Image Understanding with Deep Architectures

Mid-Term Progress Report

Hanlin GOH

Laboratoire d'Informatique de Paris 6
UPMC – Sorbonne Universités, Paris, France

Supervisors: Matthieu CORD, Joo-Hwee LIM, Nicolas THOME

Reviewers: Patrick GALLINARI, Thierry ARTIÈRES

Contents

1 Introduction

Achieving artificial intelligence (AI) in computers has been a subject of countless theses, spanning years of research. We have made significant progress in computational "intelligence" via information search techniques, that enable machines to beat chess grandmasters at their game or more recently defeat jeopardy legends. However, computational power is hardly the solution for many tasks that are seemingly trivial to humans. This is especially so problems such as visual perception.

The PhD thesis will be the marriage of two fields in AI – machine learning and computer vision. While machine learning focuses on making sense of the deluge of available data, computer vision aims to specifically tackle the visual aspects of image understanding. The aspiration is that we can eventually discover abstractions from low-level image content to high-level semantic concepts, either automatically or with as little human intervention as possible. In the recent years, highly hierarchical deep architectures have been proposed to tackle exactly this problem of bridging this semantic gap from image to semantics.

The main objective for this PhD project is to investigate how deep architectures, can help improve the performance of computational visual tasks, such as image-based object recognition and localization, scene understanding and visual search. We will be focusing our efforts on two aspects, namely:

1. Automatic discovery of visual representations, and
2. Construction of deep architecture to facilitate visual processing.

2 Foundations of deep architectures

A typical deep architecture consists of many layers of basic operations combined into a hierarchical network. The architecture takes raw input data at the lowest level and processes them via a sequence of basic computational units until the data is transformed to a suitable representation in the higher layers. The current approach of learning a deep architectures is to consider each layer as an unsupervised module and stacking them in a greedy-layer wise manner. Subsequently, an error-correcting supervised fine-tuning step can be performed to optimize the architecture for the required task. The rest of this section introduces two examples of unsupervised energy-based building blocks, namely the restricted Boltzmann machine and the encoder-decoder network.

2.1 Encoder-decoder network

The encoder-decoder network forms a fully-connected bipartite graph between a layer of I input (visible) units $\mathbf{x}$ and a layer of J latent (hidden) units $\mathbf{z}$. The layers are linked via symmetric weighted connections $\mathbf{W} \in \mathbb{R}^{I \times J}$. Additionally, each input x_i and latent z_j unit receives input from a bias – c_i or b_j respectively. The latent layer is activated from the input layer using the encoder, while the input layer is activated from the latent layer with the decoder, via encoding $f_{enc}(\cdot)$ and decoding $f_{dec}(\cdot)$ functions respectively:

$$\mathbf{z} = f_{enc}(\mathbf{W}^\top \mathbf{x} + \mathbf{b}) \tag{1}$$

$$\mathbf{x} = f_{dec}(\mathbf{W}\mathbf{z} + \mathbf{c}) \tag{2}$$

The system is governed by an energy function $E(\mathbf{x}, \mathbf{z})$ that is low when the pair of input and output vectors $(\mathbf{x}, \mathbf{z})$ exhibit compatibility or likelihood. The energy can be defined as a combination of energies from the encoder and decoder:

$$E(\mathbf{x}, \mathbf{z}) = \alpha_e \|\mathbf{z} - f_{enc}(\mathbf{x})\|_2^2 + \alpha_d \|\mathbf{x} - f_{dec}(\mathbf{z})\|_2^2, \tag{3}$$

where α_e and α_d are parameters proportional to the respective learning rates. The first term attempts to make the code $\mathbf{z}$ similar to the output of the encoder, while the latter term tries to minimize the reconstruction error of $\mathbf{z}$. Thus, the optimization concurrently learns both the encoder and the decoder.

2.2 Restricted Boltzmann machine

The restricted Boltzmann machine (RBM) is considered to be a special case of an encoder-decoder network (Ranzato et al., 2007). For an RBM with binary units, the activation probabilities of units in one layer are computed based on the states of the opposite layer, fed through a sigmoid activation function $sigm(\cdot)$:

