# Ensembles of Diverse Neural Networks

Dr. Md. Aminul Haque Akhand

# Outline of the Presentation

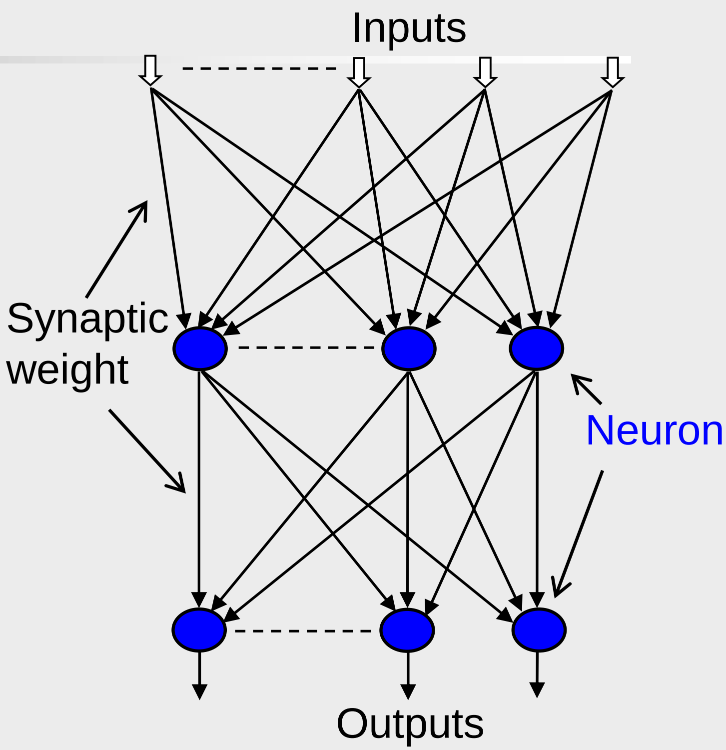
1. Introduction
2. Background of Ensemble Construction
3. Data Sampling for Ensemble Construction
4. Candidate Ensemble Methods
5. Experimental Studies
6. Motivation to better NNE Construction

# Artificial Neural Network

According to Haykin, an artificial neural network (NN) is a collection of simple processing units and it has ability to store experimental knowledge.

**NN resembles the human brain in two respects:**

1. *Knowledge is acquired by a NN from its environment through a learning process.*
2. *Interneuron connection strengths, known as synaptic weights, are used to store the knowledge.*

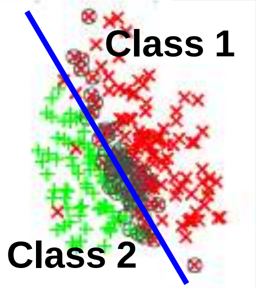



Artificial Neural Network

❖ Functionality of a NN depends on the synaptic weight values and the aim of learning is to get proper weight set.

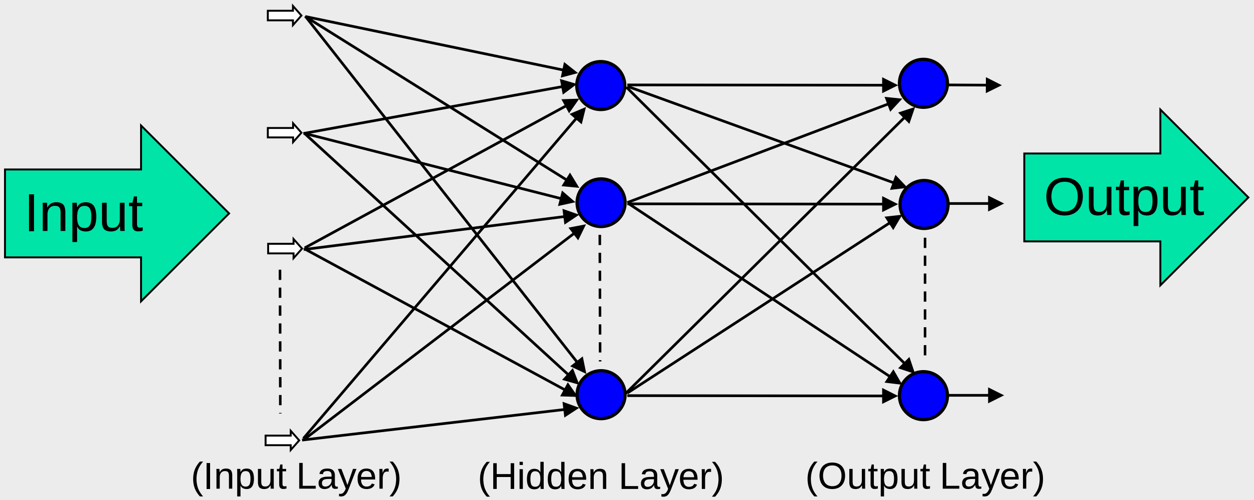
**NNs have been successfully applied for various task; for classification it performs well.**

# NNs for Classification Tasks




- ❖ Classification is one of the most frequently encountered decision making tasks of human activity.
- ❖ It occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object.

**Back-Propagation (BP) Learning Algorithm for Classification**




**Error = Desired Output - Actual Output**

Forward Pass: Error is calculated based on desired and actual output.

Backward Pass: Synaptic weights are adjusted based on calculated error.

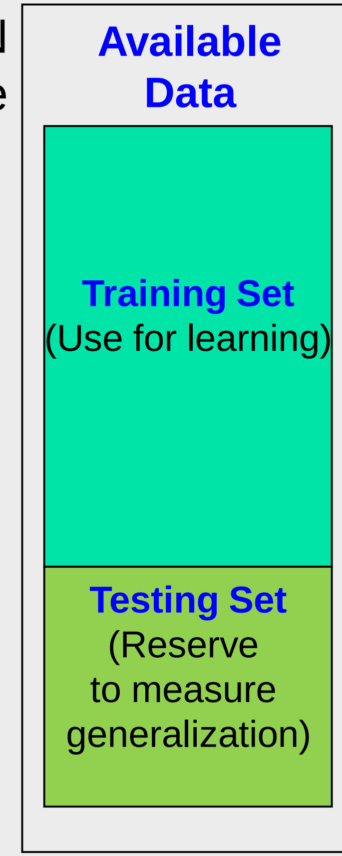
- ❖ At a time small fraction of a weight is corrected w. r. t demand correction for smooth learning; a parameter learning rate($\eta$) defines the relative size to change.

# Performance Measures

Learning and generalization are the most important topics in NN research. Learning is the ability to approximate the training data while generalization is the ability to predict well beyond the training data.

- Generalization is more desirable because the common use of a NN is to make good prediction on new or unknown objects.
- It measures on the testing set that is reserved from available data and not use in the training.
- Testing error rate (TER), i.e., rate of wrong classification on testing set, is widely acceptable quantifying measure, which value minimum is good.

$$TER = \frac{\text{Total testing set misclassified patterns}}{\text{Total testing set patterns}}$$




**Benchmark problems are used to measure TER.**

# Benchmark Problems for Evaluation

A benchmark is a point of reference by which something can be measured.

- For NN or machine learning, the most popular benchmark dataset collection is the University of California, Irvine (UCI) Machine Learning Repository (http://archive.ics.uci.edu/ml/).
- UCI contains raw data that require preprocessing to use in NN. Some preprocessed data is also available at Proben1 (ftp://ftp.ira.uka.de/pub/neuron/).
- Various persons or groups also maintain different benchmark datasets for specific purpose: Delve (www.cs.toronto.edu/~delve/data/datasets.html), Orange (www.ailab.si/orange/datasets.asp), etc.
- In this study 32 benchmark problems are considered from UCI. Well defined benchmark methodology is followed for preprocessing.

UCI Machine Learning Repository - Windows Internet Explorer

http://archive.ics.uci.edu/ml/

UCI Machine Learning Repository

UCI

# Machine Learning Repository

Center for Machine Learning and Intelligent Systems

## Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 174 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. Our old web si
format. For a general overview of the Repository, please visit our About page. For information about citing data sets in publications, please read our citation policy. If
donation policy. For any other questions, feel free to contact the Repository librarians. We have also set up a mirror site for the Repository.

