# Matrices, Least Squares and Artificial Intelligence

Extra file for Lecture 7

**Abstract**

These lecture notes provide the mathematical background needed to work through A-exercises A1–A6. The central idea is that mathematical models can be built from a small number of basic ingredients: matrices describe and transform information, linear systems describe relationships between unknown quantities, least squares estimates model parameters from measured data, and nonlinear activation functions allow neural networks to represent more complicated relationships. The notes follow the terminology and structure of the A-exercises and are intended as preparation rather than as worked solutions.

## 1 From mathematical data to models

The A-exercises connect three closely related ideas:

1. matrices and linear algebra;
2. parameter estimation from experimental data;
3. the mathematical structure of artificial intelligence models.

The same mathematical language appears throughout. A vector collects quantities into an ordered object, a matrix acts on vectors, and a model describes how inputs are transformed into outputs.

A simple linear model has the form

$$y = ax + b.$$

Here $x$ is the input, $y$ is the output, and $a, b$ are parameters of the model. In a data-analysis setting, $a$ and $b$ are not necessarily known. They must be estimated from observations.

A neural network extends this idea. In the notation used in the exercises,

$$\hat{y} = W_2\phi(W_1x + b_1) + b_2.$$

This expression contains the same basic ingredients: matrices, vectors, parameters, and a rule that transforms the result.

**Main message.** Artificial intelligence is built on mathematical foundations. Matrices describe information, least squares learns parameters, and activation functions create nonlinear models.

# 2 Matrices

## 2.1 What is a matrix?

A matrix is a rectangular collection of numbers. For example,

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}.$$

This is a $2 \times 2$ matrix: it has two rows and two columns.

More generally,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

is an $m \times n$ matrix.

The entry in row $i$ and column $j$ is denoted by $a_{ij}$.

Matrices are useful because they allow many quantities and many equations to be handled simultaneously. In applications, a matrix may contain measured data, coefficients of equations, or parameters of a model.

## 2.2 Matrix addition

Two matrices can be added when they have the same dimensions. Addition is performed entry by entry:

$$A + B = (a_{ij} + b_{ij}).$$

For example, with

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix},$$

we obtain

$$A + B = \begin{pmatrix} 1+2 & 2+0 \\ 3+1 & 4+3 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 4 & 7 \end{pmatrix}.$$

## 2.3 Matrix multiplication

Matrix multiplication is different from entry-by-entry multiplication. If $A$ is an $m \times n$ matrix and $B$ is an $n \times p$ matrix, then $AB$ is an $m \times p$ matrix.

The entry in row $i$ and column $j$ is

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

For two $2 \times 2$ matrices,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{pmatrix}.$$

Using the matrices above,

$$AB = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 10 & 12 \end{pmatrix}.$$

Reversing the order gives

$$BA = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 4 \\ 10 & 14 \end{pmatrix}.$$

Thus, in general,

$$AB \neq BA.$$

**Important.** Matrix multiplication is generally *not commutative*. The order of the factors matters.

$$AB \neq BA \quad \text{in general.}$$

### 2.4 Matrices acting on vectors

Let

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Then

$$Ax = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{pmatrix}.$$

This operation is fundamental in data analysis and artificial intelligence: a matrix can transform a vector of inputs into a vector of outputs.

## 3 Linear systems

### 3.1 A system of equations

Consider

$$\begin{cases} 2x + y = 5, \\ x - y = 1. \end{cases}$$

There are two equations and two unknowns. One can solve the system by ordinary algebra, but matrix notation gives a compact representation.

Define

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 5 \\ 1 \end{pmatrix}.$$

Then the system becomes

$$A\mathbf{x} = \mathbf{b}.$$

This notation is important because larger systems can be written in exactly the same way:

$$A\mathbf{x} = \mathbf{b}.$$

### 3.2 Solving the system

For the example,

$$x - y = 1 \quad \Longrightarrow \quad y = x - 1.$$

Substitution into the first equation gives

$$2x + (x - 1) = 5,$$

so

$$3x = 6, \qquad x = 2.$$

Consequently,

$$y = 1.$$

Thus

$$\mathbf{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

The matrix form does not replace the underlying equations; it organizes them in a form that can be generalized to many variables.

