Machine learning for vision

Winter 2015

Roland Memisevic

Lecture 6, February 17, 2015

Overcomplete codes

- In area V1, the dimensionality of the retinal representation gets increased (by a factor of 25 or so).
- (If sparseness is a good thing, then this is probably not surprising.)
- If we want to turn the ICA model into such an overcomplete model we will need to make $\boldsymbol{W}$ rectangular.
- Recall the definition of the ICA log likelihood:

$$\log L(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n) = T \log \left| \det \boldsymbol{W} \right| + \sum_{i=1}^{n} \sum_{t=1}^{T} \log p_i(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x}_t)$$

Overcomplete codes

- The determinant is not defined for rectangular $\boldsymbol{W}$.
- But why is $|\det \boldsymbol{W}|$ there anyway?
- Because it is the **normalizing constant** that allows us to express densities over $\boldsymbol{x}$ using the linearly transformed $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}$.

$$p_x(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{W})} p_s(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x})$$

$$Z(\boldsymbol{W}) = \int_{\boldsymbol{x}} p_s(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

- This suggests dealing with overcomplete $\boldsymbol{W}$ by trying to optimize $Z(\boldsymbol{W})$ directly.
- Unfortunately, it will be hard to compute in general.
- $Z(\boldsymbol{W})$ is also called *partition function*.

Overcomplete codes

- This also allows us to define the density of images in a more general way.
- A general formulation, retaining independence of the hiddens, is

$$p(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{W})} \exp\left(\sum_{i=1}^{n} G(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x})\right)$$

$$Z(\boldsymbol{W}) = \int_{\boldsymbol{x}} \prod_{i=1}^{n} \exp\left(G(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x})\right) \, \mathrm{d}\boldsymbol{x}$$

where G is a non-linear function.

Markov Random Fields

- A separate advantage is that we may now define the model convolutionally, by scanning small filters across a larger image:

$$\begin{aligned}&\log p(\boldsymbol{x})\\ =\ &\sum_{x,y}\sum_{i} G\Big(\sum_{\xi,\eta} w^i_{\xi,\eta} I(x+\xi, y+\eta)\Big) - \log Z\end{aligned}$$

- This is known as field-of-experts (Roth & Black, 2005).

Overcomplete codes

- Bad news: For most models, we cannot compute the (log-)likelihood, nor its derivative.
- Good news (I): We can still *compare* the probabilities between observations $\boldsymbol{x}_i, \boldsymbol{x}_j$, because $Z(\boldsymbol{W})$ does not depend on $\boldsymbol{x}$.
- Good news (II): We can do *approximate* maximum likelihood training, which often seems to work just as well in practice.
- Good news (III): In practice, we usually need the *features* $\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}$ for test-data not probabilities.

Energy based models

- To gain intuitive insights into the learning problem, it can be useful to rewrite the probability of data in an even more general form:

$$p(\boldsymbol{x}; \boldsymbol{W}) = \frac{1}{Z(\boldsymbol{W})} q(\boldsymbol{x}; \boldsymbol{W}) \ , \quad Z(\boldsymbol{W}) = \int_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{W}) \ \mathrm{d}\boldsymbol{x}$$

with the log-likelihood

$$\log p(\boldsymbol{x}; \boldsymbol{W}) = \log q(\boldsymbol{x}; \boldsymbol{W}) - \log \int_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{W}) \ \mathrm{d}\boldsymbol{x}$$

- We can think of $q(\boldsymbol{x})$ as an unnormalized ("pre-") probability.
- Usually, $q(\boldsymbol{x}) = \prod_{i=1}^{n} \exp\left(G(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x})\right)$

Energy based models

- For a set of IID points, $\boldsymbol{x}_i$, the log-likelihood and its derivative can be written

$$\begin{aligned}L(\boldsymbol{W}) &= \sum_i \log p(\boldsymbol{x}_i; \boldsymbol{W})\\ &= \sum_i \log q(\boldsymbol{x}_i; \boldsymbol{W}) - N \log \int_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{W}) \ \mathrm{d}\boldsymbol{x}\end{aligned}$$

$$\frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}} = \sum_i \frac{\partial \log q(\boldsymbol{x}_i; \boldsymbol{W})}{\partial \boldsymbol{W}} - \frac{N}{Z(\boldsymbol{W})} \int_{\boldsymbol{x}} \frac{\partial q(\boldsymbol{x}; \boldsymbol{W})}{\partial \boldsymbol{W}} \ \mathrm{d}\boldsymbol{x}$$

- Intuitively, what does maximizing these two terms do with the model?