Supported By: NATIONAL SCIENCE FOUNDATION In Collaboration With: Rexa.info • Research • People • Connections

**Latest News:**

07-23-2008: Repository mirror has been set up.
03-24-2008: New data sets have been added!
06-25-2007: Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope
04-13-2007: Research papers that cite the repository have been associated to specific data sets.
04-09-2007: Three new data sets have been added: Poker Hand, CalIt2 Building People Counts, Dodgers Loop Sensor.
09-08-2006: The Beta site has been launched.
09-01-2006: SPECTF.test has been modified by the donor.

**Newest Data Sets:**

06-26-2008: UCI Parkinsons
04-21-2008: UCI Ozone Level Detection
04-03-2008: UCI Abscisic Acid Signaling Network
03-20-2008: Hill-Valley

**Most Popular Data**

39351:
31585:
26458:
23553:

# Benchmark Problems

## Problems Related to Human Life

| Problem | Task |
|---|---|
| **Breast Cancer Wisconsin** | Predicts whether a tumor is benign (not dangerous to health) or malignant (dangerous) based on a sample tissue taken from a patient's breast. |
| **BUPA Liver Disorder** | Identify lever disorders based on blood tests along with other related information such as alcohol consumption. |
| **Diabetes** | Investigate whether the patient shows or not the signs of diabetes. |
| **Heart Disease Cleveland** | Predicting whether at least one of four major heart vessels is reduced in diameter by more than 50%. |
| **Hepatitis** | Anticipate status (i.e., live or die) of hepatitis patient. |
| **Lymphography** | Predict the situation of lymph nodes and lymphatic vessels. |
| **Lungcancer** | Identify types of pathological lung cancers. |
| **Postoperative** | Determine place to send patients for postoperative recovery. |

# Benchmark Problems

## Problems Related to Finance

| Problem | Task |
|---|---|
| **Australian Credit Card** | Classify people as good or bad credit risks depend on applicants' particulars. |
| **Car** | Evaluate cars based on price and facilities. |
| **Labor Negotiations** | Identify a worker as good or bad i.e., contract with him beneficial or not. |
| **German Credit Card** | Like Australian Card, this problem also concerns to predict the approval or non-approval of a credit card to a customer. |

## Problems Related to Plants

| Problem | Task |
|---|---|
| **Iris Plants** | Classify iris plant types. |
| **Mushroom** | Identify whether a mushroom is edible or not based on a description of the mushroom's shape, color, odor, and habitat. |
| **Soybean** | Recognize 19 different diseases of soybeans. |

# Summary of Benchmark Problems

| Abbr. | Problem | Total Examp | Input Features | | NN Architecture | | |
|---|---|---|---|---|---|---|---|
| | | | Cont. | Discr. | Inputs | Class | Hidd. Node |
| ACC | Australian Credit Card | 690 | 6 | 9 | 51 | 2 | 10 |
| BLN | Balance | 625 | - | 4 | 20 | 3 | 10 |
| BCW | Breast Cancer Wisconsin | 699 | 9 | - | 9 | 2 | 5 |
| CAR | Car | 1728 | - | 6 | 21 | 4 | 10 |
| DBT | Diabetes | 768 | 8 | - | 8 | 2 | 5 |
| GCC | German Credit Card | 1000 | 7 | 13 | 63 | 2 | 10 |
| HDC | Heart Disease Cleveland | 303 | 6 | 7 | 35 | 2 | 5 |
| HPT | Hepatitis (HPT) | 155 | 6 | 13 | 19 | 2 | 5 |
| HTR | Hypothyroid | 7200 | 6 | 15 | 21 | 3 | 5 |
| HSV | House Vote | 435 | - | 16 | 16 | 2 | 5 |
| INS | Ionosphere | 351 | 34 | - | 34 | 2 | 10 |
| KRP | King+Rook vs King+Pawn | 3196 | - | 36 | 74 | 2 | 10 |
| LMP | Lymphography | 148 | - | 18 | 18 | 4 | 10 |
| PST | Postoperative | 90 | 1 | 7 | 19 | 3 | 5 |
| SBN | Soybean | 683 | - | 35 | 82 | 19 | 25 |
| SNR | Sonar | 208 | 60 | - | 60 | 2 | 10 |
| SPL | Splice Junction | 3175 | - | 60 | 60 | 3 | 10 |
| WIN | Wine | 178 | 13 | - | 13 | 3 | 5 |
| WVF | Waveform | 5000 | 21 | - | 21 | 3 | 10 |
| ZOO | Zoo | 101 | 15 | 1 | 16 | 7 | 10 |

**Input Features of Diabetes**

1. Number of times pregnant
2. Plasma glucose concentration
3. Diastolic blood pressure
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index
7. Diabetes pedigree function
8. Age

❖Problems show variations in number of examples, input features and classes.

# Why Ensemble of Neural Networks?

- The idea of building an ensemble with several NNs is taken from sociology.

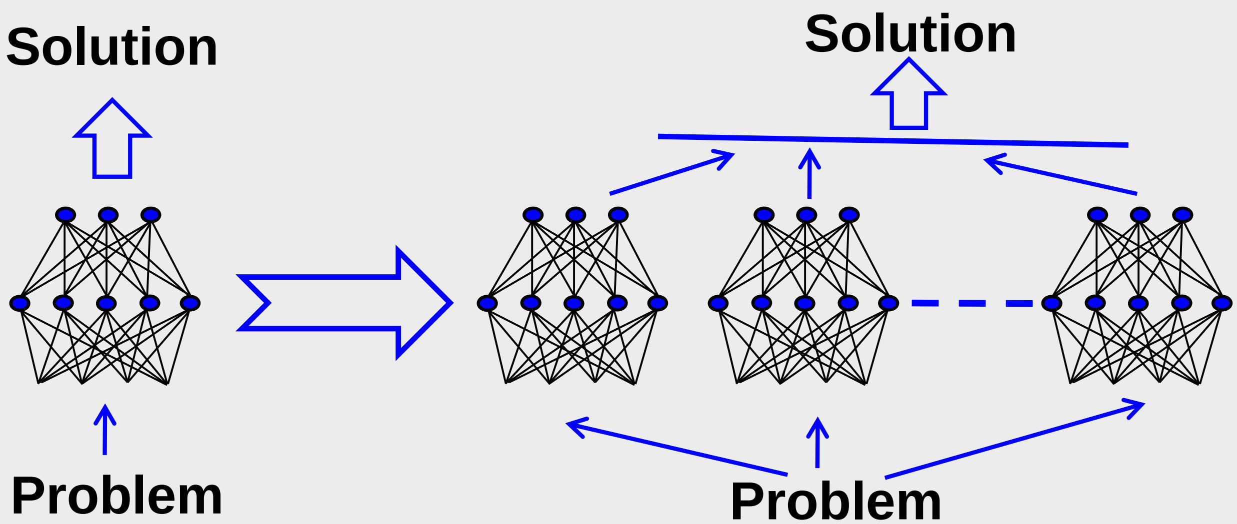
- A committee of people for an important task or building board of doctors for a major operation is a common matter.

- Each member of the committee should be as competent as possible, but they should be complementary to one another. If one or a few members make an error, the probability is high that the remaining members can correct his error.

- Several NNs together might perform better than single NN when they maintain proper diversity to compensate failure of one by others.

**The goal of ensemble is to achieve better generalization (i.e., lower TER) through producing diverse NNs.**

# Neural Network Ensemble (NNE)




In an NNE, component NNs solve the problem individually and combine their outputs for NNE's output. For better performance diversity among NNs is important.

