Explaining Neural Networks by Decoding Layer Activations

Johannes Schneider[1(✉)] and Michalis Vlachos[2]

[1] Institute of Information Systems, University of Liechtenstein, Vaduz, Liechtenstein
johannes.schneider@uni.li
[2] Department of Information Systems, HEC, University of Lausanne, Lausanne, Switzerland

Abstract. We present a 'CLAssifier-DECoder' architecture (*ClaDec*) which facilitates the comprehension of the output of an arbitrary layer in a neural network (NN). It uses a decoder to transform the non-interpretable representation of the given layer to a representation that is more similar to the domain a human is familiar with. In an image recognition problem, one can recognize what information is represented by a layer by contrasting reconstructed images of *ClaDec* with those of a conventional auto-encoder(AE) serving as reference. We also extend *ClaDec* to allow the trade-off between human interpretability and fidelity. We evaluate our approach for image classification using Convolutional NNs. We show that reconstructed visualizations using encodings from a classifier capture more relevant information for classification than conventional AEs. Relevant code is available at https://github.com/JohnTailor/ClaDec.

1 Introduction

Understanding a NN is a multi-faceted problem, ranging from understanding single decisions, single neurons and single layers, up to explaining complete models. In this work, we are interested in better understanding the decision of a NN with respect to one or several user-defined layers that originate from a complex feature hierarchy, as commonly found in deep learning models. In a layered model, each layer corresponds to a transformed representation of the original input. Thus, the NN succinctly transforms the input into representations that are more useful for the task at hand, such as classification. From this point of view, we seek to answer the question: *"Given an input X, what does the representation $L(X)$ produced in a layer L tell us about the decision and about the network?"*. To address this question, we propose a classifier-decoder architecture called *ClaDec*. It uses a decoder to transform the representation $L(X)$ produced by a layer L of the classifier, with the goal to explain that layer via a human understandable representation, i.e., one that is similar to the input domain. The layer in question provides the "code" that is fed into a decoder. The motivation for this architecture stems from the observation that AE architectures are good at (re)constructing high-dimensional data from a low-dimensional representation. The idea behind this, stems from the observation that the classifier to be

P. H. Abreu et al. (Eds.): IDA 2021, LNCS 12695, pp. 63–75, 2021.
https://doi.org/10.1007/978-3-030-74251-5_6

explained is expected to encode faithfully aspects relevant to the classification and ignore input information that does not impact decisions. Therefore, use of a decoder can lead to accurate reconstruction of parts and attributes of the input that are essential for classification. In contrast, inputs that have little or no influence to the classification will be reconstructed at lower fidelity. Attributes of an input might refer to basic properties such as color, shape, sharpness but also more abstract, higher-level concepts. That is, reconstructions of higher-level constructs might be altered to be more similar to prototypical instances.

