Maths Skills (MTCS) G5071

Lecture 5

Kingsley Sage
Room 5C16, Pevensey III
khs20@sussex.ac.uk

Lecture 5

- Introduction to matrices and vectors
  - So what is “the matrix”?
  - Neo, Trinity and Agent Smith aside, a matrix or a vector is simply a way of representing a structured collection of numbers.
  - Vectors (1D) can be used to represent sound samples, arrays and lists of things, datasets …
  - Matrices (2D) can be used to represent images, datasets, maths problems such as sets of simultaneous equations (remember them?) and the parameters of Artificial Neural Networks ANNs …

Matrices and ANNs

- For example, a linear unit and a network layer were written:

$$y = \sum_{i=1}^{N} w_i x_i \quad \text{and} \quad y_j = \sum_{i=1}^{N} w_{ji} x_i$$

- These can be re-written to represent each set of quantities by a single symbol – each such collection is called a matrix or a vector …

Matrices and ANNs

- For example, the bold character ***y*** stands for y_1,y_2 up to y_N, all put together into a single vector (1D entity)
- ***w*** stands for all weights put together into a matrix (2D entity)
- ***x*** stands for all inputs put together into a vector (1D entity)
- That is, we can write ***y=wx*** to mean exactly the same thing as

$$y = \sum_{i=1}^{N} w_i x_i \quad \text{and} \quad y_j = \sum_{i=1}^{N} w_{ji} x_i$$

Matrix-Vector multiplication

- The Σ formula from the previous slide defines the operation of multiplying a matrix and a vector and here a single layer linear network provides an example application. In the case of a network with 3 output units and 4 inputs, we can draw up a table of weights like this:

`	Input 1	Input 2	Input 3	Input 4
Unit 1	3.2	2.0	-0.5	2.3
Unit 2	-0.4	6.7	1.1	-4.2
Unit 3	1.2	-2.5	0.3	-0.8

- Here, for example, -0.4 is the weight on the connection from input 1 to unit 2 for this network i.e. $w_{21} = -0.4$

Matrix-Vector multiplication

- We can summarise this process as:

$$\mathbf{w} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

- The convention for writing down the elements of a matrix in a table is the one adopted here i.e. first subscript says which row of the table an element is in, and the second subscript says which column it is in.

Doing the calculation

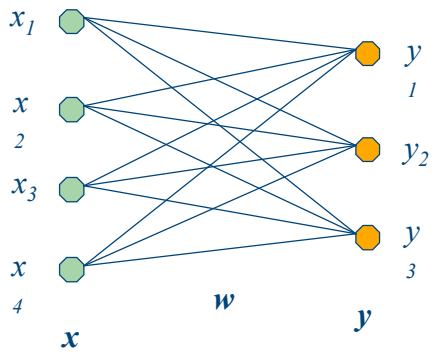
- We can write out the matrix calculation:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

- The rule for working out y_i is to take the elements of the *i'th* row of ***w*** and multiply each one by the corresponding element of the only column of ***x***, and add them up.

Doing the calculation

- You can also represent this calculation graphically:



Doing the calculation

- You should be able to draw and understand the diagram for the 3-unit, 4-input network, write the weights from the previous table in the right places, invent some input values and work out some output values.
- You should get the same result from using the diagram, using the Σ formula or MATLAB.
- Next session we shall be using MATLAB to do this.

Matrix-matrix multiplication

- Suppose the single layer network can be applied to a lot of different input vectors, and we want to specify which ones we are dealing with:

$$y_{jk} = \sum_{i=1}^{N} w_{yi} x_{ik}$$

- The subscript *k* is used to specify which of the set of inputs we are referring to. Matrix notation handles this extension very easily – objects ***x*** and ***y*** must now be proper matrices rather than vectors

Matrix-matrix multiplication

$$y_{jk} = \sum_{i=1}^{N} w_{yi} x_{ik}$$

- The previous equation defined matrix-matrix multiplication – we can still write ***y=wx.***
- Each column of the ***x*** matrix refers to a different input example, each row of ***x*** refers to a different input unit *i* in the network.
- Each column of ***y*** refers to an output from a particular input example, each row refers to a different output unit in the network.
- Assembling the output matrix involves separate calculations using each column of the input matrix as a vector.

Some simple matrix operations

- *Matrix addition* just means adding each element in one matrix to the corresponding element in another:

$$\text{If } \mathbf{z} = \mathbf{x} + \mathbf{y} \text{ then } z_{ij} = x_{ij} + y_{ij}$$

- *Matrix subtraction* is similar and *scalar multiplication* of a matrix just means multiplying each element by the same number:

$$\text{If } \mathbf{z} = k\mathbf{x}, \text{ then } z_{ij} = kx_{ij} \text{ where } k \text{ is an ordinary scalar number}$$

- Can you think of applications for these when matrices are used to represent images?

