Neural Networks: Foundations to Generative AI

Multi-layer Perceptron (MLP) +
Working with Keras API

© Gordon Burtch, 2026

This Week's Agenda

1. What's in the News?

2. Building Blocks of NNs

   - Tensors (and relevant mathematical operations)
   - Activation Functions
   - Loss Functions
   - Backpropagation: Derivatives, Gradients & the Chain Rule (with examples)
   - Optimizers

3. Building a Linear Classifier

   - Overview of Keras and Tensorflow.
   - Implementing a linear classifier in Keras (now that we know the components).

© Gordon Burtch, 2026

What's In the News?

INNOVATION & AI > TECHNOLOGY > DEVELOPER TOOLS

Introducing Agentic Vision in Gemini 3 Flash

Jan 27, 2026
5 min read

Agentic Vision, a new capability in Gemini 3 Flash, combines visual reasoning with code execution to ground answers in visual evidence.

Rohan Doshi
Product Manager, Google DeepMind

Read AI-generated summary
Share

The Mashable 101 | Creator Hub | Tech | Science | Life | Social Good | Entertainment | Deals | Shopping

Home > Tech

OpenAI says its mystery AI wearable is on track for 2026 as AI earbuds rumors spread

Are ChatGPT-powered earbuds in the works at OpenAI?

By Timothy Beck Werth on January 20, 2026

CNET YOUR GUIDE TO A BETTER FUTURE

News | AI | Tech | Home | Entertainment | Wellness

ai atlas > Tech > Services & Software > AI

Apple's Next Big Wearable Could Be an AirTag-Sized AI Pin

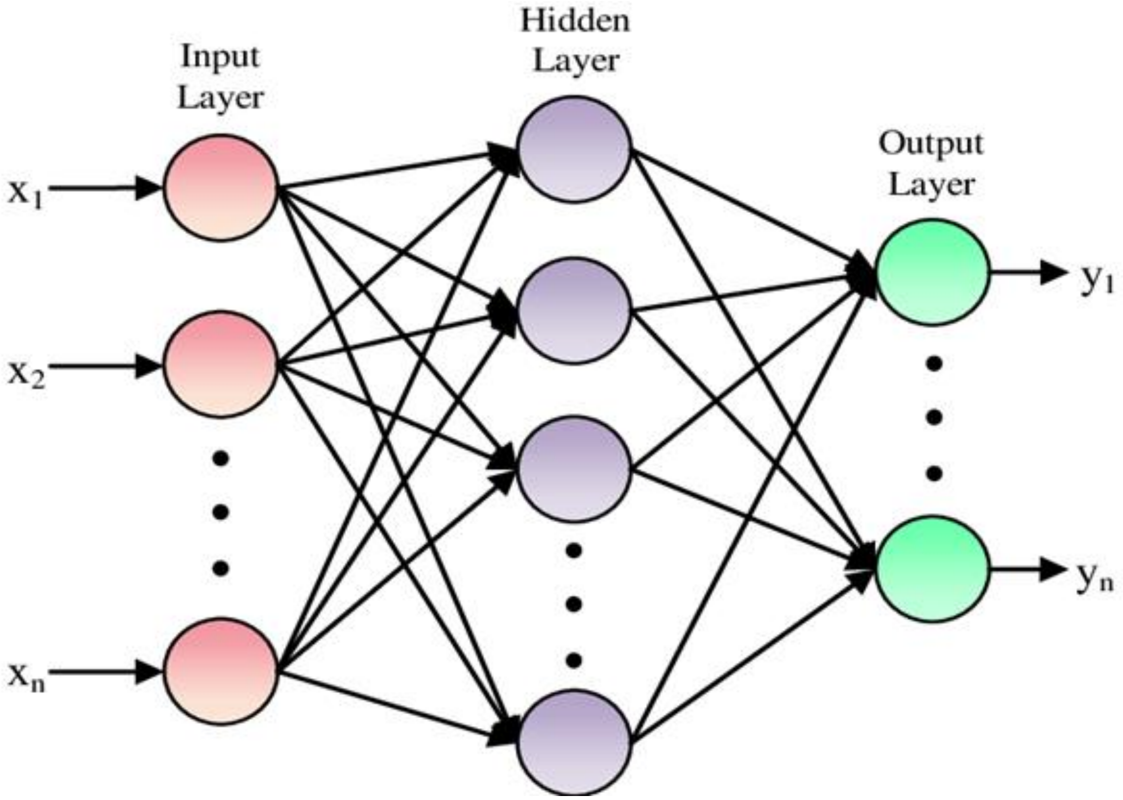
Apple is reportedly working on a small wearable device with an embedded microphone and cameras.

Omar Gallaga
Jan. 22, 2026 6:24 a.m. PT

2 min read

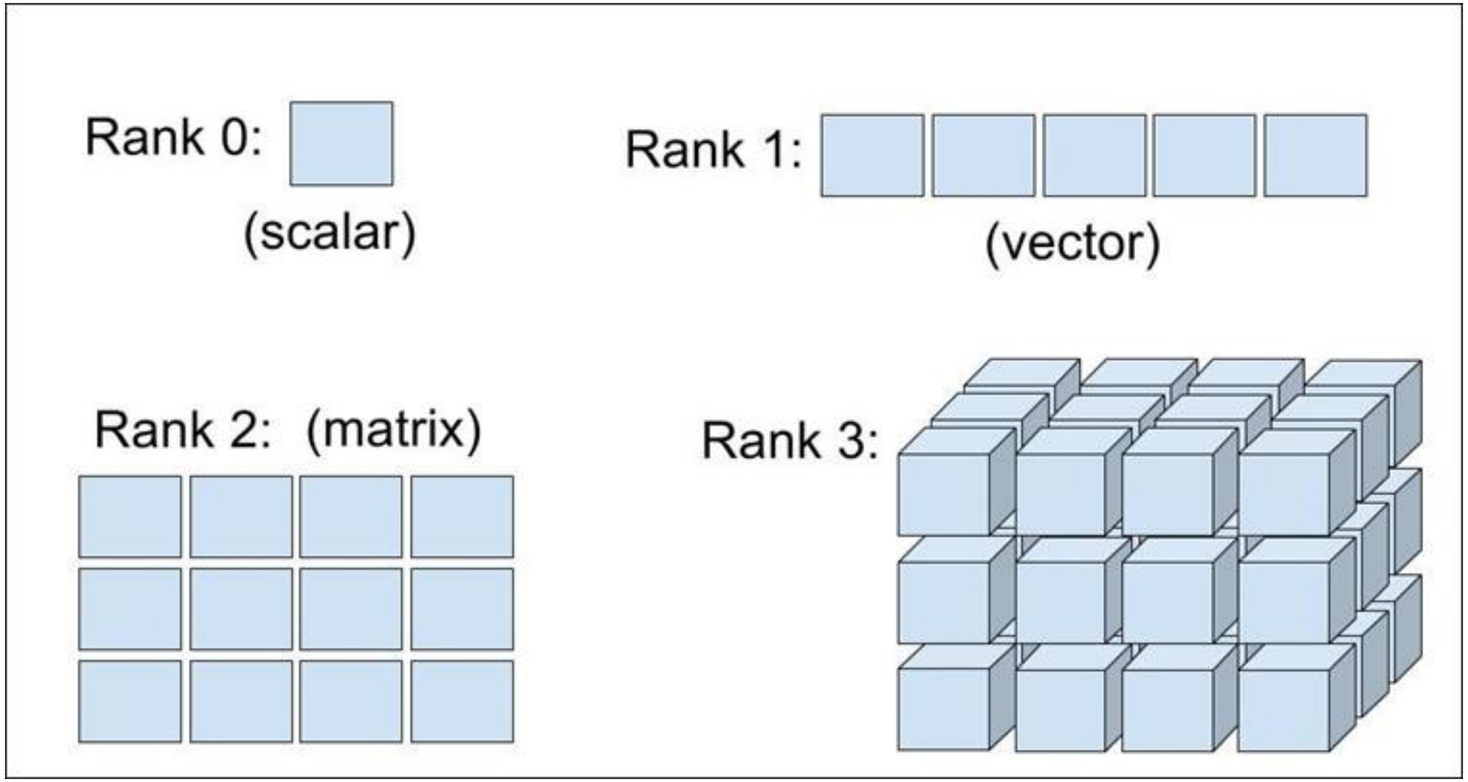
© Gordon Burtch, 2026

Recall: Multi-layer Perceptron (MLP)