- Diversity means disagreement among the NNs. Pair wise plain disagreement (*PD*) measure is the most popular among various measuring techniques.
- For a NNs pair, diversity *(div)* is equal to the proportion of the patterns on which NNs reply different class predictions. The total NNE diversity (*div_ens)* is the average for all the pairs.

$$div_{i.j} = \frac{1}{N}\sum_{n=1}^{N} Diff\,(C_i(x_n), C_j(x_n)),$$

$$div\_ens = \frac{[div_{i.j} \text{ for all NNs pairs }]}{\text{Total NNs pair}} = \frac{\sum_{i=1}^{M-1}\sum_{j=i+1}^{M} div_{i.j}}{\sum_{m=1}^{M-1} m}$$

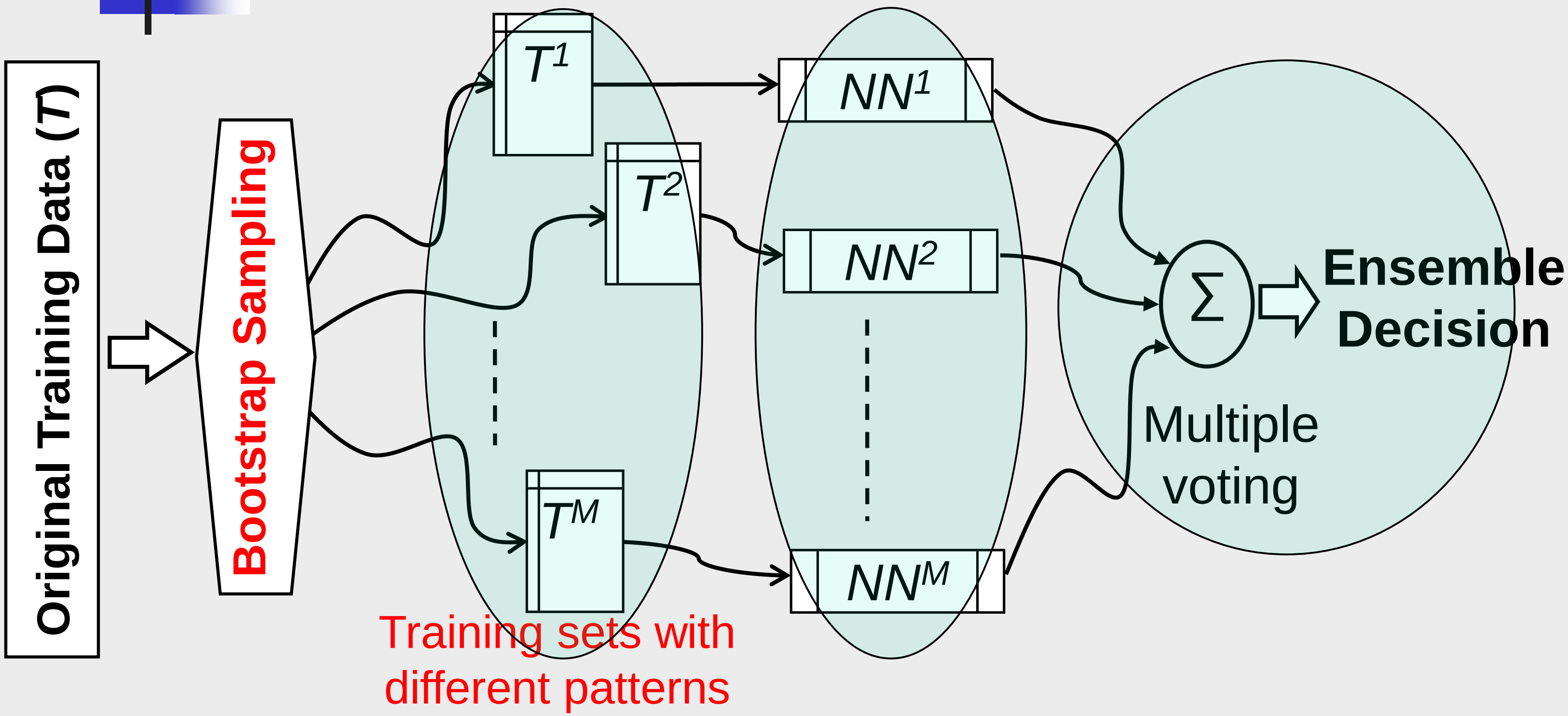
# Data Sampling for NNE Construction

Proper diversity among component NNs is an important parameter for ensemble construction so that failure of one may compensate by others.

- There are various ways one can produce diverse NNs, such as varying initial random weights, algorithm employed and training data.
- Since functionality of a NN depends on its training data, data sampling is considered as the most effective for diversity than other approaches.
- Data sampling (i.e. variation in training data) involves: bootstrapping of original training data, generation artificial data, sampling of input features, etc.

# 1. Bagging (Breiman, 1996)

L. Breiman, “Bagging Predictors” *Machine Learning*, vol. 24, pp. 23 –140, 1996.



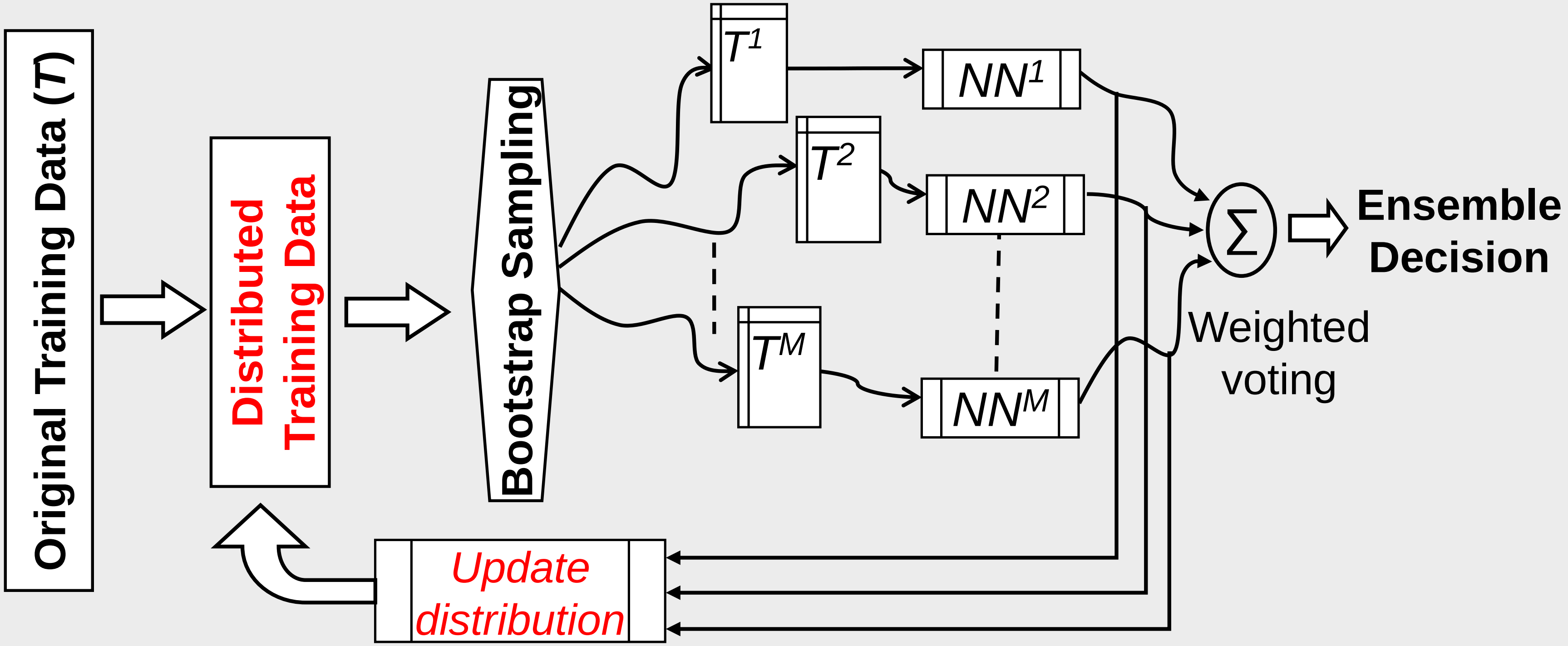

| $T$ | $T^1$ | $T^2$ |
|---|---|---|
| 1 | 4 | 10 |
| 2 | 9 | 6 |
| 3 | 6 | 3 |
| 4 | 8 | 9 |
| 5 | 1 | 2 |
| 6 | 8 | 7 |
| 7 | 5 | 3 |
| 8 | 1 | 4 |
| 9 | 2 | 2 |
| 10 | 4 | 9 |

Bootstrap sampling: Randomly selects a pattern and replace it again in its original place for later use.

