

Why is AI hard and Physics simple?

Dan Roberts

MIT, IAIFI, & Salesforce

September 21, 2021

Based on “Why is AI hard and Physics simple?” [2104.00008](#) &

The Principles of Deep Learning Theory w/ Yaida and Hanin, [2106.10165](#),
to be published by Cambridge University Press in 2022.

Initialization

The simulation is such that [one] generally perceives the sum of many billions of elementary processes simultaneously, so that the leveling law of large numbers completely obscures the real nature of the individual processes.

John von Neumann

Thanks to substantial investments into computer technology, modern **artificial intelligence** (AI) systems can now come equipped with many billions of elementary components.

Initialization

The simulation is such that [one] generally perceives the sum of many billions of elementary processes simultaneously, so that the leveling law of large numbers completely obscures the real nature of the individual processes.

John von Neumann

Thanks to substantial investments into computer technology, modern **artificial intelligence** (AI) systems can now come equipped with many billions of elementary components.

- ▶ Behind much of this success is **deep learning**: deep learning uses artificial **neural networks** as an underlying model for AI.

Initialization: Artificial Intelligence

Some functions are easily described in terms of elementary operations $\{+, -, \times, \div\}$:

Initialization: Artificial Intelligence

Some functions are easily described in terms of elementary operations $\{+, -, \times, \div\}$:

$$f(x) = x$$

Initialization: Artificial Intelligence

Some functions are easily described in terms of elementary operations $\{+, -, \times, \div\}$:

$$f(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Initialization: Artificial Intelligence

Some functions are easily described in terms of elementary operations $\{+, -, \times, \div\}$:

$$f(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

- ▶ Although there's an ∞ of terms, for many purposes it only takes a few terms to get a useful approximation.

Initialization: Artificial Intelligence

Some functions are easily described in terms of elementary operations $\{+, -, \times, \div\}$:

$$f(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

- ▶ Although there's an ∞ of terms, for many purposes it only takes a few terms to get a useful approximation.
- ▶ The description of the function only takes 1 in on the screen.*

* depending on the size of your monitor.

Initialization: Artificial Intelligence

Some functions cannot be described by $\{+, -, \times, \div\}$ on any slide:

Initialization: Artificial Intelligence

Some functions cannot be described by $\{+, -, \times, \div\}$ on any slide:

$$f(x) = \begin{cases} 1, & x = \{ \text{cat icon}, \text{cat photo}, \dots \}, \\ 0, & x \neq \{ \text{cat icon}, \text{cat photo}, \dots \}. \end{cases}$$

Initialization: Artificial Intelligence

Some functions cannot be described by $\{+, -, \times, \div\}$ on any slide:

$$f(x) = \begin{cases} 1, & x = \{ \text{🐱}, \text{🐱}, \dots \}, \\ 0, & x \neq \{ \text{🐱}, \text{🐱}, \dots \}. \end{cases}$$

- ▶ It's clear that such a function can exist – humans do it! – but unclear how to represent in terms of elementary operations.

Initialization: Artificial Intelligence

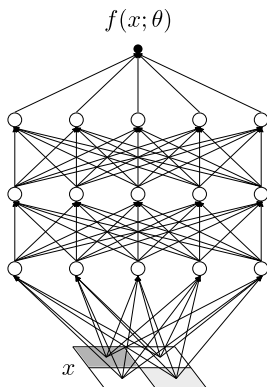
Some functions cannot be described by $\{+, -, \times, \div\}$ on any slide:

$$f(x) = \begin{cases} 1, & x = \{ \text{🐱}, \text{🐱}, \dots \}, \\ 0, & x \neq \{ \text{🐱}, \text{🐱}, \dots \}. \end{cases}$$

- ▶ It's clear that such a function can exist – humans do it! – but unclear how to represent in terms of elementary operations.
- ▶ **AI** is about functions of this sort – easy for humans to compute, but hard for humans to describe by $\{+, -, \times, \div\}$.

Initialization: Neural Networks

= A **neural network** is a recipe for computing a function built out of many computational units called **neurons**:



Neurons are then organized in parallel into **layers**, and *deep* neural networks are those composed of multiple layers in sequence.

Neural Networks Abstracted

For a moment, let's ignore all that structure and simply think of a neural network as a parameterized function,

$$f(x; \theta),$$

where x is the input to the function and θ is a vector of a large number of **parameters** controlling the shape of the function.

Neural Networks Abstracted: Function Approximation

For such a function to be useful, we need to tune the high-dimensional parameter vector θ :

Neural Networks Abstracted: Function Approximation

For such a function to be useful, we need to tune the high-dimensional parameter vector θ :

- ▶ First, we choose an **initialization distribution** by randomly sampling the parameter vector θ from a computationally simple probability distribution,

$$p(\theta).$$

Neural Networks Abstracted: Function Approximation

For such a function to be useful, we need to tune the high-dimensional parameter vector θ :

- ▶ First, we choose an **initialization distribution** by randomly sampling the parameter vector θ from a computationally simple probability distribution,

$$p(\theta).$$

- ▶ Second, we adjust the parameter vector as $\theta \rightarrow \theta^*$, such that the resulting *network function* $f(x; \theta^*)$ is as close as possible to a desired *target function* $f(x)$:

$$f(x; \theta^*) \approx f(x).$$

This is called **function approximation**.

