High Dimensional Approximation

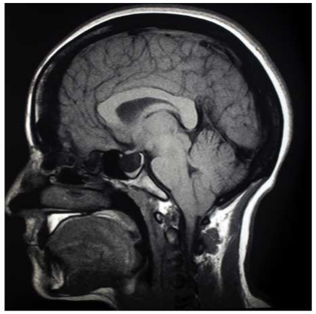
Ronald DeVore

Texas A& M University

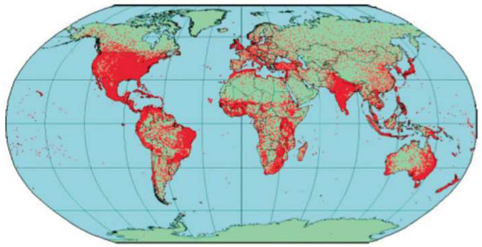
Why High Dimension?

- Some of the most pressing scientific problems challenge our computational ability
  - Atmospheric modeling: predicting climate change
  - Monitoring threat activities
  - Contaminant transport
  - Optimal engineering design
  - Medical diagnostics
  - Modeling the internet
  - Option pricing, bond valuation
  - ....

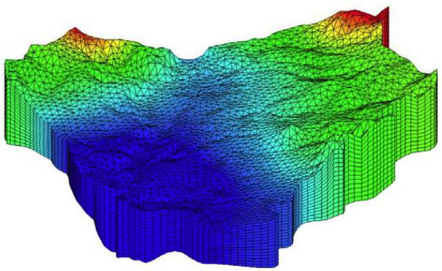
Your Favorite Application



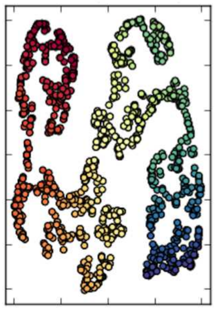
MRI



Global Temperatures



Groundwater Modeling



Manifold Learning

The HD Challenge

- One common characteristic of these problems is they involve functions with many variables or parameters
- Mathematically this means we are faced with numerically approximating a high dimensional function
  - $F : [0,1]^D \to X$
  - X a Banach space (often just $I\!R$ or $I\!R^m$)
  - D large and possibly infinite
  - Typical Computational Tasks
    - Create an approximation $\hat{F}$ to F
    - Evaluate some quantity of interest: $Q(F)$
    - Q is some linear or nonlinear functional:
      - $Q(F)$ is a high dimensional integral of F
      - $Q(F)$ is the max or min of F

Approximation Theory

- The last 50 years have been Golden Years in AT
- We briefly describe the AT setting
  - Prescribe a way to measure error: a norm $\|\cdot\|_X$
  - Specify the type of approximation, i.e., the sets of functions X_n, $n \geq 0$, which will be used to approximate
- There are typically two types of approximation
  - Linear Approximation: X_n is a linear space of dimension n in X
  - Non- Linear Approximation: X_n is a nonlinear set depending on n parameters (n degrees of freedom)
- Given F, we have the error of approximation

$$E_n(F)_X := E(F, X_n)_X := \inf_{g \in X_n} \|F - g\|_X$$

The Performance of $(X_n)_{n\geq 0}$

- There are several ways to evaluate the performance of (X_n) and compare different methods
  - Checking performance on one function F makes no sense
  - For any compact set $K \subset X$ we define

$$E_n(K)_X := E(K, X_n)_X := \sup_{F\in K} E(F, X_n)_X, \quad n \geq 0$$

  - Approximation Class: For each $r > 0$ define $\mathcal{A}^r((X_n)_{n\geq 0}, X)$ as the set of all $F \in X$ such that

$$\|f\|_{\mathcal{A}^r} := \sup_{n\geq 0} E_n(F) \ < \ \infty$$

Linear Methods of Approximation

- Simplest Example: $X = C[0,1]$
  - X_n algebraic polynomials of degree $n-1$, i.e., $P = \sum_{k=0}^{n-1} c_k x^k$
  - X_n p.w. polynomials of fixed degree k on equidistant partition of $[0,1]$
  - $X_n = \mathrm{span}(\phi_1, \ldots, \phi_n)$ with $\phi_1, \ldots, \phi_n \in X$ fixed and linearly independent
  - Splines, Fourier, Wavelets

