Neural Networks Overview

As implied by the name, neural network models take inspiration from how neurons in the brain function but do not replicate this process exactly. Early research in artificial intelligence (AI) proposed the idea that neural networks could be trained to “learn” complicated tasks by making minor adjustments in the connections between neurons. The theory of Connectionism simulates this behavior: slight changes are made to the weights of each neuron-to-neuron connection, where the individual weights indicate the strength of the connection and affect the probability of a neuron passing along a signal in a multi-layered perceptron model. In a single-layer perceptron model, we observe the output is calculated by taking the summation of inputs multiplied by weights plus biases and then processed through an activation function. In the event where the network does not produce the correct output, weights are altered by the algorithm. A multi-layered perceptron model differs from a single-layer perceptron model by incorporating hidden layers and backpropagation. Backpropagation is comprised of two stages, a forward and a backward stage, where “in the forward stage, activation functions are originated from the input layer to the output layer, and in the backward stage, the error between the actual observed value and demanded given value is originated backward in the output layer for modifying weights and bias values.” (Tyagi, 2021). Further, neural networks are classified as “the most common class of ‘black-box functions'” as their inner-workings are difficult to observe while inputs and outputs are not (Boix-Adsera, 2021). An imperative element in the training of perceptron models is a labeled training set which is used to compare the output with the expected value and then to alter weights accordingly. Therefore, we would classify this model as a supervised neural network, as opposed to an unsupervised neural network which does not rely on a labeled training set. This implies supervised neural network contain a notion of what the correct outputs are during the training stage. Convolutional neural network (CNNs), a type of neural network, “form hypotheses about images” and are structured so that the output from many layers of neurons is less affected by the main object being altered such as with respect to its position or angle (Bengio, 2016). A key observation of CNNs’ functionality is that “experiments show that artificial neurons in deeper layers in the network tend to correspond to more abstract semantic concepts: a visual object such as a desk, for instance” (Bengio, 2016). This suggests that very deep multi-layered CNNs are optimal for a model to identify objects that contain nuanced features. As might be expected, the key aim of an AI model is to generalize, and thus form accurate predictions beyond the training dataset. However, a models will typically not produce output with absolute accuracy, so we define a measure of error, loss, as “the difference between the ideal output and the actual computed output” (Constantinides, 2020).

In its history, the development of neural network models has entailed several notable challenges. The problem of optimization centers around the fact that it is often difficult to discern based on the loss function what the most optimal solution–the values of weights and biases–for a neural network should be. Another challenge in implementing a neural network model is representing the knowledge of a task–such as recognizing an image or speech recognition. Factors conducive to a task may not be clear-cut and there may be a large quantity of variables, with the most critical ones being difficult to discern. Accordingly, there is no universal machine learning algorithm as “knowledge-acquisition algorithms [need] to be tested on learning tasks and data specific to a certain situation” (Bengio, 2016). What has been conducive to the rapid advancement of AI, however, is a relatively modern development: very powerful and efficient graphics processing units (GPUs). GPUs have been heavily instrumental in accelerating the time required to train large machine learning models. Furthermore, the quantity of large labeled data sets has grown in recent years and has played an influential role for the strides we have seen in deep learning.

.

Data Pruning

Data pruning is a method that involves minimizing the size of training datasets while preserving model performance. Data pruning is a method of interest as we have recent AI models being trained on increasingly larger datasets without regard to whether each instance of data provides any value. Thus, it can be inferred that we should seek the most critical examples–being most representative–within the training dataset for this approach to be effective. A critical ramification of the high compute cost associated with the training of these models is the high amount of CO2 emissions generated, which raises an environmental concern. However, data pruning presents a solution to this problem.

