Lecture 14

Knowledge engineering:
Building neural network based systems

- Will neural network work for my problem?
  - Character recognition neural networks
  - Prediction neural networks
  - Classification neural networks with competitive learning
  - Clustering with a self-organising neural network
- Summary

Will a neural network work for my problem?

Neural networks represent a class of very powerful, general-purpose tools that have been successfully applied to prediction, classification and clustering problems. They are used in a variety of areas, from speech and character recognition to detecting fraudulent transactions, from medical diagnosis of heart attacks to process control and robotics, from predicting foreign exchange rates to detecting and identifying radar targets.

Case study 4
Character recognition neural networks

Recognition of both printed and handwritten characters is a typical domain where neural networks have been successfully applied.

Optical character recognition systems were among the first commercial applications of neural networks.

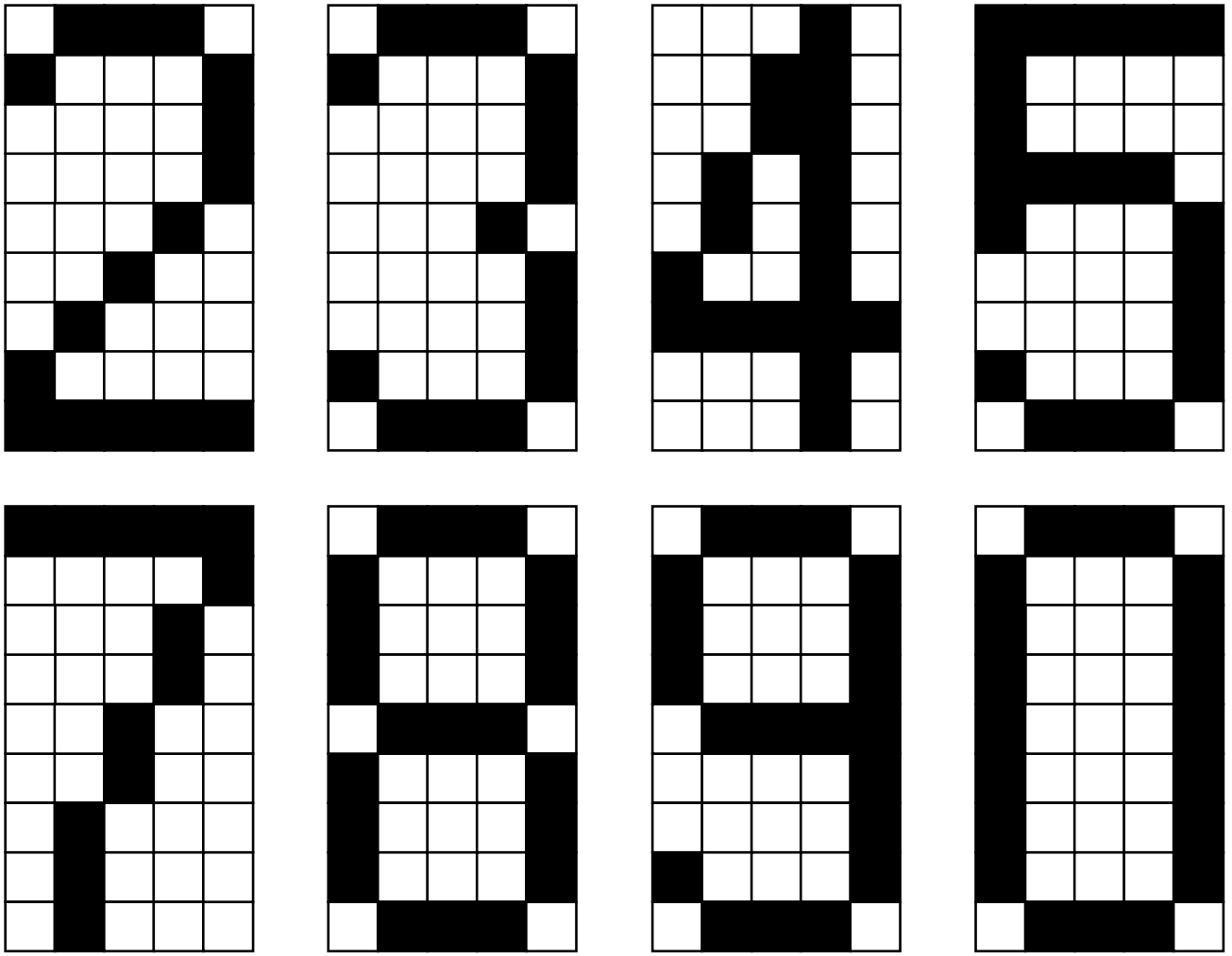
We demonstrate an application of a multilayer feedforward network for printed character recognition.

For simplicity, we can limit our task to the recognition of digits from 0 to 9. Each digit is represented by a 5×9 bit map.

In commercial applications, where a better resolution is required, at least 16×16 bit maps are used.

Bit maps for digit recognition

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45



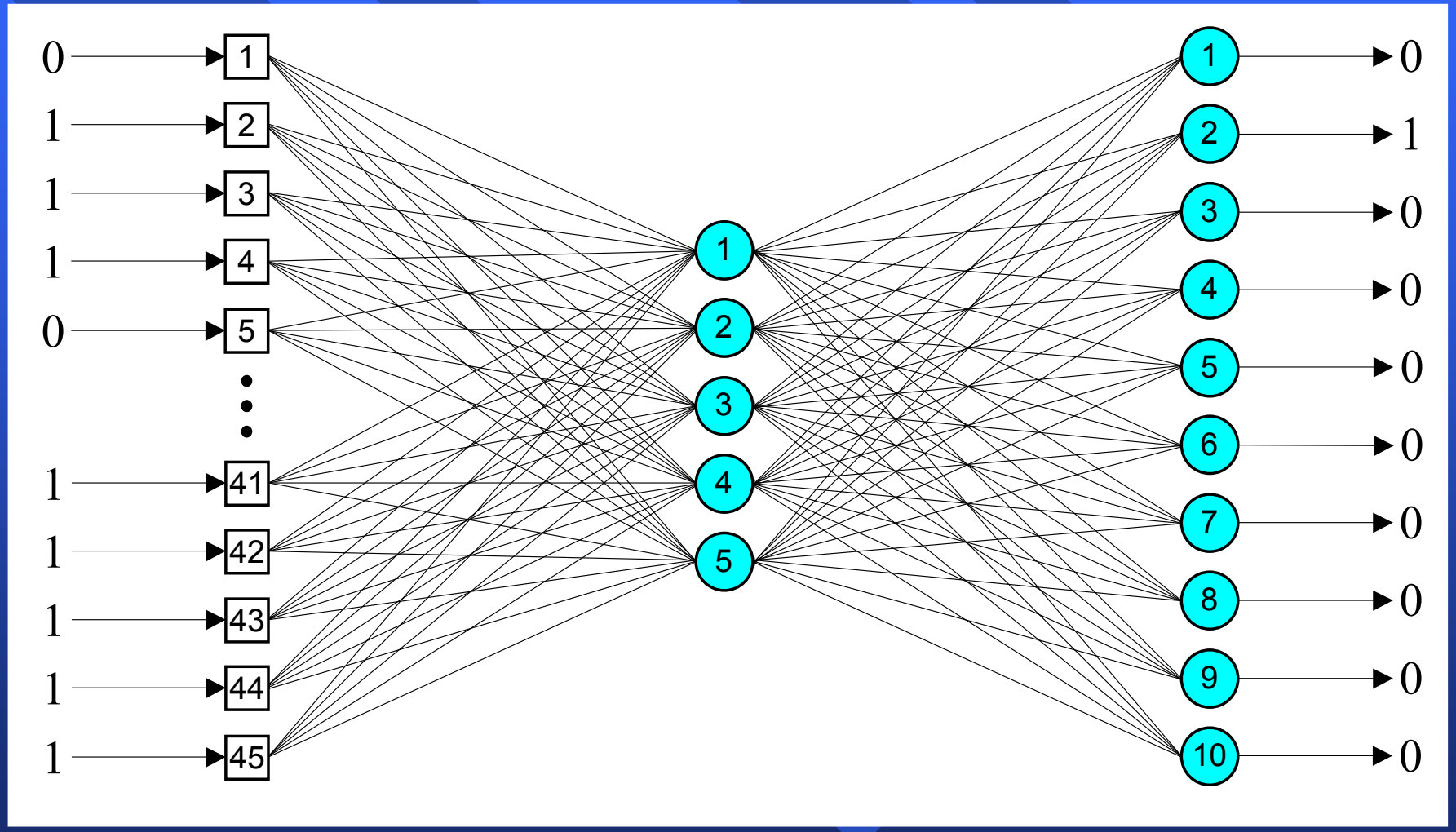
How do we choose the architecture of a neural network?

