Selected Examples of Physical Approaches to Neural Networks

Daniel S. Park

Google Brain

@ KIAS Colloquium

Motivation

Advances in deep neural network design and training have lead to remarkable progress in

- Image Recognition
- Speech Recognition
- Natural Language Processing

(and more!) abilities of machines in the last few years.

Motivation

While there have been amazing advances in the field,
it is fair to say that a scientific understanding of neural networks
is still at an unsatisfactory level.

Motivation

For example, we don't fully understand why
bigger neural networks, which have more parameters,
have better generalization properties than smaller ones.

One would expect that bigger models have more parameters,
and should over-fit more.

Motivation

As another example, given some task (say image recognition on a dataset), and two neural networks A and B, we don't have a good way of telling which network would do better on the task without actually training the two networks and testing their performance.

Motivation

On the other hand, neural networks are interesting physical systems that are ripe for systematic study:

- There exists a complete mathematical definition of the system.
- "Theorists" can run their own experiments and produce their own empirical data. (cf. Fermi-era high energy physics.)

Agenda

Today, I will review some basic definitions needed to understand neural networks and how they are trained, and will go over two examples where physics-based approaches have resulted in practical impact in the field.

If we have more time, I will give an overview of interesting research topics and potential directions of research.

Disclaimer

Most of the research I will be presenting today is done by other people. I strongly encourage the audience to read the original work, which I will provide pointers to throughout the talk.

Also, for the interest of time, I will be focusing on a very narrow sliver of even the "physics-adjacent research" done in the field, and will not have time to address large swaths of very exciting work. At the very end, I will point to some online resources for the audience to explore for themselves.

Please ask questions!

Table of Contents

Image Recognition Task

Training set images and labels: $T = \{(x^1, y^1), (x^2, y^2), ...\}$

Test set images and labels: $\hat{T} = \{(\hat{x}^1, \hat{y}^1), (\hat{x}^2, \hat{y}^2), ...\}$

Use T to train a network $\mathcal{N}$ such that the

$$(\text{test accuracy}) = \frac{(\#I \text{ s.t. } \mathcal{N}(\hat{x}^I) = \hat{y}^I)}{(\text{test set size})}$$

of the prediction of $\mathcal{N}$ is "satisfactory."

Image Recognition Task

Example (MNIST):

x: concatenated pixel values of  , y: 9

Multi-Layered Perceptrons (MLPs)

MLPs or "fully-connected networks" are
the harmonic oscillators of neural networks.

MLPs are the feed forward networks,
which means that an input is processed through
consecutive layers to yield an output.

Multi-Layered Perceptrons (MLPs)

Single input image @ 0th layer: $x = (x_{0,1}, \cdots, x_{0,N_0})$

Feed-forward @ layer ℓ:

$$z_{\ell,i} = W^{\ell}_{ij}\, x_{\ell-1,j} + b^{\ell}_{i}\,, \qquad x_{\ell,i} = \phi(z_{\ell,i})\,.$$

W, b: parameters of the network ("θ_α")
ϕ: A non-linear function (ReLU, tanh, sigmoid, etc)
N_ℓ: Width at layer ℓ (= dimension of z_ℓ)

Final Layer and Prediction

At the final layer at $\ell = L$, $z_{L,i}$ are interpreted as logits for labels, i.e.,

$$p_i(x) = \frac{\exp(z_{L,i})}{\sum_i \exp(z_{L,i})}$$

to be the probability for the image x to have label i.

The width of the final layer N_L should match the number of labels.

The network prediction is then given by

$$\mathcal{N}(x) = \text{argmax}_i \, p_i(x) \, .$$

Loss Function

We define a loss function for a sample (x, y) that we want to minimize during training:

$$\mathcal{L}(x, y) = -\log p_y(x)\,.$$

This function is minimized when $p_y(x) = 1$,
i.e., when the network says that
the probability of image x to have label y is 1.

Training

The networks are trained by stochastic gradient descent (SGD).