© Gordon Burtch, 2026

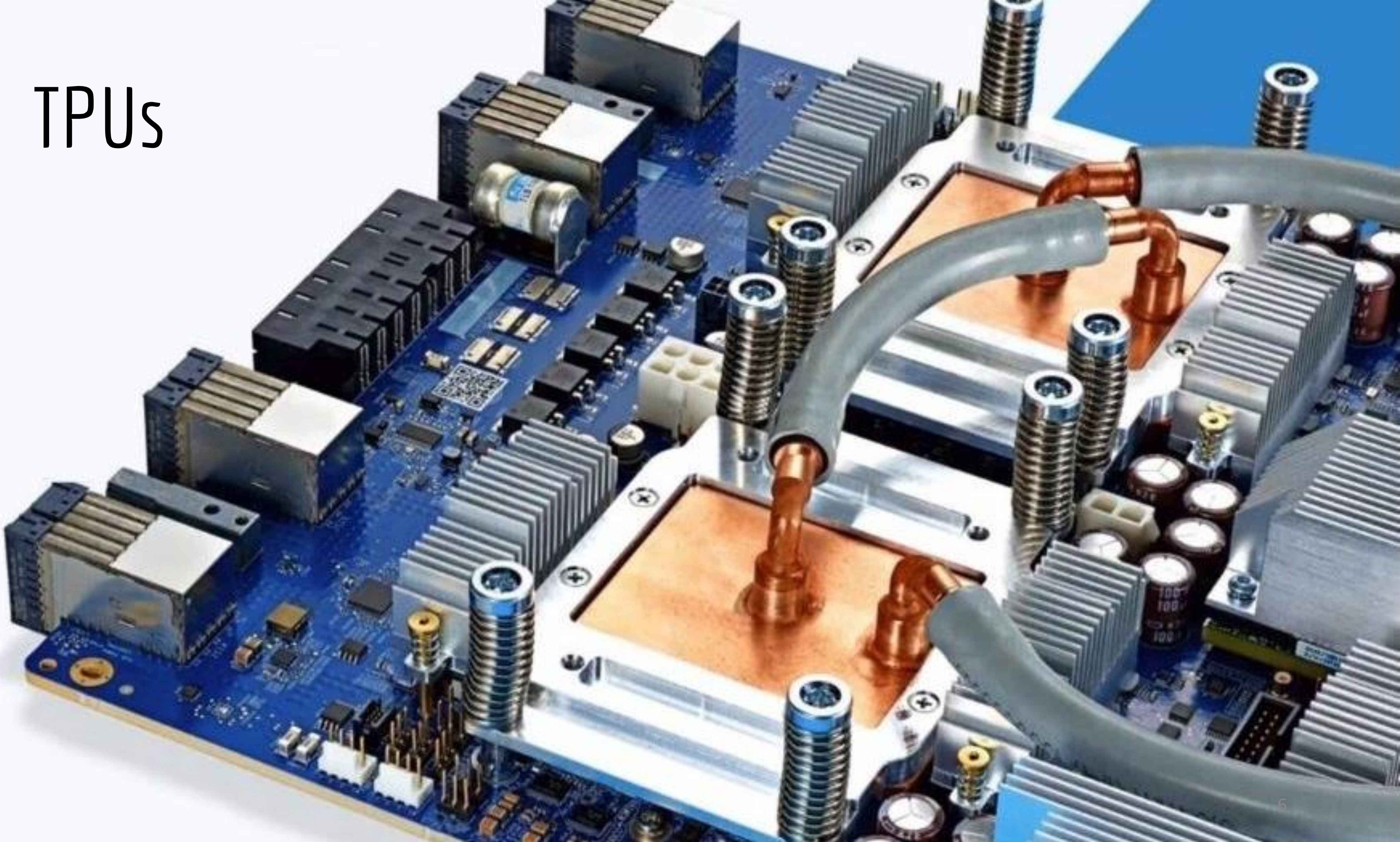
Tensors



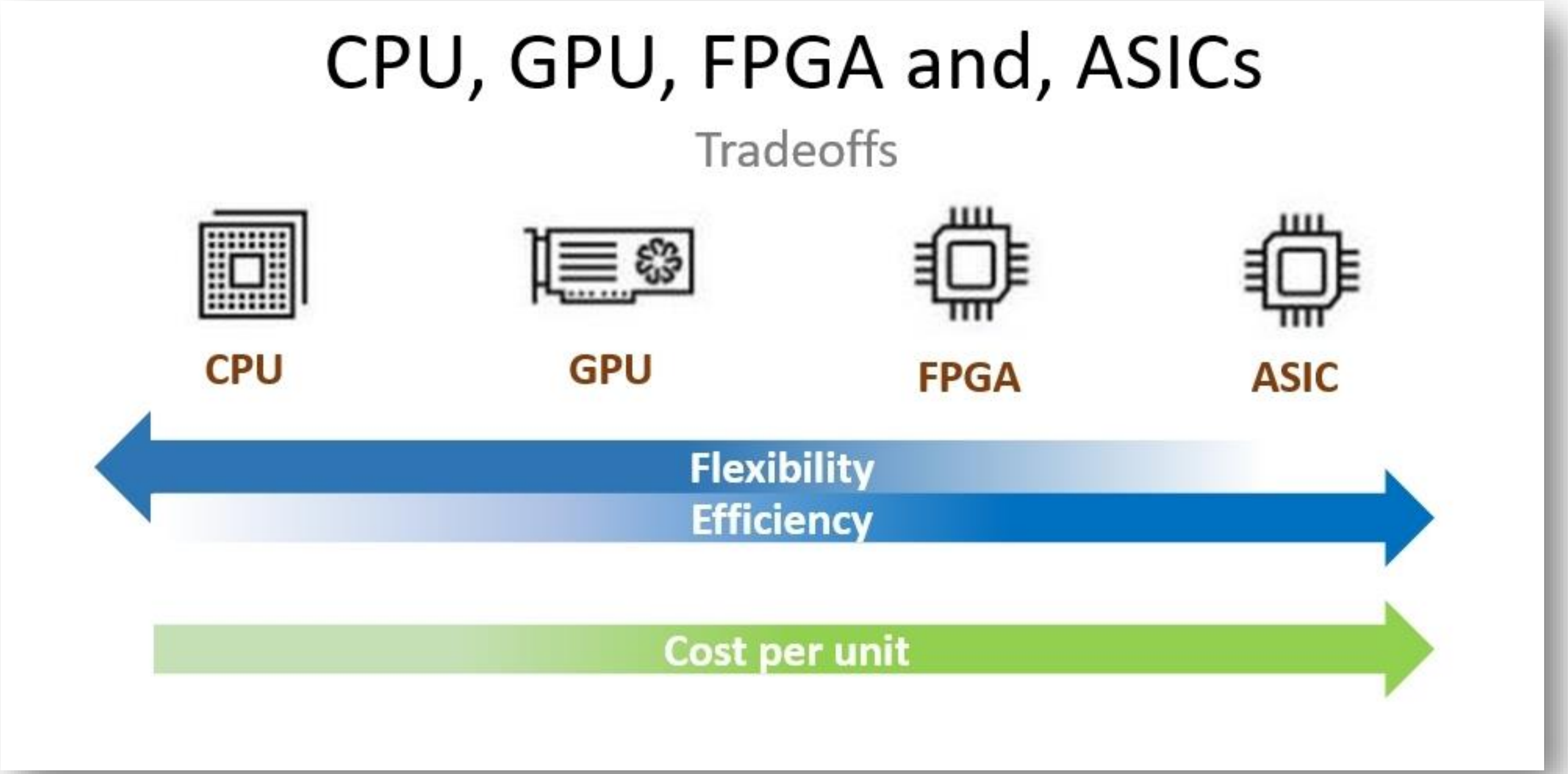
Question: What sort of data (give an example) would be stored in a rank-3 tensor? How about a rank-4 tensor?

© Gordon Burtch, 2026

TPUs

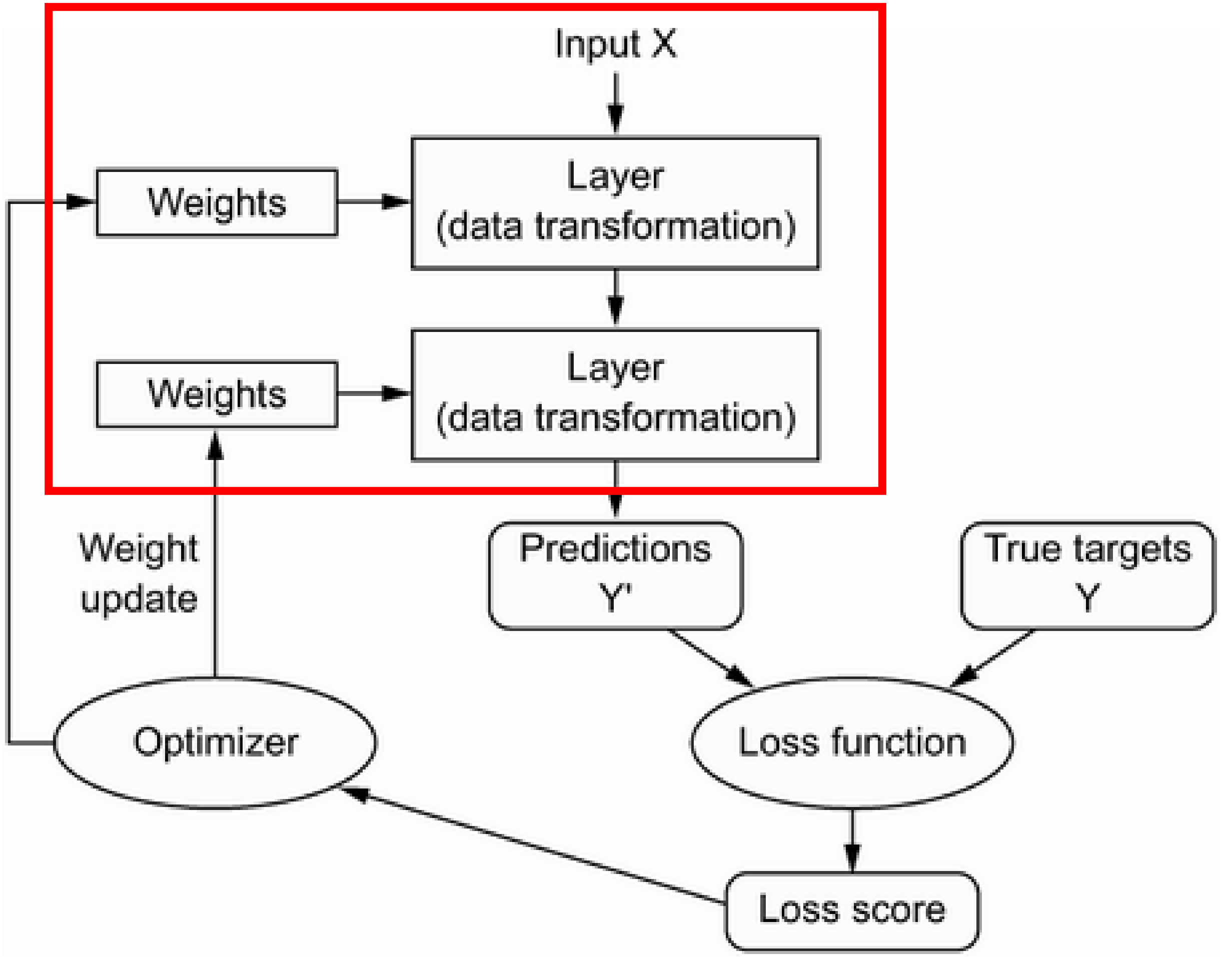


An Aside: GPU vs. ASIC



© Gordon Burtch, 2026

Forward Pass



© Gordon Burtch, 2026

The Perceptron

THE NEW YORK TIMES, TUESDAY, JULY 8, 1958. L 23

B. S. B.

Only
The BROOKLYN SAVINGS BANK
Gives You All These Banking Conveniences
UNDER ONE ROOF

HIGHEST BANK DIVIDENDS ON YOUR SAVINGS!
Latest Dividend 3¼% a year
Money deposited by JULY 15 earns dividends from JULY 1
QUARTERLY
After July 15, dividends from Day of Deposit
(3% regular plus ¼% extra)

SAVINGS BANKCHECK* SERVICE . . . THE CONVENIENCE OF A CHECKING ACCOUNT
A safe and convenient method of paying bills by using a combination of withdrawal orders and Savings Bank Money Orders or Teller Checks. 3 Money Orders Per Month Free! Only 15¢ each for additional Money Orders in amounts up to $250. No charge for Teller Checks issued for amounts over $250. FREE case holder for passbook and order forms—your name stamped on it in gold.
*Trade Mark

PERSONAL SAVINGS LOANS COST YOU LESS!
4½% simple interest on loan, less 3¼% dividend on savings

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could repro-

Books of The Times

By CHARLES POORE

"IF this were an entirely accurate account of my life in Cork," the author of "Mrs. O'"* tells us, "I should probably be writing it behind bars. So I should say that it is impressionistically true when not always factually so."

Fair enough. However, when you have finished her entertaining book, you may want to go back to that preface and wonder whether the bit about *behind bars* is a pun or an Irish bull.

Why? Because she ran a pub in Cork. The idea of doing so came to her in London one afternoon when she found herself rather rich and completely free. "My decree absolute came through on the same day as my Great Aunt's legacy—not a fortune, but such a sum as I had never dreamed of owning or saving." The fact that she happened to choose for refreshment a place called Mooney's, in London, gave the notion a proper touch of predestination.

Once in Ireland she made forays around the country. It did not take her very long to find the pub she wanted in Cork and buy it from a maiden lady who did not appreciate its seedy elegance. What names she signed to the deed we do not know, although this book is copyrighted by C. M. Forde. As author of it she calls herself, with royal simplicity, Claude, just Claude.

