International Journal of Performance Analysis in Sport

ISSN: 2474-8668 (Print) 1474-8185 (Online) Journal homepage: https://www.tandfonline.com/loi/rpan20

Artificial Neural Networks in Sports: New Concepts and Approaches

Jürgen Perl

Artificial Neural Networks in Sports: New Concepts and Approaches

Jürgen Perl,
University of Mainz, Institute for Computer Science, Germany

Artificial neural networks are tools, which – similar to natural neural networks – can learn to recognize and classify patterns, and so can help to optimise context depending acting. These abilities, which are very useful in a lot of technical approaches, seem to be as well useful in particular in analysing and planning tactical patterns in sport games or patterns of learning behaviour in training processes.

In a first attempt, in co-operation with LAMES from the University of Rostock, tactical structures in volleyball could successfully be analysed using neural networks.

However, the problem is that the special type of network that has to be used for such analyses (i.e. the so called Kohonen Feature Map or KFM) needs a huge amount of data and lacks the necessary dynamic in continuous learning.

So in order to describe, analyse, and evaluate continuous learning processes in sports a dynamically controlled network ("DYCON") has been developed, which consists of a conventional KFM combined with a time-independent neurone-driven control: Each neurone is imbedded in a dynamic performance potential control system, which had been developed for analysis and control of physiological adaptation processes in sport.

Two main advantages of DYCON are: Its learning efficiency is very high. In practice, it needs only some hundred data to coin a pattern, where a conventional KFM normally needs about 10.000 to 20.000. Moreover, it can learn continuously and so can recognise and analyse time depending pattern changes.

So, DYCON can support the study of processes in sport games in an easier and more efficient way. Moreover, it can help to analyse tactical changes of a team during a season or even during a tournament, as has been done with squash in co-operation with MCGARRY, University of Fredericton. Finally, in a co-operation with RAAB, University of Heidelberg, we try to find out if and how DYCON can be used for analysis and optimisation of training processes in sport.

1 Introduction

Neural Networks are one of the most important paradigms for the modelling of learning in the field of computer science. Various models have been developed under different aspects of learning and referring to different types of networks.

In the following a Neural Network approach is presented which allows to investigate unsupervised learning processes with simultaneous maintenance of the learning abilities. This approach belongs to the class of Self Organising Maps (SOM) the most famous representative of which is the Kohonen Feature Map (KFM), e.g. see Kohonen (1995), Wilke (1997), and Haykin (1999). In order to describe, analyse, and evaluate continuous learning processes, the idea was to design a type of **Dy**namically **Co**ntrolled **N**etwork (DyCoN), where the internal dynamic does not depend on uniformly time-dependent parameter functions (i.e. the learning rate and the radius of activation, see section 3) but on dynamic systems, which in turn are controlled by the changing states (e.g. learning progress) of the network.

The particular DyCoN-type presented in the following is dynamically controlled using the **Per**formance **Pot**ential-control-system (PerPot), which originally had been developed for analysis and control of physiological adaptation processes (see Mester et al. (2000), Mester and Perl (1999), (2000), and Perl and Mester (2001)): In this approach, each neurone controls its individual learning rate by its learning progress while the neurone's individual radius of activation is controlled by its performance potential.

Due to the properties of PerPot, such a "PerPot-driven" DyCoN is able to learn continuously and concurrently, to forget and collapse, and so can be used to model basic physiological learning behaviour. Moreover, its learning efficiency is very high. In practice, it needs only some hundred data to coin a pattern, where a conventional KFM normally needs about 10.000 to 20.000. So, DyCoN can support the study of processes in sport games in an easier and more efficient way. Moreover, it can help to analyse tactical changes of a team during a season or even during a tournament, as has been done with squash in co-operation with McGarry, University of Fredericton. Finally, in a co-operation with Raab, University of Heidelberg, we try to find out if and how DyCoN can be used for analysis and optimisation of training processes in sport.

2 The Performance Potential-Metamodel

In order to analyse and optimise physiological adaptation processes – as e.g. are training processes in sport, adaptation processes in physiology, or healing processes in medical therapy – a metamodel has been developed, which helps to simulate the interaction between load and performance (see Perl (1999), Mester and Perl (2000), Perl and Mester (2001)).

2.1 Basic PerPot

The basic concept of the performance potential-metamodel, which has been used in PERPOT, is that of antagonism: Each load impulse feeds a strain potential as well as a response potential. These buffer potentials in turn influence the performance potential, where the response potential raises the performance potential (delayed by DR) and the strain potential reduces the performance potential (delayed by DS), of course depending on the maximum capacities and on the current states of the respectively involved potentials (see Fig. 1; the term "load rate" later on in DYCON is identified by "training intensity").

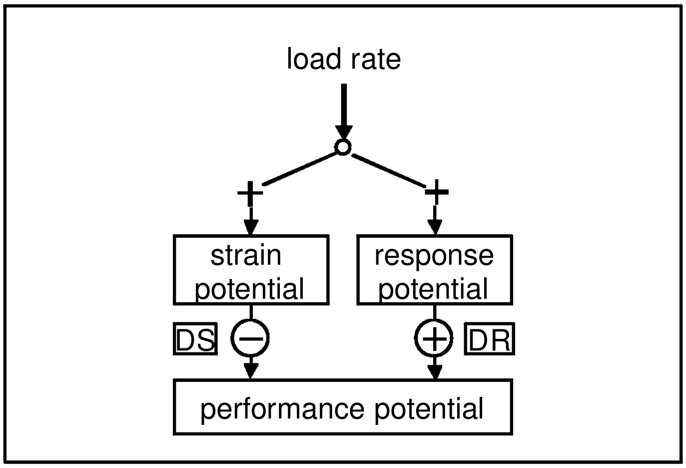

Fig. 1. Antagonistic structure of the basic performance potential-metamodel

If the delays are identical the rates compensate each other, resulting in a constant performance potential. Otherwise, the relation between the delays specify two types of system behaviour: super-compensation and balancing out (see Fig. 2). These effects are well-known from physiological adaptation processes.

