Deep Learning Foundations: Neural Networks

Daniel B. Rowe
Director of Masters of Applied Statistics
Professor of Computational Sciences
Department of Mathematics, Statistics and Computer Science

April 5, 2019

Sponsors

1. **Introduction**
   NN Structure, Activation/Score Functions, Estimation
2. **Linear Regression and NN**
   Simple & Multivariate with Gradient Descent
3. **Logistic Regression and NN**
   Simple & Multivariate with Gradient Descent
4. **NonLinear Regression and NN**
   Simple & Multivariate with Gradient Descent
5. **MultiLayer (Deep Learning) NN**
6. **Discussion**
   Reiteration and questions.

1. **Introduction**-NN Structure

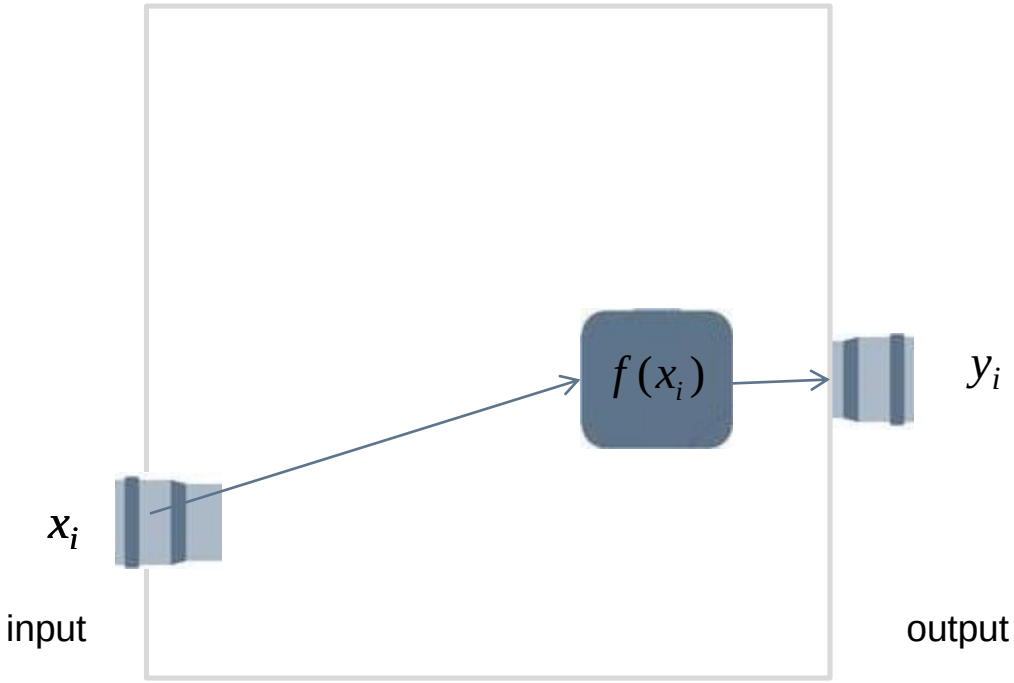
Often illustrations but no details of mathematics.

Discuss foundational ideas of neural networks.

These ideas can be expanded in many directions.

1. **Introduction**-NN Structure

Assume we are given input/output pairs, (x_i, y_i)

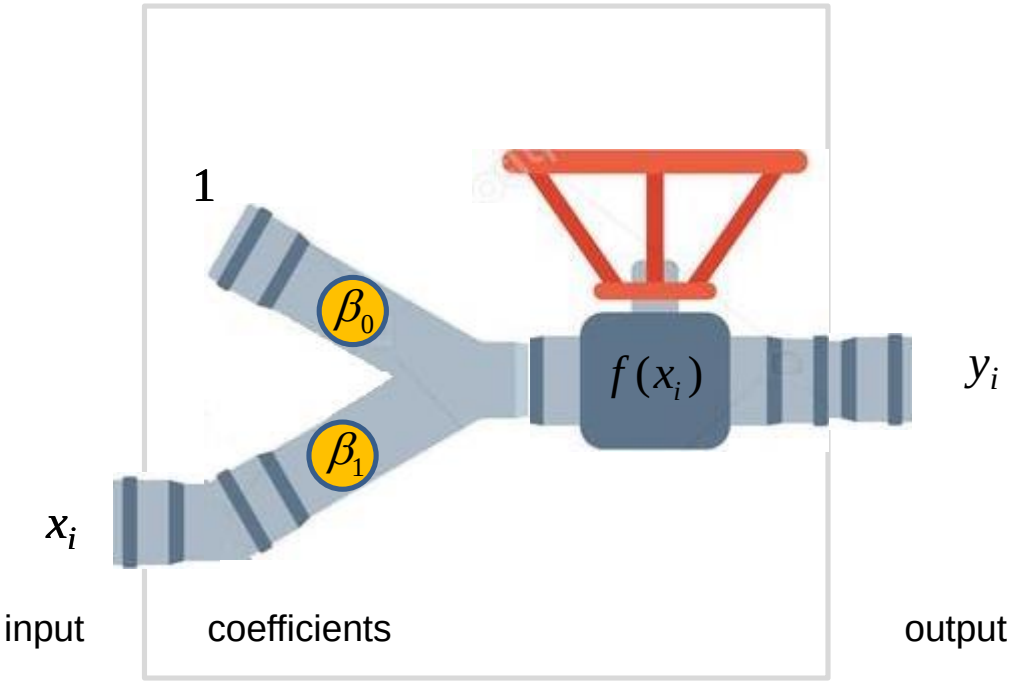


outputs	inputs
y_1	x_1
y_2	x_2
y_3	x_3
y_4	x_4
y_5	x_5
y_6	x_6
y_7	x_7
y_8	x_8
y_9	x_9
y_{10}	x_{10}

and we want to find the relationship between x and y. Learn the process that generated y from x.

1. **Introduction**-NN Structure

Assume we are given input/output pairs, (x_i, y_i)

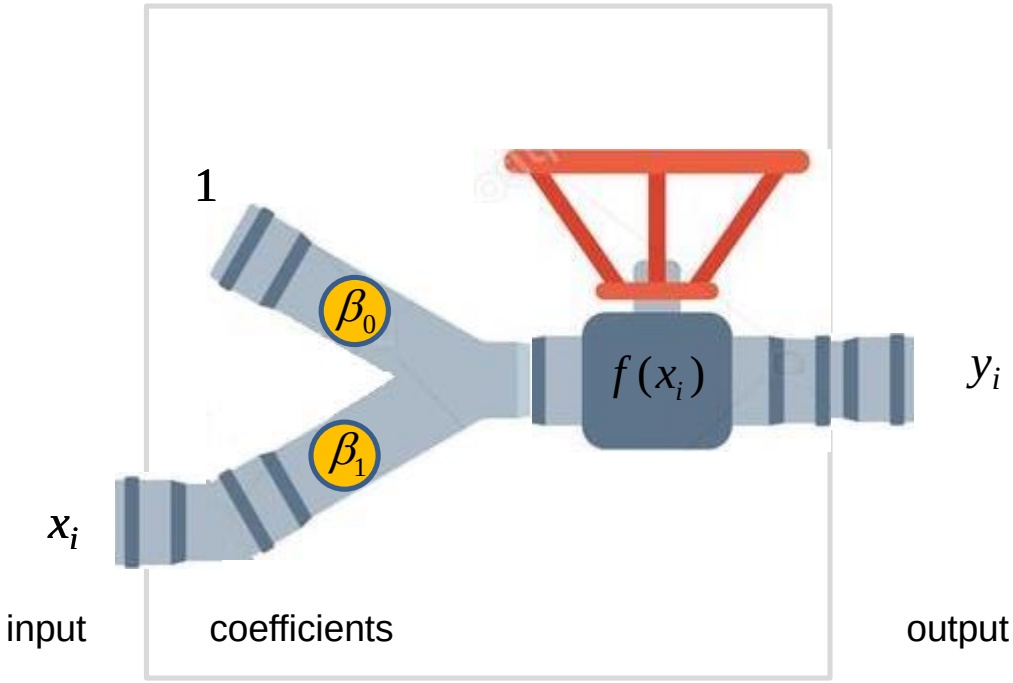


Find the relationship between x and y.

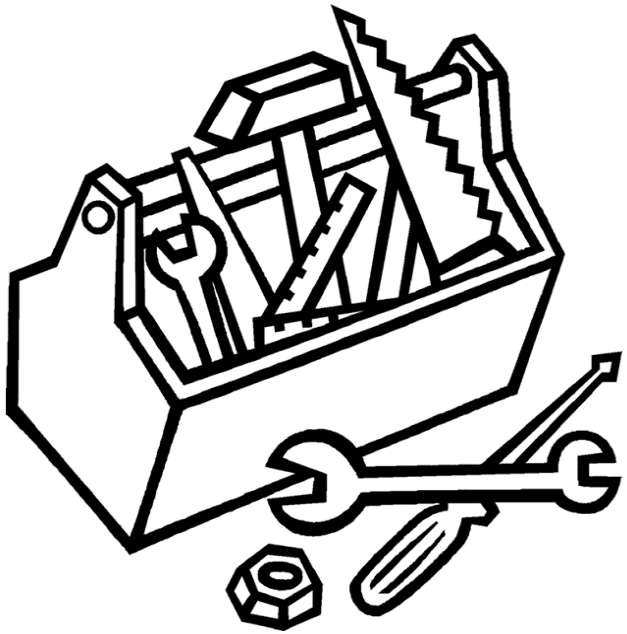
Learn process that generated y from x.

1. **Introduction**-NN Structure

Single input and a single output.



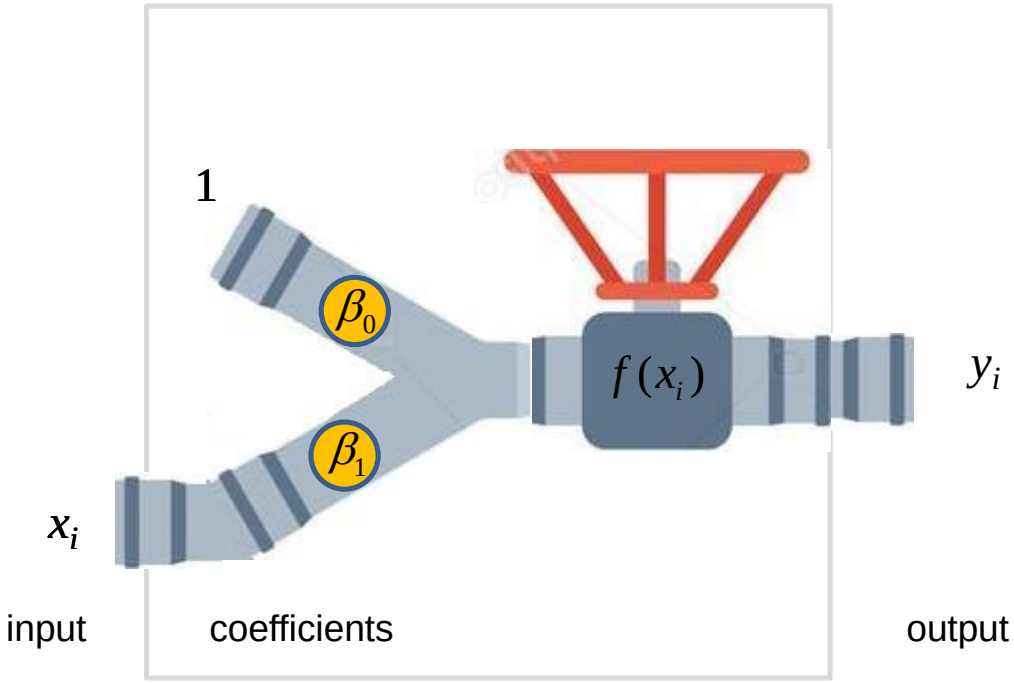
Many Tools



Math is simple for this.

1. **Introduction**-NN Structure

Single input and a single output.



Many Tools

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Objective Function

$$Q = \frac{1}{n}\sum_i (y_i - \beta_0 - \beta_1 x)^2$$

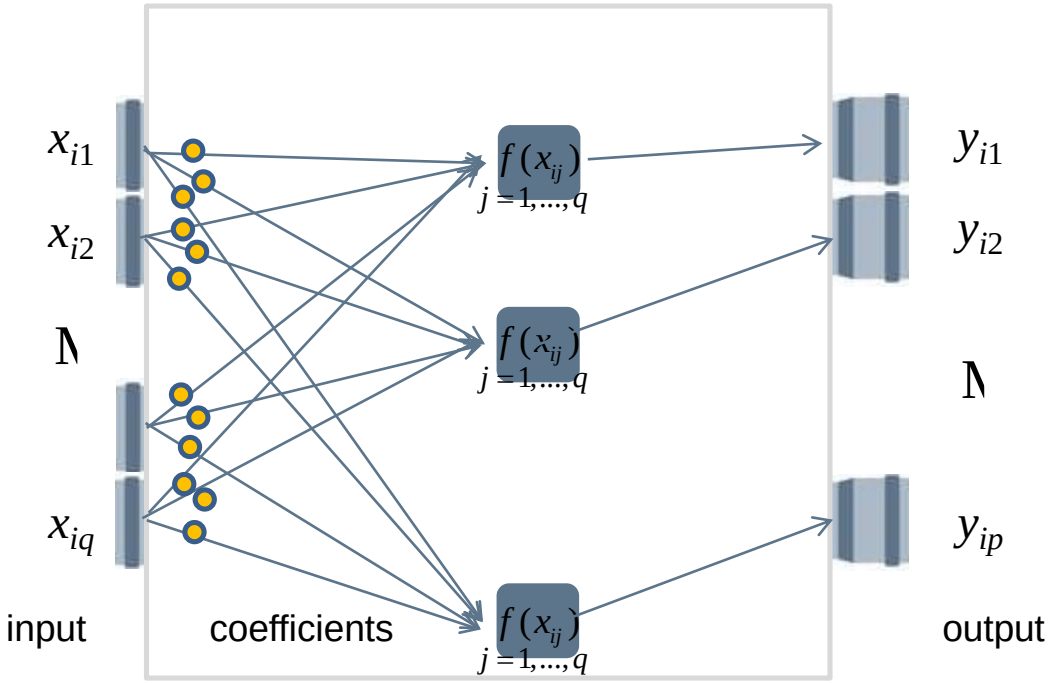
Estimate Parameters

$$(\hat{\beta}_0, \hat{\beta}_1)$$

Math is simple for this.

1. **Introduction**-NN Structure

Multiple input and multiple output.



