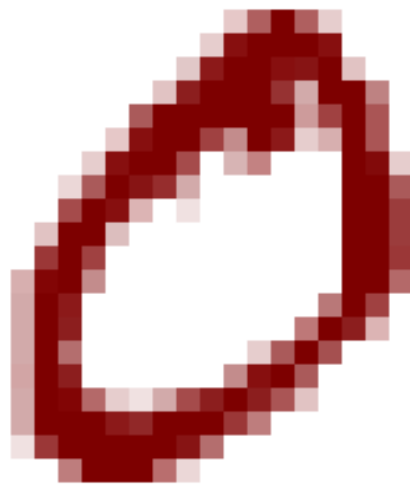


Self-classifying MNIST Digits

Achieving Distributed Coordination with Neural Cellular Automata



Summary. Each pixel is analogous to a biological cell. It decides its own color and communicates with its immediate neighbors. The goal of the cell population as a whole is to come to an agreement about what their global shape represents.

Usage. Interact with the cells by clicking or tapping on the canvas above. Press different digits to load or resample them. Press the bin to clear the canvas.





Speed: 1/10 x
(6 step/s)

Brush size:
(2 px)



Palette:
(0 deg)



This article relies on using color to demonstrate classification label. If you have trouble distinguishing the colours of digits in the above legend, please try and adjust the slider above to see if there is an alternative colour palette for you. The chosen palette will propagate throughout the article.

TRY IN A NOTEBOOK

AUTHORS

Ettore Randazzo
Alexander Mordvintsev
Eyvind Niklasson
Michael Levin
Sam Greydanus

PUBLISHED

Aug. 27, 2020

AFFILIATIONS

Google
Google
Google
Allen Discovery Center at Tufts University
Oregon State University and the ML Collective

DOI

10.23915/distill.00027.002

Contents

Model

Experiments

- Self-classify, persist & mutate
- Stabilizing classification

Robustness

Related Work

Discussion



This article is part of the [Differentiable Self-organizing Systems Thread](#), an experimental format collecting invited short articles delving into differentiable self-organizing systems, interspersed with critical commentary from several experts in adjacent fields.

[← PREVIOUS ARTICLE](#)

[Growing Neural Cellular Automata](#)

[NEXT ARTICLE →](#)

[Self-Organising Textures](#)

Growing Neural Cellular Automata [1] demonstrated how simple cellular automata (CAs) can learn to self-organise into complex shapes while being resistant to perturbations. Such a computational model approximates a solution to an open question in biology, namely, how do cells cooperate to create a complex multicellular anatomy and work to regenerate it upon damage? The model parameterizing the cells' rules is parameter-efficient, end-to-end differentiable, and illustrates a new approach to modeling the regulation of anatomical homeostasis. In this work, we use a version of this model to show how CAs can be applied to a common task in machine learning: classification. We pose the question: *can CAs use local message passing to achieve global agreement on what digit they compose?*

Our question is closely related to another unsolved problem in developmental and regenerative biology: how cell groups decide whether an organ or tissue pattern is correct, or whether current anatomy needs to be remodeled (anatomical surveillance and repair toward a specific target morphology). For example, when scientists surgically transplanted a salamander tail to its flank, it slowly remodeled into a limb - the organ that belongs at this location [2]. Similarly, tadpoles with craniofacial organs in the wrong positions usually become normal frogs because they remodel their faces, placing the eye, mouth, nostrils, etc. in their correct locations. Cell groups move around and stop when the correct frog-specific anatomical configuration has been achieved [3]. All of these examples illustrate the ability of biological systems to determine their current anatomical structure and decide whether it matches a species-specific target morphology [4]. Despite the recent progress in molecular biology of genes necessary for this process, there is still a fundamental knowledge gap concerning the algorithms sufficient for cell collectives to measure and classify their own large-scale morphology. More broadly, it is important to create computational models of swarm intelligence that explicitly define and distinguish the dynamics of the basal cognition of single cells versus cell collectives [5] [6].

The self-classifying MNIST task

Suppose a population of agents is arranged on a grid. They do not know where they are in the grid and they can only communicate with their immediate neighbors. They can also observe whether a neighbor is missing. Now suppose these agents are arranged to form the shape of a digit. Given that all the

agents operate under the same rules, can they form a communication protocol such that, after a number of iterations of communication, *all of the agents know which digit they are forming?*

Furthermore, if some agents were to be removed and added to form a new digit from a preexisting one, would they be able to know which the new digit is?

Because digits are not rotationally invariant (i.e. 6 is a rotation of 9), we presume the agents must be made aware of their orientation with respect to the grid. Therefore, while they do not know where they are, they do know where up, down, left and right are. The biological analogy here is a situation where the remodeling structures exist in the context of a larger body and a set of morphogen gradients or tissue polarity that indicate directional information with respect to the three major body axes. Given these preliminaries, we introduce the self-classifying MNIST task.



A visualisation of a random sample of digits from MNIST, each shaded by the colour corresponding its label.

Each sample of the MNIST dataset [7] consists of a 28x28 image with a single monochrome channel that is classically displayed in greyscale. The label is an integer in $[0, 9]$.

Our goal is for all cells that make up the digit to correctly output the label of the digit. To convey this structural information to the cells, we make a distinction between alive and dead cells by rescaling the values of the image to $[0, 1]$. Then we treat a cell as alive if its value in the MNIST sample is larger than 0.1. The intuition here is that we are placing living cells in a cookie cutter and asking them to identify the global shape of the cookie cutter. We visualize the label output by assigning a color to each cell, as you can see above. We use the same mapping between colors and labels throughout the article. Please note that there is a slider in the interactive demo controls which you can use to adjust the color palette if you have trouble differentiating between the default colors.

Model

In this article, we use a variant of the neural cellular automata model described in Growing Cellular Automata [1]. We refer readers unfamiliar with its implementation to the original "Model" section. Here we will describe a few areas where our model diverges from the original.

Target labels

The work in Growing CA used RGB images as targets, and optimized the first three state channels to approximate those images. For our experiments, we treat the last ten channels of our cells as a pseudo-distribution over each possible label (digit). During inference, we simply pick the label corresponding to the channel with the highest output value.

