



| Welcome

IBF Presents
**AI & DEMAND FORECASTING
WORKSHOP**

Sara Brumbaugh
sara@ceresAnalytics.com
617-519-3151



*Institute of Business
Forecasting & Planning*

FOSTERING DEMAND PLANNING,
FORECASTING, S&OP &
ANALYTICS FOR 40+ YEARS

overall Agenda



- 1 AI & Machine Learning in Demand Forecasting: Setting the Context
- 2 Neural Networks for Demand Forecasting
- 3 Python Foundations for Machine Learning (Hands-On)
- 4 Building a Machine Learning Forecast (Hands-On)
- 5 AI and the planner...

AI & Machine Learning (ML) in Demand Forecasting

Setting the Context





*AI improves forecasting performance when it is embedded within a disciplined S&OP / IBP process, **not** when treated as a standalone solution*

AI in a Forecasting Environment can...



- ✓ enhance forecasts through improvements in pattern detection, scalability, and automation
- ✓ incorporate more data sources (internal and external signals)
- ✓ enable faster scenario analysis and exception identification
- ✓ increase planners' focus on decision-making by substituting for manual data work



AI in a Forecasting Environment should not...

- ✗ be a “black box replacement” for forecasting processes
- ✗ eliminate the need for domain expertise, collaboration, and governance
- ✗ support poor data, poor processes, or lack of organizational alignment
- ✗ be positioned as “fully autonomous planning” in most real-world environments today

Is Machine Learning AI?

<https://cloud.google.com/learn/artificial-intelligence-vs-machine-learning#what-is-machine-learning>

Machine Learning as a
component of AI
1 of 3

Google Cloud

Overview Solutions Products Pricing Resources Contact Us

Q Docs Support Sign in

Artificial intelligence (AI)
vs. machine learning (ML)

What is artificial
intelligence?

What is machine
learning?

How are AI and ML
connected?

Differences between AI
and ML

Benefits of using AI and
ML together

Applications of AI and ML

Related products and
services

Take the next step

→ What is machine learning?

[Machine learning](#) is a subset of artificial intelligence that automatically enables a machine or system to learn and improve from experience. Instead of explicit programming, machine learning uses algorithms to analyze large amounts of data, learn from the insights, and then make informed decisions.

Machine learning algorithms improve performance over time as they are trained—exposed to more data. Machine learning models are the output, or what the program learns from running an algorithm on training data. The more data used, the better the model will get.

→ How are AI and ML connected?

While AI and ML are not quite the same thing, they are closely connected. The simplest way to understand how AI and ML relate to each other is:

- AI is the **broader concept** of enabling a machine or system to sense, reason, act, or adapt like a human
- ML is an **application of AI** that allows machines to [extract knowledge from data and learn from it](#) autonomously

One helpful way to remember the difference between machine learning and artificial intelligence is to [imagine them as umbrella categories](#). Artificial intelligence is the overarching term that covers a wide variety of specific approaches and algorithms. Machine learning sits under that umbrella, but so do other major subfields, such as deep learning, robotics, expert systems, and [natural language processing](#).

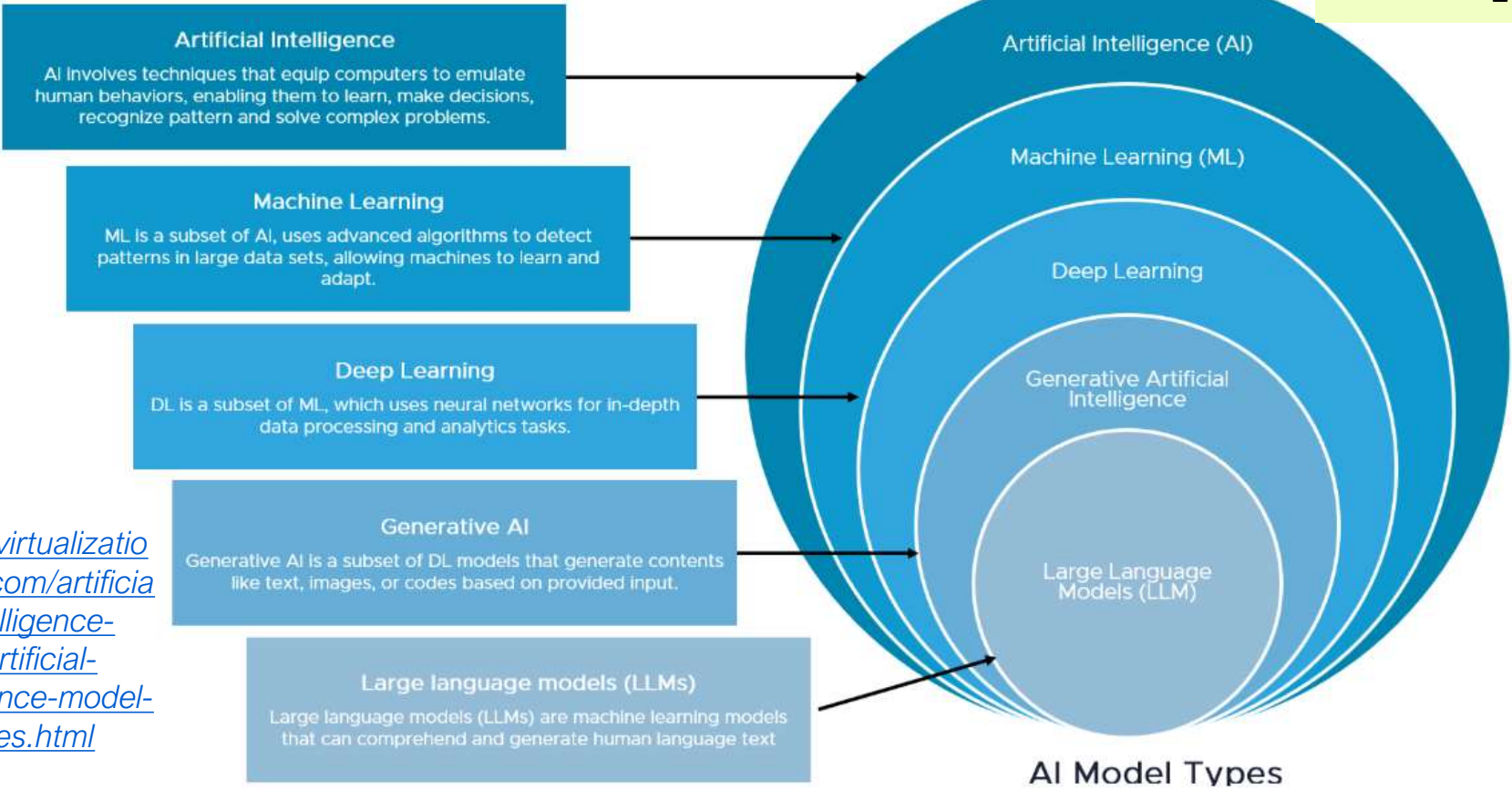
→ Differences between AI and ML

Now that you understand how they are connected, what is the main difference between AI and ML?

While artificial intelligence encompasses the idea of a machine that can mimic human intelligence, machine learning does not. [Machine learning aims to teach a machine how to perform a specific task and provide accurate results by identifying patterns.](#)



AI Model Types

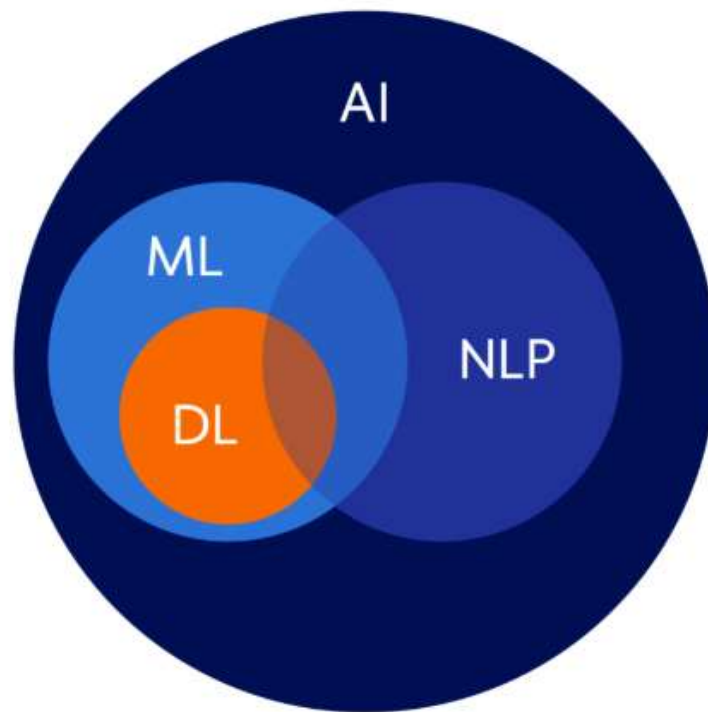


AI Evolution – Computational Complexity and energy needs

<https://virtualizationteam.com/artificial-intelligence-ai/artificial-intelligence-model-types.html>



Machine Learning as a component of AI
3 of 3



- Artificial intelligence
- Language Processing
- Machine learning
- Deep learning

4 key types of analytics

per <https://online.hbs.edu/blog/post/types-of-data-analysis>

- **Descriptive**
What happened
- **Diagnostic**
Why did it happen
- **Predictive**
What might happen in the future
- **Prescriptive**
What should we do next

The screenshot shows a web browser window displaying a blog post from Harvard Business School Online. The URL in the address bar is <https://online.hbs.edu/blog/post/types-of-data-analysis>. The page header includes the Harvard Business School Online logo and navigation links for Courses, For Organizations, Insights, and More Info. The main content area features a teal background with the title "Business Insights" and a sub-header "Harvard Business School → HBS Online → Business Insights". Below this, a paragraph states: "Harvard Business School Online's Business Insights Blog provides the career insights you need to achieve your goals and gain confidence in your business skills." A search bar is visible on the left side of the page. The main article title is "4 TYPES OF DATA ANALYTICS TO IMPROVE DECISION-MAKING". Below the title, there are sections for "TOPICS" and "COURSES". A "Subscribe to the Blog" section includes an email input field. The article image shows a person looking at a computer monitor displaying data charts. A chatbot bubble is visible in the bottom right corner of the page, asking "Hey there! Can I help with your course research?".

Descriptive Analytics

Examples of methods to learn what happened

Method	Examples	Summary
Cluster Analysis	K-Means Wards Hierarchical	group similar observations
Dimension Reduction	Principal Components Analysis	summarize data into key, underlying factors
Association Rules	Apriori FP-growth	find patterns (coinciding events)
Anomaly Detection	k-Nearest Neighbors Distance to Centroid	find patterns (outliers)



Diagnostic Analytics

Examples of methods to learn why something happened

Method	Examples	Summary
Regression with variable selection	Lasso ElasticNet Stepwise Regression	Explain demand by culling relevant drivers from a pool of many
Decision Trees	C&RT Random Forests	Explain demand (or event) based on compound rules derived from causal factors
Feature Importance & Explainability	Leave-one-feature-out (LOFO) Shapley Additive Explanations	In the context of a predictive model, explain what drove result
Anomaly detection	one-class SVM Isolation Forest	Define “normal” in order to identify unusual patterns

Predictive Analytics

Examples of methods to learn what might happen in the future

Method	Examples	Summary
Regression Analysis	Linear regression Logistic regression	Predict future demand or event occurrence with causal factors
Decision Trees	C&RT Random Forest	Predict future demand (or event) based on compound rules derived from causal factors
Time Series Methods	Decomposition Holt-Winters ARIMA	Predict future demand based on its historical pattern
Neural Networks	Feed-forward Network	Predict future demand by recognizing complex patterns of demand and causal factors

Prescriptive Analytics

Examples of methods to learn what we should do next

Method	Examples	Summary
Optimization Algorithms	Linear Programming Dynamic Programming	Find the best solution from all possible choices, given a goal and constraints
Reinforcement Learning	Q-Learning SARSA	Learns by trial and error with rewards and penalties
Simulation Methods	Monte Carlo simulation	Randomly sample inputs to estimate possible outcomes of a problem (model)
Causal Inference & Uplift Models	Instrumental Variables Causal Trees/Causal Forests T-Learner, S-Learner, X-Learner	determine whether one thing actually causes another; predict the impact of an action on an outcome.

How do neural networks and large language models relate to AI?

The screenshot shows a web browser window displaying the HarvardX CS50AI course page. The page content includes a list of tasks for students, such as watching lectures and submitting projects. A blue box highlights the word "Search" in the first task. To the right of the screenshot, seven blue-bordered boxes provide definitions for key terms: Search, Knowledge, Uncertainty, Optimization, Learning, Neural Network, and Language. The browser's address bar shows the URL learning.edx.org/course/course-v1:HarvardX+CS50AI+1T2020/home. The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system tray on the right shows the date and time as 11:01 AM on 3/13/2025.

learning.edx.org/course/course-v1:HarvardX+CS50AI+1T2020/home

7. If you're the **Facebook** type:

- join CS50's Facebook Group at facebook.com/groups/cs50
- like CS50's Facebook Page at facebook.com/cs50
- follow (and say hello to!) Brian at facebook.com/BrianYu28
- follow (and say hello to!) David at facebook.com/dmalan

8. If you're the **Snapchat** type:

- add cs50

After that, watch each of the course's seven lectures and submit each of the course's seven projects.

1. Watch **Search**. Test your understanding with Quiz 0 and submit Project 0.
2. Watch **Knowledge**. Test your understanding with Quiz 1 and submit Project 1.
3. Watch **Uncertainty**. Test your understanding with Quiz 2 and submit Project 2.
4. Watch **Optimization**. Test your understanding with Quiz 3 and submit Project 3.
5. Watch **Learning**. Test your understanding with Quiz 4 and submit Project 4.
6. Watch **Neural Networks**. Test your understanding with Quiz 5 and submit Project 5.
7. Watch **Language**. Test your understanding with Quiz 6 and submit Project 6.

And if you have any questions, start a [discussion](#) with classmates!

CS50AI 2024

CS50AI 2020 (old)

<https://www.edx.org/cs50>

Wed, Mar 12, 2025

Harvard Extension School

Search: Finding a solution to a problem, like a navigator app or figuring the next move in a game

Knowledge: Representing information and drawing inferences from it

Uncertainty: Dealing with uncertain events using probability

Optimization: Finding the *best* way to solve a problem

Learning: Improving performance based on access to data and experience

Neural Network: A program structure inspired by the human brain to perform tasks

Language: Processing natural language used by humans

Neural networks in forecasting workflows

- Many workflows for short-term forecasting rely on univariate time-series models
- When longer-term forecasts are required, cause-and-effect models may generate more reliable forecasts
- When relationships between causal variables (predictors) and the result (demand forecast) are nonlinear or subject to interactive rules, neural networks can provide additional insights
- Neural networks are especially useful for “hard to forecast” products that are of significant value to the company



Should all products be forecasted in the same way?

“Products differ widely, some have long, stable demand histories, while others are new, volatile, or exhibit intermittent patterns...”

Segmenting a product portfolio for demand forecasting is essential because it enables businesses to tailor their predictions to the unique characteristics of each product group.”

The Forecasting Trap: Why Segmentation Is Your Way Out



Charles Chase
Executive Advisor/Thought Leader/Author/Speaker



October 7, 2025

The forecasting trap refers to the common mistake of applying a single forecasting model or approach across all products, regardless of their demand behavior, lifecycle stage, or data availability. It's the assumption that one method can fit all, which often leads to poor accuracy, wasted resources, and misguided decisions. Forecasting every product the same way is a common mistake that leads to inefficiency and inaccuracy. Products differ widely, some have long, stable demand histories, while others are new, volatile, or exhibit intermittent patterns.





*Large Language Models (LLM's)
are not forecasting engines, they
play a supporting role*



LLM's in a forecasting workflow can...

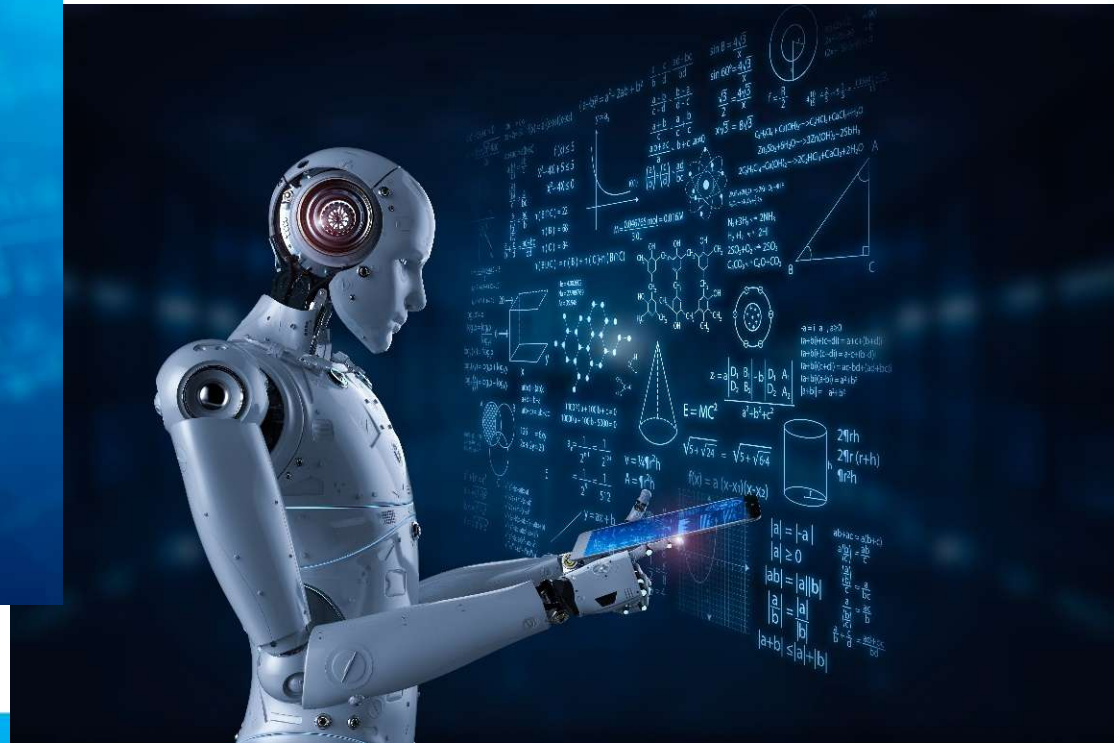
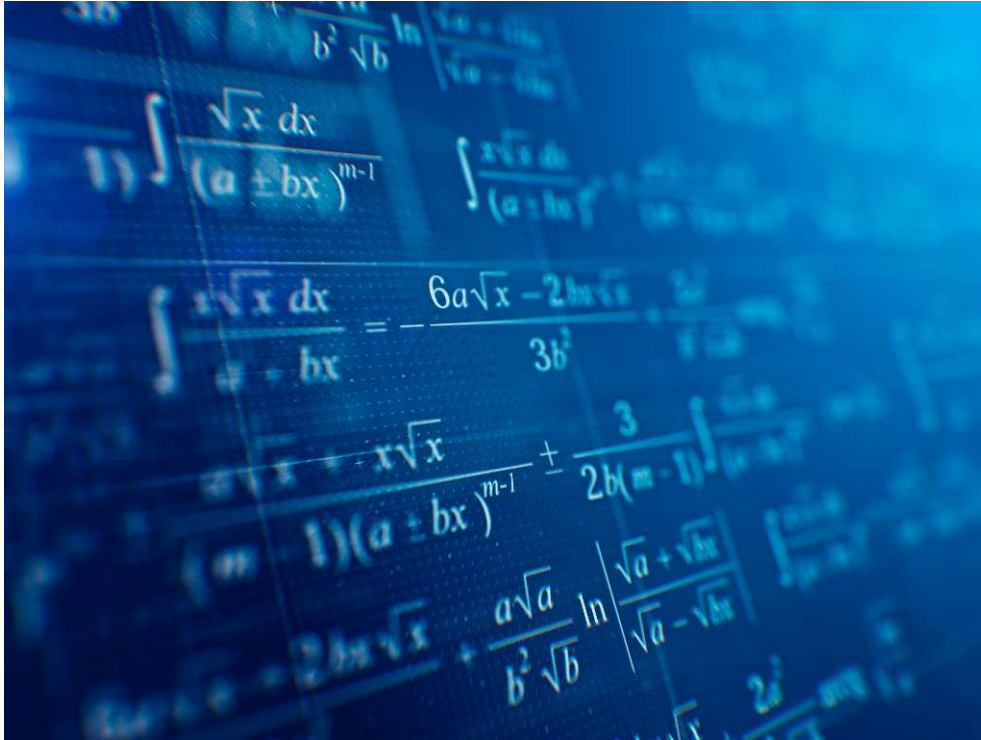
- ✓ Translate data into business insights and narratives
- ✓ Assist with forecast review explanations and reporting
- ✓ Support collaboration across teams (e.g., summarizing inputs, risks, assumptions)
- ✓ Enable faster interaction with data (natural language queries, copilots)
- ✓ Help to structure scenario discussions and decision frameworks

LLM's in a forecasting workflow should not...

