STRUTTURA PROPOSTA ESERCIZIO SU QML

S. Giagu

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

INFN ROMA

SAPIENZA UNIVERSITÀ DI ROMA

AI-INFN Tech. Meeting 11.11.2024

STRUTTURA ESERCIZIO

- **Lezione “teorica” 1h30’:** Introduzione al QML e piattaforme di sviluppo Pennylane e Qiskit
- **Hands-on 3h:**
  - classificazione con PQC (QNN) in Pennylane w/ pytorch backend
  - anomaly detection con PQC (QAE) in Pennylane w/ pytorch backend
  - problema QUBO (max cut/graph coloring) con PQX in Qiskit

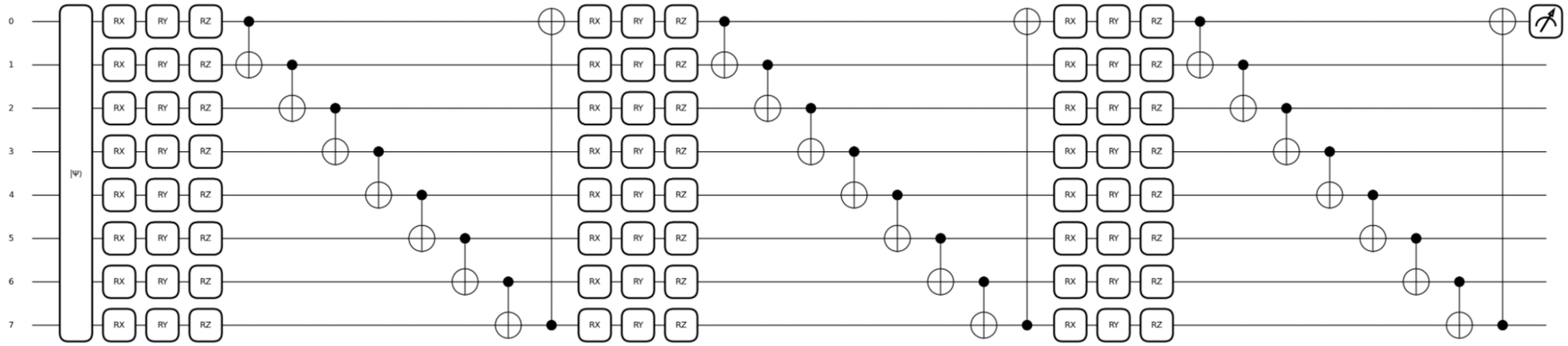
Introduzione teorica al QML

- **slide in preparazione (quasi pronte), contenuto:**
  - Quantum Computing e Quantum Computers
  - Interesse in Fisica delle particelle
  - Modello di computazione basato su circuiti quantistici (elementi)
  - qubit, operazioni su qubit, quantum gate, interferenza e entanglement
  - misura quantistica
  - Algoritmi quantistici e QML
  - QML con Circuiti Variazionali: QNN, Quantum Kernels
  - Potere espressivo di un PQC
  - Addestrabilità e limitazioni del QML
  - Esempi di applicazioni di QML
  - Sviluppo in pratica di un modello di QML: piattaforme software, esempio Pennylane e esempio Qiskit

Esercizio 1: QNN per un task di classificazione

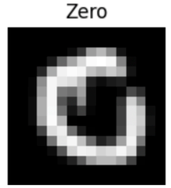
- **scopo:** capire come si programma un semplice algoritmo di QML basato su QNN su un task molto semplice, classificazione binaria di caratteri MNIST
- **Task:**
  I. encoding di dati classici
  II. programmazione di un circuito QNN multilayer
  III. addestramento e uso in predizione del circuito allenato
- **implementazione dell'esercizio:**
  - **hands-on insieme agli studenti:** implementando il codice necessario end-to-end
  - **hackathon successivo:** in cui gli studenti devono modificare il codice per implementare un task di classificazione muti-classe

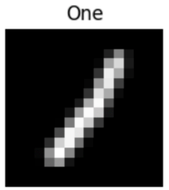
QClassifier_binary: https://drive.google.com/file/d/1MUtieOhtDQA-qm_itOR34u4vJWEl3Z4z/view?usp=sharing
QClassifier_multiclass: https://drive.google.com/file/d/1_cWXFMHlGIrWNQzJw-VuSUgLfxTSlS02/view?usp=sharing

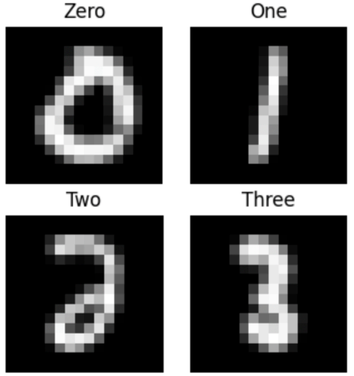


$p(|0\rangle)$

binary class.

