

Exact distribution of the output of a deep-layered machine

Thomas Fink

London Institute for Mathematical Sciences, Royal Institution, 21 Albemarle St, London W1S 4BS, UK

Deep learning builds complex global functions through the repeated composition of local functions. But the resulting distribution over global functions is unknown. We derive this distribution exactly for finite width and depth when the local functions are chosen at random. Surprisingly, depth first creates a sharply structured distribution, then erases this structure as probability drains into two absorbing states. This competition produces a crossover at a depth exponential in the number of inputs, beyond which additional layers mainly drive collapse. The transient structure may help explain why highly overparameterized networks can still generalize.

Composition is how local rules generate complex behavior, from biological networks to digital computation to artificial intelligence. In deep learning, successive layers build global functions from local operations, yet the resulting distribution over functions remains largely unknown. Here we derive the exact output distribution of a deep-layered Boolean machine at finite width and depth. The model provides a solvable setting for understanding how repeated composition reshapes function space.

Random compositions of Boolean functions have been studied extensively in the context of Boolean networks [1, 2] and the Kauffman model [3–6]. In those systems, the network architecture is random and the local Boolean functions typically depend on only a few inputs. Deep-layered machines represent a complementary regime: the architecture is highly ordered, but each node computes an arbitrary Boolean function of all of the nodes in the layer below. Previous work has characterized asymptotic properties of deep-layered machines [7, 8], in particular the collapse of the function space onto constant functions in the infinite-depth limit. Yet the asymptotic limit conceals the most consequential behavior: repeated composition first creates functional structure and then destroys it.

We consider a deep-layered Boolean machine with k arguments and network depth n , in which every node is assigned a Boolean function chosen uniformly from the 2^{2^k} possibilities (see Fig. 1). A key insight is that the probability of an output depends only on its Hamming weight: the number of input configurations for which the function returns true. This symmetry reduces the problem from a Markov process on 2^{2^k} states to one on just $2^k + 1$ states. We derive the exact transition matrix that takes the system from depth n to $n + 1$ and obtain the full spectrum of eigenvalues: $\lambda_r = \binom{2^k}{r} / 2^{k^r}$.

Our analysis reveals two competing effects as the depth n increases. First, the distribution of the bulk of the output functions (everything but true and false) develops a pronounced bias: functions with low or high Hamming weight become exponentially more likely than those with middling weight. Second, probability steadily accumulates in the constant functions true and false, which eventually dominate the distribution. Crucially, these pro-

cesses proceed at different rates: the bulk takes shape faster than it loses probability.

Strikingly, these opposing tendencies produce a crossover at a depth of order 2^k . Before this scale, added depth mainly reshapes the bulk distribution; beyond it, probability loss into true and false takes over, and additional layers progressively erase functional diversity. We confirm these predictions with independent computer experiments, using exhaustive enumeration for small systems and Monte Carlo sampling for larger ones.

Because the model strips away learning and architectural variation, it isolates the effect of repeated composition on the distribution over functions. Even though the local functions are chosen uniformly, composition generates a strongly nonuniform global function distribution. Thus depth can shape the prior over functions before training enters. Such function priors have recently been empirically associated with generalization in deep neural networks [9].

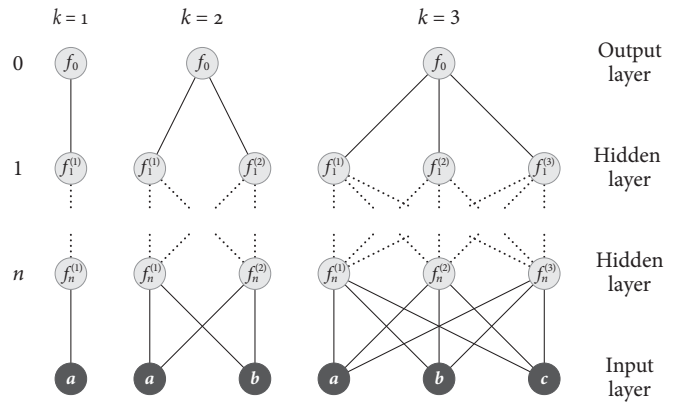


FIG. 1: Deep-layered machines. A deep-layered machine takes k Boolean arguments ($a, b, \dots$) as inputs, shown by the dark nodes. Each light node applies a random Boolean function f to the k outputs from the layer below, producing a single Boolean output. Composing these functions down to depth n produces the global function $F_n(a, b, \dots)$. Each f is chosen independently and uniformly from the 2^{2^k} Boolean functions of k inputs. The goal of this paper is to understand the distribution of the global function F_n .

