

# Input Window Size for Deep Recurrent Reinforcement Learning



Eric Muccino

Feb 27, 2019 · 4 min read

Deep Recurrent Reinforcement Learning makes use of a Recurrent Neural Network (RNN), such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) based networks, for learning a value function that maps environment states to action values. Recurrent Neural Networks are useful for modeling time-series data since the network maintains a memory, learning to retain useful information from inputs of prior model inferences. Every time the model is called, the memory is updated in correspondence with the current inputs.

RNN memory states can be highly effective, but problems may arise when input data is missing information or contains noise. A single time-step of input data that is particularly corrupt can cause faulty alterations to the network's memory, possibly erasing vital information that is needed for future time-steps. In this post, we will look at widening our model input window as an attempt to combat the fragility of RNN hidden memory against noisy inputs.

## Time-series Input Windows

One noisy input can severely corrupt a RNN's memory. In order to give our model inputs more stability, we will include previous time-steps as part of our model input. For example, an input window of size 4 will include our current time-step along with the 3 previous time-steps. The intuition for this is that the noise contained in a single time-step may be tamed by the context of the other time-steps within the window. We will look at the results of using inputs with time-series windows of 1, 4, 8, 16, and 32 in a Deep Recurrent Reinforcement Learning Setting.

## Data and Model

For this test, we will use pseudo-randomly generated data as described in a previous post [Training Recurrent Neural Networks on Long Sequences](#). The data is

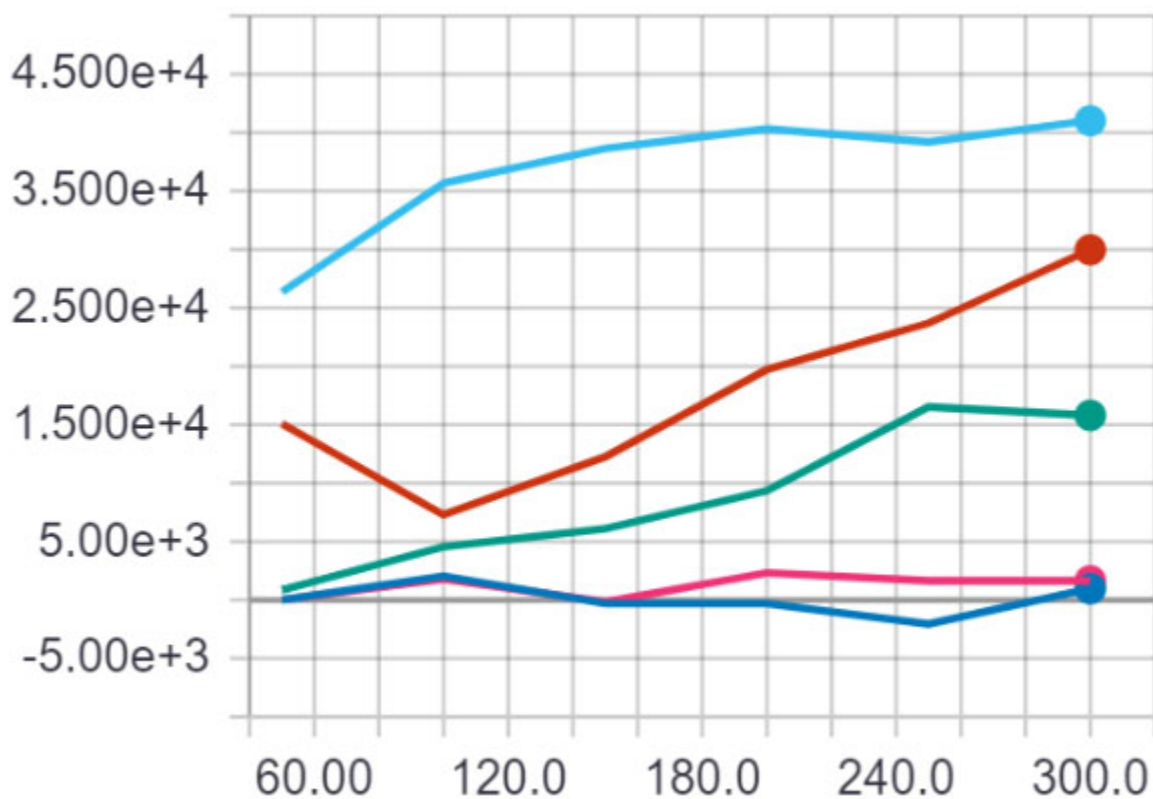
generated from an underlying wave function with stochastically varying amplitudes and additional Gaussian noise. The generated data will be treated as if it were Price data for a commodity that we can buy or sell. Our model will learn a value function, predicting the future returns of buying or selling the commodity at each time-step. The training data outputs are calculated using Bellman's Q-Value equation as discussed by Gayatri Vadali in her post [Q Matrix Update to train Deep Recurrent Q Networks More Effectively](#). We generate 20 sequences, using 15 sequences for training and 5 for testing.

Our model consists of a single GRU layer with 16 neurons. After training our network to map current Price values to future action-reward values, we use our network to decide actions during a trading simulation with our training and testing data. We train each model for 300 epochs and test every 50 epochs through trading simulation.

## Results

First, we examine simulation profit for each model using training data.

