Viterbi Algorithm and Beyond

Aiichiro Nakano

Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Quantitative & Computational Biology
University of Southern California

Email: anakano@usc.edu

Top 10 Algorithms in History

In putting together this issue of *Computing in Science & Engineering*, we knew three things: it would be difficult to list just 10 algorithms; it would be fun to assemble the authors and read their papers; and, whatever we came up with in the end, it would be controversial. We tried to assemble the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century. Following is our list (here, the list is in chronological order; however, the articles appear in no particular order):

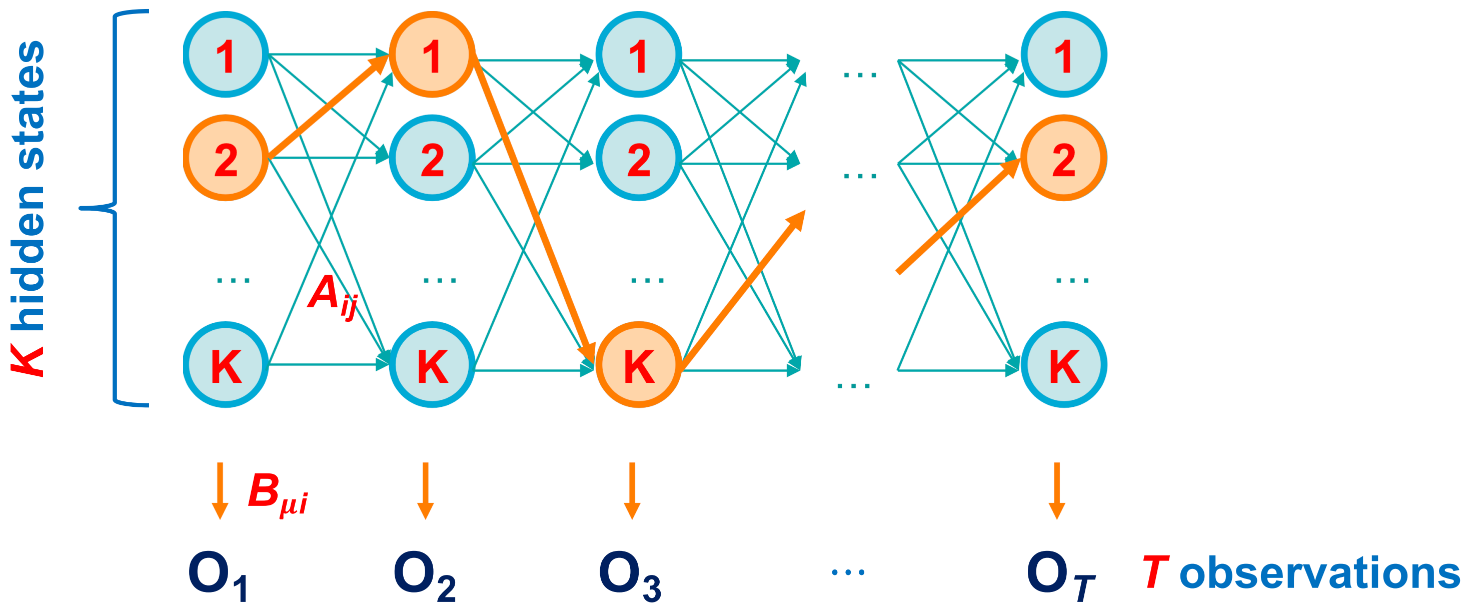
- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

PHYS 516
CSCI 596
CSCI 653

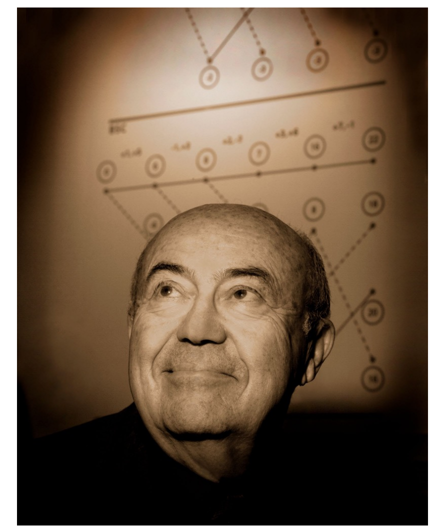
IEEE CiSE, Jan/Feb (2000)

Top 1 Algorithm in USC History?

