Interpreting Variational Autoencoders with Fuzzy Logic: A step towards interpretable deep learning based fuzzy classifiers

Kutay Bölat, Tufan Kumbasar
Department of Control and Automation Engineering
Istanbul Technical University
Istanbul, Turkey
bolatk,kumbasart@itu.edu.tr

***Abstract*—The emerging success of Deep Learning (DL) in various application areas comes also with the questions starting with "How"s and "Why"s. These questions can be answered if the DL methods are interpretable and thus provide a certain a degree of explanation. In this paper, we propose a DL framework that leverages the advantages of β-Variational Autoencoder (VAE) and Fuzzy Sets (FSs), which are disentanglement and linguistic representation, for the design of a novel DL based Fuzzy Classifier (FC). We first present a step-by-step design approach to construct the DL-FC which is composed of the encoder layer of β-VAE and a Fuzzy Logic System (FLS) followed by a softmax layer. The β-VAE is trained so that the semantic information of the high dimensional data is captured. The latent space of the β-VAE is clustered to extract FSs. The FSs are then used to define antecedents of the FLS that is trained with DL methods. We present results conducted on the MNIST dataset and showed that DL-FC is quite competitive with its deep neural network counterpart. We then try to provide an interpretation to the antecedents of FLS by examining the FSs, the latent traversals and heat-maps of each latent dimension. The results show that the antecedents of FLS can be defined with linguistic interpretations. Thus, for the first time in the literature, we showed that linguistic interpretations can be defined for the latent space of β-VAE with FSs.**

***Index Terms*—Variational autoencoder, fuzzy sets, fuzzy c-means clustering, classification, interpretation.**

I. Introduction

In the last decade, Deep Learning (DL) methods showed a great success in various fields [1], [2], [3], [4]. This is mostly due to the high learning capacity of Deep Neural Networks (DNNs) composed of hundreds to thousands of layers and neurons [5]. However, as the DNN structures get more complex, interpreting the underlying reasons of the predictions becomes more challenging [6]. One of the attempts to solve the interpretability problem is making visual inspections (such as heat-maps) on the DNN layers in order to interpret which nodes are activated for images of different classes [7]. Another approach is the representation learning which tries to find the underlying representations of the data so that even low capacity and interpretable machine learning algorithms can achieve high accuracies [8]. In this aspect, generative networks are very promising to provide an interpretation since they use the underlying representations of the data to generate unseen data [9].

Variational Autoencoders (VAEs) [10] are one of the most popular generative networks among the field of representation learning. The main difference between VAEs and its conventional autoencoder counterparts [11] is that VAEs use a Bayesian inference for representing their low dimensional latent space. For instance, in [12], VAE is used along with decision in order to design an interpretable classifier. In [13], a new loss function with a hyperparameter β is presented for the training of VAE (abbreviated as β-VAE) to obtain disentangled latent dimensions. Disentanglement in β-VAE can be defined as the assignment of certain features of high dimensional data to unique latent dimensions so that a variation in one latent dimension is sensitive only to the corresponding feature [14], [15].

For interpretability, Fuzzy Logic Systems (FLSs) are great tools thanks to their linguistic representation and rule based structure [16]. Thus, employing FLSs to machine learning algorithms can be seen as an important step for both research areas [17]. Accordingly, there has been some promising attempts to accommodate FLSs in DNN. For instance, in [18], DL methods are adapted to solve the design problem of FLSs whereas in [19], type-2 FLSs are used as activation layers to enhance the performance of DNNs. Conventional autoencoder networks are also used with FLSs for fuzzy rule reduction [20], data processing [21] and dealing with data uncertainty [22].

There are also some promising studies about employing FLSs into DNNs to provide interpretability. For instance in [23], a stacked fuzzy classifier is proposed to obtain interpretability on each layer. In [24], an architecture is proposed where convolutional DNNs are used to extract features, then these features are clustered with FSs. This approach is improved in [25] by proposing a deep fuzzy clustering algorithm to automatically extract the features.

