SYDE 750 Project

Learning Probability Distributions for Statistical Inference

Andreas Stöckel

March 5th, 2017

Lifespan Inference Task (I)

Based on *Optimal Predictions in Everyday Cognition* by Griffiths and Tenenbaum (2006)

- Experimental task...

  Insurance agencies employ actuaries to make predictions about people's lifespans—the age at which they will die—based upon demographic information. If you were assessing an insurance case for an 18-year-old man, what would you predict for his lifespan?

Lifespan Inference Task (I)

Based on *Optimal Predictions in Everyday Cognition* by Griffiths and Tenenbaum (2006)

- Experimental task...

  Insurance agencies employ actuaries to make predictions about people's lifespans—the age at which they will die—based upon demographic information. If you were assessing an insurance case for an 18-year-old man, what would you predict for his lifespan?

- ...as Bayesian inference

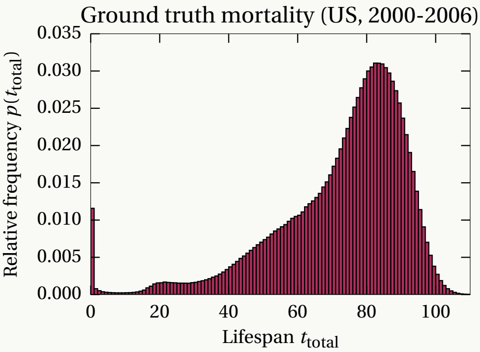
  Given t, estimate t_{total}

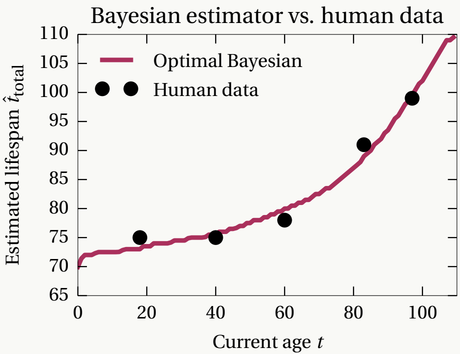
$$p(t_{\text{total}} \mid t) = \frac{p(t \mid t_{\text{total}}) \cdot p(t_{\text{total}})}{p(t)} .$$

  Median estimator, select $\hat{t}_{\text{total}}$ s.t.

$$p(t_{\text{total}} > \hat{t}_{\text{total}} \mid t) = p(t_{\text{total}} < \hat{t}_{\text{total}} \mid t) .$$

Lifespan Inference Task (II)





Mortality data: Human Mortality Database. University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany). Available at `www.mortality.org` (data downloaded on 2017/03/29). *Human experiment results adapted from:* Optimal Predictions in Everyday Cognition. Griffiths and Tenenbaum (2006)

Outline

- Motivation
- Part I: Probability Distribution Representation
  - Non-negative Mixture Model
  - Implementation
  - Results
- Part II: Lifespan Inference
  - Median Estimation as Gradient Descent
  - Implementation
  - Results
- Conclusion

PART I

Probability Distribution Representation

Non-negative Mixture Model

Goal

Find $p(x)$ which approximates empirical distribution $\mathfrak{P}$. Given samples

$$\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_N\},$$
$$\mathfrak{P}(x \mid \hat{X}) = \frac{1}{N} \cdot \sum_{i=1}^{N} \delta(x - \hat{x}_i)\,.$$

Idea

Represent $p(x)$ as non-negative mixture of k basis functions

$$p(x) = \frac{\sum_{i=1}^{k} w_i \cdot \phi_i(x)}{\sum_{i=1}^{k} w_i \cdot \int_{-\infty}^{\infty} \phi_i(x')\,\mathrm{d}x'}\,,$$

where $w_i \geq 0$, $\phi_i(x) \geq 0$, $\int p(x)\,\mathrm{d}x = 1$.

Non-negative Mixture Model

Goal

Find $p(x)$ which approximates empirical distribution $\mathfrak{P}$. Given samples

$$\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_N\},$$

$$\mathfrak{P}(x \mid \hat{X}) = \sum_{i=1}^{N} \delta(x - \hat{x}_i)\,.$$

Idea

Represent $p(x)$ as non-negative mixture of k basis functions

$$p(x) = \sum_{i=1}^{k} w_i \cdot \phi_i(x)\,,$$

where $w_i \geq 0$, $\phi_i(x) \geq 0$.

Ignore Normalization!

Learning Probability Distributions

Batch Learning

Learning Probability Distributions

Batch Learning

- Given $\hat{X}$ find $\vec{w}$ s.t.

$$p(x \mid \vec{w}) \approx \mathfrak{P}(x \mid \hat{X})$$

Learning Probability Distributions

Batch Learning

- Given $\hat{X}$ find $\vec{w}$ s.t.

$$p(x \mid \vec{w}) \approx \mathfrak{P}(x \mid \hat{X})$$

- Expectation Maximization!

Learning Probability Distributions

Batch Learning

- Given $\hat{X}$ find $\vec{w}$ s.t.

$$p(x \mid \vec{w}) \approx \mathfrak{P}(x \mid \hat{X})$$

- Expectation Maximization!
- Nope, really just L_2 regression (nonparametric ϕ_i, single E-step)

Learning Probability Distributions

Batch Learning

- Given $\hat{X}$ find $\vec{w}$ s.t.

$$p(x \mid \vec{w}) \approx \mathfrak{P}(x \mid \hat{X})$$

- Expectation Maximization!
- Nope, really just L_2 regression (nonparametric ϕ_i, single E-step)

Online Learning

- For each sample $\hat{x}_i$ find new $\vec{w}'$ s.t.

$$p(x \mid \vec{w}') \approx p(x \mid \vec{w}) + \delta(x - \hat{x}_i)$$

Learning Probability Distributions

Batch Learning

- Given $\hat{X}$ find $\vec{w}$ s.t.

$$p(x \mid \vec{w}) \approx \mathfrak{P}(x \mid \hat{X})$$

- Expectation Maximization!
- Nope, really just L_2 regression (nonparametric ϕ_i, single E-step)

Online Learning

- For each sample $\hat{x}_i$ find new $\vec{w}'$ s.t.

$$p(x \mid \vec{w}') \approx p(x \mid \vec{w}) + \delta(x - \hat{x}_i)$$

- Online Expectation Maximization!

