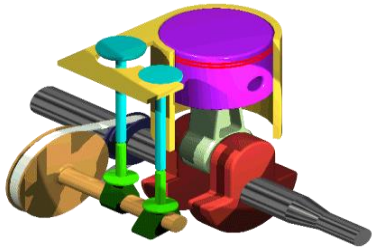
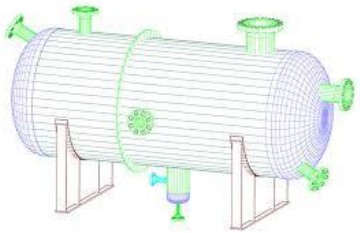
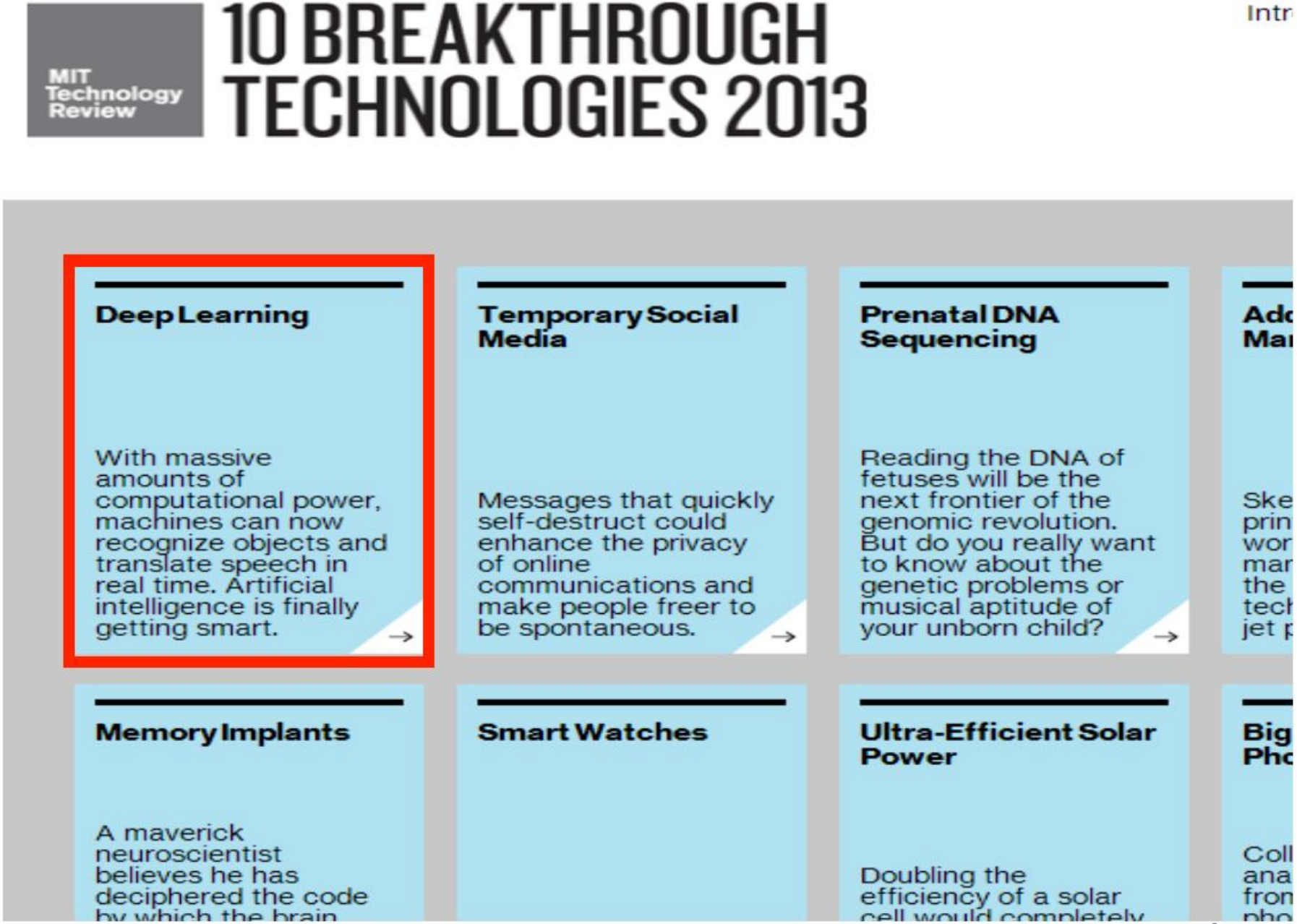
Deep Learning
and
Deep Fuzzy Networks

Nishchal K Verma, PhD,
Intelligent Data Engineering and Automation (IDEA) Laboratory,
IIT Kanpur, INDIA



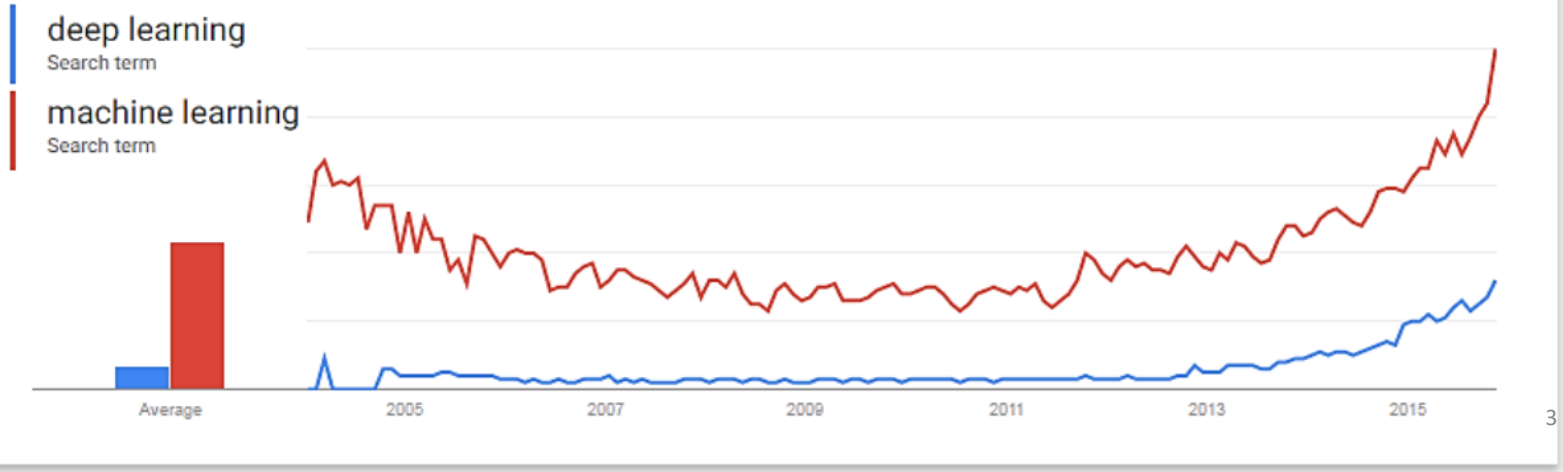



MIT Technology Review
10 BREAKTHROUGH TECHNOLOGIES 2013
Intr
Deep Learning
With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.
Temporary Social Media
Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.
Prenatal DNA Sequencing
Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?
Memory Implants
A maverick neuroscientist believes he has deciphered the code
Smart Watches
Ultra-Efficient Solar Power
Doubling the efficiency of a solar

Deep learning attracts lots of attention.

Interest

Google NGRAM & Google Trends

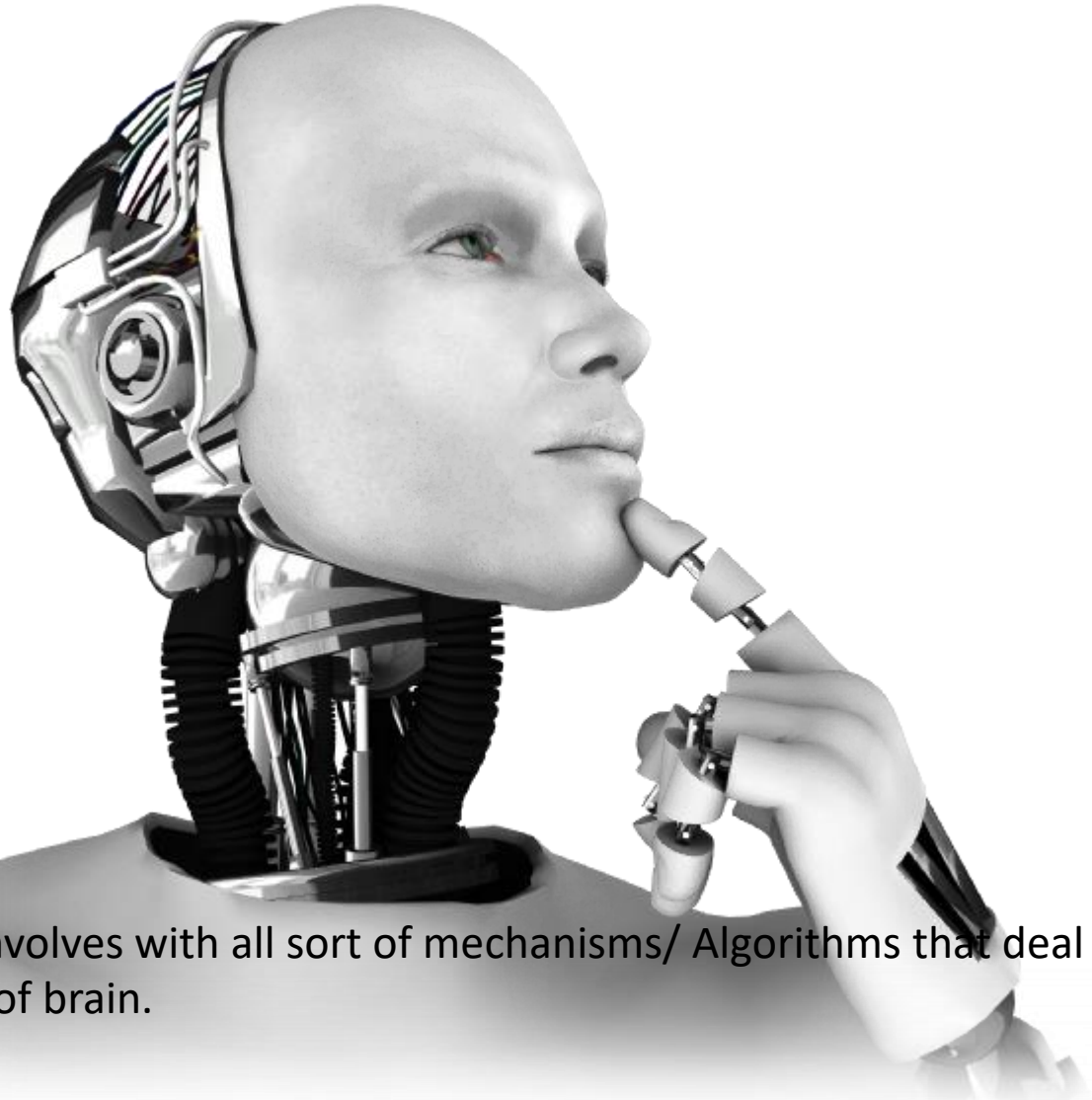


Faster Learning
Better Intelligence

Natural Intelligence in Living beings


© Rex Features

Artificial Intelligence
(or Machine Intelligence)

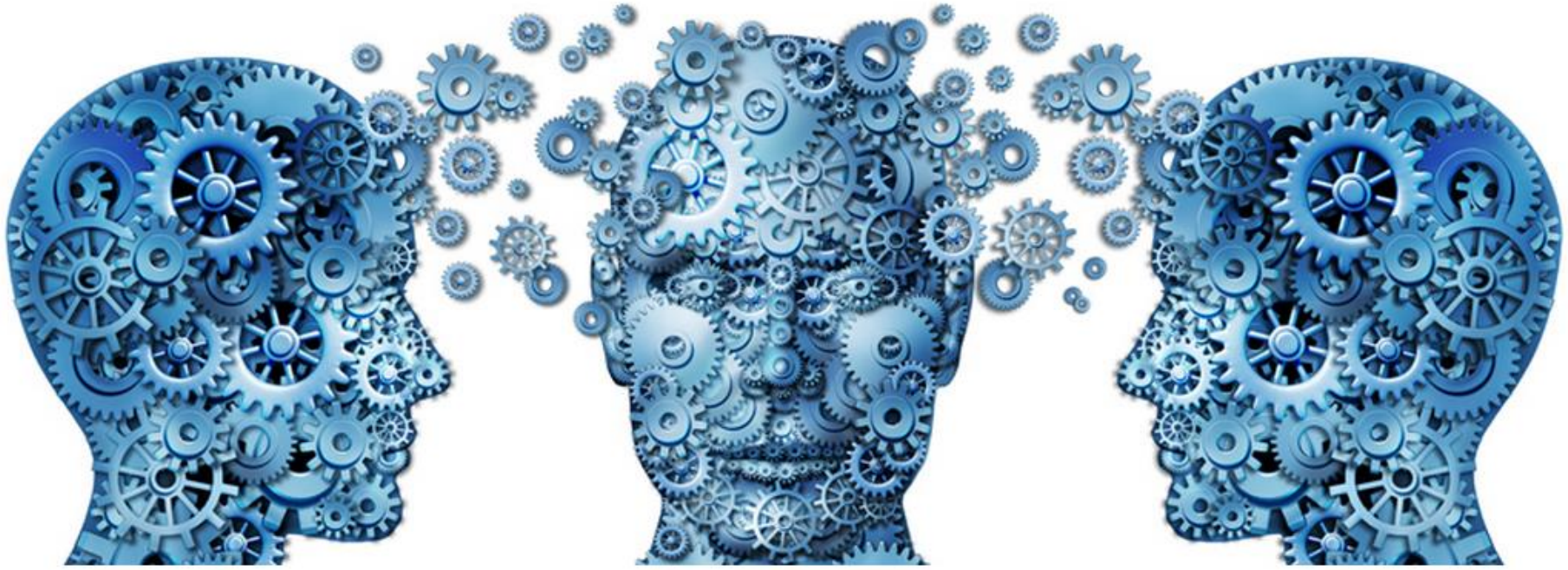


AI is a discipline which involves with all sort of mechanisms/ Algorithms that deal with mimicking the activities of brain.

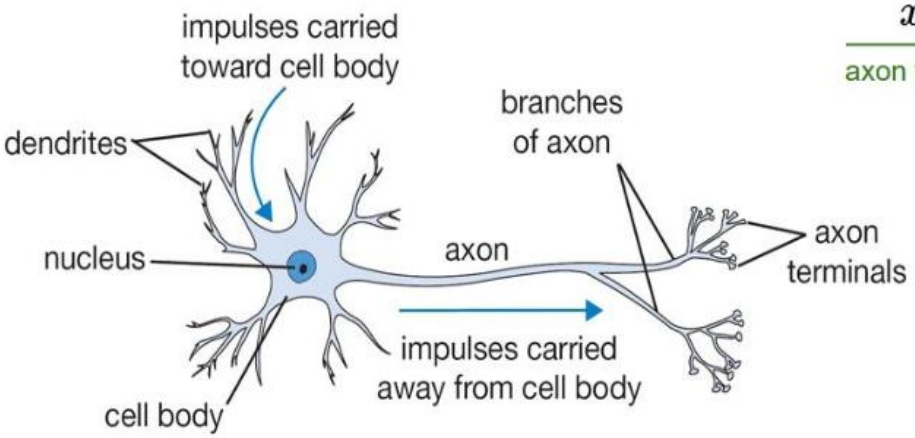
The major advantages of Artificial Intelligence:

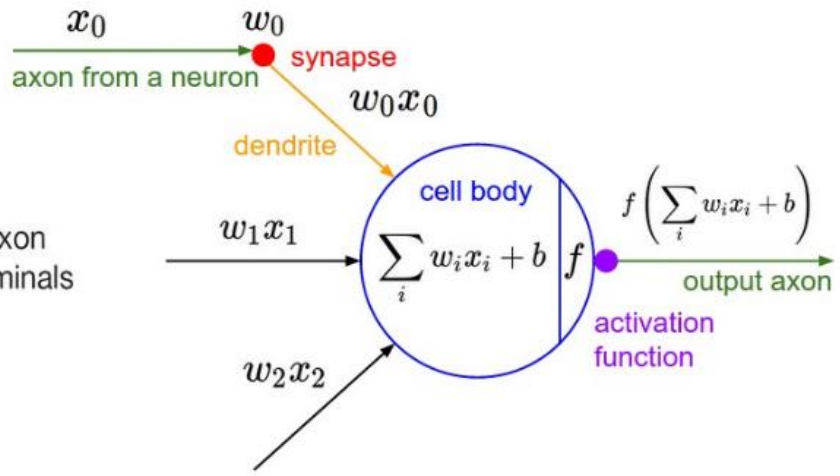
- ✓ Machines to not require sleep or breaks, and are able to function without stopping with same efficiency.
- ✓ Machines can continuously perform the same task without getting bored or tired.
- ✓ Such machines are needed to carry out dangerous tasks where the human health and safely are at risk.

Artificial Learning



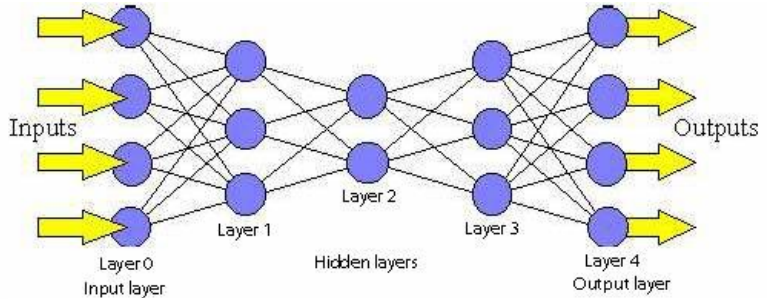
Understanding Simple Learning





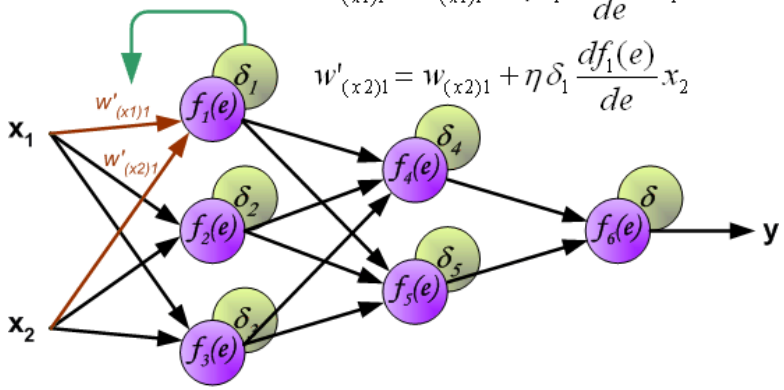
An artificial neuron contains a nonlinear activation function and has several incoming and outgoing weighted connections.

Feed forward Neural Network



$$w'_{(x1)1} = w_{(x1)1} + \eta\delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta\delta_1 \frac{df_1(e)}{de} x_2$$



Feed forward algorithm:

- Activate the neurons from the bottom to the top.

Backpropagation:

- Randomly initialize the parameters
- Calculate total error at the top, $f_6(e)$
- Then calculate contributions to error, δ_n, at each step going backwards.

Limitations of Neural Networks

Random initialization + **densely connected networks** lead to:

- Difficulty in training as the number of hidden layers increases
  - In backpropagation, gradient is progressively getting more dilute. That is, below top layers, the correction signal δ_n is minimal.
- To get stuck in local optima
  - The random initialization does not guarantee starting from the proximity of global optima.

Solution:

- Deep Learning/Learning multiple levels of representation

Deep Learning

- The first **deep-learning model** was proposed by **Alexey Grigorevich Ivakhnenko** and **Lapa** in **1965.** who used polynomial activation functions and analysed the network with statistical tools.
- In each layer, they selected the best features through statistical methods and forwarded them to the next layer.
- They did not use backpropagation to train their network end-to-end but used layer-by-layer least squares fitting where previous layers were independently fitted from later layers.

