The Application of Qubit Neural Networks for Time Series Forecasting with Automatic Phase Adjustment Mechanism

Carlos R. B. Azevedo[1] and Tiago. A. E. Ferreira[1]

[1]Centro de Ciências e Tecnologia – Universidade Católica de Pernambuco (UNICAP)
Rua do Príncipe, 526, Boa Vista, 50050-900, Recife – PE – Brazil

`{azevedo, taef} @dei.unicap.br`

Abstract. *Quantum computation, quantum information and artificial intelligence have all contributed for the new non-standard learning scheme named Qubit Neural Network (QNN). In this paper, a QNN based on the qubit neuron model is used for real world time series forecasting problem, where one chaotic series and one stock market series were predicted. Experimental results show evidences that the simulated system is able to preserve the relative phase information of neurons quantum states and thus, automatically adjust the forecast's time shift.*

1. Introduction

Quantum Computation has evolved from the theoretical studies of Feynman (1982), Deutsch (1985), and others, to an intensive research field since the discovery of a quantum algorithm which can solve the problem of factorizing a large integer in polynomial time by Shor (1994). Matsui et al (2000) proposed a Qubit Neuron Model which exhibits quantum learning abilities. This model resulted on a quantum multi-layer feed forward neural network proposal [Kouda et al, 2004] which implements Quantum Mechanics (QM) effects, having its learning efficiency proved on non-linear controlling problems [Kouda et al, 2005].

The main objective of this work is to apply the QNN model for real world time series forecasting problem, where the influence of QM effects (mainly, superposition) is expected to capture the forecast's phase information, leveling up the model prediction quality for real world time series.

This paper is structured as follows: sections 1 to 3 states the problem to be solved and justifies QNN usage; section 4 presents concepts of quantum computing; sections 5 and 6 describes the QNN model and the methodology used on experiments; sections 7 and 8 discusses the results and gives conclusions, respectively.

2. Time Series Forecasting Problem

From the classical picture, a time series (TS) is the set of the measured properties of a phenomenon (physical or not) ordered chronologically by the observer. Mathematically, a TS Z_t can be simply defined as

$$Z_t = \{ z_t \in \mathbb{R} : t = 1, \ldots, n \}, \tag{1}$$

where t is a chronological index, generally considered to be time and n is the number of observations [Ferreira *et al*., 2004]. However, a statistical definition can be more

appropriated as probabilistic properties of QM will be discussed later on. Hence, a real valued TS (or stochastic process) X is the real mapping

$$X : \mathrm{T} \times \Omega \to \mathbb{R} , \qquad (2)$$

such that, for each fixed t, $X(t, w)$ is a random variable in a probability space $(\Omega, \Sigma, \mathrm{P})$, where w are the elementary events, $T = \{1, \ldots, n\}$ is a set of indexes that enumerates sequentially the measures, Ω is the certainty event, Σ is a sigma-algebra over Ω and P is a function defined over Σ which attributes probabilities to the subsets of Ω [Fuller, 1976]. In this sense, the TS forecasting problem can be finally stated over a set of random variables $X(t, w)$, with the form

$$\{X(t_1, w_1), X(t_2, w_2), \ldots, X(t_n, w_n)\}, \qquad (3)$$

as the problem of determining the upcoming events $X(t_{n+1}, w_{n+1}), \ldots, X(t_{n+h}, w_{n+h})$, where h denotes the forecast horizon.

2.1 The Random Walk Dilemma

A simple linear model for TS forecasting is the random walk model (RW) given by

$$Z_t = Z_{t-1} + r_t, \qquad (4)$$

where Z_{t-1} is the immediate time lag of Z_t point and r_t is a noise term with a Gaussian distribution of null mean and variance σ^2 $(r_t \sim N(0, \sigma^2))$.

