Graph Transformer Networks for Image Recognition

Léon Bottou[†] and Yann Le Cun[‡]

[†] *NEC Labs of America*
4 Independence Way, Princeton NJ08540, USA.
leon@bottou.org
[‡] *The Courant Institute, New York University*
715 Broadway, New York NY10003, USA.
yann@cs.nyu.edu

This contribution takes the example of a check reading system to discuss the modeling and estimation issues associated with large scale pattern recognition systems.

1. Problem Decomposition and Model Composition

Consider a system that takes the image of a check and returns the check amount. This system locates the numerical amount, recognizes digits or other symbols, and parses the check amount. Accuracy should remain high despite countless variations in check layout, writing style or amount grammar.

From an *engineering perspective*, one must design components for locating the amount, segmenting characters, recognizing digits, and parsing the amount text. Yet it is very difficult to locate the amount without identifying that it is composed of characters that mostly resemble digits and form a meaningful check amount (not a date or a routing number). Purely sequential approaches do not work. Components must interact, form hypotheses and backtrack erroneous decisions. The orchestration is difficult to design and costly to maintain.

From a *statistical perspective*, one seeks to estimate and compare the posterior probabilities $P(Y|X)$ where variable X represents a check image and variable Y represents a check amount. Let us define a suitable parametric model $p_\theta(y|x)$, gather data pairs (x_i, y_i), and maximize the likelihood $\sum_i \log p_\theta(y_i|x_i)$. Such a direct approach leads to problems of unpractical sizes. It is therefore common to manually annotate some pairs (x_i, y_i) with detailled information such as isolated character images T, character codes C, or sequences S of character codes. One can then model $P(C|T)$ and $P(Y|S)$ and obtain components such as a character recognizer or an amount parser.

The statistical perspective suggests a principled way to orchestrate the interaction of these components: let the global model $p_\theta(y|x)$ be expressed as a *composition* of submodels such as $p_\theta(c|t)$ and $p_\theta(y|s)$. The submodels are first fit using the detailled data. The resulting parameters are used as a bias when fitting the global model $p_\theta(y|x)$ using the initial data pairs $(y_i|x_i)$. This bias can be viewed as a capacity control tool for structural risk minimization (Vapnik, 1982).

Model composition works nicely with *generative models* where one seeks to estimate the conditional input density $P(X|Y)$ instead of the posterior $P(Y|X)$. For instance, Hidden Markov Models (HMM) for speech recognition (Rabiner, 1989) use the decomposition

$$(1) \qquad p_\theta(x(1) \dots x(\ell)|y) = \sum_{s(1)\dots s(\ell)} p_\theta(s(1) \dots s(\ell)|y) \prod_{t=1\dots\ell} p_\theta(x(t)|s(t))\, p_\theta(s(t)|s(t-1))$$

where $x(1) \dots x(\ell)$ represents the sound signal, and where the sum is taken over all the sequences $s(1) \dots s(\ell)$ of states (e.g. phonemes) that can represent the target word y. Such decompositions are derived by applying the Bayes rule and making suitable conditional independence assumptions. This idea can be extended to write the much more complicated models that our check reading example demands.

Both theoretical arguments (Vapnik, 1982) and empirical evidence (LeCun et al., 1998a; Lafferty et al, 2001) indicate that higher performance can be achieved by *discriminant models* that directly estimate the posterior probability $P(Y|X)$. For instance, the parser of a discriminant check reader can recognize that a character sequence resembles a date and therefore cannot be the check amount. In contrast, generative models cannot use such negative reasonning: the parser can only describe the syntax of potential check amounts. This restriction must be compensated by a much more detailled model. In fact, generative models waste computing resources and data to indirectly estimate the prior input density $P(X)$ which is not useful for our recognition task.

Early discriminant Markov models were built by rewriting (1) as:

$$p_\theta(y|x_{(1)}\dots x_{(\ell)}) = \sum_{s(1)\dots s(\ell)} p_\theta(y|s_{(1)}\dots s_{(\ell)}) \prod_{t=1\dots\ell} p_\theta(s_{(t)}|x_{(t)}, s_{(t-1)}) \tag{2}$$

Unfortunately, such discriminant models have a severe flaw (Bottou, 1991) known as the *label-bias problem* (Lafferty, 2001). Consider again the check reader example: because posterior likelihoods are normalized, the submodel $p_\theta(c|t)$ cannot express that subimage t does not represent any recognizable character c. Yet this negative information is very useful for segmenting image fields into characters.

