Paper:

A Probabilistic WKL Rule for Incremental Feature Learning and Pattern Recognition

Jasmin Léveillé[*], **Isao Hayashi**[**], **and Kunihiko Fukushima**[**,***]

[*]Center of Excellence for Learning in Education, Science and Technology, Boston University
677 Beacon Street, Boston, Massachusetts 02215, USA
E-mail: jasminl@bu.edu
[**]Faculty of Informatics, Kansai University
2-1-1 Ryozenji-cho, Takatsuki, Osaka 569-1095, Japan
E-mail: ihaya@cbii.kutc.kansai-u.ac.jp
[***]Fuzzy Logic Systems Institute
680-41 Kawazu, Iizuka, Fukuoka 820-0067, Japan
E-mail: fukushima@m.ieice.org

[Received October 14, 2013; accepted March 28, 2014]

Recent advances in machine learning and computer vision have led to the development of several sophisticated learning schemes for object recognition by convolutional networks. One relatively simple learning rule, the Winner-Kill-Loser (WKL), was shown to be efficient at learning higher-order features in the neocognitron model when used in a written digit classification task. The WKL rule is one variant of incremental clustering procedures that adapt the number of cluster components to the input data. The WKL rule seeks to provide a complete, yet minimally redundant, covering of the input distribution. It is difficult to apply this approach directly to high-dimensional spaces since it leads to a dramatic explosion in the number of clustering components. In this work, a small generalization of the WKL rule is proposed to learn from high-dimensional data. We first show that the learning rule leads mostly to V1-like oriented cells when applied to natural images, suggesting that it captures second-order image statistics not unlike variants of Hebbian learning. We further embed the proposed learning rule into a convolutional network, specifically, the Neocognitron, and show its usefulness on a standard written digit recognition benchmark. Although the new learning rule leads to a small reduction in overall accuracy, this small reduction is accompanied by a major reduction in the number of coding nodes in the network. This in turn confirms that by learning statistical regularities rather than covering an entire input space, it may be possible to incrementally learn and retain most of the useful structure in the input distribution.

Keywords: winner-kill-loser, incremental learning, natural image statistics, neocognitron, convolutional network

1. Introduction

Unsupervised learning techniques have been extensively used to learn features for convolutional neural networks in order accomplish difficult visual recognition tasks [1, 2]. Various unsupervised learning schemes have been proposed to learn good recognition features, including collecting a dictionary of features from an image dataset [3, 4], Hebbian-inspired learning [5], contrastive divergence [6], Independent Component Analysis [7, 8], and reconstruction error minimization [9].

Dictionaries of features are built incrementally upon presentation of input patterns, by inserting new units that code for regions of the input space not covered by a current set of units. Upon insertion, a new unit's weights are set to encode the values of the input pattern currently applied and are fixed throughout the remainder of the learning phase. Although simple to implement, such incremental techniques can easily lead to prohibitively large dictionaries being learned on typical visual recognition tasks. Feature sets learned through dictionary building therefore run the risk of low coding efficiency since a large code is used to represent the data (cf. [10]).

Techniques based on Hebbian learning instead typically assume a fixed number of units whose weights are gradually learned upon repeated presentation of inputs with the product of pre- and post-synaptic activity. When used in conjunction with a suitable normalization operator and competitive interactions among feature units, the learned synaptic kernels correspond to either independent components [11] or principal components [12]. Hebbian learning can actually be related to reconstruction error minimization, in which synaptic weights are learned in order to minimize an objective function that measures the distortion between the input and the backprojected (i.e. through the synaptic matrix transpose) output [13]. One variation of Hebbian learning – the trace rule – uses low-pass filtered post-synaptic output with the hope of achieving spatial invariance through learning from continuously varying input [5, 14–16]. Trace rule learning is sometimes equivalent to Slow Feature Analysis [17], in which the goal is to promote learning of features that capture slow variations in the input for purpose of invariance [18, 19]. Despite its simplicity and the possibly limiting influence of the linearity assumption of Hebbian learning, state-of-the-art results on standard image datasets have been achieved with this framework. Nevertheless, model selection remains an issue, such that the only viable strategy to obtain good results may be to rely on high throughput parameter search [5].

Hebbian-based, competitive learning has also been used in conjunction with a dictionary-building strategy in the Neocognitron model [20]. According to this scheme, existing units compete upon presentation of inputs, and the weights of the winning unit are modified according to a variant of Hebbian learning. New units are added whenever none of the existing units is sufficiently activated by a given training input. This method has shown good results in particular on digit recognition datasets.

Contrastive divergence has also been used to learn features for convolutional networks [6]. Learning with contrastive divergence roughly follows the gradient of the log-likelihood [21]. The results obtained with that approach on standard image datasets have been comparable to that obtained with other state-of-the art approaches. Here again, the number of units is fixed in advance and model selection is an issue. Learning with Independent Component Analysis (ICA) is somewhat related to the maximum likelihood approach in that both can be traced back to optimizing a log-likelihood criterion [22], although ICA is intrinsically tied to the use of a sparsity criterion. Results obtained with ICA have been in line with that of other approaches on standard image datasets [7, 8].

Instead of performing gradient ascent on the log-likelihood function, another learning strategy is to perform gradient descent on the input reconstruction error in an autoencoder network [23]. Minimizing the reconstruction error can be shown to be equivalent to maximizing a lower bound on the mutual information between the input and output layers of the autoencoder [24]. Recent energy-based methods for learning convolutional autoencoders have yielded some of the best performing methods on standard object recognition benchmarks [1, 9].

Within the above approaches, the simplest learning rules are attractive for at least two reasons. First, simple learning mechanisms are more likely to be suitable for hardware acceleration, as shown by the fact that state-of-the-art hardware implementations of convolutional networks currently have to perform learning offline [25]. Second, learning mechanisms that rely purely on local computations and that do not require sophisticated numerical procedures are generally more adequate as explanations of learning in the biological brain [26].