- **Viterbi algorithm: Given a hidden Markov model (a set of K hidden states that transition between each other with transition-probability matrix A_{ij} & emit one of the observables with emission probability $B_{\mu i}$) & a sequence of T observations, determine the most likely state-transition path**



USC Viterbi School of Engineering



Andrew Viterbi

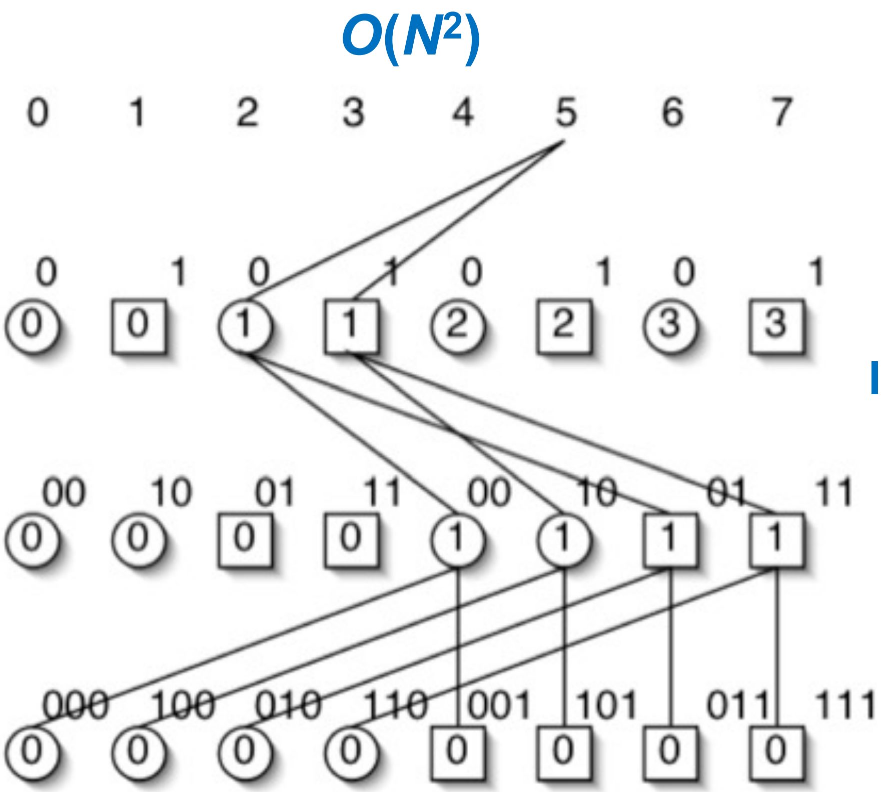
- **$O(K^T)$ way: Enumerate probability for all K^T paths**
- **$O(K^2T)$ Viterbi algorithm: Dynamic programming that recursively builds up necessary information for increasing observation time $t = 1,...,T$**

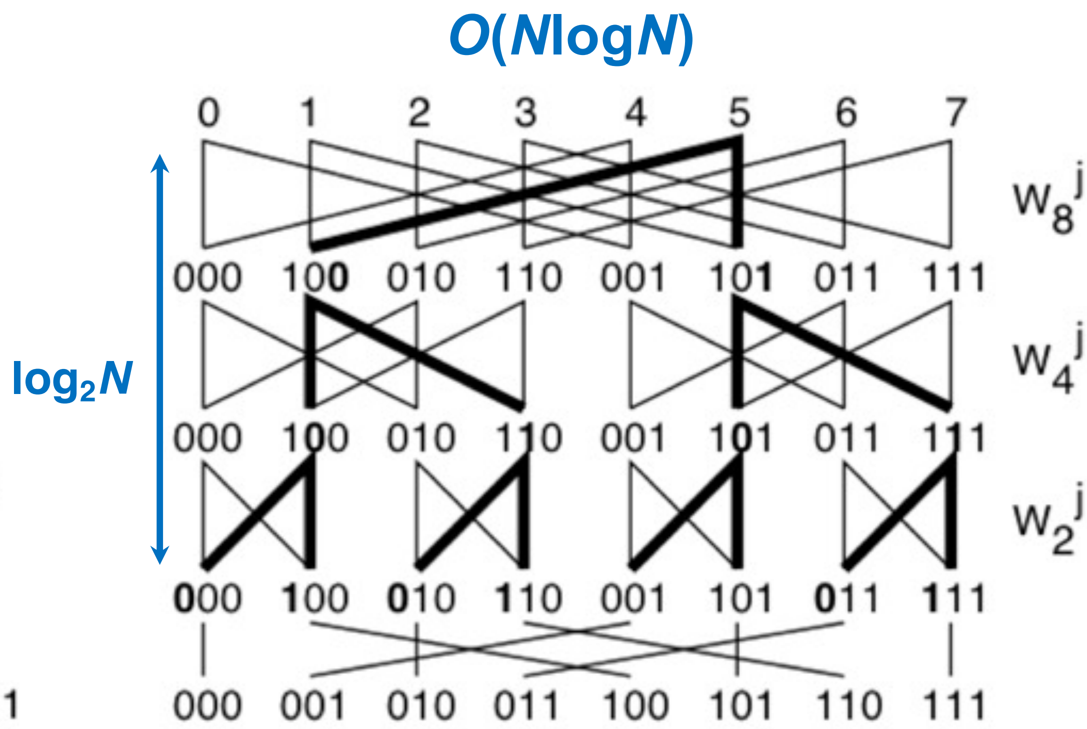
https://en.wikipedia.org/wiki/Viterbi_algorithm

