doitnow

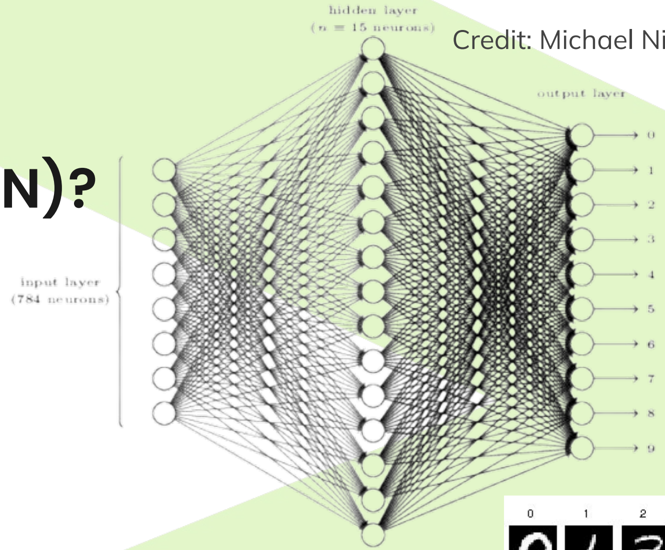
HPC Services

Neural Networks: Classical vs Quantum

Lourens van Niekerk

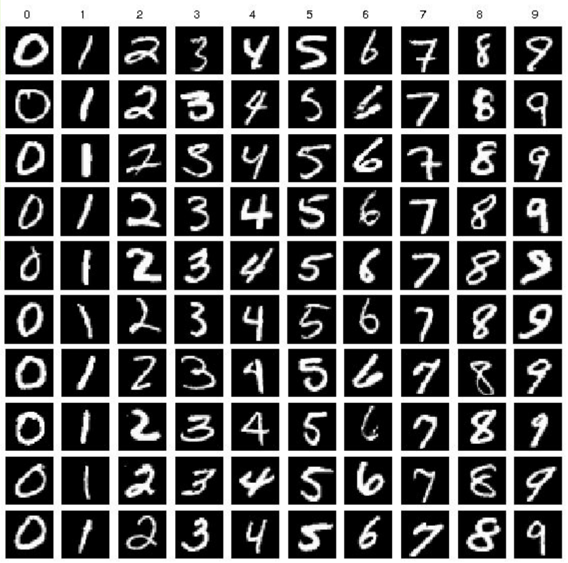
doitnow
HPC Services

What is a neural network (NN)?



Credit: Michael Nielsen

MNIST dataset



- (Artificial) neural networks simulate the learning mechanism of biological neural networks.
- Components:
    - Neurons (units) with weights that scale their input
    - Synapses (connections) with weights that distribute neuron output as input to other neurons
- Mechanism:
    - NN propagates input data through it, into output
    - Weights in NN adjust over iterations to via incentives
    - Incentives can be rewards, example matching or patterns
    - → Reinforcement, Supervised or Unsupervised Learning

Key terminology

Features

Epochs

Noise

Batch size

Learning rate

Backpropagation

Activation function

Regularization

Loss function

Classification & Regression

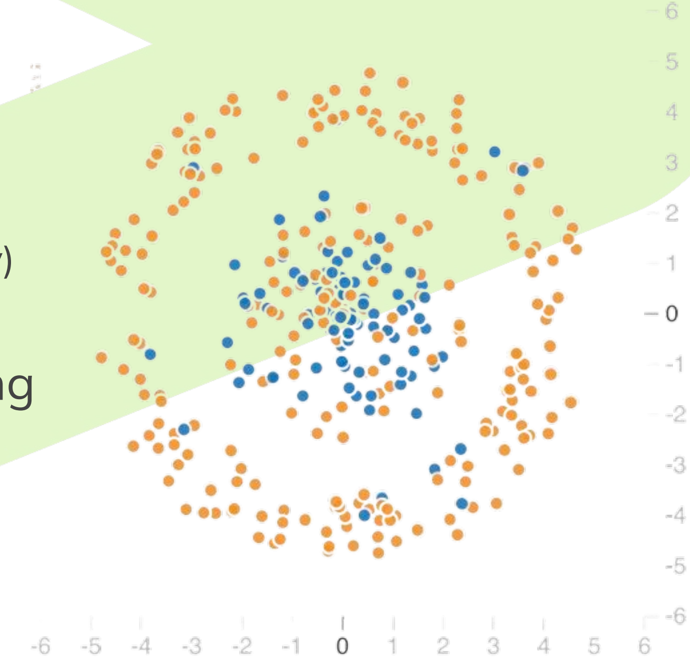
Barren plateaus

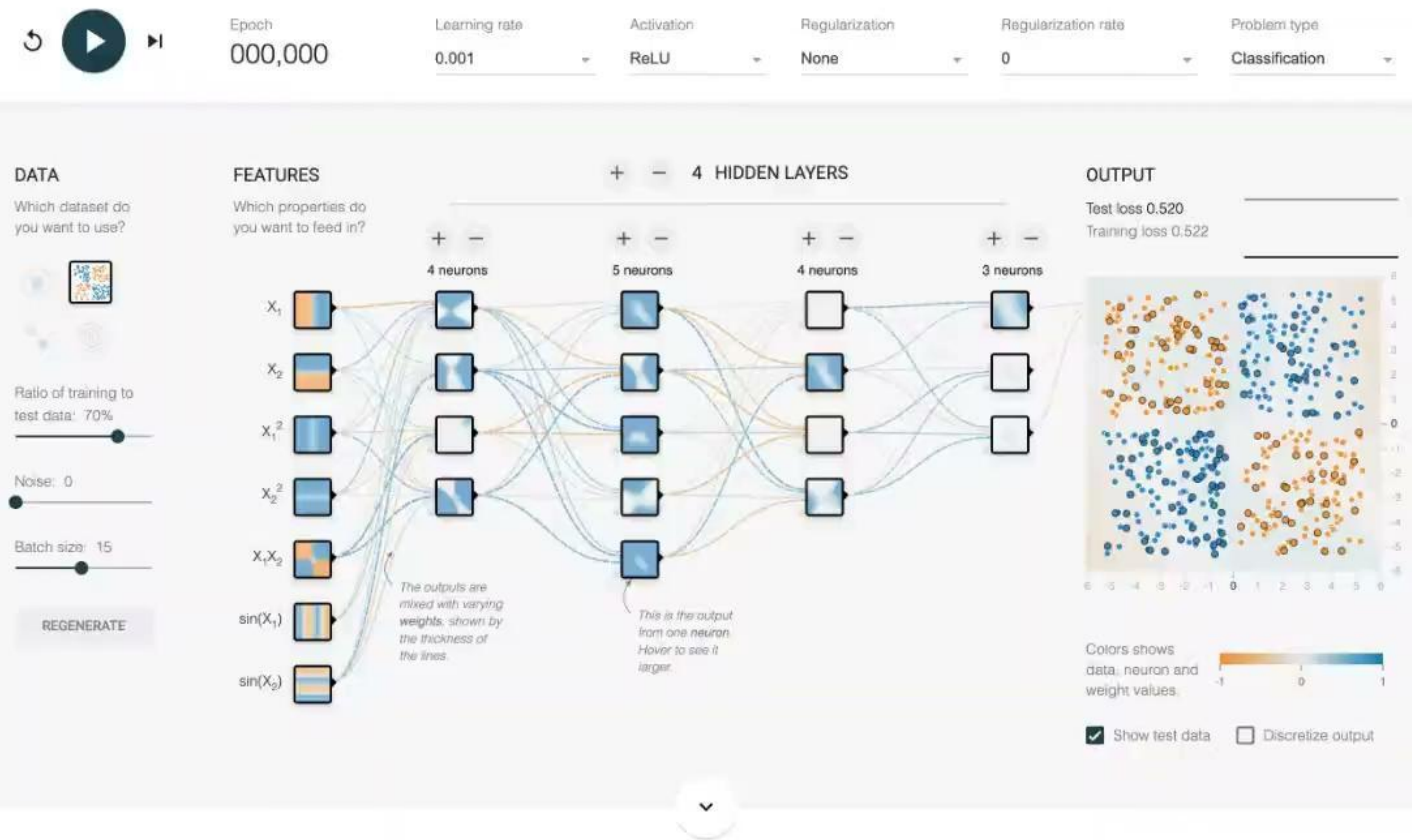
Features, Epochs & Noise

- Features are the inputs to the neural network
  - Could be raw data or transformations
- Epoch = iterations
  - How many full processes on the (NN) model occurred
  - I graduated high school after 18 epochs with a batch size of ~365 days (← my cost increased suddenly)
- Noise is how much misinformation / misinterpretation / external factors are affecting the data
  - Here it only refers to noise in the data
  - This is not noise affecting quantum operations

500 data points

Blue and Orange have spies





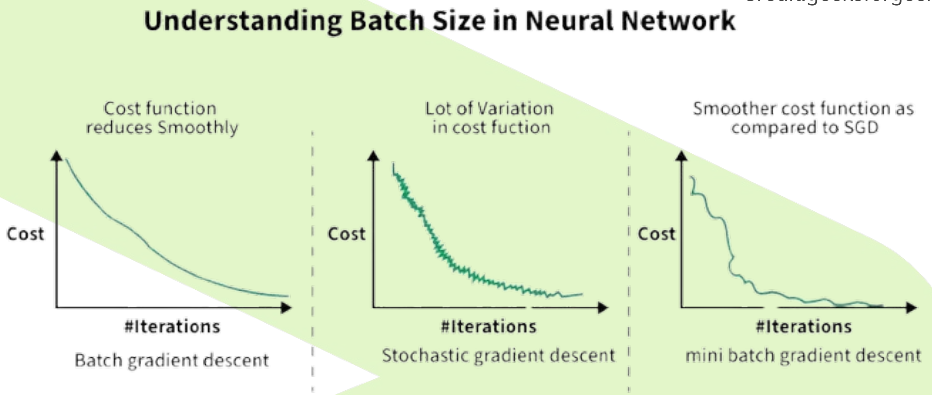
Epoch
000,000
Learning rate
0.001
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 70%
Noise: 0
Batch size: 15
REGENERATE
FEATURES
Which properties do you want to feed in?
X_1
X_2
X_1^2
X_2^2
X_1X_2
$\sin(X_1)$
$\sin(X_2)$
4 HIDDEN LAYERS
4 neurons
5 neurons
4 neurons
3 neurons
The outputs are mixed with varying weights, shown by the thickness of the lines
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.520
Training loss 0.522
Colors shows data, neuron and weight values
-1
0
1
Show test data
Discretize output

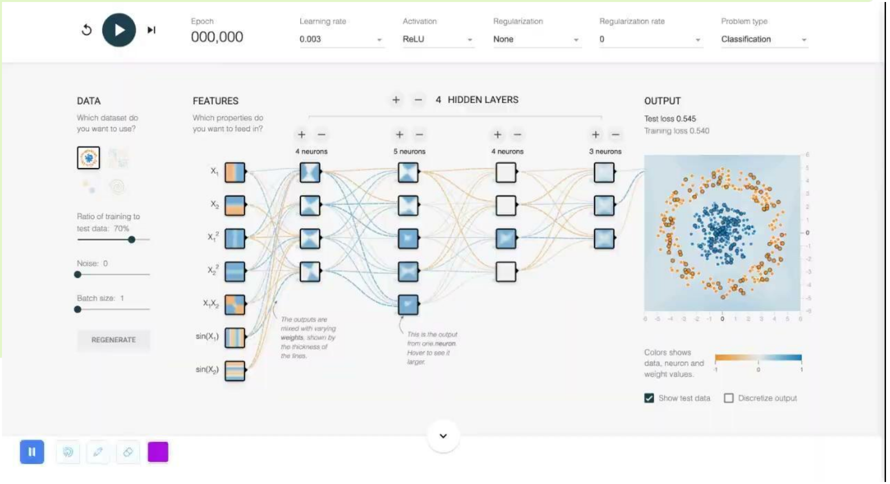
doitnow
HPC Services

Batch size

- Determines how many samples used per forward & backward pass
- After each batch, model makes predictions, and adapts weights
- Different types
    - Batch Gradient Descent (BGD)
        - All training samples
    - Mini-Batch Gradient Descent (MBGD)
        - Somewhere in-between
    - Stochastic Gradient Descent (SGD)
        - Sample of 1

