Predictive coding is a consequence of energy efficiency in recurrent neural networks

Abdullahi Ali[1*], Nasir Ahmad[1], Elgar de Groot[1,2], Marcel A. J. van Gerven[1], and Tim C. Kietzmann[1*]

[1]Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, the Netherlands

[2]Department of Experimental Psychology, Utrecht University, Utrecht, the Netherlands

[*]Corresponding Authors: a1.ali@donders.ru.nl, t.kietzmann@donders.ru.nl

February 16, 2021

Abstract

Predictive coding represents a promising framework for understanding brain function. It postulates that the brain continuously inhibits predictable sensory input, ensuring a preferential processing of surprising elements. A central aspect of this view is its hierarchical connectivity, involving recurrent message passing between excitatory bottom-up signals and inhibitory top-down feedback. Here we use computational modelling to demonstrate that such architectural hard-wiring is not necessary. Rather, predictive coding is shown to emerge as a consequence of energy efficiency. When training recurrent neural networks to minimise their energy consumption while operating in predictive environments, the networks self-organise into prediction and error units with appropriate inhibitory and excitatory interconnections, and learn to inhibit predictable sensory input. Moving beyond the view of purely top-down driven predictions, we demonstrate via virtual lesioning experiments that networks perform predictions on two timescales: fast lateral predictions among sensory units, and slower prediction cycles that integrate evidence over time.

1 Introduction

In computational neuroscience and beyond, predictive coding has emerged as a prominent theory for how the brain encodes and processes sensory information (Srinivasan et al., 1982; Mumford, 1992; Friston, 2005; Clark, 2013). It postulates that higher-level brain areas continuously keep track of the causes of lower-level sensory input, and actively inhibit incoming sensory signals that match expectation. Over time, the brain's higher-level representations are shaped to yield increasingly accurate predictions that in turn minimise the surprise that the system encounters in its inputs. That is, the brain creates an increasingly accurate model of the external world and focuses on processing unexpected sensory events that yield the highest information gain.

Adding to the increasing, albeit often indirect, experimental evidence for predictive coding in the brain (Alink et al., 2010; Näätänen et al., 2001; Summerfield et al., 2011, 2008; Squires et al., 1975; Hupé et al., 1998; Murray et al., 2002; Rao et al., 2016; Kok et al., 2012; Ekman et al., 2017; De Lange et al., 2018; Dijkstra et al., 2020; Schwiedrzik and Freiwald, 2017), computational modelling has investigated explicit implementations of predictive coding, indicating that they can reproduce experimentally observed neural phenomena (Rao and Ballard, 1999; Lee and Mumford, 2003; Friston, 2005, 2010). For example, Rao and Ballard (1999) demonstrated that a hierarchical network with top-down inhibitory feedback can explain extra-classical field effects in V1. In addition, deep neural network architectures wired to implement predictive coding were shown to work at scale in real-world tasks (Chalasani and Principe, 2013; Lotter et al., 2016; Villegas et al., 2017; Lotter et al., 2020), adding further support for the computational benefits of recurrent message passing (Spoerer et al., 2020; Kietzmann et al., 2019; Kar et al., 2019; Linsley et al., 2020; Spoerer et al., 2017; van Bergen and Kriegeskorte, 2020). The common rationale of modelling work focused on predictive coding is to test for its computational and representational effects by hard-wiring the model circuitry to mirror hierarchical and inhibitory connectivity between units that drive predictions, and units that signal deviations from said predictions (also termed sensory, or error units).

Contrasting this approach, we here ask a different question: can computational mechanisms of predictive coding naturally emerge from other, perhaps simpler, computational principles? In search for such principles, we take inspiration from the organism's limited energy budget. Expanding on previous work on efficient coding and sparse coding (Barlow et al., 1961; Bell and Sejnowski, 1995; Olshausen and Field, 1996; Bialek et al., 2006; Berkes and Wiskott, 2005; Chalk et al., 2018; Eckmann et al., 2020), we subject recurrent neural networks (RNNs) to predictable sequences of visual input and optimise their synaptic weights to minimise what constitutes the largest source of energy consumption in biological systems: action potential generation and synaptic transmission (Sengupta et al., 2010). We then test the resulting networks for phenomena typically associated with predictive coding. We compare the energy budget of

the networks to theoretically-determined lower bounds, search for inhibitory connectivity profiles that mirror sensory predictions, and explore whether the networks automatically separate their neural resources into prediction and error units.

2 Results

To better understand whether, and under which conditions, predictive coding may emerge in energy-constrained neural networks, we trained a set of RNNs on predictable streams of visual input. For this, we created sequences of MNIST digits (LeCun et al., 1998) iterating through 10 digits in numerical (category) order. Starting from a random digit, sequences wrapped around from nine to zero. The visual input to each network unit was stimulus-dependent, with each unit receiving an input which mirrored its corresponding pixel intensity. These visual inputs were not weighted and thus we prevented the models from learning to ignore the input as a shortcut to a solution for energy efficiency. Instead, we hypothesised that the models would learn to use their recurrent connectivity to counteract the input drive where possible, and thereby to minimise their energy consumption. Energy costs were defined to arise from synaptic transmission and action potentials, thereby mirroring the two most dominant sources of energy consumption in neural communication (Sengupta et al., 2010). Figure 1 shows an overview of the experimental pipeline.

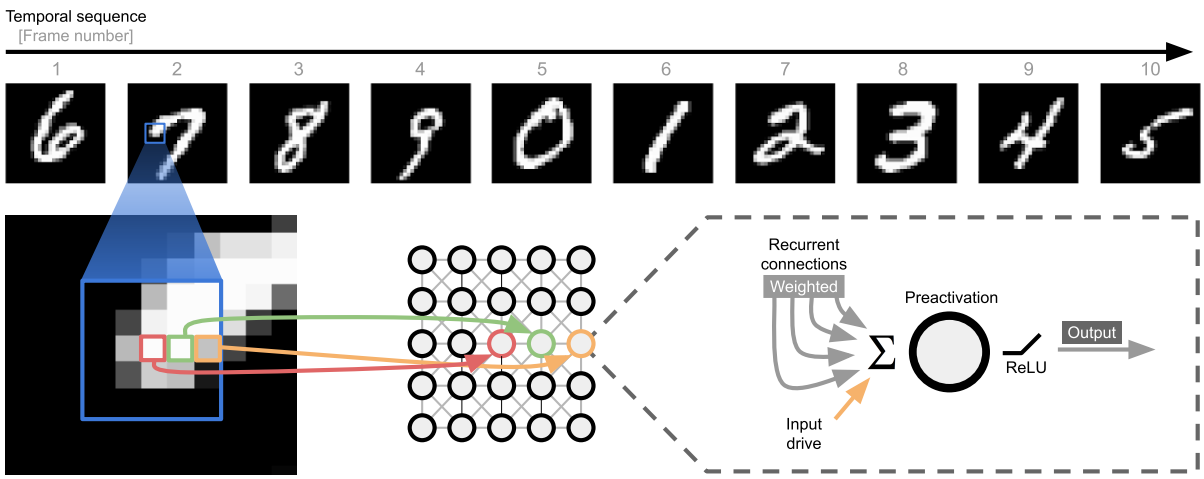

Figure 1: Overview of the model architecture and input pipeline. The upper part of the figure shows an example input sequence fed to a recurrent network. 10 MNIST digits are presented in ascending order to the network (with wraparound after digit 9). The lower part of the figure shows that each pixel drives one unit. The preactivation of a single unit constitutes the sum of the recurrent activity into the unit and the input drive determined by the intensity of the pixel that it is connected to. The output of the network is determined by taking the ReLU of the preactivations of the network units.

2.1 Preactivation as a proxy for the energy demands of the brain

Sparse coding is often implemented as an L1 objective on the unit outputs, i.e. networks are optimised to reduce the overall unit activity post non-linearity. However, given that synaptic transmission contributes considerable energy cost alongside action potentials, a different learning objective is required; i.e. a minimisation of unit outputs alone could be implemented via unreasonably strong inhibitory connections, which themselves might increase the overall energy budget. To solve this issue, we propose to instead train energy-efficient RNNs by minimising absolute (L1) unit preactivation. This has two desired properties. First, it drives unit output towards zero, mirroring a minimisation of unit firing rates (dependent naturally upon the form of activation function used). Second, we show that this objective also leads to minimal synaptic weight magnitudes in cases of noise in the system, thereby mirroring an overall reduction in synaptic transmission (see Appendix B for a theoretical derivation of why this is the case).

Following above considerations, which we regard as the first result of this paper, we confirmed that training RNNs in this manner indeed lead to a minimisation of the overall unit preactivation (i.e. a proxy for energy consumption in biological networks). We trained a total of ten network instances to ensure the generality of our results (Mehrer et al., 2020). The resulting network activity was compared to three theoretical scenarios. First, we computed the overall input drive, i.e. a scenario in which the lateral weights are removed and the RNNs are therefore completely driven by external input. This estimate measures the 'energy' that flows into the network from sensory sources alone. Second, we estimated the theoretical network activity if RNNs learned to predict the pixel-wise median value of the whole training data. This value represents a scenario in which the networks counteract the input drive based on the global dataset statistic, without sequence-specific categorical inhibitions. Third, we computed the theoretical activity in a best case scenario in which the RNNs learn to predict and inhibit the pixel-wise median value of the particular digit category being presented. This theoretical model has perfect knowledge of the current position in the input sequence and can

therefore inhibit the matching dataset's pixel-wise median. Given the training setup, this measure represents a lower bound on network preactivation (energy). A derivation for these bounds on performance is given in Section 4.3.

