Bonsai.ML

Intelligent Experimental Control

Nicholas Guilbeault and Gonçalo Lopes and Joaquín Rapela

Gatsby Computational Neuroscience Unit
NeuroGEARS Ltd.

December 5, 2024

Outline

Outline

History of Bonsai.ML

- grant application
- developing tools
- who cares about Bonsai.ML tools? $\rightarrow$ more focus on dissemination
- real-time ML

Goals of Bonsai.ML

- allow non-programmers use ML tools,
- learn from non-programmers what ML tools are useful for them

Need for Real-Time and Reactive ML

Conventional ML operates on stored datasets. We need real-time ML that operates on infinite data stream with time-varying statistical properties.

If a sensor fails, our inferences need to continue. That is, we need reactive ML (e.g., rx.infer).

Bonsai.ML demos

Outline

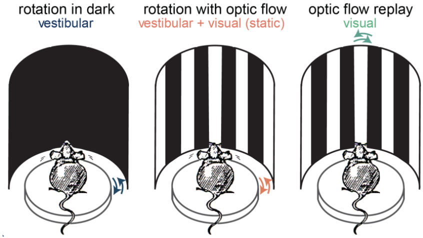
Linear Regression: Fundamental Concepts

- main concepts of linear regression
- Online Bayesian Linear Regression (can process infinite data streams, but assumes stationarity)
- Recursive least squares (can process infinite data streams, and does not assume stationarity)

Linear Regression: Practical

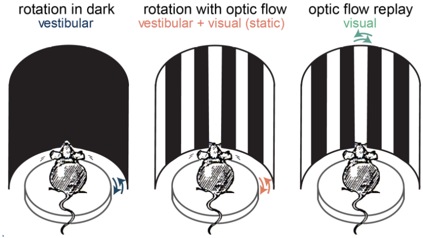
Estimation of receptive fields of visual cells from the Allen Institute.

Linear regression example

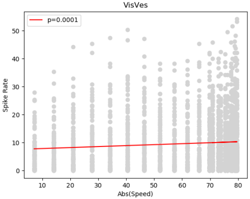


Keshavarzi et al., 2021

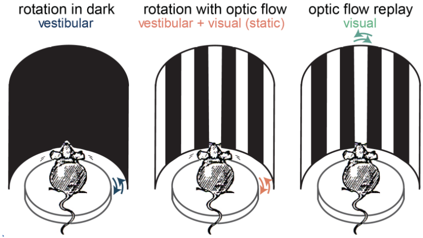
Linear regression example



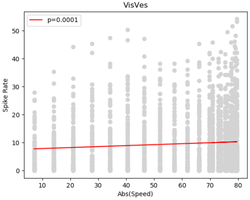
Keshavarzi et al., 2021



Linear regression example



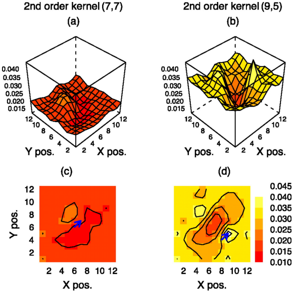
Keshavarzi et al., 2021



Is there a linear relation between the speed of rotation and the firing rate of visual cells?

Estimating nonlinear receptive fields from natural images

$$
\begin{aligned}
y(\mathbf{x}) &= k_0 + \sum_{i,j=1}^{N} k_1(i,j)x(i,j) && (1)\\
&+ \sum_{i_1,j_1,i_2,j_2=1}^{N} k_2(i_1,j_1,i_2,j_2)x(i_1,j_1)x(i_2,j_2) + \ldots\\
&+ \sum_{i_1,j_1,\ldots,i_Q,j_Q=1}^{N} k_Q(i_1,j_1,\ldots,i_Q,j_Q)x(i_1,j_1)\ldots x(i_Q,j_Q) + \varepsilon\\
y(\mathbf{x}) &= \mathbf{Aq}(\mathbf{x}) + \varepsilon && (2)
\end{aligned}
$$



Rapela et al., 2006.

