Stochastic Optimization for Deep CCA via Nonlinear Orthogonal Iterations

Weiran Wang

Toyota Technological Institute at Chicago

* Joint work with Raman Arora (JHU), Karen Livescu and Nati Srebro (TTIC)

53rd Allerton Conference on Communication, Control, and Computing

September 30, 2015

Multi-view feature learning

Training data consists of samples of a D-dimensional random vector that has some natural split into two sub-vectors:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}, \quad \mathbf{x} \in \mathbb{R}^{D_x}, \quad \mathbf{y} \in \mathbb{R}^{D_y}, \quad D_x + D_y = D.^*$$

Natural views: audio+video, audio+articulation, text in different languages ...

Abstract/synthetic: word+context words, different parts of a parse tree ...

- Task: extracting useful features/subspaces in the presence of multiple views which contain complementary information.
- Motivations: noise suppression, soft supervision, cross-view retrieval/generation ...

---

*We assume feature dimensions have zero mean for notational simplicity.

Canonical correlation analysis (CCA) [Hotelling 1936]

- Given: data set of N paired vectors $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$, which are samples of random vectors $\mathbf{x} \in \mathbb{R}^{D_x}$, $\mathbf{y} \in \mathbb{R}^{D_y}$.
- Find: direction vectors $(\mathbf{u}, \mathbf{v})$ that maximize the correlation

$$
\begin{aligned}
(\mathbf{u}, \mathbf{v}) &= \operatorname*{argmax}_{\mathbf{u},\mathbf{v}} \; \mathrm{corr}(\mathbf{u}^\top \mathbf{x}, \mathbf{v}^\top \mathbf{y}) \\
&= \operatorname*{argmax}_{\mathbf{u},\mathbf{v}} \; \frac{\mathbf{u}^\top \boldsymbol{\Sigma}_{xy} \mathbf{v}}{\sqrt{(\mathbf{u}^\top \boldsymbol{\Sigma}_{xx} \mathbf{u})(\mathbf{v}^\top \boldsymbol{\Sigma}_{yy} \mathbf{v})}}
\end{aligned}
$$

  where $\boldsymbol{\Sigma}_{xy} = \sum_{i=1}^N \mathbf{x}_i \mathbf{y}_i^\top$, $\boldsymbol{\Sigma}_{xx} = \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top$, $\boldsymbol{\Sigma}_{yy} = \sum_{i=1}^N \mathbf{y}_i \mathbf{y}_i^\top$.
- Subsequent direction vectors maximize the same correlation, subject to being uncorrelated with previous directions.

Canonical correlation analysis (CCA)

Extracting L-dimensional projections $\mathbf{U} \in \mathbb{R}^{D_x \times L}$, $\mathbf{V} \in \mathbb{R}^{D_y \times L}$

$$\max_{\mathbf{U},\mathbf{V}} \quad \mathrm{tr}\left(\mathbf{U}^\top \boldsymbol{\Sigma}_{xy} \mathbf{V}\right)$$
$$\text{s.t.} \quad \mathbf{U}^\top \boldsymbol{\Sigma}_{xx} \mathbf{U} = \mathbf{V}^\top \boldsymbol{\Sigma}_{yy} \mathbf{V} = \mathbf{I}.$$

Closed-form solution obtained by SVD of $\tilde{\boldsymbol{\Sigma}}_{xy} = \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1/2}$.

Canonical correlation analysis (CCA)

Extracting L-dimensional projections $\mathbf{U} \in \mathbb{R}^{D_x \times L}$, $\mathbf{V} \in \mathbb{R}^{D_y \times L}$

$$\max_{\mathbf{U},\mathbf{V}} \quad \operatorname{tr}\left(\mathbf{U}^\top \boldsymbol{\Sigma}_{xy} \mathbf{V}\right)$$
$$\text{s.t.} \quad \mathbf{U}^\top \boldsymbol{\Sigma}_{xx} \mathbf{U} = \mathbf{V}^\top \boldsymbol{\Sigma}_{yy} \mathbf{V} = \mathbf{I}.$$

Closed-form solution obtained by SVD of $\tilde{\boldsymbol{\Sigma}}_{xy} = \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1/2}$.

