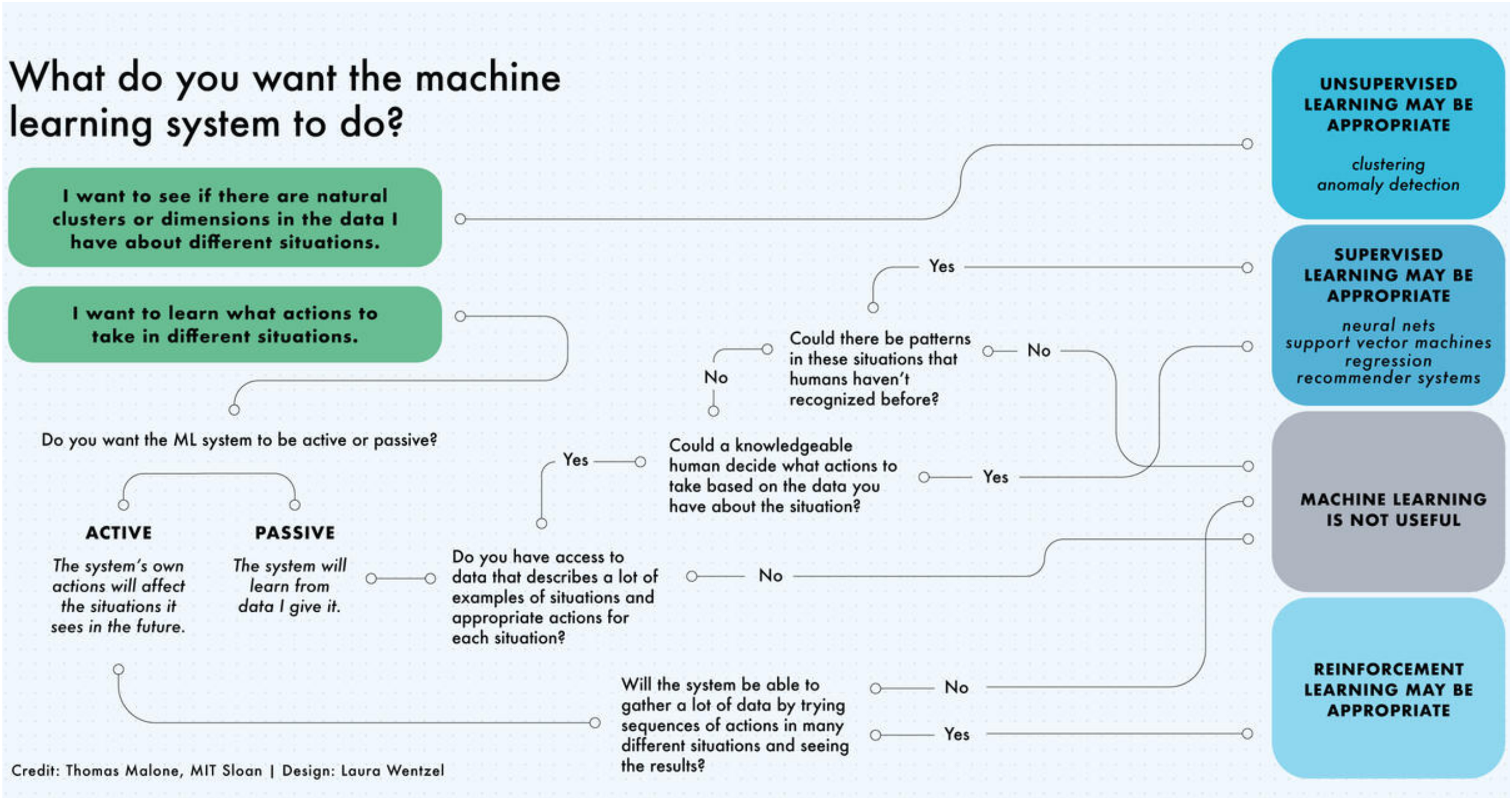
Machine Learning for Nuclear Physics
Lecture 5

Rabah Abdul Khalek
Jefferson Lab

HUGS22
Thursday 06.09.2022

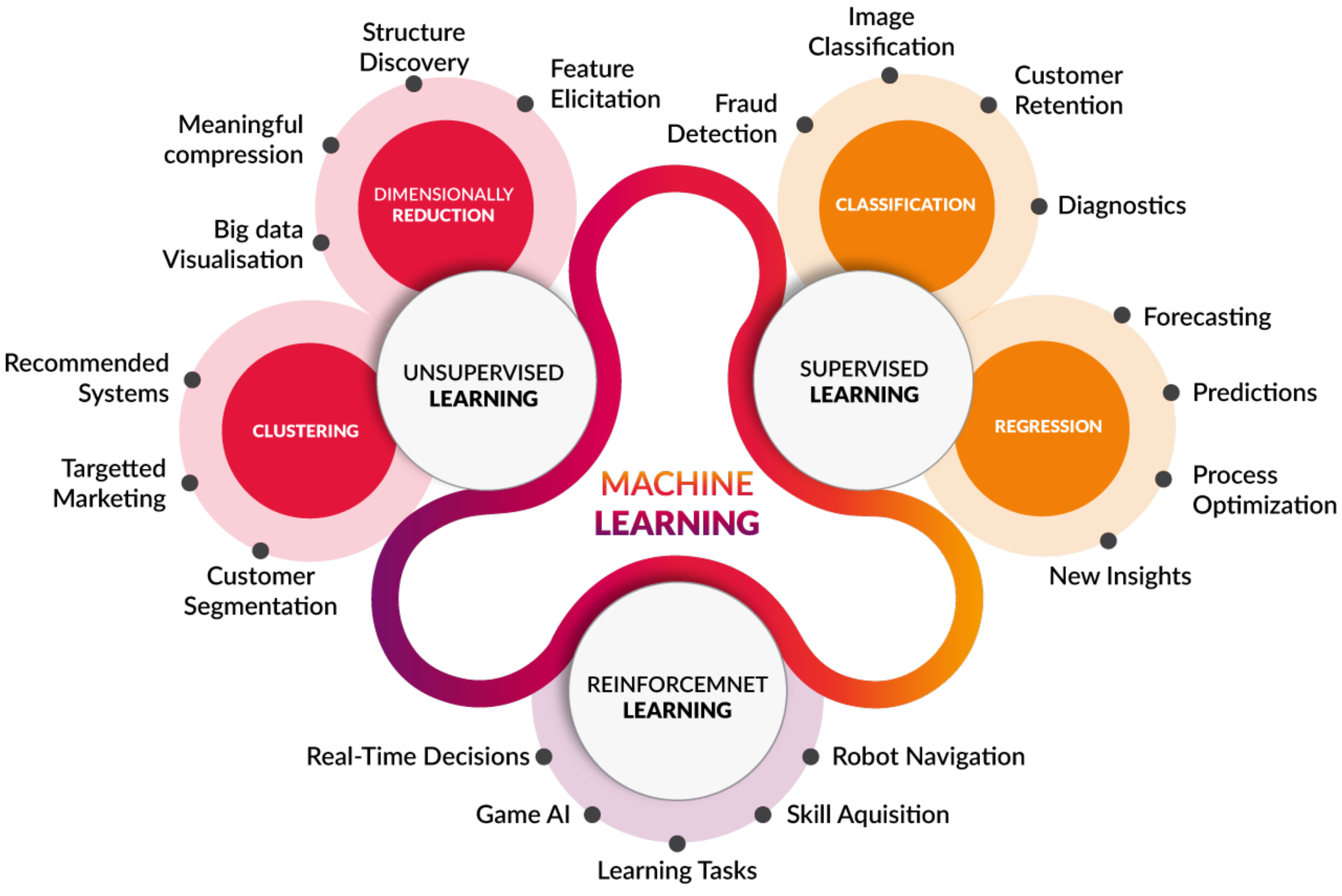
Outline

Machine Learning



Source: Thomas Malone I MIT Sloan. See: https://bit.ly/3gvRho2, Figure 2.

Machine Learning



Source: https://medium.com/analytics-vidhya/an-introduction-to-machine-learning-574bafa6fc66

A mostly complete chart of

Neural Networks

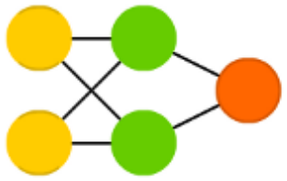
©2016 Fjodor van Veen - asimovinstitute.org

- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool

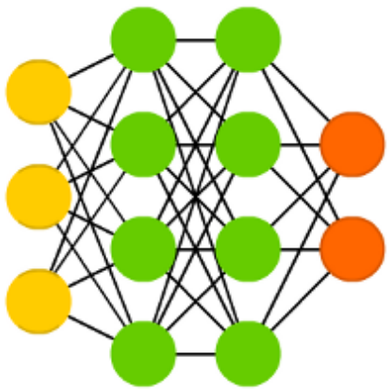
Perceptron (P)

Feed Forward (FF)

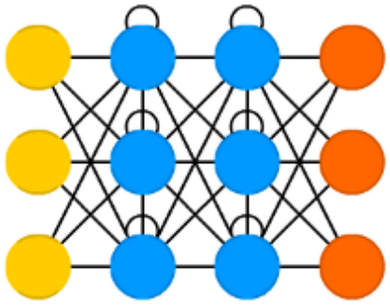
Radial Basis Network (RBF)



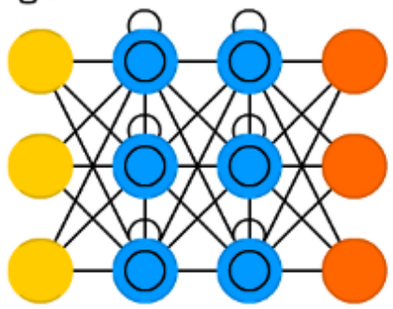
Deep Feed Forward (DFF)



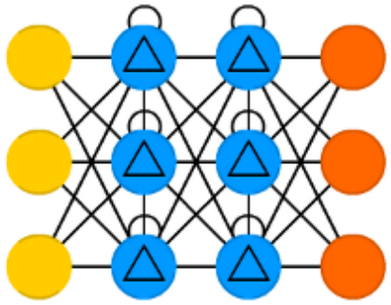
Recurrent Neural Network (RNN)



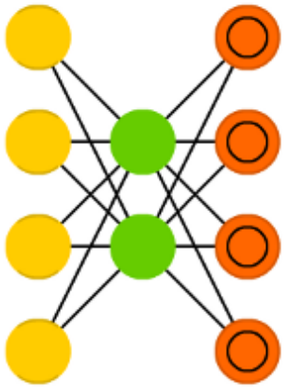
Long / Short Term Memory (LSTM)



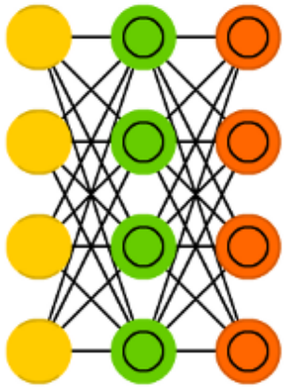
Gated Recurrent Unit (GRU)



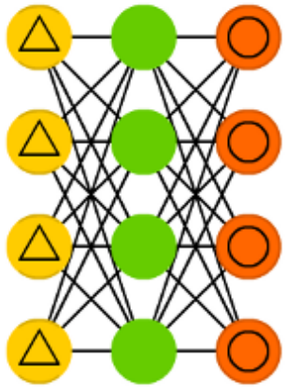
Auto Encoder (AE)



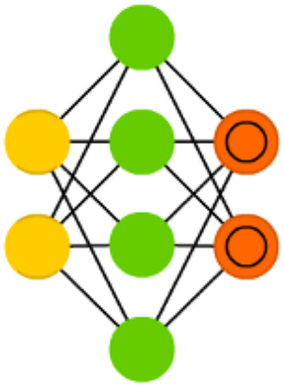
Variational AE (VAE)



Denoising AE (DAE)



Sparse AE (SAE)



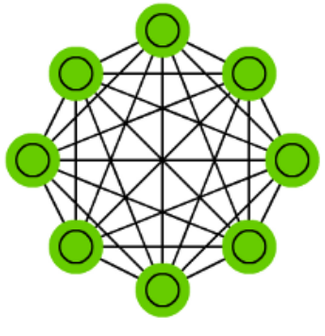
Source: https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

A mostly complete chart of

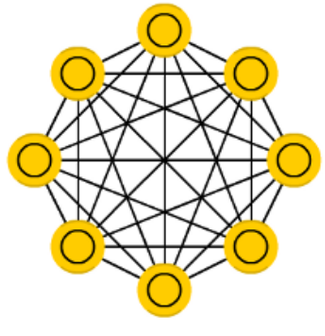
Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

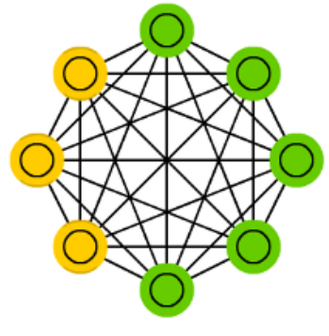
Markov Chain (MC)



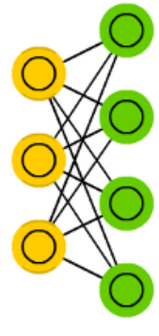
Hopfield Network (HN)



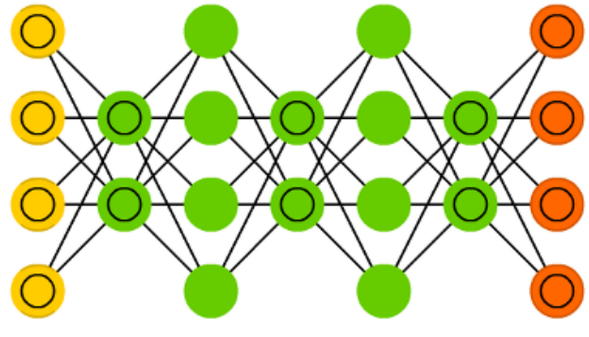
Boltzmann Machine (BM)



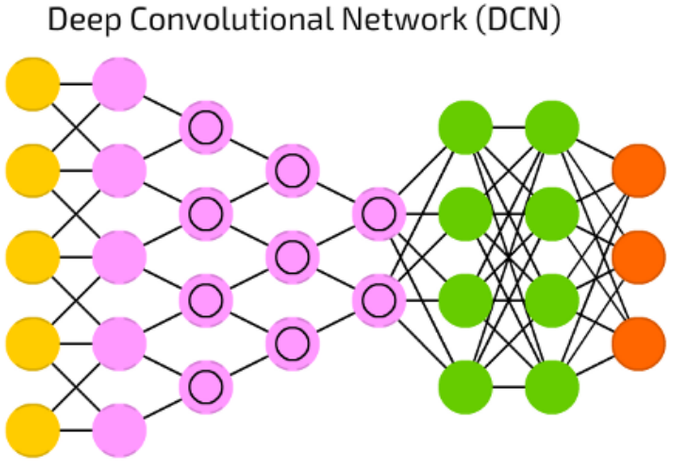
Restricted BM (RBM)



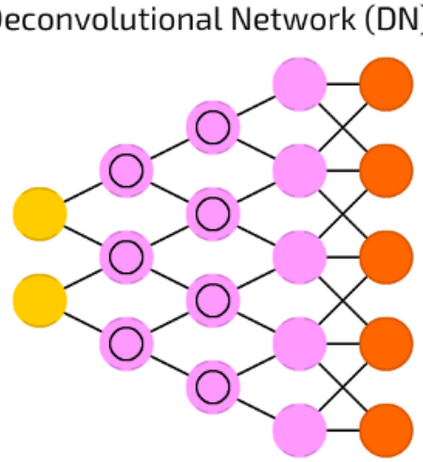
Deep Belief Network (DBN)



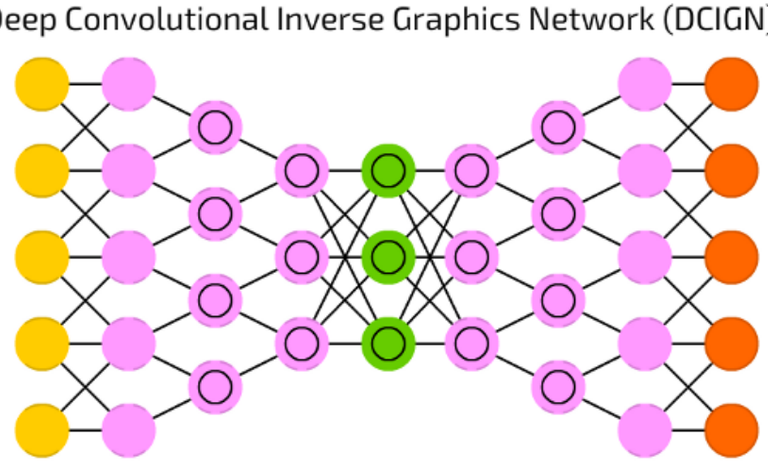
Deep Convolutional Network (DCN)



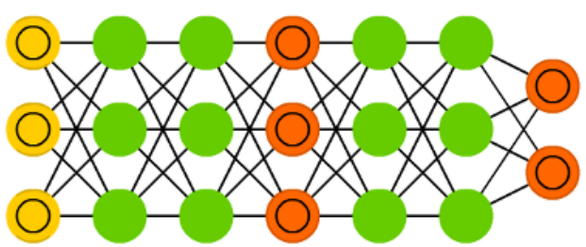
Deconvolutional Network (DN)



