Deriving a Hopfield Network from The Bayesian Perspective

Akiva Lipshitz

June 28, 2017

Of the many mathematical entities I never understood nor could I foresee the day when I would understand them is Hopfield Networks. Hopfield Networks are models of contenta ddressable memory. In other words, they are dynamical systems designed to encode N arbitrary strings and then converge to a point of perfect recall of any one of these strings if stimulated with a noisy rendition. This article derives the Hopfield Network from scratch and studies its dynamical properties. Hopfield networks are useful to study for educational experience. Not so much is understanding the particular implementation level details of hopfield networks as important as appreciating the perspectives and analytical decisions made in the process.

Deriving a Dynamical System

Implementing *content addressable memory* as a dynamical systems necessitates an ansatz equation, i.e. some initial symbolic structure to play around with. (As much as the field would like its neuronal models to be structure free, the only way to get there is by making simplifying assumptions and introducing structure.).

Let there be N unique binary strings of length K indexed by μ, $\mathbf{x}^\mu$, which can be observed with some bits randomly flipped, i.e. signal noise. Such observations are initially sampled from a uniform distribution over binary strings and then this string is permuted by sampling from a noise distribution. Observations will be denoted by the random variable $\mathbf{X}$. In particular, a single bit in random observation X_j is kept with probability $1-p$ and flipped with probability p. In particular, let the permutative process be interpreted as a random variable ζ defined over the binary sample space $\Omega = \{-1, 1\}$ (The author originally tried $\{0, 1\}$ as a sample space but came to a point where this definition made some terms very difficult to deal with analytically. As such, the paper will follow the convention of using the range of the sgn function as its binary digits).Now,

$$\zeta_j \sim \text{Bernoulli}(p)$$

such that

$$X_j = x_j \zeta_j$$

It follows that it is in fact the case $X_j = x_j$ with probability $1-p$ and $X_j = -x_j$ with probability p.

Henceforth will be derived a dynamical system to converge upon the most likely true signal given some noisy observation of it, as defined in (1). Specifically, the system will be represented by T units, each denoted by z_j which evolve over time. These units in the system are initialized with a random sample from the observation distribution.

$$z_j(0) \sim \text{Bernoulli}(p)$$

The network's prediction $\mathbf{z}$ of the true denoised signal $\mathbf{x}$ can be optimized by maximizing the likelihood that it is in fact the true signal. This means solving the program

$$\arg\max_{\mathbf{z_j}} \mathcal{P}(\mathbf{z}_j(0))$$

First, we define a uniform distribution over the set of true signals.

$$\mathcal{P}(\mathbf{x}) = \frac{1}{N}$$

Next, we write a distribution for the likelihood of observing any noised signal given the true signal.

$$\mathcal{P}(\tilde{\mathbf{x}} \mid \mathbf{x}) = \alpha^{H_\mu} \beta^{T-H_\mu}$$

where $\alpha = \frac{p}{1-p}$, $\beta = 1-p$ and H_μ is the number of pairwise different bits between the μth true signal $\mathbf{x}^\mu$ and the current estimate of the true signal z_j,

$$H_\mu = \sum_{j=1}^{T} x_j \wedge z_j$$

in which $a \wedge b$ is the logical EQUALITY operator. Equation (7) is merely a symbolic abstraction and is now rewritten explicitly.

$$H_\mu = \frac{1}{2}\left[N - \sum_{j=1}^{T} z_j x_j^\mu\right] = \frac{1}{2}\left[N - \mathbf{z}\cdot\mathbf{x}^\mu\right]$$

It is beneficial to take a moment to understand equation (8). If $z_j = x_j$, i.e. the network's prediction matches the true value, then $z_j x_j = 1$. Otherwise, $z_j x_j = -1$. Thus,

$$\sum x_j z_j = C - I$$

where C is the number of correct guesses and I is the number of incorrect guesses. We would like only the number of incorrect guesses, i.e. $H^\mu = I$. Observe that $C = N - I$, such that

$$\sum x_j z_j = N - 2I$$

Thus

$$\frac{1}{2}\left[N - \sum x_j z_j = I\right]$$

The above two distributions for $\mathcal{P}(\mathbf{x})$ and $\mathcal{P}(\mathbf{z}(\mathbf{0}) \mid \mathbf{x})$ can be combined with Bayes rule to write an expression for total probability of any initial state $z_j(0)$. While it may be advantageous to do so, for the sake of simplicity, the present discussion will not discuss the case of adding a bayesian prior.

