Polynomial Regression by Comprehensive Network

Caleb Princewill Nwokocha [1]

Abstract

The aim of this paper is to introduce a simple, applicable, and efficient machine learning theory of comprehensive function, comprehensive node, comprehensive layer, and comprehensive network. A comprehensive network or ***N*-model** has multiple layers, each layer has one or multiple nodes, and each node has one comprehensive function. Section 4 describes N-model optimization method, that an N-model optimizes by optimizing each node, and each node optimizes its thesis by simply using dynamic programming to update its error mean. A multitasking N-model was applied in linear, nonlinear, and logistic regression experiments to predict real value y given real value x. The multitasking N-model passed 20 test after learning 8 examples.

1. Introduction

In the real world, systems of linear, nonlinear, and logistic regressions are used to solve prediction problems. Assuming Bob has old data of patients' ages and their systolic blood pressures. With these data, Bob may wish to predict systolic blood pressure of new patients at certain ages. Table 1 shows the data Bob has for solving his prediction problem by linear regression.

Table 1. Systolic Blood Pressure (SBP) by age.

SBP(y)	2	27	20	31	26	37	40	30	48	47
Age(x)	35	37	42	46	53	57	61	65	69	74

Some other prediction problems can be solved by nonlinear regression. Suppose the (x, y) points of table 1 are nonlinear as the following: $(35, y = 22.5)$, $(37, y = 36.9)$, $(42, y = 76.4)$, $(46, y = 111.6)$, $(53, y = 180.9)$, $(57, y = 224.9)$, $(61, y = 272.1)$, $(65, y = 322.5)$, $(69, y = 376.1)$, $(74, y = 447.6)$. Due to this supposition, predicting future values of y is preferably done by nonlinear regression.

There are prediction problems, also called classification problems, that can be solved by logistic regression. Linear and nonlinear regressions are different from logistic regression in the sense that for the former, y is continuous, and for the latter, y is discrete. An example case of discrete y is $(35, y = 0)$, $(37, y = 0)$, $(42, y = 0)$, $(46, y = 0)$, $(53, y = 0)$, $(57, y = 1)$, $(61, y = 1)$, $(65, y = 1)$, $(69, y = 1)$, $(74, y = 1)$. Given the right patient's age, logistic regression returns either 0 or 1 as a classification value of the patient. 0 may represent low systolic blood pressure and 1, otherwise.

Linear, nonlinear, and logistic regressions can be done by using comprehensive network. A **comprehensive network** is a set N consisting of $l > 1$ comprehensive layers; here, $l \in \mathbb{N}$. The symbol l may be called the **length** of an N-model. A **comprehensive layer** is a set L consisting of $n \in \mathbb{N}$ comprehensive nodes. The symbol n may be called the **width** of a comprehensive layer. A **comprehensive node** consist of a comprehensive function. See section 3.1 for definition of comprehensive function.

2. Related Work

Earlier work with linear regression was done by (Legendre, 1805) to predict movement of heavenly bodies. Still in 19th century, (Verhulst, 1845) developed logistic regression for describing population growth. The well-known logistic regression formula $\frac{1}{1+e^{-x}}$ is not used in introductory theory of comprehensive function, node, layer, and network. Although, the 20th century work of (McCulloch & Pitts, 1943; Rosenblatt, 1958) on perceptron, do make use of the well-known logistic regression formula for modern classification problems. Toward the end of the 20th century, work in statistics and econometrics began using nonlinear regression to solve prediction problems that were intractable by linear regression (Jennrich, 1969; Malinvaud, 1970). The formula $f(X, \beta) \sim y = f(X, \beta) + \epsilon$ of nonlinear regression model is similar to formula of comprehensive node introduced in section 3.2.

3. *N*-model

A comprehensive network consist of primary components called comprehensive functions, comprehensive nodes, and comprehensive layers.

3.1. Comprehensive Function

It is possible that some mathematical functions are known to agent A, but not known to agent B. Let A be a businessperson who knows mathematical functions of commerce, and let B be a physicist who knows mathematical functions of physical matter. If there is a set U consisting of every function known to agent A and B, then there are some functions in U that are comprehensive to agent A, but not comprehensive to agent B; vice versa. Therefore, a **comprehensive function** $f_c(\cdot)$ is defined generally as a function that is known within a set of functions.

3.2. Comprehensive Node

According to (Mitchell, 1997) "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." In simple algebraic notation, a computer program is said to learn an **objective** $y \in \mathbb{R}$ with respect to **input** $x \in \mathbb{R}^n$ and **performance measure** $e \in \mathbb{R}$, if its performance given input x, as measured by e, improves with x. While the computer program has not learned y, the program knows a **hypothesis** $h \in \mathbb{R}$ that is different from y. The squared difference between h and y is measured by the performance measure e. The symbol e can also be called **error**.

Let hypothesis h be the output of a comprehensive function:

$$f_c(x) = h. \tag{1}$$

Assuming the error

$$e = (y-h)^2 = (y-h)(y-h), \tag{2}$$

it follows that

$$\frac{e}{y-h} = y - h = \frac{(y-h)^2}{y-h}, \tag{3}$$

$$y = h + \frac{(y-h)^2}{y-h} = h + y - h, \tag{4}$$

$$y = h + \sqrt{e}. \tag{5}$$

Equation (5) simply shows that a computer program will learn an objective if $\sqrt{e}$ is added to h. It is intuitive to reason that by simply adding $\sqrt{e}$, the program will learn objective y. But further analysis revealed that this intuition is not sufficient to enable learning. Further analysis of this intuition is provided in section 4.1.

Learning is limited when using merely a comprehensive node. This limitation is partly caused by the fact that a comprehensive node cannot effectively learn more than one objective. Also, if the input x has too many features, there may not be a comprehensive function that adequately computes all the features. Comprehensive layer helps to remove this limitation.

3.3. Comprehensive Layer

Figure 1 shows a comprehensive layer consisting of two nodes: node A and node B. Observe that node A take the input $x \in \mathbb{R}^n$ and node B take input $z \in \mathbb{R}^n$. The layer input is configured so that each node have different input, but it is possible to allow these two nodes take the same input.

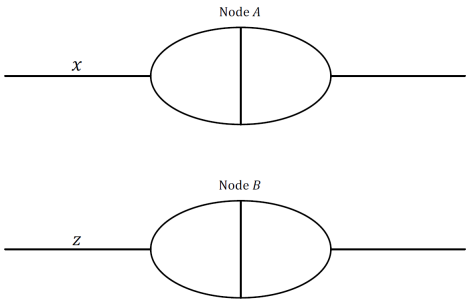

Figure 1. Comprehensive layer.

In some cases where the input q has too many features $q_1, \cdots, q_n$; $n \in \mathbb{N}$, and there is no single node with a comprehensive function that is capable of computing those features, the input features can be divided into subsets for each node in a comprehensive layer. For example, if node A comprehensive function is $f_c(x_1, \cdots, x_u)$ where $u = \frac{n}{2}; u \in \mathbb{N}$, then neither node A nor its comprehensive function can compute all q features. A solution for computing all q features is using a comprehensive layer that consists of node A and node B, and presupposing that node B also has $f_c(x_1, \cdots, x_u)$ as its comprehensive function. In this sense, node A computes one half of q features, and node B computes the other half of the features. Technically, node A comprehensive function computes the features $\{q_1, \cdots, q_u\} \subset q$, and node B comprehensive function computes the features $\{q_{u+1}, \cdots, q_n\} \subset q$.

3.4. Basic N-model

The most basic model of comprehensive layer L consist of only one node, because as L is a set of n nodes: if $n = 1$, then L consist of only one node which is the minimum number of nodes a layer can have; therefore, a layer consisting of only one node is the simplest and most basic comprehensive layer. This proves that a basic comprehensive layer and a single comprehensive node are the same model. Furthermore, a set consisting of at least two basic layers is a basic comprehensive network. This corresponds with the definition: A comprehensive network is a set N consisting of $l > 1$ layers. Hence, if $l = 2$ and the two layers are basic comprehensive layers, then the two layers form the most basic comprehensive network (also called basic N-model). A set with l basic comprehensive layers is a l-layer basic N-model.

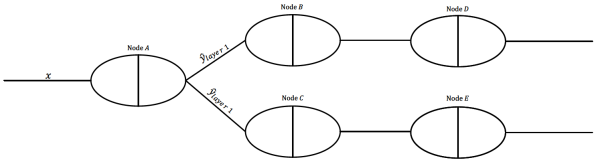

Figure 2. Comprehensive Network.

