Sparse representation, dictionary Learning, and deep neural networks: Their connections and new algorithms

Mostafa Sadeghi

Electrical Engineering Department
Sharif University of Technology
Tehran, Iran.

June 2018

Proximal Algorithms

- Efficient **first order** algorithms. Suitable for **nonsmooth**, **constrained**, **large-scale** problems.

Proximal Algorithms

- Efficient **first order** algorithms. Suitable for **nonsmooth**, **constrained**, **large-scale** problems.

Proximal mapping [Parikh and Boyd, 2014]

$g:\ \mathrm{dom}_g \to \mathbb{R} \cup \{+\infty\}$: proper, lower-semicontinuous

$$\boxed{\mathbf{prox}_g(\mathbf{u}) = \underset{\mathbf{x}\in\mathrm{dom}_g}{\mathrm{argmin}}\ \ g(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2}$$

Proximal Algorithms

- Efficient **first order** algorithms. Suitable for **nonsmooth**, **constrained**, **large-scale** problems.

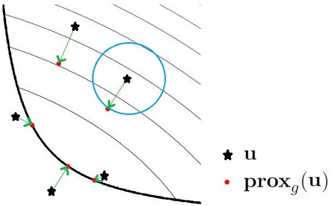
Proximal mapping [Parikh and Boyd, 2014]

$g:\ \mathrm{dom}_g \to \mathbb{R} \cup \{+\infty\}$: proper, lower-semicontinuous

$$\boxed{\mathbf{prox}_g(\mathbf{u}) = \underset{\mathbf{x}\in \mathrm{dom}_g}{\mathrm{argmin}}\ \ g(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2}$$

$$\underset{\mathbf{x}\in \mathrm{dom}_g}{\mathrm{argmin}}\ \ g(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{x}-\mathbf{u}\|_2 \leq \tau$$

$$\boxed{\mathbf{prox}_{\lambda g}(\mathbf{u}) \simeq \mathbf{u} - \lambda \nabla g(\mathbf{u})}$$



Forward-backward Splitting

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) + g(\mathbf{x})$$

- f : dom$_f \to \mathbb{R}$ smooth (convex/non-convex)
- g : dom$_g \to \mathbb{R} \cup \{+\infty\}$ non-smooth (convex/non-convex)

Forward-backward Splitting

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) + g(\mathbf{x})$$

- $f:\ \mathrm{dom}_f \to \mathbb{R}$ — smooth (convex/non-convex)
- $g:\ \mathrm{dom}_g \to \mathbb{R}\cup\{+\infty\}$ — non-smooth (convex/non-convex)

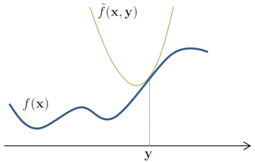
Descent lemma

$f:\ \mathrm{dom}_f \to \mathbb{R}$, smooth and L-gradient Lipschitz*, $\mu \in (0, 1/L]$

$$\forall \mathbf{x},\mathbf{y} \in \mathrm{dom} f: \quad f(\mathbf{x}) \le \tilde{f}(\mathbf{x},\mathbf{y}) \triangleq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \frac{1}{2\mu}\|\mathbf{x}-\mathbf{y}\|_2^2$$

*$\forall x, y \in \mathrm{dom} f$:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \le L\|\mathbf{x}-\mathbf{y}\|_2$$



Forward-backward Splitting

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) + g(\mathbf{x})$$

$$\boxed{\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} \; \tilde{f}(\mathbf{x}, \mathbf{x}_k) + g(\mathbf{x})}$$

Forward-backward Splitting

$$\min_{\mathbf{x}\in\mathbb{R}^n} \ f(\mathbf{x}) + g(\mathbf{x})$$

$$\boxed{\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \ \tilde{f}(\mathbf{x}, \mathbf{x}_k) + g(\mathbf{x})}$$

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{x} - \big(\mathbf{x}_k - \mu\nabla f(\mathbf{x}_k)\big)\|_2^2 + \mu \cdot g(\mathbf{x})$$

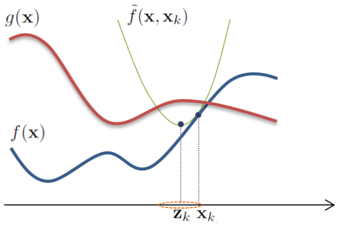
Forward-backward Splitting

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) + g(\mathbf{x})$$

$$\boxed{\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \; \tilde{f}(\mathbf{x}, \mathbf{x}_k) + g(\mathbf{x})}$$

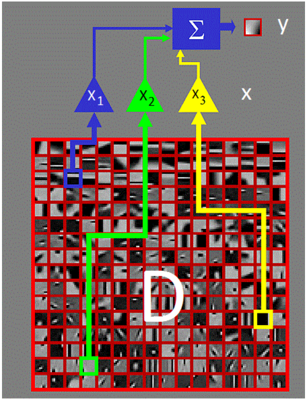
$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \; \frac{1}{2}\|\mathbf{x} - \big(\mathbf{x}_k - \mu\nabla f(\mathbf{x}_k)\big)\|_2^2 + \mu \cdot g(\mathbf{x})$$

1. Forward step: $\mathbf{z}_k = \mathbf{x}_k - \mu\nabla f(\mathbf{x}_k)$
2. Backward step: $\mathbf{x}_{k+1} = \mathbf{prox}_{\mu\cdot g}(\mathbf{z}_k)$



Sparse representation

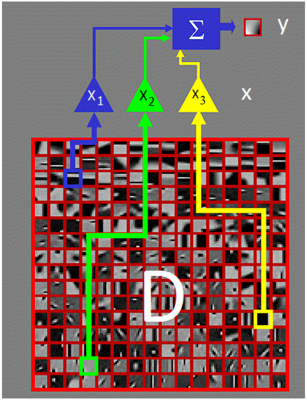
$\mathbf{y} \approx x_1\mathbf{d}_1 + x_2\mathbf{d}_2 + \ldots + x_n\mathbf{d}_m = \mathbf{Dx}$ most x_i's are zero



* Adopted from M. Elad's slides

Sparse representation

$\mathbf{y} \approx x_1\mathbf{d}_1 + x_2\mathbf{d}_2 + \ldots + x_n\mathbf{d}_m = \mathbf{Dx}$ most x_i's are zero



* Adopted from M. Elad's slides

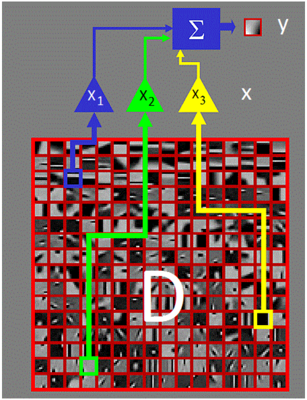
- Signal restoration:

$$\boxed{\mathbf{y} = \mathbf{Hx} + \mathbf{e}}$$

De-noising ($\mathbf{H}$ = identity), inpainting ($\mathbf{H}$ = random rows of identity), de-bluring ($\mathbf{H}$ = blurring matrix), super resolution ($\mathbf{H}$ = down sampling matrix), ...

Sparse representation

$\mathbf{y} \approx x_1\mathbf{d}_1 + x_2\mathbf{d}_2 + \ldots + x_n\mathbf{d}_m = \mathbf{D}\mathbf{x}$ most x_i's are zero



* Adopted from M. Elad's slides

- Signal restoration:

$$\boxed{\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}}$$

De-noising ($\mathbf{H}$ = identity), inpainting ($\mathbf{H}$ = random rows of identity), de-bluring ($\mathbf{H}$ = blurring matrix), super resolution ($\mathbf{H}$ = down sampling matrix), ...

$$\mathbf{x} \simeq \mathbf{D}\mathbf{a}, \quad \mathbf{a} : \text{sparse}$$

$$\boxed{\min_{\mathbf{x},\mathbf{a}} \; \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \alpha\|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 + \beta\|\mathbf{a}\|_1}$$

Sparse recovery algorithms

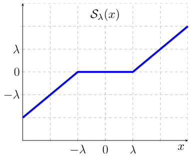
- Thresholding based algorithms

$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1}$$

- Thresholding based algorithms

$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1}$$

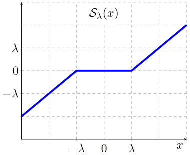
$$\boxed{\mathbf{x}^{k+1} = \mathcal{S}_{\mu_k \lambda}(\mathbf{x}^k - \mu_k(\mathbf{D}\mathbf{x}_k - \mathbf{y}))}$$



- Thresholding based algorithms

$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1}$$

$$\boxed{\mathbf{x}^{k+1} = \mathcal{S}_{\mu_k\lambda}(\mathbf{x}^k - \mu_k(\mathbf{D}\mathbf{x}_k - \mathbf{y}))}$$

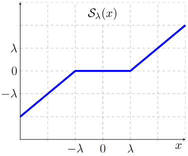


$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_0}$$

- Thresholding based algorithms

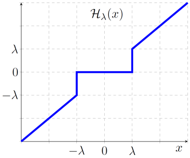
$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1}$$

$$\boxed{\mathbf{x}^{k+1} = \mathcal{S}_{\mu_k\lambda}(\mathbf{x}^k - \mu_k(\mathbf{D}\mathbf{x}_k - \mathbf{y}))}$$



$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_0}$$

$$\boxed{\mathbf{x}^{k+1} = \mathcal{H}_{\mu_k\lambda}(\mathbf{x}^k - \mu_k(\mathbf{D}\mathbf{x}_k - \mathbf{y}))}$$

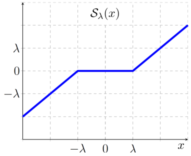


$$\mu_k \in (0, 1/\sigma_{\max}(\mathbf{D}))$$

- Thresholding based algorithms

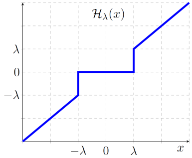
$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2+\lambda\|\mathbf{x}\|_1}$$

$$\boxed{\mathbf{x}^{k+1}=\mathcal{S}_{\mu_k\lambda}(\mathbf{x}^k-\mu_k(\mathbf{D}\mathbf{x}_k-\mathbf{y}))}$$



$$\boxed{\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2+\lambda\|\mathbf{x}\|_0}$$

$$\boxed{\mathbf{x}^{k+1}=\mathcal{H}_{\mu_k\lambda}(\mathbf{x}^k-\mu_k(\mathbf{D}\mathbf{x}_k-\mathbf{y}))}$$



$$\mu_k \in (0, 1/\sigma_{\max}(\mathbf{D}))$$

Examples: IST [Daubechies et al., 2004], GPSR [Figueiredo et al., 2007], IHT [Blumensath and Davies, 2009], AMP [Donoho et al., 2009], EMGMAMP [Vila et al., 2013], NESTA [Becker et al., 2009], SCAD [Gasso et al., 2009].