- ✗ be used as primary statistical forecasting models
- ✗ replace time-series, causal, or ML-based forecasting approaches



Statistics vs. Machine Learning



Quantitative Analysis

Vocabulary for Statistics vs. Machine Learning

Statistics	Machine Learning
Prediction	Supervised Learning
Description	Unsupervised Learning
Independent Variables (x) <i>(predictors)</i>	Features, Inputs
Dependent Variable (y)	Target or Training Value
Predicted (Fitted) Dependent Variable (\hat{y})	Output
Constant	Bias
Residuals	Errors

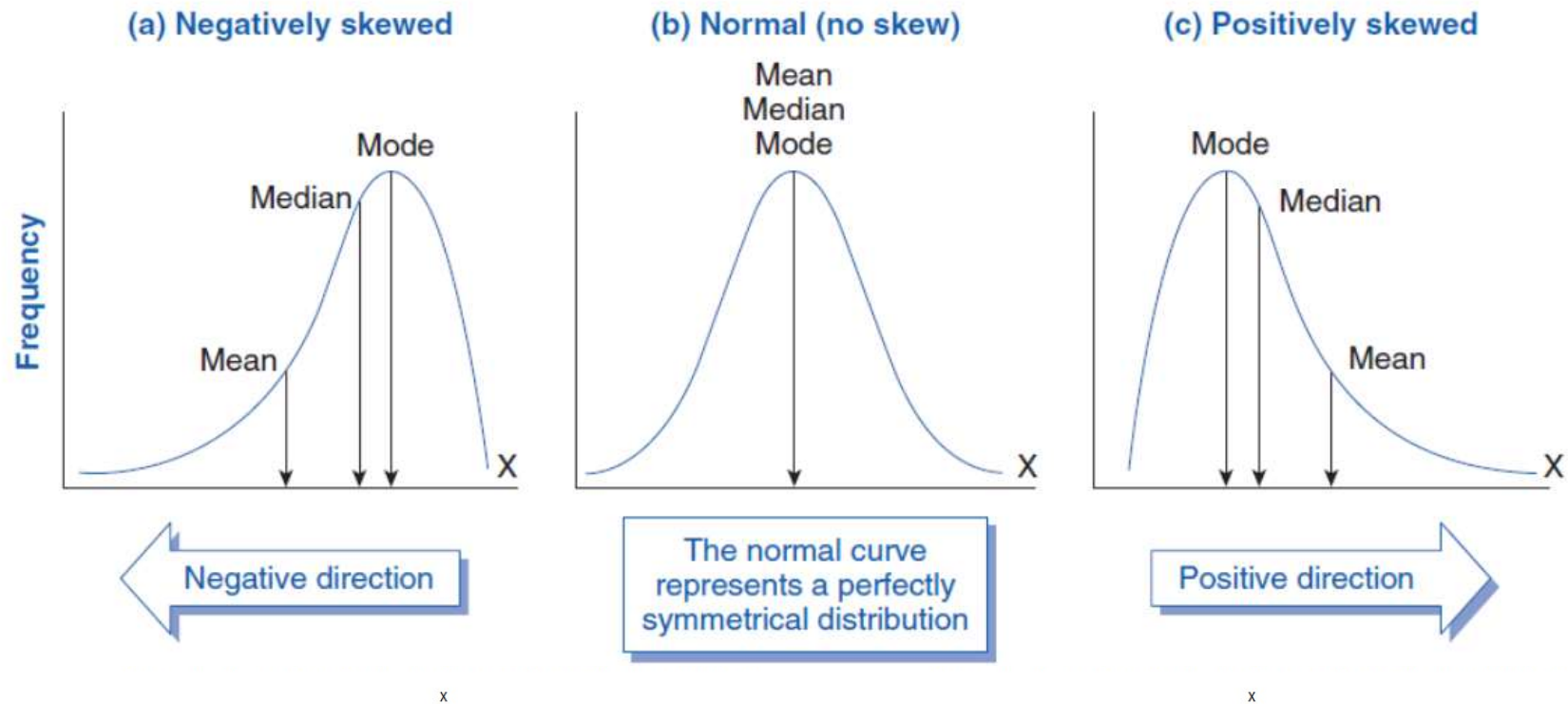
Some say that many of machine learning's techniques are simply "borrowed" from statistics...

Cause & Effect Forecasting I

Linear Regression



Distributions



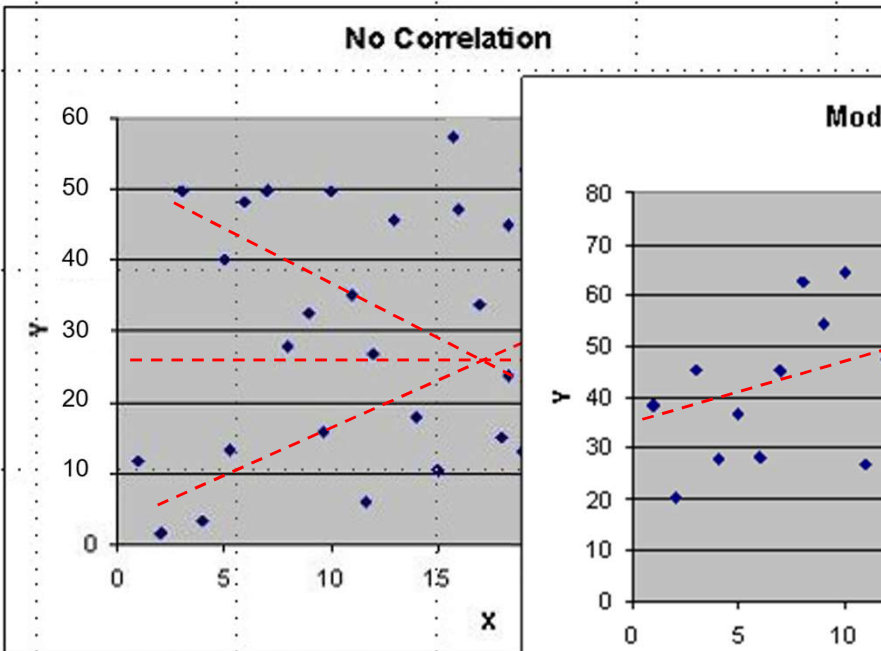
<https://www.biologyforlife.com/skew.html>

Correlation

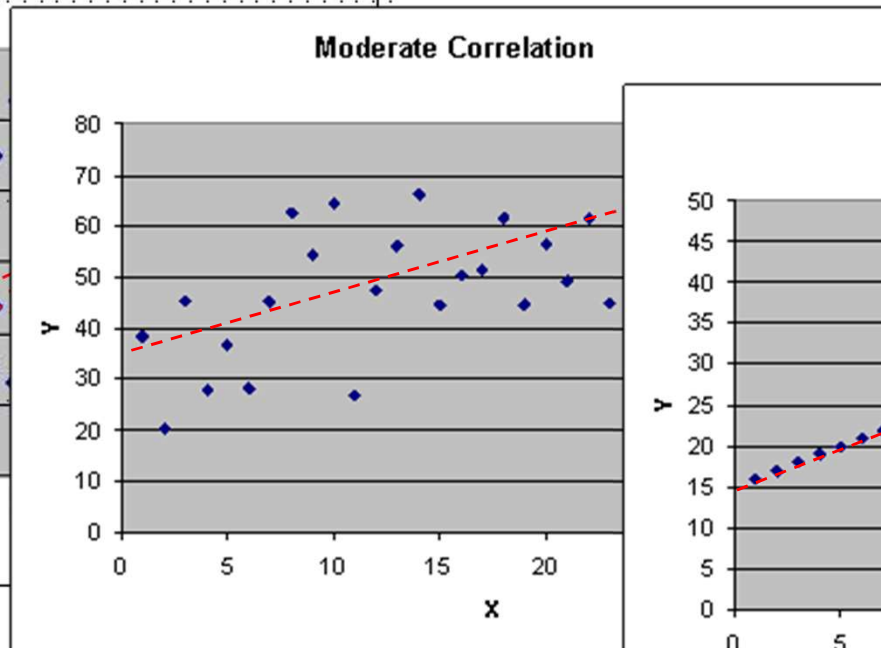
a measure of linear coincidence

Correlation

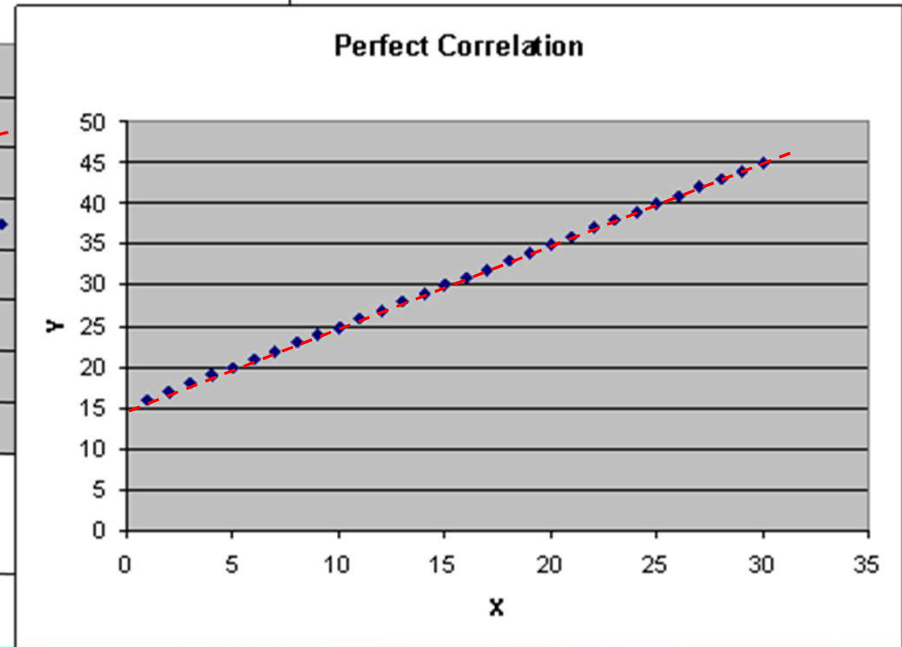
No Correlation



Moderate Correlation



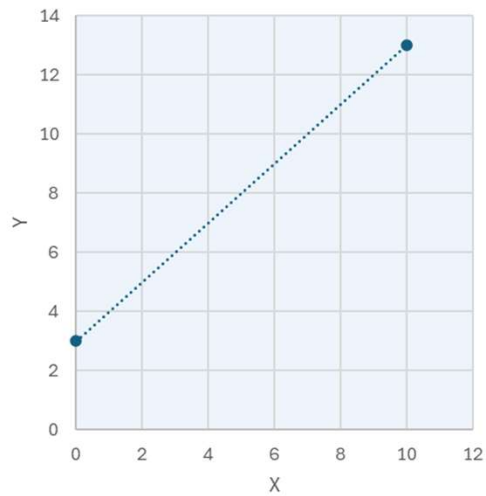
Perfect Correlation



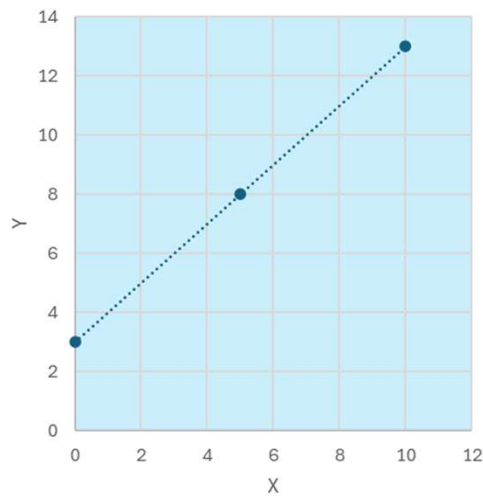
Regression Refresher

30000

Two Points Define a Line

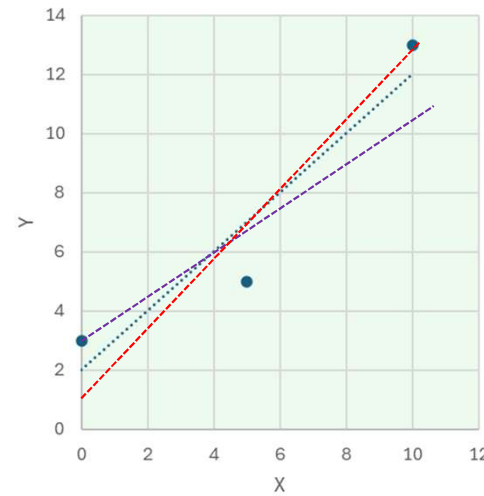


Three Points Determine It...

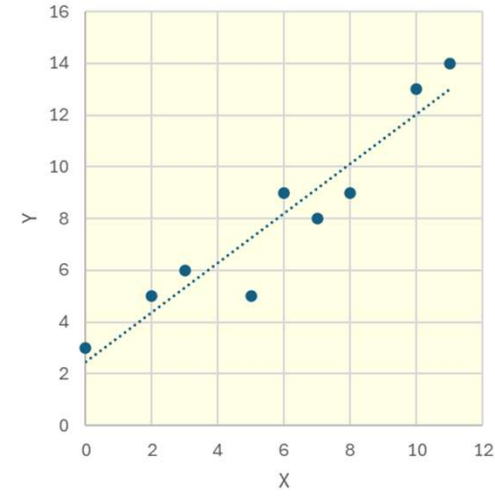


Linear Regression

...Or Require Regression

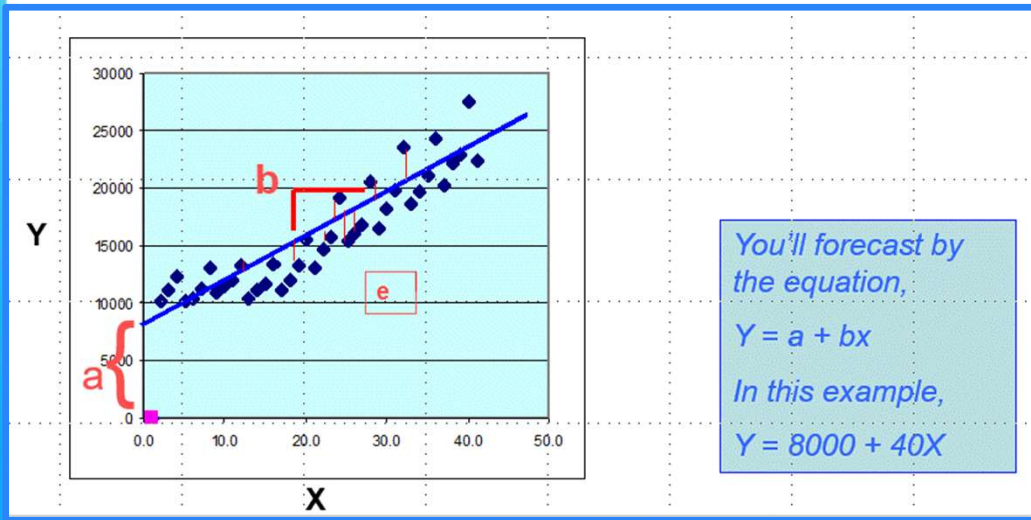


...With Many Degrees of Freedom

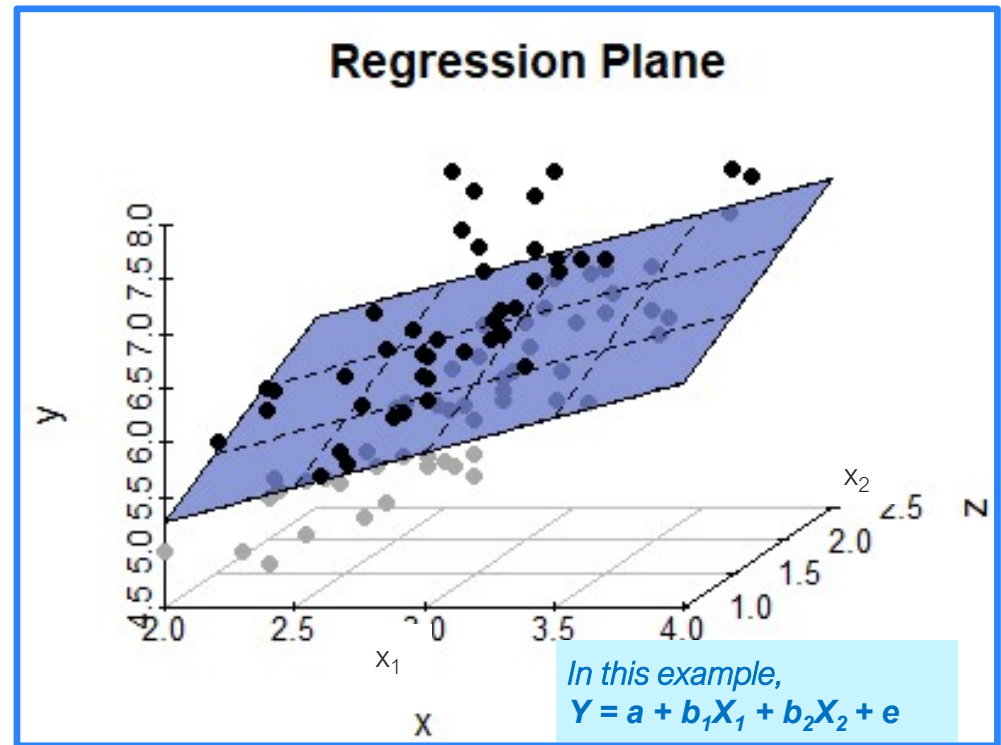


Linear Regression

Estimate an equation to predict Y from one or more X's



You'll forecast by the equation,
 $Y = a + bx$
 In this example,
 $Y = 8000 + 40X$



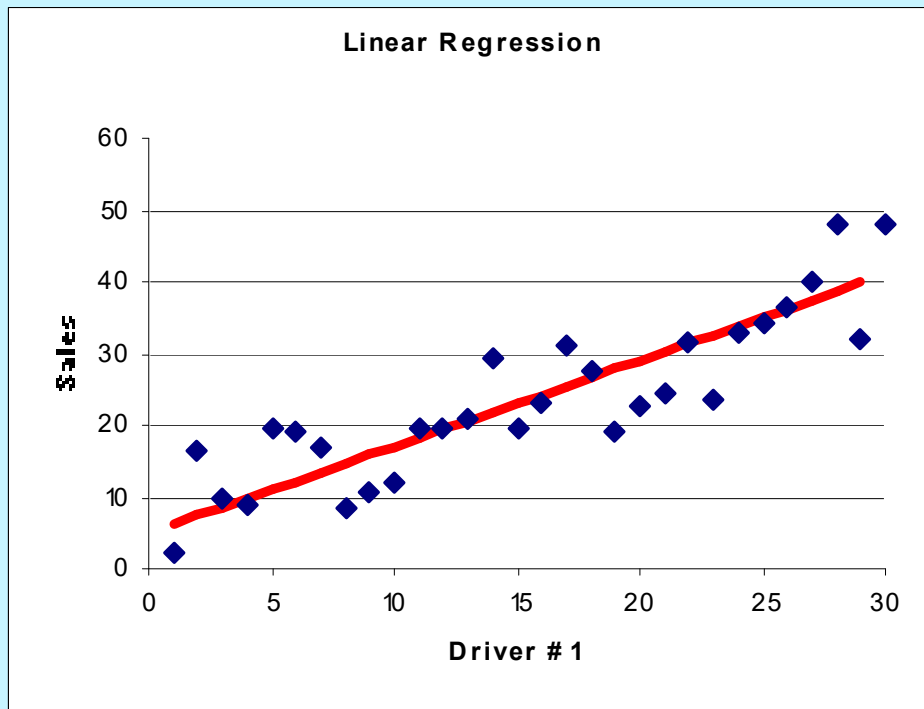
In this example,
 $Y = a + b_1X_1 + b_2X_2 + e$

modified from
<https://stackoverflow.com/questions/47344850/scatterplot3d-regression-plane-with-residuals>

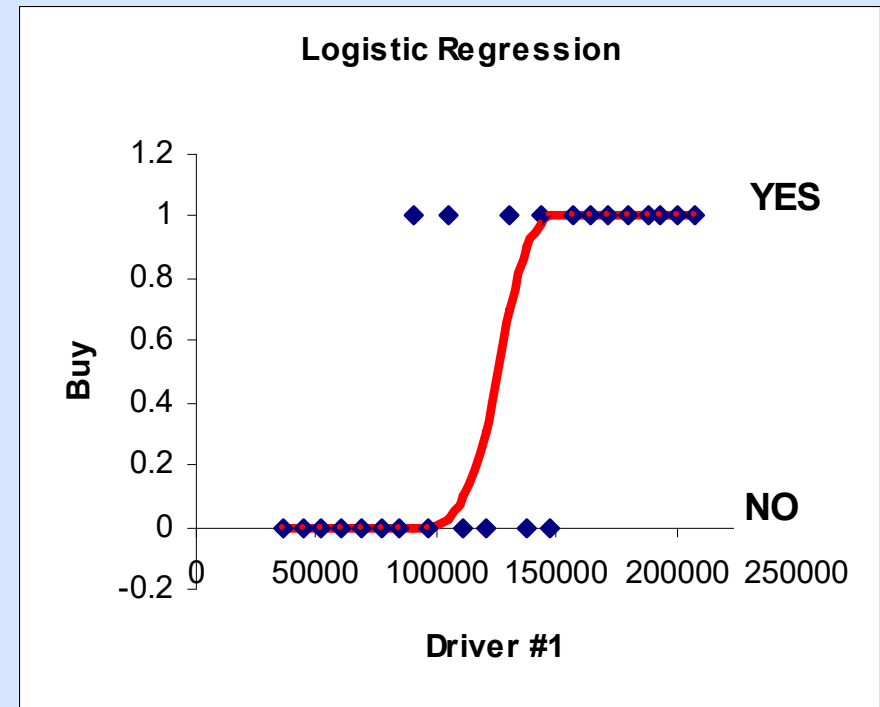


By the way...

PREDICTING LEVELS vs. EVENTS



Level: Scalar or Continuous



Limited: Event Yes/No

Predictive Model Performance

Mean Average Percent Error (MAPE)



Institute of Business
Forecasting & Planning



Events



Training



Assessments / Advisory



Certification



Knowledge



EmployAbility



Job Board

Home > Knowledge > Glossary > MAPE (Mean Absolute Percentage Error)

MAPE (Mean Absolute Percentage Error)

What is MAPE? It is a simple average of absolute percentage errors. The MAPE calculation is as follows:

$$\text{MAPE} = \frac{\sum \frac{|A-F|}{A} \times 100}{N}$$

Here A= Actual, F= Forecast, N= Number of observations, and the vertical bars stand for absolute values.

For intermittent demand (many Actuals=0), consider **Mean Absolute Error**, a.k.a. **Mean Absolute Deviation (MAD)** = $\frac{\sum |A-F|}{N}$;
see <https://robjhyndman.com/papers/foresight.pdf>



Institute of Business
Forecasting & Planning

FOSTERING DEMAND PLANNING,
FORECASTING, S&OP &
ANALYTICS FOR 40+ YEARS

Forecasting Process

is the same whether your model is linear regression or a neural network

#	Step	Details
1	Assemble data...	...for what you wish to forecast (dependent variable or target) and the predictors (independent variables or features) that you think are associated
2	Separate “ex post” holdouts	<ul style="list-style-type: none"> • “training” and “test” data • Be wary how a model fits vs. how it forecasts (MAPE, MAE) • Historic random variation can be mistaken for “the real thing” • Hence, a predictive model should be tested on data that did not create it
3	Fit the model with “training” and “test”	And see what it did with those <i>ex post</i> holdouts!
4	Re-fit with training and test data combined so your data is as current as possible
5	Forecast (extrapolate)	Given future estimates of predictors

Cause & Effect Forecasting II

Neural Network



(Artificial) Neural Network

Defined

A neural network is a computational model inspired by the human brain, consisting of interconnected processing units that work together to recognize patterns and map inputs to outputs

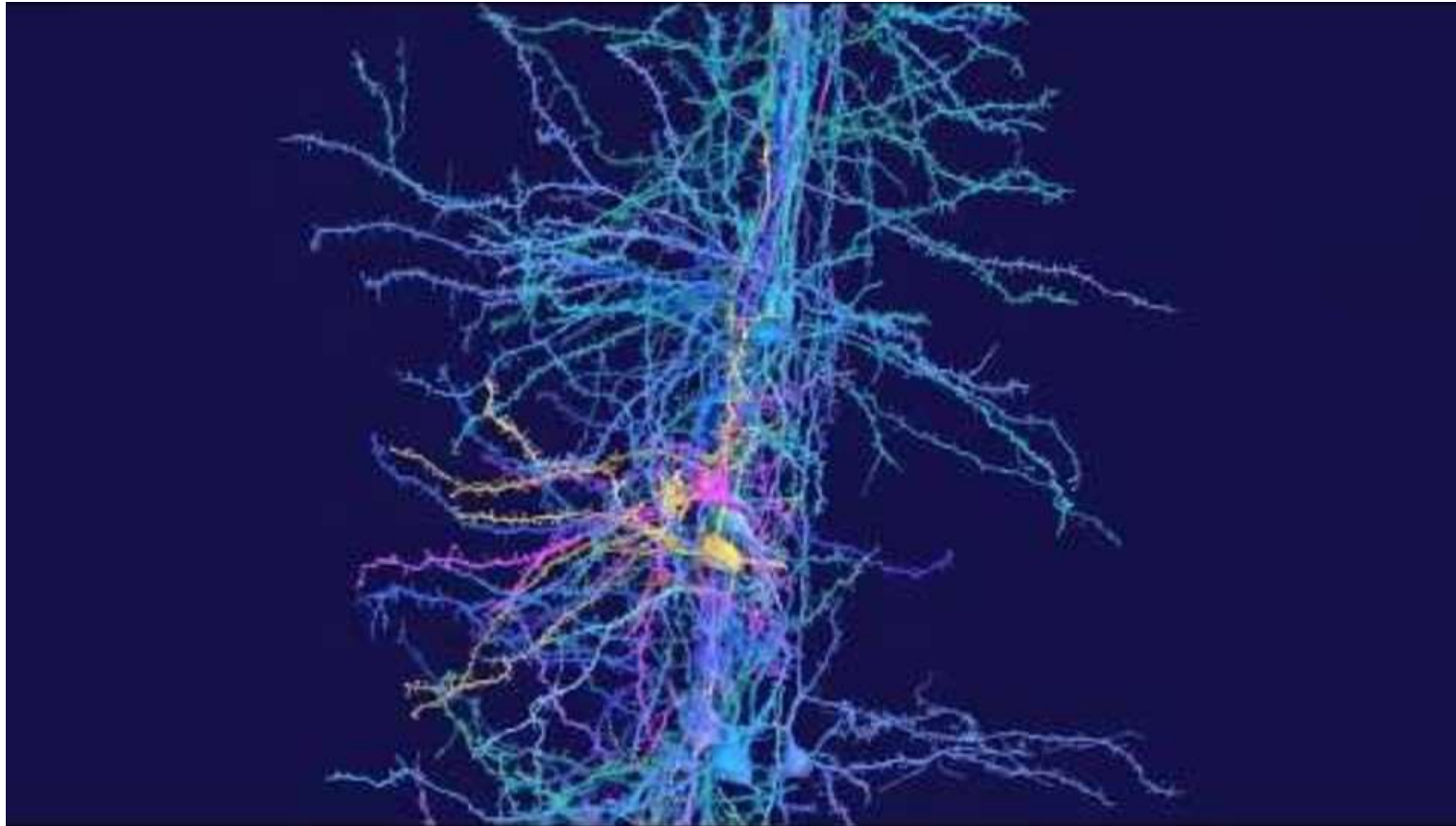
Citation (adapted from):

Haykin (2009); Russell & Norvig (2021); Goodfellow et al. (2016)

Neural Network

Comparison with Brain

Brain	Neural Network
100,000,000,000 neurons	Neurons number from few to hundreds or thousands
Pattern recognition	Pattern recognition
Vast amount of perceptions	Vast amount of data (or signals)
Learns relatively quickly	Learns relatively slowly
Parallel processing of multiple inputs is aggregated through synapses between cells	Parallel processing of multiple inputs is aggregated through arithmetic combinations between nodes

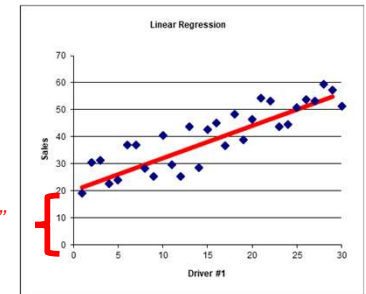
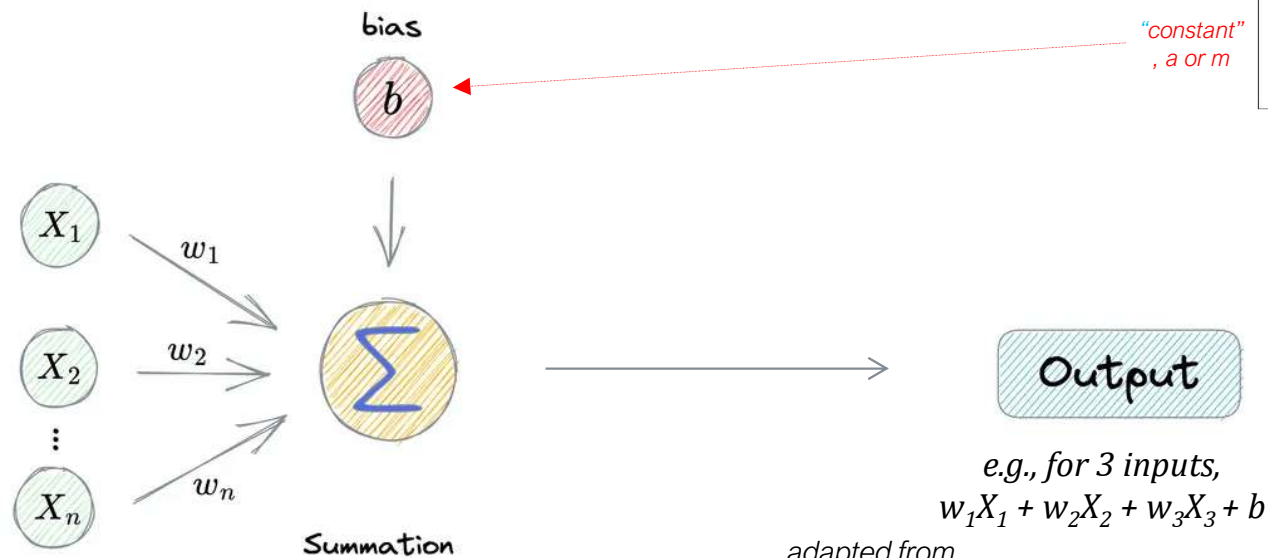


<https://youtu.be/jmdPOaTpZWA?si=jy-rbNJUhWe-JA10> per Horton JC, Adams DL. The cortical column: a structure without a function. *Philos Trans R Soc Lond B Biol Sci.* 2005 Apr 29;360(1456):837-62. doi: 10.1098/rstb.2005.1623. PMID: 15937015; PMCID: PMC1569491.

A Single Neuron

Can compute a linear combination that looks like linear regression...

Start with a linear weighted sum, where the weights are just like regression's β ...

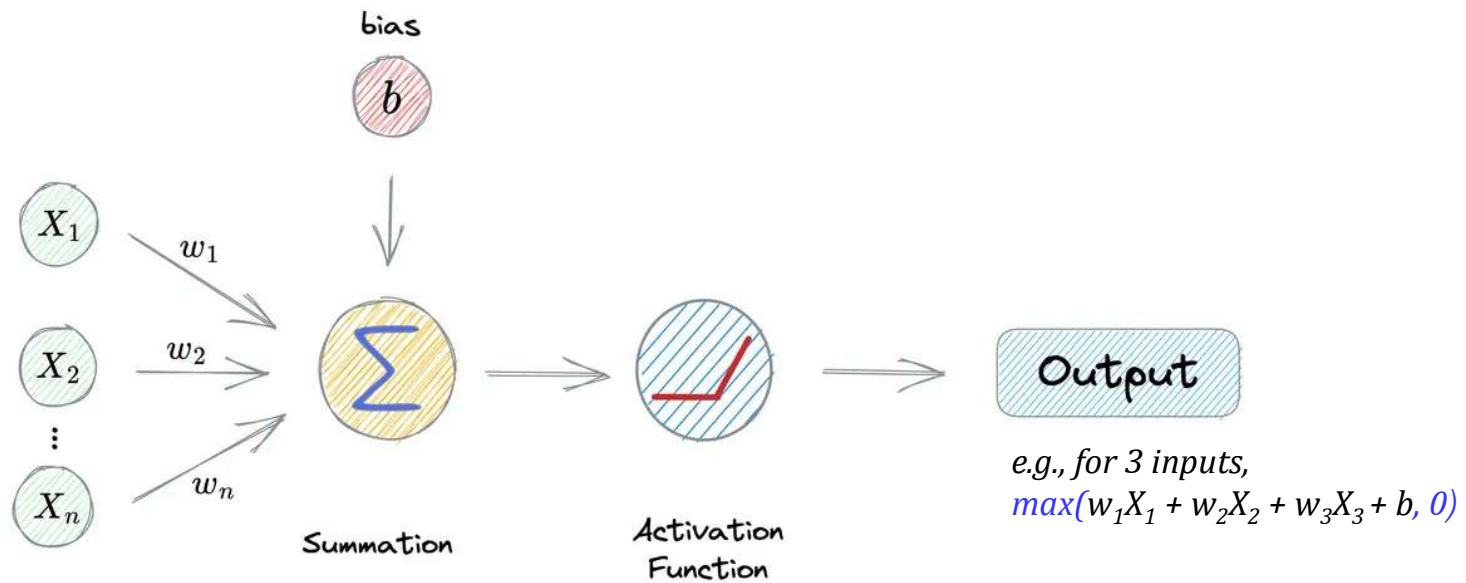


e.g., for 3 inputs,
 $w_1X_1 + w_2X_2 + w_3X_3 + b$

adapted from

<https://blog.dailydoseofds.com/p/an-intuitive-guide-to-non-linearity-9bb>

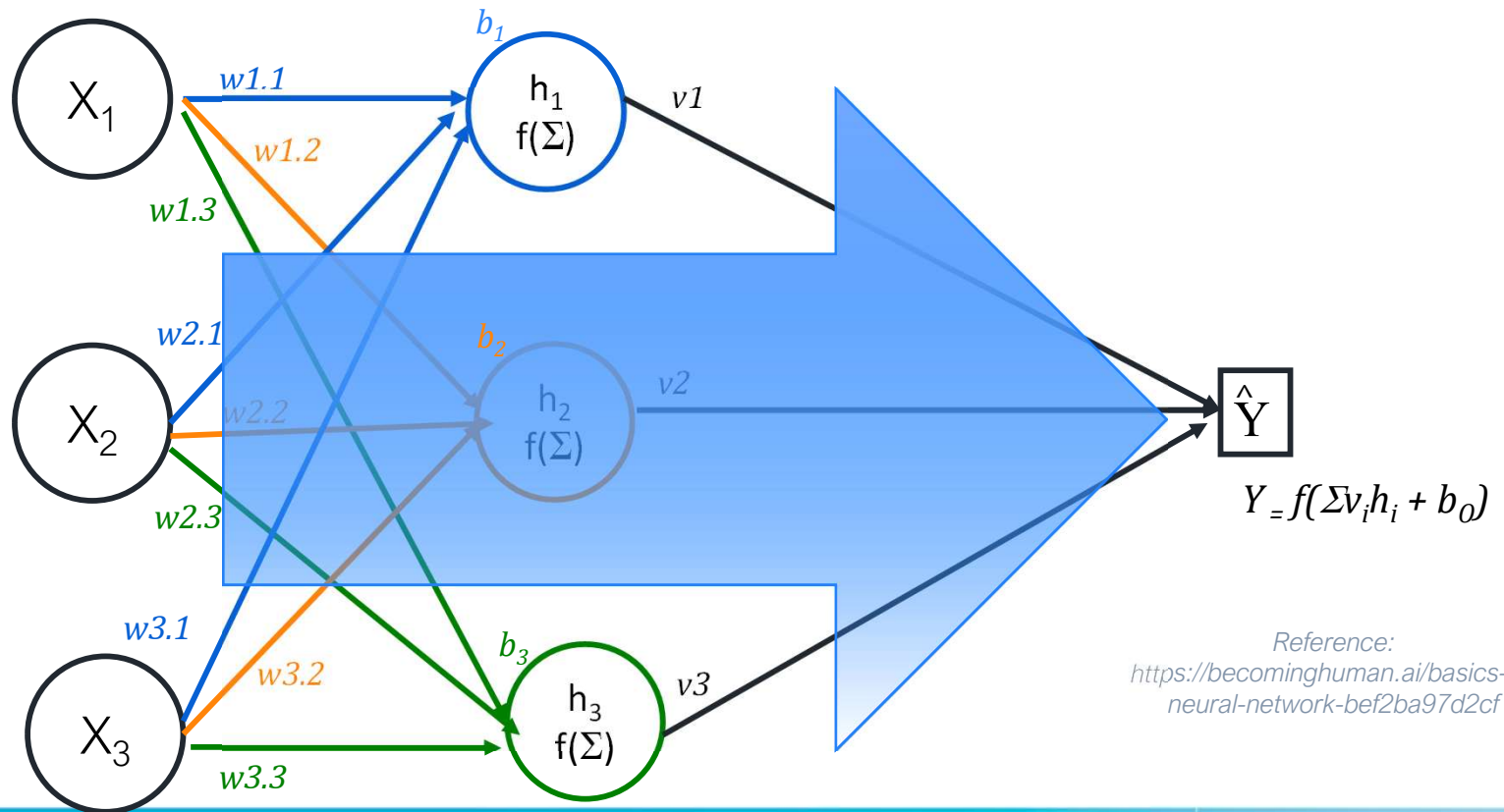
Add an Activation Function to impart non-linearity...



