Message Passing in the Hierarchical Gaussian Filter

June 27, 2018

1 Computations of Nodes in the HGF

In the HGF, each node has to perform several computations within an experimental trial, which can be ordered in time as shown in the box:

NODE i AT TRIAL k

$\leftarrow$ `receive` $\mathrm{PE}_{i-1}^{(k)}$ from node_{i-1}

UPDATE step
  `compute` $\mathrm{posterior}_i^{(k)}$
  given: $\mathrm{PE}_{i-1}^{(k)}$ and $\mathrm{prediction}_i^{(k)}$
$\rightarrow$ `send` $\mathrm{posterior}_i^{(k)}$ to node_{i-1}

PE step
  `compute` $\mathrm{PE}_i^{(k)}$
  given: $\mathrm{prediction}_i^{(k)}$ and $\mathrm{posterior}_i^{(k)}$
$\rightarrow$ `send` $\mathrm{PE}_i^{(k)}$ to node_{i+1}
$\leftarrow$ `receive` $\mathrm{posterior}_{i+1}^{(k)}$ from node_{i+1}

PREDICTION step
  `compute` $\mathrm{prediction}_i^{(k+1)}$
  given: $\mathrm{posterior}_i^{(k)}$ and $\mathrm{posterior}_{i+1}^{(k)}$

The exact computations in each step depend on the nature of the coupling (VAPE vs. VOPE) with the parent and children nodes. Critically, it is not

only the PEs that are sent upwards, but depending on the coupling also other quantities need to signalled.

2 Computations for VAPE coupling

The exact computations of the UPDATE step depend on the nature of the coupling with the child node(s), while both the PE step and the PREDICTION step depend on the coupling with the parent node(s).

2.1 Update Step

If Node i is the value parent of Node $i-1$, then the following update equations apply to Node i:

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)} \tag{1}$$

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)} + \hat{\pi}_i^{(k)}} \delta_{i-1}^{(k)} \tag{2}$$

We note here that we can let the update of the precision happen first, and therefore use it for the update of the mean:

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)} \tag{3}$$

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \tag{4}$$

In sum, at the time of the update, Node i needs to have access to the following quantities:

Its own predictions: $\hat{\mu}_i^{(k)}$, $\hat{\pi}_i^{(k)}$

Coupling strength: $\alpha_{i-1,i}$

From level below: $\delta_{i-1}^{(k)}$, $\hat{\pi}_{i-1}^{(k)}$

All of these are available at the time of the update. Node i therefore only needs to receive the PE and the predicted precision from the level below to perform its update.

2.2 Prediction Error Step

We will assume in the following, that Node i is the value child of Node $i+1$. Then the following quantities have to be sent up to Node $i+1$ (cf. necessary information from level below in a value parent):

Predicted precision: $\hat{\pi}_i^{(k)}$

Prediction error: $\delta_i^{(k)}$

Node i has already performed the PREDICTION step on the previous trial, so it has already computed the predicted precision of the current trial, $\hat{\pi}_i^{(k)}$. Hence, in the PE step, it needs to perform only the following calculation:

$$\delta_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)} \tag{5}$$

2.3 Prediction Step

Still assuming that Node i is the value child of Node $i+1$, the PREDICTION step consists of the following computations:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} + \alpha_{i,i+1}\mu_{i+1}^{(k)} \tag{6}$$

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \nu_i^{(k+1)}} \tag{7}$$

with

$$\nu_i^{(k+1)} = \exp(\omega_i)).$$

Note that if Node i additionally has a VOPE parent node, the estimated volatility $\nu_i^{(k+1)}$ that enters the precision update would also depend on the posterior mean of that volatility parent (cf. PREDICTION step for VOPE coupling).

In general, the prediction of the mean will depend only on whether Node i has a value parent or not, whereas the prediction of the precision only depends on whether Node i has a volatility parent or not.

However, without a volatility parent node, the estimated volatility only depends on the node's own learning rate ω_i, i.e., it is a constant. We can therefore simply write:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} + \alpha_{i,i+1}\mu_{i+1}^{(k)} \tag{8}$$

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \exp(\omega_i)} \tag{9}$$

Thus, the PREDICTION step only depends on knowing the node's own posteriors and receiving the value parent's posterior in time before the new input arrives.

In general, the prediction of the mean will depend only on whether Node i has a value parent or not, whereas the prediction of the precision only depends on whether Node i has a volatility parent or not.

The PREDICTION step could also be placed in the beginning of a trial. However, we usually think about the beginning of a trial as starting with receiving a new input, and of a prediction as being present before that input is received. It would probably make more sense to think of the prediction as happening (continuously) between trials. For implementational purposes, it is important to keep in mind that the posterior means of parent nodes have to be sent back to their children once they're computed, such that children nodes can prepare for the next input in time.