Sparse recovery algorithms

- ℓ_0 norm approximation. Approximate ℓ_0 norm with a smooth function.

  Smoothed L0 (SL0) [Mohimani et al., 2009], SCSA [Malek-Mohammadi et al., 2016]

- ℓ_0 norm approximation. Approximate ℓ_0 norm with a smooth function.

  Smoothed L0 (SL0) [Mohimani et al., 2009], SCSA [Malek-Mohammadi et al., 2016]

$$\boxed{\|\mathbf{x}\|_\sigma = n - \sum_{i=1}^{n} f_\sigma(x_i)}$$

$$f_\sigma(x) = \exp(-\frac{x^2}{\sigma^2})$$

Sparse recovery algorithms

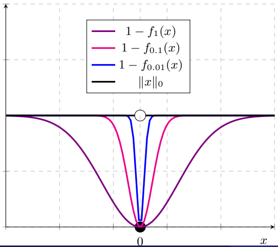
- ℓ_0 norm approximation. Approximate ℓ_0 norm with a smooth function.

  Smoothed L0 (SL0) [Mohimani et al., 2009], SCSA [Malek-Mohammadi et al., 2016]

$$\boxed{\|\mathbf{x}\|_\sigma = n - \sum_{i=1}^{n} f_\sigma(x_i)}$$

$$f_\sigma(x) = \exp(-\frac{x^2}{\sigma^2})$$

$$\boxed{\text{When } \sigma \to 0: \quad \|\mathbf{x}\|_\sigma \to \|\mathbf{x}\|_0}$$



- ℓ_0 norm approximation. Approximate ℓ_0 norm with a smooth function.

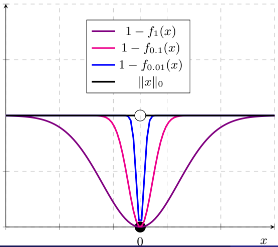
  Smoothed L0 (SL0) [Mohimani et al., 2009], SCSA [Malek-Mohammadi et al., 2016]

$$\boxed{\|\mathbf{x}\|_\sigma = n - \sum_{i=1}^{n} f_\sigma(x_i)}$$

$$\boxed{\min_{\mathbf{x}} \|\mathbf{x}\|_\sigma \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}}$$

$$f_\sigma(x) = \exp(-\frac{x^2}{\sigma^2})$$

$$\boxed{\text{When } \sigma \to 0: \ \|\mathbf{x}\|_\sigma \to \|\mathbf{x}\|_0}$$



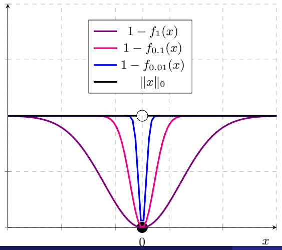
- ℓ_0 norm approximation. Approximate ℓ_0 norm with a smooth function.

  Smoothed L0 (SL0) [Mohimani et al., 2009], SCSA [Malek-Mohammadi et al., 2016]

$$\boxed{\|\mathbf{x}\|_\sigma = n - \sum_{i=1}^{n} f_\sigma(x_i)}$$

$$f_\sigma(x) = \exp(-\frac{x^2}{\sigma^2})$$

$$\boxed{\text{When } \sigma \to 0: \quad \|\mathbf{x}\|_\sigma \to \|\mathbf{x}\|_0}$$



$$\boxed{\min_{\mathbf{x}} \|\mathbf{x}\|_\sigma \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}}$$

$$\begin{cases} k = 0 \\ \mathbf{x}^0 = \mathbf{D}^\dagger \mathbf{y} \\ \text{For } i = 1, 2, \ldots \\ \begin{cases} \text{For } j = 1, 2, \ldots \\ \mathbf{x}^{k+1} = \mathbf{x}^k - \mu_{\sigma_i} \nabla \|\mathbf{x}_k\|_{\sigma_i} \\ \mathbf{x}^{k+1} = \mathbf{x}^{k+1} - \mathbf{D}^\dagger(\mathbf{D}\mathbf{x}_{k+1} - \mathbf{y}) \\ k \leftarrow k + 1 \\ \text{End} \end{cases} \\ \sigma_{i+1} = \sigma_i \cdot c \quad (0 < c < 1) \\ \text{End} \end{cases}$$

1 Iterative Sparsification-Projection (ISP)

[1] M. Sadeghi, M. Babaie-Zadeh, "Iterative Sparsification-Projection: Fast and Robust Sparse Signal Approximation", *IEEE Trans. Sig. Proc.*, vol. 64, no. 21, pp. 5536-5548, November 2016.

[1] Iterative Sparsification-Projection (ISP)

- Revisiting the SL0 algorithm:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \underbrace{\sum_{i=1}^{n}\left(1-\exp(-\frac{x_i^2}{\sigma^2})\right)}_{f_\sigma(\mathbf{x})} \quad \text{s.t.} \quad \|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 \leq \epsilon$$

[1] M. Sadeghi, M. Babaie-Zadeh, "Iterative Sparsification-Projection: Fast and Robust Sparse Signal Approximation", *IEEE Trans. Sig. Proc.*, vol. 64, no. 21, pp. 5536-5548, November 2016.

Iterative Sparsification-Projection (ISP)[1]

- Revisiting the SL0 algorithm:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \underbrace{\sum_{i=1}^{n}\left(1-\exp(-\frac{x_i^2}{\sigma^2})\right)}_{f_\sigma(\mathbf{x})} \quad \text{s.t.} \quad \|\mathbf{y}-\mathbf{Dx}\|_2 \le \epsilon$$

Lemma (Lipschitz constant)

The function f_σ (defined above) is gradient Lipschitz with constant $L = \frac{2}{\sigma^2}$.

[1] M. Sadeghi, M. Babaie-Zadeh, "Iterative Sparsification-Projection: Fast and Robust Sparse Signal Approximation", *IEEE Trans. Sig. Proc.*, vol. 64, no. 21, pp. 5536-5548, November 2016.

1 Iterative Sparsification-Projection (ISP)

- Revisiting the SL0 algorithm:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \underbrace{\sum_{i=1}^{n}\left(1-\exp\left(-\frac{x_i^2}{\sigma^2}\right)\right)}_{f_\sigma(\mathbf{x})} \quad \text{s.t.} \quad \|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 \le \epsilon$$

Lemma (Lipschitz constant)

The function f_σ (defined above) is gradient Lipschitz with constant $L=\frac{2}{\sigma^2}$.

$$\min_{\mathbf{x}\in\mathbb{R}^n} \underbrace{\sum_{i=1}^{n}\left(1-\exp\left(-\frac{x_i^2}{\sigma^2}\right)\right)}_{f(\mathbf{x})} + \underbrace{\delta_{\mathcal{C}}(\mathbf{x})}_{g(\mathbf{x})} \rightarrow \boxed{\mathbf{x}_{k+1}=\mathbf{prox}_{\delta_{\mathcal{C}}}\left(\mathbf{x}_k-\mu_\sigma\nabla f(\mathbf{x}_k)\right)}$$

$\delta_{\mathcal{C}}(\mathbf{x})=0$ if $\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2\le\epsilon$ and ∞ otherwise.

[1] M. Sadeghi, M. Babaie-Zadeh, "Iterative Sparsification-Projection: Fast and Robust Sparse Signal Approximation", *IEEE Trans. Sig. Proc.*, vol. 64, no. 21, pp. 5536-5548, November 2016.

Theorem (SL0 convergence)

Let $\{\mathbf{x}_k\}$ be the sequence generated by the SL0 algorithm for a fixed σ. Then:

- This sequence is bounded and convergent, which means that any accumulation point $\mathbf{x}^*$ of $\{\mathbf{x}_k\}$ is a critical point, and
- the sequence of objective values, i.e., $\{f_\sigma(\mathbf{x}_k) + \delta_{\mathcal{C}}(\mathbf{x}_k)\}_{k\geq 0}$, is non-increasing.

Iterative Sparsification-Projection

Theorem (SL0 convergence)

Let $\{\mathbf{x}_k\}$ be the sequence generated by the SL0 algorithm for a fixed σ. Then:

- This sequence is bounded and convergent, which means that any accumulation point $\mathbf{x}^*$ of $\{\mathbf{x}_k\}$ is a critical point, and
- the sequence of objective values, i.e., $\{f_\sigma(\mathbf{x}_k) + \delta_{\mathcal{C}}(\mathbf{x}_k)\}_{k\geq 0}$, is non-increasing.

Corollary

If $\mu_\sigma \in (0, 2/L]$, or equivalently, if $\mu \in (0, 1]$, then the SL0 algorithm converges.