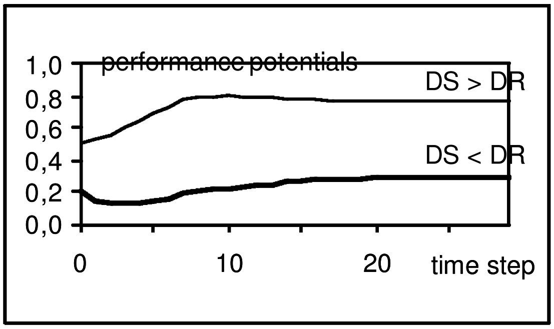

Fig. 2. Performance profiles depending on the relation between delays

2.2 Strain overflow

A further interesting observable effect not yet represented is that of collapsing: If the load integral over a period of time becomes to high, the performance breaks down spontaneously. This effect can be modelled using the concept of overflow: Potential capacities are limited. So, if in particular the strain potential is fed over its upper limit, an overflow is produced, which reduces the performance potential (with a delay DSO, see Fig. 3).

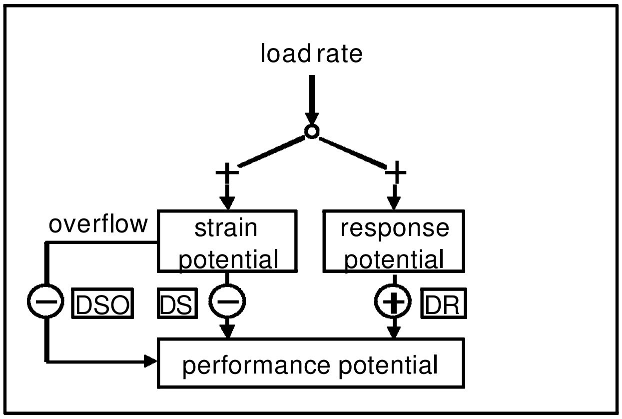

Fig. 3. PERPOT-metamodel containing strain overflow

The collapsing effect of load overflow meets specific expectations of physiological practice (see Fig. 4). In particular, the concept of potential overflow allows to introduce a reserve function as the difference between the strain potential capacity and the current strain level: Overflow starts when the current strain level becomes greater than the strain potential capacity. So the reserve R = maxSP - SP indicates how close the system is to collapsing – i.e. how "dangerous" next load steps are.

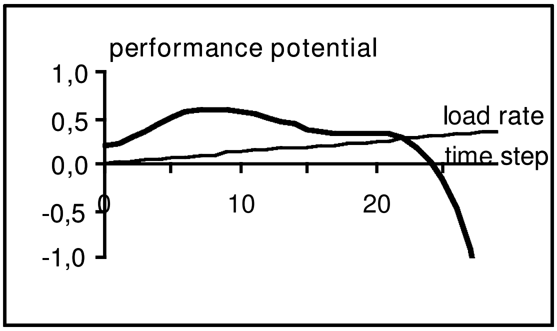

Fig. 4. Performance collapse caused by strain overflow

2.3 Atrophy

Finally, reducing performance by means of atrophy has been modelled, the typical effect of which is shown in Fig. 5. Atrophy plays an important role in general in physiological adaptation processes and in particular in learning processes. It can help to model and understand phenomena of forgetting, modifying, and superposing information, and so can be useful for planning training strategies in the context of continuous learning of changing and concurrent patterns. So atrophy as well has to be an important aspect in the behaviour of a dynamically controlled network.

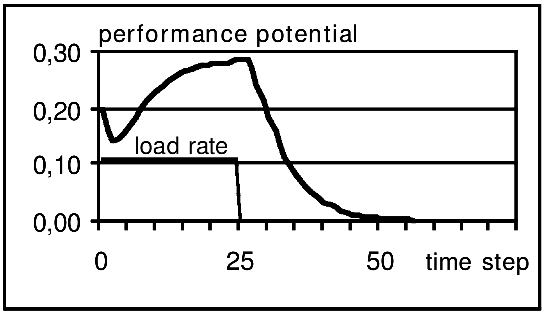

Fig. 5. Atrophy of performance after reducing the load rate to "0" (atrophy delay DA=20)

3 PERPOT-driven DYCON

3.1 Kohonen algorithm

Both Kohonen Feature Map and DYCON consider a grid of neurones N containing weight vectors that can be understood to be situated in the stimulus or input space. Each input **I** is mapped to the neurone with weight nearest to **I**. Learning is accomplished by updating the weight vectors using a learning rate LR, which describes the "amount of learning", and a radius of activation RA, which means the "region of influence".

According to the Kohonen algorithm, these internal control functions LR and RA are decreased explicitly over time: In the initial phase LR and RA have large values in order to achieve a coarse ordering of the neurone positions according to the input space respecting the network topology. After the initial phase, continuously decreasing LR- and RA-values shall effect a more and more accurate adaptation to the input distribution.

The idea of the DYCON approach in general is that of having LR and RA dynamically controlled by the developing situation of the network instead of using explicitly defined functions. In particular, in the case of PERPOT–driven DYCON the algorithms for the computation of LR and RA are of the following types.

