Width is Less Important than Depth in ReLU Networks

Report and Experiment (TDL)

Ilyas Madah - Moghit Yebari

January 23, 2026

Abstract

This report summarizes the main ideas of *Width is Less Important than Depth in ReLU Neural Networks* (Vardi, Yehudai, Shamir; 2022) and expands on one core message: *moving from depth to width is expensive*, whereas *moving from width to depth is cheap*. "Expensive" and "cheap" are understood in the standard representation–complexity sense: representing certain deep functions with shallow networks requires *exponential* width, while representing wide functions with narrow networks can be done with only a *polynomial* (often linear) overhead. We complement the theory with a controlled experiment on a small noisy dataset: (i) a shallow/wide network overfits by memorizing label noise; (ii) the paper's constructive width→depth procedure yields a narrow/deep network computing the *same function* (to numerical precision), hence inheriting the same overfitting; and (iii) training the same narrow/deep architecture from scratch also overfits (even more severely in our run), highlighting that expressivity and optimization are distinct issues.

1 Setting and questions in the paper

A ReLU network $N : \mathbb{R}^d \to \mathbb{R}$ of depth L is a composition of affine maps and ReLU activations $\sigma(z) = \max\{0, z\}$, with width $n = \max\{d, n_1, \ldots, n_{L-1}\}$ and (dense) parameter count $O(n^2 L)$. Classic universal approximation results show that *shallow* (depth 2) ReLU networks are universal when width is allowed to be large. In contrast, width smaller than d cannot be universal regardless of depth. An open question raised by Lu et al. (2017) asks whether width and depth are *incomparable*: are there functions representable by wide/shallow networks that cannot be approximated by any narrow network unless its depth is very large?

Vardi et al. provide a negative answer for ReLU networks: roughly, *any* target ReLU network can be approximated by another network whose width scales only as $O(d)$, at the cost of additional depth/parameters. Their informal main theorem (Thm. 1.1) states that for any target network N_0 of width n and depth L and any reasonable input distribution D over a bounded domain, one can construct (probabilistically) a network N of width $O(d)$ such that $|N_0(x) - N(x)| \le \varepsilon$ with high probability over $x \sim D$, using $\widetilde{O}(n^2 L^2)$ parameters (log factors hidden) [1]. Compared to the target's $O(n^2 L)$ parameters, narrowing the width costs only a factor $\approx L$ in parameters.

2 Expanded contribution: why depth→width is expensive but width→depth is cheap

2.1 Depth→width is expensive: depth separation via linear regions

A simple and rigorous way to see why depth matters is via the number of *linear regions* of a 1D ReLU network. A depth-2 (one-hidden-layer) ReLU network in 1D with width m is a continuous

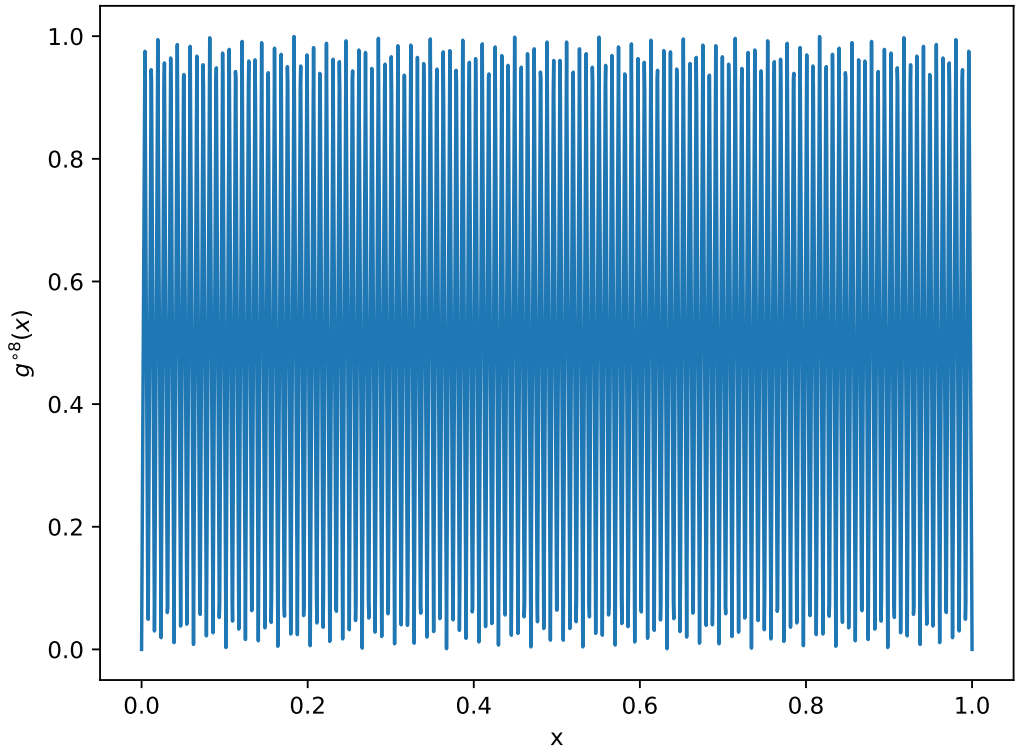


Figure 1: The iterated triangle function $g^{\circ 8}$ on $[0, 1]$ has $2^8 = 256$ linear regions. Any depth-2 ReLU network representing it exactly needs width at least 255.

piecewise-linear function with at most $m + 1$ linear pieces (breakpoints occur where a hidden unit changes sign). Therefore, any 1D function with R linear regions requires width at least $R - 1$ if restricted to depth 2.

