Improving Imputation Using Stacked denoising Autoencoder

Najmeh Abiri

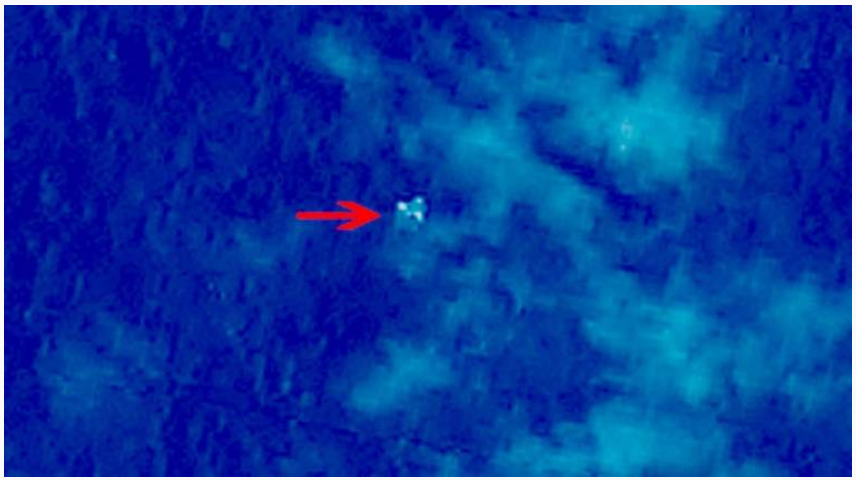
November 22, 2016

Computational Biology and Biological Physics

Missing Data

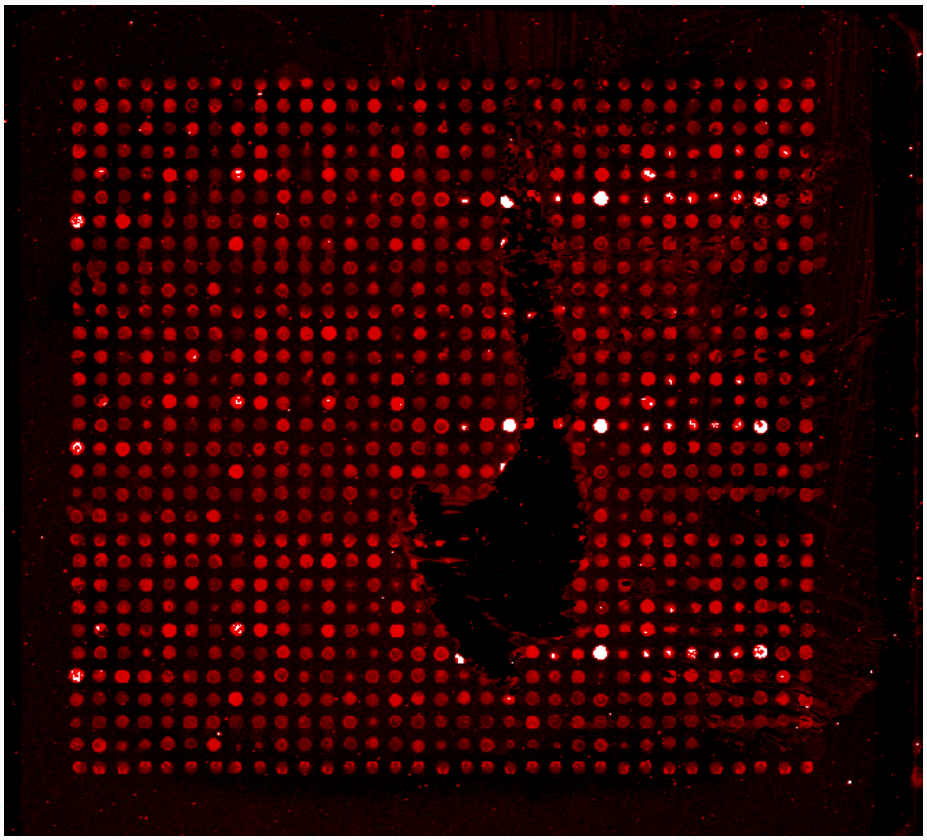
Pre-processing data

Astronomy



Outlier?

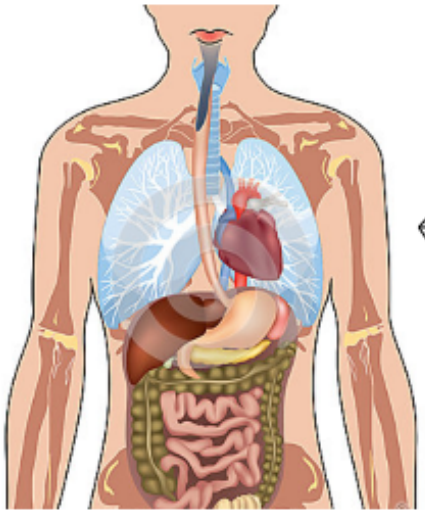
Biology

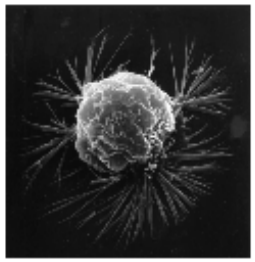


Missing Data?