Controversy arises from the RW model concerning financial TS forecasting: economic theoreticians stated that stock market prices follow a RW model and so, cannot be predicted [Malkiel, 1978]. The impossibility of predicting RW series comes from the low level of correlation between the time lags and the points which are to be forecasted. Actually, the RW model is the first approximation to financial TS, as observed in some experiments with Artificial Neural Networks (ANN) [Sitte and Sitte, 2002]. Still, Lo and McKinley (2002) argued that there exist predictable components in the stock market, implying superior long-term investment returns through disciplined active investment management. This research aims at adding value to such argument by applying quantum computation intensive strategies on financial TS forecasting problem.

3. Phase Adjustment

As a result of the RW dilemma, some forecasts obtained for TS with non-linear dependencies are one step time shifted (or out-of-phase). Ferreira (2006), in his Ph.D. thesis, has not only implemented a procedure of phase adjustment based on ANN, but has shown that this one step time shift in the prediction results is due to the asymptotically behavior of the ANN model towards a RW like model, when designing an estimator for such non-linear TS.

Let $\hat{Z}$ be the desired estimator for TS Z_t driven by a RW like model. The expected value of the distance between the estimator and the real series in an ideal predictor should tend to zero:

$$E\left[\hat{Z}_t - Z_t \right] \to 0 . \qquad (5)$$

In addition, considering the noise terms r_k, with $k \in \{1,..., t\text{-}1\}$, with independent components ($r_k \neq r_j$, for each $k \neq j$) and $E[r_k] = 0$, implies

$$E\left[\hat{Z}_t\right] \rightarrow E[Z_{t-1}].$$ (6)

The procedure of phase adjustment is conceived in the sense of transforming out-of-phase predictions into in-phase predictions by the fine-tuning of the one step time shift. The applying of QNN simulation is expected to supply an automatic phase adjustment due to QM effects on the QNN learning process in order to overcome the phase information loss, explained below in the next subsection.

3.1. The Learning of Complex Number Phase Information

In Quantum Mechanics, reality is better described (and consequently better predicted) through the use of a numerical set that preserves more information then the real set: the complex set. In the exponential form, a complex number W can be written as

$$W = A \cdot e^{i \cdot \phi},$$ (7)

where $A \in \mathbb{R}$ is the amplitude, $\phi \in \mathbb{R}$ is the phase and i is the imaginary unit. The observable part of W is a real number and, because $\mathbb{R} \subset \mathbb{C}$, it does not describe all of the original information. Mathematically, the observable part of a complex number is its squared modulus or, more specifically,

$$|W|^2 = W \cdot W^* = \left(A \cdot e^{i\phi}\right) \cdot \left(A \cdot e^{-i\phi}\right)$$ (8)

$$= A^2 \cdot \left(e^{i\phi} \cdot e^{-i\phi}\right) = A^2 \cdot e^{i \cdot (\phi - \phi)} = A^2,$$

where "*" denotes the complex conjugate. It can be clearly noted that the phase information is destroyed by a measurement. Therefore, the knowledge acquisition achieved by the usage of complex numbers within QNN model can lead to the ability of learning complex number phase information, improving the quality of the predictor.

4. Quantum Computing

The atomic piece of information of Quantum Computing (QC), analogous of the classical bit, is the quantum bit, or simply qubit. As its classical counterpart, it has two distinct states in which it can be observed, the base states $|0\rangle = \left[(1,0)\ (0,0)\right]^T$ and $|1\rangle = \left[(0,0)\ (1,0)\right]^T$, with $|0\rangle, |1\rangle \in \mathbb{C}^2$. The symbol "$|\cdot\rangle$" is part of the Dirac notation. Differently of classical bit, however, a qubit can lie on a state of superposition of the base states. Theoretically, superposition means that the amount of information that can be stored on a single qubit is infinite. However, when measured, the qubit will collapse into exactly one of the base states with a certain probability. An arbitrary qubit state $|\psi\rangle$ in superposition can be expressed as a linear combination between states $|0\rangle$ and $|1\rangle$ as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle,$$ (9)

