*II. PREVIOUS WORK*

Many researcher put their best effort in analyzing the solution to the problem that how many neurons are kept in hidden layer in order to get the best result, but unfortunately no body succeed in finding the optimal formula for calculating the number of neurons that should be kept in the hidden layer so that the neural network training time can be reduced and also accuracy in determining target output can be increased. Basically when dealing with the number of neurons in the input layer, one has to analyze about the data which is trained. For example, while dealing with handwritten numeral recognition using neural network for pin code recognition [5], the box size in which the pin number is written is taken under consideration for determining the number of neurons in the input layer. If the box size is 10x15 then 150 neurons must be taken as a input layer, so that every pixel contribute its value to the input layer individually. When the output layer is considered, then it depends on the chosen model configuration. For example, in this paper we are concentrating on the recognition of pin number in any scripting language, so 16 neurons are taken in output layer so that it can be mapped to Unicode which has 16 bit for identification. If the neural network is a regressor, then the output layer has a single node.

Now the number of neurons in the intermediate layer is taken under consideration which is the soul purpose of this paper. Before that one fact should be kept in mind that if the data is linearly separable then there is not at all any use of hidden layer. Linear Activation function can be directly implemented on the input and output layer. One hidden layer will be used when any function that contains a continuous mapping from one finite space to another. Two hidden layer can represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy [14].

When Multilayer Back Propagation Neural Network is under consideration then the number of hidden layers and approximation of neurons in each hidden layer need to be calculated. Since every neural network architecture is based on the training data samples (i.e. application specific), so in this paper, training data is taken as handwritten pin code samples of different scripting language and two hidden layers are used in order get more accuracy in recognition. Unnecessary increasing hidden layer may causes increase in the complexity of network. This is because in the weight balancing stage, weights between the first hidden layer and second hidden layer are also considered for weight updating which depends upon the error gap between the actual and target output.

许多研究人员尽最大努力来分析隐藏层中保留多少个神经元以获得最佳结果的问题的解决方案，但遗憾的是没有人成功地找到计算应保留的神经元数量的最佳公式 在隐藏层中，可以减少神经网络的训练时间，并且可以提高确定目标输出的准确性。 基本上，在处理输入层中的神经元数量时，必须分析经过训练的数据。 例如，在使用神经网络进行 pin 码识别 [5] 处理手写数字识别时，会考虑写入 pin 号的框大小来确定输入层中的神经元数量。 如果盒子大小为 10x15，则必须将 150 个神经元作为输入层，以便每个像素单独向输入层贡献其值。 当考虑输出层时，它取决于所选的模型配置。 例如，在本文中，我们专注于任何脚本语言中的 pin 号码识别，因此在输出层中采用 16 个神经元，以便可以将其映射到具有 16 位的 Unicode 进行识别。 如果神经网络是回归器，则输出层只有一个节点。

现在考虑中间层神经元的数量，这是本文的灵魂目的。 在此之前，应该记住一个事实，如果数据是线性可分的，那么根本不需要使用隐藏层。 线性激活函数可以直接在输入和输出层实现。 当任何函数包含从一个有限空间到另一个有限空间的连续映射时，将使用一个隐藏层。 两个隐藏层可以使用有理激活函数将任意决策边界表示为任意精度，并且可以将任何平滑映射近似为任意精度[14]。

当考虑多层反向传播神经网络时，需要计算隐藏层的数量和每个隐藏层中神经元的近似值。 由于每个神经网络架构都是基于训练数据样本（即特定于应用程序），因此在本文中，训练数据被视为不同脚本语言的手写 pin 码样本，并使用两个隐藏层以获得更高的识别精度。 不必要的增加隐藏层可能会导致网络复杂度的增加。 这是因为在权重平衡阶段，还考虑第一隐藏层和第二隐藏层之间的权重进行权重更新，这取决于实际输出和目标输出之间的误差差距。

*III. DETERMINING HIDDEN LAYERS AND HIDDEN NODES*

Firstly, number of hidden nodes approximation for the whole neural network is important and afterwards the decision of selecting the number of layers as well as proportion of neurons between the first and second hidden layer is required. Usually some rule-of-thumb methods are used for determining the number of neurons in the hidden nodes.