The first issue addressed in the current paper is whether one such simple learning rule, the Winner-Kill-Loser (WKL), can be used to learn oriented edge detectors similar to the receptive fields of simple cells in cortical area V1. Learning oriented edge detectors is a valuable goal as these may be considered as the building blocks for more sophisticated visual processing. Based on our investigations into the WKL rule, we modify the original rule with the intent to capture local statistical regularities in images, and test whether the proposed modification yields any benefits in a digit recognition task. If the modified rule better captures spatial information in the input image distribution, this should be observable as either an improvement in overall classification accuracy and/or a reduction in the number of units needed to reach that accuracy. As shown in the simulations reported in section 5, although our revised WKL rule does not improve accuracy over the original learning rule – in fact, it reduces it slightly – the number of units needed to reach that accuracy is drastically reduced. Most of the results on the theoretical properties of WKL and the natural image simulations have already appeared in [27], results on the digit classification task are presented here for the first time.

2. The WKL Learning Rule

The WKL rule was proposed in [20] for purpose of learning higher-order combinations of features in the neocognitron model for written digit classification [28]. The WKL rule can be described as follows. Let x be an input pattern, the activity of unit j is represented by a similarity function $s_j = f(w_j, x)$, where w_j is the unit's synaptic kernel. The similarity function f is typically implemented as a normalized dot product:

$$s_j = \frac{w_j \cdot x}{||w_j||||x||}, \quad \text{. . . . .} \quad (1)$$

where $||\cdot||$ is the L_2 norm. Common alternatives to Eq. (1) include radial-basis functions of the form:

$$s_j = \gamma e^{-\beta||x-w_j||^2}, \quad \text{. . . . .} \quad (2)$$

which, given certain assumptions on the norm of input vector x and weight vector w_j, can be shown to be nearly equivalent to Eq. (1) [29], a relationship that holds empirically [30]. Let i be the index of the most active unit within a group of units tuned to different features. Given a certain activity threshold θ, and assuming $s_i > \theta$, the weight update for the winning unit is defined as:

$$\Delta w_i = \lambda x, \quad \text{. . . . .} \quad (3)$$

for a given learning rate λ. The threshold θ may be seen as defining a neighborhood in the input space, located around the center w_i, for which patterns will strongly activate unit i. Reference [20] considered only the case of $\lambda = 1$, although here we use $\lambda = 0.05$ for all simulations. The weights for all inactive units (i.e. units whose activity is less than the threshold) are left unchanged. If no unit has an activity level beyond θ, a new node is created whose center is set to the input pattern x. Up to this point, the WKL is identical to incremental Winner-Take-All learning [31, 32]. However, the WKL includes an additional step whereby units whose activity is higher than the threshold but is less than that of the winner are removed from the network. WKL learning is thus a variant of incremental, competitive clustering techniques that include the leader algorithm [33] and adaptive resonance theory [34], and differs from these by the use of a fast decremental step.

The purpose of the decremental step is to reduce redundancy in coding while allowing for a complete covering of the input space with fewer nodes than in traditional WTA learning, leading to a form of sparse coding of the input [35]. Simulations in the neocognitron have shown that the WKL compares favorably at least on a written digit recognition task [20].

Still, one issue with incremental learning rules such as WKL is the potentially large number of nodes needed to cover the input space. With WKL learning, this problem appears in two different ways. First, the removal of existing nodes can be seen to lead to gaps in the covering, at least when using a spherical cluster neighborhood as in Eq. (1). Second, the sheer size of the input space may be so high as to require a prohibitive number of nodes to cover its volume.

The first type of problem is not really of concern, as gaps between closely connected neighborhoods can be dealt with using simple strategies [36]. Furthermore, the general influence of these gaps can be expected to decrease as the dimensionality of the input space increases. That this is the case can be seen by considering the ratio of the volumes of the gaps to that of the nodes' neighborhood. For any dimension n, the spherical neighborhoods of three adjacent nodes form an equilateral triangle on a hyperplane. Let r be the radius of those spheres. Within that hyperplane, the three spheres define another one that covers approximately the projection, onto the hyperplane, of the gap at their intersections. The radius of that sphere is given by:

$$q = \frac{r}{\cos\frac{\pi}{6}} - r, \quad \dots\dots\dots\dots \quad (4)$$

and its volume is:

$$V_n q^n, \quad \dots\dots\dots\dots\dots\dots \quad (5)$$

where V_n is the volume of a hypersphere of radius 1 [37]. The ratio of the volumes of the gap to that of the nodes' neighborhoods is then:

$$\frac{V_n q^n}{V_n r^n} = \frac{\left(\frac{r}{\cos\frac{\pi}{6}} - r\right)^n}{r^n} = \left(\frac{1}{\cos\frac{\pi}{6}} - 1\right)^n . \quad (6)$$

It is easy to see that, as $n \to \infty$, this ratio converges to 0. Thus, as dimension increases, gaps between connected neighborhoods should have less impact on the covering. The above result bears similarity with the vanishing ratio of the volume of a hypersphere to that of an enclosing hypercube, which is typically invoked in describing the curse of dimensionality [38]. **Fig. 1(b)** shows the result of Monte-Carlo simulations that confirm this intuition. Unlike the gaps between adjacent neighborhoods, increasing the input space dimensionality actually worsens the second type of covering problem (**Fig. 1(a)**).

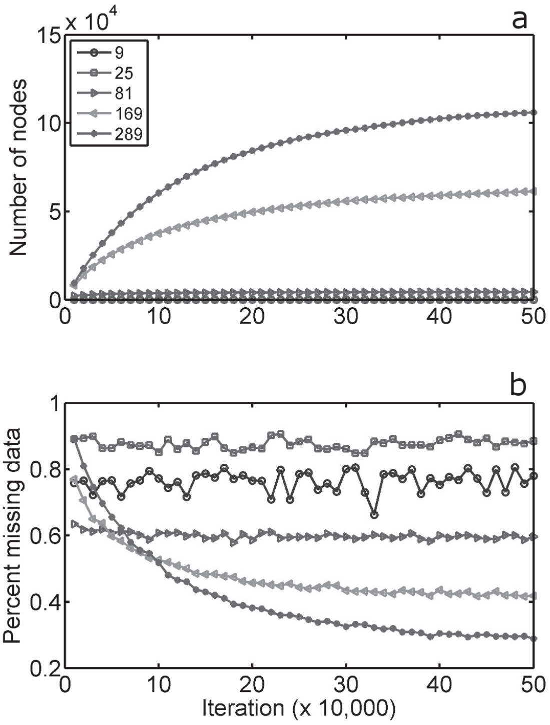

Fig. 1. WKL learning and input space dimensionality. WKL simulations were run on randomly sampled natural image patches of increasing dimensionality (from 3 × 3 to 17 × 17). a) The number of nodes learned as a function of the number of learning iterations clearly increases for high-dimensional spaces but remains roughly constant at low dimensions. b) The resulting input space covering – as measured by the percent test data points not included in the covering – correspondingly improves for high-dimensional, but not low-dimension spaces.

From a computational perspective, incremental competitive learning procedures thus seem inadequate to learn oriented receptive fields from the (high-dimensional) space of natural images. **Fig. 2** shows an example of the kind of receptive fields learned by applying the WKL rule to natural images: few cells seem to display any form of orientation selectivity. Rather, most cells appear to code for some form of surface texture.

Incremental learning procedures thus seem better suited to learn higher-order features formed by combinations of handcrafted oriented edge detectors like Gabor filters or Difference-of-Gaussians [4, 20].

3. Probabilistic WKL

The results of the previous section point out to the difficulty inherent in any attempt to find a complete coverage of the input space in the case of high-dimensional in-

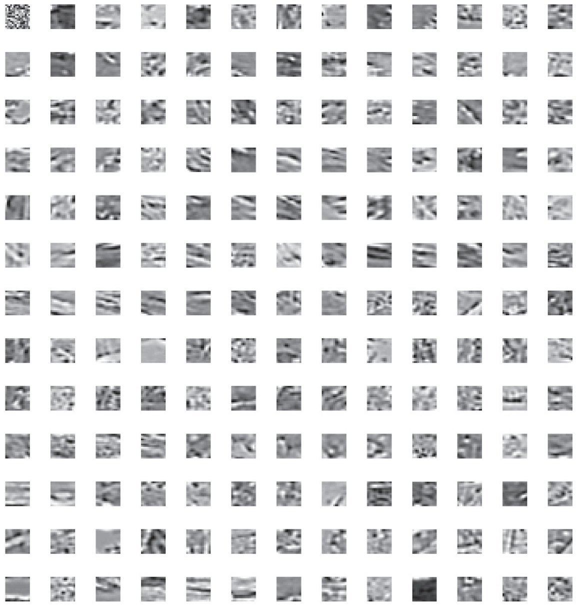

Fig. 2. Example receptive fields learned with WKL learning applied to natural image data. Most cells do not develop a pattern of orientation selectivity.

put vectors. Motivated by these results, we introduce a small generalization of the WKL learning rule that allows a useful form of incremental learning in high-dimensional space. In particular, rather than attempting to cover the entire input space, the proposed learning rule emphasizes learning regularities in the data that closely resemble second-order statistics [39].

The last reference to second-order statistics finds its justification in the following argument. Given that synaptic weights are only modified for the winning unit, and that the winning unit's activation is guaranteed to be above a nonnegative threshold, Eq. (3) can be approximated as follows:

$$\Delta w_i = \lambda x^T \approx \lambda s_i x^T. \quad \ldots\ldots\ldots\ (7)$$

Computing the expectation over many input presentations, assuming the winning unit is not killed at any point during training, the expected weight change is then given by:

$$E\Delta w_i \approx \lambda E s_i x^T = \lambda E \frac{w_j \cdot x \cdot x^T}{||w_j||||x||}, \quad \ldots\ (8)$$

where E is the expectation operator computed over the data distribution. It can be inferred, from a classical result [12] using a learning Equation very similar to Eq. (8), that granted certain assumptions on the learning rate, the weight vector w_j should converge to the first eigenvector of the Gram Matrix $E\left[xx^T\right]$, which is a second order quantity. Although this is not analytically proven here – one would have to ensure convergence properties in view of the possibility of node removal and of the use of an explicit normalization operator stemming from Eq. (1) instead of a weight decay term – the results of **Fig. 3** below suggest that this statement is at least approximately correct.

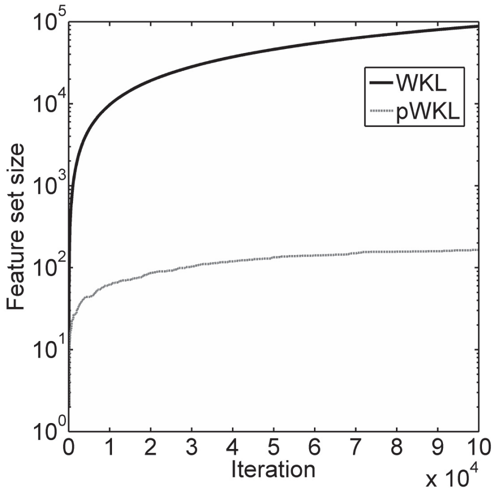

Fig. 3. Comparison of the number of nodes learned with the original WKL and the probabilistic WKL (pWKL) during training. The number of nodes learned with pWKL is clearly less than with the original learning rule.

In order to capture statistical regularities, our first modification to the original WKL rule consists in making the removal of a node inversely proportional to the number of times (n_i) it has won competition in the past. Let P_i denote the probability of removing node i given that its activity is above threshold but below that of the winning neuron. The model presented in [20] was restricted to the case where $P_i = 1$. What we propose is thus to use instead $P_i = \dfrac{1}{n_i}$. The rationale for such a mechanism is that a node that has won many times in the past is likely to cover an important part of the input space and should thus be kept.

