The role of ambient noise in the evolution of robust mental representations in cognitive systems

Douglas Kirkpatrick[1,2,4], and Arend Hintze[3,1,2,4]
[1] Department of Computer Science and Engineering
[2] BEACON Center for the Study of Evolution in Action
[3] Department of Integrative Biology [4] Michigan State University
hintze@msu.edu

Abstract

Natural environments are full of ambient noise; nevertheless, natural cognitive systems deal greatly with uncertainty but also have ways to suppress or ignore noise unrelated to the task at hand. For most intelligent tasks, experiences and observations have to be committed to memory and these representations of reality inform future decisions. We know that deep learned artificial neural networks (ANNs) often struggle with the formation of representations. This struggle may be due to the ANN's fully interconnected, layered architecture. This forces information to be propagated over the entire system, which is different from natural brains that instead have sparsely distributed representations. Here we show how ambient noise causes neural substrates such as recurrent ANNs and long short-term memory neural networks to evolve more representations in order to function in these noisy environments, which also greatly improves their functionality. However, these systems also tend to further smear their representations over their internal states making them more vulnerable to internal noise. We also show that Markov Brains (MBs) are mostly unaffected by ambient noise, and their representations remain sparsely distributed (i.e. not smeared). This suggests that ambient noise helps to increase the amount of representations formed in neural networks, but also requires us to find additional solutions to prevent smearing of said representations.

Introduction

In all but the simplest of forms, artificial and natural minds need to have experiences, remember them, and use those memories to inform future actions. Previous work has defined the term "representations" to be the information an agent has about the environment that is not present in its sensors (Marstaller et al., 2013). We developed a measure to quantify these mental representations, referred to as R. Further work expanded on this method of quantifying representations and showed that R can even be used to augment a genetic algorithm's (GA's) performance to find better solutions in a shorter amount of time than an unaugmented GA (Schossau et al., 2015). In general, the amount of representation increases over the course of evolution and allows neural substrates [1] to deal better with their environments – clearly, knowing something about the world they live in is beneficial.

By measuring which components of a cognitive agent have information about specific aspects of the world, we can even pinpoint where representations are stored (Marstaller et al., 2013). Specifically, the measure of R (see Figure 2) is applied to every node and every concept separately, resulting in a matrix (M) of measurements. This method further allows us to determine how distributed (smeared) or localized these representations are (Hintze et al., 2018) (see Equation 2 for a quantification of the smearedness of matrix M). We also found that less smeared representations are more robust against sensor noise.

All these findings support the notion that representations are crucial for intelligence, and we can use evolutionary processes to create cognitive agents that take advantage of them. We also know that in deep learning of artificial neural networks noise is often used to increase the size or quality of the training data set (Brown et al., 2003; Lauzon, 2012), but it also directly can improve training when applied for example to the gradient descent back propagation (Neelakantan et al., 2015). Similarly, *dropout* is another method to improve the deep learning process due to randomness (Srivastava et al., 2014). However, noise within the input training set has also been identified to limit the performance of ANNs (Zhu and Wu, 2004) as it makes class discrimination difficult. In the context of these conclusions, the question that we seek to investigate is how we can improve the amount and quality of representations that a cognitive system has about its environment and what the role of sensor (ambient) noise plays in the evolutionary adaptive process.

In order to test these questions, we evolve different cognitive systems in the presence of different levels and forms of sensor noise, and study their performance, their ability to form representations, and their robustness to different types of noise. From previous experiments, we know that different systems behave very differently during evolutionary adaptation. Markov Brains (Marstaller et al., 2013; Hintze et al., 2017), for example, first have to evolve structures to retain information before they can evolve to take advantage

[1] e.g. MBs or ANNs, broadly referred to as brains or agents

of them. Recurrent neural networks on the other hand retain information due to their connectedness and need to only evolve to use the information. At the same time, we know that Markov Brains have distinctively discrete representations, while neural networks tend to smear these representations across their recurrent nodes. As a consequence, recurrent neural networks in the long run lose their early evolutionary advantage of getting representations for free when compared to Markov Brains that need to also evolve mechanisms to retain information (Hintze et al., 2018). Since we also have not yet studied how representations form in other cognitive systems, here we take the opportunity and include recurrent neural networks that can evolve their topology and compare them to Markov Brains using deterministic logic gates, recurrent neural networks that use a fixed topology, and long-short term memory neural networks (LSTMs) (Hochreiter and Schmidhuber, 1997).

