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 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 (AEs), 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 a supervised learner with high performance, as evaluated on the associated test set. Although that performance may extend to examples from 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.

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 [2]. A popular sampling strategy incorporates the uncertainty of the model for each unlabeled example [2, 3]. 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 [1]. 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.

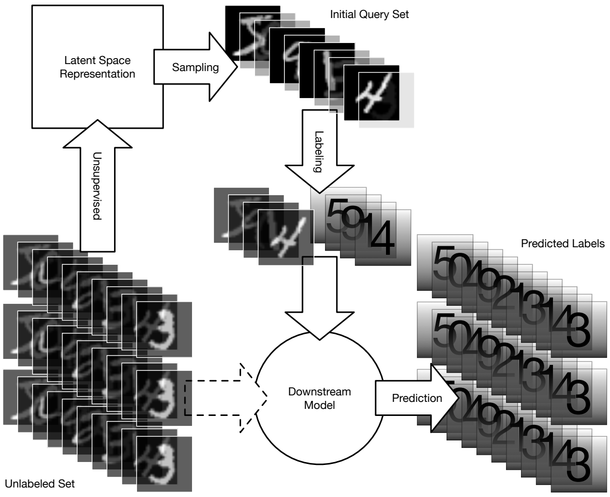

Figure 1: an overview of latent space sampling for query selection.

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.

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 sub.ject 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

[1] Code available here.

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 [4, 5]. 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.[2] Experiments with the sigmoid activation function proved unsatisfying, and so following Nair and Hinton's Rectified Linear Unit [6], 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 $t \in \mathbb{R}$, $n \in \mathbb{N}$.
Q ← {}
while $|Q| < n$ do
    Draw $z \in [0,1]^L \sim \mathrm{Unif}^L(0,1)$
    for $z_i \in Z$ do                ▷ the $i$th row of $Z$
        if $d(z, z_i) < t$ then
            $Q \leftarrow Q \cup \{i\}$
        end if
    end for
end while
return $Q$
```

3. Results

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

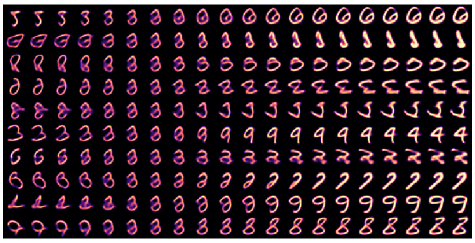

Figure 2: a visualization of the latent space $[0, 1]^{10}$ learned by a simple convolutional autoencoder. The ith row corresponds to variation of the ith latent dimension from 0 to 1, fixing all other coordinates at 0.3.

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.

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

5. References

[1] 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

[2] 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

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

[4] 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

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

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