Named by Irish Friends

It was her Irish friends and customers who gave her the name of Mrs. O'. A reference to herself, near the end of the book, as one who holds in reserve "the resignation to the inevitable that lingers in the blood of those born in fatalistic East," marks the beginning of a cosmopolitan outlook.

A beau sabreur named Sean soon spotted her as French in spite of a quickly acquired talent for Gaelic. And Claude tells us she has "drunk rye with Americans, schnapps with Dutchmen, beer with Germans, wine with Frenchmen, liqueurs with duchesses and gin with charladies." The charladies and the duchesses, presumably, carry international passports.

The problems of running a pub in Cork were often hilarious, seldom businesslike and sometimes tragic. The gamut of life she saw was as various as the life you will encounter on Manhattan Island if you follow Park Avenue down from the street fairs near 125th Street to the local Mayfair within a mile of the Grand Central Terminal.

She liked the Irish and the Irish liked her. They brought her their problems and she

Claude, author of "Mrs. O'."

witz with any attention, you would think of the long arid stretches with no hostelries."

Early on, as proprietor of her own pub, she had learned to tap a mighty keg of high stout, after just one lesson from a friendly rival called Foxy:

"Wrapping the barrel end of the tap in three thicknesses of newspaper as I had seen Foxy do, I placed the tap against the bung, raised the mallet, and thinking briefly that I should probably be the first foreigner ever to be killed by Guinness, I hit the tap two fairly light, quick blows."

Mallet's Force Augmented

It worked fine. The third whack was delivered at full strength. The tap went into place, the newspaper sealed the crack around it. One thing she was too shy to mention when congratulations, offered in awe, saluted her, was that she had, shall we say, augmented the force of the mallet with a huge horseshoe she had discovered under the bar.

A rather formally informal romance flowers in the book. It concerns Sean, whose past is a subject for gossip, and whose present is a matter of mystery. He is destined for tragedy. When he meets it, as no reader can doubt he will, Ireland loses magic for Mrs. O'.

Then there is Phelan, the ambitious burglar. He is a genial sort. But he never owns up to his limitations. In defiance of the best advice he attempts robberies that are beyond the scope of his abilities. The result is that he is often in court, sometimes in jail. The problem of Phelan is solved more happily

Books—Authors

A month before Joyce Cary's death last year the complete manuscript of his new book "Art and Reality" arrived at Harper's, his publisher. It is based in part on lectures prepared for delivery at Clare College, Cambridge. Mr. Cary described the book as "an attempt to examine the relation of the artist with the world as it seems to him and to see what he does with it." He examines the nature of intuition and truth in art, the difficulties of translating intuition into production and the character of originality in art. The book will be published Aug. 6.

Lucius Cornelius Sulla (138-78 B. C.), Roman general and politician, established a dictatorship in 82 B. C. through which he eliminated members of the opposing party. To later generations his name became a symbol of cold, calculating cruelty. He is the subject of a historical novel, "The Sword of Pleasure," by Peter Green, which World will issue Aug. 27. The story will be the first by the young English historical novelist to be published in this country.

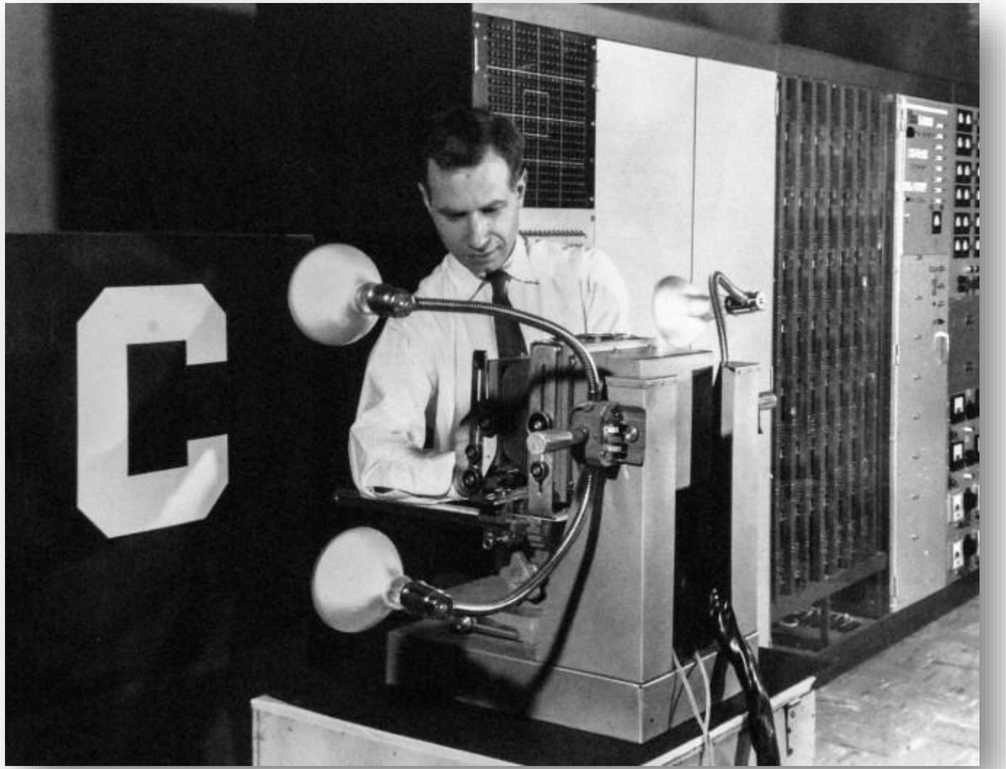
A young French writer will be introduced to the American reading public in September when Dutton issues "Jacob," by Jean Cabriès. Translated from the French by Gerard Hopkins, the novel is based on the story as told in Genesis and covers the period from Jacob's flight from Isaac's house to his reconciliation with his brother Esau. Described as a spiritual novel, it tells of a man's education.

"The Patchwork Hero," a novel by Michael Noonan, is planned for November publication by John Day. It is the story of a year in the life of a motherless boy. The youngster lives in a respectable village with his roistering tugboat-captain father and his father's friends, who share their festive and unpredictable hospitality.

Just Published!

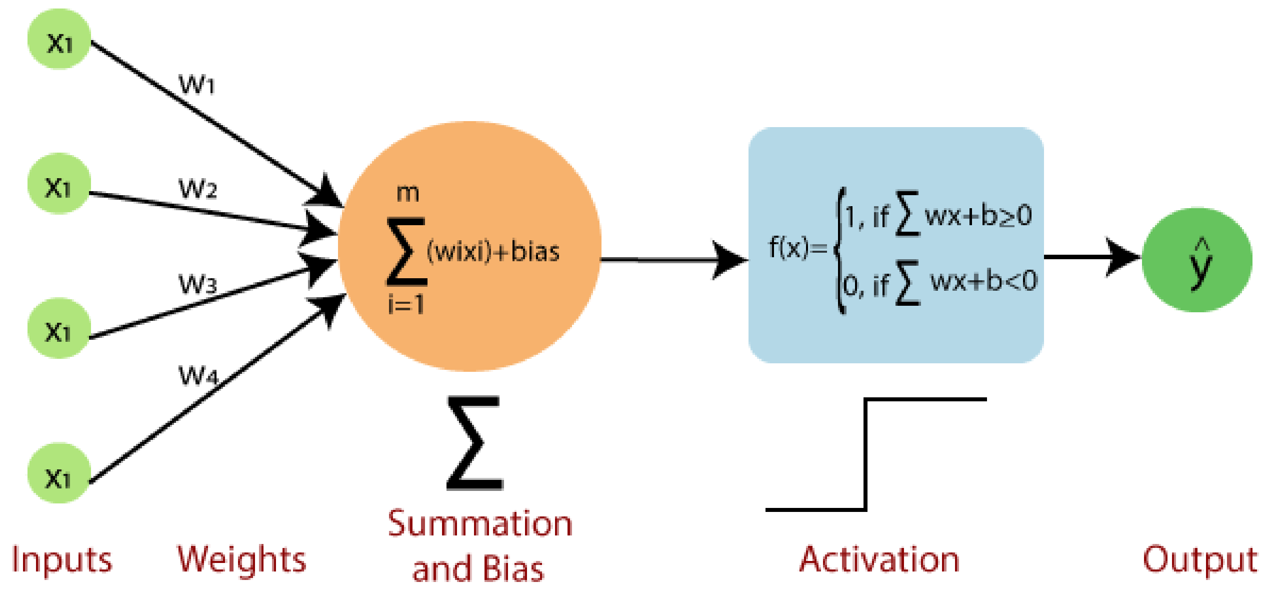
ALGERIA
The Realities
by Germaine Tillion

ALBERT CAMUS: "I have never been able to read any book on the Algerian tragedy without a feeling of unreality, discomfort, and often even anger. Only one book — Germaine Tillion's — seemed to me at once true, fair, and constructive. Germaine Tillion knows what she is talking about. And either in Algeria or throughout the world can ... the Algerian problem without ... understanding and ... misunderstood,



© Gordon Burtch, 2026

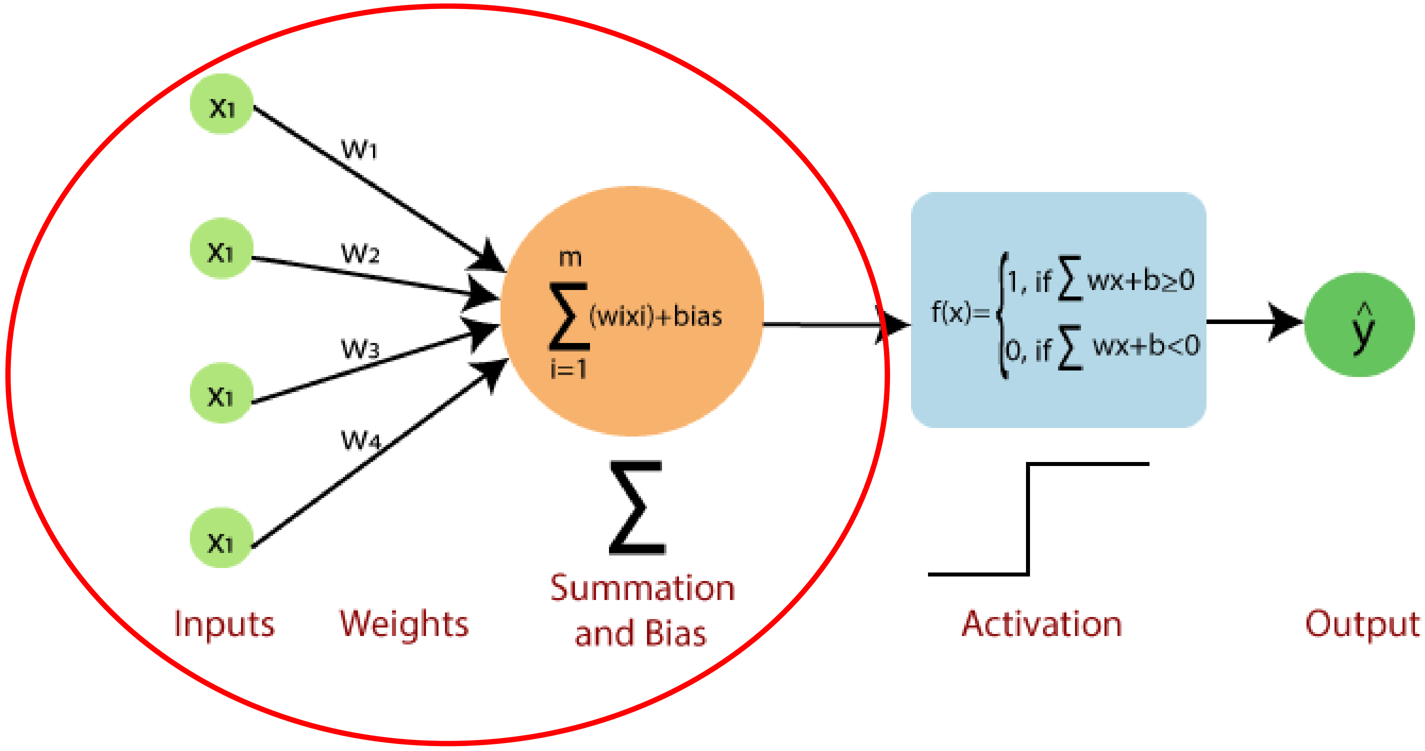
Neuron / Network Components



Question: What rank tensor are x, w and b here?
What will the shape of y be?
What is the order of operations in a forward pass?

