TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

Implicit Regularization

Implicit Regularization

Any stochastic learning algorithm, such as SGD, determines a stochastic mapping from training data to models.

The algorithm, especially with early stopping, can implicitly incorporate a preference or bias for models.

Implicit Regularization in Linear Regression

Linear regression with many more parameters than data points has many solutions.

But SGD converges to the minimum norm solution.

Implicit Regularization in Linear Regression

For linear regression SGD maintains the invariant that Φ is a linear combination of the (small number of) training vectors.

Any zero-loss (squared loss) solution can be projected on the span of training vectors to give a smaller (or no larger) norm solution.

It can be shown that when the training vectors are linearly independent any zero loss solution in the span of the training vectors is a least-norm solution.

Implicit Regularization of SGD

In a labeling problem a model with parameters Φ defines a model propbability $P_\Phi(y|x)$.

This defines a log loss $-\ln P_\Phi(y|x)$ on which we do gradient descent.

Let $\mathrm{SGD}[P_\Phi, \Phi_{\mathrm{Init}}, \mathrm{Train}]$ be the vector that results from running SGD on model P_Φ with initial parameters Φ_{Init} using training data Train (and a fixed set of hyperparameters, learning rate schedule, and fixed order in which training instances are considered, and fixed number of iterations).

Implicit Regularization of SGD

To get a generalization bound when learning a continuous parameter vector we add Gaussian noise to simulate limited precision of the real numbers.

$$\Phi' = \mathrm{SGD}[P_\Phi, \Phi_{\mathrm{Init}}, \mathrm{Train}] + \epsilon$$

The algorithm defines an **implicit prior:**

$$p\Big(\Phi' \mid P_\Phi, \Phi_{\mathrm{Init}}, \mathrm{Pop}\Big) = E_{\Big(\mathrm{Train} \sim \mathrm{Pop}^N\Big)} \; p\Big(\Phi' \mid P_\Phi, \Phi_{\mathrm{Init}}, \mathrm{Train}\Big)$$

The implicit prior $p\Big(\Phi' \mid P_\Phi, \Phi_{\mathrm{Init}}, \mathrm{Pop}\Big)$ is a valid prior! It does not depend on training data!

Implict Priors: the General Case

Let A be any algorithm mapping a training set Train to a probability density $q_{A,\mathrm{Train}}(\Phi')$ over model parameters Φ'.

The implicit prior defined by algorithm A and the given population distribution is

$$p_{A,\mathrm{Pop}}(\Phi') = E_{\left(\mathrm{Train} \sim \mathrm{Pop}^N\right)} \; q_{A,\mathrm{Train}}(\Phi')$$

A PAC-Bayes Analysis of Implicit Regularization

$$\mathcal{L}(q_{A,\mathrm{Train}}) = E_{\langle x,y\rangle\sim \mathrm{Pop},\;\Phi'\sim q_{A,\mathrm{Train}}}\,\mathcal{L}(\Phi',x,y)$$

$$\hat{\mathcal{L}}(q_{A,\mathrm{Train}}) = E_{\langle x,y\rangle\sim \mathrm{Train},\;\Phi'\sim q_{A,\mathrm{Train}}}\,\mathcal{L}(\Phi',x,y)$$

A PAC-Bayes Analysis of Implicit Regularization

With probability at least $1 - \delta$ over the draw of Train we have

$$\mathcal{L}(q_{A,\mathrm{Train}}) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(q_{A,\mathrm{Train}}) + \frac{5L_{\max}}{N_{\mathrm{Train}}}\left(KL(q_{A,\mathrm{Train}}, p_{A,\mathrm{Pop}}) + \ln\frac{1}{\delta}\right)\right)$$

There is no obvious way to calculate this guarantee.

However, it can be shown that $p_{A,\mathrm{Pop}}$ is the optimal PAC-Bayeisan prior for algorithm A run on data drawn from Pop.

END