Neural Networks from the Perspective of Physics

Jaeok Yi

Department of Physics, KAIST

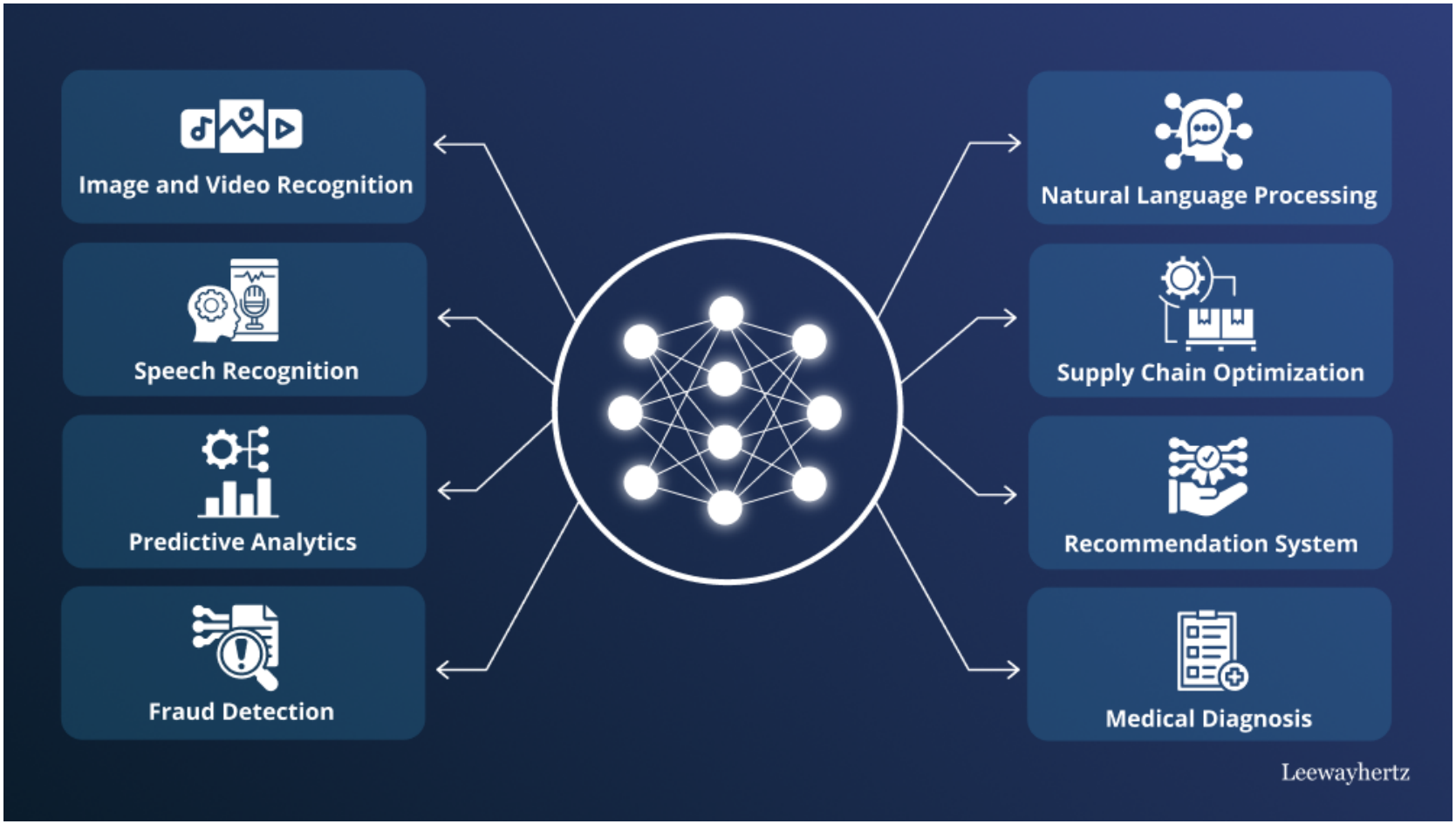
April 25, 2026

based on arXiv: 2511.02003, 2605.XXXXX
with Donghee Lee and Hye-Sung Lee

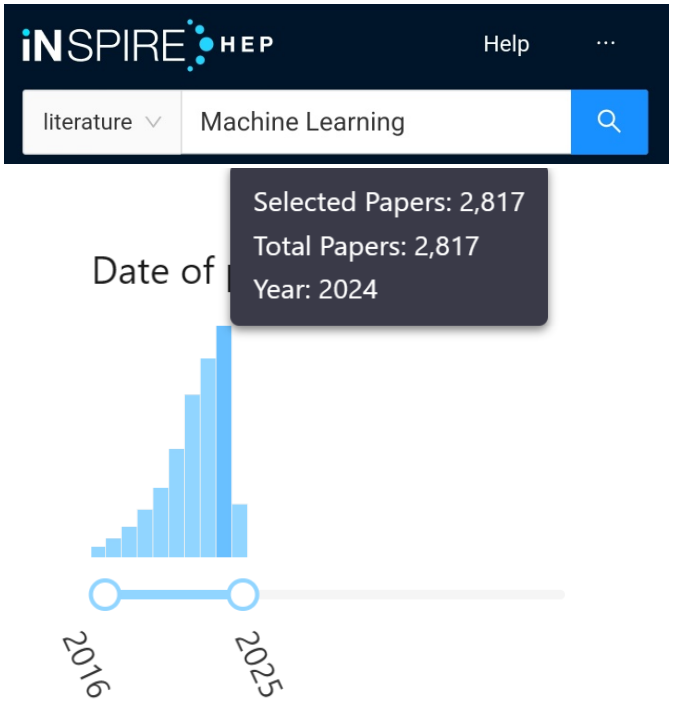
Overview

I. Introduction

Machine Learning



Machine Learning and High Energy Physics



- Machine learning is also a topic of great interest in high energy physics.
  - Parton Distribution Function
  - Jet Classification
  - Constraining Effective Field Theories
  - Anomaly Detections
  - ...

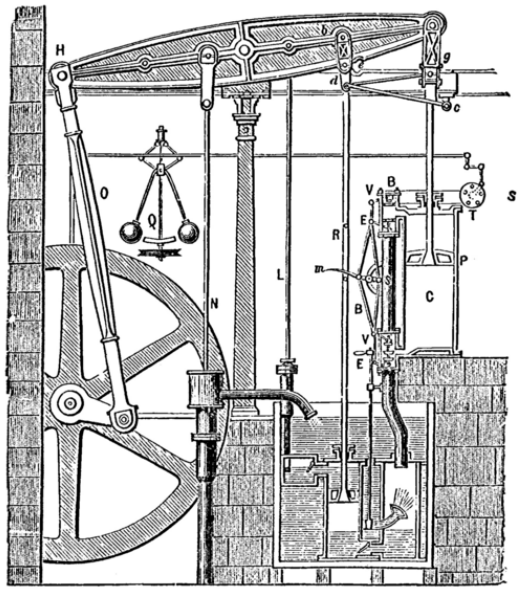
Machine Learning is Still a Mystery

We have a rough idea of what it's doing,
but when it gets complicated,
we don't know what's going on,
similar to our understanding of the brain.

Geoffrey Hinton
(2024 Nobel Laureate in Physics)

Technology and Physics

- Technological development sometimes comes before full theoretical understanding.
  - Steam Engine & Thermodynamics



[James Watt, 1774]

[Sadi Carnot, 1824]

- Once the physics is clear, progress tends to accelerate.

Physics behind Machine Learning

- Understanding the physics behind machine learning could drive its future progress.

Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
jaredk@jhu.edu

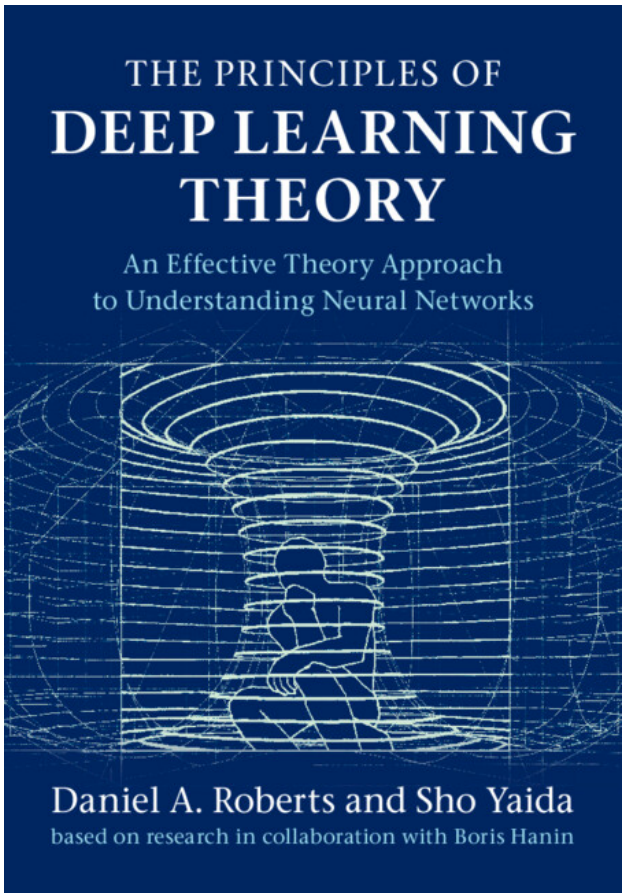
Sam McCandlish*
OpenAI
sam@openai.com

A duality connecting neural network and cosmological dynamics

Sven Krippendorf[a,1], **Michael Spannowsky**[b,c,2],

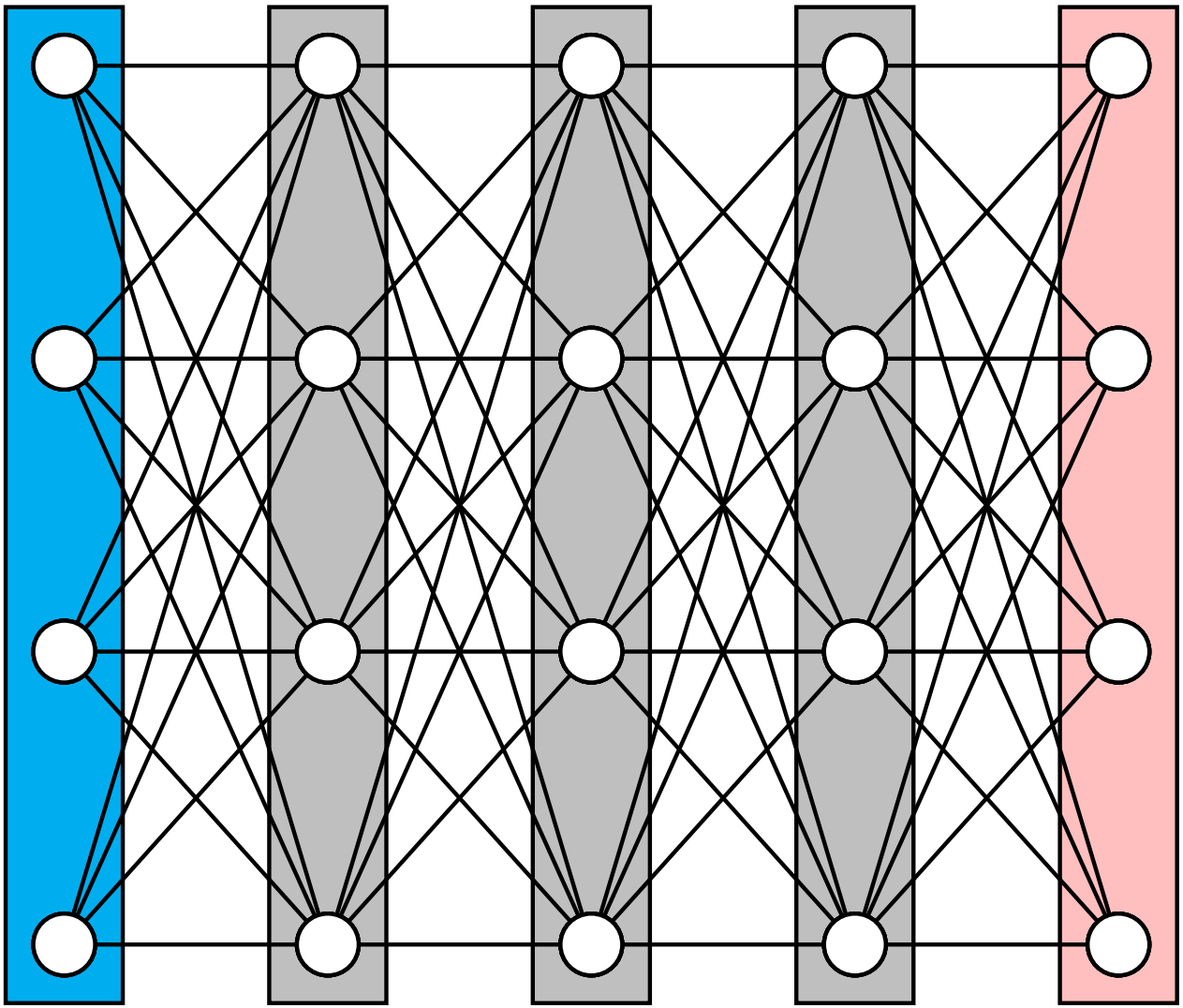
Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi[a], P. Perdikaris[b,*], G.E. Karniadakis[a]