3 Computations for VOPE coupling

As in the case of VAPE coupling, the exact computations of the UPDATE step depend on the nature of the coupling with the child node(s), while both the PE step and the PREDICTION step depend on the coupling with the parent node(s).

3.1 Update Step

If Node i is the volatility parent of Node $i-1$, then the following update equations apply to Node i:

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \frac{1}{2}(\kappa_{i-1}\nu_{i-1}^{(k)}\hat{\pi}_{i-1}^{(k)})^2 * (1 + (1 - \frac{1}{\pi_{i-1}^{(k-1)}\nu_{i-1}^{(k)}})\delta_{i-1}^{(k)}) \tag{10}$$

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{1}{2}\kappa_{i-1}\nu_{i-1}^{(k)}\frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}}\delta_{i-1}^{(k)} \tag{11}$$

Therefore, at the time of the update, Node i needs to have access to the following quantities:

Its own predictions: $\hat{\mu}_i^{(k)}$, $\hat{\pi}_i^{(k)}$

Coupling strength: κ_{i-1}

From level below: $\delta_{i-1}^{(k)}$, $\hat{\pi}_{i-1}^{(k)}$, $\nu_{i-1}^{(k)}$

At first glance, it also needs access to the posterior precision from the level below: $\pi_{i-1}^{(k-1)}$. However, to facilitate implementation, we can replace this, using the following derivation:

$$\hat{\pi}_{i-1}^{(k)} = \frac{1}{\frac{1}{\pi_{i-1}^{(k-1)}} + \nu_{i-1}^{(k)}}$$

$$\Leftrightarrow \pi_{i-1}^{(k-1)} = \frac{1}{\frac{1}{\hat{\pi}_{i-1}^{(k)}} - \nu_{i-1}^{(k)}}$$

Plugging this into Equation 1, we can rewrite the update equations for the volatility parent Node i, such that it only needs access to the aforementioned quantities:

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \frac{1}{2}(\kappa_{i-1}\nu_{i-1}^{(k)}\hat{\pi}_{i-1}^{(k)})^2 * (1 + (2 - \frac{1}{\hat{\pi}_{i-1}^{(k)}\nu_{i-1}^{(k)}})\delta_{i-1}^{(k)}) \tag{12}$$

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{1}{2}\kappa_{i-1}\nu_{i-1}^{(k)}\frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}}\delta_{i-1}^{(k)} \tag{13}$$

3.2 Prediction Error Step

The exact computation of the prediction error depends, like the computation of the new prediction, on the nature of the coupling with the parent nodes.

We will therefore assume in the following, that Node i is the volatility child of Node $i+1$. Then the following quantities have to be sent up to Node $i+1$ (cf. necessary information from level below in a volatility parent):

Volatility estimate: $\nu_i^{(k)}$

Predicted precision: $\hat{\pi}_i^{(k)}$

Prediction error: $\delta_i^{(k)}$

Node i has already performed the PREDICTION step on the previous trial, so it has already computed the predicted precision of the current trial, $\hat{\pi}_i^{(k)}$, and the volatiliy estimate for the current trial, $\nu_i^{(k)}$. Hence, in the PE step, it needs to perform only the following calculation:

$$\delta_i^{(k)} = \frac{\frac{1}{\pi_i^{(k)}} + (\mu_i^{(k)} - \hat{\mu}_i^{(k)})^2}{\frac{1}{\pi_i^{(k-1)}} + \nu_i^{(k)}} - 1 \tag{14}$$

Again, at first glance, the Node i needs access to a quantity from the trial before: its own posterior precision ($\pi_i^{(k-1)}$). We note however that, using the same trick as before (from Equation 1 to 3), the whole denominator equals to the inverse of the predicted precision on the current trial:

$$\hat{\pi}_i^{(k)} = \frac{1}{\frac{1}{\pi_i^{(k-1)}} + \nu_i^{(k)}}$$

Plugging this into Equation 5, we can rewrite the prediction error of Node i for its volatility parent, such that it only needs access to the aforementioned quantities:

$$\delta_i^{(k)} = \hat{\pi}_i^{(k)}(\frac{1}{\pi_i^{(k)}} + (\mu_i^{(k)} - \hat{\mu}_i^{(k)})^2) - 1 \tag{15}$$

3.3 Prediction Step

Still assuming that Node i is the volatility child of Node $i+1$, the PREDICTION step consists of the following two simple computations:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} \tag{16}$$

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \nu_i^{(k+1)}} \tag{17}$$

with

$$\nu_i^{(k+1)} = \exp(\kappa_i \mu_{i+1}^{(k)} + \omega_i).$$

Thus, the prediction for trial $k+1$ depends again only on receiving the posterior mean of Node $i+1$ on trial k, and knowing the Node's own posteriors.

Note that if Node i additionally has a VAPE parent node, the prediction of the new mean, $\hat{\mu}_i^{k+1}$ would also depend on the posterior mean of that value parent (cf. PREDICTION step for VAPE coupling).