To study these questions of robustness and representations, the different cognitive systems are tasked to discriminate numbers (Nieder, 2018). Specifically, two different numbers are presented in sequence, and the cognitive system has to identify which of the two numbers is larger (the first or the second). This computation, however, takes the cognitive systems some time, and in this time the systems are additionally fed either noisy or static inputs. These systems cannot just avoid those noisy inputs (for example by looking away), but instead need to evolve some cognitive mechanism to filter out the noise, since the noise is always present in their sensors.

We will show that cognitive systems tend to increase the amount of representations that they have as they encounter sensor noise. They not only increase the amount of representations, but also smear those representations across their neural substrate. While being a proper response to sensor noise, it also makes them more vulnerable to internal noise of, for example, faulty components.

Material and Methods

Number Discrimination Task

For this work we adapted the Number Discrimination Task (NDT), used in biological (Nieder, 2018; Merritt et al., 2009) and psychological (Merritt and Brannon, 2013) experiments, for use *in silico*. In this task, participants have to discriminate two different amounts of items. For bees (Nieder, 2018) there were landing sites associated with different numbers of squares, and the bees were rewarded when they landed on the site associated with the correct value. For rhesus monkeys (Merritt et al., 2009) and humans (Merritt and Brannon, 2013), the reward is given when the person or monkey selects the correct value from two collections of different colored shapes presented on a screen. The basic principle *in silico* remains the same in that the agent must first look at one quantity (the number of '1' signals in the input), then look at a second quantity (a second, different number of '1' signals in a second input), and finally make a determination of which quantity was larger or smaller.

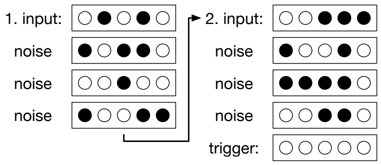

Figure 1: Example graphic of the input states for the number discrimination task. Each agent received a sequence of inputs, illustrated as black and white circles. The first and second input are used to determine the numbers to discriminate (here, 2 and 3, respectively), while the other intermittent inputs are ambient noise. Once the trigger is shown, the agent's answer about which number is higher is evaluated.

For our experiments, the agents are given an input vector consisting of x '1' signals and y '0' signals, where $x + y$ is equal to 5, and are given 3 updates to process that number. Then they are given a second input vector consisting of a different amount of j 1s and k 0s, where $x \neq j$ and $j + k$ is also equal to 5. After a second set of 3 updates to process the second number, they are given a final input of 5 '0's, after which they must indicate if x was greater than j. During the thinking period, the brains are either given a constant string of '0's as an input vector, or with probability p each of the values in the input vector is set to a random value. The pattern of input vectors is illustrated in Figure 1.

In natural systems subjects are only tested on a handful of questions. Here we need to properly assess an agent's ability to discriminate as part of the evolutionary process, and thus we test them on all possible input scenarios. For each number n between 0 and 5 inclusive, all permutations of n 1s and $5-n$ 0s are tested against all permutations of the other values in $[0, 5]$. As an example, 01001 and 11000 are both valid permutations for $n = 2$, and all such permutations of $n = 2$ would be tested for all permutations of the amounts 0, 1, 3, 4, and 5. These are additionally tested with both the larger amount presented first and the smaller number second, and vice-versa. That is, 01011 as the first amount then 11111 as the second amount is tested as well as 11111 then 01011.

Evolutionary Algorithm The fitness for this task is determined by the number of times that the agent correctly identifies the larger amount (C) versus the number of times that the agent does not identify the larger amount correctly (I). The agent's fitness is multiplied by 1.1 for every correct identification and divided by 1.1 for every incorrect identification. This results in a fitness function as shown in Equation 1.

$$W = 1.10^{(C-I)} \tag{1}$$

Once the fitness was calculated individually for the entire population, we used roulette wheel selection, which implements a fitness proportional Moran process (Moran, 1958), to determine which organisms reproduced and formed the next generation. At every reproductive event, mutations (see the following subsection for details) are applied to the offspring. The agents were evolved for 40,000 generations in populations of 100 organisms, all starting with different random genomes.

Shared Genome

Each of the brains used in this experiment was generated from the same type of genome. That genome type is a circular genome with an initial size of 5,000 sites, a maximum size of 20,000 sites, and a minimum size of 2,000 sites. A number of mutational operators were used to mutate the genomes. These were a point mutation operator that had a probability of 0.005 of changing each site, a copy insertion operator that had a per site probability of 0.00002 of copying a section of the genome with a size between 128 and 512 sites and inserting it into the genome at a randomly chosen site, and a deletion operator that had a per site probability of 0.00002 of deleting a section of the genome between 128 and 512 sites long. Brains read these genomes sequentially in order to determine their weights, computational elements, or topology when required.

Markov Brains

MBs take the form of a compact network of computational elements (called gates) that read from and write to a series of nodes. The nodes are divided up into three classes, input, output, and hidden. Input nodes take the input from the task, output nodes are used by the task to determine the agent's actions, and hidden nodes are used by the brain to store information. All of the MBs used in our experiments had 8 hidden nodes for memory. For our experiments we used two types of MBs, each using a single type of gate - Deterministic Logic or ANN. For a more detailed description see Hintze et al. (2017).

Deterministic Gates Deterministic Logic gates, as used in the experiments carried out here, read from between 1 and 4 inputs, perform a logic operation (e.g. boolean AND, boolean XOR) and write the results to between 1 and 4 outputs. If two gates write into the same node, their outputs experience a logical OR operation.

ANN Gates The ANN gates used here simulate a single-layer version of a simple feed-forward ANN (Russell et al., 2003). They read from between 1 and 4 inputs, multiply the inputs by weights stored in a table, and take the $tanh$ of the sum of the products to write to between 1 and 4 outputs. In case the outputs of these gates write into the same node, their outputs are added together.

The construction of Markov Brains with ANN gates technically allows for arbitrary topologies in the neural network, and as such breaks the paradigm that ANNs must be organized in layers. The Markov Brain/ANN hybrids are much closer technically to RNNs or LSTMs but with a changing topology.

Recurrent ANNs

While the ANNs used in the ANN Gates are simple feed-forward mechanisms, more complex versions of ANNs are possible (e.g. Hochreiter and Schmidhuber (1997); Lauzon (2012)). As the presence of memory is necessary to examine representation, we elected to use an ANN augmented with recurrent nodes (i.e. memory) that was highlighted in previous work on representations (Hintze et al., 2018), referred to as an RNN. The structure of the RNN is similar to a standard ANN with one key difference. Like ANNs, the RNN is multilayer, feed-forward, and the layer to layer update is $tanh$ of the sum of weights times the previous layer values. The difference between ANNs and RNNs is that in RNNs the input and output layers have additional recurrent nodes that are copied from the output layer to the input layer at the end of each update. These recurrent nodes allow the RNN brains to have memory and store information about the environment. Each of the RNNs used in our experiments had 8 recurrent nodes, to allow a direct comparison to the 8 hidden nodes in the Markov Brains. The number of input and output nodes is defined by the task.

LSTM Networks

As opposed to the simple recurrence of RNNs, LSTM networks (Hochreiter and Schmidhuber, 1997) implement a more sophisticated memory model. LSTMs have been shown to perform well on tasks that require memory using Deep Learning (Schmidhuber, 2015), and thus seem ideal for studying representations. These networks have two streams of memory, (C and h) intended to replicate long and short term memory, as opposed to the simple recurring nodes of RNNs. LSTMs also use a more complicated set of equations to calculate the update values of the streams and output nodes. We modified the LSTMs to have 8 hidden nodes by expanding the memory streams, as described in previous literature (Hintze et al., 2018). The number of input and output nodes is again defined by the task.

Representations and R

Representations are the information stored in brain about environment not present in the sensors. An information-theoretic measure, R, was developed to measure representations (Marstaller et al., 2013), and is visualized in Figure 2. For this task, we defined the world states to be binary states for each number that are true if the agent is seeing that number as part of the current trial, and a binary state that is true if the first value is greater than the second. This information - which numbers the agent is seeing and the relationship between the numbers - are what we deem necessary for the agent to solve the task.

As R is an information-theoretic measure, it is susceptible to the presence of noise (i.e. the presence of noise will affect the measurement of R). To ensure that the values of R would be comparable across different levels of ambient noise, we discarded the observations of World, Sensor, and Brain states when ambient noise was being applied to the sensor states.

We made one modification to the calculation of R, in how we dealt with the continuous hidden node values in MBs with ANN gates, RNNs, and LSTMs. To perform the entropy calculations, we used median discretization to first transform the hidden node values to bit values. This median discretization ensures that individual node states have maximum entropy to avoid accidentally introducing other sources of information into the calculation erroneously. With all brains having the same number and type of hidden states, we then calculated R.

Smearedness of Representations

While the measurement of representations is important, R measures nothing about the structure of those representations - how the representations are stored in the hidden nodes. Recent work (Hintze et al., 2018) identified a new measure regarding representations, the smearedness of representations. This measure records how much the representations are distributed across the hidden nodes of the brain. That paper had suggested two different types of smearedness - the smearedness of concepts across nodes (S_C) and the smearedness of nodes across concepts (S_N). We found a strong positive correlation between S_C and S_N (data not shown), so for the purposes of this paper we will only look at S_N.