Non- Linear Methods

- Simplest Example: $X = C[0,1]$
- Σ_n nonlinear set
  - Piecewise Polynomial Approximation of Degree k: $g \in_n$ is a p.p. on a partition with n cells
    - the partition can be chosen depending on F
  - n term approximation from a dictionary $\mathcal{D} = \{\psi_1, \dots, \psi_N\}$
    - $\mathcal{D}$ usually has structure: frame or basis
    - $\Sigma_n := \{g = \sum_{k\in\Lambda} c_k\psi_k : \ \#\Lambda = n\}$
  - Manifold Approximation:
    - Two mappings: $a : \ X \to I\!R^n$ and $M : \ I\!R^n \to X$
    - $\Sigma_n := \{M(z) : \ z \in I\!R^n\}$
    - The points $M(z)$ live on a manifold

Typical Approximation Questions

- How fast does $E_n(F)$ tend to zero?
  - This requires some information about F
  - F is in some model class K
  - K is a compact set in X which quantifies what we know about F from the application
  - For example a regularity theorem in PDEs
- Have we chosen the best method of approximation?
  - Best over all linear methods ?
  - Best over all nonlinear methods?
  - This is answered by concepts like widths and entropy
- Can we realize the approximation numerically?
  - This requires information about F through data or queries

Model Classes

- Classical model classes K based on smoothness
  - F has smoothness (of order s)
  - F is in C^s, Sobolev space $W^s(L_p)$, Besov space
- AT says n computations can only capture F to accuracy $C(D,s)n^{-s/D}$ where D is the number of variables
- If D is large than s must also be very large for any reasonable accuracy: Curse of Dimensionality
- But we have no control over s which is inherent in the real world problem
- So conventional assumptions on F and conventional numerical methods will not work
- Also beware that $C(D,s)$ grows exponentially with D

Example (Novak-Wozniakowski)

- To drive home the debilitating effect of high dimensions consider the following example
  $\Omega := [0,1]^D, \quad X = \mathbb{R}, \quad \mathcal{K} := \{F : \|D^\nu F\|_{L_\infty} \le 1, \forall \nu\}$
- Any algorithm which computes for each $F \in \mathcal{K}$ an approximation $\hat{F}$ to accuracy $1/2$ in L_∞ will need at least $2^{D/2}$ FLOPS
- So if $D = 100$, we would need at least $2^{50} \asymp 10^{15}$ computations to achieve even the coarsest resolution
- This is The Curse of Dimensionality
- This phenomenon cannot be defeated by some clever approximation scheme: it says every approximation scheme will suffer this effect

The Remedy

- Conventional thought is that most real world HD functions do not suffer the curse
- Need new Model Classes in HD
  - Compressibility : F is well approximated by a sum of a small number of functions from a fixed basis/frame/dictionary
  - Anisotropy/Variable Reduction: not all variables are equally important - get rid of the weak ones
  - Tensor structures: variables are intertwined
  - Superposition: F is a composition of functions of few variables - Hilbert's 13-th problem
  - Many new approaches based on these ideas: Manifold Learning; Laplacians on Graphs; Sparse Grids; Sensitivity Analysis; ANOVA Decompositions; Tensor Formats; Discrepancy; Deep Learning

New World for Approximation

- The challenge to AT is to understand whether these new model classes actually break the curse
- We need certifiable theorems given the proposed model class and to characterize the methods of approximation that achieve optimal performance
- Let $(\Sigma_n)_{n\geq 1}$ be the family of spaces to be used for approximation (linear or nonlinear)
- The performance of this family on K is given by

$$E_n(K)_X := E(K,\Sigma_n)_X := \sup_{F\in K} \operatorname{dist}(F,\Sigma_n)_X$$

