# Chapter 3 Create basis systems.

Consider the form:

where X is the N dimension vector of observation, is K dimension vector of basis functions we created and C is the coefficient in N\*K dimension. Here one input t corresponds to N output x­i

There are four commonly used basis systems:

* Constant (range)
* Monomial (range, nbasis)
* Fourier series (range, nbasis, period)
* B-spline (range, nbasis, norder, breaks)

# Chapter 4 Build Functional Data Object (fdobj)

A functional data object could be considered as a list of curves, each of which is a function or a time series xi . They use the same basis system but with different coefficients to different functions.

A functional data object can be built by applying a coefficient matrix to the basis system created. The coefficient matrix contains K rows and N columns, in which Nth column is the coefficient vector of Nth function xi .

Remark: domain, replication, range

Arithmetic operation: sum, minus, multiplication, power, scaler product point wisely. However special on multiplication and power.

Linear Differential operator (Lfd)

Specially, harmonic acceleration

Bivariate correlation function (surface):

Where is a basis function of variation over s and over t.

# Chapter 5 Smoothing: estimate coefficient matrix.

Consider the form:

where y could be a vector which means only one curve (function), or a matrix with N column for N observations, but n rows for n values to be fit. is a n by K matrix where n is the number of t’s. c is a K dimension vector if y is a n dimension vector, or a K by N matrix for N curves.

Regression smooth: estimate coefficient c by least squares

Smooth with penalty:

Coefficient estimated by:

Choose best by GCV.

Smooth with constrains: y as predicted value, x as true value, transformation from x to y like log, exp, monomial and so on.

# Chapter 6 Descriptions of functional data

Sample mean and sample variance

Estimator of bivariate covariance function:

Function probe (weight function, eigenfunction, projection):

Probe value:

Probe value can be treated as proportional to projection of function x(t) on function . It measures the similarity between those two, and scores zero with orthogonal.

Phase-plane plot:

Plot of velocity versus acceleration, suggest some period circle pattern.

# Functional data analysis

## Functional linear model:

Notification:

scalar response: y, functional response: y(t)

scalar covariate: z, functional response: x(t)

functional coefficient (weight function):

Basic assumption: all the functional data can be expressed as finite basis expansion properly.

Functional linear model with scalar response:

We aim to find the coefficients and . For traditional linear model, to estimate the coefficients, we could find the optimal solution to the loss function which is a function of scalar coefficients. However, for the functional linear model, it is not easy to the optimal solution since the some coefficients are functional and infinite dimensional. From Ramsay, they expressed , where is a finite vector of basic functions. That is they assume coefficient functions can be expressed as finite basis expansion. Then

Functional linear model with functional response

Bivariate regression model

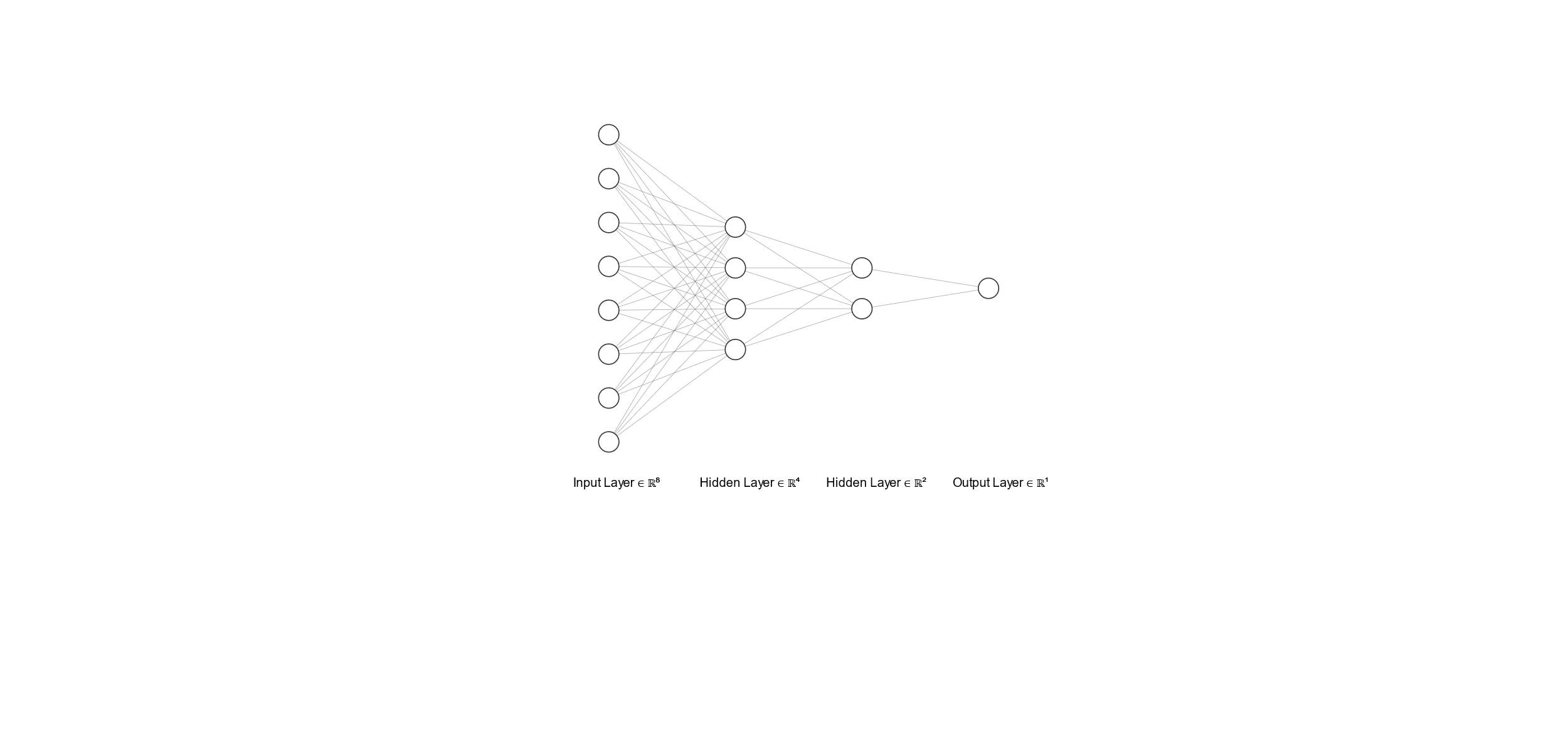
It is an auto-regression model:

## Functional neural network

There may be three ways to consider the neural network with functional input.

The first one we can come out easily is that treating all the observations of a functional data as input directly into neural network.

This kind of network is easy to understand but with some disadvantages. The first one is that the dimension may be large, however this may not be severe for a network. The second one is that it would lose some connection between observations within the same functional data, since the network would treat those data points equally unless initializing weight specially. Any permutation of the input layer may get the same result, while this should not occur.



The second network is to consider all the neuron as functional data rather than a scalar (scalar neuron can be treated as constant function). Then the following picture shows the structure. The first layer is input layer containing functional and scalar inputs. The weights are shown in blue segments which mean scalar weight. Since the linear combination of functions is still function, then the activation function shown in green should be a map from function to function which is complicated. Then the output can be certainly functional.

The last one is from FuncNN (), which does some transformation of functional data into first layer. The weight function shown in orange can be expressed as basis expansion. Then the linear combination would be summation of scalar product of coefficients of weight function and inner product of basis function and functional input, which is a function of coefficients. The later layers after first layer then are the same as the conventional one.

In this network, the dimension is less than the first network since it uses basis expansion instead of all the data points.

The dot represents inner product that is integral on an interval.