Data pruning amplifies the information the model learns from each example by removing the least meaningful cases within the training dataset. Ben Sorscher, a researcher, observed that the "performance of models tends to improve like a power law" which is not ideal if our aim is to decrease loss as we observe the law of diminishing returns take effect (Sorscher, Smith, 2023). Variables inherent in the power law that formulates a model's error rate are training size, model size, and/or compute power (Sorscher, et al., 2022). To exemplify in a hypothetical example, reducing the error rate from in half would necessitate doubling the size of the training dataset. Further, Sorscher's approach to pruning datasets involves training other models, small-scale and less compute intensive, to discern between "easy" and "hard" examples. "Hard" examples are defined as those containing more subtle features not present in the "easy" or more general examples. An important distinction Sorscher and his team identify is that the approach to data pruning is dependent on the quantity of training data available. In the event where the amount of training data is limited, Sorscher asserts, it is ideal to retain the easy examples, whereas, when training data is abundant, it is ideal to retain the hard examples. This exemplifies the feasibility of scaling a model at a rate that surpasses power laws–achieving exponential scaling or better. However, Sorscher identifies a limitation in his strategy: deciding on the 'data pruning metric' used for discerning between hard and easy examples. This raises the question on what criteria should be used to identify the 'hard' examples and how this may change based on a model's training dataset.

Our principal aim, as can be assumed, is to minimize the costs associated with compute power and scaling. Sorscher and his team assert that improvement past power law scaling is not only possible, but exponential scaling or better is feasible if we can devise a 'data pruning metric' that ranks training examples to discern which to drop to reach a certain pruned dataset size. At the core of the paper (Sorscher, et al., 2022) is the creation of a "new simple, cheap and scalable self-supervised pruning metric that demonstrates comparable performance to the best supervised metrics." Central to this approach is the observation that training examples are highly redundant, implying that training datasets can be optimized by pruning them to a smaller size containing the most 'useful' examples. To demonstrate the success of their approach, Sorcher and his team observed that "pretraining on as little as 50% of ImageNet can match or exceed CIFAR-10 performance obtained by pre-training on all of ImageNet" (Sorscher, et al., 2022). Further, they found that "... all these metrics require labels, thereby limiting their ability to prune data for large-scale foundation models trained on massive unlabeled datasets" which implies there is a need "for simple, scalable self-supervised pruning metrics" (Sorscher, et al., 2022).

Undoubtedly, a self-supervised pruning metric would permit for pruning on models as substantial as large language models (LLMs) and lower the CO2 emissions generated during their training processes. Regarding ethics, Sorscher and his team found “no substantial differential effects across classes" which implies there was no considerable bias in the models tested. A promising avenue of future exploration is “the development of scalable, unsupervised data pruning metrics” as the theory discussed in the paper “predicts that the application of pruning metrics on large scale datasets should yield larger gains by allowing more aggressive pruning.” Therefore, this approach will be instrumental in the development of ‘large foundation models’ as they are trained on ‘massive unlabeled datasets.’

Other researchers have observed that “recently, deep learning has made remarkable progress driven, in part, by training over-parameterized models on even larger datasets” (Paul et al., 2021). However, they concede that the issue of concern lies in the high computational resource cost in training models. Their approach is “a scoring method that can be used to identify important and difficult examples early in training, and prune the training datasets without large sacrifices in test and accuracy” (Paul et al., 2021). This stands in contrast with the approach undertaken by Sorcher’s team that identifies examples to be pruned much later in the process. The researchers describe that “informally, removing a training example from the training data and not hurting the generalization error suggests that the example has small ‘influence’ on the test data” (Paul et al., 2021). Thus, it is ideal to remove the examples that exemplify this characteristic in the training data set. Optimistically, their “first finding is that *very early in training* (just a few epochs), partial forgetting scores identify large fractions of data that can be pruned” (Paul et al., 2021). As a testament to the success of their approach, the researchers found that “indeed, we can prune 50% of examples from CIFAR-10 without affecting accuracy, while on the more challenging CIFAR-100 dataset, we can prune 25% of examples with only a 1% drop in accuracy" (Paul et al., 2021). Evidently, their work "introduces methods to significantly prune data without sacrificing test accuracy using *only* local information *very early* in training, sometimes even at initialization" as well as "uses the resulting methods to obtain new scientific insights into how different subsets of training examples drive the dynamics of deep learning" (Paul et al., 2021). This strategy can thus be utilized as a more efficient method of data pruning although it would warrant a comparison to other models to prove its efficacy.

