Classifiers with limited connectivity

Lyudmila Kushnir[1,2], Stefano Fusi[2,3,4]

[1] GNT - LNC, Departement d'etudes cognitives, Ecole normale superieure,
INSERM, PSL Research University, 75005 Paris, France

[2] Center for Theoretical Neuroscience, College of Physicians and Surgeons, Columbia University

[3] Mortimer B. Zuckerman Mind Brain Behavior Institute, Columbia University

[4] Kavli Institute for Brain Sciences, Columbia University

July 2, 2017

Abstract

For many neural network models that are based on perceptrons, the number of activity patterns that can be classified is limited by the number of plastic connections that each neuron receives, even when the total number of neurons is much larger. This poses the problem of how the biological brain can take advantage of its huge number of neurons given that the connectivity is extremely sparse, especially when long range connections are considered. One possible way to overcome this limitation in the case of feed-forward networks is to combine multiple perceptrons together, as in committee machines. The number of classifiable random patterns would then grow linearly with the number of perceptrons, even when each perceptron has limited connectivity. However, the problem is moved to the downstream readout neurons, which would need a number of connections that is as large as the number of perceptrons. Here we propose a different approach in which the readout is implemented by connecting multiple perceptrons in a recurrent attractor neural network. We show with analytical calculations that the number of random classifiable patterns can grow unboundedly with the number of perceptrons, even when the connectivity of each perceptron remains finite. Most importantly both the recurrent connectivity and the connectivity of a downstream readout are also finite. Our study shows that feed-forward neural classifiers with numerous long range connections connecting different layers can be replaced by networks with sparse long range connectivity and local recurrent connectivity without sacrificing the classification performance. Our strategy could be used in the future to design more general scalable network architectures with limited connectivity, which resemble more closely brain neural circuits dominated by recurrent connectivity.

1 Introduction

The performance of a neural circuit is often evaluated by determining the number of input-output functions that can be implemented, or equivalently by the number of inputs that can be classified correctly by the neural circuit. Theoretical studies on perceptrons [1] and recurrent neural circuits (see e.g. [2]) have shown that typically the performance of a neural circuit scales with the number of synaptic connections that each individual neuron receives, and not with the total number of synapses, or with the total number of neurons (see e.g. [3]). This is clearly a problem in the biological brain in which the connectivity is sparse, especially when long range connections are considered [4]. One striking example is the mammalian hippocampal circuit [5]. According to [6] a typical pyramidal neuron in the CA3 area of rodent hippocampus receives only 50 synapses from the upstream area, the dentate gyrus (DG), which contains around 10^6 neurons. This is even more striking taking into account that the neural activity in the dentate gyrus is very sparse [7, 8].

One possible way to overcome the limitations of neural circuits with sparse connectivity is to consider the strategy of what are called ”committee machines” [9], which are basically populations of classifiers. Each classifier is weak, as a perceptron with limited connectivity would be. However, the committee machine can read out a large number of these weak classifiers, producing a response based on a majority vote or on some more sophisticated way of combining the multiple classifiers. The final classification performance is significantly better than the classification performance of each individual classifier, provided that the errors committed by the individual classifiers are sufficiently independent. The term committee machines goes back to 1960-s [9], but they have also been a focus of more recent studies (see e.g. [10], [11]). In the later reincarnation, they can be seen as a continuation of a more general idea which is recognized in machine learning as *ensemble methods* or *hypothesis boosting* [12, 13] and has been long known in statistics [14, 15, 16, 17]. Some of the examples include stacking [18, 19], bagging [20], arcing [21] and adaboost [22, 23].

One class of committee machines are implemented using populations of neurons, each essentially behaving as a neural classifier, like a perceptron [24, 25, 26, 27]. Each neural classifier can have limited connectivity, and hence it can be considered as a weak classifier, if taken individually. A particularly notable study [25] shows that it is possible to compute the classification capacity when each neural classifier has sparse connectivity. The connections between the N input neurons and the $M < N$ neural classifiers were assumed to be non-overlapping (N/M connections per ”perceptron”) and plastic. The final response of the committee machine is obtained by majority vote of the M neural classifiers, which can be easily implemented by introducing a readout neuron that is connected to all the neural classifiers with equal weights. The maximum number of inputs that can be correctly classified is proportional to $N\sqrt{\log M}$, whereas each neural classifier would not go beyond N/M inputs. This is a very favorable scaling and it is similar to the one obtained in other committee machines. However, one has to keep in mind that the neural classifiers can have sparse connectivity, but the readout neuron performing the majority vote should have a number of connections that scales with N.

Here we propose a network architecture that overcomes the restrictions imposed by the limited connectivity, as in the committee machines discussed above, but it replaces the readout neuron that has extensive connectivity with a more biologically plausible recurrent network in which all the neurons have a number of connections that remains finite even when the number of classifiable inputs grows unboundedly. More specifically we show that the number of random inputs that can be correctly

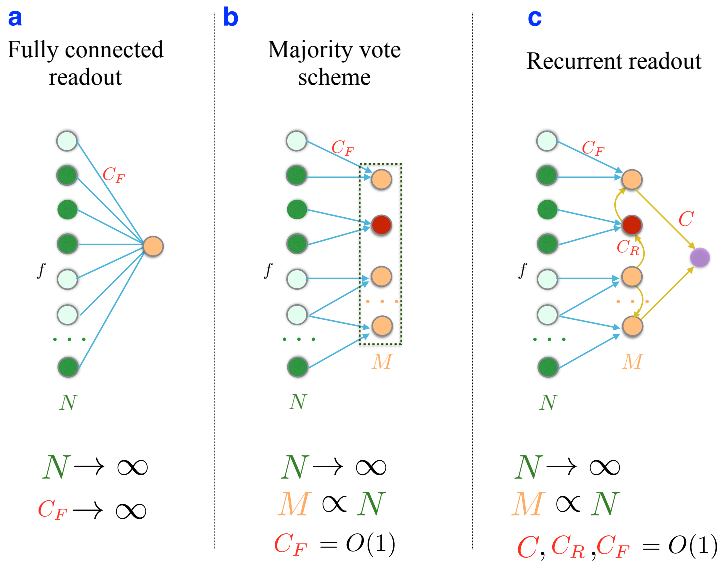

Figure 1. The architecture of the three network classifiers considered and the scaling regime.

classified scales linearly with the number of input neurons N, even when the number of connections per neural classifier C does not increase with N. The number of neural classifiers M is assumed to be proportional to N.

Interestingly, under certain conditions the recurrent scheme has larger classification capacity than the majority vote scheme. This happens for sparse input representations, the regime that is relevant for the mammalian hippocampus and that we investigate in detail.

2 Methods

2.1 Fully connected readout

In this section we derive the classification capacity of a single fully connected linear threshold readout, or *perceptron* (see figure 1a) achieved with a simple learning rule that we employ throughout this work. We assume that the input patterns and labels are random and uncorrelated, meaning that the activity of each input unit as well as the label is chosen independently, which makes calculations analytically tractable. We use a simple Hebbian-like learning rule, that is not optimal and thus leads to a lower capacity than Cover's $2N$ result [28]. However, the scaling of the maximal number of learned input patterns P with the number of input units N is still linear, as is shown below.

2.1.1 Input statistics

We assume that pairs (ξ^μ, η^μ) of a pattern ξ^μ and a label η^μ are drawn from a random ensemble of P pairs (pattern, label). The pattern components ξ_i^μ on all N input units and labels η^μ are random mutually independent variables. We assume that each component ξ_i^μ ($i = 1 \ldots N$ is the unit index and

$\mu = 1 \ldots P$ is the patterns index) is activated to 1 with probability f called *coding level* and otherwise is 0, and that label η^μ takes one of the two values: $\eta^\mu = +1$ with probability y, called the output sparseness, and $\eta^\mu = -1$ otherwise:

$$\xi_i^\mu = \begin{cases} 1, & \text{with probability } f \\ 0, & \text{with probability } 1-f \end{cases} \qquad \eta^\mu = \begin{cases} 1, & \text{with probability } y \\ -1, & \text{with probability } 1-y \end{cases} \tag{2.1}$$

2.1.2 Learning rule and the synaptic current

The linear threshold readout, or perceptron, classifies its inputs based on the sign of the weighted sum of the input components. This sum is sometimes called *synaptic current*, as it is viewed as modeling the synaptic current into a biological neuron

$$h = \sum_{i=1}^{N} w_i \xi_i$$

We say that the network has learned the association between P input patterns ξ_i^μ and P labels η^μ if for any pattern μ

$$\text{sign}\left(h^\mu - \theta\right) = \text{sign}\left(\sum_{i=1}^{N} w_i \xi_i^\mu - \theta\right) = \eta^\mu$$

Where θ is the threshold, that we further assume to be equal to zero.

Training the network means finding the set of weights w_i that satisfies the above expression.

The Hebb-like learning rule, which we use to train the weights $\{w_i\}$ of the classifier is:

$$w_i = \frac{1}{\sqrt{P}}\left(\sum_{\mu=1}^{P}(\xi_i^\mu - f)(\eta^\mu + 1 - 2y) - (1-f)(1-2y)\right) \tag{2.2}$$

In the case when patterns are equally likely to belong to either class ($y = \frac{1}{2}$), the learning rule simplifies to:

$$w_i = \frac{1}{\sqrt{P}}\sum_{\mu=1}^{P}(\xi_i^\mu - f)\eta^\mu$$

Here and in all that follows we set the threshold θ to zero.

After training, the synaptic current in response to a test pattern $\vec{\xi}^\nu$ is

$$h^\nu = \sum_{i=1}^{N} w_i \xi_i^\nu = \sum_{i=1}^{N} \frac{1}{\sqrt{P}}\left(\sum_{\mu=1}^{P}(\xi_i^\mu - f)(\eta^\mu + 1 - 2y) - (1-f)(1-2y)\right)\xi_i^\nu \tag{2.3}$$

If $\vec{\xi}^\nu$ together with its label η^ν was part of the training set, we can split the sum over patterns into the contribution from the presented pattern $\vec{\xi}^\nu$ and the contribution from other learned patterns to get

$$h^\nu = \frac{1}{\sqrt{P}}\left((1-f)\eta^\nu \sum_{i=1}^{N}\xi_i^\nu + \sum_{i=1}^{N}\left[\sum_{\mu\neq\nu}^{P}(\xi_i^\mu - f)(\eta^\mu + 1 - 2y)\right]\xi_i^\nu\right) \tag{2.4}$$

Here we used $(\xi_i^\nu)^2 = \xi_i^\nu$ because ξ_i^ν takes value 0 or 1.

We denote the number of active input units for the patterns ν by n^ν

$$n^\nu = \sum_{i=1}^{N} \xi_i^\nu \tag{2.5}$$

The value of n^ν is in *binomial distribution* of N trials with probability f, $\mathbf{B}(N, f)$. Its expected value is determined by the number of inputs N and the coding level f

$$\langle n^\nu \rangle = Nf \tag{2.6}$$

(here and throughout this text the angular brackets denote the mean over pattern realizations).

We replace the sum in the square brackets of (2.4) by $2\sqrt{Pf(1-f)y(1-y)n^\nu} z^\nu$ where we have introduced a *noise random variable* z^ν with zero mean and unit variance. The coefficient is concluded from the fact that each individual term $(\xi_i^\mu - f)(\eta^\mu + 1 - 2y)$ has variance

$$[f(1-f)^2 + (1-f)f^2][y(2-2y)^2 + (1-y)4y^2] = 4f(1-f)y(1-y) \tag{2.7}$$

and the fact that the ξ_i^μ variables are mutually independent. By the central limit theorem the noise variable z^ν can be approximated as Gaussian in the limit $P \to \infty$ with finite f and n^ν.

In terms of z^ν and n^ν the synaptic current is written as

$$h^\nu = \frac{1}{\sqrt{P}}(1-f)n^\nu \eta^\nu + 2\sqrt{f(1-f)y(1-y)n^\nu} z^\nu \tag{2.8}$$

If a pattern belongs to either class with equal probability ($y = \frac{1}{2}$), this expression simplifies to

$$h^\nu = \frac{1}{\sqrt{P}}(1-f)n^\nu \eta^\nu + \sqrt{f(1-f)n^\nu} z^\nu \tag{2.9}$$

Note that the first term is the one that reflects the correct classification of the input pattern, and the second one represents the noise caused by the interference from other patterns that were learned by the perceptron. The important parameter is the ratio of the two, which is proportional to $\sqrt{\frac{n_k^\nu}{P}}$

2.2 Committee machine

We now turn to deriving the classification capacity of a committee machine, the network shown on the figure 1b, where each out of M perceptrons receives feedforward connections from C_F input units. The connectivity C_F does not scale when the number of input units N increases.

The final decision is the majority vote of the classifiers. In other words, if classification is accurate

$$\text{sign}\left(\frac{1}{M}\sum_{k=1}^{M} \text{sign}\left(\sum_{i \in I_k} w_i^k \xi_i^\mu - \theta\right)\right) = \eta^\mu$$

Here $i \in I_k$ stands for all the input units (there are C_F of them) that are connected to the readout k, and w_i^k is the strength of the connection from the input unit i to the readout k (for the learning rule we consider w_i^k does not depend on k).

The synaptic current into the readout unit k when pattern $\vec{\xi}^{\nu}$ is presented is determined by

$$\boxed{h_k^{\nu} = \frac{1}{\sqrt{P}}(1-f)n_k^{\nu}\eta^{\nu} + \sqrt{f(1-f)n_k^{\nu}}z_k^{\nu}} \tag{2.10}$$

The number of active inputs connected to the perceptron k, n_k^{ν} is drawn from the binomial distribution $\mathbf{B}(C_F, f)$ of now C_F trials with the success rate f: and its expectation value is

$$\langle n_k^{\nu} \rangle = C_F f \tag{2.11}$$

Since the number of connections per readout C_F stays constant as the number of patterns P and the size of the network (N and M) grow, the probability of a single perceptron to classify a pattern correctly approaches the chance level. Indeed, in contrast to the fully connected perceptron, the number of active inputs n_k^{ν} does not change with the size of the network (see (2.11)). Hence, the first term of the expression (2.9) decreases in the absolute value as the number of patterns P grows, while the typical value of the second term stays the same. However, there is always a slight tendency towards the correct answer ($\langle h_k^{\nu}\eta^{\nu}\rangle > 0$), that can be utilized by having a growing number of sparsely connected classifiers that take a collective decision by majority vote. This scheme is known by the name of committee machine and has been shown to largely exceed the performance of a single classifier.

It is important to note that in order for the capacity of a committee machine to keep increasing as new classifiers (committee members) are added, the responses of different classifiers should stay sufficiently independent from each other. In the case of limited connectivity, which we consider here, the correlations automatically become smaller and smaller as we increase the number of input units. This happens because the probability of a typical pair of readouts to have a common input unit, and thus correlated responses, decreases. In order for the correlations not to be a limiting factor of the classification capacity, we need to increase the number of input units linearly with the number of perceptrons. If one introduces some other mechanism of reducing the correlations between the responses of the classifiers with common input units (like making different perceptrons learn different sets of patterns), a sublinear scaling of the number of input units N with the number of perceptrons M might be sufficient.

2.2.1 Non-overlapping case

The majority vote of M linear threshold classifiers is given by the *average vote*

$$r^{\nu} = \frac{1}{M}\sum_{k=1}^{M} r_k^{\nu}, \qquad r_k^{\nu} = \mathrm{sgn}(h_k^{\nu}) \tag{2.12}$$

where h_k^{ν} is given in (2.10). Positive $r^{\nu}\eta^{\nu}$ means that the pattern ν is classified correctly.

The expectation value of r^{ν} follows from (2.10) after integrating over the noise variable z_k^{ν}, which is approximated to be normally distributed. We make an assumption $Pf \gg n_k^{\nu}$, which is justified for a large number of patterns, and that allows us to use the approximation of the error function for small arguments to get

$$\langle r^\nu\rangle = \langle \mathrm{sgn}(h_k^\nu)\rangle_{n_k^\nu, z_k^\nu} = \left\langle \mathrm{erf}\frac{\sqrt{(1-f)n_k^\nu}\eta^\nu}{\sqrt{2Pf}}\right\rangle = \sqrt{\frac{2(1-f)}{\pi Pf}}\,\langle\sqrt{n_k^\nu}\rangle\,\eta^\nu \tag{2.13}$$

The expectation value $\langle\sqrt{n_k^\nu}\rangle$ is computed over the binomial distribution $\mathbf{B}(C_F, f)$

$$\langle\sqrt{n_k^\nu}\rangle = \sum_{n=0}^{C_F}\binom{C_F}{n} f^n(1-f)^{C_F-n}\sqrt{n} \tag{2.14}$$

In the dense regime, $C_F f \gg 1$, it can be approximated by

$$\langle\sqrt{n_k^\nu}\rangle = \sqrt{C_F f} \tag{2.15}$$

and in the extremely sparse case, when $C_F f \ll 1$ and only $n_k^\nu = \{0, 1\}$ are encountered substantially often, by

$$\langle\sqrt{n_k^\nu}\rangle = C_F f \tag{2.16}$$

To proceed with deriving the classification capacity, let us start with independent classifiers first. The independence of the responses can be achieved either by forcing the connections to be non-overlapping, or by assuming an additional mechanism that, for example, causes different classifiers to update their incoming connections in response to different subsets of the patterns.

