DNN Architecture for High Performance Prediction on Natural Videos Loses Submodule's Ability to Learn Discrete-World Dataset

Lana Sinapayen[1,*], Atsushi Noda[1,*]

[1]Sony Computer Science Laboratories, Tokyo, Japan. *These authors contributed equally to the paper.
lana.sinapayen@gmail.com

Abstract

Is cognition a collection of loosely connected functions tuned to different tasks, or is it more like a general learning algorithm? If such an hypothetical general algorithm did exist, tuned to our world, could it adapt seamlessly to a world with different laws of nature? We consider the theory that predictive coding is such a general rule, and falsify it for one specific neural architecture known for high-performance predictions on natural videos and replication of human visual illusions: PredNet. Our results show that PredNet's high performance generalizes without retraining on a completely different natural video dataset. Yet PredNet cannot be trained to reach even mediocre accuracy on an artificial video dataset created with the rules of the Game of Life (GoL). We also find that a submodule of PredNet, a Convolutional Neural Network trained alone, has excellent accuracy on the GoL while having mediocre accuracy on natural videos, showing that PredNet's architecture itself might be responsible for both the high performance on natural videos and the loss of performance on the GoL. Just as humans cannot predict the dynamics of the GoL, our results suggest that there could be a trade-off between high performance on sensory inputs with different sets of rules.

Introduction

In a world where many tasks have been automated to quasi-perfection, the next big goal for Artificial Intelligence (AI) is Artificial General Intelligence (AGI) (Wang and Goertzel (2007)). Beyond domain-specific automation, AGI is often defined as AI with human-level performance, able to generalize its knowledge across different domains (Adams et al. (2012); Wang and Goertzel (2007)). Candidate algorithms to realize AGI vary with the evolution of the field (Goertzel (2010, 2014)), and recent theories include a combination of Deep Learning Networks to emulate human cortical networks (Yamakawa et al. (2017)), neural computation at the edge of chaos (Smith (2016); Cocchi et al. (2017); Garson (1996)), and of course different implementations of predictive coding (Hawkins and Blakeslee (2007); Friston (2010); Van De Ven and Schouten (2010)).

Here we are interested in predictive coding: the idea that brains generate models of the world by learning to predict their own sensory inputs. Although the specific mechanism is up to debate (Friston (2009); Garalevicius (2007)), predictive coding has been found to take place in the human nervous system (Baldeweg (2006); Hosoya et al. (2005); Rao and Ballard (1999)), especially for visual and auditory processing. In many experiments, specific sensory illusions such as the auditory oddball effect are interpreted as hallmarks of predictive coding (Schindel et al. (2011)).

A recently proposed Deep Neural Network architecture, PredNet (Lotter et al. (2016)), has not only demonstrated high performance on natural video prediction, but has been shown to be susceptible to some of the same visual illusions as human beings (Watanabe et al. (2018)) as a side effect of its predictive abilities, despite contemporary explanations of these illusions not relying on predictive coding mechanisms. No one claims that PredNet accurately recreates the brain processes leading to predictive coding, but PredNed is currently the state of the art architecture replicating both the main consequences and side effects of predictive coding in human vision. Can PredNet be used as a general prediction machine, even in visual worlds that humans cannot predict, or is the algorithm's implementation so attuned to human visual processing that it fails in the same circumstances as humans do? In short, does high performance on natural datasets cause a loss in generality for this specific architecture?

To test this hypothesis, we choose a visual task where simple local spatial rules lead to dynamics that are hard to predict for human beings: the Game of Life automaton (GoL) (Gardener (1970), summarized by Izhikevich et al. (2015)). At each time step, the GoL updates each cell depending only on the state of its direct neighbors at the previous timestep. It is simple enough to be predicted with accuracy by a simple Convolutional Neural Network (Rapp (2015)). Most dynamics are discrete but some have the feeling of continuous dynamics, for example the "glider": a bird-like pattern that moves in a constant direction. The produced patterns can be hard to predict; even knowing the underlying rules, humans make mistakes in prediction. We can reasonably extrapolate that prediction without knowing these rules would be even harder for humans. The main dif-

ferences between a natural video such as those in the KITTI dataset (Geiger et al. (2013)) and a GoL-generated video are as follows:

- Theoretically, the GoL requires only 1 time step of memory for perfect prediction. Natural videos can show objects with speed or acceleration, which require several time steps to deduce.

