Mean Field Theory and its application to deep learning

Mudit Pandey (mp5099@nyu.edu)
Mimee Xu (mimee@nyu.edu)

May 2019

Abstract

While a class of models and techniques in deep learning has achieved empirical success, the interactions of their underlying mechanisms are under-explored. Oftentimes, researchers who seek clarity in the science of deep learning adapt theoretic tools developed in other scientific fields. In statistical mechanics, an approximation technique for complex, interactive systems called Mean Field Theory (MFT) is now broadly applied to explain why deep learning works. In particular, MFT highlights the dynamical system similarities between a deep network's parameters and interacting particles. By adapting MFT to study a network's signal propagation, theorists can explore behaviors of very large, general neural networks that experimental work alone can't cover [Saul et al., 1996]. Most recently, various papers on this topic have been gaining popularity [Hanin, 2018, Karakida et al., 2018, Kawamoto et al., 2018, Pretorius et al., 2018, Schoenholz et al., 2016].

To machine learning practitioners, however, the conference papers on the topic may be too short to be accessible. This paper serves as an introduction to mean field formalism as applied to study properties of neural networks. Readers who wish to understand the subfield should find here tools, definitions, illustrations that clarify the motivation and assumptions used in current works.

We first introduce mean field theory as in its historical context of physics, with an example on the Ising model. Then we connect MFT to machine learn through parallels drawn in variational inference. Finally, we summarize the setups of MFT modelling in recent advances to help understand neural networks' expessivity [Poole et al., 2016], ResNets [Yang and Schoenholz, 2017b], Convolutional neural networks [Xiao et al., 2018], and most recently batch normalization [Yang et al., 2019] and gradient descent dynamics. In summary, we show that application of MFT touches very popular architectures and empirical techniques in today's deep learning era.

Contents

1 MFT In Statistical Physics

Strong assumptions notwithstanding, some simplified models can explain real-world observations without resorting to very difficult mathematics. Originally, Mean Field Theory stemmed from such models physicists used to explain macroscopic phenomenons.

The **Ising Model** proposes that spins of particles arrange themselves on a chain in one-dimensional space, or a lattice in higher dimensions. Furthermore, each particle takes on a binary state: up or down. In addition, every spin's (stochastic) properties are only dependent on its nearest neighbors: two on a line, four in a plane, and $2d$ in d dimensions.

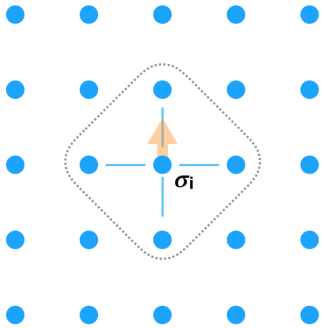

Figure 1: In a 2D Ising Model, each particle is influenced by its 4 nearest neighbors.

This section sets up a physical system under the Ising Model, and introduces the usage of mean field approximation in deriving **phase transition**, applied to magnetization, largely based on Statistical Mechanics lecture [Susskind, 2013].

1.1 Isolated Magnet In a Heat Bath

Consider a stylized field of magnets, each pointing either up or down. The system is in equilibrium at temperature T. In addition, to make the notation clear, associate with each site a spin $\sigma = +1$ be up, -1 if down. This allows us to write down the energy function, such that the magnet takes on different energies at different orientations.

Recall that the normalization constant in the Boltzmann distribution, denoted by Z, is also called the **partition function**, with varying temperatures [1]; oftentimes in physics, Z is used to calculate macroscopic properties of the system, such as energy, pressure, magnetization, and entropy.

Suppose there is only one tiny magnet in a heat bath. Suppose μ is the magnetic moment, B is the magnetic field, then we can write $E = \mu B\sigma = -J\sigma$

[1] The **Boltzmann distribution** for one particle and a system of particles at maximal

for numeric factor J. We obtain its partition function, summed over all 2 configurations.

$$Z = \sum_{\text{configs}} e^{\beta J \sigma} = e^{(+1)\beta J} + e^{(-1)\beta J} = e^{\beta J} + e^{-\beta J} = 2\cosh \beta J$$

The whole system of N individual magnets has a factor partition function, which is the individual spin's raised to the N-th power, since each one is independent. That allows us to the take its logarithm and get a sum. We calculate the expected value of the thermodynamic energy, which is the negative inverse times the derivative of the partition function with respect to the inverse temperature.

$$E_{\text{one spin}} = -\frac{1}{Z}\frac{\delta Z}{\delta \beta} = -\frac{J\sinh(\beta J)}{\cosh(\beta J)} = -J\tanh \beta J$$

The average σ, the expected value of the spin, is therefore $\langle \sigma \rangle_{\text{average}} = \tanh \beta J$. The probability of the spin pointing up is very high when J is positive, β is large. That is, low temperature corresponds to the spin pointing up in the setup.

1.2 Ising Model

The Ising Model for our purpose is a simple model with many little configurations (spins) pointing up or down, with each site spaced equally far apart. In high dimensions, they form a lattice. The Ising Model states that the energy contribution is only from a particle itself and its nearest neighbors. For historical reasons, many problems in physics and statistics are framed this way. Moving on from an isolated magnet, consider a 1-D Ising model of an array of magnets. The energy function i.e. the Hamiltonian of the whole set is $E = -J\sum \sigma_i \sigma_{i+1}$. Its partition function is thus over all configurations $Z = \sum e^{-J\beta \sum \sigma_i \sigma_{i+1}}$.

1.2.1 Correlation function

Given that a particular magnet is up, what is the conditional probability that n links down the magnet is also up? Alternatively, we can frame the question as finding the average of the product of the spins at two different locations.

---

entropy

$$p_i \propto e^{-\frac{E_i}{T}} \text{ and } \quad p_{\text{config}_i} = \left(e^{-\frac{E_i}{T}}\right) / \sum_{\text{configs}} e^{-\frac{E_j}{T}}$$

where E_i is the energy associated with the state of interest. As per convention, $\beta = 1/T$, and we will use β for inverse temperature throughout this paper.

Figure 2: A 1-D Ising Model of ferromagnetic σ's.

Intuitively, the correlation diminishes as the distance increases, yet there is a possibility that the bias propagates, as if the system has some lineage of memory. To diagnose, focus on the links between the spins rather than the spins themselves. For a finite chain with its first spin pointing up, we will sum up the partition function in that first up added by first down.

Consider a change of variable trick $\mu_i \to \sigma_i\sigma_{i+1}$. Since σ_1 is known, the product $\mu_1 = \sigma_1\sigma_2$ is sufficient for deriving σ_2. The product μ_i describes alignment and has two possibilities, parallel or anti-parallel. Knowing the μ's, as long as you know the first spin, is equally good as knowing all the σ's. Now this is useful because the energy is just made up of these bond variables:

$$E = -J\sum \sigma_i\sigma_{i+1} = -J\sum_{i}^{n-1}\mu_i$$

Notice that there are 1 fewer bonds than all the particles. In our transformation of the energy makeup, the individual bonds have no relationship among them as far as the equation goes. The information is retained, also, since it is as good to know the μ's as it is to know σ's. So now you can substitute the sum over spins in Z, the partition function, with the sum of the values of the bond:

$$Z = 2\sum_{\mu} e^{-\sum_i J\beta\mu_i}$$

The factor 2 arises from the possibilities for the first spin, which we condition on. Recall that $N = ||i|| = ||\mu|| + 1$. The Boltzmann factor here is a product, assuming N spins so $\|i\| = N$. As such, we factorize the partition function into that of one spin's energy raised to the $\|\mu\|$-th power and obtain $Z = (2\cosh\beta J)^{N-1}$, a familiar-looking partition function we saw in Section 1.1.

Figure 3: A chain of n dependent σ's can be seen as a set of $n-1$ independent ferromagnets $\vec{\mu}$.

