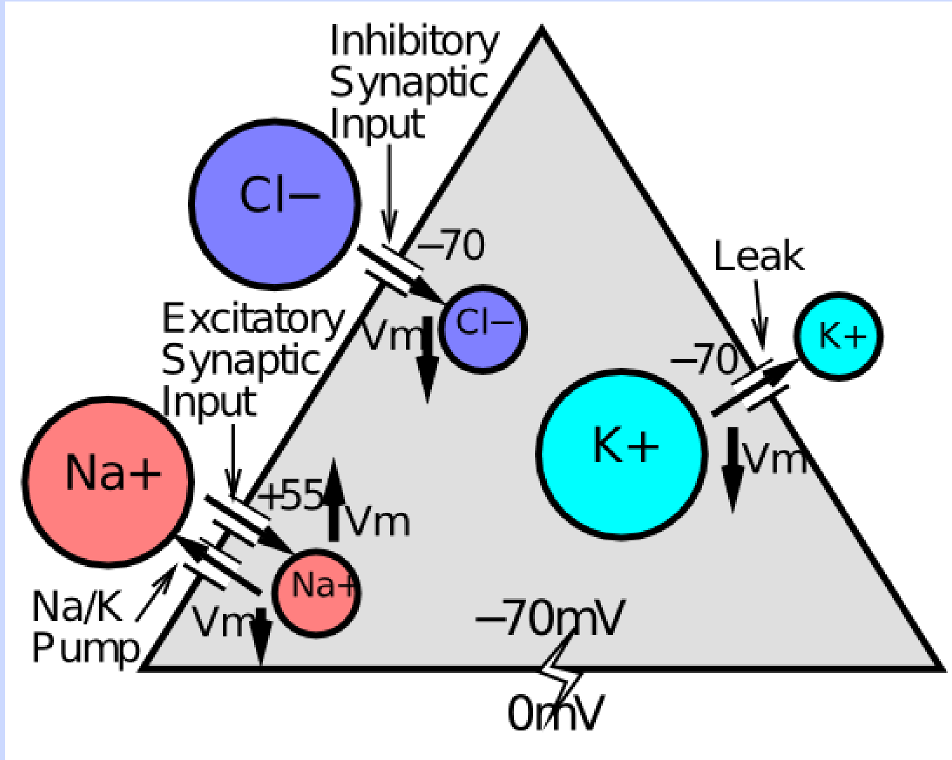
The perils of bidirection excitation..

Excitatory vs Inhibitory Neurons

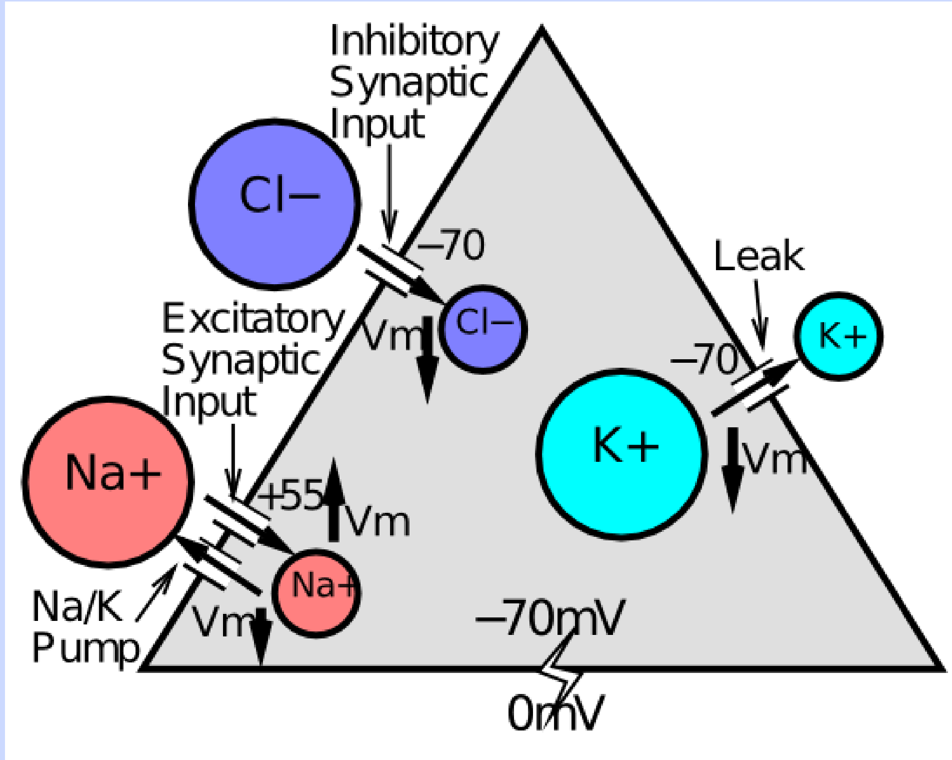
- Excitatory neurons both project locally and make long-range projections between different cortical areas
- Inhibitory neurons primarily project within small, localized regions of cortex
- Excitatory neurons carry the information flow (long range projections)
- Inhibitory neurons are responsible for (locally) regulating the activation of excitatory neurons

