Grenze International Journal of Engineering and Technology, Jan Issue

Semantic Transformation of One-Dimensional Data using a Variational Autoencoder with Double Filter Architecture

Yoga Harshitha Dudduku
Dept. of Computer Science, GITAM deemed to be University, Hyderabad, India
Email:ydudduku@gitam.in, dyogaharshitha@gmail.com

***Abstract*— This paper presents an algorithm and code for training a variational autoencoder to perform semantic transformation of USDA SR legacy food data into latent space. The term transformation is used rather than dimensionality reduction to emphasize that the main aim of the network is to learn to features from it, apart from compressing. The model is expected to learn features (glycaemic index, superfood), and filter out shallow attributes for lossless compression, which would ease the training of any other neural network it. The dimension has been reduced by 50% as a byproduct. The architecture designed to be equipping with prior knowledge [13]. The scope of this study is to realize this goal with minimum resources and maximum accuracy. Dataset was limited and diverse, which was challenging to train. Traditional dimensionality reduction techniques do not promise extraction of useful feature that condemns the scalability of the project from further application.**

***Index Terms*— Variational autoencoder, dimensionality reduction, feature extraction, GAN training, knowledge representation, USDA SR legacy food data**

I. INTRODUCTION

Motivating project behind training VAE [11] on the USDA SR Legacy dataset is to build a conditional GAN that can generate a meal plan based on a person's nutritional needs. Training the GAN on continuous and transformed data equips the model with prior knowledge to compensate for the data that cannot be provided [13][14]. The objective is not just build a blind mapping network but a meaningful transformation that can be easily scaled. Learning features by deep neural network VAE did not reproduce the data with accuracy due to limitations imposed in training network with less data. This paper proposes a VAE with a double filter architecture. We employ extra filters that are pulled from the initial layers of the encoder and are processed in the AdaScl block of the decoder. The main focus was to improve the accuracy of the model and the efficient usage of resources. The USDA SR Legacy dataset contains nutritional information about various types of food. Nutritional data are generally highly skewed and have a high standard deviation. The data can be characterized by the sparse and dispersed values of micronutrients, categorical values of type of food, and unavoidable outliers. Normalization of raw data (cleaned and pre-processed data has resulted in data loss and an increase in error (on actual data) due to high standard deviation.
Semantic transforming of data involves extracting features (GIindex, rich or poor in micro nutrition, fatty source)

Grenze ID: 01.GIJET.10.1.171

from the data, along with a reduced dimension of approximately 50%. We use a VAE to encode the data into feature space and a decoder that is trained to embed hidden patterns in the data and retrieve the data from the feature vector. This architecture can be conveniently applied to other nutritional datasets and any other one-dimensional data.

II. Literature Review

A. Related Work

The implementation of VAE on the USDA food dataset draws motivation from the most popular problem of image compression [18],[19] implemented by neural networks. Traditional dimensionality reduction [25] is unable to produce the expected performance (refer table 4.1, first row). The challenges faced with compressing USDA SR legacy data are:

- The dataset is sparse and skewed. The presence of extreme outliers that cannot be ignored
- Diverse dataset
- There is no established procedure for extracting features.

B. Realizing the objective using the deep neural network architecture

As mentioned earlier, the goal of extracting features and mapping them into a latent space that can be retrieved when required can be achieved by employing an auto-encoder network. VAE has proven advantages over traditional autoencoder[12], VAE generates latent space that is continuous [1].

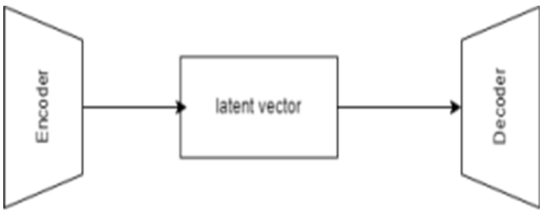

Fig. 2.1 Block diagram of the variational autoencoder

VAE [2], [3], [4] is considered a probabilistic version of the autoencoder. A gentle introduction to its internal working that led to the design model architecture with double filter by adding AdaScl block(refer fig 3.2). VAE generate *x* data with from a latent vector *z*

$$p_\theta(x,z) = p_\theta(x|z)p_\theta(z) \quad (1)$$

$p_\theta(x|z)$, $p_\theta(z)$ and θ denote respectively the generative model, the prior distribution, and the model parameters, respectively. The model is trained by minimizing the likelihood.

