Noise Enhancement: Techniques and Applications

Reyhaneh Abdolazimi and Reza Zafarani

Syracuse University

International Conference on Data Mining

Presenters

Reyhaneh Abdolazimi
Data Lab, Syracuse University

Reza Zafarani
Data Lab, Syracuse University

Introduction

- Noise usually describes undesirable disturbances or fluctuations.
- Noise in Biology
  - variability in measured data when identical experiments are repeated
  - when bio signals cannot be measured without background fluctuations distorting the desired measurement.

- Real-world data is seldom clean
  - Noise in the data can create problems for deep learning and neural networks.
    - decrease the performance of neural networks
    - less generalization power during testing on real-world data

[McDonnel et al., 2009]

Introduction

- Noise is also the fundamental enemy for communications engineers
  - Messages can be transmitted error-free and efficiently
  - When random noise in the form of electronic fluctuations corrupts transmitted messages

- If everything else is ideal, then noise is the enemy.

What if not everything is ideal?

Can an ideal system always be implemented in practice?

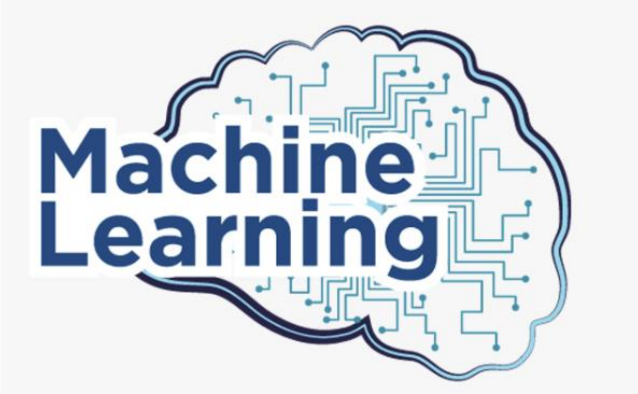
No

tradeoffs between different conflicting objectives

[McDonnel et al., 2009]

Introduction

- Noise can improve information processing
  - non-linear systems and algorithms



https://www.pngitem.com/

Image credit: seredaserhii

Image credit: Bernard Marr

Outline

- Part I
  - Signal Processing
  - Image Processing
  - Biology and Physics
- Part II
  - Expectation Maximization
  - Clustering
  - Markov Chains and HMM
- Part III
  - Benefits of Noise in Neural Networks
  - Injecting Noise in Neural Networks
- Part IV
  - Network Science
  - Privacy Preservation
  - Natural Language Processing

Part I

- Noise-Enhanced Signal Processing
  - Stochastic Resonance
  - Signal Detection
  - Parameter Estimation
- Noise-Enhanced Image Processing
- Noise-Enhanced Biology and Physics

Stochastic Resonance (SR)

- By Roberto Benzi in 1981
  - Explain the periodicity of Earth's ice ages

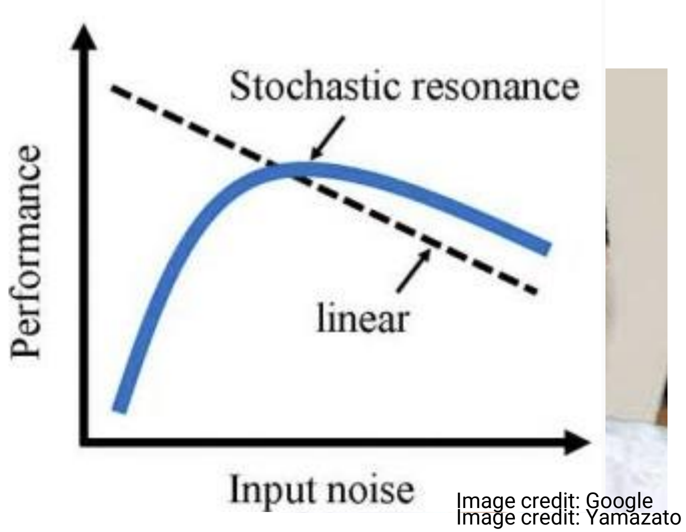


Image credit: Google
Image credit: Yamazato Laboratory

- Use in the context of **signal processing**
  - Effect of additive noise on the detectability of the signal : **signal-to-noise ratio**

- Phe

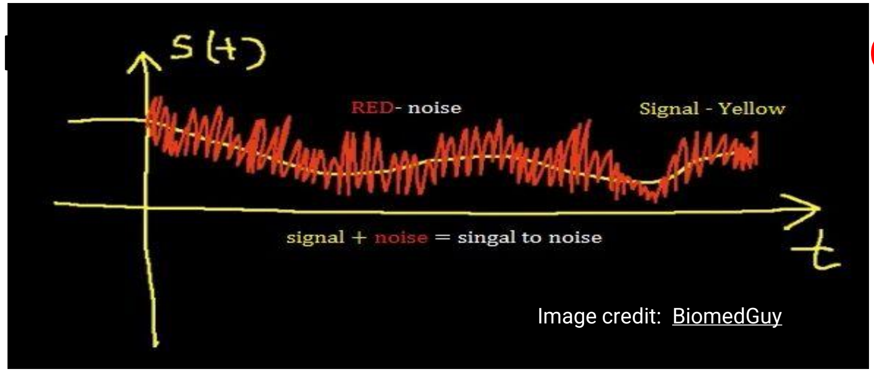


Image credit: BiomedGuy

oise is better t

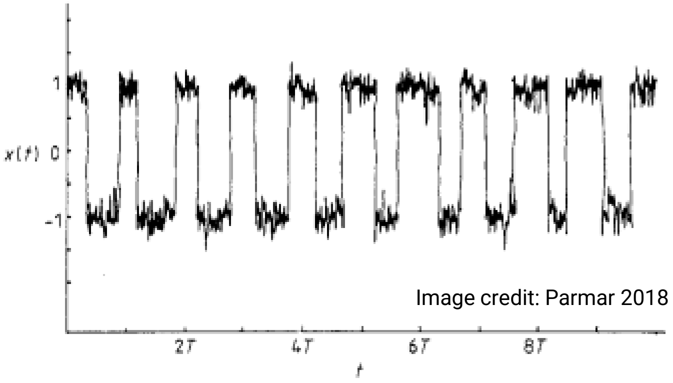
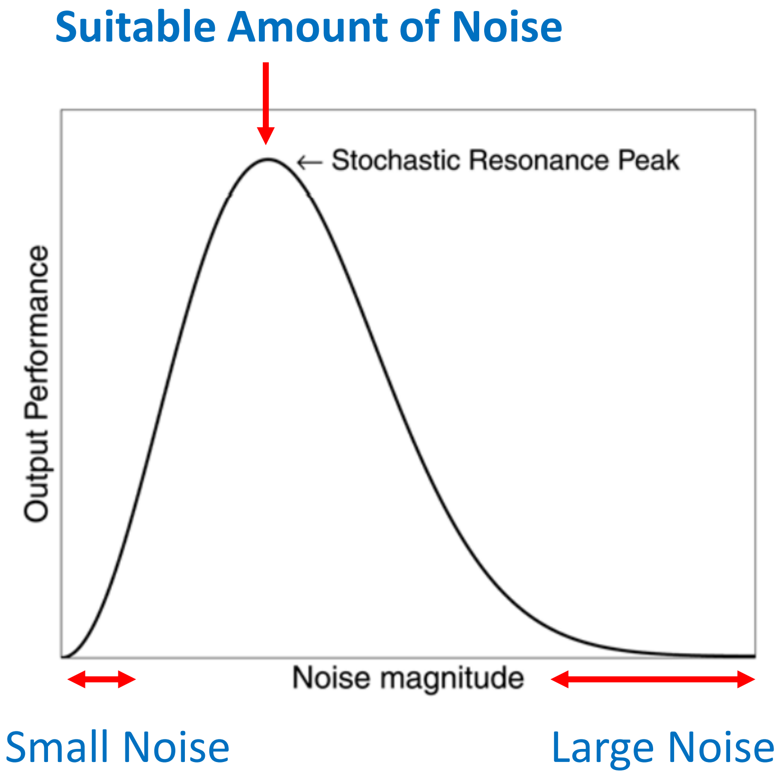


Image credit: Parmar 2018

[Benzi et al., 1981, Gammaitoni et al.,1998, Mcdonnell et al.,2009]

Stochastic Resonance

Performance improvement of a non-linear system by noise enhancement.



[McDonnell et al., 2009]

Signal Detection

- Noise-enhancement : beneficial in nonlinear detectors
  - noise ability to **increase the detectability**
- Noise : help to detect a weak sinusoid signal
  - force cooperation between sine and noise : nois
- White Gaussian noise : improve the perform
  - detect a constant signal in a Gaussian mixture n

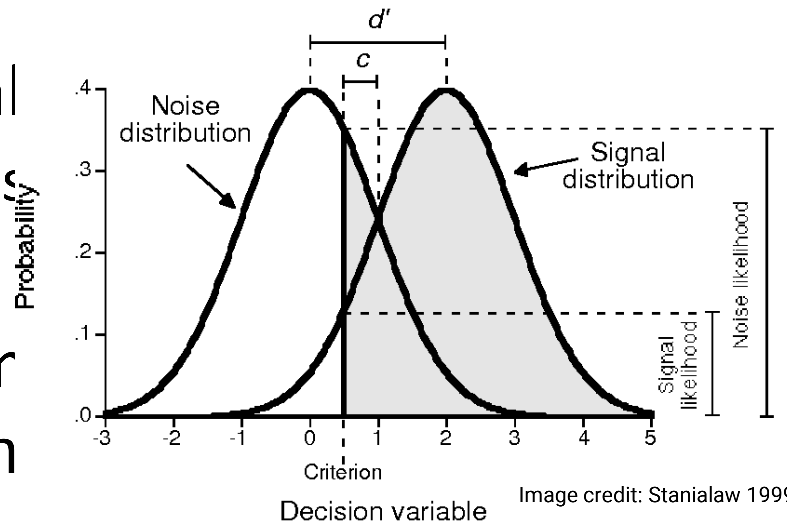


Image credit: Stanialaw 19999

[zozor et al., 2002, Kay et al., 2000]

Performance Improvement of Detectors

Improving the performance of detection system through additive noise

- The stochastic resonance effect in binary hypothesis testing problem
  - both fixed and variable detectors

3172 IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 55, NO. 7, JULY 2007

Theory of the Stochastic Resonance Effect in Signal Detection: Part I—Fixed Detectors

Hao Chen, *Student Member, IEEE*, Pramod K. Varshney, *Fellow, IEEE*, Steven M. Kay, *Fellow, IEEE*, and James H. Michels, *Fellow, IEEE*

$$
\begin{aligned}
P_{\mathrm{D}}^{y} &= \int_{R^N} p_{\mathbf{n}}(\mathbf{x}) \left(\int_{R^N} \phi(\mathbf{y}) p_1(\mathbf{y}-\mathbf{x}) d\mathbf{y} \right) d\mathbf{x} \\
&= \int_{R^N} F_{1,\phi}(\mathbf{x}) p_{\mathbf{n}}(\mathbf{x}) d\mathbf{x} = E_{\mathbf{n}}(F_{1,\phi}(\mathbf{x}))
\end{aligned}
$$

$$
\begin{aligned}
P_{\mathrm{FA}}^{y} &= p_{\mathbf{n}}(\mathbf{x}) \left(\int_{R^N} \phi(\mathbf{y}) p_0(\mathbf{y}-\mathbf{x}) d\mathbf{y} \right) d\mathbf{x} \\
&= \int_{R^N} F_{0,\phi}(\mathbf{x}) p_{\mathbf{n}}(\mathbf{x}) d\mathbf{x} = E_{\mathbf{n}}(F_{0,\phi}(\mathbf{x}))
\end{aligned}
$$

$$
F_{i,\phi}(\mathbf{x}) \equiv \int_{R^N} \phi(\mathbf{y}) p_i(\mathbf{y}-\mathbf{x}) d\mathbf{y}
$$

$$
\begin{aligned}
P_{\mathrm{E}}^{y} &= \pi_0 P_{\mathrm{FA}}^{y} + \pi_1 (1 - P_{\mathrm{D}}^{y}) \\
&= \pi_1 + \int_{R^N} (\pi_0 F_{0,\phi}(\mathbf{x}) - \pi_1 F_{1,\phi}(\mathbf{x})) p_{\mathbf{n}}(\mathbf{x}) d\mathbf{x} \\
&= \pi_1 + \pi_0 E_{\mathbf{n}} \left(F_{0,\phi}(\mathbf{x}) - \frac{\pi_1}{\pi_0} F_{1,\phi}(\mathbf{x}) \right) \\
&\overset{f_0 = F_{0,\phi}(\mathbf{x})}{\geq} \pi_1 - \pi_0 E_{\mathbf{n}_{f_0}} \left(\frac{\pi_1}{\pi_0} J_{\phi}(f_0) - f_0 \right) \\
&\geq \pi_1 - \pi_0 G_{\phi}^{o}
\end{aligned}
$$

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 56, NO. 10, OCTOBER 2008 5031

Theory of the Stochastic Resonance Effect in Signal Detection—Part II: Variable Detectors

Hao Chen, *Member, IEEE*, and Pramod K. Varshney, *Fellow, IEEE*

[Chen et al. 2007, Chen et al. 2008]

Performance Improvement of Detectors

Consider a general problem with observations $x \in X$:

- H_0 and H_1 are binary hypothesis.
  - The pdfs of x under two hypothesis H_0 and H_1 are $p_0(x)$ and $p_1(x)$
- $n \in N$ is an independent additive noise with pdf $p_n(n) \in P_N$.

$$n + x \Rightarrow y = x + n \in Y$$

- Performance of detector
  - Probability of detection P_d and Probability of false alarm P_f : Neyman–Pearson
  - Probability of error P_e : Bayesian

[Chen et al. 2007, Chen et al. 2008]

Neyman–Pearson Framework

- To maximize the detection probability P_d given $P_f <= \alpha$,
  - Form of the optimum noise : **randomization of two discrete vectors** (signals)

$$P_n^o = \boxed{\lambda}\boxed{\delta(n - n_1)} + \boxed{(1 - \lambda)}\boxed{\delta(n - n_2)}$$

where δ is a Dirac delta function, n_1 , and n_2 are appropriate noise parameters, and $\lambda \in [0, 1]$.

[Chen et al. 2007, Chen et al. 2008]

Bayesian Framework

- To minimize the probability of error P_e ,
  - Optimal noise form **: a single constant signal**

$$\boxed{P^o_{n,e} = \boxed{\delta(n - n_0)}}$$

where n_0 is an appropriate noise parameter.

[Chen et al. 2007, Chen et al. 2008]

Parameter Estimation

- Improvement of nonlinear estimators' Performance: **through noise injection to the input data**.

- **A general noise-enhanced parameter estimator**
  - with both additive and nonadditive noise
  - Determines the form of the optimal noise probability density function.

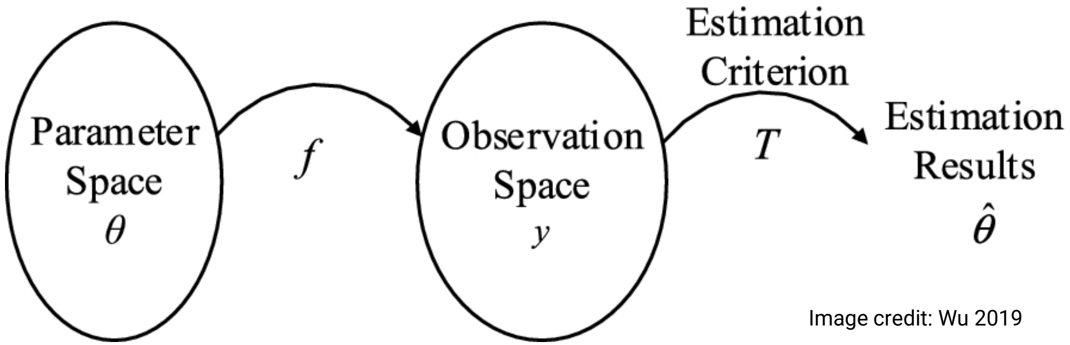


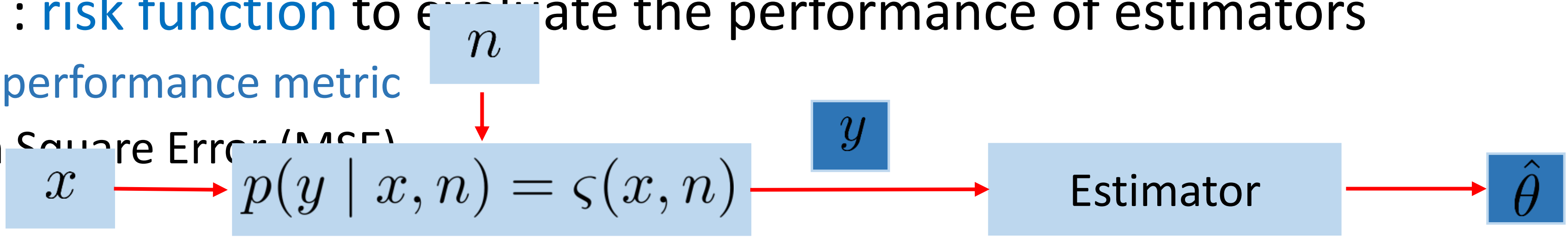
Image credit: Wu 2019

[Chen et al. 2008]

Performance Improvement of Estimators

Assumptions :

- x : input signal
- θ : estimation parameter
- $\hat{\theta} = T(x)$: an estimator that guess θ from x
- n : independent noise
  - ς : prespecified stochastic transform function
- y : input of estimator (noisy input)
- $r_i(\theta, \hat{\theta})$: risk function to evaluate the performance of estimators
  - i : performance metric
  - Mean Square Error (MSE)



[Chen et al. 2008]

Performance Improvement of Estimators

- To achieve the constraints on risk function $r_i \leq \alpha$
  - Noises distributions that can improve the estimator performance:

$$p_n(n) = \sum_{i=1}^{I} \boxed{\lambda_i} \boxed{\delta(n - n_i)}$$

- $\lambda_i \geq 0 \quad , \Sigma\lambda_i = 1$.
- n_i **are appropriate constant vectors** (signals) corresponding to the performance goals.

