Machine Learning & Neuroscience: Performance vs. Biological Relevance in Neural Networks

Corbin Diaz (Northwestern University)

October 29, 2025

Table of Contents

Table of Contents

Neurons

- Neurons receive multiple inputs from dendrites causing it to spike in voltage.

Neurons

- Neurons receive multiple inputs from dendrites causing it to spike in voltage.
- This spike is sent down the axon to release neurotransmitters in the synapse.

Neurons

- Neurons receive multiple inputs from dendrites causing it to spike in voltage.
- This spike is sent down the axon to release neurotransmitters in the synapse.
- These neurotransmitters either excite or inhibit the next neuron, affecting if it spikes in voltage.

The Action Potential

- Spikes in voltage are regulated by ion concentrations.

The Action Potential

- Spikes in voltage are regulated by ion concentrations.
- Usually, there is more sodium (Na+) ions outside the cell and more potassium (K+) ions inside the cell.

The Action Potential

- Spikes in voltage are regulated by ion concentrations.
- Usually, there is more sodium (Na+) ions outside the cell and more potassium (K+) ions inside the cell.
- These baseline concentrations establish the resting membrane potential.

The Action Potential

- Spikes in voltage are regulated by ion concentrations.
- Usually, there is more sodium (Na+) ions outside the cell and more potassium (K+) ions inside the cell.
- These baseline concentrations establish the resting membrane potential.
- When voltage increases pass threshold, voltage-gated Na+ channels release Na+ inside the cell, increasing voltage.

The Action Potential

- Spikes in voltage are regulated by ion concentrations.
- Usually, there is more sodium (Na+) ions outside the cell and more potassium (K+) ions inside the cell.
- These baseline concentrations establish the resting membrane potential.
- When voltage increases pass threshold, voltage-gated Na+ channels release Na+ inside the cell, increasing voltage.
- After some delay, voltage-gated K+ channels release K+ outside the cell, decreasing voltage.

The Action Potential

- Spikes in voltage are regulated by ion concentrations.
- Usually, there is more sodium (Na+) ions outside the cell and more potassium (K+) ions inside the cell.
- These baseline concentrations establish the resting membrane potential.
- When voltage increases pass threshold, voltage-gated Na+ channels release Na+ inside the cell, increasing voltage.
- After some delay, voltage-gated K+ channels release K+ outside the cell, decreasing voltage.
- This pattern is known as the action potential.

Modeling the Action Potential

Leaky Integrate-and-Fire Model

$$\tau \frac{dV(t)}{dt} = -\Delta V(t) + R \cdot I(t)$$

$$\Delta V(t) = V(t) - E \qquad \text{(the driving force)}$$

where $V(t)$ is the membrane voltage, $I(t)$ is the input current, E is the resting membrane potential, $\tau > 0$ is the time constant, and R is the membrane resistance.

The Main Equation

Many equations we will see have this form:

$$\tau \frac{dy}{dt} = -y + f(t), \;\; y(0) = 0$$

This is solved via integrating factor:

The Main Equation

Many equations we will see have this form:

$$\tau \frac{dy}{dt} = -y + f(t), \;\; y(0) = 0$$

This is solved via integrating factor:

$$y' + \frac{1}{\tau} y = \frac{1}{\tau} f(t)$$

The Main Equation

Many equations we will see have this form:

$$\tau \frac{dy}{dt} = -y + f(t), \;\; y(0) = 0$$

This is solved via integrating factor:

$$y' + \frac{1}{\tau} y = \frac{1}{\tau} f(t)$$
$$e^{t/\tau} y' + \frac{1}{\tau} e^{t/\tau} y = \frac{1}{\tau} f(t) e^{t/\tau}$$

The Main Equation

Many equations we will see have this form:

$$\tau \frac{dy}{dt} = -y + f(t), \;\; y(0) = 0$$

This is solved via integrating factor:

$$\begin{aligned} y' + \frac{1}{\tau} y &= \frac{1}{\tau} f(t) \\ e^{t/\tau} y' + \frac{1}{\tau} e^{t/\tau} y &= \frac{1}{\tau} f(t) e^{t/\tau} \\ (e^{t/\tau} y)' &= \frac{1}{\tau} f(t) e^{t/\tau} \end{aligned}$$

The Main Equation

Many equations we will see have this form:

$$\tau \frac{dy}{dt} = -y + f(t), \;\; y(0) = 0$$

This is solved via integrating factor:

$$\begin{aligned} y' + \frac{1}{\tau} y &= \frac{1}{\tau} f(t) \\ e^{t/\tau} y' + \frac{1}{\tau} e^{t/\tau} y &= \frac{1}{\tau} f(t) e^{t/\tau} \\ (e^{t/\tau} y)' &= \frac{1}{\tau} f(t) e^{t/\tau} \\ \int_0^{t_0} (e^{t/\tau} y)' \, dt &= \frac{1}{\tau} \int_0^{t_0} f(t) e^{t/\tau} \, dt \end{aligned}$$