In this case r^ν can be though of as drawn from a gaussian distribution with the mean given by (2.13) and the variance

$$\mathbf{cov}(r^\nu, r^\nu) = \frac{1}{M}(1 + \mathcal{O}(P^{-1})) \tag{2.17}$$

The gaussian assumption is justified by the law of large numbers.

Here and from now on we ignore the contributions of the subleading order, $\mathcal{O}(P^{-1})$ in this case.

The probability p_{correct} to classify a pattern correctly ($r^\nu\eta^\nu > 0$) can then be easily computed.

Fixing *tolerated error rate* ϵ and requiring $p_{\text{correct}} > 1-\epsilon$ leads to the expression for the maximal number of input patterns that can be classified with the accuracy $1-\epsilon$.

$$P_{\max} = \frac{\langle\sqrt{n}\rangle^2}{f}\frac{1-f}{\pi(\mathrm{erf}^{-1}(1-2\epsilon))^2}M \tag{2.18}$$

This result only holds for the case of non-overlapping connections or in the presence of a decoration mechanism. In the following section we generalize it to random connectivity.

2.2.2 Correction to classification capacity due to overlap in the connections

To derive an analogous expression for the overlapping case without a decorrelation mechanism we need to compute the variance

$$\mathbf{cov}(r^\nu, r^\nu) = \langle(r^\nu - \langle r^\nu\rangle)^2\rangle \tag{2.19}$$

of the average vote r^ν, defined by (2.12), taking into account the correlations of individual votes r^ν_k.

We start by splitting the covariance into diagonal and non-diagonal contributions:

$$\begin{aligned}\mathbf{cov}(r^\nu, r^\nu) &= \frac{1}{M^2}\sum_{k=1}^{M}\sum_{l=1}^{M}\mathbf{cov}(r^\nu_k, r^\nu_l) = \\ &= \frac{1}{M^2}\sum_{k=1}^{M}\mathbf{cov}(r^\nu_k, r^\nu_k) + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{l=1}^{M}(1-\delta_{kl})\mathbf{cov}(r^\nu_k, r^\nu_l) \overset{M\to\infty}{=} \\ &= \frac{1}{M} + \mathbf{cov}(r^\nu_k, r^\nu_l)_{k\neq l} \qquad (2.20)\end{aligned}$$

We assume that M and N scale linearly with P and $M, N, P \to \infty$. The leading terms are thus of the order $\frac{1}{M} \sim \frac{1}{N} \sim \frac{1}{P}$ and we ignore all the subleading contributions.

When the classifiers k and l share input units, the correlation between their responses is positive and is closely related to the correlation of the input currents h^ν_k and h^ν_l (2.10).

Let n^ν_{kl} be the number of input units that are connected to both the classifier k and the classifier l and are active in the pattern $\vec{\xi}^\nu$. For a large number of input units N and finite connectivity C_F we can assume that n^ν_{kl} can be either 0 or 1, but not more. The probability of n^ν_{kl} being 1 is given by

$$\text{Prob}(n_{kl} = 1) = f\frac{C_F^2}{N}$$

The number of active units that are connected to only one of the two classifiers are denoted by $\tilde{n}^\nu_k$ and $\tilde{n}^\nu_l$ respectively. In the current approximation both of them can be assumed to be distributed according to a binomial distribution $\mathbf{B}(C_F, f)$.

Then, the currents can be written as (see 2.10):

$$\begin{aligned} h^\nu_k &= \frac{1}{\sqrt{P}}(1-f)(\tilde{n}^\nu_k + n^\nu_{kl})\eta + \sqrt{f(1-f)n^\nu_{kl}}z^\nu_{kl} + \sqrt{f(1-f)\tilde{n}^\nu_k}z^\nu_k \\ h^\nu_l &= \frac{1}{\sqrt{P}}(1-f)(\tilde{n}^\nu_l + n^\nu_{kl})\eta + \sqrt{f(1-f)n^\nu_{kl}}z^\nu_{kl} + \sqrt{f(1-f)\tilde{n}^\nu_l}z^\nu_l \end{aligned} \qquad (2.21)$$

Where z^ν_k, z^ν_l and $z^\nu_k l$ are all independent gaussian variables with zero mean and unit variance.

To compute the covariance

$$\mathbf{cov}(r^\nu_k, r^\nu_l) = \langle \text{sgn}(h^\nu_k), \text{sgn}(h^\nu_l)\rangle - \langle \text{sgn}(h^\nu_k)\rangle\langle \text{sgn}(h^\nu_l)\rangle \qquad (2.22)$$

we start by integrating over the variables z^ν_k and z^ν_l to get

$$\begin{aligned}\langle \text{sgn}(h^\nu_k)\rangle_{z^\nu_k} &= \text{erf}\left(\frac{z^\nu_{kl}}{\sqrt{2}}\sqrt{\frac{n^\nu_{kl}}{\tilde{n}_k}}\right) \\ \langle \text{sgn}(h^\nu_l)\rangle_{z^\nu_l} &= \text{erf}\left(\frac{z^\nu_{kl}}{\sqrt{2}}\sqrt{\frac{n^\nu_{kl}}{\tilde{n}^\nu_l}}\right). \end{aligned} \qquad (2.23)$$

Then, (2.22) can be evaluated using the table integral [1]

$$\int_0^\infty \mathrm{erf}(az)\mathrm{erf}(bz)e^{-c^2z^2}dz = \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{ab}{c\Delta} \qquad \Delta = \sqrt{a^2+b^2+c^2} \tag{2.24}$$

In the leading order we get:

$$\boxed{\begin{aligned} \mathbf{cov}(r_k^\nu, r_l^\nu)_{k\neq l} = \mathbf{cov}(\mathrm{sign}\,(h_k)\,, \mathrm{sign}\,(h_l))_{k\neq l} = \frac{1}{N}\varphi_{C_F,f} \\ \varphi_{C_F,f} = \frac{2fC_F^2}{\pi}\left\langle \tan^{-1}\frac{1}{\sqrt{(\tilde{n}_k+1)(\tilde{n}_l+1)-1}}\right\rangle_{\tilde{n}_k,\tilde{n}_l\in\mathbf{B}(C_F,f)} \end{aligned}} \tag{2.25}$$

In the dense regime ($C_F f \gg 1$) the expression for (2.25) can be approximated as

$$\varphi_{C_F,f} = \frac{2C_F}{\pi} \tag{2.26}$$

and

$$\mathbf{cov}(r_k^\nu, r_l^\nu)_{k\neq l} = \frac{2C_F}{\pi N} \tag{2.27}$$

While in the sparse approximation ($C_F f \ll 1$),

$$\varphi_{C_F,f} = fC_F^2 \tag{2.28}$$

and

$$\mathbf{cov}(r_k^\nu, r_l^\nu)_{k\neq l} = \frac{fC_F^2}{N} \tag{2.29}$$

Plugging this result into (2.20), we get for the variance of the majority vote r^ν in the overlapping case

$$\mathbf{cov}(r^\nu, r^\nu) = \frac{1}{M} + \frac{1}{N}\varphi_{C_F,f},$$

which together with (2.13) leads for the maximal number of input patterns that the committee machine can learn to classify with the accuracy $1-\epsilon$

$$\boxed{P_{\max} = \frac{\langle\sqrt{n_k}\rangle^2}{f}\frac{1-f}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2\pi}\frac{M}{1+\frac{M}{N}\varphi_{C_F,f}}} \tag{2.30}$$

where $\varphi_{C_F,f}$ is in (2.25)(2.26)(2.28).

If both the number of input units N and the number of classifiers M increase in proportion to each other, the capacity P increases linearly with N and M.

In the case of dense representations, $C_F f \gg 1$ the last expression simplifies to

$$P_{\max} = \frac{1-f}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2\pi}\frac{C_F M}{1+\frac{M}{N}\frac{2C_F}{\pi}} \tag{2.31}$$

and in the sparse limit, $C_F f \ll 1$

$$P_{\max} = \frac{1}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2\pi}\frac{MC_F^2 f}{1+\frac{M}{N}C_F^2 f} \tag{2.32}$$

[1] See equation 18 on page 158 in [29]

2.3 Committee machine with recurrent connections

The majority rule scenario already overcomes the limitations of the connectivity of a single perceptron, but this is not the final answer to constructing a classifier with limited connectivity. The reason is that we still need to implement the majority rule and bring the classification signal to the level of a single unit. The naive way to do it would require another final readout that would have to sample the entire population of M intermediate layer perceptrons. Since M has to scale linearly with the number of learned patterns P, the connectivity of the final readout would also have to scale linearly with P (see 2.30) and would exceed any predetermined limit for sufficiently large number of learned patterns.

To implement the majority vote of the intermediate perceptrons while keeping the connectivity of any unit in the network limited, we introduce the recurrent connectivity in the layer of perceptrons. Our goal is to have two attractor states of the intermediate layer dynamics, that correspond to the two classes. The feedforward input through the connections $\{w_i^k\}$ trained in the same way as before, will be slightly biased in the positive direction for one class of the input patterns and in the negative for the other. This slight bias determines which attractor state the network will choose. It is essential that the attractors are far away and do not become closer when the number of learned patterns P increases implies that the final readout will be able to discriminate between these states, and thus indicate the class of the presented pattern, even if its connectivity does not scale with P. It turns out that for two-way classification it is enough to have random recurrent connectivity with sufficiently large but not increasing with P number of connections per unit, and the weights of these recurrent connection do not have to be tuned (no learning required for recurrent connections).

We want to compute the probability of the network of recurrently connected readouts to go to the correct attractor (the one assigned to the class of the input pattern presented) as a function of the number of input units N, number of perceptrons M and various parameters of the network.

2.3.1 Network topology

The recurrent readout network shown on the right of figure 1c consists of the input layer (green), the intermediate layer of perceptrons (orange) and the final readout unit (purple).

As before, the *input layer* of N neurons is presented with a random and uncorrelated patterns $\xi^\mu = (\xi_i^\mu)_{i=1\ldots N}$ from a set of P patterns $(\xi^\mu)_{\mu=1\ldots P}$ that the network has learned to classify.

The layer of perceptrons we now call *intermediate layer*. It consists of M linear threshold readouts, each of which is connected to a randomly chosen C_F out of N input units. Hence, the *feedforward connectivity* C_F is the number of feedforward inputs that each perceptron receives. The C_F is an important parameter in the problem as it determines the classification capacity of a perceptron considered in isolation. The intermediate layer is recurrently connected. For the case of binary classification, the probability that two readouts are connected is the same for each pair. The recurrent connections are not plastic and can be chosen to be all of equal strength α.

The *final layer* consists of a single readout unit that is connected to a randomly chosen subset of

C perceptrons in the intermediate layer, with the strength of all connections taken equal.

The connectivity matrix J_{kl}, $k, l \in [1 \ldots M]$ is assumed to be symmetric

$$J_{kl} = \begin{cases} 1 & \text{if perceptron } k \text{ and } l \text{ are connected} \\ 0 & \text{if perceptron } k \text{ and } l \text{ are not connected} \end{cases} \tag{2.33}$$

Let the C_R be the number of recurrent connections per unit

$$\sum_{l=1}^{M} J_{kl} = C_R, \qquad \forall k \in [1 \ldots M] \tag{2.34}$$

We will keep the connectivity parameters C_F, C_R and C and coding level f at fixed constant values, while sending the number of input units N, the number of intermediate perceptrons M and the number of patterns P to infinity

$$\boxed{P, M, N \to \infty; \qquad f, C_F, C_R, C \quad \text{are constant}} \tag{2.35}$$

We want to recover the linear scaling of the maximal number of patterns $P_{\max}$ that the network can learn to classify with the number of input units N, which is known to hold for the fully connected perceptron [28].

2.3.2 Discrete time dynamical model

We model the recurrent dynamics as a probabilistic dynamical process in discrete time t with the probabilistic transition rule from a network state at time t to a network state at time $t + 1$. Let $s_k(t) \in [-1, 1]$ for $k \in [1 \ldots M]$ be the dynamical variable describing the state of unit k at time t in recurrent network.

Let $\tilde{h}_k^\nu$ be the total current into the readout unit k

$$\tilde{h}_k^\nu(t) = \sum_{l=1}^{M} \alpha J_{kl} s_l^\nu(t) + h_k^\nu \tag{2.36}$$

where the first term corresponds to the recurrent contribution and the second term represents the feedforward current from the input layer (2.10) that is constant in time.

The probabilistic transition rule from the state at time t to the state to time $t + 1$ is

$$s_k(t+1) = \begin{cases} 1, & \text{with probability} \quad \frac{1}{1+e^{-2\beta \tilde{h}_k^\nu(t)}} \\ -1, & \text{with probability} \quad \frac{e^{-2\beta \tilde{h}_k(t)}}{1+e^{-2\beta \tilde{h}_k^\nu(t)}} \end{cases} \tag{2.37}$$

Here β is the *inverse temperature parameter* for the statistical model of the recurrent dynamics.

We approximate this probabilistic recurrent dynamics with the *mean field* method.

2.3.3 Mean field analysis of the recurrent dynamics

To compute the capacity of such a recurrent classifier, we analyze the recurrent dynamics in the mean field approximation. The activities of the recurrently connected units are represented by the variables $s_k = \{+1, -1\}$ with $k = 1 \dots M$. The mean field equation for the *average activation* of the recurrently connected intermediate layer in response to the pattern ν,

$$m^\nu = \frac{1}{M} \sum_{k=1}^{M} s_k$$

reads

$$m^\nu = \frac{1}{M} \sum_{k=1}^{M} \tanh\left(\beta\left(C_R \alpha m^\nu + h_k^\nu\right)\right) \qquad (2.38)$$

The average activation m^ν is close to zero if the amount of active and inactive units is approximately the same. If the majority of the units is in the active state, m^ν will be close to 1, and if the majority is inactive, m^ν will be close to -1.

Here C_R is the average number of connections per unit, α is the strength of recurrent synapses (we assume they are all excitatory and of equal strength), β is the inverse temperature parameter and h_k^ν is the feedforward input given by (2.10).

We proceed by analyzing the above equation graphically. The plot of the right-hand side is a sigmoid curve and the left-hand side is a line at 45 degrees. The intersections of these two lines determine the solutions to the equation. There are two possible situations that correspond to two different scenarios of the network dynamics.

The first scenario, shown on the figures 2a and 2b, is characterized by having only one point of intersection of the line and the sigmoid. In this case there is only one solution to the mean field equation (2.38) and only one stable state of the recurrent network. The right hand side of the equation is almost but not quite an odd function of its argument m^ν, so the sigmoidal curve representing it is slightly shifted to the left if $\frac{1}{M}\sum_{k=1}^{M} \tanh\left(\beta h_k^\nu\right) > 0$ and to the right otherwise. If the curve is shifted to the left, the single point of its intersection with the strait line passing thorough the origin will be in the right half-plane. So, for the positive input pattern ($\eta^\nu = +1$ and h_k^ν is more likely to be positive) the mean activity of the intermediate layer in the stable state m^ν will usually be positive, while for the negative input patterns it will be negative. Even though there is a relation between the sign of the mean activity of the intermediate layer in the stable state and the class of the input pattern, this is not helpful for our purposes. The reason is that we encounter exactly the same problem as for the case of no recurrent connections: the absolute value of the average activity m^ν will decrease with the number of learned patterns P, which means that the number of active and inactive units in the intermediate layer will become more and more similar. Consequently, to sample this small imbalance we would require larger and larger connectivity of the final readout. In short, the regime with one stable solution (figures 2a and 2b) is not much different from the case of no recurrent connections. Not surprisingly, this regime corresponds to relatively weak recurrent connections.

