Learning robust representations through input perturbations.
From stochastic perturbations to analytic criteria and back.

Pascal Vincent
We want to learn a continuous deterministic mapping from input $x$ to hidden representation $h$.

- Using auto-encoders

- $h(x)$ should retain enough information to allow reconstructing $x$
Auto-Encoders (AE)

hidden representation \( h = h(x) \)

Encoder: \( h \)

Decoder: \( g \)

input \( x \) \hspace{2cm} \text{reconstruction} \hspace{2cm} \text{reconstruction error} \hspace{2cm} L(x, r)

Output reconstruction: \( r = g(h(x)) \)
Auto-Encoders (AE)

Typical form

hidden representation $h = h(x)$

Encoder: $h$

Decoder: $g$

input $x$

reconstruction $r = g(h(x))$

reconstruction error $L(x, r)$
**Auto-Encoders (AE)**

**Typical form**

hidden representation \( h = h(x) = s(Wx + b) \)

Encoder: \( h \)

Decoder: \( g \)

input \( x \)

reconstruction error \( L(x, r) \)

reconstruction \( r = g(h(x)) \)
Auto-Encoders (AE)

**Typical form**

hidden representation  \( h = h(x) = s(Wx + b) \)

Encoder: \( h \)

Decoder: \( g \)

input  \( x \)

reconstruction  \( r = g(h(x)) = s_d(W'h + b_d) \)

reconstruction error  \( L(x, r) \)
Auto-Encoders (AE)

Typical form

hidden representation \( h = h(x) = s(Wx + b) \)

Encoder: \( h \)

Decoder: \( g \)

input \( x \)

reconstruction \( r = g(h(x)) \)

reconstruction error \( L(x, r) \)

squared error: \( \| x - r \|^2 \)

or Bernoulli cross-entropy
Auto-Encoders (AE)

Typical form

hidden representation \[ h = h(x) = s(Wx + b) \]

Encoder: \[ h \]

Decoder: \[ g \]

input \( x \) \quad \text{reconstruction} \quad \text{reconstruction error} \quad L(x, r)

\[ L(x, r) \]

squared error: \[ \|x - r\|^2 \]

or Bernoulli cross-entropy

\[ s \] is often a logistic sigmoid

\[ r = g(h(x)) = s_d(W'h + b_d) \]
Auto-Encoder training

- Parameters $\theta = \{W, b, W', b_d\}$
- are learned from dataset $D$
- to minimize sum of reconstruction errors

$$J_{AE}(\theta) = \sum_{x \in D} L(x, g(h(\tilde{x})))$$
- with a gradient descent technique.
Over-complete representations?

- Traditional autoencoders used for dimensionality reduction.

- **Overcomplete** case yields useless solutions: identity mapping

- Need for alternative regularization/constraining (e.g. sparsity...)

Denoising auto-encoders: motivation

(Vincent, Larochelle, Bengio, Manzagol, ICML 2008)

- Denoising corrupted input is a vastly more challenging task.
- Even in widely over-complete case... it must learn intelligent encoding/decoding.
- Will encourage representation that is robust to small perturbations of the input.
Denoising auto-encoder (DAE)

$$h = h(x)$$

features:
(hidden representation)

Encoder:

Decoder:

$$r = g(h(x))$$

$$L(x, r)$$

input $$x$$

corrupted input $$\tilde{x}$$

noise $$q(\tilde{x}|x)$$

reconstruction error

reconstruction
Denoising auto-encoder (DAE)

(Vincent, Larochelle, Bengio, Manzagol, ICML 2008)

- Autoencoder training minimizes:

\[ J_{AE}(\theta) = \sum_{x \in D} L(x, g(h(\tilde{x}))) \]

- Denoising autoencoder training minimizes

\[ J_{DAE}(\theta) = \sum_{x \in D} \mathbb{E}_{q(\tilde{x}|x)} [L(x, g(h(\tilde{x})))] \]

