Application of AI/Deep learning in studying nuclear symmetry energy from HICs

Yongjia Wang and Qingfeng Li

Huzhou University

ECT*, May 20-25, 2019

Outline

Background

- Artificial Intelligence and symmetry energy

Data Zoo

- rapidity .vs. p_t data from UrQMD model

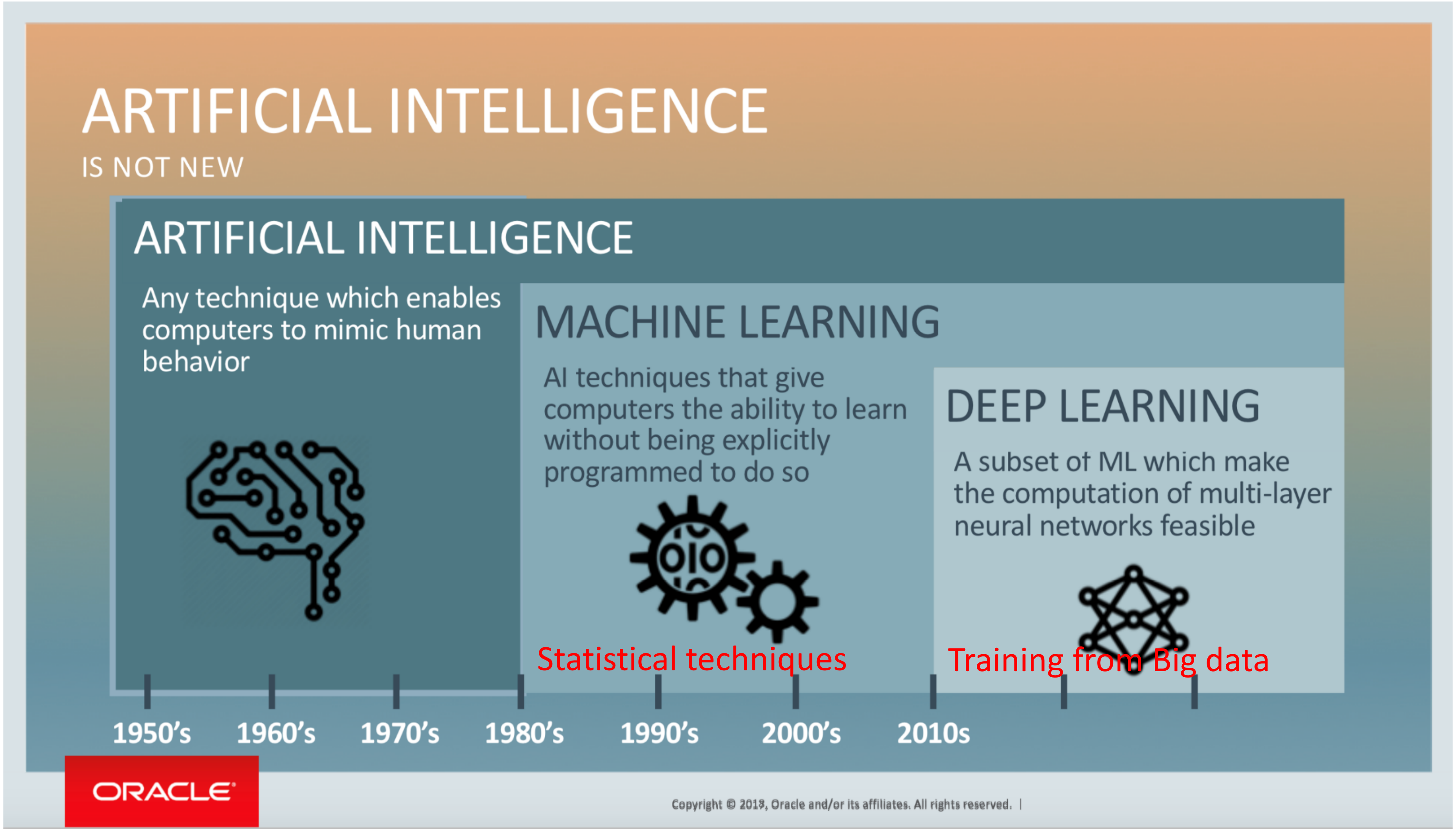
Deep learning model

- Convolutional Neural Network

Results and Outlook

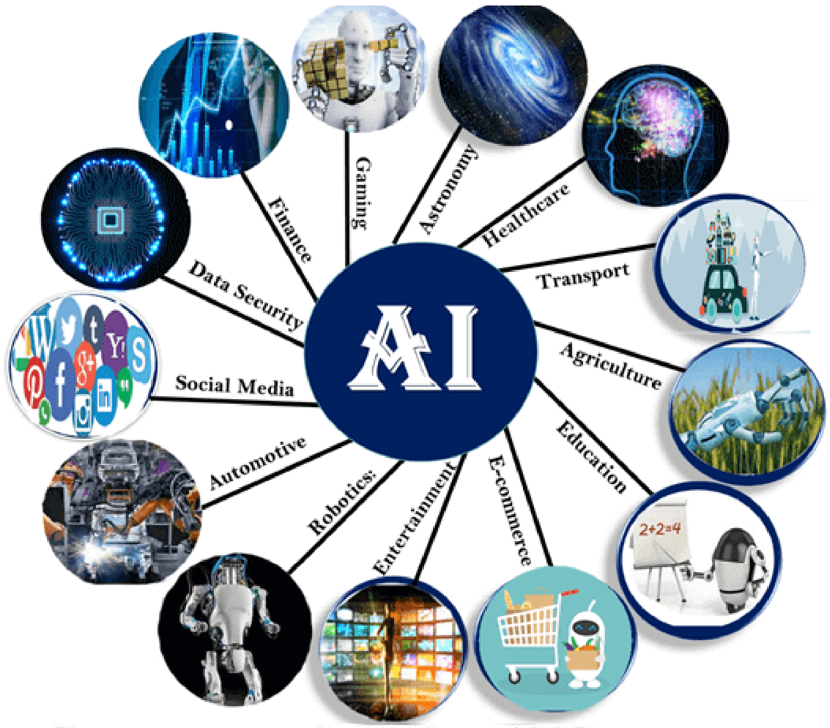
- Classification and regression

Artificial Intelligence

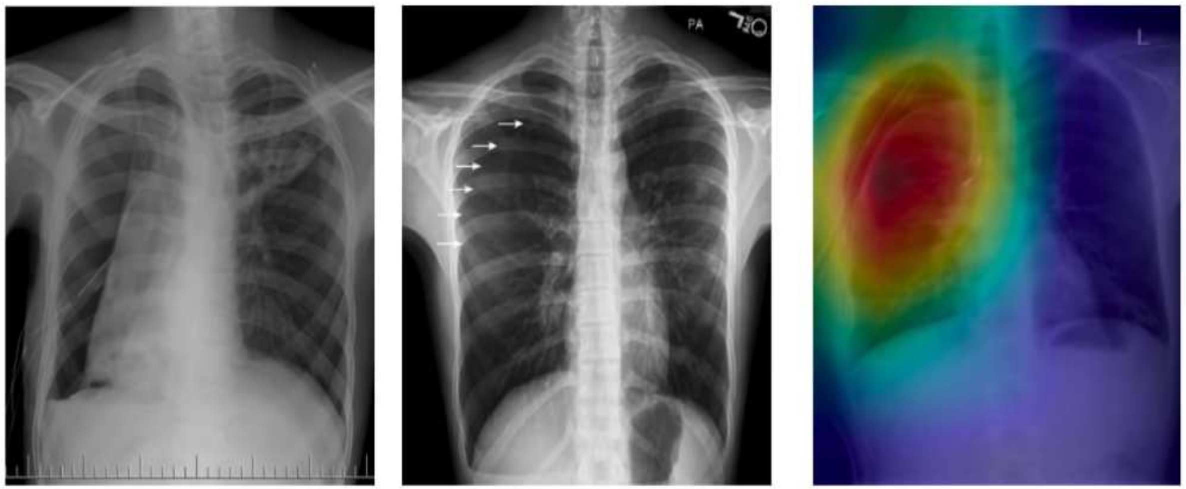


Application of Artificial Intelligence

1- Automated customer support
2- Healthcare
3- Finance
4- Smart cars and drones
5- Social media
6- Smart home devices
7- Creative arts
8- Security and surveillance
9- Education
10-Agriculture
……



Detect pneumothorax in real X-Ray scans



nature physics

Letter | Published: 13 February 2017

Machine learning phases of matter

Juan Carrasquilla & Roger G. Melko

Nature Physics **13**, 431–434 (2017) | Download Citation

nature COMMUNICATIONS

Article | OPEN | Published: 15 January 2018

An equation-of-state-meter of quantum chromodynamics transition from deep learning

Long-Gang Pang, Kai Zhou, Nan Su, Hannah Petersen, Horst Stöcker & Xin-Nian Wang

Nature Communications **9**, Article number: 210 (2018) | Download Citation

Science Home News Journals Topics Careers

SHARE

PERSPECTIVES

Machine learning for quantum physics

Michael R. Hush

See all authors and affiliations

Science 10 Feb 2017:
Vol. 355, Issue 6325, pp. 580
DOI: 10.1126/science.aam6564

nature > review articles > article

nature
International journal of science

Review Article | Published: 01 August 2018

Machine learning at the energy and intensity frontiers of particle physics

Alexander Radovic, Mike Williams, David Rousseau, Michael Kagan, Daniele Bonacorsi, Alexander Himmel, Adam Aurisano, Kazuhiro Terao & Taritree Wongjirad

