set S achieving the minimum in (1). This amounts to solving a continuous generalisation of finding small δ-relevant sets. Thus, evaluating the rate-distortion function answers questions *Q1* and *Q2*. We will show that no efficient algorithm can always fulfil this objective. Still, it can be used to numerically evaluate the quality of relevance maps that were produced by heuristic algorithms, as discussed in the next section.

3. Relevance Maps and Orderings

Most established explanation methods calculate continuous relevance scores for all input components instead of a strict partition into relevant and non-relevant components. It seems not immediately clear how the two concepts relate. However, we argue that the exact numerical values of a relevance map are generally meaningless. Instead, it is the *ordering* of the input components according to their relevance scores that is of importance. Let $\pi\colon [d] \to [d]$ be a permutation that describes a relevance ordering in the sense that $\pi(k)$ is the k-th most relevant input component and $\pi([k])$ are the k most relevant input components. This ordering can be seen as a greedy approach to solve one of the following two questions[2] for varying ϵ.

Productive Formulation If we want to *preserve* the class prediction $\Phi(\mathbf{x})$ up to a maximal distortion of $D(S) \leq \epsilon$, which is the smallest set S of components we should fix?

The opposing formulation might be equally valid depending on the application.

Destructive Formulation If we want to *destroy* the class prediction $\Phi(\mathbf{x})$ with minimal distortion $D(S^c) \geq \epsilon$, which is the smallest set S of components we should obfuscate?

[2]Fong & Vedaldi (2017) refer to these two formulations as a preservation and deletion game respectively.

Though seemingly equivalent, these questions do generally not have the same answer.[3]

We argue that all existing quantitative evaluation methods for relevance maps implicitly use one of these formulations: they are based on obfuscating or perturbing parts of the input components that are deemed most or least relevant and measure the change in the classification score. Zeiler & Fergus (2014) consider obfuscations by a constant baseline value, Samek et al. (2017) use obfuscations by random values, and Fong & Vedaldi (2017) use both types of obfuscations as well as perturbations by blurring.

We focus on the productive formulation, where the size of the optimal solution is described by our rate-distortion function $R(\epsilon)$. A good relevance ordering is one that provides good approximations to the optimal size, when we greedily include input components in descending order of their relevance until the distortion limit is satisfied. The rate function associated to an ordering π is thus given by

$$R_\pi(\epsilon) = \min\{\, k \in [d] :\ D(\pi([k]) \leq \epsilon \,\}.$$

Clearly $R(\epsilon) \leq R_\pi(\epsilon)$ holds for any ordering π. But we can evaluate relevance maps by how well the rate function associated to the induced relevance ordering approximates the optimal rate $R(\epsilon)$. It would be desirable to obtain meaningful upper bounds on the approximation error. Unfortunately, we will see that no non-trivial approximation bound can be given for any efficient method of calculating relevance maps. They cannot be proven to perform systematically better than a random ordering and do not provably find small relevant sets, even when they exist. Nevertheless, the ordering based rate functions R_π can still be used for comparing different relevance maps to each other. This results in a comparison test very similar to the test in (Samek et al., 2017).

4. Computational Complexity

The inapproximability of small relevant sets has been shown for binary functions, represented as Boolean circuits, and the uniform distribution $\mathcal{U}\big(\{0,1\}^d\big)$ on $\{0,1\}^d$ by Wäldchen et al. (2019). We generalise this result to continuous functions, represented by neural networks, and the uniform distribution $\mathcal{U}\big([0,1]^d\big)$ on $[0,1]^d$. In the following we will consider the distortion with respect to the squared difference distance, more precisely

$$\mathrm{dist}(\Phi(\mathbf{x}), \Phi(\mathbf{y})) = \frac{1}{2}(\Phi(\mathbf{x}) - \Phi(\mathbf{y}))^2.$$

However, our results can easily be generalised to other distance functions.

[3]Consider the case of redundancy, e.g. a picture with two monkeys that was classified as containing a monkey. In the productive scenario it can be sufficient to include just one of the monkeys in S, while in the destructive scenario one should try to obfuscate both monkeys equally.

4.1. Neural Network Functions

Let $L \in \mathbb{N}$ denote the number of layers of a neural network, $d_1, \dots, d_{L-1} \in \mathbb{N}$ and $d_0 = d$, $d_L = 1$. Further let $(\mathbf{W}_1, \mathbf{b}_1), \dots, (\mathbf{W}_L, \mathbf{b}_L)$ with $\mathbf{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$, $\mathbf{b}_i \in \mathbb{R}^{d_i}$ for $i \in [L]$ be weight matrices and bias vectors. From now on we consider functions of the form

$$\Phi(\mathbf{x}) = \mathbf{W}_L \varrho(\dots \varrho(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \dots) + \mathbf{b}_L,$$

with the rectified linear unit (ReLU) activation function $\varrho(x) = \max\{0, x\}$. A neural network $\Phi\colon [0,1]^d \to [0,1]$ is said to interpolate a binary classifier $\Psi\colon \{0,1\}^d \to \{0,1\}$ if Φ restricted to $\{0,1\}^d$ is equal to Ψ. We will make use of the fact that ReLU neural networks can interpolate arbitrary Boolean circuits with comparable depth and width (Mukherjee & Basu, 2017; Parberry, 1996).