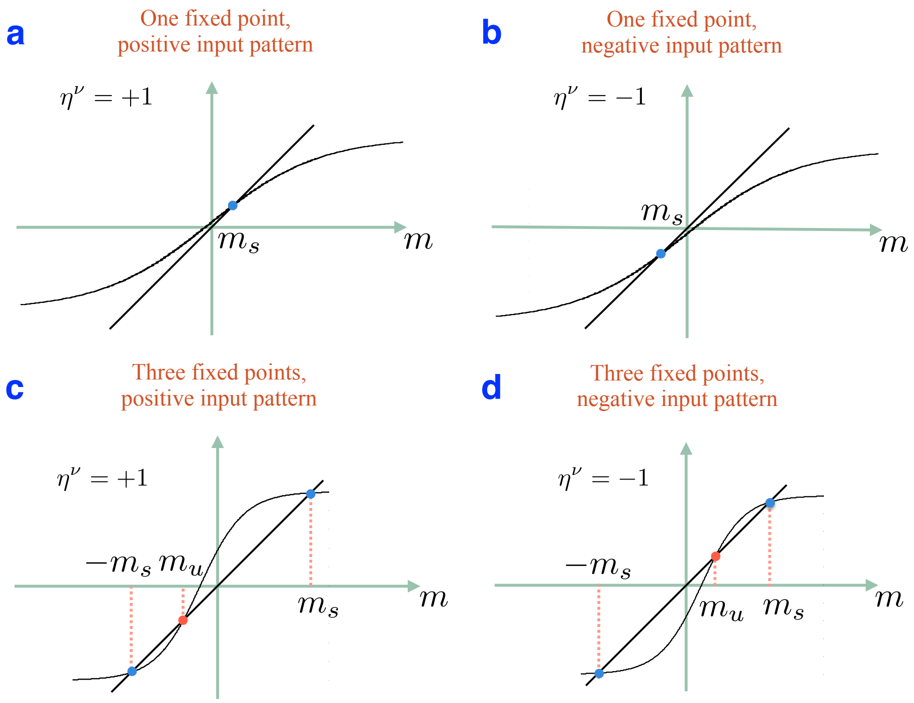

Figure 2. The graphical representation of the mean field equation 2.38

It is the other situation, shown on the bottom of the figure (figures 2c and 2d), that is actually of interest. There are three points of intersection of the sigmoid curve of the right hand side of the equation (2.38) and the straight line of the left hand side. The stable states of the network correspond to the rightmost and the leftmost solutions, that are both characterized by a large imbalance between active and inactive units ($|m^\nu| \sim 1$). Most importantly, these solutions are virtually insensitive to the distribution of h_k^ν, and hence to the number of learned patterns P. So, if we postulate that the right solution corresponds to the positive input patterns and the left solution to the negative ones, it will be easy for a downstream readout with connectivity that does not increase with P to distinguish between them.

The middle intersection point m_u corresponds to the unstable solution. When the network is initialized at the state $\{s_k^0\}$ with $m_0 = \sum_{k=1}^{M} s_k^0$ on the left of the unstable solution $m_0 < m_u$, the recurrent dynamics will most likely evolve to the left stable state, and if initialized at $m_0 > m_u$ it will evolve to the right stable state. As shown on the figure 2c,d the point of unstable equilibrium will be to the left of the origin for a positive input pattern and to the right of the origin otherwise (due to the difference in the mean of the distributions of h_k^ν). Hence, initiating the network at $m_0 = 0$ will serve the purpose of biasing the evolution of the network towards the stable state that corresponds to the class of the input pattern. If the number of learned patterns P is large, the point of unstable equilibrium is very close to zero $|m_u| \sim \frac{1}{\sqrt{P}}$, this is the manifestation of the same problem as before, namely the decrease of the signal to noise ratio with the increasing number of learned patterns. Thus, the noise in the initial state of the network m_0 should also decrease as $\frac{1}{\sqrt{P}}$. This is achieved if all the units in the intermediate layer are initialized at $s_0 = \pm 1$ with equal probabilities independently from each other, and the number of units M is linear in P (the same scaling as for the committee machine discussed earlier). We use this initialization process to derive the classification capacity and to run the simulations. In the discussion we suggest a biologically plausible way to initialize the network at the desired point.

To summarize, the information about the class of the input pattern is contained in the feedforward input to the intermediate recurrently connected layer. In the case of a single stable state (figures 2a and 2b), although average activity of the network reflects this information, the signal is very small and a fully connected downstream readout is required. In the case of two stable states (figures 2c and 2d), this small signal biases the network to choose the one corresponding to the class of the input pattern, and by doing so, the network amplifies the feedforward signal making it easy to read out by a sparsely connected downstream readout.

2.3.4 Number of classifiable inputs

As discussed in the previous section, the requirement for the correct classification of an input pattern by means of recurrently connected committee machine is that the average activity of the network at the initial moment m_0^ν is on the correct side of the point of unstable equilibrium m_u^ν, namely

$$(m_0^\nu - m_u^\nu)\eta^\nu > 0 \tag{2.39}$$

where η^ν is the required output ($\eta^\nu = \{\pm 1\}$).

In what follows we drop the pattern index ν.

The statistics of m_0 over random initializations of the network follows from its definition

$$m_0 = \sum_{k=1}^{M} s_k^0$$

where each unit is initialized at $s_k = +1$ or $s_k = -1$ with equal probability:

$$\langle m_0 \rangle = 0$$

$$\mathbf{cov}(m_0, m_0) = \langle (m_0 - \langle m_0 \rangle)^2 \rangle = \frac{1}{M}$$

Since M is a large number, we approximate the distribution of m_0 by a Gaussian distribution with these mean and variance.

The position of the unstable equilibrium point m_u, corresponding to one of the three solutions (the one that is close to zero) of the mean field equation (2.38), can not be computed analytically in the general case. However, there are parameter regimes in which we can compute the approximate first and second order statistics of m_u over random realizations of the input patterns. These parameter regimes and corresponding approximations are discussed in the following section. Once the mean μ_u, which depends on the number of learned patterns P, and the variance σ_u^2 of m_u are known, the requirement to classify P input patterns with accuracy $1-\epsilon$ can be written as (assuming the distribution of m_u to be also Gaussian)

$$1-\epsilon = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\frac{-\mu_u(P)\eta}{\sqrt{2\left(\frac{1}{M}+\sigma_u^2\right)}} \tag{2.40}$$

The expected number P of correctly classified patterns can be found by inverting the above equation.

In the following sections we consider different parameter regimes that lead to different approximations for μ_u and σ_u.

2.3.5 The uniform regime

In the current study, among other issues we are interested in the consequences of the sparsity of input representations. Since we consider the feedforward connectivity C_F to be a constant number and not to scale with the size of the network, for sparse representations there will be a substantial number of readouts that receive zero feedforward input. Unless the dynamical noise is very high, these units should be considered separately, and in the mean field approximation an additional order parameter should be introduced to describe their average activity. We call these units *free units.*

Uniform regime is the parameter regime under which it is not necessary to analyze the free units separately, and the equation (2.38) is valid without modifications. Obviously, when the input representations are dense, $C_F f \gg 1$, the network of the intermediate layer is in the uniform regime, since there are not enough of free units to make a difference. However, assuming the uniform regime is also valid independently of the number of free units, when the dynamical noise is very large in comparison with the feedforward input (see the next section).

The conditions defining the uniform regime are:

Sparse input representations and high noise $\quad C_F f \lesssim 1 \qquad \beta^{-1} \gg \sqrt{f}$

or

Dense input representations $\quad C_F f \gg 1$

2.3.6 Uniform regime, high noise

One approximation we can make to find the unstable solution m_u of the mean field equation (2.38) is the high noise approximation, which is defined by the requirement

$$\beta h_k^\nu \ll 1 \qquad \text{for most readouts } k \text{ and patterns } \nu \tag{2.41}$$

It follows from the expression (2.10) for the feedforward current that this requirement is met if

$$\text{for dense input:} \quad C_F f \gg 1 \quad \beta^{-1} \gg \sqrt{f^2(1-f)C_F}$$

$$\text{for sparse input:} \quad C_F f \lesssim 1 \quad \beta^{-1} \gg \sqrt{f}$$

The condition for having three solutions of equation (2.38) rather than one (see figure 2) is

$$C_R\alpha > \beta^{-1}$$

Since we are looking for the solution, which is close to zero and (2.41) is satisfied for most of the terms, the equation (2.38) can be approximated by replacing the hyperbolic tangent by its argument:

$$m_u^\nu = \frac{1}{M}\sum_{k=1}^{M}\beta(C_R\alpha m_u^\nu + h_k^\nu),$$

(note that this approximation is also valid for the terms with $h_k^\nu = 0$).

Solving this equation leads for the mean μ_u and standard deviation σ_u of m_u:

$$\mu_u = -\frac{1}{C_R\alpha - \beta^{-1}}\mu_h$$

and

$$\sigma_u = \frac{\sigma_h}{C_R\alpha - \beta^{-1}}\sqrt{\frac{1}{M} + \frac{C_F}{N}} \tag{2.42}$$

The mean μ_h and the standard deviation σ_h of the feedforward current h_k^ν are computed from (2.10):

$$\begin{aligned}\mu_h &= \frac{1}{\sqrt{P}}f(1-f)C_F\eta\\ \sigma_h &= \sqrt{C_F f^2(1-f)}\end{aligned} \tag{2.43}$$

The C_F/N term in (2.42) comes from the correlations between the feedforward currents h_k^ν into different readouts k due to overlapping connections (see the Appendix A1).

Now the maximum number of learned patterns for the classifier in the uniform regime for high noise approximation can be computed from (2.40) and is given by

$$P_{\max} = \frac{1-f}{2[\text{erf}^{-1}(1-2\epsilon)]^2}\frac{C_F M}{1 + \frac{M}{N}C_F + \frac{(C_R\alpha-\beta^{-1})^2}{C_F f^2(1-f)}} \tag{2.44}$$

2.3.7 Uniform regime, low noise

The other approximation in which the equation (2.38) can be solved is

$$\beta h_k^\nu \gg 1, \tag{2.45}$$

which is true if

$$\beta^{-1} \ll \sqrt{f^2(1-f)C_F}$$

Under this condition, assuming the uniform regime is only valid if the input representations are dense

$$C_F f \gg 1$$

The condition for having three solutions to the mean field equation in the low noise approximation becomes (see (2.49))

$$\sigma_h = \sqrt{f^2(1-f)C_F} < \sqrt{\frac{2}{\pi}} C_R \alpha. \tag{2.46}$$

In this case the hyperbolic tangent in the equation (2.38) can be approximated by the sign function

$$m_u^\nu = \frac{1}{M} \sum_{k=1}^{M} \operatorname{sign}\left[\beta(C_R \alpha m_u^\nu + h_k^\nu)\right]$$

Let us denote the right side of this equation by $g(m_u)$, where

$$g(m) = \frac{1}{M} \sum_{k=1}^{M} \operatorname{sign}(C_R \alpha m + h_k^\nu) \tag{2.47}$$

is a stochastic function over different realizations of $\{h_k^\nu\}$.

Note that in this case, having a substantial fraction of terms with $h_k^\nu = 0$ would lead to a discontinuity of the right hand side at $m_u^\nu = 0$.

The mean $\langle g(m) \rangle$ can be found by integrating over the distribution of h_k^ν (see (2.10))

$$\langle g(m) \rangle = \operatorname{erf}\left(\frac{C_R \alpha m + \mu_h}{\sqrt{2}\sigma_h}\right) \tag{2.48}$$

Where μ_h and σ_h are the mean and standard deviation of h_k^ν respectively, which are given by (2.43).

Thus, when averaged over training patterns, the mean field equation becomes

$$m = \operatorname{erf}\left(\frac{C_R \alpha m + \mu_h}{\sqrt{2}\sigma_h}\right) \tag{2.49}$$

and it has three solutions when the derivative of the right-hand side with respect to m at $m = 0$ is larger than 1, which for $\mu_h \ll \sigma_h$ immediately leads (2.46).

We now return to estimating the mean and the standard deviation of m_u, which is the unstable solution to the approximated mean field equation

$$m_u = g(m_u) \tag{2.50}$$

where $g(m)$ is defined by (2.47).

For $\mu_h \ll \sigma_h$, which is always the case if the number of stored patterns P is large enough, we assume that $C_R \alpha m_u$ is also small compared to σ_h and check the self-consistency later. Then, we can use the approximation for the error function at small arguments to get

$$\langle g(m) \rangle = \sqrt{\frac{2}{\pi}} \frac{C_R \alpha m + \mu_h}{\sigma_h} \tag{2.51}$$

the variance of $g(m)$ can be written as as sum of the diagonal and the non-diagonal terms

$$\mathbf{cov}(g(m), g(m)) = \frac{1}{M} + \mathbf{cov}(\operatorname{sgn}(C_R \alpha m + h_k), \operatorname{sgn}(C_R \alpha m + h_l)) \rangle_{k \neq l} \tag{2.52}$$

which is similar to the expression (2.20) for the variance of $\frac{1}{M}\sum_{k=1}^{M}\mathrm{sgn}(h_k)$ computed previously in (2.25). The only difference that here the distribution of h_k is shifted by $C_R\alpha m$. However, because the mean $\langle h_k^\nu$ did not affect the result (2.25) and $C_R\alpha m_u + \mu_h$ is still negligible compared to σ_h, we can write

$$\mathbf{cov}(g(m), g(m)) = \frac{1}{M} + \frac{\varphi_{C_F,f}}{N} \tag{2.53}$$

where $\varphi_{C_F,f}$ is given in (2.25).

As a sum of large number M of weakly correlated terms, $g(m)$ can be assumed to be normally distributed and can be written as

$$g(m) = \sqrt{\frac{2}{\pi}}\frac{C_R\alpha m + \mu_h}{\sigma_h} + \sqrt{\frac{1}{M} + \frac{\varphi_{C_F,f}}{N}}z^\nu \tag{2.54}$$

where z^ν is a Gaussian variable with zero mean and unit variance.

Plugging the expression for $g(m)$ into (2.50), and solving for m_u we get

$$m_u = -\frac{1}{\sqrt{\frac{2}{\pi}\frac{C_R\alpha}{\sigma_h}} - 1}\sqrt{\frac{2}{\pi}\frac{\mu_h}{\sigma_h}} + \frac{1}{\sqrt{\frac{2}{\pi}\frac{C_R\alpha}{\sigma_h}} - 1}\sqrt{\frac{1}{M} + \frac{1}{N}\varphi_{C_F,f}}\, z^\nu \tag{2.55}$$

where $\varphi_{C_F,f}$ is (2.25)(2.26).

So, the expectation value of m_u is given by

$$\mu_u = -\frac{1}{\sqrt{\frac{2}{\pi}\frac{C_R\alpha}{\sigma_h}} - 1}\sqrt{\frac{2}{\pi}\frac{\mu_h}{\sigma_h}} = -\frac{1}{\sqrt{\frac{2}{\pi}\frac{C_R\alpha}{\sqrt{C_F f^2(1-f)}}} - 1}\sqrt{\frac{2}{\pi}}\frac{\sqrt{C_F(1-f)}}{\sqrt{P}}\eta$$

and the standard deviation is:

$$\sigma_u = \frac{1}{\sqrt{\frac{2}{\pi}\frac{C_R\alpha}{\sqrt{C_F f^2(1-f)}}} - 1}\sqrt{\frac{1}{M} + \frac{1}{N}\varphi_{C_F,f}}$$

Because uniform regime and low noise implies dense input representation, we can use the dense approximation for $\varphi_{C_F,f}$ given by (2.26). Plugging these results into (2.40) leads the capacity for the uniform regime, low noise

$$P_{\max} = \frac{1-f}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2\pi}\frac{C_F M}{1 + \frac{M}{N}\frac{2}{\pi}C_F + \left(\sqrt{\frac{2}{\pi}\frac{C_R\alpha}{\sqrt{C_F f^2(1-f)}}} - 1\right)^2}$$

2.3.8 Non-uniform regimes

When the input representation is sparse

$$C_F f \lesssim 1, \tag{2.56}$$

there is a substantial fraction of readouts for which all inputs are silent, we call them the free units. If the noise is not very high $\beta\sqrt{f} \gtrsim 1$, these readouts are statistically different from those that do receive a non-zero input. To analyze such a system in the mean-field approximation, two order parameters and two coupled mean-field equations should be introduced. To avoid this complication we consider a simpler case, to which we refer to as the *two-subnetworks* regime. This regime is characterized by the recurrent connections that are relatively weak when compared to the feedforward ones, so that the state of those readouts that do receive non-zero feedforward input is determined by this input only. Neither recurrent input nor noise can flip them. Only the free units participate in the recurrent dynamics and their mean activity in the final state reflects the class of the input pattern. Which of the two stable states the subnetwork of free units will go to is biased by the input from the input receiving units, which do have the information about the class of the input pattern from the feedforward input.

