Challenges motivating deep learning

Sargur N. Srihari
srihari@cedar.buffalo.edu

Topics In Machine Learning Basics

1. Learning Algorithms
2. Capacity, Overfitting and Underfitting
3. Hyperparameters and Validation Sets
4. Estimators, Bias and Variance
5. Maximum Likelihood Estimation
6. Bayesian Statistics
7. Supervised Learning Algorithms
8. Unsupervised Learning Algorithms
9. Stochastic Gradient Descent
10. Building a Machine Learning Algorithm
11. Challenges Motivating Deep Learning

Topics in “Motivations”

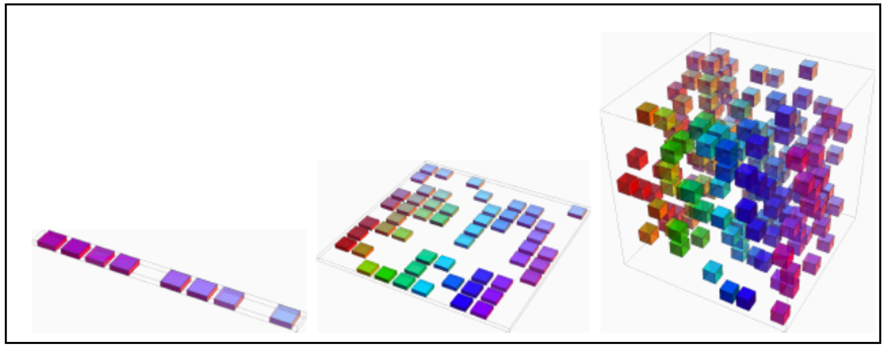
- Shortcomings of conventional ML

1. The curse of dimensionality
2. Local constancy and smoothness regularization
3. Manifold learning

Challenges Motivating DL

- Simple ML algorithms work very well on a wide variety of important problems
- However they have not succeeded in solving central problems of AI, such as recognizing speech and recognizing objects
- DL motivated by failure of traditional algorithms to generalize well on such AI tasks
  - Where generalizing to new examples becomes exponentially more difficult with high dimensionality
  - Traditional ML is insufficient to
  - Which also impose high computational costs

Curse of dimensionality



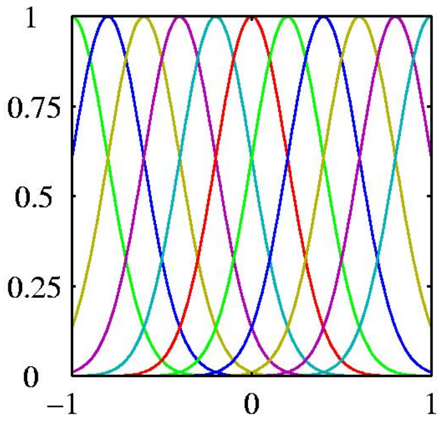
- No of possible distinct configurations of a set of variables increases exponentially with no of variables
  - – Poses a statistical challenge
- Ex: 10 regions of interest with one variable
  - – We need to track 100 regions with two variables
  - – 1000 regions with three variables

Example of Basis Function Features

Linear regression with basis functions
Model has the likelihood function

$$p(\mathrm{t} \mid X, \mathrm{w}, \beta) = \prod_{n=1}^{N} N\left(t_n \mid \mathrm{w}^T \phi(\mathrm{x}_n), \beta^{-1}\right)$$

Radial basis functions



- If we divide each feature into 2 cells
- Position two basis functions per cell
- No of basis functions needed would be 2^{D}

Local Constancy & Smoothness Regularization

- Prior beliefs
  - To generalize well ML algorithms need prior beliefs
    - Form of probability distributions over parameters
    - Influencing the function itself, while parameters are influenced only indirectly
    - Algorithms biased towards preferring a class of functions
      - These biases may not be expressed in terms of a probability distribution
- Most widely used prior is smoothness
  - Also called local constancy prior
  - States that the function we learn should not change very much within a small region

Local Constancy Prior

- Function should not change very much within a small region
- Many simpler algorithms rely exclusively on this prior to generalize well
  - Thus fail to scale statistical challenges in AI tasks
- Deep learning introduces additional (explicit and implicit) priors in order to reduce generalization error on sophisticated tasks
- We now explain why smoothness alone is insufficient