As can be seen in Figure 2, which depicts the theoretical bounds together with the average preactivation across ten trained RNN instances, the trained networks approach the optimal case with increasing network time (i.e. increasing exposure to the sequence). Whereas the RNNs cannot predict well at the first time step (they have no information about where the sequence begins), the average unit preactivation drops below the pixel-wise median of the training data after only one timestep and subsequently converge towards the category median, i.e. the lower bound. This indicates that the RNNs are able to integrate evidence over time, improving their inhibitory predictions with increasing sensory evidence for the current input state.

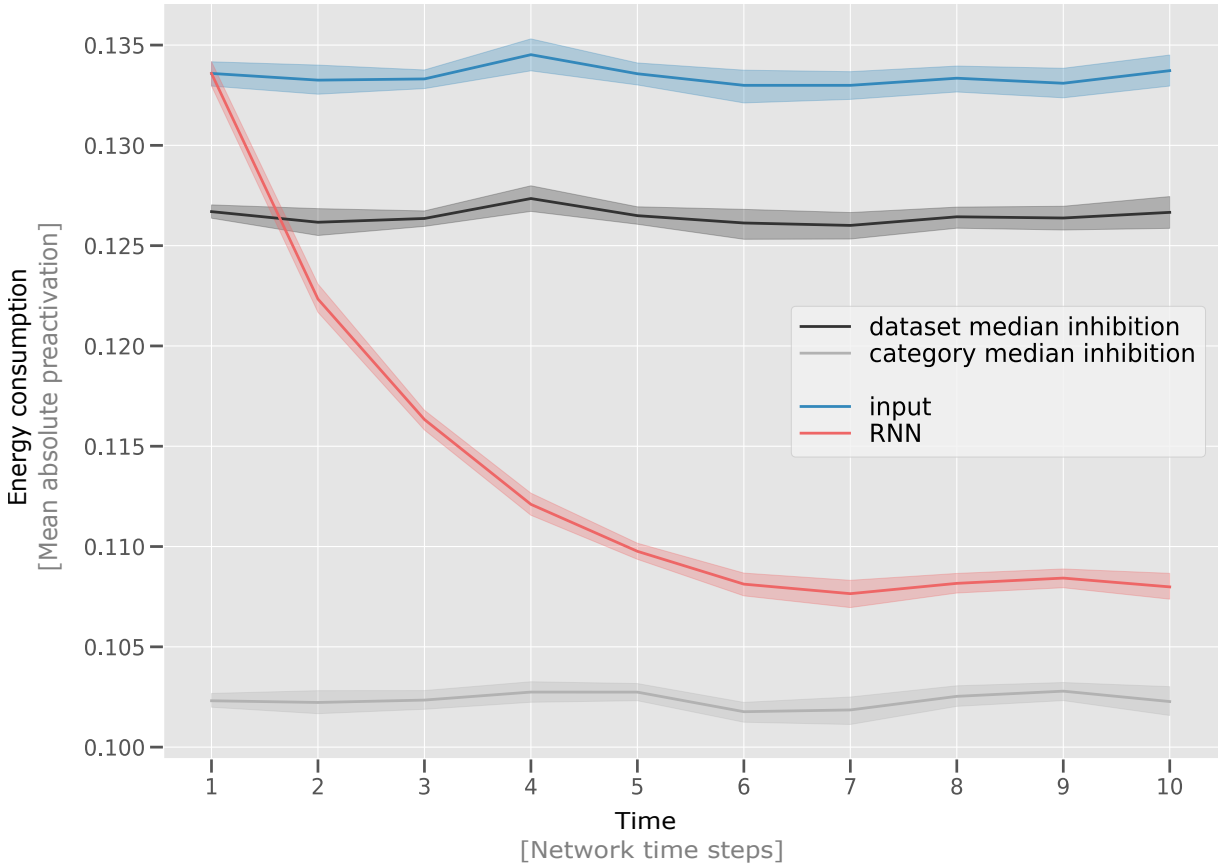

Figure 2: Evolution of the 'energy consumption' (mean absolute preactivation) of the trained RNNs, together with three theoretical scenarios. *input*: stimulation entering the networks via sensory input, *median*: network preactivation if the networks predicted and inhibited the (pixel-wise) median of the training data, *category median*: network preactivation if the networks had perfect knowledge of the current state of the sequence, and thereby inhibit the (pixel-wise) median value of the upcoming digit category. *RNN*: actual network preactivation. Mean preactivations shown with 95% confidence intervals, bootstrapped across 10 network instances with replacement.

2.2 RNNs trained for energy efficiency learn to predict and inhibit upcoming sensory signals

Having established that the trained RNNs successfully reduce their preactivation over time, we further confirmed a targeted inhibition of predicted sensory input, a hallmark of predictive coding, by visualising the network drive induced by the recurrent connections for a sample sequence in Figure 3 (top row: input drive, bottom row: (recurrent) internal drive, see Section 4.2 for these measures). Two observations are of importance in this example. First, the network clearly predicts and inhibits the upcoming digit category. Second, as time progresses the network is able to integrate information, leading to better knowledge of the current sequence position, and more targeted inhibitions (e.g. the inhibition of later digits appears more clear/targeted than earlier predictions). Better predictions, in turn, result in progressively lower network activity. Please note that this pattern of results is in line with hierarchical predictive coding architectures in which feedback from higher order areas is subtracted from sensory evidence of lower areas. However, no such hierarchical organisation was imposed in our RNNs. Rather, all effects emerge due to optimisation of the networks for energy efficiency.

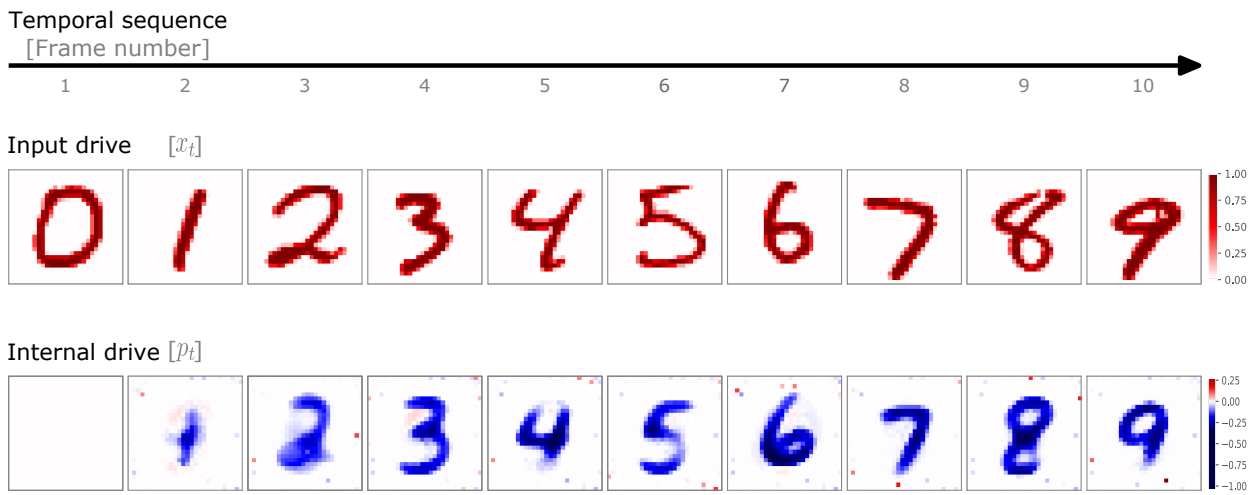

Figure 3: Network activity for an example sequence. The RNN units are organised according to the locations of their input pixels. Row 1 shows the input drive x_t and row 2 shows the recurrent drive p_t. Excitatory signals are depicted in red, while inhibitory signals are depicted in blue. The darker the colour the stronger the signal. The sequence images shown were not used for training. The observable improvements in inhibitory network drive indicate that the RNN integrates sensory evidence over time, leading to better internal knowledge of the state of the world

2.3 Separate error units and prediction units as an emergent property of RNN self-organisation

Our previous results demonstrate that RNNs, when optimised to reduce action potentials and synaptic transmission, exhibit key phenomena associated with predictive coding. Thus, predictive coding emerges, without architectural hard-wiring, as a result of energy efficiency in predictable sensory environments. A second key component postulated by the predictive coding framework is the existence of distinct neural populations that either provide sensory predictions, or which carry the deviations from such predictions, i.e. prediction error units (Bastos et al., 2012). Given the non-hierarchical, non-constrained nature of the current setup, we next asked whether a similar separation could also be observed in our energy efficient networks.

If the models had indeed developed separate populations of error and prediction units, they would be expected to have different activity profiles. Error units calculate the difference between the predicted input and the actual input by integrating excitatory signals from the input and inhibitory signals from prediction units. In the RNNs, this means that such units would receive excitatory signals from the input drive, and inhibitory signals from the recurrent connections. Prediction units, on the other hand, represent the internal representation of the external environment, and are updated over time as guided by the prediction errors. The prediction units are therefore expected to receive their primary input from the error units. To test for such activity patterns, we record, for an example network, the average excitatory signals from both the input and recurrent connections for each unit at the last timestep of the presented sequences. As can be seen from Figure 4, separate populations of network units can be observed in the trained network. While one population is predominantly driven by network-internal recurrent connections, another population receives its excitatory input predominantly from the sensory input.

Interestingly, the two populations occupy different topographic parts in the RNN, with units driven by network internal states (i.e. prediction units) residing in locations that are typically not strongly driven by sensory input. Such secondary use of neural resources that would otherwise remain unused seems in line with neuroscientific evidence for cortical plasticity (Kaas, 2002; Merabet and Pascual-Leone, 2010).