4.2. Approximating the Rate-Distortion Function

For a fixed distribution $\mathcal{V}$ on $[0,1]^d$ we say that an algorithm to calculate the rate-distortion function achieves the approximation factor $c \geq 1$ if for any signal $\mathbf{x} \in [0,1]^d$, neural network $\Phi\colon [0,1]^d \to [0,1]$, and distortion limit $\epsilon \in (0,1]$ it computes a set S of size

$$R(\epsilon, \Phi, \mathbf{x}, \mathcal{V}) \leq |S| \leq cR(\epsilon, \Phi, \mathbf{x}, \mathcal{V}),$$

satisfying $D(S, \Phi, \mathbf{x}, \mathcal{V}) \leq \epsilon$. In other words S can be larger than the optimal size $R(\epsilon, \Phi, \mathbf{x}, \mathcal{V})$ by at most a factor c. The approximation factor d can trivially be achieved by simply taking $S = [d]$, i.e., taking all input components as relevant. We will show that anything beyond this is computationally hard. There is no efficient algorithm that can do significantly better than the trivial factor d.

Theorem 4.1. *Let $\mathcal{V} = \mathcal{U}\big([0,1]^d\big)$ and assume $\mathsf{P} \neq \mathsf{NP}$. Then for any $\alpha \in (0,1]$ there does not exist a polynomial time approximation algorithm for $R(\epsilon, \Phi, \mathbf{x}, \mathcal{V})$ with an approximation factor of $d^{1-\alpha}$.*

We will prove this by reducing an NP-hard problem from the binary setting considered in (Wäldchen et al., 2019) to the problem of evaluating the rate-distortion function. Let us quickly recall some notions from the binary case.

Definition 4.2 (Wäldchen et al., 2019)**.** For $\delta \in (0,1]$ and $\gamma \in [0, \delta)$ the MIN-GAPPED-RELEVANT-INPUT problem is defined as follows.

GIVEN: A Boolean circuit $\Psi\colon \{0,1\}^d \to \{0,1\}$ and a variable assignment $\mathbf{x} \in \{0,1\}^d$.

FIND: $k \in \mathbb{N}$, $1 \leq k \leq d$ such that

1. there exists a set $S \subseteq [d]$ with $|S| = k$ and S is $(\delta - \gamma)$-relevant for Ψ and $\mathbf{x}$,
2. all sets $S \subseteq [d]$ with $|S| < k$ are not δ-relevant for Ψ and $\mathbf{x}$.

An algorithm for MIN-GAPPED-RELEVANT-INPUT is said to have an approximation factor $c \geq 1$ if, for any instance $\{\Psi, \mathbf{x}\}$, it produces an approximate solution k such that there exists a true solution $\tilde{k}$ (satisfying both conditions in Definition 4.2) with $\tilde{k} \leq k \leq c\tilde{k}$. Wäldchen et al. (2019) showed that for any $\alpha > 0$ no polynomial time approximation algorithm for MIN-GAPPED-RELEVANT-INPUT with approximation factor $d^{1-\alpha}$ exists, unless $\mathsf{P} = \mathsf{NP}$.

The idea for the proof of Theorem 4.1 can be summarised in a few steps: Given a Boolean circuit Ψ we choose an interpolating ReLU network Φ_0. For any $\eta > 0$ there exists a fixed size ReLU network Φ_η that transforms the uniform distribution $\mathcal{U}\big([0,1]^d\big)$ into the binary distribution $\mathcal{U}\big(\{0,1\}^d\big)$ up to a small error depending explicitly on η, such that $\Phi = \Phi_0 \circ \Phi_\eta$ still interpolates Ψ. The difference of the distortions $D\big(S, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big)$ and $D\big(S, \Psi, \mathbf{x}, \mathcal{U}\big(\{0,1\}^d\big)\big)$ can be shown to depend explicit on η as well. Moreover, the binary distortion is directly related to the probability lower bounded by δ in Definition 2.1. Thus, it can be shown that for the right choice of η any approximation algorithm for the rate-distortion function would also be an approximation algorithm for the MIN-GAPPED-RELEVANT-INPUT problem with the same approximation factor. The inapproximability result in (Wäldchen et al., 2019) thus carries over to the continuous setting.