Nature **560**, 41–48 (2018) | Download Citation

Application of Artificial Intelligence in physics

Two main class of problems we deal with

- Classification

 - Identify if an object belongs to one of N subgroups

 - Divide objects into distinct classes and find the discriminating feature(s)

 - Identify outliers / class of interest in a dataset

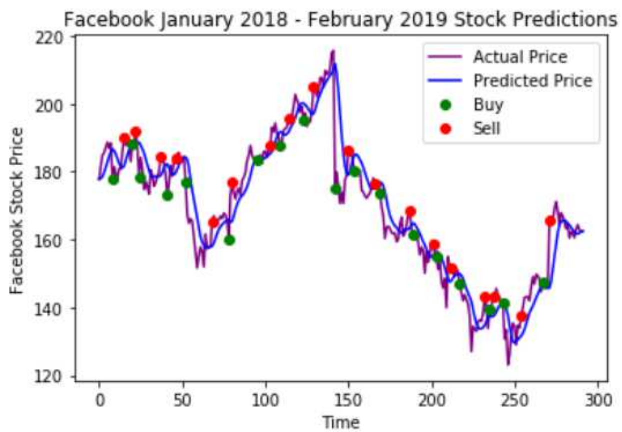
- Regression

 - Estimate the relation between observables and quantities of interest

 - Both parametric (eg. fitting a line to data) and nonparametric (eg. splining / kriging)

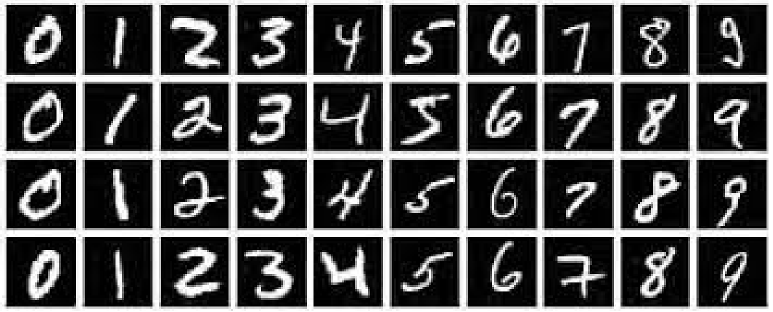
 - Interpolation and extrapolation

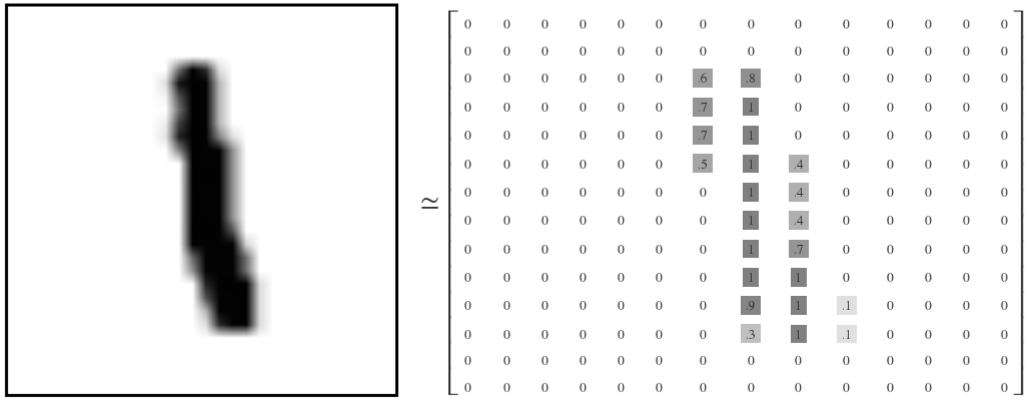
 - Prediction and forecasting.

