Machine learning with hybrid quantum-classical systems

Andrea Mari

Thomas R. Bromley   Josh Izaac   Maria Schuld   Nathan Killoran

XANADU

Quantum Computing and HPC – CINECA – Casalecchio di Reno, Bologna, Italy. – December 19th, 2019

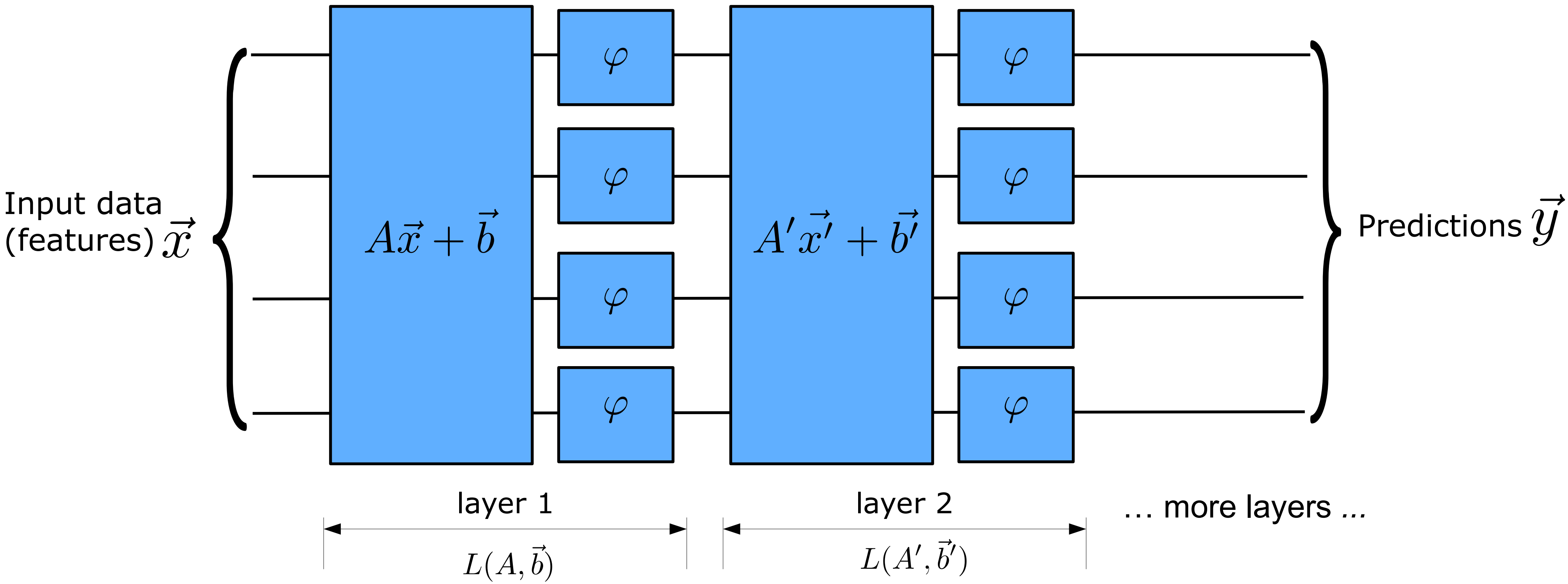
Outline

1. Machine learning with variational quantum circuits
2. Quantum transfer learning
3. Classification of high-resolution images with QPUs

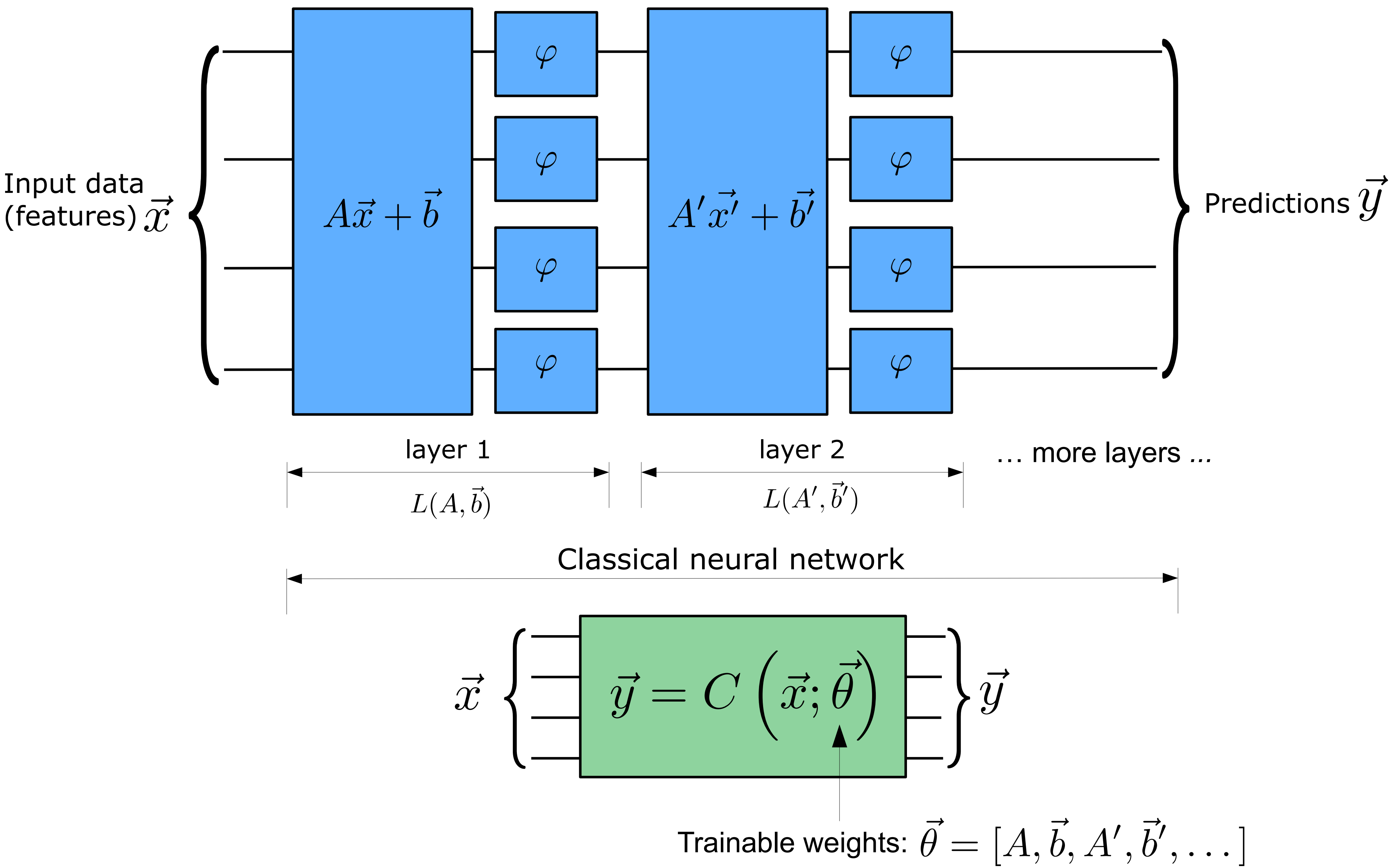
Outline

1. Machine learning with variational quantum circuits
2. Quantum transfer learning
3. Classification of high-resolution images with QPUs

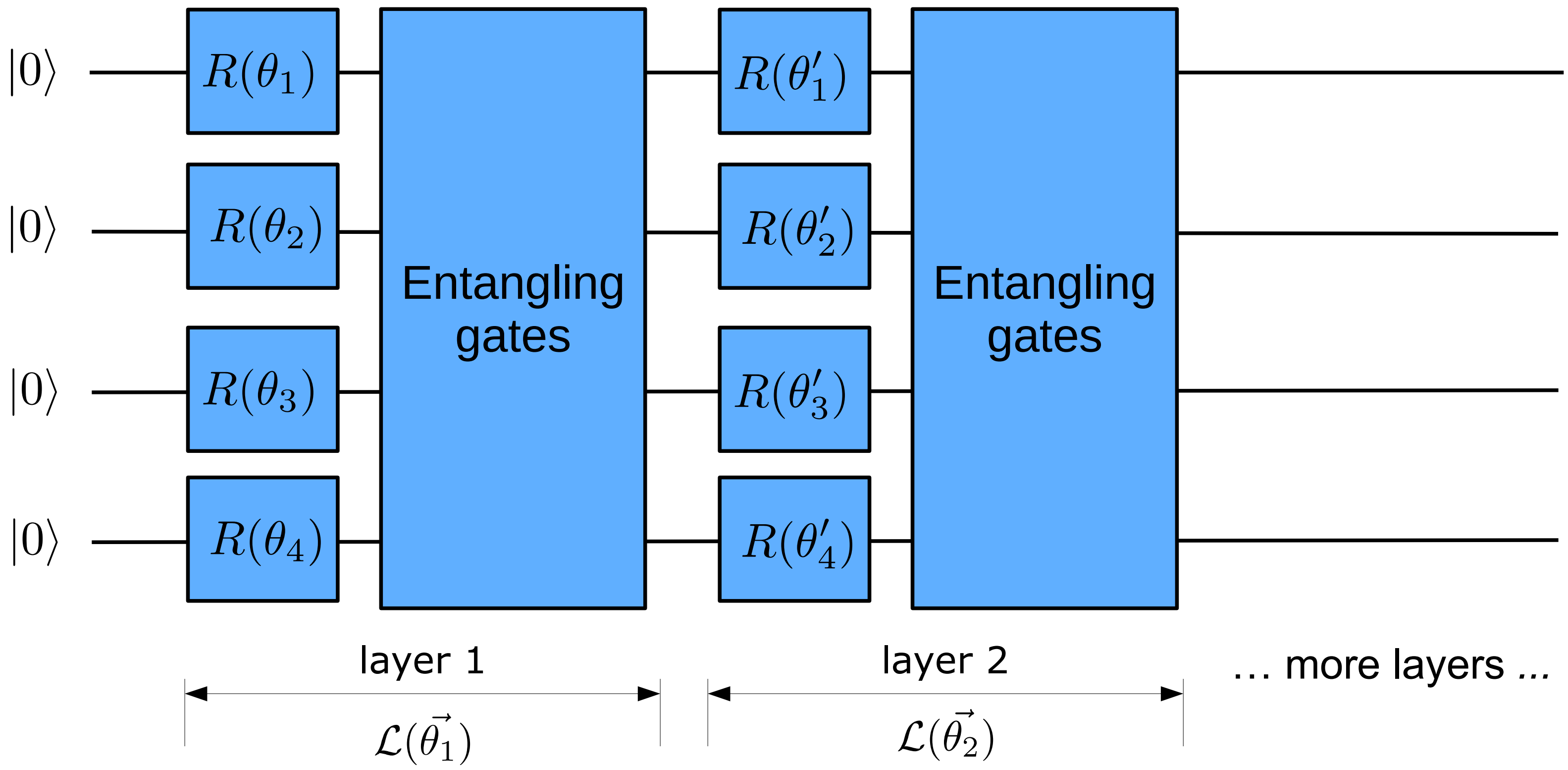
Classical neural networks



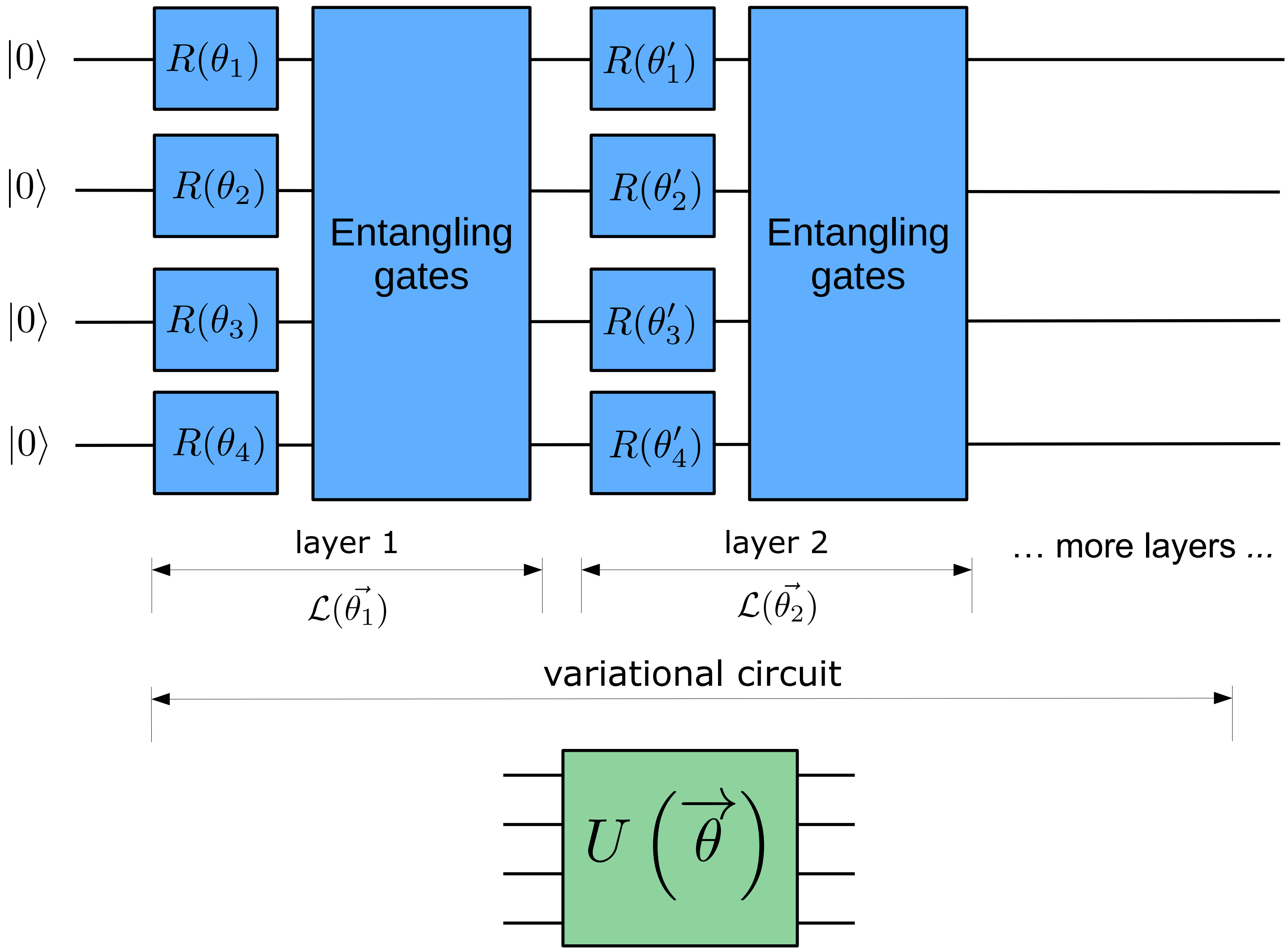
Classical neural networks



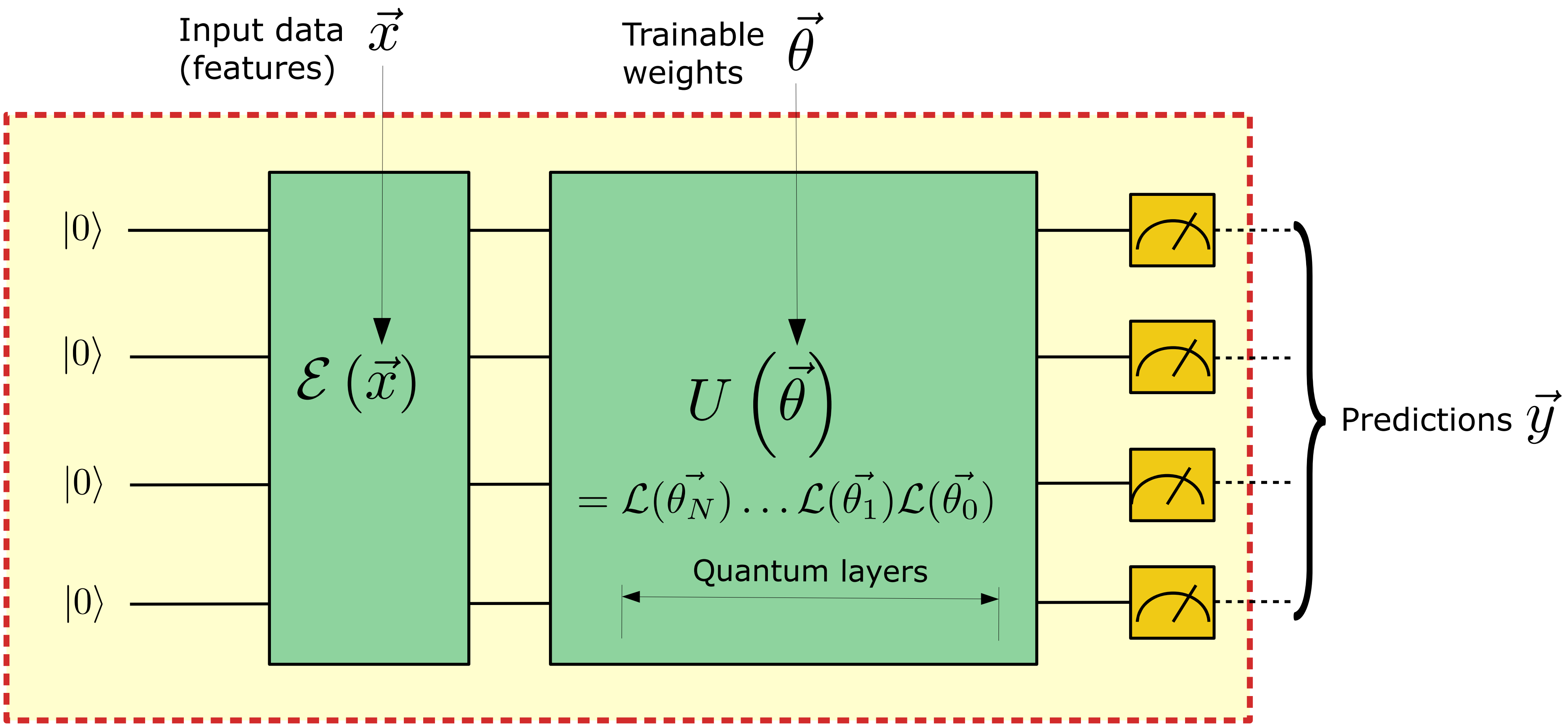