### Training Data Results (Profit v Epoch)



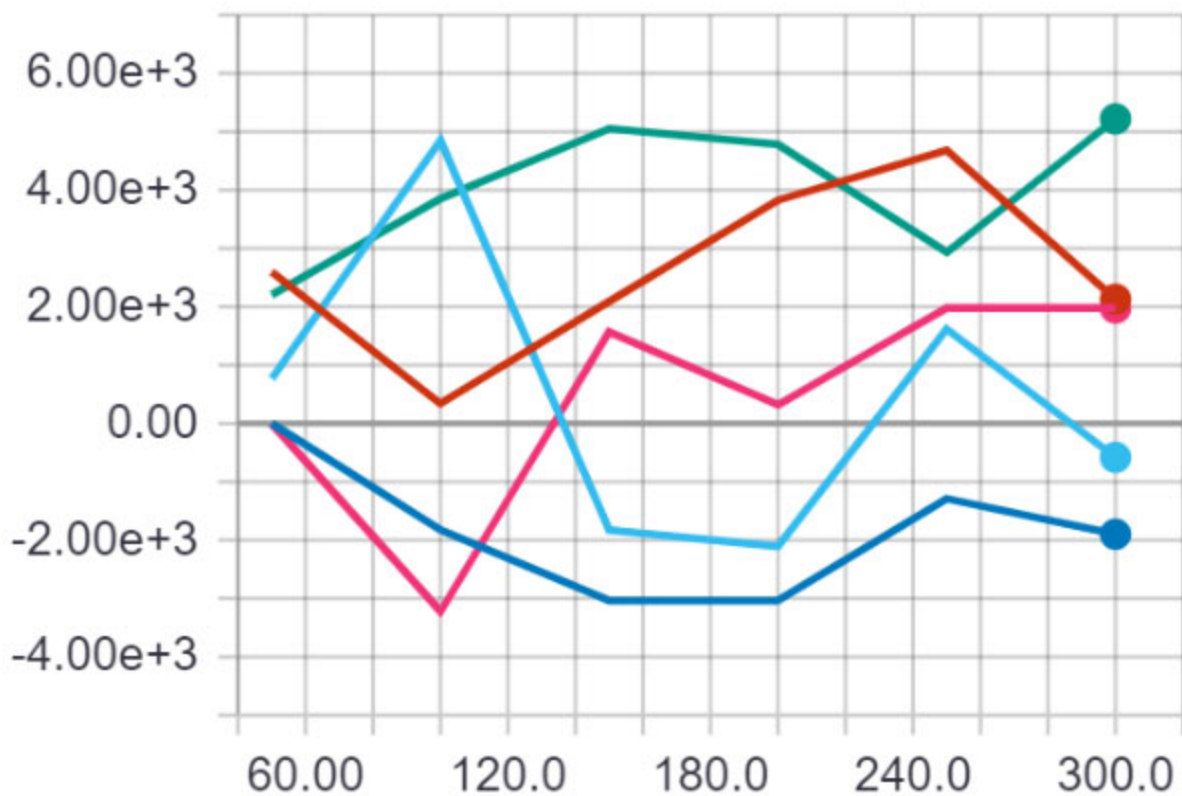
Input Window Size: Dark Blue=1, Pink=4, Green=8, Red=16, Light Blue=32

We see that the larger our input window is, the better the model performs. This is not surprising since more inputs provides the model with more unique input signatures,

making the mapping of inputs to outputs of the training data an easier task. A input window of 1 can barely learn a profitable value function because the noise within the data is too substantial. However, providing our model with an input window of 32 allows us to effortlessly learn an accurate value function.

## Test Data Results (Profit v Epoch)

Let's examine the performance of each model over the test data.



Input Window Size: Dark Blue=1, Pink=4, Green=8, Red=16, Light Blue=32

Unsurprisingly, a window of size 1 does not produce a profit. As we increase the window size to 4 and 8, our model is able to learn a value function that profits over the test data. However, further increasing the window size to 16 causes a decline in performance. A window size 32 is not capable of profiting at all. For this particular combination of data and neural network model, an input window size of 8 has the best performance on our validation data out of the 5 window sizes that were tested.

## Conclusion and Future Work

In this post, we have explored the effects of increasing the time-series input window to help manage the impact of noisy data on RNN memory states in the Deep Recurrent Reinforcement Learning setting. An experiment has shown that increasing the input window can benefit model performance. A window that is too large, however, can potentially lead to over-fitting, degrading model performance.

Appropriate window size will certainly vary from problem to problem, depending on the data and model being used. For future work, window size could be examined in conjunction with model regularization, which was not used in this experiment. Model regularization such as L2-norm or dropout may allow the use of larger window sizes that otherwise would cause over-fitting.

## Masala.AI

The Mindboard Data Science Team explores cutting-edge technologies in innovative ways to provide original solutions, including the **Masala.AI** product line. Masala provides media content rating services such as **vRate**, a browser extension that detects and blocks mature content with custom sensitivity settings. The vRate browser extension is available for download via the Chrome Web Store. Check out [www.masala.ai](http://www.masala.ai) for more info.

[Machine Learning](#)[Reinforcement Learning](#)[Algorithmic Trading](#)[Recurrent Neural Network](#)[Neural Networks](#)[About](#) [Help](#) [Legal](#)

Get the Medium app