3.2 Radius of activation and learning rate

The basic idea is to combine each neurone N of the network with a specific PERPOT-metamodel, which helps to self-control its radius of activation RA[N]. This way, RA[N] is a function of the current performance potential PP[N] and hence implicitly depends also on the values of training intensity TI (i.e. "load rate" in Figs. 1 and 3).

The idea of making RA[N] depending on PP[N] is that the performance potential this way characterizes the grade of learned information, which is encoded in its weight vector. So in particular increasing PP[N] indicates a progress in learning in the meaning of "stabilising information" and so in turn reduces the need for global adaptation – i.e. increasing PP[N] reduces RA[N]. However: Too much TI can cause the complementary effect by strain potential overflow. In the contrary, too little TI can cause atrophy. So learning can be controlled and optimised by strategies, using optimal learning step depending TI-values (also see examples in section 4). In particular, atrophy and its associated effects on RA can be used for modelling processes of continued and concurrent learning: By adjusting the delay of atrophy, the characteristic performance profile of the network – e.g. its ability of saving information – can be modelled, which then in turn determines the ability of accepting new information by means of increasing the radius of activation.

The self-control of the learning rate LR is managed by strongly coupling it to the current learning situation – i.e. by means of the distance d[N] between the weight vector of the neurone N and the current learning impulse **I**. The idea behind this attempt is that increasing d[N] indicates a progress in learning in the meaning of "unfolding" and so in turn reduces the need for further adaptation. This way, the functions LR and RA depend dynamically on the particular pattern to be learned, the training intensity TI, and the progresses in learning and stability.

As is pointed out more detailed in the following sections, the main advantages of the DYCON concept are:

- A once trained network can be used as a platform for further learning steps, only needing very few additional information for adjusting.
- A DYCON can follow and map learning processes and so can be used for modelling and analysing physiological learning phenomena.
- It can be used to model antagonistic and opposing learning phenomena and so can help to figure out and to optimise training strategies.

4 Basic properties

The following examples are used to demonstrate the typical behaviour of neurone-oriented PERPOT-driven learning. With regard to applications in physiology, sport, and medicine the n-dimensional weight- and input-vectors in general have the meaning of processes of length n, representing dynamic structures in motions, behaviour, or strategies (note that the meaning of n in the following is different from that in sections 2 and 3). Further on, according to the terms of motion *patterns*, behavioural *patterns*, or strategic *patterns* as objects that have to be learned in practice, in the following the areas of stimuli or inputs that have to be learned by the network are also called patterns.

In the particular case of 2 dimensions, however, which is dealt with as a more suggestive example in the following, input- and weight-vectors are considered as pairs of coordinates, without any interpretation as processes. So patterns can be chosen in a quite suggestive way as arbitrary 2-dimensional geometric Figs., and learning can be represented by the neurones' moving toward their respective positions.

The patterns used in the following are squares (Figs. 6 and 7), and triangles (Fig. 7). In the graphical presentations each neurone is connected with its topological neighbours (in the Manhattan topology used here these are the left, right, upper, and lower ones) – provided there Euclidian distance does not exceed a given level.

Comparing PERPOT-driven learning using DYCON with time-driven learning using conventional KFM shows two significant differences: KFM can be a bit faster than DYCON in learning an initial while DYCON often is more flexible and successful in solving difficult situations, as is shown in Fig. 6.

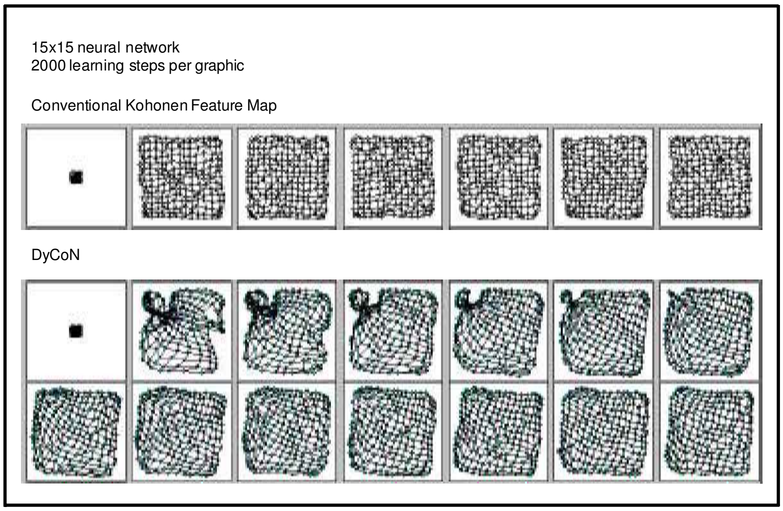

Fig. 6. Conventional Kohonen Feature Map vs. DYCON

Starting with an already learned initial pattern, however, DYCON learns new and superposing patterns very fast without forgetting the initial one, while KFM cannot continue learning, if the training is once completed (see Fig. 7). It always has to be reset and manually started whenever a new pattern is presented.