The Neuron and its Ions



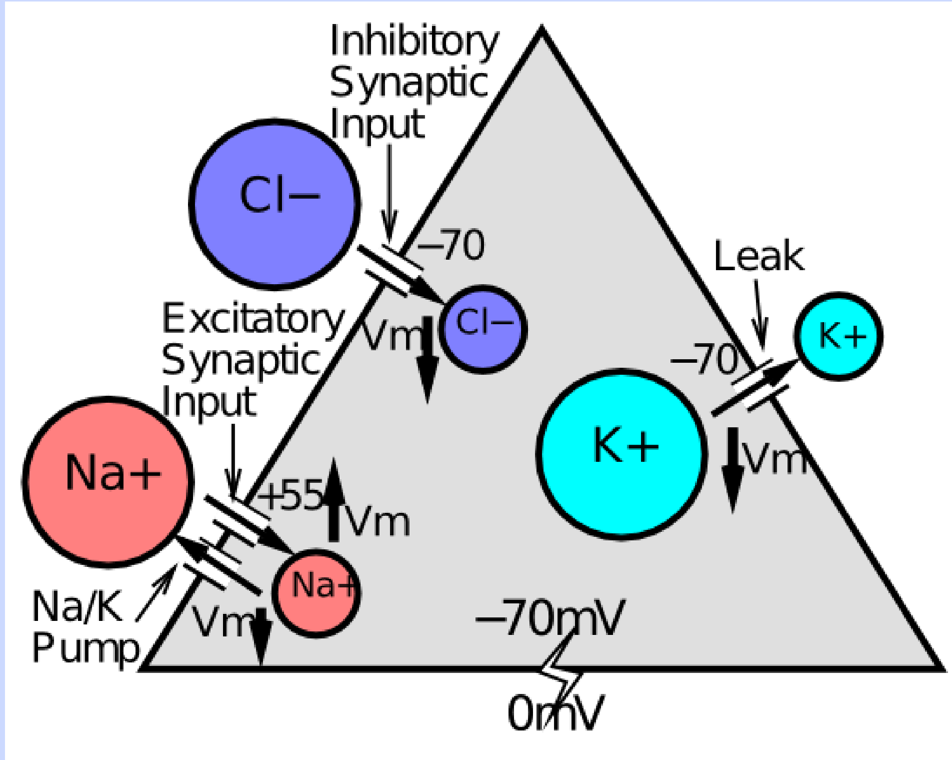
Glutamate →

The Neuron and its Ions



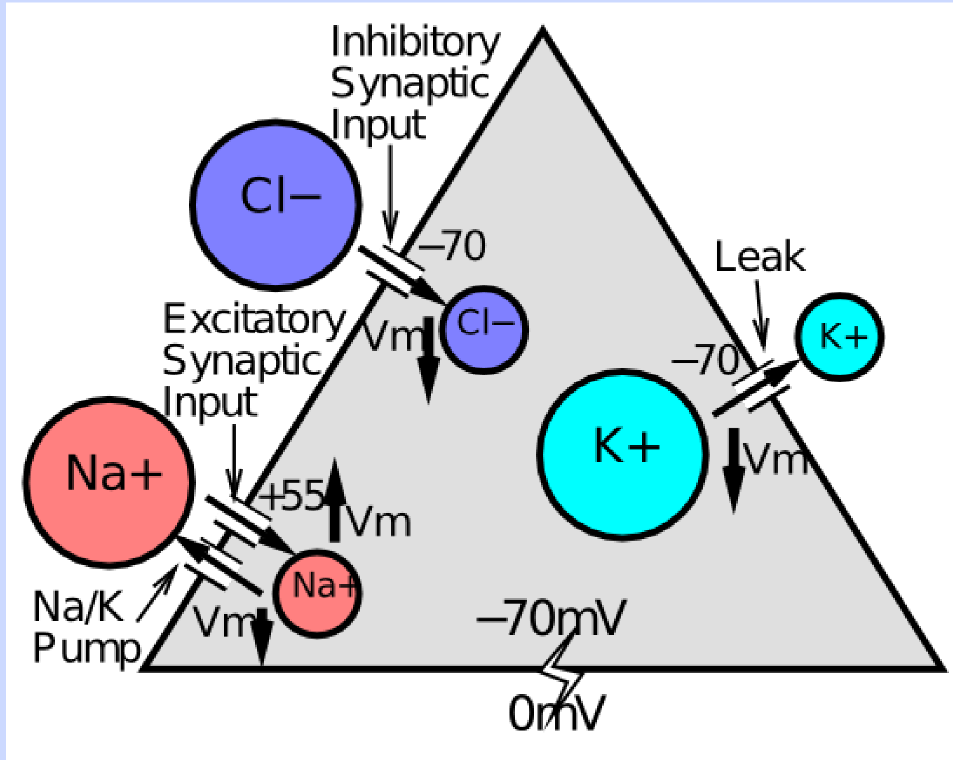
Glutamate → opens Na+ channels →

The Neuron and its Ions



Glutamate → opens Na+ channels → Na+ enters (excitatory)

