On the DeepFRAME model

Erik Nijkamp

Abstract

The FRAME (Filters, Random Fields and Maximum Entropy) model [Zhu et al., 1997] is a generative model for texture patterns. It is an energy-based model in the form of a Gibbs distribution (or Markov random field) where the energy function is the sum of non-linear potential functions of linear filter responses. We review the DeepFRAME [Xie et al., 2016] as a generalization where the energy function is defined by a convolutional neural network. The local energy minima satisfy a Hopfield auto-encoder. The model can be learned by an analysis-by-synthesis scheme in the form of contrastive divergence.

1 FRAME

Let $X(x)$ denote an image on the domain $\mathcal{D}$ where $x = (x_1, x_2)$. Let $\{F_k, k = 1, \dots, K\}$ denote a set of filters such as Gabor. Let $F_k * X$ denote a convolution and $[F_k * X](x)$ be the filter response at position x. The *stationary* FRAME model for texture patterns is in the form of a Markov random field or Gibbs distribution

$$p(X; \lambda) = \frac{1}{Z(\lambda)} \exp \left[\sum_{k=1}^{K} \sum_{x \in \mathcal{D}} \lambda_k([F_k * X](x)) \right], \tag{1}$$

where $\lambda_k()$ is a non-linear potential function to be estimated from the training data and $Z(\lambda)$ is the intractable partition function. The energy function of (1) is

$$\mathcal{E}(X; \lambda) = -\sum_{k=1}^{K} \sum_{x \in \mathcal{D}} \lambda_k([F_k * X](x)). \tag{2}$$

Note, (1) is stationary because the function $\lambda_k()$ does not depend on x. The *non-stationary* FRAME model for object patterns is of the form

$$p(X;\lambda) = \frac{1}{Z(\lambda)} \exp\left[\sum_{k=1}^{K}\sum_{x\in\mathcal{D}} \lambda_{k,x}([F_k * X](x))\right] q(X), \tag{3}$$

where the potential $\lambda_{k,x}()$ does depend on x and $q(I)$ is a reference distribution such as Gaussian (or Uniform)

$$q(X) = \frac{1}{(2\pi\sigma^2)^{|\mathcal{D}|/2}} \exp\left[-\frac{1}{\sigma^2}||X||^2\right]. \tag{4}$$

We may parameterize $\lambda_{k,x}(r) = w_{k,x}h(r)$ with rectified linear unit (ReLU) $h(r) = max(0,r)$ and estimate $w_{k,x}$. We can absorb $q(X)$ into the energy function of (3)

$$\mathcal{E}(X;\lambda) = -\sum_{k=1}^{K}\sum_{x\in\mathcal{D}} \lambda_k([F_k * X](x)) + \frac{1}{\sigma^2}||X||^2. \tag{5}$$

The FRAME model can be justified by the maximum entropy principle. Let $f(X)$ denote the true distribution from which we sample observations $\{I_m\}$. Let w^* solve the maximum likelihood equation

$$E_w([F_k * X](x)) = E_f([F_k * X](x)), \forall k, x. \tag{6}$$

Let Ω denote a set of distributions which share the statistics of f as captured by $\{F_k\}$:

$$\Omega = \{p : E_p([F_k * X](x)) = E_f([F_k * X](x)), \forall k, x\}. \tag{7}$$

Then it can be shown that $p(X;w^*)$ among all $p \in \Omega$ minimally modifies the reference q to match the statistics of f

$$p(X;w^*) = \arg\min_{p\in\Omega} KL(p||q). \tag{8}$$

If q is uniform, then $p(X;w^*)$ achieves maximum entropy among all $p \in \Omega$. If q is Gaussian, then we can absorb $||I||^2$ into the energy function and consider the model as a uniform measure with $||I||^2$ as additional feature. The maximum entropy interpretation still holds.

2 DeepFRAME

The FRAME model uses a set of pre-defined filters $\{F_k\}$. Instead of relying on pre-defined filters, the DeepFRAME [Xie et al., 2016] model learns the linear filters and non-linear potentials. It is a deep energy-based convolutional model in the form of a Gibbs distribution

$$p(X; w) = \frac{1}{Z(w)} \exp\left[f(X; w)\right] q(X), \tag{9}$$

where the scoring function $f_w : X \to \mathcal{R}$ is defined by a convolutional neural network with weights w and $q(X)$ is the reference distribution such as Gaussian $q(X) \propto \exp(-||I||^2/2\sigma^2)$. The partition function $Z(w) = \int \exp\left[f(X; w)\right] q(X) dX$ is intractable. The energy function is