Exact distribution

We use f for the local Boolean functions assigned to the network nodes and F for the global function computed by the entire network. Whereas f is a function of the k outputs of the layer immediately below it, F is a function of the k terminal arguments $a, b, \dots$ (Fig. 1).

Before solving the general case, let's consider the output distribution for small k and n . For $k = 1$ and 2, the 4 and 16 possible output functions are shown on the left of Table I, both algebraically and by their truth tables. A truth table is a binary string of length 2^k specifying the function's output for every combination of its k inputs, where 0 is false and 1 is true. We group functions with the same Hamming weight w , that is, the number of 1s in the truth table. On the right of Table I are the probabilities of the different global functions F for various n . These start out uniform but become highly nonuniform as n increases.

As a concrete example, take $k = 2$ and $n = 1$. If

A ($k = 1$)			Probability of $F(a)$			
Output $F(a)$	w		$n=0$	$n=1$	$n=2$	$n=3$
false 00	0	}	$\frac{1}{4}$	$\frac{6}{4^2}$	$\frac{28}{4^3}$	$\frac{120}{4^4}$
a 10	1		$\frac{1}{4}$	$\frac{2}{4^2}$	$\frac{4}{4^3}$	$\frac{8}{4^4}$
$\bar{a}$ 01	1		$\frac{1}{4}$	$\frac{6}{4^2}$	$\frac{28}{4^3}$	$\frac{120}{4^4}$
true 11	2	}	$\frac{1}{4}$	$\frac{6}{4^2}$	$\frac{28}{4^3}$	$\frac{120}{4^4}$

B ($k = 2$)			Probability of $F(a, b)$			
Output $F(a, b)$	w		$n=0$	$n=1$	$n=2$	$n=3$
false 0000	0	}	$\frac{1}{16}$	$\frac{680}{16^3}$	$\frac{261056}{16^5}$	$\frac{83663360}{16^7}$
ab 1000	1		$\frac{1}{16}$	$\frac{216}{16^3}$	$\frac{42048}{16^5}$	$\frac{8087040}{16^7}$
$a\bar{b}$ 0100	1					
$\bar{a}b$ 0010	1					
$\bar{a}\bar{b}$ 0001	1					
a 1100	2	}	$\frac{1}{16}$	$\frac{168}{16^3}$	$\frac{31680}{16^5}$	$\frac{6068736}{16^7}$
$\bar{a}$ 0011	2					
b 1010	2					
$\bar{b}$ 0101	2					
$a \oplus b$ 0110	2	}	$\frac{1}{16}$	$\frac{216}{16^3}$	$\frac{42048}{16^5}$	$\frac{8087040}{16^7}$
$a \oplus \bar{b}$ 1001	2					
$a + b$ 1110	3					
$a + \bar{b}$ 1101	3					
$\bar{a} + b$ 1011	3	}	$\frac{1}{16}$	$\frac{680}{16^3}$	$\frac{261056}{16^5}$	$\frac{83663360}{16^7}$
$\bar{a} + \bar{b}$ 0111	3					
true 1111	4	}	$\frac{1}{16}$	$\frac{680}{16^3}$	$\frac{261056}{16^5}$	$\frac{83663360}{16^7}$

TABLE I: **Probability of output functions.** For $k = 1$ and 2 arguments, we show the output function distribution $\mathbf{p}(n)$ at several network depths n . Functions with the same Hamming weight w have the same probability. The probability shown for each w group is that of each individual function in the group. At $n = 0$ all functions are equally likely; as depth increases, functions with low or high w become more probable. The functions are shown algebraically and by their truth tables. In our notation, $\bar{a}$ means NOT a , ab means a AND b , $a \oplus b$ means a XOR b , and $a + b$ means a OR b .

f_0 is AND, $f_1^{(1)}$ is OR, and $f_1^{(2)}$ is NAND, then $F(a, b) = f_0(f_1^{(1)}(a, b), f_1^{(2)}(a, b)) = (a + b)(\bar{a} + \bar{b}) = a\bar{b} + \bar{a}b = a \oplus b$. Repeating this process over the 16^3 possible assignments to the three local functions gives the distribution of $F(a, b)$ shown in Table IB.

