Dictionary learning and autoencoders

Michael Bianco
University of California San Diego

UCSD ECE228
April 27, 2020
Applications of machine learning in acoustics

- The applications and successes of ML in acoustics are numerous, some examples are:
  - Source localization
  - Tomography
  - Sound classification
  - Speech enhancement/de-reverberation

Source localization

Medical tomography (ultrasound)
In ML, we are often interested in training a model to produce a desired output given inputs,

\[ y = f(x) + \epsilon \]

- **Input** \( x \in \mathbb{R}^N, \) \( N \) features
- **Output** \( y \in \mathbb{R}^P, \) \( P \) outputs

- **Supervised learning**: the \( P \) outputs have labelled examples (response variables \( y \))
- **Unsupervised learning**: there are no labels. The goal is to find interesting properties from \( x \), as an autoencoder \( \tilde{x} = f(x) \)
- ….. and we *train* the model
- Most relevant to scenarios where data can’t be explained by simple principles
Supervised learning: the P outputs have labelled examples (response variables $y$)

Unsupervised learning: there are no labels. The goal is to find interesting properties from $x$, as an autoencoder $\tilde{x} = f(x)$

Linear/polynomial regression

$$y = X^T w + b$$

Principal components analysis (PCA)

$$X = Dw, \ D = [d_1, d_2], \ d_1 \perp d_2$$
Supervised and unsupervised learning

**Supervised learning**: the P outputs have labelled examples (response variables $y$)

**Unsupervised learning**: there are no labels. The goal is to find interesting properties from $x$, as an autoencoder $\hat{x} = f(x)$

---

**Linear/polynomial regression**

$y = X^T w + b$

---

**Principal components analysis (PCA)**

$X = Dw, \; D = [d_1, d_2], \; d_1 \perp d_2$

---

**Other examples:**
- Support vector machines
- Neural networks (classification/regression)
- Nearest neighbors

---

**Other examples:**
- Neural networks (autoencoders)
- K-means (clustering)
- Dictionary learning
Supervised learning: neural networks

Neural network classification of MNIST handwritten digits

- 784 inputs (28x28 pixels)
- 1 output (label [0,9])
- 2 layer NN, 1 hidden
- 25 hidden units (ReLU)
Supervised learning: neural networks

Neural network classification of MNIST handwritten digits

- 784 inputs (28x28 pixels)
- 1 output (label [0,9])
- 2 layer NN, 1 hidden
- 25 hidden units (ReLU)

Learned representation
Unsupervised learning: principal components

**PCA is unsupervised learning**: goal is to find interesting properties from \( x \), e.g. \( \tilde{x} = f(x) \)

Uses can be feature learning, visualization

**Eigen-faces (PCA)**

- 100 faces
- 36 PCs
- 100 PCs

$$X = Dw, \ D = [d_1, d_2], \ d_1 \perp d_2$$

**Eigen- sound speed profiles**

Bianco 2017, JASA
Unsupervised learning: principal components

PCA is unsupervised learning: goal is to find interesting properties from \( x \), e.g. \( \tilde{x} = f(x) \)

Uses can be feature learning, visualization

Eigen-faces

100 faces  36 PCs  100 PCs

Eigen- sound speed profiles

\[ X = Dw, \quad D = [d_1, d_2], \quad d_1 \perp d_2 \]

Bianco 2017, JASA
Hand-engineering vs. Learning representations

In Image processing this has been done:

1) Hand-engineered design: Consciously figure out exactly how to manipulate symbolic representations to perform the task and then tell the computer in detail what to do.

2) Learning: Show computers lots of examples of input with desired outputs. Let the computer learn how to map inputs to outputs using a general purpose learning procedure.
1) Hand-engineered design: wavelets, predefined functions which work well for modeling many different signals

2) Learning: learn dictionaries to represent specific signals by training on signal examples

Noisy Image (22.1307 dB, σ=20)

Denoised Image Using Adaptive Dictionary (30.8295 dB)

Hand-engineered, Discrete cosine transform (DCT)

Dictionary learned on corpus of images

Elad 2006
Why dictionary learning?

