Neural Networks

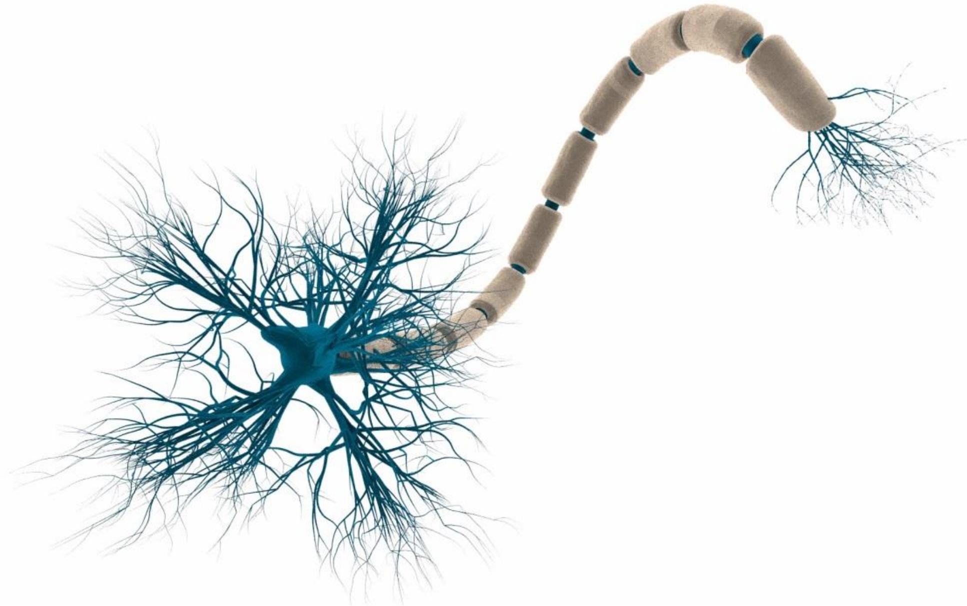
1. The brain
2. Neural networks in data mining
3. Properties of neural networks
4. Applications of neural networks