Alternative formulation

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. Then CCA equivalently solves

$$\min_{\mathbf{U},\mathbf{V}} \quad \frac{1}{2} \left\| \mathbf{U}^\top \mathbf{X} - \mathbf{V}^\top \mathbf{Y} \right\|_F^2 = \frac{1}{2} \sum_{i=1}^{N} \left\| \mathbf{U}^\top \mathbf{x}_i - \mathbf{V}^\top \mathbf{y}_i \right\|^2$$
$$\text{s.t.} \quad (\mathbf{U}^\top \mathbf{X})(\mathbf{X}^\top \mathbf{U}) = (\mathbf{V}^\top \mathbf{Y})(\mathbf{Y}^\top \mathbf{V}) = \mathbf{I}.$$

Canonical correlation analysis (CCA)

Extracting L-dimensional projections $\mathbf{U} \in \mathbb{R}^{D_x \times L}$, $\mathbf{V} \in \mathbb{R}^{D_y \times L}$

$$\max_{\mathbf{U},\mathbf{V}} \quad \mathrm{tr}\left(\mathbf{U}^\top \mathbf{\Sigma}_{xy} \mathbf{V}\right)$$

$$\text{s.t.} \quad \mathbf{U}^\top \mathbf{\Sigma}_{xx} \mathbf{U} = \mathbf{V}^\top \mathbf{\Sigma}_{yy} \mathbf{V} = \mathbf{I}.$$

Closed-form solution obtained by SVD of $\tilde{\mathbf{\Sigma}}_{xy} = \mathbf{\Sigma}_{xx}^{-1/2} \mathbf{\Sigma}_{xy} \mathbf{\Sigma}_{yy}^{-1/2}$.

Alternative formulation

Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$. Then CCA equivalently solves

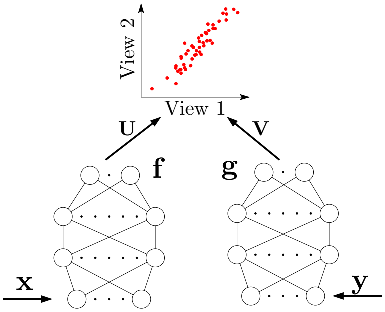
$$\min_{\mathbf{U},\mathbf{V}} \quad \frac{1}{2} \left\| \mathbf{U}^\top \mathbf{X} - \mathbf{V}^\top \mathbf{Y} \right\|_F^2 = \frac{1}{2} \sum_{i=1}^{N} \left\| \mathbf{U}^\top \mathbf{x}_i - \mathbf{V}^\top \mathbf{y}_i \right\|^2$$

$$\text{s.t.} \quad (\mathbf{U}^\top \mathbf{X})(\mathbf{X}^\top \mathbf{U}) = (\mathbf{V}^\top \mathbf{Y})(\mathbf{Y}^\top \mathbf{V}) = \mathbf{I}.$$

But CCA can only find linear subspaces ...

Deep CCA [Andrew, Arora, Bilmes and Livescu 2013]

- Transform the input of each view nonlinearly with deep neural networks (DNNs), such that the canonical correlation (measured by CCA) between the outputs is maximized.



Final projection: $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{U}^\top \mathbf{f}(\mathbf{x}),\ \ \tilde{\mathbf{g}}(\mathbf{y}) = \mathbf{V}^\top \mathbf{g}(\mathbf{y})$.

- A parametric nonlinear extension of CCA $\rightarrow$ better scaling to large data than the kernel extension of CCA [Lai & Fyfe 2000, Akaho 2001, Melzer *et al.* 2001, Bach & Jordan 2002].