$$\mathcal{E}(X; w) = -f(X; w) + \frac{1}{2\sigma^2}||X||^2. \tag{10}$$

The model $p(X; w)$ can be understood as exponential tilting of $q(X)$. In the trivial case of $f(X; w) = wX$, the effect of exponential tilting on $q(X) \sim \mathcal{N}(\mu, \sigma^2)$ reduces to mean shifting $p(X; w) \sim \mathcal{N}(\mu + w\sigma^2, \sigma^2)$. If $f(X; w)$ is a convolutional neural network with piecewise linear rectification (ReLU), then $f(X; w)$ is piecewise linear in X and $p(X; w)$ is a piecewise mean shifted truncated Gaussian. The local energy minima of (10) satisfy a Hopfield auto-encoder

$$\frac{X}{\sigma^2} = \frac{\partial f(X; w)}{\partial X}. \tag{11}$$

3 Maximum Likelihood

Suppose we observe training examples $\{X_i, i = 1, \ldots, n\}$ form an unknown distribution. We can estimate the weight parameters w by maximizing the log-likelihood

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} \log p(X_i; w). \tag{12}$$

Since we want to maximize the likelihood $L(w)$, we will derive the gradient by firstly writing down the partial derivative of (32)

$$\frac{\partial L(w)}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w}\log\left(\frac{1}{Z(w)}\exp\left[f(X_i;w)\right]q(X_i)\right) \tag{13}$$

$$\propto \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w}\left(f(X_i;w) - \log Z(w) + \frac{1}{2\sigma^2}||X_i||^2\right) \tag{14}$$

$$\propto \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w}f(X_i;w) - \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w}\log Z(w) \tag{15}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w}f(X_i;w) - E_w(\frac{\partial}{\partial w}f(X;w)). \tag{16}$$

The key to the equality (15) is the identity

$$\frac{\partial \log Z(w)}{\partial w} = \frac{1}{Z(w)}\frac{\partial Z(w)}{\partial w} \tag{17}$$

$$= \frac{1}{Z(w)}\frac{\partial}{\partial w}\int \exp(f(X;w))dX \tag{18}$$

$$= \frac{1}{Z(w)}\int \frac{\partial \exp(f(X;w))}{\partial w}dX \tag{19}$$

$$= \frac{1}{Z(w)}\int \exp(f(X;w))\frac{\partial f(X;w)}{\partial w}dX \tag{20}$$

$$= \int p(X;w)\frac{\partial f(X;w)}{\partial w}dX \tag{21}$$

$$= E_w(\frac{\partial}{\partial w}f(X;w)). \tag{22}$$

This integration in (21) is intractable. However, in the form of (22), we can see that the expectation can be numerically approximated by drawing samples from the proposed distribution $p(X;w)$. However, we can not sample from $p(X;w)$ directly as we can not evaluate the partition function $Z(w)$. But we may use many cycles of Markov Chain Monte Carlo (MCMC) sampling to transform our data distribution (drawn from the target distribution) into the proposed distribution. We may perform MCMC sampling in the form of a Langevin

dynamics (or Gibbs sampling), which itereates the following steps:

$$X_{\tau+1} = X_\tau - \frac{\delta^2}{2}\frac{\partial}{\partial X}\mathcal{E}(X_\tau; w) + \delta U_\tau \tag{23}$$

$$= X_\tau - \frac{\delta^2}{2}\left[\frac{X_\tau}{\sigma^2} - \frac{\partial}{\partial X} f(X_\tau; w)\right] + \delta U_\tau. \tag{24}$$

where τ indexes the time step, δ is the step size, and $U_\tau \sim \mathcal{N}(0, I)$ is Gaussian white noise. The Langevin dynamics can be understood as a stochastic Ornstein-Uhlenbeck process where the gradient term $\frac{\partial}{\partial X} f(X_\tau; w)$ is attracting the process towards low energy regions and the Wiener process U_τ injects randomness to overcome shallow modes.

We may run $\tilde{n}$ parallel Markov chains of Langevin dynamics according to (24) to obtain the synthesized examples $\{\tilde{X}_i, i = 1, \ldots, \tilde{n}\}$. Then the Monte Carlo approximation of (16) is

$$\frac{\partial}{\partial w} L(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial w} f(X_i; w) - E_w\left(\frac{\partial}{\partial w} f(X; w)\right) \tag{25}$$

$$\approx \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial w} f(X_i; w) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \frac{\partial}{\partial w} f(\tilde{X}_i; w) \tag{26}$$

$$= \frac{\partial}{\partial w}\left[\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \mathcal{E}(\tilde{X}_i; w) - \frac{1}{n}\sum_{i=1}^{n} \mathcal{E}(X_i; w)\right]. \tag{27}$$

We compute w by stochastic gradient ascent to maximize $L(w)$

$$w_{\tau+1} = w_\tau + \gamma \frac{\partial}{\partial w} L(w) \tag{28}$$

$$\approx w_\tau + \gamma\left(\frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial w} f(X_i; w) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \frac{\partial}{\partial w} f(\tilde{X}_i; w)\right) \tag{29}$$

where γ is the step size.