[Chen et al. 2008]

Part I

- Noise-Enhanced Signal Processing
- Noise-Enhanced Image Processing
    - Dithering
    - Image Processing Applications
- Noise-Enhanced Biology and Physics

Dithering

- **Adding random noise to pictures** before quantization in 1962
  - make a difference between the input and output
  - **randomize** quantization error
  - **Dithering** technique**: Signal-processing technique**

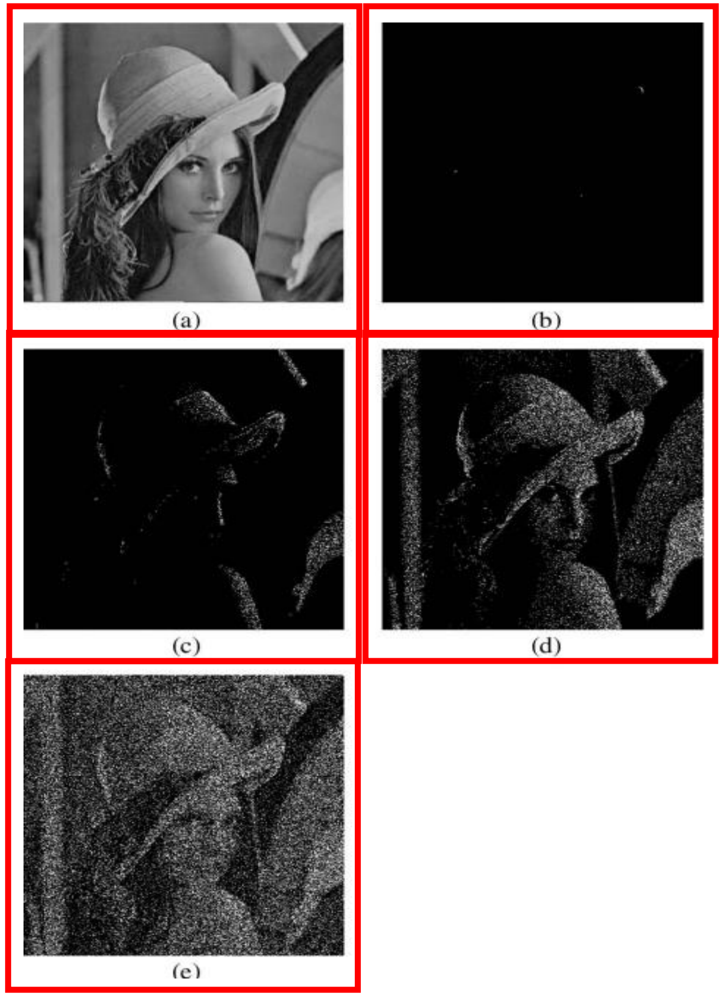
- Effect of noise on human visual perception
  - Contrast sensitivity
  - Noise in images of letters **increases the recognition sensitivity** of the human

[Roberts et al., 1962, Sasaki et al.,2006, Piana2000]

Effect of Noise on Image

Original Image

No noise



Little noise

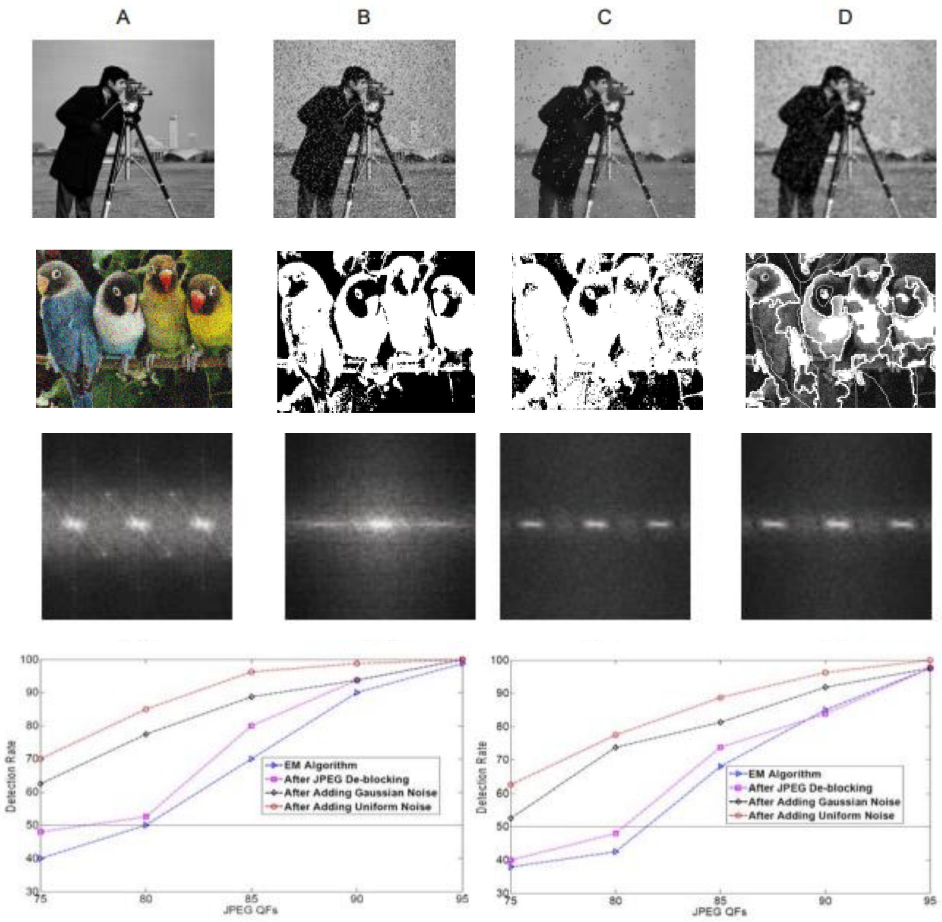
"Just right" amount of noise

Too much noise

[Chen et al. 2014]

Image Processing Applications

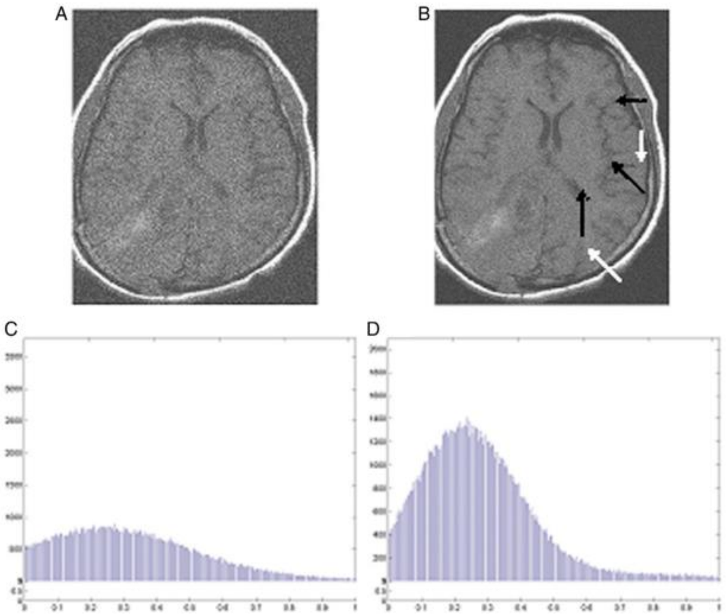
- Image restoration
  - Removing impulsive noise
- Image segmentation
  - Proper detection of objects
- Image resizing detection
  - Preventing failure after JPEG compression
- Image re-sampling detection
  - Helping the compressed JPEG images

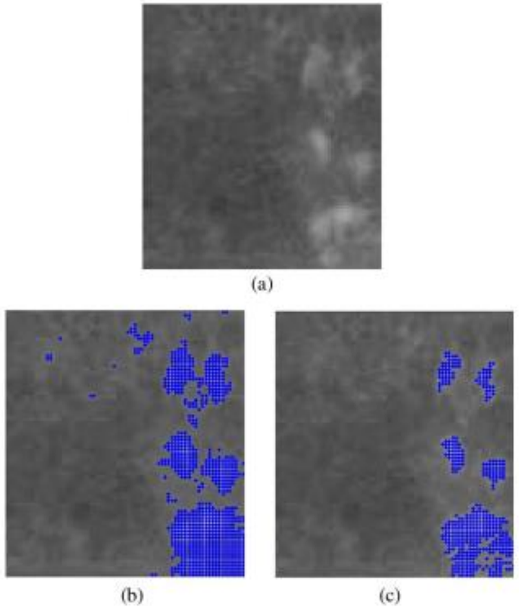


[Histace et al., 2006, Krishna et al., 2013, Nataraj et al., 2009, Nataraj et al. 2010]

Image Processing Applications

- Image enhancement:
  - Improving diagnosis of brain lesions
  - Accurate detection of micro-calcifications in mammograms





[Rallabandi & Roy, 2010, Peng et al., 2009, Blanchard et al. 2007]

Part I

- Noise-Enhanced Signal Processing

- Noise-Enhanced Image Processing

- Noise-Enhanced Biology and Physics

Noise-Enhanced Neuroscience and Biology

- Noise in the **transmission of sensory information** in neuron models
  - Investigated By Bulsara in 1991.

Image credit: Google

- Sensory system deal with weak signals
  - Timing of spiking events by **applying external noise** to the crayfish mechanoreceptors
    - SR : transmission of weak mechanical stimulus
  - SR : neurons in the cercal sensory system of a cricket

Image credit: Johnson 2015

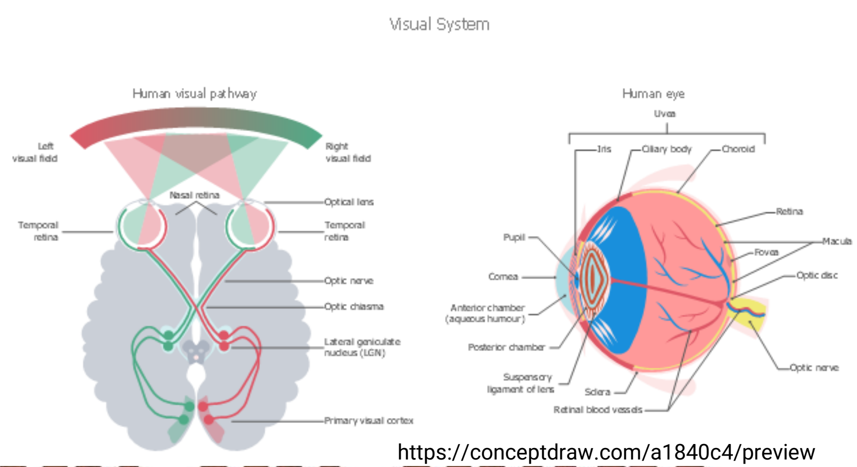
Image credit: https://www.e2f.com/blog

[Bulsara et al., 1991, Douglass et al. 1993, levin et al., 1996]

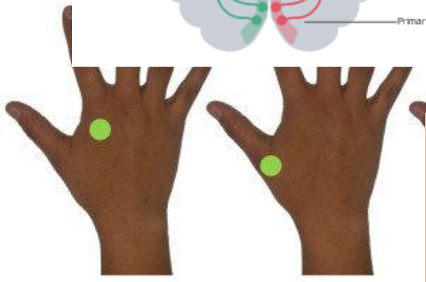
Noise Effects on Human Body

- Somatosensory function declines as people get older
  - Motor control
  - Detection and transmission in sensorimotor system
  - Balance performance
  - Vibrotactile sensitivity
- Visual system
  - Contrast detection sensitivity and visual motion d
- Human hearing
  - Perception, detection and discrimination of pure

Image credit: Razavian 2016



https://conceptdraw.com/a1840c4/preview



Dorsal Hand proximal to Index finger

Dorsal Hand proximal to Thumb

Image credit: Sanfilipo 2005

Image credit: Enders 2013

[Priplata et al., 2002, mendez et al., 2012, Dettmer et al., 2015, Liu et al., 2002]

Noise-Enhanced Quantum Physics

- Quantum noise or decoherence
  - Measure and handle Quantum noise.

- **Sto** n communication over qubit
  cha
  - nforma
  - ssical i te

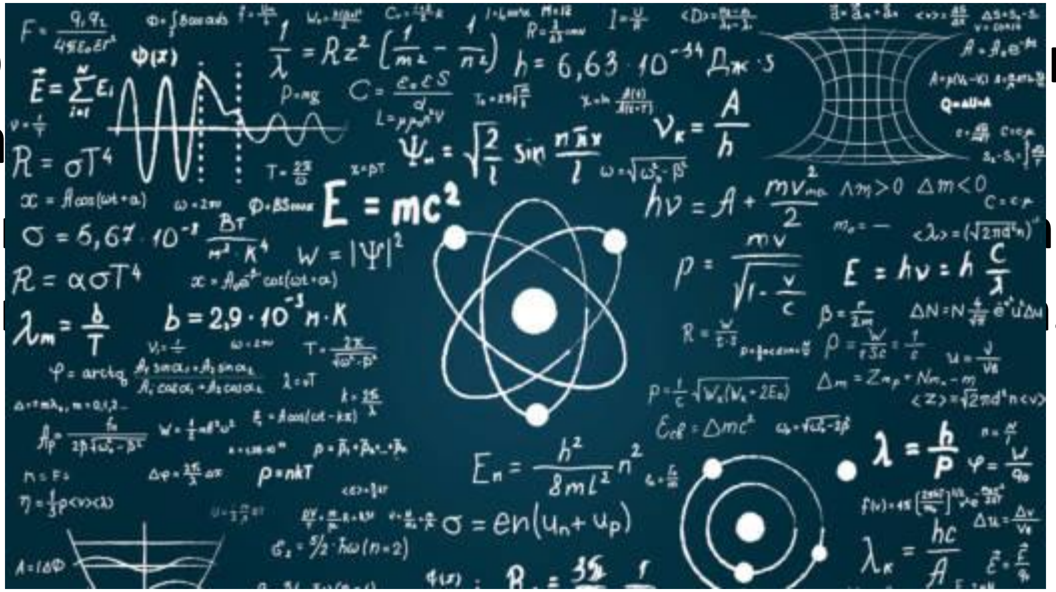
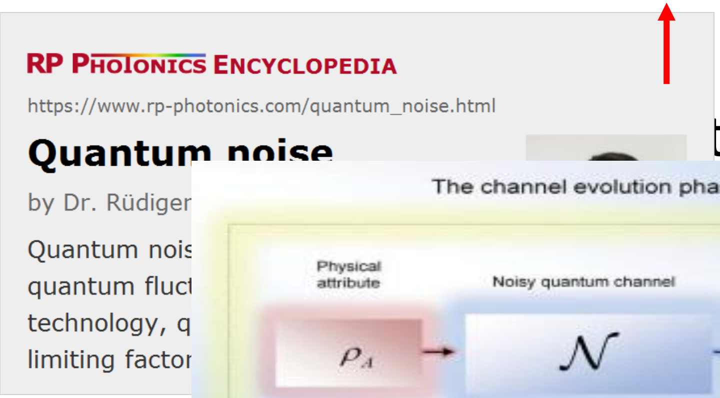


Image credit: Krancy 2018



RP Photonics Encyclopedia

https://www.rp-photonics.com/quantum_noise.html

Quantum noise

by Dr. Rüdiger

Quantum nois
quantum fluct
technology, q
limiting factor

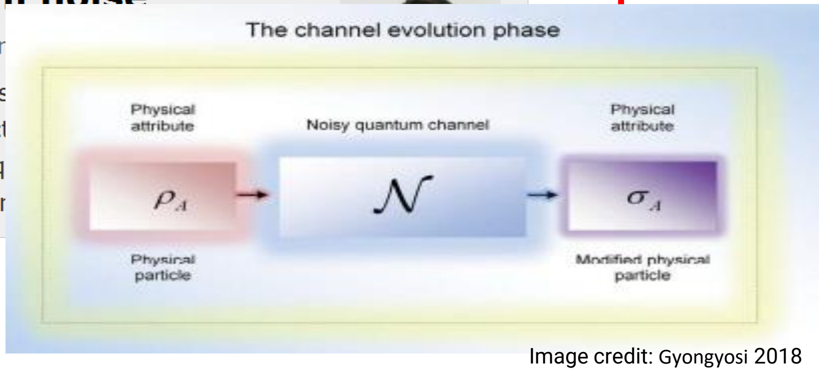


Image credit: Gyongyosi 2018

[bowen et al., 2006, bowen et al., 2004]

Noise-Enhanced Quantum Physics

- Noise enhancement : **quantum state detection or quantum state estimation**
  - Decohering environment as a thermal bath with finite temperature
    - **Increase** in noise temperature can **improve** the metrological performance from the noisy qubit.

Image credit: https://journalsofindia.com/rri-scientists-find-a-new-way-for-quantum-states-estimation/

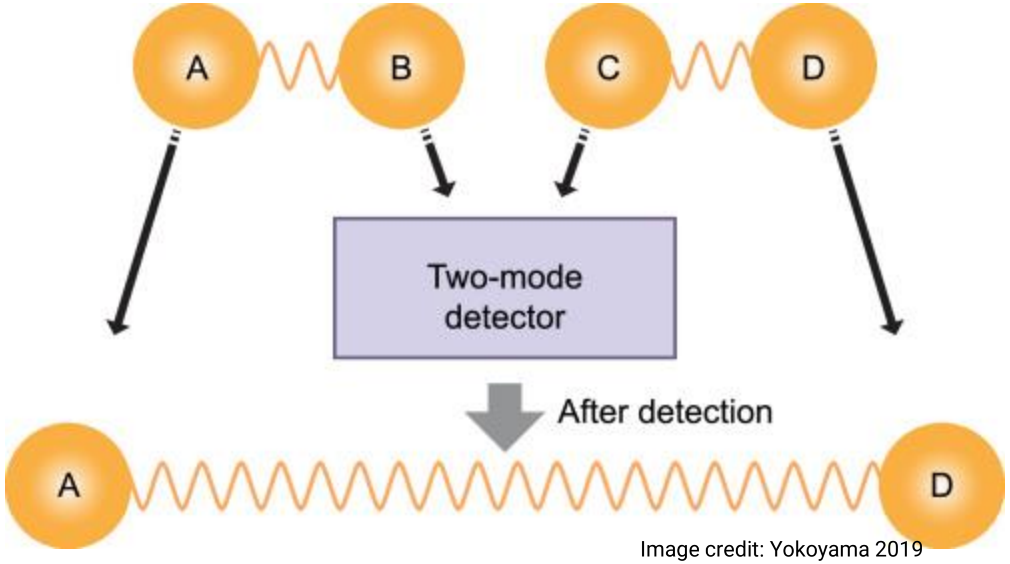


Image credit: Yokoyama 2019

[Chapeau-Blondeau 2015, Gillard et al., 2018]

Part II

- Expectation Maximization
    - Noisy Expectation Maximization Algorithm
    - Gaussian Mixture Models
- Clustering
- Markov Chains and HMM

Expectation Maximization Algorithm (EM)

- Goal : find the maximum-likelihood estimate $\hat{\theta}$ for the pdf parameter θ
  - the data Y has a parametric pdf $f(y|\theta)$

$$\boxed{\hat{\theta} = \arg\max_{\theta} \ell(\theta|y)}$$

  - $\ell(\theta|y) = \ln f(y|\theta)$ is the log-likelihood.

