Results for the SENSE project

1- A brief introduction

The SENSE project aims at designing a novel bio-inspired neuromorphic embedded system that mimics the principles of human vision from the sensor (retina) to high-level computation algorithms, through the application of the last advances in neural coding. Classical sensors are no more valid in our model. We need information/signals as provided in the mammalian eye. This is the first step to really apply the biological model in the preprocessing layer. Hence, this project originates from human retina that is one of the most well understood parts of the human vision system being itself one of the most studied cortical areas. Retina is valuable to understand how information is sensed and processed by complex neural circuits in the brain. Indeed, retina is part of the central nervous system and has a synaptic organization similar to those of other central neural structures. Moreover, the retina contains only five major classes of neurons interconnected in a complex fashion but with an orderly, layered anatomical arrangement making this organ relatively simple compared with other brain regions.

Based on the recently proposed sparse neural model named ENN, we propose to design a radically different vision system composed of three layers: perception, pre-processing and computation. SENSE aims at a paradigm shift from conventional CMOS/CCD sensors followed by traditional image processing step towards a vision system inspired by human vision. Such tightly integrated and multi-layered smart vision system will be organized into a stacked architecture that will use complex and quite understandable ENN models at each level conferring unexpectedly powerful image processing. To be specific, in the human brain, visual perception begins in the retina and occurs in two stages. Light rays entering the eye are converted into electrical signals by specialized sensory organs (retina photoreceptors i.e. cones and rods and then retina neurons i.e. bipolar, horizontal, amacrine and ganglion cells). This first step will be realized by designing a smart CMOS sensor embedding on-chip analog ENN model. These electrical signals are then sent through the optic nerve to higher centers in the brain for further processing necessary for perception (i.e. lateral geniculate nucleus located in the Thalamus, primary visual cortex, higher visual cortical areas and other brain cortical areas). This second step will be realized by designing a dedicated digital architecture implementing a hierarchical ENN in order to further abstract visual information (color hue, color saturation, color brightness, edge detection...). Finally, in order to make the SENSE vision system tunable to target specific image processing applications, a software ENN model coupled with original and dedicated image processing algorithms will be executed on a programmable many-core architecture. In addition, an artificial learning process will be defined to further enhance the flexibility of the proposed vision system.

2- Results for the Analog step

The architecture of the analog ENN is in fact mixed. A single 30-synapse 128-fanal cluster has been implemented. The implemented decision rule is the winner-takes-all (WTA). To design a clique-based network, the analog circuit is meant to be controlled by a programmable digital circuit (FPGA). These fanals are used to define the analog neural network.

The CMOS 65nm analogue encoded-neural network was sent to the foundry in late 2015. The 25 encapsulated dies were received in April 2016.

(cf our final report)

The first tests with this chip has been performed (and published) with this test configuration. We perform automatic recognition of ElectroenCephaloGram (ECG) decomposed in 5x5 pixels patterns. In each pattern, 2 pixels have been erased (results obtained in 406 cycles, 10MHz). Our first experiments show that we achieve 90% of good reconstruction with our chip.

These first results could be improved with an enhanced ENN model.

3- Results for the Digital step

A. The original GBNN model has been renamed ENN (Encoded Neural Network) but what are we talking about?

An ENN is an abstract neural network model based on sparse clustered networks that can be used to design associative memories. The principle of ENN is to gather sets of neurons in clusters. Neurons (also called *fanals*) that belong to the same cluster cannot be connected to each other (this principle is called *sparcity*). However, any neuron of a given cluster can be connected to any neuron in any other cluster. More precisely, the network consists of N binary neurons arranged into C equally-partitioned clusters and each cluster includes $L=N/C$ neurons. Each cluster is associated through one of its neurons with a portion of an input message to be learned or retrieved. A message m of K bits is thus divided into C clusters each with L fanals and the length of the sub-message associated with each cluster is $X = K/C = log2(L)$.

During the learning phase, the network memorizes that the set of activated neurons (i.e. the set of neurons that constitute the input message) are connected to each other and form a *clique*. Unlike Hopfield networks, ENN uses binary weighted connections between neurons to record if a connection exists or not. Then, to memorize a connection that exists between one neuron i of cluster j, ni,j and one neuron k from cluster g $n_{k,g}$ (with $j \neq g$), each neuron stores locally the value *'1'* in its corresponding synaptic weight w (i.e. $w_{(i,j)(k,g)}=1$ in cluster j and $w_{(k,g)(i,j)}=1$ in cluster g). All the weights are initialized to 0, i.e. no message has been learnt before training.