Variational quantum circuits



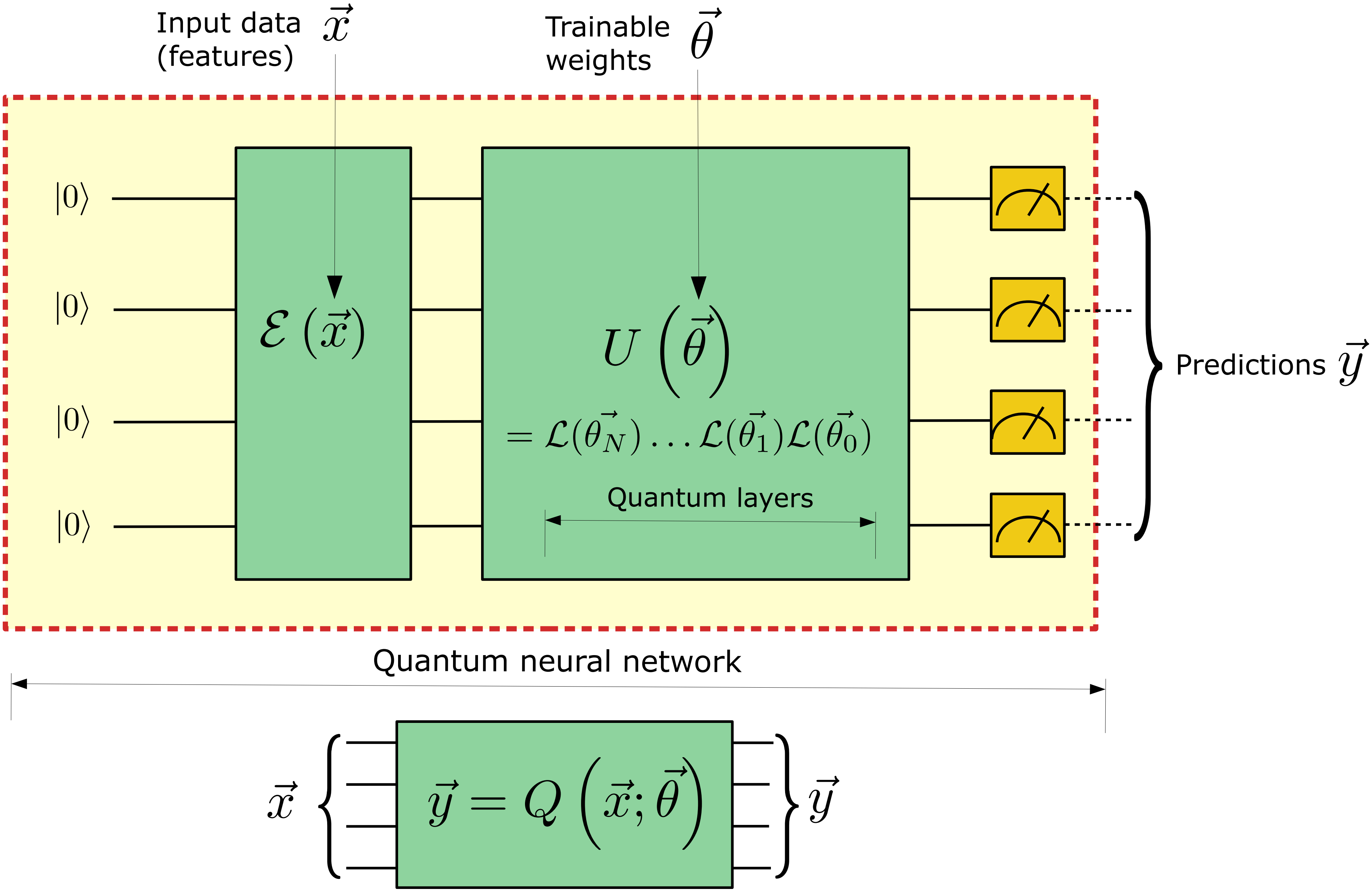
Variational quantum circuits



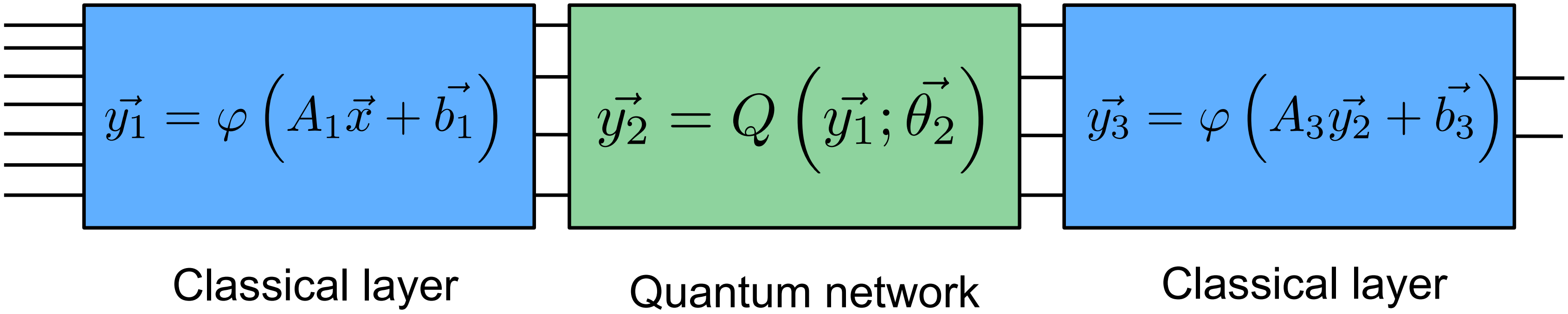
