<Understanding black box predictions via influence functions> ICML 2017 best paper

Author: Pang Wei Koh, Percy Liang

Summary:

The underlying thesis of this paper is that all the information and "knowledge" learned by a model is derived from the training examples, so it should be possible to inquire how much a particular prediction was influenced by various data points.

This paper tries to answer the following questions, as a preliminary in interpretable models:

- How can we explain the predictions of a black-box model?
- Why did the system make this prediction?
- How can we explain where the model came from?
- How would the prediction change if we up-weighted/modified a training point?

Obviously, perturbing each subset of the training examples and retraining a model is incredibly prohibitive. This is where **influence functions** come in to approximate this difference using second-order approximations (Hessians, etc.), and this paper shows how to scale influence functions to be used at the scale of neural networks.

Pros:

1. Using influence functions to **trace** a model's prediction through the learning algorithm and back to its training set.
2. Scaling up influence functions to modern machine learning models, even for **non-convex** functions and **non-differentiable** functions.
3. Several well-designed use cases, e.g., creating training-set attacks, debugging domain mismatch, helps us better understand the model behavior by looking at how it was **derived** from its training data.

Cons:

1. This paper didn't tested the training-set attacks on discrete data, which could compared with recent approach in 'adversarial example' fields.
2. Though analyzing local changes of a single training point with an infinitesimally-small ϵ allows us to derive efficient closed-form estimates, we want to get more information about global changes, and e.g., how does the model behavior changes if there are a subset of training-points being perturbed?
3. They didn't work out this computation problems for non-convex and non-differentiable functions theoretically. We can only say that this method performed well in both cases for its robustness.

<Dropout Training as Adaptive Regularization> NIPS 2013

Author: Stefan Wager, Sida Wang, Percy Liang

Summary:

The optimization landscape for the network parameters is highly non-convex with many local optima. Even with sufficient training data, the performance of a trained model could still be poor. These problems have been addressed empirically by a surprisingly simple method referred to as dropout. Even so, very little is understood **theoretically** about why this should be the case.

This paper attempts to understand the role of dropout in training deep neural networks. The authors propose modeling **dropout** as noise in the framework of generalized linear models (**GLMs**). This type of analysis has been used to show the effect of training with features that have been corrupted with additive Gaussian noise is equivalent to a form of L2-type regularization in the low noise limit.

Pros:

1. Practical trick in designing dropout noise, which make the expectation of the augmented variable $\tilde{x}_i$ to be x_i.
2. The authors shows that the dropout regularization should be better than L2-regularization for learning weights for features that are rare but highly discriminative. That's because dropout does not penalize those weights.
3. The simulation study is a creative way to verify empirically that their intuition is correct. The derivations and its applications to linear/logistic regression are also very helpful towards future work.

Cons:

1. The authors did make another contribution: semi-supervised dropout training. However, their argument is not convincing. They did not address why their proposal failed to achieve better accuracy for IMDB-2K. The gains in test accuracy amounted to less than one percent and seems to be leveling off as the amount of unlabeled data increased.
2. At the same time, the semi-supervised penalty seems a bit arbitrary. Is this a typical definition?
3. This paper assume a global dropout over the entire neural network. How does the analysis change when dropout occurs at a different rate for each layer?

<Distilling the Knowledge in a Neural Network> NIPS 2014 Deep Learning Workshop

Author: Geoffrey Hinton, Oriol Vinyals, Jeff Dean

Summary:

In short, Knowledge Distill is a simple way to make up for the lack of oversight of the classification problem.

If there is only one target such as label, then the goal of this model is to map the samples of each class in the training sample to the same point. This will actually lose the **intra-class variance** and **inter-class distance** which are helpful for training. However, using the teacher model's output can restore this information. For example, the distance in feature space between cats and dogs must be closer than cats and tables. If an animal does look like a cat or a dog, then it can provide supervision for both types.

In summary, the core idea of KD is to "break up" the supervision information that was originally compressed to a point, so that the output of the student model can be distributed as much as possible to the output of the match teacher model.

Pros:

1. Using the trained model to perform a data augmentation of the original calibration space. This augmentation occurs in the label space, not the image itself.
2. Lowing the temperature which make the association information between classes more obvious.
3. Distilling works very well for transferring knowledge from an ensemble or from a large highly regularized model into a smaller, distilled model.

Cons: (More like a discussion about further work)

1. It is not always necessary to use the teacher model. Retaining additional information during data annotation or collection can also help the training of the model.
2. KD still has many limitations. For example, when the category is small, the effect is not significant, and it is not applicable to non-classification problems.
3. It seems that soft target and label smoothing are inextricably linked. But one is sample wise and one is label wise.

<Harnessing Deep Neural Networks with Logic Rules> ACL 2016 outstanding award

Author: Zhiting Hu, Xuezhree Ma, Zhengzhong Liu, Eduard Hovy, Eric P. Xing

Summary:

This paper developed a framework which combines deep neural networks with first-order logic rules to allow integrating human knowledge and intentions into the neural models.

1. How to represent logic?
   They use soft logic (and first-order), which is characterized by continuous values between [0, 1].
2. How to emerge logic into neural network?
   Posterior regularization which came from the previous about adding rules into statistical model. Make it easy to get a closed-form solution.
3. How to train teacher-student network?
   The entire training procedure is to project student network p to a rule-regularized subspace to get a q, and then use the balance result between q 's output and the real label to update the student network's own p.

Pros:

1. Enhances **general** types of NNs with general types of **knowledge** expressed as logic rules. Although this work has not yet been tried, the continuous value of soft logic makes this job very scalable and can be added with various probabilistic information.
2. The advantage of **posterior regularization** (PR) is simple, which is actually a kind of regularization. It has a closed-form and does not add extra calculations (such as approximate algorithm).
3. The biggest contribution of this work is to change the teacher-network, which is often a separate training, into an iterative EM Algorithm-like training. And in the experiment, the author also confirmed that such an **iterative** together train is helpful.

Cons:

1. The rule of NLP task will be better described. Can other aspects of knowledge (structured knowledge) fit into this framework? Because many expressions in image are low-level pixels, the expression of knowledge will also be very different.
2. This article only checked the rule with the strength of confidence temporarily. Similar with the original Knowledge Distillation problem, although the advantage of claim knowledge distillation is that the knowledge of unlabeled data can be used, the actual effect in the experiment is better with the same **matching labeled training data**.