We now turn to the general solution for arbitrary k and n . We increase the depth by adding a new layer of k random local functions at the bottom of the machine, whose outputs become the k arguments of F_{n-1} . Thus

$$F_n(a, b, \dots) = F_{n-1}(f_n^{(1)}(a, b, \dots), \dots, f_n^{(k)}(a, b, \dots)).$$

For any fixed input configuration $(a, b, \dots)$, the outputs $f_n^{(1)}(a, b, \dots), \dots, f_n^{(k)}(a, b, \dots)$ are independent, and each is 0 or 1 with probability $1/2$. These k outputs therefore form one of the 2^k possible input configurations of F_{n-1} , each with probability $1/2^k$. Since F_{n-1} returns 1 for $w(F_{n-1})$ of these configurations, the corresponding output of F_n is 1 with probability $w(F_{n-1})/2^k$. Writing the 2^k entries of the truth table of F_n as $\sigma_1, \dots, \sigma_{2^k}$, the σ_i are thus independent and identically distributed, with

$$P(\sigma_i = 1) = \frac{w(F_{n-1})}{2^k}, \quad P(\sigma_i = 0) = 1 - \frac{w(F_{n-1})}{2^k}.$$

For convenience, let $\ell = 2^k$. The probability of obtaining a particular F_n , given F_{n-1} , is then

$$P(F_n | F_{n-1}) = \prod_{i=1}^{\ell} \left(\frac{w(F_{n-1})}{\ell} \right)^{\sigma_i} \left(1 - \frac{w(F_{n-1})}{\ell} \right)^{1-\sigma_i}.$$

Noting that $\sum_i \sigma_i = w(F_n)$, and summing over F_{n-1} , we obtain

$$P(F_n) = \frac{1}{\ell^\ell} \sum_{F_{n-1}} w(F_{n-1})^{w(F_n)} (\ell - w(F_{n-1}))^{\ell - w(F_n)} P(F_{n-1}),$$

where $0^0 = 1$, a common convention in combinatorics.

Let $\mathbf{p}(n)$ be the probability distribution of the 2^ℓ output functions F_n , in lexicographical order. For example, for $k = 1$, $\mathbf{p}(0) = (1/4, 1/4, 1/4, 1/4)$ and $\mathbf{p}(1) = (6/4^2, 2/4^2, 2/4^2, 6/4^2)$, with more examples in Table I (though not in lexicographical order). The elements of $\mathbf{p}(n)$ satisfy

$$p_i(n) = \frac{1}{\ell^\ell} \sum_{j=1}^{2^\ell} w_j^{w_i} (\ell - w_j)^{\ell - w_i} p_j(n-1),$$

where $w_i = 0, 1, 1, 2, 1, 2, 2, 3, \dots$ is the Hamming weight of the binary representation of $i - 1$, and $p_i(0) = 1/2^\ell$.

Notice that the probability of an output at level n depends on an output at level $n - 1$ only through its Hamming weight. We can therefore replace the distribution $\mathbf{p}$ over individual output functions by the simpler distribution $\mathbf{q}$ over Hamming weight. The symmetry in $\mathbf{p}$ reduces the problem from 2^{2^k} output probabilities to

$2^k + 1$ Hamming-weight probabilities. For example, for $k = 1$, $\mathbf{q}(0) = (1/4, 2/4, 1/4)$ and $\mathbf{q}(1) = (6/4^2, 4/4^2, 6/4^2)$, with more examples in Table I. The distribution of the Hamming weight satisfies

$$q_i(n) = \frac{1}{\ell^\ell} \sum_{j=0}^{\ell} \binom{\ell}{i} j^i (\ell - j)^{\ell-i} q_j(n-1).$$

In other words, $\mathbf{q}(n) = \mathbf{A}\mathbf{q}(n-1)$, where $\mathbf{A}$ is the $\ell + 1$ by $\ell + 1$ transition matrix

$$A_{i,j} = \frac{1}{\ell^\ell} \binom{\ell}{i} j^i (\ell - j)^{\ell-i}.$$

We index the rows and columns of $\mathbf{A}$ from 0 to ℓ , and recall that $\ell = 2^k$ and $0^0 = 1$. This matrix representation immediately gives

$$\mathbf{q}(n) = \mathbf{A}^n \mathbf{q}(0), \quad (1)$$

where $q_i(0) = \binom{\ell}{i} / 2^\ell$.

