Performance Analysis of Semi-supervised Learning in the Small-data Regime using VAEs

Venkata Varunbabu Mannam
vmannam@nd.edu
University of Notre Dame
Notre Dame, Indiana

Arman Kazemi
akazemi@nd.edu
University of Notre Dame
Notre Dame, Indiana

ABSTRACT

Extracting large amounts of data from biological samples is not feasible due to radiation issues and image processing in the small-data regime is one of the critical challenges when working with limited amount of data. In this work, we applied an existing algorithm named Variational Auto Encoder (VAE) that pre-trains a latent space representation of the data that captures the features in a lower-dimension for the small-data regime input. The latent space representation will be fine-tuned and its weights will be fixed. The latent space will be used as a segment of the neural network that can be used for classification. Here we will present the performance analysis of the VAE algorithm with various latent space sizes in the semi-supervised learning using the CIFAR-10 dataset.

KEYWORDS

VAE, CIFAR-10, Small Data

ACM Reference Format:
Venkata Varunbabu Mannam and Arman Kazemi. 2018. Performance Analysis of Semi-supervised Learning in the Small-data Regime using VAEs. In *CSE 40625/60625 '19: Course Project Workshop, April, 2019, Notre Dame, IN*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Artificial neural networks (ANNs), specifically Convolutional Neural Networks (CNNS), have become popular due to their success in image classification, feature extraction and object recognition and detection [4] in the recent years. CNNs leverage the huge amount of labeled data available to train networks that outperform humans in image recognition tasks. However, in the small-data regime, the accuracy of trained networks using a limited number of labeled samples is low [3]. This is a typical case when working with biological samples where exposure to radiation (in order to capture an image) is detrimental to the well-being of the sample. More images can be derived from the initial data by some augmentation methods, but it is unhelpful due to lack of labeled images.

To address this problem, there exists a framework called Auto Encoder (AE [1]) that uses all the input data, labeled and unlabeled, to train a low-dimensional embedding. AE is a neural network that takes unlabeled images as the input and regards the input itself as the label. As illustrated in Figure 1, AE is comprised of two parts: the encoder and the decoder. The encoder part tries to embed the features in a latent space that can extract the features of the original image and the decoder tries to restore the image to the original image. The process called pre-training trains the weights for both encoder and decoder parts. Once trained, the encoder part of the AE will be a representation of all the labeled and unlabeled data. This increases the amount of usable information from all of the images.

Traditional semi-supervised models consists of pretraining using Restricted Boltzmann machine (RBM)[6] or Gaussian-Restricted Boltzmann machine (G-RBM)[6]. RBM is an energy based model that is represented using an undirected graph containing a layer of observable variables and a single layer of latent variables (similar to hidden units in a multi-layer perceptron) [4, 9]. This energy based model was first introduced in 1980's [11] and have been implemented using diverse datasets including image [6] and medical data [10]. Hinton and Salakhutdinov [5] showed that RBMs can be stacked and trained in a greedy manner. This deep learning model, Deep belief networks(DBN), that utilizes RBM as the learning model has been implemented on various unsupervised and supervised learning problems. Later, Bengio et. al. [2] showed that the pre-trained undirected graphical model in semi-supervised setting performs well with deep architectures. However, the challenge working with RBM is that it is constructed using sigmoid functions as the activation function between the input and the hidden layer. As we are aware that the major drawback of using sigmoid activation function is the vanishing gradient problem. Hence, in this work, we pre-train the model using Variational Auto Encoder (VAE [1]).

After pre-training the encoder is able to observe similar images to the training images and extract the valuable features from it in a lower dimension than the initial image. Now it is possible to couple the encoder with a small neural network and train that network for classification tasks. The present work is similar to reinforcement learning where the model is trained with one dataset and uses the feature extraction part of that model to train another model for a different dataset. It is important to mention that the encoder weights are fixed and can't be changed. It is only the small neural network that will be trained. The input to this network is the small set of labeled data. This is called fine-tuning.

In this project we will implement VAE that tries to capture not only the compressed representation of the images, but also the parameters of a probability distribution representing the data. We will examine the effect of different size of the latent spaces and how it affects the accuracy of the model. Later, we will analyze the performance of the semi-supervised model with the optimum latent space.

CSE 40625/60625 '19, April, 2019, Notre Dame, IN

2 BACKGROUND

In this section we enumerate the basic details about AE and VAE.