where the scalars $\alpha \in \mathbb{C}$ and $\beta \in \mathbb{C}$ are the amplitudes of the states $|0\rangle$ and $|1\rangle$, respectively. After a measure, $|\psi\rangle$ will collapse to $|0\rangle$ with probability $|\alpha|^2$ or it will collapse to $|1\rangle$ with probability $|\beta|^2$ [Nielsen and Chuang, 2005]. Naturally,

$$|\alpha|^2 + |\beta|^2 = 1. \tag{10}$$

5. Proposed Qubit Neural Network Model

The implemented qubit neural network model is based on Mitrpanont and Srisuphab (2002) who prepared a complex-valued multilayer backpropagation neural network model to exhibit QM's effects. This model is also inspired in [Kouda et al, 2004] and will be now detailed, with adaptations.

5.1. Qubit Neuron Model

A neuron has a quantum state and it is inactive when its state is $|0\rangle$; it fires when its state is $|1\rangle$ and its arbitrary state is given by the superposition state described on Equation (9). All information that flows through the network is complex encoded. Therefore, for presenting real data to QNN, all points are mapped into complex numbers on the form given in Equation (7) (with $|W|^2 = A = 1$), through a phase $\theta_i \in (0, \frac{\pi}{2})$, in order to create the complex inputs x_i which stores the phase information:

$$x_i = e^{i\theta_i}, \tag{11}$$

where $\theta_i = \frac{\pi}{2} \cdot X_n$, with X_n denoting the normalized real data. The normalization of the series real data is done through a linear transformation, lying in the interval [0.2, 0.8], for two reasons: to avoid the saturation regions of the logistic sigmoid functions (Equation (13)) - improving the learning convergence - and to avoid orthogonal quantum states (as θ_i interacts with the qubit state), preserving, in this way, the superposition effects.
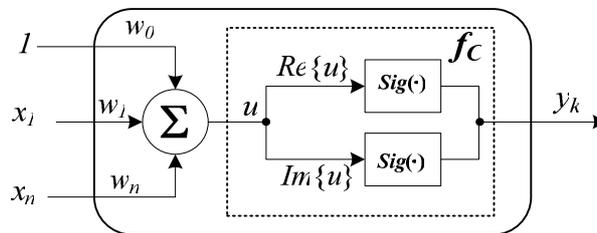

Figure 1. Schematic diagram for the processing realized by the qubit neuron.

Accordingly, the operation realized by the quantum neuron consists of two distinct steps: the weighted sum of input complex signals phase, written as

$$u = \sum_{i=0}^{N} w_i x_i, \tag{12}$$

where N is the number of signals received, $w_i \in \mathbb{C}$ are the weights (with w_0 denoting the bias) and x_i are the complex input signals. The second step consists of the non-linear activation of the quantum neuron given by

$$y_k = f_C(u) = Sig(\text{Re}\{u\}) + i \cdot Sig(\text{Im}\{u\}). \tag{13}$$

Equation (13) means that the real and imaginary parts of u in Equation (12) are submitted separately into a real valued logistic sigmoid function ($Sig(\cdot)$) for composing a complex output. The function $f_C : \mathbb{C} \to \mathbb{C}$ is continuous and differentiable, which allows its usage on the training algorithm. The superposition effects arise naturally as a result of the network dynamics due to the complex signals arithmetic which performs quantum state phase rotations. Finally, the real output Y_R of the QNN model is considered to be the inverse mapping of Equation (11) over the phase of the complex response (Equation (13)) of the output layer neuron.

