Rutgers University

Course Syllabus for MATH 495- Section 02

Tensor Networks as a bridge between Neural Networks and Quantum Physics

Fall 2025

General Information

Instructor	Mariano Echeverria	Meeting:	HILL-005 TTh 4 2:00 pm-3:20 pm
Email	mariano.echeverria@rutgers.edu		

Important Information

Prerequisites

Linear Algebra (Math 250) is the only requisite for this course, **but** the website will list 244 and 250 as prerequisites. In such a case, fill out the Prerequisite Override Form for assistance with registration.

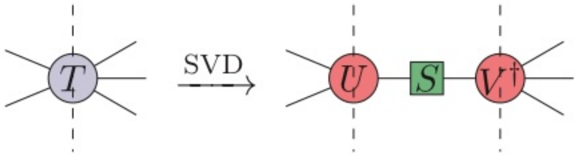
https://rutgers.ca1.qualtrics.com/jfe/form/SV_9H9FGkA49Z7BpBk

Also, since this is a topics course, being interested in these topics is equally important.

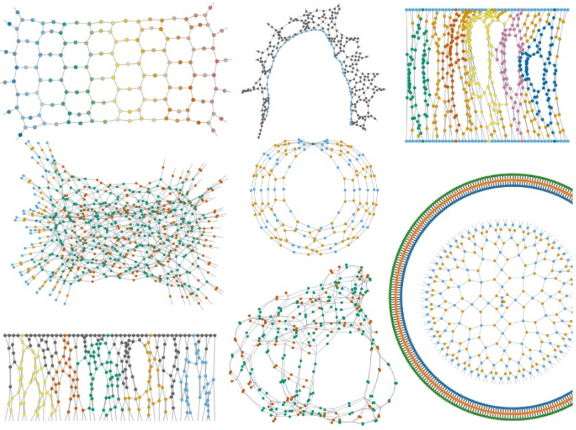
What are Tensor Networks?

Tensors are the natural generalization of vectors and matrices. Roughly, a vector $\mathbf{v}$ requires one index to specify its entries v_i, a matrix A requires two indices to specify its entries A_{ij}, and a tensor T of rank n is an object which would require n indices $i_1, i_2, \cdots, i_n$ to specify its entries $T_{i_1 i_2 \cdots i_n}$. Of course, there is a more geometric way of understanding tensors in terms of vectors spaces and the *dual* of vector spaces which we will describe in this course.

In the same way in which there are different algebraic operations between vectors or matrices, there are different algebraic operations between tensors, and **Tensor Networks** (TN) are a convenient diagrammatic representation that allows one to combine several tensors into a single tensor. For example, shown below is a TN representation of the Singular Value Decomposition (SVD) of a tensor.



Modern Deep Learning consists almost entirely of operations on or between tensors, so easily understanding tensor operations is very important for these systems. In particular, when trying to reverse-engineer the algorithms learned by a neural network in order to understand its behavior, graphical notation makes it easier to parse things (see examples below for some examples of graphical tensor notation from the QUIMB python package).



Interesting enough, Tensor Networks were developed first in Quantum Physics, especially in Many-Body Quantum Physics and Quantum Information Theory, because in these areas the systems which are being studied increase exponentially in size as a function of the number of degrees of freedom (for example, a system of N qubits uses a vector space of dimension 2^N).

In Quantum Physics, *entanglement* is the main manifestation of this exponential growth phenomenon, and in the big data world, the corresponding phenomenon is sometimes referred to as the *curse of dimensionality*, and Tensor Networks are an important tool in trying to understand these phenomena.

We will discuss how to establish bridges between these two seemingly unrelated areas (Machine Learning and Quantum Physics). As an example, one of the aims for the course will be to describe an equivalence between many-body (quantum) wave functions and mappings of convolutional and recurrent networks.

This dictionary (correspondence) will allow us import the Tensor Network technology from Quantum Physics into the Deep Learning World, and develop measures of entanglement on the Neural Networks side, which can be used as practical guidelines for choosing a specific architectures for deep convolutional networks.

Main Topics

1. Introduction to multilinear algebra: tensors and tensor operations.
2. Penrose diagrammatic notation for tensors and tensor operations.
3. Dirac's Bra-Ket notation for vectors and tensors.

4. Entanglement and measures of entanglement (entanglement entropy, geometric measure, Schmidt number).
5. Tensor Networks.
6. Convolutional and Recurrent Networks.
7. Tensor Network representation of Convolutional and Recurrent Networks.
8. Measures of Entanglement for Deep Learning.
9. Restricted Boltzmann Machines.

References and PyTorch

There won't be an official textbook for the course. As the course progresses, there will be assigned readings of some survey papers on this topic.

Since the whole point of tensor networks is that they can be represented graphically, it will be important to learn how to do this. Hence, we will learn how to use PyTorch in order to generate tensor network diagrams and work out some neural networks toy models.

Grading Criteria and Class Structure

There won't be any exams on this course. In exchange, we will have weekly assignments, and even though most meetings will consist of lectures, there will be short presentations given by the students that take place during class (especially once we reach the topic of representing tensor networks using PyTorch).

Since Tensor Networks are highly visual, a presentation may consist for example of showing tensor networks that you produced for a PyTorch assignment (this will involve some minor amount of coding).

Likewise, some of the assignment problems will be discussed during class time, and students may be asked to share part of their solutions to a problem from the assignment.