$$p_\theta(x,z) = \int p_\theta(z)\, p_\theta(x|z)\, dz \quad (2)$$

However, computing this integral is complicated and intractable. This makes the estimation of the posterior distribution pθ(z|x) impossible. Therefore, information about the characteristics of the latent vector *z* cannot be obtained. To overcome this, the VAE model resolves these problems based on variational inference. Variational inference involves approximating the posterior distribution $p_\theta(x|z)$ to a tractable one $\left(q_\phi(z|x)\right)$ by Kullback–Leibler divergence minimization.

$$\log p_\theta(x) = L(\phi,\theta,x) + KLD\left(q_\phi(z|x)\,||\,p_\theta(z|x)\right) \quad (3)$$

where KLD≥0 and L(ϕ, θ, x) are the variational lower bounds expressed by the below equation:

$$L(\phi,\theta,x) = -KLD\left(q_\phi(z|x)\,||\,p_\theta(z)\right) + E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] \quad (4)$$

III. Algorithm Design

The encoder is designed to extract high-level features and compress the data as much as possible. Before going into the details of the algorithm, understanding the data pattern is necessary.

A. Insights into data

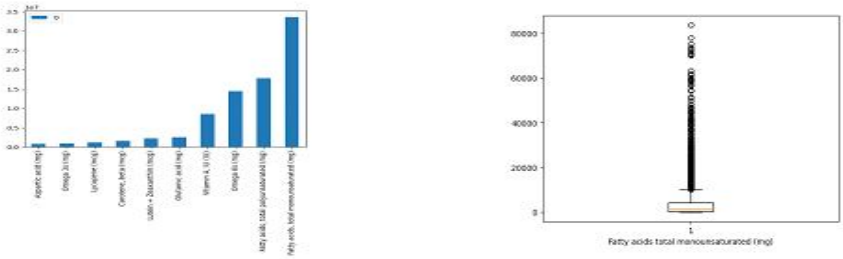

Fig 3.1 (a) Variance of various columns in dataset (b) box plot of monounsaturated fat distribution

Values are skewed and scattered to the extreme ends (bi-modal distribution of micronutrients), which requires the model to achieve an accuracy of 1e–5 when normalized. To remove the effect of outliers on the remaining data, 1000 (where values are densely populated) is taken as the cutoff value and the data is divided into two separate channels, one with data that has a cutoff at 1000 and the other with outliers above 1000. This further increase in the dimensions is welcome to decrease the effect of the outliers that are to be modelled [8]. The output generated by the model with 2 channels is combined (clipped between zero and one and then scaled. Model implemented to obtain a single vector to represent nutrition data.

B. Algorithm and architecture

For lossless compression, we use a combination of both deep-learned features and shallow scaling attributes from the data. The encoder is designed to provide multi-attention [15],[23] to both. The upper bound on the Deep feed forward Net [5x1 50,70,30 conv1D] is imposed by the number of clusters in which the data can be grouped [9]. Feed forward network is tapped at the starting layers and passed to the dense layer to filter the scaling attributes. This architecture imparts the model with prior knowledge when data cannot be provided in sufficient quantity[16]7.

The decoder passes the output from the encoder network through the transpose network, which is recalibrated by the AdaScl [5] block that takes reference from the dense filter output of the encoder.

$$AdaScl(z_a, \mu, \sigma) = \sigma z_a + \mu \quad (5)$$

$$\mu = f(z_b) \quad (6)$$

$$\sigma = g(z_b) \quad (7)$$

The latent vector is the concatenation of Z_a and Z_b. Z_a is the encoder output from convent and Z_b is branched from the dense filter

The AdaScl block [3],[24] used in the model differs from the AdaIN block used in the style GAN [7], and normalization [6] is not performed for the instance. The block recalibrates the output of the Transpose Conv layer by shifting it to the calibrated mean and multiplying it by the calibrated variance [17]. Calibration is made adaptive as it learns from the squeezed dense filters of the encoder network.

