DONDERS INSTITUTE

Neuromorphic Square, or, where Predictive Processing and Robotics meet

Johan Kwisthout, Donders Center for Cognition

Radboud University Radboudumc

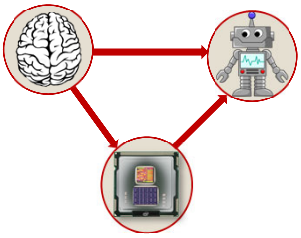
DONDERS INSTITUTE

Robotics Square, or, where Predictive Processing and Neuromorphic meet

Johan Kwisthout, Donders Center for Cognition

Radboud University Radboudumc

DONDERS INSTITUTE



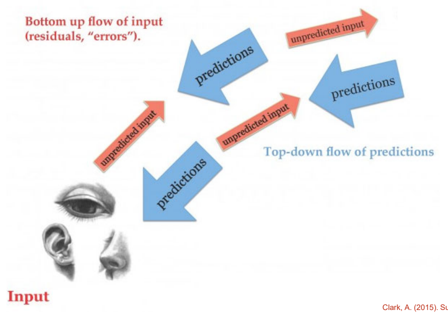
Probabilistic Graphical Models unifying all of Robotics, Predictive Processing, & Neuromorphic

Johan Kwisthout, Donders Center for Cognition

Radboud University Radboudumc

A novel view on the brain – Predictive Processing

"...essentially a sophisticated hypothesis-testing mechanism..."



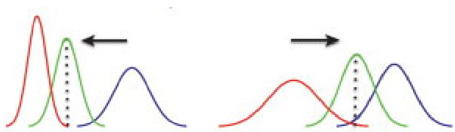
Clark, A. (2015). Surfing Uncertainty.

Key sub-processes in Predictive Processing

- Making (stochastic) **predictions** of expected input based on generative models
- Comparing predicted inputs with actual inputs and **establishing prediction error**
- **Explaining away** prediction errors (minimizing prediction error) by action or perceptual updates
- **Learning** and adapting generative models over time

The crucial role of precision weighting

- Prediction errors are **weighted** with their expected precision – pushes posterior more or less to prior



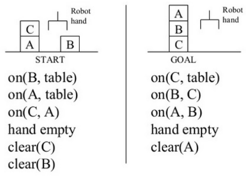
- Generative (or *inverse*) models are updated with each new experience according to principles of Bayesian updating, governed by hyperparameters

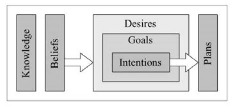
Seth, A. (2013). Interoceptive inference, emotion, and the embodied self.

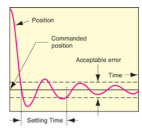
Predictive Processing as unifying concept

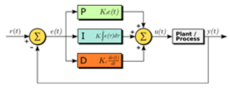
- The idea that **perception** is a two-way process (bottom-up & top-down) is not new (e.g. Helmholtz)
- Novel suggestion: action is just a logical, necessary **consequence** of a **self-created** prediction error
- Prediction not how I **believe** the world to be, but how I **intend** the world to be – think of a **set point**
- Action ('motor commands') is a **means** to reduce prediction error between current and desired state

Active inference in robotics & agent systems





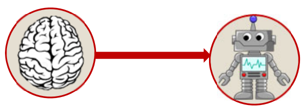




High level control

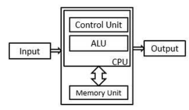
Low level control

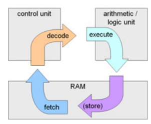
Conceptual similarity (part 1)



- Prediction error minimization as unifying principle?
  - Means-end action planning (high-level active inference)
  - PID controlled actuators (low-level active inference)
  - Inference to the best explanation (high-level perception)
  - Movement detection (low-level perception)
- Generative model + prediction mechanism + error detection + (weighted) prediction error minimization

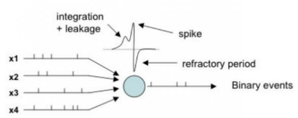
A novel view on computing





- Traditional (Von Neumann) model of computation
  - Physically separated memory and computation
  - Information and computation based on Boolean logic
  - Serial (or only very limited parallel) computation
  - Energy-demanding; symmetry (1s and 0s equal)
  - Well established computational model (Turing Machine), algorithms, computational complexity theory