The Main Equation

Many equations we will see have this form:

$$\tau \frac{dy}{dt} = -y + f(t), \;\; y(0) = 0$$

This is solved via integrating factor:

$$\begin{aligned}
y' + \frac{1}{\tau} y &= \frac{1}{\tau} f(t) \\
e^{t/\tau} y' + \frac{1}{\tau} e^{t/\tau} y &= \frac{1}{\tau} f(t) e^{t/\tau} \\
(e^{t/\tau} y)' &= \frac{1}{\tau} f(t) e^{t/\tau} \\
\int_0^{t_0} (e^{t/\tau} y)' \, dt &= \frac{1}{\tau} \int_0^{t_0} f(t) e^{t/\tau} \, dt \\
(e^{t_0/\tau}) y(t_0) &= \frac{1}{\tau} \int_0^{t_0} f(t) e^{t/\tau} \, dt
\end{aligned}$$

The Main Equation

$$\boxed{y(t_0) = \frac{1}{\tau} \int_0^{t_0} f(t) e^{-(t_0 - t)/\tau} \, dt}$$

The Main Equation

$$\boxed{y(t_0) = \frac{1}{\tau}\int_0^{t_0} f(t)e^{-(t_0-t)/\tau}\,dt}$$

- This equation models exponential approach towards $f(t)$.
  - For example, setting $f(t) = C$ gives

$$y(t_0) = C(1 - e^{-t_0/\tau})$$

The Main Equation

$$\boxed{y(t_0) = \frac{1}{\tau}\int_0^{t_0} f(t)e^{-(t_0-t)/\tau}\,dt}$$

- This equation models exponential approach towards $f(t)$.
  - For example, setting $f(t) = C$ gives

$$y(t_0) = C(1 - e^{-t_0/\tau})$$

- We can also write this another way:

$$y(t_0) = (K * f)(t_0)$$

  where K is the exponential kernel.

Modeling the Action Potential

- In 1952, Alan Hodgkin and Andrew Huxley used new voltage-clamp technology to study the action potential on the giant squid axon.

Modeling the Action Potential

- In 1952, Alan Hodgkin and Andrew Huxley used new voltage-clamp technology to study the action potential on the giant squid axon.
- They received the 1963 Nobel Prize in Physiology or Medicine.

Modeling the Action Potential

- In 1952, Alan Hodgkin and Andrew Huxley used new voltage-clamp technology to study the action potential on the giant squid axon.
- They received the 1963 Nobel Prize in Physiology or Medicine.

The Hodgkin-Huxley Model

$$C\frac{dV}{dt} = I - G_L(V - E) - \overline{G_{Na}}m^3h(V - E_{Na}) - \overline{G_K}n^4(V - E_K)$$

(m, n, h are each voltage and time dependent and are governed by their own set of equations)

Modeling voltages gets complicated fast!

Table of Contents

The Spike Train

- Information is encoded in firing rates (x), not voltage values.

The Spike Train

- Information is encoded in firing rates (x), not voltage values.
- The spike train function $s(t)$ is such that

$$s(t) = \begin{cases} 1 & \text{a spike occurred at time } t \\ 0 & \text{otherwise} \end{cases}$$

The Spike Train

- Information is encoded in firing rates (x), not voltage values.
- The spike train function $s(t)$ is such that

$$s(t) = \begin{cases} 1 & \text{a spike occurred at time } t \\ 0 & \text{otherwise} \end{cases}$$

- If we model current as exponentially decaying with each spike, we can express it as the convolution of the exponential kernel with the spike train.

The Firing Rate Equation

- This gives that current can be modeled as

$$\tau \frac{dI(t)}{dt} = -I(t) + \sum_i u_i s_i(t)$$

where u_i are the synaptic weights on the input neurons.

The Firing Rate Equation

- This gives that current can be modeled as

$$\tau \frac{dI(t)}{dt} = -I(t) + \sum_i u_i s_i(t)$$

  where u_i are the synaptic weights on the input neurons.
- If we imagine just looking at the average on a small enough time interval so that I and I' are roughly constant, we get

$$\tau \frac{dI(t)}{dt} = -I(t) + \sum_i u_i x_i(t)$$

  where x_i gives the firing rate.