- Natural videos have spatio-temporal continuity: objects maintain their shapes and there is no drastic change from one frame to the other. In the GoL, there is no continuity from one frame to the other and each cell is separate from the others. The GoL is a discrete world.

- Rules are extremely local in the GoL: a cell's next state only depends on its immediate neighbors.

If PredNet had succeeded at this task, it would suggest that the implementation of PredNet captures something fundamental beyond natural rules, and the idea of predictive coding as a general cross-domain learning rule would be strengthened. But our results show that PredNet is attuned to human performance even in its failure modes, as its sensitivity to visual illusions suggested. It comforts this architecture in its place as a replicator of human visual performance, but it also means that the improved performance on natural tasks comes to the cost of performance on tasks that can be solved by simpler networks. These results stack two different learning architectures (the human brain and PredNet) against predictive coding as a general cross-domain learning rule.

Methods

The source code used in this paper is available at `https://github.com/LanaSina/prednet_gol`. The datasets are available at `https://figshare.com/projects/PredNet_Game_of_Life/60971`.

PredNet

PredNet is a Deep Learning Neural Network with an architecture based on the principles of predictive coding (Fig. 1). The network is made of hierarchical layers. Each layer contains several modules, respectively for input, representation, prediction, and error calculation. The input to the first layer is the original image frame at time t; the error between the prediction at t and the frame at $t + 1$ is passed as input to the next layer, which must therefore learn to predict the error signal from the layer below it. The representations from the upper layer are sent as feedback to the lower layer.

For our experiments, we use the code provided by Lotter et al. (2016) at this address: `https://github.com/coxlab/prednet`. The network has 3 layers, its convolutional modules have filters of size 3×3 pixels. We tested
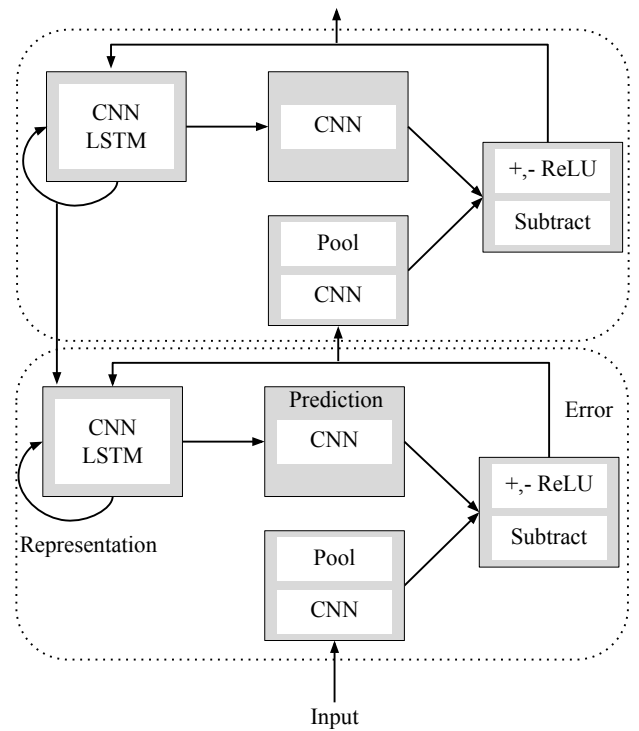

Figure 1: **The architecture of PredNet.** Figure modified from Lotter et al. (2016). The network is divided in modules (solid lines) that are stacked into layers (dashed lines). The input to the first layer is the original image frame at t; the error between the prediction at t and the frame at t+1 is passed as input to the next layer. There are 3 layers in total (only 2 represented here.)

generalization between datstets by evaluating the KITTI-trained model on other datasets. These experiments are referred to as experiments **without retraining** in the remainder of this paper. For the experiments **with retraining**, we train and evaluate PredNet on one given dataset (as is typical in Deep Learning experiments, training and testing data are separate but extracted from one same dataset).

Simple CNN

We use an Auto-Encoder Convolutional Neural Network, a minimum model that can learn the GoL rules (referred to as simple CNN in the remainder of this paper). The model is similar to the CNNs in several modules used in the PredNet architecture (notably the Prediction module) and uses the same filter size. We don't use a Long Short Term Memory (LSTM) since it is unnecessary to predict GoL images. The architecture is as follows: the encoder consists of one convolutional layer and maps pixels to a latent representation. The filter size is 3×3 and the stride is set at 1. Zero padding is done to deal with boundary conditions of the image. The decoder uses the transposed architecture. The acti-

vation function is ReLU. We implemented this model using Chainer (Tokui et al. (2015)) and trained it with the negative log-likelihood function of the Bernoulli distribution. In this model, the data at the episode boundaries is excluded from the training and evaluation data. The number of epochs × the size of training data is slightly less than one fifth of that of Prednet's KITTI learning.