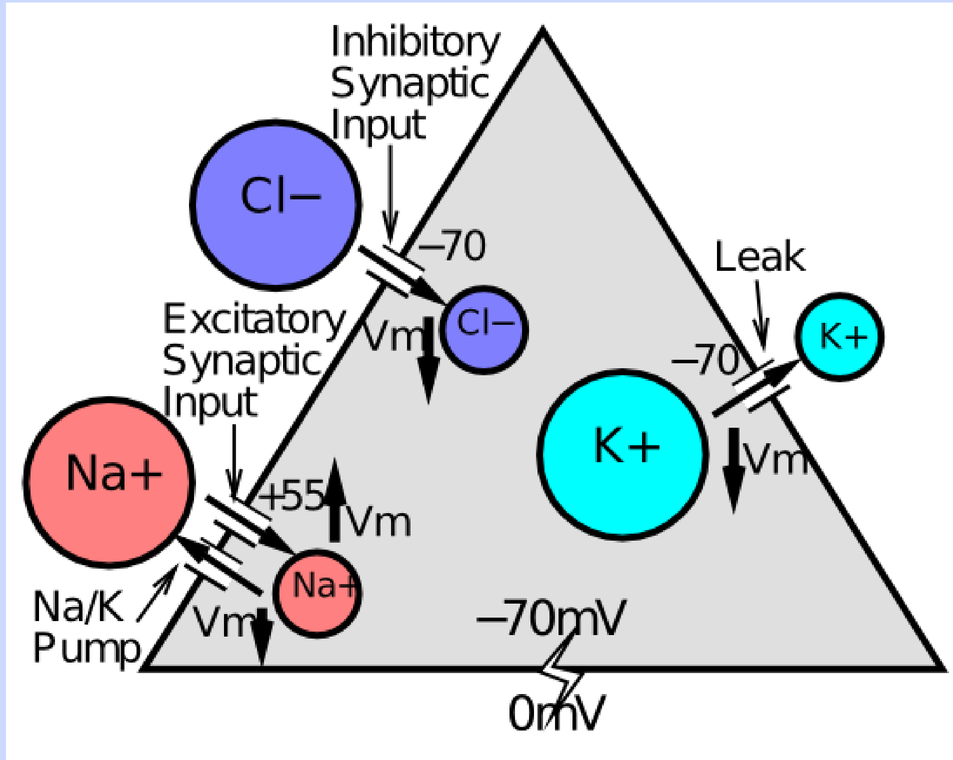
The Neuron and its Ions



Glutamate → opens Na+ channels → Na+ enters (excitatory)

GABA →

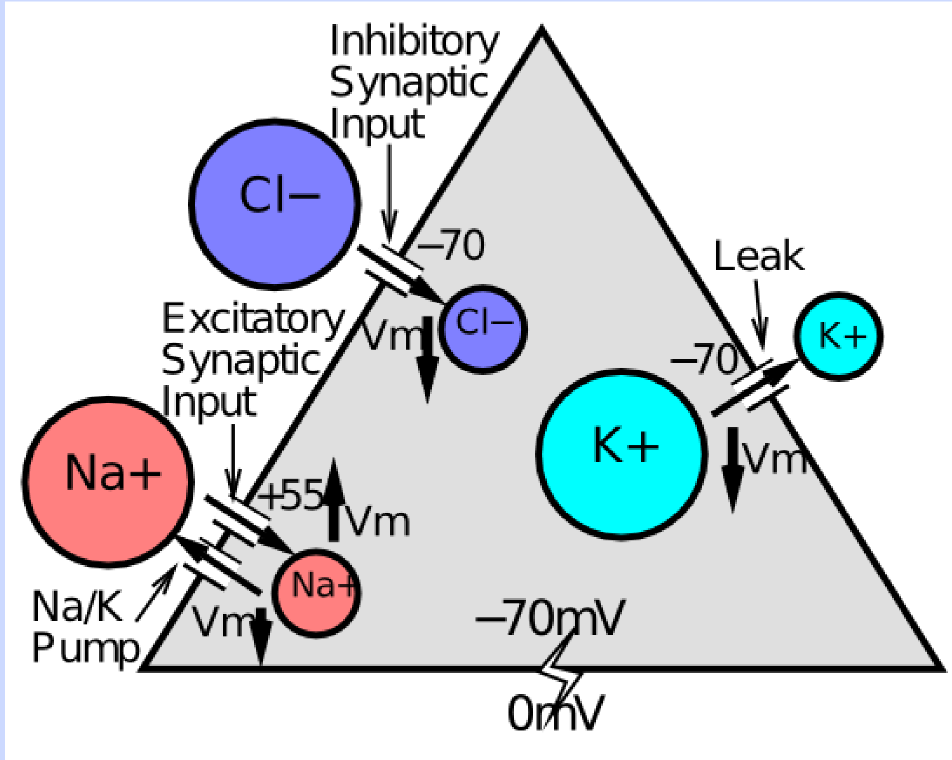
The Neuron and its Ions



Glutamate $\rightarrow$ opens Na+ channels $\rightarrow$ Na+ enters (excitatory)

GABA $\rightarrow$ opens Cl- channels $\rightarrow$

The Neuron and its Ions



Glutamate → opens Na+ channels → Na+ enters (excitatory)

GABA → opens Cl- channels → Cl- enters if $V_m \uparrow$ (inhibitory)

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 $\rightarrow$ more inhibition to keep the network from getting too “hot” (active) $\rightarrow$ **set point** behavior

Networks: Inhibition

- Controls activity (bidirectional excitation).