5.2. Quantum Training Algorithm

The proposed QNN was trained with complex backpropagation algorithm [Nitta, 1994] used to learn complex number information. The algorithm performs the descent-gradient minimization of the sum of the squared error function

$$E = \frac{1}{2} \cdot \sum_{p}^{P} \sum_{i}^{N} (desired_i^p - output_i^p)^2, \tag{14}$$

where P is the number of input patterns, N is the number of output neurons. The weight adjustment equations are now described: let C be the last layer whereas the first layer is denoted as "layer 1". For the output layer C,

$$w_{i(t+1)}^C = w_{i(t)}^C + \Delta w_i^C, \tag{15}$$

$$\Delta w_i^C = \overline{y_{pi}^{C-1}} \cdot \eta \cdot \delta_{pi}^C, \tag{16}$$

$$\delta_{pi}^C = (d_{pi} - y_{pi}^C) \cdot f_C'(u_j^C), \tag{17}$$

where $\overline{y_{pi}^{C-1}}$ denotes the complex conjugate of output y_{pi} of neuron i of layer C-1 in response to input pattern p, η denotes the learning rate, d_{pi} denotes the desired output for neuron i in response to input pattern. For intermediate processing layers c,

$$w_{i(t+1)}^c = w_{i(t)}^c + \Delta w_i^c, \tag{18}$$

$$\Delta w_i^c = \overline{y_{pi}^{c-1}} \cdot \eta \cdot \delta_{pi}^c, \tag{19}$$

$$\delta_{pi}^c = \delta_{u_{pj}^c} \cdot f_C'(u_{pj}^c), \tag{20}$$

$$\delta_{u_{pj}^c} = \sum_{k=1}^{N^{c+1}} \delta_{pk}^{c+1} \cdot w_{ki}^{c+1}, \tag{21}$$

where N^{c+1} is number of neurons at layer $c+1$.

6. Methodology for Experiments and Setup

The number of hidden layers M of QNN was fixed in $M = 1$, so was the number of nodes in output layer, $h = 1$, which means that the forecast horizon was one step ahead. The number of input nodes equals the number of lags for each TS. Once presented the lags data, the QNN will predict what would occur next in the future. The number of lags was determined by the lagplot method [Percival and Walden, 1998]. Besides, the number of hidden nodes $H \in \{3, 5, 8\}$ varied systematically in experiments. The same was done with the learning rate $\eta \in \{10^{-1}, 10^{-2}, 10^{-3}\}$.

These parameters values were chosen by preliminary testing experiments. For each parameter combination, 10 random weight's initializations were done, resulting in a total of 90 experiments per TS. Data were then divided on 3 disjoint sets: training (50% of data), validation and test (each one consisting of 25% of data) [Prechelt, 1994]. The stop criteria used in training process was the third class cross-validation [Prechelt, 1998] and a maximum of 300000 training epochs.

On analyzing results, the QNN model is said to be equivalent to other models on a particular statistic if it falls under the confidence interval constructed in other experiments with ANN and ARIMA models which can be found in [Ferreira, 2006].

6.1. Statistics and Measures for Analyzing Results

The analysis of forecasts was based on the following statistics and error measures: the Mean Square Error (MSE), Mean Absolute Percentual Error (MAPE), Normalized Mean Square Error (NMSE), Prediction On Change In Direction (POCID) and Average Relative Variance (ARV). Refer to Ferreira *et al*. (2005) for equations and details.

7. Results and Discussion

7.1. Sunspot Time Series

This benchmark TS consists of 289 annual observations of the number of dark regions formed on Sun's surface. It exhibits a chaotic behavior which corresponds to non-linear data dependencies. By the lagplot analysis, the chosen lags were Z_{t-1} to Z_{t-3} (Figure 2), composing 3-H-1 QNN topologies.
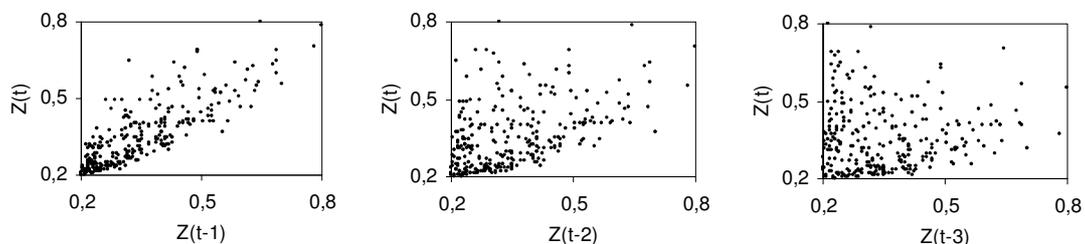

Figure 2. Lagplot for normalized Sunspot TS with chosen lags Z_{t-1} to Z_{t-3}.

