1

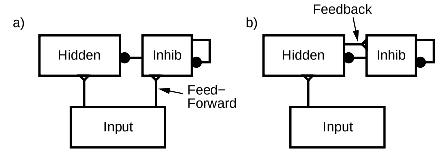
Networks: Inhibition

- Controls activity (bidirectional excitation).
- Inhibition = thermostat-controlled air conditioner
- inhibitory neurons sample excitatory activity (like a thermostat samples the temperature)
- more excitatory activity → more inhibition to keep the network from getting too "hot" (active) → **set point** behavior

1. Biology: Feedforward and Feedback.
2. Critical Parameters.
3. kWTA Simplification.

2

Types of Inhibition



Anticipates excitation

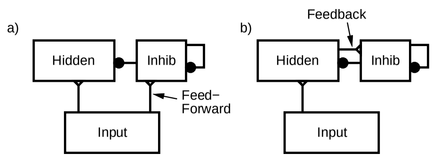
Like having thermostat outside of your house

Reacts to excitation

Like a normal (indoor) AC thermostat

3

Critical Parameters (inhib.proj)



- Inhib conductance into hidden units (`g_bar_i.hidden`)
- Inhib conductance into inhib units (`g_bar_i.inhib`)
- Strength of feedforward weights to inhib (`scale.ff`)
- Strength of feedback weights to inhib (`scale.fb`)

4

Simulations: [amp_top_down_dist.proj; inhib.proj]

5

k-Winners-Take-All (kWTA)

- The function of inhibition is to keep excitatory activity at a rough **set point**.
- We can approximate this function by enforcing a max activity level in each layer.
- kWTA: Instead of simulating inhibitory neurons, we choose an inhibitory current g_i value for each layer such that the specified number **k** of excitatory neurons are above threshold.
- *Advantages:* Much less computationally expensive, avoids oscillations.

6

How kWTA is Actually Computed

- Step 1: For each unit in a layer, compute the value of g_i needed to counteract excitation and put the unit's membrane potential at threshold

$$V_m = \frac{g_e \bar{g}_e E_e + g_i \bar{g}_i E_i + g_l \bar{g}_l E_l}{g_e \bar{g}_e + g_i \bar{g}_i + g_l \bar{g}_l}$$

$$\Theta = \frac{g_e \bar{g}_e E_e + g_i \bar{g}_i E_i + g_l \bar{g}_l E_l}{g_e \bar{g}_e + g_i \bar{g}_i + g_l \bar{g}_l}$$

set V_m to the threshold value and solve for g_i

$$g_i^{\Theta} = \frac{g_e^* \bar{g}_e (E_e - \Theta) + g_l \bar{g}_l (E_l - \Theta)}{\Theta - E_i}$$

7

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

8

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

standard kWTA:

If k = 2, set inhibition for the layer such that the two units receiving the most excitation are above threshold, but others are not.

e.g., g_i = 6.04

9

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

standard kWTA:

If k = 2, set inhibition for the layer such that the two units receiving the most excitation are above threshold, but others are not.

e.g., g_i = 6.04

10

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

Note that kWTA still allows **some** “wiggle room” in how much activity there is within a layer...

11

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

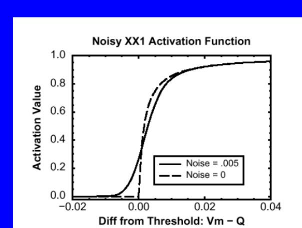
Note that kWTA still allows **some** “wiggle room” in how much activity there is within a layer...

1. If leak is high enough, fewer than k units will be active
2. The distribution of activity values is important

12

How kWTA is Actually Computed

- Recall that activation is a function of **how far** above threshold the unit is...



13

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.99	7.43
.99	7.43
.01	0.08
.01	0.08
.01	0.08
.01	0.08
.01	0.08
.01	0.08
.01	0.08
.01	0.08

In this case, if inhibition is set midway between the second and third values, inhibition = 3.76

The first and second units will be **far above** the inhibitory threshold, so they will be **strongly** active

14

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.99	7.43
.99	7.43
.98	7.35
.98	7.35
.98	7.35
.98	7.35
.98	7.35
.98	7.35
.98	7.35
.98	7.35

In this case, if inhibition is set midway between the second and third values, inhibition = 7.39

The first and second units will be **very close** to the inhibitory threshold, so they will **not** be strongly active

15

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

Note that kWTA still allows **some** "wiggle room" in how much activity there is within a layer...

1. If leak is high enough, fewer than k units will be active
2. The distribution of activity values is important
3. **A variant of kWTA called average-based kWTA gives even more wiggle room**

16

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

6.64

average-based kWTA:

1. Compute the average of the g_i values needed to put the top k (2) guys at threshold

17

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

6.64

2.74

average-based kWTA:

1. Compute the average of the g_i values needed to put the top k (2) guys at threshold
2. Compute the average of the g_i values for the other guys

18

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

6.64

2.74

average-based kWTA:

1. Compute the average of the g_i values needed to put the top k (2) guys at threshold => 6.638
2. Compute the average of the g_i values for the other guys => 2.738
3. Set inhibition somewhere between these two values

19

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

6.64

2.74

average-based kWTA:

Depending on where exactly you place the threshold, and the distribution of unit activity values, you may get fewer than 2 or more than 2 units active