This paper proposes a framework that uses the merits of β-VAEs and FSs, namely disentanglement and linguistic

This research is supported by the project (118E807) of Scientific and Technological Research Council of Turkey.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

interpretation, for the design of a DL based Fuzzy Classifier (FC). In the design of the DL-FC, firstly a β-VAE is trained so that a low dimensional disentangled latent space is obtained. The latent space of the β-VAE is extracted and then clustered with the well-known Fuzzy c-Means (FCM) algorithm [26] in order to define the latent space with FSs. The generated FSs are then used as the antecedent Membership Functions (MFs) of the FC which is defined with a multi input multi output Takagi-Sugeno-Kang (TSK) FLS. In the design of the FC, only the consequent parameters of the FLS are trained. We conducted comparative experiments on the MNIST dataset to analyze the performance and interpretation of the proposed DL-FC. We firstly investigated the classification performance of the DL-FC and showed that its performance is quite competitive with its DNN counterpart. We then presented comprehensive investigations to provide linguistic interpretations to the antecedent part of the FLS. In this context, we firstly analyzed the antecedent FSs and observed that they are quite distinguishable. Then, to define them with linguistic variables and descriptions, we analyzed the latent traversals and heatmaps of the latent dimensions. We demonstrated that the latent dimensions and their FSs can be defined with linguistic interpretations, especially for the ones that have high semantic information. Thus, for the first time in the literature (to best to our knowledge) we concluded that the latent space of β-VAE can be represented with FSs that provide them a linguistic interpretation. We believe that the results of this study will open the doors to wider use of FSs in designing interpretable DL methods.

This paper is organized as follows: Section II introduces the DL-FC and its step-by-step design method. Section III presents the comparative results in order to examine the classification performance and interpretation of the latent space extracted from β-VAE. Finally, Section V closes this paper with conclusions and future work.

II. Deep Learning based Fuzzy Classifier

Here, we present the structure of the proposed DL-FC shown in Fig. 1 and its step-by-step design which consists of following main steps:

1) Training the β-VAE: A disentangled latent space is extracted to obtain semantic information of the high dimensional data.
2) Clustering the latent space via FCM: The extracted latent space is clustered to generate FSs.
3) Training the Fuzzy Classifier: The generated FSs are used to construct an FC which is trained with DL methods.

The employed layer structures and their design steps are explained in the following subsections.

A. Step-1: Training the β-VAE

Here, we present the structure of β-VAE that is used to extract a disentangled latent space. As it can be seen from Fig. 2, β-VAE encodes the high dimensional input $\boldsymbol{x}$ to a K dimensional space which is defined with following multivariate Gaussian distribution:

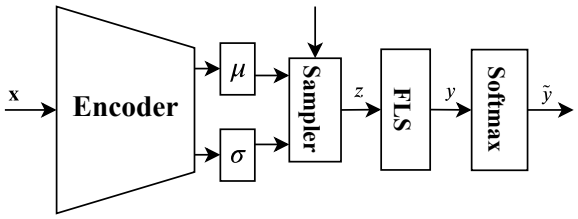


Figure 1: The internal structure of the proposed DL-FC.

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{z}_\mu, \boldsymbol{z}_\sigma^2 \boldsymbol{I}) \tag{1}$$

where $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, $\boldsymbol{z}_\mu$ and $\boldsymbol{z}_\sigma$ correspond to the variational posterior distribution and K-dimensional mean and standard deviation vectors of this distribution, respectively. $\boldsymbol{z}$ represents a K-dimensional sampled latent vector which is propagated through the decoder in order to reconstruct the original input $\boldsymbol{x}$ as $\tilde{\boldsymbol{x}}$. In the design of β-VAE, the following two objectives are considered [13]:

- assigning latent vectors $\boldsymbol{z}$ to $\boldsymbol{x}$ whose expected values are distinctive enough, so that the reconstructions are accurate ($\boldsymbol{x} \approx \tilde{\boldsymbol{x}}$)
- finding a good posterior distribution, so that the transitions within latent vectors give smooth transitions on $\tilde{\boldsymbol{x}}$.

In this context, the following loss function is defined:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \beta D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \tag{2}$$

where ϕ and θ represent the weights of the encoder and the decoder networks, respectively. The term E_{q_ϕ} defines the reconstruction loss (i.e. squared error or cross-entropy loss) while D_{KL} is the Kullback-Leibler divergence loss (KL) between the posterior and unit normal distribution, $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$, which can be seen as a regulator to keep the posterior close to unit normal distribution [13]. The scalar β is a hyperparameter for adjustment of a crucial aspect of β-VAE, namely disentanglement [13], [14] [15].

