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 where as we propose a 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 solve tasks which are solved using supervised learning methods with minimal annotation. In this respect, our goal is similar to few shot learning but our approach differs from existing few shot learning techniques.

1 INTRODUCTION

Classifying images is one of the first use cases proven to give good result 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 (10-20 samples in case of MNIST dataset) and less computing resources. We demonstrate our claim by building a classification model for MNIST dataset using only the training images not the label. The proposed approach augments a variational autoencoder with a classification layer the loss component of which is tuned by manually annotating a small number of samples at regular training intervals.

A generative model learns the distribution of data $p(x_{ij})$ where y is the class label. 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 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 much of 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 much less compared to the original image dimension.

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

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 show a method to incorporate human feed backs at regular intervals during training so that the model learns much faster and also the learned latent representations are much more explainable. Such a representation should directly translate to a human explanation for the data. For example, the digit 1 in handwritten dataset can be mentioned as 'a vertical line stroke' and digit 7 can be mentioned as 'a horizontal line stroke towards left placed above a vertical line stroke'. We demonstrate how such a description, along with meaningful properties like line thickness, can be obtained from the latent representation after the model is trained. [TODO add some results for this]

Our model is similar to multi task learning since the loss term have 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 type 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. We also show that such a model can significantly reduce the manual annotation task and training time.

The major contributions of this paper are

1. We propose an active learning approach where the model incrementally learns to perform a task like image classification. Compared to existing deep active learning frameworks our approach requires very less number of training samples and also learns a latent representation and probability distributions $p(z)$ and $p(x/z)$ from which new data samples can be drawn easily
2. The proposed approach reduces the manual annotation task and can be trained faster on CPU

The rest of the paper is organized as follows. Section 2 provides an overview of existing techniques of multi-task learning and few shot learning. Description of dataset used and variables and notations are provided in section 3. Section 4 contains details of network architecture and loss function and training process. A detailed analysis of results of experiments are provided in Section 5. Finally, we conclude our finding in Section **??**

2 RELATED WORK

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 tasks and usually this results in solutions that generalize better Ruder [2017]. While training a network for more than one tasks, other tasks can provide additional evidence for relevance or irrelevance of feature. 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 help in multiple tasks.

TODO add literature survey on few shot learning, concept learning, continual learning

3 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 validation set were used to compute the classification and reconstruction accuracy.

4 PROPOSED METHOD

4.1 DATASET

We used MNIST dataset**?** to demonstrate the proposed approach. The primary reason for selecting MNIST image is to reduce the manual annotation cost required for identifying the reconstructed images. Images in MNIST training set were split into training and validation set with stratified sampling on label column. The validation set, which consist of 128 images, were used to compute the reconstruction accuracy of 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].

4.2 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 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 (TODO format the KLD part and add it in notation in the next para)

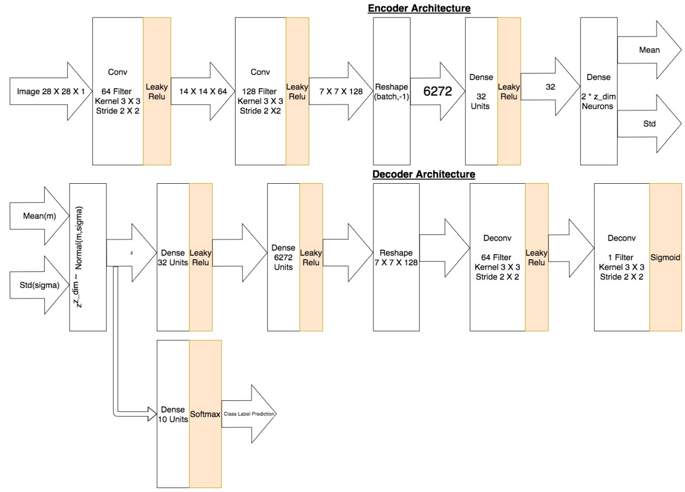

Figure 1: Proposed model architecture

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

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 and $N(0,I)$ is the standard multivariate normal distribution of dimension z_{dim}. We used $\beta = 5$ as it gave a best compromise between reconstruction quality and KL divergence. After few epochs of unsupervised training, the latent vectors corresponding to the training images are clustered using k-means algorithm. The optimum value of k were determined using elbow curve. 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 does not correspond to any valid digit image, or if it is similar to more than one digit image, the cluster is again split into two 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 confidence assigned to the cluster center by human. The confidence of training sample x^n is computed as

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

where d_n is the euclidean distance of the sample from its cluster center, p_c is the confidence assigned to the cluster center and a is a hyper parameter. Training is continued for few more epochs using a modified loss function that incorporates the manual input. The modified loss function is

$$L = L_{VAE} - \gamma \sum_{k=0}^{K} w_n y_n \ln(\hat{y}_n) \tag{3}$$

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 trained the network for 5 epochs. After every 300 steps (with a batch size of 64, this corresponds to 19200 images) of training the reconstructed images were annotated by a manual user. The annotation was done by looking at each of the 128 reconstructed validation images and trying to identify the digit manually. The reconstruction accuracy were then computed by comparing the human identified class label with the actual class label for the image. We ran the experiment 5 times and took the average accuracy.

5 RESULTS AND DISCUSSIONS

Figure 2 shows the reconstruction accuracy of the variational autoencoder on the validation images after 5 epochs of training with $\gamma = 0$ and different values 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.

Figure 3 shows that the reconstruction accuracy of VAE is improved significantly (by 6 to 10 %) when classification loss is added. 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. Figure TODO add fig shows comparison of sample reconstructed images from validation set for normal autoencoder (unsupervised) versus the autoencoder with classification loss added.

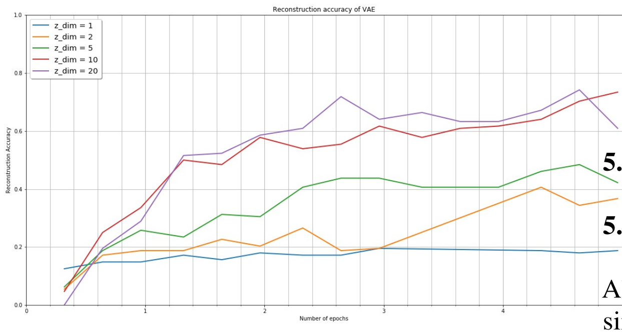

Figure 2: Reconstruction accuracy of autoencoder on validation images with different values for the latent vector dimension z_{dim} and $\gamma = 0$

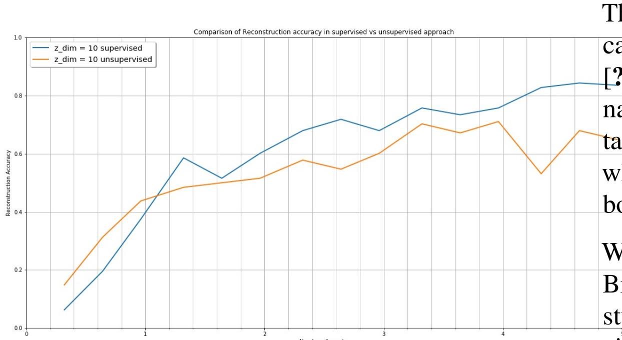

Figure 3: Comparison of reconstruction accuracy with and without classification loss

5.1 SECTIONING

Three numbered sectioning commands are provided: `\section`, `\subsection`, and `\subsubsection`. Please respect their order, so do not put a `\subsubsection` directly beneath a `\section`. One unnumbered sectioning command is provided, `\paragraph`. It can be used directly below any numbered section level. Do not use any other sectioning commands.

5.1.1 Typing the Section Titles

The `\section` and `\subsection` titles are uppercased by the class. Please type them in title case. (This is used in the PDF bookmarks.) Please also write the `\subsubsection` titles in title case.

What is title case? Wikipedia explains:

> Title case or headline case is a style of capitalization used for rendering the titles of published works or works of art in English. When using title case, all words are capitalized except for 'minor' words (typically articles, short prepositions, and some conjunctions) unless they are the first or last word of the title.

5.2 REFERENCES, CITATIONS, FOOTNOTES

5.2.1 Cross-Referencing

Always use `\label` and `\ref`—or a command with a similar effect—when cross-referencing. For example, this subsection is Section 5.2.

5.2.2 Citations

Citations should include the author's last name and year. They should be part of the sentence. An example parenthetical citation: "Good introductions to the topic are available [**?**]." An example textual citation: "**?** discusses electrodynamics of moving bodies." Do not use a parenthetical citation where a textual one is appropriate. An example of what *not* to do: "[**?**] discusses electrodynamics of moving bodies."

We strongly advise to use reference list software such as BibTEX and a citation package such as natbib. The reference style you use should be compatible with the author-year citations. Both the citation style and reference style used should be consistent.

For the original submission, take care not to reveal the authors' identity through the manner in which one's own previous work is cited. For example, writing "I discussed electrodynamics of moving bodies before [**?**]." would be inappropriate, as it reveals the author's identity. Instead, write "**?** discussed electrodynamics of moving bodies."

5.2.3 Footnotes

You can include footnotes in your text.[1] The footnote mark should follow the fragment to which it refers, so a footnote[2] for a word has a footnote mark attached to that word and a footnote for a phrase or sentence has a footnote mark attached to the closing punctuation.

[1] Use footnotes sparingly, as they can be distracting, having readers skip back and forth between the main text and the foot of the page.

[2] A footnote is material put at the foot of a page.

6 MATH

The class file does not load any math support package like amsmath[3]. We advise using the mathtools[4] package, which extends amsmath with fixes and even more useful commands. Feel free to load other support packages for symbols, theorems, etc.

Use the amsmath environments for displayed equations. So, specifically, use the `equation` environment instead of `$$...$$` and the `align` environment instead of `eqnarray`.[5] An `equation`:

$$0 = 1 - 1. \tag{4}$$

Two `align`'ed equations:

$$\begin{aligned} 1+2 &= 3, \\ 1-2 &= -1. \end{aligned}$$

Equations can also be put inline, of course. For example, Equation (4): $0 = 1 + 1$. (Notice that both inline and displayed math are part of the sentence, so punctuation should be added to displayed math.)

The amsmath and mathtools packages provide a lot of nice functionality, such as many common math operators, e.g., sin and max, and also commands for defining new ones.

7 FLOATS

Floats, such as figures, tables and algorithms, are moving objects and are supposed to float to the nearest convenient location. Please do not force them to go in the middle of a paragraph. They must respect the column width.

Two-column floats are possible. They appear at the top of the next page, so strategic placement may be necessary. For an example, see Figure 4. They may not enter the margins.

All material in floats should be legible and of good quality. So avoid very small or large text and pixelated or fuzzy lines.

7.1 FIGURES

Figures should go in the `figure` environment and be centered therein. The caption should go below the figure. Use `\includegraphics` for external graphics files but omit the file extension. Supported formats are pdf (preferred for vector drawings and diagrams), png (preferred for screenshots), and jpeg (preferred for photographs). Do not use `\epsfig` or `\psfig`. If you want to scale the image, it is better to use a fraction of the line width rather than an explicit length. For example, see Figure 5.

Do not use `\graphicspath`. If the images are contained in a subdirectory, specify this when you include the image, for example `\includegraphics{figures/mypic}`.

Table 1: An Interesting Table.

Dataset	Result
Data1	0.12345
Data2	0.67890
Data3	0.54321
Data4	0.09876

7.2 TABLES

Tables should go in the `table` environment and be centered therein. The caption should go above the table and be in title caps. For an example, see Table 1.

7.3 ALGORITHMS

You can load your favorite algorithm package, such as algorithm2e[6]. Use the environment defined in the package to create a centered float with an algorithm inside.

8 BACK MATTER

There are a some final, special sections that come at the back of the paper, in the following order:

- Author Contributions
- Acknowledgements
- References

They all use an unnumbered `\subsubsection`.

For the first two special environments are provided. (These sections are automatically removed for the anonymous submission version of your paper.) The third is the 'References' section. (See below.)

(This 'Back Matter' section itself should not be included in your paper.)

[3] See the amsmath documentation at https://ctan.org/pkg/amsmath for further details.

[4] See the mathtools documentation at https://ctan.org/pkg/mathtools for further details.

[5] For reasons why you should not use the obsolete `eqnarray` environment, see Lars Madsen, *Avoid eqnarray!* TUGboat 33(1):21–25, 2012.

[6] See the algorithm2e documentation at https://ctan.org/pkg/algorithm2e.

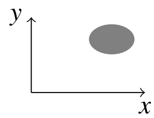

Figure 4: A Nice Filled Ellipse with a Pair of Coordinate Axes.

Figure 5: A View of a Nice City.

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.

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.

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

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.

A MATH FONT EXPOSITION

How math looks in equations is important:

$$F_{\alpha,\beta}^{\eta}(z) = \Gamma(\tfrac{3}{2}) \prod_{\ell=1}^{\infty} \eta \frac{z^{\ell}}{\ell} + \frac{1}{2\pi} \int_{-\infty}^{z} \alpha \sum_{k=1}^{\infty} x^{\beta k} \mathrm{d}x.$$

However, one should not ignore how well math mixes with text: The frobble function f transforms zabbies z into yannies y. It is a polynomial $f(z) = \alpha z + \beta z^2$, where $-n < \alpha < \beta/n \leq \gamma$, with γ a positive real number.