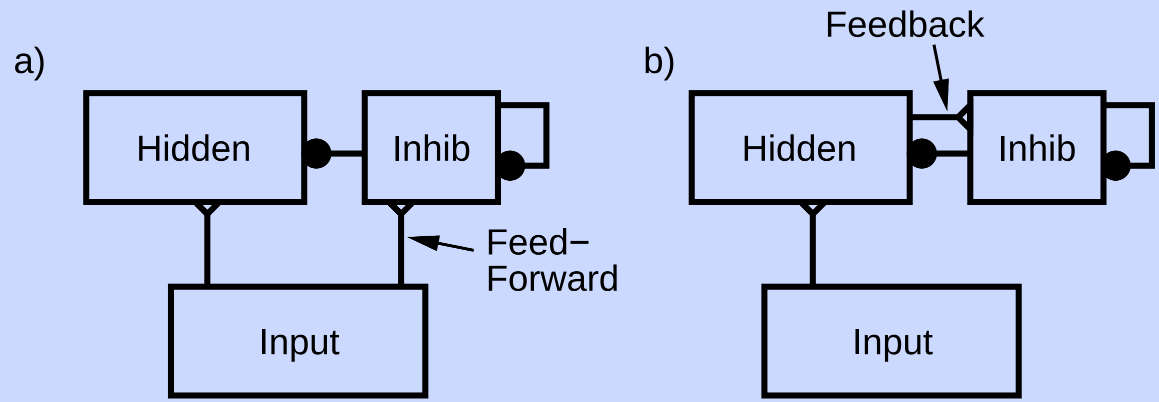
- Inhibition = thermostat-controlled air conditioner

- inhibitory neurons sample excitatory activity (like a thermostat samples the temperature)

- more excitatory activity $\rightarrow$ more inhibition to keep the network from getting too "hot" (active) $\rightarrow$ **set point** behavior

1. Biology: Feedforward and Feedback.
2. Critical Parameters.
3. Simplification (FFFB function).

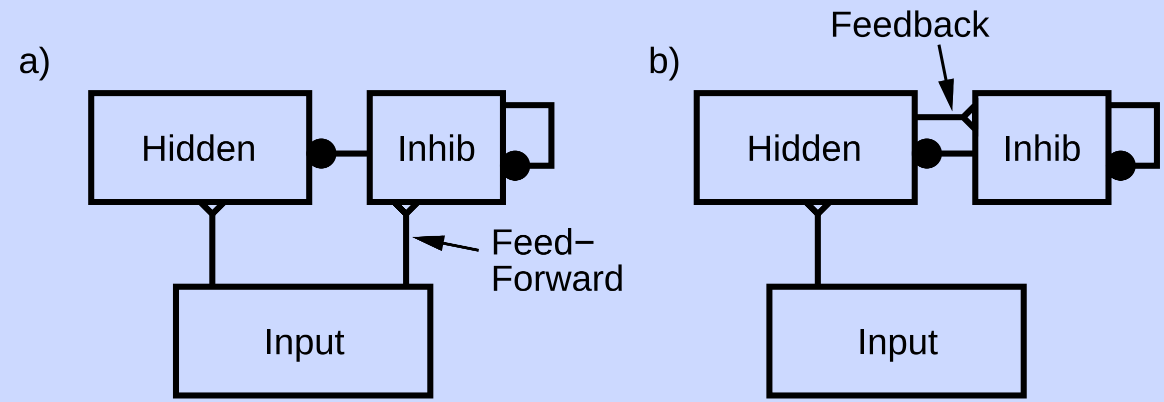
Types of Inhibition



Anticipates excitation

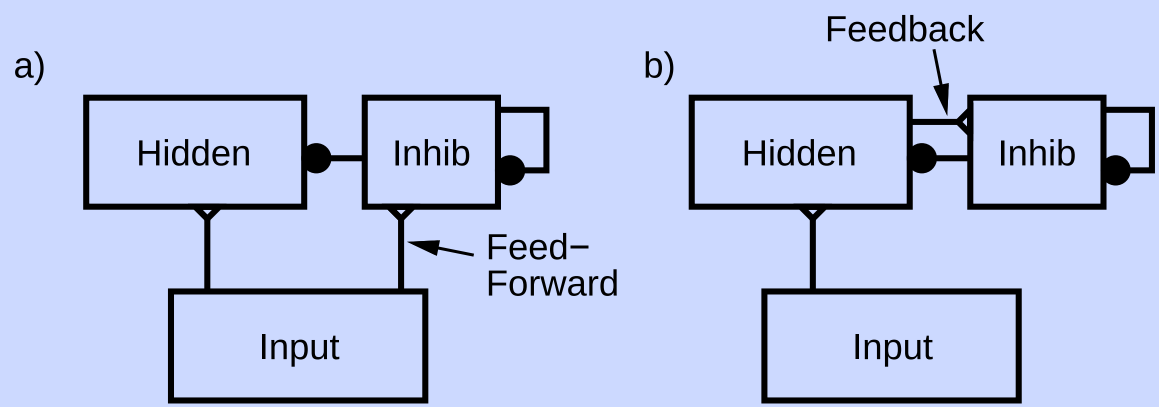
Reacts to excitation

Types of Inhibition



Anticipates excitation	*Reacts to excitation*
Like having thermostat outside of your house	Like a normal (indoor) AC thermostat