Denoising

Noisy Image (22.1307 dB, σ=20)

Denoised Image Using Adaptive Dictionary (30.8295 dB)

Inpainting (a.k.a. matrix completion)

Mairal 2009

\[
\min_{D,X} \frac{1}{N} \sum_{n=1}^{N} \|M_n(y_n - Dx_n)\|_2^2 \quad \text{subject to } \|x_n\|_0 \leq T
\]

Elad 2006

\[
\min_{D,X} \frac{1}{N} \sum_{n=1}^{N} \|y_n - Dx_n\|_2^2 \quad \text{subject to } \|x_n\|_0 \leq T
\]
Dictionary learning and sparsity

- Dictionary learning obtains "optimal" sparse modeling dictionaries directly from data.
- Dictionary learning was developed in neuroscience (a.k.a. sparse coding) to help understand mammalian visual cortex structure.
- Assumes (1) **Redundancy in data:** image patches are repetitions of a few elemental shapes; and (2) **Sparsity:** each patch is represented with few atoms from dictionary.

"Natural" images, patches shown in **magenta**

Learn dictionary \( \mathbf{D} \) describing \( \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_I] \)

- Each patch is signal \( \mathbf{y}_i \)
- Set of all patches \( \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_I] \)

Sparse model for patch \( \mathbf{y}_i \) composed of few atoms from \( \mathbf{D} \)

\[
\hat{\mathbf{x}}_i = \arg \min_{\mathbf{x}_i} \| \mathbf{y}_i - \mathbf{D} \mathbf{x}_i \|_2 \quad \text{subject to} \quad \| \mathbf{x}_i \|_0 \leq T
\]

\[
\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \ldots
\]

Olshausen 2009
Sparse models and dictionaries

- Sparse modeling assumes each signal model can be reconstructed from a few vectors from a large set of vectors, called a dictionary $\mathbf{D}$.

\[
\mathbf{y} = \mathbf{D} \mathbf{x} + \mathbf{\epsilon}
\]

- Sparse objective: $\hat{x}_i = \arg \min_{x_i} \|y_i - D x_i\|_2$ subject to $\|x_i\|_0 \leq T$

\[
\begin{align*}
\text{n x 1} & \quad \text{measurements} \\
n \times Q & \quad \text{dictionary} \\
T \text{ nonzero} & \quad \text{entries, } T \ll Q \\
Q \times 1 & \quad \text{sparse signal}
\end{align*}
\]
Unsupervised learning used for discovering ‘hidden’ structure in data, no labels

Clustering in 2D

“Dictionary learning”

Dictionary learned from image: each "atom" is high-dimensional cluster

Unsupervised learning and dictionary learning

E.g. K-means

E.g. K-SVD

Elad 2006
Unsupervised learning used for discovering ‘hidden’ structure in data, no labels

“Dictionary learning”

Clustering in 2D

Data: image patches

Dictionary learned from image:
each "atom" is high-dimensional cluster

Unsupervised learning and dictionary learning

e.g. K-means

e.g. K-SVD

Elad 2006
Just have features $\{(x_1^1, x_2^1), (x_1^2, x_2^2), (x_1^3, x_2^3)\}$
**K-means**

- **Input**: Points $x_1, \ldots, x_N \in \mathbb{R}^p$; integer $K$
- **Output**: “Centers”, or representatives, $\mu_1, \ldots, \mu_K \in \mathbb{R}^p$
- Output also $z_1, \ldots, z_N \in \mathbb{R}^K$

**Goal**: Minimize average squared distance between points and their nearest representatives:

- $\text{cost}(\mu_1, \ldots, \mu_K) = \sum_{n=1}^{N} \min_{j} \|x_n - \mu_j\|$

The centers carve $\mathbb{R}^p$ up into $k$ convex regions: $\mu_j$’s region consists of points for which it is the closest center.
K-means

\[ J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| \mathbf{x}_n - \mathbf{\mu}_k \|^2 \]  

(9.1)

Solving for \( r_{nk} \)

\[ r_{nk} = \begin{cases} 
1 & \text{if } k = \arg \min_j \| \mathbf{x}_n - \mathbf{\mu}_j \|^2 \\
0 & \text{otherwise.} 
\end{cases} \]  

(9.2)

Differentiating for \( \mathbf{\mu}_k \)

\[ \sqrt{2} \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \mathbf{\mu}_k) = 0 \]  

(9.3)

which we can easily solve for \( \mathbf{\mu}_k \) to give

\[ \mathbf{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}} \]  

(9.4)
Consider first the determination of the mean of all of the data points.

