Some Applications of Functional Networks in Statistics and Engineering

Enrique Castillo, University of Cantabria

J. M. Gutiérrez, University of Cantabria

Ali S. Hadi, Cornell University

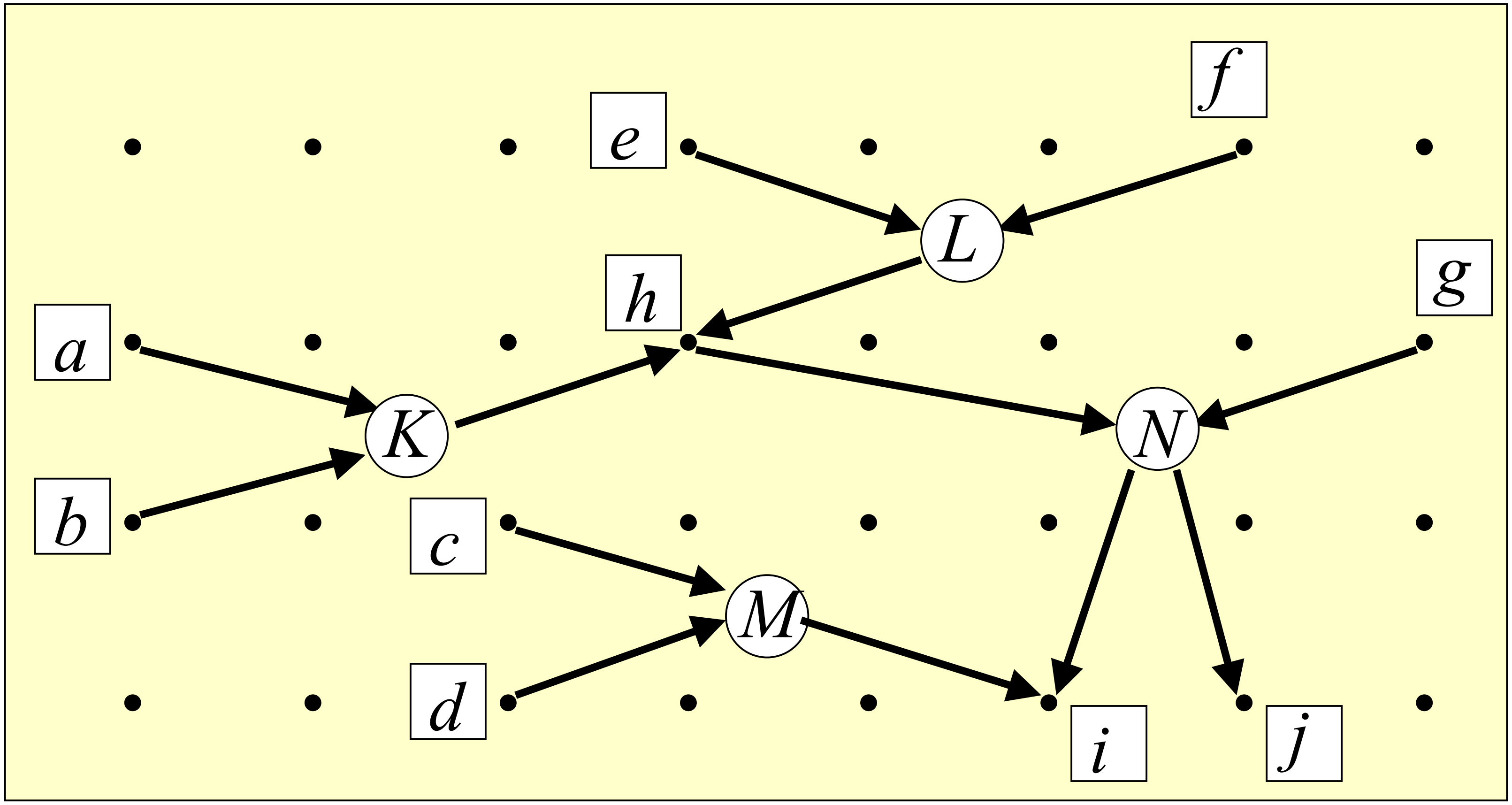
Beatriz Lacruz, University of Zaragoza

Agenda

1. A brief introduction to FNs
2. Working with FNs
3. Differences between FNs and NNs
4. Relationship to other statistical models
5. Examples of some applications of FNs
6. Summary and Conclusions

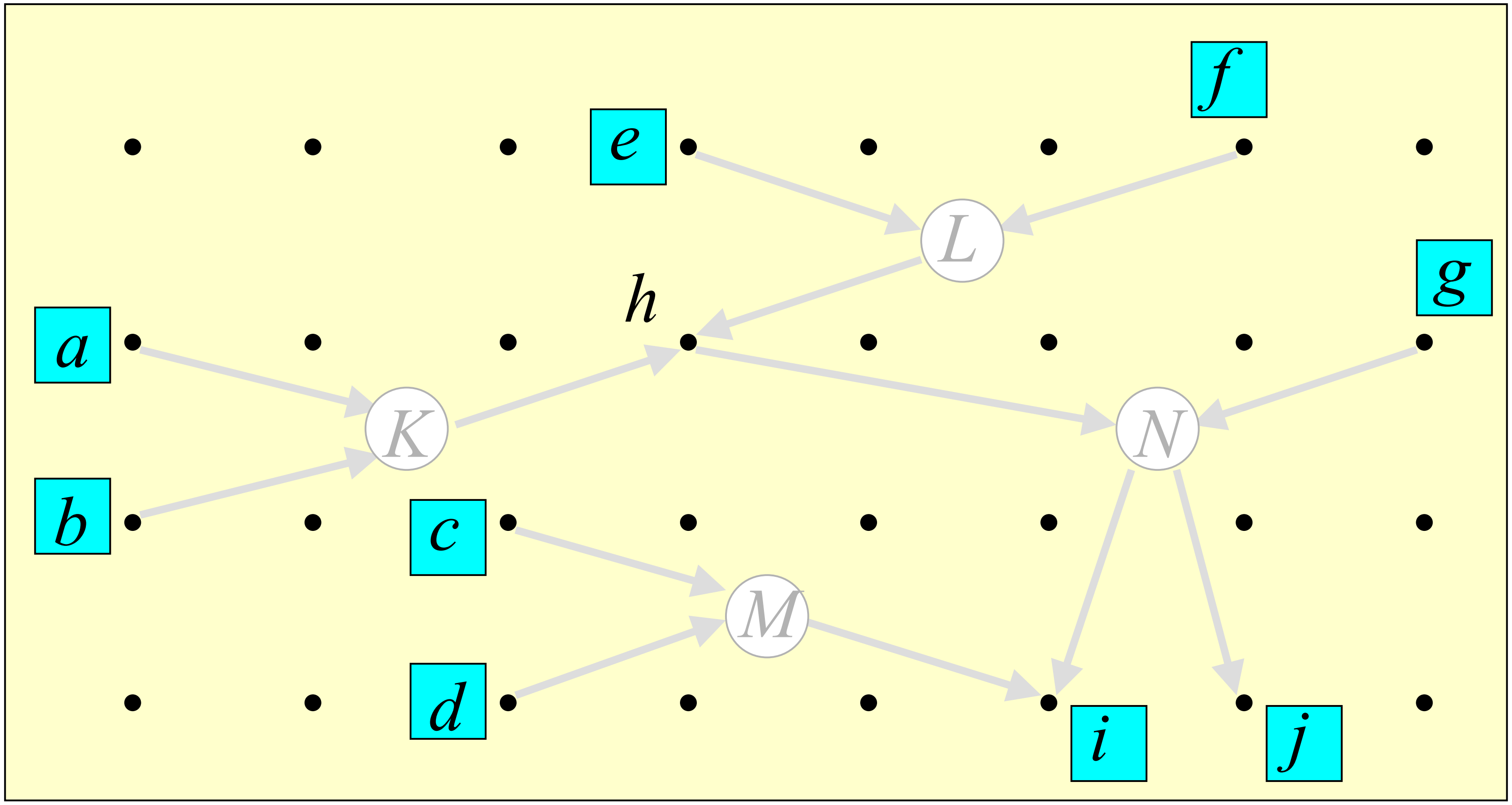
A Brief Introduction to Functional Networks (FNs)

An example of a functional network: A FN is analogous to a Printed Circuit Board (PCB)



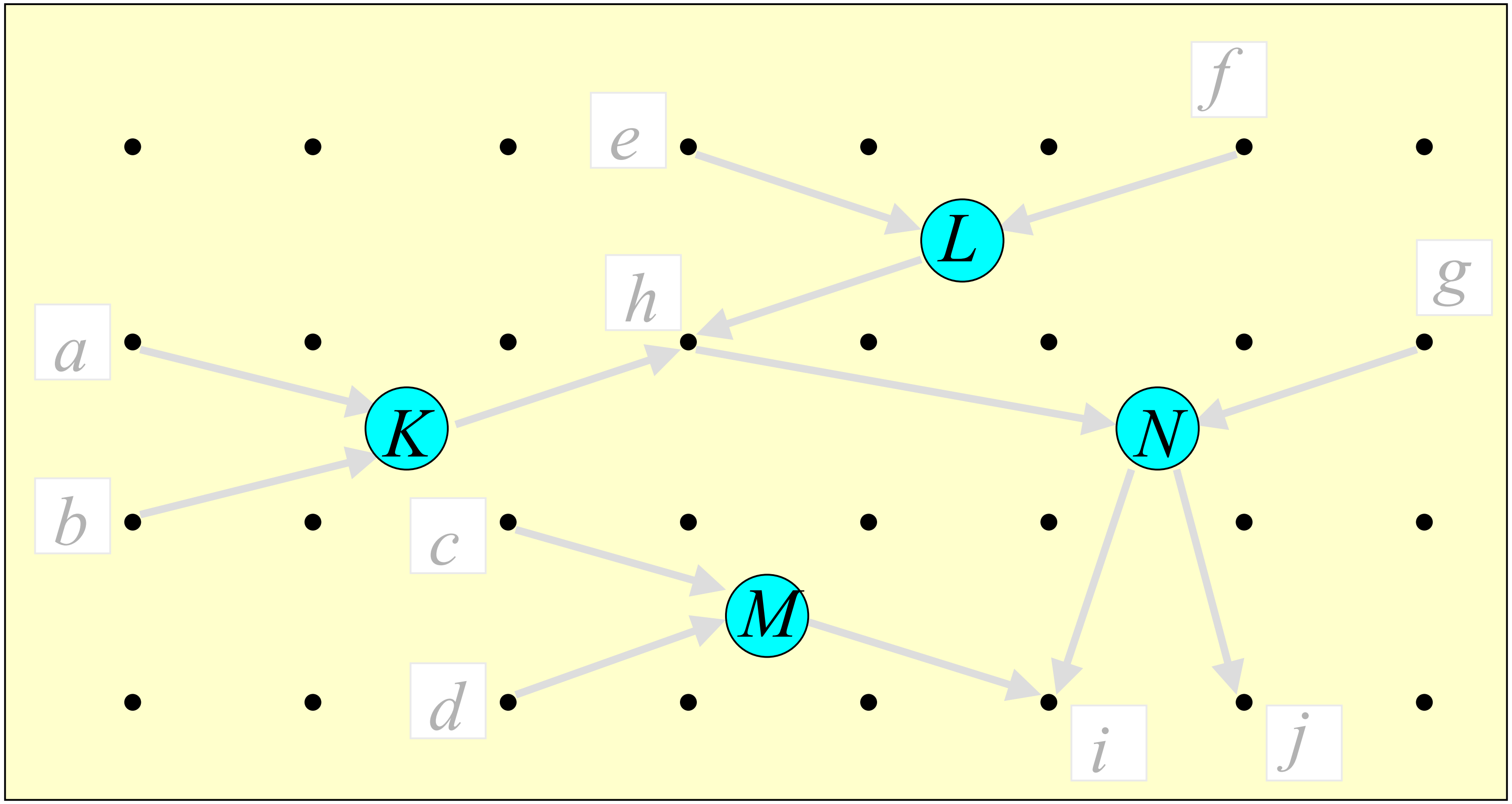
Elements of FNs:

1. Input Units: $\{a, b, c, d, e, f, g\}$



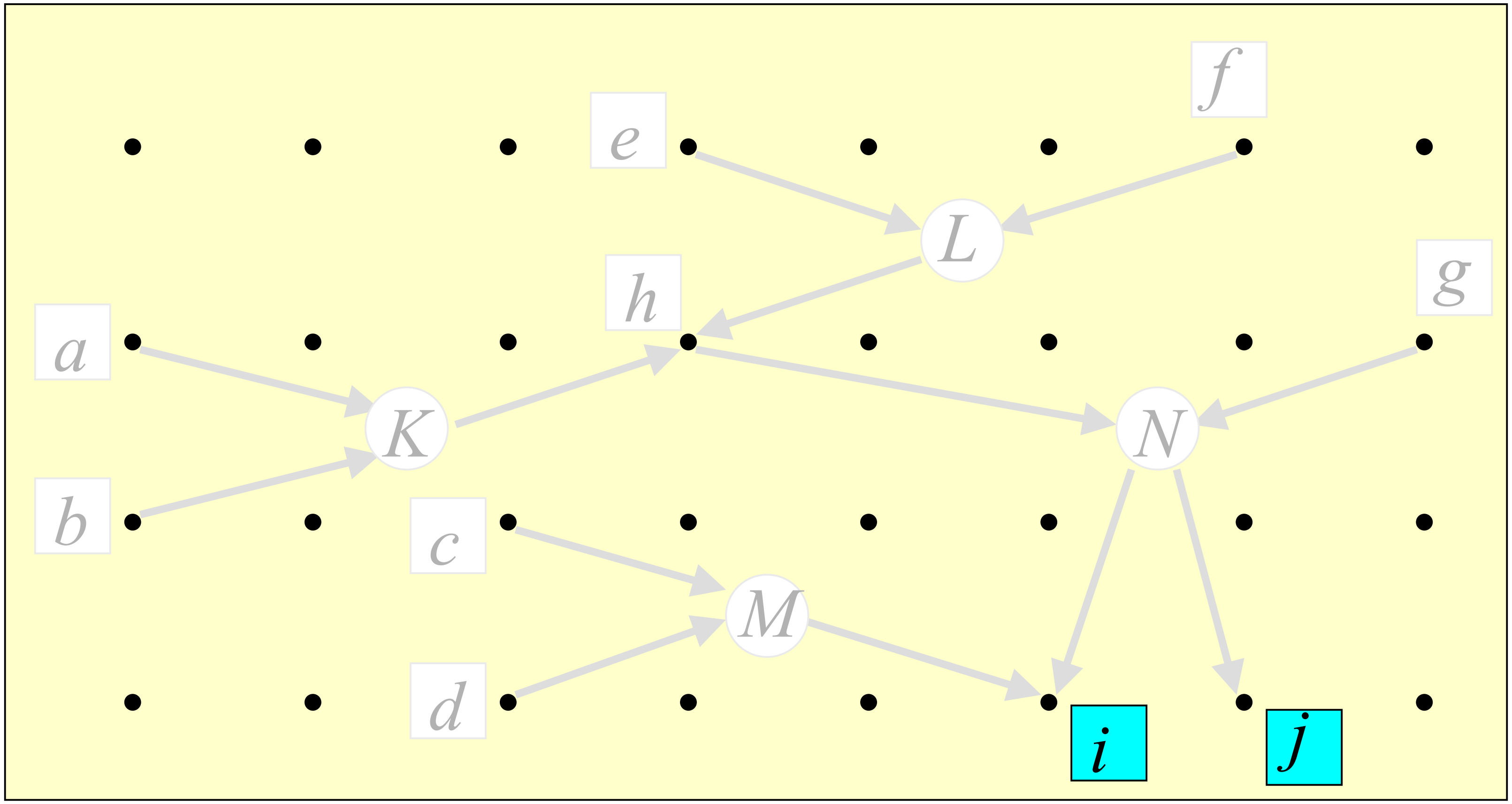
Elements of FNs:

2. Computing Neurons: $\{K, L, M, N\}$



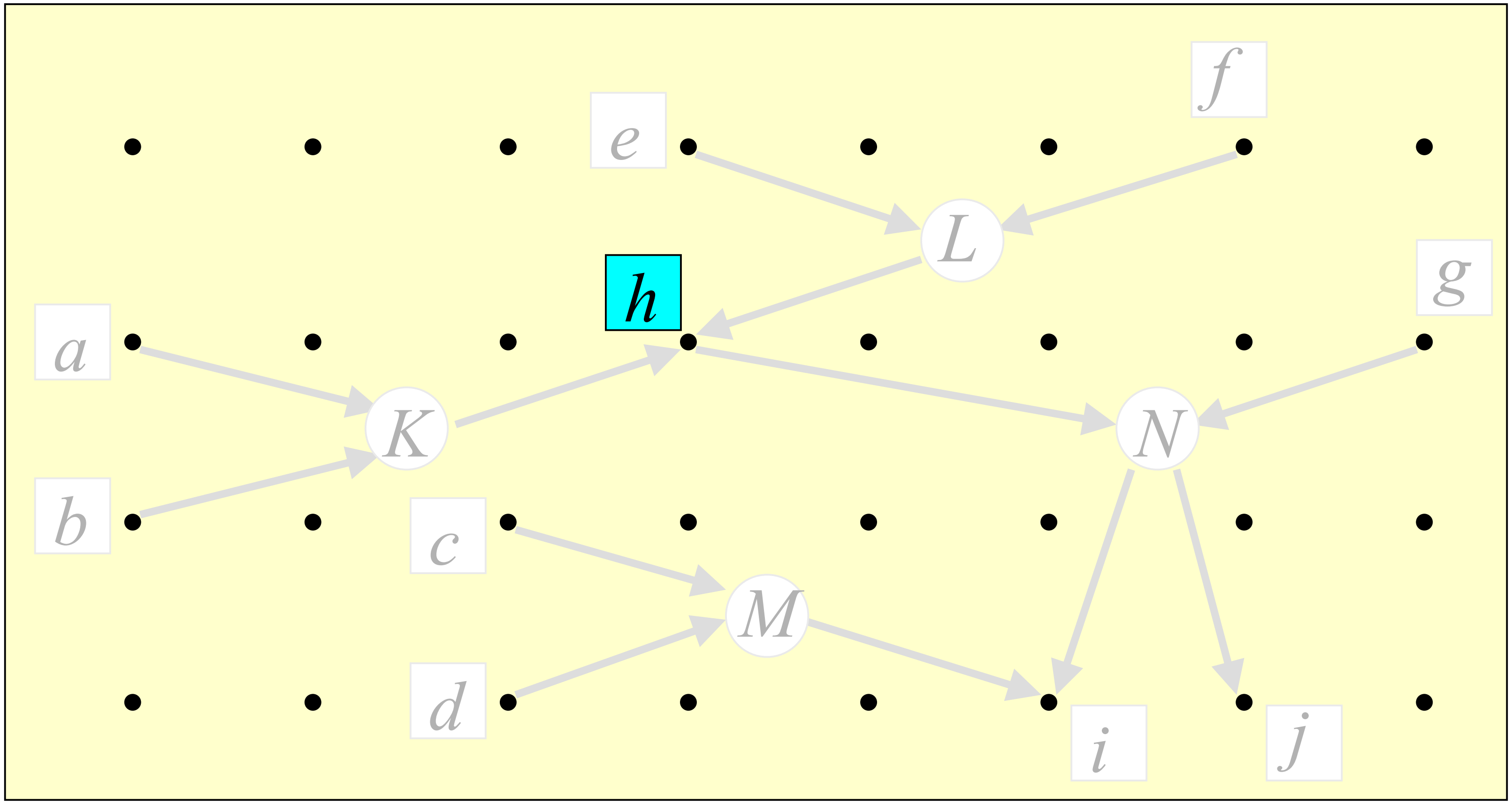
Elements of FNs:

3. Output Units: $\{i, j\}$



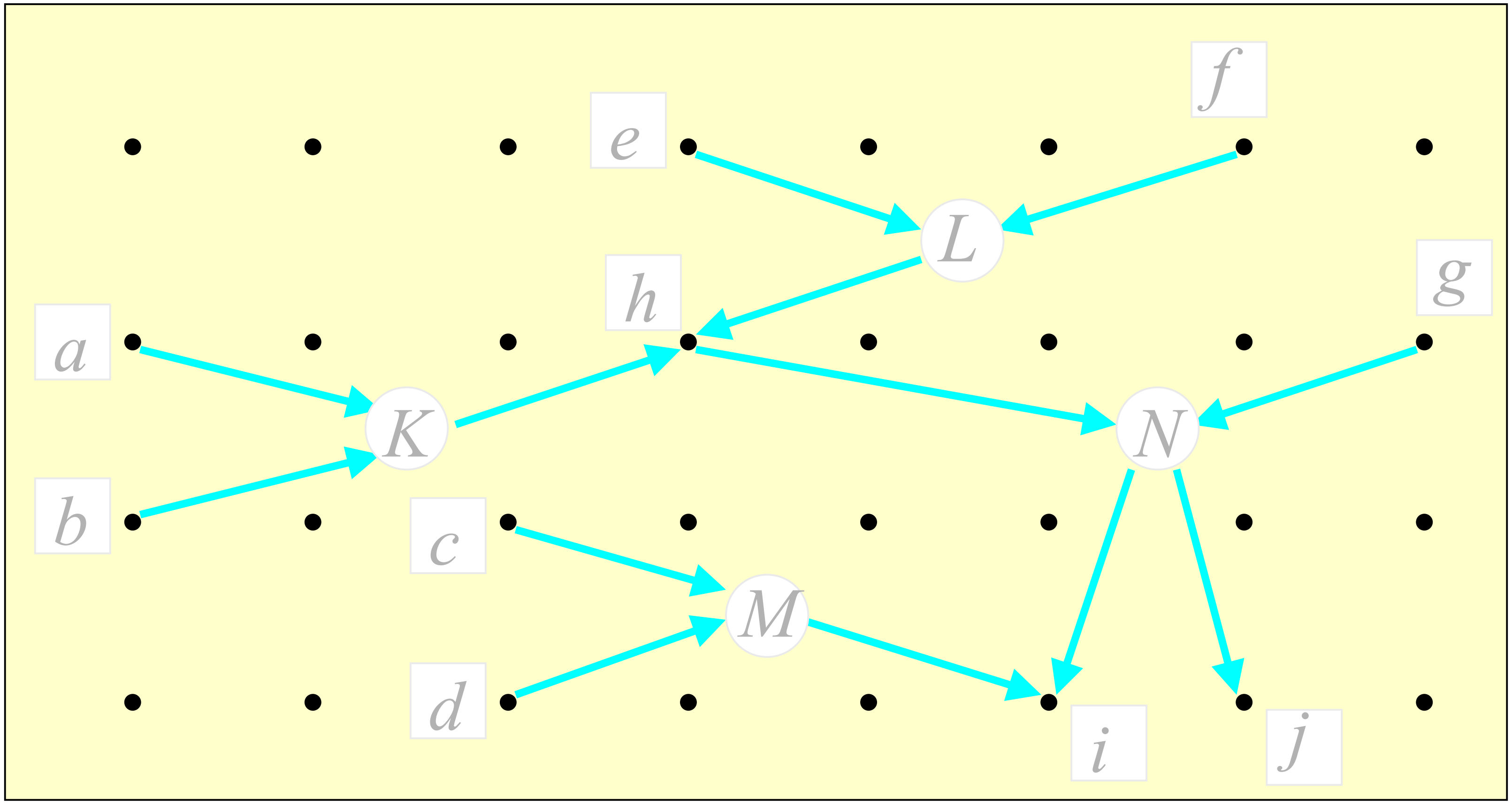
Elements of FNs:

4. Intermediate Units: $\{h\}$



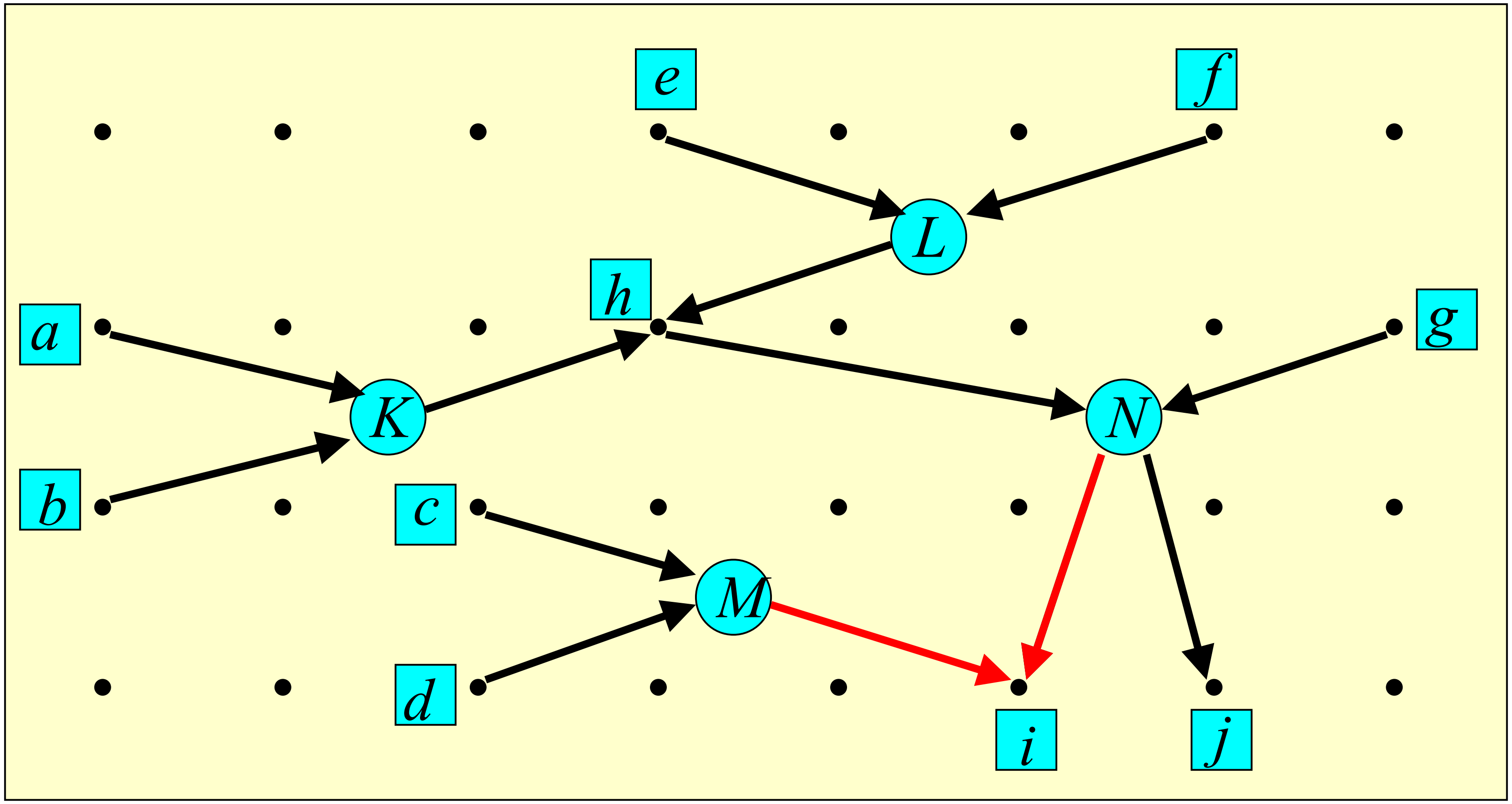
Elements of FNs:

5. Directed Links: $\{arrows\}$



Note:

N gives two outputs *i* and *j* and the *i* output of *N* must be identical to the output of *M*.



Agenda

✓1. A brief introduction to FNs
2. Working with FNs
3. Differences between FNs and NNs
4. Relationship to other statistical models
5. Examples of some applications of FNs
6. Summary and Conclusions

Working with FNs

1. Selection of the initial topology
2. Simplifying the FN
3. Uniqueness of representation
4. Parametric Learning
5. Model selection
6. Model validation

1. Selection of the Initial Topology:

Problem driven design: The selection of the initial topology of a functional network is often based on the characteristics of the problem at hand, which usually leads to a single clear network structure.

Example: Generalized Associative FNs:

Suppose the level of a disease d is a function of three symptoms: x, y and z, that is, $d = D(x, y, z)$.

We have three cases.

Case 1: We measure x and y, then z.

$$\left.\begin{array}{c} \left.\begin{array}{l} x \\ y \end{array}\right\} \; P(x,y) \\ z \end{array}\right\} \; F(P(x,y),z) = D(x,y,z)$$

Case 2: We measure y and z, then x.

