Int J Adv Manuf Technol (2001) 18:89–97

The International Journal of
Advanced
Manufacturing
Technology

Intelligent Classification of the Drop Hammer Forming Process Method

S. H. Huang, H. Xing and G. Wang

Intelligent CAM Systems Laboratory, Department of Mechanical, Industrial and Manufacturing Engineering, The University of Toledo, USA

Forging is a cost-effective way to produce net-shape or near-net-shape components. Forged components are used throughout the manufacturing sector in many different applications. Owing to the lack of an accurate process model, modern forging operations still largely rely on operator skills and experience. The expertise of skilled operators can help to develop a forging process model. However, these operators' expert knowledge is accumulated through years of hands-on experience and is often biased towards their own heuristic. It is well known that accurate acquisition of this type of knowledge is challenging and time-consuming. In this paper, an innovative approach is developed to acquire forging process knowledge automatically by combining learning ability of the neural networks with the structured knowledge representation of rule-based systems. The approach is applied to the classification of process methods used in a type of impression-die forging, namely, drop hammer forming. Specifically, process data from an aerospace company's production facility are collected. The data are processed and then used to train a back-propagation neural network. By analysing the connections and weights of the trained neural network, concise and intelligible rules are extracted. These rules can be used to allow a clearer specification of the drop hammer forming process plan and to shorten learning curves for novice operators.

Keywords: Drop hammer forming; Knowledge acquisition; Neural networks; Rule extraction

1. Introduction

Forging is the oldest known metal working process. During the forging process, a metal workpiece is plastically deformed by pressing, squeezing, or hammering forces, at temperatures ranging from ambient to 1500°C. The controlled process of deformation that takes place imparts exceptional metallurgical soundness and mechanical properties to the forging – structural integrity, impact strength, fracture toughness, fatigue life, and uniformity. Forged components are used throughout the manufacturing sector in many different applications. They are found in over 20% of the products representing the gross domestic product (GDN) of the USA, and hence, are essential to the US industrial economy [1]. Current industrial forging operations still remain skill-based to a large extent. As a result, operator experience plays an important role in forging quality and productivity. There is also much trial and error in developing processes for producing forged components, which results in significant inefficiency. In order to overcome this, it is important to develop computer-aided process planning for forging. Process planning knowledge is usually accumulated through years of hands-on experience. It is heuristic in nature, and, thus, very difficult to computerise.

Traditional techniques for computerising heuristic knowledge are based mainly on the symbolic artificial intelligence (AI) paradigm, where a knowledge engineer interviews a domain expert, acquires and documents the expert's knowledge in terms of IF-THEN rules, and transfers the knowledge into a computerised expert system. It is well known that this approach suffers from the knowledge acquisition bottleneck [2–4]. This drawback is especially severe in manufacturing applications. Manufacturing experts (including forging experts) are those who actually work with the machines on the shop floor. They do not usually receive formal scientific training and have poor articulation ability. Although they can solve a complex problem easily, they usually have difficulty explaining why and how they make a particular decision. Therefore, interview-based knowledge acquisition methods, such as unstructured or structured interviews with domain experts, verbal protocol analysis, and concept maps, have not been very successful in the manufacturing domain [5].

Recently, the advent of artificial neural networks has stirred the imagination of many in the field of knowledge acquisition. Neural networks belong to a family of models that are based on a learning-by-example paradigm in which problem-solving knowledge is generated automatically according to actual examples presented to them. The knowledge, however, is rep-

---

Correspondence and offprint requests to: Dr. S. H. Huang, Department of Mechanical, Industrial and Manufacturing Engineering, The University of Toledo, Nitschke Hall 4006, Toledo, Ohio 43606-3390, USA. E-mail: shuang@eng.utoledo.edu

resented at a subsymbolic level in terms of connections and weights. Neural networks act like black-boxes providing little insight into how decisions are made. They have no explicit, declarative knowledge structure that allows the representation and generation of explanation structures. Thus, the knowledge captured by a neural network is not transparent to users and cannot be verified by domain experts. This creates a major problem for users of neural-network-based systems. Since the knowledge representation and problem-solving processes employed by a neural-network-based system are not readily intelligible to users, users will not be able to interact competently and efficiently with the system. In addition, users will not be confident in the reasoning and advice of the system.

In this paper, an innovative approach is developed to acquire forging process knowledge automatically by combining the learning ability of neural networks with the structured knowledge representation of rule-based systems. The approach is applied successfully to the classification of process methods used in a type of impression-die forging, namely, drop hammer forming. The paper is organised as follows. In Section 2, literature on forging process planning is reviewed briefly, including the applications of expert systems and neural networks. Techniques used for extracting rules from trained neural network are also discussed. Section 3 presents the problem of process method classification for drop hammer forming, which is the first step in process planning. In Section 4, the neural-network-rule-based integrated knowledge acquisition methodology is applied to solve the process method classification problem. Finally, conclusions are drawn regarding the utility and applicability of the integrated methodology for forging process knowledge acquisition.

