

Towards Learning a Vocabulary of Visual Concepts and Operators using Deep Neural Networks

Sunil Kumar Vengalil
sunilkumar.vengalil@iiitb.ac.in

Neelam Sinha
neelam.sinha@iiitb.ac.in

International Institute of Information
Technology
Bangalore
India

Abstract

Deep neural networks have become the default choice for many applications like image and video recognition, segmentation and other image and video related tasks. However, a critical challenge with these models is the lack of explainability. This requirement of generating explainable predictions has motivated the research community to perform various analysis on trained models. In this study, we analyze the learned feature maps of trained models using MNIST images for achieving more explainable predictions. Our study is focused on deriving a set of primitive elements, here called “visual concepts”, that can be used to generate any arbitrary sample from the data generating distribution. We derive the primitive elements from the feature maps learned by the model. We illustrate the idea by generating visual concepts from a Variational Autoencoder trained using MNIST images. We augment the training data of MNIST dataset by adding about 60,000 new images generated with visual concepts chosen at random. With this we were able to reduce the reconstruction loss (mean square error) from an initial value of 120 without augmentation to 60 with augmentation. Our approach is a first step towards the final goal of achieving trained deep neural network models whose predictions, features in hidden layers and the learned filters can be well explained. Such a model when deployed in production can easily be modified to adapt to new data, whereas existing deep learning models need a re-training or fine tuning. This process again needs a huge number of data samples that are not easy to generate unless the model has good explainability.

1 Introduction

Ever since the break through success of Alexnet [4] for image classification, deep neural networks has been explored and successfully being used for many complex tasks not only in domains like vision, speech and natural language processing for which they were originally invented for, but also for other domains in industry. However, one of the major pain points of these models, when used in industry, is the lack of explainability, i.e., they are unable to provide the exact reason for prediction. One the one hand, researchers build complex ensemble and deep architectures in order to increase the classification accuracy, which results in increased accuracy but the more complex the model is, the more difficult to provide explanations for the predictions.

In this study, we propose a novel method for augmenting the training set and introducing losses in hidden layers during training so that the predictions, and the hidden layer feature maps that lead to the final predictions, are more explainable. Many existing algorithms [8] [9] [10] for building explainability into deep learning models are applied after the model is fully trained. However, our approach differs significantly as we are modifying the loss function and training data for incorporating explainability. In addition to solving the primary task of explainability, our approach also benefits from the fact that the model can be trained with a smaller number of epochs. This is because all the hidden layers, where the losses are introduced, start learning the features in parallel, whereas in existing architectures training losses are computed only at the output layer and it takes a larger number of epochs for the initial layers to learn useful features. Further, since the loss functions can be introduced at any layer directly, the approach will not suffer from vanishing gradient.

Even though our approach can be applied to any deep learning models, we pick a widely used generative model, Variational Autoencoder [11] in order to illustrate our approach. Our work is motivated by the concept of learning based on visual concepts introduced by Lake et.al. in 2015 [12]. We generate visual concepts, similar to those used in [13] and use the images of generated concepts for training a variational autoencoder, whereas in [14] the concepts are used to build a probabilistic prediction framework.

Major contributions of this paper are:

1. Propose a mechanism for data augmentations so that the trained model has better explainability at each layer and feature maps in hidden layers.
2. Propose a mechanism to introduce losses at hidden layers in a hierarchical fashion starting from primitive visual concepts and building up more complex images by applying some predefined operations on these concepts.

2 Related Work

Adding explainability to deep neural networks has been one of the key focus areas of the deep learning research community in the last decade. Explainability techniques can widely be classified as model-agnostic (which can be applied to any model) or model-dependent where the approach depends on the specific prediction algorithms used by the model. The early works on this includes the use of explanation vectors introduced by David Baehrens et.al. in their paper [15]. Local Interpretable Model-Agnostic Explanations (LIME) [16] is one of the most widely used tools to explain the predictions of any machine learning model. LIME works by approximating the model behaviour locally by a simple interpretable surrogate model like linear model or decision tree. The approximation is restricted to a small local change in the input. As this technique looks only at the input and output of the model, this technique can be applied to any machine learning models. However, the main drawback of the approach is that the surrogate model is only a local approximation and fails to capture the global behaviour. Rabold et. al. in their recent work [17] builds on top of LIME and suggests an approach where classifiers' decisions can be explained in terms of logic rules. Their approach is unique in the sense that the predictions can be explained using relationships of objects within the image, whereas in most other approaches the decision is based on presence or absence of certain features.