The Firing Rate Equation

- We can relate the firing rate of the neuron through an $f \cdot I$-curve so that r exponentially approaches some curve $f(I)$. This gives

$$\tau_s \frac{dI(t)}{dt} = -I(t) + \sum_i u_i x_i(t)$$

$$\tau_r \frac{dr}{dt} = -r + f(I)$$

The Firing Rate Equation

- We can relate the firing rate of the neuron through an $f \cdot I$-curve so that r exponentially approaches some curve $f(I)$. This gives

$$\tau_s \frac{dI(t)}{dt} = -I(t) + \sum_i u_i x_i(t)$$

$$\tau_r \frac{dr}{dt} = -r + f(I)$$

- If we assume the firing rate changes much slower than the current changes ($\tau_s \approx 0$), we can simplify into one equation and get

$$\boxed{\tau \frac{dr}{dt} = -r + f\left(\sum_i u_i x_i(t)\right)}$$

Table of Contents

The Feedforward Neural Network

- Writing the firing rate equation in matrix form gives

$$\tau^T \frac{d\mathbf{r}}{dt} = -\mathbf{r} + f(U\mathbf{x})$$

where U is the synaptic weight matrix and $\mathbf{x}$ is the firing rate of the input neurons.

The Feedforward Neural Network

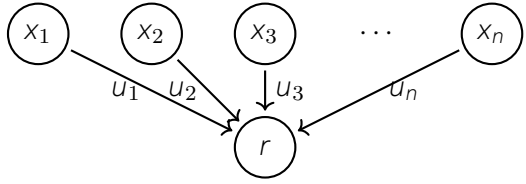
- Writing the firing rate equation in matrix form gives

$$\tau^T \frac{d\mathbf{r}}{dt} = -\mathbf{r} + f(U\mathbf{x})$$

  where U is the synaptic weight matrix and $\mathbf{x}$ is the firing rate of the input neurons.
- The steady-state of the equation is given by $\mathbf{r} = f(U\mathbf{x})$. This is a feedforward neural network with one layer.

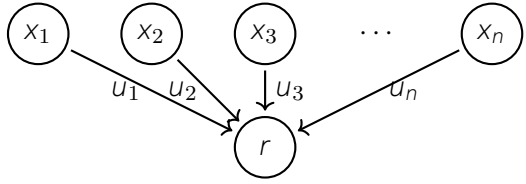
The Feedforward Neural Network

- Each input neuron feeds into every neuron.

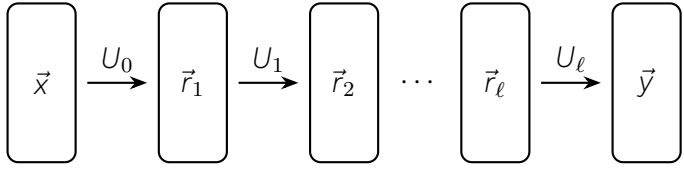


The Feedforward Neural Network

- Each input neuron feeds into every neuron.



- Having multiple layers creates a deep neural network.



- An example of this in the brain may be the visual cortex.

Synaptic Plasticity

- One way to train weights is through Hebbian plasticity: *"Neurons that fire together, wire together"*. Synaptic weights are updated based on pre-synaptic and post-synpatic activity. This is an example of unsupervised learning.

Synaptic Plasticity

- One way to train weights is through Hebbian plasticity: *"Neurons that fire together, wire together"*. Synaptic weights are updated based on pre-synaptic and post-synpatic activity. This is an example of unsupervised learning.
  - For example, weights may be updated according to

$$\Delta w \propto r_{\text{pre}} r_{\text{post}}$$

Synaptic Plasticity

- One way to train weights is through Hebbian plasticity: *"Neurons that fire together, wire together"*. Synaptic weights are updated based on pre-synaptic and post-synpatic activity. This is an example of unsupervised learning.
  - For example, weights may be updated according to

$$\Delta w \propto r_{\text{pre}} r_{\text{post}}$$

- Another way is supervised learning, which uses gradient descent on a loss function. This means weights are also updated by a third factor based on overall performance. Biologically, this can be thought of as neuromodulation, or the three-factor rule.

Synaptic Plasticity

- One way to train weights is through Hebbian plasticity: *"Neurons that fire together, wire together"*. Synaptic weights are updated based on pre-synaptic and post-synpatic activity. This is an example of unsupervised learning.
  - For example, weights may be updated according to

$$\Delta w \propto r_{\text{pre}} r_{\text{post}}$$

- Another way is supervised learning, which uses gradient descent on a loss function. This means weights are also updated by a third factor based on overall performance. Biologically, this can be thought of as neuromodulation, or the three-factor rule.
  - For example, weights may be updated according to

$$\Delta w \propto r_{\text{pre}}(r_{\text{post}} - \hat{r}_{\text{post}})$$

Backpropagation

- Since neural networks are just multiplying matrices, gradient descent is just a messy calculation involving the chain rule. The algorithm that does this efficiently is known as backpropagation.

