Explaining Representation by Mutual Information

Lifeng Gu

Abstract
Science is used to discover the law of world. Machine learning can be used to discover the law of data. In recent years, there are more and more research about interpretability in machine learning community. We hope the machine learning methods are safe, interpretable, and they can help us to find meaningful pattern in data. In this paper, we focus on interpretability of deep representation.

We propose a interpretable method of representation based on mutual information, which summarizes the interpretation of representation into three types of information between input data and representation. We further proposed MI-LR module, which can be inserted into the model to estimate the amount of information to explain the model’s representation. Finally, we verify the method through the visualization of the prototype network.

1. Introduction
The field of representation learning is developing rapidly (Chen et al., 2020; Grill et al., 2020), but some basic questions such as what is a good representation and what is a general representation are still difficult to answer. Recently, the information bottleneck method has been successfully used to explain neural network’s decision (Schulz et al., 2020), which inspired us to use mutual information (Tschannen et al., 2019) to explain and analyze representation learned by neural network. How to interpret and analyze the representation in the neural network, we believe that there should be at least three steps: determining what is encoded in the representation, determining which content in the representation is useful or decision-related, and determining which content is redundant or decision-independent.

2. Related works
Interpreting global representation The popular global interpretable methods are roughly divided into two categories: one is used to visualize convolutional networks, and the other is used to learn the mapping of network decisions/outputs to inputs. Although the second method does not directly explain the representation, we feels that the second category is more similar to the goal of this paper, and require a more precise interpretation effect than the first category. We briefly describes the second method: Gradient Maps (Baehrens et al., 2010) and Saliency Maps (Simonyan et al., 2013) are used to calculate the gradient from output neurons to input features. These two methods are very simple and natural, but they cannot be used to explain the decision, they can only explain what is in the neural network, and the value of the gradient is unstable, there will be a lot of noise. Integrated Gradient (??) averaging the gradients of multiple inputs, the problem of unstable and noisy gradients of the former is solved. There are many related works such as LRP (Bach et al., 2015a), DTD (Montavon et al., 2017), (Selvaraju et al., 2017) passed output the derivative of the middle feature layer to calculate the weight, and use the weight to assume the activation value to explain it. They also combined GuidedBP to propose GuidedGrad-CAM, which will bring more stable interpretation results.

Interpreting each element of representation Disentangled representation learning want to interpret each factor of representation. They assume that the data changes from independent factors, such as the direction of the image, the intensity of light, and the length of human hair. While changing several factors, other factors may not be affected. They assume that the disentangled representation can easily handle downstream tasks, e.g., when solving gender classification, the learner can make judgments based on the "gender" dimension of the representation. A commonly used method of learning disentangled representations is applied in the implicit representation of vae $z \sim q(z|x)$ constraints, constraints are generally added to the posterior approximation $q$, the most common objective function can be expressed: $L_{VAE} + \lambda_1(R_1(q(z|x))) + \lambda_2(R_2(q(z)))$, where $R_1$, $R_2$ are regularization items, $\lambda_1, \lambda_2 > 0$ is the weight. Higgins et al. (2016) proposes to increase the weight of the kl divergence term $D_{kl}((z|x)||p(z))$ in vae, so that vae can learn the representation with the prior distribution: the specified independent distribution of each dimension, this paradigm has led to many subsequent methods that want to explain the element-level representation. Kim & Mnih (2018) adds a measure to offset the impact after the objective function of vae. For the full correlation term of the variable dependency,
the objective function is $L_{VAE} + \lambda_2(TC(q(z)))$. The second term is full correlation, which requires density ratio techniques to estimate.

3. Method

3.1. Mutual information estimation

The representation in the neural network encodes the input information. To determine the content contained in the representation, this article can use mutual information theory to calculate the mutual information between the representation $f(x)$ and the input $x$, $I(x, f(x))$, which will tell us how much information about the input is encoded in the representation, but simply knowing $I(x, f(x))$ is not enough, and it’s interpretability is poor. For interpretable problems, we generally hope to find a mapping between the interpreted object and the input. For this purpose, we need to calculate mutual information between the local content in the input and the representation $f(x)$, that is to say, we need to know which part of the information in the input data is specifically encoded in the representation. We formalize the amount of information as:

$$I(x_i, f(x)) = E_{x_i}(KL(p(f(x)|x_i)||p(f(x))))$$

We cannot directly calculate the marginal distribution $p(x_i), p(f(x))$ and the conditional probability $p(x_i|f(x))$, conditional probability $p(f(x)|x_i)$ in the formula (2), these distributions cannot be calculated directly, we use popular information estimation method InforNCE(Tschannen et al., 2019) to estimate $I(x_i, f(x))$, let $z = f(x)$, then the mutual information between local input and representation is:

$$I(x_i, f(x)) = E_{x_i}(KL(p(x_i|f(x))||p(x_i)))$$

We need to use the method of estimating the upper bound of mutual information to optimize the formula 7. At present, there is no reliable method of estimating the upper bound of mutual information. This article can only select the experimentally reliable method The VIB method (Alemi et al., 2016) is used to optimize the formula (7). In this paper, the variational distribution $q$ is used to replace the marginal distribution $p$ in the formula (??), then we have:

$$I(x_i, z) \leq E_{x_i}(KL(p(z|x_i)||q(z)))$$

3.2. Information bottleneck

We now need to extract useful information or decision-related information in the representation. As long as the representation contains this information, the decision of the model will not be affected. To extract useful information or decision-related information, naturally, this article needs use the information bottleneck principle (Tishby & Zaslavsky, 2015), let $y$ represent the label, the information bottleneck principle is a learning principle to obtain the minimum amount of information, and the general formal description is to maximize the mutual characterization of $y$ and label $y$. At the same time of information, it reduces the mutual information between the characterization $z$ and the input $x$. Its usual optimization goals are:

$$\max I(z, y) \quad \text{s.t.} \quad I(x, z) \leq c$$

After introducing the Lagrange multiplier $\beta$, we have:

$$\max I(z, y) - \beta I(x, z)$$

We follow the approach of (Schulz et al., 2020), only optimize the second item of the formula (6), and replace the first item of the formula (6) with the objective function of the original model. This is intuitive approximation, in the learning process of the neural network, the label information is continuously encoded in the representation. In order to extract the useful part of the representation in a certain model, we need to optimize:

$$L(x) = \max l(x) - \beta \sum_{i=1}^{N} I(x_i, z)$$

Where $l(x)$ is the original objective function of the model. Similar to the previous section, this article cannot directly optimize the formula (6). Its interpretability is not good. We need to specifically extract useful information from the local input. We optimize:

$$L(x) = \max l(x) - \beta \sum_{i=1}^{N} I(x_i, z)$$

3.3. Information redundancy

We now need to identify the redundant information in the representation. The redundant information is a kind of information that is irrelevant to the decision. Removing it will
We use the Informax-estimator module in the figure 1 to estimate the mutual information in the formula (3), which can be implemented simply by a single-layer or two-layer mlp, which will participate in the calculation of the local features converting to the same dimension as the global representation, and then calculate the lower bound of the mutual information between the two according to the formula (3). As shown in 1, after inserting the MI-LR module, the local feature $x_{ij}$ and the representation $z$ will be sent to the Informax-estimator module, and then follow the formula 3 to obtain the mutual information between local features and representation, and use it as a part of the objective function (10).

### 3.4. MI-LR module

In order to estimate the above three kinds of information in a specific model to analyze representation, we need to combine them with the original objective function of the model and then learn to get it. We add the formula (3) and the formula (8) in the original objective function, the total objective function is:

$$L(x) = \max l(x) + \alpha E_{x_i}(\sum_{j=1}^{K} \log \frac{e^{f(x_i', z_j)}}{\sum_{j=1}^{K} e^{f(x_i', z_j)}}) - \beta E_{x_j}(KL(p(z|x_i)||q(z)))$$

(10)

where $\alpha$ and $\beta$ are hyperparameters. In a specific model, according to (10), we can get the three kinds of information we want to explain the representation after optimizing.

