Rethinking the Artificial Neural Networks: A Novel Approach

Usman Ahmad and Hong Song
Beijing Institute of Technology
usman715@gmail.com, anniesun@bit.edu.cn

Abstract

In this paper, we proposed a novel approach to build the Artificial Neural Network (ANN). We addressed the fundamental questions, 1) what is the architecture of the ANN model? Should it really have a layered architecture? 2) What is a neuron: a processing unit or a memory cell? 3) How neurons must be interconnected and what should be the mechanism of weights assignment? 4) How to involve prior knowledge, bias, and generalization to extract the features? In this paper, we have given an abstract view of our approach for supervised learning with text data only and explain it through examples.

1. Proposed Architecture of ANN

In this section, we proposed a non-layered architecture of ANN under control of a centralized mechanism.

1.1. Mesh of Subnets

Neurons in an ANN must not be in a layered format. Infect; the neurons in the ANN must be interconnected in a way that we call it *the mesh of subnets.* In our proposed architecture, an ANN is a combination of interconnected subnets, where a subnet is a collection of neurons. A subnet is basically a logical distribution of neurons in an ANN. A neuron must belong to a subnet or more than one subnet and most likely connected to few or all neurons of its subnet/s. Moreover, a neuron can also be connected to some other subnet's neurons.

1.2. A Centralized Mechanism

An ANN must have the main decision making, centralized and outer mechanism to control and monitor the whole working of the ANN. The centralized mechanism takes input and also decide how deeper to process the input data in the ANN and process it through ANN. Each neuron in the ANN is connected to the centralized mechanism through the subnet or directly (exceptional case). In other words, each neuron is connected and has a direct approach to all neurons in the ANN through a centralized mechanism. The centralized mechanism is responsible for that the input data is being processed and move from one neuron/subnet to the next. The centralized mechanism also works as a bypass for input data to directly move from one neuron to any other neuron in the whole ANN, through the centralized mechanism.

Centralized mechanism contains some neurons/subnets inside it for temporary storing (of input data, targeted subnets and subnets/neurons of ANN and results to compare) and to communicate with outer neurons/subnets. The centralized mechanism is also responsible to create the new neurons, subnets, and connections between them. We have many types of connections: like between the neurons of the same subnet, between the neurons of the different subnets, between the subnets and between the subnet and centralized mechanism. The centralized mechanism has a vital role in the ANN and discussed respectively in the respective sections.

1.3. Weights Assignment

During the training phase (or conditionally in the testing phase), the centralized mechanism splits the input data into single or attributes wise values and creates the new subnet and neuron in the corresponding subnet against each value to store it. Moreover, it also creates the connections between neurons/subnets according to the incoming flow of data and assigns an initial weight value for each connection. Once a weight is assigned then it gets updated on each repetition of the same input data values. The direction of the connection between the neurons just shows the sequence of input, otherwise, the data can move in both directions. Moreover, feedback loops are also allowed.

2. Neuron

In our proposed ANN, a neuron is a memory cell and contains the data which is given to the ANN for training and testing also. In some cases a neuron can contain a function in its memory; it is discussed in the examples. Neurons are interconnected and each connection has a particular weight. New connections are created and become stronger and weaker or even removed (along with neuron itself) w.r.t. the frequency of input data flows between neurons during training and testing process. The weight is the value that how much far a neuron is from a particular neuron. Wight is the distance of a neuron from the other neurons. In other words, the weight defines the place of a neuron in an ANN.

- A neuron must be a part of a subnet or it can be a part of more than one subnet.
- A neuron can move from one subnet to other subnet or new subnet.

3. Subnet

A subnet is a logical distribution of neurons and each neuron is connected to a centralized mechanism through subnet (or directly connected in exceptional cases). New subnets along with the corresponding type of data (neurons) in it are created during training and testing the ANN model. If human declares the number of subnets and the corresponding type of data in each subnet before training the model, we call it semi-trained ANN model. The connections between subnets also have a weight value. The connection and corresponding weight between two subnets depend upon the frequency of input data flow between the subnets during training and testing process.

- More than on subnets can be merged and make a new subnet and vice versa.
- A subnet may have the child and subnet parents subnets.