In a training set many patterns appear multiple times while others are left.

# 2. AdaBoost (Freund & Schapire, 1996)

Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm", *in Proc. of the 13th Int. Conf. on Machine Learning(Morgan Kaufmann, 1996)*, pp. 148–156




- NNs are trained one after another sequentially and after training a NN distribution of training data updates.
- Existence of previously miss classified patterns increases in coming training sets due to error base distribution.

# Bagging and AdaBoost

1. Let $M$ be the number of networks to be trained for an ensemble

Take original training set $T = \{(x(1), d(1)), \ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,\ldots\ldots,k\}$

2. *for i=1 to M* {

   a. Make a training set, $T_i$ by sampling $N$ patterns uniformly at random with replacement from $T$

   b. Train network $NN_i$ by $T_i$

}

3. Ensemble decision is made in multiple voting way

**Bagging**

1. Let $M$ be the number of networks to be trained for an ensemble.

Take original training set $T = \{(x(1), d(1)), \ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,\ldots\ldots,k\}$

Assign weight for each pattern of $T$, initially weights are the same, i.e., $w_1(n)=1/N$

2. *for i=1 to M* {

   a. Make a training set, $T_i$ by sampling $N$ patterns at random with replacement from $T$ based on weight distribution $w_i(n)$

   b. Train network $NN_i$ with $T_i$

   c. $\varepsilon_i = \sum_{(x(n),d(n)) \in T : NN_i(x(n)) \neq d(n)} w_i(n)$ (weighted error on training set)

   d. $\beta_i = (1-\varepsilon_i)/\varepsilon_i$

   e. *for each* $(x(n), d(n)) \in T$ ,

   $if\ NN^i(x_n) \neq d_n\ then\ w_{i+1}(n) = w_i(n).\beta_i$ , $otherwise\ w_{i+1}(n) = w_i(n)$

   f. Normalize the weights of $T$

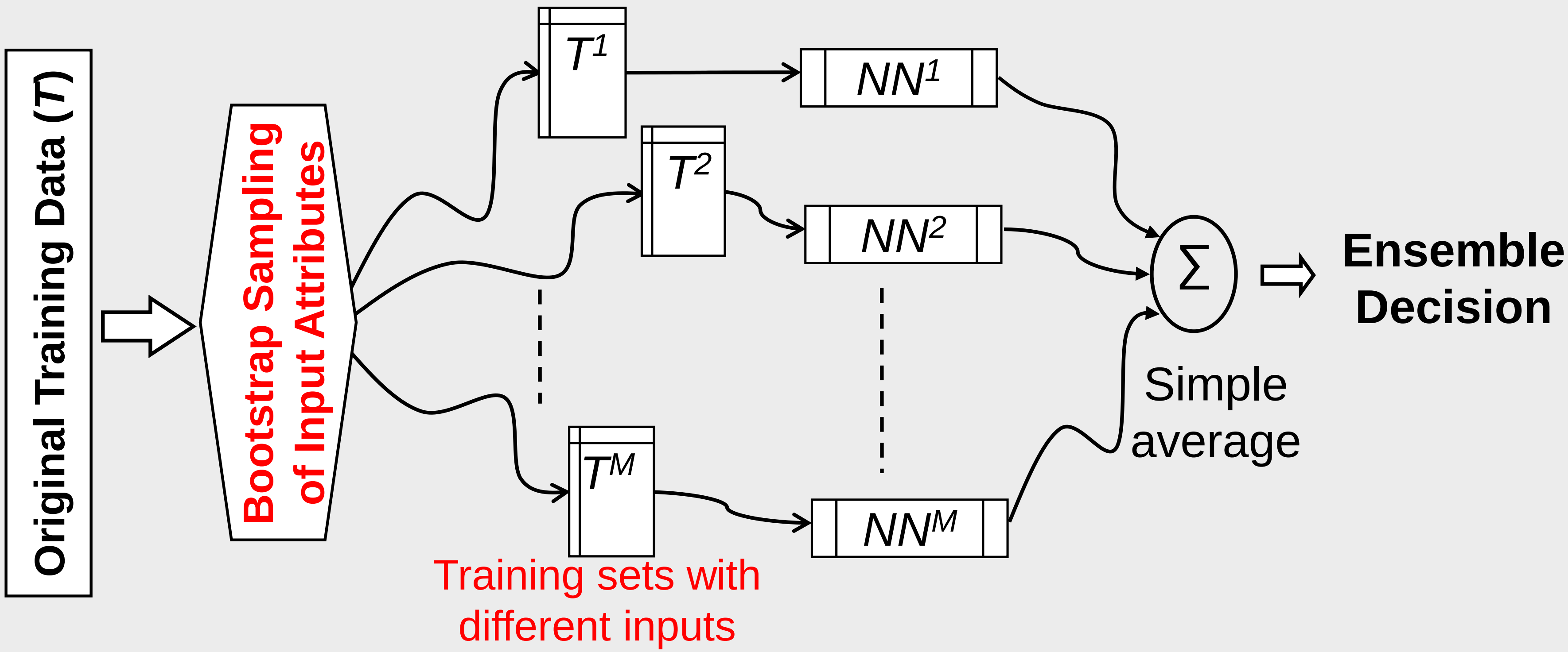
}

3. Ensemble decision is made in weighted voting way

**AdaBoost**

# 3. Random Subspace Method(RSM) (Ho, 1998)

T. K. Ho, "The random subspace method for constructing decision forests" *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.* 20, pp. 832–844, 1998




RSM require to maintain tagging with original inputs for every NN's inputs.

# RSM

1. Let $M$ be the number of networks to be trained for an ensemble

Take original training set $T = \{(x(1), d(1)), \ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,\ldots\ldots,k\}$

and feature set $F = \{1,2,\ldots\ldots, f\}$

2. *for i=1 to M* {

   a. Make a feature subset, $F_i$ by sampling $|F|$ features uniformly at random with replacement from $F$

