A Connectionist Model to Justify the Reasoning of the Judge

Filipe Borges Raoul Borges Danièle Bourcier
Laboratoire Informatique Droit Linguistique
Centre de Théorie du droit CNRS
Université de Paris 10
{*fa_borges,paercebal*}*@hotmail.com, bourcier@msh-paris.fr*

Abstract. One of the main obstacles to the use of Artificial Neural Network *(ANN)* in the legal domain comes from their inability to justify their reasoning. Justification indeed is crucial for the judge because it assures him that the reasoning carried out by *a legal machine* is legally founded. We propose in this paper a method able to overcome this constraint by developing an *algorithm of justification* applied to connectionist prototypes (Multilayer Perceptron) implemented at the Court of Appeal of Versailles. We will first describe the algorithm. We will then discuss the two main advantages offered by the ANN with regard to rule based systems. A first advantage consists of their suitability for some types of reasoning not based on explicit rules, which are specially numerous in the discretionary field of the judge. Another advantage can be emphasised as a result of our experiment: these models can be used for improving the self justification process of a decision maker (making it more precise) and even for predicting (or suggesting) new lines of reasoning based on implicit knowledge. Some examples extracted from a knowledge base on the contract of employment (clause of non-competition) will illustrate this point.

1 Artificial Neural Networks and Law (ANN): State of Works

The models of artificial intelligence generally have as their objective to mimic human reasoning. But they need first to know and to formalize the rules which are the base of the decision-making process, and ways of to using them. When it is about the law, we generally think that these rules are *given* by the legislator. We can't, however, give to the whole law the shape of rules[1]. If we try, we notice quickly that the basic elements of the law (its ontology) can be vague or indefinite and that in reality, law delegates to the judge the task of adding the implicit rules of interpretation of these elements to allow him to *legally* resolve real cases; it is what is called the discretionary power of the judge.

An ANN is particularly adapted to such vague, fuzzy and noisy knowledge. Numerous tests were made to use connectionist models in the treatment of the law ([4, 2]). Their aptness to handle the most open textured parts of the reasoning was recognized ([1]) in spite of the criticisms which are made of them ([5]). Their main defect lies in their difficulty to express the rules which underlie their results.

In certain domains (treatment of the signal, treatment of the speech, recognition of faces), the experimenter does not need to know which pixel or which neuron tipped over the decision of the network. On the other hand to know which criterion or sub-criterion took decision is indispensable for the magistrate and for the affected citizens.

[1] For example, principles require a different representation.

Filipe Borges, Raoul Borges and Danièle Bourcier, 'A Connectionist Model to Justify the Reasoning of the Judge' in T.J.M. Bench-Capon, A. Daskalopulu and R.G.F. Winkels (eds.), *Legal Knowledge and Information Systems. Jurix 2002: The Fifteenth Annual Conference*. Amsterdam: IOS Press, 2002, pp. 113-122.

Our hypothesis consists in showing that an algorithm can justify the "decision" of a neural network. The first stage is still to teach a network the knowledge of the judges, who sometimes are unaware certain important elements they use. The network can then be used by the decision-maker to auto-justify or to criticize its own criteria.

2 Theory of a Connectionist Justification

2.1 Obstacle: The Dispersal of the Information in the Structure

The reading of a ruling shows that the judge justifies himself generally *by quoting the elements of facts or law which directed his decision*. This clause is not applicable because the contract of employment foresaw the payment of a financial counterpart and because this one was not paid ; that dismissal is without real and serious cause because the letter of dismissal does not express real and precise motives; this privilege of jurisdiction is pushed aside because an agreement foresees rules of particular competence, and so on. This way, a satisfactory justification emanating from a connectionist model would consist in stating which elements of facts or law (criteria) were the most important in the decision-making process.

At present, this model can at the most (at least for the perceptron) list the inputs and show that their combination had a particular effect on the output. It is not a justification but simply a description.

What is easy for an expert system (to justify the reasoning) is hard for an ANN, because of the way the knowledge is structured. The problem is that the ANN scatters the information through its structure and the weights of its connections. This makes the procedure of recovery of relevant information particularly complex.

2.2 Hypothesis: to Track Down the Signal in the Structure

The action of the perceptron consists in converting the value of its inputs to a signal which propagates along its connections. This signal undergoes several mathematical transformations and is output by the last neuron as a value which is interpreted by the experimenter.

In fact, the main difficulty consists in understanding the way the global signal *propagates* and to discover retroactively how every neuron acted on this signal. In this way, we hope to go back to the inputs of the network and to determine the influence of each of them.

