Nonlinear models for regression

Michela Mulas

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- Simple least squares models
- Multiple least squares models
- Penalized regression models

  Ridge regression

  Lasso regression

  Elastic net regression

A quick recap

During the last lectures, we did...

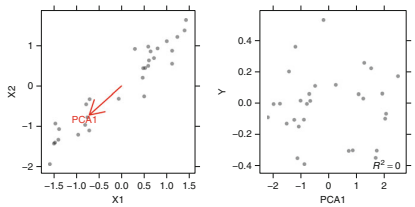
- **Linear models for regression**.
- Principal component regression, a two-step regression approach:
  1. Dimension reduction through PCA.
  2. Regression.

  The drawback of PCR us that dimension reduction via PCA does not necessarily produce new predictors that explain the response.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- Principal component regression

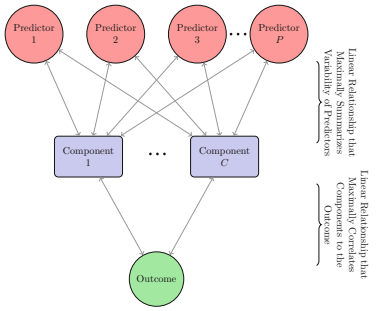


- The two predictors $X1$ and $X2$ are correlated.
- PCA summarizes this relationship using the direction of maximal variability.
- The first PCA direction contains no predictive information about the response.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- Partial least squares regression.
- PLS is also known as **projection to latent structure**.
- PLS is a dimensionality reduction technique for maximixing the covariance between the predictor matrix **X** and the outcome **y** (in a univariate response case).



A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- Partial least squares regression originated with the nonlinear iterative partial least squares (NIPALS) algorithm.

  It iteratively seeks to **find underlying, or latent, relationships among the predictors which are highly correlated with the response**.

  Each iteration of the algorithm assesses the relationship between **X** and **y**.

  PLS numerically summarizes this relationship with a vector of weights.

  Also know as a **direction**.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- Partial least squares regression originated with the nonlinear iterative partial least squares (NIPALS) algorithm.

  It iteratively seeks to **find underlying, or latent, relationships among the predictors which are highly correlated with the response**.

  X is then orthogonally projected onto the direction to generate scores.

  The scores are then used to generate loadings.

  They measure the correlation of the score vector to the original predictors.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- Partial least squares regression originated with the nonlinear iterative partial least squares (NIPALS) algorithm.

  It iteratively seeks to **find underlying, or latent, relationships among the predictors which are highly correlated with the response**.

  At the end of each iteration, the predictors and the response are “deflated” by subtracting the current estimate of the predictor and response structure, respectively.

  The new deflated predictor and response information are then used to generate the next set of weights, scores, and loadings.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- Like PCA, PLS finds linear combinations of the predictors.
- These linear combinations are commonly called components or **latent variables**.

  PCA linear combinations are chosen to maximally summarize predictor space variability.

  PLS linear combinations of predictors are chosen to maximally summarize covariance with the response.

  PLS finds components that maximally summarize the variation of the predictors while simultaneously requiring these components to have maximum correlation with the response.

  PLS therefore strikes a compromise between the objectives of predictor space dimension reduction and a predictive relationship with the response.

A quick recap

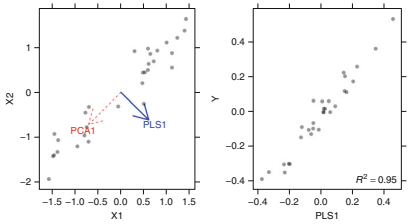
During the last lectures, we did...

- **Linear models for regression**.
- PLS strikes a compromise between the objectives of predictor space dimension reduction and a predictive relationship with the response.
- PLS can be viewed as a **supervised dimension reduction procedure**.
- PCR is an unsupervised procedure.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- PCR Vs PLSR

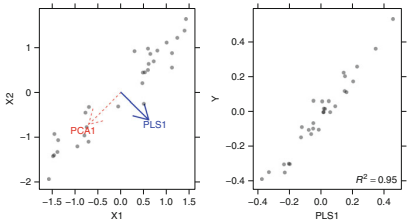


- An example of PLSR for a simple data set with two predictors and one response.
- **Left**: The first PLS direction is nearly orthogonal to the first PCA direction.
- **Right**: Unlike PCA, the PLS direction contains highly predictive information for the response.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- PCR Vs PLSR