The second modification to the WKL rule consists in making new node insertion inversely proportional to the total number of nodes (N). Let P_n denote the probability of inserting a new node given that no existing node has an activity level beyond the threshold θ. The model presented in [20] was restricted to the case where $P_n = 1$, meaning that a new node is certain to be introduced. What we suggest instead is to use $P_n = \dfrac{1}{N^\alpha}$, where alpha is a user-specified parameter. If no new node is inserted, the input pattern is nevertheless learned by the most active unit in the network, despite that its activity is less than θ. This second modification allows the network to maintain a relatively small number of nodes despite the potentially large dimension of the input space while maintaining the incremental nature of the original learning rule. Making node insertion inversely proportional to the number of existing nodes is analogous to the use of a Dirichlet prior in non-parametric Bayesian estimation [40], but without the complex sampling procedures required for statistical consistency [41].

Crucially, the use of a soft criterion in the WKL to de-

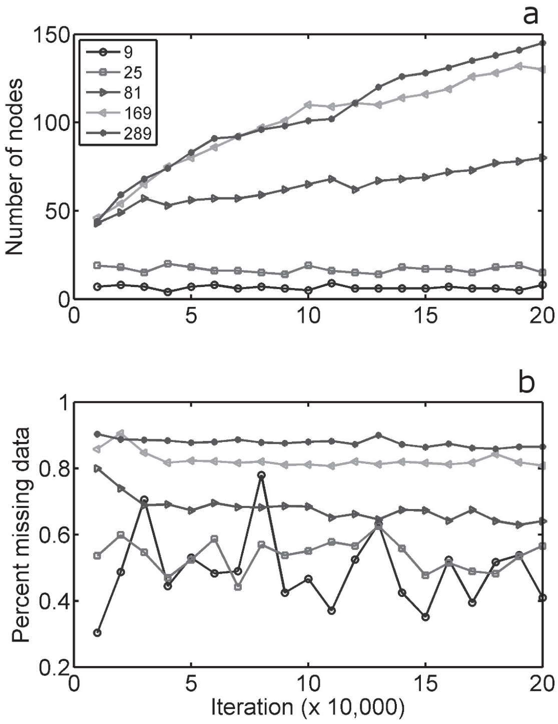

Fig. 4. Probabilistic WKL learning and input space dimensionality. a) Although approximately only half the number of patches was sampled in these simulations compared to the simulations of **Fig. 1**, a substantial reduction in the number of nodes is nevertheless apparent. b) The resulting covering, while much improved in low dimensions, remains poor in higher dimensions.

termine new node insertion allows the number of nodes to adapt to the input distribution, unlike previous learning methods which considered a fixed number of units (e.g. [11, 26, 42, 43]).

Figure 4 shows that running the probabilistic WKL rule (pWKL) over a smaller version of the Monte-Carlo task of **Fig. 1** leads to **(a)** a reduced number of units, and **(b)** a better coverage of the input space in low dimensions. Approximately half the number of patches was sampled per iteration compared to **Fig. 1**. Hence, it is useful to compare the number of nodes at iteration t in **Fig. 4** to the corresponding number at time $\frac{t}{2}$ in **Fig. 1**. For example, comparing the number of units between pWKL at time $t = 2$ and WKL at time $t = 1$ yields 60 Vs 10,000.

Table 1 summarizes the pWKL rule. The only differences with the original WKL rule are that in the latter case, $p = 1$ in steps 3b and 4, and there is no step 5.

4. Learning V1-Like Receptive Fields

In this section we demonstrate that the proposed generalization of the WKL rule is capable of learning oriented edge receptive fields from natural images. The training procedure is analogous to the ones used in typical V1 learning simulations. At each iteration, a 15-by-15 image patch is randomly gathered from a natural image and input to the network. For our simulations we use natural images gathered from a camera attached to the head of a cat as it wanders in a natural environment [44]. As in [45], raw pixel input is first pre-processed with a difference-of-Gaussians filter whose inner and outer spreads are given by 0.875 and 1.4, respectively. Cell activities are then computed according to Eq. (1) and learning proceeds as described above with an additional normalization of the input x in Eq. (3) which further reduces the dimensionality of the input space, for a maximum number of 100,000 iterations. The threshold parameter is fixed at $\theta = 0.65$, and $\alpha = 1.4$.

Table 1. WKL and probabilistic WKL.

```
Input: data $x$, weights $w_i$, $i = \{0, …, N-1\}$, activation frequencies $n_i$, $i = \{0, …, N-1\}$, parameter $\alpha$, learning rate $\lambda$, threshold $\theta$.

1. Compute normalized dot product $s_i$ for each unit (Eq.(1))
2. Find winner $k = \arg\max_i s_i$
3. If $s_k \geq \theta$, adjust winner's weights (Eq.(3)), else go to step 4, otherwise:
       a. Find losers, i.e. $\{l \mid 0 < s_l < s_k\}$
       b. Remove unit $l$ with probability $p \sim 1/n_l$
4. If instead $s_k < \theta$, add a node with probability $p \sim 1/N^{\alpha}$.
5. If no new node is inserted, adjust the winner's weights (Eq.(3))
```

Figure 3 shows the number of nodes learned as a function of the training iteration when using either the original WKL (thick black line) or its proposed generalization (pWKL; thin gray line). The number of nodes learned with the probabilistic WKL is clearly inferior – by a few orders of magnitude – to the number learned with the original WKL. The network also appears to be approaching stability, although simulations run with an even higher number of training patterns should be conducted to confirm whether stability is achieved.

Figure 5 shows the first 169 receptive fields (out of a total of 181) learned with the probabilistic WKL. In comparison to the patterns learned with the original WKL (**Fig. 2**), the probabilistic WKL appears to capture important regularities – mostly in the form of oriented edges – in the natural image input space.

In order to quantify how well the learned receptive fields approximate the near uniform distribution of edges in natural scenes, Gabor patches were fit via least-squares, and the resulting phase/frequency estimates visually inspected for correctness. In 7% of cases the fitting procedure failed to converge despite that a clear orientated receptive field had been learned. In 4% of cases, the receptive fields obtained lacked a clear orientation, and instead resembled the kind of center-surround receptive fields found in undirected V1 cells. Finally, in 14% of

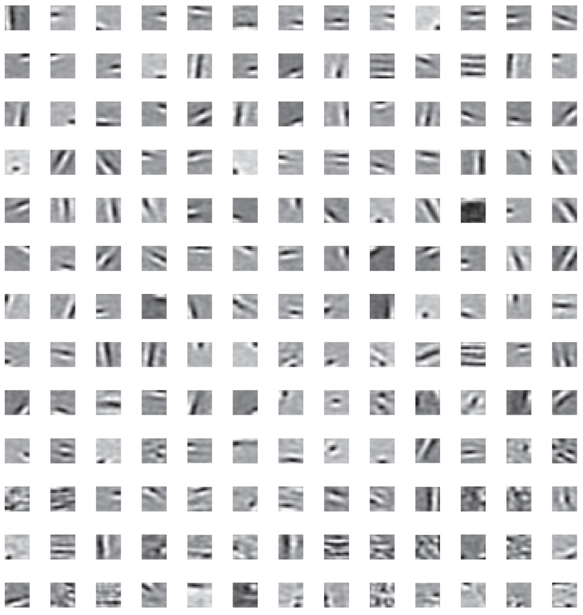

Fig. 5. V1-like receptive fields learned with pWKL. Orientation selectivity can easily be observed in a majority of units.

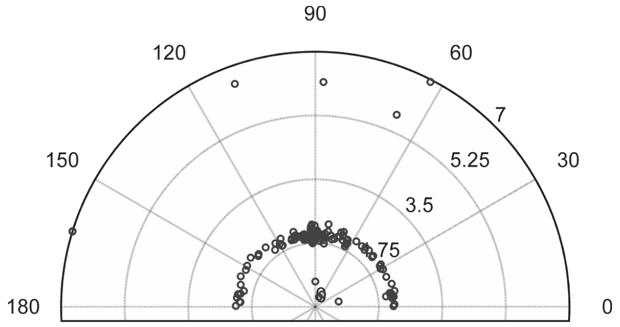

Fig. 6. Log-polar representation of the distribution of phase and frequency of filters learned with pWKL. Phase is indicated by the angle on the polar plot. Log-frequency is indicated by the radial circles. All phases are adequately represented, with a particular emphasis on vertical orientations.

cases the learning algorithm failed to yield a receptive field with a clearly identifiable structure.

The estimates obtained for all remaining receptive fields are plotted on the log-polar plane in **Fig. 6**.

As **Fig. 6** shows, the learning algorithm is able to successfully learn edge filters in all orientations. The learning rule is not as successful at spanning the space of frequencies. More investigations are needed to determine whether this limitation should be ascribed to the learning rule itself or to the pre-processing used here (in particular, to the spatially frequencies of the difference-of-Gaussians filter).

5. Optical Character Recognition Experiments

In this section we embed the pWKL rule in the Neocognitron model [20] and illustrate the resulting model's performance on the ETL1 database.

5.1. ETL1 Database

The ETL1 dataset is a subset of the larger ETL handwritten digits database collected by the Electrotechnical Laboratory – now the National Institute of Advanced Industrial Science and Technology – between 1973 and 1984. The ETL database contains approximately 1.2 million, cropped, handwritten characters and may be considered as an alternative to the MNIST database, the latter containing approximately 70,000 character [46]. Each character is encoded into a 65×65 grayscale input matrix. We use only the ETL1 subset of the full database, which contains 5,000 characters, since that is what earlier simulations of the Neocognitron with the WKL rule used and we wish to provide a fair comparison of these with our candidate pWKL rule.

5.2. Neocognitron

The Neocognitron model [28] is a hierarchical neural network that has been widely used in pattern recognition tasks, especially in the context of optical character recognition. The Neocognitron is composed of a number of layers, where each layer contains two types of cells called S-cells and C-cells, respectively, in analogy to the simple and complex cells found in primary visual cortex [47]. Although multiple refinements of the S- and C- cells have been formulated since the Neocognitron's inception, their main functions may be described as computing correlation filters and pooling operations, respectively. In order to sparsify the output of each model layer, an additional step of competition is implemented across C-cells at different spatial locations. The Neocognitron model was probably the first model to show how layered stages of filtering, pooling, in conjunction with unsupervised learning, could lead to a 2D pattern classifier capable of some spatial invariance. These features are shared with later model such as convolutional networks [1], networks based on the HMAX operator [4] as well as other similar architectures [5]. In the convolutional network literature, the operations of filtering, pooling and competition tend to be referred to as convolution, pooling and normalization, respectively.

The simulations we present here build off of a recent variant of the Neocognitron which contains both divisive and subtractive normalization at the level of the S-cells (interested readers should consult [36] for more details about the model architecture).

5.3. Simulation Procedure

Simulations were run on a Linux cluster of five Sun Ultra-20-M2 workstations with AMD Opteron DC1210 1.4 GHz, 2GB RAM, connected over 1GB Ethernet, and running Hadoop 1.0.3. The original C++ code used in [36] was ported so as to be easily used in Hadoop streaming by changing file I/O to read and write directly from/to HDFS. The simulations reported here only vary a single parameter – namely, the sparseness parameter α in the pWKL equation – and measure its effect on overall accuracy. The sparseness parameter α was varied within the interval $[0,1]$ at increments of 0.2, yielding a total of six parameter configurations. Note that when $\alpha = 0$, the pWKL is actually equivalent to the original WKL rule, which allows us to compare the effect of increased sparseness against the base implementation. Note that we also constrain α to be the same *across model layers*, which can be conceived perhaps as the most stringent test of the use of a sparseness criterion in the Neocognitron. Indeed, it is possible that sparseness may be better used at different levels in the various layers of the Neocognitron, a possibility that we do not address here.

