CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 114 January 12, 2021

From Associative Memories to Deep Networks

Tomaso Poggio

Abstract

About fifty years ago, holography was proposed as a model of associative memory. Associative memories with similar properties were soon after implemented as simple networks of threshold neurons by Willshaw and Longuet-Higgins. In these pages I will show that today's deep nets are an incremental improvement of the original associative networks. Thinking about deep learning in terms of associative networks provides a more realistic and sober perspective on the promises of deep learning and on its role in eventually understanding human intelligence. As a bonus, this discussion also uncovers connections with several interesting topics in applied math: random features, random projections, neural ensembles, randomized kernels, memory and generalization, vector quantization and hierarchical vector quantization, random vectors and orthogonal basis, NTK and radial kernels.

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

Deep Networks as Associative Nets

Tomaso Poggio

Abstract

About fifty years ago, holography was proposed as a model of associative memory. Associative memories with similar properties were soon after implemented as simple networks of threshold neurons by Willshaw and Longuet-Higgins. In these pages I will show that today's deep nets are an incremental improvement of the original associative networks. Thinking about deep learning in terms of associative networks provides a more realistic and sober perspective on the promises of deep learning and on its role in eventually understanding human intelligence. As a bonus, this discussion also uncovers connections with several interesting topics in applied math: random features, random projections, neural ensembles, randomized kernels, memory and generalization, vector quantization and hierarchical vector quantization, random vectors and orthogonal basis, NTK and radial kernels.

1 Introduction

The plan of this brief note is to show that today's deep nets can be regarded as refurbishing the old networks proposed fifty year ago as associative memories, with properties similar to holography. After this first part, I will discuss a number of intriguing relations between random features, random projections, neural ensembles, randomized kernels, memory and generalization, vector quantization and hierarchical vector quantization, random vectors and orthogonal basis, NTK and radial kernels. The third and final part of this note discusses briefly the role that associative, recurrent and deep, networks may play in our attempts to understand human intelligence.

2 From associative nets to deep nets

2.1 Willshaw Nets

Holograms store information in the form of an optical interference pattern recorded in a photosensitive optical material. Light from a single laser beam illuminates a noise-like reference image (originally produced from ground glass) as well as the pattern to be stored, producing an interference pattern stored in the hologram. Many thousand such pairs of associations can be recorded on a single hologram. Each stored data can then be read-out from the hologram by using as input its associated reference pattern.

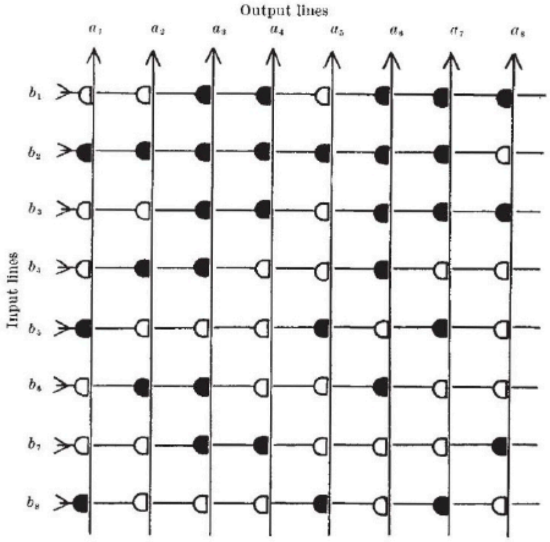

Fig. 4. An associative net.

Figure 1: *A original figure from Willshaw et al., [1] showing an associative memory network. The matrix of connections correspond to the matrix W of weights in a shallow network, see text.*

The basic associative memory $A_{X,Y}$ can be modeled as a one layer "shallow" network [1] storing the correlation matrix between input and output. Figure 1 shows the training phase of the network. In the read-out phase, the output y can be retrieved by inputting the associated x to the network, that is by computing $A_{X,Y} \circ x$ (Willshaw computed $R \circ A_{X,Y} \circ x$ where R represents a set of thresholds on the outputs to improve accuracy retrieval in a otherwise linear network).

The basic idea is as follows. Suppose that we want to associate each pattern $y_n, \quad n = 1, \cdots, N$ to a noise-like key vector x_n, where $x, y \in \mathbf{R}^D$ and $N < D$. The noise-like assumption on the x_n is equivalent to assuming that $XX^T \approx I$, where X is the matrix of all the inputs (x_n are the columns of X). The optimal least-square solution of the equation $AX = Y$ is $A = YX^T(XX^T)^{-1} = YX^T$. Thus, if I want to retrieve y_i, I input the key x_i to the network and get $Ax_i \approx y_{i,j}\delta_{i,j} = y_i$.