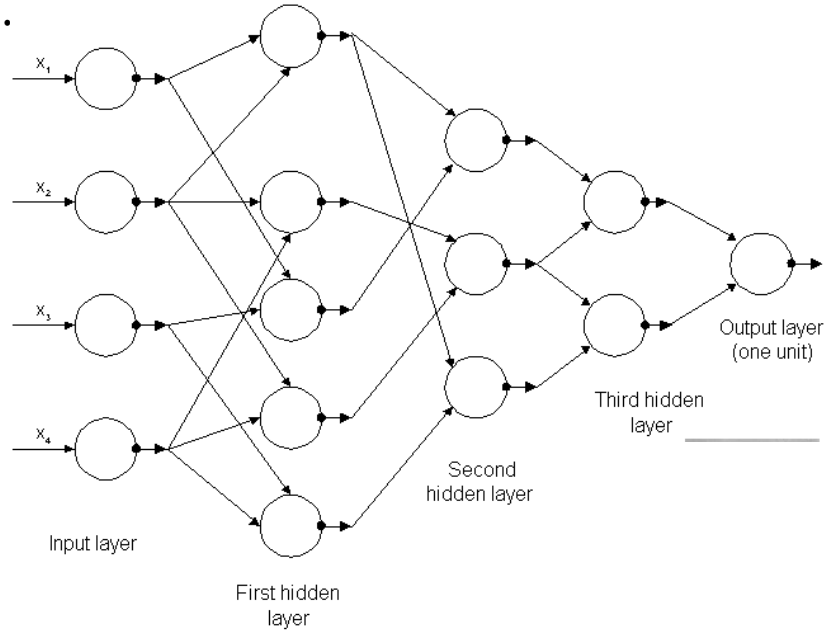


Figure 1: The architecture of the first known deep network

Source: (https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/)

- DEEP Learning (DL), as a subject deals with a set of algorithms developed to achieve deeper intuitions and intricate structures in data.

- DL framework comprises of multiple layers of nonlinear processing nodes which are trained over a large set of data [1].

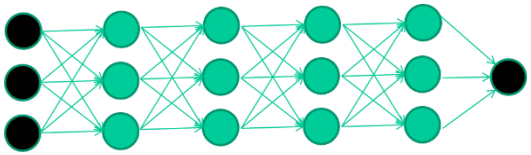
- Each layer in DL framework represents higher hierarchical level of abstraction than preceding layers.

- A DL algorithm finds complex, meaningful patterns and structures in large scale of unlabelled data.

[1] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. “Deep learning.” Nature 521, no. 7553 (2015): 436-444.

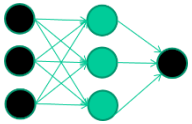
[2] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives.” IEEE transactions on pattern analysis and machine intelligence 35, no. 8 (2013): 1798-1828.

- Deep Learning uses several layers of nodes between input and output.

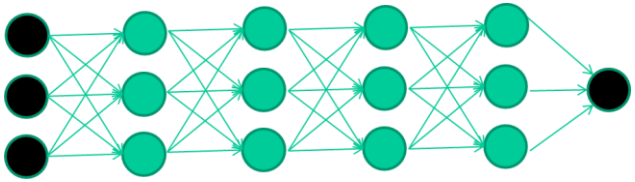


- The series of layers between input & output identify and process features in a series of stages, just as our brains.

- We have always had good algorithms for learning the weights in networks with 1 hidden layer



- But these algorithms are not good at learning the weights for networks with more hidden layers [6]



[6] Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, 1986

In 2006 Deep Breakthrough By

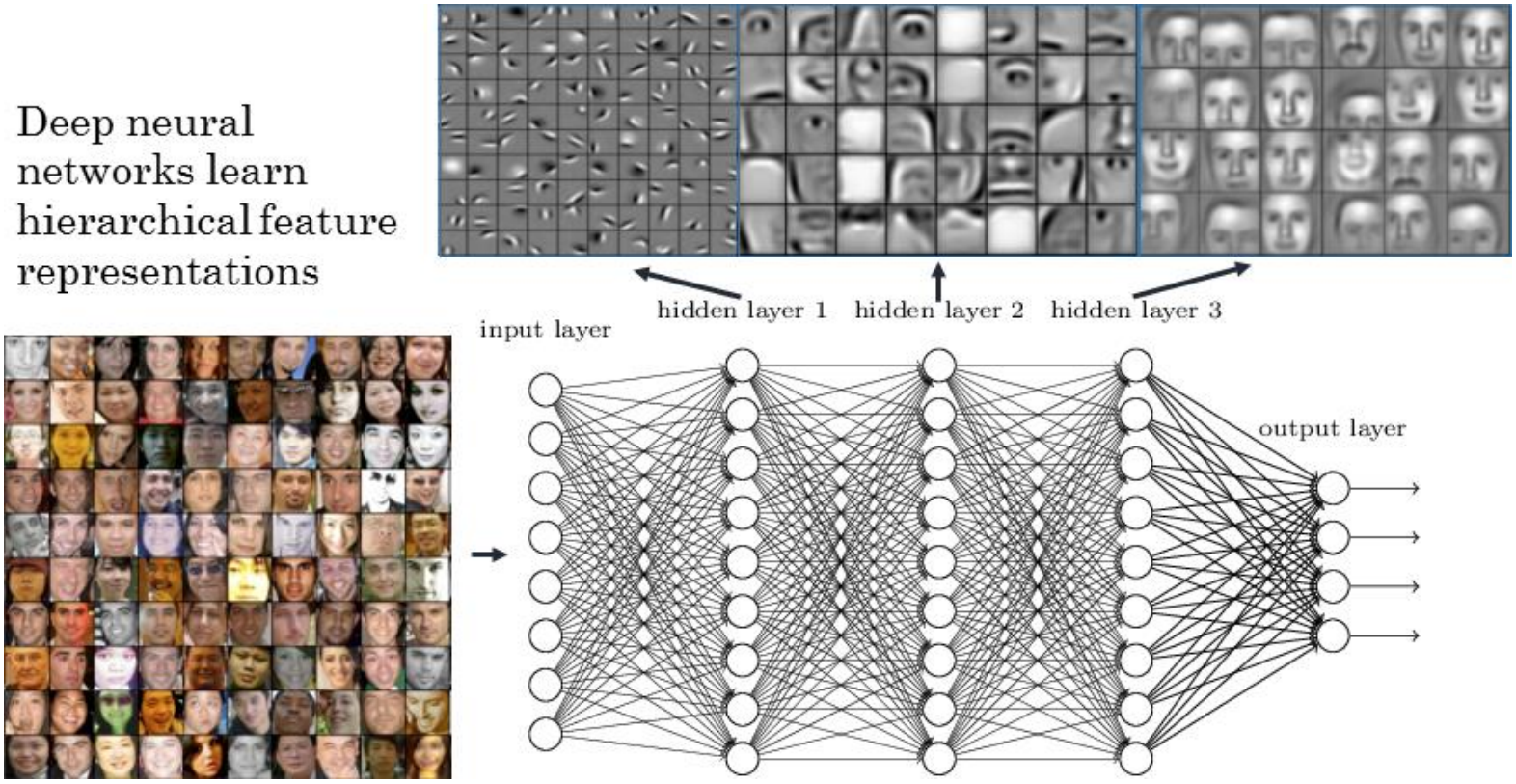
- Hinton, Osindero & Teh « A Fast Learning Algorithm for Deep Belief Nets », *Neural Computation*, 2006
- Bengio, Lamblin, Popovici, Larochelle « Greedy Layer-Wise Training of Deep Networks », *NIPS'2006*
- Ranzato, Poultney, Chopra, LeCun « Efficient Learning of Sparse Representations with an Energy-Based Model », *NIPS'2006*

Deep Learning and Computational Intelligence

Deep Neural Network

Deep Neural Network?

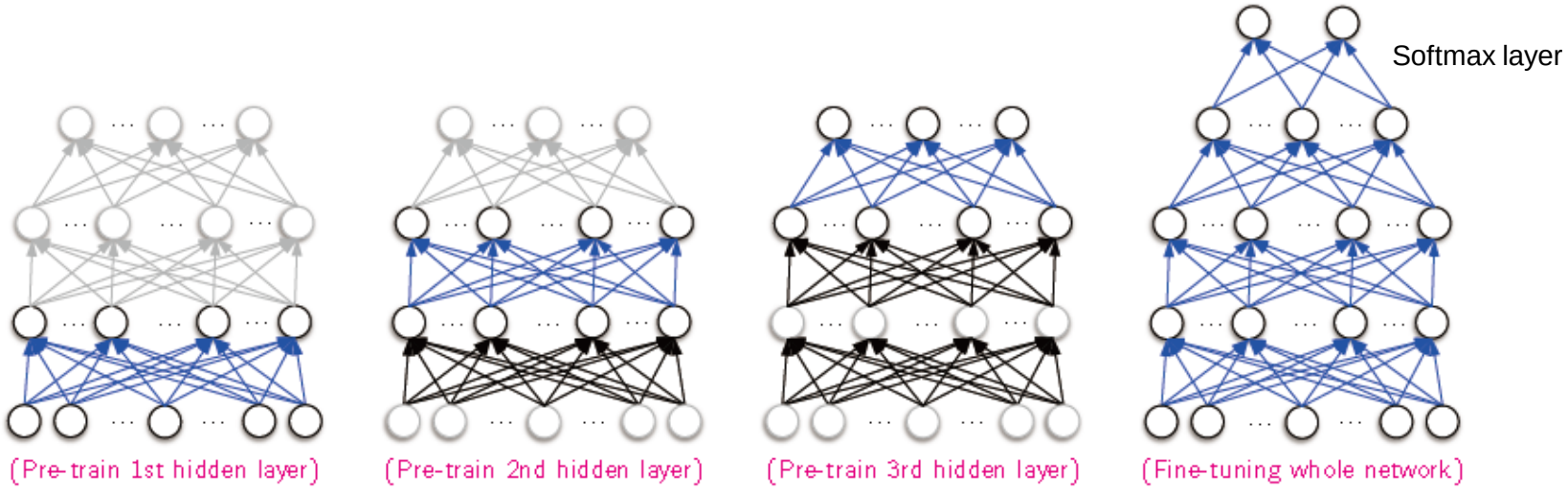
- The name **Deep neural networks (DNN)** formally has come into use in 2006.
- **DNN** is an **artificial neural networks (ANN)** with multiple hidden layers.
- The main idea of **DNN** to **extract high level features** from the **input data**.



Source: (http://www.rsipvision.com/exploring-deep-learning/)

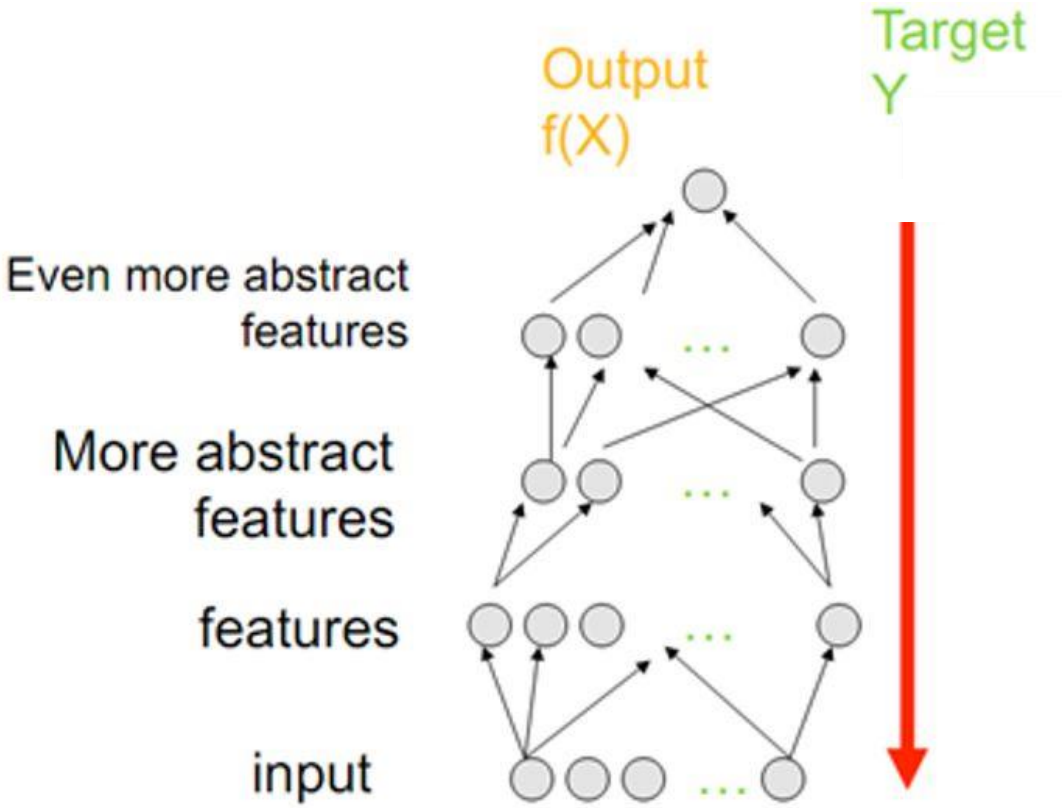
Proposed Training Approach for Deep Network by Hinton LeCun and Bengio

- Train first layer using data without the labels (unsupervised).
- Then freeze the first layer parameters and start training the second layer using the output of the first layer as the unsupervised input to the second layer.
- Repeat this for as many layers as desired. This builds our set of robust features.
- Use the outputs of the final layer as inputs to a supervised layer/model and train the last supervised.
-



Supervised Fine-tuning

Apply back propagation to adjust the whole weight of the network



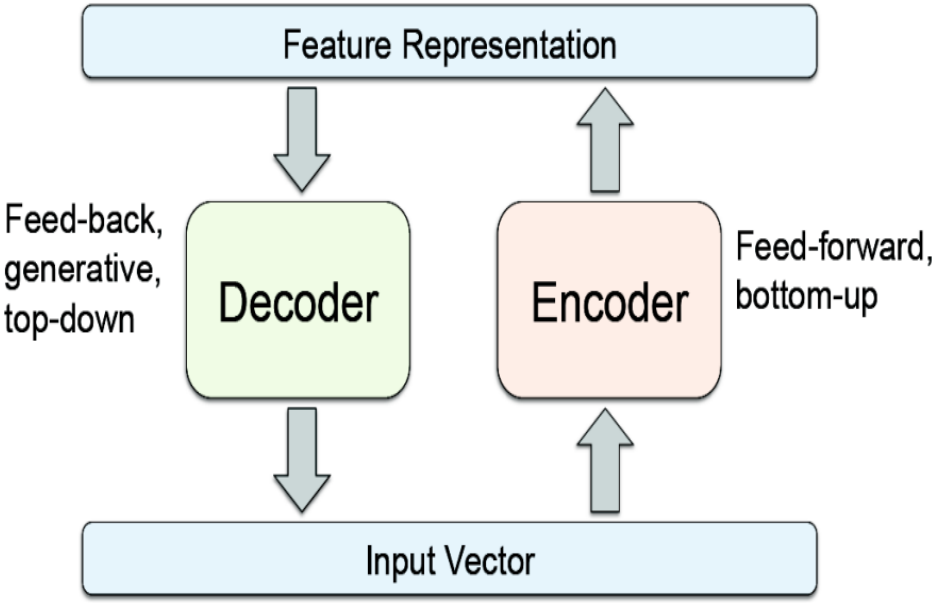
Most widely Used Deep Neural Networks Architectures are:

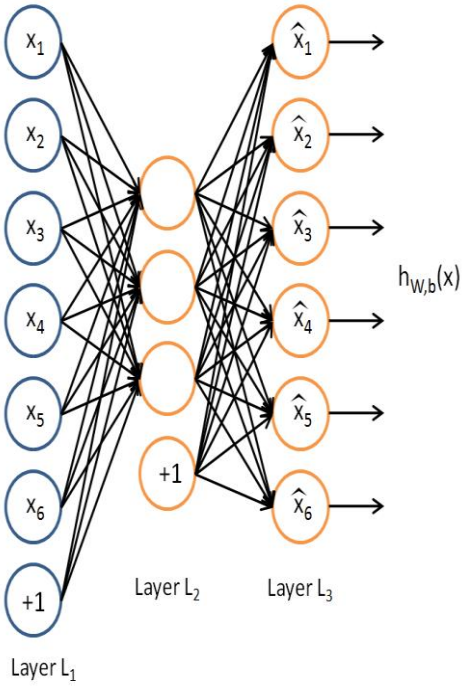
- **Deep Stacked Auto-Encoder (DSAE)**
- **Convolution Neural Network (CNN)**
- **Deep Recurrent Neural Network (DRNN) and Long and Short Term Memory (LSTM)**
- **Deep belief Network (DBN)**

Deep Stacked Auto-Encoder (DSAE)

Autoencoder:

- It is a neural network for feature extraction from unlabeled data.
- Learns an encoding of the inputs so as to recover the original input from the encodings as well as possible.



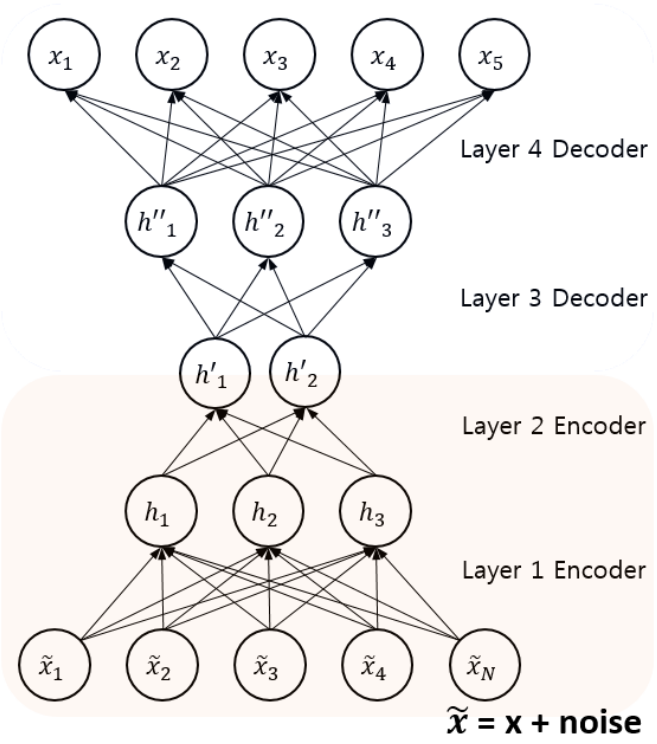


Source :(http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/)

Denoising Autoencoders

Introduce stochastic corruption in the input; e.g.:

1. Hide some features
2. Add Gaussian noise



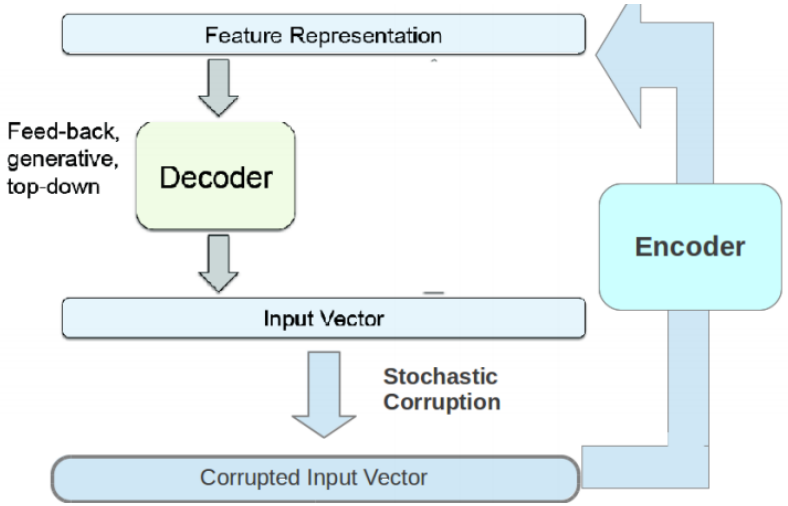
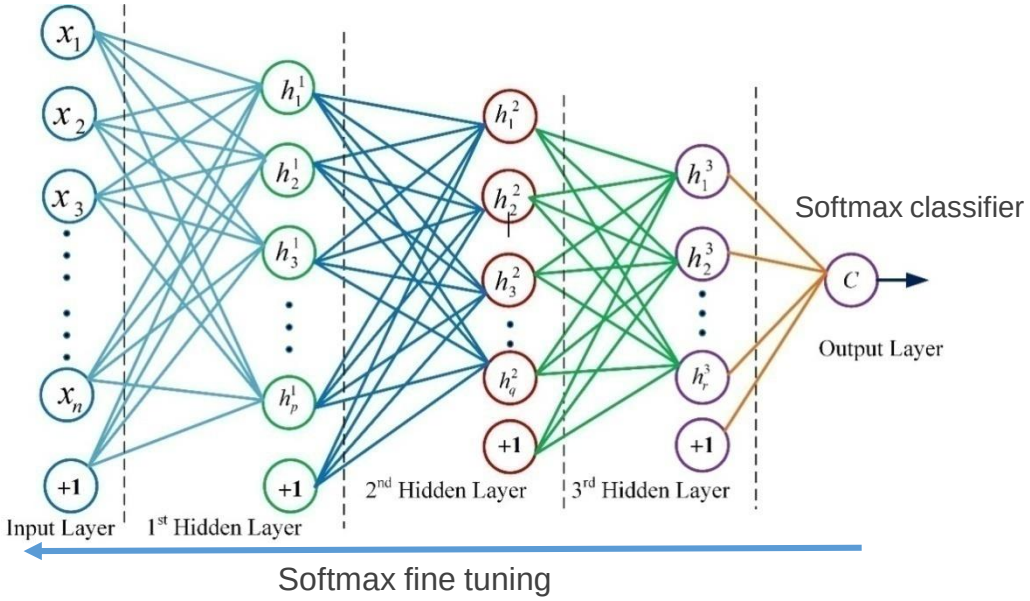


Figure : Denoising Autoencoders

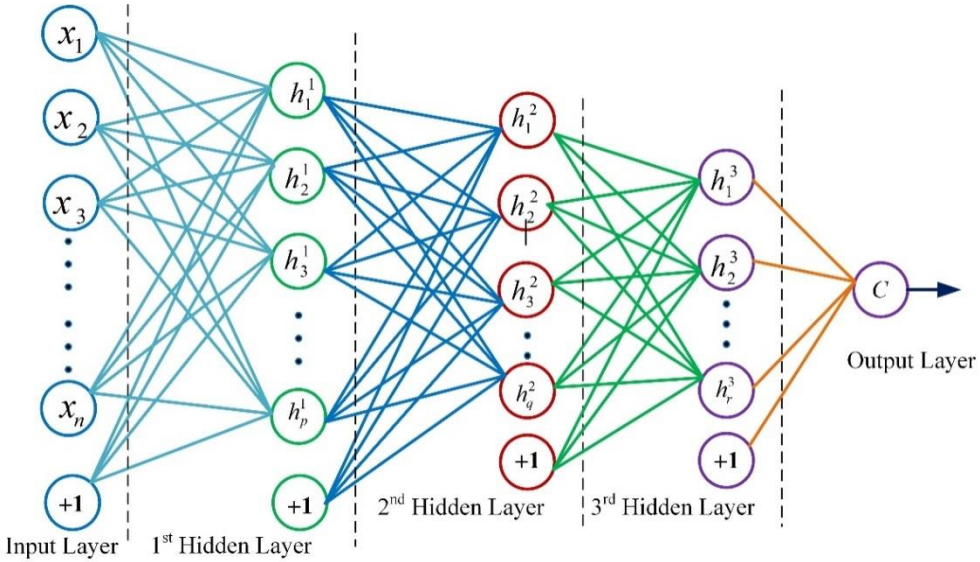
Fine tuning

- Fine tuning is a strategy that is used to increase the performance of stacked sparse auto-encoder.
- Fine tunning treats all layers of SSAE as single network so that in single iteration, we improve all the weights in SSAE.
- DNN was tuned with softmax based fine tuning and high level features are generated at third hidden layer.



Classification

- The high level features generated after fine tunning at last layer is fed as input to the classifiers.
- Softmax classifier, kernal support vector machine and Random Forest classifier etc. are used for the classification.



Softmax classifier, kernal support vector machine and Random Forest classifier etc.

Deep Fuzzy Network (DFN)

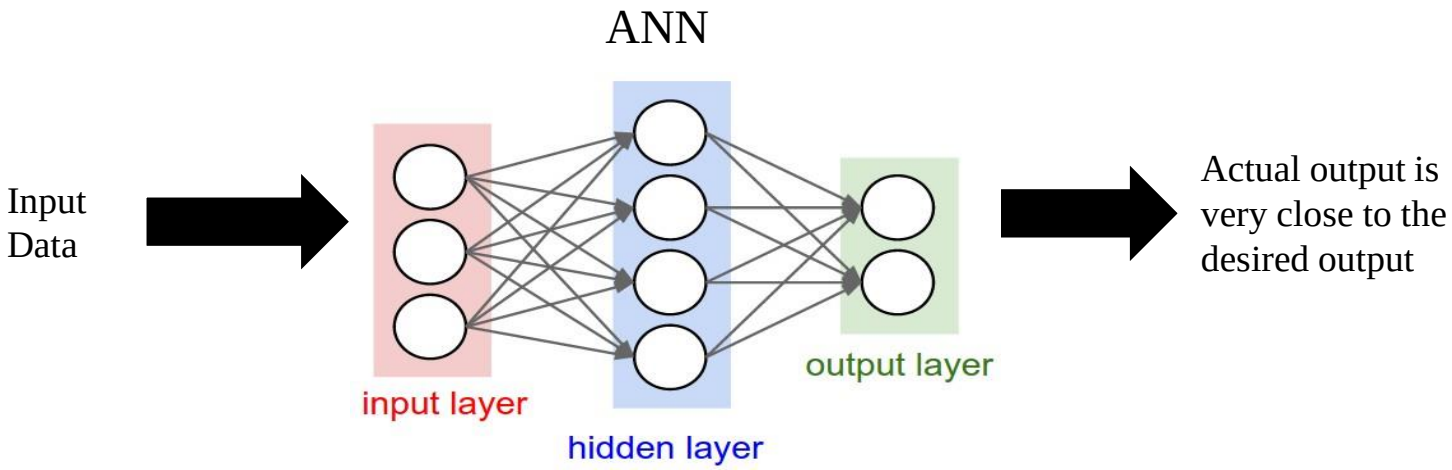


Fig : ANN can approximate a nonlinear system model with high accuracy
(Source: http://cs231n.github.io/neural-networks-1/)

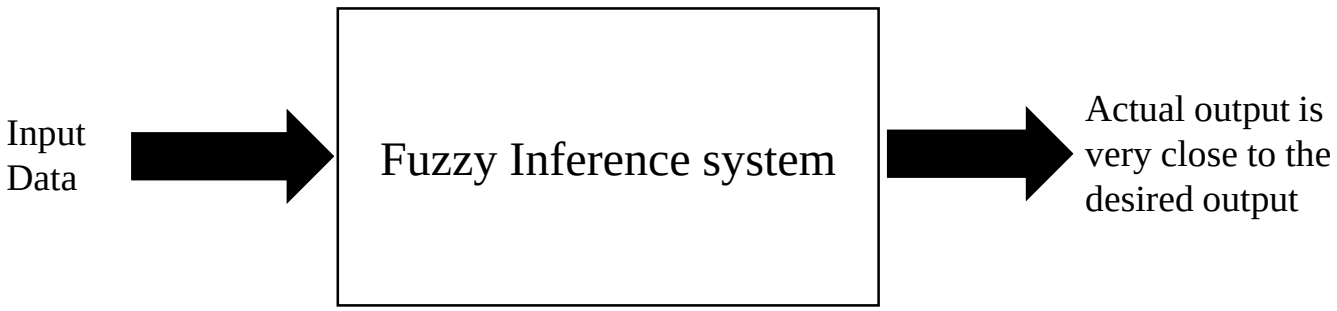


Fig : FIS can also approximate a nonlinear system model with high accuracy

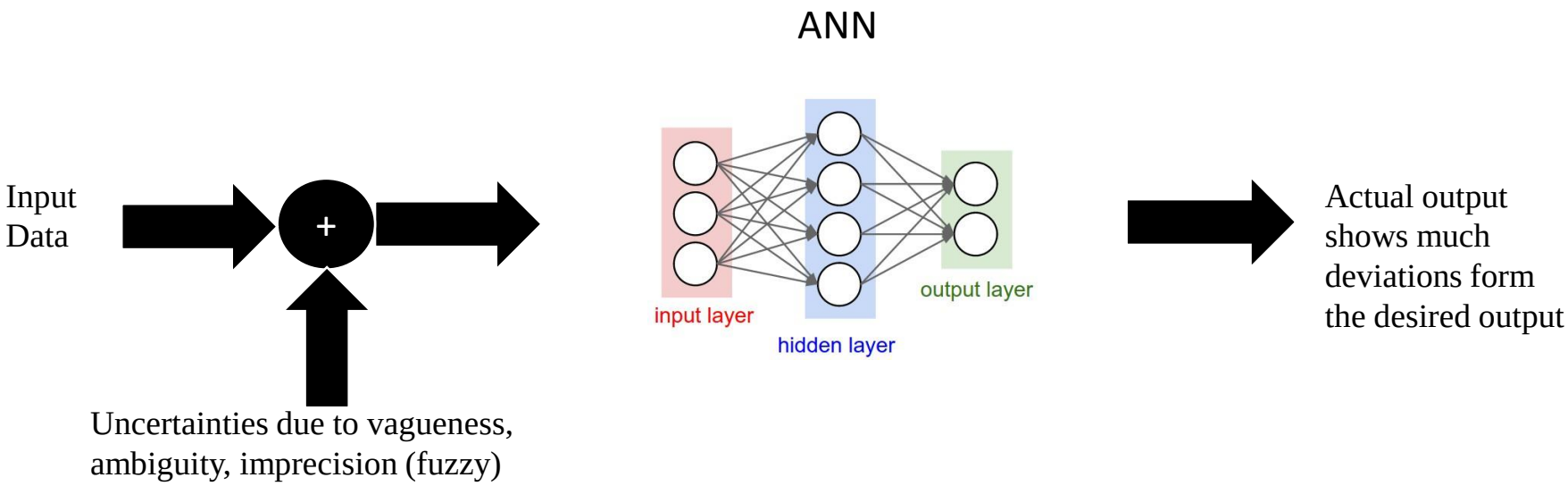


Fig : ANN can't process various kinds of uncertainties and actual output deviates from desired output.
(Source: http://cs231n.github.io/neural-networks-1/)

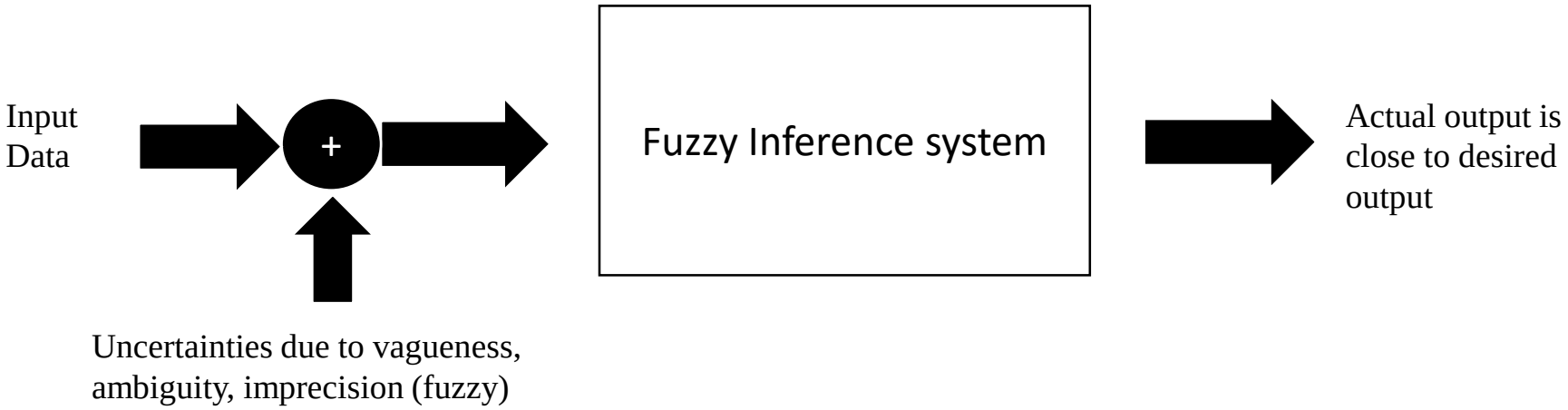


Fig : FIS can process various kinds of uncertainties and actual output is close to the desired output.

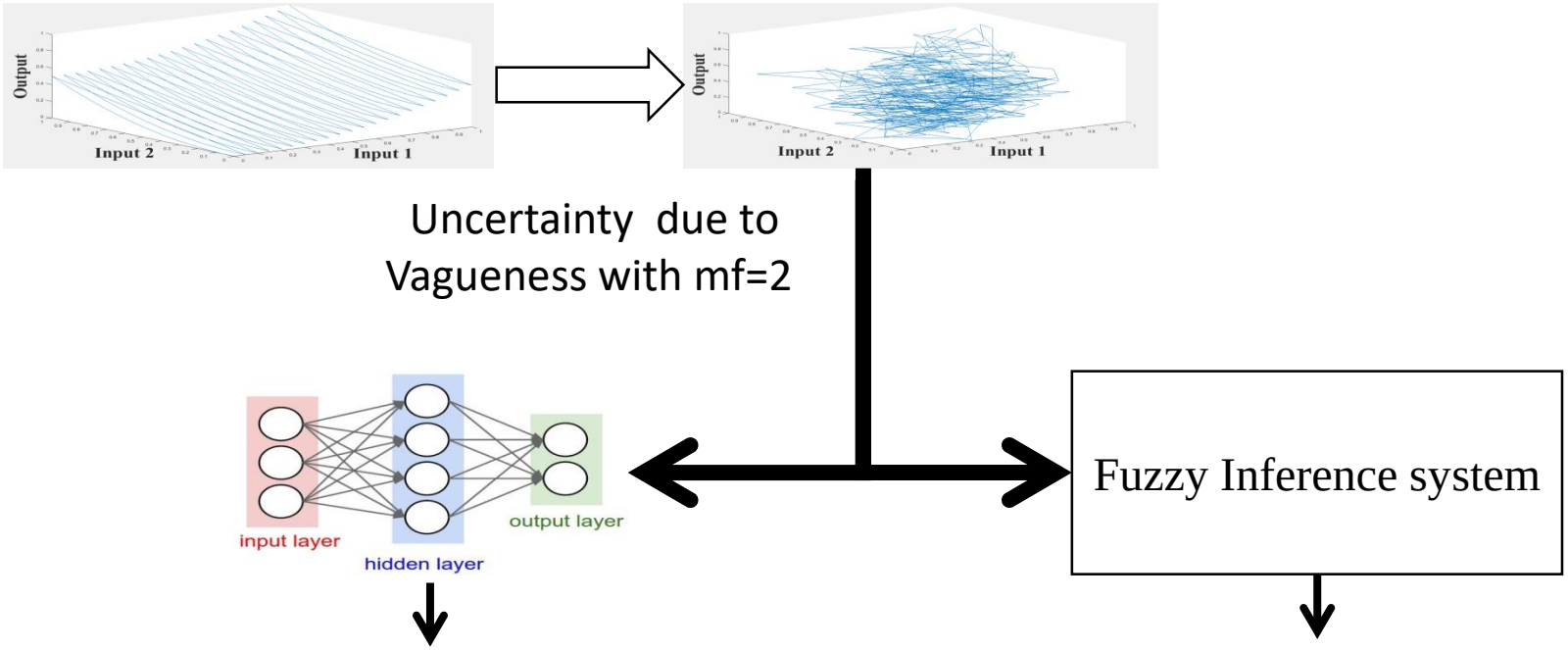


Table : Comparison of working of ANN and FIS in presence of fuzziness.
FIS outperforms ANN.

	Training Error		Testing Error	
	FIS	ANN	FIS	ANN
No. of MFs =3	3.3521	16.2122	3.3195	18.9635
No. of MFs =4	4.0311		4.4056	
No. of MFs =5	5.7251		5.5243	

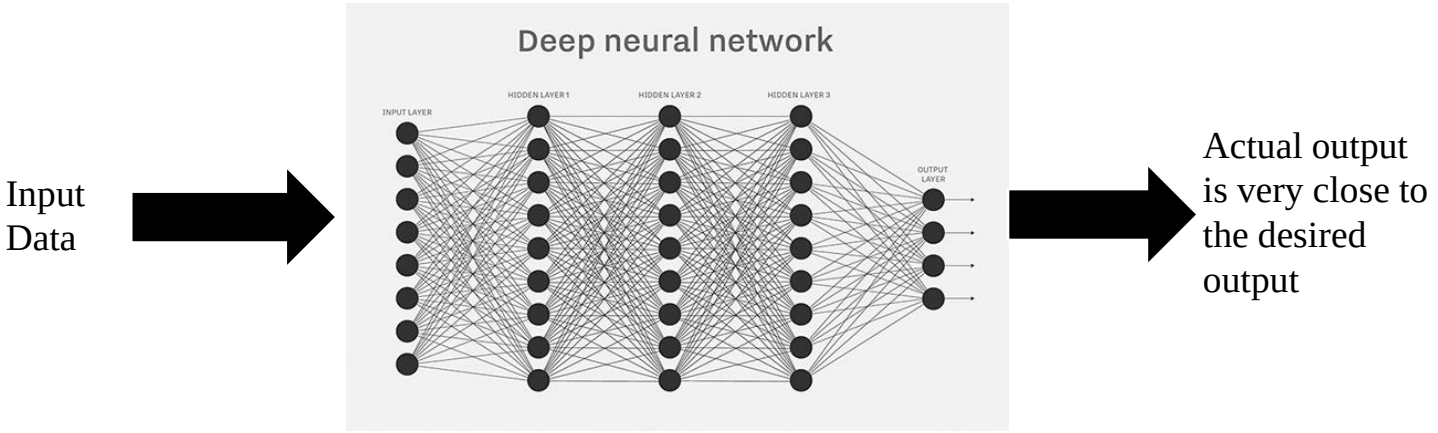


Fig : DNN can model highly complex nonlinear models of inputs very closely to desired outputs
(Source: http://searchnetworking.techtarget.com/definition/neural-network)

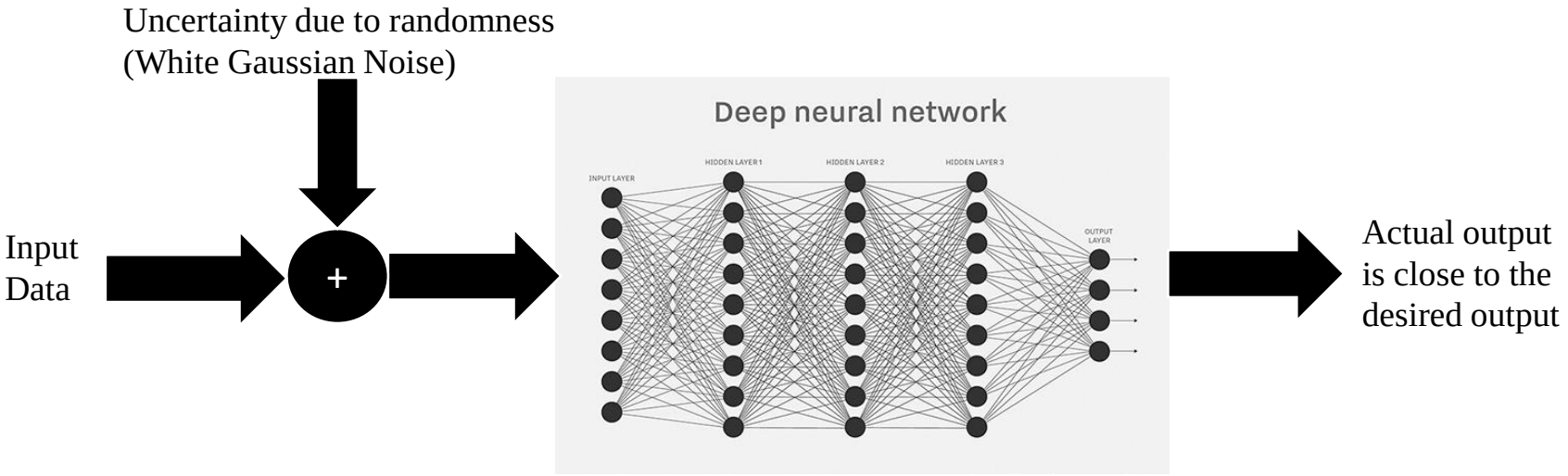
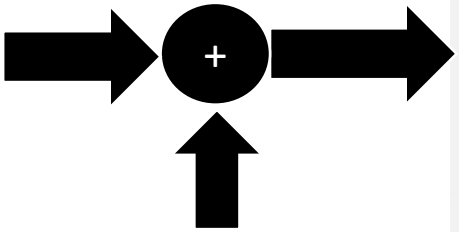


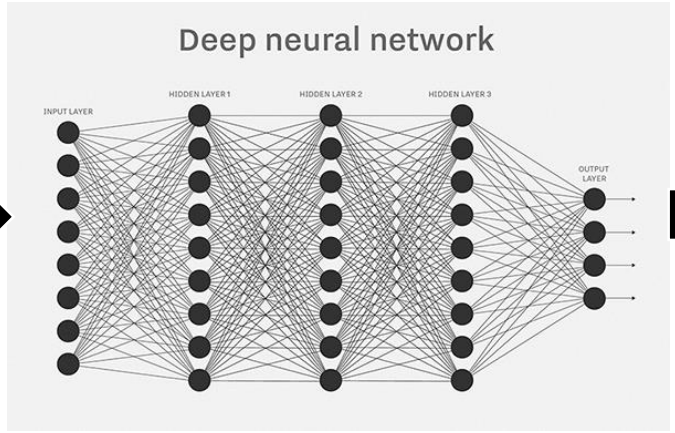
Fig : DNN can process uncertainty due to randomness (noise) in inputs very closely to desired outputs upto some extents
(Source: http://searchnetworking.techtarget.com/definition/neural-network)

Motivation 1: DNN is unable to model uncertainty due to vagueness, ambiguity and impreciseness.

