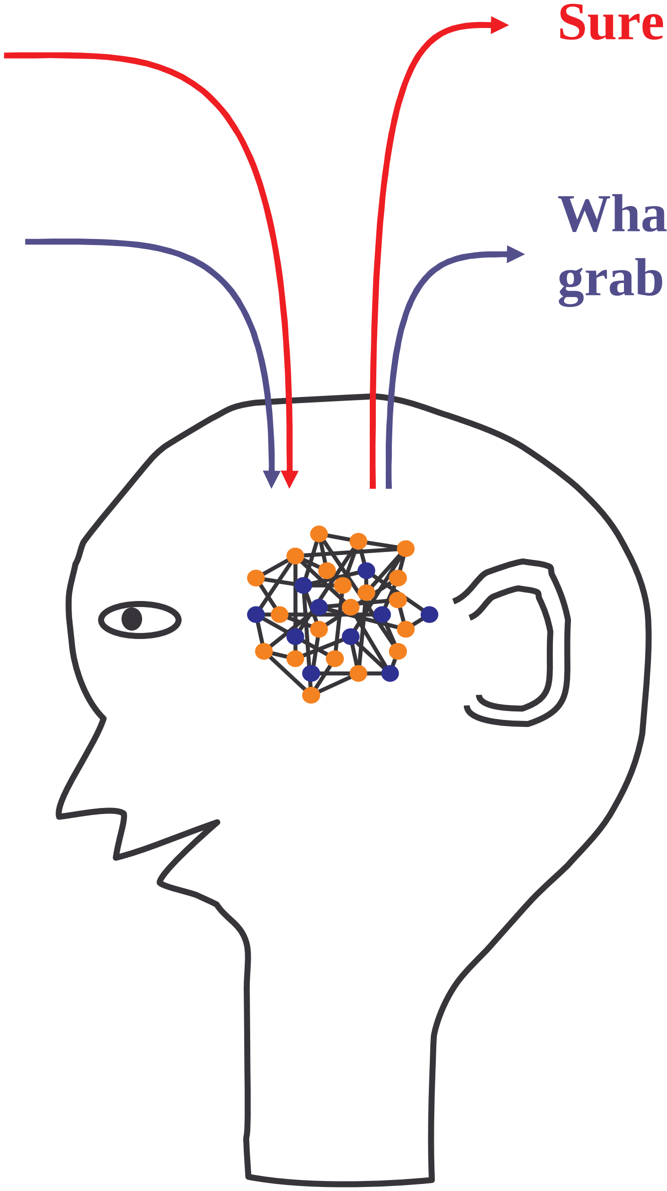
Typical exchanges in London



It’s raining outside;
want to go to the pub?

Sure; I’ll grab the umbrella.

It’s dry outside;
want to go to the pub?

What, are you insane? I’ll
grab the umbrella.

- The present state of networks depend on past input.

- For many task, “past” means 10s of seconds.

- Goal: understand how a single network can do this.

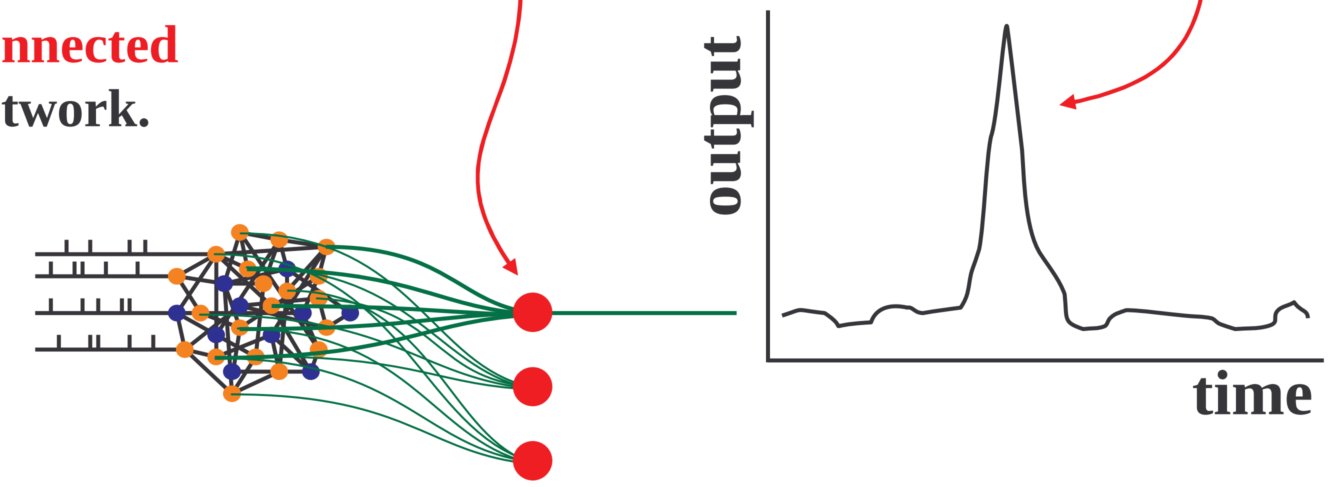
- Use an idea suggested by Jaeger (2001) and Maass et al. (2002).

