LECTURE #1

LECTURER: ANNA CHOROMANSKA SCRIBE: KIRAN VODRAHALLI

1 Abstract

The talk will focus on selected challenges in modern large-scale machine learning in two settings: i) large data setting and ii) large model (deep learning) setting. The first part of the talk will focus on the case when the learning algorithm needs to be scaled to large data. The multi-class classification problem will be addressed, where the number of classes (k) is extremely large, with the goal of obtaining train and test time complexity logarithmic in the number of classes. A reduction of this problem to a set of binary classification problems organized in a tree structure will be discussed. A top-down online tree construction approach for constructing logarithmic depth trees will be demonstrated, which is based on a new objective function. Under favorable conditions, the new approach leads to logarithmic depth trees that have leaves with low label entropy. Discussed approach comes with theoretical guarantees following from convex analysis, though the underlying problem is inherently non-convex. The second part of the talk focuses on the theoretical analysis of more challenging non-convex learning setting, deep learning with multilayer networks. Despite the success of convex methods, deep learning methods, where the objective is inherently highly non-convex, have enjoyed a resurgence of interest in the last few years and they achieve state-of-the-art performance. In the second part of the talk we move to the world of non-convex optimization where recent findings suggest that we might eventually be able to describe these approaches theoretically. The connection between the highly non-convex loss function of a simple model of the fully-connected feed-forward neural network and the Hamiltonian of the spherical spin-glass model will be established. It will be shown that under certain assumptions i) for large-size networks, most local minima are equivalent and yield similar performance on a test set, (ii) the probability of finding a bad local minimum, i.e. with high value of loss, is non-zero for small-size networks and decreases quickly with network size, (iii) struggling to find the global minimum on the training set (as opposed to one of the many good local ones) is not useful in practice and may lead to overfitting. Discussion of open problems concludes the talk.

2 eXtreme multi-class classification problem

What has been done so far? Some approaches are intractable: linear in the number of classes. One classical example is one against all. Other approaches don't address problem of learning structure. Conditional probability tree doesn't address multi-class, this is still linear in k. We also have decision trees. These are well-structured to address multi-class with logarithmic training and testing. Splitting criteria are not well-suited to the problem though. Usually you try to take max entropy etc, how do we optimize online. So we will try to decide something else.

2.1 How do you learn the structure?

Not all partitions are equally different. Let us say we do digit recognition. Then $\{1, 7\}$ and $\{3, 8\}$, this is hard (they look similar). Some intuition: better to confuse things near leaves than near the root. You do'nt want root to handle it. Leaves are underconstrained. They see a small fraction of the examples that the root

sees. The approach is top-down with a new splitting criteria which guarantees small classification error, and have a balanced tree $\implies$ logarithmic time training.

2.2 Pure split and balanced split

n_r is on the right side of partitioning and n is total number of data points, $k_r(x)$ is number of data points in same class of x. Measure of balance is simply $\frac{n_r}{n}$, and purity of split is $\frac{k_r(x)}{k(x)}$. If k is number of classes and $\mathcal{H}$ is hypothesis class, then $\pi_y = \frac{|\mathcal{X}_y}{n}$ and balance is $\mathbf{P}\{h(x) > 0\}$ and purity is $\sum_{y=1}^{k} \pi_y \min(\mathbf{P}\{h(x) > 0|y\}, \mathbf{P}\{h(x) < 0|y\})$.

2.3 Objective function

$J(h) = 2\mathbf{E}_{x,y}[|\mathbf{P}\{h(x) > 0\} - \mathbf{P}\{h(x) > 0|y\}]$. Then we have a lemma:

Balancing factor sits in $[\frac{1-\sqrt{1-J(h)}}{2}, \frac{1+\sqrt{1-J(h)}}{2}]$. Purity factor can be upper bounded by $\frac{2-J(h)}{4*\text{balance}}$ times a proportionality constant.