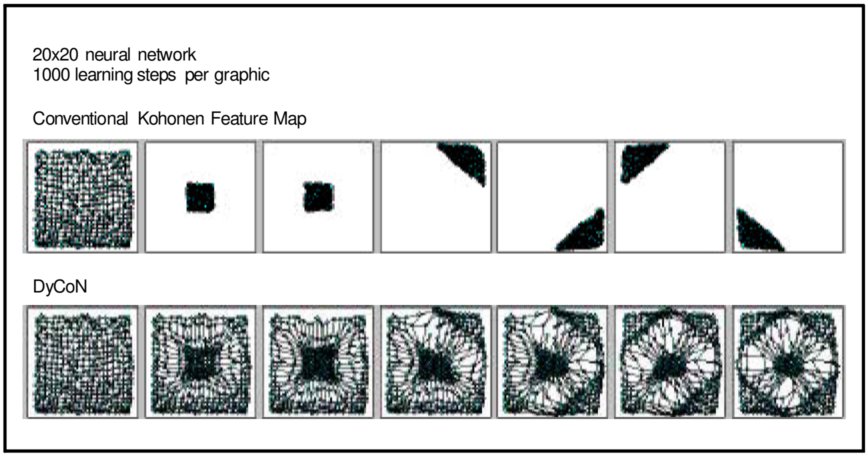

Fig. 7. Continuous learning with DYCON

4.1 n-dimensional test data and patterns

As is pointed out above, input-vectors in general mean processes, which are represented by different types of time series, the items of which are for example

- in medicine: EEG-values or lung volumes,
- in physiology: adaptation indicators like hormones or haemoglobin,
- in motion learning: motion coordinates,
- in sport games: positions on the play ground.

In any case a time series is characteristic not only for the respective individual but also for its particular state, learning situation, or playing strategy. So the aim is to recognize the specific and the suspicious aspects of that patterns in order to prepare preventive steps, develop learning strategies, or analyse weak and strong situations.

In order to do this, a time series is cut continuously into pieces of length n, which are called the processes of length or dimension n (e.g. see Sauer (1993)):

(4,4,3,2,1,4,…) <4,4,3,2> ; <4,3,2,1> ; <3,2,1,4> ; <2,1,4,…>; …

A set of those n-dimensional processes then is called a pattern of the regarding behaviour (e.g. see section 5). A pattern can characterize an individual behaviour as well as a general structure – if e.g. the used data are collected over a range of individuals or a period in time.

In most cases, the entries or components of a process ("1,2,3,4" in the example above) can be interpreted as states, which the process walks through (e.g. positions on a playground or levels of activity in an EEG-curve). The transition probabilities between these states form a transition matrix as is given in the following example, where s1, … , s4 denote the states (see Table 1).

Table 1. Example of a 4· 4-transition matrix

p(si sj)	s1	s2	s3	s4	sum
s1	0.0	0.1	0.5	0.4	1.0
s2	0.1	0.0	0.4	0.5	1.0
s3	0.1	0.1	0.5	0.3	1.0
s4	0.1	0.1	0.3	0.5	1.0

Such a k· k-transition matrix – with k the number of system states (e.g. values, coordinates, positions) – can be used to generate the system specific processes by means of Monte Carlo and so encodes the basic pattern of the system behaviour.

As technical tests as well as practical applications show, transition matrices are very helpful in particular in cases of low number or continuous flow of input data: In a first step, a DYCON can be trained with a basic system pattern using the system's transition matrix. In the following step(s) this network can be (continuously) trained with the process data derived from the original time series.

4.2 Entropy : measuring learning success and planning training strategies

In order to measure quantitatively the learning success in the meaning of already stored information, a function by means of entropy can be used: Testing a pattern P with a number of P-specific inputs results in an acceptance frequency distribution over the neurones N of the net **N**. Hence, using p(N) as the probability of N to accept P, the P-specific entropy h(P,**N**) of the network **N** mainly is given by $h(P,\mathbf{N})=-_{N\in\mathbf{N}}\, p(N)\cdot \log_2 p(N)$.

As can be seen from Fig. 8, the entropy of a pattern depends on the type of probability distribution – i.e. the distribution of frequencies with which the individual neurones accept inputs belonging to that pattern: In the left case, all neurones have identical frequency of acceptance – i.e. the pattern is uniformly spread over the network – which (with s the number of neurones and 1/s the respective probability) maximises the entropy to $-s \cdot (1/s \cdot \log_2(1/s)) = -\log_2(1/s)$, and so in the example is equal to $-\log_2(1/10) = 3.32$. Patterns that are not uniformly coined but show "conspicuousness" in form of a non-uniform acceptance have lower entropy as is shown in the middle case. Finally, the right case shows the situation of a uniformly but weakly coined pattern with a lot of not accepted inputs (represented by neurone

10), which can be interpreted as a low acceptance tolerance **T**, resulting in a reduced entropy of about 2.42.