Neural Networks Abstracted: Training

To find these tunings θ^* , we fit the network function $f(x; \theta)$ to **training data**, consisting of many pairs of the form $(x, f(x))$ observed from the desired – but only partially observable – target function $f(x)$.

- ▶ Making these adjustments is called **training**.
- ▶ The particular procedure used to tune them is called a **learning algorithm**.

Initialization: Goals

The goal of this talk is to explain a set of **principles** that enable us to theoretically analyze *deep* neural networks of *actual relevance*. To initialize you to this task, we'll try to explain

- (i) why such a goal is even attainable in theory, and
- (ii) how we are able to get there in practice.

The Theoretical Minimum

Our goal is to understand this *trained* network function:

$$f(x; \theta^*).$$

The Theoretical Minimum

Our goal is to understand this *trained* network function:

$$f(x; \theta^*).$$

One way to see the kinds of technical problems that we'll encounter in pursuit of this goal is to *Taylor expand* our trained network function $f(x; \theta^*)$ around the initialized value of the parameters θ

$$f(x; \theta^*) = f(x; \theta) + (\theta^* - \theta) \frac{df}{d\theta} + \frac{1}{2} (\theta^* - \theta)^2 \frac{d^2f}{d\theta^2} + \dots,$$

where $f(x; \theta)$ and its derivatives on the right-hand side are all evaluated at initialized value of the parameters.

The Theoretical Minimum: Problem 1

In general, the Taylor series contains an infinite number of terms

$$f, \quad \frac{df}{d\theta}, \quad \frac{d^2f}{d\theta^2}, \quad \frac{d^3f}{d\theta^3}, \quad \frac{d^4f}{d\theta^4}, \quad \dots,$$

and in principle we need to compute them all.

The Theoretical Minimum: Problem 2

Since the parameters θ are randomly sampled from $p(\theta)$, each time we initialize our network we get a different function $f(x; \theta)$, and we need to determine the mapping:

$$p(\theta) \rightarrow p\left(f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \dots\right).$$

This means that each term f , $df/d\theta$, $d^2f/d\theta^2$, \dots , in the Taylor expansion is really a *random function* of the input x , and this joint distribution will have intricate statistical dependencies.

The Theoretical Minimum: Problem 3

The learned value of the parameters, θ^* , is the result of a complicated training process. In general, θ^* is not unique and can depend on *everything*:

$$\theta^* \equiv [\theta^*] \left(\theta, f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \dots; \text{learning algorithm; training data} \right).$$

Determining an *analytical* expression for θ^* must take “*everything*” into account.

Goal, restated

If we could solve all three of these problems, then we'd have a *distribution* over trained network functions

$$p(f^*) \equiv p\left(f(x; \theta^*) \mid \text{learning algorithm; training data}\right),$$

now conditioned in a simple way on the learning algorithm and the data we used for training.

Goal, restated

If we could solve all three of these problems, then we'd have a *distribution* over trained network functions

$$p(f^*) \equiv p\left(f(x; \theta^*) \mid \text{learning algorithm; training data}\right),$$

now conditioned in a simple way on the learning algorithm and the data we used for training.

The development of a method for the analytical computation of $p(f^*)$ would let us *understand* AI systems and then let us use that knowledge to *improve* them.



“There ain’t no such thing as a free lunch.” (TANSTaaFL)

[Heinlein, Wolpert/Macready]

Why is AI (naively) hard?

- ▶ No “best” AI system when you average over all possible training examples and tasks.
- ▶ No matter how much we improve our tools for understanding, these improvements can do no better than random.


Some “examples”



Some “examples”



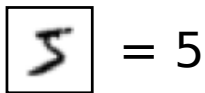
What do humans do?

 = ?

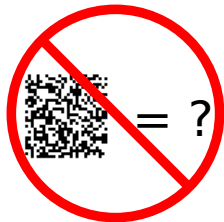
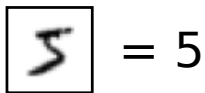
What do humans do?

$$\boxed{5} = 5$$

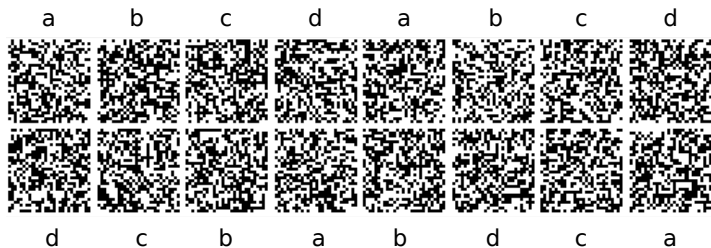
What do humans do?



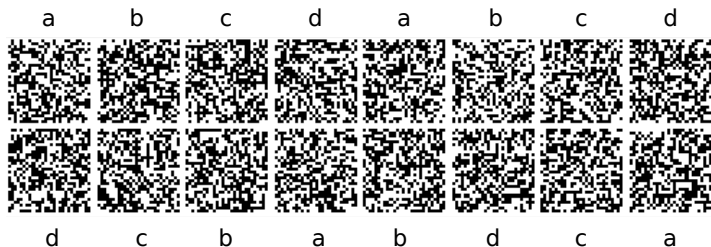
What do humans do?



What don't humans do...



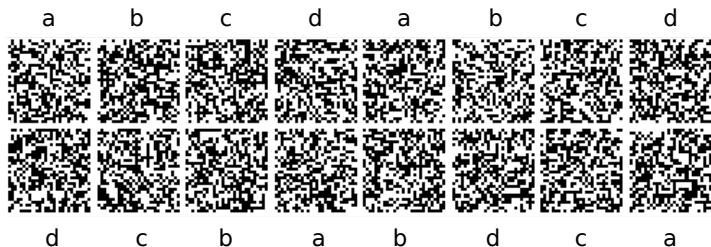
What don't humans do...



...but Neural Networks can!

[Zhang/Bengio/Hardt/Recht/Vinyals]

What don't humans do...



...but Neural Networks can!

[Zhang/Bengio/Hardt/Recht/Vinyals]

This is why (understanding how) AI (works) is hard.

Why can we learn physics?

The reason the laws of physics are even learnable at all is because the models we used to describe the universe are particularly simple models within the frameworks we used to enumerate the possible theories of physics.

$$\mathcal{L}_{SM} = \mathcal{L}_{Dirac} + \mathcal{L}_{mass} + \mathcal{L}_{gauge} + \mathcal{L}_{gf} + \mathcal{L}_{Higgs}$$

$$\mathcal{L}_{Dirac} = i\bar{e}_L^i \not{\partial} e_L^i + i\bar{\nu}_L^i \not{\partial} \nu_L^i + i\bar{e}_R^i \not{\partial} e_R^i + i\bar{u}_L^i \not{\partial} u_L^i + i\bar{d}_L^i \not{\partial} d_L^i + i\bar{u}_R^i \not{\partial} u_R^i + i\bar{d}_R^i \not{\partial} d_R^i$$

$$\mathcal{L}_{mass} = -v \left(\lambda_e^i \bar{e}_L^i e_R^i + \lambda_\nu^i \bar{\nu}_L^i \nu_R^i + \lambda_d^i \bar{d}_L^i d_R^i + \text{h.c.} \right) - M_W^2 W_\mu^+ W^{-\mu} - \frac{M_Z^2}{2 \cos^2 \theta_W} Z_\mu Z^\mu$$

$$\mathcal{L}_{gauge} = -\frac{1}{4} (G_{\mu\nu}^a)^2 - \frac{1}{2} W_{\mu\nu}^+ W^{-\mu\nu} - \frac{1}{4} Z_{\mu\nu} Z^{\mu\nu} - \frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \mathcal{L}_{WZA}$$

$$\mathcal{L}_{gf} = -g_3 A_\mu^a J_{(3)}^{\mu a} - g_2 \left(W_\mu^+ J_{W^+}^\mu + W_\mu^- J_{W^-}^\mu + Z_\mu J_Z^\mu - e A_\mu J_A^\mu \right)$$

$$\mathcal{L}_{Higgs} = \frac{1}{2} \partial_\mu h \partial^\mu h - \mu^2 h^2 - 6\lambda v h^3 - 6\lambda h^4$$

$$+ g_2 M_W h W_\mu^+ W^{-\mu} + \sqrt{g_1^2 + g_2^2} M_Z h Z_\mu Z^\mu + \frac{1}{2} g_2^2 h^2 W_\mu^+ W^{-\mu} + \frac{1}{2} (g_1^2 + g_2^2) h^2 Z_\mu Z^\mu$$

$$- h \left(\lambda_e^i \bar{e}_L^i e_R^i + \lambda_\nu^i \bar{\nu}_L^i \nu_R^i + \lambda_d^i \bar{d}_L^i d_R^i + \text{h.c.} \right)$$

$$\mathcal{L}_{WZA} = ig_2 \cos \theta_W \left[\left(W_\mu^- W_\nu^+ - W_\nu^- W_\mu^+ \right) \partial^\mu Z^\nu + W_{\mu\nu}^+ W^{-\mu} Z^\nu - W_{\mu\nu}^- W^{+\mu} Z^\nu \right]$$

$$+ ie \left[\left(W_\mu^- W_\nu^+ - W_\nu^- W_\mu^+ \right) \partial^\mu A^\nu + W_{\mu\nu}^+ W^{-\mu} A^\nu - W_{\mu\nu}^- W^{+\mu} A^\nu \right]$$

$$+ g_2^2 \cos^2 \theta_W \left(W_\mu^+ W_\nu^- Z^\mu Z^\nu - W_\mu^+ W^{-\mu} Z_\nu Z^\nu \right)$$

$$+ g_2^2 \left(W_\mu^+ W_\nu^- A^\mu A^\nu - W_\mu^+ W^{-\mu} A_\nu A^\nu \right)$$

$$+ g_2 e \cos \theta_W \left[W_\mu^+ W_\nu^- \left(Z^\mu A^\nu + Z^\nu A^\mu \right) - 2W_\mu^+ W^{-\mu} Z_\nu A^\nu \right]$$

$$+ g_2^2 \left(W_\mu^+ W_\nu^- \right) \left(W^{+\mu} W^{-\nu} - W^{+\nu} W^{-\mu} \right)$$