Artificial neural networks

Initially introduced in order to

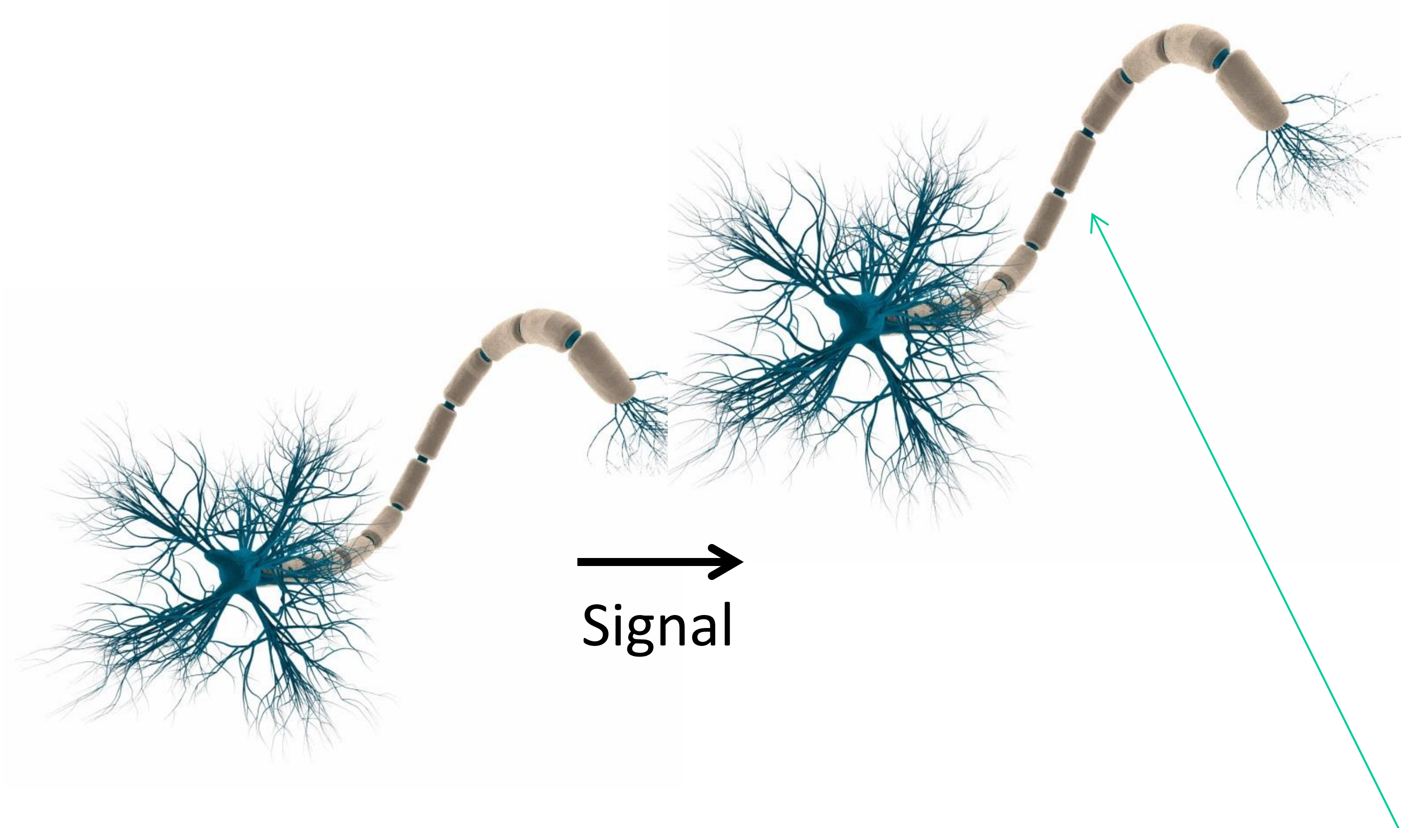
- Study the brain using computational methods
- Develop intelligent systems by mimicking the brain’s ability to learn
- The term “Data Mining” was first used to refer to the application of AI techniques to business problems
  - Although of course now the term is used more broadly (i.e., statistical data mining, non-business problems)

1. The Brain



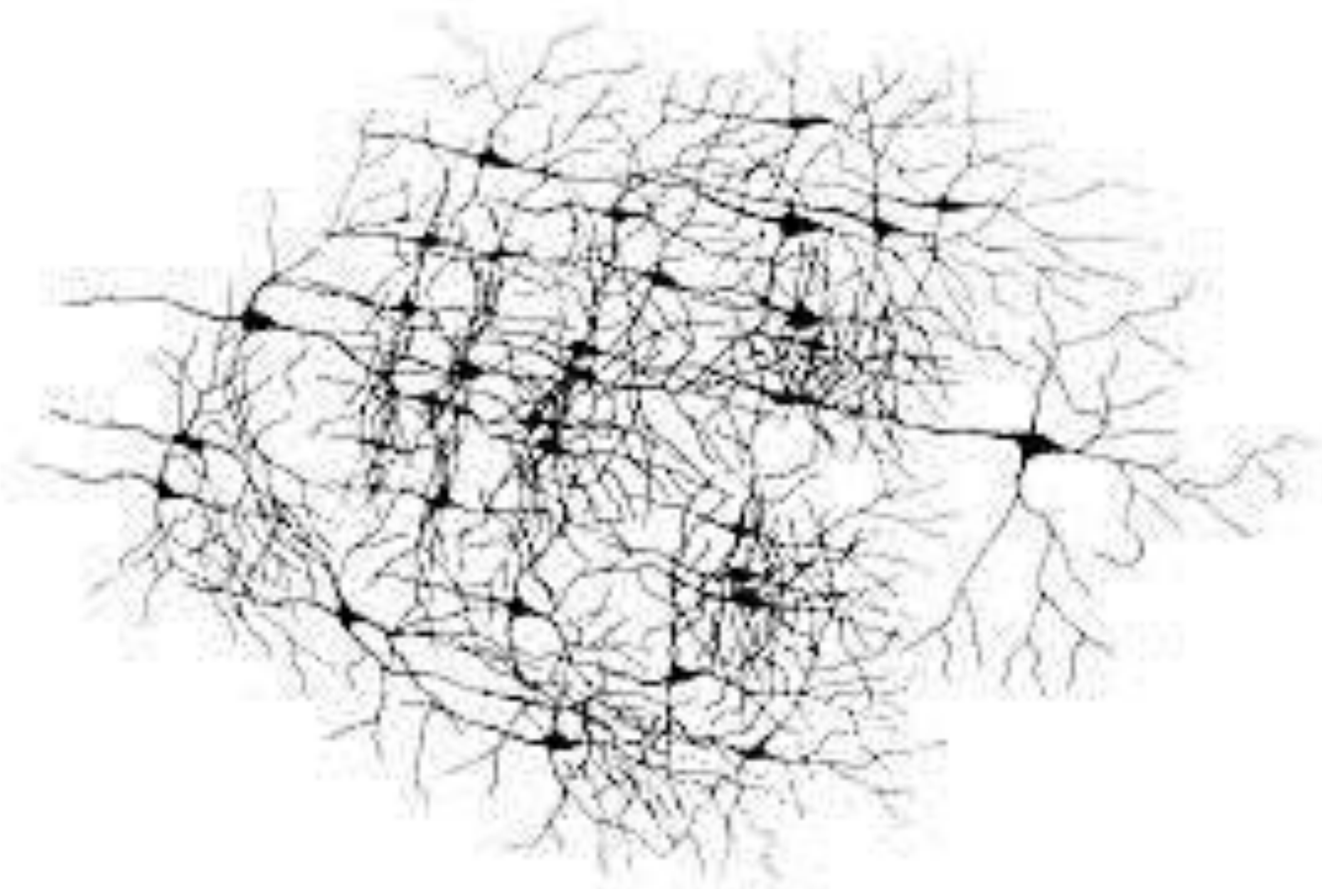
- The human brain has approx 100bn neurons
- Each neuron receives input from (typically) 1-10 thousand other neurons via its dendrites (on the left). Each input affects the excitement of the neuron: some positively and some negatively.
- When the neuron’s excitement level exceeds its excitement threshold, the neuron fires a signal to other neurons via its axon terminals (on the right)

Inter-connected neurons



The neuron cell is typically 0.1mm in diameter; the axon can be of varying length

Inter-connected neurons



Input and output neurons

- Input neurons receive inputs from the sensors outside of the brain
  - E.g., eyes, ears, …
- Output neurons send their signals outside of the brain
  - E.g., to the muscles

