Sequential Metacontrast Paradigm Modelling with Recurrent Neural Networks

Marie Anselmet
EPFL, Switzerland, July 2020

***Abstract*—Like several other psychophysic tests realized on human subjects, the Sequential Metacontrast paradigm (SQM) remains a precious tool for the study of the temporal feature integration of the human visual system. More generally, its realization provides insights about the neural processes involved in visual perception. Continuing a previous work [1], this project tried to build a robust framework based on recurrent neural networks (RNNs). It aimed to model the SQM and reproduce the practical results obtained on human subjects, in order to gain an understanding in the processes that may be involved in the temporal feature integration and the visual grouping stage observed in humans. This project failed to obtain satisfying results, therefore this paper merely summarizes the approach that was adopted and proposes some ideas of further improvements.**

I. Introduction

Even if the visual perception seems to be a natural task performed so many times during our daily life, the sensory information starting from the retina photoreceptor cells must be integrated over time to perceive, likely involving complex recurrent computations and neural dynamics. The SQM test is particularly relevant to study this temporal integration, and is described in Figure 1. From previous SQM experiments performed with human subjects, it was shown that temporal integration is mandatory, lasts up to roughly 450 ms depending on the observer, and occurs within discrete time windows [2]. More precisely, a discrete integration time window begins with the stimulus onset, features are mandatorily integrated within this same time window, and a kind of read-out is performed when the latter ”closes”. This read-out is then sent to form a percept. That is, the perception occurs only at discrete times and it is the ”belongingness” to the same time window that determines the integration of features together.

If feedforward convolutional neural networks (ffCNNs) have proved their performances on computer vision tasks like the famous AlexNet for image classification [3], it seems that human-like performances of ffCNNs do not necessarily imply human-like computations, principaly for architectural reasons [4]. On the other hand, recurrent neural networks (RNNs) allow to model recurrent neural connections and feedbacks through space and time, closer to the real dynamics of the human brain. For this reason, recurrent neural networks (RNNs) were implemented in this project [1].

Starting from a previous work [1], the aim of this project was thus to implement a model built on some hypothesis, which is able to reproduce the SQM results observed on human subjects, in order to gain insights about the human brain computations and mechanisms truly involved in this temporal feature integration and perception discreteness.

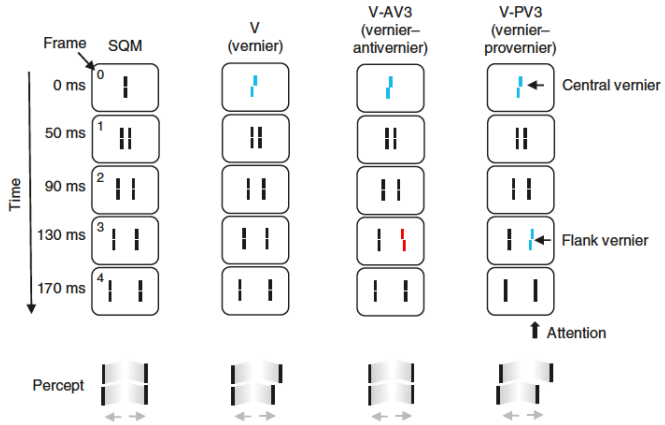

Fig. 1. Figure adapted from [2]. Only the implemented configurations are shown.
SQM: The Sequential Metacontrast paradigm (SQM). A central line is followed by pairs of flanking lines diverging from the center in opposite directions. A percept of two diverging lines is elicited in human subjects.
V condition: only the central line is offset such that a dominance of about 75% is obtained here.
V-AVn and V-PVn conditions: an additional flanking line is offset n frames after the central vernier frame, either in the opposite direction (antivernier) or in the same direction (provernier) relative to the central vernier. If the two offset frames belong to the same time window, they are integrated together and the shown resulting percept is elicited.

II. Methods

A. Implementation of the model and core project structure

For the sake of practicality, a wrapper class containing a RNN model, a reconstructor and a decoder, as well as all the methods needed to train and test the architecture, was used.

a) The PredNet model: Here, the PredNet model [5], a predictive recurrent neural network inspired by the concept of ”predictive coding” and developped for future frame prediction tasks, was used. More precisely, the PredNet deep recurrent convolutional architecture implements simultaneously both top-down and bottom-up connections. While the top-down connections update the LSTM states of the network, the bottom-up pass enables to compute the frame predictions and the resulting predictions errors, propagating them back in the layers to update the latent representations (and thus the predictions). As PredNet also produces its own reconstructions, no reconstructor had to be added.

b) The decoder: In addition to simple dense decoders, fully convolutional decoders [6] were implemented, motivated by the reduction of the number of parameters to train and the more general and reusable structure of convolutional layers.

B. Training

In order to train the framework to be able to reproduce the SQM, the first step of the approach was to train the PredNet model on future frame prediction over the most relevant possible datasets. The second step was to reuse these weights as a baseline to ensure a good internal representation of the input frames to then train the decoder on the classification task of moving verniers.

Inspired by the PredNet paper [5], the reconstruction loss was measured with a weighted L2 metric, penalizing more prediction errors made on the first frames than on later frames. The idea being to force the model to learn an accurate representation of the input starting from the first frames, enabling to build as soon as possible a reliable baseline representation for the reconstruction of the next frames. On the other hand, the decoder loss was implemented as a simple binary cross entropy loss between the distribution of the true offset directions and the expectation of the predicted ones.

For both the PredNet model and the decoder, the training was performed with the ADAM optimizer taking care of moment estimates of the gradients, and a learning rate finder was implemented to estimate the optimal initial learning rate. For the training of both PredNet and the decoder, the following schedule rule derived from [5] was used: the learning rate was divided by a factor of 10 halfway through training.

The training of the decoder was first performed on the entire sequences of frames, that is, all the reconstructed frames were sent to the decoder to train the classification of the verniers. Nonetheless, this approach lacked any explicit discrete stage, and was therefore transparent to the "readiness" state of the latent variables to build a reliable percept.

C. Implementation of the discreteness

To implement a more realistic modelling of the human-brain computations, it seemed important to integrate some discreteness when reconstructed frames are sent to the decoder to classify verniers. A direct consequence is a gain in efficiency, since this would reap the benefits of the "unconscious" latent dynamics of the PredNet model to only send the frames to the decoder when they are in a conductive state, as a "conscious percept".

We could have tried to impose a discrete readout time, but it would very likely have strongly biased the expected result by imposing a constraint with a such prior assumption. One other possibility, slightly less specific, was to measure some quantities about the latent neurons, as the entropy and the sum of the activities of the latent neurons (expected to be correlated with the sum of the prediction errors made by the network). We say here less specific in the sense that these quantities were first computed directly from the latent variables state of the PredNet model (and not designed to fit something), even if it is debatable to qualify them as emanating of the intrinsic behavior of the network given that we "orient" our computation towards something.

Therefore, based on these quantities, a simple criterion was implemented as follows: a 2D convolution was performed between a unity kernel of adjustable size and the numerical approximation of the gradient of the desired quantity (entropy or sum of the activities). A threshold on this resulting smoothed gradient was then fixed to find a region of stabilization of the latent "unconscious" dynamics, determining this way for each parallel sample the frame to send to the decoder to perform the readout of the result of the -expected- temporal integration.

D. Datasets

Different datasets were tried to train the next frame prediction task achieved by the PredNet model, with the goal of selecting the datasets that require more integration. First, moving balls bouncing on walls were used. Nonetheless, PredNet was still trying to reconstruct balls bouncing on some walls when verniers were presented to train the decoder. For the results shown in the Results section, we just trained PredNet on moving shapes, including verniers, which present the advantage to be sufficiently thin to guarantee that the network does not just copy the previous frames.

E. Adding noise

To force the integration of the frames, occluders were first added in the dataset for the training of the reconstruction. An adaptable level of noise was also added in the form of an additional Gaussian noise layer as input for both the PredNet model and the decoder. The idea was that the addition of noise in the training of PredNet would force the model to better integrate the information over several frames. Indeed, with a higher level of noise, the model can not only rely on the bottom-up connections, but is constrained to rely more on the recurrent dynamics of the latent variables. Moreover, the addition of noise in the decoder seemed not as important, thus the level of noise for the decoder was set to a very small value.

F. Testing

Using the previously trained weights for the PredNet and the decoder, testing was finally implemented for the vernier, vernier-antivernier and vernier-provernier conditions. Each test modelled a number of subjects, for which a number of trials was performed.

III. Results

Several experiments were performed, but unfortunately, our modelling of the SQM did not led to satisfying results. It was however decided to present the results of one of the experiments that were performed.