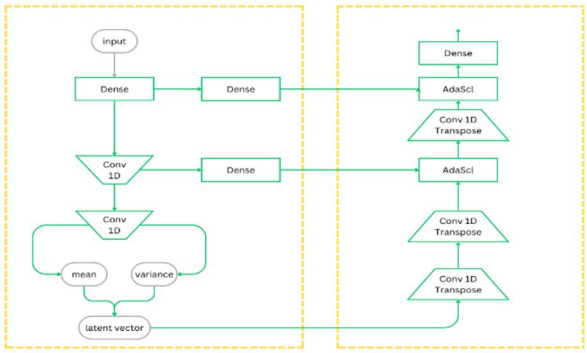

Fig. 3.2 VAE architecture used

The model architecture was selected in such a way that it would work for any one-dimensional data without compromising the model performance [20][21]. The architecture draws motivation from image processing, which has a rich set of literature for review and is customized to meet the needs of nutritional data [10].

IV. Training And Results

The model has parameters of size 2.1 MB. Training on the data took 50 epochs at a rate of 2 s/epoch on a CPU with 8 GB RAM.

A. Results

The metrics used for training the model were the mean absolute error and squared cosine similarity corrected by adding one[22].

$$loss = mae + \lambda\, gp$$

$$gp = (1 + \Sigma \cos_{\text{similarity}})\ ^2$$

The value of lambda is chosen such that the model tries to reduce the cosine similarity calculated until both distributions are at least orthogonal. Tries to minimise both MAE and cosine similarity simultaneously and then concentrates on fine tuning by reducing MAE . This strategy prevents model from taking greedy approach and getting stuck in the local minima.

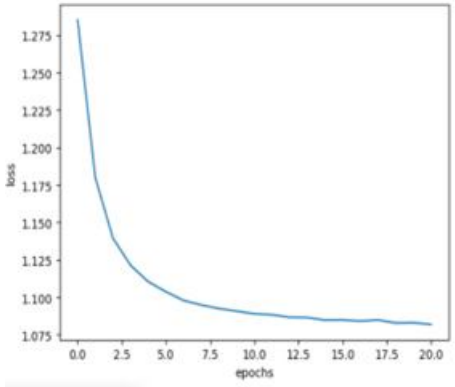

Fig. 4.1 Training VAE, total loss vs. epochs

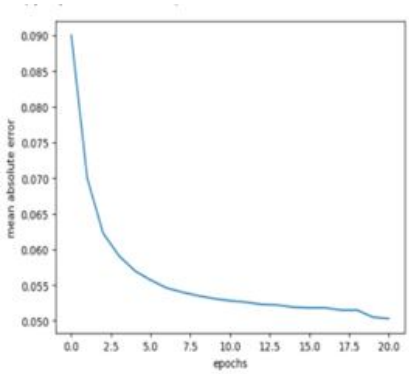

Fig. 4.2 Training VAE, mean absolute error vs. epochs

The above figure depicts the training results for the variational autoencoder. The model was able to confine the total loss to 0.0075 and MAE to 0.0071.

The KNN distance is calculated on the final result that is obtained by merging two channels into a single channel and comparing it with the actual raw data. While merging and de-normalizing the result, the accuracy is preserved through the end of the pipeline by clipping the results of the model between zero and 1. Thus clipping the effect of extreme values on the others.

B. validation

Before finalizing the model, various architectures were explored and their results were compared. The efficient utilization of resources was studied, and the most efficient model with better accuracy was finalized.

TABLE I Performance Comparison Of Various Models On The USDA SR Legacy Dataset

	MAE	KNN distance	Model size in MB
VAE with feed forward network	0.1	127	3
VAE with dense and conv layers	0.078	75	2.36
VAE with the AdaScl block with data split into 2-channel data	0.0071	23	2.1

VAE with AdaScl block with data split into 2 channels was the final model whose performance was discussed. In all the above cases, the compression was 50% resulting in feature vector of dimension 50, except last one with 65 dimensions. In order to validate the architecture, we compare with naïve model with feed forward network that has failed to converge to minimum and extract features due to small and divergent dataset. The architecture was modified to incorporate convolution network which was able to compress data but the final output after de-normalisation was not up to the mark. Filtering out the features from early layers of encoder network and feeding it to the decoder at the later stage as shown in the fig.3.2 has resulted in lossless compression, in addition to feature extraction by deep learning. The data was split into two channels to remove the effect of outliers. Even though splitting data into channels has increased the sparsity of the data, the model equipped with prior knowledge was able to overcome successfully by producing a result of MAE 0.0071

V. Conclusion

A. Limitations

The numbers associated with the paper are confined to USDA SR legacy data. The paper did not dig further into technical aspects of nutritional data. It has limited its scope to the design of model for compression of data and extraction of features for diet planner. The feasibility has to be studied before deploying the model on any other dataset. Whether extending the model on to large dataset would require in change in dimension is not studied.