Linear regression model

simple linear regression model

$$y(x_i, \mathbf{w}) = w_0 + w_1 x_i = [1, x_i] \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = [\phi_0(x_i), \phi_1(x_i)] \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$
$$= \phi(x_i)^\mathsf{T} \mathbf{w}$$

polynomial regression model

$$y(x_i, \mathbf{w}) = w_0 + w_1 x_i + w_2 x_i^2 + w_3 x_i^3 = [1, x_i, x_i^2, x_i^3] \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$
$$= [\phi_0(x_i), \phi_1(x_i), \phi_2(x_i), \phi_3(x_i)] \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \phi(x_i)^\mathsf{T} \mathbf{w}$$

basis functions linear regression model

$$y(x_i, \mathbf{w}) = \phi(x_i)^\mathsf{T} \mathbf{w} = \sum_{j=1}^{M} w_j \phi_j(x_i)$$

Linear regression model

$$
\mathbf{y}(\mathbf{x}, \mathbf{w}) = \begin{bmatrix} y(x_1, \mathbf{w}) \\ y(x_2, \mathbf{w}) \\ \cdots \\ y(x_N, \mathbf{w}) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_M(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_M(x_2) \\ \vdots & \vdots & \cdots & \vdots \\ \phi_1(x_N) & \phi_2(x_N) & \cdots & \phi_M(x_N) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}
$$

$$
= \mathbf{\Phi w}
$$

where $\mathbf{y}(\mathbf{x}, \mathbf{w}) \in \mathbb{R}^N, \mathbf{\Phi} \in \mathbb{R}^{N \times M}, \mathbf{w} \in \mathbb{R}^M$.

Notes

- We learned how to build a linear regression model.
- But, how can we learn the model parameters w?

Least-squares estimation of model parameters (Trefethen and Bau III, 1997)

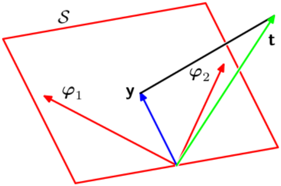
Definition 1 (Least-squares problem)

Given $\mathbf{\Phi} \in \mathbb{R}^{N \times M}, N \geq M, \mathbf{t} \in \mathbb{R}^N$, *find* $\mathbf{w} \in \mathbb{R}^M$ *such that* $E_{LS}(\mathbf{w}) = ||\mathbf{t} - \mathbf{\Phi w}||_2$ *is minimised.*

Theorem 1 (Least-squares solution)

Let $\mathbf{\Phi} \in \mathbb{R}^{N \times M} (N \geq M)$ *and* $\mathbf{t} \in \mathbb{R}^N$ *be given. A vector* $\mathbf{w} \in \mathbb{R}^M$ *minimises* $||\mathbf{r}||_2 = ||\mathbf{t} - \mathbf{\Phi w}||_2$, *thereby solving the least-squares problem, if and only if* $\mathbf{r} \perp \mathit{range}(\mathbf{\Phi})$, *that is,* $\mathbf{\Phi}^\intercal \mathbf{r} = 0$, *or equivalently,* $\mathbf{\Phi}^\intercal \mathbf{\Phi w} = \mathbf{\Phi}^\intercal \mathbf{t}$, *or again equivalently,* $P\mathbf{t} = \mathbf{\Phi w}$, *where* $P \in \mathbf{R}^{N \times N}$ *is the orthogonal projector onto* $\mathit{range}(A)$ *(i.e.,* $P = A\,(A^\intercal A)^{-1} A^\intercal$*).*

Figure 3.2 Geometrical interpretation of the least-squares solution, in an N-dimensional space whose axes are the values of $t_1, \ldots, t_N$. The least-squares regression function is obtained by finding the orthogonal projection of the data vector $\mathbf{t}$ onto the subspace spanned by the basis functions $\phi_j(\mathbf{x})$ in which each basis function is viewed as a vector $\boldsymbol{\varphi}_j$ of length N with elements $\phi_j(\mathbf{x}_n)$.