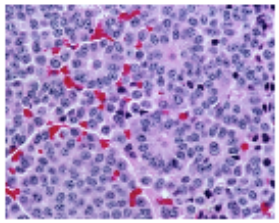
Missing data in Biology

Molecular Patterns of Life

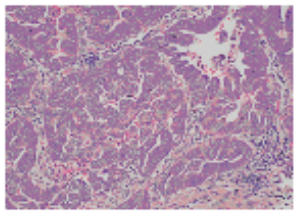




Breast



Lung



Intestinal

Blood / tissue

Disease fingerprint

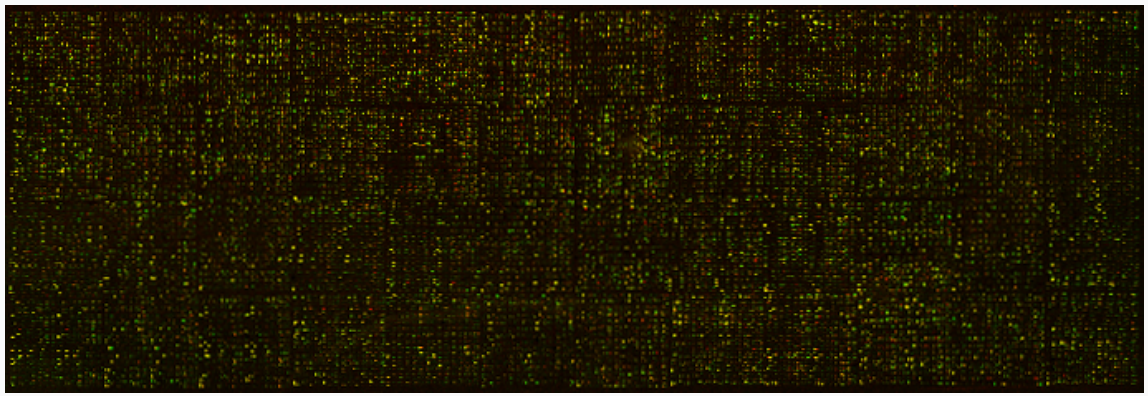
- Tse-Wen Chang, TW (1983) Antibody microarray
- Schena, M.; Shalon, D.; Davis, R. W.; Brown, P. O. (1995) DNA microarray

Missing data in Biology

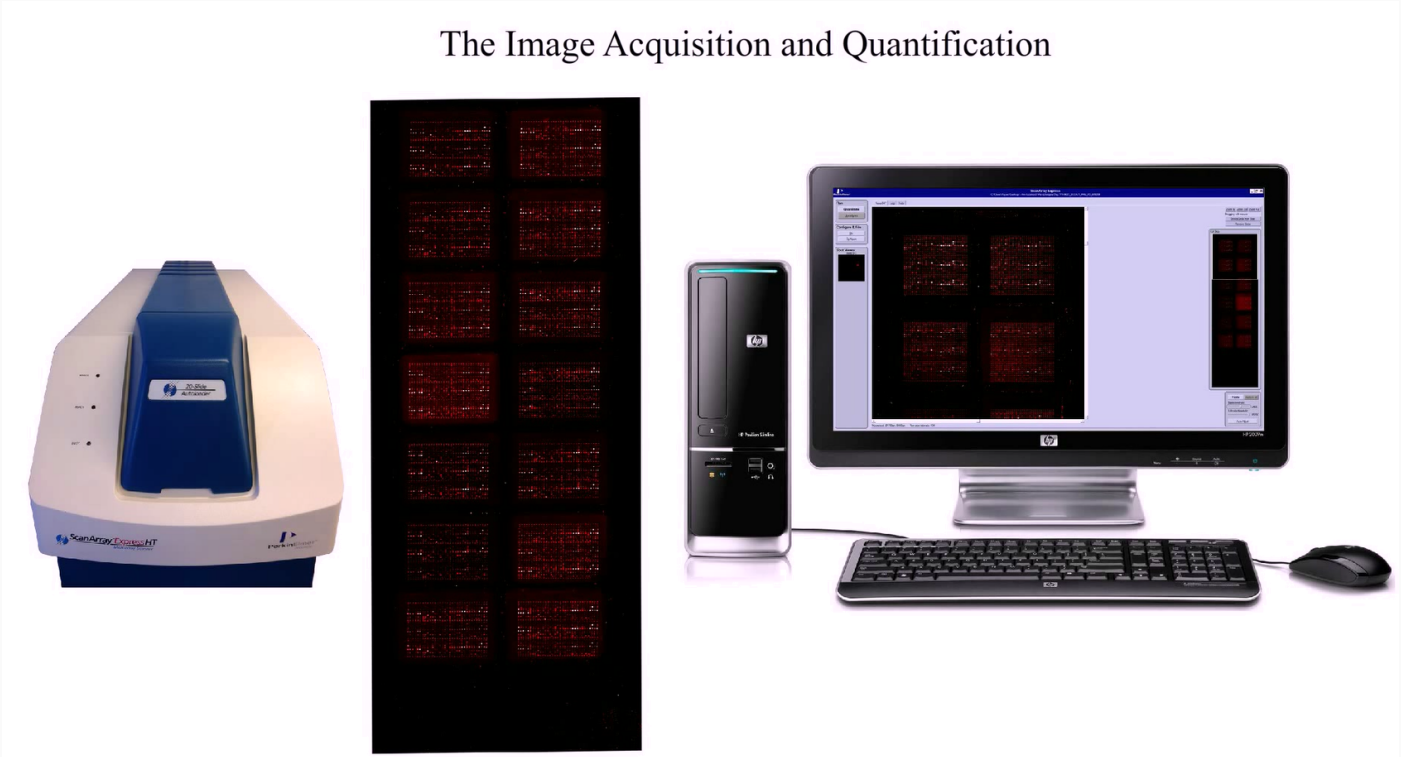
Generate detailed DNA/protein molecular fingerprints and Use them in :