Despite the partition function, the physical meaning is different from N-1 isolated magnets, because μ is the product of the neighboring spins. Now con-

sider $\langle\mu\rangle = \mathbb{E}(\mu)$, the average correlation between immediate pairs of neighbors. Through the same calculus

$$\langle\mu\rangle = \langle\sigma_i\sigma_{i+1}\rangle = \tanh\beta J$$

where positive J biases a positive value. In the physical system, this indicates a tendency towards alignment, in a way that is better than even chance. We write the correlation between i and $i+n$ spins as their product:

$$\langle\sigma_i\sigma_{i+n}\rangle = \langle\sigma_i\sigma_{i+1}\sigma_{i+1}\sigma_{i+2}\cdots\sigma_{i+n}\rangle = \langle\mu_1\mu_2\cdots\mu_{n-1}\rangle$$

If we assert the independence of the μ's in this formulation, and substitute the average.

$$\langle\sigma_i\sigma_{i+n}\rangle \approx \langle\mu\rangle^{n-1} = (\tanh\beta J)^{n-1}$$

Given this being higher than the uniformly random expectation of $\frac{1}{2}$, we see a long range *memory* in magnetization: everything will be biased to go up if the first one is up.

1.2.2 Factorization Approximation in MFT

We discuss the transformations used: $\sigma_i\sigma_{i+1} \to \mu_i$ and $\mu_i \to \langle\mu\rangle$. The variable substitution $\sigma_i\sigma_{i+1} \to \mu_i$ induces a **duality** i.e. an equivalence between a theory of spins of nearest neighbor and another system with spins are independent to each other. Originally a bond substitution, μ takes on the form of independent spins in a brand-new system made of bonds. Though the bond model seems physically removed, its abstraction greatly simplifies mathematics.

The mean-field model only exists in the dual: the mean energy $E_{MF} = -J\sum_i^{n-1}\langle\mu\rangle$ is exact from actual energy $E = -J\sum_i^{n-1}\mu_i$. In so far as we care about the summation, each of μ_i's energy contributes independently, thus justifying the the *Mean Field Approximation*. In computing the correlation statistic, MFT becomes a factorization approximation, stated as

$$\prod_i \mu_i \approx \prod_i \langle\mu\rangle = \langle\mu\rangle^{n-1}$$

Though all MFT has its origin in taking the average, for simplicity, we refer to this specific approximation strategy as "factorization".

1.3 High Dimension Ising Model and Mean Field Approximation

2D Ising model gets significantly harder. In fact, Ising himself got it wrong. In higher dimensions, we can see that the fluctuation to be small, because the number of neighbors on a lattice grows $2d$ with dimension d. Given this insight, we can see the Ising model as a small subsystemm plus a heat bath. Focusing

on one spin of a small magnet, assumed to be at equilibrium with the rest of the environment, with the partition function

$$Z_{\text{one spin}} = \sum e^{+\beta J\sigma} = e^{\beta J} + e^{-\beta J} = 2\cosh \beta J$$
$$Z_{\text{whole system}} = (e^{\beta J} + e^{-\beta J})^k = 2\cosh(\beta J)^k$$

Recall the expected energy per spin

$$E_{\text{one spin}} = -\frac{1}{z}\frac{\delta z}{\delta \beta} = -\frac{J\sinh(\beta J)}{\cosh(\beta J)} = -J\tanh \beta J.$$

So in expectation, $\langle\sigma\rangle_{\text{average}} = \tanh \beta J$. This sets up for mean field approximation: In approximating the bias, we assume a high dimensionality Ising where the average fluctuation is a lot smaller than the average bias. There, using the average is a pretty good approximation for individual behavior if the number of neighbors, $2d$, is large. For one spin, $E_i = -j\sigma_i \sum_{j \text{ neighboring } i} \sigma_j$. For simplicity, let

$$\langle\sigma_i\rangle = \bar{\bar{\sigma}} \text{ and } \langle\sigma_{\text{neighbors to } i}\rangle \approx \langle\sigma_{\text{all spins}}\rangle = \bar{\sigma}$$

Then the sum is just the number of neighbors, $2d$, times the average of neighbor spins, $E = -J\sigma_i(2d\bar{\sigma})$.

This is a particular spin sitting in the bath i.e. field of all the others, evincing a **mean field formulation**. The field here denotes the field experienced by i, sitting in the field of all the others. Now we do the partition function as usual, except we substitute a constant field $J \to 2dJ\bar{\sigma}$. The same calculus gives us

$$\bar{\bar{\sigma}} = \tanh[(2\beta dJ)\bar{\sigma}].$$

Similarly, if they have an average of $sig\bar{m}a$, then $\bar{\bar{\sigma}} = sig\bar{m}a$. This gives an equation that applies to all temperature:

$$\bar{\sigma} = \tanh[(2\beta dJ)\bar{\sigma}]$$
$$\text{Let } y = (2\beta dJ)\bar{\sigma}, \text{ then } \frac{y}{2\beta dJ} = \tanh y$$
$$\text{Recall } \beta \text{ being inverse temperature, so } T\frac{y}{2dJ} = \tanh y.$$

We plot both sides at different temperatures, as shown in Figure 4. The only possible solution is $y = 0$, at very high T, so the average of $sigma$ is 0, as expected. As we lower the temperature, the slope of this curve on the left-hand-side decreases to the point of 1, so we are tangent to the tanh(.), when $T = 2dJ$. This is a critical point and signals that our approximation shows a **phase transition**!

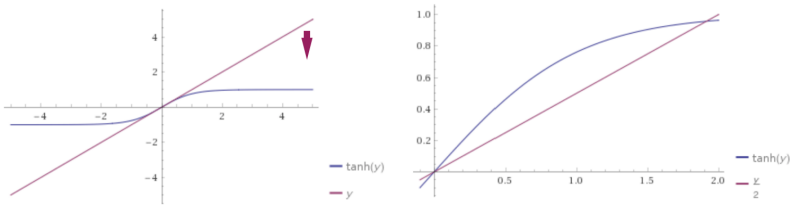

Figure 4: **phase transition** When $T = 2dJ, T\frac{y}{2dJ} = y$, we see a hyperbolic tangent curve $\tanh y$ only intersecting with line y at the origin $y = 0$, as shown on the left plot. As we lower the temperature as when $T = dJ, T\frac{y}{2dJ} = y/2$, we see an additional critical point where they intersect, as shown on the right plot. The point corresponds to a critical temperature where phase transition happens.

1.3.1 Self-averaging MFT

At high dimensions, we ignore the thermal fluctuations of particles and derive an approximation for the probabilistic dependency between spins of magnets. This resulting phase transition does not exist in the 1D Ising Model, and cannot be tested at infinite dimension. Because the spin is assumed to be no different from all others, this MFT flavor is sometimes called **self-consistent field approximation**. We call this substitution strategy "self-averaging".

As demonstrated, it takes advantage of large scale systems for what physicists call *partial understanding*: to explain qualitative phenomenon in empirical observations. This lends MFT naturally to machine learning.

1.4 Conclusion

The Ising Model is a simplified generating mechanism in statistical physics that encapsulates complex behavior. In 1D, magnets influence each other with decaying correlation at long distance. By seeing each spin as a mean of the field of spins it is in, we demonstrate two MFT flavors, factorization and self-averaging, and derive the phenomenon of magnetization.

The mean field approach conditionally simplifies the Ising Model's mathematics. In studying phase transition, the MFT relies on high dimensionality, which dominates the criterion for the derivation. Essentially, a particular situation was picked as a way to make a spin have a lot of nearest neighbors to apply the mean field.

Without mean field assumptions, a close form would be very hard to compute. Unsurprisingly, extending mean field approaches is demonstratively powerful in high dimensional statistics. The next section introduces Variational Mean Field methods in statistics.

2 MFT in statistics

One of the major problems in statistics is to approximate hard to compute probability distributions for a system. This is especially important in Bayesian Inference and statistical machine learning, where a joint probability distribution over unobserved and observed data is required. The distribution maybe easy to compute for some small models. However, for large complex models, it is not at all easy. Exact inference on such models is not practically possible. We look at a class of approximation techniques called variational methods that attempt to approximate the probability distributions as best as possible. In the section that follows, we briefly introduce the problem of inference and how a mean-field assumption helps in efficiently computing the required estimate of probability distribution. The sections is largely based on Blei et al. [2017].