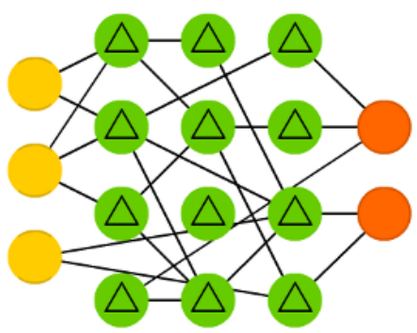
Deep Convolutional Inverse Graphics Network (DCIGN)



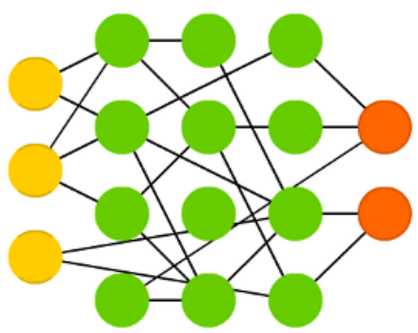
Generative Adversarial Network (GAN)



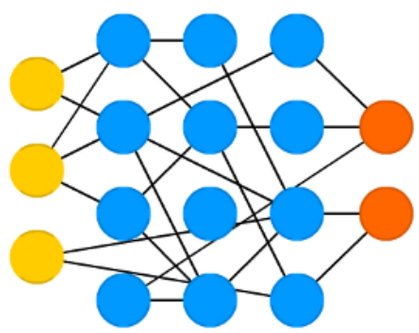
Liquid State Machine (LSM)



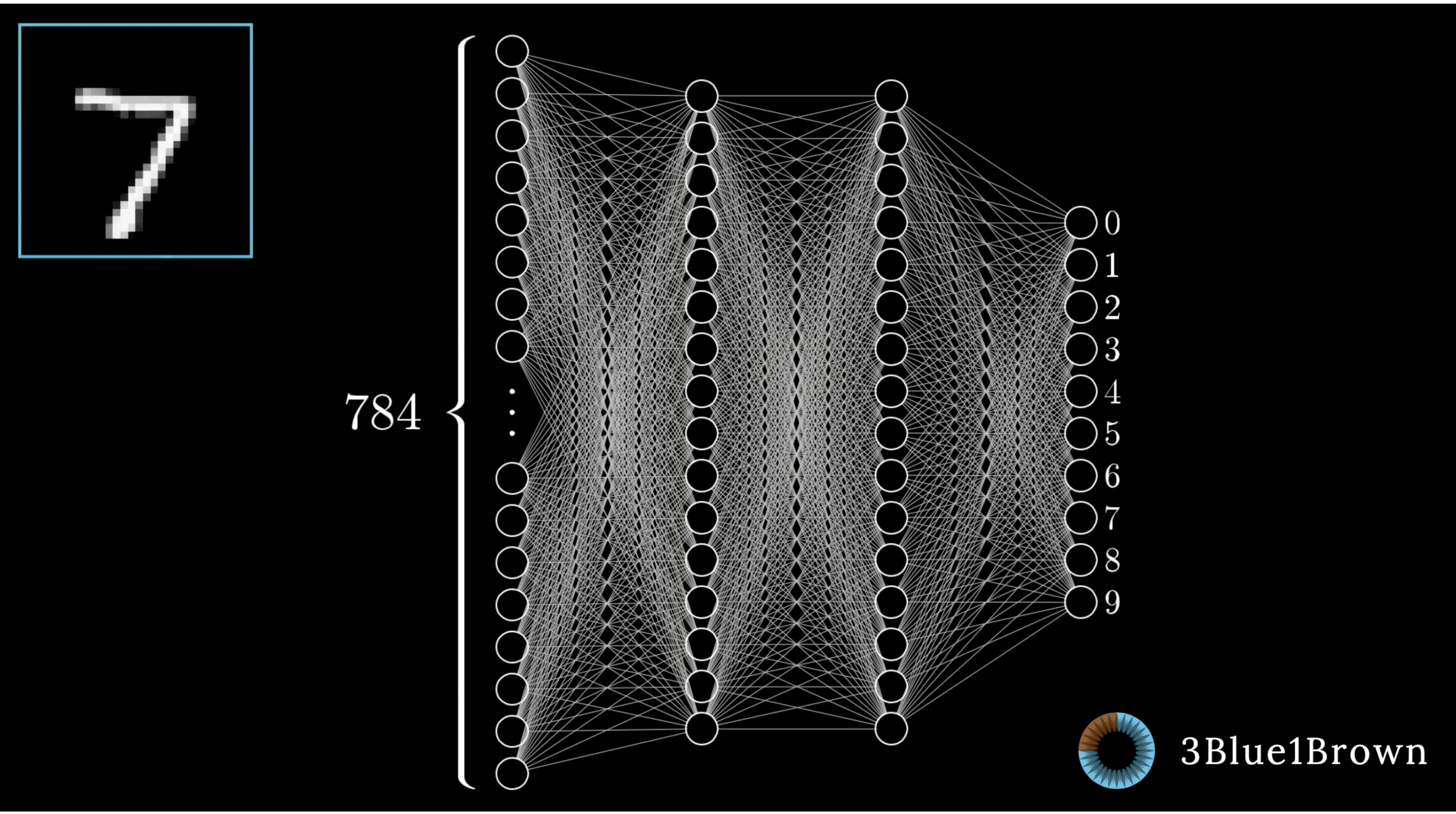
Extreme Learning Machine (ELM)



Echo State Network (ESN)