Backpropagation

- Since neural networks are just multiplying matrices, gradient descent is just a messy calculation involving the chain rule. The algorithm that does this efficiently is known as backpropagation.
- Both backpropagation and neural networks have been known for a long time, but in 1986 Geoffrey Hinton (“the Godfather of AI”) popularized the technique.

Backpropagation

- Since neural networks are just multiplying matrices, gradient descent is just a messy calculation involving the chain rule. The algorithm that does this efficiently is known as backpropagation.
- Both backpropagation and neural networks have been known for a long time, but in 1986 Geoffrey Hinton (“the Godfather of AI”) popularized the technique.
- In 2012, Hinton along with his student Ilya Sutskever used these techniques, combined with much more data and processing power, to revolutionize computer vision. This contributed to the big machine learning boom.

Backpropagation

- Since neural networks are just multiplying matrices, gradient descent is just a messy calculation involving the chain rule. The algorithm that does this efficiently is known as backpropagation.
- Both backpropagation and neural networks have been known for a long time, but in 1986 Geoffrey Hinton (“the Godfather of AI”) popularized the technique.
- In 2012, Hinton along with his student Ilya Sutskever used these techniques, combined with much more data and processing power, to revolutionize computer vision. This contributed to the big machine learning boom.
- Hinton went on to work for Google and Sutskever became a co-founder of OpenAI. In 2023, they both left their respective companies.

Is Backpropagation Biologically Relevant?

- Backpropagation relies on knowing the values of all synaptic weight matrices and iteratively multiplying by them backwards.

$$\text{error at level } k = (f_k)' \cdot W_{k+1}^T \cdot (\text{error at level } k+1)$$

$$\Delta W_k \propto (\text{error at level } k) \cdot (\text{activity at level } k-1)$$

Is Backpropagation Biologically Relevant?

- Backpropagation relies on knowing the values of all synaptic weight matrices and iteratively multiplying by them backwards.

$$\text{error at level } k = (f_k)' \cdot W_{k+1}^T \cdot (\text{error at level } k+1)$$

$$\Delta W_k \propto (\text{error at level } k) \cdot (\text{activity at level } k-1)$$

- The problem is training is *not local*. In order to change synaptic weights, neurons would have to have a separate backward pass to inform of *all* synaptic weights forward in the network.

Biologically Plausible Training

An alternative is feedback alignment (Lillicrap et al., 2016).

Biologically Plausible Training

An alternative is feedback alignment (Lillicrap et al., 2016).

Instead of the weights W_k^T, we use a random (!!) matrix B.

Biologically Plausible Training

An alternative is feedback alignment (Lillicrap et al., 2016).

Instead of the weights W_k^T, we use a random (!!) matrix B.

At first, the updates $W_k^T \mathbf{e}$ and $B\mathbf{e}$ are likely unaligned.

Biologically Plausible Training

An alternative is feedback alignment (Lillicrap et al., 2016).

Instead of the weights W_k^T, we use a random (!!) matrix B.

At first, the updates $W_k^T\mathbf{e}$ and $B\mathbf{e}$ are likely unaligned.

But $W_k^T\mathbf{e}$ is in the direction of the gradient, or the direction of greatest change. So $B\mathbf{e}$ is likely to push more towards $W_k^T\mathbf{e}$.

Biologically Plausible Training

An alternative is feedback alignment (Lillicrap et al., 2016).

Instead of the weights W_k^T, we use a random (!!) matrix B.

At first, the updates $W_k^T\mathbf{e}$ and $B\mathbf{e}$ are likely unaligned.

But $W_k^T\mathbf{e}$ is in the direction of the gradient, or the direction of greatest change. So $B\mathbf{e}$ is likely to push more towards $W_k^T\mathbf{e}$.

Overtime, $B\mathbf{e}$ is more and more aligned in the direction of $W_k^T\mathbf{e}$.

Biologically Plausible Training

An alternative is feedback alignment (Lillicrap et al., 2016).

Instead of the weights W_k^T, we use a random (!!) matrix B.

At first, the updates $W_k^T\mathbf{e}$ and $B\mathbf{e}$ are likely unaligned.

But $W_k^T\mathbf{e}$ is in the direction of the gradient, or the direction of greatest change. So $B\mathbf{e}$ is likely to push more towards $W_k^T\mathbf{e}$.

Overtime, $B\mathbf{e}$ is more and more aligned in the direction of $W_k^T\mathbf{e}$.