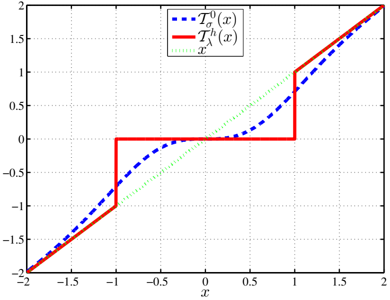
- ISP   Motivation: $\begin{cases} \mathbf{z}_k = \mathbf{x}_k - \mu_\sigma \nabla f(\mathbf{x}_k) \triangleq \mathcal{T}_\sigma^0(\mathbf{x}_k) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{z}_k) \end{cases}$

Iterative Sparsification-Projection

- ISP Motivation: $\begin{cases} \mathbf{z}_k = \mathbf{x}_k - \mu_\sigma \nabla f(\mathbf{x}_k) \triangleq \mathcal{T}_\sigma^0(\mathbf{x}_k) \\ \mathbf{x}_{k+1} = \mathcal{P}_\mathcal{C}(\mathbf{z}_k) \end{cases}$

$$\mathcal{T}_\sigma^0(x) = x \cdot (1 - \exp(-\frac{x^2}{\sigma^2}))$$

$$(\mu_\sigma = \frac{\sigma^2}{2})$$



☞ Gradient descent interpretation of proximal mapping!

$\|\mathbf{x}\|_\sigma$ smoothed of $\|\mathbf{x}\|_0$ $\Rightarrow$ $\mathbf{x} - \mu_\sigma \nabla \|\mathbf{x}\|_\sigma$ behaves like $\mathbf{prox}_{\mu_\sigma}(\mathbf{x}) =$ hard-thr

Iterative Sparsification-Projection

☞ Gradient descent interpretation of proximal mapping!

$\|\mathbf{x}\|_\sigma$ smoothed of $\|\mathbf{x}\|_0$ $\Rightarrow$ $\mathbf{x} - \mu_\sigma \nabla \|\mathbf{x}\|_\sigma$ behaves like $\mathbf{prox}_{\mu_\sigma}(\mathbf{x})$ = hard-thr

Algorithm 1 SL0

Require: $\mathbf{y}$, $\mathbf{D}$, σ_0, $\sigma_{\min}$, $c > 0$, μ, I
Initialization: $\mathbf{x} = \mathbf{D}^\dagger \mathbf{y}$, $\sigma = \sigma_0$
while $\sigma > \sigma_{\min}$ **do**
  for $i = 1, 2, \ldots, I$ **do**
    $\boxed{\tilde{\mathbf{x}} = \mathbf{x} - \mu \cdot \sigma^2 \nabla \|\mathbf{x}\|_\sigma}$
    $\mathbf{x} = \tilde{\mathbf{x}} - \mathbf{D}^\dagger (\mathbf{D}\tilde{\mathbf{x}} - \mathbf{y})$
  end for
  $\sigma \leftarrow c \cdot \sigma$
end while
Output: $\mathbf{x}$

Algorithm 2 ISP

Require: $\mathbf{y}$, $\mathbf{D}$, τ_0, $\tau_{\min}$, $c > 0$, I
Initialization: $\mathbf{x} = \mathbf{D}^\dagger \mathbf{y}$, $\tau = \tau_0$
while $\tau > \tau_{\min}$ **do**
  for $i = 1, 2, \ldots, I$ **do**
    $\boxed{\tilde{\mathbf{x}} = \mathcal{T}_\tau(\mathbf{x})}$
    $\mathbf{x} = \tilde{\mathbf{x}} - \mathbf{D}^\dagger (\mathbf{D}\tilde{\mathbf{x}} - \mathbf{y})$
  end for
  $\tau \leftarrow c \cdot \tau$
end while
Output: $\mathbf{x}$

☞ Gradient descent interpretation of proximal mapping!

$\|\mathbf{x}\|_\sigma$ smoothed of $\|\mathbf{x}\|_0$ $\Rightarrow$ $\mathbf{x} - \mu_\sigma \nabla \|\mathbf{x}\|_\sigma$ behaves like $\mathbf{prox}_{\mu_\sigma}(\mathbf{x})$ = hard-thr

Algorithm 1 SL0

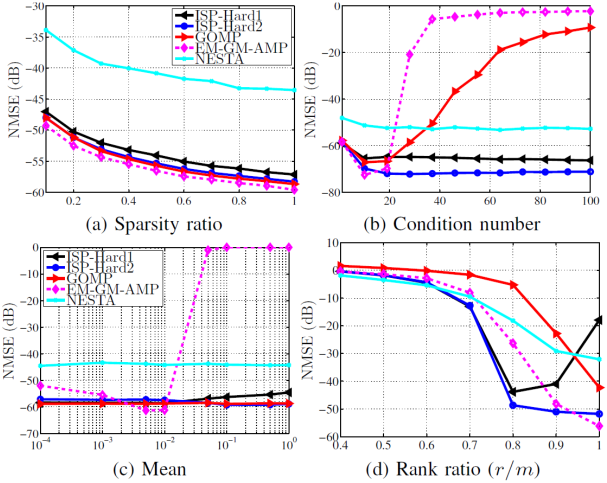
Require: $\mathbf{y}$, $\mathbf{D}$, σ_0, $\sigma_{\min}$, $c > 0$, μ, I
Initialization: $\mathbf{x} = \mathbf{D}^\dagger \mathbf{y}$, $\sigma = \sigma_0$
while $\sigma > \sigma_{\min}$ **do**
  for $i = 1, 2, \ldots, I$ **do**
    $\boxed{\tilde{\mathbf{x}} = \mathbf{x} - \mu \cdot \sigma^2 \nabla \|\mathbf{x}\|_\sigma}$
    $\mathbf{x} = \tilde{\mathbf{x}} - \mathbf{D}^\dagger(\mathbf{D}\tilde{\mathbf{x}} - \mathbf{y})$
  end for
  $\sigma \leftarrow c \cdot \sigma$
end while
Output: $\mathbf{x}$

Algorithm 2 ISP

Require: $\mathbf{y}$, $\mathbf{D}$, τ_0, $\tau_{\min}$, $c > 0$, I
Initialization: $\mathbf{x} = \mathbf{D}^\dagger \mathbf{y}$, $\tau = \tau_0$
while $\tau > \tau_{\min}$ **do**
  for $i = 1, 2, \ldots, I$ **do**
    $\boxed{\tilde{\mathbf{x}} = \mathcal{T}_\tau(\mathbf{x})}$
    $\mathbf{x} = \tilde{\mathbf{x}} - \mathbf{D}^\dagger(\mathbf{D}\tilde{\mathbf{x}} - \mathbf{y})$
  end for
  $\tau \leftarrow c \cdot \tau$
end while
Output: $\mathbf{x}$

☞ $\mathcal{T}_\tau$ a sparsifying function $= \begin{cases} x - \mu_\sigma \nabla f_\sigma(x) & (\text{ISP} - \ell_0, \text{ISP} - \ell_1) \\ \mathbf{prox}_{\mu_\sigma}(\mathbf{x}) & (\text{ISP-Hard}, \text{ISP-Soft}) \end{cases}$

Recovery performance. Synthetic data: $\mathbf{y}_{m\times 1} = \mathbf{D}_{m\times n}\mathbf{x}_{n\times 1} + \mathbf{e}_{m\times 1}$. Bernoulli-Gaussian sparse signal. Gaussian $\mathbf{D}$ and noise. $m = 400$, $n = 1000$. Different measurement matrices. $\mathbf{D}$: sparse, ill-conditioned, non-zero mean, low-rank.



Iterative Proximal Projection

$$\boxed{\min_{\mathbf{x}} \ J(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon}$$

☞ J : non-smooth, non-convex

[2] F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "Sparse signal recovery via iterative proximal projection", *IEEE Trans. Sig. Proc.*, vol. 66, no. 4, pp. 879–894, 2018.

$$\boxed{\min_{\mathbf{x}} \; J(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon}$$

☞ J : non-smooth, non-convex

2

IPP

Main idea

$$\min_{\mathbf{x},\mathbf{z}} \; J(\mathbf{z}) + \delta_{\mathcal{C}}(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{z} = \mathbf{x}$$

2 F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "Sparse signal recovery via iterative proximal projection", *IEEE Trans. Sig. Proc.*, vol. 66, no. 4, pp. 879–894, 2018.

$$\boxed{\min_{\mathbf{x}} \; J(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon}$$

☞ J : non-smooth, non-convex

[2]

IPP

Main idea

$$\min_{\mathbf{x},\mathbf{z}} \; J(\mathbf{z}) + \delta_{\mathcal{C}}(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{z} = \mathbf{x}$$

$$\min_{\mathbf{x},\mathbf{z}} \; J(\mathbf{z}) + \delta_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{z}\|_2^2$$

---

[2] F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "Sparse signal recovery via iterative proximal projection", *IEEE Trans. Sig. Proc.*, vol. 66, no. 4, pp. 879–894, 2018.

Iterative Proximal Projection

$$\boxed{\min_{\mathbf{x}} \; J(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon}$$

☞ J : non-smooth, non-convex

2

IPP

Main idea

$$\min_{\mathbf{x},\mathbf{z}} \; J(\mathbf{z}) + \delta_{\mathcal{C}}(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{z} = \mathbf{x}$$

$$\min_{\mathbf{x},\mathbf{z}} \; J(\mathbf{z}) + \delta_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{z}\|_2^2$$

$$\begin{cases} \mathbf{z}_{k+1} = \operatorname{argmin}_{\mathbf{z}} \; \alpha J(\mathbf{z}) + \frac{1}{2}\|\mathbf{z} - \mathbf{x}_k\|_2^2 \\ \mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}} \; \delta_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{z}_{k+1}\|_2^2 \end{cases}$$

2 F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "Sparse signal recovery via iterative proximal projection", *IEEE Trans. Sig. Proc.*, vol. 66, no. 4, pp. 879–894, 2018.

