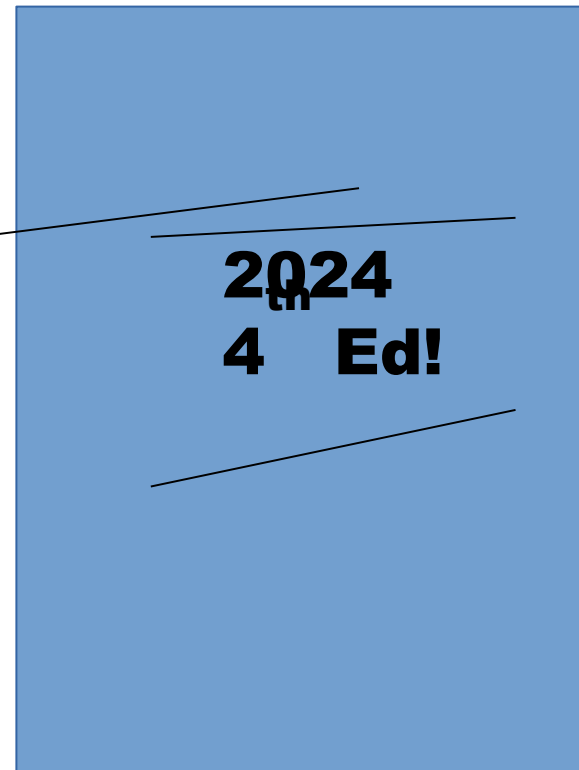
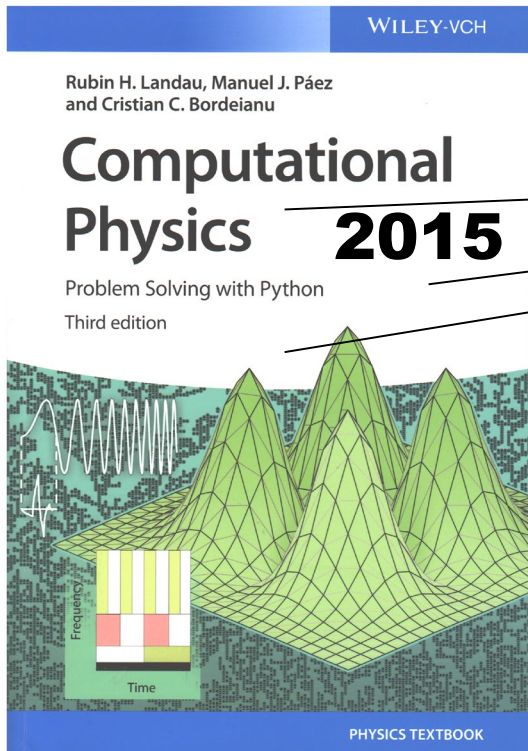