Alive cells and cell states

In Growing CA we assigned a cell's state to be "dead" or "alive" based on the strength of its alpha channel and the activity of its neighbors. This is similar to the rules of Conway's Game of Life [8]. In the Growing CA model, "alive" cells are cells which update their state and dead cells are "frozen" and do not undergo updates. In contrast to biological life, what we call "dead" cells aren't dead in the sense of being non-existent or decayed, but rather frozen: they are visible to their neighbors and maintain their state throughout the simulation. In this work, meanwhile, we use input pixel values to determine whether cells are alive or dead and perform computations with alive cells only¹. It is important to note that the values of MNIST pixels are exposed to the cell update rule as an immutable channel of the cell state. In other words, we make cells aware of their own pixel intensities as well as those of their neighbors. Given 19 mutable cell state channels (nine general purpose state channels for communication and ten output state channels for digit classification) and an immutable pixel channel, each cell perceives $19 + 1$ state channels and only outputs state updates for the 19 mutable state channels.

A note on digit topology. Keen readers may notice that our model requires each digit to be a single connected component in order for classification to be possible, since any disconnected components will be unable to propagate information between themselves. We made this design decision in order to stay true to our core biological analogy, which involves a group of cells that is trying to identify its global shape. Even though the vast majority of samples from MNIST are fully connected, some aren't. We do not expect our models to classify non-connected minor components correctly, but we do not remove them ².

Perception

The Growing CA article made use of fixed 3x3 convolutions with Sobel filters to estimate the state gradients in \vec{x} and \vec{y} . We found that fully trainable 3x3 kernels outperformed their fixed counterparts and so used them in this work.

A note on model size. Like the Growing CA model, our MNIST CA is small by the standards of deep learning - it has less than 25k parameters. Since this work aims to demonstrate a novel approach to classification, we do not attempt to maximise the validation accuracy of the model by increasing the number of parameters or any other tuning. We suspect that, as with other deep neural network models, one would observe a positive correlation between accuracy and model size.

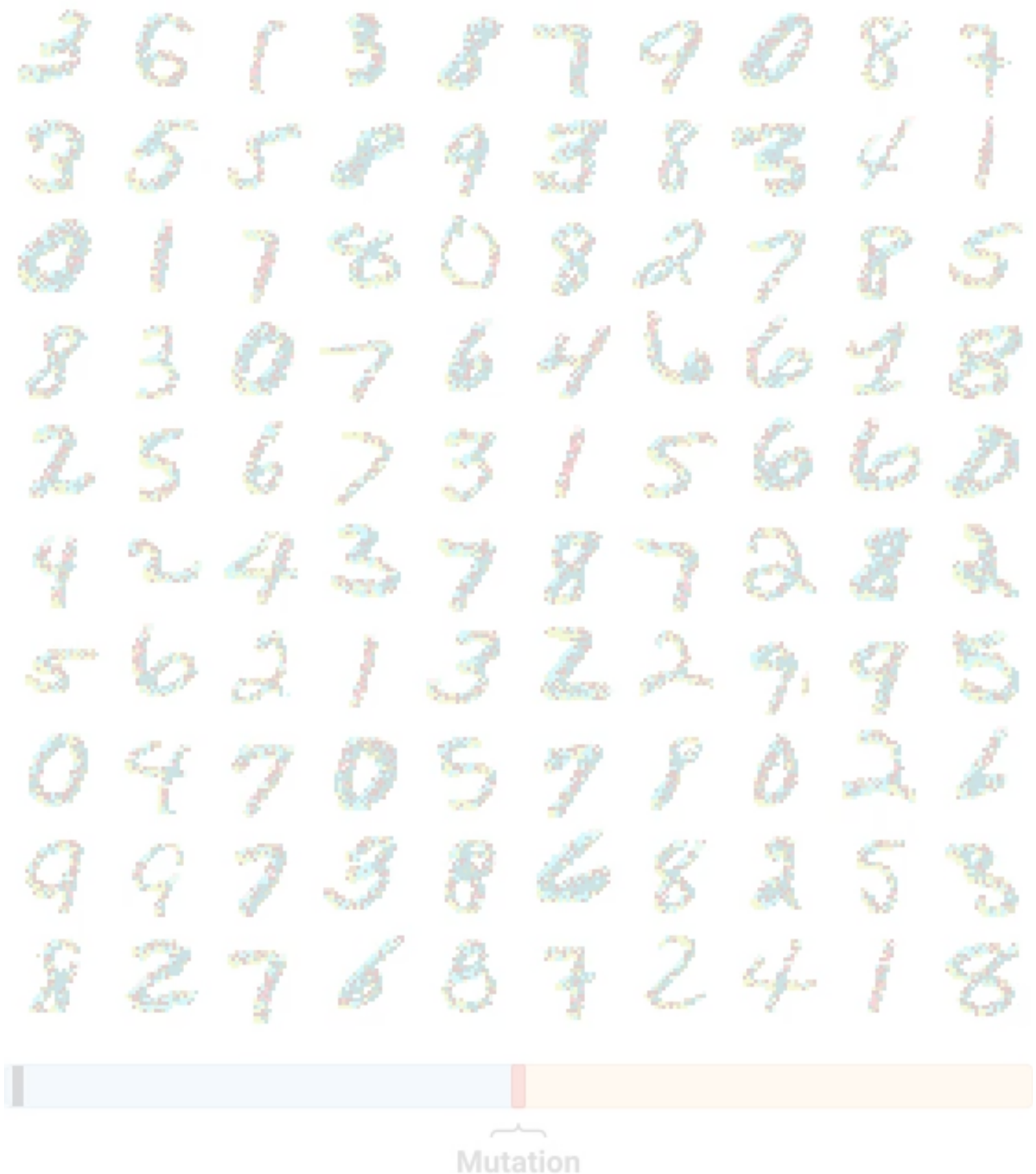
Experiment 1: Self-classify, persist and mutate

In our first experiment, we use the same training paradigm as was discussed in Growing CA. We train with a pool of initial samples to allow the model to learn to persist and then perturb the converged states. However, our perturbation is different. Previously, we destroyed the states of cells at random in order to make the CAs resistant to destructive perturbations (analogous to traumatic tissue loss). In this context, perturbation has a slightly different role to play. Here we aim to build a CA model that not only has regenerative properties, but also *has the ability to correct itself when the shape of the overall digit changes*.

Biologically, this corresponds to a teratogenic influence during development, or alternatively, a case of an incorrect or incomplete remodeling event such as metamorphosis or rescaling. The distinction between training our model from scratch and training it to accommodate perturbations is subtle but important. An important feature of life is the ability to react adaptively to external perturbations that are not accounted for in the normal developmental sequence of events. If our virtual cells simply learned to recognize a digit and then entered some dormant state and did not react to any further changes, we would be missing this key property of living organisms. One could imagine a trivial solution in the absence of perturbations, where a single wave of information is passed from the boundaries of the digit inwards and then back out, in such a way that all cells could agree on a correct classification. By introducing perturbations to new digits, the cells have to be in constant communication and achieve a "dynamic homeostasis" - continually "kept on their toes" in anticipation of new or further communication from their neighbours.