Deep CCA: objective and gradient

Objective over DNN weights $(\mathbf{W_f}, \mathbf{W_g})$ and CCA projections $(\mathbf{U}, \mathbf{V})$

$$\max_{\mathbf{W_f},\mathbf{W_g},\mathbf{U},\mathbf{V}} \quad \operatorname{tr}\left(\mathbf{U}^\top \mathbf{F}\mathbf{G}^\top \mathbf{V}\right)$$

$$\text{s.t.} \quad \mathbf{U}^\top \mathbf{F}\mathbf{F}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{G}\mathbf{G}^\top \mathbf{V} = \mathbf{I},$$

where $\mathbf{F} = \mathbf{f}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_N)]$ and $\mathbf{G} = \mathbf{g}(\mathbf{Y}) = [\mathbf{g}(\mathbf{y}_1), \dots, \mathbf{g}(\mathbf{y}_N)]$.

Deep CCA: objective and gradient

Objective over DNN weights $(\mathbf{W_f}, \mathbf{W_g})$ and CCA projections $(\mathbf{U}, \mathbf{V})$

$$\max_{\mathbf{W_f},\mathbf{W_g},\mathbf{U},\mathbf{V}} \quad \mathrm{tr}\left(\mathbf{U}^\top \mathbf{F}\mathbf{G}^\top \mathbf{V}\right)$$

$$\text{s.t.} \quad \mathbf{U}^\top \mathbf{F}\mathbf{F}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{G}\mathbf{G}^\top \mathbf{V} = \mathbf{I},$$

where $\mathbf{F} = \mathbf{f}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_N)]$ and $\mathbf{G} = \mathbf{g}(\mathbf{Y}) = [\mathbf{g}(\mathbf{y}_1), \dots, \mathbf{g}(\mathbf{y}_N)]$.

Gradient computation [Andrew, Arora, Bilmes and Livescu 2013]

Let $\mathbf{\Sigma}_{fg} = \mathbf{F}\mathbf{G}^\top$, $\mathbf{\Sigma}_{ff} = \mathbf{F}\mathbf{F}^\top$, $\mathbf{\Sigma}_{gg} = \mathbf{G}\mathbf{G}^\top$, and $\tilde{\mathbf{\Sigma}}_{fg} = \mathbf{\Sigma}_{ff}^{-1/2}\mathbf{\Sigma}_{fg}\mathbf{\Sigma}_{gg}^{-1/2} = \tilde{\mathbf{U}}\Lambda\tilde{\mathbf{V}}^\top$ be its SVD. Then

$$\frac{\partial \sum_l \sigma_l(\tilde{\mathbf{\Sigma}}_{fg})}{\partial \mathbf{F}} = 2\Delta_{ff}\mathbf{F} + \Delta_{fg}\mathbf{G},$$

$$\text{where} \quad \Delta_{ff} = -\frac{1}{2}\mathbf{\Sigma}_{ff}^{-1/2}\tilde{\mathbf{U}}\Lambda\tilde{\mathbf{U}}^\top\mathbf{\Sigma}_{ff}^{-1/2}, \quad \Delta_{fg} = \mathbf{\Sigma}_{ff}^{-1/2}\tilde{\mathbf{U}}\tilde{\mathbf{V}}^\top\mathbf{\Sigma}_{gg}^{-1/2}.$$

Gradients with respect to $(\mathbf{W_f}, \mathbf{W_g})$ are computed via back-propagation.

Stochastic optimization of deep CCA

$$\max_{\mathbf{W_f},\mathbf{W_g},\mathbf{U},\mathbf{V}} \quad \mathrm{tr}\left(\mathbf{U}^\top \mathbf{F} \mathbf{G}^\top \mathbf{V}\right)$$
$$\text{s.t.} \quad \mathbf{U}^\top \mathbf{F} \mathbf{F}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{G} \mathbf{G}^\top \mathbf{V} = \mathbf{I},$$

- Objective is not expectation of loss over samples due to the constraints. More difficult than PCA/PLS [Arora, Cotter, Livescu and Srebro 2012].
- Exact gradient computation requires feeding-forward all data through the DNNs.
- Back-propagation requires large memory for large DNNs, can not be run on GPUs.

