Learning Optimal Linear Regularizers

Matthew Streeter

Google
Setup

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

● Want to produce a model $\theta$

● Will minimize training loss + regularizer: $L_{\text{train}}(\theta) + R(\theta)$

● Ultimately, we care about test loss: $L_{\text{test}}(\theta)$

● An optimal regularizer: $R(\theta) = L_{\text{test}}(\theta) - L_{\text{train}}(\theta)$
  
  ○ suggests that a good regularizer should upper bound the generalization gap
What makes a good regularizer?

- Want to find regularizer $R$ that minimizes $L_{\text{test}}(\theta_R)$
What makes a good regularizer?

- Want to find regularizer $R$ that minimizes $L_{\text{test}}(\theta_R)$
What makes a good regularizer?

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

\[
L_{test}(\theta_R) = \max_{\theta \in \Theta} \{\text{slack}(\theta) - \text{suboptimality}(\theta)\} - \text{slack}(\theta_R) + \text{const}
\]
What makes a good regularizer?

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

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

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

- Linear regularizer: $R(\theta) = \lambda \ast \text{feature\_vector}(\theta)$
Learning linear regularizers

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

- LearnReg: given models with known training & validation loss, finds best $\lambda$ (in terms of approximation on previous slide)
Learning linear regularizers

- Linear regularizer: \( R(\theta) = \lambda \times \text{feature}_\text{vector}(\theta) \)

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

  Solves a sequence of linear programs
Learning linear regularizers

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

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

  - Solves a sequence of linear programs
  - Under certain assumptions, can “jump” to optimal \( \lambda \) given data from just \( 1 + |\lambda| \) models
Learning linear regularizers

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

- **LearnReg**: given models with known training & validation loss, finds best \( \lambda \) (in terms of approximation on previous slide)
  
  - Solves a sequence of linear programs
  - Under certain assumptions, can “jump” to optimal \( \lambda \) 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

LearnReg kicks in here
Hyperparameter tuning experiment

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

LearnReg kicks in here