Rules and Precedentes in Legal Reasoning

JüRGEN HOLLATZ

1. INTRODUCTION

If we learn a new skill such as driving a car or riding a bicycle, it would be disastrous to start without prior knowledge about the problem. Typically, we are told some basic rules, which we try to follow in the beginning, but which are then refined and altered through experience. The better our initial knowledge about a problem, the faster we can achieve good performance and the less training is required. It is highly desirable to exploit similar ideas when training a neural network. If prior knowledge can be used to pre-structure a neural network, the initial performance of the network is improved and reaching satisfactory performance requires less training time and fewer training exemplars. Also, in cases where not enough data are available, especially in networks with a high-dimensional input space, prior knowledge can be used to constrain the degrees of freedom. This technique could be used in the area of legal reasoning by analogy. A couple of prototypical court decisions are available as well as a few rules defined by an expert. For judging a new case, a jurist would compare the new facts with precedents or decide how far a new case fits under a rule. In addition, the algorithm provides a tool to extract the structure of the precedent decisions. If the network learns only the behavior of prototypical decisions, the extracted rules provide a symbolic formulation of the reasoning process.

In this paper we consider prior knowledge which can be formulated in terms of uncertain domain-specific rules. The parameter values of the premises of these rules are weighted by membership functions. As inference mechanism, we use the normalized weighted sum of the outputs of all rules. The output of the network minimizes the expected error and can be

---

* Siemens AG, ZFE ST SN 41, Corporate Research and Development, Otto-Hahn-Ring 6, 8000 München 83, Germany; email: jh@morla.zfe.siemens.de; Fax: +49 89 636 3320.
shown to be the optimal response in the framework of Bayes decision theory [HT].

The rule-based inference system can be implemented in form of a neural network of normalized basis functions, and training data can be employed to improve and refine network performance. After training the altered rules can be extracted and interpreted. Introducing new basis functions (growing network architecture) corresponds to the generation of new rules, and merging of basis functions corresponds to joining local rules to form more global rules (generalization).

Neural networks have sometimes been criticized because features which are typically available in rule-based expert systems such as programmability, conceptual capabilities and structured schemes of knowledge are missing. To get insight into working of intelligent processes is in general an important task of cognitive science. It is a fundamental contribution of neural networks that it explains how the knowledge may be distributed across the system in the connection weights, but this fact means that it is hard to 'get the knowledge out' in order specify what has been learned. It is of little use to say that the system has learned something, if the only possible specification of this knowledge is a huge table of numbers. Thus, what is required is some way of 'decompiling' the distributed knowledge. Our approach demonstrates how symbolic and subsymbolic knowledge can be combined. It allows rule-based integrated learning, rule insertion, theory refinement and rule extraction.

In this contribution, a technique of including rule-based prior knowledge to structure the network is described as well as a technique to extract symbolic information. The organization of the paper is as follows. The neural network is introduced in section 2. In the same section, the similarities to a rule-based system are discussed. In the third part of the paper, the practicability is demonstrated using an application in legal reasoning.

2. NEURAL NETWORK AND RULES

In the following the neural network and its mathematical description is given. For a more detailed description of neural networks see [HKP], for example. We consider networks that describe a mapping from an input space $x \in \mathbb{R}^n$ (the facts of the case in legal reasoning) to an output space $y \in \mathbb{R}$ (the final decision). In the widest sense, we consider a rule to be domain-specific knowledge about the same input/output mapping, which can be quantified in simple expressions of the form: if (premise) then
(conclusion), where the premise makes a statement about the input space and the conclusion makes a statement about the output space. In this section, we first describe the network architecture and then show how to initialize the network with domain specific rules.