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 (`ff_wt_scale`)
- Strength of feedback weights to inhib (`fb_wt_scale`)

Simulations: [inhib.proj]

FFFB inhibition function

FFFB inhibition function

- We can approximate feedforward (FF) and feedback (FB) aspects of inhibitory interneurons using the FFFB inhibition function:
- average net input: $< \eta > = \sum_n \frac{1}{n} \eta_i$
- average activation: $< y > = \sum_n \frac{1}{n} y_i$
- Then: $ff(t) = ff[< \eta > - ff0]_+$
- $fb(t) = fb(t-1) + dt[fb < y > - fb(t-1)]$
- Now just set g_i in target layer as a function of FF and FB: $g_i(t) = g_i[ff(t) + fb(t)]$

FFFB inhibition function

- We can approximate feedforward (FF) and feedback (FB) aspects of inhibitory interneurons using the FFFB inhibition function:
- average net input: $< \eta >= \sum_n \frac{1}{n} \eta_i$
- average activation: $< y >= \sum_n \frac{1}{n} y_i$
- Then: $ff(t) = ff[< \eta > - ff0]_+$
- $fb(t) = fb(t-1) + dt[fb < y > - fb(t-1)]$
- Now just set g_i in target layer as a function of FF and FB:
  $g_i(t) = g_i[ff(t) + fb(t)]$
- *Advantages:* Much less computationally expensive, avoids oscillations.

Simulations: [inhib_fffb.proj]

1. FFFB approximates set point behavior.

2. Allows for faster updating, reduces overall computation.

3. Can use in large networks with multiple layers, with inhibition summarized by FFFB

4. Can still capture differential amounts of inhibition in different brain areas with FFFB params: g_i, FF and FB components

5. in some applications may still want actual inhib neurons

Alternative inhibition function (optional)
: k-Winners-Take-All (kWTA)

Alternative inhibition function (optional) : 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.

Alternative inhibition function (optional) : 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.

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

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

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

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., $g_i = 6.04$

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., $g_i = 6.04$

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...

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

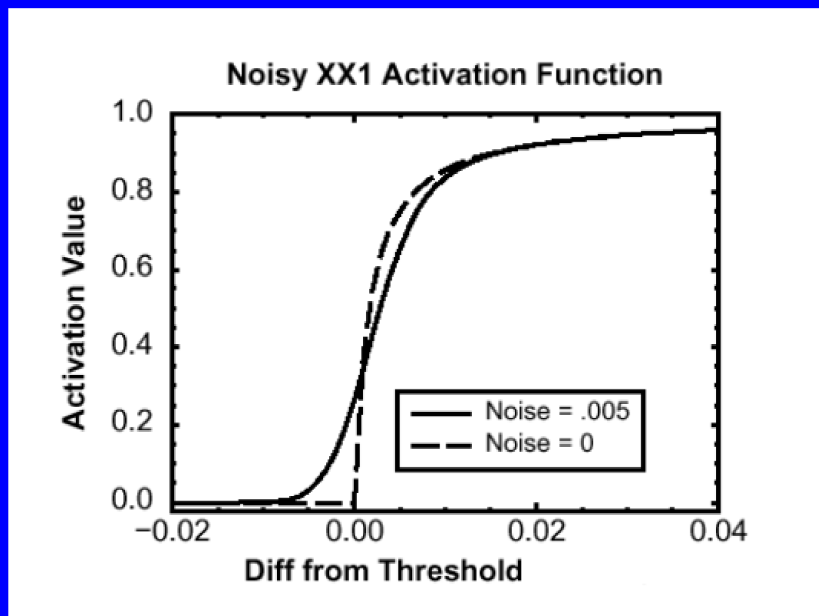
Note that kWTA still allows **some** “wiggle room” in how much activity there is within a layer…

1. If leak is high enough, fewer than k units will be active

2. The distribution of activity values is important

How kWTA is Actually Computed

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



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

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

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**

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

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

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

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

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

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

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

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

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

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, FFFB approximation)

4. Constraint Satisfaction: Putting it all together.

Constraint Satisfaction

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.

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.

Harmony

Harmony = extent to which unit activations are consistent with weights

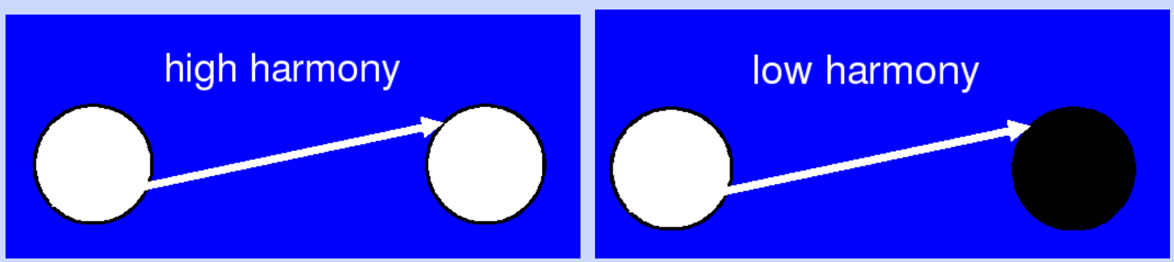
$$H = \frac{1}{2} \sum_j \sum_i a_i w_{ij} a_j$$