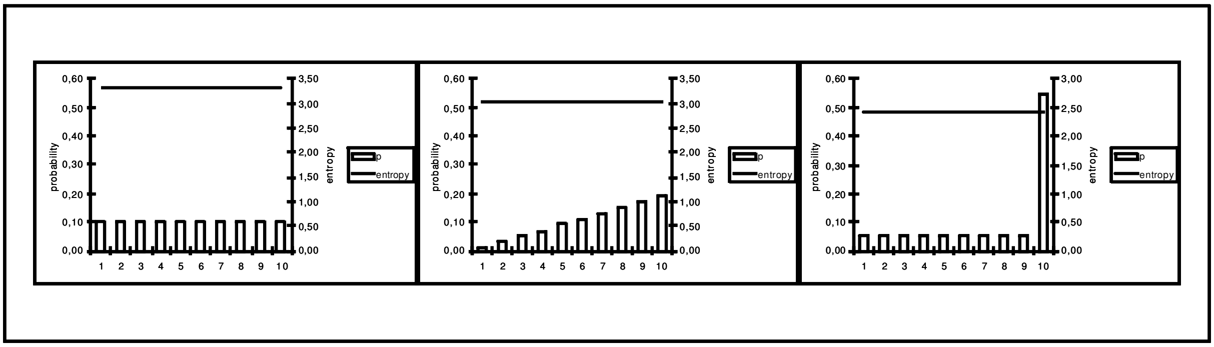

Fig. 8. Entropy depending on the type of probability distribution

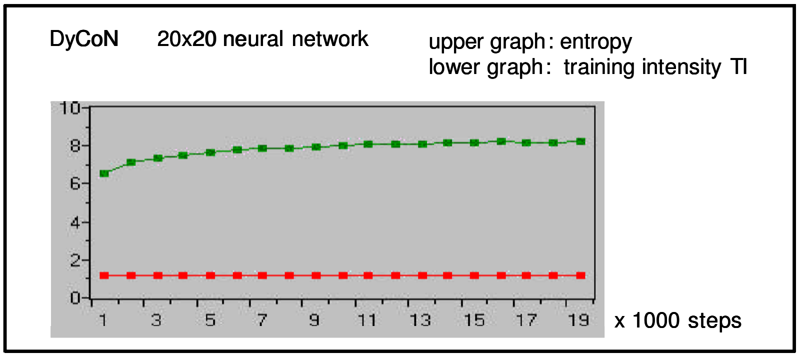

Fig. 9. Entropy in the case of continuously training the same pattern

Fig. 9 shows a typical curve of entropy values during a training process, approximating 8.2, where the maximum value is $-\log_2(20\cdot 20)=8.6$, "20· 20" the number of neurones. The value of training intensity (TI) chosen here is 0.12, guarantying learning stability.

Using entropy as a quantitative measure enables to detect and analyse long term behaviour of a network. In particular the roles of training amount and intensity as well as the cases of concurrent learning, collapsing, and atrophy are of interest, which are dealt with in the following.

4.2.1 Amount and intensity of training

As is shown in Fig. 9, continuously training the same pattern P can increase the entropy of P asymptotically to a maximum, which is given by $\log_2$ of the number of neurones. This is interesting from a principle point of view. In practice, however, it usually is not aspired to become an "idiot with just one perfect skill". Therefore the pure training amount is not really essential. Instead of, the interest normally is focused on reaching a new skill as fast as possible, with sufficient quality, and without reducing existing skills more than necessary. Short but intensive training could be the way to meet these demands. However, increasing the training intensity in the end leads to fading effects and eventually to a total drop out or collapse (see Fig. 10, where the training intensity 0.12 from Fig. 9 is piecewise increased to 0.16 at the marked points). These effects are well-known from different kinds of learning and adaptation. They are the reason for investigations in learning strategies, which optimise speed and quality and avoid fading and collapse effects.

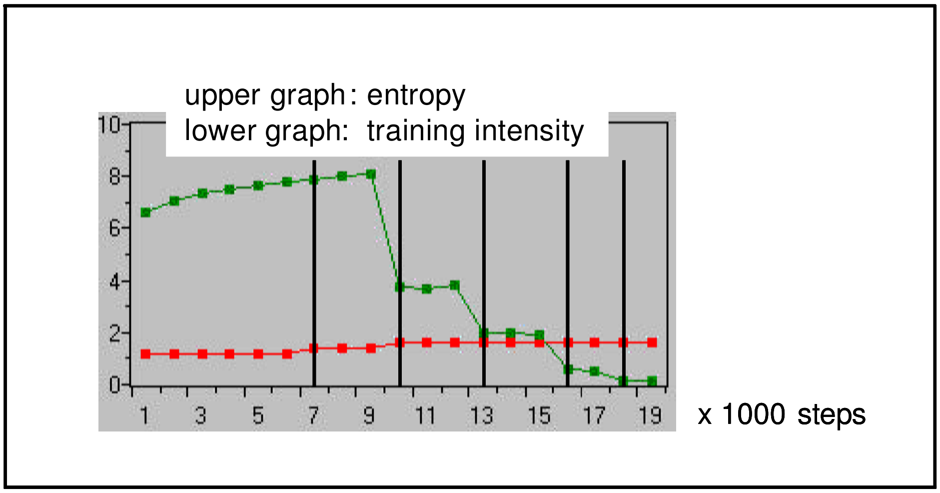

Fig. 10. Overloading and collapsing depending on increasing training intensity

The following examples demonstrate how training strategies can help to optimise the learning of concurrent patterns, and how such strategies can be used to make training more effective and faster.

4.2.2 Superposed, concurrent, and speeded up learning
Fig. 11 deals with the scenario of superposed learning as is shown in Fig. 7: The values of the respective training intensities are indicated by "u" and "l" (upper resp. lower pattern). Picture (a) repeats the situation from Fig. 7, i.e. with constant TI-values, effecting a temporal coexistence of the concurrent patterns. Using too high TI-values for the superposing process, as is shown in picture (b), effects low entropy for both patterns, indicating low grades of representation. If however superposing starts with a high TI-value and is then continued using decreasing TI, as is shown in picture (c), the effect is again bad for the superposed pattern – but is best supporting for the superposing one.

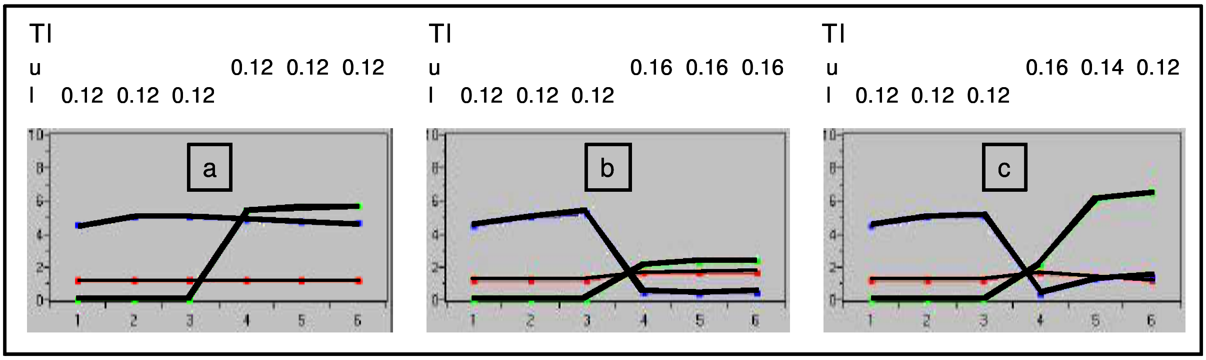

Fig. 11. Entropy in case of concurrent learning, depending on training intensities (TI); each training was done in 2 series of 3 episodes of 1000 steps each.

As further experiments have shown, the same strategy of starting with a high TI-value and continuing with decreasing values also enables very fast learning. Results like these are not only interesting from the technical point of view but might also be helpful in modelling and understanding physiological learning processes, not least in order to reduce duration and load of training.

Finally, the role of atrophy in case of concurrent learning is shown in Fig. 12, where in the first phase the lower pattern is trained, also coining the upper one because of common inputs. In the second phase the upper pattern is trained, effecting an atrophy of the lower one.

In general, atrophy appears to the one pattern because of training the respective concurrent one: During a training phase only those neurones that are accepting a certain pattern input increase their performance potential and so decrease their radius of activation, whereas meanwhile every neurone looses some potential by atrophy, independent on any acceptance. Hence, during a training phase all not involved neurones have their radius of activation decreased continuously and so step by step loose the connections to their neighbours. So by atrophy the pattern specific area of the networks in a way looses it common information so that the pattern is fading out.

To make clear what that means for the qualitative behaviour, the acceptance frequencies of the neurones in Fig. 12 are represented by their diameters. It shows that atrophy reduces the grade of representation of a pattern with respect to the number of involved neurones as well as with respect to their acceptance frequencies.

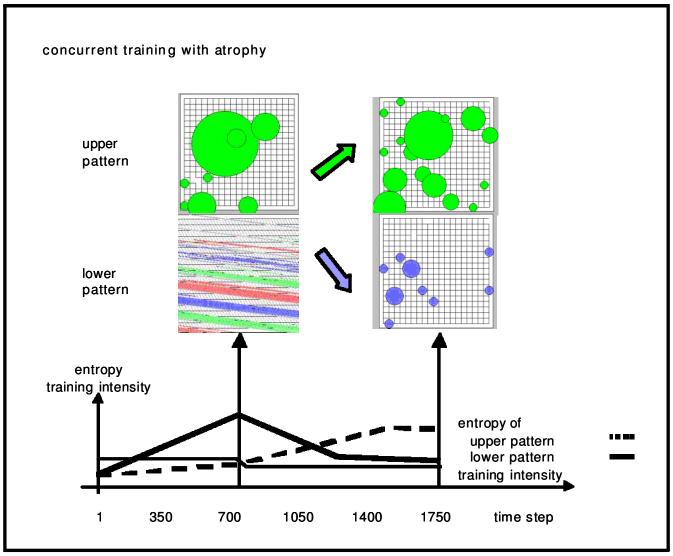

Fig. 12. Concurrent learning controlled by atrophy

According to different purposes, the atrophy in the PERPOT-driven DYCON can be "dimmed" on or off by means of its delay:

If superposed learning of one or more patterns in order to optimal classification and recognition of patterns is of interest, the affect of short term atrophy should be avoided by choosing a large delay value.

If however the focus is on modelling learning processes and optimising training strategies, atrophy has to be handled as a main influence and the delay has to be adjusted very carefully. A well calibrated network then can be used as a model to describe the interactions and relations between learning, overloading, and atrophy in the context of concurrent patterns to be learned to specific grades of representation.

5 Applications

In the following, three possible applications of DYCON are presented, which in our working group currently are dealt with in projects.

5.1 Sport: Game analysis by the example of Squash

My thanks is to Tim McGarry from the Kinesiology Department of the University of Fredericton in Canada, with whom I am in collaboration and who made the squash data available.

The squash data that are analysed exemplarily in the following are the striking positions of the players, encoded by "front-right"=1, "front-left"=2, "rear-right"=3, "rear-left"=4. A process is a sequence of the player's striking positions. (Note that it therefore is not an image of his own but of his opponent's strategy). Recorded and encoded rallies then are number series like (4, 4, 3, 2, 1, 4, …), from which the used processes are continuously cut as sequences of a fixed length – for example <4,4,3,2>;<4,3,2,1>;<3,2,1,4> and so on (compare section 4.1, "n-dimensional test data and patterns").

The main problem with the analysis of squash is the very small number of only about 100 player-specific rallies per game: With an average length of about 8, one rally yields 4 processes of length 5, in total resulting in about 500 processes per game - which is not sufficient for any network training, neither using KFM nor using DYCON.

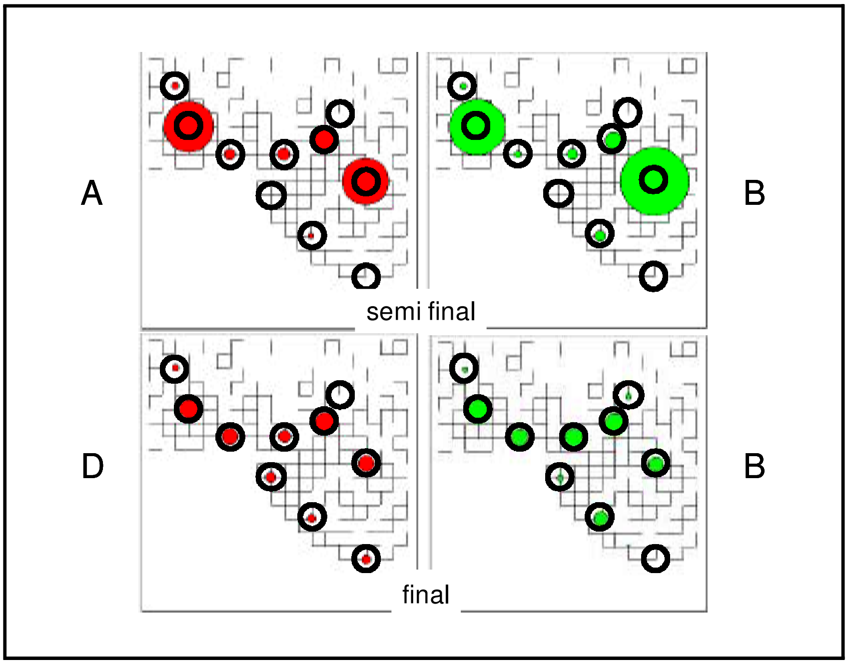

Fig. 13. Player-specific process patterns coined into a squash-specific prepared DYCON

Using DyCoN, however, this problem can be solved by the method of basic preparation: To give DYCON a squash specific basic preparation, in the first step it was trained with virtual processes generated from the 4· 4-transition matrix presented in Table 1, which represents the typical playing structure in squash. In the second step, the original processes are analysed using the prepared network. The pictures in Fig. 13 show three levels of information: On the lowest level, the squash structure is represented, which has been generated from the transition matrix. Its shape looks like a honeycomb, where the edges connect neurones that represent similar processes. (Note, that the number of edges can vary depending on the chosen sensitivity of the graphical representation.) On the second level, small transparent circles of identical size mark those processes that most frequently appear in an average game. So these circles form something like an average or standard pattern of squash. On the highest level, the original processes of a player are represented, where the most frequently played processes are marked by filled circles, the diameters of which represent its respective frequencies. So this level represents the player specific pattern.

From the squash specific point of view, there are mainly two results: The qualitative distributions are similar, independent on players and pairs. The reason is that the most frequent sequences represent long line rallies, which are quite dominant in squash. The quantitative distributions depend – however not on the players but on the pairs. This might mean that a pair of players defines its own dynamic, independent on the specific facilities of the opponents. Again the reason can be that long line rallies dominate a squash match and so influence the process structure significantly.

Finally, a once trained network can be used to analyse the time-depending phases of a game as is presented in Fig. 14: Here, as in Fig. 13, a player-specific trained network was tested with a series of 7 phases, each containing 50 of the players sequences of length 4. As a result the "strategic line" shows the changes of the opponents strategic behaviour over time: Long line rallies in the phases 1 and 2 (big dots in central and lower half), unclear distributions in phase 3 and 4 (varying small dots), stops and drop shots in phase 5 (big dots in upper half), varying long line rallies in phase 6 (big dots in central and lower half), reduction to long line rallies on the right hand side (big dot in the centre).

Deeper analysis should be up to squash experts (e.g. see McGarry et al. (1999)).

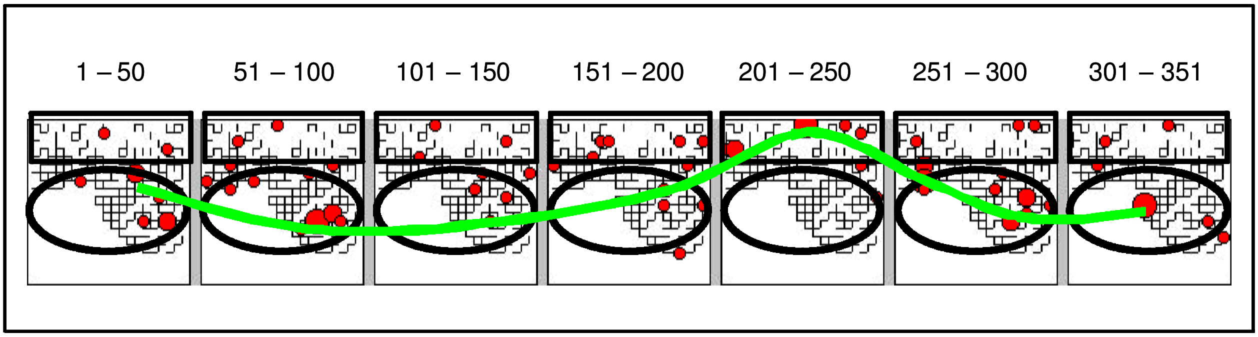

Fig. 14. Phases of a game with a "strategic line" indicating changing strategic behaviour

Among a lot of further examples, some specific approaches of neural networks in sport can be found e.g. in Lames and Perl (1999), Schöllhorn and Bauer (1997) and Wünstel et al. (1999).

5.2 Medicine: EEG analysis

In the same way as has been presented for squash analysis, data can be taken and analysed from different process types. As one example, a project has been started where EEG-curves are analysed in order to detect conspicuous patterns – i.e. patterns that forecast epilepsy attacks to come. Although EEG data are taken in millisecond intervals and so on the first glance are available in huge amounts, we learned that they have to be compressed in order to make long term structures visible. Moreover, most of the regular data are in a specific way periodic or depend on mental activities. So, after compressing and filtering, the amount of usable data again is too low for a successful network training. Very similar to the squash approach, however, DYCON can be prepared with standard patterns and is then able to analyse specific phases to recognise conspicuous behaviour – as e.g. something like the "strategic line" in Fig. 14. In the long run the idea is to feed back the results of such analyses to the patient in order to trigger auto-regulation processes, which could help to even avoid those attacks.

5.3 Physiological adaptation: Training analysis and optimisation

Learning a technique or a motion in sport (but of course not only there) often is a difficult process in so far as existing patterns have to be superposed without erasing them. This for instance is the case if one is learning to play a game: It would not be helpful, if the training of a specific technique would erase a different technique, which had been learned the weeks before. Of course, this problem normally does not appear in practice very often. However, it can be a problem to find the right mixture: In particular in the case of high performed athletes, the intensive training of one technique actually influences not only a different one badly but also can influence the whole playing structure in an unexpected way. The reason is that technical skills in a long term training process tend to interact in a subtle and unstable balance (compare Figs. 11 and 12). The way DYCON can be used here is to calibrate it to the person to be trained and then find out strategies by simulation. In the case of PERPOT this has already been done in order to optimise or minimise training load. A further interesting approach here could be to transfer the idea of concurrent training of different patterns to concurrent training of different load types. As we were told, this would be very helpful in the particular case of multi discipline events (e.g. decathlon) and in comparable training situations.

6 Conclusions

The basic aim was to analyse and model learning as a physiological adaptation process. Given the one-dimensional performance potential-metamodel PERPOT, the idea was to transfer and to generalise this approach to multi-dimensional structures using neural networks. The result is a special type of a dynamically controlled network, where each neurone internally controls itself using its own PERPOT and the information about its specific training state.

The specific skills of this PERPOT-driven DYCON are:

- time-independent control,
- continuous "life-long" learning,
- superposing pattern learning,
- dependence on training amount and intensity,
- dynamic effects of overloading and atrophy,
- measurable and comparable learn performance by means of pattern entropy.

DYCON currently has been used in some projects in the fields of sport, medicine, and physiology. The results are encouraging: The PERPOT-driven control works, and DYCON shows all the demanded and expected phenomena. So there is confidence that the approach will be successful.

7 References

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation.* Prentice Hall (2nd edition).

Kohonen, T. (1995). *Self Organizing Maps.* Springer.

McGarry, T., Khan, M. A., & Franks, I. M. (1999). On the presence and absence of behavioural traits in sport: An example from championship squash match-play. *Journal of Sport Sciences*, *17*, 297-311.

Mester, J., Hartmann, U., Niessen, M. & Perl, J. (2000). Time Series Analysis and Metamodels for Analyses of Physiological Adaptation. In J. Avela, P.V. Komi & J. Komulainen (Eds.), *5th Annual Congress of the European College of Sport Science*. University of Jyväskylä, p. 75.

Mester, J. & Perl, J. (1999). Unconventional Simulation and Empirical Evaluation of Biological Response to Complex High-Training Loads. In P. Parisi, F. Pigozzi, & G. Prinzi (Eds.), *Sport Science '99 in Europe.* Rome: University Institute of Motor Science, p. 163.

Mester, J. & Perl, J. (2000). Grenzen der Anpassungs- und Leistungsfähigkeit des Menschen aus systemischer Sicht – Zeitreihenanalyse und ein informatisches Metamodell zur Untersuchung physiologischer Adaptationsprozesse. *Leistungssport 30* (*1*), pp. 43-51.

Lames, M. & Perl, J. (1999). Identifikation von Ballwechseltypen mit Neuronalen Netzen. In K. Roth, T. Pauer, & K. Reichle (Eds.), *Dimensionen und Visionen des Sports.* Hamburg: Szwalina, p. 133.

Perl, J. (1999). Aspects and Potentiality of Unconventional Modeling of Processes in Sporting Events. In B. Scholz-Reiter, H.-D. Stahlmann, & A. Nethe (Eds.), *Process Modelling*. Springer, pp. 74-85.

Perl, J. & Mester, J. (2001). Modellgestützte und statistische Analyse der Wechselwirkung zwischen Belastung und Leistung. *Leistungssport 31* (*2*), pp. 54-62.

Sauer, T. (1993). Time Series Prediction Using Delay Coordinate Embedding. In Weigend & Gershenfeld (Eds.), *Time Series Prediction*. Institute Studies in the Science of Complexity, Proc. Vo. XV. Santa Fe. Reading, Addison-Wesley.

Schöllhorn, W. & Bauer, H.-U. (1997). Linear vs. Nonlinear Classification of Complex Time Course Patterns. In *Proceedings of the 2nd Annual Congress of ECSS*. Kopenhagen.

Wilke, P. (1997). *Soft Computing - Prinzip, Simulation und Anwendung*. Thesis. Erlangen-Nürnberg University.

Wünstel, M., Boll, M., Polani, D., Uthmann, T., & Perl, J. (1999). Trajectory clustering using self-organizing maps. In S. Sablatnög & S. Enderle (Eds.), *Workshop RoboCup at KI'99 Bremen*. Ulm University: Report 1999/12, pp. 41-46.