2.1 Auto Encoder

An autoencoder is a type of ANN used to learn efficient data encoding in an unsupervised manner. The aim of an autoencoder is to learn a representation of a set of data, typically for dimensionality reduction, by training the network to ignore signal noise. Along with the reduction side, a reconstructing side is learned, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input. An autoencoder always consists of two parts, the encoder and the decoder, which can be defined as transitions ϕ and ψ such that [1]:

$$\phi : X \rightarrow F \tag{1}$$

$$\psi : F \rightarrow Y \tag{2}$$

$$\phi, \psi = argmin_{\phi,\psi} ||X - (\psi o \phi)(X)^2|| \tag{3}$$

where the given input is X and the predicted output is Y. If the feature space F has lower dimensionality than the input space X, then the feature vector $\phi(x)$ can be regarded as a compressed representation of the input X.

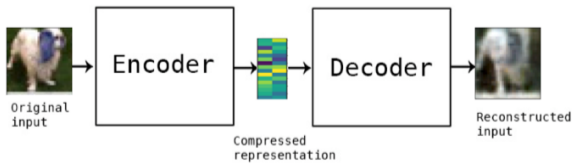

Figure 1: Autoencoder block diagram

2.2 VAE

To extend the idea used in the AE, there is a variation called VAE which uses the "KL-divergence" between the predicted probability distribution function and actual posterior distribution in the latent space whereas traditional AEs try to find the accurate mapping functions in the encoder (between input and latent space) as well as in the decoder (between the latent space and output). Using VAE, we generate a large dataset by adding the noise in the latent space which is similar to input data augmentation (adding noise to images to increase the number of examples in the input dataset).

VAE uses a variational approach for latent representation learning, which results in an additional loss component. It assumes that the data is generated by a directed graphical model $p(\mathbf{X}|\mathbf{Z})$ and that the encoder is learning an approximation $q_\phi(\mathbf{Z}|\mathbf{X})$ of the posterior distribution $p_\theta(\mathbf{X}|\mathbf{Z})$ where ϕ and θ denote the parameters of the encoder (recognition model) and decoder (generative model) respectively. We can write the conditional or posterior distribution

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})} \tag{4}$$

The denominator of above equation is the marginal distribution of the observations and is calculated by marginalizing out the latent variables from the joint distribution, i.e.

$$p(x) = \int_z p(z,x)dz \tag{5}$$

In many cases of interest this integral is not available in closed form or is intractable (requires exponential time to compute). Hence, we consider variational approximation as follows: consider a tractable distribution q(z). The goal is to find the best approximation, e.g., the one that satisfies the following optimization problem:

$$Minimize : D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})] \tag{6}$$

Therefore, the objective of the variational autoencoder in this case has the following form:

$$\mathcal{L}(\phi, \theta, \mathbf{x}) = D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\right) - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left(\log p_\theta(\mathbf{x}|\mathbf{z})\right) \tag{7}$$

where D_{KL} stands for the KL-divergence. In the VAE, the principle is to minimize the loss between input and the restored image along with the loss generated by the latent space to represent the features in the input images.

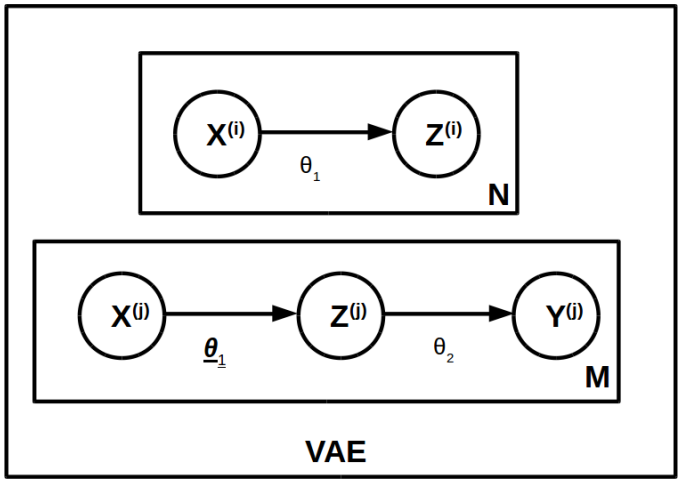

Figure 2: Network architecture: pre-training with VAE(first row) In the second row, parameter θ is underlined and bold to indicate that these parameters are freezed when the fine-tuning the network

3 METHODOLOGY

Consider a surrogate model $y = f(x, \theta)$ which is trained using limited simulation data $\mathcal{D} = \{x^i, y^i\}_{i=1}^N, \{x^j\}_{j=N+1}^D$. Where the input data, $x^i \in \mathbb{R}^{d_x \times H \times W}$ is the input from CIFAR-10 dataset. Here H and W are the height and width respectively and d_x is the number of dimensions for the input x at one location. x^j is the additional data utilized for pretraining the model. $y^i \in \mathbb{R}^1$ is the classified result. θ is the model parameter and N is the total number of training data utilized during fine-tuning and D is the total number of data utilized for pre-training. In semi-supervised model, we pre-train the model with the input data $\mathbb{R}^{d_x \times H \times W}$ and then perform image classification problem $\mathbb{R}^{d_x \times H \times W} \rightarrow \mathbb{R}^1$. For both the pre-training and fine tuning, we used stochastic gradient descent with Adam optimizer to update the network weights and biases. The

simulation was performed using Pytorch machine learning package in Python.