In our model, we achieve this dynamic homeostasis by randomly mutating the underlying digit at training time. Starting from a certain digit and after some time evolution, we sample a new digit, erase all cell states that are not present in both digits and bring alive the cells that were not present in the original digit but are present in the new digit. This kind of mutation teaches CAs to learn to process new information and adapt to changing conditions. It also exposes the cells to training states where all of the cells that remain after a perturbation are misclassifying the new digit and must recover from this catastrophic mutation. This in turn forces our CAs to learn to change their own classifications to adapt to changing global structure.

We use a pixel-wise (cell-wise) cross entropy loss on the last ten channels of each pixel, applying it after letting the CA evolve for 20 steps.



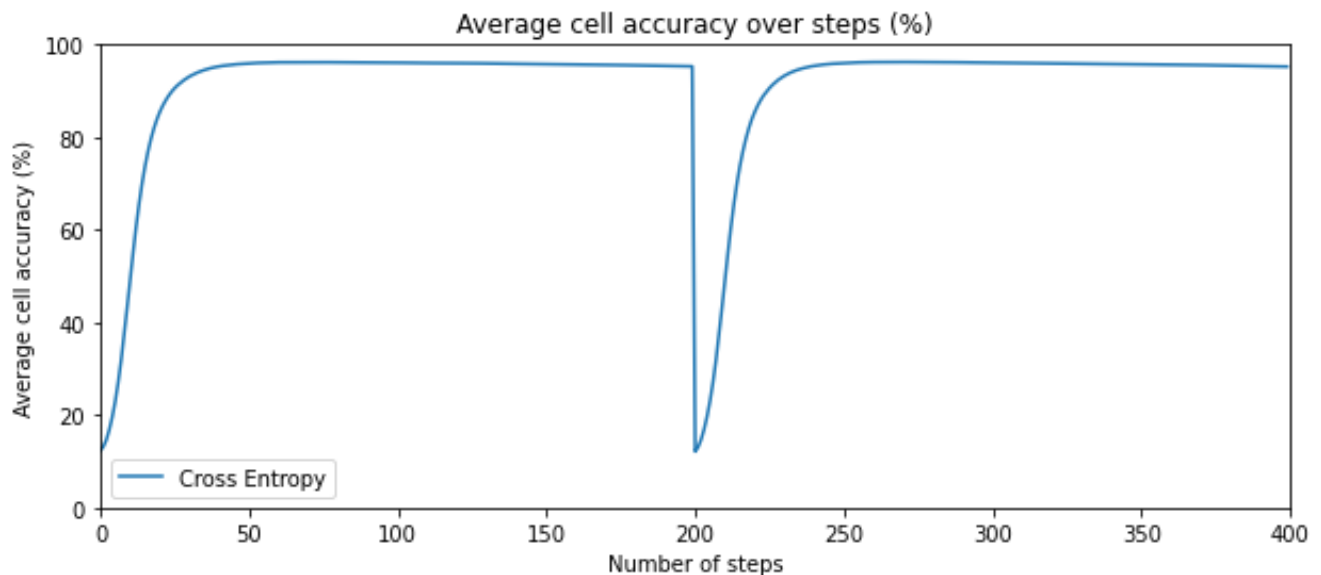
A first attempt at having the neural CAs classify digits. Each digit is a separate evolution of the neural CA, with the visualisations collated. Halfway through, the underlying digit is swapped for a new one - a "mutation".

The video above shows the CA classifying a batch of digits for 200 steps. We then mutate the digits and let the system evolve and classify for a further 200 steps.

The results look promising overall and we can see how our CAs are able to recover from mutations. However, astute observers may notice that often not all cells agree with each other. Often, the majority of the digit is classified correctly, but some outlier cells are still convinced they are part of a different digit, often switching back and forth in an oscillating pattern, causing a flickering effect in the visualization. This is not ideal, since we would like the population of cells to reach stable, total, agreement. The next experiment troubleshoots this undesired behaviour.

Experiment 2: Stabilizing classification

Quantifying a qualitative issue is the first step to solving it. We propose a metric to track **average cell accuracy**, which we define as the mean percentage of cells that have a correct output. We track this metric both before and after mutation.

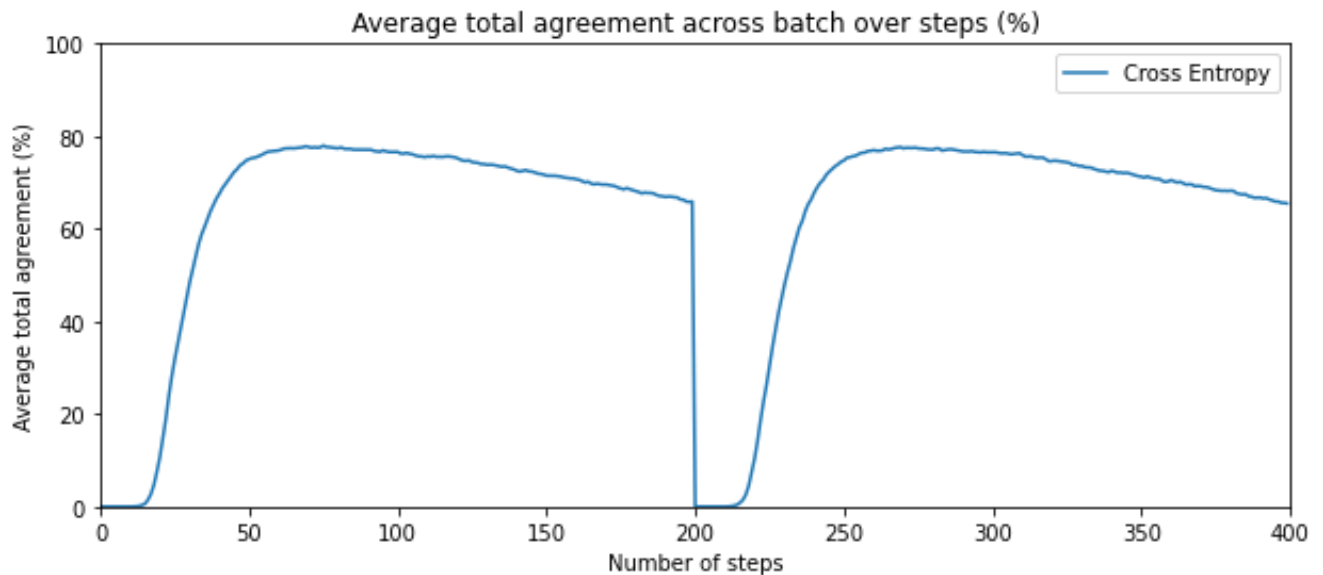


Average accuracy across the cells in a digit over time.

In the figure above, we show the mean percentage of correctly classified pixels in the test set over the course of 400 steps. At step 200, we randomly mutate the digit. Accordingly, we see a brief drop in accuracy as the cells re-organise and eventually come to agreement on what the new digit is.

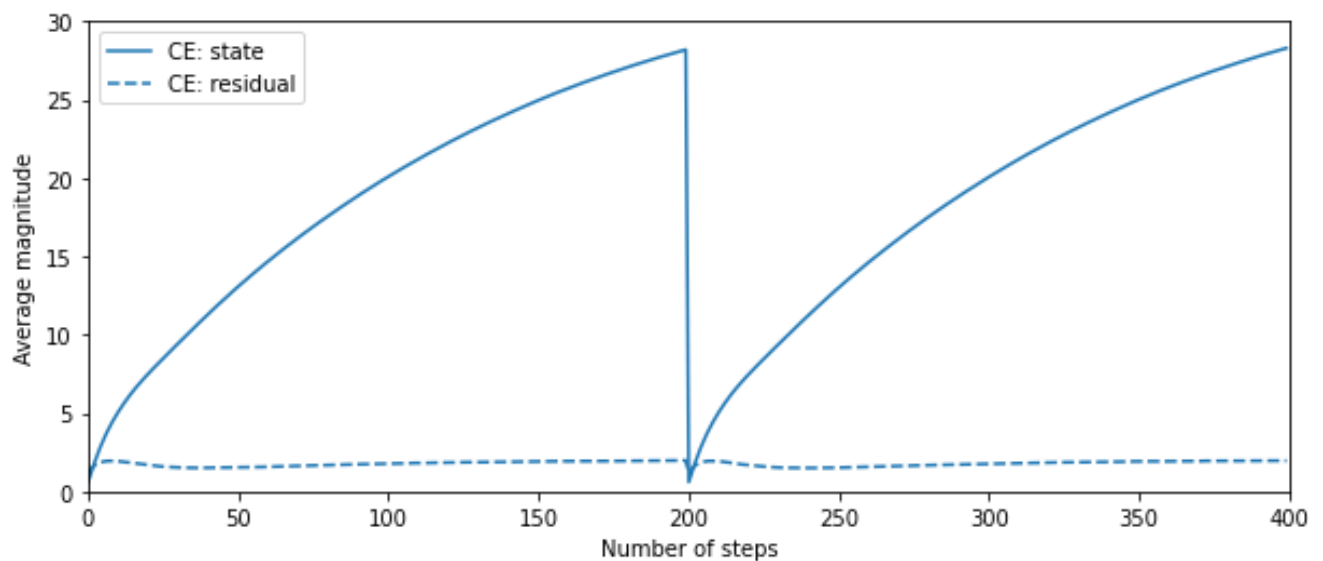
We immediately notice an interesting phenomenon: the cell accuracy appears to decrease over time after the cells have come to an agreement. However, the graph does not necessarily reflect the qualitative issue of unstable labels that we set out to solve. The slow decay in accuracy may be a reflection of the lack of total agreement, but doesn't capture the stark instability issue.

Instead of looking at the mean agreement perhaps we should measure **total agreement**. We define total agreement as the percentage of samples from a given batch wherein all the cells output the same label.



Average total agreement among cells across the test set in MNIST, over time.

This metric does a better job of capturing the issues we are seeing. The total agreement starts at zero and then spikes up to roughly 78%, only to lose more than 10% agreement over the next 100 steps. Again, behaviour after mutation does not appear to be significantly different. Our model is not only unstable in the short term, exhibiting flickering, but is also unstable over longer timescales. As time goes on, cells are becoming less sure of themselves. Let's inspect the inner states of the CA to see why this is happening.

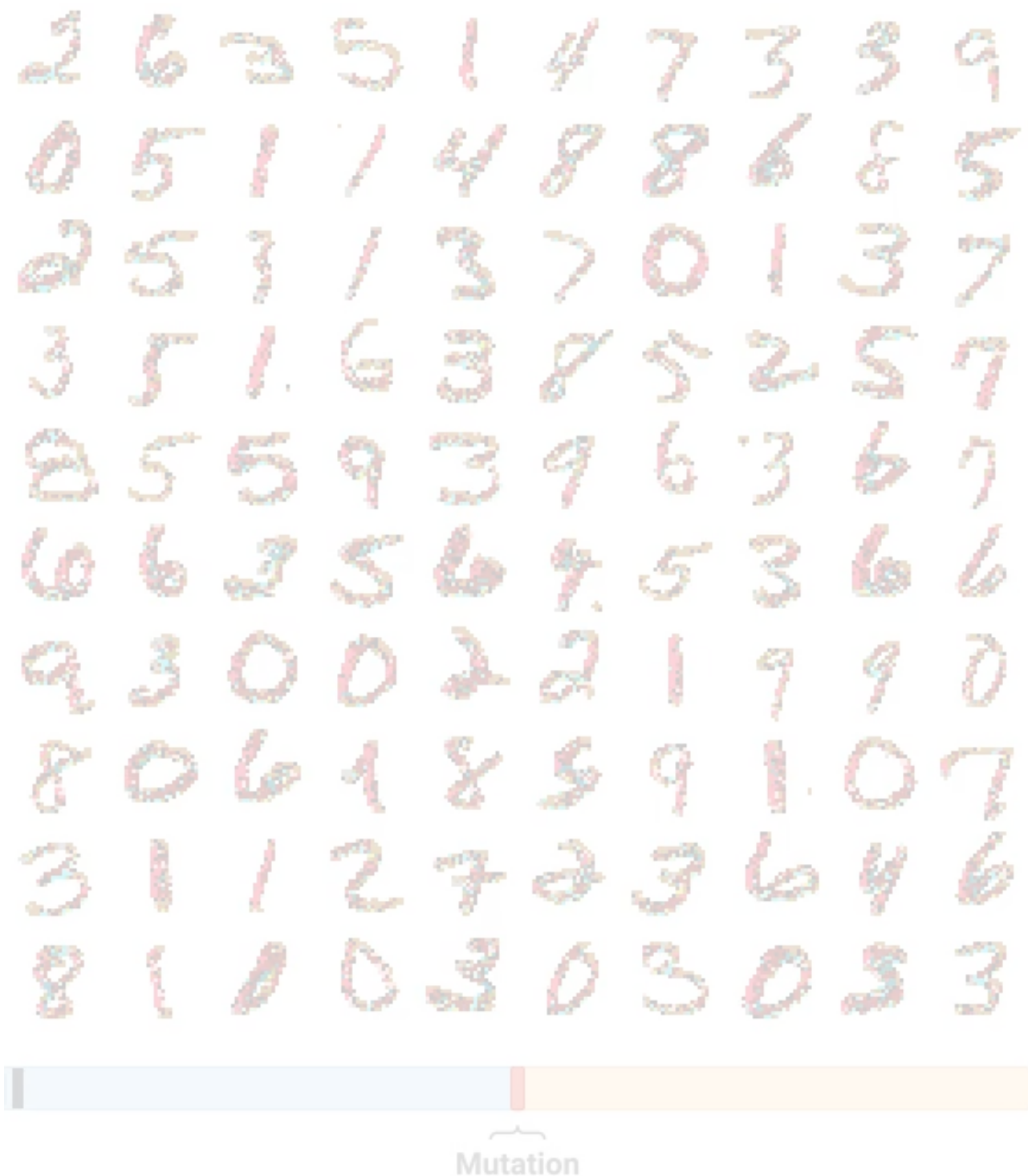


Average magnitude of the state channels and residual updates in active cells over time in the test set.

The figure above shows the time evolution of the average magnitude of the state values of active cells (solid line), and the average magnitude of the residual updates (dotted line). Two important things are happening here: 1) the average magnitude of each cell's internal states is growing monotonically on this timescale; 2) the average magnitude of the residual updates is staying roughly constant. We theorize that, unlike 1), a successful CA model should stabilize the magnitude of its internal states once cells have reached an agreement. In order for this to happen, its residual updates should approach zero over time, unlike what we observed in 2).

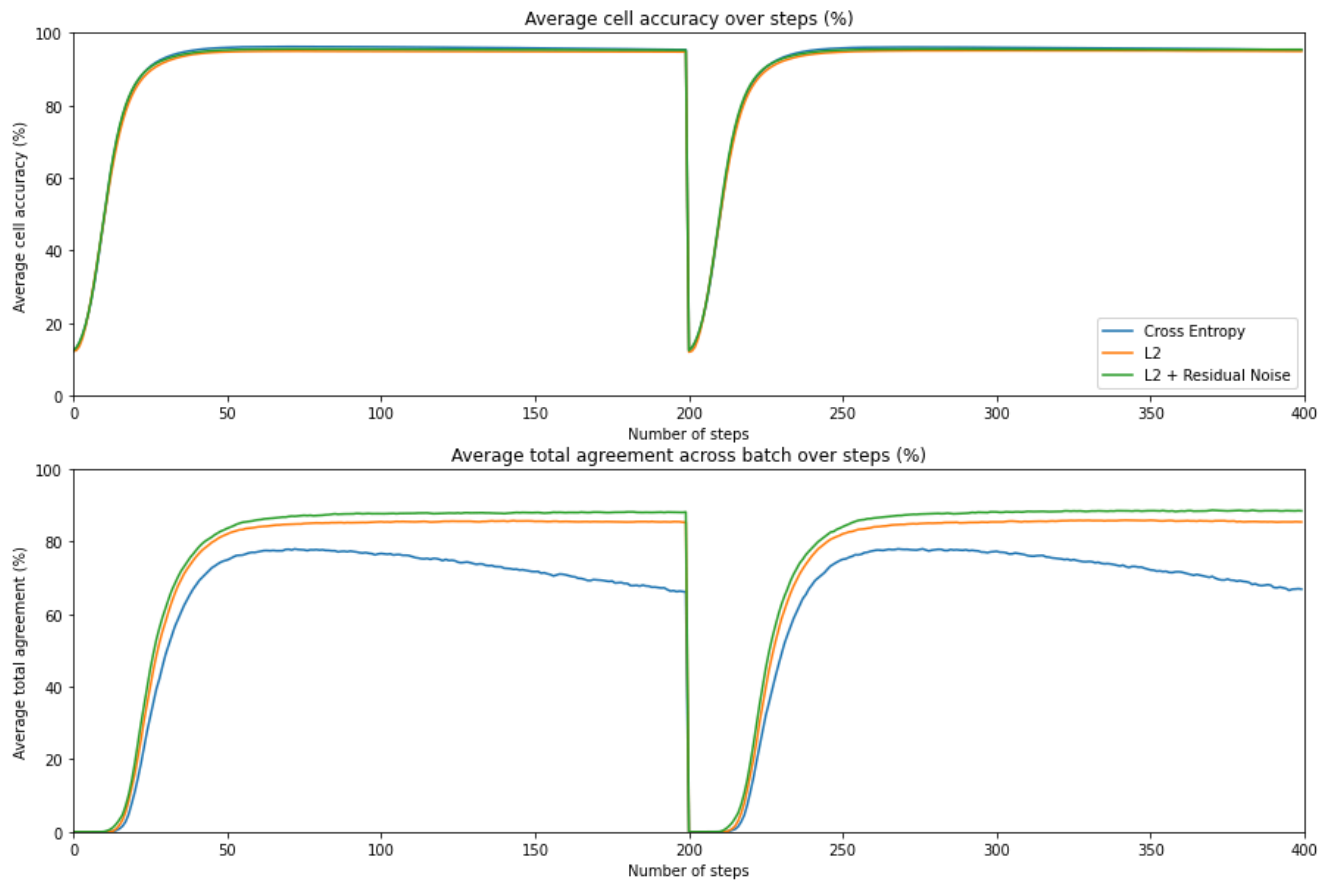
Using an L_2 loss. One problem with cross entropy loss is that it tends to push raw logit values indefinitely higher. Another problem is that two sets of logits can have vastly different values but essentially the same prediction over classes. As such, training the CA with cross-entropy loss neither requires nor encourages a shared reference range for logit values, making it difficult for the cells to effectively communicate and stabilize. Finally, we theorize that large magnitudes in the classification channels may in turn lead the remaining (non-classification) state channels to transition to a high magnitude regime. More specifically, we believe that *cross-entropy loss causes unbounded growth in classification logits, which prevents residual updates from approaching zero, which means that neighboring cells continue passing messages to each other even after they reach an agreement. Ultimately, this causes the magnitude of the message vectors to grow unboundedly.* With these problems in mind, we instead try training our model with a pixel-wise L_2 loss and use one-hot vectors as targets. Intuitively, this solution should be more stable since the raw state channels for classification are never pushed out of the range $[0, 1]$ and a properly classified digit in a cell will have exactly one classification channel set to 1 and the rest to 0. In summary, an L_2 loss should decrease the magnitude of all the internal state channels while keeping the classification targets in a reasonable range.

Adding noise to the residual updates. A number of popular regularization schemes involve injecting noise into a model in order to make it more robust [9, 10, 11, 12]. Here we add noise to the residual updates by sampling from a normal distribution with a mean of zero and a standard deviation of 2×10^{-2} . We add this noise before randomly masking the updates.