### 3.3 Why matrix form matters

Suppose there are $n$ unknowns and $m$ equations. We can write

$$A\mathbf{x} = \mathbf{b},$$

where

$$A \in \mathbb{R}^{m \times n}, \qquad \mathbf{x} \in \mathbb{R}^n, \qquad \mathbf{b} \in \mathbb{R}^m.$$

This notation is the starting point for many computational methods. It is also the natural language for least squares.

## 4 Least squares

### 4.1 From exact equations to measured data

In an ideal mathematical problem, a model may satisfy every equation exactly. Experimental data are different. Measurements contain deviations, and the points generally do not lie exactly on one straight line.

Suppose the model is

$$y = ax + b.$$

For measured data $(x_i, y_i)$, the model predicts

$$\hat{y}_i = ax_i + b.$$

The difference between the measured and predicted value is called the residual:

$$r_i = y_i - \hat{y}_i = y_i - (ax_i + b).$$

A natural way to find $a$ and $b$ is to make the residuals collectively small.

## 4.2 The least-squares criterion

Least squares minimizes the sum of the squared residuals:

$$S(a,b) = \sum_{i=1}^{N} [y_i - (ax_i + b)]^2 .$$

The parameters $a$ and $b$ are chosen so that

$$S(a,b)$$

is as small as possible.

The square is important because positive and negative residuals should not cancel one another. Squaring also gives a convenient mathematical optimization problem.

## 4.3 Matrix representation

For the linear model

$$y = ax + b,$$

the data equations can be written as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \approx \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}.$$

Define

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \qquad X = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} a \\ b \end{pmatrix}.$$

Then

$$X\boldsymbol{\beta} \approx \mathbf{y}.$$

The approximation symbol is essential: measured data usually do not satisfy the model exactly.

## 4.4 Normal equations

The least-squares estimate can be characterized by the normal equations

$$X^T X\boldsymbol{\beta} = X^T\mathbf{y}.$$

If $X^T X$ is invertible, this gives

$$\boxed{\boldsymbol{\beta} = (X^T X)^{-1} X^T \mathbf{y}.}$$

For the straight-line model,

$$\boldsymbol{\beta} = \begin{pmatrix} a \\ b \end{pmatrix}.$$

The formula shows why matrices are useful: a data-fitting problem becomes a matrix calculation.

### 4.5 Worked setup for the A3 data

The exercise gives

| $x$ | $y$ |
|---|---|
| 1 | 2.1 |
| 2 | 3.9 |
| 3 | 6.2 |
| 4 | 8.1 |

The design matrix is therefore

$$X = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 2.1 \\ 3.9 \\ 6.2 \\ 8.1 \end{pmatrix}.$$

The unknown parameter vector is

$$\boldsymbol{\beta} = \begin{pmatrix} a \\ b \end{pmatrix}.$$

The least-squares problem is

$$\min_{a,b} \left\| X \begin{pmatrix} a \\ b \end{pmatrix} - \mathbf{y} \right\|^2.$$

Students should recognize that this is exactly the matrix version of minimizing the sum of squared residuals.

## 5 Calibration and the Beer–Lambert model

### 5.1 Linear calibration

The A4 exercise considers

$$A = mc + b.$$

Here $A$ denotes the measured response and $c$ the concentration. The parameters $m$ and $b$ describe the calibration line.

The model has exactly the same mathematical form as

$$y = ax + b.$$

Thus,

$$y \longleftrightarrow A, \qquad x \longleftrightarrow c, \qquad a \longleftrightarrow m.$$

The intercept $b$ is the predicted response when $c = 0$, while $m$ describes the change in response per unit change in concentration.

### 5.2 Determining the calibration parameters

Suppose measurements

$$(c_1, A_1), \ldots, (c_N, A_N)$$

are available. We construct

$$X = \begin{pmatrix} c_1 & 1 \\ c_2 & 1 \\ \vdots & \vdots \\ c_N & 1 \end{pmatrix}, \qquad \mathbf{A} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} m \\ b \end{pmatrix}.$$

Then

$$X\boldsymbol{\beta} \approx \mathbf{A}$$

and least squares determines $m$ and $b$ by minimizing

$$\sum_{i=1}^{N} [A_i - (mc_i + b)]^2 .$$

Thus calibration is an application of the same least-squares procedure introduced above.

### 5.3 Predicting an unknown concentration

Once $m$ and $b$ have been determined, a new measurement $A$ can be used to estimate $c$. From

$$A = mc + b$$

we solve for $c$:

$$\boxed{c = \frac{A - b}{m}}$$

provided $m \neq 0$.

The important conceptual distinction is:

- calibration: use known concentrations to determine $m$ and $b$;
- prediction: use the fitted calibration relation to estimate an unknown concentration.

**Key idea.** Least squares turns experimental calibration data into a mathematical relationship that can subsequently be used for prediction.

## 6 From linear models to artificial intelligence

### 6.1 A linear transformation

A matrix and a vector can be combined as

$$z = Wx + b.$$

Here $W$ is a matrix of parameters, $x$ is the input vector, and $b$ is a bias vector.

This is a vector-valued generalization of the scalar equation

$$y = ax + b.$$

In one dimension,

$$W = a, \qquad x = x, \qquad b = b,$$

and therefore

$$y = ax + b.$$

This makes the connection between elementary linear models and neural networks explicit.

### 6.2 The role of the weights

The entries of $W$ are called weights. They determine how strongly the components of the input contribute to the output.

For

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \qquad W = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix},$$

we have

$$Wx = \begin{pmatrix} w_{11}x_1 + w_{12}x_2 \\ w_{21}x_1 + w_{22}x_2 \end{pmatrix}.$$

Thus matrix multiplication combines and transforms information.

### 6.3 The role of the bias

The vector $b$ shifts the result:

$$z = Wx + b.$$

The bias is analogous to the intercept in

$$y = ax + b.$$

Without the bias, the transformation is restricted to pass through the origin in the corresponding linear setting. Including a bias gives the model an additional degree of freedom.

## 7 Activation functions

### 7.1 Why a nonlinear function is needed

A sequence of purely linear transformations is still a linear transformation (up to the affine shifts represented by biases). To construct a nonlinear model, a nonlinear activation function is inserted between transformations.

The exercise uses the ReLU function:

$$\phi(z) = \max(0, z).$$

For a scalar $z$,

$$\phi(z) = \begin{cases} 0, & z < 0, \\ z, & z \geq 0. \end{cases}$$

Thus ReLU leaves positive values unchanged and replaces negative values by zero.

### 7.2 Vector-valued ReLU

If

$$z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix},$$

then ReLU is applied component by component:

$$\phi(z) = \begin{pmatrix} \max(0, z_1) \\ \max(0, z_2) \\ \max(0, z_3) \end{pmatrix}.$$

For the vector in A5,

$$z = \begin{pmatrix} -2 \\ 0.5 \\ 3 \end{pmatrix},$$

the procedure is simply

$$-2 \mapsto 0, \qquad 0.5 \mapsto 0.5, \qquad 3 \mapsto 3.$$

Hence

$$\phi(z) = \begin{pmatrix} 0 \\ 0.5 \\ 3 \end{pmatrix}.$$

### 7.3 Where the nonlinearity comes from

The function

$$\phi(z) = \max(0, z)$$

is not a single linear function valid for all $z$. Its behavior changes at $z = 0$.

For negative inputs the output is constant:

$$\phi(z) = 0.$$

For positive inputs,

$$\phi(z) = z.$$

This change in behavior is what introduces nonlinearity into the model.

## 8 Neural networks

### 8.1 One neural-network layer

The mathematical building block

$$z = Wx + b$$

is followed by an activation:

$$h = \phi(Wx + b).$$

Here:

- $x$ is the input;

- $W$ contains the weights;
- $b$ contains the biases;
- $\phi$ is the activation function;
- $h$ is the transformed output of the layer.

The parameters $W$ and $b$ are learned from data.

### 8.2 Two layers

The A6 exercise gives

$$\hat{y} = W_2\phi(W_1x + b_1) + b_2.$$

This can be read from inside to outside.

First compute

$$z_1 = W_1x + b_1.$$

Then apply the activation:

$$h = \phi(z_1).$$

Finally compute

$$\hat{y} = W_2h + b_2.$$

So the complete chain is

$$x \longrightarrow W_1x + b_1 \longrightarrow \phi(W_1x + b_1) \longrightarrow W_2\phi(W_1x + b_1) + b_2.$$

### 8.3 Interpretation of the parameters

The first weight matrix $W_1$ determines how the input is transformed in the first layer. The bias $b_1$ shifts that transformation.

The activation $\phi$ introduces nonlinearity.

The second matrix $W_2$ combines the transformed quantities to produce the final output, and $b_2$ provides the final shift.