The training set is split into "minibatches" of size B.
For a minibatch

$$\mathcal{B} = \{(x^1, y^1), \cdots, (x^B, y^B)\},$$

the parameters are updated by the rule[1]

$$\Delta\theta_\alpha = -\epsilon \cdot \frac{\partial \mathcal{L}(\mathcal{B})}{\partial \theta_\alpha}$$

where

$$\mathcal{L}(\mathcal{B}) = \frac{1}{B} \sum_{l=1}^{B} \mathcal{L}(x^l, y^l).$$

[1] Or variants thereof (e.g., SGD with momentum, ADAM, etc)

Initialization

Need to initialize parameters before training:

$$W_{ij}^{\ell} \sim \mathcal{N}(0, \frac{\sigma_W}{\sqrt{N_{\ell-1}}}) \,, \quad b_i^{\ell} \sim \mathcal{N}(0, \sigma_b) \,.$$

Will assume $N_\ell = N$ for "inner layers" $(\ell = 1, \cdots, L-1)$.

Table of Contents

Reference

This part of the talk is based on:

S. S. Schoenholz, J. Gilmer, S. Ganguli, J. Sohl-Dickstein,
"Deep Information Propagation," arXiv:1611.01232.

S. S. Schoenholz, J. Pennington, J. Sohl-Dickstein,
"A correspondence between random neural networks and statistical field theory," arXiv:1710.06570.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz,
J. Pennington, J. Sohl-Dickstein,
"Deep Neural Networks as Gaussian Processes," arXiv:1711.00165.

Initialization Expectation Values

Note that at initialization, expectation values can be written as matrix integrals:

$$\langle f(z^l_{\ell,i})\rangle = \int \prod_\alpha [D\theta_\alpha]\, f(z^l_{\ell,i}(x^l, \theta_\alpha))$$

where $D\theta_\alpha$ is a normalized Gaussian measure for θ_α.

Wide Limit

(Lee & Bahri et al., arXiv:1711.00165)

When $N \to \infty$, a simplification occurs.

At a fixed depth ℓ,

$$\langle f(z^I_{\ell,i})\rangle = \int \prod_i \left(\prod_I dz^I_{\ell,i} \sqrt{\frac{\det(M^\ell)}{(2\pi)^{\mathbf{N}}}} \exp(-\frac{1}{2} M^\ell_{IJ}(x) z^I_{\ell,i} z^J_{\ell,i}) \right) f(z^I_{\ell,i}) \,,$$

where M^ℓ is a matrix only dependent on the inputs x^I, the depth ℓ and the initialization variables σ_W and σ_b.

We used $\mathbf{N}$ to denote training set size.

Wide Limit

(Lee & Bahri et al., arXiv:1711.00165)

The "Gaussian Kernel" $K^\ell \equiv (M^\ell)^{-1}$ satisfies the recursion relation:

$$K_{IJ}^{\ell+1} = \sigma_b^2 + \sigma_W^2 \cdot \langle \phi(z_\ell^I)\phi(z_\ell^J) \rangle_{z_\ell^I, z_\ell^J \sim \mathcal{N}(0, K^\ell|_{IJ})} \cdot$$

This is familiar territory for physicists
—we can utilize tools and intuition used for studying RG flow!

Variance

We first find that

$$\langle (z^l_{\ell+1})^2 \rangle = \sigma_b^2 + \sigma_W^2 \cdot \langle \phi(z^l_\ell)^2 \rangle \,.$$

If we denote the variance $V_{\ell+1} = \langle (z^l_{\ell+1})^2 \rangle$ of the value of a hidden unit at depth ℓ, we see that

$$V_{\ell+1} = \sigma_b^2 + \sigma_W^2 \cdot \int \mathcal{D}z \, \phi(\sqrt{V_\ell} \cdot z)^2 \,.$$

We use the shorthand

$$\mathcal{D}z = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \cdot dz \,.$$

Note that the fixed point value V^* for the variance only depends on σ_b and σ_W.