2.1 Variational Inference

In variational inference, we model our system as a collection of random variables where some variables are hidden(Z) and some are visible. By hidden, we mean that the values of these variables are not observed directly. It follows naturally, that the random variables for which the values are observed are called visible variables. The visible variables are also called evidence variables (or data) due to the same reason. In a Bayesian setting, the hidden variables help govern the distribution over the observed variables. The influence can be modeled as a graph shown below:

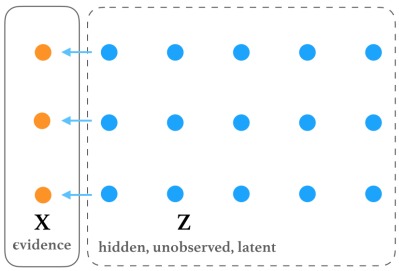

Figure 5: A graphical representation of a hidden variables influencing a visible variables

The edge drawn in the graph above relates variables Z and X as a conditional distribution $P(X|Z)$. We now look at a general problem formulation. Consider hidden variables $Z = \{Z_1, Z_2, \cdots, Z_m\}$ and visible variables $X = \{X_1, X_2, \cdots, X_n\}$. Inference in a Bayesian setting usually involves, calculating the posterior over hidden variables i.e probability of hidden variable conditioned on observed data. By Bayes theorem, we have

$$P(Z|X) = \frac{P(X, Z)}{P(X)}$$

The denominator $P(X)$ is the marginal probability of the observations also called the likelihood of evidence. This is obtained by marginalizing the hidden variables in the joint distribution $P(X, Z)$. This is simply the sum of the join distribution over all possible configurations of the hidden variables. Thus the likelihood of the evidence P(X) is represented as

$$P(X) = \sum_{Z} P(X, Z)$$

In order to calculate the posterior over hidden variables, we require the likelihood of evidence. This is all well and good for small models, but for large complex models, the number of hidden variables tends to be very large. In this case, the sum in the likelihood of evidence becomes very hard to compute since it involves summation of a very large number of terms. The number of terms in the sum increases in an exponential manner with respect to the number of hidden variables. It is now clear that the sum is intractable for large number of hidden variables and some sort of approximation is required for $P(Z|X)$.

But how do we go about approximating $P(Z|X)$? One method for approximate inference is a sampling based method called Monte Carlo Markov Chain (MCMC) sampling. MCMC algorithms are very popular and find applications in a wide number of problems. One key feature of such methods is that they provide guarantees (asymptotically) of producing exact samples from the target density (the density that had to be approximated). This makes them ideal for scenarios that require precise samples. However, MCMC tends to be computationally expensive and does not scale well (in terms of computation time) for large and complicated models. For such cases, variational inference acts as a faster alternative. Even though variational inference, does not provide guarantees similar to MCMC, they give reasonable results. Thus they are suitable for scenarios where there is huge amount of data and a fast exploration through models is needed.

In variational inference, we introduce a family of distributions $\mathcal{Q}$ over the hidden variables Z. Each member Q(Z) in the family $\mathcal{Q}$ is a potential approximation to the posterior over the hidden variables. To find the best approximation, we resort to the Kullback-Liebler divergence between our posterior $P(Z|X)$ and a member of $\mathcal{Q}$. The Q(Z) that is closest in KL divergence with our posterior is the best approximation.

$$Q^*(Z) = \underset{Q(Z)\in\mathcal{Q}}{\arg\min}\, D_{\mathrm{KL}}\left(\, Q(Z) || \, P(Z|X)\right)$$

We have now converted our inference problem into an optimization problem! However, we are not yet done. The KL divergence cannot be directly computed since we require the expectation over the logarithm of the posterior which we are trying to approximate $P(Z|X)$ in the first place.

$$D_{\mathrm{KL}}\left(\, Q(Z) || \, P(Z|X)\right) = \mathrm{E}[\log Q(Z)] - \mathrm{E}[\log P(Z|X)]$$

Here, the expectation is with respect to the distribution Q(Z). We expand $P(Z|X)$ in the KL divergence above and find that, instead of minimizing the

KL divergence directly, we can minimize a new objective which is the difference between the expectation of the logarithm of distribution Q(Z) and the expectation of the logarithm of the joint distribution of observed and hidden variables. The log-likelihood of evidence P(X) does not depend on the distribution Q(Z) and therefore remains a constant.

$$D_{\mathrm{KL}}\left(\, Q(Z) || \, P(Z|X)\right) = \mathrm{E}[\log Q(Z)] - \mathrm{E}[\log P(X,Z)] + \log P(X)$$

This new objective is called the ELBO or the evidence lower bound.

$$\mathrm{ELBO(Q)} = \mathrm{E}[\log P(X,Z)] - \mathrm{E}[\log Q(Z)]$$

The reason it is called so is because it lower bounds the log-likelihood of evidence. That is,

$$\log P(X) \geq \mathrm{ELBO(Q)}$$

In its historical context, the ELBO was derived using the properties of KL divergence and the Jensen's equality. The derivation can be found in the appendix 8.1. We observe that the ELBO is essentially the negative KL divergence plus some constant. Thus, our problem breaks down into maximizing the ELBO.

With the stage for variational inference set and done, we now introduce the **mean field assumption**. We note that the optimization of the ELBO objective depends on the variational family of distributions $\mathcal{Q}$. Thus, the complexity of the optimization directly depends on the complexity of the family. So, in the spirit of mean field methods as first used in statistical physics where complex systems are approximated by simpler independent systems, we choose a family $\mathcal{Q}$ which is simple. We assume that the hidden variables are independent and that they factorize over the mean field distribution i.e each hidden variable Z_i with its own distribution Q_i.

$$Q(Z) \approx \prod_{i=1}^{m} Q_i(Z_i)$$

The family of distributions that is chosen for Q_i is usually the exponential family. It turns out that this family along with independence assumptions simplify the optimization of the objective. We will look into this into a little more detail later. We apply the variational mean field method to an example, specifically the high dimensional Ising model. The full derivation can be found in the Appendix 8.2

We have seen the variational inference converts the original inference problem into an optimization problem that maximizes the ELBO. The posterior is then approximated with a family of *mean field distributions* i.e. factorized models that assume sparse interaction terms [2].

Superior in computability, MF sacrifices interaction terms between groups of latent variables. The independence assumption welcomes many optimization

[2]For factorization justification, see exponential-family-conditional models, a.k.a conditionally conjugate models where latent variables are independent.[Blei et al., 2017]

methods, such as **coordinate gradient ascent**: at every iteration, some coordinates are held fixed while others are optimized. Effectively, the coordinates allow the ELBO to climb to a local optima. If an exponential family is used for the mean field distribution, updates in the coordinate ascent algorithm simplify resulting in faster computations.

2.2 Limitations

Despite the performance boosts of mean-field variational inference in terms of computation costs, the method suffers from limitations. The main limitation of mean-field inference is that it explicitly ignores correlations between latent variables when making the independence assumption. As a result, despite capturing the marginal probability distribution of latent variables, it fails to capture their correlation. Moreover, the marginal variances of the approximation underrepresent the true posterior. This behaviour can be explained by the form of KL divergence used in mean field variational inference. The KL divergence penalizes mass placed in variational distribution $Q(Z)$ when the true posterior $P(Z|X)$ is small. This basically means, that $Q(Z)$ is forced to be small whenever $P(Z|X)$ is small. The above behaviour can be see as 'zero-forcing' since $P(Z|X) = 0$ implies $Q(Z) = 0$ [Minka, 2005]. This zero-forcing behaviour emphasizes on modelling the tails of the distribution rather than bulk which results in underestimating the variance of the true posterior. Another consequence of this is that mean-field variational inference does not approximate well when the true posterior is a multi-modal distribution. It tends to model the mode with highest probability mass rather than the entire distribution.

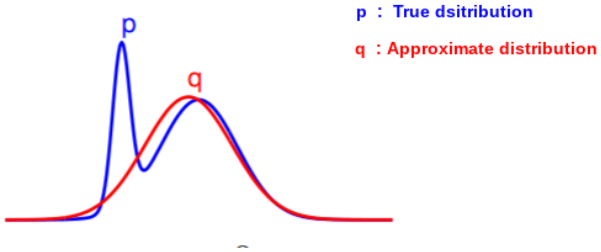

Figure 6: The approximate distribution q models the tails of the true distribution rather than the mass [Minka, 2005]

Furthermore, Wainwright and Jordan [2008] show that optimization problem becomes increasingly non-convex as more and more possible dependencies are broken by the mean-field variational distribution. Stated differently, if the variational distribution contains more structure, certain local optima do not exist. This means that as a result of the simplifying mean field assumption, the optimization can climb to an erroneous local optima.