Another famous technique called Layer-wise Relevance Propagation was introduced by Bach et.al. [18] where the classifier decision is back propagated and a relevance score is

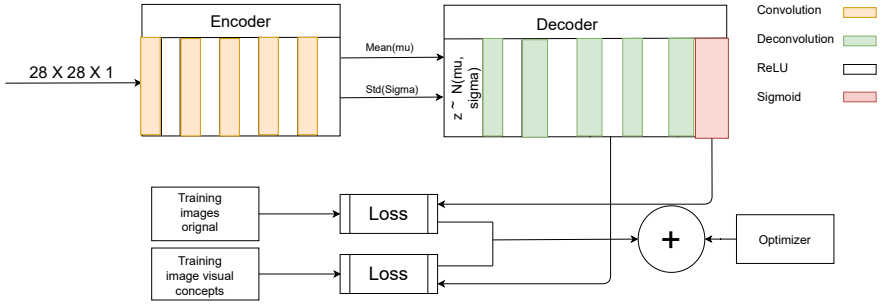


Figure 1: Architecture of the proposed model. We add an additional loss component at layer 3 of decoder in Variational Autoencoder.

computed at each of the layers backwards until the input units are assigned a relevance score. The work by Wojciech Samek et.al. [10] provides a quantitative evaluation of Layer-wise Relevance Propagation and suggests it as a better model as opposed to sensitivity based methods.

Scott et.al. introduced a unified approach for explainability which combines multiple methods and brings in a new class of additive feature importance measure (called SHapley) [6]. Their method, known as SHapley Additive ExPlanations (SHAP) considers all the interactions between features and provides an average measure of importance for each feature.

Another class of widely used algorithms for Convolutional Neural Networks targets at finding and highlighting the regions in the input image that is responsible for the prediction. These techniques, originally introduced in paper [11], generate a heatmap called Class Activation Map that will highlight the most sensitive regions in the input image. Saliency Map, introduced by Symonyan et.al. in 2014 [12] is another variant of class activation map computed by optimizing the input image using the gradient of output prediction with respect to the input image.

3 Proposed Method

3.1 Network Architecture

We use a variational autoencoder (VAE) with five layers of convolutional layers in the encoder and five layers of deconvolution layers in the decoder as shown in Figure 1. All layers, including the middle layer that generates the latent images, are fully convolutional. We replaced the middle dense layer of in VAE, with a convolutional layer in order to create a latent image (as opposed to a one dimensional latent vector) that preserves the spatial relationship. We added an additional loss component at layer 3 of decoder in order to force this layer to learn the primitive visual concepts of the dataset. The final layer of decoder has sigmoid activation function and we use binary cross entropy loss as the MNIST images can be treated as binary image.

3 2 5	5 7 0
6 1 0	7 7 1
8 3 8 0 0	9 0 1
4 - 1 1	4 4 1
2 2 -	2 7 0

Figure 2: Visual concepts generated from each of the digits in MNIST. Since there are two different types of images for digits 4 and 2 separate concepts were generated for each image. The width, height and location of these segments were used to define the mean value of a Normal distribution used for generating more samples.

3.2 Dataset

We conducted experiments on MNIST dataset by training the VAE with and without concept loss added. For adding the concept loss, the MNIST training set was first augmented by adding 3000 training images for each of the 18 concepts shown in Figure 2. Some of the similar looking concepts in Figure 2, like the horizontal line segments and vertical line segments, were combined into one group to form 18 unique visual concepts. The steps for generating concepts are detailed in the sections below.