Datasets

We use two existing datasets composed of videos from the real world (called "natural datasets" in this paper) and one dataset of videos generated from the GoL (called "artificial dataset").

- Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) Dataset The KITTI dataset is composed of videos filmed from a car driving on a road (Geiger et al. (2013)). The videos show traffic and pedestrians from the point of view of the driving car. We use the simple raw video part of the dataset (20 fps). As in the original PredNet paper, frames of the videos are downsampled to 128 × 160 pixels. We used the processing script for outputting frames from videos made available by (Lotter et al. (2016)). The script is also available in the repository published for this paper.

- First Person Social Interaction (FPSI) Dataset The FPSI dataset (Fathi et al. (2012)) is composed of videos filmed from a first-person point of view. The videos were shot by fixing cameras to the head of several people who spent a day at a theme park. The original dataset's url `http://cpl.cc.gatech.edu/projects/FPSI/` had become unresponsive at the time of the writing of this paper, so we used the video made available by Watanabe et al. (2018) at `https://figshare.com/articles/Training_data/5483668/1`. We also used their script for extracting and resizing images to 128x160 pixels, but found that the original video was encoded with the wrong FPS parameter. The corrected script is available at our published repository (output: 15 fps).

- Game of Life (GoL) Dataset We use Conway's Game of Life (GoL) (Gardener (1970); Izhikevich et al. (2015)) as artificial dataset to test PredNet. The GoL is a 2D 0-player game where an initial state, decided by the programmer, automatically changes every timestep according to a set of rules. The GoL is played on a grid where each cell is in one of two possible states: *alive* (white cells in our implementation) or *dead* (black cells). The flow of time is discrete, and the state transitions are deterministic and memoryless: each cell has 8 neighbors (4 cells adjacent orthogonally and 4 diagonally); the state the cell depends on how many neighbors are *alive*. If there are exactly 2 *alive* cells in the neighborhood, the state of the central cell does not change. If there are exactly 3 *alive* cells in the neighborhood, the cell is *alive*

in the next frame. For any other number of living cells in in the neighborhood, the next state of the cell is *dead*. These are the only rules. The interest of the GoL is that simple rules lead to complex patterns. We will briefly introduce the "glider" as an example used in experiments described later. The glider is a moving pattern composed of 5 cells. The movement of the glider has four stages; after 4 time steps, the glider has shifted one cell in the diagonal direction. In the GoL, there are patterns that move while keeping their shape like the glider, patterns that periodically repeat states, and so on.

We generate the video dataset as below. When generating an initial state, 10% of the cells are randomly set to *alive*; the other cells are *dead*. Each pixel represents a cell in the 128 × 160 images. There are no boundary conditions. The state often becomes fixed as time passes, so we generate a new initial state every 10 steps. We generate 10,000 images as training data.

Results

PredNet performance on KITTI, FPSI and GoL without retraining

We investigate how well the performance of PredNet trained exclusively on the KITTI dataset transfers to other datasets without retraining. We evaluate this model on the original KITTI dataset, on the natural video dataset FPSI, on random GoL patterns, and finally, on a GoL glider. In this experiment, the output of PredNet for the GoL was totally black frames, suggesting that the patterns were treated as noise. We therefore re-generated the GoL frames as 16×20 images and scaled them up to 8 times the original size.