Covariance and Correlation

In fact, the fixed-point equations of
the variance, covariance and correlation

$$V^* = \lim_{\ell\to\infty} \langle (z_\ell^I)^2 \rangle \,, \quad C^* = \lim_{\ell\to\infty} \langle z_\ell^I z_\ell^J \rangle \,, \quad c^* = C^*/V^* \quad (I \neq J)$$

are independent of the input (i.e., the indices I and J).

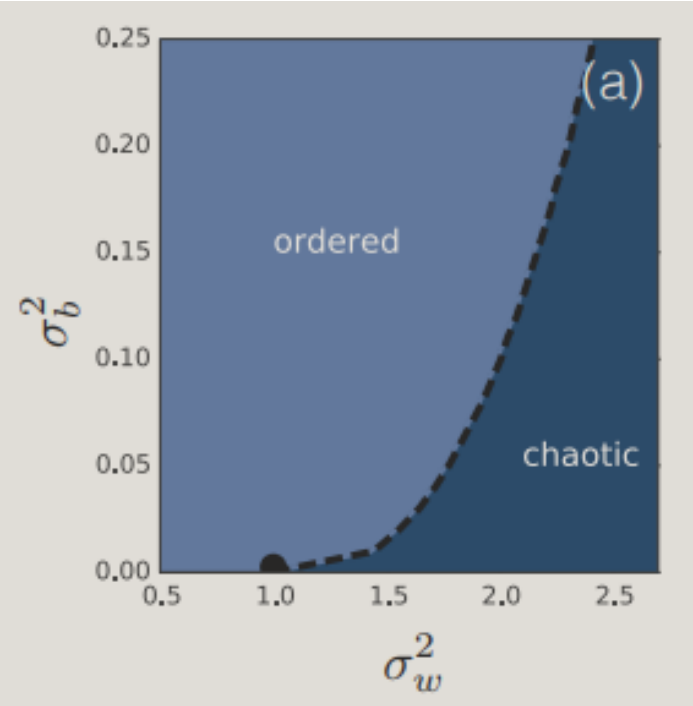
Phases

Let's restrict our attention to the case when $\phi = \tanh$.

A careful study of the aforementioned dynamical system,
which is the subject of (Schoenholz et al., arXiv:1611.01232),
shows that for deep networks

- There exist two phases of the dynamics of information propagation, depending on the value of c^*.
- In the ordered phase, $c^* = 1$ is the unique attractive fixed point.
  - $c^* = 1$ implies perfect correlation, so all pre-activations of all inputs asymptotically line-up, i.e., converge to the same vector.
- In the chaotic phase, the fixed-point $c^* = 1$ becomes repulsive and a new attractive fixed point $c^* < 1$ develops.

The Phase Diagram



* Figure taken from (Schoenholz et al., arXiv:1611.01232).

Depth Scale

The phase boundary displays critical behavior.

The correlation of pre-activations at depth ℓ, c_ℓ, has asymptotic scaling behavior