- The number of neurons in the input layer is decided by the number of pixels in the bit map. The bit map in our example consists of 45 pixels, and thus we need 45 input neurons.
- The output layer has 10 neurons – one neuron for each digit to be recognised.

How do we determine an optimal number of hidden neurons?

- Complex patterns cannot be detected by a small number of hidden neurons; however too many of them can dramatically increase the computational burden.
- Another problem is **overfitting**. The greater the number of hidden neurons, the greater the ability of the network to recognise existing patterns. However, if the number of hidden neurons is too big, the network might simply memorise all training examples.

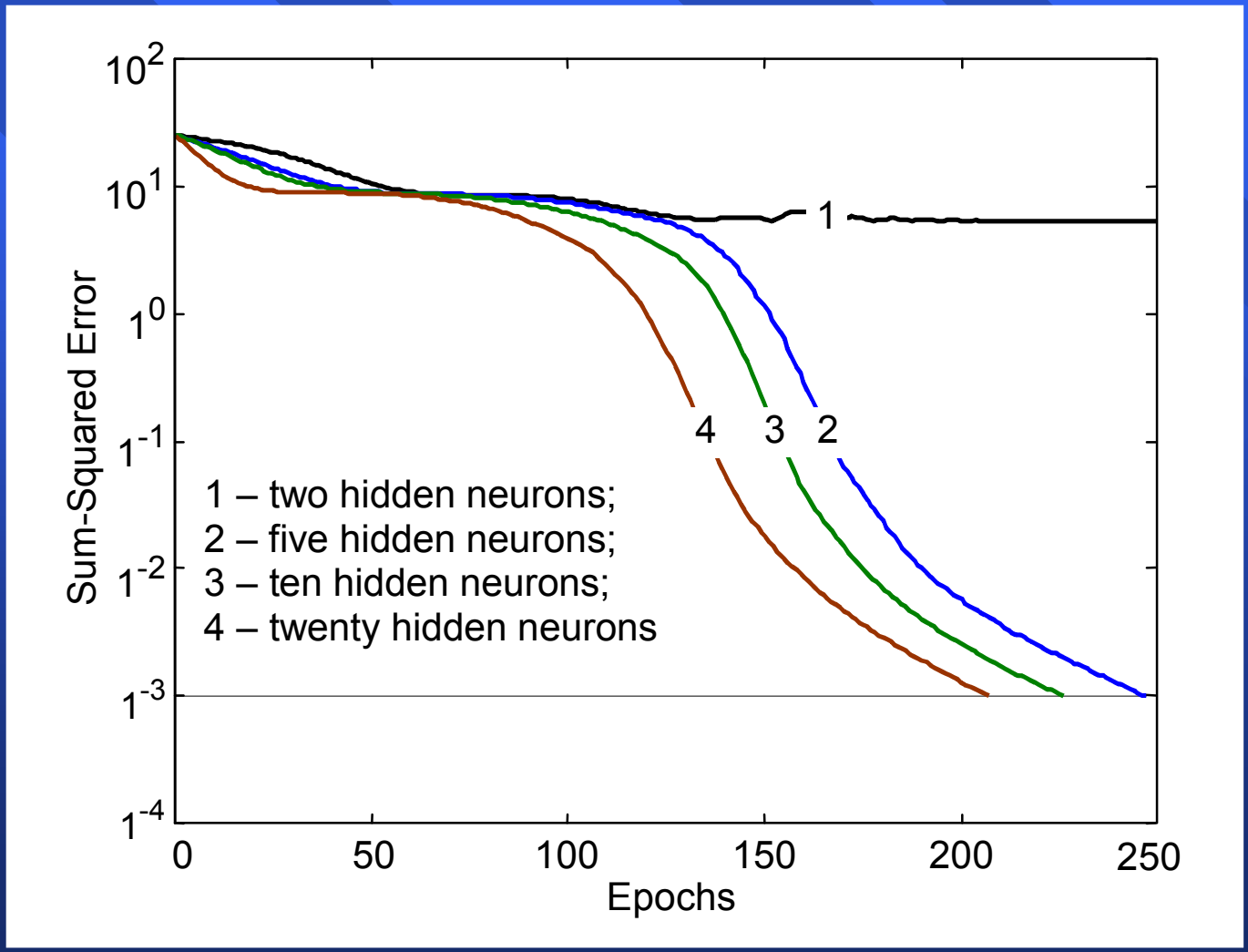
Neural network for printed digit recognition



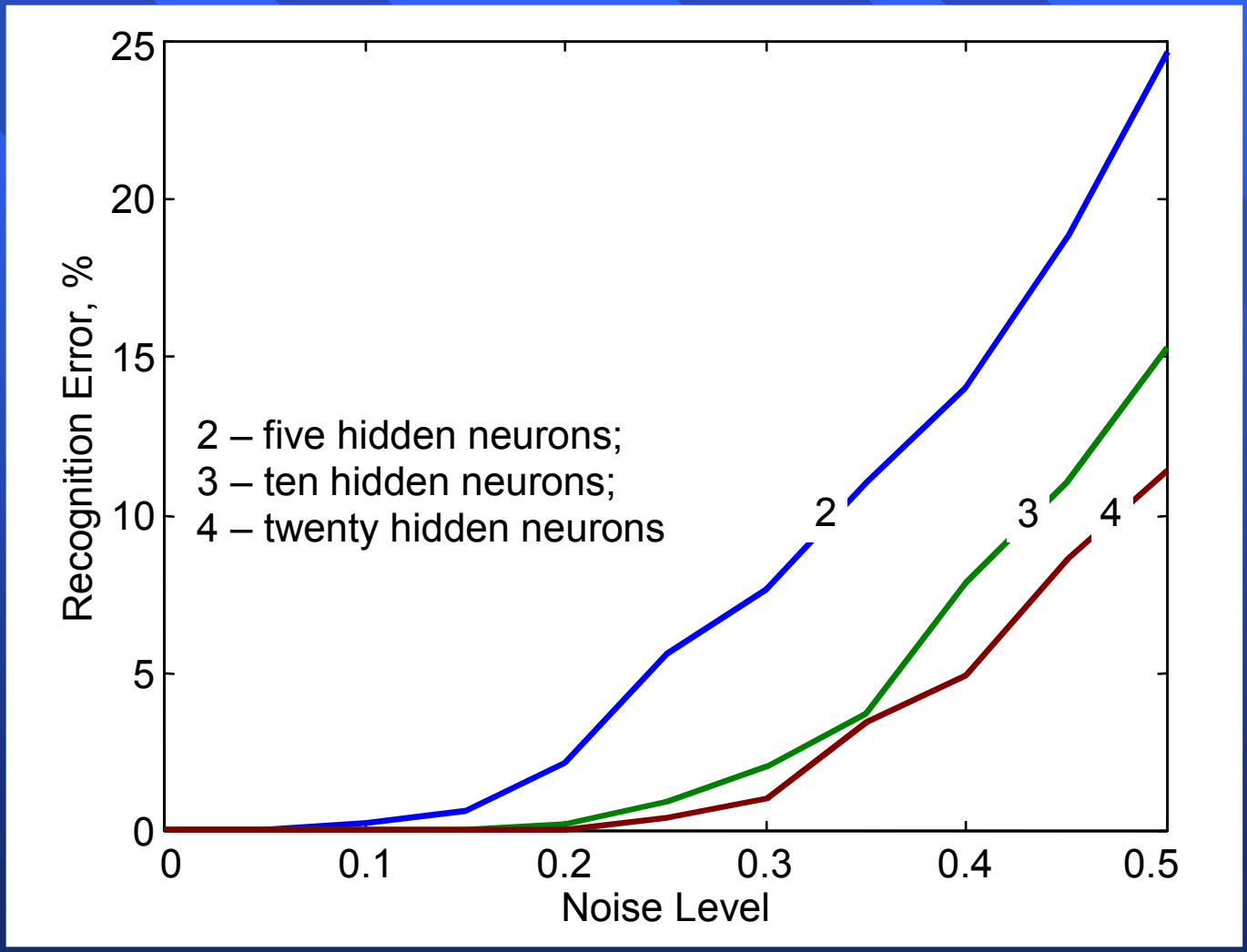
What are the test examples for character recognition?

