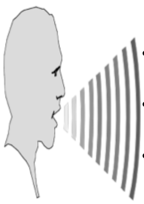
EEL5840: Elements of Machine Intelligence

Rous

Announcements



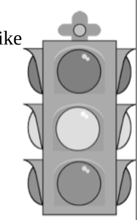
- Announcements:
  - > None
- Today's Handouts in www:
  - > Outline Class 9
- Web Site
  - > www.mil.ufl.edu/eel5840
  - > Software and Notes
  - > XLISP Documentation

EEL5840: Elements of Machine Intelligence

Rous

Today's Menu

- Introduction to ANNs (original material by Dr. Mike Nechyba)
  - > Slides 1-25
  - > Inputs & Outputs
  - > Biological Inspiration
  - > Basic Building Block
  - > Perceptrons
  - > A Simple Example



EEL5840: Elements of Machine Intelligence

Rous

LISP Lab 2

- <u>Functional Arguments</u>

  We want to write a function mapcar such that we can pass a function, say *f*, to the function with a list of instances of the single argument to the function, evaluate each instance and assemble the results into a list. For example:

```
Lisp>  (mapcar ‘oddp ‘(1 2 3))
LISP> (T Nil T)

   Thus, (mapcar ‘oddp ‘(1 2 3))  == (list (oddp 1) (oddp 2) (oddp 3))

   (defun mymap (f lis) (cond
                ((null lis)            nil)
                (‘else          (cons (f   (car lis))  (mymap f (cdr lis))))
   ))
```


EEL5840: Elements of Machine Intelligence

Rous

LISP Lab 2

- <u>Functional Arguments</u>

  Unfortunately this version looks good but it will not run! The LISP reader interprets *f* as the actual name of a function rather than a variable that represents the actual name of the function. Functional arguments (or parameters) are called *funargs* and demand special processing. The system supplies two functions to deal with funargs: FUNCALL and APPLY.

  template: (funcall #'<fname> <arg1> <arg2> ... <argN>)

  (car '(a b c)) == (funcall #'car '(a b c))

  (append '(a b) '(c d)) == (funcall #'append '(a b) '(c d))

  Apply works similarly, by using its first argument which must be a *funarg* on the elements of its second argument value, a list.

  template: (apply #'<fname> <list of arguments>)

  (car '(a b c)) == (apply #'car '((a b c)))

  (append '(a b) '(c d)) == (apply #'append '((a b) (c d)))

EEL5840: Elements of Machine Intelligence

Rous

LISP Lab 2

- Functional Arguments

  Now we can fix our mapcar:

```
(defun mymap (f lis) (cond
                ((null lis)              nil)
                ('else   (cons (funcall f   (car lis))  (mymap f (cdr lis))))
))
```

LISP

```
> (mymap 'oddp '(1 2 3))
 (T NIL T)
 > (defun plus1 (n) (+ 1 n))
PLUS1
> (mymap 'plus1 '(1 2 3))
(2 3 4)
```


EEL5840: Elements of Machine Intelligence

Rous

Introduction to Neural Networks

Introduction to Neural Networks

What are neural networks?

- Nonlinear function approximators

How do they relate to pattern recognition/classification?

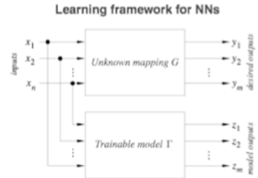
- Nonlinear discriminant functions
- More complex decision boundaries than linear discriminant functions (e.g. Fisher, Gaussians with equal covariances)

EEL5840: Elements of Machine Intelligence

Rous

Introduction to Neural Networks

Learning framework for NNs



EEL5840: Elements of Machine Intelligence

Rous

Introduction to Neural Networks

Inputs/outputs

Definitions:

$\mathbf{y} = G(\mathbf{x})$ *(e.g discriminant function we want to learn)*

$\mathbf{x} = [x_1\ x_2\ \dots\ x_n]^T$ *(n inputs)*

$\mathbf{y} = [y_1\ y_2\ \dots\ y_m]^T$ *(m process outputs)*

Trainable model:

$\mathbf{z} = \Gamma(\mathbf{x}, \mathbf{w})$ *($\mathbf{w}$ = adjustable parameters)*

$\mathbf{z} = [z_1\ z_2\ \dots\ z_m]^T$ *(m model outputs)*

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Learning goal:

Find $\mathbf{w}^*$ such that:

$$E(\mathbf{w}^*) \le E(\mathbf{w}), \forall \mathbf{w},$$

where $E(\mathbf{w})$ = error between G and Γ.

What should $E(\mathbf{w})$ be?

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Error function (ideal)

Ideally,

$$E(\mathbf{w}) = \int_{\mathbf{x}} \|\mathbf{y} - \mathbf{z}\|^2 p(\mathbf{x}) d\mathbf{x}$$

How to compute?

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Error function (practical)

Input/output data: p input-output training patterns

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_p \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_p \end{bmatrix}$$

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{in} \end{bmatrix}^T$$

$$\mathbf{y}_i = \begin{bmatrix} y_{i1} & y_{i2} & \cdots & y_{im} \end{bmatrix}^T, \mathbf{z}_i = \Gamma(\mathbf{x}_i, \mathbf{w}).$$

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{p} \|\mathbf{y}_i - \mathbf{z}_i\|^2 = \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{m}(y_{ij} - z_{ij})^2$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Artificial neural networks (NNs)

Neural networks are one type of parametric model Γ.

- Nonlinear function approximators
- Adjustable (trainable) parameters $\mathbf{w}$ (weights)
- Map inputs to outputs

Why "Neural Network?"

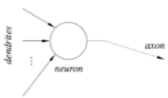
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Biological inspiration

- Structure and function loosely based on *biological* neural networks (e.g. brain).
- Relatively simple building blocks connected together in massive and parallel network.



What does a neuron do?

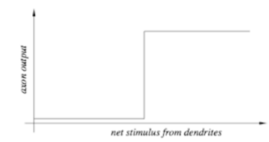
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Neuron transfer function

Rough approximation: threshold function



Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Neural networks: crude emulation of biology

- Simple basic building blocks.
- Individual units are connected massively and in parallel.
- Individual units have threshold-type activation functions.
- Learning through adjustment of the strength of connection (weights) between individual units

Caveat: **Artificial neural networks are much, much, much simpler than biological systems. Example: Human brain:**

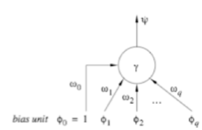
- 10^{10} neurons
- 10^{12} connections

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Basic building blocks of neural networks



EEL5840: Elements of Machine Intelligence

Rous

Introduction to Neural Networks

Basic building block: the unit

$\underset{\sim}{\phi} = \left[\phi_0\ \phi_1\ \dots\ \phi_q\right]^T$ (scalar inputs)

$\mathbf{w} = \left[\omega_0\ \omega_1\ \dots\ \omega_q\right]^T$ (weights)

γ = nonlinear activation function

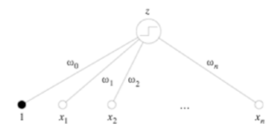
$\psi = \gamma(\mathbf{w} \cdot \underset{\sim}{\phi}) = \gamma\left(\sum_{i=0}^{q} \omega_i \phi_i\right)$ (output)

EEL5840: Elements of Machine Intelligence

Rous

Introduction to Neural Networks

Perceptrons: the simplest "neural network"



What is this?

Rosenblat & Minsky

EEL5840: Elements of Machine Intelligence

Rous

Introduction to Neural Networks

Threshold activation function

$$\gamma_t(u) = \begin{cases} 1 & u \geq \theta \\ 0 & u < \theta \end{cases}$$

EEL5840: Elements of Machine Intelligence

Rous

Introduction to Neural Networks

Perceptron output

Perceptron mapping:

$$z = \begin{cases} 1 & \mathbf{w}^t\mathbf{x} \geq 0 \\ 0 & \mathbf{w}^t\mathbf{x} < 0 \end{cases}$$

where,

$\mathbf{x} = \left[1\ x_1\ \dots\ x_n\right]^T$

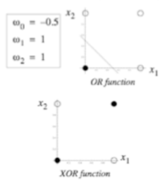
$\mathbf{w} = \left[\omega_0\ \omega_1\ \dots\ \omega_n\right]^T$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Limited mapping capability



Rous

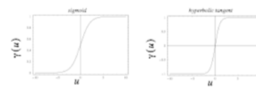
EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

More general networks: activation function

$$\gamma(u) = \frac{1}{1+e^{-u}} \text{ (sigmoid)}$$

$$\gamma(u) = \frac{e^{u}-e^{-u}}{e^{u}+e^{-u}} \text{ (hyperbolic tangent)}$$

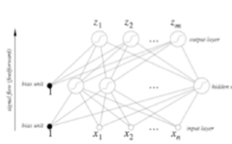


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

More general networks: multilayer perceptrons (MLPs)

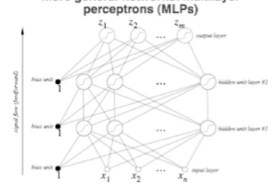


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

More general networks: multilayer perceptrons (MLPs)

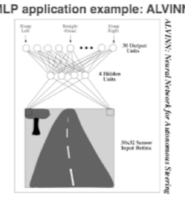


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

MLP application example: ALVINN

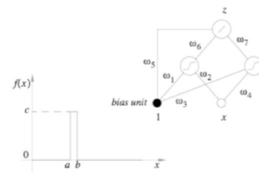


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

A simple example



Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Derivation of function $f(x)$

$$f(x) = c[\gamma_t(x-a) - \gamma_t(x-b)]$$

$$f(x) = c\gamma_t(x-a) - c\gamma_t(x-b)$$

$$\gamma_t(u) \to \gamma(ku) \text{ as } k \to \infty$$

$$f(x) \approx c\gamma[k(x-a)] - c\gamma[k(x-b)] \text{ for large } k.$$

$$z = \omega_5 + \omega_6\gamma(\omega_1 + \omega_2 x) + \omega_7\gamma(\omega_3 + \omega_4 x)$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Weight values for simple example

	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7
set #1	$-kb$	k	$-ka$	k	0	$-c$	c
set #2	$-ka$	k	$-kb$	k	0	c	$-c$

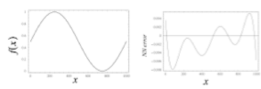
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Some theoretical properties of NNs

Single-input functions: what does the previous example say about single-input functions?



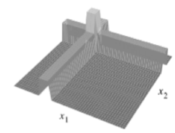
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Multi-input functions: universal function approximator?

Does the single-input example hold in general?



Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Neural networks in practice: 3 basic steps

1. Collect input/output training data.

2. Select an appropriate neural network architecture:

- Number of hidden layers
- Number of hidden units in each layer.

3. Train (adjust) the weights of the neural network to minimize the error measure,

$$E = \frac{1}{2}\sum_{i=1}^{p}\|\mathbf{y}_i - \mathbf{z}_i\|^2$$

Rous

EEL5840: Elements of Machine Intelligence

The End!


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks