The mathematical foundations of artificial intelligence

Balázs Szegedy

2018. október 30.

Some details of the program

1. The program started in September 2018
2. It runs in the frame of the National Excellence Program (NKP) (NK-FIH)
3. It involves 5 institutes: RENYI, SZTAKI, PPKE, SZTE, ELTE
4. The budget is roughly 1 billion Ft = 3.5 M dollars for 3 years

The major goals of the program

1. Mobilize the community of mathematicians to do research in AI (Hungary is traditionally good at mathematics)
2. Help in the education (university courses, PhD programs, popularizing lectures, etc...)
3. Invigorate collaboration between institutions
4. Building bridges between theory and practice in AI
5. Demonstrate the usefulness of AI to the society (Pilot program: a useful medical application)

3 levels of abstraction

1. **Applied:** **use existing models** For example: Train neural networks for concrete problems (classification of chronic wounds, car driving, etc...)
2. **Semi-theoretical:** **Improve, develope models** For example: "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift"
3. **Theoretical:** **prove theorems, understand matematical phenomena** For example: "Testing the manifold hypothesis" (Journal of the American Mathematical Society)

3 levels of abstraction

1. **Applied:** **use existing models** For example: Train neural networks for concrete problems (classification of chronic wounds, car driving, etc...)
2. **Semi-theoretical:** **Improve, develope models** For example: "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift"
3. **Theoretical:** **prove theorems, understand matematical phenomena** For example: "Testing the manifold hypothesis" (Journal of the American Mathematical Society)

We think that interaction between these three directions is essential in the development of AI. Our goal is to create a research environment in which theoretical directions can directly communicate with the more applied directions.

A few words about education

AI is related to almost everything we do.

A few words about education

AI is related to almost everything we do. AI related courses should be available at all kinds of universities and faculties.

A few words about education

AI is related to almost everything we do. AI related courses should be available at all kinds of universities and faculties. Traditionally we associate AI with computer science, information technology, programming or mathematics. In reality it is much broader.

A few words about education

AI is related to almost everything we do. AI related courses should be available at all kinds of universities and faculties. Traditionally we associate AI with computer science, information technology, programming or mathematics. In reality it is much broader. An example: In the frame of our program we started an experimental course at the medical university (SOTE) entitled "Mesterséges intelligencia szerepe az orvostudományban".

Important challenges in AI

AI is very different from classical computer science. It does not have a well founded mathematical theory. The most important concept is **generalization** which makes AI work but it is not very well understood.There is a set if clever engineering tricks that make it work. A real mathematical understanding of deep learning could lead to even better models and practical results. Theoretical mathematicians should work closely with more applied AI specialists.

There is also the problem of **interpretation:** Even if a neural network works quite well, in many cases we need to understand why it does something. For example: a self driving care makes an accident.

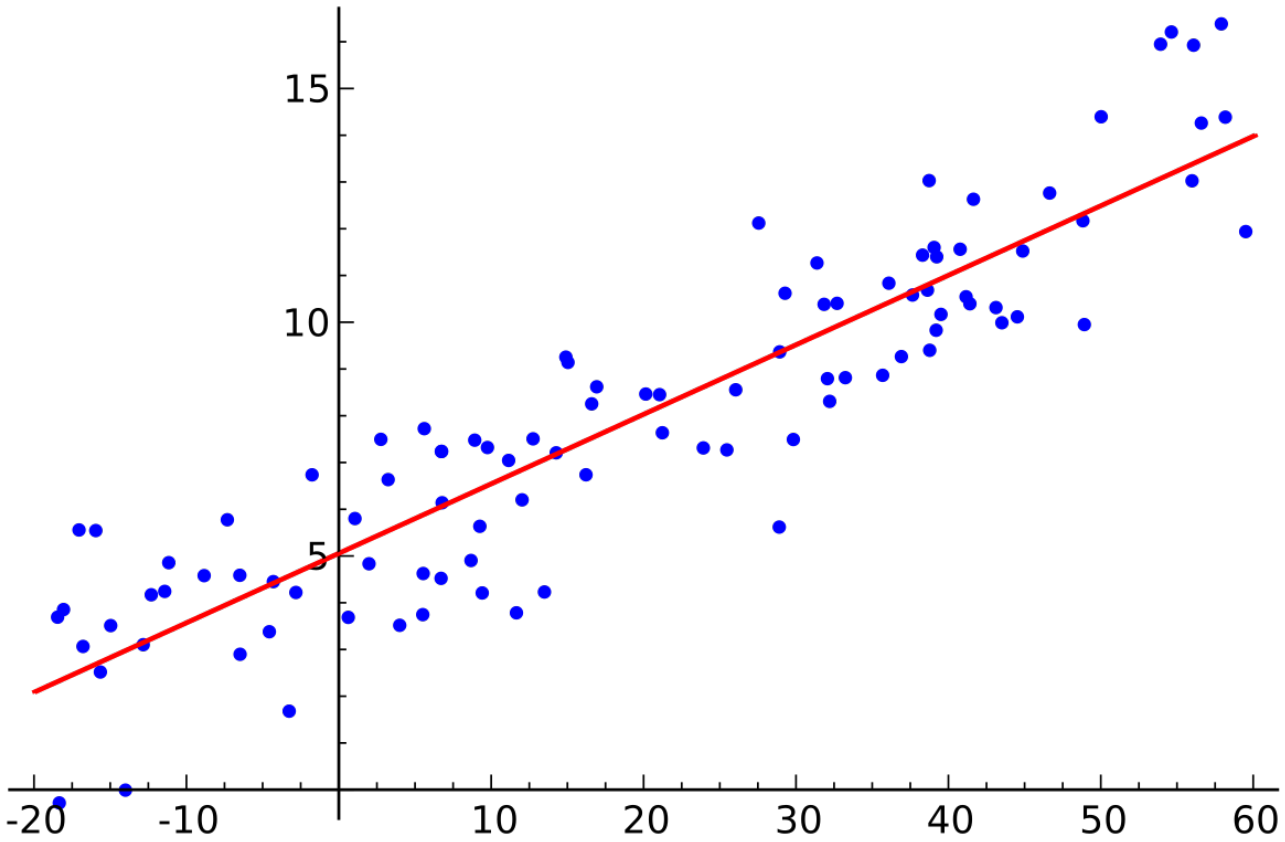
Formal reasonong: AI is not very good at this. AI should be at least as good as humans but we seem to be far from this. (AlphaGo is encouraging! it simulates intuition and it exceeds humans)

Mathematical projects

1. Find the mathematics of dimension reduction
2. Find appropriate mathematical hypothesis for the structure of real life data sets
3. Complexity notions for machine learning (how to measure the complexity of a data set?)
4. Find the mathematics of generalization
5. Mathematical analysis of various models of machine learning
6. Interesting connections with quantum mechanics, quantum computing and statistical physics

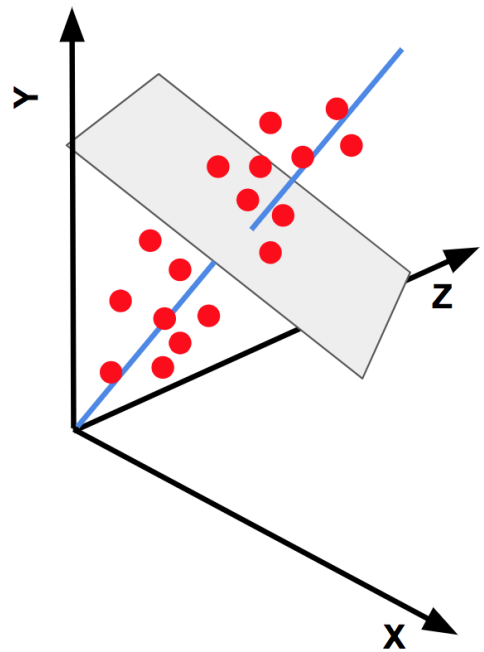
Sparsity and dimension reduction

Sparsity $\sim$ differentiating, dimension reduction $\sim$ simplifying, compressing. Both are happening in a well trained neural net but they have different roles. We focus on dimension reduction. This goes back to basic methods in statistics: linear regression, PCA.