 The number of hidden layer neurons are 2/3 (or 70% to 90%) of the size of the input layer. If this is insufficient then number of output layer neurons can be added later on.[1]

 The number of hidden layer neurons should be less than twice of the number of neurons in input layer. [2]

 The size of the hidden layer neurons is between the input layer size and the output layer size.[6]

But the above three myths are not considered to be true always because not only the input layer and the output layer decides the size of the hidden layer neurons but also the complexity of the activation function applied on the neurons, the neural network architecture, the training algorithm and most important the training samples database on which the neural network is designed to execute.

Multiple hidden layers are used in the applications where accuracy is the criteria and no limit for the training time is mentioned. Even the drawback of using multiple hidden layers in the neural network is that they are more prone to fall in bad local minima [9].

Also, experimental results in this paper shows that the number of neurons in first and second hidden layer should be kept nearly equal so that the network can be trained easily. Usually samples similar to data which contain discontinuities, such as a saw tooth waveform requires multiple hidden layers.

An “structured trial and error” method is used by maximum developer for creating a neural network’s layer approximation. As usual, input layer neurons are estimated depending upon the data samples which is to be trained. For example, if 15x10 pixel block of data is taken as sample then 150 neurons are taken as input layer. Similarly output layer neurons can be calculated depending upon the number of types of output. In this approach, 16 neurons are taken in output layer because Unicode scheme is used. For hidden layer, initially few random numbers are neurons are taken and samples are allowed to train on it. If the network fails to converge after a reasonable period, restart training up to five times, so that it is assured that it has not fallen into a local minima. If the network still fails to converge then add few more neurons in the layer and allow it to train. Still no better results found then it is necessary to add second hidden layer. Note that first and second hidden layer must contain equal number of neurons in order to train the network easily. In order to get best approximation of the hidden layer neurons, equal amount of number of neurons in both hidden layers can be reduced and again training is done so that one can check whether the network converges to the same solution even after reducing the number of hidden layer neurons.

Yinyin Liu, Janusz A. Starzyk, Zhen Zhu [9] in their approach to optimize the number of neurons in the hidden layer using benchmark datasets and estimation of the signal-tonoise- ratio figure. The method utilizes a quantitative criterion based on the SNRF to detect overfitting automatically using the training error only, and it does not require a separate validation or testing set. This method reduces the overfitting problem in the network.

首先，整个神经网络的隐藏节点数量近似很重要，然后需要决定选择层数以及第一和第二隐藏层之间的神经元比例。 通常使用一些经验法则来确定隐藏节点中神经元的数量。

隐藏层神经元的数量是输入层大小的 2/3（或 70% 到 90%）。 如果这还不够，则可以稍后添加输出层神经元的数量。[1]

 隐藏层神经元的数量应小于输入层神经元数量的两倍。 [2]

 隐藏层神经元的大小介于输入层大小和输出层大小之间。[6]

但上述三个神话并不总是正确的，因为不仅输入层和输出层决定了隐藏层神经元的大小，而且还决定了应用于神经元的激活函数的复杂性、神经网络架构、训练 算法，最重要的是神经网络设计用来执行的训练样本数据库。

在以准确性为标准并且没有提到训练时间限制的应用中使用多个隐藏层。 即使在神经网络中使用多个隐藏层的缺点是它们更容易陷入不良的局部最小值[9]。

此外，本文的实验结果表明，第一和第二隐藏层中的神经元数量应保持几乎相等，以便网络可以轻松训练。 通常，样本类似于包含不连续性的数据，例如锯齿波形，需要多个隐藏层。

大多数开发人员使用“结构化试错”方法来创建神经网络的层近似。 与往常一样，输入层神经元根据要训练的数据样本进行估计。 例如，如果将 15x10 像素的数据块作为样本，则将 150 个神经元作为输入层。 类似地，可以根据输出类型的数量来计算输出层神经元。 在这种方法中，由于使用了 Unicode 方案，因此输出层采用 16 个神经元。 对于隐藏层，最初采用神经元的几个随机数，并允许对其进行样本训练。 如果网络在合理的时间后未能收敛，请重新启动训练最多五次，以确保网络没有陷入局部最小值。 如果网络仍然无法收敛，则在该层中添加更多神经元并允许其训练。 仍然没有找到更好的结果，那么有必要添加第二个隐藏层。 请注意，第一和第二隐藏层必须包含相同数量的神经元，以便轻松训练网络。 为了获得隐藏层神经元的最佳近似，可以减少两个隐藏层中相同数量的神经元，并再次进行训练，以便即使在减少隐藏层的数量后，也可以检查网络是否收敛到相同的解决方案 层神经元。

