Generative networks (Boltzmann, RBM, GAN)

Boltzmann machine

Move from a deterministic to stochastic regime for asynchronous update:

$$\text{total input:} \quad a_i = \sum_{j=1}^{N} w_{ij} v_j$$

$$\text{update rule:} \quad P(v_i = 1) = f(a_i)$$

$$\text{where} \qquad f(a_i) = \frac{1}{1 + \exp(-a_i)}$$

System converges to an equilibrium state for the states **v** given by:

$$\text{energy function:} \quad E(\mathbf{v}) = -\frac{1}{2}\mathbf{v}^T W \mathbf{v}$$

$$\text{Boltzmann distribution:} \quad P(\mathbf{v}) = \frac{\exp(-E(\mathbf{v}))}{\sum_{\mathbf{v}} \exp(-E(\mathbf{v}))}$$

Can also introduce “hidden units” to detect higher order correlations (not just pairwise).

Restricted Boltzmann Machine (RBM)

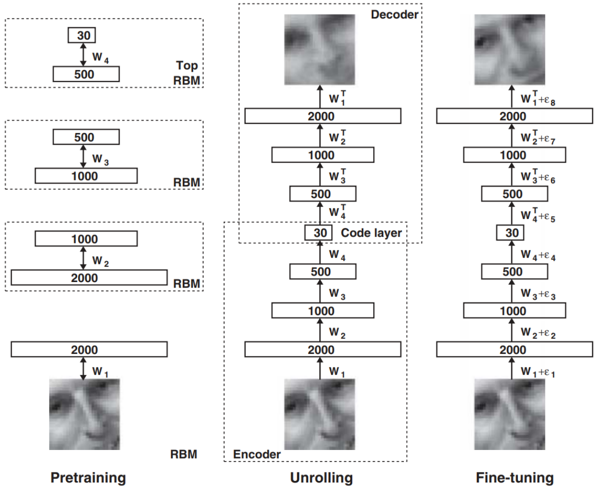
Two layer network, with input layer connected to/from hidden layer; no within-layer connections.
In a **wake** phase, input units are clamped on, and drive hidden layer. In a **sleep** phase, hidden layer units can drive inputs.
Trained using a procedure called Contrastive Divergence (Stone, Chapter 7).
Much more efficient than simulated annealing for Boltzmann machines.

Stacked RBMs

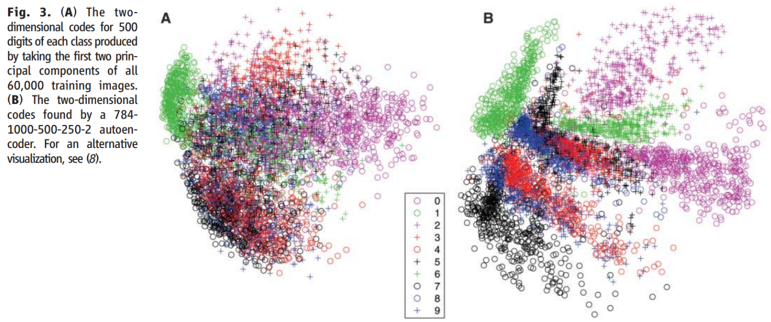
Can train a stack of RBMs one-by-one, such that a hidden layer, once trained is used as input layer to next RBM.

Autoencoders (Hinton and Salakhutdinov 2006)

After training stacked RBM, we have an encoding network, which can be "flipped" to make a decoder with same weights. Can then refine whole net with backprop.

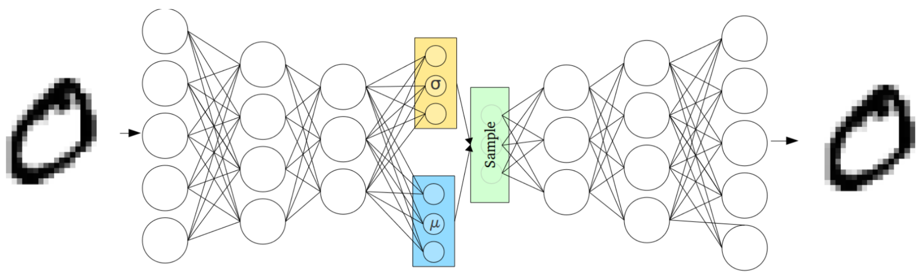


MNIST visualisation (Figure 3 of Hinton and Salakhutdinov 2006)