Input Data



Uncertainties due to vagueness, ambiguity, imprecision (fuzziness)

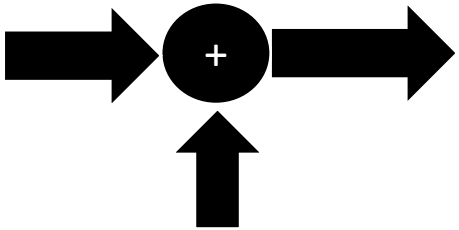


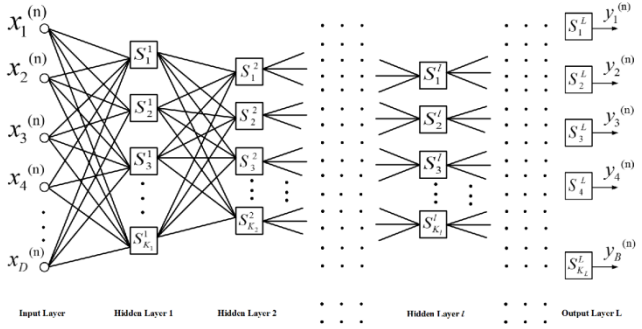
Actual output shows deviations form the desired output

Fig : DNN can't process uncertainty due to vagueness, ambiguity, imprecision (fuzziness) in inputs and actual output deviates from desired output.
(Source: http://searchnetworking.techtarget.com/definition/neural-network)

Motivation 1: Means DFN Treats uncertainty due to vagueness, ambiguity and impreciseness.

Input Data





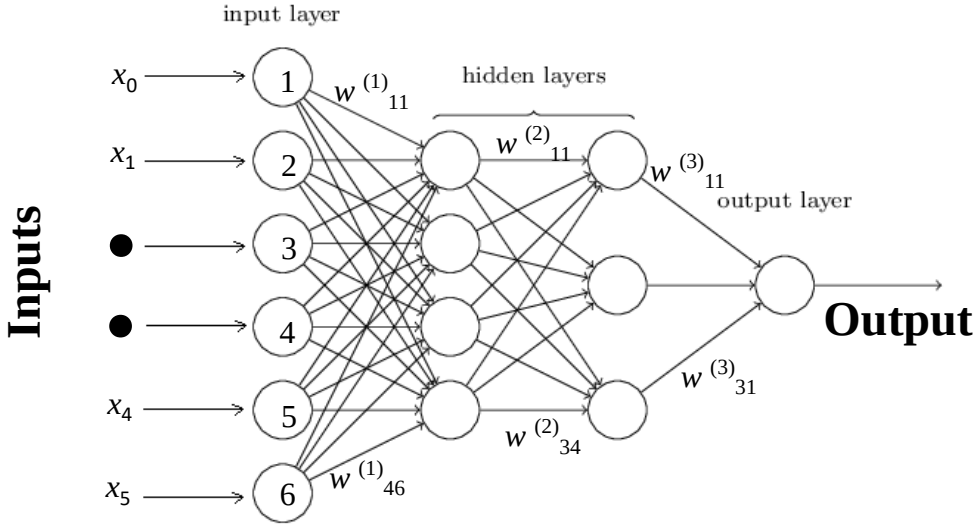
Actual output is close to desired output

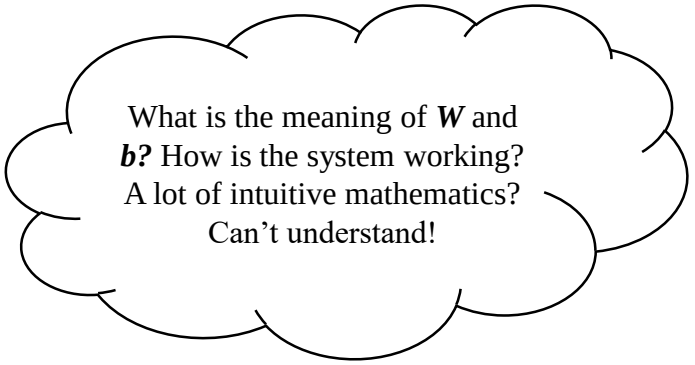
Uncertainties due to vagueness, ambiguity, imprecision (fuzzy)

Deep Fuzzy Network

DFN can process uncertainty due to vagueness, ambiguity, imprecision (fuzziness) in inputs and actual output is close to the desired output.

Motivation 2: It is very difficult to understand the working of system model underlying in DNN by only looking at its weight and bias parameters.

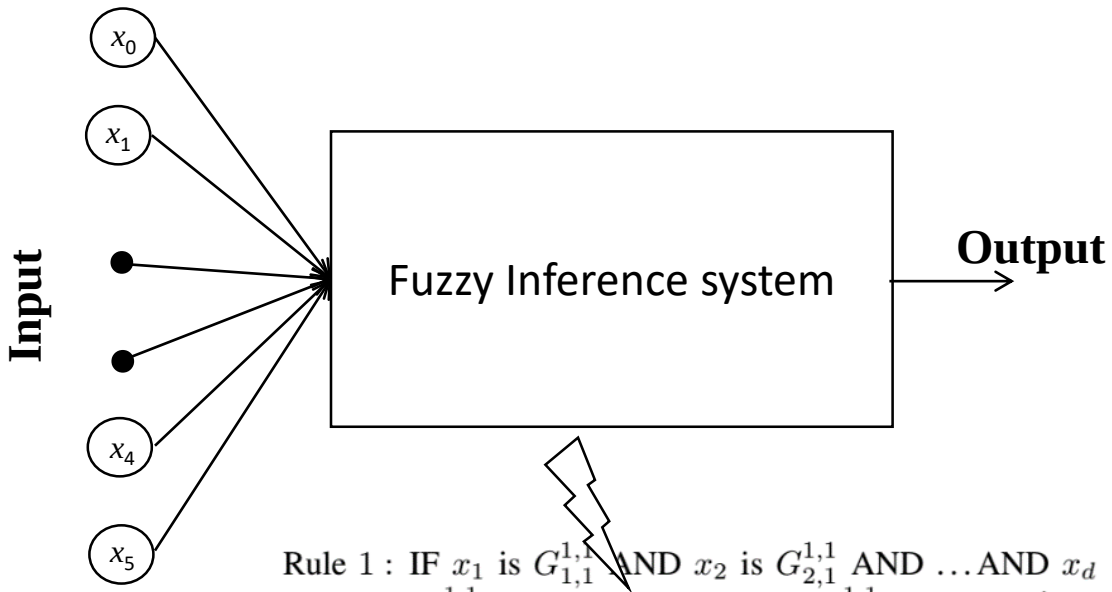




$$\boldsymbol{W}^{(1)} = \begin{bmatrix} w^{(1)}_{11} & \cdot & \cdot & w^{(1)}_{16} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ w^{(1)}_{41} & \cdot & \cdot & w^{(1)}_{46} \end{bmatrix} \qquad \boldsymbol{b}^{(1)} = \begin{bmatrix} b^{(1)}_1 \\ \cdot \\ \cdot \\ b^{(1)}_4 \end{bmatrix} \qquad \boldsymbol{W}^{(3)} = \begin{bmatrix} w^{(3)}_{11} & \cdot & \cdot & w^{(3)}_{13} \end{bmatrix}$$

$$\boldsymbol{W}^{(2)} = \begin{bmatrix} w^{(2)}_{11} & \cdot & \cdot & w^{(2)}_{14} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ w^{(2)}_{31} & \cdot & \cdot & w^{(2)}_{34} \end{bmatrix} \qquad \boldsymbol{b}^{(2)} = \begin{bmatrix} b^{(2)}_1 \\ \cdot \\ \cdot \\ b^{(2)}_3 \end{bmatrix} \qquad \boldsymbol{b}^{(3)} = \begin{bmatrix} b^{(3)}_1 \end{bmatrix}$$

Motivation 2: Explicitly explains the working of underlying system model in terms of linguistic rules intelligible to human beings.



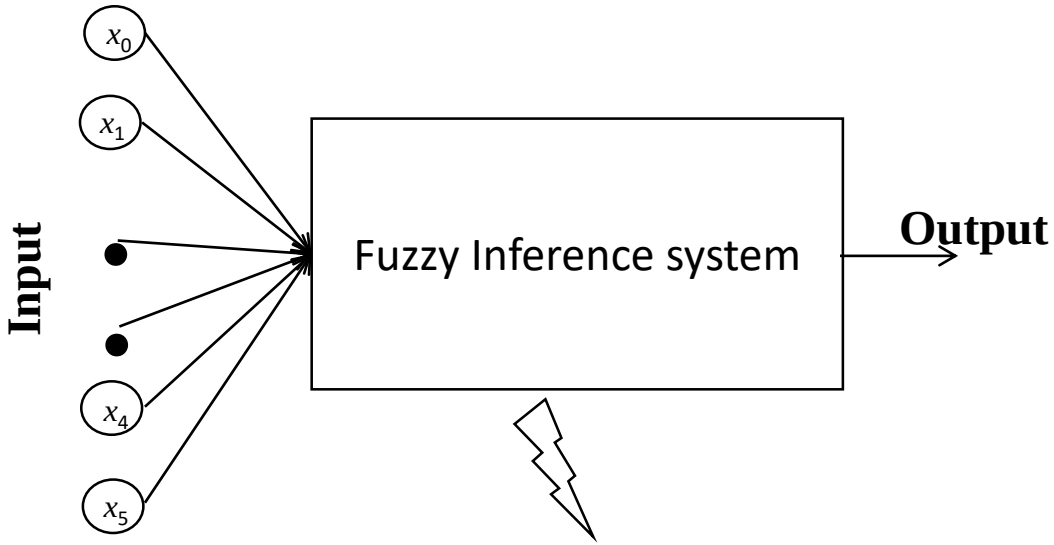
Rule 1 : IF x_1 is $G_{1,1}^{1,1}$ AND x_2 is $G_{2,1}^{1,1}$ AND ...AND x_d is $G_{d,1}^{1,1}$ AND ...AND x_D is $G_{D,1}^{1,1}$, THEN $z_1^1 = p_{1,0}^{1,1} + p_{1,1}^{1,1}x_1 + p_{1,2}^{1,1}x_2 + \cdots + p_{1,d}^{1,1}x_d + \cdots + p_{1,D}^{1,1}x_D$.

Rule 2 : IF x_1 is $G_{1,1}^{1,1}$ AND x_2 is $G_{2,1}^{1,1}$ AND ...AND x_d is $G_{d,1}^{1,1}$ AND ...AND x_D is $G_{D,2}^{1,1}$, THEN $z_1^1 = p_{2,0}^{1,1} + p_{2,1}^{1,1}x_1 + p_{2,2}^{1,1}x_2 + \cdots + p_{2,d}^{1,1}x_d + \cdots + p_{2,D}^{1,1}x_D$.

$\vdots$

Rule R_1 : IF x_1 is $G_{1,F_1}^{1,1}$ AND x_2 is $G_{2,F_1}^{1,1}$ AND ...x_d is $G_{d,F_1}^{1,1}$ AND ...AND x_D is $G_{D,F_1}^{1,1}$, THEN $z_1^1 = p_{R_1,0}^{1,1} + p_{R_1,1}^{1,1}x_1 + p_{R_1,2}^{1,1}x_2 + \cdots + p_{R_1,d}^{1,1}x_d + \cdots + p_{R_1,D}^{1,1}x_D$.

Motivation 3: Adding experts opinion.



Rule 1 : IF x_1 is $G_{1,1}^{1,1}$ AND x_2 is $G_{2,1}^{1,1}$ AND ...AND x_d is $G_{d,1}^{1,1}$ AND ...AND x_D is $G_{D,1}^{1,1}$, THEN $z_1^1 = p_{1,0}^{1,1} + p_{1,1}^{1,1}x_1 + p_{1,2}^{1,1}x_2 + \cdots + p_{1,d}^{1,1}x_d + \cdots + p_{1,D}^{1,1}x_D$.

Rule 2 : IF x_1 is $G_{1,1}^{1,1}$ AND x_2 is $G_{2,1}^{1,1}$ AND ...AND x_d is $G_{d,1}^{1,1}$ AND ...AND x_D is $G_{D,2}^{1,1}$, THEN $z_1^1 = p_{2,0}^{1,1} + p_{2,1}^{1,1}x_1 + p_{2,2}^{1,1}x_2 + \cdots + p_{2,d}^{1,1}x_d + \cdots + p_{2,D}^{1,1}x_D$.

$\vdots$

Rule R_1 : IF x_1 is $G_{1,F_1}^{1,1}$ AND x_2 is $G_{2,F_1}^{1,1}$ AND ...x_d is $G_{d,F_1}^{1,1}$ AND ...AND x_D is $G_{D,F_1}^{1,1}$, THEN $z_1^1 = p_{R_1,0}^{1,1} + p_{R_1,1}^{1,1}x_1 + p_{R_1,2}^{1,1}x_2 + \cdots + p_{R_1,d}^{1,1}x_d + \cdots + p_{R_1,D}^{1,1}x_D$.

A single TS fuzzy inference system acts as universal approximator.

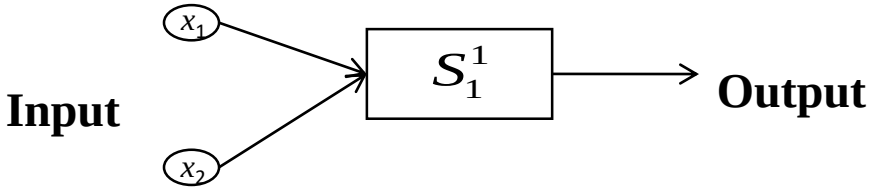
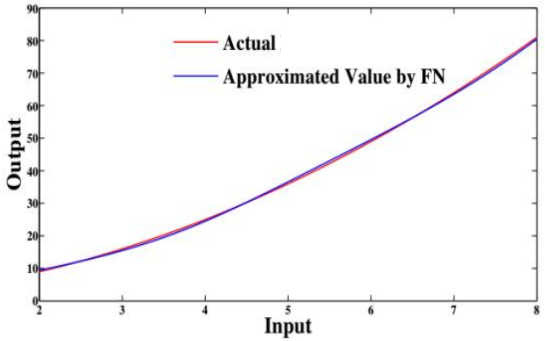
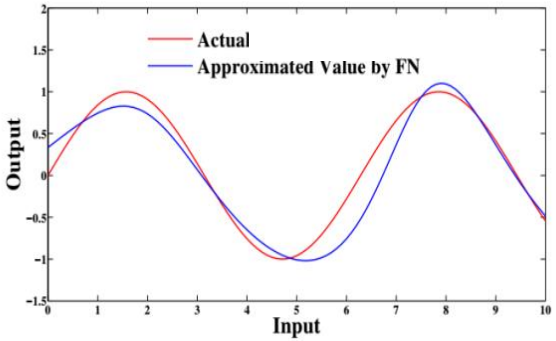


Fig : Single TS based FIS

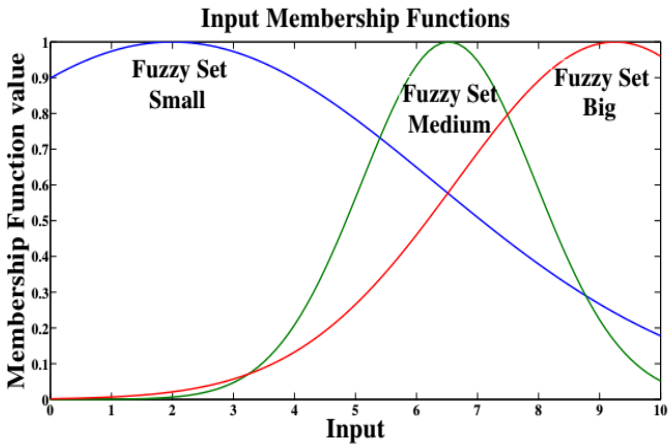


(a) Approximation of $f = x^2 + 2x + 1$

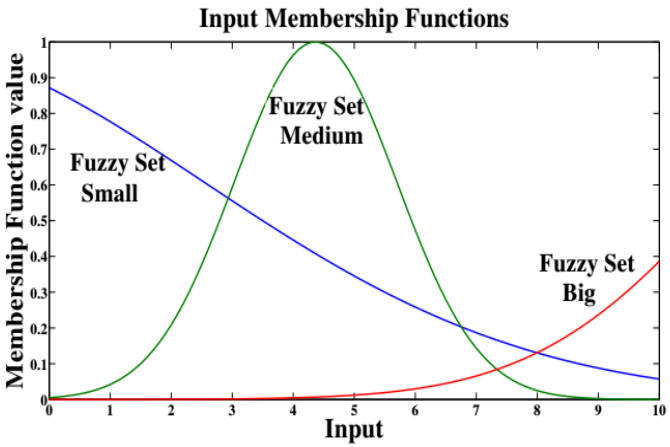


(b) Approximation of $f = sin(x)$

FIS has universal approximation capability.



(a) $f = x^2 + 2x + 1$



(b) $f = \sin(x)$

Fig : Identified fuzzy membership functions in data after universal approximation.

