Machine learning for Quantum Noise Benchmarking

Stefano Martina
stefano.martina@unifi.it

Quantum Driving And Bio-complexity

UNIVERSITÀ DEGLI STUDI FIRENZE

15 December 2020

Overview

Introduction

- ✔ We consider a Quantum Random Walker on a complex graph
  - perturbed by noise
- ✔ Discriminate quantum noise, by measuring only walker populations
  - Support Vector Machines
  - Recurrent Neural Networks
- ✔ The dynamic parameters are crucial to the classification capacity
  - short evolution time / high frequency $\longrightarrow$ easy
  - long evolution time / low frequency $\longrightarrow$ hard
- ✔ Over 90% accuracy in classification between
  - two IID noises
  - two coloured noises
  - one IID VS one coloured noises

easier

Introduction

- ✔ We consider a Quantum Random Walker on a complex graph
  - perturbed by noise
- ✔ Discriminate quantum noise, by measuring only walker populations
  - Support Vector Machines
  - Recurrent Neural Networks
- ✔ The dynamic parameters are crucial to the classification capacity
  - short evolution time / high frequency $\longrightarrow$ easy
  - long evolution time / low frequency $\longrightarrow$ hard
- ✔ Over 90% accuracy in classification between
  - two IID noises
  - two coloured noises
  - one IID VS one coloured noises

easier

Introduction

- We consider a Quantum Random Walker on a complex graph
  - perturbed by noise
- Discriminate quantum noise, by measuring only walker populations
  - Support Vector Machines
  - Recurrent Neural Networks
- The dynamic parameters are crucial to the classification capacity
  - short evolution time / high frequency $\longrightarrow$ easy
  - long evolution time / low frequency $\longrightarrow$ hard
- Over 90% accuracy in classification between
  - two IID noises
  - two coloured noises
  - one IID VS one coloured noises

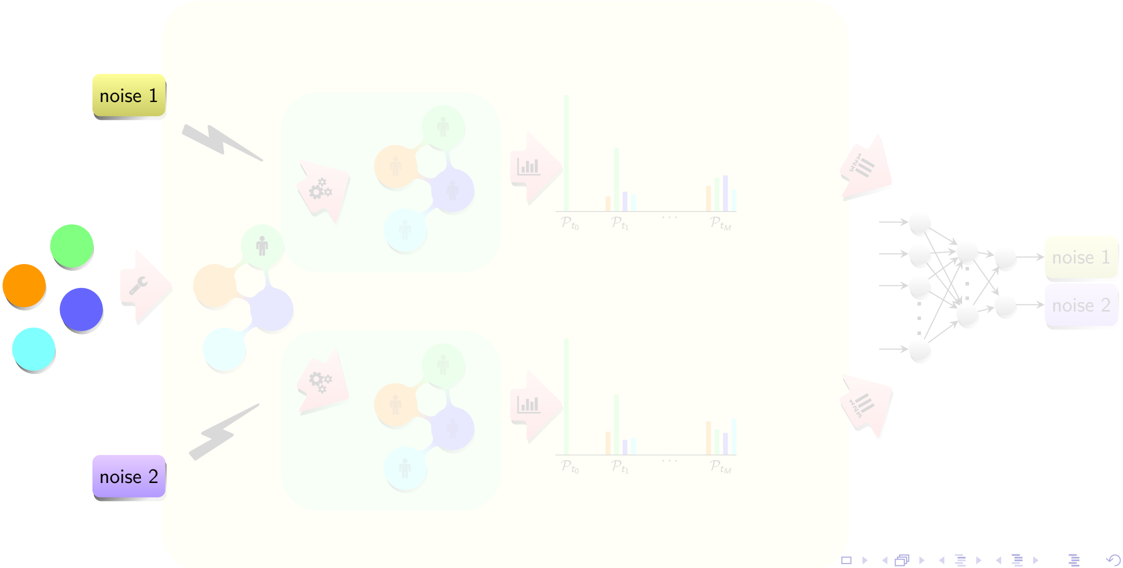
easier

Introduction

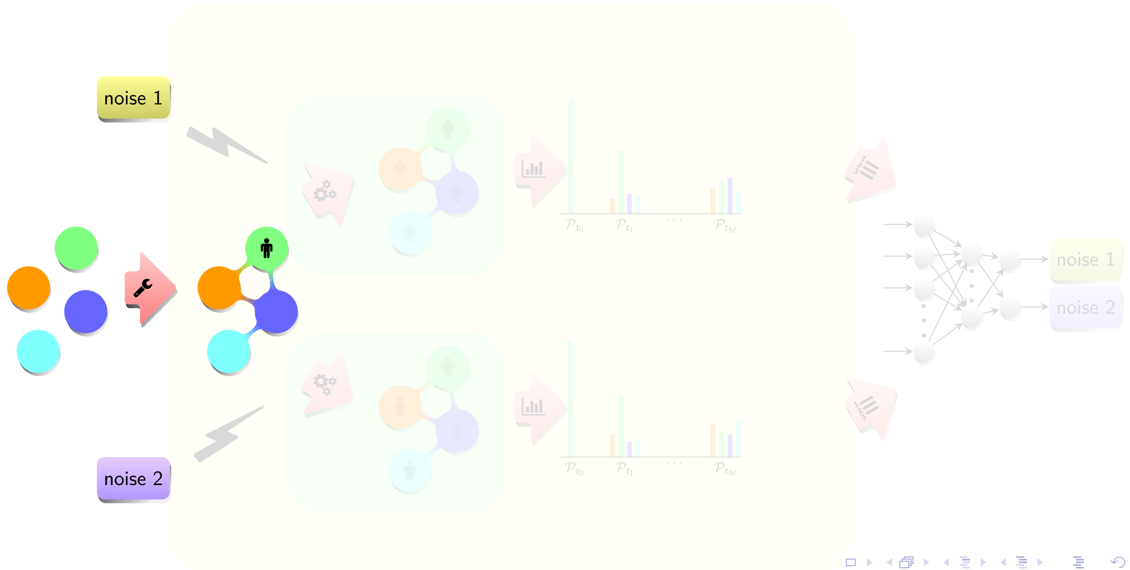
- We consider a Quantum Random Walker on a complex graph
  - perturbed by noise
- Discriminate quantum noise, by measuring only walker populations
  - Support Vector Machines
  - Recurrent Neural Networks
- The dynamic parameters are crucial to the classification capacity
  - short evolution time / high frequency $\longrightarrow$ easy
  - long evolution time / low frequency $\longrightarrow$ hard
- Over 90% accuracy in classification between
  - two IID noises
  - two coloured noises
  - one IID VS one coloured noises