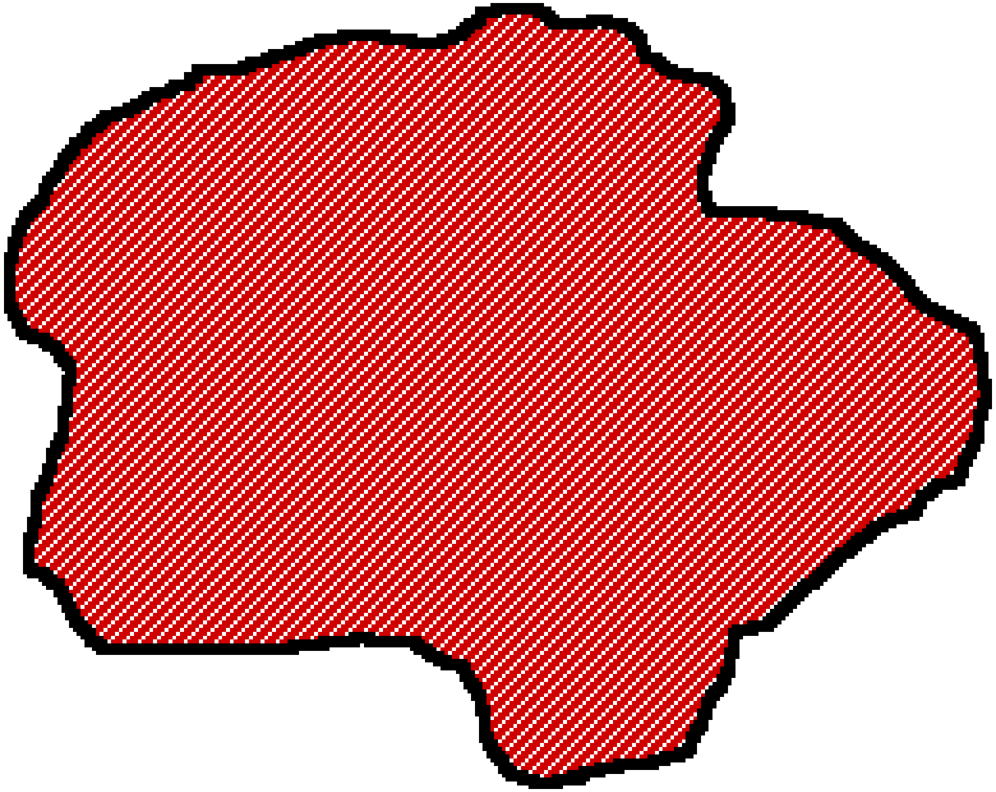
- To determine optimal performance on K we need to determine its widths and entropy

Entropy of a compact set

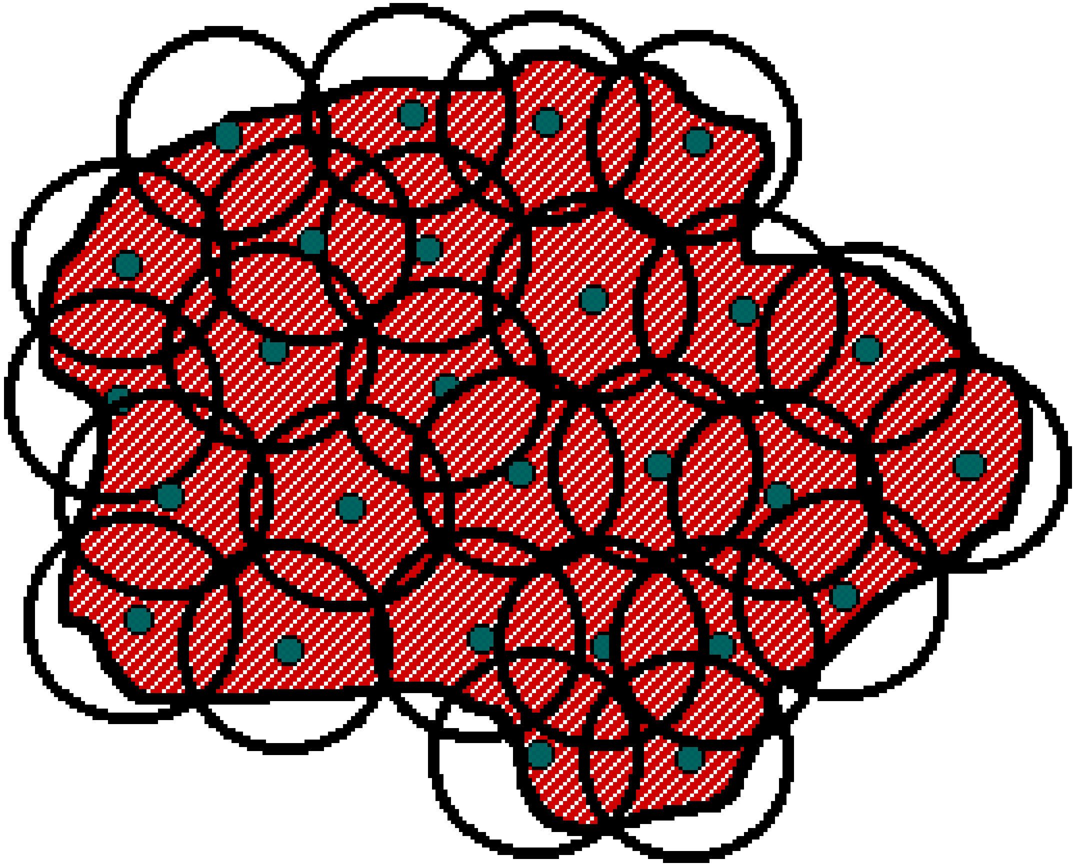
- There is a general criteria to see whether a model class $K \subset X$ is HD friendly for approximation/computation
- It is given by the Kolmogorov metric entropy of K
  - Given $\epsilon > 0$: $N_\epsilon(\mathcal{K})_Y$ denotes the smallest number of balls of radius ϵ in X we need to cover $\mathcal{K}$?
  - $H_\epsilon(K)_Y := \log_2 N_\epsilon(K)_Y$ Kolmogorov entropy
  - Heuristically any approximation will need at least $H_\epsilon(\mathcal{K})_Y$ computations to approximate all of K to accuracy ϵ
  - So if the entropy of $\mathcal{K}$ is not reasonable this is not a useful model class
  - Entropy numbers