Harmony

Harmony = extent to which unit activations are consistent with weights

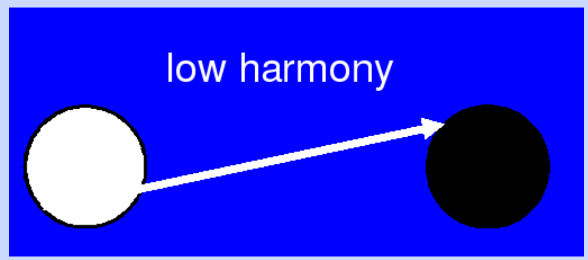
$$H = \frac{1}{2}\sum_j \sum_i a_i w_{ij} a_j$$

Harmony is high when units with strong (positive) weights are co-active



Harmony

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



Harmony

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

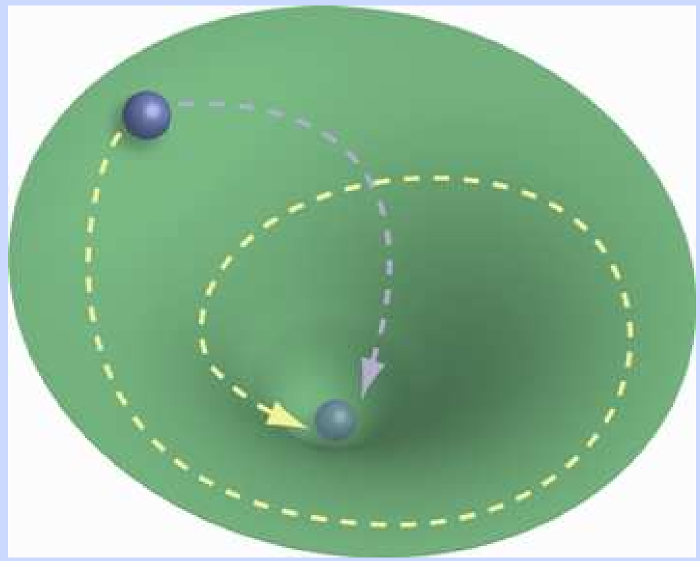
Harmony

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

network settling = moving to a more "harmonious" state

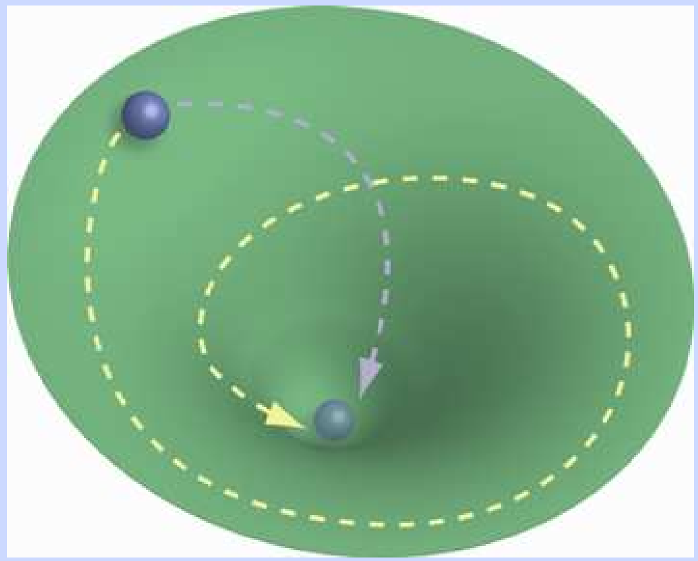
Attractors

An attractor network is a network of neurons with excitatory interconnections that can settle into a stable pattern of firing given a range of different starting states.



Attractors

An attractor network is a network of neurons with excitatory interconnections that can settle into a stable pattern of firing given a range of different starting states.



