Self Organizing Hierarchical Kernel Density Estimation

Spontaneous Representation Learning

Nathan Crock Gordon Erlebacher
Department of Scientific Computing, Florida State University
Computational Intelligence Lab

Abstract

All observed data are samples produced from unknown stochastic processes. Determining the underlying distributions giving rise to these observations is an important task in many disciplines such as machine learning and Bayesian non-parametrics. Often times, these distributions are high in dimensionality, multimodal and complex with discontinuities such as jumps and edges. It is difficult for traditional parametric models to capture the structure of these distributions. Non-parametric techniques such as Kernel Density Estimators (KDE) provide more expressive power, though still suffer from the curse of dimensionality. Here, we demonstrate a non-parametric kernel-based method that side-steps the curse of dimensionality by approximating marginals of the true distribution and combining them hierarchically to reconstruct the full joint. We start by showing how a local learning rule from neuroscience called Hebbian Plasticity minimizes the reverse Kullback-Leibler divergence between a single kernel and an unknown distribution. This leads to each kernel finding modes in the marginal distributions. We then show how subsequent kernels listening to input from the approximate marginals learn to approximate the joint on a lower dimensional manifold.

An Upper Bound for the Reverse Kullback Leibler Divergence

The Kullback-Leibler Divergence (KL) is often used to measure dissimilarity between two distributions Q and P [1].

$$\mathrm{KL}(Q||P) = \mathbb{E}_Q\left[\log \frac{q(x)}{p(x)}\right]$$

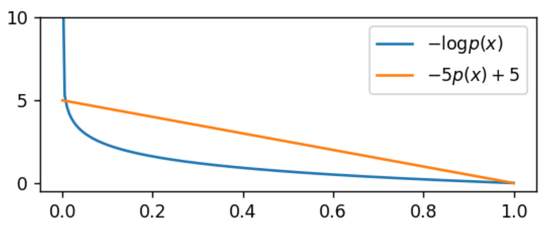
where $q(x)$ and $p(x)$ are the densities of Q and P respectively. Minimizing with respect to $q(x)$ results in mode-seeking behavior. However, if the density $p(x)$ is unknown then minimization cannot be performed. To address this we devise an upper bound that removes the density $p(x)$

$$\mathrm{KL} = \int q(x)(-\log p(x))dx + \int q(x)\log q(x)$$

The term $p(x)$ only appears in the cross entropy term so we use a straight line to mostly bound $-\log p(x)$ from above

$$-\log p(x) \leq \frac{\log \lambda}{1-\lambda}\big(p(x)-1\big)$$

Which holds for $\lambda \leq p(x) \leq 1$. For example the upper bound is shown below for $\lambda \approx 0.007$.



Substituting the upper bound for $-\log p(x)$ back into the KL we see that the unknown density $p(x)$ has been removed from within the expectation:

$$\begin{aligned}\mathrm{KL}(Q||P) &= \mathbb{E}_Q\big[-\log p(x)\big] - \mathbb{E}_Q\big[-\log q(x)\big] \\ &\leq \mathbb{E}_Q\big[k\big(p(x)-1\big)\big] - H[Q] \\ &\leq \int q(x)k\big(p(x)-1\big)dx - H[Q] \\ &\leq k\left(\int p(x)q(x)dx - 1\right) - H[Q] \\ &\leq k\big(\mathbb{E}_P\big[q(x)\big]-1\big) - H[Q]\end{aligned}$$

where $H[Q]$ is the entropy of Q and $k = \frac{\log\lambda}{1-\lambda}$. To minimize this upper bound, we no longer need to know the density $p(x)$, we need only be able to sample from the distribution $P(x)$.

Minimizing the Upper Bound

Assume the unknown distribution P is the universe. It constantly generates stimuli in the form of photons, molecules, pressure waves, and more. In other words, we don't know $p(x)$ but we can sample from P, or more precisely P constantly generates its own samples. We next assume the input and output of a neuron can be modeled by a Gaussian. The mean $\boldsymbol{\mu}$ of the Gaussian are the weights of the neuron and an isotropic covariance $I\boldsymbol{\sigma}^2$ is the bias or soft firing threshold. With these assumptions, we show that minimizing the upper bound results in a learning rule that is functionally equivalent to Hebbian Plasticity.

