Quantum mechanics with neural networks

Dane Hudson Smith

dhudsmith@gmail.com
github.com/dhudsmith
linkedin.com/in/dhudsmith

"If quantum mechanics hasn't profoundly shocked you, you haven't understood it yet."

—Niels Bohr

Bird's-eye view

Goal: Teach a neural network (NN) to rapidly solve quantum mechanics (QM) problems.

Motivation:

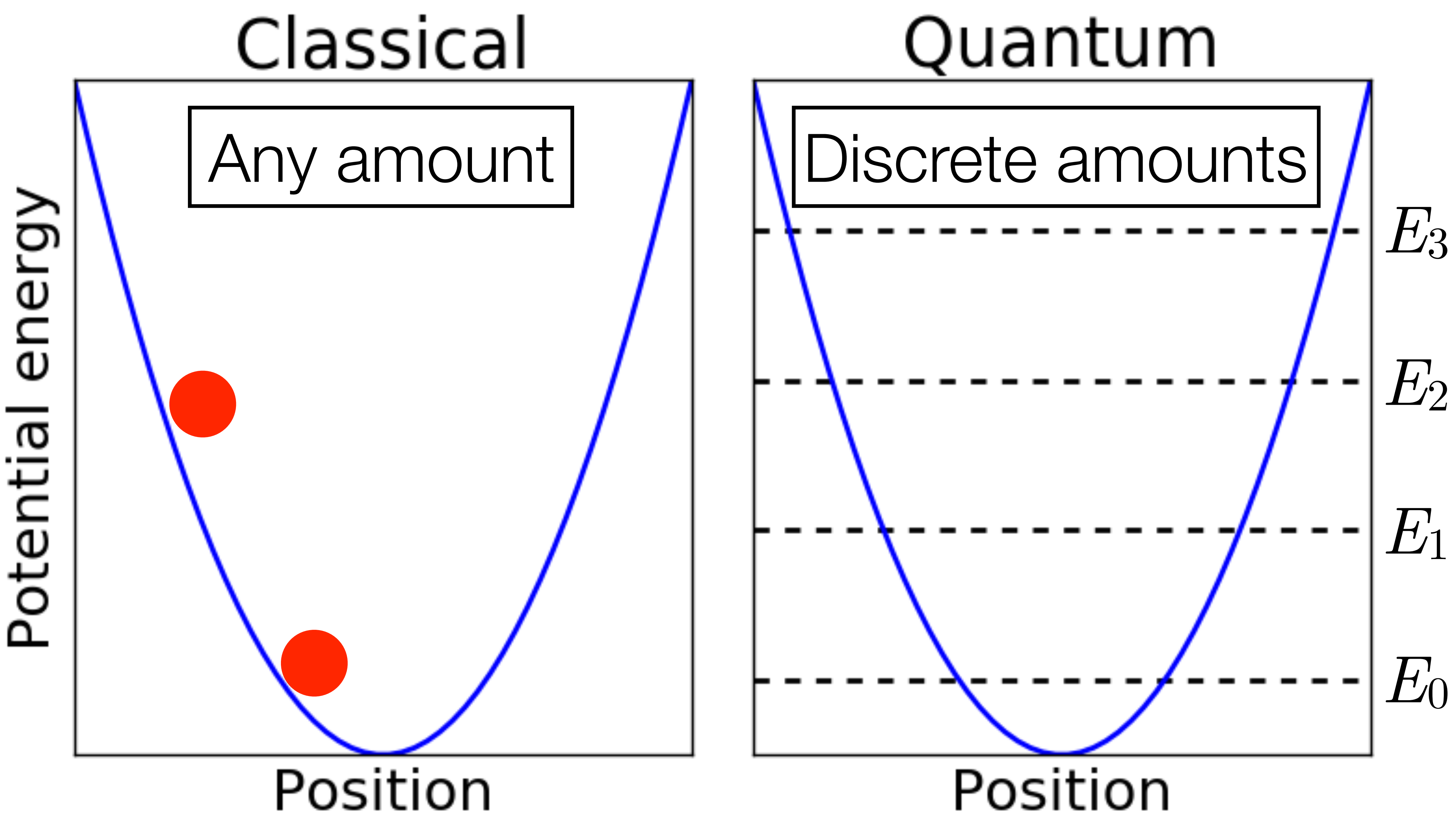
- Solving QM problems is computationally expensive
- Solution may be impractical in time-sensitive applications
- Once trained, simple NNs make quick predictions

Approach:

1. Solve a large ensemble of QM problems by brute force
2. Train a neural network (NN) to predict solution for new inputs