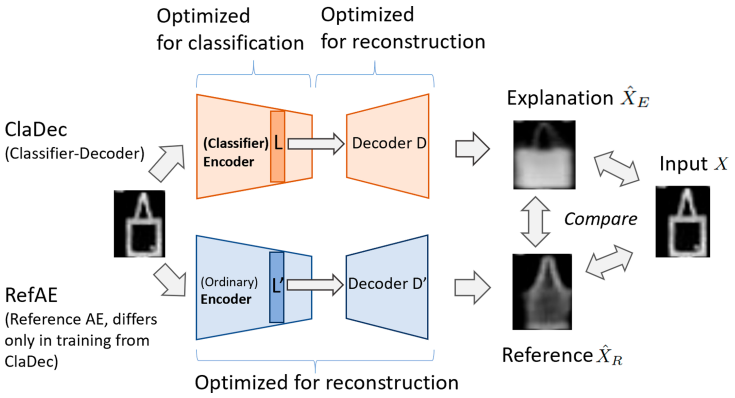

Fig. 1. Basic architecture of *ClaDec* and *RefAE* and explanation process

Explanations should fulfill many partially conflicting objectives. We are interested in the trade-off between fidelity (How accurately does the explanation express the model behavior?) and interpretability (How easy is it to make sense of the explanation?). While these properties of explanations are well-known, existing methods typically do not accommodate adjusting this trade-off. In contrast, we propose an extension of our base architecture *ClaDec* by adding a classification loss. It allows to balance between producing reconstructions that are similar to the inputs, i.e., training data that a user is probably more familiar with (easier interpretation), and reconstructions that are strongly influenced by the model to explain (higher fidelity) but may deviate more from what the user knows or has seen. Our approach relies on an auxiliary model, a decoder, to provide explanations. Similar to other methods that use auxiliary or proxy models, e.g., to synthesize inputs [10] or approximate model behavior [11], we face the problem that explanation fidelity may be negatively impacted by a poor auxiliary model. That is, reconstructions produced by AEs (or GANs) might suffer from artifacts. For example, AEs are known to produce images that might appear more blurry than real images. People have noticed that GANs can produce clearer images but they may suffer from other artifacts as shown in [10]. Neglecting that the explainability method might introduce artifacts can have an adverse impact on understandability and even lead to wrong conclusions on

model behavior. When looking at the reconstruction, a person not familiar with such artifacts might not attribute the distortion to the auxiliary model being used but she might believe that it is due to the model to be explained. While evaluation of explainability methods has many known open questions [19], this is the first work that has made this observation.

To avoid any wrongful perceptions with respect to artifacts in reconstruction, we suggest to compare outcomes of auxiliary models to a reference architecture. We employ an auto-encoder *RefAE* with the exact same architecture as *ClaDec* to generate outputs for comparison as shown in Fig. 1. The encoder of *RefAE* is not trained for classification, but the *RefAE* model optimizes the reconstruction loss of the original inputs as any conventional AE. Therefore, only the differences visible in the reconstructions of *RefAE* and *ClaDec* can be attributed to the model to be explained. The proposed comparison to a reference model can also be perceived as a rudimentary sanity check, i.e., if there are no differences then either the explainability method is of little value or the objective of the model to be explained is similar to that of the reference AE, as we shall elaborate more in our theoretical motivation. We believe that such sanity checks are urgently needed, since multiple explanation methods have been scrutinized for failing "sanity" checks and simple robustness properties [1,5,8]. For that reason, we also introduce a sanity check that formalizes the idea that inputs plus explanations should lead to better performance on downstream tasks than inputs alone. In our context, we even show that auxiliary classifiers trained on either reconstructions from *RefAE* or *ClaDec* perform better on the latter, although the reference AE leads to reconstructions that are closer to the original inputs. Thus, the reconstructions of *ClaDec* are more amendable for the task to be solved. Overall, we make the following **contributions**:
i) We present a novel method to understand layers of NNs. It uses a decoder to translate non-interpretable layer outputs into a human understandable representation. It allows to trade interpretability and fidelity.
ii) We introduce a method dealing with artifacts created by auxiliary models (or proxies) through comparisons with adequate references. This includes evaluation of methods.

2 Method and Architecture

The *ClaDec* architecture is shown on the top portion of Fig. 1. It consists of an encoder and a decoder reconstructing the input. The encoder is made of all layers of a classifier up to a user-specified layer L. The entire classifier has been trained beforehand to optimize classification loss. Its parameters remain unchanged during the explanation process. To explain layer L of the classifier for an input X, we use the activations of layer $L(X)$. The activations $L(X)$ are provided to the decoder. The decoder is trained to optimize the reconstruction loss with respect to the original inputs X. The *RefAE* architecture is identical to *ClaDec*. It differs only in the training process and the objective. For the reference AE, the encoder and decoder are trained jointly to optimize the reconstruction loss of inputs X.

In contrast, the encoder is treated as fixed in *ClaDec*. Once the training of all components is completed, explanations can be generated without further need for optimization. That is, for an input X, *ClaDec* computes the reconstruction $\hat{X}_E$ serving together with the original input and the reconstruction from *RefAE* as the explanation.