Stochastic optimization of deep CCA

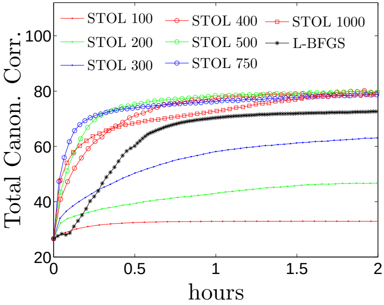
$$\max_{\mathbf{W_f},\mathbf{W_g},\mathbf{U},\mathbf{V}} \quad \mathrm{tr}\left(\mathbf{U}^\top \mathbf{F}\mathbf{G}^\top \mathbf{V}\right)$$
$$\text{s.t.} \quad \mathbf{U}^\top \mathbf{F}\mathbf{F}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{G}\mathbf{G}^\top \mathbf{V} = \mathbf{I},$$

- Objective is not expectation of loss over samples due to the constraints. More difficult than PCA/PLS [Arora, Cotter, Livescu and Srebro 2012].
- Exact gradient computation requires feeding-forward all data through the DNNs.
- Back-propagation requires large memory for large DNNs, can not be run on GPUs.

Question: can we do stochastic optimization for deep CCA?

Approach I : Large minibatches (STOL)

- Use a minibatch of n samples to estimate $\hat{\mathbf{\Sigma}}_{fg}^{(n)} = \hat{\mathbf{\Sigma}}_{ff}^{-1/2} \hat{\mathbf{\Sigma}}_{fg} \hat{\mathbf{\Sigma}}_{gg}^{-1/2}$ and the gradient. Works well for n large enough [Wang, Arora, Livescu and Bilmes 2015].



- Does not work for small n because $\mathbb{E}\left[\hat{\mathbf{\Sigma}}_{fg}^{(n)}\right] \neq \tilde{\mathbf{\Sigma}}_{fg}$, due to the nonlinearities in computing $\tilde{\mathbf{\Sigma}}_{fg}$ (matrix inversion, multiplication).
- Therefore, the gradient estimated on a minibatch is not unbiased estimate of the true gradient.

Alternating least squares for CCA

Alternating least squares [Golub & Zha 1995]: run orthogonal iterations to obtain singular vectors $(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$ of $\tilde{\mathbf{\Sigma}}_{fg}$.

Input: Data matrices $\mathbf{X}$, $\mathbf{Y}$. Initial $\tilde{\mathbf{U}}_0 \in \mathbb{R}^{d_x \times L}$ s.t. $\tilde{\mathbf{U}}_0^\top \tilde{\mathbf{U}}_0 = \mathbf{I}$.

$\mathbf{A}_0 \leftarrow \tilde{\mathbf{U}}_0^\top \mathbf{\Sigma}_{ff}^{-\frac{1}{2}} \mathbf{X}$

for $t = 1, 2, \ldots, T$ **do**

$\mathbf{B}_t \leftarrow \mathbf{A}_{t-1} \mathbf{Y}^\top \left(\mathbf{Y}\mathbf{Y}^\top \right)^{-1} \mathbf{Y}$ % Least squares regression $\mathbf{Y} \rightarrow \mathbf{A}_{t-1}$

$\mathbf{B}_t \leftarrow \left(\mathbf{B}_t \mathbf{B}_t^\top \right)^{-\frac{1}{2}} \mathbf{B}_t$ % Orthogonalize $\mathbf{B}_t$

$\mathbf{A}_t \leftarrow \mathbf{B}_t \mathbf{X}^\top \left(\mathbf{X}\mathbf{X}^\top \right)^{-1} \mathbf{X}$ % Least squares regression $\mathbf{X} \rightarrow \mathbf{B}_t$

$\mathbf{A}_t \leftarrow \left(\mathbf{A}_t \mathbf{A}_t^\top \right)^{-\frac{1}{2}} \mathbf{A}_t$ % Orthogonalize $\mathbf{A}_t$

end for

Output: $\mathbf{A}_T \rightarrow \tilde{\mathbf{U}}^\top \mathbf{X}$, $\mathbf{B}_T \rightarrow \tilde{\mathbf{V}}^\top \mathbf{Y}$ are CCA projections as $T \rightarrow \infty$.