#### 3.4.1. INFORMATION ESTIMATE

We use the Informax-estimator module in the figure 1 to estimate the mutual information in the formula (3), which can be implemented simply by a single-layer or two-layer mlp, not affect the decision of the model. We first calculate the mutual information between the input and the representation $I(x, z)$, and then calculate the useful information/decision-related information $I(x, z')$ in the representation. Finally, we use the difference between the two to represent the amount of redundant information in the representation. As before, we specifically calculate the amount of redundant information in the local input:

$$R = \sum_{i}^{N} I(x_i, z) - I(x_i, z')$$

(9)

In order to find three kinds of information in a specific model to analyze representation, we need to combine them with the original objective function of the model and then learn to get it. We add the formula (3) and the formula (8) in the original objective function, the total objective function is:

$$L(x) = \max l(x) + \alpha E_{x_i}(\sum_{j=1}^{K} \log \frac{e^{f(x_i', z_j)}}{\sum_{j=1}^{K} e^{f(x_i', z_j)}}) - \beta E_{x_j}(KL(p(z|x_i)||q(z)))$$

(10)

where $\alpha$ and $\beta$ are hyperparameters. In a specific model, according to (10), we can get the three kinds of information we want to explain the representation after optimizing.

#### 3.4.2. INFORMATION BOTTLENECK

We use the Infor-bottleneck module and mask module in the figure 1 to realize the information bottleneck.

The Infor-bottleneck module can be simply implemented by a single-layer or two-layer mlp, which converts the local features and representation involved in the calculation to the same dimension, and then calculates the upper bound of the mutual information between the two according to the formula 8. The mask module is used to assist in the realization of the information bottleneck function. Like (Schulz et al., 2020), it consists of several convolution layer whose kernel size is 1. After inputting local features, the mask $\alpha$ can be automatically inferred, whose dimension will be the same as the input local features, the mask $\alpha$ will occludes local features.

How to reduce the mutual information between local features and representation and realize the information bottleneck function? We refer to and modify the implementation in (Schulz et al., 2020). In the first stage, we first make the input $X_i$ get the global representation $z_i$ through the original model, and then collect the features of each layer in the network. In the first stage, we first let the input $X_i$ get the global representation $z_i$ through the original model, and then collect the features of each layer in the network. In the second stage, we insert the Infor-bottleneck module and the mask module into the original model, pass $X_i$ through the original model again to obtain local features $x_i$, and then use the features collected in the previous stage, we send it to the mask module to get the mask $\alpha$, $\alpha \in [0, 1]$, the dimension of $\alpha_{ij}$ is the same as the local feature $x_{ij}$, and then $x_{ij}$ is multiplied by $\alpha$ to get the occluded local feature $x_{ij}$, and finally $x_i$ and $z_i$ are sent to the calculation formula 8 in the Infor-bottleneck module to obtain useful information/decision-related information in the representation, and also participate in the optimization as a part of the objective function 10.

It can be imagined: $\alpha$ is automatically learned by the mask module. When $\alpha$ is all 1, the original value is retained between $x_i'$ and $z_i$. The amount of information, namely: $I(x_i, z_i') = I(x_i, z_i)$, when $\alpha$ is all 0, $x_i'$ here is no more mutual information between $x_i'$ and $z_i$. 

...
Explaining Representation by Mutual Information

Under normal circumstances, part of the value of $\alpha$ will be approximately equal to 0, which reduces the mutual information between local features and representation. It means we realize the information bottleneck principle and extract useful information/decision-related information in the representation.

3.4.3. INFORMATION REDUNDANCY

In the previous step, we got the original local feature $x_i$, the occluded local feature $x'_i$, and the representation $z_i$, we send them to the Informax-estimator module, and use the formula 9 to calculate the amount of redundant information. The amount of redundant information does not need to be added to the objective function to participate in optimization.

4. Experiment

Different from other interpretability methods (Baehrens et al., 2010; Bach et al., 2015b; Schulz et al., 2020), we focus on analyzing and explaining the representation of the model. We can use three kinds of information to get profound interpretation results. We use a typical prototype network as an example to illustrate how to explain and analyze the representation of the model using the methods we proposed.

4.1. Prototype Network Visualization

The prototype network is a typical few-shot learning method. The typical learning task of few-shot learning is as follows: Given a query image and a support set composed of multiple types of images, few-shot learning method is to assign the query image to the support which is concentrated in the appropriate category. According to the number of samples in each category and the number of categories in the query image and support set, few-shot learning will be divided differently, such as: 5way1shot, 5way5shot. The prototype network uses the prototype concept for assignment. For a good visualization effect, we will set the learning way is 5way1shot, that is: there will be 5 types of samples in the support set, and each type of sample will have 1 image, and we also specify 5 query samples: Selecting 1 picture as the query sample from each category in the support set. The figure 2 is an example. The support/query above each column represents whether the 5 samples in the column belong to the query sample or the support set.