The two phases of re-assigning data points to clusters and re-computing the cluster centres are repeated until there is no further change in the assignments (or until some maximum number of iterations is exceeded). Because each phase reduces the value of the objective function, such that each of the variables has zero mean and, therefore, it may converge to a local rather than global minimum of the mean square error criterion.

The terms involving different data are repeated in turn until there is no further change in the assignments.

For this example, we have chosen a linear re-scaling of the variables to unit standard deviation. For this example, we have chosen a linear re-scaling of the variables to unit standard deviation. The convergence of the algorithm was studied by MacQueen (1967).

In other words, we simply assign the data point to the closest cluster centre. More formally, this can be expressed as

\[
 r_{nk} = \begin{cases} 
 1 & \text{if } k = \arg \min_{j} \| x_n - \mu_j \|^2 \\
 0 & \text{otherwise.} 
\end{cases}
\]

\[
 \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}. 
\]

\[
 y = D \times \text{PCA} = \text{local min}. 
\]
Dictionary learning objective:

\[
\begin{align*}
\min_{D, x} & \quad \frac{1}{N} \sum_{n=1}^{N} \| y_n - D x_n \|_2^2 \\
\text{subject to} & \quad \| x_n \|_0 \leq T
\end{align*}
\]

Can be solved with alternating minimization steps:

1. Solve for sparse coefficients X
2. Solve for dictionary D

… repeat until convergence.

2D random data

K-means

K-SVD (T=1)

Error vs. Iteration
Task: Given travel times, estimate regional phase speed distribution

\[ d = A m + n, \]

- **d**: M travel times
- **A**: "Tomography matrix": ray paths through the discretized map
- **m**: N-pixel slowness image

Distance = speed \times\ time

Slowness = 1/speed

Low Velocity Region~Sedimentary basins A: San Joaquin, B: Ventura, C: L.A., D: Salton Sea, E: Peninsular range, F: Sierra Nevada

Slowness map and measurements
- stations in red
- rays in blue
Tomography: checkerboard dictionary example

\[ \mathbf{y} = \mathbf{R}_i \mathbf{s} = \mathbf{D} \mathbf{x}_i \]

10x10 pixel patches

\[ \mathbf{D} \in \mathbb{R}^{n \times Q} \]
\[ \mathbf{R}_i \in \{0, 1\}^{n \times N} \]

\[ \hat{\mathbf{x}}_i = \arg \min_{\mathbf{x}_i} ||\mathbf{y}_i - \mathbf{Dx}_i||_2 \text{ subject to } ||\mathbf{x}_i||_0 \leq T \]

\[ \mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \ldots \]

"Travel time tomography with adaptive dictionaries"
LST slowness image and sampling

Slowness map and sampling:
- Discrete slowness map $N=\mathcal{W}_1 \times \mathcal{W}_2$ pixels
- $I$ overlapping $\sqrt{n} \times \sqrt{n}$ pixel patches
- $M$ straight-ray paths

Tomography matrix (straight ray)
- $A \in \mathbb{R}^{M \times N}$
- $D \in \mathbb{R}^{n \times Q}$
  - $Q \ll I$

Bayesian formulation

"Local" objective
$$\min_{\mathcal{D}, \mathcal{X}} \frac{1}{I} \sum_{i=1}^{I} \left\| \mathbf{R}_i \mathbf{s}_s - \mathbf{D} \mathbf{x}_i \right\|_2^2 \text{ subject to } \left\| \mathbf{x}_i \right\|_0 = T$$