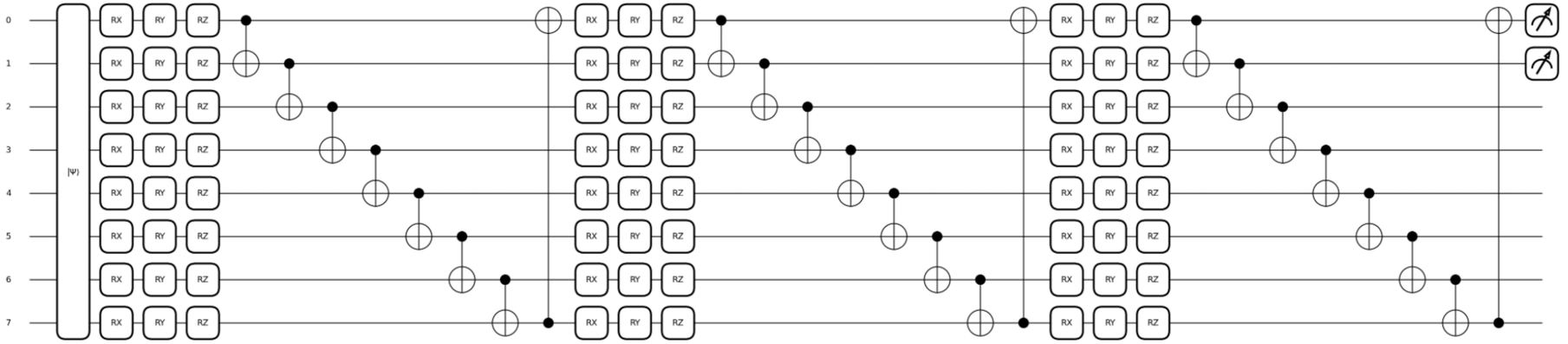


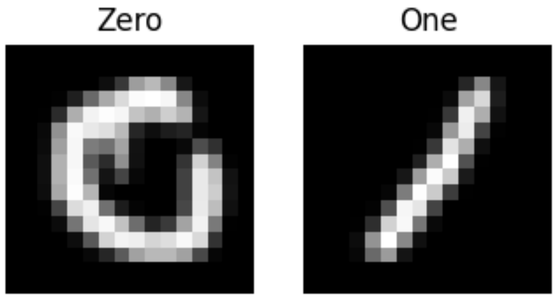


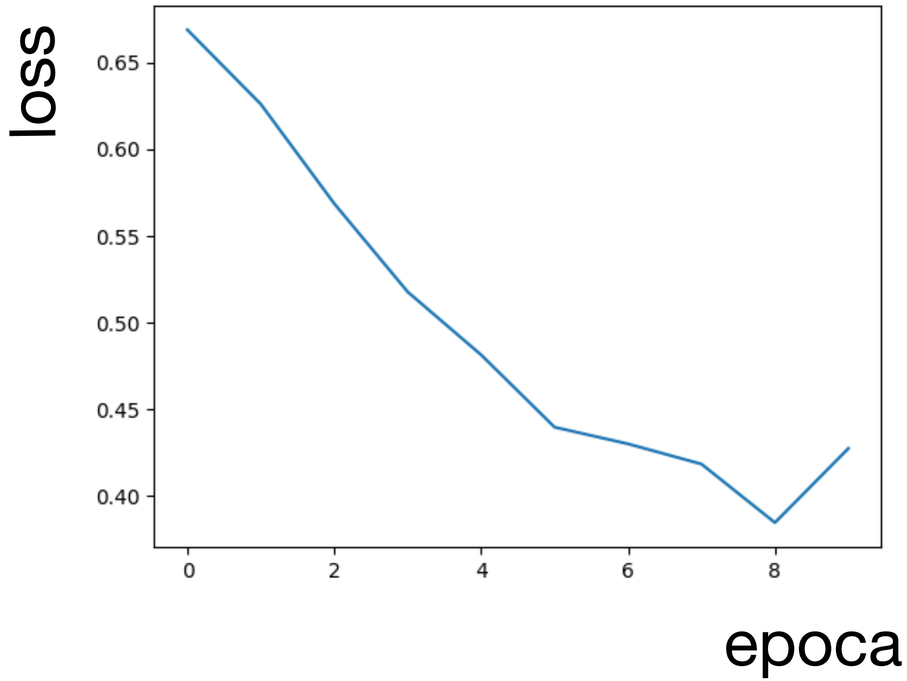


multi-class

$p(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$







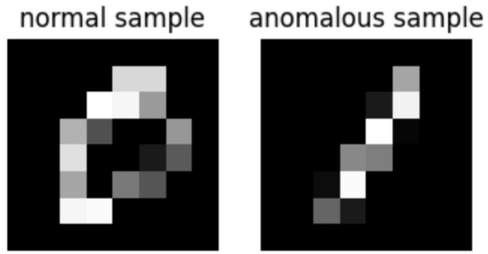

```
# test performance

run_test(BATCH_SIZE, dataloader_test, loss_fn)
```

