Learning Algebraic Models of Quantum Entanglement

JAFFALI Hamza and OEDING Luke

PhD. advisors: HOLWECK Frederic and MEROLLA Jean-Marc

FEMTO-ST, University of Bourgone Franche-Comté
Auburn University, Alabama, USA

Thursday, November 28th 2019

Quantum Entanglement is an important resource in Quantum Information and Quantum Computations, useful and sometimes essential for Quantum Algorithms and Quantum Communication Protocols.

Quantum Entanglement is an important resource in Quantum Information and Quantum Computations, useful and sometimes essential for Quantum Algorithms and Quantum Communication Protocols.

Being able to distinguish between separable and entangled states, or being able to recognize a specific type of entanglement become important to understand more precisely the role and the nature of entanglement in such computations.

Quantum Entanglement is an important resource in Quantum Information and Quantum Computations, useful and sometimes essential for Quantum Algorithms and Quantum Communication Protocols.

Being able to distinguish between separable and entangled states, or being able to recognize a specific type of entanglement become important to understand more precisely the role and the nature of entanglement in such computations.

In this work, we are interested in the classification and characterization of the entanglement under the action of the group SLOCC (*Stochastic Local Operation with Classical Communication*).

$$G_{SLOCC} = \mathrm{SL}_2(\mathbb{C}) \times \mathrm{SL}_2(\mathbb{C}) \times \cdots \times \mathrm{SL}_2(\mathbb{C})$$

Normal form	Entanglement class
$\|00\rangle + \|11\rangle$	Entangled (EPR)
$\|00\rangle$	Separable

Table: SLOCC classification of entanglement for 2-qubit states.

Normal form	Entanglement class
$\|00\rangle + \|11\rangle$	Entangled (EPR)
$\|00\rangle$	Separable

Table: SLOCC classification of entanglement for 2-qubit states.

Normal form	Entanglement class
$\|000\rangle + \|111\rangle$	GHZ
$\|001\rangle + \|010\rangle + \|100\rangle$	W
$\|000\rangle + \|110\rangle$	Biseparable AB–C
$\|000\rangle + \|101\rangle$	Biseparable B–CA
$\|000\rangle + \|011\rangle$	Biseparable A–BC
$\|000\rangle$	Separable

Table: SLOCC classification of entanglement for 3-qubit states.

$$G_{abcd} = \frac{a+d}{2}\big(|0000\rangle + |1111\rangle\big) + \frac{a-d}{2}\big(|0011\rangle + |1100\rangle\big) + \frac{b+c}{2}\big(|0101\rangle + |1010\rangle\big) + \frac{b-c}{2}\big(|0110\rangle + |1001\rangle\big)$$

$$L_{abc_2} = \frac{a+b}{2}\big(|0000\rangle + |1111\rangle\big) + \frac{a-b}{2}\big(|0011\rangle + |1100\rangle\big) + c\big(|0101\rangle + |1010\rangle\big) + |0110\rangle$$

$$L_{a_2b_2} = a\big(|0000\rangle + |1111\rangle\big) + b\big(|0101\rangle + |1010\rangle\big) + |0011\rangle + |0110\rangle$$

$$L_{ab_3} = a\big(|0000\rangle + |1111\rangle\big) + \frac{a+b}{2}\big(|0101\rangle + |1010\rangle\big) + \frac{a-b}{2}\big(|0110\rangle + |1001\rangle\big) + \frac{i}{\sqrt{2}}\big(|0001\rangle + |0010\rangle - |0111\rangle - |1011\rangle\big)$$

$$L_{a_4} = a\big(|0000\rangle + |0101\rangle + |1010\rangle + |1111\rangle\big) + i|0001\rangle + |0110\rangle - i|1011\rangle$$

$$L_{a_20_{3\oplus\bar{1}}} = a\big(|0000\rangle + |1111\rangle\big) + |0011\rangle + |0101\rangle + |0110\rangle$$

$$L_{0_{5\oplus\bar{3}}} = |0000\rangle + |0101\rangle + |1000\rangle + |1110\rangle$$

$$L_{0_{7\oplus\bar{1}}} = |0000\rangle + |1011\rangle + |1101\rangle + |1110\rangle$$

$$L_{0_{3\oplus\bar{1}}0_{3\oplus\bar{1}}} = |0000\rangle + |0111\rangle$$

Table: 9 Vestraete *et al.* (corrected) families for 4-qubits entanglement

However it is one of the rare cases (with the 3-qutrit case) where we can regroup all SLOCC orbits into families depending on parameters, while the number of orbits is infinite. Providing a full classification of SLOCC entanglement classes is a already a **hard problem** for 5-qubits systems, for instance.

However it is one of the rare cases (with the 3-qutrit case) where we can regroup all SLOCC orbits into families depending on parameters, while the number of orbits is infinite. Providing a full classification of SLOCC entanglement classes is a already a **hard problem** for 5-qubits systems, for instance.

Need to develop new tools, in order to characterize or distinguish several entanglement classes for multipartite systems.

However it is one of the rare cases (with the 3-qutrit case) where we can regroup all SLOCC orbits into families depending on parameters, while the number of orbits is infinite. Providing a full classification of SLOCC entanglement classes is a already a **hard problem** for 5-qubits systems, for instance.

Need to develop new tools, in order to characterize or distinguish several entanglement classes for multipartite systems.

Our idea is to use Machine Learning techniques to bring and build interesting tools. Our goal is not to provide a full classification, but only to recognize several types of entanglement, and thus being able to discriminate some non-SLOCC-equivalent states.

Presentation Outline

Machine Learning

Machine Learning is an emergent field in Computer Science, which aim is to study and develop algorithms, permitting computer systems to perform a specific task without using explicit instructions.

Machine Learning

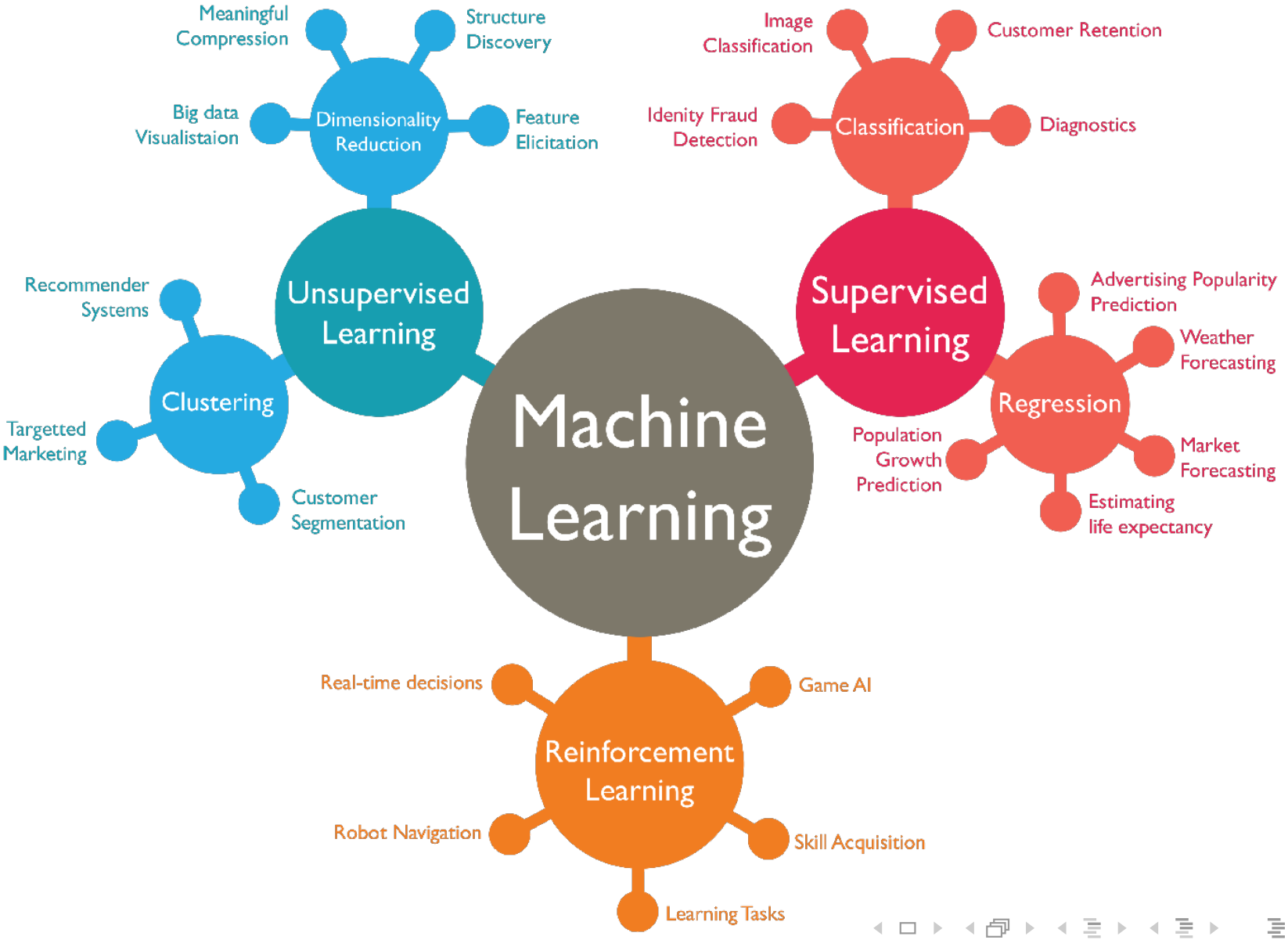
Machine Learning is an emergent field in Computer Science, which aim is to study and develop algorithms, permitting computer systems to perform a specific task without using explicit instructions.