A. Training procedure

The reconstruction achieved by PredNet was trained on moving shapes, with occluders and a high level of Gaussian noise (0.9 precisely). The training procedure was the following: 50 epochs with 9 input frames and 50 epochs with 13 input frames, with an initial learning rate of 5e-4 estimated with the implemented learning rate finder. The training loss for the reconstruction is shown in Figure 2, and its shape decrease indicates an efficient training.

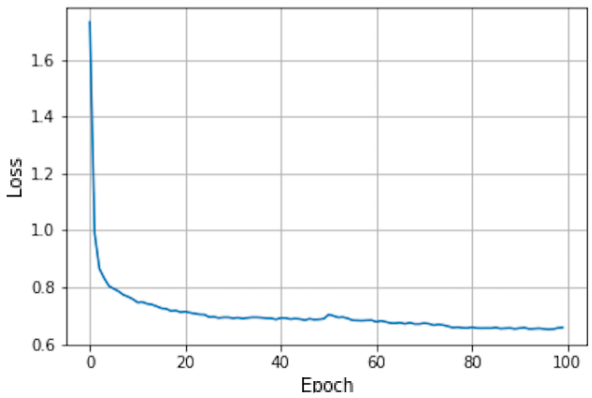

Fig. 2. Training loss of the reconstruction in function of the epoch, for the training procedure described in the Results section.

The inputs of the network and the predictions made by PredNet can be compared for the beginning and the end of the training procedure in the Figures 3 and 4.

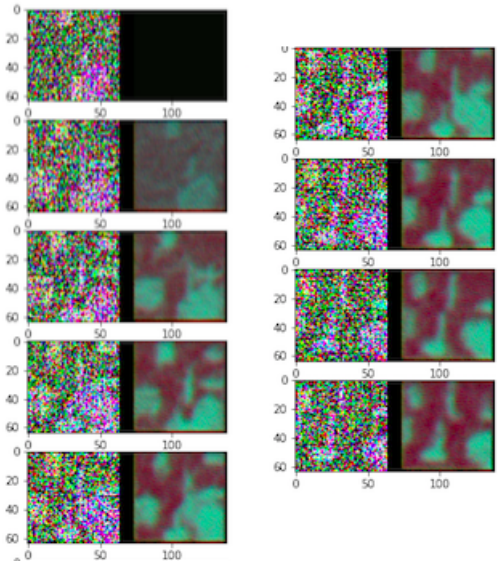

Fig. 3. "Ground truths" inputs of the network (columns 1 and 3), and the associated predictions made by the model (columns 2 and 4) at the epoch 0 of the PredNet training.

By looking at these two figures, we can observe that the training is efficient. Also, despite the high level of Gaussian noise, the model learns to generate good predictions without copying the previous frames. This can be confirmed by the predictions made by PredNet for thin SQM-like verniers (see Figure 5).

The decoder was trained on SQM-like moving verniers, since with simple randomly moving verniers it was too hard to reach the desired 75% of dominance for the vernier condition. The level of input Gaussian noise was set to 0.1 for the training of the decoder. 3 frames of background were added for the computation of the criterion to normalize the latent variable activities by "removing" the intrinsically active

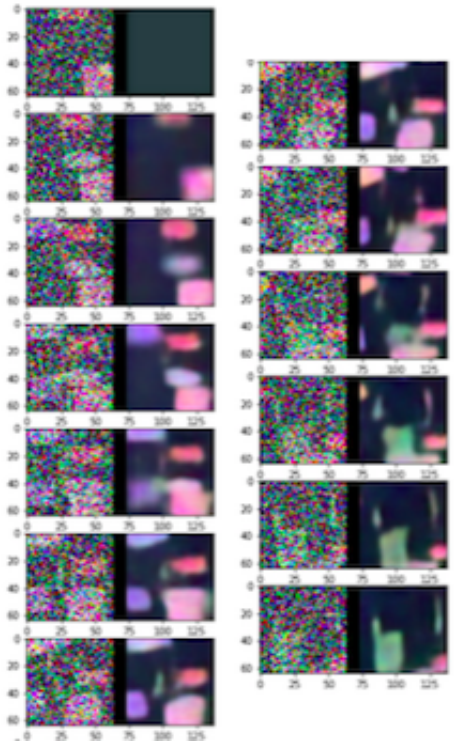

Fig. 4. "Ground truths" inputs of the network (columns 1 and 3), and the associated predictions made by the model (columns 2 and 4) at the epoch 100 of the PredNet training.