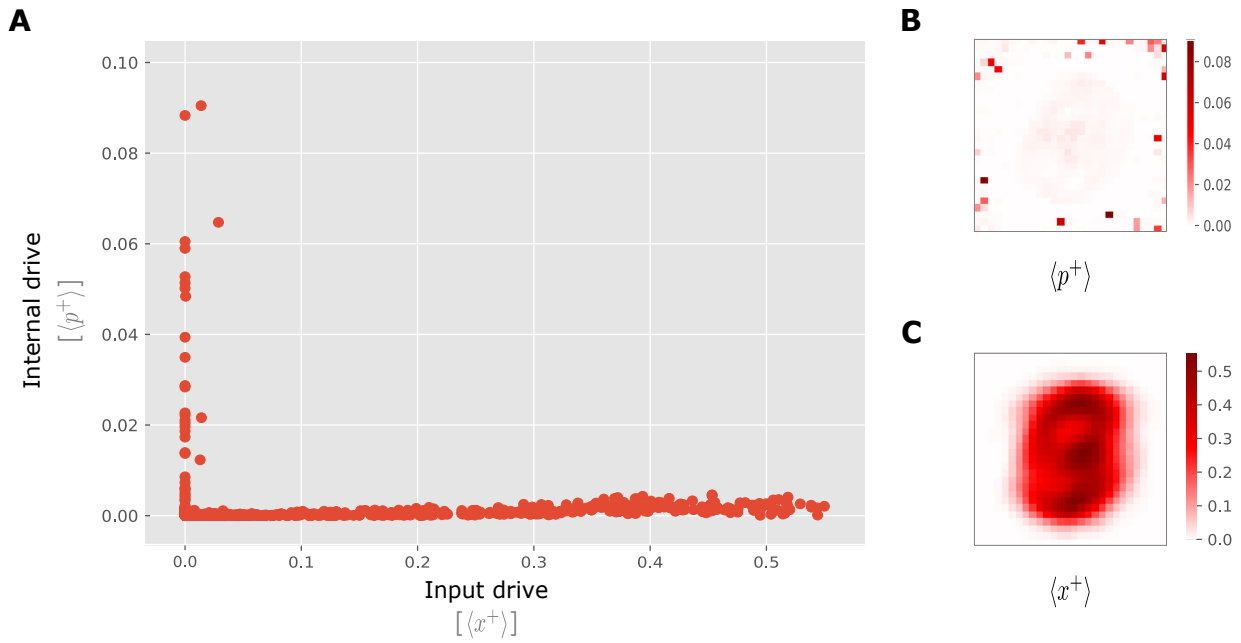

Figure 4: Trained RNNs exhibit distinct populations of input driven and feedback driven units. **A:** comparison of mean excitatory drive from input $\langle x^+ \rangle$ and recurrent feedback $\langle p^+ \rangle$ for an example network indicates a clear separation between input-driven and feedback-driven units. **B:** mean excitatory drive from recurrent connections for each unit in the RNN. **C:** mean excitatory input drive for each unit in the RNN.

2.4 Two distinct inhibitory mechanisms working at different timescales

As can be seen in Figures 2 and 3, the inhibitory predictions of the second timestep are less precise than later timesteps, but nevertheless contain some category specific predictions. This network behaviour is of interest, as prediction loops from input to prediction units and back require two timesteps to come into effect (one to activate prediction units and one for these prediction units to provide feedback). This suggests that the observed predictions in timestep 2 are rather driven by more immediate lateral connections among error units - to our knowledge a mode of predictive coding not previously considered. That is, we observe two different modes of predictive inhibitions. One operating at a faster timescale among error units, and one on a slower timescale involving prediction units with the potential to integrate evidence over time. This hypothesis of evidence integration via prediction units was confirmed by virtual lesion studies. As shown in Figure 5A, an un-lesioned RNN is capable of more fine-grained predictions with increasing sequence lengths leading up to the sensory input. On the other hand, a network in which (postulated) prediction units are lesioned remains fixed at the initial and immediate prediction, derived from lateral connections among error units, and is missing the ability to integrate evidence over time (see Supplemental Figures SA1 and SA2 for further lesion experiments targeting other digit categories). Prediction and error units were identified based upon the ratio of their input vs external drive (see Section 4.4).

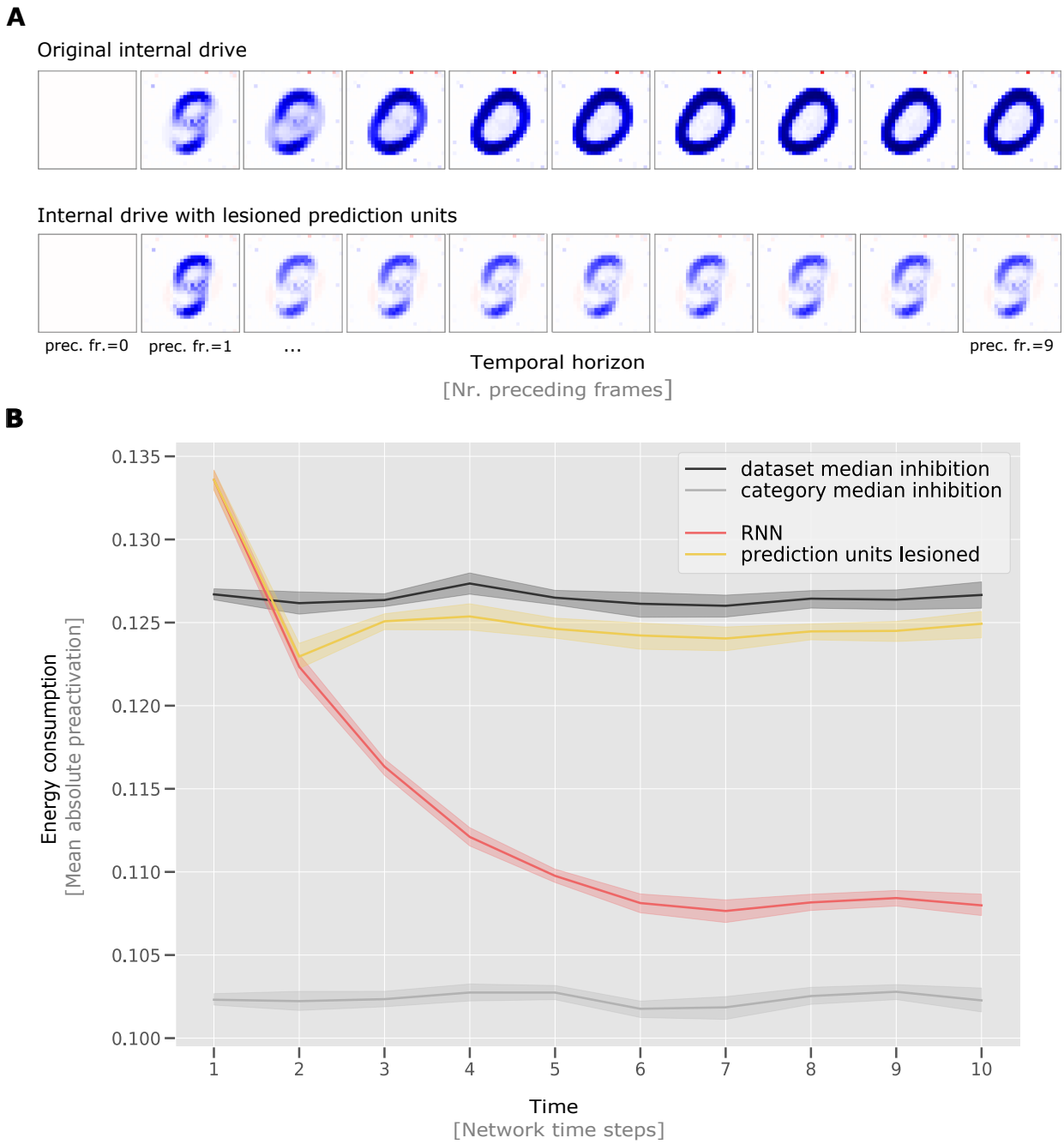

Figure 5: Evidence for two mechanisms of inhibitory predictions. **A:** internal drive (p_t) in the RNN for the '0' digit category at different points in time. Each square shows the internal drive of the network at a certain point in a sequence, e.g. at 'prec. fr.=2' we have the internal drive after the sequence '8-9' and at 'prec. fr.=5' we have the internal drive after the sequence '5-6-7-8-9'. The first row shows the predictions of a normal RNN, while the second row shows the result of lesioning RNN prediction units. Panel **B:** mean preactivations of normal RNN and lesioned RNNs. In contrast to normal RNNs, the lesioned RNNs do not reduce their preactivation over time, indicating that they are not capable of integrating evidence over time. Mean preactivations shown with 95% confidence intervals, bootstrapped across 10 network instances with replacement.

3 Discussion

Here, we demonstrate that recurrent neural network models, trained to minimise biologically realistic energy consumption (action potentials and synaptic transmissions), spontaneously develop hallmarks of predictive coding without the need to hardwire predictive coding principles into the network architecture. This opens up the possibility that predictive coding can be understood as a natural consequence of recurrent networks operating under a limited energy budget in predictive environments. In addition to category-specific inhibition and increasingly sharpened predictions with time, we observed that the RNNs self-organise into two distinct populations in line with error units and prediction units.