These technologies are also studied in the field of Quantum Computations, and many researchers are actually working in developing Quantum Machine Learning algorithms[a], exploiting the quantum speed-up to improve such algorithms.

Our approach is the opposite: we leverage classical Machine Learning to study and classify Quantum Entanglement.

[a] Alessandro Luongo *et al.* (2019). q-means: A quantum algorithm for unsupervised machine learning. In NeurIPS 2019.

Different approaches



Machine Learning – Supervised Learning

Principle

Supervised Learning is the machine learning task of learning a function that maps a given input to its correct output, by exploiting an initial knowledge of the problem.

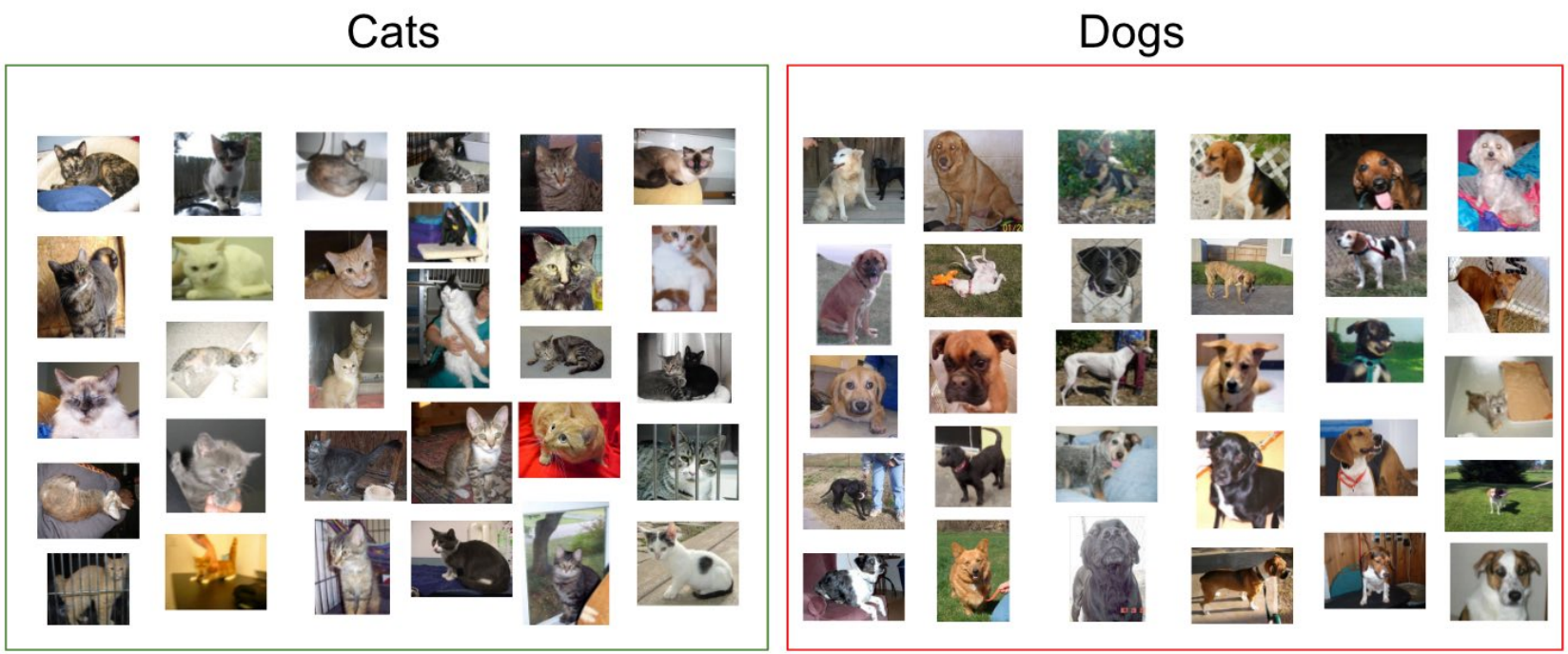
Why supervised ?

The training step require initial informations about the problem, and most of the time initial correct data to train the Machine Learning achitecture. We give to the machine what we call a **Training Dataset**. We can think of supervised learning as **teaching by example**, and in that sense, we are supervising the learning process of the machine.

The goal is to be able to make correct predictions for new data, with a high accuracy.

Different approaches – Supervised Learning – Applications

Classification



Sample of cats & dogs images from Kaggle Dataset

Supervised Learning and Quantum states

We focus here on the case of pure qubit states. A general n-qubit system $|\psi\rangle \in \mathcal{H} = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ can be represented as a $N = 2^n$ dimensional vector $x_\psi = (a_0, a_1, \ldots, a_{N-1}) \in \mathbb{C}^{2^n}$, with $|\psi\rangle$ expressed in the computational basis as:

$$|\psi\rangle = a_0|0\ldots00\rangle + a_1|0\ldots01\rangle + \cdots + a_{N-1}|1\ldots11\rangle$$

We will thus use the vector x_ψ as the feature vector for the training database. We construct then the training database:

$$D_{Train} = \{(x_{\psi_1}, y_1), ..., (x_{\psi_M}, y_M)\}$$

where y_i can refer to the entanglement class ('0' for separable, '1' for entangled) for instance.

In our work, we focused on 3 different problems of classification.

Separable states and Entangled states

The set of **separable states** define a unique orbit under the action of SLOCC (it is the orbit of the state $|00\ldots0\rangle$). Any state which is not separable is in fact **entangled**. We want then to build a binary classifier to distinguish between separable and entangled pure states.

Separable states and Entangled states

The set of **separable states** define a unique orbit under the action of SLOCC (it is the orbit of the state $|00\ldots0\rangle$). Any state which is not separable is in fact **entangled**. We want then to build a binary classifier to distinguish between separable and entangled pure states.

A separable state $|\psi_{sep}\rangle$ is a state that can be written as the tensor product of each qubit representing each particle of the system. In algebraic geometry, it is known that points of the **Segre variety** are in fact separable states.

$$|\psi_{sep}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$$

Separable states and Entangled states

The set of **separable states** define a unique orbit under the action of SLOCC (it is the orbit of the state $|00\ldots0\rangle$). Any state which is not separable is in fact **entangled**. We want then to build a binary classifier to distinguish between separable and entangled pure states.

A separable state $|\psi_{sep}\rangle$ is a state that can be written as the tensor product of each qubit representing each particle of the system. In algebraic geometry, it is known that points of the **Segre variety** are in fact separable states.

$$|\psi_{sep}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$$

How do we build the training dataset ?

- We sample separable states by computing the Kronecker product of n random qubits.
- We generate entangled states by summing several separable states (with high probability the resulting state is not separable).

Degenerate and Non-degenerate states

Degenerate states are points on the dual of the Segre variety, i.e. they define the zero-set of an homogeneous polynomial: the **hyperdeterminant**.

The hyperdeterminant is the generalization of the determinant for multidimensional matrices, and it is considered as a possible measure of entanglement [4].

Degenerate and Non-degenerate states

Degenerate states are points on the dual of the Segre variety, i.e. they define the zero-set of an homogeneous polynomial: the **hyperdeterminant**.

The hyperdeterminant is the generalization of the determinant for multidimensional matrices, and it is considered as a possible measure of entanglement [4].

$$\begin{aligned}\Delta_{222}(|\psi\rangle) = {} & a_{000}^2 a_{111}^2 + a_{011}^2 a_{100}^2 + a_{010}^2 a_{101}^2 + a_{001}^2 a_{110}^2 - 2a_{000}a_{011}a_{100}a_{111} - 2a_{000}a_{010}a_{101}a_{111} \\ & - 2a_{000}a_{001}a_{110}a_{111} + 4a_{000}a_{011}a_{101}a_{110} - 2a_{010}a_{011}a_{100}a_{101} - 2a_{001}a_{011}a_{100}a_{110} - 2a_{001}a_{010}a_{101}a_{110} \\ & + 4a_{001}a_{010}a_{100}a_{111}\end{aligned}$$

Degenerate and Non-degenerate states

Degenerate states are points on the dual of the Segre variety, i.e. they define the zero-set of an homogeneous polynomial: the **hyperdeterminant**.

The hyperdeterminant is the generalization of the determinant for multidimensional matrices, and it is considered as a possible measure of entanglement [4].

$$\begin{aligned}\Delta_{222}(|\psi\rangle) = {} & a_{000}^2 a_{111}^2 + a_{011}^2 a_{100}^2 + a_{010}^2 a_{101}^2 + a_{001}^2 a_{110}^2 - 2a_{000}a_{011}a_{100}a_{111} - 2a_{000}a_{010}a_{101}a_{111} \\ & - 2a_{000}a_{001}a_{110}a_{111} + 4a_{000}a_{011}a_{101}a_{110} - 2a_{010}a_{011}a_{100}a_{101} - 2a_{001}a_{011}a_{100}a_{110} - 2a_{001}a_{010}a_{101}a_{110} \\ & + 4a_{001}a_{010}a_{100}a_{111}\end{aligned}$$

$$\Delta_{222}(|\psi_{sep}\rangle) = \Delta_{222}(|\psi_{bisep}\rangle) = \Delta_{222}(|W\rangle) = 0$$

$$\Delta_{222}(|GHZ\rangle) \neq 0$$

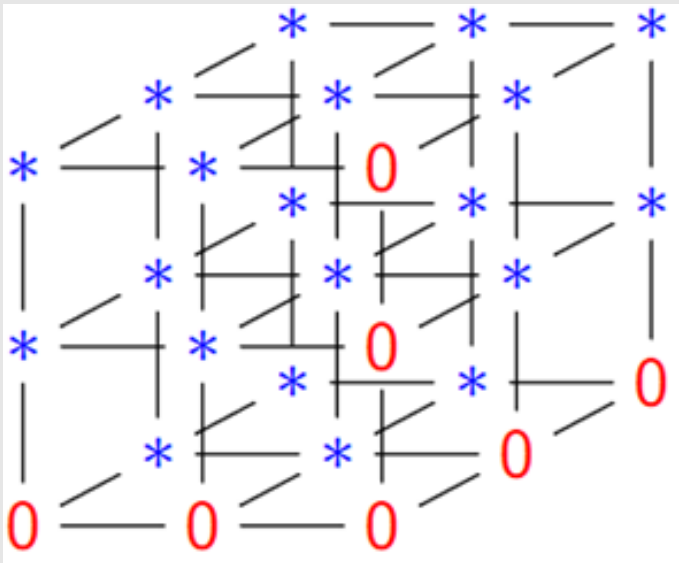
Degenerate and Non-degenerate states

We want to build a binary classifier to distinguish between degenerate and non-degenerate states.

How do we build the training dataset ?

- We sample degenerate states by applying a random SLOCC operation on the state of this form (figure to the right)
- We generate random tensor of $\mathcal{H} = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ (with high probability the resulting state is non-generate).

Example for $3 \times 3 \times 3$ states



Rank of a quantum state

Tensor rank can be used as an **algebraic measure of entanglement**[a]. It has been used in to study the entanglement of states generated by Grover's algorithm [3] .

We recall that $\mathcal{H}$ is the Hilbert space where live n-qubits states, i.e. $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$. Then $|\psi\rangle \in \mathcal{H}$ is said to be of :

- **rank 1** if $|\psi\rangle = |u_1\rangle \otimes |u_2\rangle \otimes \cdots \otimes |u_n\rangle$, with $|u_i\rangle \in \mathbb{C}^2$,
- **rank** r if $|\psi\rangle = |\psi_1\rangle + |\psi_2\rangle + \cdots + |\psi_r\rangle$, where $|\psi_i\rangle$ are rank 1 tensors, where r **is minimal**.

Thus, **rank one** tensors correspond to **separable states** and every tensor which is **not of rank one** should be considered as **entangled**.

[a] Brylinski, J. L. (2002). Algebraic measures of entanglement. Mathematics of quantum computation, 3-23.

Border rank and Secant varieties

One can also introduce the notion of **border rank** which is related with the notion of rank.

A state $|\psi\rangle \in \mathcal{H}$ has border rank $\leq R$ if there exists a family of rank R states $\{|\psi_\epsilon\rangle \mid \epsilon > 0\}$ such that $\lim_{\epsilon \to 0} |\psi_\epsilon\rangle = |\psi\rangle$.

States with border rank at most R represent points on the **Secant Variety** σ_R. For 3-qubits, we know that $|GHZ\rangle$ has rank and border rank equal to 2, while $|W\rangle$ has rank equal to 3 and border rank equal to 2.

How do we generate data ?

We sum R random states of rank 1 to generate states with border rank R (with some probability). If $|\psi\rangle$ has border rank of R, then $|\psi\rangle \in \sigma_R$ and $|\psi\rangle \notin \sigma_{R-1}$.

Neural Networks

Principle

The Neural Network (or connectionist system) is a computing system which goal is to reproduce several functions and basis structure of human brain.

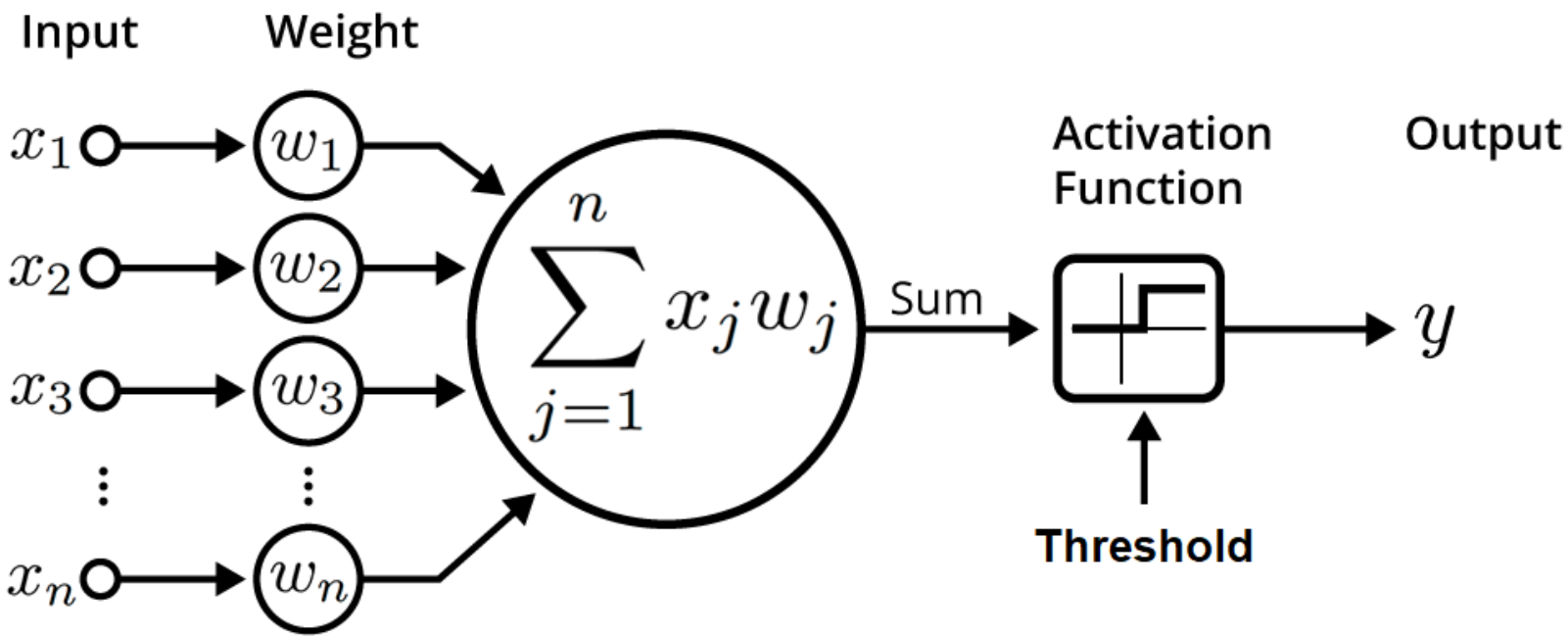
An Artificial Neural Network (ANN) is characterized by 3 main components :

1. Artificial neuron
2. Architecture of the network
3. Learning algorithm

Artificial Neuron

The first model was proposed by McCulloch and Pitts in 1943.

1. Inputs coming from other neurons
2. Weights (synaptic weights)
3. Weighted sum of the inputs
4. Threshold
5. Activation Function
6. Output

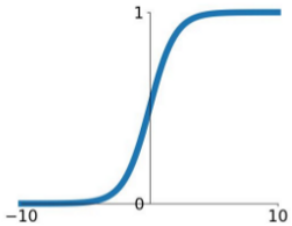


Artificial Neuron – Activation functions

Activation functions permit to introduce **non-linearity** between the inputs and the outputs: it allows to consider more difficult problems.

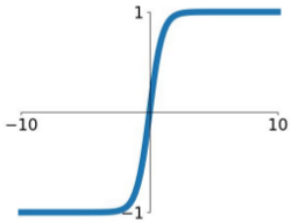
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



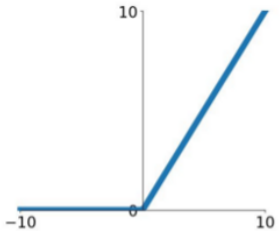
tanh

$$\tanh(x)$$



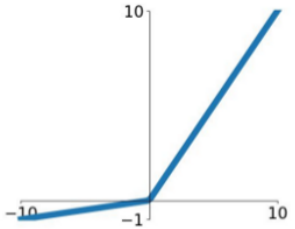
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

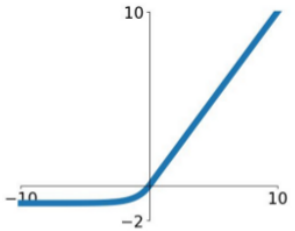


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

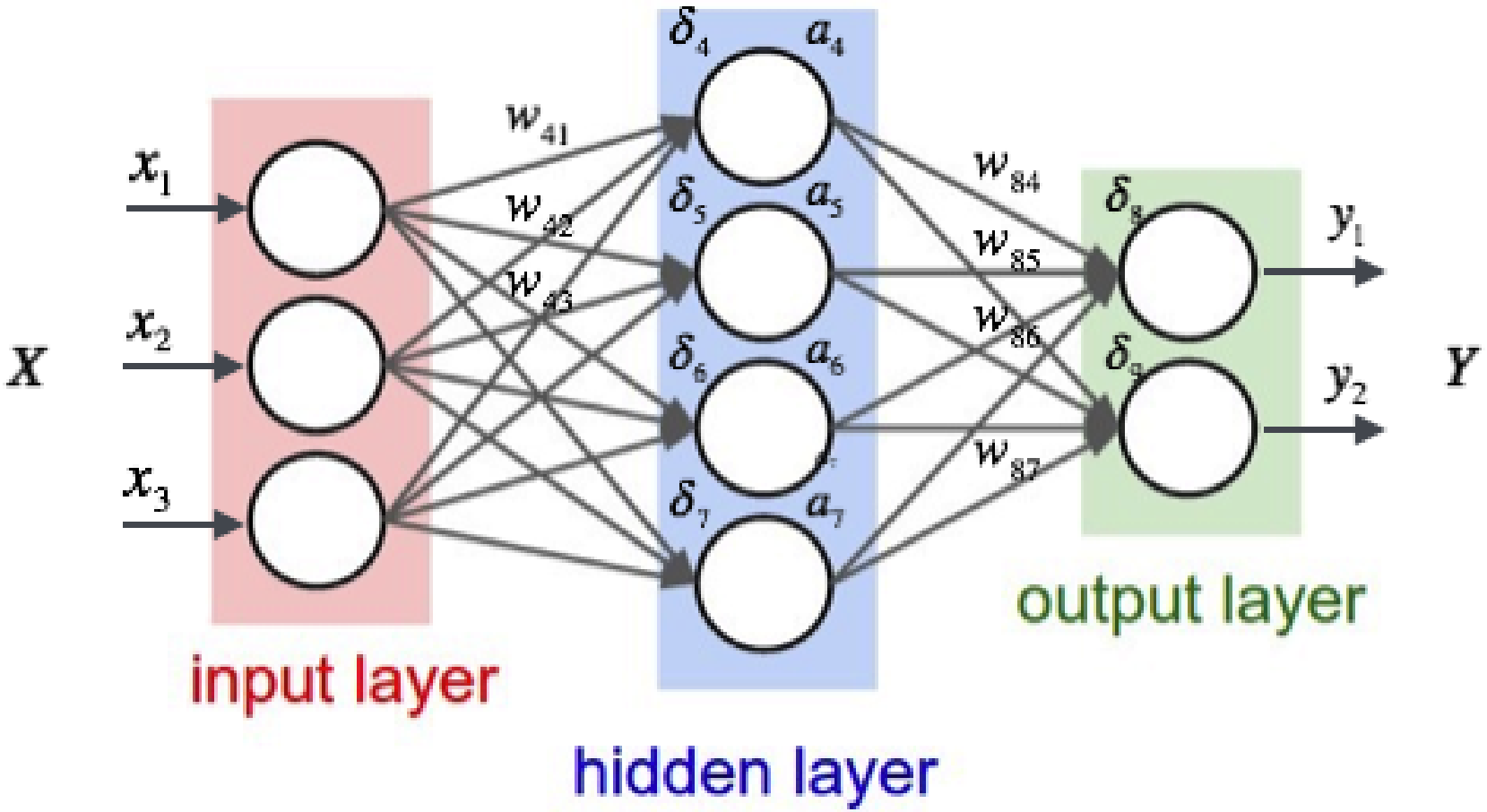
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Multi-Layer Perceptron – MLP

Most of classification problems can not be solved using a single neuron. We need to introduce more complex structures and architectures in the network. It is the aim of the **Multilayer Perceptron** model.