3 When is Mean Field Good?

In the previous sections, it appears that MFT aggressively simplifies complex mathematics without sacrificing fidelity. Though conceptually intriguing and empirically successful, it is not yet clear how such a mean field strategy can be formally employed. In particular, why should the particles be self-consistent in a large network, how will the observed transition in the mean field generalize, when are interaction terms "safe" to ignore, and why are factorized distributions a good choice in optimization. We leave the bulk of the research to the reader by summarizing some key ideas.

Suitability: Like the Ising Model where MFT originated, the probabilistic graphical models of concern are generally large. When the statistics of inter-correlation at long distances decay, the maximal terms dominate, provided that the energy is modeled as a summation dependent on interactive strengths. This justifies the first-order methods in method field analyses. In addition, part of the model's apparent success came from studying behaviors near extremal points, such as zero temperatures or very high dimensions. This strategy pushes down the influence of other terms and ensure the dominance of the averaging effects.

MFT's empirical success may be due to appropriate mean-field assumptions, because the decoupling of variables is not too far from what one sees in the true posterior due to natural clustering: some particles are closer to each other than to random particles, thus allowing the variation in parameter to capture the diversity in observation [Xing et al., 2002].

Tests: MFT trade-off is reflected in the difference between the observation and the approximation. In physics, this is done via the Gibbs-Bogoliubov-Feynman inequality; in variational statistics, it is done via testing the distance from the ELBO, since the mean-field models' marginal variances are lower bounds on the variance of real data. Specifically, TAP correction [J. Thouless et al., 1977] and second order approximations [Kappen and Wiegerinck, 2000] are commonly used in conjunction with MFT to improve the quality of mean-field results.

Heuristics: For a heuristics-based procedure, we summarize several strategies in using MFT. The setup of the problem needs to have elements of stochasticity, the number of particles need to reach a scale, so that a self-averaging behavior could be observed. In approximate inference, factorization is the most prominent in variational methods.

4 Setting Up A Mean Field Theory of Deep Learning

To recap what we learned so far, in 1.2.2 and 2.1, we reviewed factorization MFT that is widely used in variational inference. The variational mean field analysis connects statistical physics and Boltzmann machine approximations for neural networks. Going beyond the variational inference setting, and perhaps closer to the roots of physics, the rest of this survey focuses on a new line of

deep learning theory work that places mean field analysis squarely at its center, which we coin as a kind of *phenomenological deep learning*.

While large-scale neural networks work amazingly well, many phenomenons they exist remain elusive. In untangling various effects, empirical work alone is often insufficient, partly due to its scale limit in coverage across data sets, parameters, and architectural features. A phenomenology is thus desirable. This section lays down a shared mean field formalism used by [Chizat and Bach, 2018, Mei et al., 2018, Yang and Schoenholz, 2017a,b, Yang et al., 2019] to study the efficacy of deep learning itself.

4.1 A Phenomenological View

Just like statistical mechanics' view of the natural world, researchers believe that deep learning exhibits general phenomenons that can be studied, independent of its microscopic details. This belief is subtly different from the scientific method where new experimentation is used to test hypotheses; here, we seek a *description* of anticipated behavior of a system through model, ex post facto.

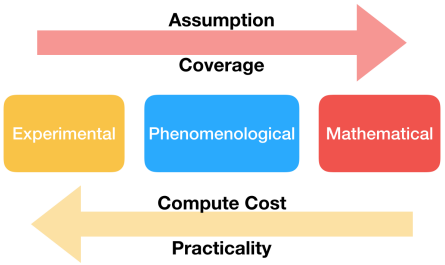

Figure 7: Methods that discover general insights in large scale learning make trade-offs between incurring expensive computation and making strict theoretic assumptions.

Mean field theory becomes a natural tool in this pursuit for generalizing insights, such as optimization behaviors at limits [Mei et al., 2018], why neural networks generalize [Jacot et al., 2018], where gradient explosion and vanishing happen [Yang and Schoenholz, 2017b], and the efficacy of BatchNorm in stabilizing training [Ioffe and Szegedy, 2015, Yang et al., 2019].

4.2 A Gaussian Processes View

A neural network with infinite number of neurons is approximated as Gaussian Processes, similar to that of **Neural Tagent Kernels** [Jacot et al., 2018, Jaehoon Lee, 2018] or deep signal propagation [Poole et al., 2016].

Common Setup

x	Network input
D	Input dimension
N	Number of Neurons
$\{W_{ij}\}$	Weight matrix
$\{b_j\}$	Bias vector
σ_w^2	Variance of weights
σ_b^2	Variance of biases
k	Number of SGD run

Common assumptions

1. Either "Over-parametrization Regime" with $N > n^c$ for constant c where the weights don't move very far, or "Mean Field Regime" where there are only a few optimization steps

$$N \gtrapprox D\,, k \in O(1) \,\text{and}\, D \lessapprox k \leq n$$

2. The hidden layer has infinite width
3. Random weights initialization
4. Fully-connected

For non-linearity ϕ acting on input x, and feeding into affine transformation to output z, we write

$$z_i(x) = b_i + \sum_{j=1}^{N} W_{ij}\phi(x) \tag{1}$$

By Central Limit Theorem, as $N \to \infty$, z will be Gaussian distributed in the limit if ϕ_i and ϕ_j are independent. This independence is easy to show in the case of assumptions 1, 2, 3, and 4: Because of random initialization, the weights $\mathbf{W}$ and biases $\vec{b}$ are drawn i.i.d. from Gaussians of variances $\frac{\sigma_w^2}{\text{layer width}}$ and σ_b^2, respectively. As a result, there is no statistical dependence between different neurons pre-activation.

We thus replace each pre-activation with Gaussian random variables pre-activation, whose coupling can be ignored. This leads to a formulation of mean field theory. Regardless of the number of layers, a feed-forward infinite-width network can be approximated with Gaussian Processes if randomly initialized. Note that is just the most common formulation, and some of these assumptions could be relaxed..

At every layer, we can take the output of the previous layer, and define a recurrence relationship *layer wise*. This effectively places a dynamical system view by studying the changes of inputs from layer to layer over the space convariance matrices. As the width goes to infinity, this Gaussian process between input and output can be seen as deterministic. This feedforward dual is studied more extensively in Section 5.1.

4.3 A Dynamical System View of Gradient Descent

If we see the training of neural network as a dynamical system, then the state update is just a step of gradient descent. This helps the study of deep learning training dynamics through the lens of optimization. Studying the training dynamics at some notion of limits leads to results on trainability, gradient explosion, and landscape at convergence.

Gradient descent updates are constructed in the form of a recurrence relation $\theta_i = \theta_{i-1} - \nabla_{\theta_{i-1}} L^{[i-1]}$. Since gradient descent is an Euler method in numerical analysis, prior work on numerical methods can be leveraged. With very small learning rate, this can be seen as an ordinary differential equation through a **gradient flow** formulation.

4.3.1 Gradient Flow

For our purpose, a gradient flow is roughly defined as

$$u' = F(u) \text{ where } \exists \epsilon \in \mathcal{R} \text{ s.t. } \int_\Omega u_t F(u) = \epsilon \frac{\mathrm{d}}{\mathrm{d}t} \psi(u(t))$$

Naturally, we choose ϵ to be our learning rate, or a scaled factor of our step size. Without loss of generality (since using constant gradient step is the same as scaling loss function by a constant), we can write

$$\dot{\theta_i} = -\nabla_{\theta_i} L(\theta_1, \cdots, \theta_n)$$

Specifically, stochastic gradient descent in this network 4.2 can be approximated with a **Wasserstein Gradient Flow**. Wassterstein gradient flow is defined on the space of probabilities with Wasserstein distance. This framework is flexible with natural equivalences to a partial differential equations definition.

$$\partial_t \rho - \nabla \cdot (\rho v) = 0 \,\text{where}\, v = \nabla[\delta F / \delta \rho]$$

4.3.2 Approximating Stochastic Gradient Descent

Studying the movement of the weights is hard when we only have a few steps of gradient descent, because the gradients can be large, and the space can be nonconvex. Instead, one can construct a gradient flow with respect to a Wasserstein metric on a probability measure for the kind of functions the network approximates.