Bishop (2016)

Instruction to run notebooks in Google Colab

1. open a notebook from here
2. replace **github.com** by **githubtocolab.com** in the URL
3. insert a cell at the beginning of the notebook with the following content

```
!git clone https://github.com/joacorapela/gcnuBridging2023.git
%cd gcnuBridging2023
!pip install -e .
```

4. from the menu **Runtime** select **Run all**.

Code for least-squares estimation of model parameters

- overfitting
- cross validation
- larger datasets allow more complex models

Notes

- We learned how to estimate the parameters of a linear regression models by least squares.
- But, how to avoid overfitting in the estimation?

Regularised least-squares estimation of model parameters

To cope with the overfitting of least squares, we can add to the least squares optimisation criterion a term that enforces coefficients to be zero. The regularised least-squares optimisation criterion becomes:

$$E_{RLS}(\mathbf{w}) = ||\mathbf{t} - \mathbf{\Phi}\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_2^2$$

where λ is the regularisation parameter that weights the strength of the regularisation.

Regularised least-squares estimation of model parameters

Claim 1 (Regularised least-squares estimate)

$$\mathbf{w}_{RLS} = \arg\min_{\mathbf{w}} E_{RLS}(\mathbf{w}) = \arg\min_{\mathbf{w}} ||\mathbf{t} - \mathbf{\Phi}\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_2^2 = (\lambda\mathbf{I} + \mathbf{\Phi}^\intercal\mathbf{\Phi})^{-1}\mathbf{\Phi}^\intercal\mathbf{t}$$

Proof.

Since $E_{RLS}(\mathbf{w})$ is a polynomial of order two on the elements of $\mathbf{w}$ (i.e., a quadratic form), we can use the *Completing the Squares* technique below to find its minimum.

$$\begin{aligned}
\boldsymbol{\mu} &= \arg\max_{\mathbf{w}} \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \Sigma) = \arg\max_{\mathbf{w}} \log \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \Sigma) \\
&= \arg\max_{\mathbf{w}} \{K - \frac{1}{2}(-2\boldsymbol{\mu}^\intercal\Sigma^{-1}\mathbf{w} + \mathbf{w}\Sigma^{-1}\mathbf{w})\} && (3) \\
&= \arg\min_{\mathbf{w}} \{-K + \frac{1}{2}(-2\boldsymbol{\mu}^\intercal\Sigma^{-1}\mathbf{w} + \mathbf{w}\Sigma^{-1}\mathbf{w})\} \\
&= \arg\min_{\mathbf{w}} \{K_1 - 2\boldsymbol{\mu}^\intercal\Sigma^{-1}\mathbf{w} + \mathbf{w}\Sigma^{-1}\mathbf{w}\} && (4)
\end{aligned}$$

To find the minimum of a quadratic form, we write it in the form of the terms inside the curly brackets of Eq. 4, and the term corresponding to $\boldsymbol{\mu}$ will be the minimum.

Regularised least-squares estimation of model parameters

Proof.

Let's write E_{RLS} in the form of the terms inside the curly brackets of Eq. 4.

$$\begin{aligned} E_{RLS} &= ||\mathbf{t} - \mathbf{\Phi w}||_2^2 + \lambda||\mathbf{w}||_2^2 = (\mathbf{t} - \mathbf{\Phi w})^{\mathsf{T}}(\mathbf{t} - \mathbf{\Phi w}) + \lambda \mathbf{w}^{\mathsf{T}}\mathbf{w} \\ &= \mathbf{t}^{\mathsf{T}}\mathbf{t} - 2\mathbf{t}^{\mathsf{T}}\mathbf{\Phi w} + \mathbf{w}^{\mathsf{T}}\mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi w} + \lambda \mathbf{w}^{\mathsf{T}}\mathbf{w} \\ &= \mathbf{t}^{\mathsf{T}}\mathbf{t} - 2\mathbf{t}^{\mathsf{T}}\mathbf{\Phi w} + \mathbf{w}^{\mathsf{T}}(\mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi} + \lambda \mathbf{I}_M)\mathbf{w} \end{aligned}$$

Calling

$$\begin{aligned} \Sigma^{-1} &= \mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi} + \lambda \mathbf{I}_M \\ \boldsymbol{\mu}^{\mathsf{T}}\Sigma^{-1} &= \mathbf{t}^{\mathsf{T}}\mathbf{\Phi} \text{ or } \boldsymbol{\mu}^{\mathsf{T}} = \mathbf{t}^{\mathsf{T}}\mathbf{\Phi}\Sigma \text{ or } \boldsymbol{\mu} = \Sigma\mathbf{\Phi}^{\mathsf{T}}\mathbf{t} = \left(\mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi} + \lambda \mathbf{I}_M\right)^{-1}\mathbf{\Phi}^{\mathsf{T}}\mathbf{t} \end{aligned}$$

we can express

$$E_{RLS} = K + 2\boldsymbol{\mu}^{\mathsf{T}}\Sigma^{-1}\mathbf{w} + \mathbf{w}\Sigma^{-1}\mathbf{w}$$

Then

$$\mathbf{w}_{RLS} = \arg\min_{\mathbf{w}} E_{RLS}(\mathbf{w}) = \boldsymbol{\mu} = \left(\mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi} + \lambda \mathbf{I}_M\right)^{-1}\mathbf{\Phi}^{\mathsf{T}}\mathbf{t}$$

□

Code for regularised least-squares estimation of model parameters

- control of overfitting

Notes

- So far we have assummed deterministic parameters. But they are random quantities.
- How can we build linear regression models for random parameters?

Maximum-likelihood estimation of model parameters

Definition 2 (Likelihood function)

For a statistical model characterised by a probability density function $p(\mathbf{x}|\theta)$ *(or probability mass function* $P_\theta(X = \mathbf{x})$*) the likelihood function is a function of the parameters* θ*,* $\mathcal{L}(\theta) = p(\mathbf{x}|\theta)$ *(or* $\mathcal{L}(\theta) = P_\theta(\mathbf{x})$*).*

Definition 3 (Maximum likelihood parameters estimates)

The maximum likelihood parameters estimates are the parameters that maximise the likelihood function.

$$\theta_{ML} = \arg\max_{\theta} \mathcal{L}(\theta)$$

Maximum-likelihood estimation for the basis function linear regression model

We seek the parameter $\mathbf{w}_{ML}$ and β_{ML} that maximised the following likelihood function

$$\mathcal{L}(\mathbf{w}, \beta) = p(\mathbf{t}|\mathbf{w}, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{\Phi}\mathbf{w}, \beta^{-1} I_N) \tag{5}$$

They are

$$\mathbf{w}_{ML} = (\mathbf{\Phi}^\intercal \mathbf{\Phi})^{-1} \mathbf{\Phi}^\intercal \mathbf{t} \tag{6}$$

$$\frac{1}{\beta_{ML}} = \frac{1}{N} ||\mathbf{t} - \mathbf{\Phi}\mathbf{w}_{ML}||_2^2 \tag{7}$$

- first regression method that assumes random observations
- if the likelihood function is assumed to be Normal, maximum-likelihood and least-squares coefficients estimates are equal.