Multi-Layer Perceptron – Learning process

How to predict the correct output ?

- We define an **error function**, which measure the error between the predicted output and the correct output (from the training dataset), and this for each input of the dataset
- The learning process is then an optimization process to find the weights of the network that minimize the error function

Multi-Layer Perceptron – Learning process

How to predict the correct output ?

- We define an **error function**, which measure the error between the predicted output and the correct output (from the training dataset), and this for each input of the dataset
- The learning process is then an optimization process to find the weights of the network that minimize the error function

Designing the network before learning

The architecture of an Artificial Neural Network mainly depends on the following choices:

- Number of layers
- Number of neurons in each layer
- Activation functions for each layer/neuron

Algebraic varieties and Polynomial equations

There is **no general rules** for choosing the right architecture, and it will depend on the classification problem studied.

In our case, we want to distinguish between points **inside and outside a given algebraic variety**. Algebraic varieties are defined as the zero locus of a set of **homogeneous polynomials**.

We want to design Artificial Neural Network to model polynomial equations, and thus detect states that satisfy a set of polynomial equations.

First example: linear equation in n variables

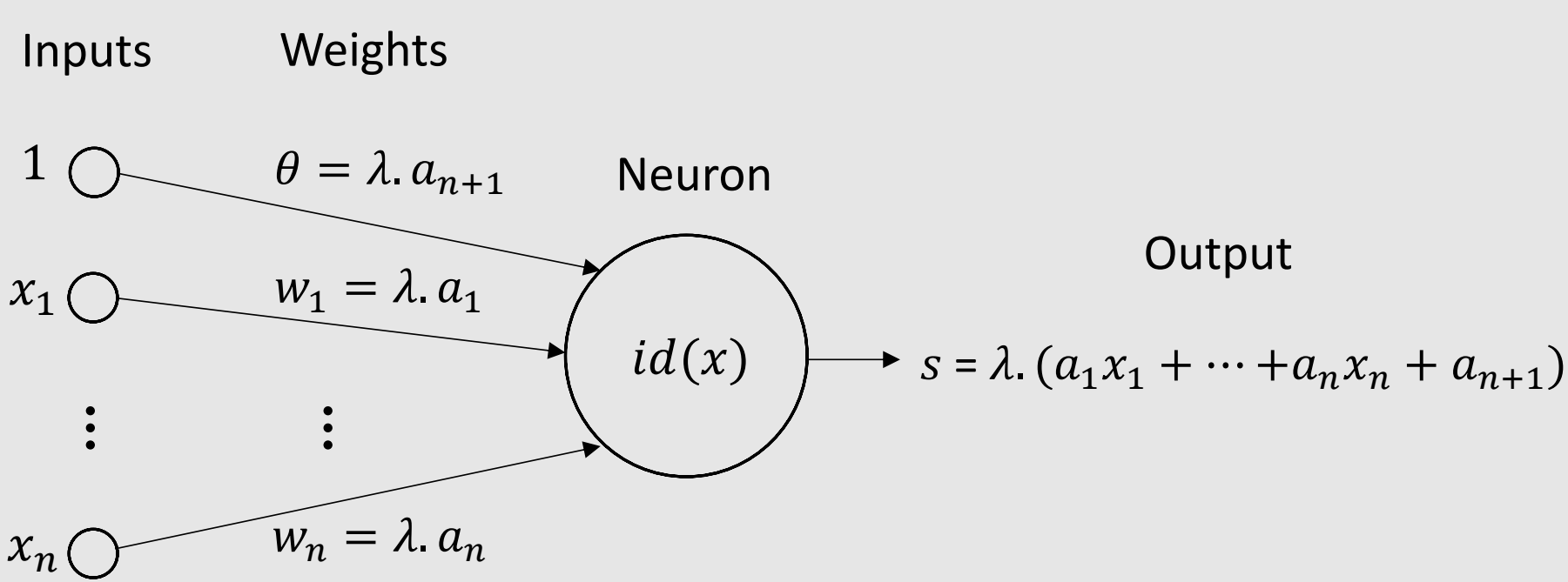
Let us suppose we want to classify points inside and outside a linear subspace defined by the following linear equation in n variables $(x_1, \ldots, x_n) \in \mathbb{R}$:

$$a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_n x_n + a_{n+1} = 0$$

First example: linear equation in n variables

Let us suppose we want to classify points inside and outside a linear subspace defined by the following linear equation in n variables $(x_1, \ldots, x_n) \in \mathbb{R}$:

$$a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n + a_{n+1} = 0$$



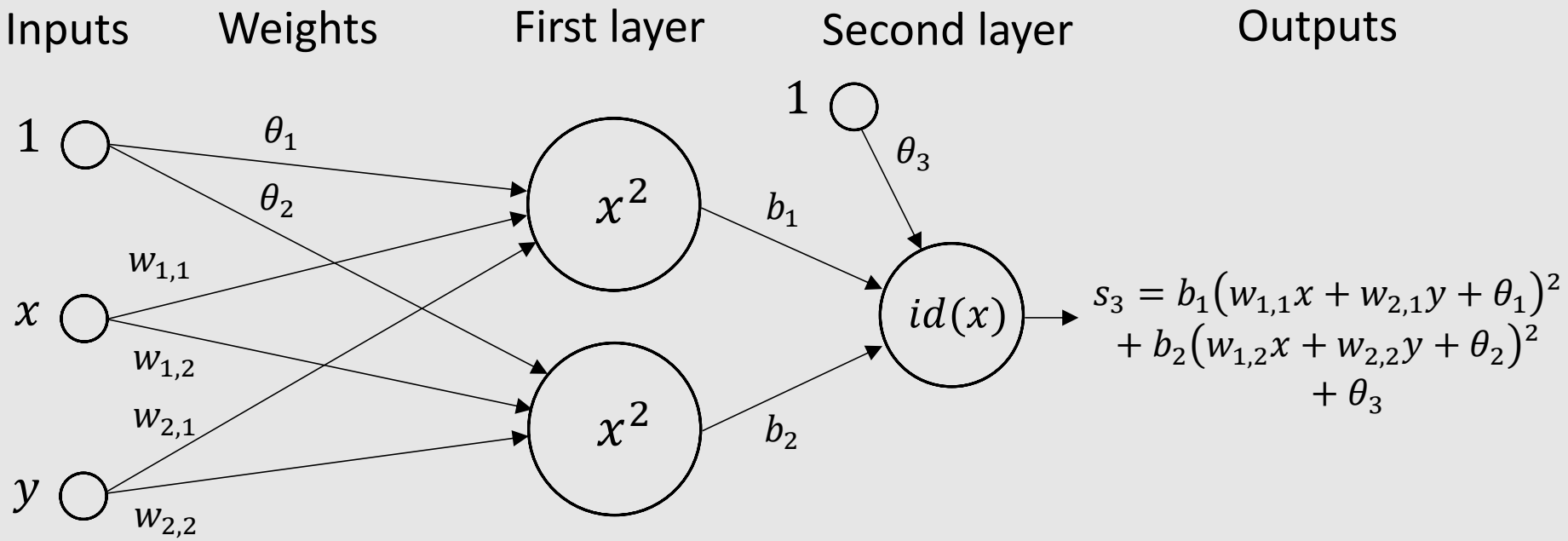
We may need to deal with polynomials of higher degrees. Let us suppose we want to model the circle equation $x^2 + y^2 - r^2 = 0$ with an Artificial Neural Network.