1- Diagnosis
2- Prognosis
3- Classification
4- Monitoring

Microarray : - Measuring many(all) proteins/mRNA at once.
- Cancer research : minimize side effects and cost.



Missing data in Biology



Library of antibodies is arrayed on the support surface(glass or silicon).

Imputation

Consequences of missing data:

- Redo the experiments $\rightarrow$ expensive
- Risk of bias $\rightarrow$ depends on the reasons why data are missing
- Non-normally distributed variables $\rightarrow$ imputation procedures could produce some implausibly low or even negative values
- Data that are missing not at random
- Computational problems

Imputation Algorithms

Traditional approaches:

- case deletion
- mean imputation: the replacement of a missing observation with the mean of the non-missing observations for that variable.

More technical methods:

- **K-nearest neighbors(KNN)**[1]
- Bayesian principal component (bPCA)
- ...

[1] Improved methods for the imputation of missing data by nearest neighbor methods,Tutz, Gerhard and Ramzan, Shahla(2015)

KNN

Data with missing value:

	var 1	var 2	var 3 . . .
S 1	123.23	21.234	234.2. . .
S 2	23.345	nan	234.2. . .
⋮	⋮	⋮	⋮

KNN approach is to find k nearest distance and calculate the weighted mean:

Distance matrix :

$d_q(x_i, x_j) = [\frac{1}{m_{ij}} \sum_{s=1}^{p} |x_{is} - x_{js}|^q]^{\frac{1}{q}}$

where m_{ij} denotes the number of valid components in the computation of distances.

Weighted imputation:

$\hat{x}_{is} = \sum_{j=1}^{k} w_{ij} x_{js}$

Using deep learning in imputation

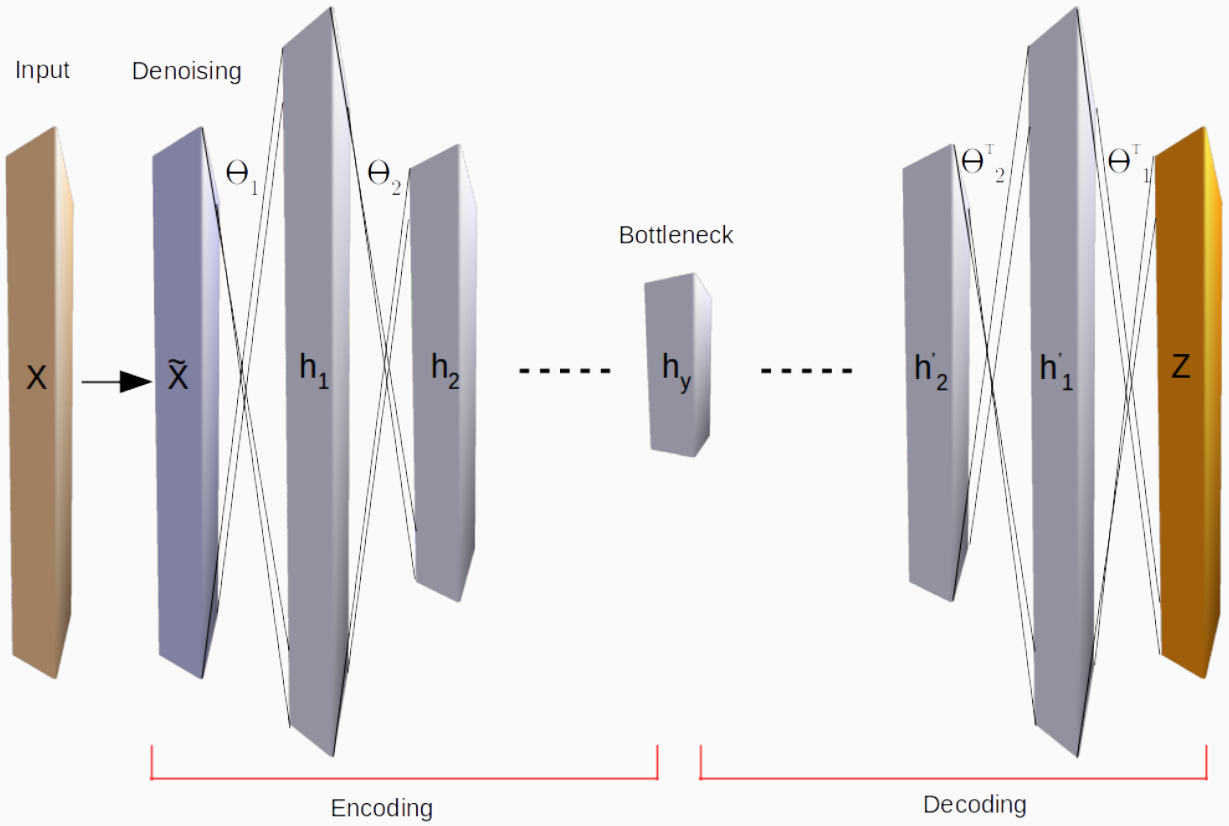
Autoencoders

- Trained to encode their input to a lower dimensional representation.
- Capture the significant features by compressing input data to low-dimensional vectors.
- ...

What form of Autoencoders:

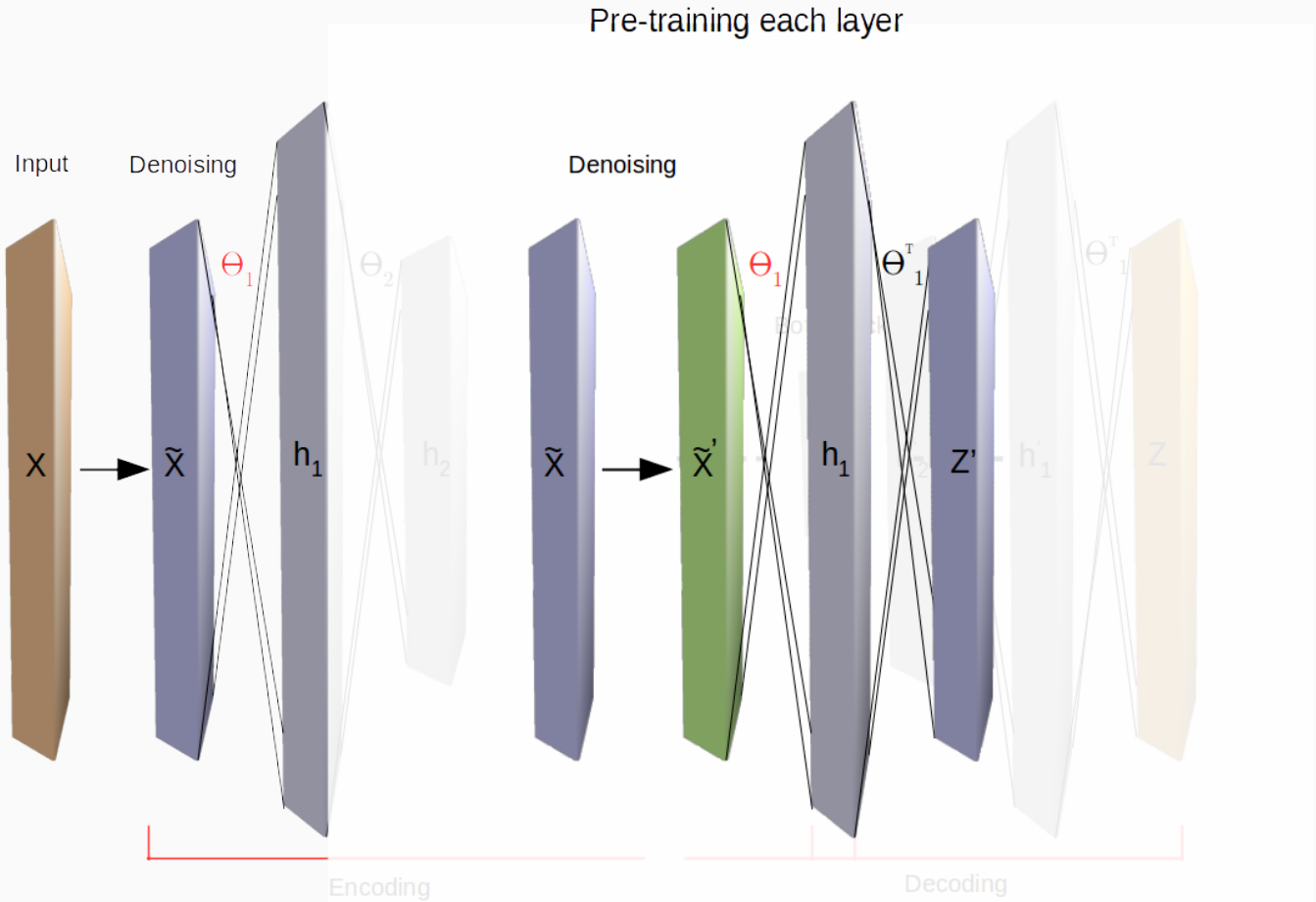
- Denoising technique.
- Tied weights: $W = W^T$, decrease the encoder probability of staying in the linear conformation.
- Symmetric decoder and encoder system $\rightarrow$ butterfly construction.
- Layer-wise unsupervised pre-training(initializing the layer parameters $\theta = \{W, b\}$), followed by supervised fine-tuning.