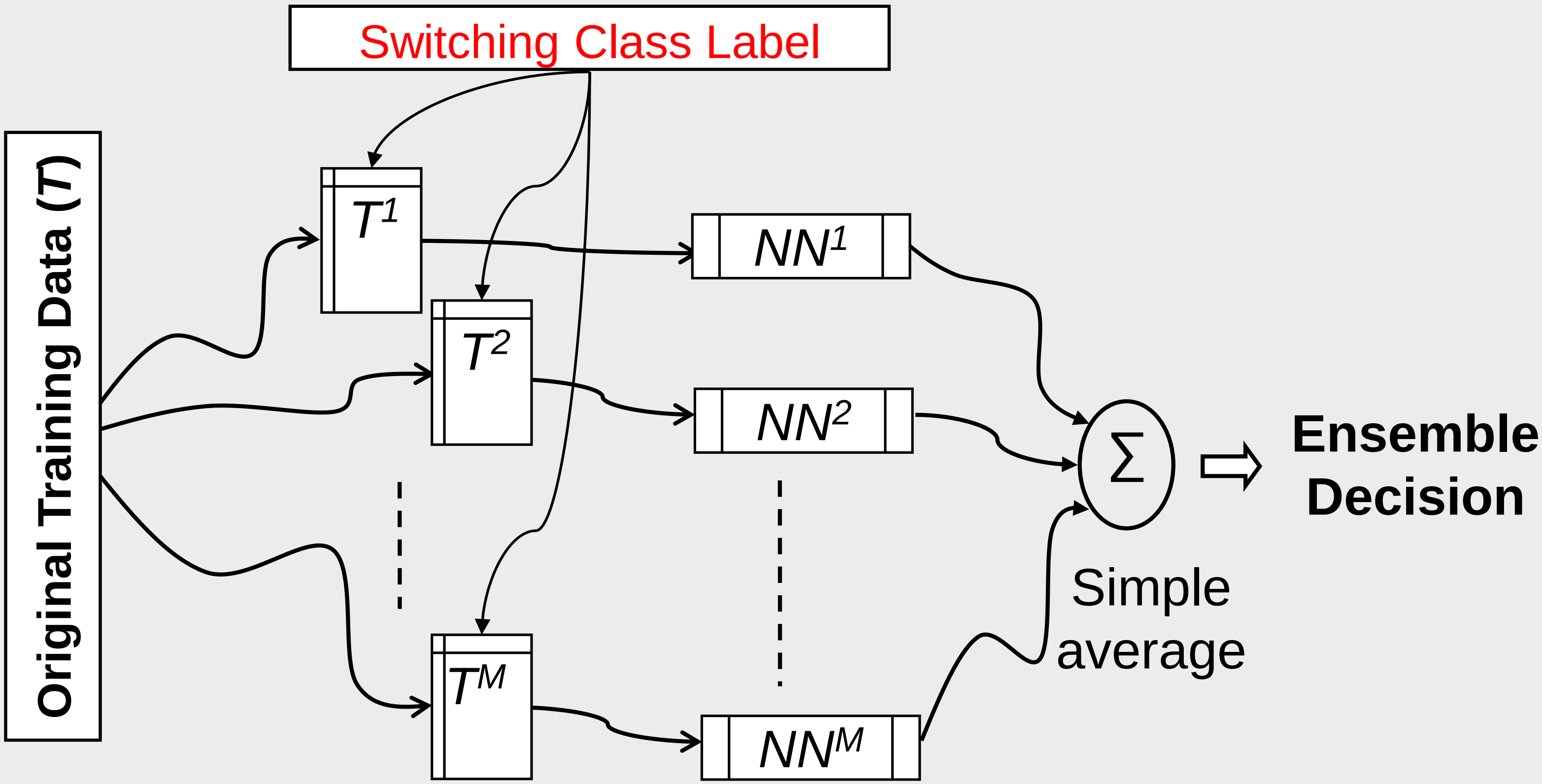
   b. Train network $NN_i$ by $T$ with feature set $F_i$

}

3. Ensemble decision is made in simple average way

# 4. Class Label Switching (Martínez-Muñoz & Suárez, 2005)

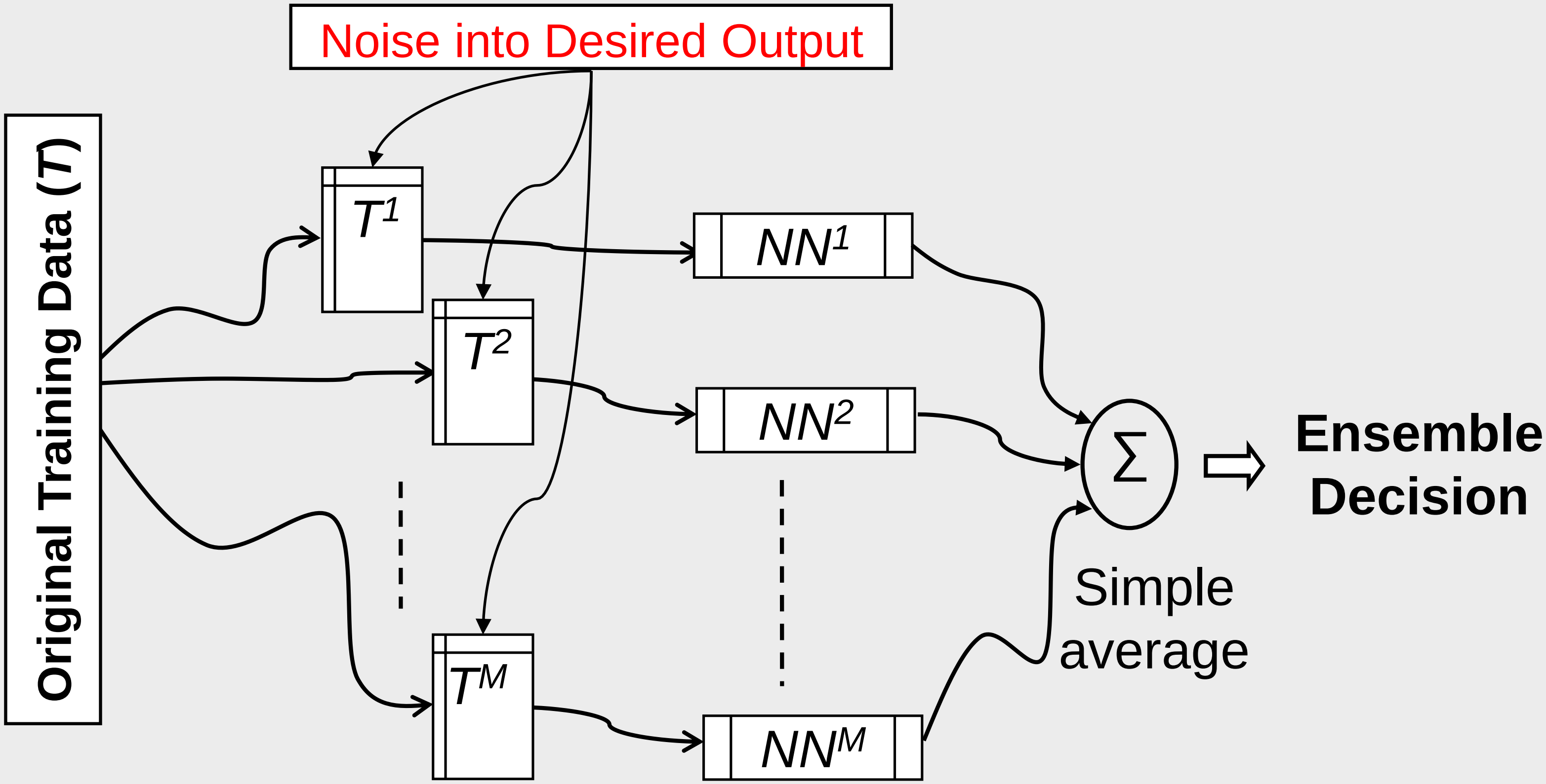
G. Mart´ınez-Mu˜noz and A. Su´arez, "Switching class labels to generate classification ensembles," *Pattern Recognition*, vol. 38, pp. 1483–1494, 2005.




A parameter $S_{fraction}$ maintains number of examples to change class label.

# 5. Smearing (Breiman, 2000)

L. Breiman, “Randomizing outputs to increase prediction accuracy,” Machine Learning, vol 40, pp. 229–242, 2000.



# Class Label Switching and Smearing

1. Let $M$ be the number of networks to be trained for an ensemble

Take original training set $T= \{(x(1), d(1)),\ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,\ldots\ldots,k\}$

Take $S_{Fraction}$ -factor that determines number of pattern to alter class label

Number of patters to switch the class label, $S = S_{Fraction} * N$

2. *for i=1 to M* {

    a. Make a training set, $T_i = T$

    b. Randomly select $S$ examples in $T_i$ and assign different class label randomly

    c. Train network $NN_i$ by $T_i$

}

3. Ensemble decision is made in simple average way

**Class Label Switching**

1. Let $M$ be the number of networks to be trained for an ensemble

Take original training set $T= \{(x(1), d(1)),\ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,\ldots\ldots,k\}$

Compute standard deviation measure ($sd_k$) is each class based on Eq. (2.1)

2. *for i=1 to M* {

    a. Make a training set, $T_i = T$

    b. Change the desired output of $T_i$ based on Eq. (2.2)

    c. Train network $NN_i$ by $T_i$

}

3. Ensemble decision is made in simple average way

**Smearing**

If $p_k$ is the proportion of the patterns in class $k$, then standard deviation measure ($sd$)

$$sd_k = 2\,(p_k\,(1-p_k))^{\wedge}0.5 \quad (2.1))$$

The new desired output for the k-th class for the n-th training pattern for a network is