Let \{Rule_i, i: 1...M\} be a set of rules. For every rule we introduce a basis function \(b_i(x)\) which is equal to one wherever the premise of the rule is valid, and equal to zero otherwise. Alternatively, we define a number (typically but not necessarily between 0 and 1) which we call the validity of the rule and which indicates the certainty that a particular rule can be applied, given the input \(x\). We assume that a basis function can be described by a multivariate Gaussian

\[
b_i(x) = k_i \exp \left[ -\frac{1}{2} (x - c_i)^\top \Sigma^{-1} (x - c_i) \right],
\]

where we assume that the covariance matrix \(\Sigma\) is diagonal and where \(\sigma_{ij}\) is the \(j\)th diagonal element in \(\Sigma\). In the following, we further assume that \(k_i = 1\). The parameter \(\lambda_j = c_j\) defines the position in the \(j\)th dimension of the input space where \(Rule_i\) has the largest validity. The parameter \(\text{range}_{ij} = \sigma_{ij}\) indicates approximately the range in which \(Rule_i\) is valid in the \(j\)th input dimension. We say that each hidden unit has its own receptive field in the input space, a region centered on \(c_j\). The width of this receptive field is proportional to \(\sigma_{ij}\). A basis function approximately corresponds to the membership function, known from fuzzy logic. We assign a parameter or weight \(w_i\) to every basis function which is equal to the (expected) value of \(y\), given that \(Rule_i\) is valid. We define the network output to be

\[
y_N(x) = \frac{\sum_i w_i \cdot b_i(x)}{\sum_i b_i(x)}. \tag{2}
\]

In regions where only one rule demonstrates significant validity, the output is equal to \(w_i\). In regions where more than one rule has a significant validity, the equation forms a weighted average of the outputs of those rules. Note that the resulting architecture described by equation 2 is identical to the neural network architectures used by Moody and Darken [MD89]. When training data arrive, the network architecture can be modified, and the network parameters (centers, widths, and weights) can be adapted using backpropagation. The topology of the network is shown in Figure 1.

Given the state of the input the network makes a prediction. If we can measure the actual state of output we can adjust the network such that the prediction improves. If we use the quadratic cost function
Figure 1. Net topology

\[ E_D = \frac{1}{2} \sum_k (y^k - y_N(x^k))^2 = \frac{1}{2} \sum (e^k)^2 \]  

with pattern number \( k \), network output \( y_N(x^k) \) and the desired output \( y^k \), the network can be adjusted by gradient descent, where

\[
\frac{\partial E_D}{\partial q_i} = -\sum_k e_k \frac{b_i(x^k)}{\sum_j b_j(x^k)}
\]

if \( q_i \) is parameter \( w_i \) and

\[
\frac{\partial E_D}{\partial q_i} = -\sum_k e_k \frac{w_i - y_N(x^k)}{\sum_j b_j(x^k)} \frac{\partial}{\partial q_i} b_i(x^k)
\]

if \( q_i \) is a parameter in \( b_i \) (center \( c_{ji} \), range \( \sigma_{ji} \)). These gradients could be inserted into the formula to adjust the parameters with a certain learning rate \( \eta \). The parameter change is given by:

\[
\Delta q_i = -\eta \cdot \frac{\partial E_D}{\partial q_i},
\]

First we have to include the prior rules by the following transformation. Let us consider as an example the case where the input space is 2-dimensional. The Rule

\[
\text{If } x_1 \text{ is } A^1_{11} (\text{range}_{11} = R^1_1) \text{ and } x_2 \text{ is } A^1_{12} (\text{range}_{12} = R^1_2) \text{ then } y \text{ is } B^1 \]
corresponds to a Gaussian centered at \( c_{11} = A^1_1, c_{12} = A^1_2 \) with \( \sigma_{11} = R^1_1 \) and \( \sigma_{12} = R^1_2 \) and attached weight \( w_1 = B^1 \). The Rule 2

\[
\text{if } x_1 \text{ is } A^2_1 \text{ (range}_2 = R^2_1) \text{ then } y \text{ is } B^2
\]

corresponds to a 1-dimensional Gaussian centered at \( c_{21} = A^2_1 \) with \( \sigma_{21} = R^2_1 \) and weight \( w_2 = B^2 \). Note that the basis function is independent of \( x_2 \).