EM Algorithm

- Challenge: incomplete data : $\ell(\theta|y)$ complicated
  - random variable Z as hidden variable
  - $\ell(\theta|y,z)$: derive a surrogate function $Q(\theta|\theta_k)$

$$\text{Expectation Step: } Q(\theta|\theta_k) = E_Z\boxed{[\ell(\theta|y,Z)|}Y = y, \theta_k]$$

$$\text{Maximization Step: } \theta_{k+1} = \arg\max_\theta Q(\theta|\theta_k)$$

[Osoba et al., 2013]

Enhancing Noise to EM

- Reduction in the average convergence time
  - Additive noise : modifies the log-likelihood and its maximizer

$$\boxed{Q(\theta|\theta_k)} \qquad \boxed{\ln f(y, Z|\theta)}$$

$$\text{Expectation Step: } \boxed{Q_N(\theta|\theta_k)} = E_{Z|y,\theta_k}\left[\boxed{\ln f(y+N, Z|\theta)}\right]$$

$$\text{Maximization Step: } \theta_{k+1,N} = \arg\max_\theta \boxed{Q_N(\theta|\theta_k)}$$

[Osoba et al., 2013]

Enhancing Noise to EM

Assumptions:

- $Q(\theta|\theta_*)$ is the final surrogate log-likelihood given the optimal EM estimate θ_*
- θ_* maximizes $Q(\theta|\theta_*)$ and $Q(\theta_*|\theta_*) \geq Q(\theta|\theta_*)$ for all θ
- N is the noise random variable with pdf $f(n|y)$
- $\{\theta_k\}$ is a sequence of EM estimates for θ
- $\theta_* = \lim_{k \to +\infty} \theta_k$ as the converged EM estimate for θ

An EM noise benefit occurs when

$$\boxed{Q_N(\theta_k|\theta_*)} \geq \boxed{Q(\theta_k|\theta_*)}$$

or

$$\boxed{(Q(\theta_*|\theta_*) - Q_N(\theta_k|\theta_*})) \leq \boxed{(Q(\theta_*|\theta_*) - Q(\theta_k|\theta_*))}.$$

[Osoba et al., 2013]

Noisy Expectation–Maximization

NEM Theorem (Noisy expectation–maximization). The noise benefit for an EM estimation iteration

$$(Q(\theta_*|\theta_*) - Q(\theta_k|\theta_*)) \geq (Q(\theta_*|\theta_*) - Q_N(\theta_k|\theta_*))$$

occurs on average if

$$E_{Y,Z,N|\theta_*}\left[\ln\left(\frac{f(Y+N,Z|\theta_k)}{f(Y,Z|\theta_k)}\right)\right] \geq 0$$

[Osoba et al., 2013]

Gaussian Mixture Models (GMM)

- Mixture models : common data models in EM applications
  - forms of noise in mixture models : benefit from NEM theorem

Corollary 1 of the general NEM Theorem:

$$E_{Y,Z,N|\theta_*}\left[\ln\left(\frac{f(Y+N,Z|\theta_k)}{f(Y,Z|\theta_k)}\right)\right] \geq 0 \text{ occurs if } f(y+n,z|\theta) \geq f(y,z|\theta)$$

for almost all y, z , and n.

[Osoba et al., 2013]

GMM–NEM Noise Benefit Condition

- Corollary 1: conditions on the noise N that produce NEM noise benefits for mixture models.

Corollary 2 of the general NEM Theorem:

Suppose $Y|_{Z=j} \sim N(\mu_j, \sigma_j^2)$ and thus $f(y|j,\theta)$ is a normal pdf.

Then

$$f(y+n|j,\theta) - f(y|j,\theta) = \Delta f_j(y,n) \geq 0$$

occurs if

$$n^2 \leq 2n(\mu_j - y).$$

[Osoba et al., 2013]

GMM–NEM Noise Benefit

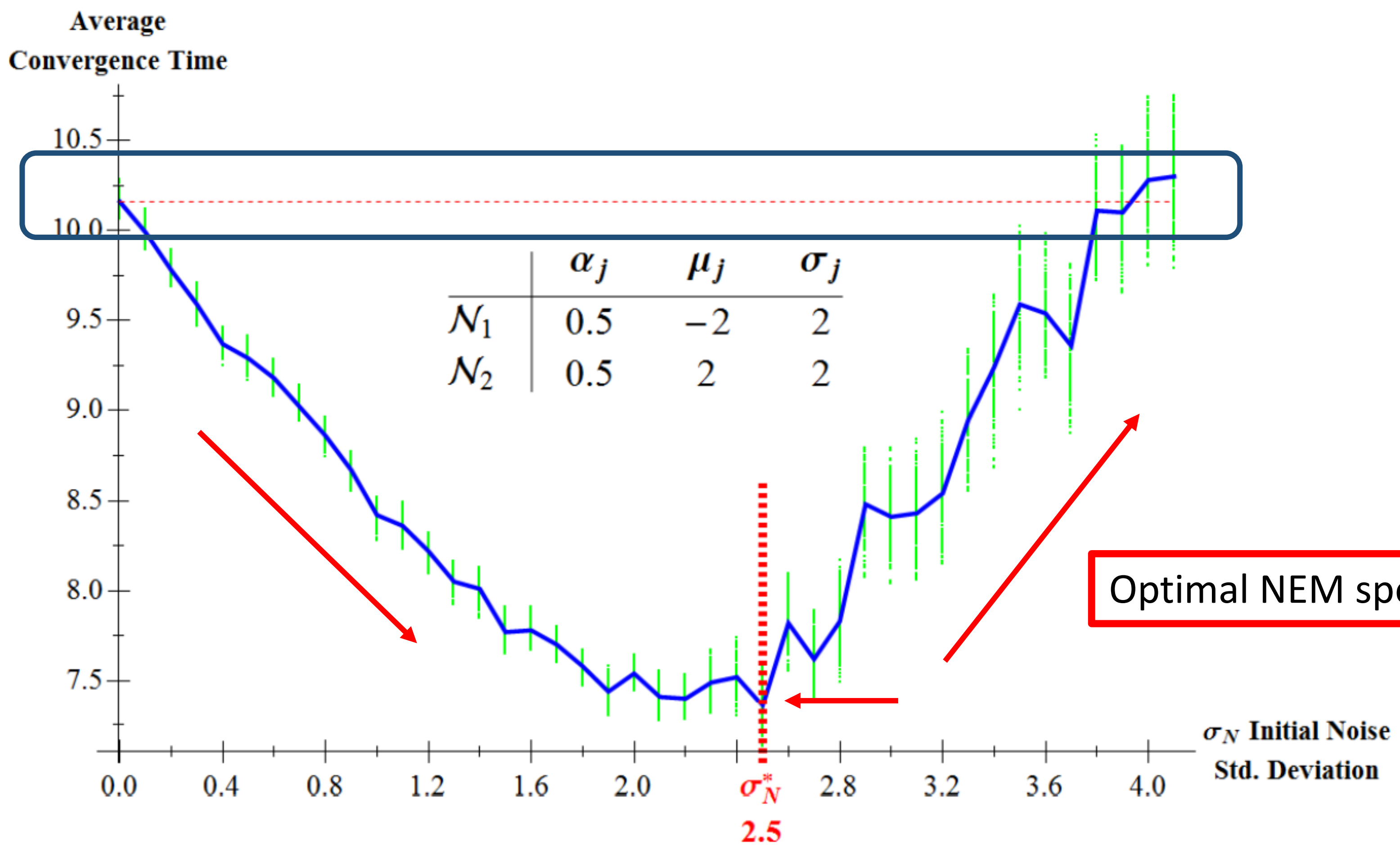
Solutions to GMM–NEM noise benefit condition: n locates in one of the following:

$$\boxed{\begin{aligned} N_j^-(y) &= [2(\mu_j - y), 0] \\ N_j^+(y) &= [0, 2(\mu_j - y)] \end{aligned}}$$

- Noise N tends to pull the data sample y away from the tails and towards the cluster of sub-population means (or locations)

[Osoba et al., 2013]

GMM–NEM Noise Benefit



	α_j	μ_j	σ_j
$\mathcal{N}_1$	0.5	-2	2
$\mathcal{N}_2$	0.5	2	2

Optimal NEM speed-up : 27.2%

[Osoba et al., 2013]

Part II

- Expectation Maximization

- Clustering
  - EM Clustering
  - k means
  - Arbitrary Noise Injection

- Markov Chains and HMM

EM Clustering

Assumptions:

- k clusters in the data
- $\theta_1, .., \theta_k$ are the pdf parameters for each cluster
- $\alpha_1, .., \alpha_k$ are the mixing proportions
- EM clustering uses the membership probability density function $p_Z(j|y, \Theta_{EM})$ as a maximum posteriori classifier for each sample y .
- The classifier assigns y to the j-th cluster if

$$\boxed{p_Z(j|y, \Theta_{EM}) \geq p_Z(k|y, \Theta_{EM})}$$

for all $k \neq j$.

[Osoba et al., 2013]

Noise Enhanced EM Clustering

$$EMclass(y) = \arg\max_j \boxed{p_Z(j|y, \Theta_{EM})}$$

$$p_Z(j|y, \Theta) = \frac{\alpha_j f(y|Z=j, \theta_j)}{f(y|\Theta)}$$

- For $\Theta = \{\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_K\}$

$$NEMclass(y) = \arg\max_j p_Z(j|y, \Theta_{NEM})$$

[Osoba et al., 2013]

K-means Clustering

k-means clustering is a special case of the GMM–EM model.

Theorem: The Expectation–Maximization Algorithm Subsumes k-Means Clustering.

[Celeux et al., 1992; Hathaway, 1986, Osoba et al.,2013].

Clustering Noise Benefit

Clustering Noise Benefit Theorem. Consider the NEM and EM iterations at the kth step. Then

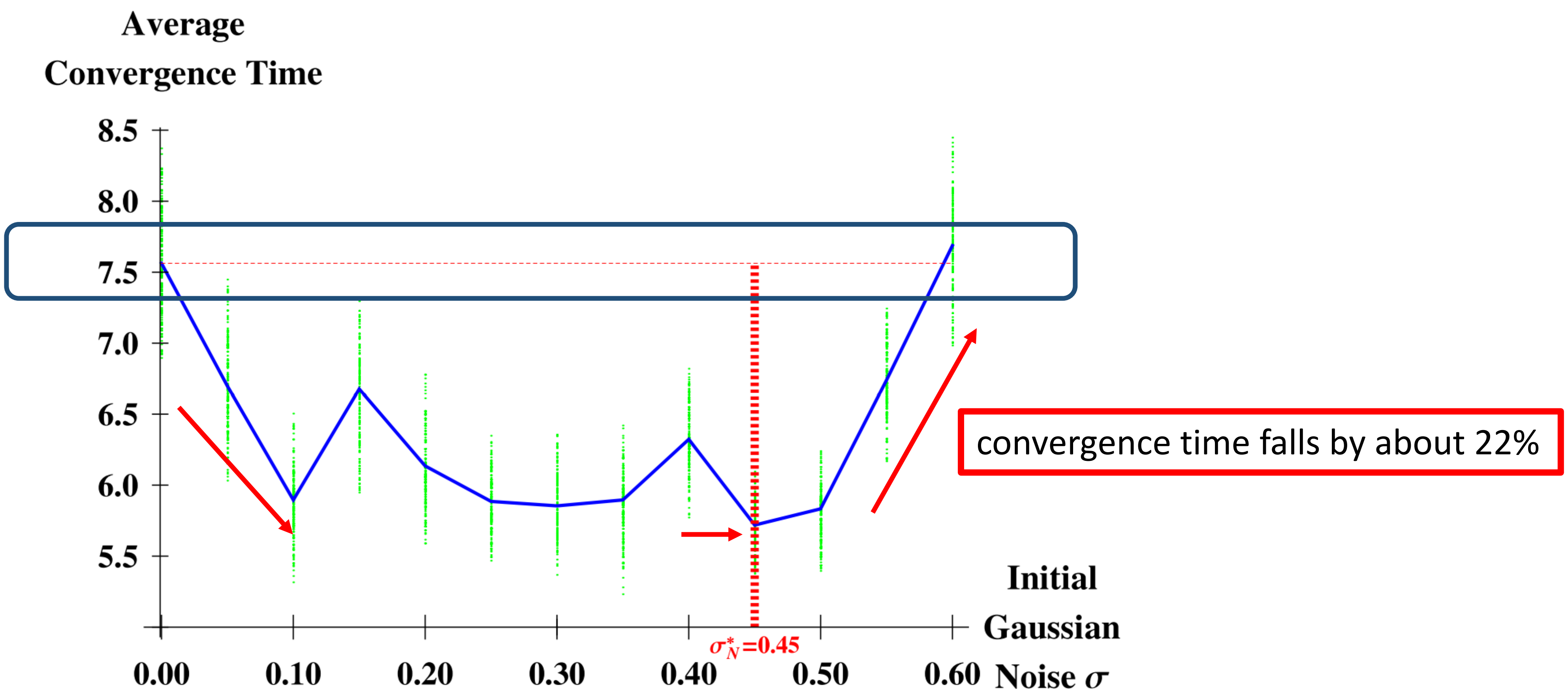
$$P_{M_N}[k] \leq P_M[k]$$

if the additive noise N in the NEM-clustering procedure satisfies the NEM Theorem condition from

$$E_{Y,Z,N|\theta_*}\left[\ln\left(\frac{f(Y+N,Z|\theta_k)}{f(Y,Z|\theta_k)}\right)\right] \geq 0$$

[Osoba et al., 2013]

Effect of Noise in KMEANS Convergence Time



[Osoba et al., 2013]

Arbitrary Noise Injection to NEM

Arbitrary Noise Injection NEM Theorem.

- Let $\phi(Y, N)$ be an arbitrary measurable mode of combining the signal Y with the noise N. Suppose the NEM average positivity condition holds at iteration k:

$$E_{Y,Z,N|\theta_*}\left[\ln\left(\frac{f(\phi(Y,N),Z|\theta_k)}{f(Y,Z|\theta_k)}\right)\right] \geq 0$$

  Then the EM noise benefit

$$Q(\theta_k|\theta_*)) \leq Q_N(\theta_k|\theta_*))$$

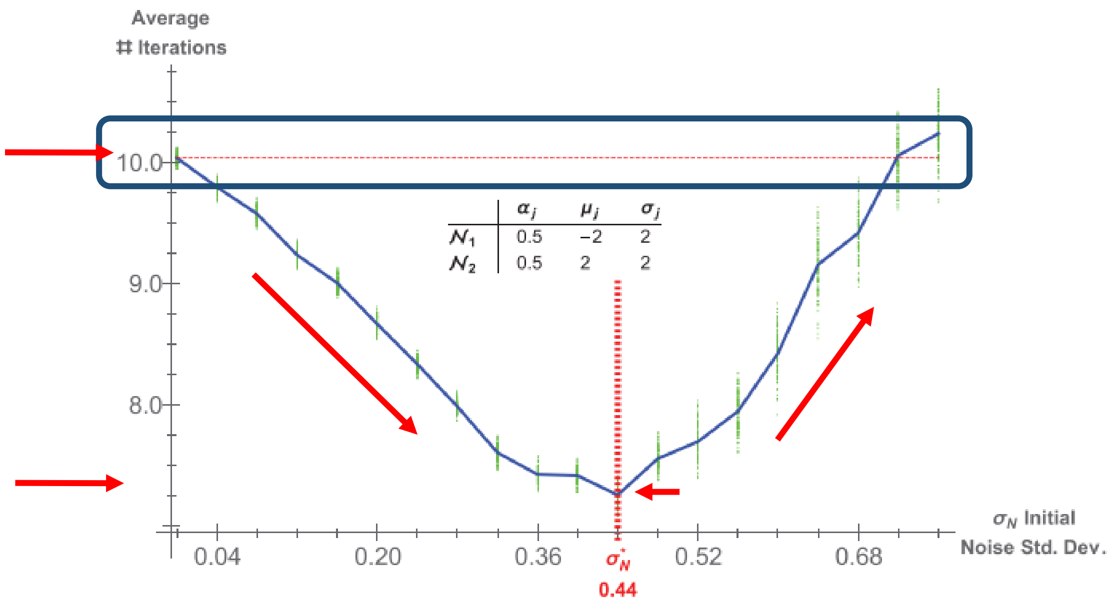
  Holds on average at iteration k:

$$E_{N,Y|\theta_k}[(Q(\theta_*|\theta_*) - Q(\theta_k|\theta_*))] \geq E_{Y|\theta_k}[(Q(\theta_*|\theta_*) - Q_N(\theta_k|\theta_*))]$$

[Osoba et al, 2016]