$$\epsilon_n(K)_X := \inf\{\epsilon : \ H_\epsilon(K)_X \le n\}, \quad n \ge 0$$

Covering



Covering



Kolmogorov Widths

- Once we have chosen a method of approximation there is an optimal way to measure performance through widths of the model class K
- Kolmogorov n widths for linear approximation:

$$d_n(K)_X := \inf_{\dim(Y)=n} \operatorname{dist}(K,Y)_X, \quad n \geq 0$$

  - No linear method of approximation using n degrees of freedom can perform better than $d_n(K)_X$ in approximating the elements of K

Non-linear Widths

- Most nonlinear methods of approximation can be viewed as form of manifold approximation
- There are two continuous mappings $a_n : X \to \mathbb{R}^n$ and $M_n : \mathbb{R}^n \to X$ and the approximation to F is $A_n(F) = M_n(a_n(F))$
- Manifold width(DeVore-Howard-Micchelli) :

$$\delta_n(K)_X := \inf_{a_n, M_n} \sup_{F \in K} \| \|F - M_n(a_n(F))\|_X$$

- Stable widths $\delta_n^*(K)_X$ (Cohen-D-Petrova-Wojtaszczyk)
  - Here we add the requirement that the mappings a and M are Lipschitz mappings

Checking

- Suppose you think you have the correct model class K for your HD application
  - Check whether K breaks the curse by determining / estimating its entropy or widths
- Suppose you think you have the mother of all approximation schemes for your application
  - Find the model classes for which the approximation scheme performs: with rate $O(n^{-r})$
- In numerical scenarios (such as data fitting) you still need to understand how the information (or lack of information) effects optimal performance
- You still need to build a numerical algorithm utilizing the information you have about F

Numerical Algorithms

- Let us turn now to constructing numerical algorithms in HD -such algorithms depend on the information we are have about F
- Setting I: Query Algorithms: We can ask questions about F in the form of Queries
  - A query is the application of a linear functional to F
    - Examples: Point evaluation or weighted integrals
  - Given that $F \in \mathcal{K}$ and a query budget n - where should we query to best reconstruct F?
- Setting II: Data Assimilation: We cannot ask questions but rather are given data in the form of some information about F?
  - Given that $F \in \mathcal{K}$ and given the data how can we best reconstruct F?

Query Algorithms

- A query algorithm prescribes where to sample F given knowledge that F is in a certain Model Class K.
  - Sampling: Extract information $\ell_1(F), \dots, \ell_n(F)$
  - Reconstruction: From the drawn information construct an approximation $A_n(F) \in Y$ to F
- The minimal distortion of a query algorithm is

$$\delta_{A_n}(K) := \inf_{A_n} \sup_{F \in \mathcal{K}} \|F - A_n(F)\|_Y$$

- Optimal performance is given by the Gelfand width

$$d^n(K)_Y := \inf_{\mathrm{codim}(V)=n} \sup_{f \in \mathcal{K} \cap V} \|f\|_Y$$

- However, often we may want to limit the types of queries
  - Standard Information: Query asks for the value of F at a point: $\mathcal{Q}_n$ is a cloud of points in HD

Strategies for $\mathcal{Q}_n$

- The best choice for $\mathcal{Q}_n$ depends on the model class K
- However choices for $\mathcal{Q}_n$ generally take two forms
  - Random Queries:
    - Monte Carlo: sampling for HD integration
    - Compressed Sensing: for recovery of sparse signals
    - Albert Cohen Theory: carefully choose the probability measure for randomness
  - Deterministic Querying:
    - Hashing
    - Discrepancy theory(Quasi Monte Carlo) based on number theory-Chinese Remainder Theorem
    - Commutative Algebra (Cohen-Macauley theory): Use finite dimensional fields

Data Assimilation

- Often we do not have the luxury to query but rather are given information about F in the form of data
  - Form of the Data?: We assume $w_j = l_j(F), \quad j = 1, \dots, m$, where l_j are linear functionals
  - Measurement map $M(F) = w := (w_1, \dots, w_m)$
- An algorithm is a mapping $A : \mathbb{R}^m \mapsto X$ where $A(M(f))$ is an approximation to $f \in X$ giving error

$$E(F, A)_X := E(F, M, A)_X := \|F - A(M(F))\|_X$$

- Optimal Recovery: Find the best algorithm A given M and the model class K: Micchelli and Rivlin in the 1970s