4 Implementation HGF coupling

If we assume that a cortical column consists of units (i.e., sub-populations of neurons), one for each computational step, then the computations for VAPE coupling as well as for VOPE coupling could be implemented as sketched in Figure 1.

A few points to note here:

- In this implementation, we assume that the estimated volatility, ν_i^k, is computed within the Prediction unit. This is motivated by the equations presented above and the fact that this unit receives information about the updated quantities (posterior means and precisions) from the level above.

- Similarly, the only unit that needs to know the value of a node's tonic learning rate, ω_i, is the Prediction unit. Therefore, this parameter could be implemented in terms of self-connections or interneurons within than neuronal population.

- A noteworthy arrow in this picture is the one from the Prediction unit of one node to the Update unit of its parent node. This seems to violate the classical assumptions that while top-down connections signal predictions, bottom-up connections would only signal prediction errors. Due to the update equations in the HGF, however, any given parent node needs access to the predicted precision from its child node to perform an update. To avoid this arrow, we could also assume that the Prediction unit sends the predicted precision to the PE unit of the

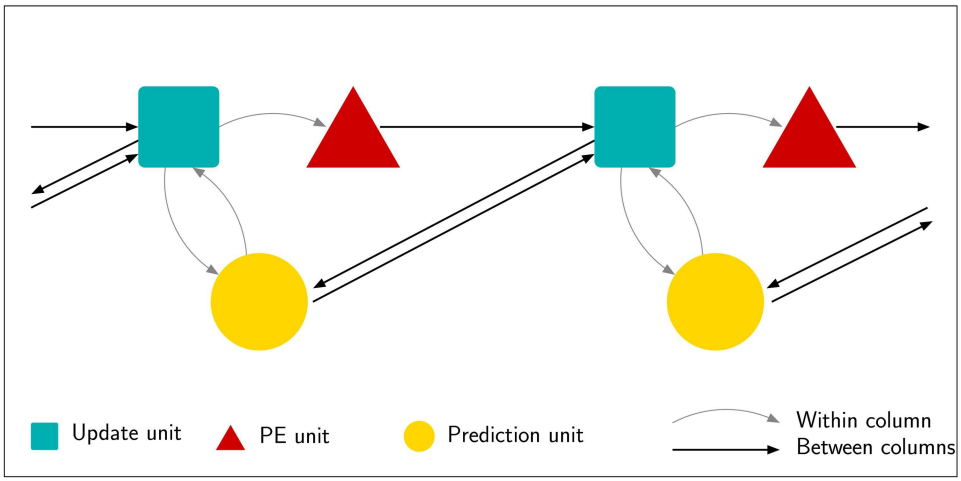

Figure 1: Example implementation of the coupling between cortical columns in the HGF. Figure shows two cortical columns that could be coupled to each other via VAPE or VOPE coupling. In each case, the messages passed along the connections and the computations within the nodes would differ according to the equations presented in the previous chapters.

same node, which then sends this quantity up to the Update unit of the parent node. However, this seems more complicated for two reasons: First, the PE unit doesn't actually need this quantity to perform its computation. Secondly, the PE is computed only somewhat after the prediction of the current trial (which is already computed after the previous trial). Thus, the two signals would be sent up at different times.

- Note that we did not distinguish between the computational steps Prediction and Update for the mean versus for the precision here. It is likely however, that these would be encoded in separate neuronal populations. Also, the implementation of VAPE and VOPE parents might differ. We will address these specifications below in comparison to classical predictive coding schemes.

5 Relation to Predictive Coding

In this section, we consider an implementation of the message passing implicated by the HGF which is as close as possible to current proposals of neural architectures for predictive coding [2].

We will separately consider VAPE and VOPE coupling and realize that the message passing for VAPE coupling is almost equivalent to the proposed