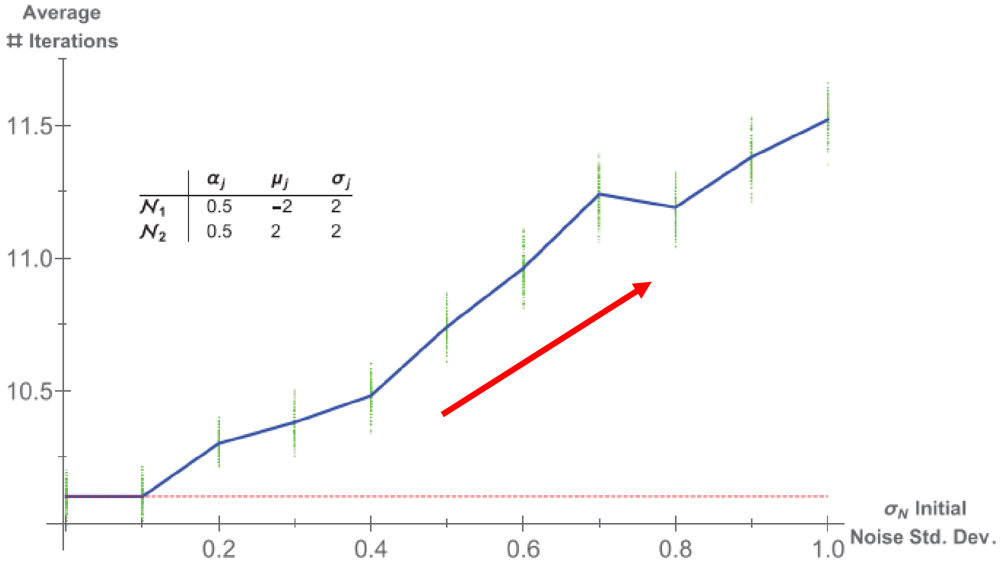
Multiplicative Noise on a GMM

Multiplicative noise (m-NeM) benefit



Optimal NEM
speed-up : 27.6%

Blind Multiplicative Noise



Increasement in the average number of iterations

[Osoba et al, 2016]

Part II

- Expectation Maximization

- Clustering

- Markov Chains and HMM
  - Markov Chains
  - Hidden Markov Models

Noise can speed convergence in Markov chain

- Assumptions
  - M is a finite time-homogeneous Markov chain
  - with N states
  - An irreducible and aperiodic state transition matrix P
  - a stationary vector $\widetilde{x}$

Markov Chain Noise Benefit Theorem

A noise benefit exists for all nonstationary state density vectors x in the sense that there exists some $A > 0$ so that for all $a \in (0, A)$

$$||[\widetilde{x}P - x^{\infty}]_i| < |[xP - x^{\infty}]_i|$$

for all states i with

$$\Delta_i = (x - x^{\infty})P_i > 0$$

where

$$\widetilde{x} = \frac{1}{1+a}(x + n)$$

Franzke et al., 2011

Markov Chain Noise Benefit Theorem

$$\widetilde{x} = \frac{1}{1+a}(x+n)$$

is the normalized state vector after adding a noise vector n with only one nonzero element

$$n_j = \begin{cases} a & j = k \\ 0 & j \neq k \end{cases}$$

for any k that satisfies $\Delta_k = (x - x^{\infty})P_k > 0.$

Franzke et al., 2011

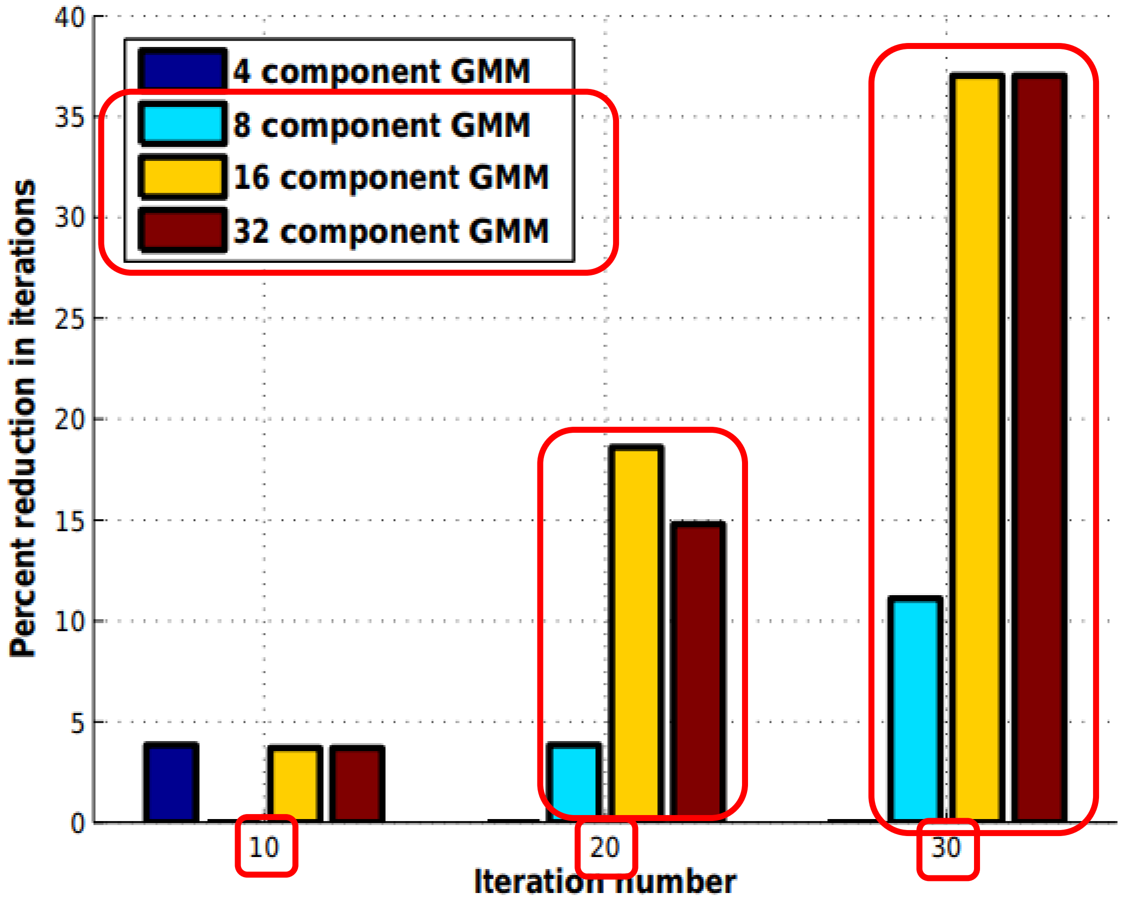
NEM Noise Benefit for Hidden Markov Models

- Hidden Markov Models (HMM): a probabilistic model for time series data
  - Speech processing and recognition

- NEM noise benefit : in HMM
  - Baum-Welch algorithm : a special case of the EM algorithm

Audhkhasi et al., 2013

Noise benefit in NHMM training



Audhkhasi et al., 2013

Part III

- Benefits of Noise in Neural Networks
  - Back Propagation
  - Classification
  - Regression

- Injecting Noise in Neural Networks

NEM Noise Benefit for Neural Networks

- Neural Networks also follow the NEM benefits.
    - Backpropagation algorithm : maximum likelihood estimation of a neural network's parameters

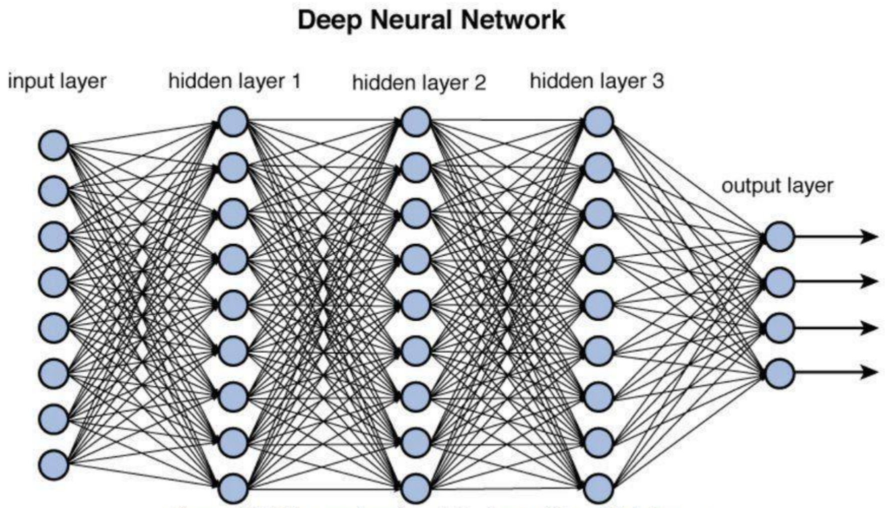
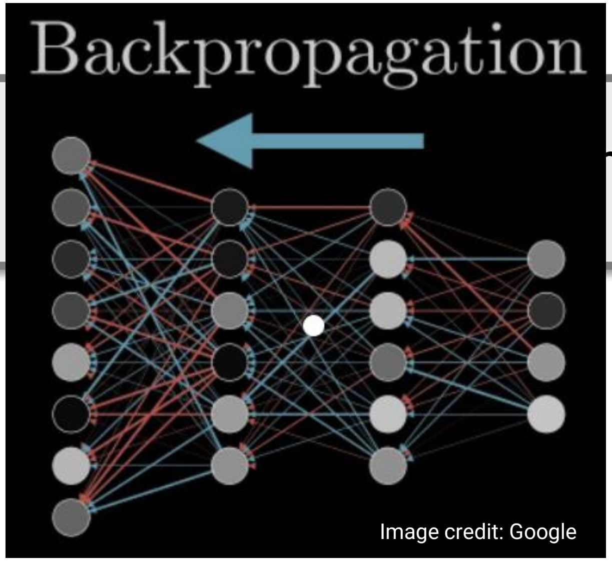


Figure 12.2 Deep network architecture with multiple layers.

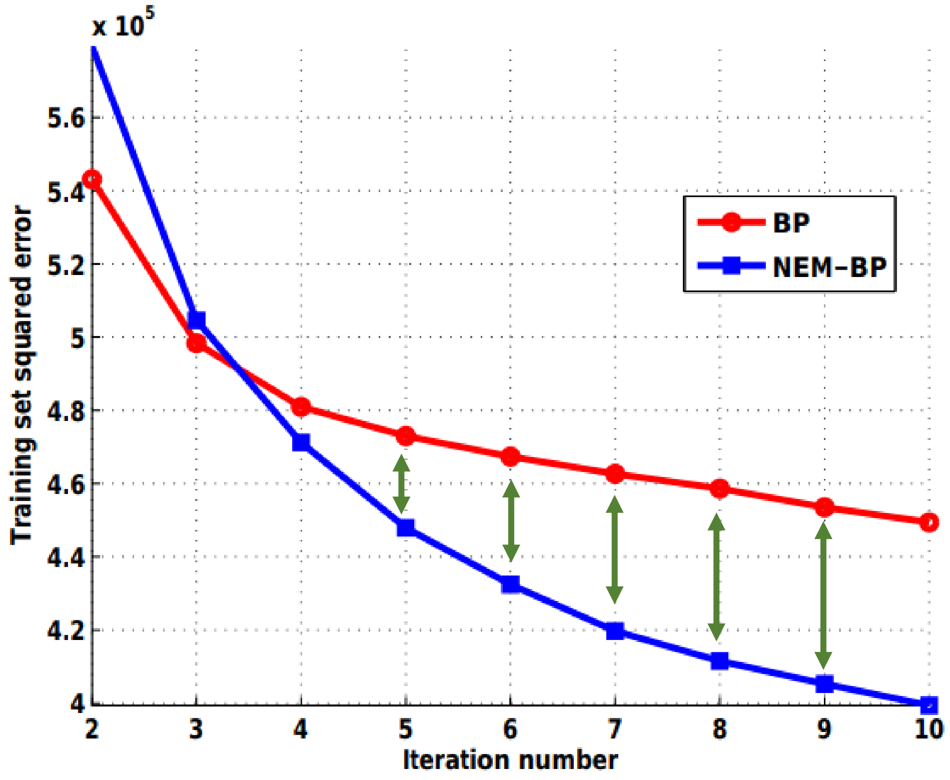
Image credit: Parmar 2018

the hm



[Audhkhasi et al., 2013]

NEM Noise Benefit for Neural Networks



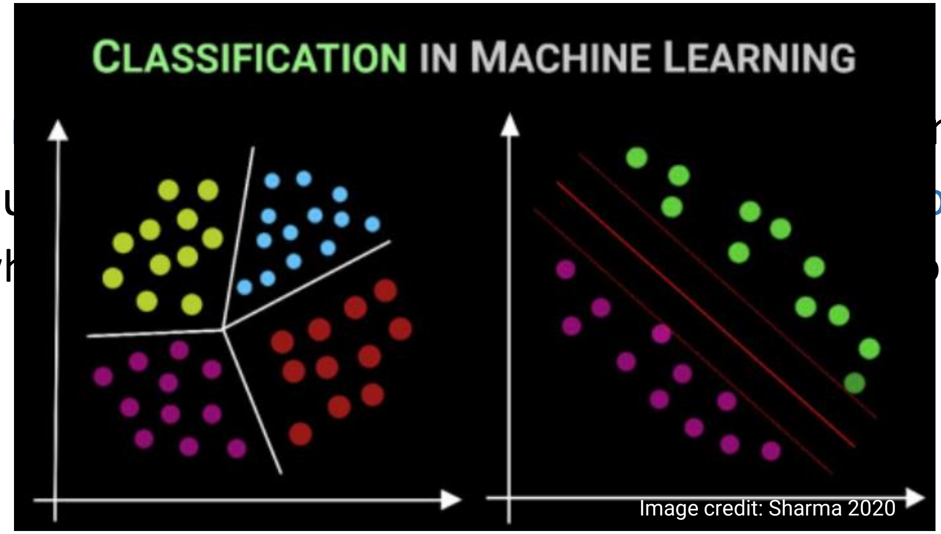
5.3% decrease in the squared error per iteration for NEM-BP

Audhkhasi et al., 2013

Classification

- Noise benefit sufficient condition for Gibbs or Softmax activation output neurons used in K-class classification

  - Assumptions:
    - x and h : … n layer.
    - y is the K-valu … oding.
      - $t_k = 1$ wh … otherwise.



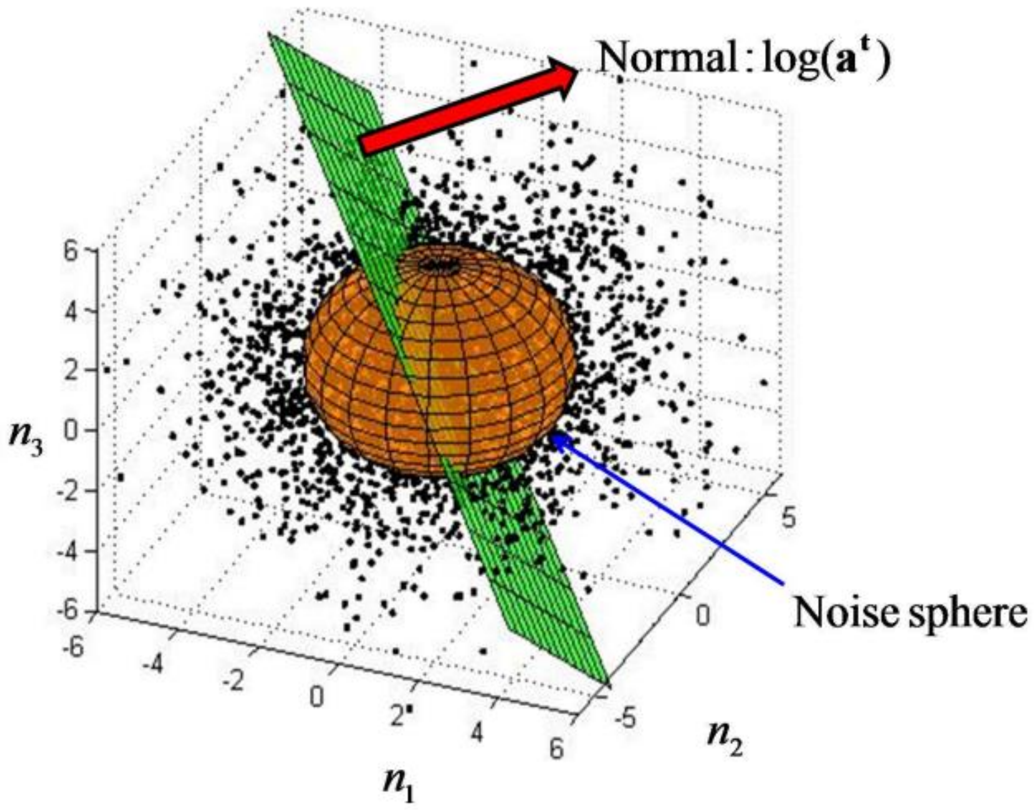
Hyperplane Noise Benefit Condition for Feedforward Neural Networks

The NEM Theorem condition is satisfied for Maximum likelihood training of feedforward neural network with Gibbs or SoftMax activation output neurons if

$$E_{t,h,n|x,\Theta_*}\left\{n^T \log(a^t)\right\} \geq 0$$

where $\log(a^t)$ denotes the vector of log-activation of output neurons.

Hyperplane Noise Benefit Condition for Feedforward Neural Networks



Audhkhasi et al., 2013

Regression

- Noise benefit sufficient condition for Gaussian output neurons used in regression networks.
    - A spherical noise-benefit region in noise space for a Gaussian target data vector $t \sim N(t|a^t, I)$.