On average, the best topology in experiments was 3-3-1 with $\eta = 10^{-1}$, though the best forecast was achieved with a 3-5-1 QNN with $\eta = 10^{-1}$ after 1775 epochs of training (Figure 3). Table 1 compares results of the best prediction achieved with each
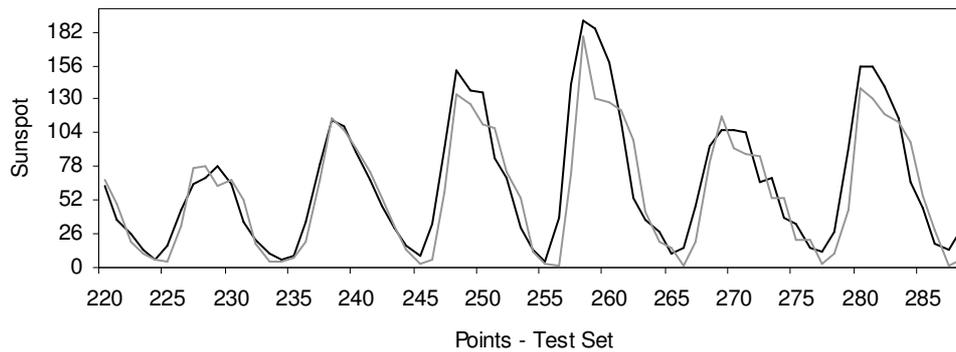
Figure 3. Sunspot series (darker line) and best forecast (lighter line) obtained by the QNN model for the test set.

model (QNN, ANN and ARIMA). Analyzing the measures for the best QNN initialization, since NMSE < 1, the predictor is better than a RW model and the forecast can be considered to be in-phase. Moreover, it is better than a head-or-tails experiment (POCID > 50%). Finally, it is better then a model which simply predicts the mean of the TS (ARV < 1).

Table 1. Measures of the best predictions obtained for Sunspot's test set.

Measure	QNN	ANN	ARIMA(9,0,1)	Best Model
MSE	0,0039	0,9205	0,0219	QNN
NMSE	0,3835	0,3443	0,7805	QNN & ANN
POCID	88,4058%	90,0000%	75,0000%	QNN & ANN
ARV	0,0127	0,1418	0,4007	QNN
MAPE	11,6140%	2,41%	42,35%	ANN

Table 2. Average and standard deviations for the results of the measures and statistics obtained in the QNN experiments with Sunspot test set.

QNN Topology	Learning Rate	MSE	NMSE	POCID	ARV	MAPE
3-3-1	10^{-1}	0,0046 ± 0,0013	0,4913 ± 0,1166	79,1304 ± 3,6203	0,0150 ± 0,0073	11,3087 ± 1,2259
	10^{-2}	0,0052 ± 0,0011	0,5863 ± 0,1674	78,8406 ± 3,7345	0,0159 ± 0,0054	11,9553 ± 1,0432
	10^{-3}	0,0053 ± 0,0022	0,5797 ± 0,1404	79,5652 ± 2,7075	0,0148 ± 0,0036	11,9300 ± 1,7010
3-5-1	**10^{-1}**	0,0057 ± 0,0029	0,5072 ± 0,1002	82,1739 ± 5,0642	0,0128 ± 0,0006	11,8542 ± 1,5819
	10^{-2}	0,0077 ± 0,0071	0,5650 ± 0,1678	78,1159 ± 3,8590	0,0133 ± 0,0011	12,9514 ± 2,8510
	10^{-3}	0,0049 ± 0,0020	0,4398 ± 0,0821	77,1014 ± 4,3863	0,0125 ± 0,0006	11,9547 ± 1,5210
3-8-1	10^{-1}	0,0156 ± 0,0109	0,6458 ± 0,2116	78,4058 ± 5,7407	0,0135 ± 0,0013	15,7728 ± 4,1987
	10^{-2}	0,0064 ± 0,0033	0,4977 ± 0,1196	81,8841 ± 3,9023	0,0130 ± 0,0006	12,7275 ± 2,0007
	10^{-3}	0,0048 ± 0,0012	0,4530 ± 0,0558	78,1159 ± 4,4173	0,0126 ± 0,0007	11,6152 ± 1,0045

