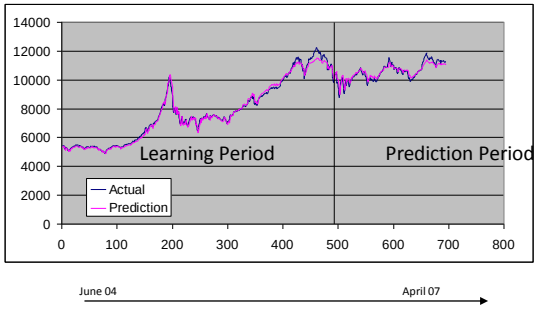
Computational Intelligence

Unit # 9

Evolutionary Neural Networks (Source: Wikipedia)

- Neuroevolution, or neuro-evolution, is a form of machine learning that uses evolutionary algorithms to train artificial neural networks.
- It is useful for applications such as games and robot motor control, where it is easy to measure a network's performance at a task but difficult or impossible to create a syllabus of correct input-output pairs for use with a supervised learning algorithm.
- In the classification scheme for neural network learning these methods usually belong in the reinforcement learning category

Assignment # 3



NN Types

- Feedforward NNs such as the standard multilayer NN, functional link NN and product unit NN receive external signals and simply propagate these signals through all the layers to obtain the result (output) of the NN. There are no feedback connections to previous layers.
- Recurrent NNs, on the other hand, have such feedback connections to model the temporal characteristics of the problem being learned.

FF NN

- A FFNN can have more than one hidden layer.
- However, it has been proved that FFNNs with monotonically increasing differentiable functions can approximate any continuous function with one hidden layer, provided that the hidden layer has enough hidden neurons.
- A FFNN can also have direct (linear) connections between the input layer and the output layer.

Activation Function in NN

- Note that each activation function can be a different function.
- It is not necessary that all activation functions be the same. Also, each input unit can implement an activation function.
- It is usually assumed that input units have linear activation functions.

Product Unit Neural Networks

- Product unit neural networks (PUNN) have neurons that compute the weighted product of input signals, instead of a weighted sum.
- While PUNNs provide the advantage of having smaller network architectures, a major drawback of PUs is an increased number of local minima, deep ravines and valleys.
- The search space for PUs is usually extremely convoluted. Gradient descent, which works best when the search space is relatively smooth, therefore frequently gets trapped in local minima or becomes paralyzed.

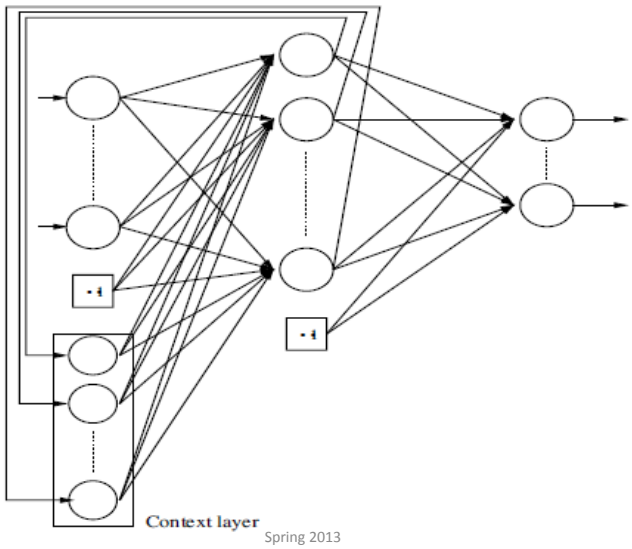
Simple Recurrent Neural Networks

- Simple recurrent neural networks (SRNN) have feedback connections which add the ability to also learn the temporal characteristics of the data set.

Elman SRNN

- The Elman SRNN, as illustrated on the next slide, makes a copy of the hidden layer, which is referred to as the *context layer.*
- *The purpose of the context layer is to* store the previous state of the hidden layer, i.e. the state of the hidden layer at the previous pattern presentation.
- The context layer serves as an extension of the input layer, feeding signals representing previous network states, to the hidden layer.

Elman SRNN (Cont'd)



Jordan SRNN

- Jordan SRNN, on the other hand, make a copy of the output layer instead of the hidden layer.
- The previous state of the output layer then also serves as input to the network.

Jordan SRNN (Cont'd)