© Gordon Burtch, 2026

Neuron / Network Components



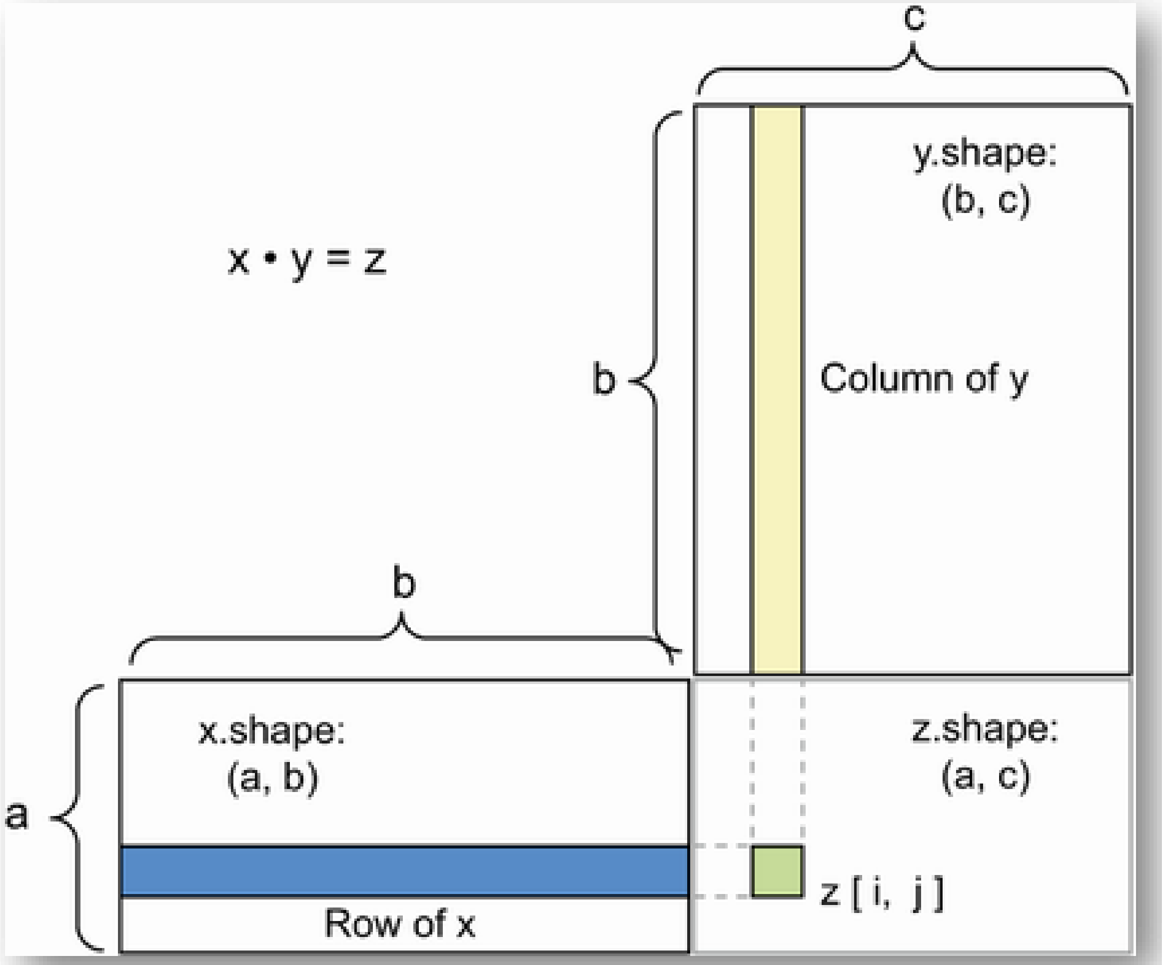
Question: Which of these values are constants?
Which are trainable parameters?

© Gordon Burtch, 2026

Multiplication

$y_1 = \varphi\,(x_1 \cdot w_1 + b_1)$

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1 x + a_2 y + a_3 z \\ b_1 x + b_2 y + b_3 z \\ c_1 x + c_2 y + c_3 z \end{bmatrix}$$



© Gordon Burtch, 2026

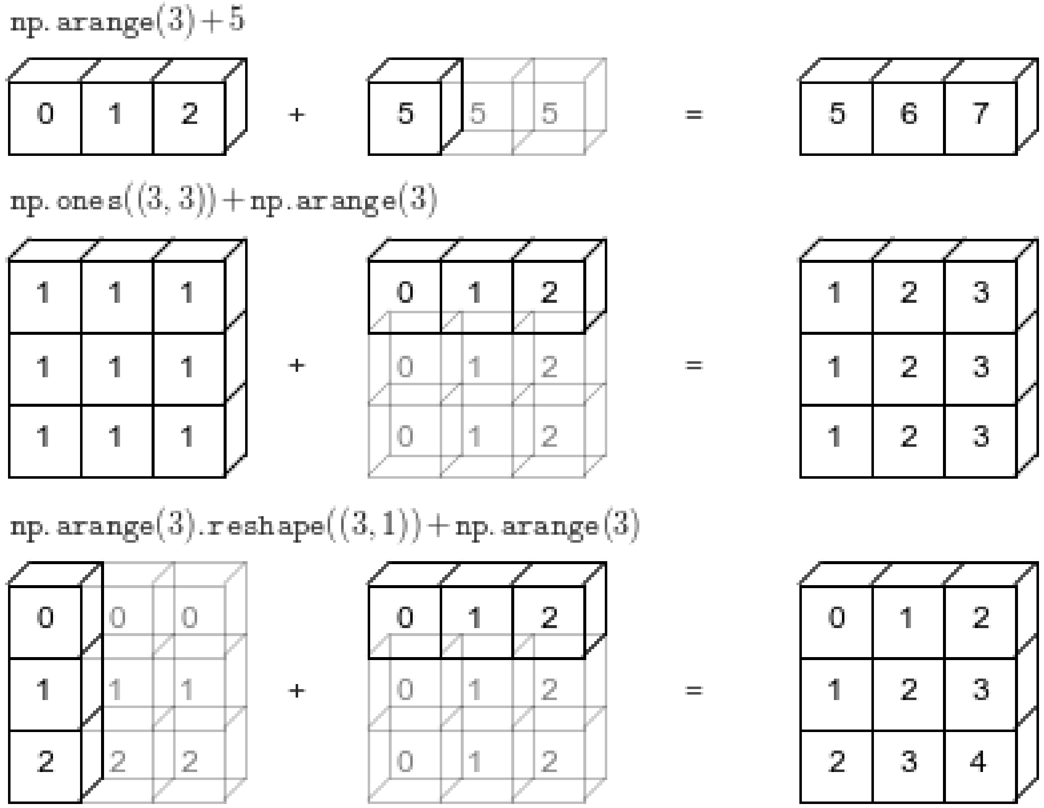
Matrix Addition (Broadcast)

$y_1 = \varphi\,(x_1 \cdot w_1 + b_1)$

Shape of the Two Tensors Needs to Conform
- A + B will only work if A is cleanly divisible by B (or vice versa)

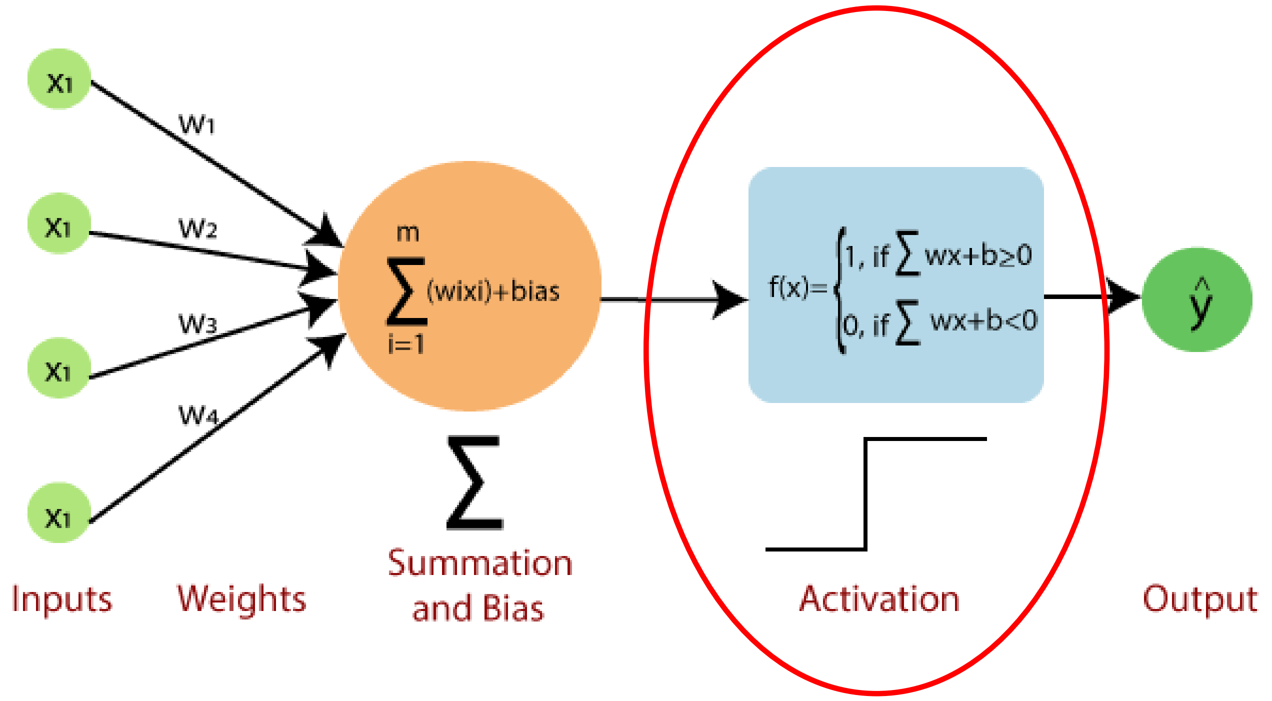
Sum Element-wise
- Replicate B until it matches A's dimensions, then perform element-wise addition.