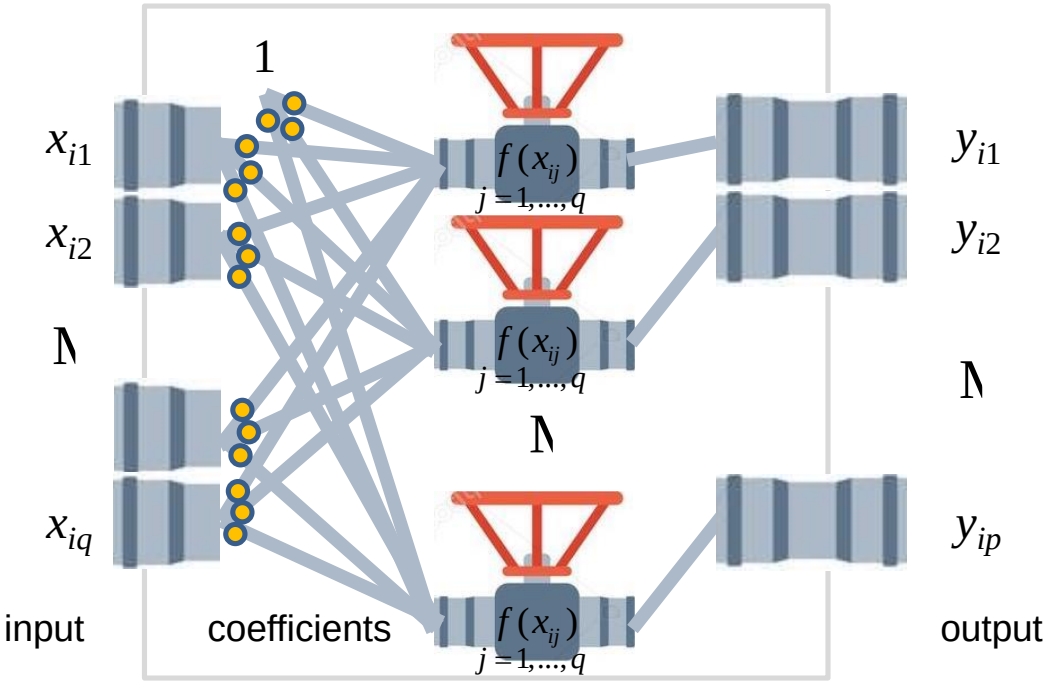
outputs	inputs
y_{i1}	x_{i1}
y_{i2}	x_{i2}
y_{i3}	x_{i3}
y_{i4}	x_{i4}
y_{i5}	x_{i5}
y_{i6}	x_{i6}
y_{i7}	x_{i7}
y_{i8}	x_{i8}
y_{i9}	x_{i9}
$y_{i,10}$	$x_{i,10}$

$i=1,\ldots,n$

Math is much more advanced.

1. **Introduction**-NN Structure

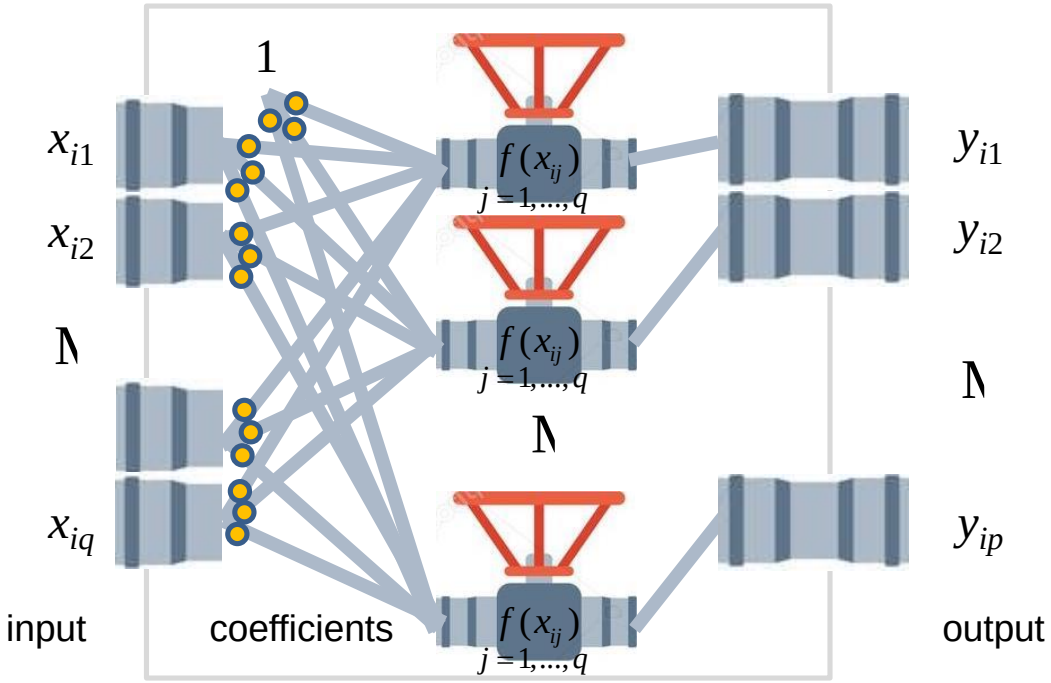
Multiple input and multiple output.



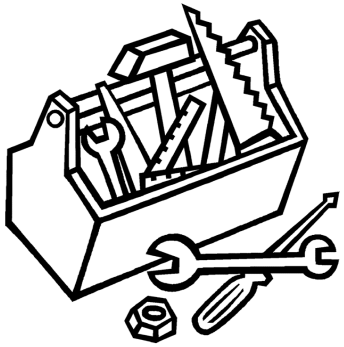
Math is much more advanced.

1. **Introduction**-NN Structure

Multiple input and multiple output.



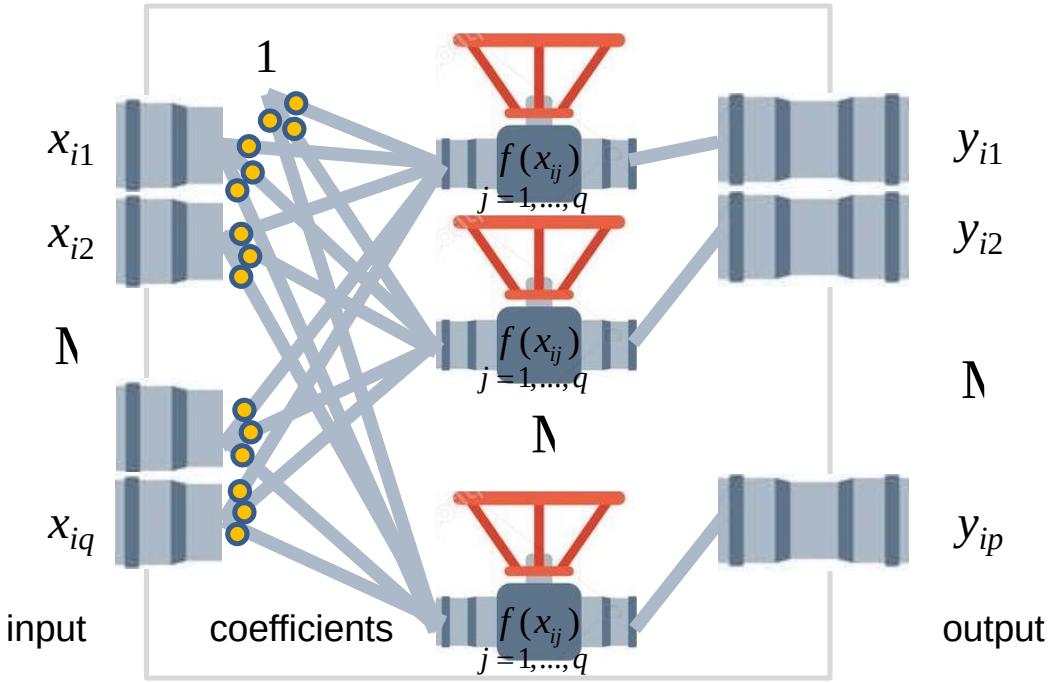
Fewer Tools



Math is much more advanced.

1. **Introduction**-NN Structure

Multiple input and multiple output.



Some Tools

$$y_i = x_i'B + \varepsilon_i$$

Objective Function

$$Q = \frac{1}{n}\sum_i (y_i - x_i'B)(y_i - x_i'B)'$$

Estimate Parameters

$$\hat{B}$$

Math is much more advanced.

1. **Introduction**-NN Structure

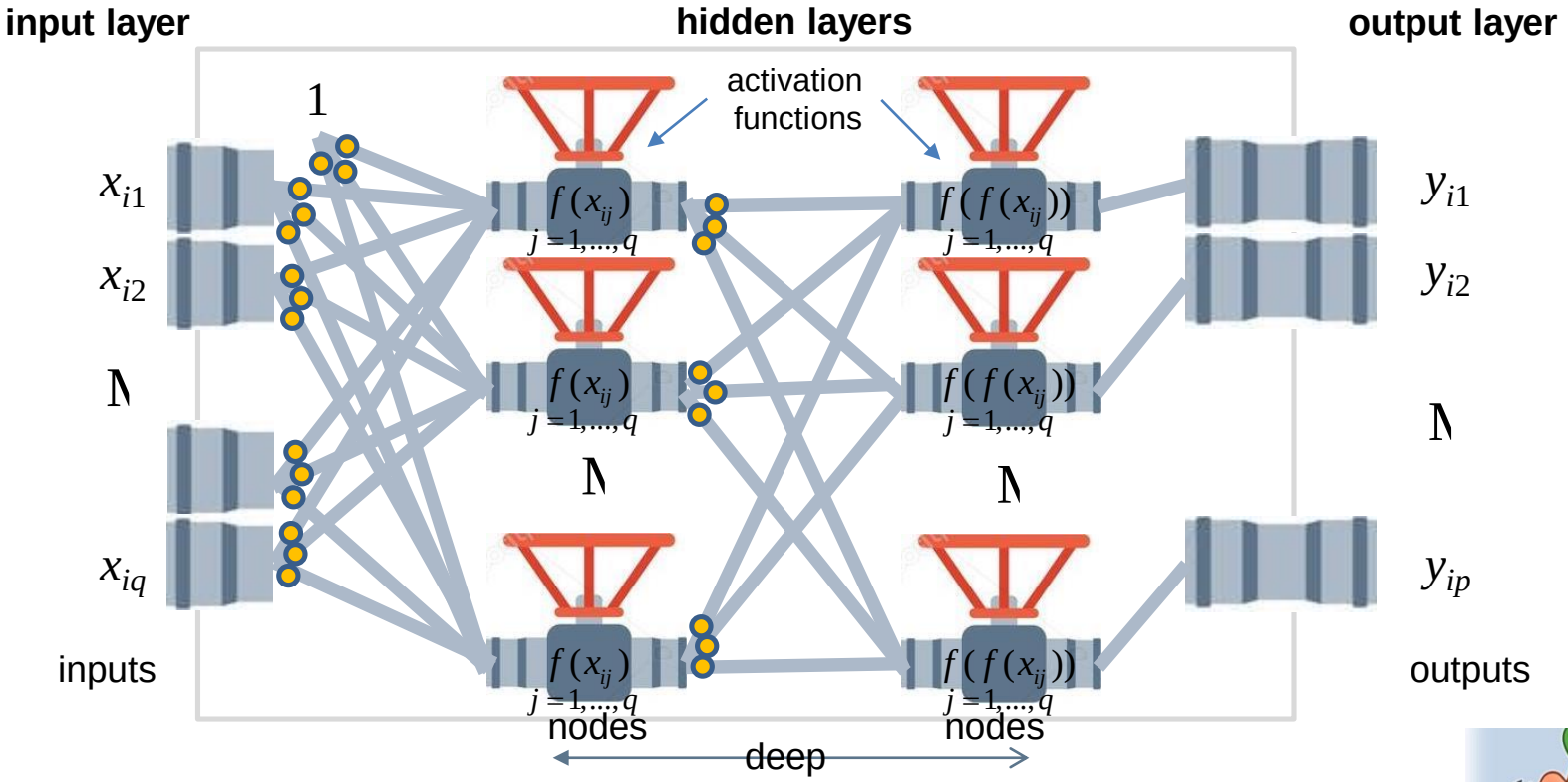
Multiple input and multiple output.

Computer science has taken a different approach.

That can be extremely complicated to understand.

1. **Introduction**-NN Structure

Multiple input and multiple output.

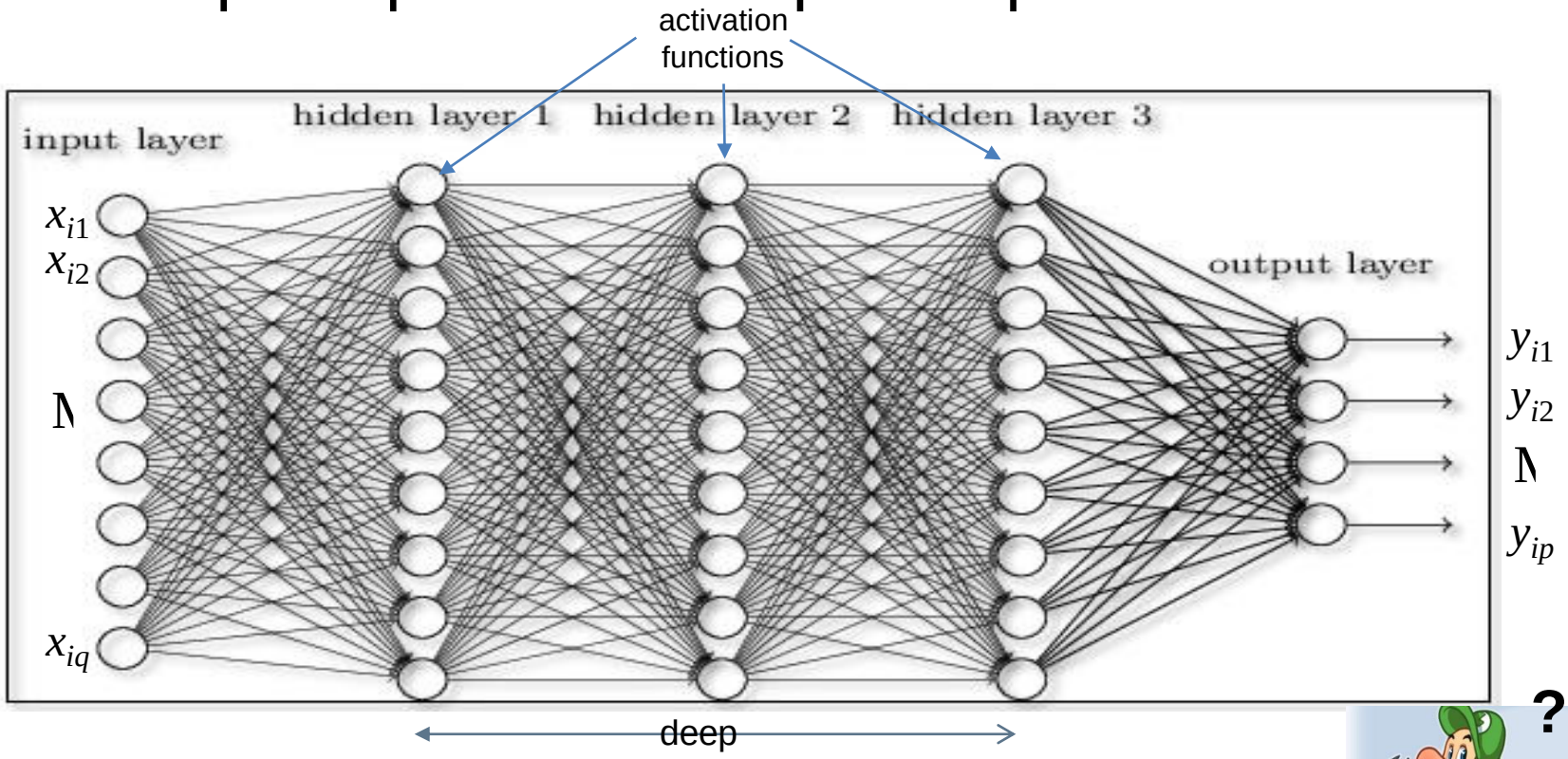


Many interconnected valves.