Principal Component Analysis (PCA)

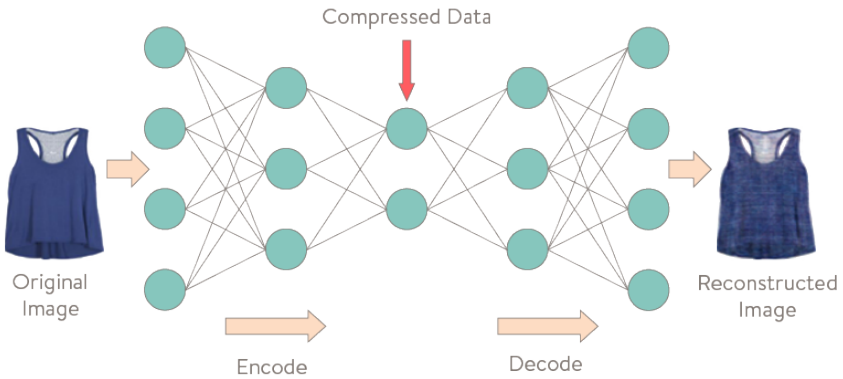
Approximating a 3 dimensional point set with 1 dimensional line.



PCA is used in machine learning and it works surprisingly well for certain problems.

Autoencoders and PCA

Auto encoders are special neural networks with a bottle neck to force the network to compress data.



If the activation function is linear then autoencoder is basically equivalent with PCA. This way autoencoders generalize PCA.

Dimension reduction

Deep learning is a very far reaching generalization of the basic methods of statistics

important facts: deep learning is very non-linear! A mathematical aspect of deep learning: *We approximate high dimensional complicated data with lower dimensional, simpler data.* (A map is constructed from the high dimensional data to the low dimensional data.)

Is there a mathematical theory for such approximations?

Answer: Yes and no. There are many theories and results. Fourier analysis is used in .jpg format. The recently emerging theory of limits of structures is fundamentally based on dimension reduction.

Tensor networks

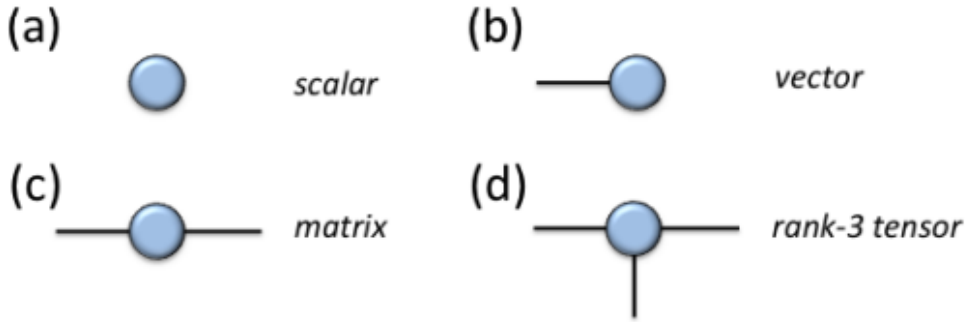
Advertisement: Linear algebra is one of the most useful subject in mathematics. Tensor networks give rise to a far reaching and beautiful extension of linear algebra and especially it generalizes matrix multiplication in a useful way. They provide a very efficient language in quantum mechanics to describe quantum states.

What is a tensor? A tensor of rank n in dimension d is described by an array of numbers $M_{i_1,i_2,\ldots,i_r}$ such that $1 \leq i_1, i_2, \ldots, i_r \leq d$. A rank 0 tensor is a scalar, a rank 1 tensor is a vector, a rank 2 tensor is a matrix, a 3 tensor is a $d \times d \times d$ array etc...

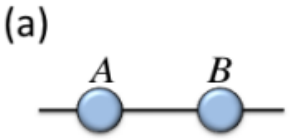
Example: A probability deistribution of $0-1$ sequences of length 3 can be looked at as a tensor of rank 3 in dimension 2. Indeed: the distribution is described by numbers $M_{i,j,k}$ where $i, j, k \in \{1, 2\}$ and $M_{i,j,k}$ is the probability of the sequence $(i-1, j-1, k-1)$.

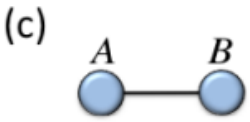
Tensor networks and picture calculus

Graphically we represent a k-tensor by a point with k open edges attached to it. We call them k-stars.