Learning Probability Distributions

Batch Learning

- Given $\hat{X}$ find $\vec{w}$ s.t.

$$p(x \mid \vec{w}) \approx \mathfrak{P}(x \mid \hat{X})$$

- Expectation Maximization!
- Nope, really just L_2 regression (nonparametric ϕ_i, single E-step)

Online Learning

- For each sample $\hat{x}_i$ find new $\vec{w}'$ s.t.

$$p(x \mid \vec{w}') \approx p(x \mid \vec{w}) + \delta(x - \hat{x}_i)$$

- Online Expectation Maximization!
- Again, way too complex...

Learning Probability Distributions – Optimal online update rule

For each new sample $\hat{x}$ minimize quadratic approximation error w.r.t. $\Delta w_j = w'_j - w_j$:

$$\frac{\partial}{\partial \Delta w_j} E = \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(p(x \mid \vec{w}') - p(x \mid \vec{w}) - \delta(x - \hat{x}) \right)^2 \, \mathrm{d}x$$

Learning Probability Distributions – Optimal online update rule

For each new sample $\hat{x}$ minimize quadratic approximation error w.r.t. $\Delta w_j = w'_j - w_j$:

$$\begin{aligned}
\frac{\partial}{\partial \Delta w_j} E &= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \big(p(x \mid \vec{w}') - p(x \mid \vec{w}) - \delta(x - \hat{x})\big)^2 \,\mathrm{d}x \\
&= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(\sum_{i=1}^{k} \Delta w_i \cdot \phi_i(x) - \delta(x - \hat{x}) \right)^2 \,\mathrm{d}x
\end{aligned}$$

Learning Probability Distributions – Optimal online update rule

For each new sample $\hat{x}$ minimize quadratic approximation error w.r.t. $\Delta w_j = w'_j - w_j$:

$$\begin{aligned}
\frac{\partial}{\partial \Delta w_j} E &= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(p(x \mid \vec{w}') - p(x \mid \vec{w}) - \delta(x - \hat{x})\right)^2 \, \mathrm{d}x \\
&= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(\sum_{i=1}^{k} \Delta w_i \cdot \phi_i(x) - \delta(x - \hat{x}) \right)^2 \, \mathrm{d}x \\
&= \int_{-\infty}^{\infty} \sum_{i=1}^{k} \Delta w_i \cdot \phi_j(x) \cdot \phi_i(x) - \phi_j(x) \cdot \delta(x - \hat{x}) \, \mathrm{d}x
\end{aligned}$$

Learning Probability Distributions – Optimal online update rule

For each new sample $\hat{x}$ minimize quadratic approximation error w.r.t. $\Delta w_j = w'_j - w_j$:

$$
\begin{aligned}
\frac{\partial}{\partial \Delta w_j} E &= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(p(x \mid \vec{w}') - p(x \mid \vec{w}) - \delta(x - \hat{x}) \right)^2 \, \mathrm{d}x \\
&= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(\sum_{i=1}^{k} \Delta w_i \cdot \phi_i(x) - \delta(x - \hat{x}) \right)^2 \, \mathrm{d}x \\
&= \int_{-\infty}^{\infty} \sum_{i=1}^{k} \Delta w_i \cdot \phi_j(x) \cdot \phi_i(x) - \phi_j(x) \cdot \delta(x - \hat{x}) \, \mathrm{d}x \\
&= \sum_{i=1}^{k} \Delta w_i \cdot \underbrace{\int_{-\infty}^{\infty} \phi_j(x) \cdot \phi_i(x) \, \mathrm{d}x}_{\gamma_{ij}} - \phi_j(\hat{x}) \stackrel{!}{=} 0
\end{aligned}
$$

Learning Probability Distributions – Optimal online update rule

For each new sample $\hat{x}$ minimize quadratic approximation error w.r.t. $\Delta w_j = w'_j - w_j$:

$$
\begin{aligned}
\frac{\partial}{\partial \Delta w_j} E &= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(p(x \mid \vec{w}') - p(x \mid \vec{w}) - \delta(x - \hat{x})\right)^2 \, \mathrm{d}x \\
&= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(\sum_{i=1}^{k} \Delta w_i \cdot \phi_i(x) - \delta(x - \hat{x}) \right)^2 \, \mathrm{d}x \\
&= \int_{-\infty}^{\infty} \sum_{i=1}^{k} \Delta w_i \cdot \phi_j(x) \cdot \phi_i(x) - \phi_j(x) \cdot \delta(x - \hat{x}) \, \mathrm{d}x \\
&= \sum_{i=1}^{k} \Delta w_i \cdot \underbrace{\int_{-\infty}^{\infty} \phi_j(x) \cdot \phi_i(x) \, \mathrm{d}x}_{\gamma_{ij}} - \phi_j(\hat{x}) \overset{!}{=} 0
\end{aligned}
$$

$$\Leftrightarrow \Delta \vec{w} = \Gamma^{-1} \cdot \vec{\phi}(\hat{x})$$

Learning Probability Distributions – Optimal online update rule

For each new sample $\hat{x}$ minimize quadratic approximation error w.r.t. $\Delta w_j = w'_j - w_j$:

$$
\begin{aligned}
\frac{\partial}{\partial \Delta w_j} E &= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(p(x \mid \vec{w}') - p(x \mid \vec{w}) - \delta(x - \hat{x})\right)^2 \; \mathrm{d}x \\
&= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(\sum_{i=1}^{k} \Delta w_i \cdot \phi_i(x) - \delta(x - \hat{x}) \right)^2 \; \mathrm{d}x \\
&= \int_{-\infty}^{\infty} \sum_{i=1}^{k} \Delta w_i \cdot \phi_j(x) \cdot \phi_i(x) - \phi_j(x) \cdot \delta(x - \hat{x}) \; \mathrm{d}x \\
&= \sum_{i=1}^{k} \Delta w_i \cdot \underbrace{\int_{-\infty}^{\infty} \phi_j(x) \cdot \phi_i(x) \; \mathrm{d}x}_{\gamma_{ij}} - \phi_j(\hat{x}) \stackrel{!}{=} 0
\end{aligned}
$$

$\Leftrightarrow \Delta \vec{w} = \Gamma^{-1} \cdot \vec{\phi}(\hat{x}) \Rightarrow \Delta \vec{w}$ linear combination of $\vec{\phi}(\hat{x})$!