To ask what role is played by every entry on the output signal of a perceptron is like asking what role is played by every tributary in the final flow of a river. Different sources (the inputs) feed various tributaries (the connections of the network) which eventually end up on the same rivers (connections of the hidden neurons), and throw themselves into the sea by the way of the same river mouth (the output of the network). If we can measure the flow of water at the mouth as well as the flow of the various rivers we can calculate the contribution of every river to the final flow, and interpret it. A weak flow can be explained by the fact that such rivers have an extremely weak flow themselves. On the contrary, one or several rivers with an enormous flow can explain why we observe a very important flow of water at the mouth.

The connectionist model used here presents a similarity with this image, at least for its connections. But it presents a supplementary complication : the existence of neurons. These neurons can convert the signal so that if we take the previous image we can obtain tributaries which, besides crossing each others, see their flow increasing and decreasing several times during the same journey, and, moreover, it happens regularly that their course is reversed. All

these complications justify the use of data processing 'to track down' and to measure all the transformations of the signal as it propagates within the neural network.

The hypothesis is that in the case of a binary output (ranging from 0 to 1 and interpreted), certain neurons have, by the way of their connections, a tendency to decrease the signal while the others tend to increase it. By adding the influence of every connection relative to every input and by comparing them with the others we can determine the sense of influence of every input (the global sense of the source) and the relative importance of their influence on the final result.

We also put forward the hypothesis that the most influential criteria within a neural network correspond to the most determining elements within a decision-making process, and identifying them is enough to justify the reasoning.

3 Protocol of Experiment

3.1 Development of an Algorithm

The method developed is valid for a multilayer perceptron with continuous inputs and outputs (ranging from 0 to 1), with one hidden layer, and one final neuron. The neural network used in this experiment has 14 entries, 1 hidden layer, 3 hidden neurons, and 1 neuron on the output layer. The values of the entries and of the output are continuous (from 0 to 1) but were used in a boolean way (0 or 1).

– We find the weights 'W_{nm}' of every connection

– We calculate the signal 'S' sent by every entry to every hidden neuron, from the input value 'X' (which can be 0 or 1) and the weight which connects it to the various hidden neurons :

$$S_{nm} = X_n.W_{nm}$$

Note : n corresponds to the index number of the neuron which is examined, and m corresponds to the index number of the neuron which is on the next layer. The index numbers go from 1 to 14 on the first layer, from 1 to 3 on the hidden layer and is unique one the output layer.

– The range of the inputs (0 to 1) and the threshold neurons complicates the problem (it would be simpler if the range was –1 to 1). In this context a value of X=0.5 corresponds to an unknown value. A value of X=1 has a 'normal' effect on the neural network, and a value of X=0 has an inverted effect. Then, we calculate the influence 'I' of each entry :

$$I_{nm} = (2X_n1)* W_{nm}$$

Note : The calculation of the influence will be made for each neuron of each layer of the neural network. Only the output neuron has a natural influence of 1. We consider that the influence and corrected influence of the final neurons are neutral and absolute. They do not modify the values of the previous influences.

– We multiply this influence 'I_{nm}' by the corrected influence 'IC_m' (which is exercised by the corresponding hidden neuron on the final neuron) to obtain the corrected influence 'IC_{nm}' (which is a more precise measure of the influence exercised by this entry by the way of this hidden neuron) :

$$IC_{nm} = |I_{nm} * IC_m|$$

Note : The absolute value is used to separate the calculation of the values and of the sense of influence. This problem is (again) linked to the chosen range of the neurons value (0 to 1).

– We determine the local sense 'SL' of the action of the entry, which is the sense of its influence on the signal going from the entry to the hidden neuron. If the signal 'S_{nm}' is negative, the local neuron acts as an inhibitor to the neuron of the next layer. Then, the local sense SL_{nm} will be equal to –1, should the opposite occur, it will be equal to +1. If the signal is equal to 0, and due to the action of the threshold neurons, the lack of signal has an effect on the output signal. The sense of the action is the opposite of the weight W which links the two neurons (neuron analyzed and neuron of the next layer). In a case of a signal equal to 0 : If the weight is negative, the local sense will be equal to –1, should the opposite occur, it will be equal to +1.

