TTIC 31230, Fundamentals of Deep Learning

David McAllester, April 2017

Architectures and Universality

Review:

- $\eta_t > 0$ and $\eta_t \to 0$ and $\sum_t \eta_t = \infty$ implies convergence.
- Momentum: $\hat{g}^{t+1} = \mu \hat{g}^t + (1-\mu)\, \nabla_\Theta\, \ell^t$ $\qquad$ Θ `-=` $\eta \hat{g}$.
- RMSprop: Θ_i `-=` $\frac{\eta}{(RMS_i + \epsilon)}\, (\nabla_\Theta\, \ell^t)_i$
- Adam: RMSprop+momentum: Θ_i `-=` $\frac{\eta}{(RMS_i + \epsilon)}\ \hat{g}_i$
  RMS_i smoothed

Architecture:

A Pattern of Feed-Forward Computation

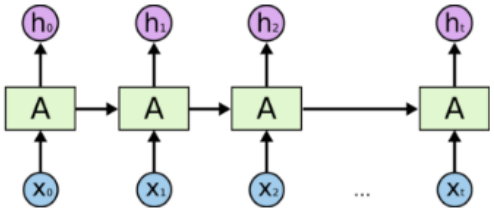
Example: **Convolution**

For filter f and signal x

$$y = Convolve(f, x)$$

$$y[i] = \sum_{j=0}^{|f|-1} x[i-j]f[j]$$

Another Example: Recurrent Neural Networks (RNNs)



$$h[t+1] = \sigma(W_R h[t] + W_I x[t])$$

Issues in Architecture

- Expressive Power
- Ease of Learning
- Embodyment of Domain Knowledge

Well Established Architectural Motifs

- Perceptron (a linear sum into an activation function)
- CNNs
- RNNs
- Pooling (max pooling or average pooling)
- Softmax
- Dropout and other Stochastic Model Perturbations
- Explicit Ensembles

Well Established Architectural Motifs

- Batch Normalization
- ReLUs
- Highway Architectures: LSTMS, Residual Networks
- Sequence to Sequence and Image to Image Architectures.
- Gating and Attention

Speculative Architectures

- Generative Adversarial Networks (GANs)
- Neural Turing Machines
- Neural Stack Machines
- Neural Logic Machines

Is There a Universal Architecture?

Noam Chomsky: By the no free lunch theorem **natural language grammar is unlearnable without an innate linguistic capacity**. In any domain a strong prior (a learning bias) is required.




Leonid Levin, Andrey Kolmogorov, Geoff Hinton and Jürgen Schmidhuber: **Universal learning algorithms exist. No domain-specific innate knowledge is required.**

Is Domain-Specific Insight Valuable?

Fred Jelinek: Every time I fire a linguist our recognition rate improves.

C++ as a Universal Architecture

Let h be any C++ procedure that can be run on a problem instance to get a loss where the loss is scaled to be in $[0, 1]$.

Let $|h|$ be the number of bits in the Zip compression of h.

Occam's Razor Theorem: With probability at least $1-\delta$ over the draw of the sample the following holds *simultaneously* for all h and all $\lambda > 1/2$.

$$\ell(h) \leq \frac{1}{1-\frac{1}{2\lambda}}\left(\widehat{\ell}(h) + \frac{\lambda}{N}\left((\ln 2)|h| + \ln\frac{1}{\delta}\right)\right)$$

See "A PAC-Bayesian Tutorial with a Dropout Bound" , McAllester (2013)

The Occam's Razor Theorem

It suffices to find any regularity in the training data where the regularity can be expressed concisely relative to the amount of training data.

The VGG vision architecture has 138 million parameters.

Imagenet contains approximately 1 trillion pixels.

The Turing Tarpit Theorem

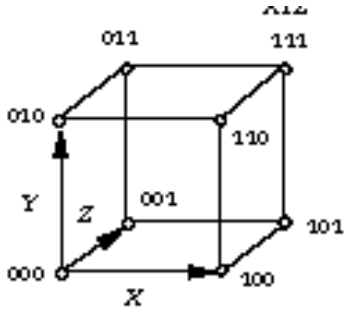
The choice of programming language does not matter.

For any two Turing universal languages L_1 and L_2 there exists a compiler $C : L_1 \to L_2$ such that

$$|C(h)| \leq |h| + |C|$$

Universality of Shallow Architectures

We will now consider theorems that apply to any continuous $f : [0, 1]^m \to \mathbb{R}$



$\xrightarrow{f}$ $\mathbb{R}$

Note that the values at the corners are independent — we can use multilinear interpolation to fill in the interior of the cube.

There are 2^m corners. Each corner has an independent value.

So it must take at least 2^m bits of information to specify f.