Credit:geeksforgeeks





doitnow
HPC Services

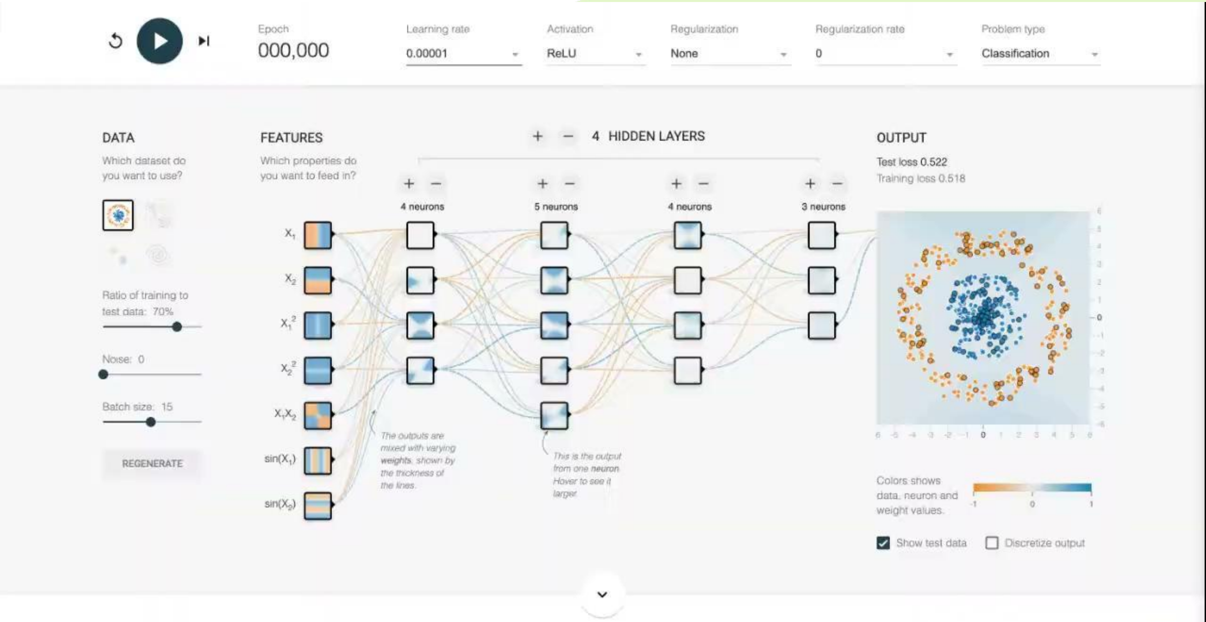
Learning rate

- How much do we allow the model to adjust after each epoch

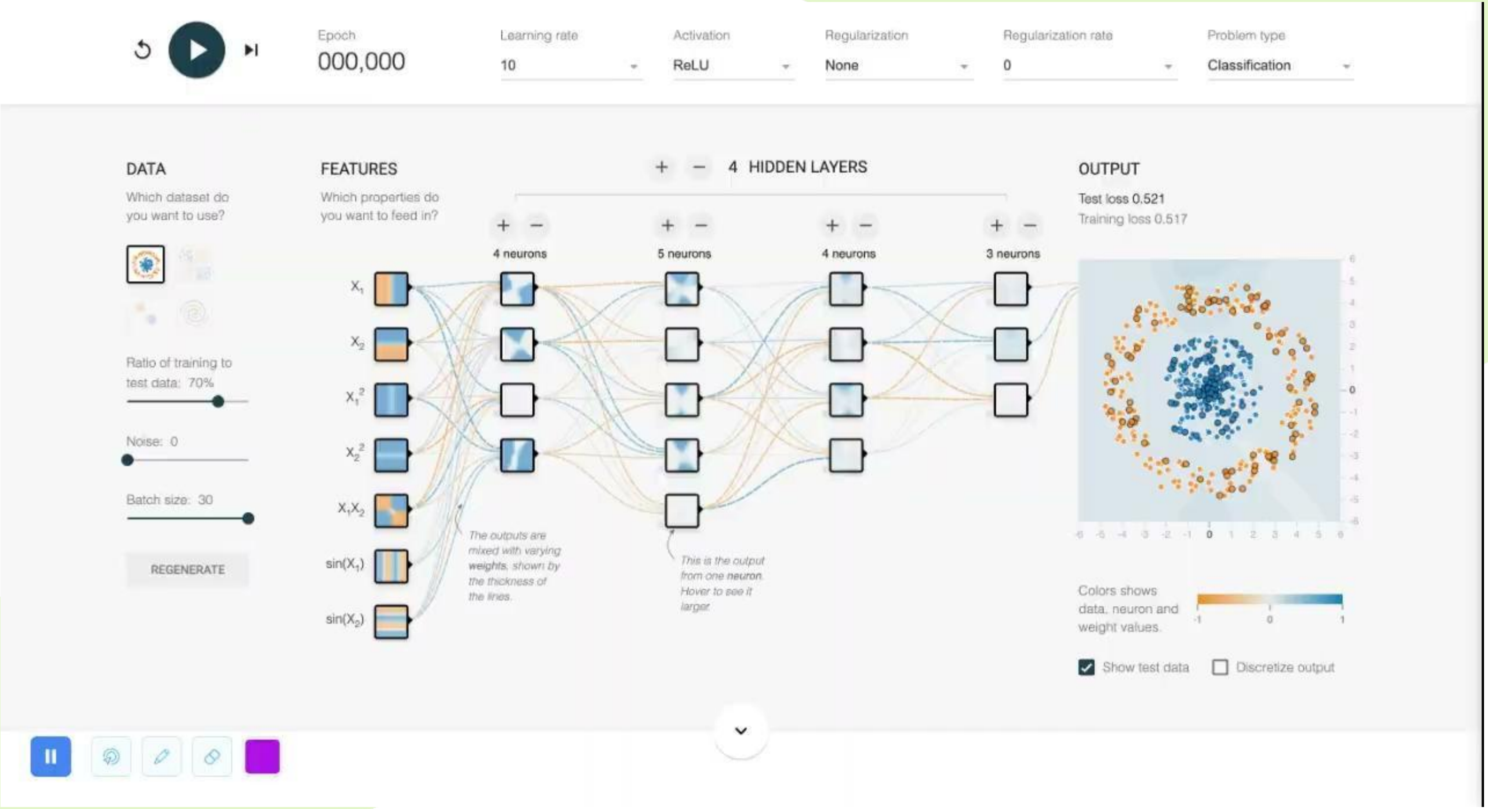
- Too low: forever to train
- Too high: unstable
- Useful to have adaptive learning rate → ADAM
- [Schedules of decay]

$$\omega_{i+1} = \omega_i - \lambda \cdot \nabla L(\omega_i)$$

- ω_i = weights of model
- λ = learning rate
- $\nabla L(\omega_i)$ = gradient of loss function



Unstable learning rate example



Loss function

A medium to facilitate optimization of model outputs (also called cost function)

Often incentive is minimization

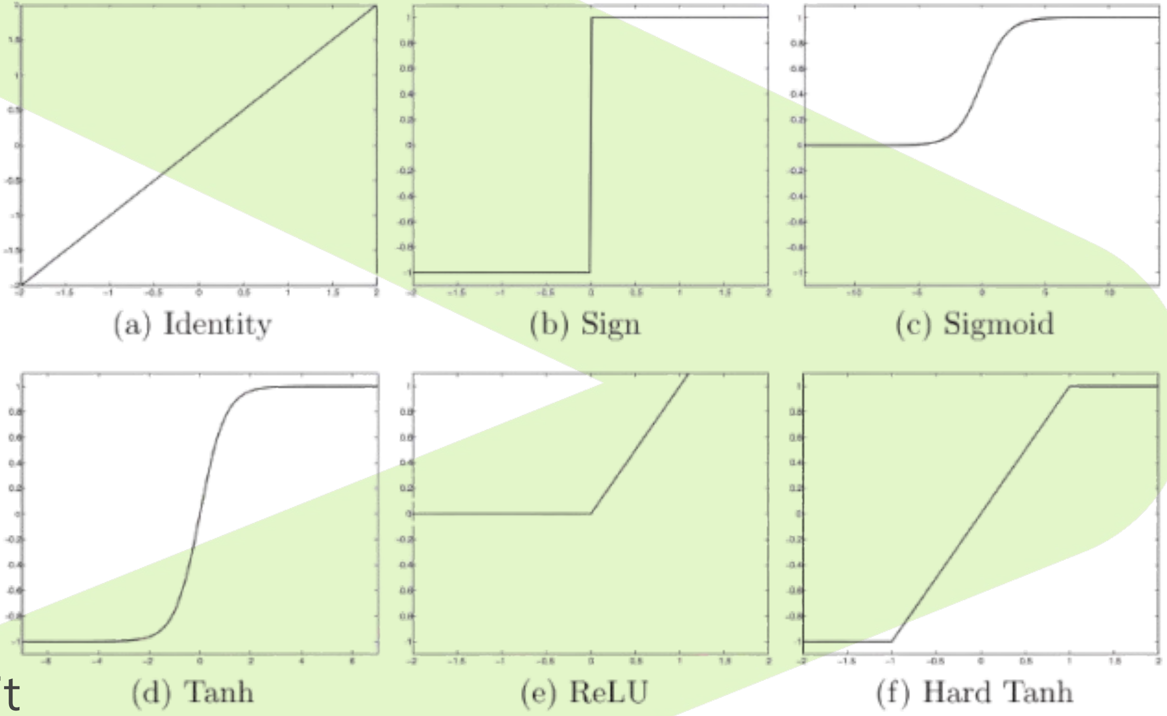
Examples:

- Mean Square Error (MSE) ← sensitive to outliers
- Mean Absolute Error (MAE)
- Binary cross-entropy (Log loss)
- Kullback-Leibler Divergence loss (KL Divergence)
  - Great for probability distributions

doitnow HPC Services

Activation function

- A function applied to neuron output
- Introduces nonlinearity to the model
- Allows classification of real-world complex data
- Perceptron just uses Sign
- ReLU: Rectified Linear Unit
- ReLU favorite, with many variants



