TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

AGI: Universality

Universality

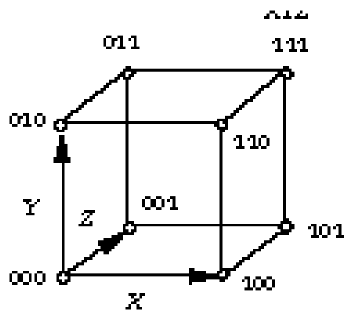
Various models of computation are “universal”.

Turing universality underlies most faith in the possibility of artificial general intelligence (AGI).

We will review some theory of universailty for deep networks, boolean circuits and computer programs.

The Kolmogorov-Arnold representation theorem (1956)

Consider continuous functions $f : [0,1]^N \to \mathbb{R}$



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

Given the corner values, the interior can be filled.

$$f(x_1, \ldots, x_N) = E_{y_1,\ldots,y_N \sim \mathrm{Round}(x_1,\ldots,x_N)}\, f(y_1, \ldots, y_n)$$

Hence each of the 2^N corners has an independent value.

The Kolmogorov-Arnold representation theorem (1956)

For continuous $f : [0,1]^N \to \mathbb{R}$ there exists continuous "activation functions" $\sigma_i : \mathbb{R} \to \mathbb{R}$ and continuous $w_{i,j} : \mathbb{R} \to \mathbb{R}$ such that

$$f(x_1, \ldots, x_N) = \sum_{i=1}^{2N+1} \sigma_i \left(\sum_{j=1}^{N} w_{i,j}(x_j) \right)$$

A Simpler, Similar Theorem

For (possibly discontinuous) $f : [0, 1]^N \to \mathbb{R}$ there exists (possibly discontinuous) $\sigma, w_i : \mathbb{R} \to \mathbb{R}$.

$$f(x_1,\ \ldots,\ x_N) = \sigma\left(\sum_i w_i(x_i)\right)$$

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

Issues with the Arnold-Kolmogorov Theorem

This theorem relies on “real number abuse” — an inherent use of infinite precision arithmetic.

Cybenko's Universal Approximation Theorem (1989)

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

$$\begin{aligned} F(x) &= \alpha^\top \sigma(Wx + \beta) \\ &= \sum_i \alpha_i \sigma \left(\sum_j W_{i,j}\, x_j + \beta_i \right) \end{aligned}$$

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

How Many Hidden Units?

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

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

$$f(x) = \sum_k \mathbf{1}[x = x^k]$$

$$\mathbf{1}[x = x^k] \approx \sigma\left(\sum_i W_{k,i} x_i + b_k\right)$$

A simpler statement is that any Boolean function can be put in disjunctive normal form.

Representing Functions as IO Tables

The Cybenko theorem implicitly treats functions as tables of a huge number of (at least exponentially many) intput-output pairs.

🙁

Representing Functions by Circuits

Deep circuits can can represent functions more concisely than shallow circuits.

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).

Limits of the Theory of Deep Circuits

Circuit complexity theory seems to be very weak and of essentially no value in the practice of deep architecture design.

☹

Representing Functions by Programs

C programs or Python programs are Turing Universal.

Furthermore we have nice universal generalization guarantees.

The Occam Guarantee (Free Lunch Theorem)

Let $|f|$ be length in bits of a compressed program f.

$$0 \leq \mathcal{L}(f, x, y) \leq L_{\max}$$

$$\begin{aligned} \mathcal{L}(f) &= E_{(x,y)\sim \mathrm{Pop}}\; \mathcal{L}(f, x, y) \\ \hat{\mathcal{L}}(f) &= E_{(x,y)\sim \mathrm{Train}}\; \mathcal{L}(f, x, y) \end{aligned}$$

Theorem: With probability at least $1 - \delta$ over the draw of the training data the following holds simultaneously for all f.

$$\mathcal{L}(f) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(f) + \frac{5L_{\max}}{N_{\mathrm{Train}}}\left((\ln 2)|f| + \ln\frac{1}{\delta}\right)\right)$$

Gradient Descent on Programs

We do not know how to effectively search over programs.

