Function Approximation

Mark A. Austin

University of Maryland

austin@umd.edu
ENCE 201, Fall Semester 2023

September 25, 2023

Overview

Part 1

Function Approximation

Motivating Ideas

Function Approximation

A function approximation asks us to select a function, g(x), among a well-defined set of options that approximates – closely matches – a second function, f(x), in a task-specific way.

Approximation Examples: Many approaches ...

Strategy 1: Polynomial Approximation

Polynomial Approximation

Replace function f(x) by a simplier polynomial approximation g(x). Then, use g(x) in computations instead of f(x).

Example 1: Replace $y = f(x) = e^{-x}$ by a quadratic approximation:

$$f(x) = e^{-x} \quad \longrightarrow \quad g(x) = 1 - x + \frac{x^2}{2}. \tag{1}$$

Example 2: Replace $y = sin(x)$ on $x \in [0, \pi]$ by a quadratic approximation:

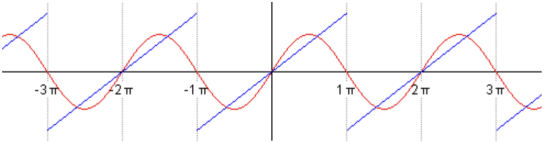
$$f(x) = sin(x) \quad \longrightarrow \quad g(x) = \frac{4x}{\pi^2}\left[\pi - x\right]. \tag{2}$$

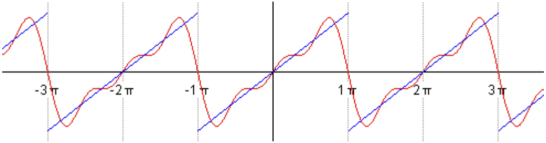
Strategy 2: Fourier Series

Fourier Series

A Fourier series is an expansion of a periodic function f(x) in terms of an infinite sum of trigonometric (i.e., sines and cosines) and/or exponential functions.

Example 1: Progressive refinement of sawtooth function:



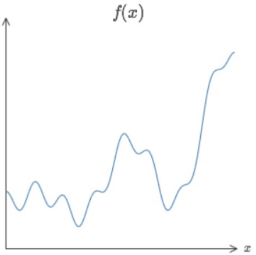


Strategy 3: Machine Learning

Neural Network

Use observable (or experimentally measured) data to train a neural network to capture (or estimate) input-to-output functionality.

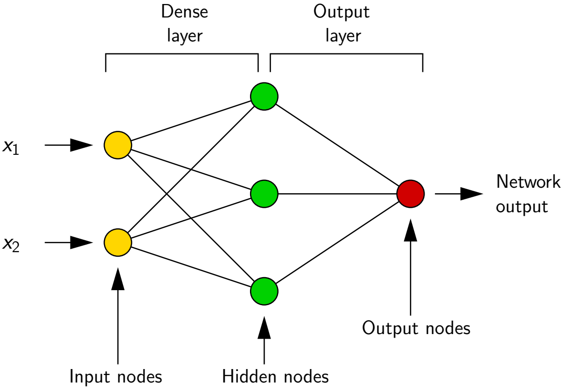
- Neural networks are universal function approximators, no matter how complex:
- Neural network architectures are highly scalable and flexible.



Caveat: Very large neural networks may be close to impossible to train and generalize correctly → AI chips.

Strategy 3: Machine Learning

Example 1: Neural Network with One Hidden Layer:



Data inputs x_1 and x_2 are transformed into a prediction $f(x_1, x_2)$.

Strategy 3: Machine Learning

Example 2: Learn how to classify univariate time series as belonging to one of six categories:

Data

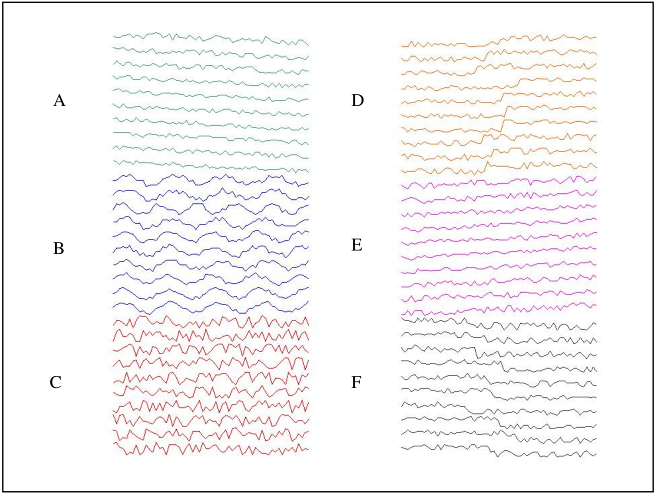
- UCI Synthetic Control Chart Time Series Data Set contains 600 sequences of data.
- Partition data: 450 items for training; 150 items for testing.

