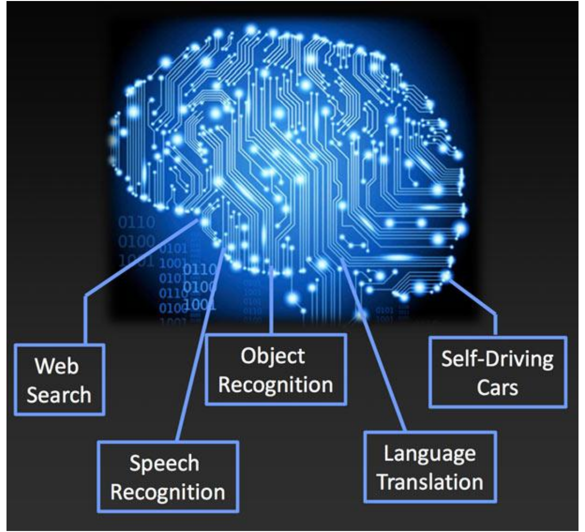
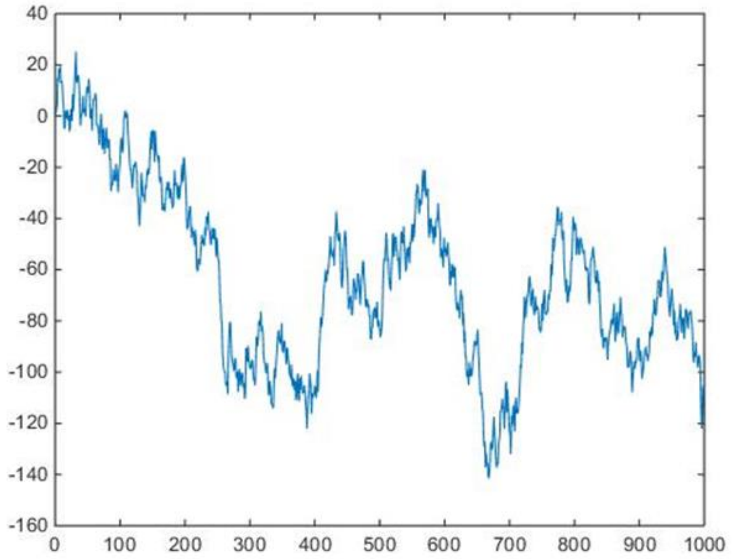


Neural Network and Deep Learning For Univariate Time-series Forecasting

Kim Minsam
Supervisor: Prof. James Kwok



OBJECTIVE

Assess univariate time-series forecasting capabilities of neural networks, and verify whether recently developed deep-learning techniques can improve result.

INTRODUCTION

Neural Networks: a family of statistical learning algorithms inspired by biological neural networks. (Wikipedia) Typical type of neural network called feed-forward neural network consists of input layer, hidden layer, output layer that processes an input to achieve a desired output, or the target. This procedure can be interpreted as a function approximation.

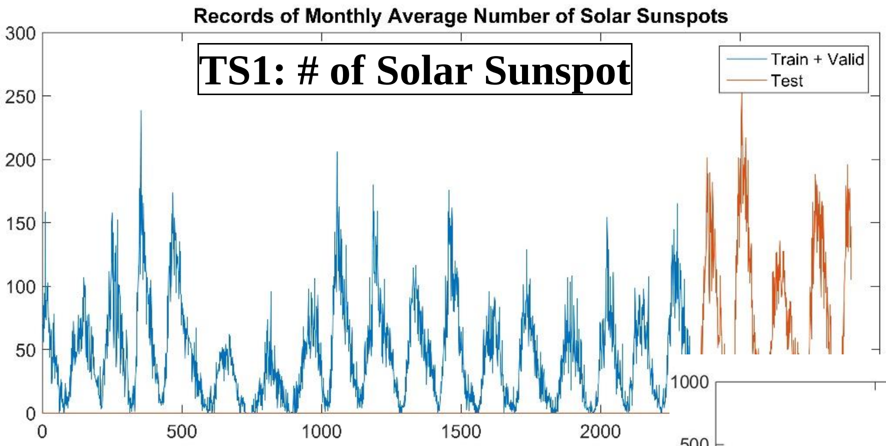
Auto-encoder: a type of neural network that tries to reconstruct the input with the output. Usually, the network has hidden layer working as an information bottleneck that naturally performs dimensionality reduction in the process of training reconstruction of inputs.