To get a sense of the structure of $\mathbf{A}$, it helps to see some examples. For $k = 1$,

$$\mathbf{A} = \frac{1}{2^2} \begin{pmatrix} \binom{2}{0} \\ \binom{2}{1} \\ \binom{2}{2} \end{pmatrix} \begin{pmatrix} 0^0 2^2 & 1^0 1^2 & 2^0 0^2 \\ 0^1 2^1 & 1^1 1^1 & 2^1 0^1 \\ 0^2 2^0 & 1^2 1^0 & 2^2 0^0 \end{pmatrix}.$$

For $k = 2$,

$$\mathbf{A} = \frac{1}{4^4} \begin{pmatrix} \binom{4}{0} \\ \binom{4}{1} \\ \binom{4}{2} \\ \binom{4}{3} \\ \binom{4}{4} \end{pmatrix} \begin{pmatrix} 0^0 4^4 & 1^0 3^4 & 2^0 2^4 & 3^0 1^4 & 4^0 0^4 \\ 0^1 4^3 & 1^1 3^3 & 2^1 2^3 & 3^1 1^3 & 4^1 0^3 \\ 0^2 4^2 & 1^2 3^2 & 2^2 2^2 & 3^2 1^2 & 4^2 0^2 \\ 0^3 4^1 & 1^3 3^1 & 2^3 2^1 & 3^3 1^1 & 4^3 0^1 \\ 0^4 4^0 & 1^4 3^0 & 2^4 2^0 & 3^4 1^0 & 4^4 0^0 \end{pmatrix}.$$

For most of this paper we work with $\mathbf{q}$, the distribution of output Hamming weights. But we're ultimately interested in $\mathbf{p}$, the distribution of individual output functions. To translate between them,

$$p_j = q_i / \binom{2^k}{i},$$

for all $j \in [1, 2^\ell]$ such that $w(F_j) = i$.

Spectral structure

The matrix $\mathbf{A}$ has $2^k + 1$ eigenvalues and eigenvectors. The first two we can work out for free. That's because the first and last columns of $\mathbf{A}$ are $(1, 0, \dots, 0)$ and $(0, \dots, 0, 1)$. So these are the first two eigenvectors, with corresponding eigenvalues $\lambda_0 = \lambda_1 = 1$.

More generally, the r th eigenvalue is

$$\lambda_r = \frac{(\ell)_r}{\ell^r},$$

where $\ell = 2^k$ and $(\ell)_r = \ell(\ell-1)\dots(\ell-r+1)$ is the falling factorial. The spectrum follows directly from the binomial form of the columns of $\mathbf{A}$. Since $\mathbf{A}$ and $\mathbf{A}^T$ have the same eigenvalues, we can obtain the spectrum by applying $\mathbf{A}^T$ to the falling-factorial basis $((i)_0, (i)_1, \dots, (i)_\ell)$. Using the factorial-moment identity for the binomial distribution,

$$\sum_{i=0}^{\ell} A_{i,j} (i)_r = \frac{(\ell)_r}{\ell^r} j^r = \frac{(\ell)_r}{\ell^r} (j)_r + \text{lower-degree terms}.$$

Thus $\mathbf{A}^T$ is triangular in the falling-factorial basis, with diagonal entries $\lambda_r = (\ell)_r / \ell^r$, which are therefore the eigenvalues of $\mathbf{A}$.

In the limit of large network depth, the first two eigenvectors dominate, so the probabilities of all non-constant outputs vanish and the output becomes a toss between true and false. But this simple asymptotic picture conceals the finite-depth structure of the bulk. Its shape is governed by the leading transient eigenvector $\mathbf{v}_2$. Although $\mathbf{v}_2$ has no simple finite- ℓ form, its bulk profile can be determined exactly in the large- ℓ limit.