In the proposed method, as the aim is to extract an interpretation, it is crucial to acquire a good disentangled latent space. The disentanglement of the latent space is closely related to the balance between E_{q_ϕ} and D_{KL} [13]. Thus, the training of β-VAE has to be accomplished with a proper β value since there is a trade-off between a good reconstruction performance and disentanglement. The training of the β-VAE is accomplished via the Adam optimization algorithm [27]. Then, the sampled latent vectors ($\boldsymbol{z}$) are extracted for the whole training dataset.

B. Step-2: Clustering the latent space via FCM

In this step, we cluster the latent space with FCM clustering algorithm to generate FSs to be used and processed by the FC. If the latent space is disentangled enough, we believe that the interpretability of the generated FSs of the DL-FC will be high.

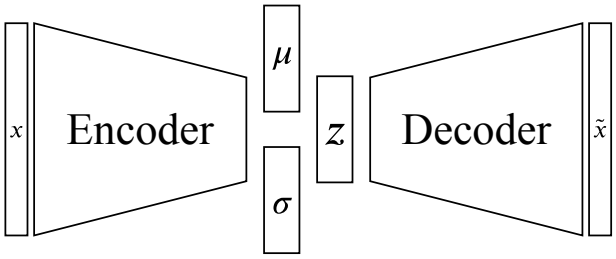


Figure 2: VAE structure.

FCM is based on the minimization of the following objective function [26]:

$$J_m = \sum_{d=1}^{D}\sum_{j=1}^{N} \mu_{dj}^{m} \|\boldsymbol{x}_d - \boldsymbol{c}_j\|^2 \tag{3}$$

where D is the number of data points, N is the number of clusters, m is fuzzy partition matrix exponent for controlling the degree of fuzzy overlap where $m > 1$, $\boldsymbol{x}_d$ is the d^{th} data point, $\boldsymbol{c}_j$ is the center of the j^{th} cluster, μ_{dj} is the degree of membership of $\boldsymbol{x}_d$ in the j^{th} cluster. For a given data point, $\boldsymbol{x}_d$, the sum of the membership values for all clusters is one. The resulting cluster centers and their standard deviations are approximated with Gaussian FSs to ease the training process of the FC.

C. Step-3: Training the Fuzzy Classifier

The rule structure of the fuzzy classifier is constructed with a K input and M output FLS which is defined as:

$$\begin{gathered}\text{R}_n\text{: IF } x^{(1)} \text{ is } A_n^{(1)} \text{ AND } \ldots \text{ AND } x^{(K)} \text{ is } A_n^{(K)} \\ \text{THEN } \boldsymbol{y}_n = \boldsymbol{b}_{n,0} + \textstyle\sum_{k=1}^{K} \boldsymbol{b}_{n,k} x^{(k)}\end{gathered}$$

where $x^{(k)}$ defines the inputs ($z^{(k)}$) which is partitioned with N Gaussian FSs $A_n^{(k)}$, $\boldsymbol{b}_{n,k}$ represents the consequent parameters of the FLS and the total number of fuzzy rules is N. The FLS uses and employs the product implication and the center of sets defuzzification method [16]. The firing level of the n^{th} rule is as follows:

$$f_n(\boldsymbol{x}) = \prod_{k=1}^{K} \mu_{A_n^{(k)}}(\boldsymbol{x}) \tag{4}$$

where $\mu_{A_n^{(k)}}(\boldsymbol{x})$ is the membership degree of the antecedent FSs. Then, the M outputs of the FLS are calculated as follows:

$$\boldsymbol{y}(\boldsymbol{x}) = \frac{\sum_{n=1}^{N} f_n(\boldsymbol{x}) \boldsymbol{y}_n(\boldsymbol{x})}{\sum_{n=1}^{N} f_n(\boldsymbol{x})} \tag{5}$$

Here, $\boldsymbol{y}$ is an M dimensional output vector of FLS. Then, the outputs of FLS y are passed to a softmax layer as follows:

$$\tilde{y}^{(m)} = \frac{exp(y^{(m)})}{\sum_{m'=1}^{M} exp(y^{(m')})} \tag{6}$$

where $\tilde{\boldsymbol{y}}$ is an M dimensional vector whose elements are scaled between 0 and 1.

In the training of the presented DL-FC, as the antecedent Gaussian MFs $A_n^{(k)}$ are generated from the FCM algorithm, only the parameter vectors $\boldsymbol{b}_{n,k}$ have been handled as learnable parameters. In this context, we define a tensor $\mathcal{B}$ with a size of MxKxN that represents the learnable parameters of the FC. The training of the DL-FC is achieved through the minimization of the cross-entropy loss function via the Adam optimization algorithm [27].