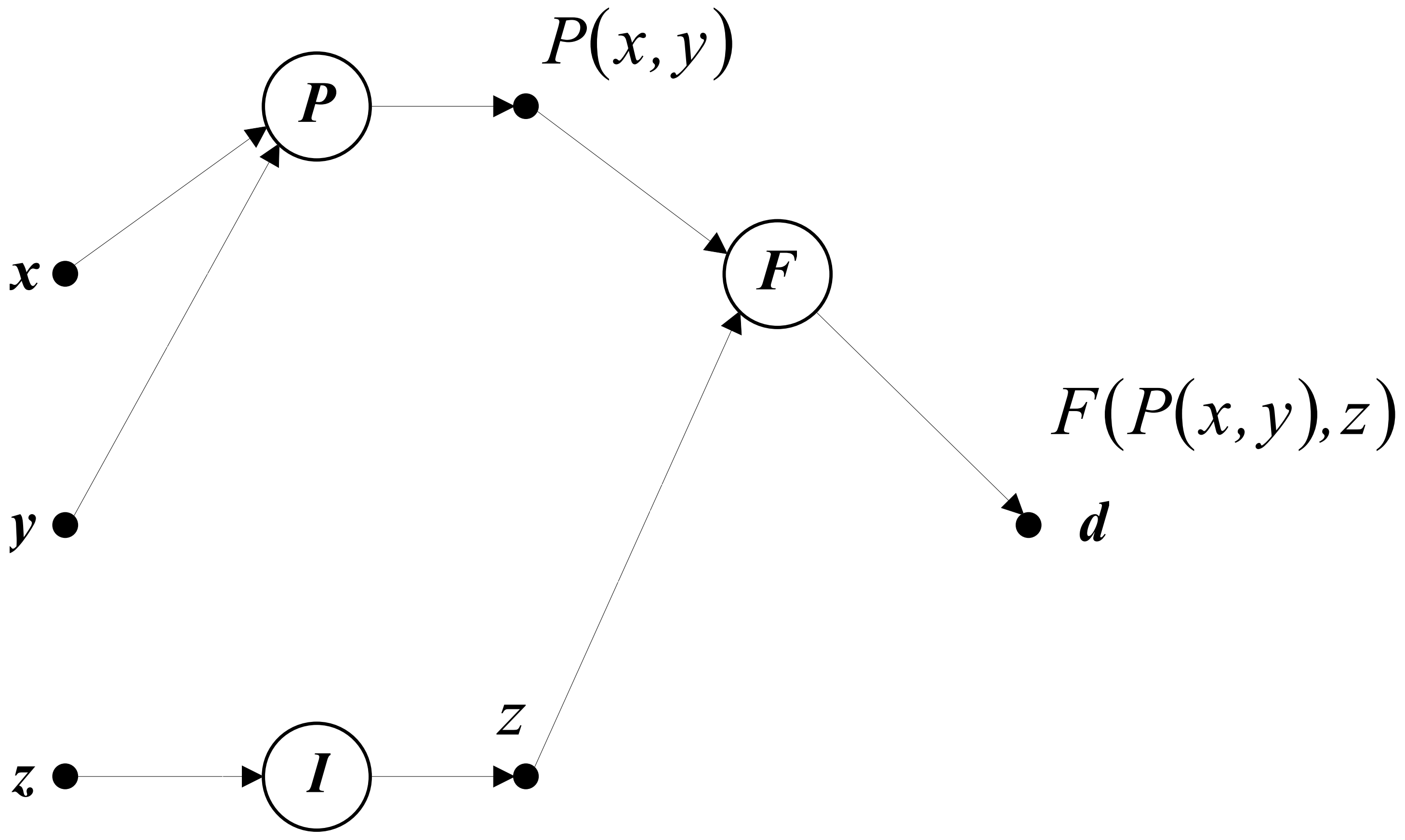
$$\left.\begin{array}{c}\left.\begin{array}{l}y\\z\end{array}\right\}\ Q(y,z)\\ x\end{array}\right\}\ G(Q(y,z),x)=D(x,y,z)$$

Case 3: We measure x and z, then y.

$$\left.\begin{array}{c}\left.\begin{array}{l}x\\z\end{array}\right\}\ R(x,z)\\ y\end{array}\right\}\ H(R(x,z),y)=D(x,y,z)$$

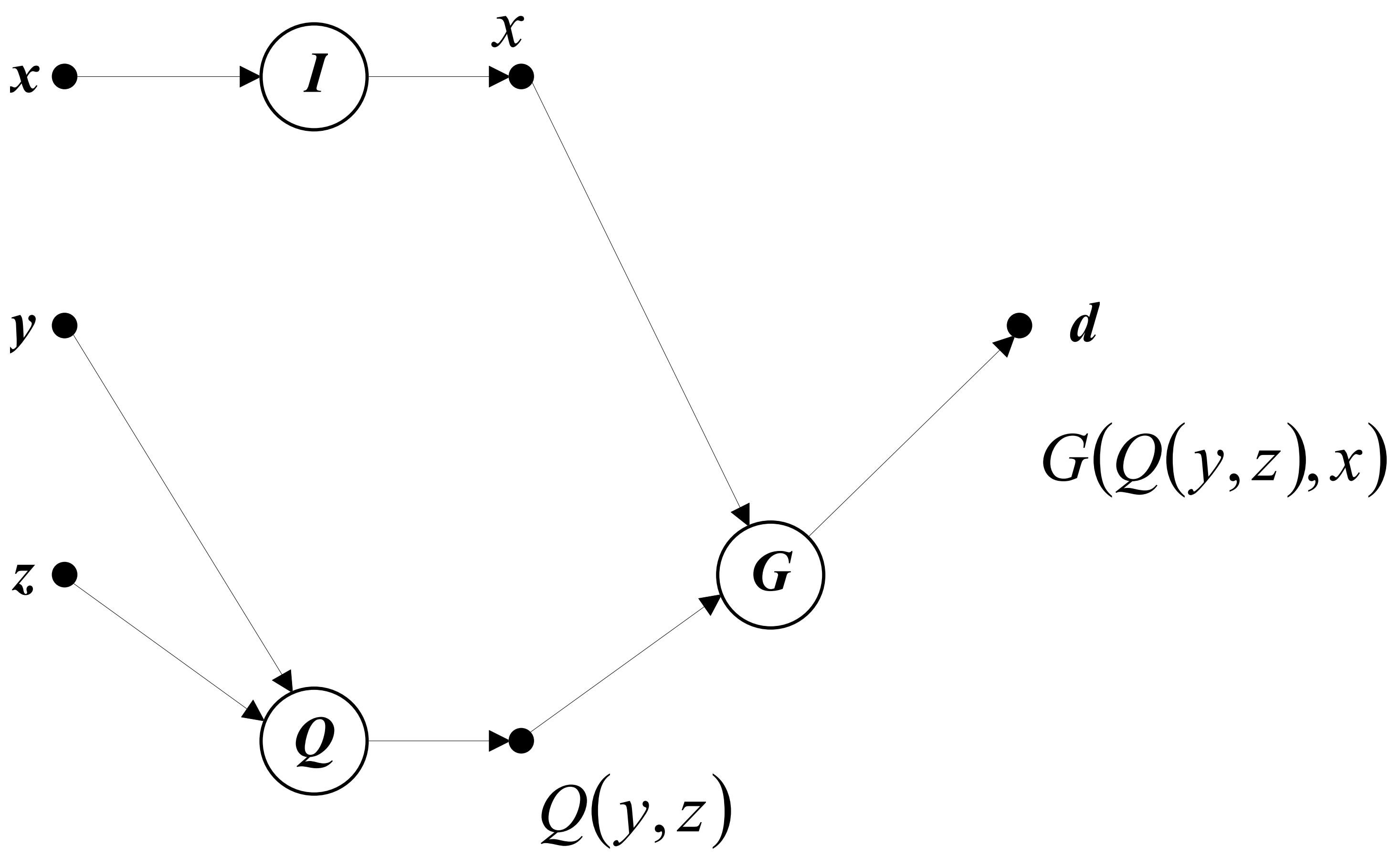
Case 1:

$$d = F(P(x,y),z)$$



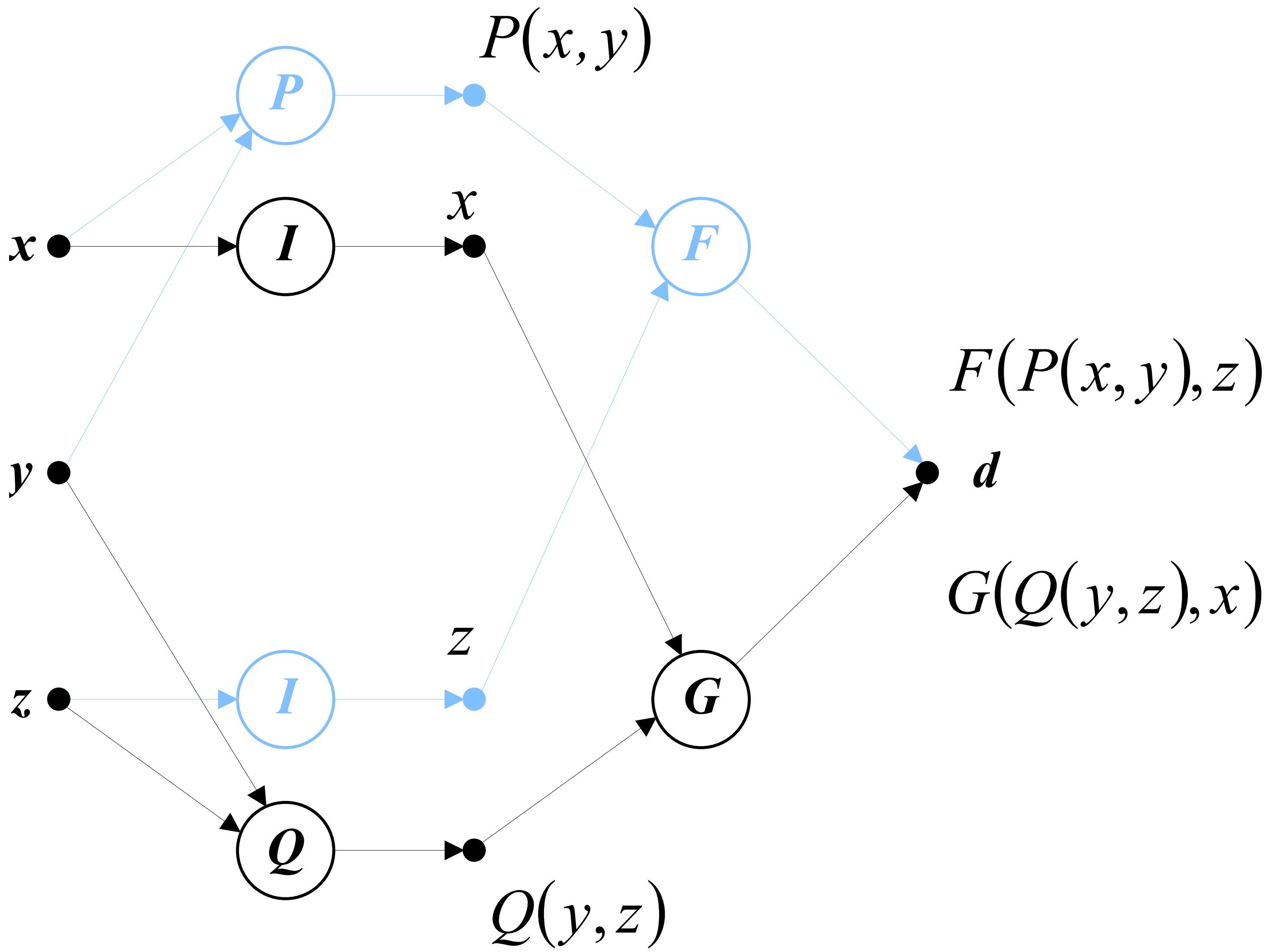
Case 2:

$$d = G(Q(y,z),x)$$



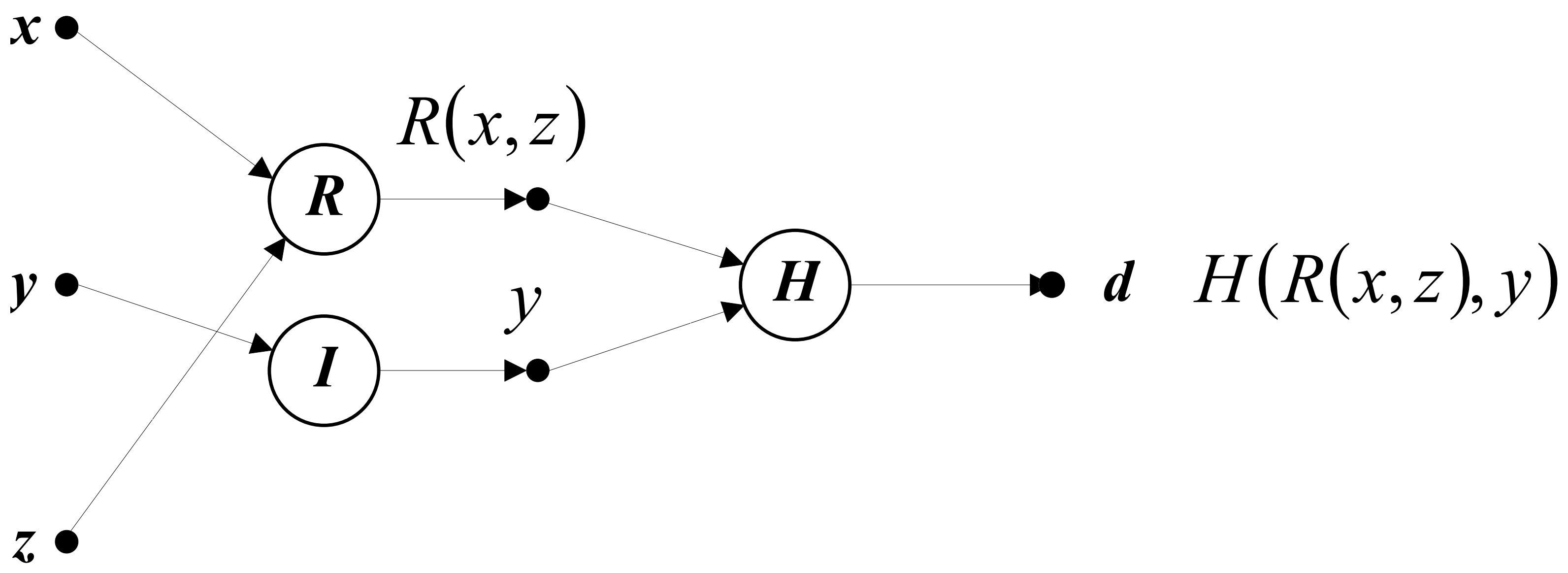
Cases 1-2:

$$d = F(P(x,y),z) = G(Q(y,z),x)$$



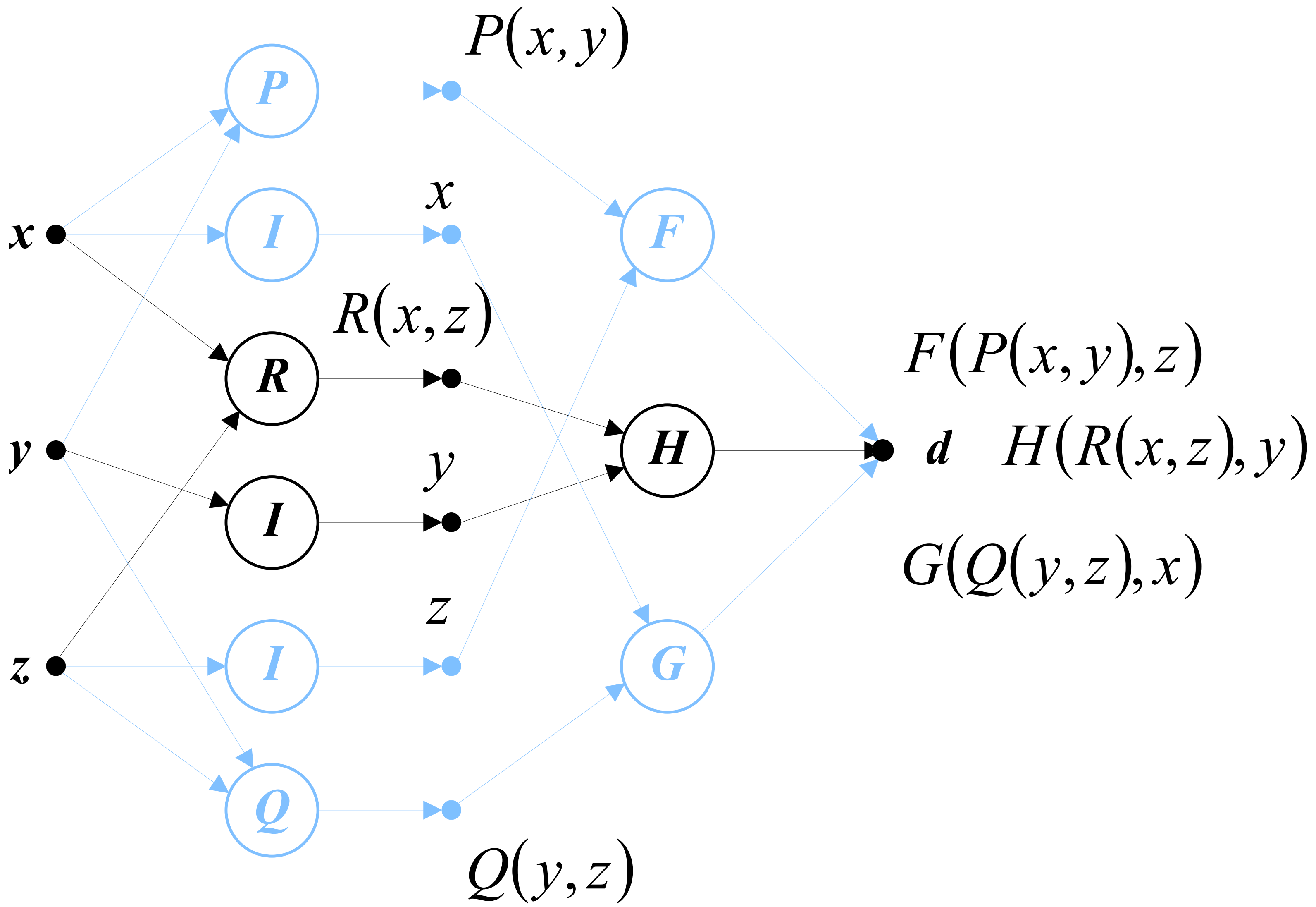
Case 3:

$$d = H\left(R(x,z), y\right)$$

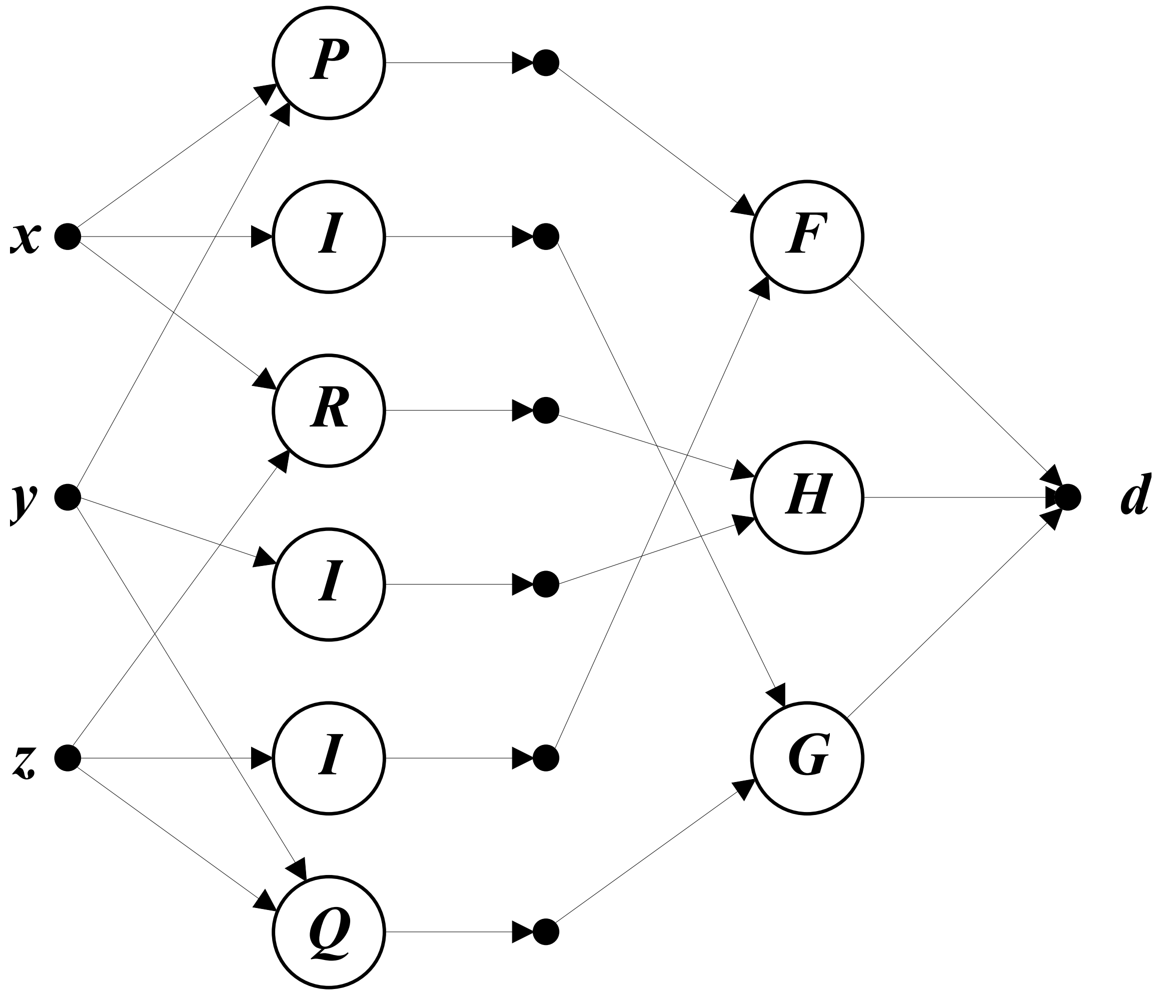


Cases 1-3:

$$d = F(P(x,y),z) = G(Q(y,z),x) = H(R(x,z),y)$$



$$d = F\big(P(x, y), z\big) = G\big(Q(y, z), x\big) = H\big(R(x, z), y\big)$$



2. Simplifying FNs:

Using functional equations: We can determine whether or not there exists another functional network giving the same output for the same input. This leads to the concept of equivalent functional networks.

Example: Generalized Associative FNs:

$$d = F(P(x,y),z) = G(Q(y,z),x) = H(R(x,z),y)$$

The general solution to these equations is:

$$F(x,y) = k[f(x) + r(y)]; \quad P(x,y) = f^{-1}[p(x) + q(y)];$$

$$G(x,y) = k[n(x) + p(y)]; \quad Q(x,y) = n^{-1}[q(x) + r(y)];$$

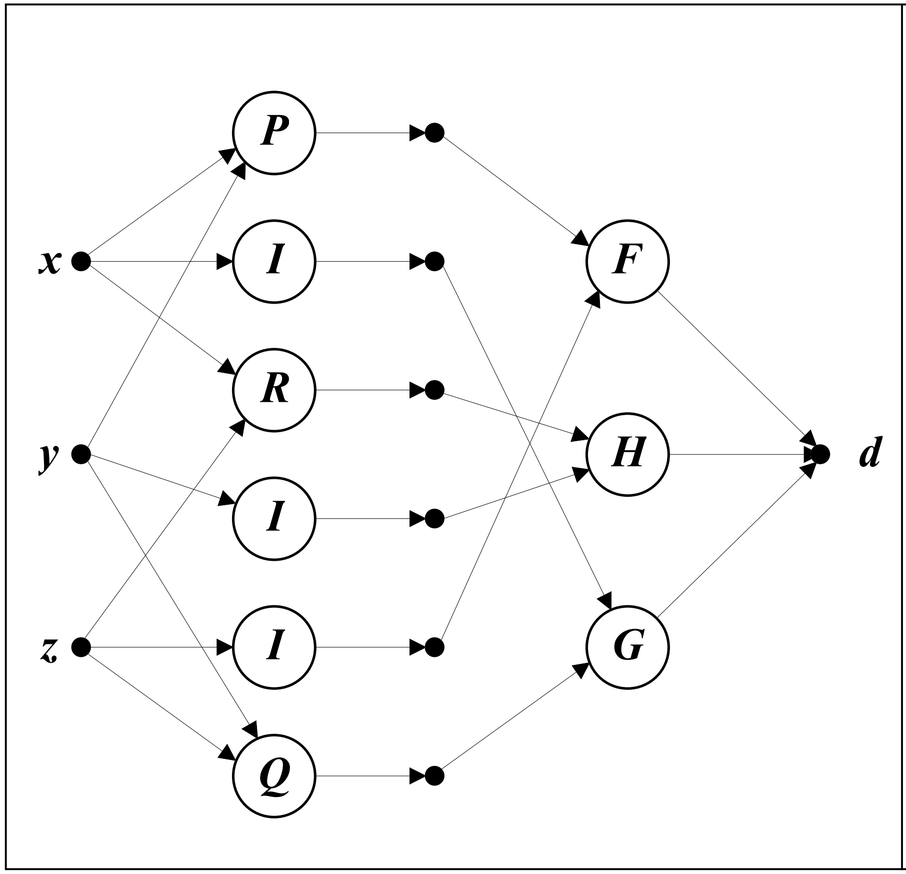
$$H(x,y) = k[m(x) + q(y)]; \quad R(x,y) = m^{-1}[p(x) + r(y)];$$

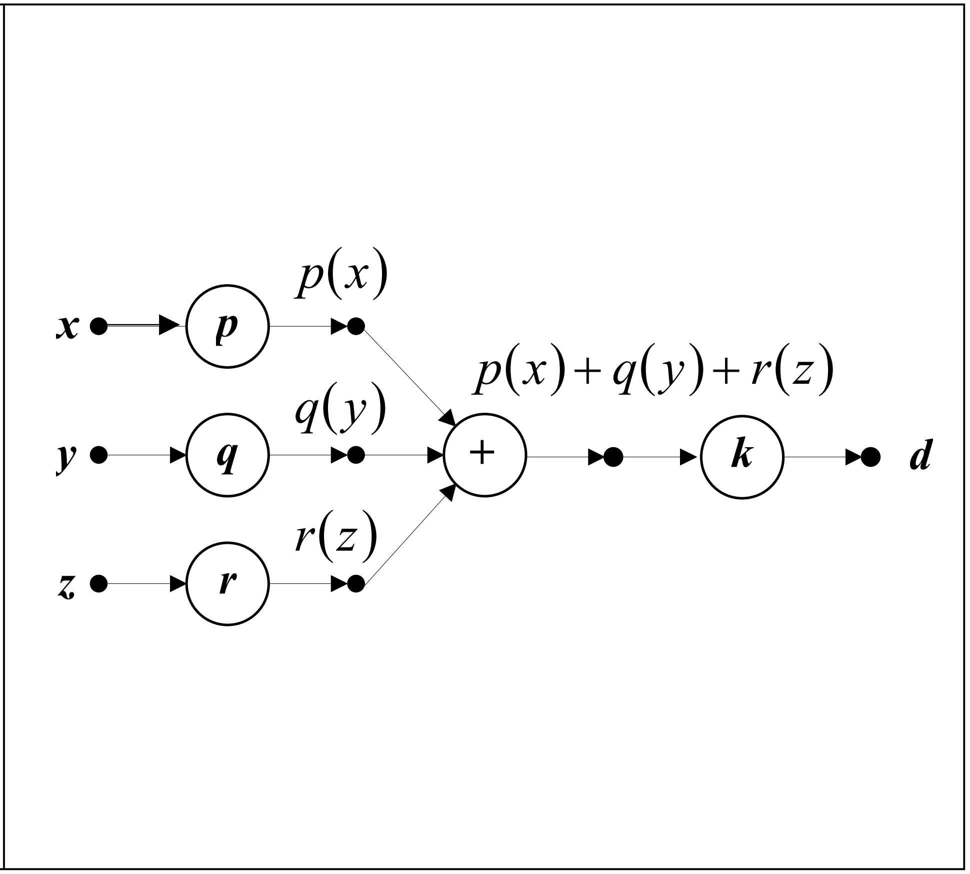
That is, $d = D(x,y,z) = k[p(x) + q(y) + r(z)]$.

Therefore,

$$F(P(x,y),z) = G(Q(y,z),x) = H(R(x,z),y)$$

is equivalent to: $k[p(x)+q(y)+r(z)]$





3. Uniqueness of Representation:

To avoid estimation problems we need to know the conditions for uniqueness (whether or not several sets of neurons (functions) lead to exactly the same output for the same input).

Example: Generalized Associative FNs:

Suppose that there exist two sets of functions

$$\{k_1,\ p_1, q_1, r_1\} \text{ and } \{k_2,\ p_2, q_2, r_2\}$$

such that

$$k_1[p_1(x)+q_1(y)+r_1(z)] = k_2[p_2(x)+q_2(y)+r_2(z)]$$

for all x, y, z.

Then,

$$k_2(u) = k_1\left(\frac{u-b-c-d}{a}\right), \quad p_2(x) = a\ p_1(x) + b,$$

$$q_2(y) = a\ q_1(y) + c, \qquad r_2(z) = a\ r_1(z) + d,$$

where a, b, c, and d, are arbitrary constants that should be fixed at some point (x_0, y_0, z_0, u_0) to guarantee the uniqueness of the solution. That is,

$$p(x_0) = \alpha_1; \quad q(y_0) = \alpha_2;$$

$$r(z_0) = \alpha_3; \quad k(u_0) = \alpha_0.$$

4. Parametric Learning:

Given a data set consisting of n observations and a family of linearly independent functions $\Phi = \{\phi_j(X), j = 1,\ldots, q\}$ and q, the number of elements in the family, there are two possibilities for the parametric learning in FNs:

- Nonlinear, and
- Linear

Nonlinear Parametric Learning:

Example: Generalized Associative FNs:

Let $\{(x_i, y_i, z_i; d_i), i = 1, \ldots, n\}$ be the data set, we can write the model as

$$d_i = k[p(x_i) + q(y_i) + r(z_i)] + \varepsilon_i, \ \ i = 1, \ldots, n.$$

These functions can be approximated by:

$$p(x) = \sum_{j=1}^{q} a_j \phi_j(x); \qquad q(y) = \sum_{j=1}^{q} b_j \psi_j(y);$$

$$r(z) = \sum_{j=1}^{q} c_j \delta_j(y).$$

Nonlinear Parametric Learning:

We can minimize:

$$Q_1 = \sum_{i=1}^{n} \varepsilon_i^2; \qquad Q_2 = \sum_{i=1}^{n} \left|\varepsilon_i\right|; \qquad Q_3 = \max_i \left|\varepsilon_i\right|;$$

subject to the uniqueness constraints.