easier

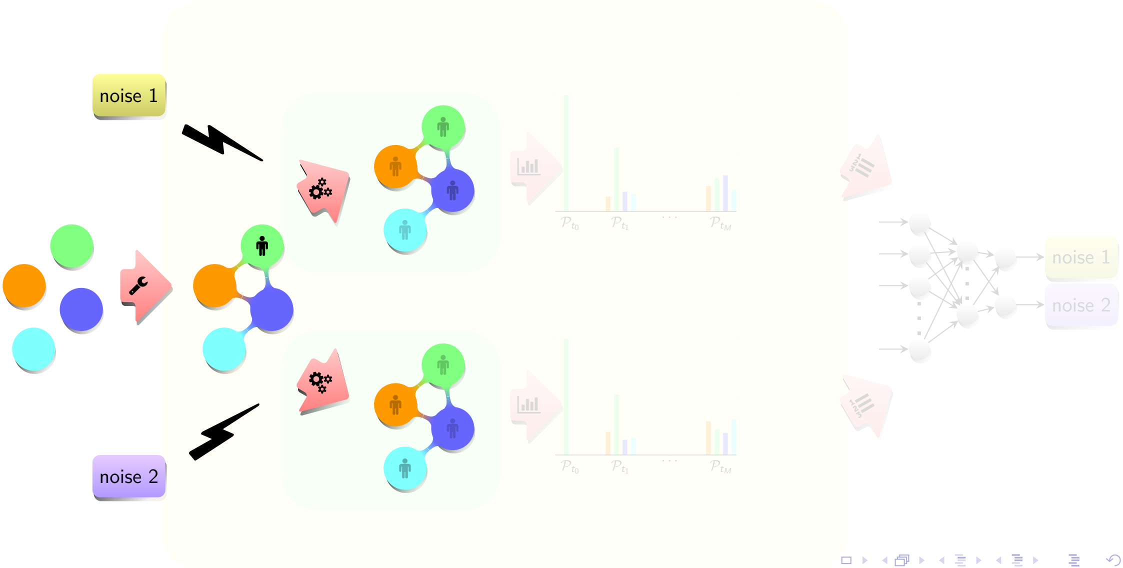
Setting definition



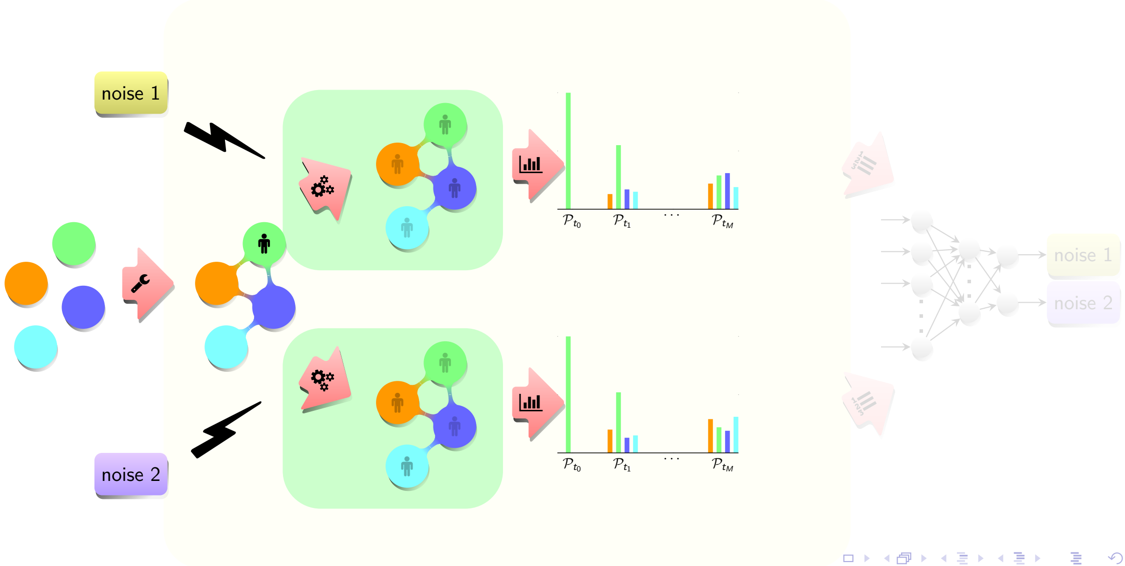
Setting definition



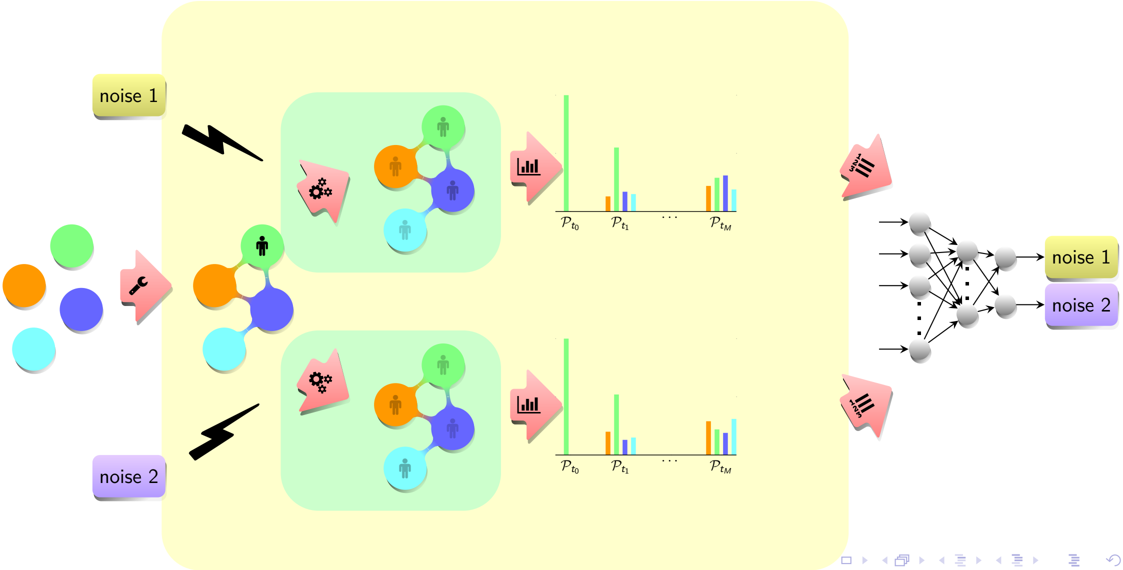
Setting definition



Setting definition



Setting definition



Example

✔ Example of populations with $t_{15} = 0.1$

	$\mathcal{P}_{t_k}^{(35)}$	$\mathcal{P}_{t_k}^{(36)}$	$\mathcal{P}_{t_k}^{(37)}$	$\mathcal{P}_{t_k}^{(38)}$	$\mathcal{P}_{t_k}^{(39)}$	$\mathcal{P}_{t_k}^{(40)}$
t_0	0.00	0.00	0.00	0.00	1.00	0.00
t_1	0.00	0.00	0.00	0.00	0.99	0.00
t_2	0.00	0.00	0.00	0.00	0.93	0.00
t_3	0.00	0.00	0.01	0.01	0.85	0.01
t_4	0.00	0.01	0.01	0.01	0.78	0.01
t_5	0.01	0.01	0.01	0.01	0.69	0.01
t_6	0.01	0.02	0.01	0.01	0.63	0.00
t_7	0.01	0.02	0.01	0.01	0.57	0.00
t_8	0.01	0.02	0.01	0.01	0.52	0.00
t_9	0.02	0.02	0.01	0.01	0.45	0.00
t_{10}	0.02	0.02	0.02	0.02	0.37	0.01
t_{11}	0.01	0.02	0.02	0.03	0.29	0.01
t_{12}	0.01	0.01	0.03	0.04	0.20	0.02
t_{13}	0.01	0.01	0.04	0.04	0.14	0.01
t_{14}	0.01	0.02	0.04	0.05	0.08	0.01
t_{15}	0.01	0.02	0.04	0.05	0.06	0.01

✔ Example of populations with $t_{15} = 1$

	$\mathcal{P}_{t_k}^{(35)}$	$\mathcal{P}_{t_k}^{(36)}$	$\mathcal{P}_{t_k}^{(37)}$	$\mathcal{P}_{t_k}^{(38)}$	$\mathcal{P}_{t_k}^{(39)}$	$\mathcal{P}_{t_k}^{(40)}$
t_0	0.00	0.00	0.00	0.00	1.00	0.00
t_1	0.02	0.02	0.01	0.01	0.45	0.00
t_2	0.01	0.01	0.02	0.03	0.05	0.02
t_3	0.00	0.00	0.00	0.00	0.13	0.02
t_4	0.02	0.01	0.01	0.01	0.12	0.02
t_5	0.01	0.01	0.03	0.03	0.06	0.01
t_6	0.01	0.01	0.01	0.01	0.01	0.00
t_7	0.04	0.03	0.01	0.06	0.11	0.00
t_8	0.04	0.00	0.03	0.11	0.11	0.03
t_9	0.03	0.00	0.03	0.01	0.01	0.10
t_{10}	0.05	0.01	0.01	0.04	0.08	0.04
t_{11}	0.01	0.03	0.02	0.00	0.08	0.02