– The global sense 'SG_{nm}' is the sense of influence a neuron exercises on the output value through one path. In our model, the global sense of each hidden neuron is equal to its local sense (because there is only one connection which links the hidden neurons to the output neuron, and this is a direct link). As the model contains 3 hidden neurons, the influence of each entry on the output neuron can be exercised through 3 different paths, and its value depends on the sense of influence of the hidden neurons. So, each input has three global sense, one per possible path. We obtain the global sense of each neuron through each path by multiplying the local sense of the neuron by the global sense of the corresponding neuron on the next layer :

$$SG_{nm} = SL_{nm} * SG_m$$

Note : The global sense of an input through a hidden neuron depends on the action it is exercising on this hidden neuron. If it strengthens the influence of the hidden neuron it adopts its sense of influence. If it weakens it, the entry adopts the opposite influence. For example, if a hidden neuron tends to lower the output value this hidden neuron has a negative influence. And if an entry weakens the influence of this hidden neuron, it has a positive influence on the output value through this hidden neuron.

– We multiply the corrected influence 'IC_{nm}' with the global sense to obtain the sense and the intensity of the neuron through the analyzed path :

$$IC_{nm} = IC_{nm} * SL_{nm}$$

– We determine in the same way the corrected influence of the input under consideration through every hidden neuron. Then we add each corrected influence to obtain the global influence 'IG' of that input :

$$IG_n = IC_{n1} + IC_{n2} + \ldots + IC_{nm}$$

The global influence of the input under consideration corresponds to the influence this neuron exercises on the output value.

Finally, the inputs with a negative global influence are those that pushed the output value towards 0, and the entries with a positive global influence are those that pushed the output value towards 1. The criteria from which the value of the global influence is furthest from '0' are those that had most impact on the decision.

3.2 Test of the Algorithm

The algorithm was implemented on a neural network modeling a dispute using 14 criteria. The conclusion of the dispute is binary (applicability or not applicability of a clause of non-competition in a contract of employment). A test on 30 different situations selected at random and compared with justifications proposed by a magistrate allowed us to give an outline of the performance of the algorithm[2].
The criteria are :

- The clause is foreseen by contract
- The clause is foreseen by a collective agreement
- The employee has been informed of the existence of the clause
- The employee had access to strategic information
- The duration of the clause is excessive
- The area of the clause is excessive
- The list of forbidden companies is excessive
- The list of the forbidden activities is excessive
- A financial counterpart is foreseen by contract
- A financial counterpart is foreseen by a collective agreement
- The financial counterpart has been paid
- The employee demands the cancellation of the clause
- The cancellation of the clause is foreseen by contract or by collective agreement
- The clause is cancelled by the employer

4 conditions must be reunited for the clause to be applicable :

1. The employee must be informed of its existence
    - By contract
    - By a collective agreement, which must be given to the employee on or before the first day of work
2. The clause must protect the interests of the employer without being excessive.
3. The clause must not be cancelled by the employee
    - Although one can cancel it if a financial counterpart, foreseen by contract or collective agreement, has not been paid.
4. The clause must not be cancelled by the employer
    - Although one can cancel it if such procedure is foreseen by contract or collective agreement.

If one of these 4 conditions is not confirmed, the clause is inapplicable.

[2]The algorithm of justification proposes percentage of influence for every criterion, but only the more important ones have been retained. The significant influences will be identified and commented upon in part 4: Results and observations.

4 Results and Observations

Results are given in the form of a list summarizing the set of criteria and allocating to each a percentage corresponding to the relative influence which it had on the decision

In every exercise the algorithm showed that it succeeded in recognizing without error the criteria which had a positive influence on the dispute (applicable clause) and those that had a negative influence (inapplicable clause).

For example :

EXERCISES[3]	**Results**	**Justification of the judge**	**Justification of the algorithm**[4]
- Clause is foreseen by contract - Clause is excessive (duration) - Employee is strategic - Counterpart paid - Cancellation foreseen - Cancellation demanded by employer	Clause inapplicable	- Clause is excessive (duration) - Cancellation foreseen - Cancellation demanded by employer	- Clause is excessive (duration) (59.9%) - Cancellation foreseen (19.2%) - Cancellation demanded by employer (18.9%)

In this way, this method of justification can be used to select an initial set of relevant criteria which can to be used as a basis of the justification.

The algorithm of justification also allows us to draw up a list of relevant criteria organized into a hierarchy according to their impact on decision-making. However, this hierarchy does not systematically correspond to the hierarchical list of the justifications given by the judge.

4.1 The Revealing of Finer Cognitive Processes

Some exercises have shown a gap between the justifications of the judge and those proposed by the justification algorithm.

For example:

EXERCISES (extract)	**Results**	**Justification of the judge**	**Justification of the algorithm**
- Clause is foreseen by contract - Employee is strategic	Clause inapplicable	- Clause is foreseen by contract - Employee is strategic	- Employee is strategic (15.7%) - Clause is excessive (enterprises) (14%)

We can see that in this example the justification proposed by the judge is not precise, and the justification proposed by the algorithm is, in fact, correct. In reality, the conclusion of the case being positive, the judge has kept the positive terms in his justification as the most important factor.