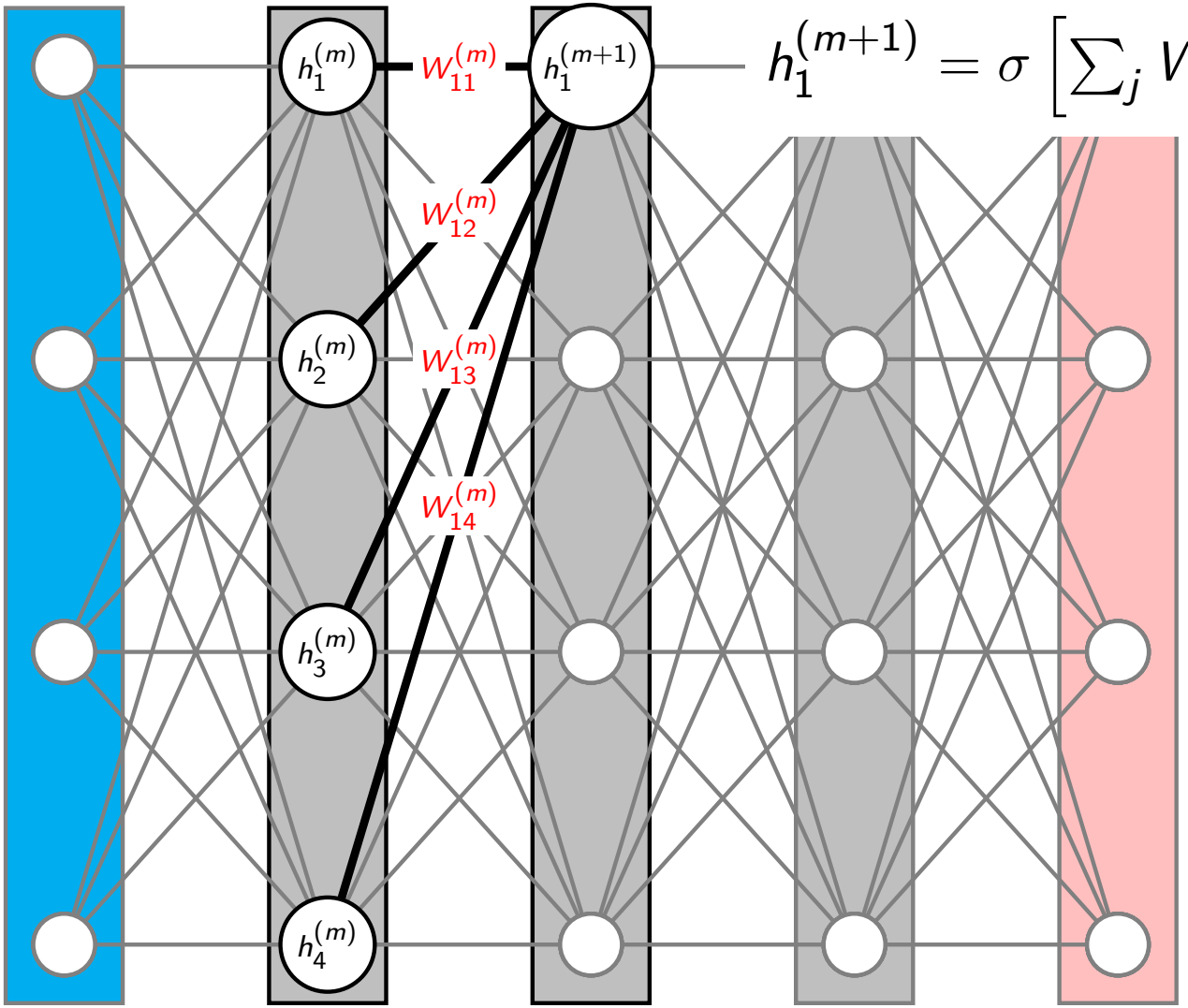


II. Machine Learning 101

Neural Networks

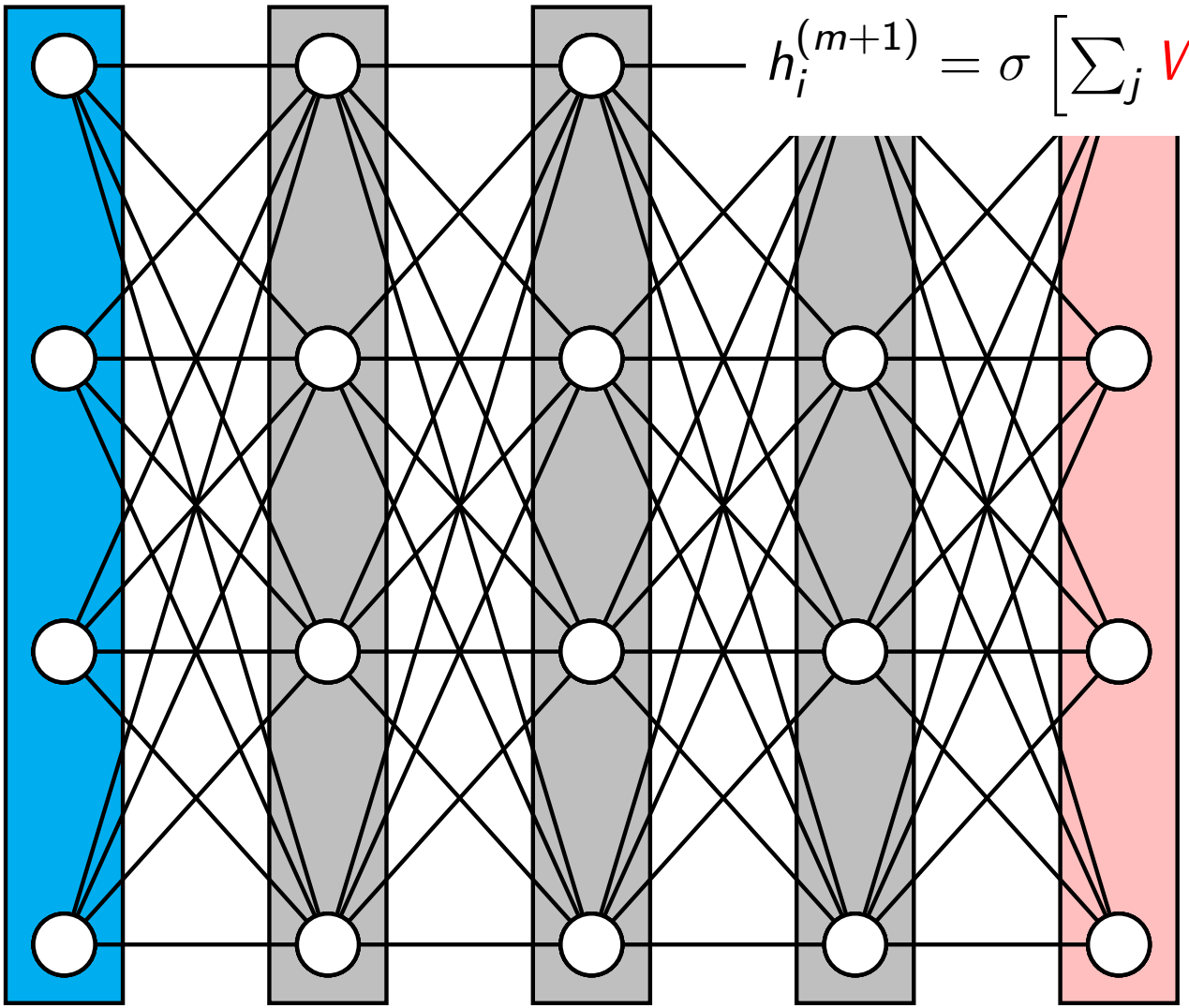


Neural Networks



$$h_1^{(m+1)} = \sigma\left[\sum_j W_{1j}^{(m)} h_j^{(m)} + b_1^{(m)}\right]$$

Neural Networks



$$h_i^{(m+1)} = \sigma\left[\sum_j W_{ij}^{(m)} h_j^{(m)} + b_i^{(m)}\right]$$

Universal Approximation Theorem

- For any arbitrary continuous function, there exists a set of synaptic weights such that a neural network can approximate it.
  - Infinite width cases: proved

Universal approximation theorem—Let $C(X, \mathbb{R}^m)$ denote the set of continuous functions from a subset X of a Euclidean $\mathbb{R}^n$ space to a Euclidean space $\mathbb{R}^m$. Let $\sigma \in C(\mathbb{R}, \mathbb{R})$. Note that $(\sigma \circ x)_i = \sigma(x_i)$, so $\sigma \circ x$ denotes σ applied to each component of x.

Then σ is not polynomial if and only if for every $n \in \mathbb{N}$, $m \in \mathbb{N}$, compact $K \subseteq \mathbb{R}^n$, $f \in C(K, \mathbb{R}^m)$, $\varepsilon > 0$ there exist $k \in \mathbb{N}$, $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$ such that

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

where $g(x) = C \cdot (\sigma \circ (A \cdot x + b))$

  - Infinite depth or bounded depth and width cases: partially proved

What We Do Not Know about Machine Learning

- The universal approximation theorem guarantees the existence of a solution but it does not provide a method for finding the solution.