Table I: β-VAE Layer Specifications and Hyperparameters.

Adam optimizer	Learning rate: 0.001 Gradient decay factor: 0.9 Squared gradient decay factor: 0.999
Layers	Input: 28x28x1 Encoder (ReLU activation): Conv 16x7x7, 32x5x5, 64x64x3, FC 3136 Latents: 10 Decoder (ReLU activation): Trans. Conv 64x7x7, 32x5x5, 16x3x3, 16x1x1

III. Experiments

Several experiments were conducted on hand-written digits dataset MNIST [28] in order to examine the performance and interpretation of the DL-FC. The first 60K samples of the MNIST dataset are used for training while the remaining 10K are used for testing. We performed all experiments in MATLAB and CUDA environments on a PC that includes Intel(R) Core(TM) i9-7900X 3.3GHz CPU, 64GB RAM and NVIDIA GeForce GTX 1080 TI GPU.

A. Classification Performance

As stated in Section II-A, we first trained a β-VAE with the layer and hyperparameter settings which are given in Table I. We have experimentally found that a latent space dimension of $K = 10$ and a β value of 0.01 result with a satisfactory reconstruction performance and disentanglement in the latent space. The resulting loss values, defined in (2), are calculated as 0.0421922 and 0.0422257 for the training and testing datasets, respectively. The trained β-VAE resulted in a disentangled latent space whose latent dimensions have the following KL values:

$$\textbf{KL=}[3.03; 2.15; 0.73; 1.99; 1.10; 1.30; 1.37; 2.54; 3.18; 0.49]$$

where the k^{th} element of the vector defines the KL value of the k^{th} latent dimension ($z^{(k)}$). Latent dimensions with higher KL values indicate that these dimensions encode relatively more information [13]. To analyze the effect of each latent dimension, we sorted the KL values and matched them with their corresponding latent dimension ($z^{(k)}$) as shown in Fig. 3. Then, as the KL value provides a measure on the encoded information, we showed in Fig. 4 that how the encoded information is increasing with the number of latent dimensions. We can observe that the first 6 dimensions of the sorted latent dimensions contain almost 80% of the total information for the dataset.

We extracted the $\boldsymbol{z}$ vectors and then clustered them to used them as the antecedent MFs of the FC. The testing performances of the DL-FCs for $N = \{3, 5, 10\}$ number of rules is given in Fig. 5. Note that, the latent dimensions are

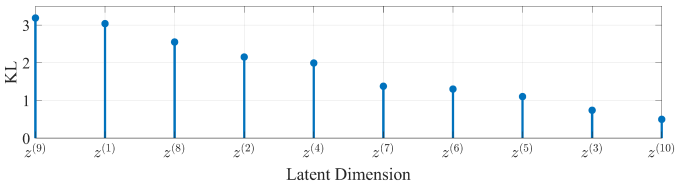


Figure 3: KL values of each latent dimension.

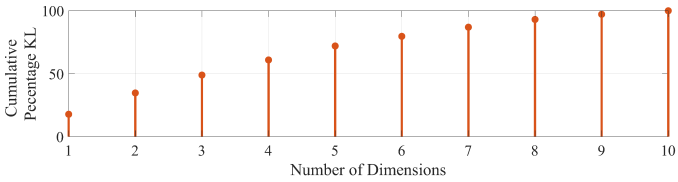


Figure 4: Variation of the cumulative KL value with respect to the total number of inputs.

sorted according to their KL values as it has been done in Fig. 4. This result shows that the increments in the accuracy with the addition of the next important latent dimensions impact similarly as discussed for the results given in Fig 4. It can be concluded that the first 6 latent dimensions, which have relatively high KL values, have the biggest impact on the accuracy as expected. It can be also observed that the number of rules does not have a huge impact on the accuracy and thus, one might conclude that the DL-FC composed of $N = 3$ rules is sufficient for the handled classification problem.

We also compared the performance of the DL-FCs with their Linear Classifier (LC) and DNN counterparts that use and process all latent dimensions. In the design of DNNs, we constructed a 5 layer DNN that uses *Tanh* as its activation function. Moreover, for a fair comparison, the total number of learnable parameters (i.e. weights) of the DNN was kept equal to the size of the tensor $\mathcal{B}$ which is MKN. The testing accuracy performance measures of the classifiers are tabulated in Table. II. Here, the DNN classifiers abbreviated as DNN-1, DNN-2 and DNN-3 are the counterparts of the DL-FCs with $N = 3$, $N = 5$ and $N = 10$, respectively. It can be observed that the performances of the DL-FCs are quite competitive with their DNN counterparts.

