Machine learning for vision

Fall 2013

Roland Memisevic

Lecture 7, October 15, 2013

Overcomplete codes

- V1 increases the dimensionality of the representation as compared to the retinal representation (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 optimizing $Z(\boldsymbol{W})$ 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 common general formulation, retaining independence of the hiddens, is:

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

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

where G_i are any non-linear functions.

Markov Random Fields

- We may now define the model convolutionally, by scanning small filters across a larger image.
- Define fine filters to have the shape of image patches, and write:

$$\begin{aligned}&\log p(W_1, \ldots, W_n) \\ = &\sum_{x,y} \sum_{i=1}^{n} G\Big(\sum_{\xi,\eta} w_{\xi,\eta} I(x+\xi, y+\eta)\Big) - \log Z(W_1, \ldots, W_n)\end{aligned}$$

- (eg. Roth & Black, 2005)

Overcomplete codes

- Bad news I: In general, we cannot compute the log-likelihood, nor its derivative.
- Bad news II: In the absence of a computable partition function, we cannot evaluate probabilities for test-data, either.

Overcomplete codes

- Good news I: In practice we often care only about computing *features* $\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x}$ for test-data.
- Good news II: We can still *compare* the probabilities between points $\boldsymbol{x}_i, \boldsymbol{x}_j$, because $Z(\boldsymbol{W})$ does not depend on $\boldsymbol{x}$.
- Good news III: We can do *approximate* maximum likelihood training, which often works just as well in practice.

Energy based models

- To get insights into the learning problem, it is useful to rewrite the probability of data in an even more general form:

$$p(\boldsymbol{x}; \boldsymbol{W}) = \frac{1}{Z(\boldsymbol{W})} q(\boldsymbol{x}; \boldsymbol{W})$$

with log-probability

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

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

Energy based models

- For a set of IID points, $\boldsymbol{x}_i$, the log-likelihood and its derivative take the form:

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

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

- 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 take the form:

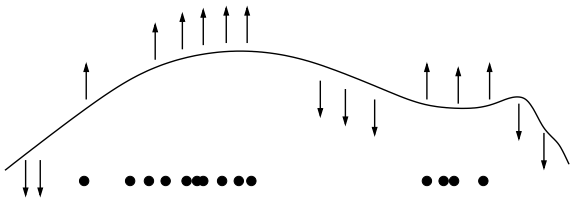
$$\begin{aligned} L(\boldsymbol{W}) &= \sum_i \log p(\boldsymbol{x}_i; \boldsymbol{W}) \\ &= \sum_i \log q(\boldsymbol{x}_i; \boldsymbol{W}) - N \log \int_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{W}) \, \mathrm{d}\boldsymbol{x} \end{aligned}$$

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

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

Maximum likelihood learning *increases* $q(\boldsymbol{x}; \boldsymbol{W})$ at the data points, and it *decreases* $q(\boldsymbol{x}; \boldsymbol{W})$ everywhere.

Energy based models



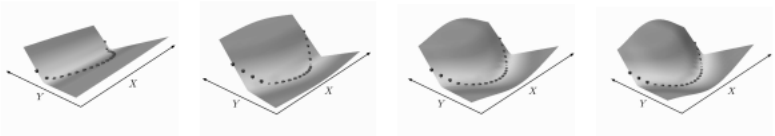
- Decreasing $q(\boldsymbol{x}; \boldsymbol{W})$, and thereby $p(\boldsymbol{x}; \boldsymbol{W})$, ensures that we get a normalized probability distribution.

Energy based models

- This view is even more liberating than replacing the determinant with $Z(\boldsymbol{W})$:
- In high-dimensional spaces, it can be easy to push down probabilities near the data and to ignore what's happening far away.
- This will be OK, if all we will ever see are points from high-density regions.
- Technically, the pushing-down can be done by sampling from the model distribution. This is inefficient in high-dimensional spaces. Solution: Sample only near the data.
- "Contrastive divergence" (Hinton, 2002)

Energy based models

- 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):

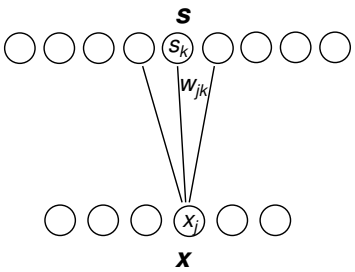


Energy based models

- It is common to define energies as $-q(\boldsymbol{x}; W)$, in which case we want to *minimize* energy near the data.
- Energy based models can be used in a variety of tasks, but for feature learning, they practically always involve hidden variables, which 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.
- Pushing up the energy away from the data typically translates into 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 $\boldsymbol{s}$