This approximation is valid if

$$\alpha\sqrt{C_R} \ll \sqrt{f} \tag{2.57}$$

$$\beta^{-1} \ll \sqrt{f} \tag{2.58}$$

To be more precise, the former condition does not guarantee that the recurrent input will not be able to flip the input receiving units close to the final state, when most of the free units are synchronized. However, if this is the case, their activity already reflects the correct classification of the input pattern, and the input receiving units will flip in the right direction.

The mean field equation (2.38) should now be seen as describing the subnetwork of free units, and should be modified in several ways.

First, the number of units in the network is

$$M_f = M\mathrm{e}^{-C_F f} \tag{2.59}$$

(since for small f the probability of all C_F independent inputs to be silent is $(1-f)^{C_F} \approx \mathrm{e}^{-C_F f}$). Second, only $C_R\mathrm{e}^{-C_F f}$ out of C_R recurrent connections per unit come from other free units. Also, the external input to the network now comes from other (input receiving) units in the intermediate layer, rather than from the input layer.

The modified mean-field equation reads:

$$\tilde{m}^\nu = \frac{1}{M_f}\sum_{k=1}^{M_f} \tanh\left(\beta\left(C_R\alpha e^{-C_F f}\tilde{m}^\nu + H_k^\nu\right)\right) \tag{2.60}$$

where $\tilde{m}^\nu$ is the average activity of the subnetwork of free readouts. The index k runs over all the free units.

$$H_k^\nu = \sum_{l=1}^{M_{IR}} \alpha J_{kl}\mathrm{sign}\left(h_l^\nu\right) \tag{2.61}$$

the summation is over the input receiving units and h_k^ν is the feedforward current of (2.10) with $n_l^\nu \neq 0$. M_{IR} is the number of these units

$$M_{IR} = M(1-\mathrm{e}^{-C_F f})$$

On average, the free unit k receives C_R inputs, and $(1-e^{-C_F f})C_R$ of them come from input receivers. So (2.61) will have on average $C_R(1-e^{-C_F f})$ non-zero terms. Assuming that this is a large number, H_k^ν is a Gaussian variable with the mean given (in the leading order) by

$$\mu_H = \alpha C_R(1 - e^{-C_F f})\langle \text{sign}\,(h_l^\nu)\rangle_{n_k \neq 0} = \alpha C_R \langle \text{sign}\,(h_l^\nu)\rangle \tag{2.62}$$

which using (2.13) becomes

$$\mu_H = \sqrt{\frac{2}{\pi}} \frac{\langle \sqrt{n} \rangle}{\sqrt{Pf}} C_R \alpha \eta^\nu \tag{2.63}$$

(assuming $1 - f \approx 1$). The number of active inputs n_l^ν connected to the intermediate unit comes from the binomial distribution, $n \sim \mathbf{B}(N, f)$.

The standard deviation of H_k^ν is

$$\sigma_H = \alpha \sqrt{C_R(1 - e^{-C_F f})} \tag{2.64}$$

(the corrections to due to correlations between different input receiving units are suppressed as $1/N$ and will become negligible for large networks when C_R does not scale with N).

To find the statistics of $\tilde{m}_u$, the point of unstable equilibrium, we again consider high and low noise approximations, but now we should compare the inverse temperature parameter β to the standard deviation of H_k^ν.

What we further call *intermediate noise* is the noise which is small on the scale of the feedforward input (2.58) but large when compared to the typical values of H_k^ν.

2.3.9 Two-subnetworks regime, intermediate noise

The following analysis is valid if in addition to the conditions (2.56), (2.57) and (2.58) the dynamical noise is high in comparison to the typical external input to the subnetwork of the free units:

$$\beta \sigma_H = \beta \alpha \sqrt{C_R(1 - \mathrm{e}^{-C_F f})} \ll 1$$

The condition for three solutions to the mean field equation (2.60) in this case reads

$$\beta \alpha C_R \mathrm{e}^{-C_F f} > 1$$

The former inequality allows us to approximate the hyperbolic tangent in (2.60) by its argument when looking for the unstable solution $\tilde{m}_u$, which is close to zero:

$$\tilde{m}_u^\nu = \beta C_R \alpha e^{-C_F f} \tilde{m}_u + \beta \frac{1}{M_f} \sum_{k=1}^{M_f} \sum_{l=1}^{M_{IR}} J_{kl} \alpha\, \text{sign}\,(h_l^\nu) \tag{2.65}$$

Each input receiving unit l has C_R outgoing connections and approximately $e^{-C_F f} C_R$ of them terminate on a free unit. Hence, the double sum can be rewritten to get

$$\tilde{m}_u^\nu = \beta C_R \alpha e^{-C_F f} \tilde{m}_u^\nu + \beta C_R \alpha e^{-C_F f} \frac{1}{M_f} \sum_{l=1}^{M_{IR}} \text{sign}\,(h_l^\nu) \tag{2.66}$$

Solving this equation for $\tilde{m}_u$ leads (see 2.59)

$$\tilde{m}_u^\nu = -\frac{\beta C_R \alpha}{\beta C_R \alpha e^{-C_F f} - 1}(1 - e^{-C_F f})\bar{r}^\nu \tag{2.67}$$

where we have introduced $\bar{r}^\nu$: the sign of the feedforward current averaged over the units for which this current is non-zero

$$\bar{r}^\nu = \frac{1}{M_{IR}} \sum_{l=1}^{M_{IR}} \mathrm{sgn}(h_l^\nu)$$

The statistics of $\bar{r}^\nu$ is closely related to previously computed statistics of r^ν (see (2.12)), which is the sign of the feedforward current averaged over all the intermediate units. Namely,

$$\langle \bar{r}^\nu \rangle = \frac{1}{1-\mathrm{e}^{-C_F f}} \langle r^\nu \rangle. \tag{2.68}$$

The expression for $\langle r^\nu \rangle$ is given in (2.13), which leads (we approximate $1-f\approx 1$)

$$\langle \bar{r}^\nu \rangle = \langle \mathrm{sgn}(h^\nu) \rangle_{n^\nu \neq 0} = \frac{1}{1-\mathrm{e}^{-C_F f}} \sqrt{\frac{2}{\pi}} \frac{\langle \sqrt{n} \rangle}{\sqrt{Pf}} \eta^\nu \tag{2.69}$$

To compute the second order statistics of $\bar{r}^\nu$, we use the relation

$$\mathbf{cov}(\mathrm{sign}\,(h_k^\nu)\,, \mathrm{sign}\,(h_l^\nu))_{k\neq l; n_k^\nu, n_l^\nu \neq 0} = \frac{1}{(1-\mathrm{e}^{-C_F f})^2} \mathbf{cov}(\mathrm{sign}\,(h_k^\nu)\,, \mathrm{sign}\,(h_l^\nu))_{k\neq l}$$

The covariance on the right-hand side was also computed in (2.25), which allows us to write

$$\mathbf{cov}(\bar{r}^\nu, \bar{r}^\nu) = \frac{1}{M_{IR}} + \frac{\varphi_{C_F,f}}{(1-e^{-C_F f})^2} \frac{1}{N} \tag{2.70}$$

Plugging in (2.69) and (2.70) to (2.67) leads the expressions for the mean and the standard deviation of $\tilde{m}_u^\nu$:

$$\mu_u = -\sqrt{\frac{2}{\pi}} \frac{\beta C_R \alpha}{\beta C_R \alpha \mathrm{e}^{-C_F f} - 1} \frac{\langle \sqrt{n} \rangle}{\sqrt{Pf}} \eta,$$

(the mean $\langle \sqrt{n} \rangle$ is computed assuming a binomial distribution for the number of active inputs n connected to a readout $n \sim \mathbf{B}(N,f)$), and

$$\sigma_u = \frac{\beta C_R \alpha}{\beta C_R \alpha \mathrm{e}^{-C_F f} - 1} \sqrt{\frac{1}{M}(1-\mathrm{e}^{-C_F f}) + \varphi_{C_F,f}\frac{1}{N}} \tag{2.71}$$

Now we can use (2.40) to compute the maximum number of learned patterns in the two-subnetworkd regime under intermediate noise. The number of units in the network M in (2.40) should be replaced by the number of free units $M\mathrm{e}^{-C_F f}$. The result is

$$P = \frac{\langle \sqrt{n} \rangle^2}{f} \frac{1}{\pi \left[\mathrm{erf}^{-1}(1-2\epsilon)\right]^2} \frac{M}{\gamma + \frac{M}{N}\varphi_{C_F,f}}$$

where

$$\gamma = 1 - \frac{2\beta C_R \alpha - e^{C_F f}}{(\beta C_R \alpha)^2}, \qquad 1-\mathrm{e}^{-C_F f} < \gamma < 1$$

It is helpful for analyzing this result to rewrite the expression for γ in terms of $\Delta = \mathrm{e}^{-C_F f}\beta C_R \alpha - 1$, which is the measure of how far the current parameters are from the transition to the one solution scenario (figures 2a and 2b), at which the current framework breaks down.

$$\gamma = 1 - \mathrm{e}^{-C_F f}\left(1 - \frac{\Delta^2}{(\Delta+1)^2}\right)$$

When the approximation is very sparse

$$C_F f \ll 1$$

we can use (2.16) and (2.28) to get

$$P = \frac{1}{\pi\left[\mathrm{erf}^{-1}(1-2\epsilon)\right]^2}\frac{MC_F^2 f}{\gamma + \frac{M}{N}C_F^2 f}$$

2.3.10 Two-subnetworks regime, low noise

We now consider the low noise approximation to the mean field equation for the subnetworks of free units (2.60). This approximation is valid when in addition to (2.56), (2.57) and (2.58)

$$\beta\sigma_H = \beta\alpha\sqrt{C_R(1-\mathrm{e}^{-C_F f})} \gg 1 \tag{2.72}$$

In this approximation, the mean field equation has three solutions if

$$\sqrt{\frac{2}{\pi}}\frac{C_R\alpha\mathrm{e}^{-C_F f}}{\sigma_H} = \sqrt{\frac{2}{\pi}}\frac{\sqrt{C_R}\mathrm{e}^{-C_F f}}{\sqrt{1-\mathrm{e}^{-C_F f}}} > 1$$

This condition is derived analogously to (2.45).

Under the assumption (2.72), the mean field equation (2.60) can then be approximated as

$$\tilde{m}^\nu = \frac{1}{M_f}\sum_{k=1}^{M_f}\mathrm{sign}\left(\beta\left(C_R\alpha e^{-C_F f}\tilde{m}^\nu + H_k^\nu\right)\right) \tag{2.73}$$

As in the section 2.3.7, let us introduce a stochastic function $g(\tilde{m})$

$$g(\tilde{m}) = \frac{1}{M_f}\sum_{k=1}^{M_f}\mathrm{sign}\left(C_R\alpha e^{-C_F f}\tilde{m} + H_k^\nu\right)$$

For small values of the argument $\tilde{m}$, the mean of $g(\tilde{m})$ over different realizations of H_k^ν is approximated as

$$\langle g(\tilde{m})\rangle = \sqrt{\frac{2}{\pi}}\frac{C_R\alpha\mathrm{e}^{-C_F f}\tilde{m} + \mu_H}{\sigma_H}$$

where μ_H and σ_H are given by (2.63) and (2.64).

To compute the variance of $g(\tilde{m})$ we need to know

$$\mathbf{cov}\left(\mathrm{sign}\left(C_R\alpha e^{-C_F f}\tilde{m} + H_k\right), \mathrm{sign}\left(C_R\alpha e^{-C_F f}\tilde{m} + H_p\right)\right)_{k\neq p} \approx \mathbf{cov}\left(\mathrm{sign}\left(H_k\right), \mathrm{sign}\left(H_p\right)\right)_{k\neq p},$$

which is calculated in the Appendix A2, and for large absolute values of the recurrent connectivity, $C_R\mathrm{e}^{-C_F f} \gg 1$ is approximated by

$$\mathbf{cov}(g(\tilde{m}), g(\tilde{m})) = \frac{2}{\pi}C_R\left(\frac{1}{M} + \frac{\varphi_{C_F,f}}{N}\frac{1}{1-e^{-C_f f}}\right) \tag{2.74}$$

Assuming H_k^ν to be Gaussian, we can write

$$g(\tilde{m}) = \sqrt{\frac{2}{\pi}}\frac{C_R\alpha\mathrm{e}^{-C_F f}\tilde{m} + \mu_H}{\sigma_H} + \sqrt{\frac{2}{\pi}C_R\left(\frac{1}{M} + \frac{\varphi_{C_F,f}}{N}\frac{1}{1-e^{-C_f f}}\right)}z^\nu \tag{2.75}$$

where z^ν is a Gaussian variable with zero mean and unit variance.

The statistics of the unstable, close to zero, solution of (2.73) can now be found by plugging in (2.75) as the right hand side of (2.73), and solving for $\tilde{m^\nu}$.

After substituting (2.63) and (2.64) for μ_H and σ_H, we get for the mean and the variance of the unstable solution $\tilde{m}_u$ (assuming $\sqrt{C_R}\mathrm{e}^{-C_F f} \gg 1$):

$$\mu_u = -\sqrt{\frac{2}{\pi}} e^{C_F f} \frac{\langle\sqrt{n}\rangle}{\sqrt{Pf}}$$
$$\sigma_u^2 = \mathrm{e}^{2C_F f}(1 - e^{-C_F f}) \left(\frac{1}{M} + \frac{\varphi_{C_F,f}}{N} \frac{1}{1 - e^{-C_F f}} \right)$$

Using these expressions and (2.40) with M replaced by the number of free units $M_f = M\mathrm{e}^{-C_F f}$, we get for the maximal number of classifiable inputs in the low noise approximation of the two-subnetworks regime:

$$P_{\max} = \frac{\langle\sqrt{n}\rangle^2}{f} \frac{1}{\pi[\mathrm{erf}^{-1}(1-2\epsilon)]^2} \frac{M}{1 + \frac{M}{N}\varphi_{C_F,f}}$$

Note, that this is the same expression as (2.30) for the majority vote scenario (see Results section for an intuitive explanation).

For very sparse representations

$$C_F f \ll 1$$

the expression simplifies to

$$P_{\max} = \frac{1}{\pi[\mathrm{erf}^{-1}(1-2\epsilon)]^2} \frac{MC_F^2 f}{1 + \frac{M}{N}C_F^2 f}$$

3 Results

3.1 The task and the network architecture

To evaluate the performance of different network architecture we consider a task in which the neural network is trained to associate a specific response to each input. The response is expressed by the activity of one output neuron, which could represent a decision, the expected value of an input stimulus or an action. Each input, for example a sensory stimulus, is a pattern of activity across N input neurons. Both, input and output neurons, are either active or inactive and hence the variables representing their activity are binary. Moreover, we assume that the inputs and the outputs are random and uncorrelated. Input neurons are active with probability f, whereas the output neuron is active on average for half of the inputs. Performing this task is equivalent to solving a binary classification problem in which each input is assigned to belong to one of two possible classes. As a measure of the performance of the network we introduce the classification capacity, the maximum number of input patterns that can be correctly classified, and determine how it scales with the total number of neurons of the network. We now consider architectures with increasing complexity and we eventually show that it is possible to design a network in which the number of classifiable inputs is large and it scales linearly with the number of neurons while each neuron has limited connectivity (i.e. the number of connections is fixed in the sense that it does not have to scale with the number of neurons).

3.1.1 Single readout.

The most basic network that we can consider is the one in which the input neurons are directly connected to the output, which is basically the classical perceptron [1] (see figure 1a). The network is trained by modifying the weights w_i that connect each input neuron i to the output. The output activity o^μ in response to stimulus μ is determined by thresholding the weighted sum of the inputs:

$$o^\mu = \text{sign}\left(\sum_{i=1}^{N} w_i \xi_i^\mu - \theta\right)$$

where θ is a threshold and ξ_i^μ is the activity of neuron i when input pattern μ is selected. The weights w_i and the threshold θ are learned to impose that $o^\mu = \eta^\mu$, where η^μ is the desired output in response to stimulus μ. We know from many studies (see e.g. [28, 30]) that the maximum number of random inputs that can be correctly classified scales linearly with the number of input units when $f = 1/2$. This is a very favorable scaling, and actually the optimal one in the benchmark that we consider. Unfortunately, the number of connections of the output neuron is equal to the number of input neurons, and hence when the number of classifiable inputs grows, also the connectivity has to increase accordingly. This is true also in the case of sparse input representations. Indeed, for an arbitrary f, when we used a simple learning rule inspired by [31]

$$w_i = \frac{1}{\sqrt{P}} \sum_{\mu=1}^{P} (\xi_i^\mu - f)\eta^\mu, \tag{3.1}$$

we obtained in the limit for large number of input neurons N and large number of input patterns P that

$$P = \frac{1-f}{2[\text{erf}^{-1}(1-2\epsilon)]^2} N \tag{3.2}$$

where ϵ is the maximum tolerated error.