These leads to a nonlinear system of equations or to a nonlinear programming problem.

We can also use other methods such as MARS, ACE, etc.

Linear Parametric Learning:

If k is invertible, we can write the model as

$$k^{-1}(d_i) = p(x_i) + q(y_i) + r(z_i) + \varepsilon_i, \quad i = 1, \ldots, n.$$

Then, approximating the functions and using the criteria given above lead to a linear system of equations.

5. Model Selection:

There are two questions to be answered when selecting a functional network:

- Which family of functions to use?
- Which terms in the family are important?

Families of Linearly Independent Functions

Polynomial family: $\Phi = \{1, X, X^2, \ldots, X^q\}$

Exponential family:

$$\Phi = \{1, e^{X}, e^{-X}, e^{2X}, e^{-2X}, \ldots, e^{qX}, e^{-qX}\}$$

Fourier family:

$$\Phi = \{1, \sin X, \cos X, \ldots, \sin(qX), \cos(qX)\}$$

Criteria for Selecting the Important Terms

Minimum Description Length (MDL)

Let $\mathbf{x}$ be a sample of size n and let θ be the set of parameters to be estimated:

$$L(\mathbf{x}) = \underbrace{-\log \pi_m(\theta)}_{\text{Prior expert opinion}} + \underbrace{\frac{k \log n}{2}}_{\text{Penalty for complexity}} + \frac{n}{2} \underbrace{\log\left(\frac{1}{n} \sum_{i=1}^{n} \varepsilon_i(\theta)^2 \right)}_{\text{Goodness of fit}}$$

6. Model Validation:

Tests for quality and/or cross validations are performed.

Care must also be given to the problem of overfitting.

See Castillo et al. (2000) for details.

Agenda

✓1. A brief introduction to FNs
✓2. Working with FNs
3. Differences between FNs and NNs
4. Relationship to other statistical models
5. Examples of some applications of FNs
6. Summary and Conclusions

Differences Between FNs and NNs:

1. The topology of a NN is chosen from among several topologies using trial and error. The initial topology in FN is problem driven and it can be simplified using functional equations.

Differences Between FNs and NNs:

2. In standard NNs the neural functions are given and some weights are learned. In FNs specification of the neural functions is not required because the neural functions can be learned from data.

Differences Between FNs and NNs:

3. In standard NNs all the neural functions are identical, univariate and single-argument (a weighted sum of input values). In FNs the neural functions can be different, multivariate, and/or multiargument.

Differences Between FNs and NNs:

4. In FNs we can connect outputs of different neurons to force them to coincide. This structure is not possible in standard neural networks because there are no intermediate layers and these connections are not allowed.

Differences Between FNs and NNs:

5. Neural functions are not restricted to be a linear combination of inputs.

Summary:

Functional Networks are a generalization of Neural Networks: Every problem that can be solved by NN can be solved by FN. The converse is not true.

Relationship to Some Other Statistical Models and Methods:

FNs are also generalizations of and can benefit from the following models:

1. Alternate Conditioning Expectation (ACE), due to Breiman and Friedman (1985)

   It can be used in parametric learning

Relationship to Some Other Statistical Models and Methods:

2. Multivariate Adaptive Regression Spline (MARS) due to Friedman (1991)

   It can be used in parametric learning

3. The Generalized Additive Models (GAM). See, e.g., Hastie and Tibshirani, (1990)

   It can be used in structural learning

4. Other statistical methods (e.g., variable selection). It can be used in parametric learning

Agenda

✓1. A brief introduction to FNs

✓2. Working with FNs

✓3. Differences between FNs and NNs

✓4. Relationship to other statistical models

5. Examples of some applications of FNs

6. Summary and Conclusions

Some Examples of Applications of FNs

FNs have numerous applications:

1. Bayesian Statistics
2. Finding stable and reproductive families of distributions
3. Time series
4. Modeling structural engineering problems
5. Transformations of variables
6. Nonlinear Regression
7. Iterative problems

Conjugate Family of Distributions

In Bayesian statistics a typical problem consists of finding a family of density functions $F(\theta,\eta)$, with hyperparameter η, so that both, the prior probability density function, $F(\theta,\eta)$, and the posterior, $S(\theta; \mathrm{x}, \eta)$, belong to the same family.

Conjugate Family of Distributions

In Bayesian statistics a typical problem consists of finding a family of density functions $F(\theta\,;\eta)$, with hyperparameter η, so that both, the prior probability density function $F(\theta\,;\eta)$, and the posterior $S(\theta\,;x,\eta)$, belong to the same family.

Bayes theorem guarantees that the posterior is proportional to the product of the prior and the likelihood $L(x,\theta)$.

$$S(\theta;x,\eta) = H(x,\theta)F(\theta,\eta),$$

Conjugate Family of Distributions

The functional equation for this problem is:

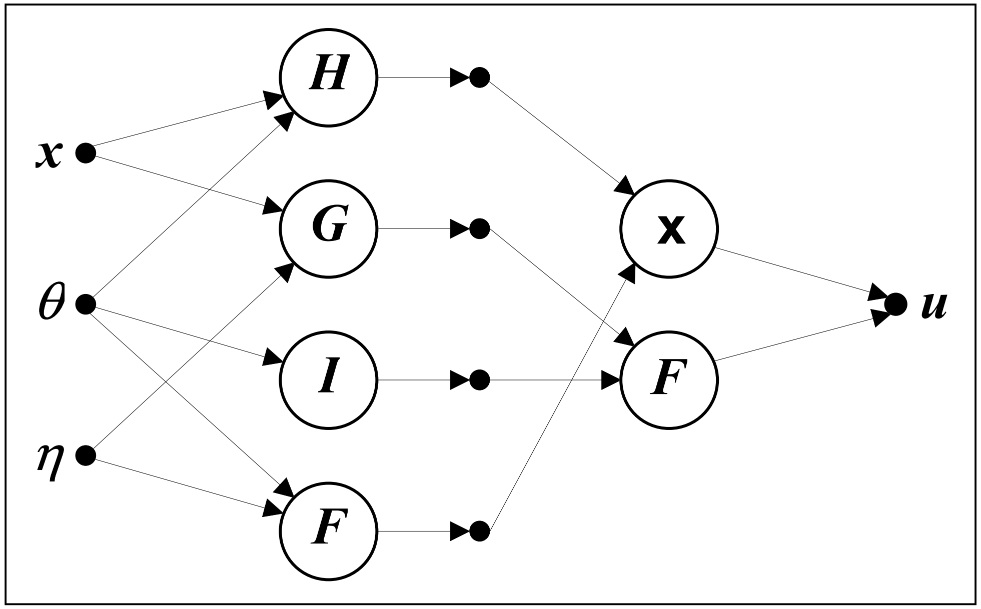
$$F(\theta, G(x,\eta)) = H(x,\theta)F(\theta,\eta),$$

Conjugate Family of Distributions

The functional equation for this problem is:

$$F(\theta, G(x,\eta)) = H(x,\theta)F(\theta,\eta),$$

which leads to the FN:



Stability with Respect to Maxima Operations

Let X and Y be two independent random variables with cumulative probability distribution functions in the parametric family

$$\{F(z,\theta),\theta \in \Theta\}.$$

The CDF of the random variable $Z = \max(X, Y)$ is

$$T(z;a,b) = F(z;a)F(z;b)$$

Stability with Respect to Maxima Operations

If we wish the family to be stable with respect to maxima operations, we must have

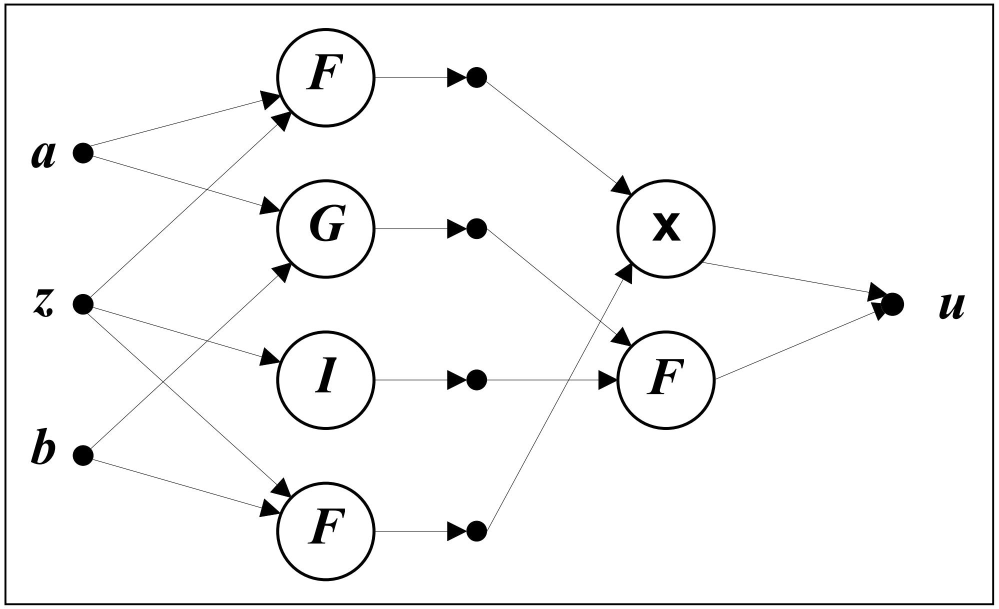
$$F(z;G(a,b)) = F(z;a)F(z;b),$$

Stability with Respect to Maxima Operations

If we wish the family to be stable with respect to maxima operations, we must have

$$F(z;G(a,b)) = F(z;a)F(z;b),$$

which leads to the FN:



Time Series: Double Logistic Model

The double logistic application is given by the following iterative equation:

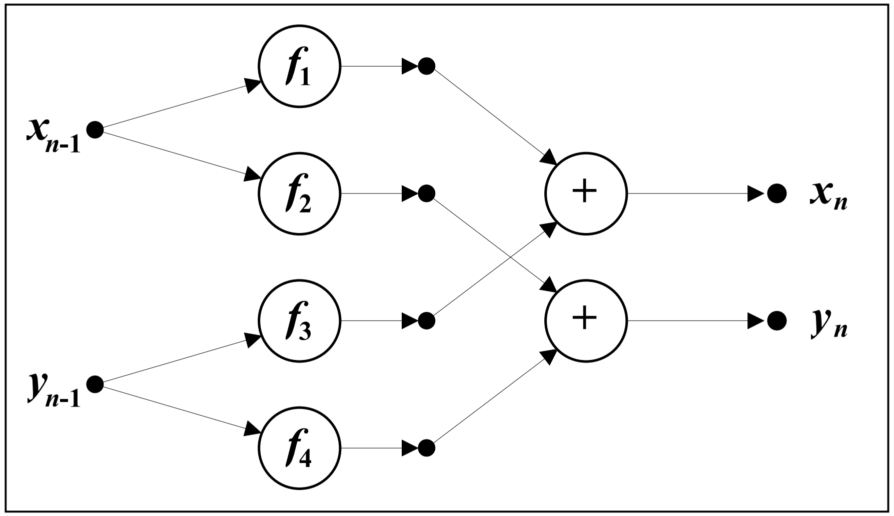
$$x_n = (1-a)x_{n-1} + 4a\,y_{n-1}(1-y_{n-1})$$
$$y_n = (1-a)y_{n-1} + 4a\,x_{n-1}(1-x_{n-1})$$

Time Series: Double Logistic Model

The double logistic application is given by the following iterative equation:

$$x_n = (1-a)x_{n-1} + 4\,a\,y_{n-1}(1-y_{n-1})$$
$$y_n = (1-a)y_{n-1} + 4\,a\,x_{n-1}(1-x_{n-1})$$

That leads to the functional network:



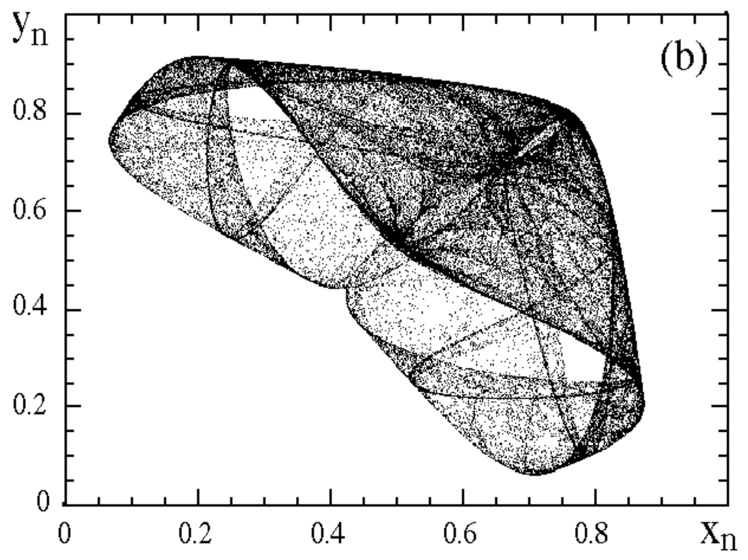
Time Series: Double Logistic Model

This family of applications is very useful to illustrate the changes in the qualitative behavior when the control parameter a is modified.

Time Series: Double Logistic Model

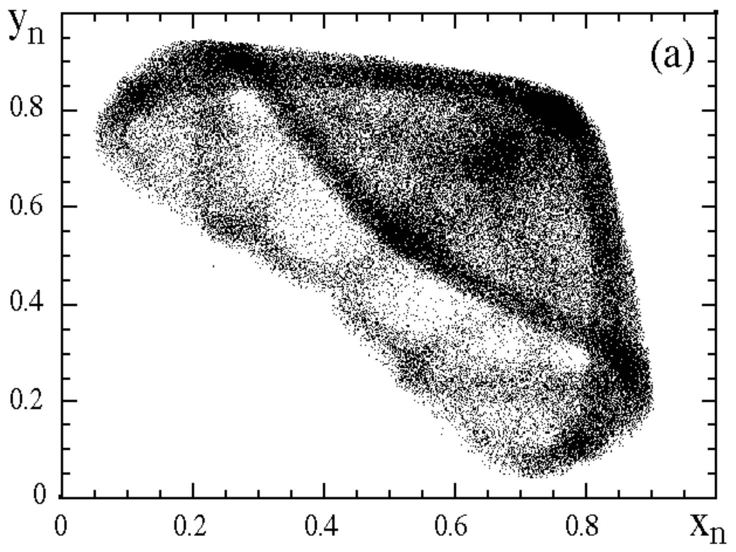
This family of applications is very useful to illustrate the changes in the qualitative behavior when the control parameter a is modified.