Learning Probability Distributions – Optimal online update rule

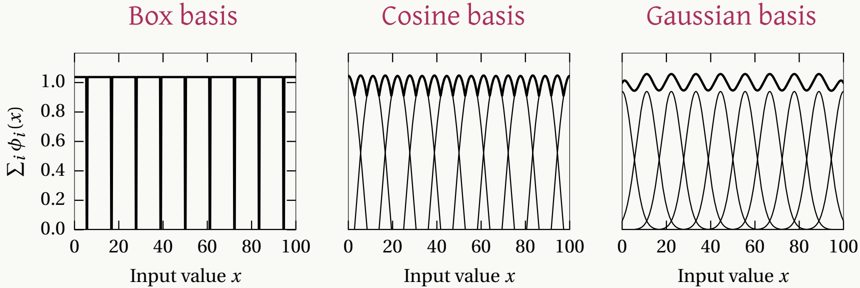
For each new sample $\hat{x}$ minimize quadratic approximation error w.r.t. $\Delta w_j = w'_j - w_j$:

$$
\begin{aligned}
\frac{\partial}{\partial \Delta w_j} E &= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(p(x \mid \vec{w}') - p(x \mid \vec{w}) - \delta(x - \hat{x})\right)^2 \,\mathrm{d}x \\
&= \frac{\partial}{\partial \Delta w_j} \int_{-\infty}^{\infty} \frac{1}{2} \left(\sum_{i=1}^{k} \Delta w_i \cdot \phi_i(x) - \delta(x - \hat{x}) \right)^2 \,\mathrm{d}x \\
&= \int_{-\infty}^{\infty} \sum_{i=1}^{k} \Delta w_i \cdot \phi_j(x) \cdot \phi_i(x) - \phi_j(x) \cdot \delta(x - \hat{x}) \,\mathrm{d}x \\
&= \sum_{i=1}^{k} \Delta w_i \cdot \underbrace{\int_{-\infty}^{\infty} \phi_j(x) \cdot \phi_i(x) \,\mathrm{d}x}_{\gamma_{ij}} - \phi_j(\hat{x}) \overset{!}{=} 0
\end{aligned}
$$

$\Leftrightarrow \Delta \vec{w} = \Gamma^{-1} \cdot \vec{\phi}(\hat{x})$ $\Rightarrow$ $\Delta \vec{w}$ linear combination of $\vec{\phi}(\hat{x})$! 🐰

Basis Functions – Radial basis

Radial basis function $\phi_i(x) = \phi\left(\frac{\|x - x_i\|}{\sigma}\right)$



$$\phi^{\text{box}}(r) = \begin{cases} 1 & \text{if } r \leq 1 \\ 0 & \text{if } r > 1 \end{cases}, \quad \phi^{\cos}(r) = \begin{cases} \cos(r) & \text{if } r \leq \frac{\pi}{2} \\ 0 & \text{if } r > \frac{\pi}{2} \end{cases}, \quad \phi^{\text{gauss}}(r) = \exp\left(-r^2\right)$$

Basis Functions – How to select σ?

Radial basis function $\phi_i(x) = \phi\left(\frac{\|x - x_i\|}{\sigma}\right)$

Matrix of inner products $(\Gamma)_{ij} = \gamma_{ij} = \int_{-\infty}^{\infty} \phi_j(x) \cdot \phi_i(x)\, \mathrm{d}x$

Two constraints

- Near-orthogonal basis

$$\Gamma = I = \Gamma^{-1} \Rightarrow \Delta\vec{w} = \vec{\phi}(\hat{x})$$

Basis Functions – How to select σ?

Radial basis function $\phi_i(x) = \phi\left(\frac{\|x-x_i\|}{\sigma}\right)$

Matrix of inner products $(\Gamma)_{ij} = \gamma_{ij} = \int_{-\infty}^{\infty} \phi_j(x) \cdot \phi_i(x)\ \mathrm{d}x$

Two constraints

- Near-orthogonal basis

$$\Gamma = I = \Gamma^{-1} \Rightarrow \Delta\vec{w} = \vec{\phi}(\hat{x})$$

- Constant support

$$\sum_{i=1}^{k} \phi_i(x) - \phi_i(y) \approx 0\,\forall x, y$$

Basis Functions – How to select σ?

Radial basis function $\phi_i(x) = \phi\left(\frac{\|x - x_i\|}{\sigma}\right)$

Matrix of inner products $(\Gamma)_{ij} = \gamma_{ij} = \int_{-\infty}^{\infty} \phi_j(x) \cdot \phi_i(x)\, \mathrm{d}x$

Two constraints

- Near-orthogonal basis

$$\Gamma = I = \Gamma^{-1} \Rightarrow \Delta\vec{w} = \vec{\phi}(\hat{x})$$

- Constant support

$$\sum_{i=1}^{k} \phi_i(x) - \phi_i(y) \approx 0\, \forall x, y$$

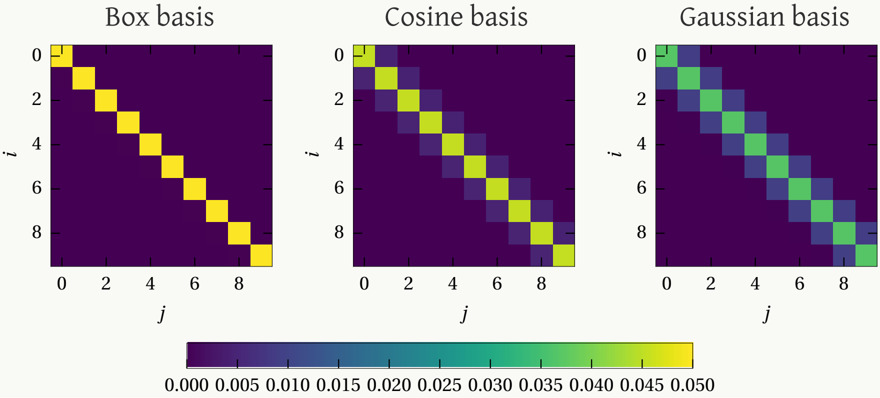
Heuristic

Numerically minimize w.r.t. σ

$$E(\sigma) = \left(1 - \min_x(f(x;\sigma))\right)^2 + \left(1 - \max_x(f(x;\sigma))\right)^2$$