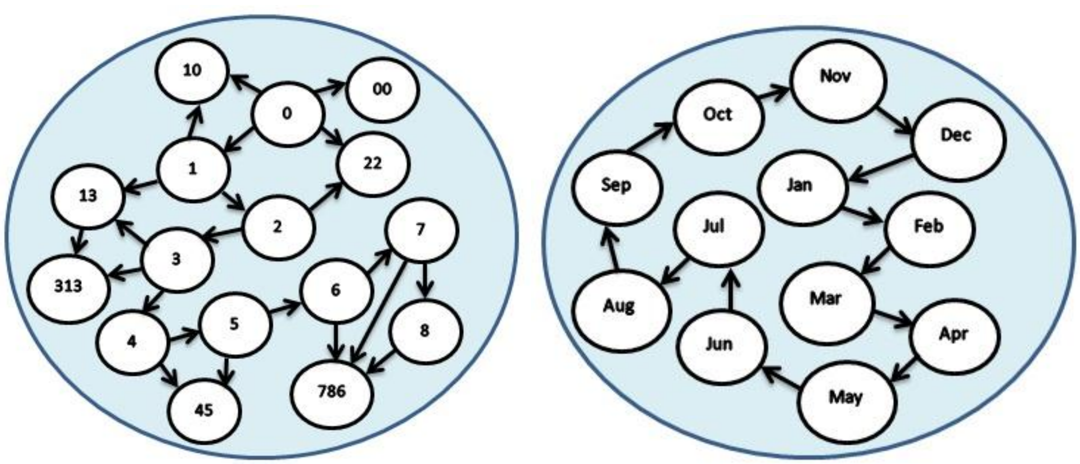
Every human has its own way to remember and recognize the things, the same way every ANN may have different subnets and neurons distribution. The subnet creation can be formalized and one way maybe that number of subnets equal to the number of logical distributions of the dataset. For the time, just consider it as the base concept for image data processing.

4. Training

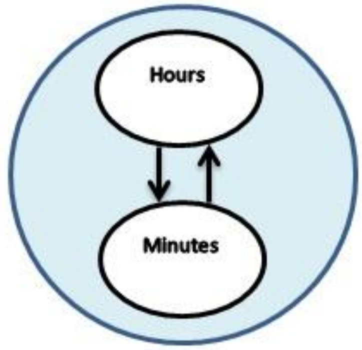
In this section, we explain the training of our proposed ANN model through an example. A sample dataset of a diabetes patient is prepared that is 2 weeks insulin amount log and contains the 30 records. The dataset contains three attributes; date, time and the amount of injected insulin. We build an ANN model to predict the amount of insulin amount to be injected on a given date and time. The 30 records injected in a queue as input to the ANN for training. Input data is fetched by the centralized mechanism and it creates the subnets, neurons, and connections between them.

Firstly, we must have the prior knowledge. We have the three subnets to contain the prior knowledge. Figure 1 illustrates the subnet architecture for subnet 1, 2, and 3.

- Subnet 1: Contains number systems i.e., 1, 2, 3…
- Subnet 2: Contains the names of months
- Subnet 3: Contains the time structure, i.e., 24 hours and 60 minutes



(a) Subnet 1 (b) Subnet 2

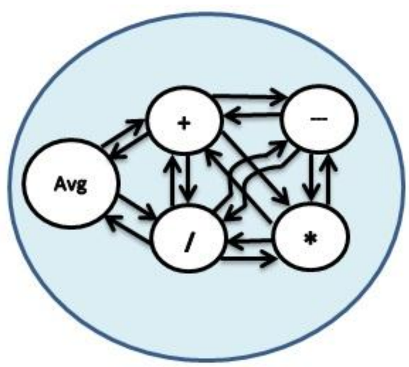


(c) Subnet 3

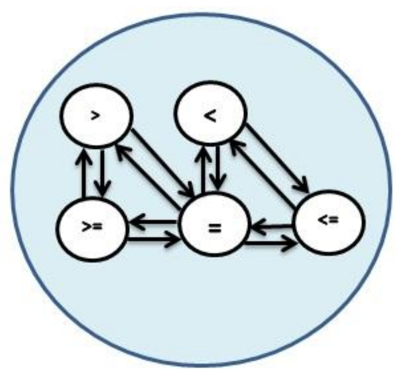
Figure 1: Subnet 1 in (a) illustrates number systems. Few neurons are shown in diagram just to demonstrate the architecture. The arrows between the neurons are accordingly to the sequence of input values. It starts from zero then 1, 2 and so on. For example, 786 is a combination of three values so linked with relevant neurons that are 7, 8, 6 and 785. Similarly, 10 is a combination of 1 and 0 so linked with three neurons that are 0, 1 and 9. Subnet 2 in (b) contains the names of month's starts from January and ends at December. Subnet 3 in (c) contains the time structure, i.e., hours, minutes and seconds. For our example, we only considered the hours.

Once the training data is stored and formulated in the neurons, then we extract the features. So we have the subnet 4 and 5 having relevant functions for generalization to extract the features and patterns, dawn in Figure 2. Subnet 4 and 5 as given below:

- Subnet 4: Contains arithmetic operators
- Subnet 5: Contains relational operators



(a) Subnet 4



(b) Subnet 5

Figure 2: Subnet 4 in (a) contains arithmetic operators and subnet 5 in (b) contains relational operators for generalization to extract the features.