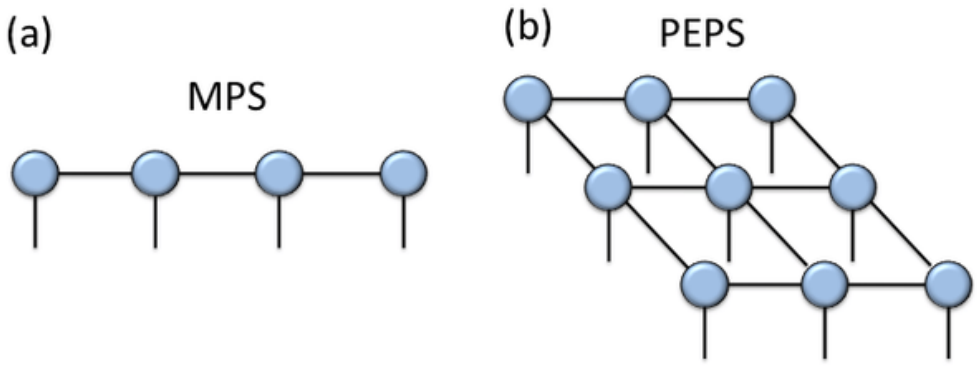
Vector-matrix operations can be generalized to higher tensors. These general operations can be represented by pictures in which we glue k-stars together along open edges. Pictures for matrix product and scalar product:





Tensor networks and picture calculus

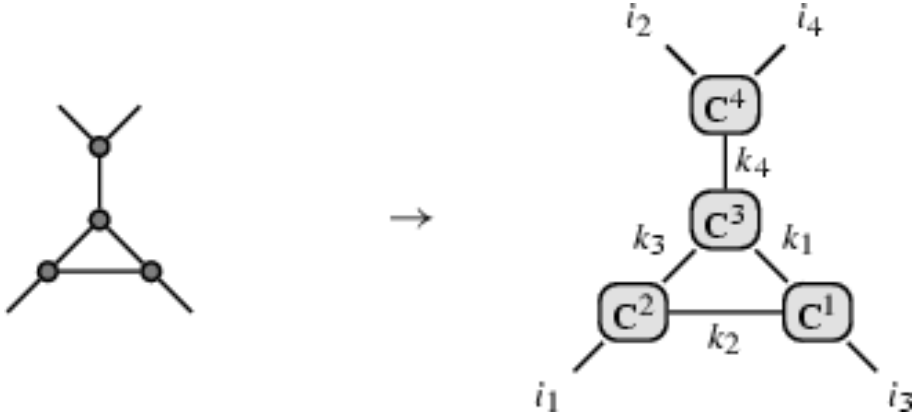
These pictures represent more complicated operations with higher tensors:



The output of the operation is a tensor whose rank is equal to the number of remaining open ends on the picture.

The formula for tensor operations

Sketch: We put labels from 1 to d in all possible ways on the edges. For each such labeling we take the product of the corresponding tensor values. We take the sum of these products over all possible labelings of the closed edges while the labels on the open edges are fixed.



$$M_{i_1,i_2,i_3,i_4} = \prod_{k_1,k_2,k_3,k_4} C^4_{i_2,i_4,k_4} C^3_{k_4,k_3,k_1} C^2_{k_3,k_2,i_1} C^1_{k_1,k_2,i_3}$$

Note: By changing C^3 on the picture we obtain a function that maps 3-tensors to 4 tensors. *In quantum mechanics, tensors describe entangled quantum states.*

Tensor networks as computers

Quantum circuits are special tensor networks composed of tensors called quantum gates. This highlights the connection between quantum computing and tensor networks.

Tensor networks as computers

Quantum circuits are special tensor networks composed of tensors called quantum gates. This highlights the connection between quantum computing and tensor networks. Quite surprisingly it turns out that tensor networks can also be used in machine learning. This is an area of active research.

Perspectives: There are many interesting open questions about tensors and tensor networks. Understanding tensor networks obtained by the iteration of a single tensor is an interesting and complicated problem. In statistical physics tensor networks are also known as edge coloring models. Partition functions of certain closed edge coloring models are characterized in

B. Szegedy: *Edge coloring models and reflection positivity*, J. Amer. Math. Soc. 20 (2007), 969-988

Results indicate that there should be some form of generalized spectral theory for k-tensors which is related to iteration. This generalized spectral theory is very likely relevant in both physics and machine learning. This is a rich subject which is part of our research program.

Fuzzy Boolean functions

A fuzzy boolean function is a function of the form $f : \{0,1\}^n \to [0,1]$. In other words the input is a string of n bits and the output is a probability. For example the bits are the pixels of a black and white picture and the output is the probability that there is a cat on the picture.

What kind of mathematical properties do fuzzy boolean functions satisfy that come from natural decision problems?