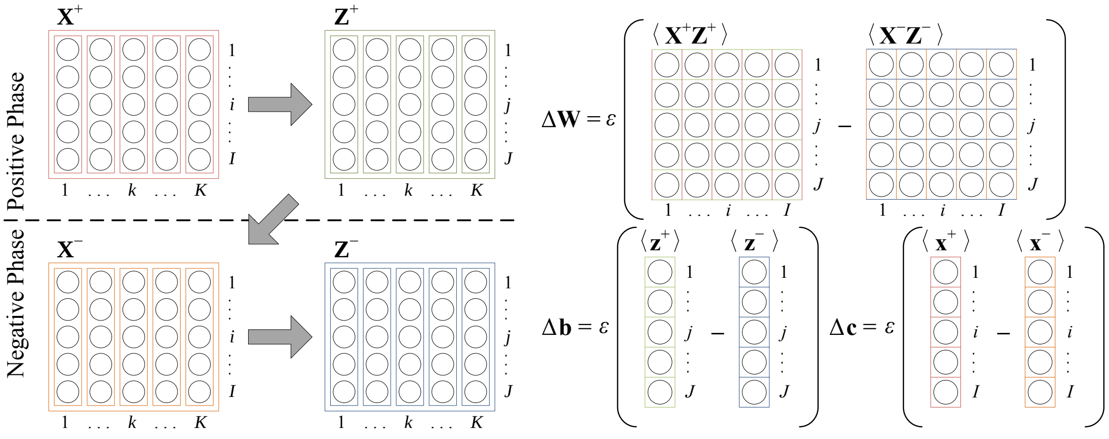
$$P\left(z_j \mid \mathbf{x}\right) = sigm\left(b_j + \sum_{i=1}^{I} w_{ij} x_i\right), \tag{4}$$

$$P\left(x_i \mid \mathbf{z}\right) = sigm\left(c_i + \sum_{j=1}^{J} w_{ij} z_j\right). \tag{5}$$

The negative log probability of a configuration of states $\{\mathbf{x}, \mathbf{z}\}$ can be defined by an energy function:

$$E\left(\mathbf{x}, \mathbf{z}\right) = -\log P\left(\mathbf{x}, \mathbf{z}\right) = -\sum_{i=1}^{I} \sum_{j=1}^{J} x_i w_{ij} z_j - \sum_{i=1}^{I} c_i x_i - \sum_{j=1}^{J} b_j z_j. \tag{6}$$

By modifying the parameters **W**, **b** and **c**, the energy of samples from the data distribution can be decreased, while raising the energy of reconstructions that the network prefers to real data.



Positive Phase
Negative Phase
$\mathbf{X}^+$
$\mathbf{Z}^+$
$\mathbf{X}^-$
$\mathbf{Z}^-$
1 ... k ... K
1 ⋮ i ⋮ I
1 ⋮ j ⋮ J
$\Delta\mathbf{W} = \varepsilon$
$\langle \mathbf{X}^+\mathbf{Z}^+ \rangle$
$\langle \mathbf{X}^-\mathbf{Z}^- \rangle$
1 ... i ... I
$\Delta\mathbf{b} = \varepsilon$
$\langle \mathbf{z}^+ \rangle$
$\langle \mathbf{z}^- \rangle$
$\Delta\mathbf{c} = \varepsilon$
$\langle \mathbf{x}^+ \rangle$
$\langle \mathbf{x}^- \rangle$

(a) Input and latent unit activations.

(b) Parameter updates.

Figure 1: The RBM learning algorithm (CD-1). (a) In the positive phase, latent units $\mathbf{Z}^+$ are activated based solely on inputs $\mathbf{X}^+$. The negative phase involves the reconstruction of $\mathbf{X}^-$ from $\mathbf{Z}^+$ and subsequently $\mathbf{Z}^-$ from $\mathbf{X}^-$. (b) The parameters **W**, **b** and **c** are updated based on the gradients from the positive and negative phases.

To train an RBM, one employs the contrastive divergence (CD) learning algorithm (Hinton, 2002) to approximate the maximum likelihood of the data and update parameters **W**, **b** and **c**. The RBM learning algorithm with one iteration of stochastic sampling (CD-1) is described in Figure 1. Given a batch of K training examples, $\mathbf{X}^+ \in \mathbb{R}^{I\times K}$ and $\mathbf{Z}^+ \in \mathbb{R}^{J\times K}$ are input and latent states resulting from sampling from the data distribution, while $\mathbf{X}^- \in \mathbb{R}^{I\times K}$ and $\mathbf{Z}^- \in \mathbb{R}^{J\times K}$ are reconstructed states. The parameters are updated after every iteration using the following update rules:

$$\begin{aligned}\Delta w_{ij} &= \varepsilon\left(\left\langle x_i^+ z_j^+\right\rangle - \left\langle x_i^- z_j^-\right\rangle\right), &(7)\\ \Delta b_j &= \varepsilon\left(\left\langle z_j^+\right\rangle - \left\langle z_j^-\right\rangle\right), &(8)\\ \Delta c_i &= \varepsilon\left(\langle x_i^+\rangle - \langle x_i^-\rangle\right), &(9)\end{aligned}$$

where ε is the learning rate and $\langle\cdot\rangle$ is defined as the average over the set of K examples.

Often, the activation probabilities of x_i and z_j are used in place of their binary states for parameter updates (see Hinton, 2010). This process is known as Rao-Blackwellization (Blackwell, 1947) and the results in an estimator with lower variance than when using binary states (Swersky et al., 2010). During parameter updates we will adopt this convention, where x_i and z_j are directly their conditional probabilities.