$$G_{\mu\nu}^a \equiv \partial_\mu A_\nu^a - \partial_\nu A_\mu^a - g_3 f^{abc} A_\mu^b A_\nu^c, \quad W_{\mu\nu}^\pm \equiv \partial_\mu W_\nu^\pm - \partial_\nu W_\mu^\pm, \quad Z_{\mu\nu} \equiv \partial_\mu Z_\nu - \partial_\nu Z_\mu, \quad F_{\mu\nu} \equiv \partial_\mu A_\nu - \partial_\nu A_\mu$$

$$\mathcal{L}_{SM} = \mathcal{L}_{Dirac} + \mathcal{L}_{mass} + \mathcal{L}_{gauge} + \mathcal{L}_{gf} + \mathcal{L}_{Higgs}$$

$$\mathcal{L}_{Dirac} = i\bar{e}_L^i \not{\partial} e_L^i + i\bar{\nu}_L^i \not{\partial} \nu_L^i + i\bar{e}_R^i \not{\partial} e_R^i + i\bar{u}_L^i \not{\partial} u_L^i + i\bar{d}_L^i \not{\partial} d_L^i + i\bar{u}_R^i \not{\partial} u_R^i + i\bar{d}_R^i \not{\partial} d_R^i$$

$$\mathcal{L}_{mass} = -v \left(\lambda_e^i \bar{e}_L^i e_R^i + \lambda_u^i \bar{u}_L^i u_R^i + \lambda_d^i \bar{d}_L^i d_R^i + \text{h.c.} \right) - M_W^2 W_\mu^+ W^{-\mu} - \frac{M_W^2}{2 \cos^2 \theta_W} Z_\mu Z^\mu$$

$$\mathcal{L}_{gauge} = -\frac{1}{4} (G_{\mu\nu}^a)^2 - \frac{1}{2} W_{\mu\nu}^+ W^{-\mu\nu} - \frac{1}{4} Z_{\mu\nu} Z^{\mu\nu} - \frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \mathcal{L}_{WZA}$$

$$\mathcal{L}_{gf} = -g_3 A_\mu^a J_{(3)}^{\mu a} - g_2 \left(W_\mu^+ J_{W^+}^\mu + W_\mu^- J_{W^-}^\mu + Z_\mu J_Z^\mu - e A_\mu J_A^\mu \right)$$

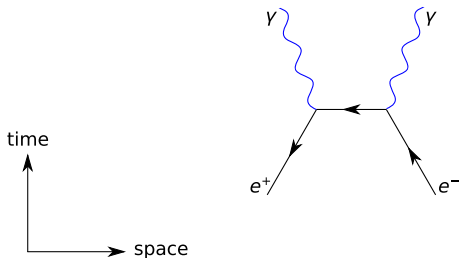
$$\begin{aligned} \mathcal{L}_{Higgs} = & \frac{1}{2} \partial_\mu h \partial^\mu h - \mu^2 h^2 - 6\lambda v h^3 - 6\lambda h^4 \\ & + g_2 M_W h W_\mu^+ W^{-\mu} + \sqrt{g_1^2 + g_2^2} M_Z h Z_\mu Z^\mu + \frac{1}{2} g_2^2 h^2 W_\mu^+ W^{-\mu} + \frac{1}{2} (g_1^2 + g_2^2) h^2 Z_\mu Z^\mu \\ & - h \left(\lambda_e^i \bar{e}_L^i e_R^i + \lambda_u^i \bar{u}_L^i u_R^i + \lambda_d^i \bar{d}_L^i d_R^i + \text{h.c.} \right) \end{aligned}$$

$$\begin{aligned} \mathcal{L}_{WZA} = & ig_2 \cos \theta_W \left[\left(W_\mu^- W_\nu^+ - W_\nu^- W_\mu^+ \right) \partial^\mu Z^\nu + W_{\mu\nu}^+ W^{-\mu} Z^\nu - W_{\mu\nu}^- W^{+\mu} Z^\nu \right] \\ & + ie \left[\left(W_\mu^- W_\nu^+ - W_\nu^- W_\mu^+ \right) \partial^\mu A^\nu + W_{\mu\nu}^+ W^{-\mu} A^\nu - W_{\mu\nu}^- W^{+\mu} A^\nu \right] \\ & + g_2^2 \cos^2 \theta_W \left(W_\mu^+ W_\nu^- Z^\mu Z^\nu - W_\mu^+ W^{-\mu} Z_\nu Z^\nu \right) \\ & + g_2^2 \left(W_\mu^+ W_\nu^- A^\mu A^\nu - W_\mu^+ W^{-\mu} A_\nu A^\nu \right) \\ & + g_2 e \cos \theta_W \left[W_\mu^+ W_\nu^- \left(Z^\mu A^\nu + Z^\nu A^\mu \right) - 2W_\mu^+ W^{-\mu} Z_\nu A^\nu \right] \\ & + g_2^2 \left(W_\mu^+ W_\nu^- \right) \left(W^{+\mu} W^{-\nu} - W^{+\nu} W^{-\mu} \right) \end{aligned}$$

$$G_{\mu\nu}^a \equiv \partial_\mu A_\nu^a - \partial_\nu A_\mu^a - g_3 f^{abc} A_\mu^b A_\nu^c, \quad W_{\mu\nu}^\pm \equiv \partial_\mu W_\nu^\pm - \partial_\nu W_\mu^\pm, \quad Z_{\mu\nu} \equiv \partial_\mu Z_\nu - \partial_\nu Z_\mu, \quad F_{\mu\nu} \equiv \partial_\mu A_\nu - \partial_\nu A_\mu$$

Physics is simple?

