Information Theory of Hierarchical Circuits

Anton Kibalnik and Luis M. A. Bettencourt

Abstract

We extend some of the methods of [1] to circuits with N inputs and then apply them to a hierarchical circuit - a branching tree network - as a toy model of the visual system [3].

1 The information theory of elementary circuits with n inputs

Analagous to the three cell analysis in [1], for groups of four cells there is a number of possible arrangements of the cells, but now this number is much greater. We list several cases here:

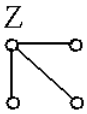
No links; $I(X_iX_j) = 0$ for all $\{i,j\}$; $R = 0$

1 link; $I(X_iX_j) \neq 0$ for a unique $\{i,j\}$; $R = 0$

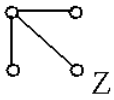
2 separate links; $I(X_iX_j) \neq 0$ for $\{i_1, j_1\}$ and $\{i_2,j_2\}$ where each i_1,j_1,i_2,j_2 are all different; $R = 0$

3 cell chain; 3 cases if Z is in chain ($R > 0$) and 3 cases if Z is not in chain ($R = 0$)

3 cell functioning unit; R < 0 if Z is inside, R < 0 if Z is outside



4 cell functioning unit; R < 0



3 cell functioning unit which relays information to Z; R < 0 ?

4 cell chain... R > 0

.
.
.

Whatever the arrangement, it can be determined (in principle) by calculation of the mutual information and R values. However, this method becomes less practical for n > 5 as the number of arrangements explodes combinatorilly. But we can still explore how the entropic quantities scale with increasing n.

The definition of R [1], [2] can be extended to for an arbitrary number of random variables:

$$R = I(X_i) - I(\{X_i\}|Z) \tag{1}$$

$$= \sum_{i=1}^{n} I(X_i Z) - I(\{X_i\}Z) \tag{2}$$

$$= \sum_{i=1}^{n} [H(X_i) - H(X_i|Z)] - [H(\{X_i\}) - H(\{X_i\}|Z)] \tag{3}$$

$$= n - \sum_{i=1}^{n} H(X_i|Z) - n + H(\{X_i\}|Z) \tag{4}$$

$$= H(\{X_i\}|Z) - \sum_{i=1}^{n} H(X_i|Z) \tag{5}$$

Now, since mutual information is invariable under permutations of the random variables we have:

$$I(\{X_i\}Z) = H(\{X_i\}) - H(\{X_i\}|Z) = H(Z) - H(Z|\{X_i\}) = I(Z\{X_i\}) \quad (6)$$

$$n - H(\{X_i\}|Z) = H(Z) \quad (7)$$

$$H(\{X_i\}|Z) = n - H(Z) \quad (8)$$

Inserting this into (5) we have an expression for R:

$$R = n - H(Z) - \sum_{i=1}^{n} H(X_i|Z) \quad (9)$$

or

$$R = \sum_{i=1}^{n} I(ZX_i) - I(Z\{X_i\}) \quad (10)$$

$$= \sum_{i=1}^{n} [H(Z) - H(Z|X_i)] - [H(Z) - H(Z|\{X_i\})] \quad (11)$$

$$= nH(Z) - \sum_{i=1}^{n} H(Z|X_i)] - H(Z) \quad (12)$$

$$R = (n-1)H(Z) - \sum_{i=1}^{n} H(Z|X_i) \quad (13)$$

From this expression we can see that R is minimized (synergy maximized) when knowledge of each input state X_i does not reduce the entropy of the output state Z significantly.

If we once again think of the arragements as boolean circuits with n independent inputs, we can again compute, for each logic function, the corresponding values of H(Z), $I(X_iX_j)$'s (mutual information between the different time series), and

R. And once again we see that different boolean functions have the same R value. This can be understood by rewriting R slightly:

$$R = (n-1)H(Z) - \sum_{i=1}^{n} H(Z|X_i) \tag{14}$$

$$= (n-1)H(P) - \sum_{i=1}^{n} [H(Z|X_i = 1) + H(Z|X_i = 0)] \tag{15}$$

$$R = (n-1)H(P) - \sum_{i=1}^{n} \frac{1}{2}[H(P_i) + H(2P - P_i)] \tag{16}$$

where $P_i = P(Z = 1|X_i = 1)$. We see that R depends only on n+1 variables: $P = k/2^n$ and $P_i = k_i/2^(n-1)$. Since different logic functions correspond to the same set P, P_i, they will have the same R value (example?).

