Exercise 2.1: Speed of convergence in Predictive Coding

In the lecture, we learned, how we can (approximately) invert models by performing a gradient ascent on a function F.[1] Here we will have a closer look on the convergence behaviour.

We use the same model as in the lecture, so we have

$$p(x) = \frac{1}{\sqrt{2\pi\Sigma_{pr}}}\,\mathrm{e}^{-\frac{(x-\mu_{pr})^2}{2\Sigma_{pr}}} \tag{1}$$

$$p(y|x) = \frac{1}{\sqrt{2\pi\Sigma_{gen}}}\,\mathrm{e}^{-\frac{(x^2-y)^2}{2\Sigma_{gen}}} \tag{2}$$

for prior $p(x)$ and likelihood $p(y|x)$.

To enable concrete calculations, we set $\Sigma_{pr} = 1$, $\Sigma_{gen} = 1$, $\mu_{pr} = 3$. In addition, we assume that we observe $y = 2$.

We want to approximate our posterior $p(x|y)$ by a delta-distribution $q_\phi(x) = \delta(x - \phi)$. We could do that for every ϕ, but we want to find the value ϕ_0, where the delta-distribution is centred around the maximum of the posterior. In the lecture, we saw that this is equivalent[2] to finding the argmax of

$$F(x) = \ln(p(y|x)) + \ln(p(x)). \tag{3}$$

(a) Plot posterior and $F(x)$ for $x \in [0, 4]$. Is the posterior normal? *(2 points)*

Our $F(x)$ is rather a $F(\phi)$. Because of our particular approximation, x and the parameter ϕ *live* on the same scale, but for the rest of the exercise we will talk about $F(\phi)$ to remind us that we are looking for ϕ_0, the parameter giving us the *best* approximation $q_{\phi_0}(x)$ of the posterior $p(x|y)$.

(b) Determine ϕ_0 (up to a small error) by performing the direct gradient ascent $\phi(\tau + \Delta\tau) = \phi(\tau) + \Delta\tau \cdot \frac{\mathrm{d}F}{\mathrm{d}\phi}$ on $F(\phi)$. Start at μ_{pr} and plot $\phi(\tau)$ against τ for some reasonable time interval. *(2 points) Hint: You can use a step size of* $\Delta\tau = 0.01$.

The (direct) gradient ascent allowed us to find ϕ_0, but as we also argued in the lecture, the difference/differential equation, we integrate in (b) might be too complex to be implemented neurally.

[1] Again, we follow Bogacz, 2017.

[2] Note that F is only equal to the log-joint for this particular choice of $q_\phi(x)$. If we would approximate the posterior by a different function class, F would have a different form.

To simplify the equation, we introduced the *prediction-error*-quantities $\tilde{\epsilon}_{gen}$ and $\tilde{\epsilon}_{pr}$. To compute them, we need two additional differential equations, so that we have now three (coupled) differential equations (but with an easier structure):

$$\dot{\phi} = \epsilon_{gen} \cdot g'(\phi) + \epsilon_{pr} \tag{4}$$

$$\dot{\epsilon}_{gen} = y - g(\phi) - \Sigma_{gen} \cdot \epsilon_{gen} \tag{5}$$

$$\dot{\epsilon}_{pr} = \mu_{pr} - \phi - \Sigma_{pr} \cdot \epsilon_{pr} \tag{6}$$

(c) Determine ϕ_0 (up to a small error) by discretizing and then integrating equations 4, 5 and 6. Use the same step size as for (b) and plot the evolution of ϕ, ϵ_{pr} and ϵ_{gen} against τ. *(5 points)*

(d) Which of the two methods used in (b) and (c) converges faster? Why? *(2 points)*

Exercise 2.2: The HGF toolbox and experimental design

This exercise is intended to get you started with the HGF toolbox and understand the relation between the generative model and the perceptual model (i.e., inverted generative model) better.

Note: In the lecture, we looked at the three level HGF for binary outcomes with perceptual parameters κ, ω, and ϑ. However, in recent versions of the toolbox, the parameters κ and ω exist on all levels. On the first level, κ_1 can be set to 1 to obtain the original model, and ω_1 is undefined. On the highest level, κ_n is undefined, and ω_n corresponds to our previous ϑ. In symmetry to lower levels however, ω_3 describes the log-variance of the Gaussian. Therefore:

$$\vartheta = \exp(\omega_3) \tag{7}$$

in the HGF with 3 levels. In this exercise sheet, we will always provide both values to avoid confusion.

(a) Download the latest version of the HGF toolbox as part of the TAPAS software package: *https://www.tnu.ethz.ch/en/software/tapas.html*. Read the manual. Add the HGF folder to your Matlab path. Go through the **`hgf_demo`** script (either as a LiveScript, if your Matlab version allows, or as regular code). What are the two main functions of the toolbox and what do they do? *(1 point)*

(b) The toolbox allows us to simulate beliefs μ and behavior y in response to sensory inputs, using the perceptual model of the HGF (and an appropriate response model). But first, we take a step back and generate sensory inputs u using the HGF. Write a function that implements the generative model for the three level HGF for binary outcomes and generates binary events u according to it. *Hint:* These equations were given in the lecture and are not implemented in the toolbox.

To start, choose the following parameter setting:

$$\kappa_2 = 1, \omega_2 = -4, \vartheta = \exp(\omega_3) = \exp(-6). \tag{8}$$

Plot the resulting traces for x_3, x_2, and the generated inputs u. Repeat this procedure a few times and examine the typical traces you get. Also try out different parameter settings, in particular, higher values for ϑ. *(4 points)*

(c) Decide on one input sequence that you generated, simulate beliefs and responses using the `tapas_hgf_binary` model with the same (or other) parameters that you used during input generation. Does your simulated agent correctly track the evolution of x_3 and x_2? *(3 points)*

(d) Why, do you think, would it be a good or bad idea to use the generative model of the HGF to generate stimulus sequences to use in an experiment? If you think it's a bad idea, why could it still be a useful model for the agent to invert during perception? *(3 points)*

Exercise 2.3: Coordinate choice and parameter identifiability in the HGF

In this exercise, we take a closer look at the meaning of the perceptual parameters of the three level HGF for binary outcomes and their relationships. The observations we will make, however, generalize to any HGF with n levels and both categorical as well as continuous outcomes.

(a) Simulate beliefs and responses using the `tapas_hgf_binary` perceptual model with parameters:

$$\begin{aligned} \mu_3^{(0)} &= 1, \sigma_3^{(0)} = 1, \\ \kappa_2 &= 2.5, \omega_2 = -4, \\ \vartheta &= \exp(\omega_3) = \exp(-6) = 0.0025, \end{aligned} \tag{9}$$

and the `tapas_unitsq_sgm` response model with parameter:

$$\zeta = 5. \tag{10}$$

Try to recover these parameters using `tapas_fitModel`, estimating ζ, $\mu_3^{(0)}$, κ_2, and $\vartheta = \exp(\omega_3)$. Inspect their posterior correlation using `tapas_fit_plotCorr`. Try again, this time estimating ζ, $\mu_3^{(0)}$, ω_2, and $\vartheta = \exp(\omega_3)$, and inspect the posterior correlation. What do you observe? *(3 points)*

(b) Again, simulate beliefs and responses using the `tapas_hgf_binary` perceptual model, now with parameters:

$$\begin{gathered} \mu_3^{(0)} = 2.5, \sigma_3^{(0)} = 6.25, \\ \kappa_2 = 1, \omega_2 = -4, \\ \vartheta = \exp(\omega_3) = \exp(-4.1674) = 0.0155, \end{gathered} \tag{11}$$

and the `tapas_unitsq_sgm` response model with parameter:

$$\zeta = 5. \tag{12}$$

Compare the belief trajectories with the ones you simulated in (a). What do you find? Can you explain this effect? *(3 points)*

(c) By setting κ_2 to 1 and/or ω_2 to 0, we can make these parameters seemingly disappear from the model. Try to write down a rule for how you would have to change the other parameters such that only μ_3 is affected by this change, but the belief trajectories on the lower levels stay the same. What does that tell you? *(3 points)*

(d) We can actually get rid of the indeterminacy observed in (b) and (c) by including a readout of μ_3 in our response model. This way, we can estimate all three, $\mu_3^{(0)}$, κ_2, and ω_2, given observed responses. Try this by changing the response model to the `tapas_unitsq_sgm_mu3`, where μ_3 determines the trial-by-trial decision temperature:

$$p(y = 1|\hat{\mu}_1, \mu_3) = \frac{\hat{\mu}_1^{\exp(-\mu_3)}}{\hat{\mu}_1^{\exp(-\mu_3)} + (1 - \hat{\mu}_1)^{\exp(-\mu_3)}} \tag{13}$$

This means that the agent will behave less deterministically the more volatile he/she believes its environment to be. Repeat the exercise of (a) with this response model and look at how the posterior correlations among the parameters change. *(3 points)*

Note: You will have to create two new files within the HGF toolbox to be able to do this:

1. `tapas_unitsq_sgm_mu3_namep`, which will be a dummy function as there are no free parameters in this models. Use the function `tapas_unitsq_sgm_namep` as a template and simply return an empty struct.
2. `tapas_unitsq_sgm_mu3_sim`, which is used to simulate responses according to this model. Use `tapas_unitsq_sgm_sim` as a template and replace `ze` with exp(`mu3`). You will get the trial-by-trial values of `mu3` from the variable `infStates(:,3,3)`.

Announcements:

- Put your solutions on moodle before the exercise session on April 16.
- Solutions to the exercises on this sheet will be presented in that exercise session.

References

[1] Rafal Bogacz. "A tutorial on the free-energy framework for modelling perception and learning". In: *Journal of Mathematical Psychology* 76 (2017), pp. 198–211. ISSN: 10960880. DOI: `10.1016/j.jmp.2015.11.003`. URL: `http://dx.doi.org/10.1016/j.jmp.2015.11.003`.