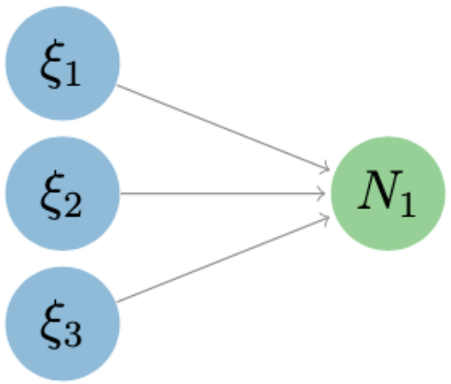


Feed-forward Neural Networks

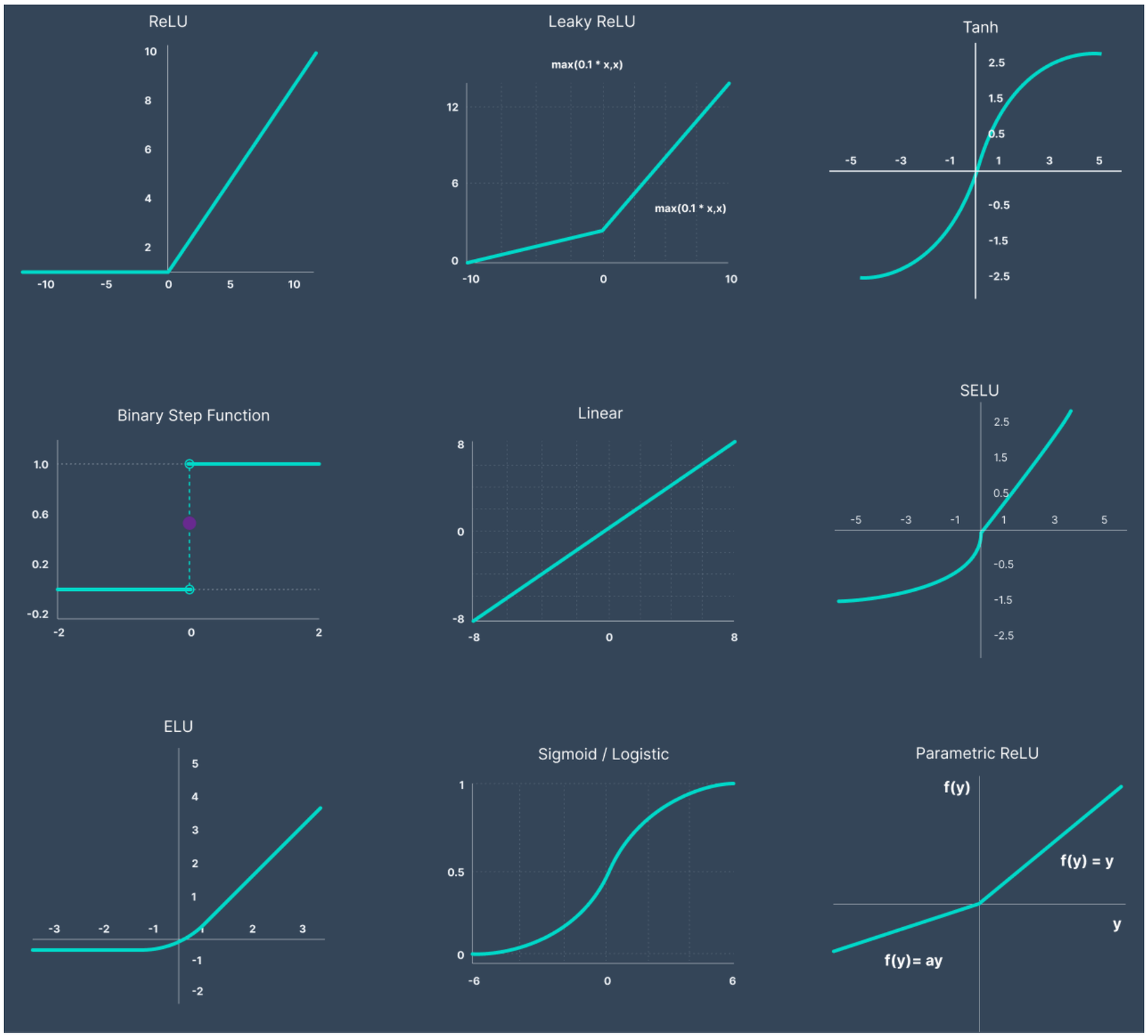


Feed-forward Neural Networks

Elementary Neural Network

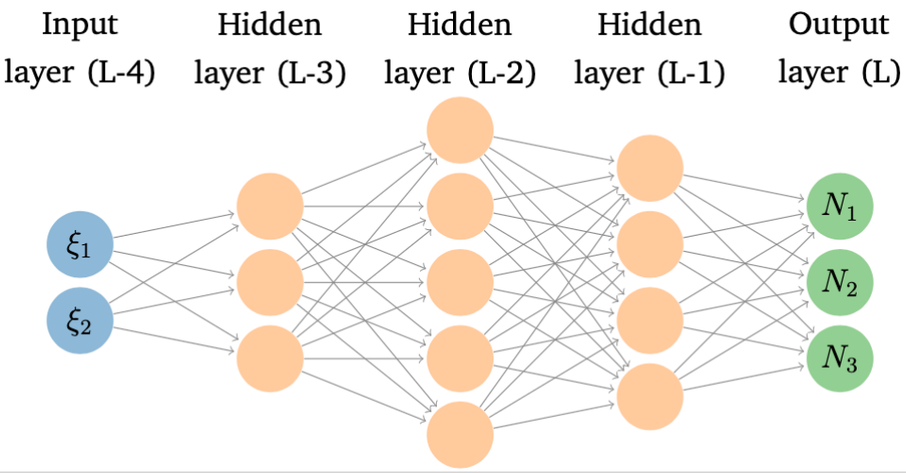


$$N_1(\boldsymbol{\xi}; \{w_{1i}, \theta_1\}) = \phi\left(\sum_i^3 w_{1i}\xi_i + \theta_1\right)$$



How does f/b-propagation work?

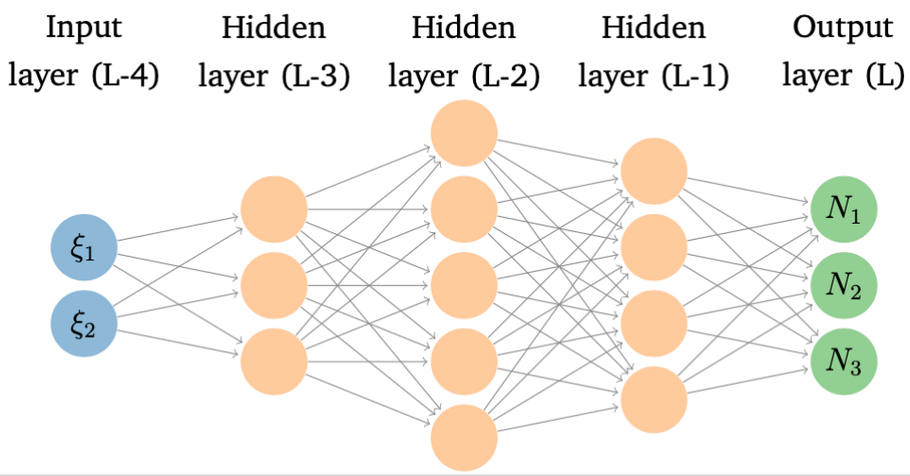
Forward-propagation



$$
\begin{aligned}
N_k(\boldsymbol{\xi}; \{\omega_{ij}^{(\ell)}, \theta_i^{(\ell)}\}) &= \phi_L \left(\sum_{j^{(1)}}^{N_{L-1}} \omega_{kj^{(1)}}^{(L)} y_{j^{(1)}}^{(L-1)} + \theta_k^{(L)} \right) \\
&= \phi_L \left(\sum_{j^{(1)}=1}^{N_{L-1}} \omega_{kj^{(1)}}^{(L)} \phi_{L-1} \left(\sum_{j^{(2)}=1}^{N_{L-2}} \omega_{j^{(1)}j^{(2)}}^{(L)} y_{j^{(2)}}^{(L-2)} + \theta_{j^{(1)}}^{(L-1)} \right) + \theta_k^{(L)} \right) \\
&= \dots .
\end{aligned}
$$

How does f/b-propagation work?

Forward-propagation



$$
\begin{aligned}
N_k(\boldsymbol{\xi};\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\}) &= \phi_L\left(\sum_{j^{(1)}}^{N_{L-1}}\omega_{kj^{(1)}}^{(L)}y_{j^{(1)}}^{(L-1)}+\theta_k^{(L)}\right)\\
&= \phi_L\left(\sum_{j^{(1)}=1}^{N_{L-1}}\omega_{kj^{(1)}}^{(L)}\phi_{L-1}\left(\sum_{j^{(2)}=1}^{N_{L-2}}\omega_{j^{(1)}j^{(2)}}^{(L)}y_{j^{(2)}}^{(L-2)}+\theta_{j^{(1)}}^{(L-1)}\right)+\theta_k^{(L)}\right)\\
&= \ldots .
\end{aligned}
$$

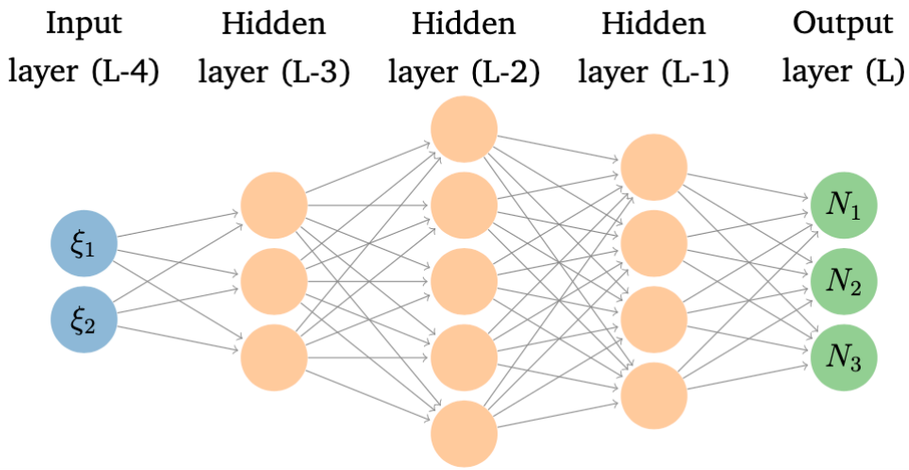
backward-propagation

$$
\chi^2[\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\}]=\sum_{k=1}^{n}\left(\frac{N_k(\boldsymbol{\xi};\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\})-\sigma_k}{s_k}\right)^2
$$

$$
\frac{\partial\chi^2}{\partial\omega_{ij}^{(\ell)}}=2\sum_{k=1}^{n}\left(\frac{N_k(\boldsymbol{\xi};\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\})-\sigma_k}{s_k^2}\right)\frac{\partial N_k}{\partial\omega_{ij}^{(\ell)}}
$$

How does f/b-propagation work?

Forward-propagation



$$N_k(\boldsymbol{\xi};\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\}) = \phi_L\left(\sum_{j^{(1)}}^{N_{L-1}}\omega_{kj^{(1)}}^{(L)}y_{j^{(1)}}^{(L-1)}+\theta_k^{(L)}\right)$$

$$= \phi_L\left(\sum_{j^{(1)}=1}^{N_{L-1}}\omega_{kj^{(1)}}^{(L)}\phi_{L-1}\left(\sum_{j^{(2)}=1}^{N_{L-2}}\omega_{j^{(1)}j^{(2)}}^{(L)}y_{j^{(2)}}^{(L-2)}+\theta_{j^{(1)}}^{(L-1)}\right)+\theta_k^{(L)}\right)$$

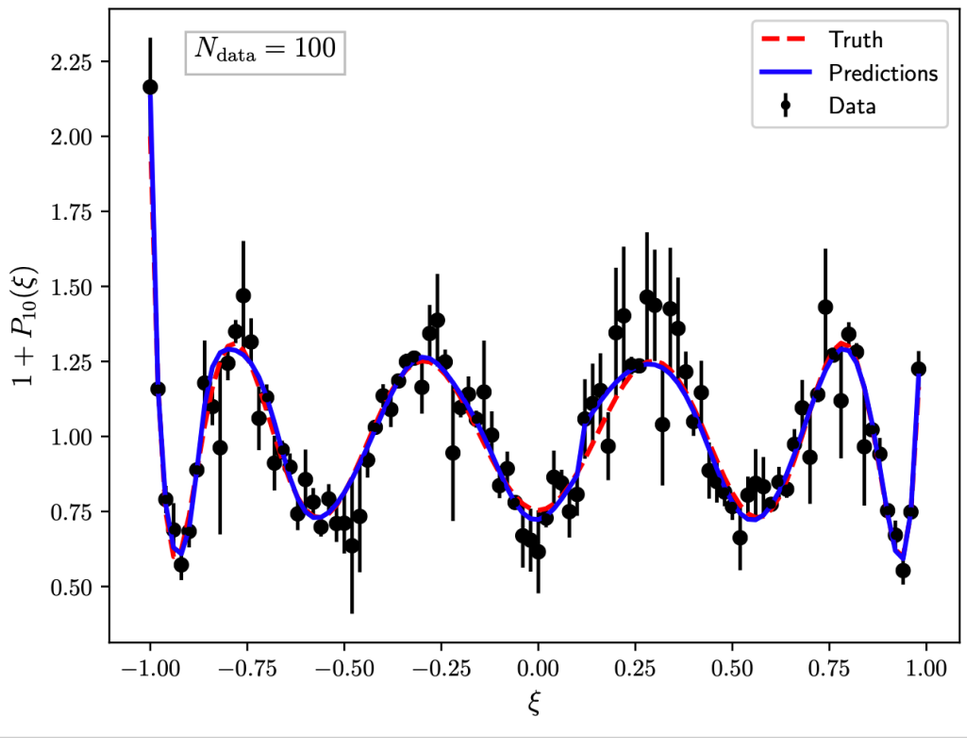
$$= \dots .$$

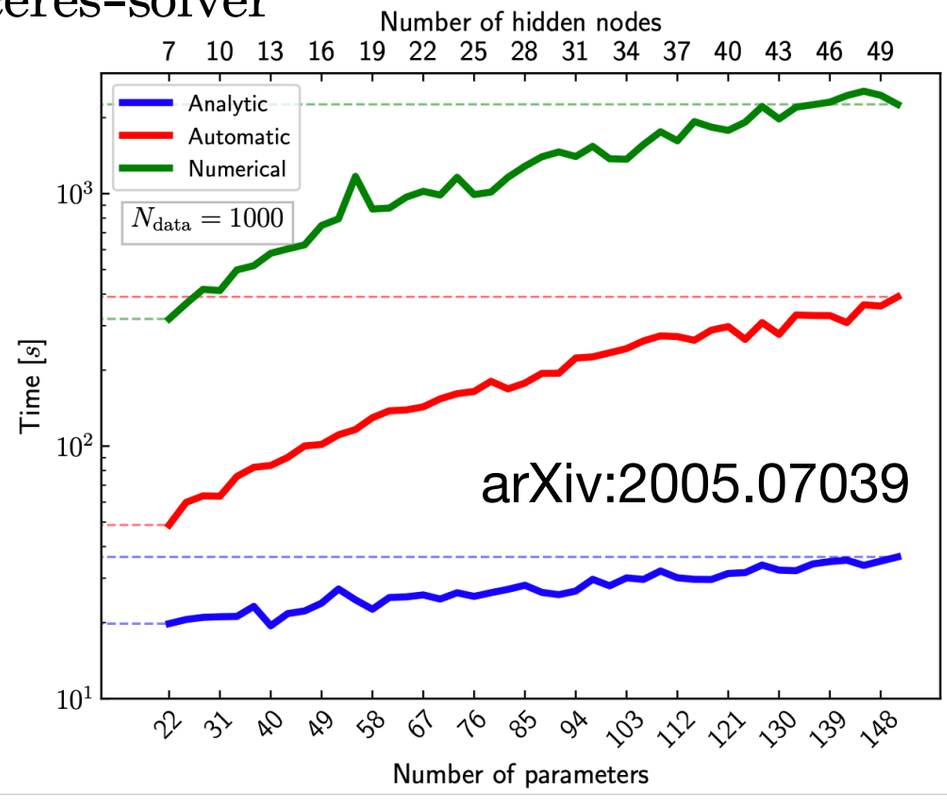
backward-propagation

$$\chi^2[\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\}] = \sum_{k=1}^{n}\left(\frac{N_k(\boldsymbol{\xi};\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\})-\sigma_k}{s_k}\right)^2$$

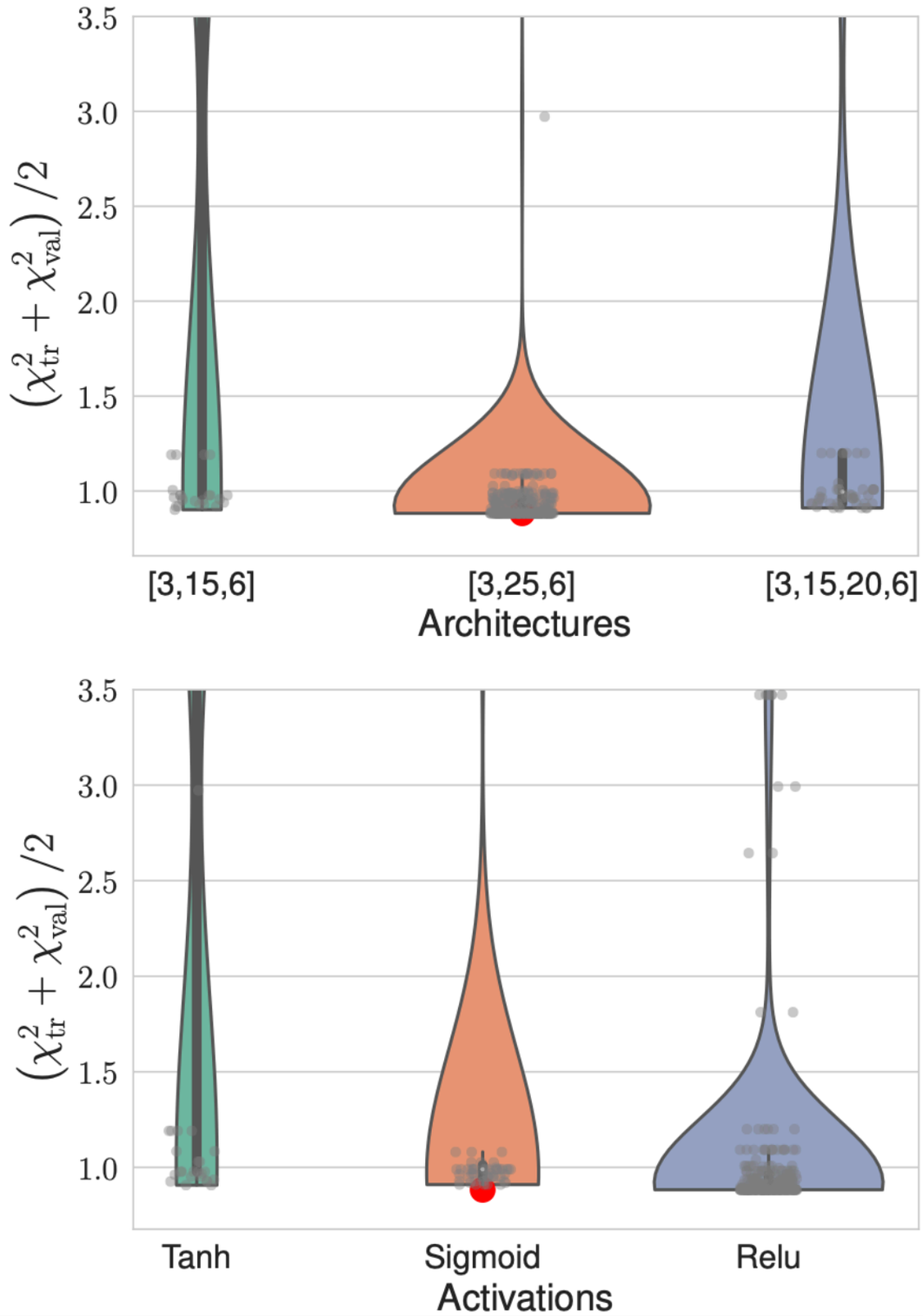
$$\frac{\partial\chi^2}{\partial\omega_{ij}^{(\ell)}} = 2\sum_{k=1}^{n}\left(\frac{N_k(\boldsymbol{\xi};\{\omega_{ij}^{(\ell)},\theta_i^{(\ell)}\})-\sigma_k}{s_k^2}\right)\frac{\partial N_k}{\partial\omega_{ij}^{(\ell)}}$$