Some simple matrix operations

- Sometimes it's useful to swap the order of the subscripts in a matrix called *transposing* it, and another important operation is *inverse*:

$$\mathbf{x} = \mathbf{w}^{-1}\mathbf{y}$$

- This is used to solve the problem – when computation is possible – to find the inputs given the outputs but if there are more inputs than outputs then the system is underdetermined, and if there are more outputs than inputs then the system is overconstrained.

Significance of the matrix inverse

- We can use the matrix inverse to find a solution for a set of linear simultaneous equations:

$$2x_1 + 3x_2 = 13$$
$$3x_1 + 4x_2 = 18$$

- We can write this problem as:

$$\begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 13 \\ 18 \end{bmatrix}$$

Significance of the matrix inverse

- We can now multiply each side by the matrix inverse of the square matrix:

$$\begin{bmatrix}2 & 3\\ 3 & 4\end{bmatrix}^{-1}\begin{bmatrix}2 & 3\\ 3 & 4\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2 & 3\\ 3 & 4\end{bmatrix}^{-1}\begin{bmatrix}13\\ 18\end{bmatrix}$$

- Any matrix multiplied by itself gives a *unity* matrix:

$$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2 & 3\\ 3 & 4\end{bmatrix}^{-1}\begin{bmatrix}13\\ 18\end{bmatrix}$$

Significance of the matrix inverse

- We can now multiply each side by the matrix inverse of the square matrix:

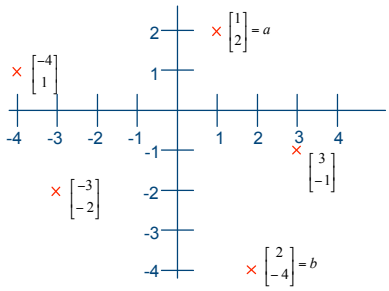
$$\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2 & 3\\ 3 & 4\end{bmatrix}^{-1}\begin{bmatrix}13\\ 18\end{bmatrix}$$

$$\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2\\ 3\end{bmatrix}$$

Checking …
(2*2) + (3*3) = 13 yes!!
(3*2) + (4*3) = 18 yes!!

Position vectors

- The idea of representing a point in space using co-ordinates *x,y* is probably familiar, for example:



- It is straightforward to extend this idea to 3 dimensions with *x,y,z*

Position vectors

- Adding and subtracting vectors is also quite straightforward:

$$\mathbf{a} + \mathbf{b} = \begin{bmatrix}1\\2\end{bmatrix} + \begin{bmatrix}2\\-4\end{bmatrix} = \begin{bmatrix}3\\-2\end{bmatrix}$$

- A classical application of this is in navigation.

Lengths and distances of vectors

- How far is the point indicated by the vector ***a*** from the origin?
- Applying Pythagoras' Theorem to the triangle gives the length of the hypotenuse (the sloping side) the value of √5.
- In general, for 2 dimensions the length is:

$$\sqrt{a_1^2 + a_2^2}$$

- For multiple dimensions, the length generalises to:

$$\sqrt{\sum_{i=1}^{N} y_i^2}$$

Lengths and distances of vectors

- So, how far apart are the points ***a*** and *b* in the original position vector diagram?

$$\sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}$$

- Note that the components are subtracted first, then the differences are squared.
- The general formula for the distance between *x* and *y* is just ||***x-y***|| and should remind of the sum of squares calculation that we used to calculate our network error *E* for the ANN learning rules

Multiplying a vector by a matrix

- Multiplying a by a matrix that only has non-zero elements on its top left to bottom right diagonal called a diagonal matrix …

$$\begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} pa_1 \\ qa_2 \end{bmatrix}$$

- What does this do to a? For a start, putting p=q is the same as multiplying by a scalar so this just stretches a out by a factor p.
- The numbers p and q are sometimes called expansion factors, and can squeeze up or stretch out more along one axis or another.

Multiplying a vector by a matrix

- Multiplying our earlier vectors by the following specific matrix:

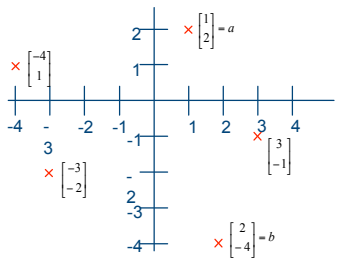
$$\begin{bmatrix} 0.7 & -0.7 \\ 0.7 & 0.7 \end{bmatrix}$$

$(-4,1) \rightarrow (-3.5,-2.1)$

$(-3,-2) \rightarrow (-0.7,-3.5)$

$(3,-1) \rightarrow (2.8,1.4)$

$(2,-4) \rightarrow (4.2,-1.4)$



- This application gives approximately a rotation – lots of uses for matrices such as these in computer graphics and video games.

Next time …

- More matrices and vectors
  - MATLAB notation
  - MATLAB matrix functions