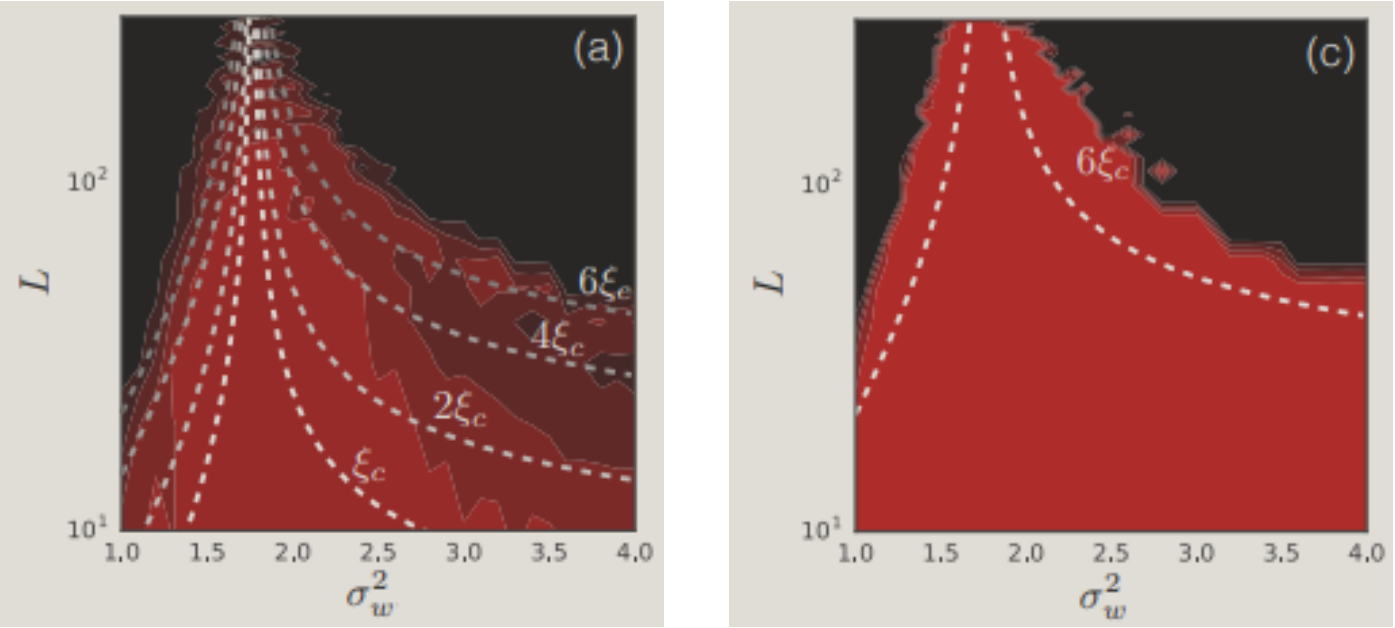
$$|c_\ell - c^*| \propto e^{-\ell/\xi_c} ,$$

where the depth scale is given by

$$\xi_c^{-1} = -\log\left[\sigma_W^2 \int \mathcal{D}z_1 \mathcal{D}z_2 \phi'(\sqrt{V^*}z_1)\phi'(\sqrt{V^*}(c^* z_1 + \sqrt{1-(c^*)^2}z_2))\right] .$$

At the phase boundary, the depth scale ξ_c diverges, which implies that the propagation depth of input information becomes infinite.

Experiments



* Figure taken from (Schoenholz et al., arXiv:1611.01232).

Depth Scale

Also, the expectation of the gradients with respect to the preactivations

$$g_\ell = \left\langle \left(\frac{\partial \mathcal{L}}{\partial z_\ell} \right)^2 \right\rangle$$

have an associated depth scale ξ_∇:

$$g_\ell = e^{-(L-\ell)/\xi_\nabla} g_L \, .$$

In the ordered phase, $\xi_\nabla > 0$ and the earlier gradients vanish, while in the chaotic phase $\xi_\nabla < 0$ and the earlier gradients explode. Meanwhile, at the phase boundary, $\xi_\nabla = 0$ and the gradient propagates uniformly through the layers of the network.

Conclusion

One should initialize at the phase boundary
to train deep networks effectively.

E.g., (Xiao et al., arXiv:1806.05393) were able to train 10,000-layer convolutional networks, just by initializing well!

Table of Contents

References

This part of the talk is based on:

S. L. Smith, Q. V. Le, "A Bayesian Perspective on Generalization and Stochastic Gradient Descent", arXiv:1710.06451.

S. L. Smith, P. Kindermans, C. Ying, Q. V. Le, "Don't Decay the Learning Rate, Increase the Batch Size", arXiv:1711.00489.

D. S. Park, J. Sohl-Dickstein, Q. V. Le, S. L. Smith, "The Effect of Network Width on Stochastic Gradient Descent and Generalization: an Empirical Study", arXiv:1905.03776.