If $x_0 = 0.1,\ y_0 = 0.9$, and $a = 0.714$, we obtain



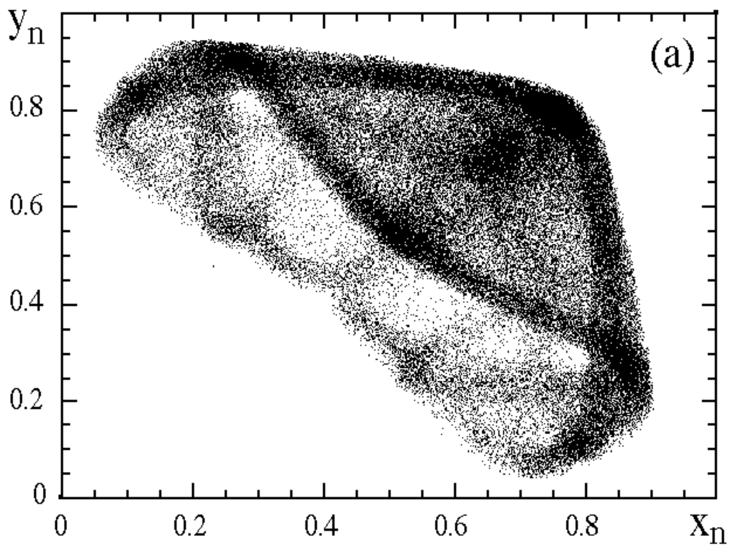
Time Series: Double Logistic Model

Adding to the model a Gaussian error with $\sigma = 0.1$, we get the plot on the right.

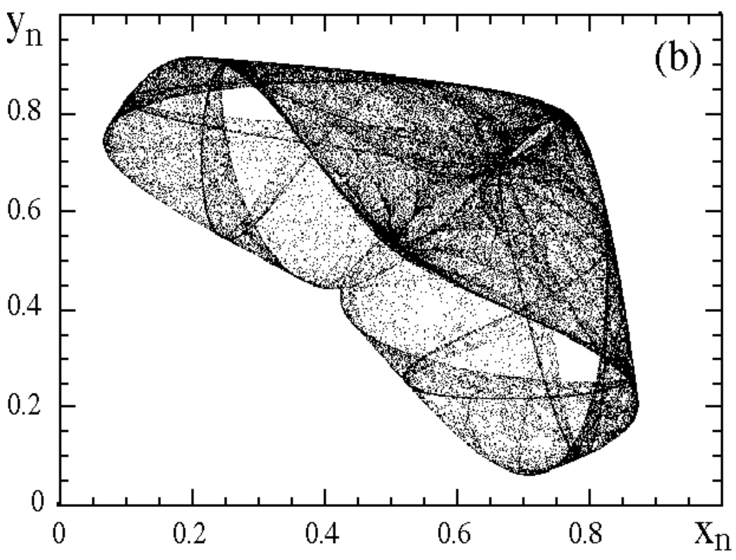


Time Series: Double Logistic Model

Adding to the model a Gaussian error with $\sigma = 0.1$, we get the plot on the right.

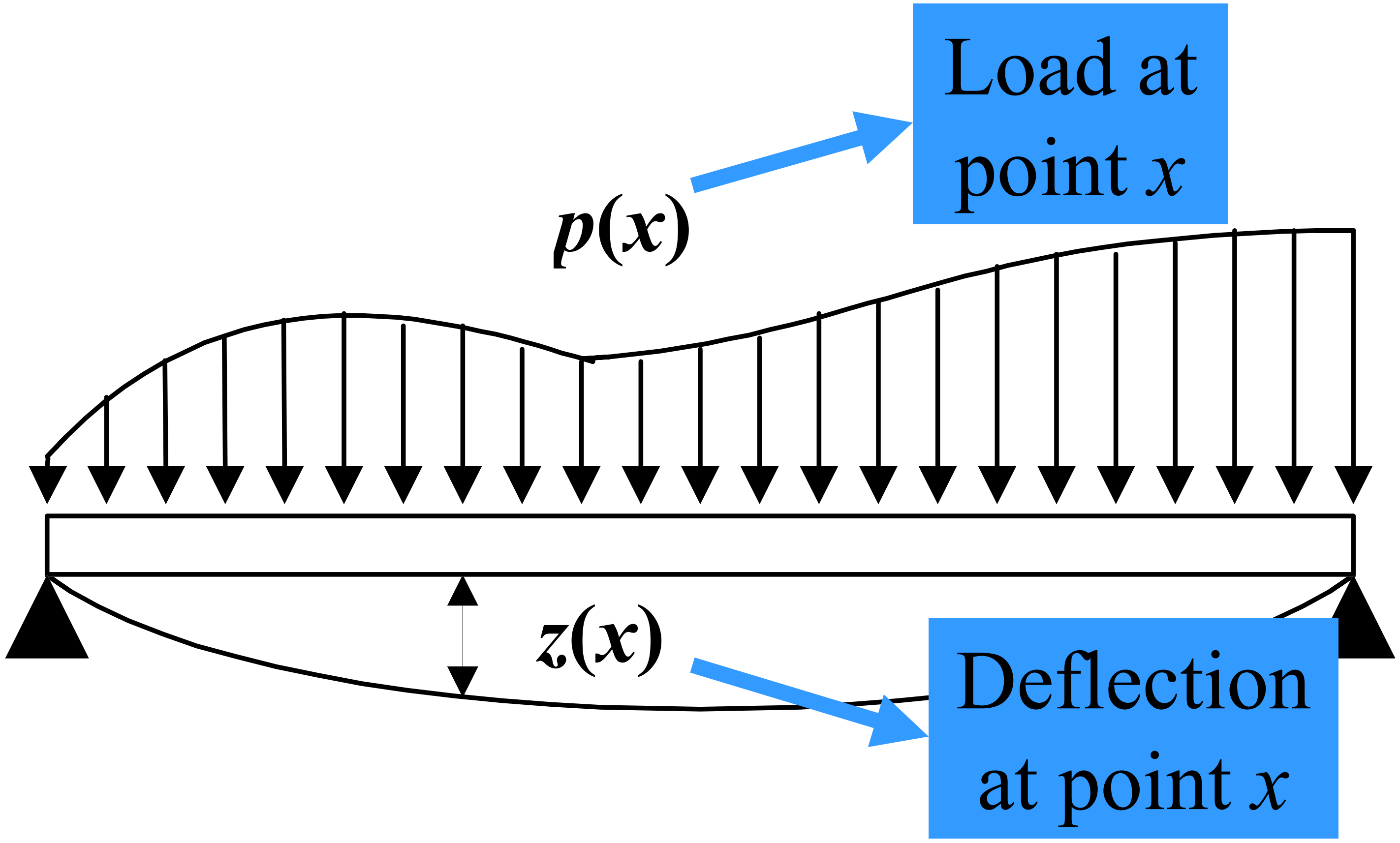


Using a FN with a polynomial family, the noise can be Eliminated, as shown in the right.



Structural Engineering: The Beam Problem

We are interested in modeling the behavior of a beam subject to given vertical forces (loads).



Structural Engineering: The Beam Problem

Data:

- Load $p(x)$ and
- Geometry of the beam.

Structural Engineering: The Beam Problem

Data:

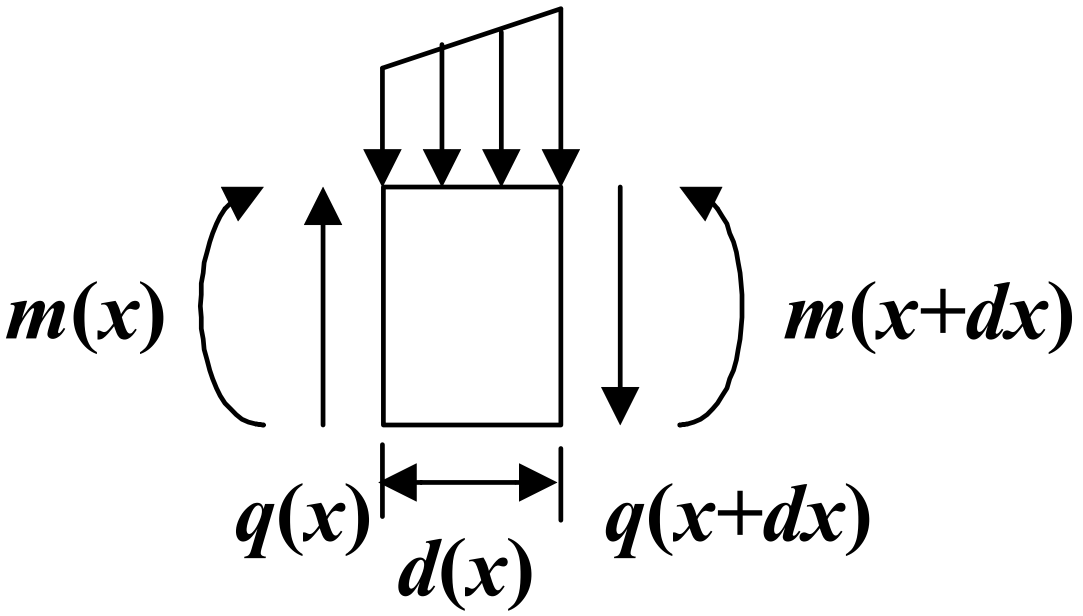
- Load $p(x)$ and
- Geometry of the beam.

Unknowns:

- Deflection $z(x)$,
- Rotation $w(x)$,
- Bending moment $m(x)$, and
- Shear force $q(x)$.

Structural Engineering: The Beam Problem

In the classical approach, the equilibrium forces are stated for differential pieces.



Structural Engineering: The Beam Problem

Using force and moment equilibrium equations plus strength of materials, we get a fourth order differential equations.

$$EIz^{(iv)} = p(x),$$

Structural Engineering: The Beam Problem

Using force and moment equilibrium equations plus strength of materials, we get a fourth order differential equations.

$$EIz^{(iv)} = p(x),$$

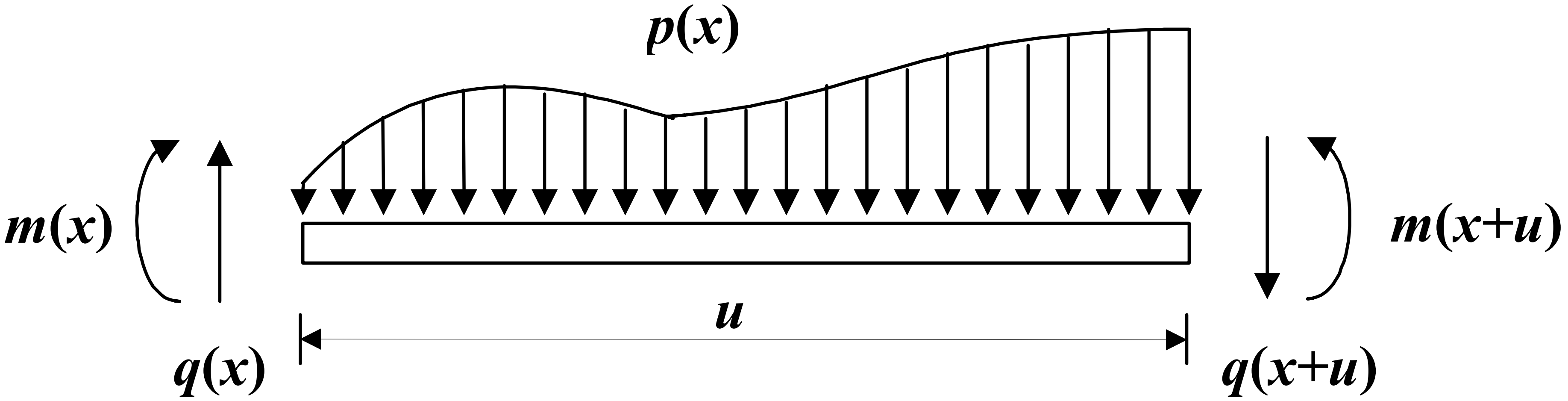
which is equivalent to a system of four first order differential equations

$$q'(x) = p(x); \quad w'(x) = \frac{m(x)}{EI};$$

$$m'(x) = q(x); \quad z'(x) = w(x).$$

Structural Engineering: The Beam Problem

With the functional networks approach, the equations are stated for discrete pieces of length u.



Structural Engineering: The Beam Problem

Using force and moment equilibrium equations plus strength of materials leads to:

$$z(x+4u) = -z(x) + 4z(x+u) - 6z(x+2u) + 4z(x+3u)$$
$$+\left(-4D(x,u) + 6D(x,2u) - 4D(x,3u) + D(x,4u)\right)/EI$$

which is a functional equation that, for constant u, can be seen as a difference equation of fourth order

Structural Engineering: The Beam Problem

This is equivalent to the system of first order functional equations:

$$q(x+u)=q(x)+A(x,u)$$
$$m(x+u)=m(x)+uq(c)+B(x,u)$$
$$w(x+u)=w(x)+\frac{1}{EI}\left[m(x)u+q(x)\frac{u^2}{2}+C(x,u)\right]$$
$$z(x+u)=z(x)+w(x)u+\frac{1}{EI}\left[m(x)\frac{u^2}{2}+q(x)\frac{u^3}{6}+D(x,u)\right]$$

where A, B, C and D can be calculated from $p(x)$.

