Intelligent Decision Support Technologies for Design and Manufacturing†

Thomas Zacharia, John D. Allen, Nenad Ivezic, and Gerard M. Ludtka

Modeling and Simulation Group, Metals and Ceramics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6140 USA
(423) 574-4897 FAX (423) 574-4839

Abstract

For many of today's complex manufacturing processes, there exists a solid body of knowledge that enables direct simulations of such processes yielding predictions about the final product and process characteristics using finite element or finite difference methods. However, the computational complexities of these simulations are such that they do not lend themselves easily to routine and timely use in optimization and control of manufacturing processes. More recently, neural network-based decision support technologies have been developed which hold the promise of bringing the body of analytical and simulation knowledge closer to the design and optimization processes in manufacturing industries. The paper discusses the application of a holistic approach wherein existing finite element, neural-network, and optical metrology methods are combined to develop a real time tool for optimization and control of the sheet metal stamping process. Significant issues in the development of such a tool and results from its application to a deformation process are discussed.

Introduction

Among the many new computational methods developed during the last decade or so are those best described by the term "connectionist." Many of these have come to be referred to by the term "artificial neural network" or, more often, simply "neural network." [1] The label arises from the asserted similarity between the individual computational elements of artificial neural networks and the fundamental entities of the human brain, the biological neurons and the dendrites and axons which support the principal mechanisms of biological neuron interconnection and activation. For most applications of relevance to materials design and process optimization, typical artificial neural networks are reasonably well defined by the following.

Definition: Artificial Neural Network

An artificial neural network is a modifiably interconnected set of active, generally non-linear, elements (accurately or not, usually called neurons) which, in some fashion, accept input signals from their environment and which both return to that environment some indication of their collective response to these signals and adjust the relevant characteristics of their interconnectivity in a manner which tends to increase the network's capacity for giving unique responses to stimuli with which, by training, it becomes increasingly familiar.

†The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-96OR22464. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
Some of the properties of neural networks which can confer advantage in materials design and process optimization applications are noted below.

1) Properly prepared neural networks are capable of solving computationally intractable "Inverse Problems." The metal forming example presented further on illustrates this capability.

2) Neural networks can codify implicit as well as explicit relationships in data. If, to pick an example from the physics of ballistic flight, a network is exposed to sparse, but well distributed, experimental data representing the flight of a cannon-fired projectile in air (data representing initial muzzle velocity, muzzle elevation angle, projectile impact point, etc.), the network will learn, as one might expect, to predict impact points for values of muzzle velocity and elevation not included in the training data as long as these values lie within or reasonably near the trained-for ranges. More importantly, the network will develop a functional representation for effects not explicitly expressed in the training data (here, air resistance, variation of g with height, etc.).

3) Neural networks can learn the often non-linear and generally time-dependent relationships between "in situ" parameter measurements of generally inaccessible signals and measurements made at points more readily accessible in an industrial setting. This point alone can be carried to great advantage in a variety of circumstances under which processing conditions make direct measurement of important variables impossible or excessively costly to effect.

4) By virtue of their ability to capture the very complex and usually at best approximately known relationships defining large-scale industrial processes, neural networks can be employed for predicting the evolution of the state variables defining process behavior. Thus, both process control and failure prediction fall within the purview of properly prepared networks.

5) Related to (4), but well worth mentioning in its own right, is a capability conferred by neural networks for performance tracking. One of the most important considerations in a modern industrial setting is the support of precisely that maintenance (and no more!) required for continued effective plant operation. Because neural networks properly incorporated into an industrial system will have been exposed to representative examples of acceptable process behavior, deviations from normal operating conditions, whether recognized by type and cause or not, will be readily noted. Moreover, these deviations will, in most cases, be apparent long before catastrophic failure occurs. An adaptive maintenance schedule based on such a neural system can substantially improve plant reliability and productivity.

6) Neural networks of many varieties can be arranged to deliver acceptable performance in the presence of both noisy and incomplete data. This capability is of great utility in an environment such as the automotive welding shop, wherein the data are likely to be frequently incomplete and almost always corrupted by high levels of arc- and emf-induced noise.

Two Representative Applications

1. Iron Aluminide Property Prediction

The Iron Aluminide (FeAl) studies were designed as a test of the capabilities of non-recurrent (i.e., static) neural networks to deal with experimental data in a complex domain for which little if anything in the way of a reliable theoretical foundation exists. The networks employed were fairly conventional, backpropagation-trained, multi-layer, perceptron systems of fifteen input nodes and three output nodes and incorporated a variation of the global gradient descent learning mechanism. Training data were extracted from a larger database comprising experimental results representing Yield Strength, Ultimate Strength, and Elongation as a function of alloy composition for a series of Iron Aluminide alloys. For the studies noted here, the fifteen network input nodes represented the
magnitudes of the various (non-Fe) alloy constituents (Al, Ti, B, TiB2, Mo, Zr, Ce, Cr, Nb, Y, Mn, C, V, Be, Si) for a single temperature (600°C). The three network output nodes corresponded to the three properties (Yield Strength, Ultimate Strength, and Elongation) whose predictions the network(s) was trained to make.

Network training was performed according to one or the other of two regimens. Under the first, a single network was created, trained, and tested in the context of a training set representing one hundred twenty or so different alloy compositions. If trained network behavior was deemed inadequate, a new system was generated and the process repeated. Under a second training scheme, it was arranged to process a group of disparate networks "simultaneously," subjecting each network in the group to a common sequence of training examples. When training had progressed sufficiently to permit clear identification of the best-performing member of the group, this single network was selected for further training and subsequent testing. Figure 1 illustrates the trained network response realized with this method. Predictive capability for cases not employed for training was of approximately the same order.

Figure 1. Train network response for yield strength for iron aluminides.

It bears comment that one other training technique was briefly explored in connection with the Iron Aluminide studies, one based loosely on the methods of genetic algorithms. Under this scheme, multiple networks were trained in the context of a common data set and training regimen and permitted to "compete" with one another, "mating" periodically to produce offspring networks which, if their emerging capabilities so suggested, ultimately replaced their "parents," "grandparents," etc., until acceptable trained behavior obtained for some member of a "final" generation. For the FeAl application, this "genetic" approach conferred insufficient advantage to compensate for the increased complexity of its execution.

2. Neural Networks For Die Design

The goal for the die design neural network studies was to develop methods for predicting, without the usual number of costly and time-consuming design iterations, a near-net die shape suitable for forming a specified sheet metal component under defined forming conditions. The principal result
of the studies was development of a neural network-based inverse problem solving method in
which a network (or a collection of networks) presented with training data defining a "forward"
process (metal deformation in response to a forming process) learns to solve problems
representative of the "inverse" process (computing die configurations for presented part
configurations). A key point here is that the forward problems, although computationally
intensive, can usually be modeled using finite-element methods with sufficient accuracy for the
preparation of network training data. The inverse problems, generally intractable by any standard
method, can be solved by the prepared network in milliseconds.

Given the enormous quantity of data required for shape representation, it is essential to success that
data be efficiently and suitably preprocessed (i.e., compressed) before application to the neural
network(s). Even the small (64 x 64) arrays employed for preparation of the results presented here
comprise 4096 elements. Each member of the training set (of which there may be hundreds)
comprises two of these arrays, one for the metal part (the network input) and one for the
corresponding die (the network training goal).

Both Two-Dimensional Fourier Transforms [2] and Two-Dimensional Wavelet Transforms [2,3,4]
were investigated in the quest for an effective data preprocessing/data compression scheme and
each proved useful for certain classes of shape. Nevertheless, although the Two Dimensional
Wavelet compression method appears to reduce the number of data required for shape
representation by a greater factor than that obtained for the Two Dimensional Fourier method, the
reduction still proved insufficient to admit of completely general network training. Both methods
required the retention of far too many elements (hundreds) for the number of data sets for which it
was feasible to process (at most, a few hundred).