Fuzzy Boolean functions

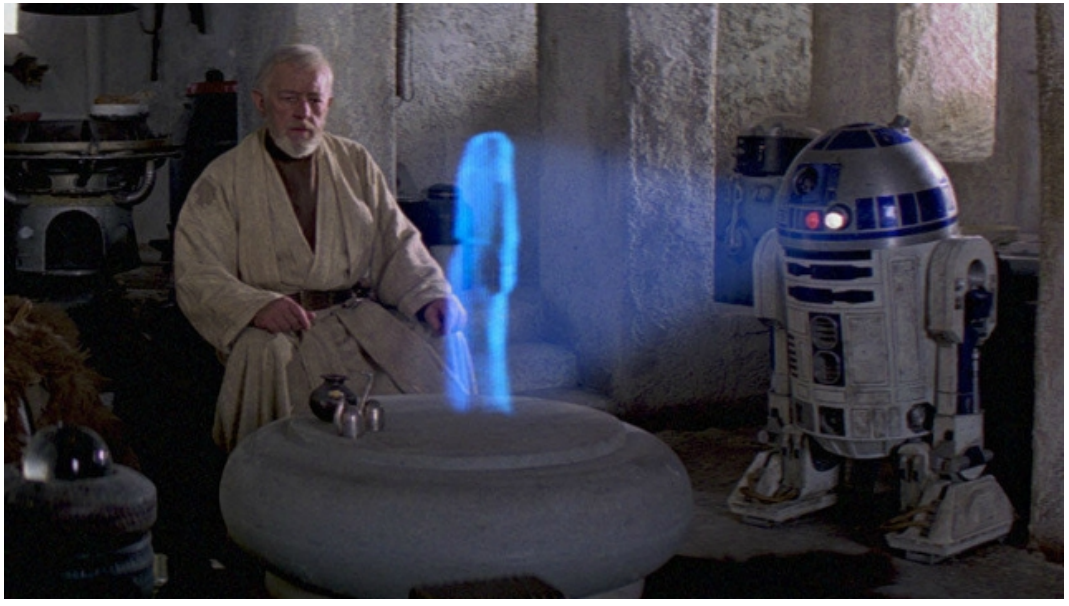
A fuzzy boolean function is a function of the form $f : \{0,1\}^n \to [0,1]$. In other words the input is a string of n bits and the output is a probability. For example the bits are the pixels of a black and white picture and the output is the probability that there is a cat on the picture.

What kind of mathematical properties do fuzzy boolean functions satisfy that come from natural decision problems?

Lipschitz property: If we change only a few pixels then the probability changes only a little.

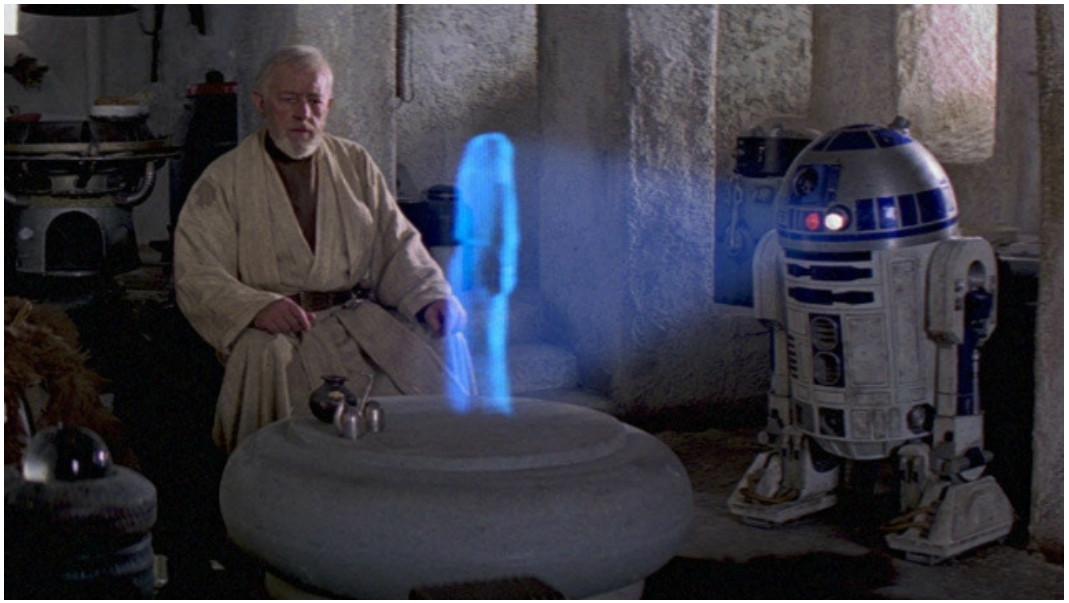
(Mote precisely $|f(v) - f(w)| \leq cd(v,w)/n$ where c is the Lipschitz constant.)

