Applications of Deep Learning in Relativistic Hydrodynamics

Huichao Song

Peking University

Second international workshop on Collectivity in Small Collision Systems (CSCS 2018)

Hengfeng Huang, Bowen Xiao, Huixin Xiong, Zeming Wu, Yadong Mu and Huichao Song, arXiv:1801.03334

June. 15, 2018
**What is deep learning?**

**AI**: the broadest term, applying to any technique that enables computers to mimic human intelligence.

**ML**: A subset of AI aiming at optimizing a performance criterion using example data or past experience, but without explicit instruction.

**DL**: A subset of ML aiming at understanding high-level representations of data using a deeper structure of multiple processing layers.
AlphaGo (by Google DeepMind) beat human champion, March, 2016


Language processing

- **Machine translation**

- **Chinese poetry generation**

- **Speech recognition**
  W. Xiong et al., IEEE/ACM Transactions on Audio Speech & Language Processing, 2016

---

(e) Multilingual

| Multi src | Bei der Metropolitního výboru pro dopravu für das Gebiet der San Francisco Bay erklärten Beamte, der Kongress könne das Problem банкротство доверительного Фонда строительства шоссейных дорог einfach durch Erhöhung der Kraftstoffsteuer lösen |
| EN ref | At the Metropolitan Transportation Commission in the San Francisco Bay Area, officials say Congress could very simply deal with the bankrupt Highway Trust Fund by raising gas taxes |
| bpe2char | During the Metropolitan Committee on Transport for the San Francisco Bay, officials declared that Congress could solve the problem of bankruptcy by increasing the fuel tax bankrupt |
| char2char | At the Metropolitan Committee on Transport for the territory of San Francisco Bay, officials explained that the Congress could simply solve the problem of the bankruptcy of the Road Construction Fund by increasing the fuel tax |
Broad Applications of Deep Learning

Computer vision

-Image style transition:

-Image generation:
A. van den Oord et al., NIPS, (2016), arXiv: 1606.05328
Broad Applications of Deep Learning

- Playing Games:

- Autonomous Driving

[Diagram of Mediated Perception and Behavior Reflex]
Categories of deep learning

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Supervised learning:
Training on a dataset contains many features and associated with a label or target.
Unsupervised learning
-experience a dataset contains many features but **without labels**, and learn useful properties of the structure of this dataset.
An example of Reinforcement Learning

- play games

Reinforcement learning concern with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.
Deep Neural Network
-- more details
Deep learning tutorial for non-experts

Somewhat similar to look for a super complex function

- Define a set of functions
- Evaluate each function
- Pick up the best

- Speech Recognition
  \[ f(\text{audio signal}) = \text{“How are you”} \]

- Image Recognition
  \[ f(\text{image}) = \text{“Cat”} \]

- Dialogue System
  \[ f(\text{“Hi” (what the user said)}) = \text{“Hello” (system response)} \]

---

H. Lee, Deep Learning Tutorial,
https://www.slideshare.net/tw_dsconf/ss-62245351
In fact, there is not a “function” but to build up a (deep) neural network with huge tunable parameters to connect the inputs and outputs.
Neuron

Linear operation

\[ z_j = \sum_{i=1}^{N} x_i w_{i,j} + b_j \]

scaling, rotating, boosting, changing dimensions

Non-linear activation function

\[ h_j = \sigma(z_j) \]

(a) Sigmoid
\[ \sigma(z) = \frac{1}{1 + \exp(-z)} \]

(b) ReLU
\[ \sigma(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases} \]

(c) PReLU
\[ \sigma(z) = \begin{cases} z, & z > 0 \\ az, & z \leq 0 \end{cases} \]
Deep Neural network-loss function

Loss function:
- To evaluate the difference between the network’s outputs and learning targets.
  \[
  \ell(\theta) = \frac{1}{2n} \sum_x [y(x) - \hat{y}(x)]^2
  \]
  \[
  \ell(\theta) = -\frac{1}{n} \sum_x [y(x) \ln \hat{y}(x) - (1 - y(x)) \ln (1 - \hat{y}(x))]
  \]

\( \hat{y} \) network prediction
\( y \) true answer
Deep neural network - back propagation & gradient decent

- Deep neural network can reduce fitting error by updating model parameters through back propagation and gradient decent.

\[ \theta = \theta - \epsilon \frac{\partial L}{\partial \theta} \]

\[ L = \sum_i (\hat{y}_i - y_i)^2 \]
Common Network Structures

**Fully Connected Network**
- recognize handwritten digits
- 1000x1000 image
- 1M hidden units
- $10^{12}$ parameters!!!
- Spatial correlation is local
- Better to put resources elsewhere!