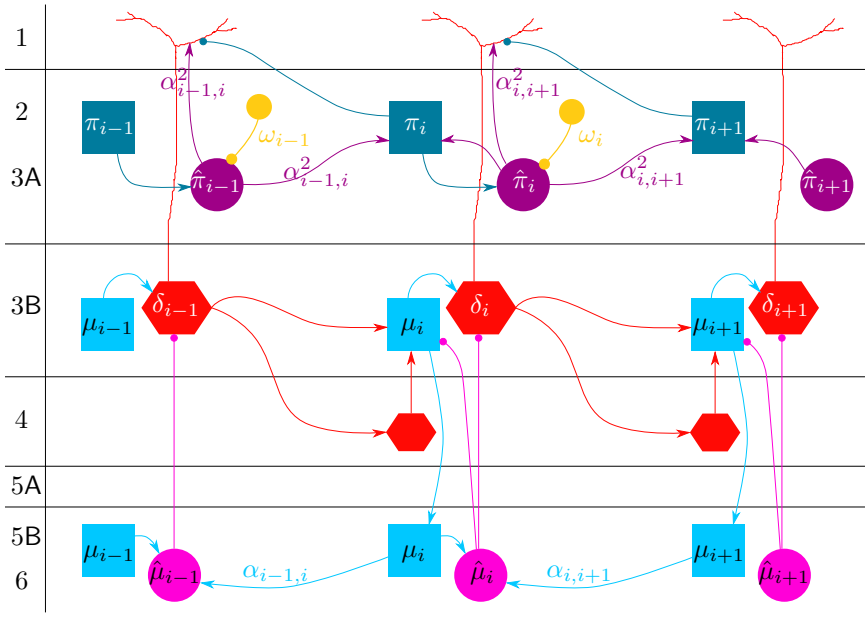

Figure 2: Overview of possible layer-specific message passing in VAPE coupling. Assignment of nodes to layers is loosely based on Figure 3 in [2]

.

PC architecture, while the VOPE coupling comes with more challenges, simply because it hasn't been addressed in detail in existing PC proposals on neural architecture.

5.1 VAPE coupling

Figure 2 sketches a possible architecture implementing all computations involved in VAPE coupling for three example cortical regions (levels). The assignment of neural elements to cortical layers follows the proposal in in figure 3 of [2]. For example, we've placed all precision-related nodes into the upper layers, while expectations and predictions of the mean live in intermediate and deep layers, respectively. In the following, we go through the different computational steps and note differences to the proposal for predictive coding.

For the PE step, we have the following computation:

$$\delta_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)}. \tag{18}$$

The message passing implicated is depicted in figure 3 and is in line with PE computations as suggested by [2].

In the Update step, the posterior estimates of mean an variance are given

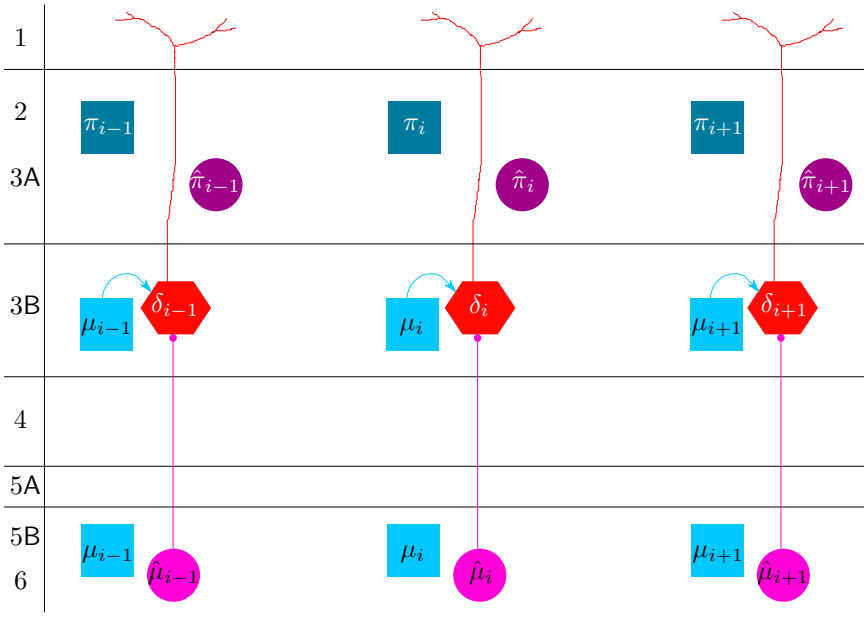

Figure 3: Implementation of VAPE PE computation on three cortical levels: $\delta_i = \mu_i - \hat{\mu}_i$.

by:

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \tag{19}$$

and

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}. \tag{20}$$

A possible message passing implementation for these updates is shown in figure 4. There are two important differences to [2] to note here.

First, in PC proposal, the update of the mean μ_i is driven both by the lower-level PE δ_{i-1} and the PE on the level itself, δ_i, which mediates the influence of the empirical prior, or the prediction, on the mean.

In the HGF equations, we do not have this negative feedback loop from δ_i to μ_i. Instead, the prediction $\hat{\mu}_i$ has a direct influence on μ_i (see equations above), hence the arrow from the $\hat{\mu}_i$ node to the μ_i node in figure 4.