For brevity, we introduce the notation

$$D^b_{\Phi,\mathbf{x}}(S) = D\big(S, \Phi, \mathbf{x}, \mathcal{U}\big(\{0,1\}^d\big)\big),$$
$$D^c_{\Phi,\mathbf{x}}(S) = D\big(S, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big),$$

for the binary and the continuous distortion respectively. For $0 < \eta \leq 1$ we set $\Phi_\eta(\mathbf{x}) = \varphi\Big(\frac{1}{\eta}\big(\mathbf{x} - \frac{1-\eta}{2}\mathbf{1}_d\big)\Big)$ with

$$\varphi(x) = \begin{cases} 0, & x \leq 0, \\ x, & 0 < x \leq 1, \\ 1, & x > 1, \end{cases}$$

and observe that Φ_η interpolates the identity on $\{0,1\}^d$ and can be realised by two ReLU layers of size $\mathcal{O}(d)$.

Before we come to the main proof we state two more useful results.

Lemma 4.3. *Let $\Psi\colon \{0,1\}^d \to \{0,1\}$ and $\mathbf{x} \in \{0,1\}^d$. Then for any $S \subseteq [d]$ we have*

$$\mathbb{P}_{\mathbf{y} \sim \mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S] = 1 - 2D^b_{\Psi,\mathbf{x}}(S).$$

Lemma 4.4. *Let $\Psi\colon \{0,1\}^d \to \{0,1\}$ and $\mathbf{x} \in \{0,1\}^d$. Then for any $\Phi_0\colon [0,1]^d \to [0,1]$ interpolating Ψ, $S \subseteq [d]$, and $0 < \eta \leq 1$ we have for $\Phi = \Phi_0 \circ \Phi_\eta$ that*

$$D^b_{\Phi,\mathbf{x}}(S) = D^b_{\Phi_0,\mathbf{x}}(S) = D^b_{\Psi,\mathbf{x}}(S)$$

as well as

$$\big|D^c_{\Phi,\mathbf{x}}(S) - D^b_{\Psi,\mathbf{x}}(S)\big| \leq \frac{d\eta}{2}.$$

Both Lemmas 4.3 and 4.4 follow from straight forward calculations. The full proofs can be found in the supplementary material. We now come to the proof of the main theorem.

Proof of Theorem 4.1. Let $\delta \in (0,1]$, $\gamma \in [0,\delta)$, and $\{\Psi, \mathbf{x}\}$ be an instance of MIN-GAPPED-RELEVANCE-INPUT. Let $\Phi_0 \colon [0,1]^d \to [0,1]$ be a neural network that interpolates Ψ, set $\eta = \frac{\gamma}{2d}$, $\Phi = \Phi_0 \circ \Phi_\eta$, and $\epsilon = \frac{1}{2}\big(1 - \delta + \frac{\gamma}{2}\big)$. We show that $R\big(\epsilon, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big)$ is a solution for the MIN-GAPPED-RELEVANCE-INPUT problem instance $\{\Psi, \mathbf{x}\}$, i.e., it fulfils both conditions in Definition 4.2. To see this, let

$$S^* = \operatorname{argmin}\big\{|S| : S \subseteq [d], D\big(S, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big) \leq \epsilon\big\},$$

and hence $|S^*| = R\big(\epsilon, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big)$.

Lemma 4.4 yields

$$\begin{aligned} 1 - 2D^b_{\Psi,\mathbf{x}}(S^*) &\geq 1 - 2\left(D^c_{\Phi,\mathbf{x}}(S^*) + \frac{d\eta}{2}\right) \\ &\geq 1 - 2\epsilon - \frac{\gamma}{2} = \delta - \gamma, \end{aligned}$$

and together with Lemma 4.3 we get

$$\mathbb{P}_{\mathbf{y} \sim \mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_{S^*} = \mathbf{x}_{S^*}] \geq \delta - \gamma,$$

showing that the first condition in Definition 4.2 is satisfied.

Similarly, for any S with $|S| < R\big(\epsilon, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big)$ we know $D\big(S, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big) > \epsilon$. Thus by Lemma 4.4

$$\begin{aligned} 1 - 2D^b_{\Psi,\mathbf{x}}(S) &\leq 1 - 2\left(D^c_{\Phi,\mathbf{x}}(S) - \frac{d\eta}{2}\right) \\ &< 1 - 2\epsilon + \frac{\gamma}{2} = \delta, \end{aligned}$$

and again using Lemma 4.3 we obtain

$$\mathbb{P}_{\mathbf{y} \sim \mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S] < \delta,$$

showing that the second condition in Definition 4.2 is satisfied as well.

Hence, any algorithm approximating the rate-distortion function $R\big(\epsilon, \Phi, \mathbf{x}, \mathcal{U}\big([0,1]^d\big)\big)$ can also be used as an approximation algorithm for MIN-GAPPED-RELEVANT-INPUT with the same approximation factor. For the latter it is known that achieving the factor $d^{1-\alpha}$ is NP-hard for any $\alpha > 0$, which completes the proof. □

This is a worst-case analysis that does not imply that the task is infeasible in practical applications. Many nonlinear optimisation problems are NP-hard in general and yet performed successfully on a regular basis. But performance guarantees cannot be proven as long as the neural networks considered are powerful enough to represent arbitrary logical functions, which is the case for ReLU networks.

This still leaves the option of more subtle restrictions on the neural networks and the inputs that depend on the actual data structures on which the networks have been trained. These, however, are not yet well enough understood. As long as this is the case we have to rely on heuristic solution strategies. We will briefly discuss a possible approach in the next section.

5. Continuous Rate-Distortion Explanations

Finding the optimal partition into S and S^c for varying distortion limits $D(S) \leq \epsilon$ is a hard combinatorial optimisation problem. As discussed before, the component orderings obtained from continuous relevance scores can serve as greedy approximations. Macdonald et al. (2019) introduced a heuristic algorithm to obtain such scores for ReLU neural networks, which is based on a continuous relaxation of the problem of finding small δ-relevant sets. Instead of binary relevance decisions (*relevant* versus *non-relevant*) encoded by the set S, it determines a continuous relevance score for each component, encoded by a vector $\mathbf{s} \in [0,1]^d$. This leads to a box-constrained optimisation problem

$$\text{minimise} \quad D(\mathbf{s}) + \lambda \|\mathbf{s}\|_1 \quad \text{subject to} \quad \mathbf{s} \in [0,1]^d \tag{2}$$

with a regularisation parameter $\lambda > 0$ penalising the "size" of $\mathbf{s}$ (in correspondence to the set size $|S|$) and balancing it against the distortion $D(\mathbf{s})$. The evaluation of the expected value in the distortion functional can be achieved through assumed density filtering (ADF), which has recently also been used for ReLU neural networks in the context of uncertainty quantification (Gast & Roth, 2018). The optimisation problem (2) is then solved via (projected) gradient descent or L-BFGS-B (Byrd et al., 1995). This approach to obtaining relevance scores for classifier decisions is called RDE (Rate-Distortion Explanation), and we refer to (Macdonald et al., 2019) for details.

6. Comparison Methods

RDE is strongly motivated by the formulation of δ-relevance and clearly aims at answering the questions *Q1* and *Q2*. But it remains, like all other efficient relevance mapping methods, a heuristic that can not provably achieve this goal in all situations. Thus, a post-hoc evaluation of relevance mappings as well as a comparison of different methods is recommended.

We advocate for a quantitative analysis complementing the visual evaluation of relevance maps, as also done in (Samek et al., 2017; Fong & Vedaldi, 2017). Relevance maps coincide with human intuition only if the relevance algorithm performs correctly and the network has learned precisely

the reasoning a human would use, which is unclear in many circumstances. In fact, the relevance method should be evaluated on quantitative terms and then be used to access the reasoning of neural networks.

In addition to the quantitative evaluation and comparison tests in (Samek et al., 2017; Fong & Vedaldi, 2017), we propose to use designed classifiers and synthetic data as baseline tests. Here, as a proof of concept, we evaluate the performance of several methods for a classification task on synthetic binary string data, where the optimal relevant sets are known.

We compare RDE to SmoothGrad[4] (Smilkov et al., 2017), SHAP[5] (Lundberg & Lee, 2017), and LIME[6] (Ribeiro et al., 2016).

6.1. Synthetic Binary Strings

As a baseline, we propose to test relevance mapping methods on a synthetic binary classification task. We consider the Boolean function

$$\Psi\colon \{0,1\}^d \to \{0,1\}, \quad \mathbf{x} \mapsto \bigvee_{i=1}^{d-k+1} \bigwedge_{j=i}^{i+k-1} x_j,$$

that checks binary strings of length d for the existence of a block of k consecutive ones.

If an input signal $\mathbf{x}$ contains a unique set of k consecutive ones, then it is clear that these variables are relevant for the classification. More precisely, the smallest rate that can achieve distortion zero is k and in fact any set S containing the block of k consecutive ones will achieve it. On the other hand any smaller set of size $|S| < k$ will have distortion at least $\frac{1}{2}$.

We construct a ReLU neural network $\Phi\colon [0,1]^d \to [0,1]$ that interpolates Ψ, see the supplement for details. Relying on the connection between the binary and continuous setting established in Section 4 we expect that a relevance mapping method should also find the block of k consecutive ones as most relevant for Φ.

We test this for two input signals of size $d = 16$ each containing a block of $k = 5$ consecutive ones. The first has no disjoint other group of five consecutive variables that is even close to being a block of ones, see Figure 2. The second also has a disjoint second group of five consecutive variables that almost forms a block of ones (four of the five are ones), see Figure 3.