Actually quite simple considering it has the ability to describe almost every *experiment* that we could perform.



Thus, *useful* physical theories are **sparse**: we can organize according to a *local* action, where interactions happen between an $O(1)$ number of particles at a point in spacetime.

Problems in Understanding Deep Learning

- ▶ **Problem 1**, we need to compute an infinite number of terms:

$$f, \quad \frac{df}{d\theta}, \quad \frac{d^2f}{d\theta^2}, \quad \frac{d^3f}{d\theta^3}, \quad \frac{d^4f}{d\theta^4}, \quad \dots$$

- ▶ **Problem 2**, each time we initialize our network we get a different function $f(x; \theta)$, and we need to determine the map:

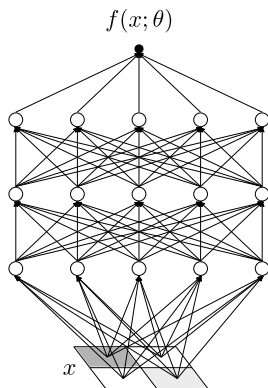
$$p(\theta) \rightarrow p\left(f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \dots\right).$$

- ▶ **Problem 3**, The learned value of the parameters, θ^* , is the result of a complicated training process:

$$\theta^* \equiv [\theta^*] \left(\theta, f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \dots; \text{learning algorithm; training data} \right).$$

Fine, Structure

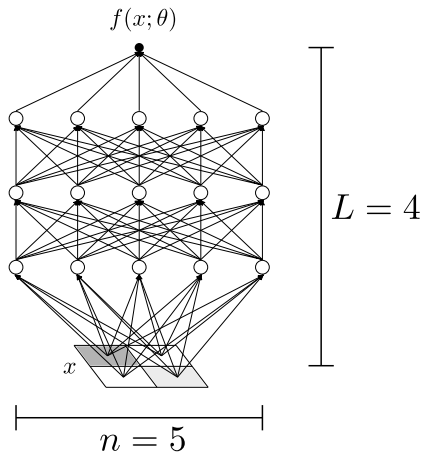
Solving our three problems for a general parameterized function $f(x; \theta)$ is not tractable. However, we only care about the functions that are deep neural networks:



To make progress we will have to make use of the particular **structure** of neural-network function.

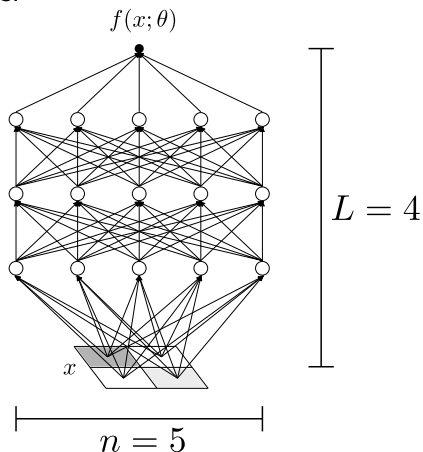
Fine, Structure

Two essential aspects of a neural network *architecture* are its **width**, n , and its **depth**, L .



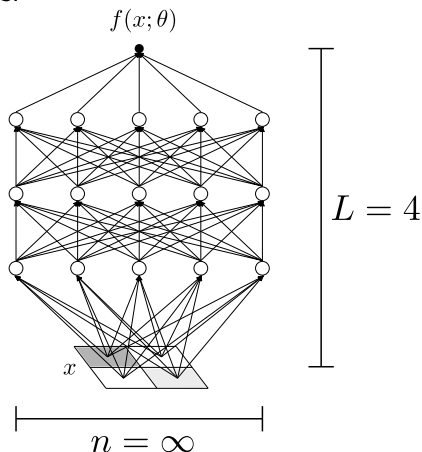
A Principle of Sparsity

There are often simplifications to be found in the limit of a large number of components.



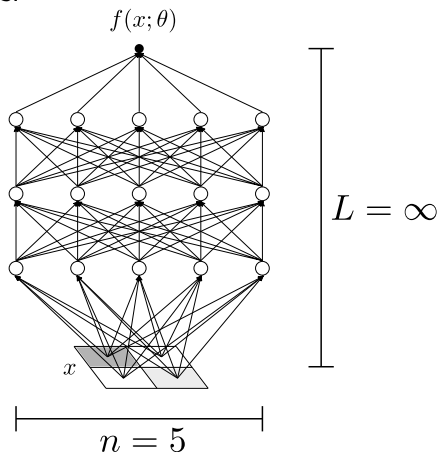
A Principle of Sparsity

There are often simplifications to be found in the limit of a large number of components.



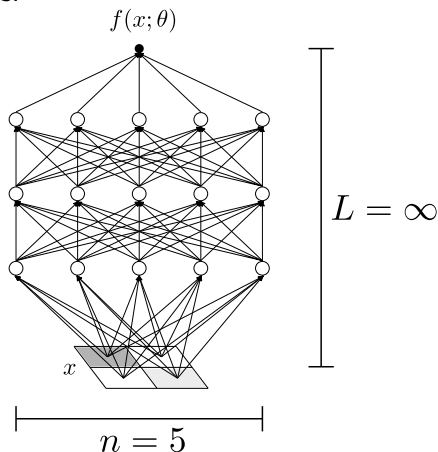
A Principle of Sparsity

There are often simplifications to be found in the limit of a large number of components.