These results show that, although achieving a better MSE and ARV then ANN model, the QNN learning efficiency was similar to classical ANN for NMSE and POCID, but is better than ARIMA(p, d, q) model in overall. This can be concluded by comparing the experiments conducted by Ferreira (2006), which used the same methodology and similar setups with equivalent ANN topologies, in terms of degrees of freedom, while applied Box & Jenkins methodology, obtaining an ARIMA(9,0,1).

7.2. Dow-Jones Industrial Average (DJIA) Time Series

The DJIA is a stock market index consisting of 30 companies. The series used on experiments is composed of 1400 daily observations starting at January 1998 until August 26, 2003. By the lagplot analysis, the chosen lags were Z_{t-1} to Z_{t-3} (Figure 4).
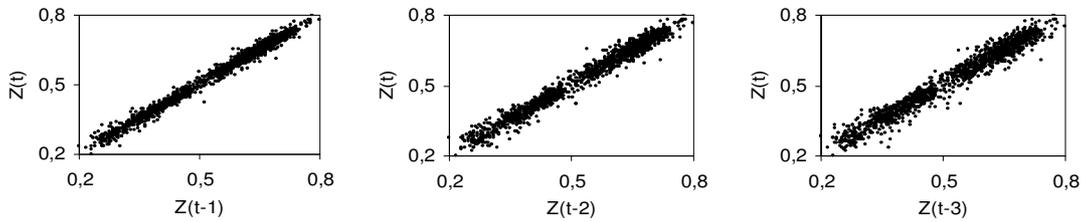

Figure 4. Lagplot for normalized DJIA time series with chosen lags Z_{t-1} to Z_{t-3}.
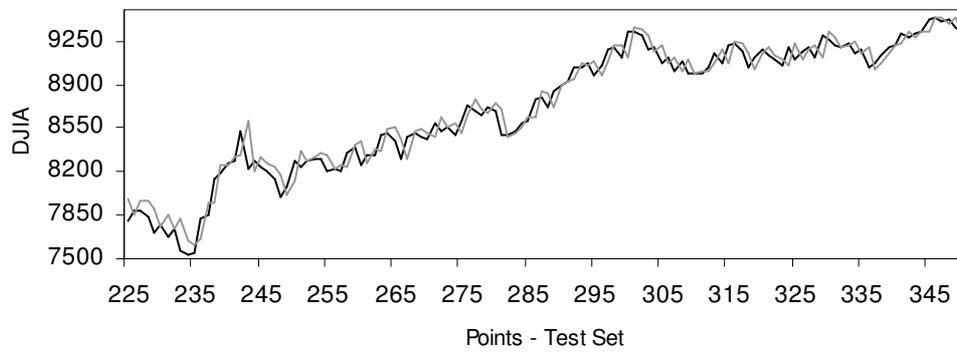

Figure 5. DJIA series (darker line) and best forecast (lighter line) obtained by the QNN model for the last 125 points of the test set.

Table 3. Measures of the best predictions obtained for DJIA's test set.

Measure	QNN	ANN	ARIMA(1,0,1)	Best Model
MSE	$3,2492 \cdot 10^{-4}$	0,0827	$5,8033 \cdot 10^{-4}$	QNN
NMSE	0,7898	0,9876	1,2649	QNN
POCID	47,2934%	46,7400%	46,1000%	QNN
ARV	$1,9891 \cdot 10^{-4}$	$3,4226 \cdot 10^{-3}$	0,0392	QNN
MAPE	3,9243%	0,37%	8,3200%	ANN

Table 4. Average and standard deviations for the results of the measures and statistics obtained in the QNN experiments with DJIA test set.

