Towards a phenomenological understanding of NNs

20.02.2024, Liverpool
Sven Krippendorf (sven.krippendorf@physik.uni-muenchen.de, @krippendorfsven)
based on 2202.11104 (MLST), 2305.00995 (MLST), and wip, in collaboration with:

Michael Spannowsky
Sam Tovey
Konstantin Nikolaou
Christian Holm
Physics to understand NN dynamics
Problems and our approach

• We cannot afford hyperparameter scans for such large networks. How to successfully predict training performance?

| Parameters | 175 billion |
| Training Time | Several months |
| Training Cost | ~ $4.6 million |

| Neurons | 86 billion |
| Object recognition time$^2$ | 150 ms |
| Energy cost$^1$ | < 20 W |

$^1$ (Sterling & Laughlin, 2015), $^2$ (Thorpe et al., 1996)

• Our NN networks are not energy efficient. How to improve efficiency of NNs to make them useful with less computational resources?

cf. Lahiri, Sohl-Dickstein, Ganguli 1603.07758
Physics to understand NN dynamics
Problems and our approach

- We cannot afford hyperparameter scans for such large networks. *How to successfully predict training performance?*

<table>
<thead>
<tr>
<th>Parameters</th>
<th>175 billion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Time</td>
<td>Several months</td>
</tr>
<tr>
<td>Training Cost</td>
<td>~ $4.6 million</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Neurons</th>
<th>86 billion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object recognition time</td>
<td>150 ms</td>
</tr>
<tr>
<td>Energy cost</td>
<td>&lt; 20 W</td>
</tr>
</tbody>
</table>


- Our NN networks are not energy efficient. *How to improve efficiency of NNs to make them useful with less computational resources?*

  cf. Lahiri, Sohl-Dickstein, Ganguli 1603.07758

Describe neural networks & dynamics as effective field theories
Why effective field theories for NNs?

Effective field theories (EFTs) describe non-linear dynamics

NNs need non-linear dynamics
Why effective field theories for NNs?

Effective field theories (EFTs) describe non-linear dynamics

Our vision

- NN dynamics (especially of large NNs) are well-described by field theories with a specified range of validity (EFTs).
- The EFT is determined by the choice of architecture, data, optimiser and loss function.
- These are non-linear but understandable* models with few collective variables as degrees of freedom.

- Exciting because these FT-models promise to make our NNs more efficient.
- Exciting for physicists: the NN dynamics seem to be closely related to other physical systems.
Why effective field theories for NNs?

Effective field theories (EFTs) describe non-linear dynamics

Our vision

- NN dynamics (especially of large NNs) are well-described by field theories with a specified range of validity (EFTs).
- The EFT is determined by the choice of architecture, data, optimiser and loss function.
- These are non-linear but understandable* models with few collective variables as degrees of freedom.
- Exciting because these FT-models promise to make our NNs more efficient.
- Exciting for physicists: the NN dynamics seem to be closely related to other physical systems.

* at least for physicists

EFT methods & NNs:
Benta, Cai, Craig, Zhang …
How do we approach this?

Today’s content

• Identify a perturbative description of NN (i.e. where perturbations of NN dynamics are small): here large width and empirical NTK

• **Step 1: Field theory description of NTK.**
  How does the field theory change with different data, architecture, and optimiser choices?
  **Application:** Making training more data efficient

• **Step 2: Non-linear model building using collective variables.**
  **Application:** WIP on dynamical optimisers (built in bias of NTK spectrum)
FT ↔ NN

How field theories and neural networks are connected?

Field theory: \( \phi(x, t) \)

Parameters: Mode expansion \( \phi(x, t) = \sum_{k} b_k(t) e^{ikx} \)

In principle infinitely many modes (parameters), but only limited numbers are relevant, i.e. we can capture dynamics with a subset of modes.

Perturbative expansion (\( \delta \phi \ll \bar{\phi} \)): \( \phi(x, t) = \bar{\phi}(t) + \delta \phi(x, t) \)

Neural network: \( \phi(x, t) \)

Parameters: \( \phi(x, t) = \phi_{\theta}(x, t) \)

Here: mode decomposition due to NTK

cf. Halverson et al.
FT ↔ NN
How field theories and neural networks are connected?

Field theory: \( \phi(x, t) \)

Parameters: Mode expansion
\[
\phi(x, t) = \sum_k b_k(t) \ e^{ikx}
\]
In principle infinitely many modes (parameters), but only limited numbers are relevant, i.e. we can capture dynamics with a subset of modes.

Perturbative expansion (\( \delta \phi \ll \bar{\phi} \)): \( \phi(x, t) = \bar{\phi}(t) + \delta \phi(x, t) \)

+ dynamics

Neural network: \( \phi(x, t) \)

Parameters:
\[ \phi(x, t) = \phi_\theta(x, t) \]

Mean field

Here: mode decomposition due to NTK

cf. Halverson et al.
FT ↔ NN
How field theories and neural networks are connected?

Field theory: \( \phi(x, t) \)

Parameters: Mode expansion \( \phi(x, t) = \sum_{k} b_k(t) e^{ikx} \)

In principle infinitely many modes (parameters), but only limited numbers are relevant, i.e. we can capture dynamics with a subset of modes.

Perturbative expansion (\( \delta \phi \ll \bar{\phi} \)): \( \phi(x, t) = \bar{\phi}(t) + \delta \phi(x, t) \)

Neural network: \( \phi(x, t) \)

Parameters: \( \phi(x, t) = \phi_{\theta}(x, t) \)

Here: mode decomposition due to NTK

+ dynamics

UV Model
\( L = (\partial \phi)^2 + \lambda \phi^4 + m^2 \phi^2 \)
+ Initial conditions

\( \phi_{\text{initial}}(x) \)

Time evolution:
Analytically, Simulation (e.g. lattice)

Final field configuration

NN initialisation: \( \phi(t = 0, x) \)
Loss function
Optimiser

cf. Halverson et al.
How do we link dynamics of NNs and field theories?
Understand NN dynamics via empirical NTK

Simplification of dynamics in large width limit

- The dynamics of a neural network $f(x, \theta)$ simplify in the infinite width limit.
- The NN equations in continuous time limit:
  \[
  \dot{\theta} = -\eta \nabla_{\theta} \mathcal{L} = -\eta \nabla_{\theta} f(y) \nabla_f(y) \mathcal{L}
  \]
  \[
  \dot{f}(x) = \nabla_{\theta} f(x) \dot{\theta} = -\eta \nabla_{\theta} f(x) \nabla_{\theta} f(y) \nabla_f(y) \mathcal{L} = -\eta \Theta(x, y) \nabla_f(y) \mathcal{L}
  \]
- NN update simplify in large width limit: Neural tangent kernel remains constant (empirical and analytical):
  \[
  \Theta(t, x, y) = \Theta(t = 0, x, y)
  \]
- Complete as all learning components included: finite data, optimisers, and NN architecture
- Not sufficient (e.g. not capturing feature learning), in practice
  \[
  \Theta(t, x, y) \approx \Theta(t = 0, x, y)
  \]
  at finite but large width.

Which simple model describes the dynamics of NTK?
Scales in NN dynamics
Hierarchical spectrum in NTK → EFT approach promising

• Diagonalise NTK (\(\Theta_{\text{NTK}}\)) NN-update equation:

\[
\dot{f}(\mathcal{D}) = -\eta \text{ diag}(\lambda_1, \ldots, \lambda_N) \mathcal{L}'(\mathcal{D})
\]

• Largest changes in modes with largest eigenvalues.

• Hierarchical spectrum in NTK, consequences:
  • Effectively dynamics take place in lower-dimensional subspace. cf. Gur-Ari, Roberts, Dyer 2018
  • There are few “collective” variables in NTK which determine the dynamics. Their time evolution is what we need to understand.
  • Limit: adding more data does not change dynamics if non-vanishing eigenvalues are not changed.
Understand NTK dynamics by matching with (non-linear) physical systems...
Understand NTK dynamics by matching with (non-linear) physical systems…

I find it easier when I have a second order differential equation.