Subnet 1, 2, 3, 4 and 5 are interconnected as shown in Figure 3. We have three attributes in our dataset and correspondence between them. So, we have the subnet 6, 7 and 8 to store the input data.

- Subnet 6: Contain the date that is first attribute of dataset
- Subnet 7: Contain the timestamps that is second attribute of dataset
- Subnet 8 (Target subnet): Contain the value of insulin injection

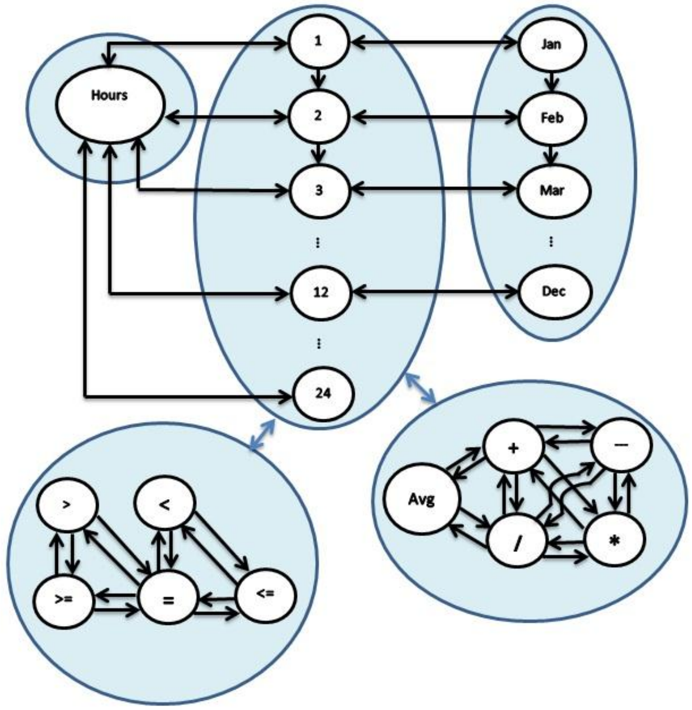
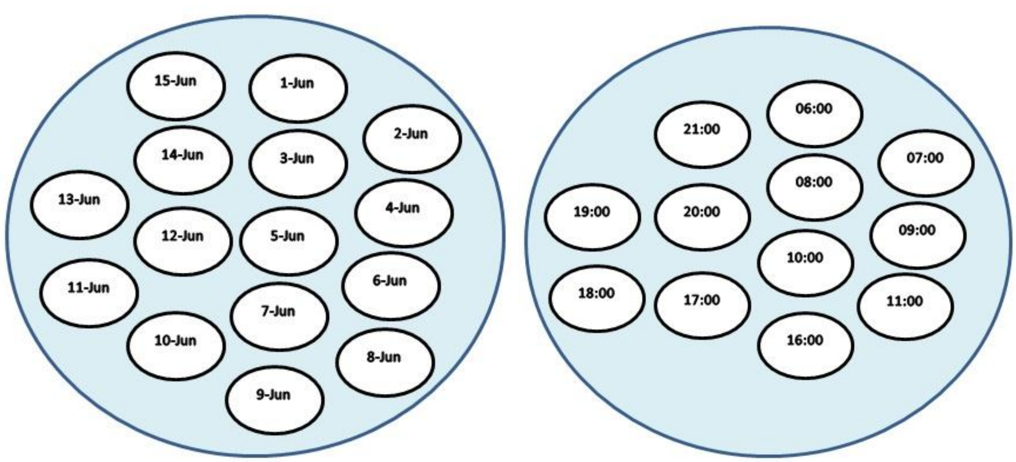
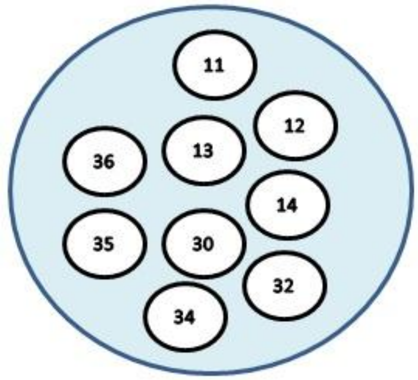


Figure 3: Subnet 1, 2, 3, 4 and 5 interconnections which make a ground for incoming training data.

In our dataset, we have 15 unique values in the date attribute, 12 unique timestamps in the second attribute and 9 unique insulin amount values in the third attribute, so subnets 6, 7 and 8 have 15, 12 and 9 neurons, respectively. Figure 4 illustrates the subnet architecture for subnet 6, 7, and 8 and Figure 5 depicts the connection between dataset attributes subnet (6, 7, and 8) and prior knowledge subnets.