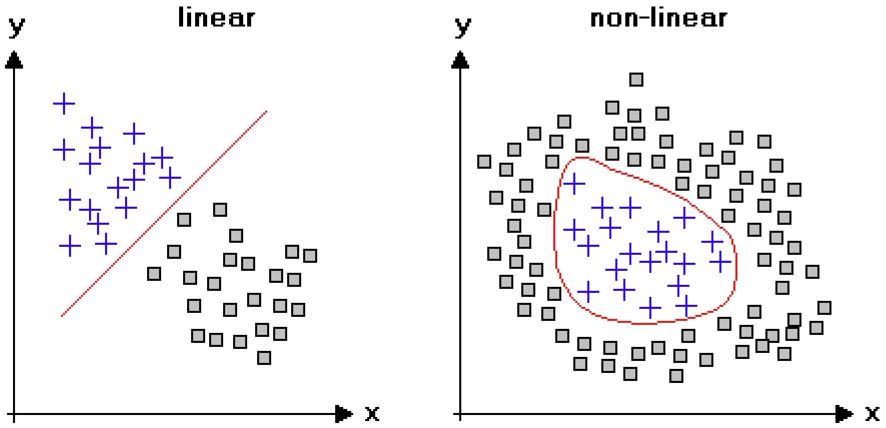


Image credit: Scott 2018

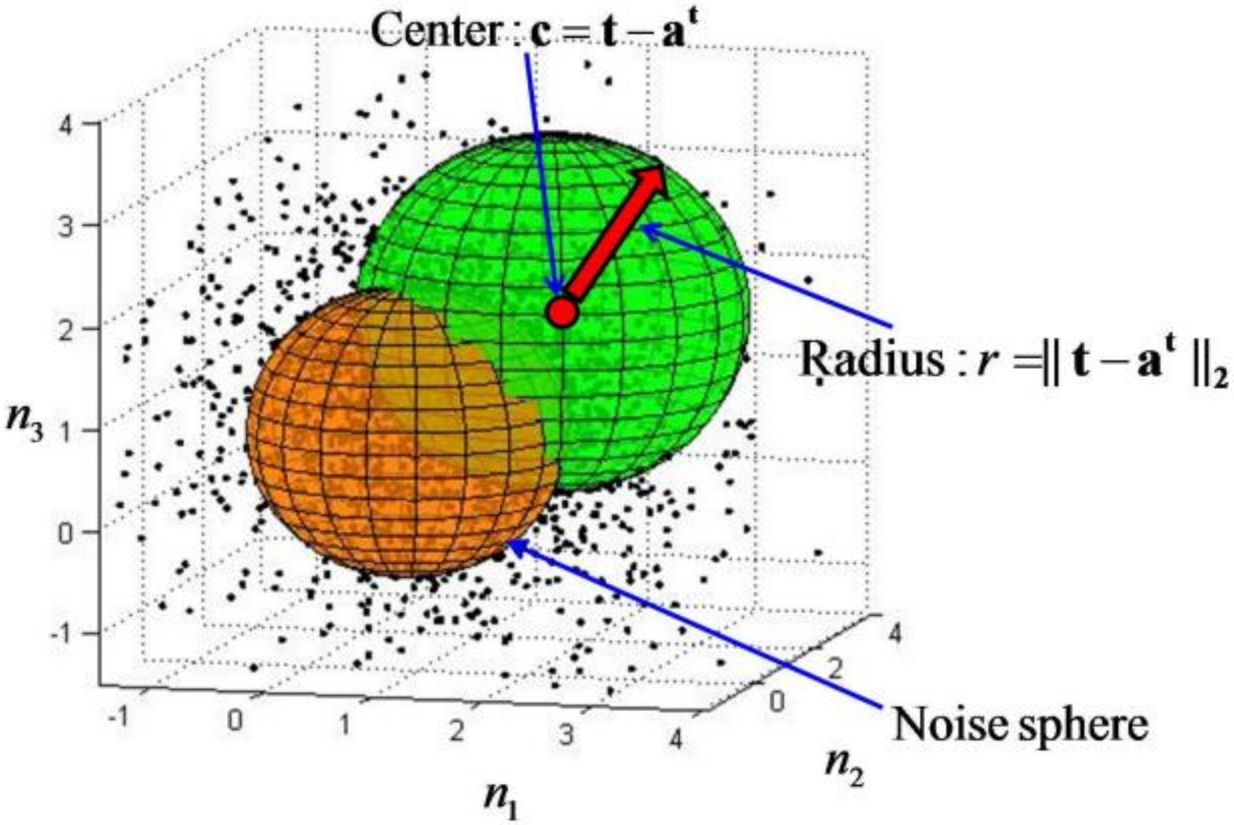
Sphere Noise Benefit Condition for Feedforward Neural Networks

The NEM Theorem condition is satisfied for Maximum likelihood training of a feedforward neural network with Gaussian output neurons if

$$E_{t,h,n|x,\Theta_*}\left\{ \left\|n - a^t + t\right\|^2 - \left\|a^t - t\right\|^2 \right\} \leq 0$$

where $\|.\|$ is the $L2$ vector norm.

Sphere Noise Benefit Condition for Feedforward Neural Networks



Audhkhasi et al., 2013

Part III

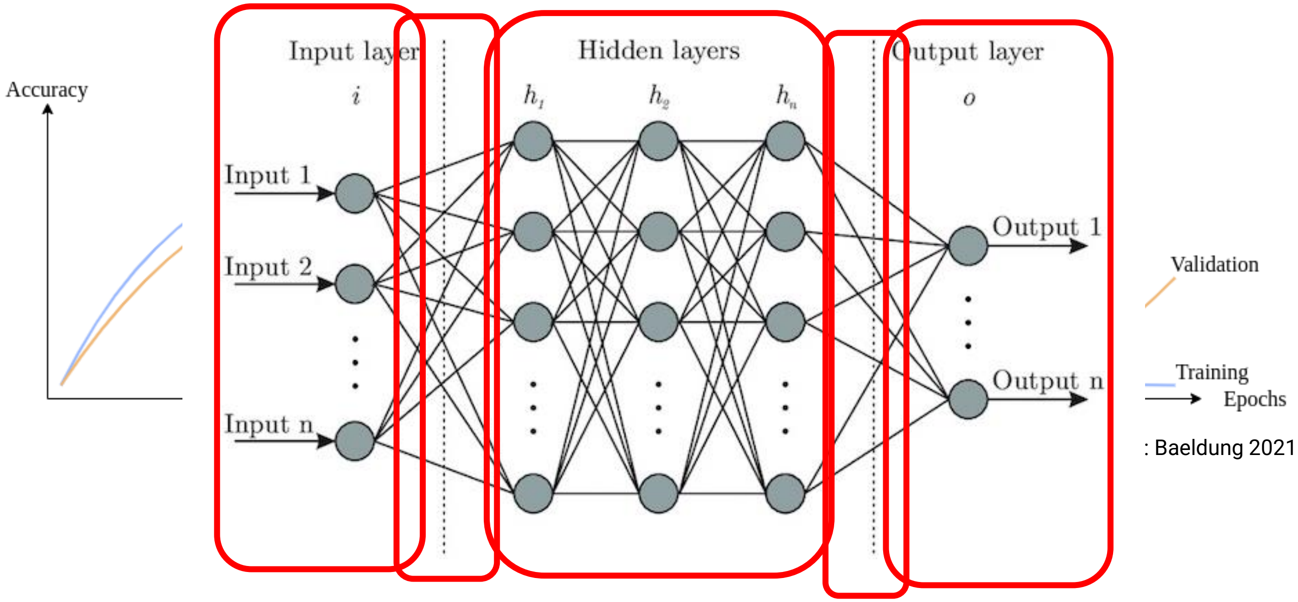
- NEM Noise Benefit in Neural Networks

- Injecting Noise in Neural Networks
  - Noise Injection to Inputs
  - Noise Injection for network Weights
  - Noise Injection for network Gradients

Training with Noise

Some methods indicates convergence as a cost for improved generalization performance

- Some techniques **improve both metrics** at the same time



: Baeldung 2021

Noise Injection for Input Training

- Regularization: Controlling the tradeoff between bias and variance
  - Improve generalization
  - Adding a random vector into the input data : Tikhonov regularization

- Adding noise to the training data of feedforward neural network
  - Better estimate the optimal weight vectors

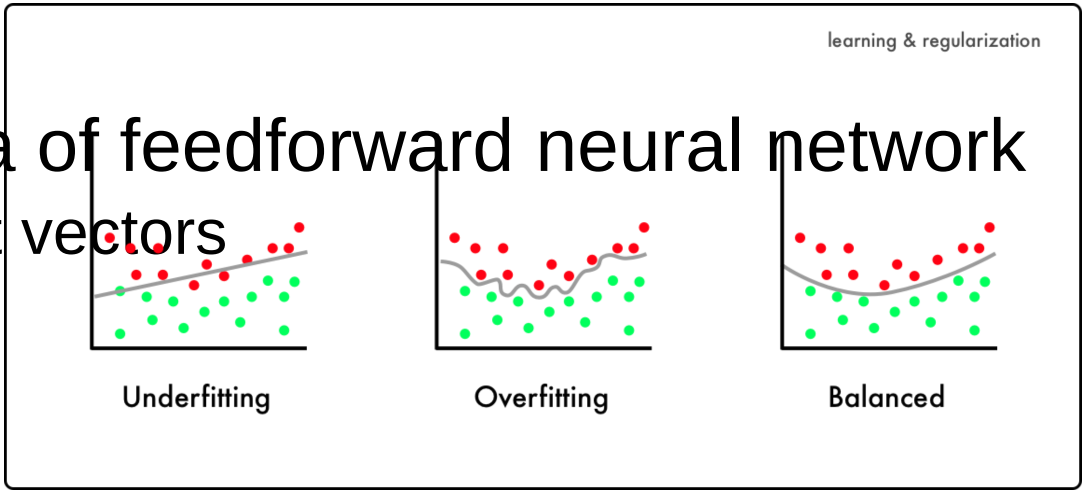


Image credit: Wallner 2017

[Bishop et al., 1995, Holmstrom 1992]

Data Augmentation

- Adding noise to the **inputs of neural networks**: data augmentation.
  - Small Input : adding noise to the original dataset
  - overcoming the problem of training on less data for a specific class.

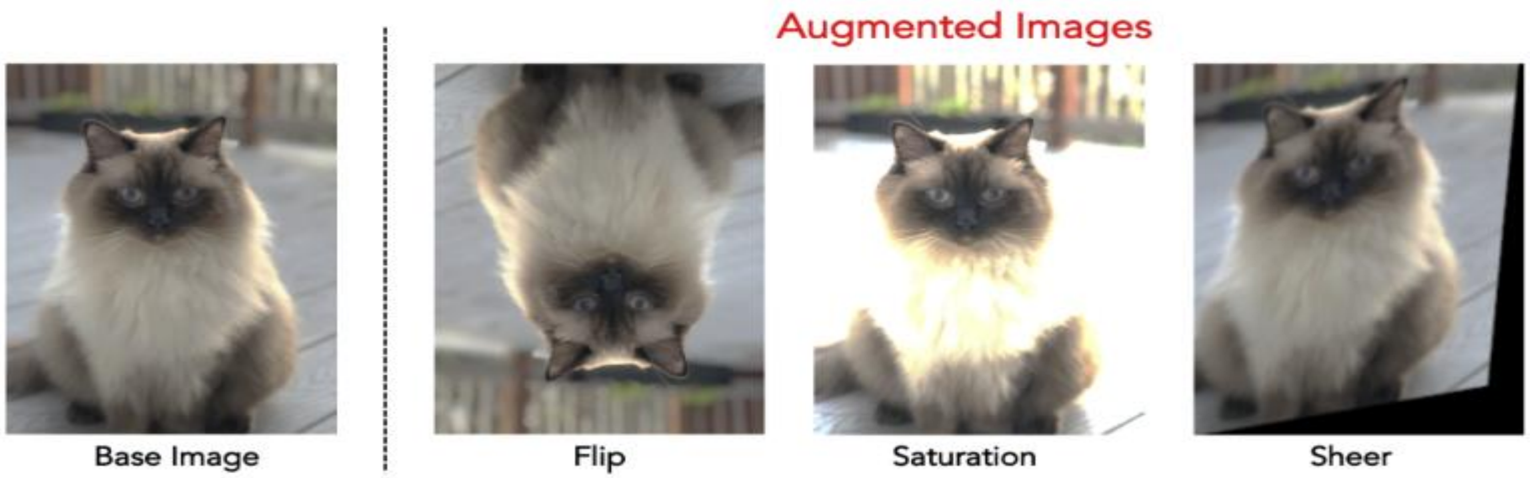


Image credit: Segura

[Goodfellow et al., 2016]

Noise Injection for Network Weights

- Supervised neural network generalize well
  - Amount of information in the weights be less than information in the output training vectors
  - Handling the amount of information in a weight : **adding Gaussian noise** to it

- Weight noise **simplifies** the ne
  - Reduces the information requir
  - Generalization improvement

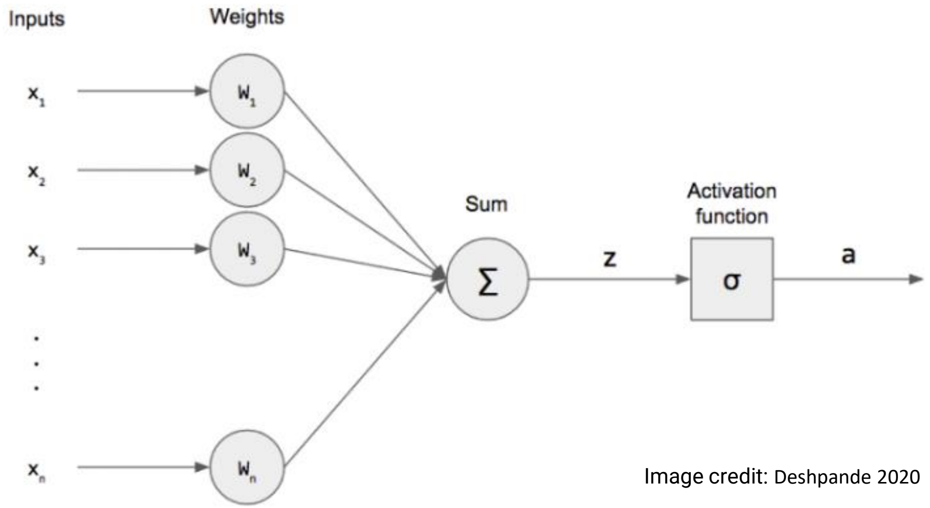


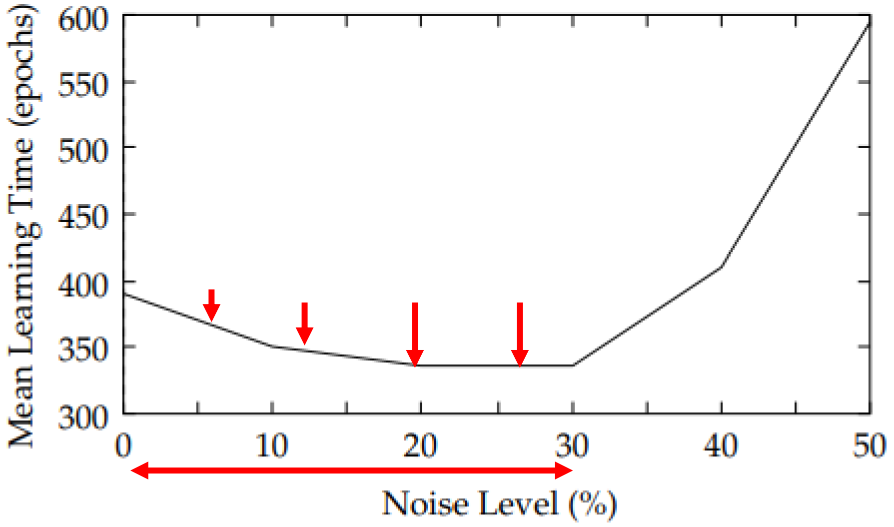
Image credit: Deshpande 2020

[Hinton et al., 1993]

Noise Injection for Network Weights

Simulation of synaptic noise on multilayer perception training

- Applying noise to the weights of each layer during training
- Noise benefits : improving **fault tolerance**, and **generalization performance**

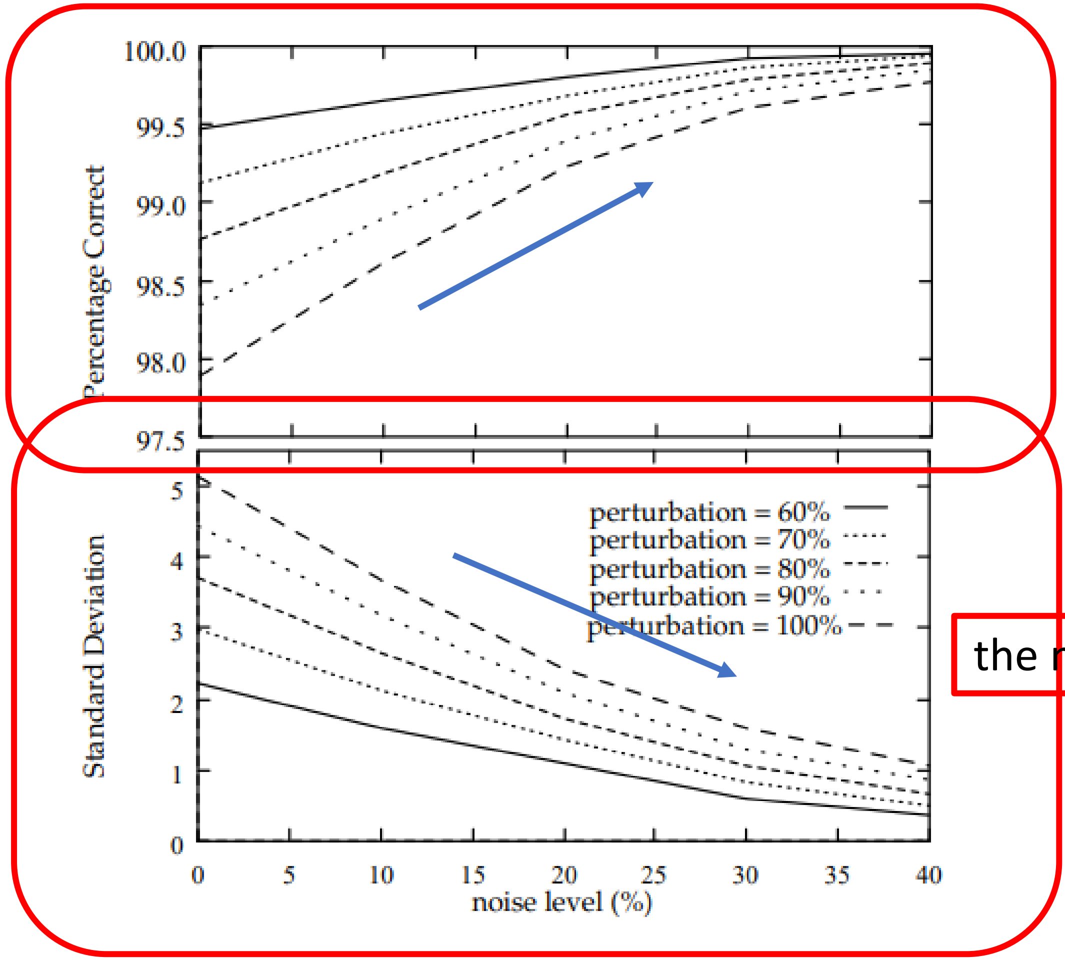


[Murray & Edwards, 1994]