3 Selective and sparse image representations

There are several existing methods to induce selectivity in RBMs (Lee et al., 2008, Nair and Hinton, 2009). In addition to the likelihood term, these methods couple a penalty term $h(\mathbf{z})$ with the original energy term in the optimization:

$$\operatorname*{arg\,min}_{\{\mathbf{W},\mathbf{c},\mathbf{b}\}} \; -\sum_{k=1}^{K} \log \sum_{\mathbf{z}} \Pr\left(\mathbf{x}^{(k)}, \mathbf{z}^{(k)}\right) + \lambda h(\mathbf{z}) \; , \tag{10}$$

where λ is a regularization constant and $h(\mathbf{z})$ is a penalty term. To achieve selectivity, Lee et al. (2008) proposed to penalize the loss function based on the mean conditional expectation of each latent unit over the set of K training examples as follows:

$$h(\mathbf{z}) = \sum_{j=1}^{J} \left| p - \frac{1}{K} \sum_{k=1}^{K} z_j^{(k)+} \right|^2 . \tag{11}$$

The parameter p is used to control the intended selectivity of each unit. Meanwhile, Nair and Hinton (2009) proposed to use the cross-entropy measure between the observed and desired distributions to regularize the learning:[1]

$$h(\mathbf{z}) = \sum_{j=1}^{J} -p \log \left(\frac{1}{K} \sum_{k=1}^{K} z_j^{(k)+} \right) - (1-p) \log \left(1 - \left(\frac{1}{K} \sum_{k=1}^{K} z_j^{(k)+} \right) \right) . \tag{12}$$

There are theoretical drawbacks with these two approaches. For a latent unit to be selective, it should respond strongly to only a few examples and have low activation probabilities for the other examples. However, these methods merely regularize the learning such that the latent activation probabilities are low on the average. Even when the selectivity objective p is satisfied, the unit's activation may not be selective (for example, $z_j^{(k)+} = p, \forall k \in K$). As such, the latent units are necessarily stochastic and binary. Furthermore, since the regularizer considers only selectivity and not sparsity, we may get units that lack differentiation between each other. A population of latent units that respond selectively to the same few examples will still individually satisfy the regularization objective.

3.1 Precise regularization of Restricted Boltzmann machines

We aim to have a more precise control of the regularization process (Goh et al., 2010). The objective p can be realised as a spatiotemporal matrix $\mathbf{P} \in \mathbb{R}^{J \times K}$, where each element $p_j^{(k)} \in [0,1]$ is a latent activation bias encoding the desired z_j in response to input example k. Each row

[1] In their work, Nair and Hinton (2009) preferred a more complex but conceptually similar exponentially decaying conditional expectation of $z_j^{(k)}$ over the standard conditional expectation as described in Equation 12.

$\mathbf{p}_j$ represents the desired temporal activation sequence of z_j, while a column $\mathbf{p}^{(k)}$ is defined across the population of latent units given example k. More generally, $\mathbf{P}$ can be designed based on any inductive principle, not just sparsity.

The optimization problem follows the framework of Equation (10), with $h(\mathbf{z})$ now defined as the cross-entropy loss, summmed over the new penalty matrix:

$$h(\mathbf{z}) = \sum_{j=1}^{J} \sum_{k=1}^{K} -p_j^{(k)} \log z_j^{(k)+} - \left(1 - p_j^{(k)}\right) \log \left(1 - z_j^{(k)+}\right). \tag{13}$$

Together with the maximum likelihood approximation provided by contrastive divergence, the average updates for w_{ij} and b_j for a set of K examples can be simplified to be:

$$\Delta w_{ij} = \varepsilon \left(\left\langle x_i^{+} s_j \right\rangle - \left\langle x_i^{-} z_j^{-} \right\rangle \right), \tag{14}$$

$$\Delta b_j = \varepsilon \left(\left\langle s_j \right\rangle - \left\langle z_j^{-} \right\rangle \right). \tag{15}$$

Here,

$$s_j^{(k)} = \phi p_j^{(k)} + (1 - \phi)\, z_j^{(k)+}, \tag{16}$$

can be seen as the revised code of $z_j^{(k)+}$, where ϕ is a hyperparameter denoting the relative learning rate of the regularizer with respect to maximum likelihood estimation. The modified algorithm is illustrated in Figure 2.