QNN Topology	Learning Rate	MSE	NMSE	POCID	ARV	MAPE
	10^{-1}	$0,0009 \pm 0,0006$	$11,5045 \pm 18,4157$	$47,0085 \pm 0,4767$	$0,0005 \pm 0,0004$	$6,5701 \pm 2,2732$
3-3-1	10^{-2}	$0,0003 \pm 0,0000$	$1,1616 \pm 0,0601$	$46,4957 \pm 0,6955$	$0,0002 \pm 0,0000$	$4,0716 \pm 0,0338$
	10^{-3}	$0,0005 \pm 0,0002$	$1,9736 \pm 1,0150$	$46,9516 \pm 0,4902$	$0,0003 \pm 0,0001$	$4,6832 \pm 1,2614$
	10^{-1}	$0,0006 \pm 0,0002$	$2,1013 \pm 0,7436$	$46,7806 \pm 0,5065$	$0,0003 \pm 0,0001$	$5,2023 \pm 1,0087$
3-5-1	10^{-2}	$0,0003 \pm 0,0000$	$1,2005 \pm 0,0650$	$46,5812 \pm 0,5443$	$0,0002 \pm 0,0000$	$4,0095 \pm 0,1429$
	10^{-3}	$0,0003 \pm 0,0001$	$1,1737 \pm 0,4272$	$46,3818 \pm 0,5949$	$0,0002 \pm 0,0000$	$4,0257 \pm 0,6448$
	10^{-1}	$0,0004 \pm 0,0001$	$2,3596 \pm 2,8116$	$46,7521 \pm 0,8495$	$0,0002 \pm 0,0001$	$4,1634 \pm 0,6810$
3-8-1	10^{-2}	$0,0004 \pm 0,0000$	$1,1871 \pm 0,0932$	$46,6667 \pm 0,6084$	$0,0002 \pm 0,0000$	$4,1738 \pm 0,2331$
	10^{-3}	$0,0003 \pm 0,0001$	$1,2111 \pm 0,4725$	$46,6952 \pm 0,5008$	$0,0002 \pm 0,0000$	$3,9648 \pm 0,3965$

The best topology in experiments was, on average, 3-3-1 with $\eta = 10^{-2}$, though the best forecast was achieved with a 3-5-1 QNN with $\eta = 10^{-1}$ after 51587 training epochs (Figure 5). Analyzing the measures for the best QNN initialization, the predictor is better than a RW model (NMSE < 1) and the forecast is considered to be in-phase. Moreover, it is equivalent to a head-or-tails experiment (POCID ≈ 50%). Finally, it is better then a model which simply predicts the mean of the TS (ARV < 1).

8. Conclusions

In this paper, experiments were realized on real world TS forecasting problem with the new non-standard QNN learning scheme. The results show that QNN performance was better than those obtained with ANN constructed in another work [Ferreira, 2006] for some measures and better than linear algebraic models ARIMA in overall, notably for DJIA series. This is encouraging, since stock market prediction is a difficult problem.

Furthermore, traces of an automatic phase adjustment mechanism could be observed on DJIA results, especially for NMSE measure bearing 0,78 which corresponds to a significant performance improvement over ANN model (≈ 20% better). This can be due to quantum phase learning abilities of the QNN model. However, more experiments with the proposed quantum model are needed in order to confirm this behavior within other real world financial TS which tends to out-of-phase predictions.