Six Categories of Datastream

- **A and E:** Decreasing and Increasing Trend.
- **B:** Cyclic.
- **C:** Normal.
- **D and F:** Upward and Downward Shift.

Strategy 3: Machine Learning

Representative Data Streams:



Strategy 3: Machine Learning

Training and Testing Corpus

- uci
  - train
    - features: 0.csv, 1.csv, 2.csv, 3.csv, ... ,449.
    - labels: 0.csv, 1.csv, 2.csv, 3.csv, ... ,449.
  - test
    - features: 0.csv, 1.csv, 2.csv, 3.csv, ... ,149.
    - labels: 0.csv, 1.csv, 2.csv, 3.csv, ... ,149.

CSV Data File Format and Label Mappings

Features: 0.csv

20.59 ← line 01
18.51
⋮ ⋮
16.32
10.72 ← line 60

Labels: 0.csv

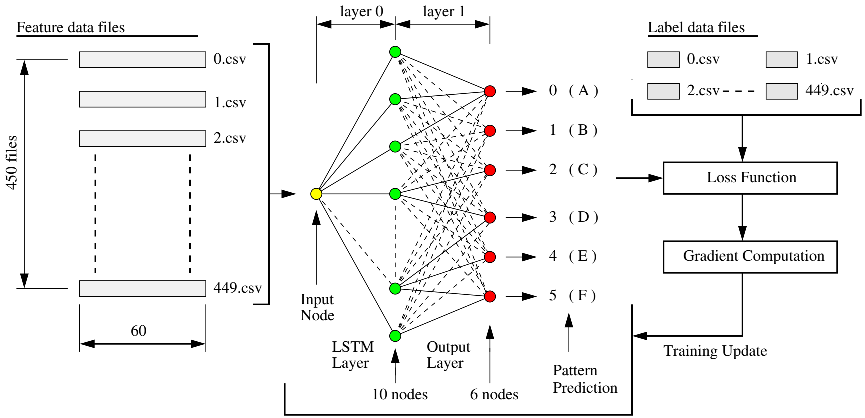
3 ← (maps to output 3)

Output Mapping

0 ← A
1 ← B
2 ← C
3 ← D
4 ← E
5 ← F

Strategy 3: Machine Learning

RNN Architecture + Sequences of Feature and Label vectors



Strategy 3: Machine Learning

Training the Model for 40 Epochs:

```
int nEpochs = 40;
net.fit(trainData, nEpochs);
```

Evaluation Metrics and Confusion Matrix:

```
Accuracy:  0.8867   Recall:    0.8890   <--- It works!
Precision: 0.8886   F1 Score: 0.8883

   0   1   2   3   4   5
-------------------
  26   0   0   0   0   0 |   0 = 0
   0  29   0   0   0   0 |   1 = 1
   0   0  15   0   7   0 |   2 = 2
   0   0   0  20   0   1 |   3 = 3
   0   0   9   0  21   0 |   4 = 4
   0   0   0   0   0  22 |   5 = 5
```

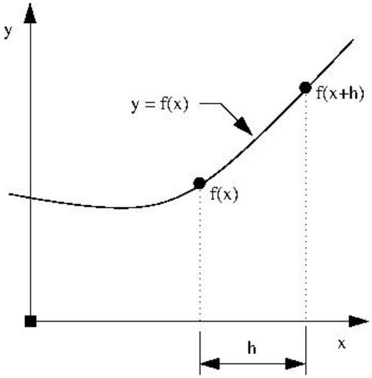
Taylor Series

Taylor Series

(Brook Taylor, 1715)

Motivating Idea

Let $y = f(x)$ be a smooth differentiable function.