However, by considering the within-trial temporal evolution of the μ_i node, we can get the same feedback loop. Noting that μ_i as computed in the above equation will be the endpoint of the update, i.e., the equilibrium value to which the node should stabilize at the end of trial k, we can, in the

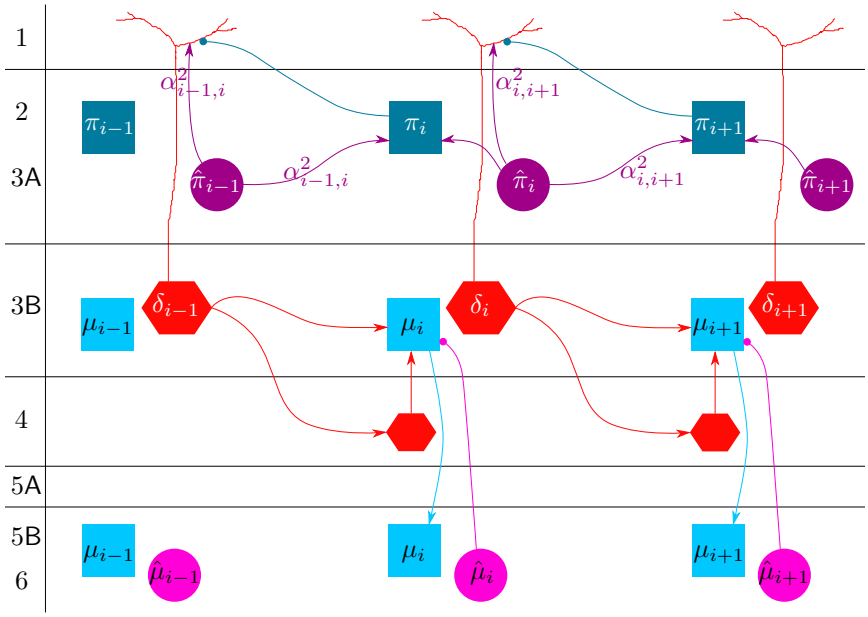

Figure 4: Message passing entailed by the Update step in VAPE coupling.

style of [1], propose the following temporal evolution:

$$\dot{\mu}_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha^2_{i-1,i}\hat{\pi}^{(k)}_{i-1}}{\pi_i^{(k)}}\delta^{(k)}_{i-1} - \mu_i^{(k)} \tag{21}$$

Clearly, a node with this dynamic converges to the required posterior value, which can be seen by setting $\dot{\mu}_i$ to zero. Now we note that

$$\dot{\mu}_i^{(k)} = \frac{\alpha^2_{i-1,i}\hat{\pi}^{(k)}_{i-1}}{\pi_i^{(k)}}\delta^{(k)}_{i-1} - \mu_i^{(k)} + \hat{\mu}_i^{(k)} \tag{22}$$

can be summarized as

$$\dot{\mu}_i^{(k)} = \frac{\alpha^2_{i-1,i}\hat{\pi}^{(k)}_{i-1}}{\pi_i^{(k)}}\delta^{(k)}_{i-1} - \delta_i^{(k)}. \tag{23}$$

Following this logic, we could also place the arrow between δ_i and μ_i. The exercise of writing down the within-trial temporal dynamics of the nodes will be repeated below for all nodes, which results in equations for within-trial network dynamics. These can be used to simulate within-trial responses of all nodes, examining network stability, and predicting neural dynamics which could for example be compared with electrophysiological data.

The second difference to the PC proposal is rather irreducible. As already mentioned in the previous section, the messages sent up to parent nodes do

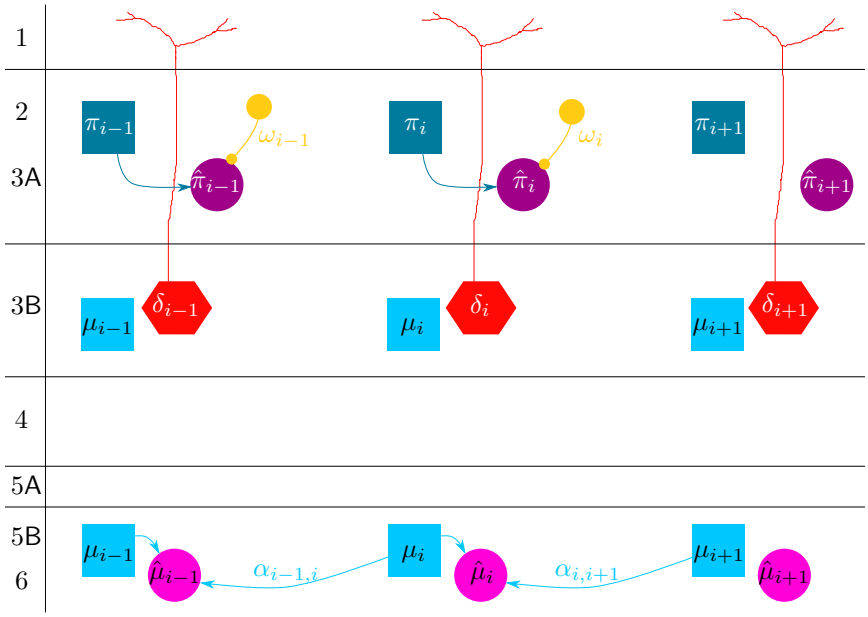

Figure 5: Message passing entailed by the Prediction step in VAPE coupling.

not only consist of prediction errors or their precision weights. Instead, to update the precision node π_i, node i also needs access to the prediction precision $\hat{\pi}_{i-1}$ from the level below. This is reflected in the violet arrow travelling from superficial layers of level i to level $i+1$.