3.2 Inducing selectivity and sparsity

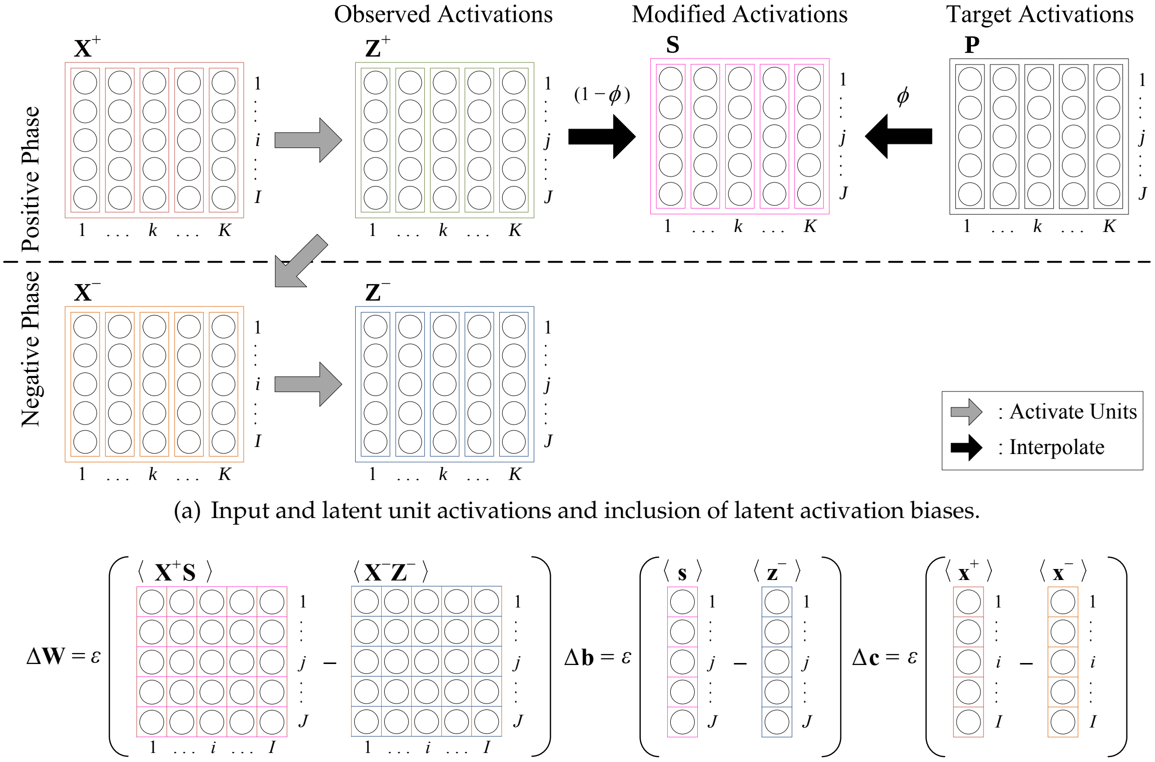
Now, unlike other methods, both selectivity and sparsity can be induced by designing an appropriate $\mathbf{P}$ matrix. By adapting the activation probabilities of latent units to fit heavy tails distributions, such as power-law, exponential and gamma distributions, in the lifetime (rows) or population (columns) domains, we can model their latent activity biases $\mathbf{P}$. For $\mathbf{P}$ to be both sparse and selective, the latent activations are fitted to desired heavy-tailed distributions. Let $\mathbf{z} \in \mathbb{R}^N$ be either row $\mathbf{z}_j^{+}$ for selectivity or column $\mathbf{z}^{(k)+}$ for sparsity. The latent activation bias p_n for z_n is computed as

$$p_n = \left(rank\left(z_n, \mathbf{z}\right)\right)^{(1/\mu)-1}. \tag{17}$$

where $rank\left(z_n, \mathbf{z}\right)$ assigns a value from 0 to 1 based on the rank of z_n in $\mathbf{z}$, with smallest given a value of 0 and the largest with 1. The target mean $0 < \mu < 1$ creates the power-law expression such that when $\mu < 0.5$, the distribution will be positively skewed.

Instead of merely getting the RBM to have low average activations (Lee et al., 2008, Nair and Hinton, 2009), the individual activations are biased such that collectively they form positively skewed distributions that have only a few highly activated units while most remain silent.

This is more precise. By inducing both selectivity and sparsity, the networks attempts to more explicitly relate specific examples to specific units.



Observed Activations
Modified Activations
Target Activations
$\mathbf{X}^+$
$\mathbf{Z}^+$
$\mathbf{S}$
$\mathbf{P}$
$(1-\phi)$
ϕ
1 : i : I
1 : j : J
1 . . . k . . . K
Positive Phase
Negative Phase
$\mathbf{X}^-$
$\mathbf{Z}^-$
: Activate Units
: Interpolate
$\Delta\mathbf{W} = \varepsilon$
$\langle \mathbf{X}^+\mathbf{S} \rangle$
$\langle \mathbf{X}^-\mathbf{Z}^- \rangle$
1 . . . i . . . I
$\Delta\mathbf{b} = \varepsilon$
$\langle \mathbf{s} \rangle$
$\langle \mathbf{z}^- \rangle$
$\Delta\mathbf{c} = \varepsilon$
$\langle \mathbf{x}^+ \rangle$
$\langle \mathbf{x}^- \rangle$

(a) Input and latent unit activations and inclusion of latent activation biases.