- A test set has to be strictly independent from the training examples.
- To test the character recognition network, we present it with examples that include "noise" – the distortion of the input patterns.
- We evaluate the performance of the printed digit recognition networks with 1000 test examples (100 for each digit to be recognised).

Learning curves of the digit recognition three-layer neural networks



Performance evaluation of the digit recognition neural networks

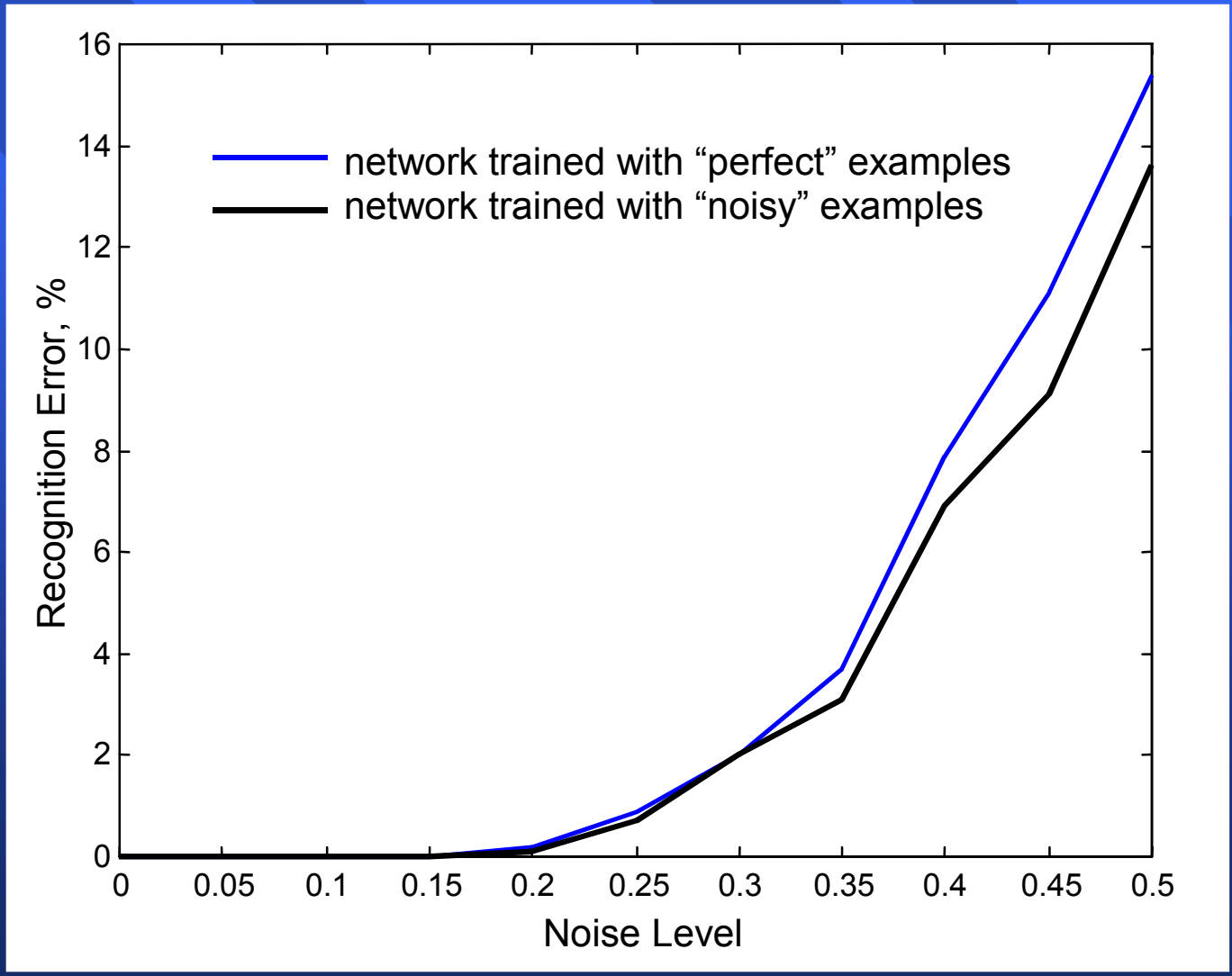


Can we improve the performance of the character recognition neural network?

A neural network is as good as the examples used to train it.

Therefore, we can attempt to improve digit recognition by feeding the network with "noisy" examples of digits from 0 to 9.

Performance evaluation of the digit recognition network trained with "noisy" examples



Case study 5
Prediction neural networks

As an example, we consider a problem of predicting the market value of a given house based on the knowledge of the sales prices of similar houses.

- In this problem, the inputs (the house location, living area, number of bedrooms, number of bathrooms, land size, type of heating system, etc.) are well-defined, and even standardised for sharing the housing market information between different real estate agencies.
- The output is also well-defined – we know what we are trying to predict.
- The features of recently sold houses and their sales prices are examples, which we use for training the neural network.

Network generalisation

An appropriate number of training examples can be estimated with **Widrow's rule of thumb**, which suggests that, for a good generalisation, we need to satisfy the following condition:

$$N = \frac{n_w}{e}$$

where N is the number of training examples, n_w is the number of synaptic weights in the network, and e is the network error permitted on test.