The idea:

Time varying input drives a **randomly connected** recurrent network.

Output is a linear combination of activity; many linear combinations are possible in the same network.

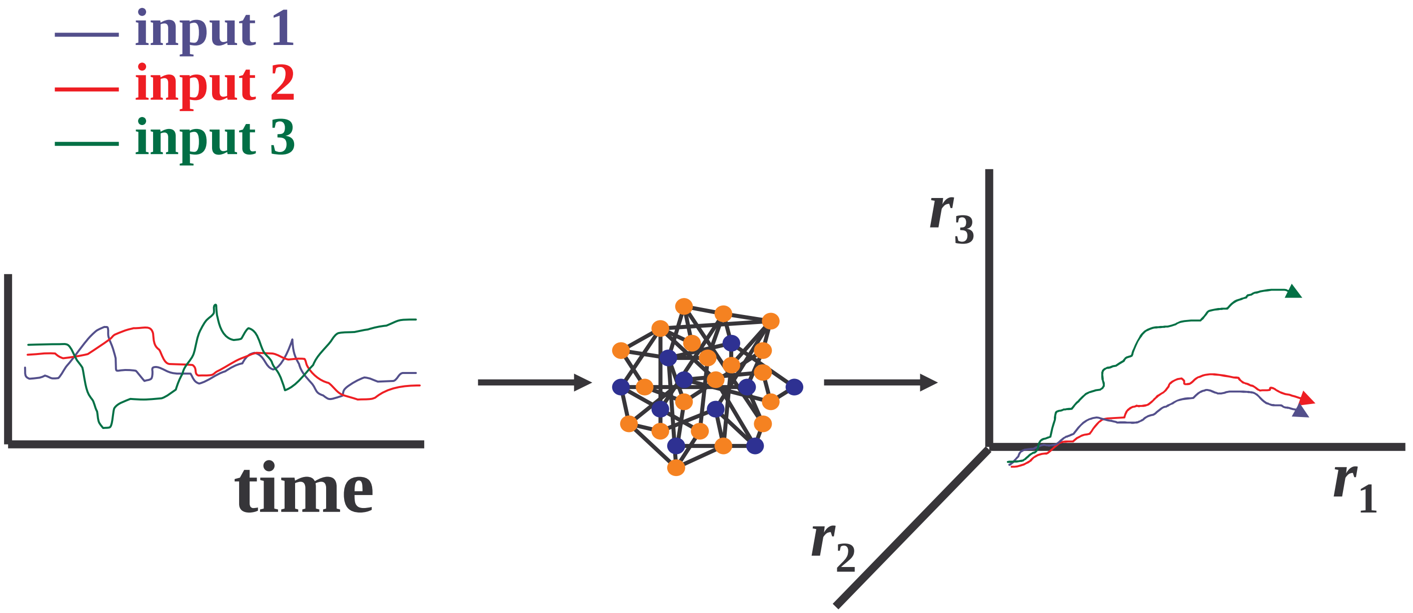
A particular input – and only that input – strongly activates an output unit.



Can **randomly connected** networks like this one do a good job **classifying input**?

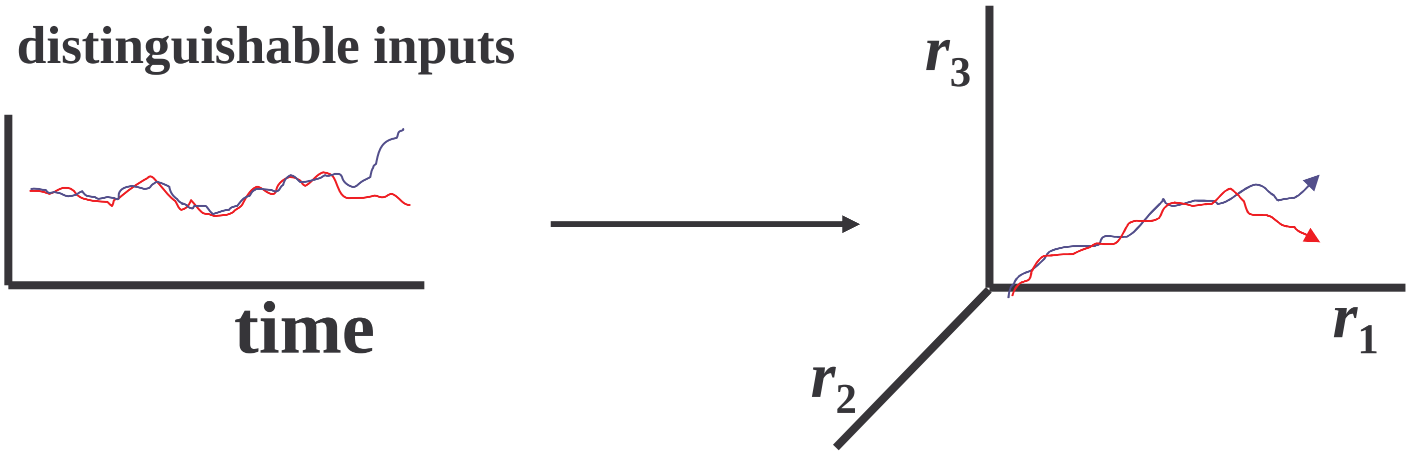
In other words: **can randomly connected networks tell that two different inputs really are different**?

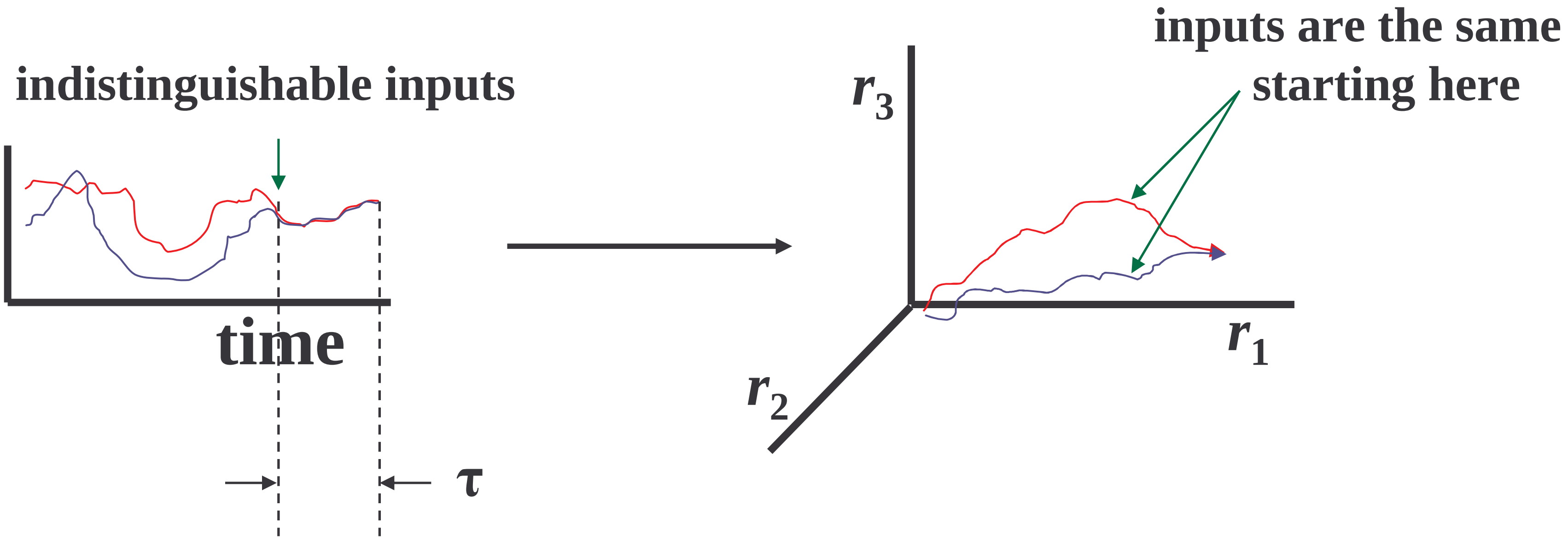
Answer can be visualized by looking at trajectories in activity space:



Activity space (N-dimensional)

