classes of ANNs in real world applications. According to him, a multilayer perceptron has three distinctive characteristics:

- The model of each neuron in the network includes usually a non-linear activation function, sigmoids or hyperbolic.

- The network contains one or more layers of hidden neurons that are not part of the input or output of the network to learn complex and highly nonlinear tasks by extracting progressively more meaningful features from the input patterns.

The network exhibits a high degree or connectivity from one layer to the next one.

III. METHODOLOGY The proposed forecasting of Foreign exchange rate used

AFERFM with the considerations of the existing HFERFM. A. Hidden Markov Model The Hidden Markov Model (HMM) is a variant of a finite

state machine having a set of hidden states Q, an output alphabet (observations), O, transition probabilities, A, output (emission) probabilities, B, and initial state probabilities, Π. The current state is not observable instead, each state produces an output with a certain probability (B). Usually the states, Q, and outputs, O, are understood, so an HMM is said to be a triple, ( A, B, Π ).

Hidden states Q = { qi; i = 1, . . . , N }. Transition probabilities A = {aij = P(qj at t +1 | qi at t)},

where P(a | b) is the conditional probability of a given b, t = 1, . . . , T is time, and qi in Q. Informally, A is the probability that the next state is qj given that the current state is qi.

Observations (symbols) O = { ok }, k = 1, . . . , M . Emission probabilities B = { bik = bi(ok) = P(ok | qi) },

where ok in O. Informally, B is the probability that the output is ok given that the current state is qi.

Initial state probabilities Π = {pi = P(qi at t = 1)}. The model is characterized by the complete set of

parameters: Λ = {A, B, Π} There are 3 canonical problems to solve with HMMs:

- Given the model parameters, compute the probability of a particular output sequence. This problem is solved

by the Forward and Backward algorithms. - Given the model parameters, find the most likely

sequence of (hidden) states which could have generated a given output sequence.This is solved by the Viterbi algorithm and Posterior decoding.

Therefore we get t (ij, ) 

( ) ( ( 1)) ( ) PO( | )

i i a b o t  ij j i1 j

The probability of output sequence can be ressed as

exp NN P   N

     i1

tt ( ) ( )

The probability of being in state q at time t

i

( , ) 1



tt j

Estimate initial probabilities p transition r 

probability a  ij

t ij i

1 1

r i  i

 

1 1

t

( , ) ()

(6) i () (5)

(i)  i j tt    ( ) ( )

N P(|) (4) : (3)

( | )  ( )a b t  11

t ij

     

ij j ( ( 1)) ( j) t 1 (2) - Given an output sequence, find the most likely set of state

transition and output probabilities solved by the Baum-Welch algorithm In this work, Baum-Welch algorithm was used.

A. Theoretical aspect of Baum-Welch Algorithm Let us define ξ t(i, j), the joint probability of being in state qi at time t and state qj at time t +1 , given the model and the observed sequence:

( , ) 

 

i j P q t q q t q

, ( 1) j | , )

 ( ( )