Massaging the data

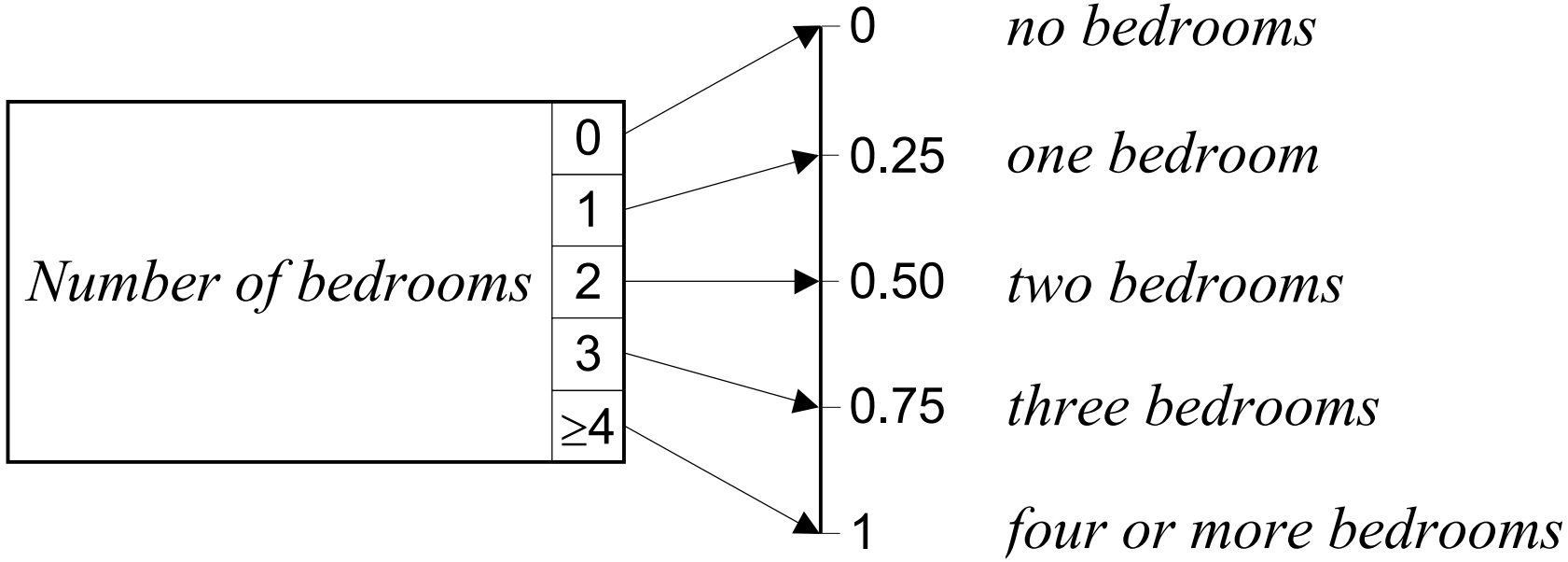
Data can be divided into three main types: continuous, discrete and categorical .

- **Continuous data** vary between two pre-set values – minimum and maximum, and can be mapped, or massaged, to the range between 0 and 1 as:

$$massaged\ value = \frac{actual\ value - minimum\ value}{maximum\ value - minimum\ value}$$

- **Discrete data**, such as the number of bedrooms and the number of bathrooms, also have maximum and minimum values. For example, the number of bedrooms usually ranges from 0 to 4.

Massaging the data

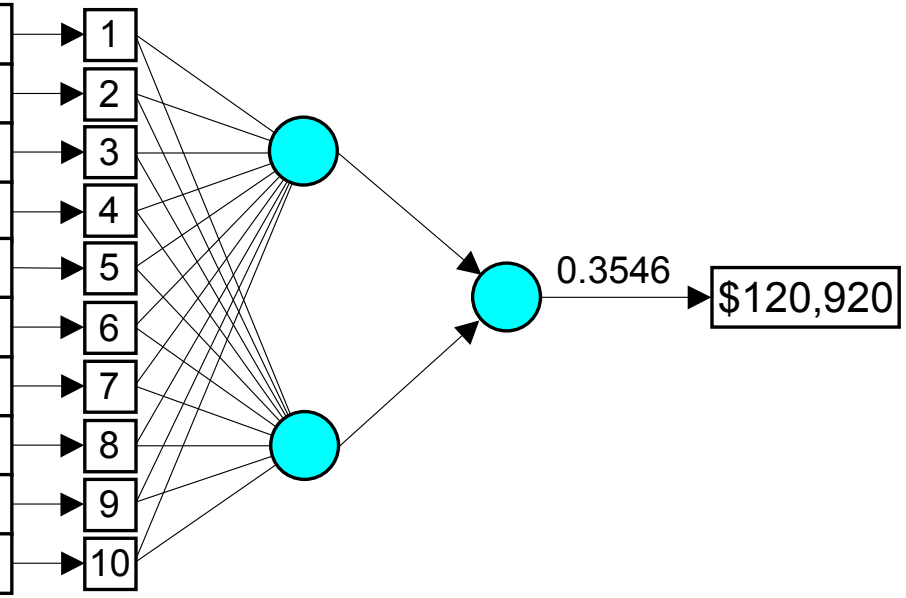


- **Categorical data**, such as gender and marital status, can be massaged by using ***1 of N coding***. This method implies that each categorical value is handled as a separate input.

  For example, marital status, which can be either single, divorced, married or widowed, would be represented by four inputs. Each of these inputs can have a value of either 0 or 1. Thus, a married person would be represented by an input vector [0 0 1 0].

Feedforward neural network for real-estate appraisal

Location rating	7	0.7000	1
House condition	8	0.8000	2
Number of bedrooms	3	0.7500	3
Number of bathrooms	1	0.5000	4
Living area	121.00 m^2	0.3550	5
Land size	682.00 m^2	0.4682	6
Garage	double	1.0000	7
Type of heating system	wood heater	1.0000	8
		0.0000	9
		0.0000	10



How do we validate results?

To validate results, we use a set of examples never seen by the network.

Before training, all the available data are randomly divided into a training set and a test set.

Once the training phase is complete, the network's ability to generalise is tested against examples of the test set.

Case study 6
Classification neural networks with competitive learning

As an example, we will consider an iris plant classification problem.

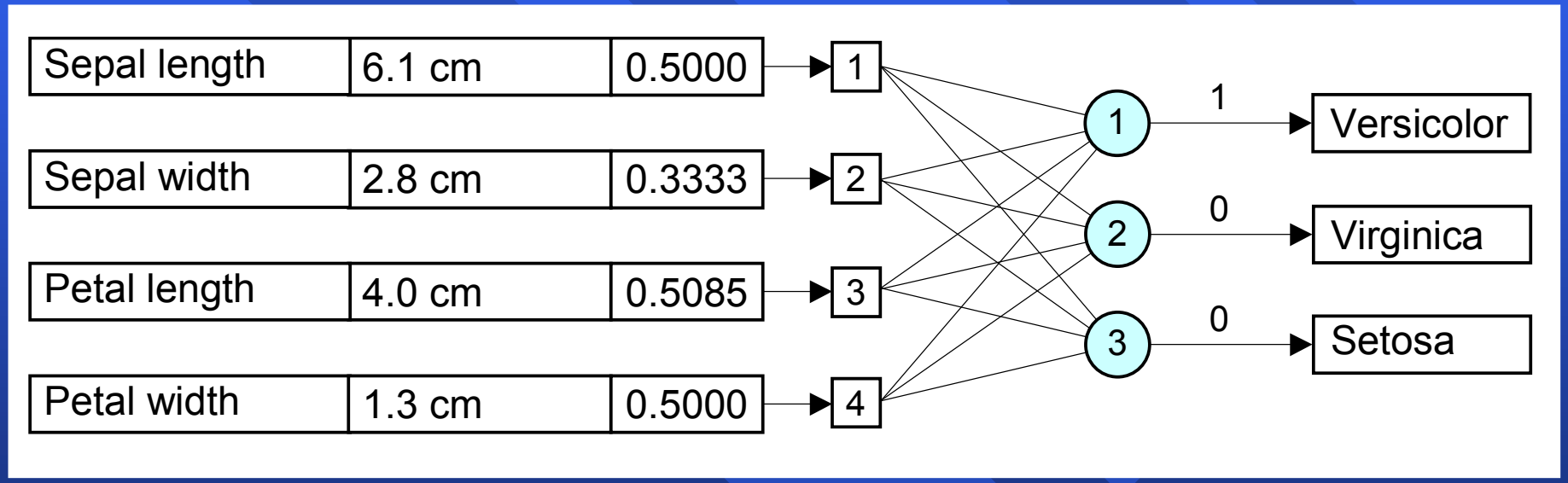
Suppose, we are given a data set with several variables but we have no idea how to separate it into different classes because we cannot find any unique or distinctive features in the data.

Clusters and clustering

- Neural networks can discover significant features in input patterns and learn how to separate input data into different classes. A neural network with competitive learning is a suitable tool to accomplish this task.
- The competitive learning rule enables a single-layer neural network to combine similar input data into groups or ***clusters***. This process is called ***clustering***. Each cluster is represented by a single output.