Again let $x = i/\ell$, and write the large- ℓ limit of the bulk of $\mathbf{v}_2$ as $v_{2,i} = V(x)/\ell$. The binomial kernel of $\mathbf{A}$ has zero drift and variance $x(1-x)/\ell$, so in the bulk

$$\mathbf{A} = \mathbf{I} + \frac{1}{\ell} \mathbf{G} + O(\ell^{-2}), \quad \mathbf{G}\phi = \frac{1}{2} \frac{d^2}{dx^2} (x(1-x)\phi(x)).$$

Since $\lambda_2 = 1 - 1/\ell$, the equation $\mathbf{A}\mathbf{v}_2 = \lambda_2 \mathbf{v}_2$ gives

$$\frac{1}{2} \frac{d^2}{dx^2} (x(1-x)V(x)) = -V(x).$$

Expanding the derivative and eliminating V , we can write

$$\frac{d}{dx} (x^2(1-x)^2 V'(x)) = 0.$$

So $x^2(1-x)^2 V'(x) = C$, a constant. Since by reflection symmetry $V(x) = V(1-x)$, the midpoint is stationary, so $V'(1/2) = 0$ and $C = 0$. Since $x^2(1-x)^2 > 0$ for $x \in (0, 1)$, $V' = 0$, so V is flat.

Now we just need to determine its value. Since $v_{2,0} = v_{2,\ell} = -1$ and the components of $\mathbf{v}_2$ sum to zero, we have

$$\lim_{\ell \rightarrow \infty} \left(-2 + \frac{\ell-1}{\ell} V \right) = 0,$$

so $V = 2$. Therefore in the limit of large ℓ , for $0 < i/\ell < 1$,

$$v_{2,i} = 2/\ell.$$

Thus the leading transient mode is flat across Hamming-weight classes in the bulk.

Finite-depth crossover

As the network depth n increases, the output distribution $\mathbf{p}$ changes in two competing ways. First, the bulk of $\mathbf{p}$ shifts from uniform to U-shaped with respect to

Hamming weight (Fig. 3). Outputs with low or high Hamming weights become exponentially more likely than those with intermediate ones. Second, the probability of the constant functions true and false shifts from negligible ($1/2^k$ each) to dominant ($1/2$ each), causing the combined probability of all of the other functions to vanish.

Strikingly, this competition produces a crossover at depth of order 2^k . The bulk acquires its characteristic shape before its probability substantially diminishes; at greater depth, loss into true and false dominates and the distribution progressively collapses onto these two absorbing states.

To estimate the crossover depth, we retain the three slowest modes in the spectral expansion of $\mathbf{q}(n)$. The first two eigenvectors describe the absorbing endpoints, while the leading transient eigenvector $\mathbf{v}_2$ governs the bulk. Then

$$\begin{aligned}\mathbf{q}(n) &\simeq c_0 \mathbf{v}_0 \lambda_0^n + c_1 \mathbf{v}_1 \lambda_1^n + c_2 \lambda_2^n \mathbf{v}_2 \\ &\simeq c_0 \mathbf{v}_0 + c_1 \mathbf{v}_1 + c_2 e^{-n/\ell} \mathbf{v}_2,\end{aligned}\quad (2)$$

since $\lambda_0 = \lambda_1 = 1$ and $\lambda_2^n = (1 - 1/\ell)^n \simeq e^{-n/\ell}$.

Now let's work out the projection coefficients. At $n = 0$, all 2^ℓ output functions are equally likely, each with probability $1/2^\ell$. Since there are $\binom{\ell}{i}$ functions of Hamming weight i , the initial Hamming-weight distribution is $\mathbf{q}_i(0) = \binom{\ell}{i}/2^\ell$. Projecting this initial condition onto the eigenvectors gives $c_0 = c_1 = 1/2$. The coefficient c_2 can be calculated in the large- ℓ limit as follows. Consider

the left eigenvector $\mathbf{u}_2$ of $\mathbf{A}$. Unlike the right eigenvector $\mathbf{v}_2$, we can readily calculate it: $u_{2,i} = i(\ell - i)$. Then

$$c_2 = \frac{\mathbf{u}_2 \mathbf{q}(0)}{\mathbf{u}_2 \mathbf{v}_2} \simeq \frac{\ell(\ell - 1)/4}{\ell(\ell + 1)/3} = \frac{3(\ell - 1)}{4(\ell + 1)} \simeq \frac{3}{4}.$$