*The career of a young theoretical physicist consists of treating harmonic oscillator in ever-increasing levels of abstraction.*

Sidney Coleman
Neural Networks

Gradient descent with momentum

• Modification of optimizer: gradient descent with momentum

\[
\theta_{i+1} = \theta_i + v_i, \quad v_i = \beta v_{i-1} - \eta \nabla_{\theta} \mathcal{L}
\]

• NN differential equation becomes second order (more familiar from scalar field dynamics)

\[
\ddot{f}(x) + \frac{1 - \beta}{\sqrt{\eta}} \dot{f}(x) + \Theta(x, y) \mathcal{L}''(f(y)) = 0 \quad (\Delta t = i \sqrt{\eta})
\]

let’s return to this system in a second…
Let’s look at some physical system:

**Scalar fields in FLRW**
Cosmological toy models

- Scalar field in FLRW
  \[ S = \int d^4x \sqrt{-g} \left( R + \frac{1}{2} g^{\mu\nu} \partial_\mu \phi \partial_\nu \phi - V(\phi) \right) \]

- Homogeneous scalar field \( \phi(x, t) = \phi(t) \) eom:
  \[ \ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0, \quad 3H^2 = \frac{\dot{\phi}^2}{2} + V(\phi) \]

- Fluctuations around homogeneous background \( \phi(x, t) = \phi(t) + \delta\phi(x, t) \):
  \[ \dddot{\delta\phi} + \nabla^2 \delta\phi + 3H \dot{\delta\phi} + V''(\phi) \delta\phi = 0 \]

Dynamical properties:

- Perturbations (≈ non-trivial features) can grow and remain.
- Modes can (temporarily) freeze during cosmological evolution.
Cosmological toy models

- Scalar field in FLRW
  \[ S = \int d^4x \sqrt{-g} \left( R + \frac{1}{2} g^{\mu\nu} \partial_\mu \phi \partial_\nu \phi - V(\phi) \right) \]

- Homogeneous scalar field \( \phi(x, t) = \phi(t) \) eom:
  \[ \ddot{\phi} + 3H \dot{\phi} + V'(\phi) = 0 \]
  \[ 3H^2 = \frac{\dot{\phi}^2}{2} + V(\phi) \]

- Fluctuations around homogeneous background \( \phi(x, t) = \phi(t) + \delta\phi(x, t) \):
  \[ \ddot{\delta\phi} + \nabla^2 \delta\phi + 3H \dot{\delta\phi} + V''(\phi) \delta\phi = 0 \]

Dynamical properties:

- Perturbations (\( \approx \) non-trivial features) can grow and remain.
- Modes can (temporarily) freeze during cosmological evolution.
Are these two dynamical systems comparable?
Are these two dynamical systems related?

Homogeneous and isotropic dynamics

- NN eqn:
  \[
  \ddot{f}(x) + \frac{1 - \beta}{\sqrt{\eta}} \dot{f}(x) + \Theta(x, y) \mathcal{L}'(f(y)) = 0
  \]

- Homogeneous NN \( f(x, t) = f(t) \): no input dependence \( \rightarrow \) \( \Theta(x, y) = \alpha \)
  \[
  \ddot{f} + \frac{1 - \beta}{\sqrt{\eta}} \dot{f} + \alpha \mathcal{L}'(f) = 0
  \]

- Scalar field:
  \[
  \ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0 , \quad 3H^2 = \frac{\dot{\phi}^2}{2} + V(\phi)
  \]

- Vacuum energy dominated Universe \( H \approx \sqrt{V_0 / 3} = \text{const.} \):
  \[
  \ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0
  \]

\[
V_{\text{eff}} = \alpha \mathcal{L} + V_0 , \quad V_0 = \frac{\tilde{\beta}^2}{3}
\]
Are these two dynamical systems related?