RDE, SHAP, and SmoothGrad identify the correct block as relevant in both cases, whereas LIME identifies the correct block in the first case but gets distracted by the incomplete block in the second case, see Figures 2 and 3. Experiments comparing these (and many more) methods on image classification tasks can be found in (Macdonald et al., 2019).

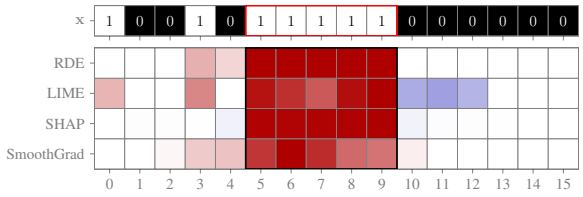

Figure 2. Relevance mappings generated by several methods for a binary string containing a block of five consecutive ones. The colourmap indicates positive relevances as red and negative relevances as blue. All methods clearly identify the correct block as relevant.

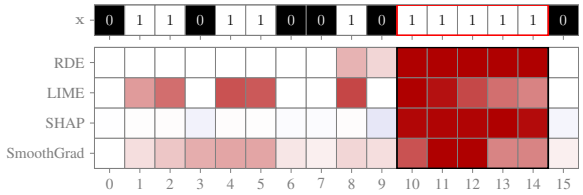

Figure 3. Relevance mappings generated by several methods for a binary string containing a complete and an incomplete block of five consecutive ones. The colourmap indicates positive relevances as red and negative relevances as blue. RDE, SHAP and SmoothGrad identify the correct block as most relevant. For SmoothGrad the distinction from the incomplete block is less pronounced.

7. Conclusion

We extended the concept of δ-relevance, a probabilistic version of prime implicants, from a binary to a continuous setting. The resulting rate-distortion framework allows us to formulate a concrete objective that relevance maps should solve and to analyse the complexity of this problem. We proved that in the worst case it is hard to solve and even hard to approximate, which justifies the use of heuristic explanation methods in practical applications.

Acknowledgements

J. M. and S. W. acknowledge support by DFG-GRK-2260 (BIOQIC). S. H. is grateful for support by CRC/TR 109 “Discretization in Geometry and Dynamics”. G. K. acknowledges partial support by the Bundesministerium für Bildung und Forschung through the “Berliner Zentrum für Machinelles Lernen”, by the Deutsche Forschungsgemeinschaft through Grants CRC 1114 “Scaling Cascades in Complex Systems”, CRC/TR 109 “Discretization in Geometry and Dynamics”, DFG-GRK-2433 (DAEDALUS), DFG-GRK-

[4] https://github.com/albermax/innvestigate
[5] https://github.com/slundberg/shap
[6] https://github.com/marcotcr/lime

2260 (BIOQIC), SPP 1798 “Compressed Sensing in Information Processing” (CoSIP), by the Berlin Mathematics Research Centre MATH+, and the Einstein Foundation Berlin.

References

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140.

Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. Adversarial patch. *CoRR*, abs/1712.09665, 2017. URL http://arxiv.org/abs/1712.09665.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. doi: 10.1137/0916069.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437, 2017.

Gast, J. and Roth, S. Lightweight probabilistic deep networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3369–3378, June 2018. doi: 10.1109/CVPR.2018.00355.

Liu, X., Yang, H., Liu, Z., Song, L., Li, H., and Chen, Y. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

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

Macdonald, J., Wäldchen, S., Hauch, S., and Kutyniok, G. A rate-distortion framework for explaining neural network decisions. *arXiv e-prints*, art. arXiv:1905.11092, May 2019.

Marquis, P. Extending abduction from propositional to first-order logic. In *International Workshop on Fundamentals of Artificial Intelligence Research*, pp. 141–155. Springer, 1991.

Marquis, P. Consequence finding algorithms. In Kohlas, J. and Moral, S. (eds.), *Handbook of Defeasible Reasoning and Uncertainty Management Systems: Algorithms for Uncertainty and Defeasible Reasoning*, pp. 41–145. Springer Netherlands, Dordrecht, 2000. ISBN 978-94-017-1737-3. doi: 10.1007/978-94-017-1737-3_3.

Mukherjee, A. and Basu, A. Lower bounds over boolean inputs for deep neural networks with relu gates. *arXiv preprint arXiv:1711.03073*, 2017.

Parberry, I. *Circuit complexity and feedforward neural networks*. Hillsdale, NJ: Lawrence Erlbaum, 1996.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 11 2017. ISSN 2162-237X. doi: 10.1109/TNNLS.2016.2599820.