B. Future scope

This architecture can be conveniently extended to similar type of data. Any extended application intended would benefit from pre-processed data and would reduce model complexity. Further research can be done in efficient representation of the food dataset for meal planner or any other application.

Acknowledgment

I am thankful for Data Science professors of GEETAM Deemed University, Hyderabad for their valuable insights. I also thank Department of Computer Science for support that lasted throughout the work.

References

[1] F. Barreto, S. Yadav, S. Patnaik, and J. Sarvaiya, "SIFT Features for Deep and Variational Autoencoders: A Performance Comparison," 2020 2nd International
[2] N. Mansouri and Z. Lachiri, "Laughter synthesis: A comparison between Variational autocoder and Autoencoder," 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 2020, pp. 1-6, doi: 10.1109/ATSIP49331.2020.9231607.
[3] D. P. Kingma and M. Welling, "An introduction to variational autoencoders", Found. Trends Mach. Learn., vol. 12, no. 4, pp. 307-392, 2019
[4] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes", Proc. Int. Conf. Learn. Represent., pp. 1-5, Dec. 2014
[5] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization", Proc. ICCV, pp. 1501-1510, Oct. 2017
[6] J.L. Ba, J.R. Kiros and G.E. Hinton, Layer normalization, 2016, [online] Available
[7] D. Ke, W. Yao, R. Hu, L. Huang, and Q. Luo and W. Shu, "A New Spoken Language Teaching Tech: Combining Multi-attention and AdaIN for One-shot Cross Language Voice Conversion," 2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP), Singapore, Singapore, 2022, pp. 101-104, doi: 10.1109/ISCSLP57327.2022.10038137.
[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", ICML, 2015
[9] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1520-1528, doi: 10.1109/ICCV.2015.178.
[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", science, vol. 313, pp. 504-507, 2006.
[11] M. A. Yílmaz, O. Kelesş, H. Güven, A. M. Tekalp, J. Malik and S. Kíranyaz, "Self-Organized Variational Autoencoders (Self-Vae) For Learned Image Compression," 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 2021, pp. 3732-3736, doi: 10.1109/ICIP42928.2021.9506041.
[12] Lucas Theis, Wenzhe Shi, Andrew Cunningham& Ferenc Huszar, LOSSY IMAGE COMPRESSION WITH COMPRESSIVE AUTOENCODERS, conference paper, ICLR 2017, arXiv:1703.00395
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS(2012)

[14] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In ECCV. Springer, 2014
[15] H Zhang, C Wu, Z Zhang et al., "ResNeSt: Split-Attention Networks[J]", 2020.
[16] Q Wang, B Wu, P Zhu et al., "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks[C]", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020
[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In CVPR, 2018.
[18] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression", Int. Conf. Learning Representations (ICLR), 2017
[19] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning Convolutional Networks for Content-Weighted Image Compression," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 3214-3223, doi: 10.1109/CVPR.2018.00339.
[20] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks", European Conference on Computer Vision, pp. 525-542, 2016
[21] O. Rippel and L. Bourdev, "Real-time adaptive image compression", International Conference on Machine Learning, pp. 2922-2930, 2017.
[22] Goodfellow, I., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Yoshua Bengio Generative adversarial nets. In NIPS, pp. 2672–2680, 2014.
[23] D. Ke, W. Yao, R. Hu, L. Huang, Q. Luo and W. Shu, "A New Spoken Language Teaching Tech: Combining Multi-attention and AdaIN for One-shot Cross Language Voice Conversion," 2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP), Singapore, Singapore, 2022, pp. 101-104, doi: 10.1109/ISCSLP57327.2022.10038137.
[24] F. Zhang, H. Gao and Y. Lai, "Detail-Preserving CycleGAN-AdaIN Framework for Image-to-Ink Painting Translation," in IEEE Access, vol. 8, pp. 132002-132011, 2020, doi: 10.1109/ACCESS.2020.3009470.
[25] Z. Zhou, J. Mo and Y. Shi, "Data imputation and dimensionality reduction using deep learning in industrial data," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 2017, pp. 2329-2333, doi: 10.1109/CompComm.2017.8322951.