$$\text{where } f(x;\sigma) = \sum_{i=1}^{k} \phi_i(x)$$

Basis Functions – Γ matrices

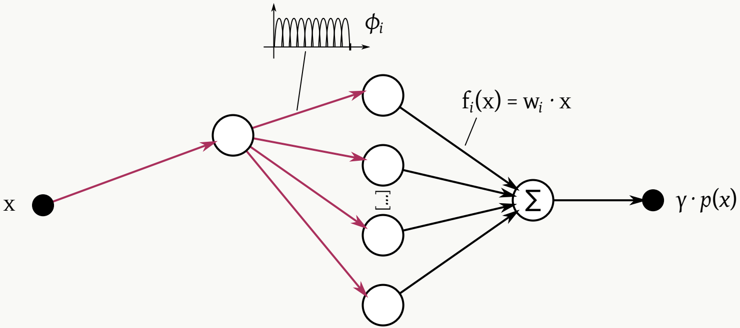


After thorough visual inspection we conclude

$$\Gamma \approx I$$

Probability Distribution Network Implementation (I)

Implementation of $p(x) = \sum_{i=1}^{k} w_i \cdot \phi_i(x)$ as a neural network



Learning Rule

Idea

Use Prescribed Error Sensitivity (PES) rule to learn functions f_i

Derivation

- *Assumption:*
  Current function $f_i(x) = w_i \cdot x$
- *Desired:*
  Updated function $f_i'(x) = w_i' \cdot x$

Learning Rule

Idea

Use Prescribed Error Sensitivity (PES) rule to learn functions f_i

Derivation

- *Assumption:*
  Current function $f_i(x) = w_i \cdot x$
- *Desired:*
  Updated function $f_i'(x) = w_i' \cdot x$

$$E_i = f_i(\phi_i(x)) - f_i'(\phi_i(x))$$

Learning Rule

Idea

Use Prescribed Error Sensitivity (PES) rule to learn functions f_i

Derivation

- *Assumption:* Current function $f_i(x) = w_i \cdot x$
- *Desired:* Updated function $f_i'(x) = w_i' \cdot x$

$$
\begin{aligned}
E_i &= f_i(\phi_i(x)) - f_i'(\phi_i(x)) \\
&= w_i \cdot \phi_i(x) - w_i' \cdot \phi_i(x) = -\Delta w_i \cdot \phi_i(x)
\end{aligned}
$$

Learning Rule

Idea

Use Prescribed Error Sensitivity (PES) rule to learn functions f_i

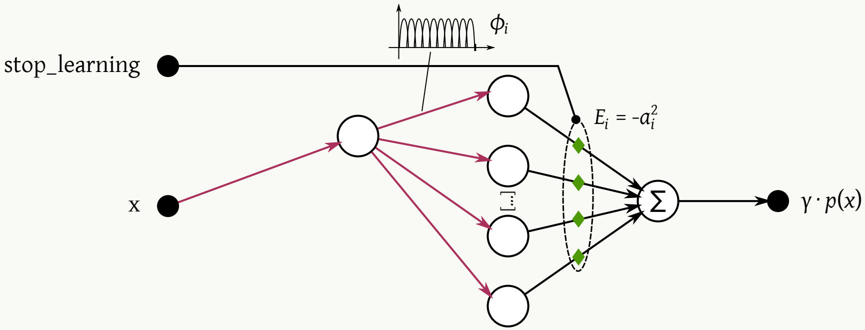
Derivation

- *Assumption:*
  Current function $f_i(x) = w_i \cdot x$
- *Desired:*
  Updated function $f_i'(x) = w_i' \cdot x$

$$\begin{aligned} E_i &= f_i(\phi_i(x)) - f_i'(\phi_i(x)) \\ &= w_i \cdot \phi_i(x) - w_i' \cdot \phi_i(x) = -\Delta w_i \cdot \phi_i(x) \\ &= -(\Gamma^{-1} \cdot \vec{\phi}(x))_i \cdot \phi_i(x) \approx -(I \cdot \vec{\phi}(x))_i \cdot \phi_i(x) = -\phi_i(x)^2 \end{aligned}$$

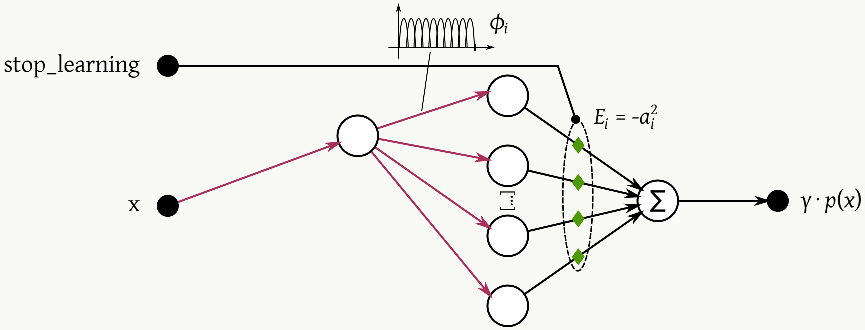
Probability Distribution Network Implementation (II)

Implementation of $p(x) = \sum_{i=1}^{k} w_i \cdot \phi_i(x)$ as a neural network



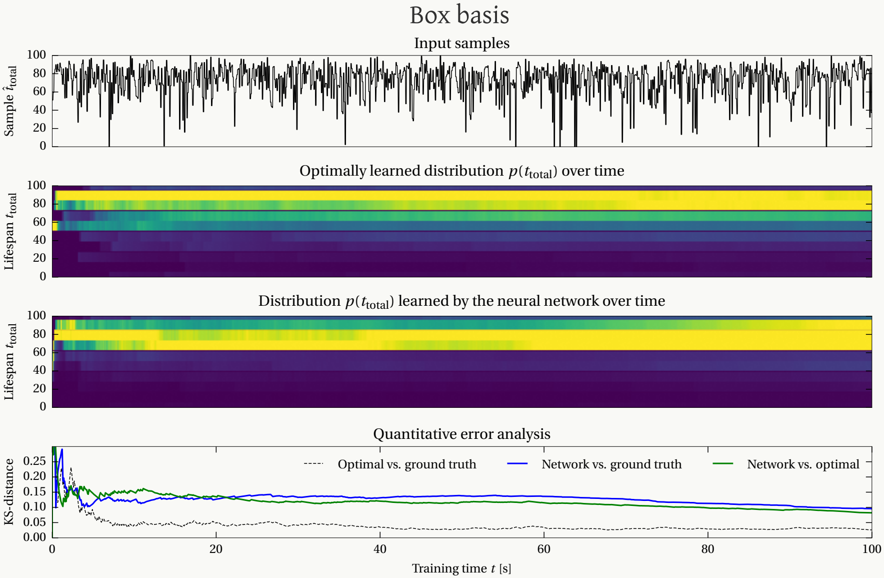
Probability Distribution Network Implementation (II)