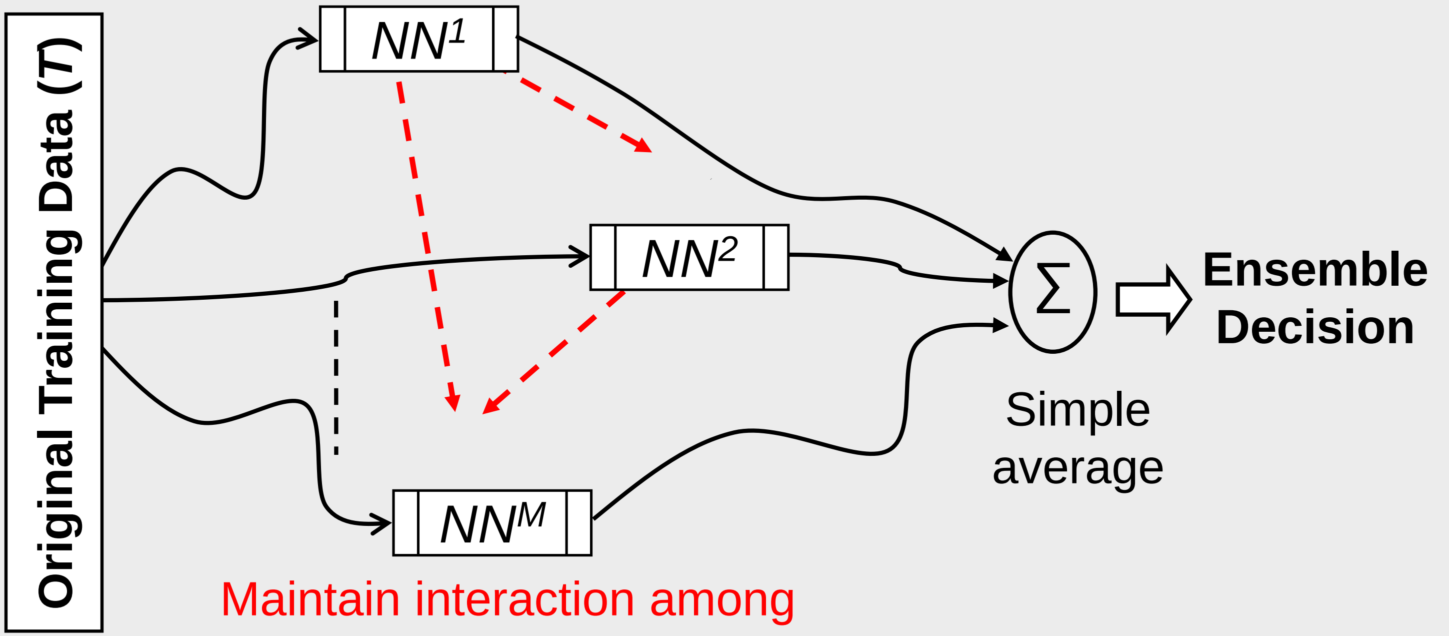
$$d_k'(n) = d_k(n) + z_k(n)sd_k \quad k=1,\ldots,K\ n=1,\ldots,N\ (2.2)$$

Where $z_k(n)$ is the random number from Gaussian distribution with a mean zero and a variance of one.

# 6. Negative Correlation Learning(NCL) (Liu & Yao, 1999)

Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12 pp. 1399–1404, 1999.

Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *IEEE Trans. Systems, Man, and Cybernetics — Part B*, vol. 29, pp. 716–725, 1999.




Maintain interaction among component NNs

$$\text{Error function, } e_i(n) = \underbrace{\frac{1}{2}\left(f_i(n) - d(n)\right)^2}_{\text{Normal BP portion}} + \underbrace{\lambda\left(\left(f_i(n) - \overline{f}(n)\right)\sum_{j \neq i}\left(f_j(n) - \overline{f}(n)\right)\right)}_{\text{Correlation penalty term}}$$

- Interaction produces negatively correlated NNs and therefore NNs motivate different functional spaces.
- The coefficient of the penalty term ($\lambda$) maintain strength of interaction.

# Negative Correlation Learning(NCL)

1. Let $M$ be the number of networks to be trained for an ensemble

   Take original training set $T = \{(x(1), d(1)), \ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1, 2, \ldots, k\}$

   Create $M$ networks, $NN_1$ ----$NN_M$

2. *for* $n=1$ *to* $N$ {

   Prepare ensemble output for pattern $n$

   *for* $i=1$ *to* $M$ {

   Train network $NN_i$ for pattern $n$ using error definition of Eq. (2.3)

   }

}

3. Ensemble decision is made in simple average way

**Negative Correlation Learning (NCL)**

$$e_i(n) = \frac{1}{2}(d(n) - f_i(n))^2 + \lambda p_i(n) \quad \text{------- Eq. 2.3}$$

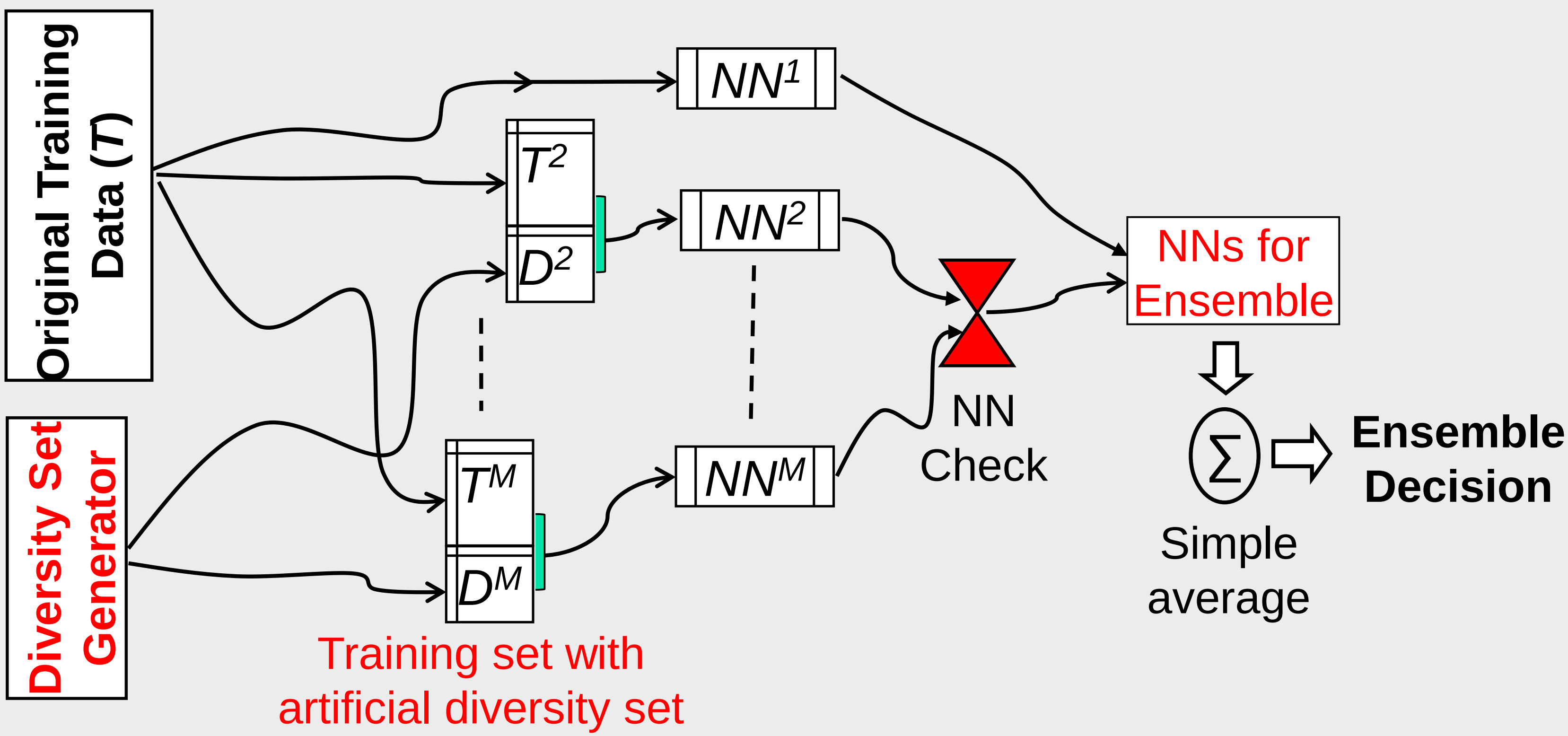
$$p_i(n) = (f_i(n) - \bar{f}(n)) \sum_{j \neq i} (f_j(n) - \bar{f}(n))$$

$$\bar{f}(n) = \frac{1}{M} \sum_{i=1}^{M} f_i(n)$$

# 7. DECORATE (Melville & Mooney, 2005)

Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples

P. Melville and R. J. Mooney, "Creating diversity in ensembles using artificial data," *Information Fusion, vol.* 6, pp. 99–111, 2005.




- Trains predefined large number of NNs to select NNs for final NNE
- A parameter $R_{size}$ maintains size of diversity set.
- Training of large number of NNs and additional diversity set increase training time.

# DECORATE

1. Let $I_{max}$ be the number of networks to be trained for an ensemble and $M$ is the desired number of networks

Take original training set $T = \{(x(1), d(1)), \ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,\ldots\ldots,k\}$

Take $R_{Size}$ -factor that determines size of diversity set

$i$=1 (for network in ensemble) and *trails* =1 (for trial network)

Train network $NN_i$ with $T$ (first network is trained with original training data)

Initialize ensemble, $Ens = \{NN_i\}$

Compute ensemble error, $\varepsilon = \left( \sum_{(x(n),d(n)) \in T : Ens(x(n)) \neq d(n)} 1 \right) / N$

2. *while trials* < $I_{max}$ and $i < M$ {

   a. Generate $R_{Size} \times |T|$ training examples, $R$

   b. Label examples in $R$ with probability of class labels inversely proportional to predictions of *Ens*

   c. Prepare training set $T_i$, $T_i = T \cup R$ and train network $NN_i$ with $T_i$

   d. $Ens = Ens \cup \{NN_i\}$

   e. Compute $\varepsilon'$ based on Step 1

   f. *if* $\varepsilon' \leq \varepsilon$ *then* $i = i + 1$ *and* $\varepsilon = \varepsilon'$, *otherwise* $Ens = Ens - \{NN_i\}$

   g. $trials = trials + 1$

}

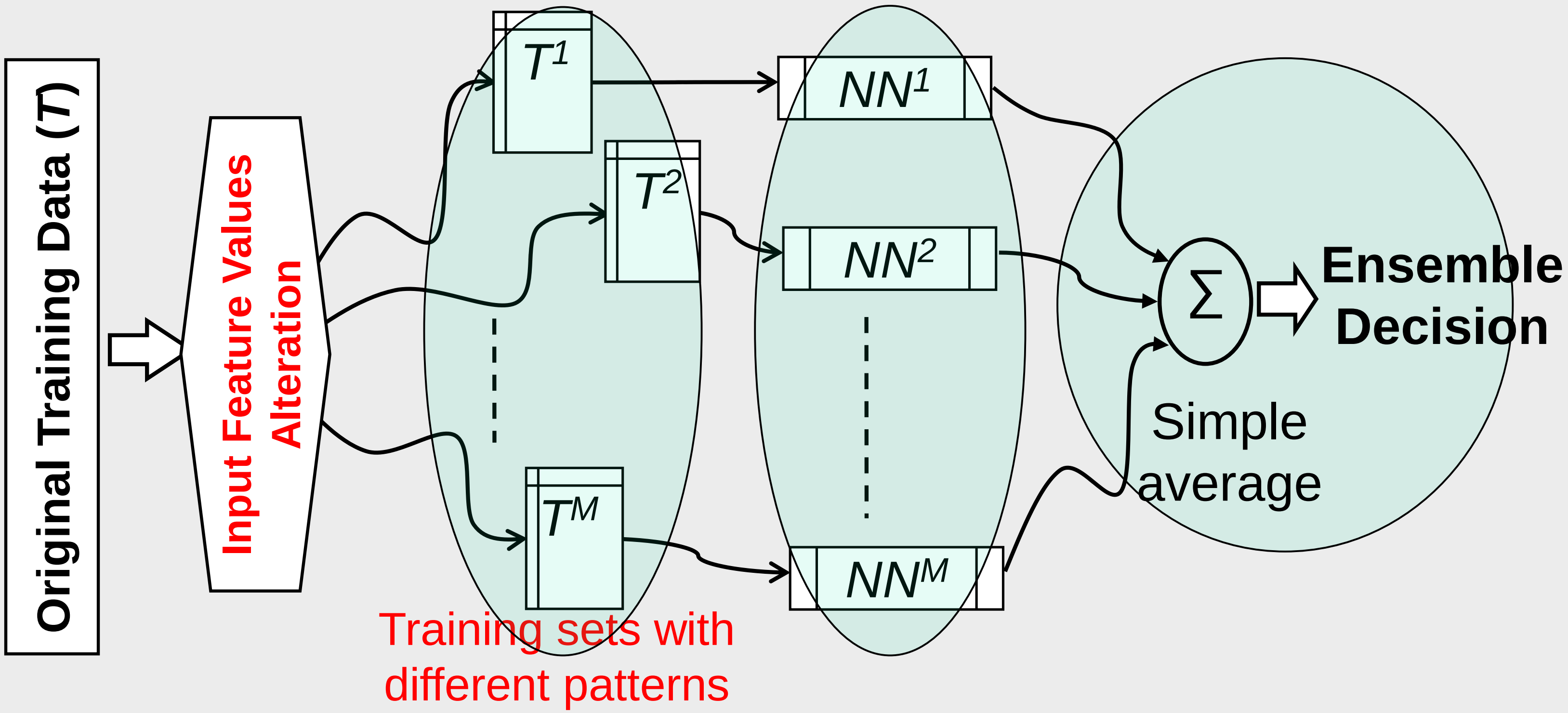
3. Ensemble decision is made in simple average way