$$\mathcal{P}(\mathbf{z}(0)) = \sum_{\mu=1}^{N} \mathcal{P}(\mathbf{z}(0) \mid \mathbf{x}^\mu)\mathcal{P}(\mathbf{x}^\mu)$$

$$\mathcal{P}(\mathbf{z}(0)) = \frac{\beta^T}{N}\sum_{\mu=1}^{N} \exp\left\{H_\mu \ln\alpha\right\}$$

By maximizing (10), we arrive at an optimal prediction $\mathbf{z}$.

$$\frac{\partial \mathcal{P}(z_j)}{\partial z_j} = \frac{\beta^T}{N}\ln\alpha \sum_{\mu=1}^{N} \exp\left\{H_\mu \ln\alpha\right\}\frac{\partial H_\mu}{\partial z_j}$$

From (6),

$$\frac{\partial H_\mu}{\partial z_j} = -\frac{1}{2}x_j^\mu$$

Noting that z_j is bounded to the closed interval $[-1, 1]$, we could use Lagrangian optimization to solve for

$$\frac{\partial \mathcal{P}(z_j)}{\partial z_j} = 0$$

Keeping z_j in a closed interval is equivalent to saying $\|z_j| = \sqrt{z_j^2} = 1$. Thus,

$$\nabla z_j = \lambda \frac{z_j}{|z_j|}$$

Because of the absolute value in the denominator, this will be really difficult to solve analytically. Do not fear. Due to the very simple bounds on z_j,gradient ascent is a viable solution but only if we place a nonlinear filter over each update to keep z_j in line. The reason for this is because the learning equations were derived with the assumption that the range is bounded; we must therefore enforce this constraint in our updates because gradient descent cannot read our minds. Furthermore, it will not be necessary to perform a numerical integration $z_j := z_j + \Delta z_j$ because we care only about the sign of z_j and nothing more. As such, whether $\Delta z_j > 0$ or $\Delta z_j < 0$ is sufficient for our interests.

$$z_j := \tanh \left\{ \kappa \sum_{\mu=1}^{N} x_j^\mu \alpha^{H_\mu} \right\}$$

where κ is a chosen rate of convergence. If updates are to be discrete then the rule becomes very similar to the true hopfield update rule

$$z_j := \operatorname{sgn} \left\{ \sum_{\mu=1}^{N} x_j^\mu \alpha^{H_\mu} \right\}$$

We have arrived at something very similar to the real Hopfield update rule by starting with maximum likelihood concerns. This shows that real Hopfield networks not only converge upon a solution, but onto a statistcally optimal one.

Analysis of The Hopfield Dynamical System

As a sanity check, we can confirm that equation (14) does in fact converge upon the true value by working backwards. Assume $z_j \approx x_j^w = 1$. Then

$$\sum_{\mu=1}^{N} x_j^\mu \alpha^{H_\mu} \approx x_j^w$$

For this to happen, it must be that

$$x_j^w a^{H_w} > \sum_{\mu \neq w} x_j^\mu \alpha^{H_\mu}$$

This is only the case if $0 < \alpha \ll 1$. Given that $\alpha = \frac{p}{1-p}$, this requires $p \ll 1 - p$, or that the signal to noise ratio is very high.

Conclusion

We have shown that a Hopfield-like system emerges quite naturally when you ask the same question that Hopfield did and take a maximum likelihood approach to answering it. where κ is the arbitrarily chosen rate of convergence. Eventually this system will converge on the denoised signal.

I usually post my blog articles before they are finished as an incentive for me to finish them. come back later and I'll do more in depth analysis of the stability and nuance in the Hopfield network we just derived

Bibliography

Hancock, Edwin R., and Josef Kittler. "A Bayesian interpretation for the Hopfield network." *Neural Networks, 1993., IEEE International Conference on.* IEEE, 1993.