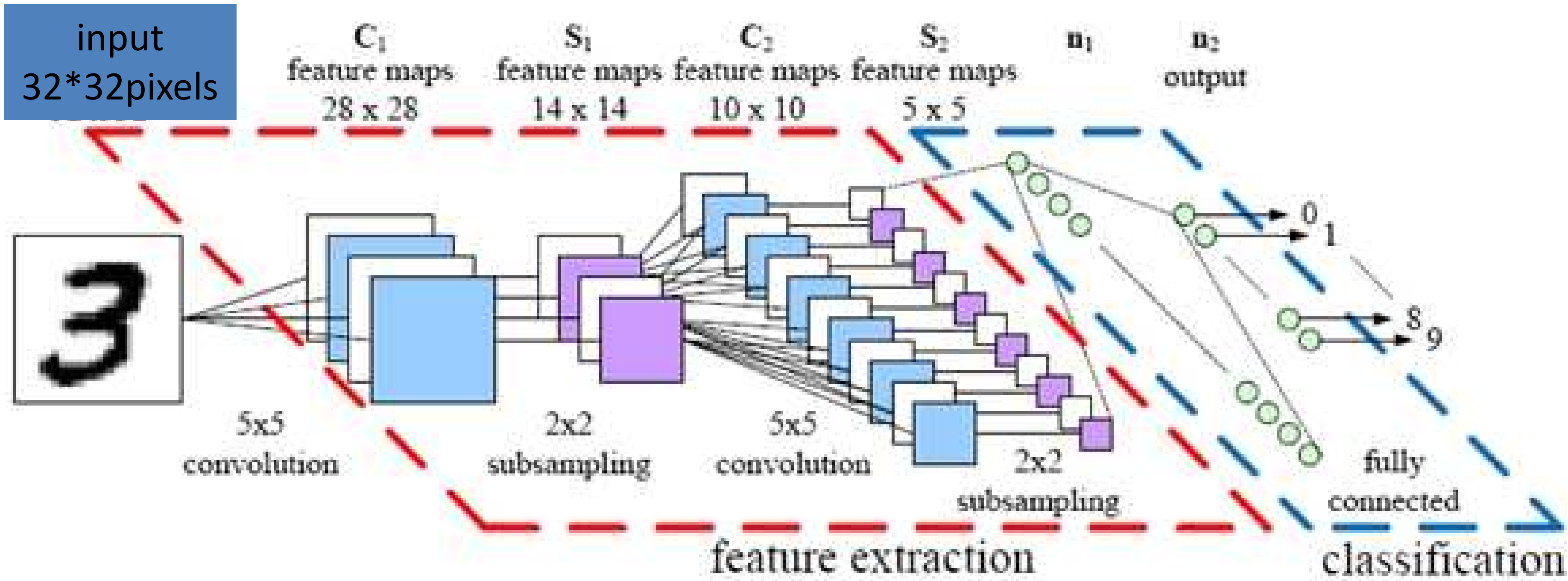


Example of handwritten digit recognition

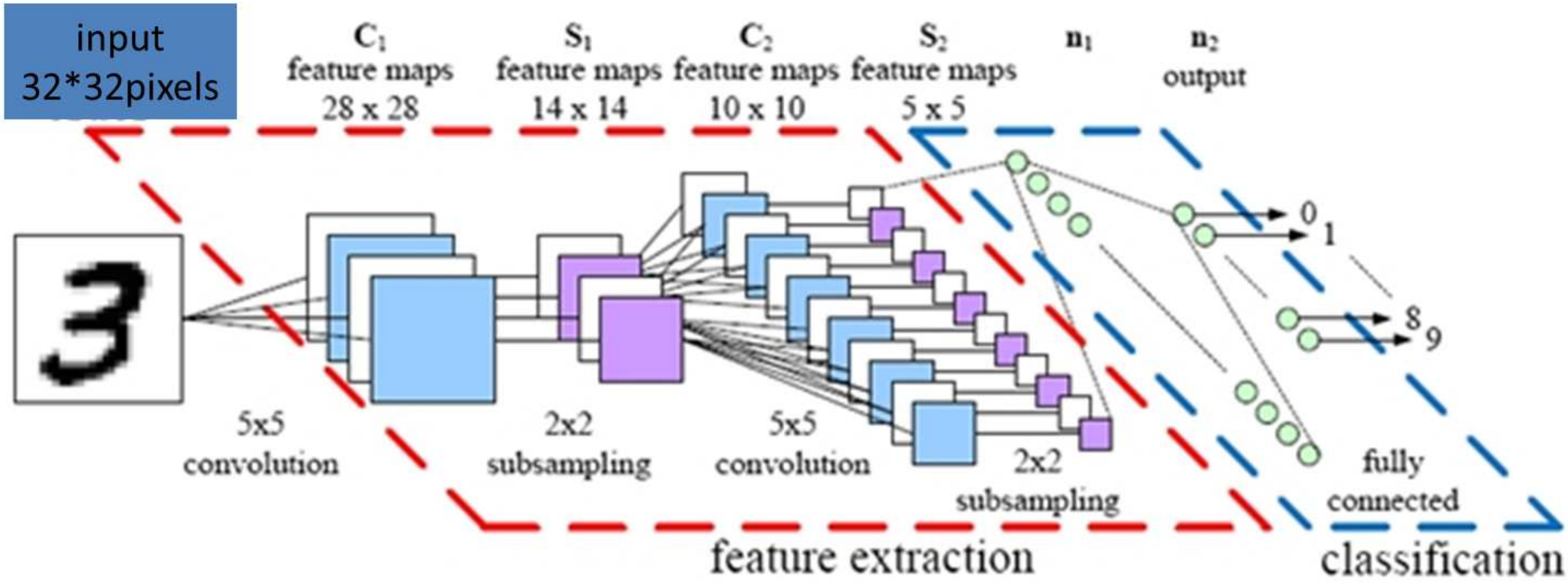
MNIST database

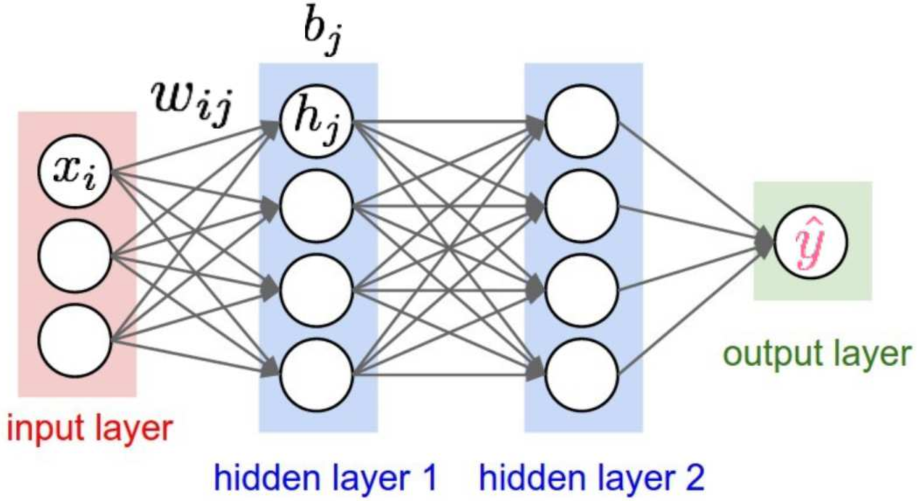






Example of handwritten digit recognition





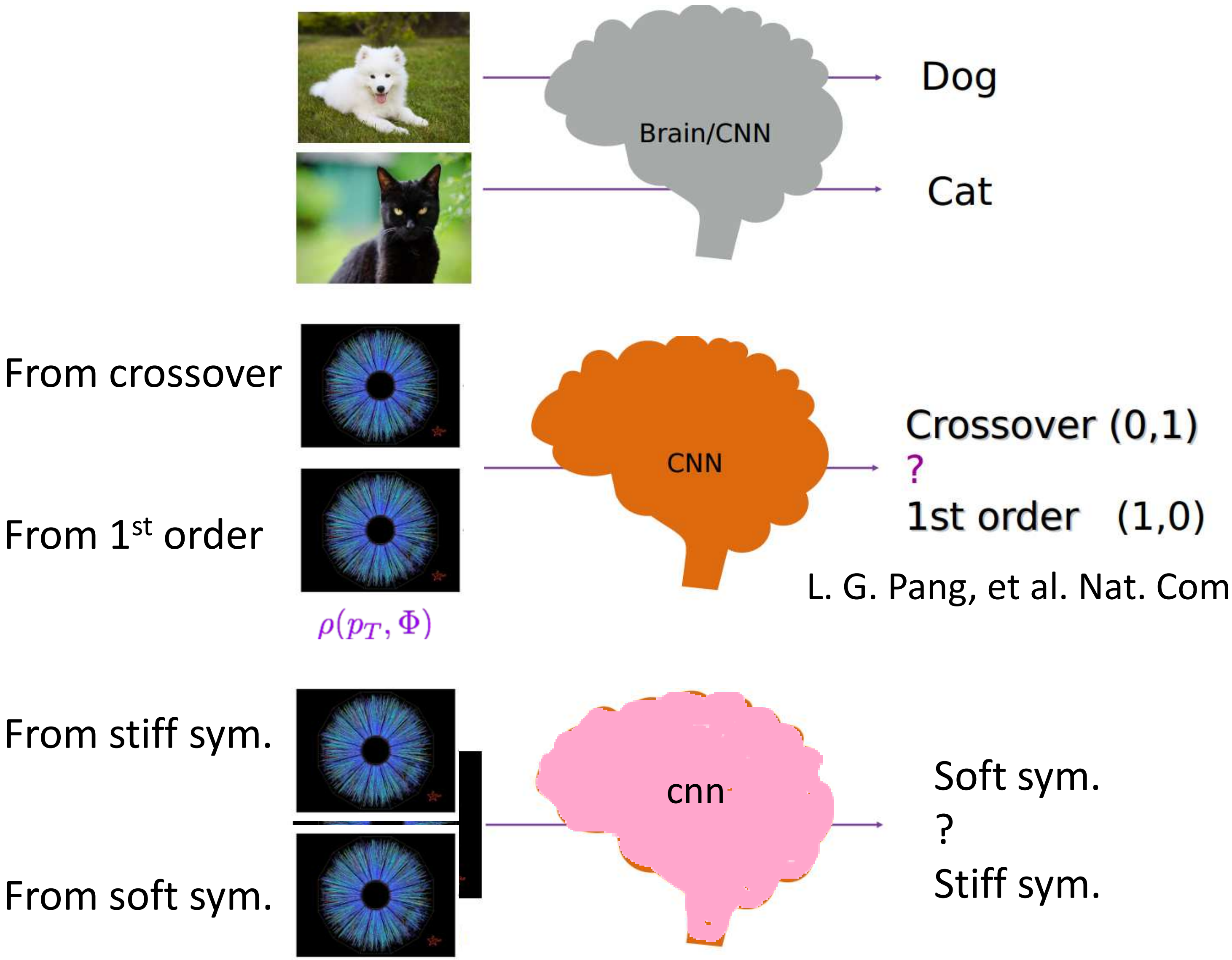
weights and biases

$$z_j = \sum_{i=1}^{N} x_i w_{ij} + b_j$$

$$h_j = \sigma(z_j)$$ Activation function

To find the particular values of weights and biases for which the loss is minimum.

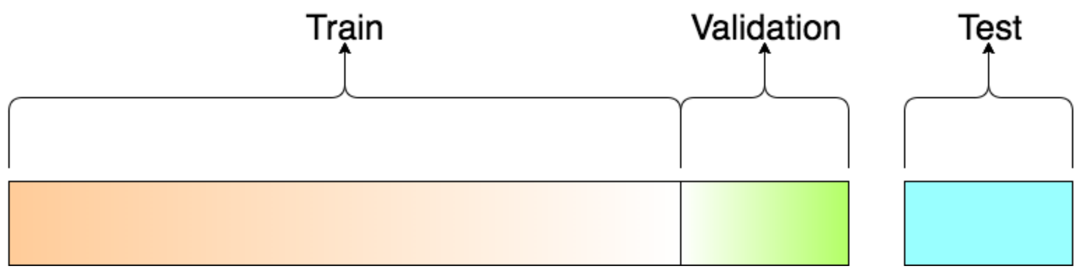
Can we use deep learning to study symmetry energy?



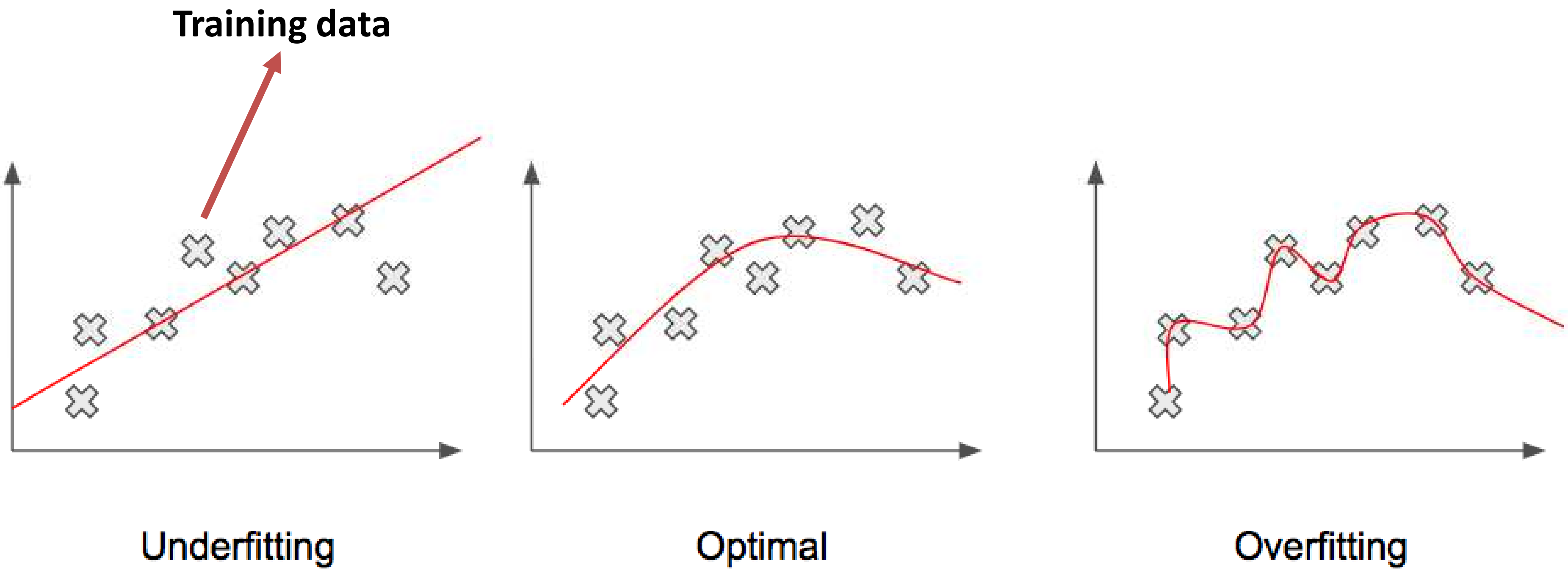
L. G. Pang, et al. Nat. Comm. 9: 210 (2018)