  - "We are not guaranteed, however, that the training algorithm will be able to learn that function."

[Goodfellow, I., Bengio, Y., & Courville, A. (2018). Deep learning. MITP.]

- It is still unknown whether training algorithms actually find the solutions guaranteed by the universal approximation theorem.

- Training Dataset -

$$7 + 2 = 9$$
$$5 + 3 = 8$$
$$4 + 2 = 6$$
$$3 + 1 = 4$$

- Test Artificial intelligence -

$$5 + 4 = ?$$

Gradient Descent

- Prepare the training set $(X_i^{[l]}, Y_i^{[l]})$ and the define the cost function:

$$C = \sum_{i,l} (Y_i^{[l]} - Z_i^{[l]})^2$$

  where $Z_i^{[l]}$ is the result of the neural network for $X_i^{[l]}$.
- Update the synaptic weights and biases using gradient descent:

$$\Delta W_{ij}^{(m)} = -\eta \frac{\partial C}{\partial W_{ij}^{(m)}}, \qquad \Delta b_i^{(m)} = -\eta \frac{\partial C}{\partial b_i^{(m)}},$$

  with the step size η.

III. Bulk-Boundary Decomposition

Lagrangian Approach to Gradient Descent

- In the continuum limit, the equation for gradient descent becomes

$$\dot{W} = -\eta \frac{\partial C}{\partial W}$$

- It can be considered as the high-viscosity limit ($\gamma = \eta^{-1} \gg 1$) of

$$\ddot{W} + \gamma \dot{W} + \frac{\partial C}{\partial W} = 0$$

- This equation can be derived from the action given as

$$S = \int dt \; e^{\gamma t} \left[\frac{1}{2} \dot{W}^2 - C \right]$$

Complexity in Cost Function

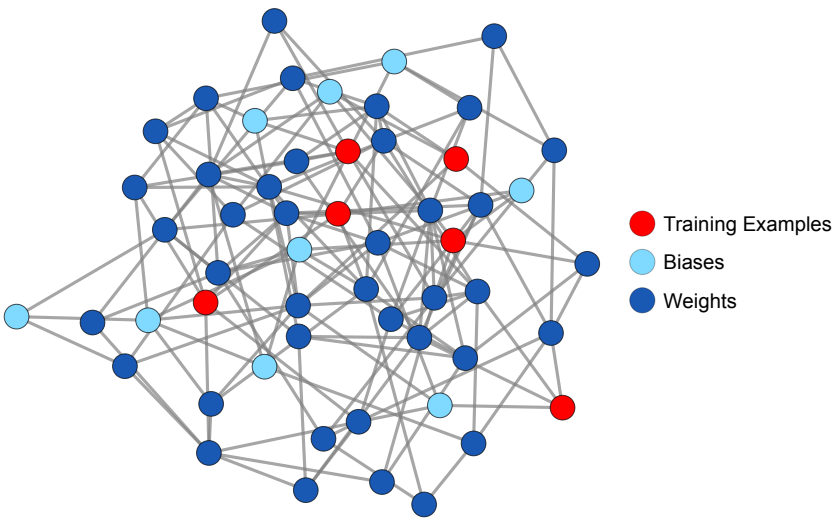
- The cost function C depends on all synaptic parameters across every layer.