Finally, in the Prediction step, we have:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} + \alpha_{i,i+1}\mu_{i+1}^{(k)} \tag{24}$$

and

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \exp(\omega_i)}. \tag{25}$$

This again seems unproblematic, although the arrow from μ_i to $\hat{\mu}_i$ would not appear in the classical PC architecture. Also, we've placed the predicted precision $\hat{\pi}$ in the superficial layers here, for convenience, as it interacts only with the posterior precision π and acts as a weight on the prediction error, and also inaccordance with [2], where precision signals arise from layers 2 and 3A.

5.2 VOPE coupling

For volatility parents, we propose the following: Each level of a cortical hierarchy implements its own volatility parent in superficial layers. Coupling

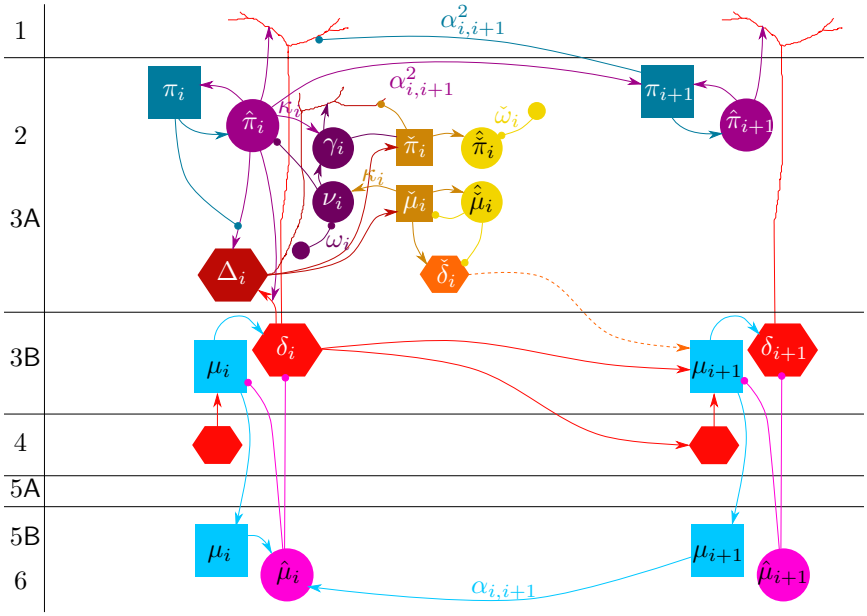

Figure 6: Overview of possible layer-specific message passing in VOPE coupling. Assignment of nodes to layers is loosely based on Figure 3 in [2]

.

across layers is exclusively value-related. This means that having a volatility parent is nothing more but a more sophisticated way of predicting and updating a level's precision estimates.

This setup excludes volatility parents of volatility parents. However, it allows a cortical hierarchy where higher levels predict either the mean, or the volatility, of lower levels, both via value coupling. In other words, any given level receives top-down predictions not only about its mean, but also about its volatility, which is implemented as the mean of its volatility parent.

Due to this proposal, we slightly change the notation from here on. Level i from now on refers to hierarchical level within a cortical hierarchy. Nodes on this level comprise the 'original' nodes μ_i, π_i, $\hat{\mu}_i$, $\hat{\pi}_i$, and δ_i, but also all the nodes belonging to a volatility parent of these nodes, which we now denote as $\check{\mu}_i$, $\check{\pi}_i$, $\hat{\check{\mu}}_i$, $\hat{\check{\pi}}_i$, and $\check{\delta}_i$. Here, $\check{\delta}_i$ refers to the value prediction error about the mean of the volatility, $\check{\mu}_i$.

We also introduce a new character for the VOPE, Δ_i, which we write as a function of the VAPE, δ_i, of level i. This PE is what communicates between a level's nodes and a level's volatility parent's nodes (see below for equations and visualizations).

Figure 6 displays one proposal for this setup, where the volatility related

nodes live within the superficial layers of each level.

