Attention Driven Memory

Steffen Grünewälder (gruenew@cs.tu-berlin.de)
Department of Computer Science, Technical University Berlin,
10587 Berlin, Germany

Klaus Obermayer (oby@cs.tu-berlin.de)
Department of Computer Science, Technical University Berlin
10587 Berlin,Germany

Abstract

Categorization is a skill which is used extensively in everyday life and as therefore an important aspect of human cognition. Consequently a variety of studies exist which address the topic and revealed that diverse factors affect human categorization performance. A critical but not extensively studied factor is time. Imagine watching a basketball game for 30 minutes. In this period of time plenty of actions will take place resulting in diverse impressions which make you afterwards categorize the game as interesting or boring. Such a categorization task is very similar to a time series classification task in the context of machine learning. In the field of machine learning a phenomen called "vanishing gradient" is known which makes it generally hard to solve such a categorization task. A prominent method that overcomes this phenomen is the long short term memory which basicly consists of a memory that is controlled by two gate units which can be interpreted as adaptive encoding and recall units. Critical points which make the processing of the structure differ from human processing concern the encoding and the storage: (1) The structure is built to massively store information instead of carefully selecting few impressions for storage in memory. (2) Reweighting of stored information due to changing constellations is not possible. Coming back to the example this would mean that a nice action at the beginning of the game has a strong impact on your categorization independent of what kind of actions - might they be impressive or not - followed afterwards. In this work we tackle these points through introducing an attention mechanism which drives the encoding and the storage of the structure. We analyse the model behavior in category learning tasks.

Keywords: LSTM, Memory, Attention, Categorization, Modelling.

Introduction

Diverse factors like category dimensionality (Shepard, Hovland, & Jenkins, 1961), covariance complexity (Alfonso-Reese, Ashby, & Brainard, 2002) or, for rule-based task, if a rule is conjunctive or disjunctive (Bourne, 1970) are responsible for human categorization performance. Another important but not extensively studied factor is time. Categorization tasks which involve the temporal domain come up all the time in every day life, think for example of the different ways you can take to your work place. You will surely have categorized them according to the time you need to arrive at work.

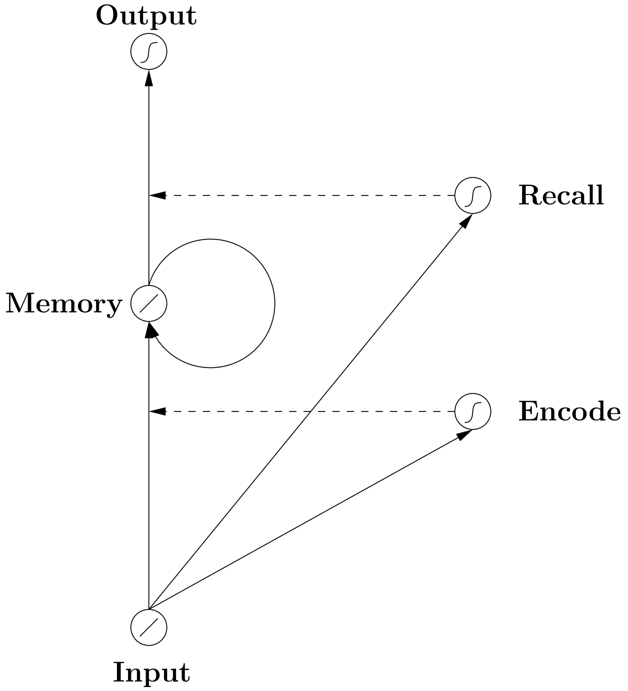

Figure 1. Interpretation of the LSTM. The central element is the memory which is controlled by adaptive encoding and recall units. Encoding/recall activation scales the memory input/output through a multiplicative relation to the ordinary pathway (dashed line). The hidden layer (memory, encode, recall) is fully connected (not drawn due to readability).

In the machine learning field similar classification problems are studied, whereas the machine learning methods here face a similar problem like a human. The information they need for classification is spread over time so not all of the relevant information is available at one moment. Just like a human these methods have to memorize relevant information to yield good performance in the task.