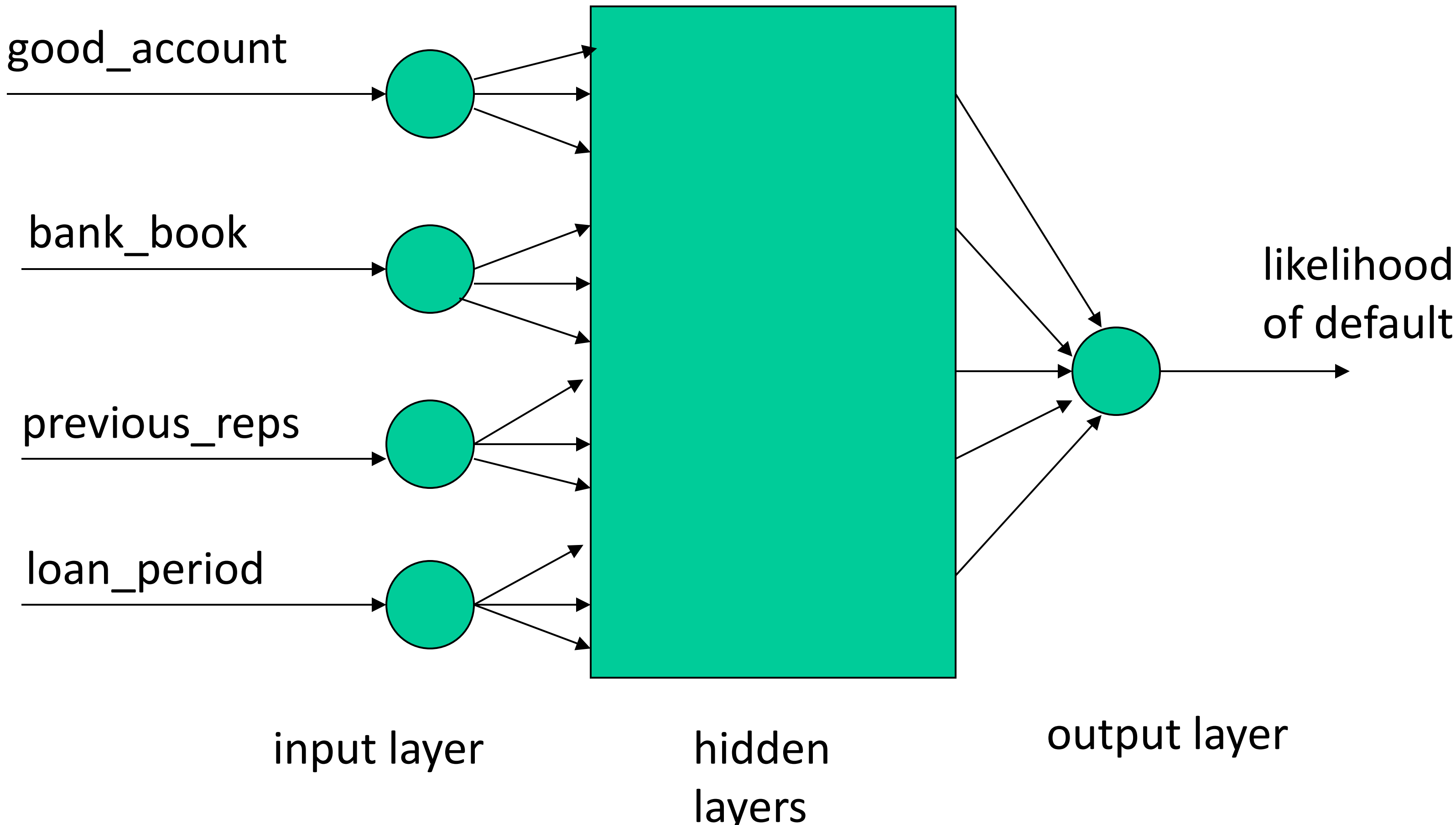
2. Neural nets in data mining

- No allusions to intelligence in the mining context . . .
- Neural nets provide a good method of fitting a model to a training set of data
  - using neural learning
- Used for directed and undirected mining
- More poorly understood than other techniques

Network architecture

- Neurons (nodes)
  - Input nodes
    - one for each input variable
  - Output nodes
    - one for each output variable
  - Hidden nodes/layers
- Connections (with weights) passing output from one node to another

