Abstract

How did an “old dog” signal processing professor approach learning and teaching the “new tricks” of generative AI? This column overviews my recent experience in preparing and delivering a new course called Computational Creativity, reflecting on the methods I adopted compared to a traditional equations-on-a-whiteboard course. The technical material is qualitatively different from traditional signal processing, and the types of students who took the class and their approach to learning are different too. I learned a lot from the experience, but also came away with bigger questions about the role of educators in the age of generative AI.

A Signal Processor Teaches Generative AI

Richard J. Radke*, *Senior Member, IEEE*

*Department of of Electrical, Computer, and Systems Engineering

Rensselaer Polytechnic Institute, Troy, NY 12180 USA

I. The Impetus

When I became a Ph.D. student in the Department of Electrical Engineering at Princeton University, I thought of myself as an applied signal processor, and acquired a solid background in DSP, image processing, and communication and information theory. Like many readers, I learned about neural networks in the pre-deep-learning era, in a class where I implemented Hidden Markov Models and backpropagation based on a dozen data samples using a Pentium PC. This class was very interesting, but wasn't on the critical path for my thesis research.

I joined Rensselaer Polytechnic Institute in 2001 as an Assistant Professor in the Department of Electrical, Computer, and Systems Engineering, and spent much of my early career teaching signal processing at various levels (e.g., our required Signals and Systems course and our elective Digital Signal Processing, Signal Processing Design, and Introduction to Image Processing courses). I always found carefully developing signal and image processing concepts in class to be satisfying, such as the elegant connection between the Fourier Series and the Fast Fourier Transform. While my Ph.D. students over the past 20 years have unavoidably turned into deep learning experts to do their research in computer vision, I have to admit that I resisted delving more deeply into the details and mechanics of modern neural networks, even as it became clear that there was no going back to the "old days". I never had any interest in trying to teach a course on deep learning; honestly, I was worried that my students would already know more about it than I did!

However, the release of OpenAI's DALL-E 2 in the spring of 2022 pushed me over the edge to finally try to learn what was under the hood of this astounding technology. My friends and I

Corresponding author: R.J. Radke (email: rjradke@ecse.rpi.edu).

spent hours throwing prompts at the beta release of the system and being amazed at the coherent, stylized images that popped out in a matter of seconds (e.g., Figure 1). While I was generally familiar with the advances that had been made in specific areas (e.g., algorithms that could create faces indistinguishable from photographs[1]), DALL-E 2's ability to render any imaginable combination of concepts seemed like science fiction — something that didn't seem like a natural extension of the simple backpropagation algorithm I learned about in grad school. I decided it was time to learn more lest I get totally left behind!

Fig. 1. Early DALL-E 2 renditions of "Cats in Devo hats".

As many of us know, the best way to really learn something is to teach it to other people, so I bit the bullet and requested to design and teach a new special topics class in Fall 2023 called

[1]e.g., https://thispersondoesnotexist.com

Computational Creativity to cover the exciting developments in generative modeling over the past decade. The obvious course title would have been "Generative AI", but I dislike how "AI" is used to mean anything and everything in media and press releases. As many have noted, the term "artificial intelligence" is an unfortunate byproduct of the 1950s, when a more accurate but less glamorous term for today's algorithms might be "applied statistics". My other motivation was to emphasize the creativity aspect, exploring how both art-school and non-establishment artists have used generative algorithms in their practice, and thinking about the role of algorithms in the authorship of creative artifacts. These include not only images, videos, and text, but also music, 3D objects, line drawings, and animated characters. I was inspired by similar courses popping up at other universities (e.g., those offered by Ali Jahanian at MIT[2] and Eryk Salvaggio at Bradley University[3]), although it didn't seem like anyone was teaching the material in a systematic way from the ground up, building on technical concepts introduced in earlier lectures. It was also important to me to do the whole course in my own voice, learning the material alongside the students, as opposed to having an overabundance of guest speakers and Powerpoint slideshows.

II. The Preparation

Even though my experience is that students rarely buy (or even refer to) textbooks anymore, it was important for me to have a somewhat self-contained reference, if only to organize my own thoughts. The by-now classical deep learning textbook by Goodfellow et al. [1] is heavy on theory and algorithms but (in my opinion) light on the real-world, creative applications I was interested in. Ultimately I settled on Foster's book on generative models [2], which was just revised to a second edition in summer 2023 and was very up-to-date. While it isn't a traditional "university" textbook, I found it to be extremely readable. I started my preparation by reading this book and another introduction to modern deep learning algorithms, and then printing out hundreds of recent research papers to supplement those basic ideas.