Interestingly, studies in the machine learning field revealed that such classification tasks are generally hard to solve. This mainly relies on the so called *vanishing gradient problem* (S. Hochreiter (1991), Bengio, Simard, and Frasconi (1994) and S. Hochreiter (1998)). This "problem" comes up when stored information decays with time. This results in the effect that information that might be strongly related to a later signal being downscaled with time and finally when the later signal comes the original information can no longer be distinguished from noise. This way the relation between the singals can not be revealed, respectively learned. This has dramatic effects on the performance of methods which have the problem. Due to this effect traditional recurrent networks[1] are hardly able to relate two signals which are 15 time steps or more apart from each other.

A prominent machine learning method that overcomes this problem is the Long Short Term Memory (LSTM, J. Hochreiter and Schmidhuber (1997)). The LSTM is a recurrent neural network structure which basicly consists of a memory unit that is controlled by adaptive encoding and recall units (see figure 1 and next section). Similarities of the LSTM structure to recent neuropsychological models of working memory exist as stated in (O'Reilly, Braver, & Cohen, 1999).

Despite these similarities some obvious differences still exist between the processing through the LSTM structure and human processing of information. Two points are especially critical: (1) The LSTM tends to massively store information in its memory in contrast to the very selective storage in humans. (2) Information that is once stored can not be reweighted due to changed conditions.

We address these two points through introducing an attention mechanism which drives the encoding unit and rescales the memory content when needed.

Furthermore, we analyse the behavior of this process model in category learning tasks.

Long Short Term Memory (LSTM)

Figure 1 shows the LSTM structure (J. Hochreiter & Schmidhuber, 1997). The key element is the memory unit in the center of the LSTM. The unit uses the identity as an activation function and is self-connected with a recurrent connection. This recurrent connection has a weight of one which garanties that no information is lost, respectively information is conserved and so the vanishing gradient problem does not apply.

A problem that occurs directly from the conservation of all information which is encoded in the unit, is that the capacity of the memory unit will be quickly exceeded. This problem is overcome through introducing an encoding unit which controlls the inflow of information into the memory. Besidea that a recall unit is used to control the flow of information out of the memory cell back into the network.

The control of the inflow and outflow is achieved through scaling the input to the memory respectively the output from the memory, with the activation of the encoding/recall unit. Whereas both units got a sigmoidal activation function in $[0, 1]$. Thus when the activation of these control units tends to zero the information flow is stopped.

An important aspect about the encoding and recall units is that they are adaptive and through training learn when to store information and when to use stored information.

Attentional Driven Encoding and Storage

Encoding in human memory is driven by attention. This has the advantage that the memory can be used according to current needs or objectives. Furthermore, storing information selectively reduces the needed memory and this way it is easier to detect a relation between past attended and present events. A disadvantage is, that when a relation exists between past and present events, but attention is not drawn to the relevant past events then the relation will hardly be detected. Encoding in the LSTM varies in these points from human encoding. The LSTM massively stores information. Therefore problems arise with the memory capacity. Addionally, through this massive storage the LSTM is on the one hand able to detect a wide variety of relations, however, on the other hand it takes a long time to extract relations. Thus, a well working attention mechanism should be able to improve the LSTM performance, make its encoding behavior more human like and thus make it a memory structure whose processing is more similar to human information processing.

We start from a basic and general assumption of how the memory content should be computed out of input and attention values at different time steps. It shows up that critical elements of the attention process are predetermined from this first assumption and thus at the core of the attention structure no variations are possible when the defined equation (1) should hold.

Basicly we want the memory content to "concentrate" on few strongly attended information and to downscale the impact of information which has a low attention value relative to the strongly attended signals. Due to the vastly reduced amount of stored information that must be processed, detecting relations between attended signals should be much easier. Further, we don't want the storage of information to rely on the special timestep when the information arrives. So basicly the memory content at time T should look like this.

$$mem_i(T) = \sum_{t=1}^{T} in_i(t) \cdot \phi(A_{max}(T) - A(t)) \qquad (1)$$

ϕ denotes a function which operates on the relative value between the strongest and the actual attention value at step t. If the relative value is high the input should be strongly downgraded.

[1] For an introduction to neural network models see (Haykin, 1999)

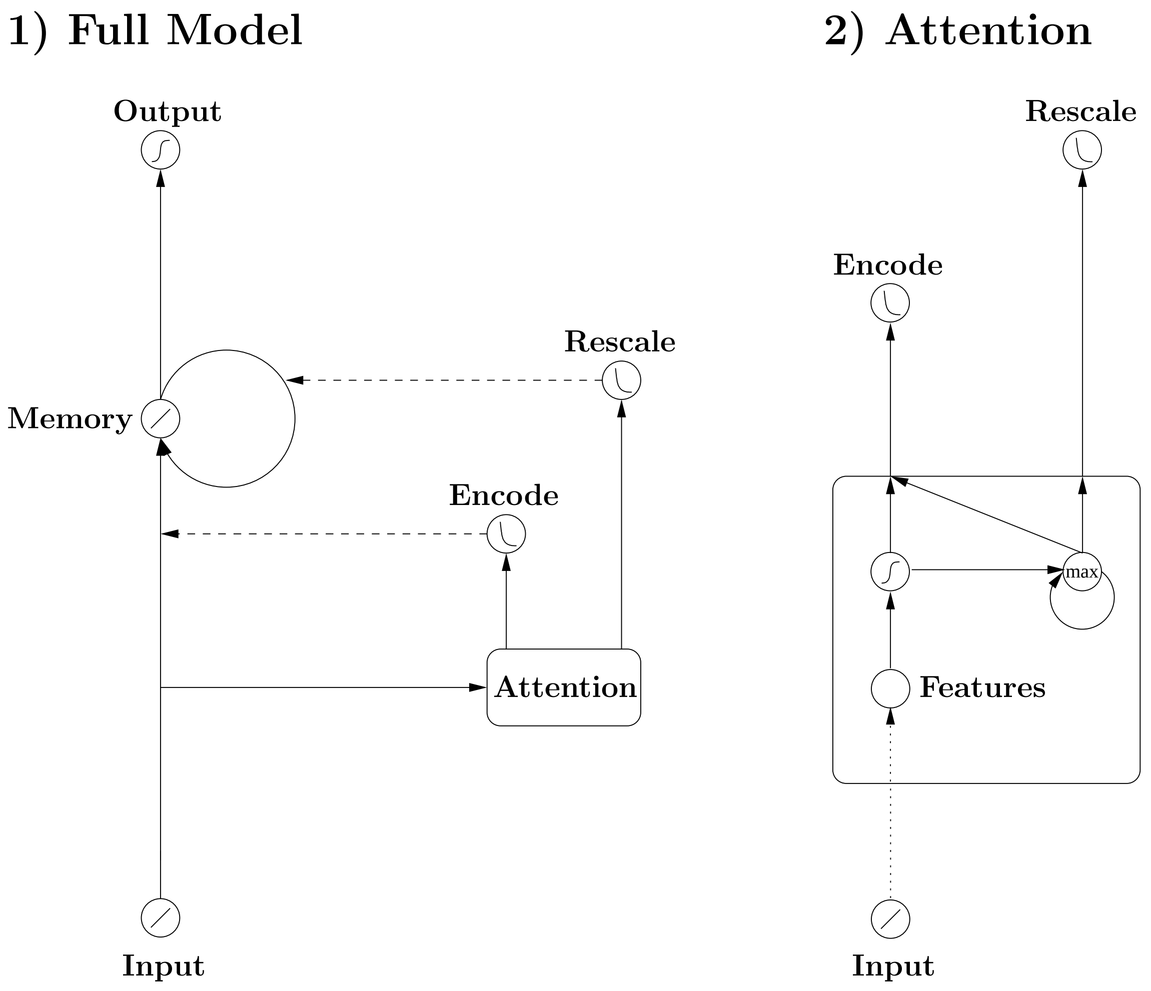

Figure 2. 1) Full Model: Modified LSTM structure. Like in the LSTM the central element is the memory cell and the input to the memory is controlled by an encoding unit. In difference to the LSTM the encode and rescale units are driven by an attention process which weights the input according to its relevance and reweights the memory when new constellations come up (e.g. a new signal attracts most attention). Encode and rescale units have exponential activation functions which are used to emphasize strong attented input and downscale input of low attention value. 2) A detailed view on the attention box of the left part. The input activates a set of features which lead to an attention value. This attention value is compared with the stored maximal attention value and replaces this when it is higher. It further drives together with the maximal attention value the encode unit. The rescale unit on the other hand is driven by the difference between the new and the old maximal attention value.

When storage is limited and thus not every $A(T)$ and $in_i(t)$ can or should be stored it is not simply possible to rescale the stored information when A_{max} changes. What we need is a function ϕ that makes it possible to rescale the stored information in the way that

$$\begin{aligned} mem_i(T) = & \overbrace{K \cdot \ \ mem_i(T-1)}^{\text{rescale}} + \\ & \underbrace{in_i(T) \cdot \ \ \phi(A_{max}(T) - A(T))}_{encode} \end{aligned}$$

To do so it must be possible to transform the '-' into a multiplicative factor, in other words $\phi(A_{max}(T) - A(t)) = \phi(A_{max}(T)) \cdot \ \ \phi(-A(t))$. This only holds for the exponential function thus $\phi = exp$. After introducing an additional minus and a scaling factor σ we get

$$mem_i(T) = \sum_{t=1}^{T} in_i(t) \cdot exp(-\sigma(A_{max}(T) - A(t)))$$

The rescaling factor K computes now to

$$K = exp(\sigma(A_{max}(T-1) - A_{max}(T)))$$

To guarantee that the equation holds in every step we only need to store the memory value $mem_i(T)$ and the highest activation value $A_{max}(T)$ that has come up till the time step T.

In the next section we present a model which is based on these equations.

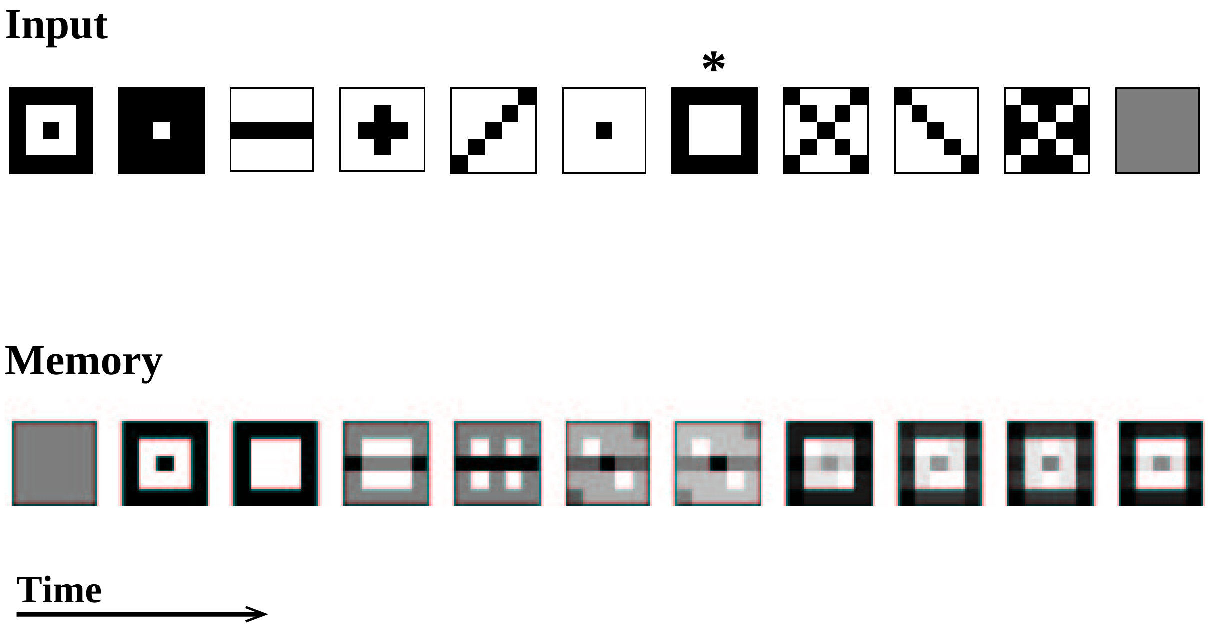

Figure 3. We presented a sequence of 15 steps to a network. In the first 10 steps cards were drawn randomly with equal probability from a pool of 20 cards, whereas each card could appear only once. In the final 5 steps no card was presented. At step 15 the network got an error signal for it's classification response. In the upper part one such sequence is shown. The sequence is presented from left to right with one card per time step. The relation of the sequence and the class was deterministic. If the black square card (* in the figure) was presented the class of the sequence was 1 otherwise the class was 0. The variables $F_i(t)$ where chosen to be reactive to only one of the cards, so we got 20 variables (one for each card). This way the calculation of $A(t)$ was trivial and the emphasis layed on the weighting over the different time steps. The according weights of $F_i(t)$ were initialized with a value of 0.2 whereas the weights between the memory and the output layer were chosen randomly in $[-0.1, 0.1]$. The bias values where negatively initialized (-2) to initially hold attention values low. The lower part shows the memory of a trained network (20 000 training steps, mean squared error < 0.1). At the position of the correct card the memory is rescaled and the stored values are strongly downgraded. At the same moment the input is stored with a high attention value and overshadows the other input signals.

Model Description

Figure 2 shows the model structure. The input is passed to the attention process and to the memory. The attention process computes an attention value for the momentary time step due to the input. According to this attention value it drives the encode and rescale units which will control if the new input is stored, respectively how "strong" it is stored, and how the memory content must be reweighted due to changed conditions.

Diverse factors exist which drive our attention in one special moment. An example is our physical condition. When we are hungry or thirsty totally different objects attract our attention than when we are tired. Also totally different conditions like a task we are working on at the moment or personal preferences effect what attracts our attention. The study of the influence of these points is beyond the scope of this work. Instead of realizing an attention mechanism which is driven by all these factors we used some abstract variables $F_i(t)$ which might be used to represent a variety of features which are relevant to control attention. The only condition the variables must fulfill is that the values the variables can take lie in a bounded interval.

Due to these features the attention value at time step t calculates in our method as follows:

$$A(t) = f(max(\hat{w}_i F_i(t) + b_i))$$

Here max is the maximum function and f is a logistic sigmoidal function. Variable $F_i(t)$ is dependent on the input and as stated above simulates a feature detector that signals if a feature i is present at time step t (in simulations the range of $F_i(t)$ was chosen to be in $[0, 1]$). A short remark: the function to calculate $A(t)$ must not have the upper form, one might for example also use a weighted sum of the different feature activations. Finally, in the above equation $\hat{w}_i$ is the weight of feature i and b_i a bias. Both are adaptive and modulated through learning.

Contrary to the choise of $A(t)$ the following variables are predefined through our initial assumption (equation 1).

As stated above the maximal attention value which appears until step t is needed to encode and rescale:

$$A_{max}(t) = max(A(t), A_{max}(t-1))$$

Building up on the value $A(t)$, $A_{max}(t)$ and $A_{max}(t-1)$ the encode and the rescale value can be calculated:

$$encode(t) = exp(-\sigma(A_{max}(t) - A(t)))$$
$$rescale(t) = exp(\sigma(A_{max}(t-1) - A_{max}(t)))$$

Finally the activation of the memory at time step t calculates as follows:

$$mem_i(t{+}1) = in_i(t){\cdot}encode(t){+}mem_i(t){\cdot}rescale(t)$$

And the output of the network:

$$out(t+1) = f(\sum_i w_i mem_i(t) + b)$$

In this simple setup adaptive parameters that must be learned are the w_i values which weight the impact of the memory content to the output of the network and the $\hat{w}_i$ values which weight the impact of the different features to the attention value at one time step. The latter ones are the interesting ones, because they define how the network uses its memory. Another remark: in the LSTM input is bundled through a weighted sum and thus concentrated in one memory unit. This can also be done here through introducing another weight layer between the input and the memory. The attention at one special moment will in this case have a large impact on the learning of the input weights. Input units that are active when attention is high and thus the encode unit stores information will be strongly effected by a learning process.