Given $f(x)$ and derivatives $f^{'}(a)$, $f^{''}(a)$, $f^{'''}(a)$, etc, the purpose of Taylor's series is to estimate $f(x+h)$ at some distance h from x.

Taylor Series Expansion

Mathematical Expansion.

$$f(x+h) = \sum_{i=0}^{\infty} \frac{f^{i}(x)}{(i)!} h^{i} = f(x) + f^{'}(x)h + \frac{f^{''}(x)}{2!}h^2 + \frac{f^{'''}(x)}{3!}h^3 + \cdots \tag{3}$$

For a Taylor series approximation containing $(n+1)$ terms

$$f(x+h) = \sum_{i=0}^{i=n} \frac{f^{i}(x)}{(i)!} h^{n} + R_n(x) \tag{4}$$

The remainder, $R_n(x)$, after truncation is:

$$R_n(x) = \frac{f^{(n+1)}(c(x+h))}{(n+1)!} h^{(n+1)} \quad \text{with} \quad [x \leq c \leq x+h]\,. \tag{5}$$

Taylor Series Expansion

We can also write:

$$f(x+h) = \sum_{i=0}^{i=n} \frac{f^i(x)}{(i)!} h^n + O(h^{(n+1)}) \tag{6}$$

The big-O notation $O(h^n)$ indicates how quickly the error will change as a function of h.

- O(0) $\rightarrow$ Magnitude of error is constant, regardless of h.
- O(h) $\rightarrow$ Magnitude of error proportional to h.
- O(h^2) $\rightarrow$ Magnitude of error proportional to h squared.

Maclaurin Series Expansion

Maclaurin Series: A Maclaurin Series is nothing more than a Taylor series expansion about a = 0, i.e.,

$$f(h) = f(0) + \frac{f^{'}(0)}{1!}h + \frac{f^{''}(0)}{2!}h^2 + \frac{f^{'''}(0)}{3!}h^3 + \cdots \tag{7}$$

Trigonometric Series

$$sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

$$cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!}x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

Surprisingly, these series converge for all values of x.

Remainder Function

Remainder Function. The formula for $R_n(x)$,

$$R_n(x) = \frac{f^{(n+1)}(c(x+h))}{(n+1)!} h^{(n+1)} \quad \text{with} \quad [x \leq c \leq x+h] \,. \tag{8}$$

This formula is derived via the "mean value theorem" (see extra slides for details) and simply states there exists a point $c(x+h)$ such that:

$$\frac{f(x+h) - f(x)}{(x+h-x)} = f^1(c) \rightarrow f(x+h) = f(x) + f^1(c)\,(h) \,. \tag{9}$$

We can estimate $R_n(x)$ without knowing $c(x+h)$ explicitly.

Ratio Test and Interval of Convergence

For the power series centered about x = a,

$$P(a+h) = C_0(a)+C_1(a)h+C_2(a)h^2+C_3(a)h^3+C_4(a)h^4+\cdots \quad (10)$$

suppose that:

$$\lim_{n\to\infty}\left[\frac{|C_n(a)|}{|C_{n+1}(a)|}\right] = R. \quad (11)$$

then:

- If R $= \infty$, then the series converges for all values.
- If $0 < R < \infty$, then the series converges for all $h < R$.
- If R = 0, then the series converges only for h = 0.

We call R the **radius of convergence.**

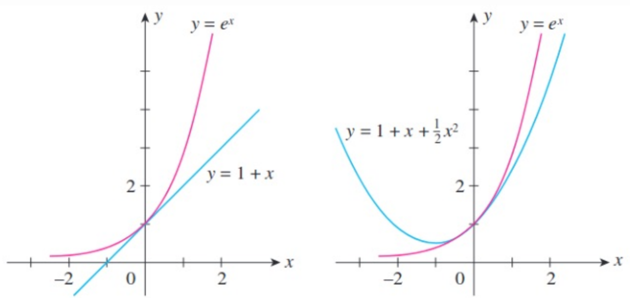
Solved Problems

Example 1: Taylor Series Expansion for e^x

Problem 1. Approximating e^x about $x = 0$.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots \tag{12}$$

Linear and quadratic approximations:



Example 1: Taylor Series Expansion for e^x

Now lets predict: $e^2 = 2.71828 * 2.71828 = 7.38904$.

