Chromaticity Correction with Physics-inspired Neural Networks

5.11.2020

Artificial Neural Networks

Artificial Neural Networks are

- able to approximately represent any function $f : \mathbb{R}^m \to \mathbb{R}^n$ with arbitrary precision

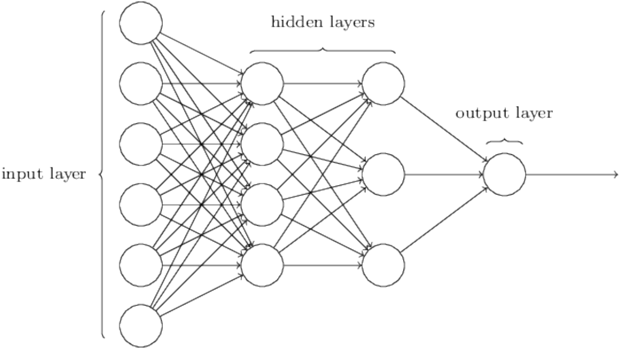


Figure: [Nie15]

Artificial Neural Networks

Layers are

- layers are composition of a linear map l and an element-wise non-linear map σ, e.g. $\boldsymbol{x} \to (\sigma \circ l)\boldsymbol{x}$
- l contains *trainable* weights M_{ij} and biases b_i
  $l(\boldsymbol{x}) = M\boldsymbol{x} + b$

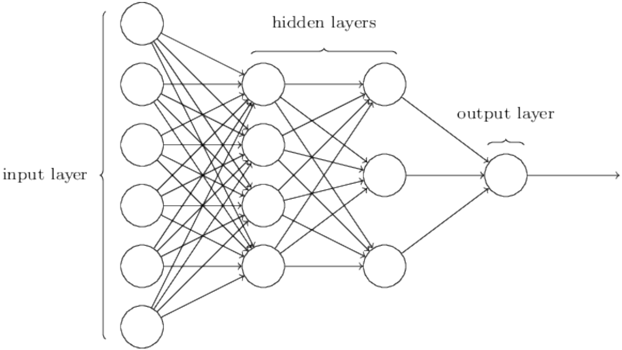


Figure: [Nie15]

Training Artificial Neural Networks

- train from a set of n examples $\{(x_n, y(x_n)\}_n$
- minimize some metric $\sum_n ||y(x_n) - \text{net}(x_n)||$ via gradient descent with respect to the trainable weights

Training Artificial Neural Networks

- train from a set of n examples $\{(x_n, y(x_n)\}_n$
- minimize some metric $\sum_n ||y(x_n) - \text{net}(x_n)||$ via gradient descent with respect to the trainable weights
- training successful if prediction is accurate for inputs not included in examples
  - the ANN is able to *generalize*

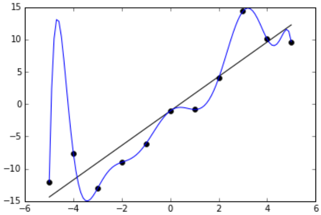


Figure: [Wik]

Physics-inspired Neural Networks

Physics-inspired Neural Networks

- exploit domain knowledge to construct network architecture
- replace layers by polynomial maps with trainable weights W_k
  $l(\boldsymbol{x}) = \boldsymbol{x} + W_1\boldsymbol{x} + W_2\boldsymbol{x}^2 + ...$
- weight matrices W_k can be obtained from beam dynamics, e.g. affiliated from MAD-X, elegant, ...
- represent each element in beamline by a layer

Physics-inspired Neural Networks

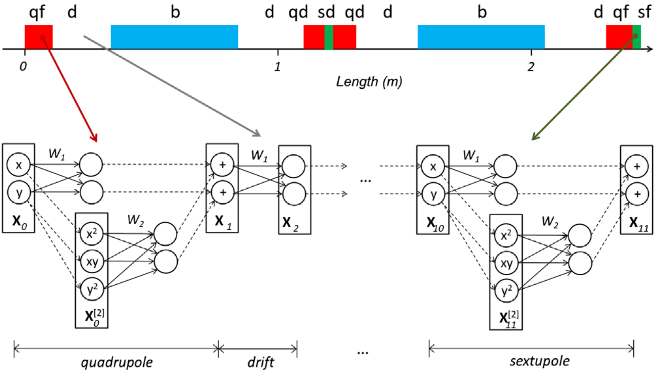


Figure: [IA20]

Physics-inspired Neural Networks

- NNs prone to overfitting
  - usually if NN is too large / training set is small

Physics-inspired Neural Networks

- NNs prone to overfitting
  - usually if NN is too large / training set is small
- can be reduced by regularization
  - append penalty term to loss by adding constraints on trainable weights

Physics-inspired Neural Networks

- NNs prone to overfitting
  - usually if NN is too large / training set is small
- can be reduced by regularization
  - append penalty term to loss by adding constraints on trainable weights
- symplectic regularization
  - derive constraints from symplectic property

$$\left(\frac{\partial M}{\partial \boldsymbol{x}}\right)^{-1} J \left(\frac{\partial M}{\partial \boldsymbol{x}}\right) - J = 0; \quad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

Physics-inspired Neural Networks

- NNs prone to overfitting
  - usually if NN is too large / training set is small
- can be reduced by regularization
  - append penalty term to loss by adding constraints on trainable weights
- symplectic regularization
  - derive constraints from symplectic property

$$\left(\frac{\partial M}{\partial \boldsymbol{x}}\right)^{-1} J \left(\frac{\partial M}{\partial \boldsymbol{x}}\right) - J = 0; \quad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

- alternatively: use thin-lens approximation
  - symplectic by design
  - fewer trainable weights

Goals

- use PNNs for fast particle tracking
- exploit common machine-learning libraries
  - offers building blocks for NNs
  - provides tools for model training like optimizers, gradient calculation, ...
  - use optimized code from high-level programming language
  - access to GPUs

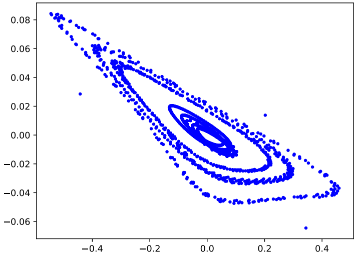
Goals

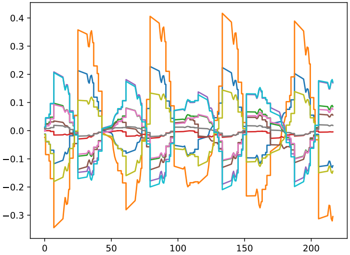
- use PNNs for fast particle tracking
- exploit common machine-learning libraries
    - offers building blocks for NNs
    - provides tools for model training like optimizers, gradient calculation, ...
    - use optimized code from high-level programming language
    - access to GPUs
- use PNNs to obtain accurate accelerator model by fitting measured data
- predict chromaticity of SIS18 and correct it

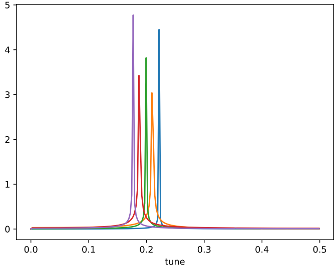
Goals

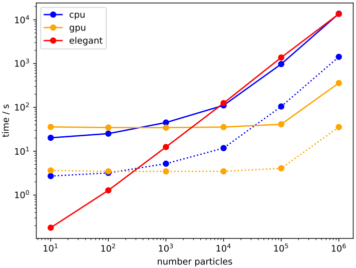
- use PNNs for fast particle tracking
- exploit common machine-learning libraries
  - offers building blocks for NNs
  - provides tools for model training like optimizers, gradient calculation, ...
  - use optimized code from high-level programming language
  - access to GPUs
- use PNNs to obtain accurate accelerator model by fitting measured data
- predict chromaticity of SIS18 and correct it
- optional: try to include space charge effects

Current Status









References I

[IA20] Andrei Ivanov and Ilya Agapov. Physics-based deep neural networks for beam dynamics in charged particle accelerators. *Physical Review Accelerators and Beams*, 23(7):074601, 2020.

[Nie15] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[Wik] Wikipedia, the free encyclopedia. Polynomial overfitting. [Online; accessed November 4, 2020].