Three terms: [Training, Testing, Validation] database

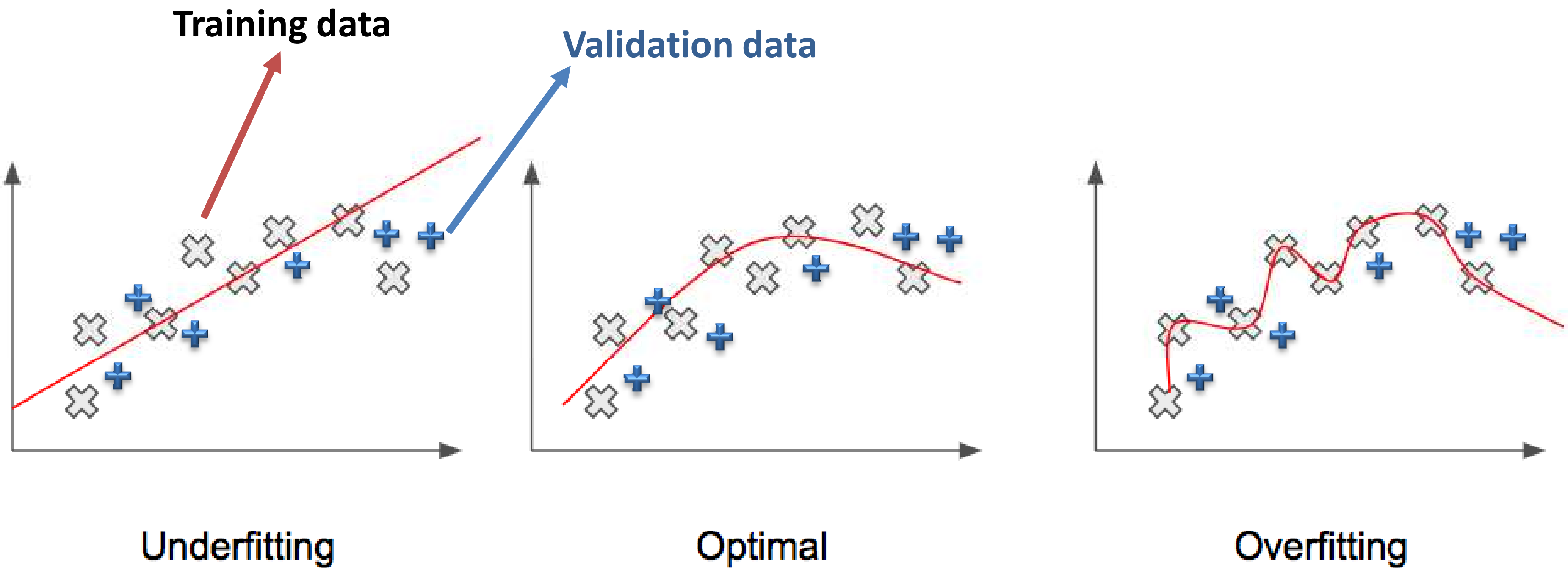
- Training - giving data to your method and letting it find a mapping between input and output variables.
- Validation - checking to see if this mapping still works when applied to data not in the training set.
- Testing - after the training is done, this last piece of data is used to check if the mapping we've got works - determines the predictive power.



Overfitting



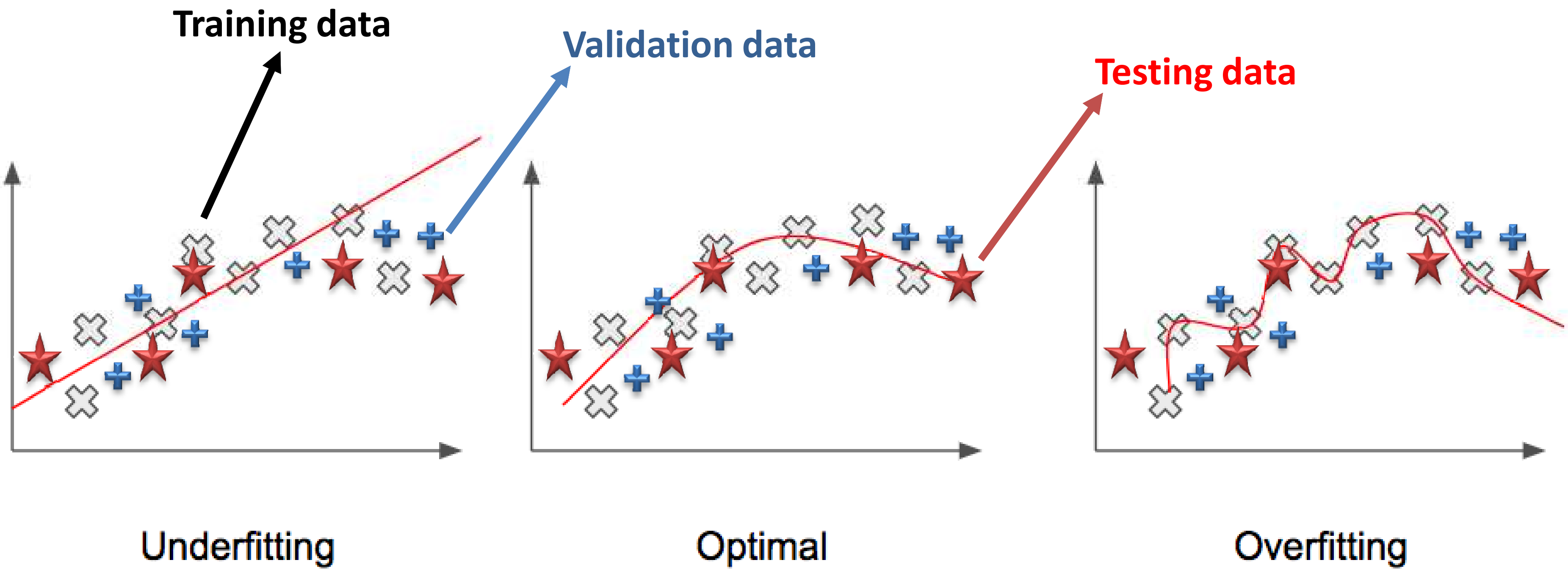
Overfitting



Mean squared error $$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i})^2$$

$*\, n$ is the number of data points
$*\, Y_i$ represents observed values
$*\, \hat{Y}_i$ represents predicted values

Overfitting

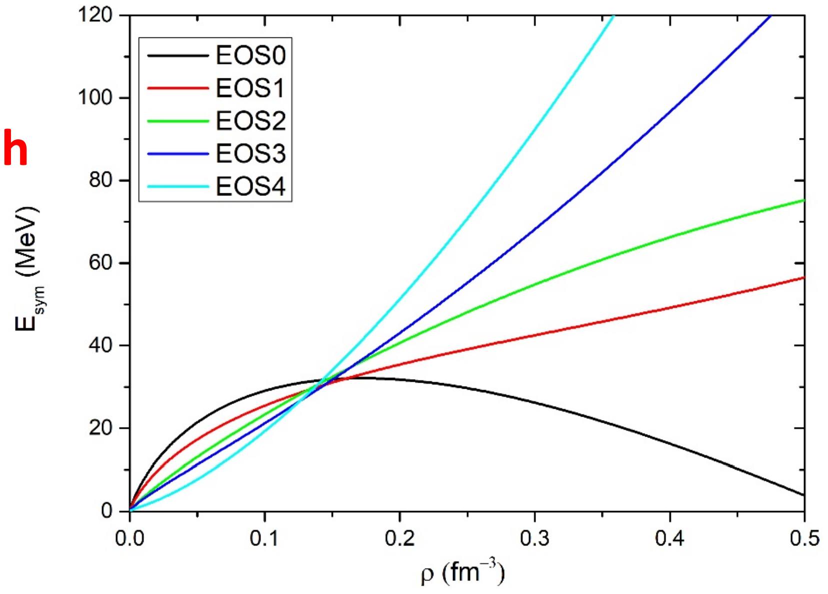


Mean squared error $$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i})^2$$

$*$ n is the number of data points
$*$ Y_i represents observed values
$*$ $\hat{Y_i}$ represents predicted values

Data Zoo

Au+Au, b=5 fm
E_{lab}=0.4 GeV/nucleon
800, 000 events for each



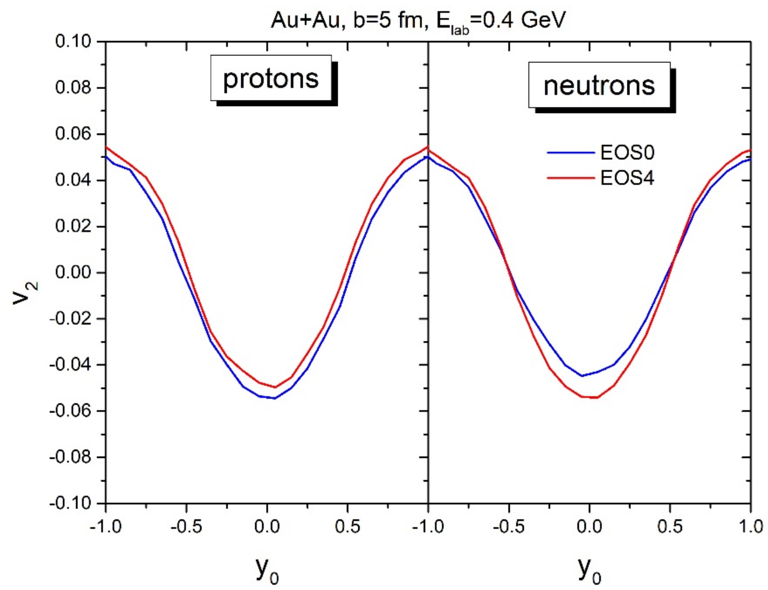
	Slope L (MeV)	Training	Validation	Testing
EOS0	5.8	60%	15%	25%
EOS1	44.3	60%	15%	25%
EOS2	81.2	60%	15%	25%
EOS3	106.4	60%	15%	25%
EOS4	159	60%	15%	25%

