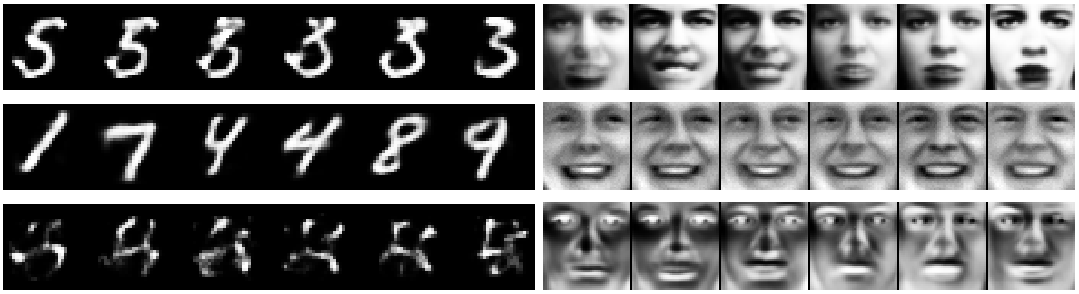

Figure 1. Samples from models trained on MNIST (left) and TFD (right). Top row: 2-layer CAE using proposed sampling procedure (Jacobian-based). Middle row: 2-layer DBN using Gibbs sampling. Bottom row: samples obtained by adding isotropic instead of Jacobian-based Gaussian noise.

images 48x48 pixels which makes it particularly interesting in the context of unsupervised learning algorithms, and 4000 labeled images with 7 facial expressions. We use the same preprocessing pipeline described in Ranzato et al. (2011).

5.1. Evaluating sample generation

We used the sampling procedure proposed in Section 3 to generate samples from two layer stacks of ordinary CAE (denoted CAE-2), that were trained on the MNIST and TFD datasets. To verify the importance of basing the stochastic perturbation of the hidden units on the CAE's Jacobian, we also run an alternative technique where we instead add isotropic noise. For comparison we also generated samples with Gibbs samling from a 2-layer Deep Belief Network denoted DBN-2 (i.e. stacking two RBMs). For the first RBM layer we used binary visible units for MNIST, and Gaussian visible units for TFD. Hidden units were binary in both cases. Figure 1 shows the generated samples for qualitative visual comparison. Figure 3 shows the evolution of the reconstruction error term, as we sample using either Jacobian-based or isotropic hidden unit perturbation. We see that the proposed procedure is able to produce very diverse samples of good quality from a trained CAE-2 and that properly taking into account the Jacobian is critical. Figure 2 shows typical first layer weights (filters) of the CAE-2 used to generate faces in Figure 1.

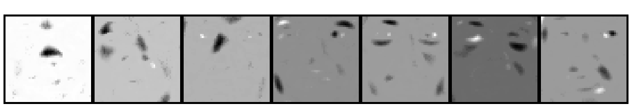

Figure 2. Typical filters (weight vectors) of the first layer from the CAE-2 used to produce face samples.

To get a more objective quantitative measure of the quality of the samples, we resort to a procedure proposed in Breuleux et al. (2011) that can be applied to compare arbitrary sample generators. It consists in measuring the log-likelihood of a test set (not used to train the samplers) under the density obtained from a Parzen-Windows density estimator[3] based on 10000 generated samples. Table 1 shows the thus measured log-likelihoods.

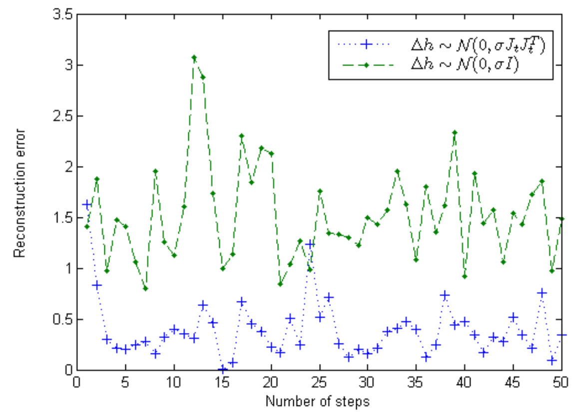

Figure 3. Evolution of the reconstruction error term, as we sample from CAE-2 trained on MNIST, starting from uniform random pixels (point not shown, way above the graph). Sampling chain using either Jacobian-based (blue) or isotropic (green) hidden unit perturbation. Reconstruction error may be interpreted as an indirect measure of likelihood.

Table 1. Log-Likelihoods from Parzen density estimator using 10000 samples from each model

	DBN-2	CAE-2
TFD	**1908.80** $\pm$ 65.94	**2110.09** $\pm$ 49.15
MNIST	**137.89** $\pm$ 2.11	121.17 $\pm$ 1.59

5.2. Evaluating the invariant-feature learning criterion

Our next series of experiments investigates the effect of the training criterion proposed in section 4 to learn

[3]using Gaussian kernels whose width is cross-validated on a validation set