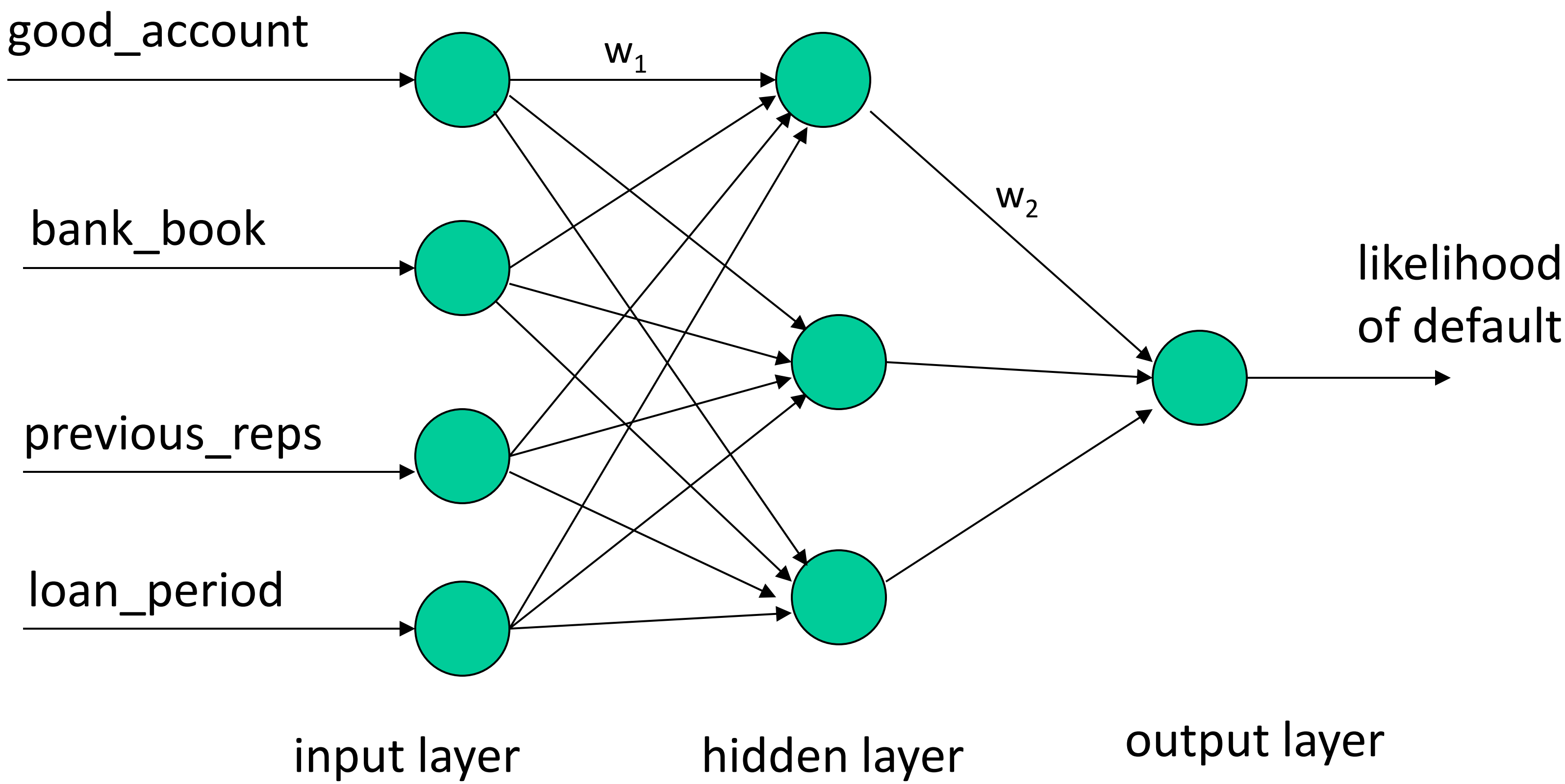
Network architecture



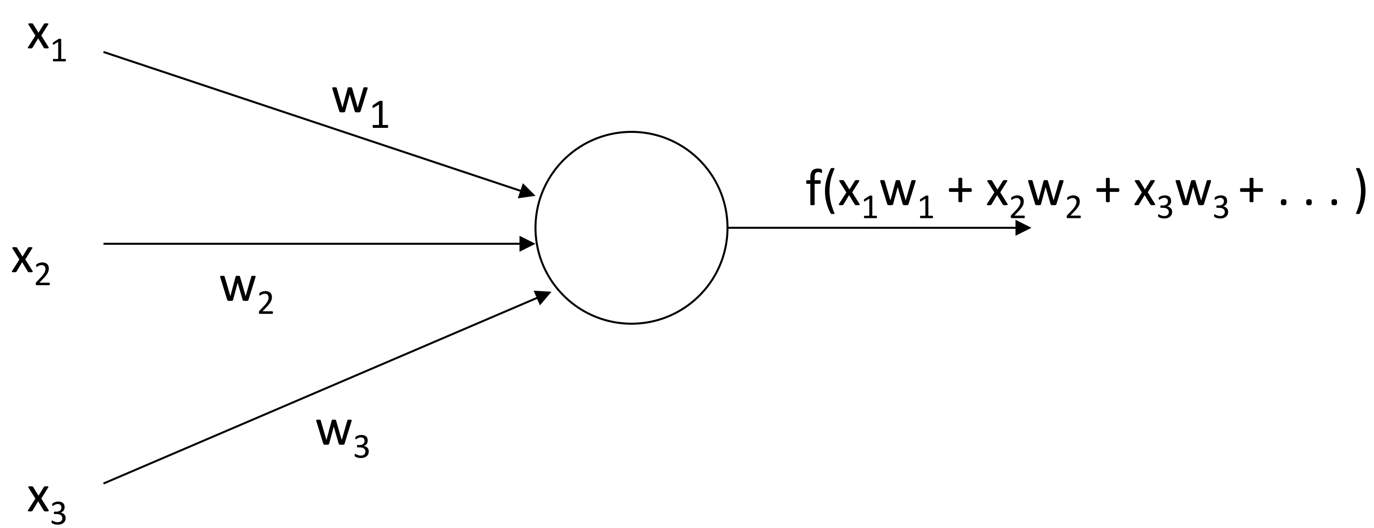
Multi-layer perceptron (MLP)

- The most common network employed in data mining is the multi-layer perceptron
  - fully connected
    - every node in layer n has a connection *to* every node in layer n+1
      - where of course layer 0 is the input layer
      - can have several hidden layers
  - feed forward
    - no connections from layer n+1 *to* layer n
  - single output node