Implementation of $p(x) = \sum_{i=1}^{k} w_i \cdot \phi_i(x)$ as a neural network

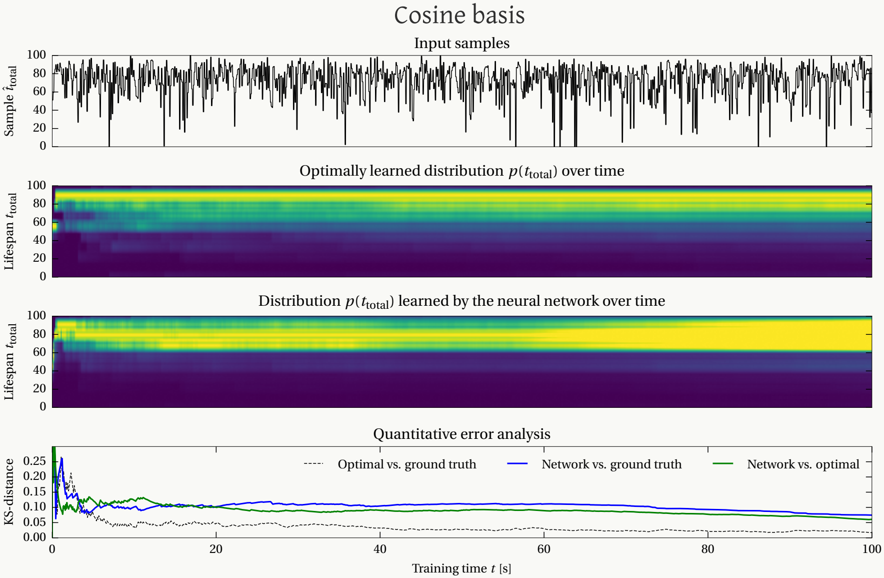


Important: Select tuning curves for E_i populations such that $\vec{a}_i = 0 \Leftrightarrow E_i = 0$

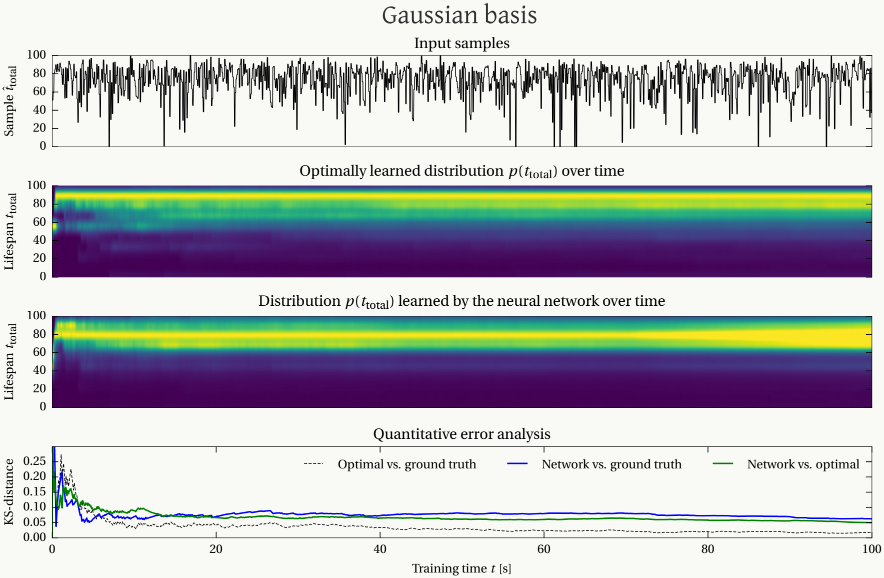
Experimental results (I)



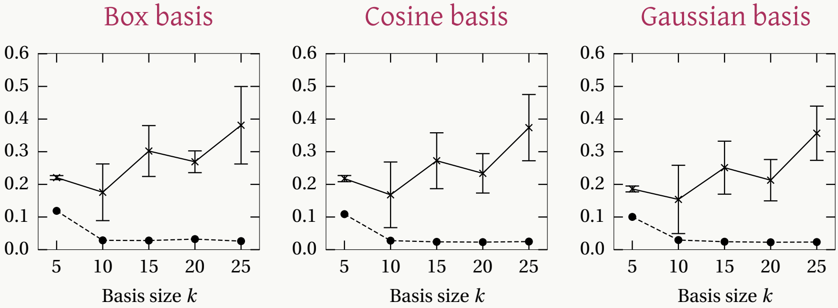
Experimental results (I)



Experimental results (I)



Experimental results (II)



Dashed line: optimal vs. ground truth. *Solid line:* learned vs. ground truth.

Errors expressed in a two-sample Kolmogorov-Smirnov metric.
Mean, standard deviation over $N = 5$ trials.

Part I – Summary

What I've shown you so far...

- We are able to represent a prior probability distribution $p(t_{\text{total}})$.
- Distribution implicitly stored as a weight vector $\vec{w}$ in the connections between neuron ensembles.
- Learning of $\vec{w}$ using a local PES rule.

Part I – Summary

What I've shown you so far...

- We are able to represent a prior probability distribution $p(t_{\text{total}})$.
- Distribution implicitly stored as a weight vector $\vec{w}$ in the connections between neuron ensembles.
- Learning of $\vec{w}$ using a local PES rule.

What is missing?

- Representation of the posterior $p(t_{\text{total}} \mid t)$.
- Median estimation.

