TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 16, No. 3, December 2015, pp. 423 ~ 430
DOI: 10.11591/telkomnika.v16i3.9375

Layer Recurrent Neural Network Based Power System Load Forecasting

Nikita Mittal[1]*, Akash Saxena[2]
[1]Department of Electrical Engineering, Yagyavalkya Institute of Technology
[2]Swami Keshvanand Institute of Technology,Jaipur, India-302017
*Corresponding author, e-mail-er.nikitamittal@gmail.com

Abstract

This paper presents a straight forward application of Layer Recurrent Neural Network (LRNN) to predict the load of a large distribution network. Short term load forecasting provides important information about the system's load pattern, which is a premier requirement in planning periodical operations and facility expansion. Approximation of data patterns for forecasting is not an easy task to perform. In past, various approaches have been applied for forecasting. In this work application of LRNN is explored. The results of proposed architecture are compared with other conventional topologies of neural networks on the basis of Root Mean Square of Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE). It is observed that the results obtained from LRNN are comparatively more significant.

Keywords: *artificial neural network, electricity load forecasting, layer recurrent neural network, linear regression, short term load forecasting*

1. Introduction

Load is a device or a set of device, which are consuming energy from the power system networks. Consumption of energy varies with respect to time because the pattern of usage of electricity by consumers cannot be controlled [1]. Forecasting of load is an important and challenging task due to its non-smooth behavior. Load forecasting helps the power dispatching department to accurately and conveniently generate electricity and helps in cost savings. Short Term Load Forecasting (STLF) is an important tool for cost savings and it helps to maintain the continuity of electricity supply [2]. STLF is important when load patterns are required to be predicted in advance. Customers are given incentives to modify their usage pattern to avoid the usage of energy at peak hours which can help to reduce the burden on electricity utilities [3].

Load forecasting can be broadly divided into three categories: short term forecasts which are usually from one hour to one week, medium-term forecasts which are usually from a week to a year, and long-term forecasts which are longer than a year [4]. In order to meet load requirements and cost efficiency, accurate load forecasting is necessary. Underestimation of load may lead to breakdown of power system network due to stability problems while overestimation leads to starting of extra generating units and a cost inefficient system. Also in smart prepaid meters, units of energy needs to be purchased in advance, so it is required to determine the accurate value of daily consumption of units in advance through load forecasting [5]. For accurate forecasting we need to know the features which affect the load pattern. These features may be temperature and other weather conditions like humidity, price of electricity, typ e of area (commercial, industrial and domestic) or type of consumers. Feature selection is the process of selecting a set of representative features that are relevant and sufficient for building a prediction model. Appropriate feature selection improves the accuracy of prediction (forecasting) models [6]. Many load forecasting techniques have been proposed and applied to forecasting models successfully. Broadly they can be classified into conventional and advanced learning (artificial intelligence) techniques. Conventional methods are based on the relationship between load and factors affecting load. These methods are simple but erroneous due to non-linear relationship between load and factors affecting load [9].

Nowadays vector machine model theory [6], artificial neural network, support vector machine model are been used for short term load forecasting. In [7] author presented self-

Received July 21, 2015; Revised October 19, 2015; Accepted November 9, 2015

similarity theory combined with fractal interpolation theory to short term power load forecasting. Various other techniques are used for load forecasting, they can be classified as traditional techniques, classical techniques or hybrid of both. Artificial neural networks are used in load forecasting due to their capability to model non-linear mapping relations between inputs and outputs and they can learn from a set of examples [8].

STLF is a potential area of research and various techniques have been employed by researchers to forecast the load [4-9]. A brief survey of literature is presented below to establish relevance of work presented in this paper. In Linear regression method, functional clustering procedure is used to classify daily load curves and then a family of functional linear regression models is defined. For forecasting, a new load curves is assigned to clusters, applying a functional discriminant analysis [10]. Kalman filter is used to estimate the load model parameters. Models proposed in conjunction with kalman filters estimation consider either the dependence of the load on the weather or on the previous load as a time series autoregressive models.Hybrid models can also be used to express the load as a combination of both to predict future loads [11]. Fuzzy expert systems can incorporate a set of IF-THEN rules and expert's opinion. Historical data are converted into fuzzy information and then forecasting is performed. Fuzzy expert systems can give results with high accuracy [12, 13].

