The Role of Gradients in Convolutional Neural Networks

The Impact of LeNet-5 and Gradient Descent on Modern Machine Learning

Maxwell Banks

May 9, 2024

1 Historical Event

This historical event analyzed by this paper is the publication of *Gradient-Based Learning Applied to Document Recognition* Lecun et al. (1998). This paper revolutionized the training of neural networks and is the antecedent to much of the modern field of machine learning. For a brief primer of the anatomy of a neural network, refer to Appendix A.

2 Summary

The breakout research paper *Gradient-Based Learning Applied to Document Recognition* Lecun et al. (1998) outlined LeNet-5, one of the first published instances of a Convolutional Neural Network (CNN). CNNs are a subtype of neural network that generally specialize in image recognition (this is not univerally true, but for brevity this paper will assume inputs are pixels of an image). Lecun et al. (1998) discusses using gradient-based learning techniques to train neural networks in recognizing letters, highlighting a variety of network architectures and explaining the effectiveness of gradient-based learning in the context of those architectures.

Lecun et al. (1998) had wide-ranging impacts on the field of machine learning as a whole, but perhaps the most notable impact was the popularization of CNNs trained using gradient-based learning. As the paper itself notes, "Gradient-Based Learning procedures have been used since the late 1950's, but they were mostly limited to linear systems." Lecun et al. (1998) The key innovation of *Gradient-Based Learning Applied to Document Recognition* was to apply gradient-based learning (specifically, gradient descent) as a training mechanism for CNNs by using gradient descent to determine the adjustment of node weights by minimizing the "loss function", a function representing how incorrect a neural network's guess was.

Gradient descent is a technique that dates back to Augustin-Louis Cauchy who outlined the technique in his pamphlet *Méthode générale pour la résolution des systèmes d'équations simultanées* Cauchy (1847). Cauchy described this technique as "...a general method which may be able to serve to resolve directly a system of simultaneous equations", and was mostly concerned with using it to determine the movement of a star with precision. This technique leverages calculus to minimize a system of equations by repeatedly calculating the gradient (interestingly, the paper itself does not appear to use the term gradient–it exclusively uses partial derivatives) at a point, then "stepping" in the opposite direction of the gradient to a "lower" point. This process is repeated until a minimum is reached.

3 Explanation of Enhancement

Lecun et al. (1998) was a formative paper in the field of machine learning, especially computer vision. Modern CNNs can trace their lineage directly back to LeNet-5, and other machine learning architectures are similarly dependent on gradient descent as a training model. Because of the performance improvements in training neural nets afforded by Lecun et al. (1998), image recognition is now cheap and fairly ubiquitous–everything from phones to kiosks is capable of analyzing images.

There are a multitude of positive advancements that stem from the progeny of LeNet-5, but perhaps the most promising field for image recognition is healthcare. Computer vision algorithms are being leveraged to help analyze medical scans and assist doctors in finding complications faster and earlier–what takes a doctor hours to scan can be done at a high rate of accuracy by a neural network in mere seconds. Rana and Bhushan (2022), a metastudy of fourty primary studies, notes that "various classical [Machine Learning] and [Deep Learning] techniques are extensively applied [in the healthcare domain] to deal with data uncertainty...". This on its own is impressive, but it further goes on to note that "...a series of experiments using MRI dataset has provided a comparative analysis of [Machine Learning] classifiers and [Deep Learning] models where CNN (97.6%) and RF (96.93%) have outperformed other algorithms." Rana and Bhushan (2022) Twenty years after the initial paper, CNNs (and the associated gradient descent training mechanism) are more relevant than ever.

CheXED is an excellent example of timesaving capabilites of CNNs in the medical field. Developed by researchers at Stanford, CheXED scans radiograph images to detect pneumonia and "...required less than a second to identify the findings on a single chest radiograph." Irvin et al. (2022) CheXED is not yet capable of independently diagnosing a patient, but it does provide powerful initial scan that can then direct doctors to potential problem areas faster. Irvin et al. (2022) concludes that "Integration of CheXED with clinical decision support systems like ePNa may help reduce time to diagnosis and improve pneumonia management in the emergency department."

