Projection learning vs correlation learning: from Pavlov dogs to face recognition

Dmitry O. Gorodnichy
Institute for Information Technology (IIT-ITI)
National Research Council of Canada (NRC-CNRC)
Montreal Rd, M-50, Ottawa, Canada K1A 0R6
http://iit-iti.nrc-cnrc.gc.ca

***Abstract*— Face recognition in video is an example of the problem which is outstandingly well performed by humans, compared to the performance of machine-built recognition systems. This phenomenon is generally attributed to the following three main factors pertaining to the way human brain processes and memorizes information, which can be succinctly labeled as 1) non-linear processing, 2) massively distributed collective decision making, and 3) synaptic plasticity.**

Over the last half a century, many mathematical models have been developed to simulate these factors in computer-based systems. In this presentation, we attempt to formalize the recognition process, as it performed in brain, using one of such mathematical models, within which the projection learning appears to be a natural improvement to the correlation learning. We show that, just as the correlation learning rule, the projection learning can also be written in incremental form. By taking in account the past data and being non-local, this rule however provides a way to automatically emphasize more important parameters and training data over the less important ones.

The presented model, while providing a simple way to incorporate the main three factors of biological memorization listed above, is very powerful. This is demonstrated by incorporating it into a face recognition system which is shown to be capable of recognizing faces in video under conditions considered to be very difficult for traditional Von-Neumann-type recognition systems.

I. Modeling associative process

Lets consider the task of associating one stimulus (lets call it Receptor R) to another (Effector E). One famous example of this task is the Pavlov dogs experiment, within which the dogs trained on a buzz(R)-food(E) stimuli were shown to salivate at the sound of the buzz (E) even when no food was presented.

Another example of this task, from our everyday life and which remains to be one the most difficult tasks for the computer to do, is associative the visual stimulus of a person's face (R) to a certain feeling (disgust, likeness) or knowledge (race, level of IQ, name) (E).

To model this association process, lets consider synapses Cij which, for simplicity and because we do not know exactly what is connected in the brain to what, are assumed to connect all parameters of stimuli pair R and E among each other. (For a general model, several layers between A and B, each connecting to one another, may be considered, which can be simulated by adding extra parameters into the aggregated simulus vector.)

These synapses have to be adjusted in the training stage so that in the recognition stage, when sensing R, which is close to what the system has sensed before, as a result of the collective decision making based on the trained synaptic values, a sense of the missing corresponding stimulus E is produced. Mathematically this can be written as follows.

Let $\vec{V} = (Ri, Ei)$ be an aggregated N-dimensional vector made of *all* binary decoded parameters of stimuli pair R_i and E_i. Binarizaion is needed to take into account the non-linear nature of data processing in brain. The synaptic matrix $\mathbf{C}$, which is an NxN matrix, has to be computed so that the collective decision making, based on the incomplete (or noisy) versions of one training stimulus $\vec{V}^m$ (m=1..M), produces the values most similar to those used in training, where the decision making process is based on summing all input parameters weighted by the synaptic values and possibly performed several times until the consensus is reached. This can be written as follows

$$\vec{V}_{noisy} = ((Ri, \not{R}i), \not{E}i) \tag{1}$$

$$\vec{V}(1) = f(\mathbf{C}\vec{V}_{noisy}), \tag{2}$$

$$\vec{V}(t+1) = f(\mathbf{C}\vec{V}(t)), \tag{3}$$

$$\text{until} \qquad \vec{V}(t^*) = f(\mathbf{C}\vec{V}(t^*)) \tag{4}$$

The last equation expressed the stability condition which describes the situation of the reached consensus. $f(\cdot)$ is a binarizing function, such a Heavyside one.

The resulting stimulus $\vec{V}(t^*)$ is then decoded into receptor and effector components: $V(t^*) = (Ai(t^*), Bi(t^*))$ for further analysis of the result of the performed association.