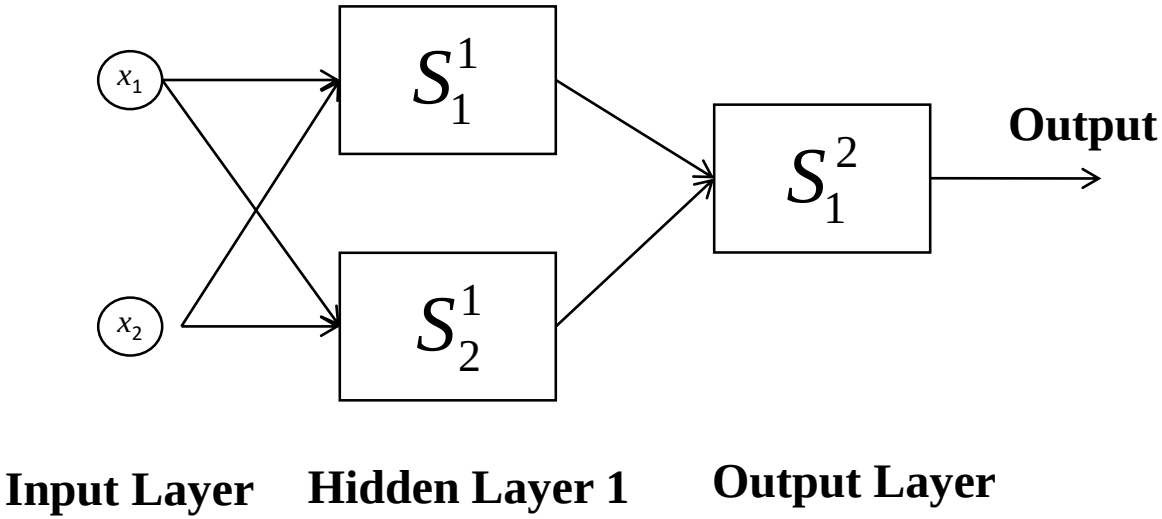


Fig : Feed-forward fuzzy network with one hidden layer

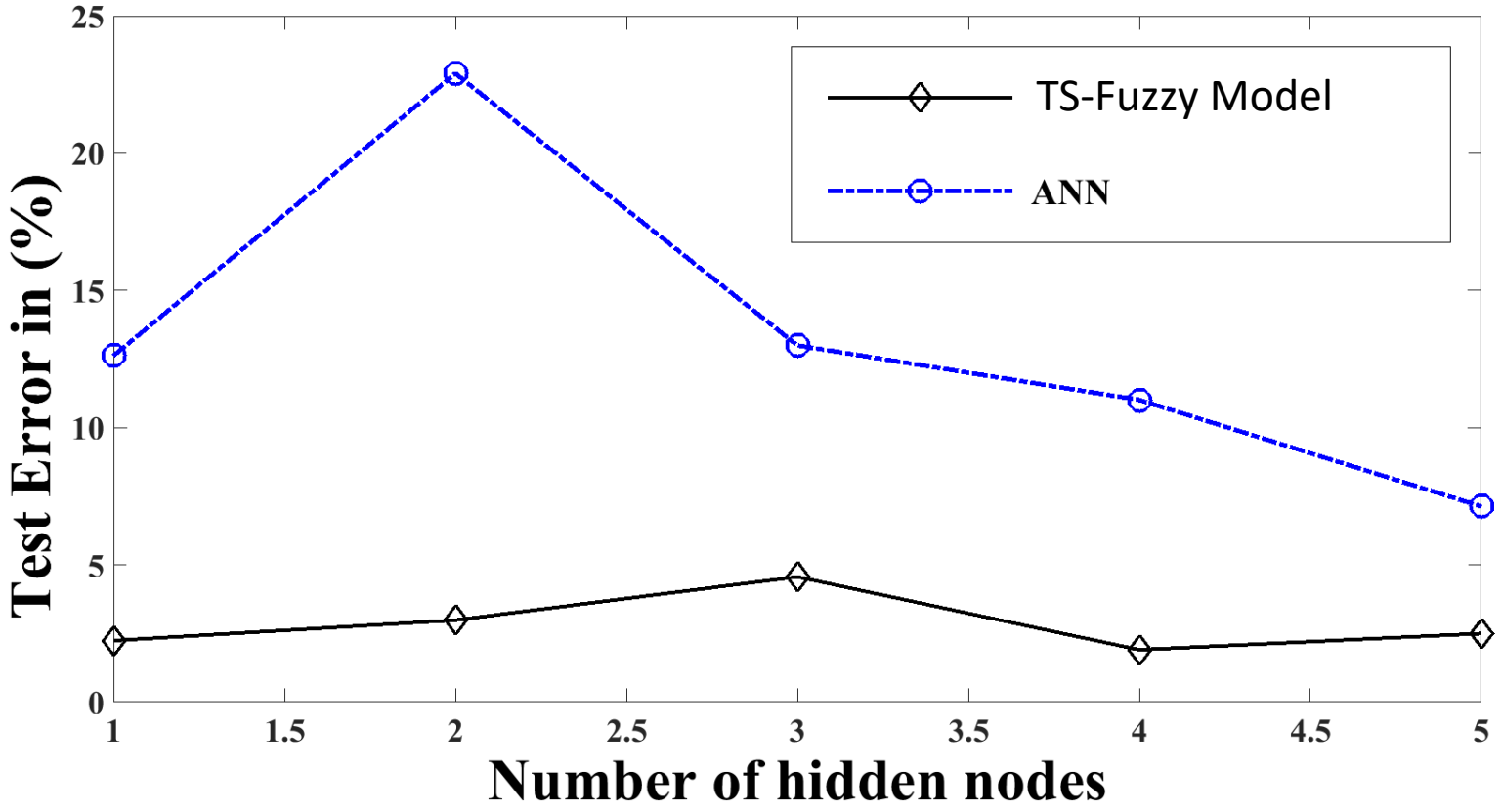


Fig : Comparison of working of ANN and feed-forward fuzzy network with single hidden layer for function approximation. Feed-forward fuzzy network with single hidden layer outperforms ANN.

Table : Comparison of working of ANN and feed-forward fuzzy network with single hidden layer for function approximatio
Feed-forward fuzzy network with single hidden layer outperforms ANN.

No. of hidden nodes	Training Error (%)		Testing Error (%)	
	FIS	ANN	FIS	ANN
1	2.2665	11.2934	2.2402	12.6192
2	3.17183	22.0352	2.9856	22.9035
3	3.8864	11.5336	4.5625	12.9957
4	2.1328	11.9352	1.9030	11.008
5	2.7925	8.1358	2.4982	7.129

Deep Fuzzy Network Architecture

- Architecture of a generalized DFN is as shown below.
- DFN considered in the proposed work has fully connected layers i.e. each layer is fully connected to next layer in DFN. Each layer in a DFN is composed of multiple fuzzy nodes which are TS fuzzy inference systems.

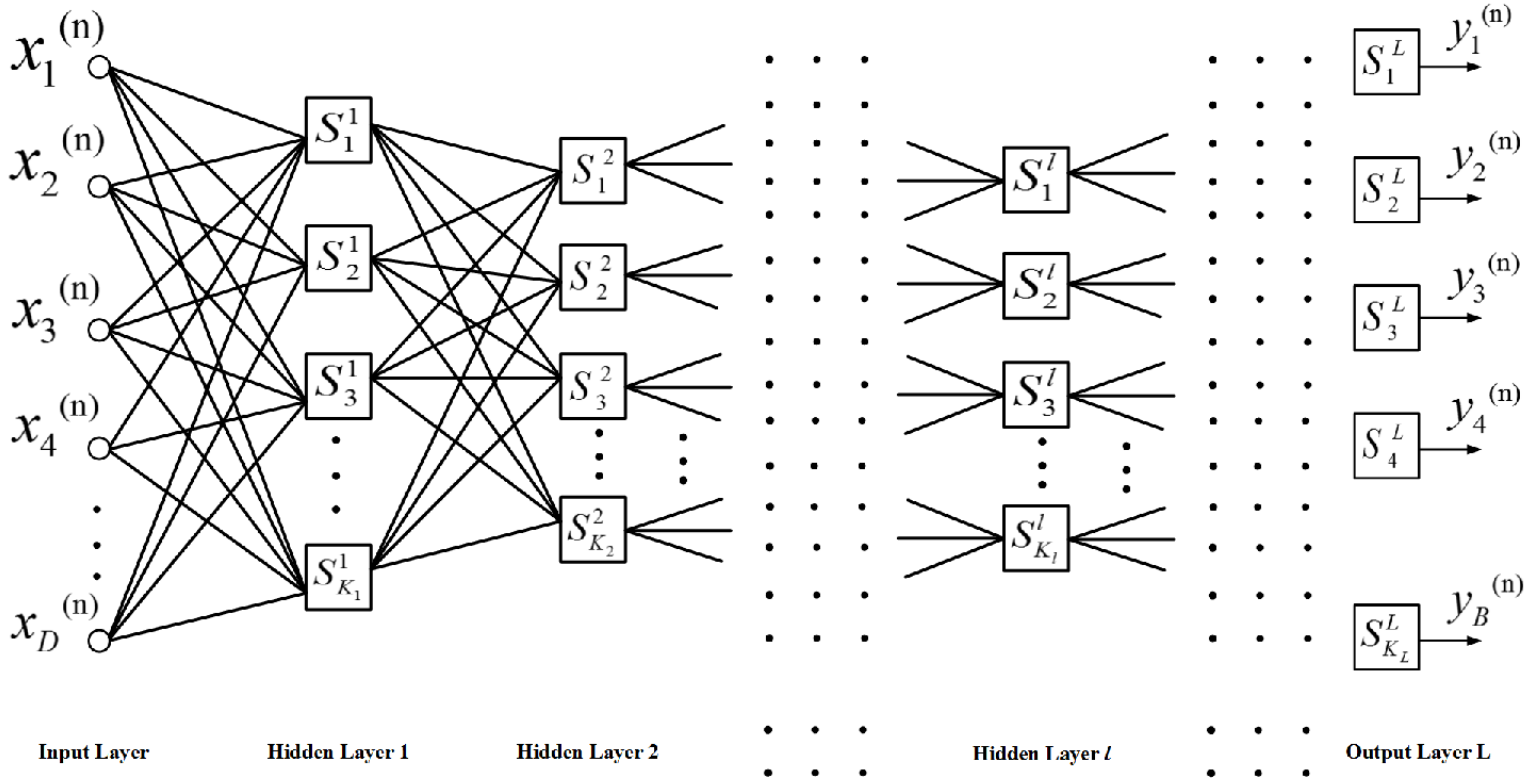
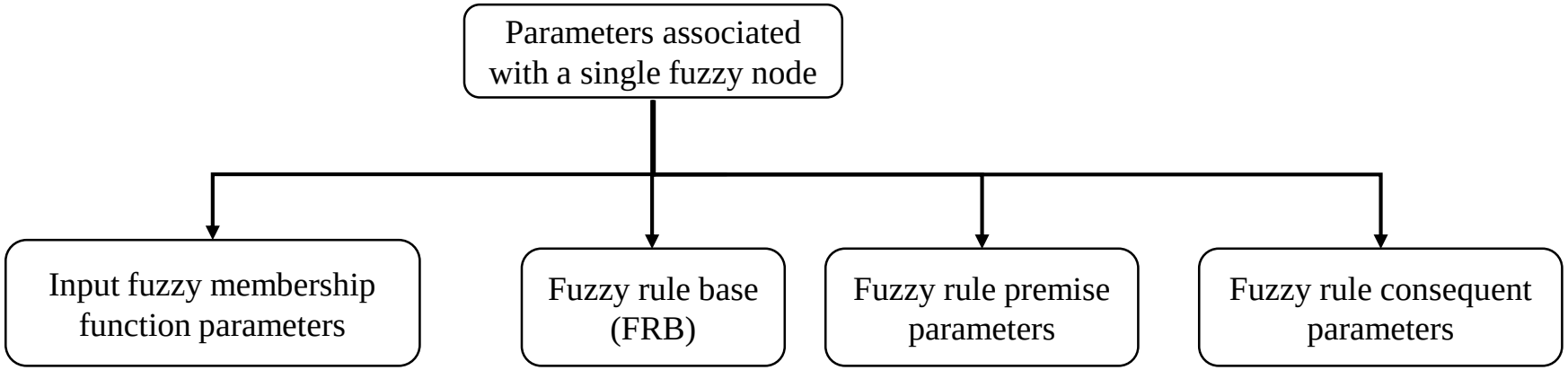


Fig : Architecture of MIMO DFN

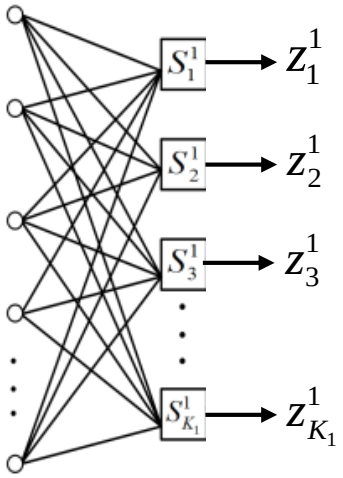


Rule 1 : IF z_1^{l-1} is $G_{1,1}^{l,k}$ AND z_2^{l-1} is $G_{2,1}^{l,k}$ AND ... AND z_d^{l-1} is $G_{d,1}^{l,k}$ AND ... AND $z_{K_{l-1}}^{l-1}$ is $G_{K_{l-1},1}^{l,k}$, THEN $z_k^l = p_{1,0}^{l,k} + p_{1,1}^{l,k} z_1^{l-1} + p_{1,2}^{l,k} z_2^{l-1} + \cdots + p_{1,d}^{l,k} z_d^{l-1} + \cdots + p_{1,K_{l-1}}^{l,k} z_{F_{l-1}}^{l-1}$.

Rule 2 : IF z_1^{l-1} is $G_{1,1}^{l,k}$ AND z_2^{l-1} is $G_{2,1}^{l,k}$ AND ... z_d^{l-1} is $G_{d,1}^{l,k}$ AND ... AND $z_{K_{l-1}}^{l-1}$ is $G_{K_{l-1},2}^{l,k}$, THEN $z_k^l = p_{2,0}^{l,k} + p_{2,1}^{l,k} z_1^l + p_{2,2}^{l,k} z_2^l + \cdots + p_{2,d}^{l,k} z_d^l + \cdots + p_{2,K_{l-1}}^{l,k} z_{F_{l-1}}^{l-1}$.

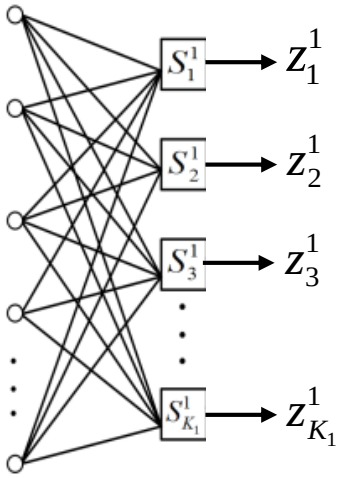
$\vdots$

Rule R_l : IF z_1^l is $G_{1,F_{l-1}}^{l,k}$ AND z_2^l is $G_{2,F_{l-1}}^{l,k}$ AND ... z_d^l is $G_{d,F_{l-1}}^{l,k}$ AND ... AND $z_{K_{l-1}}^l$ is $G_{K_{l-1},F_{l-1}}^{l,k}$, THEN $z_k^l = p_{R_l,0}^{l,k} + p_{R_l,1}^{l,k} z_1^{l-1} + p_{R_l,2}^{l,k} z_2^{l-1} + \cdots + p_{R_l,d}^{l,k} z_d^{l-1} + \cdots + p_{R_l,K_{l-1}}^{l,k} x_{K_{l-1}}$.

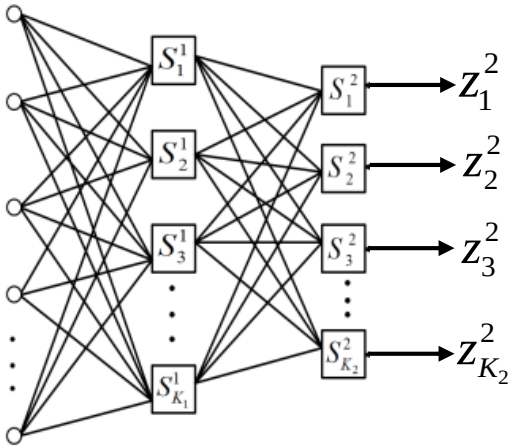


(a) Calculate output of first layer

Fig : Forward Pass in training a general L layered DFN

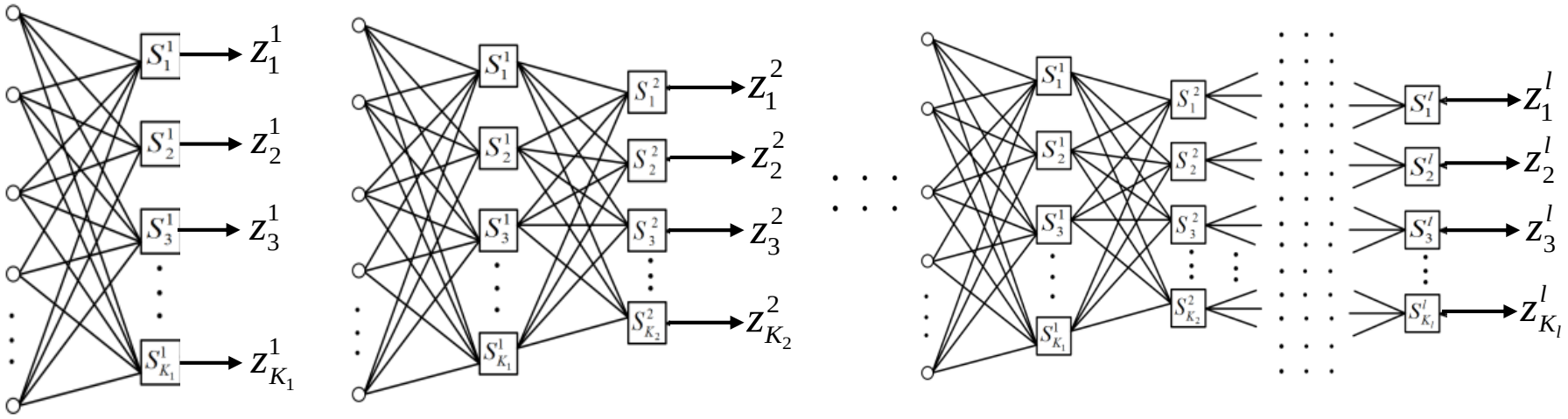


(a) Calculate output of first layer



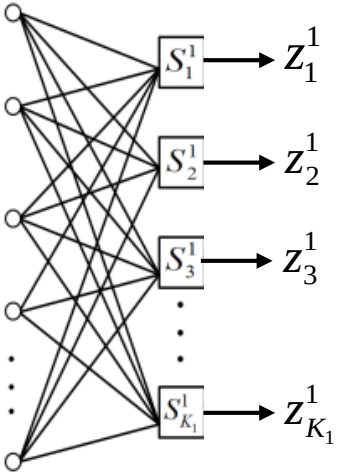
(b) Calculate output of second layer

Fig : Forward Pass in training a general *L* layered DFN

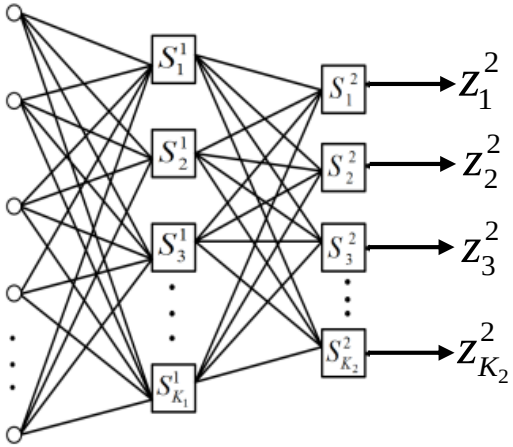


(a) Calculate output of first layer (b) Calculate output of second layer (c) Calculate output of layer l

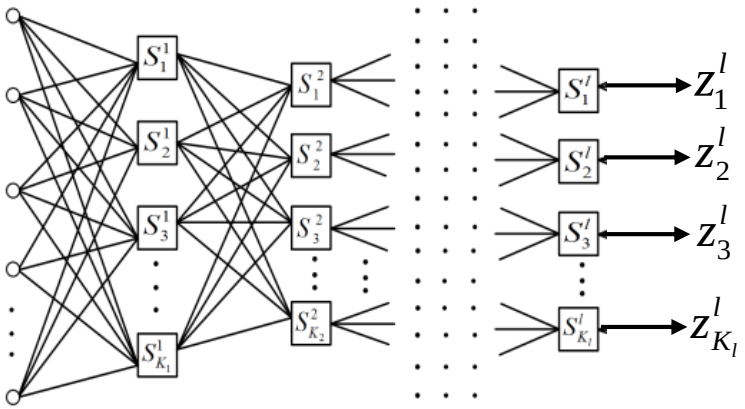
Fig : Forward Pass in training a general L layered DFN



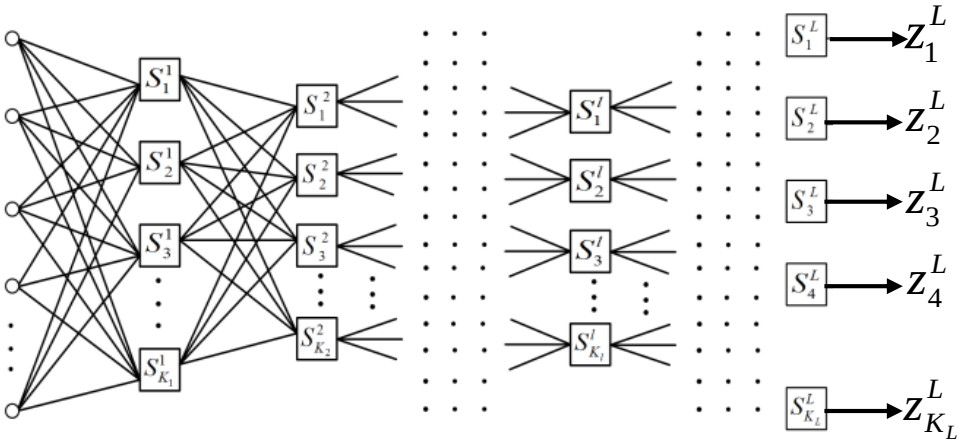
(a) Calculate output of first layer



(b) Calculate output of second layer

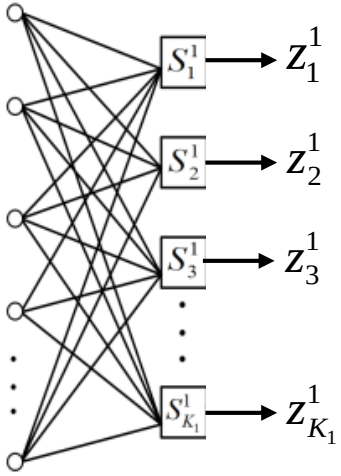


(c) Calculate output of layer l

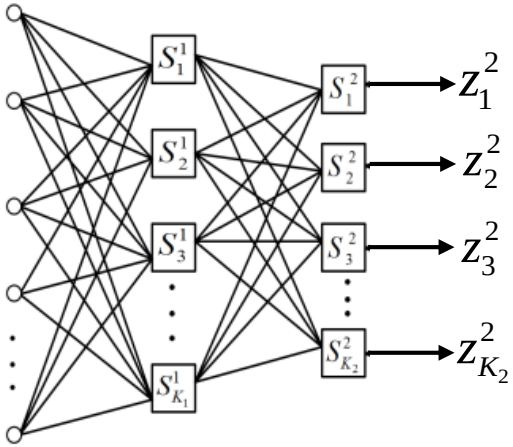


(d) Calculate output of final layer L

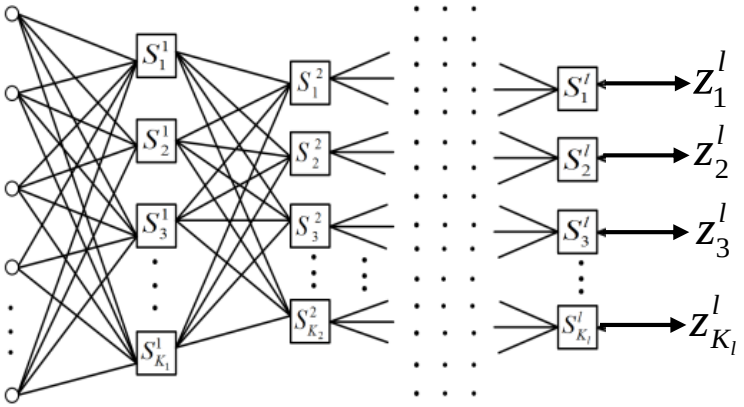
Fig : Forward Pass in training a general L layered DFN



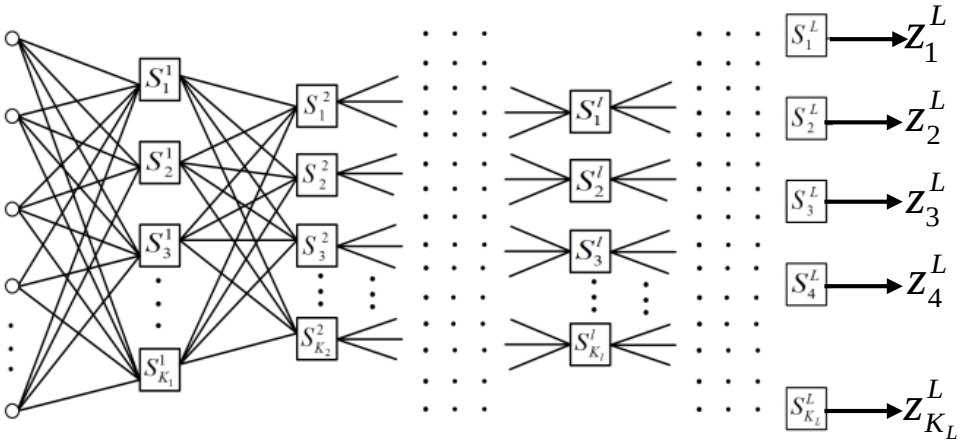
(a) Calculate output of first layer



(b) Calculate output of second layer



(c) Calculate output of layer l



(d) Calculate output of final layer L

Calculate the error at output layer and then calculate the cost function.