B. Linguistic Interpretation of Latent Dimensions

In this section, we examine the interpretability of the DL-FC composed of $N = 3$ rules/clusters since its resulting performance is quite satisfactory as shown in the preceding subsection. We start by analyzing the FSs for the latent vectors that resulted in high KL values which are $z^{(1)}$,$z^{(2)}$,$z^{(4)}$,$z^{(7)}$,$z^{(8)}$ and $z^{(9)}$.constructed. In Fig. 6 the generated FSs of each latent dimension is presented. It can be seen that the FSs are quite distinguishable which might provide an opportunity to define them with linguistic variables and descriptions, i.e. interpretation. However, the interpretability of the DL-FC does not only depends on the generated FSs but also on the semantic information which is embedded in the latent dimensions. Therefore, we examine the output sensitivity of β-VAE with respect to each latent dimension both qualitatively and quantitatively to comment on the interpretation of the DL-FC.

We analyze the latent traversals in order to find out which latent dimension captures the data generative factors, i.e. linguistic variables of the FLS. Thus, firstly a latent vector is arbitrarily chosen. Then, a traverse is performed only on the latent dimension of interest ($z^{(k)} : -3 \rightarrow 3$) and the outputs of β-VAE are analyzed. In Fig. 7, the latent traversals are given for $z^{(1)}$,$z^{(2)}$,$z^{(4)}$,$z^{(7)}$,$z^{(8)}$ and $z^{(9)}$. However, in order to conclude on the linguistic descriptions, we need to analyze various latent traversals in all latent dimensions. In this context, we generated and examined the heat-maps of each latent dimension. The generation of the heat-maps requires the gradients of each pixel between two consecutive decoder outputs of a latent traverse. An example of a consecutive gradient transition on $z^{(1)}$ is illustrated in Fig. 8 where the brighter pixels define positive gradients while darker pixels correspond to negative gradients. Note that, the gradient of each $z^{(k)}$ always starts with negative values and incrementally transforms to positive values as the latent traverse direction is always defined from $z^{(k)} : -3 \rightarrow 3$. This gradient calculation was run 1000 times and means of the results were calculated for each latent dimension in order to achieve the statistical mean of every pixel-wise gradient. In Fig. 9, the generated heat-maps of the latent dimensions are presented. In the light of Fig. 7 and Fig. 9, one may define the latent dimensions that have the highest KL values as follows:

- $z^{(1)}$ with the linguistic variable "Inclination angle of the digit". The antecedent MFs $A_1^{(1)}$, $A_2^{(1)}$, $A_3^{(1)}$, given in Fig. 6b, can be represented with the linguistics terms "*Zero*","*Negative*" and "*Positive*", respectively. This can be also clearly seen from the latent traversal shown in Fig. 7b.
- $z^{(8)}$ with the linguistic variable "Horizontal position of the digit's curvature". For this case, since $A_1^{(8)}$ and $A_3^{(8)}$ are very similar, the MFs $A_1^{(8)}$, $A_2^{(8)}$, $A_3^{(8)}$ (given in Fig. 6c) can then be defined with the linguistics terms "*Down*", "*Up*" and "*Up*", respectively. This can be clearly seen from the latent traversal given in Fig. 7c.
- $z^{(9)}$ with the linguistic variable "Circularity of the digit". The corresponding FSs, given in Fig. 6a, $A_1^{(9)}$, $A_2^{(9)}$, $A_3^{(9)}$, can then be defined with the linguistics terms "*Small*", "*Zero*" and "*Large*", respectively. This can be also clearly seen from the latent traversal presented in Fig. 7a.

It is worth mentioning that we defined the linguistic variables by taking into account the most dominant underlying factors. Note that, defining linguistic variables and terms for the latent dimensions $z^{(2)}$, $z^{(4)}$, $z^{(7)}$ and their corresponding FSs is not straightforward since their semantic information, i.e. KL values, is relatively low in comparison to the ones of $z^{(1)}$, $z^{(8)}$ and $z^{(9)}$.