The main question arises: How to compute synaptic values Cij so that the retrieved sense of effector stimulus (B) is the closest to the effector stimulus used in training. Ideally computations of synaptic values, commonly referred to as a learning rule, should be done so that 1) they would not require the system to go through the already presented stimuli (i.e. there are no iterations involved), and 2) it would update the synapses based on the currently presented stimuli pair without knowing which stimuli will follow (i.e. no batch mode involved). These two conditions represent the idea of *incremental learning*: each synaptic weight Cij undertakes a small increment dCij, the value of which, either positive or negative, is determined by the training stimuli pair vector

(V^m):

$$C_{ij}^0 = 0 \tag{5}$$

$$C_{ij}^m = C_{ij}^{m+1} + dC_{ij}^m \tag{6}$$

It is clear that increments dC_{ij}^m should be a function of the current stimulus ($\vec{V}^m$) and what has been previously memorized (which is decoded in synaptic matrix $\mathbf{C}$), i.e.

$$dC_{ij}^m = f(\vec{V}^m, \mathbf{C}). \tag{7}$$

II. Correlation learning rules

Within the formalization of the problem described above, correlation learning, which updates Cij based on the correlation of the corresponding parameters i and j of the training stimulus:

$$dC_{ij}^m = \alpha V_i^m V_j^m, \quad 0 < \alpha < 1 \tag{8}$$

is one of the most frequently used. It makes the stimuli used in training ($\vec{V}^m, m = 1...M$) the minima locations of the Hamiltonian

$$E(\vec{Y}(t)) \doteq -\frac{1}{2}\vec{Y}^T(t)\vec{S}(t) = -\frac{1}{2}\vec{Y}^T(t)\mathbf{C}\vec{Y}(t), \tag{9}$$

which is known [3] to govern the evolution of the dynamic system defined by the update rule of Eq. 6. It can be seen however that this rule essentially makes a default assumption that all training stimuli are equally important and all parameters i are equally important too, which is practically never true. As a result, the associative capability of this type of learning is poor.

Therefore, for better performance other rules, which take into account the history of learning, are used. One of most known of these is the Widrow-Hoff delta rule, which is used in back-propagation multi-layered perceptrons:

$$dC_{ij}^m = \alpha V_i^m (V_j^m - S_j^m) \quad 0 < \alpha < 1 \tag{10}$$

or another version of it

$$dC_{ij}^m = \quad \alpha (V_i^m - S_i^m)(V_j^m - S_j^m) \tag{11}$$

where $S_j^m = \sum_{i=1}^N CijVj^m$ is the postsynaptic potential computed in accordance with the update rule of 6.

It can be seen that this rule does use the knowledge of the past experience (as it uses C), but is not perfect either, since the learning rate α is the same for all training stimuli, regardless of whether the stimulus is useful or not. Nevertheless, this rule is one of the most frequently used in neural computation, where it is mainly used iteratively; rather than applying one-step increments to all synapses, this rule is executed several times on the entire training sequence, until dC_{ij}^m becomes sufficiently close to zero. However, this iterative nature of this rule makes it unacceptable for some applications such as, for instance, real-time face memorization from live video, as it assumes that all training stimuli are always available or stored somewhere, which in many cases in not possible.

This is why another incremental learning rule is needed which would incorporate the level of usefulness of each stimulus as well as the importance of each parameter of the stimulus. This rule is the projection learning rule.

III. Projection learning

The *projection learning rule*, also known as the *pseudo-inverse learning rule* is obtained from the condition that the consensus, based on the collective decision making, is reached for all training stimuli [9], [8], [4]. Using Eq. 6 this leads to the following system of equations needed to be resolved for unknown synapse values:

$$\mathbf{C}\vec{V}^m = \lambda_m \vec{V}^m, \quad (m = 1, ..., M), \lambda_m > 0, \tag{12}$$

which can be rewritten in matrix form for $\lambda_m = 1$ as

$$\mathbf{CV} = \mathbf{V}, \tag{13}$$

where $\mathbf{V}$ is the matrix made of column prototype vectors (training stimuli). Resolving this matrix equation gives us the rule:

$$\mathbf{C} = \mathbf{V}\mathbf{V}^+, \tag{14}$$

where $\mathbf{V}^+ \doteq (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$ is the *pseudoinverse* of matrix $\mathbf{V}$.