Perturbation dynamics

- NN evolution equation:
  \[ \dot{f}(x) + \beta \ddot{f} + \Theta(t, x, X) \nabla_{f(x)} \mathcal{L} = 0 \]

- Rewrite last-term:
  \[ \Theta(t, x, X) \nabla_{f(X)} \mathcal{L} = \sum_y \Theta(t, x, y) \mathcal{L}'(f(y)) = \sum_y \Theta(t, x, y)(\mathcal{L}'(\bar{f}) + \mathcal{L}''(\bar{f}) \delta f(y)) \]

- Basis transformation to diagonalise NTK-kernel:
  \[ A \Theta A^T = \text{diag}(\lambda_1, \ldots, \lambda_N), A = \left( \begin{array}{c} v_1^T \\ v_2^T \\ \vdots \\ v_N^T \end{array} \right), f_i = \sqrt{N} v_i \bar{f} + \delta f_i, \text{ and } \mathcal{L}_i = \frac{m^2}{N} (f_i - f_0) \]

- Rewriting equations \( \delta \bar{f}_i = v_i^T \cdot \delta f \):
  \[ 0 = \ddot{\bar{f}} + \beta \dot{\bar{f}} + \frac{\lambda_1}{N} \mathcal{L}'(\bar{f}), \quad 0 = \delta \ddot{\bar{f}}_i + \beta \delta \dot{\bar{f}}_i + \frac{\lambda_i}{N} \delta \bar{f}_i \mathcal{L}''(\bar{f}) \]
Are these two systems related? YES

- Exact match subject to the following assumptions: vacuum energy dominated universe, low momentum modes
  \((\nabla^2 \phi \approx 0), \lambda_i \approx \lambda_1\)
  \[
  0 = \ddot{f} + \tilde{\beta} \dot{f} + \frac{\lambda_1}{N} \mathcal{L}'(\bar{f}) , \quad 0 = \delta \ddot{f}_i + \tilde{\beta} \delta \dot{f}_i + \frac{\lambda_i}{N} \delta \bar{f}_i \mathcal{L}''(\bar{f})
  \]
  \[
  \dot{\phi} + 3H \dot{\phi} + V'(\phi) = 0 , \quad \ddot{\phi} + \nabla^2 \phi + 3H \dot{\phi} + V''(\phi)\phi = 0
  \]

- Remarkable: structurally the equations look very similar even when relaxing conditions (e.g. \(\lambda_i \approx \lambda_1\))

- Hubble scale set by optimiser parameters:
  \[
  3H = \frac{1 - \beta}{\sqrt{\eta}}
  \]

- Adding more data points = adding more modes in evolution. If modes are irrelevant, they can be neglected. Understand generalisation behaviour for finite data?
Quantitatively matching dynamics
Quantitatively matching dynamics

Scope

1. When are these equations capturing the NN dynamics accurately?

\[ 0 = \ddot{\tilde{f}} + \beta \dot{f} + \frac{\lambda_1}{N} \mathcal{L}'(\tilde{f}) , \quad 0 = \delta \dot{\tilde{f}} + \beta \delta \dot{\tilde{f}} + \frac{\lambda_i}{N} \delta \tilde{f} \mathcal{L}''(\tilde{f}) \]

2. How do the EFT parameters depend on the NN parameters, e.g. determining \( \alpha \) empirically.

- Utilising existing empirical NTK implementation

https://github.com/google/neural-tangents
Quantitatively matching dynamics
NTK contribution to EFT potential

- How does the NTK vary when changing hyperparameters:

\[ V_{\text{eff}} = \alpha \mathcal{L} + V_0 \]
\[ \mathcal{L} = \frac{m^2}{2} (f - f_0)^2 \]

- Initialisation is most sensitive, scales in \( \alpha \) vary largely
Quantitatively matching dynamics
When are we in a vacuum energy dominated regime?

- Vacuum energy needs to dominate over other contributions to Hubble:
  \[ 3H = \frac{1 - \beta}{\sqrt{\eta}} \]
  \[ \mathcal{L} = \frac{m^2}{2} (f - f_0)^2 \]
- Reasonable learning rates are allowed!
Quantitatively matching dynamics
Homogeneous case

• Train homogeneous NNs with different hyperparameters and solve differential equation to check predicted vs. actual gradient descent evolution:
  \[ \ddot{f} + \frac{\beta - 1}{\sqrt{\eta}} \dot{f} + \alpha \mathcal{L}'(f) = 0 \]

• Loss function, e.g.:
  \[ \frac{m^2}{2} (f - f_0)^2 \]