Weight Perturbation

Classification

Variations



the network is less susceptible to damage

[Murray & Edwards, 1994]

Noise Injection for Network Gradients

- Adding annealed Gaussian noise to the gradient during training
  - Help training and generalization of complicated neural networks
    - **avoid overfitting**
    - **decrease the training loss** : motivating active exploration of parameter space

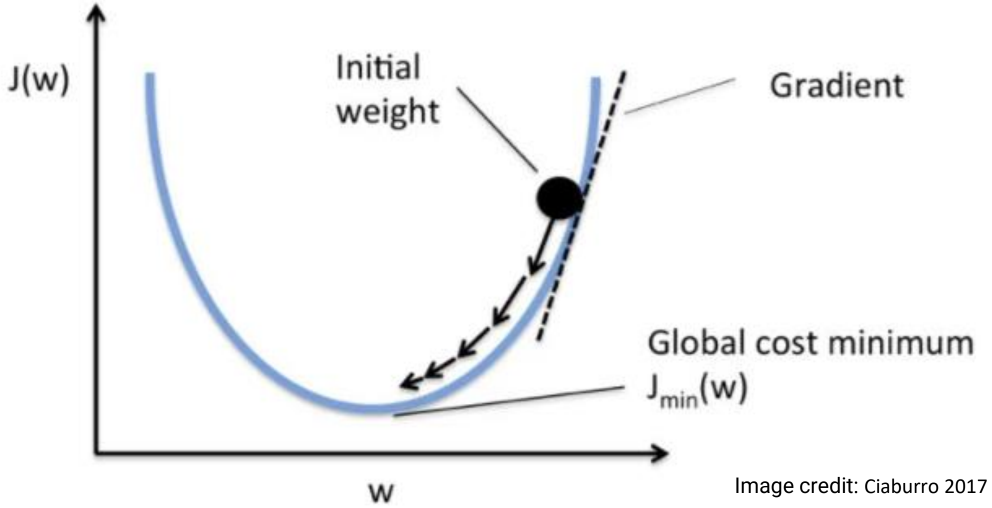
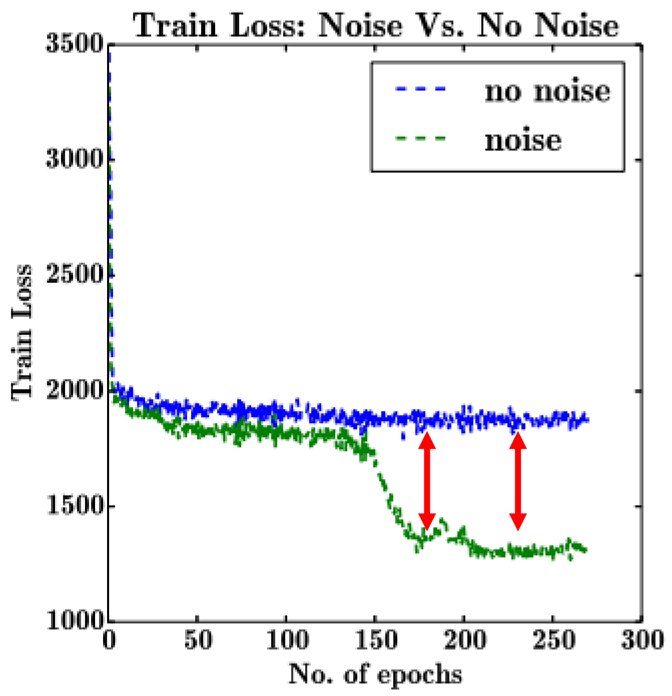
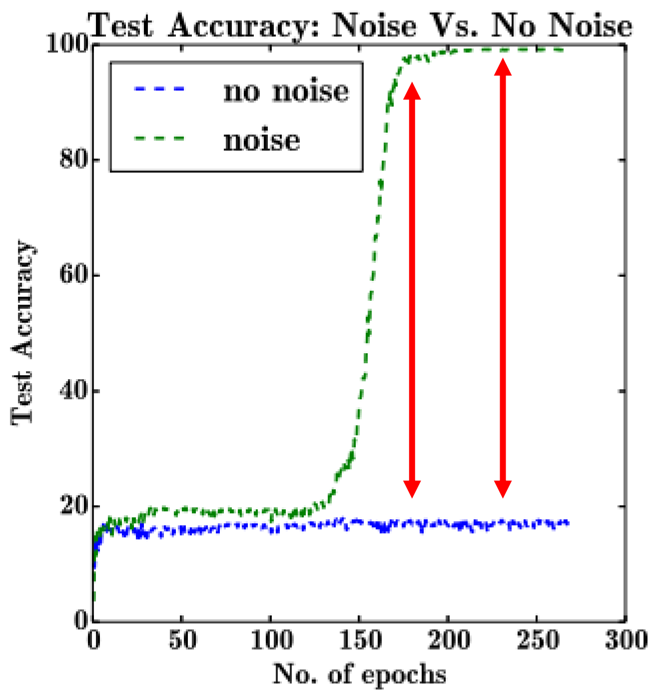


Image credit: Ciaburro 2017

[Neelakantan et al., 2015]

Gradient Noise





[Neelakantan et al., 2015]

Noise-Enhanced Optimization

- Optimization : benefit from randomness
- Random search optimization techniques may get trapped in local minima while searching for an optimum.
  - **Randomization** can help for searching the coarse regions of state space before searching finer regions

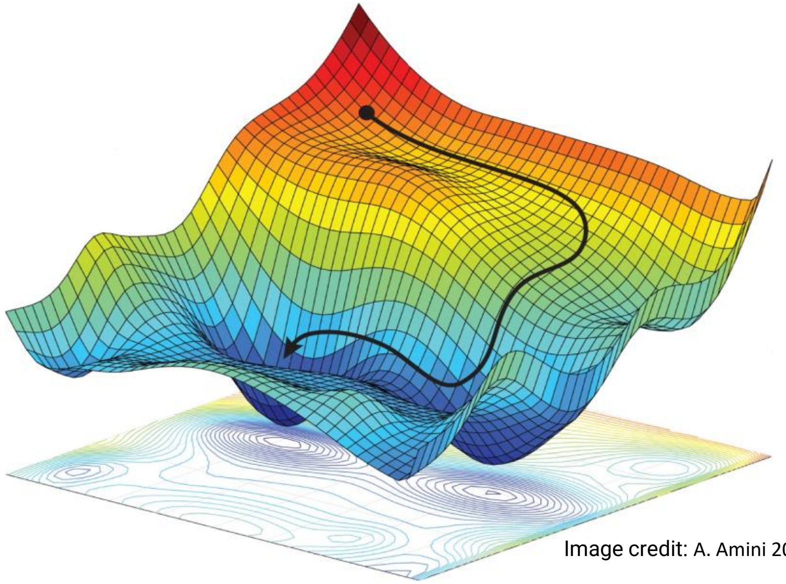


Image credit: A. Amini 2018

Genetic Algorithm (GA)

- Randomization in **crossover** and **mutation**
  - Helps avoid self-similarity in the population by avoiding local minima.

- The role of **mutation** is similar to **adding noise**
  - A suitable mutation rate can improve performance

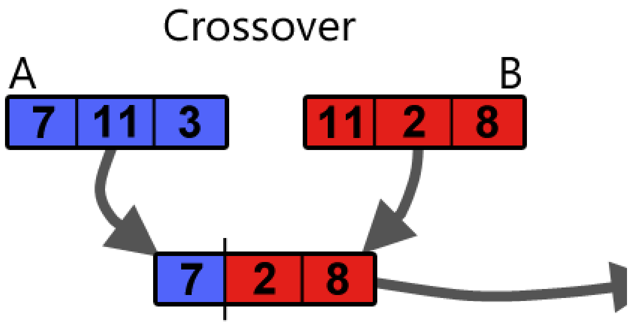


Image credit: Fernández 2017

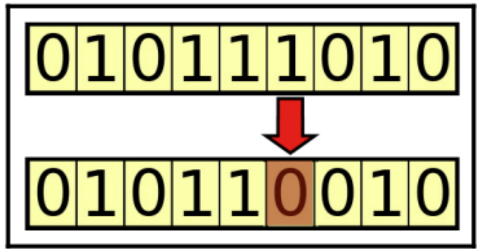


Image credit: Rajan 2021

[Tang et al., 1996, Chen et al., 2014]

Simulated Annealing (SA)

- Improving convergence

  Combinational problems like graph partitioning

- A temperature parameter T :
  - handle the **randomness** of the search procedure
  - The T progressively **decreases**
  - The higher the T, the higher chance

    SA accept a solution worse than the current one

- This **randomness in SA : noise**

```
Begin
Choose The best initial solution (S1)
Choose an initial temperature (T0)
Repeat
    While (M<m)
        S2=Generate a neighbor of the solution S1
        Delta= Objective(S1)- Objective(S2)
        If  Delta<0 Then
        S1:=S2
        Else if exp(Delta/K*T) > Random(0~1)  Then
        S1:=S2
        End if
        M:=M+1
    End while
        T=T*alpha
Until T < Tend
End
```

Image credit: Babaei 2019

[Kirkpatrick et al., 1983, Chen et al., 2014]

Part IV

- Network Science
  - Network Connectivity
  - Community Detection
  - Link Prediction
- Privacy Preservation
- Natural Language Processing

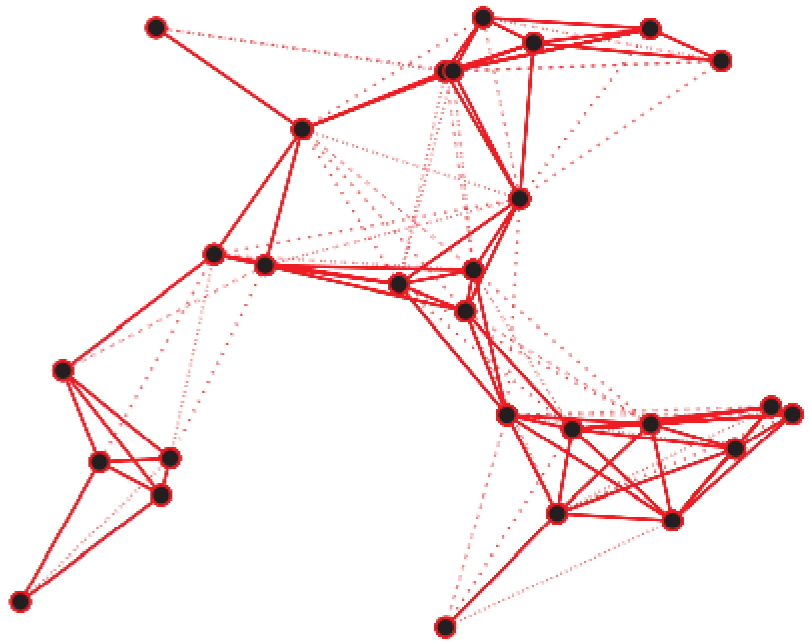
Noise-Enhanced Graph Connectivity

Adding edges to the graphs : increase the algebraic connectivity

Algebraic connectivity of the graph

↓

Second smallest eigenvalue $\lambda_2(L)$



[Gosh & Boyd 2006]

Noise-Enhanced Graph Connectivity

Assumptions:

- $G_{base} = (V, E_{base})$: a base graph
- m_c : a set of candidate edges E_{cand} on V
- $k\ (0 \leq k \leq m_c)$: the number of edges that results in the greatest increase in algebraic connectivity when added to G_{base}

$$\text{maximize } \lambda_2(L(E_{base} \cup E))$$

$$\text{subject to } |E| = k, E \subseteq E_{cand}$$

[Gosh & Boyd 2006]

Noise-Enhanced Graph Connectivity

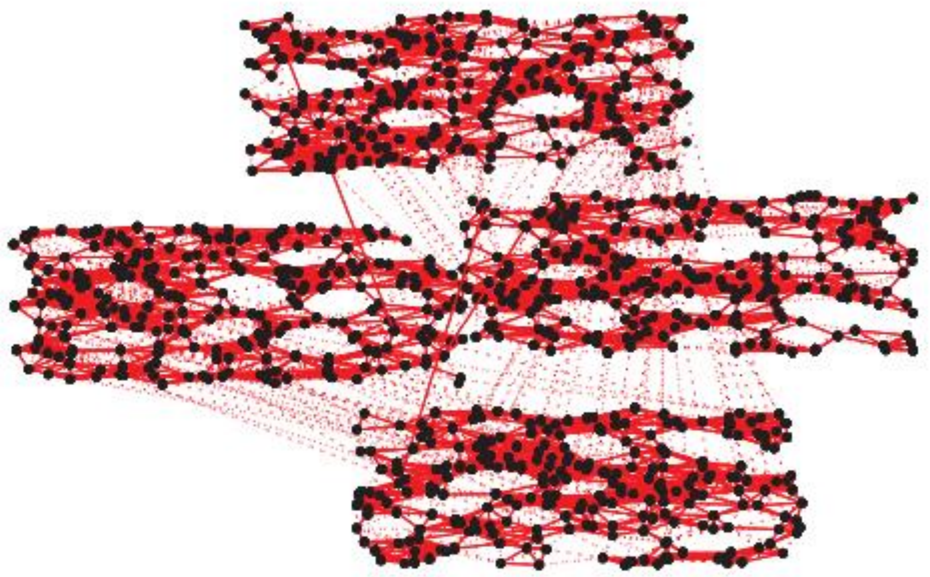
A greedy local heuristic solution :

1) find a unit eigenvector, v , corresponding to $\lambda_2(L)$ (L is the current Laplacian),
2) add an edge $l \sim (i, j)$ with the largest value of $(v_i - v_j)^2$ to the graph.

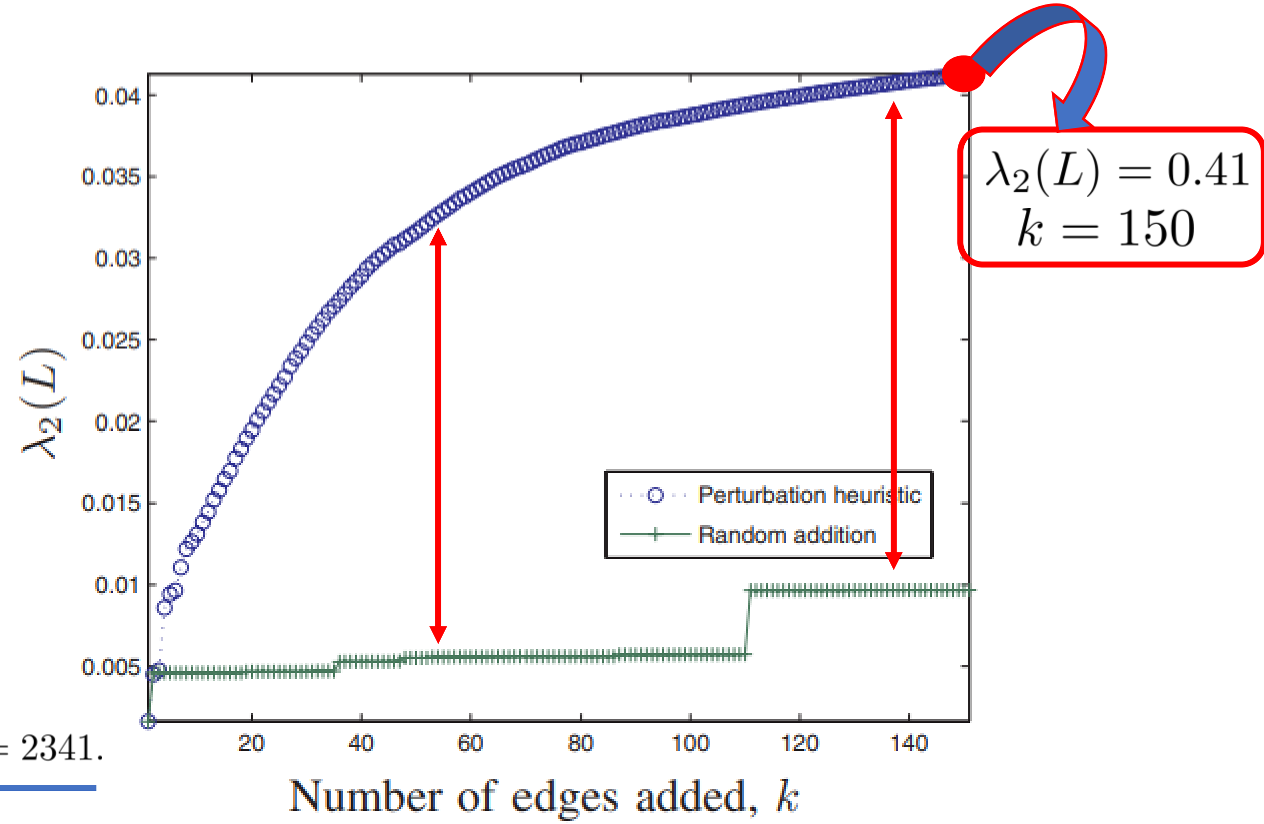
This process continues until k edges are added to the graph.

[Gosh & Boyd 2006]

Effect of Adding Edges on Algebraic Connectivity



A randomly generated graph with 1000 nodes, 5517 edges, and $m_c = 2341$.