The matrix defined by Eq. 14 is the orthogonal projection matrix into the subspace spanned by the prototype vectors $\{\vec{V^m}\}$, which explains the name of the rule.

At first, this rule may look like a non-incremental batch rule, which needs to know all prototypes prior to learning. This however is not necessarily true. As shown in [**?**], [**?**], by using the the Greville formula [1], this rule can be easily rewritten as a one-step incremental rule as

$$\mathbf{C}^0 = \mathbf{0} \tag{15}$$

$$\mathbf{C}^m = \mathbf{C}^{m-1} + \frac{(\vec{V}^m - \mathbf{C}^{m-1}\vec{V}^m)(\vec{V}^m - \mathbf{C}^{m-1}\vec{V}^m)^T}{\|\vec{V}^m - \mathbf{C}^{m-1}\vec{V}^m\|^2} \tag{16}$$

or in scalar form as

$$C_{ij}^0 = 0, \quad C_{ij}^m = C_{ij}^{m-1} + dC_{ij}^m \tag{17}$$

$$dC_{ij}^m = \tfrac{1}{E^2}(V_i^m - S_i^m)(V_j^m - S_j^m), \quad \text{where} \tag{18}$$

$$E^2 = N - \textstyle\sum_{i=1}^N V_i^m S_i^m, \quad S_k^m = \sum_{i=1}^N C_{ik}^{m-1} V_i^m \tag{19}$$

If $\vec{V}^m = \mathbf{C}^{m-1}\vec{V}^m$, which means that the prototype $\vec{V}^m$ is a linear combination of other already stored prototypes, then the weight matrix remains unchanged.

It can be seen now that the projective learning, as described by Eq. 18, looks very much like the correlation learning described in the previous section. It however takes more into account of what is already known and, as a result, provides a better associative recall for the model.

For iterative training, the following approximation of it can be used:

$$C_{ij}^m = \frac{\alpha}{E^2}(V_i^m - S_i^m)(V_j^m - S_j^m) \quad 0 < \alpha < 1 \tag{20}$$

IV. From watching a video to saying a name

There are many steps leading from video frame capturing to detection of a face, to facial feature extraction needed for the creation of an aggregated training / testing stimulus vector $\vec{V} = (R, E)$ used in the neural model described above. While all steps related to video-processing are described in detail elsewhere [5], here is how we obtain the stimuli vectors.

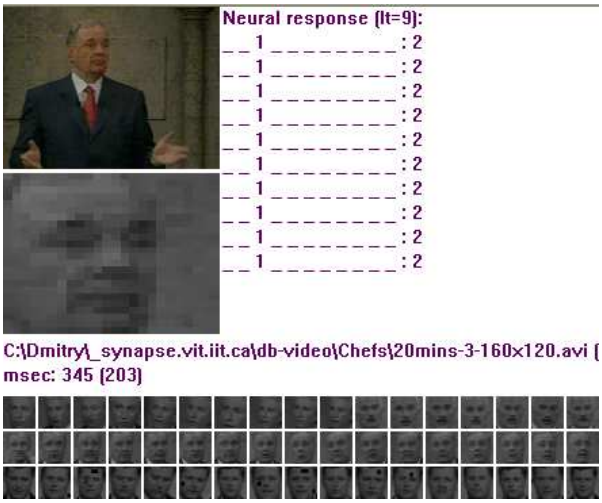

Fig. 1. Recognizing faces in a TV show using the neuro-biological model. For each video frame, the neurons corresponding to the person's nametag fire.

After a detected face is converted to the nominal resolution (which is of 12 pixels between the eyes), facial image is analyzed to produce a set of 24x24*3 binary feature attributes which are used to make a receptor stimulus R.