Training the network can be done with any gradient descent algorithm which does not rely on higher derivatives (s. below). We used back-propagation through time (Haykin, 1999) for training. The derivations are basicly straight forward we only want to single out two of them. First, the derivation $\frac{\partial A_{max}(t)}{\partial A(t)}$ because the maximum function is involved. The maximum function is derivable, however its derivation is not continuous:

$$\frac{\partial A_{max}(t)}{\partial A(t)} = \begin{cases} 1, & A(t) \geq A_{max}(t-1) \\ 0, & otherwise \end{cases}$$

This makes no problem for calculating the gradient and thus to train with a gradient descent method which does not rely on the second or higher derivations.

The other derivation which we want to single out is

$$\frac{\partial mem_i(t+1)}{\partial mem_i(t)} = rescale(t)$$

This one is important because of the vanishing gradient problem ((S. Hochreiter, 1991), (S. Hochreiter, 1998) and (Bengio et al., 1994)) which makes it hard to process temporal information. Like in the original LSTM in our model this problem does not apply. The reason for this is that the rescaling for stored information is bounded, as long as the range of A(t) lies in a bounded interval (for our case $[0, 1]$), information, respectively the gradient, can not vanish. The upper bound for rescaling is 1, which is trivial and the lower bound can be infered in the following way.

Because $A_{max}(j) - A_{max}(i) \geq -1$ for $i > j$ due to the choise of the function f and thus

$$\begin{aligned}\prod_{k=i+1}^{k=j} rescale(t) =& exp(\sigma(A_{max}(i+1) - A_{max}(i)))\cdot \\ &\cdot\ldots\cdot\ exp(\sigma(A_{max}(j) - A_{max}(j-1))) \\ =& exp(\sigma(A_{max}(j) - A_{max}(i))) \\ \geq& exp(-\sigma)\end{aligned}$$

the overall rescale value summed over all time steps is lower bounded by $e^{-\sigma}$. Thus stored information can not be rescaled arbitrary low and thus will in all situations still have a noticeable effect. This leads to the statement that the vanishing gradient problem does not apply.

Simulation

We evaluate our model in category learning tasks. The tasks we consider here are rule-based category learning tasks (Ashby & Shawn, 2001). In the simulations we presented the network a set of cards (see figure 3) whereas at each point in time only one card was presented. Thus the networks had to learn to memorize information about the seen cards to solve the categorization task. The relation between the presented cards and the category was deterministic, if a special card was present in the sequence then the according category was one, otherwise zero. Each card was represented through a 5x5 matrix of input activation. To each input unit directly one memory unit corresponded, thus we also had a matrix of 5x5 memory units. Input to the memory, respectively the rescale of the memory, was however controlled by just one unit. So a single attention mechanism was responsible for the inflow of information to the 5x5 memory units.

In the first simulation we were interested in how the memory is used from a network which had learned to solve the categorization task (see figure 3). It shows up that the network has learned to extract the relevant feature and store it in the memory whereas the other input elements are downscaled when the relevant feature is seen.

We made a second experiment to study how the performance of the network changes when attention is directly drawn towards the correct feature against the case when attention is high for irrelevant features and low for the correct one. Figure 4 shows the results. An interesting side effect became

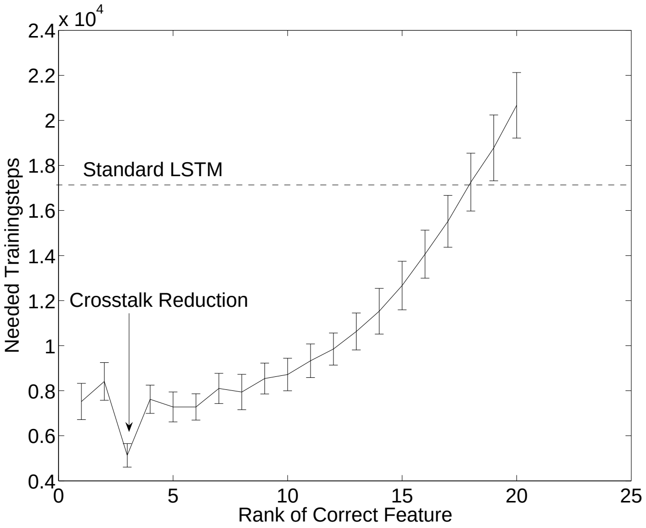

Figure 4. We measured how the initial weighting of the correct feature and its relative weighting to other features effects the performance of the model. The signals and features were chosen like in simulation described in figure 3. We used the number of trainingsteps which are needed to bring the mean squared error below 0.1 as a measurement for the performance of the network. We initialized the feature weights in the following way: $\hat{w}_i = 0.1 \cdot i$ and changed the position of the correct feature. So in the worst case the initial weight was 0.1 and in the best 2.1 for the correct feature. For each setup we made 100 runs. Notice that beside the effect of the rank also interference effects came up because the different cards we presented had similarities. Especially the cards which were similar to the black square card (this was the relevant feature) had an impact on the performance. This becomes most obvious in the plot at position 3. Here the feature responsible to detect the black square card with a dot in center (in figure 3 the card in the upper left) was initialized with a low weight which resulted in a low attention value when the card came up. Because this card is very similar to the relevant feature the reduced attention to this card boosted the performance of the network. We also made a run with the standard LSTM to compare the perfomance. We have chosen the same learning rate as in our model and initialized the other parameters similar to (J. Hochreiter & Schmidhuber, 1997). The mean number of training steps was 17640 and standard deviation was 781.28 (dashed line in the figure).

apparant in this simulation. The cards we have chosen are in parts similar. During the simulation it became clear that this has a considerable impact on the performance of the network. Especially as the performance was strongly boosted when a card which was nearly the same as the relevant card attracted no attention (the arrow in the figure).

Conclusion

An important factor in categorization is time. Categorizing events which are spread over time puts humans in a similar position as machine learning models which have to classify time series. A prominent machine learning method, the Long Short Term Memory, which successfully processes time series has some similarities to recent neuropsychological models of the working memory. However different points make its processing behavior vary from human processing. Especially as it tends to massively memorize information and is not able to reweight memorized information due to changing constellations. We showed that these points can be tackled through introducing an attention mechanism which drives the encoding and the storage process. Especially, when the attention mechanism is working well, the performance of the network increases and learning becomes faster than with standard LSTM. However, the network performance strongly relies on the quality of the attention process (in our simulations an exponential relation shows up), thus when attention is attracted by irrelevant signals and the relevant signals are omitted then performance dramatically decreases. Beside this point the simulations revealed that the network is very sensitive to the effects of interference between input signals.

References

Alfonso-Reese, L. A., Ashby, F. G., & Brainard, D. H. (2002). What makes a categorization task difficult? *Perception & Psychophysics*, *64*(4), 570-583.

Ashby, F. G., & Shawn, W. E. (2001). The neurobiology of human category learning. *TRENDS in Cognitive Sciences*, *5*(5), 204-210.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*, 157-166.

Bourne, L. E. (1970). Knowing and using concepts. *Psychological Review*(77), 546-556.

Haykin, S. (1999). *Neural networks: A comprehensive foundation* (2 ed.). Prentice Hall.

Hochreiter, J., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Hochreiter, S. (1991). *Untersuchung zu dynamischen neuronalen netzen.* Unpublished master's thesis, Technische Universität München, Germany.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(2), 107-116.

O'Reilly, R. C., Braver, T. S., & Cohen, J. D. (1999). A biologically based computational model of working memory. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (p. 375-411). New York: Cambridge University Press.

Shepard, R. N., Hovland, C. L., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs*, *75*(13).