Recurrent perceptron-like networks

Chapter 7

recurrent networks

- are networks that are capable of influencing themselves by means of ***recurrences***,
- including the network output in the following computation steps.
- There are many types of recurrent networks of nearly arbitrary form, it will be called ***recurrent multilayer perceptrons***.

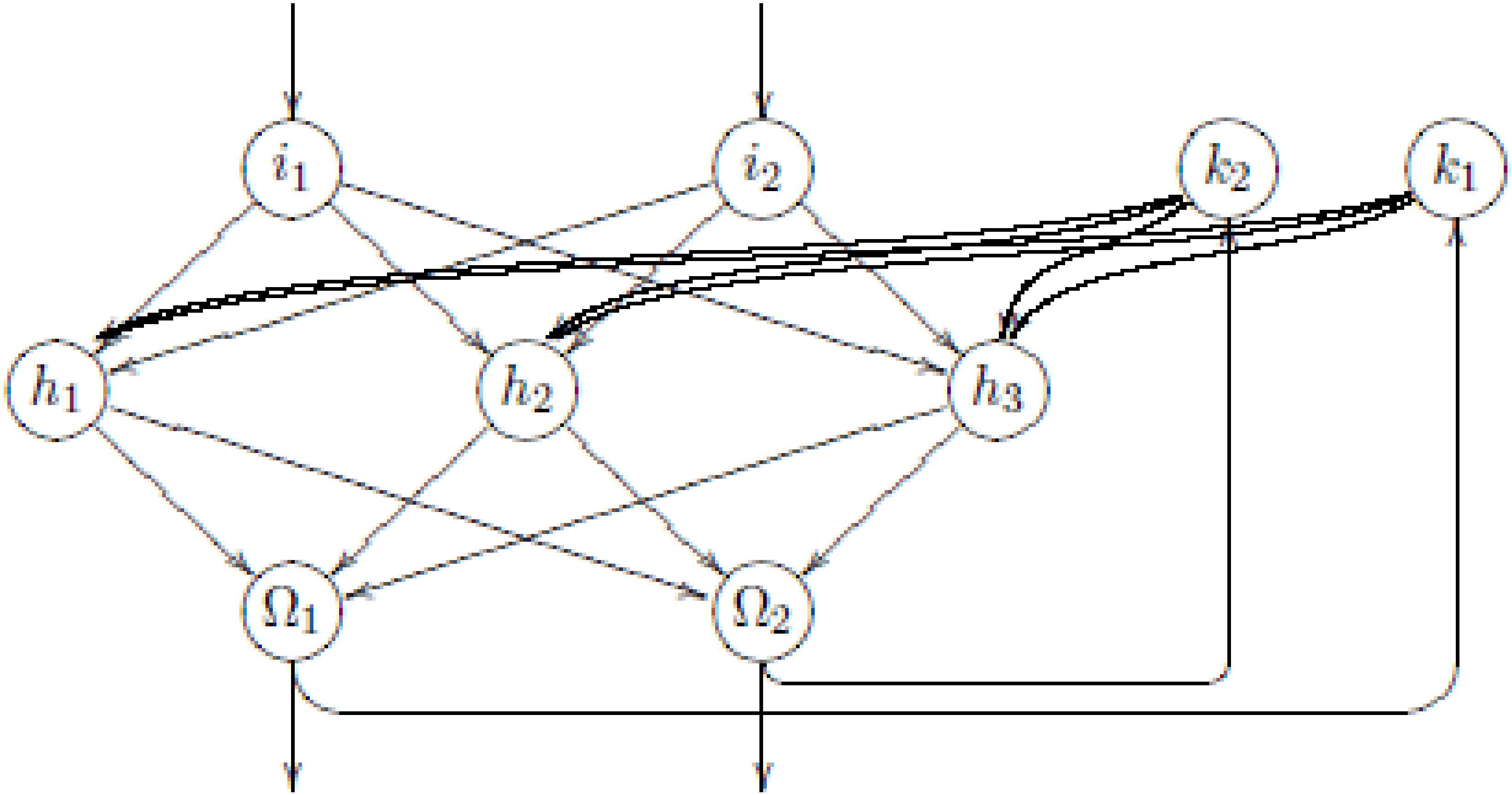
recurrent network

- an input x that is constant over time may lead to different results
- the network could converge, i.e. it could transform itself into a fixed state and at some time return a fixed output value y.
- it could never converge, or at least not until a long time later, so that it can no longer be recognized, and as a consequence, y constantly changes.

Jordan networks

- is a multilayer perceptron with a set K of so-called context neurons k_1, k_2, . . . , $k_{|K|}$. There is one context neuron per output neuron
- a context neuron just memorizes an output until it can be processed in the next time step.
- there are weighted connections between each output neuron and one context neuron.
- The stored values are returned to the actual network by means of complete links between the context neurons and the input layer.