- This procedure converges linearly under mild conditions.
- [Lu & Foster 2014] used a similar procedure for linear CCA with high dimensional sparse inputs.

Approach II : Nolinear orthogonal iterations (NOI)

Choose a minibatch b of n samples at each step, and

Approach II : Nolinear orthogonal iterations (NOI)

Choose a minibatch b of n samples at each step, and

- Adaptively estimate covariance matrix for orthogonalization

$$\boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \leftarrow \rho \boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} + (1-\rho)\frac{N}{n}\sum_{i\in b} \tilde{\mathbf{g}}(\mathbf{y}_i)\tilde{\mathbf{g}}(\mathbf{y}_i)^\top$$

  - Time constant $\rho \in [0, 1)$. Update form is similar to that of momentum and widely used in subspace tracking.
  - Saving $\boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \in \mathbb{R}^{L\times L}$ costs little memory as L is usually small.

Approach II : Nolinear orthogonal iterations (NOI)

Choose a minibatch b of n samples at each step, and

- Adaptively estimate covariance matrix for orthogonalization

$$\mathbf{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \leftarrow \rho \mathbf{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} + (1-\rho)\frac{N}{n}\sum_{i \in b} \tilde{\mathbf{g}}(\mathbf{y}_i)\tilde{\mathbf{g}}(\mathbf{y}_i)^\top$$

  - Time constant $\rho \in [0, 1)$. Update form is similar to that of momentum and widely used in subspace tracking.
  - Saving $\mathbf{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \in \mathbb{R}^{L \times L}$ costs little memory as L is usually small.

- Replace exact linear regression with nonlinear least squares and take a gradient descent step on the minibatch

$$\min_{\mathbf{W}_{\tilde{\mathbf{f}}}} \frac{1}{n}\sum_{i \in b} \left\| \tilde{\mathbf{f}}(\mathbf{x}_i) - \mathbf{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}^{-\frac{1}{2}} \tilde{\mathbf{g}}(\mathbf{y}_i) \right\|^2$$

  - Ordinary DNN regression problem. No involved gradient.

Approach II : Nolinear orthogonal iterations (NOI)

Choose a minibatch b of n samples at each step, and

- Adaptively estimate covariance matrix for orthogonalization

$$\boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \leftarrow \rho \boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} + (1-\rho)\frac{N}{n}\sum_{i\in b} \tilde{\mathbf{g}}(\mathbf{y}_i)\tilde{\mathbf{g}}(\mathbf{y}_i)^\top$$

  - Time constant $\rho \in [0, 1)$. Update form is similar to that of momentum and widely used in subspace tracking.
  - Saving $\boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \in \mathbb{R}^{L\times L}$ costs little memory as L is usually small.

- Replace exact linear regression with nonlinear least squares and take a gradient descent step on the minibatch

$$\min_{\mathbf{W}_{\tilde{\mathbf{f}}}} \frac{1}{n}\sum_{i\in b} \left\| \tilde{\mathbf{f}}(\mathbf{x}_i) - \boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}^{-\frac{1}{2}} \tilde{\mathbf{g}}(\mathbf{y}_i) \right\|^2$$

  - Ordinary DNN regression problem. No involved gradient.

Each step feeds-forward and back-propagates only n samples!

Approach II : Nolinear orthogonal iterations (NOI)

Choose a minibatch b of n samples at each step, and

- Adaptively estimate covariance matrix for orthogonalization

$$\boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \leftarrow \rho \boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} + (1-\rho)\frac{N}{n}\sum_{i \in b} \tilde{\mathbf{g}}(\mathbf{y}_i)\tilde{\mathbf{g}}(\mathbf{y}_i)^\top$$

  - Time constant $\rho \in [0, 1)$. Update form is similar to that of momentum and widely used in subspace tracking.
  - Saving $\boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \in \mathbb{R}^{L \times L}$ costs little memory as L is usually small.

