Motivation

- There is significant redundancy in the parameterization of deep learning models.
- Prevent using a full weight matrix learn a low rank factorization $W = UV$.
- Many options for $U$:
  - Easy choices: Random projections (Gaussian IID); Random connections (Columns of identity).
  - These still don’t work very well.
  - Better choice: Kernel ridge regression: $W = k_0^2(K_0 + \lambda)^{-1}V; (U = k_0^2(K_0 + \lambda)^{-1})$
  - For images use the squared exponential kernel.
  - When no topology is available use the empirical covariance, or the empirical squared covariance.
  - Pre-train layerwise as an autoencoder to initialize the kernels.

Feature Prediction

- Instead of learning a full weight matrix learn a low rank factorization $W = UV$.
- Optimize both factors: doesn’t work well.
- Factorization is still redundant: $W = (UQ)(Q^{-1}V)$. 
- Remove redundancy by fixing $U$. Learn only $V$.
- For images, use the squared exponential kernel.
- When no topology is available use the empirical covariance, or the empirical squared covariance.
- Pre-train layerwise as an autoencoder to initialize the kernels.

Two perspectives

- Feature completion: $g^{-1}(h) = v(UV) = vW$.
  - Predict $W$ using ridge regression, evaluate the network like normal.
- Spatial pooling: $g^{-1}(h) = (vU)V = vV$.
  - $U$ matrix applies a linear pooling operator to $v$.
  - $V$ contains the weights of an ordinary layer with the pooled representation as input.

Columnar architecture

- Split $W$ into blocks: $W = \{W_1, \ldots, W_j\}_j$ factor each $W_j = U_j V_j$ separately.
- Several columns of representation, each with different preprocessing.

Pooling Filters

- Examples of fixed filters in the expander matrix (squared exponential kernel).
- Left: $k_0^2(K_0 + \lambda)^{-1}$.
- Below: $k_0^2$.

Convolutional Networks on CIFAR-10

- Input: $32 \times 32 \times 3$.
- Layer 1 (convolution): 48 filters of size $8 \times 8 \times 3$.
- Layer 2 (convolution): 64 filters of size $8 \times 8 \times 48$.
- Layer 3 (convolution): 64 filters of size $5 \times 5 \times 64$.
- Layer 4 (dense): 500 hidden units.
- Layer 5 (softmax): 10 classes.

- The convolutional layers each have one column and the fully connected layer has five columns.
- Convolutional layers have a natural topological structure, so we use an expander matrix constructed with the squared exponential kernel in each convolutional layer. The input to the fully connected layer at the top of the network comes from a convolutional layer so we use ridge regression with the squared exponential kernel to predict parameters in this layer as well.

Reconstruction ICA

- Comparison of the performance of RICA with and without parameter prediction on CIFAR-10 and STL-10. Both plots use RICA with a single column.

Future Work

- Better kernels, especially when no natural topology exists.
- Deep columns.
- Better sparse sampling techniques: locally dense?
Predicting Parameters in Deep Learning

Misha Denil\(^1\) Babak Shakibi\(^1\) Laurent Dinh\(^2\) Marc'Aurelio Ranzato\(^3\) Nando de Freitas\(^1\)

\(^1\)University of British Columbia \(^2\)École Centrale de Paris \(^3\)Google Inc.

http://www.cs.ubc.ca/~mdenil