Fig : Forward Pass in training a general L layered DFN

Training DFN

- **Backward Pass :-**
  - Compute the gradient of cost function with respect to output layer parameters
  - Propagate the error in output layer backwards by using the chain rule
  - Compute gradients of cost function with respect to all the remaining network parameters
- Simultaneously update all the parameters using stochastic gradient descent

Calculate the gradients at output layer

$$(\nabla_{\Theta^L} J, \nabla_{P^L} J)$$

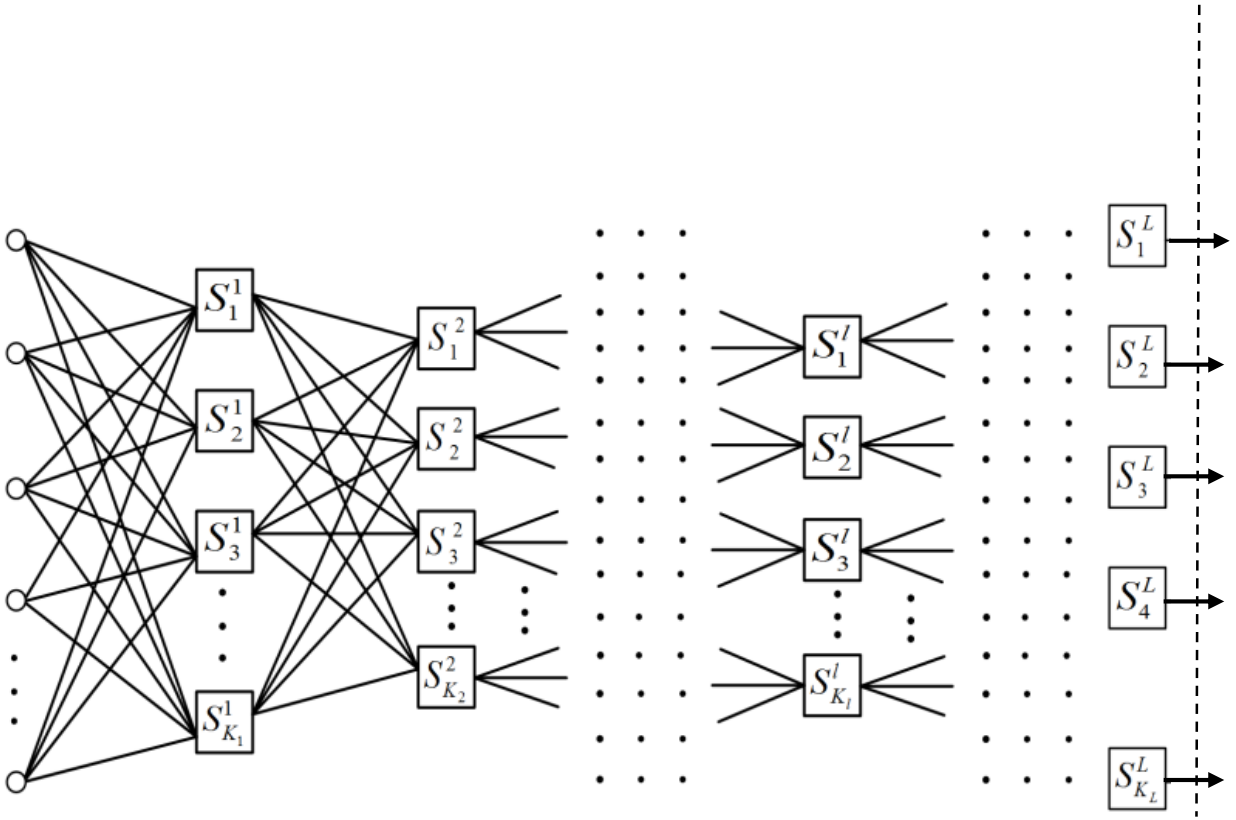


Fig : Backward Pass in training a general L layered DFN

Calculate the gradients at output layer

$$(\nabla_{\Theta^L} J, \nabla_{P^L} J)$$

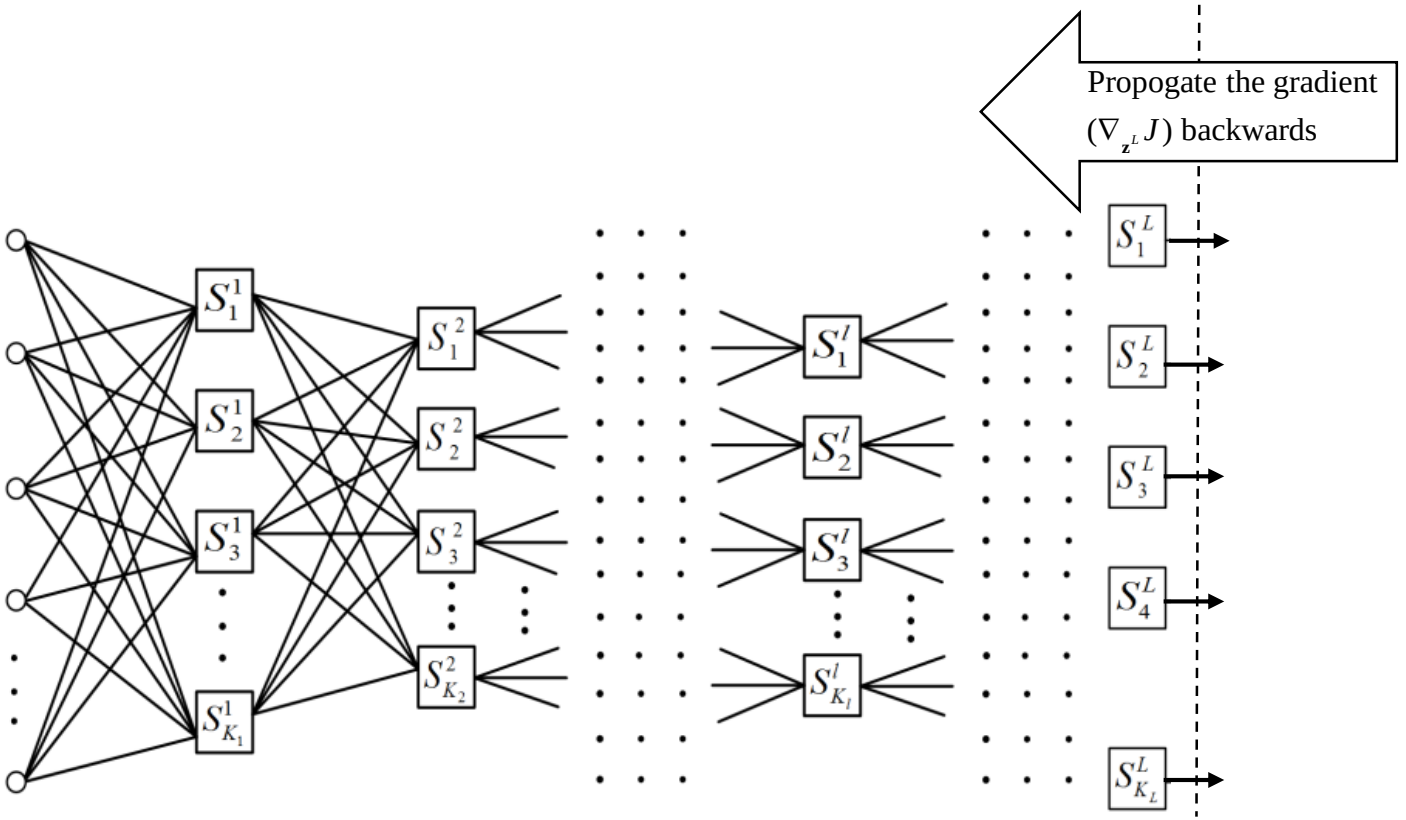


Fig : Backward Pass in training a general L layered DFN

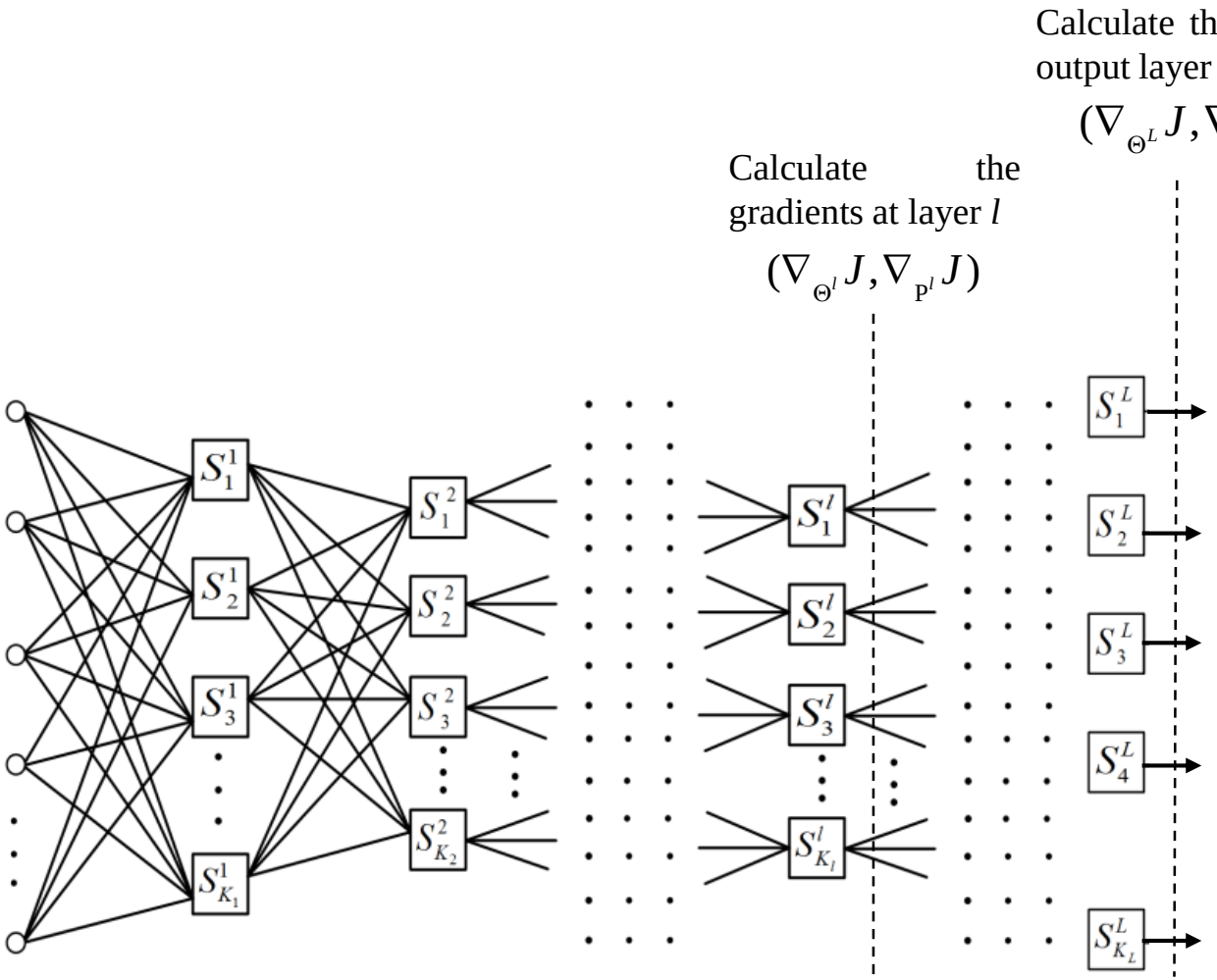


Fig : Backward Pass in training a general L layered DFN

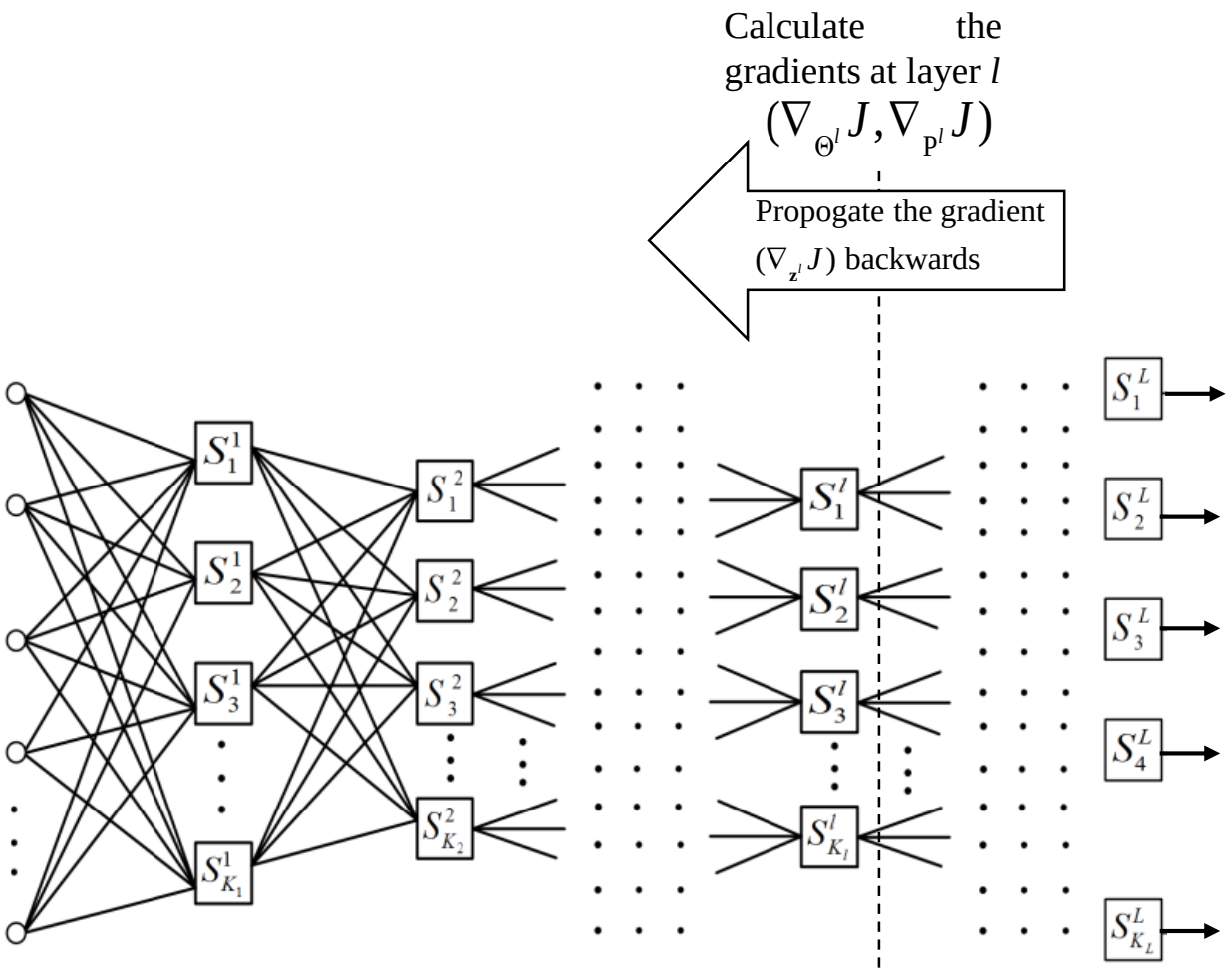


Fig : Backward Pass in training a general L layered DFN

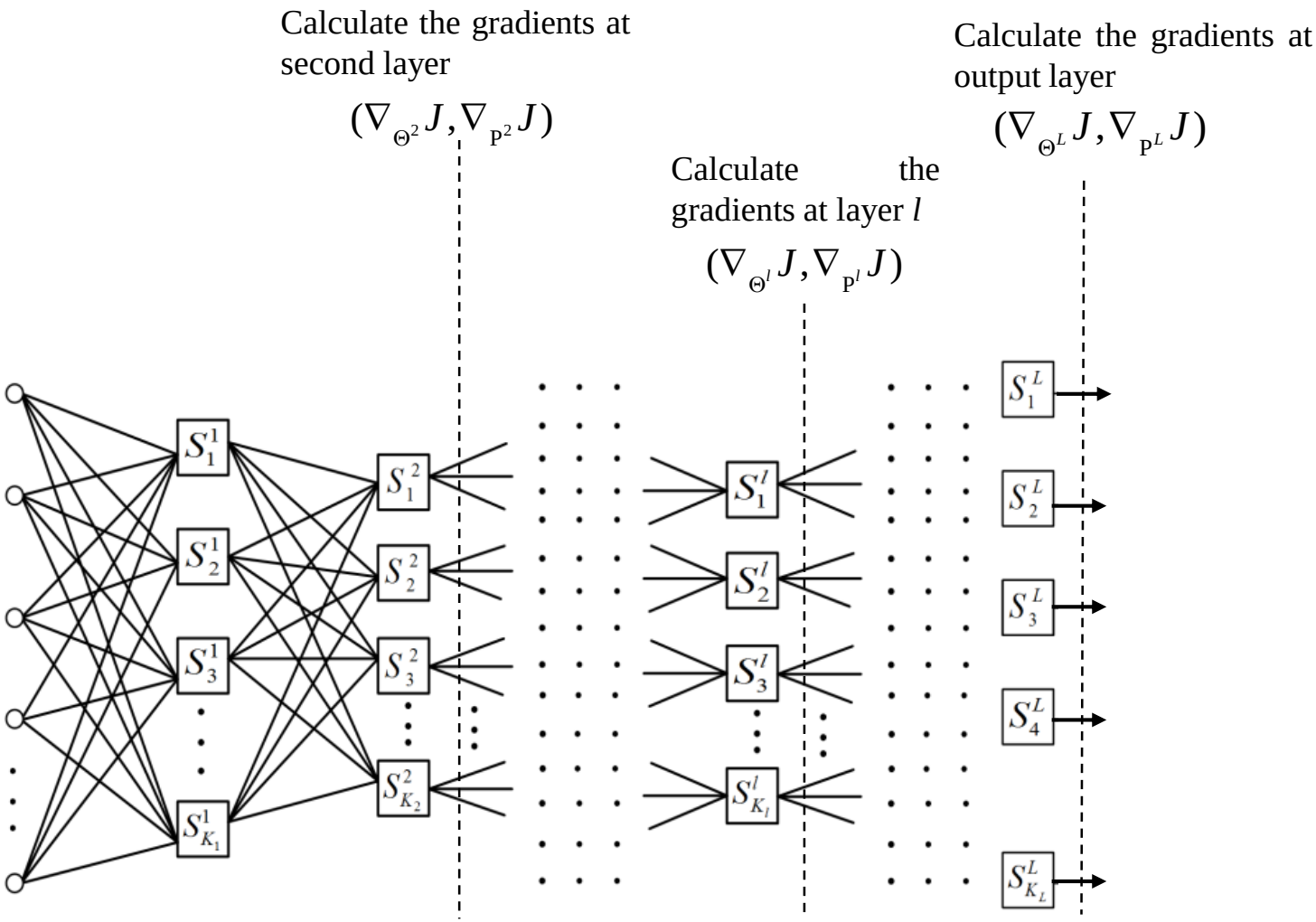