We first train the prototype network, after training the prototype network, we fix its parameters to prevent subsequent changes, and then insert the MI-LR module into the layer1 block and avgpool layer of the residual network as the feature extractor, layer1 block’s output of is used as a local feature, and the output of avgpool is used as a representation. The MI-LR module will learn the parameters and...
calculate the amount of information in the way shown in
the figure 1. Finally, we will get three kinds of information
between the output of the layer1 block and the avgpool
layer. We visualize these three kinds of information and ex-
plain representation: the output of avgpool layer. The three
types of information can tell us what content is encoded
in the representation of the query sample and the support
set sample when the prototype network is learning, so that
we can know how the prototype network is distinguished.
We will visualize the original image first, and then visualize
the heat map of the three output information and the mixed
image of the original image and the heat map.

4.1.1. Dataset
For a good visualization effect, we use the bird dataset
CUB-200-2011 (Wah et al., 2011). It is more suitable for
visualization than general image data sets.

4.1.2. Mutual Information Visualization
The picture 2 is the original image, the picture 3 is the heat
map of mutual information, and the picture 4 is the mixed
picture of the heat map and the original picture. According
to the previous sections: the mutual information between
the representation and the local input represents how much
input information the representation contains. The red part
of the heat map is the part with high information. From the
heat map 3, when the prototype network is performing a
few-shot learning task, the avgpool layer of the prototype
network almost contains all the input 2 except for the back-
ground also contains noise such as branches. Different parts
have different amounts of information, and the amount of
information on the chest of a bird is low.

This is similar to general intuition. The avgpool layer has
a higher dimension and a large capacity enough to encode
most of the image in the CUB dataset. Combined with the
original image, we can see a better interpretation effect from
the mixed image 4.

4.1.3. Visualization of decision-related
information
Figure 5 is a mixed image of the original image and the
heat map of decision-related information. According to
the previous sections: the decision-related information be-
tween the representation and the input represents the input
information contained in the representation that makes the
decision sufficient. The activation area of figure 5 is very
small, which shows which part of the input coded in the
representation is the most important for decision-making.
We can see: the bird’s mouth, the unique feathers on the
body When the prototype network performs the discrimina-
tion task, these areas are the main basis for the prototype
network. Unlike the InforNCE estimation method, the VIB
method estimates a large degree of dispersion of informa-
tion, and it is very suitable for estimating decision-related
information, because we need as few activation regions as
possible to determine the most important input information
contained in the representation, but unfortunately the VIB
method is very inaccurate. Although its visualization effect
is very good, the VIB method needs to specify the con-
firmed variational distribution, which affects it’s estimation
accuracy.

4.1.4. Visualization of redundant information
The picture 6 is a mixed picture of the heat map of the re-
dundant information between the output of the layer1 block
and the output of the avgpool layer and the original im-
age. According to the previous sections: the redundant
information in the representation refers to the input infor-
mation encoding in the representation which is redundant
for decision-making, removing it does not affect the decision-
making. Our visualization effect is not ideal, but we can still
find that the active area of redundant information is mainly
the tail and back area of the bird. These areas have no effect
on distinguishing the type of bird.

5. Conclusion
We propose a representation interpretation method based on
mutual information, which summarizes the interpretability
of the representation into three types of information between
the local input data and the representation, and further pro-
poses the MI-LR module, which can be inserted into the
model to calculate the amount of information to explain
the representation of the model. Finally, the interpretation
ability is visually demonstrated through the visualization of
the prototype network.

References
Aleimi, A. A., Fischer, I., Dillon, J. V., and Murphy, K.
Deep variational information bottleneck. arXiv preprint

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

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

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M.,
Hansen, K., and Müller, K.-R. How to explain individual
classification decisions. The Journal of Machine Learning
Figure 2. original image
Figure 3. Heat map of mutual information
Figure 4. mixed figure between heat map and original figure of mutual information
Figure 5. mixed figure between heat map and original figure of decision-related information
Figure 6. mided image between heat map and original image of redundant information
Explaining Representation by Mutual Information


