Backgrounder for a Quantum Kernel

**Idea:**

To have a shallow QML kernel so that we exploit NISQ capabilities. We get data points (x, y) such that x is data point and y is a label (+1, -1), which denotes whether x is specified within a class. All such x’s which are with +1 labels are in the same class, and all x’s labeled as -1 are in the other class for a binary classifier.

This is a simple description for the proposed classifier. The issue now is data embedding, and how to classify given a set of data points. Given a feature map for the x’s, we have the related Unitary transform and also the equation for the same as:

|𝜙(𝒙)⟩ = 𝒰(𝒙)|0 ⨂𝑛 ⟩ = (⊗𝑞=1 𝑛 𝑅𝑧(𝑥𝑞))𝑈2 𝑛 ent (⊗𝑞=1 𝑛 (𝑅𝑦(𝑥𝑞)𝑅𝑧(𝑥𝑞)𝐻)) |0 ⨂𝑛 ⟩,

NOTE: Uent is an entangling gate between the nearest two qubits. This is needed so that the circuit does not go too deep, and also far away CNOTs are noisy.

**Kernel**

The above equation describes the feature map. The estimation is about the kernel, ⟨𝜙(𝒙)|𝜙(𝒙′)⟩|^2. The x’s are data points. The inner product distance defines the kernel. To get an estimate of the distance there is a Hadamard gate which makes a superposition of all the points. The inner product is calculated by the earlier equation describing the kernel.

So after some elementary transformations the equation for the kernel is now:

𝐾(𝒙, 𝒙′) = |⟨𝜙(𝒙)|𝜙(𝒙′)⟩|^2 = |⟨0 ⨂𝑛 |𝒰† (𝒙)𝒰(𝒙 ′ )|0 ⨂𝑛 ⟩|^2.

This equation can be seen as the basis for our quantum circuit.