Semantic Explanation for Deep Neural Networks Using Feature Interactions

BOHUI XIA, XUETING WANG, and TOSHIHIKO YAMASAKI, The University of Tokyo, Japan

Given the promising results obtained by deep-learning techniques in multimedia analysis, the explainability of predictions made by networks has become important in practical applications. We present a method to generate semantic and quantitative explanations that are easily interpretable by humans. The previous work to obtain such explanations has focused on the contributions of each feature, taking their sum to be the prediction result for a target variable; the lack of discriminative power due to this simple additive formulation led to low explanatory performance. Our method considers not only individual features but also their interactions, for a more detailed interpretation of the decisions made by networks. The algorithm is based on the factorization machine, a prediction method that calculates factor vectors for each feature. We conducted experiments on multiple datasets with different models to validate our method, achieving higher performance than the previous work. We show that including interactions not only generates explanations but also makes them richer and is able to convey more information. We show examples of produced explanations in a simple visual format and verify that they are easily interpretable and plausible.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**;

Additional Key Words and Phrases: Explainability, multimedia analysis, semantics

ACM Reference format:

Bohui Xia, Xueting Wang, and Toshihiko Yamasaki. 2021. Semantic Explanation for Deep Neural Networks Using Feature Interactions. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 3s, Article 115 (November 2021), 19 pages.

<https://doi.org/10.1145/3474557>

1 INTRODUCTION

Deep learning techniques have developed quickly in recent years, achieving high performance in such diverse areas as computer vision [16, 30, 46] and natural language processing [29]. Although research has shown the discriminative power of **deep neural networks (DNNs)**, the grounds on which such networks make decisions have not been fully clarified. This is a major problem: without explanations for decisions, users cannot be certain that their models learn the proper knowledge. Moreover, convincing explanations for models have additional special value in business situations, where they can be used to improve services or products. Thus, many studies have been reported aiming at the explainability problem [1, 3, 34, 47].

This research is partially supported by the Grants-in-Aid for Scientific Research Numbers JP18H03339 and JP19K20289 from JSPS.

Authors' address: B. Xia, X. Wang, and T. Yamasaki, The University of Tokyo, Tokyo, Japan; emails: {xia, xt_wang, yamasaki}@hal.t.u-tokyo.ac.jp.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2021 Copyright held by the owner/author(s).

1551-6857/2021/11-ART115 \$15.00

<https://doi.org/10.1145/3474557>

There are various methods for explaining networks. In the case of **convolutional neural networks (CNNs)**, one of the most popular strategies is to visualize the areas where the model is paying attention [36]. These methods measure gradients of CNN weights in response to the change in output; they are not able to capture semantic information. To solve this problem, several network analysis studies have attempted to produce more semantic and interpretable visualization methods [52]. However, these were done by incorporating explanation modules into models that were designed for other tasks, sacrificing performance. Other studies developed post hoc explaining models using knowledge distillation: explainable models were trained to imitate models designed for the original tasks [6].

Chen et al. [3] also employed knowledge distillation to produce semantic and quantitative explanations: explaining models could produce explanations for prediction while learning to output the same values as the original predictors. One of the methods they introduced describes attributes by the addition of contributions of visual concepts. More specifically, after they trained a predictor for multi-label classification, they obtained the contributions of each pre-defined visual concept to a target attribute's classification result by calculating the product of a classification probability and its weight. They then trained an "explainer" so that the sum of the contributions became close to the classification result of the target attribute from the original predictor. In this way, they produced humanly comprehensible explanations without harming the performance of the original tasks; nevertheless, there is a drawback to their method: the performance of the explainer is lower than that of the original problem-solving model, predictor, by a large margin. This is due to the simplicity of the composition of the explainer; that is, the target prediction by the explainer is simply the sum of the contributions of the visual concepts.

In order to improve the performance while preserving or even enhancing the explanation, we propose a method of producing explanations using feature interactions. We were inspired by the **factorization machine (FM)** [33]. FM combines the advantages of support vector machines and factorization models. It has been a popular option for predictive tasks such as click-through rate prediction and advertisement recommendation. Although FM is not a deep-learning-based method, some recent methods using DNNs have incorporated it into their architectures [5, 11]. In our work, we extend the method of Chen et al. [3], described above, to make the best use of FM. There have been few works that introduce feature interactions as explanations of DNNs. In contrast to the interactions computed in these existing methods, ours calculates factor vectors, which are used for calculating interactions, for every input. This calculation method enables us to consider feature interactions as explanations.