Stacked Denoising Autoencoder



Networks architecture: Z is the reconstructed X

Stacked Denoising Autoencoder



SDA with initialization

Stacked denoising Autoencoder imputation

SDA imputation

SDA, an imputation box :

- Performance of an SDAi depends on the data correlation.
- Complete training set will give more accurate network.
- A network has the ability to either use the index of missing data in error optimization or calculate without notation.
- Data with higher number of samples gives a better result.
- Iterative imputation (basic algorithms) for estimating an initialization for missing date(nan).

SDA imputation

Hyperparameters :

- Number of hidden layers.
- Regularization and updating methods.
- Epoch number for pretraining and finetune training.
- Fraction of denoising for each DA layer (initialization) and fine-tuning.
- Learning rate for each DA layer and main network.
- Mini batch size (for SGD).

Result

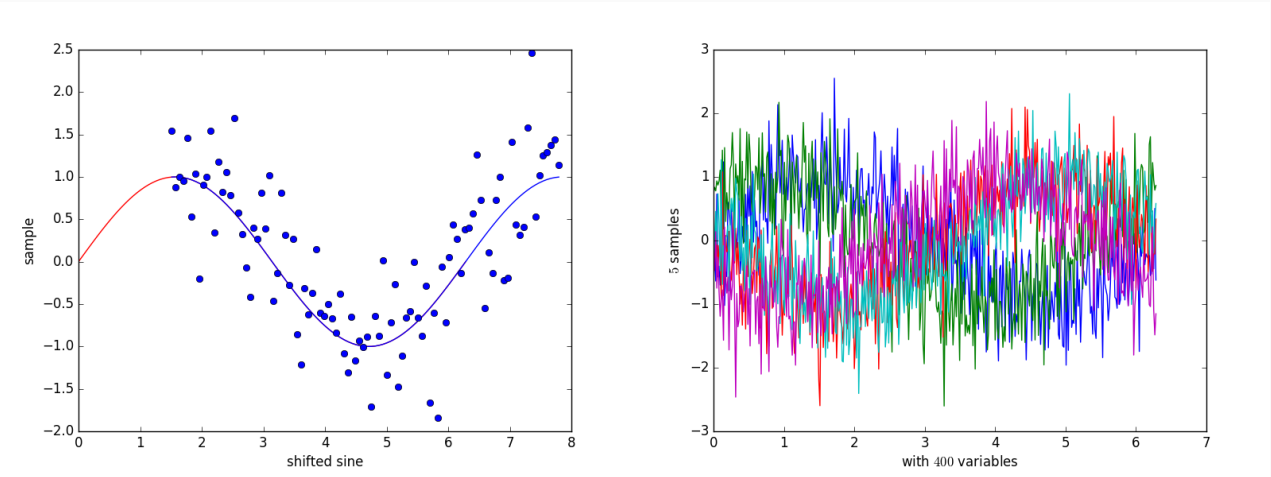
Synthetic Data

Shifting a sine function in x axis with a white noise with continuous distribution.

$$S_i = sin(x_i) + z_i,$$

$$\text{where} \quad x_i \in \mathbb{R} \quad | \quad 0 + u_i \leq x_i \leq 2\pi + u_i,$$