We Use This for the Addition Step
- Add x*w and b (bias)



© Gordon Burtch, 2026

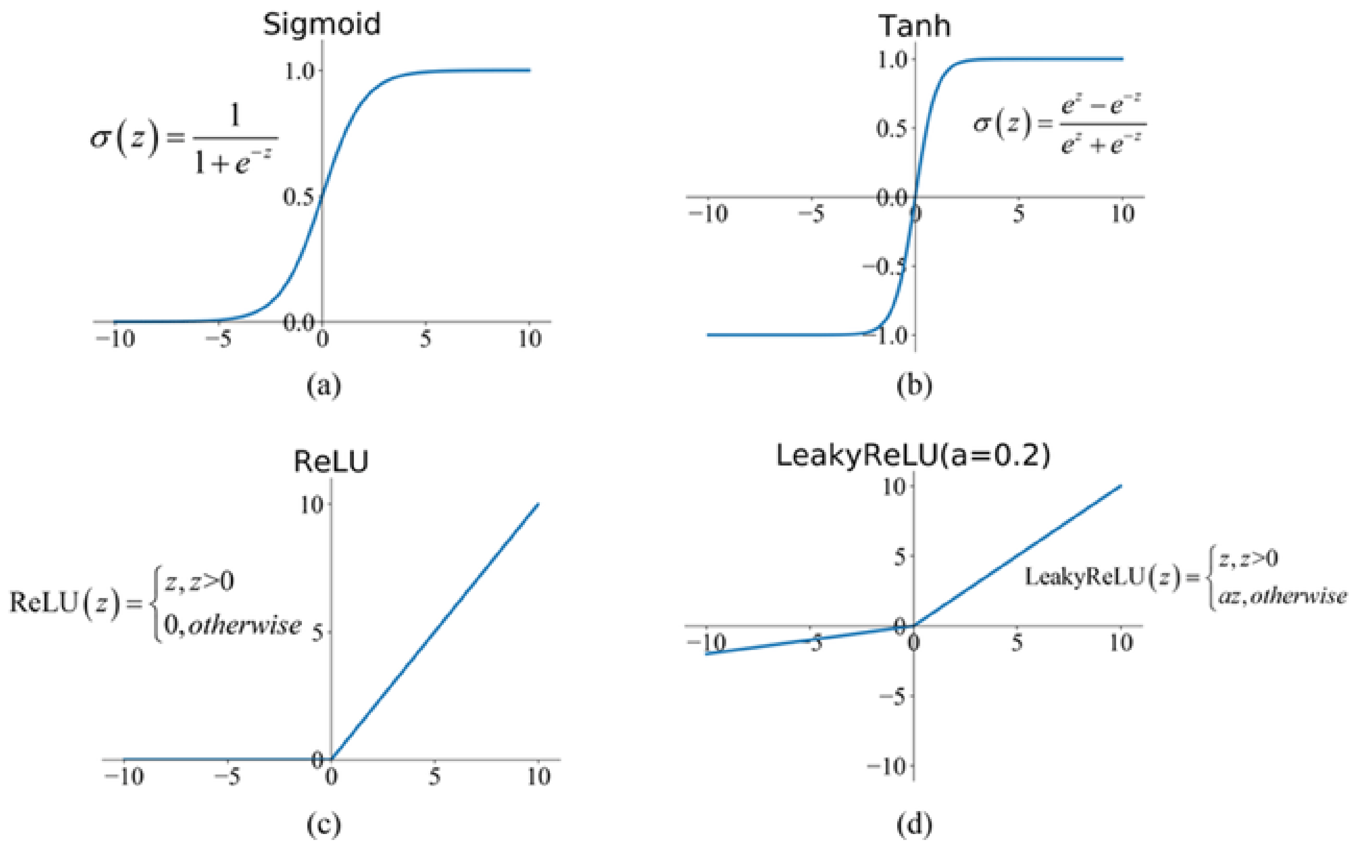
Neuron / Network Components



© Gordon Burtch, 2026

Activation Functions

$y_1 = \varphi\,(x_1 \cdot w_1 + b_1)$



© Gordon Burtch, 2026

Activation Functions

$$y_1 = \varphi\,(x_1 \cdot w_1 + b_1)$$

Hours & Admission Subscribe Account

CHM Computer History Museum

EXPLORE VISIT ABOUT JOIN & GIVE DONATE TICKETS

HOME › CHM BLOG › SOFTWARE HISTORY CENTER

CHM RELEASES ALEXNET SOURCE CODE

By Hansen Hsu | March 20, 2025

© Gordon Burtch, 2026

Multi-Class, Single-Label

$y_1 = \varphi\,(x_1 \cdot w_1 + b_1)$

<u>Softmax (MLOGIT):</u>

We have D inputs (x's).
We have k outputs (classes).

So, W is a (D,k) matrix and X is a (D,1) matrix.

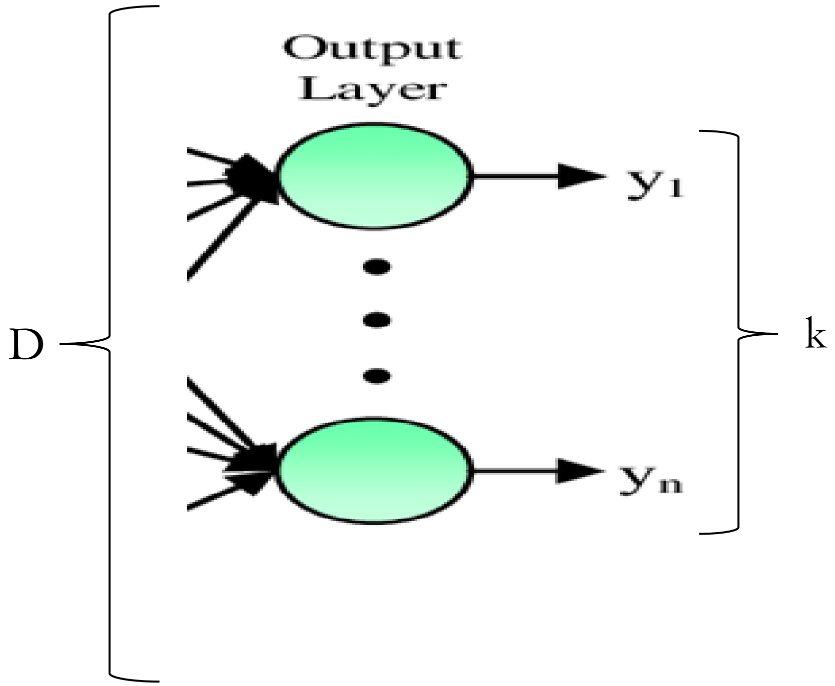
That means, A is a (k,1) matrix.

That means Y is also a (k,1) matrix.

$$A = W^T X,$$

$$Y = \text{softmax}(A),$$

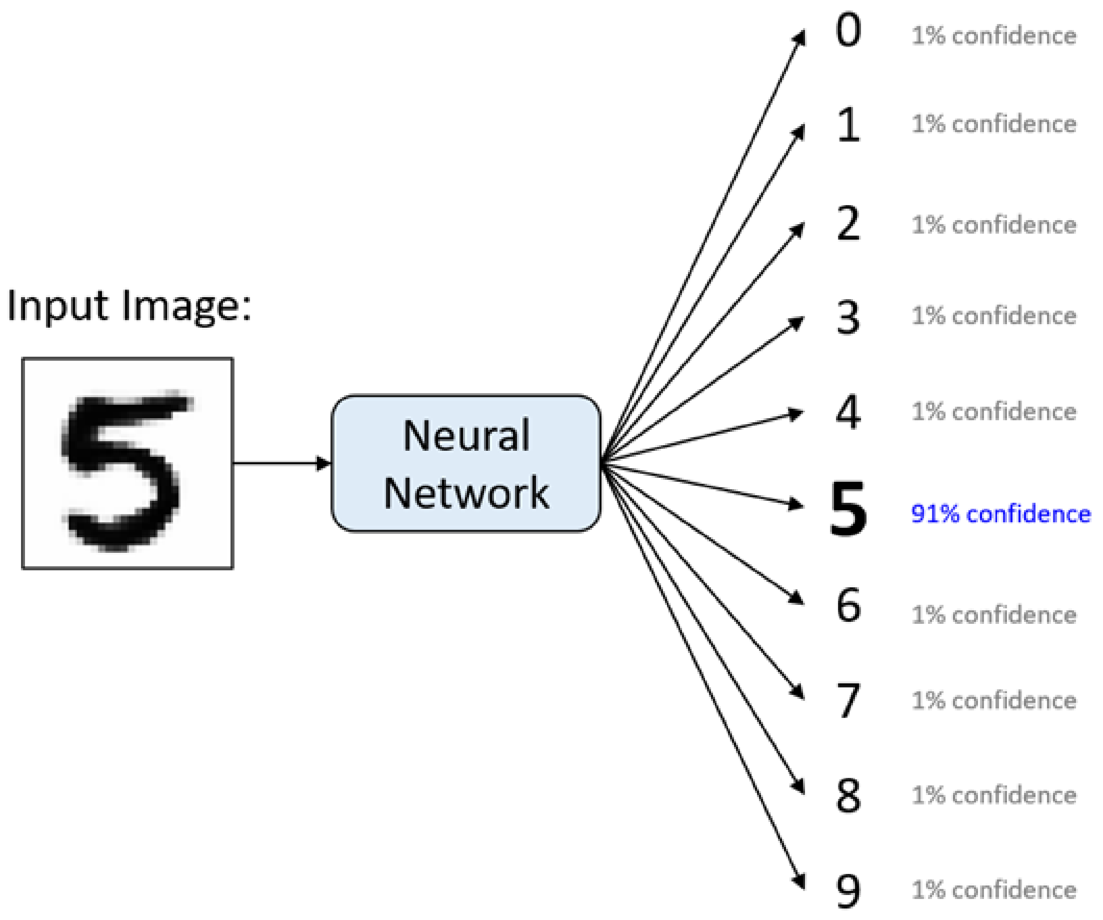
$$Y_i = \frac{e^{A_i}}{\sum_{j=1}^{k} e^{A_j}}.$$