Cannot compute expectation exactly

⇒ use stochastic gradient descent,

sampling corrupted inputs \( \tilde{x} | x \)
Denoising auto-encoder (DAE)
(Vincent, Larochelle, Bengio, Manzagol, ICML 2008)

- Autoencoder training minimizes:
  \[ J_{AE}(\theta) = \sum_{x \in D} L(x, g(h(\tilde{x}))) \]
- Denoising autoencoder training minimizes
  \[ J_{DAE}(\theta) = \sum_{x \in D} \mathbb{E}_{q(\tilde{x}|x)} [L(x, g(h(\tilde{x})))] \]

Cannot compute expectation exactly
\[ \Rightarrow \text{use stochastic gradient descent, sampling corrupted inputs } \tilde{x}|x \]

Possible corruptions \( q \):
- additive Gaussian noise
- zeroing pixels at random
- salt-and-pepper noise
- ...
Denoising auto-encoder (DAE)

(Vincent, Larochelle, Bengio, Manzagol, ICML 2008)

* Autoencoder training minimizes:

\[ J_{AE}(\theta) = \sum_{x \in D} L(x, g(h(\tilde{x}))) \]

* Denoising auto-encoder training minimizes

\[ J_{DAE}(\theta) = \sum_{x \in D} \mathbb{E}_q(\tilde{x} | x) [L(x, g(h(\tilde{x})))] \]

Cannot compute expectation exactly

\[ \Rightarrow \text{use stochastic gradient descent, sampling corrupted inputs } \tilde{x} | x \]

Possible corruptions \( q \):

- additive Gaussian noise
- zeroing pixels at random
- salt-and-pepper noise
- ...
Learned filters

a) Natural image patches  e.g.:
DAE: Geometric interpretation

- DAE learns to «project back» corrupted input onto manifold.
- Representation $h \approx$ location on the manifold

prior: examples concentrate near a lower dimensional “manifold”
Learned filters

b) MNIST digits  e.g.: 4 3 8 7

AE

DAE

Increasing noise

(d) Neuron A (0%, 10%, 20%, 50% corruption)

(e) Neuron B (0%, 10%, 20%, 50% corruption)
Encouraging representation to be insensitive to corruption

- DAE encourages **reconstruction** to be insensitive to input corruption
- Alternative: encourage **representation** to be **insensitive**

\[ J_{SCAE}(\theta) = \sum_{x \in D} L(x, g(h(x))) + \lambda E_{q(\tilde{x}|x)} \left[ \| h(x) - h(\tilde{x}) \|^2 \right] \]

- Reconstruction error
- Stochastic regularization term

- Tied weights i.e. \( W' = W^T \) prevent \( W \) from collapsing \( h \) to 0.
Encouraging representation to be insensitive to corruption

- DAE encourages **reconstruction** to be insensitive to input corruption
- Alternative: encourage representation to be insensitive

\[ J_{SCAE}(\theta) = \sum_{x \in D} E(x, g(h(x))) + \lambda \mathbb{E}_{q(\tilde{x}|x)} \left[ \| h(x) - h(\tilde{x}) \|^2 \right] \]  
  \text{Reconstruction error + stochastic regularization term}

- Tied weights i.e. \( W = W^T \) prevent \( W \) from collapsing \( h \) to 0.
From stochastic to analytic penalty

* SCAE stochastic regularization term: $\mathbb{E}_{q(\tilde{x}|x)} \left[ \| h(x) - h(\tilde{x}) \|^2 \right]$

* For small additive noise $\tilde{x}|x = x + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$

* Taylor series expansion yields $h(x + \epsilon) = h(x) + \frac{\partial h}{\partial x} \epsilon + \ldots$

* It can be showed that

$$\mathbb{E}_{q(\tilde{x}|x)} \left[ \| h(x) - h(\tilde{x}) \|^2 \right] \approx \sigma^2 \left\| \frac{\partial h}{\partial x}(x) \right\|_F^2$$

stochastic (SCAE)  analytic (CAE)
Contractive Auto-Encoder (CAE)