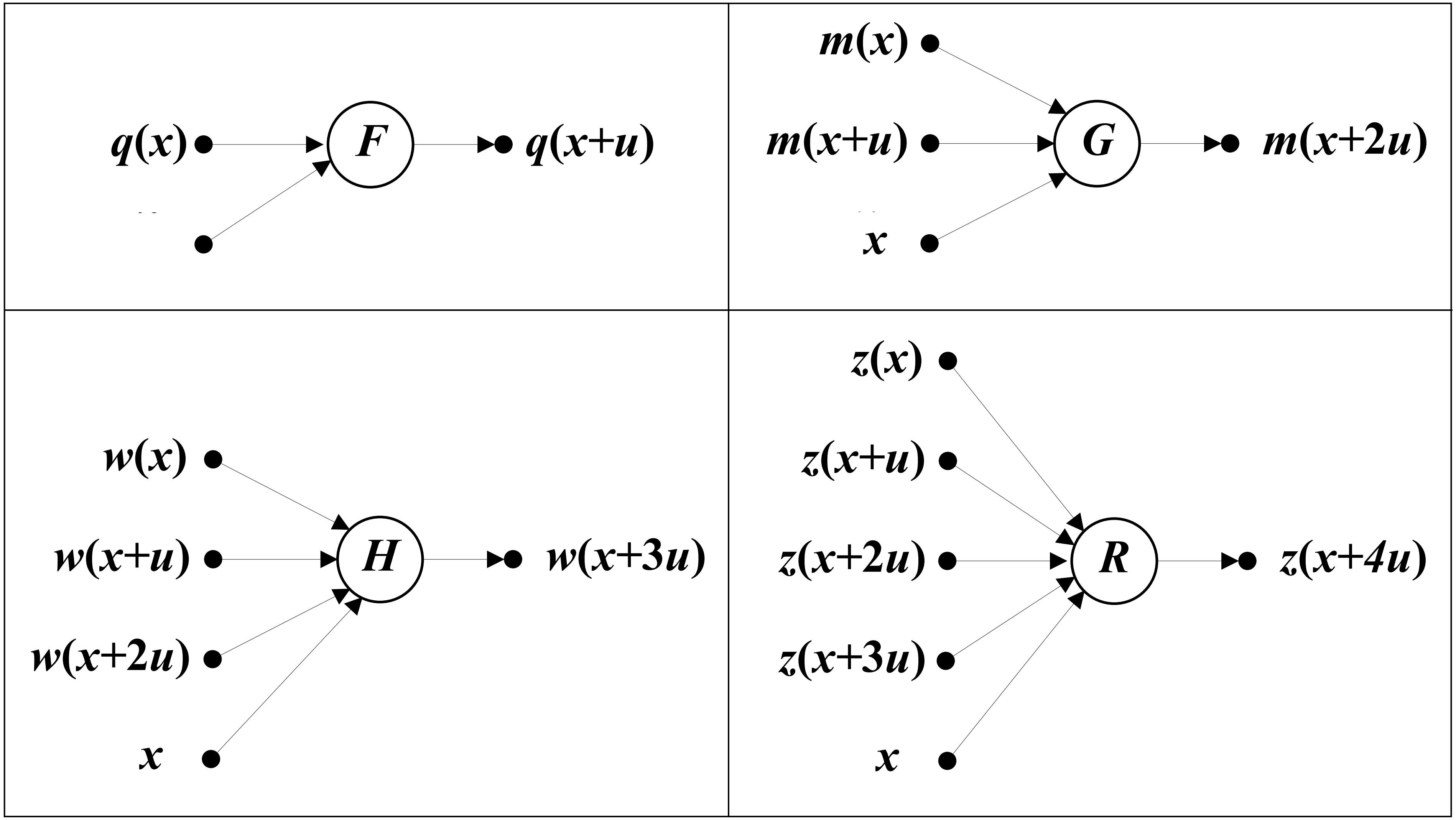
Structural Engineering: The Beam Problem

Another alternative is the set of functional equations

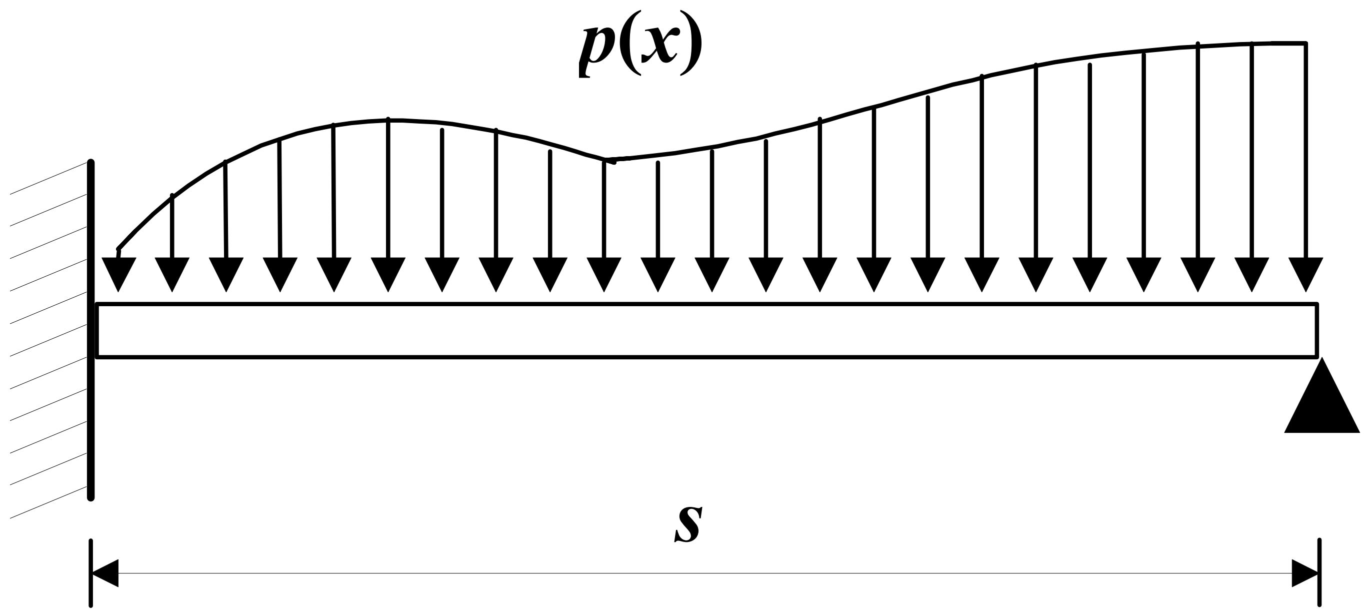
$$\begin{aligned}
z(x+4u) = {} & -z(x)+4z(x+u)-6z(x+2u)+4z(x+3u) \\
& +(-4D(x,u)+6D(x,2u)-4D(x,3u)+D(x,4u))/EI \\
w(x+3u) = {} & w(x)-3w(x+u)+3w(x+2u) \\
& +(3C(x,u)-3C(x,2u)+C(x,3u))/EI \\
m(x+2u) = {} & -m(x)+2m(x+u)-B(x,u) \\
q(x+u) = {} & q(x)+A(x,u)
\end{aligned}$$

that, for constant u, can be seen as difference equations.

Structural Engineering: The Beam Problem



Example: Supported Cantilever Beam



Boundary conditions are:

$$w(0) = z(0) = m(s) = z(s) = 0.$$

Example: Supported Cantilever Beam

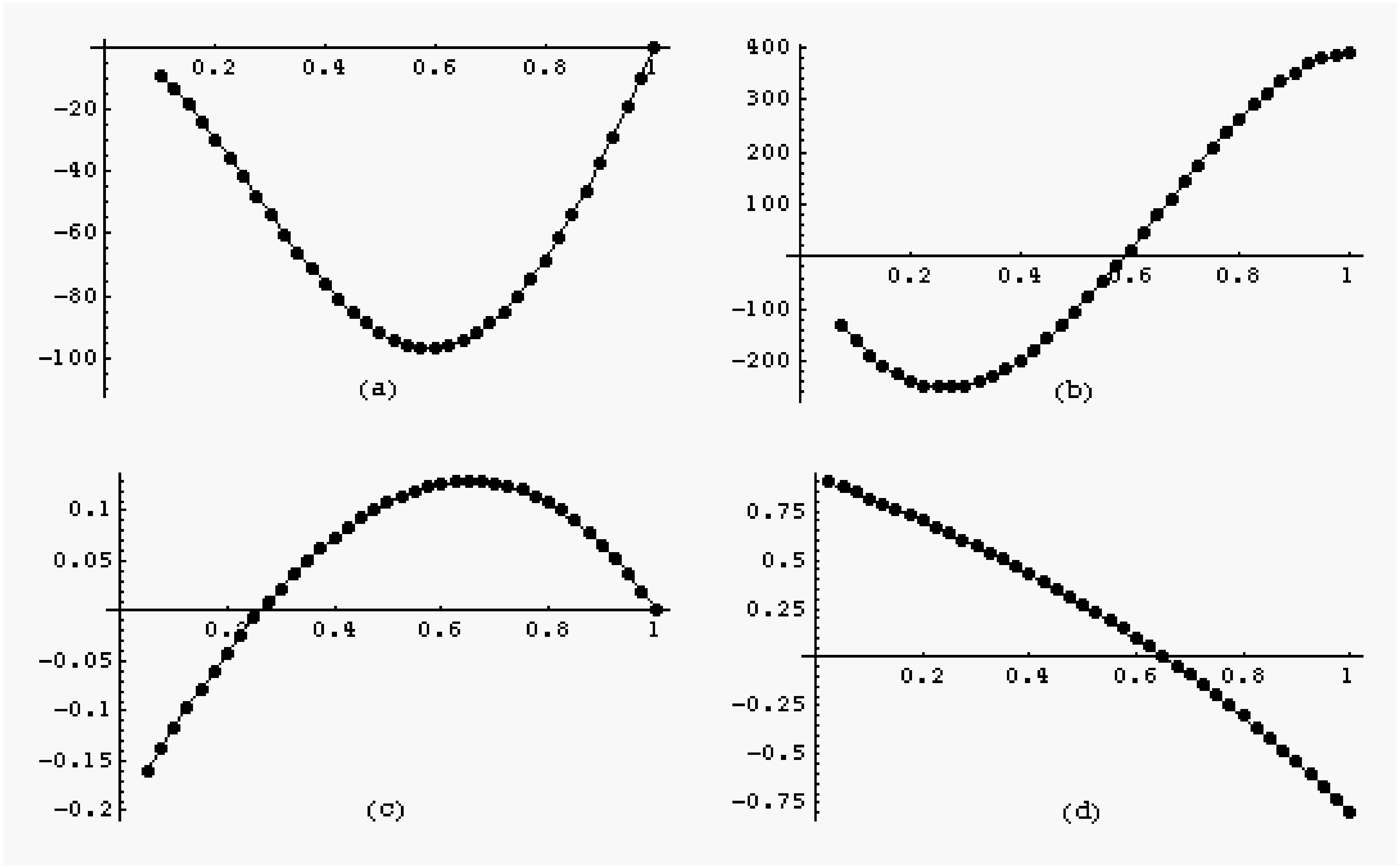
$$z(x) = -0.0040 - 0.0041\,x - 0.0021\,x^2 - 0.0007\,x^3 - 0.0002\,x^4 - 0.00003\,x^5 - 0.00001\,x^6$$

$$w(x) = -0.1620 - 0.1620\,x - 0.0810\,x^2 - 0.0270\,x^3 - 0.0069\,x^4 - 0.0012\,x^5 - 0.0004\,x^6$$

$$m(x) = -0.0006 - 0.0006\,x - 0.0003\,x^2 - 0.0001\,x^3 - 0.00003\,x^4 - 0.000005\,x^5 - 1.44\,x^6$$

$$q(x) = -0.0250 - 0.0253\,x - 0.0127\,x^2 - 0.0042\,x^3 - 0.0011\,x^4 - 0.0002\,x^5 - 0.00006\,x^6$$

Example: Supported Cantilever Beam



Example: The Iterator

Suppose that we wish to calculate the *n*-th iterate of a given function *f*, that is:

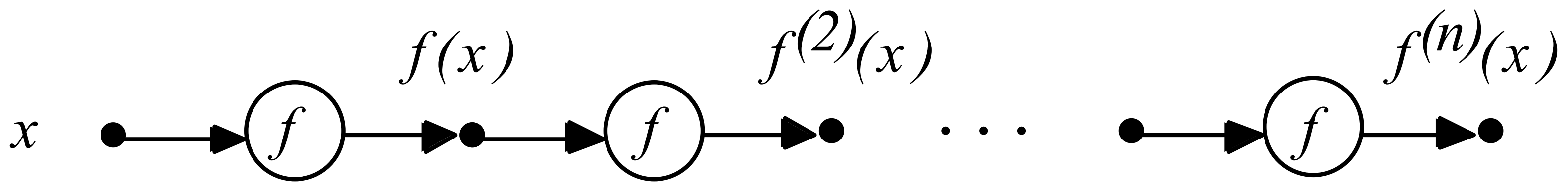
$$y = f(f(\ldots f(x))) = f^{(n)}(x)$$

Example: The Iterator

Suppose that we wish to calculate the *n*-th iterate of a given function *f*, that is:

$$y = f(f(\ldots f(x))) = f^{(n)}(x)$$

1. Selecting the Initial Topology:



2. Simplifying the Initial Topology

Let

$$f^{(n)}(x) = F(x,n) \quad \Rightarrow \quad f(x) = F(x,1)$$

2. Simplifying the Initial Topology

Let

$$f^{(n)}(x) = F(x,n) \quad \Rightarrow \quad f(x) = F(x,1)$$

Since

$$f^{(m+n)}(x) = f^{(n)}(f^{(m)}(x)),$$

2. Simplifying the Initial Topology

Let

$$f^{(n)}(x) = F(x,n) \quad \Rightarrow \quad f(x) = F(x,1)$$

Since

$$f^{(m+n)}(x) = f^{(n)}(f^{(m)}(x)),$$

then F(x,n) satisfies the functional equation

$$F(x, m+n) = F(F(x,m), n).$$

2. Simplifying the Initial Topology

With general solution

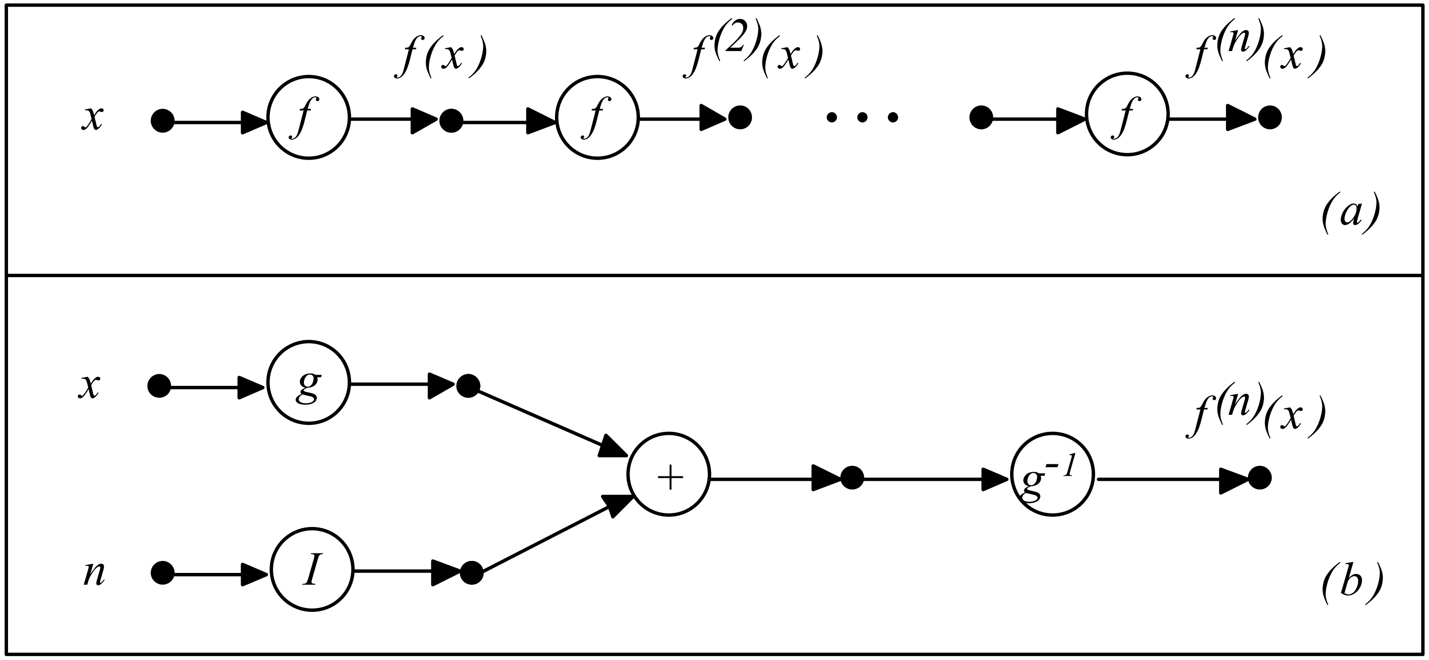
$$y = f^{(n)}(x) = F(x,n) = g^{-1}[g(x)+n].$$

2. Simplifying the Initial Topology

With general solution

$$y = f^{(n)}(x) = F(x,n) = g^{-1}[g(x)+n].$$

So, the two functional networks are equivalent:



3. Uniqueness of Representation:

Since

$$f^{(n)}(x) = F(x,n) = g^{-1}[g(x)+n].$$

3. Uniqueness of Representation:

Since

$$f^{(n)}(x) = F(x,n) = g^{-1}[g(x)+n].$$

The uniqueness of representation implies solving the functional equation:

$$g^{-1}[g(x)+n] = h^{-1}[h(x)+n],$$

3. Uniqueness of Representation:

Since

$$f^{(n)}(x) = F(x,n) = g^{-1}[g(x)+n].$$

The uniqueness of representation implies solving the functional equation:

$$g^{-1}[g(x)+n] = h^{-1}[h(x)+n],$$

with unique solution

$$g(x) = h(x)+c,$$

where c is an arbitrary constant.