3.3 Generating Visual Concepts

We augment the training set using images of randomly generated segments from MNIST images. For each of the visual concepts shown in Figure 2, 3000 images were generated by randomly sampling the height, width and location of the segment from a normal distribution. For each concept, the mean value of the height, width and location is kept the same as in Figure 2. The number of visual concepts and their corresponding distributions are hyper parameters that will vary from dataset to dataset. To generate the segments for each concept, we sampled heights, widths and location of top left corner (represented by two distributions for x and y coordinates) from the respective concept’s distributions and generated slices from randomly chosen training images. Standard deviation for all distributions were kept as 1 pixel in our experiments.

3.4 Concept Loss

We introduce a loss, which we call “concept loss” per decoder layer and per feature map in the selected hidden layers of the decoder of variational autoencoders (VAE). Sample visual concepts derived from the training images are used as ground truth for computing the loss for this layer. For our experiments, we used mean squared error between the feature map and concept ground truth as the loss component. In our case, the encoder and decoder had 5 hidden layers and the concept loss was applied at the third hidden layer as this is closer to the output layer. We leave the fourth layer of decoder to learn new derived images from the primitive visual concepts introduced at the third layer. Mean squared error between original training images and the predicted images is used as loss at the output layer as in usual VAE.

3.5 Generating Training Samples

The below steps were followed to generate additional random samples for each of the visual concept in Figure 2

1. Create multiple clusters with representative images for each class in the training dataset using below steps
 - (a) Cluster the latent images, learned by VAE, of all training images into 10 clusters.
 - (b) Decode the cluster cluster center images using the decoder of VAE. The decoded images are shown in Figure 3. This forms a set of 10 representative images one for each digit in MNIST dataset.
2. Choose a random image, either from the training set (with probability p) or from the cluster center image (with probability $1 - p$). The parameter p can be adjusted to control the diversity of samples generated for each visual concept. A value of p closer 1 will generate visual concepts from individual training samples most of the time and hence will create a dataset with large variance. On the other hand, a value close to zero will generate most of the concepts from cluster center images and hence will have less variance. In our experiments, we used $p = 0.8$.
3. Get the value of height, width and location of the segment by sampling from the respective distributions.
4. Take a segment from the selected image of the digit with the height, width and location.

Each generated sample is given a label based on the visual concept from which it was generated.

The final training set is generated by combining the original 60,000 training images in MNIST dataset and the 54,000 (300 for each of the 18 concepts) generated concepts to form a training set with 114,000 images. The labels for the original training samples were kept the same (0-9) as in MNIST dataset and additional 18 labels (10-27) were used for each of the visual concepts. We used 70% of these samples for training and the remaining 30% were used for validation.

4 Results and Discussions

We trained the network for 50 epochs and performed the following clustering operations on the latent vectors and decoder layer 3 feature maps.

1. Cluster the latent vectors corresponding to all the training images with label 0-9 (i.e., the training samples for the original MNIST images) into 10 clusters.
2. Cluster the latent vectors corresponding to all the training images with label 10-27 (i.e., the training samples for the generated visual concept) into 18 clusters.
3. Cluster the layer 3 feature maps with concept loss added into 18 clusters.
4. Cluster the layer 3 feature maps without concept loss into 18 clusters.

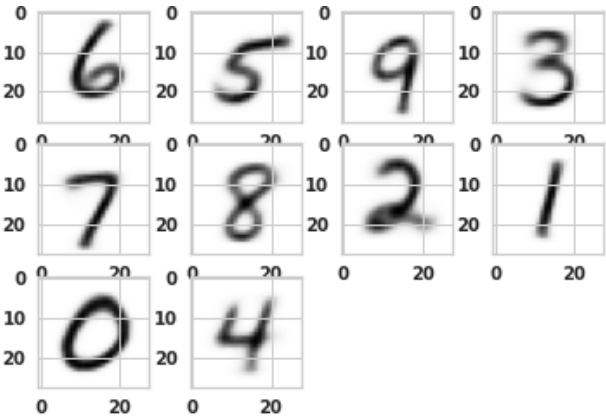


Figure 3: Decoded Cluster center images of latent vectors corresponding to images from original MNIST classes