***cf.* Waterman-Smith algorithm, RSAdleman cryptography**

cf. Fast Fourier Transform (FFT)

- **FFT algorithm reuses many overlapping path segments among N divide-&conquer trees to reduce the $O(N^2)$ brute-force evaluations to $O(N\log N)$**

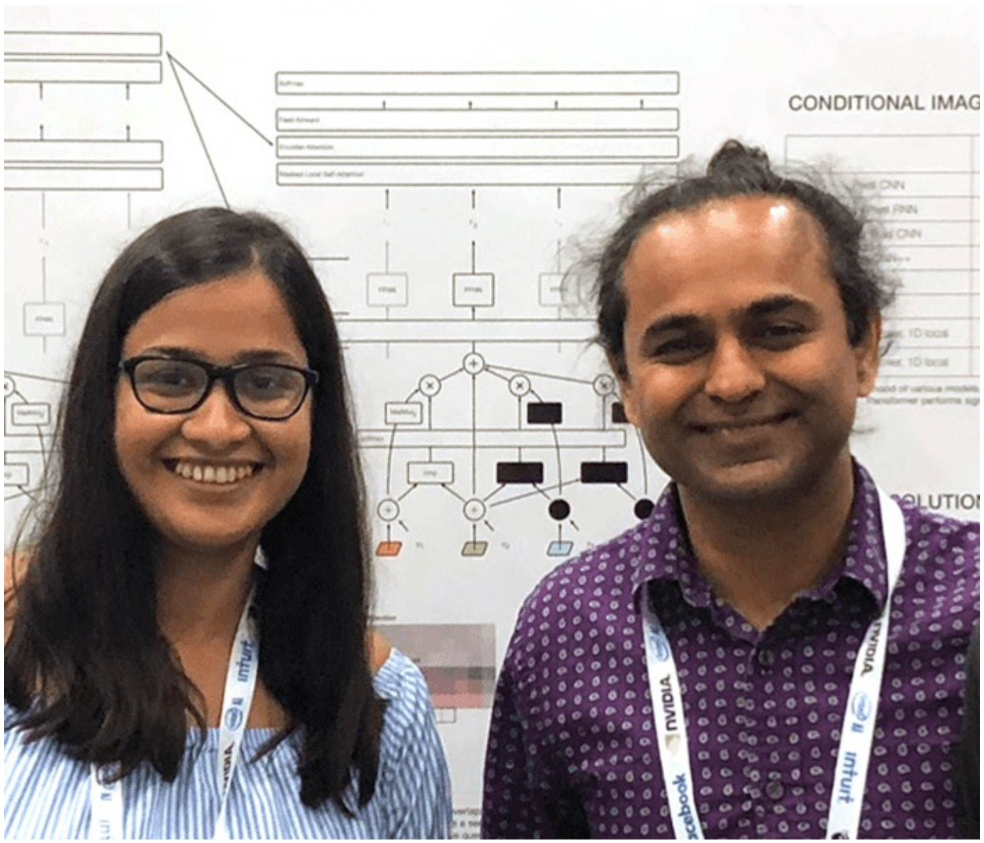




Analogy is mother of invention!

Top 1 USC Algorithm in New AI Era?

- **Attention mechanism by Ashish Vaswani, Niki Parmar (USC alumni) *et al*. (*NeurIPS'*17) has revolutionized AI**



Niki Parmar   Ashish Vaswani

Attention Is All You Need

Ashish Vaswani[*] Google Brain avaswani@google.com

Noam Shazeer[*] Google Brain noam@google.com

Niki Parmar[*] Google Research nikip@google.com

Jakob Uszkoreit[*] Google Research usz@google.com

Llion Jones[*] Google Research llion@google.com

Aidan N. Gomez[*] [†] University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser[*] Google Brain lukaszkaiser@google.com

Illia Polosukhin[*] [‡] illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

[*]Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†]Work performed while at Google Brain.

[‡]Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Have Fun Beyond Assignments

- **"If you decide you don't have to get A's, you can learn an enormous amount in college."**
  Isidor Isaac Rabi
- **In this course, we also learned:**
  1. **Singular value decomposition (SVD) & Cholesky decomposition for dimensionality reduction (low-rank approximation)**
  2. **Krylov subspace method: Lanczos algorithm for *O*(*N*) eigensolver**

     - Metropolis Algorithm for Monte Carlo
     - Simplex Method for Linear Programming
     - $\boxed{\text{Krylov Subspace Iteration Methods}}$
     - $\boxed{\text{The Decompositional Approach to Matrix Computations}}$
     - The Fortran Optimizing Compiler
     - QR Algorithm for Computing Eigenvalues
     - Quicksort Algorithm for Sorting
     - Fast Fourier Transform
     - Integer Relation Detection
     - Fast Multipole Method

And get A!

- **Extension: randomized low-rank matrix decomposition?**
  Murray *et al*., *arXiv*:2302.11474 ('23)