Of course, in dealing with binary vectors, this linear associative network can be improved by using thresholds to clean up the output as Willshaw did.

In any case, this is still a one layer network, quite different from modern multilayer networks. It turns out that Willshaw experimented "de facto" with multilayer networks when he found that a recurrent version of his one layer network performed quite well. As he reported "...it was found by computer simulation...that the initial response to a given cue could be improved by feeding the output back into the associative net and continuing until the sequence of outputs so generated converged onto a single pattern...". Furthermore, "The same "cleaning-up behavior" was seen when patterns were stored in sequence. Pattern A was associated with B. B with C. C with D. and so on, the last pattern being stored with A. When a fragment of A was used as a cue and then the output used as the next input, after a few passes the sequence of retrieved patterns converged onto the stored sequence, even when the initial cue was a very poor representation of one of the stored patterns. Simulation experiments were performed to see what cycle of outputs would result from any arbitrarily selected cue. (Because each input determines the next output and there is only a finite number of possible outputs, the sequence of outputs must eventually lead into a cycle.).."

It is quite easy to see what is going on. Using the algorithm above, one can associate x_1 to x_2 and x_2 to x_3 and so on, performing the kind of cyclic retrieval described in the second paragraph above. Of course a recurrent network is just a multilayer network with shared weights across different layers[2]. Thus fifty years ago we had already the idea and the implementation of single layer as well as recurrent associative networks !

2.2 Shallow, deep and recurrent networks

Is there something more that we can say about associative nets? The following is a simple additional observation about depth.

As we already mentioned, the optimal least square solution of $AX = Y$ is $A = YX^\dagger = YX^T(XX^T)^{-1}$. This suggests (among other possibilities) a 2-layer network with $W_1 = (XX^T)^{-1}$ and a read-out layer

$$W_2 = YX^T. \tag{1}$$

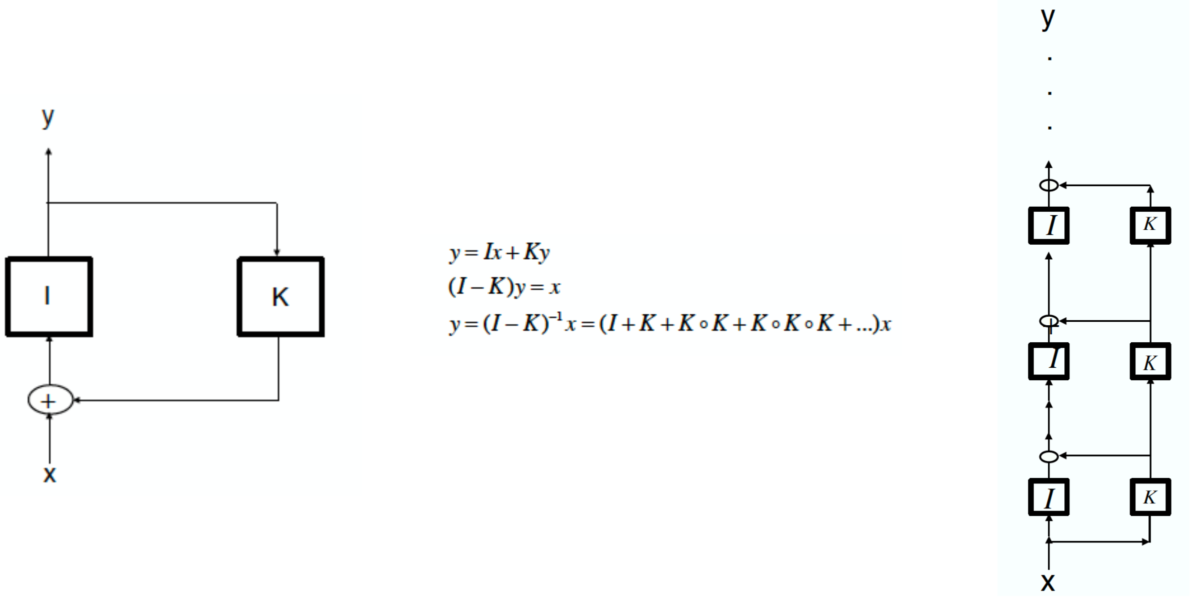
y
I
K
+
x
$y = Ix + Ky$
$(I - K)y = x$
$y = (I - K)^{-1}x = (I + K + K \circ K + K \circ K \circ K + \ldots)x$
y
I
K
I
K
I
K
x