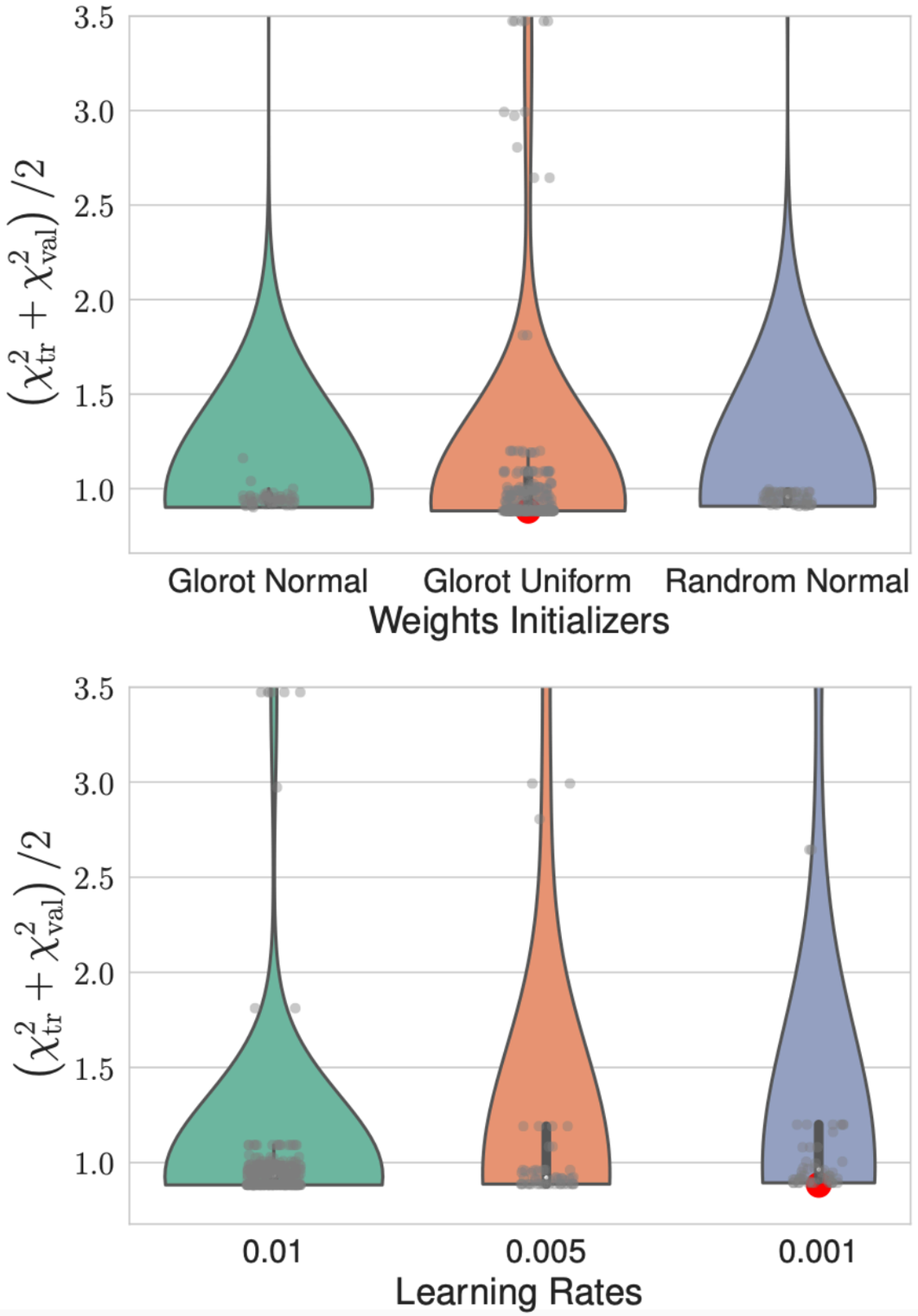
NNAD C++ library & ceres-solver





Hyper-optimisation





Monte-Carlo Inference

- **The Gaussian assumption: (data distribution)**

$$\mathcal{G}\left(\boldsymbol{x}^{(k)}\right) \propto \exp\left[\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)^{T}\cdot\boldsymbol{C}^{-1}\cdot\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)\right]$$

- **Covariance Matrix:**

$$C_{ij}=\delta_{ij}\sigma^{2}_{i,\,\mathrm{unc}}+\sum_{\beta}\sigma^{(\beta)}_{i,\,\mathrm{corr}}\sigma^{(\beta)}_{j,\,\mathrm{corr}}$$

Monte-Carlo Inference

- **The Gaussian assumption: (data distribution)**

$$\mathcal{G}\left(\boldsymbol{x}^{(k)}\right) \propto \exp\left[\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)^{T}\cdot\boldsymbol{C}^{-1}\cdot\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)\right]$$

- **Covariance Matrix:**

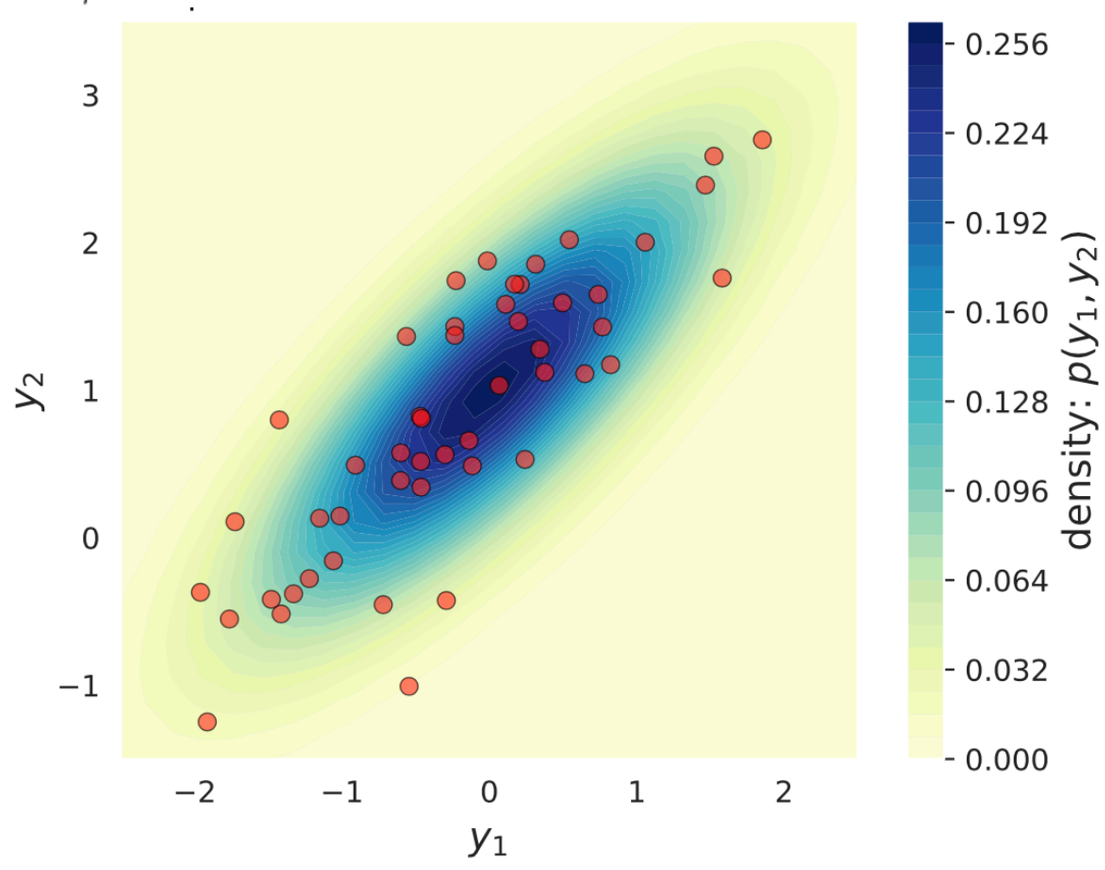
$$C_{ij} = \delta_{ij}\sigma^2_{i,\,\mathrm{unc}} + \sum_{\beta}\sigma^{(\beta)}_{i,\,\mathrm{corr}}\sigma^{(\beta)}_{j,\,\mathrm{corr}}$$

- **Replica generation:**

$$\boldsymbol{x}^{(k)} = \boldsymbol{\mu} + \boldsymbol{L}\cdot\boldsymbol{r}^{(k)}$$
$$(\boldsymbol{C} = \boldsymbol{L}\cdot\boldsymbol{L}^{T})$$

$$\rightarrow \quad \frac{1}{N_{\mathrm{rep}}}\sum_{k}^{N_{\mathrm{rep}}} x_i^{(k)} \simeq \mu_i\,,$$

$$\rightarrow \quad \frac{1}{N_{\mathrm{rep}}}\sum_{k}^{N_{\mathrm{rep}}} x_i^{(k)}x_j^{(k)} \simeq \mu_i\mu_j + C_{ij}$$