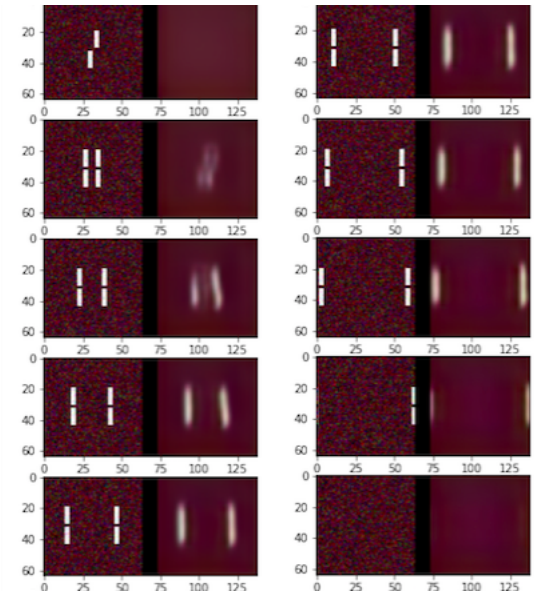

Fig. 5. "Ground truths" inputs of the network (columns 1 and 3), and the associated predictions made by the model (columns 2 and 4) on SQM-like verniers.

connections of the network. The decoding criterion that was used is the entropy, which can be seen in Figure 6.

We can observe on this curve that the entropy increases as soon as the sensory input is presented (frame 3) and then stabilizes, suggesting a stabilization of the network activity. The training of the decoder was performed over 80 epochs,

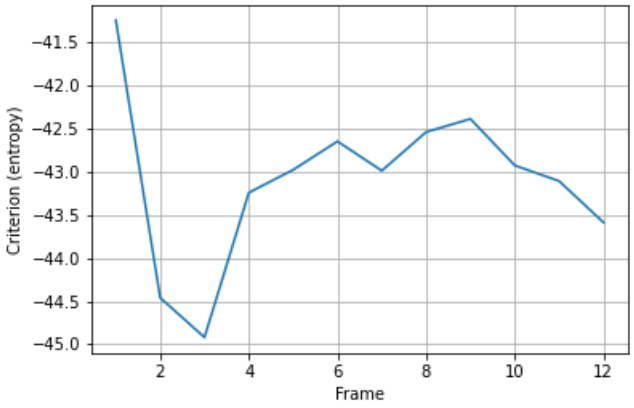

Fig. 6. Entropy of the distribution of the latent variable activities in function of the frame, for the training procedure described in the Results section. The 3 first frames are background frames. The first input is presented at the frame 3, which leads to a significant entropy increase.

with an initial learning rate of 1e-5. The decoder training loss and accuracy are shown in Figures 7 and 8.

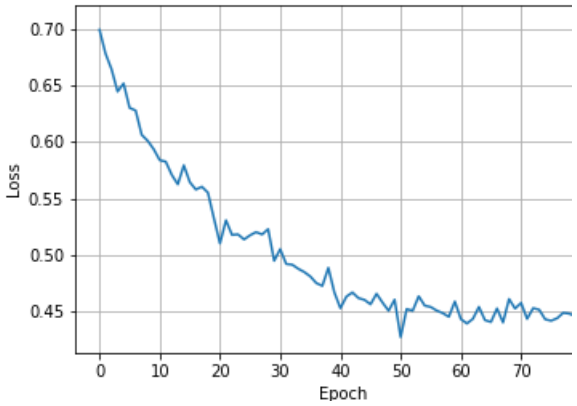

Fig. 7. Training loss of the decoder in function of the epoch, for the training procedure described in the Results section.

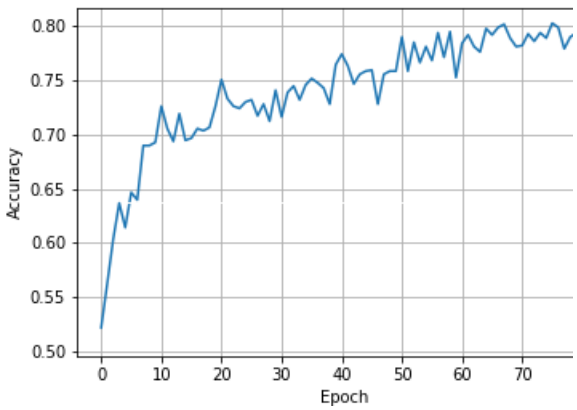

Fig. 8. Training accuracy of the decoder in function of the epoch, for the experiment described in the Results section.

The training loss curve displays a nice decrease, and the training accuracy increases well to reach very satisfying asymptotic values.