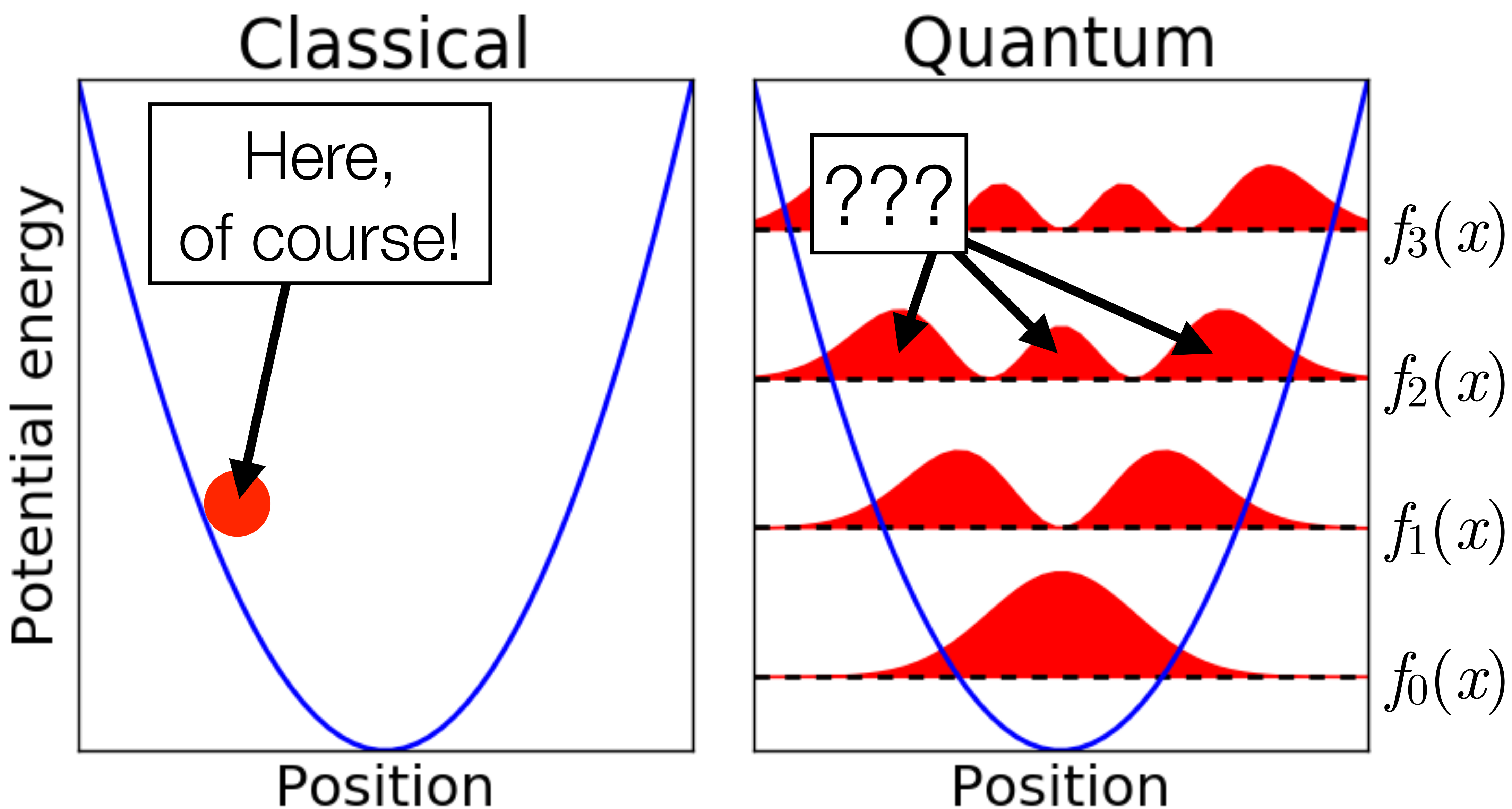
Crash course in quantum mechanics

How much energy can we give a particle in a bowl?



Crash course in quantum mechanics

Where is the particle?



Crash course in quantum mechanics

How do we determine the allowed **energies** and the **probability densities**?

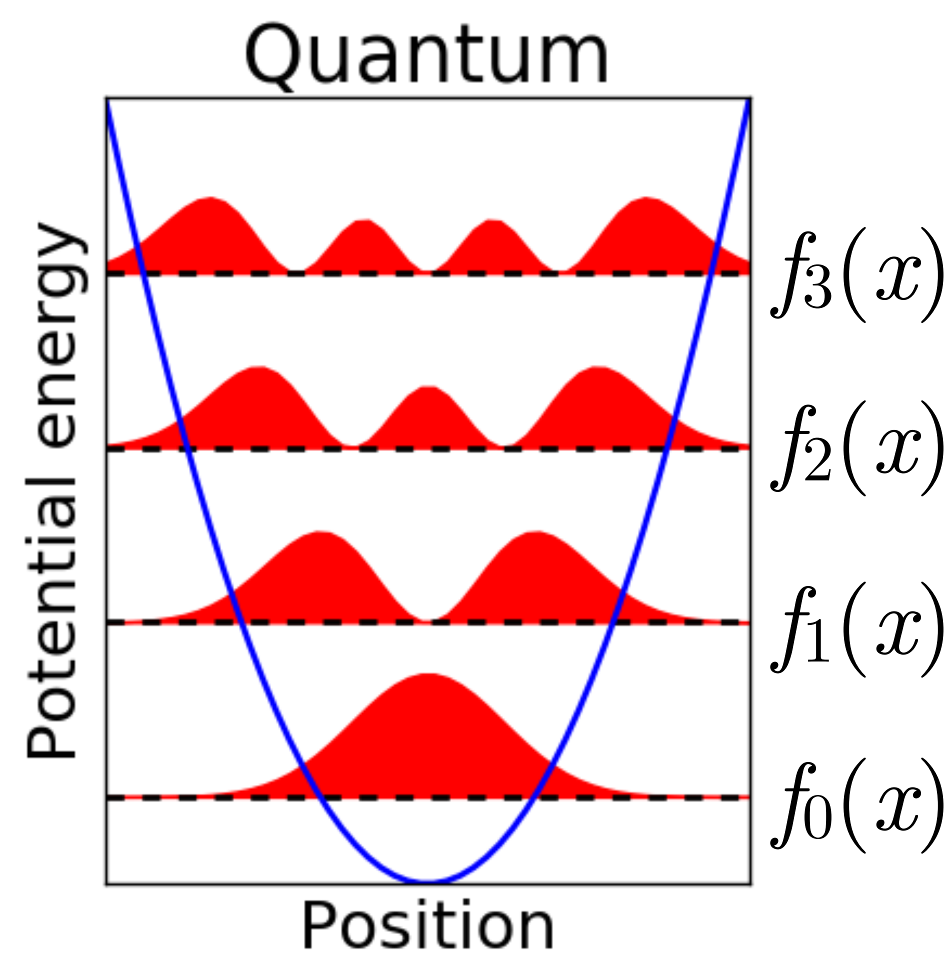
Solve Schrödinger's equation:

$$\left[-\frac{1}{2m}\frac{d^2}{dx^2}+V(x)\right]\psi(x)=E\psi(x)$$

Potential energy

"Wave function"

Energy



What is a "wave function"?

$$f(x)=|\psi(x)|^2$$

Crash course in quantum mechanics

Solving Schrödinger's equation for new $V(x)$ is computationally expensive and time consuming.

Define the *functionals:*

$$g_E\left(V\left(x\right)\right) = E$$ (**energy** functional)

$$g_f\left(V\left(x\right)\right) = f\left(x\right)$$ (**probability density** functional)

If we knew the form of $g_E(V(x))$ and $g_f(V(x))$, we would not have to solve Schrödinger's equation for each new $V(x)$.

Neural network model of a quantum system:
Big picture

Goal:

Model the functionals $g_E(V(x))$ and $g_f(V(x))$ with neural networks

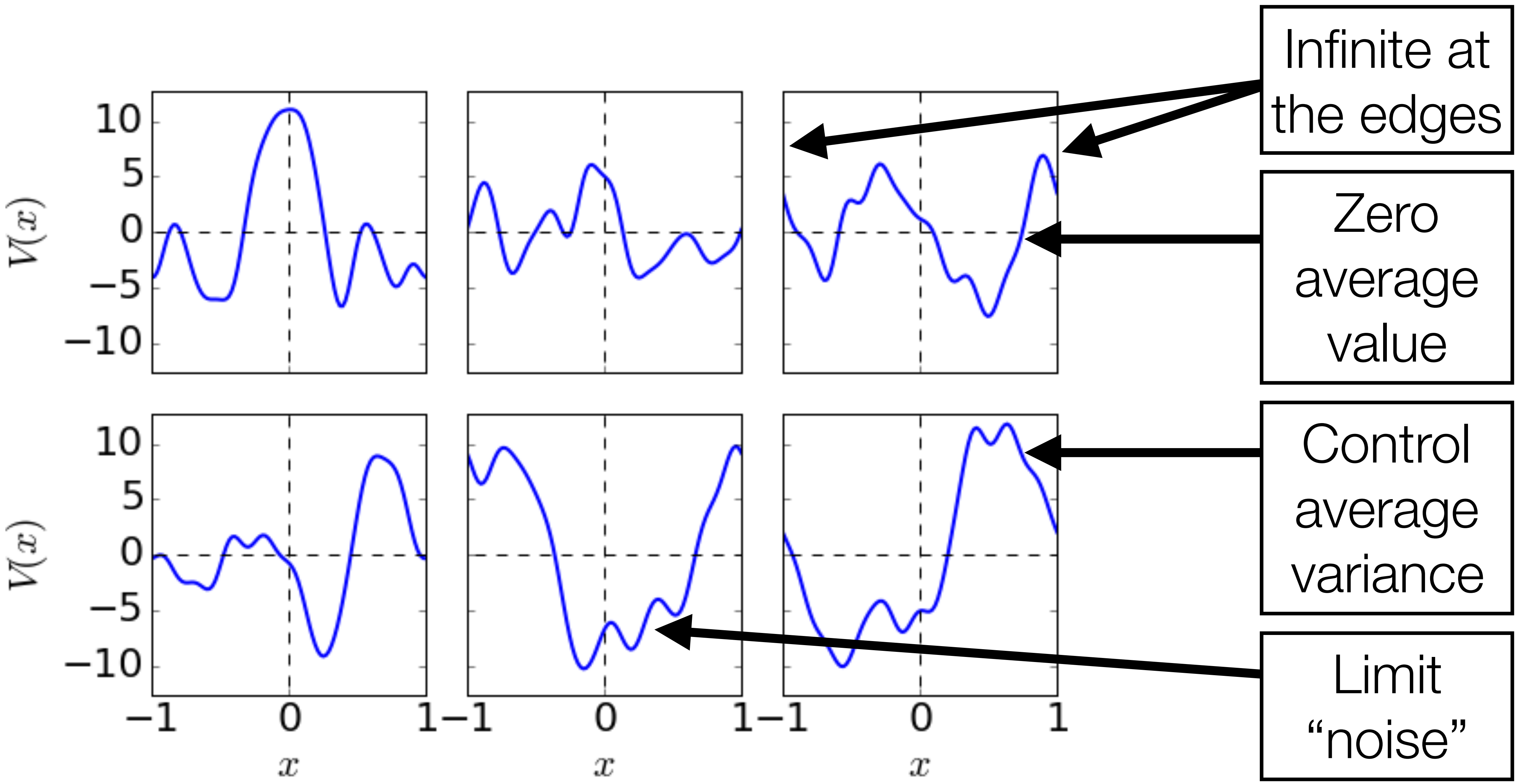
Why neural networks?