"Global" objective
$$\mathbf{t} = A \mathbf{s}_g + \epsilon \quad \hat{\mathbf{s}}_g = \arg\min_{\mathbf{s}_g} \left\| \mathbf{t} - A \mathbf{s}_g \right\|_2^2 + \lambda_1 \left\| \mathbf{s}_g - \mathbf{s}_s \right\|_2^2.$$
Formulation of LST and algorithm

Bayesian MAP objective:

\[
\{\hat{s}_g, \hat{s}_s, \hat{X}\} = \arg \min_{s_g, s_s, X} \left\{ \frac{1}{\sigma^2_c} \| t - A s_g \|_2^2 + \frac{1}{\sigma^2_g} \| s_g - s_s \|_2^2 + \frac{1}{\sigma^2_{p,i}} \sum_i \| D x_i - R_i s_s \|_2^2 \right\}
\]

subject to \( \| x_i \|_0 = T \forall i. \)

Solution via block-coordinate descent

- **Global model**: the global slowness is solved as

  \[
  \hat{s}_g = \arg \min_{s_g} \| t - A s_g \|_2^2 + \lambda_1 \| s_g - s_s \|_2^2, \quad \lambda_1 = \left(\frac{\sigma_c}{\sigma_g}\right)^2
  \]

- **Local model**: sparse coding and dictionary learning, decoupled from MAP objective

  \[
  \bar{x}_i = \arg \min_{x_i} \| D x_i - R_i \hat{s}_g \|_2^2 \text{ subject to } \| x_i \|_0 = T \quad (s_s = \hat{s}_g)
  \]

  Dictionary learning by iterative thresholding and signed K-means (ITKM) algorithm, Schnass 2015

  \[
  \max_D \sum_i \max_{|K|=T} \| D^T K Y_i \|_1,
  \]

- The sparse slowness is then solved from

  \[
  \hat{s}_s = \arg \min_{s_s} \lambda_2 \| \hat{s}_g - s_s \|_2^2 + \sum_i \| D \bar{x}_i - R_i s_s \|_2^2, \quad \lambda_2 = \left(\frac{\sigma_p}{\sigma_g}\right)^2
  \]

  "Slowness at pixel n"

  \[
  \hat{s}_s = \left(\lambda_2 I + \sum_i R_i^T R_i\right)^{-1} \left(\lambda_2 \hat{s}_g + \sum_i R_i^T \bar{x}_i\right)
  \]

  \[
  \hat{s}_{s,n} = \frac{\lambda_2 \hat{s}_{g,n} + b_n s_{p,n}}{\lambda_2 + b_n}
  \]

  \[
  b = \text{diag}\left(\sum_i R_i^T R_i\right) \in \mathbb{Z}^N \quad s_{p,n} = p_n / b_n
  \]
LST inversion example (Bianco and Gerstoft 2018, IEEE TCI), Learned dictionary

\[ \min_{y-Dx} \| y - D \|_2^2 \text{ s.t. } x \perp \perp 0 \]

1. Find dictionary \( D \)
2. Update dictionary \( D \) = \( D + \alpha C \)

LST inversion of checkerboard image

LST Code available online: https://github.com/mikebianco
Autoencoder NNs

- Autoencoder are neural networks designed to approximate their input
- Traditionally used for dimension reduction and feature learning: e.g. applied to image denoising – so called “denoising” autoencoder
- Smaller code than input forces NN to learn salient features
- Autoencoders with linear activation functions in hidden units are analogous to PCA, dictionary learning
- Nonlinear activation functions increase representational power of autoencoders
Linear autoencoder example

- Linear activation (2 layers), simple algebra:

  \[ z_1 = w_{11} \times x_1 + w_{21} \times x_2 + ... \]
  \[ z_2 = w_{12} \times x_1 + w_{22} \times x_2 + ... \]
  \[ z = W^T x \]
  \[ \hat{x} = Wz \]
  \[ \hat{x} = WW^T x \]

- Linear case has analytic solution (linear regression)
- Can also solve with back-prop