The flow of our proposed architecture is as follows. We first train a predictor to predict all the attributes; then we train an explainer that can take feature interactions into account. Weights for attributes and factor vectors are obtained from the explainer to calculate the contributions of each individual attribute and their interactions. These contributions are our explanations for the prediction results. This method enables us to analyze explanations quantitatively and semantically. The sum of the contributions and a bias term is assumed as an output value from the explainer, and the model is trained to output values close to the prediction models that the explainer model mimics. In other words, the explainer model gains knowledge from the prediction model and imitates its decision logic. Note that the interactions in our method are different from those of FM in that factor vectors are not simple embeddings of features, but variables derived from each input.

The experiments, including both classification and regression tasks, verify the efficiency of our method. Because each dataset has different types of input/output and tasks (classification and regression), we have designed and trained different models on each one. Our method, although intended mainly for visual analyses, is applicable to multimedia data, so we also tested it on a TV ads dataset. A comparison between our approaches and models without interactions shows

that ours can achieve higher performances. Furthermore, visualized explanations are displayed to prove that these explanations are humanly interpretable and become richer thanks to interactions. The contributions of this work are as follows:

- We propose a method of explaining networks with the aid of feature interactions that yields easily interpretable semantic and quantitative explanations. The weight factor vector of FM is dynamically tuned depending on the input data.
- We demonstrate different kinds of networks for the multiple datasets used in our experiments so as to cope with various types of input and output.
- Our method performs better on the datasets than does a similar method without interactions.

2 RELATED WORKS

2.1 Explainability in Deep Learning

Visualization: When a CNN is the base of a model, visualization methods are often used to explain the decisions of the networks. When the areas where the model is focusing are highlighted, it becomes clear which parts affect the prediction result significantly. One visualizing strategy is to illustrate gradients of weights in layers caused by back propagation through trained models [8, 27, 37, 48, 49]. In this strategy, inverse operations are defined for every layer in the CNN model; gradients can then be taken from any layer to analyze their nature individually. **Class activation map (CAM)** is one of the most common visualization techniques to highlight areas corresponding to specific classes, and its successor model Grad-CAMs [36] are based on the same principles. Grad-CAMs use activation maps and gradients of weights, instead of weights as in CAM, to make heatmaps. One drawback of these types of approaches is that they are mainly targeted only at classification problems.

Another major explanatory strategy aimed at visualization employs perturbation methods. These methods learned the relationship between inputs and outputs by adding changes to the inputs and calculating the extent to which the outputs are altered by the change. Perturbation has been implemented in various ways, such as sliding an occlusion mask over an input image [34, 48, 54], masking a word in input sentences, or masking a feature from a hidden layer [21]. Zintgraf et al. [55] proposed a method for calculating the relevance between regions in an input image and classes. They estimated the conditional probability of the presence of a certain class under the condition that a part of an image was perturbed; the difference between the probability and the original prediction result for the whole image was assumed to be the contribution made by the perturbed area to the classification result.

Model revision for interpretable features: There are many studies that, instead of taking a post hoc approach, attempt to improve the model itself that has been reported so that interpretable features can be obtained [42]. Generative models have developed so remarkably that the generated images are almost completely natural, but the lack of interpretability in the latent space has limited the range of applications. In order to interpret generative-model decision-making and control attributes in generated images, disentanglement has been explored [4, 15]. Zhang et al. [52] proposed an interpretable CNN in which each filter in a high convolutional layer represents a specific object part by introducing a loss for each filter.

Joint training: Visualization methods that involve analyzing pre-trained models have limited expressiveness. On the other hand, incorporating interpretability into models can damage their discriminative power. To overcome these problems, approaches have emerged that introduce additional tasks [17, 19, 31]. In this approach, not only the original model but also additional models solving other tasks are trained jointly. One of the most intrinsic strategies is to generate explanations in text format. Hendricks et al. [14] have proposed a phrase-critic model to refine candidate

explanatory sentences by comparing their accumulated scores for each noun they contain and selecting the one that is the most class and image relevant. Zellers et al. [50] have formulated a new task called visual sense reasoning, the task of answering questions with a thorough visual understanding. They have collected data that contains questions, answers, and rationales and introduced a method for solving the task that consists of grounding, contextualization, and reasoning. Another joint training method is explanation by prototypes: the prediction result of an input image is explained by a subset of the training datasets [2, 22, 28].

Knowledge distillation: To enhance interpretability, it has been suggested that an explainable model could distill knowledge from a prediction model [6, 12]. Various ways of distilling knowledge into more interpretable models, such as decision trees and graphs, have been investigated [10, 51]. This kind of strategy is applicable not only to images but also to other domains such as videos [18]. Our work is closely related to one of the distillation methods proposed by Chen et al. [3], which describes prediction results using pre-defined visual concepts, with a prior weight to prevent biased interpretation. We describe this method in more detail in Section 3. Our modification of it achieves higher performance by enhancing its discriminative power. In addition, our method can generate more detailed explanations.