Functions of functions of … of inputs.

1. **Introduction**-NN Structure

Multiple input and multiple output.



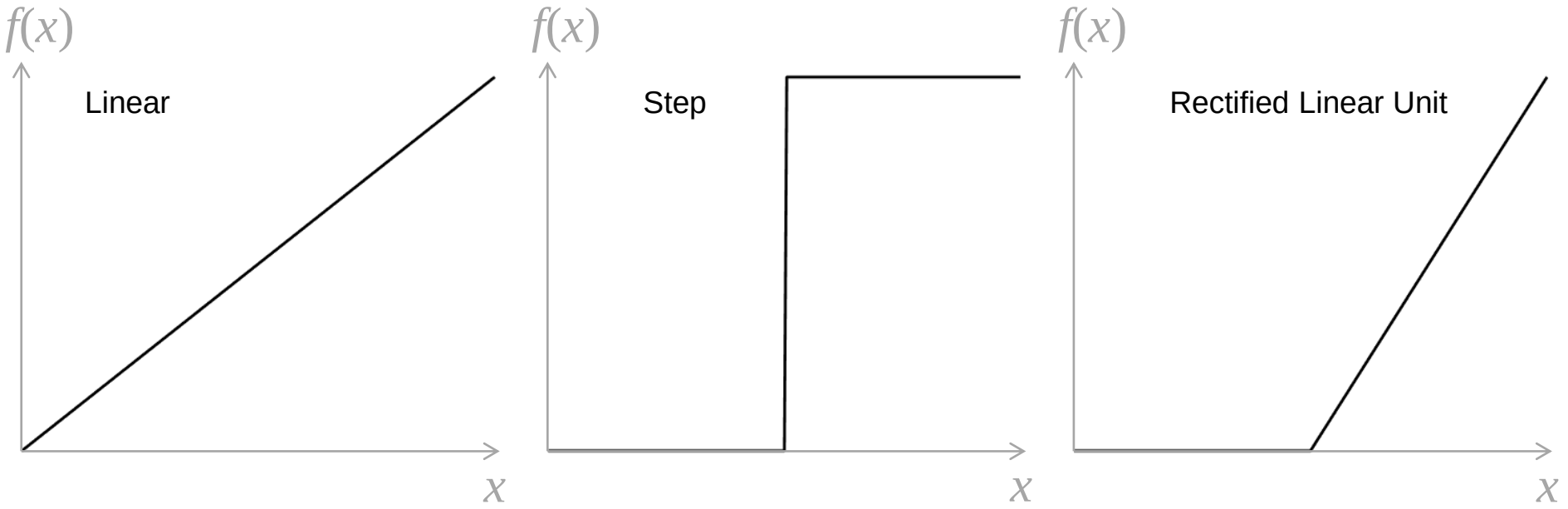
This gets very complicated very quickly.

https://www.houseofbots.com/news-detail/1442-1-what-is-deep-learning-and-neural-network

1. **Introduction**-Activation/Score Functions

There are many activation functions, $f(\cdot)$.

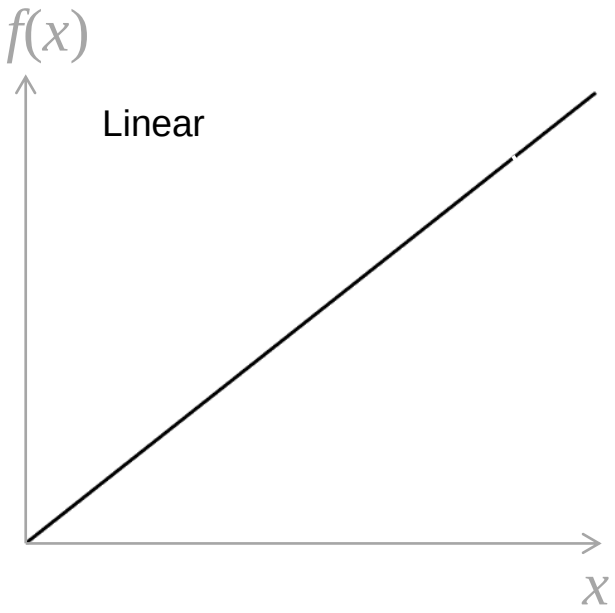
Motivated by neuronal representations.



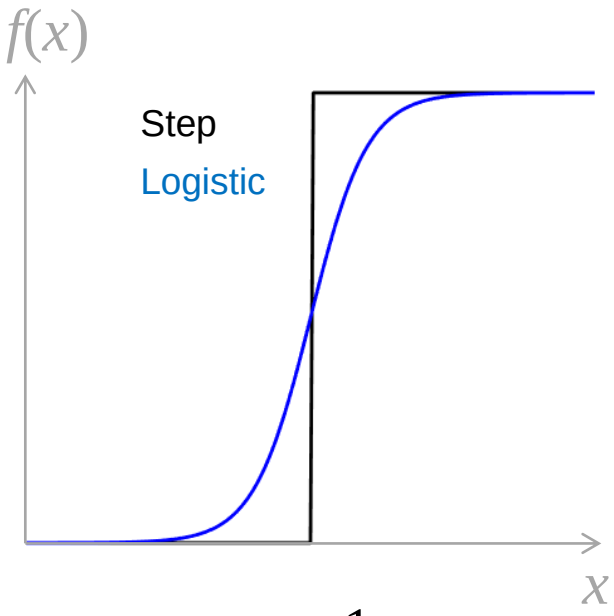
1. **Introduction**-Activation/Score Functions

Step and ReLU not differentiable for optimization.

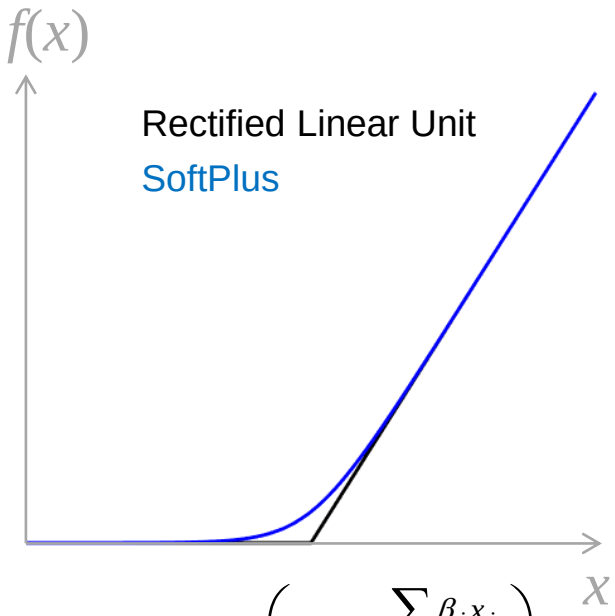
Smooth differentiable activation functions, $f(\cdot)$ used.



$$f(\cdot) = \sum_j \beta_j x_j$$



$$f(\cdot) = \frac{1}{1 + e^{-\sum_j \beta_j x_j}}$$



$$f(\cdot) = \ln\left(1 + e^{\sum_j \beta_j x_j}\right)$$

1. **Introduction**-Activation/Score Functions

NN parameters are estimated with "training" data $x_1,\ldots,x_n$ and a score function Q,

$$Q = \frac{1}{n}\sum_i [y_i - f(\sum_j \beta_j x_{ij})]^2$$

Least Squares/Normal Likelihood Score

$$Q = \sum_i y_i(\sum_j \beta_j x_{ij}) - \sum_i \ln[1+\exp(\sum_j \beta_j x_{ij})]$$

Logistic Regression/Bernoulli Likelihood Score

a function of the activation function $f(\cdot)$,

and $f(\cdot)$ is a linear combination of the x's.

$$f(\cdot) = \sum_j \beta_j x_j$$

Linear

$$f(\cdot) = \frac{1}{1+e^{-\sum_j \beta_j x_j}}$$

Logistic

$$f(\cdot) = \ln\left(1+e^{\sum_j \beta_j x_j}\right)$$

SoftPlus

Score function often motivated by probability theory!

1. **Introduction**-Parameter Estimation

Derivatives exist for smooth activation & score.

$$Q = \frac{1}{n}\sum_i [y_i - f(\cdot)]^2$$

Normal Likelihood Score

$$f(\cdot) = \sum_j \beta_j x_j$$

Linear

$$\frac{\partial Q}{\partial \beta_j} = \frac{2}{n}\sum_i [y_i - \sum_j \beta_j x_{ij}](-x_{ij})$$

Derivative

$$Q = \sum_i y_i(\sum_j \beta_j x_{ij}) - \sum_i \ln[1+\exp(\sum_j \beta_j x_{ij})]$$

Bernoulli Likelihood Score

$$f(\cdot) = \frac{1}{1+e^{-\sum_j \beta_j x_j}}$$

Logistic

$$\frac{\partial Q}{\partial \beta_j} = \sum_i x_{ij} y_i - \sum_i \frac{x_{ij}}{1+\exp(-\sum_j \beta_j x_{ij})}$$

Derivative

$$Q = \frac{1}{n}\sum_i [y_i - f(\cdot)]^2$$

Normal Likelihood Score

$$f(\cdot) = \ln(1+e^{\sum_j \beta_j x_j})$$

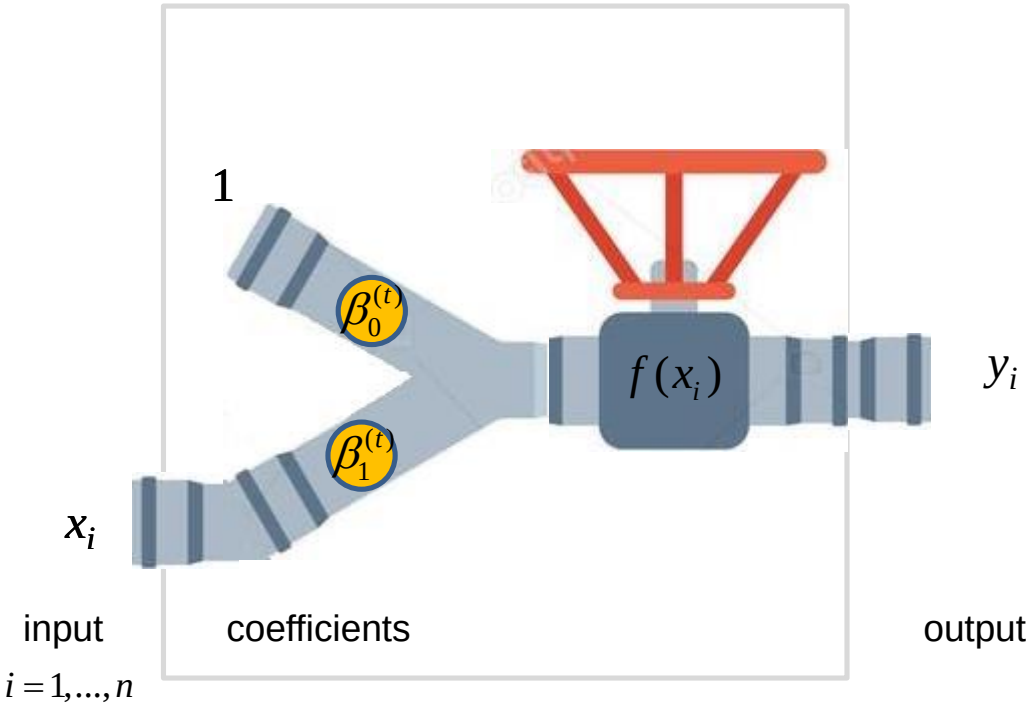
SoftPlus

$$\frac{\partial Q}{\partial \beta_j} = \frac{2}{n}\sum_i \left[\frac{x_{ij}[y_i - \ln(1+\exp(\sum_j \beta_j x_{ij}))]\exp(\sum_j \beta_j x_{ij})}{1+\exp(\sum_j \beta_j x_{ij})}\right]$$

Derivative

1. **Introduction**-Parameter Estimation

Iterative estimation, GD, NR, EM,…



1) Start with initial t=0 values $(\hat{\beta}_0^{(t)}, \hat{\beta}_1^{(t)})$

2) Run n data through

$$Q_i^{(t)} = [y_i - f(\sum_j \hat{\beta}_j^{(t)} x_{ij})]^2$$
$$i = 1,\ldots,n$$

3) Calculate score function

$$Q^{(t)} = \frac{1}{n}\sum_i [y_i - f(\sum_j \hat{\beta}_j^{(t)} x_{ij})]^2$$

4) Update coefficients GD

$(\hat{\beta}_0^{(t+1)}, \hat{\beta}_1^{(t+1)})$ ∇Q

5) Return to Step 2.

