Coarse and Fine Learning in Deep Networks

Anthony Knittel Alan Blair
aek,blair@cse.unsw.edu.au
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia

Abstract— **Evolutionary systems such as Learning Classifier Systems (LCS) are able to learn reliably in irregular domains, while Artificial Neural Networks (ANNs) are very successful on problems with an appropriate gradient. This study introduces a novel method for discovering coarse structure, using a technique related to LCS, in combination with gradient descent. The structure used is a deep feature network, with a number of properties of a higher level of abstraction than existing ANNs, for example the network is constructed based on co-occurrence relationships, and maintained as a dynamic population of features. The feature creation technique can be considered a coarse or rapid initialization technique, that constructs a network before subsequent fine-tuning using gradient descent. The process is comparable with, but distinct from, layer-wise pretraining methods that construct and initialize a deep network prior to fine-tuning. The approach we introduce is a general learning technique, with assumptions of the dimensionality of input, and the described method uses convolved features. Results of classification of MNIST images show an average error rate of 0.79% without pre-processing or pre-training, comparable to the benchmark result provided by Restricted Boltzmann Machines of 0.95%, and 0.79% using dropout, however based on a convolutional topology, and as such our system is less general than RBM techniques, but more general than existing convolutional systems because it does not require the same domain assumptions and pre-defined topology. Use of a randomly initialized network provides a much poorer result (1.25%) indicating the coarse learning process plays a significant role. Classification of NORB images is examined, with results comparable to SVM approaches. Development of higher level relationships between features using this approach offers a distinct method of learning using a deep network of features, that can be used in combination with existing techniques.**

I. Introduction

A number of paradigms have been proposed for extracting higher order features from an observed environment and capturing its underlying structure. These include Convolutional Neural Networks (CNNs) [1], Deep Belief Networks [2] and Learning Classifier Systems [3].

Artificial Neural Networks operate using gradient descent of an objective function, allowing parameters of the system to be gradually updated to improve accuracy as new instances are observed [4]. Recent techniques such as pre-training [5] give a variation to this approach, where learning is first conducted using an alternative objective, to reconstruct observed inputs, before training the network to capture classifications using gradient descent methods. This approach is very accurate and reliable on particular problem sets, however can have difficulty in domains with an irregular objective function.

Learning Classifier Systems [3], and other forms of evolutionary algorithms, operate using a population of features that each capture aspects of the problem space, and explore solutions using recombinations of existing features, and a form of reinforcement learning to capture the relationship between features and outcomes. These approaches can be reliable for addressing irregular domains, however without the precision seen in gradient descent methods. This study introduces the use of gradient descent, with a feature network that is constructed using population-based methods related to Learning Classifier Systems. The use of a deep feature network is introduced, in contrast to the redundant feature representation commonly used by evolutionary approaches. This allows the precision of gradient-based learning used in ANNs to facilitate the learning provided by the LCS learner. In addition this provides a coarse learning technique that allows capturing structure in irregular domains, before the use of gradient-descent to fine tune the solution.

The objective of a deep feature network is to capture structure of the environment being observed. In evolutionary systems, structure is captured in building-blocks [6], [7], that are shared between classifiers in the population. Building-blocks are copied between classifiers and stored redundantly, rather than being maintained in a shared manner where features are re-used by multiple classifiers. Relationships between building-blocks and constituent components are not preserved, and as such are best considered a shallow rather than deep representation.

Deep feature networks have been captured in a number of Artificial Neural Network designs. In approaches such as [8], the first layer of the network captures basic properties such as oriented edges, while features have been demonstrated in subsequent layers that capture increasingly complex structure, such as features resembling eyes and mouths, while views of faces can be identified in higher levels. Recognizable larger structure is not commonly addressed, and in [8] this structure is presented as a demonstration, while classification results are based on the simpler first few layers of the network.

Capturing hierarchical structure has been done very effectively using generative methods on a small number of layers, and with limited input dimensions, however further increases in depth and complexity of representation have not been explored, and do not appear to provide immediate practical advantages. Current approaches using fully-connected layers such as Restricted Boltzmann Machine [2] and Autoencoder networks [5], do not appear to be able to scale to capture larger and more abstract representations. The development of deep structure is a key aspect of deep network design, although limited examination of learning based on more flexible part based representations and image grammars has been shown. A second objective of this study is to introduce a representation that captures a deep network of features at a higher level of abstraction than used in existing neural networks, as a step towards the development of a feature network capturing deeper and more varied part-based relationships. This objective is connected to the introduction of a coarse learning process to construct the feature network, as the representations used by Learning Classifier Systems are based on a more abstract representation than connectionist systems.