<https://blog.dailydoseofds.com/p/an-intuitive-guide-to-non-linearity-9bb>

Neural Net: Initialize

make a *guess* about the *weights*, which—in turn—makes a guess about the output (Y)



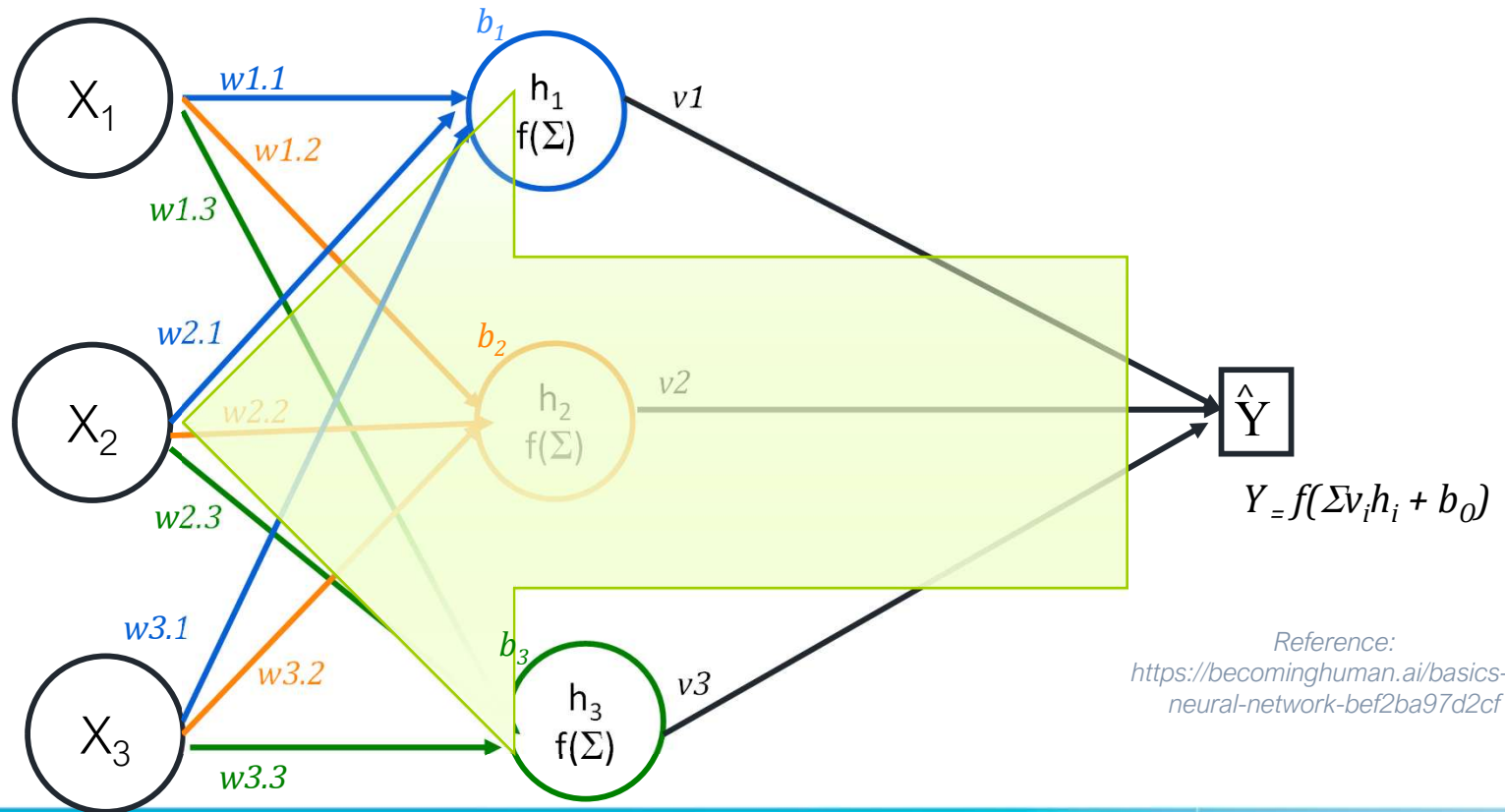
REMEMBER:
Each hidden node
 $h_i = f(\Sigma w_i X_i + b_i)$

Reference:
<https://becominghuman.ai/basics-of-neural-network-bef2ba97d2cf>

Backward Propagation

^

Recognize the extent of the error, to allow adjustment of a weight



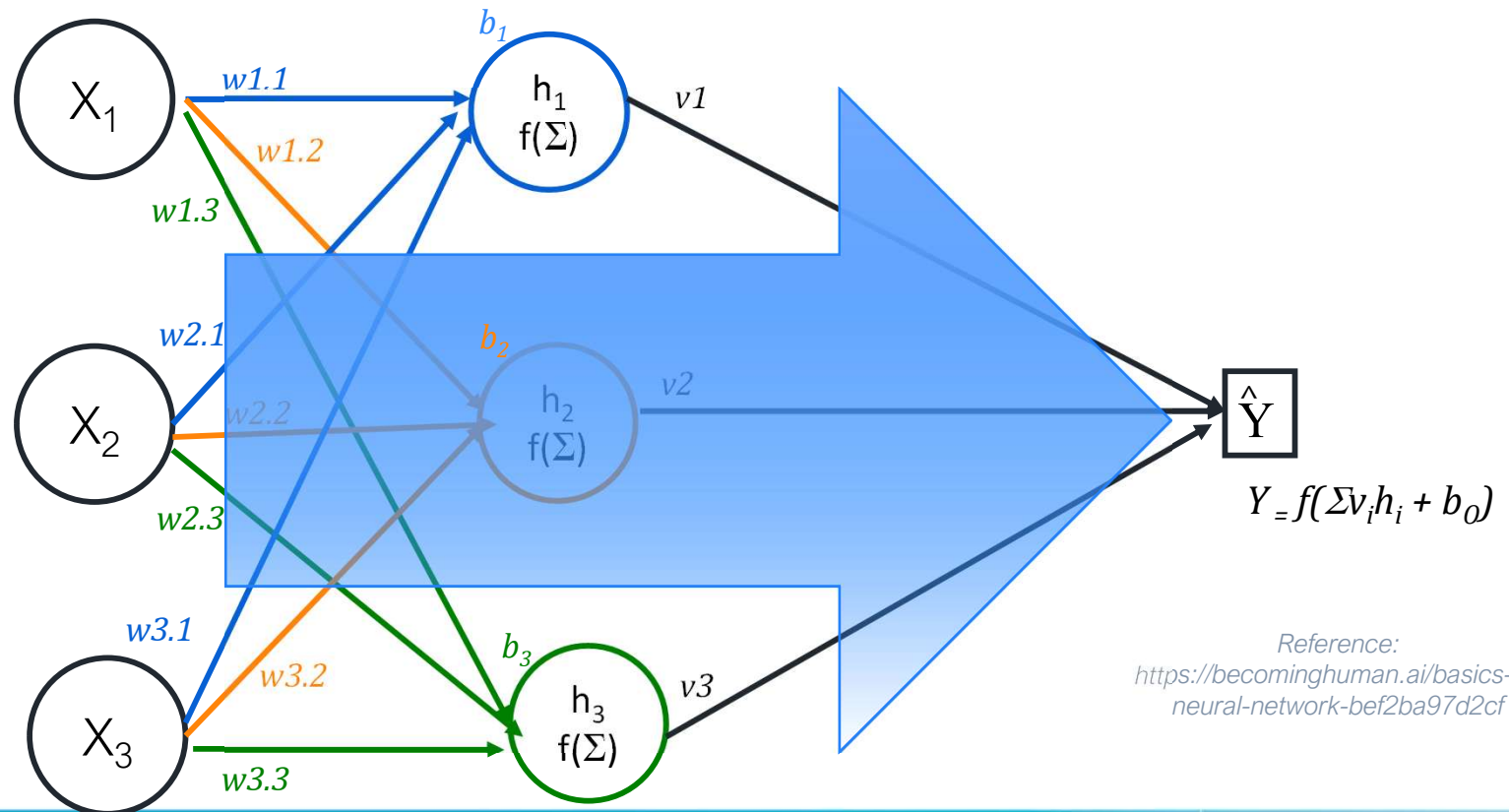
REMEMBER:
Each hidden node
 $h_i = f(\sum w_i X_i + b_j)$

Reference:
<https://becominghuman.ai/basics-of-neural-network-bef2ba97d2cf>

Feed-Forward

given weights, compute the output (Y)

^



REMEMBER:
Each hidden node
 $h_i = f(\sum w_i X_i + b_i)$

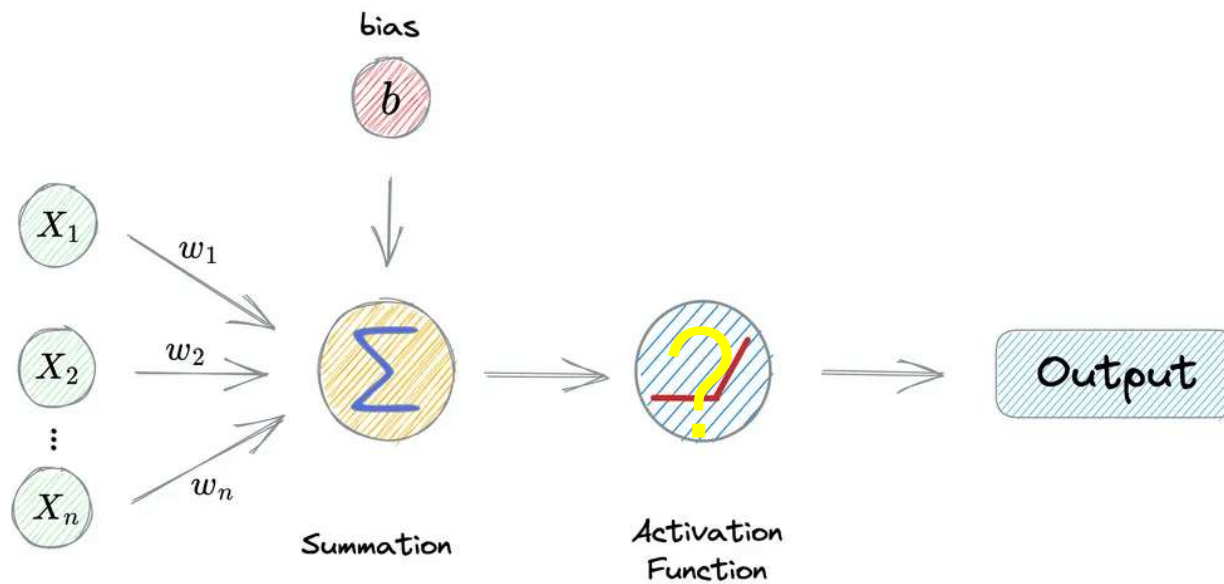
Reference:
<https://becominghuman.ai/basics-of-neural-network-bef2ba97d2cf>



May we please have a
Volunteer?



What just happened?



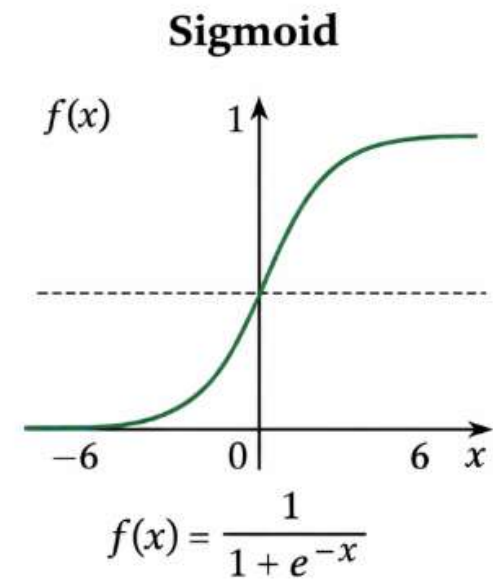
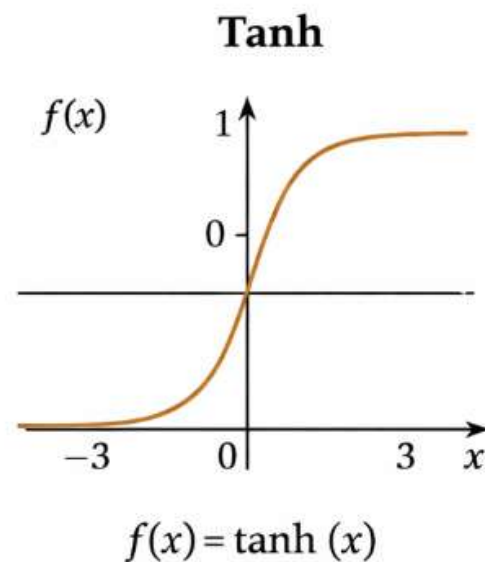
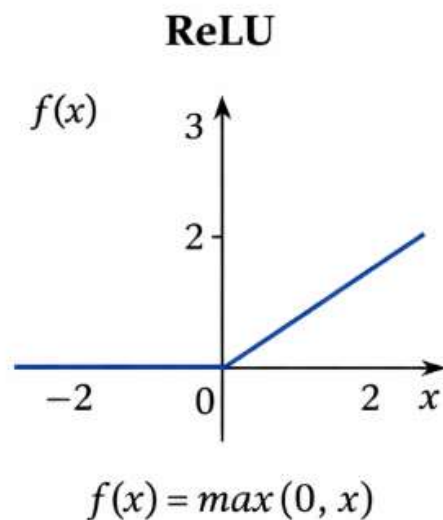
Neural Net: Nuts and Bolts

- **Partial derivatives** essentially tell you how changing the error can be attributed to the weights
- **Cost function**—like linear regression’s minimizing the sum of squared errors—is how those errors should be interpreted for optimization
- **Learning rate** is how big an adjustment you allow when a weight changes
- **Activation function** is how you process output that would otherwise be linear (e.g., to separate classification tasks)

Reference: <https://becominghuman.ai/basics-of-neural-network-bef2ba97d2cf>

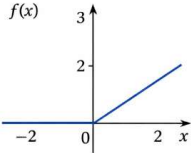
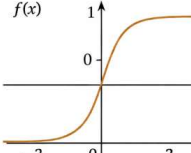
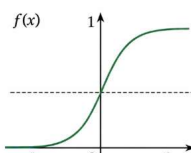
Neural Network's Advantage Over Regression

is Nonlinearity of Activation Function
within its hidden layers



Nonlinear Activation Functions

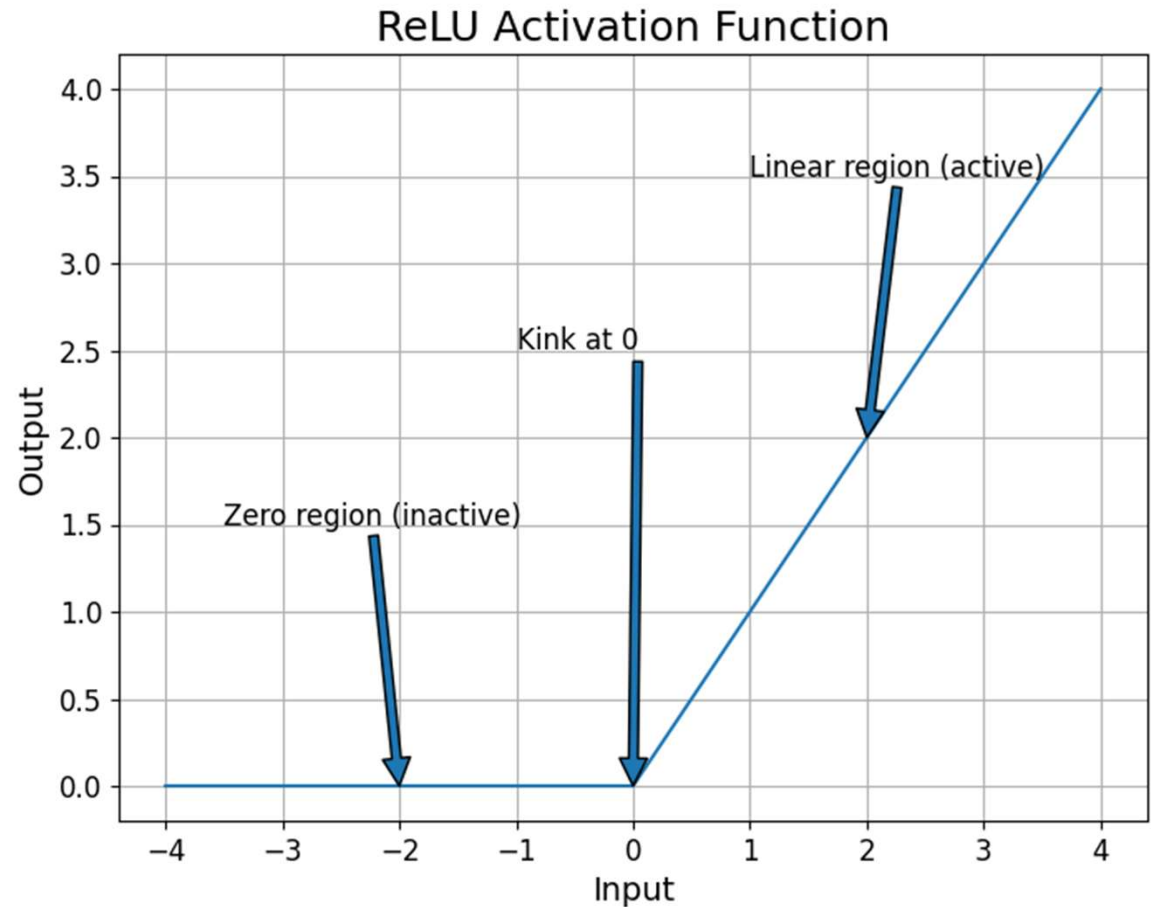
Examples

Non-linear function	Advantages
<p>ReLu</p>  <p>$f(x) = \max(0, x)$</p>	<ul style="list-style-type: none"> • Many ReLU nodes will combine to approximate a curvilinear surface with multiple inflection points • Works well in deep networks
<p>Tanh</p>  <p>$f(x) = \tanh(x)$</p>	<ul style="list-style-type: none"> • Output centered around 0 • Middle essentially linear, with extremes dampened
<p>Sigmoid</p>  <p>$f(x) = \frac{1}{1 + e^{-x}}$</p>	<ul style="list-style-type: none"> • Output between 0 and 1, yielding a probability for event prediction • Historically important in early neural networks

What is “ReLU”?