2. Linear Regression and NN-Simple

Often we believe that there is a linear relationship between an independent variable x, and a dependent variable y with measurement error.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad i = 1, \ldots, n$$

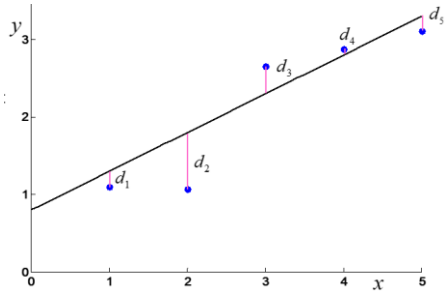
Could assume normal error or use least squares.

$$Q = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

2. **Linear Regression and NN**-Simple

We can estimate the "best" linear relationship between x and y using score function

$$Q = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$



by taking derivatives wrt β's

$$\frac{\partial Q}{\partial \beta_j} = -\frac{2}{n}\sum_i x_{ij}(y_i - \sum_j \beta_j x_{ij}) \quad , \qquad x_{i0} = 1 \; , \quad i = 1,\ldots,n \; , \quad j = 1,\ldots,q$$

written in vector form as

$$\nabla Q = -\frac{2}{n}(X'y - X'X\beta) \qquad y = (y_1,\ldots,y_n)'$$

$$\nabla Q = \left(\frac{\partial Q}{\partial \beta_j}\right)$$

setting equal to 0 and solving to get

$$\hat{\beta} = (X'X)^{-1}X'y \; .$$

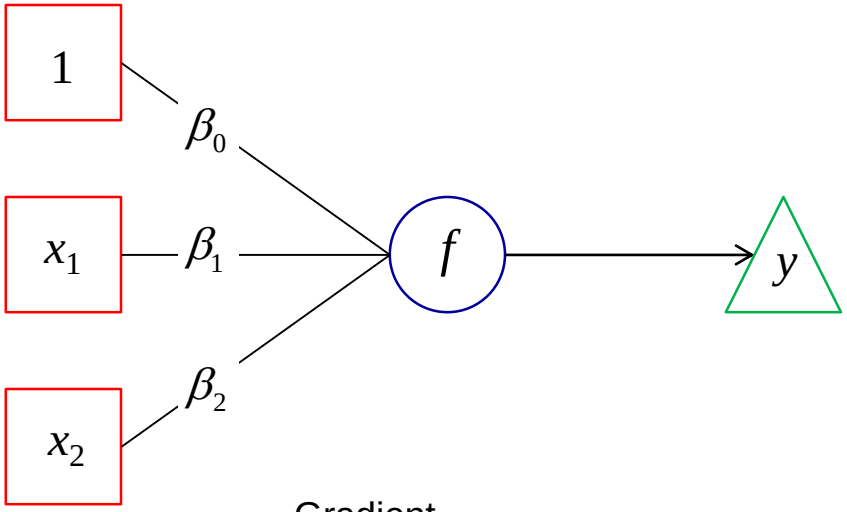
$$\beta = \begin{pmatrix} \beta_0 \\ \mathrm{M} \\ \beta_q \end{pmatrix}$$

2. **Linear Regression and NN**-Simple

A NN is a way to do multiple linear regression with linear activation & normal likelihood score function.

Linear Activation

$$f(\cdot) = \sum_j \beta_j x_j$$



$$\nabla Q = \begin{bmatrix} \frac{\partial Q}{\partial \beta_0} \\ \frac{\partial Q}{\partial \beta_1} \\ \frac{\partial Q}{\partial \beta_2} \end{bmatrix}$$

Normal Likelihood Score

$$Q = \frac{1}{n}\sum_i [y_i - f(\cdot)]^2$$

Derivatives

$$\frac{\partial Q}{\partial \beta_j} = -\frac{2}{n}\sum_i x_{ij}(y_i - \sum_j \beta_j x_{ij})$$

$$j = 0,1,2$$

Gradient

$$\nabla Q = -\frac{2}{n}(X'y - X'X\beta)$$

can set to 0 and get

$$\hat{\beta} = (X'X)^{-1}X'y$$

$$y = (y_1, \ldots, y_n)'$$

Gradient Descent

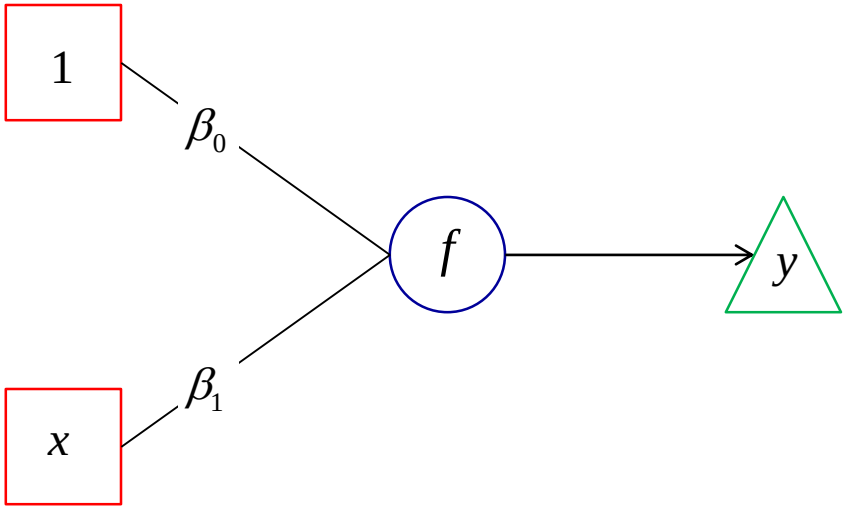
$$\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} - \gamma \nabla Q(\hat{\beta}^{(t)})$$

2. **Linear Regression and NN**-Simple

Example:

Given observed data:

$$f(\cdot) = \beta_0 + \beta_1 x$$



(1,1.4)
(2,2.3)
(3,1.7)
(4,3.0)
(5,3.4)

use the NN structure and GD

to iteratively estimate the parameters.

$$\gamma = .0001$$

$$Q = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \qquad \nabla Q = -\frac{2}{n}(X'y - X'X\beta) \qquad \hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} - \gamma \nabla Q(\hat{\beta}^{(t)})$$

$$y = (y_1, \ldots, y_n)'$$

2. Linear Regression and NN-Simple

Example:

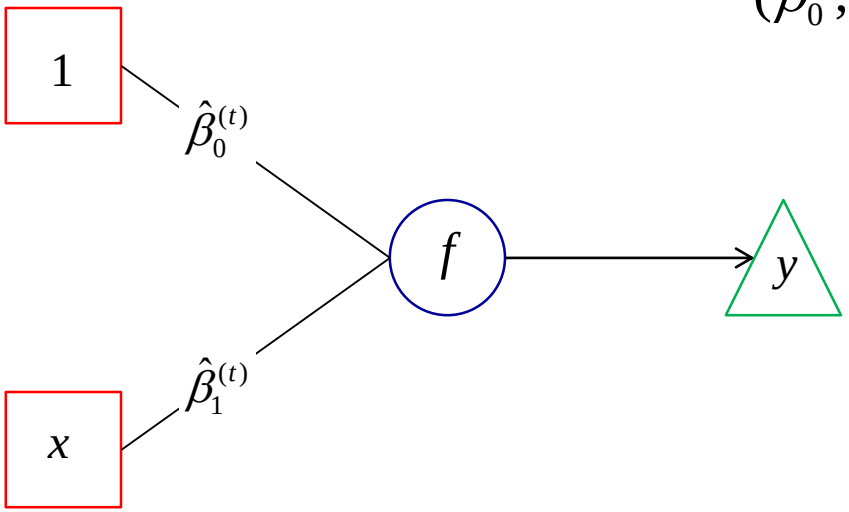
Given observed data:

True Values

$(\beta_0, \beta_1) = (0.80, 0.50)$

(1,1.4)
(2,2.3)
(3,1.7)
(4,3.0)
(5,3.4)

$f(\cdot) = \beta_0 + \beta_1 x$



t=0

$(\hat{\beta}_0^{(0)}, \hat{\beta}_1^{(0)}) = (1.50, 1.00)$

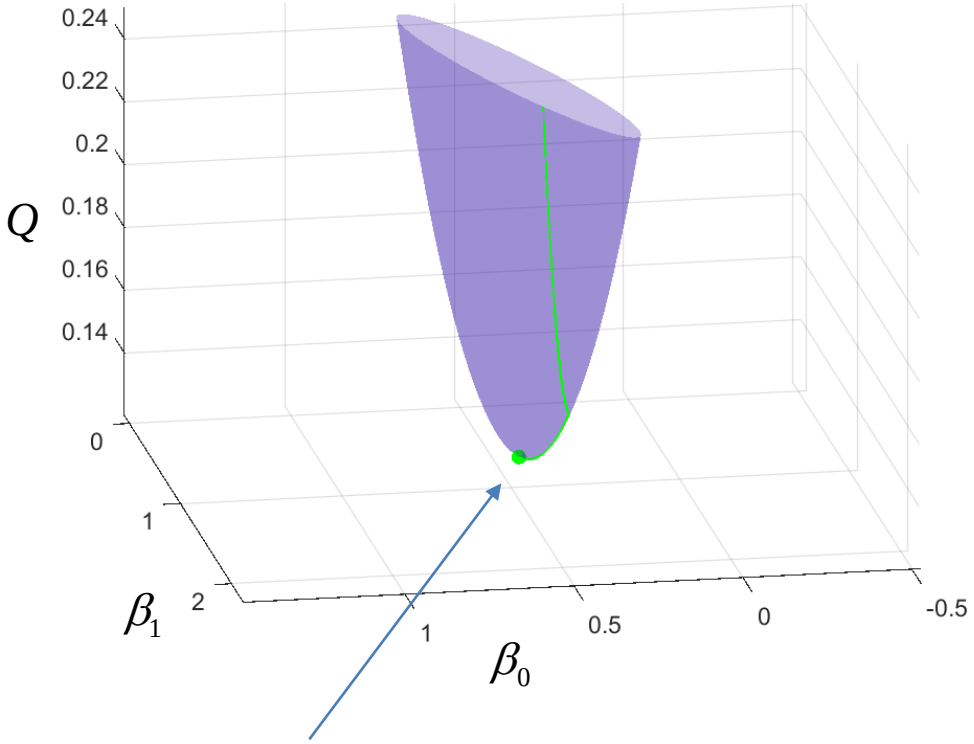
→Run data through with $\hat{\beta}^{(t)} = (\hat{\beta}_0^{(t)}, \hat{\beta}_1^{(t)})'$

Calculate $\nabla Q(\hat{\beta}^{(t)}) = -\frac{2}{n}(X'y - X'X\hat{\beta}^{(t)})$, $\gamma = .0001$

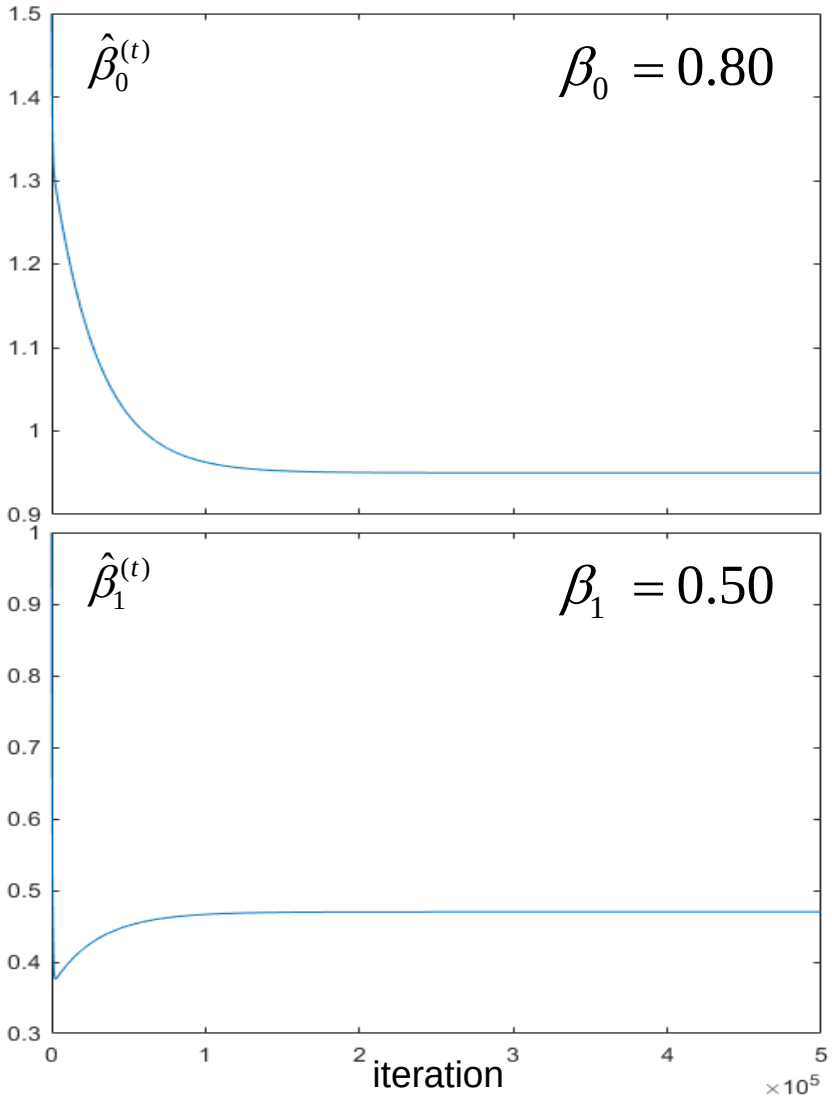
Calculate new $\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} - \gamma \nabla Q(\hat{\beta}^{(t)})$, $t = t+1$

2. Linear Regression and NN-Simple

Example:



$$\hat{\beta} = \begin{bmatrix} 0.95 \\ 0.47 \end{bmatrix} \qquad Q = 0.1286$$



2. **Linear Regression and NN**-Multivariate

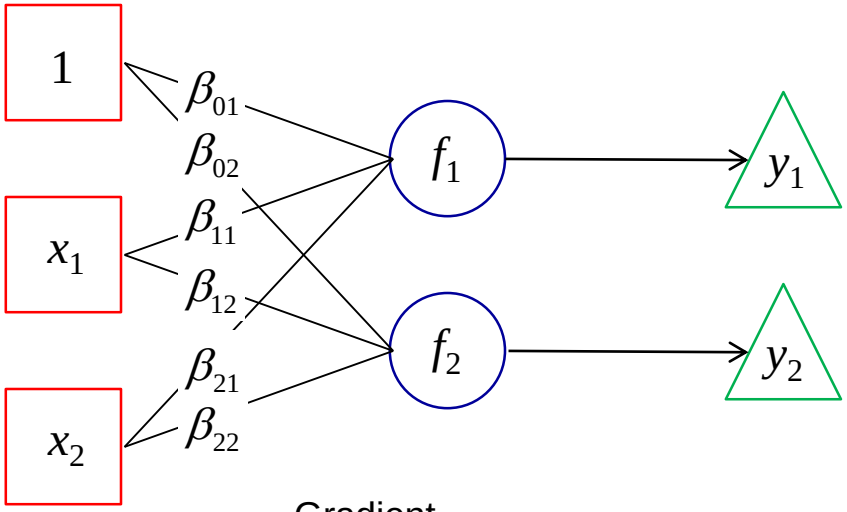
A NN is a way to do multivariate linear regression w/ linear activation and normal likelihood score function.