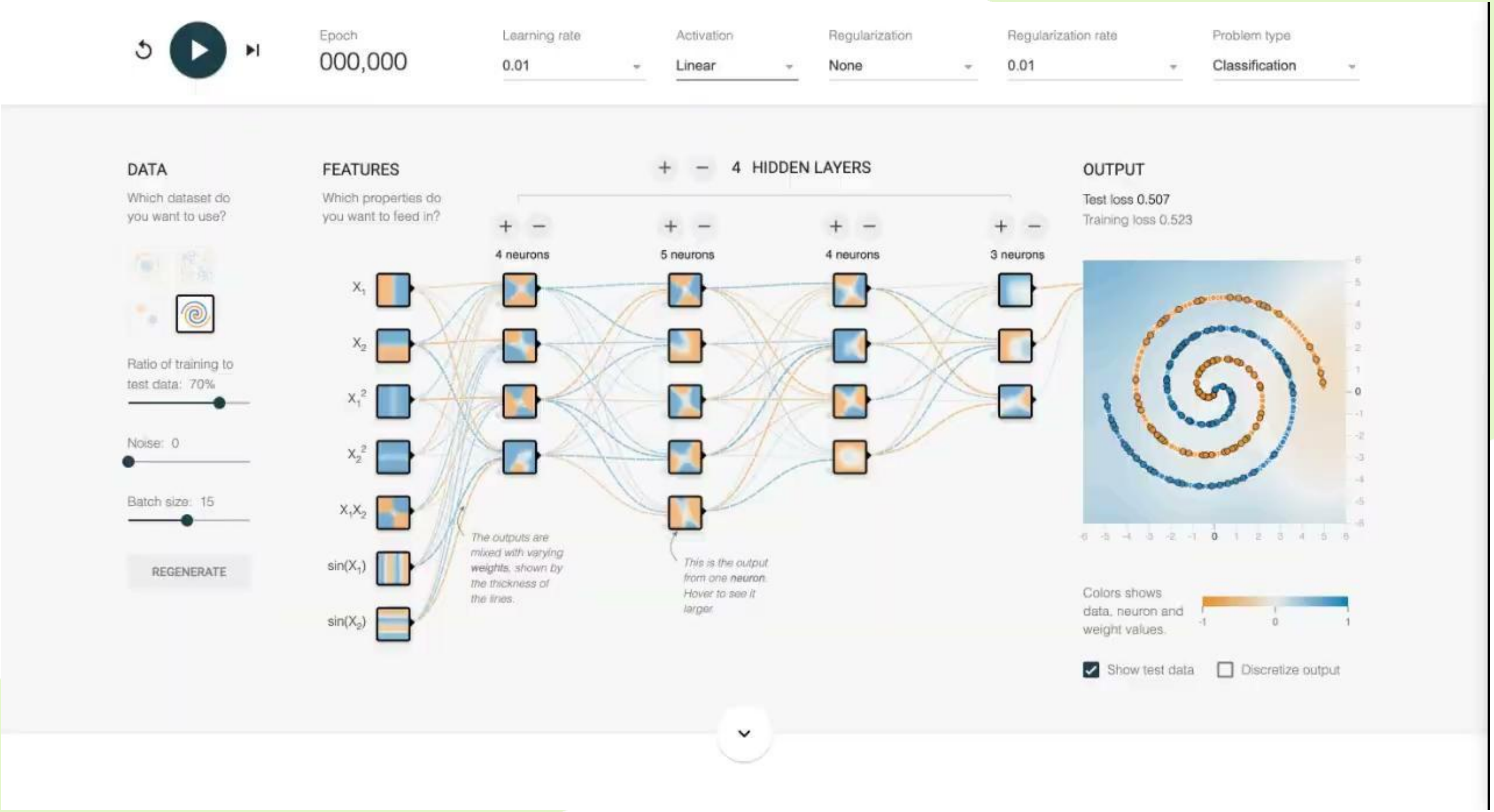
Credit: Aggarwal

$$ReLU(x) = max(0, x)$$

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

Vanishing gradient problem

Activation function



Data can be obviously impossible to separate linearly

doitnow
HPC Services

Linear activation function

Data could be easy to separate sometimes...

VS

Model: “No problem!”

doitnow HPC Services

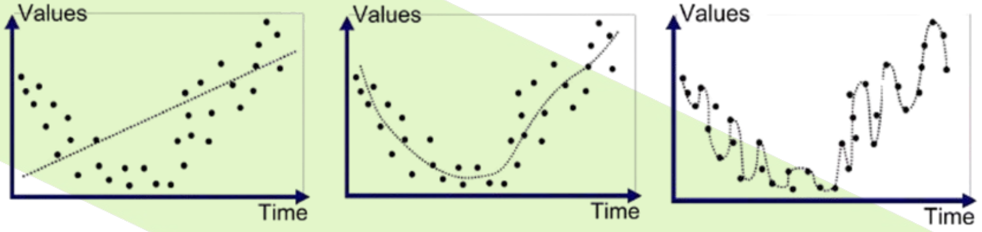
The problem

Things are bit more complex though...