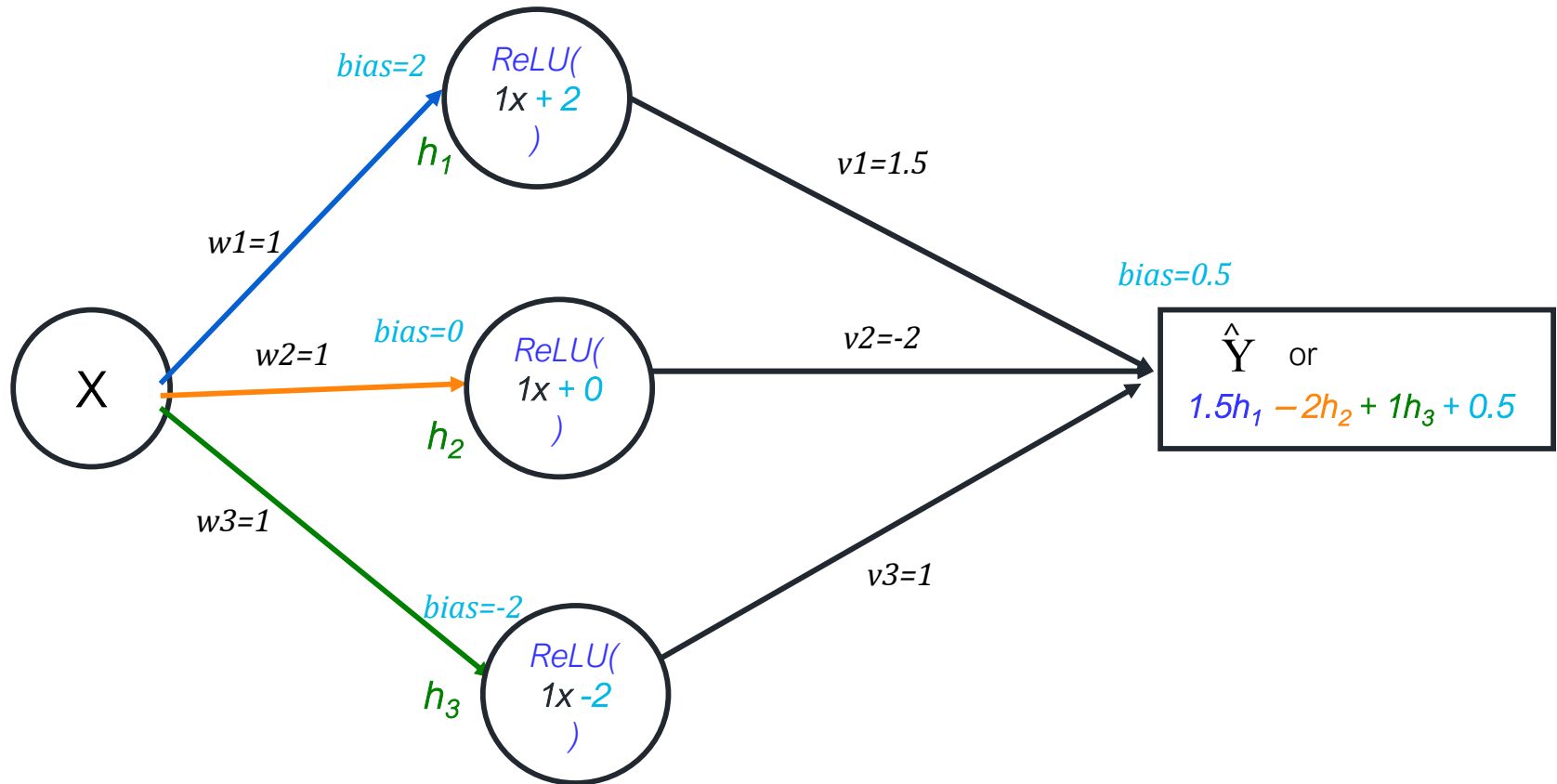
(Rectified Linear Unit)

- An activation function which returns a linear result only when it is positive
- We can think of this “positive only” result as a signal that is “turned on” only when certain conditions are true
- Those conditions are the combinations of X’s which—when the current weights and bias are applied—yield a positive result



ReLU example

1 input, 1 hidden layer, 1 output



ReLU Example

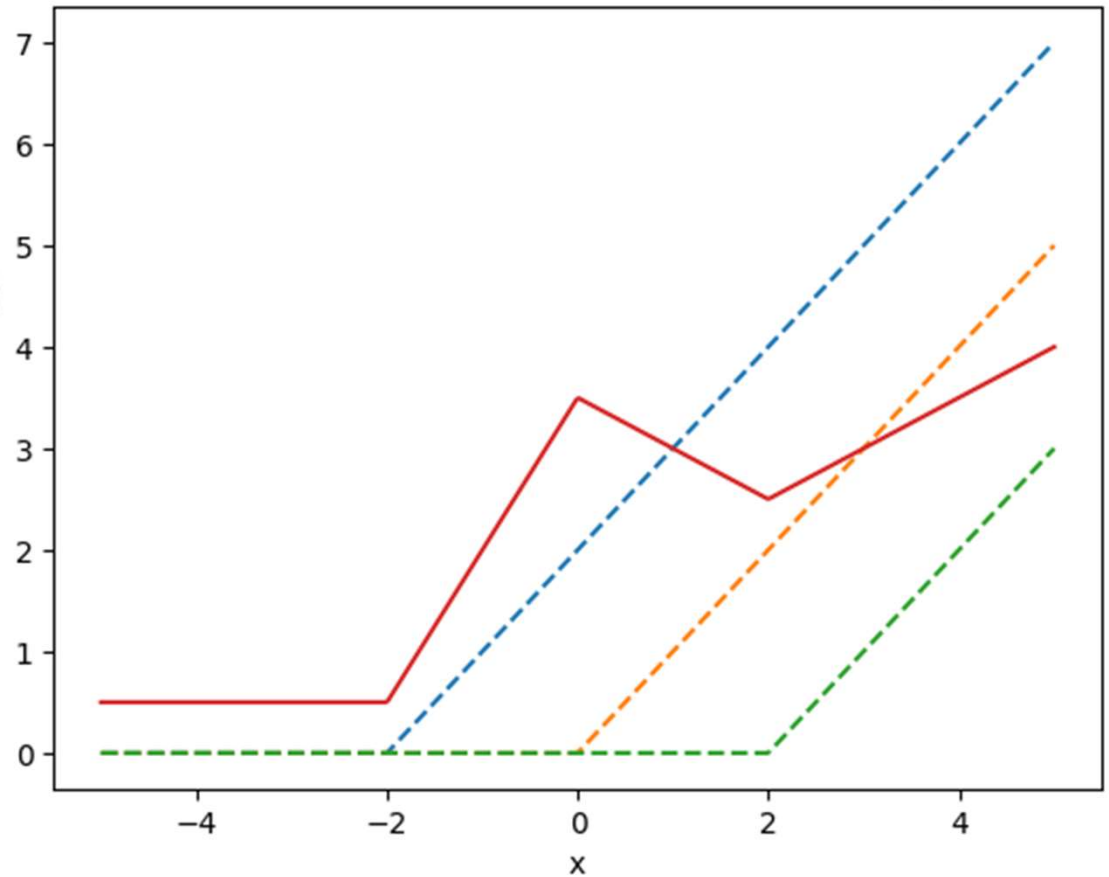
--- $h_1 = \text{ReLU}(x + 2) \rightarrow$ “turns on” at $x = -2$

--- $h_2 = \text{ReLU}(x) \rightarrow$ “turns on” at $x = 0$

--- $h_3 = \text{ReLU}(x - 2) \rightarrow$ “turns on” at $x = 2$

— $Output = 1.5h_1 - 2h_2 + 1h_3 + 0.5$

Three ReLU Nodes and Their Combined Output



ReLU Example

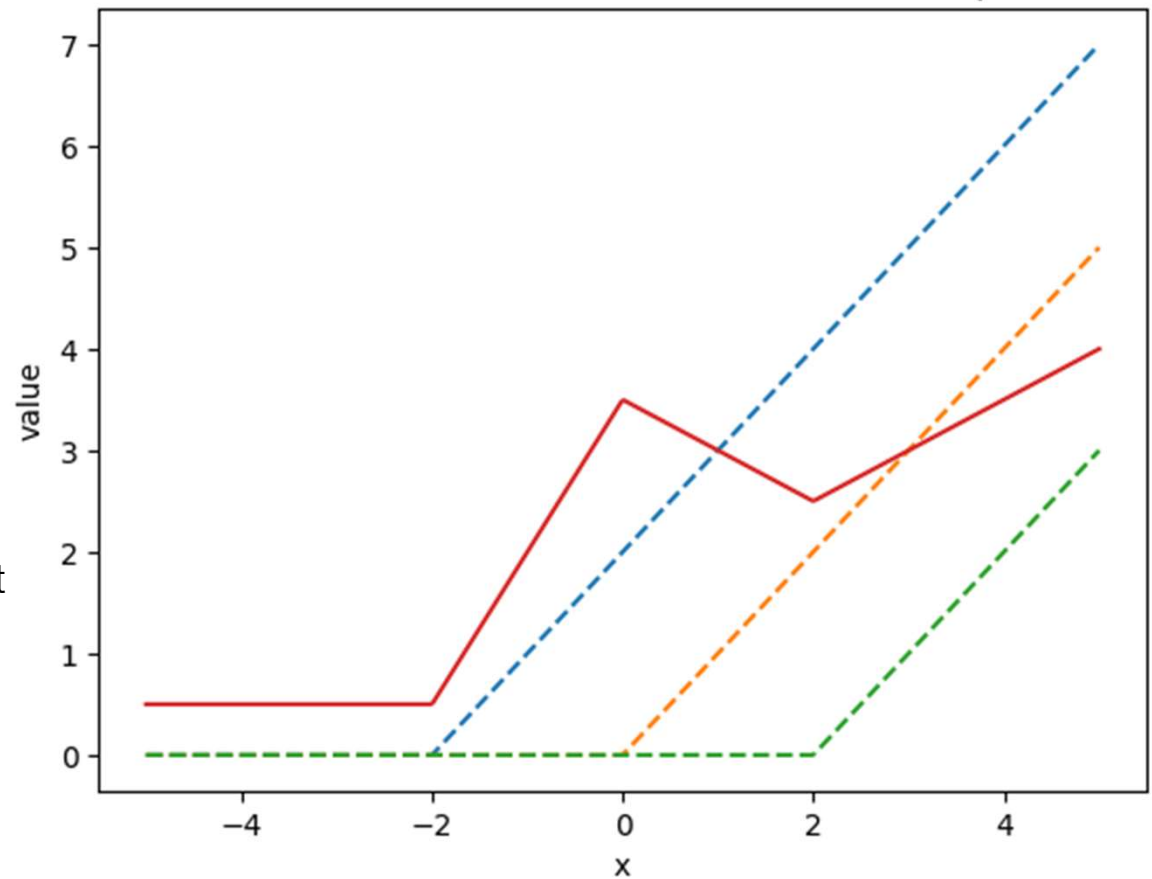
Each ReLU node contributes:
“After my cutoff point, I change the slope of the function”

Assembly of network result:

1. Start with a **baseline** (the **constant +0.5**)
2. Add **slope changes** at different points
3. **Build a curve** by stitching together straight segments

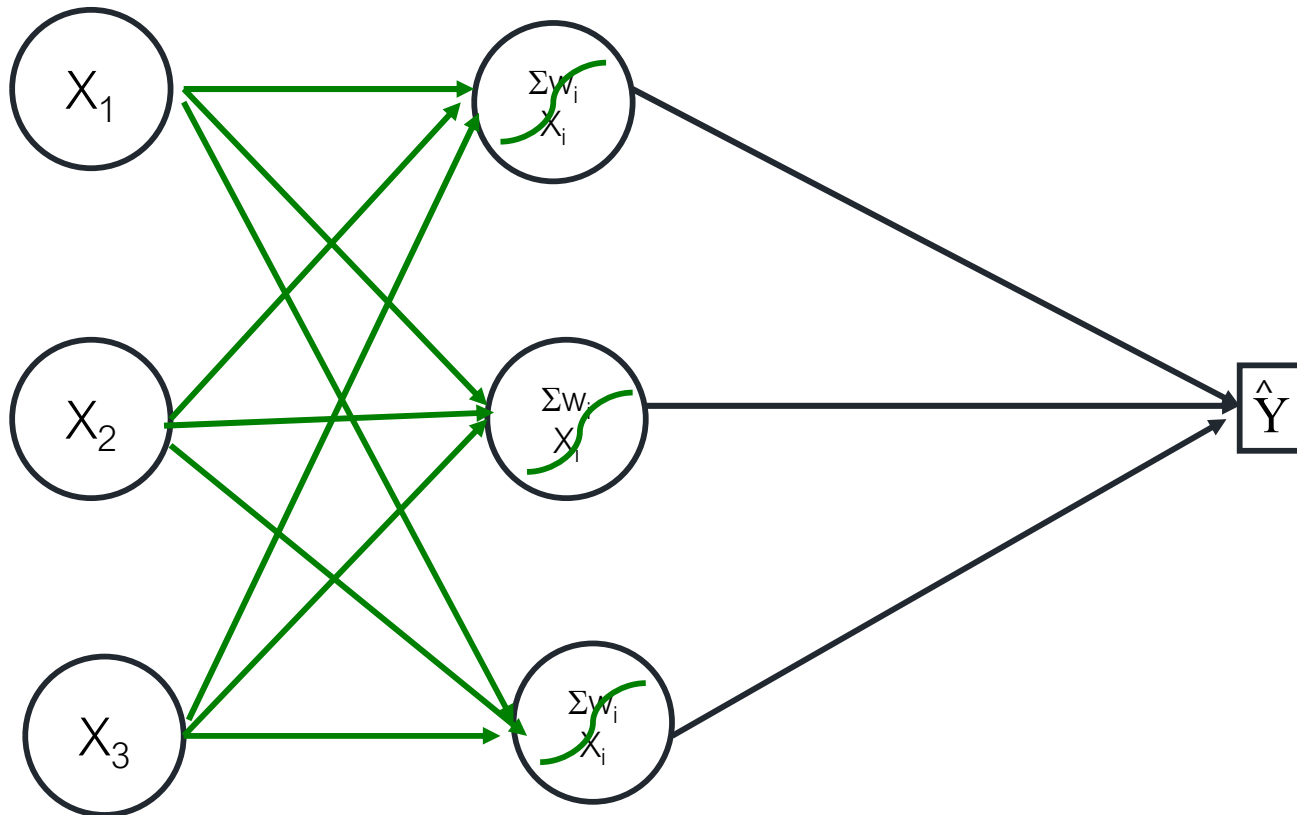
NOTE: the more segments, the smoother and more curve-like the result

Three ReLU Nodes and Their Combined Output



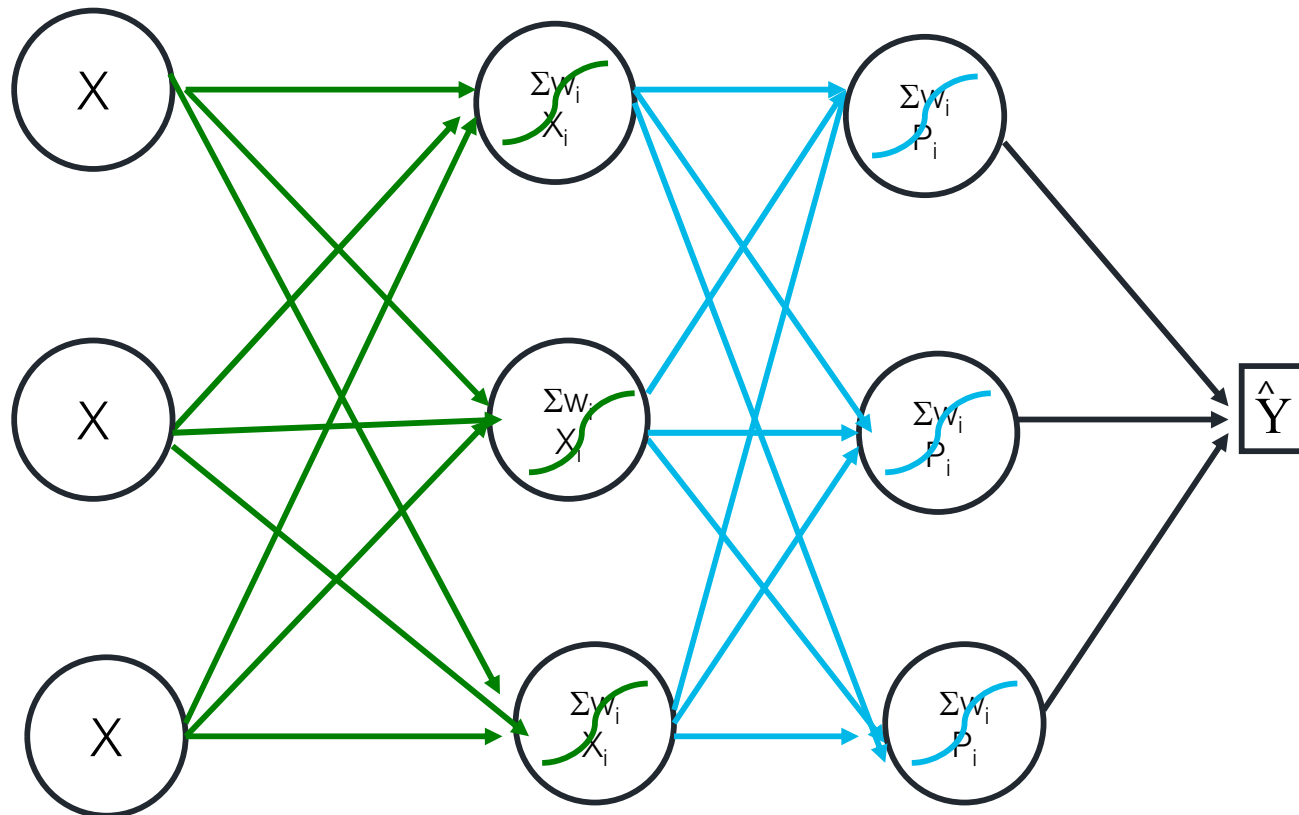
Conventional Depiction

3-input network with 1 hidden layer of 3 nodes



Conventional Depiction

3-input network with 2 hidden layers of 3 nodes each



Excel: Data setup for Tiberius (You can paste in your own!)

	A	B	C	D	E	F	G	H	I	J	K	L
1	bias	TIME	WASA	EMPL	QTR_4	BDLG						
2	1	1	1193.3	87973	0	9392						
3	1	2	1217.3	90021	0	13637						
4	1	3	1254.4	90120	0	14392						
5	1	4	1284.7	91180	1	13301						
6	1	5	1319.8	89832	0	10411						
7	1	6	1335.1	90668	0	13057						
8	1	7	1360.1	89850	0	13621						
9	1	8	1411.6	91276	1	13354						
10	1	9	1451.7	89964	0	11056						
11	1	10	1478.1	91566	0	14559						
12	1	11	1512.6	91405	0	14067						
13	1	12	1530.6	91691	1	12286						
14	1	13	1542.7	89341	0	9658						
15	1	14	1563.9	90310	0	13659						
16	1	15	1579.8	89324	0	12351						
17	1	16	1586.0	89436	1	12307						
18	1	17	1610.7	88815	0	10508						
19	1	18	1643.5	89920	0	16287						
20	1	19	1671.3	90480	0	16697						
21	1	20	1715.4	92367	1	15523						
22	1	21	1755.8	91637	0	13540						
23	1	22	1793.1	94256	0	19190						
24	1	23	1819.5	95021	0	18827						
25	1	24	1847.7	96547	1	17215						
26	1	25	1882.7	95450	0	13848						
27	1	26	1939.8	97657	0	20319						
28	1	27	1976.0	97985	0	20403						
29	1	28	2012.8	99383	1	19179						
30	1	29	2044.1	98213	0	16790						

"bias" column must be populated with value of 1 for every observation

Target (dependent variable) must be in rightmost column

No missing values are permitted (for now)

Excel files at forecastMethods.com



Forecast Methods for Workshops and Training

Developed by Ceres Analytics

Welcome!

[Jupyter Python Notebooks for IBF Workshop](#)

[Web app for IBF Next Level Forecasting](#)

Downloads to run neural networks in Excel:

- [neuralNet_cfTiberius.xlsm](#)
- [HowToRun_neuralNet_cfTiberius.pptx](#)



neural network software - Microsoft Internet Explorer

Address: http://www.philbrierley.com/main.html?code/vba.html&codeleft.html

Phil Brierley
Software & Data Analysis Services

Select...

Visual Neural Data Mining

Visual Basic / Excel

(If you are looking for a fully functioning simulator you could also take a look at **TIBERIUS**, use the drop down list box above to find it.)

*Download and use this application for free!
There are screen shots at the bottom of this page!!
If you want the source code then it can be purchased!!!*

About

Tiberius for Excel is a feedforward multilayer perceptron trained with the backpropagation algorithm. It is written completely in VBA, the macro language of Microsoft Excel and is self contained in an Excel workbook.

The data to model is simply pasted into a spreadsheet in the application. Tiberius for Excel makes use of the charting capabilities of Excel so there are several graphs that are displayed that show the progress of the network as it is being trained.

This application is designed for people who want to develop their own neural network applications, experiment with algorithms or generally investigate neural networks. The underlying neural code will get you going with a learning algorithm that works.

Tiberius for Excel was created in Excel2000. Please download it and have a play.

Download

Click [here](#) to download (86KB zip file - tiny)

Purchase Source Code

As it stands, the code modules are password protected. If you want to access the code and develop it for yourself then we ask for a small contribution of US\$15.

Once you have access to the code you are free to do whatever you want with it.

Payments can be made though PayPal by clicking the link below. Include the code that appears at the bottom of the 'About' screen in TiberiusXL.

Code

Buy Now

Once we have email confirmation from PayPal that a payment has been made we will email you the password that will give you access to the code modules. You will be asked to give your email address on the PayPal screen. Please be patient



Tiberius original code by Phil Brierley, PhD

...Then vs. Now (www.philbrierley.com) ...

Neural Network Software

Welcome to Phils pages...

Tiberius	Algorithms include neural nets, regression splines and logistic regression. Demo version is fully functioning with no time restrictions.
Tiberius for Excel	A neat little neural net developed completely in Excel. Ideal for students/researchers who want to develop their own applications but need a kick start with a bit of code in an easy development environment.
Source code	FREE multilayer perceptron source code in several languages including MATLAB, C#, C++, C, Fortran, Perl, Java, PHP and VBA. This will get you started if you want to code your own neural networks. Also optimisation source code based on genetic algorithms.
Algorithm proof	A derivation of the popular neural network backpropagation learning algorithm.
Resources	Where to look on the web for neural network and data analysis information.
Who am I?	Maybe skip this page.

Site last updated on

"The more I use your software the more cool it becomes. I like how easy it is to use from the start, now it is almost automatic. I've tried just about every NN package out there and they all seem more interested in eye candy and making it seem scientific. So much so that getting the stuff to run is about impossible. Thanks."

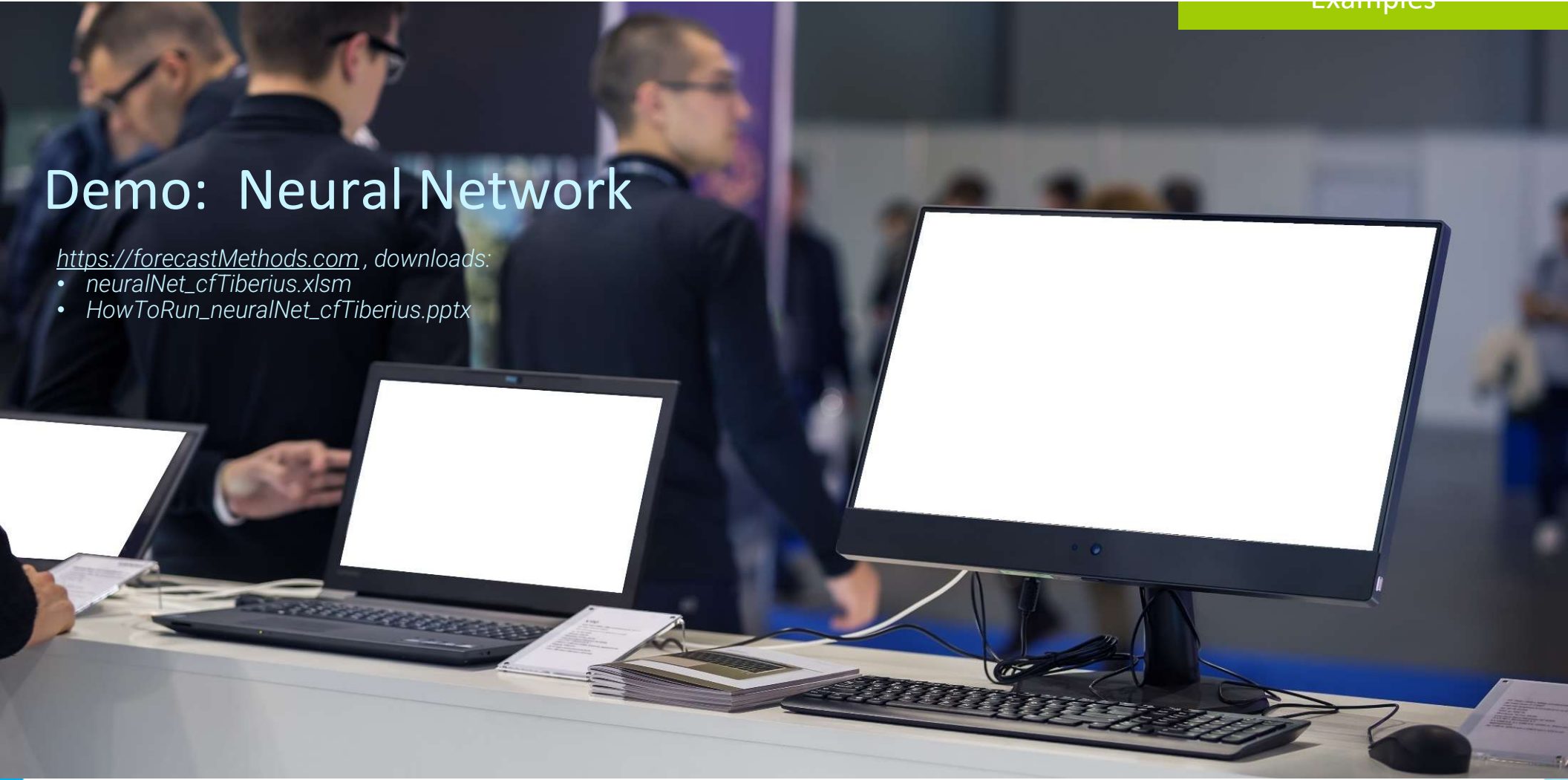
[Take me to Tiberius!](#)



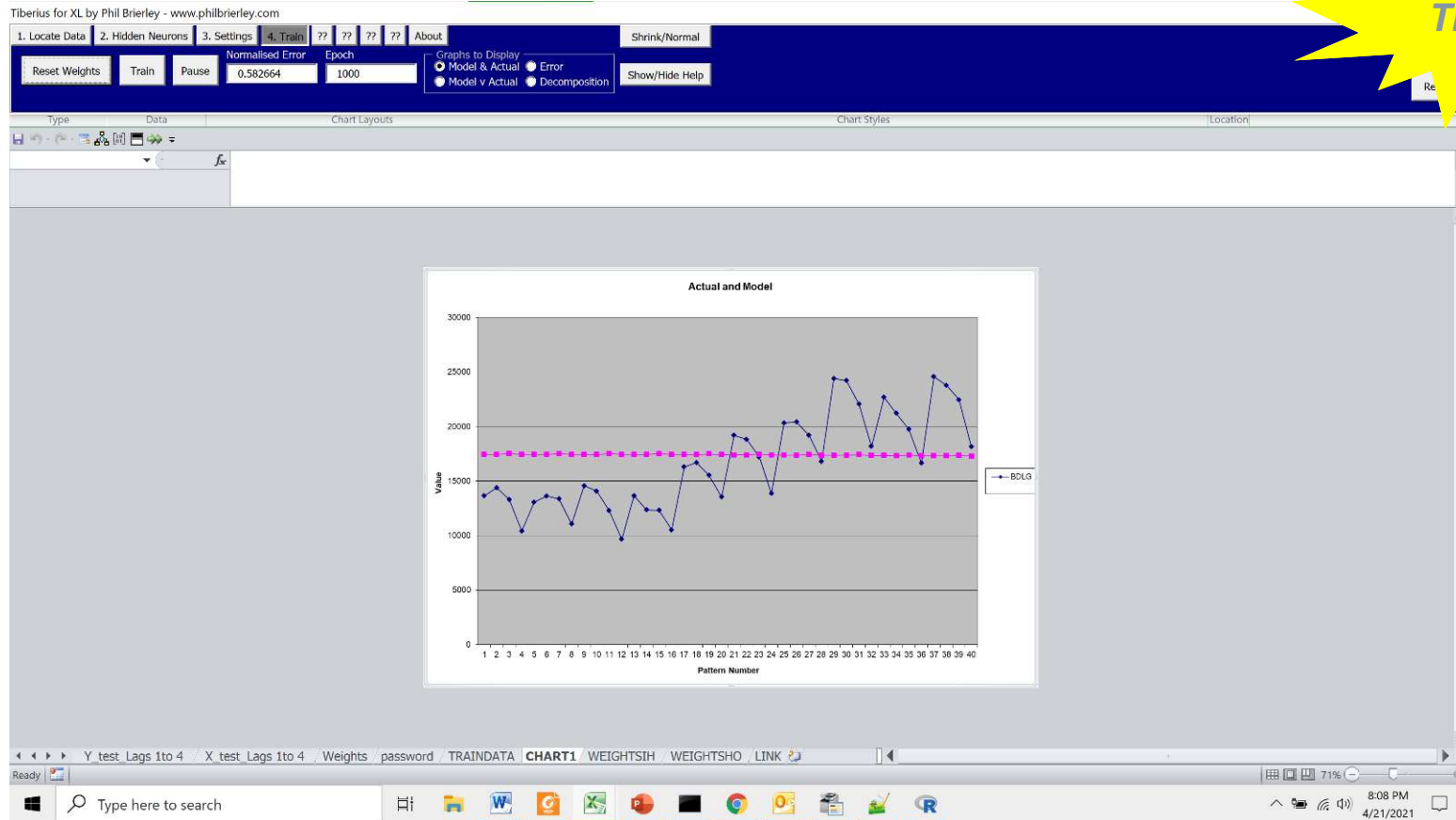
Demo: Neural Network

<https://forecastMethods.com/downloads/>

- [neuralNet_cfTiberius.xlsm](#)
- [HowToRun_neuralNet_cfTiberius.pptx](#)



<https://forecastmethods.com/>, click "neuralNet_cfTiberius.xlsm"



A Working Introduction to Python





Python is not a required tool for forecasting, but it is increasingly common across advanced analytics teams, especially where forecasting is evolving beyond traditional tools

Why Python is widely used for forecasting and analytics



- ✓ Strong libraries for forecasting and ML
- ✓ Flexibility to integrate data engineering, modeling, and deployment
- ✓ Large global community and continuous innovation
- ✓ Ability to handle both structured and unstructured data
- ✓ Compatibility with modern data stacks and cloud environments

How Python is like Excel

Concept	Excel	Python
Data table	Spreadsheet	pandas DataFrame
Calculations	Formulas	Python operations
Graphs	Charts	matplotlib and seaborn libraries
Scripting	Macros (VBA)	Python script (or program)
Prepared modules	Add-ins	Libraries

BUT Python is a way to give step-by-step instructions to a computer, using powerful pre-built tools, to do data work faster and more reliably than Excel

Libraries are ready-made tools

(so you don't have to create everything from scratch)

Example	Purpose
pandas	working with tables (cleaning, filtering, aggregating)
numpy	fast math (used under the hood)
matplotlib / seaborn	charts
statsmodels	statistical models
scikit-learn	machine learning
Prophet	time-series forecasting

NOTE: libraries are installed at a command-line interface
See <https://pip.pypa.io/en/latest/getting-started/>

Exercises in Python _basics folder

Program	Topics
0_YesYouCanTypeCode	Blank for your experimentation
1_Functions	Printing; addition; variable assignments; creating and applying simple functions
2_Comments_and_types	Documenting code; variable types (string, integer,etc.)
3_Operators_and_Logic	Arithmetic; comparison; if-then logic
4_Collections_and_Loops	Lists, tuples, and dictionaries; repeating code for an element of each
5_Libraries_and_DataInput	Using libraries; reading external data

A more thorough place to start...

in python, free Harvard course through edX platform

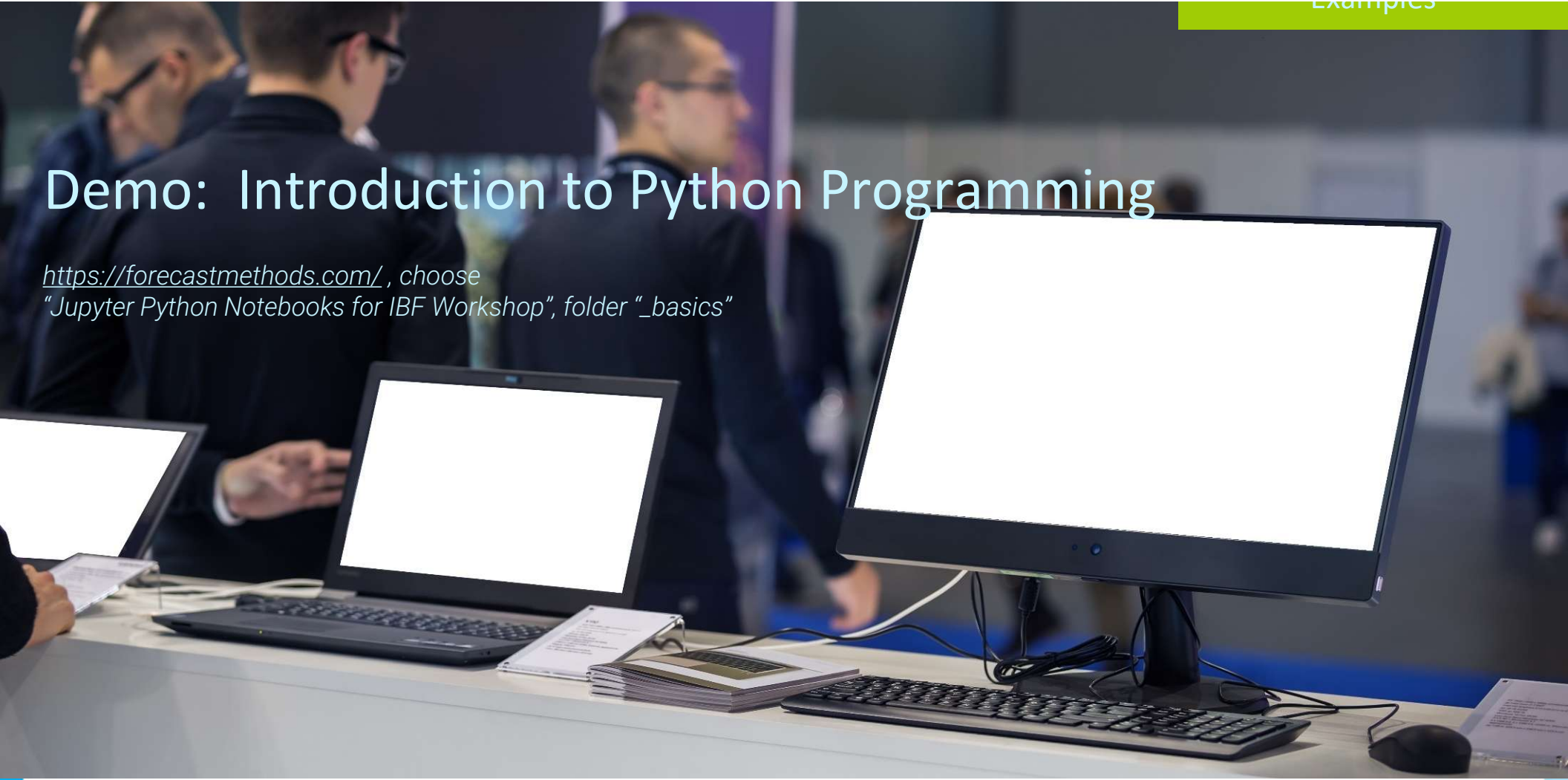
<https://www.edx.org/learn/python/harvard-university-cs50-s-introduction-to-programming-with-python>

The screenshot shows a web browser displaying the edX platform page for the HarvardX: CS50's Introduction to Programming with Python course. The browser's address bar shows the URL: <https://www.edx.org/learn/python/harvard-university-cs50-s-introduction-to-programming-with-python>. The edX logo and a search bar are visible at the top. A navigation breadcrumb trail reads: Home > Learn > Python > HarvardX: CS50'S Introduction To Programming With Python. The main content area features the Harvard University logo and the course title: **HarvardX: CS50's Introduction to Programming with Python**. Below the title, it states: "An introduction to programming using Python, a popular language for general-purpose programming, data science, web programming, and more." To the right, a box indicates the course starts on Jun 01 and ends on Dec 31, with a button to "Earn Harvard Certificate" and a note that 1,349,548 learners are enrolled. There is also a link to "Audit course". At the bottom, a row of icons and text describes the course as: **Introductory** (No prior experience required), **Self-paced** (Progress at your own speed), **10 weeks** (3-9 hours per week), and **Earn a certificate** (Advance your career today!). A chat bubble on the right says: "Hi! I'm Xpert, an AI assistant to help you find things." The edX logo is also present in the bottom right corner of the page.



Demo: Introduction to Python Programming

<https://forecastmethods.com/>, choose
"Jupyter Python Notebooks for IBF Workshop", folder "_basics"



Before you begin at home...

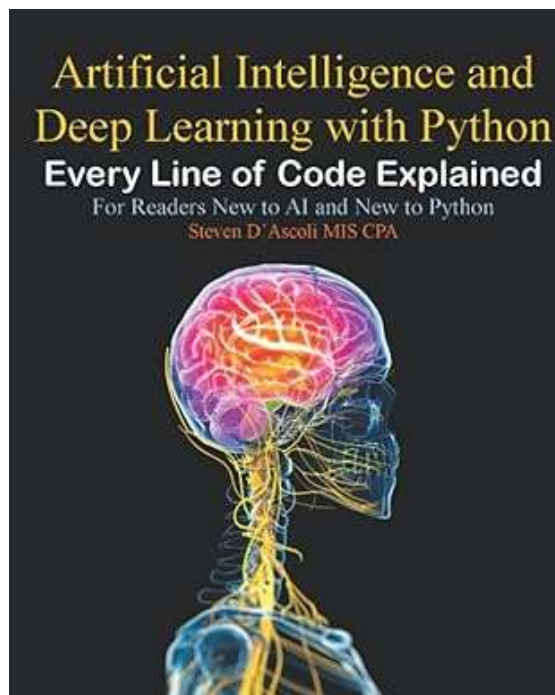
- Installing Python: <https://www.python.org/downloads/>
- Installing Jupyter: <https://jupyter.org/install>
- Installing libraries: See <https://pip.pypa.io/en/latest/getting-started/>
- If you need graphviz for network plots: <https://graphviz.org/download/>

Neural Networks in Python



Foundation of today's scripts

at <https://forecastmethods.com/>



D'Ascoli, Steven. Artificial Intelligence and Deep Learning with Python Every Line of Code Explained,
For Readers New to AI and Python

Key data setup for CSV files in these Python forecasting scripts

YearQtr	Product1	Product2	Product3	Unemp_Pc	CPI_Pct	RGDP	Time	Q4
2017Q3	384	1215	699	4.33	1.9259	18128	23	0
2017Q4	350	1169	1002	4.17	0.2200	18810.8	24	1
2018Q1	355	1222	788	4.03	3.3017	18437.1	25	0
2018Q2	353	1156	818	3.93	2.3092	18565.7	26	0
2018Q3	389	1331	681	3.77	1.7496	18699.7	27	0
2018Q4	365	1403	860	3.83	1.4691	18733.7	28	1
2019Q1	370	1446	856	3.87	0.9574	18835.4	29	0
2019Q2	400	1248	750	3.6	3.1804	18962.2	30	0
2019Q3	435	1289	851	3.63	1.4842	19130.9	31	0
2019Q4	401	1215	1031	3.6	2.4582	19215.7	32	1
2020Q1	428	1422	744	3.8	1.2983	18989.9	33	0
2020Q2	323	1371	848	12.97	-3.359	17378.7	34	0
2020Q3	287	1226	866	8.83	4.7939	18743.7	35	0
2020Q4	389	1279	1113	6.77	2.2408	18924.3	36	1
2021Q1	321	1261	679	6.2	4.1187	19216.2	37	0
2021Q2	295	1006	781	5.9	8.1871	19544.2	38	0
2021Q3	301	1073	909	5.1	6.7159	19872.6	39	0
2021Q4	372	927	994	4.23	7.0122	20006.2	40	1
2022Q1	354	965	724	6.8	9.2009	19924.1	41	0
2022Q2	302	900	886	3.6	10.5309	19895.3	42	0
2022Q3	392	1169	901	3.57	5.6856	20039.4	43	0
2022Q4				3.7	5.4	20239.6	44	1
2023Q1				3.8	4.5	20277.5	45	0
2023Q2				4	3.5	20322	46	0
2023Q3				4.3	3.1	20499.5	47	0
2023Q4				4.4	2.9	20920.8	48	1

Each row *must* be labeled with a variable that, when sorted, causes observations to be ordered by time

Targets (dependent variables) must be blank for future time periods

Features (independent variables) must be populated for future time periods



Demo: Neural Networks in Python

<https://forecastmethods.com/>, choose
"Jupyter Python Notebooks for IBF Workshop"
Step1_ReadData_and_EDA
Step2_Data_Prep
Step3_BuildNeuralNet

LLM's in the Forecasting Workflow



Note: Use of Online LLM's



- Proprietary data should not be exposed, such as by entry or uploading, into an online tool unless company management has authorized such exposure, having made provisions to assure confidentiality of data and their analyses
- Even if de-identified, non-proprietary data are to be entered online, company policies may prohibit use of online tools, even if the IT department has not blocked access to them

In other words, if you wish to enter data online into websites like <https://forecastMethods.com/> or <https://chatgpt.com/> , please assure that your company approves

Here's what ChatGPT advises (next slides):

ChatGPT Auto

- New chat
- Search chats
- Images
- Apps
- Deep research
- Codex
- Projects

GPTs

- ChatGPT Classic
- Explore GPTs

Recents

- Data Analysis Without LLMs
- Inventory Metrics in SCP
- Shiny Error Detail Options
- Scatter Plots in R
- Novice Guide to Forecasting
- Proxy Voting Authorization ...

Sara Brumbaugh
Ceres Analytics LLC

Invite team members

If I want to analyze data but my employer hasn't authorized entry into an LLM tool like ChatGPT, what should I do?

If your employer hasn't authorized the use of an LLM tool like ChatGPT for analyzing data, the safest and most professional approach is to **avoid using it with company data until you have approval**. Many organizations restrict these tools due to **data privacy, confidentiality, regulatory, and IP risks**. Here are practical options you can consider:

1 Follow your company's policy first

- Check your **AI / data security / acceptable use policies**.
- If the policy says external AI tools are not allowed for company data, **don't upload or paste internal data** into them.
- Violating this can create **security risks and compliance issues** for the company.

2 Ask for approval or a sanctioned tool

You could ask your manager, IT, or security team:

- Whether **LLM tools are allowed for data analysis**
- Whether they have an **approved AI platform** (many companies deploy enterprise versions such as internal LLMs or secure APIs)
- Whether you can use **ChatGPT Enterprise / Azure OpenAI / internal tools**

Often companies **do allow AI tools but only through approved environments**.

3 Use traditional approved tools

+ Ask anything

ChatGPT Auto

- New chat
- Search chats
- Images
- Apps
- Deep research
- Codex
- Projects

GPTs

- ChatGPT Classic
- Explore GPTs

Recents

- Data Analysis Without LLMs
- Inventory Metrics in SCP
- Shiny Error Detail Options
- Scatter Plots in R
- Novice Guide to Forecasting
- Proxy Voting Authorization ...

Sara Brumbaugh
Ceres Analytics LLC

Invite team members

3 Use traditional approved tools

If AI tools aren't approved yet, rely on tools typically allowed:

- Python (pandas, numpy, matplotlib)
- R
- Excel / Google Sheets
- SQL
- BI tools (Tableau, Power BI, Looker)

These can handle most analysis workflows safely inside company systems.

4 Use LLMs only for general guidance (no company data)

If policies allow it, you can still use LLMs to:

- Learn analysis techniques
- Ask about coding patterns
- Get help with generic examples

Example (safe):

"How do I group by a column and calculate a moving average in pandas?"

But don't paste real company data, queries, or proprietary schemas.

5 Propose a safe AI workflow

If AI would meaningfully help your work, suggest a compliant solution:

- Enterprise LLM access
- A private company-hosted model
- Data anonymization pipelines
- Using synthetic or sampled data

+ Ask anything

ChatGPT can make mistakes. OpenAI doesn't use Ceres Analytics LLC workspace data to train its models.

ChatGPT Auto

- New chat
- Search chats
- Images
- Apps
- Deep research
- Codex
- Projects

GPTs

- ChatGPT Classic
- Explore GPTs

Recents

- Data Analysis Without LLMs
- Inventory Metrics in SCP
- Shiny Error Detail Options
- Scatter Plots in R
- Novice Guide to Forecasting
- Proxy Voting Authorization ...

Sara Brumbaugh
Ceres Analytics LLC

Invite team members

5 Propose a safe AI workflow

If AI would meaningfully help your work, suggest a compliant solution:

- Enterprise LLM access
- A private company-hosted model
- Data anonymization pipelines
- Using synthetic or sampled data

Framing it as a productivity improvement with proper safeguards can help approval.

✓ Simple rule:

- No approval → don't use external LLMs with company data.
- Use approved tools or request authorization.

If you'd like, I can also explain:

- How companies safely deploy LLMs for internal data analysis
- What security teams worry about with ChatGPT
- Ways analysts use LLMs without exposing sensitive data.



+ Ask anything

ChatGPT can make mistakes. OpenAI doesn't use Ceres Analytics LLC workspace data to train its models.

Accelerating analysis without replacing judgment

What's on the agenda today?

I am uploading you a file with a number of columns. Product1, Product2, and Product3 are all targets that I'd like to explain. YearQtr is a time-period label. What you see when you compare Product1 with all the other variables that are not targets? And can you tell me how you "saw" these things?



Add photos & files Ctrl + U

Recent files >

Company knowledge

Create image

Deep research

Web search

... More >



Company knowledge

Accelerating analysis without replacing judgment

What's on the agenda today?



quarterlySales.csv
Spreadsheet



I am uploading you a file with a number of columns. Product1, Product2, and Product3 are all targets that I'd like to explain. YearQtr is a time-period label. What you see when you compare each Product with all the other variables that are not targets? And can you tell me how you "saw" these things?



Accelerating analysis without replacing judgment

What's on the agenda today?



quarterlySales.csv
Spreadsheet



I am uploading you a file with a number of columns. Product1, Product2, and Product3 are all targets that I'd like to explain. YearQtr is a time-period label. What you see when you compare each Product with all the other variables that are not targets? And can you tell me how you "saw" these things?



ChatGPT ▾

	YearQtr	Product1	Product2	Product3
1	2012Q1	250	1215	671
2	2012Q2	264	1760	807
3	2012Q3	282	1233	639
4	2012Q4	240	1231	900
5	2013Q1	296	1200	812
6	2013Q2	298	1225	733

I am uploading you a file with a number of columns. Product1, Product2, and Product3 are all targets that I'd like to explain. YearQtr is a time-period label. What you see when you compare each Product with all the other variables that are not targets? And can you tell me how you "saw" these things?

I loaded your data and looked at how each **target (Product1, Product2, Product3)** relates to the *non-target variables* (Unemp_Pct, CPI_Pct, RGDP, Time, Q4). Here's what stands out—and importantly, how I "saw" it.

What I see

1. Product1

- Strong positive relationship with RGDP (~ +0.61)

+ Ask anything



ChatGPT ▾

1. Product1

- **Strong positive relationship with RGDP** (~ +0.61)
→ As the economy grows, Product1 tends to increase.
- **Moderate positive trend over time** (~ +0.55)
→ There's a general upward trend (growth or adoption).
- **Strong negative relationship with unemployment** (~ -0.64)
→ When unemployment is high, Product1 drops.

👉 **Interpretation:**

Product1 behaves like a **pro-cyclical product**—it does well in strong economic conditions.

2. Product2

- **Strong negative relationship with CPI** (~ -0.75)
→ Higher inflation is associated with *lower* Product2 values.
- **Moderate negative relationship with RGDP** (~ -0.46) and **Time** (~ -0.42)
→ It's declining over time and during economic expansion.

👉 **Interpretation:**

Product2 looks like something that:

- struggles in inflationary environments
 - may be **declining structurally** or being replaced
-

3. Product3

- **Very strong relationship with Q4** (~ +0.76)
→ This is the clearest pattern in the dataset.
- Mild positive relationships with RGDP and Time (~ +0.27–0.29)

👉 **Interpretation:**

Product3 is **highly seasonal**, especially driven by Q4 (↓ ly holiday-driven demand).

Data Cleansing

Are there any irregularities in these data:

YearQtr	Product1	Product2	Product3	Unemp_Pct	CPI_Pct	RGDP
Time Q4						
2012Q1	247	1158	823	8.27	2.2718	16180.0 1 0
2012Q2	289	1349	713	8.20	0.8471	16253.7 2 0
2012Q3	277	1385	668	8.03	1.8181	16282.2 3 0
2012Q4	240	1365	1051	7.80	2.6865	16300.0 4 1
2013Q1	293	1017	650	7.73	6.6175	16441.5 5 0
2013Q2	330	1548	725	7.53	-0.4372	16464.4 6 0
2013Q3	263	1450	789	7.23	2.1810	16594.7 7 0
2013Q4	299	1078	1014	6.93	1.4873	16712.8 8 1
2014Q1	273	1273	776	6.67	2.5145	16654.2 9 0
2014Q2	330	1432	734	6.20	2.1413	16868.1 10 0
2014Q3	340	1067	823	6.07	1.0273	17064.6 11 0
2014Q4	366	1630	1074	5.70	-0.9901	17141.2 12 1
2015Q1	322	1377	714	5.53	-2.5641	17280.6 13 0
2015Q2	339	1404	789	5.43	2.7558	17380.9 14 0
2015Q3	367	1445	669	5.10	1.5194	17437.1 15 0
2015Q4	379	1277	1061	5.03	-0.0303	17462.6 16 1
2016Q1	367	1225	815	4.90	-0.2481	17565.5 17 0
2016Q2	303	1047	835	4.93	3.2377	17618.6 18 0
2016Q3	200	1284	726	4.90	1.7087	17724.5 19 0
2016Q4	340	1211	1045	4.77	2.5634	17812.6 20 1
2017Q1	307	1395	760	4.57	2.8448	17889.1 21 0
2017Q2	320	1594	658	4.37	0.4623	17979.2 22 0

● 1. Major Structural Break: COVID Shock (2020)

The most obvious irregularity is 2020Q2, which creates a sharp discontinuity across multiple variables:

- Unemployment (Unemp_Pct) jumps:
 - 2020Q1: 3.80 → 2020Q2: 12.97
- RGDP collapses:
 - 18989.9 → 17378.7
- CPI_Pct turns strongly negative:
 - 1.2983 → -3.3590

This is not a data error—it's a **real economic shock**—but statistically it behaves like an **extreme outlier / regime change**.

👉 Implication:

- Any model assuming stability (linear regression, ARIMA without intervention terms, etc.) will struggle unless you explicitly handle this break.

● 2. Outliers in Product Series

Product1

- 2016Q3 = 200
 - Much lower than surrounding values (~300–370)
- 👉 Likely an outlier or special event (inventory issue, demand shock, etc.)

Product2

- Large swings throughout:
 - 2013Q2 = 1548
 - 2014Q4 = 1630
 - 2019Q3 = 1538
 - 2021Q2 = 825 (sudden drop)

👉 This series is:

- Highly volatile
- Possibly **heteroskedastic** (variance changes over time)

Product3

- Strong **Q4 spikes**:
 - 2012Q4 = 1051
 - 2014Q4 = 1074
 - 2017Q4 = 1015
 - 2021Q4 = 1129

👉 This is not an error—it's **seasonality**, and your Q4 dummy confirms it.

3. Suspicious CPI Behavior

- 2013Q1 = 6.6175 → unusually high vs nearby values
- 2015Q1 = -2.5641 → strong deflation
- 2021–2022 = very high inflation (8–10%)

👉 These are:

- Realistic economically
- But statistically **extreme relative to earlier periods**



4. RGDP Pattern Irregularity

- Smooth growth from 2012–2019
- Then:
 - Sharp drop in 2020Q2
 - Quick rebound afterward

👉 This creates:

- Nonlinear trend
- Violation of constant-growth assumptions

5. Possible Data Issue (Minor)

Product2 in 2021Q2 = 825

- Much lower than typical range (~1000–1600)

This *might* be:

- A real drop
- Or a data entry inconsistency

Worth double-checking.

6. Structural Observations (Not Errors, But Important)

✓ Seasonality

- Product3 clearly seasonal (Q4 spikes)
- Q4 dummy variable is appropriate



6. Structural Observations (Not Errors, But Important)

✓ Seasonality

- Product3 clearly seasonal (Q4 spikes)
- Q4 dummy variable is appropriate

✓ Trend

- RGDP trending upward pre-2020
- Unemployment trending downward pre-2020

✓ Regime Change

- Pre-2020 vs post-2020 behave like **different systems**

✓ Bottom Line

The key irregularities are:

1. Major structural break

- 2020Q2 (COVID shock)

2. Clear outliers

- Product1: 2016Q3
- CPI: 2013Q1, 2015Q1
- Product2: several extreme swings

3. Strong seasonality

- Product3 (especially Q4 spikes)

4. Variance instability



📁 > ChatGPT ▾

✅ Bottom Line

The key irregularities are:

1. Major structural break

- 2020Q2 (COVID shock)

2. Clear outliers

- Product1: 2016Q3
- CPI: 2013Q1, 2015Q1
- Product2: several extreme swings

3. Strong seasonality

- Product3 (especially Q4 spikes)

4. Variance instability

- Product2 (changing volatility)

📊 If you were modeling this data

You would almost certainly want to:

- Add **COVID dummy variables** (2020Q2–2020Q3 at least)
- Model **seasonality explicitly** (Q4 or full quarterly dummies)
- Consider:
 - **Robust regression** or
 - **log transforms** (especially Product2)
- Possibly split into:
 - **pre-2020 model**




Can you please suggest a substitute value for the Product1 outlier?