Optimal Recovery Performance

- We must pay a price for the lack of full information about F when only given data
- Let
$$E(K,M) := \inf_A \sup_{F\in K} E(F,M,A)$$
be the optimal error in recovery of K from the given measurement map M
  - We can always write $E(K,M) = \mu(K,M)d^m(K)_X$ where d^m is the Gelfand width
  - $\mu \geq 1$ is the price we pay for not having the optimal m measurements for K
  - One can often determine μ from the null space $\mathcal{N} := \{F \in K : \ \ell_j(F) = 0\}$

Examples

- The remainder of this talk will discuss a few prominent examples of HD Model Classes and HD approximation
- I have to be very selective because of time

Non-Democracy of Variables

- Simplest Example: $F \in C[0,1]^D$ depends on D variables but only d are active- the d active variables are unknown to us and may vary with F
- K is the set of all such F with $\|D^\nu F\|_{L_\infty} \le 1$, $|\nu| \le k$
  - $F(x_1, \dots, x_D) = g(x_{j_1}, \dots, x_{j_d})$, where $g \in C^k$
- This problem and many generalizations were studied by DeVore-Petrova-Wojtaszczyk
- Σ_n consists of piecewise polynomials of total degree $k-1$ on a partition of $[0,1]^D$ into n cells
- The polynomial pieces have only d active variables and the partitions depend on F

Optimal Algorithmss

- The point clouds in Query Algorithms have two tasks:
  - Determine change coordinates $j_1, \dots, j_d$
  - Give a uniform grid with spacing $h \asymp n^{-1/d}$ for all d dimensional space spanned by a possible $j_1, \dots, j_d$
- Such point clouds are constructed using Hashing
  - A Hashing query touches every coordinate
  - It identifies the change coordinate and creates the piecewise polynomial approximation after gathering all the information
- DPW Theorem: Error of algorithm on K for n queries is

$$\delta_{A_n}(K) \leq C\delta^m(K) \leq Cn^{-k/d} \log D$$

Anisotropic analyticity

- I choose this next example for several reasons
  - $D = \infty$ and F is Banach space valued
  - Application to parametric PDEs
  - We know model classes via regularity theorems
- $F : U \to X$ is a Banach space valued function depending on $d = \infty$ variables
  - U the unit ball in $\ell_\infty(\mathbb{N})$
  - The elements $z \in \ell_\infty(\mathbb{N})$ are bounded sequences $(z_1, z_2, \dots)$ of complex numbers
- Let $\rho := (\rho_1, \rho_2, \dots)$ be an increasing sequence or real numbers with $\rho_1 > 1$ and define the polydisc D_ρ of z satisfying $|z_j| \le \rho_j$
- H_ρ the space of F analytic on D_ρ and contnuous on $\bar{D}_\rho$

Approximation by Polynomials

- We want to approximate F in the norm

$$\|\cdot\| := \|\cdot\|_{L_\infty(U,X)}$$

- We approximate F by X valued polynomials
  - Let $\mathcal{F} := \{\nu = (\nu_1, \nu_2, \ldots\}$ where the entries in ν are nonnegative integers and only a finite number of the ν_j are nonzero
  - Given a finite set $\Lambda \subset \mathcal{F}$, then $\mathcal{P}_\Lambda := \{P : P = \sum_{\nu\in\Lambda} c_\nu z^\nu$
  - The possible sets Λ can be quite complicated and so we restrict ourselves to lower sets which mean that $\nu \in \mathcal{F}$ and $\mu \leq \nu$ implies $\mu \in \mathcal{F}$
  - $E(F, \mathcal{P}_\Lambda) := \inf_{P\in\mathcal{P}_\Lambda} \|F - P\|$

Model Classes

- Each $F \in H_\rho$ has a Taylor expansion

$$F = \sum_{\nu \in \mathcal{F}} t_\nu z^\nu, \quad z \in U$$

where the Taylor coefficients t_ν are in X and satisfy $\|t_\nu\|_X \le \|F\|_{L_\infty} \rho^{-\nu}$

- Model Classes (Bachmyar-Cohen-Migliorati):
- For ρ and $0 < p \le \infty$ we say $F \in B_{\rho,p}$ if
  - F has Taylor coefficients $t_\nu,\ \nu \in \mathcal{F}$
  - $F = \sum_{\nu \in \mathcal{F}} t_\nu z^\nu$ unconditionally on U
  - $\|F\|_{B_{\rho,p}} := (\rho^\nu \|t_\nu\|_X)_{\nu \in \mathcal{F}} < \infty$.
- These classes are anisotropic

