Interpreting variational autoencoders using Shapley additive explanations

Prashansa Singh with Supervisor Heejung Shim

Introduction

A Variational Autoencoder (VAE) is a black-box machine learning (ML) model. VAEs take in data x as input, learn its lower-dimensional representation z and output a reconstruction of the data $\hat{x}$ [1].

However, it is difficult to understand exactly how the input is converted into the output (hence the term "black-box" model). This project explores how Shapley Additive Explanations (SHAP) can be used to understand the relationship between the input and output, thereby making VAEs more interpretable.

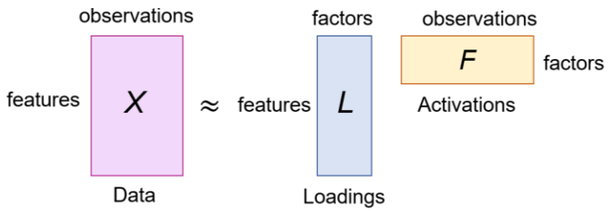

Fig. 1: Linear dimensionality reduction method

What is a VAE?

VAEs take in high-dimensional data x as input, which can be expressed as a matrix X of (high-dimensional) features and their observations. Then, VAEs compress the data by learning a lower-dimensional 'latent' representation z of the data. This is the 'Activations' matrix F of (lower-dimensional) factors and observations.

There are two perspectives on understanding VAEs; (1) the neural network perspective and (2) the probability model perspective.

(1) VAE: Neural Network Perspective

A VAE consists of an encoder and decoder (which are both neural networks) as shown in Figure 2.

A neural network contains many layers of connected nodes which process data through a series of algorithms to recognise relationships in datasets.

The input to the encoder is the input data. In the hidden layers of the neural network, the data is encoded into a latent (hidden) representation space z. The decoder receives this latent variable as input and outputs a reconstruction of the input data, $\hat{x}$.

(2) VAE: Probability Model Perspective

A VAE consists of the probability distributions of data x and latent variable z. The data is drawn from the likelihood $p(x|z)$ and the latent variables are drawn from a prior $p(z)$.

The aim is to infer values of the latent variables given the observed data by finding an approximation of the posterior $p(z|x)$.

After inputting data x into an inference network (encoder), the VAE finds $E[z]$ and $V[z]$, where z follows a Gaussian distribution with this mean and variance. Then, we sample z from this distribution and pass it into a generative network (decoder) to generate the reconstruction $\hat{x}$.

VAE Visualisation

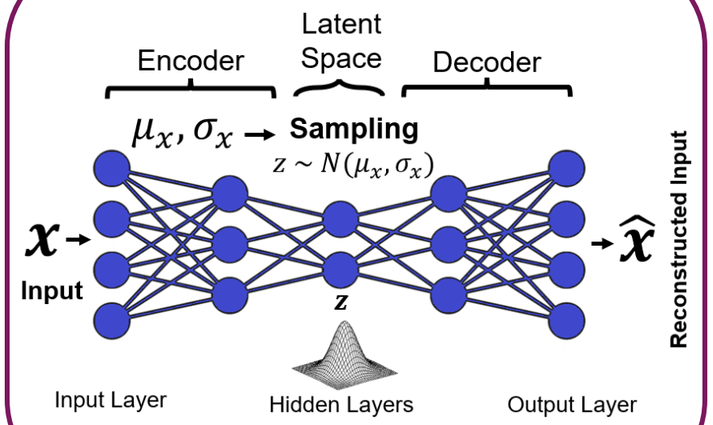

Fig. 2: Visualisation of VAE [2]

Aim of Project: Make VAEs Interpretable

We want to determine how these latent variables (factors) are related to the features and find a matrix similar to the 'Loadings' matrix *L*.

However, due to the black-box nature of VAEs, it is difficult to find out how the input (features) and learned latent variables (which determine the output) are related. This is where we propose SHAP can be used to quantify this relationship, and thereby extract a matrix similar to *L*.