We need on average that

$$(W_k^T\mathbf{e})^T B\mathbf{e} > 0 \qquad \text{so } \theta < 90^\circ$$
$$\mathbf{e}^T W_k B\mathbf{e} > 0$$

Performance vs. Efficiency

- Machine learning is mostly focused with evaluating model *performance*.

Performance vs. Efficiency

- Machine learning is mostly focused with evaluating model *performance*.
- But from the perspective of neuroscience, these models are super *inefficient* in the sense that it takes insane amounts of data, training, and processing power.
    - Your brain didn't need the entire internet and a datacenter to learn how to hold a conversation!

Performance vs. Efficiency

- Machine learning is mostly focused with evaluating model *performance*.
- But from the perspective of neuroscience, these models are super *inefficient* in the sense that it takes insane amounts of data, training, and processing power.
  - Your brain didn't need the entire internet and a datacenter to learn how to hold a conversation!
- Computational Neuroscience also focuses on biological relevance to learn how the brain actually processes information.

Performance vs. Efficiency

- Machine learning is mostly focused with evaluating model *performance*.
- But from the perspective of neuroscience, these models are super *inefficient* in the sense that it takes insane amounts of data, training, and processing power.
  - Your brain didn't need the entire internet and a datacenter to learn how to hold a conversation!
- Computational Neuroscience also focuses on biological relevance to learn how the brain actually processes information.
- The brain is the best computer we know: learning more about the brain can help enlighten machine learning.

Recurrent Neural Networks

Add coupled weights to our inputs

$$\tau^T \frac{d\mathbf{r}}{dt} = -\mathbf{r} + f(W\mathbf{r} + U\mathbf{x})$$

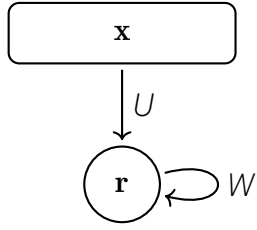
Our "steady-state" is now $\mathbf{r} \leftarrow f(W\mathbf{r} + U\mathbf{x})$, giving a recurrent neural network.

Recurrent Neural Networks

Add coupled weights to our inputs

$$\tau^T \frac{d\mathbf{r}}{dt} = -\mathbf{r} + f(W\mathbf{r} + U\mathbf{x})$$

Our "steady-state" is now $\mathbf{r} \leftarrow f(W\mathbf{r} + U\mathbf{x})$, giving a recurrent neural network.

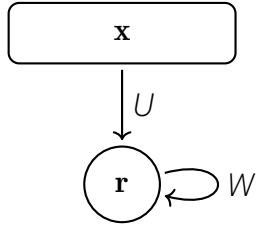


Recurrent Neural Networks

Add coupled weights to our inputs

$$\tau^T \frac{d\mathbf{r}}{dt} = -\mathbf{r} + f(W\mathbf{r} + U\mathbf{x})$$

Our "steady-state" is now $\mathbf{r} \leftarrow f(W\mathbf{r} + U\mathbf{x})$, giving a recurrent neural network.



Non-linear RNNs have effectively described many behaviors such as memory (Hopfield, 1982), decision making (Wong & Wang, 2006), and vision (Ozeki et al., 2009).

Recurrent Neural Networks

Observe a Linear RNN:

$$\mathbf{r} \leftarrow W\mathbf{r} + U\mathbf{x}$$

Recurrent Neural Networks

Observe a Linear RNN:

$$\mathbf{r} \leftarrow W\mathbf{r} + U\mathbf{x}$$
$$\mathbf{r} \leftarrow VDV^{-1}\mathbf{r} + U\mathbf{x}$$

Recurrent Neural Networks

Observe a Linear RNN:

$$\begin{aligned}
\mathbf{r} &\leftarrow W\mathbf{r} + U\mathbf{x} \\
\mathbf{r} &\leftarrow VDV^{-1}\mathbf{r} + U\mathbf{x} \\
V^{-1}\mathbf{r} &\leftarrow DV^{-1}\mathbf{r} + V^{-1}U\mathbf{x}
\end{aligned}$$

Recurrent Neural Networks

Observe a Linear RNN:

$$\mathbf{r} \leftarrow W\mathbf{r} + U\mathbf{x}$$
$$\mathbf{r} \leftarrow VDV^{-1}\mathbf{r} + U\mathbf{x}$$
$$V^{-1}\mathbf{r} \leftarrow DV^{-1}\mathbf{r} + V^{-1}U\mathbf{x}$$
$$\mathbf{a} \leftarrow D\mathbf{a} + B\mathbf{x}$$

decoupled!