We may need to deal with polynomials of higher degrees. Let us suppose we want to model the circle equation $x^2 + y^2 - r^2 = 0$ with an Artificial Neural Network.

The idea here is to introduce a **square activation function** to generate degree 2 terms from the weighted sum.

We may need to deal with polynomials of higher degrees. Let us suppose we want to model the circle equation $x^2 + y^2 - r^2 = 0$ with an Artificial Neural Network.

The idea here is to introduce a **square activation function** to generate degree 2 terms from the weighted sum.



We want generalize this result to **any homogeneous polynomial of degree** d, since secant varieties, the Segre variety and its dual variety are defined by a set of homogeneous polynomials.

The question is: **how much neurons** with activation function $x \mapsto x^d$ should we combine in the first layer ?

We want generalize this result to **any homogeneous polynomial of degree** d, since secant varieties, the Segre variety and its dual variety are defined by a set of homogeneous polynomials.

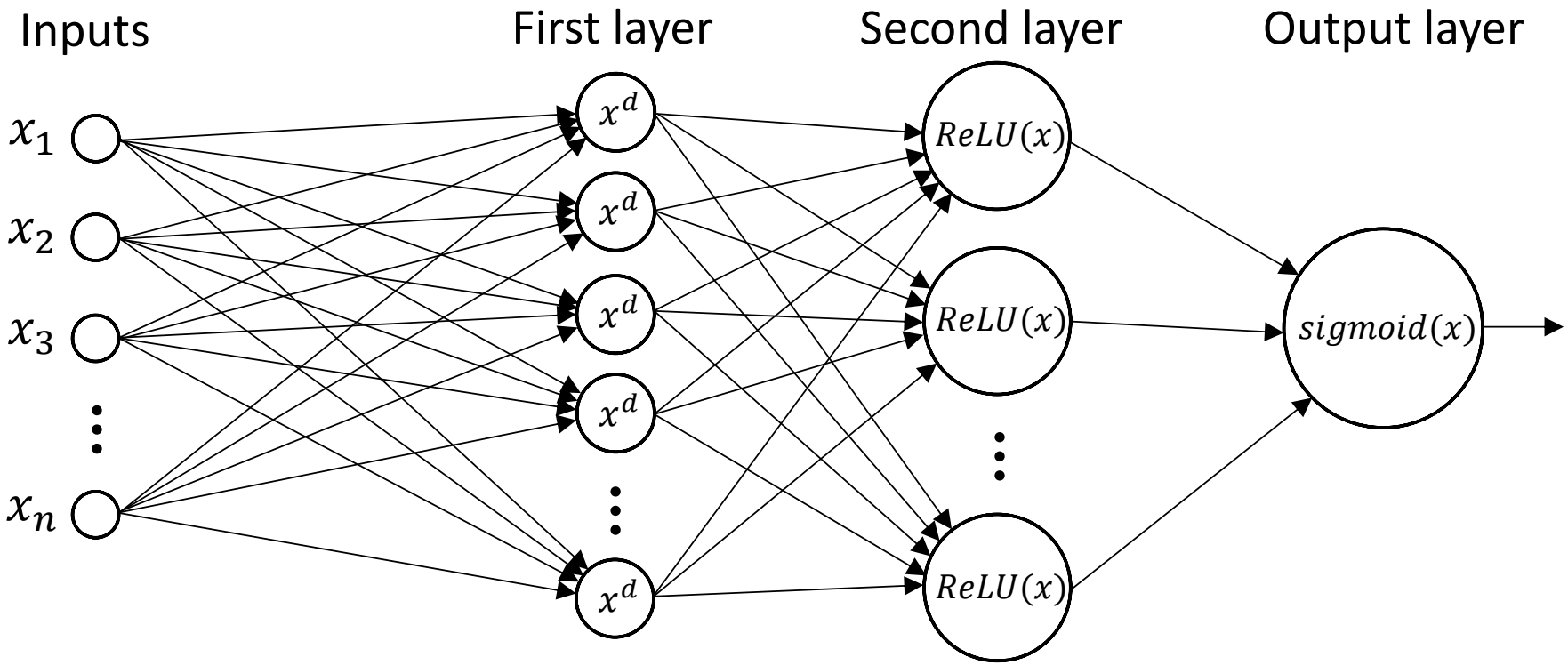
The question is: **how much neurons** with activation function $x \mapsto x^d$ should we combine in the first layer ?

Theorem – Alexander-Hirschowitz

Any homogeneous polynomial p of degree d in n variables can be written as the sum of $T = \lceil \frac{1}{n}\binom{d+n-1}{d} \rceil$ d-th powers of linear forms, s.t.

$$p(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{T} \left(\sum_{i=1}^{n} a_{ij} x_i \right)^d ,$$

except in the following cases:
$\{d = 2\}$; $\{n = 2, d = 4\}$; $\{n = 3, d = 4\}$; $\{n = 4, d = 3\}$; $\{n = 4, d = 4\}$.


Inputs
First layer
Second layer
Output layer
x_1
x_2
x_3
x_n
x^d
$ReLU(x)$
$sigmoid(x)$

Classifiers and accuracy

Classifier 1 – Binary classifier – Separable/Entangled

Tensor size	Architecture	Training acc.	Validation acc.	Testing acc.	Loss
2×2	(4,4,1)	96.65%	96.60%	96.63%	0.092
$2 \times 2 \times 2$	(21,8,1)	94.57%	94.06%	94.44%	0.15
$2^{\times 4}$	(1188,8,1)	91.72%	91.60%	91.33%	0.26
$3 \times 3 \times 3$	(332,12,1)	94.68%	92.89%	92.94%	0.15

Tensor size	Architecture	Training acc.	Validation acc.	Testing acc.	Loss
2×2	(100,50,25,16,1)	98.92%	98.78%	98.83%	0.043
$2 \times 2 \times 2$	(100,50,25,16,1)	97.80%	97.42%	97.55%	0.074
$2^{\times 4}$	(100,50,25,16,1)	99.62%	99.50%	99.53%	0.016
$2^{\times 5}$	(100,50,25,16,1)	98.83%	98.55%	98.55%	0.037
$3 \times 3 \times 3$	(100,50,25,16,1)	98.55%	98.01%	97.92%	0.051

Classifiers and accuracy

Classifier 2 – Binary classifier – Degenerate/Non-degenerate

Tensor size	Architecture	Training acc.	Validation acc.	Testing acc.	Loss
$2 \times 2 \times 2$	(60,10,1)	92.49%	92.18%	92.09%	0.1837

Tensor size	Architecture	Training acc.	Validation acc.	Testing acc.	Loss
$2 \times 2 \times 2$	(100,50,25,16,1)	93.44%	92.53%	92.74%	0.1629
$2^{\times 4}$	(200,100,50,16,1)	99.50%	95.95%	95.94%	0.01791
$2^{\times 5}$	(100,50,25,16,1)	99.95%	98.74%	98.83%	0.001533
$3 \times 3 \times 3$	(100,50,25,16,1)	98.18%	96.78%	96.83%	0.04770

