Effective Description of Time Series

Tong (Owen) Yang
Department of Physics
Department of Economics
Boston College

Outline

- Time series structure and prediction problems
- Recurrent Neural Network for time-series model
- Dimension-reduction and tensor-train decomposition
- Matrix Product States and entanglement entropy area law
- Proposal

Outline

- Problem
- Model
- Difficulty and a trick
- Reason of the trick
- Proposal

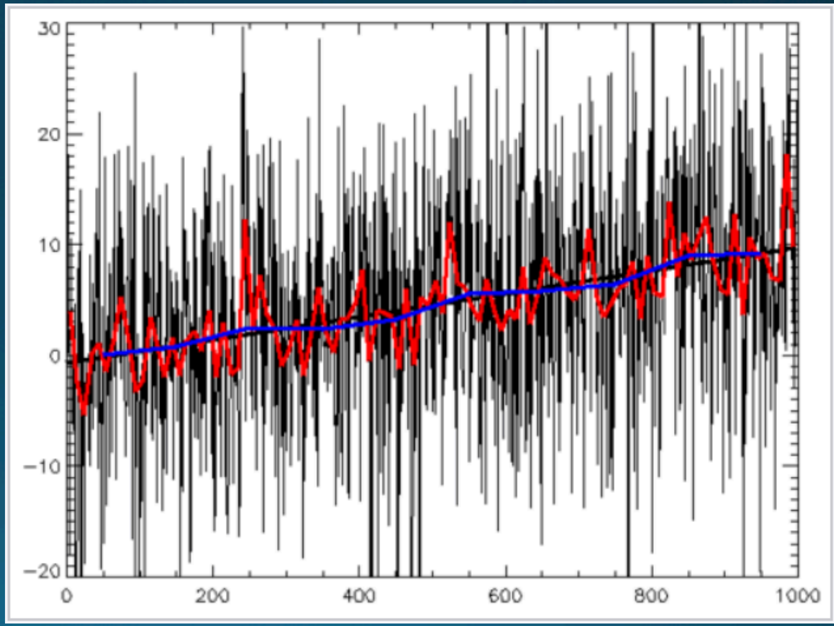
Outline

- Time series structure and prediction problems
- Recurrent Neural Network for time-series model
- Dimension-reduction and tensor-train decomposition
- Matrix Product States and entanglement entropy area law
- Proposal

Time series

Examples:

- Stock price
- Weather information
- Traffic current
- ……

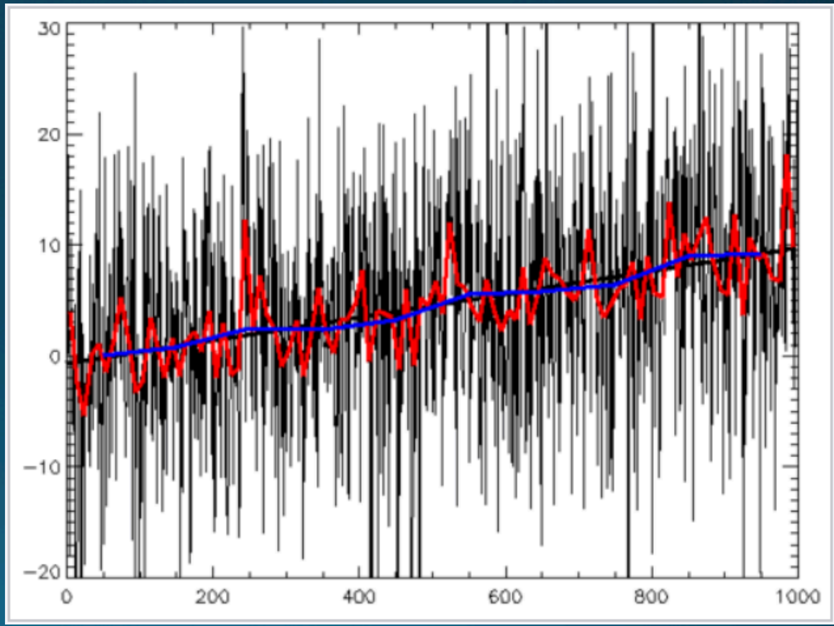


Time series

$\{X_t\}$

Prediction inference:

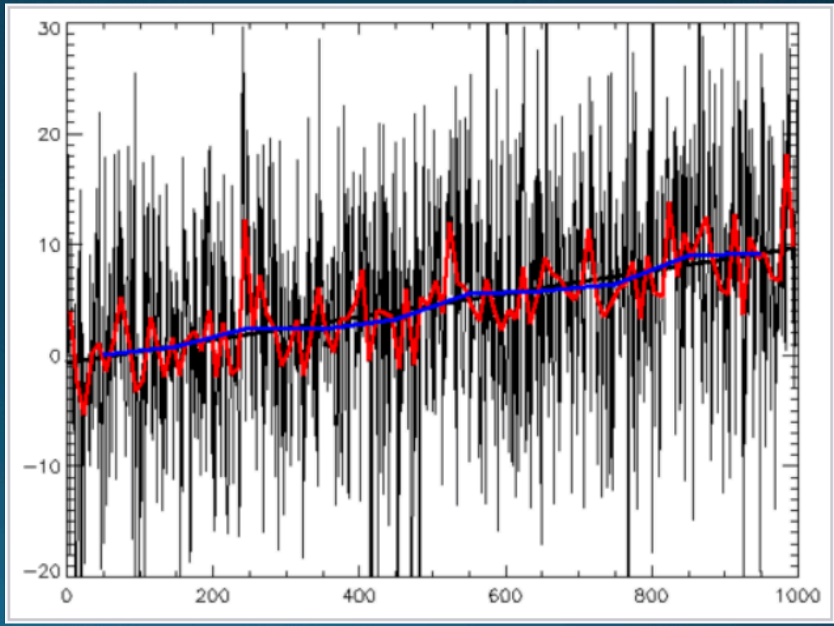
Predict future value according to past history