The idea is that in the suitable scaling limit, the reduction of population loss is captured by ρ; on the other hand, ρ is the solution to a partial differential equation. As a result, SGD can be approximated using Wasserstein gradient flow. This is useful in studying macroscopic phenomenons, because the scale limit can be then derived from the PDE formulation, often by examining the dependency of the critical points with respect to the variables. This flavor is very similar to the duality view in physics, as seen in Section 1.2.2, which often accompanies the application of MFT. This duality is, however, not exact, and thus requires further examination to hold.

Notably in [Mei et al., 2018], the cost function in the space of $(\mathscr{P}, W_2)$ is viewed as a gradient flow. The results and approach are further summarized in Section 6.

4.4 Notable Challenges

As an exercise in the formalism, we introduce two examples of mean-field centric approaches to explaining deep learning phenomenons, roughly broken down by architecture and optimization. Because the works introduced in each one are exceedingly similar in their assumptions, they face similar limitations inherit to the framework, such as not being able to discover interesting higher order behaviors.

An easy criticism of *feature MFT* such as those by Yang et al. [2019] is that it is a super fancy technique to study the explosion of gradients. A couple of strong assumptions have not been generally relaxed: infinite width, convex activation, and in the case for the optimization papers, the number of layers of the network. In addition, since the mean field theory is incomplete, with each added feature, new mathematics are needed, which makes the MFT non-trivial to extend. Finally, as a problem-solving strategy MFT *can* only yields results on the phenomenological level, and can thus not clarify narratives on microscopic behavior, as seen in 2.2 and 3.

5 Neural Network Features

A group of researchers extend this mean field analysis to different network features of the neural networks, such as residual networks [Yang and Schoenholz, 2017b], convolutional layers [Xiao et al., 2018], and batch normalization [Yang et al., 2019]. The motivation is clear: since these are features seen in deep learning that has become empirically irreplaceable, it is assumed to be fruitful to understand why they work, whether they break down, in order to come up with better theories and training schemes.

5.1 Feed-forward networks

The paper by Poole et al. [2016] is concerned with signal propagation in feed forward neural networks. Combining concepts of mean field theory and Riemann

geometry, they give a theoretical formulation that proves that the expressivity of neural networks increases exponentially with depth. We briefly discuss their approach in this section.

Signal propagation in deep neural networks can be understood by studying the geometry of simple manifolds in the input layer $x^{[0]}$. Essentially, we would like to know how the geometry is modified as the manifold propagates through numerous layers. For the simplest case of a single vector, one can track its 'length' i.e. the squared norm, represented as:

$$q^l = \frac{1}{N_l} \sum_{i=1}^{N_l} (z_i^l)^2$$

where q^l is the normalized squared norm of pre-activations at layer l. Under the mean field assumption, we can obtain a iterative map for q^l by propagating Gaussians through layers.

$$q^l = \mathcal{V}(q^{l-1}|\sigma_w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z \phi \left(\sqrt{q^{l-1}} z\right)^2 + \sigma_b^2 \tag{2}$$

where $\mathcal{D}z$ is the standard Gaussian measure. The function $\mathcal{V}$ is a length-map that predicts how the length of an input changes as it propagates through the network. For a monotonic non-linear activation (assumed to be sigmoidal in the paper), the length map is a monotonically increasing concave function. We can say that a fixed point $q^*(\sigma_w, \sigma_b)$ has been reached when the length q^l does not change with respect to the length in the previous layer q^{l-1} i.e $\frac{q^l}{q^{l-1}} = 1$. In other words, the fixed points are obtained by observing the intersections of the length map with the unity line.

5.1.1 Transient chaos

Consider a slightly complex scenario where we study the layer-wise propagation of two inputs[3] to a layer. The geometry of the two inputs as they propagate through the network is captured by a 2×2 matrix of inner product

$$q_{12}^l = \frac{1}{N_l} \sum_{i=1}^{N_l} z^l{}_i(x^1) z^l{}_i(x^2)$$

Similar to (2), we can derive a correlation map for q_{12}^l

$$q_{12}^l = \mathcal{C}(c_{12}^{l-1}, q_{11}^{l-1}, q_{22}^{l-1}|\sigma_w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z_1 \mathcal{D}z_2 \phi(u_1)\phi(u_2) + \sigma_b^2 \tag{3}$$

where $c_{12}^l = q_{12}^l (q_{11}^l q_{22}^l)^{-1/2}$ is the correlation coefficient[4]. Together (2) and (3), form a theoretical prediction for the geometry of a pair of points[5] as they

[3] These can be from the input layer $x^{[0]}$ or pre-activations in intermediate layers

[4] Also corresponds to the cosine similarity between pre-activations

[5] Points in the input manifold for a layer or in other words two inputs

propagate through a neural network. Analyzing the the equations in the σ_w and σ_b plane reveals and interesting order to chaos transition for the system. The relation between two points can be tracked by the correlation coefficient c_{12}^l. Using the fixed point $q^*(\sigma_w, \sigma_b)$ for the length of a single vector, we calculate an iterative correlation coefficient map ($\mathcal{C}$-map) as

$$c_{12}^l = \frac{1}{q^*}\mathcal{C}(c_{12}^{l-1}, q^*, q^*|\sigma_w, \sigma_b)$$

The $\mathcal{C}$-map has a fixed point at 1 ($c^* = 1$). However, the stability of the fixed point depends on the slope at 1

$$\chi_1 \equiv \left.\frac{\partial c_{12}^l}{\partial c_{12}^{l-1}}\right|_{c=1} = \sigma_w^2 \int \mathcal{D}z \left[\phi^{'}(\sqrt{q^*}z)\right]^2$$

The equation $\chi_1(\sigma_w, \sigma_b) = 1$ yields a phase transition boundary in the σ_w and σ_b plane, separating it into a chaotic and an ordered phase. For a fixed σ_b, we get a phase transition as σ_w increases. For small σ_w, $c = 1$ is a stable fixed point ($\chi_1 < 1$). This corresponds to the ordered phase where two points converge to each other as they propagate through the network. In the region with large σ_w, $c = 1$ is no longer a stable fixed point. The weights dominate the biases and de-correlate the inputs. This corresponds to the chaotic region ($\chi_1 > 1$) where nearby points separate as they propagate through the network eventually becoming orthogonal (maximum de-correlation; $c^* = 0$). In the intermediate region near the phase transition boundary ($\chi_1 \to 1$), also called the **edge of chaos**[6], an equal competition exists between the weights (which along with the non-linearity, decorrelate the inputs) and biases (which correlate the inputs) exist leading to a finite fixed point c^* ($0 < c^* < 1$). Thus χ_1, can be considered as a multiplicative stretch factor.

Poole et al. [2016] extend the above results for a complete manifold in input space and prove that in the chaotic phase of the neural network, the simple input manifold de-correlates and becomes increasingly (exponentially) complex with depth. This implies that a deep network is able to compute exponentially complex functions over simple low dimensional manifolds.

5.1.2 Gradients

While Poole et al. [2016] investigates the nature of the signal as it propagates through the network in a forward dynamics, Schoenholz et al. [2016] study the nature of gradients drawing in a duality between forward and backward propagation.