- For this case study, we will use a data set of 150 elements that contains three classes of iris plants – *setosa*, *versicolor* and *virginica*.
- Each plant in the data set is represented by four variables: sepal length, sepal width, petal length and petal width. The sepal length ranges between 4.3 and 7.9 cm, sepal width between 2.0 and 4.4 cm, petal length between 1.0 and 6.9 cm, and petal width between 0.1 and 2.5 cm.

Neural network for iris plant classification



Massaging the data

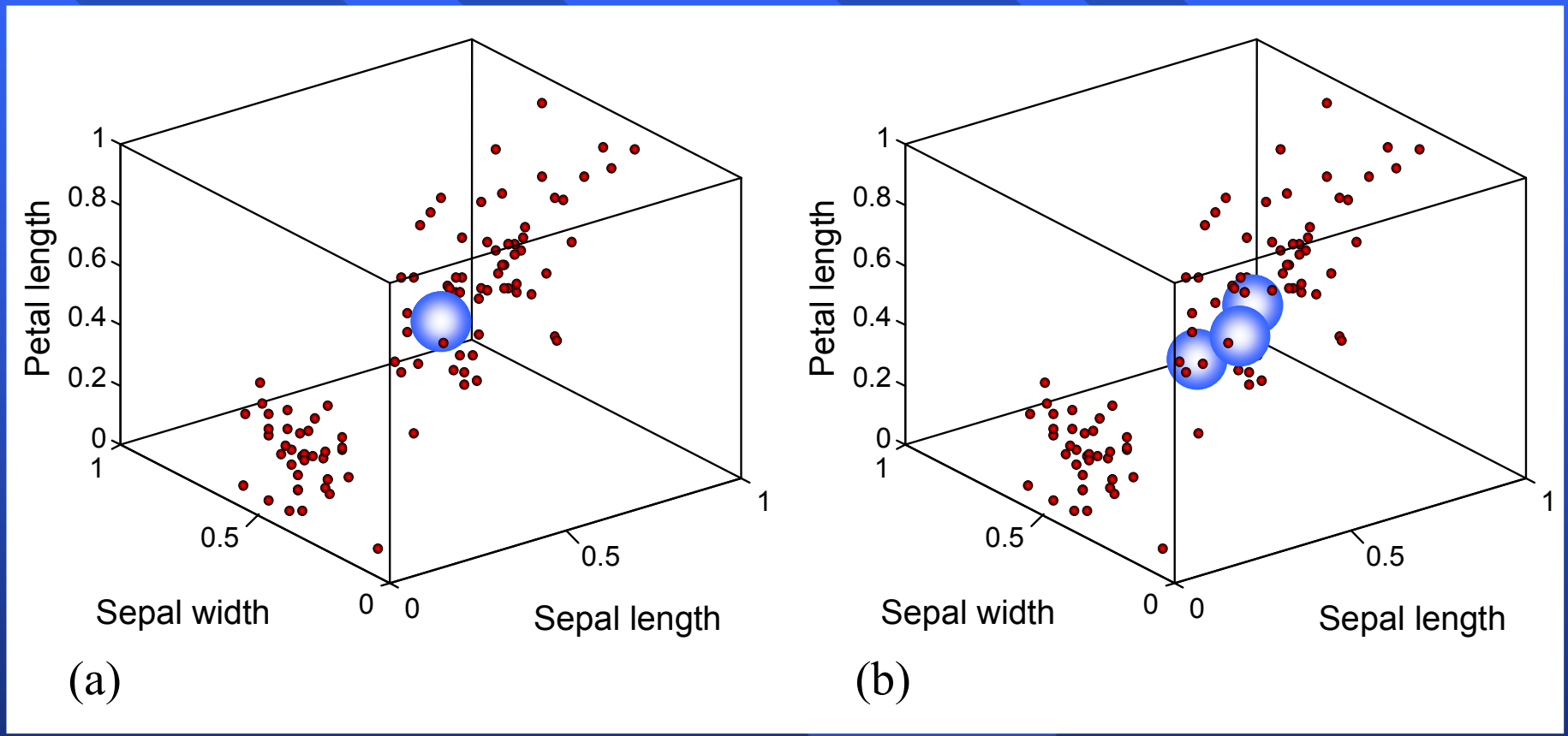
- Before the network is trained, the data must be massaged and then divided into training and test sets.
- The Iris plant data are continuous, vary between some minimum and maximum values, and thus can easily be massaged to the range between 0 and 1 using the following equation.

$$massaged\ value = \frac{actual\ value - minimum\ value}{maximum\ value - minimum\ value}$$

Massaged values can then be fed to the network as its inputs.

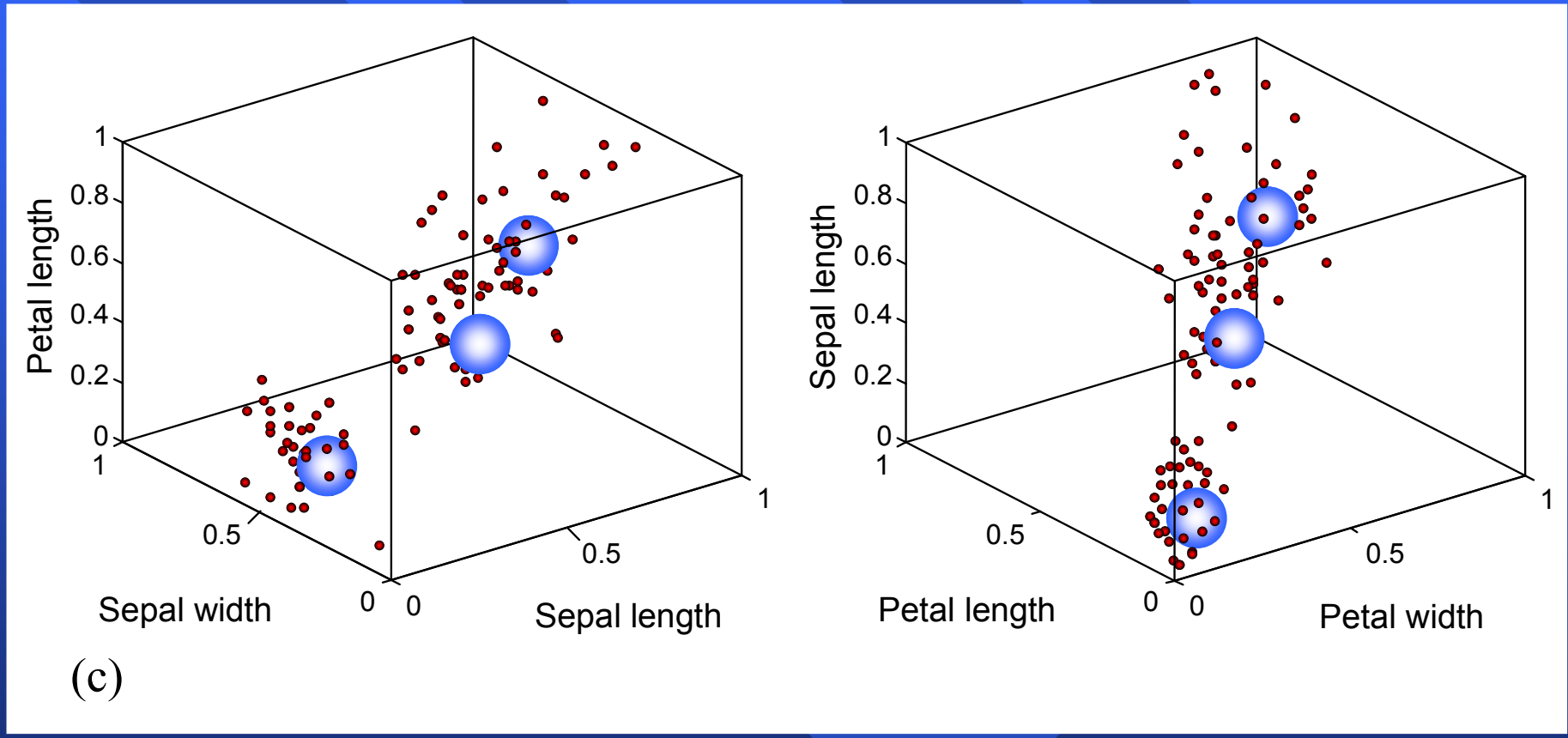
- The next step is to generate training and test sets from the available data. The 150-element Iris data is randomly divided into a training set of 100 elements and a test set of 50 elements.
- Now we can train the competitive neural network to divide input vectors into three classes.

Competitive learning in the neural network for iris plant classification



(a) initial weights; (b) weights after 100 iterations

Competitive learning in the neural network for iris plant classification

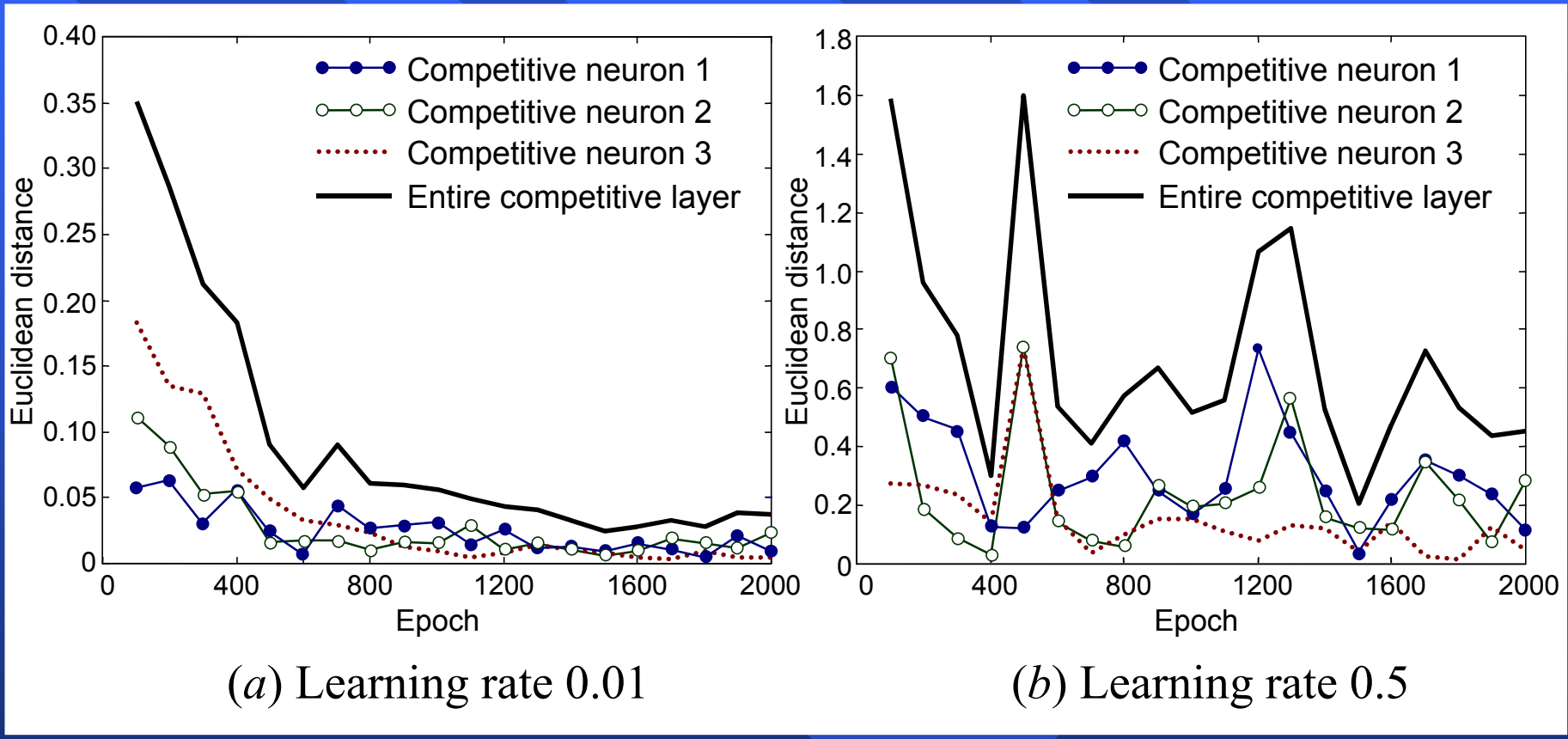


(c) weights after 2,000 iterations

How do we know when the learning process is complete?

- In a competitive neural network, there is no obvious way of knowing whether the learning process is complete or not. We do not know what desired outputs are, and thus cannot compute the sum of squared errors – a criterion used by the back-propagation algorithm.
- Therefore, we should use the ***Euclidean distance*** criterion instead. When no noticeable changes occur in the weight vectors of competitive neurons, a network can be considered to have converged.

Learning curves for competitive neurons of the iris classification neural network



(*a*) Learning rate 0.01

(*b*) Learning rate 0.5

How do we associate an output neuron with a particular class?

- Competitive neural networks enable us to identify clusters in input data. However, since clustering is an unsupervised process, we cannot use it directly for labelling output neurons.
- In most practical applications, the distribution of data that belong to the same cluster is rather dense, and there are usually natural valleys between different clusters. As a result, the position of the centre of a cluster often reveals distinctive features of the corresponding class.

Labelling the competitive neurons

Neuron	Weights		Dimensions of the Iris plant, cm		Class of the Iris plant
1	w_{11}	0.4355	Sepal length	5.9	*Versicolor*
	w_{21}	0.3022	Sepal width	2.7	
	w_{31}	0.5658	Petal length	4.4	
	w_{41}	0.5300	Petal width	1.4	
2	w_{12}	0.6514	Sepal length	6.7	*Virginica*
	w_{22}	0.4348	Sepal width	3.0	
	w_{32}	0.7620	Petal length	5.5	
	w_{42}	0.7882	Petal width	2.0	
3	w_{13}	0.2060	Sepal length	5.0	*Setosa*
	w_{23}	0.6056	Sepal width	3.5	
	w_{33}	0.0940	Petal length	1.6	
	w_{43}	0.0799	Petal width	0.3	

How do we decode weights into Iris dimensions?

To decode the weights of the competitive neurons into dimensions of the iris plant we simply reverse the procedure used for massaging the iris data. For example,

$$\textit{Sepal length}_{w11} = 0.4355 \times (7.9 - 4.3) + 4.3 = 5.9 \text{ cm}$$

Once the weights are decoded, we can ask an iris plant expert to label the output neurons.

Can we label the competitive neurons automatically without having to ask the expert?

- We can use a test data set for labelling competitive neurons automatically. Once training of a neural network is complete, a set of input samples representing the same class, say class *Versicolor*, is fed to the network, and the output neuron that wins the competition most of the time receives a label of the corresponding class.

- Although a competitive network has only one layer of competitive neurons, it can classify input patterns that are not linearly separable.
- In classification tasks, competitive networks learn much faster than multilayer perceptrons trained with the back-propagation algorithm, but they usually provide less accurate results.

Case study 7

Clustering with a self-organising neural network

As an example, we will build an intelligent system to identify potentially failing banks.

The danger of bank failures would be reduced significantly if we could identify banks with potential problems before they face solvency and liquidity crises.

- The reasons for bank failures include high risk-taking, interest rate volatility, poor management practices, inadequate accounting standards, and increased competition from non-depository institutions.
- In this case study, we “flag” potentially failing banks using cluster analysis.

What is cluster analysis?

- Cluster analysis is an exploratory data analysis technique that divides different objects into groups, called ***clusters***, in such a way that the degree of association between two objects is maximised if they belong to the same cluster and minimised otherwise.
- The term "cluster analysis" was first introduced over 70 years ago by **Robert Tryon**.