Neural CA trained with L_2 loss, exhibiting less instability after converging to a label.

The video above shows a batch of runs with the augmentations in place. Qualitatively, the result looks much better as there is less flickering and more total agreement. Let's check the quantitative metrics to see if they, too, show improvement.

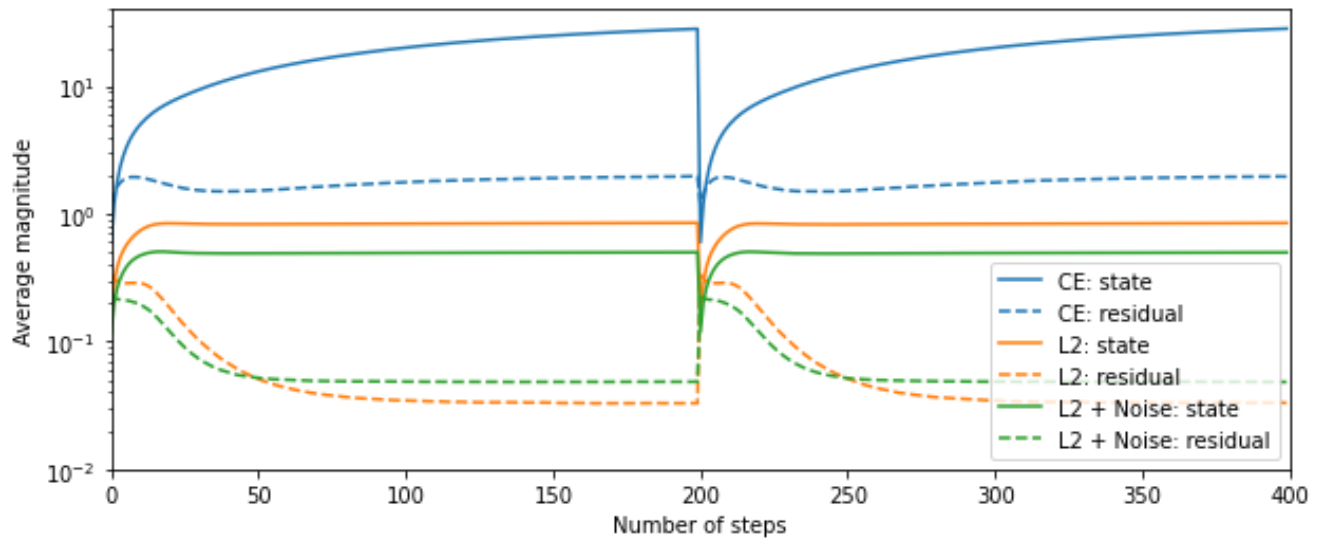


Comparison of average accuracy and total agreement when using cross-entropy and when using L_2 loss.

Model	Top accuracy	Accuracy at 200	Top agreement	Agreement at 200
CE	96.2 at 80	95.3	77.9 at 80	66.2
L_2	95.0 at 95	94.7	85.5 at 175	85.2
L_2 + Noise	95.4 at 65	95.3	88.2 at 190	88.1

The figure and table above show that cross-entropy achieves the highest accuracy of all models at roughly 80 steps. However, the accuracy at 200 steps is the same as the L_2 + Noise model. While accuracy and agreement degrade over time for all models, the L_2 + Noise appears to be the most stable configuration. In particular, note that the total agreement after 200 steps of L_2 + Noise is 88%, an improvement of more than 20% compared to the cross-entropy model.

Internal states



Average magnitude of state channels over time for L_2 loss and cross-entropy loss.

Let's compare the internal states of the augmented model to those of the original. The figure above shows how switching to an L_2 loss stabilizes the magnitude of the states, and how residual updates quickly decay to small values as the system nears agreement.



Visualisation of internal state channel values during mutations. Note the accelerated timeline after a few seconds showing the relative stability of the channel values.

To further validate our results, we can visualize the dynamics of the internal states of the final model. For visualization purposes, we have squashed the internal state values by applying an element-wise \arctan , as most state values are less than one but a few are much larger. The states converge to stable configurations quickly and the state channels exhibit spatial continuity with the neighbouring states. More specifically, we don't see any stark discontinuities in state values of neighbouring pixels. Applying a mutation causes the CA to readapt to the new shape and form a new classification in just a few steps, after which its internal values are stable.

Robustness

Recall that during training we used random digit mutations to ensure that the resulting CA would be responsive to external changes. This allowed us to learn a dynamical system of agents which interact to produce stable behavior at the population level, even when perturbed to form a different digit from the original. Biologically, this model helps us understand the mutation insensitivity of some large-scale anatomical control mechanisms. For example, planaria continuously accumulate mutations over millions of years of somatic inheritance but still always regenerate the correct morphology in nature (and exhibit no genetic strains with new morphologies) [13].

This robustness to change was critically important to our interactive demo, since the cells needed to reclassify drawings as the user changed them. For example, when the user converted a six to an eight, the cells needed to quickly re-classify themselves to an eight. We encourage the reader to play with the interactive demo and experience this for themselves. In this section, we want to showcase a few behaviours we found interesting.



Demonstration of the CA successfully re-classifying a digit when it is modified by hand.

The video above shows how the CA is able to interactively adjust to our own writing and to change classification when the drawing is updated.

Robustness to out-of-distribution shapes

In the field of machine learning, researchers take great interest in how their models perform on out-of-

distribution data. In the experimental sections of this article, we evaluated our model on the test set of MNIST. In this section, we go further and examine how the model reacts to digits drawn by us and not sampled from MNIST at all. We vary the shapes of the digits until the model is no longer capable of classifying them correctly. Every classification model inherently contains certain inductive biases that render them more or less robust to generalizing to out-of-distribution data. Our model can be seen as a recurrent convolutional model and thus we expect it to exhibit some of the key properties of traditional convolutional models such as translation invariance. However, we strongly believe that the self-organising nature of this model introduces a novel inductive bias which may have interesting properties of its own. Biology offers examples of “repairing to novel configurations”: 2-headed planaria, once created, regenerate to this new configuration which was not present in the evolutionary “training set” [\[14\]](#).



Demonstration of some of the failure cases of the CA.

Above, we can see that our CA fails to classify some variants of 1 and 9. This is likely because MNIST training data is not sufficiently representative of all writing styles. We hypothesize that more varied and extensive datasets would improve performance. The model often oscillates between two attractors (of competing digit labels) in these situations. This is interesting because this behavior could not arise from static classifiers such as traditional convolutional neural networks.