Notice that the factor containing the coding level of the patterns f cannot change the scaling properties of P, even in the case in which the inputs become very sparse (i.e. when $f \to 0$ as $1/N$). This seems to be in contradiction with the results of [31, 32] in which P can scale as N^2 when the inputs are sparse. However, it is important to remind that the N^2 scaling can be achieved only when both the input and output are sparse and in the cases that we analyzed here the output is dense (i.e. active in half of the cases).

We now consider a different architecture that partially overcome the limitation imposed by the limited connectivty assumption.

3.1.2 Committee machines

Consider now the architecture of Figure 1b in which multiple perceptrons are combined together. We assume that each perceptron has limited connectivity, or more precisely, that when the number of input neurons becomes large (mathematically we consider the limit for $N \to \infty$), the number of input connections per perceptron, C_F, does not increase (i.e. C_F remains finite when $N \to \infty$). As a consequence, each perceptron will sample only a small fraction of the input neurons, and for this reason, it will misclassify most of the inputs when P becomes large ($P \to \infty$). More quantitatively, the fraction of correctly classified inputs will be slightly above chance level (1/2), approximately $1/2 + a/\sqrt{P}$ when P is large, a is a constant.

In this situation, each perceptron is said to be a weak classifier. However, if the responses of different perceptrons are sufficiently independent they can be combined together to perform significantly better than any individual perceptron. Multiple perceptrons combined together make what is called a committee machine. Typically the class of an input is decided by the committee using a majority vote rule: if the majority of perceptrons are active then the output neuron should also be active, otherwise it should be inactive. The majority rule can be easily implemented by summing with equal weights the outputs of all perceptrons.

As mentioned in section 1, adding new readouts without increasing the number of input units N can not increase the classification capacity indefinitely, unless an additional mechanism is introduced to decorrelate the responses of different readouts. Such mechanisms may very well exist in the real brain. For example, one could imagine some local changes of synaptic plasticity during the learning phase, that make different readouts update their connections during presentation of different subsets of patterns. However, in this paper we stick to the simple learning rule (3.1), and do not consider any decorrelation mechanisms. So, in the present contexts, the only way of increasing the classification capacity of the network without reaching the saturation is to increase the number of input units N. Also, in order to satisfy the requirement of limited connectivity, the number of connections converging onto the same readout, C_F can not increase with N, and we need to add new readouts to connect to the newly added input units. We denote the number of readouts (number of committee members) by M and we derive the classification capacity $P_{\max}$ under the assumption that N, M and Pf are large numbers and the C_F connections of every readout are chosen randomly and independently of any other (there will be a random overlap).

If we use the simple local learning rule (3.1), the maximum number of classifiable inputs is:

$$P = \frac{\langle\sqrt{n}_k\rangle^2}{f} \frac{1-f}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2 \pi} \frac{M}{1 + \frac{M}{N}\varphi_{C_F,f}} \tag{3.3}$$

where $\varphi_{C_F,f}$ is of the order of C_F, and depends on C_F and the coding level f, but not on N or M. $\langle\sqrt{n}\rangle$ is the mean of $\sqrt{n}$ over the binomial distribution $\mathbf{B}(C_F - 1, f)$, which is approximately $\sqrt{C_F f}$ in the case $C_F f \gg 1$ (dense regime), and $C_F f$ in the case $C_F f \ll 1$ (ultra-sparse regime). Using also the approximations for $\varphi_{C_F,f}$ in these two cases we get

$$P = \frac{1-f}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2\pi}\frac{C_F M}{1+\frac{2}{\pi}C_F\frac{M}{N}} \quad \text{for } C_F f \gg 1 \tag{3.4}$$

and

$$P = \frac{1}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2\pi}\frac{C_F^2 f M}{1+C_F^2 f\frac{M}{N}} \quad \text{for } C_F f \ll 1 \tag{3.5}$$

So, the dependence of P on the coding level f is weak, unless $C_F f$ becomes smaller than 1. For sparser representations, the capacity becomes proportional to $C_F f$. This is not too surprising because when $C_F f < 1$ a significant proportion of perceptrons will read out only inactive neurons, which are not informative about the input. However, even for very sparse representations the capacity can be restored by increasing the expansion ratio M/N (see figures 4c and 4f).

When N and M grow at the same rate, the number of classifiable patterns increases linearly with N, as in the case of the fully connected single perceptron that we previously considered. However, now the connectivity of each perceptron is just C_F, which does not scale with N or M. This means that it is possible to overcome the limitations of sparse connectivity. Unfortunately, this is not a satisfactory solution as it just moves the problem of limited connectivity to the readout output neuron, which now has to count the votes of all M perceptrons, and hence needs to be connected to M neurons. So again, we will need a number of connections per neuron that grows linearly with N. We will now propose an alternative way of implementing a commitee machine, which is based on the use of recurrent connections and it will not require a fully connected output neuron (see figure 3).

3.2 Committee machines with recurrent connections

One way to count the votes of all perceptrons while respecting the limited connectivity constraint, would be to introduce additional layers of neurons: each neuron in the first layer would count the votes of different C_F perceptrons. The neurons in the second layer would then count the votes of the first layer neurons, and so on. For this architecture, the number of neurons would decrease by a factor C_F in every new layer, leading to total number of neurons which would scale as $\log(M)$ or, equivalently, as $\log(N)$. It is also possible to set up a multi-layer network with the same number of layers in which every layer contains the same number of neurons M. This network would require more neurons, though it would be functionally equivalent to the first one that we considered. An interesting aspect of this architecture is that it can be interpreted as a recurrent network unfolded in time: if one assumes that the network dynamics is discrete in time, then every layer could be seen as the same recurrent network at a different time step. Importantly, the weights of the synaptic connections should be the same for every layer, as it is always the same network but at different time steps. As this network would also

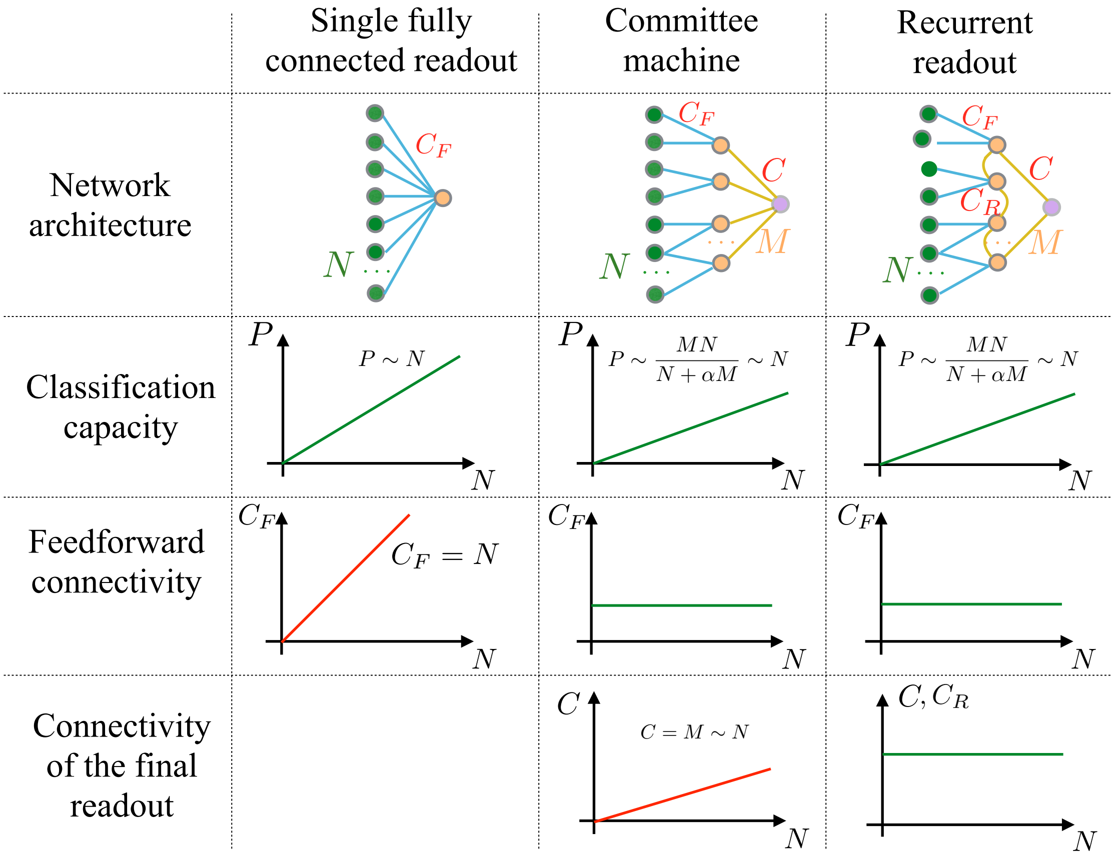

Figure 3. Summary of their scaling properties of the three architectures considered in the study

be functionally equivalent to the first multi-layer network that we discussed, a recurrent network can in principle replace a complex multi-layer readout which would require significantly more neurons.

These considerations induced us to study the architecture represented in Fig.1c: each perceptron of the committee machine is now connected to a randomly chosen set of the others through recurrent connections, whose weights are all the same and equal to α. The number of recurrent connections per perceptron is C_R.

The recurrent dynamics has basically the role of stabilizing only two attractor states of the network: one in which all perceptrons are in the active state, and one in which they are all in the inactive state. These two states represent the two possible responses of the output and correspond to the two classes the input could belong to. The system is equivalent to a spin glass in the ferromagnetic state, or for a more biologically relevant analogy, to the recent decision making network of spiking neurons [33] in which only the two states corresponding to the possible decisions become stable when a sensory stimulus is presented.

Once the network has relaxed into one of the two stable states, it becomes easy to determine the class to which the input belongs, as in principle it is sufficient to read out a single perceptron. However, a single neuron readout would not be robust to noise, and hence we will consider the situation in which a number of different perceptrons are read out. We will show that this number remains finite when N and M become large, which is equivalent to saying that it is possible to construct a network, in which all the neurons, including the output neuron, have limited connectivity and the number of classifiable inputs grows linearly with N.

The number of classifiable inputs is derived analytically in the Methods (section 2) using a mean field approach. This number depends on the parameters that characterize the network architecture (i.e. the number and the connectivity of the different type of neurons), and on the statistics of the inputs that have to be classified. Depending on the assumptions about the parameters, there are different regimes that lead to different analytical expressions.

There are two distinct regimes that depend on whether all the recurrently connected neurons can be considered statistically equivalent or not. We call *uniform* the regime in which all the neurons can be assumed to be equivalent. This is a reasonable assumption in many situations that we discuss below, but it might not be when the number of neurons that receive no feed-forward input, which can behave differently from the others, is sufficient large. This number is negligible when $C_F f \gg 1$. The uniform regime is the first one that we will study systematically. Then we will discuss the non-uniform regime.

3.2.1 The uniform regime

Another factor that determines the parameter regime is the amount of noise that is injected in the neurons. It is important to test the neural system in realistic conditions and to show that it is robust to noise. We introduced noise as in the Hopfield model: the state of each neuron is stochastic and its total synaptic current determines the probability distribution of the states. The noise is characterized by a parameter β, which in the language of statistical mechanics would be the inverse temperature parameter. When β is large, the noise is small and the neurons are basically deterministic. As β goes to zero, the neurons become more noisy and less dependent on the total synaptic input.

As we know from previous studies on attractor neural networks (see e.g. [2]), the noise cannot be too large, otherwise the attractor states remain stable only for a short time. More specifically, the noise should be smaller than the recurrent input when the network already settled in one of the two attractors and most of the presynaptic neurons are in the right state. In the uniform regime, this requirement is expressed as $\beta^{-1} < C_R\alpha$. Moreover, in order to guarantee attractor stability, the recurrent input should also dominate over the feed-forward one. More formally this condition can be expressed as $A < C_R\alpha$, where A is approximately the range in which the feed-forward synaptic input varies when different inputs are presented. It basically determined the selectivity to the inputs in the absence of the recurrent connections (see Methods for more details).

The relation between A and β is less constrained: the network architecture that we are discussing can work in different regimes that depend on how large the noise is compared to the typical amplitude of the feed-forward input.

In the *high noise* regime the noise is so large compared to the feed-forward input ($\beta^{-1} \gg A$) that all the different recurrent neurons can behave similarly (uniform regime) even when the feed-forward input is so sparse ($C_F f \lesssim 1$) that many neurons receive zero input. It is important to remind that in this regime the noise is large compared to A, but still small compared to the recurrent input. The number of classifiable patterns P for the high noise, always uniform regime is given by

$$P = \frac{1-f}{2[\mathrm{erf}^{-1}(1-2\epsilon)]^2} \frac{C_F \frac{M}{N}}{1 + C_F \frac{M}{N} + \frac{(C_R\alpha - \beta^{-1})^2}{C_F f^2 (1-f)}} N \tag{3.6}$$

As in the committee machine case, if the number of input units N and the number of intermediate readouts M are increased in the same proportion, the number of classifiable inputs scales linearly with M or N (see figure 4a). However, now there is not a single neuron that is required to have a connectivity that scales with N, the connectivity of each neuron can remain a finite number even when N and M become arbitrarily large. α is the strength of the recurrent connections and ϵ is the maximum tolerated error rate.

The rate at which P grows with the number of input neurons N (we define P/N as the capacity of the system) depends on the expansion ratio M/N (the number of intermediate readouts per input neuron), the coding level f and the parameters of the recurrent dynamics, β and α. The slope of the curves in Figure 4a, which represents the capacity, increases with the expansion ratio, but only up to a certain point. For $f = 0.5$ (dense representations) the capacity already saturates at $M/N \approx 1$ (see the dashed lines on figure 4a). It saturates at larger M/N for sparser representations ($f = 0.03$). Changing the coding level while keeping the expansion ratio fixed can either increase or decrease the capacity. The dependence of the capacity P/N on the coding level for different values of the expansion ratio is illustrated in figure 4d.

When the noise is low compared to both the recurrent and the feedforward input, the density of the input representations starts playing a crucial role in determining whether the network is in a uniform or non-uniform regime. If the input representation is dense $C_F f \gg 1$, the network is in a uniform regime. As before, all the neurons have the same average activity, but the main source of inhomogeneity is the feedforward input rather than the noise. The number of classifiable inputs in this *uniform low noise* regime is:

$$P = \frac{1-f}{[\mathrm{erf}^{-1}(1-2\epsilon)]^2\pi}\frac{C_F\frac{M}{N}}{1+\frac{2}{\pi}C_F\frac{M}{N}+\left(\sqrt{\frac{2}{\pi}}\frac{C_R\alpha}{\sqrt{C_F f^2(1-f)}}-1\right)^2}N \tag{3.7}$$

This formula is similar to one for the high noise regime. One obvious difference is that the inverse temperature parameter β does not appear because we assumed to be in the low noise limit $\beta \to \infty$. The dependence of the capacity on the expansion ratio M/N is similar to one for the high noise regime. The dependence of capacity P/N on the coding level is summarized by the parts of the plots in which f is large in Figures 4e,f.

When compared to the dense limit ($C_F f \gg 1$) of the majority vote result (3.4), the low noise regime formula (3.7) entails a smaller capacity. The difference comes from the last term at the denominator, which reduces the capacity. This term can be made small by tuning the parameters of the recurrent dynamics C_R or α, but it cannot become zero, because this would correspond to the case in which the recurrent dynamics has only one stable state, which is not suitable for performing a classification task.

3.2.2 Non-uniform regimes

When the noise is small compared to the feed-forward input and the representations are sparse, the uniform approximation is not valid and the recurrent network behaves in a qualitatively different way: for each input pattern, there would be two distinct populations of neurons: the *free neurons*, which receive zero feed-forward input, and hence are not constrained (free) by the input, and all the others, the *input-receivers*. The two populations would be different for different inputs, they would have different activity distributions and would evolve in time differently, although they constantly interact.

In the general case such a regime is intractable with the mean field method, so we need to make the additional assumption that the feed-forward synapses are sufficiently strong relative to the recurrent ones, so that, the non-zero feed-forward inputs are typically larger than the total recurrent inputs in the initial state (before the network reaches the final state when most of the neurons have the same activity). Furthermore, we need to assume that these feed-forward inputs are also much larger than the noise. Under all these assumptions, the state of the input-receivers is determined by the feed-forward input, at least in the initial stages of the dynamics, while the network is deciding which stable state to choose. We then need only to consider the dynamics of the sub-network of free units, treating the recurrent input from the input receivers as a fixed external input. It is this input, that contains the information about the correct classification.