Linear Activation

$$f_k(\cdot) = \sum_j \beta_{jk} x_j$$

Normal Likelihood Score

$$Q_k = \frac{1}{n}\sum_i [y_{ik} - f_k(\cdot)]^2$$



$$\nabla Q_k = \begin{bmatrix} \frac{\partial Q_k}{\partial \beta_{0k}} \\ \frac{\partial Q_k}{\partial \beta_{1k}} \\ \frac{\partial Q_k}{\partial \beta_{2k}} \end{bmatrix}$$

Derivatives

$$\frac{\partial Q_k}{\partial \beta_{jk}} = -\frac{2}{n}\sum_i x_{ij}(y_{ik} - \sum_j \beta_{jk} x_{ij})$$

$j = 0,1,2 \quad k = 1,2$

Gradient

$$\nabla Q_k = -\frac{2}{n}(X'y_k - X'X\beta_k) \qquad k = 1,2$$

can set to 0 and get

$$\hat{\beta}_k = (X'X)^{-1}X'y_k$$

$$y_k = (y_{1k},\ldots,y_{nk})'$$

Gradient Descent

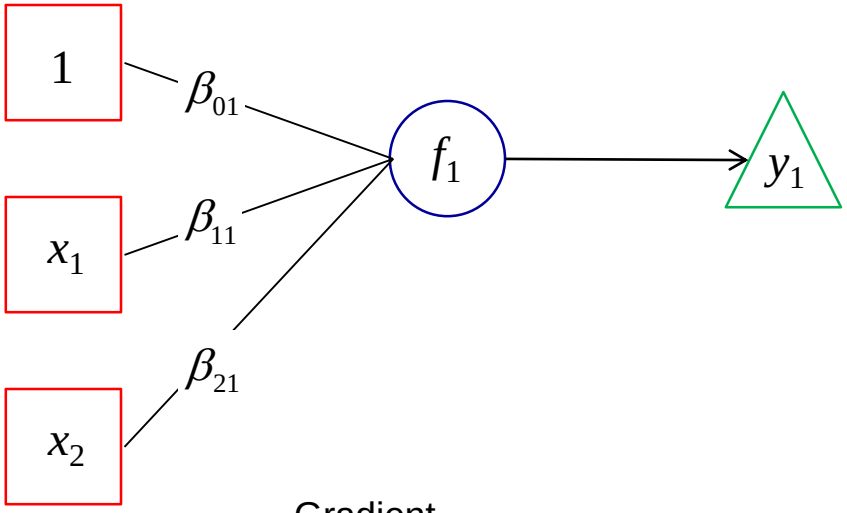
$$\hat{\beta}_k^{(t+1)} = \hat{\beta}_k^{(t)} - \gamma \nabla Q_k(\hat{\beta}_k^{(t)})$$

2. **Linear Regression and NN**-Multivariate

A NN is a way to do multivariate linear regression w/ linear activation and normal likelihood score function.

Linear Activation

$$f_1(\cdot) = \sum_j \beta_{j1} x_j$$



Normal Likelihood Score

$$Q_1 = \frac{1}{n}\sum_i [y_{i1} - f_1(\cdot)]^2$$

$$\nabla Q_1 = \begin{bmatrix} \frac{\partial Q_1}{\partial \beta_{01}} \\ \frac{\partial Q_1}{\partial \beta_{11}} \\ \frac{\partial Q_1}{\partial \beta_{21}} \end{bmatrix}$$

Derivatives

$$\frac{\partial Q_1}{\partial \beta_{j1}} = -\frac{2}{n}\sum_i x_{ij}(y_{i1} - \sum_j \beta_{j1} x_{ij})$$

$$j = 0,1,2$$

Gradient

$$\nabla Q_1 = -\frac{2}{n}(X'y_1 - X'X\beta_1)$$

can set to 0 and get

$$\hat{\beta}_1 = (X'X)^{-1}X'y_1$$

$$y_1 = (y_{11},\ldots,y_{n1})'$$

Gradient Descent

$$\hat{\beta}_1^{(t+1)} = \hat{\beta}_1^{(t)} - \gamma \nabla Q_1(\hat{\beta}_1^{(t)})$$

2. **Linear Regression and NN**-Multivariate

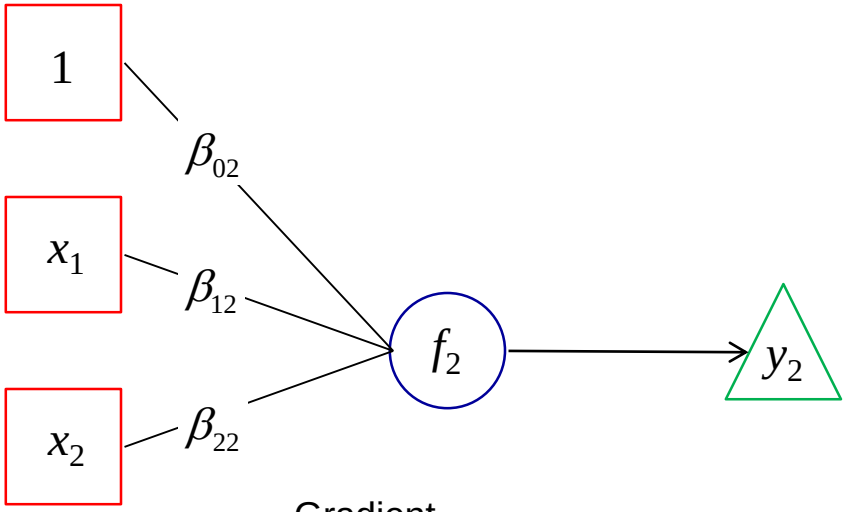
A NN is a way to do multivariate linear regression w/ linear activation and normal likelihood score function.

Linear Activation

$$f_2(\cdot) = \sum_j \beta_{j2} x_j$$

Normal Likelihood Score

$$Q_2 = \frac{1}{n}\sum_i [y_{i2} - f(\cdot)]^2$$



$$\nabla Q_2 = \begin{bmatrix} \frac{\partial Q_2}{\partial \beta_{02}} \\ \frac{\partial Q_2}{\partial \beta_{12}} \\ \frac{\partial Q_2}{\partial \beta_{22}} \end{bmatrix}$$

Derivatives

$$\frac{\partial Q_2}{\partial \beta_{j2}} = -\frac{2}{n}\sum_i x_{ij}(y_{i2} - \sum_j \beta_{j2} x_{ij})$$

$$j = 0,1,2$$

Gradient

$$\nabla Q_2 = -\frac{2}{n}(X'y_2 - X'X\beta_2)$$

can set to 0 and get

$$\hat{\beta}_2 = (X'X)^{-1}X'y_2$$

$$y_2 = (y_{12},\ldots,y_{n2})'$$

Gradient Descent

$$\hat{\beta}_2^{(t+1)} = \hat{\beta}_2^{(t)} - \gamma \nabla Q_2(\hat{\beta}_2^{(t)})$$

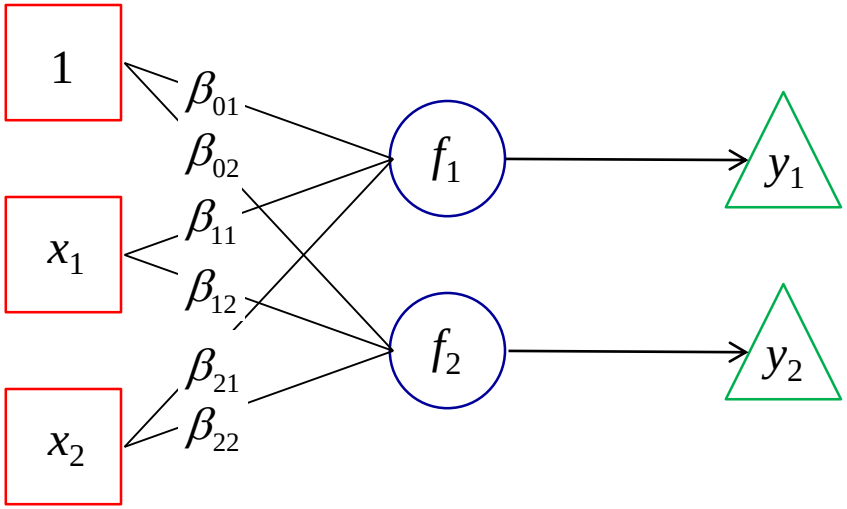
2. Linear Regression and NN-Multivariate

Two independent parallel regressions yields same result as simultaneous multivariate regressions

$$\hat{B} = (X'X)^{-1}X'Y$$

$$\hat{B} = (\hat{\beta}_1, \hat{\beta}_2)$$

$$Y = (y_1, y_2)$$



$$\hat{\beta}_1 = (X'X)^{-1}X'y_1$$

$$\hat{\beta}_2 = (X'X)^{-1}X'y_2$$

$$y_1 = (y_{11}, \ldots, y_{n1})'$$

$$y_2 = (y_{12}, \ldots, y_{n2})'$$

$$\hat{\beta}_1^{(t+1)} = \hat{\beta}_1^{(t)} - \gamma \nabla Q_1(\hat{\beta}_1^{(t)})$$

$$\hat{\beta}_2^{(t+1)} = \hat{\beta}_2^{(t)} - \gamma \nabla Q_2(\hat{\beta}_2^{(t)})$$

3. Logistic Regression and NN-Simple

Often the probability p of an event E depends upon an independent variable x, such as the probability p of getting an A on the final in my class depends on the number of hours that a student studies x.

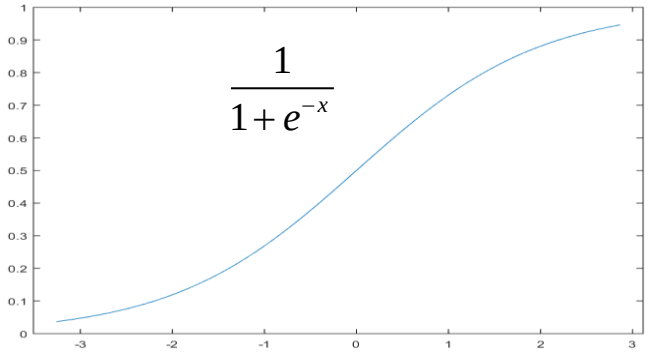
So p is a function of x, $p(x)$. $0 \leq p(x) \leq 1$

Hours (x)	A (y)
6	0
8	0
10	0
12	0
14	0
16	1
18	0
20	0
22	0
24	0
26	1
28	0
30	0
32	1
34	1
36	1
38	1
40	1

3. Logistic Regression and NN-Simple

This dependency of a probability $p(x), 0 \le p(x) \le 1$ on an independent variable x, $-\infty \le x \le \infty$, is generally described through the logistic mapping function

$$p = p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$



If the event E occurs, then we say y=1 and if not y=0.

$P(y=1)=p$ and $P(y=0)=1-p$. … This is a Bernoulli trial.

3. Logistic Regression and NN-Simple

The likelihood function

$$L(\beta_0,\beta_1)=\prod_{i=1}^{n}[p(x_i)]^{y_i}[1-p(x_i)]^{1-y_i}$$

$$y_i=\{0,1\} \quad 0\le p(x_i)\le 1$$

$$-\infty \le x_i \le \infty$$

where $p(x_i)=\dfrac{1}{1+e^{-\beta_0-\beta_1 x_i}}$

has log likelihood function

$$\begin{aligned}
LL(\beta_0,\beta_1) &= \sum_{i=1}^{n}\left[y_i \ln[p(x_i)]+(1-y_i)\ln[1-p(x_i)]\right] \\
&= \sum_{i=1}^{n} y_i \ln[p(x_i)]+\sum_{i=1}^{n}(1-y_i)\ln[1-p(x_i)] \\
&= \sum_{i=1}^{n}\ln[1-p(x_i)]+\sum_{i=1}^{n} y_i \ln[p(x_i)/(1-p(x_i))] \\
&= \sum_{i=1}^{n} y_i[\beta_0+\beta_1 x_i]-\sum_{i=1}^{n}\ln[1+e^{\beta_0+\beta_1 x_i}] \quad .
\end{aligned}$$

3. **Logistic Regression and NN**-Simple

We can estimate the "best" logistic relationship between x and $0/1$ y using score function

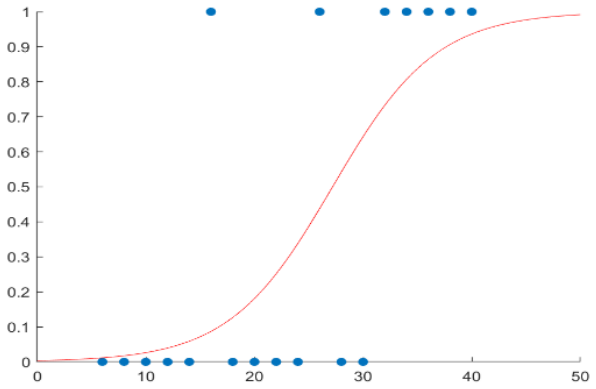
$$Q = \sum_i y_i (\sum_j \beta_j x_{ij}) - \sum_i \ln[1 + \exp(\sum_j \beta_j x_{ij})]$$

by taking derivatives wrt β's

$$\frac{\partial Q}{\partial \beta_j} = \sum_i x_{ij} y_i - \sum_i \frac{x_{ij}}{1 + \exp(-\sum_j \beta_j x_{ij})}, \quad x_{i0} = 1, \quad i = 1,\ldots,n, \quad j = 1,\ldots,q$$

with no closed form solution.

$$f(\cdot) = \frac{1}{1 + \exp(-\sum_j \beta_j x_j)}$$

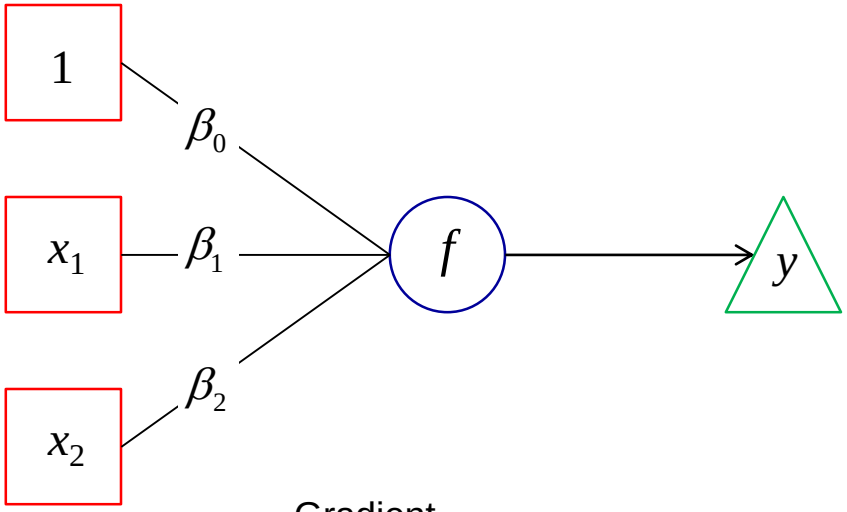


3. **Logistic Regression and NN**-Multiple

A NN is a way to do multiple logistic regression with logistic activation & Bernoulli likelihood score function.