(b) Modified parameter updates.

Figure 2: The modified RBM learning algorithm facilitated by latent activation biases. (a) In the positive phase, latent units $\mathbf{Z}^+$ are re-activated as $\mathbf{S}$ with additional influences from latent activation biases $\mathbf{P}$ interpoliated by ϕ. (b) When updating parameters $\mathbf{W}$ and $\mathbf{b}$, the modified activation $\mathbf{S}$ replaces $\mathbf{Z}^+$, only the positive phase. $\Delta\mathbf{c}$ is unmodified from the original algorithm.

3.3 Evaluating sparse features

The RBMs were biased with selectivity and sparsity to efficiently represent natural images (Doi et al., 2003) and handwritten digits (LeCun et al., 1998). Using natural images, the result a set of Gabor-like edge detectors (Figure 3), consistent with other related methods (Doi and Lewicki, 2005, Lee et al., 2008, Olshausen and Field, 1996, Ranzato et al., 2007, Teh et al., 2004). With handwritten digits, the learned filters appear to encode handwritten strokes (Figure 4).

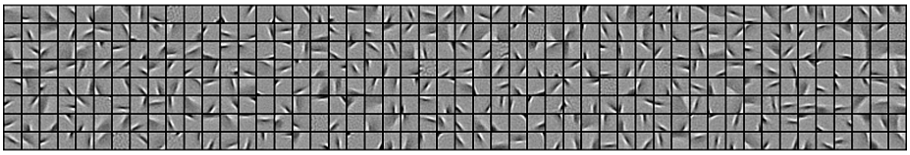


Figure 3: An example of a filter bank learned by an RBM biased with selectivity and sparsity. The filters are Garbor-like with varying orientation, spatial location and spatial frequency.

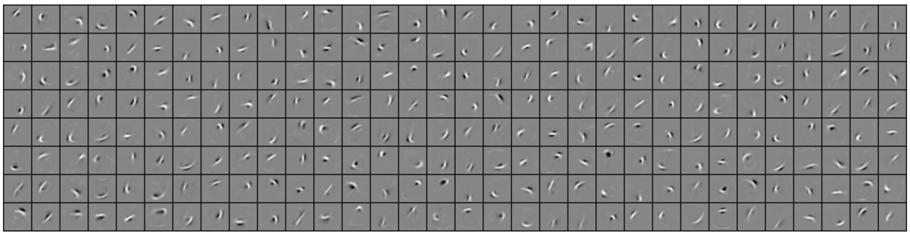
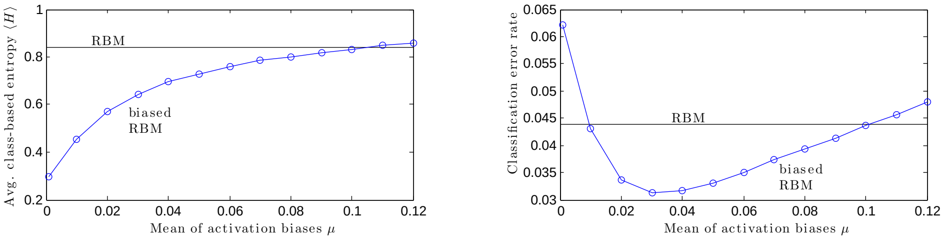


Figure 4: Filters learned on handwritten digits resemble local handwritten strokes.

Biasing RBMs improves discriminative performance. For each latent unit, the activation with respect to each class is totaled and normalized across classes. We then computed the Shannon entropy of each latent unit and finally averaged it across the population. This metric $\langle H \rangle$ gives us an indication of the level of class-based discrimination of the latent units, where lower $\langle H \rangle$ values signify fewer the number of classes each unit encodes. We also trained a simple multinomial logistic regression classifier from the activations of the latent layer (without backpropogating the features) and computed the classification error rate. Since there are 10 classes, one for each digit and roughly uniformly distributed, we consider that for a unit to be selective, it should respond to less than 10% of the samples. Hence, we conducted our study in the range of $0.001 \leq \mu \leq 0.12$.

From Figure 5(a), we observe a monotonic relationship of $\langle H \rangle$ with respect to μ. When μ is lowered, a unit responds to fewer examples. If examples from the same class have similar appearances, then it is more likely that these examples belong to the same class, thus lowering $\langle H \rangle$. Figure 5(b) shows that the relation between the classification error and μ is no longer monotonic. The model has poor generalization when μ nears 0 as units encode individual examples too specifically (Gross, 2002). For this data set, the biased RBM achieves better result than the standard RBM in the approximate range of $0.01 \leq \mu \leq 0.1$.