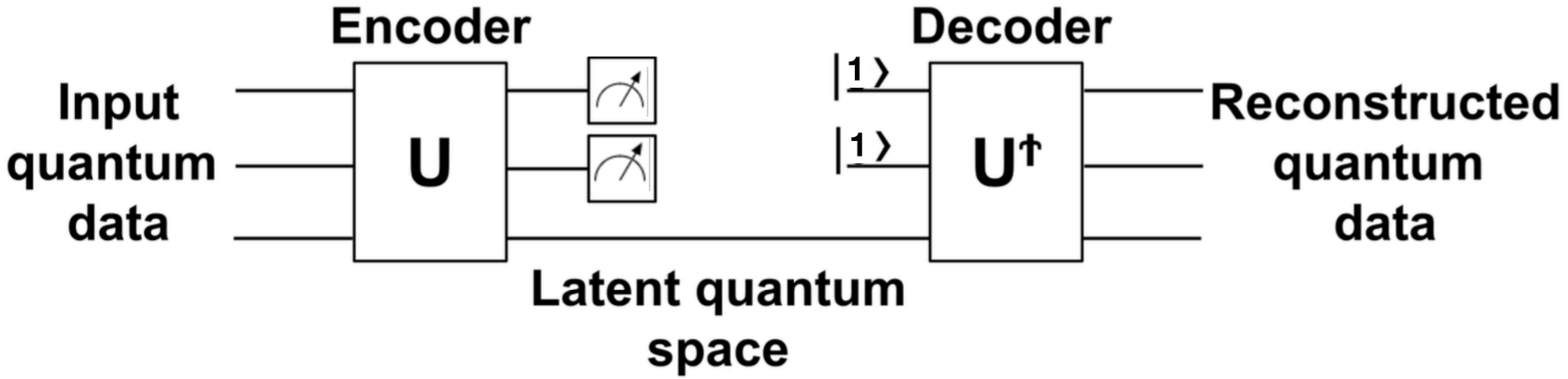
```
Test loss:  0.3952110265381634
Test accuracy:  0.96923828125
```

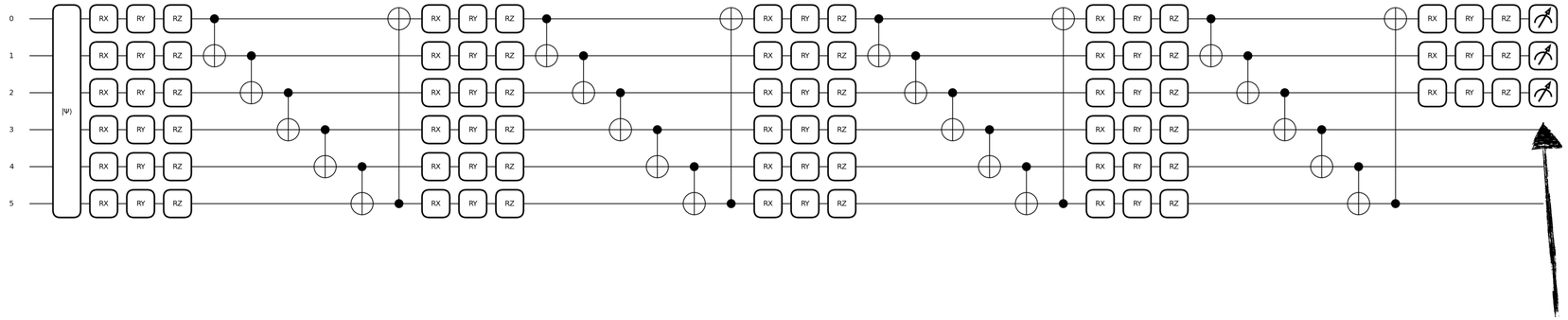
Esercizio 2: QAE per un task di AD

- **scopo:** capire come si possono sfruttare le caratteristiche di un PQC per implementare un Auto-Encoder quantistico, e allenarlo a riconoscere caratteri MNIST “anomali” (“uni”) rispetto a quelli considerati normali (“zeri”)
- ispirato a S.Bordoni, D.Stanev, T.Santantonio, S.Giagu, “Long-Lived Particles Anomaly Detection with Parametrized Quantum Circuits”, Particles 2023, 6(1), 297-311; https://doi.org/10.3390/particles6010016
- **Task:**
  I. programmazione di un QAE
  II. addestramento e uso in predizione del circuito allenato
- **implementazione dell’esercizio:**
  - **hands-on breve insieme agli studenti:** spiegazione dell’idea dietro all’implementazione di un QAE
  - **hackathon:** in cui gli studenti devono implementare parti del codice completo per implementare il circuito AE, allenare il modello, e testarne le prestazioni