Recurrent Neural Networks

Observe a Linear RNN:

$$\begin{aligned} \mathbf{r} &\leftarrow W\mathbf{r} + U\mathbf{x} \\ \mathbf{r} &\leftarrow VDV^{-1}\mathbf{r} + U\mathbf{x} \\ V^{-1}\mathbf{r} &\leftarrow DV^{-1}\mathbf{r} + V^{-1}U\mathbf{x} \\ \mathbf{a} &\leftarrow D\mathbf{a} + B\mathbf{x} \end{aligned}$$

decoupled!

Essentially puts $\mathbf{r}$ into a new coordinate system, given by the eigenvectors of W.

Recurrent Neural Networks

Return back to our full differential equation:

$$\tau_i \frac{da_i}{dt} = -a_i + \lambda a_i + \text{“inputs”}$$

Recurrent Neural Networks

Return back to our full differential equation:

$$\tau_i \frac{da_i}{dt} = -a_i + \lambda a_i + \text{“inputs”}$$

Rewrite as

$$\frac{\tau_i}{1-\lambda_i}\frac{da_i}{dt} = -a_i + \frac{\text{“inputs”}}{1-\lambda_i}$$

Recurrent Neural Networks

Return back to our full differential equation:

$$\tau_i \frac{da_i}{dt} = -a_i + \lambda a_i + \text{“inputs”}$$

Rewrite as

$$\frac{\tau_i}{1-\lambda_i}\frac{da_i}{dt} = -a_i + \frac{\text{“inputs”}}{1-\lambda_i}$$

- When $\lambda_i < 1$, this is exponential approach.
- When $\lambda_i > 1$, this is exponential growth.

Hence, larger λ_i contribute more to the overall system. This is principal component analysis (PCA).

Underlying Latent Dynamics

- Dimensionality reduction helps discover macro-level latent dynamics.

Underlying Latent Dynamics

- Dimensionality reduction helps discover macro-level latent dynamics.
- This has been shown to be important to understanding certain complex behaviors such as motor control (Vyas et al., 2020).

Underlying Latent Dynamics

- Dimensionality reduction helps discover macro-level latent dynamics.
- This has been shown to be important to understanding certain complex behaviors such as motor control (Vyas et al., 2020).
- Research shows that brain activity is largely dominated by few internal signals, such as in decision making (Bondy et al., 2024).

Underlying Latent Dynamics

- Dimensionality reduction helps discover macro-level latent dynamics.
- This has been shown to be important to understanding certain complex behaviors such as motor control (Vyas et al., 2020).
- Research shows that brain activity is largely dominated by few internal signals, such as in decision making (Bondy et al., 2024).
- In other words, the brain is doing a lot with seemingly very little!
- Thus, one might shift away from model training and more towards understanding optimal latent dynamics and inputs.

Table of Contents

Latent Dynamics Model

Assume the following RNN architecture (Schimel et al., 2022):

$$\begin{aligned} \text{latent dynamics} \quad & \mathbf{r}_{t+1} = f_\theta(\mathbf{r}_t, \mathbf{x}_t) \\ \text{observations} \quad & \mathbf{o}_t | \mathbf{r}_t \sim p_\theta(\mathbf{o}_t | \mathbf{r}_t) \end{aligned}$$

Latent Dynamics Model

Assume the following RNN architecture (Schimel et al., 2022):

$$\begin{aligned} \text{latent dynamics} \quad & \mathbf{r}_{t+1} = f_\theta(\mathbf{r}_t, \mathbf{x}_t) \\ \text{observations} \quad & \mathbf{o}_t|\mathbf{r}_t \sim p_\theta(\mathbf{o}_t|\mathbf{r}_t) \end{aligned}$$

Low dimensional neuronal dynamics $\mathbf{r}$ are controlled by an RNN f_θ with inputs $\mathbf{x}$.

Latent Dynamics Model

Assume the following RNN architecture (Schimel et al., 2022):

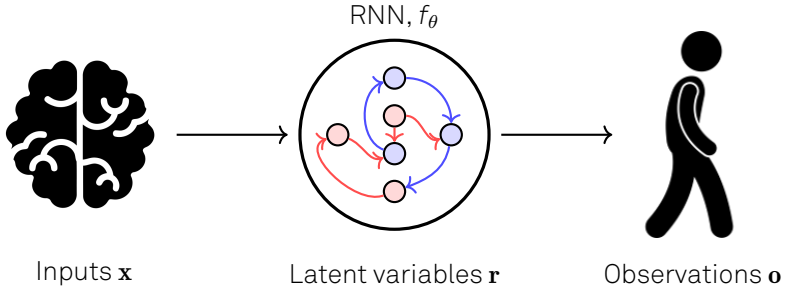
$$\begin{aligned} \text{latent dynamics} \quad & \mathbf{r}_{t+1} = f_\theta(\mathbf{r}_t, \mathbf{x}_t) \\ \text{observations} \quad & \mathbf{o}_t|\mathbf{r}_t \sim p_\theta(\mathbf{o}_t|\mathbf{r}_t) \end{aligned}$$

Low dimensional neuronal dynamics $\mathbf{r}$ are controlled by an RNN f_θ with inputs $\mathbf{x}$.