Stated thus, it is suggested a further investigation of a probabilistic interpretation of QNN output and its influence on the quality of the predictor. Hence, the state $|\varphi\rangle$ of a qubit neuron should be represented by its complex-valued output y_k, given by

$$y_k = e^{i\theta_k} = \cos\theta_k + i \cdot \sin\theta_k,\tag{22}$$

which stores the quantum state phase information θ_k, as well as the base amplitudes ($\alpha = \cos\theta_k$ and $\beta = \sin\theta_k$). The real output Y_R must be given by

$$Y_R = \left|\beta\right|^2 = \left|\sin\theta_k\right|^2,\tag{23}$$

representing the probability of the state $|1\rangle$. In addition, the inverse cumulative distribution function would be responsible for the correct prediction, according to $\left|\beta\right|^2$, supposing a normal distribution for the TS with estimated mean $\bar{x}$ and estimated standard deviation σ. In this process, it can be necessary a convolution mechanism in order to eliminate eventual heteroscedasticity on TS data set.

References

Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, pages A400:97-117.

Feynman, R. (1982) Simulating physics with computers. In: *International Journal of Theoretical Physics*, pages 21:67-488.

Ferreira, T. A. E., Vasconcelos, G. C. and Adeodato, P. J. L. (2004). A hybrid intelligent system approach for improving the prediction of real world time series. *IEEE* In *Proceedings of the Congress on Evolutionary Computation*, pages 736-743. IEEE.

Anais do XXVII Congresso da SBC
ENIA • VI Encontro Nacional de Inteligência Artificial
30 de junho a 06 de julho de 2007
Rio de Janeiro, RJ

Ferreira, T. A. E., Vasconcelos, G. C. and Adeodato, P. J. L. (2005). A new hybrid approach for enhanced time series prediction. In *Anais do XXV Congresso da Sociedade Brasileira de Computação*, pages 831-840. SBC.

Ferreira, T. A. E. (2006). Uma nova metodologia híbrida inteligente para a previsão de séries temporais. Thesis (Ph.D.), Centro de Informática UFPE.

Fuller, W. A. (1976), Introduction to statistical time series, John Wiley & Sons, 2nd edition.

Kouda, N. Matsui, N. and Nishimura, H. (2004). A multilayered feedforward network based on qubit neuron model. In *Systems and Computers in Japan*, pages 35(13):43-51.

Kouda, N., Matsui, N., Nishimura, H. and Peper, F. (2005). An examination of qubit neural network in controlling an inverted pendulum. In *Neural Processing Letters*, pages 22(3):277-290, Springer Netherlands.

Lo, A. W. and MacKinlay, A. C. (2002). A non-random walk down Wall Street, Princeton University Press, 6th edition.

Malkiel, B. G. (1973). A random walk down Wall Street, W. W. Norton & Company Inc., 6th edition.

Matsui, N, Takai, M. and Nishimura, H. (2000). A network model based on qubit-like neuron corresponding to quantum circuit. In *The Institute of Electronics Information and Communications in Japan (Part III: Fundamental Electronic Science)*, pages 83(10):67-73.

Mitrpanont, J. L. and Srisuphab, A. (2002). The realization of quantum complex-valued backpropagation neural network for pattern recognition problem. In *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on*, pages 1:462-466.

Nielsen, M. A. and Chuang, I. L. (2005). Computação quântica e informação quântica, Bookman, 1ª edição.

Nitta, T. (1994). Structure of learning in the complex numbered back-propagation network. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence, 1994 IEEE International Conference on*, pages 1:269-274.

Percival, D. B. and Walden, A. T. (1998). Spectral analysis for physical applications – multipaper and conventional univariate techniques, Cambridge University Press.

Prechelt, L. (1994). Proben1: a set of neural network benchmark problems and benchmarking rules. Technical Report 21/94. Fakultat fur Informatik, Karlsruhe.

Prechelt, L. (1998) Automatic early stopping using cross-validation: quantifying the criteria. In *Neural Networks,* pages 11(4):761-767.

Shor, P. W. (1994) Algorithms for quantum computation: discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings, 35th Annual Symposium on*, (ed. S. Goldwasser) p. 124-134. IEEE Computer Society Press.

Sitte, R. and Sitte, J. (2002) Neural networks approach to the random walk dilemma of financial time series. *Applied Intelligence*, 16(3):163-171.