No Terms	Numerical Estimate
1	1.0000
2	$1 + 2 \rightarrow 3.0000$
3	$1 + 2 + 4/2! \rightarrow 5.00000$
4	$1 + 2 + 4/2! + 8/3! \rightarrow 6.33333$
5	$1 + 2 + 4/2! + 8/3! + 16/4! \rightarrow 6.99999$

Estimate of Maximum Error: After five terms:

$$R_5(2) \leq \frac{e^{c(2)}}{6!} 2^6 = \frac{e^2}{6!} 2^6 = \frac{7.38904 * 64}{720} = 0.657. \qquad (13)$$

The actual error is: 0.389.

Example 1: Taylor Series Expansion for e^x

Test for Convergence. We have:

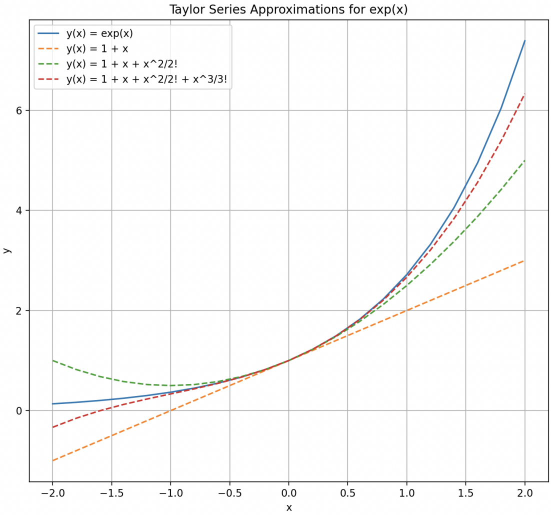
$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{14}$$

The limit

$$\lim_{n\to\infty} \left[\frac{|C_n(a)|}{|C_{n+1}(a)} \right] = \lim_{n\to\infty} \left[\frac{\frac{1}{n!}}{\frac{1}{(n+1)!}} \right] = \lim_{n\to\infty} (n+1) = \infty. \tag{15}$$

gives $\mathrm{R} = \infty$ and the series converges for all values of x.

Example 1: Taylor Series Expansion for $exp(x)$



Example 2: Taylor Series Expansion for $cos(x)$

Example 2. Taylor Series expansion for $cos(x)$ about x = 0.

We have: f(0) = cos(0), $f^1(0) = -sin(0)$, and so forth. Hence,

$$cos(x) = cos(0) - \frac{sin(0)}{1!}x + \frac{-cos(0)}{2!}x^2 + \frac{sin(0)}{3!}x^3 + \frac{cos(0)}{4!}x^4 + \cdots$$
$$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots$$

Can also use the same procedure to show:

$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots \tag{16}$$

Example 2: Taylor Series Expansion for $cos(x)$

Test for Convergence. We have:

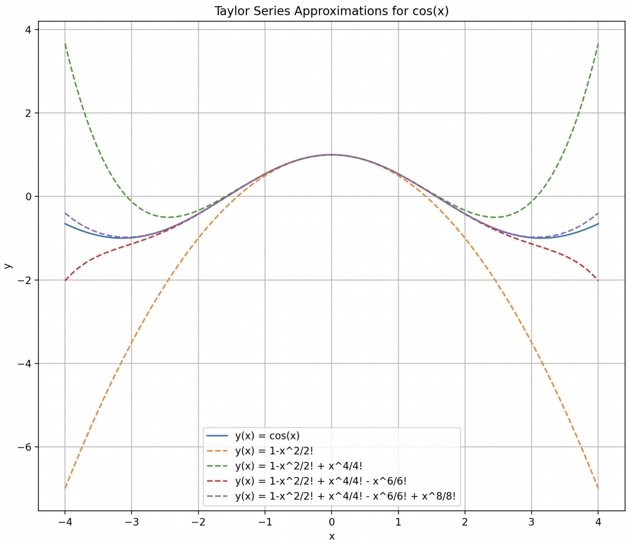
$$cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}. \tag{17}$$

Hence, $C_n = (-1)^n/(2n)!$ and:

$$\lim_{n\to\infty} \left[\frac{|C_n(a)|}{|C_{n+1}(a)|} \right] = \lim_{n\to\infty} \left[\frac{\frac{(-1)^n}{2n!}}{\frac{(-1)^{(n+1)}}{(2n+2)!}} \right] = \lim_{n\to\infty} (2n+1)(2n+2) = \infty. \tag{18}$$

The series converges for all real x.