2. Background

2.1 Forging Process Planning

Computer-aided process planning (CAPP) is a critical link between computer-aided design (CAD) and computer-aided manufacturing (CAM). Many workers have devoted their efforts to CAD/CAM/CAPP research over the past three decades. However, much of the work has been carried out on material removal processes, and less attention has been given to process planning and modelling of forging. The earliest attempt in developing a CAPP system for forging is probably that reported by Gokler et al. [6]. They applied the knowledge-based expert system approach, where a set of rules is collected and stored in a knowledge base. An inference engine operates on these rules to generate a forging sequence, including the most suitable process parameters for each operation. This approach was also adopted by a number of workers including Barcellona et al. [7], Alberti et al. [8], Osakada et al. [9,10], Yang and Osakada [11], Bariani and Knight [12], and Sevenler et al. [13].

As mentioned before, the development of a knowledge-based expert system requires both the efforts of a knowledge engineer and a domain expert. The knowledge engineer interviews the domain expert, extracts the domain expert's knowledge, and transfers the knowledge into an expert system. Forging experts are usually shop floor machinists who are not articulate; thus, rendering interview-based knowledge acquisition techniques inadequate. Therefore, alternative knowledge acquisition methods are being sought. One interesting approach is the application of neural networks. Since neural networks can be trained with examples to solve a problem, they have been applied to the acquisition of forging knowledge. Works reported in the literature include Li et al. [14], Alberti et al. [15,16], Barcellona [17], and Osakada and Yang [18].

Although neural networks excel in automated knowledge acquisition, they do not have a humanly understandable knowledge representation. In neural networks, knowledge is not represented symbolically, but in the form of distributed processing and localised decision rules [19]. This subsymbolic representation makes neural networks more resistant to noise than the traditional symbolic representation. However, the representation is very difficult for a human user to comprehend. As previously mentioned, neural networks have been regarded as black-boxes, and user trust becomes a major issue. Engineers and operators from the forging industry are usually skeptical about neural-network-based techniques because of their opacity. This hampers the wide application of neural networks in forging knowledge acquisition. It is becoming increasingly apparent that neural networks alone are not sufficient for knowledge acquisition and a humanly intelligible representation of knowledge acquired by neural networks must be developed. Hence, workers are interested in extracting rules from trained neural networks. Extracted rules provide an explanation capability for neural networks, and have potential to unravel the neural network black-box mystery.

2.2 Rule Extraction from Trained Neural Networks

Andrews et al. [20] classified rule extraction techniques into two categories, namely, decompositional and pedagogical. The decompositional approach focuses on extracting rules at the level of individual neurons within a trained neural network, whereas the pedagogical approach aims to extract rules that map inputs directly into outputs. Rules extracted using these approaches are crisp rules. Several studies showed that under certain assumptions, a neural network could be approximated to any degree of accuracy using a fuzzy system, and vice versa [21,22]. Therefore, there is interest in integrating neural networks with fuzzy systems to facilitate rule extraction, which falls into a third category. An in-depth survey of literature relevant to rule extraction is summarised in Table 1.

To test different rule extraction techniques, a case study was conducted using the *N-M-N* encoder/decoder problem [38]. An *N-M-N* encoder/decoder means a three-layer neural network with N neurons for the input and output layers, and M neurons for the hidden layer. The network is given with N distinct input patterns, for each of which only one bit is turned on, and all other bits are turned off. The network should duplicate the input pattern in the output layer. The result showed that decompositional rule extraction techniques are more robust because they are developed based on theoretical analysis of the behaviour of neurons in a neural network. A significant advantage of the decompositional approach is its transparency, since it relies on the analysis of the connections and weights

Table 1. Literature relevant to rule extraction.

Technique	Reference	Decompositional	Category Pedagogical	Fuzzy
RuleNet	McMillan et al. [23]	✓		
Subset	Fu ([24,25], Towell and Shavlik [26]	✓		
M-of-N	Towell and Shavlik [26], Maire [27]	✓		
RULEX	Andrews and Geva [28,29]	✓		
Similar-weight	Saito and Nakano [30]		✓	
VLA	Thrun [31]		✓	
Extraction-as-learning	Craven and Shavlik [32]		✓	
REFuNN	Kasabov [33]			✓
NEFCLASS	Nauck and Kruse [34,35]			✓
Fuzzy-MLP	Mitra [36]			✓
FuNe I	Halgamuge and Glesner [37]			✓

of a trained neural network. In terms of shedding light to the neural network black-box mystery, the decompositional approach is more desirable than the pedagogical approach. Unfortunately, the majority of decompositional rule extraction research focuses on replicating the performance of the underlying neural network instead of the transparency of the extracted rules. As a result, the extracted rules are often in such an obscure form that they do not make any sense to a domain engineer.

3. Drop Hammer Forming

Impression-die forging, often referred to as closed-die forging, accounts for the bulk of commercial forging production. Drop hammer forming is a type of impression-die forging that is used extensively in the production of aircraft engine nacelles. Figure 1 shows a drop hammer forming station. Two matching dies are used. The bottom die, fixed at the station table (anvil), is called a tool. The top die, attached to the hammer head, is called a punch. Drop hammer forming derives its forming forces from the dead-weight fall of the hammer head and punch. The type of stress applied is largely controlled by the configuration of the die. The use of traps and beads in an excess area of the die locks the part material and causes the material to be stretched or drawn into the die cavity. The metal-to-metal contact of the punch to the tool provides the compressive action. This dual forming ability provides a means for forming shapes that would be difficult by any other method.