Hadoop allows us to speed-up simulations through *mapReduce*. In the *map step*, each compute node in the cluster is assigned a specific parameter configuration, whereas the *reduce* step only accumulates the computed accuracy values across parameter configurations. Training was performed separately for each layer over 5000 iterations, where each iteration consisted in the presentation of one randomly selected pattern from the ETL1 subset. Overall classification accuracy was computed over a subset of 1,000 patterns.

5.4. Results

We simulated the Neocognitron model with pWKL for various values of sparseness parameter α. **Fig. 7** shows the accuracy computed as a function of α, and as a function of the number of training iterations.

From **Fig. 7**, it is clear that the accuracy on the test set reaches a somewhat stable point already after presentation of only 1000 training patterns, for any value of the sparseness parameter α. Increasing sparseness uniformly across layers of the model actually reduces accuracy slightly by about 3% (when $\alpha = 1$).

On the other hand, **Fig. 8** shows that this comes with a substantial reduction in the number of nodes.

As expected, the higher α is, the fewer units remain in the model. Perhaps what is most surprising, however, is that the 3% decrease in accuracy seen when increasing sparseness uniformly from 0 (which is equivalent to the original WKL) to 1, is accompanied by a 64% reduction in the number of nodes in the network. Hence, it appears that more than half the nodes in the original Neocognitron (i.e. in the Neocognitron trained with WKL) contribute little to nothing to the classification accuracy. One useful future direction of research would be to devise a measure of information – in a spirit somewhat similar to Akaike's information criterion [48] – for non-Bayesian models like the Neocognitron in order to quantitatively assess the relative importance of classification accuracy as a function of the number of network nodes.

Another interesting feature of **Fig. 8** is that the decrease in number of nodes varies across layers, despite that the same α parameter was used. In particular, as α increases the number of nodes kept appears to converge such that it is higher for later layers (i.e. layer 3 has more nodes than layer 2, and layer 4 has more nodes than layer 3). This could indicate an intrinsic difference in the number of spheres needed for a full volume covering as dimensionality expands from layer 2 to 4, since the input space dimension does indeed increase as one goes up in the layers of the network.

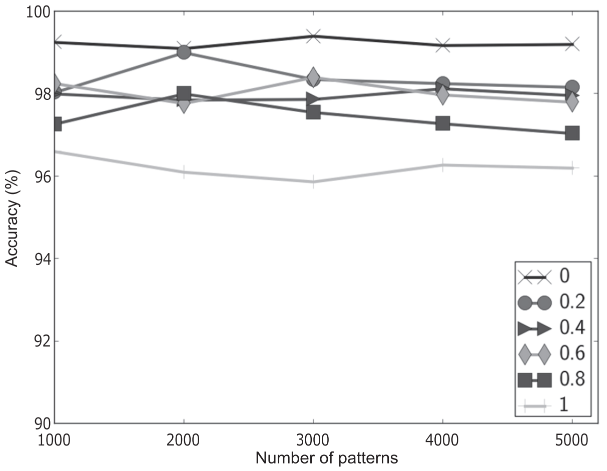

Fig. 7. Classification accuracy over ETL1 as a function of α and of the number of training iterations. Each curve shows the accuracy for a given value of the sparseness parameter $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Accuracy decreases proportionally to the value of α from approximately 99% to 96%.

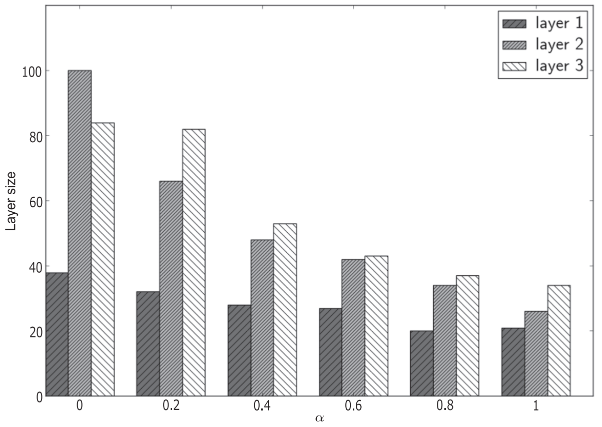

Fig. 8. Number of units per layer, as a function of sparseness parameter α. As expected, increasing α proportionally reduces the number of coding units learned in the model.

6. Discussion

Our primary objective in this article is to see whether a simple incremental learning rule such as WKL can be used to learn statistical regularities in the high-dimensional space induced by natural images, and whether it can be used in the context of classification tasks. Upon presentation of natural images, in its original version, the WKL rule leads to an explosion in the number of learned filters (**Fig. 3**). In addition, most learned filters do not capture essential regularities in the data (**Fig. 2**). On the other hand, the WKL rule learns these regularities when generalized so as to make a given node's removal probability inversely proportional to the number of times it has won competition, and so that new node insertion is inversely proportional to the total number of nodes. The WKL rule learns not only strongly oriented receptive fields, but also, to a lesser extent, undirected receptive fields. The latter resemble the zero-phase (ZCA) filters of [42], and their emergence among a wider group of oriented cells is consistent with the fact that non-oriented, black-white cells are also present in cortical area V1 [49]. **Fig. 5** reveals that learned undirected cells were of the off-center on-surround type exclusively. Future work is needed to understand why this type of cells was learned over on-center off-surround cells.

Both proposed generalizations require minimal changes to the original WKL. In particular, the quantity P_i requires only evaluating local computations. The quantity P_n requires knowing the total number of nodes N which, despite being a global quantity, remains simple to compute. Although this question is beyond the scope of this article, it is possible that such a quantity would be computed implicitly in the brain by considering that the number of neurons in a given cortical volume remains roughly constant.

In this work, the probabilistic terms P_i and P_n were kept as simple as possible. It remains a possibility, however, that the postulated forms limit the range of learned features. For example, using a heavy-tailed function for P_n might lead to more useful features being learned. Conversely, stable learning may be impaired due to the fact that nodes are initially prone to removal due to the high value of P_i.

