Stanford | ENGINEERING
Mechanical Engineering

Accelerating Fluid Simulations with Neural Networks

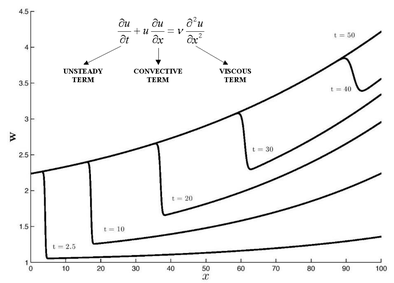
A neural network architecture for reduced order modeling of PDEs

Tina White crwhite@stanford.edu

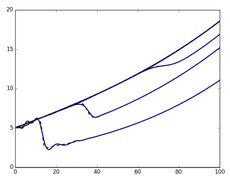
Data Type

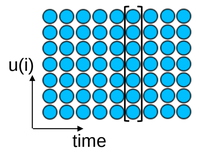
Burgers' Equation Test Case

A one-dimensional application of an initial-boundary-value problem that models the movement of a shockwave.



Model Order Reduction

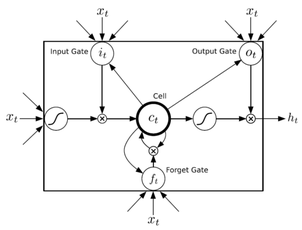




The current state of the art for this problem is a ROM (reduced order model). ROMs can show unphysical Gibbs' oscillations.

LSTMs

The Long Short Term Memory (LSTM unit)



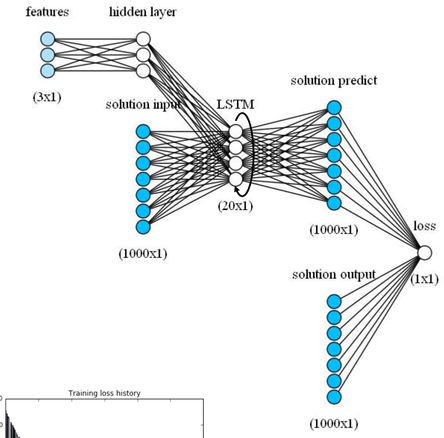
$$
\begin{aligned}
\mathbf{i}_t &= \sigma\left(W_{xi}\mathbf{x}_t + W_{hi}\mathbf{h}_{t-1} + W_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i\right), \\
\mathbf{f}_t &= \sigma\left(W_{xf}\mathbf{x}_t + W_{hf}\mathbf{h}_{t-1} + W_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f\right), \\
\mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t\tanh\left(W_{xc}\mathbf{x}_t + W_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c\right), \\
\mathbf{o}_t &= \sigma\left(W_{xo}\mathbf{x}_t + W_{ho}\mathbf{h}_{t-1} + W_{co}\mathbf{c}_t + \mathbf{b}_o\right), \\
\mathbf{h}_t &= \mathbf{o}_t\tanh(\mathbf{c}_t).
\end{aligned}
$$

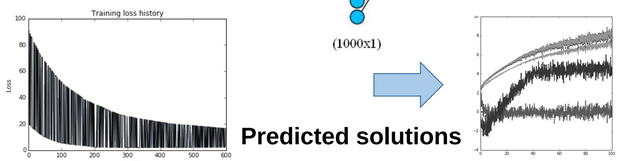
Each LSTM unit has a cell which has a state c_t at time t. The cell is like a memory unit. Access to the memory unit for reading or modifying is controlled through sigmoidal gates – the input gate i_t, forget gate f_t and output gate o_t.

Model Description

Part 1 – LSTM Network

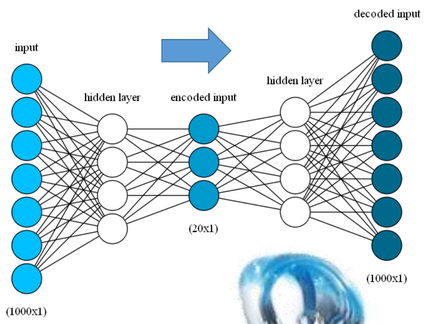
The model uses Recurrent Neural Nets (RNNs) made partially of LSTM units. The current implementation is analogous to an image captioning system with features instead of images, and sequences of input solutions instead of sequences of words. .