• Colour: \( \alpha(t_{\text{max}}) - \alpha(t = 0) \)

• Hyperparameters: \{1,2,5\} hidden layers, \{erf, relu\} activation, \{100,1000\} width, initialisations, learning and momentum rates

Comparing scale of evolution and error of evolution
Quantitatively matching dynamics

Full dynamics

- Train networks with 100 points and vary hyperparameters to check whether predicted and actual dynamics match:

\[ 0 = \ddot{f} + \ddot{\beta} \cdot f + \frac{\lambda_1}{N} \mathcal{L}'(\bar{f}) \]

\[ 0 = \delta\ddot{f}_i + \ddot{\beta} \cdot \delta\ddot{f}_i + \frac{\lambda_i}{N} \cdot \delta\ddot{f}_i \cdot \mathcal{L}''(\bar{f}) \]

- Here: most perturbation modes are frozen. Dynamics happen in sub-space
Result: $\text{FT} \leftrightarrow \text{NN}$ also for dynamics

Can we use it?  
*Making NN training more data efficient*
EFT and data points
EFT and data points

• #data points (N) sets # modes as empirical NTK is N x N matrix:

$$\nabla_{\theta} f(x_i) \nabla_{\theta} f(x_j)$$

• Choice of data points sets which modes are chosen.
EFT and data points

• #data points (N) sets # modes as empirical NTK is N x N matrix:

\[ \nabla_{\theta}f(x_i) \nabla_{\theta}f(x_j) \]

• Choice of data points sets which modes are chosen.

  ▶ Can we make training more data efficient by selecting good data points?

\[ \dot{f}(x) = -\eta \nabla_{\theta}f(x) \nabla_{\theta}f(y) \nabla_{f(y)}\mathcal{L} \]

important for update!
Variables to capture significant changes in spectrum
Overall magnitude of NTK (trace) and diversity entropy
Variables to capture significant changes in spectrum

Overall magnitude of NTK (trace) and diversity entropy

• We see that the maximal eigenvalues of the NTK is very dominant and was relevant in the mean evolutions of the network:

\[ \text{Tr}(\Theta_{\text{NTK}}) = \sum_i \lambda_i \approx \lambda_{\text{max}} \]
Variables to capture significant changes in spectrum
Overall magnitude of NTK (trace) and diversity entropy

• We see that the maximal eigenvalues of the NTK is very dominant and was relevant in the mean evolutions of the network:

\[ \text{Tr}(\Theta_{\text{NTK}}) = \sum_i \lambda_i \approx \lambda_{\text{max}} \]

• The # of relevant modes differs between tasks. A variable which is independent of the # of modes is the following entropy:

\[ S^{VN} = - \sum_i \hat{\lambda}_i \log \hat{\lambda}_i \]

(here: \( \hat{\lambda}_i \) normalised eigenvalues of \( \Theta_{\text{NTK}} \))
Variables to capture significant changes in spectrum

Overall magnitude of NTK (trace) and diversity entropy

• We see that the maximal eigenvalues of the NTK is very dominant and was relevant in the mean evolutions of the network:

\[ \text{Tr}(\Theta_{NTK}) = \sum_i \lambda_i \approx \lambda_{\text{max}} \]

• The # of relevant modes differs between tasks. A variable which is independent of the # of modes is the following entropy:

\[ S^{VN} = - \sum_i \hat{\lambda}_i \log \hat{\lambda}_i \]

(here: \( \hat{\lambda}_i \) normalised eigenvalues of \( \Theta_{NTK} \))

‣ How do these two variables correlate with neural network behaviour?
Identifying collective variables
Testing behaviour in experiments

• How do our collective variables before training correlate with model behaviour after training on test set?

• Same network: different initialisations and different size of training data

• Trace and entropy are correlated with test loss in this example. But we also see, more data is leading to better results?
Disentangling: data size and collective variables

Data selection: Random Network Distillation (RND)

- What happens when we select datasets of the same size but with different collective variables values?
- We use Random Network Distillation (Burda et al. 2018) and randomly selected samples.
- RND selected samples show larger collective variables.