[Gosh & Boyd 2006]

Noise-Enhanced Graph Connectivity

Perturbation heuristic adds edges

1) Connects two connected components
2) Links nodes that most strongly belong to different components
   - Large $(v_i - v_j)^2$
3) Connects nodes that are farthest from each other in the linear embedding

[Gosh & Boyd 2006]

Noise Enhanced Community Detection

- Design new community detection algorithms
  - **Alternative**: keep the algorithm but modify the input data: **network**
- Motivation: **Noise benefits**
  - Modify data: introduce **noise**
- Introduce noise in a network: **add edges**

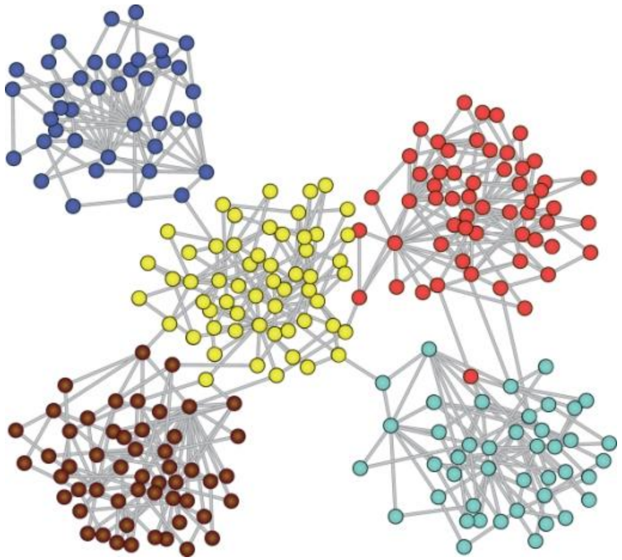
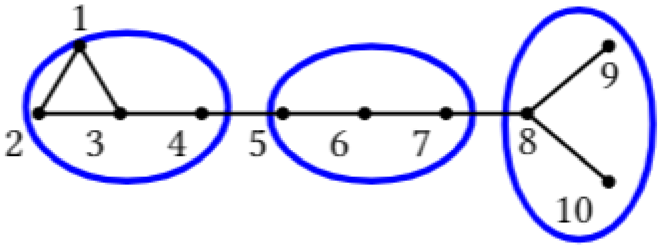


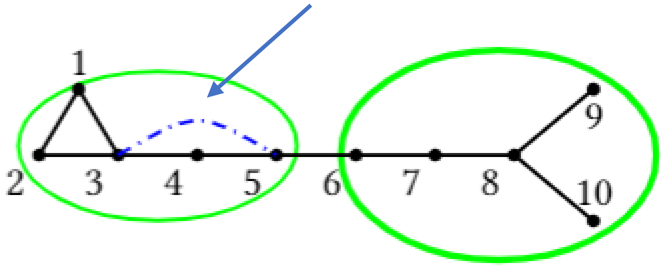
Image credit: Rozemberczki 2020

[Abdolazimi & Zafarani 2020]

Community Detection



(a) original graph



(b) noisy graph

add a **single** noise edge (3, 5)

evaluate communities: edge cut=1.3

- The same community detection method
  - **fewer** communities
  - **better** communities: edge cut=1

%30 decrease in edge cut

[Abdolazimi & Zafarani 2020]

Noise Injection Framework

1. Some noisy edges are added to the graph.
2. Communities are identified in the noisy graph using the same community detection algorithm
3. The detected communities are evaluated using an objective function.

Preprocessing

1. Sort nodes based on their degrees,
2. Select the top p percent of sorted nodes as **candidates**, and
3. Add edges within candidates

Noise Injection Methods

1) **Random Noise**: randomly choose from the candidates

2) **Weighted Noise**: degrees of candidates

3) **Frequency Noise**: degree distribution of the candidates

[Abdolazimi & Zafarani 2020]

Experimental Setup

- **Evaluation Metrics:**
  1) Expected First Success (EFS)

     Expected number of times that we need to add noise to the network to ensure that we improve communities at least once.

     Example:
     - total 100 iterations
     - improve communities in 34 iterations
     - EFS=3

  2) Relative Objective Improvement (ROI)

     How much we improve the objective function relatively.

     Example:
     - $Obj_{original} = 10$
     - $Obj_{new} = 11$
     - ROI=10%

[Abdolazimi & Zafarani 2020]

Theoretical Analysis

- **Spectral Analysis**

$$m_2 = E(d_i)E(\frac{1}{d_i d_j})$$

- Theorem: Let graph G'=(V',E') be obtained from graph G=(V,E) by connecting nodes u and v. Then,

$$m_2' - m_2 = \frac{2}{(d_u+1)(d_v+1)} \cdot (1 - E_{x:x\sim u}(\frac{d_v+1}{d_x}) - E_{y:y\sim v}(\frac{d_u+1}{d_y}))$$

Two disconnected high degree nodes are more likely to be in **different subsets** of the minimum cut.

Our noise-enhanced method increases the chance of **re-partitioning**

[Abdolazimi & Zafarani 2020]

Theoretical Analysis

Objective Function Analysis

Graph G with communities $S1$ and $S2$

$$d_i = d_{i,in} + d_{i,out}$$

1. Theorem **Modularity** Change. If $d_{i,in} < d_{i,out}$, moving v_i from S_1 to S_2 increases modularity.
2. Theorem **Edge cut** Change. If $d_{i,in} < d_{i,out}$, moving v_i from S_1 to S_2 decreases edge cut.
3. Theorem **Conductance** Change. If $d_{i,in} < d_{i,out}$, moving v_i from S_1 to S_2 decreases conductance.
4. Theorem **Normalized Cut** Change. If $d_{i,in} < d_{i,out}$, moving v_i from $S1$ to S_2 decreases cut size.

[Abdolazimi & Zafarani 2020]

Noise-enhanced Community Detection in Real-World Networks

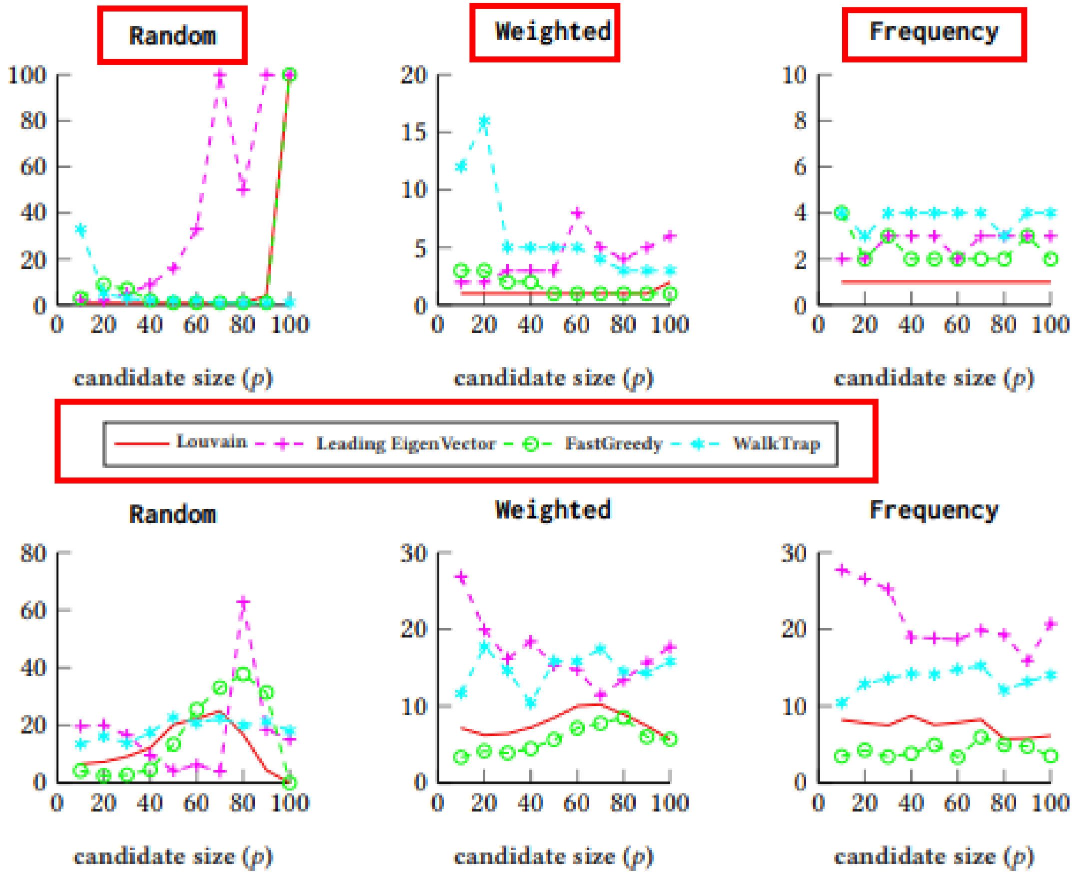
↓ EFS

Expected First Success ($EFS_{conductance}$)

↑ ROI

Relative Objective Improvement ($ROI_{Conductance}$)

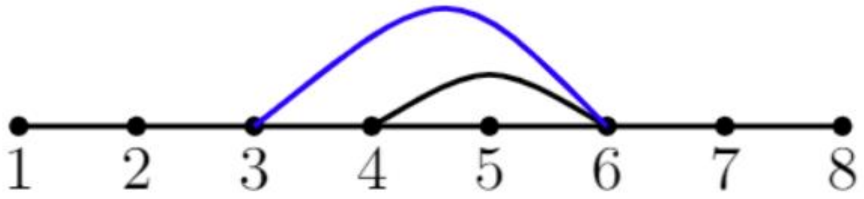
Conductance of communities detected in HepPH



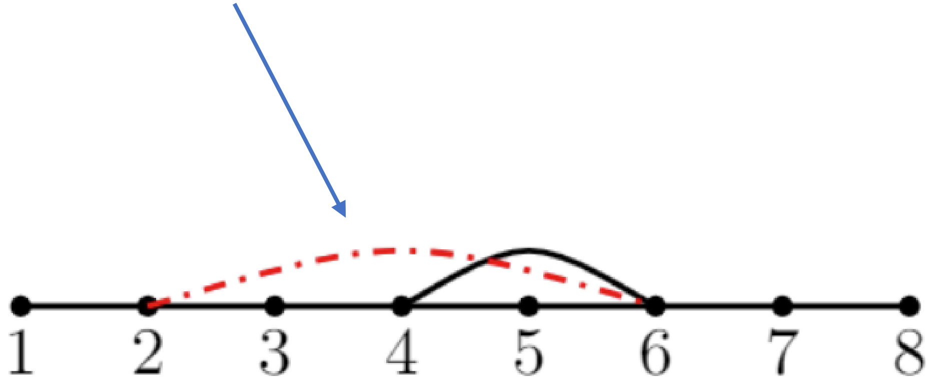
$EFS = 3$

[Abdolazimi & Zafarani 2020]

Noise-Enhanced Link Prediction



(a) original graph



(b) noisy train graph

add a **single** noise edge (2, 6)

- Evaluate the predicted links:
  Average Precision=0.14
  ROC=0.76

- Adamic/Adar measure:
  - **increasing** accuracy of the predicted links
  - **better** ranking on the edges predicted

250% increase in *AP*
23% increase in ROC

[Abdolazimi & Zafarani 2021]

Experimental Setup

- **Synthetic Datasets**

Graph Model	**Graph size (n)**	**Parameters**
$\boxed{\textbf{Random}(n, p)}$	$1,000$	$\boxed{p \in \{0.001, 0.003, 0.006, 0.007\}}$
$\boxed{\textbf{Small-world}(n, k, p)}$	$1,000$	$\boxed{p \in \{0.0001, 0.001, 0.01, 0.1, 1\},\ k = 10}$
$\boxed{\textbf{Configuration}(deg - seq)}$	$1,000$	$\boxed{\text{powerlaw } deg - seq}$

Theoretical Analysis

Theorem **Common Neighbor** Score Change. Connecting two High Degree nodes can increase common neighbor score the most.

Theorem **Adamic/Adar** Score Change. Connecting two HighDegree nodes can increase the Adamic/Adar score the most

Theorem **Preferential Attachment** Score Change. Connecting twoHighDegree nodes can increase the preferential attachment score the most.

Theorem **Katz** Score Change. linking two HighDegree nodes increases Katz score the most.

Theorem 1 (Common Neighbor Score Change). *Connecting two* HighDegree *nodes can increase common neighbor score the most.*

Proof. There are two ways to look at this. First, consider adding edges between pairs of nodes to increase the common neighbor score. How we can increase $CN(i, j)$? This increase is possible by (1) connecting neighbors x of i to j, i.e., $x \sim j$, such that $x \in \Gamma(i)$; or (2) neighbors y of j to i, i.e., $y \sim i$, $y \in \Gamma(j)$ [illegible] select neighbors x and y such that $x, y \in$ HighDegree, more nodes find common neighbor with j and i. Similarly, if $j, i \in$ HighDegree, we will also increase the common neighbor of more nodes (i.e., i and j now act as the common neighbors). Therefore, connecting two HighDegree nodes, $x \sim j$ and $y \sim i$, increases common neighbor scores among more node pairs in G.

Secondly, we can show this property relatively compared to edges between lower degree nodes. Consider $i, j \in$ HighDegree and $i', j' \notin$ HighDegree, then $i \sim j$ increases common neighbor scores more compared to $i' \sim j'$. This is because the edge $i \sim j$ increases score for all $CN(k, j)$, $k \in \Gamma(i)$ and $CN(i, l)$, $l \in \Gamma(j)$. Similarly, edge $i' \sim j'$ increases $CN(k', j')$, $k' \in \Gamma(i')$ and $CN(i', l')$, $l' \in \Gamma(j')$. Since $|\Gamma(i)| > |\Gamma(i')|$ and $|\Gamma(j)| > |\Gamma(j')|$, $i \sim j$ can increase the common neighbor score between more nodes in G compared to $i' \sim j'$.

[Abdolazimi & Zafarani 2021]

Noise-enhanced Link Prediction in Real-World Networks

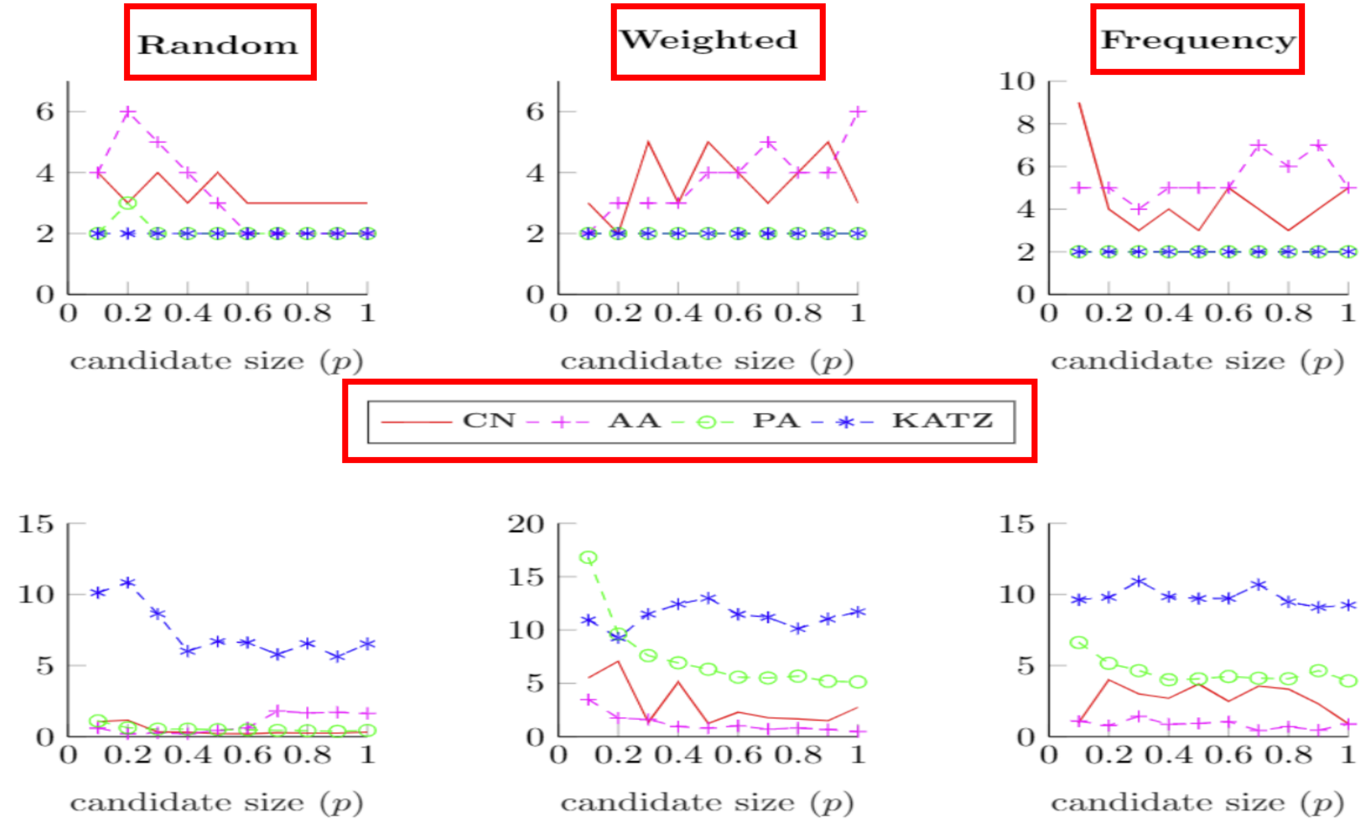
↓ EFS

Expected First Success (EFS_{PR})

Precision-Recall of predicted Links in Bitcoin

↑ ROI

Relative Performance Improvement (RPI_{PR})



$EFS = 3$

$RPI = 4.2$

[Abdolazimi & Zafarani 2021]

Noise-enhanced Link Prediction in Synthetic Networks

1. Which link prediction measure works best under noise?

2. Which noise-enhanced measure performs the best for each network model?

3. Which type of network model in general yields better results? Which one has the worst results?

[Abdolazimi & Zafarani 2021]

Which type of noise works best for each network model?