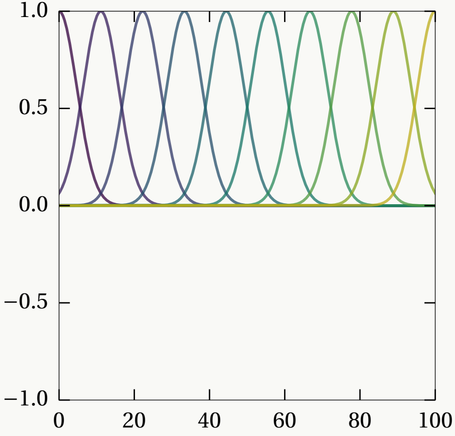
PART II

Lifespan Inference

Basis function transformation (I)

Probability distribution $p(x)$

$$p(x) = \sum_{i=1}^{k} w_i \cdot \phi_i(x)$$

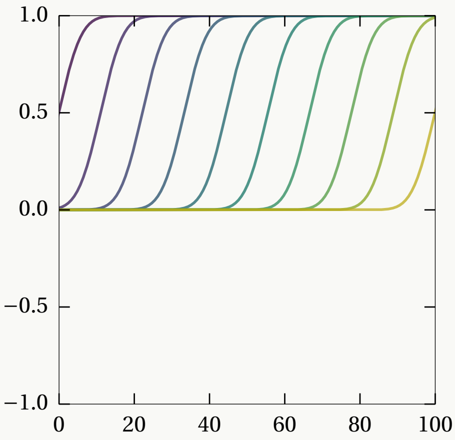


Basis functions $\phi_i(x)$

Basis function transformation (I)

Cumulative distribution $p(x \leq x')$

$$
\begin{aligned}
p(x \leq x') &= \int_{-\infty}^{x'} \sum_{i=1}^{k} w_i \cdot \phi_i(x)\,\mathrm{d}x \\
&= \sum_{i=1}^{k} w_i \cdot \int_{-\infty}^{x'} \phi_i(x)\,\mathrm{d}x \\
&= \sum_{i=1}^{k} w_i \cdot \Phi_i'(x')
\end{aligned}
$$



Basis functions $\Phi_i'(x)$

The Median Gradient

Idea

- Per definition $g(x') = p(x \leq x') - p(x \geq x') = 0 \Leftrightarrow x'$ is the median of $p(x)$.

The Median Gradient

Idea

- Per definition $g(x') = p(x \leq x') - p(x \geq x') = 0 \Leftrightarrow x'$ is the median of $p(x)$.

$\Rightarrow$ Dynamical system $\frac{\mathrm{d}}{\mathrm{d}t}x' = -g(x')$ converges to the median over time t!

The Median Gradient

Idea

- Per definition $g(x') = p(x \leq x') - p(x \geq x') = 0 \Leftrightarrow x'$ is the median of $p(x)$.

$\Rightarrow$ Dynamical system $\frac{\mathrm{d}}{\mathrm{d}t}x' = -g(x')$ converges to the median over time t!

- *Note:* In contrast to $p(x) = 0.5$ above median definition invariant to normalization.

The Median Gradient

Idea

- Per definition $g(x') = p(x \leq x') - p(x \geq x') = 0 \Leftrightarrow x'$ is the median of $p(x)$.

$\Rightarrow$ Dynamical system $\frac{\mathrm{d}}{\mathrm{d}t}x' = -g(x')$ converges to the median over time t!

- *Note:* In contrast to $p(x) = 0.5$ above median definition invariant to normalization.

The Median Gradient

Idea

- Per definition $g(x') = p(x \leq x') - p(x \geq x') = 0 \Leftrightarrow x'$ is the median of $p(x)$.

$\Rightarrow$ Dynamical system $\frac{\mathrm{d}}{\mathrm{d}t} x' = -g(x')$ converges to the median over time t!

- *Note:* In contrast to $p(x) = 0.5$ above median definition invariant to normalization.

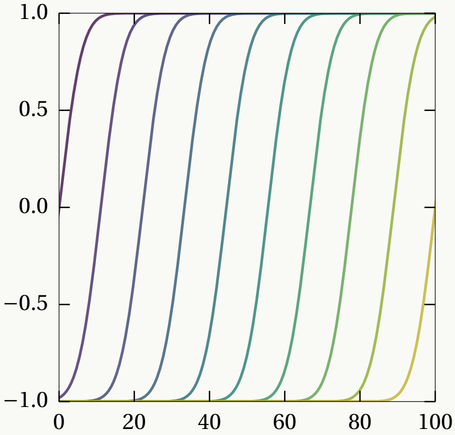
Basis transformation

$$
\begin{aligned}
p(x \leq x') - p(x \geq x') &= \int_{-\infty}^{x'} \sum_{i=1}^{k} w_i \cdot \phi_i(x) \, \mathrm{d}x - \int_{x'}^{\infty} \sum_{i=1}^{k} w_i \cdot \phi_i(x) \, \mathrm{d}x \\
&= \sum_{i=1}^{k} w_i \cdot \left(\int_{-\infty}^{x'} \phi_i(x) \, \mathrm{d}x - \int_{x'}^{\infty} \phi_i(x) \, \mathrm{d}x \right) = \sum_{i=1}^{k} w_i \cdot \Phi_i(x')
\end{aligned}
$$

Basis function transformation (II)

Median gradient $g(x')$

$$p(x \leq x') - p(x \geq x') = \sum_{i=1}^{k} w_i \cdot \Phi_i(x')$$



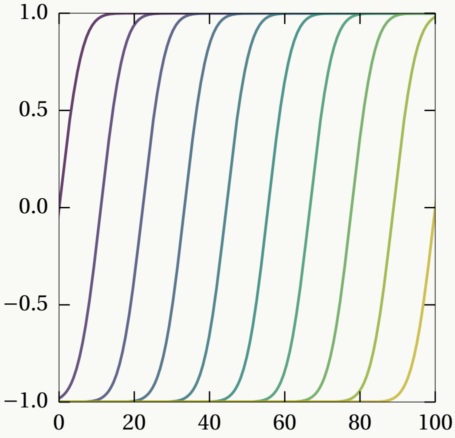
Basis functions $\Phi_i(x)$

Basis function transformation (II)

Median gradient $g(x')$

$$p(x \leq x') - p(x \geq x') = \sum_{i=1}^{k} w_i \cdot \Phi_i(x')$$

$\Rightarrow$ We can calculate the median gradient *without* changing the learned $\vec{w}$.



