Hyperparameter Optimization

Benjamin Roth

CIS LMU München

Gliederung

Today

- Attention Q&A
- Hyperparameter Optimization
- Project Q&A (with Ben)

Why Hyperparameter Optimization?

- In the last exercise, some groups used the SGD optimizer, while others used Adam
- Turns out that the embedding-average model failed when trained by SGD
- This is hard to know in advance, so finding the right hyper-parameters is part of the job

Why Hyperparameter Optimization?

- In the last exercise, some groups used the SGD optimizer, while others used Adam
- Turns out that the embedding-average model failed when trained by SGD
- This is hard to know in advance, so finding the right hyper-parameters is part of the job
- Beware of claims like:
- *“My model has 98 % accuracy and trains in under 1 hour*”*

*But I had to search over 1000 hyperparameter configurations, and on a new data set this needs to be done again.

Hyperparameter Optimization

- List as many neural network hyperparameters that you can think of

Hyperparameter Optimization

- List as many neural network hyperparameters that you can think of
  - embedding_size $\in \{10, \ldots, 1000\}$
  - hidden_size $\in \{10, \ldots, 1000\}$
  - l1_regularizer $\in \{0, 0.00001, \ldots, 1.0\}$
  - l2_regularizer $\in \{0, 0.00001, \ldots, 1.0\}$
  - dropout $\in \{0, \ldots, 1.0\}$
  - optimizer $\in \{$rmsprop,adagrad,sgd,$\ldots\}$
  - initial learning rate
  - number of layers
  - nonlinearities

Practical tips for continous parameters

- Open-ended continuous parameters (e.g., hidden size) should be discretized on a (approximate) log scale, e.g.,
  - $0.01, 0.1, 1, 10, 100$
  - $0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100$
  - $0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100$
- Closed-ended continuous parameters (e.g., dropout) should be discretized on a uniform scale

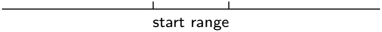
Grid search

- Try all possible combinations in a set of nested for-loops
- Select the combination that does best on the validation set
- Parallelizable, but the search space grows exponentially

Random sampling

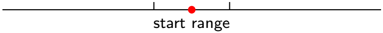
- Set a budget of iterations (e.g., 100)
- On every iteration, sample every parameter independently
- Intuition: Some parameters have a big effect on performance, others have a small effect. A random combination that gets the first group right will be almost as good as the optimal combination.

Intelligent Hyperparameter Optimization



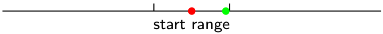
- For every parameter, define a small (*reasonable*) subrange of values
- Use the random sampling method for n iterations.
- If the best value for a parameter lies on (or close to) the boundary of the subrange, shift/extend subrange to the left or right

Intelligent Hyperparameter Optimization



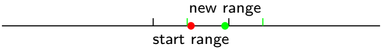
- For every parameter, define a small (*reasonable*) subrange of values
- Use the random sampling method for n iterations.
- If the best value for a parameter lies on (or close to) the boundary of the subrange, shift/extend subrange to the left or right

Intelligent Hyperparameter Optimization



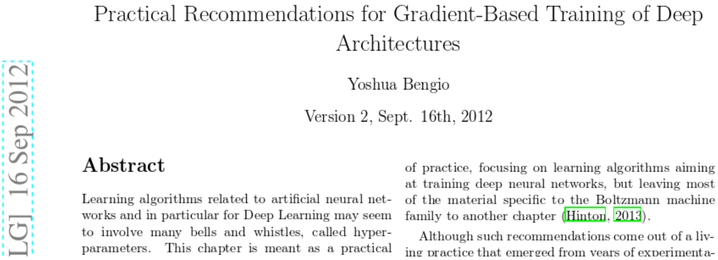
- For every parameter, define a small (*reasonable*) subrange of values
- Use the random sampling method for *n* iterations.
- If the best value for a parameter lies on (or close to) the boundary of the subrange, shift/extend subrange to the left or right

Intelligent Hyperparameter Optimization



- For every parameter, define a small (*reasonable*) subrange of values
- Use the random sampling method for *n* iterations.
- If the best value for a parameter lies on (or close to) the boundary of the subrange, shift/extend subrange to the left or right

More ideas



LG] 16 Sep 2012

Practical Recommendations for Gradient-Based Training of Deep Architectures

Yoshua Bengio

Version 2, Sept. 16th, 2012

Abstract

Learning algorithms related to artificial neural networks and in particular for Deep Learning may seem to involve many bells and whistles, called hyperparameters. This chapter is meant as a practical of practice, focusing on learning algorithms aiming at training deep neural networks, but leaving most of the material specific to the Boltzmann machine family to another chapter (Hinton, 2013).

Although such recommendations come out of a living practice that emerged from years of experimenta-

Validation vs. test set

- Hyperparameter Optimization means overfitting to the validation set
- Never validate on the test set
- The test set should only be used once with your final model. This is the score you report.