Figure 2: *Setting $K = I - XX^T$ allows the recurrent network as well as its unrolled deep network counterpart to compute $(XX^T)^{-1}$.*

Interestingly, the computation of $W_1 = (XX^T)^{-1}$ can be performed by a recurrent network. Assume that the weight matrix of the recurrent network is set to

$$W_i = (I - XX^T), \quad \forall i = 1, \cdots, L - 1., \tag{2}$$

with the last read-out layer set to be $W_L = YX^T$. Since division of operators can be approximated by its power expansion, that is $\frac{I}{I-K} = (I + K + K^2 + \cdots)$, a recurrent network as shown in Figure 2 computes $(I - K)^{-1}$. If $K = I - XX^T$, the recurrent network computes $(XX^T)^{-1}$. Alternatively, a recurrent network can be replaced by a deep residual network (ResNet) of $L - 1$ layers with the same K (see Figure and [3, 2]. Convergence requires the condition $||XX^T - I|| < 1$, which is usually satisfied if the weight matrices are normalized (for instance by batch normalization). Estimates about retrieval errors in such associative memories and ways to reduce them by using thresholds are given in [1, 4].

Thus training a recurrent network under the square loss on a training set (X, Y) by unrolling it in L layers and imposing shared weights for the first $L - 1$ layers should converge to the quasi-optimal solution suggested by Equations 1 and 2.

So far I have described linear networks. The RELU nonlinearity after unit summation can be added as follows. Let us assume a deep network written as

$$f(x) = (V_L \sigma(V_{L-1} \cdots \sigma(V_1 x))) \tag{3}$$

where $\sigma(x) = \sigma'(x)x$, which captures the homogeneity property of the RELU activation. The equation can be rewritten for each training example as

$$f(x_j) = V_L D_{L-1}(x_j) V_{L-1} \cdots \cdots V_{k+1} D_k(x_j) V_k \cdots D_1(x_j) V_1 x_j \tag{4}$$

where $D_k(x_j)$ is a diagonal matrix with 0 and 1 entries depending on whether the corresponding RELU is active or not for the specific input x_j, that is $D_{k-1}(x_j) = diag[\sigma'(N_k(x_j)]$ with $N_k(x_j)$ the input to layer k.

The presence of the $D(x)$ matrices makes the networks much more powerful in terms of approximating any continous functions instead of just linear functions. It also affectss the linear analysis described earlier.

Remarks

- The convergence of a recurrent network for $L \to \infty$ – where L is the number of iterations – is guaranteed by Brower's fixed point theorem if the operator $Tz = Wz$ is non-expansive, that is if $||Tx - Ty|| \leq ||x - y||$. The fact that the operator corresponding to the transformation of each layer of the network is non-expanding follows from the fact that $||Wz|| \leq ||W||||z||$, assuming that $||W|| = 1$ because of batch normalization(BN) (see [5] for the importance of BN). Notice that this holds for linear networks but also for networks with RELU nonlinearities. If the inputs x satisfy $||x|| \leq 1$ the set of fixed points of T contains a unique minimum norm element (see [6])
- Deep networks with $L - 1$ layers of identical input and output dimensionality and shared weights across layers are equivalent to a one-layer recurrent network run for $L - 1$ iterations. Empirically it seems[2] that non-shared weights give only a small advantage despite the much larger number of parameters with respect to equivalent shared-weights networks. From this perspective, multiple layers may be required only to exploit the blessing of compositionality[7, 8]. In other words, depth's main purpose may be to allow pooling at certain stages (even just by subsampling).
- Consider instead of $W_{i,j} = (XX^T)_{i,j}$ the choice

  $$W_{i,j} = K(x_i, x_j) = \sum_{\ell}^{\infty} \lambda_\ell \phi_\ell(x_i) \Phi_\ell(x_j) = \Phi(x_i)\Phi^T(x_j) \tag{5}$$

  where the (infinite) column vector $\Phi(x) = \lambda_i^{\frac{1}{2}} \phi_\ell(x)$ and λ_ℓ are the eigenvalues of the integral operator associated with K. A shift-invariant kernel such as the Gaussian kernel has $\phi_\ell(x)$ which are orthonormal Fourier eigenfunctions. It can be approximated by random Fourier features $e^{-i\omega x}$ with ω drawn from a Gaussian distribution [9].
- The "holographic" scheme of using a "noiselike" key vector associated with a signal is almost exactly the algorithm used in the spread spectrum CDMA techniques used to encode and decode cell phones communication.