In this example, the first criterion proposed by the judge (clause foreseen in contract) is less influential than the second criterion proposed by algorithm (excessive clause), since if this clause is not foreseen by contract it can always be foreseen by a collective agreement (if the employee is informed in time of the existence of this clause in the agreement). This criterion can therefore be replaced by two other criteria. Whereas, if the clause presents an excessive character, no other combination will be able to counterbalance its influence. In this example, the clause is applicable essentially because the employee is strategic and because it is not excessive.

Certain cognitive processes occurring in our own process of justification allow us to select a certain number of criteria among those which can play a part in our decision making. These

[3] Only the more influential criteria are proposed in this column. All the others have a negative value.

[4] Also, only the most influential criteria are given in this column. They are extracted from the hierarchical list proposed by the algorithm of justification.

criteria are sometimes selected because they are the most important, but sometimes because they are the most suited with regard to the demand for justification.

In this category of examples, the indistinctness of our cognitive process of justification is at the origin of the errors. In fact, the algorithm works well enough to question the validity of our own rules of justification.

4.2 Surprises of Learning

It sometimes happens that the algorithm of justification does not attribute to some criteria the influence they deserve.

For example:

EXERCISES	**Results**	**Justification of the judge**	**Justification of the algorithm**
- Clause is foreseen by contract - Employee is strategic - Counterpart paid - Cancellation foreseen - Cancellation demanded by employer	Clause inapplicable	- Cancellation foreseen - Cancellation demanded by employer	- Employee not informed (36.2%) - Clause not foreseen by contract (38.6%) - Cancellation foreseen (12.9%) - Cancellation demanded by employer (12.3%)

After checks against the reference point of the judge certain criteria seem to have their influence overvalued and some others their influence understated.

More disturbing still, in the modeled dispute, the model regularly grants a disproportionate importance to one particular criterion. In certain combinations, the influence almost suggested that it was the fundamental criteria which tipped over the decision, while in the state of the law at the time of the experiment this criterion always needed the combination of at least two other criteria to play a determining role :

EXERCISES	**Results**	**Justification of the judge**	**Justification of the algorithm**
- Clause is foreseen by contract - Employee is strategic - Clause is excessive (duration)	Clause inapplicable	- Clause excessive (duration)	- Counterpart not paid (47%) - Clause is excessive (duration) (37.8 %)

A check of the algorithm of justification allowed us to conclude that this problem of hierarchical organization was due to the mode of learning of the perceptron.

When a series of examples is taught to a neural network in the phase of learning, the network develops a capacity of generalization to new examples which had not been presented to it. It is expected to know how to resolve correctly new cases if these new examples present a sufficient resemblance to the examples on which it was trained. In fact, the phase of learning consists in finding a logic, a ‘route’ which satisfies all the examples which are presented to it. The learning of this logic allows the model to be used in new cases implying new combinations of the same criteria.

But what happens when several routes can end in the same result ? For example : IF A is TRUE OR B is TRUE THEN C is TRUE ; if C is effectively confirmed, we can explain it by 3 different ‘routes’ :

- A was true
- B was true
- A and B were true

If this case is presented to a neural network, it will select one of the 3 solutions and so 2 chances in 3 to be different from the one that the experimenter had expected.

The perceptron will take the route that seems to be the easiest and the most coherent with its learning base ; even if it is not the way that was chosen by the judge.

So, the defects of hierarchical organization criteria can be explained by the fact that neural networks can use ‘shortcuts’, allowing it obtain good results on all the examples (16000 combinations have been tested in this case). It may be that the model has managed to use its property of generalization to escape the logic we wanted to impose to it. By using other criteria in a fortuitous way, it may have discovered another logic which also gives the results wanted by the experimenter.

But, its learning is not as unpredictable that this example leads us to suppose. Numerous examples can control it and allow the model to understand the real effects of every criterion on each of the others. So, the precision of the justification algorithm is directly bound to the precision of the learning base.

It is also possible that the reduction of the number of the hidden neurons had forced the neural network to complicate its reasoning, or to build ’logics’ other than those used by the judge. So, increasing the number of hidden neurons can possibly allow to obtain more precise justifications.

4.3 A Predictive Neural Network?

After this observation different phases of learning have been tested to try to supervise the reasoning of the neural network more effectively. The learning base has been increased (from 135 examples to 1024), and more neurons were affected to the hidden layer.

The results we got from justification algorithm increased the correctness of the justifications for all the examples except those implying the criterion of disproportionate importance (‘Counterpart not paid’, see above).