The Kolmogorov-Arnold representation theorem (1956)

Any continuous function of m inputs can be represented **exactly** by a small (polynomial sized) two-layer network.

$$f(x_1, \ldots, x_m) = \sum_{i=1}^{2m+1} g_i \left(\sum_{j=1}^{m} h_{i,j}(x_j) \right)$$

Where g_i and $h_{i,j}$ are continuous scalar-to-scalar functions.

A Simpler, Similar Theorem

For any (possibly discontinuous) $f : [0,1]^m \to \mathbb{R}$ we have

$$f(x_1,\ \ldots,\ x_m) = g\left(\sum_i h_i(x_i)\right)$$

for (discontinuous) scalar-to-scalar functions g and h_i.

Proof: Any single real number contains an infinite amount of information.

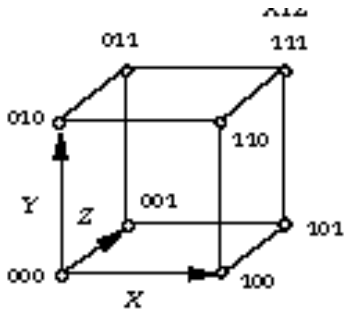
Select h_i to spread out the digits of its argument so that $\sum_i h_i(x_i)$ contains all the digits of all the x_i.

A Reference

F. Girosi and T. Poggio. Representation properties of networks: Kolmogorov s theorem is irrelevant. Neural Comp., 1:465 469, 1989.

Cybenko's Universal Approximation Theorem (1989):

Again consider any continuous $f : [0,1]^m \rightarrow \mathbb{R}$



$\xrightarrow{f}$ $\mathbb{R}$

Again note that f must contain an exponential amount of information (an independent value at every corner).

Cybenko's Universal Approximation Theorem (1989)

Any continuous function can be approximated arbitrarily well by a two layer perceptron.

For any continuous $f : [0,1]^m \to \mathbb{R}$ and any $\varepsilon > 0$, there exists

$$F(x) = \alpha \cdot \sigma(Wx + \beta)$$

$$= \sum_i \alpha_i \sigma \left(\sum_j W_{i,j}\, x_j + \beta_i \right)$$

such that for all x in $[0,1]^m$ we have $|F(x) - f(x)| < \varepsilon$.

How Many Hidden Units?

Consider Boolean functions $f : \{0,1\}^m \rightarrow \{0,1\}$.

For Boolean functions we can simply list the inputs $x^0, \ldots, x^k$ where the function is true.

$$f(x) = \sum_k I[x = x^k]$$

$$I[x = x^k] \approx \sigma\left(\sum_i W_{k,i} x_i + b_k\right)$$

This is analogous to observing that every Boolean function can be put in disjunctive normal form.

The Cybenko Theorem is Irrelevant

The number of inputs where a Boolean function f is true is typically exponential in the number of arguments to f.

The number of hidden units (channels) needed for Cybenko's theorem is exponential.

Shallow Circuit Inexpressibility Theorems

Building on work of Ajtai, Sipser and others, Hastad proved (1987) that any bounded-depth Boolean circuit computing the parity function must have exponential size.

Matus Telgarsky recently gave some formal conditions under which shallow networks provably require exponentially more parameters than deeper networks (COLT 2016).

Circuit Complexity Theory

Circuit complexity theory — the study of circuit size as a function of the components allowed and the depth allowed — is a special case of the study of deep real-valued computation.

Depth and size requirements are hard to prove. Little has been proven about depth and size requirements when additive threshold gates are allowed.

Still, we believe that deeper circuits can be smaller in total size than shallow circuits.

Finding the Program

The Occam's Razor theorem says that a concise program that does well on the training data will do well on the test data.

But the theorem does not tell us how to find such a program.

Levin's Universal Problem Solver (Levin Search)

Leonid Levin observed that one can construct a universal solver that takes as input an oracle for testing proposed solutions and, if a solution exists, returns it.

One can of course enumerate all candidate solutions.

However, Levin's solver is universal in the sense that it is not more than a constant factor slower than any other solver.

Levin's Universal Solver

We time share all programs giving time slice $2^{-|h|}$ to program h where $|h|$ is the length in bits of h.

The run time of the universal solver is at most

$$O(2^{-|h|}(h(n) + T(n)))$$

where $h(n)$ is the time required to run program h on a problem of size n and $T(n)$ is the time required to check the solution.

Here $2^{-|h|}$ is independent of n and is technically a constant.

Bootstrapping Levin Search

While Levin search sounds like a joke, Jurgin Schmidhuber (inventor of LSTMs and other deep architectural motifs) takes it seriously.

He has proposed ways of accelerating a search over all programs and has something called the Optimal Ordered Problem Solver (OOPS).

The basic idea is bootstrapping — we automate a search for methods of efficient search.

Deep Learning and Evolution

The Baldwin Effect

In a 1987 paper entitled "How Learning Can Guide Evolution", Goeffrey Hinton and Steve Nowlan brought attention to a paper by Baldwin (1896).

The basic idea is that learning facilitates modularity.