Shapley, L. S. A value for n-person games. In Kuhn, H. W. and Tucker, A. W. (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Shih, A., Choi, A., and Darwiche, A. A symbolic approach to explaining bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 5103–5111. AAAI Press, 2018. ISBN 9780999241127.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. *CoRR*, abs/1704.02685, 2017. URL http://arxiv.org/abs/1704.02685.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Smilkov, D., Thorat, N., Kim, B., Viégas, F. B., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL http://arxiv.org/abs/1706.03825.

Wäldchen, S., Macdonald, J., Hauch, S., and Kutyniok, G. The computational complexity of understanding network decisions. *arXiv e-prints*, art. arXiv:1905.09163, May 2019.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. (eds.), *Computer Vision – ECCV 2014*, pp. 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

Supplementary Material for "Explaining Neural Network Decisions Is Hard"

Jan Macdonald [1] **Stephan Wäldchen** [1] **Sascha Hauch** [1] **Gitta Kutyniok** [1 2]

A. Proof of Lemma 4.3

Lemma. *Let* $\Psi\colon \{0,1\}^d \to \{0,1\}$ *and* $\mathbf{x} \in \{0,1\}^d$. *Then for any* $S \subseteq [d]$ *we have*

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S] = 1 - 2D^b_{\Psi,\mathbf{x}}(S).$$

Proof. The claim follows directly from the definition of $D^b_{\Psi,\mathbf{x}}$. We have

$$\begin{aligned} D^b_{\Psi,\mathbf{x}}(S) &= \frac{1}{2}\mathbb{E}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}\big[(\Psi(\mathbf{y}) - \Psi(\mathbf{x}))^2 \,\big|\, \mathbf{y}_S = \mathbf{x}_S\big] \\ &= \frac{1}{2}\mathbb{P}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) \neq \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S] \\ &= \frac{1}{2}\big(1 - \mathbb{P}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S]\big) \end{aligned}$$

and thus

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}[\Psi(\mathbf{y}) = \Psi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S] = 1 - 2D^b_{\Psi,\mathbf{x}}(S). \qquad \square$$

B. Proof of Lemma 4.4

Lemma. *Let* $\Psi\colon \{0,1\}^d \to \{0,1\}$ *and* $\mathbf{x} \in \{0,1\}^d$. *Then for any* $\Phi_0\colon [0,1]^d \to [0,1]$ *interpolating* Ψ, $S \subseteq [d]$, *and* $0 < \eta \leq 1$ *we have for* $\Phi = \Phi_0 \circ \Phi_\eta$ *that*

$$D^b_{\Phi,\mathbf{x}}(S) = D^b_{\Phi_0,\mathbf{x}}(S) = D^b_{\Psi,\mathbf{x}}(S)$$

as well as

$$\big|D^c_{\Phi,\mathbf{x}}(S) - D^b_{\Psi,\mathbf{x}}(S)\big| \leq \frac{d\eta}{2}.$$

Proof. The first part of the claim follows directly from the fact that both Φ_0 and $\Phi = \Phi_0 \circ \Phi_\eta$ interpolate Ψ.

For the second part we consider the event

$$C_S = \left\{ \mathbf{y} \in [0,1]^d \,:\, \exists i \in S^c \text{ such that } y_i \in \left(\frac{1-\eta}{2}, \frac{1+\eta}{2}\right) \right\},$$

and observe that

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{U}([0,1]^d)}[\mathbf{y} \in C_S] = 1 - (1-\eta)^{d-|S|}.$$

[1]Institut für Mathematik, Technische Universität Berlin, Berlin, Germany [2]Department of Physics and Technology, University of Tromsø, Tromsø, Norway. Correspondence to: Jan Macdonald <macdonald@math.tu-berlin.de>, Stephan Wäldchen <stephanw@math.tu-berlin.de>.

Presented at the XXAI Workshop, 37^{th} *International Conference on Machine Learning (ICML)*, 2020.

Using the abbreviated notation

$$A_S = \mathbb{E}_{\mathbf{y}\sim\mathcal{U}([0,1]^d)}\big[(\Phi(\mathbf{y}) - \Phi(\mathbf{x}))^2 \,\big|\, \mathbf{y}_S = \mathbf{x}_S, \mathbf{y} \in C_S\big]$$
$$B_S = \mathbb{E}_{\mathbf{y}\sim\mathcal{U}([0,1]^d)}\big[(\Phi(\mathbf{y}) - \Phi(\mathbf{x}))^2 \,\big|\, \mathbf{y}_S = \mathbf{x}_S, \mathbf{y} \notin C_S\big]$$

we can split the expectation value in the continuous distortion term as

$$\begin{aligned} D^c_{\Phi,\mathbf{x}}(S) &= \frac{1}{2}\mathbb{E}_{\mathbf{y}\sim\mathcal{U}([0,1]^d)}\big[(\Phi(\mathbf{y}) - \Phi(\mathbf{x}))^2 \,\big|\, \mathbf{y}_S = \mathbf{x}_S\big] \\ &= \frac{1}{2}A_S\mathbb{P}_{\mathbf{y}\sim\mathcal{U}([0,1]^d)}[\mathbf{y} \in C_S] + \frac{1}{2}B_S\mathbb{P}_{\mathbf{y}\sim\mathcal{U}([0,1]^d)}[\mathbf{y} \notin C_S] \\ &= \frac{1}{2}A_S\Big(1 - (1-\eta)^{d-|S|}\Big) + \frac{1}{2}B_S(1-\eta)^{d-|S|}. \end{aligned}$$