© Gordon Burtch, 2026

Multi-Class, Single-Label

$y_1 = \varphi\,(x_1 \cdot w_1 + b_1)$

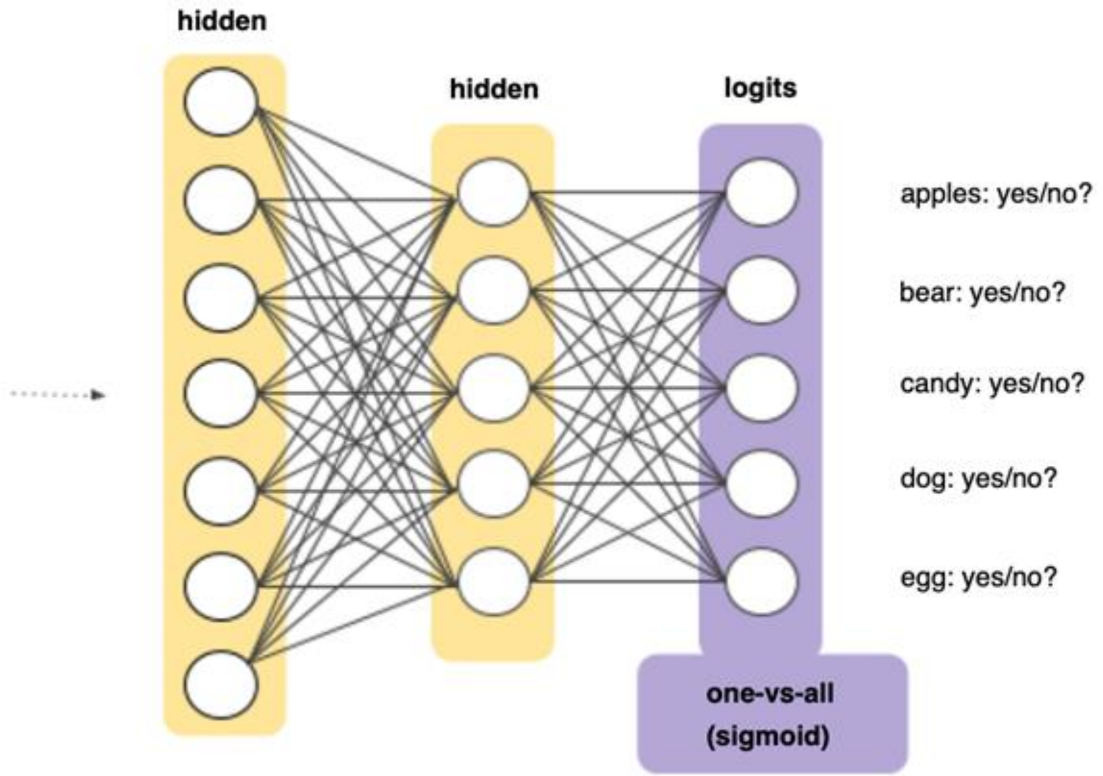


© Gordon Burtch, 2026

Multi-Class, Multi-Label

Many Non-Exclusive Labels

- We would create a sigmoid output layer with one output for each class we are predicting.
- Train on all labels together.



© Gordon Burtch, 2026

We Know Enough for a Forward Pass

Calculate Output of Each
Node Sequentially

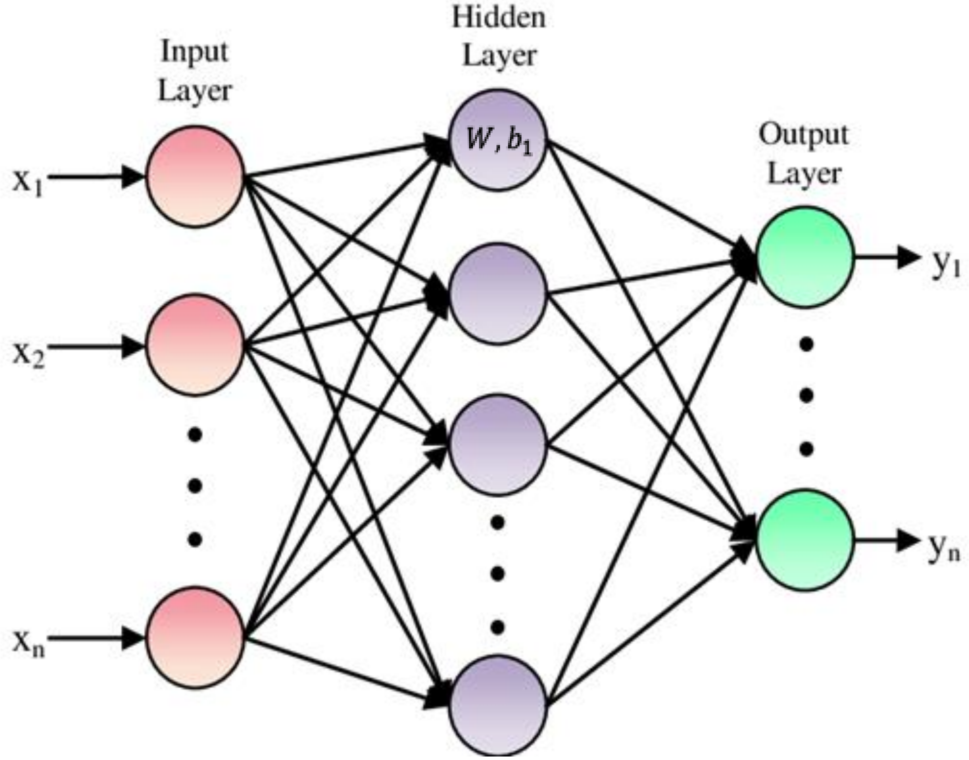
$$y_1 = \varphi\left(x_1 \cdot w_{1,1} + x_2 \cdot w_{1,2} + \cdots + b_1\right)$$

$$y_2 = \varphi\left(x_1 \cdot w_{2,1} + x_2 \cdot w_{2,2} + \cdots + b_2\right)$$

...

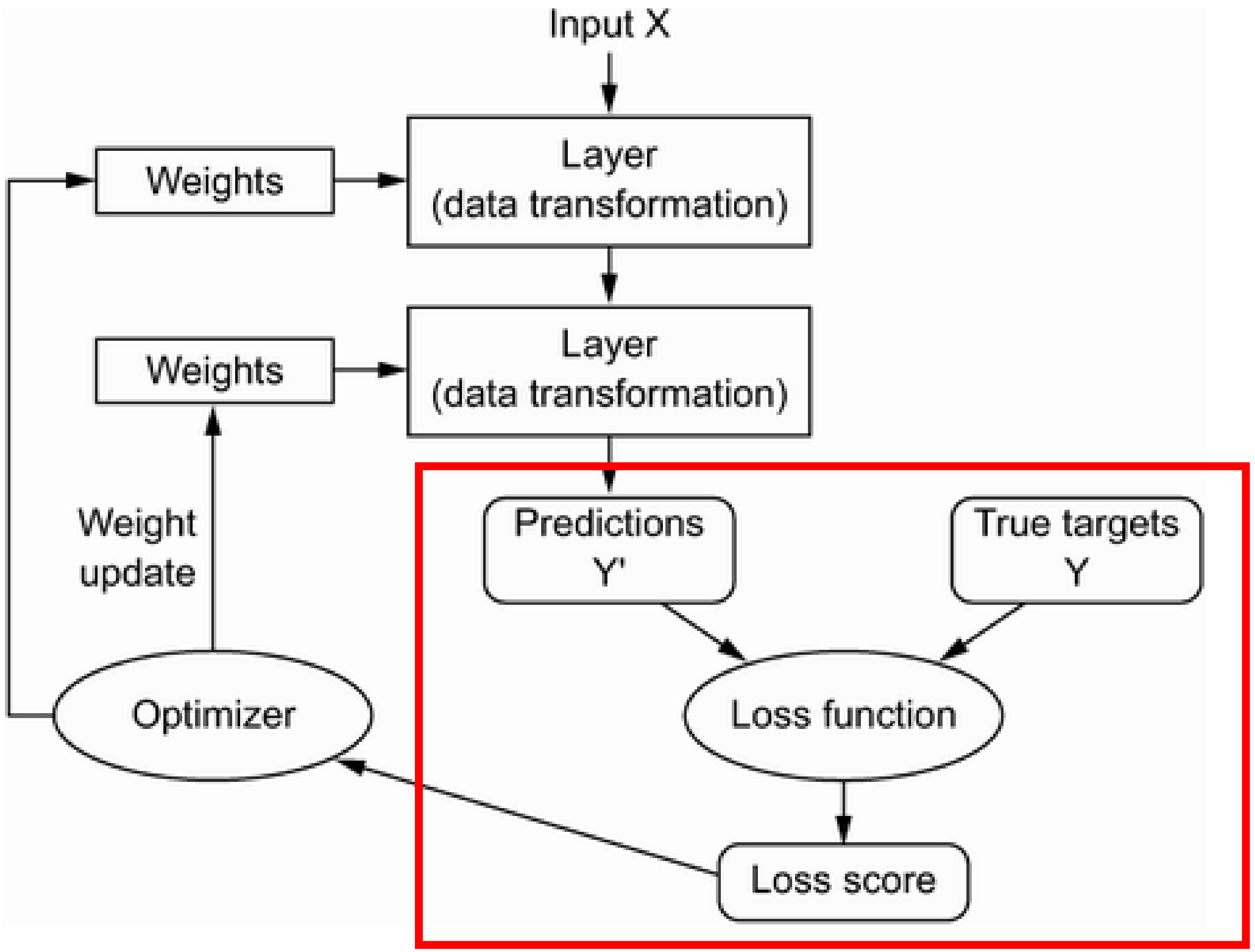
Eventually We Obtain Model's
Predictions

Multi-Layer Perceptron (MLP) –
Dense, Fully-connected, Feed-forward



© Gordon Burtch, 2026

Calculate Loss



© Gordon Burtch, 2026

Loss Functions

Cross-Entropy / Log-Loss

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

- Typical for binary outcomes. Value grows exponentially larger as the predicted probability moves away from the true 0,1 label.
- Multi-category outcomes have an analogous loss function known as categorical cross-entropy.

$$CE = -\sum_{i}^{C} t_i log(s_i)$$

MAE / L1 Loss

$$MAE = \frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{n}$$

- Typical for continuous outcomes. Errors are penalized homogenously, in magnitude and direction. This should look familiar!