Maximum likelihood: exercise

Exercise 1

Derive the formulas for the maximum likelihood estimates of the coefficients, $\mathbf{w}$*, and noise precision,* β*, of the basis functions linear regression model given in Eqs. 6 and 7.*

Solution.

$$
\begin{aligned}
\mathcal{L}(\mathbf{w}, \beta) &= p(\mathbf{t}|\mathbf{w}, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{\Phi w}, \beta^{-1}\mathbf{I}) \\
&= \frac{1}{(2\pi)^{\frac{N}{2}} |\beta^{-1}\mathbf{I}|^{\frac{1}{2}}} \exp\left(-\frac{\beta}{2}||\mathbf{t} - \mathbf{\Phi w}||_2^2\right) \\
\log \mathcal{L}(\mathbf{w}, \beta) &= -\frac{N}{2}\log 2\pi + \frac{N}{2}\log\beta - \frac{\beta}{2}||\mathbf{t} - \mathbf{\Phi w}||_2^2 \\
\mathbf{w}_{ML} &= \arg\max_{\mathbf{w}} \log \mathcal{L}(\mathbf{w}, \beta) = \arg\min_{\mathbf{w}} ||\mathbf{t} - \mathbf{\Phi w}||_2^2 = (\mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathsf{T}}\mathbf{t} \\
\frac{\partial}{\partial\beta}\log p(\mathbf{t}|\mathbf{w}_{ML}, \beta) &= \frac{N}{2}\frac{1}{\beta} - \frac{1}{2}||\mathbf{t} - \mathbf{\Phi w}_{ML}||_2^2 \\
\frac{\partial}{\partial\beta}\log p(\mathbf{t}|\mathbf{w}_{ML}, \beta_{ML}) &= 0 \quad \text{iff} \quad \frac{1}{\beta_{ML}} = \frac{1}{N}||\mathbf{t} - \mathbf{\Phi w}_{ML}||_2^2
\end{aligned}
$$

Notes

- We have learned how to estimate random parameters in linear regression models.
- How can we incorporate prior assumptions in this estimation?

Batch Bayesian linear regression: posterior distribution of parameters

In Bayesian linear regression we seek the posterior distribution of the weights of the linear regression model, $\mathbf{w}$, given the observations, which is proportional to the product of the likelihood function, $p(\mathbf{t}|\mathbf{w})$, and the prior, $p(\mathbf{w})$; i.e.,

$$p(\mathbf{w}|\mathbf{t}) \propto p(\mathbf{t}|\mathbf{w})p(\mathbf{w}) \tag{8}$$

To calculate this posterior below we use the likelihood function defined in Eq. 5 and the following prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

Using the expression of the conditional of the Linear Gaussian model we obtain

$$\begin{aligned} p(\mathbf{w}|\mathbf{t}) &= \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \\ \mathbf{m}_N &= \beta \mathbf{S}_N \boldsymbol{\Phi}^\intercal \mathbf{t} && (9) \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^\intercal \boldsymbol{\Phi} && (10) \end{aligned}$$

Batch Bayesian linear regression: exercise

Exercise 2

Derive the formulas for the Bayesian posterior mean (Eq. 9) and covariance (Eq. 10) of the basis function linear regression model.

Exercise 3

Show that

$$\log p(\mathbf{w}|\boldsymbol{t}) = K - \frac{\beta}{2}||\mathbf{t} - \boldsymbol{\Phi}\mathbf{w}||_2^2 - \frac{\alpha}{2}||\mathbf{w}||_2^2 \tag{11}$$

Therefore, the maximum-a-posteriori parameters of the basis function linear regression model are the solution of the regularised least-squares problem with $\lambda = \alpha/\beta$. Note that, as we will show next, Bayesian linear regression uses the full posterior of the parameters to make predictions or to do model selection, and not just the maximum-a-posteriori parameters.

Batch Bayesian linear regression: demo code

Available here

Notes

- Now we know how to do batch Bayesian linear regression.
- However, in Bonsai we don't want to work with batch data. We want to do online processing of infinite data streams. How can we do this?

Recursive update of posterior distribution of the parameters for conditionally independent observations

Claim 2 (recursive update)

If the observations, $\{\mathbf{t}_1, \ldots, \mathbf{t}_n, \ldots\}$, are linearly independent when conditioned on the model parameters, $\boldsymbol{\theta}$, then for any $n \in \mathbb{N}$

$$p(\boldsymbol{\theta}|\mathbf{t}_1, \ldots, \mathbf{t}_n) = K\ p(\mathbf{t}_n|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{t}_1, \ldots, \mathbf{t}_{n-1}) \tag{12}$$

where K is a quantity that does not depend on $\boldsymbol{\theta}$.

Recursive update of posterior distribution of the parameters for conditionally independent observations

Proof.

By induction on $H_n : p(\boldsymbol{\theta}|\mathbf{t}_1, \ldots, \mathbf{t}_n) = K \; p(\mathbf{t}_n|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{t}_1, \ldots, \mathbf{t}_{n-1})$.

H_1

$$p(\boldsymbol{\theta}|\mathbf{t}_1) = \frac{p(\boldsymbol{\theta}, \mathbf{t}_1)}{p(\mathbf{t}_1)} = \frac{p(\mathbf{t}_1|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{t}_1)} = K \; p(\mathbf{t}_1|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

$H_n \rightarrow H_{n+1}$

$$\begin{aligned}
p(\boldsymbol{\theta}|\mathbf{t}_1, \ldots, \mathbf{t}_{n+1}) &= \frac{p(\boldsymbol{\theta}, \mathbf{t}_1, \ldots, \mathbf{t}_{n+1})}{p(\mathbf{t}_1, \ldots, \mathbf{t}_{n+1})} \\
&= \frac{p(\mathbf{t}_{n+1}|\boldsymbol{\theta}, \mathbf{t}_1, \ldots, \mathbf{t}_n)p(\boldsymbol{\theta}, \mathbf{t}_1, \ldots, \mathbf{t}_n)}{p(\mathbf{t}_1 \ldots, \mathbf{t}_{n+1})} \\
&= \frac{p(\mathbf{t}_{n+1}|\boldsymbol{\theta})p(\boldsymbol{\theta}, \mathbf{t}_1, \ldots, \mathbf{t}_n)}{p(\mathbf{t}_1 \ldots, \mathbf{t}_{n+1})} \\
&= \frac{p(\mathbf{t}_{n+1}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{t}_1, \ldots, \mathbf{t}_n)p(\mathbf{t}_1, \ldots, \mathbf{t}_n)}{p(\mathbf{t}_1 \ldots, \mathbf{t}_{n+1})} \\
&= K \; p(\mathbf{t}_{n+1}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{t}_1, \ldots, \mathbf{t}_n)
\end{aligned}$$

Note: the third equality above holds because the observations are assumed to be conditional independent given the parameters.

$\square$

Recursive update of the posterior distribution of the parameters for a conjugate prior

Claim 3

If

$$P(\mathbf{w}|\mathbf{t}_1, \ldots, \mathbf{t}_n) = \mathcal{N}(\mathbf{w}|\mathbf{m}_n, \mathbf{S}_n) \quad (13)$$

$$P(\mathbf{t}_{n+1}|\mathbf{w}) = \mathcal{N}(\mathbf{t}_{n+1}|\mathbf{\Phi}\mathbf{w}, \beta^{-1}\mathbf{I}) \quad (14)$$

then

$$P(\mathbf{w}|\mathbf{t}_1, \ldots, \mathbf{t}_{n+1}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_{n+1}, \mathbf{S}_{n+1})$$

with

$$S_{n+1} = S_n - (\beta^{-1} + \phi(\mathbf{x}_{n+1})^\intercal S_n \phi(\mathbf{x}_{n+1}))^{-1} S_n \phi(\mathbf{x}_{n+1}) \phi(\mathbf{x}_{n+1})^\intercal S_n \quad (15)$$

$$\begin{aligned} m_{n+1} &= \beta t_{n+1} S_{n+1} \phi(\mathbf{x}_{n+1}) + \mathbf{m}_n - \\ &\quad (\beta^{-1} + \phi(\mathbf{x}_{n+1})^\intercal S_n \phi(\mathbf{x}_{n+1}))^{-1} \phi(x_{n+1})^\intercal \mathbf{m}_n S_n \phi(x_{n+1}) \end{aligned} \quad (16)$$