Demonstration of the inherent robustness of the model to unseen sizes and variants of numbers.

By construction, our CA is translation invariant. But perhaps surprisingly, we noticed that our model is also scale-invariant for out-of-distribution digit sizes up to a certain point. Alas, it does not generalize well enough to classify digits of arbitrary lengths and widths.



Demonstration of the behaviour of the model with chimeric configurations.

It is also interesting to see how our CA classifies “chimeric digits”, which are shapes composed of multiple digits. First, when creating a 3-5 chimera, the classification of 3 appears to dominate that of the 5. Second, when creating a 8-9 chimera, the CAs reach an oscillating attractor where sections of the two digits are correctly classified. Third, when creating a 6-9 chimera, the CAs converge to an oscillating attractor but the 6 is misclassified as a 4. These phenomena are important in biology as scientists begin to develop predictive models for the morphogenetic outcome of chimeric cell collectives. We still do not have a framework for knowing in advance what anatomical structures will form from a combination of, for example leg-and-tail blastema cells in an axolotl, heads of planaria housing stem cells from species with different head shapes, or composite embryos consisting of, for example, frog and axolotl blastomeres [15, 16]. Likewise, designing information signals that induce the emergence of desired tissue patterns from a chimeric cellular collective, in vitro or in vivo, remains an open problem.

Related Work

This article is follow-up work to Growing Neural Cellular Automata [1], and it is meant to be read after the latter. In this article, we purposefully skim over details of the original model and we refer the reader to the Growing Neural Cellular Automata article for the [full model description](#) section and [related work](#) section.

MNIST and CA. Since CAs are easy to apply to two dimensional grids, many researchers wondered if they could use them to somehow classify the MNIST dataset. We are aware of work that combines CAs with Reservoir Computing [17, 18], Boltzmann Machines [19], Evolutionary Strategies [20], and ensemble methods [21, 22]. To the best of our knowledge, we are the first to train end-to-end differentiable Neural CAs for classification purposes and we are the first to introduce the self-classifying variant of MNIST wherein each pixel in the digit needs to coordinate locally in order to reach a global agreement about its label.

Discussion

This article serves as a proof-of-concept for how simple self-organising systems such as CA can be used for classification when trained end-to-end through backpropagation.

Our model adapts to writing and erasing and is surprisingly robust to certain ranges of digit stretching and brush widths. We hypothesize that self-organising models with constrained capacity may be inherently robust and have good generalisation properties. We encourage future work to test this hypothesis.

From a biological perspective, our work shows we can teach things to a collective of cells that they could not learn individually (by training or engineering a single cell). Training cells in unison (while communicating with each other) allows them to learn more complex behaviour than any attempt to train them one by one, which has important implications for strategies in regenerative medicine. The current focus on editing individual cells at the genetic or molecular signaling level faces fundamental barriers when trying to induce desired complex, system-level outcomes (such as regenerating or remodeling whole organs). The inverse problem of determining which cell-level rules (e.g., genetic information) must be changed to achieve a global outcome is very difficult. In contrast and complement to this approach, we show the first component of a roadmap toward developing effective strategies for communication with cellular collectives. Future advances in this field may be able to induce desired outcomes by using stimuli at the system's input layer (experience), not hardware rewiring, to re-specify outcomes at the tissue, organ, or whole-body level [23] [24].



This article is part of the [Differentiable Self-organizing Systems Thread](#), an experimental format collecting invited short articles delving into differentiable self-organizing systems, interspersed with critical commentary from several experts in adjacent fields.

[← PREVIOUS ARTICLE](#)

[Growing Neural Cellular Automata](#)

[NEXT ARTICLE →](#)

[Self-Organising Textures](#)

Acknowledgments

We thank Zhitao Gong, Alex Groznykh, Nick Moran, Peter Whidden for their valuable conversations and feedback.

Author

Contributions

Research: Alexander came up with the Self-Organising Asynchronous Neural Cellular Automata model and Ettore contributed to its design. Alexander came up with the self-classifying MNIST digits task. Ettore designed and performed the experiments for this work.

Demos: Ettore, Eyvind and Alexander contributed to the demo.

Writing and Diagrams: Ettore outlined the structure of the article, created graphs and videos, and contributed to the content throughout. Eyvind contributed to the content throughout, including video making and substantive editing and writing. Michael made extensive contributions to the article text, providing the biological context for this work. Sam extensively contributed to the text of the article.

Implementation details

TF.js playground. The demo shown in this work is made through Tensorflow.js (TF.js). In the colaboratory notebook described below, the reader can find customizable sizes of this playground, as well as more options for exploring pretrained models, trained without sampling from a pool of different initial states, or mutation mechanisms, or using a cross-entropy loss.

Colaboratory Notebook. All of the experiments, images and videos in this article can be recreated using the single notebook referenced at the beginning of the article. Furthermore, more training configurations are easily available: training without pooling, without mutations, with a different loss, with or without residual noise. In the colab, the user can find pretrained models for all these configurations, and customizable TF.js demos where one can try any configuration.

Comments on the Decentralized Review Process

In lieu of traditional peer review, part of the Threads experiment was to conduct a decentralized review of this article using the SelfOrg Slack channel. The editors' objective was to make the review process faster and more efficient by encouraging real-time communication between the authors and the researchers who care about the topic.

At the time of review, the SelfOrg channel contained 56 members. Six of them participated in the public review process. Others may have participated anonymously. The decentralized review process improved the article by:

- Updating the demo's color scheme to assist those with color blindness
- Improving the demo's overall API
- Quickly resolving an enormous number of word and sentence-level issues. Over two hundred comments were made and resolved in one week.
- Raising and resolving several technical issues

Although there were technical discussions, the majority of reviews focused on improving the article's clarity and formatting. This was an important contrast compared to Distill's default, and more traditional, peer-review processes. In that process, the majority of the feedback tends to be technical. Since much of this article's technical details were similar to those of the original Growing CA article, we found that the emphasis on clarity and usability was quite useful here. We suspect that some blend of traditional peer review to resolve technical issues and decentralized peer review to resolve clarity and usability would be optimal.

In fact, this "optimal blend" of review styles already happens informally. Many industry and academic research labs have an internal review process aimed at improving communication and writing quality. After this informal review process, researchers submit papers to a double-blind process which specializes in technical feedback. At Distill, we are interested in recreating this blended two-step review process at scale. We see it as a way to 1) bring more diverse perspectives into the review process and 2) give the authors more thorough feedback on their papers.