The process felt like being a graduate student again, with the same highs (this stuff is so cool!) and lows (there is so much being published and I don't really understand it on a gut level!). One of my favorite classes to teach is our undergraduate Engineering Probability course[4], the source material for which fundamentally hasn't changed in a couple hundred years. In contrast, major

[2]https://ali-design.github.io/deepcreativity

[3]https://www.cyberneticforests.com/ai-images

[4]https://www.youtube.com/playlist?list=PLuh62Q4Sv7BXkeKW4J_2WQBlYhKs_k-pj

new generative models are being released almost on a weekly basis (e.g., GPT4 and DALL-E 3 were both publicly released in the middle of my class). To an old-school researcher, the pace of developments in this area is dizzying. Seminal innovations like DreamBooth [3] (a tool for inserting your own concepts into image generation) and ControlNet [4] (a way of making generated images conform to an example image's shape or pose) were initially released on arxiv while under review, and thousands of people began to use them and show off the results months before the papers were officially accepted, much less published. Stuffily insisting that "a paper doesn't count until it's accepted" doesn't cut it in the generative modeling world!

Another new-to-me concept (but totally familiar to my undergrad and grad students) was the ease with which research code is now disseminated on Github via conda environments and docker containers that allow large codebases with complex multi-package dependencies to be installed on your own computer with just a few mouse clicks, and (most of the time) everything immediately works. This meant that I wouldn't have to spend hours getting up to speed on Python programming from scratch, but could instead install a codebase and press "run" on a Jupyter notebook to get millions-of-parameters models immediately running for class examples.

Determining prerequisites for the course was an important question. I wanted to attract undergraduates since a big focus of the course was having fun using cutting-edge tools to create images, video, text, and audio. In my previous experience teaching my course *Computer Vision for Visual Effects*, I knew that the enthusiasm and willingness to have fun among undergraduate teams made for a much better classroom dynamic than the typical graduate student team that may know more math but seems to have forgotten how to be playful. In the end, I settled on the prerequisite of "any machine learning course" which I hoped would subsume the basic math/probability/programming knowledge I expected, as well as "a creative mindset and enthusiasm for making fun images and videos". Ultimately the collection of students I had for my trial run of the course was incredibly knowledgable about programming tools and environments, enthusiastic about having fun with the homework assignments, and interested to learn the underlying mathematical and algorithmic principles.

III. The Class

It was important for me to settle on the level of abstraction/depth for my lectures early on. I knew I didn't want to get tangled up in deep learning arcana that I frankly find boring – the typical network structure diagrams and numerical details about layers and activation functions

that I don't think add very much to big-picture understanding of why an algorithm is interesting and worth discussing. It was most important to me to introduce the loss function for a given algorithm – i.e., what is being optimized and why? – than to dwell on the details of the network that are used to actually carry out the optimization. Many of these choices (e.g., why 14 layers and not 12? why a ReLU activation function and not a sigmoid? why normalize a vector one way instead of another?) ultimately seem to come down to "because everyone agreed it empirically worked better that way" rather than to an intuitive or convincing mathematical explanation. These "correct decisions" can also change quickly, as the community pivots to the new method that "just works better".

In terms of the actual technical material I decided to cover, it was important to me to first survey computational tools for making art that existed long before deep learning. This included discussing the algorithmic work of such pioneering artists as Vera Molnar, Lillian Schwartz, Harold Cohen, and Sol LeWitt. I also wanted to discuss the non-photorealistic rendering wave of the late 1990s/early 2000s with such classic papers as Hertzmann's painterly rendering work [5]. The natural starting point for the generative modeling material was variational autoencoders (VAEs), introduced in 2014 [6] and refined over the next several years. These were supplanted around 2018 by generative adversarial networks (GANs), exemplified by StyleGAN2 [7] and the like, and extended to create images guided by other images such as the pix2pix framework [8]. Many artists, both professional and self-trained, began to use GANs to create computer-mediated works using a combination of research code and their own datasets (e.g., Sofia Crespo[5], Helena Sarin[6], and Mario Klingemann[7]). GANs themselves were overtaken by diffusion models [9] around 2020, and these form the basis for most production generative models today (e.g., DALL-E, Stable Diffusion, and Midjourney).

Going beyond still image generation, it was essential to discuss the key concepts of attention and transformers [10], which underlie almost every "foundation" model, especially for time-series data like natural language. We discussed large language models like GPT (generalized pretrained transformer) and their connections to images via the key CLIP (Contrastive Language–Image Pre-training) algorithm [11]. In the final section of the class, I devoted lectures to generative

[5]https://sofiacrespo.com

[6]https://www.neuralbricolage.com

[7]https://quasimondo.com

models for music (which had at least a few connections back to traditional signal processing!), 3D objects, vector graphics, and animation. 3D object generation in particular took us on a detour into neural rendering fields (NeRFs) [12], a separate technology that has transformed swaths of computer vision research in recent years.