IV. Conclusions and Future Work

In this study, we proposed a DL-FC that leverages the advantages of β-VAE and FSs which are disentanglement

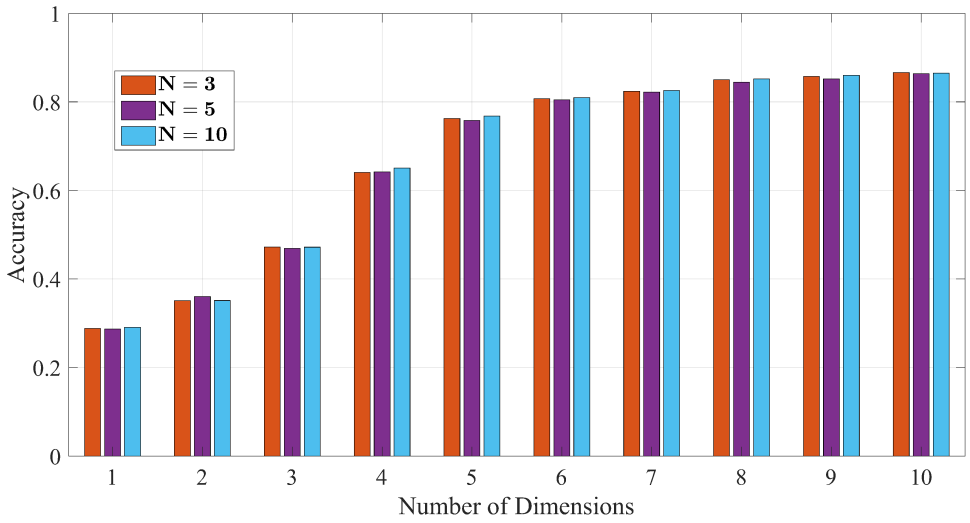


Figure 5: Testing performances of the DL-FCs.

Table II: Accuracy values of LC, DNN and DL-FC structures.

Classifer	LC	DNN-1	DNN-2	DNN-3	DL-FC (N=3)	DL-FC (N=3)	DL-FC (N=3)
Accuracy	0.830	0.871	0.883	0.883	0.869	0.864	0.867