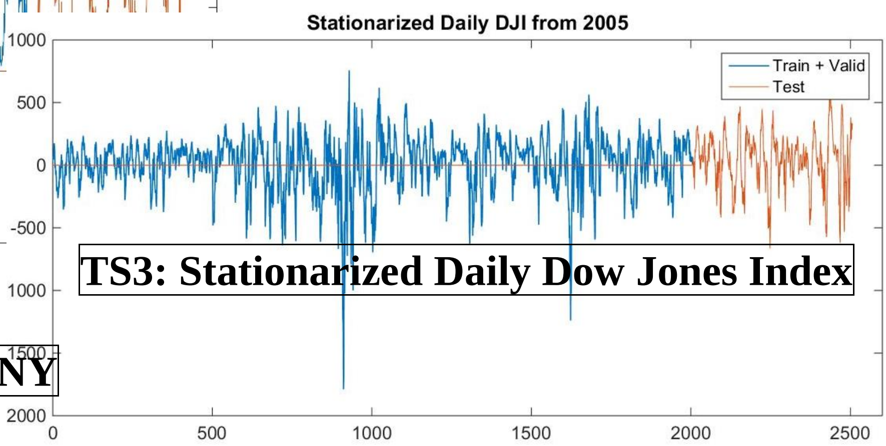
Deep Learning: a subset of neural networks that possess multiple hidden layers or recurrent connection. Although such structure enables the network to be more flexible and efficient to approximate complicated functions, the complexity makes the learning procedure very difficult. However, recent achievements from 2006 significantly improved the performances, thus became a hot topic in both industry and academia.

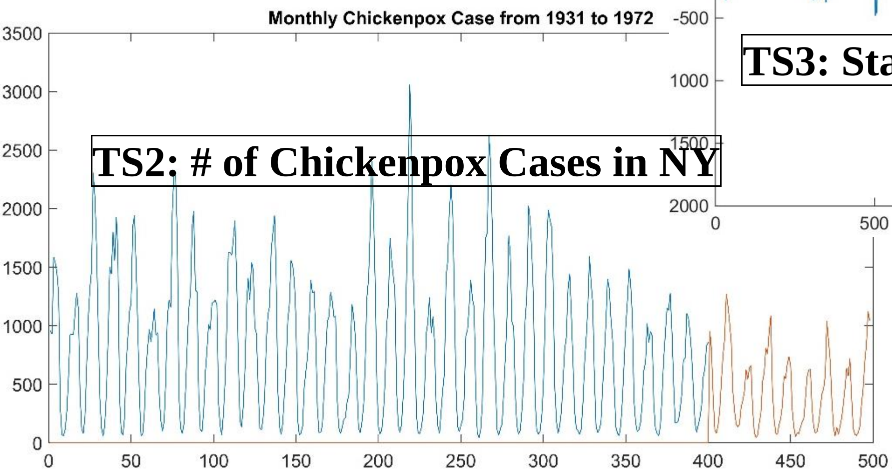
Time-series: A time series is a sequence of data points, typically consisting of successive measurements made over a time interval. Examples of time series are ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average. (Wikipedia)

THREE TIME SERIES DATA USED

(These are frequently used time-series data with meaningful structural differences.)