I brought in two guest experts to speak authoritatively on connections between technology and society: Aaron Hertzmann of Adobe Research, who overviewed the long debate about whether computers can be said to create art, and Pamela Samuelsson of the UC Berkeley Law School, who described the ongoing legal challenges to generative AI companies.

My pedagogical approach has always been to combine class notes written by hand in real time, live examples of results from running code, and image and video clips to introduce results from key papers. This means I'm constantly switching between feeds from my classroom document camera, the remote desktop of a powerful computer running in my research lab a few floors away, my local desktop with PowerPoint and a web browser, and so on. On top of that, I wanted to record my lectures live so that I could put them on YouTube without needing any further editing. The tools that made all this possible were (1) the free Open Broadcasting Software (OBS) program[8], and (2) a gadget called the Loupedeck, a small panel of context-dependent reconfigurable buttons. These are both daily tools of Twitch streamers and similar live broadcasters, allowing me to set up and switch "scenes" (i.e., window configurations) with a single button push, instead of recording my messy desktop and manually hunting around for which window I wanted to show. While every lecture required me to plug 5 or 6 cables and dongles into my laptop, I'm very pleased with the final product, which is able to reach a much larger audience than the small group of students who take the course in person[9]. I also took the opportunity to learn about using Github pages to collect and manage my course material, which looks much cleaner than typical university learning management systems. A complete list of concepts, papers, websites, and software tools discussed in the course is available on a public Github page[10].

I assigned several mini-projects that required the students to work in teams to collect their own datasets and train generative models to mimic them. It was very important to me that students

[8]https://obsproject.com

[9]https://www.youtube.com/playlist?list=PLuh62Q4Sv7BX-uwS15xlbi7KUcgKUmWNL

[10]https://richradke.github.io/computationalcreativity

collect their own data instead of grabbing a convenient dataset off the web; for one thing, I think it reinforces the ethical issue of being responsible and intentional about creating and acquiring your own data, and knowing exactly what is in your dataset. Many of computer science's legal and ethical troubles stem from indiscriminately slurping up data that one didn't directly create and may have never actually laid eyes on. The students collected hundreds of images of leaves on campus plants, climbing holds at a local gym, Legos, and video game characters that they used throughout several homeworks.

In addition to my personal assessment, every assignment also involved a peer assessment in which about 10 other students commented on each submission; in my experience this raises the standard of work for the entire class (both in terms of the quality of the results and the written reports). The final project required students to leverage multimodal generative tools to create a final course video or software system, and the results were quite impressive[11]. These included a system that generated an entire voice-overed multi-scene video from a single short prompt, a drawing/guessing agent that could play both sides of a game of Pictionary, and an anime-styled tour of the engineering side of campus (Figure 2).

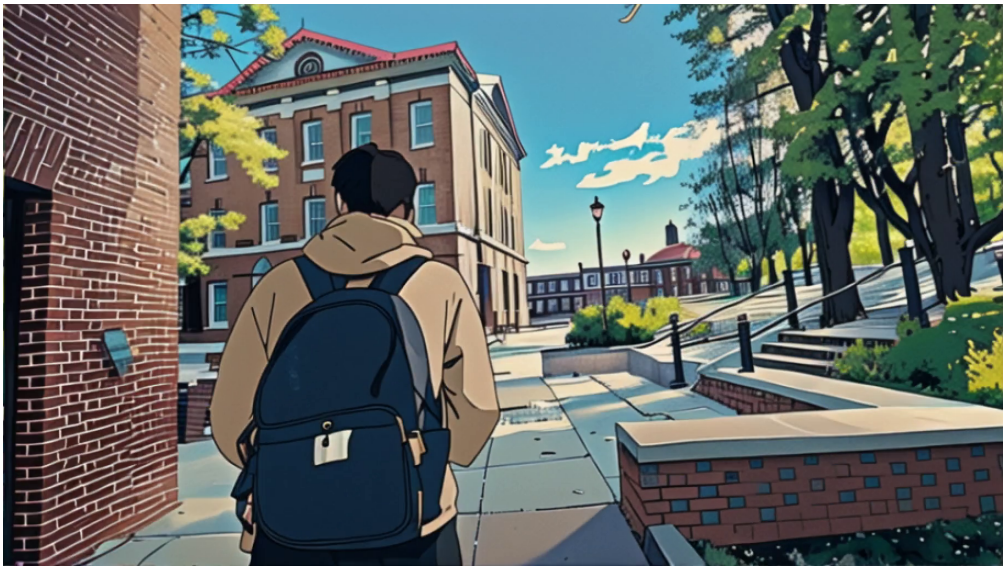

Fig. 2. An anime-style rendition of the Rensselaer Polytechnic Institute campus from a student project, created using generative video synthesis.