Furthermore, we can see (how?) that the highest synergy value for any P corresponds to the case where $P = P_i$ for all i, in which case,

R = -H(P) = -H(Z)

This makes sense as the greatest reduction of uncertainty in the state of Z is the negative of its entropy. For example, if P = 1/2, then $P_i = 1/2$ will give the highest synergy of -H(1/2) = -1. Here are the six possible logic tables for n = 3 with R = -1:

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	1	1	0	0	0	0	1	1

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	0	1	1	0	0	1	1	0

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	0	0	1	1	1	1	0	0

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	1	0	1	0	0	1	0	1

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	1	0	0	1	1	0	0	1

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	0	1	0	1	1	0	1	0

P defines how often Z is on and therefore a number of possible divisions of these spiking events among the times that any of the inputs X_i are on. We see that R is minimized when each input is spiking during the same amount of time of spiking of the output Z. This is interesting because the concept of synergy can be applied in other situations where there is a number of agents X_i each having some information about the state of some agent Z ($I(X_iZ) \neq 0$). These agents' information is assumed to be independent of each other ($I(X_iX_j) = 0$); that is, they do no take into account each others opinions. The question we are interested in is then, what is the best way to aggregate their information or opinions to make the most informed guess about the state of Z? In this binary state case, it turns out that we weigh each agent's information equally. That is, we guess that Z is on P_i fraction of the time any of the agents are on. (more?)

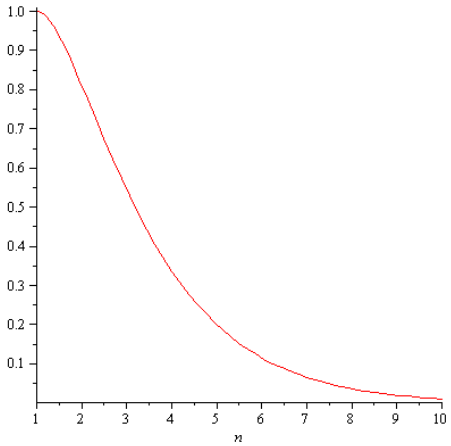
It is also interesting to see how the entropy of H scales with n for some function. If k is the number of states for which the ouput is on (defines the class of functions),

$$H(X) = -\sum_x P(x) ln_2(P(x)) \tag{17}$$

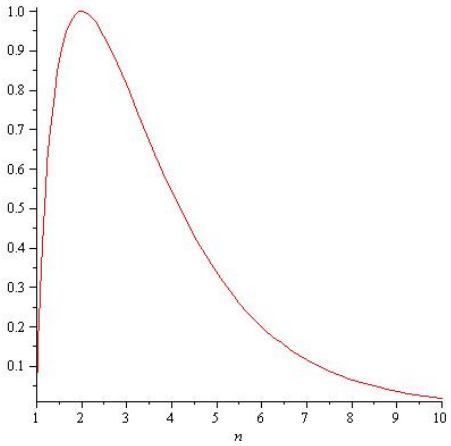
$$= \frac{k}{2^n} ln(\frac{k}{2^n}) + \frac{2^n - k}{2^n} ln(\frac{2^n - k}{2^n}) \tag{18}$$

$$H(n,k) = n - \frac{1}{2^n}[kln(k) + (2^n - k)ln(2^n - k)] \tag{19}$$

So for example, for the AND function P = $1/2^n$:



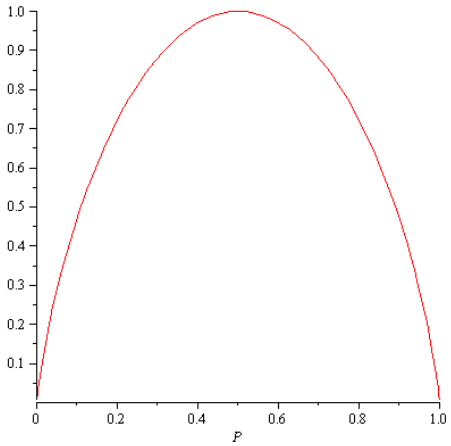
For some other function where k = 2 -> P = $2/2^n$:



We see that although the peak for which H is greatest (1) is different in each case ($k/2^n = 1/2 -> k = 2(n-1)$), the drop off with n is the same in each case. We can simplify this by writing H as a functions of $k/2^n = P$:

$$H(n,k) = H(\frac{k}{2^n}) = H(P) = -Pln(P) - (1-P)ln(1-P) \qquad (20)$$

This is of course a Bernoulli distribution which peaks at 1 for P = 1/2:



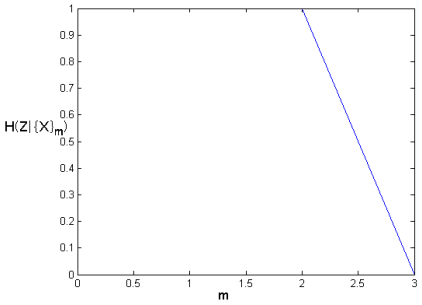
So we can say that H is independent of both n and k as long as $k/2^n = P$ is constant.

The entropy drops off very quickly with n for functions which are very selective to the input. This simply means that the cell is barely spiking because it is being very selective (or it is relaying information from another selective cell).

