Learning Optimal Linear Regularizers

Matthew Streeter

Google

Setup

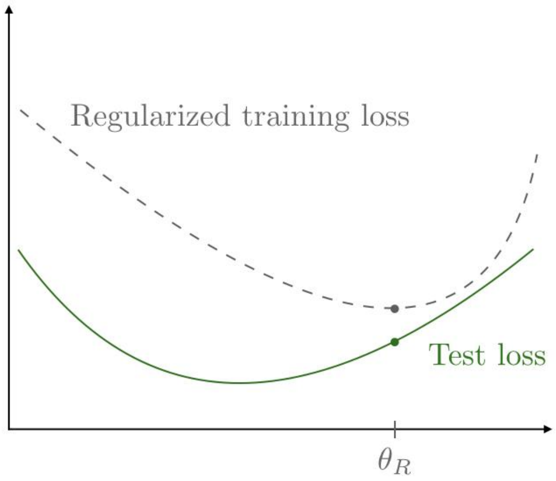
- Want to produce a model θ
- Will minimize training loss + regularizer: $L_{train}(\theta) + R(\theta)$
- Ultimately, we care about test loss: $L_{test}(\theta)$

Setup

- Want to produce a model θ
- Will minimize training loss + regularizer: $L_{train}(\theta) + R(\theta)$
- Ultimately, we care about test loss: $L_{test}(\theta)$
- An optimal regularizer: $R(\theta) = L_{test}(\theta) - L_{train}(\theta)$
  - *suggests that a good regularizer should upper bound the generalization gap*

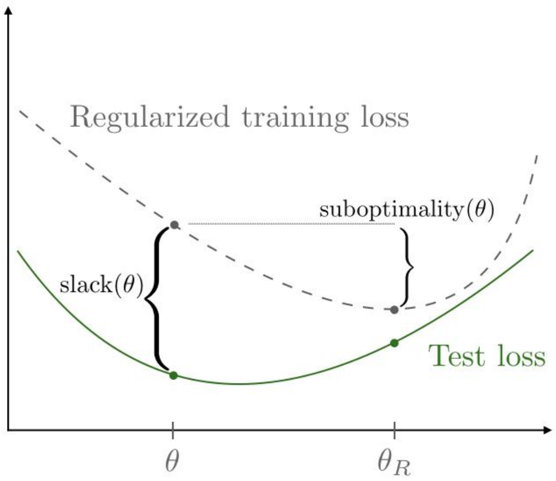
What makes a good regularizer?

- Want to find regularizer R that minimizes $L_{test}(\theta_R)$



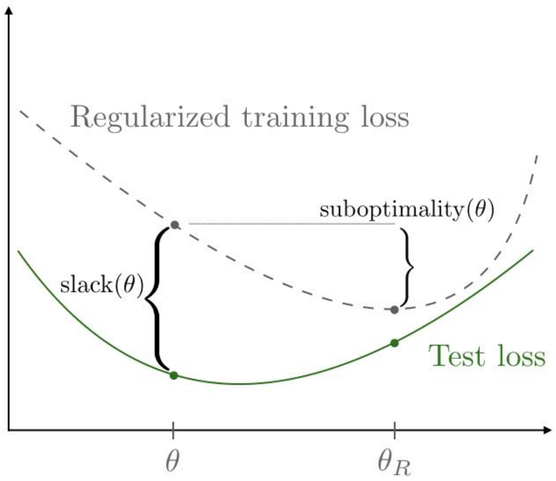
What makes a good regularizer?

- Want to find regularizer R that minimizes $L_{test}(\theta_R)$



What makes a good regularizer?