Energy based models

- For a set of IID points, $\boldsymbol{x}_i$, the log-likelihood and its derivative can be written

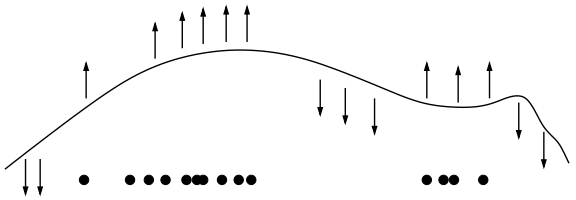
$$\begin{aligned} L(\boldsymbol{W}) &= \sum_i \log p(\boldsymbol{x}_i; \boldsymbol{W}) \\ &= \sum_i \log q(\boldsymbol{x}_i; \boldsymbol{W}) - N \log \int_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{W}) \, \mathrm{d}\boldsymbol{x} \end{aligned}$$

$$\frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}} = \sum_i \frac{\partial \log q(\boldsymbol{x}_i; \boldsymbol{W})}{\partial \boldsymbol{W}} - \frac{N}{Z(\boldsymbol{W})} \int_{\boldsymbol{x}} \frac{\partial q(\boldsymbol{x}; \boldsymbol{W})}{\partial \boldsymbol{W}} \, \mathrm{d}\boldsymbol{x}$$

- Intuitively, what does maximizing these two terms do with the model?

Maximum likelihood learning will *increase* $q(\boldsymbol{x}; \boldsymbol{W})$ at the data points, and *decrease* $q(\boldsymbol{x}; \boldsymbol{W})$ everywhere.

Energy based models



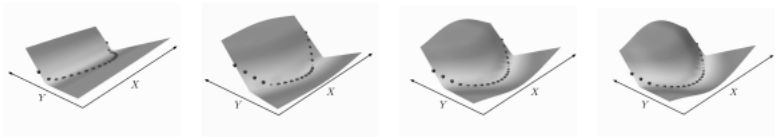
- Decreasing $q(\boldsymbol{x}; \boldsymbol{W})$, and thereby $p(\boldsymbol{x}; \boldsymbol{W})$, ensures that we get a normalized probability distribution.

Energy based models, contrastive divergence

- This view is even more liberating than replacing the determinant with $Z(\boldsymbol{W})$:
- In high-dimensional spaces, pushing down probabilities *near* the data may be good enough if pushing down everywhere is to expensive.
- This will be OK, if all we will ever see are test-cases from high-density regions.
- Technically, the pushing-down terms can also be derived by sampling from the model distribution, and restricting the sampling region to the vicinity of training examples is an approximation known as "contrastive divergence" (Hinton, 2002)

Energy based models, non-probabilistic

- We may eliminate the partition function altogether and define the model as an "energy landscape" that we form through learning.
- This gives us even more freedom in devising schemes that push or pull on the energy landscape.
- (LeCun, et al. 2006)

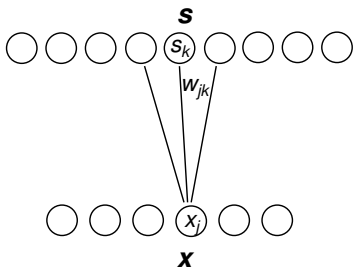


Energy based models, comments

- In practice, it is common to define energies as $-q(\boldsymbol{x}; W)$, in which case we want to *minimize* energy near the data.
- For feature learning energy-based models practically always contain hidden variables that are connected to pixels in a bi-partite graph.
- In other words, most feature learning models are based on a variation of the linear encoder/decoder equations.
- The effect of pushing down (or up) the energy away from the data usually corresponds to (and is the result of) a *capacity constraint* on the hiddens.

Feature learning and bi-partite networks

$$\boldsymbol{s} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}$$



- PCA is a special case with linear dependencies and low-dimensional $\boldsymbol{s}$
- ICA is a special case with linear dependencies and sparse $\boldsymbol{s}$
- Other special cases: restricted Boltzmann machines, autoencoder networks, sparse coding, k-means clustering