We refer to the described scenario as to the *two-subnetworks* regime. The classification capacity in two-subnetworks scenario depends also on the noise. The noise has to be small in comparison to the feed-forward input, but it can be either small or large when compared to the amplitude of the recurrent input coming from the input-receivers. This comparison distinguishes between the *two-subnetwork low noise* and the *two-subnetwork intermediate noise* regimes.

A third, two-subnetwork intermediate noise regime is realized when the representations are sparse ($C_F f \lesssim 1$) and the noise is small relative to the feed-forward input but large in the subnetwork of free neurons, namely relative to the input into free neurons from the input-receivers. This regime leads to the classification capacity of

$$P = \frac{\langle\sqrt{n}\rangle^2}{f}\frac{1}{\pi[\mathrm{erf}^{-1}(1-2\epsilon)]^2}\frac{M/N}{\gamma + \frac{M}{N}\varphi_{C_F,f}}N \tag{3.8}$$

where $\varphi_{C_F,f}M/N$ comes from the correlations between the input-receivers, $\varphi_{C_F,f}$ is of the order of C_F, and depends on C_F and on the coding level f. $\langle\sqrt{n}\rangle$ is the mean of $\sqrt{n}$ over the binomial distribution $\mathbf{B}(C_F - 1, f)$ and γ is a quantity given by:

$$\gamma = 1 - \frac{2\beta C_R\alpha - e^{C_F f}}{(\beta C_R\alpha)^2} \qquad 1 - \mathrm{e}^{-C_F f} < \gamma < 1 \tag{3.9}$$

which is the smallest (highest capacity) when the network is close to transitioning from three fixed points (2c,d) to one fixed point (2a,b).

In the sparse limit, $C_F f \ll 1$ the expression for P becomes

$$P = \frac{1}{\pi[\mathrm{erf}^{-1}(1-2\epsilon)]^2}\frac{\frac{M}{N}C_F^2 f}{\gamma + \frac{M}{N}C_F^2 f}N \tag{3.10}$$

Figure 4b shows the linear dependence of P on the number of input neurons N for different expansion ratios. We can see that unless the expansion ratio is very high, even for very sparse representations ($f = 0.004$, $C_F f = 0.2$) the capacity grows at a similar rate or even faster compared to the case of dense representations ($f = 0.5$, $C_F f = 25$) in the high noise regime. The dependence of the capacity on the coding level is summarized in figure 4e, where the curves in the low f region correspond to the two-subnetworks intermediate noise regime (3.8), and the segments at high values of f - to the uniform low noise regime (3.7). Apart from the coding level, the only parameter that differs between the two discontinuous parts of the plot for a given expansion ratio is the strength of the recurrent connections α. We decided to choose different αs for two parts because keeping all the parameters the same while satisfying all the conditions for the two-subnetworks intermediate noise regime at low f and uniform low noise regime at high f would have required unrealistically high values of the recurrent connectivity C_R. The dotted lines on the plot represent the results for committee machine with same expansion ratios.

The minimal possible value of γ in 3.9 is $1 - \mathrm{e}^{-C_F f}$ To see this we rewrite the expression 3.9 as

$$\gamma = 1 - \mathrm{e}^{-C_F f}\left(1 - \frac{\Delta^2}{(\Delta+1)^2}\right) \tag{3.11}$$

where

$$\Delta = \mathrm{e}^{-C_F f}\beta C_R\alpha - 1$$

Δ must be positive in order to have three fixed points (figure 2c,d). This implies that the expression in the paranthesis of 3.11 is positive and less than 1, from where it follows that $1 - \mathrm{e}^{-C_F f} < \gamma < 1$.

The lower bound for γ is approximately equal to $C_F f$ in the sparse limit. Plugging this value into 3.8 leads to a capacity that is basically independent from f. This would mean that one can decrease the coding level way below $1/C_F$ without sacrificing the classification performance. However, keeping γ of the order of $C_F f$ requires having Δ of the order of $\sqrt{C_F f}$ or smaller, which entails a progressively finer adjustment of the inverse temperature parameter β as f decreases. In order to have a capacity that does not become infinitesimal when f goes down to f_{min}, β should be adjusted with the a maximum error $\sqrt{C_F f_{min}}/C_R\alpha$.

Figure 6 (dashed lines) shows the capacity P/N as a function of $C_F f$ for a different number of feed-forward connections per input unit $c = MC_F/N$. Along these curves, Δ is kept at a fixed value $\Delta = 0.2$ by choosing a new β for every value of f. Note that the curves are almost flat as long as $C_F f > C_F f_{min}$ where $C_F f_{min} \approx \Delta^2 = 0.04$

Clearly, the capacity in the two-subnetwork intermediate noise regime is larger than in the case of a majority vote committee machine when one assume that the sparseness of the representations is the same (see 3.3). This result is counterintuitive, but it can be explained: in the majority vote scenario, both the input receiving units and the free units contribute to a collective decision, even though the free units carry no information about the class of the input pattern and they actually generate noise as we assume that initially they are in a random state. In contrast, in the recurrent case, the collective state of the network is initially determined mostly by the input receiving units, which then drive the free units to the right state. The noise contained in the initial state of the free units does not affect much the initial relaxation dynamics provided that the noise is sufficient large (relatively low β).

In the case of the majority vote committee machine, the class is decided in only one time step and the initially random free units generate a certain amount of noise that depends on their number. In the case of the recurrent dynamics, the connectivity is sparse and each neuron that participates in it samples the noisy neurons a number of times that depends on the relaxation time. If these neurons can flip randomly at every time step, then their noise is averaged out and the final effect of the free units can be smaller than in the majority vote committee machine.

In the two-subnetwork low noise regime, the capacity is identical to the sparse limit of the majority vote scenario (see 3.3)

$$P_{\max} = \frac{\langle\sqrt{n}\rangle^2}{f}\frac{1}{\pi[\mathrm{erf}^{-1}(1-2\epsilon)]^2}\frac{M/N}{1+\frac{M}{N}\varphi_{C_F,f}}N \tag{3.12}$$

And in the sparse limit, $C_F f \ll 1$:

$$P_{\max} = \frac{1}{\pi[\mathrm{erf}^{-1}(1-2\epsilon)]^2}\frac{\frac{M}{N}C_F^2 f}{1+\frac{M}{N}C_F^2 f}N \tag{3.13}$$

This result is summarized graphically on the figures 4c and 4f. The low f curve segments on the figure 4f correspond to the two-subnetwork low noise regime (identical to the result for the committee machine), and the high f segments - to the uniform low noise regime 3.7. The only parameter that differs between the segments of the same color is the strength of the recurrent connections α.

Another way to visualize the results for the two-subnetwork regime is presented in Figure 6 where we plot the capacity as a function of the product $C_F f$, for the two-subnetwork low noise regime in the sparse limit 3.13 (solid curves) and the same quantity for the two-subnetwork intermediate noise 3.10 (dashed curves). For the intermediate noise regime we tune the amount of noise so that $\beta C_R \alpha \mathrm{e}^{-C_F f}$ is always equal to 1.2. The closer this value is to 1, the larger the advantage of the intermediate noise regime over the low noise regime (majority vote). The different colors represent different values of the number of feed-forward connections per input neuron $c = C_F M/N$. The dotted lines show the maximum possible capacity for a given value of c, which is achieved for the dense limit of majority vote 3.4, assuming that C_F is large and f is small, so that $f \ll 1$, while $C_F f \gg 1$.

It can be seen from Figure 6 that, as discussed before, being in the intermediate noise regime allows for very sparse input representation without sacrificing much the classification capacity, which is only slightly smaller than in the dense case (this ratio increases with c but saturates at $P_{dense}/P_{sparse} = \pi/2$).

This means that the representations can be very sparse, despite the limited connectivity. If there is any other computational reason for preferring sparse representations, then the readout system that we propose can still be used because it can tolerate a high degree of sparseness. This might be the case of the network architecture in the hippocampus in which the representations in the dentate gyrus (DG) are extremely sparse, and the downstream readout neurons in CA3, which would be analogous to the neurons in our intermediate layer, have very sparse connectivity. Although we know that moderate sparseness ($f \sim 0.1$) can be highly beneficial for generalization [34], we do not know why the representations in the DG are so sparse ($f \sim 0.01$). However, our study shows that this elevated degree of sparseness does not necessarily impair the ability of the readout to perform efficiently a classification task.

3.3 Simulation Results

Figures 5 show the results of numerical simulations run to verify the predictions of the above calculations. The two plots correspond to two different regimes characterized by different coding level of the input patterns representation. Figure 5a corresponds to dense input representation $C_F f = 10 \gg 1$. The theoretical predictions for the majority vote scheme (committee machine) and the recurrent readout are represented by the two dashed lines, and the solid lines with the error bars depict the results of the simulations in the two scenarios. The case of sparse input representations $C_F f = 1$ is presented on Figure 5b. Paradoxically, the recurrent readout scenario leads to higher classification capacity compared to the majority vote, which is confirmed by the simulation.

The simulation plots were obtained as follows. We fix the required accuracy of the classification

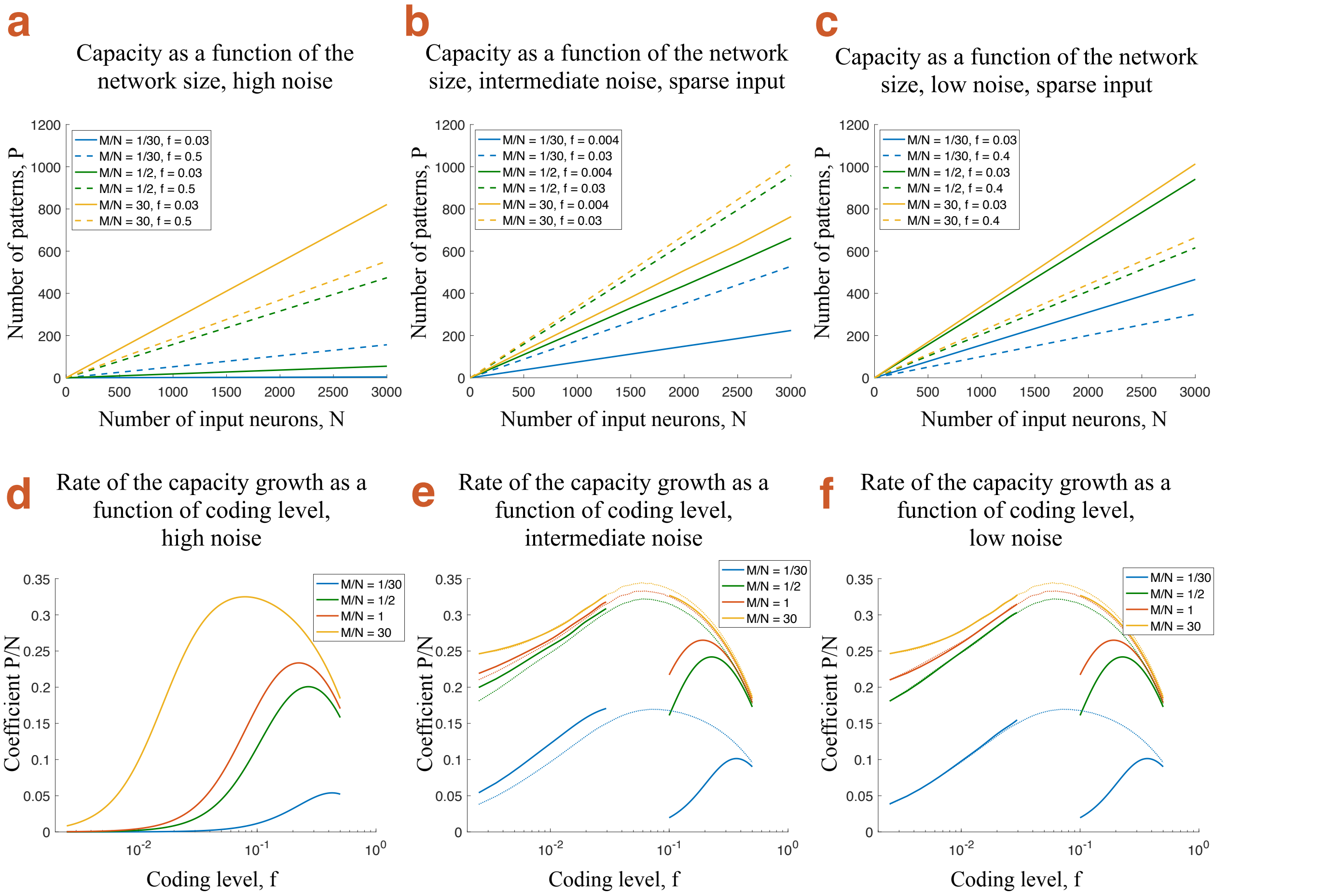

Figure 4. **a**) The dependence of the classification capacity of the recurrent readout on the number of input units N, when the number of readouts M is increased proportionally to N. High noise regime, see equation (3.6). Solid curves correspond to sparse input representation, coding level $f = 0.03$ ($C_F f = 1.5$) and dashed curves - to dense representation, $f = 0.5$. Different colors represent different expansion ratios M/N. The other parameters the same as in (d). **b**). Same as (a) but for the case of intermediate noise and sparse input (two-subnetworks intermediate noise regime), see formula (3.8). Solid curves: $f = 0.004$ ($C_F f = 0.2$), dashed curves: $f = 0.03$. The parameters are the same as for the low f segment of (e). **c**) Same for the low dynamical noise. Solid curves correspond to sparse input representation, $f = 0.03$, two-subnetwork low noise regime, see 3.12. Dashed curves are for dense representations $f = 0.4$ ($C_F f = 20 \gg 1$) in the uniform low noise regime, see 3.7. The parameters are the same as in (f), the values of α differ for the solid and dashed curves. **d**) The dependence of the rate of the capacity growth on the coding level of input representation f. High noise regime, see equation (3.6). Different curves correspond to different expansion ratios M/N. The other parameters are $C_F = 50$, $C_R = 700$, $\alpha = 0.035$, $\beta = 0.05$ and $\epsilon = 0.05$ **e**) Same for the intermediate noise regime. Different colors correspond to different expansion ratios M/N, the curves are discontinuous because different formulas are valid for sparse and dense representations. See (3.8) and (3.7). The parameters are $C_F = 50$, $C_R = 700$, $\alpha = 0.000075$ for $f < 0.03$ and $\alpha = 0.006$ for $f > 0.1$, $\beta = 100$ and $\epsilon = 0.05$. The dotted curves show the committee machine result. **f**) Same as d) and e) for low noise regime. See equations (3.12) and (3.7). Parameters: $C_F = 50$, $C_R = 700$, $\alpha = 0.00075$ for $f < 0.03$ and $\alpha = 0.006$ for $f > 0.1$, $\beta = 9000$ and $\epsilon = 0.05$. The dotted curves show the committee machine result.