Model Distillation

Model distillation is defined as the "task of replacing a complicated machine learning model with a simpler one that approximates it well" (Boix-Adsera, 2021). The usefulness of model distillation is evident, as it is less computationally intensive yet delivers similar output to models that require far more compute resources. An approach taken by a team of researchers to implement this method initiates with "training an ensemble of models on the same dataset and then averaging their corresponding predictions" (Hinton et al., 2015). The usefulness of this approach is further demonstrated by a team of researchers who devised a method to enhance compute efficiency as a "large model or ensembles of models can often be distilled to smaller models which are deployable at lower computational cost" (Boix-Adsera, 2021). The authors of the paper also "prove that distillation has a very low sample complexity, *whenever it is possible to distill*" (Boix-Adsera, 2021). This implies that model distillation is dependent on how intricate the dataset is, and thus, this method imposes a limitation. Further research into model distillation is required to expand its implementation as "scaling these methods to foundation models such as LLMs seems like it will pose a significant engineering challenge, and require new ideas" (Boix-Adsera, 2021). This is particularly noteworthy, as we are in the advent of the employment of large language models (LLMs), which are now easily accessible the public but come at the cost of being computationally expensive. However, the a team of researchers succeed in "...transferring knowledge from an ensemble or from a large highly regularized model into a smaller, distilled model" (Hinton et al., 2015). Therefore, a strategy to compact the knowledge contained in various models or in an ensemble into a singular and smaller model is not only possible, but demonstrates success. Model distillation will prove instrumental especially since the "deployment to a large number of users, however, has much more stringent requirements on latency and computational resources" (Hinton et al., 2015). This approach of creating smaller, less computationally intensive models that maintain accuracy will advance existing models, especially those that are publicly accessible with a large user base. Interestingly, "if the cumbersome model generalizes well because, for example, it is the average of a large ensemble of different models, a small model trained to generalize in the same way will typically do much better on test data than a small model that is trained in the normal way on the same training set as was used to train the ensemble" (Hinton et al., 2015). This conveys that the implementation of model distillation on a smaller model–as opposed to training it the conventional way–achieves more optimal output. Further proof of the efficacy of model distillation is demonstrated on MNIST datasets as "...distillation works remarkably well even when the transfer set that is used to train the distilled model lacks any examples of one or more of the classes" (Hinton et al., 2015). This is particularly surprising as it establishes that model distillation is a feasible method where even in the event that examples of classes are nonexistent in the distilled model, accuracy is retained.

Works Cited

Bengio, Y. (2016). Machines who learn. *Scientific American*, *314*(6), 44–51. https://www.jstor.org/stable/26046989

Boix-Adsera, E. (2024). Towards a theory of model distillation. *arXiv preprint arXiv:2403.09053*.

Constantinides, G. A. (2020). Rethinking arithmetic for deep neural networks. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, *378*(2166), 1–15. https://www.jstor.org/stable/26917449

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Paul, M., Ganguli, S., & Dziugaite, G. K. (2021). Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, *34*, 20596-20607.

Sorscher, B., Smith, C. (2023). *Data Pruning for Efficient Machine Learning | Ben Sorscher | Eye on AI #117* [Video]. YouTube. https://www.youtube.com/watch?v=KyIq3NhbT5w

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., & Morcos, A. (2022). Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, *35*, 19523-19536.

Tyagi, N. (2021, December 13). Understanding the perceptron model in a neural network. *Medium*. https://medium.com/analytics-steps/understanding-the-perceptron-model-in-a-neural-network-2b3737ed70a2#:~:text=A%20single%2Dlayer%20perceptron%20model.