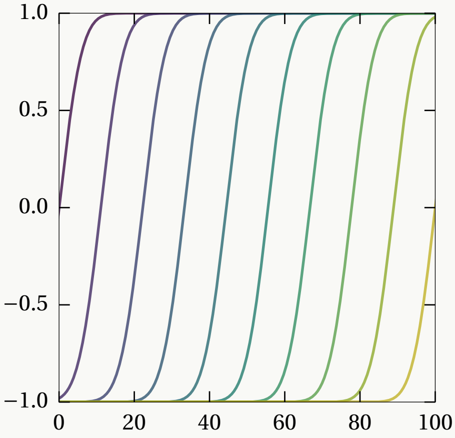
Basis functions $\Phi_i(x)$

Basis function transformation (II)

Median gradient $g(x')$

$$p(x \leq x') - p(x \geq x') = \sum_{i=1}^{k} w_i \cdot \Phi_i(x')$$

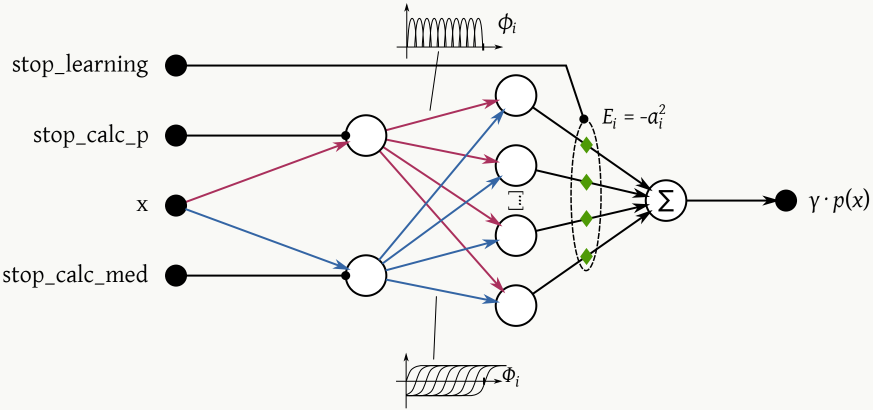
- $\Rightarrow$ We can calculate the median gradient *without* changing the learned $\vec{w}$.
- $\Rightarrow$ Just change the decoded function.



Basis functions $\Phi_i(x)$

Probability distribution representation and median estimation network

Implementation of $p(x \le x') - p(x \ge x') = \sum_{i=1}^{k} w_i \cdot \Phi_i(x)$ as a neural network



Inference

- *Goal:*
  Calculate median of the posterior distribution

$$P(t_{\text{total}} \mid t) \propto P(t \mid t_{\text{total}}) \cdot P(t_{\text{total}}) = \begin{cases} \frac{P(t_{\text{total}})}{t_{\text{total}}} & \text{if } t < t_{\text{total}} \\ 0 & \text{otherwise} \end{cases}$$

- *Basis functions $\Phi_i(x, y)$:*

$$\Phi_i(x, y) = \int_{-\infty}^{\hat{x}} p(y \mid x') \cdot \phi_i(x') \, \mathrm{d}x' - \int_{\hat{x}}^{\infty} p(y \mid x') \cdot \phi_i(x') \, \mathrm{d}x'$$

Wait! There is a problem!

Wait! There is a problem!

Learned $f(x) = w_i \cdot x$ is not defined over negative x!

Wait! There is a problem!

Learned $f(x) = w_i \cdot x$ is not defined over negative x!

- PES rule: $\Delta\vec{d_i} = \kappa \cdot E \cdot \vec{a}_i$

Wait! There is a problem!

Learned $f(x) = w_i \cdot x$ is not defined over negative x!

- PES rule: $\Delta\vec{d_i} = \kappa \cdot E \cdot \vec{a}_i$
- Basis functions $\phi_i(x)$ are non-negative
  $\Rightarrow$ activation $\vec{a}_i$ never represents negative values!

Wait! There is a problem!

Learned $f(x) = w_i \cdot x$ is not defined over negative x!

- PES rule: $\Delta \vec{d}_i = \kappa \cdot E \cdot \vec{a}_i$
- Basis functions $\phi_i(x)$ are non-negative
  $\Rightarrow$ activation $\vec{a}_i$ never represents negative values!
- *But:* basis functions $\Phi_i(x, y)$ are defined over $[-1, 1]$!

Wait! There is a problem!

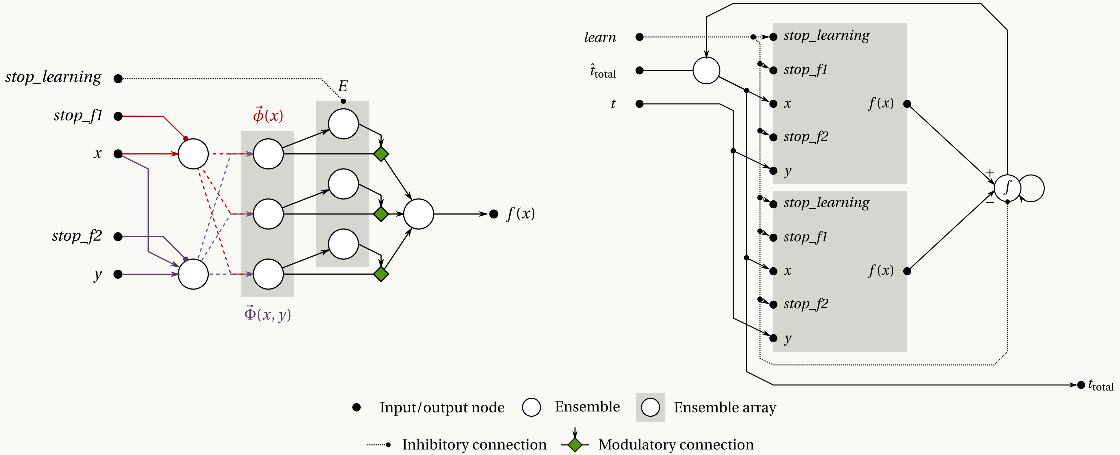
Learned $f(x) = w_i \cdot x$ is not defined over negative x!