(a) Class-based entropy

(b) Classification error

Figure 5: Discriminative performance of RBMs biased with selectivity and sparsity. (a) $\langle H \rangle$ varies monotonically with μ. (b) Classification error is minimum when μ is low, but not at the lowest. There is a range of μ whereby biasing the RBM improves generalization performance.

4 Transformation invariant feature maps

Previously, RBMs are regularized by sparse priors to increase feature differentiation and discriminative power (Goh et al., 2010, Lee et al., 2008, Nair and Hinton, 2009). However, the representations might not be invariant to input transformations. We propose that if there is structured similarity between the features, then representations will smoothly vary with respect to the transformations and invariance can be achieved (Goh et al., 2011).

4.1 Inducing topographical organization

A two-layered scheme (Hyvärinen and Hoyer, 2001, Kavukcuoglu et al., 2009) is adapted to regularize the RBM. From latent representation $\mathbf{Z}^+$, we first compute a new set of activations $\widehat{\mathbf{Z}}$ based on fixed pooling weights. Subsequently, sparsity is induced to obtain target activations $\mathbf{P}$. The output of the second step is used to regularize the learning of the RBM.

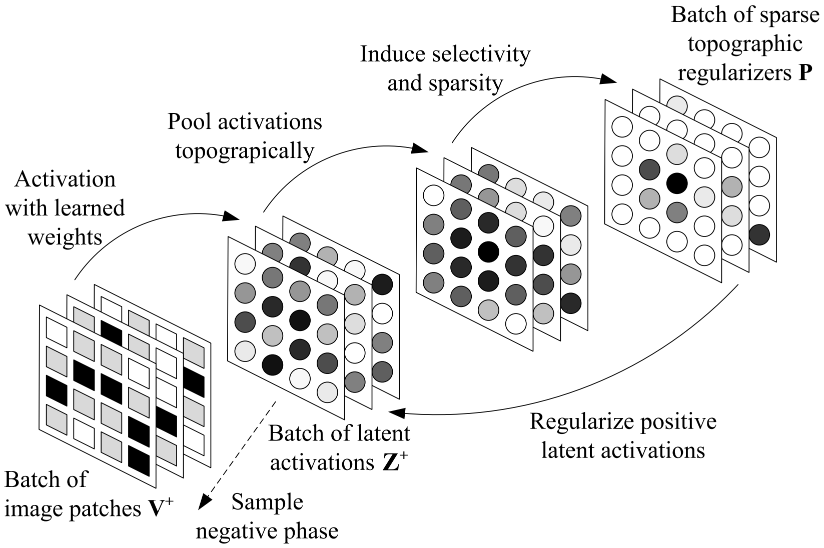

Batch of sparse topographic regularizers **P**
Induce selectivity and sparsity
Pool activations topograpically
Activation with learned weights
Batch of latent activations **Z**⁺
Regularize positive latent activations
Batch of image patches **V**⁺
Sample negative phase

Figure 6: The framework for inducing both sparseness and topographical organization. From a batch of pixel inputs $\mathbf{X}^+$, the latent units are activated $\mathbf{Z}^+$ via learned weights. The activations are then topographically pooled based on the locality of z_j in the feature map via fixed weights. Subsequently, population and lifetime sparseness are induced to obtain $\mathbf{P}$. Finally, $\mathbf{P}$ is used to regularize the learning of the parameters.

The layer of latent units is structured into a 2-dimensional feature map. In the 2D feature map, a topographical structural dependence is induced by introducing between the latent units via a new layer $\widehat{\mathbf{Z}}$, where each $\widehat{z}_j^{(k)}$ pools activations from the neighborhood of $z_j^{(k)+}$. Each unit in $\mathbf{Z}^+$ activates units in $\widehat{\mathbf{Z}}$ depending on the relative locality of the units:

$$\widehat{h}_j^{(k)} = \sum_{m=1}^{M} h_m^{(k)+} \omega\left(j, m\right) \tag{18}$$

where the fixed topographic pooling weights $\omega(\cdot,\cdot)$ are functions of the topographic distance between two units. A Gaussian kernel with wrap around was used.