Beyond these processes, we observe two distinct mechanisms of prediction, a faster inhibitory loop among error units, and a slower predictive loop that enables RNNs to integrate information over longer timescales and thereby to generate increasingly fine-grained predictions. This observation can be interpreted as a rudimentary form of hierarchical

self-organisation in which the predictive units can be viewed as a higher-order cortical population operating on longer timescales and the error units as a lower-order cortical population operating on shorter time scales. This interpretation is consistent with hierarchical predictive coding architectures such as Rao and Ballard (1999) or Lotter et al. (2016) as well as the organisation of the brain in terms of a hierarchy of timescales (Kiebel et al., 2008; Baldassano et al., 2017).

It needs to be noted that the current experiments rely on MNIST digit sequences and therefore represent a proof of concept. Future work will explore larger-scale recurrent systems and their interplay with more natural image statistics. In terms of model details, future work could also explore energy efficiency in spiking network models where membrane voltages and spike-times must be considered. In addition, how a predictive system should optimally process differing spatial and temporal scales of dynamics and deals with unpredictable but information-less inputs (e.g. chaotic inputs and random noise) are key areas for future consideration.

In summary, our current set of findings suggests that predictive coding principles do not necessarily have to be hard-wired into the biological substrate but can be viewed as an emergent property of a simple recurrent system that minimises its energy consumption in a predictable environment. Notably, we have shown here that minimising unit preactivations, under mild assumptions of noise in the system, implies minimising both unit activity and synaptic transmission. This observation opens further interesting avenues of research into efficient coding and neural network modelling.

4 Materials and methods

4.1 Data generation

The input data are sequences of images drawn from the MNIST database of handwritten digits LeCun et al. (1998). Images are of size 28×28 with pixel intensities in the range $[0, 1]$. There are 60,000 images in the training set and 10,000 images in the test set. Each set of images can be divided into 10 categories; one for each digit. The frequency of each digit category in the dataset varies slightly (frequencies lie within 9% - 11% of the total dataset (70,000 samples)). Sequences are generated by choosing random digits as starting points. Digits from numerically subsequent categories are then randomly sampled and appended to the sequence until the desired sequence length is reached. The categories are wrapped-around such that after category 'nine' the sequence continues from category 'zero'. The sequence length can be chosen in advance, but all sequences are constrained to have the same length (in our simulations we take a sequence length of 10). The sequences are organised into batches. The images are drawn without replacement and the process is stopped when there are no image samples available for the next digit. Incomplete batches, in which there are no remaining image samples to complete a sequence, are discarded.

4.2 RNN architecture and training procedure

We created a recurrent neural network (RNN) consisting of $N = 784$ units (equivalent to the number of pixels in the input images) that are fully recurrently connected to each other. Each unit is driven by an input signal. The equations that determine the RNN dynamics are:

$$\mathbf{p_t} = \mathbf{W}\mathbf{h_{t-1}}\,, \tag{1}$$

$$\mathbf{a_t} = \mathbf{p_t} + \mathbf{x_t}\,, \tag{2}$$

$$\mathbf{h_t} = f\left(\mathbf{a_t}\right)\,, \tag{3}$$

where $\mathbf{x_t} \in \mathbb{R}^N$ denotes the input drive, $\mathbf{a_t} \in \mathbb{R}^N$ denotes the *preactivation*, $\mathbf{W} \in \mathbb{R}^{N\times N}$ denotes the recurrent weight matrix of the RNN (the only learnable parameters), $\mathbf{p_t} \in \mathbb{R}^N$ denotes the recurrent feedback, and $\mathbf{h_t} \in \mathbb{R}^N$ the unit outputs (in our study f is the ReLU non-linearity). The subscripts, t and $t-1$, refer to the discrete (integer) timestep of the system since all of these variables (aside from the weights) are iteratively updated. The weight matrix is uniformly initialised in $[-1, 1]$ and scaled by $N^{-\frac{1}{2}}$, i.e. proportionally scaled by the number of units in the weight matrix.

The objective of minimising energy is captured through the following loss function:

$$\ell = \frac{1}{NT}\sum_{t=1}^{T} ||\mathbf{a}_t||_1\,, \tag{4}$$

where T is the number of time steps, N is the number of units, and $\mathbf{a}_t$ is the preactivation of the units when processing the t-th element (or timestep) in a given sequence. We trained 10 model instances for 200 epochs with batch-size 32 (32 sequences per batch) with the Adam optimiser ($\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate = 0.0001) (Kingma and Ba, 2014). Model training differed in weight initialisation and training sequence. We tested these same 10 model instances (lesioned models in figure 5 panel B) but with unit activity $\mathbf{h_{t-1}}$ masked such that the feedback driven units did not affect the computation of $\mathbf{a_t}$. No extensive hyperparameter search was performed. The models were trained using PyTorch 1.4.0 and Python 3.7.6.

4.3 Derivation and calculation of bounds on preactivation

To benchmark how well the model minimises preactivation we derived three reference measures. These were: the input activity alone (i.e. no prediction), the pixel-wise median of the entire dataset removed from each image, and the pixel-wise median by digit category removed from each image. To relate these benchmarks to prediction, consider that the loss function, Equation 4, measures the L1-norm of the preactivation. This preactivation is composed of the input drive and some recurrent input. Now consider that this recurrent input can be an inhibition (i.e. negatively signed) which cancels the input drive and minimises this loss. Given this consideration, we can state the three benchmarks for performance, e_1, e_2, e_3, as (inhibitory feedback) prediction of three forms:

$$e_1 = ||\mathbf{x}_t||_1 \,, \tag{5}$$

$$e_2 = \; ||\mathbf{x}_t - \mu^{\frac{1}{2}}(\mathbf{X})\,||_1 \,, \tag{6}$$

$$e_3 = ||\mathbf{x}_t - \mu^{\frac{1}{2}}\left(\mathbf{X}^{(c)}\right)||_1 \,, \tag{7}$$

where $\mathbf{X} \in \mathbb{R}^{N \times M}$ is the entire (MNIST) dataset (of M digits), $X^{(c)} \in \mathbb{R}^{N \times M^{(c)}}$ denotes the $M^{(c)}$ data points that belong to (image) category c and $\mu^{\frac{1}{2}}(\mathbf{X}) \in \mathbb{R}^N$ denotes the pixel-wise median of a dataset $\mathbf{X}$. In this formulation, e_1 reflects the loss computed based upon input activity induced by $\mathbf{x}_t$ without any attempt of suppression by the RNN, e_2 is the loss based upon residue activity left after subtracting the (pixel-wise) median intensity of the entire dataset, and e_3 is the loss based upon residue activity left after subtracting the (pixel-wise) median intensity of that image category c. To understand why these specific (pixel-wise median) benchmarks make sense we have to understand what the optimal behaviour is under the loss function.

Our loss function, Equation 4, is an L1 norm on the preactivation of the units of our network. This preactivation consists of a (fixed) input drive which is modified by recurrent inputs. We can therefore consider, what form of recurrent contribution would minimise this L1 norm. Lets suppose that our model (and therefore recurrent contribution) has perfect knowledge of the category of the upcoming input, labelled below c. Disregarding the temporal element of our RNN, and considering minimisation of a single timestep input, let us label $\mathbf{b} \in \mathbb{R}^N$ as our estimation of the recurrent (inhibitory) contribution. The loss, summed across all samples of the category c, is given by:

$$\ell_c = \sum_{m=1}^{M^c} ||\mathbf{x}_m^{(c)} - \mathbf{b}||_1 \,, \tag{8}$$

where m now indexes the set of examples of stimuli in category c (stimuli labelled $\mathbf{x}_m^{(c)}$) of which there are a total of $M^{(c)}$ samples. Note that this summing is required since despite knowledge of the category c's upcoming presentation, the specific sample from within this category is unknown and all samples are equally likely. Thus, to minimise the loss across all possible outcomes we minimise their summation.

The value for the recurrent contribution ($\mathbf{b}$) that minimises this equation is given by:

$$\nabla_{\mathbf{b}} \ell_c = \sum_{m=1}^{M^{(c)}} \mathrm{sgn}\left(\mathbf{x}_m^{(c)} - \mathbf{b}\right) = \mathbf{0} \,. \tag{9}$$

This equation is satisfied when the number of cases in which the data is greater than the recurrent contribution ($\mathbf{x}_m^{(c)} > \mathbf{b}$) is precisely equivalent to the number of cases in which the data is smaller than the recurrent contribution ($\mathbf{x}_m^{(c)} < \mathbf{b}$). This must be true at a pixel-wise level (i.e. for each element of the vectors $\mathbf{x}_m^{(c)}$ and $\mathbf{b}$). These inequalities are consistent when our estimation is equal to the the median pixel value (per pixel) of the respective category in the dataset ($\mathbf{b} = \mu^{\frac{1}{2}}(\mathbf{X}^C)$). Hence, this is the prediction which would minimise our L1 loss function given the constraint of random sampling of images from each category (our benchmark e_3 above). The above derivation supposed that the model had perfect knowledge of the category of upcoming samples. However, if the model was not capable of identifying which category it should expect, the best it could do would be to predict the median of the entire dataset. This forms another of our benchmarks on performance (e_2).

The model preactivations in Figures 2 and 5 panel B are averaged over 891 sequence presentations from the test data. The benchmarks that we have produced here, e_1, e_2, e_3, are computed over the entire training dataset (60,000 MNIST Images), or per category (as in e_2), and the outcome of these estimations are applied to the individual digits in each of the (891) sequences and averaged by time-step. Each data point of model preactivation represents the mean over 10 model instances. Error bars are determined with 95% confidence intervals, bootstrapped with replacement over 10000 repetitions. Each data point of the theoretical bounds is likewise determined over 10 model instances with 95% confidence intervals, bootstrapped with replacement over 10000 repetitions. The variance in the bounds is due to the variation in input sequence generation and presentation across the 10 models. The variation for model preactivation is in addition to that driven by variation in weight initialisation.