MLP architecture



The activation function

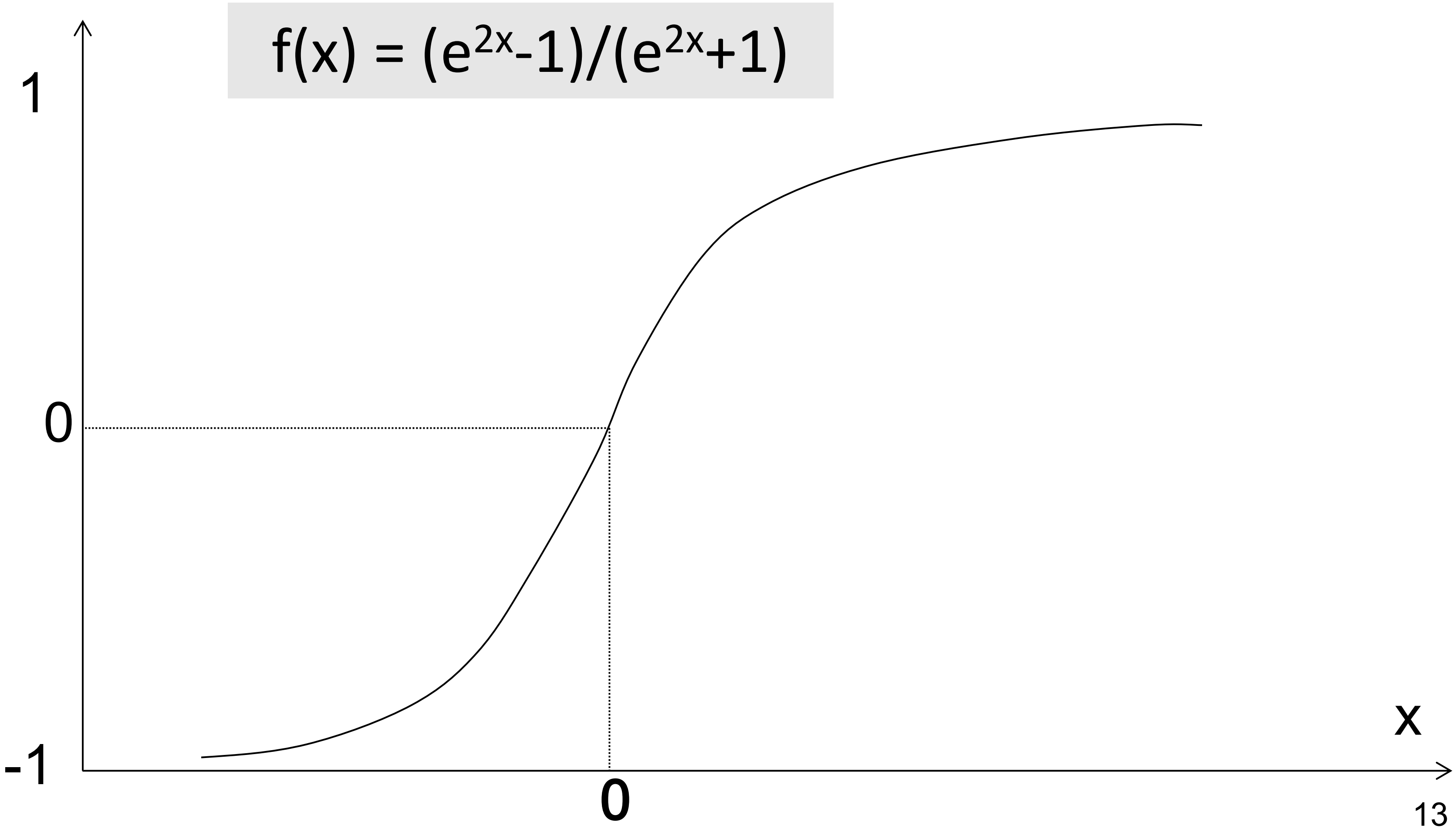


If the node has inputs $x_1, x_2, x_3, \ldots$ along connections with weights $w_1, w_2, w_3, \ldots$, then the output of the node is given by the activation function

$$f(x_1w_1 + x_2w_2 + x_3w_3 + \ldots)$$

where f is a *transfer function*

Sigmoidal transfer function



Training neural nets

Back-propagation - requires a Training Set of records for which the target value is known

Define initial network

Repeat - a *large* number of times

{ Feed data record (from Training Set) into network

  Error = result produced - known target value

  Use the Error to modify the weights *slightly*

}

Supervised learning

Training neural nets

- Learning parameters need to be set with care - "fast learning" may be subject to instability
- Training needs stopping criteria
- Usually a single hidden layer will suffice
- Usually employ fewer nodes in hidden layer than input layer
- Tools may find the best number of nodes for the hidden layer
  - overfitting