- In clustering, there are no predefined classes – objects are grouped together only on the basis of their similarity. For this reason, clustering is often referred to as unsupervised classification.
- There is no distinction between independent and dependent variables, and when clusters are found the user needs to interpret their meaning.

What methods are used in cluster analysis?

- We can identify three major methods used in cluster analysis. These are based on statistics, fuzzy logic and neural networks.
- In this case study, we will apply a self-organising neural network.

- For our case study, we select 100 banks and obtain their financial data from the Federal Deposit Insurance Corporation (FDIC) annual report for the last year.
- We adapt the following five ratings based on the **CAMELS** (*C*apital adequacy, *A*sset, *M*anagement, *E*arnings, *L*iquidity, and *S*ensitivity to market risk) system:

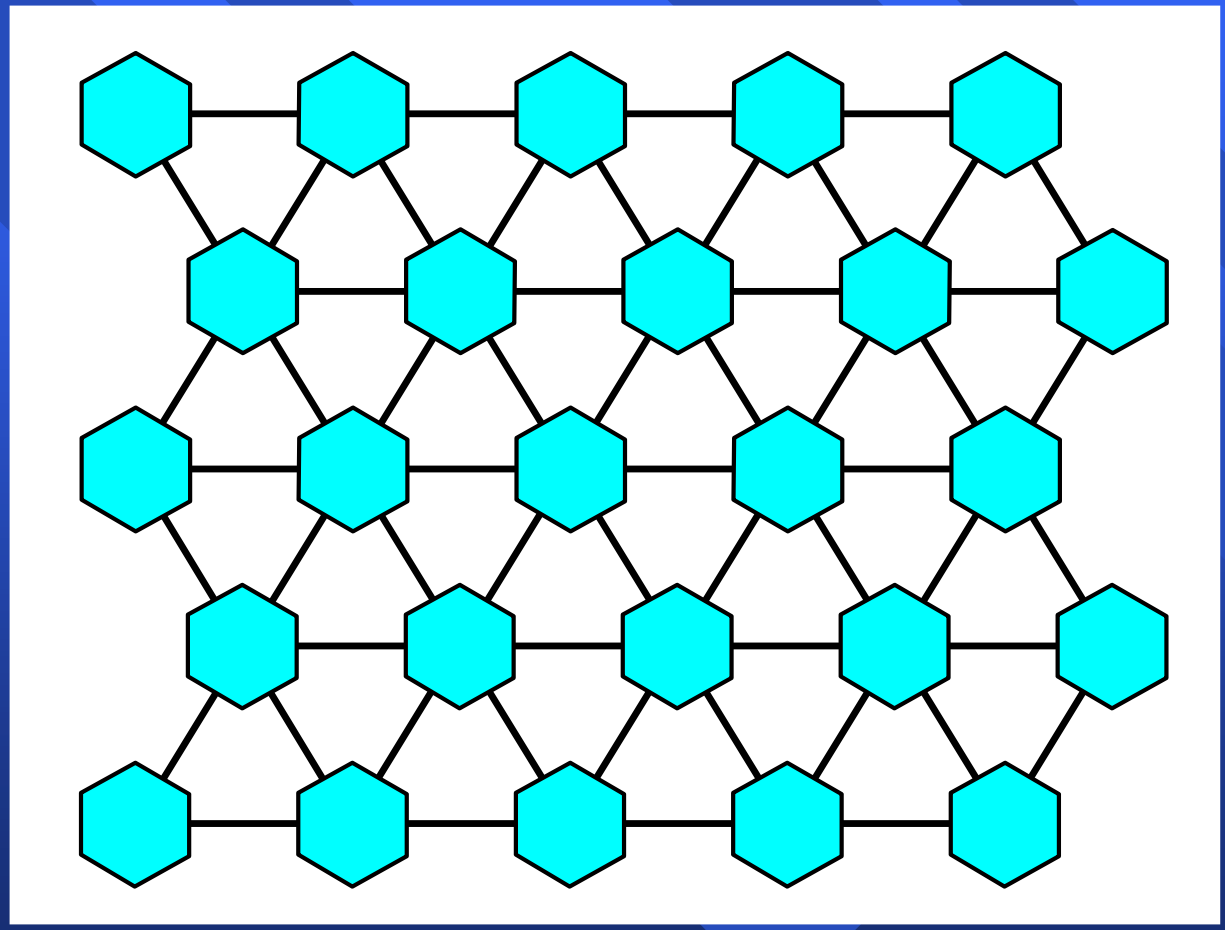
1. **NITA** – *N*et *I*ncome divided by *T*otal *A*ssets. NITA represents return on assets. Failing banks have very low or even negative values of NITA.

2. **NLLAA** – *N*et *L*oan *L*osses divided by *A*djusted *A*ssets. Adjusted assets are calculated by subtracting the total loans from the total assets. Failing banks usually have higher NLLAA values than healthy banks.
3. **NPLTA** – *N*on-*P*erforming *L*oans divided by *T*otal *A*ssets. Non-performing loans consist of loans that have past their due dates by 90 days and non-accrual loans. Failing banks usually have higher values of NPLTA than healthy banks.
4. **NLLTL** – *N*et *L*oan *L*osses divided by *T*otal *L*oans. Failing banks have higher loan losses as they often make loans to high-risk borrowers. Thus, failing banks usually have higher values of NLLTL than healthy banks.

5. **NLLPLLNI** – Sum of *N*et *L*oan *L*osses and *P*rovision for *L*oan *L*osses divided by *N*et *I*ncome. The higher the NLLPLLNI value, the poorer the bank performance.

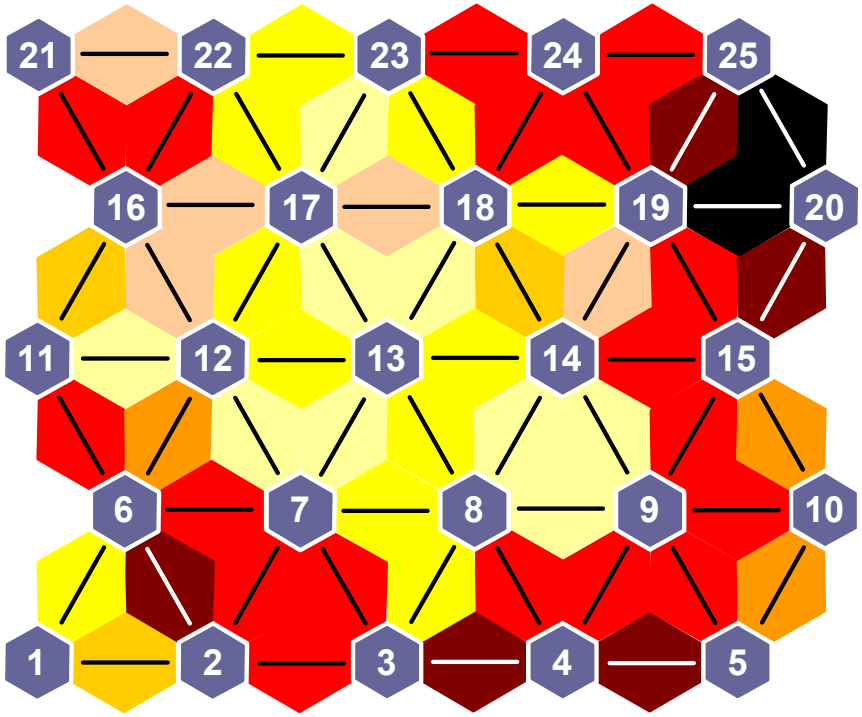
Preliminary investigations of the statistical data can reveal that a number of banks may experience some financial difficulties. Clustering should help us to identify groups of banks with similar problems.

The self-organising map (SOM) structure

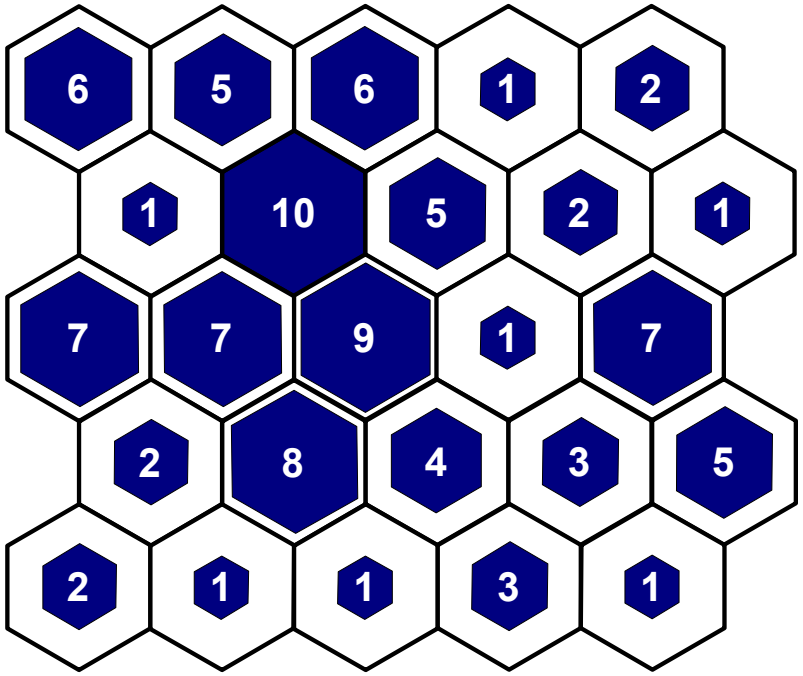


- The input data are normalised to be between 0 and 1.
- The network is trained for 10,000 iterations with a learning rate of 0.1.
- After training is complete, the SOM forms a semantic map where similar input vectors are mapped close together while dissimilar apart.
- Similar input vectors tend to excite either the same neuron or neurons closely located to each other in the Kohonen layer. This SOM property is visualised using the weight distance matrix, also known as the *U-matrix*.

The 5-by 5 SOM after training: (*a*) the U-matrix; (*b*) the sample hit plot



(a)



(b)

What do these clusters actually mean?

- Unlike classification where the number of classes is decided beforehand, in SOM-based clustering the number of clusters is unknown, and assigning a label or interpretation to each cluster requires some prior knowledge and domain expertise.
- The centre of a cluster often reveals features that separate one cluster from another. Therefore, determining the average member of a cluster should enable us to interpret the meaning of the entire cluster.

Clustering results of the 5-by-5 SOM

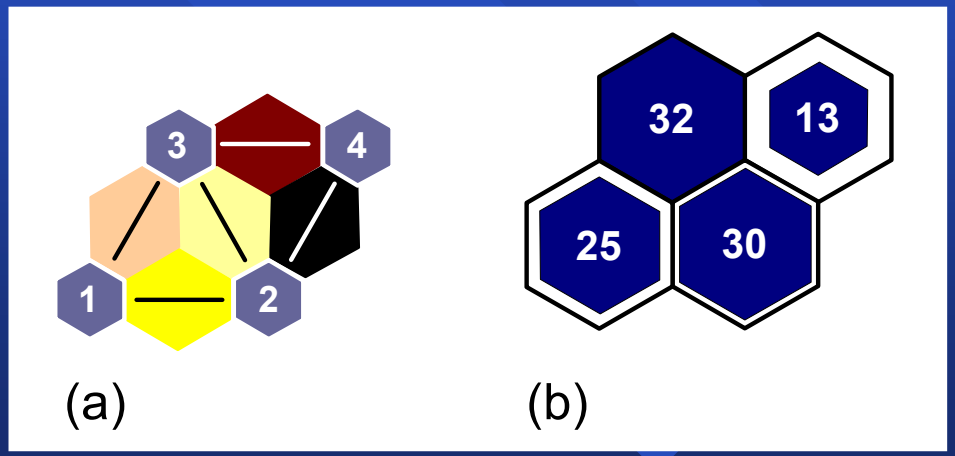
Cluster	Size	Neuron number	Financial profile of the cluster														
			NITA			NLLAA			NPLTA			NLLTL			NLLPLLNI		
			Mean	Median	STD	Mean	Median	STD	Mean	Median	STD	Mean	Median	STD	Mean	Median	STD
A	4	1 6	0.0369	0.0369	0.0043	-0.1793	-0.1340	0.2516	0.0125	0.0100	0.0055	0.0050	0.0057	0.0036	0.2839	0.2839	0.0164
B	1	2	0.0121	0.0121	0	-0.4954	-0.4954	0	0.0323	0.0323	0	0.0006	0.0006	0	1.1522	1.1522	0
C	75	3 7 8 9 11 12 13 14 16 17 18 19 21 22 23	0.0101	0.0094	0.0097	-0.0899	-0.0701	0.1646	0.0153	0.0144	0.0102	0.0143	0.0121	0.0093	0.8399	0.6973	0.7252
D	3	4	0.0066	0.0041	0.0064	0.4448	0.4528	0.0672	0.0190	0.0185	0.0058	0.0133	0.0145	0.0068	0.1894	0.1676	0.1617
E	13	5 10 15	-0.0006	-0.0010	0.0044	0.0363	0.0357	0.0257	0.0205	0.0166	0.0144	0.0388	0.0376	0.0108	8.0965	7.1786	3.9200
F	1	20	-0.0092	-0.0092	0	0.0089	0.0089	0	0.0215	0.0215	0	0.0055	0.0055	0	9.4091	9.4091	0
G	1	24	-0.0060	-0.0060	0	0.0199	0.0199	0	0.0198	0.0198	0	0.0662	0.0662	0	0.3612	0.3612	0
H	2	25	0.0014	0.0015	0.0019	0.0225	0.0225	0.0048	0.0164	0.0164	0.0029	0.0740	0.0740	0.0052	10.9785	10.9785	1.2720

An important part of cluster analysis is to identify outliers – objects that do not naturally fall into any larger cluster – Cluster B, Cluster F and Cluster G. While conventional clustering algorithms, such as K-means clustering, do not handle outliers well, a SOM can easily identify them.

How many clusters are required?

When a clustering algorithm attempts to create larger clusters, outliers are often forced into these clusters. This may result not only in poorer clustering but, even worse, in failing to distinguish unique objects.

The 2-by 2 SOM after training: (*a*) the U-matrix; (*b*) the sample hit plot



How can we test the performance of a SOM?

- We need a test set. From the FDIC Annual Report we obtain a list of failed banks, and collect appropriate financial statement data.
- To test the SOM performance, we select 10 banks that failed last year, and collect their one-year-prior financial statement data.

Financial profile of the failing banks

NITA			NLLAA			NPLTA			NLLTL			NLLPLLNI		
Mean	Median	STD	Mean	Median	STD	Mean	Median	STD	Mean	Median	STD	Mean	Median	STD
-0.0625	-0.0616	0.0085	0.0642	0.0610	0.0234	0.0261	0.0273	0.0065	0.0341	0.0339	0.0092	7.3467	6.9641	3.8461

- Now we can apply 10 input vectors to see the SOM response.
- As expected, in the 2-by-2 SOM, all 10 input vectors are attracted by neuron 4.
- In the 5-by-5 SOM, the situation is more complicated. Six input vectors are attracted by neuron 5, two by neuron 10, one by neuron 20 and two by neuron 24.
- Thus, in both cases, failing banks are clustered correctly.

A word of caution…

- Although a SOM is a powerful clustering tool, the exact meaning of each cluster is not always clear, and a domain expert is usually needed to interpret the results.
- Also a SOM is a neural network, and any neural network is only as good as the data that goes into it. In this case study, we have used only five financial variables.
- However, to identify problem banks well in advance of their failure, we might need many more variables that hold additional information about bank performance (researchers in the industry use up 29 financial variables based on the CAMELS rating system).