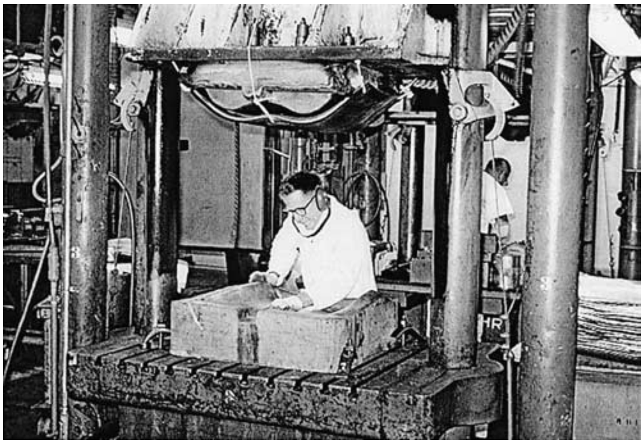

Fig. 1. A drop hammer forming station.

The drop hammer forming process methods can be classified into three categories:

1. Bare punch.
2. Blow down.
3. Blow up.

Bare punch is the simplest procedure. Parts are produced using a single hammer blow and no forming aids are needed. Blow down and blow up procedures require the use of forming aids and multiple punch blows (Fig. 2). The most commonly used forming aids are made of rubber, since this has the ability to flow like liquid when squeezed or impacted. The rubber sheets

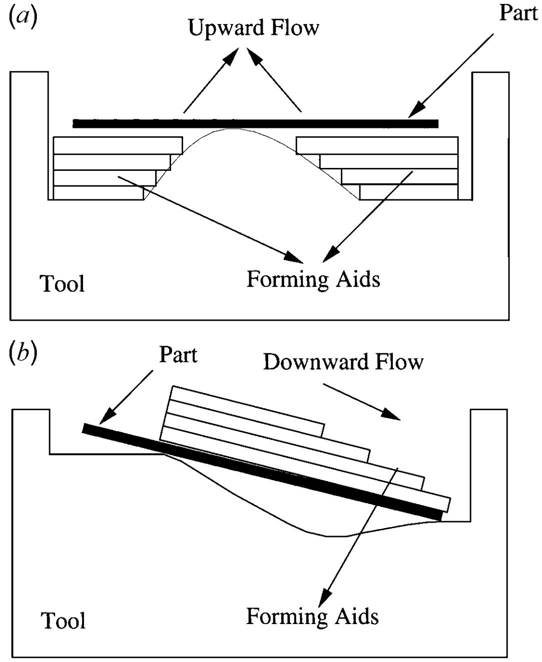

Fig. 2. Process methods that use forming aids: (*a*) blow up, and (*b*) blow down.

are placed on (blow down) or under (blow up) the part to form a pyramid. During the forming process, the pyramid causes the impact force to flow in the direction of the least rubber build-up, thus preventing wrinkles and folding. After each punch blow, one or two sheets of rubber will be removed before the next punch blow is applied. The distance of the punch blow may vary depending on the part and die configuration.

The determination of the process method is the first step in generating a process plan for drop hammer forming. Experienced operators could look at a certain part and die configuration, and choose a process plan quickly. However, they were unable to clearly explain the rationale in their decisions. Instead, they attributed their ability to "many years of experience with a lot of knowledge involved". Fortunately, they did identify several key factors that they paid close attention to when determining the process plan. These factors are part material type, type of tool (male or female), maximum draw depth of the tool, and number of corners of the part. They influence the drop hammer forming process as discussed in the following.

Part material type. Three types of part material are used frequently, namely, aluminium, steel, and titanium. The material of the part can determine the selection of different types of rubber as forming aids. For example, because titanium is very hard, usually the part is heat treated (above 1100°F for 10 min) before the forming process is applied. As a result, harder and heat resistant rubbers are needed as forming aids.

Type of tool. Generally, there are two types of tool according to their geometric configuration, namely, male and female. The geometric configuration determines where forming aids (rubber sheets) can be placed, i.e. on or under the part. The determination of tool type is not always straightforward. Not only the tool geometry but also the part geometry must be considered. For example, consider the two tools shown in Fig. 3. The tool in Fig. 3(*a*) is a male tool since the part is formed using the rib located in the middle of the tool. The tool in Fig. 3(*b*) is a female tool, because the part will be drawn into the cavity during forming.

Maximum draw depth. The maximum draw depth, to some extent, decides on the number of hits the operators should apply during the drop hammer forming process. Softer materials, such as aluminium and mild steel, can be drawn in one stage to around 2 in. However, hard material, such as titanium, can only be drawn in one stage to about 0.5 in.

Number of corners. The number of corners is determined by the shape of the part to be produced. Sometimes, the number of corners can be determined by simply looking at the tool. For example, the tool shown in Fig. 3(*b*) is used to produce parts with 4 corners. However, in other circumstances, the determination of the number of corners requires checking both the tool and the part. For example, the tool shown in Fig. 4(*a*) appears to have 7 corners. However, knowing that the part is produced as shown in Fig. 4(*b*), we can conclude that the number of corners is actually 3. These three corners are:

1. The sharp corner at the left.
2. The smooth corner at the upper right.
3. The small corner at the lower right (around the bulge at the bottom of tool shown clearly in Fig. 4*(a*).

The four corners of the tool are not counted because they are not used to form the part.

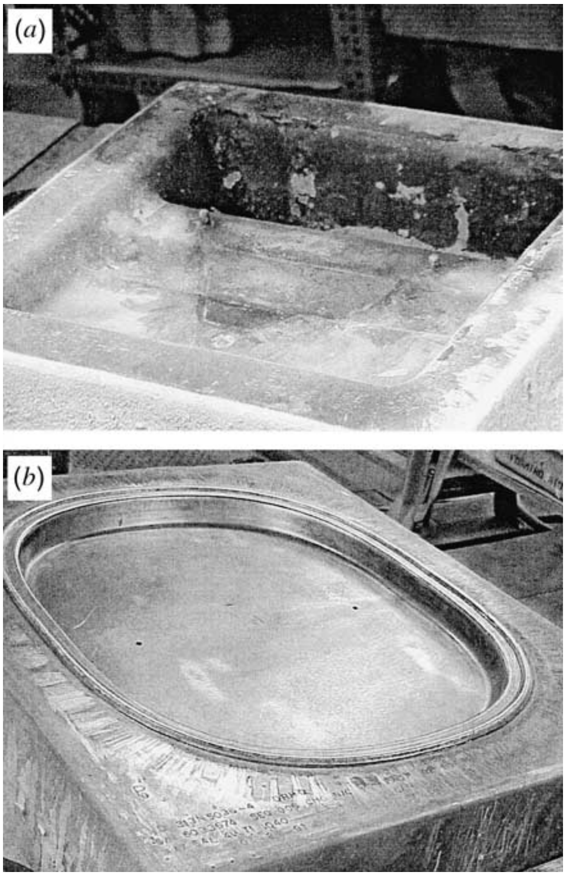

Fig. 3. (*a*) A male tool. (*b*) A female tool.

4. Extraction of Classification Knowledge

As mentioned previously, although expert machinists could help to identify major factors that influence their decisions, they were unable to provide any concrete IF-THEN rules to classify drop hammer forming process methods. In this section, a neural network based approach is used to acquire the classification knowledge automatically from empirical data.

4.1 Methodology

The methodology builds on and enhances previous decompositional rule extraction approaches by introducing the concept of linguistic terms into data processing to serve as a basis for extracting easy to understand IF-THEN rules. Recall that the binary output of an artificial neuron is defined as:

$$O = f(net) = \begin{cases} 1, & net > 0 \\ 0, & net < 0 \end{cases}$$

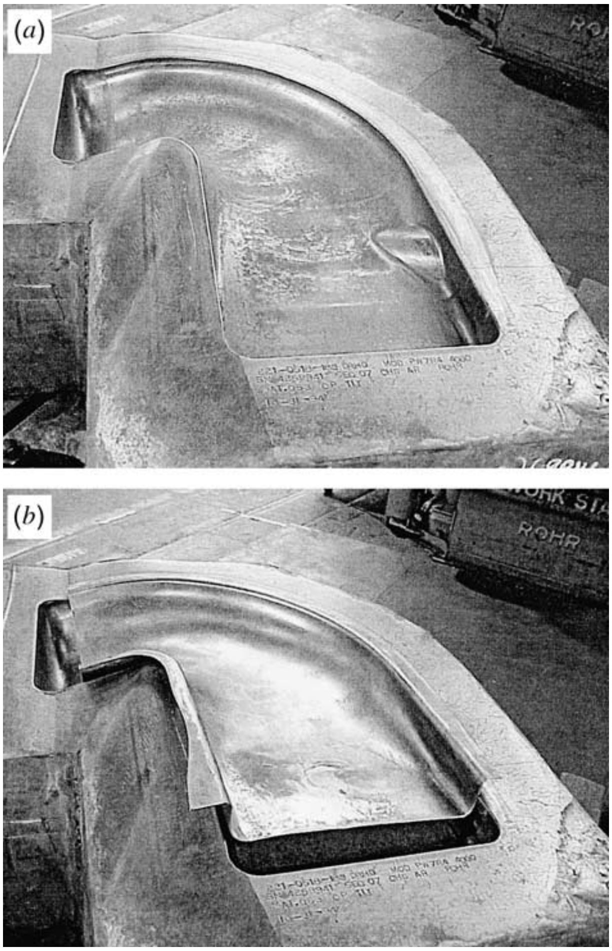

Fig. 4. Corners of part: (*a*) tool, and (*b*) part to be formed by the tool.

in which

$$net = \sum_{i=1}^{n} w_i x_i - B$$

Therefore, if a certain combination of weighted input values of a neuron is greater than its bias, then the status of the neuron is on; otherwise, its status is off. Based on this neuron behaviour, rules can be extracted if the input and output parameters of the neural network are binary [24–26].

In real-world applications, continuous-valued parameters are involved most often. To deal with this problem, the following method is used:

Step 1. Classify the continuous-valued inputs into sets described using a linguistic term.

Step 2. Represent the sets using a binary scheme.

Step 3. Construct a neural network with binary inputs.

Step 4. Train the constructed neural network.

Step 5. Extract rules from the trained neural network based on the behaviour of neurons.

