Progress Update: 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 these labels are not always readily available. The field of active learning addresses this issue by considering "curious" learners which iteratively select query sets that, when annotated, will most inform training. Crucially, this selection depends upon the performance of a supervised learner on already-labeled examples, initially selected randomly. Using techniques from unsupervised learning, our approach leverages the unlabeled data to inform both query selection as well as supervised training. Using an auto-encoder (AE), we consider various sampling strategies in the learned latent space which will produce informative examples. We further investigate the transfer of knowledge from the AE to a supervised learner, in this case a classifier.

Index Terms: active learning, unsupervised learning, computer vision

1. Introduction

The collection and annotation of large datasets has been critical to the success of deep learning. Wherever such a dataset is available, it seems, the focus of the community eventually results in a supervised learner with high performance *on that dataset*. Although that performance may extend to examples from similar tasks, many application domains are outside the scope of established datasets, and relevant annotations may not exist at all. This is the case especially for scientific image analysis, where the unique nature of each experiment sets it apart from conventional tasks and the labeling itself is the ultimate goal. In this context, one aims to train a supervised learner on as few labeled examples as possible, possibly leveraging the unlabeled data to do so.

2. Method

Given the nature of this report, we separate the discussion of our envisioned method, in Section 4, with the description of our current method, given here. This method is subject to change, especially with regard to the auto-encoder architecture and sampling strategy.

Latent space sampling uses a large unlabeled dataset to identify examples which should be labeled, according to one of several strategies (which we expound on in this report). Initially, we consider data matrix $X \in \mathbb{R}^{N \times D}$, where N is the number of examples and D is the dimensionality of each example. In our experiments, we use $28 \times 28 \times 1$ images, so $D = 784$. From this unlabeled set, we use an auto-encoder to learn a low-dimensional or latent space representation $Z \in \mathbb{R}^{N \times L}$ of X, where L is the dimensionality of the latent space.

Currently, we use a convolutional auto-encoder (CAE) to learn a 10-dimensional encoding of the dataset [1, 2]. The CAE consists of an encoder and a decoder. The encoder uses multiple 3×3 convolutional layers broken up with two 2×2 max-pooling to reduce the image dimensions to $7 \times 7 \times 64$ in the last convolutional layer. This is followed by two dense layers with 1024 and L nodes respectively. The decoder reverses this architecture, replacing max-pool layers with 2×2 transpose convolutions. In each layer, we use the ReLU activation function, except for the encoder's final layer, the "representation layer." The choice of activation for the representation layer is driven by a combination of performance considerations—what generates good encodings—as well as sampling considerations. Initially, we envisioned that constraining the latent space to the hypercube $[0, 1]^L$ would result in an easily sampled representation with good spread.[1] Experiments with the sigmoid activation function proved unsatisfying, and so following Nair and Hinton's Rectified Linear Unit [3], we employ a clipped linear unit (CLU) given by $f(x) = \min(1, \max(0, x))$, in our initial experiments. This choice was made in order to guarantee a hypercube constrained representation $Z \in [0, 1]^{N \times L}$ while still providing the performance advantages of the ReLU.

Once we obtain Z, we select a sampling or index set $Q \subseteq [N]$ based on the distribution of Z. One strategy, which we employ in our preliminary experiments, is to sample from this space according to a random uniform distribution. As described in Algorithm 1, we draw a point $z \in [0, 1]^L$ uniformly and, if any data-point representation z_i exists within a given distance $d(z, z_i)$, then we add i to the query set. Figure 1 provides an overview of this representation learning and sampling process.

Algorithm 1 Approximate a uniform sampling of the latent-space hypercube $[0, 1]^L$ from the encoding Z.

Require: encoding $Z \in [0, 1]^{N \times L}$, distance metric $d : [0, 1]^{2 \times L} \to \mathbb{R}$, distance threshold $t \in \mathbb{R}$, $n \in \mathbb{N}$.
1: $Q \leftarrow \{\}$
2: **while** $|Q| < n$ **do**
3: Draw $z \in [0, 1]^L \sim \mathrm{Unif}^L(0, 1)$
4: **for** $z_i \in Z$ **do** ▷ the ith row of Z
5: **if** $d(z, z_i) < t$ **then**
6: $Q \leftarrow Q \cup \{i\}$
7: **end if**
8: **end for**
9: **end while**
10: **return** Q

3. Results

Although our main work until now has focused on implementation, we have some early results of training that are worth

[1] This vision has since changed. See Section 4 for details.

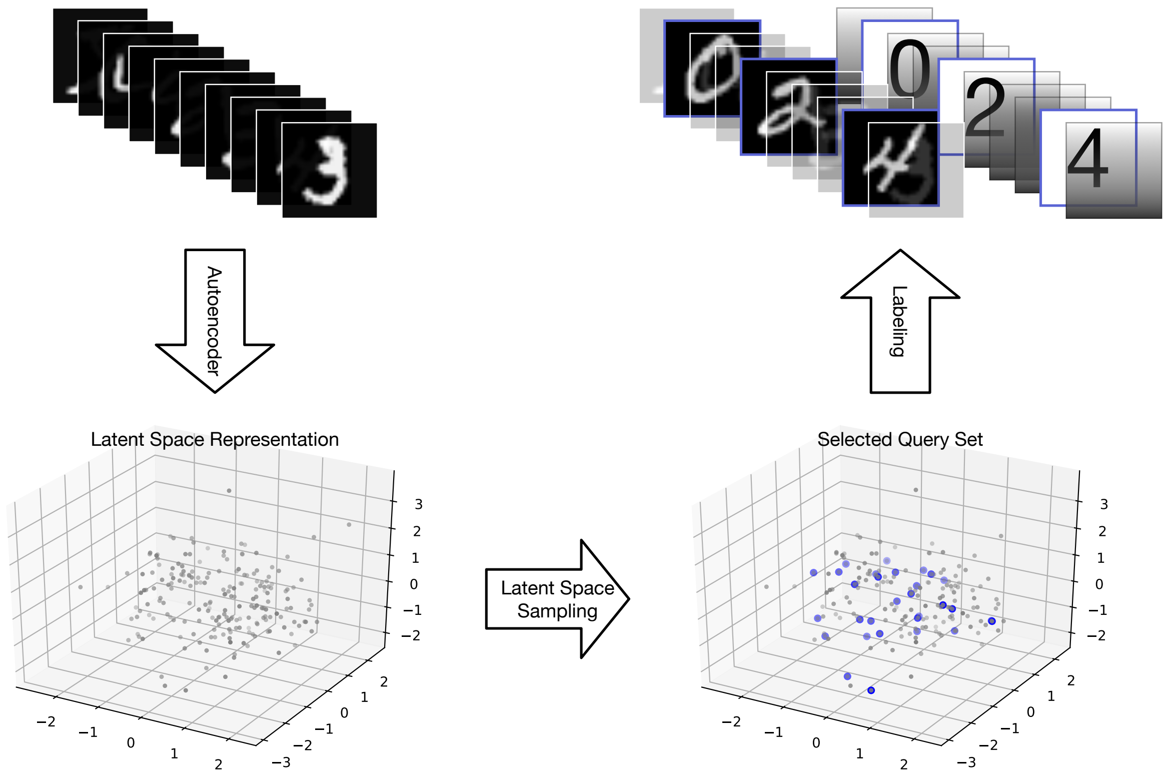

Figure 1: an overview of the query selection process. Given a large, unlabeled dataset, we train an autoencoder to generate latent space representations of each example. Next, the "latent space sampling" identifies examples which are well-spread-out in the low-dimensional encoding. Finally, the selected examples in the original dataset are labeled for training.

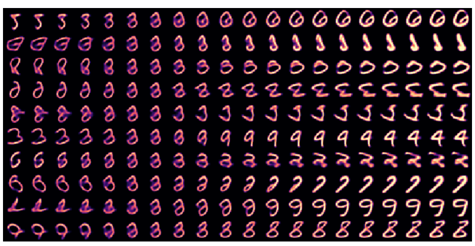

Figure 2: a visualization of the latent space $[0, 1]^{10}$ learned by a simple convolutional autoencoder. The ith row corresponds to variation of the ith latent dimension from 0 to 1, fixing all other coordinates at 0.3.

showing. Firstly, we visualize the behavior of an auto-encoder trained with $L = 10$. This auto-encoder was trained for 20 epochs on the 50,000 image training set. It achieved a mean absolute error on the test set of 0.0604. Figure 2 shows decodings of points in latent-space, varying a different dimension in each row of the image. As can be seen, the representation produces many images that are meaningful, but it does not fully explore the available space as might be desired for our purposes.

Sampling the latent representation Z for 1000 examples according to our method, we trained a simple convolutional classifier from scratch. This achieved 93.1% accuracy on the 10,000 example test set. Although seemingly promising, we note that a random sampling of the training set produced similar results. Additionally, this preliminary finding trained the classifier from scratch, without using any transfer learning.

4. Discussion

Because of the difficulties we face with a simple CAE, we plan to implement a variational auto-encoder (VAE) with convolution layers, restricting the latent-space to an even lower dimension such as $L = 2$ in the hopes of producing a more meaningful and easily-sampled Z, while still using Algorithm 1. After this step, it should no longer be necessary to restrict the representation layer to the hypercube $[0, 1]^L$.

Furthermore, we have some interest in exploring latent-space points which are not in the original training set. If time allows, we plan to use the decodings of regularly sampled points in the latent space as a training set for a classifier, either employing an actual human annotator to provide labels for these novel examples or using a separate classifier trained on the full dataset to simulate a human labeler.

5. References

[1] 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

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

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