The effector stimulus, which decodes the face nametag (E) is obtained by fixing the neuron corresponding to the person's ID excited (+1), while keeping other neurons unexcited (-1), with the number of neurons equal to the total number of nametags. When the person's ID is unknown, as in recognition stage, all effector neurons are set unexcited (-1).

The entire process from capturing image to memorizing a face (in learning) or retrieving the face ID (in recognition) takes about 150 msecs on Pentium 4 processor, of which memorization/recognition takes about half. This makes it possible to memorize and recognize faces from video on fly in real time. What is most important however is that the presented neuro-associative model allows one to approach the video-based recognition problem within a new, non Von-Neumann, biologically inspired framework, which is very much needed for this problem [6], since it takes advantage of video over still imagery and can deal with the continuous flow of video data. Rather than storing a continuous amount of individual video frames, the approach uses the incoming flow of visual data to continuously tune the synaptic connections of a multi-connected neural network, similar to the process of associative memorization in the visual cortex. Then, when a new video stimulus is presented to the retina, the neural network converges to a state which is best described by the past seen/learnt experience.

V. Conclusion

The Face Recognition in Video program which incorporates the projective learning described in this paper is freely available for testing from our technical web site: http://synapse.vit.iit.nrc.ca/memory. The website also provides a video-based facial database and video-clips recorded from TV shows which were used to test the program. While being perfectly suited for humans in terms of their ability to recognize the individuals there, these TV clips pose a great challenge for conventional face recognition systems. Faces there are of very low resolution - 1/16th of 320 by 240 screen is a common scenario, showing a lot of variation in orientation and expression.

Figure 1 shows a frame from a TV show, in which our program can automatically on the fly identify the persons. There are four of them there. To memorize thm15-sec video clips of each of them were shown to the program prior to recognition stage, selected frames of which (shown in Figure) were used as the training stimili memorized using the projection learning. In recognition stage, the program was tested on the entire prerecorded 20 mins from the show and achieved the recognition rate of 95%. This is a very good result for an automatic face recognition system, taking into account the unconstrained environment within which it was performed.

It is clear also that the same neuro-biological model based on the projective learning can be applied to other recognition and associative tasks. The only thing to watch for is not too saturate the model (when some weights significantly dominate the others as a result of the limited size of the system), which can be avoided by either using the iterative approximation of the rule (Eq. 20) or by using the desaturation mechanism described in [7], [8].

References

[1] A. Albert. *Regression and Moore-Penrose pseudoinverse,* Academic New-York, 1972.

[2] S. Amari. Neural theory of association and concept formation, *Biological Cybernetics*, vol 26, pp. 175-185, 1977.

[3] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci. USA*, vol 79, pp. 2554-2558, 1982.

[4] D.O. Gorodnichy, The Optimal Value of Self-connection or How to Attain the Best Performance with Limited Size Memory. Proc. of IJCNN'99 ("Best Presentation" award), Washington, July 12-17, 1999

[5] D.O. Gorodnichy, "Video-based framework for recognizing people in video," in *Second Workshop on Face Processing in Video (FPiV'05). In Proc. of CVR'05, Victoria, Canada, May 9-12, 2005*, 2004.

[6] D.O. Gorodnichy, "Recognizing faces in video requires approaches different from those developed for face recognition in photographs," in *Proceedings of NATO IST - 044 Workshop on*.

[7] D.O. Gorodnichy and A.M. Reznik, "Increasing attraction of pseudo-inverse autoassociative networks," *Neural Processing Letters*, vol. 5, no. 2, pp. 123–127, 1997.

[8] D.O. Gorodnichy, "Investigation and design of high performance neural networks," *PhD dissertation, Glushkov Cybernetics Center of Ac.Sc. of Ukraine in Kiev (http://www.cs.ualberta.ca/∼ai/alumni/dmitri/PINN)*, 1997.

[9] L. Personnaz, I. Guyon and G. Dreyfus. Collective computational properties of neural networks: New learning mechanisms, *Phys. Rev. A* vol 34, pp. 4217-4228, 1986.