A. Deep feature networks

Artificial Neural Networks (ANNs) are based on local processing and connectivity, reflecting to some degree relationships between neurons in the brain. Cognitive properties identified at the neural level have provided practical benefits for artificial systems, such as the model of structures of the visual cortex used in Convolutional Neural Networks (CNNs) [1]. This is based on a model of simple and complex cells such as that in [9], and reflects aspects of visual processing that act in a bottom-up manner.

In addition to properties of interactions between individual neurons, recognizable processes have been found that act on a broader scale, as captured in cognitive models such as ACT-R [10]. These models describe systematic relationships between identifiable structures, although the described structure is abstract and identified from behavioural observations, and there is often limited understanding of the manner in which they relate to processes at the neural level. Models of visual cognition also describe more complex interactions than the bottom-up effects captured in CNNs. These include effects that occur at specific timescales, such as early interpretation of context, and means of providing top-down facilitation to influence interpretations [11]. Each of these processes are characterized by independent learning objectives, and processes that act on a broader scale than the interaction between neurons at a local level.

Aside from the use of back-propagation, further independent processes have been addressed in deep learning systems, such as the use of pre-training to identify features based on unsupervized learning, and top-down effects in Deep Belief Networks [2], that capture influences on lower level activations as a result of activation properties of higher level features.

The top-down connections in Deep Belief Networks address a form of top-down effect, based on direct relationships between elements in neighbouring layers. More detailed models describe the role of context in the interpretation of details [11], where contextual interactions are largely governed by associative interactions [12], and act using recognizable local rules. Such processes play a significant role in allowing the rapid identification of relevant objects from a vast collection of knowledge. Capturing such effects in artificial systems requires the ability to introduce processes that act on different scales, and according to independent learning rules, governed by recognizable local properties. In some cases this implies specialization of design, to capture effects specific to a domain such as vision specific properties used by CNNs, however there are many common traits in higher level processes, suggesting they reflect more general principles of learning [13].

In this study an independent learning process is used, to allow the development of features, according to locally defined processes that act on a coarse scale. This provides a means of constructing feature relationships based on observations, using co-occurrence relationships, and the maintainance of a population of features according to reinforcement with use[1]. Learning at a coarse scale allows the development of more abstract properties, such as grammar-based and logical relationships of higher level features. This provides an important step towards the broader goal of capturing modularity in artificial learning.

II. Self-organising systems at higher levels of abstraction

Learning Classifier Systems [3] are self-organising systems, that act on properties at a higher level of abstraction than ANNs. They are considered ‘evolutionary’ systems and are typically based on Genetic Algorithms, however they were originally conceived as abstractions of cognitive processes [14]. Learning Classifier Systems use a form of reinforcement-based learning, both in terms of Q-learning style Reinforcement Learning to update predictions on multistep problems, and Psychology related reinforcement to influence maintenance of the population[2], and as such it is possible to view these algorithms in the context of cognitive processes. Connections between Reinforcement Learning methods and cognitive processes are well established [15], and there are further similarities between the population reinforcement methods used and models of abstract cognitive processes, such as reinforcement of memory traces [16], [10]. The term ‘Evolutionary Computing’ is commonly used to describe this family of techniques, and the terminology will be maintained to refer to LCS and related systems, however the evolutionary paradigm is not significant for this study. ‘Evolutionary’ systems can be effective and reliable for optimization on irregular learning tasks, providing an effective form of coarse learning, however without the same capabilities of fine-tuning available to gradient-descent based methods.

The implementation provided in this study addresses the use of reinforcement based methods to capture coarse learning in a self-organising deep network. This allows more abstract properties of learning to be captured, based on local learning rules acting on a different scale, alongside gradient descent methods that allow fine-tuning.

[1] Note that the term ‘reinforcement’ used here is more closely related to the use of the term in Psychology, such as relating to reinforcement of memory traces, rather than the ‘reinforcement’ referred to in Computer Science literature relating to Reinforcement Learning.

[2] however the term ‘fitness’ is commonly use in GA based systems

The coarse learning process presented constructs a form of grammar of features, creating new features based on relationships between existing features, according to observed co-occurrences resulting from observations. These relationships are captured using weights and nodes of an Artificial Neural Network. This allows learning to take place on two scales, using reinforcement processes related to LCS to reinforce and select features in a fixed size population, while weights are adjusted through gradient descent.

A. *Coarse learning design*

Learning Classifier Systems (LCS) are typically based on logical structures. Many use real-valued representations, however classification rules are interpreted as matching the observation in a discrete manner. The population of classifiers contains general and specific rules, where a given observation may be matched by a number of classifiers of various degrees of specialization. Common features are captured in building blocks, and are preserved through crossover in the genetic algorithm. Classifiers can also be defined using a hierarchical structure, employing a population of re-used features, where features are recorded discretely instead of redundantly, and recombined in various combinations to produce classification rules [16]. The hierarchical approach maintains features according to reinforcement, and creates new features based on combinations of existing ones, without the use of a genetic algorithm. Lowest level features, known as 'atomic features', can be tested for activation directly against the observed environment. Higher level features, known as 'composite features', capture relationships between lower features, which may be either atomic features or other composites. Features can be produced with varying degrees of specificity, and are connected to classification terms in a manner similar to other LCS rules. This structure allows the definition of classification rules with varying degrees of specialization, where additional specialization can be produced from a general feature by the addition of further properties. A representation of a hierarchical feature network, as used in this study, is shown in Figure 1.