However, comparing the reconstruction $\hat{X}_E$ to the input X may be difficult and even misleading, since the decoder can introduce distortions. Image reconstruction in general by AEs or GANs is not perfect. Therefore, it is unclear, whether the differences between the input and the reconstruction originate from the encoding of the classifier or the inherent limitations of the decoder. This problem exists in other methods as well, e.g. [10], but it has been ignored. Thus, we propose to use both the *RefAE* (capturing unavoidable limitations of the model or data) and *ClaDec* (capturing model behavior). The evaluation proceeds by comparing the reconstructed "reference" from *RefAE*, the explanation from *ClaDec* and the input. Only differences between the input and the reconstruction of *ClaDec* that do not occur in the reconstruction of the reference can be attributed to the classifier. Figure 2 shows an extension of the base architecture of *ClaDec* (Fig. 1) using a second loss term for the decoder training. It is motivated by the fact that *ClaDec* seems to yield reconstructions that capture more aspects of the input domain than of the classifier. That is, reconstructions might be easy to interpret, but in some cases it might be preferable to allow for explanations that are more fidel, i.e. capturing more aspects of the model that should be explained.

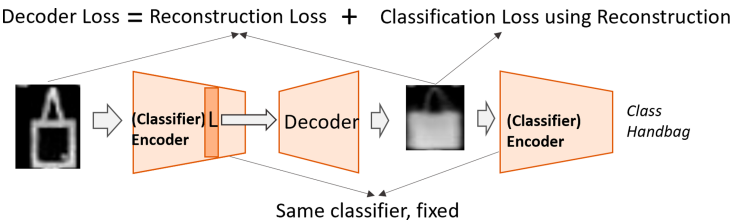

Fig. 2. Extension of the *ClaDec* architecture. The decoder is optimized for reconstruction and classification loss

More formally, for an input X, a classifier C (to be explained) and a layer L to explain, let $L(X)$ be the activations of layer L for input X, and $Loss(CL(X), Y)$ the classification loss of X depending on the true classes Y. The decoder D transforms the representation $L(X)$ into the reconstruction $\hat{X}$. For *ClaDec* the decoder loss is:

$$Loss(X) := (1-\alpha) \cdot \sum_i (X_i - \hat{X}_{E,i})^2 + \alpha \cdot Loss(CL(\hat{X}_E), Y) \quad \text{with } \hat{X}_E := D(L(X)) \text{ and } \alpha \in [0,1] \tag{1}$$

The trade-off parameter α allows to control whether reconstructions $\hat{X}_E$ are more similar to inputs with which the domain expert is more familiar, or reconstructions that are more shaped by the classifier and, thus, they might look more different than training data a domain expert is familiar with. For reconstructions $X_{R,i}$ of *RefAE* the loss is only the reconstruction loss $\sum_i (X_i - \hat{X}_{R,i})^2$.

3 Theoretical Motivation of *ClaDec*

We provide rational for reconstructing explanations using a decoder from a layer of a classifier that should be explained, and comparing it to the output of a conventional AE, i.e. *RefAE* (see Fig. 1). AEs perform a transformation of inputs to a latent space and then back to the original space. This comes with information loss on the original inputs because reconstructions are typically not identical to inputs. To provide intuition, we focus on a simple architecture with a linear encoder (consisting of a linear model that should be explained), a single hidden unit and a linear decoder as depicted in Fig. 3. An AE, i.e. the reference AE *RefAE*, aims to find an encoding vector E and a reconstruction vector R, so that the reconstruction $\hat{x} = R \cdot y$ of the encoding $y = E \cdot x$ is minimal using the L2-loss, i.e. $\min_{R,E} ||x - R \cdot E \cdot x||^2$. The optimal solution which minimizes the reconstruction loss stems from projecting onto the eigenvector space (as given by a Principal Component Analysis) [3]. That is, given there is just a single latent variable, the optimal solution for $W = R \cdot E$ is the first eigenvector u_1. This is illustrated in Fig. 3 in the upper part with $y = u_1 \cdot x$. For *ClaDec* the goal is to explain a linear regression model $y = E \cdot x$. The vector E is found by solving a regression problem. We fit the decoder R to minimize the reconstruction loss on the original inputs given the encoding, i.e. $\min_R ||x - R \cdot y||^2$ with $y = E \cdot x$. The more similar the regression problem is to the encoding problem of an AE, the more similar are the reconstructions. Put differently, the closer E is to u_1 the lower the reconstruction loss and the more similar are the optimal reconstructions for the reference AE and *ClaDec*. Assume that E differs strongly from u_1, i.e. say that the optimal solution to the regression problem is