New Computational Physics to Include

Rubin H Landau

Physics Professor Emeritus, Oregon State U





Additions to New Edition

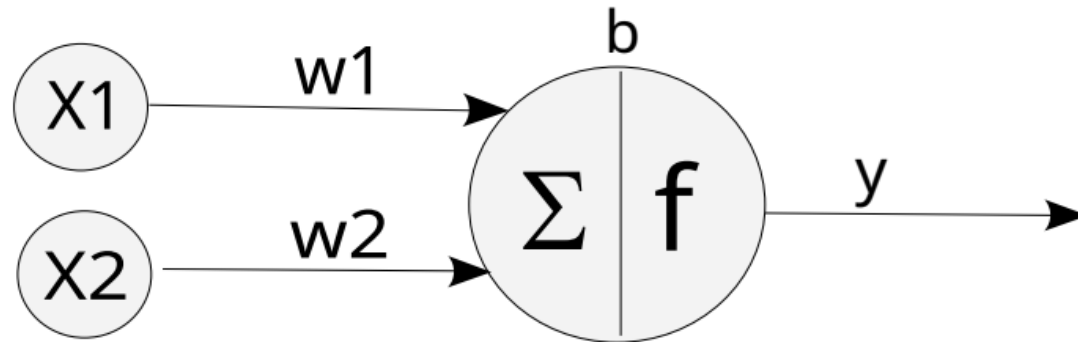
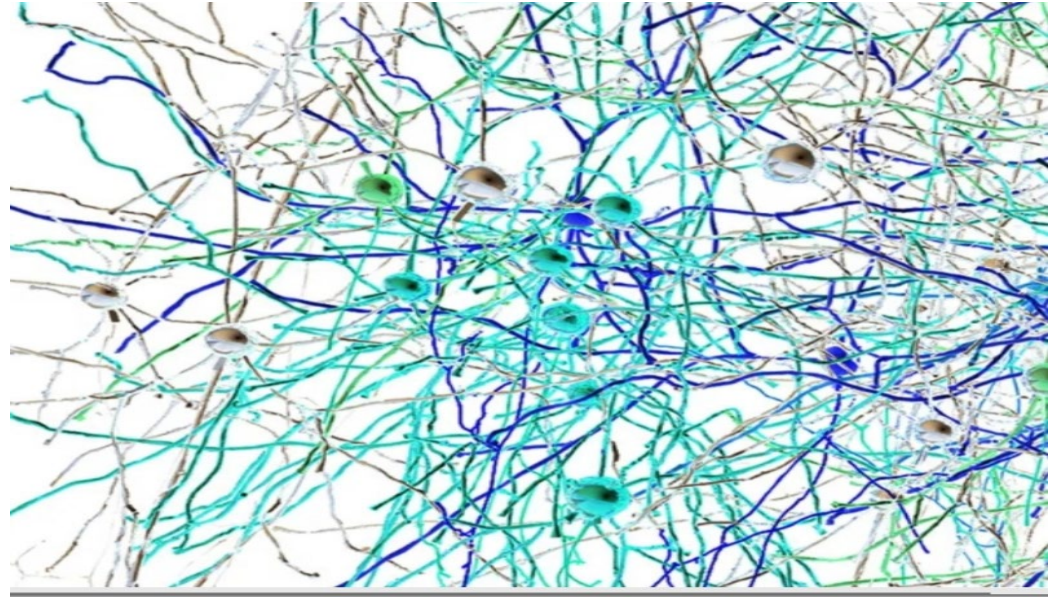
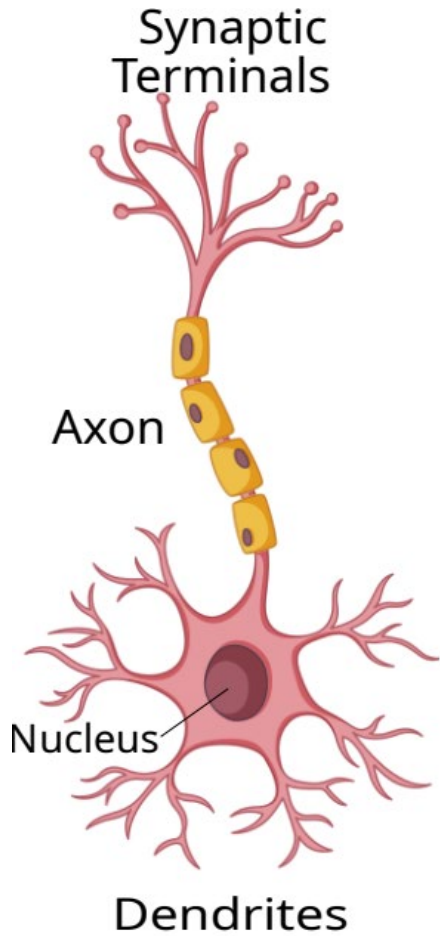
Neural Networks and Artificial Intelligence
(Data Mining)