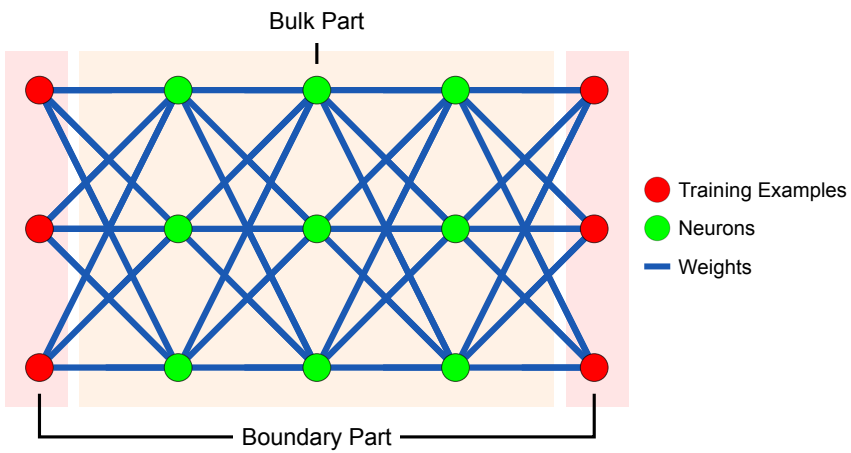
$$C = \sum_{i,I} (Y_i^{[I]} - Z_i^{[I]})^2 = \sum_{i,I} \left(Y_i^{[I]} - \sigma \left[\cdots \sigma \left[\sum_j W_{ij}^{(1)} X_j^{[I]} + b_i^{(1)} \right] \cdots \right] \right)^2$$

- Since the synaptic parameters, not the neurons, are the degrees of freedom, the Lagrangian is extremely complicated.
- If we treat the neurons as the degrees of freedom, the Langrangian may be simplified.

Revealing Locality

- Describing the interaction in terms of synaptic parameters through the connectivity of graph implies a nonlocal, complex structure.
- The standard visualization of neural networks intuitively suggests an inherent locality, since the neurons are denoted explicitly.





- To reveal this locality, we can attempt to promote the neurons to degrees of freedom.

Stochastic Gradient Descent

- In stochastic gradient descent, a single randomly chosen training example is used at each step.

$$\dot{W} = -\eta \frac{\partial C(t)}{\partial W} = -\eta \frac{\partial}{\partial W}[Y_i(t) - Z_i(W, X_i(t))]^2$$

  where $(X_i(t), Y_i(t))$ is the training example randomly chosen at time t and $Z_i(W, X_i(t))$ is the output of the neural network corresponding to the input $X_i(t)$.
- Almost all training algorithms are based on stochastic gradient descent.
  - Nearly all of deep learning is powered by one very important algorithm: stochastic gradient descent.

[Goodfellow, I., Bengio, Y., & Courville, A. (2018). Deep learning. MITP.]

Change of Variables

- Since a single data is inovlved at each time, there is one to one correspondence between neuron $z_i^{(m+1)}$ and bias $b_i^{(m)}$.

$$\begin{aligned} Z_i &= h_i^{(M)} \\ h_i^{(m+1)} &= \sigma(z_i^{(m+1)}), \\ z_i^{(m+1)} &= \sum_j W_{ij}^{(m)} h_j^{(m)} + b_i^{(m)}. \end{aligned}$$

- From the last relation, we may try the change of variables from b to z.

Bulk-Boundary Decomposition

- Such change of variables naturally introduces the decomposition of bulk and boundary part in the Lagrangian.

$$L = L_{\text{bulk}} + L_{\text{boundary}},$$

$$\begin{aligned} L_{\text{bulk}} = \frac{1}{2}\sum_{i,j,m}\left(\dot{W}_{ij}^{(m)}\right)^2 + \frac{1}{2}\sum_{i,m}\left(\dot{z}_i^{(m)}\right)^2 \\ - \sum_{i,j,m}\dot{z}_i^{(m+1)}\partial_t\left(W_{ij}^{(m)}\sigma(z_j^{(m)})\right) \\ + \frac{1}{2}\sum_{i,m}\left[\sum_j \partial_t\left(W_{ij}^{(m)}\sigma(z_j^{(m)})\right)\right]^2, \end{aligned}$$

