
Machine Learning Blog post recommendations

by: Johannes Losert, Columbia '25. NIST (‘22), Google (‘23), Amazon (‘24) Intern.

With special help from:
Berkan Ottlik , Columbia 25’. ML Theory Research Assistant, ML/Algorithms TA.

This series of blog posts and papers walks through the most important ideas in machine learning. Note that the ordering/selection of this list assumes introductory graduate-level understanding (Link to the course I took) of general machine learning learning concepts. Here is a brilliant Nature review by the godfather of deep learning, Yann Lecun that should do the trick if you are an advanced reader.

---

1. Berkan's Transformer Explainer

This blog post goes into the technical details of transformers, which form the backbone of large language models such as ChatGPT. This blog post won't give you a deeper understanding, but it is a neat read if you are just getting started! Berkan provides a detailed graphic that breaks down the architecture that can act as a handy reference:

Berkan's Transformer Architecture Diagram

---

2. Anthropic's Mathematical Framework for Transformer Circuits

This paper dissects the anatomy of the transformer and provides the intuitions leading AI researchers have about what these models *actually* do. If at this point you are confused about the meaning/selection of the QK and V matrices, this paper should resolve most of your questions. The paper explains that QK and OV operations of can be understood as two low rank approximations to very intuitive linear maps. The QK map tells us how much each input token wants to attend to each ‘key’ token, while the OV map tells us how attending to a particular ‘key’ token affects the output logits. Understanding everything up to "Path Expansion of Attention Scores QK Circuit" is important to understand why transformers work. Beyond that point the observations the paper makes tend to reach a bit (in that they probably do not extend to models with non-linearities). Also shoutout to Anthropic for the awesome appendix.

As you are digesting this paper, it is helpful to watch Neel Nanda's explainer on YouTube. He co-authored this paper, but he no longer works for Anthropic. The video is long and quiet, but he offers some great insights:

Neel Nanda's Youtube Explainer

3. Anthropic's Toy Models of Superposition

Great, this transformer architecture can efficiently model language, but why does it work at all? Welcome to mechanistic interpretability - a nascent field that seeks to explain how machine learning algorithms ‘really’ work! The above is a canonical paper from Anthropic that makes deep connections between transformers and the field of compressed sensing. There are a ton of interesting theories presented in this work, but for me the main takeaway is that transformers *most likely* represent ‘features’ as linear directions within the residual stream, and that they *most likely* utilize superposition to read and write information between exponentially more feature vectors than there are residual stream dimensions. For more mechanistic interpretability, follow Anthropic’s Transformer Circuits thread!

Watch Neel Explain this Paper (Note he was not involved in writing it)

4. Long Range Dependencies Without Transformers - S4

Wow, finally something that is not a transformer! You might be questioning how this ties into the previous line of research, and you'd be right to do so. For a while everyone was focusing on how we could speed up the inherently quadratic nature of transformer training by making self-attention as sparse as possible. Classic papers like BigBird come to mind. Meanwhile a graduate student at Stanford (Albert Gu) would soon revolutionize sequence modeling with structured state spaces (his dissertation).

This digestible explainer written by Stanford researchers (not Albert, he is now a professor at Carnegie Mellon) derives how structured state space models work. Structured State Space Models are great for modeling sequences with continuous nature. The Toy Models paper presented before posits that a transformer learns to represent exponentially more linear features than it has model dimensions (a.k.a. the dimension of the residual stream) by leveraging superposition and composition (For more on this I recommend: Distributed Representations: Composition & Superposition). State Space Models operate on a new paradigm - they treat the *discrete* input token sequences as online *continuous* functions and try by using a discretization rule. Simply training the relevant matrices (A, B, C) with random initialization performs badly - they achieve only 60% accuracy on MNIST classification benchmark! However, if you view the original input as a linear combination of orthogonal polynomials, such as those of the Chebyshev or Legendre type, online approximation using a simple discretization rule is much easier. In fact, using Legendre as your basis improves accuracy to 98% on MNIST.

5. Mamba - Selective S4 for Discrete Sequences

Mamba combines the strengths of the transformer and state space machine architectures. While the original S4 is suited for data with inherently continuous nature like audio, it suffers when confronted with discrete tasks, like modeling genomic sequences. S4 also has no way of filtering out information it deems less important. Mamba solves this issue with a hardware aware selection mechanism that takes advantage of the GPU memory hierarchy. The selection mechanism involves learning a linear projection of your input data that scales \Delta based on how 'important' the input token is for making predictions. A large \Delta, approaching infinity indicates a reset to the state, whereas a small \Delta, approaching zero persists state and forgets the current token. (Read "Interpretation of Selection Mechanisms" for more info).

A naive implementation of the discretization rule with varying \Delta would be slow because you have to find the inverse of a matrix. The paper solves this by placing objects that change frequently in SRAM (where computations are made). Objects that change frequently include the residual stream, which is constantly being read from/added to/subtracted from, and the transition matrix A, which determines what is added to the residual stream and must be newly inverted for each token you encounter (reminder: this is a cubic operation). More static elements of the computation are kept in GPU High Bandwidth Memory, which has more memory capacity. Table 3 of this paper shows that Mamba blows past Transformer architectures on most tasks. Most notably, Mamba is able to perform the induction heads task on sequences that are tens of thousands of times longer than the sequences the model was trained on. This is where hype is at the moment.

I hope this was useful! If you want to talk about this stuff with me, you can reach me at johannes.losert@columbia.edu

---

Berkan's Footnote

The state of AI reports always have the headliner empirical papers: https://www.stateof.ai/

Here are some key papers and references before 2020:

Overall references:
- This book seems like a decent overview: https://d2l.ai/chapter_introduction/index.html
- This coursera specialization is a decent overview, just quite slow: https://www.coursera.org/specializations/deep-learning

- Some people like the deep learning book, others hate it: https://www.deeplearningbook.org/ (online for free)
- I also like the probabilistic machine learning books (there's book 1 and 2, both available online for free) by murphy, although they have a lot of not deep learning too.
- Here is also a github that has a nice compilation of papers: https://github.com/hurshd0/must-read-papers-for-ml

Important deep learning papers (star next to imo most important ones):
- Deep Learning (easy read)
- DropOut (dropout, weight decay (literally just L2 regularization), learning rate decay (making the learning rate smaller over the course of training) are a few of the most popular regularization techniques for deep learning)
- Adam Optimizer (you really don't need to know the math, just know that everyone uses Adam, maybe some people also use sophia)
- Vision
  - *AlexNet (integral in kicking off the deep learning revolution)
  - ResNets
  - GANs (don't worry too much about the details I'd say)
  - Diffusion Models (this is just a post from lilians blog)
- Language:
  - Overview
  - Attention (I like the original attention paper, I also have a blog post about it. Attention is simply the idea that you can use your data to do feature selection. Basically, you use the data to determine which part of the data to use. You can easily define a linear regression model with attention.)
  - Transformers (I don't love the original transformers paper, but it is the original transformers paper)