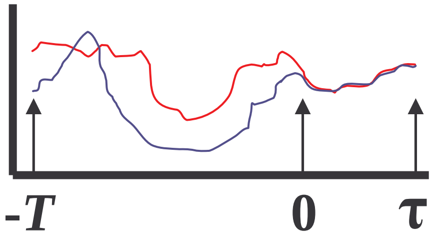
There is a subtlety involving time:



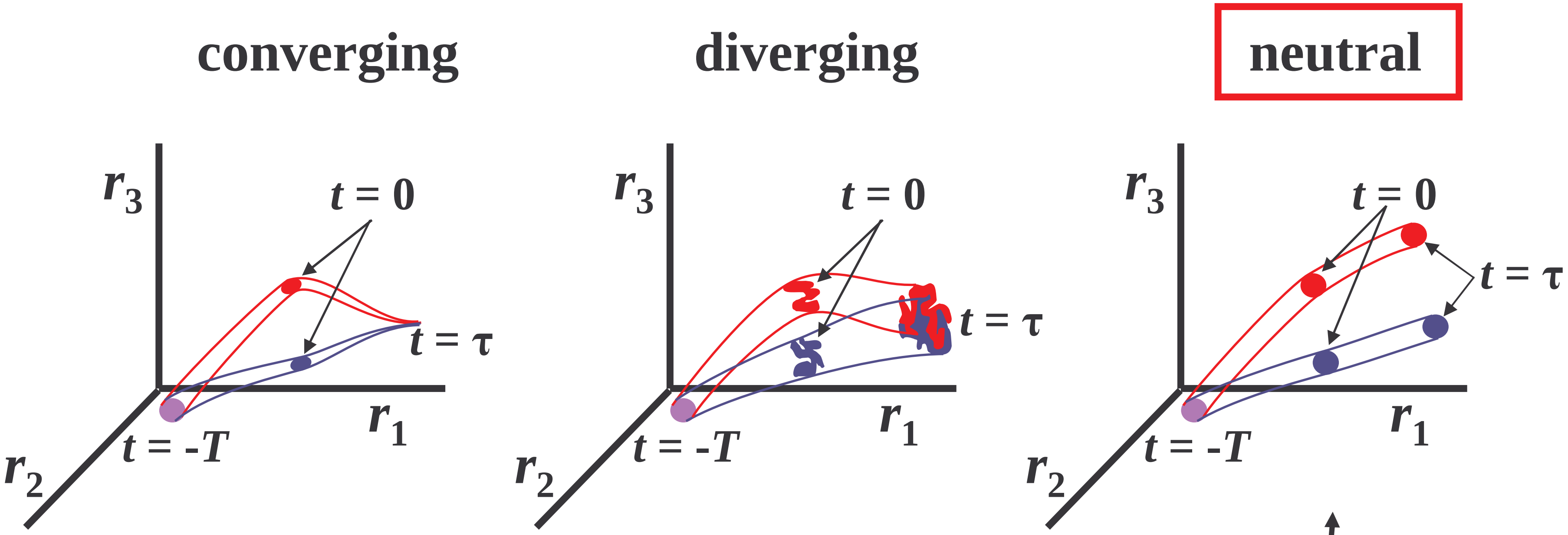


How big can we make τ before the inputs are indistinguishable?

input:



Three main regimes:


converging

diverging

neutral

Can we build a network that operates here?

Reduced model*:

temporally uncorrelated
input

$$x_i(t+1) = \mathrm{sign}[\textstyle\sum_j w_{ij}\, x_j(t) + u_i(t)]$$

random matrix
mean = 0
variance = σ^2/N
number of neurons = N

Question: what happens to nearby trajectories?

*Bertschinger and Natschläger (2004): low connectivity.
Our network: high connectivity.
Analysis is virtually identical.

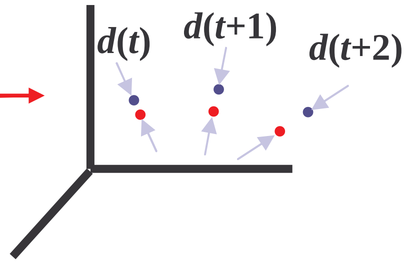
Analysis

Two trajectories:

$x_{1,i}(t)$ **and** $x_{2,i}(t)$ **(different initial conditions)**

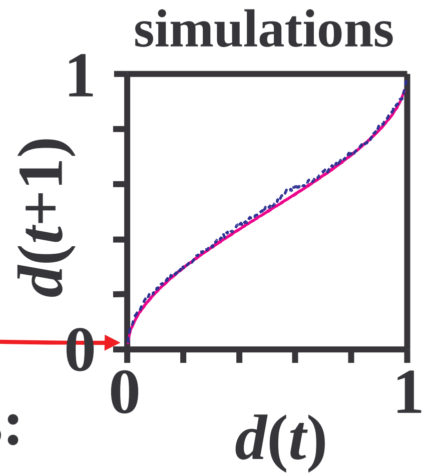
Normalized Hamming distance:

$$d(t) = (1/N) \sum_i |x_{1,i}(t)-x_{2,i}(t)|/2$$



How does $d(t)$ **evolve in time? For small** d**,**

$$d(t+1) \sim d(t)^{1/2}$$



This leads to very rapid growth of small separations:

$$d(t) \sim d(0)^{1/2^t} \quad \Rightarrow \quad d(t) \sim 1 \text{ when } t \sim \log \log [1/d(0)]$$

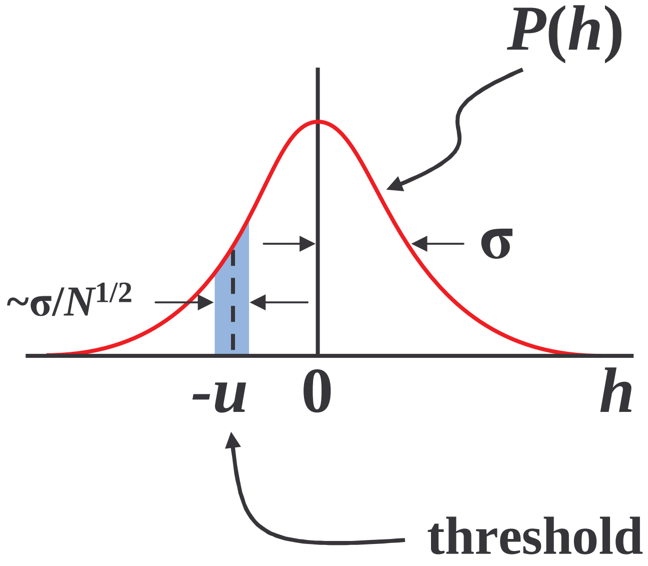
“Derivation”

$$x_i(t+1) = \text{sign}[h_i(t) + u]$$

$$\xrightarrow{} \sum_j w_{ij}\, x_j(t)$$

What happens if one neuron (neuron *k*) is different between the two trajectories?

$$x_{1k} = -x_{2,k}$$
$$h_{1,i} = h_{2,i} \pm 2w_{ik}$$
$$= h_{2,i} + \text{Order}(\sigma/N^{1/2})$$



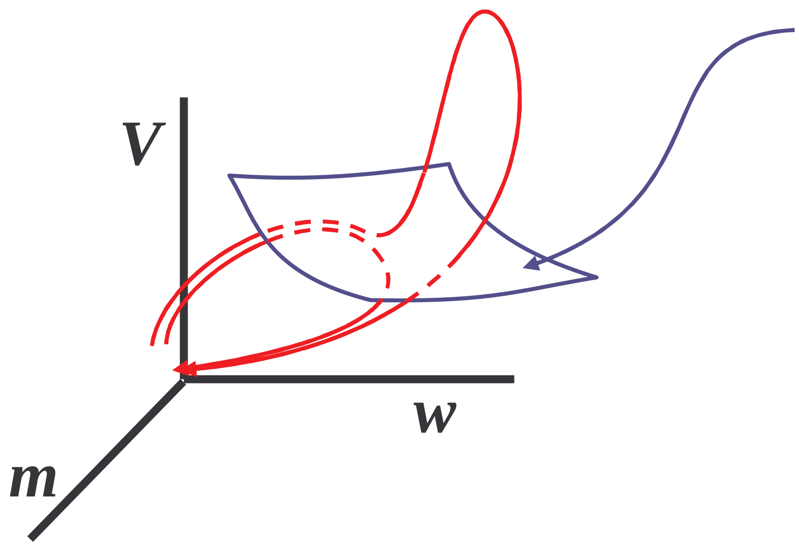
=> $N\,O(\sigma/N^{1/2})/\sigma = O(N^{1/2})$ neurons are different on the next time step.