The data conversion scheme can also be applied to output parameters. Details of the data conversion procedure can be found in [39]. Since different classification methods result in different numbers of linguistic terms being used, a generalised representation is used to describe the result of data conversion. The following notation is used:

N = number of input parameters of the original data set
M = number of output parameters of the original data set
I_i = input parameter i ($i = 1, 2, \ldots, N$)
O_j = output parameter j ($j = 1, 2, \ldots, M$)
L_i = number of linguistic terms used to describe I_i
K_j = number of linguistic terms used to describe O_j
$T(I_i, a)$ = ath linguistic term used to describe I_i ($a = 1, 2, \ldots, L_i$)
$T(O_j, b)$ = bth linguistic term used to describe O_j ($b = 1, 2, \ldots, K_j$)

Once the neural network is trained using the converted data, intelligible rules can be extracted. Figure 5 shows a partial neural network with one output neuron that stands for the consequent "O_j is $T(O_j, b)$." Since, if the sum of a neuron's weighted input is greater than its bias then the neuron is on, the problem of extracting antecedents that will result in the consequent "O_j is $T(O_j, b)$" can be formulated as follows:

Find all x_{ia} that satisfy

$$\sum_{i=1}^{N} \sum_{a=1}^{L_i} x_{ia} \times w_{ia} - B \geq 0$$

subject to

$$x_{ia} \in \{0, 1\} \tag{1}$$

$$\sum_{a=1}^{L_i} x_{ia} = 1, \forall\, i \tag{2}$$

Constraint (1) reflects the fact that after data conversion, the inputs are binary. Constraint (2) ensures that a certain value of an input parameter can be described using one linguistic term only as restricted by the data conversion procedure.

An algorithm, based on a dynamic depth-first search principle, has been developed to solve the rule extraction problem.

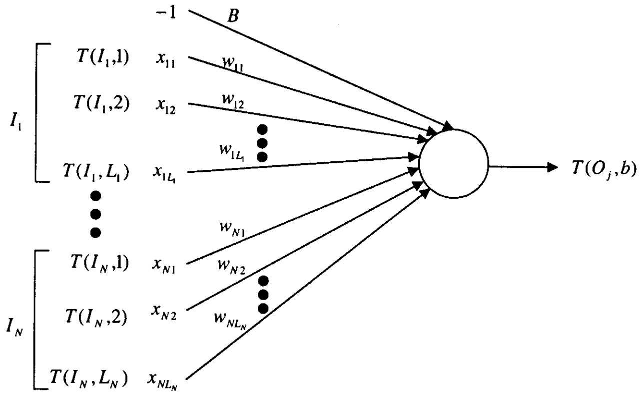

Fig. 5. A binary neural network with one output neuron.

In order to reduce the search space, all the negative weights of the neural network are converted into positive ones. The procedure is based on the weight transformation theory discussed in [40,41]. Details of the procedure can be found in [42]. After conversion, the weights e_{ij} are sorted in descending order to facilitate the dynamic depth-first search. A rule tree is constructed as follows. The root is established with L_1 weighted edges ($e_{11}, e_{12}, \ldots, e_{1L_1}$), thus determining the first level nodes (there are L_1 nodes, denoted, $v_1, v_2, \ldots, v_{L_1}$ respectively). Each first-level node has L_2 children nodes with weighted edges $e_{21}, e_{22}, \ldots, e_{2L_2}$. Therefore, there are $L_1 \times L_2$ second-level nodes, denoted $v_{11}, v_{12}, \ldots, v_{1L_2}, v_{21}, v_{22}, \ldots, v_{2L_2}, \cdots, v_{L_1L_2}$, respectively. Continuing the process until the last level (the Nth level), we will have $\sum_{i=1}^{N} L_i$ leaves (nodes that have no children). It can be seen that the time required to traverse the rule tree completely increases drastically with the increase of the number of input parameters N and the number of linguistic terms used for each parameter (L_i, $i = 1, 2, \ldots, N$). Fortunately, converting and sorting the weights allow us to terminate the search without fully exploring the rule tree. Since all the weights are positive, if we encounter a node that results in a rule (i.e. the sum of weights of the traversed edges is greater than or equal to the bias B), then we do not need to explore its child nodes further. The search procedure can be further simplified. Note that for each parameter i, $e_{i1}, e_{i2}, \ldots, e_{iL_i}$ are sorted in descending order. Therefore, if we reach a leaf (say, $v_{a_1a_2,\ldots,a_N}$), and the sum of weights of the traversed edges is less than the bias (i.e. $\sum_{i=1}^{N} e_{ia_i} - B < 0$), then a leaf $v_{b_1b_2,\ldots,b_N}$ (called a less promising leaf) does not need to be traversed if $b_i > a_i$, $\forall\ i \in \{1, 2, \ldots, N\}$. This means that the rule tree can be created dynamically. As a result, a dynamic depth-first search algorithm for rule extraction is developed as follows:

```
procedure dynamic_depth-first_search(e, B)
{e – converted weight matrix; B – bias}
  {let $v_{t_1t_2,\ldots,t_N}$ denote the target leave to be explored}
  initialise all $t_i$ to 1, in which $i = 1, 2, \ldots, N$ {the left most leaf}
    loop
      $s \leftarrow -B$; $i \leftarrow 1$
      while ($s < 0$ and $i \le N$) do   $s \leftarrow s + e_{it_i}$ {traverse the edges}
                                          $i \leftarrow i + 1$
      if $s \ge 0$ then {a rule is found}
        report $v_{t_1t_2,\ldots,t_{i-1}}$ as a rule
        for $j \leftarrow i$ to $N$
          $t_j \leftarrow L_j$ {no need to explore child nodes}
      else {rule not found when reaching a leaf}
        {no need to explore less promising leaves}
        $j \leftarrow N$
        while ($j \ge 1$ and $t_j = 1$) do   $t_j \leftarrow L_j$
                                            $j \leftarrow j - 1$
        if $j = N$ then
          $t_N \leftarrow L_N$
      $j \leftarrow N$; $t_j \leftarrow t_j + 1$ {advance to the next leaf}
      while ($t_j = L_j + 1$ and $j > 1$) do   $t_j = 1$
                                               $j \leftarrow j - 1$
                                               $t_j \leftarrow t_j + 1$
      if ($t_1 = L_1 + 1$) then
        exit loop {no more leaves to traverse}
      end loop
```