Support vector machines (SVM) generate a model which will predict unknown output based on known input [21]. SVM's are based on principle of structural risk minimization which is used in neural networks [6]. In [14] author have applied back-propagation learning algorithm to train ANN for forecasting time series.In [15] author has demonstrated that ANN can be applied in STLF with accepted accuracy. Dillon et al. [16] used adaptive pattern recognition and self-organizing techniques for STLF.

Different techniques namely; regression, multiple regression, exponential smoothing, iterative reweighted least squares, adaptive load forecasting, stochastic time series autoregressive, ARMA model [17], ARIMA model [18], support vector machine [19], soft computing based models- genetic algorithms, fuzzy logic, neural networks and knowledge based expert systems etc. have been applied to load forecasting. It is concluded that demand forecasting techniques based on soft computing methods are gaining major advantages for their effective use [20].

In this paper we present STLF using LRNN.LRNN contains at least one feedback connection, so the activations can flow round in a loop.This enables the network to do temporal processing and learning sequences.It is observed that LRNN leads to a reduction in a forecasting error. Levenberg-Marquardt algorithm is used to train the network which is the most widely used optimization algorithm.

This paper is organized as follows: section 2 presents an overview of load forecasting using ANN and Levenberg-Marquardt algorithm is explained.Section 3 presents the Problem formulation and evaluation criterion. Section 4 presents the results and analysis. Section 5 presents the conclusion of the work.

2. Artificial Neural Network

The artificial neural network was invented in1958 by psychologist Frank Rosenblatt [22]. ANN operates by creating many different processing elements, each analogous to a single neuron in a biological brain. The ANNs are trained by adapting a network and comparing the output obtained with the input training and target data. The training is carried out to match the network output to target data. The ANN consists of an input layer and an output layer. The layer in between these two layers is hidden layer. The neural networks use weights for each input variable and a bias that act as a threshold to produce outputs. A sigmoid function is chosen as a function that calculates output using weights and biases as it shows a great similarity to real neurons.

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{1}$$

Let the input vector at layer 1 be denoted by *x.* Let $w^{1}_{j,k}$ denote the weight for connection from the kth neuron in the (l-1)th layer to the jth neuron in the lth layer. Let the total number of neurons in the (l-1)th layer and lth layers be K and J, respectively. Let L be the total number of layers, b_j^1 denote the bias of the jth neuron in the lth layer. It is denoted as:

$$= \sigma\left(\sum_k w_{j,k}^{1} a_k^{l-1} + b_j^{1} \right) , \qquad \text{j=1,……,}J \qquad (2)$$

Given the last layer, the activation function for the jth neuron is computed as:

$$a_j^L = \rho\left(\sum_k w_{j,k}^{L} a_k^{L-1} + b_j^{L} \right), \qquad \text{j=1,……..,}J \qquad (3)$$

Where $\rho(z) = z$ is a linear function. The weight $w_{j,k}^l$ is an entry of the weight matrix Wl defined as:

$$W^l = \begin{bmatrix} w_{1,1}^l & w_{1,2}^l & \cdot & \cdot & \mathrm{w}_{1,k}^{\mathrm{l}} \\ w_{2,1}^l & \mathrm{w}_{2,2}^{\mathrm{l}} & \cdot & \cdot & \mathrm{w}_{2,k}^{\mathrm{l}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ w_{J,1}^l & \mathrm{w}_{J,2}^{\mathrm{l}} & \cdot & \cdot & \mathrm{w}_{J,K}^{\mathrm{l}} \end{bmatrix}_{J \times K} \qquad (4)$$

Where a_j^1 and b_j^1 are entries of the vectors defined as:

$$a^l = [a_1^l, \mathrm{a}_2^{\mathrm{l}}, \ldots\ldots.., \ \mathrm{a}_{\mathrm{J}}^{\mathrm{l}}]$$

$$b^l = \left[b_1^l, \mathrm{b}_2^{\mathrm{l}}, \ldots\ldots\ldots \mathrm{b}_{\mathrm{J}}^{\mathrm{l}}\right] \qquad (5)$$

This activation function is the output of the neural networks. We choose sigmoid and linear functions. The reason is that a network consisting of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function having a finite number of discontinuities. The neural network finds weights and biases by minimizing the following cost function:

$$C = \frac{1}{2} \sum_t \left(y_t - a_t^L \right)^2 \qquad (6)$$

Where y_t is the required known output at instant t, and a_t^L is the output from the final Lth layer at instance t.