4.4 Determination of predictive and error units

In order to determine whether units were primarily driven by feedback or input in Figure 4 we divided the test set into $K = 891$ sequences of length $T = 10$ and processed them through the trained RNNs. For a particular unit n, we

calculated the input drive $\langle x_n^+ \rangle$ and recurrent feedback drive $\langle p_n^+ \rangle$ by taking the rectifier of the input and feedback at the final time point and averaging them over the sequences

$$\langle x_n^+ \rangle = \frac{1}{K} \sum_k^K \max\left(0, x_n^{k_T}\right) , \tag{10}$$

$$\langle p_n^+ \rangle = \frac{1}{K} \sum_k^K \max\left(0, p_n^{k_T}\right) , \tag{11}$$

where $x_n^{k_T}$ is the value of the pixel that drives the unit in the k-th sequence at time point T and $p_n^{k_T}$ is the activity induced by reciprocal connectivity in the k-th sequence at time point T. We classified a unit as input driven if $[\langle x_n^+ \rangle - \langle p_n^+ \rangle] > 0$ and feedback driven if $[\langle x_n^+ \rangle - \langle p_n^+ \rangle] < 0$.

5 Acknowledgements

The authors are thankful to Courtney J Spoerer for feedback and helpful discussions during an early phase of the project.

6 Data availability

All data and analysis code along with instructions is available at:
`https://github.com/KietzmannLab/EmergentPredictiveCoding`

References

Alink A, Schwiedrzik CM, Kohler A, Singer W, Muckli L. Stimulus predictability reduces responses in primary visual cortex. Journal of Neuroscience. 2010; 30(8):2960–2966.

Baldassano C, Chen J, Zadbood A, Pillow JW, Hasson U, Norman KA. Discovering event structure in continuous narrative perception and memory. Neuron. 2017; 95(3):709–721.

Barlow HB, et al. Possible principles underlying the transformation of sensory messages. Sensory communication. 1961; 1(01).

Bastos AM, Usrey WM, Adams RA, Mangun GR, Fries P, Friston KJ. Canonical microcircuits for predictive coding. Neuron. 2012; 76(4):695–711.

Bell AJ, Sejnowski TJ. An information-maximization approach to blind separation and blind deconvolution. Neural computation. 1995; 7(6):1129–1159.

van Bergen RS, Kriegeskorte N. Going in circles is the way forward: the role of recurrence in visual inference. Current Opinion in Neurobiology. 2020; 65:176–193.

Berkes P, Wiskott L. Slow feature analysis yields a rich repertoire of complex cell properties. Journal of vision. 2005; 5(6):9–9.

Bialek W, Van Steveninck RRDR, Tishby N. Efficient representation as a design principle for neural coding and computation. In: *2006 IEEE international symposium on information theory* IEEE; 2006. p. 659–663.

Chalasani R, Principe JC. Deep predictive coding networks. arXiv preprint arXiv:13013541. 2013; .

Chalk M, Marre O, Tkačik G. Toward a unified theory of efficient, predictive, and sparse coding. Proceedings of the National Academy of Sciences. 2018; 115(1):186–191.

Clark A. Whatever next? Predictive brains, situated agents, and the future of cognitive science. Behavioral and brain sciences. 2013; 36(3):181–204.

De Lange FP, Heilbron M, Kok P. How do expectations shape perception? Trends in cognitive sciences. 2018; 22(9):764–779.

Dijkstra N, Ambrogioni L, Vidaurre D, van Gerven M. Neural dynamics of perceptual inference and its reversal during imagery. Elife. 2020; 9:e53588.

Eckmann S, Klimmasch L, Shi BE, Triesch J. Active efficient coding explains the development of binocular vision and its failure in amblyopia. Proceedings of the National Academy of Sciences. 2020; 117(11):6156–6162.

Ekman M, Kok P, de Lange FP. Time-compressed preplay of anticipated events in human primary visual cortex. Nature Communications. 2017; 8(1):1–9.

Friston K. A theory of cortical responses. Philosophical transactions of the Royal Society B: Biological sciences. 2005; 360(1456):815–836.

Friston K. The free-energy principle: a unified brain theory? Nature reviews neuroscience. 2010; 11(2):127–138.

Hupé J, James A, Payne B, Lomber S, Girard P, Bullier J. Cortical feedback improves discrimination between figure and background by V1, V2 and V3 neurons. Nature. 1998; 394(6695):784–787.

Kaas JH. Sensory loss and cortical reorganization in mature primates. Progress in brain research. 2002; 138:167–176.

Kar K, Kubilius J, Schmidt K, Issa EB, DiCarlo JJ. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. Nature neuroscience. 2019; 22(6):974–983.

Kiebel SJ, Daunizeau J, Friston KJ. A hierarchy of time-scales and the brain. PLoS Comput Biol. 2008; 4(11):e1000209.

Kietzmann TC, Spoerer CJ, Sörensen LK, Cichy RM, Hauk O, Kriegeskorte N. Recurrence is required to capture the representational dynamics of the human visual system. Proceedings of the National Academy of Sciences. 2019; 116(43):21854–21863.

Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014; .

Kok P, Jehee JF, De Lange FP. Less is more: expectation sharpens representations in the primary visual cortex. Neuron. 2012; 75(2):265–270.

LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86(11):2278–2324.

Lee TS, Mumford D. Hierarchical Bayesian inference in the visual cortex. JOSA A. 2003; 20(7):1434–1448.

Linsley D, Kim J, Ashok A, Serre T. Recurrent neural circuits for contour detection. arXiv preprint arXiv:201015314. 2020; .

Lotter W, Kreiman G, Cox D. Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:160508104. 2016; .

Lotter W, Kreiman G, Cox D. A neural network trained for prediction mimics diverse features of biological neurons and perception. Nature Machine Intelligence. 2020; 2(4):210–219.

Mehrer J, Spoerer CJ, Kriegeskorte N, Kietzmann TC. Individual differences among deep neural network models. Nature communications. 2020; 11(1):1–12.

Merabet LB, Pascual-Leone A. Neural reorganization following sensory loss: the opportunity of change. Nature Reviews Neuroscience. 2010; 11(1):44–52.

Mumford D. On the computational architecture of the neocortex. Biological cybernetics. 1992; 66(3):241–251.

Murray SO, Kersten D, Olshausen BA, Schrater P, Woods DL. Shape perception reduces activity in human primary visual cortex. Proceedings of the National Academy of Sciences. 2002; 99(23):15164–15169.

Näätänen R, Tervaniemi M, Sussman E, Paavilainen P, Winkler I. 'Primitive intelligence'in the auditory cortex. Trends in neurosciences. 2001; 24(5):283–288.

Olshausen BA, Field DJ. Natural image statistics and efficient coding. Network: computation in neural systems. 1996; 7(2):333–339.

Rao HM, Mayo JP, Sommer MA. Circuits for presaccadic visual remapping. Journal of Neurophysiology. 2016; 116(6):2624–2636.

Rao RP, Ballard DH. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature neuroscience. 1999; 2(1):79–87.

Schwiedrzik CM, Freiwald WA. High-level prediction signals in a low-level area of the macaque face-processing hierarchy. Neuron. 2017; 96(1):89–97.

Sengupta B, Stemmler M, Laughlin SB, Niven JE. Action potential energy efficiency varies among neuron types in vertebrates and invertebrates. PLoS Comput Biol. 2010; 6(7):e1000840.

Spoerer CJ, Kietzmann TC, Mehrer J, Charest I, Kriegeskorte N. Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. PLoS computational biology. 2020; 16(10):e1008215.

Spoerer CJ, McClure P, Kriegeskorte N. Recurrent convolutional neural networks: a better model of biological object recognition. Frontiers in psychology. 2017; 8:1551.

Squires NK, Squires KC, Hillyard SA. Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man. Electroencephalography and clinical neurophysiology. 1975; 38(4):387–401.

Srinivasan MV, Laughlin SB, Dubs A. Predictive coding: a fresh view of inhibition in the retina. Proceedings of the Royal Society of London Series B Biological Sciences. 1982; 216(1205):427–459.

Summerfield C, Trittschuh EH, Monti JM, Mesulam MM, Egner T. Neural repetition suppression reflects fulfilled perceptual expectations. Nature neuroscience. 2008; 11(9):1004.

Summerfield C, Wyart V, Mareike Johnen V, De Gardelle V. Human scalp electroencephalography reveals that repetition suppression varies with expectation. Frontiers in Human Neuroscience. 2011; 5:67.

Villegas R, Yang J, Zou Y, Sohn S, Lin X, Lee H. Learning to generate long-term future via hierarchical prediction. In: *international conference on machine learning* PMLR; 2017. p. 3560–3569.

Appendix A: supplementary figures

Supplementary figures accompanying *Predictive coding emerges in energy efficient recurrent neural networks*.