The dynamic depth-first search algorithm will extract all possible rules from a trained neural network. These rules have different levels of importance. The importance of a rule can be represented using its corresponding activation value, denoted as the rule's weight. For a rule represented by $v_{t_1t_2,\ldots,t_i}$, its weight is calculated as follows

$$w = \begin{cases} e_{1t_1} + e_{2t_2} + \ldots + e_{it_i}, \text{ if } i = N \\ e_{1t_1} + e_{2t_2} + \ldots + e_{it_i} + e_{(i+1)1} + \ldots + e_{N1}, \text{ if } i < N \end{cases}$$

To improve the generalisation ability of the extracted rules, we can set a threshold value, T, to control the number of important rules. In other words, if the weight of a rule is greater than T, then the rule is reported as an important rule. A general practice is to set $T = 2.197$, which results in a neuron output of 0.9 when a unipolar sigmoid function, i.e. $f(net) = 1/(1 + \exp(-net))$, is used as the neuron activation function.

4.2 Results

The methodology previously described was applied to extract knowledge for classifying drop hammer forming process methods. Twenty data samples were collected, as shown in Table 2. Three linguistic terms are used to describe part material type, namely, soft, medium, and hard. Aluminium is soft, steel is medium, whereas titanium is hard. Two linguistic terms, small and large, are used to describe maximum draw depth. If a maximum draw depth is less than 2.875, then it is small; otherwise, it is large. Number of corners is described using three linguistic terms, namely, none (0), low (1, 2, and 3), and high (4, 5, and 6). Two linguistic terms, male and female, are used to describe type of tool. These linguistic terms are determined with the help of expert operators.

The converted data samples are used to train a neural network. It was found that a neural network without any hidden neurons could be trained to reduce the error to 0.001 after 1926 epochs. After weight conversion, the dynamic depth-search algorithm is applied and the following rules are extracted:

Rule 1. IF (*Number of corners* is None) AND (*Type of tool* is Male) THEN (*Process method* is Bare punch) (Weight 4.73)

Rule 2. IF (*Maximum draw depth* is Small) AND (*Part material type* is Soft) AND (*Number of corners* is None) THEN (*Process method* is Bare punch) (Weight 0.81)

Rule 3. IF (*Maximum draw depth* is Small) AND (*Part material type* is Medium) AND (*Number of corners* is None) THEN (*Process Method* is Bare punch) (Weight 0.53)

Table 2. Drop hammer forming process data.

Sample	Part number	Part material type	Maximum draw depth	Number of corners	Type of tool	Process method
1	221-0130-11	Steel	1.00	3	Female	Blow down
2	221-0518-13	Titanium	2.50	3	Female	Blow down
3	221-0518-7	Titanium	0.25	1	Female	Blow down
4	221-0518-9	Titanium	2.00	1	Female	Blow down
5	221-0526-31	Titanium	3.00	4	Male	Blow up
6	221-0526-35	Titanium	3.00	2	Female	Blow down
7	290-0031-515	Aluminium	2.00	0	Male	Bare punch
8	313N5036-4	Titanium	2.50	4	Female	Blow down
9	313N5094-8	Titanium	2.50	2	Male	Blow up
10	5938058-93	Steel	0.75	4	Female	Blow down
11	5938068-53	Aluminium	0.50	2	Female	Blow down
12	5958896-15	Titanium	0.75	2	Female	Blow down
13	65C18554-765	Steel	1.44	0	Male	Bare punch
14	715-0383-501	Steel	1.00	3	Female	Blow down
15	740-0515-1581	Aluminium	1.00	1	Female	Blow down
16	LJ71232-1	Titanium	0.50	2	Female	Blow down
17	LJ75051-35	Steel	5.50	4	Female	Blow down
18	LJ75051-60	Steel	1.25	6	Female	Blow down
19	LJ75055-81	Steel	0.50	1	Female	Blow down
20	LJ75639-502	Titanium	0.25	1	Female	Blow down

Rule 4. IF (*Type of Tool* is Female) THEN (*Process method* is Blow down) (Weight 5.84)

Rule 5. IF (*Part material type* is Hard) AND (*Number of corners* is Low) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up) (Weight 4.48)

Rule 6. IF (*Part material type* is Hard) AND (*Number of corners* is High) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up) (Weight 4.45)

Rule 7. IF (*Maximum draw depth* is Large) AND (*Part material type* is Hard) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up) (Weight 0.25)

Rule 8. IF (*Maximum draw depth* is Large) AND (*Part material type* is Soft) AND (*Number of corners* is Low) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up) (Weight 0.20)

Rule 9. IF (*Maximum draw depth* is Large) AND (*Part material type* is Soft) AND (*Number of corners* is High) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up) (Weight 0.17)