Logistic Activation

$$f(\cdot) = \frac{1}{1+\exp(-\sum_j \beta_j x_j)}$$



Bernoulli Likelihood Score

$$Q = \sum_i y_i (\sum_j \beta_j x_{ij}) - \sum_i \ln[1+\exp(\sum_j \beta_j x_{ij})]$$

Derivatives

$$\frac{\partial Q}{\partial \beta_j} = \sum_i x_{ij} y_i - \sum_i \frac{x_{ij}}{1+\exp(-\sum_j \beta_j x_{ij})}$$

$$j = 0,1,2$$

Gradient

$$\nabla Q = \left(\frac{\partial Q}{\partial \beta_j} \right)$$

$$\nabla Q = \begin{bmatrix} \frac{\partial Q}{\partial \beta_0} \\ \frac{\partial Q}{\partial \beta_1} \\ \frac{\partial Q}{\partial \beta_2} \end{bmatrix}$$

Gradient Descent

$$\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} + \gamma \nabla Q(\hat{\beta}^{(t)})$$

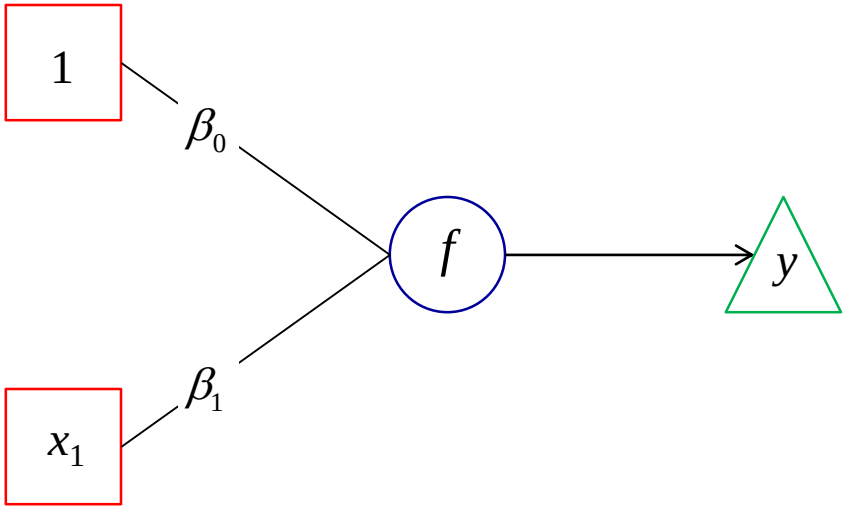
3. **Logistic Regression and NN**-Simple Example

Simple Logistic Regression

Given observed data:

Hours (x)	A (y)
6	0
8	0
10	0
12	0
14	0
16	1
18	0
20	0
22	0
24	0
26	1
28	0
30	0
32	1
34	1
36	1
38	1
40	1

$$f(\cdot) = \frac{1}{1+\exp(-\sum_j \beta_j x_j)}$$



use the NN structure and GD

to iteratively estimate the parameters.

$$\nabla Q = \left(\frac{\partial Q}{\partial \beta_j} \right)$$

$$\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} - \gamma \nabla Q(\hat{\beta}^{(t)})$$

3. Logistic Regression and NN-Simple Example

Simple Logistic Regression

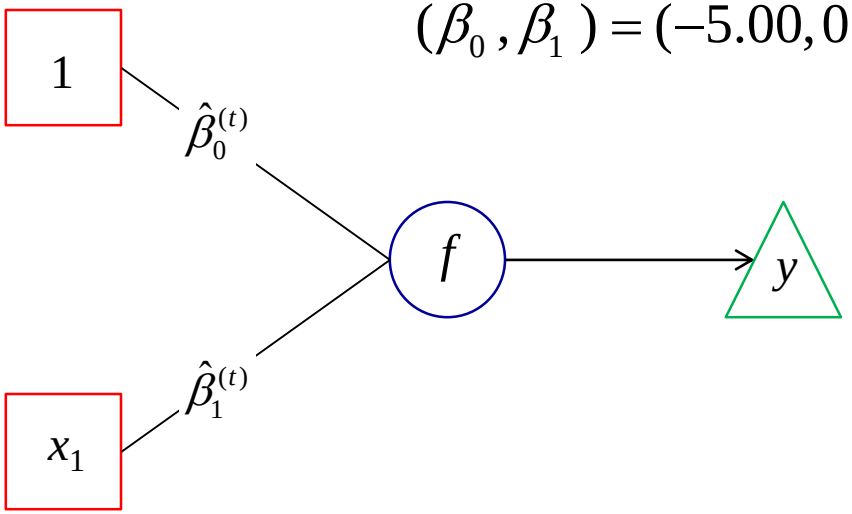
Given observed data:

Hours (x)	A (y)
6	0
8	0
10	0
12	0
14	0
16	1
18	0
20	0
22	0
24	0
26	1
28	0
30	0
32	1
34	1
36	1
38	1
40	1

True Values

$$(\beta_0, \beta_1) = (-5.00, 0.20)$$

$$f(\cdot) = \frac{1}{1+\exp(-\sum_j \beta_j x_j)}$$



t=0

$$(\hat{\beta}_0^{(0)}, \hat{\beta}_1^{(0)}) = (3.00, 0.50)$$

$\gamma = .001$

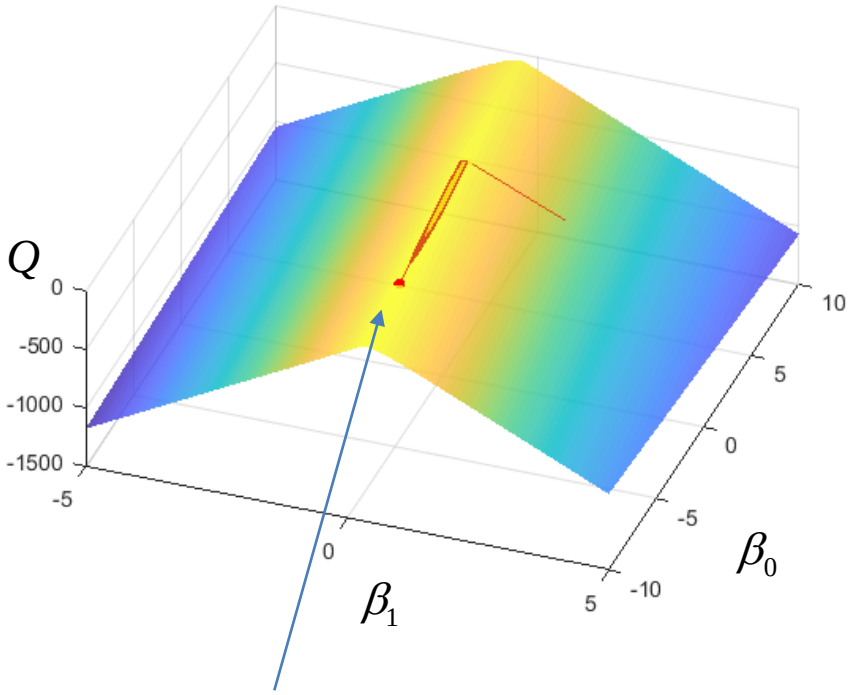
→Run data through with $\hat{\beta}^{(t)} = (\hat{\beta}_0^{(t)}, \hat{\beta}_1^{(t)})'$

Calculate $\nabla Q(\hat{\beta}^{(t)}) = \left[\sum_i x_{ij} y_i - \sum_i \frac{x_{ij}}{1+\exp(-\sum_j \beta_j x_{ij})}\right]$,

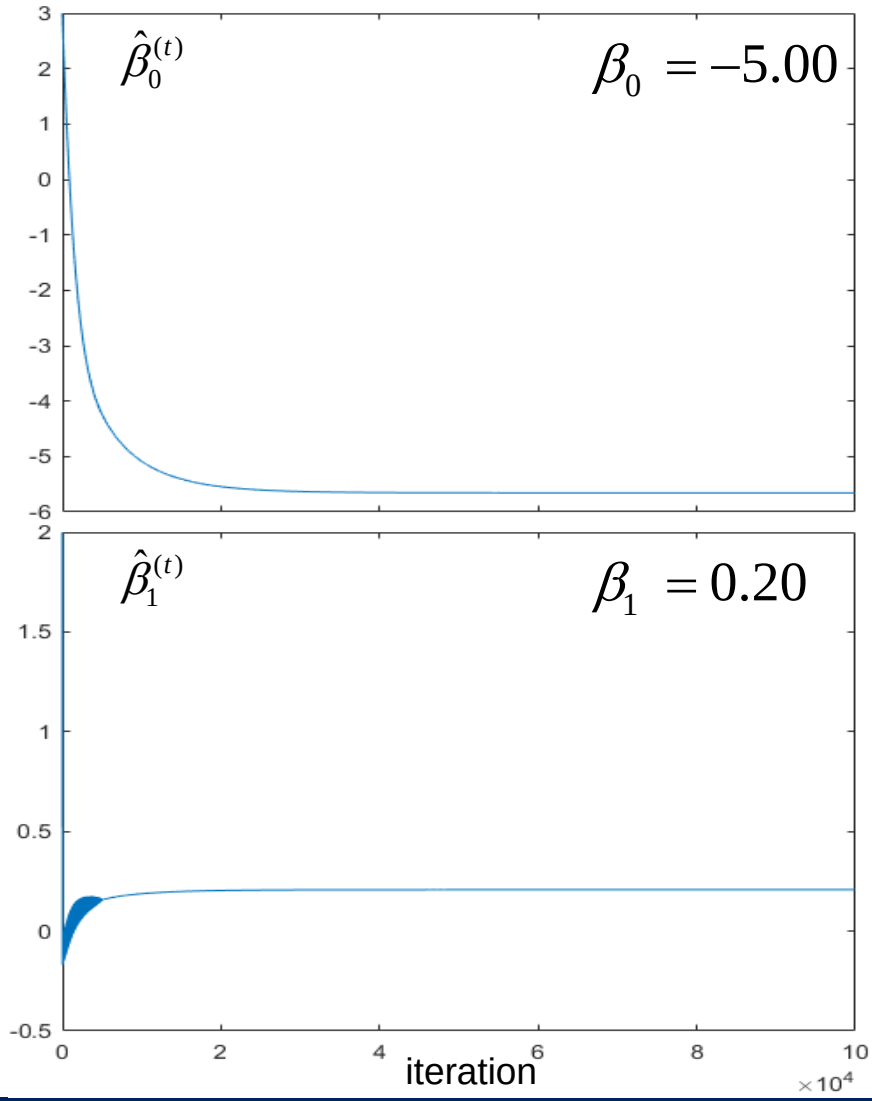
Calculate new $\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} - \gamma \nabla Q(\hat{\beta}^{(t)})$, ^{j} $t=t+1$

3. Logistic Regression and NN-Simple Example

Simple Logitic Results

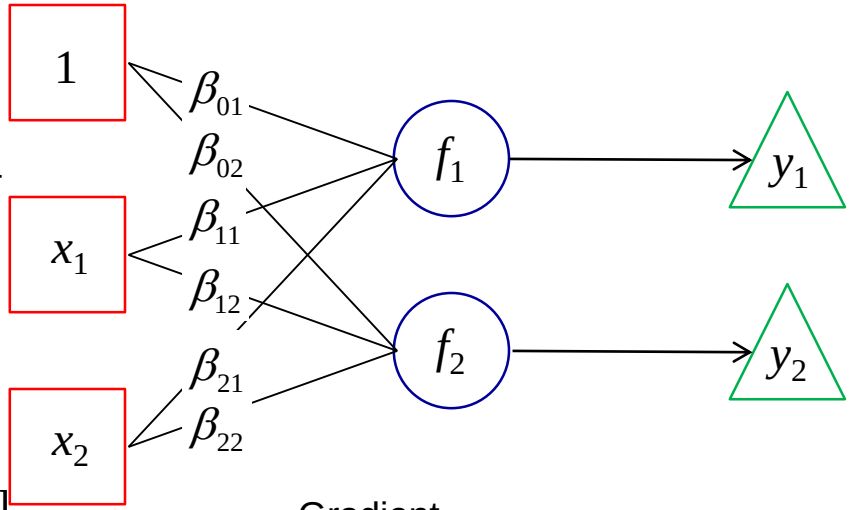


$$\hat{\beta} = \begin{bmatrix} -5.65 \\ 0.21 \end{bmatrix}$$



3. **Logistic Regression and NN**-Multivariate

A NN is a way to do multivariate logistic regression w/ logistic activation & Bernoulli likelihood score function.



Logistic Activation

$$f_k(\cdot) = \frac{1}{1+\exp(-\sum_j \beta_{jk} x_j)}$$

Bernoulli Likelihood Score

$$Q_k = \sum_i y_{ik}(\sum_j \beta_{jk} x_{ij}) - \sum_i \ln[1+\exp(\sum_j \beta_{jk} x_{ij})]$$

Derivatives

$$\frac{\partial Q_k}{\partial \beta_{jk}} = \sum_i x_{ij} y_{ik} - \sum_i \frac{x_{ij}}{1+\exp(-\sum_j \beta_{jk} x_{ij})}$$

$j = 0,1,2 \quad k = 1,2$

Gradient

$$\nabla Q_k = \left(\frac{\partial Q_k}{\partial \beta_{jk}} \right)$$

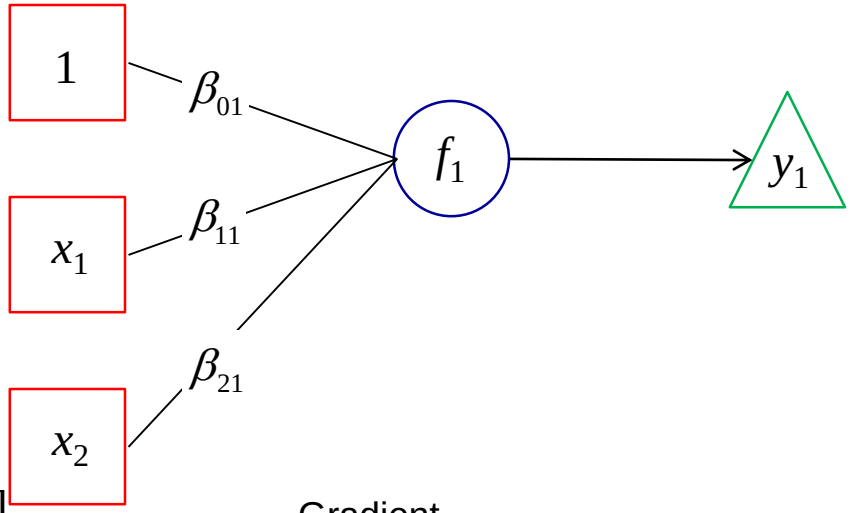
$$\nabla Q_k = \begin{bmatrix} \frac{\partial Q_k}{\partial \beta_{0k}} \\ \frac{\partial Q_k}{\partial \beta_{1k}} \\ \frac{\partial Q_k}{\partial \beta_{2k}} \end{bmatrix}$$

Gradient Descent

$$\hat{\beta}_k^{(t+1)} = \hat{\beta}_k^{(t+1)} + \gamma \nabla Q_k(\hat{\beta}_k^{(t+1)})$$

3. **Logistic Regression and NN**-Multivariate

A NN is a way to do multivariate logistic regression w/ logistic activation & Bernoulli likelihood score function.



