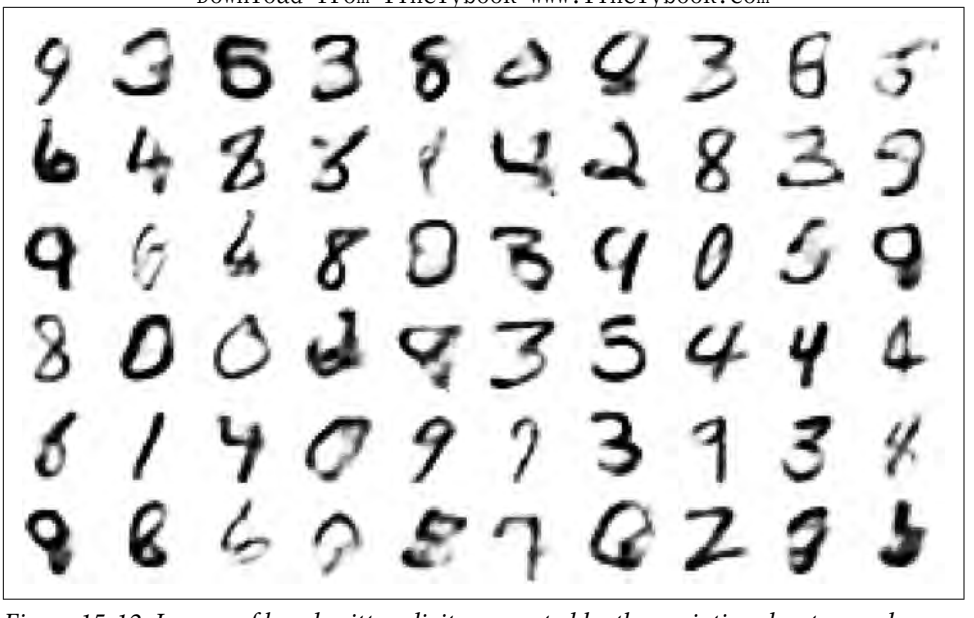

Figure 15-12. Images of handwritten digits generated by the variational autoencoder

A majority of these digits look pretty convincing, while a few are rather "creative." But don't be too harsh on the autoencoder—it only started learning less than an hour ago. Give it a bit more training time, and those digits will look better and better.

Other Autoencoders

The amazing successes of supervised learning in image recognition, speech recognition, text translation, and more have somewhat overshadowed unsupervised learning, but it is actually booming. New architectures for autoencoders and other unsupervised learning algorithms are invented regularly, so much so that we cannot cover them all in this book. Here is a brief (by no means exhaustive) overview of a few more types of autoencoders that you may want to check out:

Contractive autoencoder (CAE)[8]
: The autoencoder is constrained during training so that the derivatives of the codings with regards to the inputs are small. In other words, two similar inputs must have similar codings.

8 "Contractive Auto-Encoders: Explicit Invariance During Feature Extraction," S. Rifai et al. (2011).

Stacked convolutional autoencoders[9]
: Autoencoders that learn to extract visual features by reconstructing images processed through convolutional layers.

Generative stochastic network (GSN)[10]
: A generalization of denoising autoencoders, with the added capability to generate data.

Winner-take-all (WTA) autoencoder[11]
: During training, after computing the activations of all the neurons in the coding layer, only the top *k*% activations for each neuron over the training batch are preserved, and the rest are set to zero. Naturally this leads to sparse codings. Moreover, a similar WTA approach can be used to produce sparse convolutional autoencoders.

Adversarial autoencoders[12]
: One network is trained to reproduce its inputs, and at the same time another is trained to find inputs that the first network is unable to properly reconstruct. This pushes the first autoencoder to learn robust codings.

Exercises

1. What are the main tasks that autoencoders are used for?
2. Suppose you want to train a classifier and you have plenty of unlabeled training data, but only a few thousand labeled instances. How can autoencoders help? How would you proceed?
3. If an autoencoder perfectly reconstructs the inputs, is it necessarily a good autoencoder? How can you evaluate the performance of an autoencoder?
4. What are undercomplete and overcomplete autoencoders? What is the main risk of an excessively undercomplete autoencoder? What about the main risk of an overcomplete autoencoder?
5. How do you tie weights in a stacked autoencoder? What is the point of doing so?
6. What is a common technique to visualize features learned by the lower layer of a stacked autoencoder? What about higher layers?
7. What is a generative model? Can you name a type of generative autoencoder?

9 "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," J. Masci et al. (2011).

10 "GSNs: Generative Stochastic Networks," G. Alain et al. (2015).

11 "Winner-Take-All Autoencoders," A. Makhzani and B. Frey (2015).

12 "Adversarial Autoencoders," A. Makhzani et al. (2016).