Stochastic Gradient Descent (SGD)

Recall that neural networks are trained by SGD.

The training set is split into "minibatches" of size B.
For a minibatch

$$\mathcal{B} = \{(x^1, y^1), \cdots, (x^B, y^B)\},$$

the parameters are updated by

$$\Delta\theta_\alpha = -\epsilon \cdot \frac{\partial \mathcal{L}(\mathcal{B})}{\partial \theta_\alpha}$$

where

$$\mathcal{L}(\mathcal{B}) = \frac{1}{B} \sum_{l=1}^{B} \mathcal{L}(x^l, y^l).$$

Stochastic Gradient Descent (SGD)

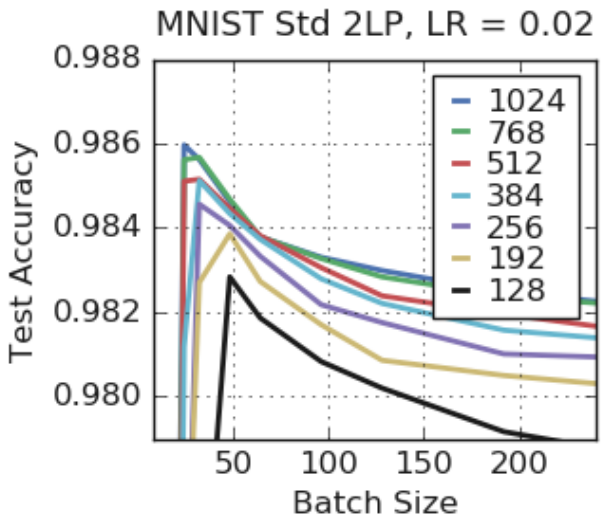
The gradients applied at each optimization step,
does not exactly match the "true gradient,"
where the gradient is taken with respect to the total loss

$$\mathcal{L}_\tau = \frac{1}{N} \sum_{l=1}^{N} \mathcal{L}(x^l, y^l)$$

We now use N to denote the training set size.

Stochastic Gradient Descent (SGD)

SGD with $B < N$ introduces noise to gradient descent,
which turns out to be beneficial to generalization performance.



I should note that networks train to have near-perfect performance
on the training set, while the test set performance is not.
This gap is broadly referred to as the "generalization gap."

SGD as Langevin Dynamics

(Smith & Le, arXiv:1710.06451)

In the regime of "small learning rate," ($dt = \epsilon/N << 1$)
SGD can be well-approximated by Langevin dynamics:

$$d\theta_\alpha = -\partial_\alpha \mathcal{L}_\tau dt + d\xi_\alpha \,, \quad \mathbb{E}[d\xi_\alpha d\xi_\beta]|_{w,t} = 2gF_{\alpha\beta}(w)dt$$

Here, g is the "noise scale" and F the covariance of the gradient:

$$g = \frac{\epsilon N}{2B} \,, \quad F_{\alpha\beta} = \frac{1}{N}\sum_s \frac{\partial}{\partial\theta_\alpha}\left(\mathcal{L}(s) - \mathcal{L}_\tau\right)\frac{\partial}{\partial\theta_\beta}\left(\mathcal{L}(s) - \mathcal{L}_\tau\right) \,.$$

We know how to deal with this.
(e.g. See Professor Noh's lecture from the 2017 Winter Camp.)

SGD as Langevin Dynamics

The probability density of the weights $u(\theta, t)$
evolves according to the Fokker-Planck equation:

$$\frac{\partial}{\partial t} u = \partial_\alpha [(\partial_\alpha \mathcal{L}_\tau) u] + g \partial_\alpha \partial_\beta (F_{\alpha\beta} u) \,.$$

Under favorable assumptions, asymptotes to $u_\infty(\theta)$:

$$\partial_\alpha [(\partial_\alpha \mathcal{L}_\tau) u_\infty] + g \partial_\alpha \partial_\beta (F_{\alpha\beta} u_\infty) = 0 \,.$$

This means, given the loss and dataset,
g is the only determining factor of u_∞!