Variational quantum neural networks



Variational quantum neural networks



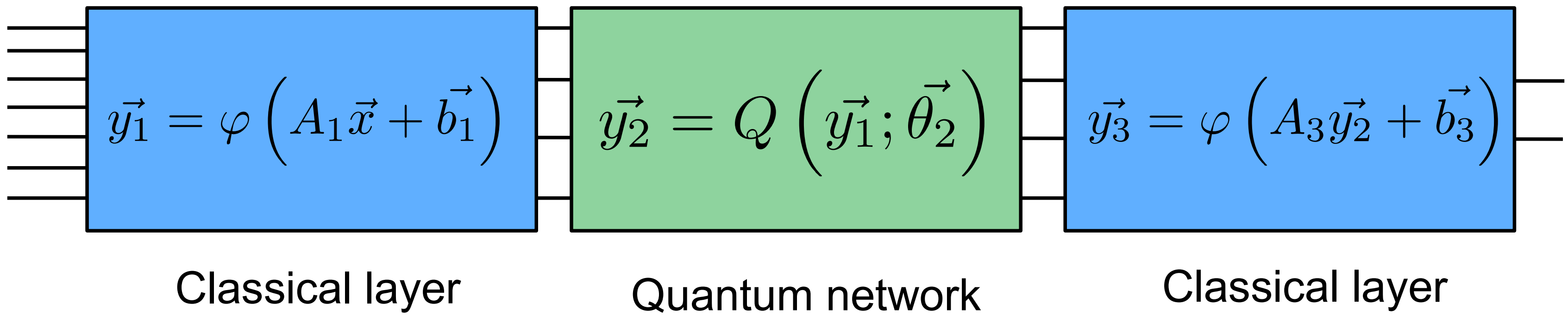
"Dressing" a quantum circuit with two classical layers