- $g_E(V(x))$ and $g_f(V(x))$ highly nonlinear
- 1D "image processing"
- Fast predictors

Neural network model of a quantum system:

Building the neural network

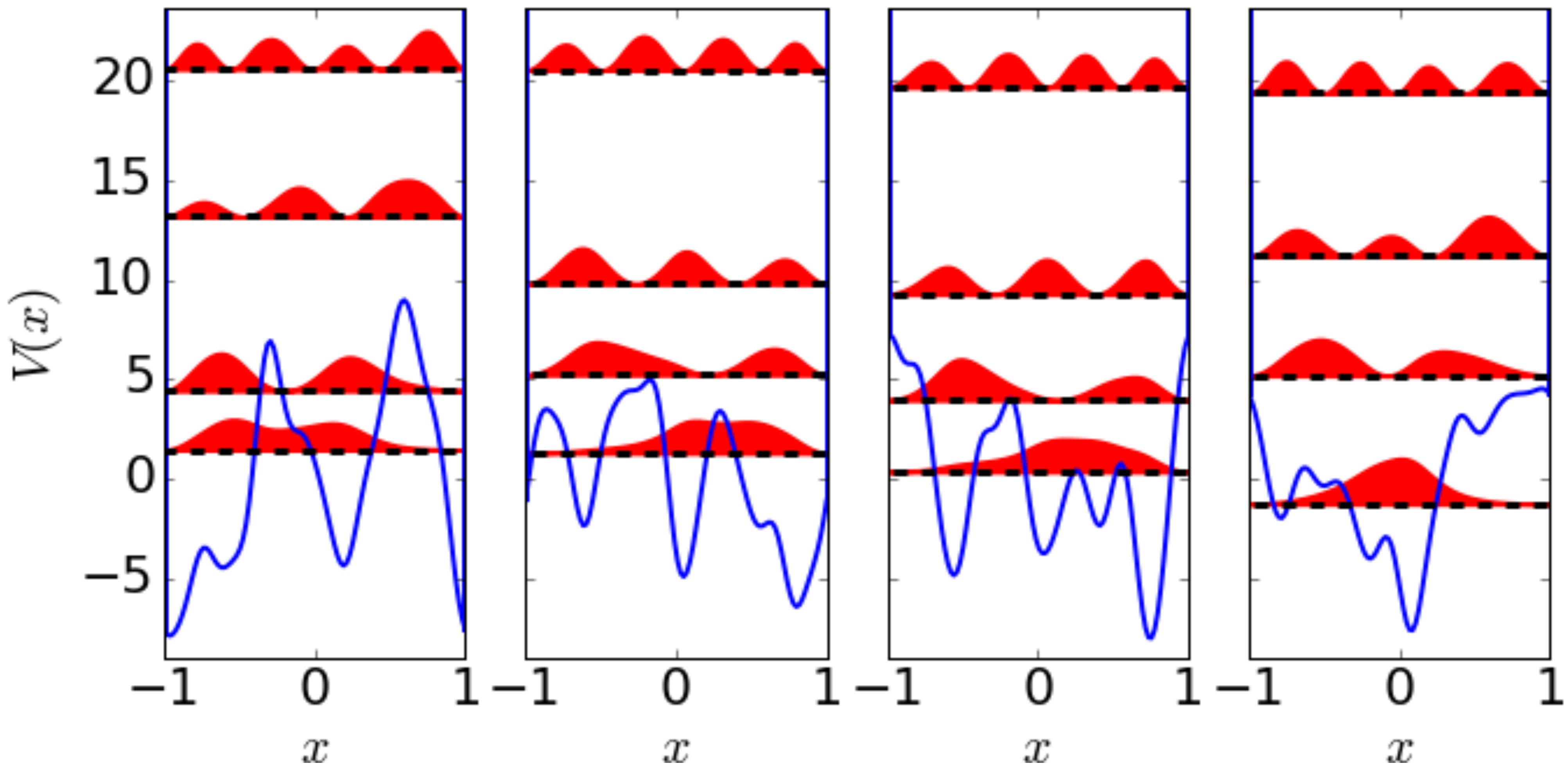
Generate an **ensemble** of random potentials:



Neural network model of a quantum system:

Building the neural network

Solve Schrödinger's equation for the entire **ensemble**



Extract **probability distributions** and **energy levels**.

Neural network model of a quantum system

Building the neural network

Generate a ensemble of examples of $g_f(V(x))$ and $g_E(V(x))$:

$V(x)$	$f_0(x)$	$f_1(x)$	...	E_0	E_1	...
$V^{(1)}(x)$	$f_0^{(1)}(x)$	$f_1^{(1)}(x)$	...	$E_0^{(1)}$	$E_1^{(1)}$	...
$V^{(2)}(x)$	$f_0^{(2)}(x)$	$f_1^{(2)}(x)$	...	$E_0^{(2)}$	$E_1^{(2)}$	...
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$	...
$V^{(N)}(x)$	$f_0^{(N)}(x)$	$f_1^{(N)}(x)$	...	$E_0^{(N)}$	$E_1^{(N)}$	...

$g_f(V(x))$ $\qquad$ $g_E(V(x))$

Use a neural network to model of $g_f(V(x))$ and $g_E(V(x))$.