Although the space of orientations is appropriately covered by the learning rule (**Fig. 6**), variations in frequency do not seem to be well handled by the learning rule. Such a tight clustering of learned frequencies has already been observed when using either Independent Component Analysis (ICA) or sparse coding techniques [50]. It is not clear at present why such a tight clustering would occur.

Simulations show that the proposed learning rule can learn orientation-selective receptive fields – akin to the kinds of receptive fields found in area V1 – as well as perform near the same level of accuracy as the WKL rule on a digit recognition benchmark while drastically reducing the number of units needed to reach that level of performance. It is possible that the pWKL rule shows no accuracy improvement over the WKL rule since the latter's performance on the ETL1 dataset is already near perfect (>99%). As suggested by the results of **Figs. 3** and **5**, accuracy improvements may be more noticeable on tasks whose input space is of higher dimension, such as tasks designed to study general object recognition in natural images [51, 52].

7. Conclusion

In this article, we generalize the WKL rule to learn oriented receptive fields from natural image data and test whether it can lead to improvements on a standard recognition benchmark dataset. Our generalization retains some of the essential benefits of the original WKL – namely its simplicity and biological plausibility – while being able to deal with a high-dimensional input space. The main drawback of the proposed method is that it does not lead to a dense covering of frequency space, and that it does not show any clear improvements in terms of classification accuracy, at least on the ETL1 dataset.

Given that the present results hint to the possibility of learning incrementally the second-order statistics of a an input distribution, three questions that naturally arise and could be addressed in later work include characterizing exactly the convergence properties of weight vectors beyond the simple analysis of Eqs. (7) and (8), deriving a concise energy function for *networks* of pWKL units [16], and using the pWKL rule to learn non-stationary distributions [53].

Acknowledgements

This work was supported in part by the Overseas Research Program of Kansai University for the year of 2010.

References:

[1] K. Jarrett, K. Kavukcuoglu, M.-A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" Proc. ICCV, pp. 2146-2153, 2009.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. of the IEEE, pp. 2278-2324, 1998.

[3] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," Proc. CVPR, pp. 11-18, 2006.

[4] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust Object Recognition with Cortex-like Mechanisms," IEEE Trans. on Pattern Analysis and Machine Intelligence, 29, pp. 411-426, 2007.

[5] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox, "A high-throughput screening approach to discovering good forms of biologically inspired visual representations," PLOS Computational Biology, 5, e1000579, 2009.

[6] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," Proc. ICML, pp. 609-616, 2009.

[7] H. Lee, E. Chaitanya and A. Y. Ng, "Sparse deep belief network for visual area V2," NIPS, 2007.

[8] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P. Koh, and A. Y. Ng, "Tiled convolutional neural networks," NIPS, 2010.

[9] M.-A. Ranzato, F.-J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," Proc. CVPR, 2007.

[10] P. D. Grünwald, "The Minimum Description Length Principle," MIT Press, 2007.

[11] N. Zhang and J. Weng, "Sparse representation from a winner-take-all neural network," Proc. IJCNN 2004, pp. 2209-2214, 2004.

[12] E. Oja, "Simplified neuron model as a principal component analyzer," J. of Mathematical Biology, Vol.15, pp. 267-273, 1982.

[13] J. L. Jr Wyatt and I. M. Elfadel, "Time-domain solutions of Oja's equations," Neural Computation, Vol.7, pp. 915-922, 1995.

[14] P. Földiák, "Learning invariance from transformation sequences," Neural Computation, Vol.3, pp. 194-200, 1991.

[15] E. T. Rolls and T. Milward, "Model of Invariant Object Recognition in the Visual System: Learning Rules, Activation Functions, Lateral Inhibition, and Information-Based Performance Measures," Neural Computation, Vol.12, pp. 2547-2572, 2000.

[16] S. Becker, "Unsupervised learning procedures for neural networks," The Int. J. of Neural Systems, 1-2, 17-33, 1991.

[17] H. Sprekeler, C. Michaelis, and L. Wiskott, "Slowness: An Objective for Spike-Timing-Dependent Plasticity?" PLoS Comput Biol, 3, 2007.

[18] J. Shawe-Taylor, "Symmetries and discriminability in feedforward network architectures," IEEE Trans. on Neural Networks, Vol.4, pp. 816-826, 1993.

[19] J. Léveillé and T. Hannagan, "Learning spatial invariance with the trace rule in non-uniform distributions," Neural Computation, Vol.5, pp. 1261-1276, 2013.

[20] K. Fukushima, "Neocognitron trained with winner-kill-loser rule," Neural Networks, Vol.23, pp. 926-938, 2010.

[21] G. Hinton, "Training product of experts by minimizing contrastive divergence," Neural Computation, Vol.14, pp. 1771-1800, 2002.

[22] A. Hyvärinen, J. Hurri, and P. O. Hoyer, "Natural Image Statistics – A probabilistic approach to early computational vision," Springer-Verlag, 2009.

[23] G. W. Cottrell, P. Munro, and D. Zipser, "Learning internal representations from gray-scale images: An example of extensional programming," Proc. Cognitive Science Society, 1987.

[24] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," J. of Machine Learning Research, Vol.11, pp. 3371-3408, 2010.

[25] C. Farabet, Y. LeCun, K. Kavukcuoglu, E. Culurciello, B. Martini, P. Akselrod, and S. Talay, "Large-scale FPGA-based convolutional networks," R. Bekkerman, M. Bilenko, and J. Langford, (Eds.), Scaling up Machine Learning: Parallel and Distributed Approaches, Cambridge University Press, 2011.

[26] L. N. Cooper, N. Intrator, B. S. Blais, and H. Z. Shouval, "Theory of cortical plasticity," Singapore, World Press Scientific, 2004.

[27] J. Léveillé, I. Hayashi, and K. Fukushima, "Online learning of feature detectors from natural images with the probabilistic WKL rule," 2012 Joint 6th Int. Conf. on Soft Computing and Intelligent Systems (SCIS) and 13th Int. Symp. on Advanced Intelligent Systems (ISIS), 177-182, 2012.