Time series

Typical Models for prediction:

- Auto Regression(AR)
- Moving average(MA)
- ARMA
- ARIMA
- ARCH
- GARCH
- ……

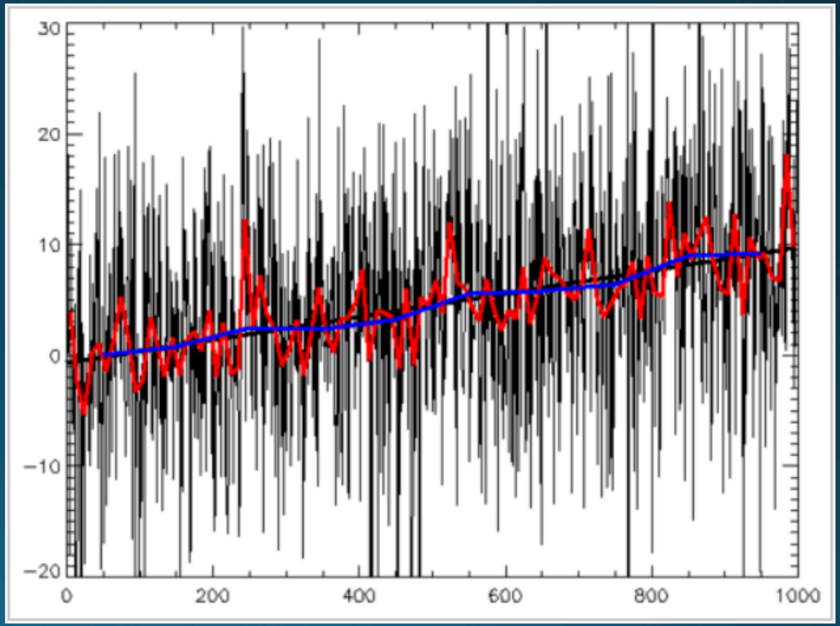


Time series

Challenge:

Real systems always have high-order nonlinear dynamics, especially worse for long-term predictions

$$\left\{\xi^i\left(\mathbf{x}_t, \frac{d\mathbf{x}}{dt}, \frac{d^2\mathbf{x}}{dt^2}, \ldots; \phi\right) = 0\right\}_i$$

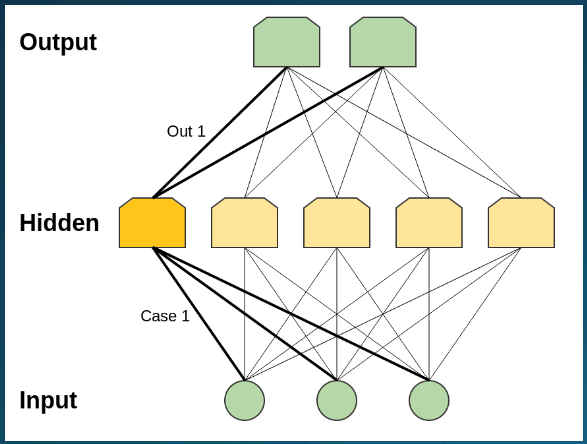


Outline

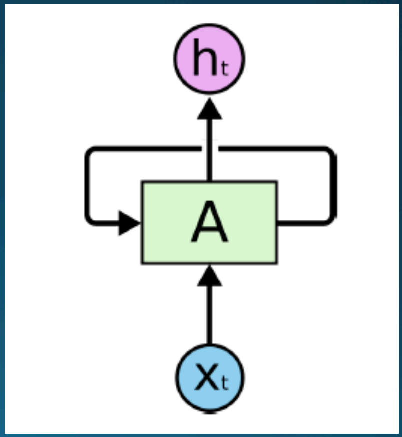
- Time series structure and prediction problems
- Recurrent Neural Network for time-series model
- Dimension-reduction and tensor-train decomposition
- Matrix Product States and entanglement entropy area law
- Proposal

Recurrent Neural Network

- A type of neural network that is designed to capture correlations in sequential data



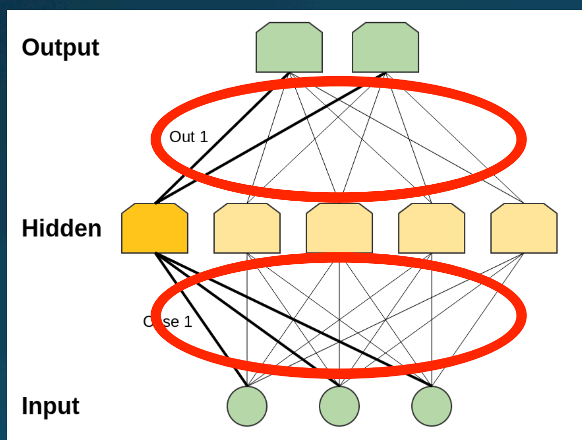
general neural network structure



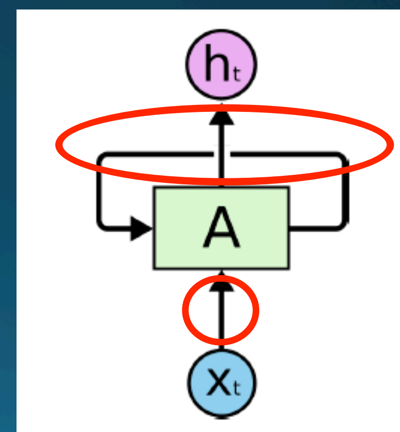
recurrent neural network structure

Recurrent Neural Network

- A type of neural network that is designed to capture correlations in sequential data