N-models such as figure 2 may appear in design. If figure 2 is considered as a directed graph G, then there exists a node set or vertex set $V = \{A, B, C, D, E\}$ and an edge set $E = \{(A, B), (A, C), (B, D), (C, E)\}$. The set E can be separated into two subsets $E_1 \subset E$ and $E_2 \subset E$, where $E_1 = \{(A, B), (B, D)\}$ and $E_2 = \{(A, C), (C, E)\}$. Likewise, set V can be separated in two subsets $V_1 \subset V$ and $V_2 \subset V$, where $V_1 = \{A, B, D\}$ and $V_2 = \{A, C, E\}$. Using E_1, E_2, V_1, and V_2, the graph G can be described a set consisting of two subgraphs G_1 and G_2, where $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. The graph G_1 is a three-layer basic comprehensive network, and G_2 is another three-layer base comprehensive network.

It is assumed that all nodes in a l-layer basic N-model have the same objective y, whether or not the nodes have the same comprehensive function. By this assumption, it follows that all nodes in G_1 have the objective y_1 and all nodes in G_2 have y_2. In some cases, $y_1 = y_2$, so node A (of figure 2) can have objectives y_1 and y_2 at the same time. However, if $y_1 \neq y_2$, then node A cannot have y_1 and y_2 at the same time. A solution for such cases where $y_1 \neq y_2$ is that node A objective function should be a mathematical mean of y_1 and y_2.

4. N-model Optimization

Figure 2 shows $\hat{y}_{layer1}$ of node A. The thesis $\hat{y}_{layer1}$ is the input of nodes B and C, while x is the input of node A. Further, the thesis of node B is the input of node D, and the thesis of node C is the input of node E. The theses of nodes D and E form the output of the network in figure 2. Furthermore, optimization of N-model occurs by optimizing its layers, and a comprehensive layer is optimized by optimizing individual nodes in the layer.

4.1. Error Mean

The purpose of the node error mean is data loss prevention. In supervised learning, a comprehensive node trained on $v_1, \cdots, v_g$ examples is assumed to make the errors $e_1, \cdots, e_g$ with respect to each example and objective y. Suppose that optimizing the node on the first example v_1 requires error e_1, and optimizing on v_2 requires e_2, then generally, optimizing on example v_i requires

error e_i. Here, $i, g \in \mathbb{N}$ and $i \leq g$. If only these requirements are assumed, then at example v_i during optimization, the node requires only e_i; previous errors $e_1, \cdots, e_{i-1}$ are not required. This implies that in supervised training of a node, using only the above requirements causes the node to 'neglect all previous errors' or 'loss data of previous errors'. To prevent loss of previous errors $e_1, \cdots, e_{i-1}$ at example v_i, let

$$\bar{e}_i = \left(\frac{(i-1)\bar{e}_{i-1}^{\phi} + e_i^{\phi}}{i} \right)^{\frac{1}{\phi}}, \tag{6}$$

where $\phi \in \mathbb{R}$. The symbol ϕ may be called **power** of comprehensive node. Equation (6) calculates power/hölder mean (Sykora, 2009) of a node error by using dynamic programming to minimize cost of computing $\bar{e}_i$.

The Brute-force Error Power Mean $BEPM = \left(\frac{e_1^{\phi}+e_2^{\phi}+\cdots+e_i^{\phi}}{i} \right)^{\frac{1}{\phi}}$ cost $O(n)$ time and space; in contrast, the Dynamic Error Power Mean $DEPM$ cost $O(1)$ time and space. To derive $DEPM$ formula from $BEPM$ formula, the following question is asked: If the $BEPM$ of $e_1, \cdots, e_i$ is known at example i, and later at example $i+1$, a new error e_{i+1} is known; is $\left(\frac{BEPM+e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}} = \left(\frac{e_1^{\phi}+e_2^{\phi}+\cdots+e_i^{\phi}+e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}}$? Answer to question: If the statement $\left(\frac{BEPM+e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}} = \left(\frac{e_1^{\phi}+e_2^{\phi}+\cdots+e_i^{\phi}+e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}}$ is a contradiction, then to avoid this contradiction,

$$\left(\frac{BEPM + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}} \neq \left(\frac{e_1^{\phi} + e_2^{\phi} + \cdots + e_i^{\phi} + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}}. \tag{7}$$

In statement (6), the inequality is caused by

$$BEPM + e_{i+1}^{\phi} \neq e_1^{\phi} + e_2^{\phi} + \cdots + e_i^{\phi} + e_{i+1}^{\phi}. \tag{8}$$