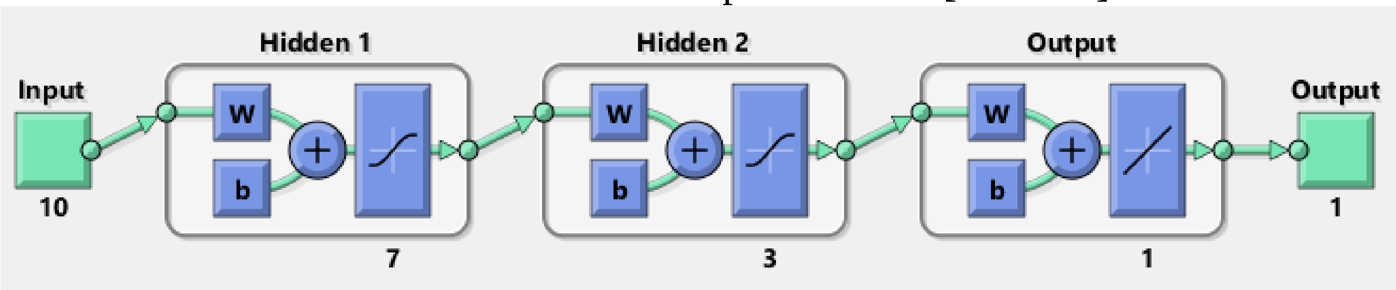


NEURAL NETWORK MODELS

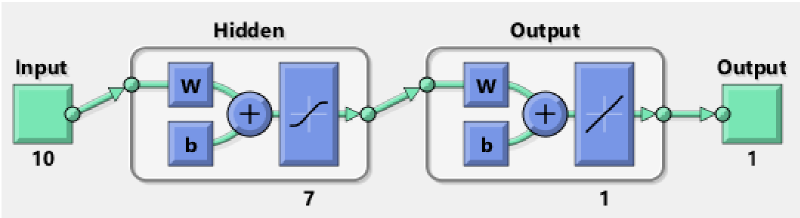
*(**Input:** Time-delay vector that contains past values of time-series data that correspond to the number of input neuron units in the network. **Output:** Forecasted value of the next time-index. Optimization was done with **scaled conjugate gradient**.)*

Simple Feedforward Neural Network (FNN)

Example of a feedforward neural network with 2 hidden layers with 7 and 3 neuron units in each. This forecasting network receives 10 previous time-series values to forecast the next value. Notation: the structure below is represented as [10 7 3 1].



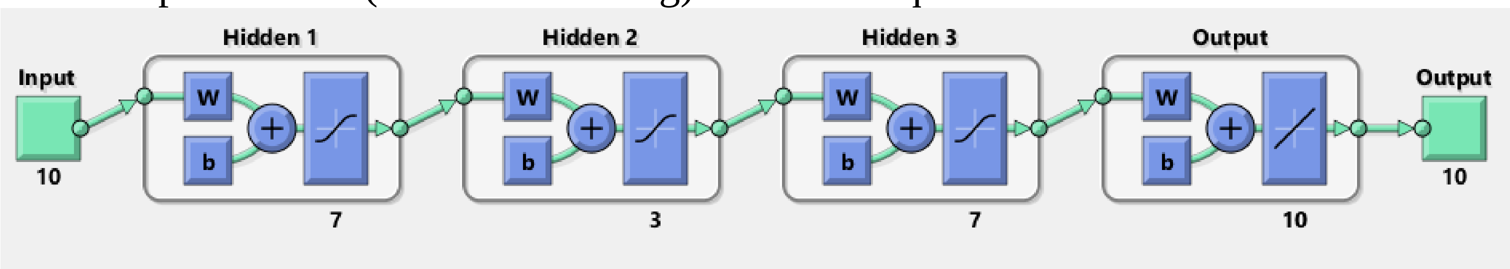
FNN with Greedy Supervised Pre-training



The final structure is identical with the above, but the weight vector **W** and bias **b** of each layer are initialized with greedy supervised pre-training method. Regarding the above neural network diagram, the first **W** and **b** in **hidden 1** is initialized by the result of training the network above with structure [10 7 1]. **W** and **b** in **hidden 2** is initialized by training another network with structure [7 3 1], with the transformed input from the first layer.

FNN with Unsupervised Pre-training with AE

This time, the weights are initialized by first training an auto-encoder neural network that performs non-linear dimensionality reduction. The W and b from hidden 1 and hidden 2 in below diagram are used to initialize the network of the first diagram in this page. Further optimization (called fine-tuning) is done in equivalent manner with above cases.



Generative Auto-encoder Model

Using this original model, the auto-encoder neural network can also forecast by itself without any separate network specifically trained for forecasting. Using the same network structure of [10 7 3 7 10] as above, one can make an initial guess about the forecast value and consider it as the last dimension input value among the 10. Then the output of the network includes the reconstruction of 10^{th} unit that better correlates with the other 9 inputs, based on the training with previous data.

FORECASTING RESULT

(X-axis: Different Models / Y-axis: Mean Squared Error Performance)

[Both pre-training methods showed significant improvement.]
[Wider structure increases flexibility, but requires good pre-training.]