Holographic property



Unfortunatelly the Lipschitz property is too general. It seems that natural classification problems satisfy a stronger property.

Holographic property



Unfortunatelly the Lipschitz property is too general. It seems that natural classification problems satisfy a stronger property.

If we sample say 1000 random pixels we can guess with (maybe) 10% error the probability that there is a cat on the picture.

Definition: A fuzzy Boolean function is k, ϵ-holographic if for every input $v \in \{0,1\}^n$ if we choose k random bits $b_1, b_2, \ldots, b_k$ form v (together with their location) then with probability at least $1-\epsilon$ they determine $f(v)$ with an error at most ϵ.

Examples

Let us fix an error, say $\epsilon = 0.01$. The smallest k for which a fuzzy function f is k, ϵ-holographic measures the "complexity" of f relative to the error ϵ. The lower the complexity is the more holographic the function is.

Examples

Let us fix an error, say $\epsilon = 0.01$. The smallest k for which a fuzzy function f is k, ϵ-holographic measures the "complexity" of f relative to the error ϵ. The lower the complexity is the more holographic the function is.

Example 1. Average: $f(v) = \frac{1}{n} \sum_{i=1}^{n} v_i$. is low complexity. Why?

Examples

Let us fix an error, say $\epsilon = 0.01$. The smallest k for which a fuzzy function f is k, ϵ-holographic measures the "complexity" of f relative to the error ϵ. The lower the complexity is the more holographic the function is.

Example 1. Average: $f(v) = \frac{1}{n} \sum_{i=1}^{n} v_i$. is low complexity. Why? Law of large numbers

Examples

Let us fix an error, say $\epsilon = 0.01$. The smallest k for which a fuzzy function f is k, ϵ-holographic measures the "complexity" of f relative to the error ϵ. The lower the complexity is the more holographic the function is.

Example 1. Average: $f(v) = \frac{1}{n} \sum_{i=1}^{n} v_i$. is low complexity. Why? Law of large numbers

Example 2. Smooth weighted average $f(v) = \frac{1}{n} \sum_{i=1}^{n} \lambda_i v_i$. The vector λ_i is bounded in the maximum norm.

Examples

Let us fix an error, say $\epsilon = 0.01$. The smallest k for which a fuzzy function f is k, ϵ-holographic measures the "complexity" of f relative to the error ϵ. The lower the complexity is the more holographic the function is.

Example 1. Average: $f(v) = \frac{1}{n} \sum_{i=1}^{n} v_i$. is low complexity. Why? Law of large numbers

Example 2. Smooth weighted average $f(v) = \frac{1}{n} \sum_{i=1}^{n} \lambda_i v_i$. The vector λ_i is bounded in the maximum norm.

Example 3. Smooth weighted average substituted into a continuous function : a single neuron.

Examples

Let us fix an error, say $\epsilon = 0.01$. The smallest k for which a fuzzy function f is k, ϵ-holographic measures the "complexity" of f relative to the error ϵ. The lower the complexity is the more holographic the function is.

Example 1. Average: $f(v) = \frac{1}{n}\sum_{i=1}^{n} v_i$. is low complexity. Why? Law of large numbers

Example 2. Smooth weighted average $f(v) = \frac{1}{n}\sum_{i=1}^{n} \lambda_i v_i$. The vector λ_i is bounded in the maximum norm.

Example 3. Smooth weighted average substituted into a continuous function : a single neuron.

Example 4. Bounded layer smooth neural network.

Examples

Let us fix an error, say $\epsilon = 0.01$. The smallest k for which a fuzzy function f is k, ϵ-holographic measures the "complexity" of f relative to the error ϵ. The lower the complexity is the more holographic the function is.

Example 1. Average: $f(v) = \frac{1}{n}\sum_{i=1}^{n} v_i$. is low complexity. Why? Law of large numbers

Example 2. Smooth weighted average $f(v) = \frac{1}{n}\sum_{i=1}^{n} \lambda_i v_i$. The vector λ_i is bounded in the maximum norm.