For example, longer arms are easier to evolve if arm control is learned — arm control is then independent of arm length. Arm control and arm structure become more modular.

If each "module" is learning to participate in the "society of mind" then each model can more easily accommodate (exploit) changes (improvements) in other modules.

Recent Neuroscience: Quanta Magazine, January 10, 2017

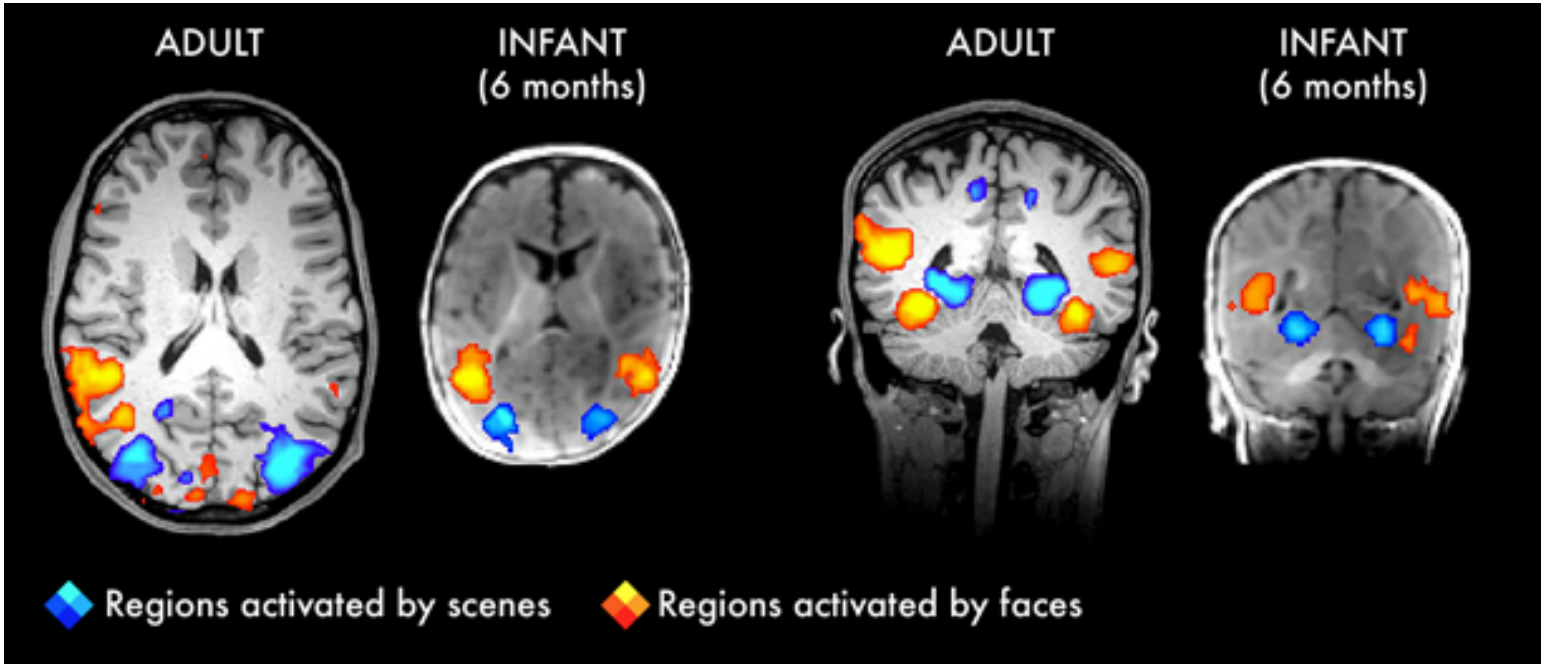


Image from Department of Brain and Cognitive Sciences and McGovern Institute, Massachusetts Institute of Technology

The g factor

Subtest intercorrelations in a sample of Scottish subjects who completed the WAIS-R battery. The subtests are Vocabulary, Similarities, Information, Comprehension, Picture arrangement, Block design, Arithmetic, Picture completion, Digit span, Object assembly, and Digit symbol. The bottom row shows the *g* loadings of each subtest.[6]

	V	S	I	C	PA	BD	A	PC	DSp	OA	DS
V	-										
S	.67	-									
I	.72	.59	-								
C	.70	.58	.59	-							
PA	.51	.53	.50	.42	-						
BD	.45	.46	.45	.39	.43	-					
A	.48	.43	.55	.45	.41	.44	-				
PC	.49	.52	.52	.46	.48	.45	.30	-			
DSp	.46	.40	.36	.36	.31	.32	.47	.23	-		
OA	.32	.40	.32	.29	.36	.58	.33	.41	.14	-	
DS	.32	.33	.26	.30	.28	.36	.28	.26	.27	.25	-
g	.83	.80	.80	.75	.70	.70	.68	.68	.56	.56	.48

END