Specifying smoothness

- Several methods to encourage learning a function f^* that satisfies the condition

$$f^*(\boldsymbol{x}) \approx f^*(\boldsymbol{x} + \varepsilon)$$

  - For most configurations $\boldsymbol{x}$ and small change ε
- If we know a good answer for input $\boldsymbol{x}$ then that answer is good in the neighborhood of $\boldsymbol{x}$
- An extreme example is k-nearest neighbor
  - Points having the same set of nearest neighbors all have the same prediction
  - For $k=1$, no of regions $\leq$ no of training examples

Kernel machines and smoothness

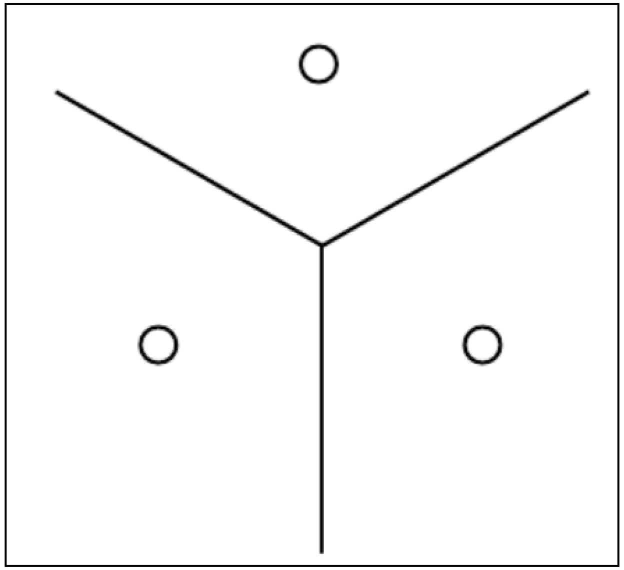
- Kernel machines interpolate between training set outputs associated with nearby training examples
- With local kernels: $k(\boldsymbol{u},\boldsymbol{v})$ is large when $\boldsymbol{u}=\boldsymbol{v}$ and decreases as $\boldsymbol{u}$ and $\boldsymbol{v}$ grow further apart
- Can be thought of as a similarity function that performs template matching
  - By measuring how closely test example $\boldsymbol{x}$ resembles training example $\boldsymbol{x}^{(i)}$
- Much of deep learning is motivated by limitations of template matching

Decision Trees and Smoothness

- Also suffers from exclusively smoothness-based learning
  - They break input space into as many regions as there are leaves and use a separate parameter in each region
  - For n leaves, at least n training samples are required
  - Many more needed for statistical confidence

No. of examples and no. of regions

- All of the above methods require:
  - $O(k)$ regions need $O(k)$ examples;
  - $O(k)$ parameters with $O(1)$ parameters associated with $O(k)$ regions
- Nearest-neighbor : each training sample (circle) defines at most one region
  - y value associated with each example defines the output for all points within region

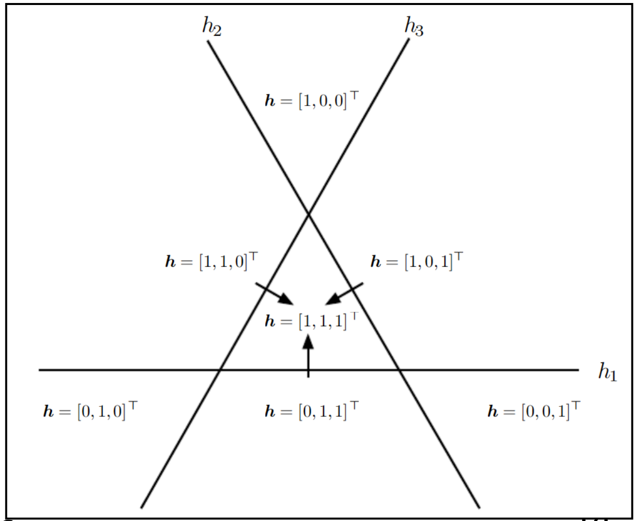


More regions than examples

- Suppose we need more regions than examples
- Two questions of interest
  1. Is it possible represent a complicated function efficiently?
  2. Is it possible for the estimated function to generalize well for new inputs?
- Answer to both is yes
  - $O(2^k)$ regions can be defined with $O(k)$ examples
    - By introducing dependencies between regions through assumptions on data generating distribution

