Semi-supervised Learning using Variational Autoencoder - A Cluster based Approach

No Author Given

No Institute Given

Paper ID 53

Abstract. The successful application of deep neural networks for solving complex tasks like image classification, object detection and segmentation depends critically on the availability of large number of labelled training samples. To achieve good generalization for a reasonably complex model with about 60 million parameters, as in alexnet, one needs about one million labelled training samples. In almost all practical applications, like natural image classification and segmentation, plenty of unlabelled samples are available but labelling these samples is a tedious manual task. We introduce a novel mechanism to automatically label all the samples in an unlabelled dataset. Starting with completely unlabelled dataset, an iterative algorithm incrementally assigns labels along with a confidence to all training samples. During each iteration, 10-30 new representative samples are generated in a latent space learned using a variational autoencoder and labels for these samples are obtained from a human expert. The proposed idea is demonstrated on MNIST dataset without using the labels provided in the dataset. At regular intervals of training, the low dimensional latent vectors are clustered and only cluster centers are annotated. The manual labels of cluster centers are propagated to other samples in the cluster based on the distance and a confidence function. The loss function in successive training is modified to incorporate the manual information provided. We run multiple experiments with different choices of clustering algorithm, confidence function and distance metric and compare the results. With GMM clustering, best classification accuracy of 93.9% was obtained on MNIST test images after 5 iterations.

Keywords: Semi-supervised Learning · Variational Autoencoder · Active Learning · Clustering

1 Introduction

Deep neural networks have been successfully applied for performing machine learning tasks like classification [1–3], object detection [7, 6] and segmentation [4, 5] in images and videos. Recently, multi-layer deep neural networks have been identified as effective choices for generative models also, Variational Autoencoder (VAE) [11] and Generative Adversarial networks (GAN) [12] being the most common examples. One of the key tenets on which all these models work is based

upon the ability to learn a complex, parameterized unknown function using back propagation. When trained with a huge amount of data for long enough durations and with appropriate regularization techniques, these networks are capable of learning functions that generalize well. However, all these models suffer from the following drawbacks:

1. Lack of explainability.
2. Need for huge amount of labelled data for supervised tasks like classification, object detection and segmentation.
3. Need for large computing power to train complex models with millions of parameters.

In this work, our key focus is on how to train a neural network faster, and with less number of annotated samples, by augmenting the training process with human feedback at regular intervals. Instead of annotating all individual samples in the training set, we propose a novel mechanism where only a few representative samples (10-30) are annotated and the label of these samples are propagated to other training samples which are closer to the labelled sample in the latent space. It is very common to use generative models, like Variational Autoencoder (VAE) [11] and Generative Adversarial Networks [12] in order to learn a low dimensional latent representation of data. It is reasonable to assume, as validated by our experimental results, that the samples which are closer in latent space have the same label. In this work, we augment a popular generative model, Variational Autoencoder, by adding a classification loss so that the latent representations of samples from different classes becomes more separable.

Our work can also be looked at as a novel active learning framework using deep neural networks. Active learning algorithms iteratively select (or generate) samples for labelling and these labelled samples are used to improve the model performance. The goal is to obtain sufficiently good performance of the trained model using as few labelled samples as possible.

We perform various experiments with different clustering algorithms, k-means and Gaussian mixture model (GMM), and also with different distance metrics (Mahalanobis and Euclidean distance) and provide a detailed comparison of results.

The major contributions of this paper are

1. We propose a novel active learning framework where a deep learning model incrementally learns to perform a task like image classification, while at the same time learning a low dimensional representation for the input data.
2. Our approach starts with a completely unlabelled dataset and iteratively finds labels for the entire dataset by labelling only 10-30 representative samples during each iteration.
3. Our experiments show that the distribution of latent vectors becomes multimodal, and the multiple modes become more separable with the addition of a classification loss to the β-VAE loss function [8].

2 Related Work

We are not the first to propose active learning using deep generative modelling. Many of the existing work on active learning are based on either query-synthesizing or pool-based methods. Where as in query synthesizing new informative samples are generated using generative models, pool-based methods [15, 16] uses various sampling strategies to select a set of samples for labelling. Many query-synthesizing methods use adversarial networks [13, 14] to generate new samples for labelling. Our method is a combination of both query synthesizing and pool-based approach as we are generating new representative samples using clustering while at the same time selecting unlabelled samples from selected bad performing clusters. In their work, Samarth et al. [10] uses Variational Autoencoder to learn a latent space along with an adversarial network to select samples for labelling from an unlabelled pool.

Our generative model is similar to the model described in [9]. In [9], a low dimensional embedding of unlabelled data is learned using a deep generative model. The parameters of the generative model are optimized using variational lower bound and then the latent vectors in lower dimension are classified using SVM. However, our work differs from [9] in respect that we introduce a novel mechanism for labelling by clustering the latent vectors. Also instead of using a separate SVM classifier, we augment the VAE by adding additional classification layer.

3 Proposed Method

3.1 Dataset

We demonstrate the proposed approach by performing experiments on MNIST dataset. We split the 50,000 training samples in MNIST dataset into train (70%) and validate (30%). Performance on the validation set is used for hyper parameter tuning.

3.2 Neural Network architecture

Figure 1 shows the architecture of the proposed model. The model follows the usual encoder-decoder architecture found in VAE. The encoder consists of one down-sampling convolutional layer followed by two dense layers. The decoder is made up of two dense layers and one up-sampling de-convolution layer. An additional softmax classification layer was added to classify the latent vectors into one of the 10 class labels.

3.3 Semi-supervised loss function and training

The variational autoencoder is initially trained until the reconstruction loss converges using the beta-VAE loss function [8] given below.

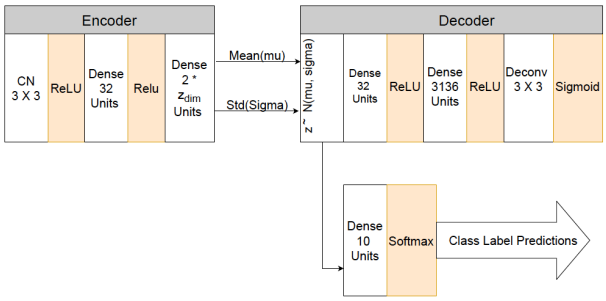

Fig. 1: Architecture of the proposed model. The VAE architecture, comprising of an encoder and decoder, is augmented with a softmax classification layer

$$L_{VAE} = -\sum_{i,j}(x_{ij}^n \ln \hat{x}_{ij}^n + (1 - x_{ij}^n) \ln(1 - \hat{x}_{ij}^n)) + \beta KLD(p(z), N(0, I)) \quad (1)$$

where x_{ij} is the pixel value at position (i, j) of the input image, $\hat{x}_{ij}$ is the pixel value of reconstructed image, $p(z)$ is the prior distribution of latent vectors, $N(0, I)$ is the standard multivariate normal distribution and KLD() denotes KL divergence. We use binary cross entropy as reconstruction loss for MNIST images, since the output activation function is sigmoid and MNIST images are treated as binary images in our experiments.

Once a reasonably good latent representation is learned using the loss function mentioned above, the latent vectors of the training samples were clustered into $k = 10$ clusters. The cluster centers were decoded using the decoder and the resulting images of cluster centers were manually given a label and a confidence. Figure 2a shows the reconstructed images of cluster centers after unsupervised training.

If the cluster center for a cluster does not correspond to any valid digit, all the samples in that cluster are again clustered into k sub-clusters and centers of the sub-clusters are labelled. The process can be continued to form clusters at multiple levels based on the available manual annotation budget. The labels assigned to the cluster center is then propagated to all other samples in the cluster using the following strategy

1. Each sample in the cluster is assigned with the same label as the cluster center.
2. Each sample is also given a confidence based on its distance from cluster center and a manually assigned confidence in the range of [0,1]. The overall confidence of training sample x^n is computed as

$$w_n = p_c f(d_n)$$

where d_n is the distance of the sample from its cluster center, p_c is the confidence manually assigned to the cluster center and $f : d \mapsto [0, 1]$ is a monotonically decreasing function that maps distance to a confidence value in unit interval [0,1].

(a) With unsupervised training (b) Semi-supervised - 5 iterations

Fig. 2: Reconstructed images of cluster centers

We performed experiments with different choices for the distance metric (Euclidean and Mahalanobis) and confidence decay function- exponential (Equation 2) vs Gaussian (Equation 3).

$$w_n = p_c e^{-ad_n} \tag{2}$$

$$w_n = p_c e^{-ad_n^2} \tag{3}$$

where a is a hyper parameter determining how fast the confidence decreases with the distance from cluster center.

Training is continued for more epochs using the modified loss function, given below, that incorporates the manual labels and confidence into account.

$$L = L_{VAE} - \gamma \sum_{k=0}^{K} w_n y_{nk} \ln(\hat{y}_{nk}) \tag{4}$$

where y_{nk} is the one-hot encoded label and $\hat{y}_{nk}$ is the predicted softmax probability. The new term that we added to the loss is the weighted multi-class cross entropy loss for classification task.

We performed experiments using GMM and k-means clustering. With GMM, we directly used the product of posterior probability of the sample and cluster center confidence p_c as the overall sample confidence.

4 Results and Discussions

In order to choose the optimum hyper-parameters like clustering algorithm, distance metric and confidence decay function, we ran experiments with different combinations of hyper parameters and the results are compared in Figures 3. It is important to note that, with k-means performance is better when Gaussian confidence mapping function is used as opposed to exponential function. This is expected as exponential function decreases at a faster rate near zero. Hence, the confidence for samples closer to the cluster center decreases at a faster rate as it moves away from cluster center. Whereas with Gaussian confidence function, since the gradient of the curve is close to zero near cluster center, samples closer to the cluster center will have almost the same confidence as the cluster center confidence p_c.

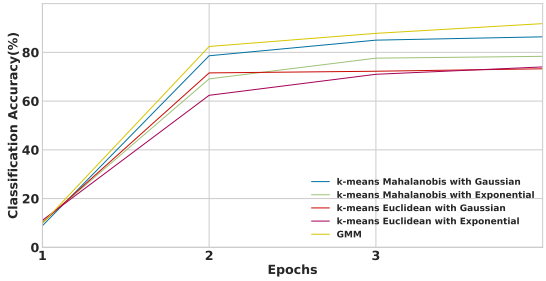

Fig. 3: Comparison of classification accuracy with different choices of hyper-parameters distance metric, confidence function and clustering algorithms

With k-means clustering, using Mahalanobis distance instead of Euclidean distance gives a better classification accuracy during initial iterations. From the latent space distribution shown in Figure 5b and Figure 6b, it is evident that the clusters have different variance along different directions which is captured well in Mahalanobis distance. For the same reason, GMM performs better than k-means clustering. Note that with GMM, posterior probability of samples is used as confidence. Hence, there is no need for confidence function to map distance to confidence.

The reconstructed images of cluster centers shown in Figure 2a and Figure 2b shows that as training progresses, the clusters formed in latent space correspond to different classes in MNIST dataset. In Figure 2a some of the cluster centers do not correspond to any valid digit, and some digits like 4 are missing. This is happening because the latent vectors for some classes, like 4 and 9 for example, were very similar and got clustered into the same cluster. However, after a few iterations of semi-supervised training, the latent vectors for symbol 9 and 4 are getting clustered into different clusters. This is also evident from the t-SNE visualization of latent space shown in Figure 5a and Figure 5b.

Figure 6 shows the 2-d t-SNE visualization of the latent vector distribution for the entire training set with unsupervised and semi-supervised training. It is observed that the distribution of samples from different classes becomes more separable with semi-supervised learning.

Figure 4 shows the classification accuracy on test dataset. Initial 10 epochs were trained in unsupervised mode and classification loss was added after that. After 5 iterations of semi-supervised training, we were able to get a classification accuracy of 93.9% on MNIST test dataset.

5 Conclusion

In this paper, we propose a novel deep learning based active learning framework for solving complex tasks like image classification using clusters formed in the

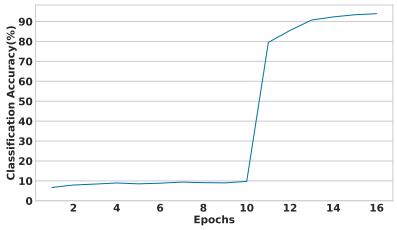

Fig. 4: Classification accuracy on MNIST test dataset with 5 iterations of semi-supervised training. Classification loss was added after 10 epochs

(a) With unsupervised training (b) Semi-supervised - 5 iterations

Fig. 5: Latent space distribution, with t-SNE, for samples from class 4 and 9

low dimensional latent space learned by a variational autoencoder. Our approach is demonstrated by classifying MNIST images without using labels provided in the dataset. We performed multiple experiments with different choices for hyper parameters like clustering algorithm, distance metric and confidence function. Our experiments shows that GMM, when used as a clustering algorithm, gives the best classification accuracy of 93.9% on MNSIT test dataset after 5 iterations. It is observed, using 2-d t-SNE visualization, that semantically valid and separable clusters are formed as a result of semi-supervised learning.

(a) With unsupervised training (b) Semi-supervised - 5 iterations

Fig. 6: Latent space distribution (with t-SNE) for the entire training set. With supervised training, semantically valid and well separated clusters are formed

References

1. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25 (2012): 1097-1105
2. Simonyan, Karen, and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
3. He, Kaiming, et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
4. Chen, Liang-Chieh, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 40.4 (2017): 834-848.
5. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.
6. Redmon, Joseph, et al. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
7. Ren, Shaoqing, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497 (2015).
8. Higgins, Irina, et al. beta-vae: Learning basic visual concepts with a constrained variational framework. (2016)
9. Kingma, Diederik P., et al. Semi-supervised learning with deep generative models. arXiv preprint arXiv:1406.5298 (2014)
10. Sinha, Samarth, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019
11. Kingma, Diederik P., and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).
12. Goodfellow, Ian J., et al. Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014).
13. Mahapatra, Dwarikanath, et al. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2018.
14. Mayer, Christoph, and Radu Timofte. Adversarial sampling for active learning. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020.
15. Wang, Keze, et al. Cost-effective active learning for deep image classification. IEEE Transactions on Circuits and Systems for Video Technology 27.12 (2016): 2591-2600.
16. Beluch, William H., et al. The power of ensembles for active learning in image classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.