Exploring Generative Models for Semi-Supervised Learning

Daniel De Freitas Adiwardana, Akihiro Matsukawa, Jay Whang

Stanford Center for Professional Development / Google

Background

Semi-supervised learning tries to take advantage of unlabeled data from the same distribution to improve performance on a supervised task. One common class of techniques is based on label propagation, which attempts to "smear" the ground truth labels from the labeled examples to the unlabeled ones based on some similarity heuristic. Another class of techniques attempts to use the unlabeled data directly in the training objective, for example to first train an autoregressive model to initialize good weights, or to train for some joint objective. It is this second class that we focus on in this work.

Dataset

We measure the performance of semi-supervised techniques on MNIST digit classification [1]. Our baseline is a convolutional neural network (CNN). The purpose of this model is to provide fair supervised classification performance. One key metric we focus on is the number of labeled data points required to reach similar level of performance as the baseline model. If we are to truly benefit from semi-supervised techniques, the use of unlabeled data should allow us to reach similar performance with fewer labeled examples.

Problem Statement

Note that this problem statement is presented fairly generally, so as to be applicable to different generative models. We are given a dataset $D = \{X, Y, X'\}$ where (X, Y) are the labeled points, and X' is the rest of the unlabeled data, which is often orders of magnitude larger than X. Our experiments involve first training a generative model $U(\{X, X'\})$ then transferring its features to a supervised task $S(X, Y; U)$.

We compare the performance of this to the model trained only on the available supervised data, $S'(X, Y)$ in prediction ability and amount of data required to converge to good results. We also jointly train a generative model with a discriminative model (i.e. classifier) using all of $\{X, Y, X'\}$. This approach was inspired by the work of Salimans et al. [3].

Samples from Generative Models

To learn the data distribution of unlabeled example images, we considered the two most powerful generative models: Deep Convolutional GAN [2] and PixelRNN [4]. Here we include samples of digit images generated by the models we trained.

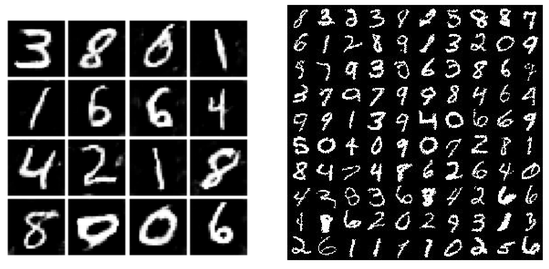

Figure 1: Samples from DCGAN (left) and PixelRNN (right).

PixelRNN Results

- We generated PixelRNN hidden states for every image in the MNIST and used that as input for supervised classification. In other words, for each MNIST image we obtained a 28x28x64 embedding that contains features learned via the unsupervised training of the PixelRNN model.
- The "Golden 10"
  - Each image embedding contains 28x28x64 = 50,176 unsupervised features.
  - Using all features with just a linear classifier yields 99% test accuracy.
  - We then trained a linear model with just the 10 features that maximize the Pearson linear correlation scores (instead of 50,176) and obtained test accuracy of 84.16% indicating the existence of unsupervised features that are strongly correlated with the concept of digit types.

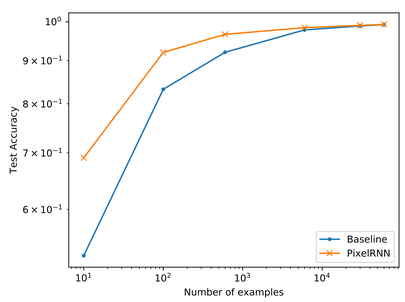

Figure 2: Test set accuracy of using PixelRNN embeddings vs. baseline usage of pixels. The classifier is a CNN for both.

PixelRNN Results (cont'd)

- Which RNN time steps are the most useful?
  - We plotted the weights of a linear classifier trained using embeddings, MNIST labels and $L1$ regularization to find which of the embedding time steps were the most predictive.

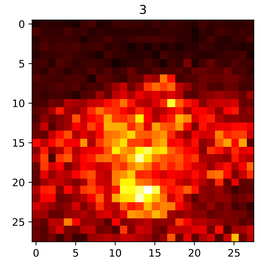

Figure 3: Linear classifier weights for digit 3.

Multi-class GAN Results

By combining the DCGAN architecture [2] and the work of Salimans et al. [3], we created a semi-supervised DCGAN model where the discriminator loss consists of three components:

- (supervised loss) the cross-entropy loss from the predicted distribution over $K(= 10)$ digit classes: $-\mathbb{E}_{\boldsymbol{x},y\sim p_{data}}[\log p_{model}(y|\boldsymbol{x}, y < K + 1)]$
- (unsupervised loss) the loss from classifying unlabeled data points as real, i.e. class $\neq K + 1$: $-\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log(1 - p_{model}(y = K + 1|\boldsymbol{x}))]$
- (GAN sample loss) the loss from classifying generate4d images as fake, i.e. class $= K + 1$: $-\mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{model}(y = K + 1|\boldsymbol{x})]$

Notice that an artificial "fake" class is added, corresponding to the class $K + 1$. This approach allows us to jointly train the discriminator network to serve two functions: as a classifier over K classes, and as a discriminator between real and fake images (K real classes vs. the "fake" class). Our model performs better than what was reported in [3], presumably because of the convolutional generator network based on DCGAN instead of a fully-connected one.

Multi-class GAN Results (cont'd)

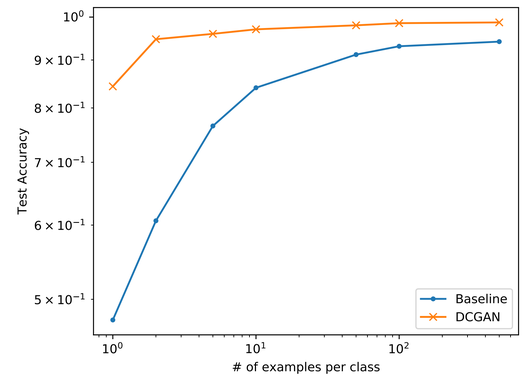

Figure 4: Test set accuracy, trained on subsets of training data: baseline CNN vs. DCGAN-based model. Notice that the DCGAN-based model performs much better than the baseline model, especially when there are very few examples. It achieves 84% accuracy when only single image is provided for each class for a total of 10 labeled training examples.

Analysis and Conclusion

Generative models can learn semantic representations and constraints that used to come from labeled data. PixelRNN embeddings promoted an increase in test accuracy of up to 30% when only training with 1 example of each digit. Augmenting the usual supervised loss with the DCGAN's unsupervised loss allowed us to obtain 96% test accuracy with only 50 examples. Further improvements to generative models and scaling up to larger amounts of unlabeled data could yield even more performance gains on supervised tasks and label reduction.

References

[1] Y. LeCun and C. Cortes.
MNIST handwritten digit database.

[2] A. Radford, L. Metz, and S. Chintala.
Unsupervised representation learning with deep convolutional generative adversarial networks.

[3] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen.
Improved techniques for training gans.

[4] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu.
Pixel recurrent neural networks.