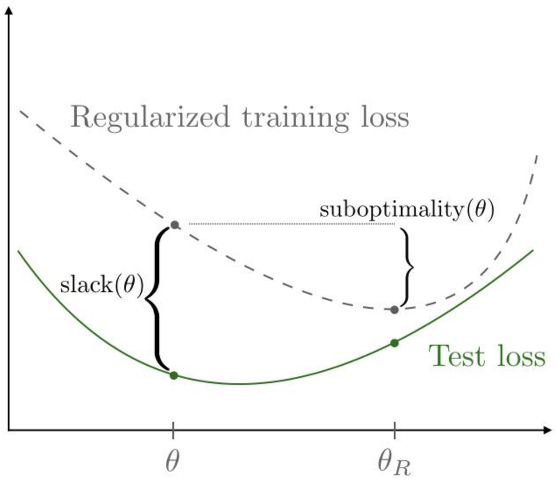
- Want to find regularizer R that minimizes $L_{test}(\theta_R)$



$$\mathrm{L}_{\mathrm{test}}(\theta_{\mathrm{R}}) = \max_{\theta \in \Theta} \{\mathrm{slack}(\theta) - \mathrm{suboptimality}(\theta)\} - \mathrm{slack}(\theta_{\mathrm{R}}) + \mathrm{const}$$

What makes a good regularizer?

- Want to find regularizer R that minimizes $L_{test}(\theta_R)$



$$L_{\text{test}}(\theta_{\text{R}}) = \max_{\theta \in \Theta} \{\text{slack}(\theta) - \text{suboptimality}(\theta)\} - \text{slack}(\theta_{\text{R}}) + \text{const}$$

Approximate by maximizing over small set of models
(estimating test loss using validation set)

Learning linear regularizers

- Linear regularizer: $R(\theta) = \lambda * \text{feature\_vector}(\theta)$

Learning linear regularizers

- Linear regularizer: $R(\theta) = \lambda * \text{feature\_vector}(\theta)$

- **LearnReg**: given models with known training & validation loss, finds best λ (in terms of approximation on previous slide)

Learning linear regularizers

- Linear regularizer: $R(\theta) = \lambda * \text{feature\_vector}(\theta)$

- **LearnReg**: given models with known training & validation loss, finds best λ (in terms of approximation on previous slide)

▶ Solves a sequence of linear programs

Learning linear regularizers

- Linear regularizer: $R(\theta) = \lambda * \text{feature\_vector}(\theta)$
- **LearnReg**: given models with known training & validation loss, finds best λ (in terms of approximation on previous slide)

▶ Solves a sequence of linear programs

▶ Under certain assumptions, can "jump" to optimal λ given data from just $1 + |\lambda|$ models

Learning linear regularizers

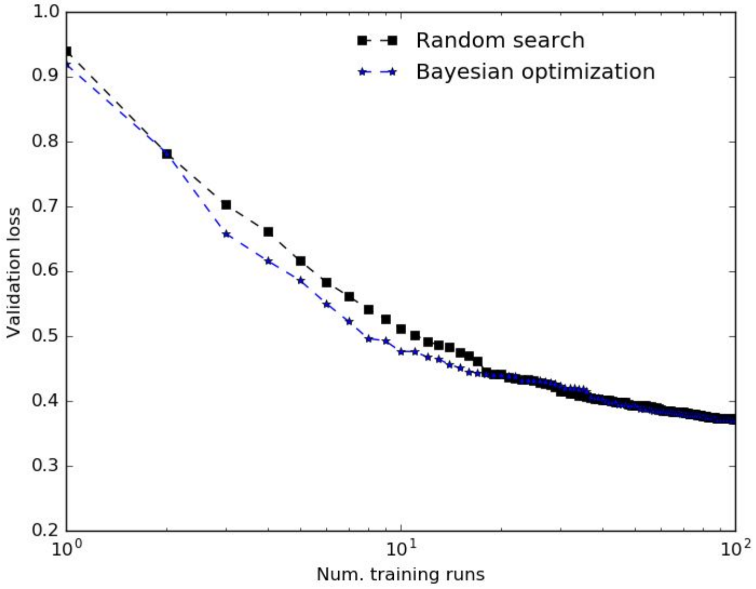
- Linear regularizer: $R(\theta) = \lambda * \text{feature\_vector}(\theta)$

- **LearnReg**: given models with known training & validation loss, finds best λ (in terms of approximation on previous slide)
  - ▶ Solves a sequence of linear programs
  - ▶ Under certain assumptions, can "jump" to optimal λ given data from just $1 + |\lambda|$ models