But,

$$i\left((BEPM)^{\phi}\right) + e_{i+1}^{\phi} = e_1^{\phi} + e_2^{\phi} + \cdots + e_i^{\phi} + e_{i+1}^{\phi}. \tag{9}$$

So, if $i\left((BEPM)^{\phi}\right) + e_{i+1}^{\phi} = e_1^{\phi} + e_2^{\phi} + \cdots + e_i^{\phi} + e_{i+1}^{\phi}$, then equation (10) is true.

$$\left(\frac{i\left((BEPM)^{\phi}\right) + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}} = \left(\frac{e_1^{\phi} + e_2^{\phi} + \cdots + e_i^{\phi} + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}}. \tag{10}$$

Let $BEPM = \bar{e}_i$,

$$\left(\frac{i\left((\bar{e}_i)^{\phi}\right) + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}} = \left(\frac{e_1^{\phi} + \cdots + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}}, \tag{11}$$

$$\left(\frac{i\left(\bar{e}_i^{\phi}\right) + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}} = \left(\frac{e_1^{\phi} + \cdots + e_{i+1}^{\phi}}{i+1} \right)^{\frac{1}{\phi}}, \tag{12}$$

$$\left(\frac{(i-1)\,\bar{e}_{i-1}^{\phi} + e_i^{\phi}}{i} \right)^{\frac{1}{\phi}} = \left(\frac{e_1^{\phi} + \cdots + e_i^{\phi}}{i} \right)^{\frac{1}{\phi}}, \tag{13}$$

$$DEPM = \left(\frac{(i-1)\,\bar{e}_{i-1}^{\phi} + e_i^{\phi}}{i} \right)^{\frac{1}{\phi}} = \bar{e}_i. \tag{14}$$

Equation (14) is written in terms of equation (6).

4.2. Divergence Problem

From equation (6), it is understood that

$$y \sim \hat{y}_i = h_i + \sqrt{\bar{e}_i} \tag{15}$$

where $\hat{y}_i$ denotes **thesis** of the node on input i, and $\bar{e}_i$ denotes **error mean** of the node on input i.

Experiments on basic N-model that consist of two nodes of form as formula (15), showed that during supervised training, the output of a basic N-model: (i) Converges to y if the nodes comprehensive functions are linear functions. (ii) Diverges from y if the nodes comprehensive functions are nonlinear. This divergence is a problem that can be solved by using the general formula of comprehensive node:

$$y \sim \hat{y}_i = \sqrt[d]{|h_i|} + \sqrt{\bar{e}_i}. \tag{16}$$

Formula (16) applies to both cases of linear and nonlinear comprehensive functions in a basic N-model. The **degree** $d \in \mathbb{R}$ of a node is the polynomial degree of the node comprehensive function. If $f_c(x) = x + 1$ is the comprehensive function of a node, then the degree of this node is 1 because the polynomial degree of $x + 1$ is 1. If $f_c(x) = x^2$, then 2 is the node degree. In formula (16), the purpose of calculating absolute hypothesis $|h_i|$ is to avoid complex number arithmetic. If $h_i = -1$ and $d = 2$, then $\sqrt[d]{h_i} = \sqrt{-1} = imaginary\ unit \neq \sqrt[d]{|h_i|} = \sqrt{|-1|} = 1$. The $\sqrt[d]{|h_i|}$ transforms $|h_i| = |f_{c,i}(x)|$ to a linear function if it is nonlinear. This transformation is necessary because basic N-model converges to y if its comprehensive functions are linear functions. Furthermore, by this transformation, it follows that $e_i = (y - \sqrt[d]{|h_i|})^2$.

A basic N-model has the following properties: network length, layers widths, comprehensive functions, powers, degrees, etc. These properties affects performance of basic comprehensive networks. Some experiments showed that a 5-layer basic N-model performed better than a 2-layer basic N-model in linear regression. Other properties should also be accounted for this better performance.

5. Multitask and Transfer Learning

Multitask learning (Caruana, 1997) involves learning more than one task by transferring previously learned knowledge from task j to learning of task $j+1$; $j \in \mathbb{N}$. This knowledge transfer process is based on the condition that task $j+1$ and task j are similar. Let the task be the objective y_j, and the knowledge (acquired during supervised training) be the error mean $\bar{e}_j$. Recall (see sections 3.2 and 3.4) that a single comprehensive node and a single basic N-model are assumed to learn only one objective, so a comprehensive layer can learn $n \in \mathbb{N}$ objectives if it consist of n nodes. Likewise, an N-model can learn n objectives if it consist of n basic comprehensive networks.