As shown on Fig. 2-a), b), and Table 1, the performance transfers well between natural datasets. PredNet trained on the KITTI dataset still has better performance on the FPSI dataset than a simple "copy the last frame" model. This result is a testament to the robustness of the PredNet architecture. On the other hand, as shown on Fig. 2-c) and d), the performance does not transfer to datasets with artificial rules. As expected, the pretrained PredNet cannot predict the GoL, or even gliders. Even if gliders have a partly translational motion, the dynamics of the GoL are too far from the simple translations and rotations that might be responsible for the majority of variations in natural videos. In addition, we see an interesting error on frame 3 at Fig. 2-b): the translational motion of the cap that appeared at frame 2 is extrapolated, leading to the prediction of a floating cap at frame 3. This model cannot know that the cap should be on a human head, even if the person wearing the cap can be seen a few frames earlier. This would require either excellent use of short term memory, existence of an internal world model, or high level inference.
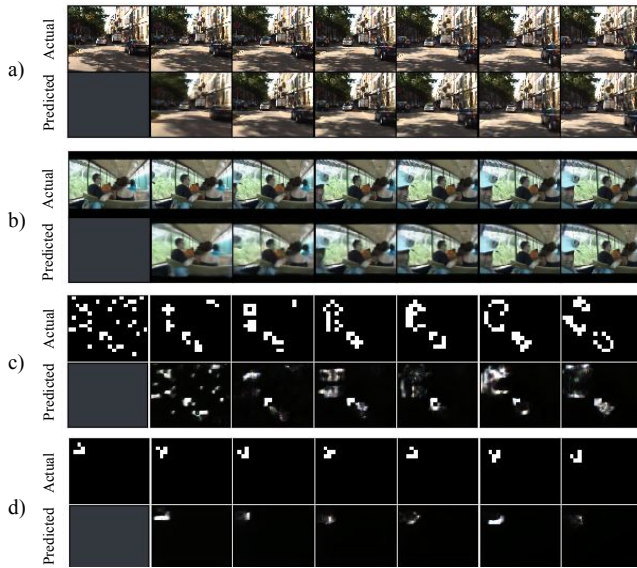
Figure 2: **Predictions generated by a PredNet trained on the KITTI dataset.** See Table 1 for measured performance. (a) Predictions for KITTI, the original training dataset. The performance is high. (b) Predictions for the FPSI dataset. The performance is also high. (c) Predictions for a random initial value of the GoL. The performance is poor, as expected. (d) Predictions for a GoL glider. The performance is also poor.

Retrained PredNet performance on the GoL

Since the performance of PredNet on natural images does not automatically transfer to the artificial dataset, in the first part of this experiment we re-train PredNet from scratch on the GoL dataset. Figures. 3 show zoomed-in results (clipped upper left images to make it easy to see). The network is unable to learn and performs poorly. This result is unexpected considering that this task is solved by a CNN as shown in the next section, and PredNet has several CNNs with the appropriate filter size of 3×3 in its architecture. Training on upscaled images gave the same result. This suggests that high performance on natural videos comes to the cost of generality for videos exhibiting different types of dynamics (even if these dynamics are totally deterministic and predictable).

and 4,

Finally, we train and evaluate PredNet on a single sequence showing a glider moving from the upper left to the lower right of the frame. In this setting we use the same data for learning and evaluation (it is therefore a pure memorization task). As shown on Fig. 4, even with these extremely gentle conditions, PredNet is unable to make any predictions. Training on upscaled images forced the network to have.
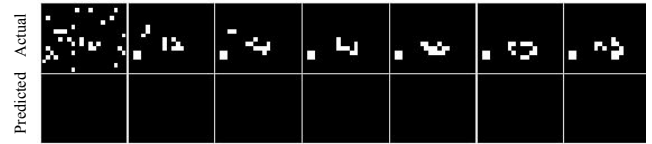

Figure 3: **Predictions generated by a PredNet trained and evaluated on random GoL patterns**. PredNet no longer outputs anything as predictions. The rules of the GoL seem not simply impossible to predict using PredNet, but impossible to learn.
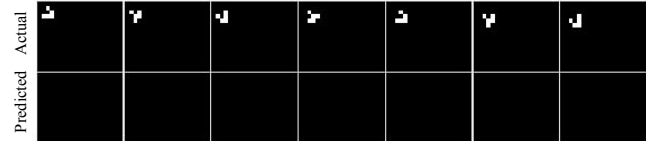

Figure 4: **Predictions generated by a PredNet trained only on the exact sequence shown here**. In this case, training data and test data are equal. Even for this simple task, PredNet no longer outputs anything as predictions.

Simple CNN performance on the GoL

We train the CNN on GoL videos generated from initial random patterns. We then test the CNN on two sequences: (a) a random pattern and (2) a glider pattern where a glider moves from the top left of the scene to the bottom right. As shown on Fig. 5, the CNN trained on the random patterns performs almost perfectly on both datasets, as previously reported in Rapp (2015). This result is not surprising considering that we fixed the surface of the convolution filters (receptive field of hidden neurons) to 3x3 cells: all the information necessary to predict the next step is contained into every individual filter. In addition, the rules are the same for the whole image, so that rules learned locally can be successfully applied anywhere in the image.