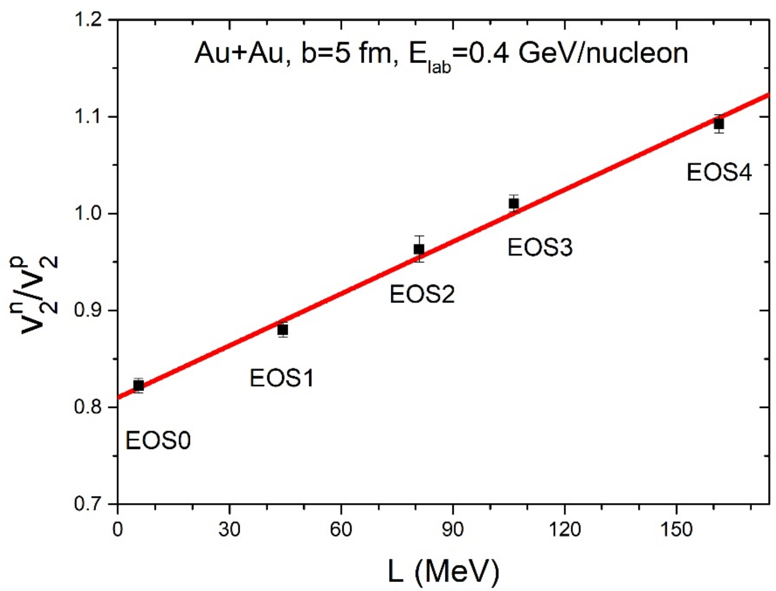
Au+Au, b=5 fm

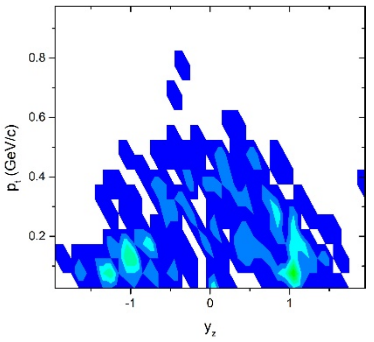
E_{lab}=0.4 GeV/nucleon

800, 000 events for each

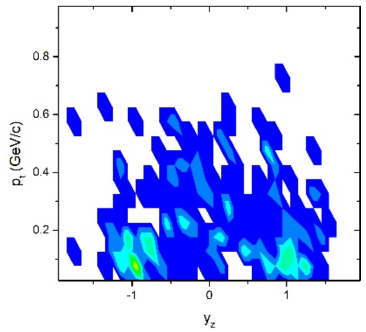
Elliptic flow



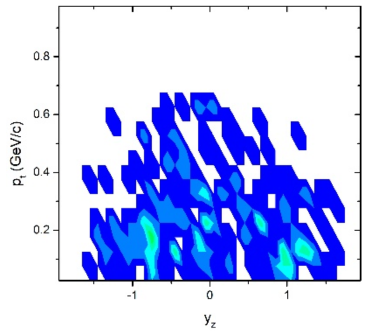




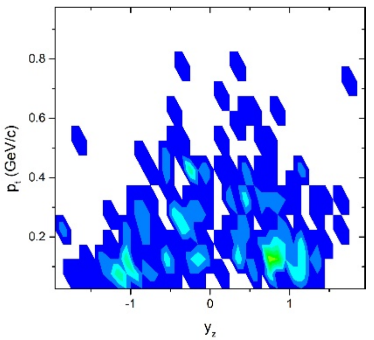
EOS0



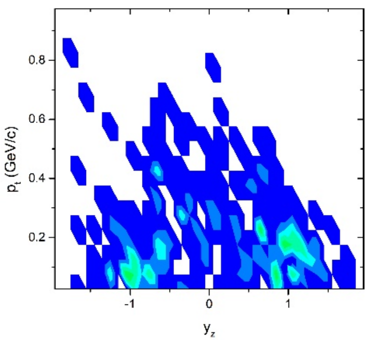
EOS1



EOS2



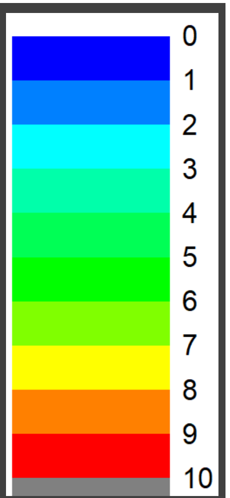
EOS3



EOS4

y_z: -2.0 to 2.0 with 40 bins

p_t: 0 to 1 GeV/c with 20 bins

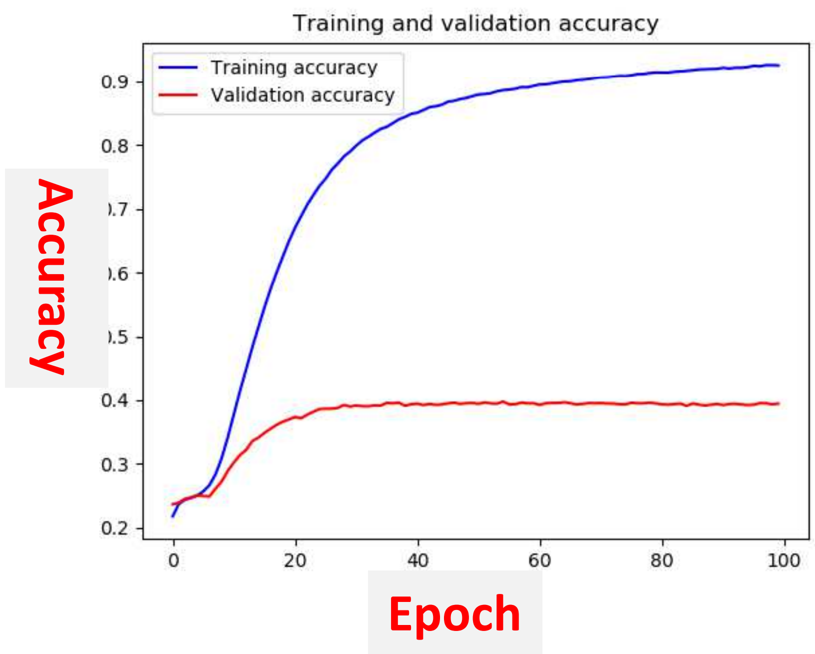


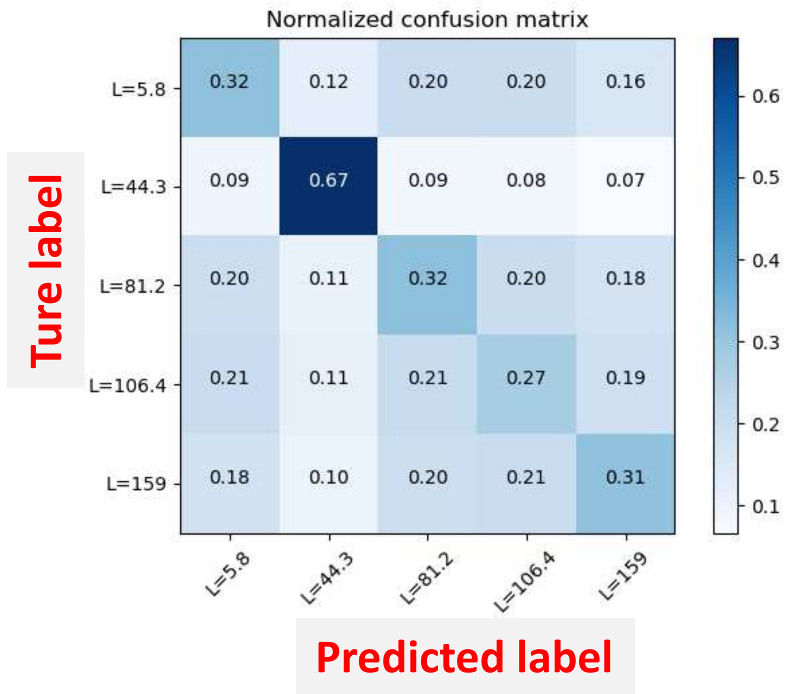
Data Zoo: y .vs. p_t of proton from 1 events

20*40=800 pixels

Data Zoo: from 1 events

Epoch: The number times that the learning algorithm will work through the entire training dataset.