Yes—let's handle that carefully and defensibly.

The outlier in Product1

- 2016Q3 = 200
- Surrounding values:
 - 2016Q2 = 303
 - 2016Q4 = 340

 This is clearly a **local drop inconsistent with trend and neighbors**

Best practice for replacement

For time series like this, the most defensible approach is:

- ✓ **Linear interpolation between adjacent quarters**

This preserves:

- Trend
- Continuity
- Minimal distortion



Knowing what your OS needs for tools

ChatGPT ▾

is graphviz installation for the operating system included in the "networkx" python package or library distribution?

No—NetworkX does *not* include Graphviz.

NetworkX is a **pure Python library** for creating and analyzing graphs. It does not bundle or install any system-level tools like Graphviz.

However, NetworkX *can optionally use Graphviz* for layout and visualization via integrations like:

- `nx.nx_agraph` (uses `pygraphviz`)
- `nx.nx_pydot` (uses `pydot`)

Those integrations:

- **require Graphviz to be installed separately on your system**
- depend on either `pygraphviz` or `pydot`, which themselves rely on Graphviz

So the dependency chain looks like this:

```
NetworkX
├── (optional) pygraphviz / pydot
│   └── Graphviz (system install required)
```

If Graphviz isn't installed, those features will fail (typically with missing `dot` executable errors).

Bottom line:

Installing `networkx` via pip or conda will *not* install Graphviz—you still need to install it at the OS level if you want Graphviz-based layouts or rendering.



Refining and explaining code

VII. Textbook code: future "forecast"

```
•[18]: #create fictitious variables for each of the independent
#variables in order to make a prediction about
# total_crime_reported_per_1_million_res
X_new = [[30,74,11,31,20]]
print(type(X_new)) #saras note: WHEN A METHOD IS NOT DEFINED SPECIFICALLY FOR A CLASS, IT IS NOT AN ATTRIBUTE, SO YOU APPLY IT AS *method(object)*

# Use the prediction function to predict
# total_crime_reported_per_1_million_res
# from the variables entered in the X_new list
prediction = model.predict(X_new)

# print the prediction
print("The total crime reported per 1 million residents would be:", \
      prediction)
```

```
<class 'list'>
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[18], line 11
      5 print(type(X_new))
      6 #saras correction:
      7
      8 # Use the prediction function to predict
      9 # total_crime_reported_per_1_million_res
     10 # from the variables entered in the X_new list
----> 11 prediction = model.predict(X_new)
     12
     13 # print the prediction
     14 print("The total crime reported per 1 million residents would be:", \
     15       prediction)
```

```
File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\keras\src\utils\traceback_utils.py:122, in filter_traceback.<locals>.error_handler(*arg
```

*Error message was that
prediction could not be
based on a list*

ChatGPT ▾

how do i convert a list of numbers to an object that can be used with the "predict" method of a tensorflow neural network model?

To use a list of numbers with a TensorFlow/Keras model's `predict()` method, you need to convert it into a **NumPy array (or tensor) with the correct shape**.

The key issue is not just conversion—it's **getting the shape right**.

1. Basic conversion (list → NumPy array)

```
Python
```

```
import numpy as np

my_list = [1, 2, 3, 4]
x = np.array(my_list)
```

```
Run
```

2. Add batch dimension (very important)

Keras models expect input shaped like:

```
(batch_size, number_of_features)
```

If your list represents **one sample**, you need to reshape it:

```
Python
```

```
x = x.reshape(1, -1)
```

```
Run
```

or directly:

```
+ Ask anything
```

```
Microphone icon
```



ChatGPT ▾

If your list represents **one sample**, you need to reshape it:

```
Python
```

```
x = np.array(my_list).reshape(1, -1)
```



```
Run
```

or directly:

```
Python
```

```
x = np.array(my_list).reshape(1, -1)
```



```
Run
```

3. Use with `predict()`

```
Python
```

```
pred = model.predict(x)
```



```
Run
```

4. Example

```
Python
```

```
import numpy as np

# Example input: 4 features
my_list = [5.1, 3.5, 1.4, 0.2]

x = np.array(my_list).reshape(1, -1)

prediction = model.predict(x)
print(prediction)
```



```
Run
```

VII. Textbook code: future "forecast"

```
[23]: #create fictitious variables for each of the independent
#variables in order to make a prediction about
# total_crime_reported_per_1_million_res
X_new = [[30,74,11,31,20]]
```

```
#sara's fix per ChatGPT:
#NOTE: to use numpy directly, you must import it. Although TensorFlow uses it behind the scenes,
# its importation is limited to inside TensorFlow, unless you explicitly import it for use in your program
import numpy as np
X_new = np.array(X_new).reshape(1, -1)
```

```
# Use the prediction function to predict
# total_crime_reported_per_1_million_res
# from the variables entered in the X_new list
prediction = model.predict(X_new)

# print the prediction
print("The total crime reported per 1 million residents would be:", \
      prediction)
```

```
1/1 ————— 0s 207ms/step
The total crime reported per 1 million residents would be: [[700.209]]
```

VIII. Textbook code: a plot to represent structure

```
[ ]: # The following code displays the structure of the model
from tensorflow.keras.utils import plot_model
plot_model(model, to_file='model_plot1.png', \
           show_shapes=True, show_layer_names=True)
```

Just in Case You're Wondering...

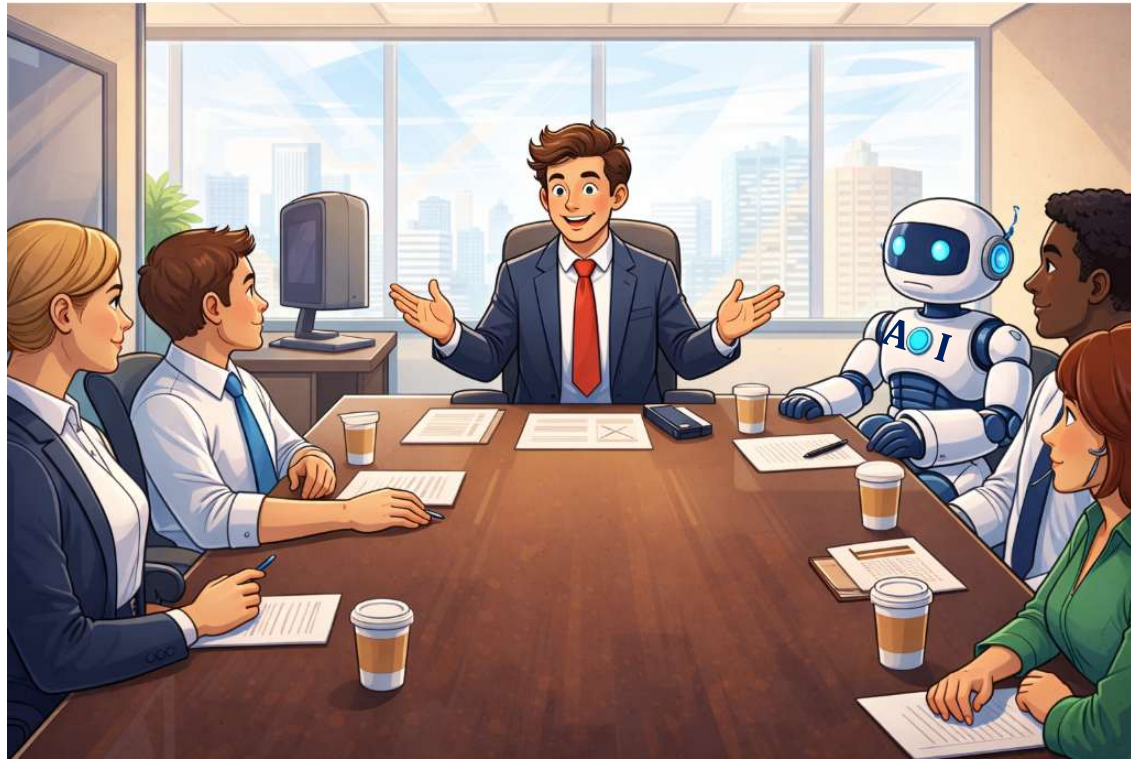
WHY NOT JUST REPLACE THE DEMAND PLANNER WITH AI?



Is AI a **problem** *that puts the planner's job at risk?*



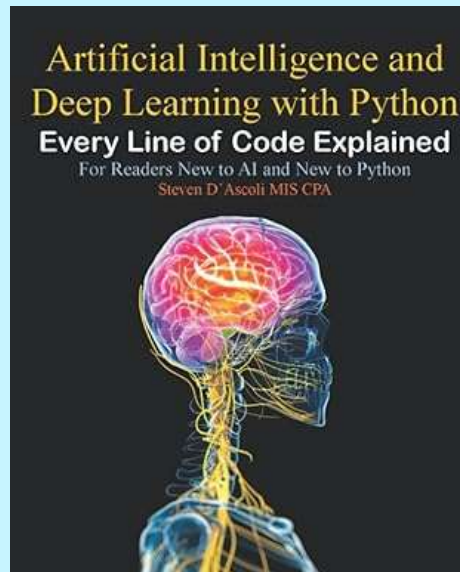
Or is AI a **goal**
of which the planner is the best master?



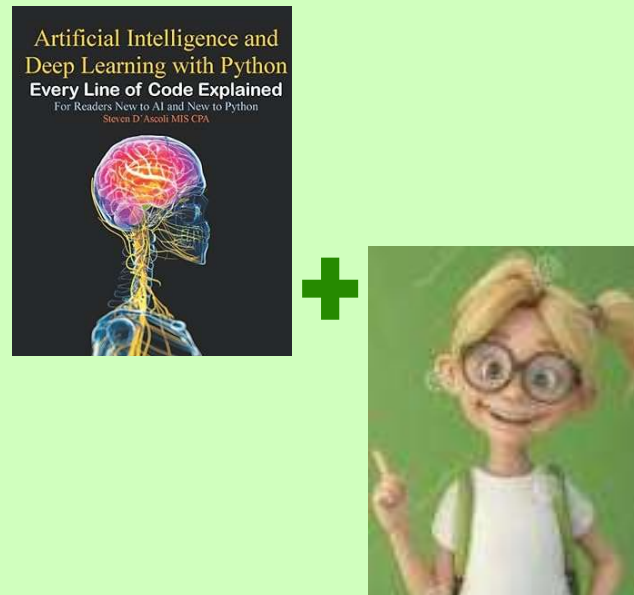
Two key Python neural net formulations so far

How about having AI LLM generate a third?

V1: Textbook script



V2: Textbook script
+ human forecasting
expertise



V3: Let AI LLM write the script



```
print(results.to_string(index=False))

print("\n=== Metrics on last 4 periods ===")
print(f"MAE : {mae:,.4f}")
print(f"RMSE: {rmse:,.4f}")
print(f"R^2 : {r2:,.4f}")

# Save output next to source file
out_path = os.path.join(os.path.dirname(CSV_PATH), "Product1_last4_forecast_dropna.csv")
results.to_csv(out_path, index=False)
print(f"\nSaved forecast table to: {out_path}")
```

Copy code

TEST 2

- Assure that the human's advantage is not random
- i.e., have ChatGPT initialize the random seed to equal Sara's



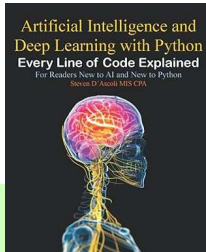
Can you provide me with a Python script that will provide a forecast, for the last 4 periods, based on a neural network, for 'Product1', based on 'Unemp_Pct', 'CPI_Pct', 'RGDP', 'Time', 'Q4' with data in my file C:/ibf/fromBkupDrive/shiny_IBF/www/quarterlySales.csv ? You may ignore all missing values, which Python would treat as "NaN". Set the random seed value ↓ 27.

+ Ask anything



Key difference in methodology

```
# -----  
# 3) Train/Test split: forecast last 4 periods  
# -----  
train = data.iloc[:-LAST_N].copy()  
test = data.iloc[-LAST_N:].copy()  
  
X_train, y_train = train[FEATURES], train[TARGET]  
X_test, y_test = test[FEATURES], test[TARGET]  
  
# -----  
# 4) Neural network model  
# -----  
model = Pipeline(steps=[  
    ("scaler", StandardScaler()),  
    ("mlp", MLPRegressor(  
        hidden_layer_sizes=(32, 16),  
        activation="relu",  
        solver="adam",  
        alpha=1e-4,  
        learning_rate="adaptive",  
        learning_rate_init=1e-3,  
        max_iter=5000,  
        early_stopping=True,  
        n_iter_no_change=25,  
        random_state=RANDOM_STATE  
    ))  
)  
  
model.fit(X_train, y_train)  
  
# Forecast last 4 rows  
y_pred = model.predict(X_test)
```



YearQtr	Unemp_Pct	CPI_Pct	RGDP	Time	Q4	Product1	Product1_Pred	Error
2021Q4	4.23	7.9122	20006.2	40	1	372.0	336.284515	35.715485
2022Q1	3.80	9.2009	19924.1	41	0	354.0	326.735229	27.264771
2022Q2	3.60	10.5309	19895.3	42	0	302.0	310.511566	8.511566
2022Q3	3.57	5.6856	20039.4	43	0	392.0	390.454651	-1.545349

=== Forecast for last 4 periods (final 4 rows after dropping NAs) ===

Unemp_Pct	CPI_Pct	RGDP	Time	Q4	Product1	Product1_Pred	Error
4.23	7.9122	20006.2	40	1	371.515071	397.847508	-26.332437
3.80	9.2009	19924.1	41	0	354.304691	367.152498	-12.847807
3.60	10.5309	19895.3	42	0	302.370640	372.080366	-69.709726
3.57	5.6856	20039.4	43	0	392.004569	418.028659	-26.024090

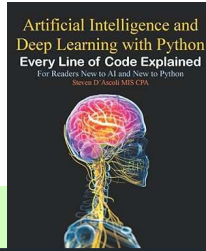
=== Metrics on test dataset, calculated like ChatGPT script ===

MAE : 18.2593
 RMSE: 22.8790
 R^2 : 0.5314

=== Metrics on last 4 periods ===

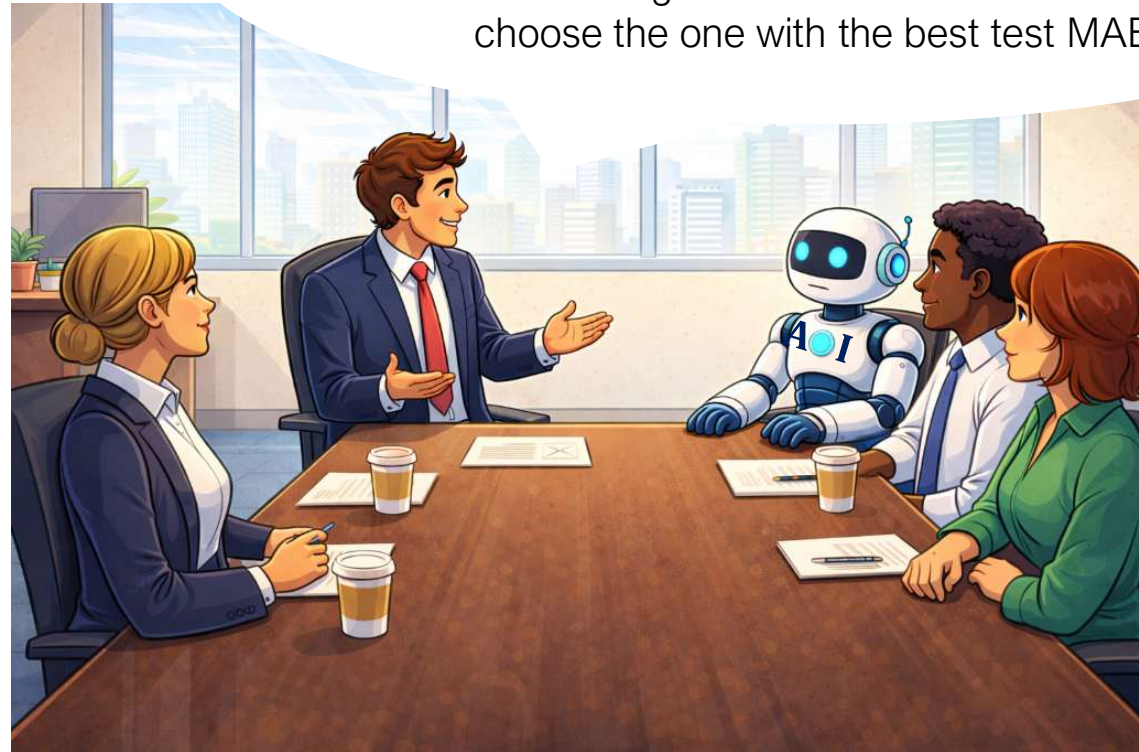
MAE : 33.7285
 RMSE: 39.9849
 R^2 : -0.4494





<p>Changed</p>		<ul style="list-style-type: none"> • Minor errors associated with libraries (e.g., usage of methods, need to import differently) • Random seed to be same as competitor (resulted in improvement)
<p>Left as-is</p>	<ul style="list-style-type: none"> • Neural network library: Tensorflow • Neural network tool: keras • Activation function: ReLU • Optimizer Adam • Learning rate adjusted to 0.01 after trials • Fewer nodes and layers • Early stopping if no change after 50 epochs • Dropout layers to prevent overfitting • 20% internal cross-validation (one split) in training • Final model updates training with test, then generates forecast 	<ul style="list-style-type: none"> • Neural network library: sklearn • Neural network tool: MLP Regressor • Activation function: ReLU • Optimizer Adam • Learning rate suggested as 0.001 • More nodes and layers • Early stopping if no change after 25 epochs • No dropout layers are possible in MLP Regressor • No internal cross-validation in training (though it is possible to do, with more than one split) • No final model

AI, your neural network script needs not only dropout layers, but also cross-validation within the training data. Please have the script sample different learning rates between .01 and .001, then choose the one with the best test MAE.



*Use your
knowledge, training
and experience to
“Be AI’s boss”*

Putting it all together



AI & Machine Learning in Forecasting

- AI can be viewed not as a replacement for the planner, but instead a tool the planner can master
- AI, through LLM's and ML, can be most effective when combined with human expertise and judgment
- A good place to start with ML is high-profit products that are difficult to forecast, or any product segment that doesn't perform well with "one size fits all" methods

Key takeaways and next steps *for real-world application*

Discussion