Iterative Proximal Projection

$$\boxed{\min_{\mathbf{x}} \; J(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon}$$

☞ J : non-smooth, non-convex

2

IPP

Main idea

$$\min_{\mathbf{x},\mathbf{z}} \; J(\mathbf{z}) + \delta_{\mathcal{C}}(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{z} = \mathbf{x}$$

$$\min_{\mathbf{x},\mathbf{z}} \; J(\mathbf{z}) + \delta_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{z}\|_2^2$$

$$\begin{cases} \mathbf{z}_{k+1} = \mathrm{argmin}_{\mathbf{z}} \;\; \alpha J(\mathbf{z}) + \frac{1}{2}\|\mathbf{z} - \mathbf{x}_k\|_2^2 \\ \mathbf{x}_{k+1} = \mathrm{argmin}_{\mathbf{x}} \;\; \delta_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{z}_{k+1}\|_2^2 \end{cases} \qquad \begin{cases} \tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\text{prox}_{\alpha . J}(\tilde{\mathbf{x}}_k)\Big) \end{cases}$$

2 F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "Sparse signal recovery via iterative proximal projection", *IEEE Trans. Sig. Proc.*, vol. 66, no. 4, pp. 879–894, 2018.

Iterative Proximal Projection

- Difficult to prove convergence. An alternative algorithm with convergence proof!

Iterative Proximal Projection

- Difficult to prove convergence. An alternative algorithm with convergence proof!

IPP

An approximate solver:

$$\begin{cases} \tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}\left(\text{prox}_{\alpha \cdot J}(\tilde{\mathbf{x}}_k)\right) \end{cases}$$

Iterative Proximal Projection

- Difficult to prove convergence. An alternative algorithm with convergence proof!

IPP

An approximate solver:

$$\begin{cases} \tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}\left(\text{prox}_{\alpha \cdot J}(\tilde{\mathbf{x}}_k)\right) \end{cases} \rightarrow \begin{cases} \tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{z}_{k+1} = \text{prox}_{\mu_z \cdot \alpha J}(\mathbf{z}_k + \mu_z(\tilde{\mathbf{x}}_k - \mathbf{z}_k)) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{x}_k + \mu_x(\mathbf{z}_{k+1} - \mathbf{x}_k)) \end{cases}$$

☞ $0 < \mu_x, \mu_z < 1$. If $\mu_x, \mu_z \rightarrow 1$ the two algorithms coincide!

Iterative Proximal Projection

- Difficult to prove convergence. An alternative algorithm with convergence proof!

IPP

An approximate solver:

$$\begin{cases} \tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}\left(\operatorname{prox}_{\alpha \cdot J}(\tilde{\mathbf{x}}_k)\right) \end{cases} \rightarrow \begin{cases} \tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{z}_{k+1} = \operatorname{prox}_{\mu_z \cdot \alpha J}(\mathbf{z}_k + \mu_z(\tilde{\mathbf{x}}_k - \mathbf{z}_k)) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{x}_k + \mu_x(\mathbf{z}_{k+1} - \mathbf{x}_k)) \end{cases}$$

☞ $0 < \mu_x, \mu_z < 1$. If $\mu_x, \mu_z \rightarrow 1$ the two algorithms coincide!

Algorithm Iterative Proximal Projection (IPP)

```
Inputs: y, A, ε, α_i, α_f, τ, 0 < c < 1, w, 0 < μ_x, μ_z < 1
Initialization: k = 0, x_0 = z_0 = A†y, α = α_i
while α > α_f do
    while ||x_k − x_{k−1}||_2 > τ do
        x̃_k = x_k + w · (x_k − x_{k−1})
        z_{k+1} = prox_{μ_z·α·J}(z_k + μ_z(x̃_k − z_k))
        x_{k+1} = P_{A_ε}(x_k + μ_x(z_{k+1} − x_k))
        k → k + 1
    end while
    α ← c · α
end while
Output: x_k
```

Iterative Proximal Projection

Theorem (IPP convergence)

In the IPP algorithm, assume that $0 \leq w < \frac{1}{\max(\mu_x,\mu_z)} - 1$. The sequence $\left\{\mathbf{u}_k \triangleq (\mathbf{x}_k, \mathbf{z}_k)\right\}_{k=0}^{\infty}$ generated by IPP for each value of α (the inner-loop iterations) converges to a critical point, $\mathbf{u}^*$, of the cost function. Furthermore, if the cost function satisfies the Kurdyka-Łojasiewicz (KL) property [Bolt et al., 2014] with $\psi(t) = c \cdot t^{1-\theta}$ for some $t > 0$ and $\theta \in [0,1)$, then:

- If $\theta = 0$ then the sequence $\{\mathbf{u}_k\}_{k\geq 0}$ converges in a finite number of steps.
- If $\theta \in (0, 1/2]$ then there exist $d > 0$ and $\tau \in [0,1)$ such that $\|\mathbf{u}_k - \mathbf{u}^*\|_2 \leq d \cdot \tau^k$.
- If $\theta \in (1/2, 1)$ then there exist $d > 0$ such that $\|\mathbf{u}_k - \mathbf{u}^*\|_2 \leq d \cdot k^{\frac{\theta-1}{2\theta-1}}$.

Simulations

Block based compressed image recovery: 50% overlapping blocks, Gaussian measurement, $\delta =$ sampling rate.

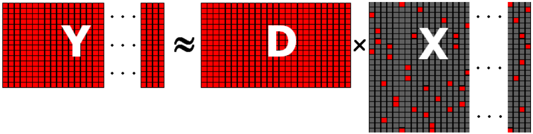
	$\delta = 0.1$			$\delta = 0.2$		
	House	Barbara	Monarch	House	Barbara	Monarch
ℓ_q	25.13	22.99	19.87	28.56	25.30	22.71
GOMP	25.00	22.38	19.03	26.68	23.94	21.23
SCSA	25.11	22.96	19.88	**28.62**	25.26	22.64
EMGMAMP	25.02	22.94	19.69	27.96	25.06	22.22
IPP ($w = 0$)	**25.31**	**23.08**	**20.46**	27.72	**25.36**	**22.95**
IPP ($w = 0.85$)	**25.57**	**23.38**	**20.67**	**28.65**	**25.63**	**23.45**

	$\delta = 0.3$			$\delta = 0.4$		
	House	Barbara	Monarch	House	Barbara	Monarch
ℓ_q	31.17	27.38	24.97	34.15	30.17	27.65
GOMP	30.85	27.14	24.43	33.59	29.69	27.10
SCSA	**31.35**	27.37	**24.99**	**34.49**	30.23	**27.79**
EMGMAMP	30.91	27.15	24.65	33.64	29.72	27.06
IPP ($w = 0$)	31.11	**27.61**	24.73	33.56	**30.41**	27.70
IPP ($w = 0.85$)	**32.50**	**28.17**	**25.55**	**35.01**	**30.92**	**28.47**

- Learn a sparsifying dictionary from training data: $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_L]$.

Dictionary Learning Problem

$$\min_{\mathbf{D},\mathbf{X}} \; \sum_{i=1}^{L} \frac{1}{2}\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 = \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \mathbf{D} \in \mathcal{D}, \; \mathbf{X} \in \mathcal{X}$$

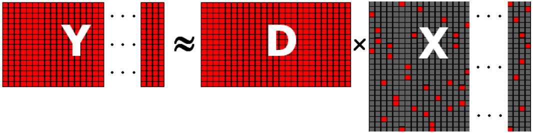


Dictionary Learning

- Learn a sparsifying dictionary from training data: $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_L]$.

Dictionary Learning Problem

$$\min_{\mathbf{D},\mathbf{X}} \ \sum_{i=1}^{L} \frac{1}{2}\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 = \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \mathbf{D} \in \mathcal{D}, \ \mathbf{X} \in \mathcal{X}$$



Alternating minimization

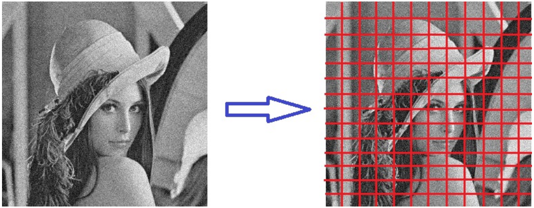
- Start with $(\mathbf{D}^{(0)}, \mathbf{X}^{(0)})$. Alternate between:
  1. Sparse representation: $\mathbf{X}^{(k+1)} = \operatorname{argmin}_{\mathbf{X}\in\mathcal{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{D}^{(k)}\mathbf{X}\|_F^2$
     - ☞ OMP, IST, SL0, ...
  2. Dictionary update: $\mathbf{D}^{(k+1)} = \operatorname{argmin}_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}^{(k+1)}\|_F^2$
     - ☞ MOD, KSVD, ...