t_{12}	0.00	0.05	0.02	0.04	0.00	0.06
t_{13}	0.01	0.03	0.00	0.02	0.05	0.07
t_{14}	0.00	0.00	0.00	0.01	0.12	0.00
t_{15}	0.00	0.00	0.01	0.04	0.10	0.01

Model

$$\hat{\mathbf{y}} = f(\mathbf{x}; \theta, \xi)$$

- f non-linear function parametrized with
  - parameters θ (modified during training)
  - hyperparameters ξ (define different configurations)

Learning

Target

$$\theta^* = \arg\min_{\theta} L_{\mathcal{D}}(\theta, \xi)$$

Theoretical risk function

$$L_{\mathcal{D}}(\theta, \xi) \equiv \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \left[\ell\left(f(\mathbf{x}; \theta, \xi), \mathbf{y}\right)\right]$$

- ✔ $\mathcal{D}$ is unknown

Learning II

Target

$$\theta^* = \arg\min_\theta L_S(\theta, \xi)$$

Empirical risk function

$$L_S(\theta, \xi) = \frac{1}{|S|} \sum_{(\mathbf{x},\mathbf{y})\in S} \ell\left(f(\mathbf{x}; \theta, \xi), \mathbf{y}\right)$$

- ✔ $S = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$ training dataset
- ✔ Monitor $L_{S'}(\theta, \xi)$ on validation set $S' \neq S$
  - Detect overfitting
  - Tune hyperparameters ξ
- ✔ Report $L_{S''}(\theta, \xi)$ (and metrics) on test set $S'' \neq S' \neq S$

Learning II

Target

$$\theta^* = \arg\min_{\theta} L_S(\theta, \xi)$$

Empirical risk function

$$L_S(\theta, \xi) = \frac{1}{|S|} \sum_{(\mathbf{x},\mathbf{y}) \in S} \ell\left(f(\mathbf{x}; \theta, \xi), \mathbf{y}\right)$$

- ✔ $S = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$ training dataset
- ✔ Monitor $L_{S'}(\theta, \xi)$ on validation set $S' \neq S$
  - Detect overfitting
  - Tune hyperparameters ξ
- ✔ Report $L_{S''}(\theta, \xi)$ (and metrics) on test set $S'' \neq S' \neq S$