(Rifai, Vincent, Muller, Glorot, Bengio, ICML 2011)

\[ J_{\text{CAE}} = \sum_{x \in D} n \sum_{n} L(x, g(h(x))) + \lambda \left\| \frac{\partial h(x)}{\partial x} \right\|^2 \]

- Minimize
- For training examples, encourages both:
  - small reconstruction error
  - representation insensitive to small variations around example

Reconstruction error
analytic contractive term
Contractive Auto-Encoder (CAE)

(Rifai, Vincent, Muller, Glorot, Bengio, ICML 2011)

Minimize

\[ J_{\text{CAE}} = \sum_{x \in D} \mathcal{L}(x, g(h(x))) + \lambda \left\| \frac{\partial h(x)}{\partial x} \right\|^2 \]

- Reconstruction error
- Analytic contractive term

For training examples, encourages both:
- small reconstruction error
- representation insensitive to small variations around example
We defined \( h = h(x) = s(Wx + b) \)

Further suppose: \( s \) is an elementwise non-linearity
\( s' \) its first derivative.

Let \( J(x) = \frac{\partial h}{\partial x}(x) \)

\( J_j = s'(b + x^T W_j)W_j \) where \( J_j \) and \( W_j \) represent \( j^{th} \) row

CAE penalty is:
\[
\| J \|_F^2 = \sum_{j=1}^{d_h} s'(a_j)^2 \| W_j \|_F^2
\]

Compare to L2 weight decay:
\[
\| W \|_F^2 = \sum_{j=1}^{d_h} \| W_j \|_F^2
\]
Higher order Contractive Auto-Encoder (CAE+H)

(Rifai, Mesnil, Vincent, Muller, Bengio, Dauphin, Glorot; ECML 2011)

- CAE penalizes Jacobian norm
- We could also penalize higher order derivatives
- Computationally too expensive: second derivative is a 3-tensor, ...
- **Stochastic approach for efficiency:**
  Encourage Jacobian at $x$ and at $x + \epsilon$ to be the same.

$$J_{CAE+H} = \sum_{x \in D}^n L(x, g(h(x))) + \lambda \left\| \frac{\partial h}{\partial x}(x) \right\|^2$$

$$+ \gamma \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2)} \left[ \left\| \frac{\partial h}{\partial x}(x) - \frac{\partial h}{\partial x}(x + \epsilon) \right\|^2 \right]$$
Higher order Contractive Auto-Encoder (CAE+H)

(Rifai, Mesnil, Vincent, Muller, Bergio, Dauphin, Glorot; ECML 2011)

- CAE penalizes Jacobian norm
- We could also penalize higher order derivatives
- Computationally too expensive: second derivative is a 3-tensor, ...  
- **Stochastic approach for efficiency:** Encourage Jacobian at $x$ and at $x+\epsilon$ to be the same.

\[
\mathcal{J}_{\text{CAE+H}} = \sum_{x \in D} L(x, g(h(x))) + \lambda \left\| \frac{\partial h}{\partial x}(x) \right\|^2 \\
+ \gamma \mathbb{E}_{\epsilon \sim N(0, \sigma^2)} \left[ \left\| \frac{\partial h}{\partial x}(x) - \frac{\partial h}{\partial x}(x + \epsilon) \right\|^2 \right]
\]
Learned filters
Usefulness of these regularized autoencoders

- Suitable for **layerwise pretraining** of deep networks (unsupervised)
- Alternative to RBM
- **Won** *Unsupervised and Transfer Learning Challenge* (ICML 2012)
- Allows **modeling** and leveraging data **manifold** tangent space (*Manifold Tangent Classifier*, Rifai et al. NIPS 2011)
- DAE related to «dropouts» technique (Krizhevsky, Sutskever, Hinton 2012) that yields impressive object recognition performance.

\[ y = f(x) \]
DAE and CAE relationship to RBMs

- Same functional form as RBM:
  \( h(x) \) is expected hidden given visible
  \( g(h) \) is expected visible given hidden

- With linear reconstruction and squared error, DAE amounts to learning the following energy

\[
E(x; \theta) = -\frac{1}{2} \|x\|^2 + \sum_{j=1}^{d_h} \text{softplus} \left( \langle W_j, x \rangle + b_j \right)
\]

using a stochastic variant of Hyvärinen’s score matching inductive principle.