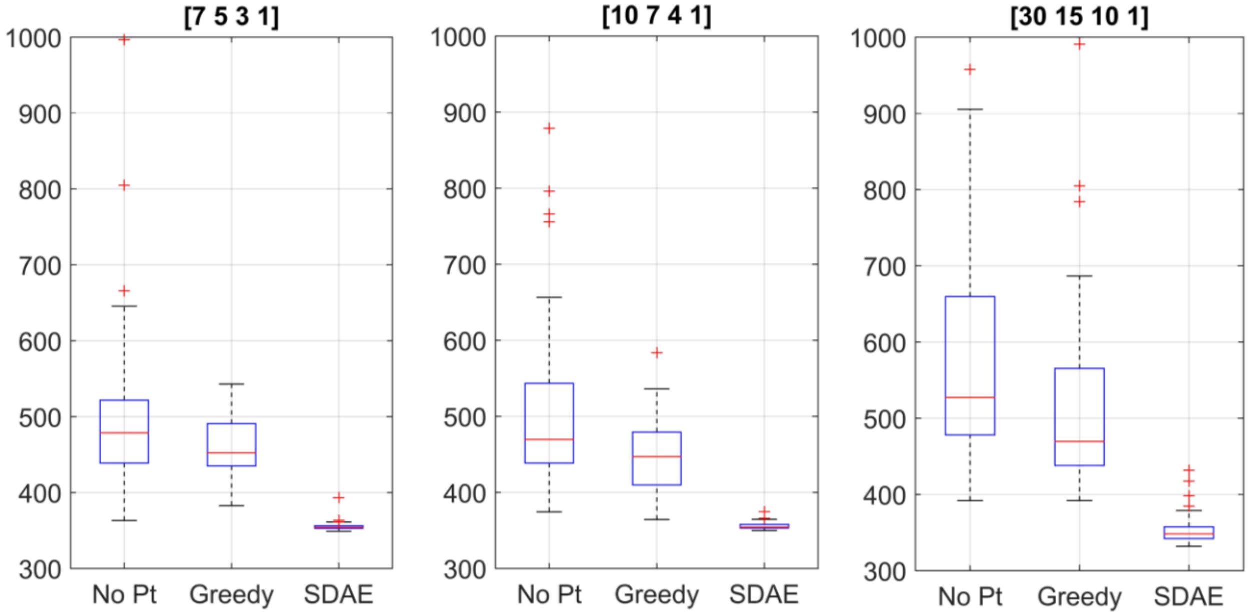


Figure. First three neural network models tested on the sunspot time-series data

[Deeper network shows better performance with proper pre-training.]

% *however, the network must be relatively 'wide' and complicated in order to harvest the benefit of the deeper structure. Otherwise, deeper structure can make the weights stuck in local optima.*

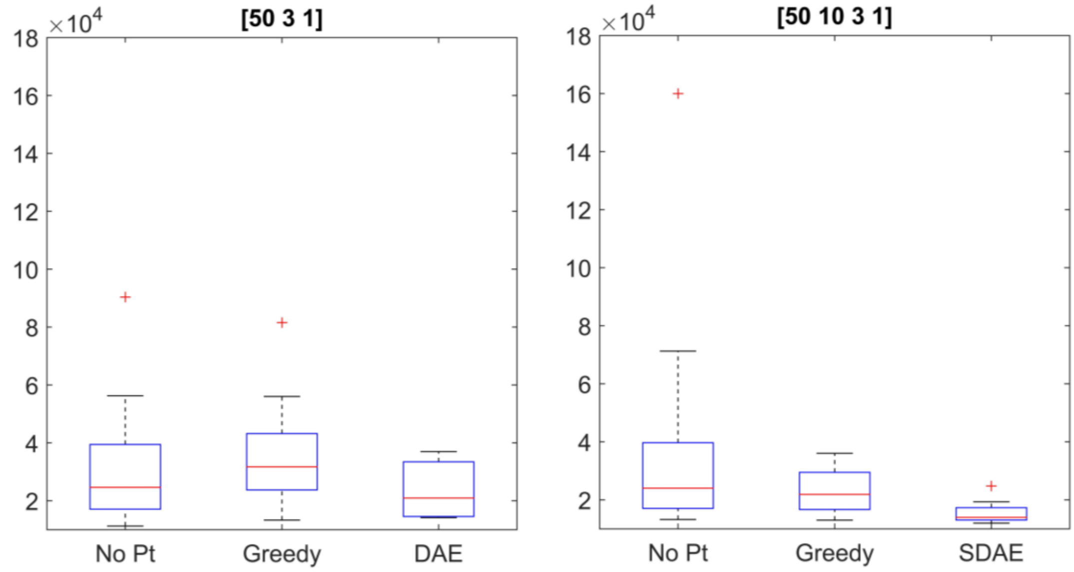


Figure. Neural network models tested on chickenpox data.

METHODS OF FURTHER IMPROVEMENT

(Traditional methods of further improving model performances are tested and proven to be useful.)

Model Averaging:

Averaging the forecast value from two models are useful, when the correlation of the two error is low, for statistical reason, and this turned out to be the case for different models used in this study, for time-series forecasting.

Hybrid Model:

Pre-training the lower layer and upper layer in auto-encoder and greedy supervised method separately may improve the result, and there are other methods to mix two models for logically sound objective.