Finally, the Rule 3

\[
\text{if } x_1 \text{ is } A^3_1 \text{ (range}_3 = R^3_1) \text{ or if } x_2 \text{ is } A^3_2 \text{ (range}_3 = R^3_2) \text{ then }
\]

\[ y \text{ is } B^3 \]

is first decomposed into the two rules Rule 3a

\[
\text{if } x_1 \text{ is } A^3_1 \text{ (range}_3 = R^3_1) \text{ then }
\]

\[ y \text{ is } B^3 \]

and Rule 3b

\[
\text{if } x_2 \text{ is } A^3_2 \text{ (range}_3 = R^3_2) \text{ then }
\]

\[ y \text{ is } B^3 \]

and then we can proceed as we did with Rule 2. In this way we can construct the network from the set of rules.

3. Application in Legal Reasoning

In legal reasoning, the judge follows rules defined by the written law, but also includes precedents in his decision process. Often the interpretation of the law varies to a large extent among judges and it is difficult to find a common ground.

In this paper, a problem was chosen where it is not possible at all to apply the written law directly and judges have to rely to a large extent on precedents. This legal methodology of analyzing prototype decisions is very similar to the neural network philosophy: The neural network learns to make a prediction on a new case given these prototypical cases.

In general, there are two difficulties: (1) representing the law in rule-like
sentences ("if... then...") which could be entered in a computer, and (2) describing the circumstances in parametric form. Both require the expert's experience. For a more general treatment of this question see [Phil89a]. In the problem described here, we had the advantage that data (precedents) were available in tabulated, parametric form, and we had a set of rules defined by a legal expert.

The described algorithm provides the possibility to translate legal decision making into a computational form containing the following steps: Select the most essential features for the problem as inputs for the network. Enter the prototypical cases into the network and learn by adopting the parameters. Afterwards the network will be able to produce a reasonable output for a new constellation that is not prototypical. Due to its receptive fields properties the network effectively compares a certain input to the most analogous cases.

The neural network provides also the possibility of pruning the inputs and rules. The network's architecture is variable. In principle, after training the rules can be extracted again from the network, but we have to make sure that the set of rules is as concise as possible, otherwise the value of the extracted rules is limited. We would like to find the smallest number of rules that can still describe the knowledge sufficiently. Also the network should be encouraged to find the smallest number of conjuncts, which in this case means that a basis function (rule) is only dependent on a minimal amount of inputs. Both constraints can be implemented through additional penalty terms to the quadratic cost function. For this technique see [Holl92]. For the legal example, this pruning procedure produced a quite optimal set of rules and conjuncts.

In this particular application we examine the identification of precedents in the area of the law of immaterial damages (see also [Phil89b]). As input, several circumstances (type, seriousness and duration of injury, seriousness and duration of consequences, sex, impairment of occupation, particular severity, medical malpractice) are used. As output serves the amount of compensation of immaterial damages in German Marks. Our data base consisted of 200 court decisions. These cases including the factual circumstances and the legal consequences are available in tabulated format from the German automobile club [ADAC]. Our system was required to learn to predict the magnitude of expected compensation of immaterial damages. Because of the relatively large and consistent data base, this problem is well-suited for solving with a neural approach.

We tried two different experiments. First, the network learns without prestructuring, and after the learning phase rules were extracted and
analyzed. In the second experiment, the improvement in generalization ability due to the prestructuring of the network was measured as a function of the number of rules (hidden units) which were used to prestructure the network. During network training, rule refinement takes place and after the training phase, it is possible to extract rules as explanation component for decision processes.

In the first experiment, we want to find rules extracted from the given data set, here the 200 court decisions. As described in section 2, we used a two-layer feed-forward network with gaussian radial basis units and with normalized network output (eq. (2)). The architecture used consisted of 9 input, 15 hidden, and 1 output unit. The parameters $A_i$ and $B_i$ were updated by a gradient descent learning rule. After 1000 cycles, the rules were extracted and presented as shown in figure 2.