- **TuneReg:** uses LearnReg iteratively to do hyperparameter tuning

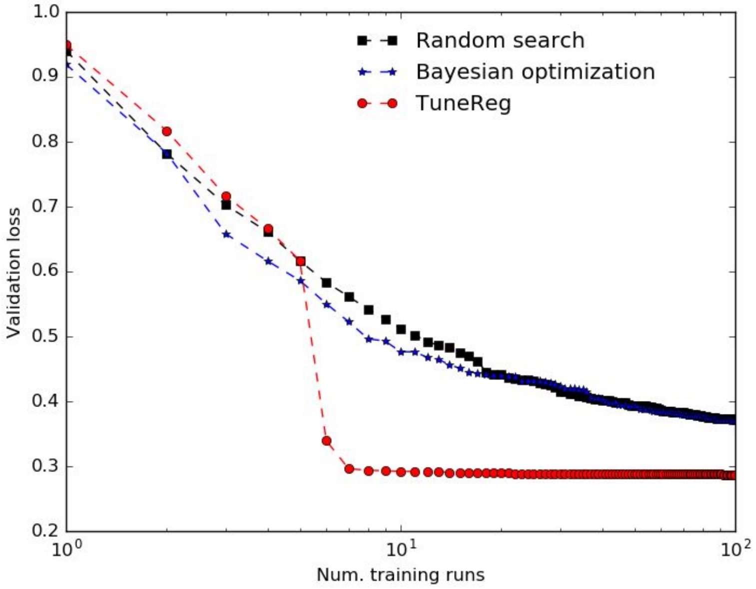
Hyperparameter tuning experiment

- Inception-v3 transfer learning problem, linear combination of 4 regularizers



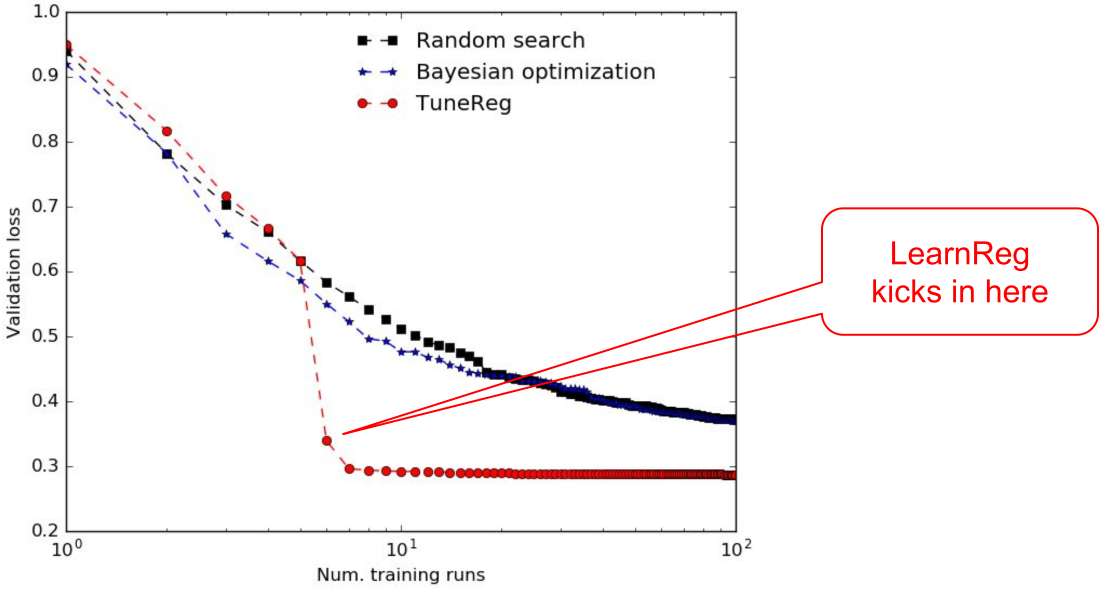
Hyperparameter tuning experiment

- Inception-v3 transfer learning problem, linear combination of 4 regularizers



Hyperparameter tuning experiment

- Inception-v3 transfer learning problem, linear combination of 4 regularizers



Hyperparameter tuning experiment

- Inception-v3 transfer learning problem, linear combination of 4 regularizers