- PES rule: $\Delta \vec{d}_i = \kappa \cdot E \cdot \vec{a}_i$
- Basis functions $\phi_i(x)$ are non-negative
  $\Rightarrow$ activation $\vec{a}_i$ never represents negative values!
- *But:* basis functions $\Phi_i(x, y)$ are defined over $[-1, 1]$!

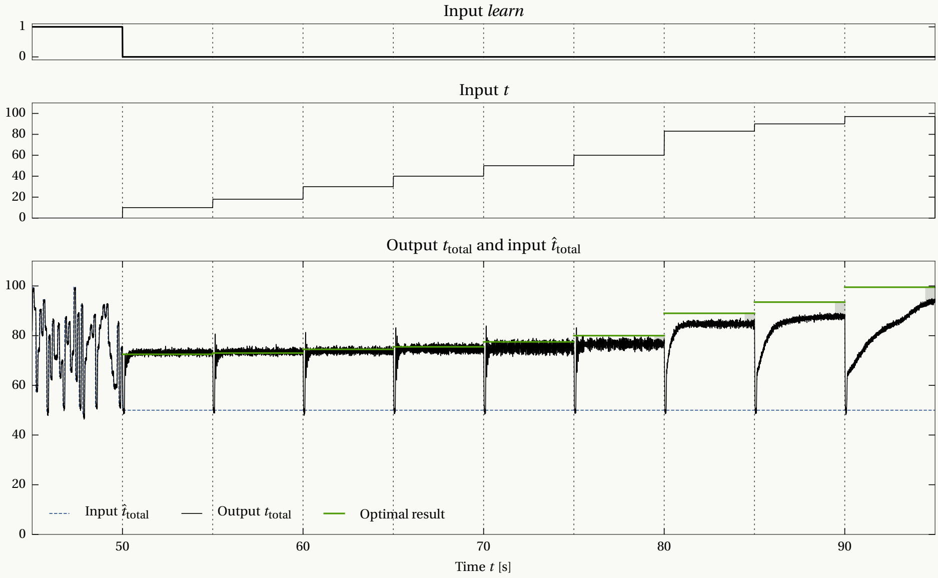
Solution

$\Rightarrow$ Two symmetric probability distribution networks representing the positive and negative branch of $\Phi(x, y)$

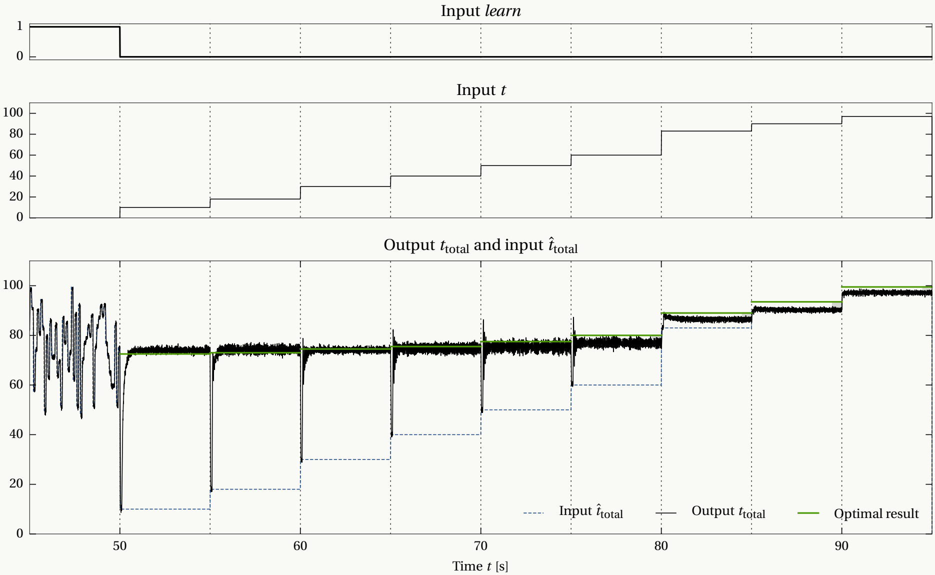
Lifespan Inference Network Overview



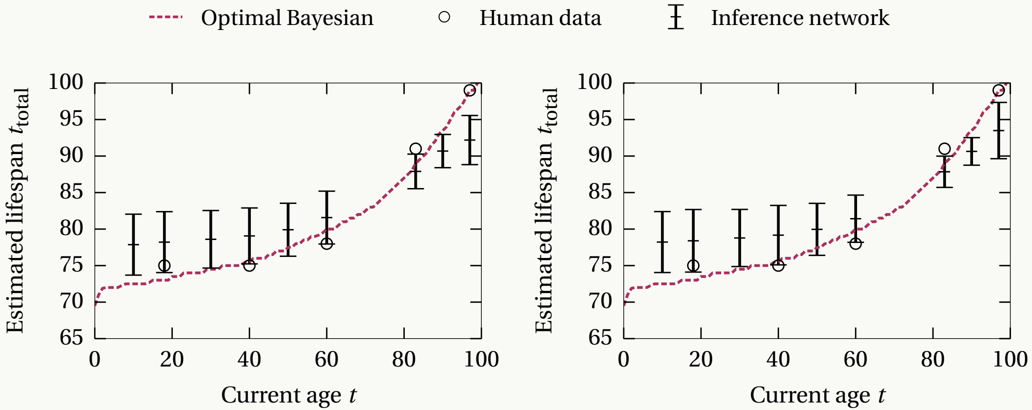
Lifespan Inference Results



Lifespan Inference Results



Lifespan Inference Results



(a) Constant $\hat{t}^{total}$

(b) $\hat{t}^{total} = t$

Conclusion

Results

- Network capable of learning the prior distribution $p(t_{\text{total}})$.
- Lifespan inference not perfect, but shows the right qualitative behaviour.

Questions

- Switching decoders by inhibiting populations an interesting computational principle?
- Statistical inference as gradient descent plausible?
- Derivation for $E_i = -a_i^2$ might not be correct (dynamics of the PES rule).
- Learning rule for linear functions which does not require symmetric populations?
- Some way to normalize weights, e.g. $\|\vec{w}\| = 1$?

Thank you for your attention!

Questions? Comments?