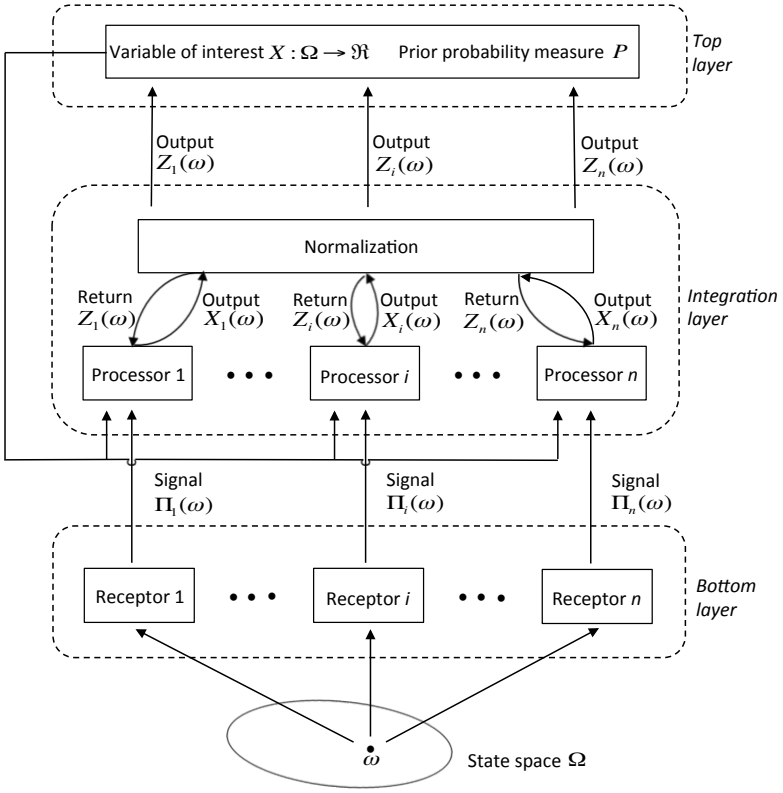
# Concurrence and Common Knowledge in a Neural Network*

Adam Brandenburger†, Pierfrancesco La Mura‡, and Kai Steverson§

January 22, 2017

**Preliminary Notes**

## 1 Introduction




*We are grateful to Paul Glimcher, Kenway Louie, and Jan Zimmerman for suggestions. Financial support from the NYU Stern School of Business and J.P. Valles is gratefully acknowledged.

†Stern School of Business, Tandon School of Engineering, Institute for the Interdisciplinary Study of Decision Making, New York University, New York, NY 10012, U.S.A., adam.brandenburger@stern.nyu.edu, http://www.adambrandenburger.com

‡HHL - Leipzig Graduate School of Management, 04109 Leipzig, Germany, plamura@hhl.de

§Institute for the Interdisciplinary Study of Decision Making, New York University, New York, NY 10012, U.S.A., ksteverson@nyu.edu, https://sites.google.com/site/ksteverson/

We consider the preceding abstract **neural network**, consisting of a bottom layer, a top layer, and an integration layer.

There are $n$ receptors, indexed by $i = 1, \ldots, n$, in the **bottom layer** of the network. Each receptor receives an imperfect signal about an underlying state of the world. This is formalized by introducing a set $\Omega$ (assumed finite, for simplicity) of possible underlying states. The signal received by each receptor will be described in terms of a partition of $\Omega$. Thus, associated with receptor $i$ is a partition $\Pi_i$ of $\Omega$. If the true state of the world is $\omega \in \Omega$, then the signal which receptor $i$ receives is the element that contains $\omega$ of the partition $\Pi_i$. We denote this element by $\Pi_i(\omega)$.[1] Receptors do not process signals but simply feed them forward to the integration layer of the network.

The **top layer** of the network houses the (Bayesian) prior probabilities of the states of the world, represented by a probability measure $p$ on $\Omega$. (To keep things simple, we suppose $p(\omega) > 0$ for all $\omega \in \Omega$.) The top layer also determines the (state-dependent) object of interest for the network. We model this as a strictly positive-valued random variable $X : \Omega \to \mathbb{R}_{++}$.