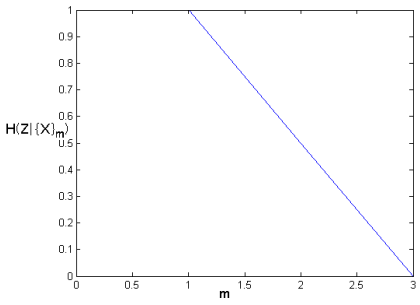
If it interesting to see if $Z = f(X_i)$, how knowledge of the state of the inputs reduces the entropy of state Z. That is, how does the quantity $H(Z|X_m)$ scale with m? We try this for several boolean functions (with the same P). For a boolean circuit represented by the table:

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	1	1	0	0	0	0	1	1

the R value is is -1. What we see when we calculate $H(Z|X_m)$ is that the decrease in entropy with m depends on which input states X_i (or combination of) are known first. So we have several possible plots for this function:



X_1, X_3 or X_2, X_3 known first.

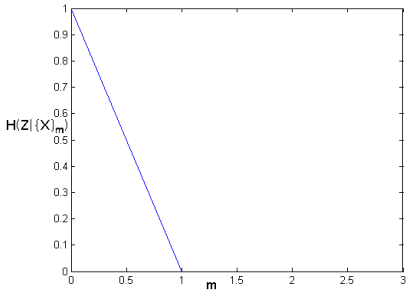


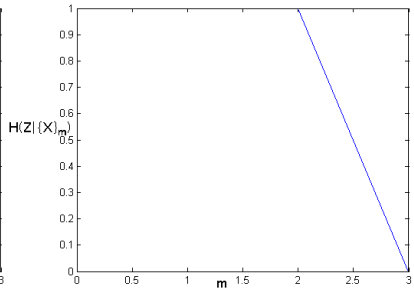
X_1, X_2 known first.

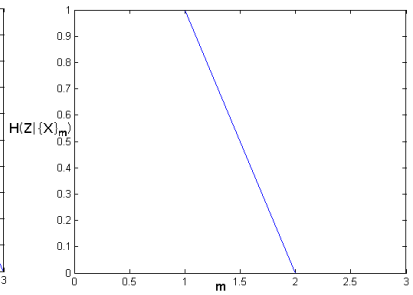
This is true for any logic function. Here are the plots for the logic functions represented by the tables:

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	0	0	0	1	0	1	1	1

(R = 0)



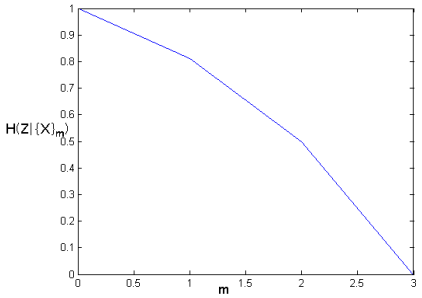




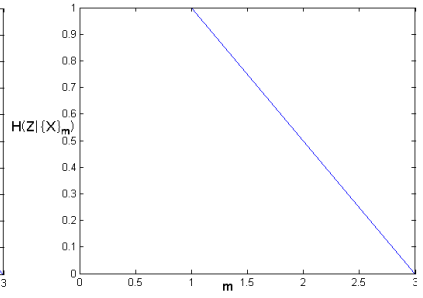
‘ In order: X_1 known first, last, or second.

X1	0	0	0	1	0	1	1	1
X2	0	0	1	0	1	0	1	1
X3	0	1	0	0	1	1	0	1
Z	0	0	1	0	0	1	1	1

(R = -.623)



X_1 known first.



X_2 or X_3 known first.

But we do take note that the drop off in entropy increases with each m... Conclusion?

2 Hierarchical composition of circuits and information processing

Here we will discuss how these entropic quantities scale in a "'hierarchical"' type of circuit where the cells are arranged in a branching tree with some branching ratio n. The reason this type of circuit is interesting is because it may capture some of the properties of the hierarchically arranged visual system in which the input image goes through a series of (eight) alternating layers of "'simple"' and "'complex"' cells which process the incoming data from the previous layer in some way [3]. The response of the simple cells is modeled by:

$$y_s = e^{\frac{-1}{2\sigma^2}\sum_j (w_j - x_j)^2} \tag{21}$$

where x_j is the signal from the jth cell in the previous layer and w_j is some weight (in this case a preferred orientation). And the complex cells' response is given by:

$$y_c = max(x_j) \tag{22}$$

In order to apply our analysis and to simplify the model we change these functions to logic gates. In the response of the simple cells:

$y_s = e^{\frac{-1}{2\sigma^2}\sum_j (w_j - x_j)^2}$

$= \prod e^{\frac{-1}{2\sigma^2}(w_j - x_j)^2}$

we see that the response is strong only when all of the x_j are approximately w_j. This can be thought of as an AND function, which is very selective in its input.