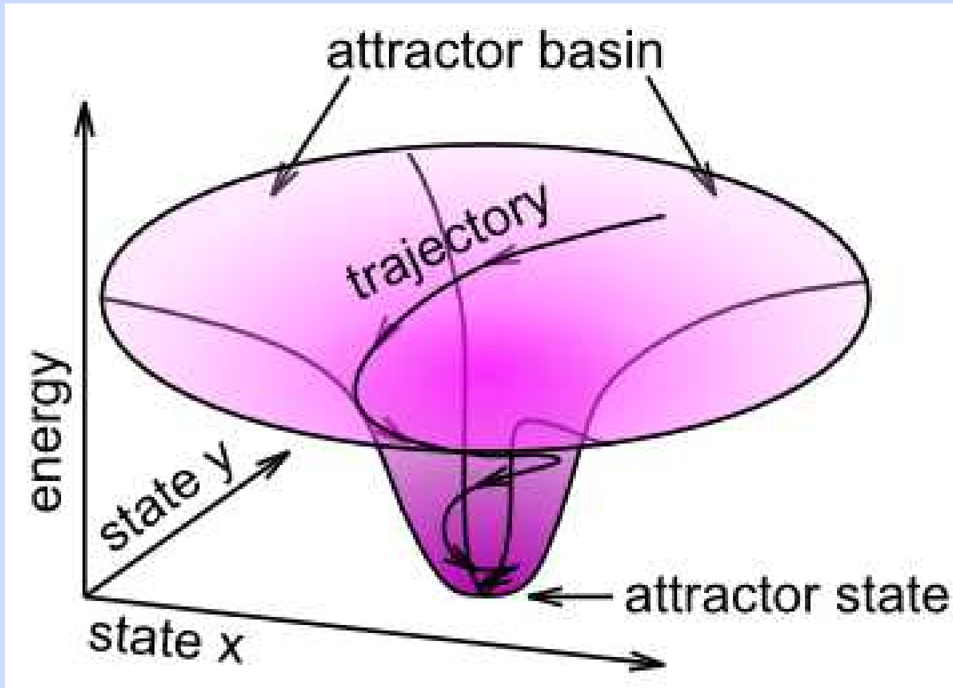
[here we consider only *fixed point* attractors, but cyclical or chaotic attractors are also possible]

Attractor Dynamics

Bidirectional excitation causes a network to *settle* into a particular *stable state* over time: the *attractor*.

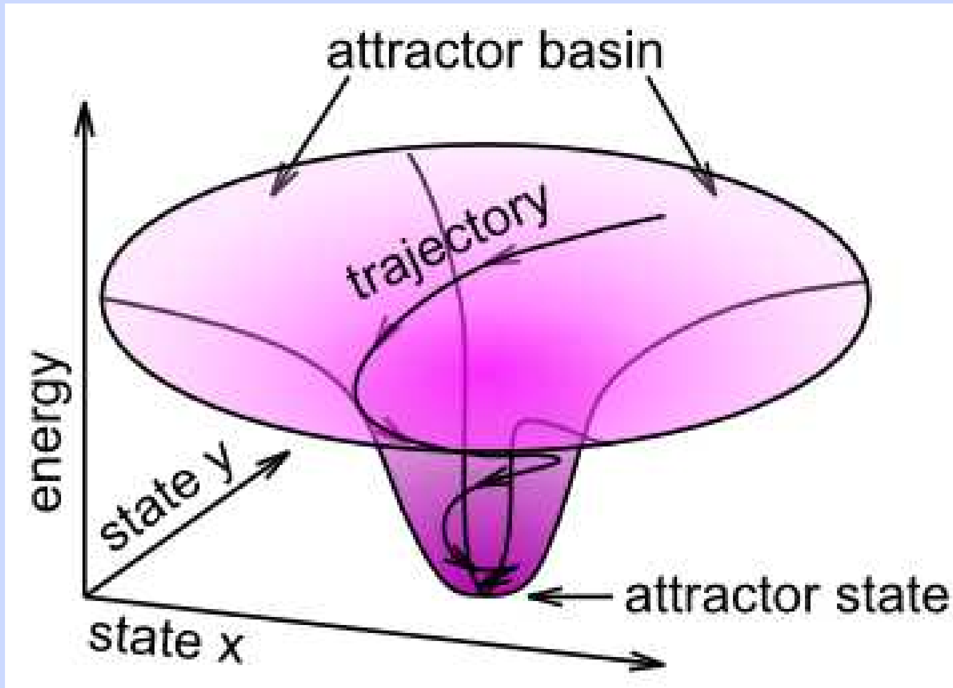
Attractor Dynamics

Bidirectional excitation causes a network to *settle* into a particular *stable state* over time: the *attractor*.