Monte-Carlo Inference

- **The Gaussian assumption: (data distribution)**

$$\mathcal{G}\left(\boldsymbol{x}^{(k)}\right) \propto \exp\left[\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)^{T}\cdot\boldsymbol{C}^{-1}\cdot\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)\right]$$

- **Covariance Matrix:**

$$C_{ij}=\delta_{ij}\sigma^{2}_{i,\,\mathrm{unc}}+\sum_{\beta}\sigma^{(\beta)}_{i,\,\mathrm{corr}}\sigma^{(\beta)}_{j,\,\mathrm{corr}}$$

- **Replica generation:**

$$\boldsymbol{x}^{(k)}=\boldsymbol{\mu}+\boldsymbol{L}\cdot\boldsymbol{r}^{(\boldsymbol{k})}$$

$$(\boldsymbol{C}=\boldsymbol{L}\cdot\boldsymbol{L}^{T})$$

- **Maximum-Likelihood estimate:**

$$\max_{\boldsymbol{\theta}}\ln\left(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{\mu})\right)\rightarrow\min_{\boldsymbol{\theta}}\chi^{2}(\boldsymbol{\theta})\rightarrow\hat{\theta}$$

$$\max_{\theta^{(1)}}\{\ln[L(\theta^{(1)}\,|\,x^{(1)})]\}\rightarrow\min_{\theta^{(1)}}\chi^{2}(\theta^{(1)})\rightarrow\hat{\theta}^{(1)}$$

$$\vdots$$

$$\max_{\theta^{(k)}}\{\ln[L(\theta^{(k)}\,|\,x^{(k)})]\}\rightarrow\min_{\theta^{(1)}}\chi^{2}(\theta^{(k)})\rightarrow\hat{\theta}^{(k)}$$

$$\vdots$$

$$\max_{\theta^{(N_{rep})}}\{\ln[L(\theta^{(N_{rep})}\,|\,x^{(N_{rep})})]\}\rightarrow\min_{\theta^{(N_{rep})}}\chi^{2}(\theta^{(1)})\rightarrow\hat{\theta}^{(N_{rep})}$$

Monte-Carlo Inference

- **The Gaussian assumption: (data distribution)**

$$\mathcal{G}\left(\boldsymbol{x}^{(k)}\right) \propto \exp\left[\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)^{T}\cdot\boldsymbol{C}^{-1}\cdot\left(\boldsymbol{x}^{(k)}-\boldsymbol{\mu}\right)\right]$$

- **Covariance Matrix:**

$$C_{ij}=\delta_{ij}\sigma^{2}_{i,\,\mathrm{unc}}+\sum_{\beta}\sigma^{(\beta)}_{i,\,\mathrm{corr}}\sigma^{(\beta)}_{j,\,\mathrm{corr}}$$

- **Replica generation:**

$$\boldsymbol{x}^{(k)}=\boldsymbol{\mu}+\boldsymbol{L}\cdot\boldsymbol{r}^{(k)}$$

$$(\boldsymbol{C}=\boldsymbol{L}\cdot\boldsymbol{L}^{T})$$

- **Maximum-Likelihood estimate:**

$$\max_{\boldsymbol{\theta}}\ln\left(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{\mu})\right)\rightarrow\min_{\boldsymbol{\theta}}\chi^{2}(\boldsymbol{\theta})\rightarrow\hat{\theta}$$

In order to find the maximum of the log-likelihood $\ln\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{\sigma})$ subject to N_{EC} equality constraints $\{(h_1(\boldsymbol{\theta})=0),\cdots,(h_{N_{\mathrm{EC}}}(\boldsymbol{\theta})=0)\}$, the method of Lagrange multipliers state that one should instead find the maximum of the Lagrange function defined as:

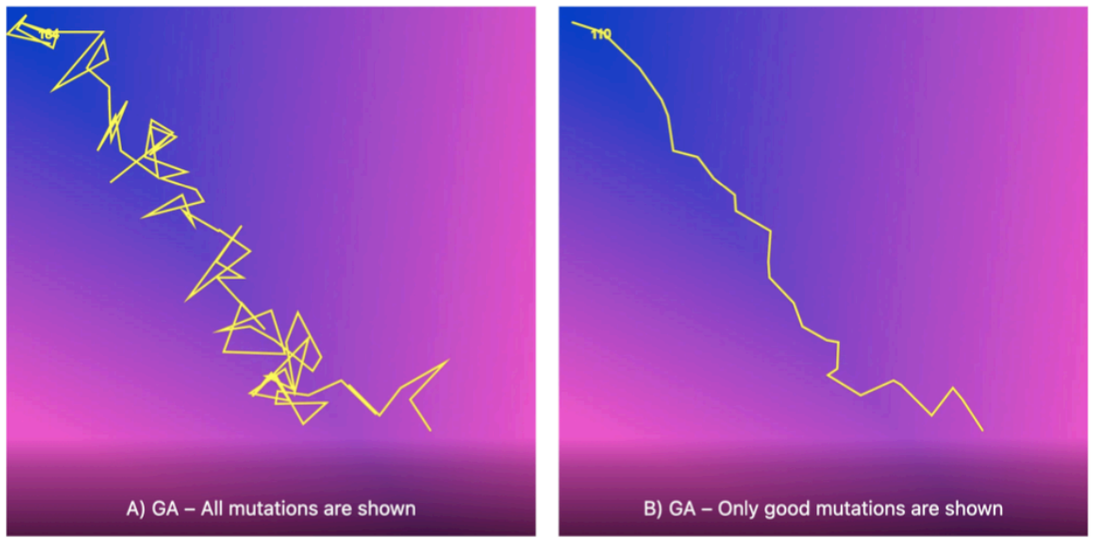
$$L(\boldsymbol{\theta},\boldsymbol{\sigma},\lambda)=\ln\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{\sigma})-\sum_{i}^{N_{\mathrm{EC}}}\lambda_i h_i(\boldsymbol{\theta})$$

$$\max_{\theta}L(\boldsymbol{\theta},\boldsymbol{\sigma},\lambda)=\min_{\theta}\left[\chi^{2}(\boldsymbol{\theta}|\boldsymbol{\sigma})+\sum_{i}^{N_{\mathrm{EC}}}\lambda_i h_i(\boldsymbol{\theta})\right]$$

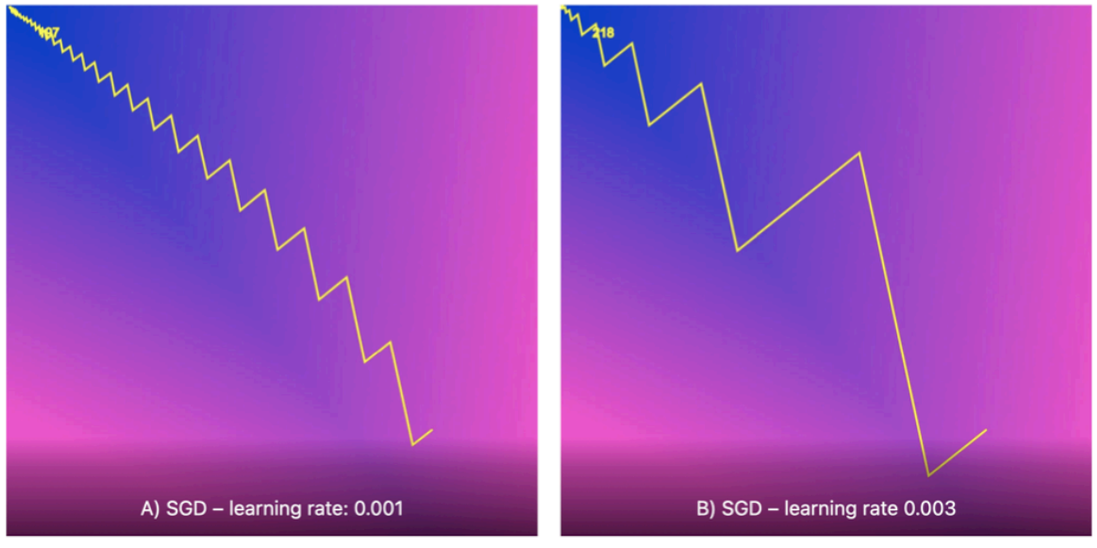
where λ_i are called the lagrange multipliers that can be chosen in a way to determine the size of the constraint contribution during the minimisation.