In other words,

$d(0) = 1/N$

$d(1) \sim N^{1/2}/N = N^{-1/2} = d(0)^{1/2}$

Real neurons:



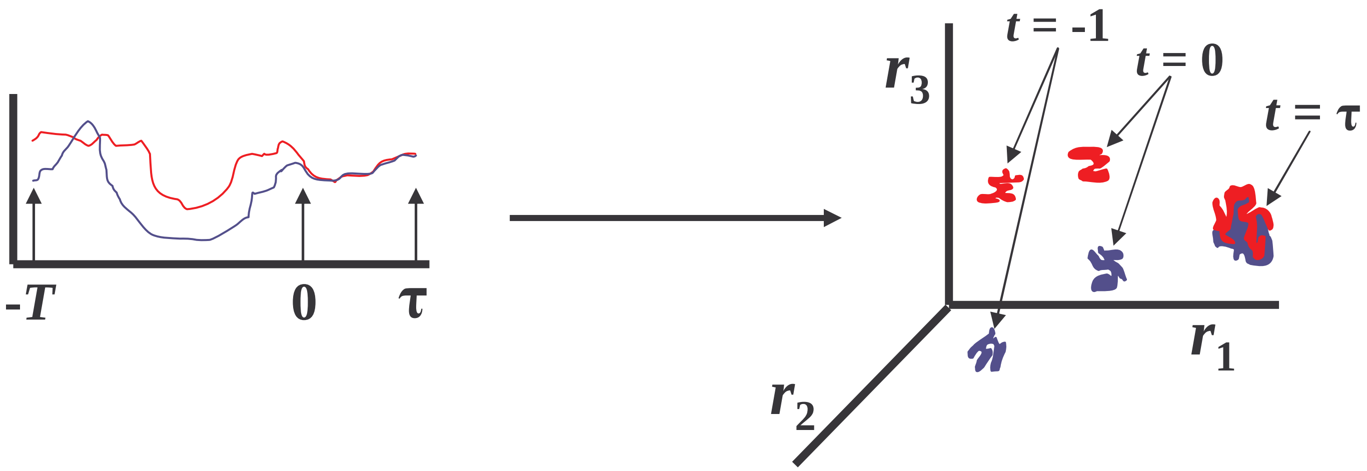
spike generation surface: small differences in initial conditions are strongly amplified (=> chaos).

van Vreeswijk and Sompolinsky (1996)
Banerjee (2001)

Operation in the neutral regime (on the edge of chaos) is not an option in realistic networks.

Implications

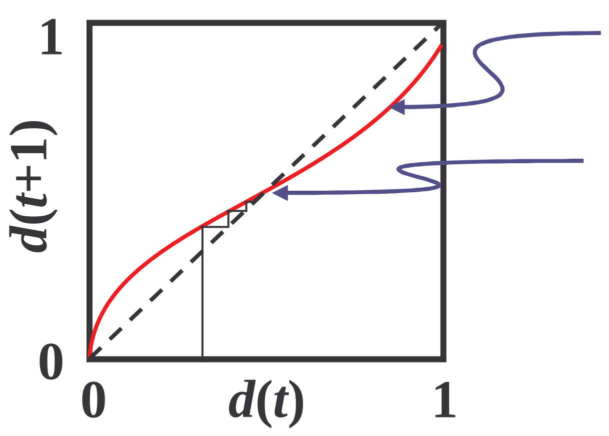
input:



- Trajectories evolve onto chaotic attractors (blobs).
- Different initial conditions will lead to different points on the attractor.

- What is the typical distance between points on an attractor?
- How does that compare the typical distance between attractors?

Typical distance between points on an attractor: *d**.



f(*d*)

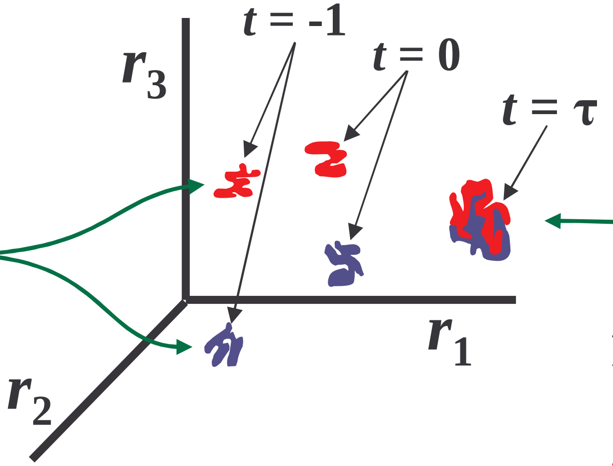
stable equilibrium, *d**

near attractor, $d(t+1)-d^* = f'(d^*)\,(d(t)-d^*)$

=> $d(t)-d^* \sim \exp[t \log(f'(d^*))]$

Typical distance between attractors: d_0 at time 0; *d** at long times.

