SUPPLEMENTARY MATERIAL FOR
“BAYESIAN DEEP DECONVOLUTIONAL LEARNING”

Yunchen Pu, Xin Yuan and Lawrence Carin
Department of Electrical and Computer Engineering, Duke University, Durham, NC, 27708, USA
{yunchen.pu,xin.yuan,lcarin}@duke.edu

1 CONDITIONAL POSTERIOR DISTRIBUTIONS FOR GIBBS SAMPLING

In \( l \)-th layer, the model is:

\[
X^{(n,k_{l-1},l)} = \sum_{k_l=1}^{K_l} D^{(k_l,l)} \ast \left( Z^{(n,k_l,l)} \odot W^{(n,k_l,l)} \right) + E^{(n,k_{l-1},l)}.
\] (1)

For brevity, we define the following symbols (operations):

\[
X^{-(n,k_{l-1},l)} = X^{(n,k_{l-1},l)} - \sum_{k_l} D^{(k_l-1,k_l,l)} \ast \left( Z^{(n,k_l,l)} \odot W^{(n,k_l,l)} \right).
\] (2)

\[
X^{-k_l^{(n,k_{l-1},l)}} = X^{(n,k_{l-1},l)} - \sum_{t \neq k_l} D^{(k_l-1,t,l)} \ast \left( Z^{(n,t,l)} \odot W^{(n,t,l)} \right).
\] (3)

\[
(\text{\textit{D}}^{(k_l-1,k_l,l)})^2 = \text{\textit{D}}^{(k_l-1,k_l,l)} \odot \text{\textit{D}}^{(k_l-1,k_l,l)}.
\] (4)

\[
(\text{\textit{W}}^{(n,k_l,l)})^2 = \text{\textit{W}}^{(n,k_l,l)} \odot \text{\textit{W}}^{(n,k_l,l)}.
\] (5)

\[
(X^{-(n,k_{l-1},l)})^2 = X^{-(n,k_{l-1},l)} \odot X^{-(n,k_{l-1},l)}.
\] (6)

where \( \odot \) is the element-wise product operator and \( \odot \) is the element-wise division operator.

\( A = B \oplus C \) denotes that if \( B \in \mathbb{R}^{n_B \times n_B} \) and \( C \in \mathbb{R}^{n_C \times n_C} \), \( A \in \mathbb{R}^{(n_B-n_C+1) \times (n_B-n_C+1)} \) with element \((i,j)\)

\[
A_{i,j} = \sum_{p=1}^{n_C} \sum_{q=1}^{n_C} B_{p+i-1,q+j-1} C_{p,q}.
\] (7)

We use ‘\( \cdot \)’ in the following equations to denote the conditioning parameters of the ensured distributions. For each MCMC iteration, the samples are drawn from:

- Sampling \( \text{\textit{D}}^{(k_l-1,k_l,l)} \):

\[
\text{\textit{D}}^{(k_l-1,k_l,l)} \sim N(\mu^{(k_l-1,k_l,l)}, \Sigma^{(k_l-1,k_l,l)})
\] (8)

\[
\Sigma^{(k_l-1,k_l,l)} = 1 \odot \left( \sum_{n=1}^{N} \gamma^{(n,k_l-1,l)} \| Z^{(n,k_l,l)} \odot W^{(n,k_l,l)} \|_2^2 + \gamma^{(k_l-1,k_l,l)} \right)
\] (9)

\[
\mu^{(k_l-1,k_l,l)} = \Sigma^{(k_l-1,k_l,l)} \odot \left( \sum_{n=1}^{N} \gamma^{(n,k_l-1,l)} X^{-k_l^{(n,k_{l-1},l)}} \odot (Z^{(n,k_l,l)} \odot W^{(n,k_l,l)}) \right)
\] (10)

- Sampling \( \gamma^{(k_l-1,k_l,l)} \):

\[
\gamma^{(k_l-1,k_l,l)} \sim \text{Gamma}\left( a_d + \frac{1}{2}, b_d + \frac{1}{2} \left( D^{(k_l-1,k_l,l)} \right)^2 \right)
\] (11)
Our model can take account of the missing data problem by adding one term to the model at the data level:

\[ Y^{(n)} = M^{(n)} \odot X^{(n)} = M^{(n)} \odot \left[ \sum_{k=1}^{K} (Z^{(n,k)} \odot W^{(n,k)}) + D^{(k)} \right] + E^{(n)}, \]

where \( M^{(n)} \) is the binary mask indicating which data point is observed, and \( Y^{(n)} \) is now the observed data. Recently, patch-based dictionary learning methods (Zhou et al. [2012]) have demonstrated superior performance on missing data interpolation. However, it may fail in the case of large-area missing since the model is built on small patches. The deep convolutional model developed in this paper, because of its hierarchical nature, can cope with this large-area missing problem in a natural way (Lee et al. [2009]). The essence is that the high-layer filters (dictionaries) will cover information of entire images and therefore large missing areas can be recovered using these dictionaries. Decent results are shown in the experiments of the main paper. Inference of this model is similar to the generative model, thus omitted here.
3 More Results

Figure 1: Trained dictionaries per class with each class (from Caltech 101) trained independently. Row 1-4: nautilus, pyramid, revolver, umbrella. Column 1-4: training images after local contrast normalization, layer 1 dictionary, layer 2 dictionary, layer 3 dictionary.

Figure 2: Dictionaries (layer 1 to layer 3) trained by 6 classes (ketch, nautilus, revolver, stop sign, umbrella, windsor chair) images from Caltech 101.

Fig. 1 plots dictionaries trained per class and four classes are shown. It can be seen the layer one dictionary elements are pretty similar while the upper layer dictionaries vary from each other. We also performed the model on a joint dataset with multi-class images, with trained dictionary elements shown in Fig. 2. It can be seen that the first layer and second layer dictionaries are more general in this joint training process. For instance, the first layer filters in Fig. 2 is an integration of filters of each class in Fig. 1.

References