TMVA Project in Machine Learning

Abhinav Moudgil

Mentors
Sergei Gleyzer
Omar Zapata
Lorenzo Moneta

What is TMVA?

- Toolkit for Multi-Variate Analysis (TMVA)
- Machine Learning Environment
- Integrated into ROOT, a Data Analysis Framework
- Used for processing and parallel evaluation of sophisticated multivariate classification techniques

Objective

Project aims to provide support for feature engineering in TMVA by implementing some advanced feature extraction methods

Variance Threshold

Deep Autoencoders

Feature Clustering

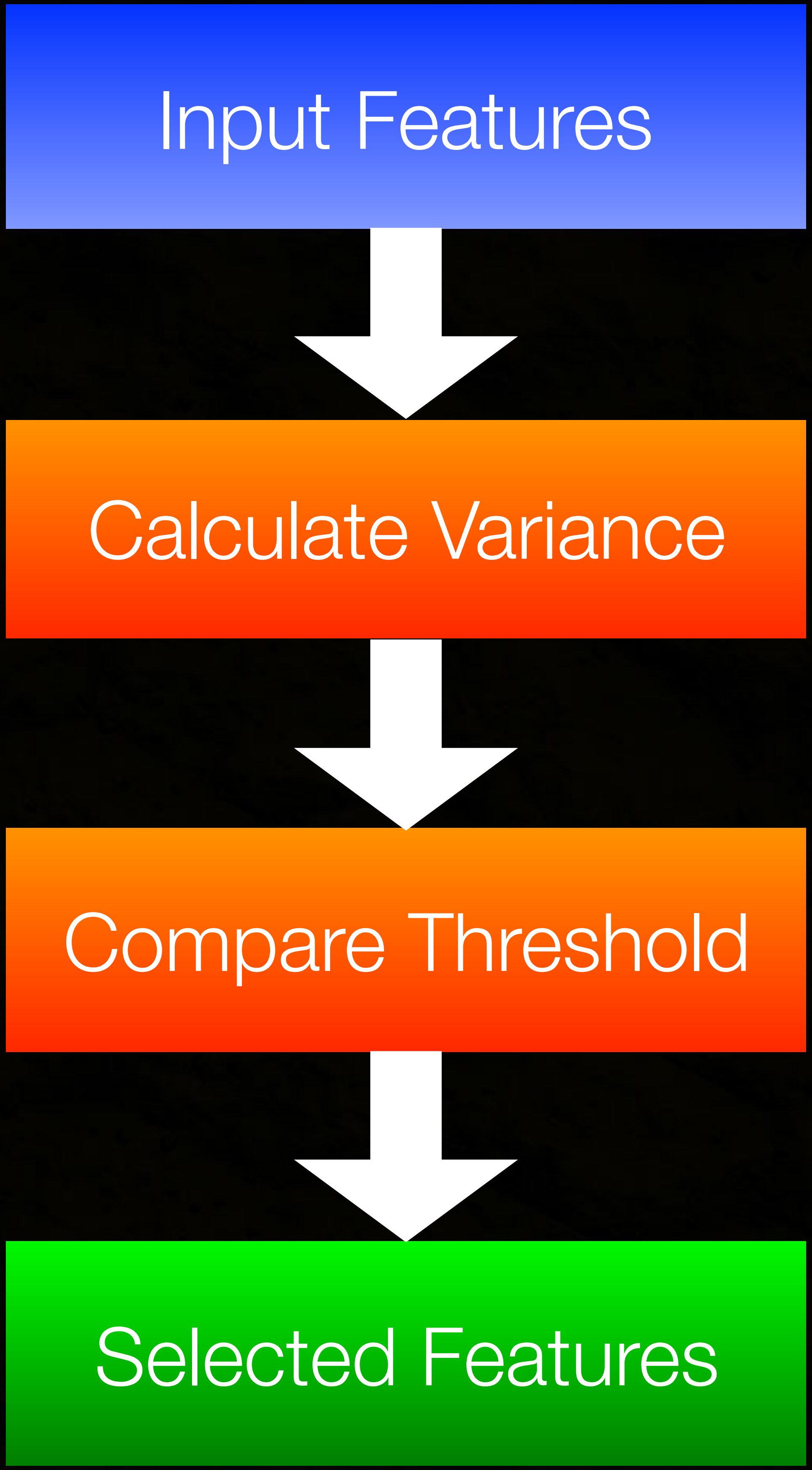
Random Projection

Overview

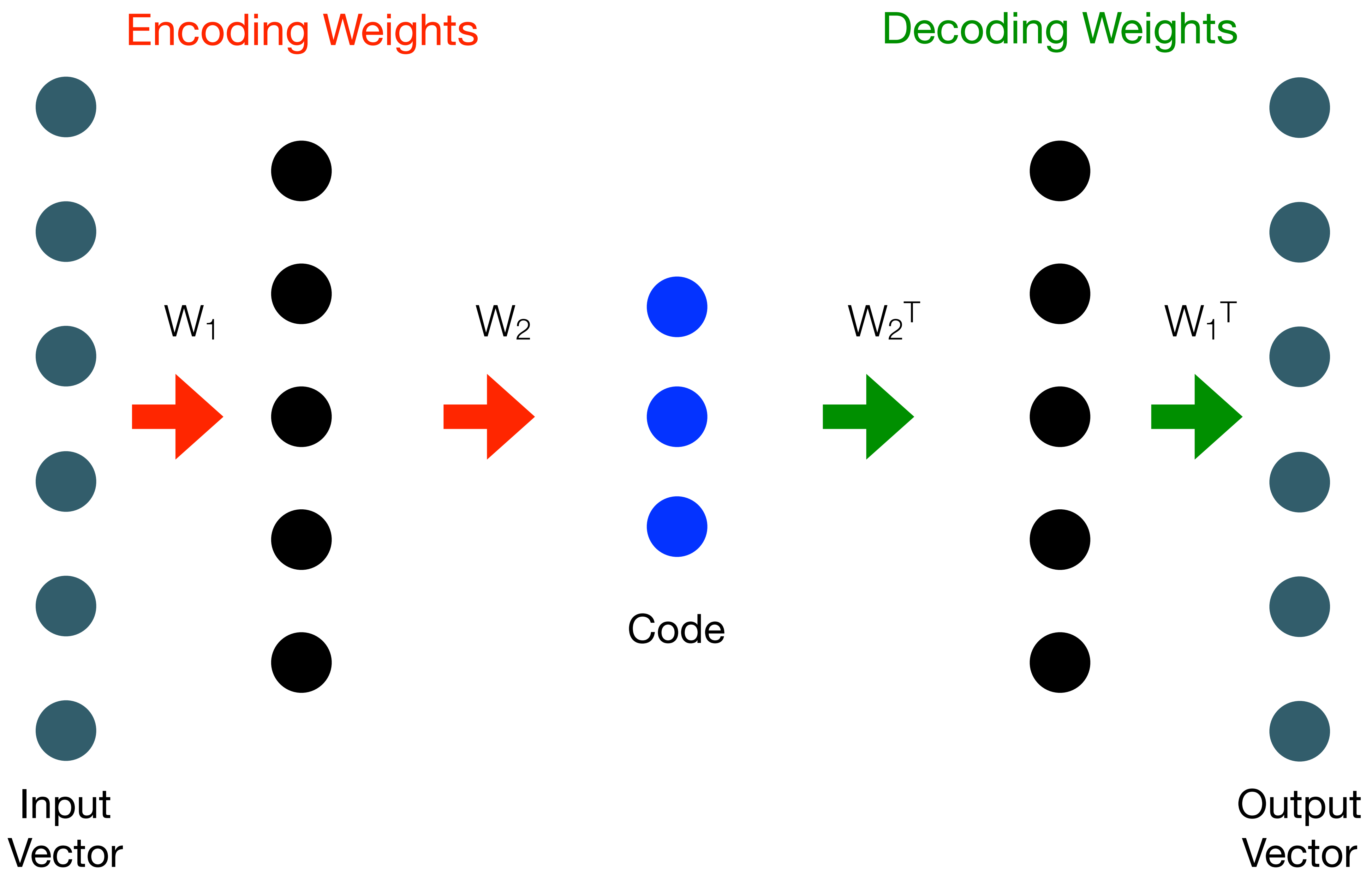
Brief introduction to each feature extraction method implemented

Variance Threshold

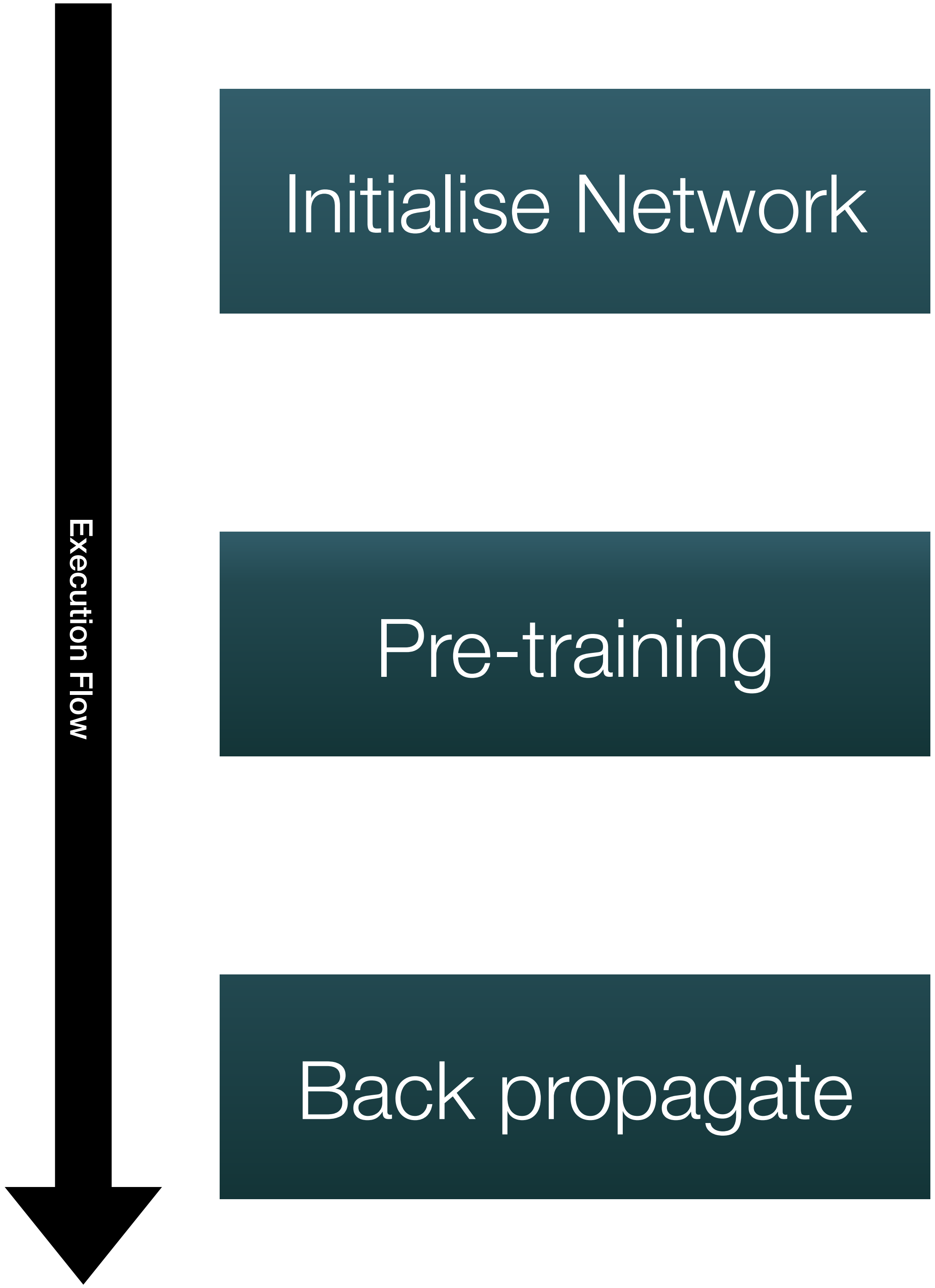
- Unsupervised feature selection method
- Takes a threshold value as input
- Select features whose variance lie above the threshold value



Deep Autoencoders Architecture



Deep Autoencoders Training



Execution Flow

Initialise Network

Build a symmetric network with input vector same as output vector

Pre-training

Layer by layer unsupervised pre-training which set the initial encoding weights. Decoding weights are set as transpose of encoding weights

Back propagate

Fine tune the network by applying backpropagation

Feature Clustering

- Partition features into different clusters
- Features in the same cluster contain similar structural information of the given instances
- Obtained feature subset consists of features from variant clusters, so similarity between selected features will be low

Reference:
Cheung, Yiu-ming, and Hong Jia. "Unsupervised Feature Selection with Feature Clustering." Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on. Vol. 1. IEEE, 2012.

Random Projection

- A simple and computationally efficient way to reduce the dimensionality of the data
- Suitable approximation technique for distance based method
- Embeds a set of points from high dimensional space to low dimensional space such that distances between the points are nearly preserved

Reference:
Bingham, Ella, and Heikki Mannila. "Random projection in dimensionality reduction: applications to image and text data." Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2001.

Software Implementation

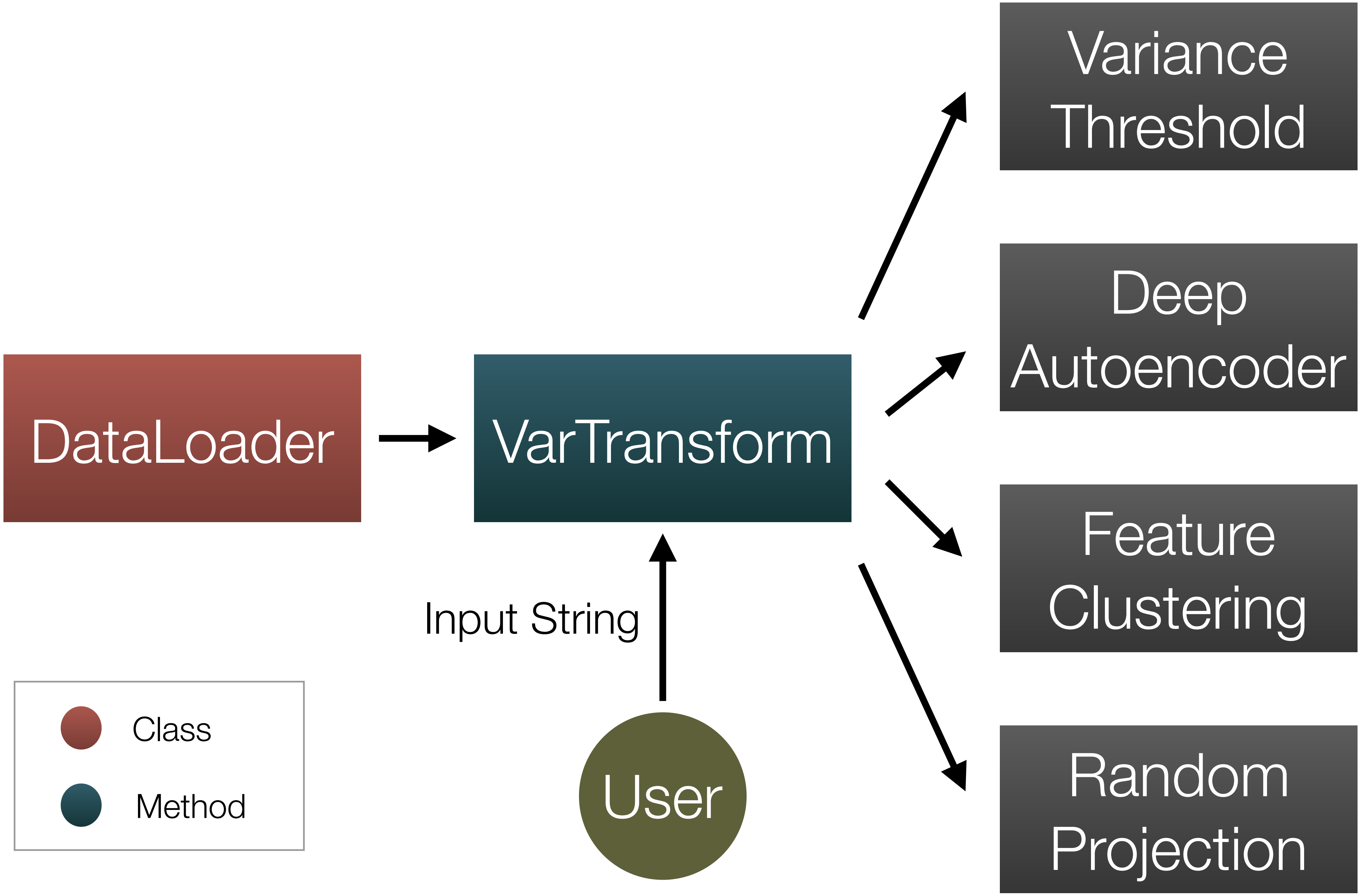
Implementation & Design in TMVA

```
TMVA::DataLoader* TMVA::DataLoader::VarTransform(TString trafoDefinitio
{
   TString trOptions = "0";
   TString trName = "None";
   if (trafoDefinition.Contains("(")) {

      // contains transformation parameters
      Ssiz_t parStart = trafoDefinition.Index( "(" );
      Ssiz_t parLen   = trafoDefinition.Index( ")", parStart )-parStart

      trName = trafoDefinition(0,parStart);
      trOptions = trafoDefinition(parStart,parLen);
      trOptions.Remove(parLen-1,1);
      trOptions.Remove(0,1);
   }
   else
      trName = trafoDefinition;

   // variance threshold variable
   if (trName == "VT") {

      // find threshold value
      Double_t threshold = 0.0;
      if (!trOptions.IsFloat()){
         Log() << kFATAL << " VT transformation must be passed a floa
         return this;
      }
      else
         threshold =  trOptions.Atof();
      Log() << kINFO << "Transformation: " << trName << Endl;
      Log() << kINFO << "Threshold value: " << threshold << Endl;

      // calculate
      if(!fCalcNorm)
         CalcNorm();

      const UInt_t nvars = DefaultDataSetInfo().GetNVariables();
      Log() << kINFO << "Number of variables before transformation:
      std::vector<VariableInfo>& vars = DefaultDataSetInfo().GetVa
```

TMVA Basics and Nomenclature

- Features are called variables in TMVA toolkit
- Samples in a dataset are called events
- ROOT Trees are like spreadsheets which contain variables of different datatypes. They are stored in files with extension “.root”
- DataLoader contains all the information about dataset. It is passed to Factory object which books, trains and tests all the classification and regression methods

Design



DataLoader

VarTransform

Input String

User

Variance Threshold

Deep Autoencoder

Feature Clustering

Random Projection

● Class

● Method

User Interface

- User gives option string to VarTransform method there in DataLoader class.
- Option string has different format corresponding to each variable transformation method
- Current DataLoader processes the option string, apply variable transformation and returns a new DataLoader with transformed variables

Sample code

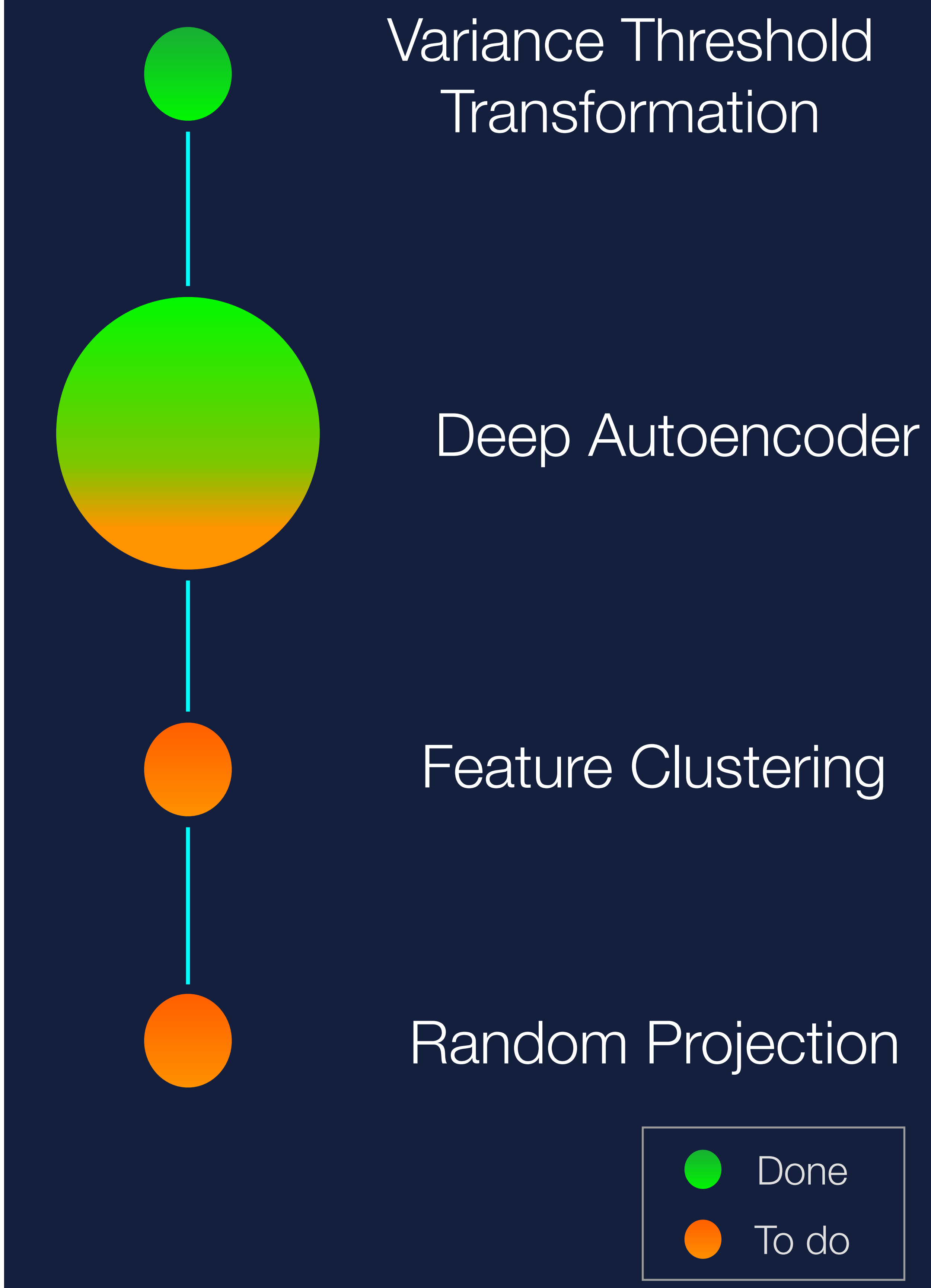
```
outputFile = TFile("TMVAOutput.root",
"RECREATE")
inputFile  = TFile("../datasets/
mydataset.root")
TMVA.Tools.Instance()
# initialise factory
factory =
TMVA.Factory("TMVAClassification",
                       outputFile,
                       "!V:ROC:!
Correlations:!Silent:Color:!
DrawProgressBar:AnalysisType=Classification
")
# initialise DataLoader
loader = TMVA.DataLoader(“mydataset”)
# apply variance threshold transform
newLoader = loader.VarTransform(“VT(2.95)")
# book TMVA ANN: MLP Multilayer Perceptrons
factory.BookMethod(myloader,
                   TMVA.Types.kMLP,
                   "MLP",
                   “!
V:NeuronType=tanh:VarTransform=N:NCycles=15
0:HiddenLayers=N+5:TestRate=5:!
UseRegulator")
factory.TrainAllMethods()
factory.TestAllMethods()
```

Milestones

- Implemented Variance Threshold feature selection method
- Designed and implemented autoencoders, which uses only backpropagation

What's Next?

- Implement pre-training in Deep Autoencoders
- Implement Feature Clustering and Random Projection



★ Size of circle represent the tentative workload.

References

- Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." Science 313.5786 (2006): 504-507.

- Bengio, Yoshua, et al. "Greedy layer-wise training of deep networks." Advances in neural information processing systems 19 (2007): 153.

- Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.

- Le Roux, Nicolas, and Yoshua Bengio. "Representational power of restricted Boltzmann machines and deep belief networks." Neural computation 20.6 (2008): 1631-1649.

Thank you.