Classifiers and accuracy

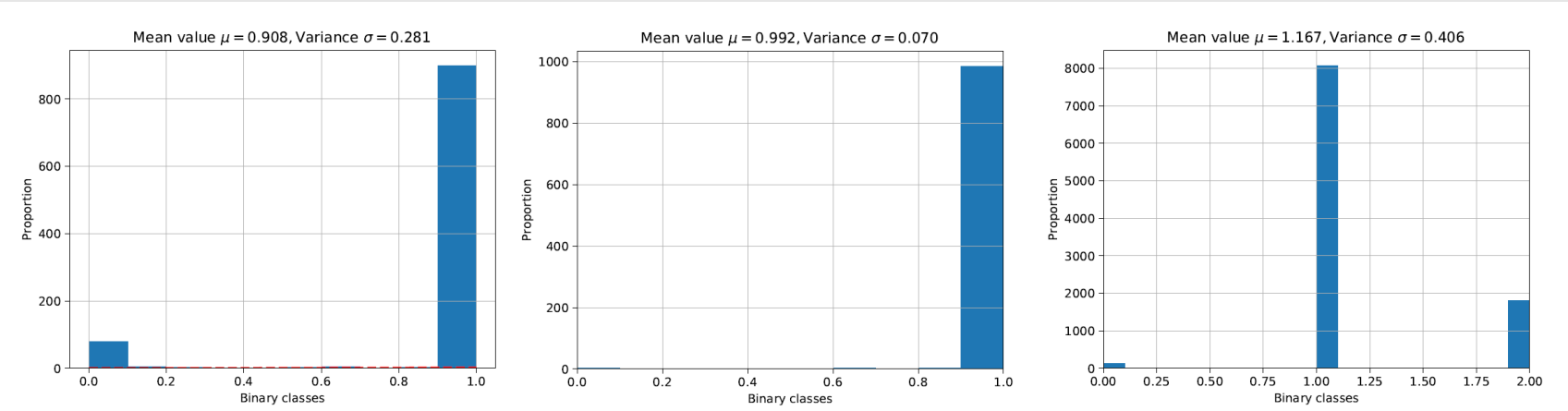
Classifier 3 – Multiclass classifier – Rand and Border rank

Tensor size	Architecture	Training acc.	Validation acc.	Testing acc.	Loss
$2 \times 2 \times 2$	(169,25,3)	88.19%	88.03%	87.95%	0.3028

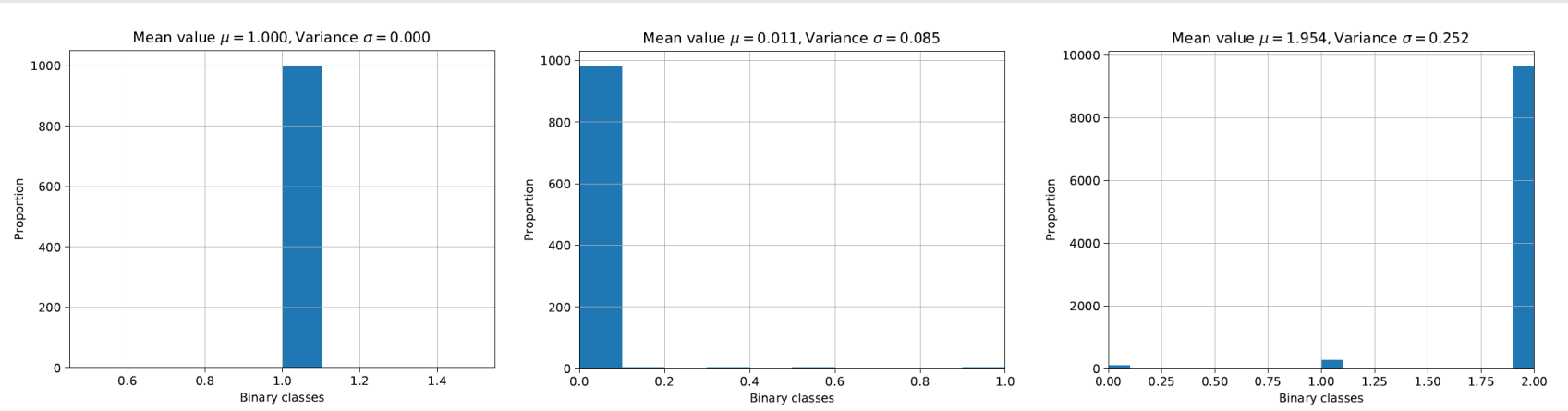
Tensor size	Architecture	Training acc.	Validation acc.	Testing acc.	Loss
$2 \times 2 \times 2$	(200,100,50,25,3)	94.19%	94.07%	93.79%	0.1674
$2^{\times 4}$	(200,100,50,25,3)	85.49%	84.45%	84.47%	0.3144
$2^{\times 5}$	(200,100,50,25,3)	81.39%	79.88%	79.77%	0.4230