Similarly, the response of the complex cells can be modeled with an OR circuit which will respond as long as there is an input signal (not selective).

How does entropy and synergy scale at each level L for some branching ratio n? To answer this question we need to generalize our formula for R for the case when the individual inputs X_i are not on for half the time but for some variable fraction $P(X_i = 1) = \alpha_i$:

$$R = (n-1)H(Z) - \sum_{i=1}^{n} H(Z|X_i) \tag{23}$$

$$= (n-1)H(P) - \sum_{i=1}^{n}[H(Z|X_i = 1) + H(Z|X_i = 0)] \tag{24}$$

$$= (n-1)H(P) - \sum_{i=1}^{n}[(-P_i ln(\frac{P_i}{\alpha_i}) - (\alpha_i - P_i)ln(\frac{\alpha_i - P_i}{\alpha_i})) +$$

(- (P - P_i)$ln(\frac{P-P_i}{1-\alpha_i}) - (1 - \alpha_i - (\alpha_i - P_i))ln(\frac{1-\alpha_i-(\alpha_i-P_i)}{1-\alpha_i}))]$(25)

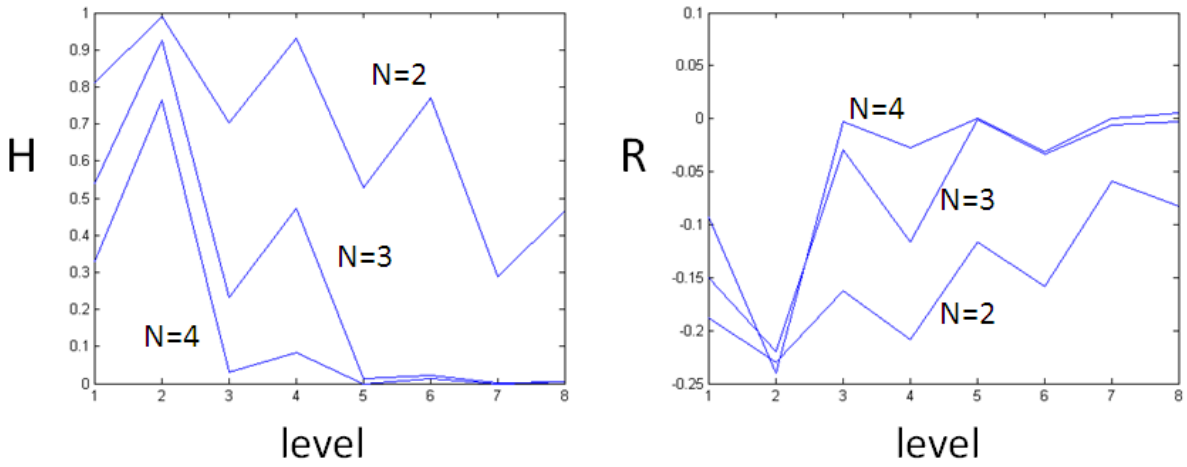
$$= (n-1)H(P) - \sum_{i=1}^{n}[\alpha_i H(P_i) + (1-\alpha_i)H(\frac{P - \alpha_i P_i}{1-\alpha_i})] \tag{26}$$

In our case the α_i are all equal at any one level, but vary as we go down each level. So we rename α_i as α_L. α_L is determined from how often the AND and OR functions respond to n independent inputs each on some fraction of the time:

$\alpha_1 = 1/2$ (initially white noise)
$\alpha_2 = (\alpha_1)^n$
$\alpha_3 = 1 - (1 - \alpha_2)^n$
$\alpha_4 = (\alpha_3)^n$
$\alpha_5 = 1 - (1 - \alpha_4)^n$
$\alpha_6 = (\alpha_5)^n$
$\alpha_7 = 1 - (1 - \alpha_6)^n$

$\alpha_8 = (\alpha_7)^n$

With this we then compute R and H for several branching ratios n:



We see that even for very low branching ratios n (typical n in the visual system is 10^4) the entropy drops off very quickly (although it rises at each OR level). What this simply means is that the system filters out or does not repond to noise. It would now be interesting to see how this system responds to some sort of image...

Also, it may be possible to make the model more exact by working with the actual analytical functions rather than their boolean simplifications. It might also be interesting to carry out the four cell analysis on the entire network of cells just as in [1] to see if there are any significant differences with the three cell analysis...

References

[1] M. I. Ham L. M. A. Bettencourt, G. J. Stephens and G. W. Gross. Functional structure of cortical neuronal networks grown in vitro. *Phys. Rev. E*, 75(021915), 2007.

[2] E. Schneidman, S. Still, Michael J. Berry II, and W. Bialek. *Phys. Rev. Lett.*, 91 (238701), 2003.

[3] T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. 2006.