**Assignment Week 6**

Due: 3/4/2024

Shimon Greengart

Frank Rosenblatt's article from 1958 in the Psychological Review described one of the earliest neural networks. His goal was not to invent a problem-solving algorithm so much as it was to model the brain, and specifically how the brain processes sensory input. I'm not sure how successful he was at that goal, but his perceptron became the foundation of all neural networks in computer science.

Rosenblatt was operating under an assumption that all neurons in a brain were either on or off. If the stimulus didn't have the characteristic it tracked strongly enough, it would be off. Otherwise, it would send a signal to the neurons it was connected to. He had the idea for a layered network, with not just input and output layers, but also hidden layers between them that would allow for the recognition of subpatterns. Interestingly, unlike modern networks, his layers weren't unidirectional; neurons could send signals back to earlier layers, though not to the input layer.

He also had the idea of positive and negative reinforcement with a loss function (though he doesn't call it that), when, given a response based on success or failure of its categorization, the neurons could adjust their signaling and improve the network's accuracy. He even thought of clustering, where a network could discover categories on its own. The only thing he didn't think his perceptron could do was examine relationships between objects.

But while Rosenblatt designed a neural network, he had no idea how to build one. Over time, people added more ideas to the neural network, such as non-binary activations for neurons. But even as machine learning progressed, people didn't know how to train a neural network with hidden layers like what Rosenblatt described.

In 1986, David Rumelhart, Geoffrey Hinton, and Ronald Williams solved that problem by inventing backpropagation. They used calculus to find the first derivative of the loss function on every weight and bias. Each epoch, they would update each weight by a factor of its derivative. This allowed every weight to be updated relatively quickly, making the training of layered networks possible. While it also made models vulnerable to local minima, the authors didn't think that was likely.

As time went on, people began using neural networks for image processing, much like Rosenblatt's original perceptron. Convolutional neural networks were invented to better process them. But the networks used were relatively shallow and could only perform so well, and they took a long time to train.

In 2012, Alex Krizhevsky, Ilya Stutskever, and Geoffrey Hinton made a model much better than anything that came before. The ImageNet dataset contained 1.2 million images with 1,000 labels, and their model outperformed all previous ones. One reason was that their network was much deeper than those that came before. They used eight layers, five convolutional and three densely connected, and said that removing just one layer caused a significant increase in loss. This heralded the rise of deep learning, with neural networks with many more layers than what came before.

But the authors' main goal was to make as big a model as they could without overfitting. To do that, they used tricks to increase the amount of input data they had. But in order to make a model so big, another important factor was training time; the more they trained their model, the better it got. But they were already training their model for several days. So, to speed up training, they programmed two graphics cards to train their model while the CPU was modifying inputs. This allowed them to train their model more in a smaller amount of time.

This article heralded the current state of neural networks. Ridiculously large models take months to train while gobbling a small country's worth of electricity. Meanwhile, Nvidia has made graphics cards specifically designed to train neural networks, sending their stock into over a trillion dollars with it being too late for you to buy. This article, along with the other two, was instrumental in creating the current state of machine learning.