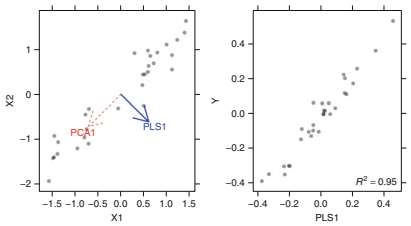


- The two directions are nearly orthogonal, indicating that the optimal dimension reduction direction was not related to maximal variation in the predictor space.
- PLS identified the optimal predictor space dimension reduction for the purpose of regression with the response.

A quick recap

During the last lectures, we did...

- **Linear models for regression**.
- PCR Vs PLSR



- Prior to performing PLS, the predictors should be centered and scaled, especially if the predictors are on scales of differing magnitude.
- PLS will seek directions of maximum variation while simultaneously considering correlation with the response.

A quick recap

During the last lecture, we did...

- Guest lecture on Bayesian inference.

Brian Hayes, *An Adventure in the Nth Dimension*, American Scientist, Vol. 99, (2011)

Harold Jeffreys, *Scientific Inference 3rd edition*, 1931

Today's goal

Today, we going to ...

- Nonlinear models for regression.
- **Neural networks**

Reference

Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*, Springer (2014)

Simon Haykin. *Neural Networks: A Comprehensive Foundation*, Pearson (2008, 3rd edition).

Definitions

Neural networks are powerful nonlinear regression techniques inspired by theories about how the brain works.

Haykin definition of neural network sounds like:

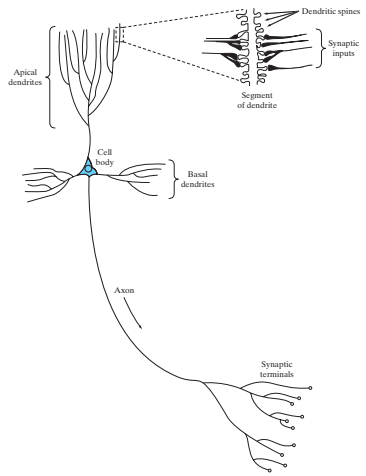
A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two aspects:

1. Knowledge is acquired by the network from its environment as a learning process.
2. Interneuron connection strengths, knows as synaptic weights, are used to store the acquired knowledge.

Naive neurobiology
Biological neuron

Functional units of a biological neuron:

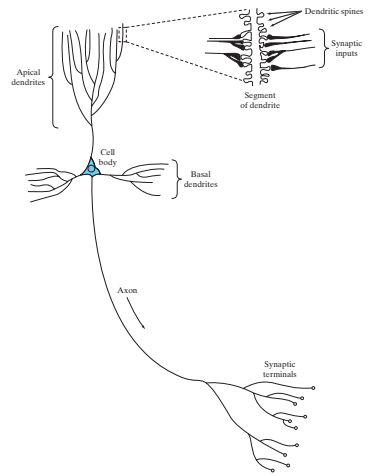
- **Cell body** has a nucleus that contains information about heredity traits, and a plasma that holds the molecular equipment used for producing the material needed by the neuron.
- **Dendrites** receive signals from other neurons and pass them over to the cell body.
- **Axon**, which branches into collaterals, receives signals from the cell body and carries them away through the synapse to the dendrites of neighbouring neurons.



Naive neurobiology
Biological neuron

Functional units of a biological neuron:

- A **stimulus** travels within the dendrites and through the cell body towards the pre-synaptic membrane of the synapse.
- A **neurotransmitter** is released from the vesicles in quantities proportional to the strength of the incoming signal.
- The neurotransmitter diffuses within the **synaptic gap** towards the post-synaptic membrane, and eventually into the dendrites of neighbouring neurons.



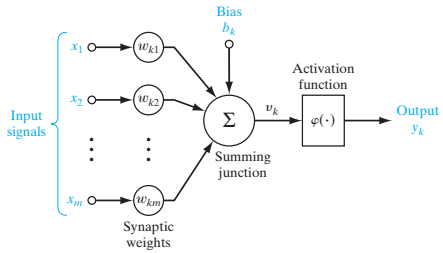
The single perceptron
Mimicking nature

An artificial neuron is a naive model of a biological neuron, which does not take into account much of the biological one but it is a model of it.

Each artificial neuron receives signals from the environment, or other neurons, gathers these signals and when fired, transmits these signal to all connected neurons.

- Input signals are inhibited or exited trough negative or positive numerical weights associated with each connection (**synaptic weights**).

  A signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}.



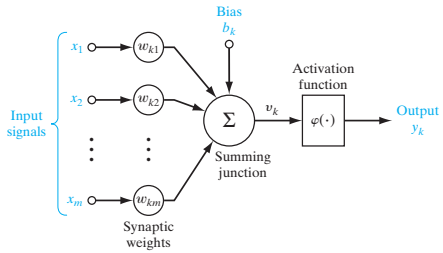
The single perceptron
Mimicking nature

An artificial neuron is a naive model of a biological neuron, which does not take into account much of the biological one but it is a model of it.

Each artificial neuron receives signals from the environment, or other neurons, gathers these signals and when fired, transmits these signal to all connected neurons.

- The neuron collects all incoming signals, and compute a net input signal as a function of the respective weights (**summing junction**).

  The summing junction also includes an externally applied **bias**, b_k, for increasing or lowering the net input of the activation function.



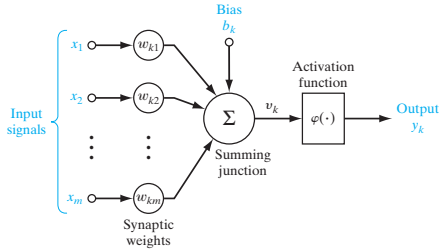
The single perceptron
Mimicking nature

An artificial neuron is a naive model of a biological neuron, which does not take into account much of the biological one but it is a model of it.

Each artificial neuron receives signals from the environment, or other neurons, gathers these signals and when fired, transmits these signal to all connected neurons.

- The firing of a neuron and the strength of the existing signal are controlled by a function (**activation function**).

  It squashes (limits) the permissible amplitude range of the output signal to some finite value.



The single perceptron
Mimicking nature

An artificial neuron is a naive model of a biological neuron, which does not take into account much of the biological one but it is a model of it.

Each artificial neuron receives signals from the environment, or other neurons, gathers these signals and when fired, transmits these signal to all connected neurons.

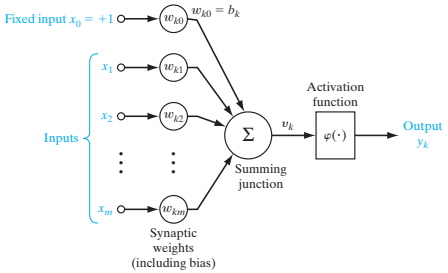
We might describe the neuron k as:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j$$

$$y_k = \varphi(v_k)$$

We account for the presence of the bias by:

- Adding a new input signal fixed at $+1$.
- Adding a new synaptic weight equal to the bias b_k.



The single perceptron
Activation functions

The activation function, $\varphi(v)$, defines the output of a neuron in terms of the induced local field v.

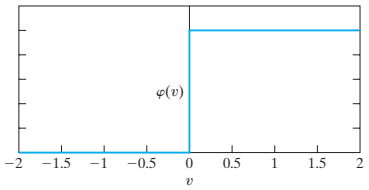
Threshold Function

$$\varphi(v) = \begin{cases} 1 \text{ if } v \geq 0 \\ 0 \text{ if } v < 0 \end{cases}$$

In engineering, this form of a threshold function is commonly referred to as a Heaviside function.

Correspondingly, the output of neuron k employing such a threshold function is expressed as:

$$y_k = \begin{cases} 1 \text{ if } v_k \geq 0 \\ 0 \text{ if } v_k < 0 \end{cases}$$



The single perceptron
Activation functions

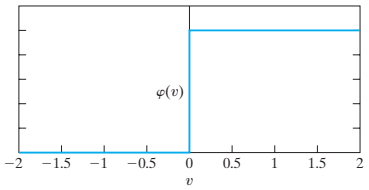
The activation function, $\varphi(v)$, defines the output of a neuron in terms of the induced local field v.

Threshold Function

$$y_k = \begin{cases} 1 \text{ if } v_k \geq 0 \\ 0 \text{ if } v_k < 0 \end{cases}$$

In this model, the output of a neuron takes on the value of 1 if the induced local field of that neuron is nonnegative, and 0 otherwise.

This statement describes the all-or-none property of the **McCulloch-Pitts model**.



The single perceptron

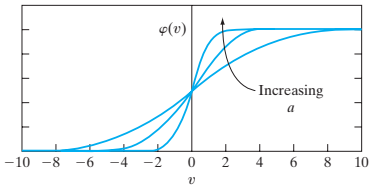
Activation functions

The activation function, $\varphi(v)$, defines the output of a neuron in terms of the induced local field v.

Sigmoid Function

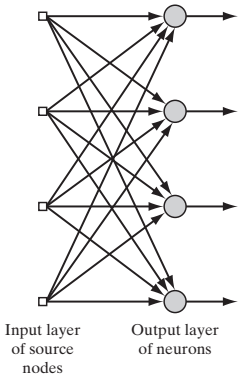
$$\varphi(v) = \frac{1}{1 + exp(-av)}$$

- Example of sigmoid: **logistic function**.
- a is the slope parameter.
- By varying the parameter a, we obtain sigmoids of different slopes.



Network architectures

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network.

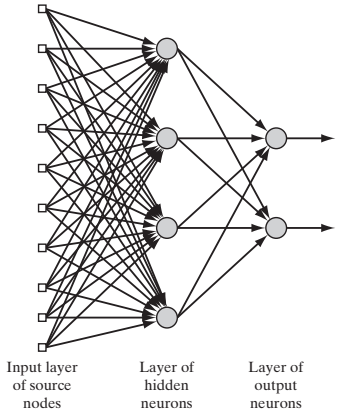


Single-Layer Feedforward Networks

- In a layered neural network, the neurons are organized in the form of layers.
- Simplest form: An input layer of source nodes that projects directly onto an output layer of neurons (computation nodes), but not vice versa.

Network architectures

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network.

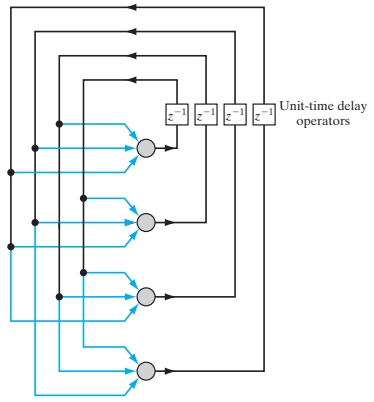


Multilayer Feedforward Networks

- We add layers: Hidden layers.
- They intervene between the external input and the network output in some useful manner.
- By adding one or more hidden layers, the network is enabled to extract higher-order statistics from its input.

Network architectures

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network.

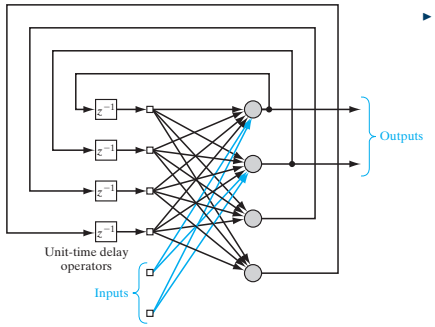


Recurrent Networks

- They have at least one feedback loop.
- The presence of feedback loops has a profound impact on the learning capability of the network and on its performance.
- The feedback loops involve the use of particular branches com- posed of unit-time delay elements (z^{-1}), which result in a nonlinear dynamic behavior, assuming that the neural network contains nonlinear units.

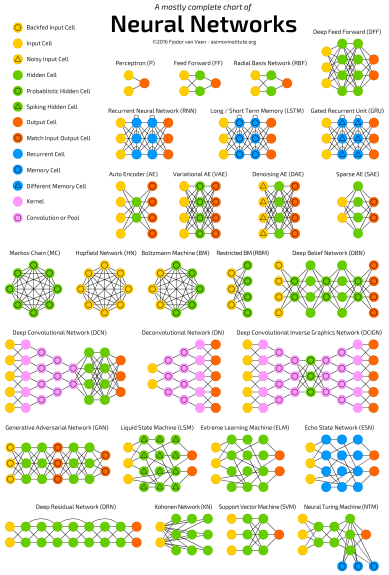
Network architectures

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network.



Recurrent Networks

- The feedback connections may also originate from the hidden neurons as well as from the output neurons.



A big family ...