To determine the smearedness of the representations, one must calculate atomic R - information stored about one aspect of the environment in one hidden state not including the information from the sensor. With the atomic R M_{ji} for every environmental concept j and hidden node i, we can then use Equation 2 to calculate S_N. To handle the continuous hidden node values in MBs with ANN gates, RNNs, and LSTMs, we used the same discretization process used in the computation of R (see the above section for details).

$$S_N = \sum_{i} \sum_{j>k} min(M_{ji}, M_{ki}) \tag{2}$$

The approach used here measures the representations each node has about each concept separately. However, sets of nodes can share (encrypt) information about concepts, and *vice-versa*. One could compute all information between all subsets, which would allow for a more precise way to identify how information is smeared across nodes, and about which concepts such information is smeared. The complexity for a system of n nodes and c concepts is unfortunately 2^{n+c} which is technically impractical.

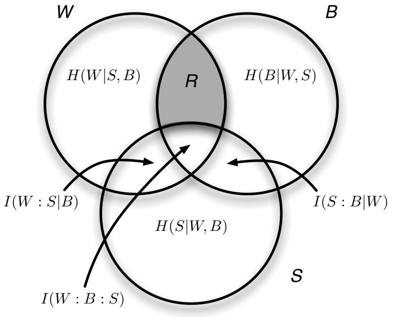

Figure 2: Venn diagram of entropies and informations for the three random variables W (world), S (sensor), and B (brain) describing all possible states the system can be in. The representation $R = H(W{:}B|S)$ is shaded.

Robustness to Internal Noise

In order to determine if the agents are forming meaningful representations (i.e. representations that are useful to performance) we introduce a new measurement, referred to as robustness. This measure is designed to have a higher value when the agent is more robust to noise in its hidden states and a lower value when the agent is less robust to noise in its hidden states. Robustness to internal noise is designed to be an approximate measure of how well an agent could handle faulty memory or internal states. To calculate this robustness, we test each agent on the task while applying noise to the internal states (e.g. the hidden states in Markov Brains or the recurrent nodes in RNNs). The noise is quantified by a probability p, where p is the chance that for each hidden state that the agent has, on each update of the agent that state will be set to a random value between -1 and 1. We test the agents with p at a range of values P between 0 and 1, and find the percentage of the trials that the agents get correct, defining the fitness function. The robustness is the sum of the percentages over all values of p (Equation 3).

$$Robustness = \sum_{p \in P} PercentCorrect_p \tag{3}$$

For our experiments, $P =$ [0.0, 0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.15, 0.2, 0.3, 0.4, 0.5, 1.0].

Experimental Setup

For each of the 4 brain types outlined above, we evolved populations on the Number Discrimination Task with varying levels of ambient noise. The noise levels tested (i.e. the probability that each sensor input becomes randomized instead of being kept at 0) were 0.0, 0.25, 0.50, 0.75, and 1.0. We ran 400 independent evolutionary experiments for each brain and noise level condition, each with a different random starting condition. At the end of evolution, we analyzed the brains along the line of descent (Lenski et al., 2003), measuring their performance, R, the smearedness of nodes, and the robustness to internal noise.

Results

As we are dealing with a different environment than previously studied in the context of representations and R (Marstaller et al., 2013), we must first establish that the agents can evolve to perform and develop representations in this environment. We find that regardless of brain type (Markov Brain logic gates, Markov Brain ANN gates, RNN, or LSTM) and regardless of noise level (0, 0.25, 0.5, 0.75, 1.0), all brains generally evolved to perform the task (see Figure 3 left column). By the end of evolution, around 11% of the MB agents evolved with deterministic logic gates, 3.5% of the MB agents evolved with ANN gates, 35% of the RNN agents, and 4% of the LSTM agents had reached perfect performance, averaged across all noise conditions. However, it seems as if there is a specific difference between MBs (logic or ANN gates) and the two other types (RNN and LSTM) in the way evolutionary adaptation is affected by noise. While MBs experience a drop in performance when noise is introduced (0.25), they become better adapted the noisier the environments are (see Figure 3 right column). RNNs and LTSMs evolve immediately better with low noise (0.25) and struggle to become optimal the more noise they encounter.

The degree to which the different types of brains evolve representations on the other hand is affected differently by noise. The total information that MBs evolve to have about their environment seems to be unaffected by noise, while all other systems evolve to store more information the more noise they encounter (see Figure 4). As previously observed (Marstaller et al., 2013) MBs using deterministic logic gates end up having the largest amount of R initially, which might explain that additional sensor noise does not encourage them to evolve even more R. The other types of brains instead evolve to have more representations given more noise. We speculate that MBs with ANN gates, RNNs, and LSTMs need to do that in order to compensate for the noise. Intuitively, this is supported by the idea that one needs to remember better if one's sensors are flooded with noise. While this is a conjecture at this point, we will further investigate this question, but we need to first ask to what degree the representations that these systems evolve are smeared or distinct.