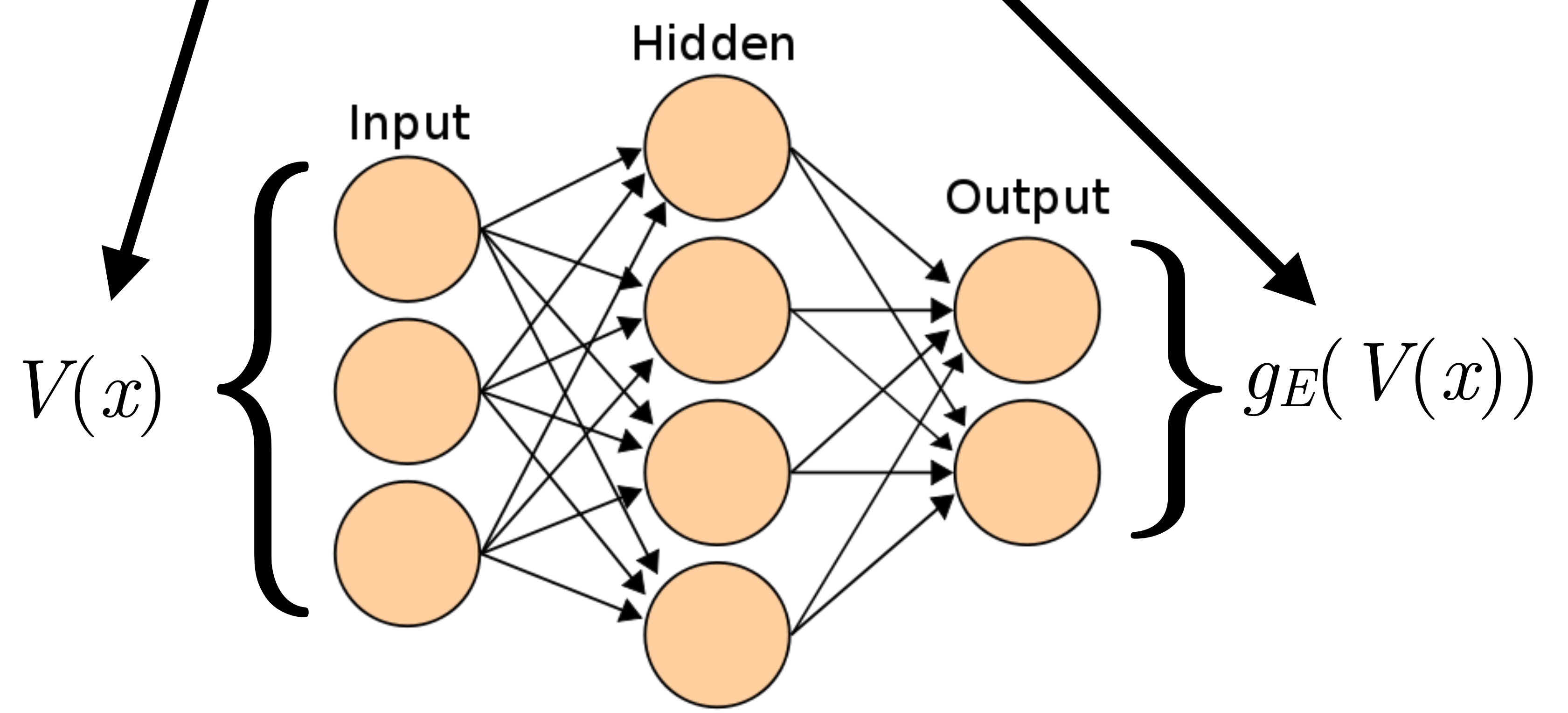
Neural network model of a quantum system

Building the neural network

Focus on $g_E(\,V(x))$:

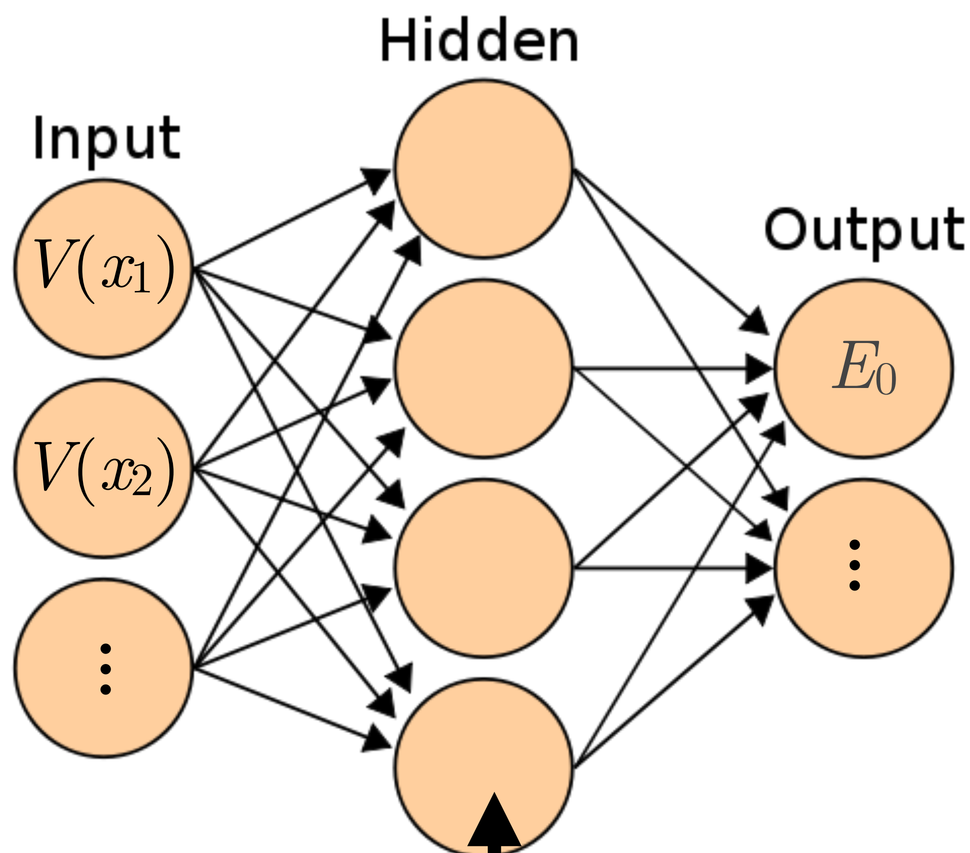
$V(x)$	E_0	E_1	...	E_9
$V^{(1)}(x)$	${E_0}^{(1)}$	${E_1}^{(1)}$	...	${E_9}^{(1)}$
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
$V^{(1e5)}$	${E_0}^{(1e5)}$	${E_1}^{(1e5)}$	...	${E_9}^{(1e5)}$

