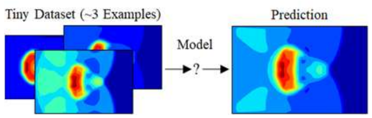
Fast Neural Network Predictions from Constrained Aerodynamics Datasets

Tina White crwhite@stanford.edu

Background

For engineers and scientists running finite element or computational fluid dynamics simulations, one simulation may take hours, days, or weeks. They can't collect as many data points as they want, and what they want is to fully explore some parameter space. What if they could get accurate estimates in minutes or seconds, given the information they already have from the simulations they have already run?

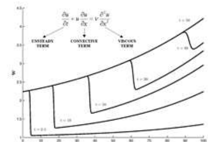


How do you make good predictions based on limited experience?

Two Test Cases

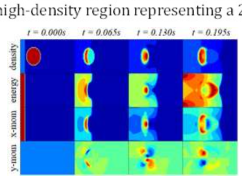
The Burgers' equation test case is commonly used for evaluating ROMs. It is a 1D application of an initial-boundary-value problem, which models a shockwave moving across a tube.

Burgers' Equation



The shock bubble test case is a 2D application of the compressible Navier-Stokes equations, modeling a shockwave moving across a high-density region representing a 2D bubble.

Shock Bubble



The Baseline - ROMs

Reduced order models (ROMs) are the state-of-the-art method that addresses this problem in industry. ROMs solve the original governing equations after projecting them into a smaller state space, called a basis, reducing run time. ROMs use knowledge gained from previous simulations to construct the basis vectors.

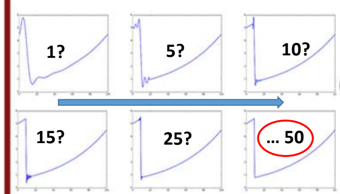
But ROMs have issues:

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

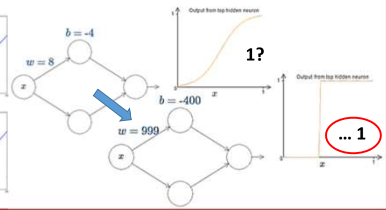
1. Hard coded. Intrusive.
2. You have to know the model, the governing equations.
3. For highly nonlinear problems, run time may not improve.

Motivation for Neural Networks

ROMs and Discontinuities
How many *basis vectors* does it take to capture a discontinuity in Burgers' equation?



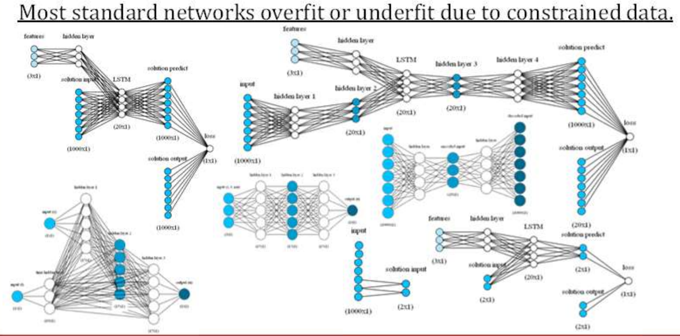
NNs and Discontinuities
How many *artificial neurons* does it take to capture a discontinuity in a function?



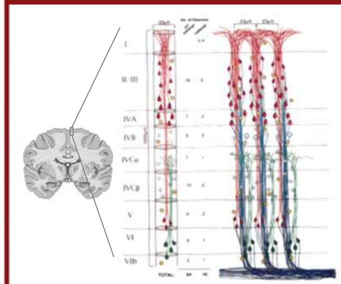
Standard Networks Don't Work

Most modern deep learning methods only work well in big data regimes and learning from few examples is an area of active research. Different configurations for LSTM networks, variational autoencoders, fully connected, and message passing networks were tested as well as strategies for regularization like dropout. Several types of nonlinearities and loss functions were also explored.