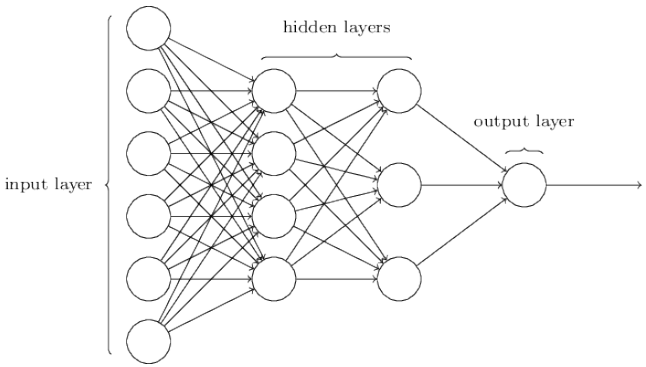
Example 3. Smooth weighted average substituted into a continuous function : a single neuron.

Example 4. Bounded layer smooth neural network.

Example 5. $f(v) = v_1$ (coordinate function) is very much not holographic. $f = \sum_{i=1}^{n} v_i \mod 2$ is even less holographic.

The main theorem

The main theorem says that example 4 (bounded smooth neural networks) covers the holographic functions.

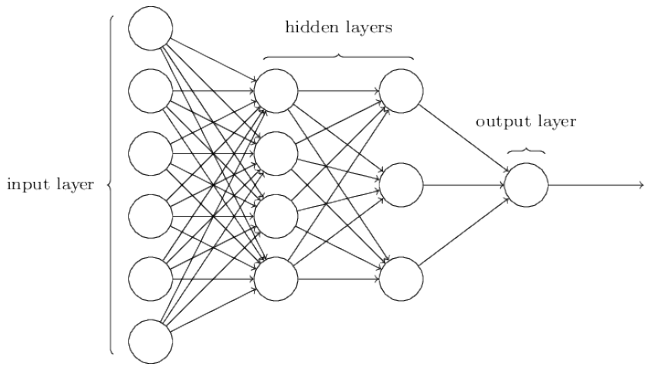


+ complexity conditions $\approx$

Low holographic complexity

The main theorem

The main theorem says that example 4 (bounded smooth neural networks) covers the holographic functions.



+ complexity conditions $\approx$

Low holographic complexity

Proof uses a version of the hypergraph regularity lemma - a fancy modern tool in combinatorics.

Problems: 1.) The smoothness restrictions on the Neural networks may not be completely natural. 2.) Quantitative problems.

Complexity notions related to machine learning?

Standard computer science uses complexity classes such as P and NP. Some experts speculate that machine learning works well because there is a low complexity phenomenon for data emerging from the physical world. **How should we measure the complexity in deep learning?**

Holographic property leads to a complexity notion for fuzzy boolean functions. There are many other promising complexity notions. For example we can say that a function f has low complexity if it can be well approximated by low degree polynomials.

Boltzmann machines (See work of Goeffrey Hinton) motivate similar complexity notions for probability distributions on $\{0,1\}^n$. This may be related to the theory in quantum mechanics which says that quantum states appearing in the physical world can be modeled by tensor networks of "reasonable" size.

How are all these complexity notions related to each other?

Other mathematical problems related the theoretical machine learning and limit theories

Let S be a data set in large dimension.

Other mathematical problems related the theoretical machine learning and limit theories

Let S be a data set in large dimension. What is the approximative geometry of S?

Other mathematical problems related the theoretical machine learning and limit theories

Let S be a data set in large dimension. What is the approximative geometry of S? Manifold learning: **Can we approximate S with a low dimensional manifold?** This question is closely related to non linear PCA.

Other mathematical problems related the theoretical machine learning and limit theories

Let S be a data set in large dimension. What is the approximative geometry of S? Manifold learning: **Can we approximate S with a low dimensional manifold?** This question is closely related to non linear PCA.

Hope: Limit theories may give new insight into the "large scale geometry" of S.

Other mathematical problems related the theoretical machine learning and limit theories

Let S be a data set in large dimension. What is the approximative geometry of S? Manifold learning: **Can we approximate S with a low dimensional manifold?** This question is closely related to non linear PCA.

Hope: Limit theories may give new insight into the "large scale geometry" of S.

Fuzzy Boolean functions may be interpreted as fuzzy data sets. Our theorem gives a new approach to study the large scale geometry. Work in progress.