Applications of Artificial Intelligence

Sebastian Iwanowski
FH Wedel

Chapter 4:
Knowledge-Based Systems

4.3: Case-Based Reasoning

2. Case-Based Diagnosis

Input to knowledge base:

- Cases with complete symptom vector and associated faults (classified unambiguously)

a) Classical AI, with similarity measure:

- Similarity measure for incomplete symptom vectors (often weighted between different types of symptoms)

Structure of knowledge base:

- Points in vector space
- Similarity measure

Job of inference engine:

- For a new vector given, find the most similar symptom vector of the knowledge base.
- Assign the same fault to the new vector as associated to the reference vector in the knowledge base (possibly with a probability value).

2. Case-Based Diagnosis

Input to knowledge base:

- Cases with complete symptom vector and associated faults (classified unambiguously)

b) with neural networks:

- Neural network with input layer (for symptom vector) and output layer (for faults) and (optionally) intermediate layer of nodes and edges, marked by variable weights.

Structure of knowledge base:

- Points in vector space
- Neural network with clearly defined weights (dependent on trained symptom vectors and associated faults)

Job of inference engine:

- Apply new symptom vector to the input layer of the network.
- Read the associated fault from the output layer.

Case-Based Reasoning (CBR)

Generalisation of case-based diagnosis to arbitrary case-based reasoning strategies:

Principle:

- Given cases as vectors (*complete symptom vectors*): These are “learnt” and build the knowledge base.
- Given new vectors, of which not all parameters are known (*incomplete symptom vectors*): These are to be classified.
- Assign values to the unknown parameters.

Job of inference engine (simple variant):

- For the new vector, find the closest symptom vector learnt by the knowledge base.
- For the unknown parameters of the new vector, assign the same values as in the associated symptom vector learnt by the knowledge base.

This method only makes sense when the unknown values come from a discrete (better finite) domain !

Case-Based Reasoning (CBR)

Improvement for continuous value domains:

Job of inference engine (better variant):

- For the unknown parameters of the new vector, assign values “in between” values of “nearby” symptom vectors learnt by the knowledge base.

Other mathematical formulation of this method:

- Consider the unknown parameters of the new vectors as function values of the known parameters: Find a continuous function where all vectors learnt by the knowledge base are contained.
- Of this function, assign the function values of the known parameters to the unknown parameters.

Query: **How do we get an appropriate function for a given set of reference vectors?**

Answer:

- Take a class of functions, each function differing by certain parameters.
- Determine the parameters solving an equation system obtained from the known reference vectors.

Case-Based Reasoning (CBR)

Determining parameters in function classes (regression):

Linear regression:

- Find the weights in a linear function of the form: $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} W_i x_i$

Generalisation:

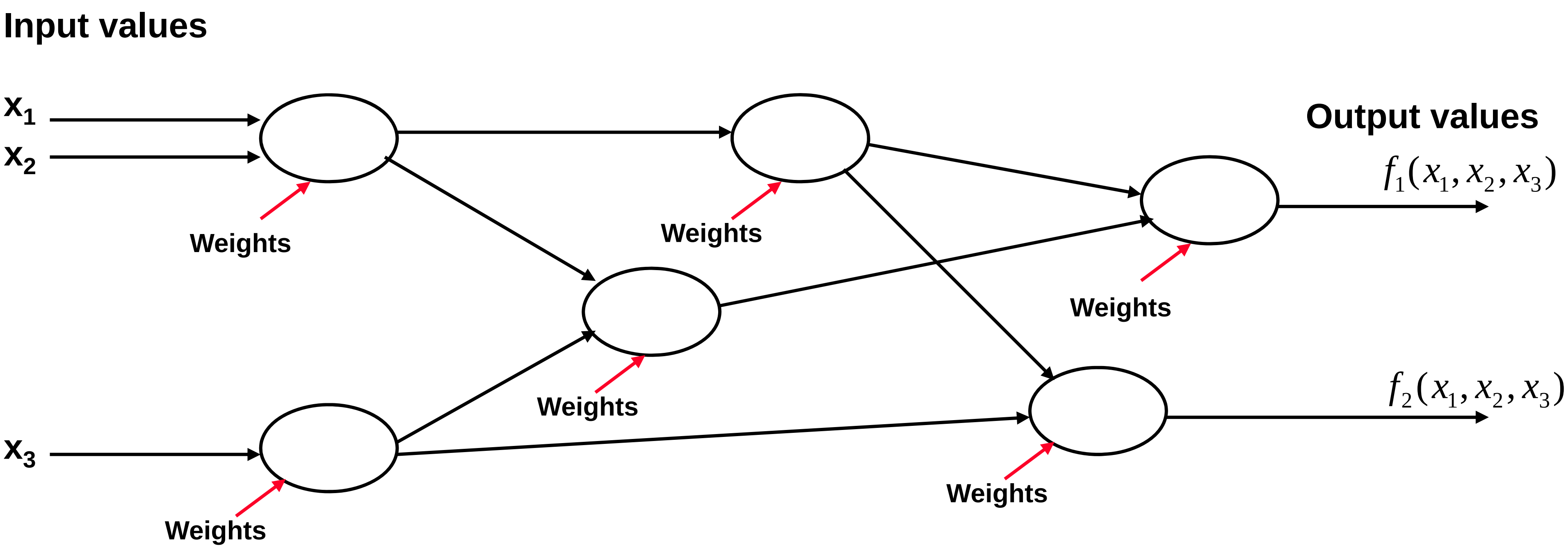
1. Find the weights in a linear equation **system**.
2. Find the weights in linear equation systems of higher order.
3. Find the weights in parametrised inequality systems.

- **Case-based reasoning is designed for systems which can*not* be modeled easily.**
- **This is why a higher order equation system does not make sense.**
- **It is better to work with many weakly connected equation systems and distribute the unknown knowledge.**

Case-Based Reasoning (CBR)

Idea of neural networks:

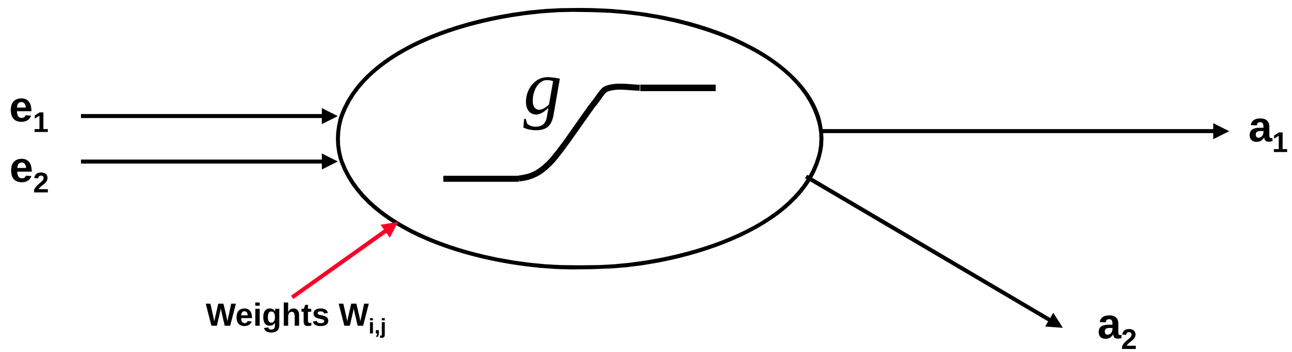
Given a multi-valued function f (notation: $f_i(x_1, x_2, \ldots, x_n)$)



- The weights may be preset but are adapted to the examples learnt.
- Function values of new inputs are obtained applying the neural network.

Case-Based Reasoning (CBR)

Functionality of a single neuron:



$$a_i(e_1, e_2, \ldots, e_n) = g\left(\sum_{j=1}^{n} W_{i,j} \cdot e_j\right)$$

- g is a generalised threshold function which is the same for all outputs of the same neuron.

Case-Based Reasoning (CBR)

Different layered neural networks:

Neral networks without intermediate layers:

- Neurons of the first layer accepting the inputs are connected to neurons of the second layer providing the outputs.

Neural networks with intermediate layers

- Input and output layers are connected by further “hidden” intermediate layers.

Neural networks with feedback:

- Generation of “memory”

Case-Based Reasoning (CBR)

What is the crucial difference between neural networks and „classical“ CBR systems?

➢ **Neural networks distribute the knowledge about the cases learnt.**

Theoretical advantages of distribution:

- Arbitrariness of function class chosen does not play such an important role.
- Intransparent cases are handled by an intransparent method:
  The distributed method is “self-adjusting”.

Practice shows:

- Good neural networks need fewer training cases than classical CBR systems.
- Neural networks provide better classification results.

Neural networks and AI

Are neural networks knowledge-based ?

Are neural networks expert systems ?

What may be called „Artificial Intelligence“ ?

Summary: Case-Based Reasoning

Advantages and Disadvantages:

- **The method is simple.**
  - The diagnosis of the run time component is very fast.
  - Knowledge acquisition can easily be automatised.
  - The knowledge base can only be generated for systems where experience is given.
  - The knowledge base consumes a lot of storage (similarity measure only).

Summary: Case-Based Reasoning

Advantages and Disadvantages:

- **The knowledge base does not contain any other structural knowledge than the similarity measure or the NN.**

  - All application domains are equally suited.
  - The same inference engine may be applied for totally different application domains.
  - Even with a small change of the system, the knowledge base cannot be used reliably.
  - Similarity measure and neural network are arbitrary.
  - Each run time diagnosis may be wrong.
  - The result is not comprehensible (at least for neural networks).