Distributed Compressive Sensing: A Deep Learning Approach

Sai Subramanyam Thoota

February 8, 2020

SPC Lab, Department of ECE
Indian Institute of Science

Introduction

- Compressive Sensing (CS): Solve for $\mathbf{x} \in \mathbb{R}^{N \times 1}$ from an underdetermined system of linear equations

$$\mathbf{y} = \mathbf{\Phi x}$$

where

  - $\mathbf{y} \in \mathbb{R}^{M \times 1}$ is the known measured vector, $M < N$
  - $\mathbf{\Phi}$ is a random measurement matrix
- To uniquely recover $\mathbf{x}$, it must be sparse in a given basis $\mathbf{\Psi} \in \mathbb{R}^{N \times N_1}$
  - Complete: $N = N_1$
  - Overcomplete: $N < N_1$

$$\mathbf{x} = \mathbf{\Psi s}$$

where $\mathbf{s}$ is $K -$ sparse

- Problem: Recover $\mathbf{s}$ from $\mathbf{y} = \mathbf{As}$, where $\mathbf{A} = \mathbf{\Phi\Psi}$

- Distributed CS: Recovery of jointly sparse vectors from multiple measurement vectors (MMV)
- To reconstruct $\mathbf{S}$ from $\mathbf{Y} = \mathbf{A}\mathbf{S}$, where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_L] \in \mathbb{R}^{M \times L}$, $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_L] \in \mathbb{R}^{N \times L}$
- Recovery algorithms
    - Greedy methods (Simultaneous OMP etc.)
    - Bayesian methods (M-SBL etc.)
    - Relaxed mixed norm minimization methods

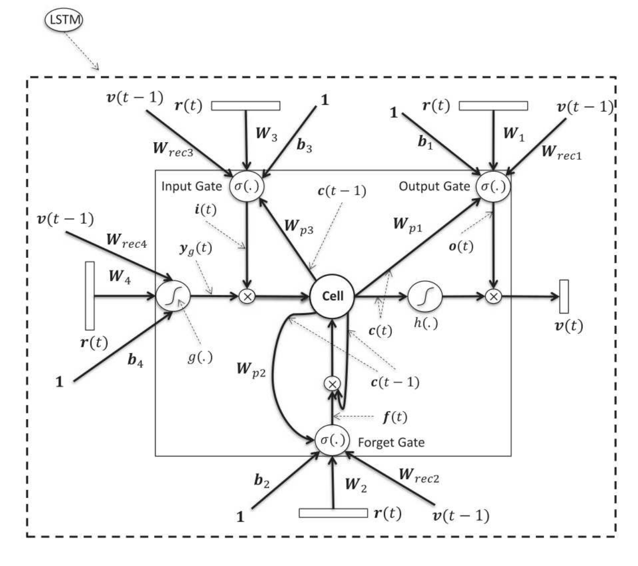
Questions!!!

- Traditional recovery algorithms do not rely on the use of training data
    - Large training data are generally available
    - Examples: Camera recordings of the environment, images of the same class etc.
- Can we learn the structure of the sparse vectors in $\mathbf{S}$ by a data driven approach using the available training data?
    - Sparse vectors need not be joint sparse in many practical applications
    - But the entries can be dependent
- How do we effectively use the learned structure to reconstruct the sparse vectors?

Overview of Proposed Method

- Two step greedy reconstruction algorithm
- Step 1: Support recovery:
  - In iteration $j+1$, for each column $\mathbf{s}_i$, find $P(\mathbf{s}_i[n] \neq 0 \,|\, \mathbf{R}_j)$, $i = 1, \ldots, L$, $n = 1, \ldots, N$ and $\mathbf{R}_j = \mathbf{Y} - \mathbf{A}\mathbf{S}_j$
  - Recurrent neural network (RNN) with long short term memory (LSTM)
- Step 2: Signal recovery by solving least squares

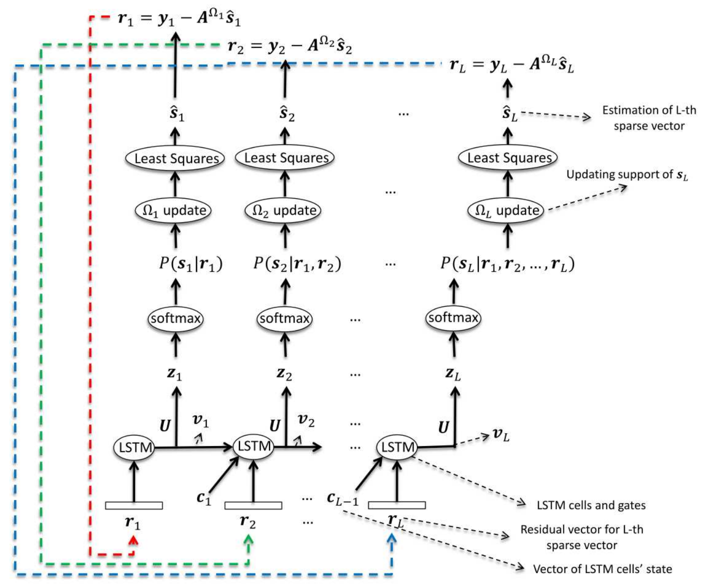
Block Diagram of LSTM



Forward pass for LSTM

$$
\begin{aligned}
\mathbf{y}_g(t) &= g\left(\mathbf{W}_4\mathbf{r}(t) + \mathbf{W}_{\mathrm{rec}4}\mathbf{v}(t-1) + \mathbf{b}_4\right) \\
\mathbf{i}(t) &= \sigma\left(\mathbf{W}_3\mathbf{r}(t) + \mathbf{W}_{\mathrm{rec}3}\mathbf{v}(t-1) + \mathbf{W}_{p3}\mathbf{c}(t-1) + \mathbf{b}_3\right) \\
\mathbf{f}(t) &= \sigma\left(\mathbf{W}_2\mathbf{r}(t) + \mathbf{W}_{\mathrm{rec}2}\mathbf{v}(t-1) + \mathbf{W}_{p2}\mathbf{c}(t-1) + \mathbf{b}_2\right) \\
\mathbf{c}(t) &= \mathbf{f}(t) \circ \mathbf{c}(t-1) + \mathbf{i}(t) \circ \mathbf{y}_g(t) \\
\mathbf{o}(t) &= \sigma\left(\mathbf{W}_1\mathbf{r}(t) + \mathbf{W}_{\mathrm{rec}1}\mathbf{v}(t-1) + \mathbf{W}_{p1}\mathbf{c}(t) + \mathbf{b}_1\right) \\
\mathbf{v}(t) &= \mathbf{o}(t) \circ h(\mathbf{c}(t))
\end{aligned}
$$

Proposed Method



Training data generation

- Residuals
- One hot vectors of the correct support

Learning method

- Cross entropy loss function

$$L(\boldsymbol{\Lambda}) = \min_{\boldsymbol{\Lambda}} \left\{ \sum_{i=1}^{nB} \sum_{r=1}^{Bsize} \sum_{\tau=1}^{L} \sum_{j=1}^{N} L_{r,i,\tau,j}(\boldsymbol{\Lambda}) \right\}$$

$$L_{r,i,\tau,j}(\boldsymbol{\Lambda}) = -s_{0,r,i,\tau}(j) \log\left(s_{r,i,\tau}(j)\right)$$

where $\boldsymbol{\Lambda}$ are the model parameters to be learnt

- Experimental results provided

THANK YOU!