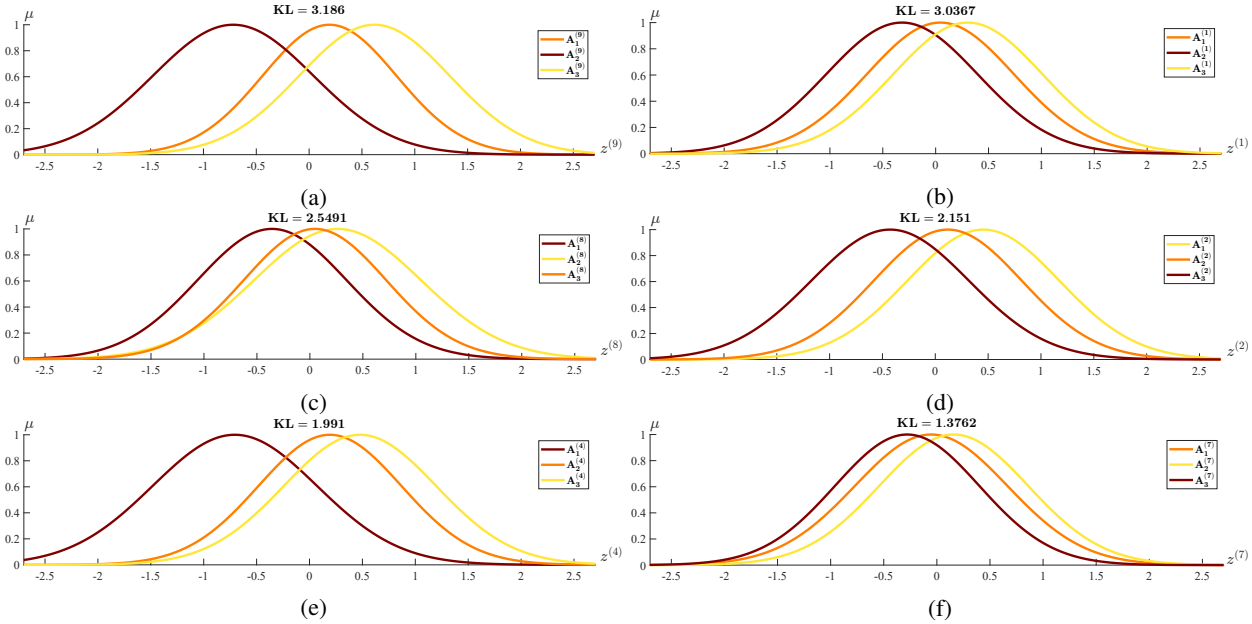
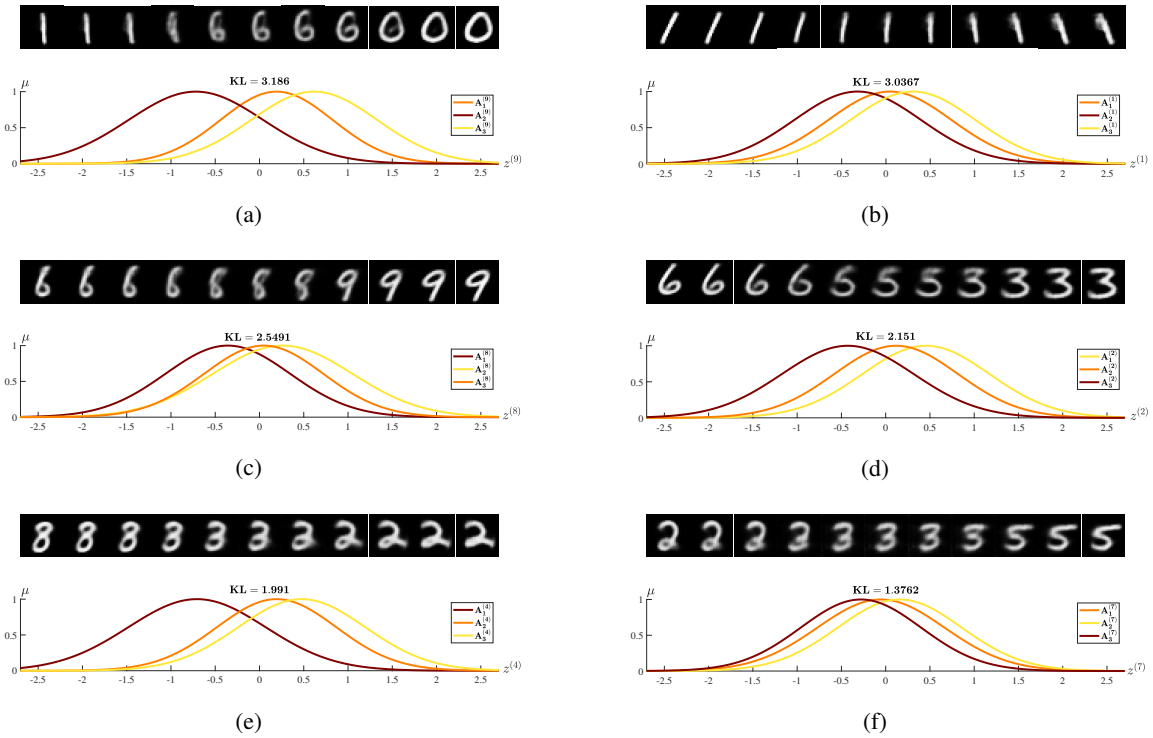


Figure 6: Antecedent MFs of DL-FC.



(a) (b) (c) (d) (e) (f)

Figure 7: Latent traversals (a)$z^{(9)}$,(b)$z^{(1)}$,(c)$z^{(8)}$,(d)$z^{(2)}$,(e)$z^{(4)}$,(f)$z^{(7)}$

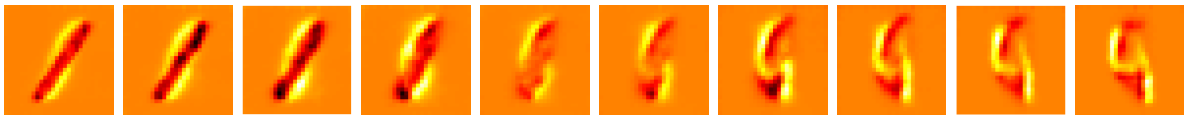


Figure 8: Transition of bit-wise gradients for an arbitrary z-Traverse on $z^{(1)}$.