3. Uniqueness of Representation:

Since

$$f^{(n)}(x) = F(x,n) = g^{-1}[g(x)+n].$$

The uniqueness of representation implies solving the functional equation:

$$g^{-1}[g(x)+n] = h^{-1}[h(x)+n],$$

with unique solution

$$g(x) = h(x) + c,$$

where c is an arbitrary constant.

So, the function g must be fixed at a point.

4. Learning the Model

To learn the function g we write the expression

$$y = f(x) = F(x,1) = g^{-1}[g(x)+1].$$

in the more convenient form.

$$g(y_t) = g(x_t)+1; \quad t = 1,2,\ldots,m$$

4. Learning the Model

To learn the function g we write the expression

$$y = f(x) = F(x,1) = g^{-1}[g(x)+1].$$

in the more convenient form.

$$g(y_t) = g(x_t) + 1; \quad t = 1,2,\ldots,m$$

To learn g(x), we consider a linear combination of basic functions

$$g(x) = \sum_{i=1}^{k} c_i \phi_i(x),$$

4. Learning the Model

And define the errors

$$\varepsilon_t = g(y_t) - g(x_t) - 1 = \sum_{i=1}^{k} c_i(\phi_i(y_t) - \phi_i(x_t)) - 1,$$

4. Learning the Model

And define the errors

$$\varepsilon_t = g(y_t) - g(x_t) - 1 = \sum_{i=1}^{k} c_i(\phi_i(y_t) - \phi_i(x_t)) - 1,$$

Then, we estimate the coefficients by minimizing.

$$Q = \sum_{t=1}^{m} \varepsilon_t^2 = \sum_{t=1}^{m} \ [\sum_{i=1}^{k} c_i(\phi_i(y_t) - \phi_i(x_t)) - 1]^2$$

subject to

$$g(x_0) = \sum_{i=1}^{k} c_i \ \phi_i(x_0) = x_0.$$

An Example

Consider the function

$$f(x) = \log(1 + exp(x)),$$

and suppose that we are interested in its n-iterate.

An Example

Consider the function

$$f(x) = \log(1 + exp(x)),$$

and suppose that we are interested in its n-iterate.

Assume also that we have a set of data points

$$\{(x_t, y_t) \mid t = 1, 2, \ldots, m\},$$

where $y_t = f(x_t)$, to learn the function f(x).

An Example

Consider the function

$$f(x) = \log(1 + exp(x)),$$

and suppose that we are interested in its *n*-iterate.

Assume also that we have a set of data points

$$\{(x_t, y_t) \mid t = 1, 2, ..., m\},$$

where $y_t = f(x_t)$, to learn the function *f*(*x*).

Then, we select a polynomial family and learn

$$g(x) = \sum_{i=1}^{8} c_i x^i$$

An Example

Assume that we have the following data points:

x_t	y_t	x_t	y_t	x_t	y_t
0.010	0.698	0.428	0.930	0.415	0.922
0.580	1.020	0.187	0.791	0.198	0.797
0.696	1.100	0.866	1.220	0.310	0.860
0.310	0.860	0.906	1.250	0.971	1.290
0.305	0.857	0.296	0.852	0.242	0.822
0.646	1.070	0.575	1.020	0.536	0.997
0.191	0.793	0.635	1.060	0.017	0.702
0.820	1.190	0.340	0.877	0.304	0.857
0.007	0.697	0.392	0.908	0.925	1.260
0.724	1.120	0.820	1.180	0.194	0.795

Then, we select a polynomial family and minimize

$$Q = \sum_{t=1}^{m} \varepsilon_t^2 = \sum_{t=1}^{30} \left[\sum_{i=1}^{8} c_i (y_t^i - x_t^i) - 1\right]^2$$

An Example

We get the following models:

Exhaustive and Backward-Forward Method

Step	Approximating Functions	Quality Measures	
	$g(x)$	RMSE	L_1
1	$\{x, x^2, x^3, x^4, x^5, x^6, x^7, x^8\}$	3.28×10^{-6}	-365.201
2	$\{x, x^2, x^3, x^4, x^6, x^7, x^8\}$	4.30×10^{-8}	-496.792

$$g(x) = x + 0.5x^2 + 0.165x^3 + 0.047x^4 + 0.0087x^6 + 0.003x^7 + 0.0007x^8$$

$$RMSE = 4.3 * 10^{-8}; \quad L = -496.792$$

An Example

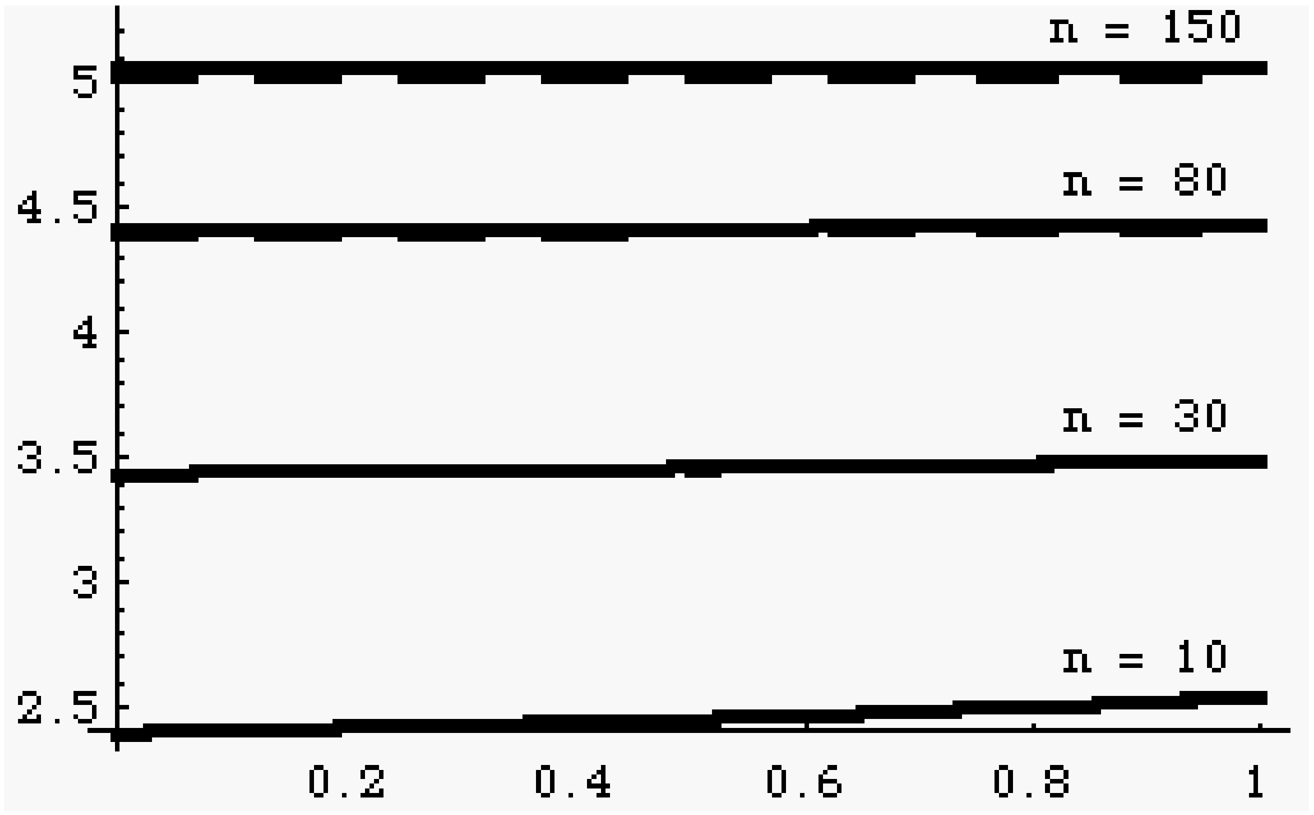
Forward-backward method:

Step	Approximating Functions	Quality Measures	
	$g(x)$	RMSE	L_1
1	$\{x^2\}$	1.1×10^{-1}	-64.043
2	$\{x^2, x\}$	2.5×10^{-2}	-107.499
3	$\{x^2, x, x^4\}$	9.7×10^{-4}	-202.926
4	$\{x^2, x, x^4, x^3\}$	8.5×10^{-5}	-274.418
5	$\{x^2, x, x^4, x^3, x^6\}$	1.9×10^{-6}	-386.595
6	$\{x^2, x, x^4, x^3, x^6, x^5\}$	1.0×10^{-7}	-472.299
7	$\{x^2, x, x^4, x^3, x^6, x^5, x^8\}$	5.5×10^{-8}	-489.394

$$g(x) = x + 0.5x^2 + 0.167x^3 + 0.042x^4 + 0.0079x^5 + 0.002x^6 + 0.00007x^8$$

$$RMSE = 5.5 * 10^{-8}; \quad L = -489.394$$

An Example



Exact and predicted values for different values of *n*.

Agenda

✓1. A brief introduction to FNs

✓2. Working with FNs

✓3. Differences between FNs and NNs

✓4. Relationship to other statistical models

✓5. Examples of some applications of FNs

6. Summary and Conclusions

Summary and Conclusions

1. Functional networks have many practical applications in probability, statistics and engineering.

2. The physical understanding of the problem to be solved gives rise to an initial topology of the functional network. No black boxes as in NN.

Summary and Conclusions

3. Functional equations allow simplifying the initial topology leading to a much simplified network.

4. This problem driven design is but one feature of functional networks that differentiates them from the standard neural networks approach.

Summary and Conclusions

5. After the uniqueness problem has been solved, learning reduces to finding unique neural functions.

6. Model selection can be performed using the minimum description length measure in conjunction with procedures such as the forward-backward or the backward-forward algorithms.

Summary and Conclusions

7. A final model validation step must be applied to prevent model overfitting.

Some Applications of Functional Networks in Statistics and Engineering

Enrique Castillo,	University of Cantabria
J. M. Gutiérrez,	University of Cantabria
Ali S. Hadi,	Cornell University
Beatriz Lacruz,	University of Zaragoza

Example: Nonlinear Regression

Consider the semi-parametric regression model:

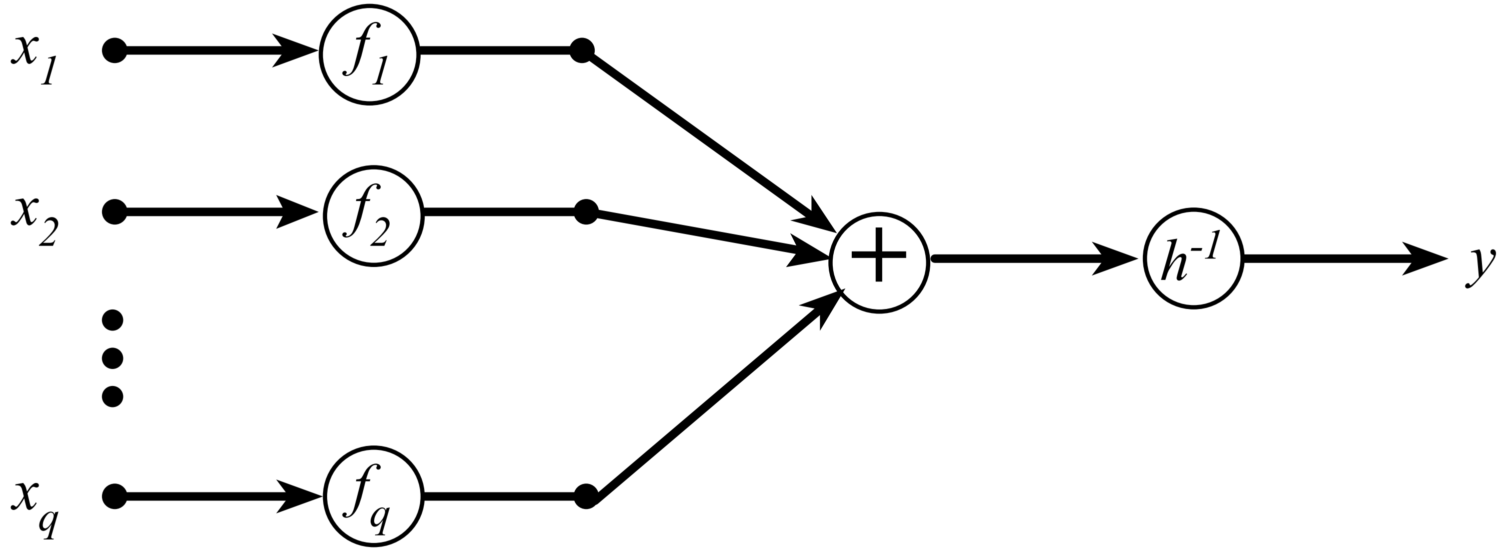
$$h(y) = f_1(x_1) + \ldots + f_q(x_q) \qquad (1)$$

FNs do not require the functions *h*(.), f_1(.), …, f_q(.) to be known.

Assuming that $h(.)$ is invertible, the semi-parametric regression model can be represented by:

$$y = h^{-1}[f_1(x_1) + \ldots + f_q(x_q)]$$

and the following FN:



Numeric Example

A data set consisting of $n = 40$ observations is generated from:

$$Y^3 = X_1 + X_1^2 + X_2^2 + X_2^3 + \varepsilon,$$

where X_1, X_2 are $U(0, 1)$ and ε is $U(-0.005, 0.005)$.

We now fit the following model to these data:

$$f(y) = f_1(x_1) + f_2(x_2),$$

which is equivalent to:

$$y = f^{-1}[f_1(x_1) + f_2(x_2)].$$