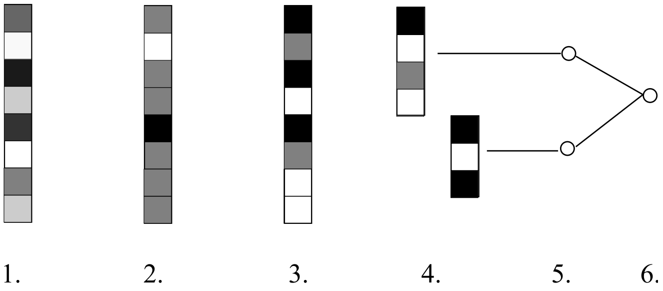

Fig. 1. 1. a continuous valued input state, 2. a general classifier with two binary values and six general 'don't care' values, 3. a more specialized classifier with only two generalized values, 4. re-used features that match part of the state, 5. general classifiers each produced from a single re-used feature, 6. a more specialized classifier constructed from the conjunction of the two general features.

The population of features is dynamic, with a fixed maximum population size of atomic and composite features. Atomic features are tested against various positions of the input, allowing a feature to be identified in various locations, and as such act in a convolutional manner, producing an activation value for each position, captured in a match map. Features are constructed as templates from observations, producing a representation that responds to a region of a given observation.

Composite features are defined using a set of lower level features, which may be atomic or composite, and a vector defining their relative positions. New composites are constructed as a random set of the active features for a given observation, based on the relative positions of activity observed. This is captured as an 'and' relationship, generated from an observed co-occurrence of features.

To identify the active features for a given observation, each atomic feature is tested at each position of the input, producing an activation map. The activation of composite features is achieved by examining the activation values of each child at its relative position, producing an activation map for the composite.

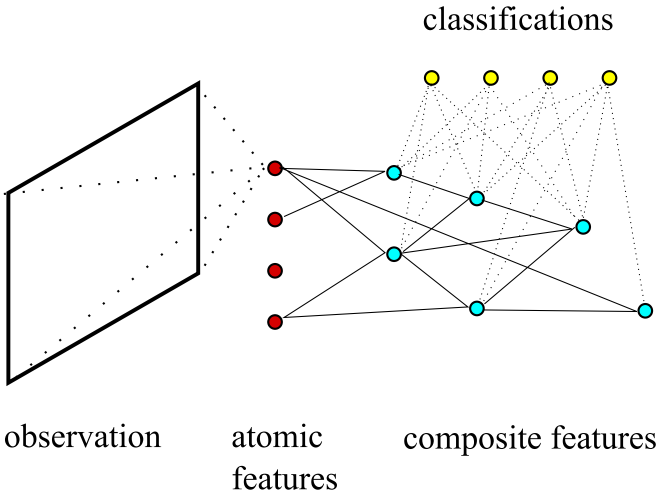

Fig. 2. Relationships between features in the constructed network.

In most LCS systems, reinforcement of classifiers is based on expected reward, representing the probability that a given classifier will accurately predict the outcome [17]. In the hierarchical representation, features are re-used between various classifications, and as such a measure of expected reward is less meaningful. Reinforcement takes place following an analogy with cognitive memory traces, that are reinforced through use, as described in the ACT-R model [10].

The population of re-used features is maintained according to a measure of *accessibility*. This is related to base-level activation in ACT-R, determined as a quantity that is increased with use, and decreases over time, captured as a summation and exponential decay (Equation 1). To provide more stable values this has been simplified as a running average (Equation 2). A constant quantity of reinforcement r_t

is provided each time step, in the current implementation this is provided to the feature with the highest predictive value and its child elements. The predictive value is determined using a 'softmax' calculation for each feature, to determine the probability value of the feature predicting the correct class. Distributed reinforcement approaches produce similar behaviour, however the use of reinforcement of the highest predictive feature has the advantage of producing a recognizable relationship between generalized and specialized features. Specialized features, which have a higher predictive quality, will be given preference for reinforcement, and the limited size population will necessitate a balance between general and specialized features.

$$B_i = \log \sum_{j}^{n} t_j^{-d} \tag{1}$$

$$\Delta f_t = \alpha(r_t - f_{t-1}) \tag{2}$$

B. Gradient descent model

Our implementation captures the construction of a gradient descent network based on the feature grammar constructed using the above method of creation and reinforcement, and is referred to as an Abstract Deep Network. Each feature is represented as a unit of the network, and as such the topology of the network is altered as new elements are produced. The topology is altered as new atomic features or composites capturing connections between existing units are produced, and as features are removed. This approach is distinct from the method used by NEAT systems [18]. NEAT is an evolutionary approach for construction of a neural network topology using Genetic Algorithms, where each member in the population is a complete network, and refinement takes place by selection over multiple independent networks. In contrast, our method is based on a population of features, where each is independently selected for usefulness according to local rules, based on the accuracy and frequency of use of the feature. The structure of the network is shown in Figure 3.