3 Discussion

We have described how deep and recurrent networks can be regarded as stacked associative one-layer networks of the Willshaw type. This perpective is interesting for two main reasons. First, it connects deep networks with several classical ideas such as random quasi-orthogonal basis, kernels, randomized RKHS features, the key role of normalization and compositionality. Second, if deep networks are "just" associative memories, what is their role in explaining intelligence? In other words: is associative memory a key part of human intelligence?

3.1 Connections between deep learning and signal processing

- The old associative networks assumed noise-like inputs that are approximately orthogonal (like in the original concept of holography implementing an associative memory), that is $x_i^T x_j = \delta_{i,j}$. A recent analysis [5] of deep network trained under the square loss identifies a bias towards orthogonality induced by normalization techniques such as batch normalization. Quasi-orthogonality makes it easy to invert a deep network as it is required in an autoencoder. Notions related to random projections and the Johnson-Lindestrauss lemma may also be relevant.

- I did not say much about convolutional networks. The architecture of convolutional networks reflects a specific type of Directed Acyclic Graph (DAG). It turns out that all functions of several variables can be decomposed according to one or more DAGs as compositional functions, that is functions of functions[8]. Often such decompositions satisfy a hierarchical locality condition: even if the dimension of the overall function is arbitrarily high, the constituent functions are of small, bounded dimensionality. For these functions and these decompositions, approximation theory proves[8] that deep networks reflecting the underlying compositional DAG can avoid the curse of dimensionality, whereas shallow networks cannot. Convolutional networks are an example of this (locality of the kernel rather than weight sharing is the key property in avoiding exponential complexity). Not accidentally, convolutional networks represent one of the main success stories of deep learning. Thus the main reason for deep networks as opposed to shallow, recurrent networks may in fact be to escape the curse of dimensionality by exploiting compositionality: this requires what we called earlier "pooling", that is stages at which the outputs of constituents functions undergoes aggregation , as in Figure 3.

- Compositional architectures can be regarded as reflecting iterated functional relations of the kind "compose parts" as in $f(x_1, x_2, x_3) = f_1(f_2(x_2, x_3), f_3(x_3))$, where f_1 reflects the composition of f_2 and f_3 and f_2 composes x_1 and x_2. A deep associative network of this type is then closely related to what is called "hierachical vector quantization (VQ)"[10]. The similarity is especially strong if we assume weight matrices that are derived from RBF kernels. This corresponds to memorizing, at the lowest level, the association of basic features and then the association of their associations (think of hierarchical JPEG

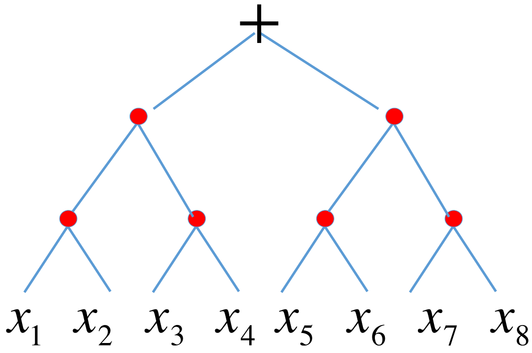

Figure 3: The figure shows the graph of a function of eight variables with constituent functions of dimension two.

encoding)[1].

- The claim that deep networks are quite similar to "linear" RBF networks is supported by recent results[11] on the Neural Tangent Kernel (NTK). It turns out that under certain training conditions (e.g. starting with "largish" norms for the matrices weights) a deep network converges to a set of weight matrices that corresponds to a standard kernel machine with the NTK kernel. Furthermore, classification performance is quite good – though not the best possible – and the NTK itself is equivalent[12] to a classical RBF kernel, the Laplacian.

- An alternative to deep networks as models of the brain are neural assemblies. The idea received new life from some recent very interesting work [13]. The obvious question is about connections between neural assemblies and associative memories.

3.2 Is human intelligence "just" associative memory?

Thirty years ago I wrote a paper[14] proposing " that much information processing in the brain is performed through modules that are similar to enhanced look-up tables". I had in mind associative memories and implentations such as RBF networks (see[2] Equation 5): for instance for a Gaussian kernel, increasing σ changes the network from a look-up table kind of memory, that

[1] Starting from a small number of primitive features, there is a hierarchy of more complex features each one being an association of simple features. If the simple features are stored then only some of the more complex ones – only the ones which are used – need to be stored as associations. This is similar to a dictionary storing only some of the infite number of words that may be created from a finite alphabet of letters.

[2] RBF networks are usually thought as Gaussian unit computing $e^{-\frac{(x-x_i)^2}{\sigma^2}}$ where x_i is the "center" of unit i; in Equation 5 the network reflect the dual form of a RBF network in terms of the Fourier features of the Gaussian.

recognizes only the training data, to a “learning” system that generalizes beyond the training data.

Willshaw only looked at his network as a memory. It was better than a pure look-up table since it could work well with noisy or partial inputs but its function was to memorize and retrieve. The machine learning and neural network community has looked only at generalization beyond the training data. In fact, the boundary between associative networks – shallow or deep — and learning networks is very thin, since the underlying machinery is very much the same and the difference is just in parameter values.

This was the reason I wrote that the idea of intelligence grounded on associative memory “suggests some possibly interesting ideas about the evolution of intelligence...There is a duality between computation and memory. ... Given that the brain probably has a prodigeous amount of memory ... is it possible that part of intelligence may be built from a set of interpolating look-up look-up tables? One advantage of this point of view is to make perhaps easier to understand how intelligence may have evolved from simple associative reflexes...”.

Clearly human intelligence is not just associative memory. Because of the previous discussion this also means that intelligence is not just deep learning. It is possible, however, that the intelligence of a dog may be explainable in terms of associative memory modules or equivalently deep or shallow networks. It is also very likely that human intelligence evolved from associative memories and that associative networks are still an important part of how we think, from visual and speech recognition to Kahneman’s System One which is fast, intuitive, and emotional whereas System Two is slower, more deliberative, and more logical. The question then is: how did logic and language based thinking evolve from associative memories? What are the differences in the circuits underlying them with respect to associative networks? I regard this as the core question in our quest to understand human intelligence and replicate it in machines.

Acknowledgments I am grateful to Andrzej Banbuski for finding a bug in section 2.2 and to Arturo Deza and for useful comments. This material is based upon work supported by the Center for Minds, Brains and Machines (CBMM), funded by NSF STC award CCF-1231216, and part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

[1] D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222(5197):960–962, 1969.

[2] Q. Liao and T. Poggio. Bridging the gap between residual learning, recurrent neural networks and visual cortex. *Center for Brains, Minds and Machines (CBMM) Memo No. 47, also in arXiv*, 2016.

[3] T. Poggio and W. Reichardt. On the representation of multi-input systems: Computational properties of polynomial algorithms. *Biological Cybernetics, 37, 3, 167-186.*, 1980.

[4] G Palm. On associative memory. *Biological Cybernetics*, 36:19–31, 1980.

[5] T. Poggio and Q. Liao. Generalization in deep network classifiers trained with the square loss. *CBMM Memo No. 112*, 2019.

[6] Paulo Jorge S. G. Ferreira. The existence and uniqueness of the minimum norm solution to certain linear and nonlinear problems. *Signal Processing*, 55:137–139, 1996.

[7] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Theory I: Why and when can deep - but not shallow - networks avoid the curse of dimensionality. Technical report, CBMM Memo No. 058, MIT Center for Brains, Minds and Machines, 2016.

[8] H.N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, pages 829– 848, 2016.

[9] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *NIPS*, pages 1177–1184, 2007.

[10] T. Poggio, F. Anselmi, and L. Rosasco. I-theory on depth vs width: hierarchical function composition. *CBMM memo 041*, 2015.

[11] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On Exact Computation with an Infinitely Wide Neural Net. *arXiv e-prints*, page arXiv:1904.11955, April 2019.

[12] Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency Bias in Neural Networks for Input of Non-Uniform Density. *arXiv e-prints*, page arXiv:2003.04560, March 2020.

[13] Christos H. Papadimitriou, Santosh S. Vempala, Daniel Mitropolsky, Michael Collins, and Wolfgang Maass. Brain computation by assemblies of neurons. *Proceedings of the National Academy of Sciences*, 117(25):14464–14472, 2020.

[14] T. Poggio. A theory of how the brain might work. In *Cold Spring Harbor Symposia on Quantitative Biology*, pages 899–910. Cold Spring Harbor Laboratory Press, 1990.