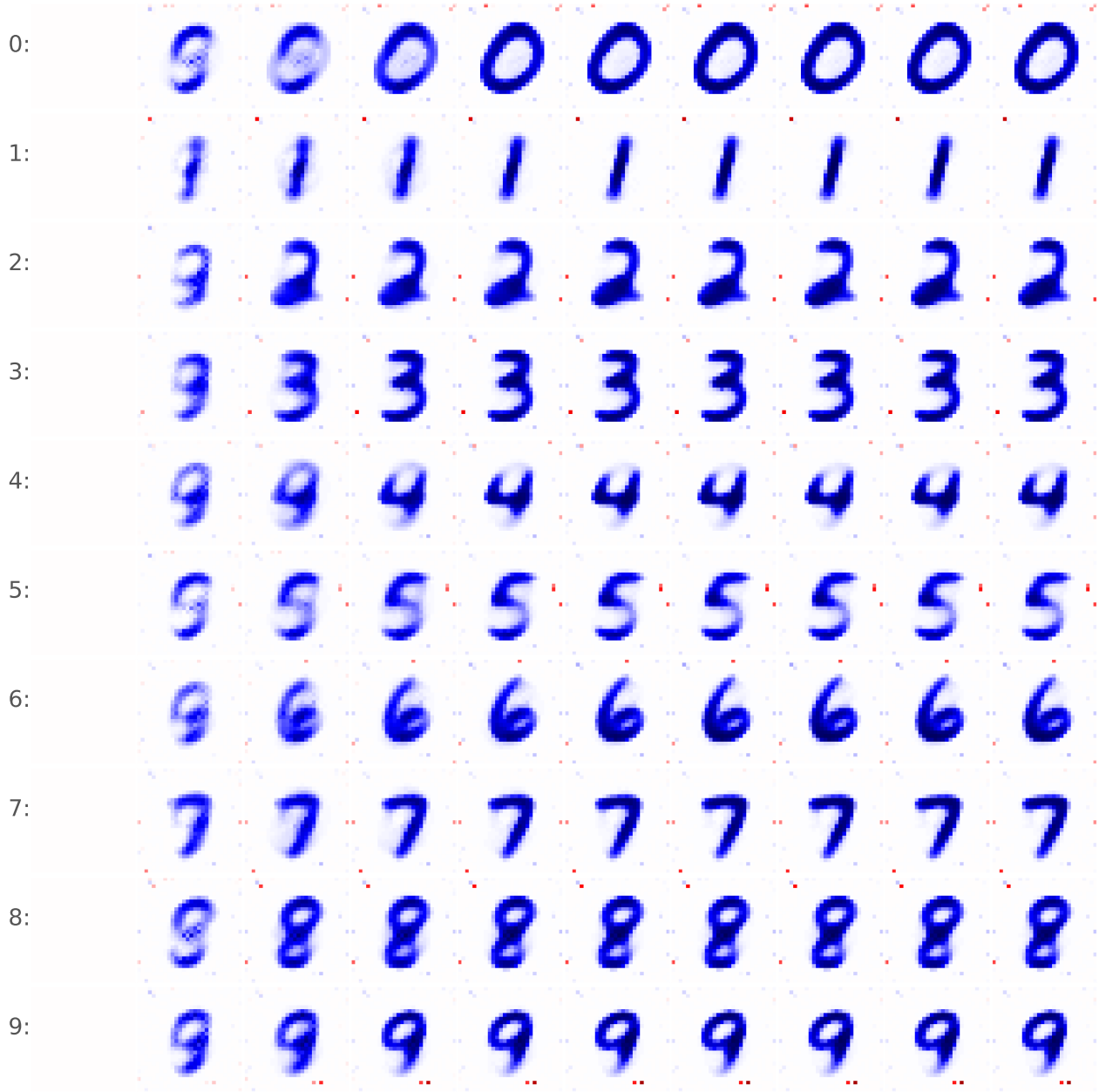

Figure A1: Recurrent feedback ($\mathbf{p_t}$) in the RNN for different input categories at different points in time. Each row shows the recurrent feedback at the step with a specific input category will be expected. Each column shows the recurrent feedback after a different number of preceding images in the sequence. The inhibitory effect (i.e. predictions) gets more pronounced as sequences progress.

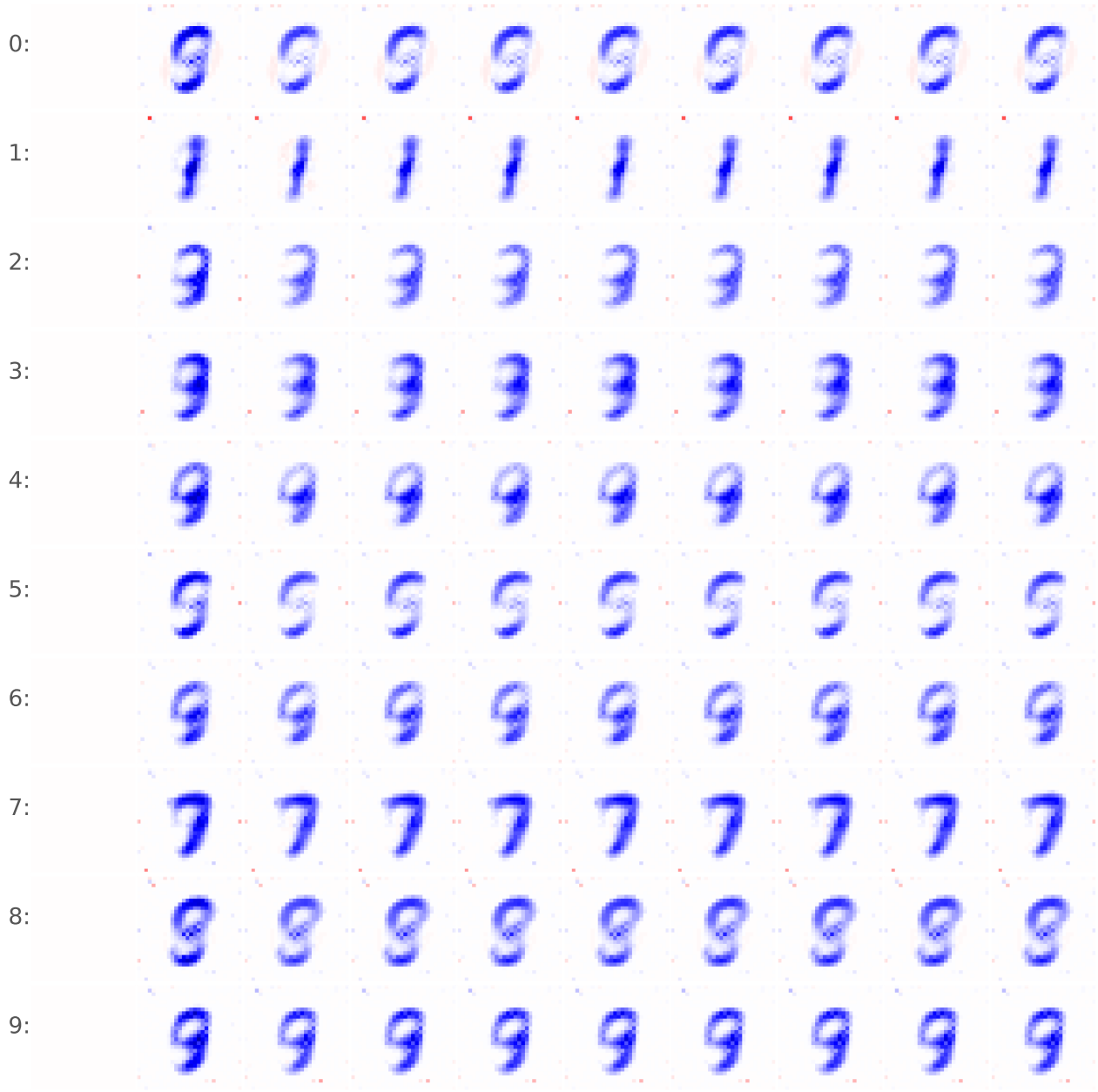

Figure A2: Recurrent feedback ($\mathbf{p_t}$) in the lesioned RNN model for different input categories at different points in time. Each row shows the recurrent feedback at the step with a specific input category will be expected. Each column shows the recurrent feedback after a different amount of preceding images in the sequence. The inhibitory effect with lesioned prediction units is weak and remains constant throughout the sequence.

Appendix B: theoretical motivation for preactivation

Supplementary material accompanying *Predictive coding emerges in energy efficient recurrent neural networks*.

This supplement provides a derivation for the claim in the main manuscript that minimising preactivation leads to smaller weights and thus synaptic transmission. We do this by analysing what weight changes are induced in the case where noise causes predictions to deviate from the optimal predictions.

Parameter updates induced by our L1 loss function

Here we consider the form of weight updates for the system described in this paper (see Materials and methods for extended description).

In short, consider a dataset of M samples, each of dimension N, $\mathbf{X} \in \mathbb{R}^{M \times N}$, where the individual samples, indexed m, are labelled $\mathbf{x_m} \in \mathbb{R}^N$. The individual units of our system are each driven by a single element of some sampled input data and also receive some feedback from recurrent projections. Thus, we can formulate the preactivation of a given unit, indexed i in our system as $x_{mi} - b_i$, where b_i is the recurrent feedback computed as $b_i = \sum_j w_{ji} h_j$. The loss function of our system aims to minimise the L1 norm of this preactivation. We are interested in the form of the weight updates over all possible input drives which might appear (note the specific input stimulus is randomly chosen from within a category for our main paper).