The method finally adopted for data compression is termed a Patch Representation, a method
based on the reasonable assertion that shape gradients near a point on a surface are more likely to
influence the response of material at that point to deformation forces (imposed, for instance, by a
punch and die) than are shape gradients at positions more removed from the point. Equally
reasonable is the assertion that "near" effects must be represented more completely to the neural
network than more "distant" ones if the network is to capture the important metal forming
relationships.

The Patch Method is characterized by a fixed pattern of "averaging regions" that is scanned over
the data array representing a material part (typically a sheet-metal component) of interest. In a
typical implementation, there may be 25 averaging regions of graduated sizes (12 per axis and one
central element, the smallest being one array element square, the largest of the order of 8 to 10
array elements in length by one element in width) comprising a cross shaped "Patch Geometry" of
the general form suggested by the schematic representation of Figure 2. These averaging regions
are distributed along orthogonal axes (axes typically defined by the length and breadth of the part)
and comprise in each axis such a total length that, whatever the position on the part array of the
central averaging region (itself comprising, in the usual case, exactly one array element), the full
extent of the part along an axis is covered by averaging regions along the same axis (elements of
averaging regions extending past the part array boundary are filled with zeros). It is arranged that
the value computed for each averaging region be presented to a corresponding network input node.

The training goal for a network employing the Patch Method is to produce at a single output node
a signal representing the value of the difference between part elevation and corresponding die
elevation at that point for each element of the part array. From these differences, the true die
configuration can be derived.

The Patch Method confers the advantage that a somewhat more accurate representation of the part
surface is retained in the data employed by the network than is the case with either the truncated
2D Fourier or the truncated 2D Wavelet data compression methods. Perhaps more important than
this is the fact that the network, by virtue of the fact that it treats each point on a part/die surface as
defining a distinct training example, can develop a much better codification of relationships linking deformation forces to resultant shape than can a network employed for either of the other methods. There are two penalties paid for realizing these advantages. First is the requirement that the network "visit" each of the elements of the part definition array may times during training. For the 64 x 64 case noted here, there are 4096 such elements, each of which may be visited several thousand times during a typical training session. The second penalty is the memory intensive character of the method. For each data set (of which there may be hundreds) it is necessary to store the 25 average values for each of the 4096 potential patch-center locations (for a total of 25 x 4096, or 102400 values). Nevertheless, where memory size and processing speed are not of concern, these are inconsequential issues.

Figure 2. Patch method schematic diagram.

Figure 3 illustrates a typical training data set comprising one die (the training goal, 3a) and one part (the network input, 3b). When the network has been fully trained on a suitable collection of these data sets, it is ready for predicting the die shape required for producing a user-specified part. Figure 4a illustrates such a user-supplied part. Figure 4b illustrates the die predicted by the network in response to the presented test part. Figure 4c depicts the "true" die that a perfectly trained network would have produced. It should be noted that the configuration classes of test part and the training data differ. The implied capability derives from the patch method's ability to capture general material deformation characteristics.
Figure 3. Typical training data set.

Figure 4. Comparison between predicted and actual die shapes.

Summary

The two examples presented illustrate some of the capabilities of artificial neural networks of relevance to materials design and process optimization. The important capability for dealing with large quantities of experimental data so as to codify inherent relationships and make them accessible for material property prediction and material design is well illustrated by the Iron Aluminide results. The metal stamping results provide an extreme example of a network's capability, when the data are properly prepared, for dealing with data sets representing three-dimensional shapes. Also deriving from the second example is a general method for solving complex and computationally intractable inverse problems, particularly those for which the corresponding "forward" problem can be effectively handled by finite-element methods. Both examples illustrate efficiency-enhancing, productivity-increasing, and cost-saving techniques of broad applicability in the modern industrial setting.
Acknowledgments

The authors would like to acknowledge Drs. S. Simunovic and G. Sarma for their review of the manuscript. Research was sponsored by the Division of Materials Sciences, U.S. Department of Energy, under contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation.

References


Report Number (14) CONF-970767--2

Publ. Date (11) 199706
Sponsor Code (18) DOE/ER XF
UC Category (19) UC - 400, DOE/ER