The "Multi-layer perceptron" model:



Neural network model of a quantum system

Building the neural network



Input:

$\{V(x_1),\ V(x_2),\ \ldots,\ V(x_{100})\}$

Output:

$\{E_0,\ E_1,\ \ldots,\ E_9\}$

Need to choose:

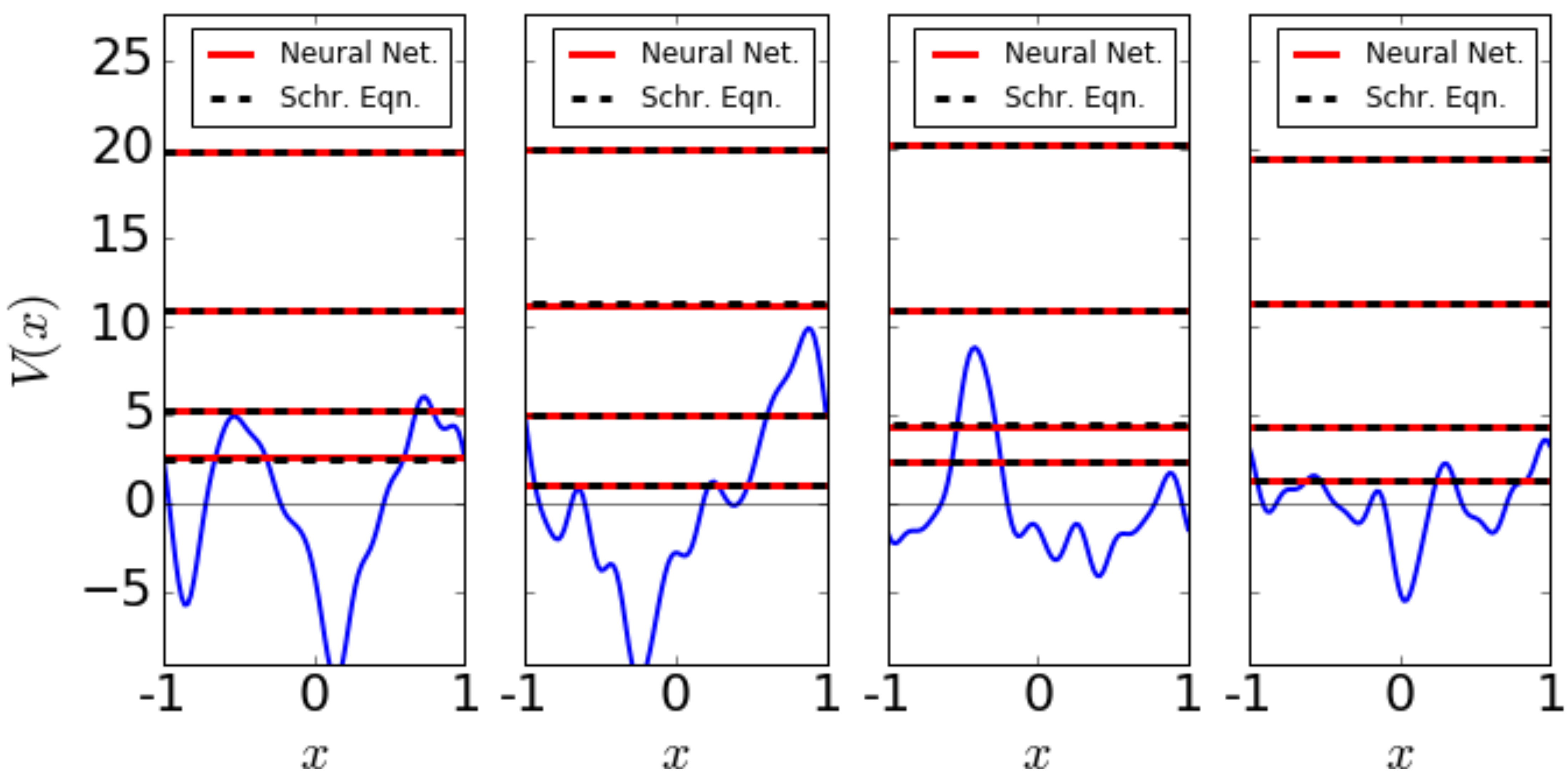
- **number of hidden layers**
- **number of nodes** per layer