For the second term, we get

$$\begin{aligned} B_S &= \mathbb{E}_{\mathbf{y}\sim\mathcal{U}([0,1]^d)}\big[(\Phi_0 \circ \Phi_\eta(\mathbf{y}) - \Phi_0 \circ \Phi_\eta(\mathbf{x}))^2 \,\big|\, \mathbf{y}_S = \mathbf{x}_S, \mathbf{y} \notin C_S\big] \\ &= \mathbb{E}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}\big[(\Phi_0(\mathbf{y}) - \Phi_0(\mathbf{x}))^2 \,\big|\, \mathbf{y}_S = \mathbf{x}_S\big] \\ &= \mathbb{E}_{\mathbf{y}\sim\mathcal{U}(\{0,1\}^d)}\big[(\Psi(\mathbf{y}) - \Psi(\mathbf{x}))^2 \,\big|\, \mathbf{y}_S = \mathbf{x}_S\big] \\ &= 2D^b_{\Psi,\mathbf{x}}(S), \end{aligned}$$

where the second equality follows from the choice of Φ_η and $\mathbf{y} \notin C_S$ and the third equality follows from the fact that Φ_0 interpolates Ψ. Thus, we conclude

$$\begin{aligned} D^c_{\Phi,\mathbf{x}}(S) &= \frac{1}{2}A_S\Big(1 - (1-\eta)^{d-|S|}\Big) + D^b_{\Psi,\mathbf{x}}(S)(1-\eta)^{d-|S|} \\ &= D^b_{\Psi,\mathbf{x}}(S) + \frac{1}{2}\Big(1 - (1-\eta)^{d-|S|}\Big)(A_S - B_S) \end{aligned}$$

which using Bernoulli's inequality and $0 \le A_S, B_S \le 1$ finally results in

$$\big|D^c_{\Phi,\mathbf{x}}(S) - D^b_{\Psi,\mathbf{x}}(S)\big| \le \frac{1}{2}\Big(1 - (1-\eta)^{d-|S|}\Big) \le \frac{(d-|S|)\eta}{2} \le \frac{d\eta}{2}. \qquad \square$$

C. Choice of the Reference Distribution

C.1. Non-Uniform Distributions

We use the uniform distribution $\mathcal{U}\big([0,1]^d\big)$ as a reference or baseline distribution $\mathcal{V}$ in our proof for the hardness result on approximating the rate distortion function. This can easily be extended to more general probability measures μ on $[0,1]^d$. We can choose μ as a product of any independent one-dimensional measures μ_i for the individual input components as long as for every $i \in [d]$ and $0 < \eta \le 1$ there exist lower and upper thresholds $a_i, b_i \in [0,1]$ with $a_i < b_i$ such that

$$\mu_i((-\infty, a_i]) = \mu_i([b_i, \infty)) \qquad \text{and} \qquad \mu_i([a_i, b_i]) \le \eta,$$

which is possible for all probability measures on $[0,1]$ without point masses, such as truncated Gaussian or exponential distributions. In that case we simply have to adapt the function Φ_η as $\varphi((\mathbf{x} - \mathbf{a}) \oslash (\mathbf{b} - \mathbf{a}))$ and proceed with the remaining proof as before (here $\oslash$ denotes component-wise division).

C.2. Conditional versus Marginal Distributions

We want to make another remark regarding the choice of the distributions $\mathcal{V}_S$ used in our rate distortion framework. We define the obfuscation $\mathbf{y}$ to be deterministically given by $\mathbf{y}_S = \mathbf{x}_S$ on S and distributed according to $\mathbf{y}_{S^c} = \mathbf{n}_{S^c}$ with $\mathbf{n} \sim \mathcal{V}$ on the complement S^c. This means that the resulting distribution $\mathcal{V}_S$ of $\mathbf{y}$ corresponds to $\mathcal{V}$ marginalised over all components in S. One might be tempted to condition on the given components $\mathbf{x}_S$ instead of marginalising. But this could actually be detrimental to uncover how the classifier operates. Let us illustrate this with an example. Consider a classifier that is trained to detect ships, but actually only learned to detect the water surrounding the ship, as in (Lapuschkin et al.,