QAE: https://drive.google.com/file/d/16VtyoJwq1HXR73SB0j_mFb5rM0vw5zOl/view?usp=sharing





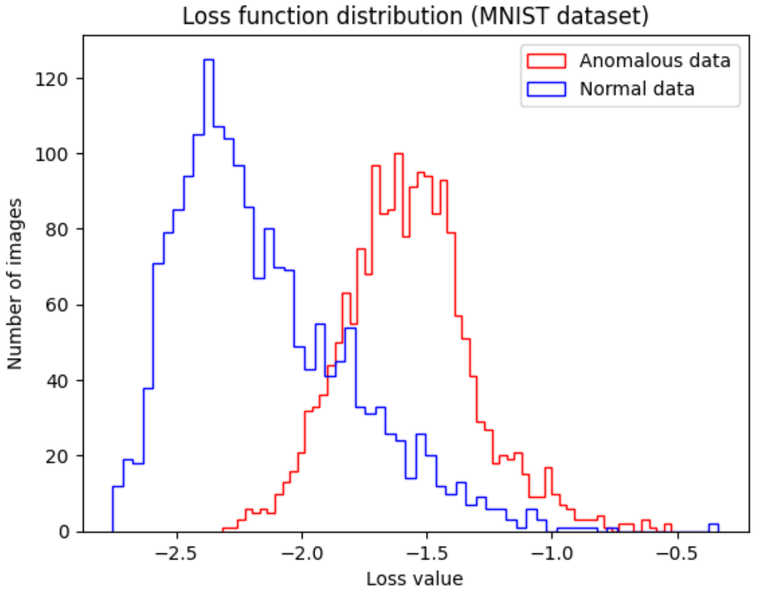


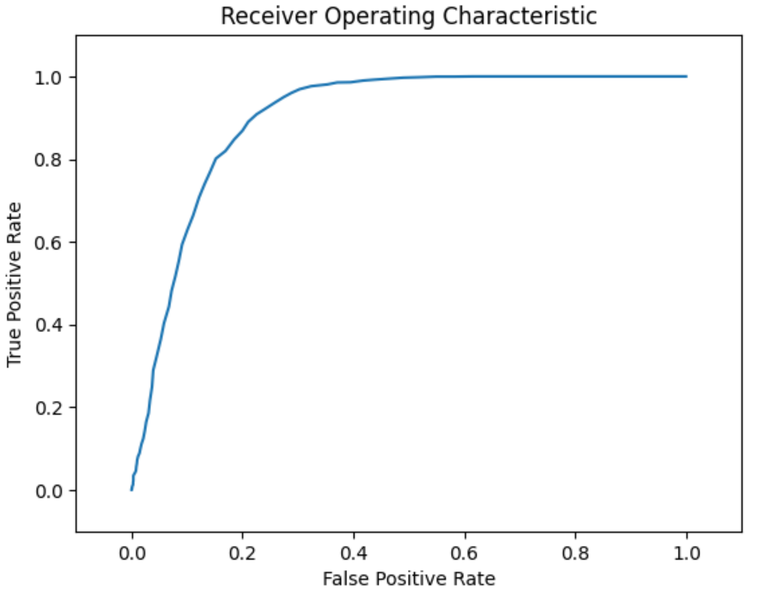
basta allenare solo l'Encoder
Decoder: aggiunto (trasposto coniugato) dell'encoder

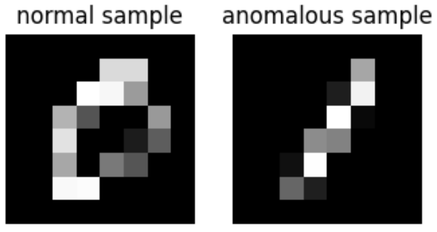
forzati ad essere $|1\rangle$

loss: valore atteso dell'operatore Z di Pauli $1 : |0\rangle \quad 0 \quad -1|1\rangle$

minimizzare il valore atteso corrisponde a forzare a $|1\rangle$ il qubit


Loss function distribution (MNIST dataset)
Anomalous data
Normal data
Number of images
Loss value
0
20
40
60
80
100
120
−2.5
−2.0
−1.5
−1.0
−0.5


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


normal sample

anomalous sample