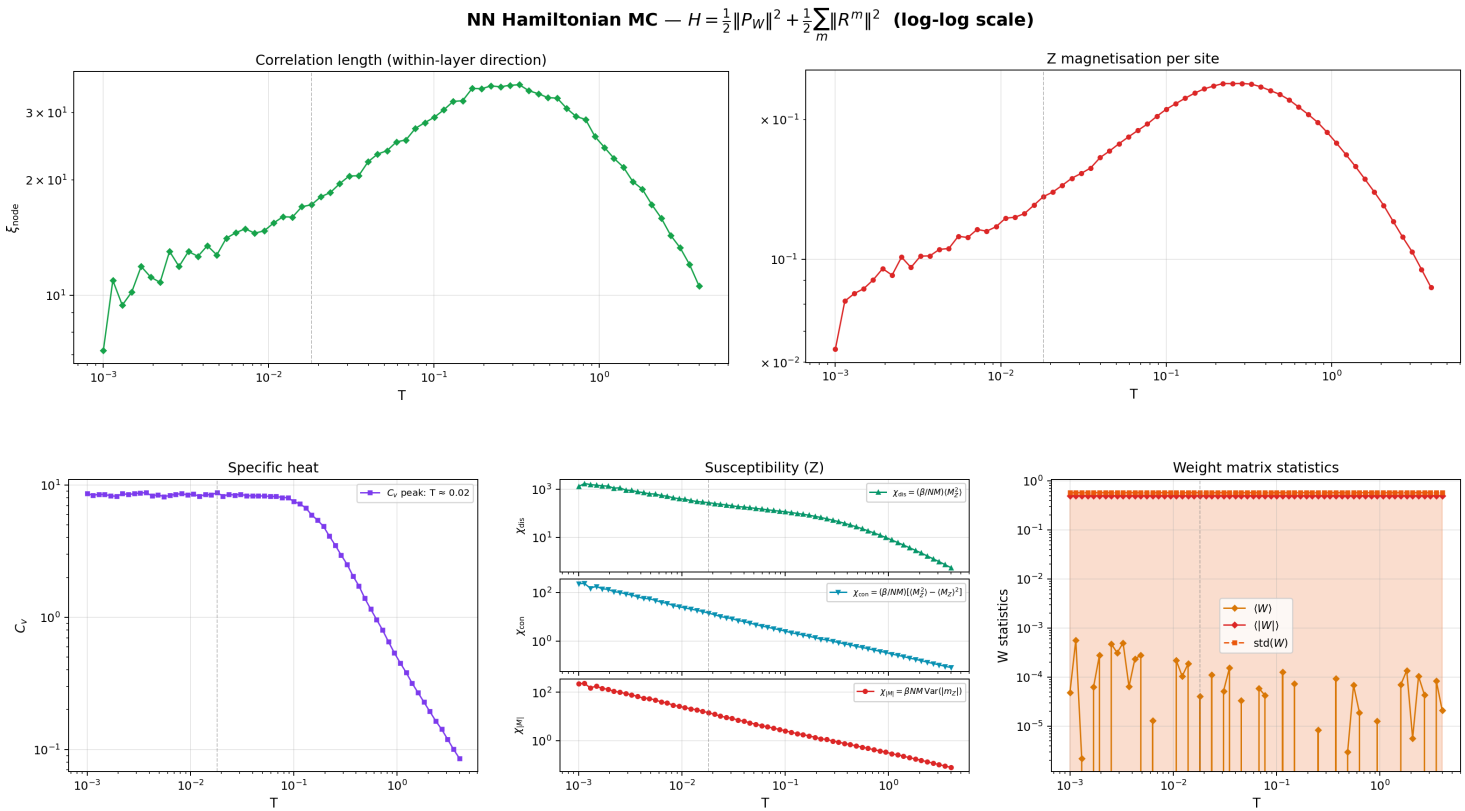
$$L_{\text{boundary}} = \sum_i\left[\sum_j \partial_t\left(W_{ij}^{(0)}X_j\right)\right]^2 - \sum_{i,j}\dot{z}_i^{(1)}\partial_t\left(W_{ij}^{(0)}X_j\right) - \ell(Z,Y).$$

Implications

- Under this decomposition, L reveals a local structure. In other words, the interaction is restricted to the nearby layers.
- L_{bulk} does not directly couple to the training data (X, Y) and is expected to encode the architectural properties of the neural network.
- L_{boundary} directly couples to the training data (X, Y) and would capture the stochastic properties arising from the training dataset.
- This decomposition, bulk-boundary decomposition, may enable a divide-and-conquer approach, allowing the architecture and the data dependence to be studied separately.
- Furthermore, it exhibits a repetitive structure, built as a sum over terms of the same form, similar to condensed matter systems.

Future Aspects

- Bulk part is connected to the stochastic background (boundary part)
  $\Rightarrow$ Thermodynamic Analysis?



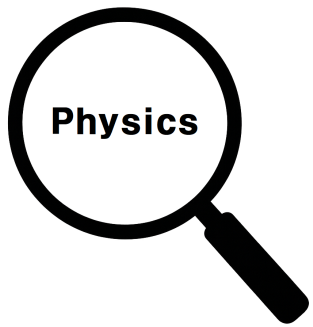
V. Summary

Summary

- Understanding the physics of neural networks can drive its future progress.
- Promoting neurons to dynamical degrees of freedom reveals the hidden locality of neural networks.
- Stochastic gradient descent naturally limits the number of dynamical neurons at each time step, simplifying the analysis.
- The bulk-boundary decomposition separates architectural properties from the data dependence, enabling a divide-and-conquer approach.

Final Note

- Understanding the nature of deep learning is the mission of physicists so more physics is warranted.