Learning II

Target

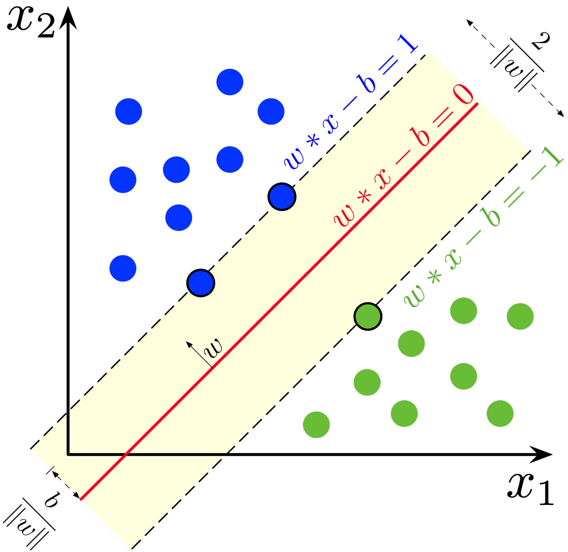
$$\theta^* = \arg\min_{\theta} L_S(\theta, \xi)$$

Empirical risk function

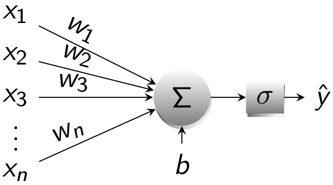
$$L_S(\theta, \xi) = \frac{1}{|S|} \sum_{(\mathbf{x},\mathbf{y})\in S} \ell\left(f(\mathbf{x}; \theta, \xi), \mathbf{y}\right)$$

- ✔ $S = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$ training dataset
- ✔ Monitor $L_{S'}(\theta, \xi)$ on validation set $S' \neq S$
  - Detect overfitting
  - Tune hyperparameters ξ
- ✔ Report $L_{S''}(\theta, \xi)$ (and metrics) on test set $S'' \neq S' \neq S$

Support Vector Machines



Multi Layer Perceptron



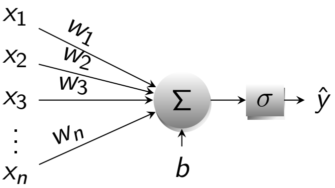
$$\hat{y} \equiv \sigma(\mathbf{w}^T \cdot \mathbf{x} + b)$$

$$\mathbf{h}[0] \equiv \mathbf{x}$$

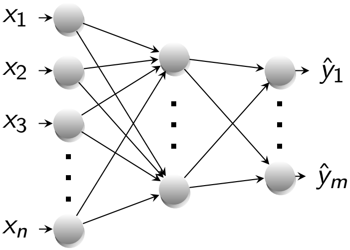
$$\mathbf{h}[l] \equiv \sigma\left(W[l]^T \cdot \mathbf{h}[l-1] + \mathbf{b}[l]\right)$$

$$\hat{\mathbf{y}} \equiv \mathbf{h}[L]$$

Multi Layer Perceptron



$$\hat{y} \equiv \sigma(\mathbf{w}^T \cdot \mathbf{x} + b)$$



$$\mathbf{h}[0] \equiv \mathbf{x}$$

$$\mathbf{h}[l] \equiv \sigma\left(W[l]^T \cdot \mathbf{h}[l-1] + \mathbf{b}[l]\right)$$

$$\hat{\mathbf{y}} \equiv \mathbf{h}[L]$$

Loss

Categorical Cross Entropy

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{j=1}^{O} y^{(j)} \log \hat{y}^{(j)}$$

- ✔ Common used in classification tasks
- ✔ Measure distance between two probability distributions
  - $\hat{\mathbf{y}}$ needs to be a probability distributions
  - obtained with softmax function:

$$\sigma^{(i)}(\mathbf{z}) = \frac{e^{z^{(i)}}}{\sum_{j=1}^{O} e^{z^{(j)}}}$$

Loss

Categorical Cross Entropy

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{j=1}^{O} y^{(j)} \log \hat{y}^{(j)}$$

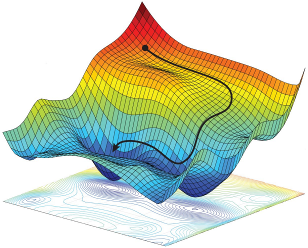
- ✔ Common used in classification tasks
- ✔ Measure distance between two probability distributions
  - $\hat{\mathbf{y}}$ needs to be a probability distributions
  - obtained with softmax function:

$$\sigma^{(i)}(\mathbf{z}) \equiv \frac{e^{z^{(i)}}}{\sum_{j=1}^{O} e^{z^{(j)}}}$$

Optimization

Stochastic Gradient Descent

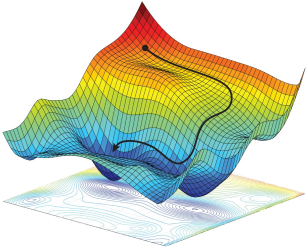
$$\theta_i = \theta_{i-1} - \eta \nabla_\theta L_{S_b}(\theta_{i-1}, \xi)$$



- ✔ η is the learning rate
  - Can also be adaptively updated (e.g. ADAM)
- ✔ S_b is a minibatch

Stochastic Gradient Descent

$$\theta_i = \theta_{i-1} - \eta \nabla_\theta L_{S_b}(\theta_{i-1}, \xi)$$

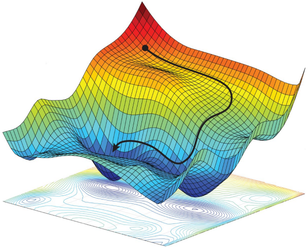


- ✔ η is the learning rate
  - Can also be adaptively updated (e.g. ADAM)
- ✔ S_b is a minibatch