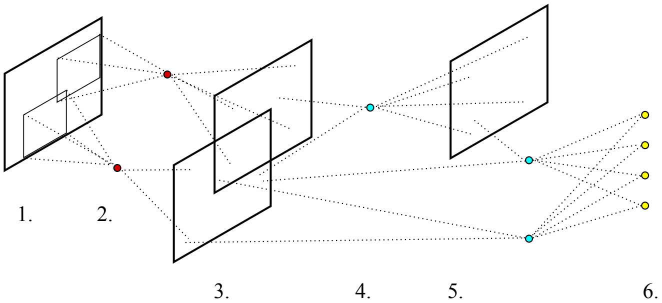

Fig. 3. Structure of the feature network, including convolutional maps. 1. observed input, 2. atomic features, 3. activation map, 4. composite features, which are constructed from a number of child atomic or composite features, 5. composite activation map, 6. softmax output, connected to all composite features.

Weights of new atomic features are created according to an observed region, and are chosen such that the response of the feature to the observation matches a constant value k, by initialising weights according to observed values, and normalising such that the sum of squares is k. Feature activation values are continuous, and in order to capture features being active or inactive, as used by LCS systems, a threshold value is employed, however in practice this is simply used to limit the features considered in the construction of new features.

For composite features, the logical relationship between elements described in the LCS feature network is approximated by the weights of the network elements, capturing a soft 'and' relationship. When a new composite is created, each weight is set to the current activation value of the child feature, and the weights normalized such that the square sum is a constant k. By initialising the bias value to $-(k - \epsilon)$, the composite is initialized to be active approximately when the conjunction of the child elements are active.

The relationships between features and classifications are captured in fully-connected links between each composite and a 'softmax' classification layer. This produces a topology that is not strictly layered, and is necessary due to the self-organising topology of composite features.

As each feature produces an activation value for each position, the mapping between the composite feature layer and the classification layer is based on a single activation value for each feature, using the maximum value. To perform backpropagation, adjustment of child elements is performed based on the activation value for the position of the highest value in the top-level feature, and the activation value for the relative position of each child feature. Further details of the backpropagation method are described in [19].

C. Specialization and generalization

This implementation captures 'and' relationships, producing a hierarchy of features of increasing specialization. There may be advantages in using an 'and-or' structure, as it allows a greater variety of representation including invariance between higher level structures, however an 'and'-based network is used to limit complexity. 'And-or' properties are seen in cognitive structures, such as in the ventral stream of the visual cortex, and a simplified representation is captured in models of simple and complex cells, such as Convolutional Neural Networks, based on a pre-defined topology.

The development of 'and-or' networks to capture grammatical representations of images based on parts has been studied in [20], addressing the development of higher level structures than those typically captured in artificial neural networks. This is based on an analytical approach, using retrospective analysis of the network as a whole to develop the feature network. The method described in this paper addresses a similar objective, the ability to capture higher level structural relationships, however this is done using self-organising principles based on local rules.

Candidate 'and' relationships can be identified from observed co-occurrences of features. Meaningful 'or' relationships are more difficult to identify through local operations, as individual instances are not indicative of significant disjunctions, and it is likely that retrospective analysis of some kind is necessary to identify meaningful 'or' relationships. As such a population based on 'and' relationships, representing increasing specialization of features, is addressed in this study.

A balance of generalized and specialized features is produced by the reinforcement process. Based on the current LCS implementation this can be considered as the maximally specialized population of rules sufficient to cover the observed environment. Other approaches such as XCS [17] emphasize maximally general rules, preferring a general rule over a specialized one if the expected outcome is similar, however this is based on direct examination rather than a self-organising operation based on reinforcement. LCS systems produce overlapping features, where a population contains general representations as well as specializations that provide refinements for particular examples. This is related to cognitive basic level and subordinate categorization, such as a 'bird' and a 'penguin', where a specialized category captures specific properties that are different from the general class [21]. The described implementation captures specializations where advantage is provided over a general representation, through preferential selection and reinforcement. The use of feature representations composed from parts allows incremental specialization.