Let $q(\boldsymbol{x}|\theta)$ be the pdf for $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ where $\theta = \{\boldsymbol{\mu}, \Sigma\}$, $\boldsymbol{x}, \boldsymbol{\mu} \in \mathbb{R}^k$ and $\Sigma \in \mathbb{R}^{k\times k}$. With $q(x)$ parameterized, the upper bound can be expressed as a function $f(k,\theta)$ because both expectations are over x

$$f(k,\theta) = k\big(\mathbb{E}_p\big[q(\boldsymbol{x}|\theta)\big] - 1\big) - \frac{1}{2}\log|2\pi e\Sigma|$$

Where we used the analytic form of the differential entropy for a multivariate Guassian distribution.

Stochastic Optimization

We now proceed with traditional optimization techniques by setting the gradient with respect to θ equal to zero. Here we use the pathwise derivative (PD) to calculate the gradient of the expectation. This amounts to moving the gradient inside the integral and thus inside the expectations. This procedure is always valid provided the argument of the expectation is continuous and everywhere differentiable.

$$\nabla_\theta f(k,\theta) = k\mathbb{E}_p\big[\nabla_\theta q(x|\theta)\big] - \big(0, \frac{1}{2}\Sigma^{-1}\big)^T$$

To allow more transparent analysis we express the gradient in terms of its components

$$\begin{aligned}\frac{\partial f}{\boldsymbol{\mu}} &= k\mathbb{E}_p\big[\frac{\partial q(x|\boldsymbol{\mu},\Sigma)}{\partial \boldsymbol{\mu}}\big] \\ \frac{\partial f}{\partial \Sigma} &= k\mathbb{E}_p\big[\frac{\partial q(x|\boldsymbol{\mu},\Sigma)}{\partial \Sigma}\big] - \frac{1}{2}\Sigma^{-1}\end{aligned}$$

Substituting the known partials for the multivariate Gaussian back into our gradient terms we arrive at

$$\begin{aligned}\frac{\partial f}{\boldsymbol{\mu}} &= k\mathbb{E}_p\big[q(x|\theta)\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\big] \\ \frac{\partial f}{\partial \Sigma} &= k\mathbb{E}_p\big[-q(x|\theta)\frac{1}{2}(\Sigma^{-1} - \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1})\big] - \frac{1}{2}\Sigma^{-1}\end{aligned}$$

We now employ the Robbins-Monroe algorithm to determine the optimal parameters for the Gaussian $q(x)$ [2]. To ease notation, let $q_n = q(x_n|\boldsymbol{\mu}_n, \Sigma_n)$ and $C_n = (\boldsymbol{x}_n - \boldsymbol{\mu}_n)(\boldsymbol{x}_n - \boldsymbol{\mu}_n)^T$ which is the empirical Fisher matrix that approximates the Hessian giving us second order information in our update. After observing the n^{th} sample from p we update the Gaussian as follows.

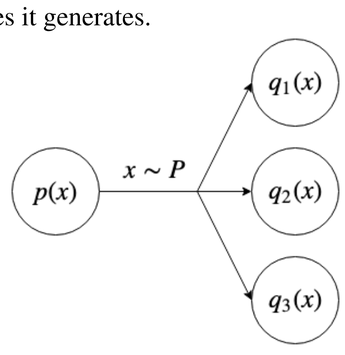
$$\begin{aligned}\boldsymbol{\mu}_{n+1} &= \boldsymbol{\mu}_n + \alpha k q_n \Sigma_n^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_n) \\ \Sigma_{n+1} &= \Sigma_n + \beta\big[kq_n\big(\Sigma_n^{-1}C_n\Sigma_n^{-1} - \Sigma_n^{-1}\big) + \Sigma^{-1}\big]\end{aligned}$$