2.2 Feature Interaction

Implicit interaction interpretation: There have been some approaches to detect and interpret feature interactions [9, 35]. Some works have tackled interpretation of complex models using features [26, 39]. However, feature interactions were not discussed in these methods. One of the most recent works about detecting interactions in neural architectures is [44], the goal of which is closely related to ours. Our method, however, is aimed to explicitly consider interactions and furthermore is capable of comprehensive interpretation including not only interactions but also each feature separately.

Learning explicit interaction: In order to improve prediction performances for that high-dimensional and sparse data that often can be seen in recommender systems, factorization models have gained popularity, such as Matrix Factorization [20, 41]. Rendle [33] proposed FM, which combines the advantages of support vector machines and the factorization models. FM works as a general predictor with any real-valued feature vector. By using factorized parameters, it can model interactions between all the input parameters even in the case of sparsity. As DNNs have gained popularity, there have been many other attempts to integrate interactions in deep learning models [5, 11, 23, 32, 40, 45, 53].

3 APPROACH

In this section, we explain how our model works in detail. First, we provide an overview of our whole architecture, and then illustrate its components individually.

3.1 Overview

An overview of our model is shown in Figure 1. Our goal is to explain prediction results by the contributions from every individual attribute and their interactions; hence, our model is a combination of two sub-models that we term the “performer” and “explainer,” following [3]. First, a prediction task is solved by the performer; then, an explanation is generated by the explainer using the predicted values. The prediction results are explained by the contributions of single features and their interactions. More precisely, we acquire vectors from the explainer whose sizes are the total number of explanatory attributes plus that of combinations of every attribute. The vector represents numerical contributions to the prediction result, by which we can tell how much each

Fig. 1. Overview of our model. It is composed of a performer model that performs a prediction task and an explainer model that explains the prediction result. The explainer distills knowledge from the performer. Face image is from CelebA dataset [25].

attribute or interaction contributes to the result, and we treat the vector as an “explanation” for the object network.

We will define the variables used in Figure 1 and the following sections before entering into the details. We assume the output value from the performer is $\hat{y}_{pf}$, and the output value from the explainer is $\hat{y}_{ex}$ given an input instance I . The predictors of a target attribute in the performer are denoted by F , and those of the other attributes by $f = \{f_1, f_2, \dots, f_n\}$, where n represents the number of attributes used to explain the prediction. The attribute’s prediction results are $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$, whereas $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ is a vector that denotes the weight for single attributes and $\mathbf{V} = [{}^t\mathbf{v}_1, {}^t\mathbf{v}_2, \dots, {}^t\mathbf{v}_n]^T$ is a matrix containing factor vectors. The explainer consists of g (for producing $\mathbf{w}$) and h (for producing $\mathbf{V}$). The size of a factor vector is set to l . These variables are used to calculate the contributions, as we explain in the next section.

3.2 Explainer Algorithm

In this section, we illustrate how the explainer works and how to acquire explanations for the predictor results. We formulate the explainer model as

$$\begin{aligned}
 \hat{y}_{ex} &= b + \langle g(I), f(I) \rangle + \sum_{i=1}^n \sum_{j=i+1}^n \langle h_i(I), h_j(I) \rangle f_i(I) f_j(I) \\
 &= b + \sum_{i=1}^n w_i y_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle y_i y_j,
 \end{aligned} \tag{1}$$

where $y_i = f_i(I)$, and $\langle \cdot \rangle$ means an inner product of two vectors. The first term in the equation is a bias, the second refers to the contributions from attributes, and the third to contributions made by attribute interactions. In the second term, $w_i y_i$ is the product of a weight from the explainer and a predicted value from the performer, measuring the extent to which the i th attribute contributes to the target attribute. Regarding the third term, we newly add this to the formula of the previous method [3]. $\langle \mathbf{v}_i, \mathbf{v}_j \rangle y_i y_j$ represents the interaction between the i th and j th attributes. It is defined

as the product of their predicted values from the performer with the inner product of the i th and j th factor vectors.

It is important to note that the weights $\mathbf{w}$ and the factor vectors $\mathbf{V}$ are dynamically calculated depending on the attributes as well as the input instance and the parameters in the explainer, which means that every set of input data produces a different set of factor vectors; this contrasts with the usual linear regression or FM model. These variables thus give us more expressive power than do traditional methods.

We will now take a closer look at the interaction term. A straightforward approach to computing the contribution of the interactions would be calculating each interaction for every pair of attributes and then adding them up. However, this would be a very time-consuming operation, because its computational complexity is of $O(ln^2)$. To reduce the computational load, we follow the reformulation used in the original FM [33]. The third term in Equation (1) is calculated as follows in an actual model, omitting deformations in the middle:

$$\sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle y_i y_j = \frac{1}{2} \sum_{f=1}^l \left(\left(\sum_{i=1}^n v_{i,f} y_i \right)^2 - \sum_{i=1}^n v_{i,f}^2 y_i^2 \right). \quad (2)$$

This reduces the complexity to $O(ln)$. In the training phase, interactions are calculated in this way. By contrast, in the testing phase, we calculate all pairwise interactions so as to clarify which kinds of attribute interactions make large contributions and which do not.

3.3 Training Process

First, the performer is trained to solve a prediction task, for example, multilabel classification and regression. We use cross entropy as the performer's loss function for classification tasks and mean squared error for regression tasks. Then, the explainer is trained using the output values from the performer. The value predicted by the explainer, which is calculated in the manner illustrated in the previous section, is expected to be similar to the prediction results by the performer because we train the explainer to mimic the behavior of the performer. The loss function for training the explainer is defined as follows:

$$Loss = \|\hat{y}_{ex} - \hat{y}_{pf}\|^2 + PriorLoss. \quad (3)$$

The loss function works for minimizing the mean squared error between the performer and explainer outputs. *PriorLoss* was proposed in [3] as a solution for the problem of biased interpretation: simply minimizing the error between the performer and explainer outputs makes the explainer select fewer attributes, resulting in biased explanations. To avoid this, the prior weights $\mathbf{p}$ are approximated as the derivatives of $\hat{y}_{pf}$ with respect to y_k ($k \in 1, 2, \dots, n$), and the difference between the priors and the weights is minimized. The definition of the loss function is $PriorLoss = \frac{\beta}{\sqrt{t}} \|\frac{\mathbf{w}}{\|\mathbf{w}\|^2} - \frac{\mathbf{p}}{\|\mathbf{p}\|^2}\|_2^2$, where t represents the current epoch, β is a constant, and $\|\cdot\|_2$ means the L-2 norm. We use this loss function in some of our experiments to penalize the weights of the additive function of attribute contributions.

4 EXPERIMENTS

We used three datasets to validate our method. Each dataset has a different domain of input data and a different type of annotation. By testing our model in different settings, it can be verified to be useful in many applications.

Fig. 2. Model architecture used for the experiments on CelebA dataset and DeepFashion dataset. The Face image is from CelebA dataset [25].

Table 1. Architectural Specification of the Performer and Explainer's Prediction Head

Layer	Specification
Linear	2048×256
ReLU	–
Dropout	–
Linear	256×256
ReLU	–
Dropout	–
Linear	$256 \times (n-1 \text{ or } l)$

The performer's prediction head follows VGG16, and the explainer's follows ResNet152. The dimension of the last output is l only in h_k to regress factor vectors.

4.1 Experiment 1: CelebA Dataset

The first dataset is the CelebA dataset [25]. This is a face attribute dataset including 200K celebrity images, each with 40 attribute annotations such as “Eyeglasses” and “Smiling.” In this experiment, we set “Attractive,” “Heavy Makeup,” “Male,” and “Young” as the attributes to be explained by the rest. These global attributes are selected as targets because they can be intuitively explained by combinations of other local features.

The model architecture is illustrated in Figure 2. The input is an image. We use VGG16 [38] as a base model for a performer and ResNet152 [13] for an explainer; both are pretrained on ImageNet [7]. F and f_k serve as the performer's prediction heads that output predictions for each attribute. The explainer's prediction heads, g and h_k , which regress the weight and factor-vector for each attribute, share the same architecture with different parameters following ResNet152.

Layers composing these models are listed in Table 1. The number of explanatory attributes n is 39. The dimension of factor vectors l is set to 2.

We first train a performer with a cross-entropy loss and then an explainer with the loss function of Equation (3). In *PriorLoss*, β is set to 10.

4.2 Experiment 2: DeepFashion Dataset

The second dataset is the DeepFashion dataset [24]. The DeepFashion database includes many benchmarks available for various purposes. We select the Category and Attribute Prediction Benchmark because it contains rich annotations suitable for explanations. Coarse annotation has five types of attributes: texture, figure, shape, part, and style. Because the annotation includes as many as 1,000 attributes, we reduce the number of attributes and data. The benchmark includes many kinds of clothes (denim jacket, long skirt, T-shirt, etc.). In order to limit the number of attributes, we choose only data from the tops. Then the 100 most frequent attributes are selected, for example, “Print,” “Knit,” and “Shirt;” the rest are abandoned. As a result, the number of data points is about 140K. In this experiment, we select “Classic,” “Basic,” “Cute,” and “Soft” as attributes to be explained by the other attributes, as they describe clothes’ global features.