← Model

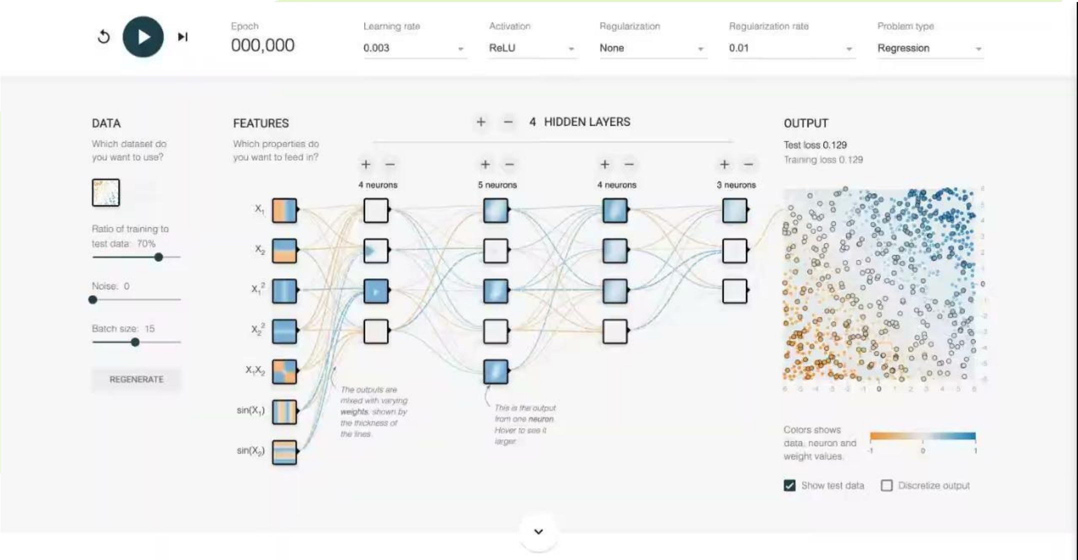
doitnow
HPC Services

Regularization



Underfitted | Good Fit/Robust | Overfitted

- Methods to counter for overfitting
  - Perfect fit on training data causes poor fit on test data
- Dropout: Drops the learning rate when static for an amount of epochs
  - Also good for barren plateaus
- L1 - Lasso regression
  - + absolute magnitude of feature coefficient
- L2 - Ridge regression
  - + square magnitude of feature coefficient



doitnow HPC Services

Backpropagation

$$\omega_{ij} = \text{Weight from neuron i to j}$$
$$\lambda = \text{Learning rate}$$
$$e_i = \text{Error at neuron i}$$
$$A_j(x) = \text{Activation function at neuron j}$$
$$E = l(a_{out}) = \text{Output error / cost}$$

Error after forward pass used to adjust weights in model backwards

Steps:

$$a_j = \sum_{\omega_{kj} \forall k} \cdot \omega_{ij} \cdot A_j(o_i)$$

1) Recursively assign neuron errors backwards to previous layers via weights

$$e_i = a_i \left(1 - a_i\right) \omega_{ij} \cdot e_j \; ; \; e_{out} = a_{out} \left(1 - a_{out}\right) E$$

2) Recursively calculate weight error gradient via neuron errors

$$\omega_{ij}^{E} = \frac{\partial E}{\partial \omega_{ij}} = e_j \cdot \lambda \cdot A_j(o_i)$$

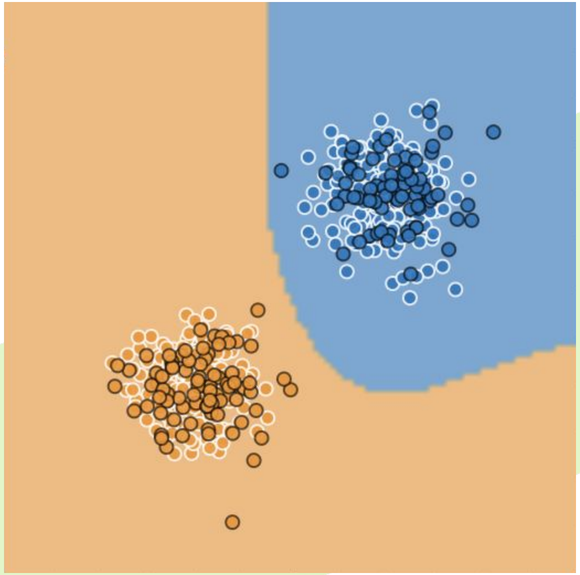
3) Combine weights with weight error gradients

$$\omega^{new} = \omega + \omega_{ij}^{E}$$

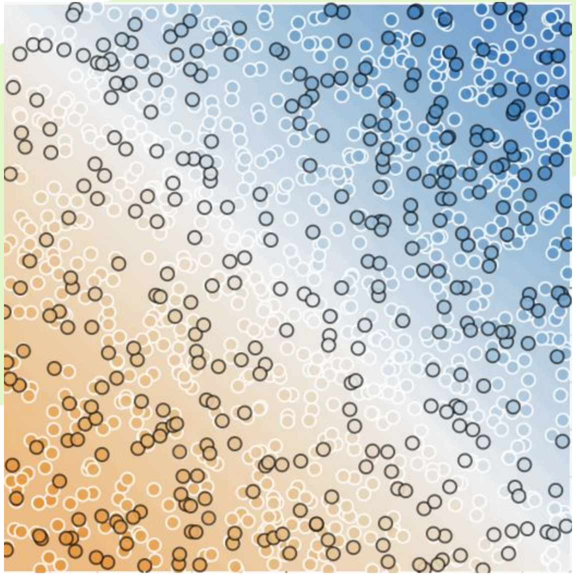
Classification vs Regression

Discrete modelling vs Continuous modelling

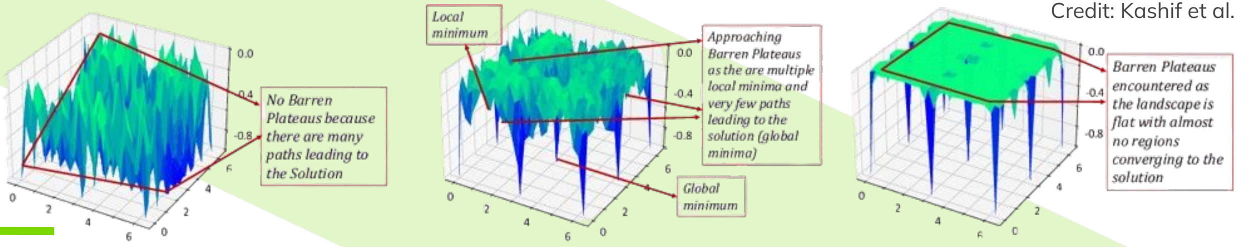
“Which color is this?” vs “what value is this?”