Footnotes

1. As introduced in the previous section, cells are considered alive if their normalized grey value is larger than 0.1. [\[↗\]](#)

2. This choice complicates comparison between the MNIST train/test accuracies of neural network classifiers vs. CAs. However, such a comparison is not in scope of this article. [[↩](#)]

References

1. Growing Neural Cellular Automata
Mordvintsev, A., Randazzo, E., Niklasson, E. and Levin, M., 2020. Distill. DOI: 10.23915/distill.00023
2. The transformation of a tail into limb after xenoplastic transplantation
Farinella-Ferruzza, N., 1956. *Experientia*, Vol 12(8), pp. 304–305. DOI: 10.1007/bf02159624
3. Normalized shape and location of perturbed craniofacial structures in the *Xenopus* tadpole reveal an innate ability to achieve correct morphology
Vandenberg, L.N., Adams, D.S. and Levin, M., 2012. *Developmental Dynamics*, Vol 241(5), pp. 863–878. DOI: 10.1002/dvdy.23770
4. Top-down models in biology: explanation and control of complex living systems above the molecular level
Pezzulo, G. and Levin, M., 2016. *Journal of The Royal Society Interface*, Vol 13(124), pp. 20160555. DOI: 10.1098/rsif.2016.0555
5. On Having No Head: Cognition throughout Biological Systems
Baluška, F. and Levin, M., 2016. *Frontiers in psychology*, Vol 7, pp. 902.
6. The biogenic approach to cognition
Lyon, P., 2006. *Cognitive processing*, Vol 7(1), pp. 11–29.
7. Gradient-based learning applied to document recognition
Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. *Proceedings of the IEEE*, Vol 86(11), pp. 2278–2324.
8. MATHEMATICAL GAMES [[link](#)]
Gardner, M., 1970. *Scientific American*, Vol 223(4), pp. 120--123. *Scientific American*, a division of Nature America, Inc.
9. Dropout: A Simple Way to Prevent Neural Networks from Overfitting [[HTML](#)]
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. *Journal of Machine Learning Research*, Vol 15(56), pp. 1929–1958.
10. Auto-Encoding Variational Bayes
Kingma, D.P. and Welling, M., 2013.
11. Practical Variational Inference for Neural Networks [[PDF](#)]
Graves, A., 2011. *Advances in Neural Information Processing Systems* 24, pp. 2348--2356. Curran Associates, Inc.
12. Noisy Networks for Exploration
Fortunato, M., Azar, M.G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C. and Legg, S., 2017.
13. Planarian regeneration as a model of anatomical homeostasis: Recent progress in biophysical and computational approaches [[link](#)]
Levin, M., Pietak, A.M. and Bischof, J., 2019. *Seminars in Cell & Developmental Biology*, Vol 87, pp. 125 - 144. DOI: <https://doi.org/10.1016/j.semcd.2018.04.003>
14. Long-range neural and gap junction protein-mediated cues control polarity during planarian regeneration [[link](#)]
Oviedo, N.J., Morokuma, J., Walentek, P., Kema, I.P., Gu, M.B., Ahn, J., Hwang, J.S., Gojobori, T. and Levin, M., 2010. *Developmental Biology*, Vol 339(1), pp. 188 - 199. DOI: <https://doi.org/10.1016/j.ydbio.2009.12.012>
15. Bioelectrical Mechanisms for Programming Growth and Form: Taming Physiological Networks for Soft Body Robotics [[link](#)].

Mustard, J. and Levin, M., 2014. Soft Robotics, Vol 1(3), pp. 169–191. DOI: 10.1089/soro.2014.0011

16. Interspecies chimeras

Suchy, F. and Nakauchi, H., 2018. Current opinion in genetics & development, Vol 52, pp. 36–41. DOI: 10.1016/j.gde.2018.05.007

17. Reservoir Computing Hardware with Cellular Automata

Morán, A., Frasser, C.F. and Rosselló, J.L., 2018.

18. Energy-Efficient Pattern Recognition Hardware With Elementary Cellular Automata

Morán, A., Frasser, C.F., Roca, M. and Rosselló, J.L., 2020. IEEE Transactions on Computers, Vol 69(3), pp. 392–401.

19. Asynchronous network of cellular automaton-based neurons for efficient implementation of Boltzmann machines

Matsubara, T. and Uehara, K., 2018. Nonlinear Theory and Its Applications, IEICE, Vol 9, pp. 24–35. DOI: 10.1587/nolta.9.24

20. An Approach to Searching for Two-Dimensional Cellular Automata for Recognition of Handwritten Digits

Oliveira, C.C. and de Oliveira, P.P.B., 2008. MICAI 2008: Advances in Artificial Intelligence, pp. 462--471. Springer Berlin Heidelberg.

21. Biologically inspired cellular automata learning and prediction model for handwritten pattern recognition [\[link\]](#)

Wali, A. and Saeed, M., 2018. Biologically Inspired Cognitive Architectures, Vol 24, pp. 77 - 86. DOI: <https://doi.org/10.1016/j.bica.2018.04.001>

22. Pattern Classification with Rejection Using Cellular Automata-Based Filtering

Jastrzebska, A. and Toro Sluzhenko, R., 2017. Computer Information Systems and Industrial Management, pp. 3--14. Springer International Publishing.

23. The body electric 2.0: recent advances in developmental bioelectricity for regenerative and synthetic bioengineering

Mathews, J. and Levin, M., 2018. Current opinion in biotechnology, Vol 52, pp. 134–144.

24. Re-memembering the body: applications of computational neuroscience to the top-down control of regeneration of limbs and other complex organs

Pezzulo, G. and Levin, M., 2015. Integrative biology: quantitative biosciences from nano to macro, Vol 7(12), pp. 1487–1517.

Updates and Corrections

If you see mistakes or want to suggest changes, please [create an issue on GitHub](#).

Reuse

Diagrams and text are licensed under Creative Commons Attribution [CC-BY 4.0](#) with the [source available on GitHub](#), unless noted otherwise. The figures that have been reused from other sources don't fall under this license and can be recognized by a note in their caption: "Figure from ...".

Citation

For attribution in academic contexts, please cite this work as

Randazzo, et al., "Self-classifying MNIST Digits", Distill, 2020.

BibTeX citation

```
@article{randazzo2020self-classifying,  
  author = {Randazzo, Ettore and Mordvintsev, Alexander and Niklasson, Eyvind and Levin,  
Michael and Greydanus, Sam},  
  title = {Self-classifying MNIST Digits},  
  journal = {Distill},  
  year = {2020},  
  note = {https://distill.pub/2020/selforg/mnist},  
  doi = {10.23915/distill.00027.002}  
}
```

Distill is dedicated to clear explanations of machine learning

[About](#) [Submit](#) [Prize](#) [Archive](#) [RSS](#) [GitHub](#) [Twitter](#) [ISSN 2476-0757](#)