Neural network model of a quantum system

Results

Hidden layers: 1

Nodes in hidden layer: 50

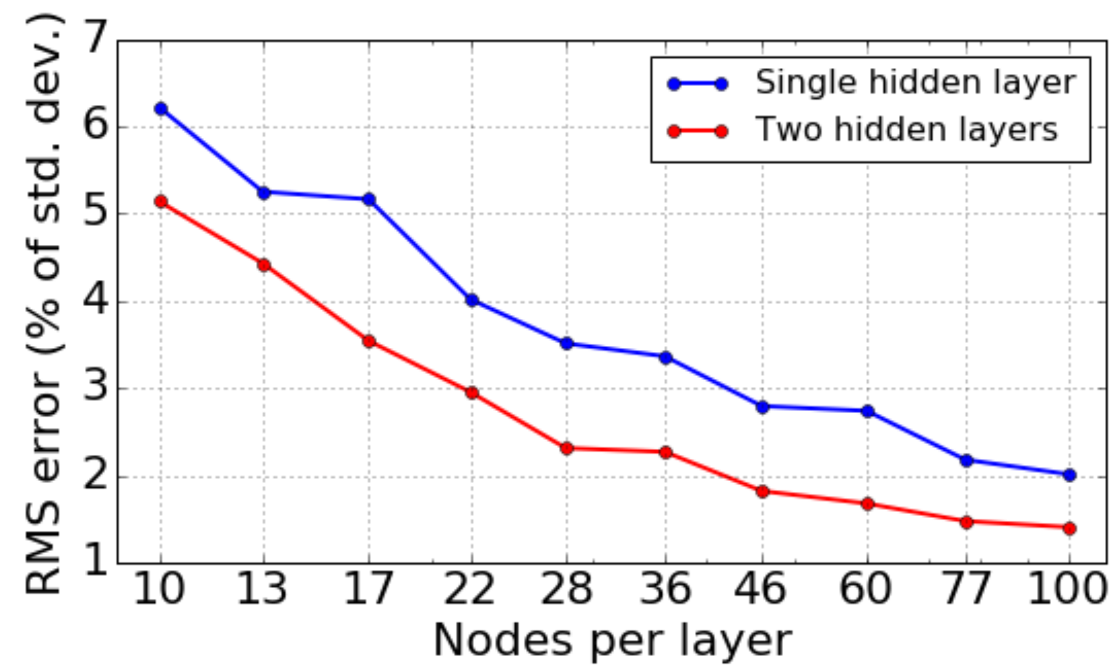


Neural network model of a quantum system

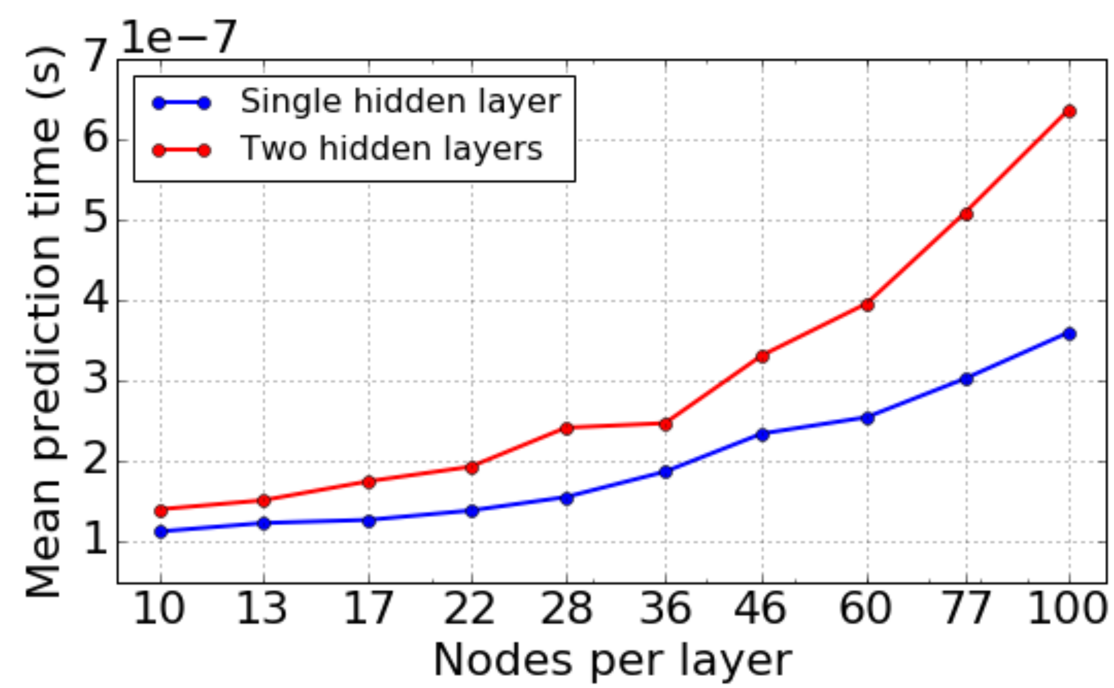
Results

How do **accuracy** and **speed** depend upon the network architecture?

- NNs with 2 hidden layers are more accurate, slower.
- NNs with more nodes are more accurate, slower.



Fast: compare with ~1e-5 s from conventional solution.