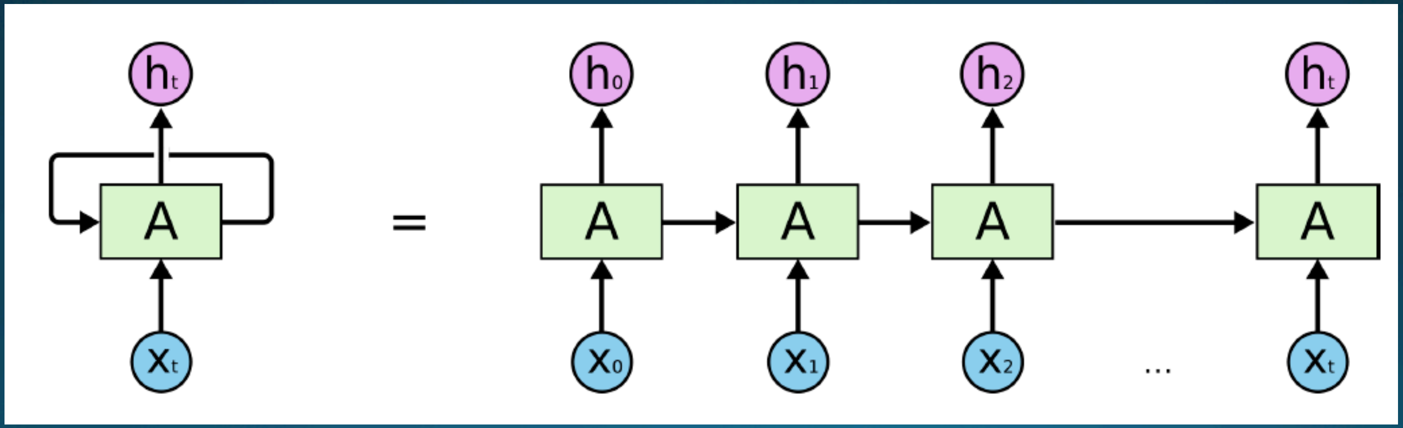
general neural network structure



recurrent neural network structure

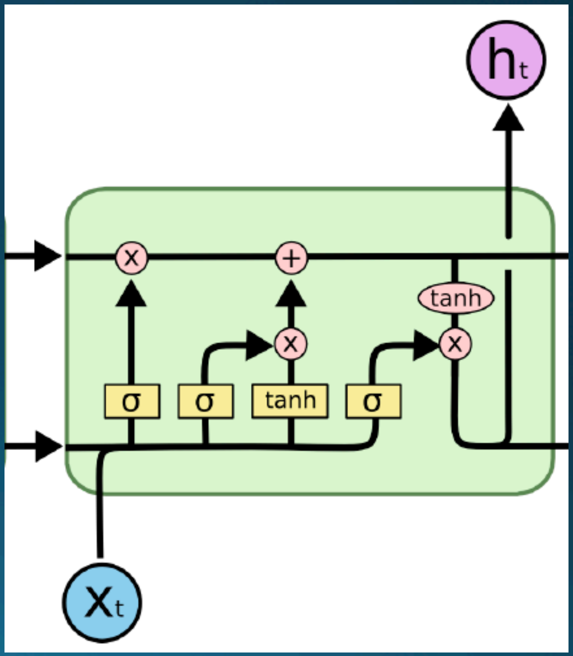
Recurrent Neural Network

- A type of neural network that is designed to capture correlations in sequential data



Recurrent Neural Network

- Several popular RNN models:
  - ➢Echo State Network (ESN)
  - ➢Long Short-term Memory (LSTM)
  - ➢Gated Recurrent Unit (GRU)
  - ➢Neural history compressor
  - ➢……



Example model: LSTM

Recurrent Neural Network

- Areas applied RNN successfully:
  - ❖Natural Language Processing(NLP); e.g. speech recognition (Soltau, 2016)
  - ❖Demand Forecasting (Flunkert, 2017)
  - ❖Video analysis (LeCun, 2015)
  - ❖Nonlinear dynamics; e.g. traffic prediction, weather prediction (Yu, 2018)
  - ❖……

Recurrent Neural Network

- Areas applied RNN successfully:
  - ❖Natural Language Processing(NLP); e.g. speech recognition (Soltau, 2016)
  - ❖Demand Forecasting (Flunkert, 2017)
  - ❖Video analysis (LeCun, 2015)
  - ❖Nonlinear dynamics; e.g. traffic prediction, weather prediction (Yu, 2018)

complicated nonlinear time correlation

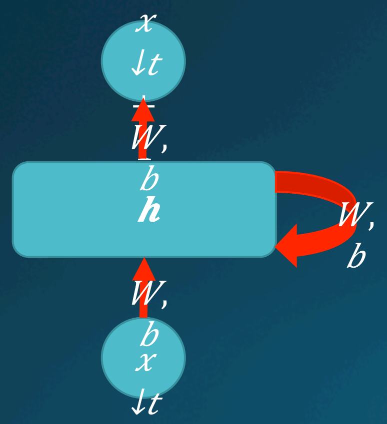
Outline

- Time series structure and prediction problems
- Recurrent Neural Network for time-series model
- Dimension-reduction and tensor-train decomposition
- Matrix Product States and entanglement entropy area law
- Proposal

“Dimension curse” of highly nonlinear time-series

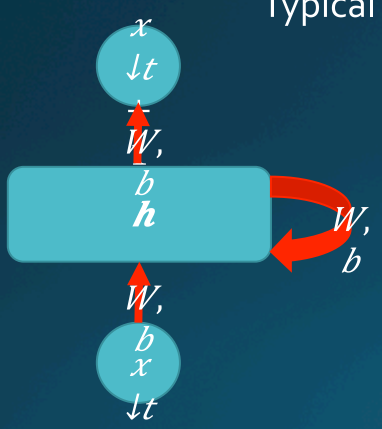
“Dimension curse” of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)



"Dimension curse" of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)



Typical mathematical structure of conventional RNN:

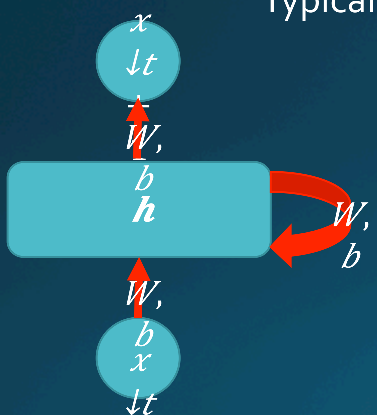
$$\mathbf{h}_t = f(W^{hx}\mathbf{x}_t + W^{hh}\mathbf{h}_{t-1} + \mathbf{b}^h),$$
$$\mathbf{x}_{t+1} = W^{xh}\mathbf{h}_t + \mathbf{b}^x,$$

h: hidden (auxiliary) units

x: time series values

“Dimension curse” of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)



Typical mathematical structure of conventional RNN:

$$\mathbf{h}_t = f(W^{hx}\mathbf{x}_t + W^{hh}\mathbf{h}_{t-1} + \mathbf{b}^h),$$
$$\mathbf{x}_{t+1} = W^{xh}\mathbf{h}_t + \mathbf{b}^x,$$

h: hidden (auxiliary) units

x: time series values

First-order Markov model

“Dimension curse” of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)

$$\left\{\xi^{i}\left(\mathbf{x}_{t}, \frac{d\mathbf{x}}{dt}, \frac{d^{2}\mathbf{x}}{dt^{2}}, \ldots; \phi\right) = 0\right\}_{i}$$

“Dimension curse” of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)
- Current difficulty: complex correlation in TS → requires longer history & higher order

$$\left\{\xi^i\left(\mathbf{x}_t, \frac{d\mathbf{x}}{dt}, \frac{d^2\mathbf{x}}{dt^2}, \ldots; \phi\right) = 0\right\}_i$$

“Dimension curse” of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)
- Current difficulty: complex correlation in TS → requires longer history & higher order

First order Markov model: $$[\mathbf{h}_t]_\alpha = f\left(W_\alpha^{hx}\mathbf{x}_t + W_\alpha^{hh}\mathbf{h}_{t-1}\right)$$

"Dimension curse" of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)
- Current difficulty: complex correlation in TS → requires longer history & higher order

First order Markov model:

$$[\mathbf{h}_t]_\alpha = f\left(W_\alpha^{hx}\mathbf{x}_t + W_\alpha^{hh}\mathbf{h}_{t-1}\right)$$

L-lag Markov process:

$$\mathbf{s}_{t-1}^T = [1 \;\; \mathbf{h}_{t-1}^\top \;\; \ldots \;\; \mathbf{h}_{t-L}^\top]$$

$$\dim[S_{t-1}] = HL+1$$

"Dimension curse" of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)
- Current difficulty: complex correlation in TS → requires longer history & higher order

First order Markov model:

$$[\mathbf{h}_t]_\alpha = f\left(W_\alpha^{hx}\mathbf{x}_t + W_\alpha^{hh}\mathbf{h}_{t-1}\right)$$

L-th order Markov model:

$$[\mathbf{h}_t]_\alpha = f(W_\alpha^{hx}\mathbf{x}_t + \sum_{i_1,\cdots,i_p} \mathcal{W}_{\alpha i_1\cdots i_P} \underbrace{\mathbf{s}_{t-1;i_1} \otimes \cdots \otimes \mathbf{s}_{t-1;i_p}}_{P})$$

P: order of polynomials

"Dimension curse" of highly nonlinear time-series

- Conventional RNN: only explicitly use data of last moment (t-1)
- Current difficulty: complex correlation in TS → requires longer history & higher order

First order Markov model:

$$[\mathbf{h}_t]_\alpha = f\left(W_\alpha^{hx}\mathbf{x}_t + W_\alpha^{hh}\mathbf{h}_{t-1}\right)$$

L-th order Markov model:

$$[\mathbf{h}_t]_\alpha = f(W_\alpha^{hx}\mathbf{x}_t + \sum_{i_1,\cdots,i_p} \mathcal{W}_{\alpha i_1 \cdots i_P} \underbrace{\mathbf{s}_{t-1;i_1} \otimes \cdots \otimes \mathbf{s}_{t-1;i_p}}_{P})$$

Therefore increase the learning (modeling) difficulty.

More technically, searched **parameter space** would be way **too large**

“Dimension curse” of highly nonlinear time-series

- Dimension counting and difficulty analysis:

$$\mathcal{W}_{i_1 \cdots i_P}$$

$$\mathbf{dim}[\textbf{parameter space}] = (\mathbf{HL+1})^{\mathbf{P}}$$

H: number of hidden units
L: length of time-lag
P: order of polynomials

Dimension reduction

- Tensor-train decomposition:

$$\mathcal{W}_{i_1 \cdots i_P} = \sum_{\alpha_1 \cdots \alpha_{P-1}} \mathcal{A}^1_{\alpha_0 i_1 \alpha_1} \mathcal{A}^2_{\alpha_1 i_2 \alpha_2} \cdots \mathcal{A}^P_{\alpha_{P-1} i_P \alpha_P}$$

$\dim[\text{parameter space}] = (HL+1)R^2 P$

H: number of hidden units
L: length of time-lag
P: order of polynomials
R: bond dimension

Dimension reduction

- Tensor-train decomposition:

$$\mathcal{W}_{i_1 \cdots i_P} = \sum_{\alpha_1 \cdots \alpha_{P-1}} \mathcal{A}^1_{\alpha_0 i_1 \alpha_1} \mathcal{A}^2_{\alpha_1 i_2 \alpha_2} \cdots \mathcal{A}^P_{\alpha_{P-1} i_P \alpha_P}$$

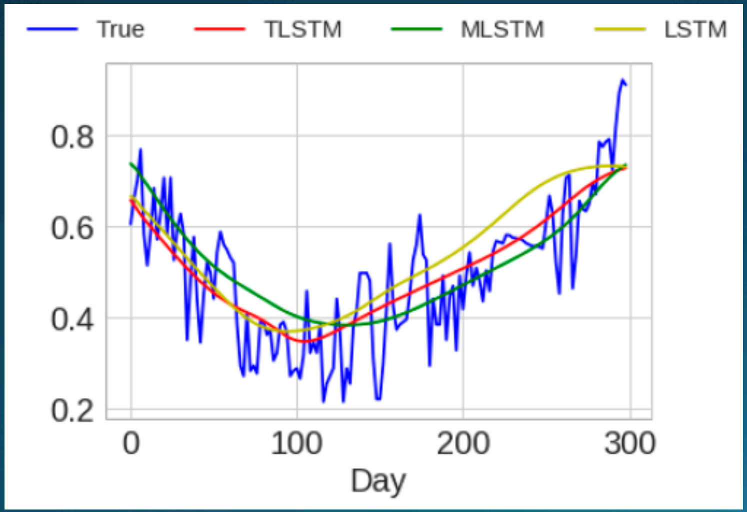
$$\mathbf{dim}[\textbf{parameter space}] = (\mathbf{HL+1})\mathbf{R}^{\mathbf{2}}\,\mathbf{P}$$

IMPORTANT SIMPLIFICATION

Dimension reduction

- Tensor-train decomposition:

Daily max-temperature prediction:



Yu, etc. (2017)

It works!

At least captures the main trend

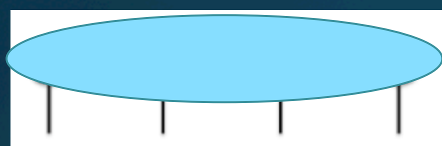
(Given 2 months input, predict 300 days ahead)

Outline

- Time series structure and prediction problems
- Recurrent Neural Network for time-series model
- Dimension-reduction and tensor-train decomposition
- Matrix Product States and entanglement entropy area law
- Proposal

Tensor networks in condensed matter physics

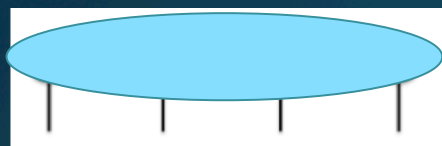
- Matrix Product State of *1-dim* wave-functions:



$$|\Psi\rangle = \sum_{i_1,i_2\cdots i_P} \mathcal{W}_{i_1,i_2\cdots i_P} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_P\rangle$$

Tensor networks in condensed matter physics

- Matrix Product State of *1-dim* wave-functions:

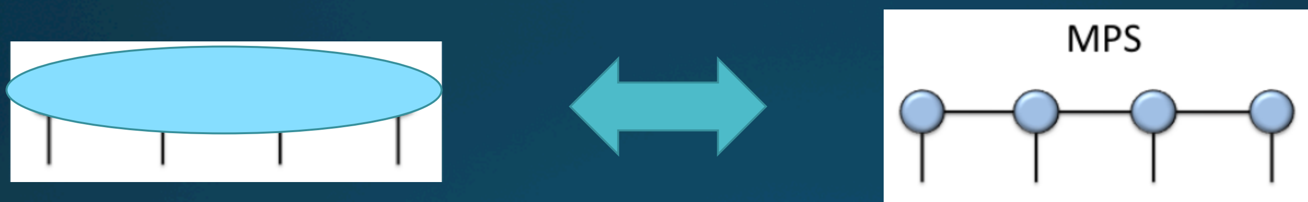


$$|\Psi\rangle = \sum_{i_1, i_2 \cdots i_P} \mathcal{W}_{i_1, i_2 \cdots i_P} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_P\rangle$$

$$\mathcal{W}_{i_1 \cdots i_P} = \sum_{\alpha_1 \cdots \alpha_{P-1}} \mathcal{A}^1_{\alpha_0 i_1 \alpha_1} \mathcal{A}^2_{\alpha_1 i_2 \alpha_2} \cdots \mathcal{A}^P_{\alpha_{P-1} i_P \alpha_P}$$

Tensor networks in condensed matter physics

- Matrix Product State of *1-dim* wave-functions:

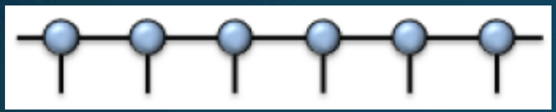


$$\mathcal{W}_{i_1 \cdots i_P} = \sum_{\alpha_1 \cdots \alpha_{P-1}} \mathcal{A}^1_{\alpha_0 i_1 \alpha_1} \mathcal{A}^2_{\alpha_1 i_2 \alpha_2} \cdots \mathcal{A}^P_{\alpha_{P-1} i_P \alpha_P}$$

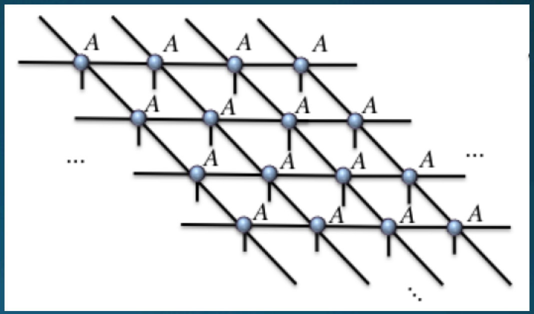
Tensor networks in condensed matter physics