Applications of dictionary learning

- Signal/image restoration and enhancement
  - Image denoising [Elad et al., 2006]:

$$\mathbf{y} = \mathbf{x} + \mathbf{e}$$

Learn the dictionary from the noisy image itself!



$$\hat{\mathbf{x}} = \underset{\mathbf{x},\mathbf{D},\{\alpha_{ij}\}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \lambda \sum_{i,j} \|\mathbf{R}_{ij}\mathbf{x} - \mathbf{D}\alpha_{ij}\|_2^2 \quad \text{s.t.} \quad \|\alpha_{ij}\|_0 \leq \tau$$

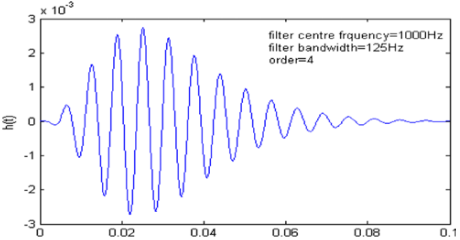
Applications of dictionary learning

- Speech denoising [Sigg et al., 2012; Jafari et al., 2011]
- Parameter dictionary learning [Yaghoobi et al., 2009; Ataee et al., 2010]:

Learning structured atoms:

- Example. Gammatone filters have shown similarities with the human auditory system:

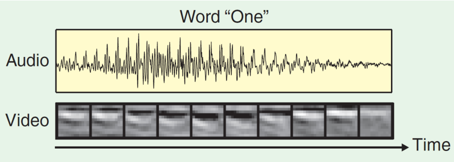
$$\boxed{g(t) = at^{n-1}e^{-2\pi bBt}\cos(2\pi f_c t)}$$



https://www.researchgate.net/publication/264837246
_Voice_biometric_feature_using_Gammatone_filterbank_and_ICA/figures?lo=1

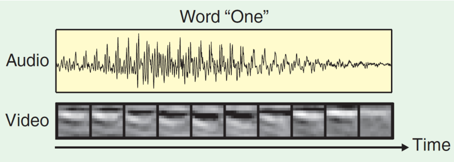
Applications of dictionary learning

- Multi-modal dictionary learning [Monaci et al., 2007; Zhuang et al., 2013]
  - Learning multi-modal atoms to describe underlying generating cause, speaker localization, and so on

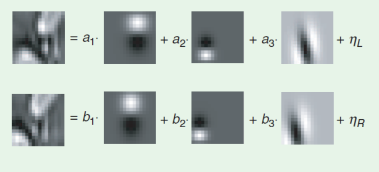


Applications of dictionary learning

- Multi-modal dictionary learning [Monaci et al., 2007; Zhuang et al., 2013]
  - Learning multi-modal atoms to describe underlying generating cause, speaker localization, and so on



- Stereo image representation [Tosic et al., 2011]
  - Efficient image representation to perform different vision task like camera pose estimation



Applications of dictionary learning

- Supervised dictionary learning [Mairal et al., 2010; Zhang et al., 2010]

$$\min_{\mathbf{D},\mathbf{X}} \quad \underbrace{\frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2}_{\text{representation power}} \quad + \quad \underbrace{\lambda\|\mathbf{T}-\mathbf{W}\mathbf{X}\|_F^2}_{\text{discrimination power}} \qquad \text{s.t.} \quad \|\mathbf{X}\|_0 \le \tau$$

  - $\mathbf{T}$: label matrix
  - $\mathbf{W}$: linear classifier

Learning Low Coherence Dictionaries

Mutual Coherence

For a dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$, its mutual coherence is defined as

$$\mu(\mathbf{D}) \triangleq \max_{i \neq j} \frac{|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|}{\|\mathbf{d}_i\|_2 \cdot \|\mathbf{d}_j\|_2}$$

The Welch bound:

$$\mu \geq \sqrt{\frac{m-n}{n(m-1)}}$$

Learning Low Coherence Dictionaries

Mutual Coherence

For a dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$, its mutual coherence is defined as

$$\mu(\mathbf{D}) \triangleq \max_{i \neq j} \frac{|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|}{\|\mathbf{d}_i\|_2 \cdot \|\mathbf{d}_j\|_2}$$

The Welch bound:

$$\mu \geq \sqrt{\frac{m-n}{n(m-1)}}$$

Low MC dictionaries

A dictionary with low MC is desired for sparse recovery algorithms.

☞ Guaranteed sparse recovery as long as $\|\mathbf{x}\|_0 \leq \frac{1}{2}(1 + \frac{1}{\mu(\mathbf{D})})$

Learning Low Coherence Dictionaries

- Regularized: $\boxed{\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_F^2}$

$$\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_F^2 = \sum_{i\neq j}|\langle \mathbf{d}_i, \mathbf{d}_j\rangle|^2 + \sum_i (\langle \mathbf{d}_i, \mathbf{d}_i\rangle - 1)^2$$

Bounded Self Coherence (BSC) [Sigg *et. al*, 2012], Gradient Projection (GP) [Bao *et. al*, 2015]

Learning Low Coherence Dictionaries

- Regularized: $\boxed{\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\|\mathbf{Y}-\mathbf{DX}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_F^2}$

$$\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_F^2 = \sum_{i\neq j}|\langle \mathbf{d}_i, \mathbf{d}_j\rangle|^2 + \sum_i (\langle \mathbf{d}_i, \mathbf{d}_i\rangle - 1)^2$$

Bounded Self Coherence (BSC) [Sigg *et. al*, 2012], Gradient Projection (GP) [Bao *et. al*, 2015]

- Constrained: $\boxed{\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\|\mathbf{Y}-\mathbf{DX}\|_F^2 \quad \text{s.t.} \quad \mu(\mathbf{D}) \leq \mu_0}$

Iterative Projection Rotation (IPR) [Barchiesi and Plumbley, 2013]

Learning Low Coherence Dictionaries

- Regularized: $\boxed{\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_F^2}$

$$\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_F^2 = \sum_{i\neq j}|\langle \mathbf{d}_i, \mathbf{d}_j\rangle|^2 + \sum_i (\langle \mathbf{d}_i, \mathbf{d}_i\rangle - 1)^2$$

Bounded Self Coherence (BSC) [Sigg *et. al*, 2012], Gradient Projection (GP) [Bao *et. al*, 2015]

- Constrained: $\boxed{\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \mu(\mathbf{D}) \leq \mu_0}$

Iterative Projection Rotation (IPR) [Barchiesi and Plumbley, 2013]

☞ Uses a two-step procedure: decorrelation + rotation

Mutual coherence-based DL

Low coherence DL

$$\mu(\mathbf{D}) = \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \quad \|\mathbf{A}\|_\infty \triangleq \max_{i,j} |a_{ij}|$$

Proposed problems[a]:

$$\begin{cases} \min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \\ \min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \le \mu_0 \end{cases}$$

---

[a] M. Sadeghi and M. Babaie-Zadeh, "Learning low-coherence dictionaries for sparse representation", *Sig. Proc.*, 2018 (submitted).

Mutual coherence-based DL

Low coherence DL

$$\mu(\mathbf{D}) = \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \quad \|\mathbf{A}\|_\infty \triangleq \max_{i,j} |a_{ij}|$$

Proposed problems[a]:

$$\begin{cases} \min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \\ \min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \leq \mu_0 \end{cases}$$

[a] M. Sadeghi and M. Babaie-Zadeh, "Learning low-coherence dictionaries for sparse representation", *Sig. Proc.*, 2018 (submitted).

Main idea

- Introduce $\mathbf{G} = \mathbf{D}^T\mathbf{D} - \mathbf{I}$. Use penalty method + proximal algorithm!

$$\boxed{\min_{\mathbf{D}\in\mathcal{D},\mathbf{G}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 + \frac{1}{2\alpha}\|\mathbf{G} - \mathbf{D}^T\mathbf{D} + \mathbf{I}\|_F^2 + \lambda g(\mathbf{G})}$$

Mutual coherence-based DL

Low coherence DL

$$\mu(\mathbf{D}) = \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \quad \|\mathbf{A}\|_\infty \triangleq \max_{i,j} |a_{ij}|$$

Proposed problems[a]:

$$\begin{cases} \min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \\ \min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty \leq \mu_0 \end{cases}$$

[a] M. Sadeghi and M. Babaie-Zadeh, "Learning low-coherence dictionaries for sparse representation", *Sig. Proc.*, 2018 (submitted).

Main idea

- Introduce $\mathbf{G} = \mathbf{D}^T\mathbf{D} - \mathbf{I}$. Use penalty method + proximal algorithm!

$$\boxed{\min_{\mathbf{D}\in\mathcal{D},\mathbf{G}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \frac{1}{2\alpha}\|\mathbf{G} - \mathbf{D}^T\mathbf{D} + \mathbf{I}\|_F^2 + \lambda g(\mathbf{G})}$$

$$g(\mathbf{G}) = \begin{cases} \|\mathbf{G}\|_\infty & \text{(regularized)} \\ \delta_{\mathcal{C}}(\mathbf{G}), \quad \mathcal{C} \triangleq \{\mathbf{G} \mid \|\mathbf{G}\|_\infty \leq \mu_0\} & \text{(constrained)} \end{cases}$$

Mutual coherence-based DL

Updating $\mathbf{G}$

Let g denote the function $\eta\|.\|_\infty : \mathbb{R}^{N\times N} \to \mathbb{R}$. The proximal mapping of g is given by

$$\text{prox}_g(\mathbf{U}) = \mathbf{U} - P_{\frac{\eta}{1}}(\mathbf{U})$$

where, $P_{\frac{\eta}{1}}(.) : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ is the projection onto the ℓ_1 norm-ball of radius η.

Mutual coherence-based DL

Updating $\mathbf{G}$

Let g denote the function $\eta\|.\|_\infty : \mathbb{R}^{N\times N} \to \mathbb{R}$. The proximal mapping of g is given by

$$\text{prox}_g(\mathbf{U}) = \mathbf{U} - P_{\overset{\eta}{1}}(\mathbf{U})$$

where, $P_{\overset{\eta}{1}}(.) : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ is the projection onto the ℓ_1 norm-ball of radius η.