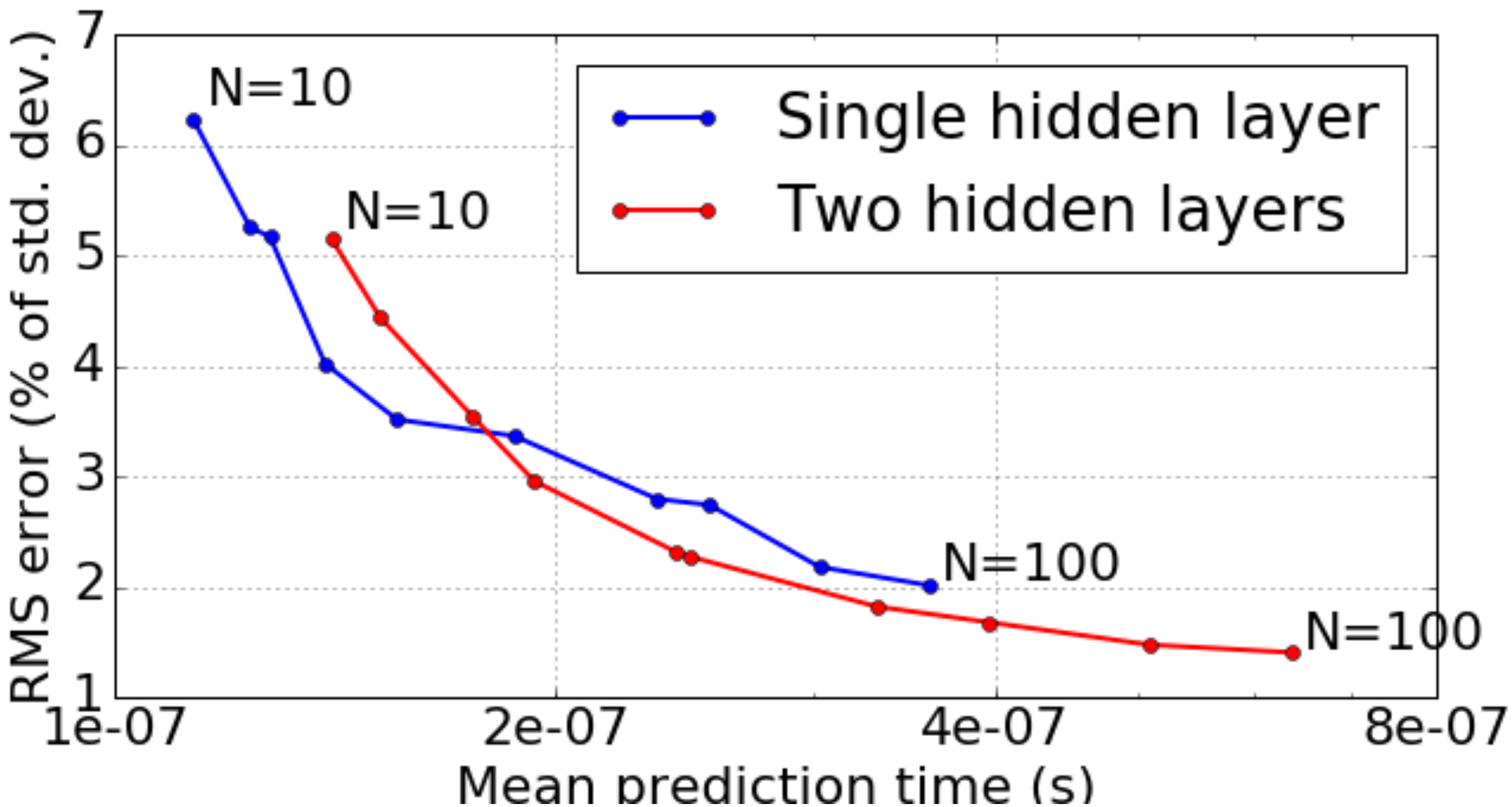


Neural network model of a quantum system

Results

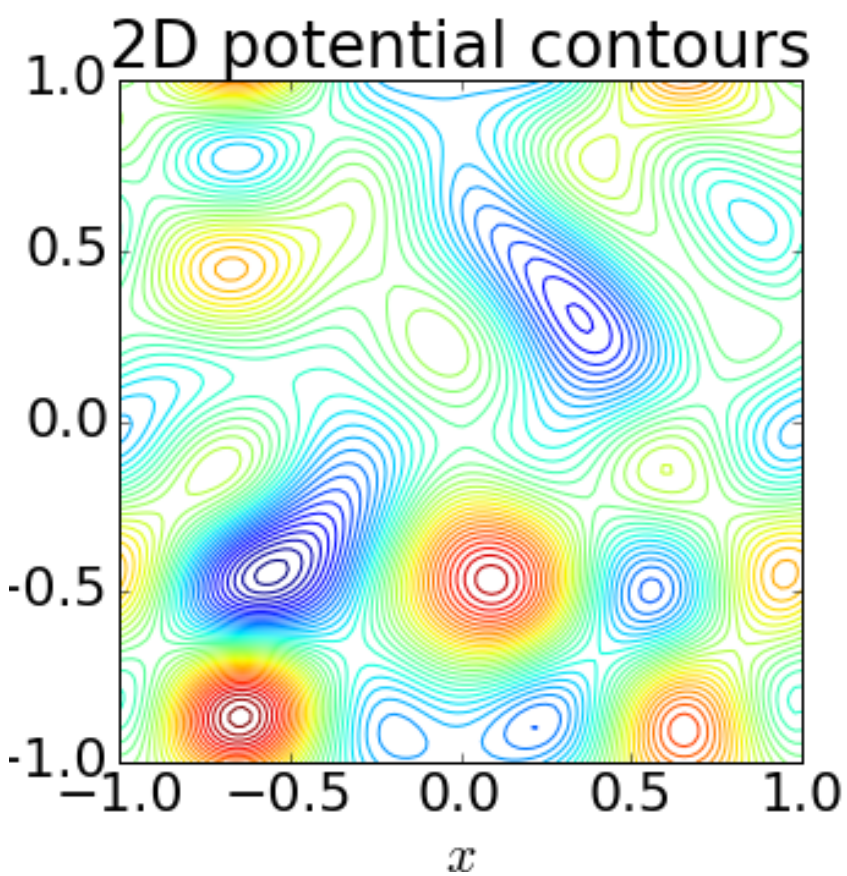
How do we choose an “optimum” network architecture?

Trade-off between prediction **accuracy** and prediction **speed**



To do

1. Model the probability density functional, $g_f(V(x))$
2. Publish!
3. Try other nonlinear models
4. Model 2D and 3D quantum systems



Summary

- Neural networks can accurately model quantum mechanical systems
- Prediction times are much faster than conventional solutions
- Model selection achieved by optimizing prediction time and accuracy

Thank you!

References

- **Source code:**
  Github
  https://github.com/dhudsmith/QuantumML
- **Neural network:**
  Scikit-learn, development version
  http://scikit-learn.org/dev/documentation.html
- **Plots:**
  Matplotlib
  http://matplotlib.org/contents.html
- **Multilayer perceptron graph:**
  https://github.com/nikolaypavlov/MLPNeuralNet