Both learning in (29) and sampling in (24) involve the gradients of $f(X; w)$ with respect to w and X, respectively. The derivatives can be computed efficiently in the traditional backpropagation and both derivatives share most of the chain rule.

4 Contrastive Divergence

Consider a Gibbs distribution with parameters w in the form of

$$p_w(X) = \frac{1}{Z(w)} \exp\left[f(X; w)\right], \tag{30}$$

where $Z(w)$ denotes the intractable partition function and $f(X; w)$ is the energy function. Let $\{X_i, i = 1, \ldots, n\}$ denote a sample. Then the empirical data distribution is

$$p_0(X) = \frac{1}{n} \sum_{i=1}^{n} \delta(X - X_i), \tag{31}$$

where $\delta()$ is the Dirac delta function. The log-likelihood is

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} \log p(X_i; w). \tag{32}$$

In maximum likelihood learning of parameters w we perform gradient ascent

$$w_{\tau+1} = w_\tau + \gamma \frac{\partial}{\partial w} L(w). \tag{33}$$

As shown in (16) the gradient of the log-likelihood is given by

$$\frac{\partial L(w)}{\partial w} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w} f(X_i; w) - E_{p_w}(\frac{\partial}{\partial w} f(X; w)) \tag{34}$$

$$= E_{p_0}(\frac{\partial}{\partial w} f(X; w)) - E_{p_w}(\frac{\partial}{\partial w} f(X; w)) \tag{35}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w} f(X_i; w) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \frac{\partial}{\partial w} f(\tilde{X}_i; w) \tag{36}$$

$$= E_{p_0}(\frac{\partial}{\partial w} f(X; w)) - E_{p_n}(\frac{\partial}{\partial w} f(X; w)). \tag{37}$$

We transform p_0 into p_n by starting a Markov chain from the data distribution p_0 and run the chain for n number of steps. That is, $p_n = T^n p_0$ where T is the Markov operator.

The log-likelihood gradient (35) performs traditional maximum likelihood (ML) learning, but it is intractable. To avoid this computational difficulty, Hinton proposed contrastive divergence (CD) [Hinton, 2002] learning where the gradient (37) approximatively follows the gradient of a different function. CD does noisy gradient ascent (noise due to averaging over a finite number of samples) on an objective function which approximates the log-likelihood.

Maximum likelihood learning minimizes the Kullback-Leibler divergence

$$KL(p_0||p_w) = \sum_X p_0(X) \log \frac{p_0(X)}{p_w(X)} \tag{38}$$

$$= \sum_X p_0(X) \log p_0(X) - \sum_X p_0(X) \log p_w(X) \tag{39}$$

$$= -H(p_0(X)) - \sum_X p_0(X) \log p_w(X) \tag{40}$$

$$\propto -\sum_X p_0(X) \log p_w(X) \tag{41}$$

$$= -\sum_X \left[\frac{1}{n} \sum_{i=1}^{n} \delta(X - X_i) \right] \log p_w(X) \tag{42}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \log p(X_i; w) = L(w). \tag{43}$$

Contrastive divergence learning approximatively follows the gradient of two divergences

$$CD_n = KL(p_0||p_w) - KL(p_n||p_w). \tag{44}$$

The first term minimizes divergence between p_w and p_0 which increases the likelihood of the sample $\{X_i, i = 1, \ldots, n\}$ (by reducing the corresponding energy), while the second term maximizes the divergence between p_n and p_w which decreases the likelihood of MCMC samples generated by p_w.

In practice, we can recover the gradient (37) by neglecting the

third term

$$\frac{\partial}{\partial w} CD_n = \frac{\partial}{\partial w} \left[KL(p_0||p_w) - KL(p_n||p_w) \right] \tag{45}$$

$$= E_{p_0}(\frac{\partial}{\partial w} f(X; w)) - E_{p_n}(\frac{\partial}{\partial w} f(X; w)) \tag{46}$$

$$- \frac{\partial}{\partial w} p_n(X) \frac{\partial}{\partial p_n(X)} KL(p_n||p_w) \tag{47}$$

$$\approx E_{p_0}(\frac{\partial}{\partial w} f(X; w)) - E_{p_n}(\frac{\partial}{\partial w} f(X; w)). \tag{48}$$

References

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL `http://dx.doi.org/10.1162/089976602760128018`.

Jianwen Xie, Wenze Hu, Song Chun Zhu, and Ying Nian Wu. A theory of generative convnet. *International Conference on Machine Learning*, 2016.

Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997. doi: 10.1162/neco.1997.9.8.1627. URL `https://doi.org/10.1162/neco.1997.9.8.1627`.