Dealing with overfitting

Define initial network

Repeat

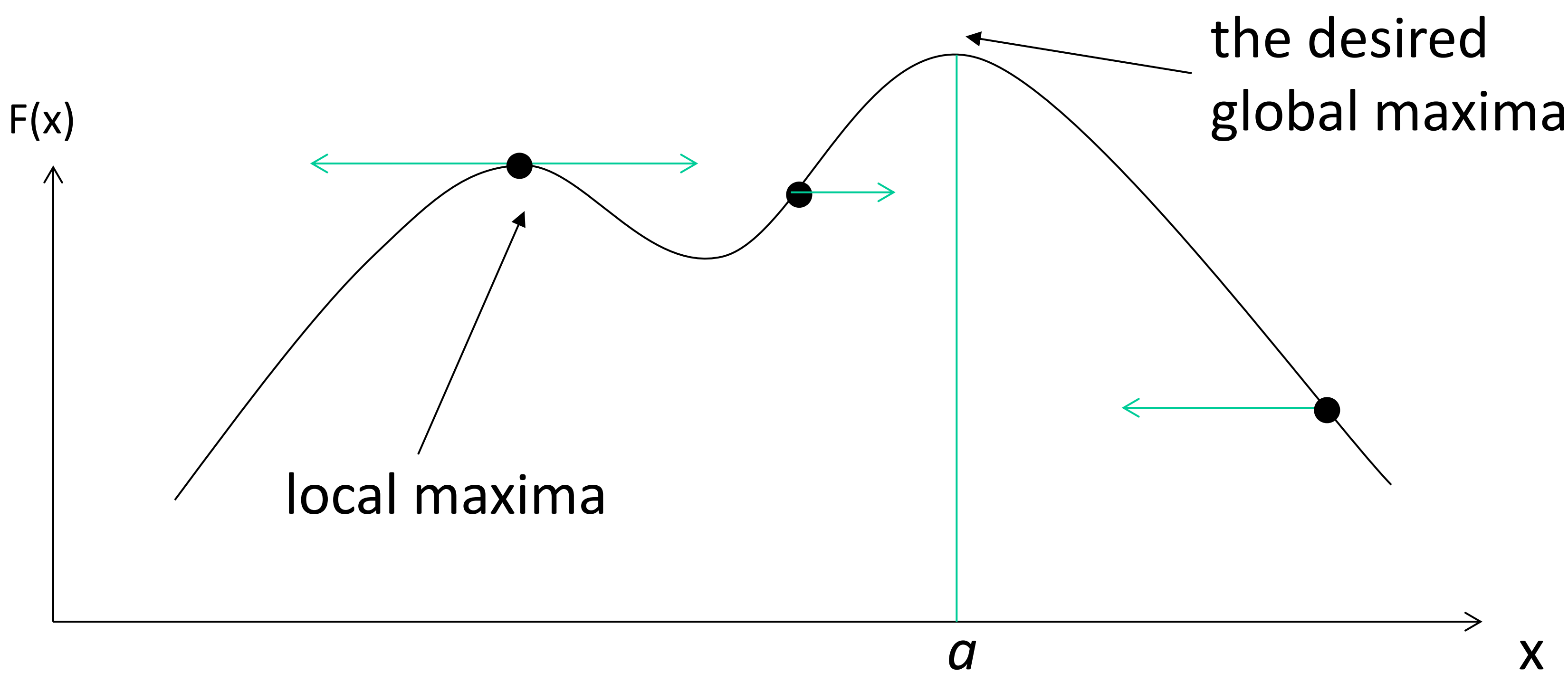
{ Train network using one iteration of Training Set

Evaluate performance of network on Test Set

}

Pick network version with best performance (on Test Set)

Optimisation and hill climbing



A *hill climbing* attempt to find the global maxima (i.e., to find *a*) can easily get stuck at the local maxima – because at each step it only makes a local analysis

3. Properties of neural nets

- Data sets need to provide good coverage
  - Which may not be representative
- May benefit from reduction in the number of input variables: preliminary/statistical analysis. More input variables:
  - demand a larger training set in order to have good coverage & require greater training times
  - increase chances of finding sub-optimal solutions

Properties of neural nets

- Does not produce insights - black box
- Missing input values must be estimated
- Categoricals must be converted to numeric without introducing spurious ordering, e.g., by flattening
  - e.g., replace variable *colour* by binary variables red, blue, green, . . . introduces a lot of new input variables

Properties of neural nets

- Inputs should be normalised (to say (-1, 1))
  - If we use a sigmoidal transfer function in the input neurons – this would have this effect
- With a sigmoidal transfer function in the output node, the output will be normalised too

4. Applications of neural nets

- Forecasting TV audiences
- Credit scoring
- Predicting loan defaults
- Forecasting customer loyalty
- Detecting credit card fraud
- Predicting house prices
- Modelling time series (e.g., stock market prediction)
- Forecasting product demand
- Medical data classification
- …