The premises are connected with AND-operators and its values $A_i$ are normalized to the interval $[0;1]$. All ranges $r_{ij}$ are fixed to 0.25 and not updated. The conclusion $B_i$ is the real unnormalized value given in German Marks. Most rules could be interpreted as easily as the ones shown in figure 2: Rule 11 is an example of a rule which covers cases with low compen-

![Figure 2. Two typical rules extracted from the network after learning](image)

**RULE 11:**

<table>
<thead>
<tr>
<th>IF</th>
<th>THEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of injured body parts</td>
<td>0.45</td>
</tr>
<tr>
<td>highest severity of injury</td>
<td>0.13</td>
</tr>
<tr>
<td>highest duration of injury</td>
<td>0.14</td>
</tr>
<tr>
<td>diminishment of earning capacity</td>
<td>0.00</td>
</tr>
<tr>
<td>duration of diminishment</td>
<td>-0.33</td>
</tr>
<tr>
<td>sex</td>
<td>0.46</td>
</tr>
<tr>
<td>impairment of occupation</td>
<td>0.00</td>
</tr>
<tr>
<td>particular severity</td>
<td>0.16</td>
</tr>
<tr>
<td>medical malpracticenumber</td>
<td>0.00</td>
</tr>
<tr>
<td>amount of immaterial damages</td>
<td>5.31 DM</td>
</tr>
</tbody>
</table>

**RULE 14:**

<table>
<thead>
<tr>
<th>IF</th>
<th>THEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of injured body parts</td>
<td>0.40</td>
</tr>
<tr>
<td>highest severity of injury</td>
<td>1.00</td>
</tr>
<tr>
<td>highest duration of injury</td>
<td>1.01</td>
</tr>
<tr>
<td>diminishment of earning capacity</td>
<td>0.99</td>
</tr>
<tr>
<td>duration of diminishment</td>
<td>0.92</td>
</tr>
<tr>
<td>sex</td>
<td>0.47</td>
</tr>
<tr>
<td>impairment of occupation</td>
<td>0.00</td>
</tr>
<tr>
<td>particular severity</td>
<td>0.36</td>
</tr>
<tr>
<td>medical malpracticenumber</td>
<td>0.23</td>
</tr>
<tr>
<td>amount of immaterial damages</td>
<td>1007.86 DM</td>
</tr>
</tbody>
</table>
sation and Rule$_{14}$ is an example for a rule with high compensation. After rule extraction we could conclude, for example, the following: (1) not the number, but the severity of the injury is important for the decision finding process, and also (2) that the diminishment of the earning capacity is an important factor. But we could also conclude (3) that the decision is independent of sex (0 = female, 0.5 not mentioned in decision, 1 = male). For some rules the interpretation is more difficult, which can be partially attributed to the fact that often the court decisions are somewhat contradictory.

**Figure 3. Mean error of the network with and without prior knowledge in the generalization phase**

![Graph showing mean error of the network with and without prior knowledge.

Figure 3 compares the generalization ability of networks with and without prestructuring as a function of the number of hidden units (rules). The same network structure as in the previous experiment was used. 180 patterns were given in the training set and 20 patterns were used for testing the generalization ability. Prestructuring the network consisted of presetting the values of $A_i$ and $B_i$. It can be seen that prestructured networks learn faster and generalize better due to the additional knowledge used in network training.

4. **Summary**

In this paper an easily implementable way of combining uncertain rule-based domain-specific knowledge with the inductive learning capability of neural networks is presented. The advantage of the presented method is the ability to transform knowledge from one domain to the other. This can
supplement neural network knowledge, increase performance of the trained network and explain the decision making process. The practicability and consistence of the approach was demonstrated in the presented legal application.

5. ACKNOWLEDGEMENT

I would like to thank Volker Tresp for helpful ideas and his support in preparing this paper. I also wish to thank the members of the neural network group at Siemens Corporate Research in Munich for stimulating discussions.

REFERENCES