Optimization

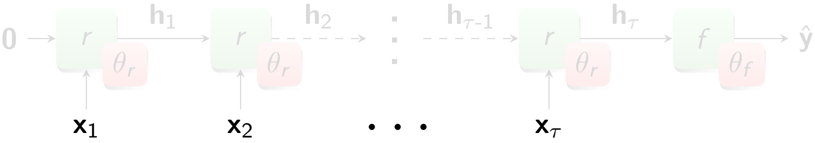
Stochastic Gradient Descent

$$\theta_i = \theta_{i-1} - \eta \nabla_\theta L_{S_b}(\theta_{i-1}, \xi)$$



- ✔ η is the learning rate
  - Can also be adaptively updated (e.g. ADAM)
- ✔ S_b is a minibatch

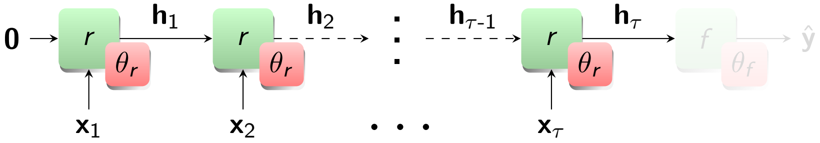
Unidirectional Recurrent Neural Network (RNN)



$$\mathbf{h}_t = r(\mathbf{x}_t, \mathbf{h}_{t-1};\, \theta_r)$$
$$\hat{\mathbf{y}} = f(\mathbf{h}_\tau;\, \theta_f)$$

- ✔ Used on sequential data
- ✔ Processed iteratively by non-linear function r
  - r parametrized with shared set of weights θ_r
  - $\mathbf{h}_t$ sort of memory
- ✔ $\mathbf{h}_\tau$ representation of all the sequence
  - In classification $\mathbf{h}_\tau$ can be processed by MLP f

Unidirectional RNN

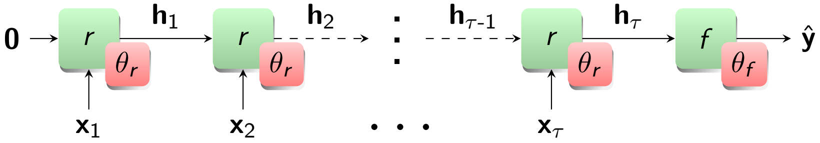


$$\mathbf{h}_t = r(\mathbf{x}_t, \mathbf{h}_{t-1};\, \theta_r)$$

$$\hat{\mathbf{y}} = f(\mathbf{h}_\tau;\, \theta_f)$$

- ✔ Used on sequential data
- ✔ Processed iteratively by non-linear function r
  - r parametrized with shared set of weights θ_r
  - $\mathbf{h}_t$ sort of memory
- ✔ $\mathbf{h}_\tau$ representation of all the sequence
  - In classification $\mathbf{h}_\tau$ can be processed by MLP f

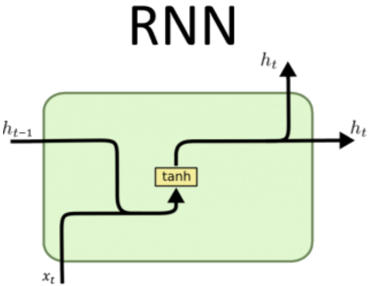
Unidirectional RNN

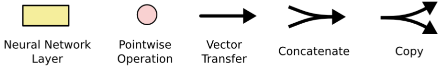


$$\mathbf{h}_t = r(\mathbf{x}_t, \mathbf{h}_{t-1};\, \theta_r)$$
$$\hat{\mathbf{y}} = f(\mathbf{h}_\tau;\, \theta_f)$$

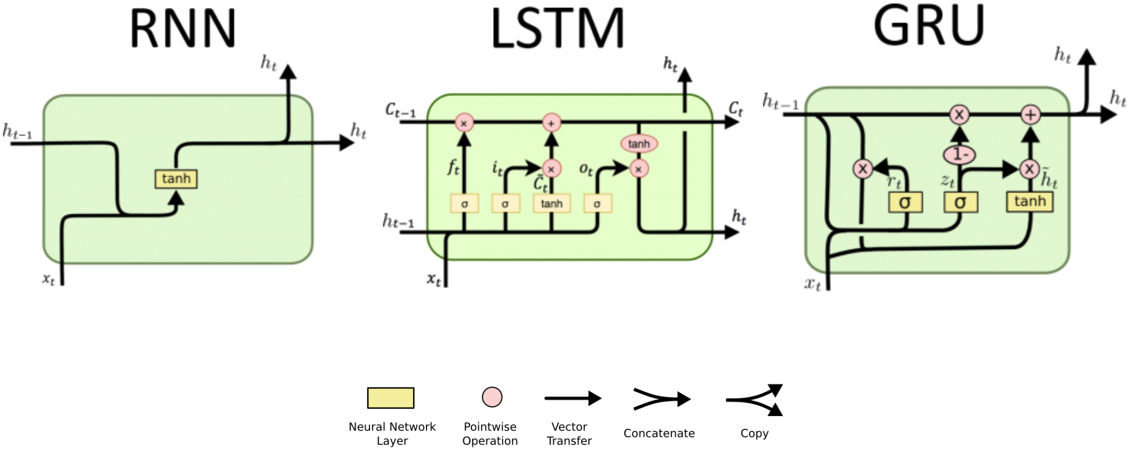
- ✔ Used on sequential data
- ✔ Processed iteratively by non-linear function r
  - r parametrized with shared set of weights θ_r
  - $\mathbf{h}_t$ sort of memory
- ✔ $\mathbf{h}_\tau$ representation of all the sequence
  - In classification $\mathbf{h}_\tau$ can be processed by MLP f