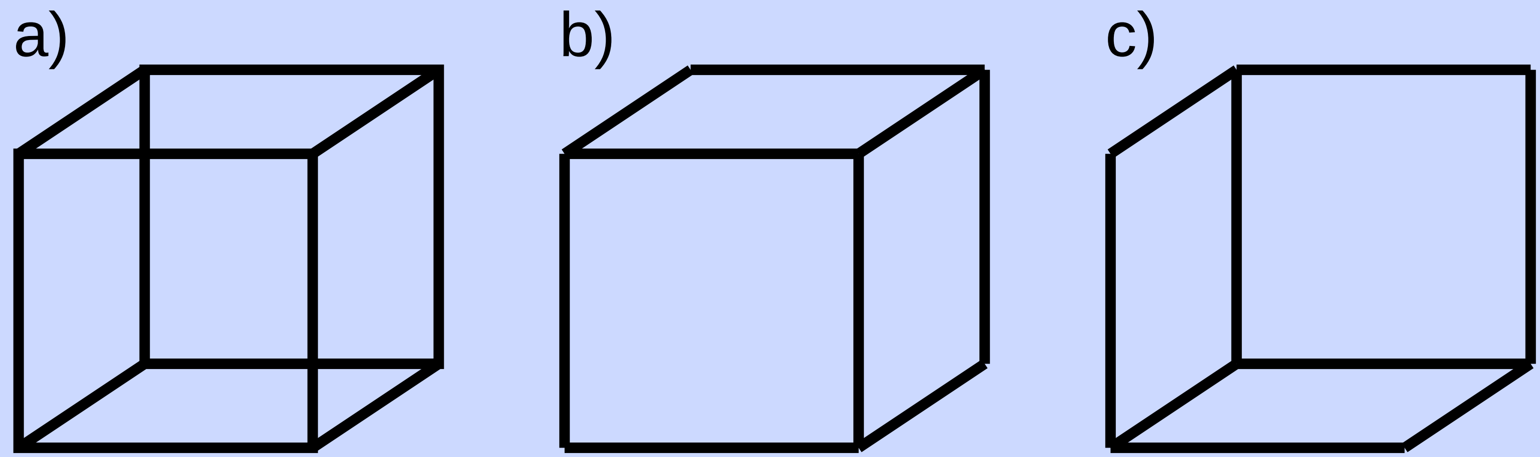
Attractor Dynamics

Bidirectional excitation causes a network to *settle* into a particular *stable state* over time: the *attractor*.



Maximize harmony given inputs and weights.

The Necker Cube



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

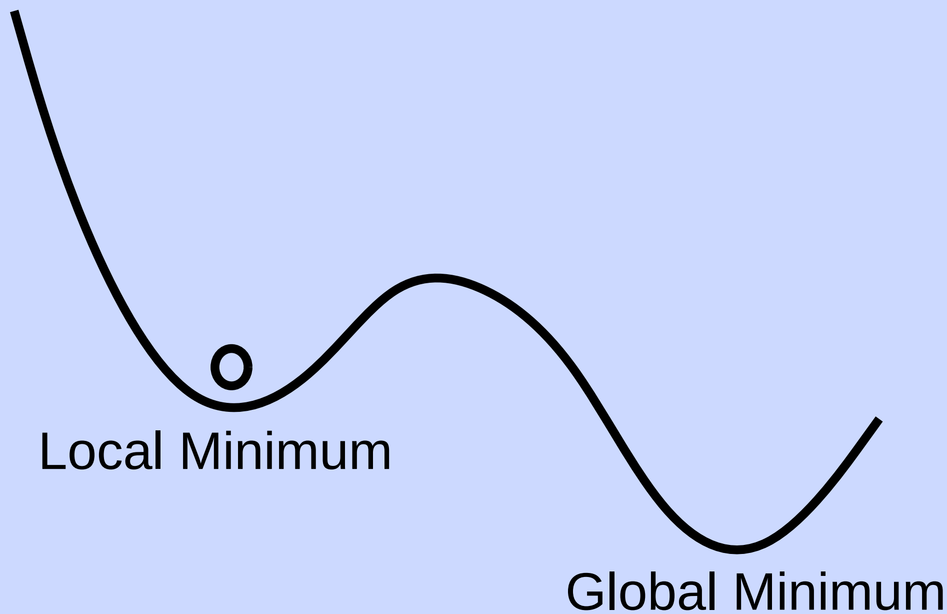
Other example

The Role of Noise

How might noise be useful in your brain?

The Role of Noise

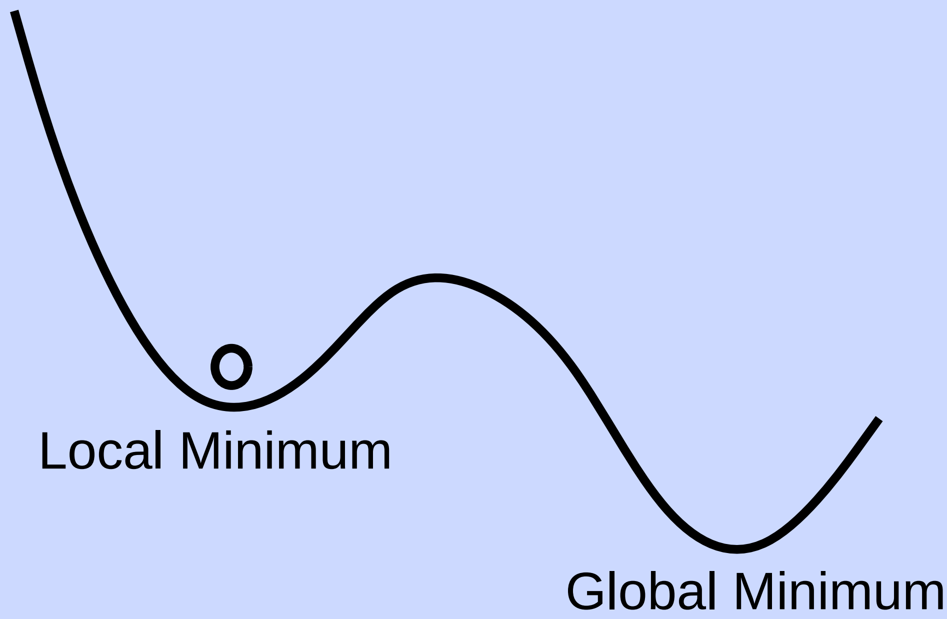
How might noise be useful in your brain?



(local maximum for harmony)

The Role of Noise

How might noise be useful in your brain?



(local maximum for harmony)

Example: skiing...

[necker_cube.proj]

Role of noise

Accommodation

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.

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.