B. SQM results

The SQM test was performed over 15 subjects with 5 trials for each tested condition, until 8 relative frames between the central and the flanked vernier. The results are shown in Figure 9. Regarding this figure, it can be noticed that after a few frames, the vernier-antivernier condition seems to have roughly no effect over the prediction of the central vernier. On the other hand, the vernier-provernier condition seems to have an impact on the accuracy, in the expected central vernier direction. Nonetheless, the chronology is surprising. Indeed, the accuracies obtained for the relative frame 1, as well as for frames 5 and 6, are particularly strange regarding the behavior of the curve for the following frames, that may lead to believe an integration. In addition, when performed on more frames, the SQM did not enabled us to infer that any integration happened.

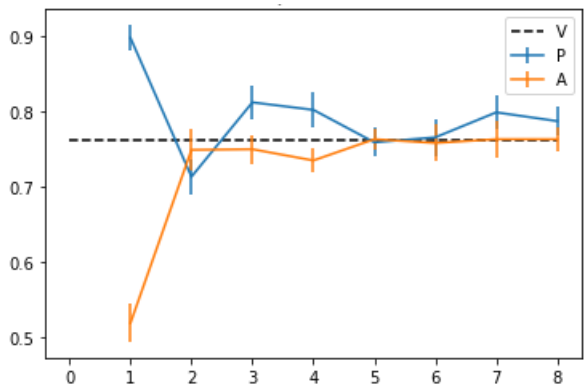

Fig. 9. Results of the SQM test performed on 15 subjects, 5 trials for each. The means over subjects and their standard errors are plotted against the frame number of the flanked vernier, relative to the central vernier frame number.

IV. Discussion and further improvements

While the obtained results were not satisfying, we can, however think of possible changes that may improve the current implementation.

First, some more complex datasets that better force the integration could be implemented, like growing shapes with varying viewing angles for example. Maybe, more complex forms of noise (Gaussian noise is quite easy to deconvolve) could also be added, forcing the model to rely more on its recurrences and latent representations.

Probably more determining for the final results is the criterion curve, which varied a lot between the batches. One may also argue that the choice and the design of any decoding criterion to model the discreteness ineluctably introduces a bias. In this way, we could implement and train a kind of ”discretizer” that learn by itself the ”readiness” state of the latent variables of the model, and returns the frames to send to the decoder.

An other thing that could reduce the specificity of the learned PredNet model for some given datasets, like the

problem encountered with the balls bouncing on walls when presenting verniers to the model, may be to introduce some regularization.

Finally, even if the intrinsic architecture of PredNet, with both top-down and bottom-up connections, seems to generate relevant internal representations and capture very well the computational behavior of this part of the visual processing, the latter seems to omit some physiological details that may play a role in feature integration. Indeed, when an input frame lasting some milliseconds is presented to the model, all top-down and bottom-up connections are stimulated and activated by this sensory input at this same moment in time. In contrast, in the human brain, the visual sensory inputs are continually stimulating the photoreceptor cells. If some connections are activated due to the current input, some other connections are either inactive or activated by other inputs, implying in addition some delays. Therefore, an idea is to test if the introduction of some delays and the selective activation of connections in the model gives a more realistic representation of the human-brain computations, and if this has an impact on the obtained results for the SQM test.

V. Summary

To summarize, this paper explains the approach that was adopted to model the SQM paradigm with recurrent neural networks, and to add some discreteness in the previous modelling. It was also the opportunity to revisit the model and improve some implementation details that did not work before. Finally, even if this project did not enable us to obtain the expected results on the SQM, this paper tries to give some ideas of improvements. Overall, this project was the ideal opportunity to learn more about the plethora of tools and implementation possibilities offered by TensorFlow, as well as about some current state-of-the-art practices in deep learning applications.

VI. Acknowledgements

I wish to thank Alban Bornet for his supervision and his precious help all along this project.

References

[1] N. Scheidwasser-Clow, “Modelling the sequential metacontrast paradigm with recurrent neural networks,” 2020.
[2] L. Drissi-Daoudi, A. Doerig, and M. H. Herzog, “Feature integration within discrete time windows,” 2019.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” 2012.
[4] A. Doerig, A. Bornet, O. Choung, and M. Herzog, “Crowding reveals fundamental differences in local vs. global processing in humans and machines,” 2020.
[5] W. Lotter, G. Kreiman, and D. Cox, “Deep predictive coding networks for video prediction and unsupervised learning,” 2017.
[6] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” 2015.