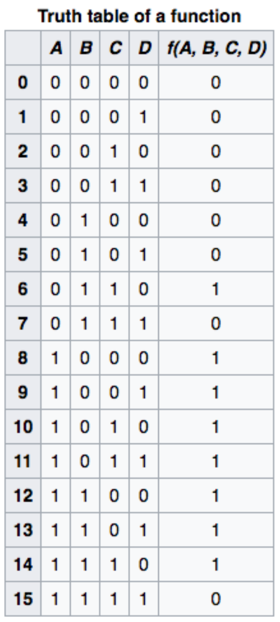
☹️

More on this later.

END

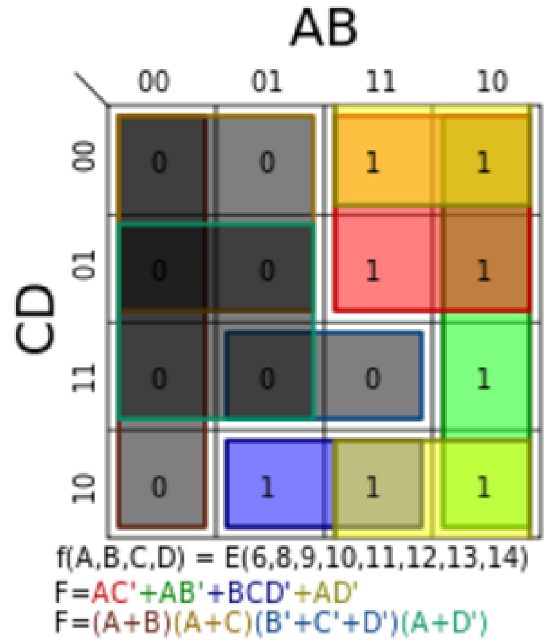
The Kaurnaugh Model of DNNs

The Karnaugh map, also known as the K-map, is a method to simplify boolean algebra expressions.



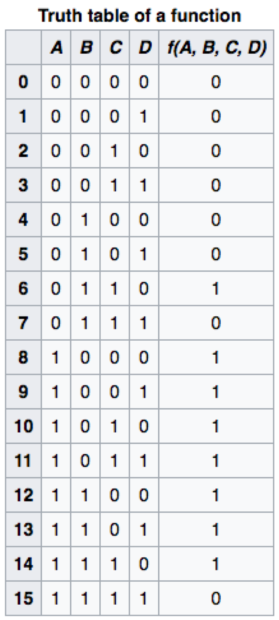
Truth table of a function

	A	B	C	D	f(A, B, C, D)
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0



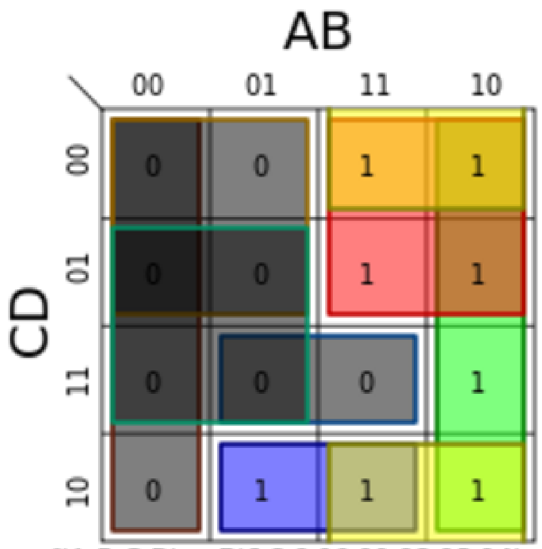
$$\begin{aligned} F(A,B,C,D) &= AC' + AB' + BCD' + AD' \\ &= (A+B)(A+C)(B'+C'+D')(A+D') \end{aligned}$$

The Kaurnaugh Model of DNNs



Truth table of a function

	A	B	C	D	f(A, B, C, D)
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0



f(A,B,C,D) = E(6,8,9,10,11,12,13,14)

F=AC'+AB'+BCD'+AD'

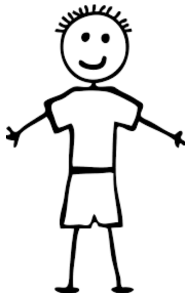
F=(A+B)(A+C)(B'+C'+D')(A+D')

Many very different circuits compute the same function.

A Kaurnaugh Person Detector

Wheel or Face

Hand or Flower   Hand or Flower

Leg or Tree  Leg or Tree

The set of locally minimal models (circuits) could be vast (exponential) without damaging performance.