Given this setup, we can frame an L1 loss function (which is used to train our system) for this particular unit as:

$$\ell_i = \sum_m^M |x_{mi} - b_i| \,. \tag{1}$$

This L1 loss is computed over all possible input samples following some network dynamic up to that moment. The minimum of this equation is reached when the derivative of this equation is 0. That is,

$$\frac{d\ell_i}{db_i} = \frac{d}{db_i} \sum_m^M |x_{mi} - b_i| = 0 \,. \tag{2}$$

If we rewrite $|x_{mi} - b_i|$ as $\sqrt{\left(x_{mi} - b_i\right)^2}$ We can write out the derivative as:

$$\frac{d}{db_i}\left(\left(x_{mi} - b_i\right)^2\right)^{(1/2)} = \frac{d}{du} u^{(1/2)} \frac{du}{db_i} = \frac{d}{du} u \frac{du}{dv} \frac{dv}{db_i} \,, \tag{3}$$

where $u = \left(x_{mi} - b_i\right)^2$, and $v = x_{mi} - b_i$. We obtain

$$\frac{d\ell_i}{db_i} = -\sum_m^M \frac{2\left(x_{mi} - b_i\right)}{2\sqrt{\left(x_{mi} - b_i\right)^2}} = -\sum_m^M \frac{\left(x_{mi} - b_i\right)}{|\left(x_{mi} - b_i\right)|} = -\sum_m^M \operatorname{sgn}\left(x_{mi} - b_i\right) \tag{4}$$

and the derivative, $\frac{d\ell_i}{db_i}$, would be at zero (and therefore the loss function at a minimum) when we have an equal number of cases in which the feedback projection is greater than versus less than the data (i.e. the sum over signs would be zero). This would be achieved by setting the recurrent feedback equal to the median value of the entire input datasets, $b_i = \mu^{1/2}(\mathbf{X}_i)$, where $\mathbf{X}_i$ is the set of all ith elements of our samples, $\mathbf{X}_i \in \mathbb{R}^{M \times 1}$.

In addition to this analysis we can also compute the derivative with respect to a specific recurrent projecting weights (from a pre-synaptic unit indexed j) as:

$$\frac{\partial \ell_i}{\partial w_{ji}} = \frac{d\ell_i}{db_i} \frac{\partial b_i}{\partial w_{ji}} = -\sum_m^M \operatorname{sgn}\left(x_{mi} - b_i\right) h_j = -h_j \sum_m^M \operatorname{sgn}\left(x_{mi} - b_i\right) \,. \tag{5}$$

Note our ability to extract the recurrent activation, h_j, from the summation since we assume the network past is fixed and we are computing its loss over all possible 'next' samples.

The effect of noise in the recurrence

Ideally $b_i = \mu^{1/2}\left(X_i\right)$, but due to noise in the recurrent prediction, b_i can diverge from this optimum. We will describe below how the loss will be minimised in this instance by first rewriting the recurrence into its positive and negative terms. Remember that we formulate our parameter b_i as being determined by the transformation of some hidden state $\mathbf{h}$ by a set of weights $\{w_{ji}\}$, $b_i = \sum_j w_{ji} h_j$. In order to separate this term into positive and negative contributions, we make use of the Heaviside function

$$H(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

This allows us to simply define

$$b_i = \sum_z H\left(w_{zi}h_z\right) w_{zi}h_z + \sum_z H\left(-w_{zi}h_z\right) w_{zi}h_z \,. \tag{7}$$

Let us now take the case in which there is noise in our recurrent activation, h_j, such that $\tilde{h_j} = h_j + \epsilon_j$. Therefore, we can also re-define our recurrent prediction as

$$\begin{aligned}
\tilde{b}_i &= \sum_z H\left(w_{zi}h_z\right) w_{zi}\tilde{h_z} + \sum_z H\left(-w_{zi}h_z\right) w_{zi}\tilde{h_z} && (8)\\
&= \sum_z H\left(w_{zi}h_z\right) w_{zi}\left(h_z + \epsilon_z\right) + \sum_z H\left(-w_{zi}h_z\right) w_{zi}\left(h_z + \epsilon_z\right) && (9)\\
&= b_i + \sum_z H\left(w_{zi}h_z\right) w_{zi}\epsilon_z + \sum_z H\left(-w_{zi}h_z\right) w_{zi}\epsilon_z \,, && (10)
\end{aligned}$$

i.e. we have noise which impacts our estimation of b_i. Note now that for the purposes of our derivation here, we separate the positive and negative terms (through the Heaviside function) based upon the noiseless values of recurrent unit activities. We can now discern a few possible situations, specifically symmetric and asymmetric noise.

Symmetric noise One simple outcome is that the effect of the noise is equal on the components , i.e.:

$$\sum_z H\left(w_{zi}h_z\right) w_{zi}\epsilon_z = -\sum_z H\left(-w_{zi}h_z\right) w_{zi}\epsilon_z \,. \tag{11}$$

This is a special condition in which all the effects of noise are precisely cancelled such that $\tilde{b} \to b$. It is unlikely that our network dynamics would reliably inhabit this regime.

Asymmetric noise In the case of asymmetric noise we can consider two key conditions,

$$\sum_z H\left(w_{zi}h_z\right) w_{zi}\epsilon_z > -\sum_z H\left(-w_{zi}h_z\right) w_{zi}\epsilon_z \tag{12}$$

or

$$\sum_z H\left(w_{zi}h_z\right) w_{zi}\epsilon_z < -\sum_z H\left(-w_{zi}h_z\right) w_{zi}\epsilon_z \,. \tag{13}$$

Both of these cases cause deviation of our estimated parameter $\tilde{b}$, from the desired median value, b_i. For an optimal value of b_i, such deviations shifts our loss away from its minimum and the greater the deviation, the greater the increase in loss. Thus, since the deviation magnitude is only dependent upon the scale of the weights and the random noise (see the factor of w_{zi}), we would expect the loss to increase in proportion to the scale of the weights. This alone suggests that minimal weight values would reduce the magnitude of the loss. However it could also be that selected weights increase instead in order to find a solution in which errors on the left and right are balanced. Thus, we must further analyse the specific weight changes induced by noise.

Expressing weight update dependence upon noise

Given our suggestion above (that the magnitude of our loss function in the face of noise is related to the scale of the weights), it is important to show the precise relationship between the impact of noise and the preferred direction of weight change. For this purpose, we can take the derivative of our noise-affected loss function with respect to the synaptic weights. We can express the derivative of our loss function with respect to some specific weight, w_{ji} as

$$\frac{\partial \ell_i}{\partial w_{ji}} = \frac{d\ell}{d\tilde{b}_i}\frac{\partial \tilde{b}_i}{\partial w_{ji}} = -\sum_m^M \operatorname{sgn}\left(x_{mi} - \tilde{b}_i\right)\left(h_j + \epsilon_j\right) = -\left(h_j + \epsilon_j\right)\sum_m^M \operatorname{sgn}\left(x_{mi} - \tilde{b}_i\right) \,. \tag{14}$$

We can further expand this into the terms causing deflections from the noise-less prediction (b_i) such that

$$\frac{\partial \ell_i}{\partial w_{ji}} = -\left(h_j + \epsilon_j\right)\sum_m^M \operatorname{sgn}\left(x_{mi} - \tilde{b}_i\right) = -\left(h_j + \epsilon_j\right)\sum_m^M \operatorname{sgn}\left(x_{mi} - b_i - \sum_z w_{zi}\epsilon_z\right) \,. \tag{15}$$

This expression can further be separated by considering our sum of signs as composed of multiple sign summations. In particular, we can consider this equivalent to

$$\frac{\partial \ell_i}{\partial w_{ji}} = -\left(h_j + \epsilon_j\right)\sum_m^M \left(\operatorname{sgn}\left(x_{mi} - b_i\right) - 2S_{mi}\operatorname{sgn}\left(\sum_z w_{zi}\epsilon_z\right)\right), \tag{16}$$

where $S_{mi} = |H\left(\left(x_{mi} - \tilde{b}_i\right)\right) - H\left(\left(x_{mi} - b_i\right)\right)|$ is a binary variable indicating if the sign of $x_{mi} - b_i$ was different to $x_{mi} - \tilde{b}_i$. With this redefinition, we can separate updates to weights caused by an error in the underlying prediction (b_i) and an error caused by sign-flips through noise. In particular,

$$\frac{\partial \ell_i}{\partial w_{ji}} = -(h_j + \epsilon_j) \sum_m^M \text{sgn}\left(x_{mi} - b_i\right) + \left(h_j + \epsilon_j\right) \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z\Big) \tag{17}$$

where the first term is precisely equivalent to the noiseless update aside from an additional factor of noise.

In the case of ideal prediction ($b_i = \mu^{1/2}(\mathbf{X}_i)$), the number of positive signs and number of negative signs are equivalent and therefore this initial term is zero. Let us therefore exclude from our further analysis the first term of this loss function. We can ignore it for two reasons: it is an update only relevant for learning the task and in the case of an ideal prediction has a value of zero. Thus we are left with a term involving the impact of noise upon predictions. In particular,

$$\frac{\partial \ell_i}{\partial w_{ji}} = \left(h_j + \epsilon_j\right) \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z\Big)\,. \tag{18}$$

This reformulation is illuminating as it shows that the sign change does not depend upon the data (x_{mi}), but is a fixed sign change that will occur a specific number of times (the number of times that $S_n = 1$). The number of sign changes therefore depends entirely on the noise (ϵ) and the weights.

To progress further, we must assume sampling over the noise (as well as over the data). In particular, let us consider the weight update for this unit carried out over a set of noise samples, such that we can describe the change in weight as

$$\Delta w_{ji} = -\frac{1}{K}\sum_k^K \frac{d\ell_i}{d\tilde{b}_i^k}\frac{\partial \tilde{b}_i^k}{\partial w_{ji}} \tag{19}$$

$$= -\frac{1}{K}\sum_k^K \left(h_j + \epsilon_j^k\right)\left(\sum_m^M \text{sgn}\left(x_{mi} - \tilde{b}_i^{\,k}\right)\right) \tag{20}$$

$$= -\frac{1}{K}\sum_k^K \left(h_j + \epsilon_j^k\right)\left(\sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big)\right) \tag{21}$$

where we sample K instances of noise, each instance identified ϵ_j^k (indexed k).