The retrieving process is based on a Scoring step and a Winner Takes All (WTA) step to detect which neuron, in a cluster associated to a missing part of the message, is the most "stimulated" one. Equation (1) defines the original scoring function used to compute the "score" of a neuron $n_{i,j}$ at time instance $t+1$, $s^{t+1}n_{i,j}$. This score depends on all the values of the other neurons k from all the other clusters g (i.e. $n_{k,g}$) in the ENN computed at time instant t (i.e. in the previous iteration of the network) and on the corresponding synaptic weights (i.e. value of $w_{(k,g)(i,j)}$) that have been stored during the learning process.

(cf our final report for the equation 1)

The WTA equation (2) allows defining in each cluster c_i the neuron $n_{i,j}$ or the group of neurons to activate i.e. the neuron or the group of neurons that achieves the maximum score S_{max}.

(cf our final report for the equation 2)

The network converges in a few time instances i.e. iterations. At the end of the process, the clusters which originally had no selected neuron are provided with the selection of a neuron or a group of neurons. The answer of the network is then defined by the set of neurons that were chosen to represent each single cluster.

The next figure presents an ENN neural network based on 3 clusters of 3 neurons. Let us consider that the network has learned three messages represented by cliques: $(n_{1,0}, n_{0,1}, n_{0,2})$, $(n_{2,0}, n_{1,1}, n_{0,2})$ and $(n_{2,0}, n_{2,1}, n_{0,2})$. These cliques are represented by using a binary synaptic-weight matrix (also called "*Interconnection matrix*") thanks to a classical representation of adjacency matrix as depicted in this figure. Each line of the matrix contains the weights storing the existence -or not- of connections between the neurons of a cluster and the neurons of other clusters. There is no element in the diagonal since the neurons of a given cluster cannot be connected between them. Note that this means that these connections are not represented in the synaptic weights matrix (i.e. the diagonal of the matrix is missing). For example, the message $(n_{1,0}, n_{0,1}, n_{0,2})$ is memorized in the matrix through weights:

(cf our report for the example)

Next, if a partial message $(\_, n_{1,1}, n_{0,2})$ is presented to the network, where _ denotes a missing symbol (i.e. sub-message associated to cluster *0* is missing), then the network must take a decision. The values of the known neurons are first activated (i.e. the values of the neurons associated to known sub-messages are set to 1) and the values of all the neurons are then broadcasted through the network. At the end of the scoring step, neurons $n_{1,0}$ and $n_{2,0}$ in cluster c_0 have a score of 1 and 2 respectively. Indeed, neuron $n_{2,0}$ receives two non-null values since it is linked to two active neurons (i.e. neurons $n_{1,1}$ and $n_{0,2}$) while neuron $n_{1,0}$ receives only one non null value (from neuron $n_{0,2}$). Hence, at the end of this iteration, neuron $n_{2,0}$ will be selected as the activated neuron by the Winner Take All algorithm.

B. What about ENN based architecture for SENSE?

First of all, it appears that the classical version of the ENN model was not smart enough to meet our expectations. This was not new for us, since we originally planned to explore the possible evolutions of the ENN model in order to be able to use it in the SENSE vision system.

However, our work in order to explore and to define these new formal aspects (and associated hardware architectures) has been performed thanks to another PhD thesis (i.e. not funded by the SENSE project). This ENN model offers a storage capacity exceeding the one issued from Hopfield networks when the information to be stored has a uniform distribution. Methods improving performance for non-uniform distributions and hardware architectures implementing the ENN networks were proposed. However, on one hand, these solutions are very expensive in terms of hardware resources and on the other hand, the proposed architectures can only implement networks with fixed sizes and they are not scalable. The objectives of this thesis are: (1) to design ENN inspired models outperforming the state of the art, (2) to propose architectures cheaper than existing solutions and (3) to design a generic architecture implementing the proposed models and able to handle various sizes of networks.

The results of these works are : (1) the concept of clone based neural networks and its variants has been explored and the proposed Clone-based ENN offers better performance than the state of the art for the same memory cost, even when a non-uniform distribution of the information to be stored is considered. The hardware architecture optimizations have also been introduced and they demonstrate significant cost reduction in terms of resources. Finally, a generic scalable architecture able to handle various sizes of networks is proposed.

C. ***How to use ENN based architecture for vision?***

To design embedded visual systems, two complementary axes can be considered. The first is the conception of new hardware architectures able to efficiently implement vision algorithms. This task has been explored in collaboration with a post-doctorant. The second is the conception of algorithms that are less resource hungry in terms of computations and storages.

In our work, we propose less complex models for visual processing based on our ENN and efficient numeric architectures to implement them. The models we are working on are connectionist models. They are computing models based on networks of small processing elements inspired by biological neural networks. Our work is then divided into two parts, each one targeting a specific task of computer vision.