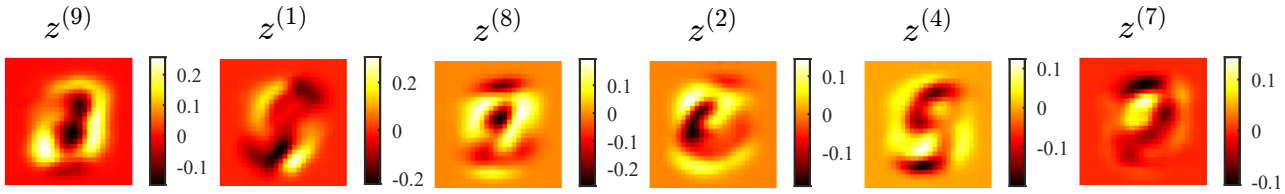


Figure 9: Heat-maps of latent dimensions that have highest KL values.

and linguistic representation. We first presented a step-by-step design method for the DL-FC. We trained a β-VAE such that a disentangled latent space is obtained. Then, this latent space of β-VAE is clustered to extract FSs. The generated FSs are used to construct an FC that is trained with DL methods. We presented various experimental results and showed that the performance of DL-FC is quite satisfactory. More importantly, we defined linguistic interpretations to the latent space of β-VAE by examining the antecedent MFs, latent traversals and heat-maps of the latent dimensions with high KL values. Although we believe that the results of the paper are quite important milestones in placing FSs in the research area of DL, there is still the following question to be answered: "How does the DL-FC make predictions?"

As for our future work, we plan to first try to provide an interpretation for the rule base of DL-FC. We also plan to employ type-2 FSs as antecedent MFs as they are powerful tools to represent the higher levels of uncertainties [16].

REFERENCES

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298594

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," 2011.

[4] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015. [Online]. Available: http://dx.doi.org/10.3115/v1/P15-1001

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[6] C. Buckner, "Deep learning: A philosophical introduction," *Philosophy Compass*, vol. 14, no. 10, p. e12625, 2019, e12625 PHCO-1206.R1. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/phc3.12625

[7] Q.-s. Zhang and S.-C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018.

[8] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, p. 1798–1828, Aug 2013. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2013.50

[9] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," 2016.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[11] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3377–3381.

[12] P. Quint, G. Wirka, J. Williams, S. D. Scott, and N. V. Vinodchandran, "Interpretable classification via supervised variational autoencoders and differentiable decision trees," 2018.

[13] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[14] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv preprint arXiv:1804.03599*, 2018.

[15] H. Sikka, W. Zhong, J. Yin, and C. Pehlevan, "A closer look at disentangling in β-vae," *arXiv preprint arXiv:1912.05127*, 2019.

[16] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions, 2nd Edition*, 2nd ed. Springer Publishing Company, Incorporated, 2017.

[17] E. Hüllermeier, "Does machine learning need fuzzy logic?" *Fuzzy Sets and Systems*, vol. 281, pp. 292 – 299, 2015, special Issue Celebrating the 50th Anniversary of Fuzzy Sets. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011415004133

[18] D. Wu, Y. Yuan, and Y. Tan, "Optimize tsk fuzzy systems for regression problems: Mini-batch gradient descent with regularization, droprule and adabound (mbgd-rda)," 2019.

[19] A. Beke and T. Kumbasar, "Learning with type-2 fuzzy activation functions to improve the performance of deep neural networks," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 372 – 384, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197619301551

[20] R. K. Sevakula and N. K. Verma, "Fuzzy rule reduction using sparse auto-encoders," *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–7, 2015.

[21] R. K. Sevakula, A. Shah, and N. K. Verma, "Data preprocessing methods for sparse auto-encoder based fuzzy rule classifier," *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, pp. 1–6, 2015.

[22] B. Costa and J. Jain, "Fuzzy deep stack of autoencoders for dealing with data uncertainty," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, June 2019, pp. 1–6.

[23] T. Zhou, F. Chung, and S. Wang, "Deep tsk fuzzy classifier with stacked generalization and triplely concise interpretability guarantee for large data," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1207–1221, Oct 2017.

[24] M. Yeganejou, S. Dick, and J. Miller, "Interpretable deep convolutional fuzzy classifier," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2019.

[25] M. Yeganejou and S. Dick, "Improved deep fuzzy clustering for accurate and interpretable classifiers," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, June 2019, pp. 1–7.

[26] J. C. Bezdek, R. I. Ehrlich, and W. E. Full, "Fcm: The fuzzy c-means clustering algorithm," 1984.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[28] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/