Image Classification using Unlabelled Data - A Semisupervised Approach using Variational Autoencoder and Topdown Hierarchical Clustering

Abstract

The success of deep learning for solving complex tasks like image classification, segmentation, speech and natural language processing, has caused wide-spread interest in the machine learning community to focus on developing models and representations that are more explainable, and generalize better. In this work, we propose an approach for using representations learned by unsupervised generative learning for solving tasks like image classification at a reduced manual annotation cost. Our method is an alternate paradigm for supervised learning. In existing supervised learning methods, all samples are labelled prior to start of training whereas we propose an active learning mechanism where manual hints are given at regular intervals during training. We demonstrate the proposed idea by training a variational autoencoder on MNIST data set. After every epoch of training, the low dimensional latent vectors are clustered and cluster centers are annotated. The loss function in successive training is modified to incorporate the manual annotation. In addition to achieving classification of digits, the approach also results in improved reconstruction accuracy and more regular features of autoencoder. Our network architecture and cost function look similar to multi task learning with hard parameter sharing. However, unlike other multi task learning models, our main goal is to reduce the manual annotation required for supervised tasks like image classification and segmentation. The approach is demonstrated by classifying the images in MNIST data without using the label provided in the dataset. We obtain a classification accuracy of 93.47% on MNIST test images with 5 epochs of training.

1 INTRODUCTION

Classifying images is one of the first use cases proven to give good results using deep neural networks. Recently, there has been a lot of work on generative models like variational autoencoder(VAE)Kingma and Welling [2013] and generative adversarial network(GAN) Goodfellow et al. [2014] on using deep neural network for learning distribution of high dimensional data. In this work, we propose a method whereby a generative model like VAE can easily be converted into a classification model which is currently solved by a supervised classification method. Note that the existing deep learning approaches for classification need a lot of annotated training data and enormous training time on GPUKrizhevsky et al. [2012]Simonyan and Zisserman [2014]He et al. [2016]. The approach proposed in this paper needs very less amount of manual annotation (10-16 samples in case of MNIST dataset) and less computing resources. We demonstrate our claim by building a classification model for MNIST dataset using all traning images in an unsupervised way and annotations are done only for selected representative samples. The proposed approach augments a variational autoencoder with a classification layer the loss component of which is tuned using manually annotated samples at regular training intervals.

A generative model learns the distribution of data $p(x_{ij})$. A new image of a given digit can be generated by sampling from this distribution. In the case of an image, this is usually a complex distribution in a high dimensional space of dimension $W \times H$. Such a distribution in the original high dimensional space is not of much use as it is not easy to visualize and contains too many minute details. Specifically, the properties of interest, like line thickness in case of handwritten digit, are not explicitly evident from such a distribution. All generative models essentially solve this problem by transforming the original image into a much low dimensional latent space, which we denote by Z. For each image $x^n \in X$, there exists a latent vector $z^n \in Z$ where z^n is of dimension z_{dim}. The dimension of latent space z_{dim} is

Submitted to the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021). **To be used for reviewing only**.

much less compared to the original image dimension. However, one of the major issues with these trained models is that the concepts represented by latent dimensions need not make any sense and hence they lack one of the much needed properties: the model explainability.

In this work, we propose an active learning framework using variational autoencoder for the task of image classification. Variational autoencoder is used for learning a low dimensional latent representation which is used for classification task.

Our model is similar to multi task learning since the loss term has both reconstruction and classification losses. However, unlike most other multi-task models, see Ruder [2017] and Crawshaw [2020] for a complete review of existing multi-task learning techniques, our approach combines different types of machine learning tasks like classification, generative modeling and representation learning. The approach described can easily be extended to even more complex tasks like semantic segmentation.

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. We show that compared to existing deep active learning frameworks our approach requires very less number of training samples and also learns a latent representation and probability distribution in the latent space from which new data samples can be drawn easily
3. The proposed approach reduces the manual annotation task and can be trained quickly on a CPU, as opposed to complex models that usually needs several hours of training on GPU

The rest of the paper is organized as follows. Section 2 provides an overview of existing techniques of multi-task learning and active learning. Section 3 contains details of network architecture, loss function and training process. A detailed analysis of results of experiments are provided in Section 4. Finally, we conclude our findings in Section 5

2 RELATED WORK

Most of the approaches for implementing active learning using deep learning models is by query sampling. In query sampling, during each iteration of training, an algorithm or a machine learning model selects a few samples and get it annotated by a domain expertSinha et al. [2019]Sener and Savarese [2017]. The sampling technique should be such that the number of samples to be annotated is as minimum as possible without compromising much on model accuracy.

Multi-task learning where multiple related tasks, from a single domain, like combining facial landmark detection with head pose detection and facial attribute detection Zhang et al. [2014] have helped in increasing robustness in detection with reduced model complexity. The basic tenet of multi-task learning is that the model prefers a hypothesis that explains more than one task and usually this results in solutions that generalize better Ruder [2017]. While training a network for more than one task, other tasks can provide additional evidence for relevance or irrelevance of features. Liu et al. introduces task specific attention modules attached to a shared convolutional pool along with a multi-task loss function to train a single network for multiple tasks like semantic segmentation, depth estimation and detection of surface normal Liu et al. [2019].

Our approach is similar to hard parameter sharing as in Zhang et al. [2014] Dai et al. [2016], but differs in respect that we are trying to solve a task like image classification, which is traditionally addressed as a supervised task requiring large amount of manually annotated data, using information obtained from unsupervised representation learning. Our approach results in reduced manual annotation and less number of training epochs along with other benefits of multi-task learning such as learning a generic representation that helps in multiple tasks.

3 PROPOSED METHOD

3.1 PROBLEM FORMULATION

Consider a grey-scale image, I_n $1 \leq n \leq N$, of height H and width W. The grey value at a location (i, j) of the image is denoted as $x_{ij}^n \in [0, 1]$ where $1 \leq i \leq H$ and $1 \leq j \leq W$. In our experiments, we use MNIST in which $N = 59872, H = 28, W = 28$. During the training phase, we did not use the labels of the training set. The labels of the validation set were used to compute the classification and reconstruction accuracy.

3.2 DATASET

We used MNIST datasetLeCun and Cortes [2010] to demonstrate the proposed approach. The primary reason for selecting MNIST dataset is to reduce the manual annotation cost required for identifying the reconstructed images. Images in the MNIST training set were split into training and validation sets with stratified sampling on the label column. The validation set, which consists of 128 images, were used to compute the reconstruction accuracy of the autoencoder. Rest of the 59872 images were used to train the model. The images were normalized before feeding to the input of the network so that the 256 grey values are converted into real numbers in the unit interval [0,1].

3.3 NEURAL NETWORK ARCHITECTURE AND LOSS FUNCTION

Figure 1 shows the architecture of the proposed model. We used a variational autoencoderKingma and Welling [2013], with 4 layers of encoder and 4 layers in the decoder, augmented by adding a $K-$node softmax classification layer in order to classify the latent vector z into one of K different classes. The encoder output has linear activation function so that the image is encoded into a latent vector, z of dimension z_{dim}, each dimension taking continuous values. The decoder output activation is sigmoid so that most of the reconstructed pixel values are concentrated around 0 or 1 by design. Initially, for the first few epochs, the network is trained only using the autoencoder loss function and hence labels are not required. The loss function used for training during initial epochs is

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

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 probability density function of latent vectors, $N(0,I)$ is the standard multivariate normal distribution of dimension z_{dim} and KLD() deontes KL divergence. We used $\beta = 5$ as it gave the best compromise between reconstruction quality and KL divergence. After a few epochs of unsupervised training, the latent vectors corresponding to the training images are clustered into k clusters using k-means algorithm. Optimum value of the number of clusters k were determined by elbow method. The cluster centers were decoded using the decoder part of VAE and the resulting images corresponding to cluster centers were manually given a label and a confidence. if the cluster center for a cluster does not correspond to any valid digit image, that cluster is again split into k clusters and a further attempt is made to label the cluster centers of 2nd level cluster. Each sample in the cluster is assigned with the same label as the cluster center. 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

$$w_n = p_c f(d_n) \quad (2)$$

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].

Experiments were performed with different choices for the distance metric(Euclidean and Mahalanobis) and confidence decay function f. For the confidence decay function, we tried both the exponential decay function and gaussian function with different decay rates as given in equation 3 and 4

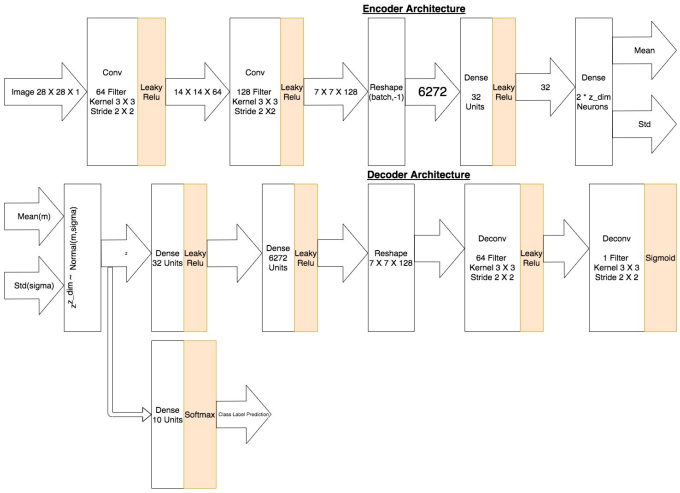

Figure 1: Proposed model architecture

respectively. We tried two different choices for distance 1) Euclidean distance and 2) Mahalanobis distance and two different choices for function f 1) an exponentially decaying function and 2) a function defined by normal curve

$$w_n = p_c e^{-ad_n} \quad (3)$$

$$w_n = p_c e^{-ad_n^2} \quad (4)$$

where a is a hyper parameter determining how fast the confidence decreases

Training is continued for few more epochs using a modified loss function that incorporates the manual input. The modified loss function is

$$L = L_{VAE} - \gamma \sum_{k=0}^{K} w_n y_n \ln(\hat{y}_n) \quad (5)$$

where y_n is the label given to the training images and $\hat{y}$ is the predicted label of the image. The new term added to the loss is the weighted multi-class cross entropy loss for classification task.

We also performed experiments by using gmm instead of k-means for clustering. When using gmm for clustering we directly used the product of posterior probability of the sample and cluster center confidence p_c as the overall sample confidence.

As expected we obtained the best performance when using gmm. It is also observed that when k-means is used the Mahalanobis distance along with a gaussian confidence decay function gave the best result as opposed to Euclidean distance with exponential decay of confidence

4 RESULTS AND DISCUSSIONS

We trained the network for a total of 5 epochs in two different ways.

1. Completely unsupervised using only the variational autoencoder loss function mentioned in equation 1 for all 5 epochs.
2. Using semi-supervised loss function mentioned in equation 5 after first two epochs

Figure7 shows the reconstruction accuracy of the variational autoencoder on the validation images after 5 epochs of training with $\gamma = 0$ plotted as a function of latent vector dimension z_{dim}. It is observed that increasing z_{dim} beyond 10 does not result in an increase in accuracy in the same proportion. This is because the number of nodes in the 3rd layer were fixed at 32 which limits the representational capacity of that and all the subsequent layers.

We choose $z_{dim} = 10$ to run the semi-supervised model, experiment 2 mentioned above, with both reconstruction and classification loss. We ran semi-supervised experiment with different choices of clustering algorithms (k-means and gmm) and using different confidence decay function. Figure 6 shows the classification accuracy on MNIST test images after 5 epochs with different choices of clustering and confidence function. Figure4 shows the decoded cluster center images after the first epoch and Figure5 shows the cluster centers after 5 epochs of training using gmm clustering. It is evident from this figure that as the training progresses the cluster centers get closer to the real semantic classes.

The model gave a classification accuracy of 93.47% on test images within 5 epochs with only 10-16 images (corresponding to each cluster center) annotated after each epoch. The labels given to the cluster center were propagated to all other samples in the cluster and this increases the effective number of labelled samples by a huge factor.

A comparison of reconstruction accuracy with and without classification loss added is shown in Figure 8. The blue curve in figure shows the reconstruction accuracy when the latent vectors were clustered and a label were assigned to the reconstructed images corresponding to cluster centers at the end of every epoch. The figure shows that the reconstruction accuracy of VAE is improved significantly (by 6 to 10 %) when classification loss is added, which indicates that the learned representation generalizes better as in the case of many multi-tasking models.

The distribution of data points were visualized by reducing the latent vector dimension from 10 to 2. Fig 2 shows the distribution with 5 epochs of unsupervised training i.e using only autoencoder loss. Fig 3 shows the same for semi-supervised classification with gmm clustering. From the distribution it is evident that

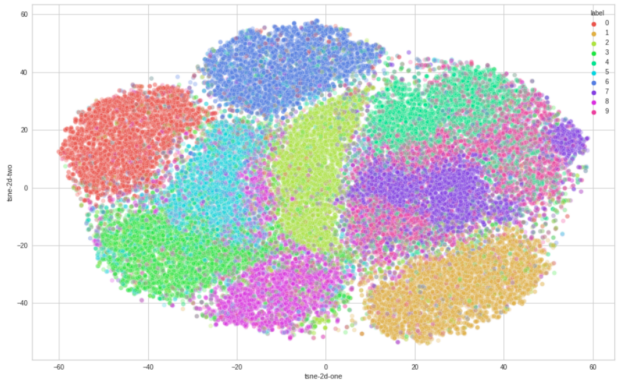

Figure 2: 2 dimensional tsne Visualization of latent space for MNIST imgaes obtained using unsupervised autoencoder

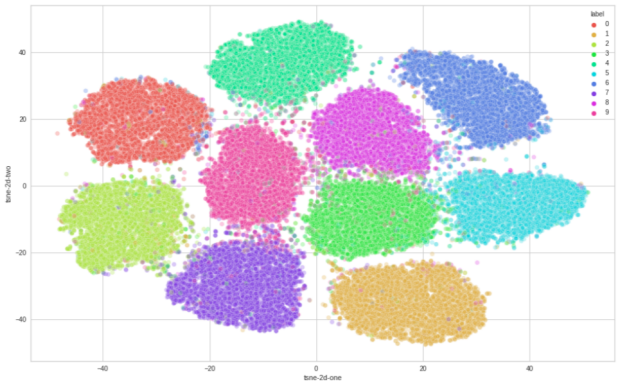

Figure 3: 2 dimensional tsne Visualization of latent space for MNIST imgaes obtained using semisupervised training

1. Semi-supervised learning generates semantically valid clusters
2. Clusters with semisupervised learning are more separated as compared to the latent space learned by unsupervised auto-encoder

5 CONCLUSION

We propose a novel deep learning based active learning framework for solving complex tasks like image classification and segmentation. The proposed approach is demonstrated by classifying MNIST images by annotating only 10-16 images, corresponding to each cluster center, after each epoch. We were able to obtain a classification accuracy of 93.47% after 5 epochs of training. It is possible to increase this accuracy further by increasing the number of clustering levels. It is also observed that the reconstruction accuracy of the generative model improves as a result of performing an auxiliary task of classification Another key observation of our study is that the formation of semantically valid clusters as a result of adding supervised loss and it is also observed that these clusters become more separable

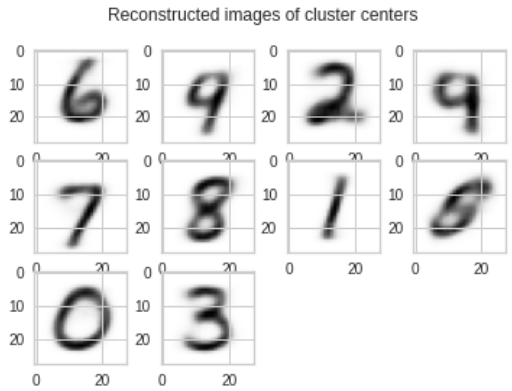

Figure 4: Images of decoded cluster center after first epoch

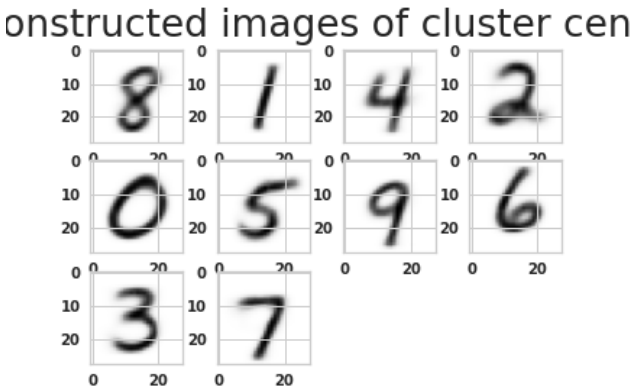

Figure 5: Images of decoded cluster center after 5th epoch

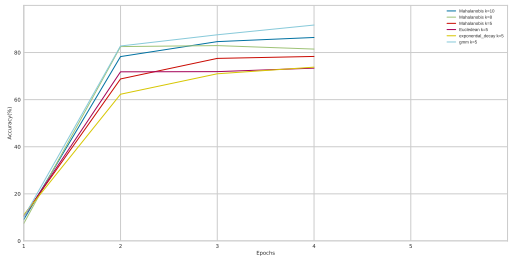

Figure 6: Classification accuracy on test images using semi-supervised trained with $z_{dim} = 10$ and $\gamma = 150$

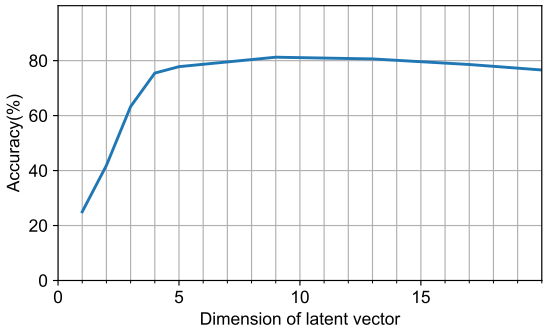

Figure 7: Reconstruction accuracy of autoencoder on validation images as a function the latent vector dimension z_{dim} and $\gamma = 0$

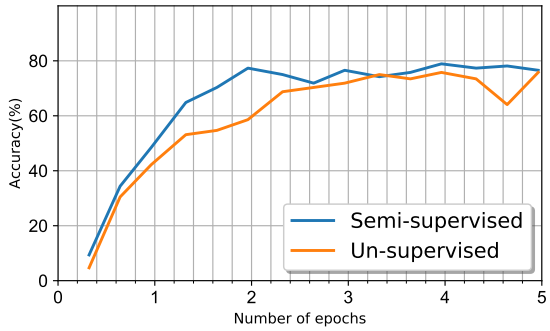

Figure 8: Comparison of reconstruction accuracy with and without classification loss

compared to the purely unsupervised counterpart

References

Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3150–3158, 2016.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European conference on computer vision*, pages 94–108. Springer, 2014.