Telgarsky's depth-separation construction uses an elementary "triangle wave" function g and its iterated composition $g^{\circ L}$, which has 2^L oscillations/linear regions on $[0, 1]$ [3]. Consequently, any depth-2 network that represents $g^{\circ L}$ exactly needs width

$$m \;\geq\; 2^L - 1, \tag{1}$$

which is *exponential* in depth. This gives a concrete sense in which flattening depth into width is expensive. (Other depth-separation results, e.g. Eldan–Shamir (2016), show similar phenomena in higher dimensions [4].)

Figure 1 visualizes $g^{\circ 8}$, whose oscillatory structure already requires width at least $2^8 - 1 = 255$ at depth 2.

2.2 Width→depth is cheap: narrow networks can simulate wide ones

The paper proves that width is often not the bottleneck: wide networks can be approximated by *narrow* ones (with width $O(d)$) at a cost that is only polynomial in the original size (and often only linear in depth).

At a high level, the proof builds a narrow network that *simulates* the computation of the target wide network. A key identity exploited repeatedly is the decomposition of an input coordinate into

its positive and negative parts:

$$x \;=\; \sigma(x) - \sigma(-x),$$

applied coordinate-wise. This allows a narrow network to keep nonnegative "state" variables that can reconstruct linear forms $\langle w, x\rangle$ by combining $\sigma(x)$ and $\sigma(-x)$. The construction then processes units of the target network sequentially (a "state machine" viewpoint), updating a running accumulator that stores partial sums.

In Appendix C.1, the authors give an explicit *exact* construction for converting a depth-2 network with hidden width n into a narrow but deeper network of width $2d + 3$ and depth $2n + 2$ [1]. The resulting model is sparse: most weights can be set to zero or identity, and only $O(nd)$ weights encode the original first-layer vectors and output coefficients. This is the canonical sense in which width→depth is cheap: the overhead is *linear* in n (and polynomial in d).

3 Experiment: overfitting and the width→depth construction

3.1 Protocol

We use a binary classification task (`make_moons`) with standardization. We sample $N = 600$ points and keep only $n_{\text{train}} = 32$ for training (very small on purpose), with Gaussian noise level 0.25. We inject label noise by flipping each training label independently with probability $p_{\text{flip}} = 0.4$ (here 11 labels were flipped). The test set (568 points) keeps the clean labels. We train:

- **Shallow/wide**: a depth-2 ReLU MLP with hidden width $n = 32$ (129 parameters), trained for 8000 epochs with Adam (lr=0.01).
- **Deep (converted)**: apply the Appendix C.1 conversion to obtain a width-$2d + 3 = 7$, depth $2n + 2 = 66$ network that computes exactly the trained shallow function (up to numerical precision).
- **Deep (from scratch)**: the same width-7, depth-66 architecture trained directly on the noisy labels (2000 epochs, Adam lr=0.002) with near-identity initialization for intermediate layers to stabilize optimization.

Accuracy on the training set is measured against the *noisy* labels used for training; we also report training accuracy w.r.t. the *clean* labels to quantify memorization of corrupted labels.

3.2 Results

Table 1 summarizes the final metrics. The shallow model achieves high training accuracy (0.9688) but much lower test accuracy (0.6021), indicating overfitting. The converted deep network matches the shallow network's outputs essentially exactly: the maximum absolute difference in logits over all 600 points (float64 evaluation) is 2.27×10^{-13} (mean 2.68×10^{-14}), hence it inherits identical train/test performance. Training the same deep architecture from scratch also memorizes the noisy labels (train accuracy 1.0000) and generalizes worse here (test accuracy 0.4806), emphasizing that representational power does not automatically imply good generalization.

3.3 Discussion: how this illustrates "expensive" vs "cheap"

This experiment directly instantiates the paper's width→depth message: a wide/shallow model can be turned into a narrow/deep model (width 7 here) while preserving the learned function essentially

Model	Depth	Width	Params	Nonzero	Train acc	Test acc
Shallow (trained)	2	32	129	129	0.9688	0.6021
Deep (converted)	66	7	3613	583	0.9688	0.6021
Deep (scratch)	66	7	3613	3613	1.0000	0.4806

Table 1: Final accuracy metrics (training accuracy measured on noisy training labels). The converted deep model is functionally equivalent to the trained shallow model; its parameter matrix is mostly sparse/identity, hence only 583 weights are nonzero in the construction.

exactly. The conversion overhead is modest in the representational sense: the constructed network has only 583 nonzero weights and depth 66, i.e. linear in the original width $n = 32$.