A Principle of Sparsity

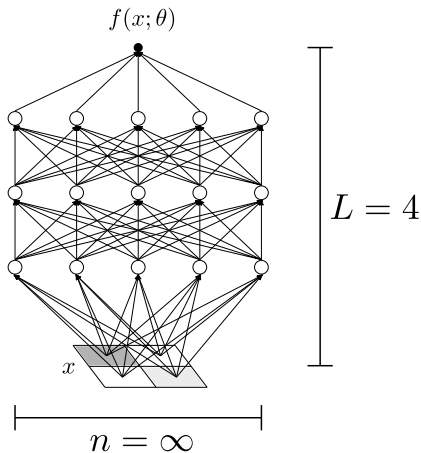
There are often simplifications to be found in the limit of a large number of components.



It's not enough to consider any massive macroscopic system, and taking the right limit often requires some care.

A Principle of Sparsity

In this case, the $n = \infty$ will make everything really simple, while the $L = \infty$ will be hopelessly complicated and useless in practice.



The Infinite-Width Limit

Let's begin by formally taking the limit

$$\lim_{n \rightarrow \infty} p(f^*),$$

and studying an *idealized* neural network in this limit.

[Neal, Lee/Bahri/..., Matthews/..., Jacot/..., ...]

The Infinite-Width Limit

Let's begin by formally taking the limit

$$\lim_{n \rightarrow \infty} p(f^*),$$

and studying an *idealized* neural network in this limit.

- ▶ This is known as the **infinite-width limit** of the network, and as a strict limit it's rather *unphysical* for a network: obviously you cannot directly program a function to have an infinite number of components on a finite computer.

[Neal, Lee/Bahri/..., Matthews/..., Jacot/..., ...]

The Infinite-Width Limit

Let's begin by formally taking the limit

$$\lim_{n \rightarrow \infty} p(f^*),$$

and studying an *idealized* neural network in this limit.

- ▶ This is known as the **infinite-width limit** of the network, and as a strict limit it's rather *unphysical* for a network: obviously you cannot directly program a function to have an infinite number of components on a finite computer.
- ▶ However, this extreme limit does massively simplify the distribution over trained networks $p(f^*)$, rendering each of our three problems completely benign.

[Neal, Lee/Bahri/..., Matthews/..., Jacot/..., ...]

Simplicity at Infinite Width

- ▶ Addressing **Problem 1**, higher derivative terms will effectively vanish, and we only need to keep track of two terms:

$$f, \frac{df}{d\theta}.$$

Simplicity at Infinite Width

- ▶ Addressing **Problem 1**, higher derivative terms will effectively vanish, and we only need to keep track of two terms:

$$f, \quad \frac{df}{d\theta}.$$

- ▶ Addressing **Problem 2**, the distributions of these random functions will be independent,

$$\lim_{n \rightarrow \infty} p\left(f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \dots\right) = p(f) p\left(\frac{df}{d\theta}\right),$$

with each marginal distribution factor taking a simple form.

Simplicity at Infinite Width

- ▶ Addressing **Problem 1**, higher derivative terms will effectively vanish, and we only need to keep track of two terms:

$$f, \quad \frac{df}{d\theta}.$$

- ▶ Addressing **Problem 2**, the distributions of these random functions will be independent,

$$\lim_{n \rightarrow \infty} p\left(f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \dots\right) = p(f) p\left(\frac{df}{d\theta}\right),$$

with each marginal distribution factor taking a simple form.

- ▶ Addressing **Problem 3**, the training dynamics become *linear* and *independent* of the details of the learning algorithm, giving θ^* in a *closed form* analytical solution:

$$\lim_{n \rightarrow \infty} \theta^* = [\theta^*]\left(\theta, f, \frac{df}{d\theta}; \text{training data}\right).$$

Simplicity at Infinite-Width

These simplifications are the consequence of a **principle of sparsity**, and the fully-trained distribution,

$$\lim_{n \rightarrow \infty} p(f^*),$$

is a simple **Gaussian distribution** with a nonzero mean.

Too Much Simplicity at Infinite-Width

The formal infinite-width limit, $n \rightarrow \infty$, leads to a poor model of *deep* neural networks in practice:

Too Much Simplicity at Infinite-Width

The formal infinite-width limit, $n \rightarrow \infty$, leads to a poor model of *deep* neural networks in practice:

- ▶ The distribution over *real* trained networks *does* depend on the properties of the learning algorithm used to train them.

Too Much Simplicity at Infinite-Width

The formal infinite-width limit, $n \rightarrow \infty$, leads to a poor model of *deep* neural networks in practice:

- ▶ The distribution over *real* trained networks *does* depend on the properties of the learning algorithm used to train them.
- ▶ Infinite-width networks don't have **representation learning**: for any input x , its transformations in the hidden layers, $z^{(1)}$, $z^{(2)}$, \dots , $z^{(L-1)}$, will remain unchanged from initialization.