- General tensor-network wave-functions:

1-dim MPS:



2-dim PEPS:



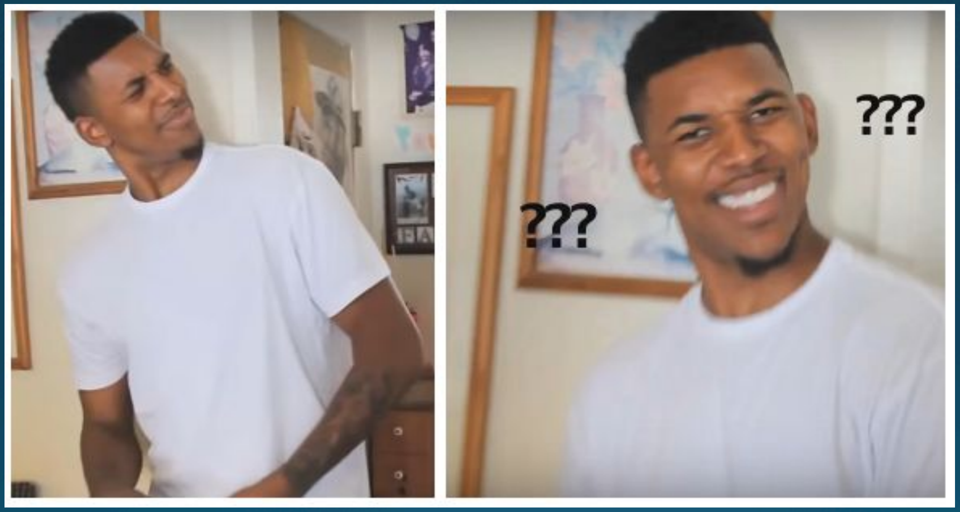
Tensor networks in condensed matter physics

- General tensor-network wave-functions:

$$|\psi\rangle = \sum_{\{k_s\}} \mathrm{tTr}\left((T^1)^{k_1} \ldots (T^{N_s})^{k_{N_s}} B_1 \ldots B_{N_b}\right) |k_1 \ldots k_{N_s}\rangle$$

Generally, tensor-network represents a tensor decomposition from higher order (dimensional space) tensors to lower order tensors

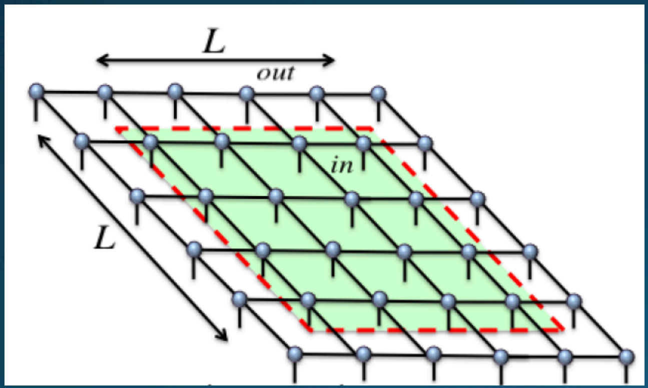
Deep reason of ***"decomposibility"***



Why the heck can we do this ???

Deep reason of ***"decomposibility"***

- Von Neumann Entanglement Entropy:



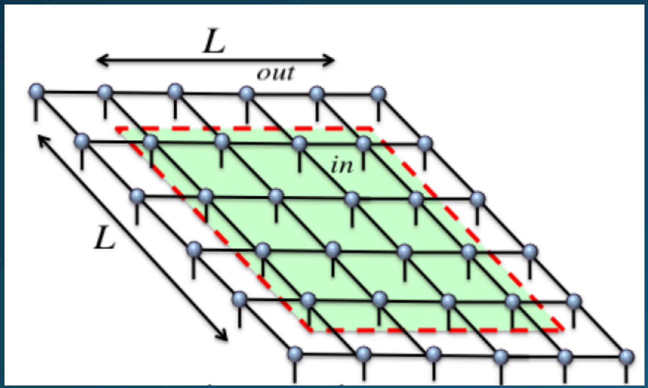
Density Matrix:

$$\rho_{AB} = |\Psi\rangle\langle\Psi|_{AB}$$

$$\mathcal{S}(\rho_A) = -\,\mathrm{Tr}[\rho_A \log \rho_A]$$

Deep reason of ***"decomposibility"***

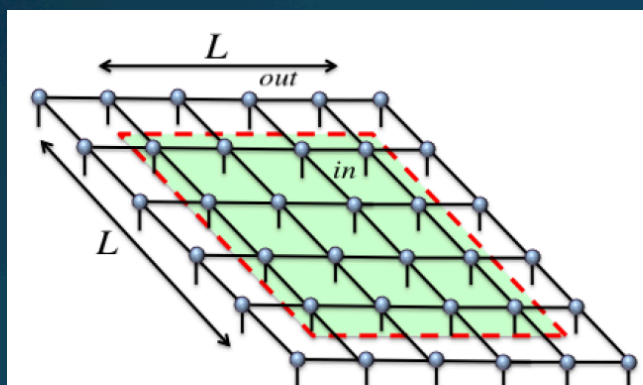
- Von Neumann Entanglement Entropy:



$$S_A \propto V_A$$

Deep reason of ***"decomposibility"***

- ***Area-law of Entanglement Entropy of gapped many-body ground state:***

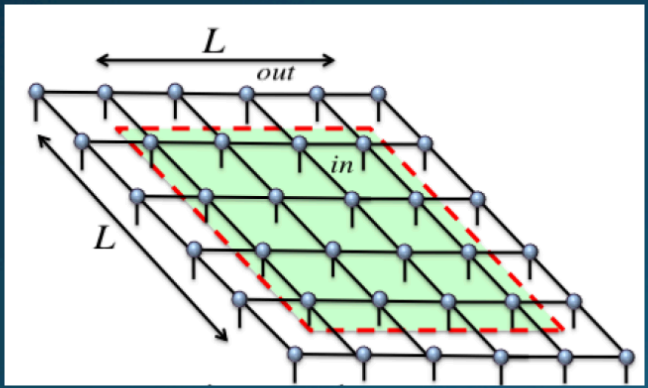


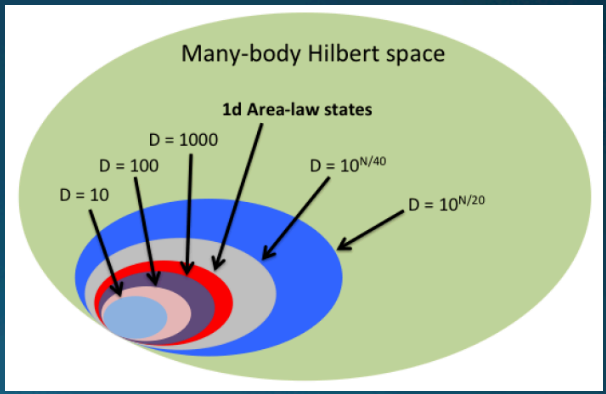
$$S_A \propto V_A$$

$$S_A \propto \partial V_A$$

Deep reason of ***"decomposibility"***

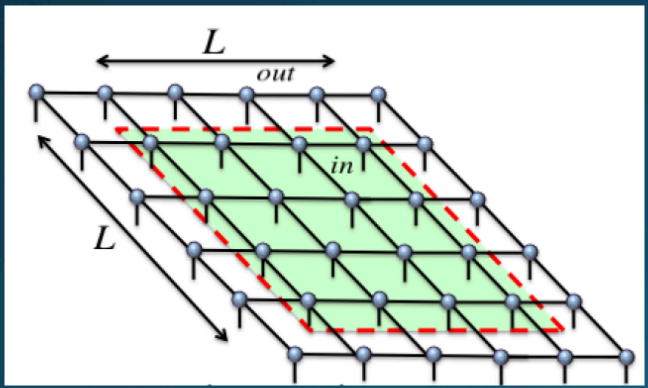
- ***Area-law of Entanglement Entropy of gapped many-body ground state:***

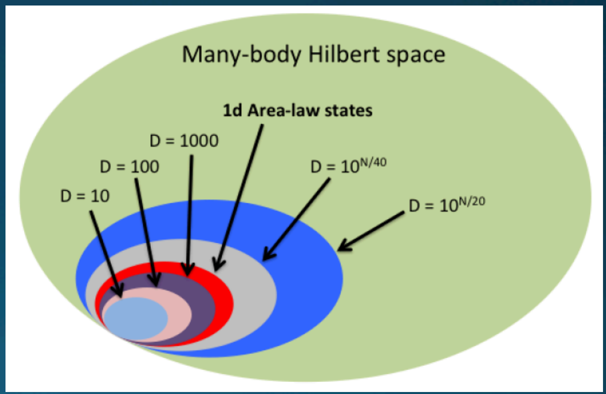




Deep reason of ***"decomposibility"***

- ***Area-law of Entanglement Entropy of gapped many-body ground state:***





And it has been proved: MPS and PEPS wave-functions obey area law

Outline

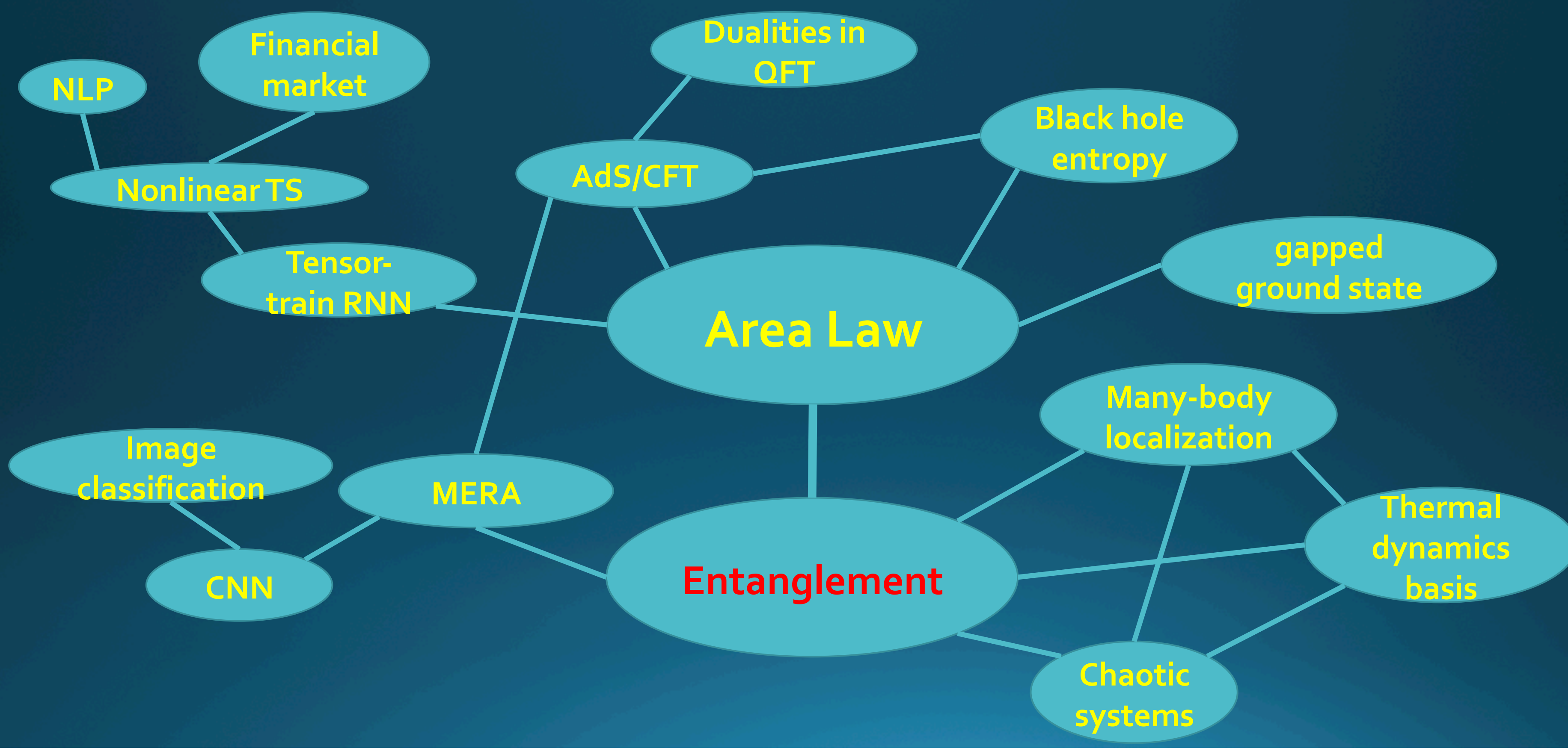
- Time series structure and prediction problems
- Recurrent Neural Network for time-series model
- Dimension-reduction and tensor-train decomposition
- Matrix Product States and entanglement entropy area law
- Proposal