Two advantages: 1) Complete flexibility in the number of inputs and outputs

2) Classical layers can "learn" how to "use" the quantum circuit

“Dressing” a quantum circuit with two classical layers

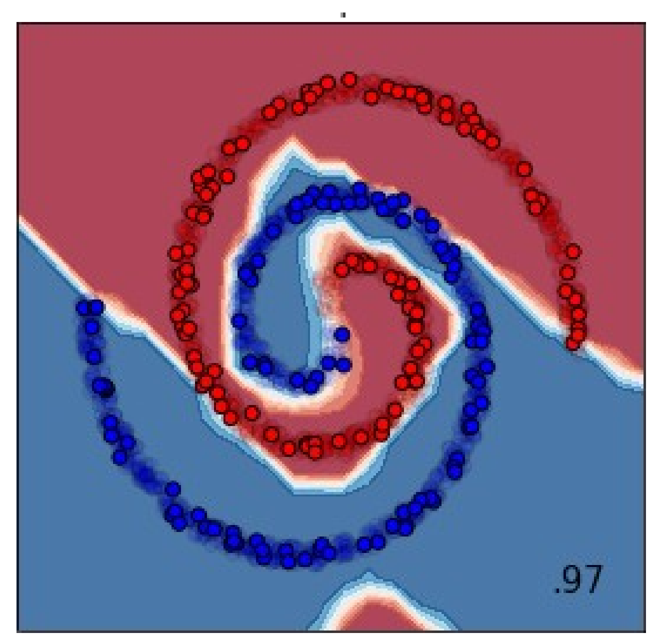


Two advantages: 1) Complete flexibility in the number of inputs and outputs

2) Classical layers can “learn” how to “use” the quantum circuit

Example: classification a benchmark dataset

- 2000 training points
- 200 test points
- 1000 iterations (Adam optimizer)
- Cross entropy loss function



Outline

1. Machine learning with variational quantum circuits

2. Quantum transfer learning

3. Classification of high-resolution images with QPUs

Transfer learning

"i.e., using some previous knowledge for learning something new"

Daily-life examples:

- Learning a second language (knowing the first)
- Learning how to write (knowing how to read)
- Learning to play football (knowing how to run)

The same idea has been applied to **classical neural networks!**

Transfer learning

"i.e., using some previous knowledge for learning something new"

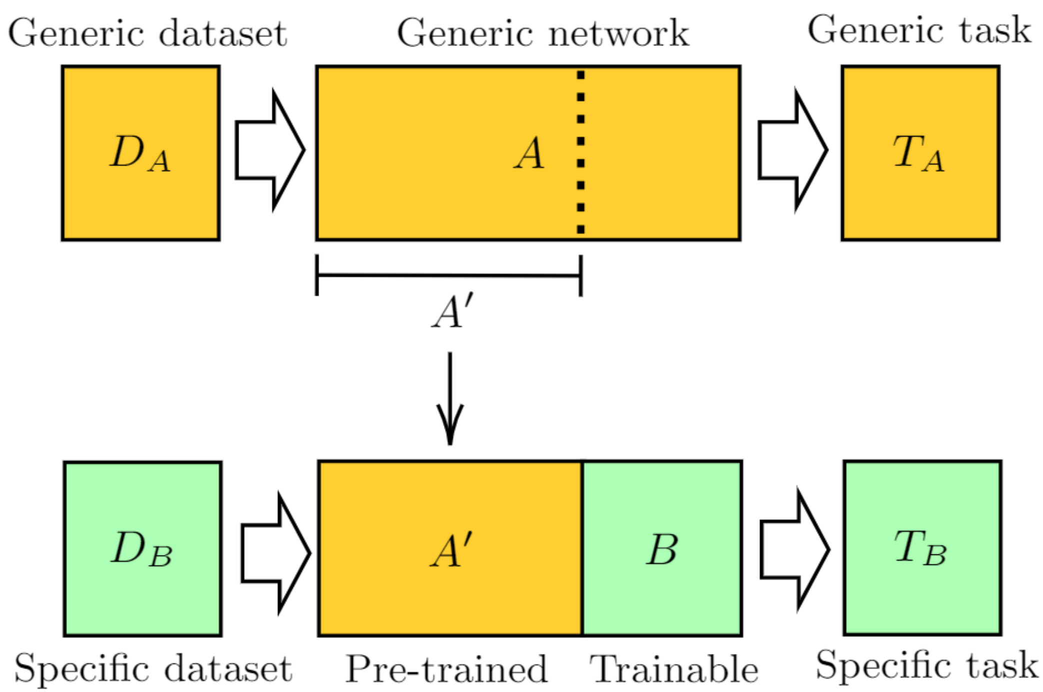
Daily-life examples:

- Learning a second language (knowing the first)
- Learning how to write (knowing how to read)
- Learning to play football (knowing how to run)

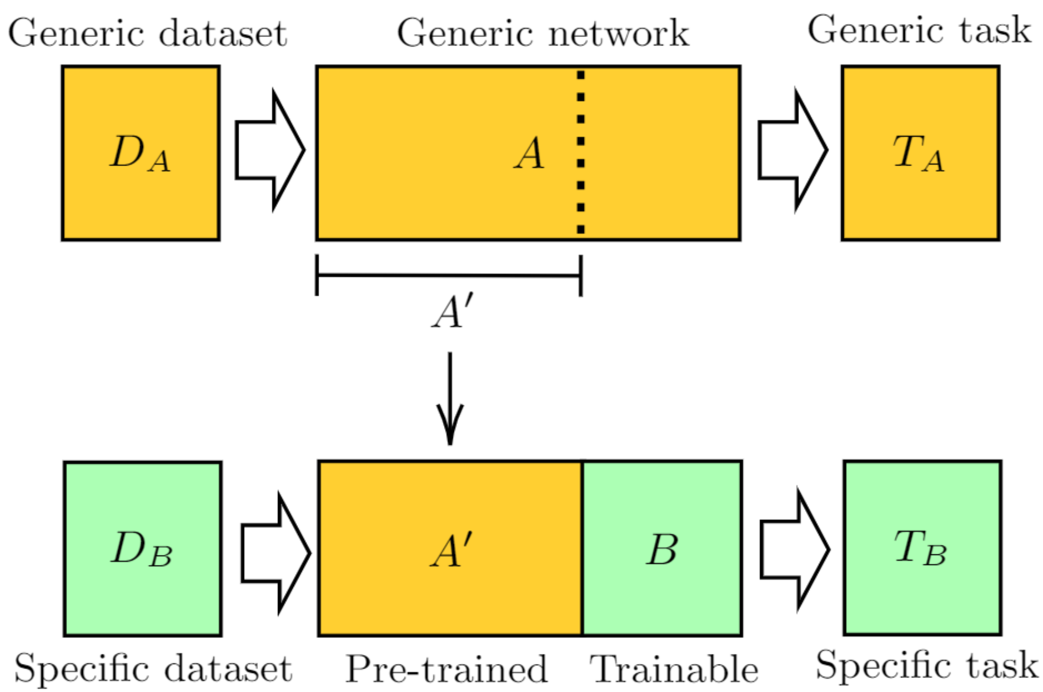
The same idea has been applied to **classical neural networks!**

- What about **quantum variational networks?**
- Is it possible to transfer some ***knowledge*** at the classical-quantum interface?