Example 2: Taylor Series Expansion for $cos(x)$



Summary: Taylor Series Expansion

Strengths:

- When they are used to approximate complex functions with polynomials, the latter are much easier to differentiate and integrate.
- Can be easily manipulated to provide finite difference approximations to function derivatives.
- Can be used to get theoretical error bounds.

Weaknesses:

- Taylor series approximations require underlying function to be continuously differentiable.
- Series convergence for a wide range of values may only occur with an unreasonably large number of terms, a task that may be computationally infeasible.

Extra Slides

Extra Slides

Mean Value Theorem

Let function f be continuous on $[a, b]$

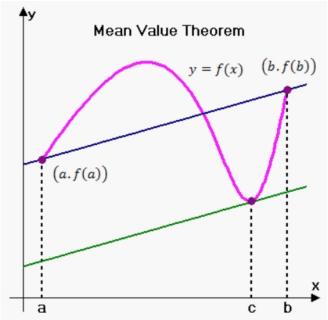
and differentiable on $[a, b]$.

There exists a point c in $[a, b]$ such that:

$$f^1(c) = \left[\frac{f(b) - f(a)}{b - a)}\right]. \qquad (33)$$

The right-hand side of the equation is the secant line connecting end points.

The green line is the tangent slope at point c.



Python Code Listings

Code 1: Taylor Series Approximations for cos(x)

```
# ==========================================================================
# TestTaylorSeries02.py: Compute and plot Taylor Series approximations for
# cos(x).
#
# Written By: Mark Austin                                      August 2023
# ==========================================================================

import math;
import numpy as np;
import matplotlib.pyplot as plt;

# Taylor series approximation for cos(x) ...

def func_cos(x,n):
    cos_approx = 0
    for i in range(n):
        coeff = (-1)**i
        num   = x**(2*i)
        denom = math.factorial(2*i)
        cos_approx = cos_approx + (coeff)*(num/denom)

    return cos_approx

# main method ...

def main():
    print("--- Enter TestTaylorSeries02.main()                ... ");
    print("--- ============================================== ... ");
```

Code 1: Taylor Series Approximations for cos(x)

```

# Part 1: Compute x values for numerical computation ...

x = np.linspace( -4.0, 4.0, num = 101)
y = np.cos(x)

print(x)
print(y)

# Part 2: Two-term polynomial approximation: y(x) = 1 - x^2/2! ...

yapprox2 = []
for xi in x:
   yapprox2.append( func_cos(xi,2) );

# Part 3: Three-term polynomial approximation: y(x) = 1 - x^2/2! + x^4/4! ...

yapprox3 = []
for xi in x:
   yapprox3.append( func_cos(xi,3) );

# Part 4: Four-term polynomial approximation: y(x) = 1 - x^2/2! + x^4/4! - x^6/6! ..

yapprox4 = []
for xi in x:
   yapprox4.append( func_cos(xi,4) );

# Part 5: Five-term polynomial approximation: y(x) = 1 - x^2/2! + x^4/4! - x^6/6! +

yapprox5 = []
```

Code 1: Taylor Series Approximations for cos(x)

```
    for xi in x:
       yapprox5.append( func_cos(xi,5) );

    # Part 6: Plot cos(x) and various polynomial approximations ...

    plt.plot(x,        y, label="y(x) = cos(x)",            linestyle="-")
    plt.plot(x, yapprox2, label="y(x) = 1-x^2/2!",          linestyle="--")
    plt.plot(x, yapprox3, label="y(x) = 1-x^2/2! + x^4/4!", linestyle="--")
    plt.plot(x, yapprox4, label="y(x) = 1-x^2/2! + x^4/4! - x^6/6!", linestyle="--")
    plt.plot(x, yapprox5, label="y(x) = 1-x^2/2! + x^4/4! - x^6/6! + x^8/8!", linestyle=
    plt.title("Taylor Series Approximations for cos(x)")
    plt.xlabel('x')
    plt.ylabel('y')
    plt.grid()
    plt.legend()
    plt.show()

    print("--- ============================================= ... ");
    print("--- Leave TestTaylorSeries02.main()                ... ");

# call the main method ...

main()
```