Models	**Parameters**	**Noise Priority Ranking**		
Random(n,p)	all p	Random(1^{st})	**Weighted**(2^{nd})	Frequency(3^{rd})
Small-world	$p <= 0.1$	Frequency(1^{st})	**Weighted**(2^{nd})	Random(3^{rd})
	$p = 1$	Random(1^{st})	**Frequency**(2^{nd})	**Weighted**(2^{nd})
Configuration	Powerlaw	Weighted(1^{st})	**Frequency**(2^{nd})	Random(3^{rd})

[Abdolazimi & Zafarani 2021]

Which noise-enhanced link prediction measures works best for each type of graph?

Models	**Parameters**	**Noise-enhanced link prediction methods Ranking**			
Random(n, p)	all p	Katz(1^{st})	**CN**(2^{nd})	AA(3^{rd})	-
Small-world	$p <= 0.1$	AA(1^{st})	**CN**(2^{nd})	PA(3^{rd})	Katz$(4^{t}h)$
	$p = 1$	Katz(1^{st})	**PA**(2^{nd})	CN(3^{rd})	AA$(4^{t}h)$
Configuration	Powerlaw	Katz(1^{st})	**AA**(2^{nd})	CN(3^{rd})	PA$(4^{t}h)$

[Abdolazimi & Zafarani 2021]

Part IV

- Network Science

- Privacy Preservation
  - Microdata Protection
  - Differential Privacy

- Natural Language Processing

Noise in Microdata Protection

- Masking approaches : main microdata protection techniques
  - Transform the original data to generate new valid data for statistical analysis
    - To preserve the confidentiality of respondents

- Random no
  - Perturbs ttribute with a random va

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Disease	DH	Chol	Temp
		Asian	64/09/27	F	94139	Divorced	Hypertension	3	260	35.2
		Asian	64/09/30	F	94139	Divorced	Obesity	1	170	37.7
		Asian	64/04/18	M	94139	Married	Chest pain	40	200	38.1
		Asian	64/04/15	M	94139	Married	Obesity	7	280	37.4
		Black	63/03/13	M	94138	Married	Hypertension	2	190	35.3
		Black	63/03/18	M	94138	Married	Short breath	3	185	38.2
		Black	64/09/13	F	94141	Married	Short breath	5	200	36.5
		Black	64/09/07	F	94141	Married	Obesity	60	290	39.8
		White	61/05/14	M	94138	Single	Chest pain	7	170	37.6
		White	61/05/08	M	94138	Single	Obesity	10	300	40.1
		White	61/09/15	F	94142	Widow	Short breath	5	200	36.9

An example of de-identified medical microdata table

[Brand 2002, Domingo-Ferrer et al. 2004, Ciriani 2007]

Random Noise

- Assumptions:
  - N : number of tuples
  - X_j is the j -th column of the original microdata table : a sensitive attribute

The uncorrelated additive random noise:

$$\boxed{x_{ij}} \Rightarrow \boxed{x_{ij} + \epsilon_{ij}}$$

$$\boxed{i = 1, \ldots, N} \quad \boxed{\epsilon_j \sim N(0, \sigma^2_{\epsilon_j})} \quad \boxed{\sigma^2_{\epsilon_j} = \alpha \sigma^2_{X_j}}$$

- Preserves the mean and the covariance

[Ciriani et al. 2007]

Differential Privacy

- Goal: Minimize the probability of **identifying a single record**

- Common method : **add random noise** to the original data

- Th
  1
  2

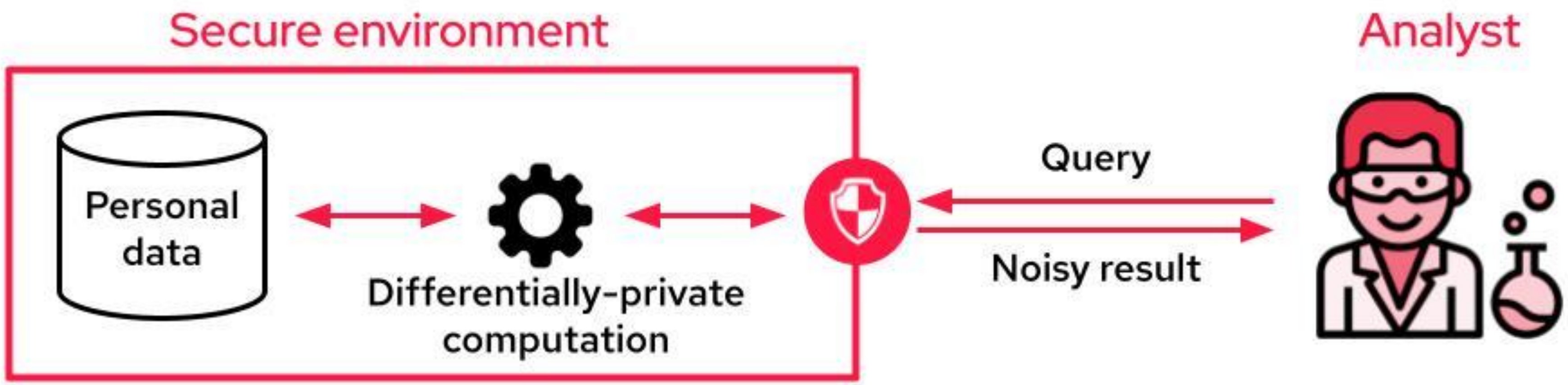


Image credit: Brett Jordan

[Dwork 2006, Dwork 2008]

Differential Privacy

- Preserve ϵ-differential privacy : add Laplacian noise
- Preserve (ϵ, δ)-differential privacy : add Gaussian noise

- Assume for $f : D \to R^k$, the sensitivity of f is

$$\boxed{\Delta f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1}$$

for all databases D_1 , D_2 differing in at most one element.

[Dwork 2008]

ϵ-Differential Privacy Theorem

For $f : D \to R^k,$ the mechanism K_f that adds independently generated noise with distribution $\mathrm{Lap}(\frac{\Delta f}{\epsilon})$ to each of the k output terms preserves the ϵ-differential privacy.

[Dwork 2008]

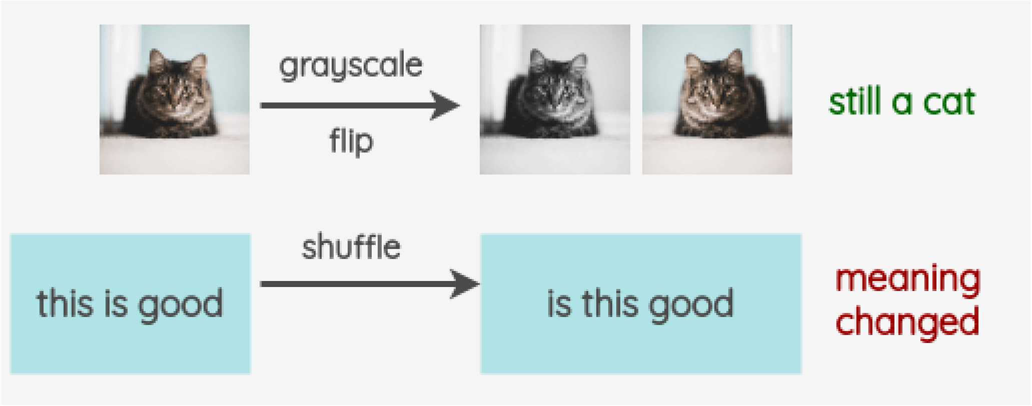
Part IV

- Network Science
- Privacy Preservation
- Natural Language Processing
  - Random noise injection

Noise-Enhanced Natural Language Processing

- Injection of noise into a neural network : data augmentation

- Data augmentation techniques for speech data or computer vision
  - Text data has not many popular techniques for data augmentation.
    - Natural language data are **difficult to process.**
    - It is too **hard to generate** realistic textual data.



[Goodfellow 2016, Coulombe et al.,2018, Chaudhary 2020]

Noise-Enhanced Natural Language Processing

- Data augmentation techniques used in artificial vision to NLP
  - Involving data transformations at the data pre-processing stage

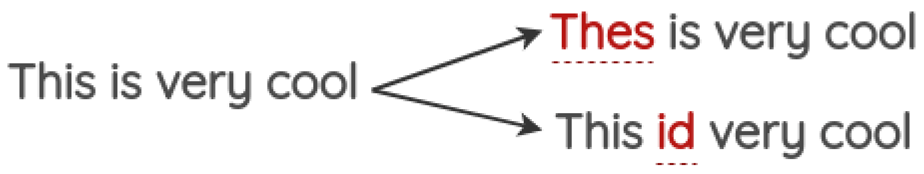
- Textual noise injection : text augmentation techniques
  - Making some changes in the texts : adding, deleting, modifying the letters in words, and changing the punctuation.

Image credit: Brett Jordan

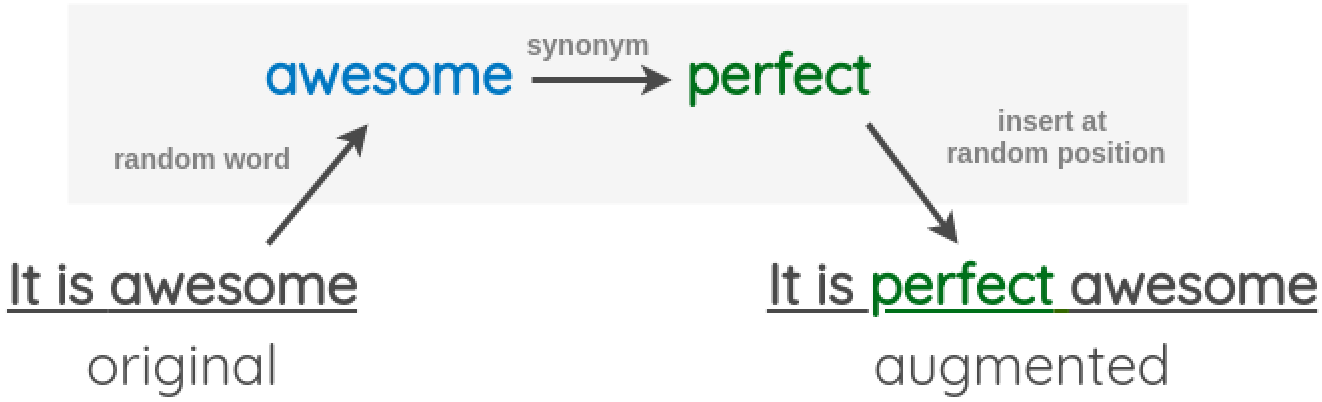
[Coulombe et al.,2018]

Random Noise Injection in NLP

- **Spelling Errors Injection**
  - Generates texts containing common misspellings
  - To train more robust models



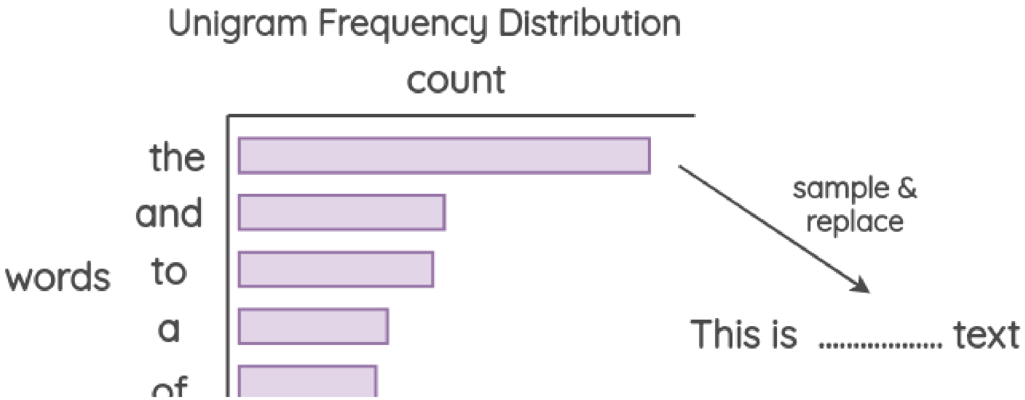
- **Random Insertion**
  - Find synonym of a random word and insert that into a random position in the sentence.



[Xie et al., 2020, Coulombe et al.,2018, Chaudhary 2020]

Random Noise Injection in NLP

- **Unigram Noising**
  - Perform replacement with words sampled from the unigram frequency distribution



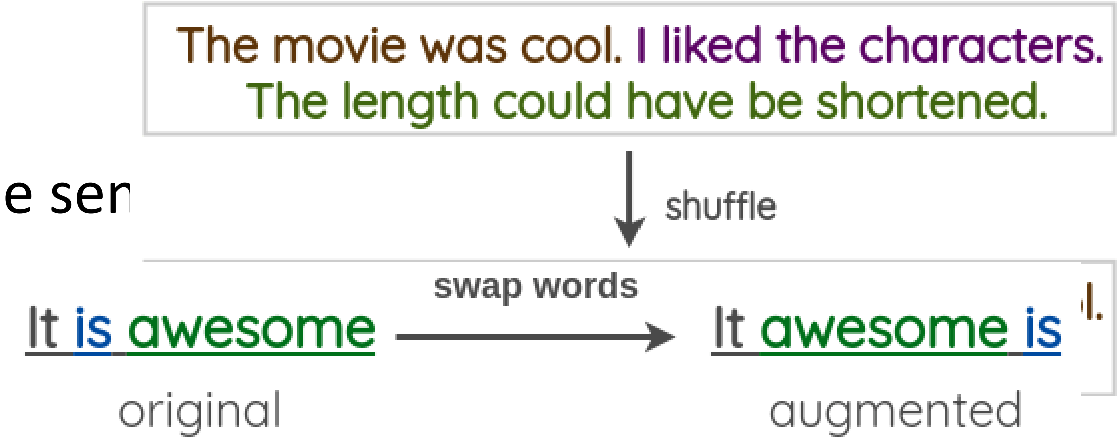
- **Blank Noising**
  - Replace a random word with a placehc

[Xie et al., 2020, Xie et al, 2017, Chaudhary 2020]

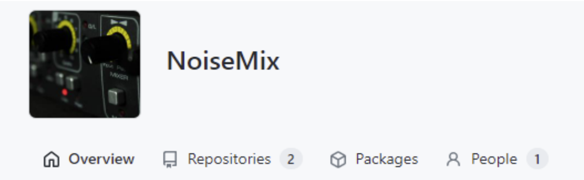
Random Noise Injection in NLP

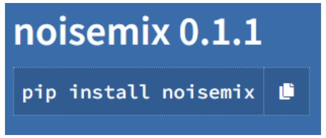
- **Sentence Shuffling**
  - Shuffle sentences present in a training text to create an augmented version

- **Random Swap**
  - Randomly swap any two words in the sen



- **Python library** of noise enhancem
  - It adds new rows to datasets

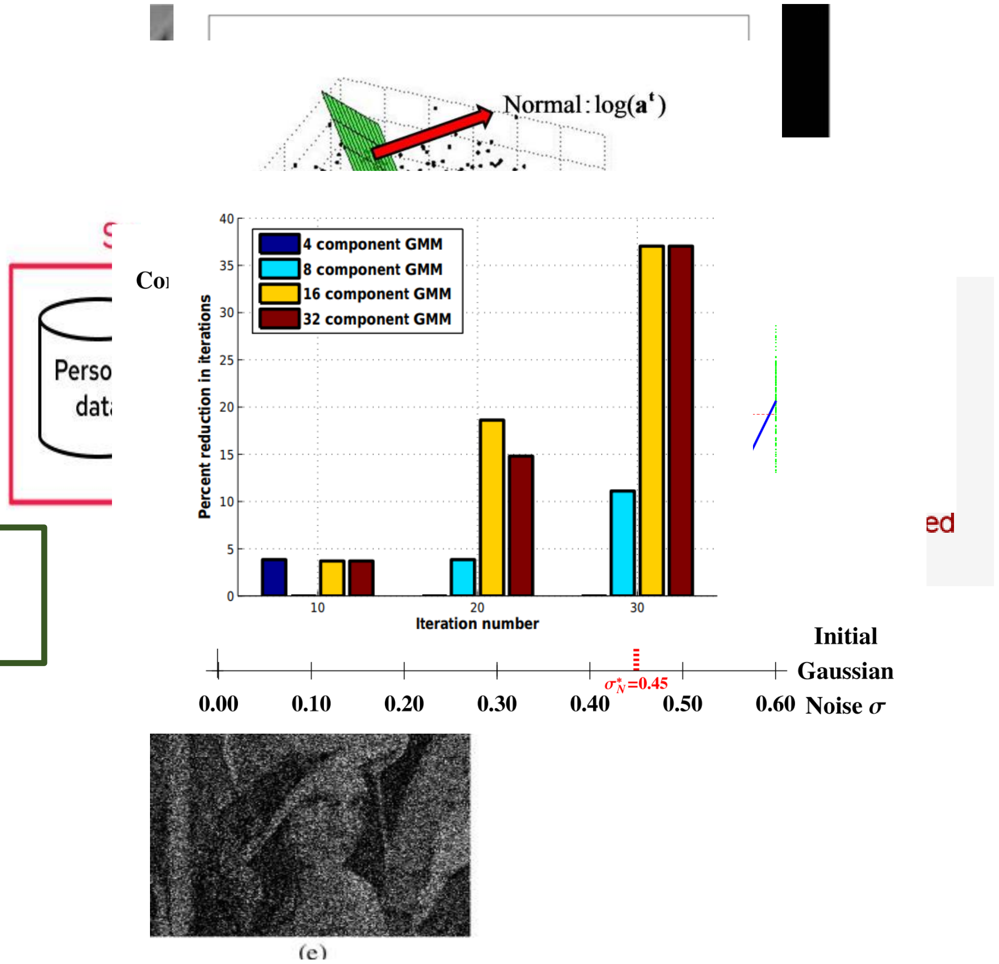




[Xie et al., 2020, Chaudhary 2020]

Conclusions

- Part I
  - Signal Processing
  - Image Processing
  - Biology and Physics
- Part II
  - Expectation Maximization
  - Clustering
  - Markov Chains and HMM
- Part III
  - Benefits of Noise in Neural Networks
  - Injecting Noise in Neural Networks
- Part IV
  - Network Science
  - Privacy Preservation
  - Natural Language Processing



Q&A

Reyhaneh Abdolazimi
Data Lab, Syracuse University

Reza Zafarani
Data Lab, Syracuse University