Rule 10. IF (*Maximum draw depth* is Large) AND (*Part material type* is Medium) AND (*Number of corners* is Low) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up) (Weight 0.17)

Rule 11. IF (*Maximum draw depth* is Large) AND (*Part material type* is Medium) AND (*Number of corners* is High) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up) (Weight 0.14)

Rules 2, 3 and 7 to 11 are dropped because they have very small weights. These rules are actually valid but too specific. For example, experts regarded rule 2 as correct, but they pointed out that parts with large maximum draw depth can also be produced using bare punch as long as they do not have any corners (Fig. 3(*a*) shows the tool for such a part). Rules 5 and 6 can be combined. As a result, three important rules are extracted, one for each process method. These rules are:

Rule A. IF (*Number of corners* is None) AND (*Type of tool* is Male) THEN (*Process method* is Bare punch)

Rule B. IF (*Type of Tool* is Female) THEN (*Process method* is Blow down)

Rule C. IF (*Part material type* is Hard) AND (*Number of corners* is NOT None) AND (*Type of tool* is Male) THEN (*Process Method* is Blow up)

Upon examining these rules, we can see that maximum draw depth is not included in the rule premises, which means it is not important in determining process methods. However, they were identified by expert machinists as key factors in process planning for drop hammer forming. When this issue was discussed, the expert machinists were quick to point out that maximum draw depth is a critical factor in determining the number of punch blows in blow down and blow up. They agree that it is probably not important in determining the process methods.

5. Conclusion

The acquisition of process knowledge is critical in developing a computer-aided forging process planning system. This paper presents an innovative approach to facilitate process knowledge acquisition from domain experts. The approach is applied to the classification of drop hammer forming process methods with promising results. It relies on the learning and generalisation ability of neural networks to capture heuristic process knowledge automatically, and uses a rule-based paradigm for transparent knowledge representation. The extracted process knowledge is represented using highly intelligible and intuitive rules. These rules can be verified easily by domain experts.

They can then be used to:

1. Construct a computerised system to allow clearer and consistent process specification.
2. Educate novice operators to shorten their learning curves.

The neural-network-rule-based integrated approach for automated process knowledge acquisition provides a new dimension in process planning and modelling of the forging process, where traditional knowledge acquisition methods have not been very successful. Currently, the approach is applied only to classification problems. In the future, we will investigate the application of the approach to deal with function approximation problems, e.g. determining the height of punch blow.

Acknowledgements

This work is supported in part by the Ohio Aerospace Institute (OAI) under the 1999 Collaborative Core Research Program Phase II award (CCRP 98-1-002) and a summer research internship from B. F. Goodrich Aerospace/Aerostructures Group. The authors would like to thank Mr Nigel Barker and Mr Dan Stoll for serving as industrial supervisors for this project. Special appreciation is given to the late Mr Leo Paulino, who was instrumental in helping the research team by sharing his expertise in drop hammer forming. The inputs from shop operators are also acknowledged.