To measure the smearedness of representations, one first identifies which component of the brain has how much information about the concepts of the world. This is comparable with the idea of "grandmother neurons", where specific neurons represent specific memories, here of the grandmother (for a more critical discussion see Quiroga et al. (2013)). Smearedness does not assume the existence of grandmother neurons, but only quantifies the degree to which representations are either stored in individual nodes (or neurons) or distributed over many. The less smeared R is the sparser and more distinct representations are. We find that, except for MBs using deterministic logic gates, all other brain types evolve to have more smeared representations the more noise they encounter (see Figure 5).

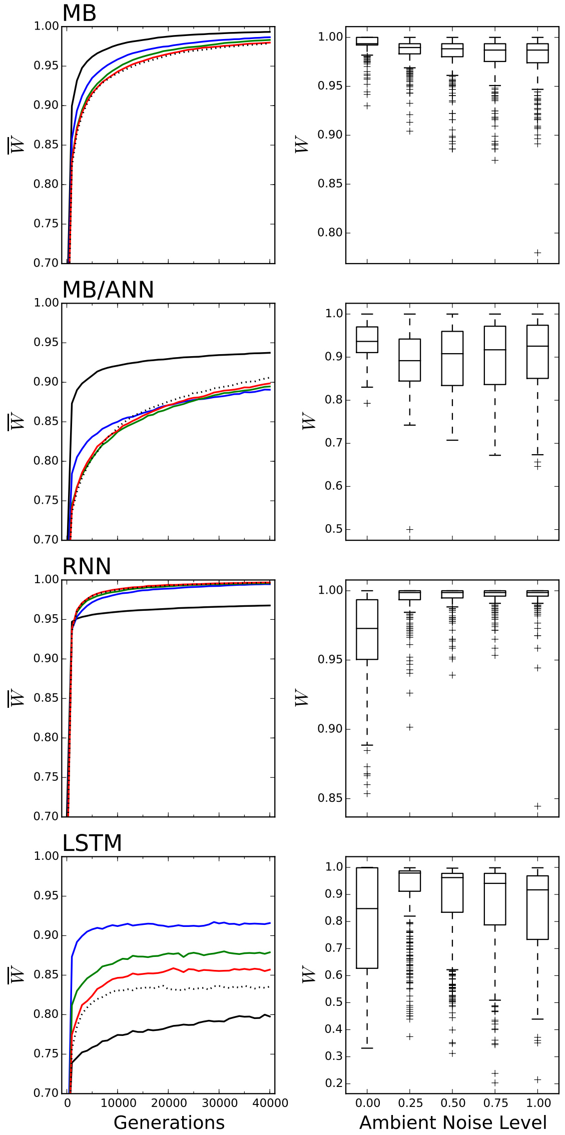

Figure 3: **Performance Over Time**. For each type of brain (MB, MB using ANN gates, RNN, and LSTM see labels) the average performance on the line of descent for all replicate experiments (left column). The black solid line is 0.0 noise, the blue line is 0.25 noise, the green line is 0.5 noise, the red line is 0.75 noise, and the black dotted line is 1.0 noise. The right column shows the distribution of performances including the 25% and 75% confidence intervals and their outliers for all brains under all experimental noise conditions.