GRU/LSTM

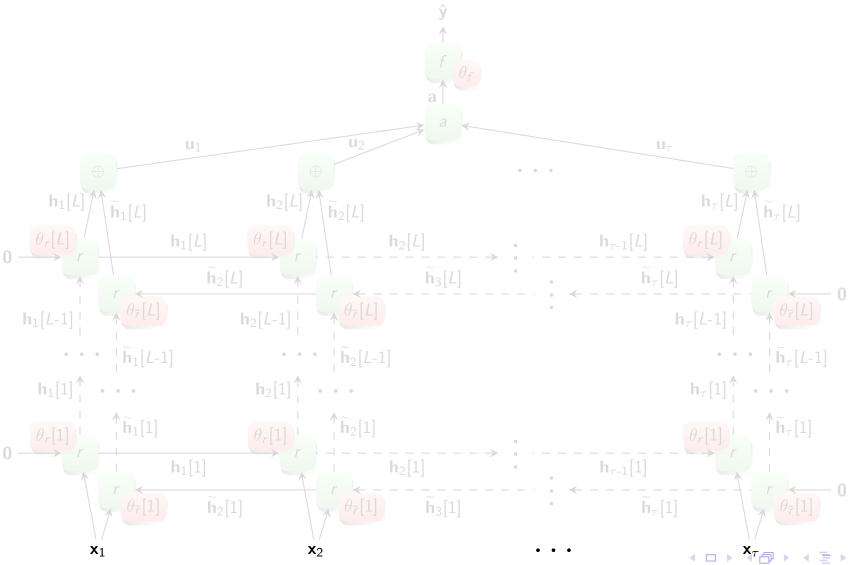

RNN
h_t
h_{t-1}
h_t
tanh
x_t


Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

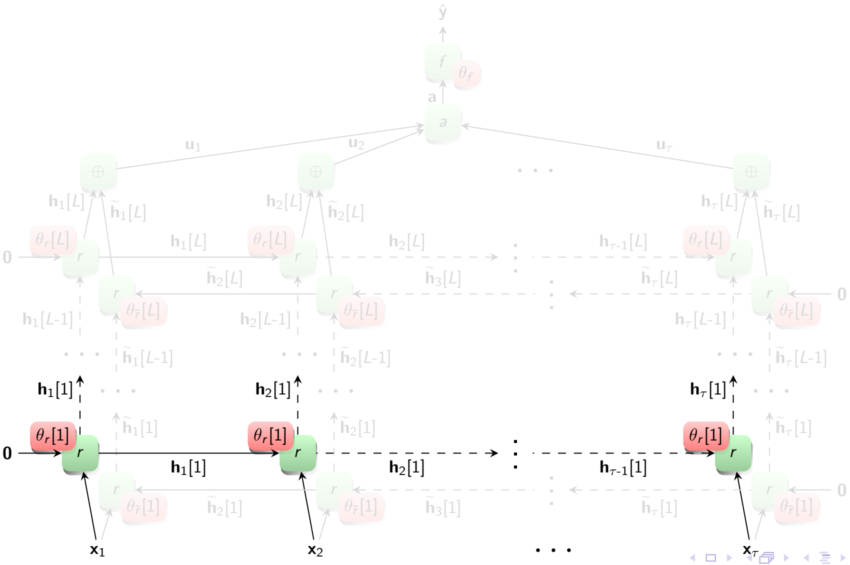
GRU/LSTM


RNN
LSTM
GRU
tanh
σ
1-
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

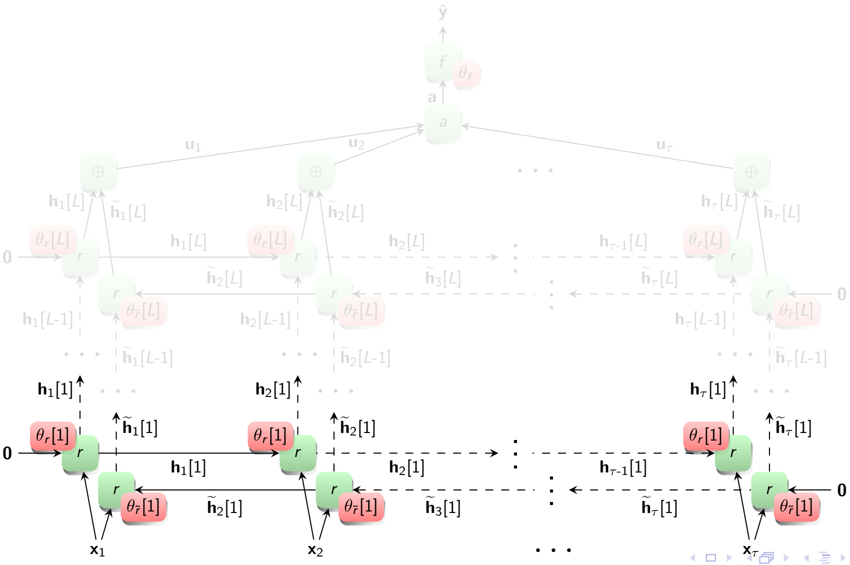
Bidirectional RNN with aggregation



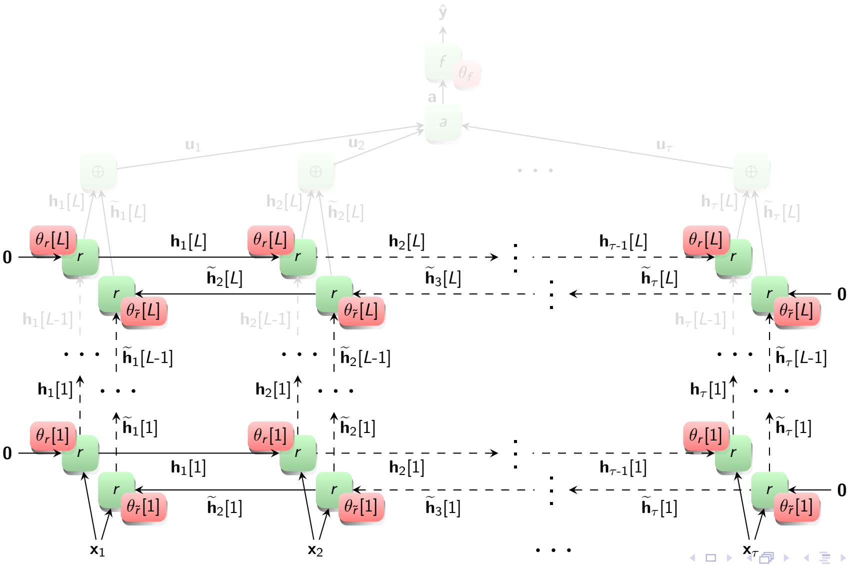
Bidirectional RNN with aggregation



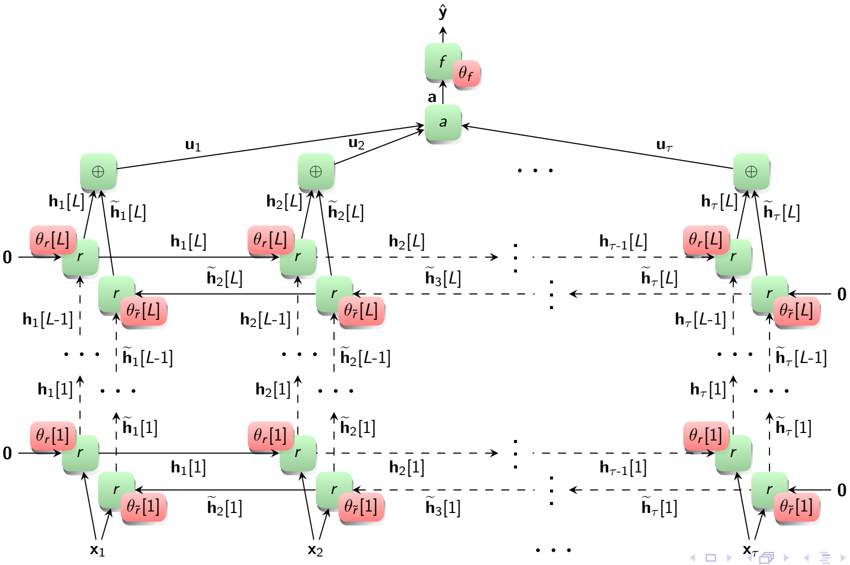
Bidirectional RNN with aggregation



Bidirectional RNN with aggregation



Bidirectional RNN with aggregation



Aggregation

Standard

$$\mathbf{a} = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_1[L]$$

Max pooling

$$\mathbf{u}_t = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_t[L]$$

$$\mathbf{a}^{(j)} = \max_t \mathbf{u}_t^{(j)}$$

Attention mechanism

$$\mathbf{u}_t = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_t[L]$$

$$\mathbf{v}_t = \tanh\left(\mathbf{W}^T \cdot \mathbf{u}_t + \mathbf{b}\right)$$

$$\alpha_t = \frac{e^{\langle \mathbf{v}_t, \mathbf{c} \rangle}}{\sum_{j=1}^{\mathcal{T}} e^{\langle \mathbf{v}_j, \mathbf{c} \rangle}}$$

$$\mathbf{a} = \sum_{t=1}^{\mathcal{T}} \alpha_t \mathbf{u}_t$$

Aggregation

Standard

$$\mathbf{a} = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_1[L]$$

Max pooling

$$\mathbf{u}_t = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_t[L]$$
$$\mathbf{a}^{(j)} = \max_t \mathbf{u}_t^{(j)}$$

Attention mechanism

$$\mathbf{u}_t = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_t[L]$$
$$\mathbf{v}_t = \tanh\left(\mathbf{W}^T \cdot \mathbf{u}_t + \mathbf{b}\right)$$
$$\alpha_t = \frac{e^{\langle \mathbf{v}_t, \mathbf{c} \rangle}}{\sum_{j=1}^{\mathcal{T}} e^{\langle \mathbf{v}_j, \mathbf{c} \rangle}}$$
$$\mathbf{a} = \sum_{t=1}^{\mathcal{T}} \alpha_t \mathbf{u}_t$$

Aggregation

Standard

$$\mathbf{a} = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_1[L]$$

Max pooling

$$\mathbf{u}_t = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_t[L]$$
$$\mathbf{a}^{(j)} = \max_t \mathbf{u}_t^{(j)}$$

Attention mechanism

$$\mathbf{u}_t = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_t[L]$$
$$\mathbf{v}_t = \tanh\left(\mathbf{W}^T \cdot \mathbf{u}_t + \mathbf{b}\right)$$
$$\alpha_t \equiv \frac{e^{\langle \mathbf{v}_t, \mathbf{c} \rangle}}{\sum_{j=1}^{T} e^{\langle \mathbf{v}_j, \mathbf{c} \rangle}}$$
$$\mathbf{a} \equiv \sum_{t=1}^{T} \alpha_t \mathbf{u}_t$$

Results

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	NM	VS	IID	NM	VS
$\mathcal{P}_{t_{15}}$	SVM	97.0	82.3	96.5	50.3	51.2	49.5
	MLP	96.9	80.7	96.6	49.5	50.7	50.2
$\mathcal{P}_{t_0}, \dots, \mathcal{P}_{t_{15}}$	SVM	96.4	80.1	96.3	73.6	61.9	75.0
	GRU	96.5	91.5	96.7	90.5	73.3	88.2
	LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	bi-GRU	96.6	92.2	96.6	91.0	74.6	90.6
	bi-LSTM	96.7	89.7	96.5	90.8	70.6	87.2
	bi-GRU-att	97.0	91.6	96.1	90.9	73.4	87.9
	bi-LSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	bi-GRU-max	96.6	92.6	96.6	91.8	76.1	90.4
	bi-LSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

✔ Paper in submission: Stefano Martina, Stefano Gherardini and Filippo Caruso. Machine learning approach for quantum non-Markovian noise classification

Results

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	**NM**	**VS**	**IID**	**NM**	**VS**
$\mathcal{P}_{t_{15}}$	**SVM**	97.0	82.3	96.5	50.3	51.2	49.5
	MLP	96.9	80.7	96.6	49.5	50.7	50.2
$\mathcal{P}_{t_0}, \dots, \mathcal{P}_{t_{15}}$	**SVM**	96.4	80.1	96.3	73.6	61.9	75.0
	GRU	96.5	91.5	96.7	90.5	73.3	88.2
	LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	bi-GRU	96.6	92.2	96.6	91.0	74.6	90.6
	bi-LSTM	96.7	89.7	96.5	90.8	70.6	87.2
	bi-GRU-att	97.0	91.6	96.1	90.9	73.4	87.9
	bi-LSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	bi-GRU-max	96.6	92.6	96.6	91.8	76.1	90.4
	bi-LSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

✔ Paper in submission: Stefano Martina, Stefano Gherardini and Filippo Caruso. Machine learning approach for quantum non-Markovian noise classification

Results

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	**NM**	**VS**	**IID**	**NM**	**VS**
$\mathcal{P}_{t_{15}}$	**SVM**	97.0	82.3	96.5	50.3	51.2	49.5
	MLP	96.9	80.7	96.6	49.5	50.7	50.2
$\mathcal{P}_{t_0}, \ldots, \mathcal{P}_{t_{15}}$	**SVM**	96.4	80.1	96.3	73.6	61.9	75.0
	GRU	96.5	91.5	96.7	90.5	73.3	88.2
	LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	bi-GRU	96.6	92.2	96.6	91.0	74.6	90.6
	bi-LSTM	96.7	89.7	96.5	90.8	70.6	87.2
	bi-GRU-att	97.0	91.6	96.1	90.9	73.4	87.9
	bi-LSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	bi-GRU-max	96.6	92.6	96.6	91.8	76.1	90.4
	bi-LSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

✔ Paper in submission: Stefano Martina, Stefano Gherardini and Filippo Caruso. Machine learning approach for quantum non-Markovian noise classification

Results

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	**NM**	**VS**	**IID**	**NM**	**VS**
$\mathcal{P}_{t_{15}}$	**SVM**	97.0	82.3	96.5	50.3	51.2	49.5
	MLP	96.9	80.7	96.6	49.5	50.7	50.2
$\mathcal{P}_{t_0}, \ldots, \mathcal{P}_{t_{15}}$	**SVM**	96.4	80.1	96.3	73.6	61.9	75.0
	GRU	96.5	91.5	96.7	90.5	73.3	88.2
	LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	bi-GRU	96.6	92.2	96.6	91.0	74.6	90.6
	bi-LSTM	96.7	89.7	96.5	90.8	70.6	87.2
	bi-GRU-att	97.0	91.6	96.1	90.9	73.4	87.9
	bi-LSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	bi-GRU-max	96.6	92.6	96.6	91.8	76.1	90.4
	bi-LSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

✔ Paper in submission: Stefano Martina, Stefano Gherardini and Filippo Caruso. Machine learning approach for quantum non-Markovian noise classification

Results

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	**NM**	**VS**	**IID**	**NM**	**VS**
$\mathcal{P}_{t_{15}}$	**SVM**	97.0	82.3	96.5	50.3	51.2	49.5
	MLP	96.9	80.7	96.6	49.5	50.7	50.2
$\mathcal{P}_{t_0}, \ldots, \mathcal{P}_{t_{15}}$	**SVM**	96.4	80.1	96.3	73.6	61.9	75.0
	GRU	96.5	91.5	96.7	90.5	73.3	88.2
	LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	bi-GRU	96.6	92.2	96.6	91.0	74.6	90.6
	bi-LSTM	96.7	89.7	96.5	90.8	70.6	87.2
	bi-GRU-att	97.0	91.6	96.1	90.9	73.4	87.9
	bi-LSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	bi-GRU-max	96.6	92.6	96.6	91.8	76.1	90.4
	bi-LSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

✔ Paper in submission: Stefano Martina, Stefano Gherardini and Filippo Caruso. Machine learning approach for quantum non-Markovian noise classification

Results

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	**NM**	**VS**	**IID**	**NM**	**VS**
$\mathcal{P}_{t_{15}}$	**SVM**	<u>97.0</u>	<u>82.3</u>	96.5	<u>50.3</u>	<u>51.2</u>	49.5
	MLP	96.9	80.7	<u>96.6</u>	49.5	50.7	<u>50.2</u>
$\mathcal{P}_{t_0}, \dots, \mathcal{P}_{t_{15}}$	**SVM**	96.4	80.1	96.3	73.6	61.9	75.0
	GRU	96.5	91.5	<u>96.7</u>	90.5	73.3	88.2
	LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	bi-GRU	96.6	92.2	96.6	91.0	74.6	<u>90.6</u>
	bi-LSTM	96.7	89.7	96.5	90.8	70.6	87.2
	bi-GRU-att	<u>97.0</u>	91.6	96.1	90.9	73.4	87.9
	bi-LSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	bi-GRU-max	96.6	<u>92.6</u>	96.6	<u>91.8</u>	<u>76.1</u>	90.4
	bi-LSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

✔ Paper in submission: Stefano Martina, Stefano Gherardini and Filippo Caruso. Machine learning approach for quantum non-Markovian noise classification

Results

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	**NM**	**VS**	**IID**	**NM**	**VS**
$\mathcal{P}_{t_{15}}$	**SVM**	<u>97.0</u>	<u>82.3</u>	96.5	<u>50.3</u>	<u>51.2</u>	49.5
	MLP	96.9	80.7	<u>96.6</u>	49.5	50.7	<u>50.2</u>
$\mathcal{P}_{t_0}, \ldots, \mathcal{P}_{t_{15}}$	**SVM**	96.4	80.1	96.3	73.6	61.9	75.0
	GRU	96.5	91.5	<u>96.7</u>	90.5	73.3	88.2
	LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	bi-GRU	96.6	92.2	96.6	91.0	74.6	<u>90.6</u>
	bi-LSTM	96.7	89.7	96.5	90.8	70.6	87.2
	bi-GRU-att	<u>97.0</u>	91.6	96.1	90.9	73.4	87.9
	bi-LSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	bi-GRU-max	96.6	<u>92.6</u>	96.6	<u>91.8</u>	<u>76.1</u>	90.4
	bi-LSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

✔ Paper in submission: Stefano Martina, Stefano Gherardini and Filippo Caruso. Machine learning approach for quantum non-Markovian noise classification

Thank you! Questions?

We have many open postdoc positions on Quantum Machine Learning, Quantum Sensing, and more in general Quantum Information Theory in QDAB, starting at any time! If you are interested, contact Prof. Filippo Caruso:

filippo.caruso@unifi.it — qdab.org