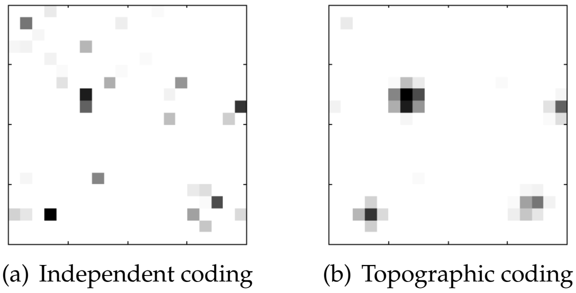
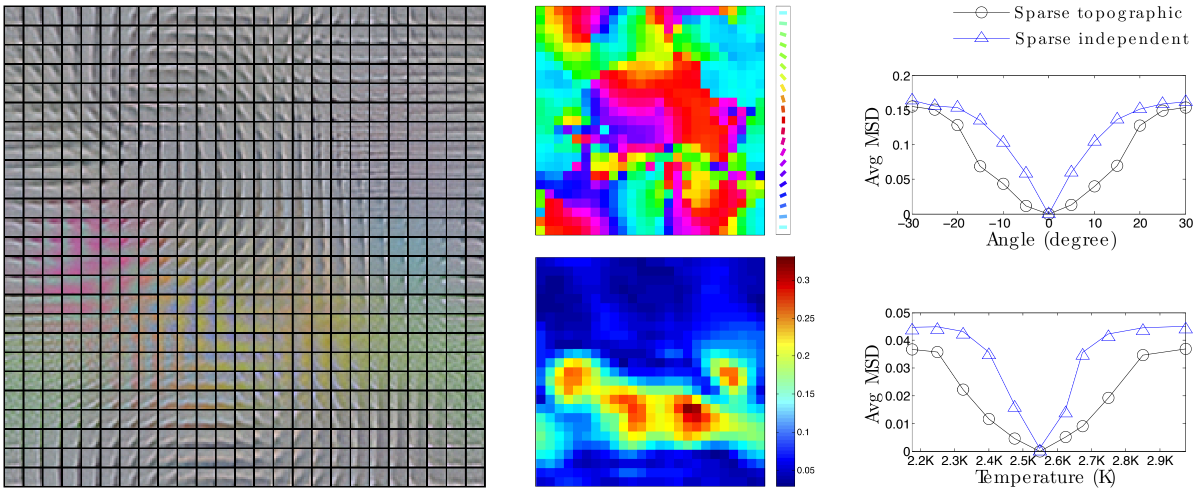

(a) Independent coding (b) Topographic coding

Figure 7: Comparing activations of independent coding (a) and topographic coding (b). Each pixel shows the activation of a unit in the feature map, where darker color denotes a higher activation. When topographic organization is induced, the activations are spatially grouped within the 2D feature map.

4.2 Evaluating feature invariance

We trained an RBM with sparse topographic regularization using colored natural image patches from the McGill Calibrated Colour Image Database (Olmos and Kingdom, 2004). The resulting 2D feature map (Fig. 8(a)) consists of Gabor-like filters with varying spatial frequency, position, orientation and color (Fig. 8(b)). The visual appearance of filters vary smoothly across the feature map. To our knowledge, this is the first 2D feature map that models color information.



(a) 2D topographical feature map (b) Analyses of feature map (c) Feature invariance

Figure 8: (a) (b) Appearance of filters vary smoothly across the feature map when broken down to their components such as orientation (above) and color saturation (below). (c) Comparing the invariance of between sparse topographic features and sparse independent features for rotation (above) and varying illumination color (below).

We evaluate the features learned based on their invariance to affine transformations (translation, rotation, scaling) and changes to illumination color. Patches of varying degrees of transformation, relative to a non transformed patch, were sampled from the data sets. For every input patch, the output signature of latent unit activations was recorded. To quantitively measure invariance, we took the mean squared difference (MSD) between the signatures of the transformed input and that of the untransformed input. The MSD was then averaged across the samples and plotted in Fig. 8(c). In every evaluation task, when the transformation is low, topographic features are more invariant than independent ones. There is little difference between the two feature types under large transformations. The signature of a slightly transformed input is highly similar to the original signature. As the amount of transformation increases, the signature gradually shifts and invariance reduces.

5 Current work

There are two aspects to learning deep architectures, which are learning between layers and system-level modeling. Having explored the manipulation of the feature coding process between two layers in a deep architecture, we will move on to constructing deep architectures at a systems level and the combination of information processing modules. We want to the architecture to specialize in image understanding tasks, particularly object recognition and localization. We currently have a few research directions under investigation.

5.1 Learning hierarchical visual dictionaries

A popular method for modeling images for object recognition is to build visual dictionaries from local image descriptors, like the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005), scale invariant feature transform (SIFT) (Lowe, 1999) and speeded-up robust features (SURF) (Bay et al., 2008). Although more abstract than image pixels, a semantic gap still exists between the low-level image descriptors and higher-level concepts. Our objective is to bridge this gap by further transforming bottom-up visual information by using modules such as the sparse RBMs that we have described. This helps alleviate the role of the classifier in the later stage in the architecture. In our experiments, we found that it is not a trivial problem to use the RBM to encode SIFT descriptors. We are exploring various data normalization techniques to seamlessly fit the modules together.