vs

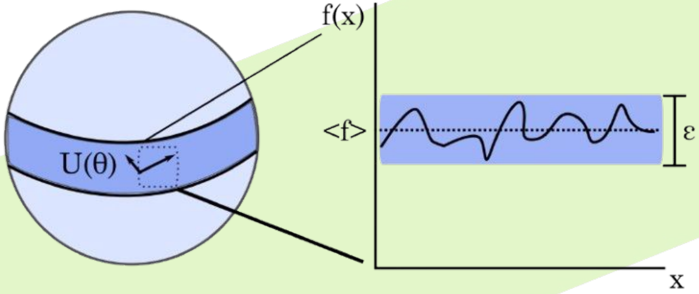


Credit: Kashif et al.



Barren plateaus

- A space where gradient descent-like approaches struggle
- Particularly a problem in quantum, as McClean et al. pointed out ([2018](#))
  - Variance in VQCs exponentially decrease to 0 as the qubits and/or layers increase
  - Make claim due to 2-design of random circuits
- Random quantum circuits are approx. 2-designs - [Harrow and Low (2009)](#)
  - Random gates over time yield a (Haar-distributed) unitary.
  - A 2-design has its variance equal to the Haar distribution.



Credit: McClean et al.

“If you have a barren plateau, all hope of quantum speedup or quantum advantage is lost”

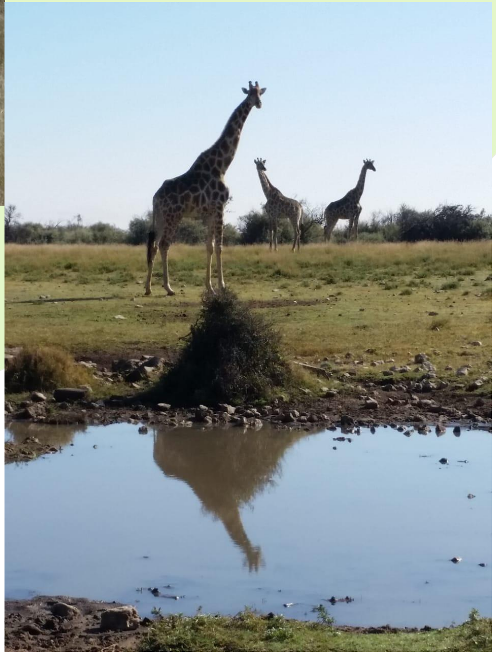
- Marco Cerezo (Los Alamos, 2021)

Barren plateaus

Example:

Find the (best) watering hole

A long neck helps...



- Roam around aimlessly and patiently
- Awareness without impulsiveness for descents
- Use the right tools:
  - Improved initialization → random init. in PQCs
  - Adaptive learning rates → ADAM optimizer
  - Regularization → entanglement regularization
  - Random walks → quantum walk search

Barren plateaus (BPs)

Reviews and surveys about BPs in VQCs: 2024, 2025

Techniques to reduce BPs:

- Shallow circuits
- Iteratively change circuit ← Variable structured QNNs
- Alternative initialization strategies
    - Classical pre-training, structured random init., parameter transfer, initialization ansatz

Techniques that are false promises:

- Changing optimizer ← can still require exponentially many shots
- Error mitigation in noise-induced BPs ← exponentially many resources

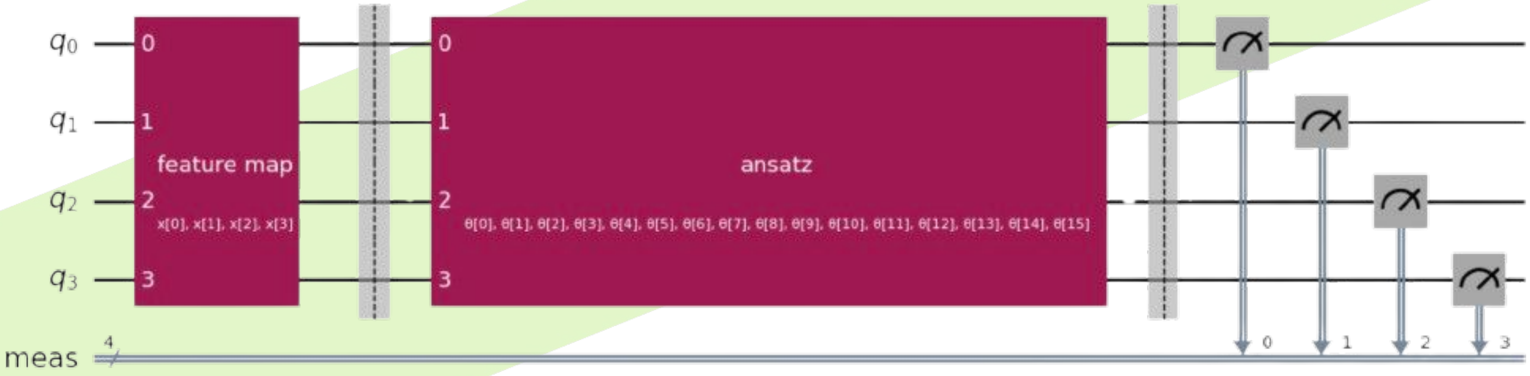
Quantum Neural Networks

Parameterized Quantum Circuits (PQCs)