- Above energy closely related to free energy of Gaussian-binary RBM (identical for \( \sigma=1 \))
Score matching
(Hyvärinen 2005)

We want to learn a p.d.f.:
\[ p_{\theta}(x) = \frac{1}{Z(\theta)} e^{-E_\theta(x)} \]

with intractable partition function \( Z \)

Score matching: alternative inductive principle to max. likelihood

Find parameters that minimize objective:
\[
J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right)
\]
Score matching
my geometric interpretation

\[ J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right) \]
Score matching
my geometric interpretation

\[ J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right) \]

\[ \| J_E(x) \|^2 \]
Score matching
my geometric interpretation

\[ J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x} (x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2} (x) \right) \]

\[ \left\| J_E(x) \right\|^2 \]

First derivative encouraged to be small: ensures training points stay close to local minima of E
Score matching
my geometric interpretation

\[ J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right) \]

First derivative encouraged to be small: ensures training points stay close to local minima of E
Score matching
my geometric interpretation

$$J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right)$$

$\| J_E(x) \|^2$
First derivative encouraged to be small: ensures training points stay close to local minima of $E$

Encourage large positive curvature in all directions

$\text{Tr}(H_E(x))$

Laplacian
Score matching
my geometric interpretation

\[ J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right) \]

First derivative encouraged to be small: ensures training points stay close to local minima of \( E \)

Encourage large positive curvature in all directions

Encourage large positive curvature in all directions
Score matching
my geometric interpretation

\[ J_{SM}(\theta) = \sum_{x \in D} \left( \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right) \]

First derivative encouraged to be small: ensures training points stay close to local minima of \( E \)

Encourage large positive curvature in all directions

\[ \| J_E(x) \|^2 \]

\[ \text{Encourage large positive curvature in all directions} \]

\[ \text{First derivative encouraged to be small: ensures training points stay close to local minima of } E \]

\[ \text{Laplacian} \]

\[ \text{Tr}(H_E(x)) \]

\[ \text{sharply peaked density} \]
Score matching variants

Original score matching (Hyvärinen 2005):

\[ J_{SM}(\theta) = \sum_{x \in D} \left( \frac{1}{2} \left\| \frac{\partial E}{\partial x}(x) \right\|^2 - \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \right) \]

Regularized score matching (Kingma & LeCun 2010):

\[ J_{SM_{\text{reg},\lambda}}(\theta) = J_{SM} + \sum_{x \in D} \lambda \sum_{i=1}^{d} \frac{\partial^2 E}{\partial x_i^2}(x) \]

Denoising score matching (Vincent 2011)

\[ J_{DSM,\sigma} = \sum_{x \in D} \left( \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)} \left[ \frac{1}{2} \left\| \frac{\partial E}{\partial x}(x + \epsilon) - \frac{1}{\sigma^2} \epsilon \right\|^2 \right] \right) \]

Analytic

Analytic

Stochastic
Analytic v.s. stochastic?

a) Analytic approximation of stochastic perturbation

- Equiv. to tiny perturbations: does not probe far away
- Potentially more efficient. Ex: CAE’s Jacobian penalty probes sensitivity in all $d$ directions in $O(dh^d)$. With DAE or SCAE it would require encoding $d$ corrupted inputs: $O(dh^d^2)$

b) Stochastic approximation of analytic criterion

- Can render practical otherwise computationally infeasible criteria. Ex: CAE+H
- Less precise, more noisy

CAE+H actually leverages both
Thank you to past and current students who did most of the hard work

and to my colleague and mentor

Questions?