2.1. Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm is the most widely used optimization algorithm [23]. It outperforms simple gradient descent and other conjugate gradient methods in a wide variety of problems. In fitting a function y'(t;**p**) of an independent variable and a vector of n parameters **p** to a set of m data points (t_i ,y_i), it is customary and convenient to minimize the sum of the weighted squares of the errors between the measured data y(t_i) and the curve-fit function y'(t_i ;**p**). This scalar-valued goodness-of-fit measure is called the chi-squared error criterion.

$$X^2(\mathbf{p}) = \sum_{i=1}^{m}\left[\frac{y(t_i) - y'(t_i;\mathbf{p})}{w_i}\right]^2 \tag{7}$$

$$= (y - y'(\mathbf{p}))^T W (y - y'(\mathbf{p}))$$

$$= y^T W y - 2y^T W y' + y'^T W y' \tag{8}$$

The value w_i is a measure of the error in measurement *y(t$_i$)*

This algorithm adaptively varies the parameter updates between the gradient descent update and the gauss newton update,

$$\left[J^T W J + \lambda I\right] h_{lm} = J^T W (y - y') \tag{9}$$

Where a small values of the algorithmic parameter λ result in a Gauss-Newton update and large values of λ result in a gradient descent update. The parameter λ is initiated to be large so that first updates are small steps in a steepest-descent direction. If an iteration happens to result in a worse approximation, λ is increased. As the solution improves, λ is decreased, the Levenberg-Marquardt method approaches the Gauss-Newton method, and the solution typically accelerates to the local minimum. Marquardt's suggested update relationship makes the effects of the particular values of λ less problem specific, and is used in the Levenberg-Marquardt algorithm implemented in the MATLAB function.

$$\left[J^T W J + \lambda \, diag\left(J^T W J\right)\right] h_{lm} = J^T W (y - y') \tag{10}$$

3. Preliminaries

3.1. Problem Formulation

In this work, the load forecasting in the order of a few minutes is under consideration. The basic prediction can be expressed in a time-series model, in which the future load exclusively relies on the historical data of load.In the modeling process, the data is selected from a distributed grid. A set of 1000 data points is selected for training and another set of 1000 data points is usedfor model validation. Number of layers in LRNN is 2 and data division is random.Training algorithm used in LRNN is Levenberg-Marquardt algorithm. Number of neurons in first layer is 10. Figure 1 depicts some portions of the training and validation datasets.