Let us compare the update rule in a single dimension to the traditional Hebbian Learning rule. Where we choose $\alpha = \sigma^3$ and $\beta = \sigma^4$.

$$\begin{aligned}\boldsymbol{\mu}_{n+1} &= \boldsymbol{\mu}_n + k\sigma_n q_n(\boldsymbol{x}_n - \boldsymbol{\mu}_n) \\ \sigma^2_{n+1} &= \sigma^2_n + k\sigma_n\frac{q_n}{2}\Big[\big(\boldsymbol{x}_n - \boldsymbol{\mu}_n\big)^2 - \sigma^2_n\Big] + \sigma^3_n\end{aligned}$$

Example in 1D

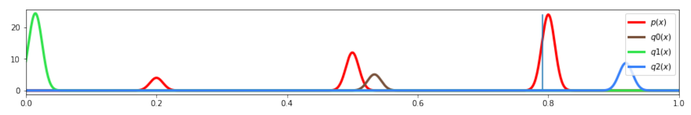
To demonstrate the learning rule we create an artificial distribution P in one dimension and allow three neurons to listen to the random samples it generates.



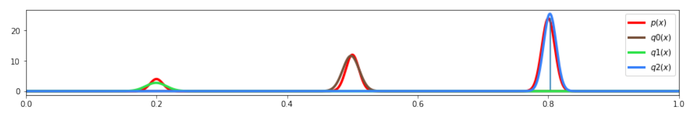
We visualize the initial state of the randomly configured neurons and the final state after observing many sample from P. We choose P to be a three mode Gaussian mixture model all with equal variance $\sigma^2 = 0.001$

$$p(x) = \frac{1}{10}\mathcal{N}(0.2,\sigma^2) + \frac{3}{10}\mathcal{N}(0.5,\sigma^2) + \frac{6}{10}\mathcal{N}(0.8,\sigma^2)$$

We randomly initialize the three Gaussian neurons q_i with different means and show the initial state below.



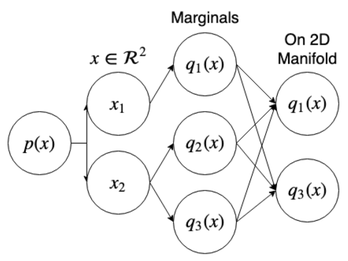
The blue vertical line is a sample x drawn from the P distribution. It is presented to each neuron and their weights and threshold are updated according to the update rule derived above. As we showed, this update rules stochastically minimizes the reverse Kullback-Leibler divergence leading to mode-seeking behavior.



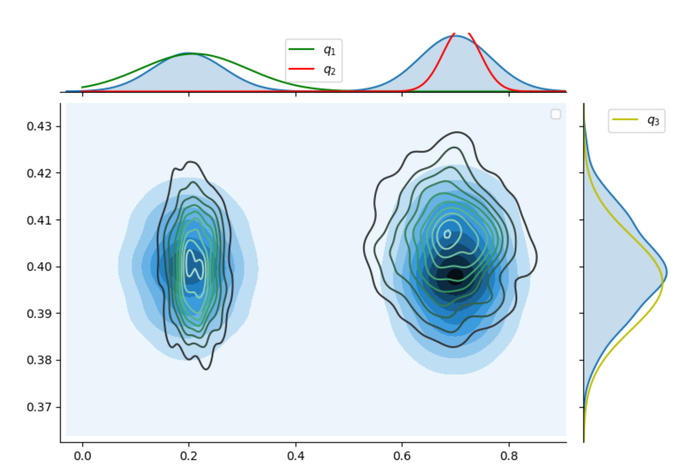
As we see above, each neuron converged to a mode or local extrema of the data generating distribution's pdf $p(x)$. In this way, the neuron has become a detector for how much of that feature is present in any given observation.

Higher Dimensions

In higher dimensions we see that the hierarchical nature of neural networks allows us to approximate more complex distributions. In particular the first layer will approximate lower dimensional marginals.



Subsequent layers project into higher dimensions and learn to approximate the true distribution on a lower dimensional manifold



References

[1] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[2] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.