A novel view on computing

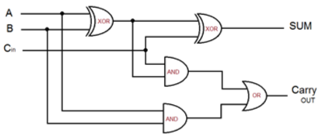


- Neuromorphic (brain-inspired) model of computation
  - Co-located memory and computation
  - Information and computation based on spiking neurons
  - Massively parallel computation
  - Energy-lean (at least potentially); spikes cost energy
  - Computational model, algorithms, computational complexity results are still in their infancy

A different way of thinking

- Neuromorphic architectures allow for (and dictate!) a **different way of thinking** about computations
- Analog behavior → get rid of digital regime
- Stochastic behavior → randomized algorithms
- Define connectivity matrix rather than instructions
- Characteristics of novel materials (e.g spin glasses) let you think differently about **synchronization**
- Spiking behavior allows **temporal** data structures

A different way of thinking

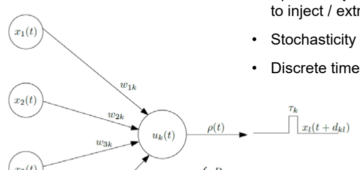


Digital adder circuit

- Numbers are encoded as binary digits in memory
- Computation (adding two numbers) implemented by a logical circuit within the CPU
- Spiking neural networks allow for a different way of **representing** information and **computing** with it

Neuronal model: basically simple LIF model

- Specific **input** and **readout** neurons to inject / extract information
- Stochasticity or determinism
- Discrete time steps



$$u_k(t) = \begin{cases} R_k, & \text{if } u_k(t-1) \geq T_k; \\ m_k u_k(t-1) + \sum_j w_{jk} x_j(t - d_{jk}), & \text{otherwise.} \end{cases}$$

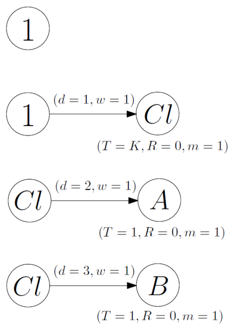
$R_k = \text{reset voltage}, T_k = \text{threshold}, m_k = \text{leakage constant}$

$$\rho_{\text{det}}(t) = \begin{cases} 1, & \text{if } u_k(t) \geq T_k; \\ 0, & \text{otherwise.} \end{cases}$$

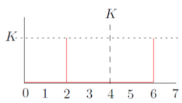
$\rho_{\text{stoc}}(t) \propto \exp(u_k(t))$

$$x_k(t) = \begin{cases} 1, & \text{if a spike was released in } \langle t - \tau_k, t]; \\ 0, & \text{otherwise.} \end{cases}$$

Example problem: adding two numbers mod K

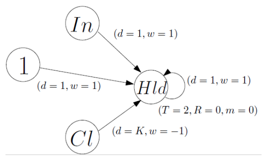


- Special neuron Const fires at every time step >= 0 (bias = threshold)
- Clock neuron fires every K timesteps
- A and B fire at time d mod K (d = 2,3)

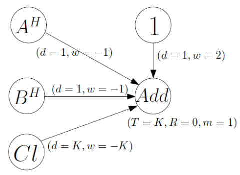


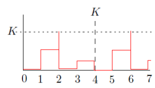
- Spiking time of the neuron relative to clock encodes its value!

Example problem: adding two numbers mod K

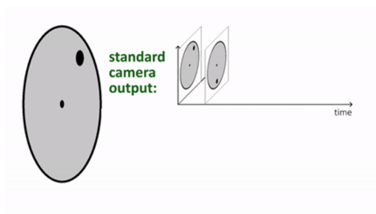


- Circuit ensuring that once a neuron fires, it fires until reset by the clock
- Adder circuit spikes on carry-out, neuron potential at clock intervals encodes A+B mod K



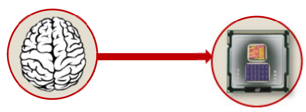


Neuromorphic computing in robotics



- Event-driven sensors (e.g. dynamic vision sensors)
  - Report *change* rather than full frame
  - Temporal information (*when* as well as *what*)
  - Energy lean (in a stable environment)

Conceptual similarity (part 2)



- Event-driven signaling as unifying principle?
  - Predictive Processing: only process information that was not *anticipated* (surprising / stochastic information)
  - Spike-based neuronal communication: spikes (neuron firing) signifies relevant event
  - But: prediction error ≠ spike (save maybe at the very lowest input level, e.g., retinal cells)

How to tie it all together?