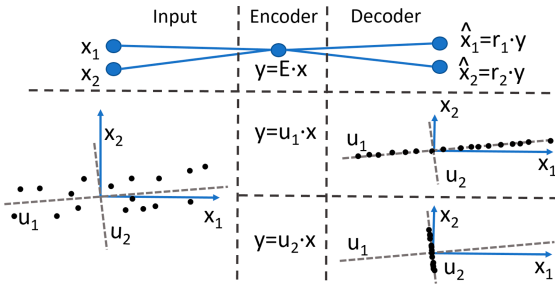

Fig. 3. An AE with optimal encoder $y = u_1 \cdot x$ (and decoder) captures more information than any other encoder. But a regression/classification model serving as encoder, e.g. $y = u_2 \cdot x$, combined with an optimized decoder, might capture some input attributes more accurately, e.g. x_2.

the second eigenvector $y = u_2 \cdot x$. This is shown in the lower part of Fig. 3. When comparing the optimal reconstruction of the *RefAE*, i.e. using $y = u_1 x$, and the illustrated reconstruction of *ClaDec*, i.e. using $y = u_2 x$, it becomes apparent that for the optimal encoding $y = u_1 x$ the reconstructions of both coordinates x_1 and x_2 are fairly accurate on average. In contrast, using $y = u_2 x$, coordinate x_2 is reconstructed more accurately (on average), whereas the reconstruction of x_1 is mostly very poor.

Generally, this suggests that a representation obtained from a model (trained for some task) captures less information than an encoder optimized towards reconstructing inputs. But aspects of inputs relevant to the task should be captured relatively in more detail than those that are irrelevant. Reconstructions from *ClaDec* should show more similarity to original inputs for attributes relevant to classification and less similarity for irrelevant attributes. But, overall reconstructions from the classifier will show less similarity to inputs than those of an AE.

4 Assessing Interpretability and Fidelity

Fidelity is the degree to which an explanation captures model behavior. That is, a "fidel" explanation captures the decision process of the model accurately. The proposed evaluation (also serving as sanity check) uses the rational that fidel explanations for decisions of a well-performing model should be helpful in performing the task the model addresses. Concretely, training a new classifier C^E_{eval} on explanations and, possibly, inputs should yield a better performing classifier than relying on inputs only. That is, we train a baseline classifier $C^R_{eval}(\hat{X}_R)$ on the reconstructions of the *RefAE* and a second classifier with identical architecture $C^E_{eval}(\hat{X}_E)$ on explanations from *ClaDec*. The latter classifier should achieve higher accuracy. This is a much stronger requirement than the common sanity check demanding that explanations must be valuable to perform a task better than a "guessing" baseline. One must be careful that explanations do not contain additional external knowledge (not present in the inputs or training data) that help in performing the task. For most methods, including ours, this holds true. Therefore, it is not obvious that training on explanations allows to improve on classification performance compared to training on inputs that are more accurate reconstructions of the original inputs. Improvements seem only possible if an explanation is a more adequate representation to solve the problem. Formally, we measure the similarity between the reconstructions $\hat{X}_R$ (using *RefAE*) and $\hat{X}_E$ (of *ClaDec*) with the original inputs X. We show that explanations (from *ClaDec*) bear less similarity with original inputs than reconstructions from *RefAE*. Still, training on explanations $\hat{X}_E$ yields classifiers with better performance than training on the more informative outputs $\hat{X}_R$ from *RefAE*.

Interpretability is the degree to which the explanation is human understandable. We build upon the intuitive assumption that a human can better and more easily interpret explanations made of concepts that she is more familiar with. We argue that a user is more familiar with real-world phenomena and concepts as captured in the training data than possibly unknown concepts captured

in representations of a NN. This implies that explanations that are more similar to the training data are more interpretable than those with strong deviation from the training data. Therefore, we quantify interpretability by measuring the distance to the original input, i.e. the reconstruction loss. If explanations show concepts that are highly fidelitous, but non-intuitive for a user (high reconstruction loss) a user can experience difficulties in making sense of the explanation. In contrast, a trivial explanation (showing the unmodified input) is easy to understand but it will not reveal any insights into the model behavior, i.e., it lacks fidelity.