Thus the roles requested in A6 can be summarized as follows:

| | |
|---|---|
| $W$ | weights: determine how information is combined, |
| $b$ | biases: shift the transformations, |
| $\phi$ | activation: introduces nonlinearity. |

## 9 How the concepts fit together

The A-exercises are not six unrelated mathematical questions. They form a progression.

### Step 1: Matrices

We begin with

$$A + B, \qquad AB, \qquad BA.$$

This develops the basic language for manipulating collections of numbers.

### Step 2: Linear systems

The notation

$$A\mathbf{x} = \mathbf{b}$$

shows how matrices represent systems of equations.

### Step 3: Least squares

Real data do not usually satisfy a model exactly. We therefore solve an approximate problem:

$$X\boldsymbol{\beta} \approx \mathbf{y}.$$

Least squares finds the parameters that minimize the total squared error.

### Step 4: Calibration

A physical or chemical calibration relation such as

$$A = mc + b$$

is an immediate application of linear least squares.

### Step 5: Activation functions

The transformation

$$z = Wx + b$$

is still essentially a linear/affine operation. Applying

$$\phi(z) = \max(0, z)$$

creates a nonlinear model.

### Step 6: Neural networks

Combining these ingredients gives

$$\hat{y} = W_2\phi(W_1x + b_1) + b_2.$$

Therefore the conceptual chain is

| matrices → linear systems → least squares → calibration → nonlinear models → neural networks. |
|---|

## 10 Practical checklist for the A-exercises

### A1: Matrix operations

When multiplying matrices:

1. check the dimensions;
2. remember that rows of the first matrix interact with columns of the second;
3. calculate each entry as a dot product;
4. do not assume $AB = BA$.

### A2: Linear system

To write

$$A\mathbf{x} = \mathbf{b},$$

identify:

1. the coefficients of the unknowns $\rightarrow A$;
2. the unknowns $\rightarrow \mathbf{x}$;
3. the right-hand sides $\rightarrow \mathbf{b}$.

### A3: Least squares

For

$$y = ax + b,$$

construct

$$X = \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix}.$$

Then use

$$\boldsymbol{\beta} = \begin{pmatrix} a \\ b \end{pmatrix}$$

and the least-squares equations

$$X^T X\boldsymbol{\beta} = X^T\mathbf{y}.$$

### A4: Calibration

Recognize

$$A = mc + b$$

as the same straight-line model. Explain the physical interpretation of the slope and intercept, then explain how the fitted relation can be inverted to predict an unknown concentration.

### A5: ReLU

Apply the function component by component:

$$z_i \mapsto \max(0, z_i).$$

Then explain that the function has different behavior for negative and positive input values.

### A6: Neural-network interpretation

Read

$$\hat{y} = W_2\phi(W_1x + b_1) + b_2$$

from inside to outside:

$$x \rightarrow W_1x + b_1 \rightarrow \phi(\cdot) \rightarrow W_2(\cdot) + b_2.$$

Identify weights, biases, and activation functions separately.

## 11 Summary

The mathematical foundations needed for these exercises can be summarized in a small number of formulas:

$$\boxed{A + B = (a_{ij} + b_{ij})}$$

for matrix addition,

$$\boxed{(AB)_{ij} = \sum_k a_{ik} b_{kj}}$$

for matrix multiplication,

$$\boxed{A\mathbf{x} = \mathbf{b}}$$

for a linear system,

$$\boxed{\min_{\boldsymbol{\beta}} \|X\boldsymbol{\beta} - \mathbf{y}\|^2}$$

for least squares,

$$\boxed{A = mc + b}$$

for linear calibration,

$$\boxed{\phi(z) = \max(0, z)}$$

for ReLU, and

$$\boxed{\hat{y} = W_2\phi(W_1 x + b_1) + b_2}$$

for the neural-network model used in the exercises.

The most important conceptual point is that these formulas are connected. Matrices provide the language for representing transformations and systems. Least squares provides a way of learning parameters from imperfect data. Activation functions extend linear transformations into nonlinear models. Together these ideas provide the mathematical foundation for the A-exercises and for the stated theme of modern data analysis and artificial intelligence.

**Final message**
Artificial intelligence is built on mathematical foundations.

$$\boxed{\text{Matrices} \quad \longrightarrow \quad \text{Learning parameters} \quad \longrightarrow \quad \text{Nonlinear models}}$$

The A-exercises are designed to make each part of this chain explicit.