Popular structure is VQAs, which is only what we will consider

Other structures include:

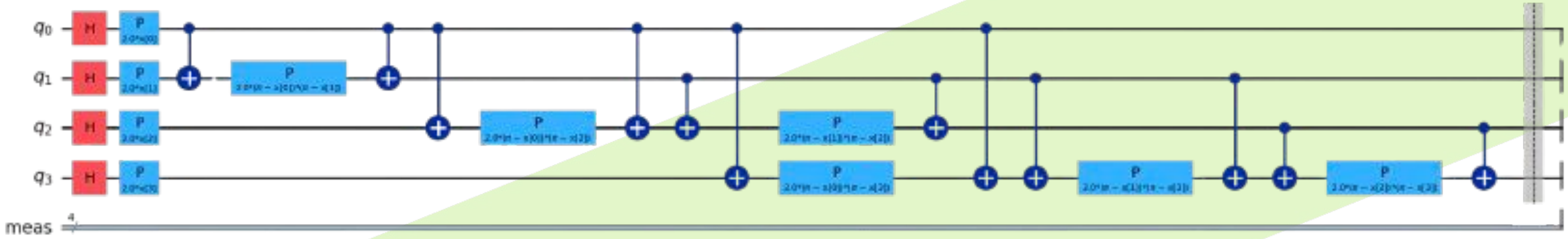
- [Continuous-Variable architecture (CV)]
- [Repeat-Until-Success (RUS)]

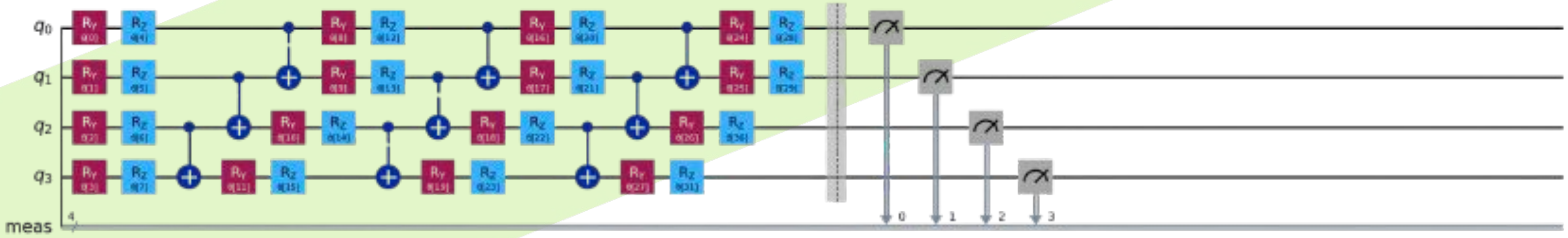


doitnow
HPC Services

QNN changes to NN

- Entanglement layers ~ Activation function
- [Backpropagation via traces]
    - [Efficient quantum backpropagation idea (2023)]
- Optimization still classical
- Inherent noise more than in classical
- Epochs technically more due to measurements & shots

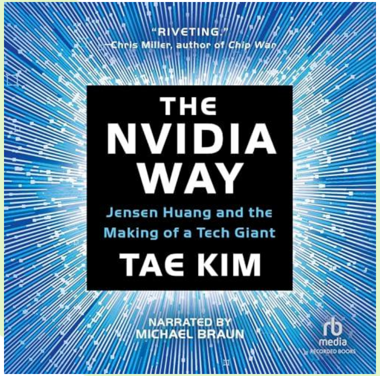




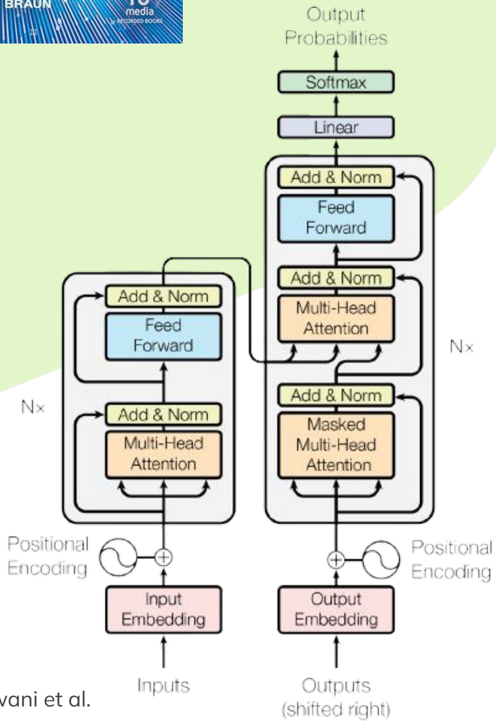
doitnow
HPC Services

Evolution of neural networks

- 1958: Rosenblatt invents the Perceptron
  - Limited to linear separability
- 1969: Minsky & Papert love Perceptrons
  - Basically invent parallelism, but also cause ‘AI Winter’
- 80s-90s: New techniques & models spur NNs again
  - Multilayer Perceptrons (MLPs), backpropagation, regularization
- 1990s: Nvidia GPUs perfect for neural networks
- 2010s: NN baby boom → DNNs, CNNs, RNNs, GANs…
- 2017: Google develops the Transformer
  - Self-attention mechanism allows dynamic focus of most relevant components.
- 2010s: Ethics and explainability of NNs (in AI) explored
- 2020s: ChatGPTs causes ‘AI Summer’



Great audiobook!



Credit: Vaswani et al.

doitnow
HPC Services

What's on the QNN menu?