Logistic Activation

$$f_1(\cdot) = \frac{1}{1+\exp(-\sum_j \beta_{j1} x_j)}$$

Bernoulli Likelihood Score

$$Q_1 = \sum_i y_{i1}(\sum_j \beta_{j1} x_{ij}) - \sum_i \ln[1+\exp(\sum_j \beta_{j1} x_{ij})]$$

Derivatives

$$\frac{\partial Q_1}{\partial \beta_{j1}} = \sum_i x_{ij} y_{i1} - \sum_i \frac{x_{ij}}{1+\exp(-\sum_j \beta_{j1} x_{ij})}$$

$$j = 0,1,2$$

Gradient

$$\nabla Q_1 = \left(\frac{\partial Q_1}{\partial \beta_{j1}} \right)$$

$$\nabla Q_1 = \begin{bmatrix} \frac{\partial Q_1}{\partial \beta_{01}} \\ \frac{\partial Q_1}{\partial \beta_{11}} \\ \frac{\partial Q_1}{\partial \beta_{21}} \end{bmatrix}$$

Gradient Descent

$$\hat{\beta}_1^{(t+1)} = \hat{\beta}_1^{(t+1)} + \gamma \nabla Q_1(\hat{\beta}_1^{(t+1)})$$

3. **Logistic Regression and NN**-Multivariate

A NN is a way to do multivariate linear regression w/ logistic activation & Bernoulli likelihood score function.

Logistic Activation

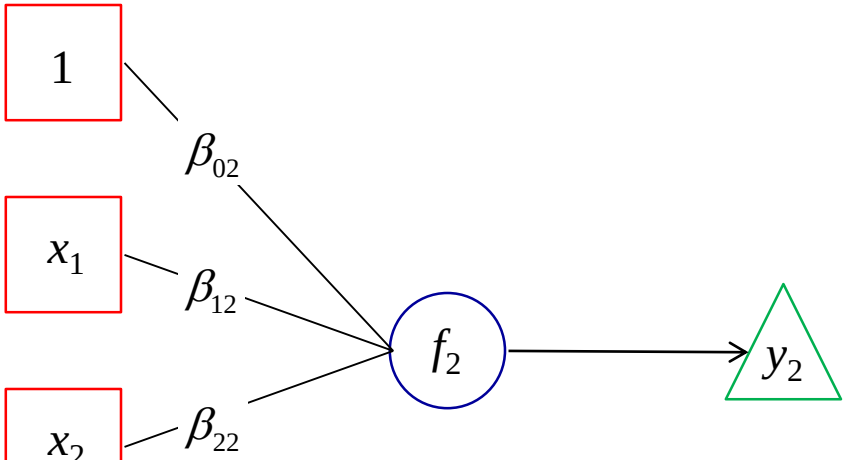
$$f_2(\cdot) = \frac{1}{1+\exp(-\sum_j \beta_{j2} x_j)}$$

Bernoulli Likelihood Score

$$Q_2 = \sum_i y_{i2}(\sum_j \beta_{j2} x_{ij}) - \sum_i \ln[1+\exp(\sum_j \beta_{j2} x_{ij})]$$

Derivatives

$$\frac{\partial Q_2}{\partial \beta_{j2}} = \sum_i x_{ij} y_{i2} - \sum_i \frac{x_{ij}}{1+\exp(-\sum_j \beta_{j2} x_{ij})} \quad j = 0,1,2$$



Gradient

$$\nabla Q_2 = \left(\frac{\partial Q_2}{\partial \beta_{j2}} \right)$$

$$\nabla Q_2 = \begin{bmatrix} \frac{\partial Q_2}{\partial \beta_{02}} \\ \frac{\partial Q_2}{\partial \beta_{12}} \\ \frac{\partial Q_2}{\partial \beta_{22}} \end{bmatrix}$$

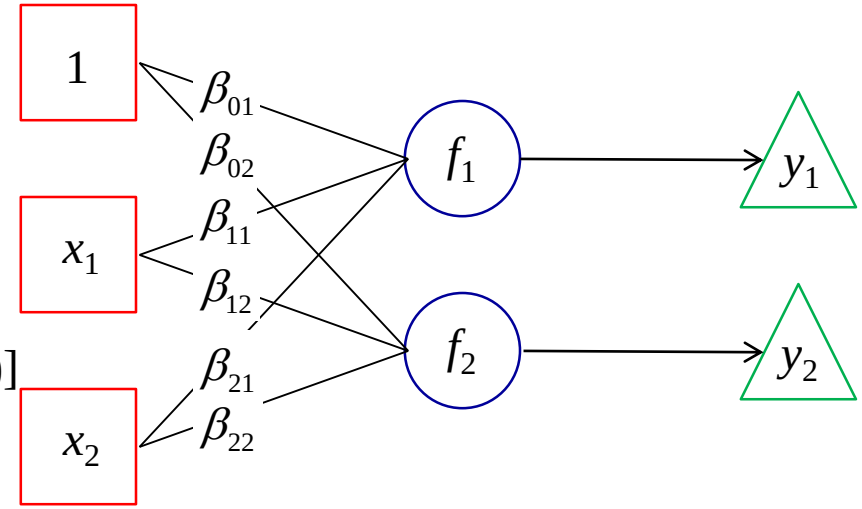
Gradient Descent

$$\hat{\beta}_2^{(t+1)} = \hat{\beta}_2^{(t+1)} + \gamma \nabla Q_2(\hat{\beta}_2^{(t+1)})$$

3. Logistic Regression and NN-Multivariate

Two independent parallel regressions yields same result as simultaneous multivariate regressions.

$$\hat{B} = (\hat{\beta}_1, \hat{\beta}_2)$$



Maximizing Likelihoods

$$Q_1 = \sum_i y_{i1}(\sum_j \beta_{j1} x_{ij}) - \sum_i \ln[1 + \exp(\sum_j \beta_{j1} x_{ij})]$$

$$Q_2 = \sum_i y_{i2}(\sum_j \beta_{j2} x_{ij}) - \sum_i \ln[1 + \exp(\sum_j \beta_{j2} x_{ij})]$$

Independently Via Gradient Descent

$$\hat{\beta}_1^{(t+1)} = \hat{\beta}_1^{(t)} + \gamma \nabla Q_1(\hat{\beta}_1^{(t)})$$

$$\hat{\beta}_2^{(t+1)} = \hat{\beta}_2^{(t)} + \gamma \nabla Q_2(\hat{\beta}_2^{(t)})$$

4. Non-Linear Regression and NN-Simple

We might believe that there is a non-linear

relationship between an independent variable x,

and a dependent variable y with measurement error.

$$y_i = f(x_i) + \varepsilon_i \qquad i = 1,\ldots,n$$

Could assume normal error or use least squares.

$$Q = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

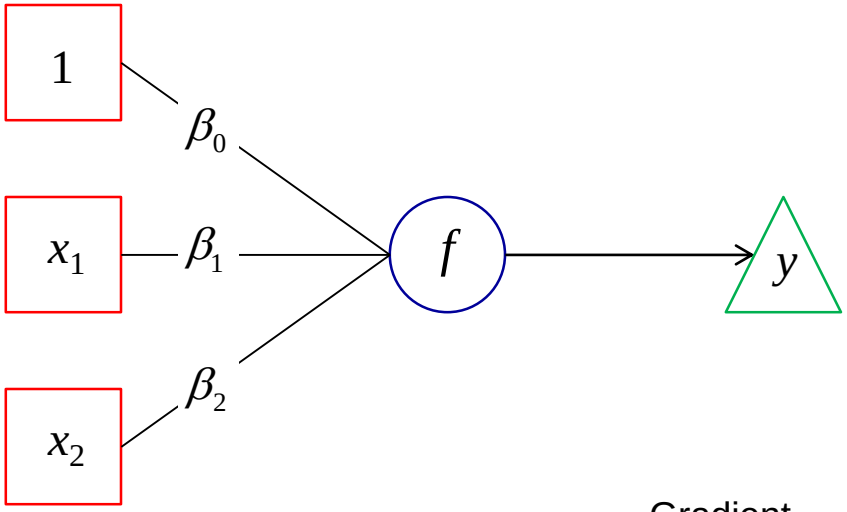
As an example consider the Softplus $f(\cdot) = \ln(1 + e^{\sum_j \beta_j x_j})$.

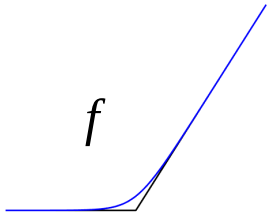
4. Non-Linear Regression and NN-Simple

A NN is also a way to do non-linear regression with softplus activation & normal likelihood score function.

Softplus Activation

$$f(\cdot) = \ln(1 + e^{\sum_j \beta_j x_j})$$





Normal Likelihood Score

$$Q = \frac{1}{n}\sum_i [y_i - f(\cdot)]^2$$

Derivatives

$$\frac{\partial Q}{\partial \beta_j} = \frac{2}{n}\sum_i \left[\frac{x_{ij}[y_i - \ln(1+\exp(\sum_j \beta_j x_{ij}))]\exp(\sum_j \beta_j x_{ij})}{1+\exp(\sum_j \beta_j x_{ij})} \right]$$

$$j = 0, 1, 2$$

Gradient

$$\nabla Q = \left(\frac{\partial Q}{\partial \beta_j} \right)$$

$$\nabla Q = \begin{bmatrix} \frac{\partial Q}{\partial \beta_0} \\ \frac{\partial Q}{\partial \beta_1} \\ \frac{\partial Q}{\partial \beta_2} \end{bmatrix}$$

Gradient Descent

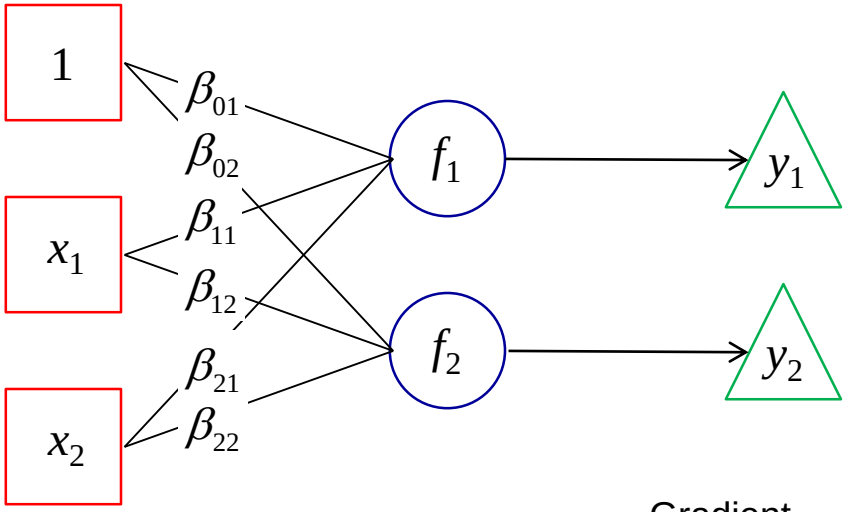
$$\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} - \gamma \nabla Q(\hat{\beta}^{(t)})$$

4. NonLinear Regression and NN-Multivariate

These univariate non-linear regressions can be put together similarly to do multivariate regression

Softplus Activation

$$f_k(\cdot) = \ln(1 + e^{\sum_j \beta_j x_j})$$



Normal Likelihood Score

$$Q_k = \frac{1}{n}\sum_i [y_{ik} - f_k(\cdot)]^2$$

Derivatives

$$\frac{\partial Q_k}{\partial \beta_{jk}} = \frac{2}{n}\sum_i \left[\frac{x_{ij}[y_{ik} - \ln(1+\exp(\sum_j \beta_{jk} x_{ij}))]\exp(\sum_j \beta_{jk} x_{ij})}{1+\exp(\sum_j \beta_{jk} x_{ij})} \right]$$

$j = 0,1,2 \quad k = 1,2$

Gradient

$$\nabla Q_k = \left(\frac{\partial Q_k}{\partial \beta_{jk}} \right)$$

$$\nabla Q_k = \begin{bmatrix} \frac{\partial Q_k}{\partial \beta_{0k}} \\ \frac{\partial Q_k}{\partial \beta_{1k}} \\ \frac{\partial Q_k}{\partial \beta_{2k}} \end{bmatrix}$$

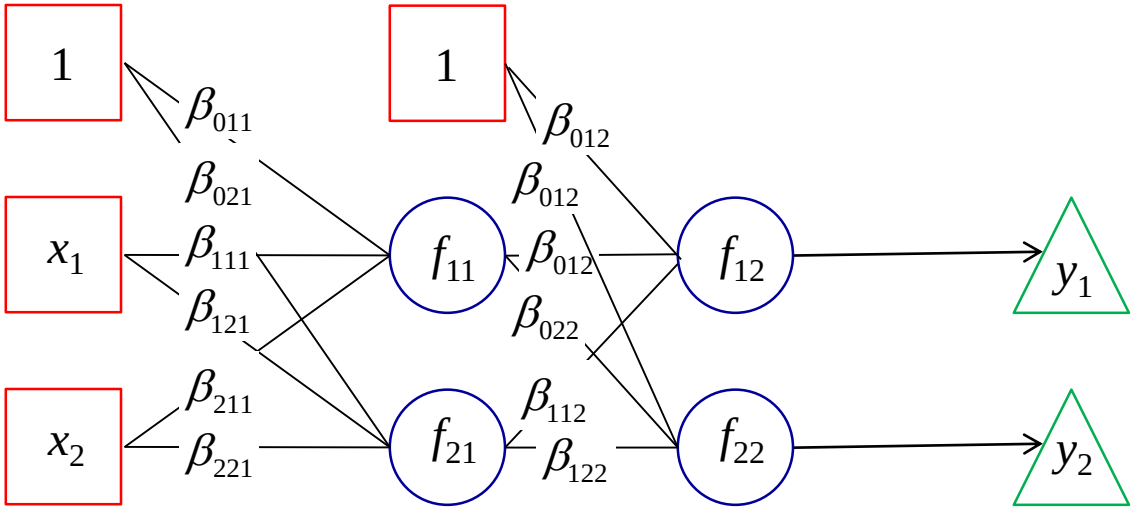
Gradient Descent

$$\hat{\beta}_k^{(t+1)} = \hat{\beta}_k^{(t+1)} - \gamma \nabla Q_k(\hat{\beta}_k^{(t+1)})$$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



Let's go through the process as multiple one layer Neural Nets, from right to left, Backpropagation.