Application in time-series prediction

- ***Knowledge we have:***
  1. *RNN works for non-linear time-series prediction*
  2. *Tensor-train decomposition can simplify RNN*
  3. *Decompositions correspond to MPS*
  4. *MPS works when area law is satisfied*

- ***Simple question we can ask:***
  - *Given a time series, is there a "area law" in the structure?*
  - *Or inversely, what kind of time-series obey area law?*

Global picture of entanglement



Thanks

General Optimization problem

It's not a day-dream, seriously……

Example:
correspondence between image-classification problem and
tensor-network representation of many-body ground state

	image-classification	tensor-network ground-state
real-space coordinates	$\vec{r}$	$\vec{r}$
Hilbert space basis	$s_i(\vec{r})$	$\phi_i(\vec{r})$
target functions	$F(\vec{r}) = \sum_i F(s_i) s_i(\vec{r})$	$\Psi(\vec{r}) = \sum_i \lambda(\phi_i)\phi_i(\vec{r})$
coefficients value	$F(s_i) = F_0 \cdot \mathcal{I}(s_i \in targets)$	$\lambda(\phi_i) \in \mathbb{C}$
normalization	$\sum_i \lvert F(s_i)\rvert^2 = 1$	$\sum_i \lvert\lambda(\phi_i)\rvert^2 = 1$
tuning parameters	hyper-parameters: W	tensors: w, u
approximating targets	$f_{W_0}(s_i)$	$\lambda_{w_0,u_0}(\phi_i)$
minimization	$E(W) \approx \sum_i' V\left[f_W(s_i) - f_{W_0}(s_i)\right]$	$E(w,u) \approx E(\lambda_{w,u}, H)$
density matrix	$\rho_{ij} = f_W(s_i) \cdot f_W^*(s_j)$	$\rho_{ij} = \lambda_{w,u}(\phi_i) \cdot \lambda_{w,u}(\phi_j)$

Two more interesting topics…

- Network structural differences in decentralized systems:
  - ❖ blockchain: difference network structure
  - ❖ personal credit records
  - ❖ micro-finance: loans for the poor, small volume
  - ❖ data: applied in Africa already
  - ❖ Question: what's the network structural impact on credit records?

- Machine learning for crowd-funding
  - ❖ analyze historical info
  - ❖ predict success rate of campaigns
  - ❖ predict refunding probability

Thanks

General Optimization problem

- ***Find the proper function mapping from input to output:***

$$|\Psi\rangle = \sum_{i_1, i_2 \cdots i_P} \mathcal{W}_{i_1, i_2 \cdots i_P} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_P\rangle$$

General Optimization problem

- ***Find the proper set of parameters:***

$$|\Psi\rangle = \sum_{i_1, i_2 \cdots i_P} \mathcal{W}_{i_1, i_2 \cdots i_P} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_P\rangle$$

General Optimization problem

- ***Find the proper set of parameters:***

$$|\psi\rangle = \sum_{\{k_s\}} \mathrm{tTr}\left((T^1)^{k_1}...(T^{N_s})^{k_{N_s}} B_1...B_{N_b}\right)|k_1\ldots k_{N_s}\rangle$$

Question:

Is there a lower-dim Hilbert space for the parameter searching?

General Optimization problem

- ***Find the proper set of parameters:***

$$|\psi\rangle = \sum_{\{k_s\}} \mathrm{tTr}\left((T^1)^{k_1}...(T^{N_s})^{k_{N_s}} B_1...B_{N_b}\right) |k_1 \dots k_{N_s}\rangle$$

Idea:

Correlation/Entanglement structure is the key!!!

General Optimization problem

- ***Proposal:***
  - ***Set up the Hilbert space for a general optimization problem***
  - ***Analytical study:***
    - ***Analyze the Hilbert space according to correlation behavior***
    - ***Classify problems in restricted Hilbert space***
  - ***Empirical study:***
    - ***Test performance of tensor-decomposition in different NN***
  - ***Compare results from two sides***