Consider least squares loss, single hidden unit \( z \), find \( w \)

\[
L(X, \hat{X}) = \| \hat{X} - w w^T X \|^2_2
\]

\[
\hat{X} = WW^T x
\]
Linear autoencoder example

- Linear activation (2 layers), simple algebra:

  \[ z_1 = w_{11} x_1 + w_{21} x_2 + \ldots \]
  \[ z_2 = w_{12} x_1 + w_{22} x_2 + \ldots \]

  \[ z = W^T x \]
  \[ \hat{x} = Wz \]
  \[ \hat{x} = WW^T x \]

  - Linear case has analytic solution (linear regression)
  - Can also solve with back-prop

Example with sinusoid+noise

PCA-like features in weights
Autoencoders powerful with non-linearities

- Linear activation (2 layers), simple algebra:

\[ z_1 = w_{11} \times x_1 + w_{21} \times x_2 + \ldots \]
\[ z_2 = w_{12} \times x_1 + w_{22} \times x_2 + \ldots \]
\[ z = W^T x \rightarrow z_n = h(w_n^T x) \]
\[ \hat{x} = Wz \rightarrow \hat{x}_m = h(w_m z) \]

- Find weights with backprop

Speech denoising with autoencoder

Lu et al. 2013
Deep generative modeling… autoencoders?

• In acoustics (and many fields) there is ample data for learning, but few LABELS!
• Deep generative models: neural network-based models capable of generating samples of observed data
• Deep generative models are good for representation learning
  – By learning to generate data, learn explanatory data distributions

Faces generated by generative adversarial network (GAN): None of these people exist!

StyleGAN2, Karras 2019 (NVIDIA)
Variational autoencoders

- One weakness of GANs is that the latent model $p(z|x)$ is not explicit
  - Cannot infer latent code that produced sample
- Variational autoencoders (VAEs) explicitly encode, $q(x|z)$, and generate, $p(x|z)$, data

$$B(\theta, \phi; x_n) = \mathbb{E}[\log p_\theta(x_n|z)] - KL(q_\phi(z|x_n)||p_\theta(z))$$

Visualization of 2D MNIST manifold learned with VAE

Kingma 2014
Variational autoencoders: Probabilistic spin on AEs

- Variational autoencoders (VAEs) explicitly encode, $q(x|z)$, and generate, $p(x|z)$, data
- First interested in finding latent code for input $p_{\theta}(z|x)$
- From Bayes’ rule we can see it depends on intractable evidence

$$p_{\theta}(z|x) = \frac{p_{\theta}(z, x)}{p_{\theta}(x)} \quad p(x) = \int p(z, x)dz$$

- Propose approximate posterior $q_{\phi}(z|x) \approx p_{\theta}(z|x)$

$$\{\phi, \theta\} = \arg\min_{\theta, \phi} KL(q_{\phi}(z|x) || p_{\theta}(z|x))$$

$$KL = \int q_{\phi}(z|x) \log \frac{q_{\phi}(z|x)}{p_{\theta}(z|x)} dz$$

$$= \mathbb{E}_{q_{\phi}(z|x)} \left[ \log q_{\phi}(z|x) \right] - \mathbb{E}_{q_{\phi}(z|x)} \left[ \log p_{\theta}(z|x) \right] + \mathbb{E} \log p_{\theta}(x)$$

Autoencoder network (approximates input)
Variational autoencoders (VAEs) explicitly encode, $q(x|z)$, and generate, $p(x|z)$, data

First interested in finding latent code for input $p_\theta(z|x)$

From Bayes’ rule we can see it depends on intractable evidence

$$p_\theta(z|x) = \frac{p_\theta(z, x)}{p_\theta(x)} \quad p(x) = \int p(z, x)dz$$

Propose approximate posterior $q_\phi(z|x) \approx p_\theta(z|x)$

$$\{\phi, \theta\} = \arg \min_{\theta, \phi} \text{KL}(q_\phi(z|x) || p_\theta(z|x))$$

$$\text{KL}(q_\phi(z|x) || p_\theta(z|x)) = \mathbb{E}[\log q_\phi(z|x)] - \mathbb{E}[\log p_\theta(z, x)] + \log p_\theta(x)$$

$$= -\text{ELBO} + \log p_\theta(x)$$

ELBO = evidence lower bound, $\text{ELBO} \leq \log p_\theta(x)$
Variational autoencoders: Probabilistic spin on AEs