We measure the quality of the tree using entropy. The goal is to show maximizing $J(h)$ actually reduces the entropy. Under Weak Hypothesis Assumption, for any $\epsilon \in [0, 1]$, to obtain $G_T \leq \epsilon$, it suffices to make $\left(\frac{1}{\epsilon}\right)^{\frac{4(1-\gamma)^2 \ln(k)}{\gamma^2}}$. γ is some parameter which is telling you a lower bound on the objective function on a particular node of the tree. This means $\mathcal{O}(\log(k))$ is tree depth: training and testing time. Weak hypothesis just means that there is a hypothesis class that has half the objective lower bounded by γ.

2.4 OMtree algorithm

Recall the objective function. We get discrete optimization problem, and then relax by dropping indicator function and look at margins instead.

It becomes: $J(h) = 2\mathbf{E}_{x,y}[|\mathbf{E}_x[h(x)] - \mathbf{E}_x[h(x)|y]|]$. Keep the online empirical estimates of the expectations. The sign of the difference decides which direction you send the node.

The expected margin is equal to 0.

3 Large Models

Develop theoretical characteristic for given large model.

Why deep learning? We are trying to do theory for deep learning, and trying to find out if there are any models (spin-glass) which theoretical understanding thereof helps us understand empirical observations of deep learning model. Deep learning is learning hierarchical representations. Why the community focuses on deep learning is because it achieves state of the art on the usual things. ImageNet is conv nets.

The goal here is to understand the loss function in deep learning.

Some works on this topic: Ian Goodfellow - empirical studies, Alexander Sachs, and etc. Not a very large group of people that look at deep learning.

Some questions we find interesting:

1. Why the result of multiple experiments with multilayer networks consistently give similar performance?
2. The role of saddle points?
3. Is the surface of multilayered neural nets structured or not?

3.1 Multilayer network and spin-glass model

Look at output of multilayer network.

$Y = \Lambda^{(H-1)/2} \sum_{i=1}^{m} X_i A_i \prod_{k=1}^{H} w_i^k$. H is depth of network, m is number of connections.

X_i is Gaussian random var, A_i are Bernoulli.

We will model max operator as Bernoulli random variables, assume each path is active with same probability. Assume network parametrization is redundant (Denil et al, 2013, Denton et al 2014). Assume that some Λ weights are uniformly distributed in network: every H-length product has equal weight. Also we have spherical assumption on inputs (Gaussians). These are realistic.

Some unrealistic assumptions we'd like to drop. Assume paths have independent input data. Also assume activation mechanism of any path MA_i is independent of the input X_i.

We can say something using spin-glass model.

Basically use energy barriers to analyze and view this network.

You can compute energy barries for spin-glass model. The variance of the solutions you get gets smaller. You observe same thing in deep learning model as you increase the size of the model.

3.2 Spin-glass Model

You have an Ising model. Every side of the grid you have a spin. You have interactions between spins. If these spins interact you get spin-glass model. The probability of the configuration is Gibbs distribution depending on temperature. Ising model makes temperature go down - then spins will freeze, but overall magnetization will not be zero. In spin-glass, when you decrease temperature, the spins will also freeze, but they will freeze so that overall magnetize to 0.

3.3 Why spin-glass and deep-learning

This line of research is just starting.

Theorem 3.1. *Conjecture For large-size networks most local minima are equivalent and yield similar performance on test set.*

Theorem 3.2. *Conjecture The probability of finding a "bad" local minimum is non-zero for small size networks and decreases with network size.*

This kind of makes sense if you have an energy barrier, which arises in the case of a large spin-glass model.

Historically, neural networks were rejected for finding bad quality solutions. The main difference between networks we used to use and what we use now is the size of the models.

Theorem 3.3. *Conjecture*

Saddle points play a key-role in the optmization problem in deep learning.

With overwhelming probability one can find only high-index (large number of negative eigenvalues of the Hessian) saddle points above a certain energy barrier.

Theorem 3.4. *Conjecture Struggling to find the global minimum on the training set as opposed to one of the many good local ones is NOT USEFUL in practice and may lead to overfitting.*

This is a big point of difference with the convex optimization people.

3.4 Take-aways

Optimization landscape is highly non-convex but structured.

For large-size networks, most local minima are equivalent and yield similar performance on a test set.

The probability of finding a bad local minimum is non zero for small nets and decreases for large nets.

Struggling to find global optimum is not the strategy.