Fig : Backward Pass in training a general L layered DFN

Calculate the gradients at second layer

$(\nabla_{\Theta^2} J, \nabla_{P^2} J)$

Propogate the gradient $(\nabla_{z^2} J)$ backwards

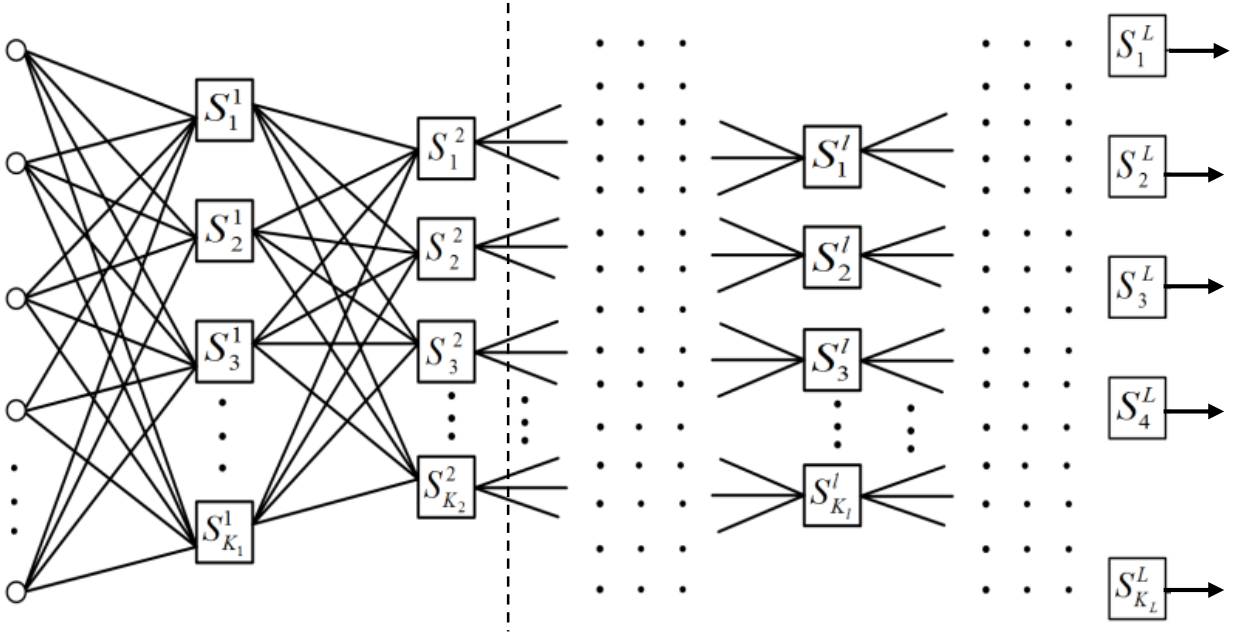


Fig : Backward Pass in training a general *L* layered DFN

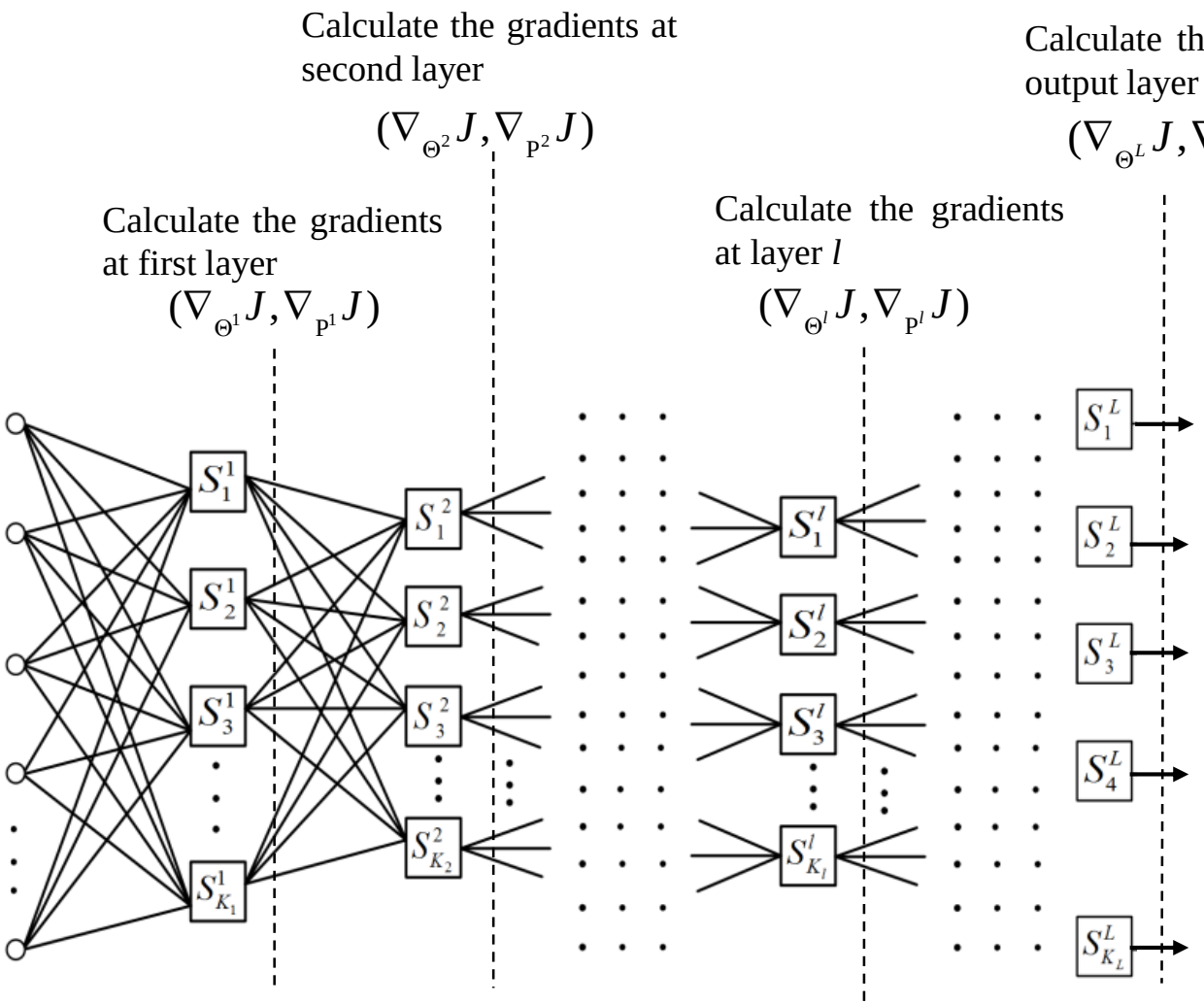
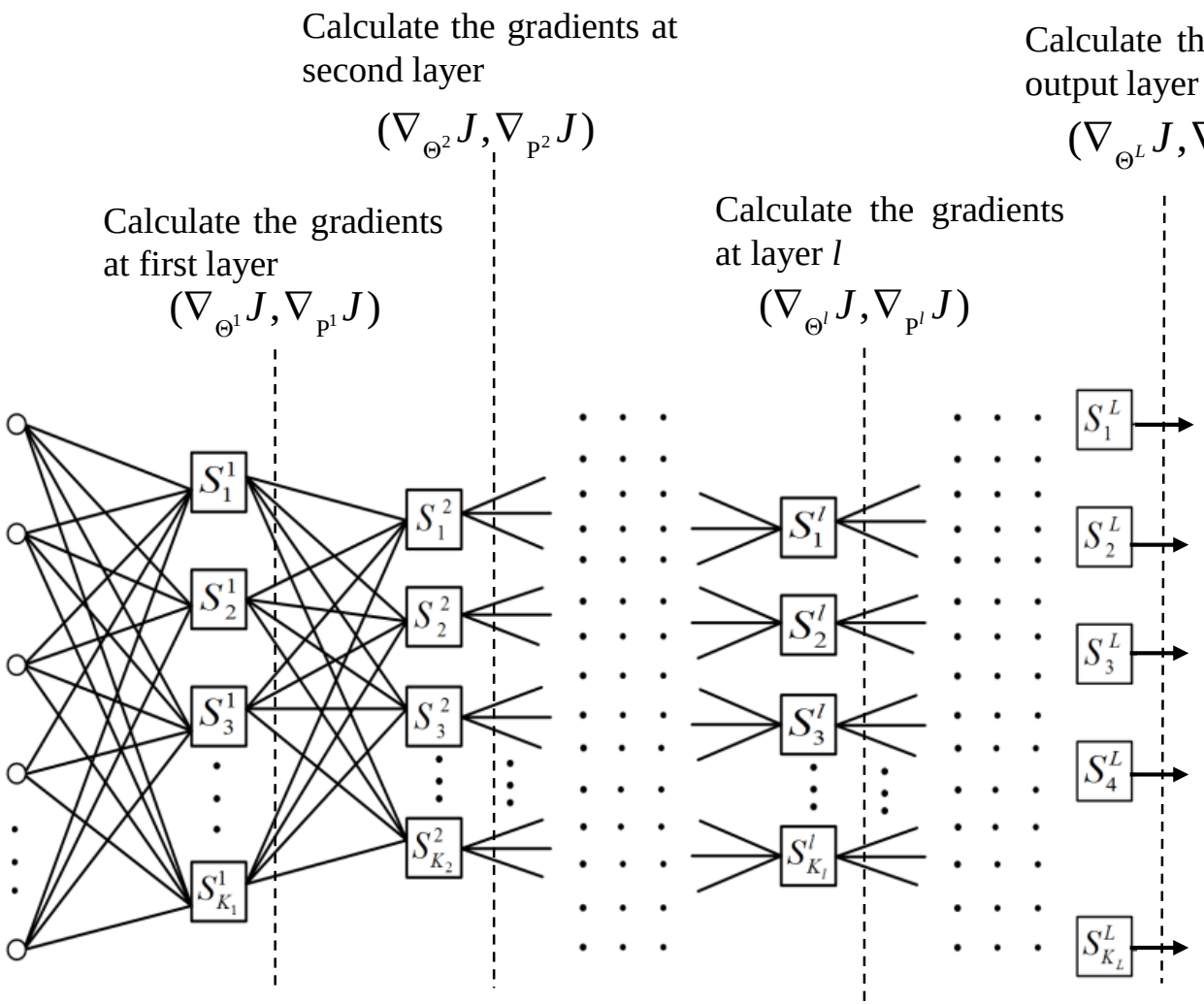


Fig : Backward Pass in training a general L layered DFN



Update Rule - $\Theta^l{}_{new} = \Theta^l{}_{old} - \eta \nabla_{\Theta^l} J, \mathbf{P}^l{}_{new} = \mathbf{P}^l{}_{old} - \eta \nabla_{P^l} J;$ for $l = 1, 2, ..., l, ..., L$

Fig : Update all the DFN parameters simultaneously

Training DFN

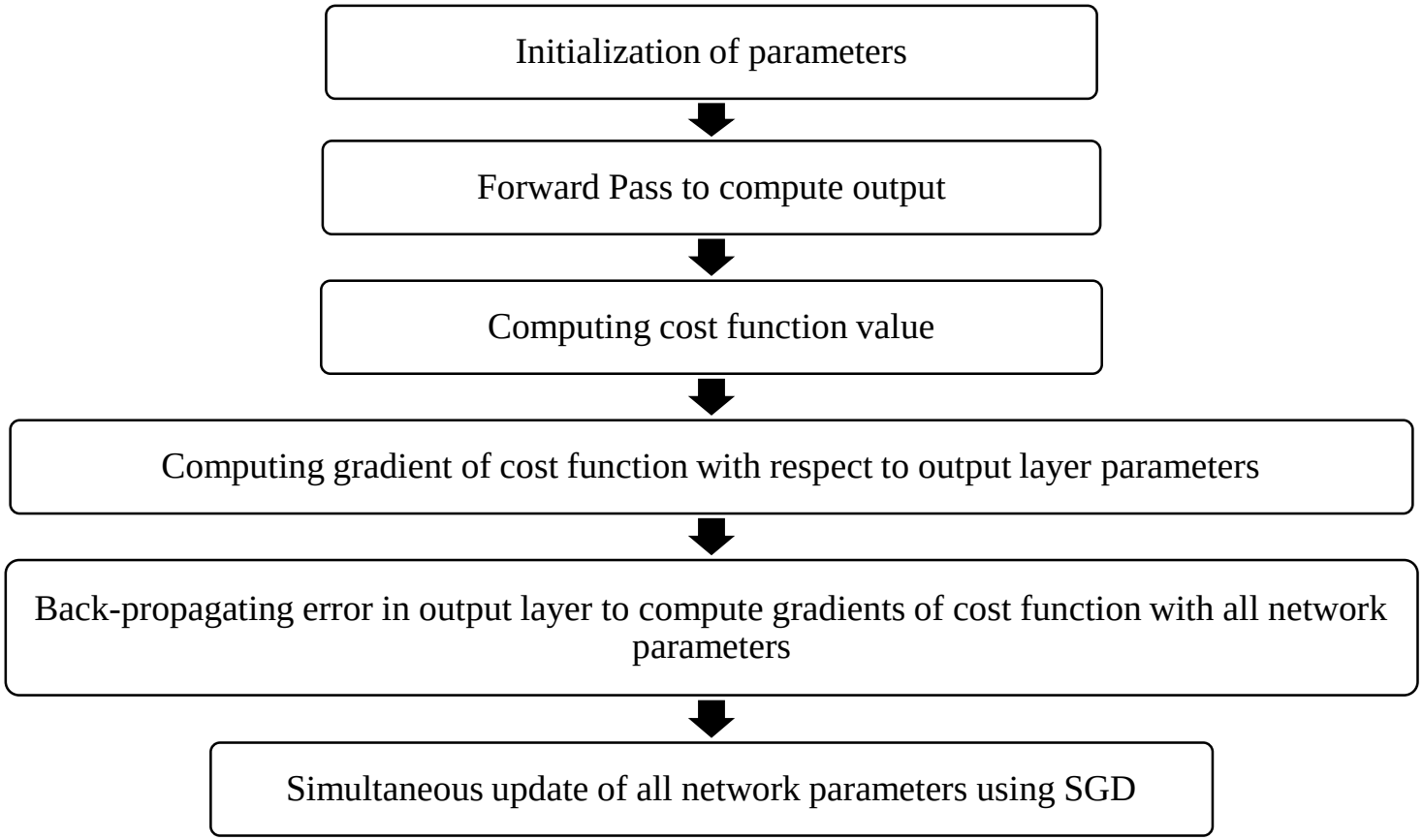
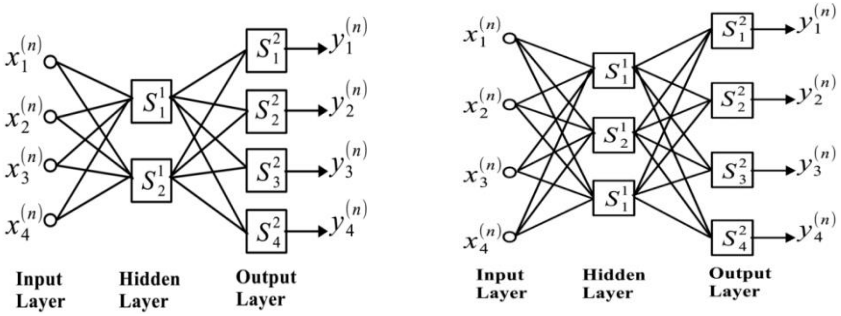


Fig : Flowchart of an iteration of training DFN

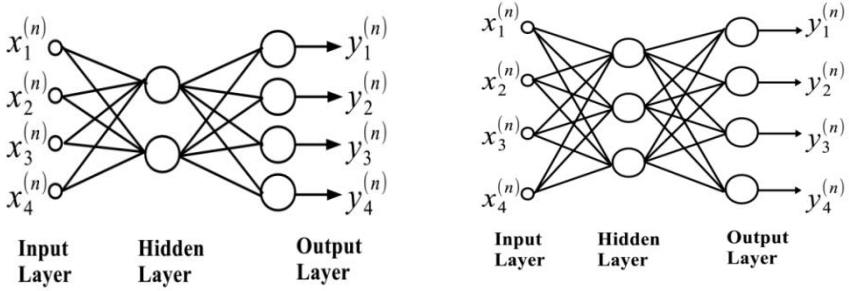
Application of DFN as Auto-Encoder

- Since DFN is capable of modelling highly intricate nonlinear and complex patterns in data, it can be used as auto-encoder for feature learning.



(a) two hidden nodes (b) three hidden nodes

Fig : DFN as auto-encoder



(a) two hidden nodes (b) three hidden nodes

Fig : Sparse Auto Encoder

Results of classification

Table 1: Comparison between softmax classifier accuracies of DFN auto-encoder and DNN auto-encoder

No. of hidden nodes (N_H)	Accuracy (in %)	
	DFN-AE	DNN-SAE
2	95.33	76.00
3	97.33	66.00

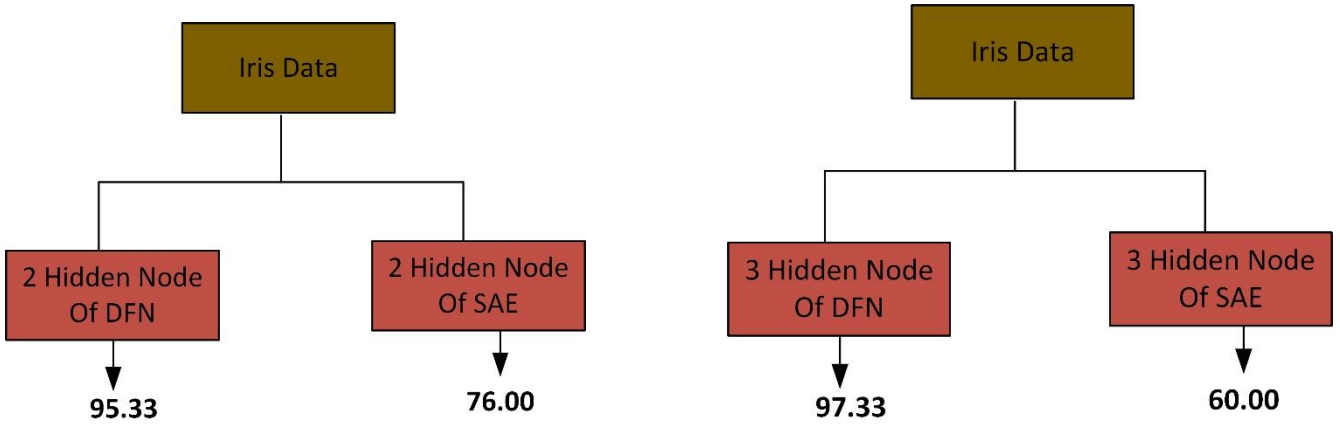


Table 2: Comparison between softmax classifier accuracies of DFN auto-encoder and DNN auto-encoder in presence of additive white Gaussian noise

No. of hidden nodes (N_H)	Accuracy (in %)													
	SNR=70		SNR=60		SNR=50		SNR=40		SNR=30		SNR=20		SNR=10	
	DFN -AE	DNN-SAE	DFN -AE	DNN-SAE	DFN -AE	DNN-SAE	DFN -AE	DNN-SAE	DFN -AE	DNN-SAE	DFN -AE	DNN-SAE	DFN -AE	DNN-SAE
2	**95.33**	48.00	**94.67**	48.00	**95.33**	48.00	**94.00**	47.33	**96.67***	47.33	**87.33**	34.67	**72.00**	33.33
3	**97.33***	52.00	**97.33***	48.00	**97.00**	48.00	**94.33**	47.33	**89.67**	46.67	**84.33**	39.67	**84.33**	36.00

* in a row indicates the best performance of softmax classifier for DFN-AE with corresponding number of hidden nodes among various SNR valued AWGN affected data used for training DFN-AE and DNN-SAE

For more details please refer to:

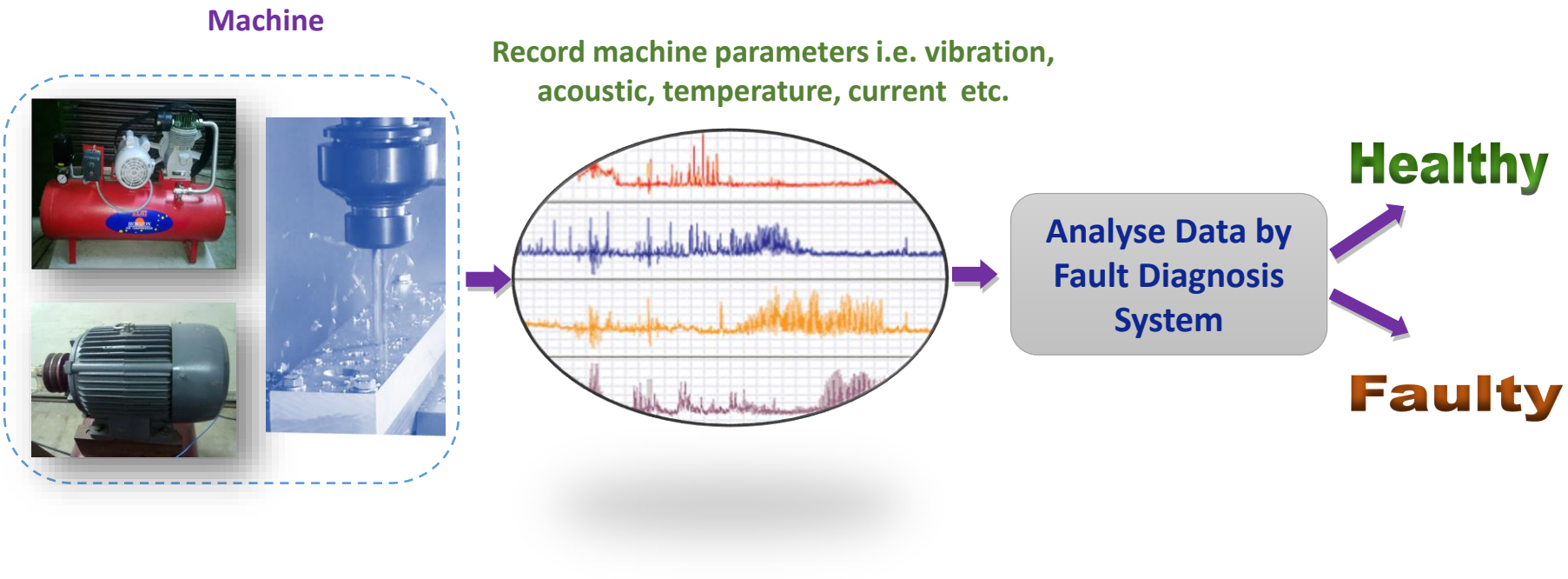
Shreedharkumar Rajurkar and **Nishchal K. Verma**, “Developing Deep Fuzzy Network with Takagi Sugeno Fuzzy Inference System”, IEEE International Conference on Fuzzy Systems, 2017 (FUZZ-IEEE 2017).

Thank you

Case study-1
DNN for Machine Fault Diagnosis

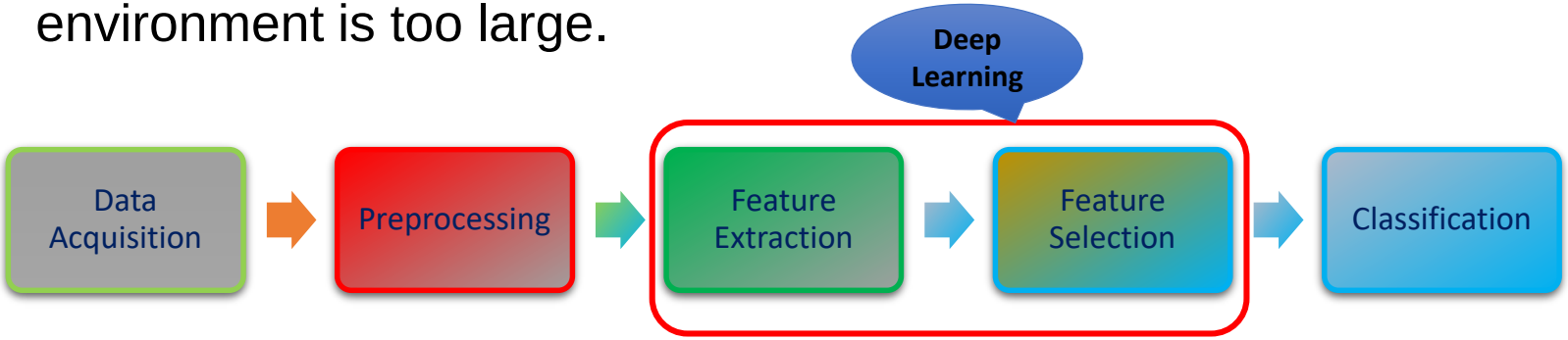
Machine Fault Diagnosis

Fault diagnosis system is a decision system which includes detection and identification of faults.



Key Challenges

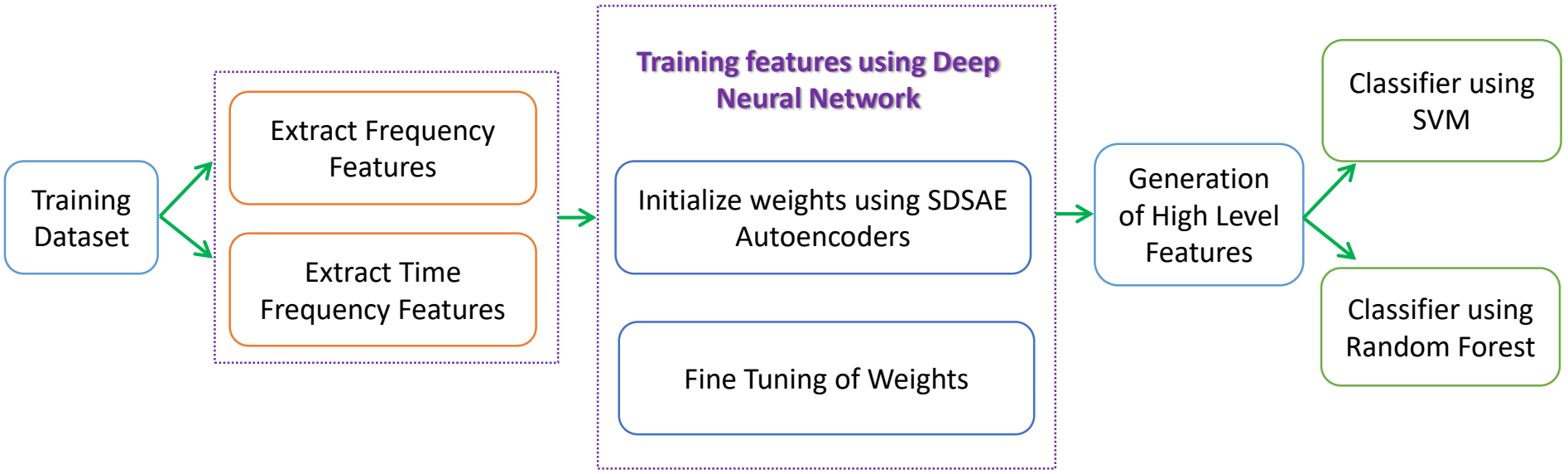
- **Manual feature extraction** requires domain knowledge or diagnostic expertise.
- **Learning complex non-linear relationships** from machine data which is non stationary in nature.
- **Size of data** collected from large no. of sensors in industrial environment is too large.



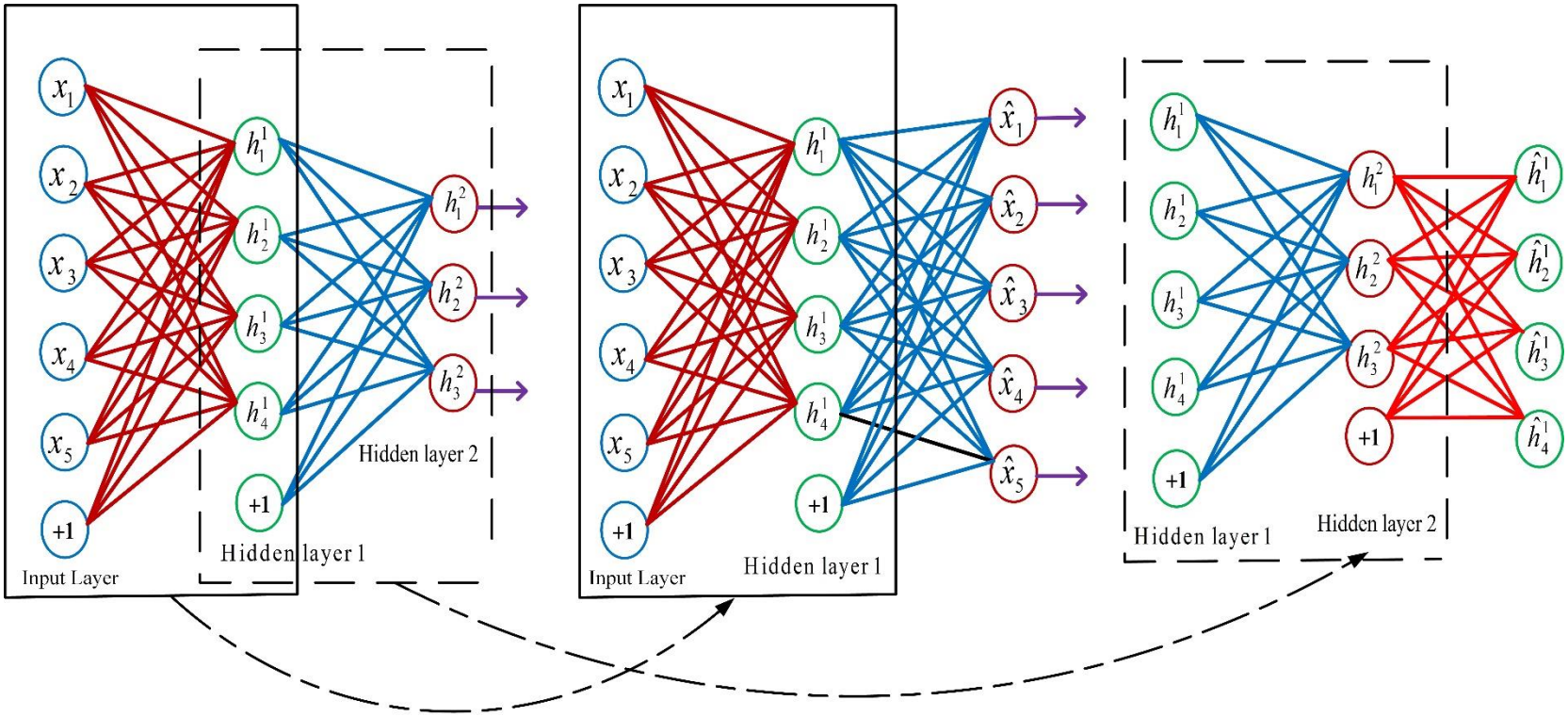
- **Deep learning** holds the potential to overcome the aforementioned deficiencies in current fault diagnosis methods.

Framework based on Deep Learning

- Benchmark dataset was taken and features were obtained via frequency and time frequency analysis. Afterwards high level features were generated using DNN. The output of the deep network served as the input for classifiers for final class prediction.

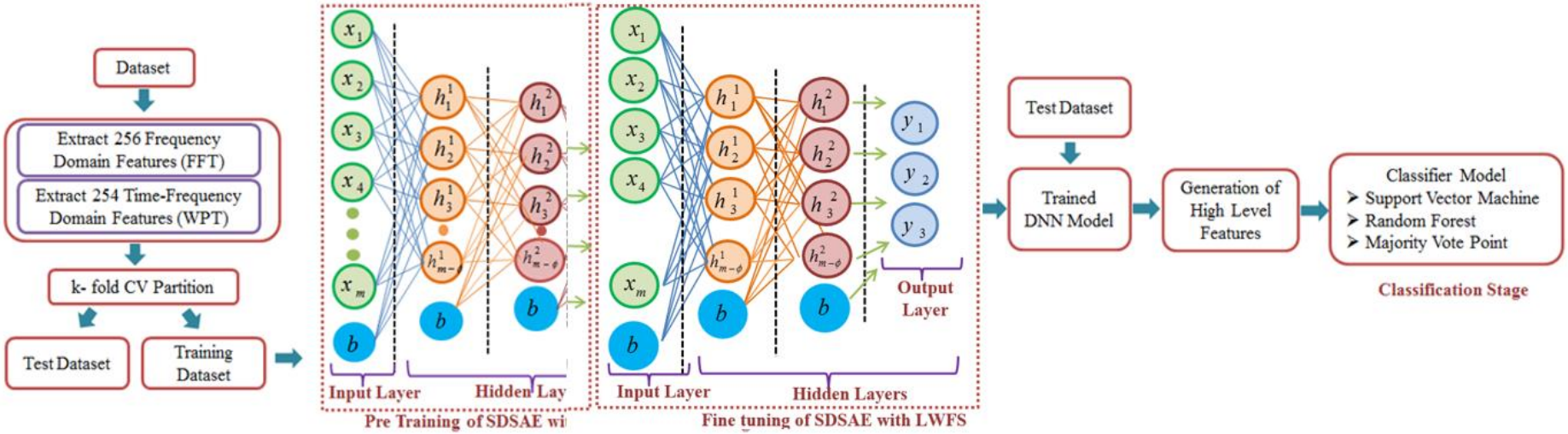


Fault Diagnosis Framework 1



Stacked Autoencoder Autoencoder 1 Autoencoder 2

Fault Diagnosis Framework 1

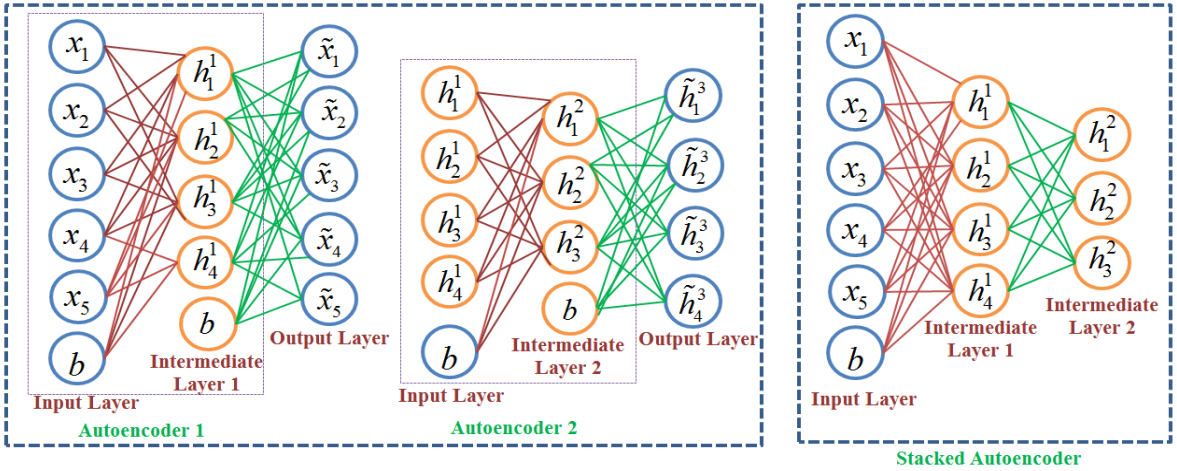


- Network Architecture (Air compressor, Drill bit, IMS Rexford, Bearing fault)
  - Input Layer: 256/254 nodes
  - 1st Hidden layer: 125 nodes
  - 2nd Hidden Layer: 49 nodes

Generation of High Level Features using DNN

It is the training phase where features are further processed to capture complex and non linear relationship present among data.

- **Initialization of weights:** Herein network weights are initialized using weights obtained from training a stacked autoencoder.



- **Finetuning:** In this part, the previously initialized weights are tailored for the given classification problem to give better performance.

Experimentation…Dataset

- **Air compressor dataset:** The dataset [54] comprises of 1800 acoustic recordings evenly spread over 8 different states of a single stage reciprocating air compressor. A set of 225 recordings of five second duration was collected for each of the eight compressor states at a sampling rate of 50kHz.

- **Drill bit dataset:** This dataset [55] includes vibration recordings of 4 different drill bit states namely Healthy state, Flank wear state, Chisel wear state and Outer Corner wear state. The dataset comprises of 120 recordings for each of the 4 states, collected at varying feed rates and cutting speeds of the drill bit.

Results with SVM (in % Accuracy)

Dataset / Features	FFT	WPT	Deep Learning based on FFT features				Deep Learning based on WPT features			
			SAE		SDAE		SAE		SDAE	
			FT1	FT2	FT1	FT2	FT1	FT2	FT1	FT2
Air Compressor	94.67	95.83	99.44	**99.72**	**99.74**	99.61	99.22	**99.82**	**99.61**	99.28
Drill Bit	93.3	96.46	**99.79**	99.58	99.58	**100.00**	**99.1**	98.21	**99.38**	90.21

Conclusions

- DFM has the capability to handle more complexity and abstraction in data with smaller architecture compared with DNN.
- Working of trained model is understandable to human beings and underlying working can easily be comprehended by human beings by just looking at network parameters.
- DFN has the asset of dealing with uncertainty of various kinds such as vagueness, ambiguity, imprecision, etc.
- DFM is robust in presence of noise in data.
- Prior human supervisor knowledge can be easily incorporated into the architecture of DFN.
- Closely matches with human cognitive thinking

References

- R.J. Alfredson and J. Mathew, “Time domain methods for monitoring the condition of rolling element bearings,” Technical Report, NASA STI/Recon, A 86: 22166, 1984
- J. Meier , C. Raymond and S. V. Gable, “Frequency domain engine defect signal analysis,” U.S. Patent 4,424,709, Jan., 1984.
- Z. J. He, Z. YanYang, Q. F. Meng and J. Zhao, “Fault diagnosis principle of non-stationary signal and applications to mechanical equipment,” ,2001, pp. no. 133-146
- G.E. Hinton, and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786,pp. 504-507, 2006.
- V. Pascal, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” The Journal of Machine Learning Research, vol. 11, pp. no. 3371-3408, 2011.
- N. K. Verma, R. K. Sevakula, S. Dixit and A. Salour, “Intelligent Condition Based Monitoring Using Acoustic Signals for Air Compressors,” in IEEE Transactions on Reliability, vol. 65, no. 1, pp. 291-309,Mar. 2016.
- N.K. Verma, S. Khatravath, A. Salour, “Cost Benefit Analysis for Condition Based Maintenance,” IEEE Conference on Prognostics and Health Management, Maryland, USA, June 2013, pp. 1-6.

- R.J. Alfredson and J. Mathew, “Time domain methods for monitoring the condition of rolling element bearings,” Technical Report, NASA STI/Recon, A 86: 22166, 1984
- J. Meier , C. Raymond and S. V. Gable, “Frequency domain engine defect signal analysis,” U.S. Patent 4,424,709, Jan., 1984.
- Z. J. He, Z. YanYang, Q. F. Meng and J. Zhao, “Fault diagnosis principle of non-stationary signal and applications to mechanical equipment,” ,2001, pp. no. 133-146
- G.E. Hinton, and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786,pp. 504-507, 2006.
- V. Pascal, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” The Journal of Machine Learning Research, vol. 11, pp. no. 3371-3408, 2011.
- N. K. Verma, R. K. Sevakula, S. Dixit and A. Salour, “Intelligent Condition Based Monitoring Using Acoustic Signals for Air Compressors,” in IEEE Transactions on Reliability, vol. 65, no. 1, pp. 291-309,Mar. 2016.
- N.K. Verma, S. Khatravath, A. Salour, “Cost Benefit Analysis for Condition Based Maintenance,” IEEE Conference on Prognostics and Health Management, Maryland, USA, June 2013, pp. 1-6.

Thank you