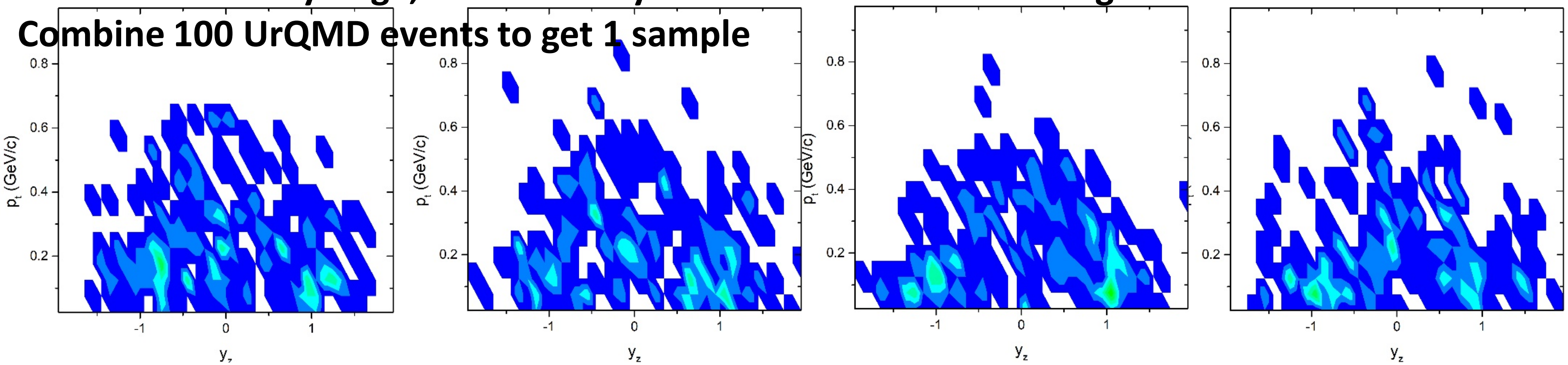
One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters.

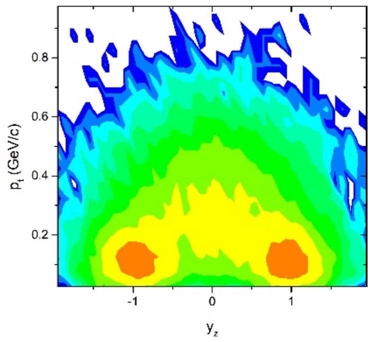
Data Zoo: from 1 events

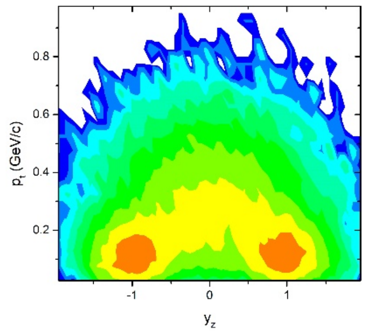
0	0	0	0	0	0	0	0	1	0	0	0	2	1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1	2	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	1	0	1	1	0	0	3	0	1	0	1	0	1	2	0	0	2	0	0	1	1	0	2	3	3	0	0	1	1	1	0	0	0	0
0	0	0	0	0	0	2	1	1	2	2	2	3	1	0	4	0	1	2	2	1	0	1	0	0	0	0	1	0	4	1	3	4	2	0	0	1	0	0	0
0	0	0	0	0	2	0	1	2	1	2	1	5	1	0	0	0	0	3	0	0	1	1	0	2	1	0	1	0	1	1	2	1	0	0	1	0	0	0	0
0	0	0	0	0	1	0	2	0	1	1	3	2	1	1	1	2	1	0	4	2	0	0	0	1	0	4	1	1	1	1	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	2	2	1	1	0	0	0	1	0	0	1	2	1	0	0	0	2	1	0	0	1	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	1	0	1	2	3	0	2	1	1	0	0	1	0	0	1	1	1	2	0	1	1	0	0	0	0
0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	2	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0	2	1	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	2	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

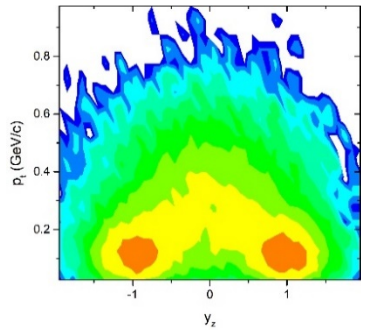
Fluctuation is very large, thus it is very different to determine weights and biases.

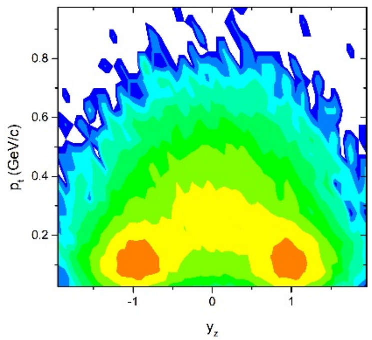
Combine 100 UrQMD events to get 1 sample

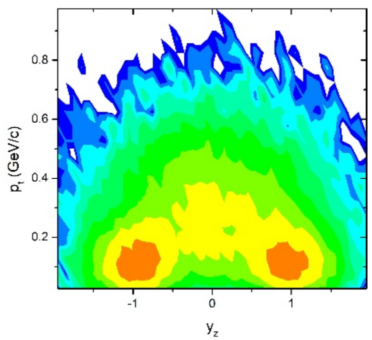


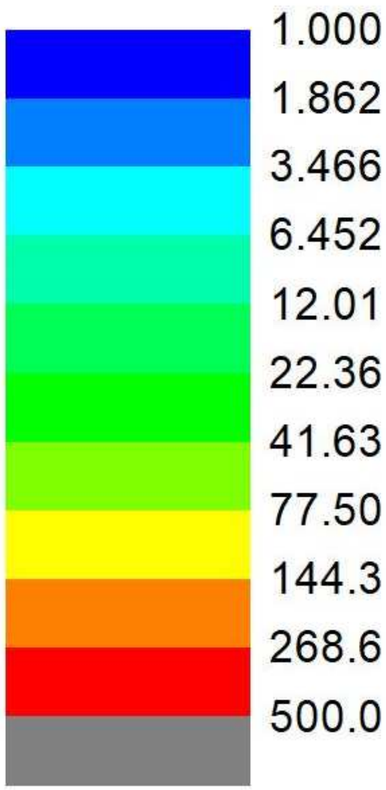




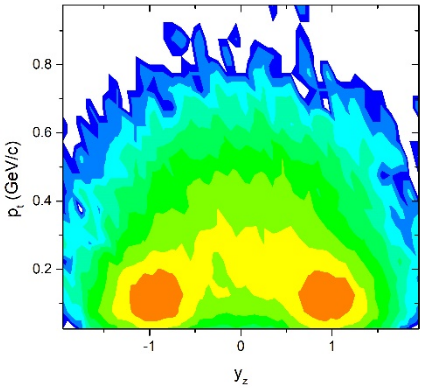




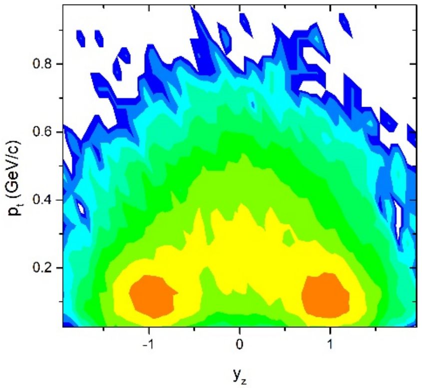




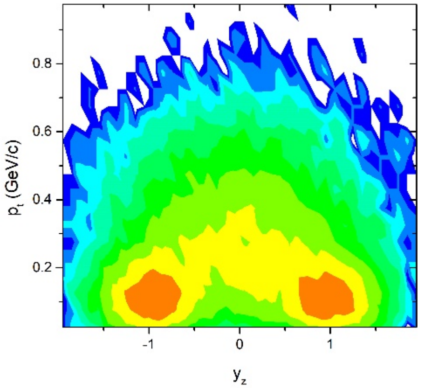
Data Zoo: 1 sample combine with 100 events



