Title: Machine Learning (2021/2022)

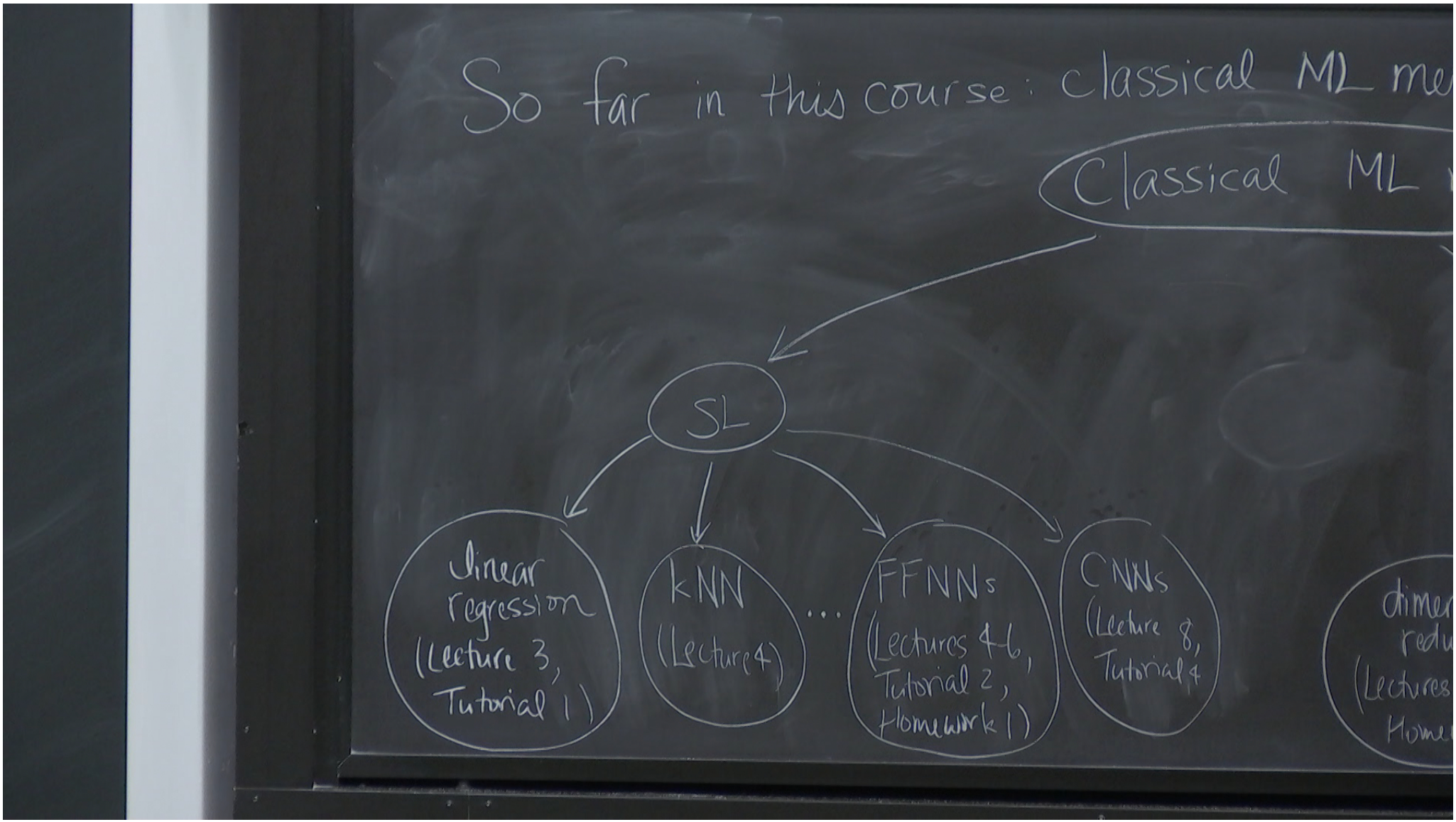
Speakers: Lauren Hayward

Collection: Machine Learning (2021/2022)

Date: May 02, 2022 - 11:30 AM

URL: https://pirsa.org/22050009

Abstract: This course is designed to introduce modern machine learning techniques for studying classical and quantum many-body problems encountered in condensed matter, quantum information, and related fields of physics. Lectures will focus on introducing machine learning algorithms and discussing how they can be applied to solve problem in statistical physics. Tutorials and homework assignments will concentrate on developing programming skills to study the problems presented in lecture.


So far in this course: classical ML me
Classical ML
SL
linear regression (Lecture 3, Tutorial 1)
kNN (Lecture 4)
...
FFNNs (Lectures 4-6, Tutorial 2, Homework 1)
CNNs (Lecture 8, Tutorial 4
dimen redu (Lectures Home

ical ML methods to gain insight about many-body systems

ssical ML methods

RL (not covered in this course)

UL

dimensional reduction (Lectures 9-10, Homework 2)

generative modelling (Lectures 11-12)

stems

Today: work at the intersection of "quantum" and ML

Quantum and ML

ML with data from quantum systems

ML with quantum processing devices ("quantum machine learning")

ML with data from quantum systems (arXiv:2101.11099)

We can use classical ML methods to study quantum many-body systems

Examples:

1. Quantum phase transitions (QPTs)

So far: PTs in classical systems (Ising model, $\mathbb{Z}_2$ classical gauge theory)

We can use similar methods to study QPTs (data: projective measurements from simulation or experiment)

② Quantum state tomography

Goal: given a (limited) set of measurement data, reconstruct the corresponding (unknown) quantum state

Recently, generative modelling with NNs has been used

(2) Quantum state tomography

Goal: given a (limited) set of measurement data, reconstruct the corresponding (unknown) quantum state

Recently, generative modelling with NNs has been used

③ Finding approximate ground states

Given the Hamiltonian of a quantum many-body system, can train a generative model to approximate the ground state (as discussed by Mohamed)

simulation)
ent

Quantum Machine Learning (QML)

(a.k.a. "quantum-assisted ML", "quantum-enhanced" ML)

References:

- Schuld & Petruccione, "ML with Quantum Computers" book (2021)
- Schuld & Carrasquilla, tutorial at NeurIPS (2021)

Goal: Use quantum computers/devices to help with (speed up or innovate) ML tasks

Here, we will focus on the goal of using quantum computers/devices (including "near-term" devices that are already available) as ML models known as "variational circuits"

Variational circuits

↳ a class of ML models consisting of quantum circuits that depend on a set of tunable parameters $\Theta = \{\theta_1, \theta_2, \ldots\}$

tum

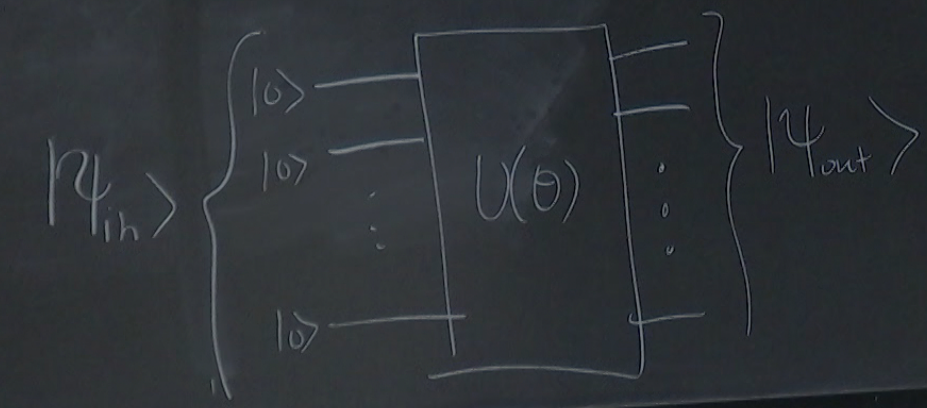
t are

iational

Consider applying a unitary $U(\theta)$ to an input state $|\psi_{in}\rangle \equiv |00$ of n qubits

$$|\psi_{out}\rangle = U(\theta)|\psi_{in}\rangle$$

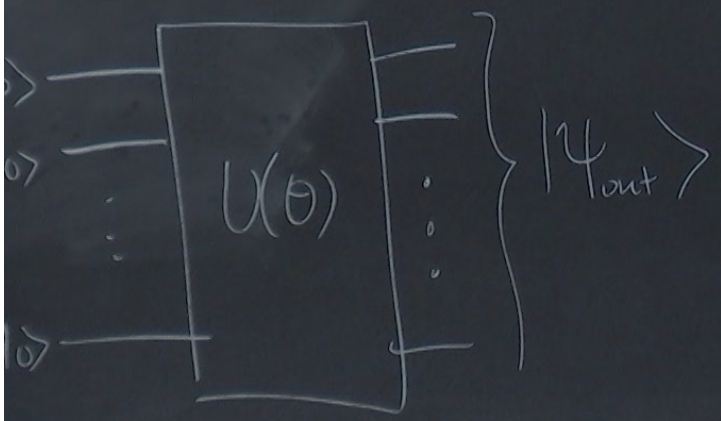
In circuit notation:



Consider applying a unitary $U(\theta)$ to an input state $|\psi_{in}\rangle \equiv |00\cdots0\rangle$ of n qubits

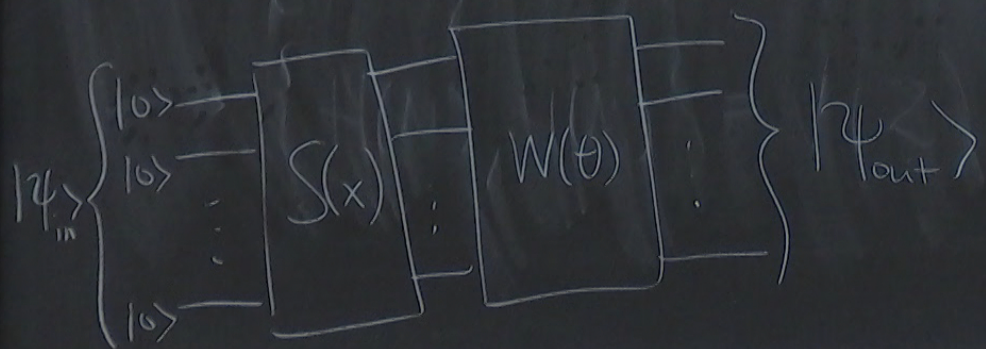
$$|\psi_{out}\rangle = U(\theta)\,|\psi_{in}\rangle$$

In circuit notation:



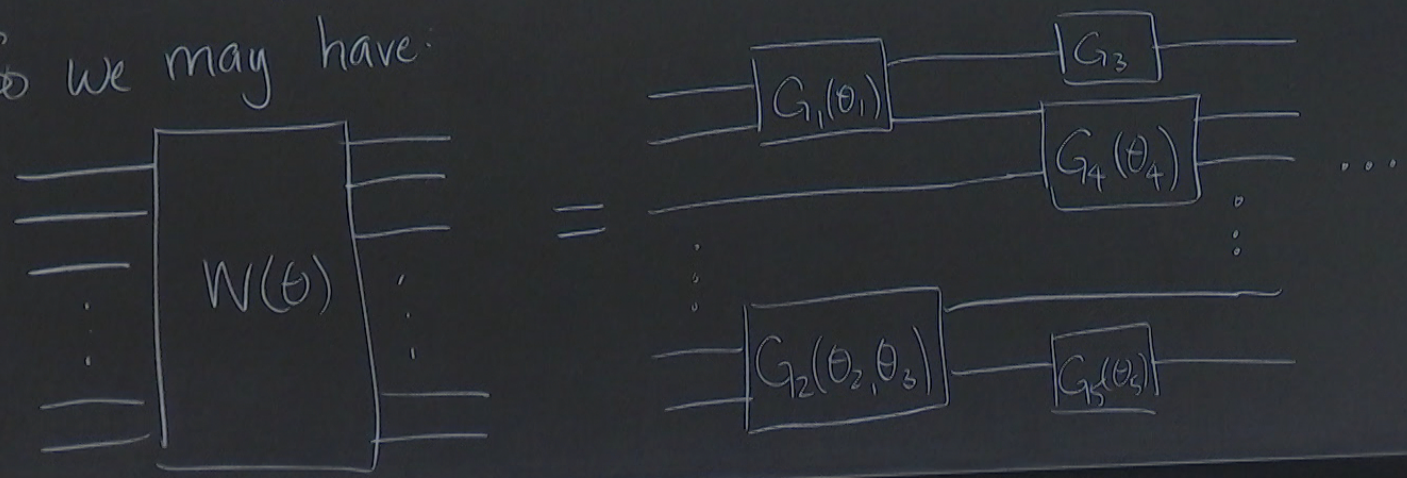
Often, we first apply a series of gates that depend on non-tunable parameters $x = \{x_1, x_2, \ldots\}$ to encode the data and get the desired input to process

$$|\psi_{out}\rangle = U(x,\theta)\,|\psi_{in}\rangle, \text{ with } U(x,\theta) \equiv \underbrace{W(\theta)}_{\text{processing}}\,\underbrace{S(x)}_{\text{encoding}}$$