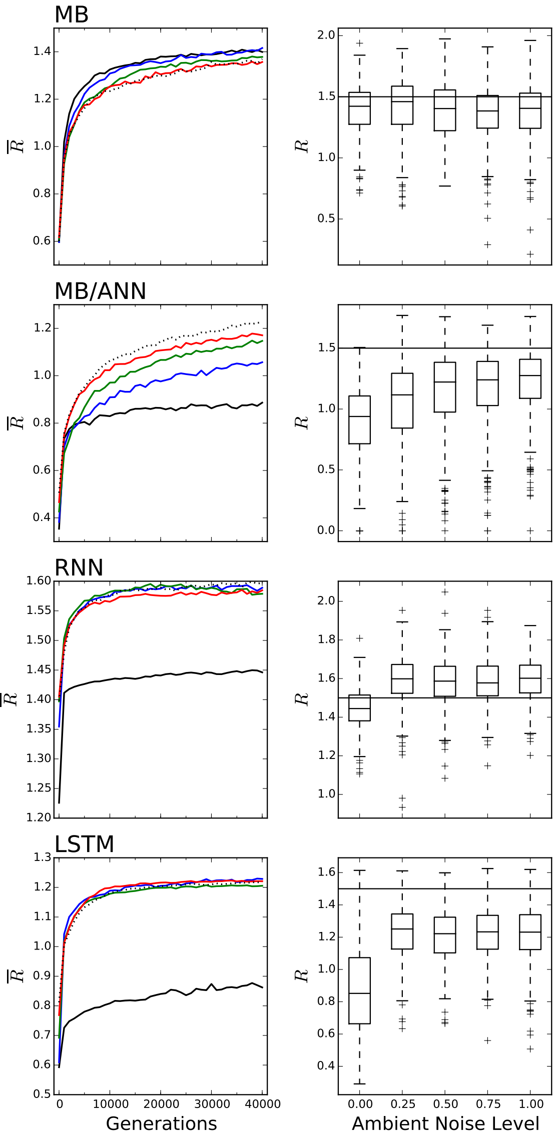

Figure 4: **R Over Time**. For each type of brain the average value of R on the line of descent for all replicate experiments (left column). The black solid line is 0.0 noise, the blue line is 0.25 noise, the green line is 0.50 noise, the red line is 0.75 noise, and the black dotted line is 1.0 noise. The right column shows the distribution of performances including the 25% and 75% confidence intervals and their outliers for all brains under all experimental noise conditions. There is a line in every subgraph in the right column at the height 1.5 to allow for a comparison between brain types.

We further find, that the degree of this smearing is correlated to the amount of R each agent evolved (see Figure 6). Those agents that evolve to have a lot of information about the environment, also smear these representations.

As mentioned before, we assume that brains evolve to have more representations as a functional response to more noise in the environment, and that the smearing of these representations over their hidden states is functional as well. If this is the case, the representations that get smeared need to contribute to performance, otherwise it could be coincidental or a measurement artifact. To test this, we determine the robustness to internal noise and correlate this robustness to

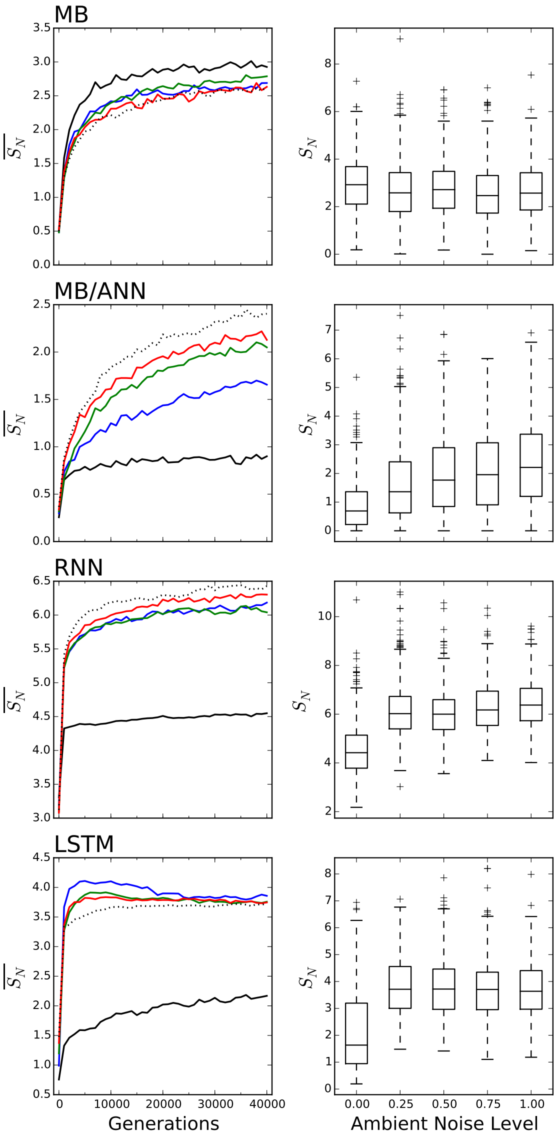

Figure 5: **Smearedness of Nodes Over Time**. For each type of brain the average value of S_N on the line of descent for all replicate experiments (left column). The black solid line is 0.0 noise, the blue line is 0.25 noise, the green line is 0.50 noise, the red line is 0.75 noise, and the black dotted line is 1.0 noise. The right column shows the distribution of performances including the 25% and 75% confidence intervals and their outliers for all brains under all experimental noise conditions.