SGD as Langevin Dynamics

If the noise were constant and isotropic:

$$F_{\alpha\beta} = \Gamma \delta_{\alpha\beta},$$

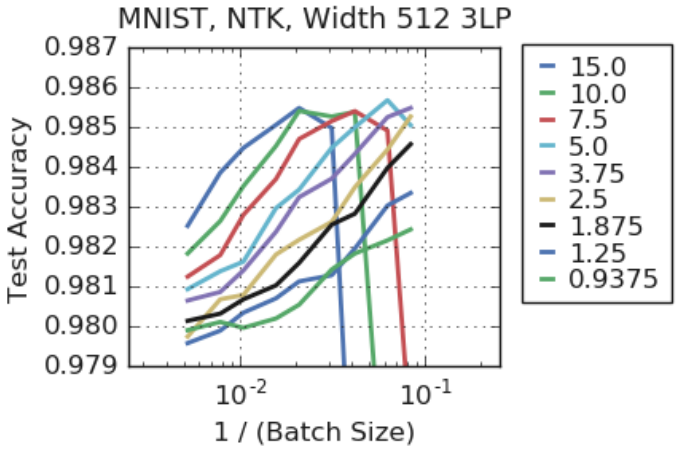
the stationary state becomes a Boltzmann distribution:

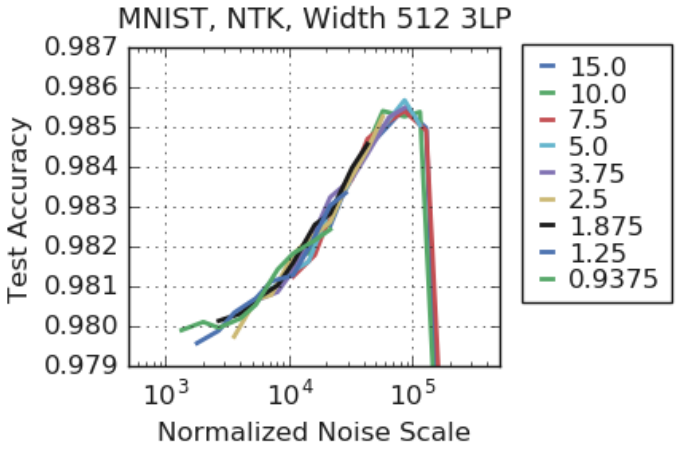
$$u_\infty = \exp(-\mathcal{L}_\tau / T)\,, \quad T = g\Gamma\,.$$

* $F_{\alpha\beta}$ is far from being isotropic for neural networks.

A Consequence

In the small learning rate regime, the late-time performance of a network does not depend on ϵ and B separately, but rather on the combination g:





Practical Application

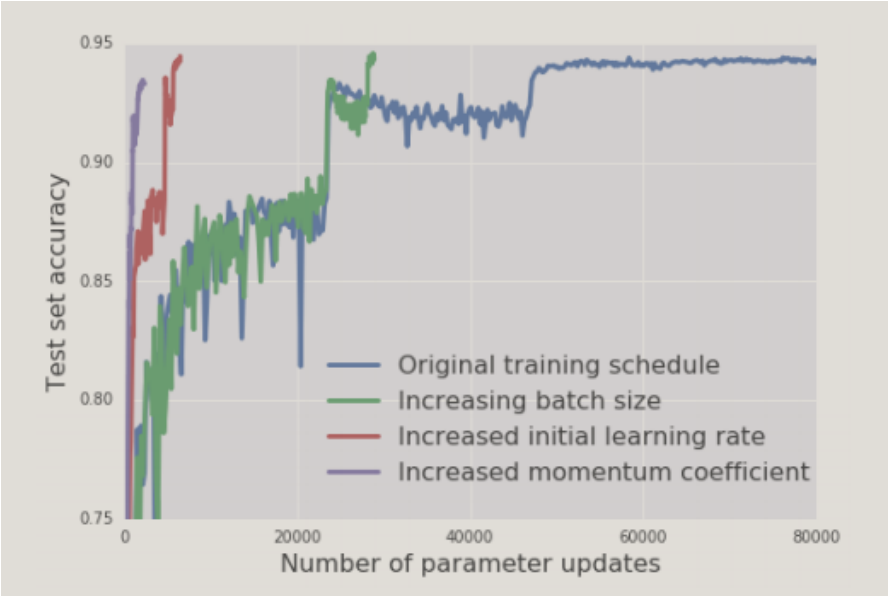
This observation resulted in acceleration of training time.
In practical network training, an "annealing schedule" is applied to the learning rate, where the learning rate is dropped during training.