Core idea of deep learning

- Assume data was generated by composition of factors, at multiple levels in a hierarchy
  - Many other similarly generic assumptions
- These mild assumptions allow exponential gain in no of samples and no of regions
  - An example of a distributed representation is a vector of n binary features
    - It can take 2^n configurations
      - Whereas in a symbolic representation, each input is associated with a single symbol (or category)
      - Here h_1, h_2 and h_3 are three binary features


h_2
h_3
$\boldsymbol{h} = [1,0,0]^\top$
$\boldsymbol{h} = [1,1,0]^\top$
$\boldsymbol{h} = [1,0,1]^\top$
$\boldsymbol{h} = [1,1,1]^\top$
h_1
$\boldsymbol{h} = [0,1,0]^\top$
$\boldsymbol{h} = [0,1,1]^\top$
$\boldsymbol{h} = [0,0,1]^\top$

Manifold Learning

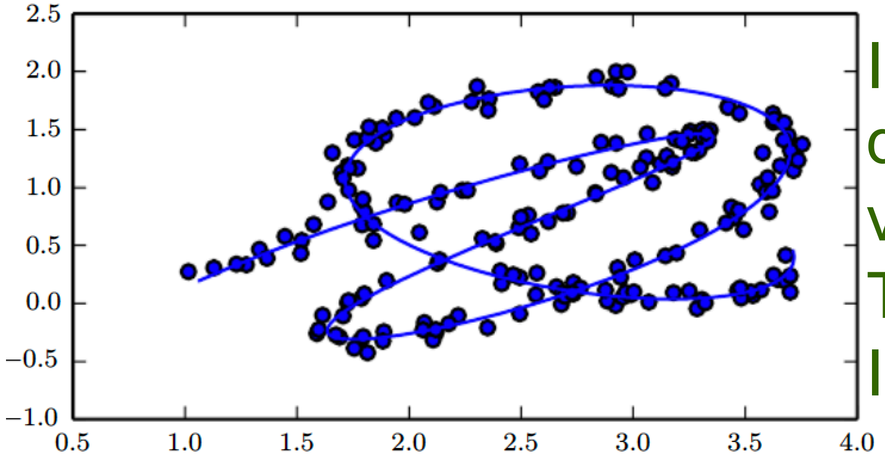
- An important idea underlying many ideas in machine learning
- A manifold is a connected region
  - Mathematically it is a set of points in a neighborhood
  - It appears to be in a Euclidean space
    - E.g., we experience the world as a 2-D plane while it is a spherical manifold in 3-D space

Manifold in Machine Learning

- Although manifold is mathematically defined, in machine learning it is loosely defined:
  - A connected set of points that can be approximated well by considering only a small no of degrees of freedom embedded in a higher-dimensional space

Training data lying near a 1-D Manifold in a 2-D space

The solid line indicates the underlying manifold that the learner should infer



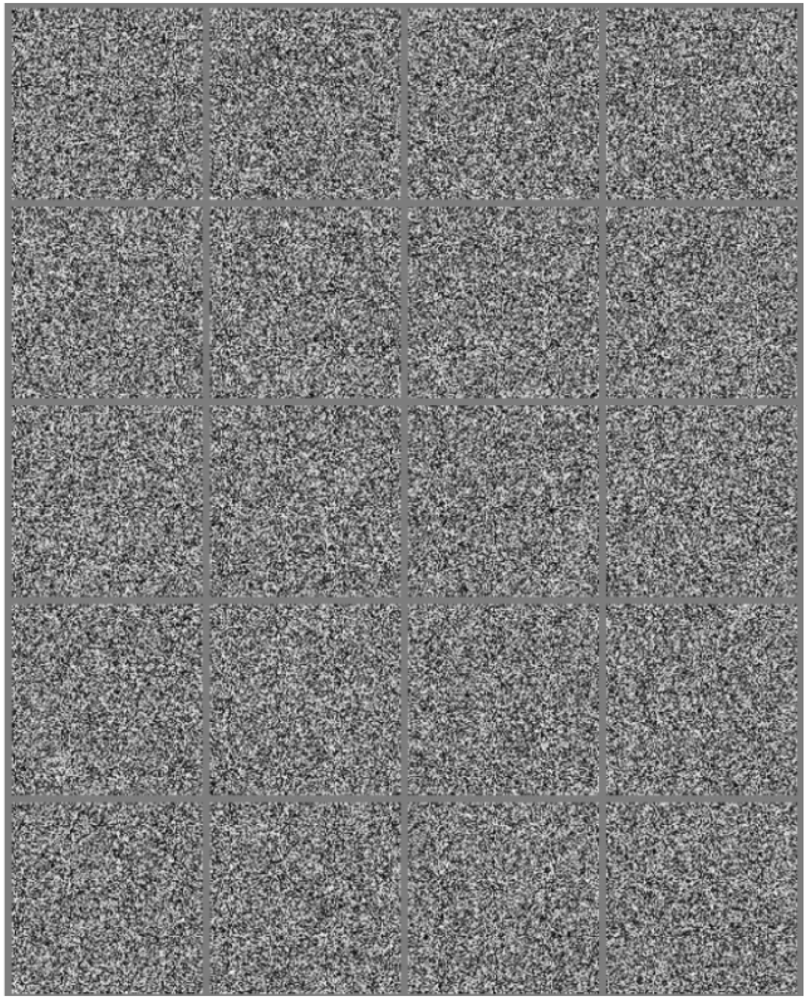
In machine learning we allow the dimensionality of the manifold to vary from one point to another. This often happens when a manifold Intersects itself, as in a figure-eight