Distance between attractors is $d_0 > d^*$

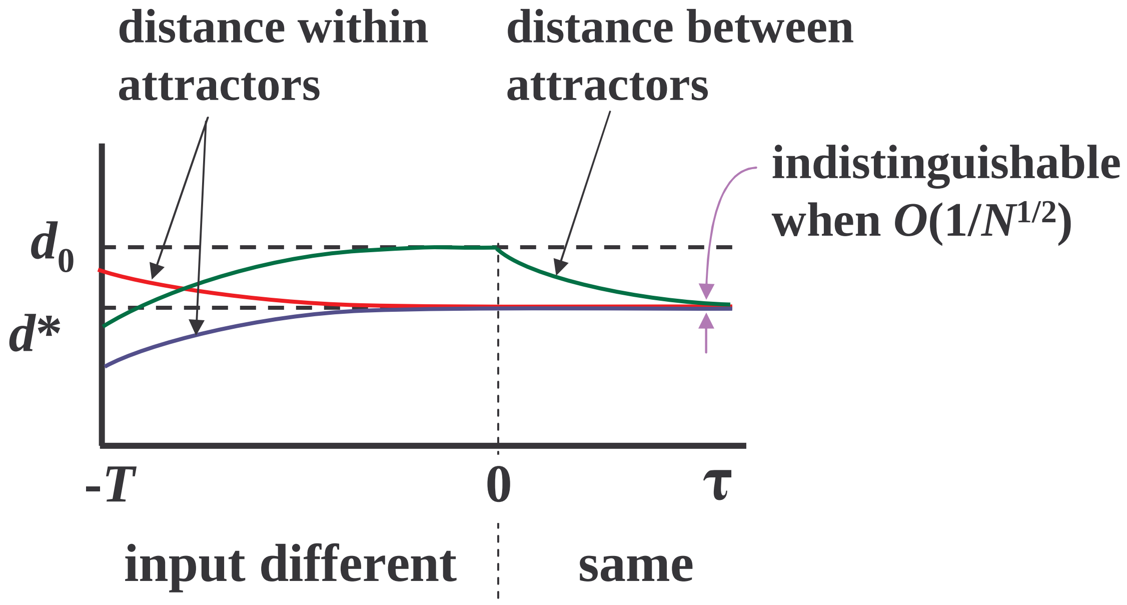


After a long time, the distance between attractors decays to *d**. At that point, inputs are no longer distinguishable (with caveat).

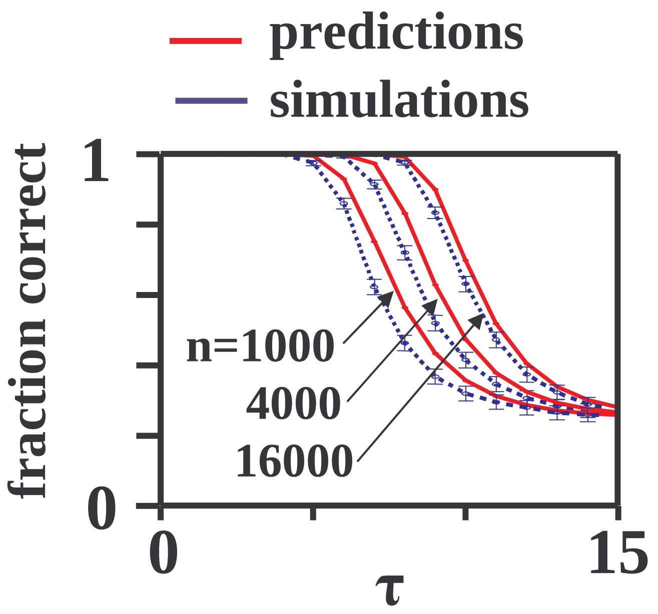
All points on the attractor are a distance $d^*+O(1/N^{1/2})$ apart.
Distance between attractors is $d^*+(d(0)-d^*)\exp[t \log(f'(d^*))]+O(1/N^{1/2})$.

State of the network no longer provides reliable information about the input when $\exp[\tau \log(f'(d^*))] \sim 1/N^{1/2}$, or:

$$\boxed{\tau \sim \frac{\log N}{-2 \log(f'(d^*))}}$$


distance within attractors
distance between attractors
indistinguishable when $O(1/N^{1/2})$
d_0
d^*
$-T$
0
τ
input different
same

Linear readout



Conclusions

1. Expanding on a very simple model proposed by Bertschinger and Natschläger (2004), we found that randomly connected networks cannot exhibit a temporal memory that extends much beyond the time constants of the individual neurons.