Aside: Representation Learning

In the typical discussion of representation learning, we start with the fine-grained representation of an input such as an image in terms of its pixels:

$$x = \text{cat}.$$

For a classification task, a network might output a *coarse-grained* description of that image:

$$f(x) = \text{cat}.$$

In between, the signals at the hidden-layer neurons form intermediate **representations**.

Too Much Simplicity at Infinite-Width

The formal infinite-width limit, $n \rightarrow \infty$, leads to a poor model of *deep* neural networks in practice:

- ▶ The distribution over *real* trained networks *does* depend on the properties of the learning algorithm used to train them.
- ▶ Infinite-width networks don't have **representation learning**: for any input x , its transformations in the hidden layers, $z^{(1)}$, $z^{(2)}$, \dots , $z^{(L-1)}$, will remain unchanged from initialization.

The *central limiting* problem is that the input of an infinite number of signals is such that the leveling law of large numbers completely obscures the subtle correlations between neurons that get amplified over the course of training for representation learning.

Interacting Neurons

We'll need to find a way to restore and then study the **interactions** between neurons that are present in realistic *finite-width* networks.

Interacting Neurons

We'll need to find a way to restore and then study the **interactions** between neurons that are present in realistic *finite-width* networks.

To do so, we can use **perturbation theory** and study deep learning using a **$1/n$ expansion**, treating the inverse layer width, $\epsilon \equiv 1/n$, as our small parameter of expansion:

$$p(f^*) \equiv \left\{ \lim_{n \rightarrow \infty} p(f^*) \right\} + \frac{p^{\{1\}}(f^*)}{n} + \frac{p^{\{2\}}(f^*)}{n^2} + \dots,$$

Interacting Neurons

We'll need to find a way to restore and then study the **interactions** between neurons that are present in realistic *finite-width* networks.

To do so, we can use **perturbation theory** and study deep learning using a **$1/n$ expansion**, treating the inverse layer width, $\epsilon \equiv 1/n$, as our small parameter of expansion:

$$p(f^*) \equiv \left\{ \lim_{n \rightarrow \infty} p(f^*) \right\} + \frac{p^{\{1\}}(f^*)}{n} + O\left(\frac{1}{n^2}\right).$$

Near-Simplicity at Finite Width

- ▶ Addressing **Problem 1**, most derivatives will contribute as $O(1/n^2)$ or smaller, so we only need to keep track of 4 terms:

$$f, \quad \frac{df}{d\theta}, \quad \frac{d^2f}{d\theta^2}, \quad \frac{d^3f}{d\theta^3}.$$

Near-Simplicity at Finite Width

- ▶ Addressing **Problem 1**, most derivatives will contribute as $O(1/n^2)$ or smaller, so we only need to keep track of 4 terms:

$$f, \quad \frac{df}{d\theta}, \quad \frac{d^2f}{d\theta^2}, \quad \frac{d^3f}{d\theta^3}.$$

- ▶ Addressing **Problem 2**, the distribution of these random functions at initialization will be *nearly* simple at order $1/n$:

$$p\left(f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \frac{d^3f}{d\theta^3}\right),$$

Near-Simplicity at Finite Width

- ▶ Addressing **Problem 1**, most derivatives will contribute as $O(1/n^2)$ or smaller, so we only need to keep track of 4 terms:

$$f, \quad \frac{df}{d\theta}, \quad \frac{d^2f}{d\theta^2}, \quad \frac{d^3f}{d\theta^3}.$$

- ▶ Addressing **Problem 2**, the distribution of these random functions at initialization will be *nearly* simple at order $1/n$:

$$p\left(f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \frac{d^3f}{d\theta^3}\right),$$

- ▶ Addressing **Problem 3**, the *nonlinear* training dynamics can be tamed with *dynamical perturbation theory*, giving θ^* in a *closed form* analytical solution:

$$\theta^* = [\theta^*] \left(\theta, f, \frac{df}{d\theta}, \frac{d^2f}{d\theta^2}, \frac{d^3f}{d\theta^3}; \text{learning algorithm; training data} \right).$$

Near-Simplicity at Finite Width

These near-simplifications are a further consequence of the **principle of sparsity**, and our *dual effective theory* description of the fully-trained distribution at order $1/n$,

$$p(f^*) \equiv \left\{ \lim_{n \rightarrow \infty} p(f^*) \right\} + \frac{p^{\{1\}}(f^*)}{n} + O\left(\frac{1}{n^2}\right),$$

will be a **nearly-Gaussian distribution**.

The Role of Depth as the Effective Theory Cutoff

An important byproduct of the analysis is a careful understanding of the *deep* in deep learning. Defining the **aspect ratio**

$$r \equiv L/n,$$

we can recast our understanding of *infinite-width vs. finite-width* and *shallow vs. deep*:

The Role of Depth as the Effective Theory Cutoff

An important byproduct of the analysis is a careful understanding of the *deep* in deep learning. Defining the **aspect ratio**

$$r \equiv L/n,$$

we can recast our understanding of *infinite-width vs. finite-width* and *shallow vs. deep*:

- ▶ In the strict limit $r \rightarrow 0$, the interactions between neurons turn off: the infinite-width limit is actually a decent description, but these networks are **not really deep**, as their relative depth is zero: $L/n = 0$.