Manifold learning surmounts R^n

- It is sometimes hopeless to learn functions with variations across all of R^n
- Manifold learning algorithms surmount this obstacle by assuming most of R^n consists of invalid inputs
  - And that intersecting inputs occur only along the manifolds
- Introduced for continuous data and in unsupervised learning, the probability concentration idea can be generalized to discrete and unsupervised settings

Manifold hypothesis for Images

- Manifold assumption is justified since:
- Distributions are highly concentrated
  - Uniformly sampled points look like static noise, never structured
    - Although there is a non-zero probability of generating a face, it is never observed

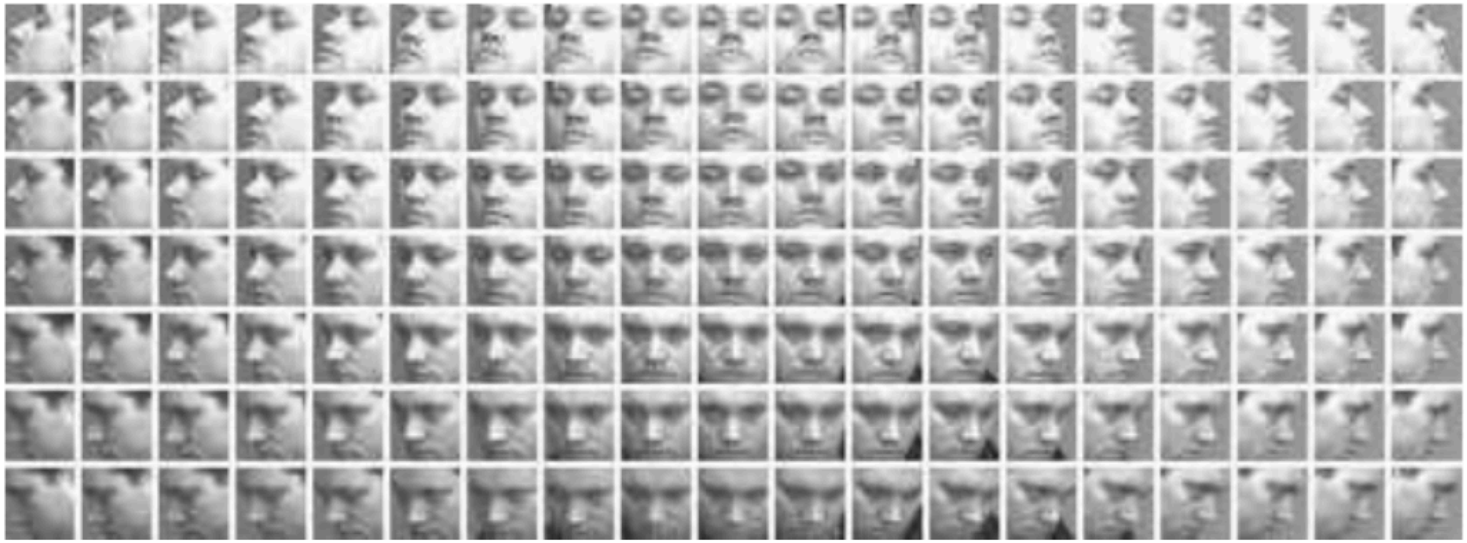


Manifold justified in Text domain

- If you generate a document by randomly generating text, it is a near zero probability of generating meaningful text
- Natural language sequences occupy a small volume of total space of sequences of letters

Manifolds traced by transformations

- Manifolds can be traced by making small transformations
- Manifold structure of a dataset of human faces



Manifolds discovered for Human Faces

- Variational autoencoder discovers underlying two-dimensional coordinate system:
  1. Rotation
  2. Emotion