Basic description of BA-Sindy algorithm

Inputs: $X, \dot{X} \in \mathbb{R}^{m\times D}$, with $\Theta = [\theta_1, ...\theta_p]$, and $\theta_i : \mathbb{R}^d \to \mathbb{R}$

Paramaters: Pretrain epochs n_{pre}, outer loops epochs n_{train}, inner loop 'bagging' epochs n_{bag}, inner loop train epochs n_{sub}, refinement epochs n_{ref}

Learning rates for pre-training, bagging and subtraining, $lr_{pre}, lr_{bag}, lr_{sub}$

Bag size q and bag number p Loss weightings $\lambda_0, \lambda_1, \lambda_2, \lambda_3$

Activation and bagging thresholds ε_{act} and ε_{bagg}. Noise paramater α

1. Train regular Sindy autoencoder without sequential thresholding for n_{pre} epochs, learning rate lr_{pre}. This yields an encoder $\zeta_0 : \mathbb{R}^D \to \mathbb{R}^d$, a decoder $\psi_0 : \mathbb{R}^D \to \mathbb{R}^d$ and a Sindy coefficient matrix $\Xi_0 \in \mathbb{R}^{d\times p}$.

Set $\mathcal{L}_{bag} = \lambda_1\mathcal{L}_{dx/dt} + \lambda_2\mathcal{L}_{dz/dt} + \lambda_3\mathcal{L}_{reg}$

Set $\mathcal{L}_{full} = \lambda_0\mathcal{L}_{encoder} + \lambda_1\mathcal{L}_{dx/dt} + \lambda_2\mathcal{L}_{dz/dt} + \lambda_3\mathcal{L}_{reg}$

Set coefficient mask $\Lambda^{(0)} = 1^{d\times p}$

2. Let $i_1, ...i_p \subset \{1, ...m\}$ and $|i_k| = q$. Choosen with replacement.

We typically choose $q = \text{int}(1.5m/p)$ Use indexes $i_1, ...i_p$ to create data bags $X_{i_1}, ...X_{i_p}$, and $\dot{X}_{i_1}, ...\dot{X}_{i_p}$,where $X_{i_k}, \dot{X}_{i_k} \in \mathbb{R}^{q\times D}$.

3. For $k = 0, ...n_{train}$:

  For $j = 1, ...p$:

    Get $\Xi_{rand} \in \mathbb{R}^{p\times d}$ with $(\Xi_{rand})_{i,j} \sim \mathcal{N}(0, 1)$

    $\Xi^{(k,j)} = \text{copy}(\Xi^{(k)}) + \alpha\Xi_{rand}$

    Update $\Xi^{(k,j)}$ using $\nabla_\Xi\mathcal{L}_{bag}(X_{i_j}\dot{X}_{i_j})$ for n_{bag} epochs, with lr_{bag}

  Set $\Xi^{(k)} = \frac{1}{p}\sum_j \Xi^{(k,j)}$ and $\overline{\Lambda}^{(k)}_{i,j} = \mathbb{1}\{\sum_h \mathbb{1}\{|\Xi^{(k,h)}_{i,j}| > \varepsilon_{act}\} > \varepsilon_{bag}\}$.

  Set $\Lambda^{(k+1)} = \overline{\Lambda}^{(k)} \odot \Lambda^{(k)}$

  Update $\Xi^{(k)}, \zeta^{(k)}, \psi^{(k)}$, using $\nabla_{\Xi,\zeta,\psi}\mathcal{L}_{full}(X, \dot{X})$, for n_{sub} epochs w lr_{sub}.

  Set $\Xi^{(k+1)}, \zeta^{(k+1)}, \psi^{(k+1)} = \Xi^{(k)}, \zeta^{(k)}, \psi^{(k)}$

4. Update $\Xi^{(n)}, \zeta^{(n)}, \psi^{(n)}$, using $\nabla_{\Xi,\zeta,\psi}\mathcal{L}_{full}(X, \dot{X})$, for n_{ref} epochs w lr_{sub}.

Preliminary results on Lorentz system

Train & test data was of dim 128×5000, generated from Lorentz system.

Loss and coefficient count for each averaged over 5 runs is given below.

avg_loss.png

avg_coeff.png

On some runs we uniformly beat regular auto encoder Sindy, achieving better test loss greater sparsity for the same # of epochs. Examples below:

run1_loss.png	run3_loss.png
run1_coeffs.png	run3_coeffs.png