Quantum (and classical) transfer learning

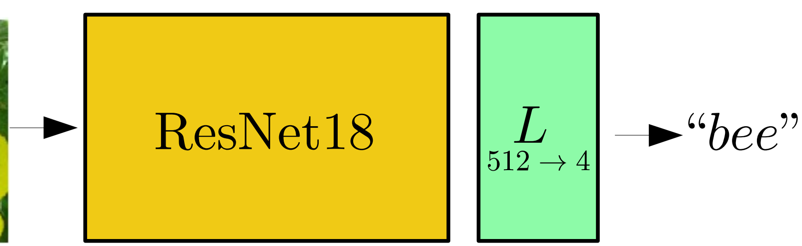


Quantum (and classical) transfer learning



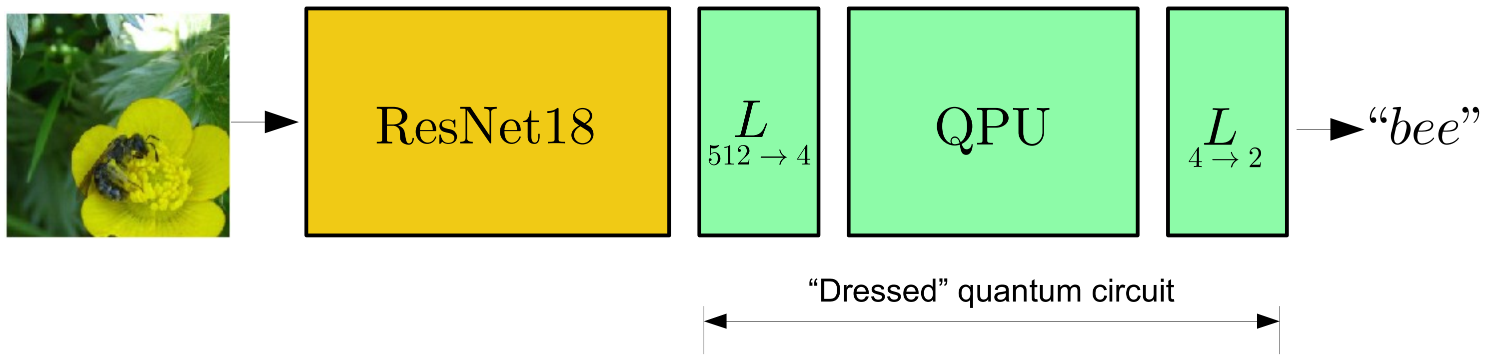
• Example with classical networks: “ants” vs “bees” (PyTorch tutorial)

~300 images of *ants* and *bees*



Pre-trained on ImageNet (1.2 million images of 1000 classes)

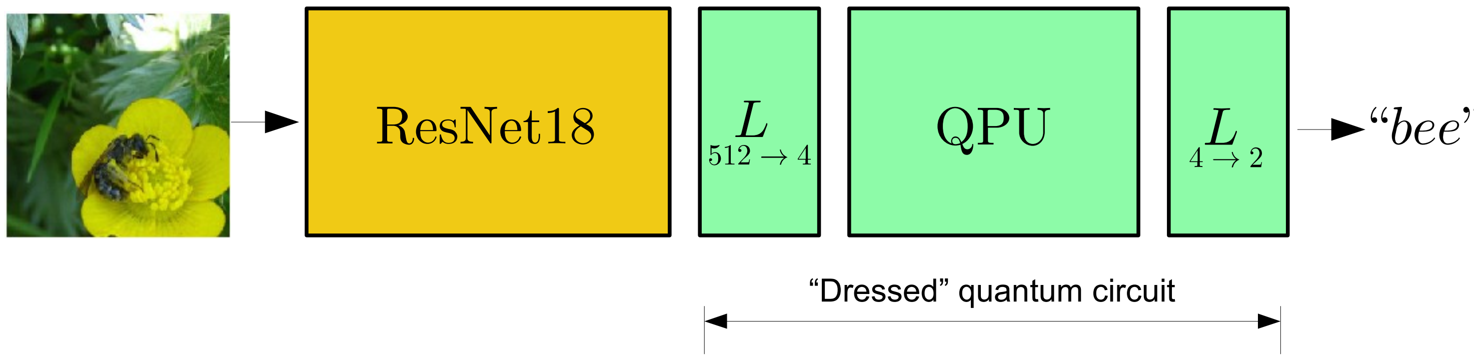
Quantum-to-classical transfer learning



Ok but, in practice, how can we train a hybrid classical-quantum network?

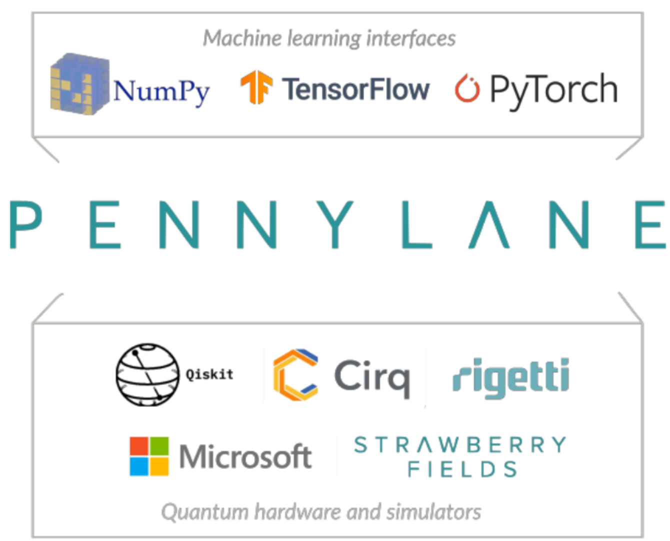
Is there any "quantum analog" of *TensorFlow* or *PyTorch*?

Quantum-to-classical transfer learning



Ok but, in practice, how can we train a hybrid classical-quantum network?

