Abstract

- Many architectures share the fundamental design principle of constructing a deeply nested mapping from inputs to outputs.
- Learning these architectures is challenging because nesting (i.e., function composition) produces inherently nonconvex functions.
- Backprop suffers from vanishing gradients and is hard to parallelize, is only applicable if the mappings are differentiable with respect to the parameters, and needs careful tuning of learning rates.
- Selecting the best architecture, for example the number of units in each layer of a deep net, or the number of filterbanks in a speech post-processing step, is an inherently combinatorial search.
- We describe a general optimization strategy called method of auxiliary coordinates (MAC). It has provable convergence, is easy to implement, and reusing existing algorithms for single layers, can be parallelized trivially and massively, applies even when parameter derivatives are not available or not desirable, can perform model selection on the fly, and is competitive with state-of-the-art nonlinear optimizers even in the serial computation setting, often providing reasonable models within a few iterations.

The method of auxiliary coordinates (MAC)

A typical objective function to learn a deep net with $K$ hidden layers:
\[
\min_w \frac{1}{2} \sum_{n=1}^{N} ||y_n - f(x_n; W)||^2
\]
\[
f(x; W) = f_K(\ldots f_1(x; W_1); W_2)\ldots W_K)
\]
where each layer function has the form $f_i(x; W_k) = \sigma(W_i x)$.

1. Introduce one auxiliary variable per data point/hidden unit and define the following equality-constrained optimization problem:
\[
\min_{W,Z} \frac{1}{2} \sum_{n=1}^{N} ||y_n - f_K(z_{K,n}; W_K)||^2
\]
\[
s.t. \ \{z_{K,n} = f_K(z_{K-1,n}; W_K)\} \ \{z_{1,n} = f_1(x_n; W_1)\} \ \ n = 1,\ldots, N.
\]
Each $z_{k,n}$ can be seen as the coordinates of $x_n$ in an intermediate feature space, or as the hidden unit activations for $x_n$.

2. Apply the quadratic-penalty method (or aug. Lagrangian). Optimize the following function over $(W,Z)$ for fixed $\mu > 0$ and drive $\mu \to \infty$:
\[
\min_{w,z} \frac{1}{2} \sum_{n=1}^{N} ||y_n - f_K(z_{K,n}; W_K)||^2 + \frac{\mu}{2} \sum_{k=1}^{K} \sum_{n=1}^{N} ||z_{k,n} - f_k(z_{k-1,n}; W_k)||^2.
\]
This defines a continuous path $(W^{\mu}(\mu), Z^{\mu}(\mu))$ which converges to a minimum of the constrained problem (2) and original problem (1).

3. Apply alternating optimization over $W$ and $Z$:
- W-step: Minimizing over $W$ for fixed $Z$ results in a separate minimization over the weights of each hidden unit—a single-layer, unit-bound problem that is solved with existing algorithms.
- Z-step: Minimizing over $Z$ for fixed $W$ separates over the coordinates $z_n$ for each data point $n = 1,\ldots, N$:
\[
\min_{z_n} \frac{1}{2} ||y - f_K(z_K; W_K)||^2 + \frac{\mu}{2} \sum_{k=1}^{K} \sum_{n=1}^{N} ||z_{k,n} - f_k(z_{k-1,n}; W_k)||^2,
\]
another nonlinear least squares problem that can be solved using the derivatives w.r.t. $z$ of the single-layer functions $f_1,\ldots, f_{K+1}$.

After stopping, we can apply a fast post-processing step to reduce the objective, achieve feasibility and eliminate the auxiliary coordinates.

Model selection

- Model selection may be achieved “on the fly” by having the W-step do model selection separately for each layer (e.g., with criteria like BIC, AIC or minimum description length, or cross-validation).
- Instead of testing $M^K$ deep nets, with MAC we can test only $M\times K$ single-layer nets (in parallel) at each model-selection iteration.

Deep autoencoder ($K = 3$) with USPS handwritten digit images.

RBF autoencoder with COIL-20 images. Learning the architecture of RBF autoencoder. Parallel processing speedup of MAC.