A comprehensive layer can be constructed in a manner, such that, upon every new objective y_{j+k}, a new node $node_{j+k}$ is added to the layer for learning the new y_{j+k}, and the error mean $\bar{e}_{j+k}$ of $node_{j+k}$ is initially set to $\bar{e}_{j+k-1}$. The error mean $\bar{e}_{j+k-1}$ is obtained from $node_{j+k-1}$ after $node_{j+k-1}$ has learned old objective y_{j+k-1}. Likewise, an N-model can be constructed in a manner such that upon every new objective y_{j+k}, a new basic comprehensive network N_{j+k} is added to the N-model for learning y_{j+k}, and all error-means $\bar{e}_{j+k,1}, \cdots, \bar{e}_{j+k,l}$ of N_{j+k} are initially set to $\bar{e}_{j+k-1,1}, \cdots, \bar{e}_{j+k-1,l}$ which is obtained from

N_{j+k-1} after N_{j+k-1} has learned old objective y_{j+k-1}. This kind of construction allows transfer of error-means/knowledge between $node_{j+k-1}$ and $node_{j+k}$, and between N_{j+k-1} and N_{j+k}. The more similar y_{j+k-1} is to y_{j+k}, the lesser training examples needed to optimize $node_{j+k}$ and N_{j+k}.

For the case of $y_j, \cdots, y_{j+k}$: It may be computationally inefficient to search through $y_j, \cdots, y_{j+k-1}$ for an objective that is similar to y_{j+k}; sometimes, there may not even be any objective in $y_j, \cdots, y_{j+k-1}$ that is similar to y_{j+k}. A method for avoiding this inefficient search is setting initially the error-means of any newly added $node_{j+k}$ or N_{j+k} to the dynamic mean of error-means of $node_j, \cdots, node_{j+k-1}$ or $N_j, \cdots, N_{j+k-1}$; respectively.

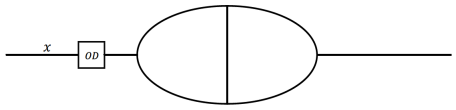

Figure 3. Comprehensive node and outlier detector.

A comprehensive node can have an **outlier detector** OD as shown in figure 3. An OD may require an **outlier detector coverage** ODC. During model validation where data from the testing set is fed to a trained comprehensive layer, the data is automatically fed to every node in the layer. The OD of each node verifies if this test data is an outlier of its training examples. If the data is an outlier, then the node will not produce hypothesis and thesis. Only those nodes to which the data is not outlier, will produce hypothesis and thesis. The ODC of an outlier detector is a value that determines the strength of detector.

Suppose OD uses of the formula

$$r = m(i) \pm [ODC \cdot sd(i)], \tag{17}$$

where r = input range, $m(\cdot)$ = mean function, i = input, and $sd(\cdot)$ = standard deviation function; then, a comprehensive node will produce hypothesis and thesis on only test inputs that are within ODC times standard deviation away from its training examples mean. If the test input is an outlier of the training examples, then the comprehensive node will produce a **null thesis**. For efficient computation in N-model, let only the input layer have ODs so that a node in the input layer that discover an outlier will produce a null thesis to notify nodes in further layers about nullity of an outlying test input. This notification avoids the need of checking for outlier in further layers. The notification prompts nodes in further layers to automatically produce null theses.

6. Regression Experiments

Three experiments (linear, nonlinear, and logistics regressions) was done with a multitasking N-model consisting of Q basic N-models. The basic N-models used in these experiments have the following properties: (i) Network length is 2. (ii) The width of every layer is 1. (iii) All nodes have the comprehensive function $f_c(x) = x^2$. (iv) Every node degree is 2. (v) Every node power is -6. (vi) Every node ODC is 4.

All experiments were designed to train the basic N-models on input vector $v^{train}_{*,j} \in \mathbb{R}^{g \times Q}$ to predict objective $y_j \in \mathbb{R}^Q$. Table 2 shows the $\mathbb{R}^{g \times Q}$ matrix design.In the table 2, $1 \geq i \leq g = 8$, and $1 \geq j \leq Q = 9$. The input vector $v^{train}_{*,j}$ consist of random $v^{train}_{i,j}$ values that are constrained to specified range. Similarly, table 3 has $g = 20$, and its vector $v^{test}_{*,j}$ consist of random $v^{test}_{i,j}$ values that are constrained to specified range – although, these constrains may not be necessary in real-world use of N-model.

