Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning

Maksim Tyurikov

MIPT, 2023

November 13, 2023

Motivation

Main idea

There are many approaches to selecting hyperparameters of models. Most of them rely on validation data, which may not be readily available. In this work, are presented a scalable marginal-likelihood estimation method to select both hyperparameters and network architectures, based on the training data alone.

Bayesian models

Bayesian models

We denote by $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ the C - dimensional real-valued output of a neural network with parameters $\boldsymbol{\theta} \in \mathbb{R}^P$, specified by a model M which typically consists of a network architecture and hyperparameters.

Posterior distribution

$$p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}) \propto p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}) p(\boldsymbol{\theta}|\mathcal{M}) \tag{1}$$

A Bayesian model can then be defined using a likelihood and a prior, to get the posterior distribution.

Marginal likelihood

We assume that the data examples are sampled i.i.d. from $p(y_n|f(x_n,\theta),\mathcal{M})$. The normalizing constant of the posterior, also known as the marginal likelihood, is given by the following expression:

$$p(\mathcal{D}|\mathcal{M}) = \int \prod_{i=1}^{N} p(\boldsymbol{y_n}|\boldsymbol{f}(\boldsymbol{x_n},\boldsymbol{\theta}),\mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})d\boldsymbol{\theta} \tag{2}$$

The model $\mathcal{M}$ might consist of the choice of network architecture (CNN, ResNet, etc.) and hyperparameters of the likelihood and prior, for example, observation noise and prior variances. Some of these are continuous parameters while others are discrete.

Empirical Bayes

A simple method is to pick the model that assigns the highest probability to the training data

$$\mathcal{M}_* = \arg\max_{\mathcal{M}}(p(\mathcal{D}|\mathcal{M})) \tag{3}$$

This procedure is also called type-II maximum likelihood estimation or empirical Bayes.

Laplace's method

Approximation

The method relies on a local quadratic approximation of $\log p(\boldsymbol{\theta}|\mathcal{D},\mathcal{M})$, around a maximum $\boldsymbol{\theta}_*$, resulting in a Gaussian approximation to $p(\boldsymbol{\theta}|\mathcal{D},\mathcal{M})$, denoted by $q(\boldsymbol{\theta}|\mathcal{D},\mathcal{M})$, and an approximation to the marginal likelihood, denoted by $q(\mathcal{D}|\mathcal{M})$ and shown below:

$$\log p(\mathcal{D}|\mathcal{M}) \simeq \log q(\mathcal{D}|\mathcal{M}) := \log p(\mathcal{D},\boldsymbol{\theta}_*|\mathcal{M}) - \frac{1}{2} log \mid \frac{1}{2\pi}\boldsymbol{\mathcal{H}}_{\boldsymbol{\theta}_*} \mid \quad (4)$$

with Hessian: $\boldsymbol{\mathcal{H}}_{\boldsymbol{\theta}_*} := \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \log p(\mathcal{D},\boldsymbol{\theta}|\mathcal{M})$

Laplace's method

MargLik

However, computing $\boldsymbol{\mathcal{H}}_{\boldsymbol{\theta}}$, a large matrix of size P $\times$ P, and its determinantcis infeasible in general. We refer to the log marginal likelihood as ***margLik***.

Hessian approximations

Generalized Gauss-Newton (GGN)

$$\boldsymbol{\mathcal{H}}_{\boldsymbol{\theta}} \simeq \boldsymbol{\mathcal{H}}_{\boldsymbol{\theta}}^{\mathcal{GGN}} = \boldsymbol{\mathcal{J}}_{\boldsymbol{\theta}}^{T} \boldsymbol{\mathcal{L}}_{\boldsymbol{\theta}} \boldsymbol{\mathcal{J}}_{\boldsymbol{\theta}} + \boldsymbol{\mathcal{P}}_{\boldsymbol{\theta}} \tag{5}$$

The complexity is $\mathcal{O}(P^2NC + PNC^2)$

Empirical Fisher (EF)

$$\boldsymbol{\mathcal{H}}_{\boldsymbol{\theta}} \simeq \boldsymbol{\mathcal{H}}_{\boldsymbol{\theta}}^{\mathcal{EF}} = \boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}}^{T} \boldsymbol{\mathcal{G}}_{\boldsymbol{\theta}} + \boldsymbol{\mathcal{P}}_{\boldsymbol{\theta}} \tag{6}$$

The complexity is $\mathcal{O}(P^2N)$

MAP

To optimize the network parameters $\boldsymbol{\theta}$ we perform regular neural network training on the maximum a posteriori (MAP).

$$\log p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{M}) = \sum_{n=1}^{N} p(\boldsymbol{y_n}|\boldsymbol{f}(\boldsymbol{x_n}, \boldsymbol{\theta})) + \log p(\boldsymbol{\theta}) \tag{7}$$

Continuous hyperparameters

To optimize the continuous hyperparameters $\mathcal{M}$, we perform gradient ascent on the marginal likelihood estimate

$$\mathcal{M}^{\partial} \leftarrow \mathcal{M}^{\partial} + \gamma \nabla_{\mathcal{M}^{\partial}} \log q(\mathcal{D}|\mathcal{M}) \tag{8}$$

Algorithm

Algorithm 1 Marginal likelihood based training

Input: dataset $\mathcal{D}$, likelihood $p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})$, prior $p(\boldsymbol{\theta}|\mathcal{M})$. Initial model $\mathcal{M}$, step size γ, steps K, burn-in epochs B, marglik frequency F. GGN or EF.
initialize $\boldsymbol{\theta}$ of the neural network
for each *epoch* **do**
  $\boldsymbol{\theta} \leftarrow trainEpoch$(objective in Eq. 6)
  if $epoch > B$ and $epoch \mod F = 0$ **then**
    compute $\log q(\mathcal{D}|\mathcal{M})$ (Eq. 3) with $\mathbf{H}_{\boldsymbol{\theta}}^{\text{GGN}}$ or $\mathbf{H}_{\boldsymbol{\theta}}^{\text{EF}}$
    for K steps **do**
      $\mathcal{M}^{\partial} \leftarrow \mathcal{M}^{\partial} + \gamma \nabla_{\mathcal{M}^{\partial}} \log q(\mathcal{D}|\mathcal{M})$
      update $\mathcal{M}$ and $\log q(\mathcal{D}|\mathcal{M})$
    end for
  end if
end for
Return marginal likelihood $\log q(\mathcal{D}|\mathcal{M})$ and optionally posterior approximation $q(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$.

