Abstract

The Vapnik–Chervonenkis (VC) dimension of a learning machine is a key concept in the area of model complexity control. The VC dimension is a measure of the capacity of a learning machine. With rapid increases in the availability of training data, and subsequent advent of monolithic deep learning architectures, capacity control of a learning machine has become even more significant. The number of weights in many deep architectures significantly outnumbers the dataset size. The capacity of a machine directly affects its generalizability, due to which architectures with excess capacity tend to over-fit, resulting in poor test performance.

The relationship between the capacity of a learning machine and its generalizability may be best understood from Vapnik’s risk formula, which provides an upper bound on the total risk of a learning machine in terms of its empirical risk and structural risk for a given number of training samples. It shows that there exists an optimal VC dimension for which upper bound on total risk is minimum and if we increase machine capacity beyond this, bound on total risk exhibits increasing trend, leading to poor generalization. This problem is further aggravated for cases where training data is having class imbalance and for some deep learning architectures which are in general difficult to stabilize and control such as generative adversarial networks (GANs).

Several attempts has been made in literature to estimate bounds on structural risk, but they do not translate into a tractable problem to be optimized in terms of learning parameters of the machine. The recently proposed Minimal Complexity Machine (MCM) has shown that a bound on the VC dimension can be directly incorporated into the objective function of a learning machine, resulting in better generalizing capabilities and sparser solutions. The present work focuses primarily on a minimal complexity implementation of Neural Networks, termed as the Low Complexity Neural Network (LCNN). This is done by deriving a differentiable and scalable loss function, that not only minimizes the empirical risk, but also the structural risk of a classifier. This implementation was extended to various deep learning architectures and tested on multiple benchmark datasets.

The Low Complexity loss function was also derived and implemented for Generative Models. It was shown that by introducing complexity term in Generative Adversarial Networks (GANs) and its variants, training stability and quality of generated image was improved and issue of mode collapse was avoided.

Next, Twin Neural Networks, a novel neural network architecture for handling class imbalance is described. Proposed class of machine not only provides a scalable framework for large sized datasets but also keeps structural risk at check resulting in good generalization. It was shown over several benchmark
datasets with very high class imbalance that Twin Neural Networks produces better results than that of competing methods. A real world application of Twin Neural Network for classification of ElectroEncephaloGram (EEG) signals is also discussed.

Subsequent to this, a direct realization of the Minimal Complexity Machine (MCM) was implemented via a neurodynamical system. The proposed system solves coupled differential equations in a multilayer neural network setting, enabling stochastic learning and implicit kernel optimization. The resulting system converges to MCM solution providing better generalization and scalability. Numerical experiments on benchmark datasets show that the proposed approach is scalable and accurate, and learns models with improved accuracies and fewer support vectors.

The last part of the thesis takes an alternative route to Risk Minimization via Data Augmentation. The proposed method is termed as Eigenbag, which presents framework for augmenting data at scale. This becomes important for scenarios, where the number of samples available for training are small. Eigenbag generates new data samples in a lower dimensional space via bagging and projects it back to the original dimension. A deep Learning version of Eigenbag is also discussed that uses Convolutional Autoencoder to increase training sample size at scale. The proposed method is also tested against competing methods.