- Replace exact linear regression with nonlinear least squares and take a gradient descent step on the minibatch

$$\min_{\mathbf{W}_{\tilde{\mathbf{f}}}} \frac{1}{n}\sum_{i \in b} \left\| \tilde{\mathbf{f}}(\mathbf{x}_i) - \boldsymbol{\Sigma}_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}^{-\frac{1}{2}} \tilde{\mathbf{g}}(\mathbf{y}_i) \right\|^2$$

  - Ordinary DNN regression problem. No involved gradient.

Each step feeds-forward and back-propagates only n samples!

[Ma, Lu and Foster 2015] proposed a similar algorithm *AppGrad* for CCA with $\rho \equiv 0$.

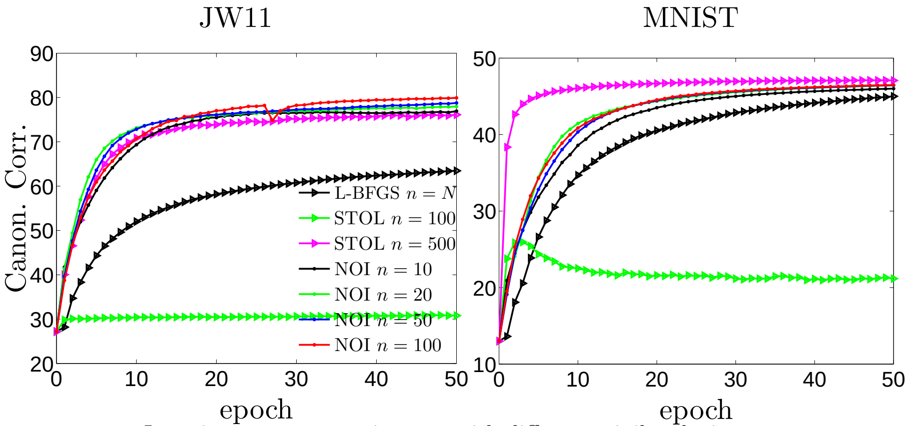
Experiments: Datasets

- Compare three optimizers on two real-world datasets
  - Limited-memory BFGS run with full-batch gradient
  - Stochastic optimization with large minibatches (STOL)
  - Nonlinear Orthogonal Iterations (NOI)
- Hyper-parameters (time constant ρ, minibatch size n, learning rate η) are tuned by grid search.
- STOL/NOI run for a maximum number of 50 epochs.

Statistics of two real-world datasets.

dataset	training/tuning/test	L	DNN architectures
JW11	30K/11K/9K	112	273-1800-1800-112 112-1200-1200-112
MNIST	50K/10K/10K	50	392-800-800-50 392-800-800-50

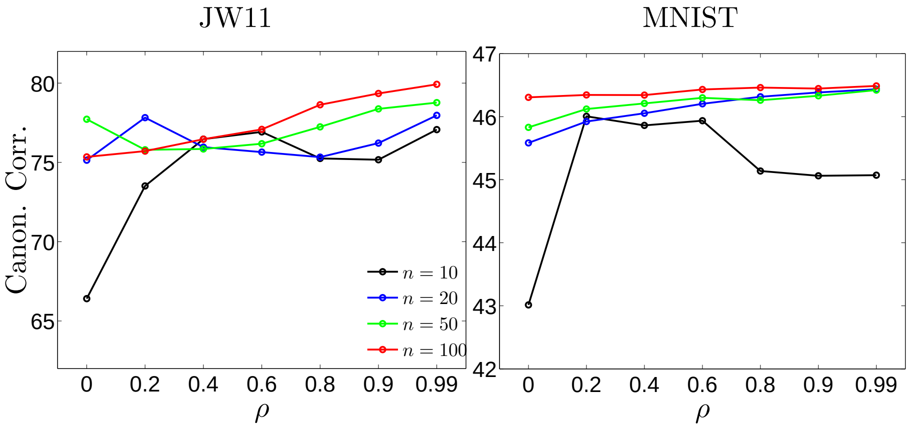
Experiments: Effect of minibatch size n



Learning curves on tuning sets with different minibatch size n.