[11]https://www.youtube.com/watch?v=6zTdQ_xpT2Y

IV. Reflections

At the end of the day, what did I learn from the experience? For one thing, it made me question my assumptions about what today's electrical and computer engineering students really need to be taught as core material in 2024. While I love the mathematics and elegance of signal processing, I have to admit that there is very little of it in the generative AI material I taught. The closest concepts were the frequency-domain-inspired "positional encoding" used to represent scalar inputs inside the common formulation for attention in transformers and large language models [10], as well as the idea of image filtering implicit in convolutional neural networks. Basic concepts like the Fourier Transform, sampling and aliasing, and filter design were rarely required. Several Electrical and Computer Engineering programs have recently switched from requiring a traditional Signals and Systems course for all students to requiring a Data Science course that resembles an introductory machine learning course, and that may be the direction our field is going.

I was also surprised by the ease with which students can pick up complex software tools for generative modeling (and in fact are already well-versed in them). Many students came to my class knowing more about complex packages like Stable Diffusion[12] than I did (and had computers back in their dorm rooms more powerful than the ones in my lab)! However, this didn't translate to really understanding the underlying math and algorithms, so I think there's still an important role for professors who systematically develop the key concepts.

In one of the last lectures of the class, we had an open discussion about students' perceptions of the impact of generative AI on business, education, creativity, art, law, and ethics. There was a surprising diversity of opinion on whether using AI guidance in writing or coding was morally acceptable, whether AI systems can be said to think or create, how AI algorithms should be commercialized and released, and whether AI represented a threat to humanity in their lifetime. Certainly as educators and researchers we should be aware that many students have no qualms about having a ChatGPT window open on their desktop at all times to help them come up with ideas and answer questions! Conversely, I think that students feel uncertain about their career prospects in an age where entire subfields are suddenly threatened by generative models, and having a candid conversation about it was refreshing for everybody.

[12]e.g., https://github.com/AUTOMATIC1111/stable-diffusion-webui

I plan to teach my Computational Creativity class again in Spring 2025, by which time I expect many of my Fall 2023 lectures will need to be significantly updated. New paradigm-changing tools continue to be released abruptly, like OpenAI's Sora for coherent video generation. In the meantime, this semester I'm back to teaching Engineering Probability, but the process is newly unsettling. As an experiment, I asked my teaching assistant to feed each of my homeworks and exams into GPT4, and despite my tricky word problems, GPT is averaging about 90% on the assignments, showing its work and getting the right answers. If you haven't tried this in your own class yet, you might be surprised (and terrified!).

I've had to rethink what it means to be an effective educator in the face of this game-changing technology that somehow seems more threatening than the graphics calculator, WolframAlpha, Chegg, and Wikipedia. My best answer is that good professors can instill intuition about what the right answers *should* look like, whether outputs from automatic tools are sensible and trustworthy, and when and why certain approaches make sense. I'm less concerned about whether a student can do an integral or solve a linear system by hand than if they have set up the right problem in the first place and have a gut feeling or back-of-the-envelope estimate of what they expect the result to be. It will take a lot of work from all of us to rethink what the role of signal processing in a modern electrical engineering curriculum should look like, and how students' understanding can be fairly evaluated.

In the same vein, I'm finding that being a research advisor has become more challenging when the first impulse of a new graduate student is to solve a problem with a billion-parameter transformer model (that can be implemented in just a few lines of code) instead of proceeding from commonsense signal and image processing first principles that I find more intuitive. Trying to instill this intuition is difficult for a generation of students that grew up with powerful, high-performing "black box" algorithms. I'm very grateful to the IEEE Signal Processing Society for recognizing me with the 2023 Regional Distinguished Teacher Award so I can continue thinking about these issues.

V. Acknowledgements

Thanks to Aaron Hertzmann for comments on the manuscript, and to John Wen for supporting the development of my course. Figure 1 was generated by the beta release of DALL-E 2 based on a prompt from the author and Figure 2 was generated with a combination of open-source

generative modeling tools by Rensselaer Polytechnic Institute students Marklin Junning and Jason Zheng for a class project.

REFERENCES

[1] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] D. Foster, *Generative Deep Learning: Teaching Machines To Paint, Write, Compose, and Play*, 2nd ed. O'Reilly Media, Inc., 2023.

[3] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[4] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[5] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, 1998, pp. 453–460.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proceedings of the International Conference on Learning Representations*, 2014.

[7] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021.

[12] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," *Computer Graphics Forum*, 2022.