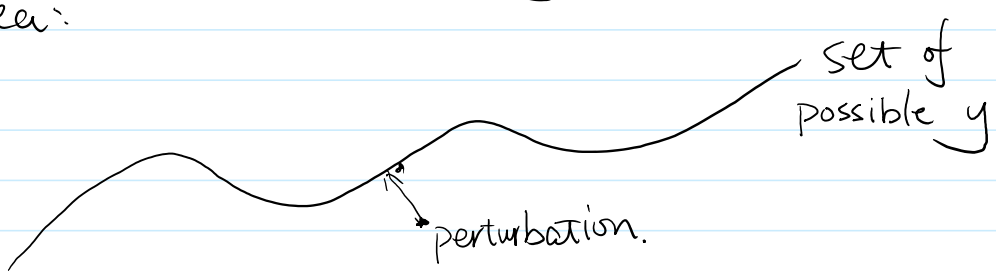
Lecture 22 Reversible/Random-like neural network

Tuesday, November 15, 2016 1:44 PM

- reversible neural network.
  - auto encoders
    - way to do unsupervised learning using NN.
    - encoding function $enc(y) = \sigma(Ay+b)$
    - decoding function $dec(x) = \sigma(A'x+b')$
    - goal: for any possible x

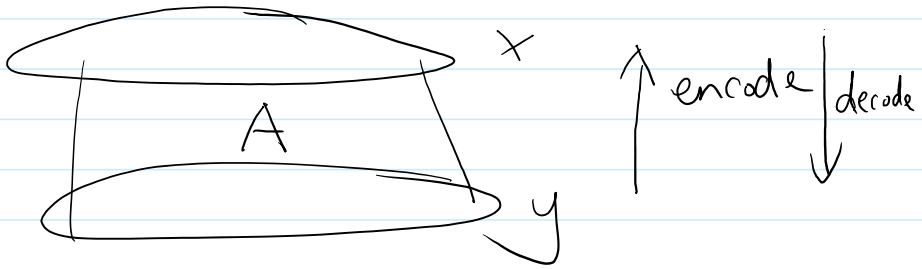
      $$enc(dec(x)+noise) \approx x$$

      (denoising auto encoder)
      also, for any observation y $dec(enc(y+noise)) \approx y$
  - idea:

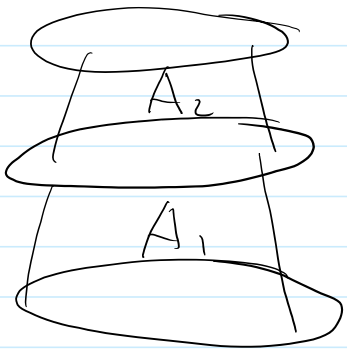


usually require the code to contain less information
(x is low rank / sparse)
recall: compression $\Rightarrow$ good for learning

- weight tying: $A' = A^T$



- stacked autoencoders.



can stack multiple layers of autoencoders.
the denoising property should hold for each layer.

- "random-like" neural network.

Q: What matrices form good denoising autoencoders?

A: The matrix should be "random-like"

Claim: A random matrix is a denoising autoencoder.

Recall: Sparse recovery

x: sparse vector

$y = Ax$

Given y, find x.

We saw: When columns of A are incoherent,

$$x_i \approx \langle a_i, y \rangle$$

$$x \approx \text{truncate}(A^T y)$$

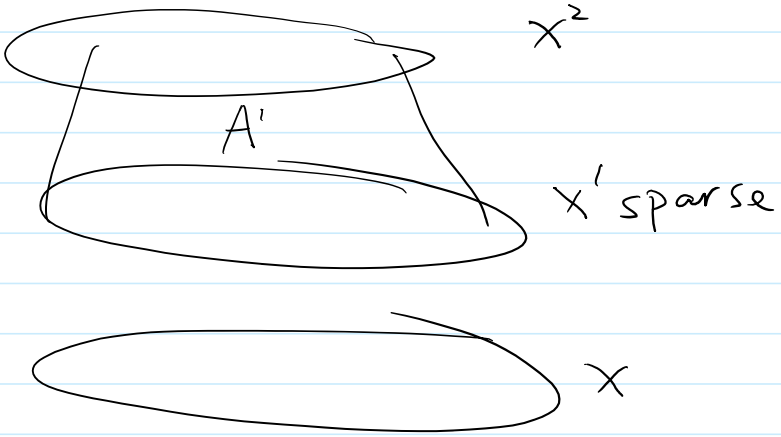
$$\text{truncate}(x) = \begin{cases} 0 & \text{if } |x| \text{ small} \\ x & \text{o.w.} \end{cases}$$

- random vectors are incoherent.
- infact, if A is a random matrix

$x \approx \text{truncate}(A^T A x)$ even when sparsity is higher.

- Stacked autoencoders

problem: the code x^1 is already sparse



but $x^1 = \sigma((A^1)^T x^2 + b)$, how to make sure x^1 is sparse?

- 2 ideas

① make both x^2 and A^1 sparse

[ABGM]: can learn these deep nets.

② play with σ and b

more plausible in practice.