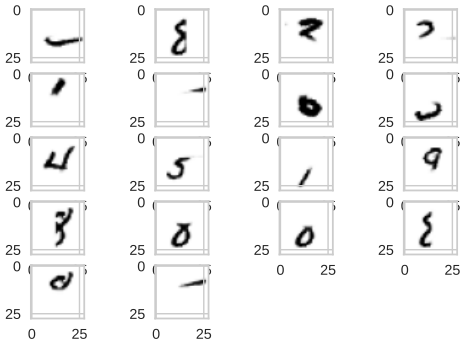


Figure 4: Cluster centers of the images corresponding to visual concepts. The images were obtained after clustering latent vectors corresponding to images of visual concepts and then decoding the cluster centers.

Cluster centers of latent images were decoded using the decoder of the VAE. Figure 3 shows the cluster center images corresponding to original images present in the MNIST training set. As is evident from the figure each cluster corresponds to one of the MNIST class. The average reconstruction loss for the validation samples reduced from 120 to 60 with the augmented data and added reconstruction loss.

Figure 4 shows the cluster centers for the images corresponding to the visual concepts. As evident from the figure, VAE is able to reconstruct all the generated visual concepts.

Figure 5 and Figure 6 shows the comparison of feature maps in layer 3 with and without concept loss added. As evident from this figure, when the concept loss is added the feature maps in layer 3 is able to learn to reproduce the visual concepts.

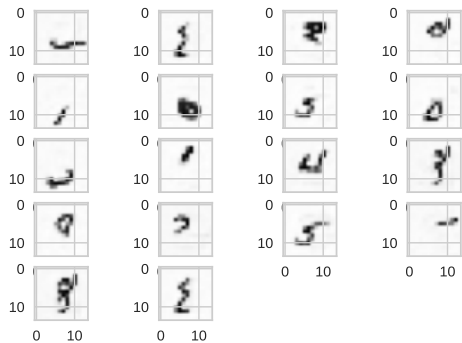


Figure 5: Cluster center images of layer 3 feature maps with concept loss added

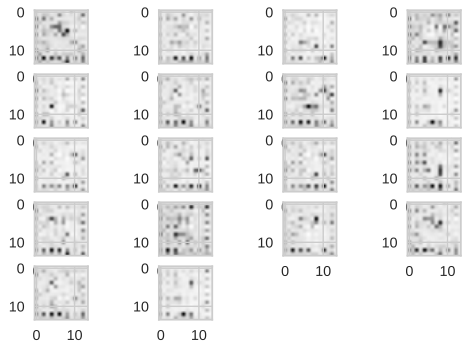


Figure 6: Cluster center images of layer 3 feature maps without concept loss

5 Conclusion

In this work, we propose a novel method to augment the training data and add an additional loss component in the hidden layer for deep neural networks so that the features in the trained model, and hence the final prediction, is more explainable. We generate random images of primitive visual concepts, with a predefined distribution, and augment the training set with these images. An additional loss component is added for the generated samples and the loss is directly computed on the feature maps of the hidden layer. We verified our approach by training a Variational Autoencoder on MNIST images. We analyzed the feature maps of hidden layers and our analysis shows that the network is able to learn more explainable and semantically meaningful features when the concept loss is added. Further, we see significant reduction in reconstruction loss from 120, without using concept loss, to a value of 60 with concept loss added. Our approach is not restricted to VAE and can be applied to any other deep learning model used for tasks like classification and segmentation.

References

- [1] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.
- [2] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.
- [3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [5] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- [6] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.
- [7] Johannes Rabold, Hannah Deininger, Michael Siebers, and Ute Schmid. Enriching visual with verbal explanations for relational concepts—combining lime with aleph. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 180–192. Springer, 2019.
- [8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

-
- [9] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.
 - [10] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. 2014.
 - [11] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.