The **integration layer** of the network consists of $n$ processors, indexed by $i = 1, \ldots, n$. Processor $i$ receives signal $\Pi_i(\omega)$ from receptor $i$ (in the bottom layer) and the prior $p$ and random variable $X$ (from the top layer), and it uses these to form the conditional expectation of $X$:

$$X_i(\omega) \triangleq \mathbf{E}(X \mid \Pi_i(\omega)) = \frac{1}{p(\Pi_i(\omega))} \sum_{\omega' \in \Pi_i(\omega)} p(\omega') X(\omega').$$

The quantity $X_i(\omega)$ is the output of processor $i$ when the true underlying state is $\omega$. These quantities, one for each processor $i = 1, \ldots, n$, feed forward in the network.

# 2 Normalization

The central feature of the network we consider is that there are normalization and feedback processes at work in the integration layer. Specifically, the output signals $X_1(\omega), \ldots, X_n(\omega)$ enter a **neural normalization** step in the integration layer, which computes a generalized sum of these signals and divides each signal by the sum. Formally, for each $i = 1, \ldots, n$, let $f_i : \mathbb{R} \to \mathbb{R}$ be a strictly increasing function. The normalization step computes the sum

[1] This method of modeling signals is quite general and, for example, can handle signals that take the form of random variables defined on the underlying state space $\Omega$.

$$Y(\omega) \triangleq \sum_{i=1}^{n} f_i(X_i(\omega))$$

and then the normalized signals

$$Z_i(\omega) \triangleq \frac{X_i(\omega)}{Y(\omega)} \text{ for } i = 1, \ldots, n.$$

These signals go forward into the top layer, for unspecified further processing. But, of interest to us in this note is that they also feed backward to the respective processors. This means that each processor $i$ actually receives two signals: first, the basic input signal $\Pi_i(\omega)$, and, after some interval, backward propagation of the normalized signal $Z_i(\omega)$.

Next, processor $i$ extracts the information in the normalized signal and outputs a new signal. To describe this process, we add superscripts, and we write $\Pi_i^0$ for the partition $\Pi_i$, and $\Pi_i^0(\omega)$ and $Z_i^0(\omega)$ for the preceding two signals. We also write $X_i^0(\omega) \triangleq X_i(\omega)$ and $Y^0(\omega) \triangleq Y(\omega)$. Notice that the normalization variable $Y^0$ induces a partition $\Psi^0$ of $\Omega$, where the element of $\Psi^0$ that contains $\omega$ is given by

$$\Psi^0(\omega) \triangleq \left\{\omega' \in \Omega : Y^0(\omega') = Y^0(\omega)\right\}.$$

We assume that each processor $i$ can extract the value of the normalization variable $Y^0$ from its backward-propagated signal $Z_i^0$ (the computation is just $Y^0 = X_i^0/Z_i^0$). Let $\vee$ denote the **coarsest common refinement** (also called the **join**) of two or more partitions. Then the partitions

$$\Pi_i^1 \triangleq \Pi_i^0 \vee \Psi^0 \text{ for } i = 1, \ldots, n,$$

give the information that is available to each of the processors $i = 1, \ldots n$ after they have each received and processed a basic input signal and a backward propagation of the normalized output signal.[2]

This whole process is repeated. Thus, each processor $i$ next computes the conditional expectation

$$X_i^1(\omega) \triangleq \mathbf{E}(X \mid \Pi_i^1(\omega)),$$

[2]Since $X_i^0$ is measurable with respect to the partition $\Pi_i^0$, the same partition $\Pi_i^1$ is generated by forming the join of the partition $\Pi_i^0$ and the partition generated by the signal $Z_i^0$. The step of extracting the value of $Y^0$ from $Z_i^0$ is therefore redundant. We add this step only to simplify some arguments to come.

the normalization step computes

$$Y^1(\omega) \triangleq \sum_{i=1}^{n} f_i(X_i^1(\omega))$$

and the normalized signals

$$Z_i^1(\omega) \triangleq \frac{X_i^1(\omega)}{Y^1(\omega)},$$

which are then fed forward and also backward propagated. Letting $\Psi^1$ be the partition of $\Omega$ generated by the normalization variable $Y^1$, we arrive at new partitions

$$\Pi_i^2 \triangleq \Pi_i^1 \vee \Psi^1 \text{ for } i = 1, \ldots, n, \tag{1}$$

Iterated processing continues as the processors again compute new conditional expectations, and the normalization and backward propagation steps again take place. We ask two questions about this process:

- Does the iterated processing converge?
- What is the output of each processor once convergence takes place (if it does)?

# 3 Convergence

It is easy to establish **convergence** when the state space $\Omega$ is finite, which is the case we consider here.[3] Generalizing Equation (1) above, we can write

$$\Pi_i^{t+1} \triangleq \Pi_i^t \vee \Psi^t \text{ for } i = 1, \ldots, n \text{ and } t = 0, 1, \ldots. \tag{2}$$

Thus, for each processor $i$, the associated partition $\Pi_i^{t+1}$ at iteration $t+1$ refines its partition $\Pi_i^t$ at iteration $t$. Since $\Omega$ is finite, there must be a $T_i$ such that $\Pi_i^{t+1} = \Pi_i^t$ for all $t \geq T_i$. Let $T = \max_i T_i$. Then we can conclude that the partition at every processor, i.e., the information at every processor about the underlying state, stabilizes at iteration $T$ by the latest.

We can also see that the normalization variable

$$Y^t(\omega) \triangleq \sum_{i=1}^{n} f_i(X_i^t(\omega)).$$

[3]For general (infinite) $\Omega$, a martingale convergence argument would be used.

converges. For this, simply note that

$$X_i^{T+1}(\cdot) = \mathbf{E}(X \mid \Pi_i^{T+1}(\cdot)) = \mathbf{E}(X \mid \Pi_i^T(\cdot)) = X_i^T(\cdot),$$

and therefore $Y^{T+1}(\cdot) = Y^T(\cdot)$, and, indeed, $Y^{T+v}(\cdot) = Y^T(\cdot)$ for all $v = 1, 2, \ldots$.

An additional property of $Y^T(\cdot)$ will be central to the next section. Let $\Pi^t$ denote the partition that is the **finest common coarsening** (also called the **meet**) of the partitions $\Pi_1^t, \ldots, \Pi_n^t$; symbolically,

$$\Pi^t \triangleq \Pi_1^t \wedge \cdots \wedge \Pi_n^t.$$

Using the stabilization of the partitions at $t = T$, and substituting $t = T$ in Equation (2), gives

$$\Pi_i^{T+1} = \Pi_i^T = \Pi_i^T \vee \Psi^T \text{ for } i = 1, \ldots, n,$$

which implies that the partition $\Psi^T$ is a (weak) coarsening of each partition $\Pi_i^T$. From this we can conclude that the partition $\Psi^T$ is a (weak) coarsening of the meet $\Pi^T$ of the partitions $\Pi_1^T, \ldots, \Pi_n^T$.

# 4 Concurrence

We next establish that there is eventual concurrence among the outputs of the processors. That is, the outputs not only converge, they do so to a common value. The proof of the following result is from Nielsen, Brandenburger, Geanakoplos, McKelvey, and Page (1990), who established the result in an abstract setting (without neural content).