But actual behaviors $\mathbf{o}$ vary according to some probability function p_θ.

- $p_\theta(\mathbf{o}_t|\mathbf{r}_t)$ is chosen to be either a normal or a student t-distribution.

Latent Dynamics Model



iLQR - VAE

Estimate $q_\phi(\mathbf{x}_t|\mathbf{o}_t)$ based on *optimal inputs*. This is through an iterative Linear Quadratic Regulator (Li & Todorov, 2004).

iLQR - VAE

Estimate $q_\phi(\mathbf{x}_t|\mathbf{o}_t)$ based on *optimal inputs*. This is through an iterative Linear Quadratic Regulator (Li & Todorov, 2004).

- We change inputs via a running cost based on differences in desired observations and actual observations given by $\log p_\theta(\mathbf{o}_t|\mathbf{x}_t)$.

iLQR - VAE

Estimate $q_\phi(\mathbf{x}_t|\mathbf{o}_t)$ based on *optimal inputs*. This is through an iterative Linear Quadratic Regulator (Li & Todorov, 2004).

- We change inputs via a running cost based on differences in desired observations and actual observations given by $\log p_\theta(\mathbf{o}_t|\mathbf{x}_t)$.
- Inputs are made optimal by an energetic cost given by $\log p_\theta(\mathbf{x}_t)$.

iLQR - VAE

Estimate $q_\phi(\mathbf{x}_t|\mathbf{o}_t)$ based on *optimal inputs*. This is through an iterative Linear Quadratic Regulator (Li & Todorov, 2004).

- We change inputs via a running cost based on differences in desired observations and actual observations given by $\log p_\theta(\mathbf{o}_t|\mathbf{x}_t)$.
- Inputs are made optimal by an energetic cost given by $\log p_\theta(\mathbf{x}_t)$.

$$\mathbf{x}^\star(t) = \arg\min_{\mathbf{x}} [\log p_\theta(\mathbf{o}_t|\mathbf{x}_t) + \log p_\theta(\mathbf{x}_t)]$$

iLQR - VAE

Estimate $q_\phi(\mathbf{x}_t|\mathbf{o}_t)$ based on *optimal inputs*. This is through an iterative Linear Quadratic Regulator (Li & Todorov, 2004).

- We change inputs via a running cost based on differences in desired observations and actual observations given by $\log p_\theta(\mathbf{o}_t|\mathbf{x}_t)$.
- Inputs are made optimal by an energetic cost given by $\log p_\theta(\mathbf{x}_t)$.

$$\mathbf{x}^\star(t) = \arg\min_{\mathbf{x}} [\log p_\theta(\mathbf{o}_t|\mathbf{x}_t) + \log p_\theta(\mathbf{x}_t)]$$

The resulting function q is a normal distribution centered at the optimal inputs $\mathbf{x}^\star$.

iLQR - VAE

Finally, we optimize our parameters θ, ϕ on our RNN based on a cost function on our latent variables $\mathbf{r}_t$ gathered by our optimal inputs.

iLQR - VAE

Finally, we optimize our parameters θ, ϕ on our RNN based on a cost function on our latent variables $\mathbf{r}_t$ gathered by our optimal inputs.

Done through variational inference, which aims to make $\log p_\theta(\mathbf{r}_t|\mathbf{o}_t)$ close to what we expect based on optimal inputs from $\log q_\phi(\mathbf{x}_t|\mathbf{o}_t)$

iLQR - VAE

Finally, we optimize our parameters θ, ϕ on our RNN based on a cost function on our latent variables $\mathbf{r}_t$ gathered by our optimal inputs.

Done through variational inference, which aims to make $\log p_\theta(\mathbf{r}_t|\mathbf{o}_t)$ close to what we expect based on optimal inputs from $\log q_\phi(\mathbf{x}_t|\mathbf{o}_t)$

The resulting algorithm aims to optimize both inputs and parameters to make the system efficient at achieving desired behavior.

iLQR - VAE

Recall:

$$\text{latent dynamics} \quad \mathbf{r}_{t+1} = f_\theta(\mathbf{r}_t, \mathbf{x}_t, t)$$
$$\text{observations} \quad \mathbf{o}_t|\mathbf{r}_t \sim p_\theta(\mathbf{o}_t|\mathbf{r}_t)$$

iLQR - VAE

Recall:

$$\begin{aligned} \text{latent dynamics} \quad & \mathbf{r}_{t+1} = f_\theta(\mathbf{r}_t, \mathbf{x}_t, t) \\ \text{observations} \quad & \mathbf{o}_t|\mathbf{r}_t \sim p_\theta(\mathbf{o}_t|\mathbf{r}_t) \end{aligned}$$

The key is choosing the right RNN!

iLQR - VAE

Recall:

$$\begin{aligned} \text{latent dynamics} \quad & \mathbf{r}_{t+1} = f_\theta(\mathbf{r}_t, \mathbf{x}_t, t) \\ \text{observations} \quad & \mathbf{o}_t|\mathbf{r}_t \sim p_\theta(\mathbf{o}_t|\mathbf{r}_t) \end{aligned}$$

The key is choosing the right RNN!

Traditional powerful RNN's are able to recover optimal inputs from motor behavior in zebrafish, but due to biological irrelevance, latent variables are less informative (Mullen et al., 2024)

iLQR - VAE

Recall:

$$\begin{aligned} \text{latent dynamics} \quad & \mathbf{r}_{t+1} = f_\theta(\mathbf{r}_t, \mathbf{x}_t, t) \\ \text{observations} \quad & \mathbf{o}_t|\mathbf{r}_t \sim p_\theta(\mathbf{o}_t|\mathbf{r}_t) \end{aligned}$$

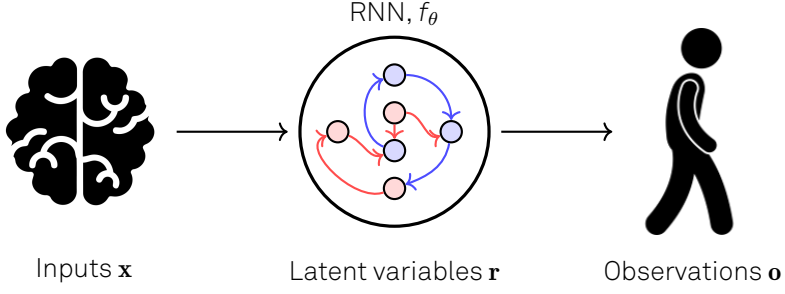
The key is choosing the right RNN!

Traditional powerful RNN's are able to recover optimal inputs from motor behavior in zebrafish, but due to biological irrelevance, latent variables are less informative (Mullen et al., 2024)

A biologically relevant RNN could better explain what the brain is actually doing, illuminating how it efficiently performs calculations.

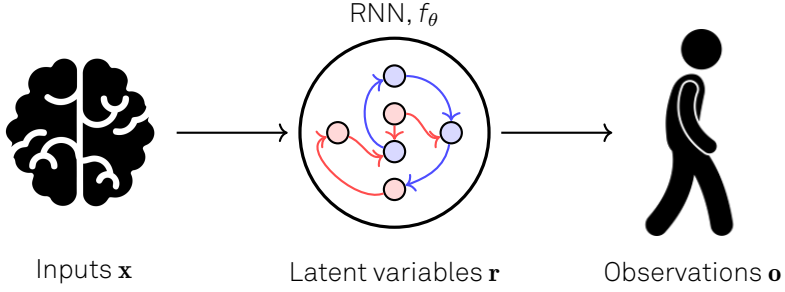
Machine Learning & Neuroscience

- The brain works efficiently.



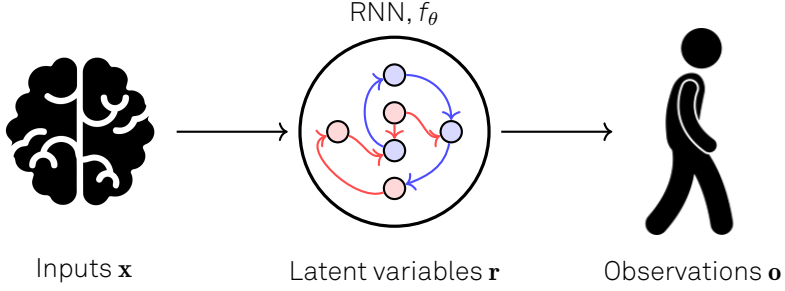
Machine Learning & Neuroscience

- The brain works efficiently.
- Modeling the brain is key to replicating that efficiency.



Machine Learning & Neuroscience

- The brain works efficiently.
- Modeling the brain is key to replicating that efficiency.
- But if not biologically relevant, we can't get that efficiency.



Machine Learning & Neuroscience

- The brain works efficiently.
- Modeling the brain is key to replicating that efficiency.
- But if not biologically relevant, we can't get that efficiency.
- More importantly, if not biologically relevant, we can't *understand* that efficiency.