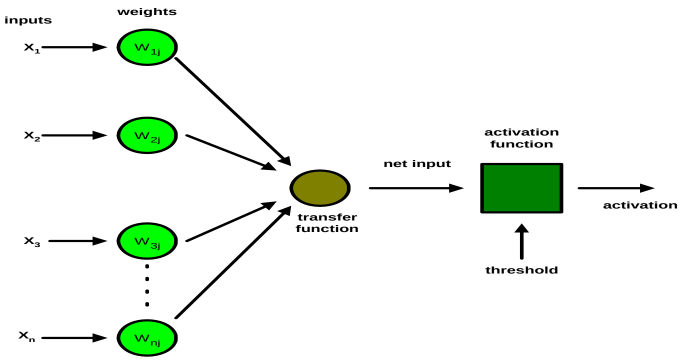

Figure 1. Architecture of Layer Recurrent Neural Network

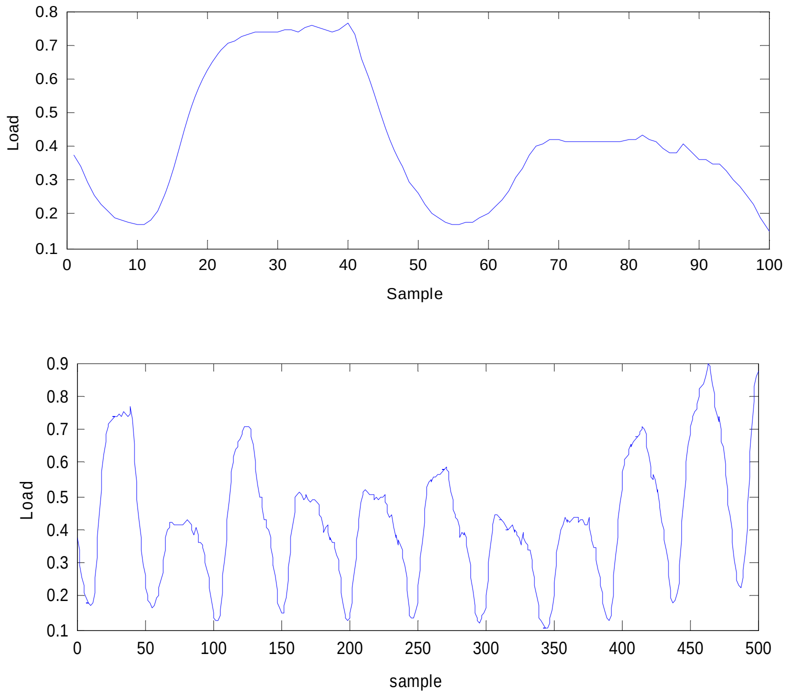

Figure 2. Load pattern selected for (a) training; (b) validation

3.2. The Evaluation Criteria

For evaluating the performance of the model for load forecasting, define errors (e), the root mean square errors (RMSE), the mean absolute errors (MAE), and the mean absolute percentage errors (MAPE) as:

$$e = \frac{[y_{act}(i) - y_{pre}(i)]}{y_{act}(i)} \tag{11}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{act}(i) - y_{pre}(i)\right)^2} \tag{12}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left| y_{act}(i) - y_{pre}(i)\right| \tag{13}$$

$$MAPE = \frac{1}{N}\left|\frac{[y_{act}(i) - y_{pre}(i)]}{y_{act}(i)}\right| \times 100 \ \% \tag{14}$$

Where N=1000 is the number of validation data, y_{act} is the real output and y_{pre} is the predicted output in this paper.Comparisons were made among the FFNN and the LRNN with the same input output data shown in Figure 1.

4. Result and Analysis

Load is forecasted with the help of five input features i.e. dry bulb, dew point, wet bulb, humidity and electricity price. Rich data patterns of hourly load of 500 days are taken. All the input and output data is normalized in the range of (0.1-0.9) to avoid convergence problem. Load is forecasted with the help of two topologies of neural networks i.e. FFNN and LRNN.

Figure 3 shows the model output and the plant output. It is observed from the Figure 3(a) that LRNN model gives the best possible match between the actual load data and predicted load data. Figure 3(b) shows the model output and plant output for FFNN model which is comparatively less accurate. Figure 4 shows the distribution of errors obtained from the forecasted results. It can be seen that FFNN has small error variance, and the errors obtained by using LRNN has lowest variances the plot of error distribution using LRNN is narrower.

Table 1 shows the error indices for the LRNN and FFNN models. It is seen that LRNN has the least value of errors i.e. MAE is 0.0795, RMSE is 0.1059 and MAPE is 0.2261. For FFNN the value of errors is slightly more i.e. MAE is 0.0871, RMSE is 0.1169 and MAPE is 0.2339. From Table 1 it is clear that the accuracy of LRNN model is best, and accuracy of FFNN is little lower than LRNN.