Machine learning in healthcare is still in early stages, but stands to save lives by minimizing the turnaround time between inspection and diagnosis. At the forefront of this wave of innovation are CNNs, which would not have been possible without the training improvements pioneered by Lecun et al. (1998).

4 Calculus Steps

Gradient descent relies on an important principle: *the direction of a gradient evaluated at a point is the direction of steepest ascent at that point.* This can be proven trivially from the definition of a directional derivative, as seen below:

0	Find $max(D_{\vec{u}}f)$ at point P_0	Problem
1	Let $\vec{u}$ be some arbitrary vector	Assumption
2	Let θ be the angle between $\vec{u}$ and ∇f	Assumption
2	$\vec{\mathbf{u}} = \frac{\vec{u}}{\lvert\vec{u}\rvert}$	Definition of unit vector
3	$D_{\vec{u}}f = \nabla f \cdot \vec{\mathbf{u}}$	Definition of directional der.
4	$D_{\vec{u}}f = \frac{\nabla f \cdot \vec{u}}{\lvert\vec{u}\rvert}$	Substitution of line 2 into 3
5	$D_{\vec{u}}f = \frac{\lvert\nabla f\rvert\lvert\vec{u}\rvert cos(\theta)}{\lvert\vec{u}\rvert}$	Definition of dot product
6	$D_{\vec{u}}f = \lvert\nabla f\rvert cos(\theta)$	Reduction
7	$max(cos(\theta)) \rightarrow \{\theta \lvert \theta = 2\pi n, n \in \mathbb{Z}\}$	Definition of cosine
8	$max(cos(\theta)) = cos(2\pi n)$	Extension of line 7
9	Let $n = 0$	Selection based on line 7
10	$max(cos(\theta)) = cos(2\pi n) = cos(0)$	Substitution of line 9
11	$max(D_{\vec{u}}f) = max(\lvert\nabla f\rvert cos(\theta))$	Identity
12	$max(D_{\vec{u}}f(P_0)) = max(\lvert\nabla f(P_0)\rvert cos(\theta))$	Evaluating at point
13	$max(D_{\vec{u}}f(P_0)) = \lvert\nabla f(P_0)\rvert max(cos(\theta))$	Pulling out constant
14	$max(D_{\vec{u}}f(P_0)) = \lvert\nabla f(P_0)\rvert cos(0)$	Substitution of line 10
15	$max(D_{\vec{u}}f(P_0)) = \lvert\nabla f(P_0)\rvert$	Reduction
16	$max(D_{\vec{u}}f(P_0)) = \lvert\nabla f(P_0)\rvert$	Result

As demonstrated above, the maximium value of the directional derivative taken at a point is the gradient evaluated at that point.

Gradient descent leverages this to know which direction to *avoid*–to go downhill, simply don't go uphill. With that in mind, we can define the process for gradient descent:

0	Move *away* from the direction of steepest ascent	Problem
1	Let P be an arbitrary starting position	Assumption
2	Let γ be the scalar distance of a step	Assumption
3	$P_{n+1} = P_n - \gamma\nabla f(P_n)$	Gradient Descent

With the formula for gradient descent defined, we can demonstrate an example:

0	Minimize f using gradient descent	Problem
1	Let $f(x,y) = x^2 + y^2$	Assumption
2	Let $\gamma = \frac{1}{2}$	Assumption
3	Let $P_0 = (2,2)$	Assumption
4	$P_{n+1} = P_n - \gamma \nabla f(P_n)$	Gradient Descent
5	$P_1 = (2,2) - \frac{1}{2}\nabla f(P_0)$	Substituting lines 1,2,3
6	$\nabla f = \frac{\partial}{\partial x} f(x,y)\hat{\mathbf{i}} + \frac{\partial}{\partial y} f(x,y)\hat{\mathbf{j}}$	Definition of gradient
7	$\frac{\partial}{\partial x} f(x,y) = 2x$	Reduction
8	$\frac{\partial}{\partial y} f(x,y) = 2y$	Reduction
9	$\nabla f = 2x\hat{\mathbf{i}} + 2y\hat{\mathbf{j}}$	Substituting lines 7,8
10	$\nabla f(P_0) = 2(2)\hat{\mathbf{i}} + 2(2)\hat{\mathbf{j}}$	Substituting line 3
11	$P_1 = (2,2) - \frac{1}{2}(4\hat{\mathbf{i}} + 4\hat{\mathbf{j}})$	Combining lines 5,10
12	$P_1 = (2,2) - (2\hat{\mathbf{i}} + 2\hat{\mathbf{j}})$	Reducing
13	$P_1 = (0,0)$	Reducing
14	$P_2 = P_1 - \frac{1}{2}\nabla(P_1)$	Setting up next step
15	$\nabla f(P_1) = 2(0)\hat{\mathbf{i}} + 2(0)\hat{\mathbf{j}}$	Substituting from line 9
16	$\nabla f(P_1) = 0$	Reducing
17	$P_2 = P_1 - \frac{1}{2}(0)$	Combining lines 14,16
18	$P_2 = P_1$	Reducing
19	$P_2 = P_1$	Done, converged

The process ends when we converge on an answer (this is not always the case, depending on step size we can diverge). In this case the descent was fairly trivial and took only two iterations, but a more complex function like $f(x,y) = \frac{x^2 cos(x)}{16} + \frac{y^2 sin(y)}{24}$ would likely take more steps (and more room than this paper has...).

5 What If...

If the gradient descent calculations had not been done correctly in Lecun et al. (1998), it is unlikely that the paper would have found any training improvements of note. In a best-case scenario, the errors would be caught during peer review and the only impact would be a several-month delay in the publishing of the paper, with the presumed knock-on effect of delaying our current progress in machine learning by several months.

In a worst-case scenario, the paper fails to find anything meaningful and is overlooked. Until some other researcher stumbled upon the idea of using gradient descent to teach neural networks (with the *proper* math in place), neural network research would likely have remained much less accessible due to the ineffectiveness of training techniques. As a result, many common staples of today's technology such as image recognition would be underdeveloped, with unpredictable but certainly negative economic impacts.

The most concerning second-order effects would be to the medical advancements mentioned above–current cutting-edge medical machine learning applications rely heavily on gradient descent. Any delay to the development of these technologies indirectly harms future patients by reducing their potential quality of care. As such, in a worst-case scenario incorrect calculations in Lecun et al. (1998) would cost lives, and conversely the correct calculations may well turn out to save lives.

References

Cauchy, A.-L. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 25:536 – 538. Translated by Richard J. Pulskamp, Department of Mathematics & Computer Science, Xavier University, Cincinnati, OH. July 17, 2010.

Irvin, J. A., Pareek, A., Long, J., Rajpurkar, P., Eng, D. K., Khandwala, N., Haug, P. J., Jephson, A., Conner, K. E., Gordon, B. H., Rodriguez, F., Ng, A. Y., Lungren, M. P., and Dean, N. C. (2022). Chexed: Comparison of a deep learning model to a clinical decision support system for pneumonia in the emergency department. *Journal of Thoracic Imaging*, 37(3):162 – 167.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324.

Rana, M. and Bhushan, M. (2022). Machine learning and deep learning approach for medical image analysis: diagnosis to detection. *Multimedia Tools and Applications*, Dec 24:1 – 39. Epub ahead of print.

A Appendix: How a Neural Network Works

Neural networks are a machine learning architecture commonly used for pattern recognition. Named for their similarity to human brains, traditional neural networks are composed of groups of nodes (called *layers* of *neurons*). Each node in a layer is connected to nodes in the preceding and following layers, and contains some innate value (called a *weight*). To process data with a neural network, data is broken into discrete chunks that are fed into the first (input) layer of neurons. These neurons mutate the input according to their weight, then pass the result to the next (hidden) layer to continue the process. This process continues until the final (output) layer, wherein each node outputs a single number. These numbers can be used to classify inputs–for example, a neural net classifying images into "dog" or "cat" might have two output nodes O_1 and O_2 corresponding to "dog" and "cat" respectively. The classification of an image into "dog" or "cat" would be decided by $max(O_1, O_2)$.

A Simple Diagram of a Neural Network