**Convolutional Neural Network**
- image recognition
- image classification
- Example: 1000x1000 image
- 1M hidden units
- Filter size: 10x10
- 100M parameters

**Recurrent Neural Network**
- speech recognition
- Example: 1000x1000 image
- $10^{12}$ parameters
- Spatial correlation is local
- Better to put resources elsewhere!
Applications of Deep Learning in Physics
Why Deep Learning in Physics?

“Unlike earlier attempts … Deep Learning systems can see patterns and spot anomalies in data sets far larger and messier than human beings can cope with.”

Can “Black-box” models learn patterns and models solely from data without relying on scientific knowledge?
Applications of Deep Learning in Physics

- Pierre Baldi, Peter Sadowski, and Daniel Whiteson, Nature Commun. 5 (2014) 4308
- Long-Gang Pang et al., Nature Commun. 9 (2018) no.1, 210
- ...
Searching for Exotic Particles in High-Energy Physics

Motivation:

- Finding the rare particles in high-energy particle colliders requires a successful distinguish of the signal from the huge & messy background.

- Traditional ‘shallow’ machine learning models that have a limited capacity for such task.

Searching for Exotic Particles in High-Energy Physics

A) Generating training/testing data
MadGraph (collisions) +PYTHIA (showering & hadronization) +DELPHES (detector response)

B) Training DNN/NN: supervised learning with 11 million data (low level / high level)

C) Testing DNN/NN
-DNN improves AUC by 8% compared to NN

Deep learning can improve the power for the collider search of exotic particles

Motivation:
- Traditionally, the study of phases transition of condense matter systems is to calculate the associated order parameter, measure specific heat, etc.
- Can deep learning identify phases and phase transitions?

Classifying the Phase of Ising Model

A) Generating training/testing data:

traditional MC method from the Boltzmann distribution

\[ p(\sigma_1, \sigma_2, ..., \sigma_N) = \frac{e^{-\beta E(\sigma_1, \sigma_2, ..., \sigma_N)}}{Z(\beta)} \]

B) training fully connected network:

with these raw configurations of square-lattice Ising-model

\[ E = -J \sum_{\langle i, j \rangle} \sigma_i \sigma_j \]

C) testing the trained network

-The trained fully connected network also past the test of triangular-lattice Ising model, showing its abilities of generalize to task beyond their original design

Classifying the Phase of Ising Model

A) **Training/testing data:**
traditional MC method from the Boltzmann distribution

\[ p(\sigma_1, \sigma_2, ..., \sigma_N) = \frac{e^{-\beta E(\sigma_1, \sigma_2, ..., \sigma_N)}}{Z(\beta)} \]

B) **training fully connected network:**
with these raw configurations of square-lattice Ising-model

\[ E = -J \sum_{\langle i, j \rangle} \sigma_i \sigma_j \]

Further test- Ising gauge theory

\[ H = -J \sum_p \prod_{i \in p} \sigma_i^z \]

Fully connected network fails (50% accuracy) - equal to simply guessing

No free lunch theorem[1]

No machine learning algorithm is consistently better than another. In other words, there is no silver bullet, deep learning and neural networks not exempted. In fact, the most universal feedforward neural network does worse than tree based methods or SVM on many problems. Therefore, when adapting a model to new problems, one should be aware of model assumptions and ensures that they still holds.


Classifying the Phase of Ising Model

A) **Training/testing data:**

traditional MC method from the Boltzmann distribution $\beta F(\sigma, \sigma_i)$.

B)  training fully connected network:

with these raw configurations of square lattice Ising model.

Classifying the Phase of Ising Model

For the case of Ising gauge theory

\[ H = -J \sum_p \prod_{i \in p} \sigma_i^z \]


The trained CNN discriminates high-temperature from ground States with very high accuracy in spite of the lack of an order Parameter or qualitative differences in the spin-spin correlations.

Neutral network can be used to encode phases of matter and discriminate phase transitions in correlated many-body systems.
Identify QCD Phase Transition with Deep Learning

Motivation:
- Traditionally, the properties of the QCD matter are extracted from the event averaged observables
- Can deep learning identify different EoS from the raw data of heavy ion collisions?

A) Generating training/testing data:
- Run Hydro with EOS L and EOS Q
- Particle spectra - image (15*48 pixels)

\[ \rho(p_T, \phi) \equiv \frac{dN_i}{dY_{PT}dpTd\phi} = g_i \int d\sigma f_i, \]

B) Training CNN

Table 1: The training data set Hydro CLVis (AMPT)

<table>
<thead>
<tr>
<th>Training data set</th>
<th>$\eta/s = 0$</th>
<th>$\eta/s = 0.08$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EOSL</td>
<td>EOSQ</td>
</tr>
<tr>
<td>Au-Au $\sqrt{s_{NN}} = 200$ GeV</td>
<td>7435</td>
<td>5328</td>
</tr>
<tr>
<td>Pb-Pb $\sqrt{s_{NN}} = 2.76$ TeV</td>
<td>4967</td>
<td>2828</td>
</tr>
</tbody>
</table>

C) Testing the trained network

One can efficiently decode the EOS information from the complex final particle info event by event using deep learning.