$$u_i \sim \mathcal{U}(0, 2\pi) \quad \& \quad z_i \sim \mathcal{N}(0, 0.5).$$



result

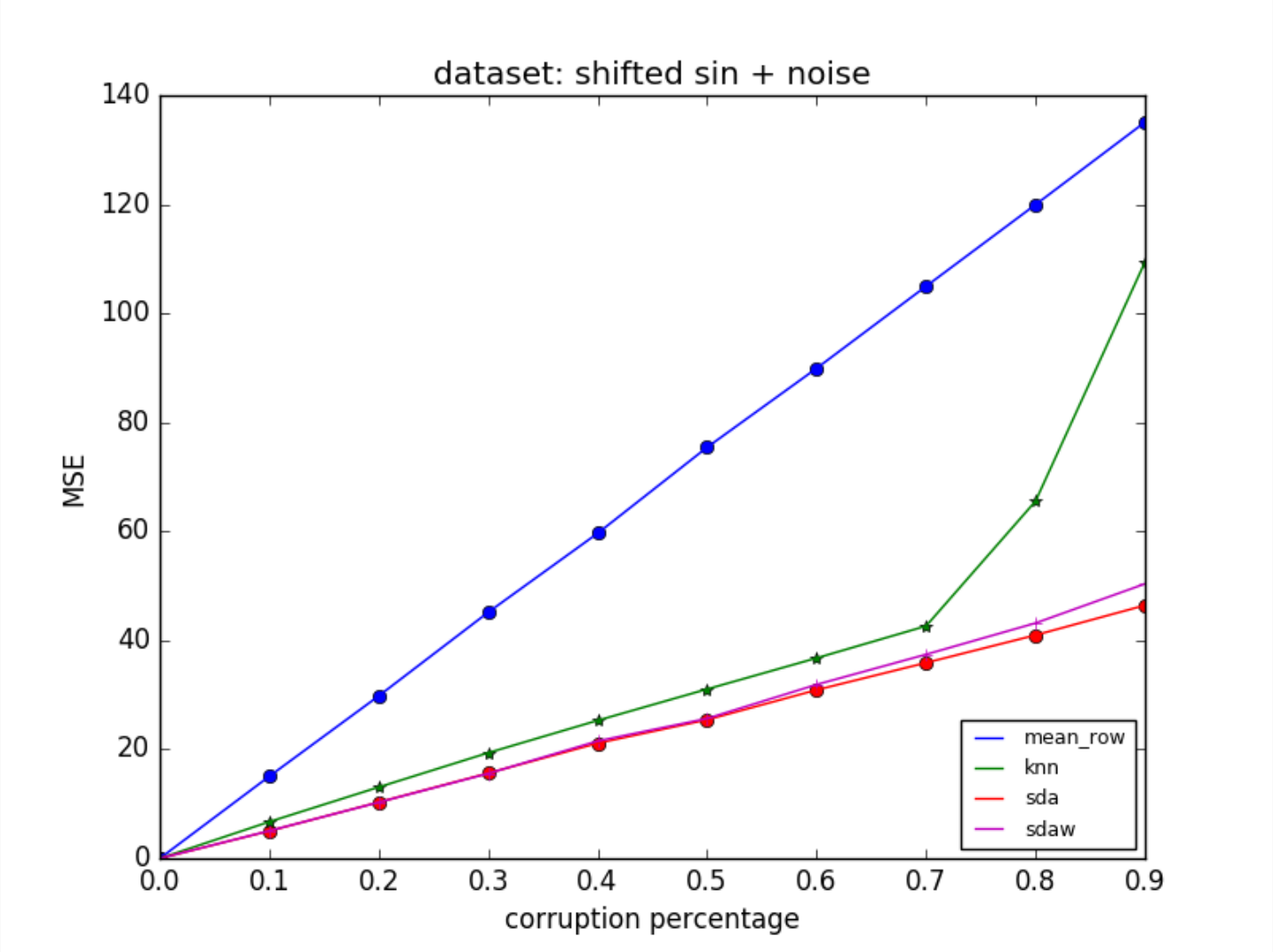


Figure 1: SDA with and without initialization. Layers= [100,20,2], fraction: range(.0,.9)

