A New Belief System

Random Matrix Theory Applied to Deep Belief Signaling Networks

A Formal Belief System

$$\text{Unity of Knowledge} = I(ABB_{openAI_{GYM}}(GAN;NAS;ME;MLE;AI)) \tag{1}$$

$$Inference = Belief(I(ABB_{openAI_{GYM}}(GAN;NAS;ME;MLE;AI))) \tag{2}$$

$$Z \rightarrow Inference \tag{3}$$

$$\text{Then, to learn on the inference:} \tag{4}$$

$$min_T \sum_i \sum_j T_{ij} Z_{ij} \sum_i \sum_j T_{ij} Z_{ij} + \frac{a}{2} ||T||_F^2 + \frac{a}{2} ||T||_2^2 s.t. ||T|| = n \tag{5}$$

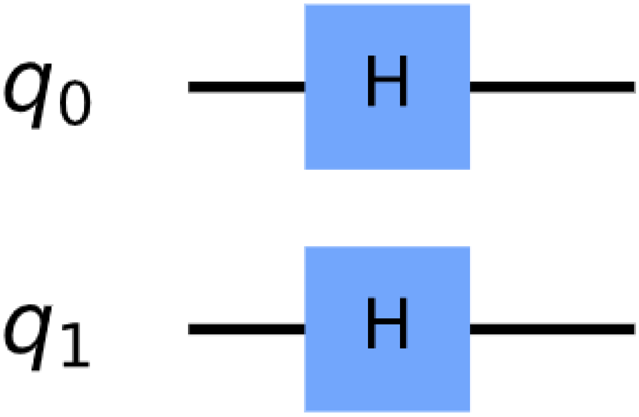
Python and Qiskit Implementation

```
# Run inference on information shared between random populations of...
belief_prop = bp.random(population, environments, neural_architectures:
neural_ode, gan, cnn, rnn; depth: multi, ...)

# Analyze intersection of neural architectures and environments(graph
signal processing)
GSP.engine(analysis(union for belief_prop), algo_seq: [forward, backward,
forward])
```

`GSP.engine` can be further optimized through quantum topological search:

```
# Initialization
import matplotlib.pyplot as plt
import numpy as np

# Importing Qiskit
from qiskit import IBMQ, Aer, QuantumCircuit, ClassicalRegister,
QuantumRegister, execute
from qiskit.providers.ibmq import least_busy
from qiskit.quantum_info import Statevector

# Import basic plot tools
```

```python
from qiskit.visualization import plot_histogram

# Initialize quantum components
n = 2 # qubits
grover_circuit = QuantumCircuit(n)
grover_circuit = initialize_s(grover_circuit, [0,1])
grover_circuit.draw()

def initialize_s(qc, qubits):
    """Apply a H-gate to 'qubits' in qc"""
    for q in qubits:
        qc.h(q)
    return qc
```



. . . . And so on

See references for quantum computing, graph signal processing, and belief propagation: 📖