Back to 3-qubit classification

$|GHZ\rangle$

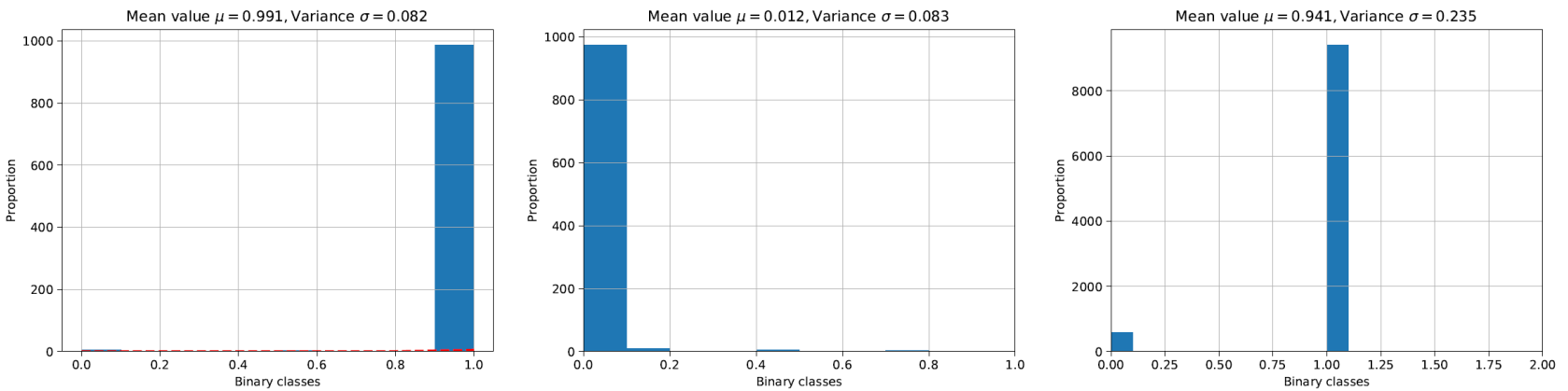


$|W\rangle$



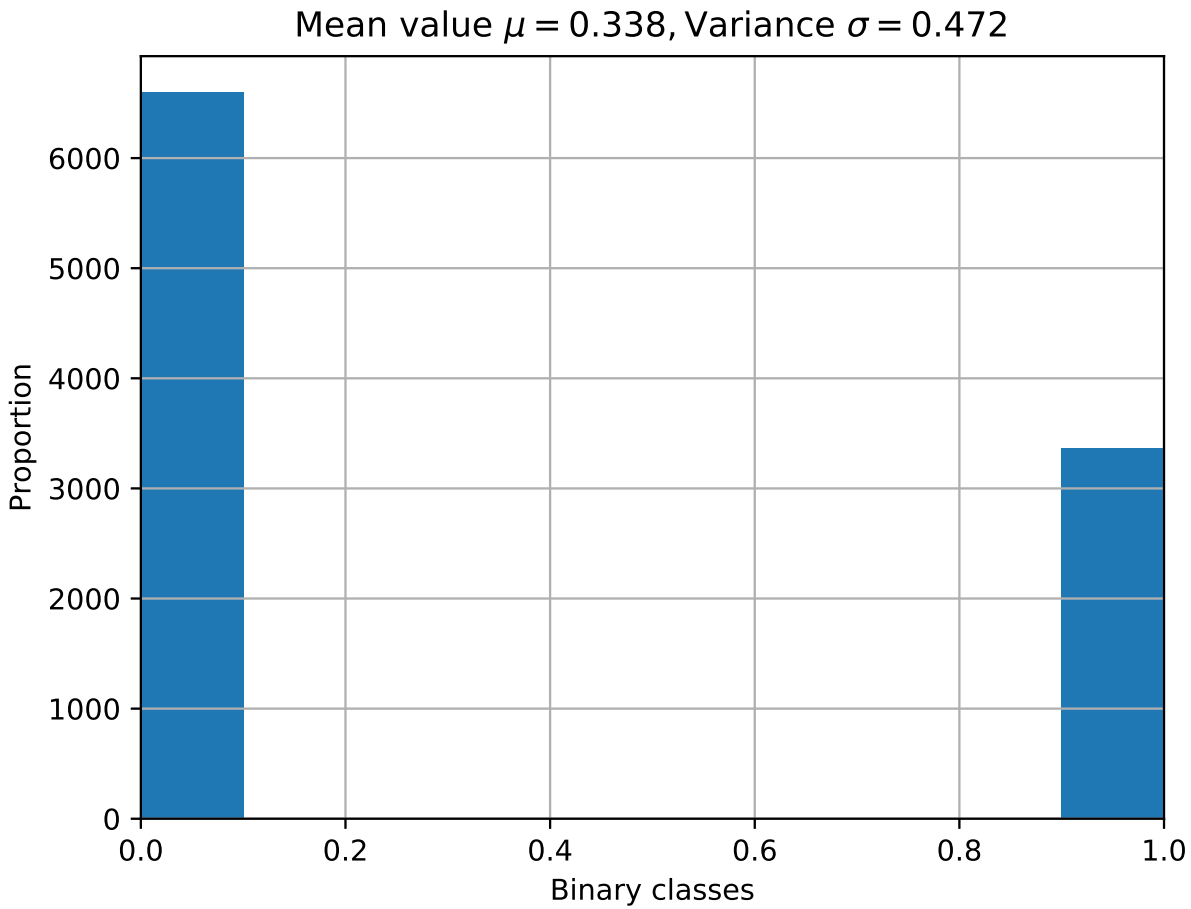
Back to 3-qubit classification

$$|\Psi_{bisep}\rangle$$



5-qubits entanglement – Dual variety

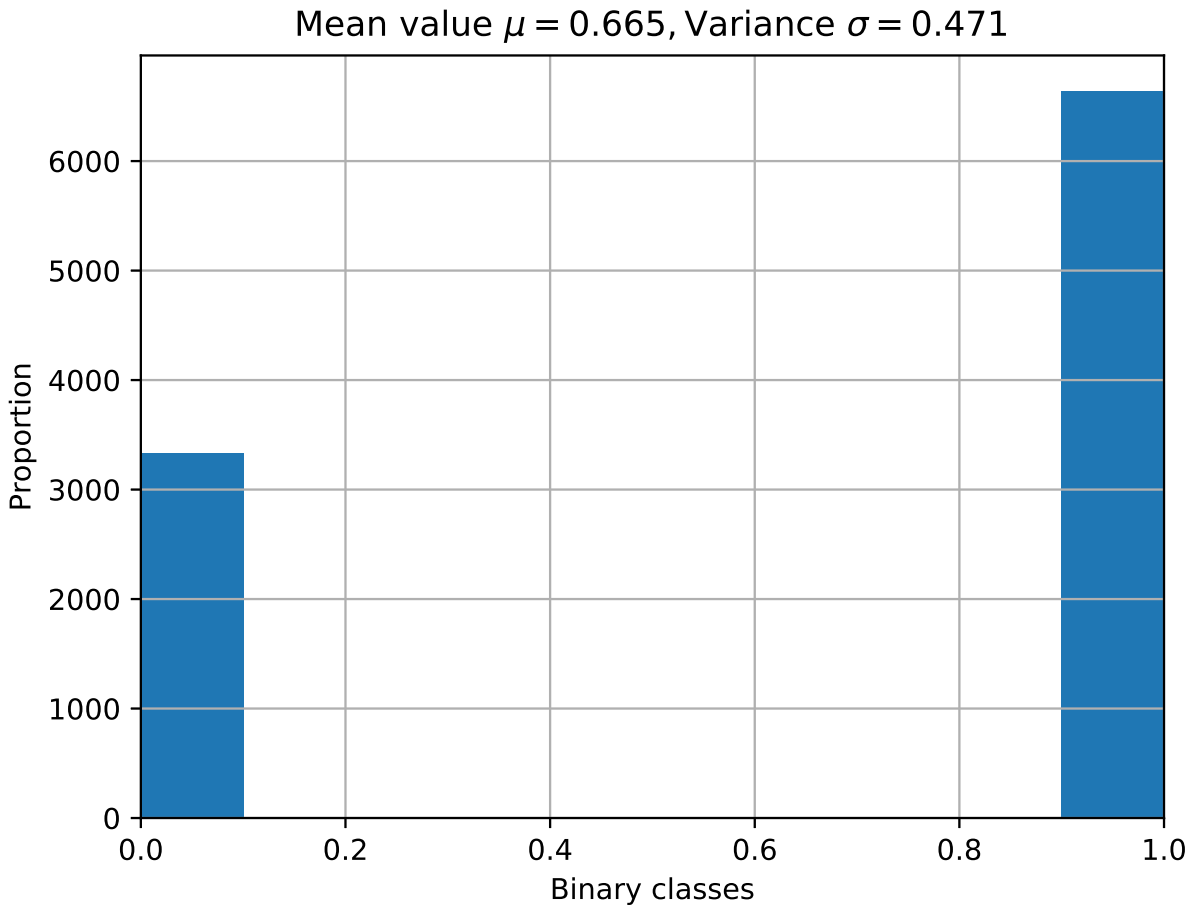
$$|\Phi\rangle = \frac{1}{\sqrt{6}}\left(\sqrt{2}|11111\rangle + |11000\rangle + |00100\rangle + |00010\rangle + |00001\rangle\right)$$



This implies that $\Delta_{22222}(|\Phi\rangle) = 0$

5-qubits entanglement – Dual variety

$$|\delta\rangle = \frac{1}{\sqrt{11}}\big(|00000\rangle + |00100\rangle + |00111\rangle + |01010\rangle - |01101\rangle + |10001\rangle + |10011\rangle + |10111\rangle - |11000\rangle + |11110\rangle\big)$$



This implies that $\Delta_{22222}(|\delta\rangle) \neq 0$

Thank you for your attention - Questions ...

- Kerenidis, I., Landman, J., Luongo, A., & Prakash, A. (2019). q-means: A quantum algorithm for unsupervised machine learning. In NeurIPS 2019 : Thirty-third Conference on Neural Information Processing Systems.
- Brylinski, J. L. (2002). Algebraic measures of entanglement. Mathematics of quantum computation, 3-23.
- Holweck, F., Jaffali, H., & Nounouh, I. (2016). Grover's algorithm and the secant varieties. Quantum Information Processing, 15(11), 4391–4413.
- Gour, G., & Wallach, N. R. (2010). All maximally entangled four-qubit states. Journal of Mathematical Physics, 51(11), 112201.