MNIST

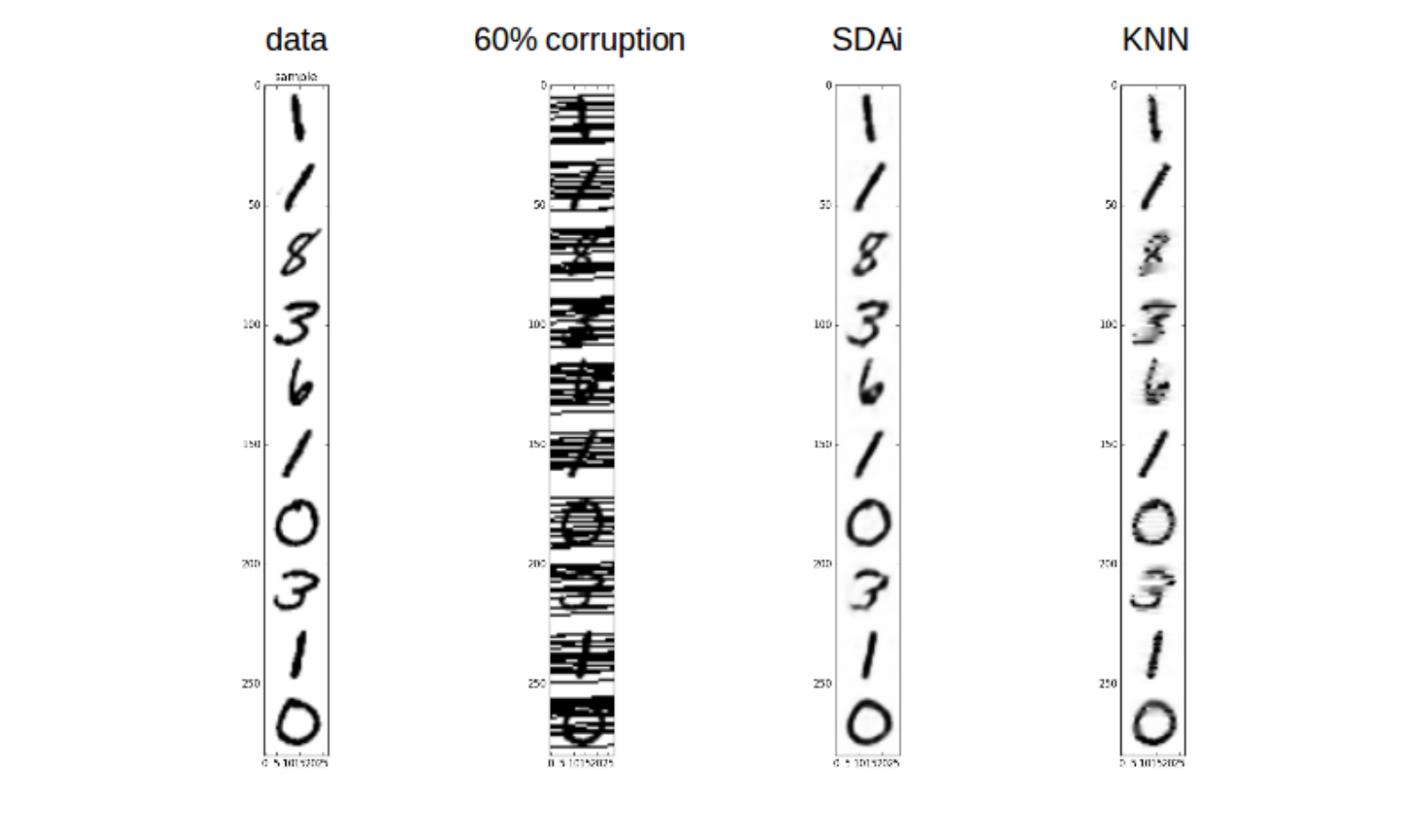


Figure 2: MNIST with 60% corruption

result

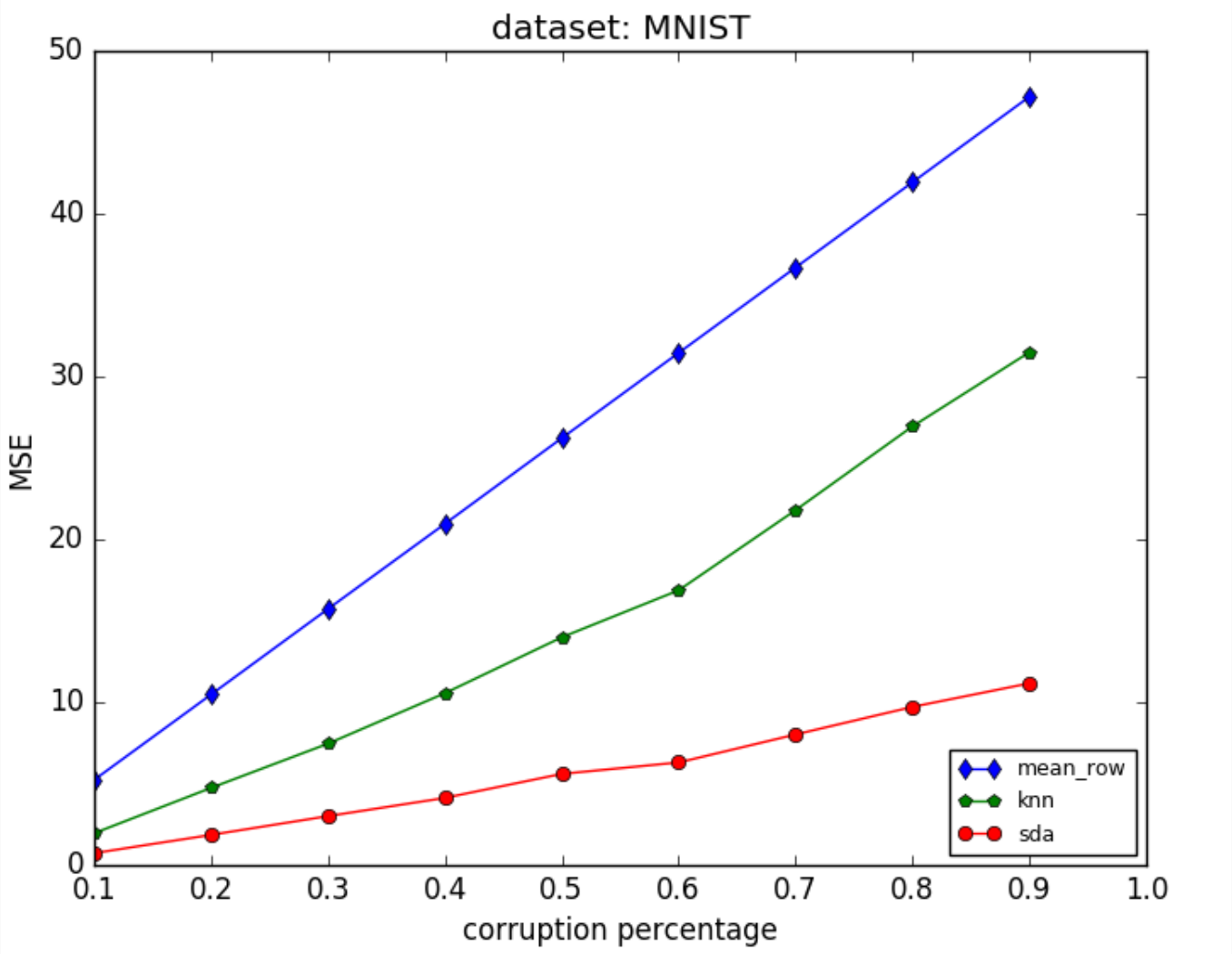


Figure 3: SDA with initialization. Layers= [1000,500,10]

RNA

Data shape : 172 samples and 5000 features. PCA shows 168 important eigenvalues.