Is there any "quantum analog" of *TensorFlow* or *PyTorch*? **Yes!**



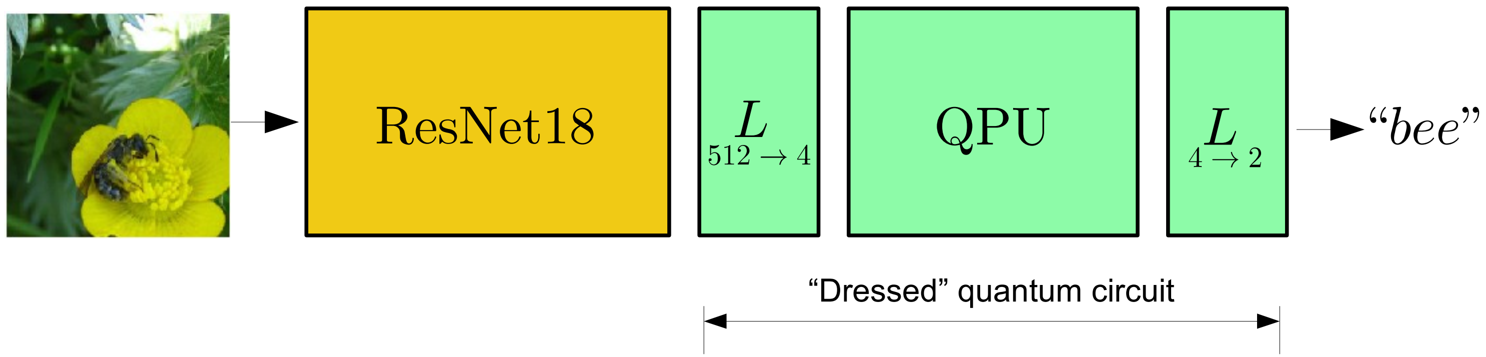
- Automatic (quantum) differentiation
- Device independent
- Open source
  github.com/XanaduAI/pennylane

Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Carsten Blank, Keri McKiernan, and Nathan Killoran. PennyLane: Automatic differentiation of hybrid quantum-classical computations. 2018. **arXiv:1811.04968**

Outline

1. Machine learning with variational quantum circuits
2. Quantum transfer learning
3. Classification of high-resolution images with QPUs

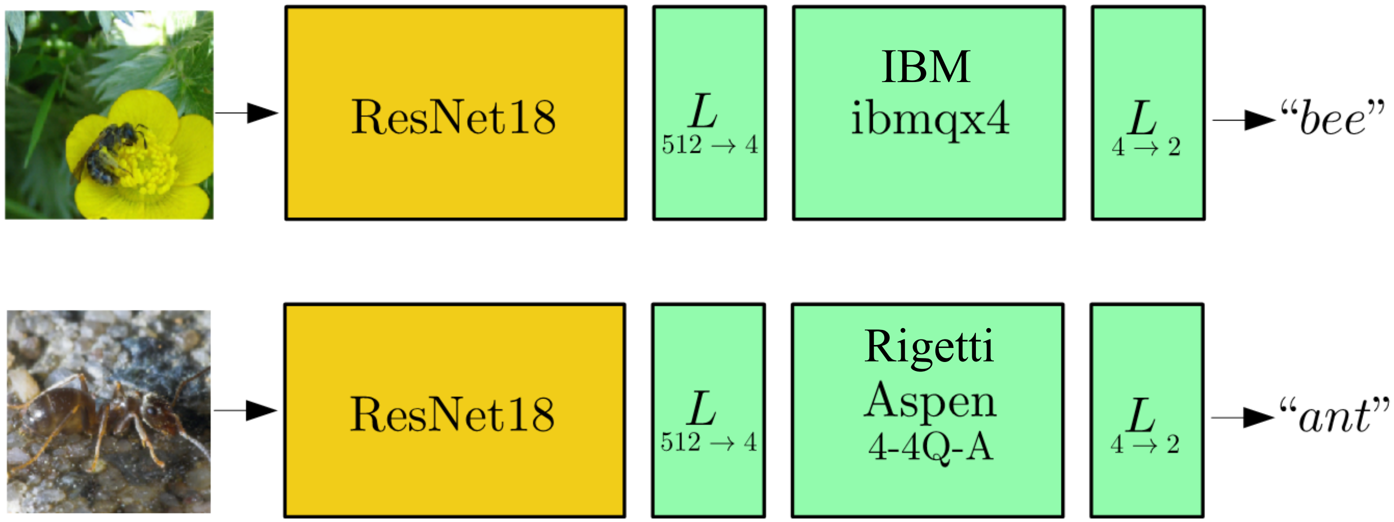
Quantum-to-classical transfer learning



Results:

	ants/bees	**dogs/cats**	**planes/cars**
Quantum depth	6	5	4
Number of epochs	30	3	3
Batch size	4	8	8
Learning rate	0.0004	0.001	0.0007
Accuracy	0.976	0.8270	0.9605

Experiments with real quantum processors

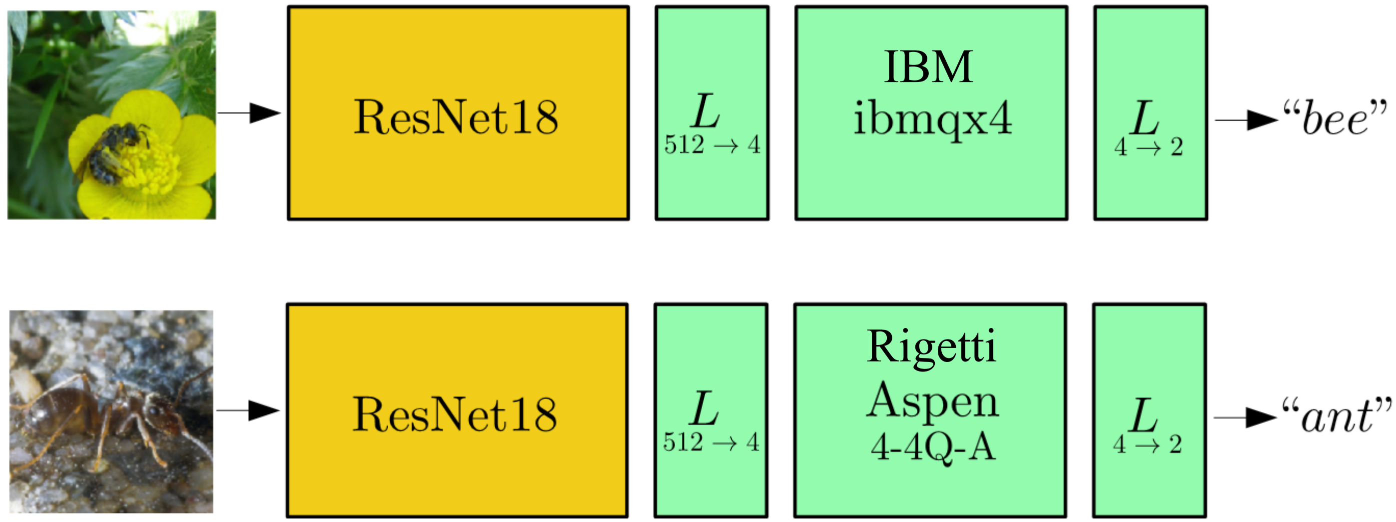


In PennyLane you can do it with a **single line of code!**

IBM: `dev = qml.device("qiskit.ibmq", wires=n_qubits, backend="ibmqx4", ibmqx_token=token)`