The volatility prediction error (VOPE) can be written as a function of the value prediction error (VAPE), where we now use δ_i only for VAPEs:

$$\delta_i^{(k)} \equiv \delta_i^{(k,VAPE)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)}, \tag{26}$$

We define the VOPE as

$$\begin{aligned}
\Delta_i^{(k)} \equiv \delta_i^{(k,VOPE)} &= \frac{\frac{1}{\pi_i^{(k)}} + (\mu_i^{(k)} - \hat{\mu}_i^{(k)})^2}{\frac{1}{\pi_i^{(k-1)}} + \nu_i^{(k)}} - 1 \\
&= \hat{\pi}_i^{(k)} \left(\frac{1}{\pi_i^{(k)}} + (\mu_i^{(k)} - \hat{\mu}_i^{(k)})^2 \right) - 1 \\
&= \hat{\pi}_i^{(k)} \left(\frac{1}{\pi_i^{(k)}} + (\delta_i^{(k)})^2 \right) - 1 \\
&= \frac{\hat{\pi}_i^{(k)}}{\pi_i^{(k)}} + \hat{\pi}_i^{(k)} (\delta_i^{(k)})^2 - 1.
\end{aligned} \tag{27}$$

That means we are introducing a second prediction error unit Δ_i which is concerned with deviations from predicted uncertainty and is informed by value prediction errors and other estimates of uncertainty. It is this prediction error - a function of the unweighted (squared) value prediction error with a new precision weight - which communicates between a level's nodes and a level's volatility parent's nodes.

With the definition of Δ_i as given in the PE step and a new term of *expected precision* γ_i (defined for convenience in the Prediction step), the Update steps in volatility coupling become:

$$\check{\mu}_i^{(k)} = \hat{\check{\mu}}_i^{(k)} + \frac{1}{2} \frac{\gamma_{i-1}^{(k)}}{\check{\pi}_i^{(k)}} \Delta_{i-1}^{(k)} \tag{28}$$

$$\check{\pi}_i^{(k)} = \hat{\check{\pi}}_i^{(k)} + \frac{1}{2} (\gamma_{i-1}^{(k)})^2 + (\gamma_{i-1}^{(k)})^2 \Delta_{i-1}^{(k)} - \frac{1}{2} \gamma_{i-1}^{(k)} \Delta_{i-1}^{(k)} \tag{29}$$

Finally, In the Prediction step, we now need to compute four nodes: the predicted (volatility) mean

$$\hat{\check{\mu}}_i^{(k+1)} = \check{\mu}_i^{(k)}, \tag{30}$$

the precision of that prediction

$$\hat{\check{\pi}}_i^{(k+1)} = \frac{1}{\frac{1}{\check{\pi}_i^{(k)}} + \nu_i^{(k+1)}}, \tag{31}$$

the predicted environmental uncertainty (as a function of the next higher level in the hierarchy, μ_{i+1})

$$\nu_i^{(k+1)} = \exp(\kappa_{i,i+1}\mu_{i+1}^{(k)} + \omega_i), \tag{32}$$

and the new (auxiliary) expected precision

$$\gamma_i^{(k+1)} = \kappa_{i+1,i}\nu_i^{(k+1)}\hat{\pi}_i^{(k+1)}. \tag{33}$$

The last node is only defined for convenience in terms of simplifying the equations and the corresponding message passing.

6 Within-trial time dynamics

In this section, we derive differential equations which lay out the within-trial update dynamics entailed by the HGF. To this end, we consider the posterior values of all our nodes (quantities) as given by the HGF update equations as the equilibrium point towards which all dynamics must converge.

We will note that these equations imply a slightly different coupling between nodes compared to the previous section, which in most cases simplifies the implementation.

6.1 VAPE coupling

For the PE node, we want to reach the following posterior:

$$\delta_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)}. \tag{34}$$

This can be easily achieved by a node with these dynamics:

$$\dot{\delta}_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)} - \delta_i^{(k)}. \tag{35}$$

From this it follows that we require an additional inhibitory self-connection for the PE node.

In the Update step, the posterior estimate of the mean is given by:

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}}\delta_{i-1}^{(k)}. \tag{36}$$

Here, we propose the following temporal evolution:

$$\dot{\mu}_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}}\delta_{i-1}^{(k)} - \mu_i^{(k)} \tag{37}$$

Clearly, a node with this dynamic converges to the required posterior value, which can be seen by setting $\dot{\mu}_i$ to zero. Now we note that

$$\dot{\mu}_i^{(k)} = \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} - \mu_i^{(k)} + \hat{\mu}_i^{(k)} \tag{38}$$

can be summarized as

$$\dot{\mu}_i^{(k)} = \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} - \delta_i^{(k)}. \tag{39}$$

We will therefore consider a connection between δ_i and μ_i, instead of connecting the prediction node $\hat{\mu}_i$ with the mean μ_i. This also resolves one difference to classical PC proposals, as already outlined in the previous section.

For the posterior precision, we have

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}. \tag{40}$$

Here, we stick with the same approach as for the PE node, and simply add a self-inhibitory connection to implement the dynamics:

$$\dot{\pi}_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)} - \pi_i^{(k)}. \tag{41}$$

Finally, in the Prediction step, we want to reach the following prediction of the mean:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} + \alpha_{i,i+1} \mu_{i+1}^{(k)}. \tag{42}$$

Rewriting this as a differential equation

$$\dot{\hat{\mu}}_i^{(k+1)} = \alpha_{i,i+1} \mu_{i+1}^{(k)} + \mu_i^{(k)} - \hat{\mu}_i^{(k+1)}, \tag{43}$$

we again see the PE node appear:

$$\dot{\hat{\mu}}_i^{(k+1)} = \alpha_{i,i+1} \mu_{i+1}^{(k)} + \delta_i^{(k)}. \tag{44}$$

Note that $\delta_i^{(k)}$ is actually defined in terms of $\hat{\mu}_i^{(k)}$, and not $\hat{\mu}_i^{(k+1)}$, but given that we are in continuous time now, we suspect that this is equivalent here.