Simple CNN performance on KITTI

We train the simple CNN on the KITTI dataset. We do not expect good performance, as PredNet was implemented with the sole goal of achieving state of the art performance and simple CNNs are notoriously poor at natural video prediction. The results Fig. 6 show that indeed, the perfomance is poor. We stopped learning when the number of epochs was 10 because the value of loss did not fall any more. The output of the CNN resembles a fuzzy copy of the last presented frame. As recorded in Table 1, the simple CNN's error is an order of magnitude bigger than PredNet's error, and slightly lower than last frame copy. This result again shows a trade off, but opposite of PredNet: we have high performance on the artificial dataset and poor performance on the natural dataset.
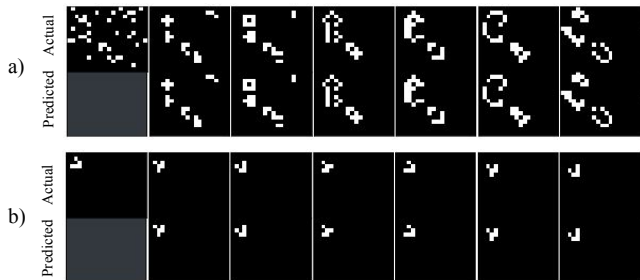
Figure 5: **Predictions generated by the CNN after training on random GoL patterns**. (a) Predictions from a random initial state. (b) Predictions for a initial state composed of one glider. The performance is close to perfect, an expected result considering that the convolutional filters are the same size as the local neighborhood of the GoL cells.

Figure 6: **Predictions generated by the CNN after training on the KITTI dataset**. The performance is poor, especially compared to PredNet (an expected result for natural videos).

Prednet-as-CNN: performance on the GoL

The goal of this experiment is to see if PredNet's unexpected inability to predict discrete data might be due to implementation details (parameters or bugs), or if the architecture itself might be responsible for the inhibition of the ability of the inner modules to predict discrete data. We fix all the weights to their initial random values (i.e., we prevent learning on these weights by setting the learning rate to 0) except for one of the CNN modules. The resulting model, "Prednet-as-CNN", is therefore structurally equivalent to a simple CNN: if it fails to learn GoL predictions, it will indicate that implementation details are to blame. If Prednet-as-CNN learns the GoL, it indicates that the failure of the original PredNet model is due to its architecture. The training data is the same as for the simple CNN experiment: random GoL patterns.

The results, shown on Fig. 7, show two noteworthy elements. First, the predicted cells are pink instead of white. This most likely suggests either an issue with the code, or less likely, an issue with the architecture introducing a random bias towards low loss on the red channel of the rgb input. Second, the network does output some predictions, not simply black frames. The performance indicates that the predictions are better than simple copies of the previous frame (Table 1), but to the naked eye they do mostly look like copies of the previous step. The results are the

Table 1: Mean squared error (MSE) of next frame predictions. As mentioned in the Dataset's subsection, all datasets are set to the same pixel size, so it is meaningful to compare values side by side.

Dataset	Method	MSE
KITTI	Previous Frame	0.0256
	PredNet	0.0073
	Simple CNN	0.0164
FPSI	Previous Frame	0.0110
	PredNet (pretrained)	0.0059
GoL (random)	Previous Frame	0.0332
	PredNet (pretrained)	0.0330
	PredNet (re-trained)	0.0348
	PredNet-as-CNN	0.0265
	Simple CNN	0.0002

same for 150 (original parameter), 250, and 500 epochs, suggesting that no learning takes place beyond the pink-copy phase. This result is difficult to interpret, and additional experiments are needed; but this half failure (compared to the simple CNN), half success (compared to the unmodified PredNet and its blank predictions) suggests that at least part of the blame goes to the architecture itself.
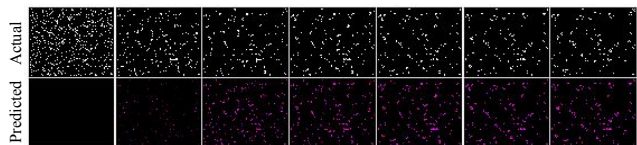

Figure 7: **Random GoL pattern prediction by the Prednet-as-CNN model**. The predicted cells are pink instead of white, suggesting an issue with the code, or an issue with the architecture introducing a random bias towards low loss on the red channel of the rgb input. Contrarily to PredNet, the network does output predictions and not simply black frames.