- Approximation Theorem for $B_{\rho,p}$, $1 \le p \le \infty$:
  - Rearrange the sequence $(\rho^{-\nu})_{\nu\in\mathcal{F}}$ into decreasing order: $\delta_n$is the n-th largest term
  - Let Λ_n is the lower set of size corresponding to the n largest of the $\rho^{-\nu}$
  - If q is the conjugate index to p: $1/p + 1/q = 1$

$$\|F - \sum_{\nu\in\Lambda_n} t_\nu z^\nu\| \le (\sum_{k>n} \delta_k^q)^{1/q}, \quad n \ge 0$$

  - This estimate is in a certain sense optimal
  - δ_n and Λ_n found by sorting
  - The asymptotic behavior of (δ_n) can be found by counting lattice points inside simplices determined by ρ

Other Settings

- No time to discuss in detail other important settings:
- Sparsity Model Classes
  - Best queries are random Kashin-Gluskin
  - Recovery from queries: Donoho-Candes (see Cohen-Dahmen-DeVore)
- Tensor Structures
- Not enough good Approximation Theory
- Rank one tensors -Bachmyar-Dahmen–DeVore-Grasedyk
- -best queries given by discrepancy theory
- Wolfgang Dahmen: “I can do anything but not everything ”

Deep Neural Networks

- One of the highest profile HD approximation methods is given by deep Neural Networks (talk of Gitta Kutyniok)
- There is still not satisfactory theory to explain its success
- However we are gaining new insights and I want to give my take on this subject
- Surprisingly, I will speak about using deep Neural Networks to approximate univariate functions
- My justification is that even the univariate case is not well enough understood and HD will be even more complex
- I am sure Gitta will be more HD

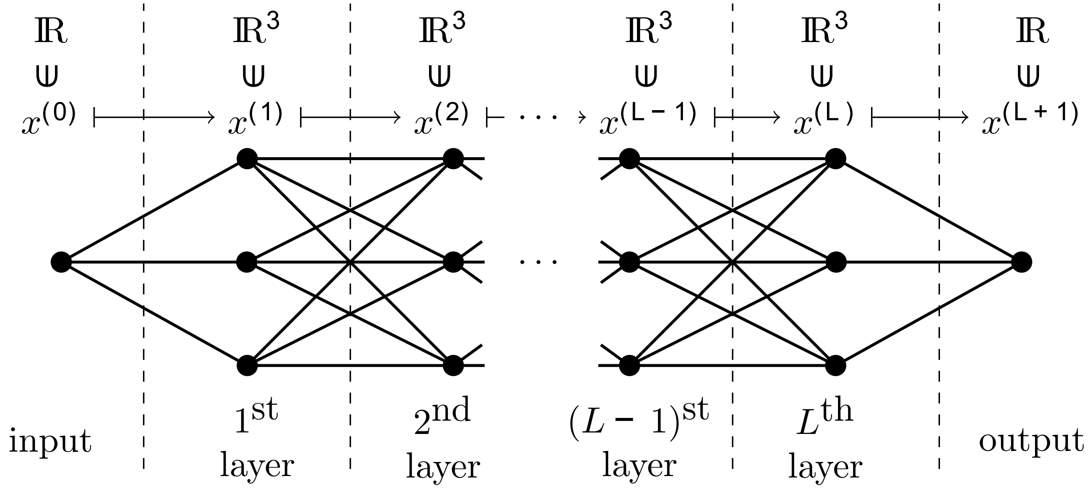
Deep ReLU Networks

- I limit my discussion to the ReLU Networks since these are the most prominent
- $ReLU(y) := \max(y, 0) = y_+$
- Here is a graphic of a NN of width $W = 3$ and depth L
- Each node is called a neuron
  - Each neuron at a layer ℓ in row i has an associated function $\eta_{i,\ell}$
  - This function takes the form

$$\eta_{i.\ell} = ReLU(\sum_{j=1}^{W} a_{i,j}\eta_{j,\ell-1} + b_i)$$

where the the sum is taken over all neurons of the previous layer that feed to $\eta_{i,\ell}$

NN graphic with Width =3



The layers

- The first layer consists of the functions $(a_i x + b_i)_+$
- Subsequent layers are described by a matrix of size $W \times W$ and a vector $b \in \mathbb{R}^W$
- Output layer just a linear combinations of the functions in layer L
- So the number of parameters used to describe the NN is $n(W, L) = 2W + (L-1)(W^2 + W) + W \approx LW^2$
- Sometimes one imposes conditions on the matrices that greatly reduce the number of parameters
  - sparse matrices or convolution structure
- Υ_W^L is the set of functions (outputs) of such networks of width W and depth L. This is our approximation family