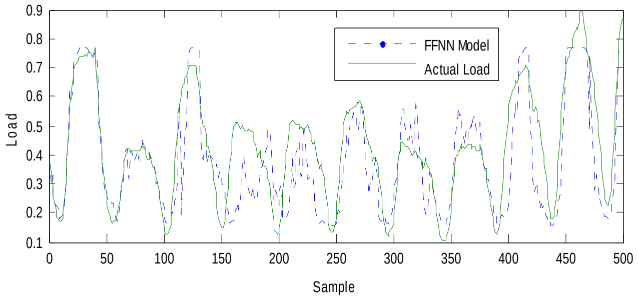

Figure 3(a). Comparison of Actual Load and Forecasted load by LRNN Model

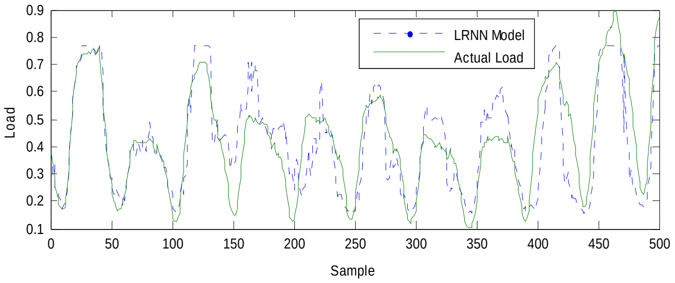

Figure 3(b). Comparison of Actual Load and Forecasted load by FFNN Model

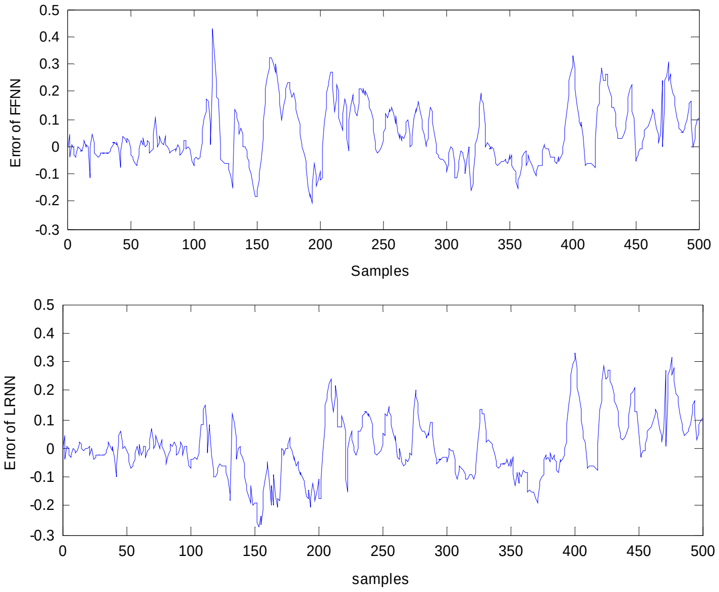

Figure 4. Errors between target data and predicted model output

To compare the efficacy of the networks various indices are defined in section 3.2. Table 1 shows the values and fig. 5 shows the pictorial representation of the indices.

Table 1. The comparison of error indices.

	MAE	RMSE	MAPE
FFNN	0.0871	0.1169	0.2339
LRNN	0.0795	0.1059	0.2261

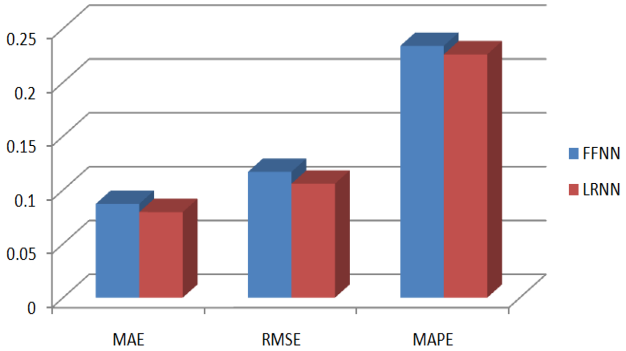

Figure 5. Errors with FFNN and LRNN model

5. Conclusion

This paper presents an application of LRNN in hourly forecast of the load of a large distribution power network. Five significant features have been chosen for training the neural net.The results obtained from LRNN are significant and tested with four statistical parametric tests MAE, MAPE and RMSE. Results of proposed architecture is compared with FFNN technique. It is observed that the LRNN shows promising results as obtained errors in a very narrow margin and trained network possess higher regression results.

References

[1] AS Khwaja, M Naeem, A Anpalagan, A Venetsanopoulos, B Venkatesh. Improved short-term load forecasting using bagged neural networks. *Electric Power Systems Research.* 2015; 125: 109-115.