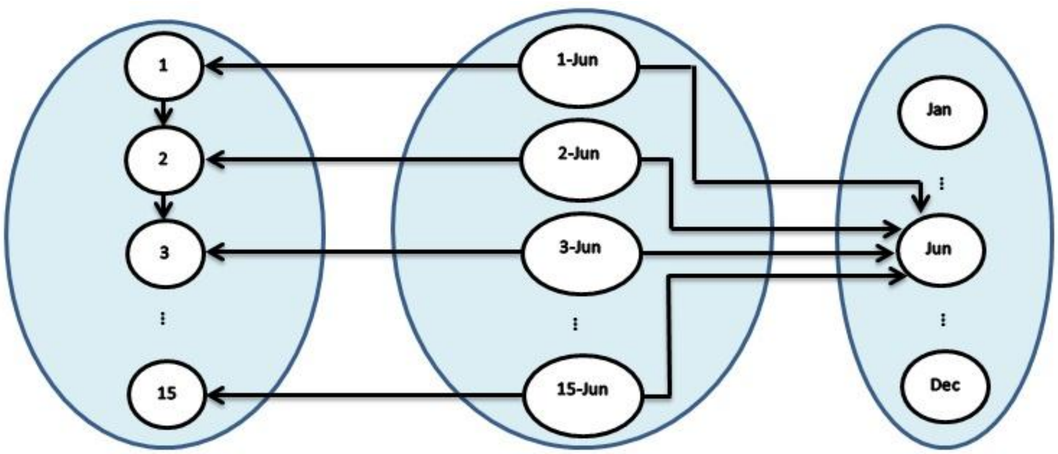


(a) Subnet 6 (b) Subnet 7

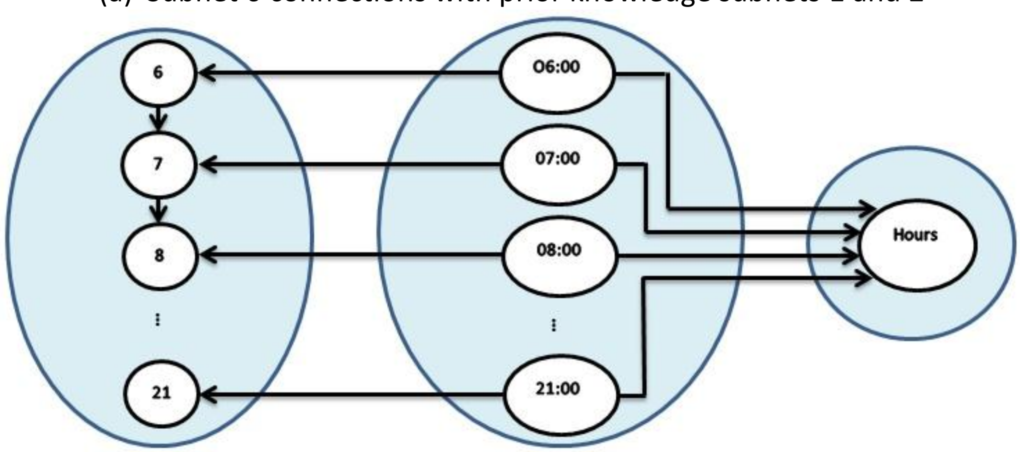


(c) Subnet 8 (target subnet)

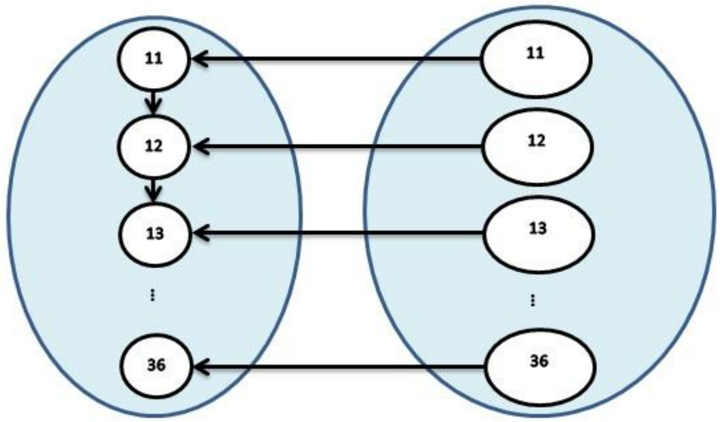
Figure 4: Subnet 6 in (a) contain the first attribute of the dataset that is date and holds 15 neurons. Subnet 7 in (b) contains the timestamps that are the second attribute of the dataset and holds 12 neurons. Subnet 8 is the target subnet contains the value of insulin injection and holds 9 neurons.



(a) Subnet 6 connections with prior knowledge subnets 1 and 2



(b) Subnet 7 connections with prior knowledge subnets 1 and 3



(c) Subnet 8 connections with prior knowledge subnet 1

Figure 5: Connection between dataset attributes subnet neurons and prior knowledge subnets.

In Figure 6, we have drawn the neural network for the first 10 records in our dataset. For this paper, we have assigned the initial weight value 1. The weight of the corresponding connection is decreased by .25% for all next iterations. In the first 10 records of our dataset, the time attribute value 18:00 against insulin amount 32 and 10:00 against insulin amount 12 are repeated. So the weight between these neurons is 0.75. Please note that we have used the value stored in the neuron as the name of the neuron.

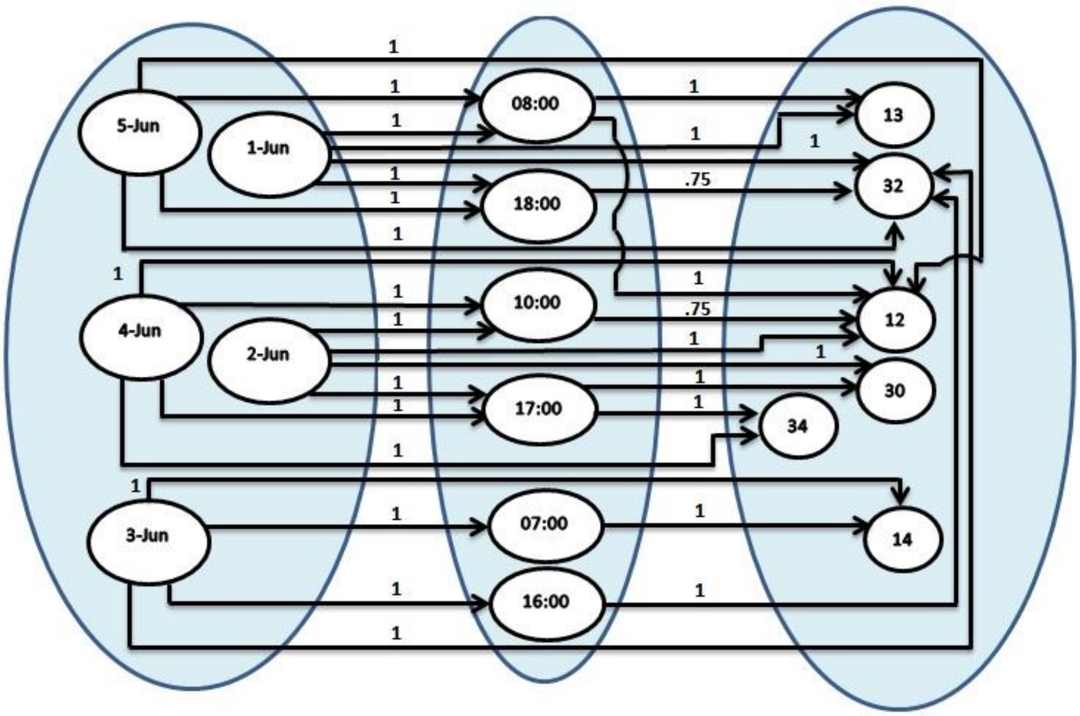


Figure 6: Depicted the structure of subnet 6, 7 and 8 (left to right) of ANN for the first 10 records in our insulin log dataset.

5. Testing

Let's take the 11th record as a test record, so we have to predict the inulin amount against the 6-Jun and 11:00. The centralized mechanism is responsible to take input and forward it to the appropriate subnet. So, the date (6-Jun) value put in the subnet 6 and the time (11:00) value put in the subnet 7. Activation function compares the given input value with the neuron's value and generates a result (Y/N). If value not found then finds the nearest matching value/s.

First input value i.e., 6-Jun doesn't exist and 5-Jun is the nearest value in subnet 6. The 5-Jun neuron is approaching the four neurons. Find such neurons from these four neurons which belong to the subnet 7 (contain time data) and must also approach to the same neuron in target subnet as the 5-Jun neuron is approaching; those are 08:00 and 18:00 neurons. In other words, don't consider the target subnet neurons at this stage. Now, the activation function compares the second input (11:00) value with these two filtered neuron's values and generates a result (Y/N). If value not found then find the nearest matching value/s. The nearest neuron is 08:00 approaching to neuron containing value 13 in target subnet.