Thank you for listening

Back-up Slides

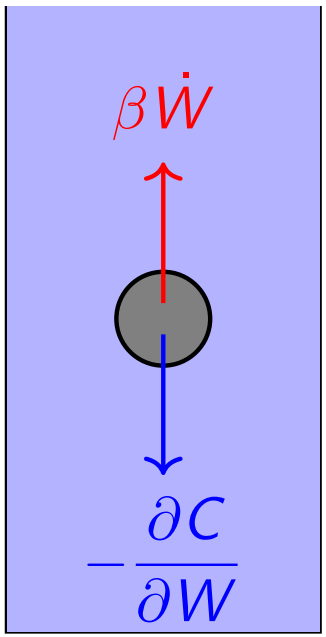
Discussions

- The linear activation is used and results in the bilinear action.
  - If we use non-polynomial activations, then higher order interaction terms should be considered.
- The previous example is not Lorentz invariant.
  - With a time-dependent training algorithm such as stochastic gradient and carefully designed training datasets, a Lorentz-invariant theory may emerge.
- The previous example results in the local action due to the simple structure and the practical indexing convention.
  - For the finite depth neural networks, the naive approach of continuum limit will result in nonlocal terms like

$$\int d^d\mathbf{x} d^d\mathbf{y}\ A(\mathbf{x},\mathbf{y}) w(t,\mathbf{x}) w(t,\mathbf{y})$$

  - As shown in the example, the structure and the indexing convention will be related to the geometry and locality of the space of the theory.

High-viscosity Limit



- High Viscosity Medium
- Large Drag Force
- Terminal Velocity
- $\ddot{W} = 0$

High Viscosity Limit

- In the high-viscosity limit, $\eta = 1/\gamma$ is small, allowing a perturbative expansion:

$$W = \mathbb{W}^{(0)} + \eta \mathbb{W}^{(1)} + \mathcal{O}(\eta^2).$$

- The equation from the action becomes

$$\frac{1}{\eta}\dot{\mathbb{W}}^{(0)} + \left(\ddot{\mathbb{W}}^{(0)} + \dot{\mathbb{W}}^{(1)} + \left.\frac{\partial C}{\partial W}\right|_{W=\mathbb{W}^{(0)}}\right) + \mathcal{O}(\eta) = 0.$$

- At $\mathcal{O}(\eta^{-1})$, we find $\dot{\mathbb{W}}^{(0)} = 0$ at any t, which implies $\ddot{\mathbb{W}}^{(0)} = 0$. Therefore, at $\mathcal{O}(\eta^0)$, we have

$$\dot{\mathbb{W}}^{(1)} + \left.\frac{\partial C}{\partial W}\right|_{W=\mathbb{W}^{(0)}} = 0.$$

It is the equation of motion for the training of neural networks.

Field Theoretic Approach to Neural Networks

- There exist some previous works trying to apply field theory to neural networks.
- Krippendorf and Spannowsky attempted to develop an effective theory of outputs of neural network and proposed a relationship between neural networks and cosmology.

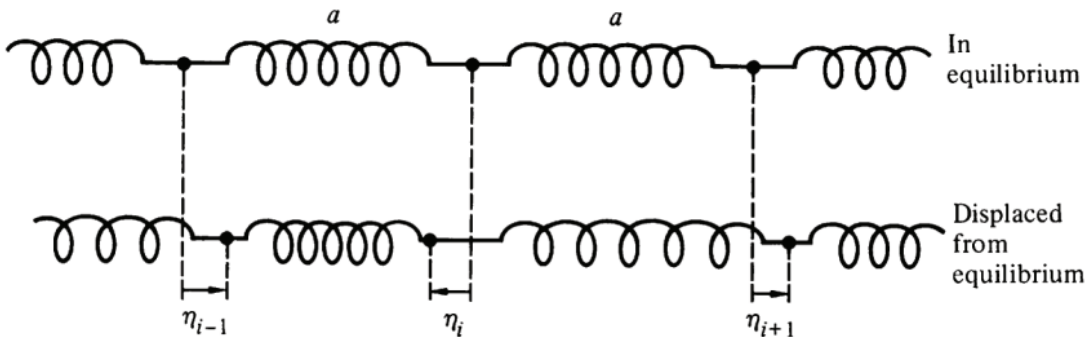
[S. Krippendorf and M. Spannowsky Mach.Learn.Sci.Tech. 3 (2022) 3, 035011]

- To do so, they considered the limit where the effect from synaptic weights and biases becomes a constant.
- Since weights and biases are fundamental building blocks, their effects should not be neglected.
- The theory dealing with fields developed by the continuum limit of weights and biases is worth studying.

Continuum Limit

- Here is a typical example of taking continuum limit.

[H. Goldstein, C. Poole, J. Safko (2002). Classical Mechanics, Pearson.]



$$L = \frac{1}{2}\sum_i [m\dot{\eta}_i^2 - k(\eta_{i+1} - \eta_i)^2] \quad \Rightarrow \quad L = \frac{1}{2}\int \left[\mu\dot{\eta}^2 - Y\left(\frac{d\eta}{dx}\right)^2\right] dx$$

- This example gives a local Lagrangian because every term involves only variables with the same index.

Future Directions

- The stochasticity is an important component to train neural network.
- It is implemented by the time dependent training algorithm such as the stochastic gradient descent.
- This time dependence would be interpreted as the time dependent sources.
- Investigating this possibility would further enrich the study of synaptic field theory.