The precision of this prediction should converge to:

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \exp(\omega_i)}. \tag{45}$$

We can write the dynamics as:

$$\begin{aligned}\dot{\hat{\pi}}_i^{(k+1)} &= 1 - (\frac{1}{\pi_i^{(k)}} + \exp(\omega_i))\hat{\pi}_i^{(k+1)} \\ &= 1 - \frac{\hat{\pi}_i^{(k+1)}}{\pi_i^{(k)}} + \hat{\pi}_i^{(k+1)} \exp(\omega_i).\end{aligned} \tag{46}$$

To simplify the implementation, we introduce an additional node p_i:

$$p_i = \frac{\hat{\pi}_i^{(k+1)}}{\pi_i^{(k)}}. \tag{47}$$

For simplicity, and because we are working in continuous time, we will skip the trial indices from now on.

The new node

$$p_i = \frac{\hat{\pi}_i}{\pi_i} \tag{48}$$

can have the following dynamcis:

$$\dot{p}_i = \hat{\pi}_i - \pi_i p_i, \tag{49}$$

and the dynamics for the precision of the prediction $\hat{\pi}_i$ become

$$\dot{\hat{\pi}}_i = 1 - p_i + \hat{\pi}_i \exp(\omega_i). \tag{50}$$

We now consider an alternative implementation for the PE computation, which will turn out to be very useful.

Instead of considering the unweighted δ_i, we can also directly model a node which corresponds to the precision-weighted prediction error ε_i:

$$\varepsilon_i = \frac{\alpha^2 \hat{\pi}_i}{\pi_{i+1}}(\mu_i - \hat{\mu}_i) = \frac{\alpha^2 \hat{\pi}_i}{\pi_{i+1}}\delta_i. \tag{51}$$

This node would evolve according to

$$\dot{\varepsilon}_i = \mu_i - \hat{\mu}_i - \frac{\pi_{i+1}}{\alpha^2 \hat{\pi}_i}\varepsilon_i. \tag{52}$$

We can then re-introduce the unweighted PE δ_i:

$$\begin{aligned}\delta_i &= \frac{\pi_{i+1}}{\alpha^2 \hat{\pi}_i}\varepsilon_i \\ &= \frac{\pi_{i+1}(\mu_i - \hat{\mu}_i)\alpha^2 \hat{\pi}_i}{\alpha^2 \hat{\pi}_i \pi_{i+1}} \\ &= \mu_i - \hat{\mu}_i,\end{aligned} \tag{53}$$

such that

$$\dot{\varepsilon}_i = \mu_i - \hat{\mu}_i - \delta_i \tag{54}$$

and

$$\dot{\delta}_i = \frac{\pi_{i+1}}{\alpha^2}\varepsilon_i - \hat{\pi}_i\delta_i. \tag{55}$$

This has the advantage that the weighted PE ϵ_i can travel up the hierarchy to update higher levels, while the unweighted PE δ_i can be used to feedback to the mean on the same level.

6.2 VOPE coupling

7 Open Questions and Issues

- To Do:
  - Complete differential equations for VOPE coupling and visualize the resulting network of nodes and their connections.
  - Implement these differential equations, examine the stability of the network and present exemplary simulations.
- Equations: The coupling parameters α and κ must be available to both the connection between a node's PE unit and the parent's UPDATE unit, as well as the connection between a node's PREDICTION unit and the parent's UPDATE unit. How can it be ensured that the coupling strength is the same for both connections? For this question we would actually have to consider the learning or update equations for the parameters of the model (see for example the approach in [1]). However, even without writing them down, we could assume that under 'healthy' conditions, the weights of these different connections automatically converge to the same (or at least very similar) values. This opens up new possibilities of modeling abnormal inference: Using the simulations presented above, we could find out how an agent's inference would be altered if these values were not the same, and what range of difference would actually lead to observable changes in the inference.
- Implementation: To whom does the knowledge about parameters, in particular coupling strengths, belong? Ideally, this would belong to the connections themselves and be accessible to both child and parent node. However, in terms of synaptic signalling, how can a node have direct access to a coupling strength other than via the signalled value (which is the product of the coupling strength and the signal)?

- Not covered here: Computations of input nodes and binary input nodes.

References

[1] Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76:198–211.

[2] Shipp, S. (2016). Neural elements for predictive coding. *Frontiers in Psychology*, 7(NOV):1–21.