Discussion

We showed that PredNet can generalize without retraining on a completely different natural video dataset. Yet we also found that unmodifed, PredNet cannot learn to predict the dynamics of the GoL. Worse yet, it cannot memorize a short 10-frame sequence of the GoL: a task that should be solvable rote memorization of the mapping between input and output. This result is surprising since a simple CNN with only one conv-deconv layer did learn the GoL rules, and PredNet was not able to learn despite internally containing modules that are themselves simple CNNs. When making PredNet structurally equivalent to a simple CNN ("Prednet-as-CNN"), the model improved its GoL predictions, albeit

still showing poor performance. This suggests that the architecture of PredNet, which makes it so good with continuous input, is also a cause of the inhibition of its submodules ability to predict discrete input.

This paper shows that there is at least one unsuitable task for the PredNet architecture as it is. In this models, past observations are abstracted and stored internally, and it seems that input images are shifted by an appropriate amount using estimated velocity and rotation information. This is especially striking in the prediction error on frame 3 in Fig. 2-b): the translational motion of the cap that appeared at frame 2 is extrapolated, leading to the prediction of a floating cap at frame 3. This explains why these models are good for continuous dynamics such as movement of objects in the real world. On the other hand, in the GoL world, every cell changes in a discrete manner according to the GoL rules. This can be captured by CNNs, but this ability seems to be somehow inhibited by the very architecture that makes PredNet good at shifting and rotating. To find out the cause of the difference in performance, we still need to test hybrid datasets that are missing some characteristics of real world video.

Note that we just assumed that PredNet, as current state of the art in video prediction, was a kind of "master algorithm" for natural videos; in reality its failure does not mean that all high-performance predictive algorithms would fail at predicting discrete datasets. Yet we have to wonder if a single architecture would be able to perform on both natural and artificial datasets, or if the trade-off in performance cannot be avoided. Supporting this latter hypothesis, the GoL is extremely difficult to predict for human brains; even its own creator, knowing the rules, had to use a physical board with go stones to compute states and still made errors. If there really is a divergence between optimizing predictions for natural and synthetic datasets, then when thinking about an hypothetical general artificial intelligence, we should also consider how to compensate for the counter-intuitive weaknesses that such superior models create for themselves compared to lower-performance models. More generally, it suggests that "prediction" does not have the same meaning in natural and artificial worlds.

References

Adams, S., Arel, I., Bach, J., Coop, R., Furlan, R., Goertzel, B., Hall, J. S., Samsonovich, A., Scheutz, M., Schlesinger, M., et al. (2012). Mapping the landscape of human-level artificial general intelligence. *AI magazine*, 33(1):25–42.

Baldeweg, T. (2006). Repetition effects to sounds: evidence for predictive coding in the auditory system. *Trends in cognitive sciences*.

Cocchi, L., Gollo, L. L., Zalesky, A., and Breakspear, M. (2017). Criticality in the brain: A synthesis of neurobiology, models and cognition. *Progress in neurobiology*, 158:132–152.

Fathi, A., Hodgins, J. K., and Rehg, J. M. (2012). Social interactions: A first-person perspective. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1226–1233. IEEE.

Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13(7):293–301.

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127.

Garalevicius, S. J. (2007). Memory-prediction framework for pattern recognition: Performance and suitability of the bayesian model of visual cortex. In *FLAIRS Conference*, pages 92–97.

Gardener, M. (1970). Mathematical games: The fantastic combinations of john conway's new solitaire game" life,". *Scientific American*, 223:120–123.

Garson, J. W. (1996). Cognition poised at the edge of chaos: A complex alternative to a symbolic mind. *Philosophical Psychology*, 9(3):301–322.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.

Goertzel, B. (2010). Toward a formal characterization of real-world general intelligence. In *3d Conference on Artificial General Intelligence (AGI-2010)*. Atlantis Press.

Goertzel, B. (2014). Artificial general intelligence: concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, 5(1):1–48.

Hawkins, J. and Blakeslee, S. (2007). *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan.

Hosoya, T., Baccus, S. A., and Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, 436(7047):71.

Izhikevich, E. M., Conway, J. H., and Seth, A. (2015). Game of Life. *Scholarpedia*, 10(6):1816. revision #150735.

Lotter, W., Kreiman, G., and Cox, D. (2016). Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning.

Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79.

Rapp, D. (2015). Learning game of life with a convolutional neural network. https://danielrapp.github.io/cnn-gol/. Accessed: 2019-02-28.

Schindel, R., Rowlands, J., and Arnold, D. H. (2011). The oddball effect: Perceived duration and predictive coding. *Journal of Vision*, 11(2):17–17.

Smith, L. S. (2016). Deep neural networks: the only show in town? a position paper for the workshop on can deep neural networks (dnns) provide the basis for artificial general intelligence (agi) at agi 2016, july 2016.

Tokui, S., Oono, K., and Hido, S. (2015). Chainer: a next-generation open source framework for deep learning.

Van De Ven, A. and Schouten, B. A. (2010). A minimum relative entropy principle for agi. In *3d Conference on Artificial General Intelligence (AGI-2010)*. Atlantis Press.

Wang, P. and Goertzel, B. (2007). Introduction: Aspects of artificial general intelligence. In *Proceedings of the 2007 conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006*, pages 1–16. IOS Press.

Watanabe, E., Kitaoka, A., Sakamoto, K., Yasugi, M., and Tanaka, K. (2018). Illusory motion reproduced by deep neural networks trained for prediction. *Frontiers in psychology*, 9:345.

Yamakawa, H., Arakawa, N., and Takahashi, K. (2017). Reinterpreting the cortical circuit. In *Architectures for Generality & Autonomy Workshop at IJCAI*, volume 17.

Additional Details

Retrained PredNet performance on the upscaled glider

As in the Section "Retrained PredNet performance on the GoL", we also upscaled the glider images and trained PredNet to memorize them. Fig. 8 shows that PredNet cannot predict perfectly even with these extremely gentle conditions. What it seems to have learned is a set of periodic translations of period 5 for simple patterns.

Simple CNN with smaller filters

When using 2×2 convolutional filters instead of 3×3 filters, the performance of the simple CNN model sharply declined as shown in Fig. 9. These filters are too small to represent the GoL rules: in the GoL, the next state of a cell is completely determined by the state of eight neighboring cells. Necessary information is contained in a 3×3 neighborhood.

Reversed video prediction

Although physical laws are time reversible, it is famously difficult for humans to predict videos where the flow of time is reversed. We tested whether a PredNet trained on regular-time videos from the KITTI dataset would be able to predict reverse-time videos. The model turned out to be just as good for regular- and reverse-time videos, with a MSE of 0.006 in both cases. Fig 10 shows the generated predictions.
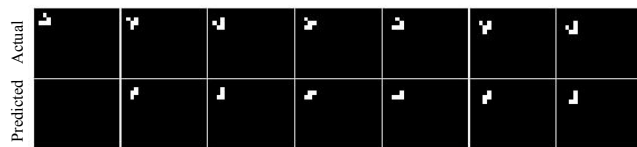

Figure 8: **Predictions generated by a PredNet trained only on the exact (upscaled glider) sequence shown here**. In this case, training data and test data are equal. Even for this simple task, PredNet seems to only be able to imperfectly predict the pattern, with a constant one-cell mistake. In reality, it seems that what was learned was a simple translation of elementary patterns of period 5.
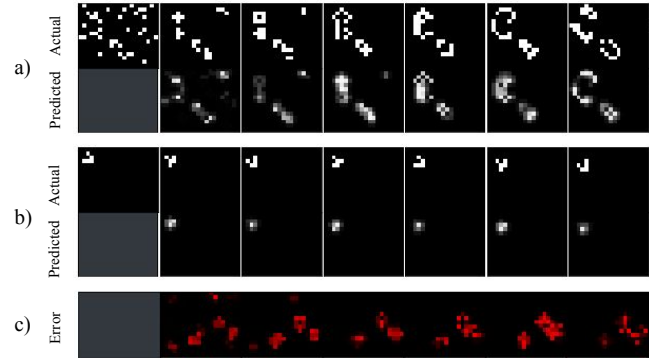

Figure 9: **Predictions with smaller convolutional filters.** The settings are the same as for Fig. 5, but the filter size is 2×2 instead of 3×3. (a) Predictions from a random initial state. (b) Predictions for a initial state composed of one glider. (c) The difference between actual and predicted images. The performance is poor. The filters are too small to capture neighborhood information.

Figure 10: **Predictions generated on the FPSI dataset**. (a) Predictions on the regular-time video. (b) Predictions on the reverse-time video. The predictions are not exactly the same but the performance is similar in both cases.