Consider the backpropagtion of a given loss $\mathcal{E}$,

$$\frac{\partial \mathcal{E}}{\partial W_{ij}^l} = \delta_i^l \phi(z_j^{l-1}) \qquad \delta_i^l = \frac{\partial \mathcal{E}}{\partial z_i^l}$$

[6]Term coined by Yang and Schoenholz [2017b]

Within mean field theory, it is clear that the scale of fluctuations of the gradient of weights in a layer will be proportional to the second moment of δ_l^i [Schoenholz et al., 2016]. The authors note that unlike the pre-activations in forward propagation, δ_i^l will **not** be a Gaussian distribution even for $N \to \infty$. However, we can obtain a recurrence relation for $\tilde{q}_{aa}^l = \mathrm{E}\left[(\delta_i^l)^2\right]$ under the assumption that the weights used during backpropagation are drawn **independently** from the weights used in forward propagation.

$$\tilde{q}_{aa}^l = \tilde{q}_{aa}^{l+1}\chi_1$$

Note: The equation above also contains a factor proportional to N_{l+1}/N_l which is unity for our setup. Since χ_1 depends only on the asymptotic c^*, the above equation has an exponential solution resulting in a phase transition boundary similar to what was discussed in the previous section, but for gradients. When in the ordered phase ($\chi_1 < 1$), the gradients are expected to vanish over a depth whereas in the chaotic phase ($\chi_1 > 1$), gradients are expected to explode. On the edge of chaos, namely region $\chi_1 \to 1$, the gradients should be stable regardless of depth.

The results in 5.1.1 and 5.1.2 lead to a trainability vs expressivity trade-off for fully-connected neural networks. While deep networks operating in the chaotic phase tend to be more expressive (with expressivity increasing with depth up to a fixed point), the gradients for such networks tend to explode with increase in depth. For networks on the edge of chaos, extremely deep neural networks can be trained. This is because information about the inputs is able to propagate forward and information on gradients are also able to propagate backwards through the deep network.

5.2 Resnets

In the previous sections, we have seen that the exponential forward dynamics of sigmoidal feed forward neural networks causes a rapid collapse of the input geometry[7]. A similar scenario exists for the backward dynamics causing gradients to drastically vanish or explode. Yang and Schoenholz [2017b] build on previous works ([Poole et al., 2016],[Schoenholz et al., 2016]) and show that by adding skip connections, the network adopts a sub-exponential or polynomial forward and backward dynamic (depending on the non-linearity). This slower convergence to the fixed points allows residual networks to 'hover' over the edge of chaos longer. This provides some theoretical justification as to why ResNets with a large number of layers work well in practice.

The main results in Yang and Schoenholz [2017b] are:

- The forward dynamics for tanh and α-ReLU ($\alpha < 1$) is polynomial with depth.
- The backward dynamic for tanh is sub-exponential whereas the backward dynamics for α-ReLU ($\alpha < 1$) is asymptotically polynomial.

[7] The input geometry exponentially converge to the fixed point

- ReLu exhibits exponential forward and backward dynamics asymptotically. One interesting observation is that not all gradient signals exhibit exponential behaviour. The gradient norm with respect to the weights w is independent of how far the gradient has propagated (it is constant). This, however, is not the case with bias b for which it increases exponentially.

5.3 CNN

Convolutional Neural Networks have been crucial to the success of deep learning. However, most of these deep models are only trainable by employing techniques like residual connections and batch normalization. Although we have seen some justification in support of techniques like residual connections (Section 5.2), it is still unclear whether deep CNNs **necessarily** require these techniques for successful training. Xiao et al. [2018] develop a mean field theory of CNNs to investigate this issue by furthering works discussed above. One key difference in the mean field assumption for CNNs is that instead of considering the large width assumption i.e $N \to \infty$ we assume a large number of channels i.e the channels c in a filter tend to infinity.

Xiao et al. [2018] find that the mean field derivation for signal propagation in CNNs is similar to that of [Poole et al., 2016] and the stability condition is precisely the one that govern fully-connected networks (as discussed in Section 5.1.1). Moreover, the fixed point analysis for CNNs leads to the same result as in the case of feed-forward neural network. This means that for CNNs too, there exists a phase transition boundary at $\chi_1 = 1$. For $\chi_1 < 1$, $c^* = 1$ is a stable fixed point and the network exists in an ordered phase where all pixels converge to the same value. For $\chi_1 > 1$, there exists stable fixed point $c^* < 1$. This corresponds to the chaotic phase where all pixels values de-correlate.

The analysis for the backward propagation of the signal leads to the same result as the one derived in Section 5.1.2. Thus, the network must stay in the edge of chaos to ensure that gradient signals neither explode nor vanish as they back-propagate through a convolutional network.

The authors note that although the order-to-chaos phase boundaries of fully-connected and convolutional networks look identical, the underlying mean-field theories differ. A novel aspect of the convolutional theory is the existence of multiple depth scales that control signal propagation at different spatial frequencies. In the large depth limit, signals can only propagate along modes with minimal spatial structure; all other modes end up deteriorating, even at criticality.

Xiao et al. [2018] push their analysis beyond mean-field theory by incorporating dynamical isometry [Pennington et al., 2017] for CNNs. They develop a modified initialization scheme that allows for balanced propagation of signals among all frequencies. They call this scheme Delta-Orthogonal initialization. This scheme allows them to train ultra deep vanilla CNNs with no degradation in performance.

5.4 BatchNorm and Gradients

BatchNorm remain elusive in machine learning theory. On one hand, it clearly works well in practice. On the other, there is no clear theory on why it works; some theorized pre-conditioning leading to some notion of stability in training because the landscape to optimize is much smoother, and the story seems to have stopped there since [Santurkar et al., 2018]. Similar to Phase Transition in Section 1.3, Yang et al. [2019] find the limits of phenomenons of interest: at $L < 50$, gradient explosion is small because the gradients are small compared to the weights and the weights don't change much; at $L > 50$, explosion dominates W's: weight norm decreases, and from $t = 0$ to $t = 1$, gradients cross the threshold of $|W| = |\nabla(\cdot)|$. The major contributions enabled by MFT includes the mathematics to show that BatchNorm causes gradient explosion, enlarging gradient norm with every layer by 1.47, and a linear activation is suggested to minimize the explosion rate to $\frac{b-2}{b-3}$ where b denotes the batch size. This conclusion is at odds with several other theories that postulate the stability benefits of BatchNorm, suggesting that BatchNorm works through other benefits. In this way, MFT made a difficult-to-test theory more feasible to study.

6 Stochastic Gradient Descent

The literature on the theory of deep learning is rapidly expanding, mostly to cover more general networks and relaxing some of the assumptions on data initialization scheme, or from angles previously not used before. Since each paper has its own setup, it is difficult to enumerate all the variations. For the purpose of their paper, we touch on the use of mean field theory by providing its role in some proof sketches. In studying the optimization dynamic of large neural networks, mean field theory is employed to simplify the problem. A popular limit chosen is in the number of neurons per layer, and the type of resulting network is often infinitely wide, and only two layers, often fully-connected [Chizat and Bach, 2018, Jacot et al., 2018, Mei et al., 2018]. We now describe the mathematical tools associated with the most popular regime shared among recent work.

6.1 Two Layer Neural Network Converges To Global Minima

The paper by Mei et al. [2018] is concerned with coming up with a gradient flow in the population risk. By taking advantage of the law of large numbers, their mean-field formulation appears more inspired by physics than deep learning. We sketch the proof strategy used in this paper.

The setup is a "mean-field regime", where $N \gtrsim D, D \lesssim \bar{k} \leq n$. Assuming that at every time t we draw each parameter $\theta_i^0 \sim$ i.i.d. ρ_0, and that the σs of our neurons are bounded by C in L_∞-norm. Then we have the gradient with respect to δ, σ to be a random variable because the input to the layer is assumed

to be random.

$$\dot{\theta}_i = -\nabla\Psi(\theta_i)\hat{\rho}_t^{(N)})$$
$$\hat{\rho}_t^{(N)} = \frac{1}{N}\sum_{j=1}^{N}\delta_{\theta_j}(t)$$

The system describes a particle moving in the force field defined by these other particles, following a non-linear dynamic. Recall that these are i.i.d. trajectories, because θ_i^0 is drawn from ρ_0. On the other hand, we formulate a different system with $\bar{\theta}_i(t)$, which describes n independent initialization. The evolution is described in a way akin to that of a system of particles in physics:

$$\dot{\bar{\theta}}(t) = \nabla\Psi(\bar{\theta}(t), \rho_t).$$

The system of $\bar{\theta}$ is then used to relate to θ, with $\bar{\theta}_i(0) = \theta_i^0$ so that they live in the same probability space. Eventually, to show that the PDE written from gradient flow approximates SGD, it suffices to bound the distance of this approximation for some bounded function M:

$$d_{\theta,\bar{\theta}}(t) = \frac{1}{N}\sum_i |\theta_i(t) - \bar{\theta}_i(t)|^2 \leq M(N, t, D)$$

Crucial to the main result, Mei et al. [2018] writes

$$\text{SGD Dynamic can be seen as } \begin{cases} \partial_t\rho_t = \nabla_\theta(\rho_t\nabla_\theta\Psi(\theta,\rho_t)) \\ \Psi(\theta,\rho) = V(\theta) + \int V(\theta,\tilde{\theta})\rho(\delta\hat{\theta}) \end{cases}$$
$$\text{where } V(\theta) = -\mathbb{E}(y\sigma(x_i\theta)) \text{ and } U(\theta_1\theta_2) = \mathbb{E}_x\sigma(x_i\theta_1)\sigma(x_i\theta_2).$$

This PDE describes the evolution of the particle in the force field provided by the "density" of all the other particles. This strips N from the optimization at large N, showing that the optimization does not infinitely scale with the number of neurons. While a non-actionable result in empirical machine learning, this suggests that over-parametrization is only part of the story why neural nets work. Additionally, this formulation effectively reduces a $N \times D$-dimensional problem to a problem of only D dimension, and a very random process to a somewhat deterministic one.