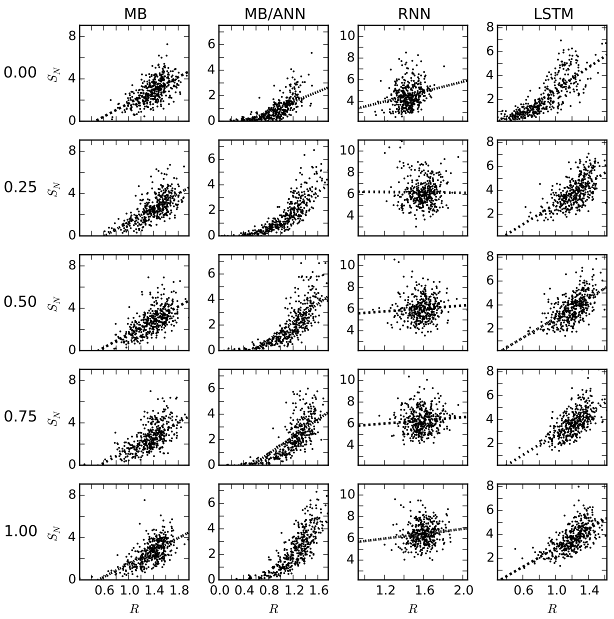

Figure 6: R **vs Smearedness of Nodes**. Each subplot is the scatterplot distribution of the value of S_N versus robustness to hidden noise, measured at the end of each evolutionary run. The dotted line in each subplot is the line of best fit of the data for each condition. Each column corresponds to a specific brain type, and each row corresponds to a given ambient noise level

R as well as the smearedness S_N. We find that in the majority of cases, the more representations the brains evolve the more they are effected by internal noise (see Figure 7). Only RNNs are somewhat unaffected and do not show a strong correlation between R and robustness to internal noise.

The same correlation can be found between the smearedness of representations and robustness to internal noise (see Figure 8 for the correlation of S_N and robustness). This shows that as hypothesized the increase of representations and their smearing over internal nodes is indeed functional and not just coincidental or a measuring artifact.

Discussion

Here we investigated the role that sensor noise has on the ability of different cognitive systems to evolve representations. Specifically, MBs with deterministic logic gates and ANN gates, RNNs, and LSTMs were used as examples of cognitive systems. We already knew that the smearing of representations, as happens in RNNs and LSTMs, makes these systems more vulnerable to internal noise. The hope was to improve our ability to evolve these systems by introducing sensor noise. While adding noise improves the performance of RNNs and LSTMs, MBs seem to be unaffected regardless of using logic gates or ANN gates. However, except for RNNs, those systems that improve their performance due to noise increase the amount of information that they have about the environment (R) and also smear these representations over their hidden states. Furthermore, these smeared representations are functional since they make the networks more susceptible to internal noise. Even though we did not test our hypotheses on more systems, we assume that our results can generalize to other types of computational cognitive models. Similarly, we only used the number discrimination task, and it is possible that other tasks respond differently. Both points suggest to test the effect of sensor noise in other cognitive systems and on other cognitive tasks. Here we also focused on evolutionary adaptation, and in most other cases RNNs and LSTMs are not trained by a genetic algorithm but instead gradient descent deep learning. This strongly suggests that we need to explore how these systems develop representations and to what degree the representations are smeared in the deep learning context.

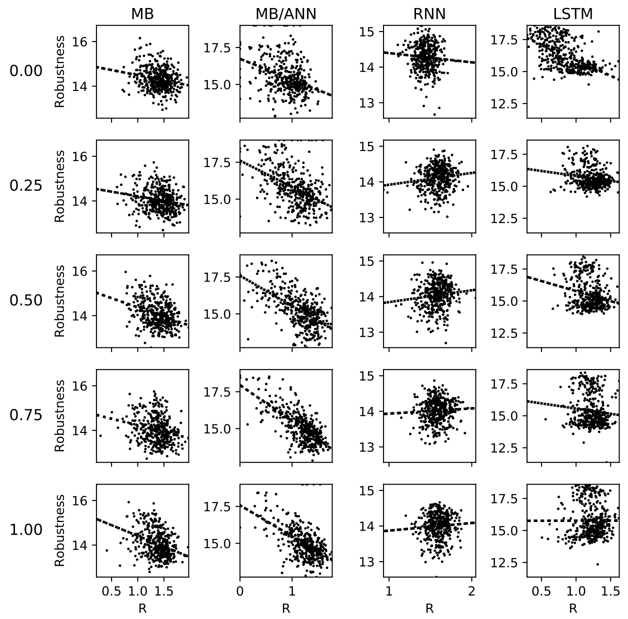

Figure 7: R **vs. Robustness**. Each subplot is the scatterplot distribution of the value of R versus robustness to hidden noise, measured at the end of each evolutionary run. The dotted line in each subplot is the line of best fit of the data for each condition. Each column corresponds to a specific brain type, and each row corresponds to a given ambient noise level