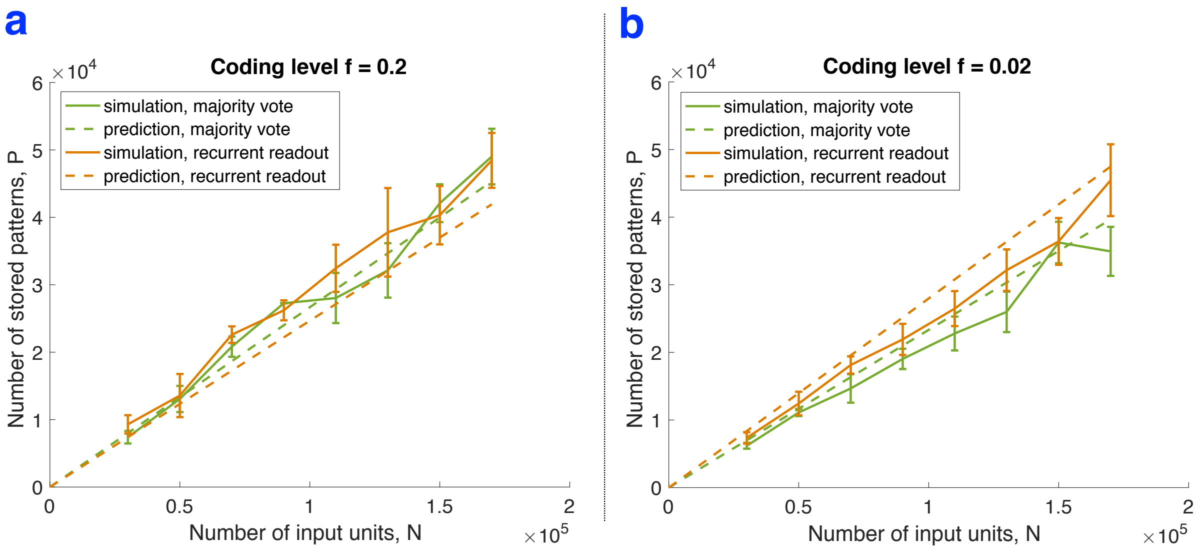

Figure 5. **a**. Simulation results (solid lines) and theoretical predictions (dashed lines) for the case of dense input representations, $C_F f = 10$. The green curves correspond to majority vote scenario (committee machine) and the orange - to the recurrent readout in the uniform regime with relatively high noise. **b**. Same for the case of sparse input representation, $C_F f = 1$. The recurrent dynamics of the intermediate layer is in two-subnetwork regime with relatively high noise.

at $1 - \epsilon = 0.9$ and compute the predicted classification capacity. We fix the number of feedforward connections per readout at $C_F = 50$. For each number of the input units N, we chose the corresponding number of the intermediate readouts $M = N/30$, and train the network with the set of P_1 random and uncorrelated patterns, where P_1 is equal to the theoretically predicted classification capacity. We then test the classification performance on the subset of 500 learned patterns and recorde the obtained accuracy $1 - \check{\varepsilon}$. At the next step we train the network with the same number of input units and recurrent readouts (and the same structure of the feedforward connectivity) on the new set of $P_2 = P_1 \left[\frac{\mathrm{erf}^{-1}(1-2\check{\epsilon})}{\mathrm{erf}^{-1}(1-2\epsilon)}\right]^2$ random patterns. Here $1 - \check{\epsilon}$ is the classification accuracy achieved for the set of P_1 patterns, and $1 - \epsilon$ is the required accuracy. We repeat the procedure 10 times, and keep only those runs, where the accuracy differed from the required one by no more then 2 percent ($|\check{\epsilon} - \epsilon| < 0.02$). We then compute the mean and the standard error of the corresponding values of P. For the recurrent readout scenario, the recurrent connectivity was random with all the connections having of the same strength, and the number of connections per unit being fixed at $C_R = 200$. The connectivity matrix was chosen to be symmetrical and the recurrent dynamics run for 30 steps of synchronous update. The parameters of the recurrent dynamics for the plot of figure 5a (dense input) were $\beta = 0.5$ and $\alpha = 0.015$, and for the figure 5b (sparse input): $\beta = 33$ and $\alpha = 0.0005$

3.4 Optimizing the architecture

3.4.1 Optimizing the architecture under the constraint on the total number of long-range connections

One interesting question to ask, given the capacity results (3.6) - (3.13) is how to maximize the classification capacity given the constraint on the total number of feedforward connections, that are seen as corresponding to the long-range connections in the biological brain. To address this question we can think of the number of inputs N and the total number of long-range connections $C_F M$ as fixed and ask what value of C_F (or M) will maximize the classification capacity $P_{\max}$.

For the majority vote scenario, the separate dependence of the capacity on C_F disappears once $C_F f$ becomes much larger than 1 (see 3.4). So, for a fixed value of the coding level f, once $C_F f \gg 1$, regrouping feedforward connections, i.e. changing C_F while keeping $c = C_F M/N$ constant does not affect the classification capacity.

The same is true for the uniform low noise regime 3.7, which is valid only if $C_F f \gg 1$. Even though C_F enters the formula in the last term of the denominator without being multiplied by M, it enters in combination with the parameters of the recurrent dynamics $C_R \alpha$ that can be adjusted to achieve the optimal performance for any C_F.

In the case of high noise and uniform regime 3.6, which is applicable for both dense and sparse representations, increasing C_F, while keeping $C_F M$ constant increases the capacity unless the last term in the denominator is already negligible, which happens later for low values of f and faster for large values of c.

To see what happens for smaller values of C_F, when $C_F f \ll 1$, we can look again at the figure 6 (see section 3.2 and figure captions), but think of f as being fixed and C_F as changing along the horizontal axis. The dotted lines now represent the limit of the ratio P/N as $C_F \to \infty$ for different numbers of feedforward connections per input neuron, c, assuming the majority vote scheme (3.4 if f is negligible compared to 1). One can see that in the low noise regime for low values of c as long as C_F is large enough to expect one active input per readout, the classification capacity is already only slightly smaller than the high C_F limit. Going to intermediate noise regime with tuned β ($\beta C_R \alpha \mathrm{e}^{-C_F f} \gtrsim 1$) decreases this difference and allows to decrease C_F without sacrificing the performance. For high values of c the decrease in performance at $C_F f = 1$ compared to large C_F limit is more profound, and the advantage of the intermediate noise regime becomes apparent only at even lower values of $C_F f$.

3.4.2 Optimizing the architecture under the constraint on the total number of units

Another biologically expired constraint that we analyze is the constraint on the total number of units. The question, which we can answer given the results (3.6) - (3.13), is how to divide the units between input and readout layers in order to maximize the classification capacity. Asking this question one should keep in mind, however, that the analysis is valid only for random and uncorrelated input patterns, so by formulating the problem like this, we have assumed that there is enough external

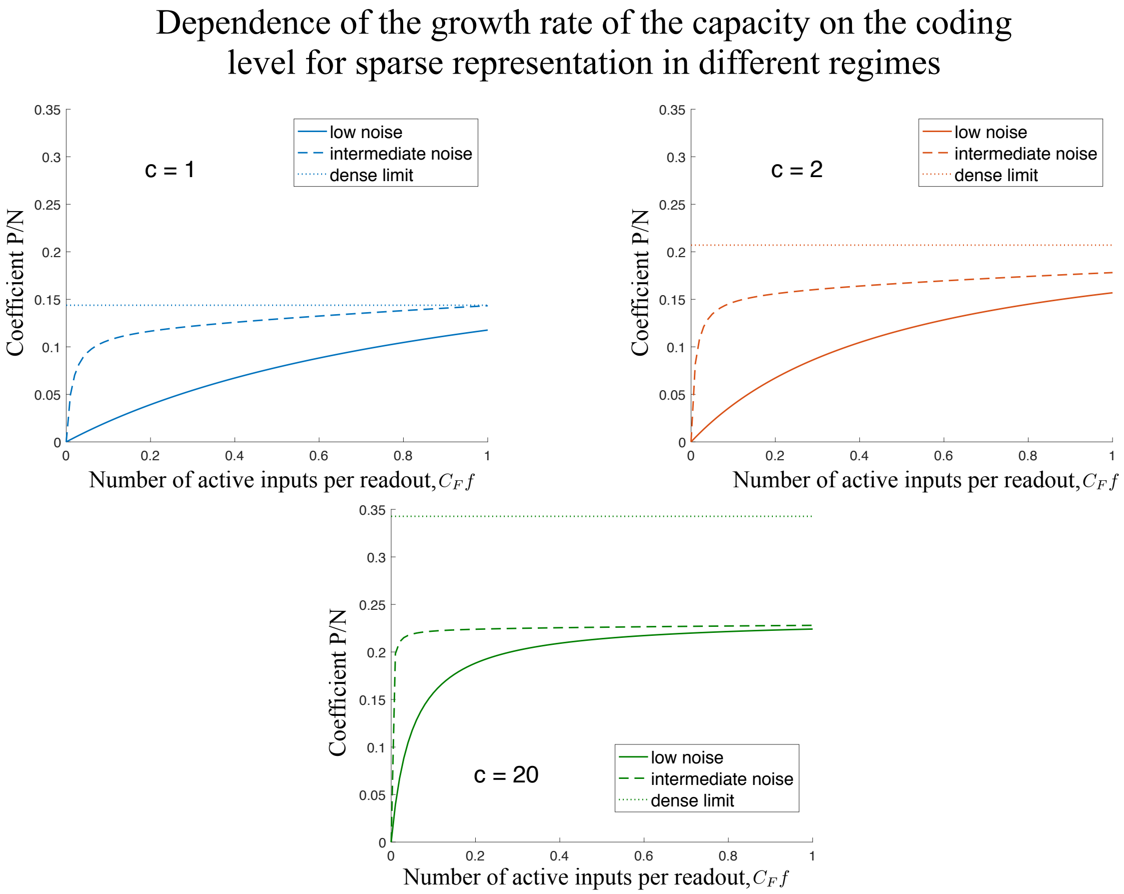

Figure 6. The coefficient P/N as a function of sparsity of the input representation, expressed as $C_F f$ for the sparse limit of two-subnetwork low and intermediate noise regimes, 3.13 and 3.10, $\epsilon = 0.05$. Different colors correspond to different values of $c = MC_F/N$, which is the number of outgoing connections per input neuron. Solid curves correspond to the low noise regime, which is equivalent to the sparse limit of the majority vote result (see 3.13 and 3.5), dashed lines represent the result for the intermediate amount of noise 3.10. The inverse temperature parameter β is different for different values of $C_F f$ so as to keep $\beta \alpha K \mathrm{e}^{-C_F f} = 1.2$. The dotted lines show the maximum capacity possible for given c. This is computed from the dense limit of the majority vote result 3.4, assuming that C_F is large enough so that when f is small enough to be neglected in comparison 1 in the numerator of 3.4, we are still in the dense regime $C_F f \gg 1$.

information to generate N uncorrelated input components.

It is straightforward to derive the optimal expansion ratio M/N from the formulas (3.6) - (3.13) under the constraint $M + N = const$. In the uniform regime (see equations (3.6) and (3.7)), if the parameters of the recurrent dynamics are not too far from the optimal ones, the expansion ratio that maximizes the capacity can be approximated up to the factors of order one by

$$\frac{M}{N} \approx \frac{1}{\sqrt{C_F}}$$

This corresponds to more readouts than one readout per every C_F inputs in non-overlapping design. To be more precise, a typical input is connected to approximately $C_F M/N \approx \sqrt{C_F}$ readouts. However, the number of readouts M is still much smaller than the number of inputs N.

For the two-subnetwork low noise regime in the sparse limit $C_F f \ll 1$, the optimal expansion ratio is approximated by

$$\frac{M}{N} \approx \frac{1}{\sqrt{C_F}\sqrt{C_F f}}$$

This corresponds to $1/\sqrt{f}$ connections per input unit. For sparser representataions the optimal proportion of units in the readout layer increases.

For intermediate noise

$$\frac{M}{N} \approx \frac{1}{\sqrt{C_F}}\sqrt{\frac{\gamma}{C_F f}}$$

where γ is given by 3.9

3.5 Multinomial Classification

We now turn to a more difficult problem of classifying the inputs into more than two categories. The scheme presented above can be generalized in a straightforward way to serve as multinomial classifier. We first present straightforward method and show that changing to the case of multiple classes does not substantially change the classification capacity of the network. We later discuss a more realistic scenario for which we can not compute the capacity analytically, but we demonstrate with the simulations that decrease in the capacity is less than two-fold and, most importantly, the linear scaling with the network size is preserved.

3.5.1 Structured output

The immediate generalization of the recurrent readout scheme to multinomial classification task is to introduce several population of the intermediate readouts, each of which would correspond to one class. The recurrent connectivity within a population would be as described before, while no recurrent connections would exist between the units belonging to distinct populations. The desired output for each class is then structured so that the population corresponding to the given class is active while the others are inactive. The final readout has now to be replaced by multiple final readout, one for each class. Their connectivity can still be sparse and random, but the sign of the connections would have to be adjusted based on whether it comes from the population selective for the same class as the given final readout or not.

The classification capacity can now be computed in the same way as above, by noticing that each population is now doing a binary classification, selecting for one out of L classes. The only difference is that the proportion of 'positive' patterns (the output sparseness) is now $y = 1/L$ instead of $1/2$. The capacity formula for the case of sparse output is derived in the Methods section 2 and it differs from the capacity for a dense case by a factor, that depends on y.

$$P_y(N, M) = \frac{1}{4y(1-y)} P_{0.5}(N, M)$$

It should be noted, that the number of intermediate readouts M, entering this formula is the number of units in the population selective for a particular class. So, if total number of intermediate readouts is M_{total}, and all population have equal size, it is $M = M_{total}/L = yM_{total}$, that should enter the formulas for the capacity. So, in terms of the total number of intermediate units, in the regime in which formula (3.8) was derived, we have

$$P = \frac{L}{4(L-1)} \frac{1}{\pi[\mathrm{erf}^{-1}(1-2\epsilon)]^2} \frac{M_{total} C_F^2 f}{\gamma + \frac{M_{total}}{LN} C_F^2 f}$$

Where γ is given by (3.9). There are two differences of this result compared to the binary classification (3.8). The first is the prefecture, which is equal to $1/2$ for the case of two classes ($L = 2$). This is the reflection of the fact, that when only two classes are possible, the current scheme is redundant - when the first population is active, the other is not, and vice versa. In the limit of large number of classes, the prefactor is equal to $1/4$. The other difference is in the second term in the denominator which rescales N, the number of the input units. Namely, given the number of intermediate readouts and all other parameters, the number of input units in the multinomial classification scheme, required to achieve the same capacity as in the binary classification, is L times smaller. This is because there is no interference between the readouts belonging to different classes.

For the case of two classes ($L = 2$), the result is the same as (3.8) if γ is small.

3.5.2 Random output

Another, more realistic scenario is to assign the output patterns that correspond to each of L classes randomly and train the existing recurrent connections with a plausible learning rule.

$$J_{kl} = \zeta_k^a \zeta_l^a \tag{3.14}$$

Where ζ_k^a is the output patterns corresponding to the class a, $(a = 1 \cdots L)$.

We do not analyze this scenario analytically, but we present the results of the simulations that show the linear scaling of the classification capacity with the number of input neurons N.

Figure 7 shows the results of the simulation for 5-way classification ($L = 5$) of dense input patterns with high dynamical noise (the parameter values are given in the figure caption). The dashed line

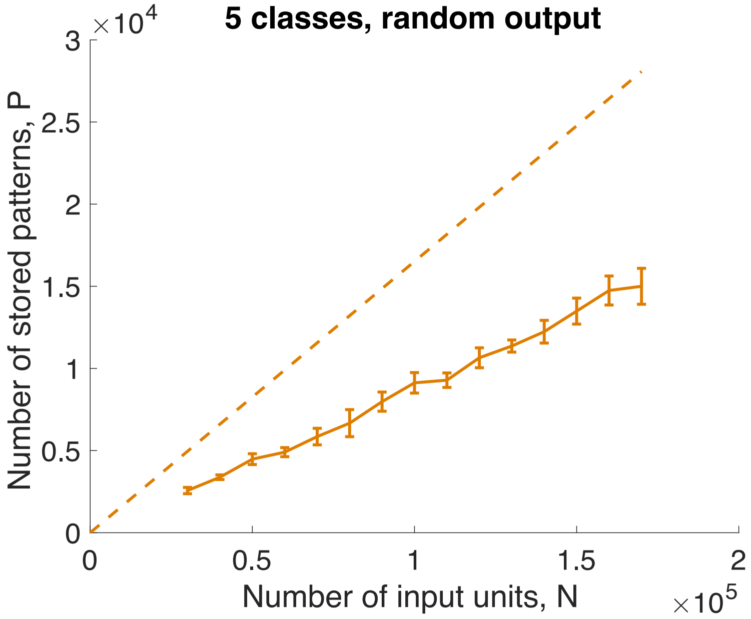

Figure 7. The results of the simulation for multinomial classification. The output patterns corresponding to $L = 5$ classes are chosen randomly with the coding level $y = 1/2$. The recurrent connectivity is sparse and the strength of the synapses are trained with the learning rule 3.14. The network of recurrently connected intermediate readouts is in the high noise regime with dense input representations $(C_F = 50, f = 0.2, C_R = 200, \alpha = 0.015, \beta = 0.5)$. The dashed line is the estimation of the capacity from the formula (3.6) assuming two equal size subpopulations of the readouts.

on the figure indicates the capacity given by the formula (3.6) assuming that the population of M intermediate readouts is split into two segregated subpopulations, whose activity is opposite in all the output patterns.

4 Discussion

We presented a model network based on perceptrons that satisfies the limited connectivity constraint but whose classification capacity still scales linearly with the size of the network. The limitations on classification capacity of the individual perceptrons imposed by the limited connectivity are overcome by means of collective decision mechanism that is similar to the majority vote in committee machines. The difference from the standard committee machine is that the voting procedure is implemented through recurrent attractor dynamics of the network of intermediate classifiers (committee members). This allows to bring the collective decision to the level of single unit activity without violating the limited connectivity constraint.