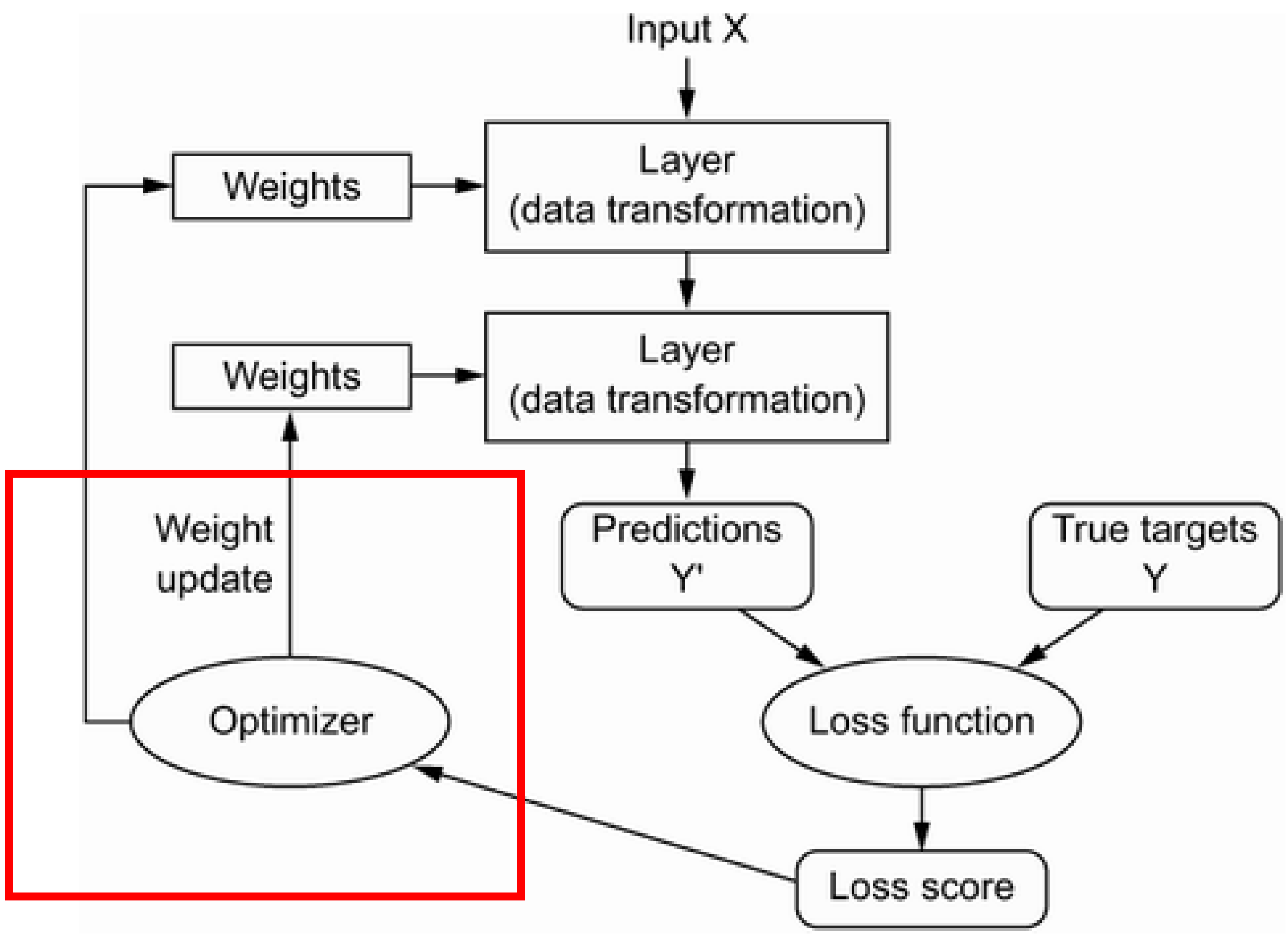
MSE / Quadratic / L2 Loss

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$$

- Typical for continuous outcomes, larger errors penalized exponentially more. This should look familiar!

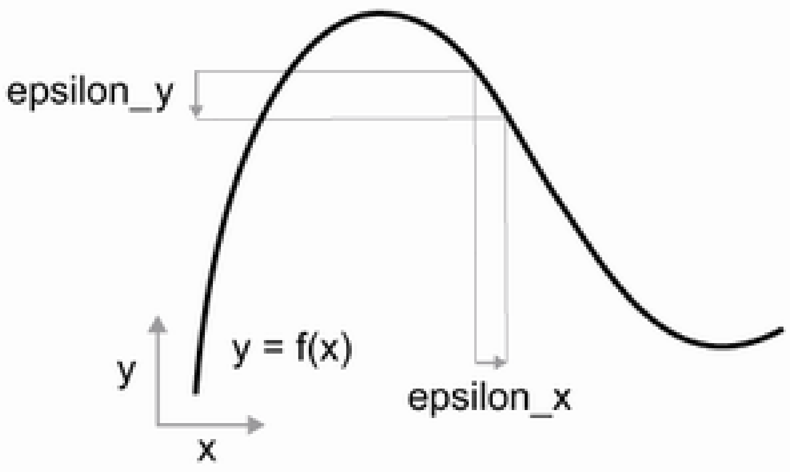
© Gordon Burtch, 2026

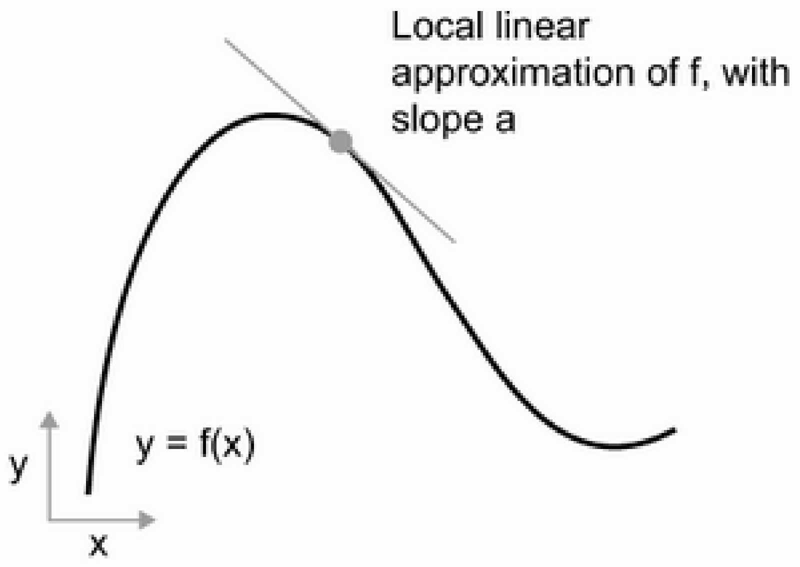
Backpropagation



© Gordon Burtch, 2026

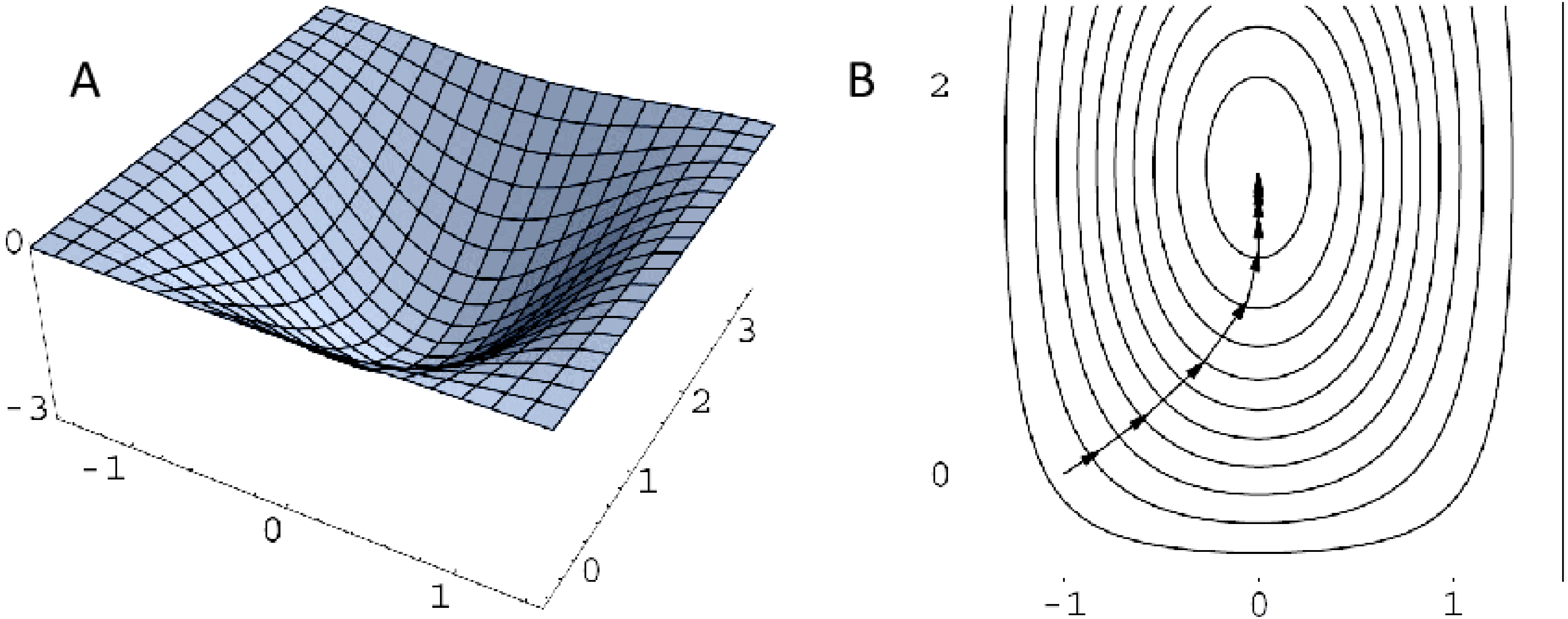
Derivative = “Rate” of Change





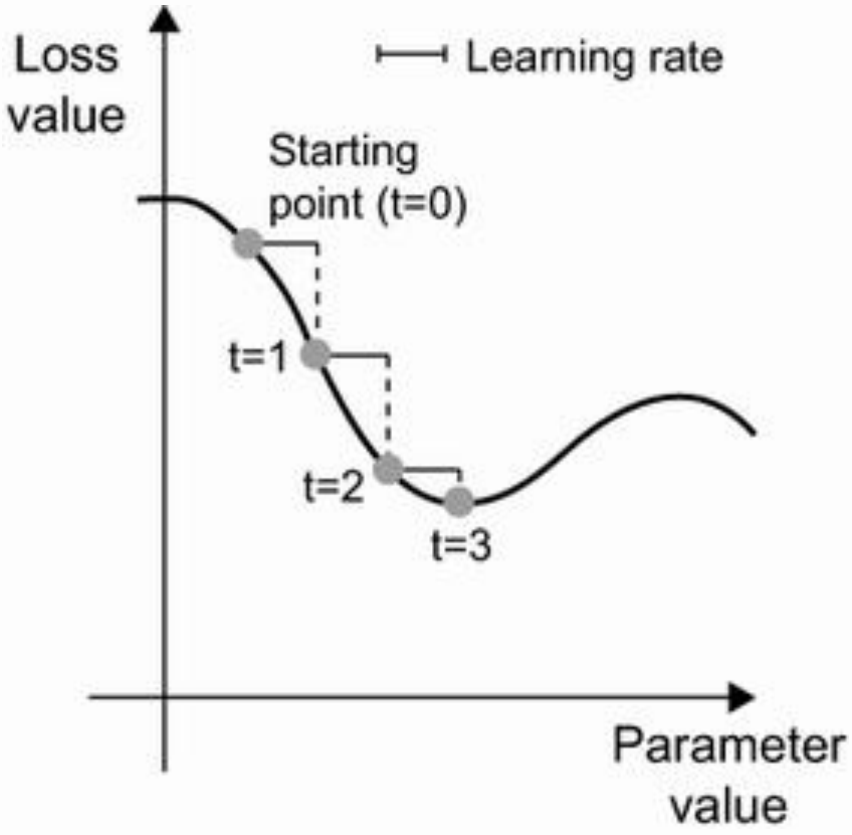
© Gordon Burtch, 2026

Gradient = Derivative in Multiple Dimensions



© Gordon Burtch, 2026

Gradient Descent



© Gordon Burtch, 2026

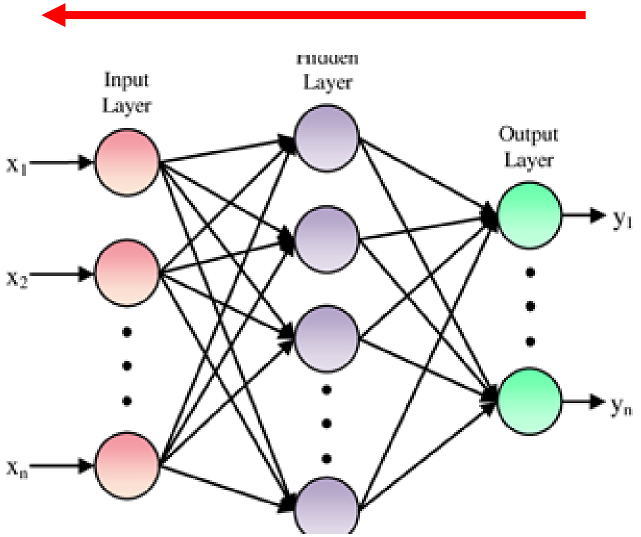
Derivatives of Loss w.r.t All Parameters

Recall that Each Node's Output Can be Expressed as a Function of the Prior Nodes' Outputs

$$y_1 = \varphi\left(x_1 \cdot w_{1,1} + x_2 \cdot w_{1,2} + \cdots + b_1\right)$$

$$y_2 = \varphi\left(x_1 \cdot w_{2,1} + x_2 \cdot w_{2,2} + \cdots + b_2\right)$$

...