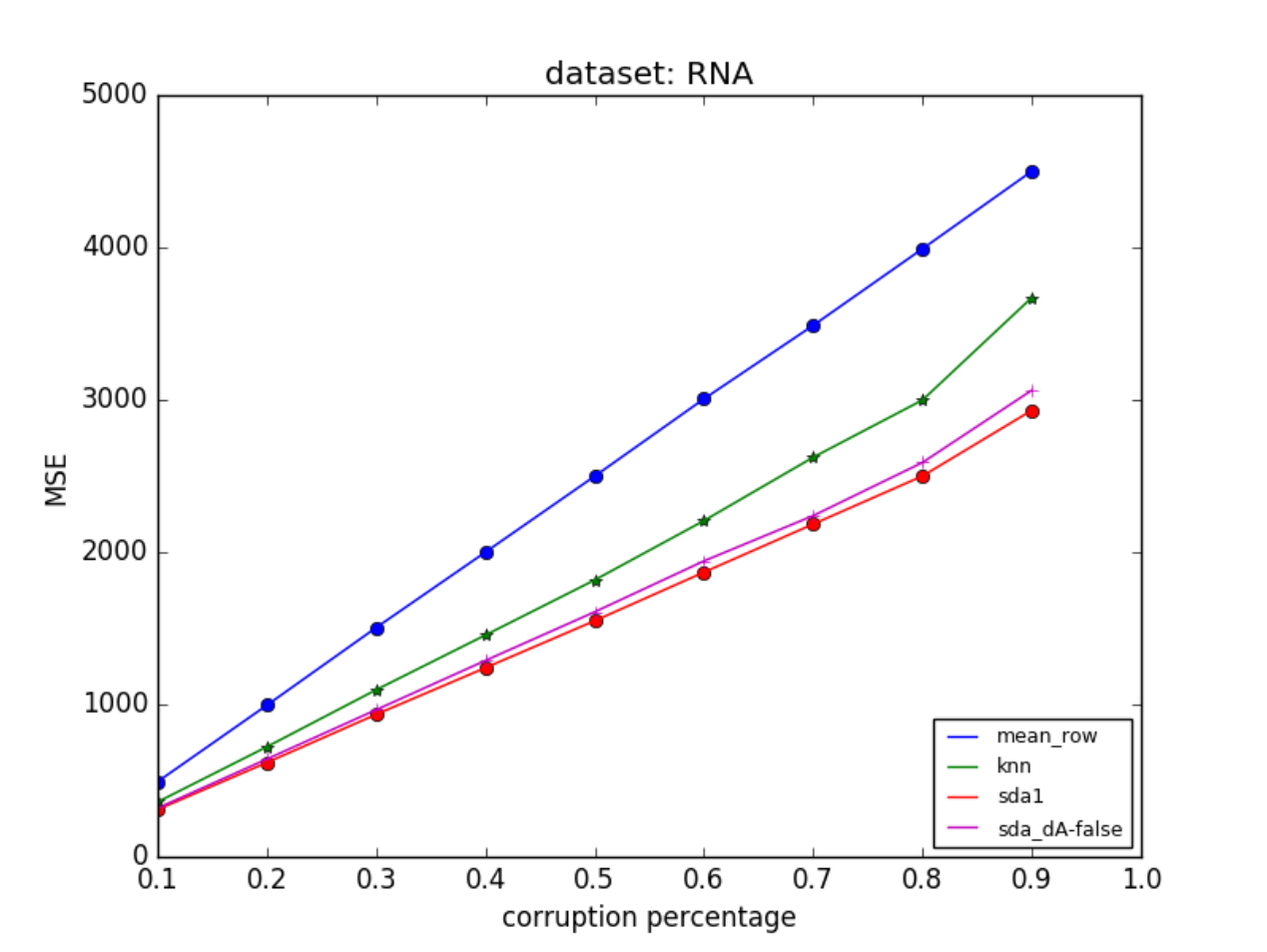


Figure 4: SDA with initialization. Layers= [4000,1000,168]

Next?

- Other data sets
- Classification

Homework:

- My homework: SDAi from theano to tensorflow
- Our homework : ▸ The SDA article ▸ Data