2016). The classifier can achieve high accuracy as long as the data set only contains ships on water and no other objects surrounded by water. Now assume we have a relevance map selecting a subset of pixels showing a ship as relevant. If we complete the rest of the image with random values from a conditional distribution, we will most likely see water in the completion, as most images with a ship will also have water surrounding it. The classifier would correctly classify the completed image with high probability. The potentially small subset of pixels containing the ship will thus give a small distortion and will be considered relevant. However, this result is not useful to uncover the underlying workings of the network. It does not tell us that the network does not recognise ships but only the surrounding water. Using a very data adapted and restricted conditional distribution compensates the shortcoming of the network. That is why we advocate for using a less data adapted marginal distribution. In fact, we believe that using maximally uninformed distributions like uniform or truncated Gaussian distributions is beneficial for uncovering the network's reasoning.

D. Description of the Synthetic Binary Strings Experiment

Network Architecture Recall that the underlying binary classifier is given by the Boolean function

$$\Psi\colon \{0,1\}^d \to \{0,1\}, \quad \mathbf{x} \mapsto \bigvee_{i=1}^{d-k+1} \bigwedge_{j=i}^{i+k-1} x_j,$$

that checks binary strings of length d for the existence of a block of k consecutive ones. A ReLU network with two hidden layers that interpolates Ψ can be constructed as

$$\Phi(\mathbf{x}) = \mathbf{W}_3 \varrho\left(\mathbf{W}_2 \varrho(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2\right) + \mathbf{b}_3$$

with

$$\begin{aligned}
\mathbf{W}_1 &= \begin{bmatrix} \sum_{j=i}^{i+k-1} \mathbf{e}_j^T \end{bmatrix}_{i=1}^{d-k+1} \in \mathbb{R}^{(d-k+1)\times d} && \text{and} \quad \mathbf{b}_1 = -(k-1)\cdot \mathbf{1}_{d-k+1} \in \mathbb{R}^{d-k+1}, \\
\mathbf{W}_2 &= -\mathbf{1}_{d-k+1}^\top \in \mathbb{R}^{1\times(d-k+1)} && \text{and} \quad \mathbf{b}_2 = 1 \in \mathbb{R}^1, \\
\mathbf{W}_3 &= -1 \in \mathbb{R}^{1\times 1} && \text{and} \quad \mathbf{b}_3 = 1 \in \mathbb{R}^1,
\end{aligned}$$

where $\mathbf{e}_j$ is the j-th unit vector in $\mathbb{R}^d$. This network is purely constructed and not trained on any data. We use $d = 16$ and $k = 5$ in our experiment.

RDE Optimisation For the RDE optimisation we used the regularisation parameter $\lambda = 1.67 \cdot 10^{-3}$ and solved the resulting box-constrained optimisation problem via L-BFGS-B (Byrd et al., 1995).

The initial guess was simply chosen as the mean of $\mathcal{U}\big([0,1]^d\big)$ and not further tuned. As reference distribution $\mathcal{V}$ we used the Gaussian distribution with mean and variance equal to the mean and variance of $\mathcal{U}\big([0,1]^d\big)$. To estimate a good value for the regularisation parameter λ we solved the RDE optimisation problem for values $\lambda = 10^q$ with ten values of q spaced evenly in $[-5, 0]$. We compared the results visually and saw that $1.67 \cdot 10^{-3}$ yields a relevance map with a sparsity that corresponds well to the true block size $k = 5$.

Comparison Methods We used the Innvestigate[1] (Alber et al., 2018) toolbox for generating relevance mappings according to SmoothGrad (Smilkov et al., 2017) with a noise scale of 0.5 and 64 noise samples. We used the SHAP[2] toolbox to generate relevance mappings according to SHAP (Lundberg & Lee, 2017) and used the `DeepExplainer` method for deep network models with 1024 reference inputs drawn randomly from $\mathcal{U}\big([0,1]^d\big)$. Finally, we used the LIME[3] toolbox to generate relevance mappings according to LIME (Ribeiro et al., 2016). We used the local explanations of the `LimeTabularExplainer` method with 1024 reference inputs drawn randomly from $\mathcal{U}\big([0,1]^d\big)$.

[1] https://github.com/albermax/innvestigate
[2] https://github.com/slundberg/shap
[3] https://github.com/marcotcr/lime

References

Alber, M., Lapuschki, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Samek, W., Müller, K., Dähne, S., and Kindermans, P. iNNvestigate neural networks! *CoRR*, abs/1808.04260, 2018. URL http://arxiv.org/abs/1808.04260.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. doi: 10.1137/0916069.

Lapuschkin, S., Binder, A., Montavon, G., Muller, K.-R., and Samek, W. Analyzing classifiers: Fisher vectors and deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2912–2920, 2016.

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

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

Smilkov, D., Thorat, N., Kim, B., Viégas, F. B., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL http://arxiv.org/abs/1706.03825.