The model architecture used in this experiment and its training process is the same as that of Experiment 1. The input is an image. The number of explanatory attributes n is 99. The dimension of factor vectors is the same as that of Experiment 1, which is 2.

4.3 Experiment 3: TV Ads Dataset

The last dataset used is the TV ads dataset, a collection of 14,990 commercial videos that were actually broadcast on TV in Japan between January 2006 and April 2016. Each video was evaluated and annotated by 600 participants. The dataset was collected to predict the following four impressional and emotional effects:

- Favorability rating (F): how much participants liked the content of the advertisement itself
- Interest rating (I): how much participants became interested in the product/service
- Willingness rating (W): how much participants felt like buying the product/service
- Recognition rating (R): how much participants remembered the advertisement

Besides the videos, the dataset contains metadata such as information about the casts featured in the ad. In addition, scores are given to 26 attributes that describe the ad, such as “Good story” and “Impressive.” In the present experiment, we attempt to explain each of the four effects of the attributes.

The effects and the attributes are continuous values, not binary labels, and the performer’s prediction task is necessarily a regression problem, in contrast to the previous two experiments. Hence, a different architecture is needed. We illustrate the model in Figure 3. Input data consists of frame deep features extracted from video, sound, metadata, cast data, text in frames, and narration data. As the base model for both the performer and the explainer, we employ a multimodal fusion model using an attention mechanism proposed in previous research [43]. F regresses one of the four effects; f outputs a vector $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ (where n is 26) that represents the predicted attributes. In contrast to the model in Figure 2, F and f output the target prediction and explanatory attributes prediction independently. The architecture of the explainer is otherwise similar to that in Figure 2: g and h share the base model, and its branches produce $\mathbf{w}$ and $\mathbf{V}$. We set β in Equation (3) to 0; that is, we do not employ *PriorLoss* in this experiment. l is set to 2 similarly.

Fig. 3. The model architecture used in Experiment 3. The performer solves regression tasks. The performer and explainer have the same multimodal prediction module.

4.4 Results

We show the accuracy or correlation coefficients of each experiment in Tables 2, 4, and 6. We also show the conditional entropy in Tables 3, 5, and 7. The first row shows the result from the explainer in the method of Chen et al. [3], the second row shows our method’s explainer, and the third row shows the performer. In the experiment, we compared our method to the previous method to show that feature interaction improves the explainer’s performance. Conditional entropy of explanation presented in the previous work [3] is not appropriate for evaluating our method because the weights of attributes and their interactions were not approximated as they were in the previous one.

For Experiments 1 and 2, we reimplemented the previous method to use it as a comparative method. There could be a slight difference between our implementation and that of Chen et al. [3], since their paper misses some details about the model; nevertheless, as the first and third rows in Table 2 show, the performer and explainer implemented by us achieve almost the same performances as those reported in the paper. This implies that our implementation can accurately reproduce the method of Chen et al. [3].

Table 2 shows the results of the experiment on the CelebA dataset. As mentioned above, our proposed method is compared with an interaction-free method from the literature. From the table it can be seen that explainers perform better with feature interaction whatever the target attribute, and attain accuracies close to those of performers. This indicates that feature interaction is capable of increasing both explainability and the model’s discriminative power at the same time. To verify that our model using interaction can produce reasonable explanations, we display an example in Figures 4 and 5, picked from test data in the CelebA dataset. These are explanations of why the performer judged the image to be “Attractive.” The horizontal axis is the attribute label and the vertical axis is the contribution to the prediction. Figure 4 shows an explanation produced with the method of Chen et al. [3]. The 20 largest contributions are sorted in descending order. Figure 5 shows an explanation produced by our method. The top row shows the contribution from single attributes and the bottom row shows that from attribute interactions (the 20 largest for each).

Table 2. Results of Experiment 1 on the CelebA dataset

	Attractive	Heavy Makeup	Male	Young
Explainer w/o interaction [3]	0.789	0.899	0.960	0.812
Explainer w/ interaction (ours)	0.815	0.911	0.970	0.875
Performer	0.819	0.912	0.977	0.881

The evaluation metric is classification accuracy.

Table 3. Results of Experiment 1 on the CelebA Dataset

	Attractive	Heavy Makeup	Male	Young
Explainer w/o interaction [3]	9.81	9.80	9.81	9.81
Explainer w/ interaction (ours)	9.80	9.81	9.81	9.82
Performer	9.85	9.86	9.81	9.88

The evaluation metric is the conditional entropy of the prediction.