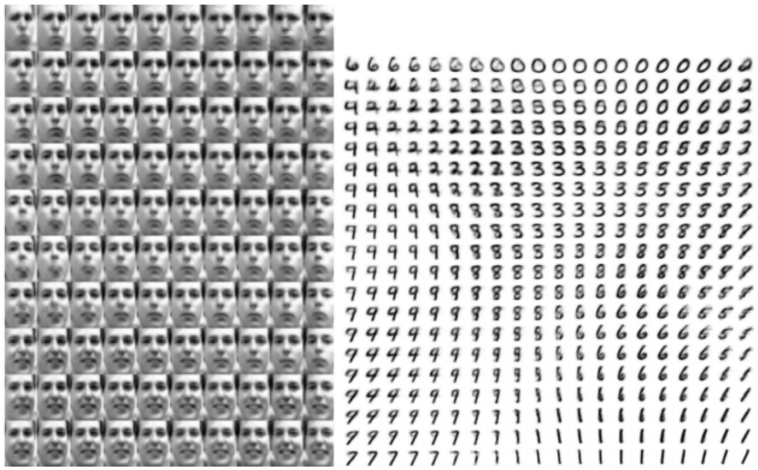


Sup layer above autoencoder classified MNIST with 1.6% error. (Stone, p96).
Netflix 1 million USD prize won by team using SVD + RBMs; not used as films moved to online delivery.
`https://www.techdirt.com/articles/20120409/03412518422/why-netflix-never-implemented-algorithm-that-won-netflix-1-mil`

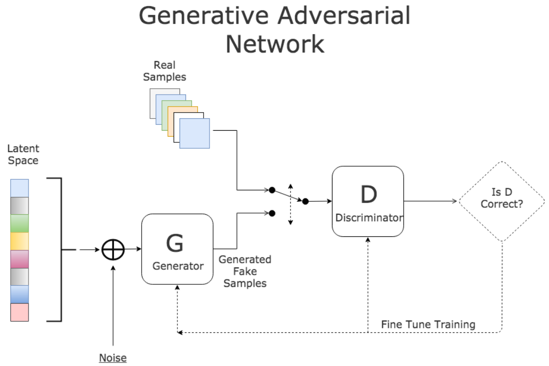
Variational autoencoders



Sampling the latent space (Goodfellow , Figure 20.6)



Generative adversarial neworks (Goodfellow et al 2014)



1. **Discriminator** spots real vs fake training samples. Adjust weights to increase discrimination.
2. **Generator** adjusts weights to generate images that are more likely to be classified as training images.

Source: `https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html`

For further information:
`http://bamos.github.io/2016/08/09/deep-completion/`

Radford et al. (2015), figure 4

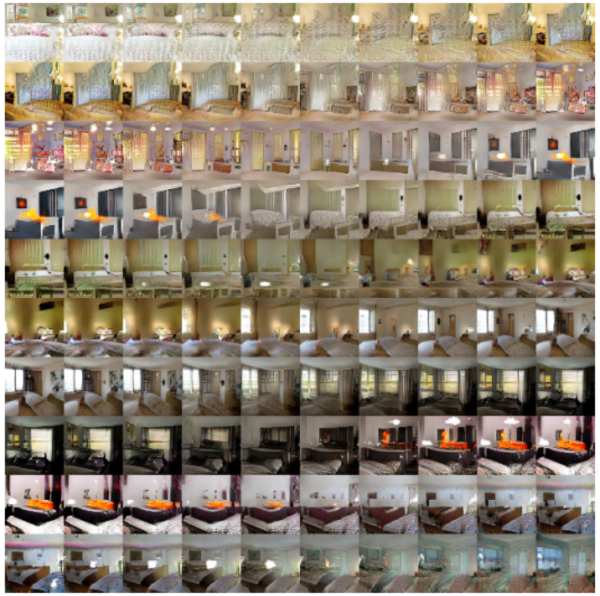
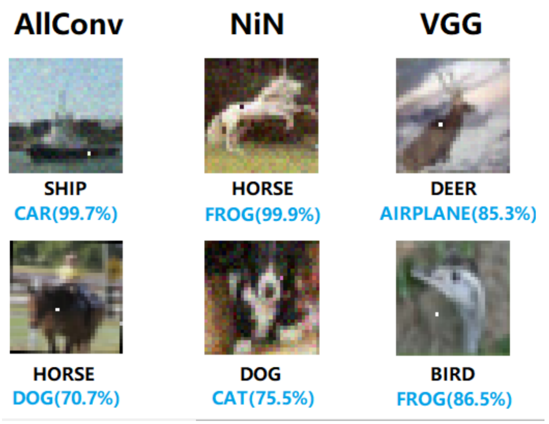


Figure 4: Top rows: Interpolation between a series of 9 random points in Z show that the space learned has smooth transitions, with every image in the space plausibly looking like a bedroom. In the 6th row, you see a room without a window slowly transforming into a room with a giant window. In the 10th row, you see what appears to be a TV slowly being transformed into a window.

Fooling Deep Networks with adversarial samples



Su et al (2017)
See also `https://arxiv.org/pdf/1707.08945.pdf` for robust attacks on stop signs.