References

1. Energetics Inc., "Forging industry technology roadmap", sponsored by Forging Industry Association, Forging Industry Educational and Research Foundation, and the US Department of Energy/Office of Industrial Technologies, November 1997.
2. A. J. Gonzalez and D. D. Dankel, The Engineering of Knowledge-Based Systems: Theory and Practice, Prentice-Hall, Englewood Cliffs, NJ, 1993.
3. F. Hayes-Roth, D. Waterman and D. B. Lenat, Building Expert Systems, Addison-Wesley, Reading, MA, 1983.
4. E. A. Feigenbaum, "Themes and case studies of knowledge engineering", in D. Michie (ed.), Expert Systems in the Micro-Electronic Age, Edinburgh University Press, pp. 3–25, 1979.
5. I. Ham and S. C.-Y. Lu, "Computer-aided process planning: the present and the future", Annals CIRP, 37(2), pp. 1–11, 1988.
6. M. I. Gokler, T. A. Dean and W. A. Knight, "Computer-aided sequence design for hot upset forgings", The Twenty Second International Machine Tool Design and Research Conference, Manchester, pp. 457–466, 16–18 September 1982.
7. A. Barcellona, F. Micari and F. Satmè, "Computer-aided choice of working sequences", The Third International Conference on Advanced Manufacturing Systems and Technology, Udre, pp. 137–143, 26–27 April 1993.
8. N. Alberti, L. Cannizzaro and F. Micari, "Knowledge-based systems and FE simulation in metal forming process design: an integrated approach", Annals CIRP, 40, pp. 295–298, 1991.
9. K. Osakada, T. Kado, T. Nakamura and K. Mori, "Expert system for cold forging process based on FEM simulation", Annals CIRP, 39, pp. 249–252, 1990.
10. K. Osakada, T. Kado and G. B. Yang, "Application of AI-technique to process planning of cold forging", Annals CIRP, 37, pp. 239–242, 1988.
11. G. B. Yang and K. Osakada, "An expert system for process planning of cold forging", Advanced Technology of Plasticity, 1, pp. 109–114, 1990.
12. P. Bariani and W. A. Knight, "Computer-aided cold forging process design: a knowledge-based system approach to forming sequence generation", Annals CIRP, 37, pp. 243–246, 1988.
13. K. Sevenler, P. S. Raghupathi and T. Altan, "Forming sequence design for multistage cold forging", Journal of Mechanical Working Technology, 14, pp. 121–135, 1987.
14. M. Li, X. Zhang, X. Liu, S. Wu, F. Zeng and Z. Cui, "Self-adaptive prediction of microstructure and performance after forging with die", *Journal of Materials Engineering* 1, pp. 42–45, 1998.
15. N. Alberti, R. Di Lorenzo, F. Micari, R. Teti, P. Buonadonna and A. Manzoni, "Intelligent computation techniques for process planning of cold forging", Journal of Intelligent Manufacturing, 9, pp. 353–359, 1998.
16. N. Alberti, A. Barcellona, F. Micari and R. Teti, "Neural network application to cold forging", The Second AITEM Conference, Padua, pp. 21–30, 18–20 September 1995.
17. A. Barcellona, "Neural network techniques for metal forming design", The Thirtieth International Machine Tool Design and Research Conference, Manchester, pp. 361–365, 31 March–1 April 1993.
18. K. Osakada and G. B. Yang, "Neural networks for process planning of cold forging", Annals CIRP, 40, pp. 243–246, 1991.
19. Y.-H. Pao and D. J. Sobajic, "Neural networks and knowledge engineering", IEEE Transactions on Knowledge and Data Engineering 3(2), pp. 185–192, 1991.
20. R. Andrews, J. Diederich and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks", Knowledge-Based Systems, 8(6), pp. 373–389, 1995.
21. J. M. Benitez, J. L. Castro and I. Requena, "Are artificial neural networks black boxes?" IEEE Transactions on Neural Networks, 8(5), pp. 1156–1164, 1997.
22. J. J. Buckley, Y. Hayashi and E. Czogala, "On the equivalence of neural nets and fuzzy expert systems", Fuzzy Sets and Systems, 53, pp. 129–134, 1993.
23. C. McMillan, M. C. Mozer and P. Smolensky, "The connectionist scientist game: rule extraction and refinement in a neural network", Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, Hillsdale, NJ, 1991.
24. L. M. Fu, "Rule learning by searching on adapted nets", Proceedings of the Ninth National Conference on Artificial Intelligence, Anaheim, CA, 1991.
25. L. M. Fu, "Rule generation from neural networks", IEEE Transactions on Systems, Man, and Cybernetics, 28(8), pp. 1114–1124, 1994.
26. G. G. Towell and J.W. Shavlik, "Extracting refined rules from knowledge-based neural networks", Machine Learning, 13, pp. 71–101, 1993.
27. F. Maire, "A partial order for the M-of-N rule-extraction algorithm", IEEE Transactions on Neural Networks, 8(6), pp. 1542–1544, 1997.
28. R. Andrews and S. Geva, "Inserting and extracting knowledge from constrained error back propagation networks", Proceedings of the Sixth Australian Conference on Neural Networks, Sydney, Australia, 1995.
29. R. Andrews and S. Geva, "Rule extraction from a constrained error back propagation MLP", Proceedings of the Fifth Australian Conference on Neural Networks, Brisbane, Australia, 1994.
30. K. Saito and R. Nakano, "Medical diagnostic expert system based on PDP model", Proceedings of IEEE International Conference on Neural Networks, San Diego, CA, 1988.
31. S. B. Thrun, "Extracting provably correct rules from artificial neural networks", Technical Report IAI-TR-93-5, Institute for Informatik III Universitat, Bonn, Germany, 1994.
32. M. W. Craven and J. W. Shavlik, "Using sampling and queries to extract rules from trained neural networks", Machine Learning: Proceedings of the Eleventh International Conference, San Francisco, CA, 1994.
33. N. K. Kasabov, "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems", Fuzzy Sets and Systems, 82, pp. 135–149, 1996.

34. D. Nauck and R. Kruse, "Obtaining interpretable fuzzy classification rules from medical data", Artificial Intelligence in Medicine, 16, pp. 149–169, 1999.
35. D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data", Fuzzy Sets and Systems, 89, pp. 277–288, 1997.
36. S. Mitra, "Fuzzy MLP based expert system for medical diagnosis", Fuzzy Sets and Systems, 65(2/3), pp. 285–296, 1994.
37. S. K. Halgamuge and M. Glesner, "Neural networks in designing fuzzy systems for real world application", Fuzzy Sets and Systems, 65(1), pp. 1–12, 1994.
38. S. H. Huang and M. R. Endsley, "Providing understanding of the behavior of feedforward neural networks", IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, 27(3), pp. 465–474, 1997.
39. S. H. Huang and H. Xing, "Heuristic Manufacturing Knowledge Modelling: A Neural Network/Fuzzy Logic Hybrid Approach", 4th International Conference on Frontiers of Design and Manufacturing, Hangzhou, P.R. China, pp. 80–90, June 16–19, 2000.
40. R. Krishnan, G. Sivakumar and P. Bhattacharya, "A search technique for rule extraction from trained neural networks", Pattern Recognition Letters, 20, pp. 273–280, 1999.
41. I. K. Sethi and J. H. Yoo, "Symbolic mapping of neurons in feedforward networks", Pattern Recognition Letters, 17, pp. 1035–1046, 1996.
42. S. H. Huang, H. Xing and M. Benjamin, "Automated knowledge acquisition for design and manufacturing: the case of micromachined atomizer", Journal of Intelligent Manufacturing, accepted.