Jordan networks



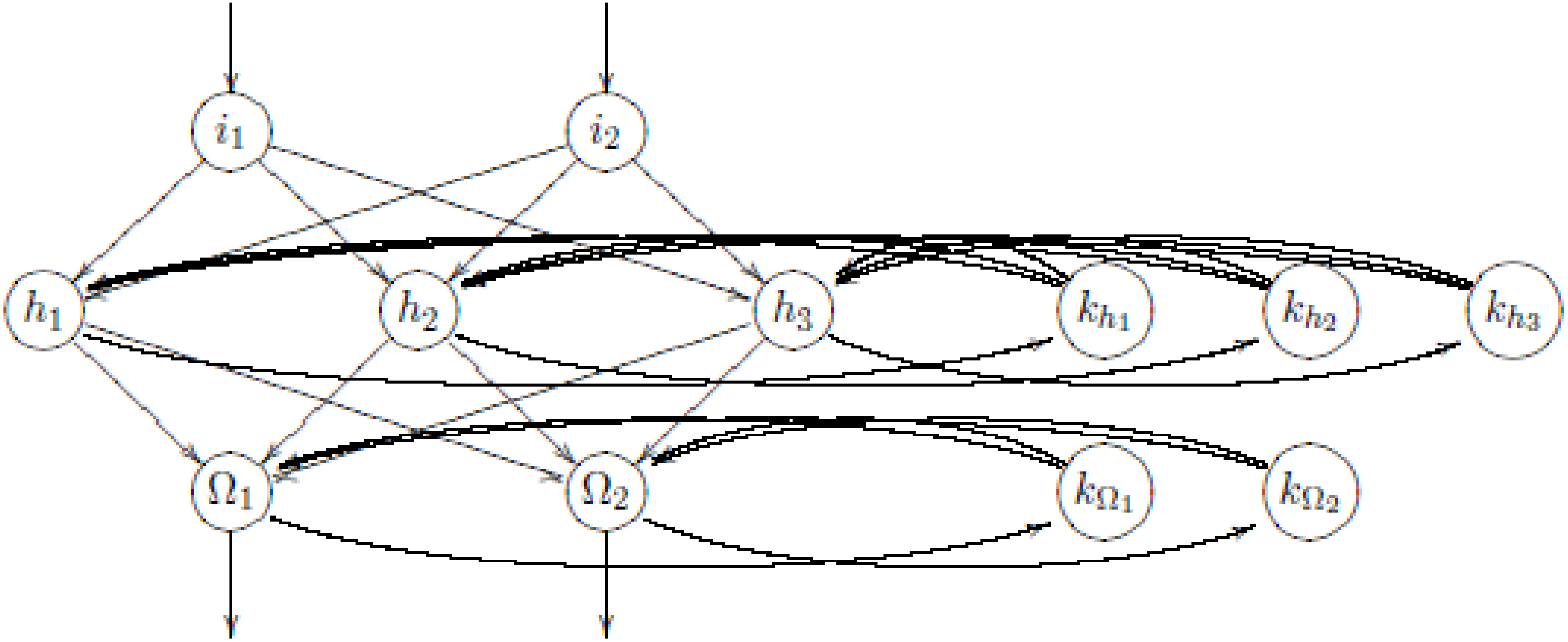
Jordan networks

- **Definition** 7.1 (Context neuron). A context neuron k receives the output value of another neuron i at a time t and then reenters it into the network at a time (t + 1).
- **Definition** 7.2 (Jordan network). A Jordan network is a multilayer perceptron with one context neuron per output neuron. The set of context neurons is called K. The context neurons are completely linked toward the input layer of the network.

Elman networks

- have context neurons but one layer of context neurons per information processing neuron layer
- the outputs of each hidden neuron or output neuron are led into the associated context layer
- from there it is reentered into the complete neuron layer during the next time step
- So the complete information processing part of the MLP exists a second time as a "context version" – which once again considerably increases dynamics and state variety.

Elman networks



Elman networks

- **Definition** 7.3 (Elman network). An Elman network is an MLP with one context neuron per information processing neuron. The set of context neurons is called K. This means that there exists one context layer per information processing neuron layer with exactly the same number of context neurons. Every neuron has a weighted connection to exactly one context neuron while the context layer is completely linked towards its original layer.

Training recurrent networks

- Unfolding in time
  - recurrent networks the delta values would backpropagate cyclically through the network again and again, which makes the training more difficult.
  - we cannot know which of the many generated delta values for a weight should be selected for training.
  - we cannot definitely know when learning should be stopped.
  - The advantage of recurrent networks are great state dynamics within the network;
  - The disadvantage of recurrent networks is that these dynamics are also granted to the training and therefore make it difficult.