Minimisation

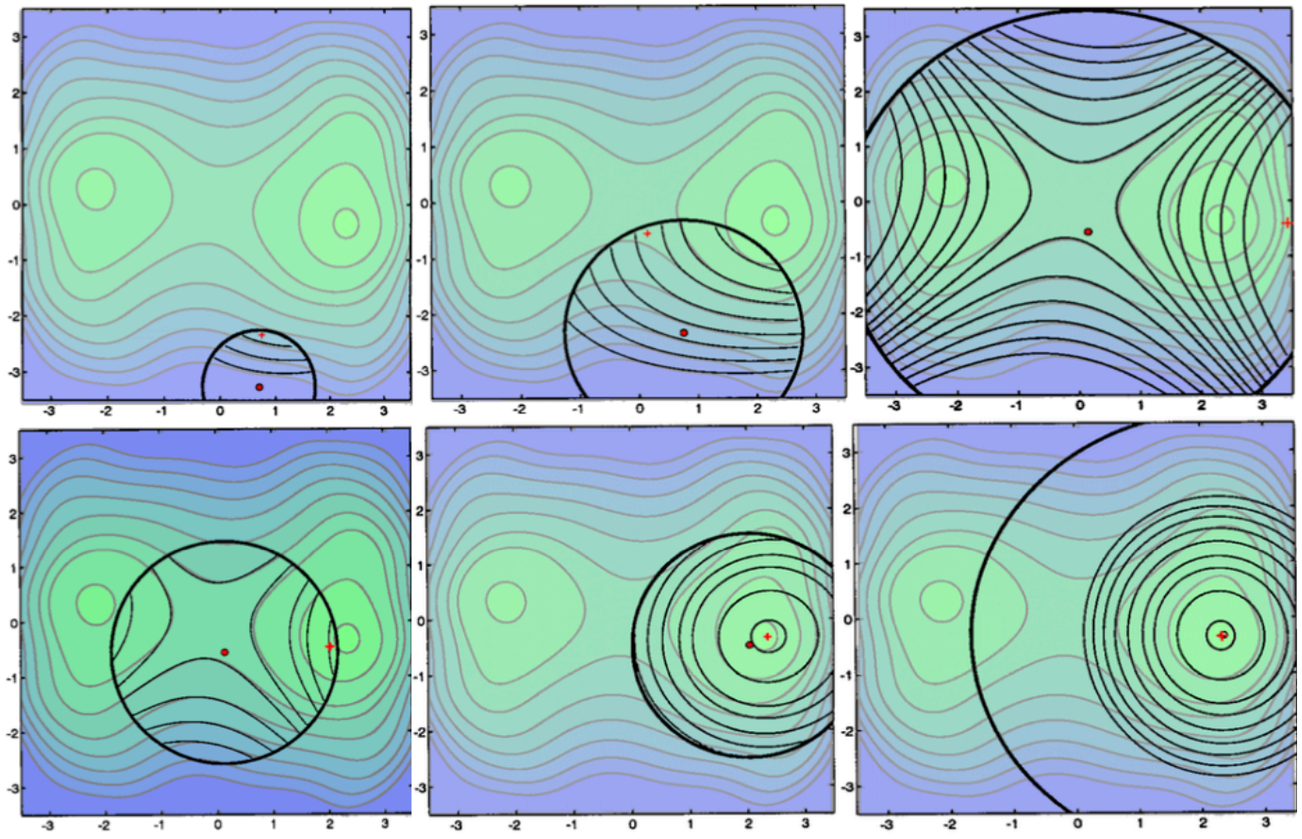
Genetic Algorithms



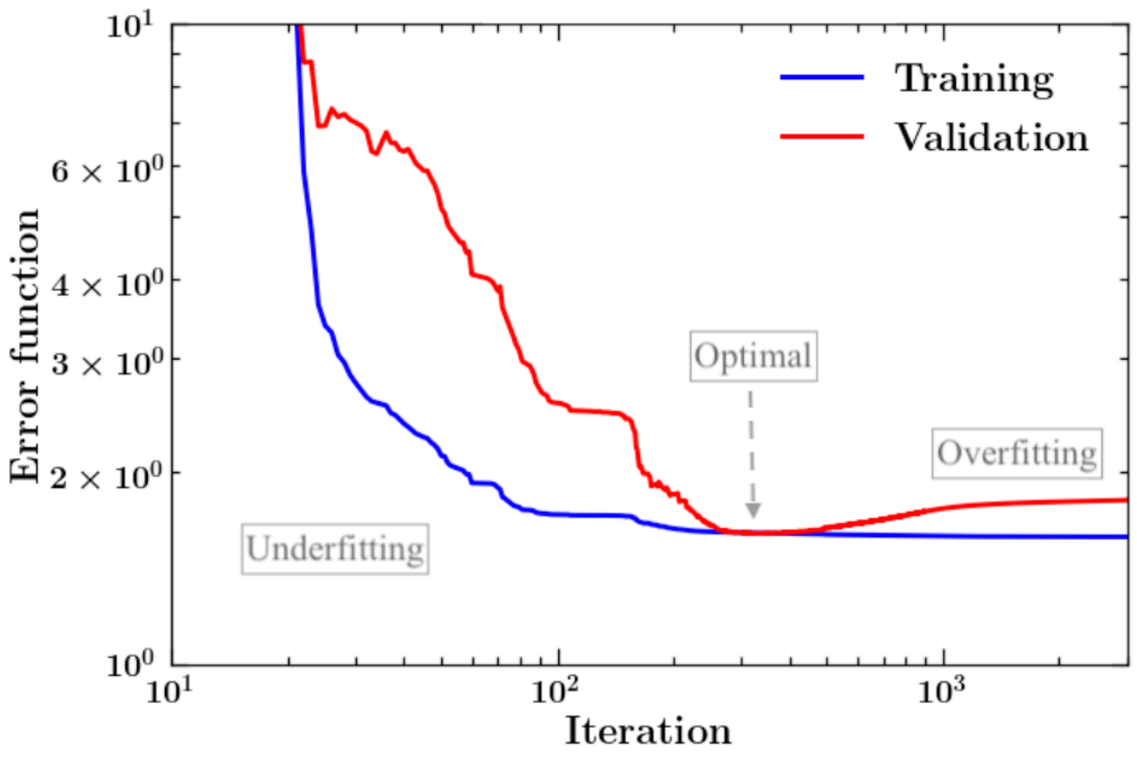
Gradient descent algorithms



Trust-region algorithms



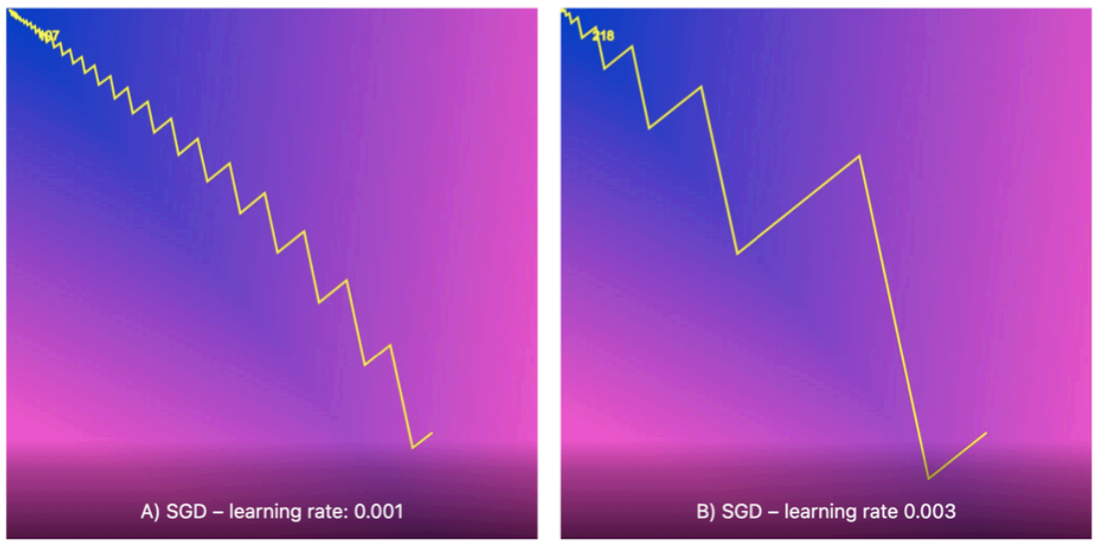
Cross-validation



Tutorial 1 — Minimisation

Time for Tutorial 1!

Gradient descent algorithms



HUGS 2022

Tutorial 1: Optimisation with Stochastic Gradient Descent in 1D

Author and references

Author: Daniel Newman - https://github.com/dtnewman
Edits: Rabah Abdul Khalek - khalek@jlab.org (Postdoc at JLab | Theory division)

Learning Goals

This notebook will serve as an introduction to **Gradient descent**, also known as **steepest descent**, an optimization algorithm for finding the local minimum of a function.