**Different datasets**

![Graphs showing different datasets](image)

**Workflow of RND**

1. \( p_i \in \mathcal{P} \)
2. \( \mathcal{D}(G(p_i), F(p_i)) = d \)
3. If \( d \leq \delta \) then re-train \( G \) on \((T, F(T))\)
4. If \( d > \delta \) then \( T = T \cup \{p_i\} \)
Disentangling: data size and collective variables

Data selection: Random Network Distillation (RND)

- RND datasets: higher collective variables and better test performance.

**Different datasets**

**Test performances**

**Workflow of RND**

\[ p_i \in \mathcal{P} \]

\[ D(G(p_i), F(p_i)) = d \]

\[ T = T \cup \{p_i\} \]

\[ \mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^M \]

\[ \mathcal{G} : \mathbb{R}^N \rightarrow \mathbb{R}^M \]

\[ d \leq \delta \]

\[ d > \delta \]
Collective variables

\[ \text{Tr}(\Theta_{\text{NTK}}), S = - \sum \lambda_i \log \lambda_i \]

- Data selection changes the spectrum and subsequently our collective variables:

\[ \dot{f}(x) = \nabla_{\theta} f(x) \quad \dot{\theta} = - \eta \Theta(x, y) \nabla_{f(y)} \mathcal{L} \]

- We see that both collective variables (trace and entropy of empirical NTK) correlate with generalisation behaviour.

- These collective variables seem like an interesting window into data selection.
How do collective variables evolve during training dynamics?
Dynamics of Collective Variables

Signs of universal evolution

• Train ensembles of networks on MNIST (1000 images) with SGD (lr=0.01), loss: cross-entropy

• Collective variables evolve in time. Trace evolves over few orders of magnitude.

• Interesting long-term behaviour.

• Universality within a class but visible differences among architectures.
Can we use collective variables for improved NN dynamics?

Collective variables for optimiser design

See also:
Lewkowycz et al. 2020
Kalra and Barkeshli 2023
Optimiser landscape
Roads for biasing optimisers

(Stochastic) gradient descent
Optimiser landscape
Roads for biasing optimisers

(Stochastic) gradient descent

Natural gradient descent (NGD)*:
\[
\dot{\theta} = -\eta F^{-1} \nabla L
\]

• Expensive (#parameters\(^2\)) and often only approximations available
• Hard to estimate: small eigenvalues (typically uncertain) change updates dramatically

*Think of it as steepest descent in model space with distance measured by KL-divergence
Review: Martens 2019
Optimiser landscape
Roads for biasing optimisers

Natural gradient descent (NGD)*:
\[ \dot{\theta} = -\eta F^{-1} \nabla \mathcal{L} \]
- Expensive (#parameters\(^2\)) and often only approximations available
- Hard to estimate: small eigenvalues (typically uncertain) change updates dramatically

(Stochastic) gradient descent

e.g. ADAM:
- Avoids expensive Fisher calculation
- Uses only diagonal entries of Fisher approximation
- Other modifications (memory)

*Think of it as steepest descent in model space with distance measured by KL-divergence
Review: Martens 2019
Optimiser landscape
Roads for biasing optimisers

Natural gradient descent (NGD)*: 
\[ \dot{\theta} = -\eta F^{-1} \nabla \mathcal{L} \]
- Expensive (#parameters\(^2\)) and often only approximations available
- Hard to estimate: small eigenvalues (typically uncertain) change updates dramatically

(Stochastic) gradient descent

Spectral Optimisers:
- Avoids expense of \( F^{-1} \)
- Uses spectral information

Today: Largest eigenvalue/Trace → Trace optimiser

e.g. ADAM:
- Avoids expensive Fisher calculation
- Uses only diagonal entries of Fisher approximation
- Other modifications (memory)

*Think of it as steepest descent in model space with distance measured by KL-divergence
Review: Martens 2019
Connecting collective variables and optimiser
Relation with Fisher information?

- Benefit of human designed collective variables: allow us to relate to collective variables in other physical systems.