Rigetti: `dev = qml.device("forest.qpu", device="Aspen-4-4Q-A", shots=1024)`

Experiments with real quantum processors



In PennyLane you can do it with a **single line of code!**

IBM:
```
dev = qml.device("qiskit.ibmq", wires=n_qubits, backend="ibmqx4", ibmqx_token=token)
```

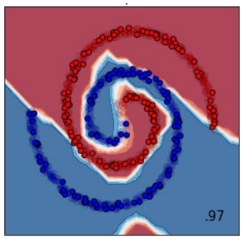
Rigetti:
```
dev = qml.device("forest.qpu", device="Aspen-4-4Q-A", shots=1024)
```

Results:

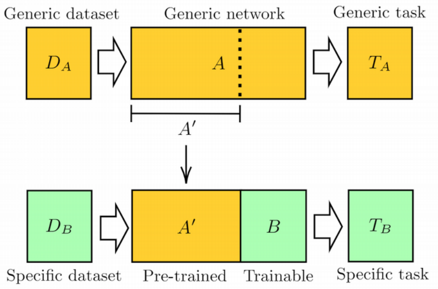
QPU	Accuracy
Simulator	0.967
ibmqx4	0.95
Aspen-4-4Q-A	0.80

Conclusions

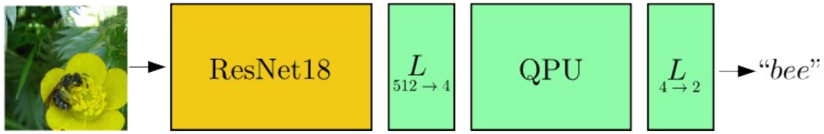
- Quantum circuits "dressed" by classical layers



- Theoretical framework for *quantum transfer learning*



- Experimental classification of high-resolution images

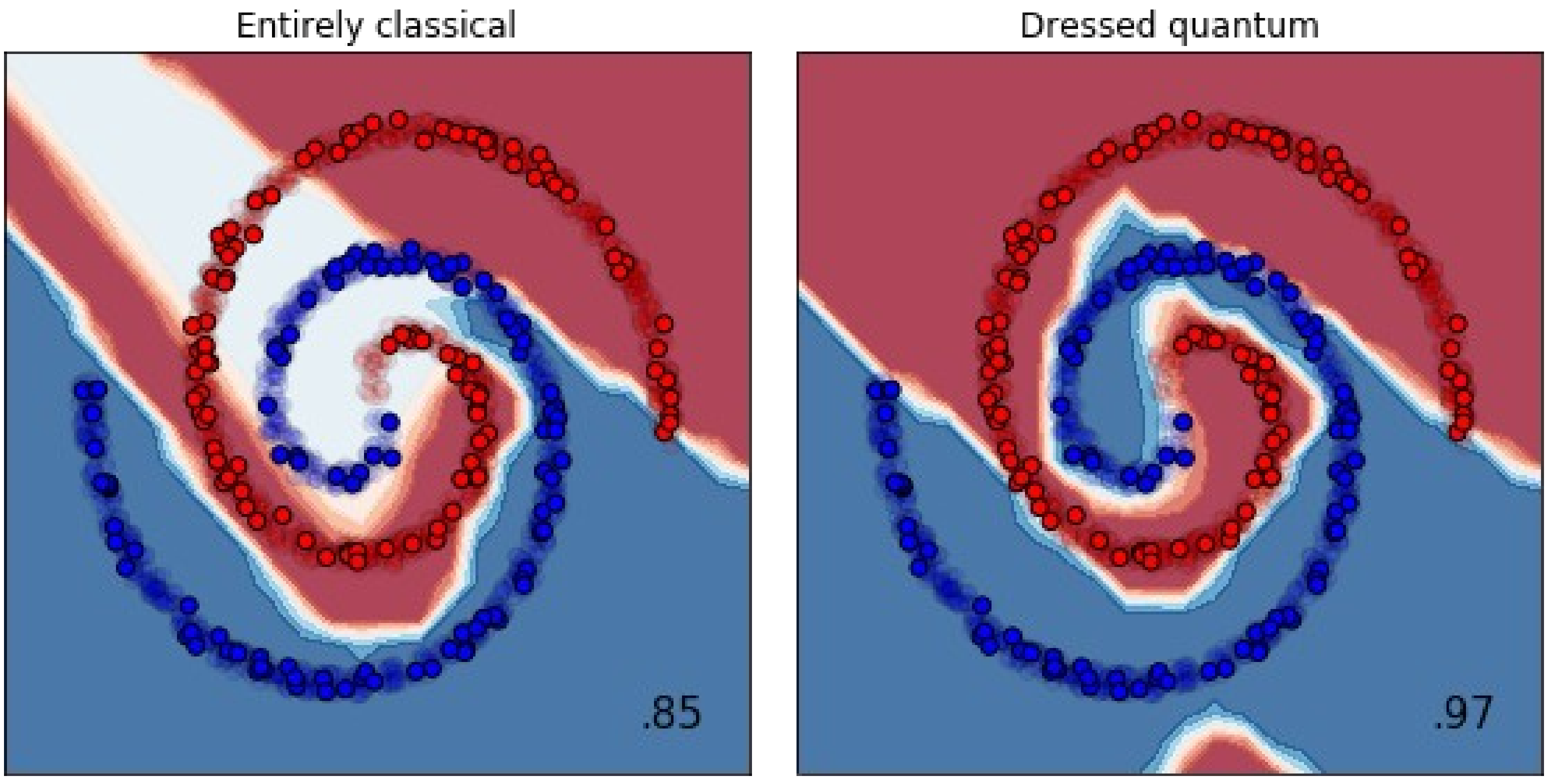


For much more details like, e.g.,
quantum-to-classical or quantum-to-quantum transfer learning ...

Transfer learning in hybrid classical-quantum neural networks.
Andrea Mari, Thomas R. Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran.
arXiv:1912.XXXX, (2019).

Code available at:
github.com/XanaduAI/quantum-transfer-learning

Supplementary material



$$C = L_{4\to2} \circ L_{4\to4} \circ L_{2\to4} \qquad \tilde{Q} = L_{4\to2} \circ Q \circ L_{2\to4}$$