Here the numerator $\ell(\ell - 1)/4$ is exact, whereas for the denominator we use the flat finite- ℓ approximation $v_{2,i} \simeq 2/(\ell - 1)$, which preserves the exact interior sum. Then eq. (2) becomes

$$\mathbf{q}(n) \simeq (\tfrac{1}{2}, 0, \dots, 0, \tfrac{1}{2}) + 3/4 e^{-n/\ell} (-1, \tfrac{2}{\ell-1}, \dots, \tfrac{2}{\ell-1}, -1).$$

The crossover depth occurs when the two endpoints and the bulk have equal probability, giving

$$n_c \simeq \ln(3)2^k.$$

This is marked in Fig. 2.

Importantly, the normalized bulk of $\mathbf{q}$ approaches $\mathbf{v}_2$ much faster than its total probability decays. Since the transition matrix and the binomial initial condition are symmetric under $i \leftrightarrow \ell - i$, the projections onto the antisymmetric eigenvectors vanish, so $c_3 = c_5 = \dots = 0$. The leading correction is therefore $\mathbf{v}_4$, whose amplitude relative to $\mathbf{v}_2$ decays as

$$(\lambda_4/\lambda_2)^n = ((1 - 2/\ell)(1 - 3/\ell))^n \simeq e^{-5n/\ell}.$$

By contrast, the total bulk probability decays as $e^{-n/\ell}$. Thus the bulk acquires its characteristic shape well before it disappears, creating a distinct finite-depth regime before collapse onto true and false.

Comparison with experiments

To confirm our theoretical prediction of the output distribution, we conducted extensive experiments for various values of the number of arguments k and network depth n . We did complete enumeration when possible, and otherwise sampled from the ensemble of configurations. In all cases, our computer experiments match our theory.

For $k = 2$, there are $16^3, 16^5, 16^7$ and 16^9 configurations for depths $n = 1, 2, 3$ and 4 . We enumerated all of these configurations and, for each, determined the network's output function. We plot the probability $\mathbf{q}_i(n)$ of obtaining an output with Hamming weight i in Fig. 3A (points). This exactly matches our prediction given by eq. (1): the solid line shows the probabilities of the bulk and the dashed line the outputs true ($w = 4$) and false ($w = 0$). As n increases, the likelihood of true and false approach $1/2$ and the likelihoods of all other outputs vanish.

For $k = 3$, there are $256^4, 256^7$ and 256^{10} configurations for $n = 1, 2$ and 3 . For $n = 1$, we enumerated all of the configurations. For $n = 2$ and 3 , this is computationally infeasible, so we sampled them instead.

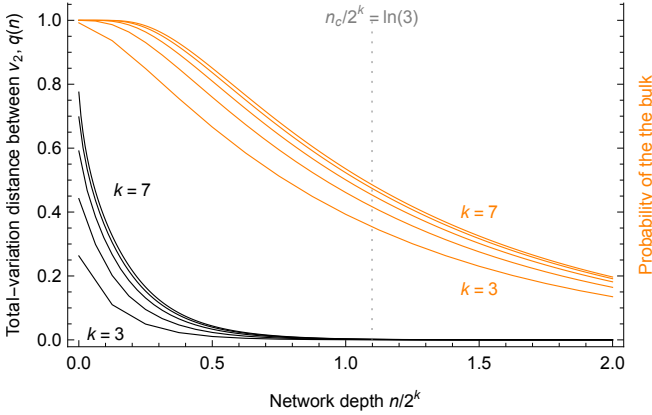


FIG. 2: **Bulk relaxation and crossover.** The orange curves show the total bulk probability, which decays as probability accumulates in the absorbing states true and false. The black curves show the total-variation distance between the normalized interior of $\mathbf{q}(n)$ and that of $\mathbf{v}_2$. The leading deviation from $\mathbf{v}_2$ decays as $\exp(-5n/2^k)$, consistent with the rapidly decreasing black curves, whereas the total bulk probability decays as $\exp(-n/2^k)$. Thus the bulk acquires its dominant shape well before it disappears. The vertical line marks the crossover depth $n_c \simeq \ln 3 2^k$, where true and false together carry half the probability.

We randomly assigned one of the 256 functions to each of the nodes and then determined the network output, repeating this two million times. These are plotted in Fig. 3B (points). We calculated error bars, but these are negligible. Our theory agrees with the computer experiments exactly for $n = 1$ and to within statistical

significance for $n = 2$ and 3.

For $k = 4$, there are 65536^5 and 65536^9 configurations for $n = 1$ and 2. These are too many to enumerate, so we sampled as above. These are plotted in Fig. 3C. Once again, our experiments confirm our theory.

Discussion

We have derived the exact output distribution of a deep-layered Boolean machine at finite width and depth. Increasing depth does not simply drive the output distribution toward its trivial infinite-depth limit. Rather, repeated composition first creates a strongly structured distribution over the bulk (everything but true and false). Once the leading transient mode dominates,

$$P(F) \simeq \frac{3e^{-n/2^k}}{2^{k+1} \binom{2^k}{w(F)}}.$$

At greater depth, the distribution ultimately collapses onto a coin toss between true and false.

The separation of relaxation rates creates distinct finite-depth regimes. At first, adding layers reshapes the bulk distribution, with little probability reaching true and false; later, the bulk has largely acquired its characteristic form while probability increasingly drains into true and false. Mozeika, Li and Saad observed that “random deep ReLU networks compute only constant Boolean functions in the infinite-depth limit” [8]. Our finite-depth solution agrees with this asymptotic result, but reveals what it conceals: an extended transient regime in which the bulk distribution develops its strongest structure.

Our model is tractable because a vast microscopic state space admits an exact macroscopic description. Although there are 2^{2^k} possible output functions, symmetry reduces the dynamics to $2^k + 1$ Hamming-weight classes. The transpose of the resulting transition matrix $\mathbf{A}$ is the conventional neutral Wright-Fisher matrix. Its action on polynomial observables preserves degree, making the eigenvalues and left eigenvectors of $\mathbf{A}$ readily accessible. The finite-depth output distribution, however, evolves under $\mathbf{A}$, not $\mathbf{A}^T$, and therefore depends on the much harder right eigenvectors of $\mathbf{A}$.

Fig. 2 suggests that the finite-depth dynamics may become universal at large k . When depth is rescaled by 2^k , the curves for different k appear to approach a limiting form. We conjecture that, as k becomes large, the dynamics is governed by the rescaled depth $n/2^k$. Such a limit would replace the family of finite- k machines by a single continuum description, providing a thermodynamic picture of deep composition.

Overparameterized neural networks have vastly more microscopic configurations than are needed to fit the training data, yet generalize well to unseen data. A potential resolution to this puzzle is that sampling these