Table 2. Objectives and random inputs for training.

y	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
$v^{train}_{1,*}$	$v^{train}_{1,1}$	$v^{train}_{1,2}$	$v^{train}_{1,3}$	$v^{train}_{1,4}$	$v^{train}_{1,5}$	$v^{train}_{1,6}$	$v^{train}_{1,7}$	$v^{train}_{1,8}$	$v^{train}_{1,9}$
$v^{train}_{2,*}$	$v^{train}_{2,1}$	$v^{train}_{2,2}$	$v^{train}_{2,3}$	$v^{train}_{2,4}$	$v^{train}_{2,5}$	$v^{train}_{2,6}$	$v^{train}_{2,7}$	$v^{train}_{2,8}$	$v^{train}_{2,9}$
$v^{train}_{3,*}$	$v^{train}_{3,1}$	$v^{train}_{3,2}$	$v^{train}_{3,3}$	$v^{train}_{3,4}$	$v^{train}_{3,5}$	$v^{train}_{3,6}$	$v^{train}_{3,7}$	$v^{train}_{3,8}$	$v^{train}_{3,9}$
$v^{train}_{4,*}$	$v^{train}_{4,1}$	$v^{train}_{4,2}$	$v^{train}_{4,3}$	$v^{train}_{4,4}$	$v^{train}_{4,5}$	$v^{train}_{4,6}$	$v^{train}_{4,7}$	$v^{train}_{4,8}$	$v^{train}_{4,9}$
$v^{train}_{5,*}$	$v^{train}_{5,1}$	$v^{train}_{5,2}$	$v^{train}_{5,3}$	$v^{train}_{5,4}$	$v^{train}_{5,5}$	$v^{train}_{5,6}$	$v^{train}_{5,7}$	$v^{train}_{5,8}$	$v^{train}_{5,9}$
$v^{train}_{6,*}$	$v^{train}_{6,1}$	$v^{train}_{6,2}$	$v^{train}_{6,3}$	$v^{train}_{6,4}$	$v^{train}_{6,5}$	$v^{train}_{6,6}$	$v^{train}_{6,7}$	$v^{train}_{6,8}$	$v^{train}_{6,9}$
$v^{train}_{7,*}$	$v^{train}_{7,1}$	$v^{train}_{7,2}$	$v^{train}_{7,3}$	$v^{train}_{7,4}$	$v^{train}_{7,5}$	$v^{train}_{7,6}$	$v^{train}_{7,7}$	$v^{train}_{7,8}$	$v^{train}_{7,9}$
$v^{train}_{8,*}$	$v^{train}_{8,1}$	$v^{train}_{8,2}$	$v^{train}_{8,3}$	$v^{train}_{8,4}$	$v^{train}_{8,5}$	$v^{train}_{8,6}$	$v^{train}_{8,7}$	$v^{train}_{8,8}$	$v^{train}_{8,9}$

For the linear regression experiment, $y_j = 10j$, and the random $v^{train}_{i,j}$ and $v^{test}_{i,j}$ are constrained to $j < v^{train}_{i,j} < j + 1$ and $j < v^{test}_{i,j} < j + 1$. For the nonlinear regression experiment, $y_j = 10^j$, and the random $v^{train}_{i,j}$ and $v^{test}_{i,j}$ are constrained to $j < v^{train}_{i,j} < j + 1$ and $j < v^{test}_{i,j} < j + 1$. For the logistic regression experiment, $y_j = j \mod 2$, and the random $v^{train}_{i,j}$ and $v^{test}_{i,j}$ are constrained to $j < v^{train}_{i,j} < j + 1$ and $j < v^{test}_{i,j} < j + 1$.