We can always explain this observation by a defect on the algorithm of justification. However, this observation must be compared with a turnover of jurisprudence, which has occurred some months after the modeling of this dispute. This reversal made this criterion a fundamental point of this dispute.

We can not deduce that neural network ‘wisely’ anticipated the supreme court decision. But we can suppose that the importance of the criterion was present in the learning base, in some ‘scattered’ way. And it has affected the way the model has learnt the dispute. The scattered information in the learning base affects the structure of the neural network. But, a justification algorithm is necessary to reveal this information.

If the increasing importance of one criterion is spread out on 300 case of jurisprudence, a person will not be able to discover this importance unaided. But, a neural network will not be able to ignore it. This observation can not allow us to draw conclusions about the reasons why the influence of a criterion has been overestimated. Nevertheless, we will have to study the possibility of using neural networks as predictive models of case law evolution. Let us recall that the predictive function of dynamic systems has been detected in an earlier work ([3]): the case studied there involved the anticipation of a jurisprudence turnover (subject: pledge) by analyzing the vocabulary used during 20 years of jurisprudence in that domain.

5 Perspectives

From the point of view of legal research, a method allowing the quantification of the influence of every element of information in the decision making of a decision maker offers ’feedback’ to better understand the mode of application of legal rules, to analyze decision making, and to

indicate the relative importance of the criteria which were used as basis of the reasoning. In this way, we have an effective illustration of the interest of empirical searches in jurimetrics.

Nevertheless, we have to ask the question of the correctness of obtained measures. The problem is that we do not have procedure to verify this correctness. The justification experiment based on this dispute (applicability of the clauses of non-competition in a contract of employment) allows us to verify the equivalence of this measure by comparing them with the justifications given by the judge. But when it indicates that such a criterion has a value of 135.4 and a global influence of 23.73 % on definitive decision, do we have a way of verifying that this influence is not 23.5 % or 24.2 % ? And even before this problem arises, can we always study the reasoning of the judge from the realized model ?

The scenario proposed first, explaining that the ANN selected one 'route' from others which reach the same solutions as the human decision making modeled, is at the source of this difficulty. Inputs would be the same, as would the output value be, but the reasoning could be completely different. In which case, the justification proposed by the ANN would not be applicable in return to original decision-making.

Effectively, nothing can indicate that the ANN used the same reasoning. However, the surprising agreement of the justification with the idea that one forms at the outset as to the relative importance of criteria (whether it is during a vague process or during a completely mastered process), as well as the stability of the justification in spite of the fact that the ANN was trained several times with the same learning base, lets us think that this method remains relevant to study real decision-making.

Now, that the empirical results have shown that this method can be interesting, the next step will be to mathematically prove the correct of the algorithm, by using a variant of the retropropagation algorithm, on a neural network with entries going from -1 to 1. In this model it can be possible to consider precisely the action of the threshold neurons. Empirical tests have been made with continuous inputs (continuous values from 0 to 1). But, even if they were consistent with the judge's point of view, only a mathematical validation of the algorithm, and the multiplication of the experimentations can allow us to judge the correct of the results.

6 Conclusion

This paper confirms that ANN can be an adequate model to handle vague legal knowledge and that they can complete systems with rule bases to model the exercise of discretionary power of the judge. Also, we were able to push aside one of the major obstacles in the application of connectionist networks in law, namely their opaqueness, by presenting a justification algorithm. Finally, another virtue could be attributed to the "connectionist judge": it would illuminate tendencies in the evolution of the law, which could lead to a turnover or even to suggest new standards to the legislator.

References

[1] Bench-Capon Trevor, 1993. Neural Networks and open texture. In *Proceedings of the Fourth International Conference on Artificial intelligence and Law.* Oxford, June 15-18, 1993. New York : ACM Press 1993, pp. 292-297.

[2] Bochereau, Laurent, Bourcier Danièle, and Bourgine Paul, 1991. Extracting legal knowledge by means of a multilayer neural networks : application to municipal jurisprudence. In *Proceedings of the Third International Conference on Artificial intelligence and Law.* Oxford, June 25-28, 1991. New York : ACM Press 1991, pp.288-296.

[3] Bourcier Danièle, Clergue, Gérard, 1999. From a rule-based conception to dynamic patterns. Analysing the self organisation of legal systems. In *Artificial Intelligence and Law* 7 :211-225, 1999.

[4] Fernhout F., 1989. Using parallel distributed processing model as part of the legal expert system. . In *Proceedings of the Third International Conference Logica, Informatica, Diritto*, Firenze, 1989.

[5] Warner, David R. 1993. A neural network-based law machine : the problem of legitimacy, *Law, Computers and Artificial Intelligence* 2, 135.