Yinyin Liu、Janusz A. Starzyk、Zhen Zhu [9] 使用基准数据集和信噪比图的估计来优化隐藏层中神经元的数量。 该方法利用基于 SNRF 的定量标准，仅使用训练误差自动检测过拟合，并且不需要单独的验证或测试集。 这种方法减少了网络中的过拟合问题。

Rivals I. & Personnaz L.[7] used techniques based on least squares estimation and statistical tests for estimating the number of neurons in the hidden layer. Two approaches were used phase wise , first the bottom-up phase in which initially the number of neurons are increased up to an extent till the network becomes ill-conditioned. Afterwards in the next phase i.e. the top-down approach the neural network has to go through statistical Fisher tests. The second phase usually reduces the complexity of the neural network.

F. Fnaiech in [8] make an attempt to prune the hidden nodes of the feed forward architecture by initially creating a non linear activation function of hidden nodes as Taylor’s expansion and then NARX (nonlinear auto regressive with exogenous input) model is used. Afterwards nonlinear order selection algorithm proposed by Kortmann-Unbehauen most relevant signal of the NARX are selected and finally BPNN algorithm is used in order to prune the hidden nodes.

Stuti Asthana and Rakesh K Bhujade in [5] made an approach to use two hidden layer for recognizing multiscript number recognition using artificial neural network on postcard, keeping accuracy as a chief criteria. More than 95% accuracy was obtained due to the use of two hidden layer. Moreover equal number of neurons are used in both layers so as to get best accuracy. This work has been tested on five different popular Indian scripts namely Hindi, Urdu, Tamil ,English and Telugu and satisfactory results were obtained under ideal condition.

The decision method for selecting number of neurons to getting optimal solution using two phase method is describe in [3] by Kazuhiro Shin-ike . In two phase method the termination condition is same as the trial and error method but in new approach dataset is divided into four groups in which two groups of data are used in first phase to train the network and one group of remaining data set is used in second phase to train the network and last group of data set is used predict the output values of the trained network and this experiment is repeated for different number of neurons to get the minimum number of error terms for selecting the number of neurons in the hidden layer. Using this approach correctness rate is achieve 87.5 percent.

对手 I. & Personnaz L.[7] 使用基于最小二乘估计和统计测试的技术来估计隐藏层中的神经元数量。 分阶段使用了两种方法，第一个是自下而上阶段，其中最初神经元的数量增加到一定程度，直到网络变得病态。 之后，在下一阶段，即自上而下的方法中，神经网络必须经过统计费希尔测试。 第二阶段通常会降低神经网络的复杂性。

F. Fnaiech 在[8]中尝试通过最初创建隐藏节点的非线性激活函数作为泰勒展开式，然后使用 NARX（外生输入非线性自回归）模型来修剪前馈架构的隐藏节点。 然后采用Kortmann-Unbehauen提出的非线性阶次选择算法选择NARX最相关的信号，最后使用BPNN算法来修剪隐藏节点。

Stuti Asthana 和 Rakesh K Bhujade 在 [5] 中提出了一种使用两个隐藏层的方法，使用人工神经网络在明信片上进行多文字数字识别，并将准确性作为主要标准。 由于使用了两个隐藏层，获得了超过 95% 的准确率。 此外，两层使用相同数量的神经元，以获得最佳精度。 该作品在印地语、乌尔都语、泰米尔语、英语和泰卢固语五种不同的流行印度文字上进行了测试，在理想条件下获得了令人满意的结果。

Kazuhiro Shin-ike 在 [3] 中描述了使用两阶段方法选择神经元数量以获得最佳解决方案的决策方法。 在两阶段方法中，终止条件与试错法相同，但在新方法中，数据集被分为四组，其中两组数据用于第一阶段训练网络，一组剩余数据集用于第二阶段 第二阶段训练网络，最后一组数据集用于预测训练网络的输出值，并对不同数量的神经元重复该实验，以获得选择隐藏层神经元数量的误差项的最小数量 。 使用该方法正确率达到87.5%。