Since the dynamics of training is entirely governed by $g \propto \epsilon/B$, the idea is to bump up B, instead of dropping ϵ.

Since $dt = \epsilon/N$, training at larger ϵ means less training steps.

Faster Training

By increasing the parallel compute budget, the training time can be drastically shortened without sacrificing generalization performance.



* Figure taken from (Smith et al., arXiv:1711.00489).

Table of Contents

Neural Networks at Initialization

Spectrum of input-output Jacobian and good initialization conditions

- J. Pennington, S. S. Schoenholz, S. Ganguli, "Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice," arXiv:1711.04735.

Neural networks as Gaussian processes

- J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, J. Sohl-Dickstein, "Deep Neural Networks as Gaussian Processes," arXiv:1711.00165.
- G. Yang, "Tensor Programs I: Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes," arXiv:1910.12478.

Training Dynamics

Wide-limit training dynamics (full-batch regime)

- A. Jacot, F. Gabriel, C. Hongler, "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," arXiv:1806.07572.
- J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, J. Pennington, "Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent," arXiv:1902.06720.
- E. Dyer, G. Gur-Ari, "Asymptotics of Wide Networks from Feynman Diagrams," arXiv:1909.11304.

Fluctuation dissipation theorem (small learning rate regime)

- S. Yaida, "Fluctuation-dissipation relations for stochastic gradient descent," arXiv:1810.00004.

Empirical Study of Networks

The bulk of the research in the field (including mine) focuses on making network performance better.

The number of papers carrying out systematic, scientific studies of neural network behavior is relatively small.

While the immediate reward for such studies might be small, I think the accumulation of such efforts would benefit the field long term.

Understanding Data Augmentation

Recently, a lot of performance gains of networks have resulted from data augmentation of one form or another.

The network is trained by feeding it "augmented data," where the data is obscured or distorted to help the network learn relevant features.

For example, in cut-out, small boxes of the image are "cut-out" before the image is fed into the network.
Sometimes color distortions or rotations/translations are applied.

Understanding Data Augmentation

Currently, data-augmentation is applied in an ad-hoc fashion.

A collection of useful augmentations are assembled by the researcher, and in the best-case scenario, there is some systematic method of selecting which combination of them to apply to help the network learn.

A more scientific approach would be desirable.

Personal Interest

The fantasy would be to be able to define macroscopic quantities that we can assign to tasks, networks or task-network pairs that can be computed efficiently to predict performance.

Taking this one step further, one might hope to optimize this measure on "network space" to find networks tailored to certain tasks.

Additional Resources for Physicists

Talks (slides/video) from past workshops

- Theoretical Physics for Machine Learning (Aspen, 2019)
- Physics Meets ML (Microsoft, 2019)
- Theoretical Physics for Deep Learning (ICML, 2019)

Review

- Y. Bahri, J. Kadmon, J. Pennington, S. Schoenholz, J. Sohl-Dickstein and S. Ganguli, "Statistical Mechanics of Deep Learning," vol 11. of Annual Review of Condensed Matter Physics (2020).

Thank you!