- NOI works well with various small minibatch sizes.
- NOI gives steep improvement in the first few passes over the data.

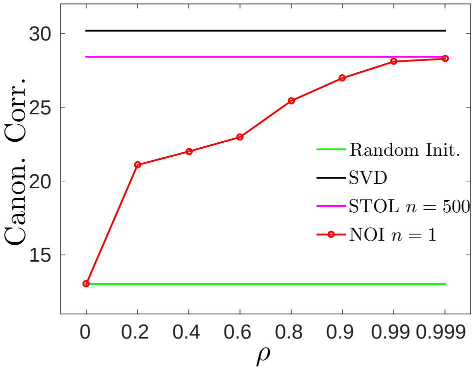
Experiments: Effect of time constant ρ



Total correlation achieved by NOI on tuning sets with different ρ.

- NOI works well for a wide range of ρ.
- Beneficial to use large ρ to incorporate the previous estimate of covariance for small n.

Experiments: Pure stochastic optimization



Total correlation achieved on MNIST training sets at different ρ for linear CCA.

- *AppGrad* [Ma, Lu and Foster 2015] used $n \sim \mathcal{O}(L)$ with $\rho = 0$.
- Pure stochastic optimization $n = 1$ works well with large ρ.

Conclusions

- We have developed NOI for training deep CCA with small minibatches, alleviating the memory cost.
- NOI performs competitively to previous batch and stochastic optimizers.

Conclusions

- We have developed NOI for training deep CCA with small minibatches, alleviating the memory cost.
- NOI performs competitively to previous batch and stochastic optimizers.
- Future directions:
  - Gradients of nonlinear least squares problems in NOI are not unbiased estimate of gradient of deep CCA objective.
  - Need to better understand the convergence properties of NOI.

Conclusions

- We have developed NOI for training deep CCA with small minibatches, alleviating the memory cost.
- NOI performs competitively to previous batch and stochastic optimizers.
- Future directions:
  - Gradients of nonlinear least squares problems in NOI are not unbiased estimate of gradient of deep CCA objective.
  - Need to better understand the convergence properties of NOI.

Thank you!

Bibliography I

Shotaro Akaho. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*, Osaka, Japan, 2001. Springer-Verlag.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In Sanjoy Dasgupta and David McAllester, editors, *Proc. of the 30th Int. Conf. Machine Learning (ICML 2013)*, pages 1247–1255, Atlanta, GA, June 16–21 2013.

Raman Arora, Andy Cotter, Karen Livescu, and Nati Srebro. Stochastic optimization for PCA and PLS. In *50th Annual Allerton Conference on Communication, Control, and Computing*, pages 861–868, Montcello, IL, October 1–5 2012.

Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, July 2002.

Gene H. Golub and Hongyuan Zha. *Linear Algebra for Signal Processing*, volume 69 of *The IMA Volumes in Mathematics and its Applications*, chapter The Canonical Correlations of Matrix Pairs and their Numerical Computation, pages 27–49. Springer-Verlag, 1995.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, December 1936.

P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *Int. J. Neural Syst.*, 10(5):365–377, October 2000.

Bibliography II

Zhuang Ma, Yichao Lu, and Dean Foster. Finding linear structure in large datasets with scalable canonical correlation analysis. In Francis Bach and David Blei, editors, *Proc. of the 32st Int. Conf. Machine Learning (ICML 2015)*, pages 169–178, Lille, France, July 7–9 2015.

Thomas Melzer, Michael Reiter, and Horst Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proc. of the 11th Int. Conf. Artificial Neural Networks (ICANN'01)*, pages 353–360, Vienna, Austria, August 21–25 2001.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP'15)*, Brisbane, Australia, April 19–24 2015.