In contrast, the depth-separation example demonstrates that collapsing depth into width can require an exponential blow-up (e.g. width $\geq 2^L - 1$ for $g^{\circ L}$ at depth 2). Together, these results support the qualitative slogan of the paper: *depth is the more fundamental resource* for expressivity in ReLU networks.

4 Conclusion and brief critique

The paper resolves an important conceptual question by showing that, for ReLU networks and natural input distributions, width is not essential beyond $O(d)$: sufficiently deep narrow networks can approximate any wide network with only polynomial overhead in parameters [1]. A subtle but important point is that the theorem is about *existence* of approximating networks, not necessarily about whether gradient-based training will find them efficiently. Our experiment reflects this: both shallow and deep models can memorize label noise, but their optimization dynamics and inductive biases differ.

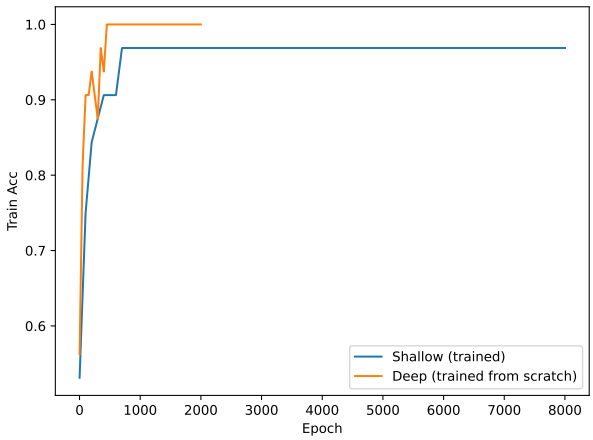
References

[1] G. Vardi, G. Yehudai, and O. Shamir. Width is less important than depth in ReLU neural networks. *arXiv:2202.03841*, 2022.

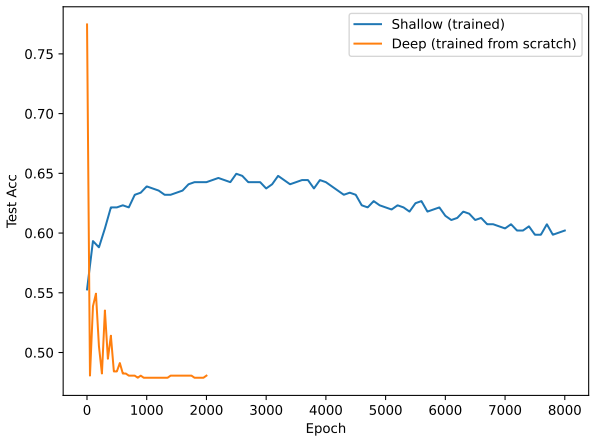
[2] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *NeurIPS*, 2017.

[3] M. Telgarsky. Benefits of depth in neural networks. In *COLT*, 2016.

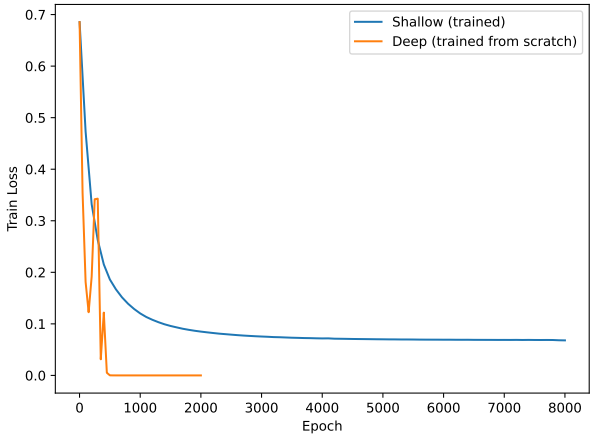
[4] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *COLT*, 2016.



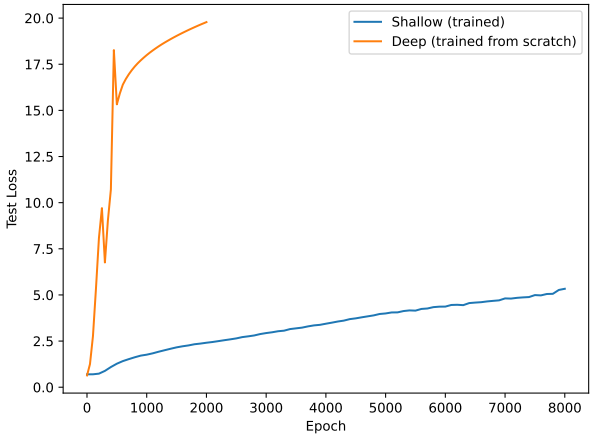
(a) Training accuracy



(b) Test accuracy



(c) Training loss



(d) Test loss

Figure 2: Learning curves for the trained shallow model (8000 epochs) and the deep-from-scratch model (2000 epochs). The converted deep model has identical curves to the trained shallow model (it computes the same function).