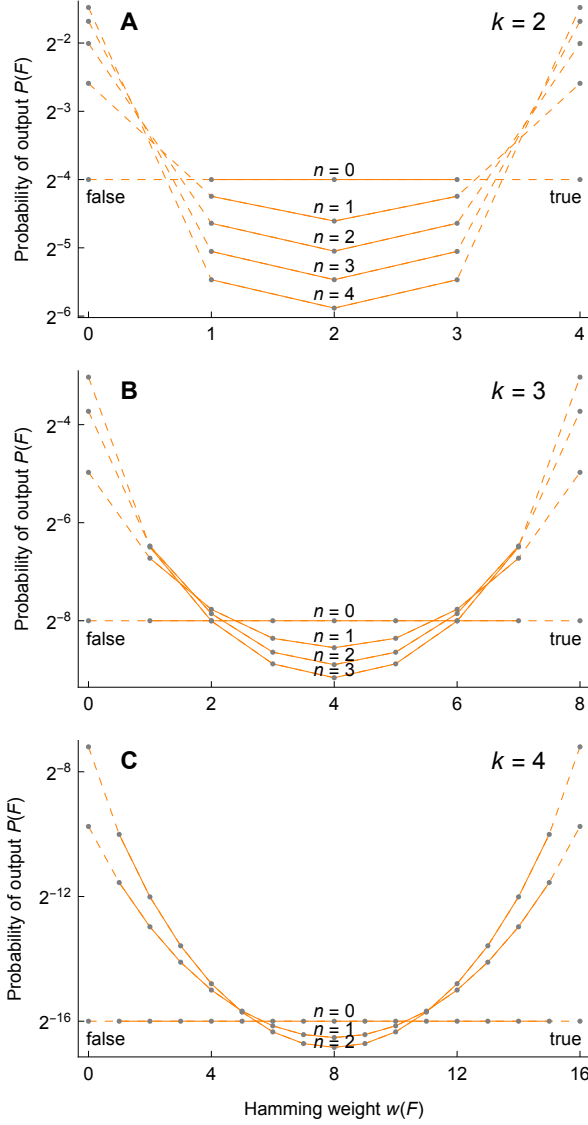


FIG. 3: Output distributions and tests of the theory. We show the output function probability $P(F_i) = p_i$ against Hamming weight $w(F_i)$ for small values of k and n , including those shown in Table I. Functions with the same Hamming weight have the same probability. Starting from the uniform distribution at $n = 0$, increasing depth makes the bulk of $P(F)$ U-shaped. Meanwhile, probability drains from the bulk and accumulates in the constant functions true and false. The curves are our exact predictions; the points are independent computer experiments, obtained by exhaustive enumeration where possible and Monte Carlo sampling otherwise. The experiments agree exactly with theory or, when sampled, within statistical uncertainty.

configurations induces a highly nonuniform distribution over global functions. Recent work by Mingard et al. [9] found empirical evidence that the function prior induced by deep neural networks helps explain their ability to generalize. Our model isolates repeated composition as one source of such a prior. The effective number of functions available to learning can be far smaller than the nominal size of function space.

Our exactly solvable model provides a tractable setting for understanding simplicity bias in input-output maps. In such maps, many microscopic descriptions generate a smaller space of macroscopic outcomes. Input-output maps occur across biology, physics and technology, and mounting empirical evidence suggests that they strongly favor simple outputs [9, 11, 12]. Because the output probabilities of our machine can be calculated analytically, it offers a rare setting in which this bias can be derived rather than just observed. We investigate the simplicity bias of deep-layered machines elsewhere [13].

[1] A. Mozeika, D. Saad, Dynamics of Boolean networks: An exact solution, *Phys Rev Lett* **106**, 214101 (2011).

[2] B. Drossel, Random Boolean networks, in *Reviews of Nonlinear Dynamics and Complexity* (Wiley-VCH, Weinheim, 2008).

[3] B. Samuelsson, C. Troein, Superpolynomial growth in the number of attractors in Kauffman networks, *Phys Rev Lett* **90**, 098701 (2003).

[4] B. Drossel, T. Mihaljev, F. Greil, Number and length of attractors in a critical Kauffman model with connectivity one, *Phys Rev Lett* **94**, 088701 (2005).

[5] T. Fink, Exact behavior of the critical Kauffman model with connectivity one, *Phys Rev Res* **6**, 043315 (2024).

[6] T. Fink, F. Sheldon, Number of attractors in the critical Kauffman model is exponential, *Phys Rev Lett* **131**, 267402 (2023).

[7] B. Li, D. Saad, Exploring the function space of deep-learning machines, *Phys Rev Lett* **120**, 248301 (2018).

[8] A. Mozeika, B. Li, D. Saad, Space of functions computed by deep-layered machines, *Phys Rev Lett* **125**, 168301 (2020).

[9] C. Mingard et al., Deep neural networks have an inbuilt Occam's razor, *Nat Commun* **16**, 220 (2025).

[10] L. Zdeborová, Understanding deep learning is also a job for physicists, *Nat Phys* **16**, 602 (2020).

[11] K. Dingle, C. Q. Camargo, A. A. Louis, Input-output maps are strongly biased towards simple outputs, *Nat Commun* **9**, 761 (2018).

[12] I. G. Johnston et al., Symmetry and simplicity spontaneously emerge from the algorithmic nature of evolution, *P Natl Acad Sci USA* **119**, e2113883119 (2022).

[13] T. Fink, Deep-layered machines have a built-in Occam's razor, arXiv:2603.01217.