More Comments on several examples of supervised learning

**Image classification**

- Dog or Cat? Yes or No?
Deep learning can do more ...

**Image classification**

Dog or Cat?

**Image generation**

A. van den Oord et al., NIPS, (2016), arXiv: 1606.05328
For the non-linear hydro system, can the “Black-box” network learn patterns solely from data without relying on scientific knowledge? (conservation laws)

For hydrodynamics, can we use deep learning to learn/predict the pattern transform between initial and final profiles?

- Initial energy density profiles
  -------- > final energy density velocity profiles

- Initial energy density profiles
  -------- > final particle profiles

- Final particle profiles
  -------- > Initial energy density profiles
Applications of deep learning to relativistic hydrodynamics

H.Huang, B.Xiao, H.Xiong, Z.Wu, Y. Mu and H.Song

arXiv: 1801.03334
Traditional hydrodynamics

\[ \partial_\mu T^{\mu \nu}(x) = 0 \]

Deep Learning

training

self-learning

testing

well-trained

-Such deep learning systems do not need to be programmed with the hydro equation \( \partial_\mu T^{\mu \nu}(x) = 0 \). Instead, they learn on their own.
Deep Learning

Step 1) Generate the training/testing data sets from hydro (VISH2+1)
Initial & final energy momentum tensor profiles ----> initial & final image sets

Step 2) Design & train the deep neural network
Training sets: initial & final profiles from hydro with MC-Glauber initial conditions

Step 3) Test the deep neural network
Compare DNN predictions with hydro results for different testing initial conditions
(MC-Glauber, MC-KLN, AMPT Trento)
Deep Learning for 1+1-hydro

1) Generate the training/testing data sets from 1+1-d hydro

\[ T^{\tau \tau},_x + (\bar{v}_x T^{\tau \tau}),_x = -\frac{p + T^{\tau \tau}}{\tau} - (p \bar{v}_x)_x, \]

\[ T^{\tau x},_x + (\bar{v}_x T^{\tau x}),_x = -p_x - \frac{T^{\tau x}}{\tau} \]

EoS: \( p = e/3 \), MC-Glauber/MC-KLN initial conditions \( \tau - \tau_0 = 2.0, 4.0, 6.0 \) fm/c

2) Design / train neural network (CNN)

![Neural Network Diagram]

Loss Function \( L(\theta) = \frac{1}{2} \sum \frac{(\hat{y} - y)^2}{\max\{y\}} \)

Huang & Song unpublished notes
Deep Learning for 1+1-hydro

3 ) Testing the neural network

-CNN predictions vs. hydro simulation: CNN works for 1+1-d hydrodynamics
-A first hint that a well designed network could capture the non-linear evolution of hydrodynamics

Huang & Song unpublished notes
Deep Learning (CNN): an extension to 2+1-hydro

For 2+1-d hydro, CNN does not work, so does not other common network, such as local connected layer.

From 1+1-d hydro to 2+1-d hydro, the pixel of the imagine sets increased from 200 to 40000 (200*200).
Deep Learning (CNN): an extension to 2+1-hydro

No free lunch theorem\[^1\]

No machine learning algorithm is consistently better than another. In other words, there is no silver bullet, deep learning and neural networks not exempted. In fact, the most universal feedforward neural network does worse than tree based methods or SVM on many problems. Therefore, when adapting a model to new problems, one should be aware of model assumptions and ensures that they still holds.

Stacked U-net for 2+1-d hydro

The activation function:

\[ \text{Leaky ReLU } f(x) = \max \{x, 0.03x\} \]

The loss function:

normalized MAE loss \[ \text{Loss} = \frac{|y_1 - y_0|}{|y_0|} \]

Training / Testing data sets from 2+1-d hydro

\[ T^{\tau \tau},_\tau + (\bar{v}_x T^{\tau \tau}),_x + (\bar{v}_y T^{\tau \tau}),_y = -\frac{p+T^{\tau \tau}}{\tau} - (p \bar{v}_x)_x - (p \bar{v}_y)_y \]

\[ T^{\tau x},_\tau + (\bar{v}_x T^{\tau x}),_x + (\bar{v}_y T^{\tau x}),_y = -p_x - \frac{T^{\tau x}}{\tau} \]

\[ T^{\tau y},_\tau + (\bar{v}_x T^{\tau y}),_x + (\bar{v}_y T^{\tau y}),_y = -p_y - \frac{T^{\tau y}}{\tau} \]