**DECORATE**

# 8. Ensemble through Input Values Alteration (EIVA)

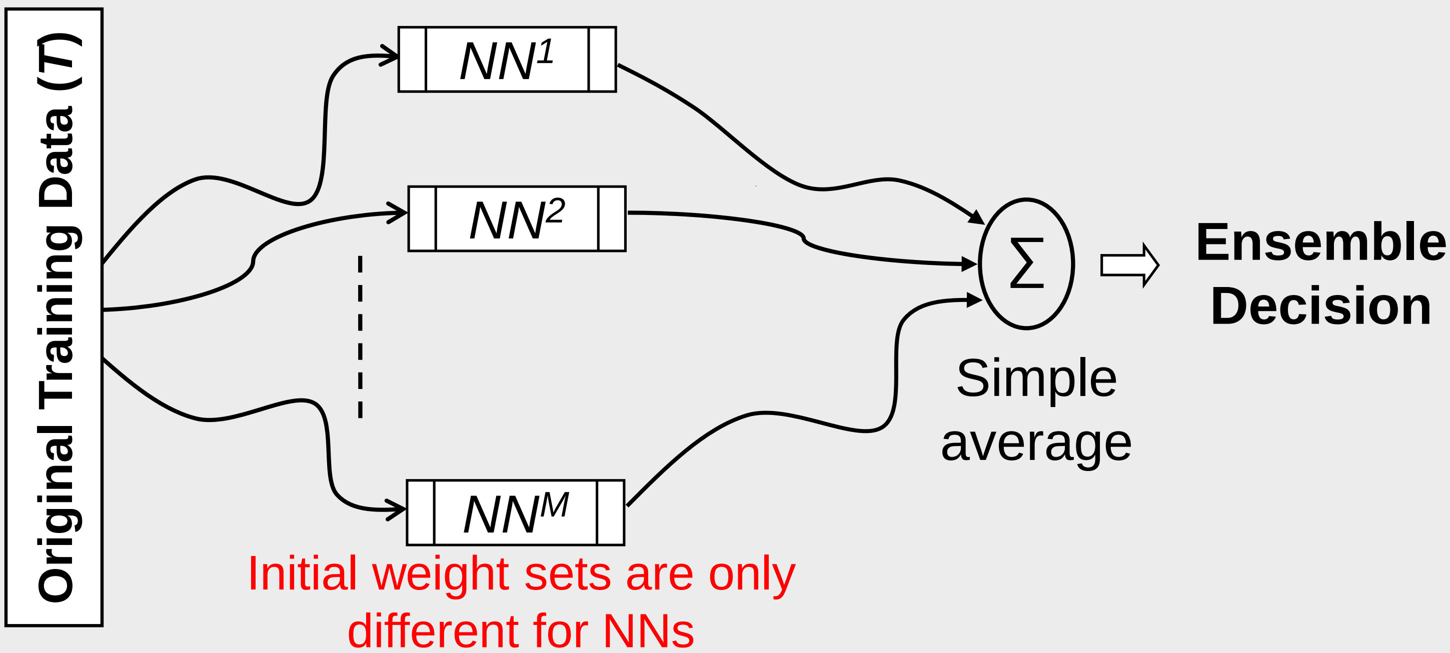
**(M.A.H. Akhand & K. Murase, 2012)**

M. A. H. Akhand, and K. Murase , "Ensembles of Neural Networks based on the Alteration of Input Feature Values " *International Journal of Neural Systems*, vol. 22, no.1, pp. 77-87, 2012.




Feature Values Alteration: Randomly selects a pattern and alter some of its feature values with the value of another one.

# 9. Simple NNE(sNNE)




- To evaluate performance of data sampling based NNE methods, sNNE is considered as base line.
- Less diversity due to same data for all NNs; therefore, performance is worse.

# Motivation to Comparative Study

- A number of ensemble methods already proposed using various data sampling techniques.
- Proposed methods are investigated on heterogeneous test bed.
- Some methods are proposed for decision trees and empirical result for NN is not available.

**Comparative study of the proposed methods on a common ground is necessary to evaluate their effectiveness.**

(Eight prominent NNE methods are considered for investigation.)

# Experimental Studies

**Common Settings:**

- For an NNE 20 NNs are considered.
- Each NN is trained for equal 50 / 75 / 100 iteration.
- Learning rate of back propagation was set 0.15.
- 10-fold cross validations was followed for result presentation.

**Built in Parameter Settings:**

- For DECORATE $R_{Size}$ value was 0.5, 0.75 or 1. Maximum trial NNs was 30.
- In class label switching $S_{fraction}$ was 0.1, 0.2 or 0.3.
- NCL was tested with $\lambda$ value 0.25, 0.5 or 0.75.

# TER Comparison over 50 Indp. Runs

| Problem | sNNE 20NNs/NNE | Bagging 20NNs/NNE | AdaBoost 20NNs/NNE | DECORATE (NNs/NNE) | RSM 20NNs/NNE | Switching 20NNs/NNE | Smearing 20NNs/NNE | NCL 20NNs/NNE |
|---|---|---|---|---|---|---|---|---|
| ACC | 0.1522 | 0.1417 | 0.1568 | **0.14**(8.34) | 0.1461 | 0.1423 | 0.1414 | 0.1443 |
| BCW | 0.0348 | 0.0322 | 0.0322 | 0.0299(6.40) | 0.0296 | **0.029** | 0.0319 | 0.0313 |
| CAR | 0.1128 | 0.0995 | **0.0799** | 0.1203(2.00) | 0.1647 | 0.118 | 0.1193 | 0.1036 |
| DBT | 0.2379 | 0.2321 | **0.2305** | 0.2342(1.14) | 0.2318 | 0.2382 | 0.2366 | 0.2308 |
| GCC | 0.2462 | 0.2424 | 0.2476 | 0.2652(1.88) | 0.2414 | 0.2416 | 0.2482 | **0.2402** |
| HDC | 0.1633 | 0.1573 | 0.1653 | **0.152**(6.78) | **0.152** | 0.1567 | 0.1567 | 0.1627 |
| HPT | 0.1547 | 0.1627 | 0.172 | 0.16(1.16) | 0.1573 | 0.1587 | 0.1547 | **0.152** |
| HTR | 0.0531 | 0.0518 | **0.0263** | 0.0528(1.02) | 0.0559 | 0.056 | 0.0558 | 0.0522 |
| INS | 0.1343 | 0.1297 | 0.1034 | **0.0606**(12.9) | 0.1429 | 0.132 | 0.1806 | 0.1366 |
| IRP | **0.0267** | 0.0293 | 0.028 | **0.0267**(1.00) | 0.0293 | **0.0267** | **0.0267** | **0.0267** |
| LMP | 0.1486 | 0.1529 | 0.1729 | **0.1371**(4.74) | 0.1486 | 0.14 | 0.1586 | 0.15 |
| PRM | 0.068 | 0.068 | 0.072 | **0.066**(20.00) | 0.068 | 0.07 | 0.078 | 0.068 |
| SGM | 0.07 | 0.0648 | **0.0432** | 0.0761(3.14) | 0.0681 | 0.0724 | 0.0765 | 0.0677 |
| SNR | 0.195 | 0.194 | 0.181 | **0.166(**7.58) | 0.20 | 0.201 | 0.209 | 0.195 |
| SPL | 0.1419 | 0.1556 | 0.1529 | 0.1823(2.36) | **0.0873** | 0.1527 | 0.1725 | 0.1409 |
| STL | 0.144 | 0.1379 | **0.1358** | 0.1525(2.98) | 0.1435 | 0.1419 | 0.1556 | 0.145 |
| WVF | 0.1327 | **0.1297** | 0.132 | 0.1308(4.18) | 0.1352 | 0.1339 | 0.1345 | 0.1312 |

# Result Summary over 30 Problems

| | sNNE | Bagging | AdaBoost | DECORATE | RSM | Switching | Smearing | NCL |
|---|---|---|---|---|---|---|---|---|
| **Average TER** | **0.1505** | **0.1393** | **0.1380** | **0.1441** | **0.1562** | **0.1512** | **0.1581** | **0.1462** |
| **Best/Worst** | **2/3** | **5/1** | **11/6** | **7/2** | **2/7** | **2/3** | **1/9** | **6/0** |
| **NNE Method** | **Pair Wise Win/Draw/Loss Summary** | | | | | | | |
| **sNNE** | - | 23/1/6 | 19/0/11 | 17/1/12 | 12/2/16 | 11/1/18 | 8/2/20 | 20/4/6 |
| **Bagging** | | - | 16/1/13 | 11/0/19 | 10/2/18 | 7/1/22 | 5/0/25 | 10/1/19 |
| **AdaBoost** | | | - | 13/0/17 | 11/0/19 | 12/0/18 | 10/0/20 | 15/0/15 |
| **DECORATE** | | | | - | 11/1/18 | 10/1/19 | 5/1/24 | 16/1/13 |
| **RSM** | | | | | - | 17/0/13 | 12/0/18 | 22/1/7 |
| **Switching** | | | | | | - | 6/2/22 | 19/1/10 |
| **Smearing** | | | | | | | - | 24/1/5 |

**Conclusions from the Comparative Study:**

- No one is superior to others for all the problems.
- DECORATE performs better for problems with limited examples, e.g., INS, LPM, PRM. NCL also good for small problems.
- For large sized problems bagging and AdaBoost are the best, e.g., CAR, HRT, WVF. AdaBoost might show very good result for very large problems.
- RSM performs well for sufficient input set, e.g., SPL.

# Conclusions

- Basic introduction about NNs and NNE is presented.
- Comparative study of prominent existing methods is given and identified effectiveness of the methods.
- Given an Outline for better NNE construction.