5.2 Combining bottom-up and top-down deep learning

So far, deep architectures are constructed by stacking unsupervised modules and adding a supervised module in the last step. However, the object recognition task has supervised elements, such as the labels of objects. Although the labels are often coarse, we think that these information should not be discarded, even when learning lower level features. Based on the current approaches, not knowing the nature of the supervised task, features that are automatically discovered via unsupervised learning may not be suitable for the task. We hope to learn the entire deep architecture as whole, instead of layer-by-layer. More importantly top-down information from the labels should be employed to enhance the representation. In this sense, as we do deeper into the architecture, it untangles the manifold of classes from image space to higher level abstract or semantic spaces. Based-on our initial studies and experimentation, the standard encoder-decoder network appears to be a good starting point to realize such deep networks.

5.3 Attention-based feature pooling in image space

Image information manifest in 2-dimensional image space. However, this space is rarely exploited in the image modeling process. Current methods revolve around the use of the convolution operation followed by a pooling step to gain invariance to the spatial position of features (Lee et al., 2009). Using cognitive-inspired concepts, such as visual saliency and top-down attention, we hope to be able to perform visual scanning of images in a manner that is more human-like. We believe that this will help greatly in object localization and visual search tasks. Human retina vision consists of a highly-sensitive central fovea area used mainly for appearance representation and the coarser peripheral vision used for attentional purposes. As we fixate around different parts of the scene, we pool groups of local representations together. This pooling operation models the intrinsic image-space structure of different objects in an image.

6 Conclusion

For this PhD, we focus on the fusion of two research topics – deep learning and computer vision. We approach the problem in two phases, attempting to tackle the problem at a micro level and a systems level. The micro level problem revolves around the automatic discovery of feature representations. In the last 18 months, we explored the discovery of representations beneficial to vision tasks. The work on sparse coding, being more theoretical, has been presented at the NIPS workshop on deep learning and unsupervised feature learning in 2010 (Goh et al., 2010). Meanwhile, the research on topographic coding is more image-oriented and was just presented at the International Conference on Image Processing (Goh et al., 2011).

We are currently exploring the system level problem, which involves the combination of basic processing modules in a manner that facilitates image understanding. Three complementary approaches are currently under investigation, namely 1) learning hierarchical visual dictionaries from image features, 2) combining bottom-up and top-down information during the deep learning process, and 3) using attention to catalyze feature pooling in image space. The current work is projected to be completed in about 15 months and cumulating in PhD defense in the first quarter of 2013.

References

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346–359.

Blackwell, D. (1947). Conditional expectation and unbiased sequential estimation. *Annals of Mathematical Statistics*, 18:105–110.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 –893.

Doi, E., Inui, T., Lee, T.-W., Wachtler, T., and Sejnowski, T. J. (2003). Spatiochromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. *Neural Computation*, 15:397–417.

Doi, E. and Lewicki, M. S. (2005). Sparse coding of natural images using an overcomplete set of limited capacity units. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 377–384. MIT Press, Cambridge, MA.

Goh, H., Kusmierz, L., Lim, J.-H., Thome, N., and Cord, M. (2011). Learning invariant color features with sparse topographic restriced Boltzmann machines. In *International Conference on Image Processing (to appear)*.

Goh, H., Thome, N., and Cord, M. (2010). Biasing restricted Boltzmann machines to manipulate latent selectivity and sparsity. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Gross, C. G. (2002). Genealogy of the “grandmother cell”. *Neuroscientist*, (8):512–518.

Hinton, G. (2010). A practical guide to training restricted boltzmann machines. Technical Report UTML TR 2010–003, Department of Computer Science, University of Toronto.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hyvärinen, A. and Hoyer, P. O. (2001). A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41(18):2413–2423.

Kavukcuoglu, K., Ranzato, M., Fergus, R., and LeCun, Y. (2009). Learning invariant features through topographic filter maps. In *CVPR*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, H., Ekanadham, C., and Ng, A. (2008). Sparse deep belief net model for visual area V2. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 873–880. MIT Press, Cambridge, MA.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616, New York, NY, USA. ACM.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150 –1157.

Nair, V. and Hinton, G. (2009). 3D object recognition with deep belief nets. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1339–1347.

Olmos, A. and Kingdom, F. A. A. (2004). McGill calibrated colour image database. http://tabby.vision.mcgill.ca.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607.

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2007). Efficient learning of sparse representations with an energy-based model. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 1137–1144. MIT Press, Cambridge, MA.

Swersky, K., Chen, B., Marlin, B., and de Freitas, N. (2010). A tutorial on stochastic approximation algorithms for training restricted boltzmann machines and deep belief nets. In *Information Theory and Applications (ITA) Workshop*.

Teh, Y. W., Welling, M., Osindero, S., and Hinton, G. E. (2004). Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(7-8):1235–1260.