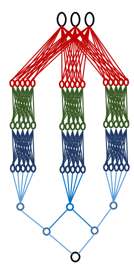
Most standard networks overfit or underfit due to constrained data.



Inspiration from the Cortex

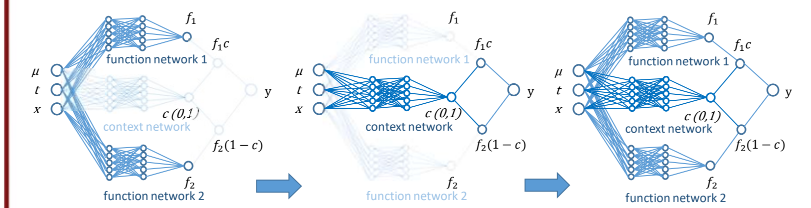


Unlike modern deep learning methods, which use a single monolithic network, the human cortex seems to be segmented into narrow chains of nearly independent networks containing relatively few neurons each (approximately 10,000). In 1978, Mountcastle described the cortex as a collection of these structures, which he called cortical columns.



A New Kind of Network: Cluster Networks

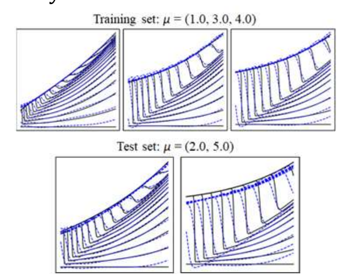
The cluster network architecture is a simple feed-forward network, except with distinct connected clusters. The network performs a mapping from (x,t,μ) tuples to (y) where x represents space, t represents time, and y represents the output of the network, which approximates the function to be learned, in this case, velocity w or u. Also, μ represents a hyperparameter, which refers to any parameter in the design or governing equation than can change from one simulation to the next. The *context* networks determine how much to turn the *function* networks on or off.



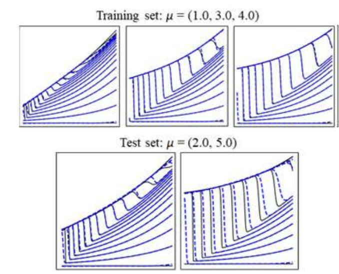
Results and Conclusions

To predict Burgers equation results, the model was trained on μ = 1,3,4 and tested on μ = 2,5. It interpolates and extrapolates well, while running faster than state-of-the-art ROMs.

Fully Connected Network Results



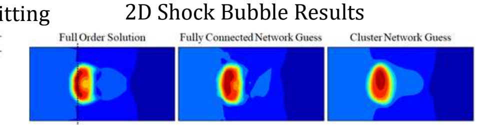
Cluster Network Results

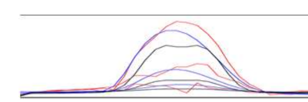


Comparison of NNs, ROMs, Fitting

Method	CPU (s)	% Error	QoI % Error
FOM	16.273		
ROM 100	12.981	0.07	0.05
ROM 50	5.690	0.20	0.16
ROM 20	3.575	0.86	0.60
ROM 5	2.415	4.04	3.00
ROM 3	2.119	5.47	5.28
FCN 4 20	0.245	2.43	3.04
ClusterNet 2 3 5	0.227	0.71	0.42
QoI Poly Fit 3	<0.001		7.28
QoI Poly Fit 2	<0.001		7.15
QoI Poly Fit 1	<0.001		8.42
QoI Poly Fit 0	<0.001		15.03

2D Shock Bubble Results





Humans solve and simplify problems by breaking them into parts. The cluster network is designed explicitly for that purpose. Each training point is assumed to fall into one context or another, or along a boundary in between. Thus, the Burgers' equation test case is well suited to this architecture because the problem *is* more easily solved by breaking it into parts – solutions in front of the shockwave and behind it can be thought of as two functions that behave almost independently, with a boundary (the shockwave) between them.