The previous method already achieved quantitative and semantic explanations; however, ours is able to consider not only single-attribute contributions but also interactions, resulting in more unbiased and insightful explanations. Examining Figure 5 in more detail, it is suggested by the explainer that attributes such as “Double chin” and “Bushy eyebrows” contribute to “Attractive” for this face image and so do the attribute interactions, including “No beard & Young” and “Male & Young.” The explanation is reasonable and easily interpretable by humans. In addition, it is observed in the explanation that contributions made by feature interactions such as “No beard & Young” are larger than those made by single features such as “Double chin.” It can be estimated that the performer highlights the feature interaction when it performs prediction, and our method is able to successfully detect that.

Table 4 shows the results of the experiment on the DeepFashion dataset. As can be seen, introducing interactions to the explainer is not helpful for improving prediction performance in this domain. There are two possible reasons for this. The first is that the annotations are so coarse that the interactions contain more error. Since interaction is the product of the probabilities of two attributes and factor vectors, error tends to be amplified. The second possible reason is that the task itself is so simple that explanation models can easily converge to the optimum regardless of variation of explanatory variables. We find that conditions such as the number of attributes or complexity of the model are strongly related to the quality of explanation and prediction performance and thus need to be carefully designed. For a more detailed analysis, we present an example of explanations produced by the method of Chen et al. and by ours in Figures 6 and 7, respectively. These explanations were produced to explain why the image displayed on the top is “Classic.” For example, according to Figure 6, the second most important reason for being “Classic” is “New York,” while it is hard to determine whether the cloth can be categorized as “New York.” Furthermore, Figure 7 shows that interactions between “Collar & Button” or between “Collar & Pleated” are the most significant factors, although “Button & Pleated” are unseen in the image. Similarly, other examples contain wrong attributes in their explanations.

Table 6 compares prediction results of the experiment on the TV ads dataset. Different from the previous two experiments, the results are evaluated by Pearson’s correlation coefficients, since the targets are continuous values. It can be seen that the explainer achieves higher performance when interactions are incorporated, except on the Favorability rating. This implies that

Fig. 4. Example of explanations produced by the explainer without interactions in experiments on the CelebA dataset. This and the one below explain a prediction result for “Attractive.”

Fig. 5. Example of explanations produced by the explainer with interactions in experiments on the CelebA dataset [25]. Note that the scale between the two rows is different.

considering interactions is valid for various tasks including regression. Figures 8 and 9 give examples of the explanations produced by the two methods explaining the ad’s Favorability rating.¹ For the interaction-free explanation, attributes such as “Familiar” and “Empathetic” are dominant causes. By contrast, the explanation with interactions similarly takes “Familiar” as one of the most

¹Zenyaku Kogyo Co., Ltd. October 3 2005. *Jikininn*.

Table 4. Results of Experiment 2 on the DeepFashion Dataset

	Classic	Basic	Cute	Soft
Explainer w/o interaction [3]	0.9700	0.9909	0.9921	0.9920
Explainer w/ interaction (ours)	0.9700	0.9906	0.9920	0.9920
Performer	0.9699	0.9909	0.9921	0.9920

The evaluation metric is classification accuracy.

Table 5. Results of Experiment 2 on the DeepFashion Dataset

	Classic	Basic	Cute	Soft
Explainer w/o interaction [3]	9.79	9.79	9.79	9.79
Explainer w/ interaction (ours)	9.80	9.80	9.80	9.80
Performer	9.82	9.83	9.83	9.81

The evaluation metric is the conditional entropy of the prediction.

Fig. 6. An example of explanations produced by the explainer without interactions in the experiment on the DeepFashion dataset. This chart and the one below show explanation of a prediction result for “Classic.” Note that a cloth of interest in this image is a jacket.

important reasons; however, it is different from the other one in that the second most emphasized attribute is “Celebrity/Character,” which is aligned with our intuition. Although in this case the effects of interaction are much less important than in the other two experiments, our method is able to produce reasonable quantitative and semantic explanations just as the other cases.

Tables 3, 5, and 7 show that the conditional entropy is almost the same as the explainer without interaction as well as the performer. However, as pointed out in [3], this is not directly related to the ground truth of explanations. We believe that higher accuracy and correlation coefficients are more important because it means the distillation from the performer is more successful.

For more experimental results, please refer to Figures 10, 11, 12, and 13 in the appendix.

Fig. 7. An example of explanations produced by the explainer with interactions in experiments on the Deep-Fashion dataset [24]. Note that the scale between the two rows is different.