Conclusion

We found our initial intuition confirmed that the evolution of cognitive systems can be improved by using sensor noise, more so with RNNs and LSTMs than with Markov Brains. However, this improvement comes with a price which one might only pay under certain conditions. Using sensor noise to improve a neural networks ability to evolve will only help

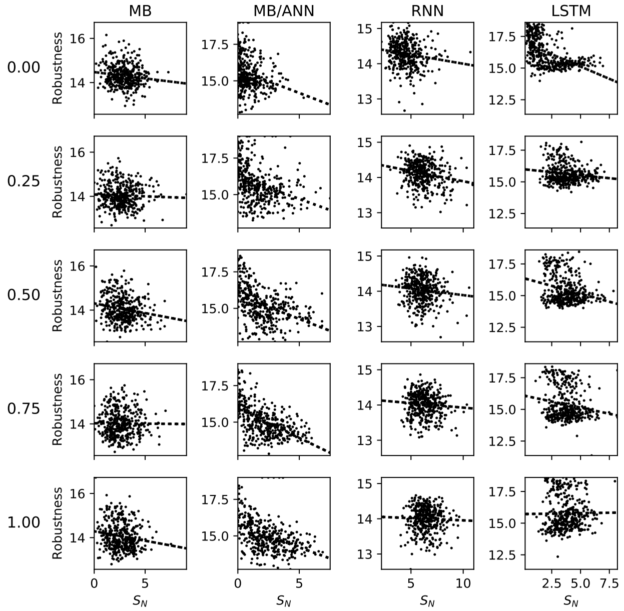

Figure 8: **Smearedness of Nodes vs. Robustness**. Each subplot is the scatterplot distribution of the value of S_N versus robustness to hidden noise, measured at the end of each evolutionary run. The dotted line in each subplot is the line of best fit of the data for each condition. Each column corresponds to a specific brain type, and each row corresponds to a given ambient noise level

if internal noise can be prevented.

Markov Brains using only deterministic logic gates experience a slight loss in performance due to noise, and do not form more representations and also do not further smear them. While this means that they do not experience a benefit from sensor noise, they also naturally do not smear representations. However, they also tend to have more representations at the end of evolutionary adaptation anyways. This suggests that one should use sensor noise in order to increase the amount of representations cognitive systems such as RNNs and LSTMs evolve. But at the same time, we should find another way to prevent systems from smearing said representations.

Acknowledgements

This work was supported in part by Michigan State University through computational resources provided by the Institute for Cyber-Enabled Research, and is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454.

References

Brown, W. M., Gedeon, T. D., and Groves, D. I. (2003). Use of noise to augment training data: a neural network method of mineral–potential mapping in regions of limited known deposit examples. *Natural Resources Research*, 12(2):141–152.

Hintze, A., Edlund, J. A., Olson, R. S., Knoester, D. B., Schossau, J., Albantakis, L., Tehrani-Saleh, A., Kvam, P., Sheneman, L., Goldsby, H., et al. (2017). Markov brains: A technical introduction. *arXiv preprint arXiv:1709.05601*.

Hintze, A., Kirkpatrick, D., and Adami, C. (2018). The structure of evolved representations across different substrates for artificial intelligence. In *Artificial Life Conference Proceedings*, pages 388–395. MIT Press.

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

Lauzon, F. Q. (2012). An introduction to deep learning. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pages 1438–1439. IEEE.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.

Marstaller, L., Hintze, A., and Adami, C. (2013). The evolution of representation in simple cognitive networks. *Neural computation*, 25:2079–2107.

Merritt, D. J. and Brannon, E. M. (2013). Nothing to it: Precursors to a zero concept in preschoolers. *Behavioural processes*, 93:91–97.

Merritt, D. J., Rugani, R., and Brannon, E. M. (2009). Empty sets as part of the numerical continuum: conceptual precursors to the zero concept in rhesus monkeys. *Journal of Experimental Psychology: General*, 138(2):258.

Moran, P. A. P. (1958). Random processes in genetics. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 54, pages 60–71. Cambridge University Press.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. (2015). Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

Nieder, A. (2018). Honey bees zero in on the empty set. *Science*, 360(6393):1069–1070.

Quiroga, R. Q., Fried, I., and Koch, C. (2013). Brain cells for grandmother. *Scientific American*, 308(2):30–35.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D. (2003). *Artificial intelligence: A modern approach*. Prentice Hall, Upper Saddle River.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schossau, J., Adami, C., and Hintze, A. (2015). Information-theoretic neuro-correlates boost evolution of cognitive systems. *Entropy*, 18(1):6.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210.