Quantum Computing

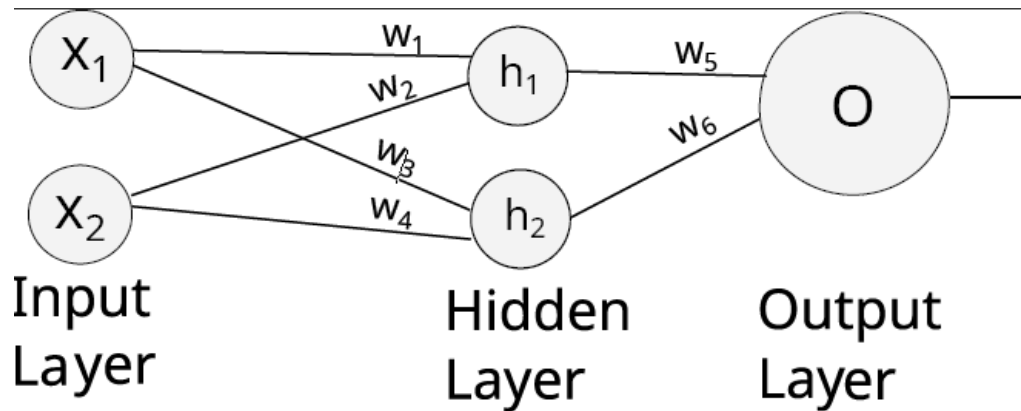
Principal Components
(Data Mining)

(General Relativity)

Neural Networks and Artificial Intelligence



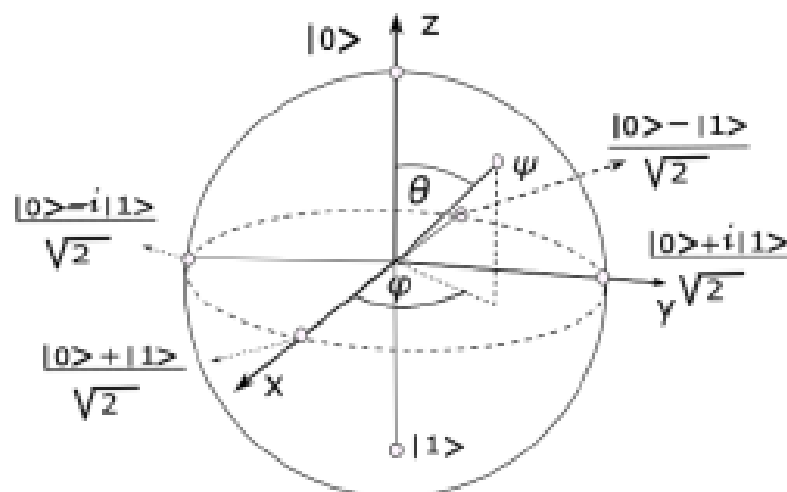
Neural Networks and Artificial Intelligence



```
# Neuron.py: An AI neuron

import numpy as np
def f(x) : return 1./ (1. + np.exp(-x))      # Activation function
class Neuron :
    def __init__(self, weights, bias) :
        self.weights = weights
        self.bias = bias
    def feedforward(self, inputs) :          # Process input
        Sum = np.dot (self.weights , inputs) + self.bias
        return f(Sum)
weights = np.array([-1., 1.])              # w1 = -1, w2 = 1
bias = 0
n = Neuron(weights , bias)
x = np.array ([12 ,8])                    # x1 = 12 , x2 = 8
print (n.feedforward(x))
# output: 0.01798620996209156
```

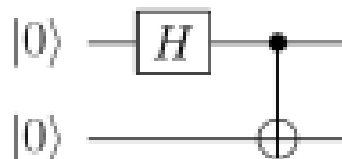
Quantum Computing



Hadamard gate H : converts qubits that are eigenstates of Z to ones that are eigenstates of X :

$$H |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle, \quad H |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle, \quad (12.48)$$

$$H = (|+\rangle \langle 0| + |-\rangle \langle 1|) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (12.49)$$



A quantum circuit for creating an entangled state $|\beta_{00}\rangle$.