While a number of broad goals have been outlined, the current study explores the interaction between a coarse learning process for creating and reinforcing features in a hierarchical structure, and fine-tuning through gradient descent. This is addressed as a general learning system, with limited assumptions about the domain. Low level features respond to a limited region of the input and are convolved on the input space, and relationships between features are defined using a vector relationship, implying a particular dimensionality. Further assumptions such as those used in CNNs, including connectivity between higher features based on a limited spatial region, or specific operations such as a fixed topology of simple and complex cells, such as pooling operations, are not included, to allow learning based on limited self-organising principles.

III. Experiments

A. MNIST

The MNIST dataset of handwritten digits has been used as a benchmark for many different techniques, and is based on a large set of 70,000 images for training and testing, of size 28x28. Training of our system was performed in two phases. The first phase involved use of the population reinforcement method for creation and selection of features, to develop the network topology. This acted in tandem with the gradient descent method, providing adjustment to the feature weights. This phase was conducted for 2×10^5 sample presentations, to construct the network topology based on the coarse learning technique. Subsequent fine-tuning was conducted with the topology fixed, acting using only the gradient descent method to adjust weights.

Two network structures are examined. The first uses a large set of 1500 atomic (first level) features and 5000 composites (approx 2×10^5 weights total), the second uses 100 atomic features and 10,000 composites (1.4×10^5 weights). These configurations are referred to as 1500A-5000C and 100A-10000C respectively. The first network allows a larger number of feature maps, while the second relies on the re-use of a limited set of features through composition relationships.

The network with a large set of atomic features showed an average error rate of 0.79% (classification errors on each run out of 10,000: 63, 69, 71, 71, 80, 81, 81, 83, 83, 83, 89, 93), while the small set network showed an average error of 0.86%. Further runs were conducted using a randomly generated topology comparable to that produced by the self-organising system, with randomly initialized weights. The use of random initialization produced an average error rate of 1.25%. A minibatch size of 100 was used, however similar results were found using stochastic updates, with greater variation. Results shown are based on Rectified Linear Units (ReLU) [22], similar performance results were found using hyperbolic tan units, however ReLUs provided slightly faster convergence and less processing. No pre-processing of data is used, other than to scale inputs with mean zero and $\sigma = 1$.

Previous results using standard ANN techniques are typically limited to approximately 1.6% error, while experiments using Restricted Boltzmann Machine based systems have shown an error rate of 0.95%, and 0.79% using 'dropout' [23], described as a record for systems without prior knowledge or enhanced training sets. Lower error rates have been shown with systems that use significant pre-processing, or are based on a specific topology, for example 0.6% has been shown using pre-training and sparse feature selection in a convolutional network [24].

Our system uses less assumptions than existing convolutional networks, however it is not as general as RBM based systems. Our result is comparable to that shown by RBM techniques using dropout, also without the use of enhanced training sets. Some further assumptions have been included in our design, based on the assumed dimensionality of the input, and the use of shared weights to allow convolution of features. The topology used by our system is self-organising, and does not make use of specific functions common to CNNs, such as pooling and local contrast normalization. The topology and functions used by CNNs are a fundamental aspect of their design, to the extent that the specific use of rectification and normalization functions can produce top-level results even when random features are used [25]. The Convolutional Deep Belief Network [8] provides another approach (0.82% error), where pre-trained features are used in a convolutional architecture, however this approach focuses on the development of low level features that are used by a Support Vector Machine, with a kernel function specialized towards image domains [26]. In contrast our approach is focused on the development of higher level relationships, with minimal domain-specific assumptions.