Deep Networks

- In Deep Networks we fix W and let L get large
- We want to understand the advantages of depth over shallow networks and other methods of approximation
- The functions in Υ^L_W are Continuous pw Linear (CPwL)
- So the closest classical approximation family to deep networks are the spaces Σ_n, $n \geq 1$ where Σ_n consists of all CPwL functions with n arbitrary break points
- Notice that both Υ^L_W and Σ_n are nonlinear spaces: when adding functions in these spaces the result is not generally in the space
- Also both spaces are examples of manifold approximation

Comparing Σ_n and Υ_W^L

- To make a fair comparison between these two families of spaces we fix W and define $\Upsilon_n := \Upsilon_W^{L_n}$ where L_n is chosen so that Υ_n is determined by $\approx n$ parameters
- Two ways to compare
  - How do these two spaces of functions compare (Expressive power)?
  - How well do they approximate?
    - Approximation Classes: Given $r > 0$ the class $\mathcal{A}^r((\Upsilon_n), X)$ consists of all $F \in X$ such that

$$\operatorname{dist}(F, \Upsilon_n)_X \leq M n^{-r}, \quad n \geq 0$$

    - Smallest M is $\|F\|_{\mathcal{A}^r}$
- The following results come mainly from Daubechies-DeVore-Foucart-Hanin-Petrova

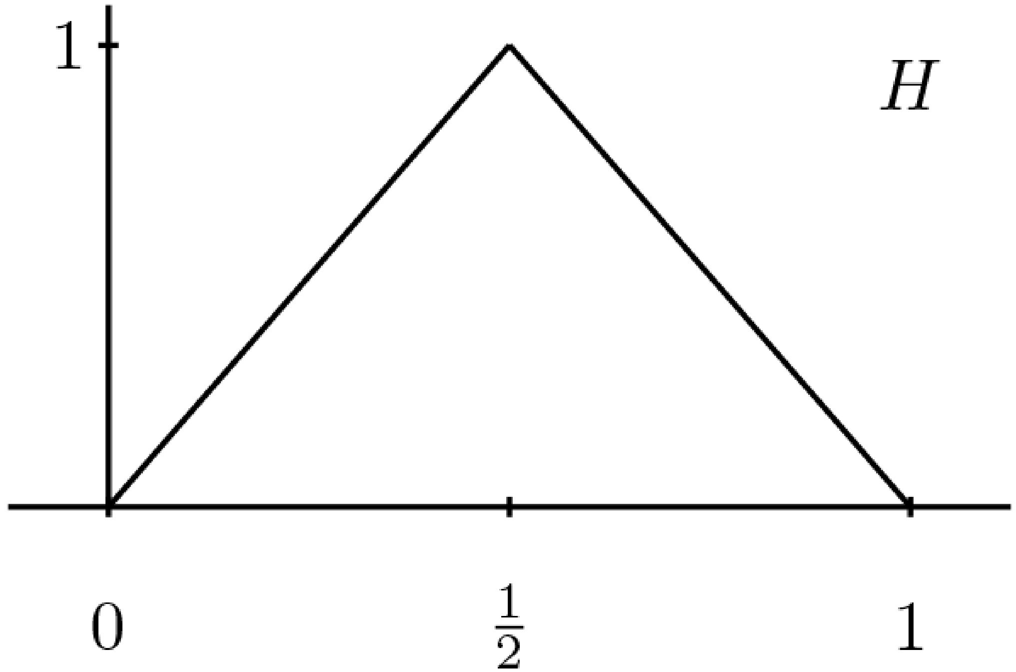
First Question

- Theorem: Σ_n contained in Υ_{Cn} for $n \geq 1$ with an absolute constant, e.g. $C = 12$
  - So Υ_n is at least as expressive as Σ_n
- There are many examples of functions S that are in Υ_n but far from being in Σ_n
- They are obtained by exploiting the most important property that Υ_n has that Σ_n does not
  - Given functions F, G, we let $F \circ G := F(G)$ be the composition of these two functions
  - $F^{\circ n}$ denotes the n fold composition of F with itself
- If $S \in \Upsilon_n$ and $T \in \Upsilon_m$ then $S \circ T$ is in Υ_{n+m}
- On the other hand, If $S \in \Sigma_n$ and $T \in \Sigma_m$ then the best we can say is $S \circ T$ is in Σ_{nm}