- Fisher information metric:

\[
I_{ij} \approx \frac{1}{N} \sum_{k \in \text{data}} \partial_\theta_i L \partial_\theta_i L = \frac{1}{N} \sum_k \frac{\partial f_k}{\partial \theta_i} \frac{\partial f_k}{\partial \theta_j} \left( \frac{\partial \mathcal{L}}{\partial f_k} \right)^2
\]

- Trace of Fisher connection with trace of NTK:

\[
I_{ii} \approx \frac{1}{N} \sum_{i,k} \frac{\partial f_k}{\partial \theta_i} \frac{\partial f_k}{\partial \theta_i} \left( \frac{\partial \mathcal{L}}{\partial f_k} \right)^2 = \frac{1}{N} \sum_k \Theta_{kk} \left( \frac{\partial \mathcal{L}}{\partial f_k} \right)^2 \approx \lambda_{\text{max}} (L)^2 \approx \text{Tr}(\Theta) (L)^2
\]

(* Hierarchical spectrum of NTK dominated by largest eigenvalue)

- Aside, trace of Fisher has been used as order parameter in lattice statistical systems (cf. Prokopenko, Lizier, Obst, Wang)
Connecting collective variables and optimiser

Relation with Fisher information?

• Benefit of human designed collective variables: allow us to relate to collective variables in other physical systems.

• Fisher information metric:

\[ I_{ij} \approx \frac{1}{N} \sum_{k \in \text{data}} \partial_{\theta_i} L \partial_{\theta_j} L = \frac{1}{N} \sum_k \frac{\partial f_k}{\partial \theta_i} \frac{\partial f_k}{\partial \theta_j} \left( \frac{\partial \mathcal{L}}{\partial f_k} \right)^2 \]

• Trace of Fisher connection with trace of NTK:

\[ I_{ii} \approx \frac{1}{N} \sum_{i,k} \frac{\partial f_k}{\partial \theta_i} \frac{\partial f_k}{\partial \theta_i} \left( \frac{\partial \mathcal{L}}{\partial f_k} \right)^2 = \frac{1}{N} \sum_k \Theta_{kk} \left( \frac{\partial \mathcal{L}}{\partial f_k} \right)^2 \approx \lambda_{\text{max}}(L)^2 \approx \text{Tr}(\Theta) \ (L)^2 \]

(* Hierarchical spectrum of NTK dominated by largest eigenvalue)

• Aside, trace of Fisher has been used as order parameter in lattice statistical systems (cf. Prokopenko, Lizier, Obst, Wang)

This motivates to rescale the learning rate with the trace of the NTK to emulate a very coarse feature of NGD:

\[ \dot{\theta} = -\eta \ F^{-1} \nabla \mathcal{L} \]
Trace Optimiser
Optimiser with single spectral property

• Trace optimiser, initial learning rate at 1:

\[
\theta \rightarrow \theta - \frac{\text{Tr } \Theta_{NTK}(t = 0)}{\text{Tr } \Theta_{NTK}(t)} \nabla_{\theta} \mathcal{L}
\]

• MNIST, Dense: less overfitting, less erratic, similar speed to ADAM

• On-going: other architectures, tasks, speed comparisons; when do we need to go beyond trace optimiser
Conclusions

• Effective field theories for non-linear neural network dynamics

• Connection to physical systems: NN dynamics \( \approx \) scalar field in FLRW (vacuum dominated)

\[
0 = \ddot{\bar{f}} + \tilde{\beta} \dot{\bar{f}} + \frac{\lambda_1}{N} \mathcal{L}'(\bar{f}) , \quad 0 = \delta \ddot{\bar{f}}_i + \tilde{\beta} \delta \dot{\bar{f}}_i + \frac{\lambda_i}{N} \delta \bar{f}_i \mathcal{L}''(\bar{f})
\]

• Hierarchical/subspace dynamics of parameter space \( \rightarrow \) collective variables

• Collective variables useful to improve training: data selection

• Collective variables evolve in time (simple dynamics of NTK?)

• Spectrum adapted optimizers to keep good features of NGD at efficient cost
Thank you!

based on 2202.11104 (MLST*), 2305.00995 (MLST*), and wip

* Recent journal for articles at the interface of physics and machine learning.

New Master of Physics with a specialisation in AI @ LMU Munich
Collective Variables in Teacher-Student Setting
Ongoing exploration of various settings — stay tuned