2. Scaling with the size of the network is not favorable: memory scales as $\log N$.

3. Our arguments were based on the observation that high connectivity, recurrent networks are chaotic (Banerjee, 2001), and so our conclusions should be very general.

Technical details

Mean field limit:

$$d(t) = \text{prob}\{\text{sign}[\textstyle\sum_j w_{ij}\, x_{1,j}(t) + u_i(t)] \neq \text{sign}[\textstyle\sum_j w_{ij}\, x_{2,j}(t) + u_i(t)]\}$$

Define:

$$h_{k,i} = \textstyle\sum_j w_{ij}\, x_{k,j}(t), \qquad k\text{=1, 2}$$

$h_{k,i}$ is a zero mean Gaussian random variable.

Covariance matrix:

$\sigma^2\delta_{jj'}$

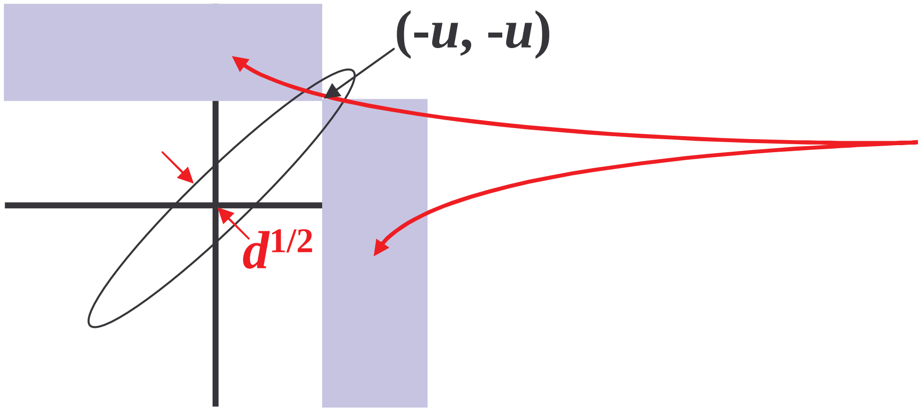
$$R_{kl} = \langle h_k\, h_l \rangle = (1/N)\textstyle\sum_i \sum_{jj'} w_{ij}\, x_{k,j}(t)\, w_{ij}'\, x_{l,j}'(t)$$

$$= (\sigma^2/N)\textstyle\sum_j x_{k,j}(t) x_{l,j}(t) = \sigma^2[1 - 2d(t)\,(1\text{-}\delta_{kl})]$$

More succinctly:

$$R = \sigma^2 \begin{pmatrix} 1 & 1-2d \\ 1-2d & 1 \end{pmatrix}$$

Can compute $d(t+1)$ as a function of $d(t)$ by doing Gaussian integrals:



integral is over
these regions

The $d^{1/2}$ scaling is generic; it comes from the fact that the Gaussian ellipse has width $d^{1/2}$ in the narrow direction.

This scaling also holds for more realistic reduced models with excitatory and inhibitory cells and synaptic and cellular time constants.

$$x_i(t+1) = \mathrm{sign}[\textstyle\sum_j w_{xx,ij}\, z_{xj}(t) - \sum_j w_{xy,ij}\, z_{yj}(t) + u_i(t)] + (1-\alpha)x_i(t)$$

$$y_i(t+1) = \mathrm{sign}[\textstyle\sum_j w_{yx,ij}\, z_{xj}(t) - \sum_j w_{yy,ij}\, z_{yj}(t) + u_i(t)] + (1-\beta)y_i(t)$$

leaky integrator

$$z_{xi}(t+1) = x_i(t) + (1-\kappa)\, z_{xi}(t)$$

$$z_{yi}(t+1) = x_i(t) + (1-\gamma)\, z_{yi}(t)$$

synapses with
temporal dynamics

References:

H. Jaeger, *German National Research Center for Information Technology, GMD Report 148* (2001).

W. Maass, T. Natschläger, and H. Markram, *Neural Computation* 14:2531-2560 (2004).

Bertschinger and Natschläger, *Neural Computation* 16:1413-1436 (2004).

C. van Vreeswijk and H. Sompolinsky, *Science* 274:1724-1726 (1996).

A. Banerjee, Neural Computation 13:161-193, 195-225 (2001).