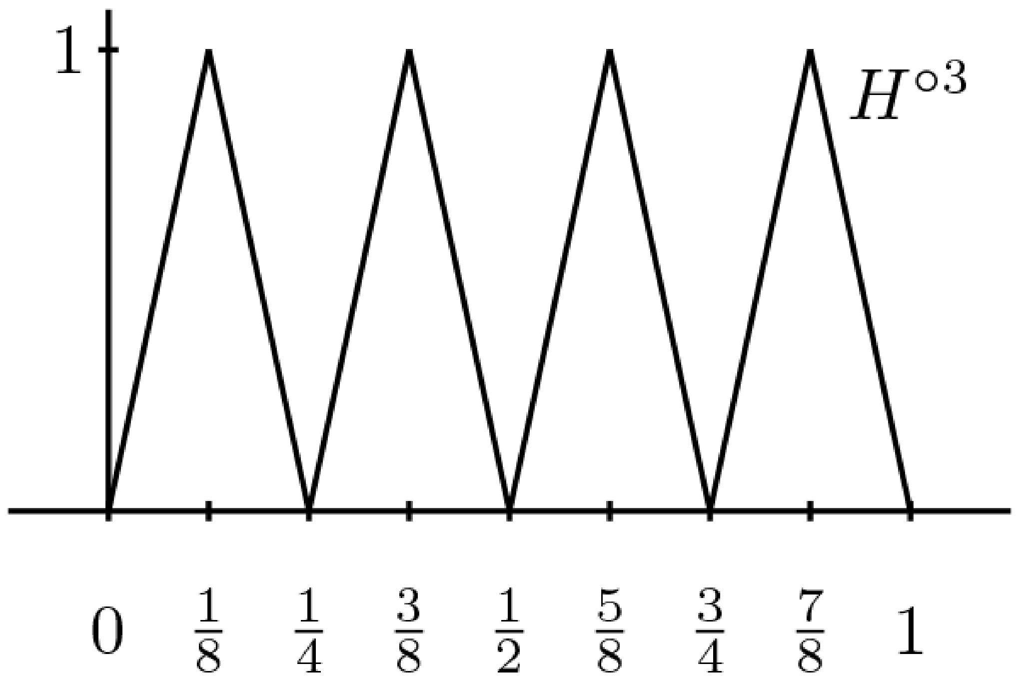
Examples

- Simplest Example the hat function
  - n fold composition $H^{\circ n}$ is a saw tooth with 2^n hats
- Piecewise self similar functions
  - If S is in Υ_k with $S(0) = S(1) = 0$
  - $I_1, \ldots, I_m$ is a partitioning of $[0, 1]$ into m intervals
  - any function which is a scaled version of S on each of these intervals is in Υ_{k+6m}
    - We call S a pattern
    - So we can replicate patterns cheaply
    - Such a function is in Σ_{km}
  - More generally we can create bases and redundant frames of CPwL

Hat Function



Composition



The Approximation Classes

- I take $X = C[0,1]$ and $\|\cdot\| := \|\cdot\|_X$
- General Principal
  - Let $\phi_k \in \Upsilon_k$ with $\|\phi_k\| = 1$
  - $(\alpha_k) \in \ell_1$
  - $\sum_{k \geq n} |\alpha_k| \leq M n^{-2r}$
  - Then $F := \sum_{k=1}^{\infty} \alpha_k \phi_k$ is in $\mathcal{A}^r((\Upsilon_n), X)$
  - Same property holds with (Υ_n) replaced by (Σ_n)
- The General Principle can be used to construct may interesting F in $\mathcal{A}^r((\Upsilon_n), X)$
  - The Tagoki Function: $F_T := \sum_{k=1}^{\infty} 2^{-k} H^{\circ k}$
    - This functiont is nowhere differentiable
    - It can be approximated to exponential accuracy: It is in all $\mathcal{A}^r((\Upsilon_n), X)$, $r > 0$

Many other examples

- Dynamical systems, iterated function systems, fractals, refinement equations give functions that can be approximated with exponential accuracy but the functions are not smooth
- On the other side of the spectrum
  - All analytic functions can be approximated with exponential accuracy
  - This uses the fact that all power function x^k, $k = 1, 2, \ldots$ can be approximated to exponential accuracy

Other surprizes

- Yarotsky: Any Lip 1 function can be approximated to accuracy $O((n \log n)^{-1})$
- The appearance of the log is a surprise
- This result generalizes to many other classical function spaces
- What is going on?
  - The manifold width of Lip 1 is $\geq Cn^{-1}$
  - Also the entropy numbers of the class Lip 1 are $\geq C/n$ with an absolute C
  - This means the mapping of F to its approximant cannot be continuous
  - This cautions us to be careful about the Stability of Algorithms