SHAP and its Application to VAEs

Shapley Additive Explanations (SHAP) is a technique used to quantify the contribution each feature brings to the output of a ML model [3].

The SHAP value of a feature is calculated by finding the difference between the model output value when the feature is included compared to when it is excluded.

However, as features are interlinked with each other, this is more complicated than just adding and removing features. We must consider all possible subsets of features and take a weighted average to find the total contribution (the SHAP value) of a feature.

As the number of subsets grows exponentially, this can become very complex. Hence, we usually perform SHAP using approximations and sampling.

We can perform SHAP on the VAE model to quantify the relationship between features in the input data and the latent variables that the VAE learns. Thus, SHAP can be used to find a structure similar to the Loadings matrix *L*.

Results of Simulation

To evaluate how effective SHAP is at finding a matrix similar to the Loadings matrix *L*, sparse matrices for *L* and *F* were simulated (*L is* shown in Figure 3).

In this simulation, I reproduced the analysis of former Masters of Science student Gyu Hwan Park. As $X \approx L \times F$, we simulated the data *X* from a known *L* and *F* structure. Then, we inputted *X* into the VAE and fit the model. After performing SHAP on the fitted model, the SHAP values were compared to the known entries of *L*.

Note that features 150 and 600 (for factor 1) correspond to non-zero and zero values, respectively. Hence, we expect the histograms of approximate SHAP values to have a wider x-axis range for feature 150 and a smaller x-axis range for feature 600 as seen in Figure 4.

This shows that SHAP values have a potential to provide a structure similar to the Loadings matrix *L*.

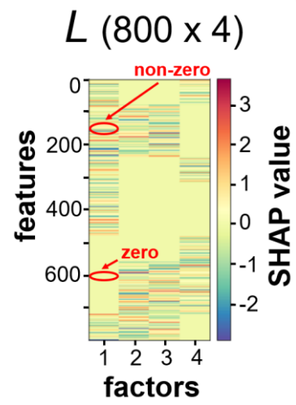

Fig. 3: Simulated matrix *L*

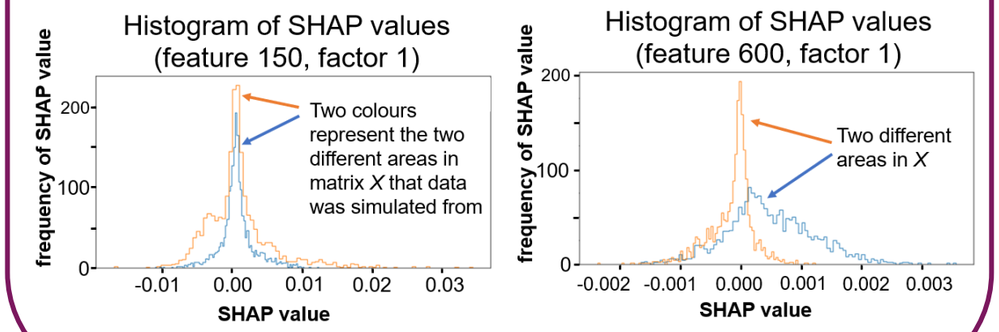

Fig. 4: Histograms of approximate SHAP values

Acknowledgements

I want to thank my supervisor, Dr Heejung Shim, for countless meetings and for guiding me through this project. I'd also like to thank Gyu Hwan Park for sharing his code and research material on SHAP and VAE. I'm very grateful to the School of Mathematics and Statistics for this invaluable opportunity to conduct mathematics research.

References

[1] Doersch, C., 2021. *Tutorial on Variation Autoencoders*. pp. 1-11

[2] Image modified from: Kan, E., 2018. What the heck are Vae-Gans? *What The Heck Are VAE-GANs?*

[3] Lundberg, S., Lee, S., 2017. *A Unified Approach to Interpreting Model Predictions*. 31st Conference on Neural Information Processing Systems

Resources for more background on VAE and SHAP:

- Altosaar, J., 2019. *Tutorial - What is a variational autoencoder?*.
- Mazzanti, S., 2020. *SHAP explained the way I wish someone explained it to me*.