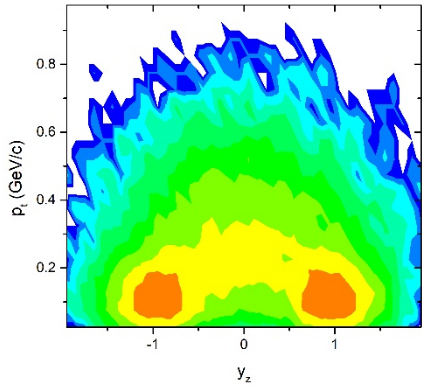
EOS4



EOS0



EOS0



EOS4

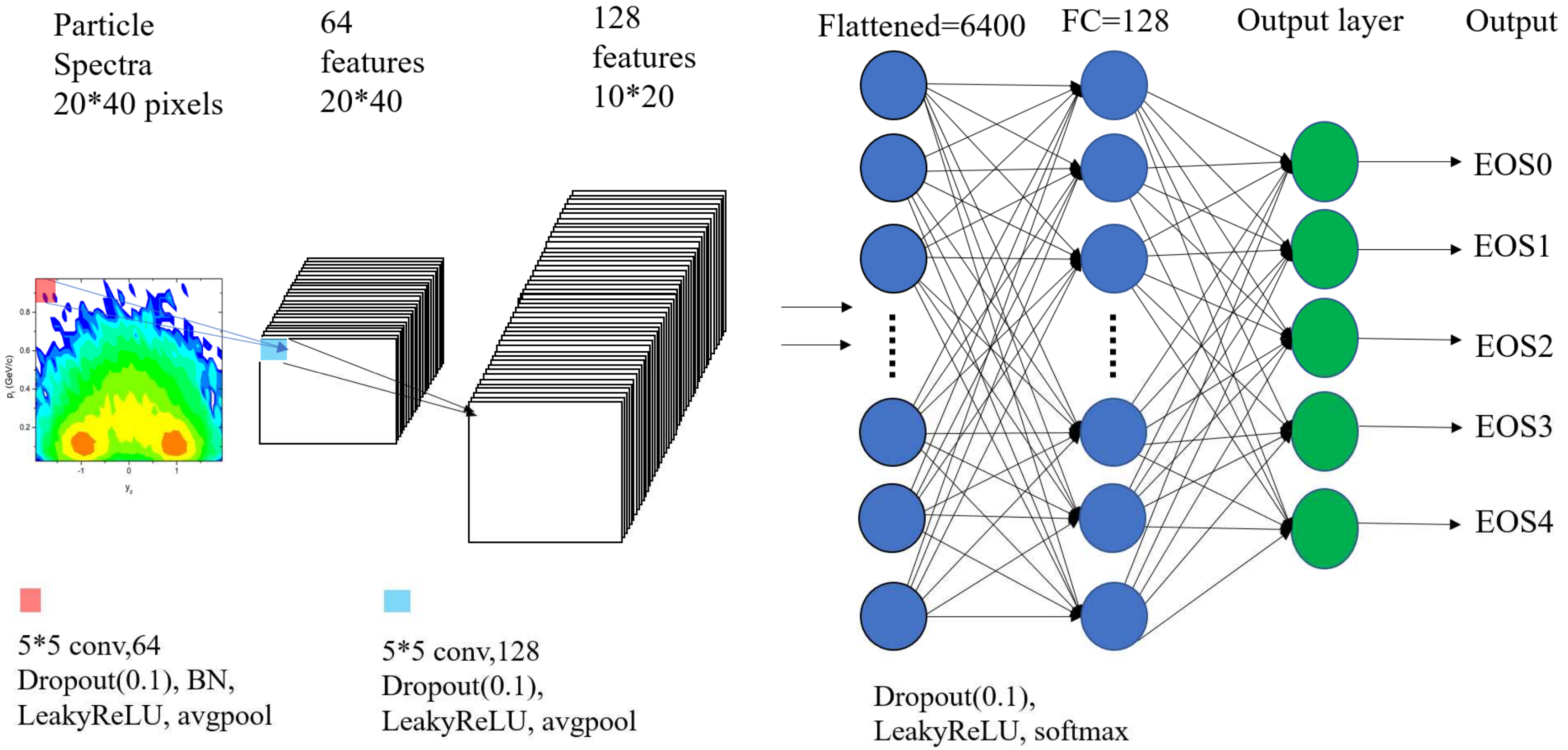
Comparison of EOS0 and EOS4

Data Zoo : y .vs. p_t for proton

Totally 40,000 samples

	Slope L (MeV)	Training	Validation	Testing
EOS0	5.8	4800	1200	2000
EOS1	44.3	4800	1200	2000
EOS2	81.2	4800	1200	2000
EOS3	106.4	4800	1200	2000
EOS4	159	4800	1200	2000

Convolution neural network (CNN) architecture



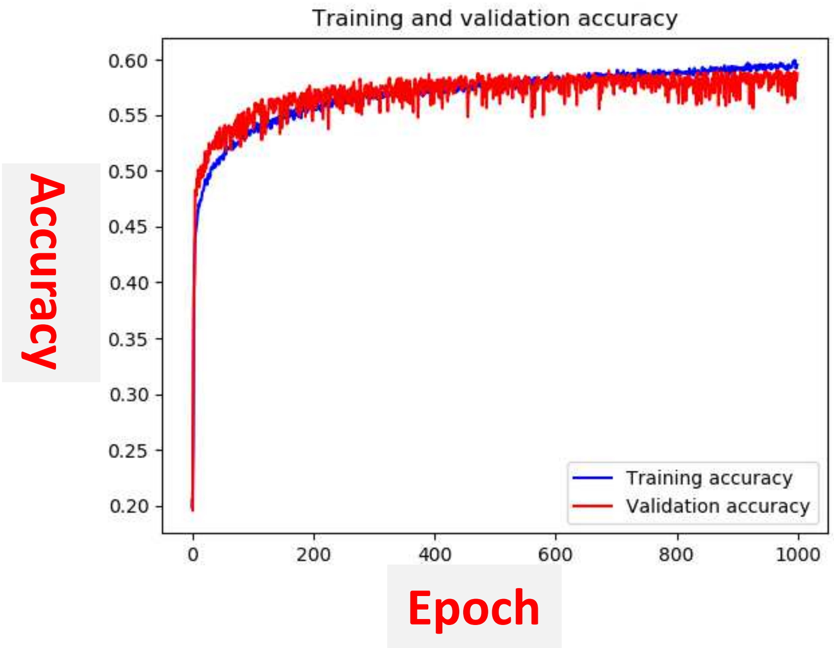
Two hidden layers

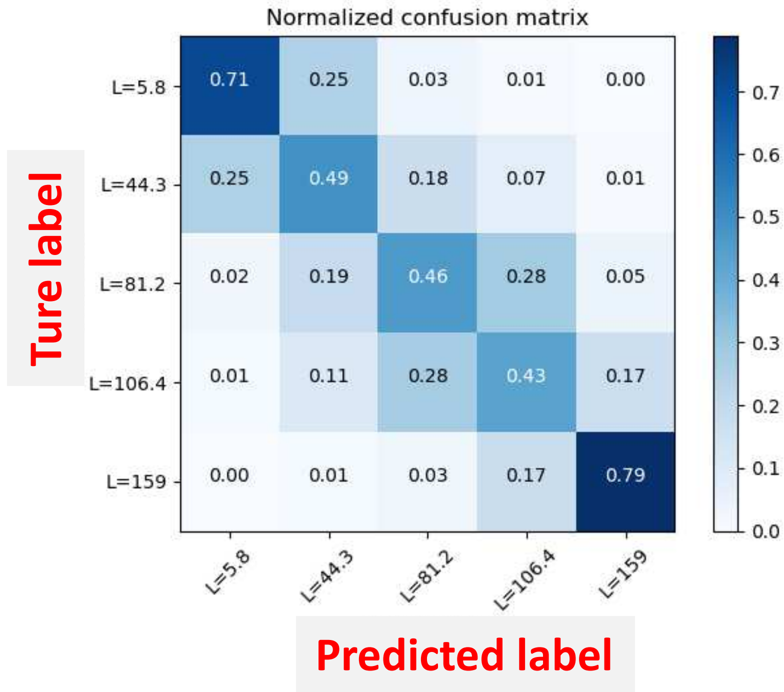
```
(12000, 20, 40, 1) (3000, 20, 40, 1) (12000, 5) (3000, 5)
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 20, 40, 64)        1664
_________________________________________________________________
batch_normalization_1 (Batch (None, 20, 40, 64)        160
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 20, 40, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 20, 40, 64)        0
_________________________________________________________________
average_pooling2d_1 (Average (None, 10, 20, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 10, 20, 128)       204928
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 10, 20, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 10, 20, 128)       0
_________________________________________________________________
average_pooling2d_2 (Average (None, 5, 10, 128)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6400)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               819328
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 128)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 645
=================================================================
Total params: 1,026,725
Trainable params: 1,026,645
Non-trainable params: 80
_________________________________________________________________
Train on 12000 samples, validate on 3000 samples
```