5 Evaluation

In our qualitative and quantitative evaluation we focus on image classification using CNNs and the following experiments: (i) Explaining different layers for correct and incorrect classifications, (ii) Varying the fidelity and interpretability tradeoff. Our decoder follows a standard design, i.e. using 5×5 deconvolutional layers. For the classifier (and encoder) we used the same architecture, i.e. a VGG-5 and ResNet-10. For ResNet-10 we reconstructed after each block. Both architectures behaved similiarly, thus we only report for VGG-5. Note, that the same classifier architecture (but trained with different input data) serves as encoder in *RefAE*, classifier in *ClaDec* and for classifiers used for evaluation of reconstructions, i.e. classifier C^E_{Eval} (for assessing *ClaDec*) and C^R_{Eval} (for *RefAE*). The evaluation setup is shown in the right panel of Fig. 4 for *ClaDec*. Thus, we denote by "Acc Enc *ClaDec*" the validation accuracy of the encoder, i.e. classifier, of the *ClaDec* architecture and by "Acc Eval *RefAE*" the validation accuracy of the classifier C^R_{Eval} used for evaluation as shown in Fig. 4 trained on reconstructions from the reference AE. Other combinations are analogous.

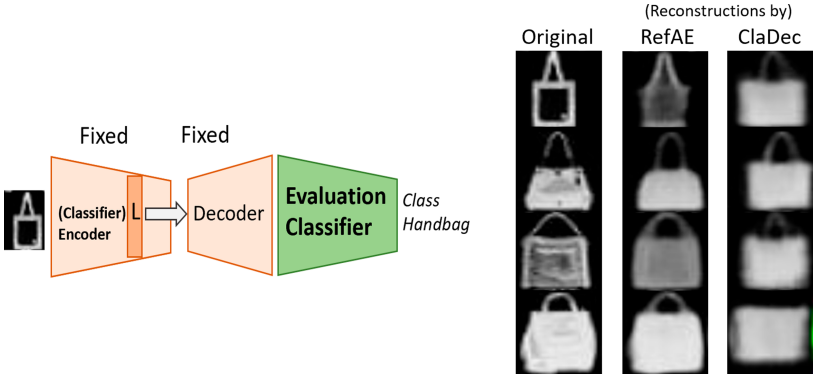

Fig. 4. Left panel: Evaluation setup using a dedicated evaluation classifier. Right panel: Comparison of original inputs and reconstructions using the FC layer of the encoder for handbags. Comparing *RefAE* and *ClaDec* shows that both do not reconstruct detailed textures. The classifier does not rely on graytones, which are captured by *RefAE*. It uses prototypical shapes.

Note that the decoder architecture varies depending on which layer is to be explained. The original architecture allows to either obtain reconstructions from the last convolutional layer or the fully connected layer. For a lower layer, the highest deconvolutional layers from the decoder have to be removed, so that the reconstructed image $\hat{X}$ has the same width and height as the original input X. We employed three datasets namely Fashion-MNIST, MNIST and TinyImageNet. Since all datasets behaved similarly, we focus on Fashion-MNIST consisting of 70000 28 × 28 images of clothing stemming from 10 classes that we scaled to 32 × 32. 10000 samples are used for testing. We train all models using the Adam optimizer for 64 epochs. That is, the *refAE*, the decoder of *ClaDec*, the classifier serving as encoder in *ClaDec* as well as the classifiers used for evaluation. We conducted 5 runs for each reported number. We show both averages and standard deviations.

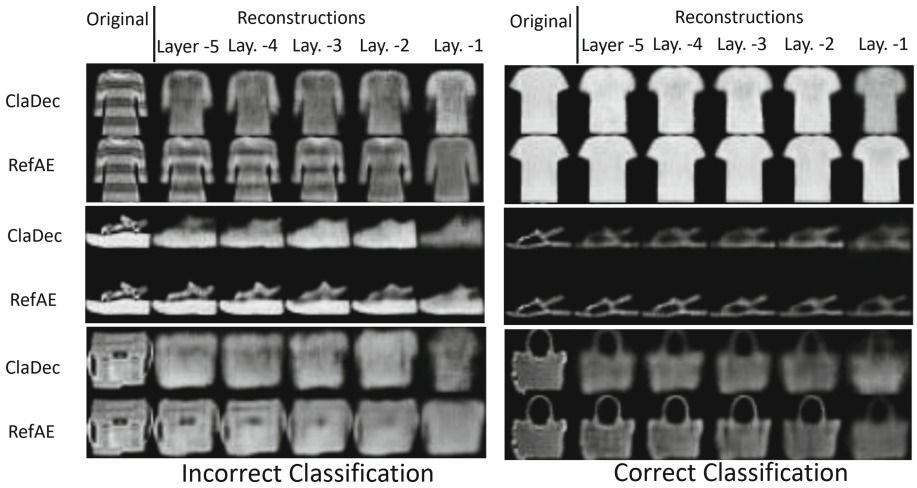

Fig. 5. Comparison of original inputs and reconstructions using multiple layers of the encoder. For incorrect samples it shows a gradual transformation into another class. Differences between *RefAE* and *ClaDec* increase with each layer

5.1 Qualitative Evaluation

Varying Explanation Layers: Reconstructions based on *RefAE* and *ClaDec* are shown in Figs. 4 and 5. For the last layer, i.e. the fully connected (FC) layer, there is only one value per class, implying a representation of 10 dimensions for Fashion-MNIST. For the handbags depicted in Fig. 4 and explained in the caption, comparing the original inputs and the reconstructions by *RefAE* and *ClaDec* shows clear differences in reconstructions. Some conclusions are knowledge of precise graytones is not used to classify these objects. Reconstructions from *ClaDec* resemble more prototypical, abstract features of handbags. Figure 5 shows reconstructions across layers. For all samples one can observe a gradual

abstraction resulting in change of shape and graytones as well as loss of details. The degree of abstraction varies significantly among samples, e.g. modest for T-Shirt and strong for handbags in the right panel. Reconstructions from *ClaDec* are more blurry than for *RefAE*. Blurriness indicates that the representation of the layer does not contain information needed to recover the details. However, the reason is not (primarily) distortions inherent in the decoder architecture, since *RefAE* produces significantly sharper images, but rather the abstraction process of the classifier. This is most apparent for incorrectly classified samples (left panel). One can observe a gradual modification of the sample into another class. This helps in understanding, how the network changed input features and at which layer. For example, the sandal (second row) appears like a sneaker at layer -3, whereas the reconstruction from *RefAE* still maintains the look of a sandal. The black "holes" in the sandals have vanished at layer -3 for the incorrectly classified sandal, whereas for the correctly classified sandal (right panel, same row), these holes remain. The bag (third row) only shows signs of a T-shirt in the second layer, where stumps of arms appear.

Fig. 6. Adding classification loss ($\alpha > 0$) yields worse reconstructions for the last conv. layer. Using classification loss only, reconstructions are not human recognizable.

Fidelity and Interpretability Trade-off: Figure 6 shows for the last conv. layer (second to last overall) the impact of adding a classification loss (Fig. 2) to modulate how much the model impacts reconstructions. Neglecting reconstruction loss, i.e. $\alpha = 1$, yields non-surprisingly non-interpretable reconstructions (not shown in Figure). Already modest reconstruction loss leads to well-recognizable shapes. The quality of reconstructions in terms of sharpness and amount of captured detail constantly improves the more emphasis is put on reconstruction loss. It also becomes evident that the NN learns "prototypical" samples (or features) towards which reconstructed samples are being optimized. For example, the shape of handbag handles shows much more diversity for values of α close to 0, it is fairly uniform for relatively large values of α. Thus, the parameter α provides a means to reconstruct a compromise between the sample that yields minimal classification loss and a sample that is true to the input. It suggests that areas of the reconstruction of *ClaDec* that are similar to the original input are also similar to a "prototype" that minimizes classification loss.

That is, the network can recognize them well, whereas areas that are strongly modified, resemble parts that seem non-aligned with "the prototype" encoded in the network.