[28] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," Pattern Recognition, Vol.15, pp. 455-469, 1982.

[29] M. Maruyama, G. Federico, and T. Poggio, "A connection between GRBF and MLP," MIT AI Lab Memo AIM-1291, 1992.

[30] M. Kouh and T. Poggio, "A general mechanism for tuning: Gain control circuits and synapses underlie tuning of cortical neurons," MIT AI Lab Memo 2004-031, 2004.

[31] S. Grossberg, "Contour enhancement, short-term memory, and constancies in reverberating neural networks," Studies in Applied Mathematics, 52, 1973.

[32] T. Kohonen, "Self-organized formation of topologically correct feature maps," Biological Cybernetics, 43, pp. 59-69, 1982.

[33] J. A. Hartigan, "Clustering algorithms," New York, John Wiley & Sons Inc, 1975.

[34] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," Cognitive Science, 11, pp. 23-63, 1987.

[35] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature, Vol.381, pp. 607-609, 1996.

[36] K. Fukushima, I. Hayashi, and J. Léveillé, "Neocognitron trained by winner-kill-loser with triple threshold," ICONIP, 2011.

[37] J. H. Conway and N. J. A. Sloane, "Sphere packing, lattices and groups," New York, Springer-Verlag, 1988.

[38] J. A. Lee and M. Verleysen, "Nonlinear dimensionality reduction," Springer, 2007.

[39] G. Hinton, "To recognize shapes, first learn to generate images," Progress in Brain Research, Vol.165, pp. 535-547, 2007.

[40] Y. W. Teh, "Dirichlet processes," Encyclopedia of Machine Learning, Springer, 2010.

[41] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," J. of Computational and Graphical Statistics, Vol.9, pp. 249-265, 2000.

[42] A. J. Bell and T. J. Sejnowski, "The independent components of natural scenes are edge filters," Vision Research, Vol.23, pp. 3327-3338, 1997.

[43] R. Mikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh, "Computational maps in the visual cortex," Springer, 2005.

[44] B. Betsch, W. Einhäuser, K. Körding, and P. König, "The world from a cat's perspective – statistics of natural videos," Biological Cybernetics, Vol.90, pp. 41-50, 2004.

[45] T. Masquelier, T. Serre, S. J. Thorpe, and T. Poggio, "Learning complex cell invariance from natural video: a plausibility proof," CBCL Paper. Massachusetts Institute of Technology, Cambridge, MA, 2007.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. of the IEEE, Vol.86, Issue 11, pp. 2278-2324, Nov. 1998.

[47] D. H. Hubel and T. N. Wiesel, "Receptive Fields Of Single Neurones In The Cat's Striate Cortex," J. of Physiology, Vol.148, pp. 574-591, 1959.

[48] H. Akaike, "A new look at the statistical model identification," IEEE Transactions on Automatic Control, Vol.19, Issue 6, pp. 716-723, 1974.

[49] M. S. Livingstone and D. H. Hubel, "Anatomy and physiology of a color system in the primate visual cortex," J. of Neuroscience, Vol.4, pp. 309-356, 1984.

[50] Y. Karklin and M. S. Lewicki, "Is early vision optimized for extracting higher-order dependencies?" NIPS, 2005.

[51] G. Griffin, A. Holub, and P. Perona, "The caltech-256 object category dataset," Technical Report, Caltech , 2007.

[52] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," Int. J. of Computer Vision, Vol.88, pp. 303-338, 2010.

[53] M. Sugiyama and M. Kawanabe, "Machine learning in non-stationary environments: Introduction to covariate shift adaptation," MIT Press, 2012.

Name:
Jasmin Léveillé

Affiliation:
Center of Excellence for Learning in Education, Science and Technology, Boston University

Address:
677 Beacon Street, Boston, Massachusetts 02215, USA

Brief Biographical History:

2001-2002 M.Sc. Evolutionary and Adaptive Systems, University of Sussex

2005-2010 Ph.D. Cognitive and Neural Systems, Boston University

2010-2012 Postdoctoral Associate, Department of Cognitive and Neural Systems, Boston University

Main Works:

- J. Léveillé, and T. Hannagan, "Learning spatial invariance with the trace rule in non-uniform distributions," Neural Computation, Vol.5, pp. 1261-1276, 2013.
- J. Léveillé, and A. Yazdanbakhsh, "Speed, more than depth, determines the strength of induced motion," J. of Vision, Vol.10, 1-9, 2010.

Name:
Isao Hayashi

Affiliation:
Professor, Graduate School of Informatics, Kansai University

Address:
2-1-1 Ryozenji-cho, Takatsuki, Osaka 569-1095, Japan

Brief Biographical History:
1981-1983 Sharp Corporation
1987-1993 Panasonic Corporation
1993-2004 Hannan University
2004- Kansai University

Main Works:
- "Vitroid – The Robot System with an Interface Between a Living Neuronal Network and Outer World," Int. J. of Mechatronics and Manufacturing System, Vol.4, No.2, pp. 135-149, 2011.

Membership in Academic Societies:
- International Fuzzy Systems Association (IFSA)
- The Institute of Electrical and Electronics Engineers (IEEE)
- Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
- Japanese Neural Network Society (JNNS)

Name:
Kunihiko Fukushima

Affiliation:
Senior Research Scientist, Fuzzy Logic Systems Institute

Address:
680-41 Kawazu, Iizuka, Fukuoka 820-0067, Japan

Brief Biographical History:
1958-1989 NHK (Senior Research Scientist, Broadcasting Science Research Lab., etc.)
1989-1999 Professor, Osaka University
1999-2006 Professor, University of Electro-Communications, then Tokyo University of Technology

Main Works:
- "Artificial vision by multi-layered neural networks: Neocognitron and its advances," Neural Networks, Vol.37, pp. 103-119, Jan. 2013.

Membership in Academic Societies:
- Institute of Electronics, Information and Communication Engineers (IEICE)
- International Neural Network Society (INNS)
- Japanese Neural Network Society (JNNS)