[2] L Hernandez, C Baladron, J Aguiar, B Carro, A Sanchez-Esguevillas, J Lloret, J Massana. A survey on electric power demand forecasting: future trends insmart grids, microgrids and smart buildings. *IEEE Commun. Surveys Tutor.* 2014; 16(3): 1460-1495.

[3] I Fernandez, C Borges, Y Penya. *Efficient building load forecasting*. IEEE 16th Conference on Emerging Technologies and Factory Automation. 2011: 1-8.

[4] SC Chan, KM Tsui, HC Wu, Y Hou, YC Wu, FF Wu. *Load/price forecasting and managing demand response for smart grids.* IEEE Signal Proc. Mag. 2012: 68-85.

[5] P Day, M Fabian, D Noble, G Ruwisch, R Spencer, J Stevenson, R Thoppay. *Residential power load forecasting.* Proc. Comput. Sci. 2014; 28: 457-464.

[6] SR Gunn. *Support Vector Machines for Classification and Regression.* Technical Report, Image Speech and Intelligent Systems Research Group. University of Southampton. 1998.

[7] Ming-Yue Zhai. A new method for short-term load forecasting based on fractal interpolation and wavelet analysis. *Electrical Power & Energy Systems.* 2015; 69: 241-245.

[8] Chin Wang Lou, Ming Chui Dong. A novel random fuzzy neural network for tackling uncertainties of electric load forecasting. *Electrical Power and energy systems.* 2015; 73: 34-44.

[9] T Senjyu, H Takara, K Uezato, T Funabashi. One-hour-ahead load forecasting using neural network. *IEEE Trans. Power Syst*. 2002; 17(1): 113-118.

[10] Goia A, May C, Fusai G. Functional clustering and linear regression for peak load forecasting. *International journal of forecasting*. 2010; 26(4): 700-711.

[11] Al-Hamadi H, Soliman S. Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model. *Electr Pow Syst Res.* 2004; 68(1): 47-59.

[12] D Ranaweera, N Hubele, G Karady. Fuzzy logic for short term load forecasting*. Int. J. Electr. Power Energy Syst*. 1996; 18(4): 215-222.

[13] Kiartzis S, Bakirtzis A, Theocharis J, Tsagas G. *A fuzzy expert system for peak load forecasting, Application to the greek power system*. 10th Electrotechnical conference. 2000; 3: 1097-1100.

[14] DE Rumelhart, GE Hinton, RJ Willams. *Learning internal representation by error propagation*. Parallel Distributed Processing. Cambridge. 1986; 1: 318-362.

[15] KY Lee, YT Cha, JH Park. *Artificial neural network methodology for short-term load forecasting*. NSF Workshop on Artificial Neural Network Methodology in Power System Engineering. Clemson University.1990.

[16] Dillon TS, Morsztyn K, Phua K. *Short term load forecasting using adaptive pattern recognition and self-organizing techniques*. Proceedings Fifth World Power System Computation Conference. Cambridge. 1975; 2(4/3): 1-15.

[17] Pappas S, Ekonomou L, Karamousantas D, Chatzarakis G, Katsikas S, Liatsis P. Electricity demand loads modeling using autoregressive moving average (ARMA) models. *Energy*. 2008; 33(9): 1353-1360.

[18] Lee CM, Ko CN. Short-term load forecasting using lifting scheme and {ARIMA} models. *Expert Systems with Applications*. 2011; 38(5): 5902-5911.

[19] Cheng Ting Lin, Li Der Chou. A novel economy reflecting short-term load forecasting approach*. Energy conservation and management.* 2013; 65: 331-342.

[20] Arunesh Kumar Singh, Ibraheem, S Khatoon, Md Muazzam, DK Chaturvedi. *Load forecasting techniques and methodologies: A review.* 2nd power control and embedded systems conference. 2012: 1-10.

[21] Feng Lv, Fengning Kang, Hao Sun. The Predictive Method of Power Load Based on SVM. *Telekomonika Indonesian Journal of Electrical Engineering.* 2014; 12(4): 3068-3077.

[22] Rosenblatt F. The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. *Psychological Review.* 65(6): 386-408.