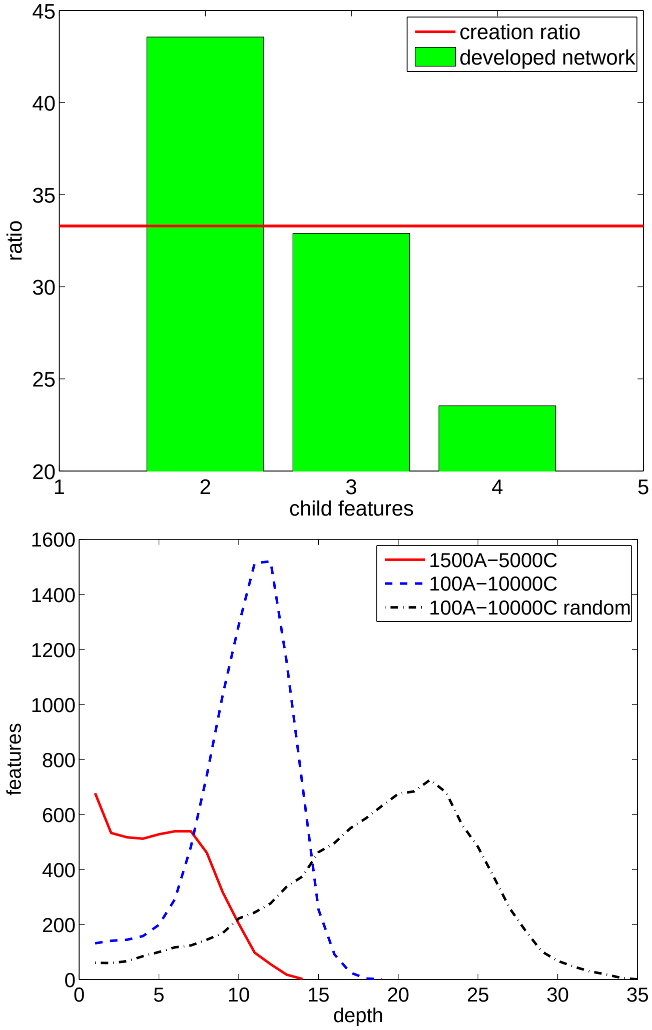

Fig. 4. (top) Distribution of the number of child elements per composite feature in the developed network, and the distribution at feature creation, showing self-organising preference for fewer child elements. (bottom) Depth distribution of elements in the developed 1500A-5000C network, the developed 100A-10000C network, and a randomly constructed 100A-10000C network using a similar creation process, showing preference for lower depth.

Details of the network produced by our system are shown in Figure 4. The developed network shows the distribution of the number of child elements per feature skewed towards a smaller number than the creation distribution, resulting from selective pressure, and indicating a self-organising preference towards a smaller branching factor. The number of child elements per feature is several orders of magnitude smaller than that used in layer-wise fully connected networks, while the depth of the network is greater. The depth distribution is shown for the network produced using 1500A-5000C features, for the network produced using 100A-10000C, and for a randomly constructed network using 100A-10000C. This shows that a much flatter network is produced when a larger number of atomic features are available. The use of 100A-10000C features in a self-organized network is deeper, however compared to the distribution from a randomly constructed network using the same creation method, the average depth is smaller. This is a result of a bias from co-occurrences of active features, and selective preference towards a shallower network. This may be indicative of a point where addition of specialization features to existing representations produces structures that are not useful and are not reinforced, related to ‘terminal features’ [27].

‘Dropout’ has been shown to improve performance and generality of ANNs at a cost of increased training time, improving the error rate of a standard feedforward network from 1.6% to about 1.3% using dropout on the hidden units, and about 1.1% using dropout of visible units. Introducing dropout into our system showed significantly degraded performance. This may be due to the use of weights initialized to represent conjunction relationships, as inhibited activation of units will lead to greater reduction in activation of parent nodes. Our network captures sparse relationships using a different approach, and as such the random inhibition behaviour of dropout does not appear to be beneficial.

B. *Normalized-uniform NORB*

Tests were conducted on the *normalized-uniform* (small) NORB dataset, with minimal changes to the model and parameters where possible. This dataset is a set of images based on photographs of objects at various angles, of size 96x96 stereo images. Results using Support Vector Machines have shown 11.6%, logistic regression 19.6%, k-nearest-neighbours 18.4%, and Convolutional Neural Networks 5.6% error [28], [29]. Previous general approaches using neural networks have used reduced dimension inputs for tractability. Using a ‘foveal’ representation of dimension 72x72, [30] showed results of 6.5% error using a multi-layer network with a greedy pre-trained first layer, without the use of data augmentation. Further results using 32x32 subsampling have shown 8.9% error using pre-training and fine-tuning [31].

We have used a similar model as for the MNIST example, using 100 atomic (first layer) features, and 10,000 sparse higher level features. These parameters may be considered a meta-model, as the actual topology is self-organising. Our model has shown 11.96% error. This result is promising considering a largely general model has been used, with minimal domain assumptions or parameter tuning, and without the use of pre-training.

First level features produced from the MNIST and NORB datasets are shown in Figure 5.

IV. Discussion

Our system has demonstrated the use of a deep feature network constructed using a coarse learning method, in combination with gradient descent learning, acting as an Artificial Neural Network. This provides a novel means for constructing the network topology and for initialising the network, that shows faster learning and higher accuracy than a similar network initialized randomly. The method demonstrated uses convolutional features, however fewer

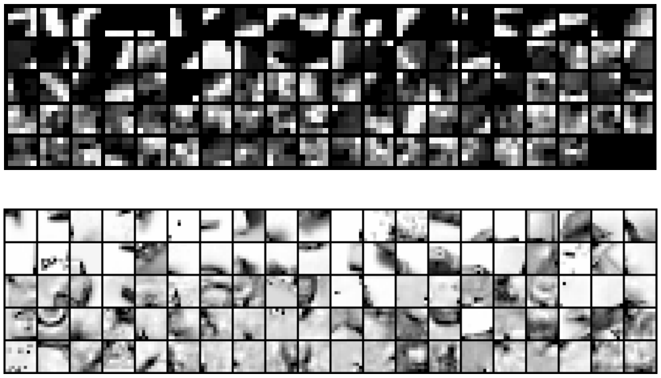

Fig. 5. First-level features developed from the MNIST dataset (top), and NORB (bottom)

domain-specific assumptions are used than Convolutional Neural Networks. Our system addresses a different problem, emphasising discovery of higher level relationships between features, and the combination of coarse and fine learning, rather than discovery of low level features. As such a complementary relationship with existing systems may be found.