Updating $\mathbf{D}$

The gradient of $\frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \frac{1}{2\alpha}\|\mathbf{G} - \mathbf{D}^T\mathbf{D} + \mathbf{I}\|_F^2$ with respect to $\mathbf{D}$ is Lipschitz continuous over $\mathcal{D}$ with constant

$$L = \|\mathbf{X}\mathbf{X}^T\| + \frac{6N + 2\|\mathbf{G}\|_F}{2\alpha}$$

Mutual coherence-based DL

Regularized Incoherent DL (RINC-DL) and Constrained Incoherent DL (CINC-DL)

Algorithm 1 RINC-DL

Require: $\mathbf{Y}$, $\mathbf{D}_0$, τ, λ, c, L_2, ϵ, I, J
Initialization: $\mathbf{D} = \mathbf{D}_0$, $\mathbf{G} = \mathbf{0}$
while stopping criterion for DL not met **do**
  1. Sparse approximation: $\mathbf{X} = \mathrm{SD}(\mathbf{Y}, \mathbf{D}, \tau)$
  2. Dictionary update:
  $L_1 = \|\mathbf{X}^T\mathbf{X}\|$
  $\alpha = 3 \cdot \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty$
  $i = 1$
  while $i \leq I$ and $\|\mathbf{G} - \mathbf{D}^T\mathbf{D}\|_F > \epsilon$ **do**
    $\mu_d = 1/(L_1 + \alpha^{-1}L_2)$
    for $j = 1, 2, \cdots, J$ **do**
      $\mathbf{G} = \mathbf{D}^T\mathbf{D} - P_{\mathcal{B}_1^{\lambda \cdot \alpha}}(\mathbf{D}^T\mathbf{D} - \mathbf{I})$
      $\boxed{\mathbf{G} = \mathbf{I} + P_{\mathcal{B}_\infty^{\mu_0}}(\mathbf{D}^T\mathbf{D} - \mathbf{I})}$ CINC-DL
      $\mathbf{D} = P_{\mathcal{D}}(\mathbf{D} - \mu_d \nabla f(\mathbf{D}))$
    end for
    $\alpha \leftarrow c \cdot \alpha$
    $i \leftarrow i + 1$
  end while
end while
Output: $\mathbf{D}$, $\mathbf{X}$

Mutual coherence-based DL

Regularized Incoherent DL (RINC-DL) and Constrained Incoherent DL (CINC-DL)

Algorithm 1 RINC-DL

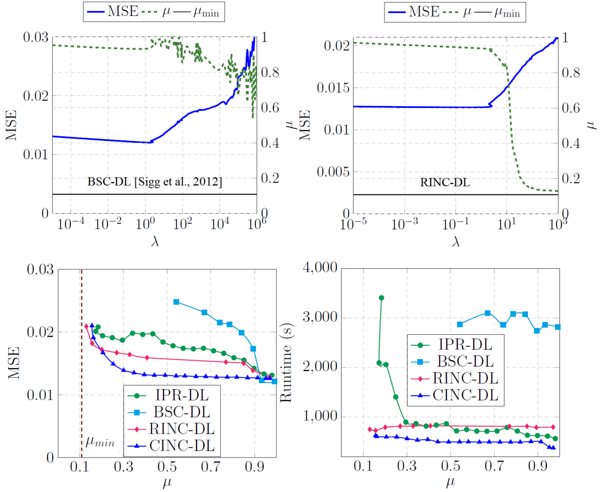
```
Require: Y, D_0, τ, λ, c, L_2, ε, I, J
Initialization: D = D_0, G = 0
while stopping criterion for DL not met do
    1. Sparse approximation: X = SD(Y, D, τ)
    2. Dictionary update:
    L_1 = ||X^T X||
    α = 3 · ||D^T D − I||_∞
    i = 1
    while i ≤ I and ||G − D^T D||_F > ε do
        μ_d = 1/(L_1 + α^{-1} L_2)
        for j = 1, 2, ···, J do
            G = D^T D − P_{B_1^{λ·α}}(D^T D − I)
            [ G = I + P_{B_∞^{μ_0}}(D^T D − I) ]   CINC-DL
            D = P_𝒟(D − μ_d ∇f(D))
        end for
        α ← c · α
        i ← i + 1
    end while
end while
Output: D, X
```

Algorithm	Complexity
IPR-DL	$\mathcal{O}(nNM + MN^2 + 3n^2N + 2n^3 + 2N^3)$
CINC-DL	$\mathcal{O}(nNM + nN^2)$
RINC-DL	$\mathcal{O}(nNM + nN^2)$

- n : signal dimension
- N : number of atoms
- M : number of training signals

Simulations

Low-coherence DL for 8×8 image blocks: $M = 50,000$, $\mathbf{D}_0 = \mathrm{DCT}_{64\times 256}$ ($\mu_{\min} = 0.1085$), $s = 10$.

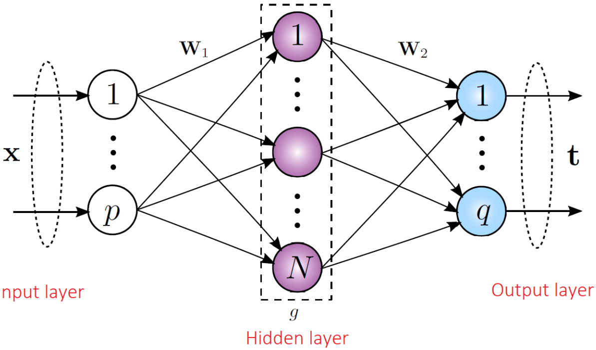


(a) MSE (b) Runtime

Artificial Neural Networks

Estimate the mapping function $\mathbf{t} = f(\mathbf{x})$ given some training data $\{(\mathbf{x}_i, \mathbf{t}_i)\}_i$

Single layer NN: $\boxed{\hat{\mathbf{t}} = \mathbf{W}_2 \cdot g(\mathbf{W}_1 \cdot \mathbf{x})}$ g: non-linear activation function

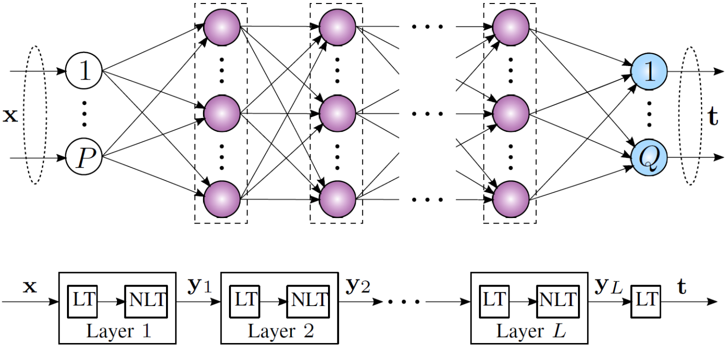


It can approximate any continuous function under mild conditions (universal approximation)

- To approximate complex functions, increase the number of hidden nodes
- Leads to very wide networks!

Solution: Deepen the network instead of widening it!

Multilayer NN:



Training algorithms:

$$\boxed{\min_{\mathbf{W}_1,\mathbf{W}_2} \frac{1}{M}\sum_{i=1}^{M}\|\mathbf{t}_i - \mathbf{W}_2\cdot g(\mathbf{W}_1\cdot\mathbf{x}_i)\|_2^2}$$

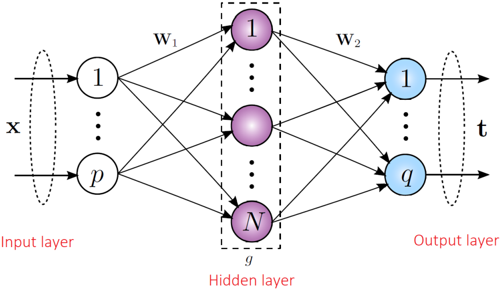
- Stochastic Gradient Descent (SGD) – Backpropagation
- Gradient-free training using ADMM [Taylor et al. 2016]
- Proximal backpropagation [Frerix et al. 2017]
- Extreme learning machine (ELM) [Huang et al. 2006]:
  - ☞ Set $\mathbf{W}_1$ randomly
  - ☞ Find $\mathbf{W}_2$ using least squares: $\min_{\mathbf{W}_2} \frac{1}{M}\sum_{i=1}^{M}\|\mathbf{t}_i - \mathbf{W}_2\cdot g(\mathbf{W}_1\cdot\mathbf{x}_i)\|_2^2$

Dictionary Learning and Neural Networks

Dictionary learning has some similarities with single-layer NN:

- DL: $\boxed{\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \mathbf{X} \text{ is sparse}}$
- SNN:: $\boxed{\min_{\mathbf{W}_1,\mathbf{W}_2} \|\mathbf{T} - \mathbf{W}_2 \cdot g(\mathbf{W}_1 \cdot \mathbf{X})\|_F^2}$

☞ Both $\mathbf{X}$ and $g(\mathbf{W}_1 \cdot \mathbf{X})$ with g being ReLU: $g(x) = \max(x, 0)$ are non-linear and sparse representation of data



More explicit connections between DL (dictionary learning) and DL (deep learning):

- V. Papyan, Y. Romano, J. Sulam, and M. Elad, "**Theoretical Foundations of Deep Learning via Sparse Representations**," to appear in *IEEE Signal Processing Magazine*.
- J. Sulam, V. Papyan, Y. Romano, and M. Elad, "**Multi-Layer Convolutional Sparse Modeling: Pursuit and Dictionary Learning**," to appear in *IEEE Trans. on Signal Processing*.

From multi-layer convolutional sparse coding (CSC) to convolutional neural networks (CNNs)

Artificial Neural Networks

Questions to address in a multilayer NN:

- How to choose number of layers in a network?
- How to choose number of nodes in each layer?
- How to guarantee that increase in size results in better (non-increasing) optimized cost for training data?
- How to design with appropriate regularization of network parameters to avoid over-fitting to training data?
- Can we use random weight matrices to keep the number of parameters to learn in balance?

Progression Learning Network

Definition (Progression Property)

A non-linear $g(.)$ function holds the progression property (PP) if there are two known linear transformations $\mathbf{V} \in \mathbb{R}^{M\times N}$ and $\mathbf{U} \in \mathbb{R}^{N\times M}$ such that $\forall \boldsymbol{\gamma} \in \mathbb{R}^{N}$[a]:

$$\mathbf{U}\mathbf{g}(\mathbf{V}\boldsymbol{\gamma}) = \boldsymbol{\gamma}$$

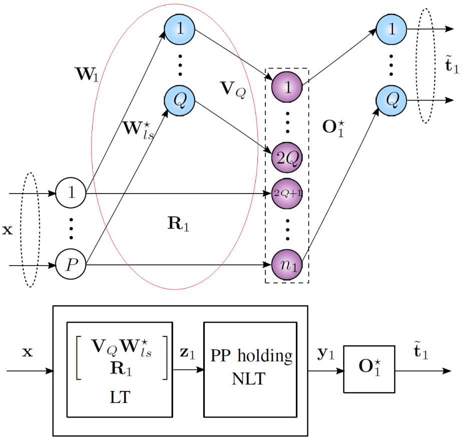
[a] S. Chatterjee, A. M. Javid, M. Sadeghi, P. P. Mitra and M. Skoglund, "Progressive learning for systematic design of large neural networks", *IEEE Trans. Neural Networks and Learning Systems*, 2017 (submitted).

- Example: The rectified linear unit (ReLU) function [Glorot et al. 2011]

$$g(\gamma) = \max(\gamma, 0) = \begin{cases} \gamma, & \text{if } \gamma \geq 0 \\ 0, & \text{if } \gamma < 0. \end{cases}$$

If $\mathbf{V} \triangleq \mathbf{V}_N = [\mathbf{I}_N, \ -\mathbf{I}_N]^T \in \mathbb{R}^{2N\times N}$ and $\mathbf{U} \triangleq \mathbf{U}_N = [\mathbf{I}_N \ \ -\mathbf{I}_N] \in \mathbb{R}^{N\times 2N}$ then ReLU holds PP. Here $\mathbf{I}_N$ denotes identity matrix of size N $(M = 2N)$.

Single layer PLN



- $\tilde{\mathbf{t}} = \mathbf{O}_1\, \mathbf{g}(\mathbf{W}_1\mathbf{x})$
- $\mathbf{R}_1$ is a random matrix

Single layer PLN

- $\mathbf{W}_{ls}^{\star} = \underset{\mathbf{W}_{ls}}{\arg\min} \sum_j \|\mathbf{t}^{(j)} - \mathbf{W}_{ls}\mathbf{x}^{(j)}\|_p^p \text{ s.t. } \|\mathbf{W}_{ls}\|_q^q \leq \epsilon$
- $\mathbf{O}_1^{\star} = \underset{\mathbf{O}_1}{\arg\min} \sum_j \|\mathbf{t}^{(j)} - \mathbf{O}_1\mathbf{y}_1^{(j)}\|_p^p \text{ such that } \|\mathbf{O}_1\|_q^q \leq \alpha\|\mathbf{U}_Q\|_q^q,$

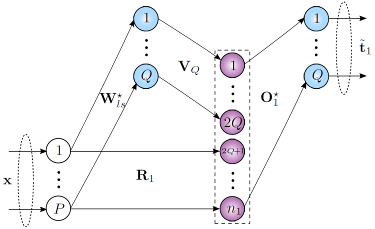
$$C_{ls}^{\star} \triangleq C(\mathbf{W}_{ls}^{\star}) = \sum_j \|\mathbf{t}^{(j)} - \mathbf{W}_{ls}^{\star}\mathbf{x}^{(j)}\|_p^p$$

$$C_1^{\star} = C(\mathbf{O}_1^{\star}) = \sum_j \|\mathbf{t}^{(j)} - \mathbf{O}_1^{\star}\mathbf{y}_1^{(j)}\|_p^p = \sum_j \|\mathbf{t}^{(j)} - \mathbf{O}_1^{\star}\,\mathbf{g}(\mathbf{W}_1\mathbf{x}^{(j)})\|_p^p$$

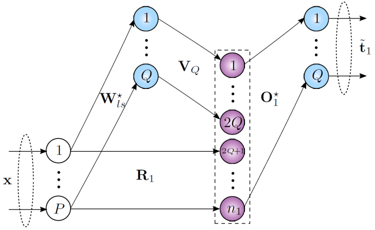
☞ Relation between optimal linear system and single layer PLN costs: $C_1^{\star} \leq C_{ls}^{\star}$. At the equality condition: $\mathbf{O}_1^{\star} = [\mathbf{U}_Q\ \mathbf{0}]$

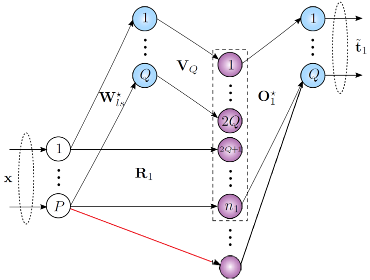
☞ Adding nodes to the layer: $C_1^{\star}(n_1 + \Delta) \leq C_1^{\star}(n_1)$

Multilayer layer (deep) PLN

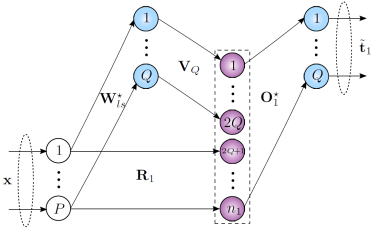


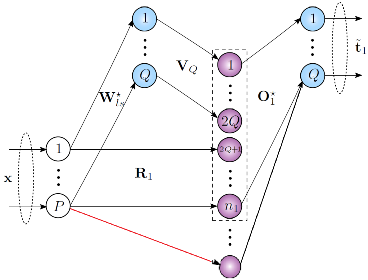
Multilayer layer (deep) PLN

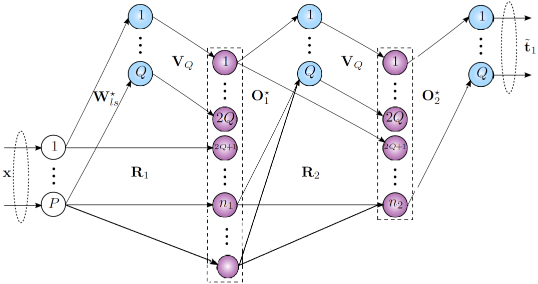




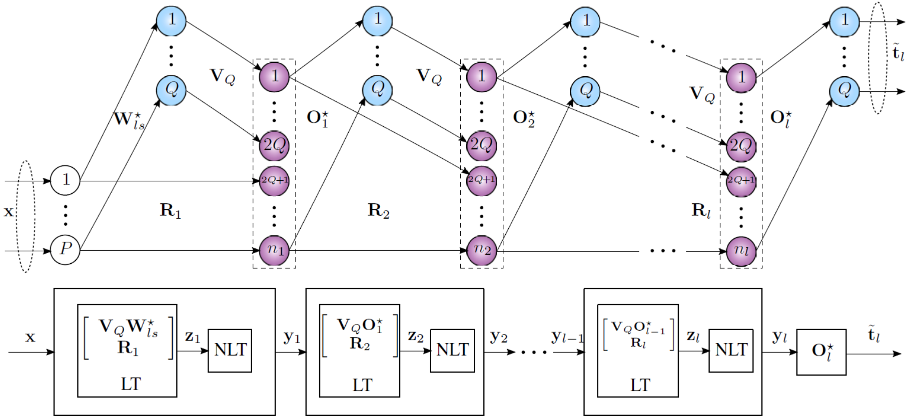
Multilayer layer (deep) PLN







Multilayer layer (deep) PLN



Multilayer layer (deep) PLN

- $\mathbf{O}_l^\star = \underset{\mathbf{O}_l}{\arg\min} \sum_j \|\mathbf{t}^{(j)} - \mathbf{O}_l \mathbf{y}_l^{(j)}\|_p^p$ such that $\|\mathbf{O}_l\|_q^q \leq \alpha \|\mathbf{U}_Q\|_q^q$,
- $C_l^\star = C(\mathbf{O}_l^\star) = \sum_j \|\mathbf{t}^{(j)} - \mathbf{O}_l^\star \mathbf{y}_l^{(j)}\|_p^p$

Proposition (Small approximation error)

Using PP and under the technical condition $\forall l$, $\mathbf{O}_l^\star \neq [\mathbf{U}_Q \, \mathbf{0}]$ *where* $\mathbf{0}$ *denotes a zero matrix of size* $Q \times (n_l - 2Q)$, *the optimized cost is monotonically decreasing with increase in number of layers, that is* $C_l^\star < C_{l-1}^\star$. *For a large number of layers, that means when* $l \to \infty$, *we have* $C_l^\star \leq \kappa$ *where* κ *is an arbitrarily small non-negative real scalar.*

☞ If we increase Δ nodes (random nodes) in the l'th layer then we have $C_l^\star(n_l + \Delta) \leq C_l^\star(n_l)$

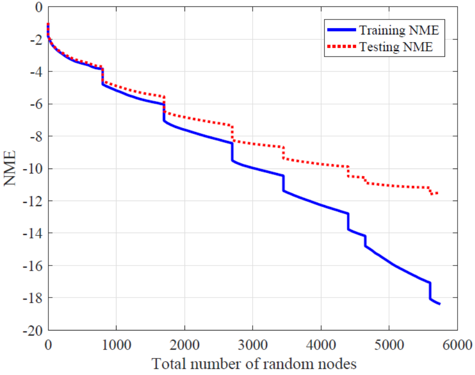
Simulation results

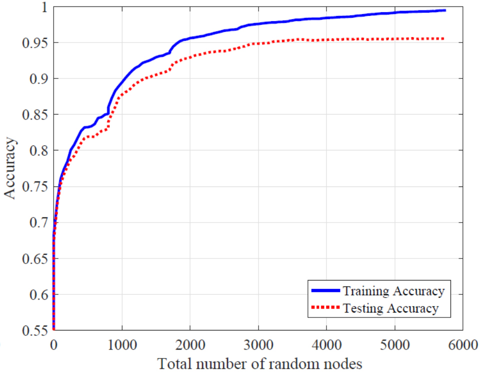
Classification:

Dataset	Regularized LS				Regularized ELM				PLN			
	Training NME	Testing NME	Test Accuracy	Training Time(s)	Training NME	Testing NME	Test Accuracy	Training Time(s)	Training NME	Testing NME	Test Accuracy	Training Time(s)
Vowel	-1.06	-0.81	28.1 ± 0.0	0.0035	-6.083	-1.49	53.8 ± 1.7	0.0549	**-72.54**	**-2.21**	**60.2** ± 2.4	1.2049
Extended YaleB	-7.51	-4.34	96.9 ± 0.6	0.0194	-12.75	-6.39	**97.8** ± 0.5	0.3908	**-49.97**	**-12.0**	97.7 ± 0.5	2.5776
AR	-3.82	-1.82	96.1 ± 0.6	0.0297	-9.019	-2.10	97.2 ± 0.7	0.5150	**-35.53**	**-7.69**	**97.6** ± 0.6	4.0691
Satimage	-2.82	-2.73	68.1 ± 0.0	0.0173	-7.614	-5.22	84.6 ± 0.5	0.8291	**-11.73**	**-7.92**	**89.9** ± 0.5	1.4825
Scene15	-8.68	-5.03	99.1 ± 0.2	0.6409	-7.821	-5.78	97.6 ± 0.3	2.7224	**-42.94**	**-14.7**	**99.1** ± 0.3	4.1209
Caltech101	-3.22	-1.29	66.3 ± 0.6	1.1756	-4.784	-1.21	63.4 ± 0.8	8.1560	**-14.66**	**-4.13**	**76.1** ± 0.8	5.3712
Letter	-1.00	-0.99	55.0 ± 0.8	0.0518	-9.217	-6.29	95.7 ± 0.2	20.987	**-18.60**	**-11.5**	**95.7** ± 0.2	12.926
NORB	-2.47	-1.54	80.4 ± 0.0	1.7879	**-15.97**	-6.77	**89.8** ± 0.5	23.207	-13.39	**-6.90**	86.1 ± 0.2	10.507
Shuttle	-6.17	-6.31	89.2 ± 0.0	0.1332	-18.31	-12.2	99.6 ± 0.1	1.8940	**-26.26**	**-25.0**	**99.8** ± 0.1	4.6345
MNIST	-4.07	-4.04	85.3 ± 0.0	0.8122	-9.092	-8.46	**96.9** ± 0.1	27.298	**-11.42**	**-10.9**	95.7 ± 0.1	14.181
CIFAR-10	-1.33	-1.33	50.3 ± 0.0	10.753	-2.004	-2.01	60.3 ± 0.3	53.842				
CIFAR-100	-0.20	-0.13	14.9 ± 0.0	12.883								

* The vowel database is for vowel recognition task (a speech recognition application) and all other databases are for image classification (computer vision applications).

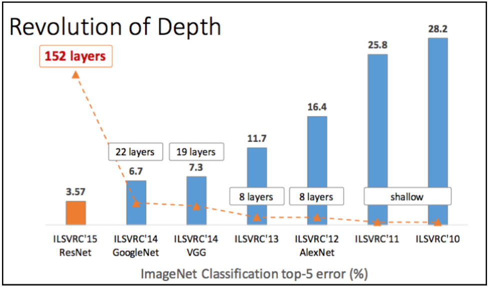
NME and accuracy versus number of nodes for the “Letter” dataset with 8-layer PLN:





Deep networks create deep trouble!

- Deep neural networks have many layers and nodes, with hundreds of millions of parameters $\rightarrow$ hundreds of megabytes for storage
- Needs more storage and computational resources
- Limiting their application in real-time tasks, and smart phones/ wearable devices



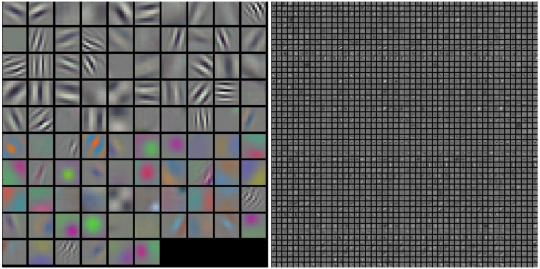
Background

Solutions [see Cheng et al., 2018]:

- Parameter pruning and sharing: Reducing redundant parameters that are not sensitive to the performance, using e.g. pruning weak connections, network quantization
- Low-rank factorization: Using matrix/tensor decomposition to estimate the informative parameters
- Transferred/compact convolutional filters: Designing special structural convolutional filters to save parameters

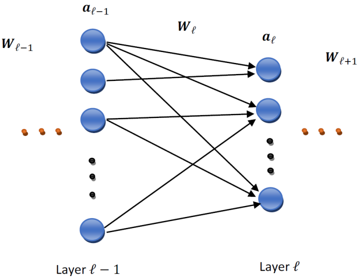
Motivation:

Weight vectors and filters corresponding to each node in a neural network exhibit some structure. So, they can be written as sparse linear combinations of e.g. DCT atoms.



Typical-looking filters on the first CONV layer (left), and the 2nd CONV layer (right) of a trained AlexNet.

http://cs231n.github.io/understanding-cnn/



Let $\mathbf{w}_\ell$ be a row of $\mathbf{W}_\ell$. Then, $\boxed{\mathbf{w}_\ell = \mathbf{s}_\ell \boldsymbol{\Phi} = \sum_i s_\ell^i \cdot \phi_i}$ and $\mathbf{s}_\ell$ is sparse.

Using sparse representation

$$\boxed{\mathbf{W}_\ell = \mathbf{S}_\ell \mathbf{\Phi}}$$

- $\mathbf{\Phi}$ is a complete (i.e., square matrix) like DCT for images or Gabor dictionary for speech data
- $\mathbf{S}_\ell$ is a matrix with sparse rows

Advantages:

1. Low memory consumption, as $\mathbf{S}_\ell$'s are sparse and the basis $\mathbf{\Phi}$ is shared among all the layers
2. Low computational complexity due to the sparseness of $\mathbf{S}_\ell$'s and that the multiplications with $\mathbf{\Phi}$ can be done very efficiently for particular transforms like DCT and Fourier
3. Preventing overfitting
4. Efficient training: SGD + projection

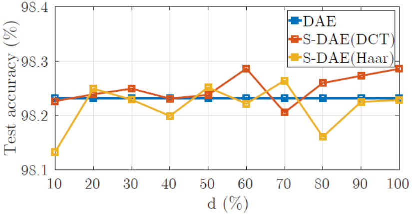
Another sparse structure:

$$\boxed{\mathbf{W}_\ell = \mathbf{S}_\ell^1 \cdot \mathbf{S}_\ell^2}$$

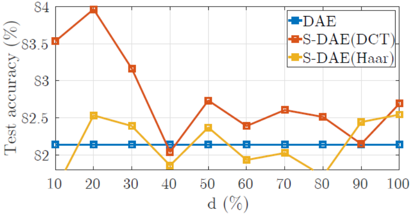
- Both factors $\mathbf{S}_\ell^1$ and $\mathbf{S}_\ell^2$ are sparse matrices
- Sparsity is global not row-wise

Again, training is easy: SGD (via backpropagation) + projection

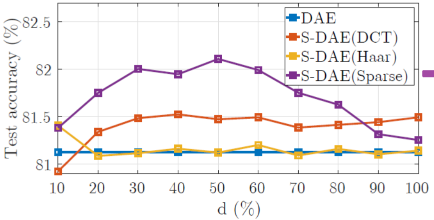
Test accuracy vs the percentage of non-zeros in each row of $\mathbf{S}$ for a 3-layer structure: $\boxed{\mathbf{W}_\ell = \mathbf{S}_\ell \boldsymbol{\Phi}}$



(a) MNIST dataset



(b) NORB dataset

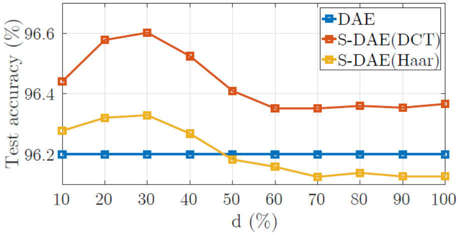


(c) SVHN dataset

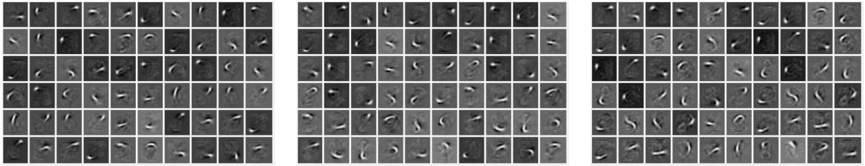
$\boxed{\mathbf{W}_\ell = \mathbf{S}_\ell^1 \cdot \mathbf{S}_\ell^2}$

Simulation results

Overfitting effect: MNIST($50,000 \to 10,000$ training samples)



Receptive fields:



(a) KLD method (b) S-KLD (Haar) method (c) S-KLD (DCT) method

Conclusions

- New algorithms were proposed for sparse recovery and dictionary learning problems based on penalty and ADMM methods combined with proximal algorithms
- The proposed sparse recovery algorithms gave new insights into some previous algorithms like SL0
- Inspired by a progression property, we develop progressive neural networks to learn architecture of neural networks
- Structured weight matrices were proposed using sparse representation to save memory and computation in deep networks

Thank You!