The Role of Depth as the Effective Theory Cutoff

An important byproduct of the analysis is a careful understanding of the *deep* in deep learning. Defining the **aspect ratio**

$$r \equiv L/n,$$

we can recast our understanding of *infinite-width vs. finite-width* and *shallow vs. deep*:

- ▶ In the regime $0 < r \ll 1$, there are nontrivial interactions between neurons: the finite-width effective theory truncated at order $1/n$ gives an accurate accounting $p(f^*)$. These networks are **effectively deep**.

The Role of Depth as the Effective Theory Cutoff

An important byproduct of the analysis is a careful understanding of the *deep* in deep learning. Defining the **aspect ratio**

$$r \equiv L/n,$$

we can recast our understanding of *infinite-width vs. finite-width* and *shallow vs. deep*:

- ▶ In the regime $r \gg 1$, the neurons are strongly coupled: networks will behave chaotically, and there is no effective description due to large fluctuations from instantiation to instantiation. These networks are **overly deep**.

The Role of Depth as the Effective Theory Cutoff

An important byproduct of the analysis is a careful understanding of the *deep* in deep learning. Defining the **aspect ratio**

$$r \equiv L/n,$$

we can recast our understanding of *infinite-width vs. finite-width* and *shallow vs. deep*:

- ▶ In the regime $r \gg 1$, the neurons are strongly coupled: networks will behave chaotically, and there is no effective description due to large fluctuations from instantiation to instantiation. These networks are **overly deep**.

Networks of *practical use* have small aspect ratios: $r \sim r^* \ll 1$.

The Principle of Sparsity and Model Complexity

Consider a fixed combined training and test dataset of size $N_{\mathcal{D}}$:

- ▶ For the *infinite-width* **Gaussian distribution**, we only need

$$n_{\text{out}} N_{\mathcal{A}} + \left[\frac{N_{\mathcal{D}}(N_{\mathcal{D}} + 1)}{2} \right] + \left[\frac{N_{\mathcal{D}}(N_{\mathcal{D}} + 1)}{2} \right] = O(N_{\mathcal{D}}^2)$$

numbers in order to completely specify the distribution.

The Principle of Sparsity and Model Complexity

Consider a fixed combined training and test dataset of size N_D :

- ▶ For the *finite-width* **nearly-Gaussian distribution** with $0 < r \ll 1$, we will instead need $O(N_D^4)$ numbers, with the counting dominated by the finite-width tensors.

The Principle of Sparsity and Model Complexity

Consider a fixed combined training and test dataset of size N_D :

- ▶ For an accuracy $O(L^k/n^k)$, a *macroscopic description*

$$p(z(\infty)) = \sum_{m=0}^k \frac{p^{\{m\}}(z(\infty))}{n^m} + O\left(\frac{L^{k+1}}{n^{k+1}}\right),$$

will need $O(N_D^{2k})$ numbers in general.

The Principle of Sparsity and Model Complexity

Consider a fixed combined training and test dataset of size N_D :

- ▶ For an accuracy $O(L^k/n^k)$, a *macroscopic description*

$$p(z(\infty)) = \sum_{m=0}^k \frac{p^{\{m\}}(z(\infty))}{n^m} + O\left(\frac{L^{k+1}}{n^{k+1}}\right),$$

will need $O(N_D^{2k})$ numbers in general.

- ▶ The **1/n expansion** gives a sequence of effective theories with increasing accuracy at the cost of increasing complexity.

The Principle of Sparsity and Model Complexity

As r increases, we'll need to include more of these higher-order terms, making our *effective theory* description more complex:

The Principle of Sparsity and Model Complexity

As r increases, we'll need to include more of these higher-order terms, making our *effective theory* description more complex:

- ▶ In the strict limit $r \rightarrow 0$, the *sparse* $O(N_D^2)$ **Gaussian** description of the infinite-width limit will be accurate.

The Principle of Sparsity and Model Complexity

As r increases, we'll need to include more of these higher-order terms, making our *effective theory* description more complex:

- ▶ In the strict limit $r \rightarrow 0$, the *sparse* $O(N_{\mathcal{D}}^2)$ **Gaussian** description of the infinite-width limit will be accurate.
- ▶ In the regime $0 < r \sim r^* \ll 1$, the *nearly-sparse* $O(N_{\mathcal{D}}^4)$ **nearly-Gaussian** description of the finite-width effective theory truncated at order $1/n$ will be accurate.

The Principle of Sparsity and Model Complexity

As r increases, we'll need to include more of these higher-order terms, making our *effective theory* description more complex:

- ▶ In the strict limit $r \rightarrow 0$, the *sparse* $O(N_D^2)$ **Gaussian** description of the infinite-width limit will be accurate.
- ▶ In the regime $0 < r \sim r^* \ll 1$, the *nearly-sparse* $O(N_D^4)$ **nearly-Gaussian** description of the finite-width effective theory truncated at order $1/n$ will be accurate.
- ▶ For larger r , a more generic $O(N_D^{2k})$ **non-Gaussian** description would in principle be necessary.

Conclusion

The practical success of deep learning in with large numbers of parameters is really telling us that useful theories of neural networks should be **sparse** – but not too sparse – so that they are also **deep**.

Conclusion

The practical success of deep learning in with large numbers of parameters is really telling us that useful theories of neural networks should be **sparse** – but not too sparse – so that they are also **deep**.

Thank You!