6.2 SGD Mean Field Discussion

This mean field setup differs tremendously from those in Section 5, taking a purely phenomenological view of the training dynamic. It states that neural networks parameters have a dual of interacting particles dictated by a potential loss landscape, so that training describes an evolution of the interaction. This allows the study of gradient dynamic that is very far from initialization, a large expansion from its contemporary counterparts.

The two mean fields may merge in the future. As of now, however, the biggest weakness is that the assumptions made are extremely restrictive; an open challenge lies in not just how to limit the width of each layer, but also in how to extend this to multiple layers, mostly recently attempted by Nguyen [2019] "non-rigorously". While the scaling limits between neuron size N, number of steps k and dimension D are reasonable, the number of hidden layers staying at 1 is unacceptable. In addition, the particle descent formulation requires continuity equation, which is not rigorously shown to be convergent in realistic settings. As is, this MFT formulation is thus unlikely to offer generalizing insights to practitioners. However, this prototypical framing of particle evolution abstracts away training dynamic from network features, thus allowing for the novel application of diverse mathematics to study what makes deep learning work, as seen in [Chizat and Bach, 2018] [Rotskoff and Vanden-Eijnden, 2018].

7 Conclusion

Explaining deep learning's empirical success requires an intersection of acute observation and appropriate approximation. Mean Field Theory is a powerful technique, uniquely applied to the scale and practice of deep learning. As practitioners set out to fully understand and apply MFT, it is essential to understand the situations under which MFT is effective, the strategies to use, and the limitations where the theory is inappropriate.

This survey paper motivates the use of MFT in deep learning through the historical practice of mean-field methods in physics and statistics, with flavors of factorization and self-averaging. Its general philosophy states that the study of the phenomenon of the system can be divorced from the study of its parts, and that the parts are self-consistent. In studying why neural networks work, this abstraction is drastically different from the experimental methods that try to isolate the widgets of the most affect. At the heart of its mathematics, mean field approximation assumes some extent of independence among entities, making it a suitable theoretic for studying practical regimes of over-parametrized neural nets.

Throughout the survey, we discuss the ample restrictions in each of the MF approximations. Like all theoretical models, MFT is wrong when its assumptions deviate from practice e.g. inconsistency near critical conditions, because the fluctuations and correlations between particles are not modelled, or the details of the phenomenon studied may also be much more diverse than the averaging effects mean field theory assumes. To mitigate, higher order methods are used, such as TAP correction.

We summarize current progress on connecting MFT to deep learning, a fast-moving area of research. We introduce a specific formalism which is agnostic to the scale of data and model. This MFT has been successfully applied to study the behaviors of a variety of popular deep learning architectures. Though powerful, MFT comes at a cost: mean-field modeling in deep learning necessarily

simplifies the entire phenomenon. The results obtained are correct in the weak sense: they are only exact under strict assumptions, and are otherwise approximations. In complement, experimental work is used verify the phenomenons derived through mathematics under those unrealistic assumptions. Future work in MFT should consider second-order interactions when there are finite neurons while extending the mathematics to be universal for all neural networks features.

References

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112 (518):859–877, 2017.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 582–591. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/7339-which-neural-net-architectures-give-rise-to-exploding-and-vanishing-gradients.pdf`.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

D J. Thouless, Philip Anderson, and R G. Palmer. Solution of 'solvable model of a spin glass'. *Phil. Mag.*, 35:593–601, 03 1977. doi: 10.1080/14786437708235992.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Roman Novak Sam Schoenholz Jeffrey Pennington Jascha Sohl-dickstein Jaehoon Lee, Yasaman Bahri. Deep neural networks as gaussian processes. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1EA-M-0Z`.

Hilbert J. Kappen and Wim Wiegerinck. Second order approximations for probability models. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 220–226, Cambridge, MA, USA, 2000. MIT Press. URL `http://dl.acm.org/citation.cfm?id=3008751.3008784`.

Ryo Karakida, Shotaro Akaho, and Shun ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach, 2018.

Tatsuro Kawamoto, Masashi Tsubaki, and Tomoyuki Obuchi. Mean-field theory of graph neural networks in graph partitioning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4361–4371. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/7689-mean-field-theory-of-graph-neural-networks-in-graph-partitioning.pdf`.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Thomas Minka. Divergence measures and message passing. 01 2005.

Phan-Minh Nguyen. Mean field limit of the learning dynamics of multilayer neural networks. *arXiv preprint arXiv:1902.02880*, 2019.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4788–4798, 2017. URL `http://papers.nips.cc/paper/7064-resurrecting-the-sigmoid-in-deep-learning-through-dynamical-isometry-theory-and-practice`.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in neural information processing systems*, pages 3360–3368, 2016.

Arnu Pretorius, Elan Van Biljon, Steve Kroon, and Herman Kamper. Critical initialisation for deep signal propagation in noisy rectifier neural networks. In *NeurIPS*, 2018.

Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915*, 2018.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493, 2018.

Lawrence K. Saul, Tommi S. Jaakkola, and Michael I. Jordan. Mean field theory for sigmoid belief networks. *J. Artif. Intell. Res.*, 4:61–76, 1996.

Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

Leonard Susskind. Statistical Mechanics lecture 9. https://www.youtube.com/watch?v=AT4_S9vQJgc, 2013. Accessed: 2019-04-26.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008. ISSN 1935-8237. doi: 10.1561/2200000001. URL http://dx.doi.org/10.1561/2200000001.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5393–5402, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/xiao18a.html.

Eric P Xing, Michael I Jordan, and Stuart Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 583–591. Morgan Kaufmann Publishers Inc., 2002.

Ge Yang and Samuel Schoenholz. Mean field residual networks: On the edge of chaos. In *Advances in neural information processing systems*, pages 7103–7114, 2017a.

Greg Yang and Samuel S. Schoenholz. Mean field residual networks: On the edge of chaos. In *NIPS*, 2017b.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A mean field theory of batch normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyMDXnCcF7.

8 Appendix

8.1 Derivation for ELBO

The log likelihood of the evidence can be written as follows,

$$\begin{aligned}
\log P(X) &= \log \sum_Z P(Z, X) \\
&= \log \sum_Z Q(Z|X) \frac{P(X, Z)}{Q(Z|X)} && \text{(Introduce a distribution } Q(Z|X)) \\
&\geq \sum_Z Q(Z|X) \log \frac{P(X, Z)}{Q(Z|X)} && \text{(By Jensen's Inequality)} \\
&\geq \mathrm{E}_Q \left[\log \left(\frac{P(X, Z)}{Q(Z|X)} \right) \right] \\
&\geq \mathrm{E}_Q \left[\log P(X, Z) \right] - \mathrm{E}_Q \left[Q(Z|X) \right]
\end{aligned}$$

Thus,

$$\mathrm{ELBO(Q)} = \mathrm{E}_Q \left[\log P(X, Z) \right] - \mathrm{E}_Q \left[Q(Z|X) \right]$$

8.2 Variational Mean Field for the Ising model

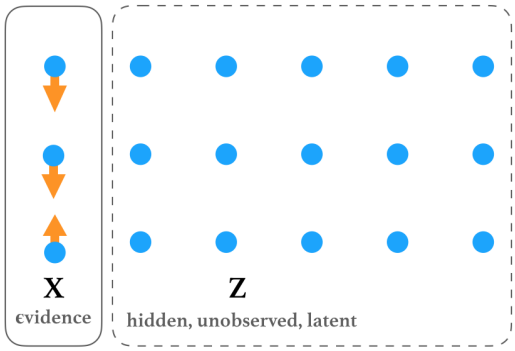

Figure 8: Partition an Ising Model into observed and unobserved σ's.