**Proposition 1.** *Fix a state $\omega \in \Omega$. There is a (finite) iteration $T$ after which the values of all the output signals $X_1^T, \ldots, X_n^T$ are equal.*

*Proof.* Define

$$X^t(\omega) \triangleq \mathbf{E}(X \mid \Pi^t(\omega)).$$

By what we said at the end of the previous section, we know that the value of $Y^T(\cdot)$ is constant on $\Pi^T(\omega)$, from which it follows that we can write, for $t = T$,

$$0 = \sum_{\omega' \in \Pi^T(\omega)} \sum_{i=1}^{n} f_i(X_i^T(\omega'))[X(\omega') - X^T(\omega')]p(\omega'),$$

which we can rearrange as

$$0 = \sum_{i=1}^{n} \sum_{\omega' \in \Pi^T(\omega)} f_i(X_i^T(\omega'))[X(\omega') - X^T(\omega')]p(\omega').$$

By the properties of conditional expectation, we then have

$$0 = \sum_{i=1}^{n} \sum_{\omega' \in \Pi^T(\omega)} f_i(X_i^T(\omega'))[X_i^T(\omega') - X^T(\omega')]p(\omega').$$

We can next insert a term to get

$$0 = \sum_{i=1}^{n} \sum_{\omega' \in \Pi^T(\omega)} [f_i(X_i^T(\omega')) - f_i(X^T(\omega'))][X_i^T(\omega') - X^T(\omega')]p(\omega'). \tag{3}$$

Since each $f_i$ is strictly increasing, we know that

$$f_i(X_i^T(\omega')) \gtreqless f_i(X^T(\omega')) \text{ if and only if } X_i^T(\omega') \gtreqless X^T(\omega'),$$

(as $\omega'$ varies), so that for each $i$,

$$\sum_{\omega' \in \Pi^T(\omega)} [f(X_i^T(\omega')) - f_i(X^T(\omega'))][X_i^T(\omega') - X^T(\omega')]p(\omega') \geq 0, \tag{4}$$

with strict inequality if $X_i^T(\omega') \neq X^T(\omega')$ for some $\omega' \in \Pi^T(\omega)$. But strict inequality in Equation (4) contradicts Equation (3). We can therefore conclude that $X_i^T(\omega) = X^T(\omega)$ for all $i = 1, \ldots n$. □

# 5 Common Knowledge

We next investigate some epistemic properties of the processors. Fix an underlying state $\omega \in \Omega$ and an **event** $A \subseteq \Omega$. We say that processor $i$ **knows** event $A$ at $\omega$ if $\Pi_i(\omega) \subseteq A$. An illuminating way to write this condition is that for all $\omega' \in \Pi_i(\omega)$, it is the case that $\omega' \in A$. In words, every underlying state that is consistent with processor $i$'s information is also consistent with the occurrence of $A$. Notice that the event that processor $i$ knows $A$, to be denoted by $K_i(A)$, is

$$\{\omega \in \Omega : \Pi_i(\omega) \subseteq A\}.$$

Let $K(A)$ be the event that every processor knows $A$, that is,

$$K(A) = \bigcap_{i=1}^{n} K_i(A).$$

An event $A$ is **common knowledge** at a state $\omega$ if every processor knows $A$ at $\omega$, every processor knows at $\omega$ that every processor knows $A$, and so on indefinitely, that is, $\omega \in K(A) \cap K(K(A)) \cap \cdots$. Let $\Pi$ denote the meet of the partitions $\Pi_1, \ldots, \Pi_n$, that is,

$$\Pi \triangleq \Pi_1 \wedge \cdots \wedge \Pi_n.$$

Aumann (1976) showed (in a general setting) that the common-knowledge condition $\omega \in K(A) \cap K(K(A)) \cap \cdots$ is equivalent to the condition $\Pi(\omega) \subseteq A$.

Combining this characterization of common knowledge with the argument in the previous section, we see that not only is there eventual concurrence among the outputs of the processors (Proposition 1), there is also eventual common knowledge among the processors that their outputs are all equal to a common value. Formally:

**Proposition 2.** *Fix a state $\omega \in \Omega$. There is a (finite) iteration $T$ after which it is common knowledge among the processors that every processor's output is equal to the common value $X^T(\omega)$.*

# References


1. Aumann, R., "Agreeing to Disagree," *Annals of Statistics*, 4, 1976, 1236-1239.

2. Nielsen, L., A. Brandenburger, J. Geanakoplos, R. McKelvey, and T. Page, "Common Knowledge of an Aggregate of Expectations," *Econometrica*, 58, 1990, 1235-1239.