Abstract

When doing regression with inputs and outputs that are high-dimensional, it often makes sense to reduce the dimensionality of the inputs before mapping to the outputs. We propose a method where both the dimensionality reduction and the regression mapping can be nonlinear and are estimated jointly. Our key idea is to define an objective function where the low-dimensional coordinates are free parameters, in addition to the dimensionality reduction and the regression mapping. This has the effect of decoupling many groups of parameters from each other, affording a more effective optimization, and to use a good initialization from other methods.

Initialization and Validation

We jointly optimize over both mappings \( g \) and \( f \) using auxiliary coordinates \( Z \) (a \( 4 \)-dimensional mapping, \( \theta \)). This has the effect of decoupling many groups of parameters from each other, affording a more effective optimization, and to use a good initialization from other methods.

Optimization over \( f \) and \( g \)

We use nested functions \( F \)–RBFs on \( \Phi \) for \( f \) and \( g \).

• Linear: \( g \) is a direct regression that acts on a lower input dimension \( D_n \), reduces to least squares problem.

• RBFs \( (\theta) \): \( W \Phi \) with \( \Phi = [\omega_1, \omega_2, \ldots, \omega_{D_n}] \). Gaussian RBFs \( \omega_\alpha \sim N(0, \rho^2) \). Centers \( \alpha \) are chosen by \( \mathcal{L}_2 \) on \( \mathcal{L}_2 \) (since few iterations, initialized at previous centers).

• Weights \( W \) have a unique solution given by a linear system.

• Time complexity: \( O(N D_n + D_n^2) \), linear in training set size.

• Complexity: \( O(D_n D_m) \), \( D_m \) linear in output size.

Low-dimensional regression using auxiliary coordinates

Given a training set \( \mathcal{X}_{\text{train}}, \mathcal{Y}_{\text{train}} \) instead of directly optimizing

\[
L(X, f) = \frac{1}{2n} \sum (y_i - f(x_i))^2 + \lambda \|g\|_2^2 + \lambda_2 \|F\|_2^2
\]

where \( \lambda_2 \geq 0 \) for dimension reduction mapping \( f \) and regression mapping \( g \), we let the low-dimensional coordinates \( X_{\text{auxcoord}}, Y_{\text{auxcoord}} \) be independent, auxiliary parameters to be optimized over, and unfold the squared error into two terms that decouple given \( Z \):

\[
L(X, f, Z) = \frac{1}{2n} \sum (y_i - g(Z_i)) - f(x_i))_{\|Z\|_2^2}^2 + \lambda_2 \|F\|_2^2
\]

This has the effect of decoupling many groups of parameters from each other, affording a more effective optimization, and to use a good initialization from other methods.

Advantages of low-dimensional regression

• We jointly optimize over both mappings \( f \) and \( g \), unlike one-shot methods.

• Our optimization is more efficient than using a deep network with nested mappings (pretty good model/prety fast).

• The low-dimensional regressor has fewer parameters when \( D_n \) is small or \( \mathcal{RBFs} \) is small.

• The smooth functions \( f \) and \( g \) improve regularization on the regressor and may result in a better generalization performance.

Optimization over \( Z \)

For fixed \( f \) and \( g \), optimization of the objective function decouples over each \( \alpha \) \( \in \mathcal{R}^D \).

• We have \( N \) independent nonlinear minimizations each on \( D_n \) parameters, of the form

\[
\text{minimize } w_{\alpha}(X) = \|y - g(Z(X)) - f(x)\|^2 + \lambda_2 \|g\|^2
\]

• If \( g \) is linear, then \( \alpha \) can be solved in closed form by solving a linear system of \( \alpha \).

• If \( g \) is nonlinear, we use Gauss-Newton method with line search.

• Cost over all \( Z \): \( O(N D_n) \), \( D_n \) in training set size.

• The distribution of the coordinates \( Z \) changes dramatically in the first few iterations, while the error decreases quickly, but after that \( Z \) changes little.

Experimental evaluation

• We use \( g \) as a test on the test set (ESR), and \( f \) as our regression function for testing, which is the natural “out-of-sample” extension for above optimization.

• We use \( g \) as a test on the test set (ESR), and \( f \) as our regression function for testing, which is the natural “out-of-sample” extension for above optimization.

• Early stopping for training, usually happens in 100 iterations.

Validation of \( D_m \) by our algorithm.

Comparison of run time of our approach and optimizing the nested objective function.