5.2 Quantitative Evaluation

Varying Explanation Layers: Results in Table 1 contain two key messages: First, the reconstruction loss is lower for *RefAE* than for *ClaDec*. This is expected since *RefAE* is optimized entirely towards minimal reconstruction loss of the original inputs. Second, the classification (evaluation) accuracy is higher, when training the evaluation classifier C_{Eval} using reconstructions from *ClaDec* than from *RefAE*. This behavior is not obvious, since the reconstructions from *ClaDec* are poorer according to the reconstruction loss. That is, they contain less information about the original input than those from *RefAE*. However, it seems that the "right" information is encoded using a better suited representation. Aside from these two key observations there are a set of other noteworthy behaviors: As expected the reconstruction loss increases the more encoder layers, i.e. the more transformations of the input, are used. The impact is significantly stronger for *ClaDec*. The difference between *RefAE* and *ClaDec* increases the closer the layer to explain is to the output. This is not surprising, since lower layers are known to be fairly general, i.e. in transfer learning lower layers are the most applicable to work well for varying input data. There is a strong increase for the last layer, this is also no surprise, since the last layer consists of fairly fewer dimensions, i.e. 10 dimensions (one per class) compared to more than 100 for the second last layer. The classification accuracy for the evaluation classifier somewhat improves the more layers are used as encoder, i.e. of the classifier that should be explained. The opposite holds for *RefAE*. This confirms that *RefAE* focuses on the wrong

Table 1. Explaining layers: *ClaDec* has larger reconstruction loss but the evaluation classifier has higher accuracy on *ClaDec*'s reconstructions

Layer	Rec Loss *ClaDec*	Rec Loss *RefAE*	Δ	Acc Eval *ClaDec*	Acc Eval *RefAE*	Δ
-1	$28.6_{\pm 0.6851}$	$8.48_{\pm 0.3799}$	20.2	$0.89_{\pm 0.0031}$	$0.83_{\pm 0.0105}$	0.06
-3	$4.56_{\pm 0.0921}$	$3.63_{\pm 0.0729}$	0.93	$0.877_{\pm 0.0074}$	$0.863_{\pm 0.0093}$	0.014
-5	$1.93_{\pm 0.1743}$	$1.87_{\pm 0.0933}$	0.06	$0.878_{\pm 0.0073}$	$0.875_{\pm 0.0048}$	0.003

Table 2. Adding classification loss $\alpha > 0$ (Eq. 1) yields worse reconstructions, but higher evaluation accuracy

α	Total Loss *ClaDec*	Rec Loss	Classifier Loss	Acc Eval *ClaDec*
.0	$0.01_{\pm 0.0033}$	$285.5_{\pm 52.01}$	$0.0_{\pm 0.0}$	$0.9028_{\pm 0.0035}$
.001	$0.03_{\pm 0.0023}$	$25.4_{\pm 0.9292}$	$0.03_{\pm 0.0009}$	$0.9033_{\pm 0.0022}$
.1	$0.84_{\pm 0.0132}$	$8.35_{\pm 0.1195}$	$0.75_{\pm 0.0106}$	$0.9011_{\pm 0.0026}$
1.0	$7.49_{\pm 0.1119}$	$7.49_{\pm 0.1119}$	$4.4_{\pm 0.254}$	$0.8824_{\pm 0.0042}$

information, whereas the classifier trained towards the task focuses on the right information and encodes it well.

Fidelity and Interpretability Tradeoff: Table 2 shows that evaluation accuracy increases when adding a classification loss, i.e. $\alpha > 0$ yields an accuracy above 90% whereas $\alpha = 0$ gives about 88%. Reconstructions that are stronger influenced by the model to explain (larger α) are more truthful to the model, but they exhibit larger differences from the original inputs. Choosing α slightly above the minimum, i.e. larger than 0, already has a strong impact.

6 Related Work

We categorize explainability methods [13] into methods that synthesize inputs (like ours and [10,20]) and methods that rely on saliency maps [18] based on perturbation [11,21] or gradients [2,16]. Saliency maps show feature importance of inputs, whereas synthesized inputs often show higher level representations encoded in the network. Perturbation-based methods include occlusion of parts of the inputs [21] and investigating the impact on output probabilities of specific classes. Linear proxy models such as LIME [11] perform local approximations of a black-box model using simple linear models by also assessing modified inputs. Saliency maps [18] highlight parts of the inputs that contributed to the decision. Many explainability methods have been under scrutiny for failing sanity checks [1] and being sensitive to factors not contributing to model predictions [8] or adversarial perturbations [5]. We anticipate that our work is less sensitive to targeted, hard to notice perturbations [5] as well as translations or factors not impacting decisions [8], since we rely on encodings of the classifier. Thus, explanations only change if these encodings change, which they should. The idea to evaluate explanations on downstream tasks is not new, however a comparison to a "close" baseline like our *RefAE* is. Our "evaluation classifier" using only explanations (without inputs) is more suitable than methods like [14] that use explanations together with inputs in a more complex, non-standard classification process. Using inputs and explanations for the evaluation classifier is diminishing differences in evaluation outcomes for any compared methods since a network might take missing information in the explanation from the input. So far, inputs have only been synthesized to understand individual neurons through activation maximization in an optimization procedure [10]. The idea is to identify inputs that maximize the activation of a given neuron. This is similar to the idea to identify samples in the input that maximize neuron activation. [10] uses a (pre-trained) GAN on natural images relevant to the classification problem. It identifies through optimization the latent code that when fed into the GAN results in a more or less realistic looking image that maximally activates a neuron [20] uses regularized optimization as well, yielding artistically more interesting but less recognizable images. Regularized optimization has also been employed in other forms of explanations of images, e.g. to make human understand how they can alter visual inputs such as handwriting for better recognizability by a CNN [12]. [6,7] allow to investigate high level concepts that are relevant to a

specific decision. DeepLift [17] compares activations to a reference and propagates them backwards. Defining the reference is non-trivial and domain specific. [9] estimates the impact of individual training samples. [4] uses a variational AE for contrastive explanations. They use distances in latent space to identify samples which are closest to a sample X of class Y but actually classified as Y'.

7 Conclusions

Our explanation method synthesizes human understandable inputs based on layer activations. It takes into account distortions originating from the reconstruction process. It is verified using novel sanity checks. In the future, we plan to investigate differences among networks, e.g. a result of model fine-tuning as in personalization [15] or to look at subsets of layer activations.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Neural Information Processing Systems (2018)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
3. Baldi, P., Hornik, K.: Neural networks and principal component analysis: Learning from examples without local minima. Neural Netw. **2**(1), 53–58 (1989)
4. van Doorenmalen, J., Menkovski, V.: Evaluation of CNN performance in semantically relevant latent spaces. In: Berthold, M.R., Feelders, A., Krempl, G. (eds.) IDA 2020. LNCS, vol. 12080, pp. 145–157. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44584-3_12
5. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: AAAI Conference on Artificial Intelligence (2019)
6. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. In: Advances in Neural Information Processing Systems (2019)
7. Kim, B., et al.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). arXiv preprint arXiv:1711.11279 (2017)
8. Kindermans, P.J., et al.: The (un) reliability of saliency methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning (2019)
9. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of International Conference on Machine Learning, pp. 1885–1894 (2017)
10. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Advances in Neural Information Processing Systems, pp. 3387–3395 (2016)
11. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you?: explaining the predictions of any classifier. In: SIGKDD (2016)
12. Schneider, J.: Human-to-AI coach: improving human inputs to AI systems. In: International Symposium on Intelligent Data Analysis (2020)
13. Schneider, J., Handali, J.P.: Personalized explanation for machine learning: a conceptualization. In: European Conference on Information Systems (ECIS) (2019)

14. Schneider, J., Vlachos, M.: Reflective-net: learning from explanations. In: arxiv: 2011.13986 (2020)
15. Schneider, J., Vlachos, M.: Personalization of Deep Learning. Data Science – Analytics and Applications, pp. 89–96. Springer, Wiesbaden (2021). https://doi.org/10.1007/978-3-658-32182-6_14
16. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision (ICCV), pp. 618–626 (2017)
17. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: International Conference on Machine Learning, pp. 3145–3153 (2017)
18. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv: 1312.6034 (2013)
19. Yang, F., Du, M., Hu, X.: Evaluating explanation without ground truth in interpretable machine learning. arXiv preprint arXiv:1907.06831 (2019)
20. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579 (2015)
21. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53