Committee Machines

Techniques based on the Divide and Conquer Principal

Committee Machines

- A set of multi-layer perceptrons cooperating to solve a common problem
- Can be viewed as a set of cooperating experts
- Typically 2 or more levels of Perceptrons
- The outputs from each layer are combined by the next layer to form the output of that layer
- The input space is typically divided into a set of subspaces among the input level perceptrons

Motivations

- Divide the input problem into sub problems, solve them separately and combine the results to obtain a better result by one of the following methods:
  - Use several neural networks to solve the same problem and average the result to obtain a better result
  - Use several neural networks to solve the same problem and “vote” for the best result
  - Use several neural networks to solve the same problem and select the final result based on a majority vote
Types of Committee Machines

- Static Committee Machines:
  - 2 layers: method of combination at the second level does not depend on the input
- Dynamic Committee machines:
  - 2 or more layers: method of combination at the second level depends on the input
  - More than 2 layers can be constructed by applying the "divide and conquer" principal recursively

Static Committee Machines

- Ensemble Averaging Machines
- Boosting Machines

Ensemble Averaging Machines
Experts Layer

The networks at the expert layer are trained using the same training set (or different sets of similar distribution) but may be:
- Identical neural networks trained using different training parameters (e.g. initial conditions, learning rate)
- Neural networks based on different models

Combiner Layer

- The combiner layer consists of:
  - A single layer of linear perceptrons
  - For this layer: $\sum w_i = 1$
  - Typically: all weights are equal

Advantages

- By averaging, the average error is reduced
- Over training risk is minimized
- The effect of one (or more) of the experts training to a local minima is minimized
Boosting

- Similar to ensemble averaging except:
  - Experts are trained on different data sets with different distributions
  - Relies on the use of weak learning models

Weak Learning Models

- A variant on probably approximately correct (PAC) learning models
- After learning the network produces a correct result with a probability slightly greater than 0.5

Boosting Models

- Boosting by filtering:
  - The input data is filtered out of a virtually unlimited supply of training data to select the training set for each of the experts
- Boosting by Subsampling:
  - The training set for each expert is selected from a fixed training set according to some probability distribution
- Boosting by re-weighting:
  - The weighted version of the same fixed training set is used for each of the experts
Boosting by Filtering

- Suitable for pattern classification problems
- Relies on multiple networks at the expert level, each individually trained using a weak learning technique
- The output layer selects based on a majority vote (or a variant of majority vote)

Expert Layer Training

1. Train the first expert using N randomly selected entries
2. Train the second expert using N entries selected such that half of the entries produce a correct result from the first expert, and the other half produces an incorrect result
3. Train the third expert using N entries that produce different results at the first and second layers

Combiner Layer

- The combiner Layer may take any of several forms:
  - Simple majority vote
  - Weighted majority vote (e.g., the second expert having a higher weight than the first and the third having a higher weight than the second to account for them being trained to “harder” problems
  - Several weighted averages are formed out of the outputs of the individual experts followed by a selection using one of the above voting methods
Error Rate

- Dynamic Committee Machines
  - Combining layer depends on input vector
  - Simplest form: a single layer of linear neurons with weights dynamically computed as a function of the input vector

Mixture of Experts Model
Mixture of Experts Model

- Each expert specializes in a range of inputs
- The weights of the combining layer are computed from the gating network as a function of the input vector

Gating Network

- A single layer of non-linear neurons
- One neuron for each of the experts
- The output of each neuron is a weight for the combining network

\[ g_j = \exp(a_j^T X) \]
\[ \sum_{j=1}^{K} g_j = 1 \]
Gating Network

- At the limit, it is a realization of a winner takes all network
- How do we select the weights $a_k$?

Hierarchical Mixture of Experts Model

- Input space subdivided into nested subspaces
- Can be decomposed hierarchically into any number of layers
- Can be viewed as a soft decision tree
**Decision Tree Design**

**Heuristic**

- Manually design a hard decision tree for the target domain
- Use the hard decision tree as a framework for designing the initial hierarchy