The topology produced is deep and very sparse. This is a property of the process that develops the network, a larger branching factor has not been used as there is a clear preference for smaller numbers of connections with each composite feature. The depth of the network is mostly a reflection of the creation process and the number of composite units, as new features are constructed based on observed co-occurrences of active features, and in order to capture a comparable number of weights with other systems a large number of units are used.

On each of these tasks improved performance can be found using more specialized techniques, particularly CNNs, which introduce assumptions about the connectivity of the network and the use of specific functions at each layer. In order to develop advances in artificial learning it is important to explore both domain-specific and more general learning techniques.

Studies of vision-specific CNNs have shown that a critical aspect of performance is the choice of topology, notably the use of absolute value rectification, local contrast normalization and average pooling [25], and that such a structure allows top-level results regardless of the feature development method used. This occurs to the extent that the use of random features within the given topology provides results comparable to those with more sophisticated features.

A contrary perspective in [32] shows that benchmark results can be obtained with simple models such as a single-layer topology, when specific choices of hyperparameters and pre-processing are used. Does this mean that the method of feature development and the processes involved are irrelevant? While benchmark results can be found through such widely varying approaches on standard datasets, it is not necessarily the case that the same approaches will be effective as the scale and complexity of problems change. One of the principles behind deep learning is to capture functional and representational modularity that allows the structure of a problem or environment to be captured [28], in order to provide advantages in terms of scaling and versatility. As such addressing representational and functional issues is a way of maintaining these goals, to work towards allowing autonomy in more challenging environments.

In human cognition, rapid processing is used to identify context and establish expectations of objects that are likely to be present. The brain is able to store information on an immense range of experiences and retrieve relevant information quickly, and it is likely that the processes used, along with the cognitive biases they exhibit, such as the misinterpretation of objects according to context [33], [12], are significant for allowing such tasks to be performed. As such, capturing the interaction between specific operations that act using independent local rules is likely to offer important benefits in addressing larger and more complex tasks. The use of coarse learning based on reinforcement, and the development of higher level feature relationships, are a step towards the broader aims of capturing cognitive processes that operate at higher levels of abstraction, such as contextual effects that provide top-down facilitation [13].

Recent experiments in computer vision have shown that increasing the amount of computing power enable CNNs to scale to allow identification of a large collection of objects, and with a higher resolution than those used in tasks performed on desktop computers [34]. This is a promising indication of the effectiveness of the techniques, however open questions remain regarding the ability to combine and integrate information, and to be able to develop functional and representational modularity. More versatile algorithms will likely be needed for these tasks, our approach aims to provide a contribution towards these goals, by presenting alternative means for feature discovery and representation.

V. Conclusions

Capturing modularity and higher level effects is an important goal in allowing learning systems to address more complex tasks based on generalized learning techniques. Artificial Neural Networks are an effective technique for capturing relationships based on local operations, although they are typically based on optimization of a single objective function, with limited modularity. Recent techniques such as pre-trained networks introduce independent learning goals of feature discovery and fine-tuning, however other effects such as coarse learning processes require further abstraction.

This study has demonstrated the use of a coarse learning process, related to more abstract cognitive processes than neural level effects commonly addressed in ANNs. The system acts using population-based measures related to Learning Classifier Systems. This allows the development of a feature network using local operations that act on a broader scale, related to the construction of symbolic grammars. The feature network is captured using the weights of a neural network, providing a coarse learning process that constructs and alters the topology of the network. This introduces a novel approach for capturing self-organising principles at a higher level of abstraction. This provides a means for identification of features in a deep network before fine-tuning takes place, related to the task addressed by pre-trained neural network techniques. Experiments have shown high level results on the benchmark MNIST task, showing a classification accuracy of the same order as the best general learning techniques (0.79% error), and with greater generality than specialised CNN techniques. Current RBM and CNN techniques show significant improvement over the first results presented by the techniques (1.23% and 0.95% with no distortions respectively). The self-organising network developed by the coarse learning process provides a significant improvement over a similar random network using randomly initialized weights (0.79% vs 1.25%), indicating a significant role for this process in the early stages of learning.