Table 6. Results of Experiment 3 on TV Ads Dataset

	F	I	W	R
Explainer w/o interaction [3]	0.631	0.552	0.724	0.653
Explainer w/ interaction (ours)	0.613	0.568	0.728	0.674
Performer	0.687	0.692	0.816	0.716

The evaluation metric is Pearson’s correlation coefficients. The columns refer to Favorability, Interest, Willingness, and Recognition.

Table 7. Results of Experiment 3 on TV Ads Dataset

	F	I	W	R
Explainer w/o interaction [3]	6.81	6.82	6.82	6.81
Explainer w/ interaction (ours)	6.82	6.82	6.81	6.80
Performer	6.82	6.81	6.81	6.82

The evaluation metric is the conditional entropy of the prediction. The columns refer to Favorability, Interest, Willingness, and Recognition.

4.5 Discussion

In the previous section, we reviewed our experimental results and argued that our method with interactions is able to produce more accurate and insightful explanations than a similar one without them. Experiments 1 and 3 demonstrated better prediction results by explainers with interactions, while Experiment 2 resulted in almost the same performances. Here, we would like to consider an aspect of the architecture in more detail. Let’s discuss the regression model used in Experiment 3 on the TV ads dataset. This model is distinguished from the other two in that a sigmoid function

Fig. 8. An example of explanations produced by the explainer without interactions in experiments on TV ads dataset. This explains the favorability of a TV ad featuring famous actors promoting popular over-the-counter medicine.

Fig. 9. An example of explanations produced by the explainer with interactions in experiments on the TV ads dataset. Note that the scale between the two rows is different.

is attached to the end of g and h , which induces weights $\mathbf{w}$ and factor vectors $\mathbf{V}$, respectively. The reason we add a sigmoid to this model is that the prediction performance drops sharply otherwise, as illustrated in Table 8. Our original intention was to allow weights and factor vectors to take negative values to provide more flexibility to the models, as in [3] and the other two experiments. However, we find that the range of the weights and vectors needs to be restricted to produce plausible explanations and at the same time maintain an acceptable level of performance. We suppose that whether an explainer needs a sigmoid or another appropriate activation function depends on the type of prediction task: when explainer models are built, the fine details of their design will depend on the problems for which they are intended.

Table 8. Performance of Our Interaction-including Model on the TV Ads Dataset with and without a Sigmoid

	F	I	W	R
Explainer w/o sigmoid	0.579	0.518	0.633	0.514
Explainer w/ sigmoid	0.613	0.568	0.728	0.674

The columns refer to Favorability, Interest, Willingness, and Recognition.

5 CONCLUSION

In this article, we have proposed a method to add explainability to an existing prediction model regardless of the type of prediction task. More specifically, our method can produce quantitative and semantic explanations that are easily interpretable. Our method developed from previous work by Chen et al. [3] that attempted to explain a prediction result by the addition of contributions from attributes, without including interactions. However, this method had a defect, in that there was a trade-off between performance and explainability. Inspired by the factorization machine, we addressed this problem by introducing feature interactions to the method. We verified the effectiveness of our proposal through experiments on multiple datasets with multiple prediction problems. We conducted qualitative and quantitative evaluations of the explainer of our study and showed it superior to that of the no-interactions model.

In future work, the interactions not only between attributes in the same domain but also between attributes in different domains may be considered to acquire more insightful explanations.

APPENDIX

A RESULTS FROM EXPERIMENT 1

Fig. 10. Example of explanations produced by the explainer without interactions in experiments on the CelebA dataset. This and the one below explain a prediction result for “Attractive.”

Fig. 11. Example of explanations produced by the explainer with interactions in experiments on the CelebA dataset [25]. Note that the scale between the two rows is different.

Fig. 12. Another example of explanations produced by the explainer without interactions in experiments on the CelebA dataset [25]. This and the one below explain a prediction result for “Attractive.”

Fig. 13. Example of explanations produced by the explainer with interactions in experiments on the CelebA dataset corresponding to Figure 12. Note that the scale between the two rows is different.