Practically, it is more common to implement $S(x)$ and $W(\theta)$ as a series of gates that each act on a small # of qubits (usually one or two)

So we may have:



and $W(\theta)$ as a

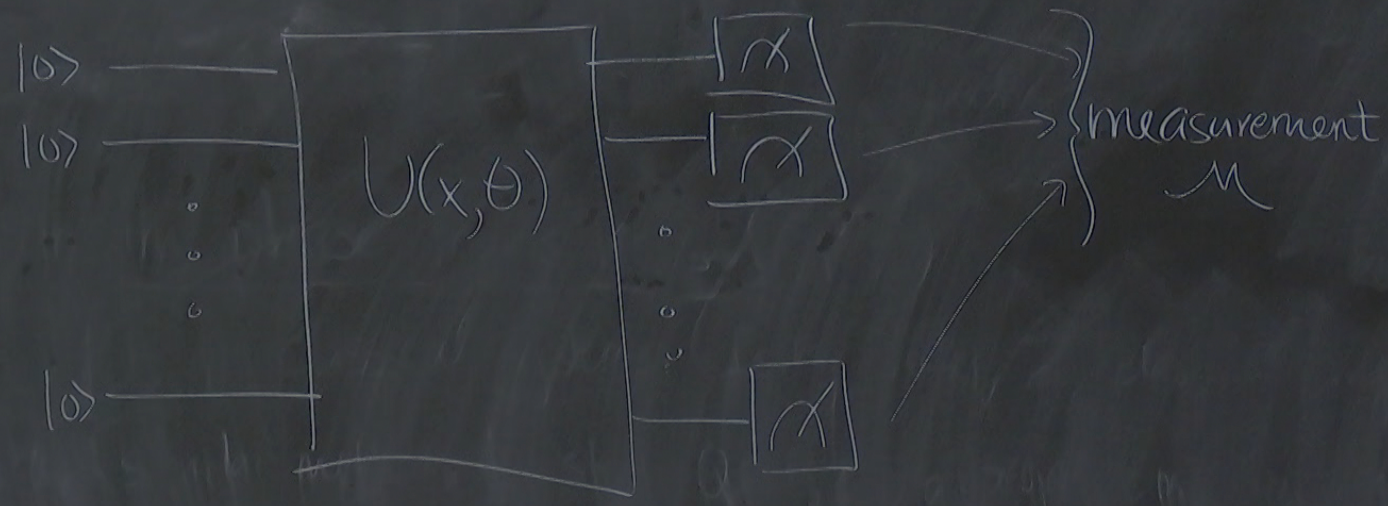
s (usually one or two)

(and similarly for $S(x)$)

...

Cost functions for variational circuits

The cost func. that we use in our learning algorithm will depend on the results of a measurement $\mathcal{M}$:



For a deterministic SL problem, we can average the meas. results to get

$$f(x, \theta) = \langle 0 | U^{\dagger} M U | 0 \rangle$$

function we would fit to our labels

If we <u>don't</u> average, the result will correspond to a probability
be used within probabilistic SL or UL problems

stic SL problem, we can average the meas.

$$\theta) = \langle 0 | U^{\dagger} M U | 0 \rangle$$

we would fit to our labels

erage, the result will correspond to a probability dist. that can
robabilistic SL or UL problems

function we would fit to our labels

If we <u>don't</u> average, the result will correspond to a probability

be used within <u>probabilistic SL</u> or UL problems

(dataset $\mathcal{D}$ could contain (x, y_1) and (x, y_2))

Training algorithms

Training corresponds to finding the params

$$\underbrace{\Theta = \{\theta_1, \theta_2, \ldots\}}_{\text{analogous to finding the weights and biases of a NN}}$$ that minimize the cost

Search for Θ using an algorithm like gradient descent

(dataset $\mathcal{D}$ could contain (x, y_1) and (x, y_2))

To compute gradients of the cost:

- use finite diff. methods or "parameter shift rules"
- be aware of "barren plateaus": regions in param. space where all gradients are approximately zero

Studying and efficiently calculating these gradients is an active area of research

Programming QML models

Several software libraries are available:

- PennyLane → See Tutorial 6
- TensorFlow Quantum
- Yao

We use these libraries to run code on quantum simulators or on quantum hardware devices