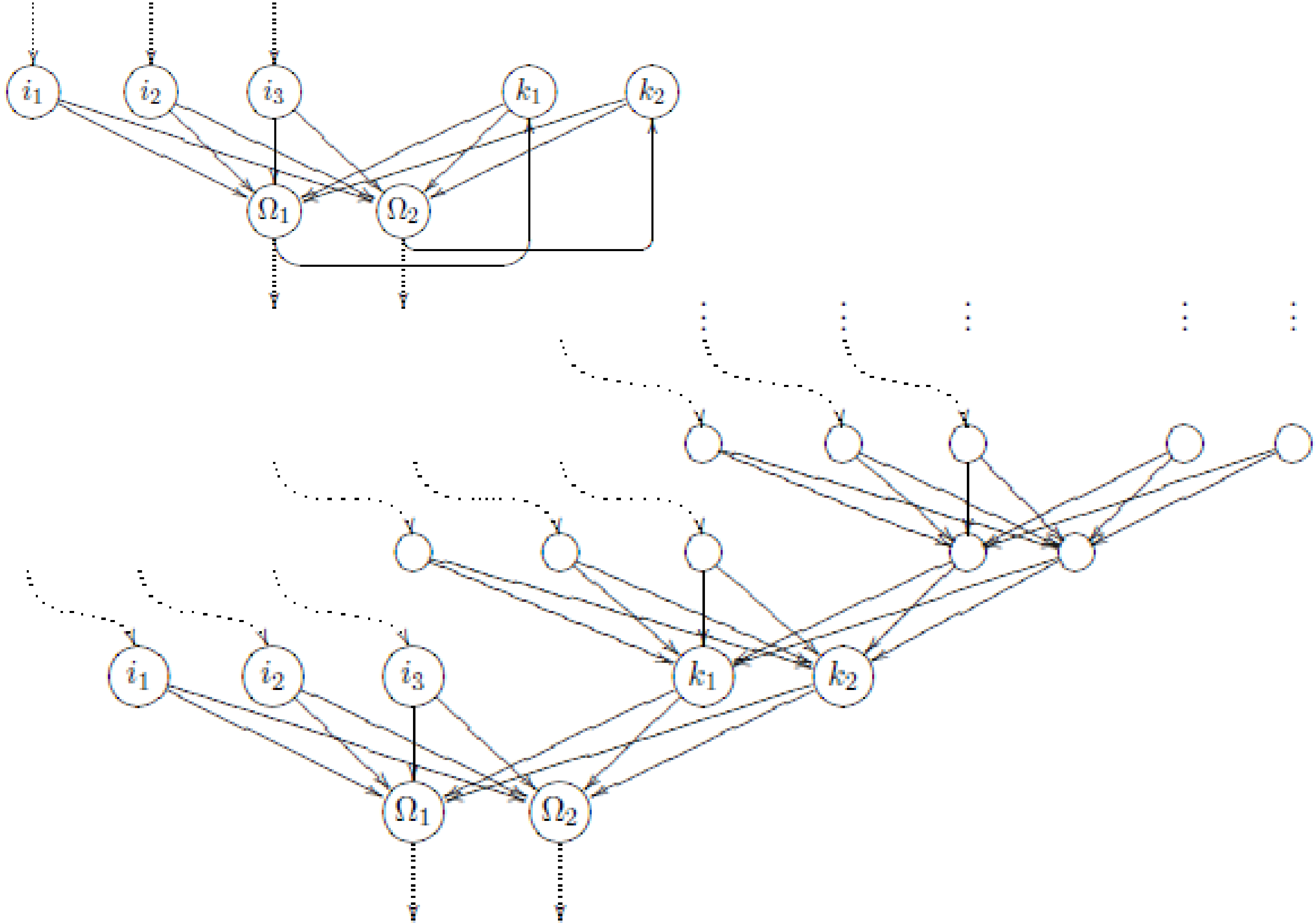
Unfolding in time

- The advantage of recurrent networks are great state dynamics within the network;
- The disadvantage of recurrent networks is that these dynamics are also granted to the training and therefore make it difficult.

Unfolding in time

- we have to backtrack the recurrences and place earlier instances of neurons in the network – thus creating a larger,
- but forward-oriented network without recurrences.
- This enables training a recurrent network with any training strategy developed for non-recurrent ones.
- the input is entered as teaching input into every "copy" of the input neurons.
- This can be done for a discrete number of time steps.

Unfolding in time



Unfolding in time

- is particularly useful if we receive the impression that the closer past is more important for the network than the one being further away.
- The reason for this is that backpropagation has only little influence in the layers farther away from the output

Disadvantages of unfolding in time

- the training of such an unfolded network will take a long time since a large number of layers could possibly be produced.
- A problem that is no longer negligible is the limited computational accuracy of ordinary computers, which is exhausted very fast because of so many nested computations

References

- **A Brief Introduction to Neural Networks,** David Kriesel, http://www.dkriesel.com/en/science/neural_networks