One million parameters

Results: y .vs. p_t for protons

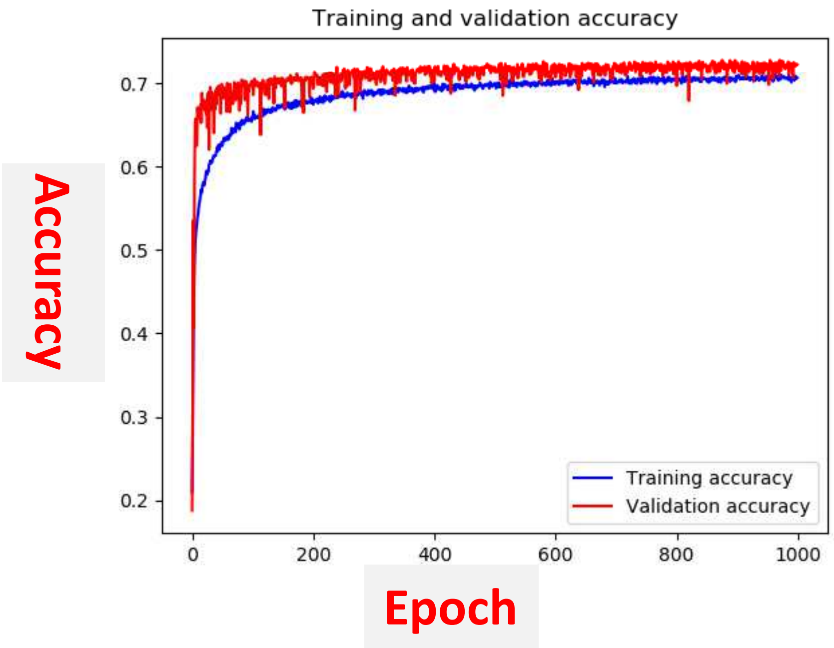
Classification

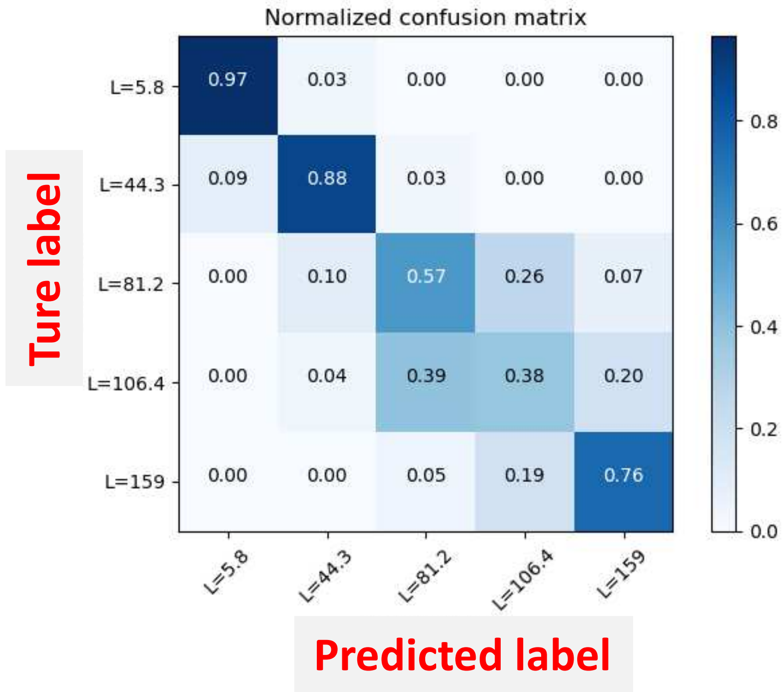




Results: y .vs. p_t for neutrons

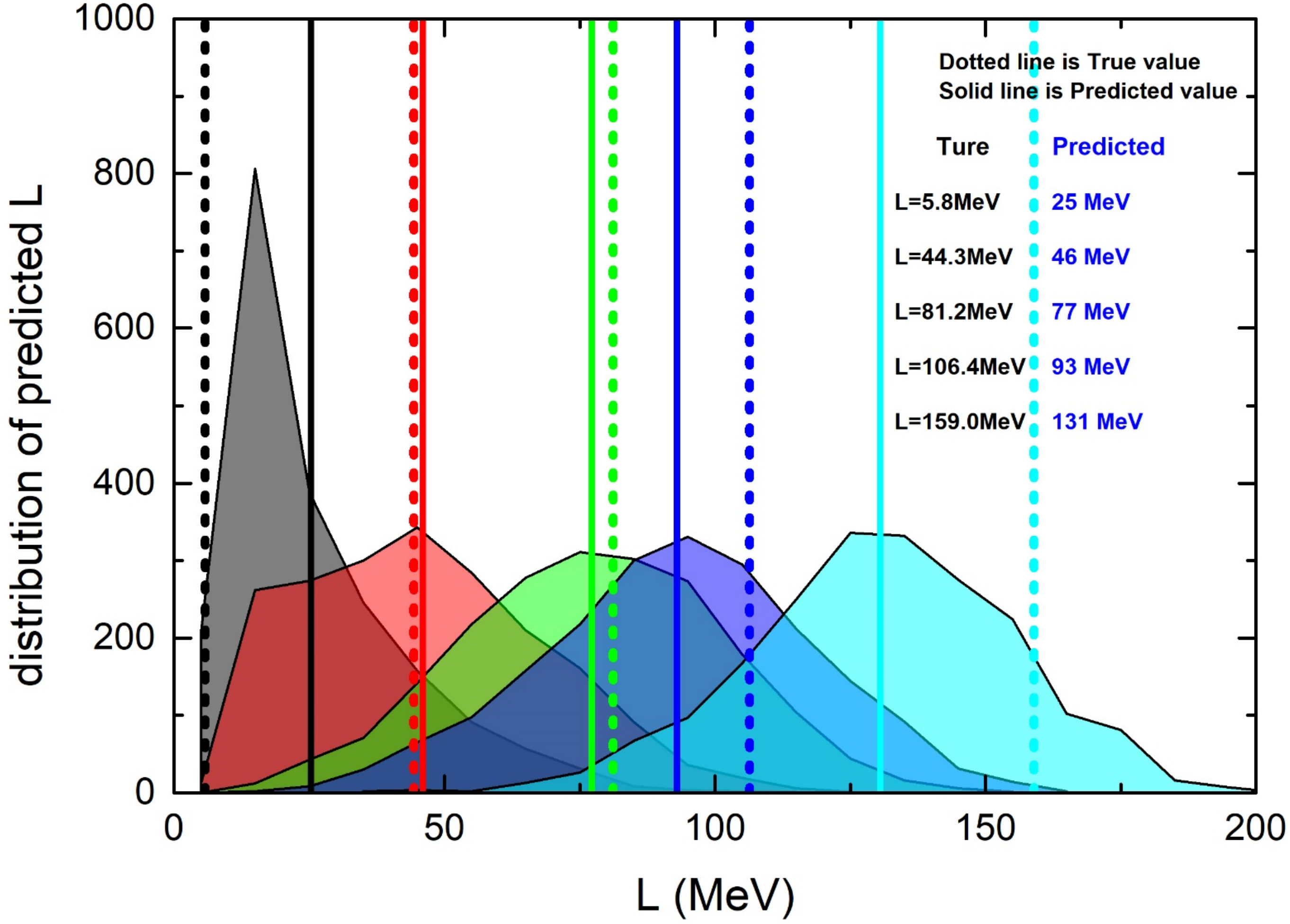
Classification





Results: y .vs. p_t for protons

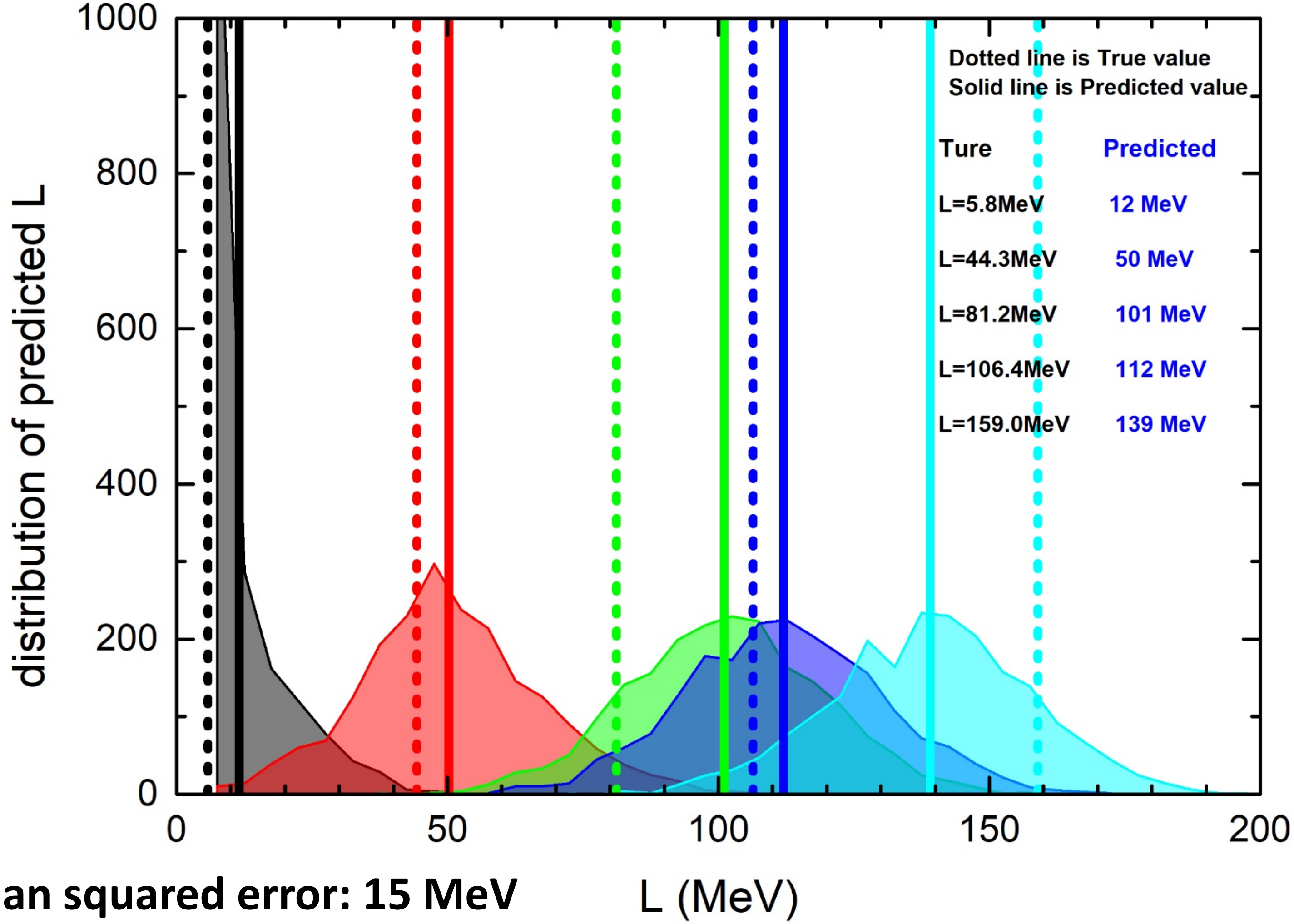
Regression



Mean squared error: 22 MeV

Results: y .vs. p_t for neutrons

Regression



Mean squared error: 15 MeV

Summary

- Deep learning can be used to study nuclear symmetry energy.
- 100 events = 1 samples, with proton y-p_t data, accuracy is about 55%, but with neutron data, accuracy can reach to 70%.
- For regression task, mean squared error are about 22 and 15 MeV for proton and neutron data, respectively.

Outlook

- Collecting different data, e.g., y-p_t-v_1/v_2,p_t-θ/φ
- Collecting more data from different models
- Using data from experiments
- The use of AI in heavy-ion physics has yet to reach its full potential.

Thanks for your attention!