20

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

6.64
6.60

2.74

average-based kWTA:

Depending on where exactly you place the threshold, and the distribution of unit activity values, you may get fewer than 2 or more than 2 units active

21

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

6.64

4.69

2.74

average-based kWTA:

Depending on where exactly you place the threshold, and the distribution of unit activity values, you may get fewer than 2 or more than 2 units active

22

How kWTA is Actually Computed

g_e	g_i needed to put V_m at threshold
.90	6.75
.87	6.53
.74	5.55
.63	4.73
.50	3.75
.49	3.68
.32	2.50
.12	0.90
.08	0.60
.04	0.03

6.64

3.00
2.74

average-based kWTA:

Depending on where exactly you place the threshold, and the distribution of unit activity values, you may get fewer than 2 or more than 2 units active

23

kWTA: Summary

- Simple shortcut we use instead of actual inhibitory interneurons
- Captures basic idea that inhibition maintains activity at a **set point** for a given layer
- Specify inhibition value for a layer such that *k* units are active
- *k* is a parameter: percent activity levels vary across different brain regions!
- kWTA still allows for some wiggle room in overall activation

24

Benefits of Inhibition

- Controls activity (bidirectional excitation)
- Inhibition forces units to *compete* to represent the input: Only the most appropriate (best-fitting) units survive the competition

25 Simulations: [inhib.proj]

1. kWTA approximates set point behavior.
2. Allows for faster updating.

26 Networks

1. Biology: The cortex
2. Excitation:
   - Unidirectional (transformations)
   - Bidirectional (top-down processing, pattern completion, amplification)
3. Inhibition: Controls bidirectional excitation (feedforward, feedback, set point, kWTA approximation)
4. Constraint Satisfaction: Putting it all together.

27 Constraint Satisfaction

Process of trying to satisfy various constraints (from environment, connection weights, activations).

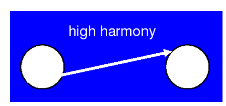
Bidirectional excitation and inhibition form part of this larger computational goal.

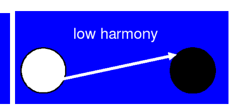
1. Energy/harmony.
2. Attractor Dynamics.
3. Noise.

28 Harmony

Harmony = extent to which unit activations are consistent with weights

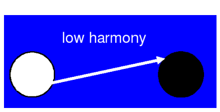
$H = \frac{1}{2} \sum_j \sum_i a_i w_{ij} a_j$ *Harmony is high when units with strong (positive) weights are co-active*





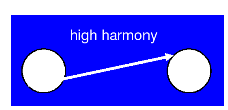
29 Harmony

John Hopfield showed that harmony tends to increase monotonically as the network settles



30 Harmony

John Hopfield showed that harmony tends to increase monotonically as the network settles

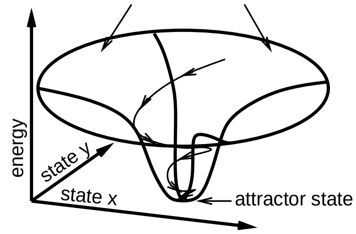


network settling = moving to a more "harmonious" state

31

Attractor Dynamics

A network will *settle* into a *stable state* over time: the *attractor*.



Maximize harmony given inputs and weights.

32

Circle Attractor

Attractor = stable set of states from many starting points.

33

[catsanddogs.proj]

Knowledge is in the weights! (built in here)

Harmony as a function of constraints

34

Constraint Satisfaction

- Bidir networks like "cats and dogs" constitute a **content-addressable memory**: you can access any part of a memory from any other
- "Who likes to eat grass and play with feathers?"
- In contrast, simple uni-directional nets are more like an address book: you can look up the address from someone's last name, but can't get names from addresses.

35

Constraint Satisfaction

- **Spontaneous generalization**: The network can extract regularities from its base of knowledge
- "What are dogs like?"
- "What are medium dogs like?"

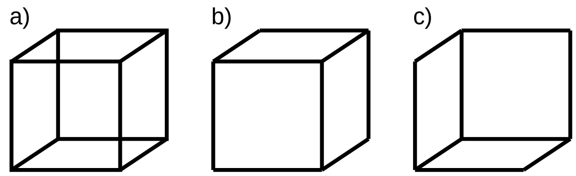
36

Constraint Satisfaction

1. Attractor Dynamics.
2. Energy/harmony
3. Noise.

37

The Necker Cube

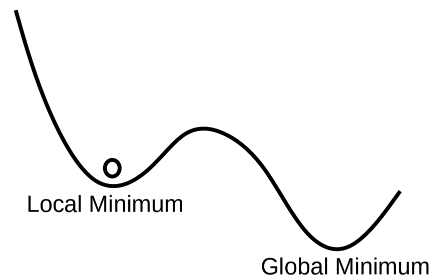


- Two different interpretations
- Can't perceive both at once
- Alternate between perceptions: *bistability*

38

The Role of Noise

How might noise be useful in your brain?



(local maximum for harmony)

39

[necker_cube.proj]

Role of noise

Accommodation

40

Networks

1. Biology: The cortex
2. Excitation:
   - Unidirectional (transformations)
   - Bidirectional (top-down processing, pattern completion, amplification)
3. Inhibition: Controls bidirectional excitation (feedforward, feedback, set point, kWTA approximation)
4. Constraint Satisfaction: Putting it all together.