Repeat the process for second input value i.e., 11:00 which also doesn't exist in subnet 7 and 10:00 is the nearest value is picked. The 10:00 neuron is approaching the three neurons. Find such neurons from these three neurons which belong to the subnet 6 (contain date data) and must also approach to the same neuron in target subnet as 10:00 neuron is approaching; those are 2-Jun and 4-Jun neurons. Now, the activation function compares the first input (5-Jun) value with these two filtered neuron's values. The nearest neuron is 4-Jun approaching to neuron containing value 12 in target subnet. Now take the average of these two selected neuron's values of target subnet that are 13 and 12 and the result is 12.5 or 13 in the round.

Now, let's take the 12th record as a test record, so we have to predict the inulin amount against the 6-Jun and 17:00 values. The date (6-Jun) value put in the subnet 6 and the time (17:00) value put in the subnet 7. First input value i.e., 6-Jun doesn't exist and 5-Jun is the nearest value in subnet 6. The 5-Jun neuron is approaching the four neurons and two filtered neurons are 08:00 and 18:00 neurons. The nearest neuron to the second input value (17:00) is 18:00 approaching to neuron containing value 32 in target subnet. For second input value i.e., 17:00 exist in subnet 7. The 17:00 neuron is approaching the four neurons and two filtered neurons are 2-Jun and 4-Jun neurons. The nearest neuron to the first input value (6-Jun) is 4-Jun approaching to neuron containing value 34 in target subnet. Take the average of 32 and 34 and the result is 33.

If a neuron belongs to more than one neuron in the target subnet, then we will consider the neuron having the minimum weight, and if have the same weight then take the average. We depict a scenario in Figure 7, just to understand the role of weights in testing. For example the 16-Jun and 07:00 are the selected neurons and both belong to two neurons in the target subnet that are 14 and 30. As the neuron containing value 30 have minimum weight so it will be selected rather than neuron contains value 14.

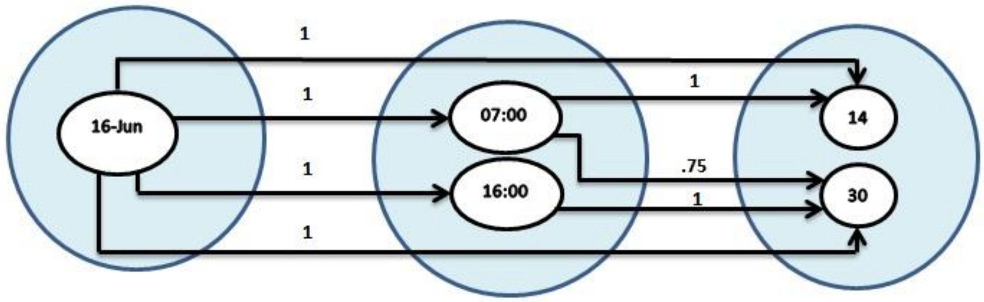


Figure 7: Scenario to understand the role of weights in testing.

Bias is an input data stored in a dedicated area of ANN. It is stored in the same way as training data and used to process test data. Neurons are interlinked with the neurons containing biased data so have an impact on the output. Figure 8 depicts a scenario to understand the bias. For example, if a person has taken an extra quantity of sugar then he should add 2 extra points of inulin in the routine insulin amount value. So during testing, we will add the value 2 in the predicted value of insulin amount.

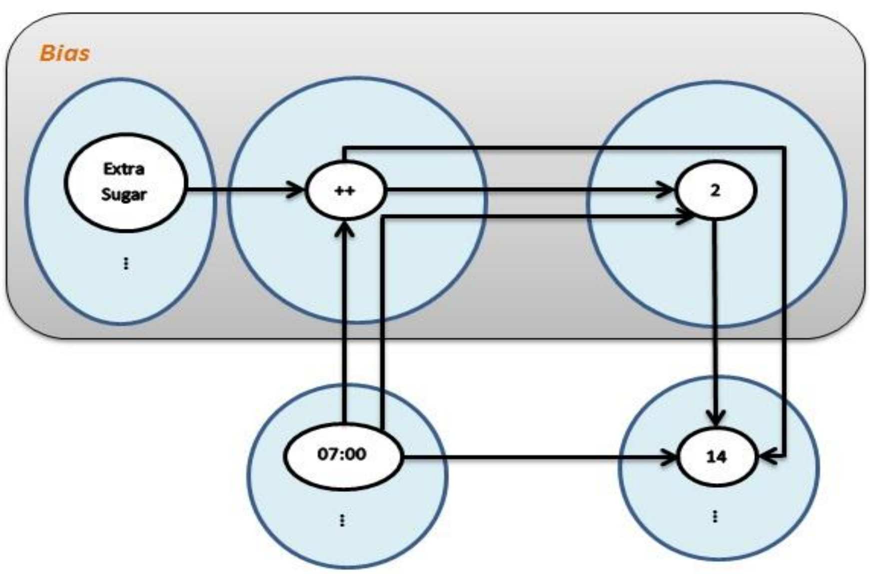


Figure 8: Scenario to understand the role of bias in testing.

If the testing results are 100 percent true then the weight value of neurons through which the information has flowed during testing will be updated in the same way as updated in training.

6. Iris Dataset Example

We have five attributes in Iris dataset and correspondence between them. Please refer to Table 1 for details for unique records/neurons in Iris dataset. Please refer to insulin pump log example for the details of prior knowledge.

	Attribute	Unique values/No of Neurons
Subnet 1	Sepal length	35
Subnet 2	Sepal width	23
Subnet 3	Petal length	43
Subnet 4	Petal width	22
Subnet 5 (target subnet)	Species	3

Table 1: Subnets and unique records/neurons in Iris dataset subnets

Record #	Sepal length	Sepal width	Petal length	Petal width	Species
35	4.9	3.1	1.5	0.2	Setosa
94	5.0	2.3	3.3	1.0	Versicolor

Table 2: Randomly selected record to test form Iris dataset

6.1. Testing

We take the 35th record as a test record, given in Table 2. Figure 9 illustrates the ANN for the first input value of our test record that is sepal length (4.9). The value 4.9 repeated 5 times in the Iris dataset (excluding our test record) and has 16 unique values in subnet 2, 3, 4, and 5, refer to Figure 9. The values 1.4 and 0.1 repeated in subnet 3 and 4, respectively. Sepal length, sepal width, petal length, and petal width put in the relevant subnet 1, 2, 3 and 4, respectively and we have to predict the species.

First input value i.e., 4.9 exists in subnet 1. The 4.9 neuron is approaching to the 16 neurons. Find such neurons from these 16 neurons which belong to the subnet 2, 3 and 4 and must also approach to the same neuron in target subnet as 4.9 neuron is approaching; which are total 13 neurons (5 neurons in subnet 2, 4 neurons in subnet 3 and 4 neurons in subnet 4). Then, the activation function compares the first input value i.e., sepal width (3.1) value with 5 filtered neuron's values in subnet 2. 3.1 value also exists and approaching to neuron containing value setosa in target subnet. The activation function compares the petal length (1.5) value with his 4 filtered neuron's values in subnet 3. The value 1.5 also exists and approaching to neuron containing value setosa in target subnet. The activation function compares the petal width (0.2) value with his 4 filtered neuron's values in subnet 4. The value 0.2 also exists and approaching to neuron containing value setosa in target subnet.

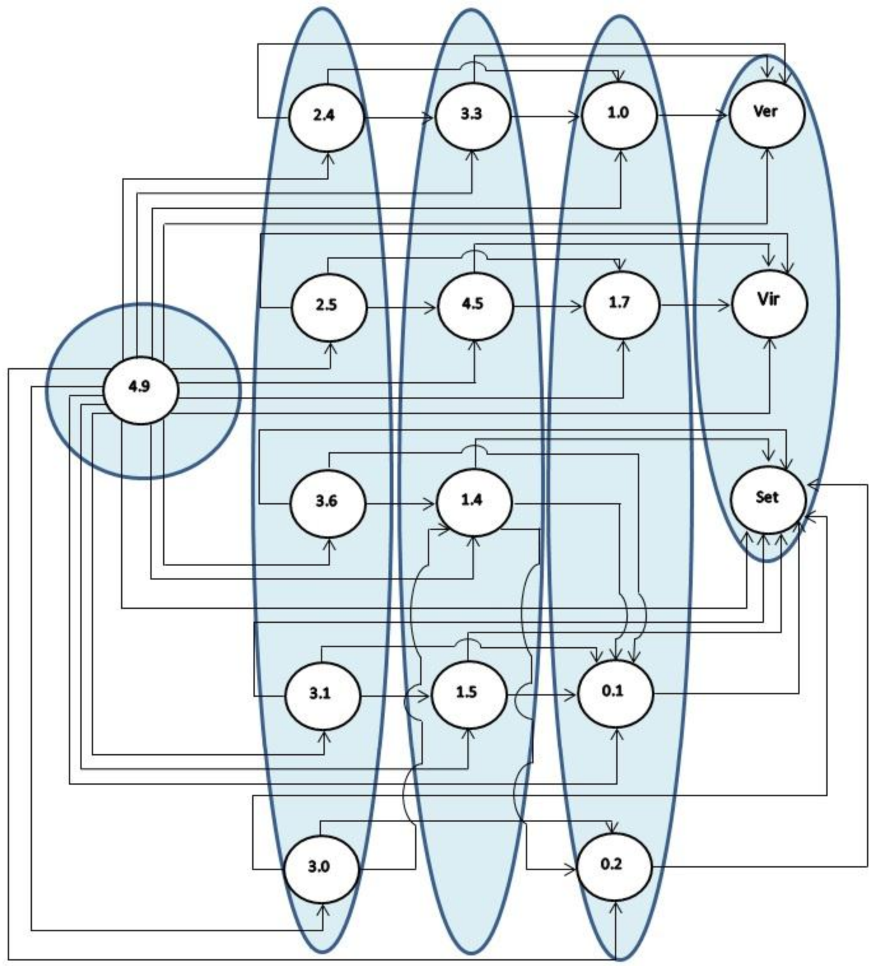


Figure 9: The ANN architecture of subnet 1, 2, 3 and 4 (from left to right) to test the 35th record in Iris dataset.

Repeat the process in the above paragraph for the second input value. Second input value i.e., 3.1 exists in subnet 2. The 3.1 value is repeated 10 times in Iris dataset excluding the test record. The 3.1 neuron is approaching to the total 26 neurons. Find such neurons from these 26 neurons which belong to the subnet 1, 3 and 4 and must also approach to the same neuron in target subnet as 3.1 neuron is approaching; which are 23 neurons (6 neurons in subnet 1, 9 neurons in subnet 3 and 8 neurons in subnet 4). Now, the activation function compares the first input value i.e., sepal length (4.9) value with his 6 filtered neuron's values in subnet 1. The value 4.9 also exists and approaching to neuron containing value setosa in target subnet. The activation function compares the petal length (1.5) value with his 9 filtered neuron's values in subnet 3. The value 1.5 also exists and approaching to neuron containing value setosa in target subnet. The activation function compares the petal width (0.2) value with his 5 filtered neuron's values in subnet 4. The value 0.2 also exists and approaching to neuron containing value setosa in target subnet. By repeating this process for thirds and fourth input values we also get the setosa value.

In case the neurons approach to more than one neuron in the target subnet, then we will select the value having the maximum occurrence rate. We take the 94^{th} record as a test case; it also gives the accurate result that is versicolor.

7. Discussion and Future Work

It is the duty of centralized mechanism to select the function to apply on the input data in order to extract the features and the centralized mechanism perform this task by previous experience and testing results. Backpropagation also performed by the centralized mechanism. Moreover, centralized mechanism contains the information of the whole ANN subnets (and neurons in it) and extracted features so needs to be refreshed and up to date.

The neurons and subnets in an ANN is a storage mechanism and centralized mechanism is the main processing unit which has authority to decide that how deeper to process the information. For example, the centralized mechanism can process the test record by finding 3 nearest values of input record (in given examples we are limited to find one nearest value) and take the average. Or the deeper processing also approaches to process the bias more deeply. Each neuron is connected and has a direct approach to all other neurons in the ANN through the centralized mechanism. In deeper processing, the centralized mechanism may pass the input data directly from one neuron to any other neuron in the whole ANN and weight values between subnets will also be considered in it.

In this paper, we explained our proposed model as simple as much possible to make understandable by everyone. We need to involve mathematics, statistics and some other fields. For example, we can use mathematical schemes during the testing phase where we need to find the corresponding neurons of a selected neuron that must approach to the same neurons in the target subnet as our selected neuron is approaching. Or these fields can make our model more simplified like the usage of logarithms; by which we can express large numbers in a convenient way. The future work includes the comparative analysis with old fashion ANN models based on code and benchmark datasets.

The scope of this paper is limited to the text data. We have planned to propose this model for the image data in the near future. One relevant question in this context is how to deal with large datasets? The answer to this question will help in understanding the feasibility of this solution for large dataset. First of all, today memory is not an issue unless the performance is not compromised. Secondly, in our proposed architecture, data never repeat but in the form of updating the weight value. Moreover, we are intended to cover our perspective about to "*believe in quality not the quantity of data for feature extraction"* in our future article.

Dataset

Sr. No.	Date	Time	Insulin Amount
1	1-June	08:00	13
2	1-June	18:00	32
3	2-June	10:00	12

4	2-June	17:00	30
5	3-June	07:00	14
6	3-June	16:00	32
7	4-June	10:00	12
8	4-June	17:00	34
9	5-June	08:00	12
10	5-June	18:00	32
11	6-June	11:00	13
12	6-June	17:00	30
13	7-June	08:00	11
14	7-June	19:00	32
15	8-June	07:00	12
16	8-June	18:00	34
17	9-June	07:00	11
18	9-June	20:00	30
19	10-June	06:00	13
20	10-June	17:00	35
21	11-June	10:00	14
22	11-June	18:00	32
23	12-June	09:00	12
24	12-June	19:00	36
25	13-June	09:00	12
26	13-June	16:00	32
27	14-June	08:00	13
28	14-June	21:00	30
29	15-June	10:00	12
30	15-June	20:00	34