Verbatim Machine Learning Model Manifestation in Manageable Neighborhoods

Xianlong Zeng*, Fanghao Song*, Zhongen Li*, Krerkkiat Chusap*, Chang Liu*
*School of Electrical Engineering and Computer Engineering, Ohio University, Athens, OH 45701 USA
liuc@ohio.edu

We propose to develop a way to explain machine learning models and facilitate software developers' understanding with verbatim (instead of approximate) model manifestation and only in manageable local neighborhoods of data points so that the scope is manageable for average human beings with average mental capacity. The framework is shown in Figure 1. Explanations are provided through the following steps: An VAE (Variational Autoencoder) model is first trained to capture the data distribution of the given dataset (X, Y), where X is the input data and Y is the label data. Next, VAE is used to compute the latent vector of the point of interest. The point of interest can be any data instance, most frequently a misclassified one. For example, a data point $(x_i, y_i = c_1)$ is misclassified as class c_2. Then, the latent vector of the item-of-interest is used as the center to grow a neighborhood in all latent dimensions, until the neighborhood includes at least some instances from Class c_1 and Class c_2. With the boundary of the neighborhood defined (the trained model is applied to predict the samples in the neighborhood), sample density is computed so that the total number of samples to show in the neighborhood is manageable (e.g., 15x15=225 or 20x20=400). Algorithm 1 explains our framework with more details. At the end of the algorithm, the neighborhood is visualized to provide a verbatim manifestation of the actual model because the color-coded classification results is from the actual underlying classifier.

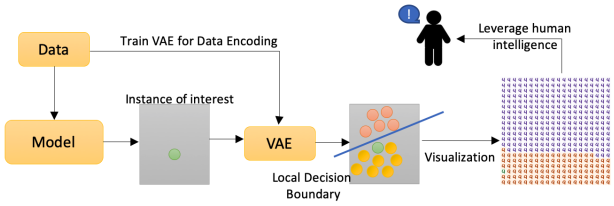

Fig. 1: Overview of the proposed framework

MNIST was used as the dataset in our experiments. Convolutional variational autoencoders and convolutional neural networks (CNN) were used as the VAE and the classification model, respectively. A CNN is trained and shows 97% accuracy on the MNIST dataset. To better understand how this trained CNN works and why it makes mistakes, a misclassified digit 4 is selected for interpretation.

Figure 2 demonstrates the result generated by our framework. Digit 4 (in green) is mispredicted as digit 9. The

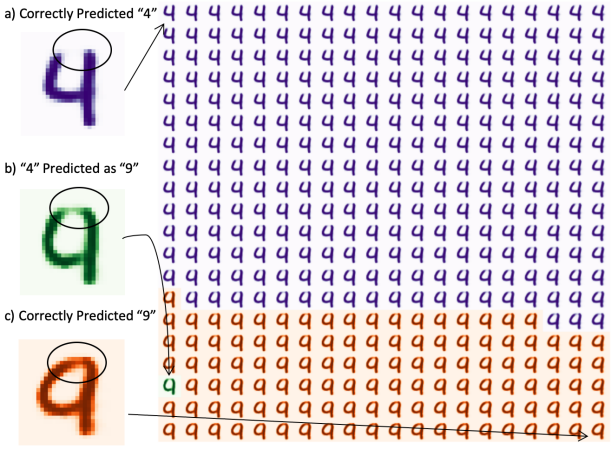

Fig. 2: A misclassified digit 4 from MNIST.

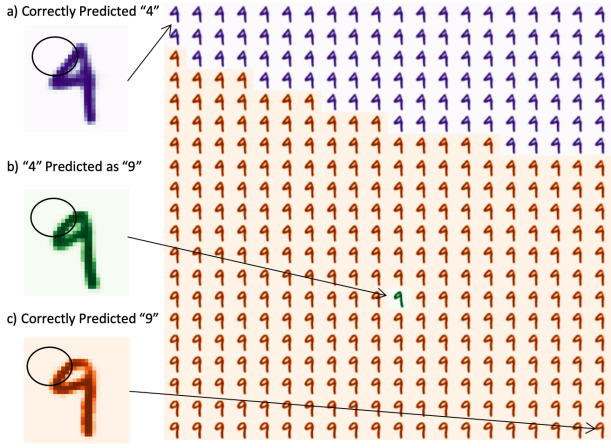

Fig. 3: Another misclassified digit 4.

nearest correctly predicted digit 4 (in purple) and digit 9 (in orange) are identified. Neighbors around those are generated to provide interpretation. By examining the morphing within the neighborhood, the circled area can be identified by a human as an explanation for the misprediction. Another example is shown in Figure 3. The local decision boundary of the model near the selected instance-of-interest is displayed. End users can better understand the model behavior by visually examine these samples. In this case, the misprediction is likely caused by the circle areas in the image's top-left region.