3.1 VAE pre-training

We implemented dense-net [7],[13] version of VAE for the pre-training part. Dense-net contains the encoder and decoder blocks along with the dense-layer that has the simple and complex features.

3.2 VAE fine-tuning

We implemented a simple fully connected layer to classify the input images on the CIFAR-10 [8] data. This is due to the expectation that the latent space is smaller (either 4×4 or 8×8) (image size is 32×32). If the number of channels are large at the latent space, we will add more fully connected layers for classification of images.

4 DATA

We perform the simulations on the CIFAR-10 dataset with ten image classes with three input channels ($C = 3$) of size 32×32 (W $\times$ H). CIFAR-10 dataset has 50000 training images and 10000 test images.

5 RESULTS

In this section we enumerate the results obtained using CIFAR-10 data. We consider the following latent dimensions: 6400, 10,000 and 14,400. In order to evaluate the performance of the model for above three latent space, we consider the distribution estimated for the values at various pixel location.

5.1 Pre-training

For the results presented in this section, we have a dataset with 50,000 $\{x^k\}_{k=1}^{50000}$ examples for pre-training and the test set consist of 10000, $\{x^k\}_{k=1}^{10000}$ examples. Adam optimizer was used for training 100 epochs, with learning rate of 1e-4 and a plateau scheduler on the test RMSE. Batch size is always smaller than the number of training data. In this work, a batch size of 16 for pre-training was used. Weight decay was set to 1e-3 for pre-training.

We consider Equation: 7 loss function to evaluate the trained model on test data and also to monitor the convergence.

From figure 1, we observe that the solution is converged after 50 epochs and most importantly the loss for the three latent spaces is similar.

From figure 4, we observe that even when the latent size is small (Batch $\times$ 100 (channels) $\times$ 8 $\times$ 8) and (Batch $\times$ 100 (channels) $\times$ 10 $\times$ 10) the reconstructed density estimate is close to actual input data. The PDF with the latent size 10000 is closer to the actual input and also 6400 & 14400 latent space. Since, all the latent spaces yield the similar outputs,we fine-tune and compare the classification accuracy in the next section.

5.2 Fine-tuning

In this section we freeze the parameters (weights and bias) used in the pre-training stage and fine tune the parameters (weights and bias) in the classification network. For this problem, we consider small data from the given CIFAR-10 dataset and use fully connected layers to perform classification. Ccross entropy loss function is

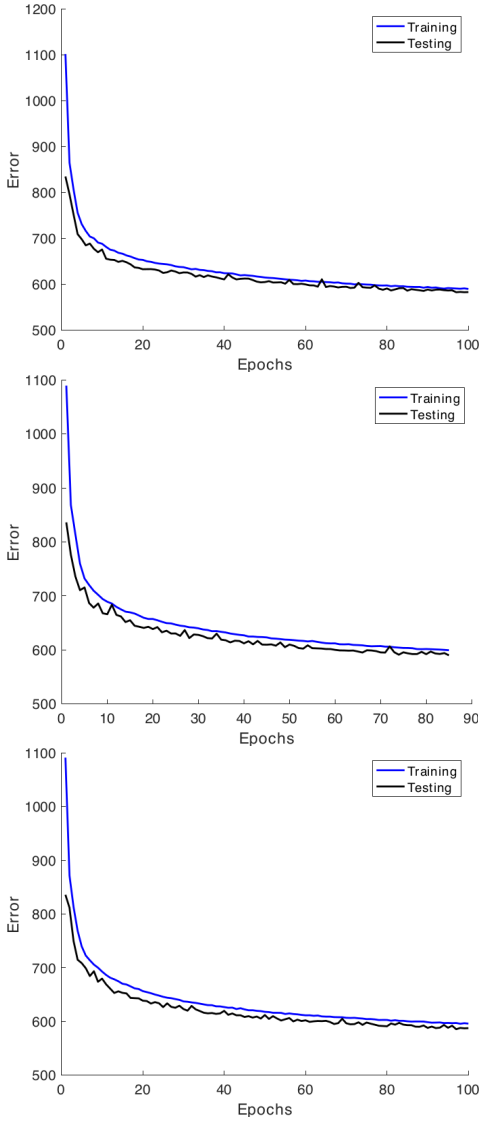

Figure 3: Error v/s epoch for 6400 latent space (top), Error v/s epoch for 10000 latent space (middle) and Error v/s epoch for 14400 latent space (bottom)

commonly used for all classification problems, we implemented cross entropy to measures the performance of a classification mode. From figure 5, we observe the latent space 10000 yields better accuracy than other two latent spaces (6400 and 14400). The smaller the latent space the lower the test accuracy, this is due to insufficient features to classify the data. Also, for the large latent space, the test accuracy is low, this is due to model complexity [4].

