Query Selection based on Latent Space Sampling

Benjamin Killeen

University of Chicago

killeen@uchicago.edu

Abstract

The advent of deep learning has facilitated remarkable success on increasingly complex tasks. Large datasets are integral to this success, providing example labels which guide training, but many tasks are unrelated to existing datasets. In these cases, the choice of an *initial query set* for annotation is crucial. A well-sampled query set can facilitate downstream approaches like semi-supervised or active learning, both of which require a small, labeled dataset. In this paper, we focus on sampling strategies using latent-space representations learned by auto-encoders, a self-supervised variant of deep neural networks. Furthermore, we constrain our latent-space to just two dimensions. Although potentially limiting, this focus allows for an easy visualization of latent spaces well-suited for initial exploration of the topic. We evaluate each sampling strategy using a simple classifier trained on just the initial query set.[1]

Index Terms: semi-supervised learning, active learning, computer vision

1. Introduction

The collection and annotation of large datasets has been critical to the success of deep learning. Wherever such data is available, it seems, the focus of the community eventually results in an effective design for a supervised model. Although that model's performance may extend to similar tasks, many application domains exist outside the scope of established datasets. This is the case especially for scientific image analysis, where the unique nature of each experiment sets it apart from conventional tasks and obtaining labels can require significant investment, monetary- or labor-wise, due to task complexity. Object detection, for instance, requires more in-depth annotation than simple classification. In general, these scenarios call for an approach that minimizes the labeling required to learn a specific task, starting from scratch.

Several sub-fields of machine learning confront this challenge. An active learning approach, firstly, iteratively selects new examples for labeling in the hopes of improving model performance at every step [1]. A popular sampling strategy incorporates the uncertainty of the model for each unlabeled example [1, 2]. Here, the *initial query set* is used to train the first iteration of the model, before any active learning can take place. Typically, the initial query set is selected randomly. This works well in many cases, but we hypothesize that a well-sampled initial query set would result in a better initial model, accelerating the active learning process. Although such tests are beyond the scope of this paper, they remain an active area of inquiry. Secondly, a semi-supervised approach, uses both labeled and unlabeled data during training [3]. Since the initial query set is actually the *only* query set, a well-sampled training set is of the utmost importance for semi-supervised learning.

At this point, the meaning of "well-sampled" is, admittedly, not readily apparent. One would hope, for instance, that a well-sampled query set contains instances from every class, in the case of classification, or from a wide range of values, in the case of regression. Moreover, the representation of these classes or values seemingly ought to be well-balanced, as if sampled *i.i.d.* not from the larger data, which may contain biases, but from the natural distribution itself. However, these properties, however, are distinctly qualitative. One way to quantitatively evaluate the quality of a sampling would be through the performance of a downstream approach that uses an initial query set, such as semi-supervised or active learning. In our experiments, we choose a much simpler downstream approach, namely using the initial query set as the sole training set for a simple classifier. The accuracy of this classifier, measured on a withheld test set, serves as our measure for the "well-sampled-ness" of a query set.

In this paper, we focus on sampling strategies that rely on a latent space representation of the unlabeled set. In particular, we explore latent spaces learned by an auto-encoder (AE) because of the opportunity for similar network design [4]. Our sampling strategies, meanwhile, attempt to approximate a probability distribution in the latent space. There are many choices of AE and distribution; here, we present several options that illustrate the properties of each representation.

2. Related Work

We draw on numerous works for guidance. We are inspired, firstly, by the potential for well-sampled initial query sets in the areas of active and semi-supervised learning, sub-fields of machine learning laid out by Settles *et al.* and Zhu, respectively [1, 3]. We draw form Krixhevsky *et al.* for the design of our convolutional AE, as well as on the work of Kingman *et al.* and Maaten *et al.* for their work on variational auto-encoders and t-SNE embeddings, respectively.

3. Method

Our method for initial query selection relies first on an AE to learn an encoding in latent space and second on a sampling strategy in that latent space. Although we restrict the latent space to $\mathbb{R}^2$ in our experiments (for the sake of visualization and understanding), we discuss here the method for a general latent space $\mathbb{R}^L$. To that end, we consider a set of images $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, where N is the number of examples, and their encoding matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times L}$. We refer to each AE's encoder function $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^L$ and decoder $g_{\boldsymbol{\phi}} : \mathbb{R}^L \to \mathbb{R}^D$ where D is the dimensionality of each input image $\boldsymbol{x}$, and $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are parameter vectors for each model.

[1] Code and additional figures available at github.com/bendkill/latens.

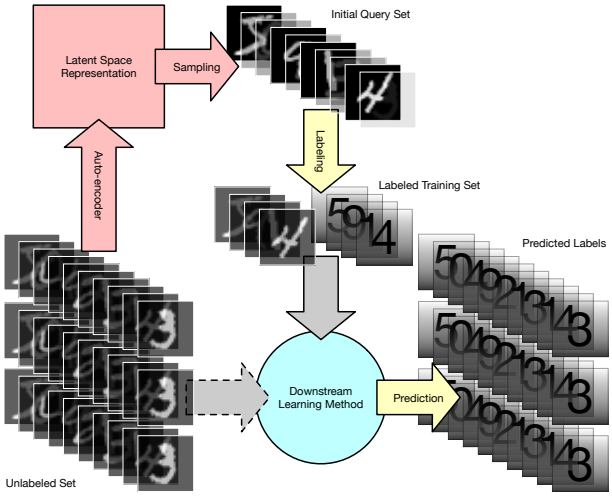


Figure 1: our initial query selection method (red) in relation to a downstream learning method (blue) like active or semi-supervised learning.

3.1. Auto-encoder Architectures

We decided to explore latent space representations learned by an AE in large part because of the opportunity for shared network design. The encoder part of any auto-encoder can be reformatted as a classifier by changing the representation layer to a prediction layer and the loss function to categorical cross-entropy. This also allows for transfer learning knowledge between the AE and the classifier, although that is beyond the scope of this paper.

We explore the representations learned by three AE types: a Convolutional Auto-encoder (CAE), a Variational Auto-encoder (VAE), and a somewhat novel "t-SNE Style" AE. Although the CAE is the only AE with "Convolutional" in its name, each type does in fact use both convolution layers and transpose convolution layers for encoding and decoding, with the main differences between types being the representation layer and loss function.

3.1.1. Convolutional Auto-encoder

Our CAE consists of 28 layers. Inspired by the design in Krizhevsky *et al.*, each layer uses the $ReLU$ activation function and employs zero-padding to maintain image dimensions unless otherwise noted [5, 6]. The first three layers of the encoder $f_{\boldsymbol{\theta}}$ apply convolutions with 64 kernels of shape 3×3. A 2×2 max-pooling follows, reducing image size to 14×14. We apply another three convolution layers, this time with 32 kernels each, along with a second 2×2 max-pool, and three more convolution layers with 32 kernels each. We flatten the resulting $7 \times 7 \times 32$ image and feed it through two 1024-node fully connected layers. Next, the representation layer uses 2 fully-connected nodes with no activation function. The activation at this layer constitutes the encoding $z = g_{\boldsymbol{\phi}}(x)$ of the input image $\boldsymbol{x}$.

The decoder $g_{\boldsymbol{\phi}}$ emulates this design, feeding the representation z through the same layers in reverse order (without tying weights). In place of each max-pool layer, we use a 2×2 transpose convolution with a stride of 2 to increase image dimensions. Finally, a 1×1 convolution layer outputs the reconstructed image $g_{\boldsymbol{\phi}}(z)$. We use a simple MSE loss between the original image $\boldsymbol{x}$ and the reconstruction

$$\mathcal{L}_{\text{CAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = ||g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})) - \boldsymbol{x}||^2 \tag{1}$$

and take the mean over each batch.

3.1.2. Variational Auto-encoder

Our VAE uses the same basic structure while following conventional design for VAEs, a full discussion of which is beyond the scope of this paper. [7]. In place of the CAE's representational layer, we use 2 fully-connected nodes to represent $\boldsymbol{\mu}_z$, taken as z for the encoding, and a further 2 nodes to represent $\log(\boldsymbol{\sigma}_z)$. We draw the input to the decoder from a desired normal distribution so that

$$f_{\boldsymbol{\theta}} = \begin{cases} \boldsymbol{\mu}_z + \epsilon \boldsymbol{\sigma}_z & \text{training} \\ \boldsymbol{\mu}_z & \text{else} \end{cases} \tag{2}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ for every batch.

The loss used by VAEs merits its own discussion, but we present a brief summary here. Variational auto-encoding presumes that some "true" latent-space encoding has a prior distribution $p(\boldsymbol{z})$, which we take as a standard normal distribution. The VAE approximates this posterior $p(\boldsymbol{z}|\boldsymbol{x})$ of p with $q(\boldsymbol{z}|\boldsymbol{x})$ by minimizing the Kullback-Leibler divergence between the two. Minimizing this quantity directly turns out to be computationally intractible, so VAEs maximize the Evidence Lower BOund:

$$\begin{aligned} \text{ELBO}(\boldsymbol{x}) =& \mathbb{E}\left[q(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}|\boldsymbol{z})\right] \\ & - D_{\text{KL}}\left(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right) \end{aligned} \tag{3}$$

We can convert Equation 3, which concerns the probability model of VAEs, into a loss function for encoders and decoders, by noting the resemblance of $q(\boldsymbol{z}|\boldsymbol{x})$ to our encoder $f_{\boldsymbol{\theta}}(x)$ and $q(\boldsymbol{x}|\boldsymbol{z})$ to our decoder $g_{\boldsymbol{\phi}}(z)$. We negate and expand Equation 3 to obtain the loss

$$\begin{aligned} \mathcal{L}_{\text{VAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) =& \text{BCE}(\boldsymbol{x}, g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(x))) \\ & - \frac{1}{2}\mathbb{E}\left[1 + \log(\boldsymbol{\sigma}_z) - \boldsymbol{\mu}_z^2 - \boldsymbol{\sigma}_z\right] \end{aligned} \tag{4}$$

where $\boldsymbol{\mu}_z$ and $\log(\boldsymbol{\sigma}_z)$ are taken directly from activations in the representation layer and BCE and $\mathbb{E}$ denote the binary cross entropy and mean, respectively, over each batch.

3.1.3. "t-SNE Style" Auto-encoder

As implied by the name, our "t-SNE Style" AE emulates the behavior of a t-SNE embedding in its representational layer [8]. It uses the same architecture as the CAE in Section 3.1.1 with a different loss function that incorporates the matrix of representations $\boldsymbol{Z}_B$ over each batch B. Recall the t-SNE relies on the conditional probabilities

$$p_{i|j} = \begin{cases} \frac{\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_k||^2/2\sigma_i^2)} & i \neq j \\ 0 & i = j \end{cases} \tag{5}$$

which we take over each batch, tuning each σ_i to match the desired perplexity. The joint probabilities are then $p_{ij} = (p_{i|j} + p_{j|i})/2|B|$. We also define a joint distribution over the representations:

$$q_{ij} = \frac{\left(1 + ||f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_j)||^2\right)^{-1}}{\sum_{k \neq l}\left(1 + ||f_{\boldsymbol{\theta}}(\boldsymbol{x}_k) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_l)||^2\right)^{-1}} \tag{6}$$

and incorporate the KL divergence between these two distributions in our loss. We have

$$\mathcal{L}_{\text{t-SNE Style}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \text{BCE}(\boldsymbol{x}, g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(x))) + \sum_i \sum_j p_{ji} \log\left(\frac{p_{ji}}{q_{ji}}\right) \quad (7)$$

where each summation is over the current batch.

3.2. Sampling Strategies

Once we obtain the latent space representation Z of data points X, the question remains of how best to sample Z. In this paper, we explore four options that rely on specific probability distributions in the latent space, approximating samples taken from these distributions. Two of these options rely on a hierarchical clustering to capture the natural clustering of points in the latent space [9].

Algorithm 1 shows how a distribution (or set of distributions on each cluster) can be approximately sampled from while still selecting original data points. Our strategies are as follows:

- "Uniform" uses a uniform distribution over Z

$$p = \text{Unif}(\min(Z), \max(Z)).$$

- "Multi-normal" fits a mutlivariate normal distribution to Z, using

$$p = \mathcal{N}(\mathbb{E}(Z), \text{Cov}(Z)).$$

- "Uniform Cluster" uses the uniform distribution over each cluster

$$p_S = \text{Unif}(\min(Z_S), \max(Z_S)).$$

- "Multi-normal" fits a mutlivariate normal distribution to each cluster Z_S, using

$$p_S = \mathcal{N}(\mathbb{E}(Z_S), \text{Cov}(Z_S)).$$

We compare these strategies with two baseline sampling strategies. The first, "Random," selects n random examples from the dataset. The second "AE-error," selects the n examples with the highest average absolute error $\mathbb{E}|\boldsymbol{x_i} - g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(x_i))|$.

Algorithm 1 Select examples from the clustered encoding Z according to a distribution p_S on each cluster S. Applies to sampling strategies without clustering if $\mathbf{S} = \{[N]\}$.

Require:

- the encoding $Z \in \mathbb{R}^L$,
- clustering $\mathbf{S} = \{S_1, \ldots, S_C\}$ on Z,
- distributions $\{p_S : \mathbb{R}^L \to \mathbb{R} | S \in \mathbf{S}\}$, and
- distance threshold $t \in \mathbb{R}$, $n \in \mathbb{N}$.

$Q \leftarrow \{\}$
for Cluster $S \in \mathbf{S}$ **do**
  $n_S \leftarrow \text{round}\left(n \frac{|S|}{N}\right)$
  $Q_S \leftarrow \{\}$
  while $|Q_S| < n_S$ **do**
    Draw $z \sim p_S$
    for $i \in S$ **do** ▷ the ith row of Z, in cluster S
      if $||z - z_i||_2 < t$ **then**
        $Q_S \leftarrow Q_S \cup \{i\}$
      end if
    end for
  end while
  $Q \leftarrow Q \cup Q_S$
end for
return Q

4. Results

We present our results on two versions of the MNIST dataset, the original and a second, unbalanced version [10]. In all our experiments, we use the first $N = 50000$ examples for AE training and sampling, with the remaining 20000 examples reserved for validation and test sets. We use two versions of MNIST in order to demonstrate the main advantage of latent space sampling when starting with no labels. For a balanced dataset with equal representation from each class, taking a random subset of X results in a sufficiently balanced query set even for relatively small a sample size. We created an unbalanced version of MNIST by increasing the representation of each even digit by 2000 and removing the same number of examples for each odd digit. To create new examples, we apply a small translation and rotation to randomly selected originals of the same class. These changes do not affect the validation and test sets. For each dataset, we experimented with two sample or query set sizes, $n = 1000$ and $n = 100$.

To evaluate our latent space representations and sampling strategies quantitatively, we train a convolutional neural network (CNN) with the same design as the encoders in Section 3.1. We train each CNN for 50 epochs on the 1000-example query sets and 500 epochs on the 100-examples sets. In this way, each classifier has an identical number of update steps for training. Figure 2b shows the validation-set accuracies of a classifier as it trains on the query set of 100 examples from the unbalanced MNIST data. As is clear, the "Multi-normal Cluster" sampling strategy outperforms random sampling, likely because it has a much more balance distribution of classes. Classifiers that used query sets with other sampling strategies struggled to train.

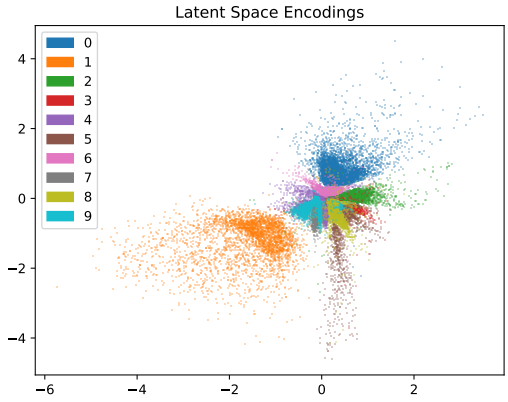
As can be seen in Table 1, this was not the case for our other experiments. Especially when query sets sampled from the original MNIST data, random sampling outperformed or nearly equaled every other strategy. This is to be expected. When sampling from random distributions, as we do, the already class-balanced distribution of the original dataset is optimal. The performance of "Multi-normal Cluster" sampling is encouraging in this light because, although it fails to do better than random sampling, it recovers the same balance in its sampling. Our experiments with the unbalanced dataset confirm this suspicion.

5. Discussion

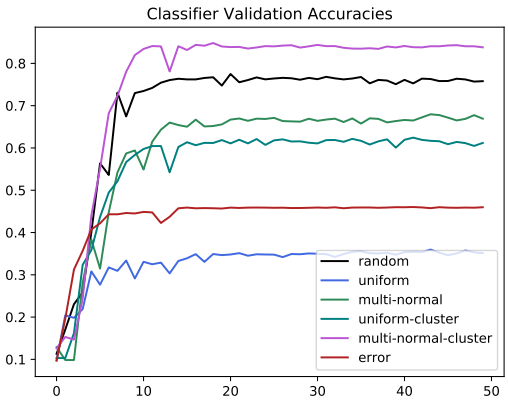
The success of our "Multi-normal Cluster" sampling strategy using the VAE representation is encouraging, but many areas remain open to inquiry. One could explore higher dimensional latent spaces, for instance, and while this could improve performance, we argue that it offers little insight into the underlying problems of initial query selection. Those problems consist of two main questions: "What is the best space to sample from?" and "What is the best sampling strategy for that space?" Unfortunately, one cannot investigate each question separately. Consider the poor performance of uniform-distribution-based samplers in our experiments, which does not necessarily extend to every latent space representation of the data. While the similarity of AEs to the supervised models used in downstream

AE Type	Sampling Strategy	Original MNIST		Unbalanced MNIST	
		1000	100	1000	100
—	Random	$94.0_{\pm.2}$	$80.9_{\pm1.8}$	$92.6_{\pm.4}$	$76.6_{\pm.6}$
—	AE-error	$93.8_{\pm.3}$	$76.6_{\pm.4}$	$57.8_{\pm.2}$	$46.4_{\pm.2}$
CAE	Uniform	72.7	38.1	62.9	44.6
	Multi-normal	93.1	70.3	90.5	66.3
	Uniform Cluster	89.6	71.6	89.2	71.5
	Multi-normal Cluster	94.4	80.9	92.8	79.6
VAE	Uniform	68.2	38.8	62.4	35.3
	Multi-normal	92.1	69.9	91.7	66.5
	Uniform Cluster	86.1	67.3	86.8	62.1
	Multi-Normal Cluster	94.3	**82.9**	92.3	**85.0**
t-SNE Style	Uniform	69.3	56.6	85.7	53.0
	Multi-normal	93.6	82.3	91.7	77.0
	Uniform Cluster	87.3	71.6	91.6	76.6
	Multi-normal Cluster	94.4	78.9	92.4	75.9

Table 1: Test-set accuracies (%) for a classifier trained on query sets from both the balanced and unbalanced MNIST data. As can be seen, the advantage of latent space sampling is most apparent when the original data is unbalanced and the sample size is small enough to reflect it.



(a)



(b)

Figure 2: (2a) shows latent-space encoding for the unbalanced MNIST training set using a VAE. (**??**) shows the validation set loss during training for a classifier on a training set of size 100. (Epoch count in 10s.)

approaches motivated their use in our experiments, other embedding methods may yield more effective performance when combined with our sampling strategies. Additionally, more nuanced sampling strategies could prove effective for a variety of latent space representations.

Although we used distribution-based sampling strategies in our experiments, we believe that deterministic approaches could be more reliable. Future work should identify examples which result in an initial query set balanced not only with respect to classes, like our random sampling baseline on original MNIST, but also with respect to different forms classes. The same digit may be drawn in different styles, for example, each of which should be present in an initial query set.

6. Conclusions

We have shown the potential for latent space sampling to identify a useful initial query set, especially from an unbalanced data. We have shown the effectiveness (or ineffectiveness) of four simple sampling strategies on three types of latent spaces compared to two simple baselines: random selection and error-based selection. In particular, we have compared the test-set accuracies of a classifier trained on each query set. The success of cluster-based methods in this space calls for further exploration of cluster-based sampling strategies.

7. Acknowledgements

Thanks to Gordon Kindlmann for computational resources. Implementation and sampling strategies are original work, as well as the t-SNE Style AE.

8. References

[1] B. Settles, "Active Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, Jun. 2012. [Online]. Available: https://www.morganclaypool.com/doi/abs/10.2200/S00429ED1V01Y201207AIM018

[2] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR94*. Springer, 1994, pp. 3–12.

[3] X. J. Zhu, "Semi-Supervised Learning Literature Survey," University of Wisconsin-Madison Department of Computer Sciences, Technical Report, 2005. [Online]. Available: https://minds.wisconsin.edu/handle/1793/60444

[4] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning." [Online]. Available: http://www.deeplearningbook.org/contents/autoencoders.html

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[6] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," p. 8.

[7] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, Dec. 2013, arXiv: 1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114

[8] L. Maaten, "Learning a parametric embedding by preserving local structure," in *Artificial Intelligence and Statistics*, 2009, pp. 384–391.

[9] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967. [Online]. Available: https://doi.org/10.1007/BF02289588

[10] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/