We represent the state of each atom i by a random variable σ_i which takes values +1 or -1. The marginal probability of a configuration of states $\boldsymbol{\sigma}$, is represented as:

$$p(\boldsymbol{\sigma}) \propto e^{-\beta H(\boldsymbol{\sigma})}$$

The normalization factor $Z = \sum_{\boldsymbol{\sigma}} e^{-\beta H(\boldsymbol{\sigma})}$ requires the sum over a huge number of configurations and so the exact marginal probability is often intractable (Sum

over 2^N and 2^{N^2} terms in case of 1D and 2D lattices which can be very large as N increases).

Our main goal is to use mean field methods to approximate the probability of a configuration of lattice points. One important thing to note is that, typically, the true distribution is not in the variational family obtained by mean field. We approximate the exact probability p($\boldsymbol{\sigma}$) with q($\boldsymbol{\sigma}$) such that q belongs to the exponential family of distributions. We consider that q is fully factorizable :-

$$q(\boldsymbol{\sigma}) = \prod_{i \in \mathcal{V}} q_i(\sigma_i) = \prod_{i=1}^{N} q_i(\sigma_i)$$

Our goal is to find q that acts a best approximation for p. For this we consider minimizing the Kullback-Liebler divergence between the two distributions.

$$q(\boldsymbol{\sigma}) = \arg\min_{q} D_{\mathrm{KL}}\left(\, q || \, p\right)$$

The above optimization is generally done in a coordinate descent fashion where we optimize the KL divergence with respect to a q_i one at a time. We first derive the result for minimizing the KL divergence between q and p with respect to some constituent q_k.

$$\min_{q_k} D_{\mathrm{KL}}\left(\, q || \, p\right) = \min_{q_k} D_{\mathrm{KL}}\left(\prod_{i=1}^{N} q_i \middle|\middle| \, p \right)$$

$$\begin{aligned}
D_{\mathrm{KL}}\left(\prod_{i=1}^{N} q_i \middle|\middle| \, p \right) &= \int (\prod_{i=1}^{N} q_i) \log \left(\frac{\prod_{j=1}^{N} q_j}{p} \right) \, d\boldsymbol{\sigma} \\
&= \int \prod_{i=1}^{N} q_i \sum_{j=1}^{N} \log q_j d\boldsymbol{\sigma} - \int \prod_{i=1}^{N} q_i \log p \; d\boldsymbol{\sigma} + C \\
&= \int \prod_{i=1}^{N} q_i \log q_k \; d\boldsymbol{\sigma} + \int \prod_{i=1}^{N} q_i \sum_{j \neq k} \log q_j \; d\boldsymbol{\sigma} - \int \prod_{i=1}^{N} q_i \log p \; d\boldsymbol{\sigma} + C \\
&= \int q_k \log q_k \; d\sigma_k + \underbrace{\int \prod_{i \neq k} q_i \sum_{i \neq k} \log q_i \; d\boldsymbol{\sigma_{\text{-}k}}}_{\text{constant wrt to } q_k} - \int \prod_{i=1}^{N} q_i \log p \; d\boldsymbol{\sigma} + C \\
&= \int q_k (\log q_k - \int \prod_{i \neq k} q_i \log p \; d\boldsymbol{\sigma_{\text{-}k}}) d\sigma_k + C^{'}
\end{aligned}$$

Let us consider $r(\sigma_k) = \int \prod_{i \neq k} q_i \log p \; d\boldsymbol{\sigma_{\text{-}k}}$. We can normalize this to obtain a

distribution $s(\sigma_k) = \frac{e^{r(\sigma_k)}}{\int e^{r(\sigma_i)} d\sigma_i}$. Now, the above equation can be written as:

$$\begin{aligned} D_{\mathrm{KL}}\left(\prod_{i=1}^{N} q_i \middle|\middle| p\right) &= \int q_k(\log q_k - \log s_k + \underbrace{\log \int e^{r(\sigma_i)} d\sigma_i)}_{\text{constant wrt to } q_k} d\sigma_k + C' \\ &= \int q_k \log \frac{q_k}{s_k} d\sigma_k + C" \underbrace{\int q_k d\boldsymbol{\sigma}_k}_{=1} + C' \\ &= D_{\mathrm{KL}}\left(\, q_k || \, s_k\right) + \text{Constant wrt to } q_k \end{aligned}$$

Hence, in order to minimize the KL divergence of variational distribution q and p with respect to q_k, we need to minimize the KL divergence between distribution q_k and s_k. Setting $q_k = s_k$, we get

$$\begin{aligned} q_k &= s_k \\ \log q_k &= \int \prod_{i \neq k} q_i \log p \; d\boldsymbol{\sigma_{-k}} + \text{Constant wrt to } \sigma_k \\ &= \mathrm{E}_{q_{-k}}[\log p] + C \end{aligned}$$

We will now use this result and apply to the Ising model. We know that that the joint probability of states is:

$$p(\boldsymbol{\sigma}) \propto e^{-\beta H(\sigma)}$$

where $H(\boldsymbol{\sigma}) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J\sigma_i\sigma_j - \sum_{i=1}^{N} B\sigma_i$

$$\begin{aligned} \log q_k(\sigma_k) &= \mathrm{E}\left[q_{-k}\right] - \beta H(\sigma) + \text{Constant} + C \\ &= \beta \mathrm{E}_{q_{-k}} \left[J\sigma_k \sum_{i \in Nbr(k)} \sigma_i + B\sigma_k + \text{Constant wrt to } \sigma_k \right] + C' \\ &= \beta\sigma_k (J \sum_{i \in Nbr(k)} \underbrace{\mathrm{E}\left[\sigma_i\right]}_{\mu_i} + B) + C" \\ &= \beta\sigma_k \underbrace{(J \sum_{i \in Nbr(k)} \mu_i + B)}_{\mathrm{H}} + C" \\ q_k(\sigma_k) &= C e^{\beta H \sigma_k} \end{aligned}$$

We will now find the value of the constant C.

$$\begin{aligned}
\int q_k(\sigma_k) d\sigma_k &= 1 \\
q_k(\sigma_k = 1) + q_k(\sigma_k = -1) &= 1 \\
Ce^{\beta H} + Ce^{-\beta H} &= 1 \\
C &= \frac{1}{e^{\beta H} + e^{-\beta H}}
\end{aligned}$$

Thus,

$$\begin{aligned}
q_k(\sigma_k) &= \frac{e^{\beta H \sigma_k}}{e^{\beta H} + e^{-\beta H}} \\
q_k(\sigma_k = 1) &= \text{Sigmoid}(2\beta H) \\
q_k(\sigma_k = -1) &= \text{Sigmoid}(-2\beta H) \\
\mu_k &= \text{E}\left[\sigma_k\right] \\
&= q_k(\sigma_k = 1) - q_k(\sigma_k = -1) \\
&= \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}} \\
&= \tanh(\beta H)
\end{aligned}$$

Note:

- Nbr(k) represents the nearest neighbours of lattice point k
- In the calculations above, we replaced $\text{E}_{q_{-k}}[\sigma_i]$ with $\text{E}\left[\sigma_i\right]$ because:

$$\begin{aligned}
\text{E}_{q_{-k}}\left[\sigma_i\right] &= \int \prod_{j \neq k} q_j \sigma_i d\sigma_{-k} \\
&= \int q_i \sigma_i d\sigma_i \underbrace{\int \prod_{j \neq \{k,i\}} q_j d\sigma_{-\{k,i\}}}_{1} \\
&= \int q_i \sigma_i d\sigma_i = \text{E}_{q_i}[\sigma_i]
\end{aligned}$$