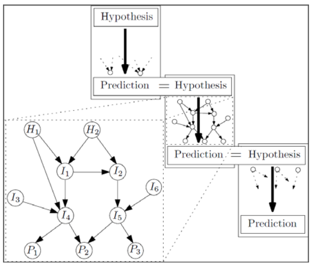
- Prediction error minimization (active and perceptual inference) seems a useful **paradigm** for robotics
- Prediction error as information carrier almost begs to be **implemented** by neuromorphic architectures
- **Mission:**
  Formulate PP in a *unifying* computational framework
  - **One language** for describing high- and low-level operation
  - Framework **fits naturally** with neuromorphic architectures
  - **Scientific** as well as engineering benefits
  - **Explainable** & justifiable – avoid black boxes

Probabilistic graphical models as pivot element

- PGMs can describe **high-level** structured information as well as **low-level** statistical regularities
- Computational Bayesian models of **cognition**
- **Same** principles for low-level vision, motor control
- **Explainable AI** comes 'for free' with PGMs – nodes represent stochastic variables
- Offer meaningful descriptions at all Marr's levels of explanation: computation, algorithm, implementation

PP at the Computational level

- Example: **Causal Bayesian networks** describing (context-specific, structured, complex) relation between hypotheses and predicted consequences



Hypothesis variables
Hyp = {H_1, H_2}

Prediction variables
Pred = {P_1, P_2, P_3}

Intermediate variables
Int = {I_1, … I_6}

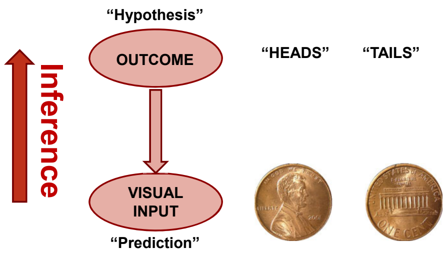
Hyperparameters (Beta or Dirichlet distribution)

PP at the Computational level

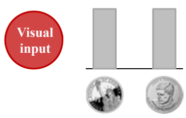
- Prediction and Observation are **probability distributions** over the prediction variables *Pred*
- Prediction is defined as computing the **marginal distribution** $\Pr_{(Pred)}$ given the parameters in the network
- Prediction error is set difference $\Pr_{(obs)}$ - $\Pr_{(Pred)}$
- The size of this error is defined as the **KL-divergence** between predicted distribution and observed distribution

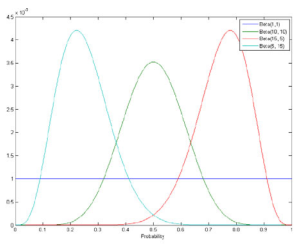
$$D_{\mathrm{KL}}(\Pr_{(\mathrm{Obs})} \| \Pr_{(\mathrm{Pred})}) = \sum_{\mathbf{p} \in \Omega(\mathrm{Pred})} \Pr_{\mathrm{Obs}}(\mathbf{p}) \log\left(\frac{\Pr_{\mathrm{Obs}}(\mathbf{p})}{\Pr_{\mathrm{Pred}}(\mathbf{p})}\right)$$

Example: tossing coins



Beta distribution

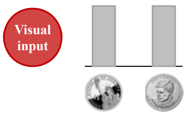


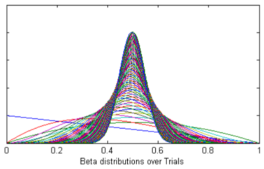


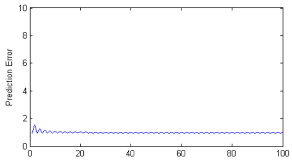
$$P(p;\alpha,\beta) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$$

$$\mathrm{B}(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\,dt = \frac{\Gamma(x)\,\Gamma(y)}{\Gamma(x+y)}. \qquad \mathrm{P}(X=x) = \frac{\alpha}{\alpha+\beta}$$

Bayesian Updating







s = heads, n-s = tails

$$P(p;\alpha,\beta) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$$

PP at the Computational level

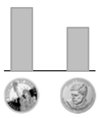
- All relevant concepts in PP can be precisely defined
  - Precision of predictions: entropy of P(predictions)
  - Expected precision of observations
  - Precision-weighted prediction error
  - Impact of prediction error on generative model
  - Intervention (do-calculus) for active inference
  - Belief revision (explaining away) for perceptual inference
- But a few conceptual challenges come to play
  - State-space granularity, level-of-detail of predictions
  - Non-trivial action selection to minimize prediction error
  - Learning beyond Bayesian updating (e.g. new concepts)