Example approximation

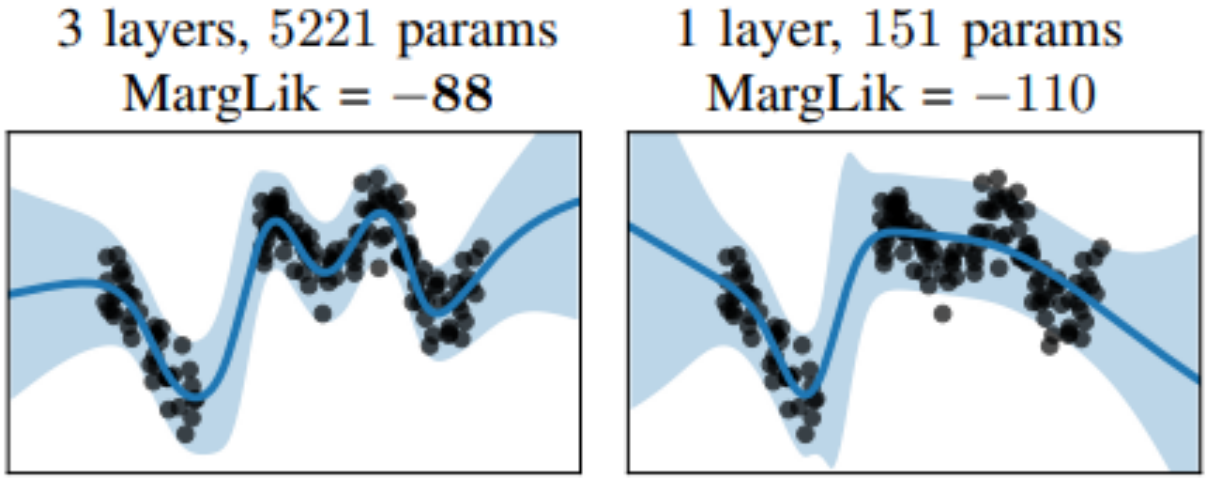

Figure: A deeper 3-layer network (left) has a better marginal likelihood compared to a 1-layer network (right). This agrees with the fit where the deeper network appears to explain the 'sinusoidal' trend better.

The connection between accuracy and MargLik

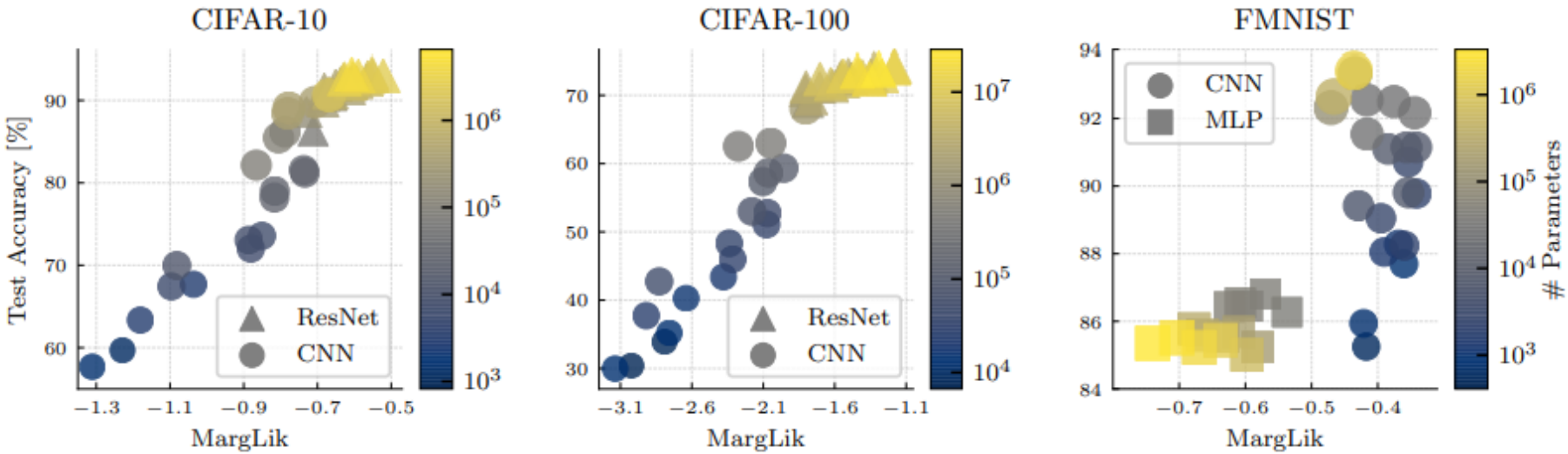

Figure: Each dot above shows a model of different size and/or architecture (around 40 models per plot of varying widths and depths). Models with higher training marginal-likelihood tend to have higher test accuracy. For similar performance, smaller models tend to have a higher marginal-likelihood as desired. Marker size and color changes with the number of parameters.

Literature

1. **Main article** Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning.