Low Rank, Identifiable Neural Networks

Learning Successive Refinements of a Neural Network in Orthogonal Subspaces

Dustin Angerhofer, Joseph Du Toit, Max Fennimore

Theory of Predictive Modeling

CS 580/PHSCS 513R

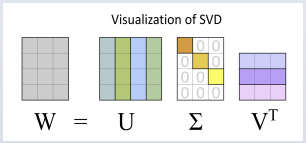
Motivation

Neural networks (NNs) have become the standard model for learning problems and have seen unprecedented success in computer vision, natural language, and medical diagnosis. However, NNs are notoriously over parameterized, rendering them uninterpretable. In this project, we describe a new method to train NNs via low-rank, successive approximation inspired by the singular value decomposition, which lends itself to better interpretability and training complexity than comparable standard NNs.

Methods

A single layer NN can be represented as $\sigma(Wx + b)$, where x is the data, W is a learnable matrix of weights, b is a learnable bias term, and σ is an activation function (such as ReLU, tanh, or sigmoid). Several layers can be composed to create a deep NN.

The key insight of our proposed method is that any weight matrix can be written as its singular value decomposition (SVD) $W = U\Sigma V^T$ where U and V are orthonormal and Σ is diagonal. A rank n SVD uses only the first n columns of U, the first n rows and n columns of Σ, and the first n rows of V. The rank n SVD of W is guaranteed to be the best rank n approximation of W in the Frobenius norm.

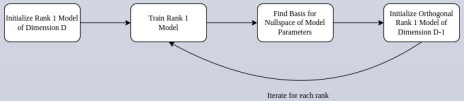


$$\sigma\left(\begin{bmatrix}\mathbf{u}\end{bmatrix}\begin{bmatrix}\mathbf{v}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{x}\end{bmatrix}+\mathbf{b}\right)$$

A rank 1 NN layer. Notice that in our setup, no actual matrix-vector multiplications take place, decreasing computational complexity of a forward pass through a single layer from $O(n^2)$ to $O(n)$.

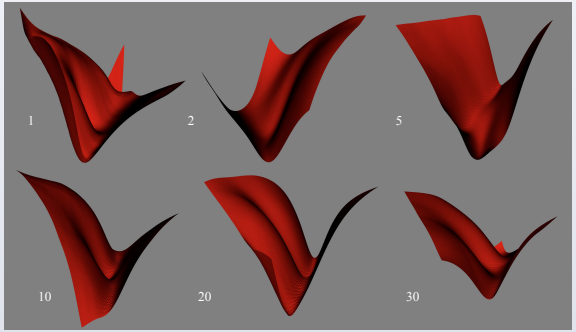
We train the NN successively by beginning with a rank 1 NN. After the rank 1 model has been trained, we freeze the weights of U_1 and V_1 and search their orthogonal complements (the nullspace of the parameters) for the next "singular vectors" to learn a rank 2 approximation. We continue in this manner until we have learned a rank n model. In this way, we first learn the coarse, high importance features, and refine successively with less important, fine level features. This can give intuition about when a NN has learned all of the important features and is now learning noise. Furthermore, each iteration becomes a rank 1 problem, which should be easier to solve.

Successive NN Training Flowchart



Loss Landscapes

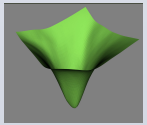
We test our concept on the California Housing dataset, where we try to predict the price of houses based on eight features. We compare our method against a fully-connected NN, using the same learning rate and 256 hidden features. The loss landscapes for ranks 1, 2, 5, 10, 20, and 30 are visualized below.



Although these landscapes are low dimensional approximations of much higher dimensional surfaces and have no sense of scale, each has a noticeable canyon suggesting that the model learns until it reaches unidentifiable parameters at every rank update. Observing the norms of the gradients at each rank shows that the loss landscapes get progressively flatter as well. It should also be noted that these surfaces have dimensionality of order 10^2, whereas the corresponding NN's loss surface has dimensionality of order 10^4.

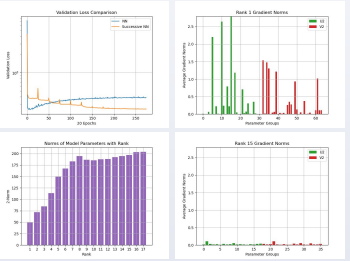
Discussion and Analysis

The loss landscapes of the traditional NN, as pictured to the right, look more convex and do not have the long canyons of unidentifiability. While this looks much nicer from an optimizer's point of view, the minimizer in this case corresponds with the model that overfits on the noise—the NN is too expressive to distinguish between noise and signal. Our low rank models are only expressive enough to learn the most important coarse-level features at its current rank and in its respective orthogonal subspace, which precludes noise until much higher rank updates. If our model cannot fit to the noise, then the unidentifiability inherent in the model should manifest itself, as we see.



Loss landscape of traditional NN on the California Housing data.

More Results and Identifiability Analysis



In the upper left plot, each orange spike corresponds with a rank update of our model, which occurs every 500 epochs. It is important to note that the loss curve is otherwise smooth as compared to an NN. This suggests that our model avoids fitting to noise. In the right two plots, U2 and V2 correspond to U and V from the SVD of the second layer weight matrix. Note on the bottom left, entire parameter groups drop out as the current rank exceeds that group's rank. Adding in their norms would show a more dramatic increasing trend. The upper and lower right plots show the average gradient norms of two parameter groups, U1 and U2, during training of the rank 1 and rank 15 models.

The conventional NN overfits the data early in training, whereas our model continues to improve, eventually outperforming the NN. With each rank update, the amount of improvement diminishes, showing that the most important features are learned in the low ranks. Furthermore, the norms of the gradients generally decrease with each rank update, and the parameter norms generally increase, suggesting that the learnable parameters become increasingly unidentifiable.

Conclusions and Future Work

Successively training low rank approximations for NNs shows potential for increasing identifiability and preventing overfitting. This technique could prove very useful for data with low signal-to-noise ratio.

A key advantage of our method is that it is amenable to identifiability analysis. By isolating the identifiable parameters, we could remove the long, flat canyons in our loss landscapes, making the model simpler and training more efficient. Alternatively, we could use/develop an optimizer that is tailored to the flat loss landscapes that we observe.

Our method also seems useful for identifying noise levels in the data. The rank at which our model starts leaning noise gives us a natural notion of the complexity of the problem of extracting all useful information from the data. Taking an information theoretic approach to assess problem/model complexity and to inform model selection would be interesting.

Furthermore, we could employ a boosting/adaptive sampling strategy to learn better successive ranks by weighing more heavily the residuals on data points that were not well-fitted in the previous rank.