6 CONCLUSIONS AND FUTURE WORK

The present document outlines the development of surrogate model for semi-supervised problem. In this work, we have implemented VAE as a pre-training model and a feed forward deep learning model for the classification. The results obtained for differently

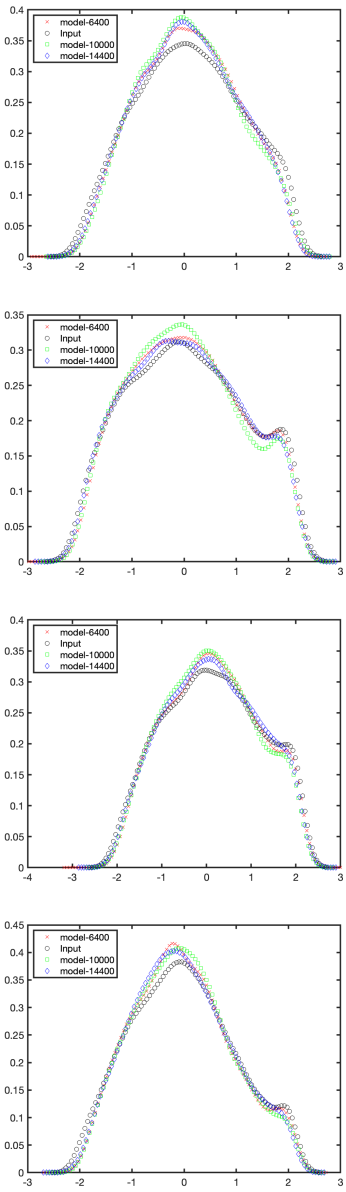

Figure 4: Distribution estimate for the values at various location of the square domain for 6400, 10000 and 14400 latent space

sized latent spaces are presented. It was observed that there is a slight improvement in the test accuracy when the latent space is 10000 in comparison with latent space of 6400 and 14400.

For future work, a Bayesian approach can be explored. Due to a limited amount of data, it is necessary to model appropriate surrogate, since it is important to quantify the epistemic uncertainty induced by limited data [12], [14] and hence, a Bayesian probabilistic approach is a natural way of addressing this challenge.

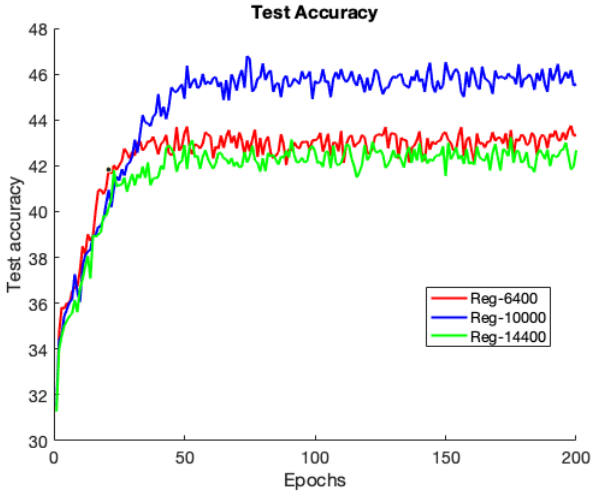

Figure 5: Fine tuning results with three different latent space models

REFERENCES

[1] Autoencoder. 2019. Autoencoder — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Autoencoder

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*. 153–160.

[3] Nitesh V Chawla et al. 2005. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research* 23 (2005), 331–366.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[5] G Hinton and R Salakhutdinov. 2012. An efficient learning procedure for deep Boltzmann machines. *Neural Computation* 24, 8 (2012), 1967–2006.

[6] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[8] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.

[9] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.

[10] Tu Dinh Nguyen, Truyen Tran, Dinh Phung, and Svetha Venkatesh. 2013. Latent patient profile modelling and applications with mixed-variate restricted Boltzmann machine. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 123–135.

[11] Paul Smolensky. 1986. *Information processing in dynamical systems: Foundations of harmony theory*. Technical Report. Colorado Univ at Boulder Dept of Computer Science.

[12] Rohit K Tripathy and Ilias Bilionis. 2018. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *J. Comput. Phys.* 375 (2018), 565–588.

[13] Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. 2018. A Poisson-Gaussian Denoising Dataset with Real Fluorescence Microscopy Images. *arXiv preprint arXiv:1812.10366* (2018).

[14] Yinhao Zhu and Nicholas Zabaras. 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.* 366 (2018), 415–447.