The implementation has shown the development of a representation capturing features of increasing specialization, where the population maintains a balance of generalized and specialized rules. The method provides a means of addressing irregular domains with ANNs, and a means of introducing gradient-descent based learning with ‘evolutionary’ systems. This provides a new approach for capturing modularity with the aim of allowing more scalable, flexible learning with general learning algorithms.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, nov 1998.

[2] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[3] J. Drugowitsch, *Design and Analysis of Learning Classifier Systems: A Probabilistic Approach*. Berlin: Springer, 2008.

[4] D. E. Rumelhart and J. L. MacClelland, *Parallel distributed processing*. IEEE, 1988, vol. 1.

[5] Y. Bengio, A. Courville, and P. Vincent, “Unsupervised feature learning and deep learning: A review and new perspectives,” *arXiv preprint arXiv:1206.5538*, 2012.

[6] D. E. Goldberg, *The Design of Innovation : Lessons from and for Competent Genetic Algorithms*. Boston: Kluwer, 2002.

[7] M. V. Butz, M. Pelikan, X. Llorà, and D. E. Goldberg, “Automated global structure extraction for effective local building block processing in XCS,” *Evolutionary Computation*, vol. 14, no. 3, pp. 345–380, 2006.

[8] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *International Conference on Machine Learning (ICML)*. New York, NY, USA: ACM, 2009, pp. 609–616.

[9] M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nature neuroscience*, vol. 2, pp. 1019–1025, 1999.

[10] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, “An integrated theory of the mind,” *Psychological Review*, vol. 111, no. 4, pp. 1036–1060, 2004.

[11] M. Bar, “Visual objects in context,” *Nature Reviews Neuroscience*, vol. 5, no. 8, pp. 617–629, 2004.

[12] M. Bar, “The proactive brain: using analogies and associations to generate predictions,” *Trends in Cognitive Sciences*, vol. 11, no. 7, pp. 280–289, 2007.

[13] A. Knittel, “Abstract representations in deep networks, capturing rapid and top-down cognitive processes in artificial learning,” Ph.D. dissertation, University of New South Wales, 2013.

[14] J. Holland and J. Reitman, “Cognitive systems based on adaptive algorithms,” *ACM SIGART Bulletin*, no. 63, p. 49, 1977.

[15] R. Samson, M. Frank, and J.-M. Fellous, “Computational models of reinforcement learning: the role of dopamine as a reward signal,” *Cognitive Neurodynamics*, vol. 4, no. 2, pp. 91–105, 2010.

[16] A. Knittel, “An activation reinforcement based classifier system for balancing generalisation and specialisation (ARCS),” in *Proceedings of the 12th annual conference on genetic and evolutionary computation*. New York, NY, USA: ACM, 2010, pp. 1871–1878.

[17] M. Butz and S. W. Wilson, “An algorithmic description of XCS,” in *Revised Papers from the Third International Workshop on Advances in Learning Classifier Systems*. London, UK: Springer-Verlag, 2001, pp. 253–272.

[18] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[19] A. Knittel and A. Blair, “An abstract deep network for image classification,” in *AI 2012: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, M. Thielscher and D. Zhang, Eds. Berlin, Heidelberg: Springer, 2012, vol. 7691, pp. 156–169.

[20] Z. Si and S.-C. Zhu, “Learning AND-OR templates for object recognition and detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, p. 1, 2013.

[21] J. Tanaka, “The entry point of face recognition: evidence for face expertise.” *Journal of Experimental Psychology: General*, vol. 130, no. 3, pp. 534–543, 2001.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.

[24] M. A. Ranzato, C. Poultney, S. Chopra, Y. L. Cun *et al.*, “Efficient learning of sparse representations with an energy-based model,” in *Advances in neural information processing systems*, 2006, pp. 1137–1144.

[25] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2146–2153.

[26] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classification with sets of image features,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1458–1465.

[27] S. Fidler, M. Boben, and A. Leonardis, “Similarity-based cross-layered hierarchical representation for object categorization,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[28] Y. Bengio and Y. LeCun, “Scaling learning algorithms towards AI,” in *Large-Scale Kernel Machines*, ser. Neural Information Processing Series, L. Bottou, Ed. MIT Press, 2007, pp. 321–360.

[29] Y. LeCun, F. J. Huang, and L. Bottou, “Learning methods for generic object recognition with invariance to pose and lighting,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–97.

[30] V. Nair and G. E. Hinton, “3d object recognition with deep belief nets,” in *Advances in Neural Information Processing Systems*, 2009, pp. 1339–1347.

[31] H. Luo, R. Shen, C. Niu, and C. Ullrich, “Learning class-relevant features and class-irrelevant features via a hybrid third-order rbm,” in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 470–478.

[32] A. Coates, A. Y. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[33] A. Torralba, “Contextual priming for object detection,” *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.

[34] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, “Building high-level features using large scale unsupervised learning,” *arXiv preprint arXiv:1112.6209*, 2011.