- Variational autoencoders (VAEs) explicitly encode, \( q(x|z) \), and generate, \( p(x|z) \), data
- First interested in finding latent code for input \( p_\theta(z|x) \)
- From Bayes’ rule we can see it depends on intractable evidence

\[
p_\theta(z|x) = \frac{p_\theta(z,x)}{p_\theta(x)} \quad p(x) = \int p(z,x)dz
\]

- Propose approximate posterior \( q_\phi(z|x) \approx p_\theta(z|x) \)

\[
\{\phi, \theta\} = \arg \min_{\theta, \phi} \text{KL}(q_\phi(z|x) || p_\theta(z|x))
\]

\[
= \mathbb{E}\left[ \log q_\phi(z|x) \right] - \mathbb{E}\left[ \log p_\theta(z,x) \right] + \log p_\theta(x)
= -\text{ELBO} + \log p_\theta(x)
\]

- ELBO = evidence lower bound, \( \text{ELBO} \leq \log p_\theta(x) \)
- Minimize \(-\text{ELBO}\) equivalent to minimizing KL

\[
B(\theta, \phi; x) = \mathbb{E}\left[ \log q_\phi(z|x) - \log p_\theta(x|z)p_\theta(z) \right]
= \mathbb{E}\left[ \log \frac{q_\phi(z|x)}{p_\theta(z)} - \log p_\theta(x|z) \right]
\leq \text{KL}(q_\phi(z|x) || p_\theta(z)) + \mathbb{E}\left[ \log p_\theta(x|z) \right]
\]
Variational autoencoders (VAEs) explicitly encode, $q(x|z)$, and generate, $p(x|z)$, data.

First interested in finding latent code for input $p_\theta(z|x)$.

Minimize $-\text{ELBO}$ equivalent to minimizing $\text{KL}$.

$$B(\theta, \phi; x) = \mathbb{E} \left[ \log q_\phi(z|x) - \log p_\theta(x|z)p_\theta(z) \right]$$

$$= \mathbb{E} \left[ \log \frac{q_\phi(z|x)}{p_\theta(z)} - \log p_\theta(x|z) \right]$$

$$= \text{KL}(q_\phi(z|x)||p_\theta(z)) - \mathbb{E} \left[ \log p_\theta(x|z) \right]$$

$q_\phi(z|x) = \mathcal{N}(z; \mu, \Sigma)$

Latent embedding of MNIST digits

Autoencoder network (approximates input)

Inference $q_\phi(z|x)$

Generation $p_\theta(x|z)$
Summary

- Machine learning and deep learning offer many interesting research opportunities in acoustics and beyond
- Important to recognize limitations of machine learning
  - Don’t reinvent wheel
  - Future research can use ML to build on insights from physical models/domain knowledge

Our recent review of machine learning in acoustics is available free online:
https://doi.org/10.1121/1.5133944

Machine learning in acoustics: Theory and applications

Michael J. Bianco,1,a) Peter Gerstoft,1 James Traer,2 Emma Ozanich,1 Marie A. Roch,3 Sharon Gannot,4 and Charles-Alban Deledalle5
1 Scripps Institution of Oceanography, University of California San Diego, La Jolla, California 92093, USA
2 Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
3 Department of Computer Science, San Diego State University, San Diego, California 92182, USA
4 Faculty of Engineering, Bar-Ilan University, Ramat-Gan 5290002, Israel
5 Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, California 92093, USA

(Received 9 May 2019; revised 23 September 2019; accepted 14 October 2019; published online 27 November 2019)

Acoustic data provide scientific and engineering insights in fields ranging from biology and communications to ocean and Earth science. We survey the recent advances and transformative potential of machine learning (ML), including deep learning, in the field of acoustics. ML is a broad family of techniques, which are often based in statistics, for automatically detecting and utilizing

Comprehensive review, including:
- Introduction to ML theory
- Deep learning
- Source localization in speech processing
- Source localization in ocean acoustics
- Bioacoustics
- Human perception of sound