Separating terms

Given our new formulation of sampled noise, we can now further remove the impact of the activation of recurrent units. In particular, consider separating the terms involving noise and recurrent activity such that

$$\Delta w_{ji} = -\frac{1}{K}\sum_k^K (h_j + \epsilon_j^k)\left(\sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big)\right) \tag{22}$$

$$= -\frac{1}{K}\sum_k^K h_j \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big) - \frac{1}{K}\sum_k^K \epsilon_z^k \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_j^k\Big). \tag{23}$$

Considering the first term in this separation, we have presented a factor determined by recurrent activation (h_j). This term can be reframed (over the infinite sampling of noise) in terms of an expectation such that we have an expectation over the sign changes induced by noise and the recurrent activation (these are independent terms)

$$\frac{1}{K}\sum_k^K h_j \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big) = h_j\langle\sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big)\rangle_k\,. \tag{24}$$

Since the terms involving noise and recurrent activation are independent (and the recurrent activation is fixed), we could take the expectation of the noise affected term separately. Furthermore, in the limit of large sampling of noise, and assuming that the noise has zero mean (i.e. does not introduce a net bias), our expectation over these signs is zero ($\langle\sum_m^M 2S_{mi}\ \text{sgn}\big(\sum_z w_{zi}\epsilon_z^k\big)\rangle_k = 0$). Thus this term can be ignored . This allows us to simplify our above formulation to ignore this term such that

$$\Delta w_{ji} = -\frac{1}{K}\sum_k^K \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big)h_j - \frac{1}{K}\sum_k^K \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big)\epsilon_j^k \tag{25}$$

$$\approx -\frac{1}{K}\sum_k^K \sum_m^M 2S_{mi}\ \text{sgn}\Big(\sum_z w_{zi}\epsilon_z^k\Big)\epsilon_j^k\,. \tag{26}$$

Now we can finally consider the impacts of noise alone upon the weight changes to our recurrent synaptic connections. Since the sign of our noise is a key factor which modifies the direction of weight change, we separate the terms involving cases of positive and negative noise such that

$$\Delta w_{ji} \approx -\frac{1}{K}\sum_{k}^{K}\epsilon_j^k \sum_{m}^{M} 2S_{mi}\ \mathrm{sgn}\big(\sum_{z} w_{zi}\epsilon_z^k\big) \tag{27}$$

$$\approx -\frac{2}{K}\sum_{k}^{K}\left(\epsilon_j^k H\left(\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(\sum_{z} w_{zi}\epsilon_z^k\big) + \epsilon_j^k H\left(-\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(\sum_{z} w_{zi}\epsilon_z^k\big)\right) \tag{28}$$

$$\approx -\frac{2}{K}\sum_{k}^{K}\left(\epsilon_j^k H\left(\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(w_{ji}\epsilon_j^k + \sum_{z\neq j} w_{zi}\epsilon_z^k\big) + \epsilon_j^k H\left(-\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(w_{ji}\epsilon_j^k + \sum_{z\neq j} w_{zi}\epsilon_z^k\big)\right). \tag{29}$$

This expression separates the terms which involve cases of positive/negative ϵ_j^k through filtering by the Heaviside function. Furthermore, it makes clear that the arguments of the sign function are composed of terms involving the pre-synaptic unit j and those involving all other units. Unless we now make assumptions regarding the distribution of the data and noise we cannot simplify these expressions further. However we can say something about the sign of weight change in a number of cases.

A single recurrent synaptic connection First, if we consider a case in which there is only a single recurrent input, h_j, then our formulation above simplifies significantly. In particular, $\sum_z w_{zi}\epsilon_z^k = w_{ji}\epsilon_j^k$ and therefore

$$\Delta w_{ji} \approx -\frac{2}{K}\sum_{k}^{K}\left(\epsilon_j^k H\left(\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\left(w_{ji}\epsilon_j^k\right) + \epsilon_j^k H\left(-\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\left(w_{ji}\epsilon_j^k\right)\right). \tag{30}$$

In this case, without the confound of other recurrent synaptic connections, we can immediately make statements regarding the sign of the weight change.

Consider if the noise is sufficient to create a sign flip (this would always be true for noise in a case in which we have an odd number of datapoints, i.e. the median lies upon a datapoint). Then we can consider what our weight update equation would be in two separate cases, one in which the current synaptic weight is positive and one in which it is negative such that

$$\text{if } w_{ji} > 0\text{: then } \sum_{m}^{M} S_{mi}\ \mathrm{sgn}\left(w_{ji}\epsilon_j^k\right) \propto \mathrm{sgn}\left(\epsilon_j^k\right). \quad \text{Assuming } \sum_{m}^{M} S_{mi}\ \mathrm{sgn}\left(w_{ji}\epsilon_j^k\right) = \mathrm{sgn}\left(\epsilon_j^k\right),$$

$$\Delta w_{ji} \approx -\frac{2}{K}\sum_{k}^{K}\left(|\epsilon_j^k| H\left(\epsilon_j^k\right) + |\epsilon_j^k| H\left(-\epsilon_j^k\right)\big)\right) = -\frac{2}{K}\sum_{k}^{K}|\epsilon_j^k|$$

$$\text{if } w_{ji} < 0\text{: then } \sum_{m}^{M} S_{mi}\ \mathrm{sgn}\left(w_{ji}\epsilon_j^k\right) \propto -\mathrm{sgn}\left(\epsilon_j^k\right). \quad \text{Assuming } \sum_{m}^{M} S_{mi}\ \mathrm{sgn}\left(w_{ji}\epsilon_j^k\right) = -\mathrm{sgn}\left(\epsilon_j^k\right),$$

$$\Delta w_{ji} \approx -\frac{2}{K}\sum_{k}^{K}\left(-|\epsilon_j^k| H\left(\epsilon_j^k\right) - |\epsilon_j^k| H\left(-\epsilon_j^k\right)\big)\right) = \frac{2}{K}\sum_{k}^{K}|\epsilon_j^k|$$

This above logic dictates that in the case of a single recurrent connections, the sign of the weight change is always in the direction of reduced magnitude, regardless of the noise magnitude or sign. In particular, if the weight, w_{ji} is positive, then any positive or negative noise will produce a net negative change in the synaptic weight. Conversely, if the weight w_{ji} is negative, then any positive or negative noise will produce a net positive change of the weight. This derivation shows that our L1 loss function combined with noise in a single unit recurrent system will ensure that aside from updates related to training the network task, noise induces a reduction in the magnitudes of weights.

The more complex case of multiple recurrent connections Though we above demonstrate that noise ensures a minimisation of the recurrent synaptic weights for a single unit, showing this for multiple recurrent units is rather more involved. Consider separating the positive and negative noise cases and taking an expectation over many samples, such that

$$\Delta w_{ji} \approx -\frac{2}{K}\sum_{k}^{K}\left(\epsilon_j^k H\left(\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(w_{ji}\epsilon_j^k + \sum_{z\neq j} w_{zi}\epsilon_z^k\big) + \epsilon_j^k H\left(-\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(w_{ji}\epsilon_j^k + \sum_{z\neq j} w_{zi}\epsilon_z^k\big)\right) \tag{31}$$

$$\approx -2\langle\epsilon_j^k H\left(\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(w_{ji}\epsilon_j^k + \sum_{z\neq j} w_{zi}\epsilon_z^k\big)\rangle_k - 2\langle\epsilon_j^k H\left(-\epsilon_j^k\right)\sum_{m}^{M} S_{mi}\ \mathrm{sgn}\big(w_{ji}\epsilon_j^k + \sum_{z\neq j} w_{zi}\epsilon_z^k\big)\rangle_k \tag{32}$$

$$\approx -\Delta^+ w_{ji} - \Delta^- w_{ji}\,. \tag{33}$$

Without making assumptions about the distribution of the noise and our data, we cannot provide a closed form solution for the magnitude of weight reduction. However, we can make some statements which provide implications for the sign of weight changes. In general it is true that $\epsilon_j^k H\left(\epsilon^k\right) \geq 0$ and $\epsilon_j^k H\left(-\epsilon^k\right) \leq 0$. This immediately ensures that the key factor which shall determine the sign of weight updates are the sign function components. Moreover, by virtue of the way how we group positive and negative noise terms we know that the expectation of of $\sum_{z \neq j} w_{zi}\epsilon_z^k$ will be zero when averaged over large K. This is since these terms are independent random variables (scaled by the respective weights). Importantly therefore we can say that ϵ_j^k and $\sum_m^M S_{mi}\ \mathrm{sgn}\left(w_{ji}\epsilon_j^k + \sum_{z \neq j} w_{zi}\epsilon_z^k\right)$ will be correlated positively/negatively when the synaptic weight w_{ji} is positive/negative. This directly provides evidence that the signs of $\Delta^+ w_{ji}$ and $\Delta^- w_{ji}$ will correlate to the sign of the weight, w_{ji}. Thus, given the weight change is composed of the negative sum of these components, it is implied that under the influence of recurrent noise the synaptic weights will tend towards smaller magnitudes. A reader should note that this derivation describes how network weights evolve due to noise alone and in practice this weight change will be summed with weight changes affecting task performance (see Equation 17).