REFERENCES

- [1] Sercan O. Arik and Tomas Pfister. 2020. ProtoAttend: Attention-based prototypical learning. *Journal of Machine Learning Research* 21, 210 (2020), 1–35.
- [2] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. 2019. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, Vol. 32. 8930–8941.
- [3] Runjin Chen, Hao Chen, Jie Ren, Ge Huang, and Quanshi Zhang. 2019. Explaining neural networks semantically and quantitatively. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.
- [4] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 29. 2172–2180.
- [5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ipsir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide and deep learning for recommender systems. In *Proceedings of the Workshop on Deep Learning for Recommender Systems*. 7–10.
- [6] Edward Choi, Mohammad Taha Bahadori, Joshua A. Kulas, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Proceedings of the International Conference on Neural Information Processing Systems*. 3512–3520.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.
- [8] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing higher-layer features of a deep network. *Technical Report*, Université de Montréal.
- [9] Jerome H. Friedman and Bogdan E. Popescu. 2008. Predictive learning via rule ensembles. *Annals of Applied Statistics* 2, 3 (2008), 916–954.
- [10] Nicholas Frosst and Geoffrey E. Hinton. 2017. Distilling a Neural Network Into a Soft Decision Tree. arXiv:1711.09784.
- [11] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1725–1731.
- [12] Michael Harradon, Jeff Druce, and Brian Ruttenberg. 2018. Causal Learning and Explanation of Deep Neural Networks via Autoencoded Activations. arXiv:1802.00541.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- [14] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. 2018. Grounding visual explanations. In *Proceedings of the European Conference on Computer Vision*.
- [15] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations*.
- [16] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. 2019. A survey of deep learning-based object detection. *IEEE Access* 7 (2019), 128837–128868.
- [17] Atsushi Kanehira and Tatsuya Harada. 2019. Learning to explain with complementary examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [18] Atsushi Kanehira, Kentaro Takemoto, Sho Inayoshi, and Tatsuya Harada. 2019. Multimodal explanations by predicting counterfactuality in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [19] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. 2018. Textual explanations for self-driving vehicles. In *Proceedings of the European Conference on Computer Vision*.
- [20] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 426–434.
- [21] Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Understanding Neural Networks through Representation Erasure. arXiv:1612.08220.
- [22] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [23] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. XDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1754–1763.
- [24] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.
- [25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*. 3730–3738.
- [26] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. 2013. Accurate intelligible models with pairwise interactions. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 623–631.
- [27] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [28] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. 2019. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 903–913.
- [29] Daniel W. Otter, Julian R. Medina, and Jugul K. Kalita. 2021. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems* 32, 2 (2021), 604–624.
- [30] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. 2019. Recent progress on generative adversarial networks (GANs): A survey. *IEEE Access* 7 (2019), 36322–36333.
- [31] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [32] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems* 37, 1 (2018), Article 5, 35 pages.
- [33] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the IEEE International Conference on Data Mining*. 995–1000.
- [34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144.
- [35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [36] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.
- [37] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning*, Vol. 70. 3145–3153.
- [38] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*.

- [39] Chandan Singh, W. James Murdoch, and Bin Yu. 2019. Hierarchical interpretations for neural network predictions. In *Proceedings of the International Conference on Learning Representations*.
- [40] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. 1161–1170.
- [41] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. 2004. Maximum-margin matrix factorization. In *Proceedings of the International Conference on Neural Information Processing Systems*. 1329–1336.
- [42] Austin Stone, Hua-Yan Wang, Michael Stark, Yi Liu, D. Scott Phoenix, and Dileep George. 2017. Teaching compositionality to CNNs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5058–5067.
- [43] Li Tao, Xueting Wang, Tatsuya Kawahara, and Toshihiko Yamasaki. 2020. Television advertisement analysis using attention-based multimodal network. In *Proceedings of the Annual Conference of JSAL*. 1H4OS12b01–1H4OS12b01.
- [44] Michael Tsang, Dehua Cheng, Hanpeng Liu, Xue Feng, Eric Zhou, and Yan Liu. 2020. Feature interaction interpretability: A case for explaining ad-recommendation systems via neural interaction detection. In *Proceedings of the International Conference on Learning Representations*.
- [45] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep and cross network for ad click predictions. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Article 12, 7 pages.
- [46] Zhihao Wang, Jian Chen, and Steven C. H. Hoi. 2020. Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2020), 3365–3387.
- [47] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. 2020. Explainable Deep Learning: A Field Guide for the Uninitiated. arXiv:2004.14545.
- [48] Matthew Zeiler, Dilip Krishnan, Graham Taylor, and Robert Fergus. 2010. Deconvolutional networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2528–2535.
- [49] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*. 818–833.
- [50] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6720–6731.
- [51] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpreting CNN knowledge via an explanatory graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [52] Q. Zhang, Y. N. Wu, and S. Zhu. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8827–8836.
- [53] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *Advances in Information Retrieval*. 45–57.
- [54] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene CNNs. In *Proceedings of the International Conference on Learning Representations*.
- [55] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. In *Proceedings of the International Conference on Learning Representations*.

Received December 2020; revised May 2021; accepted July 2021