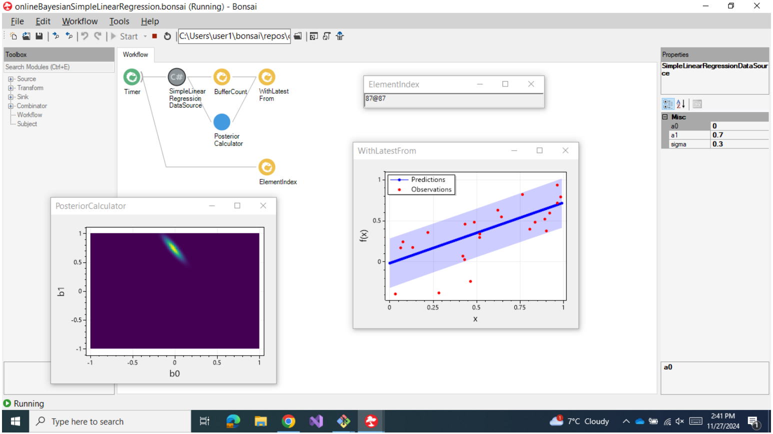
Python code for online Bayesian linear regression

Available here.

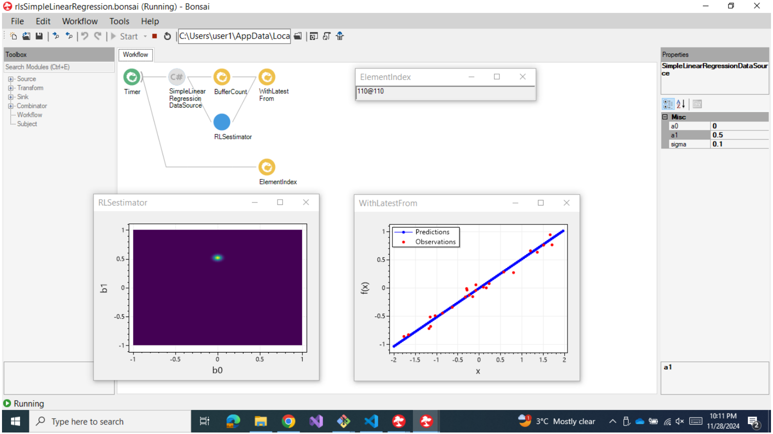
Notes

- We have learned how to do linear regression in Python.
- However, we are in the first Bonsai conference. Let's do online Bayesian linear regression in Bonsai.

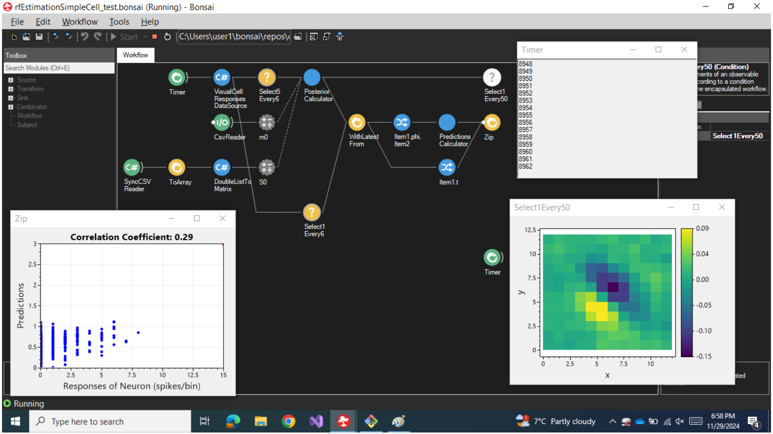
Online Bayesian linear regression in Bonsai



Recursive least squares in Bonsai



Estimating receptive fields of cortical visual neurons in Bonsai



References

Bishop, C. M. (2016). *Pattern recognition and machine learning*. Springer-Verlag New York.

Trefethen, L. n. and Bau III, D. (1997). Numerical linear algebra.