Predicted solutions

Part 2 – Autoencoder

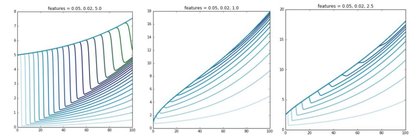


An autoencoder determines features from the solution inputs in order to compress the data prior to being fed into the LSTM network. Without the autoencoder, the full Burgers' solution is input into the LSTM network, and the run time of both the network training and testing is unnecessarily increased. With an autoencoder, the information can be highly compressed prior to being fed into the LSTM.

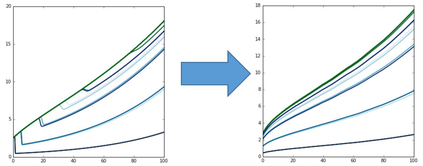
The autoencoder only outputs representations of the original inputs, and does not output new solutions in time. It only compresses the data.

Preliminary Results

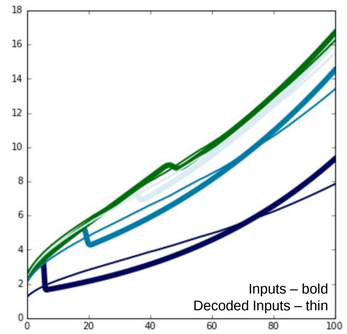
The decoded inputs from the autoencoder reveal why the LSTM network implementation is not converging – the neural network is unable to learn the locations of the discontinuities.



Input Variation



Sample Inputs **Sample Decoded Inputs**

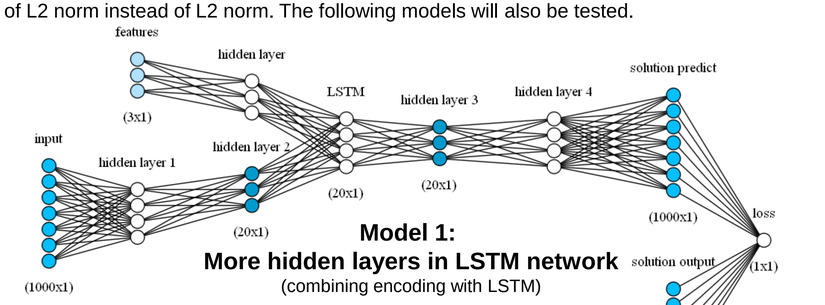


Sample Inputs and Decoded Inputs

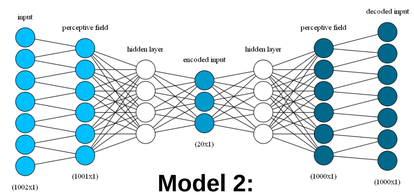
Alternative Models

The following modifications were attempted, but did not result in improvements to the preliminary results:

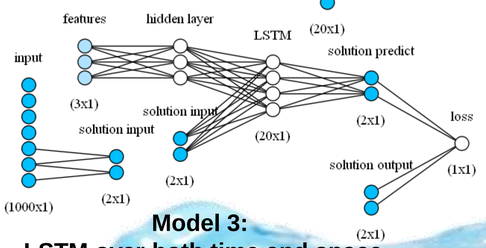
(1) regularization (2) dropout (2) sigmoid nonlinearity instead of relu and (4) loss function calculated with square of L2 norm instead of L2 norm. The following models will also be tested.



Model 1:
More hidden layers in LSTM network
(combining encoding with LSTM)



Model 2:
Perceptive field in autoencoder
(like a convolution but insensitive to spacing)



Model 3:
LSTM over both time and space
(memory may "remember" discontinuity locations)

A perceptive layer is proposed instead of a convolution because the input data in general will not be spatially invariant, and convolution is sensitive to spacing