LeCun et al. (1998a) solves the label-bias problem by modeling measures $\tilde{p}(\cdot)$ instead of probabilities $p(\cdot)$. Measures obey the same axioms as probabilities except for the normalization condition. This change does not affect the estimation process because we still estimate the parameters by maximizing likelihoods computed by normalizing the measures:

$$\max_\theta \sum_i \log \frac{\tilde{p}_\theta(y_i|x_i)}{\sum_y \tilde{p}_\theta(y|x_i)} \tag{3}$$

The change only affects how complex models are built by composing submodels. Although we add or multiply submodel measures as we would add or multiply probabilities, we do not enforce normalization constraints when composing models. Normalization only occurs at the ultimate level when estimating parameters.

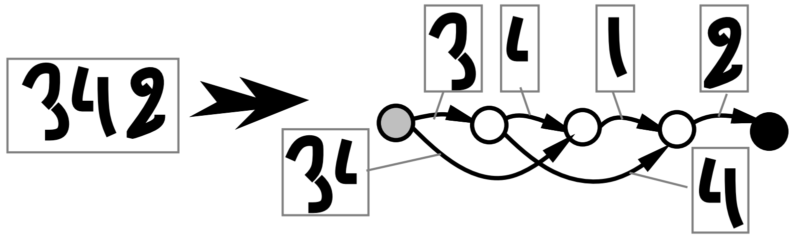
Lafferty et al. (2001) elegantly reach the same solution by casting discriminant Markov models as Markov Random Fields: the Hammersley-Clifford theorem relates their joint probability with Gibbs distributions; each submodel score corresponds to a term of the potential function; the final normalization factor is the partition function. This interpretation provides an additional justification for our framework.

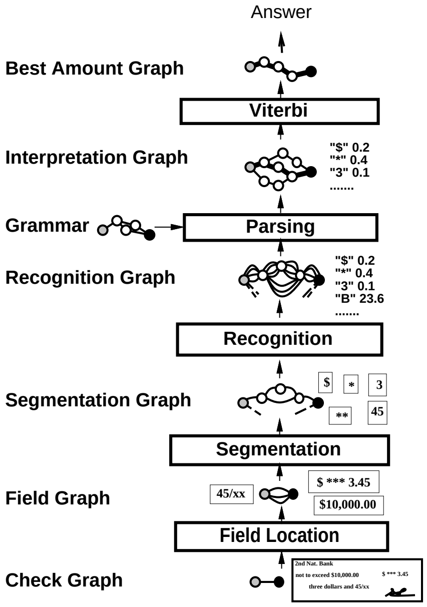
2. Graph transformers networks

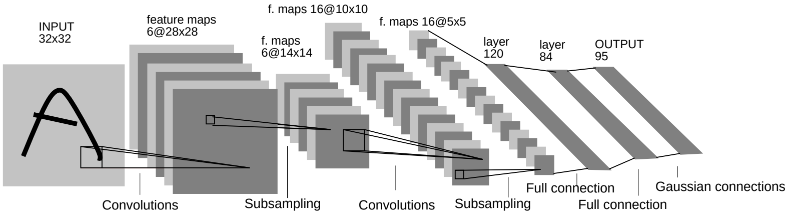
While Hidden Markov model deal with variable length sequences, our check reader must deal with more complicated variabilities (check layout, character segmentation, fractions, etc.). Writing the global discriminant model as a big formula like (2) is difficult and not very useful: it does not suggest tractable algorithms for computing the quantities of interest.

Graph Transformer Networks (Bottou et al., 1997) provide a more convenient way to express such models. Graph transformer networks use weighted acyclic directed graphs to track multiple hypothesis. Figure 1 shows how a graph can be used to represent segmentation hypotheses for an image representing a sequence of digits. Each hypotesis is represented by a path linking the start node to the end node. The score $\tilde{p}(\cdot)$ of an hypothesis is the product of the scores of each path component. The most likely hypothesis is easily found using the Viterbi algorithm. The sum of all hypotheses scores is easily computed using similar factorization techniques.

Figure 2 shows how a check reader model $\tilde{p}(y|x)$ is expressed as a sequence of graph transformations A first transformer produces a graph representing the various fields of the check. A second transformer refines this graph by describing segmentation hypotheses for each field. The next transformer emits character hypotheses C for each segment T. The next transformer composes this graph with a grammar transducer. This composition produces a







Figures 1 (top left), 2 (right), and 3 (bottom left)

graph representing the amount hypotheses independently of the detailled syntax of the character strings. The final step produces the most likely amount using a Viterbi algorithm.

This structure was suggested by Pereira et al. (1994) in the case of generative models. Instead of storing graphs in memory, they advocate representing graphs procedurally by defining accessor functions that can be called to navigate the graph structure. Furthermore, they show how graph transformations can be expressed as the composition of the input graph with a third graph named a *transducer*. The composition operation defines the accessor functions of the output graph on the basis of the accessor functions of the input graph and the accessor functions of the transducer graph. The composition function is generic. All the specifics of a particular graph transformer are neatly encapsulated in the accessor functions of the transducer graph.

The composition operation defines how output graph scores are computed by combining input graph scores and transducer graph scores. The parameters of a particular graph transformer are expressed through the transducer graph scores. For instance, scores on the recognition graph are computed by a convolutional neural network (figure 3) whose 60000 weights represent the majority of the tunable parameters in the global model. This network is invoked by the accessor functions of the transducer graph representing the recognition transformer.

All the critical graph transformer algorithms are implemented in terms of abstract graph accessor functions. This level of abstraction provides a clear separation between the generic model composition machinery, and the specific heuristics expressed through the transducer accessor functions. For instance, the field location and segmentation stages of the check reader (figure 2) were simply lifted from a previous system. Time-proven heuristic algorithms were seamlessly integrated into the global statistical framework.

3. Fitting the model

Model fitting starts with fitting each submodel with detailled data. For instance, the character recognition network was initially trained using 500000 labeled character images, randomly distorted using simple affine transformations. The network was then further trained on character images produced by the segmentation transformer and manually labelled. It was also trained to produces low scores on non-character images resulting from erroneous segmentation hypotheses. In contrast, the field locator and the segmenter parameters were simply copied from a previous hand-tuned check reader. The trained models were then inserted into the check reading system. Because only a few thousand check images were available, only the network last layer and the grammar score scaling factors were optimized by maximizing the global model likelihood. This was just enough to make the various submodels work smoothly together.

The numerical optimzation of such large scale systems requires specific techniques whose full discussion exceeds the scope of this paper. Technical details can be found in (LeCun et al., 1998b) and a posteriori justifications in (Bottou et al. 2005).

On 646 business checks that were automatically categorized as machine printed the performance of this system was 82% correctly recognized checks, 1% errors, and 17% rejects. This can be compared to the performance of the previous system on the same test data: 68% correct, 1% errors, and 31% rejects. An independent test performed by systems integrators showed the superiority of this system over commercial check reading systems. This check reader was integrated in NCR's line of check reading systems. It has been fielded in several banks across the US since June 1996. It has been reading millions of checks per month since then.

4. Conclusion

Graph Transformer Networks provide a very expressive language for discriminant statistical models with very high structural complexity. Our check reader experience shows how they can harness time–proven heuristics into the single framework of a statistical model.

ACKNOWLEDGEMENTS

This work is the product of a long term collaboration with Yoshua Bengio, Patrick Haffner, Chris Burges and John S. Denker. The check reader discussed in section 2 was built by many researchers and engineers working at AT&T Laboratories, Lucent Bell Labs, and NCR.

REFERENCES

V. N. Vapnik (1982): *Estimation of Dependences Based on Empirical Data.* Springer-Verlag.

L. R. Rabiner (1989): "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, 77(2):257-286.

L. Bottou (1991): "Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole", Ph.D. thesis, Université de Paris XI, Orsay, France.

F. Pereira, M. Riley, R. Sproat (1994): "Weighted rational transductions and their application to human language processing", in *ARPA Natural Language Processing Workshop.*

L. Bottou, Y. LeCun, Y. Bengio (1997): "Global Training of Document Processing Systems using Graph Transformer Networks", *Proceedings of IEEE Computer Vision and Pattern Recognition 1997*, 489-493.

Y. LeCun, L. Bottou, Y. Benjio, and P. Haffner (1998a): "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, 86(11):2278-2324.

Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller (1998b): "Efficient Backprop", in *Neural Networks, Tricks of the Trade*, LNCS 1526, Springer-Verlag.

J. Lafferty, A. McCallum, F. Pereira (2001): "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *Proceedings of the 18th International Conference on Machine Learning*, Morgan Kaufman.

L. Bottou, Y. LeCun (2005): "On-line Learning for Very Large Datasets", *Applied Stochastic Models in Business and Industry*, special issue, to appear.

RÉSUMÉ

Les systèmes réels de reconnaissance d'images sont composés de modules successifs: prétraitement, segmentation, classification, interpretation, etc. L'ensemble de ces modules peut être vu comme un modèle statistique unique dont les paramètres doivent minimiser une fonction de coût unique. Cela pose en pratique des problemes considérables. Les variables d'entrée sont très simples (pixels), mais le modèle possede une structure trè riche. Le nombre de paramêtres peut être très élevé et la fonction de coût, non convexe. Cette contribution présente une'approche générale, les "Graphs Transformer Networks" et les solutions pratiques que nous avons retenues. Cette discussion est illustrée par une application significative de lecture de chèques.