PP at the Algorithmic level

- Approximate Bayesian inference
  - Making predictions by sampling or variational methods
  - Minimizing prediction error by local search methods
  - Learning by (approximately) updating hyper-parameters
- Heuristics, stereotypes, exemplars may help approximate action selection to a satisfactory level
- But a few practical challenges remain
  - Tractability remains an issue also for approximation!
  - Conceptual issues must be solved first before they can be implemented at the algorithmic level

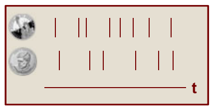
PP at the Implementational level

- Probability distributions and approximate Bayesian inferences can be realized with Spiking Neural Nets
  - See work in Wolfgang Maass' group
- Hyperparameters of generative models can be encoded in spiking frequency (**rate** / temporal)



P(heads) = 0.66

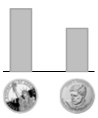
α = 2, β =1



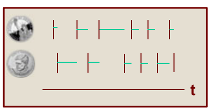
α = 20, β =10

PP at the Implementational level

- Probability distributions and approximate Bayesian inferences can be realized with Spiking Neural Nets
  - See work in Wolfgang Maass' group
- Hyperparameters of generative models can be encoded in spiking frequency (rate / **temporal**)



P(heads) = 0.66



α = 20, β =10

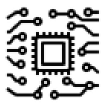
Still a long way to go

- There still remain challenges in realizing
  - Learning hyperparameters
  - Establishing prediction error
  - Weighting errors with precision
  - Processing weighted prediction error
  - ….
- We don't yet know very well **how** and **what** to compute efficiently with SNNs and how PP information processing can be realized
- Work in progress in our lab: Neuromorphic complexity theory (INRC project)

Research agenda – challenges

Conceptual & computational

Algorithm & implementation



Hardware level

Computational and conceptual challenges

- Translation of some conceptual aspects of PP to computational model still work in progress:
  - How to learn and represent expected precision in generative models?
  - How to learn and represent expected state space granularity in generative models?
  - How to formalize counterfactual reasoning?
- Creative part of learning mostly ignored in PP
  - When do we need a new concept or context-dependency? How to integrate that in our generative models?
- How do generative models / PP mechanisms change in development?

Implementational challenges

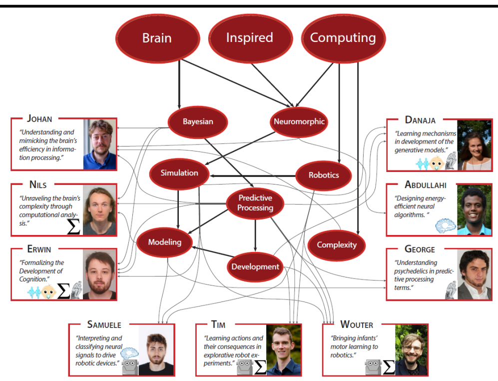
- Basically we have just started…
- Challenge of different levels of abstraction
  - Prediction can be single cell firing / silent (retina)
  - Prediction can be complex combination of intentions
  - Need different neural encoding – how to integrate?
- Challenge of brains vs neuromorphic architecture
  - Dopamine is believed to be associated with precision error modulaton, serotonin with state-space-granularity
  - No direct translation to spiking neural networks!

Hardware / sensor challenges

- Does it even make sense?
- Further develop the concept of "prediction-driven sensors" (signal when deviating from prediction)
  - How to 'load' a prediction in such a sensor and 'retrieve' an error? What is the overhead and is it even worth it?
  - How to translate abstract concepts (I am walking in a busy street and expect to see 'movement' around me) to concrete predictions that such a sensor can use?
  - What inspiration can we get from the brain (e.g., movement sensitive cells in V1)

Conclusion

- PP as unifying theory for both "life as we know it" as artificial intelligence / robotics may be fruitful
- PGMs as natural computational modeling and descriptive language for PP at all levels of description
- Neuromorphic computation is a natural candidate for implementing PP principles realized by PGM computations
- Work needs to be done on prediction-error based sensors, computations, and implementations



DONDERS INSTITUTE

NWO Netherlands Organisation for Scientific Research

intel