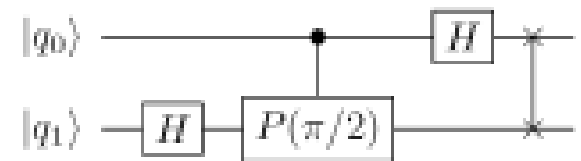
IBM Quantum Computer

The screenshot shows the IBM Quantum Composer interface. On the left is a library of quantum operations including Hadamard (H), CNOT, Toffoli, and various rotation gates (RZ, RX, RY, RXX, RZZ, U). The main workspace contains a quantum circuit visualization. Below the circuit is a bar chart showing the probability of computational basis states (00 and 11). To the right is a Bloch sphere visualization with a blue dot representing the state. Further right is a code editor showing OpenQASM 2.0 code:

```

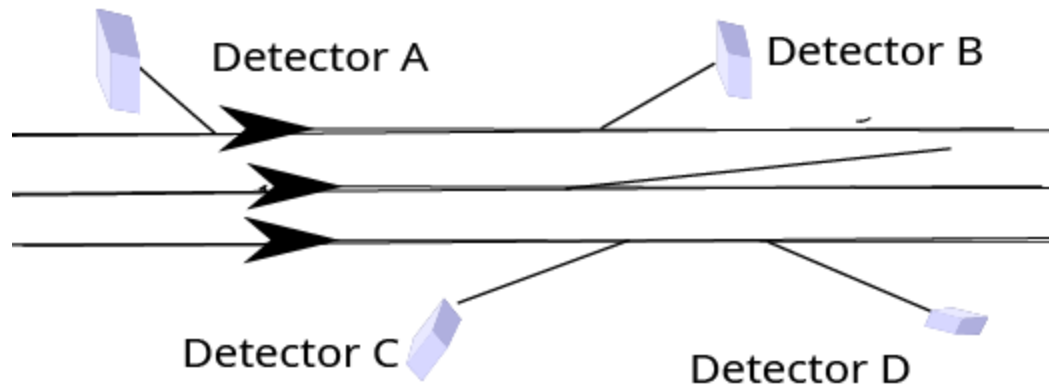
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 cnot q[0], q[1];
5 cnot q[1], q[0];
6
7 reset q[0];
8 h q[0];
9 barrier q[0];
10 h q[1];
11 measure q[0] -> c[0];
    
```

This block provides a detailed view of a quantum circuit for two qubits, $q[0]$ and $q[1]$. The circuit consists of a Hadamard gate on $q[0]$ followed by a CNOT gate with $q[0]$ as control and $q[1]$ as target. Below the circuit is a bar chart showing the amplitude of the computational basis states (00, 01, 10, 11). The chart shows equal amplitudes for states 00 and 11. To the right is a phase plot showing the phase of the state, which is $\pi/2$ for state 00 and $3\pi/2$ for state 11. An "Output state" box displays the state vector:

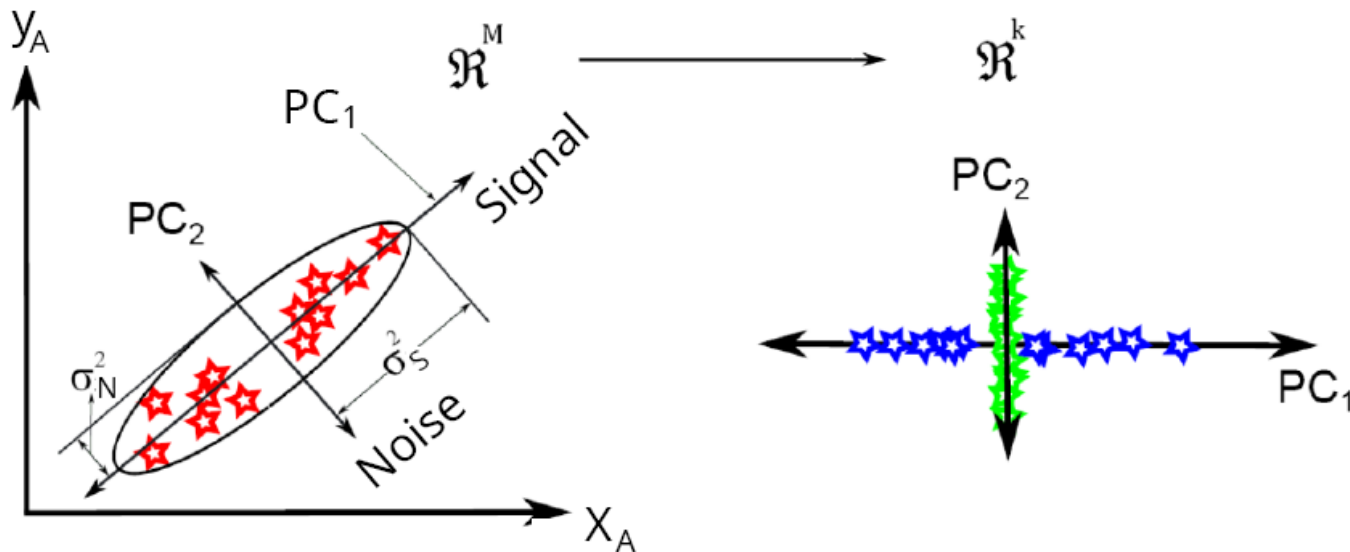
$$[0.707+0j, 0+0j, 0+0j, 0.707+0j]$$


Quantum Fourier transform circuit for 2-qubits.

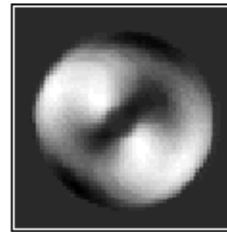
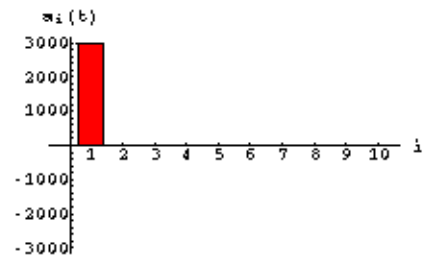
Principal Components Analysis



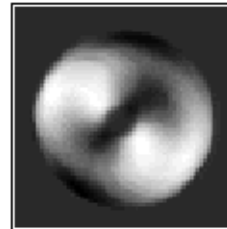
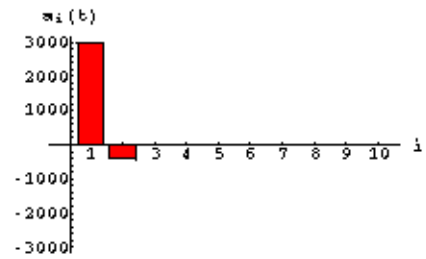
$$\mathbf{X}' = [x'_a \quad y'_a \quad x'_b \quad y'_b \quad x'_c \quad y'_c \quad x'_d \quad y'_d].$$



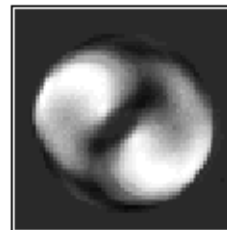
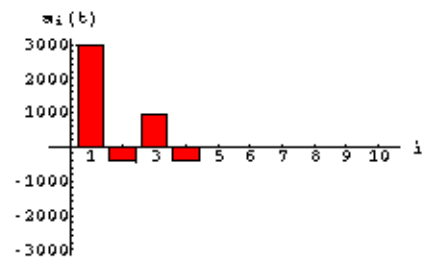
Successive KL Reconstructions (Frame 0001)



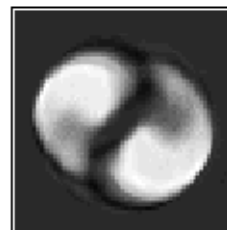
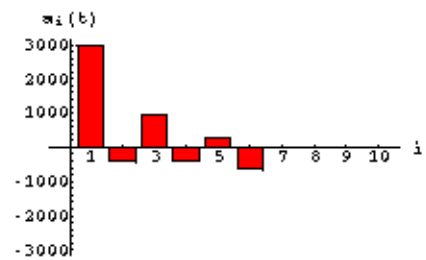
1 Modes



2 Modes



4 Modes



6 Modes