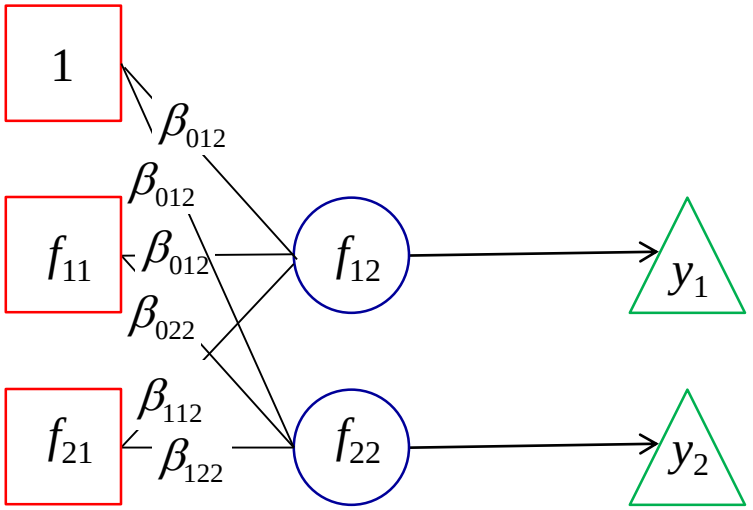
coefficient node layer

$j = 0,1,2 \quad k = 1,2 \quad l = 1,2$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



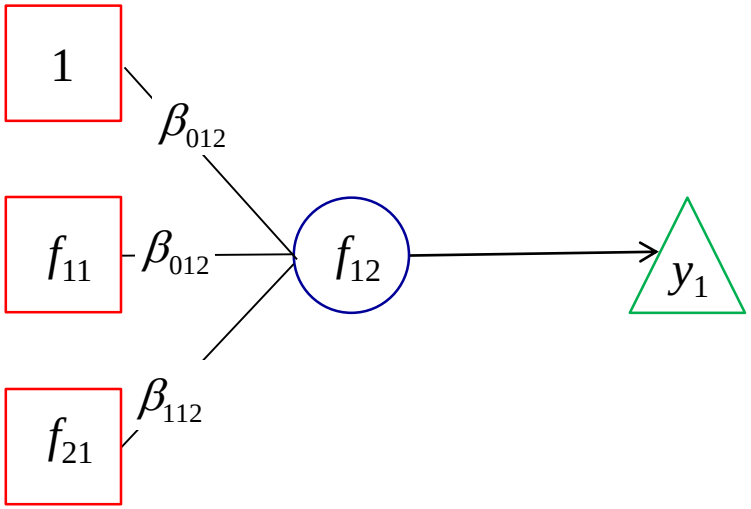
We consider the first output layer as input to the second layer. Estimate coefficients.

coefficient	node	layer
$j = 0,1,2$	$k = 1,2$	$l = 1,2$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



And first focus only on the first node.

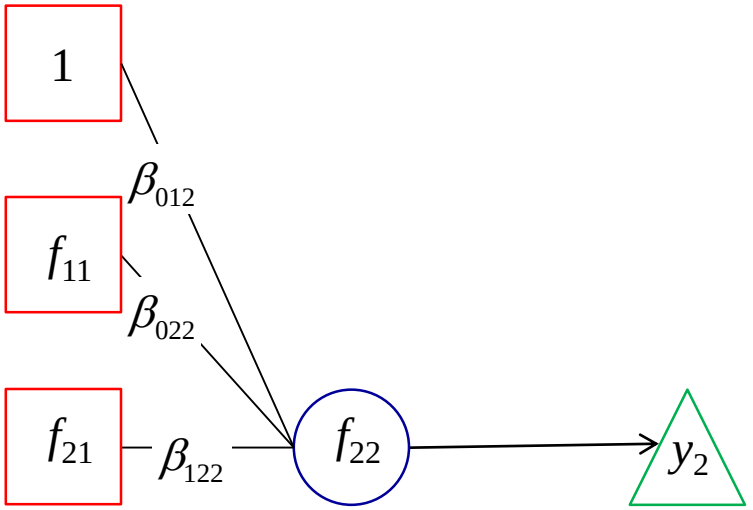
Estimate coefficients.

coefficient	node	layer
$j = 0,1,2$	$k = 1,2$	$l = 1,2$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



Then focus on the second node.

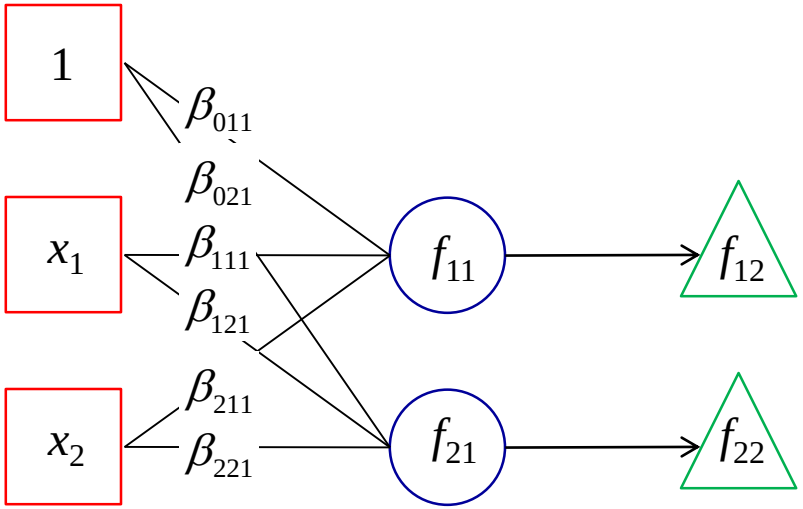
Estimate coefficients.

coefficient	node	layer
$j = 0,1,2$	$k = 1,2$	$l = 1,2$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



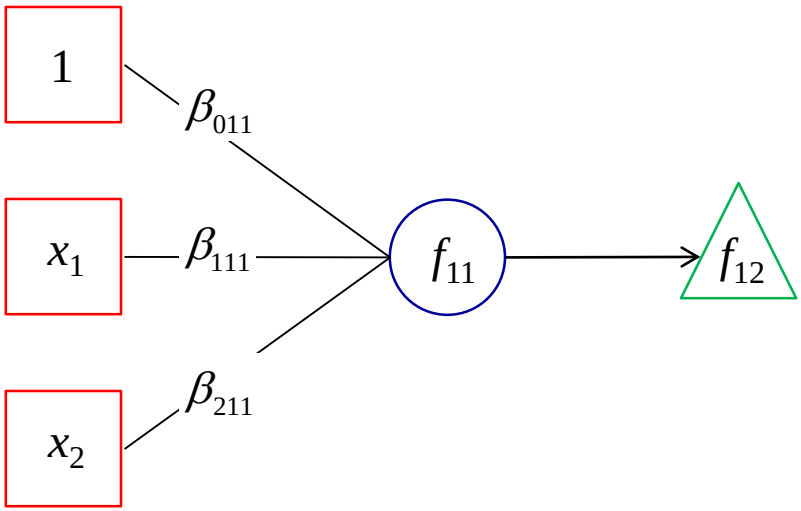
And now the second layer is considered the output of the first layer. Estimate coefficients.

coefficient	node	layer
$j = 0,1,2$	$k = 1,2$	$l = 1,2$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



And first focus only on the first node.

Estimate coefficients.

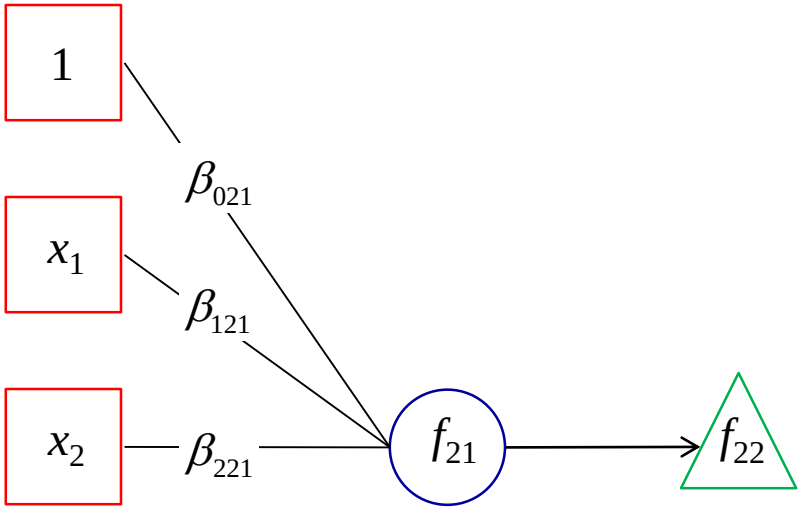
coefficient node layer

$j = 0,1,2$ $k = 1,2$ $l = 1,2$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



Then focus on the second node.

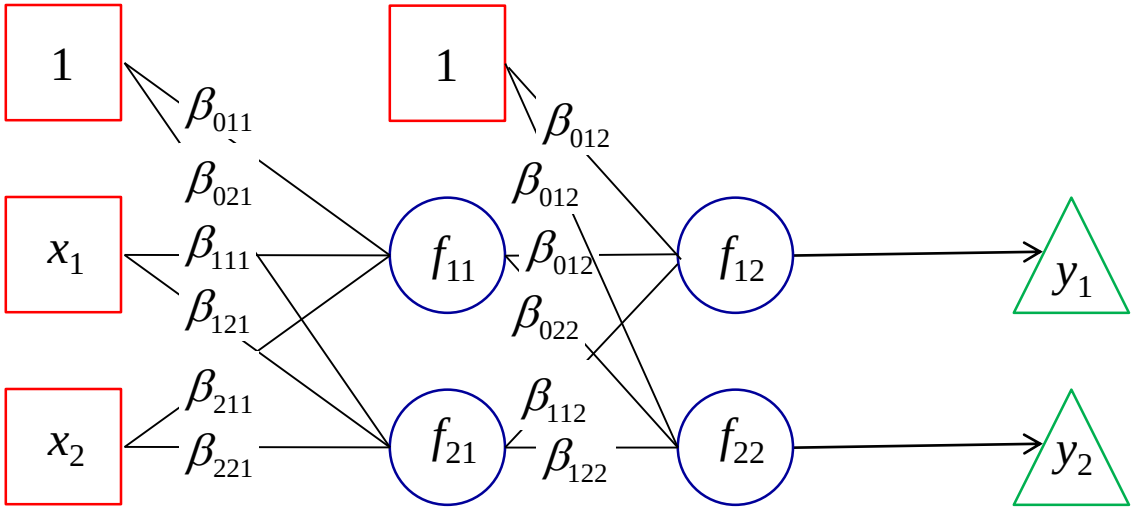
Estimate coefficients.

coefficient	node	layer
$j = 0,1,2$	$k = 1,2$	$l = 1,2$

5. MultiLayer (Deep Learning) NN-Multivariate

NN's can have more than one "hidden" layer.

The outputs from one layer become the inputs to the next.



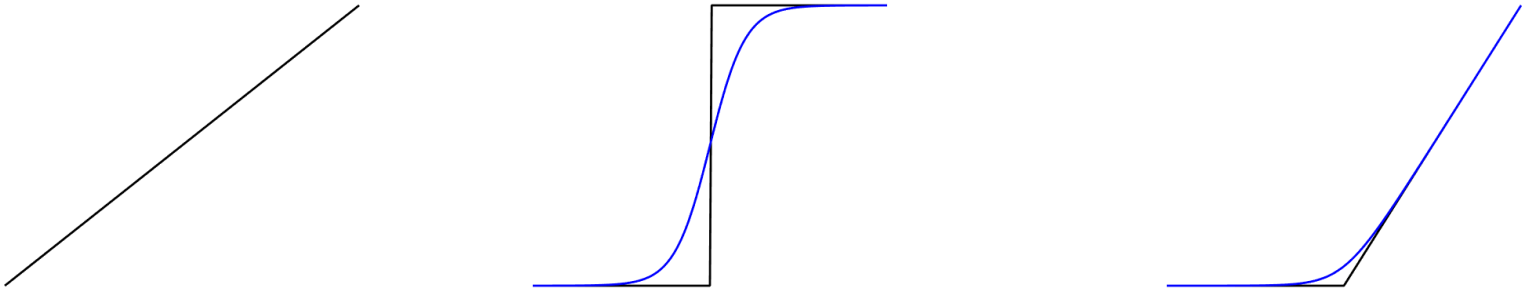
And hence solve the two or multi layer problem.

coefficient node layer

$j = 0,1,2 \quad k = 1,2 \quad l = 1,2$

6. Discussion

Linear, Logistic, NonLinear Regression as NN's

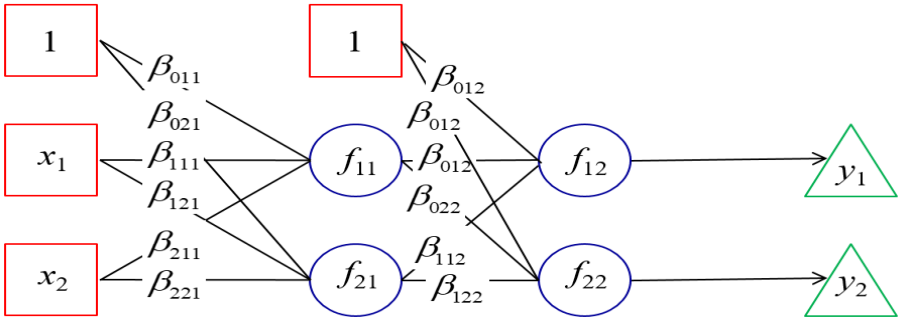


-Gradient Descent Backpropagation

$$\nabla Q = \left(\frac{\partial Q}{\partial \beta_j} \right) \qquad \hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} - \gamma \nabla Q(\hat{\beta}^{(t)})$$

Discussed foundational ideas of neural networks.

These ideas can be expanded in many directions.



Thank You!

Questions?

Daniel.Rowe@Marquette.Edu