In the first part we consider the nearest neighbor search. Given a query vector, the goal is to retrieve in a large set of vectors the closest ones using a given metric like Euclidean distance. It is a well-known function in computer vision and it is used for applications like image retrieval, descriptor matching and non-parametric classification. Depending of the application, a vector can be a visual descriptor or a set of features representing an image. To perform that task, we improve our existing model of associative memory (based on ENN). An associative memory is a storage system where the stored data are accessed using their content or a noisy version of it rather than an index. That model uses binary neurons and connections and offers a large storage capacity. Those properties allow design energy efficient systems. The original model cannot be used for the nearest neighbor search. Therefore, we propose several evolutions of the original model to perform nearest neighbor search and we evaluate the impact of these evolutions on the hardware architectures.

In the second part, we consider the problem of image classification. Image classification consists in associating an index to an image according to their content. In terms of model, we use deep convolutional neural networks (DNNs) which are the models that own the highest performance regarding the classification accuracy. While DNNs give the best classification accuracy, they are resource hungry (memory and computation) and cannot be implemented on embedded systems. Several works solve that problem by compressing the network after training it. For that, they use several methods like vector quantization after the training phase. The compression process reduces the resources required by the network but it also leads to a degradation of the performance. To reduce the complexity of such network while maintaining high classification accuracy we propose to tightly couple the compression step with the training phase, in a DNN using on our enhanced ENN model.

4- Results for the Software step

Initially, the ENN model has been validated on randomly generated messages, following a uniform distribution (RGU). But, since we are looking to work with images, the distribution becomes non uniform and implies a new challenge called ambiguity.

Indeed, learning non uniform data such as images, or randomly generated data following a Gaussian distribution (RGNU) with an ENN implies that few neurons will support all the information in its memory. Hence, the retrieval phase will not be able to dissociate properly which neurons should be activated in order to represent the correct message, it is an ambiguity. The last

publications in the literature rely on the reduction of the ambiguities given RGNU data. The contributions done by IRISA was also on how to reduce the ambiguities and how to let the ENN generalize its information. We thus list the contributions by their problematic.

A. Ambiguities

A first step was to dissociate learnt messages by expanding the network in layers. By generalizing the expansion, the model is able to get better results than the original ENN with RGU and RGNU data.
Another contribution was to use the structural form of a message to be used as constraint in the retrieval phase. Indeed, since the output must be a clique, by its definition, all the neurons are fully connected, hence their number of connection, ie their degree, is the same. Therefore we define a *penalization* term in order to let the neurons to be activated if and only if they are fully connected to others with the same degree needed to be a clique. We used this penalization on images giving compelling results and it can even eradicate the ambiguities in specific cases. This has led to a publication in the international conference ESANN (European Symposium on Artificial Neural Networks).

B. Generalization

While these methods can reduce, and even cancel the ambiguities, the network is still unable to recover unknown messages. We designed a new model of network to build an abstraction of the messages learnt, in order to help the ENN to generalize its information. It is called Bidirectional Clique based Associative memory (BCAM). The initial version is composed of 2 layers connected by a bidirectional matrix. The first layer is composed of sub CbNNs associated to each patches from a decomposed image. This layer will extract the features based pixel of the images. Then, the second layer will encode the connections between the patches to get an upper level of information over the images. This is how we create the generalization. The retrieval phase is done bidirectional between the layers. With the use of this model, we greatly improve the capability of the network to generalize the information. It is to be submitted in the Transactions on Neural Network journal (TNN).
The use of only memory to do image processing directly on the pixels is difficult because of the dimension. Hence we designed a framework with an ENN to work on extracted features by an Autoencoder. This neural network is composed of 3 layers and aims to recover in the output layer the image from the input layer by the extracted features of the hidden layer. We use the sparse neural activations in the hidden layer as a sequence to be encoded in an ENN. Because if we submit degraded data, the autoencoder can recover until a specific rate of degradation. Our aim is to use what we learnt from the activation during the learning phase, and recover those by an ENN. But since it is unable to generalize, a BCAM is used instead.

C. Classification

All these contributions aim to recover information from an associative memory and then be classified by a classifier such as a Softmax. But the ENN can classify data by itself. During the encoding in the network's memory we allocate a part for the images' labels. The classification is done by recovering this last missing part of the information using the ENN's correction mechanics.

5- A first try for SENSE Vision System?

This step has been performed at this time. It mainly consists of the integration of the different stages of the SENSE system and the production of SENSE vision system prototypes along with an experimental validation of the prototype on some test applications.
At this stage, the genericity and reusability of the sensor for different feature extraction and vision tasks must be evaluated and possible improvements for the design and configuration of the different layers of the physical to software architectures could be proposed.