Initial conditions: MC-Glauber, MC-KLN, AMPT, Trento  
EoS: p=e/3, hydro evolution time: \( \tau - \tau_0 = 2.0, 4.0, 6.0 \text{ fm/c} \)

The Training Data Sets

<table>
<thead>
<tr>
<th>2+1-d hydro</th>
<th>10000 events</th>
</tr>
</thead>
<tbody>
<tr>
<td>VISH 2+1</td>
<td>MC-Glauber</td>
</tr>
</tbody>
</table>

The Testing Data Sets

<table>
<thead>
<tr>
<th>2+1-d hydro</th>
<th>MC-Glauber</th>
<th>MC-KLN</th>
<th>AMPT</th>
<th>Trento</th>
</tr>
</thead>
<tbody>
<tr>
<td>VISH 2+1</td>
<td>10000 events</td>
<td>10000 events</td>
<td>10000 events</td>
<td>10000 events</td>
</tr>
</tbody>
</table>

Predictions: Stacked U-net vs. CNN

Initial condition

Hydro results

\( \tau_0 = 0.6 \text{ fm/c} \)

\( \tau = 2.6 \text{ fm/c} \)

CNN prediction

\( \tau = 2.6 \text{ fm/c} \)

Initial condition

Hydro results

sUnet predictions

\( \tau_0 = 0.6 \text{ fm/c} \)

\( \tau = 2.6 \text{ fm/c} \)

-sUnet is the proper directions that works
Due to the size of GPU memory, it is costly to increase the number of U-net, so we divide the whole evolution time $\tau_0 - \tau$ into 3 parts with equal time interval $\Delta \tau$: $0.6 \text{fm/c} - 2.6 \text{fm/c}$, $2.6 \text{fm/c} - 4.6 \text{fm/c}$, $4.6 \text{fm/c} - 6.6 \text{fm/c}$. 

**Combined sUnet for long time evolution**
sUnet prediction vs. hydro simulations

\[ \tau - \tau_0 = 2.0 \text{fm/GeV} \]
sUnet prediction vs. hydro simulations

\[ \tau - \tau_0 = 2.0 \text{fm/c} \]

For a closer look
sUnet prediction vs. hydro simulations

\[ \tau - \tau_0 = 4.0 \text{fm/c} \]

-for a closer look
sUnet prediction vs. hydro simulations

\[ \tau - \tau_0 = 6.0 \text{fm}/c \]
sUnet prediction vs. hydro simulations

\[ \tau - \tau_0 = 6.0 \text{fm}/c \]

-for a closer look
Eccentricity distributions:

\[
\mathcal{E}_n e^{i n \Phi_n} = -\frac{\int dx dy r^2 e^{i n \phi} e(x,y)}{\int dx dy r^2 e(x,y)}
\]
sUnet prediction vs. hydro simulations

\[ \varepsilon_n e^{i n \Phi_n} = -\frac{\int dxdy r^2 e^{i n \phi} e(x,y)}{\int dxdy r^2 e(x,y)} \]

Histograms of \( \varepsilon_n \)
Summary & outlook
Traditional hydrodynamics

\[ \partial_{\mu} T^{\mu \nu} = 0 \]

Deep Learning

training

self-learning

testing

well-trained
For hydrodynamics can we use deep learning to learn/predict the pattern transform between initial and final profiles?

Initial energy density profiles
-------- > final energy density velocity profiles

Summary

-Using 10000 initial and final profiles generated from VISH2+1 with MC-Glauber initial condition, we train the network called sU-net.
-We use the well-train network to predict the final profiles from with different initial conditions, including MC-Glauber, TRENTo, AMPT and MC-KLN.
-Out results show that deep learning can predict the magnitude and inhomogeneous structures of the final energy density and flow velocity, which can also describe the related eccentricity distribution $P(\gamma)$.
-Deep learning can capture the main features of the non-linear evolution of hydrodynamics.
Outlook

For hydrodynamics, can we use deep learning to learn/predict the pattern transform between initial and final profiles?

Initial energy density profiles
--------- > final energy density velocity profiles

Initial energy density profiles
--------- > final particle profiles

Final particle profiles
--------- > Initial energy density profiles

For flow in large and small systems
open for discussions
Thank You
Fast automated analysis of strong gravitational lenses


- Inputing the lensed galaxy images, CNN can estimate lensing parameters in an extremely fast and automated way.
Unet & Stacked Unet

-Inspiration from biomedical image segmentation