Table 3. Objectives and random inputs for testing.

y	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
$v^{test}_{1,*}$	$v^{test}_{1,1}$	$v^{test}_{1,2}$	$v^{test}_{1,3}$	$v^{test}_{1,4}$	$v^{test}_{1,5}$	$v^{test}_{1,6}$	$v^{test}_{1,7}$	$v^{test}_{1,8}$	$v^{test}_{1,9}$
$v^{test}_{2,*}$	$v^{test}_{2,1}$	$v^{test}_{2,2}$	$v^{test}_{2,3}$	$v^{test}_{2,4}$	$v^{test}_{2,5}$	$v^{test}_{2,6}$	$v^{test}_{2,7}$	$v^{test}_{2,8}$	$v^{test}_{2,9}$
$v^{test}_{3,*}$	$v^{test}_{3,1}$	$v^{test}_{3,2}$	$v^{test}_{3,3}$	$v^{test}_{3,4}$	$v^{test}_{3,5}$	$v^{test}_{3,6}$	$v^{test}_{3,7}$	$v^{test}_{3,8}$	$v^{test}_{3,9}$
$v^{test}_{4,*}$	$v^{test}_{4,1}$	$v^{test}_{4,2}$	$v^{test}_{4,3}$	$v^{test}_{4,4}$	$v^{test}_{4,5}$	$v^{test}_{4,6}$	$v^{test}_{4,7}$	$v^{test}_{4,8}$	$v^{test}_{4,9}$
$v^{test}_{5,*}$	$v^{test}_{5,1}$	$v^{test}_{5,2}$	$v^{test}_{5,3}$	$v^{test}_{5,4}$	$v^{test}_{5,5}$	$v^{test}_{5,6}$	$v^{test}_{5,7}$	$v^{test}_{5,8}$	$v^{test}_{5,9}$
$v^{test}_{6,*}$	$v^{test}_{6,1}$	$v^{test}_{6,2}$	$v^{test}_{6,3}$	$v^{test}_{6,4}$	$v^{test}_{6,5}$	$v^{test}_{6,6}$	$v^{test}_{6,7}$	$v^{test}_{6,8}$	$v^{test}_{6,9}$
$v^{test}_{7,*}$	$v^{test}_{7,1}$	$v^{test}_{7,2}$	$v^{test}_{7,3}$	$v^{test}_{7,4}$	$v^{test}_{7,5}$	$v^{test}_{7,6}$	$v^{test}_{7,7}$	$v^{test}_{7,8}$	$v^{test}_{7,9}$
$v^{test}_{8,*}$	$v^{test}_{8,1}$	$v^{test}_{8,2}$	$v^{test}_{8,3}$	$v^{test}_{8,4}$	$v^{test}_{8,5}$	$v^{test}_{8,6}$	$v^{test}_{8,7}$	$v^{test}_{8,8}$	$v^{test}_{8,9}$
$v^{test}_{9,*}$	$v^{test}_{9,1}$	$v^{test}_{9,2}$	$v^{test}_{9,3}$	$v^{test}_{9,4}$	$v^{test}_{9,5}$	$v^{test}_{9,6}$	$v^{test}_{9,7}$	$v^{test}_{9,8}$	$v^{test}_{9,9}$
$v^{test}_{10,*}$	$v^{test}_{10,1}$	$v^{test}_{10,2}$	$v^{test}_{10,3}$	$v^{test}_{10,4}$	$v^{test}_{10,5}$	$v^{test}_{10,6}$	$v^{test}_{10,7}$	$v^{test}_{10,8}$	$v^{test}_{10,9}$
$v^{test}_{11,*}$	$v^{test}_{11,1}$	$v^{test}_{11,2}$	$v^{test}_{11,3}$	$v^{test}_{11,4}$	$v^{test}_{11,5}$	$v^{test}_{11,6}$	$v^{test}_{11,7}$	$v^{test}_{11,8}$	$v^{test}_{11,9}$
$v^{test}_{12,*}$	$v^{test}_{12,1}$	$v^{test}_{12,2}$	$v^{test}_{12,3}$	$v^{test}_{12,4}$	$v^{test}_{12,5}$	$v^{test}_{12,6}$	$v^{test}_{12,7}$	$v^{test}_{12,8}$	$v^{test}_{12,9}$
$v^{test}_{13,*}$	$v^{test}_{13,1}$	$v^{test}_{13,2}$	$v^{test}_{13,3}$	$v^{test}_{13,4}$	$v^{test}_{13,5}$	$v^{test}_{13,6}$	$v^{test}_{13,7}$	$v^{test}_{13,8}$	$v^{test}_{13,9}$
$v^{test}_{14,*}$	$v^{test}_{14,1}$	$v^{test}_{14,2}$	$v^{test}_{14,3}$	$v^{test}_{14,4}$	$v^{test}_{14,5}$	$v^{test}_{14,6}$	$v^{test}_{14,7}$	$v^{test}_{14,8}$	$v^{test}_{14,9}$
$v^{test}_{15,*}$	$v^{test}_{15,1}$	$v^{test}_{15,2}$	$v^{test}_{15,3}$	$v^{test}_{15,4}$	$v^{test}_{15,5}$	$v^{test}_{15,6}$	$v^{test}_{15,7}$	$v^{test}_{15,8}$	$v^{test}_{15,9}$
$v^{test}_{16,*}$	$v^{test}_{16,1}$	$v^{test}_{16,2}$	$v^{test}_{16,3}$	$v^{test}_{16,4}$	$v^{test}_{16,5}$	$v^{test}_{16,6}$	$v^{test}_{16,7}$	$v^{test}_{16,8}$	$v^{test}_{16,9}$
$v^{test}_{17,*}$	$v^{test}_{17,1}$	$v^{test}_{17,2}$	$v^{test}_{17,3}$	$v^{test}_{17,4}$	$v^{test}_{17,5}$	$v^{test}_{17,6}$	$v^{test}_{17,7}$	$v^{test}_{17,8}$	$v^{test}_{17,9}$
$v^{test}_{18,*}$	$v^{test}_{18,1}$	$v^{test}_{18,2}$	$v^{test}_{18,3}$	$v^{test}_{18,4}$	$v^{test}_{18,5}$	$v^{test}_{18,6}$	$v^{test}_{18,7}$	$v^{test}_{18,8}$	$v^{test}_{18,9}$
$v^{test}_{19,*}$	$v^{test}_{19,1}$	$v^{test}_{19,2}$	$v^{test}_{19,3}$	$v^{test}_{19,4}$	$v^{test}_{19,5}$	$v^{test}_{19,6}$	$v^{test}_{19,7}$	$v^{test}_{19,8}$	$v^{test}_{19,9}$
$v^{test}_{20,*}$	$v^{test}_{20,1}$	$v^{test}_{20,2}$	$v^{test}_{20,3}$	$v^{test}_{20,4}$	$v^{test}_{20,5}$	$v^{test}_{20,6}$	$v^{test}_{20,7}$	$v^{test}_{20,8}$	$v^{test}_{20,9}$

Both table 2 and 3 are similar to table 1. In terms of table 1, the $y_1, \cdots, y_9$ of table 2 and 3 can be called systolic blood pressures, while $v^{train}_{1,j}, \cdots, v^{train}_{8,j}$ be called training set of ages, and $v^{test}_{1,j}, \cdots, v^{test}_{20,j}$ be called testing set of ages. The major difference between table 1 and 2 (and likewise table 1 and 3), is that each objective y_j in table 1 is paired with only one age input $x_{1,j}$, while each y_j in table 2 and 3 is paired with multiple inputs in $v^{train}_{1,j}, \cdots, v^{train}_{8,j}$ as well as in $v^{test}_{1,j}, \cdots, v^{test}_{20,j}$. Results of the three experiments can be found at https://github.com/calebnwokocha/N-model.

7. Conclusion

This paper explains fundamental concepts of comprehensive network. The paper also illustrates the design of regression experiments with comprehensive network, claiming that a comprehensive network is capable of learning from only a few examples. However, there is more work to be done with comprehensive networks. Further work with comprehensive network should (in no particular order) study the effects of complex numbers in formula (16), derive methods for rule extraction from N-model, use N-model to solve computer vision problems, use N-model to solve natural language processing problems, and so on.

References

Caruana, R. Multitask learning. *Machine Learning*, 28: 41–75, 1997.

Jennrich, R. I. Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643, 1969.

Legendre, A. M. *New methods for determining the orbits of comets*. F. Didot Paris, 1805.

Malinvaud, E. The consistency of nonlinear regressions. *The Annals of Mathematical Statistics*, 41(3):956–969, 1970.

McCulloch, W. S. and Pitts, W. A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5:127–147, 1943.

Mitchell, T. M. *Machine Learning*. McGraw-Hill, New York, 1997. ISBN 978-0-07-042807-2.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

Sykora, S. Mathematical means and averages: Basic properties. 07 2009. doi: 10.3247/SL3Math09.001.

Verhulst, P. F. Mathematical Researches into the Law of Population Growth Increase. *The Royal Academy of Brussels and the University of Louvain*, 1845.