Interestingly, the proposed recurrent readout scheme can outperform the majority vote of the committee machine of sparsely connected perceptrons for the case of sparse input representations (see sections 3.2.2 and 2.3.9). For the majority vote scheme, the classification capacity drops drastically when the input representations become very sparse because the fraction of classifiers whose inputs are all silent becomes substantial (the difference from the results of [31] is explained by the fact that the output is still dense in the present case). However, for the recurrent readout in the certain parameter regime, the classification capacity can be kept high even for very sparse representations. The lower limit on the coding level f, below which the capacity drops is determined by the amount of noise in the recurrent dynamics, the expansion ration and the number of feedforward connections per perceptron (see figure 4).

This work was largely motivated by the question of what is the advantage of sparse representations, posed by the observations in the mammalian dentate gyrus. We show, that for the recurrent readout under intermediate noise condition (see sections 3.2.2 and 2.3.9), the classification capacity stays within a reasonable range even when the expected number of active units per perceptron is less than 1 (see figures 4e, 4f and 6). This result is complimentary to [34], where the authors show that correlated input patterns are more efficiently separated by introducing a randomly connected intermediate layer with sparse activity. We show that the resulted sparse representations can be read out while respecting the limited connectivity constraint.

A crucial aspect that allows to implement the majority vote by means of recurrent dynamics, which was omitted so far, is how the network of the recurrently connected perceptrons can be initiated at the state with unbiased average activity $m_0 = 0$ before every classification (see section 2.3.3). Here we propose one of the ways this initialization can be realized in a biological network.

We assume that before the input pattern for classification is presented to the input layer, the input layer is spontaneously active. This spontaneous activity generates a feedforward input h_k^{sp} to the layer of recurrently connected perceptrons that is chosen from a distribution other than h_k^{μ}, which is the feedforward current when an input pattern is presented (see 2.10). Consequently, it is possible to have the disordered state ($m = 0$) as the only stable state of the recurrent network (see figure 2a) before the presentation of the pattern. There are two conditions on the statistics of h_k^{sp} that are required to have $m = 0$ as the only stable state of the system in the mean field approximation. The first requirement is that h_k^{sp} has zero expectation value, which is satisfied if the patterns of spontaneous activity are not correlated with the training patterns. The second requirement is that the standard deviation of the distribution is large enough, to make the slope of the sigmoidal curve of figure 2 smaller than 1. For instance, in the uniform regime (see section 2.3.5), the latter requirement is $\sigma_h^{sp} > \sqrt{\frac{2}{\pi}} C_R \alpha$, where σ_h^{sp} is the standard deviation of the feedforward current due to spontaneous activity. This requirement is opposite to the condition (2.46) of having three solutions when an input pattern is presented.

5 Acknowledgements

LK acknowledges support from the grants ANR-10-LABX-0087 IEC, ANR-10-IDEX-0001-02 PSL

6 Appendix

6.1 A1

In this section we derive the variance of the feedforward current, averaged over the intermediate units

$$\bar{h}^{\nu} = \frac{1}{M}\sum_{k=1}^{M} h_k^{\nu} \tag{6.1}$$

which is used to obtain (2.42).

The variance of $\bar{h}^{\nu}$ is contributed by the diagonal terms and the non-diagonal terms, in the limit $M \to \infty$ approximated by

$$\mathbf{cov}(\bar{h}^{\nu}, \bar{h}^{\nu}) = \frac{1}{M}\mathbf{cov}(h_k^{\nu}, h_k^{\nu}) + \mathbf{cov}(h_k^{\nu}, h_l^{\nu})_{k\neq l} \tag{6.2}$$

For the non-diagonal terms, neglecting $\frac{1}{P}$ corrections coming from the signal and using representation (2.21), we find

$$\mathbf{cov}(h_k^{\nu}, h_l^{\nu})_{k\neq l} = f(1-f)\langle n_{kl}\rangle \tag{6.3}$$

Here $\langle n_{kl}\rangle$ is the expectation value of the number of common active input neurons for readouts k and l (see details in section 2.2.2).

Therefore, comparing with (2.9) we find

$$\mathbf{cov}(h_k^{\nu}, h_l^{\nu})_{k\neq l} = \mathbf{cov}(h_k, h_k)\frac{\langle n_{kl}\rangle}{\langle n_k\rangle} \tag{6.4}$$

The $\langle n_{kl}\rangle$ in the limit $N, M \to \infty$ and finite C_F, f can be estimated as

$$\langle n_{kl}\rangle = fN\left(\frac{C_F}{N}\right)^2 = \frac{C_F^2}{N}f \tag{6.5}$$

which gives

$$\frac{\langle n_{kl}\rangle}{\langle n_k\rangle} = \frac{C_F}{N} \tag{6.6}$$

and therefore

$$\mathbf{cov}(h_k^{\nu}, h_l^{\nu})_{k\neq l} = \frac{C_F}{N}\mathbf{cov}(h_k, h_k) \tag{6.7}$$

Hence (6.2) reduces to

$$\mathbf{cov}(\bar{h}^{\nu}, \bar{h}^{\nu}) = \left(\frac{1}{M} + \frac{C_F}{N}\right)\sigma_h^2 \tag{6.8}$$

and implies (2.42).

6.2 A2

In this section we derive the formula (2.74) for the covariance of the signs of the external (from input receiving units) currents into two different free units in the two-subnetworks regime

$$\mathbf{cov}\left(\text{sign}\left(C_R\alpha e^{-C_F f}\tilde{m} + H_k\right), \text{sign}\left(C_R e^{-C_F f}\tilde{m} + H_p\right)\right)_{k\neq p}. \tag{6.9}$$

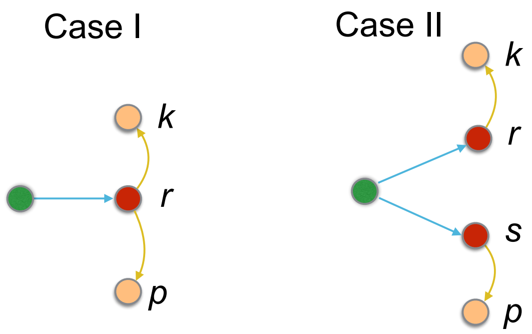

Figure 8. Two sources of input correlations for the subnetwork of free units (orange circles), referred in the text as case I and case II. On the left diagram two free units are connected to the same input receiving unit in the intermediate layer (red circle). On the right diagram there is no input receiving unit that is connected to both free units, but the correlation arises from an active unit in the input layer (green circle), which is connected to the two free units indirectly.

Following the section 2.3.10, we introduce a notation

$$g_k(\tilde{m}) = \text{sign}\left(C_R \alpha e^{-C_F f}\tilde{m} + H_k\right)$$

and without loss of generality assume $\alpha = 1$.

There are two cases of contributions to the correlation, that we will call case I and case II, see figure 8.

The case I contribution to this correlation comes from the free units k and p being connected to the same input receiving unit r. We neglect the probability that the overlap will be over more than one input receiving unit since we keep connectivity C_R fixed when we scale the number of units M. To the leading order in C_R/M, the probability of this situation to occur for a randomly chosen pair of free units is

$$p_{\text{I}} = \frac{C_R}{M} C_R (1 - e^{-C_F f}) \tag{6.10}$$

This is because a typical free unit is connected to $C_R(1-e^{-C_F f})$ out of $M(1-e^{-C_F f})$ input receiving units.

The case II contribution comes from the possibility that there is an input layer unit that is active and connects to both free units in a randomly chosen pair via different input receiving units. The approximate robability of this to happen, assuming C_F/N is small, is given by

$$p_{\text{II}} = f\frac{C_F^2}{N}C_R^2 \tag{6.11}$$

To derive this probability, recall that the probability of any two intermediate units to be connected to the same active input unit is fC_F^2/N, and there are C_R^2 pairs of intermediate units (red units on figure 8) connected to the given pair of free units (orange units). The probability that both units in this pair are input receiving units is already taken into account by the factor fC_F^2/N.

In the case I the relevant correlation is

$$\begin{aligned}\mathbf{cov}^{\mathrm{I}}(g_k(\tilde{m}), g_p(\tilde{m})) = \mathbf{cov}\Bigg(&\operatorname{sign}\left(C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_r) + \sqrt{C_R(1-e^{-C_F f})}z_k\right), \\ &\operatorname{sign}\left(C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_r) + \sqrt{C_R(1-e^{-C_F f})}z_p\right)\Bigg)_{k\neq p} = \\ = \frac{2}{\pi}\mathbf{cov}\left(\frac{C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_r)}{\sqrt{C_R(1-e^{-C_F f})}}, \frac{C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_r)}{\sqrt{C_R(1-e^{-C_F f})}}\right) &= \frac{2}{\pi}\frac{1}{C_R(1-e^{-C_F f})}\end{aligned} \tag{6.12}$$

The indices of the units correspond to those on figure 8. The number of recurrent connections per unit is assumed to be large, so that $C_R - 1 \approx C_R$. z_k and z_p are independent Gaussian variables with zero means and unit variances. The Gaussian assumption is valid if C_R is a large number. We also assumed $C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_r) \ll \sqrt{C_R(1-e^{-C_F f})}$ to get from the first line to the second.

In the case II the relevant correlation

$$\begin{aligned}&\mathbf{cov}^{\mathrm{II}}(g_k(\tilde{m}), g_p(\tilde{m})) = \\ &= \mathbf{cov}\Bigg(\begin{aligned}&\operatorname{sign}\left(C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_r) + \sqrt{C_R(1-e^{-C_F f})}z_k\right), \\ &\operatorname{sign}\left(C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_s) + \sqrt{C_R(1-e^{-C_F f})}z_p\right)\end{aligned}\Bigg)_{k\neq p, r\neq s} = \\ &= \frac{2}{\pi}\left\langle\frac{C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_r)}{\sqrt{C_R(1-e^{-C_F f})}}\frac{C_R e^{-C_F f}\tilde{m} + \operatorname{sign}(h_s)}{\sqrt{C_R(1-e^{-C_F f})}}\right\rangle = \\ &= \frac{2}{\pi}\frac{1}{C_R(1-e^{-C_F f})}\frac{2}{\pi}\left\langle\tan^{-1}\sqrt{\frac{1}{(n_r+1)(n_s+1)-1}}\right\rangle_{n_r, n_s \in \mathbf{B}(C_F-1, f)} = \\ &= \frac{2}{\pi}\frac{1}{C_R(1-e^{-C_F f})}\frac{\varphi_{C_F,f}}{fC_F^2}\end{aligned} \tag{6.13}$$

where n_r and n_s are from binomial distribution on $C_F - 1$ trials with probability f. The correlation $\langle\operatorname{sign}(h_r)\operatorname{sign}(h_s)\rangle$ was computed in (2.25).

Now we can compute (6.9) in the leading order as $p_{\mathrm{I}}, p_{\mathrm{II}}$ probability weighted sum of the contributions from case I and case II:

$$\begin{aligned}\mathbf{cov}(g_k(\tilde{m}), g_p(\tilde{m})) = p_{\mathrm{I}}\mathbf{cov}^{\mathrm{I}}(g_k(\tilde{m}), g_p(\tilde{m})) + p_{\mathrm{II}}\mathbf{cov}^{\mathrm{II}}(g_k(\tilde{m}), g_p(\tilde{m})) = \\ = \frac{2}{\pi}\frac{C_R}{M} + \frac{2}{\pi}\frac{\varphi_{C_F,f}}{N}\frac{C_R}{1-e^{-C_f f}}\end{aligned} \tag{6.14}$$

At the diagonal terms we have simply

$$\mathbf{cov}(g_k(\tilde{m}_u), g_k(\tilde{m}_u)) = 1 \tag{6.15}$$

Alltogether, combining the contribution from diagonal and non-diagonal terms as in (6.2) we find

$$\mathbf{cov}(g(\tilde{m}), g(\tilde{m})) = \frac{1}{M_f} + \frac{2}{\pi}C_R\left(\frac{1}{M} + \frac{\varphi_{C_F,f}}{N}\frac{1}{1-e^{-C_f f}}\right) \tag{6.16}$$

We will assume that $C_F f \lesssim 1$ so that $M_f \simeq M$ and that even though C_R does not scale linearly with M, N, P still

$$C_R e^{-C_F f} \gg 1 \tag{6.17}$$

then we can, in fact, drop the diagonal term in (6.16) and take the approximation

$$\mathbf{cov}(g(\tilde{m}), g(\tilde{m})) = \frac{2}{\pi} C_R \left(\frac{1}{M} + \frac{\varphi_{C_F,f}}{N} \frac{1}{1 - e^{-C_f f}} \right) \tag{6.18}$$

References

[1] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[2] Daniel J Amit. *Modeling brain function: The world of attractor neural networks*. Cambridge University Press, 1992.

[3] Yasser Roudi and Peter E Latham. A balanced memory network. *PLoS computational biology*, 3(9):e141, 2007.

[4] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature Reviews Neuroscience*, 13(5):336–349, 2012.

[5] Liam J Drew, Stefano Fusi, and René Hen. Adult neurogenesis in the mammalian hippocampus: why the dentate gyrus? *Learning & Memory*, 20(12):710–729, 2013.

[6] David G Amaral, Norio Ishizuka, and Brenda Claiborne. Neurons, numbers and the hippocampal network. *Progress in brain research*, 83:1–11, 1990.

[7] MW Jung and BL McNaughton. Spatial selectivity of unit activity in the hippocampal granular layer. *Hippocampus*, 3(2):165–182, 1993.

[8] MK Chawla, JF Guzowski, V Ramirez-Amaya, P Lipa, KL Hoffman, LK Marriott, PF Worley, BL McNaughton, and CA Bavs. Sparse, environmentally selective expression of arc rna in the upper blade of the rodent fascia dentata by brief spatial experience. *Hippocampus*, 15(5):579–586, 2005.

[9] Nils J Nilsson. *Learning machines: foundations of trainable pattern-classifying systems*. McGraw-Hill, 1965.

[10] Bambang Parmanto, Paul W Munro, and Howard R Doyle. Reducing variance of committee prediction with resampling techniques. *Connection Science*, 8(3-4):405–426, 1996.

[11] C Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn. *Springer, New York*, 2007.

[12] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms.* CRC Press, 2012.

[13] Michael Kearns. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105, 1988.

[14] JNK Rao and Kathleen Subrahmaniam. Combining independent estimators and estimation in linear regression with unequal variances. *Biometrics*, pages 971–990, 1971.

[15] Bradley Efron and Carl Morris. Combining possibly related estimation problems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 379–421, 1973.

[16] Donald B Rubin and Sanford Weisberg. The variance of a linear combination of independent estimators using estimated weights. *Biometrika*, 62(3):708–709, 1975.

[17] Edwin J Green and William E Strawderman. A james-stein type estimator for combining unbiased and possibly biased estimators. *Journal of the American Statistical Association*, 86(416):1001–1006, 1991.

[18] Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.

[19] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[20] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[21] Leo Breiman. Bias, variance, and arcing classifiers. 1996.

[22] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996.

[23] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[24] GJ Mitchison and RM Durbin. Bounds on the learning capacity of some multi-layer networks. *Biological Cybernetics*, 60(5):345–365, 1989.

[25] C Kwon and JH Oh. Storage capacities of committee machines with overlapping and non-overlapping receptive fields. *Journal of Physics A: Mathematical and General*, 30(18):6273, 1997.

[26] Rémi Monasson and Riccardo Zecchina. Weight space structure and internal representations: a direct approach to learning and generalization in multilayer neural networks. *Physical review letters*, 75(12):2432, 1995.

[27] R Urbanczik. Storage capacity of the fully-connected committee machine. *Journal of Physics A: Mathematical and General*, 30(11):L387–L392, 1997.

[28] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965.

[29] Murray Geller and EW Ng. A table of integrals of the error function. II. Additions and corrections. *J. Res. Natl. Bur. Stand*, 75:149–163, 1971.

[30] E. Gardner. Maximum storage capacity in neural networks. *EPL (Europhysics Letters)*, 4(4):481, 1987.

[31] MV Tsodyks and MV Feigel'Man. The enhanced storage capacity in neural networks with low activity level. *Europhysics Letters*, 6(2):101–105, 1988.

[32] Daniel J Amit and Stefano Fusi. Learning in neural networks with material synapses. *Neural Computation*, 6(5):957–982, 1994.

[33] Xiao-Jing Wang. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, 36(5):955–968, 2002.

[34] Omri Barak, Mattia Rigotti, and Stefano Fusi. The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off. *The Journal of Neuroscience*, 33(9):3844–3856, 2013.