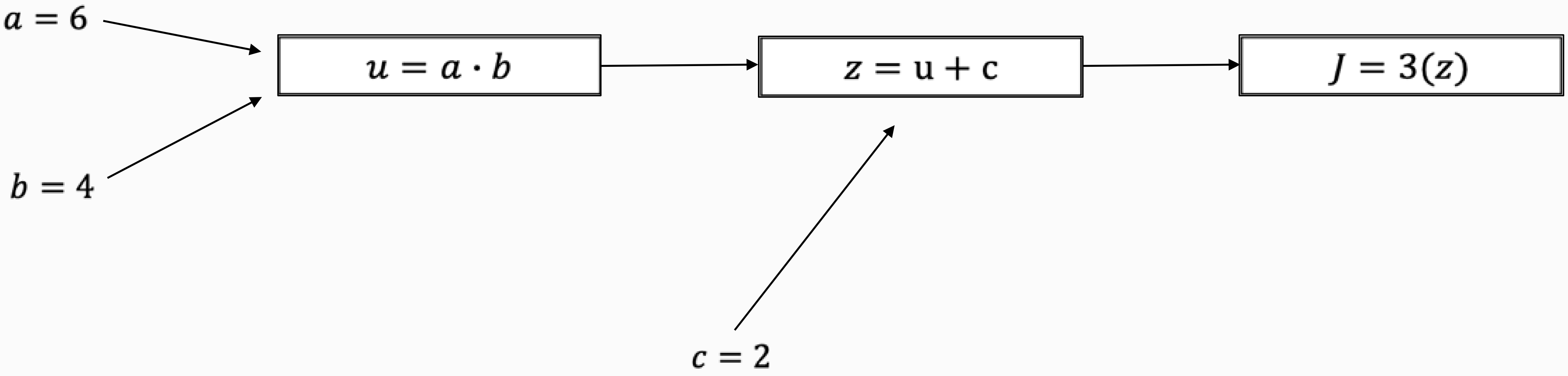
Start at the final nodes in the network and work backwards

- We calculate partial derivatives w.r.t. their inputs / weights.
- Then, use those partial derivatives and work backward into earlier layers to get partial derivatives w.r.t. *their* inputs / weights, and so on.

© Gordon Burtch, 2026

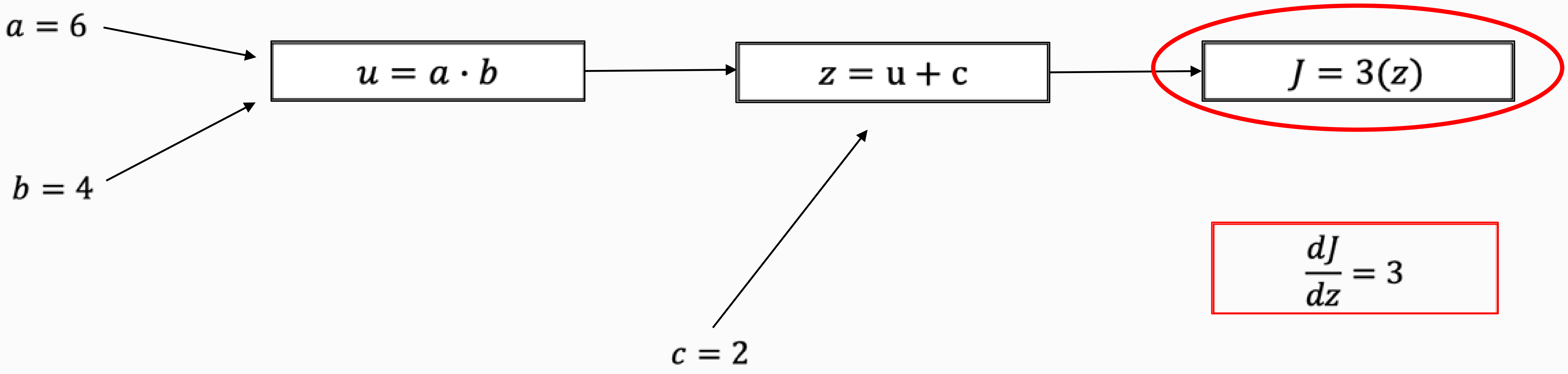
Simplifying Gradients: The Computation Graph

$$J = 3(a \cdot b + c)$$



Backpropagation = Working Backwards

$$J = 3(a \cdot b + c)$$



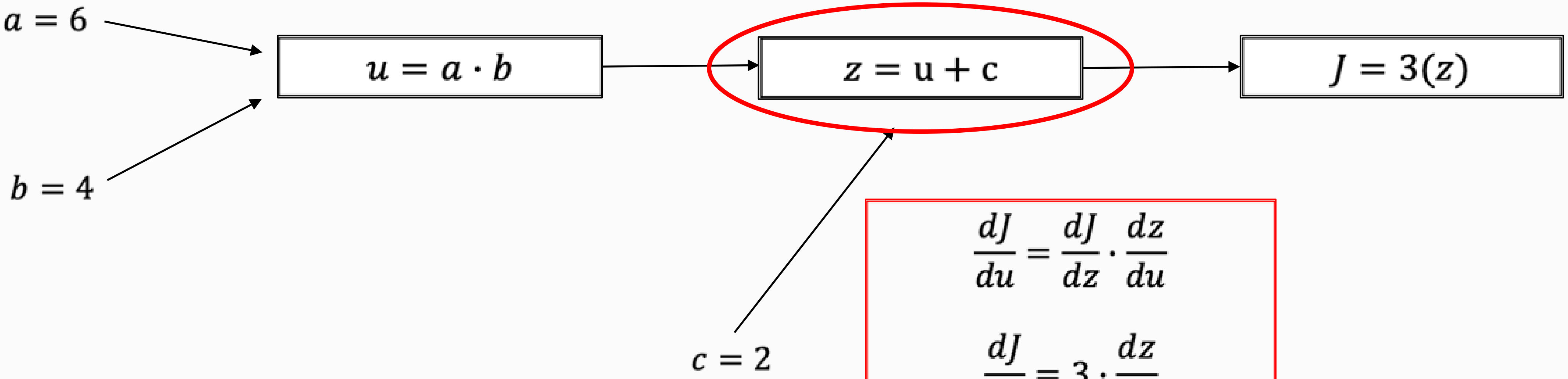
$$\boxed{\frac{dJ}{dz} = 3}$$

© Gordon Burtch, 2026

Backpropagation = Work Backwards

$$\frac{dJ}{dz} = 3$$

$$J = 3(a \cdot b + c)$$



$$\frac{dJ}{du} = \frac{dJ}{dz} \cdot \frac{dz}{du}$$

$$\frac{dJ}{du} = 3 \cdot \frac{dz}{du}$$

$$\frac{dJ}{du} = 3 \cdot 1 = 3$$

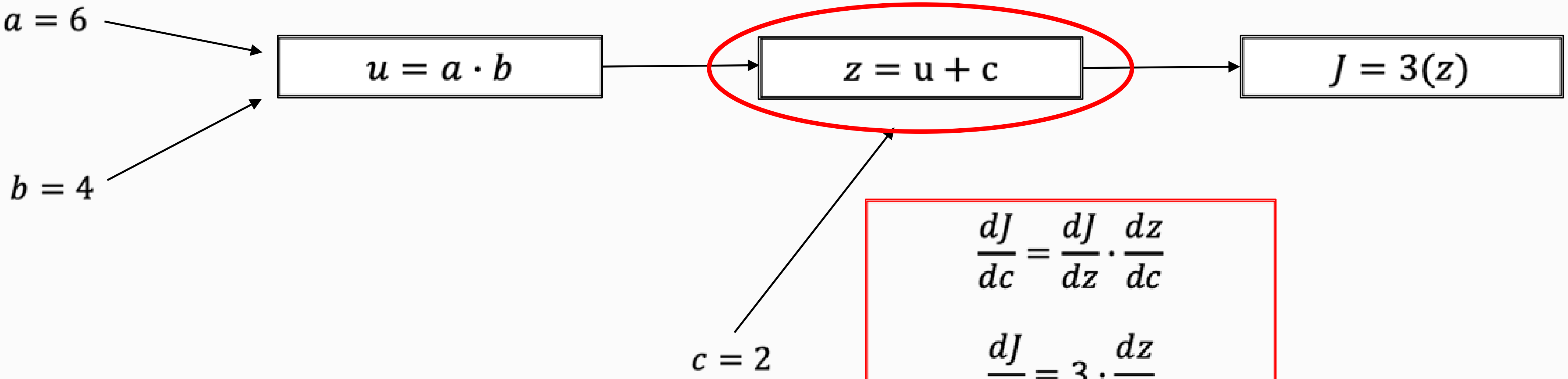
© Gordon Burtch, 2026

Backpropagation = Work Backwards

$$\frac{dJ}{dz} = 3$$

$$\frac{dJ}{du} = 3$$

$$J = 3(a \cdot b + c)$$



$$\frac{dJ}{dc} = \frac{dJ}{dz} \cdot \frac{dz}{dc}$$

$$\frac{dJ}{dc} = 3 \cdot \frac{dz}{dc}$$

$$\frac{dJ}{dc} = 3 \cdot 1 = 3$$

© Gordon Burtch, 2026

Backpropagation = Work Backwards

$$\frac{dJ}{dz} = 3$$

$$\frac{dJ}{du} = 3$$

$$\frac{dJ}{dc} = 3$$

$$J = 3(a \cdot b + c)$$

$$\begin{tikzcd}
a = 6 \arrow[dr] & & & \\
& \boxed{u = a \cdot b} \arrow[r] & \boxed{z = \mathrm{u} + \mathrm{c}} \arrow[r] & \boxed{J = 3(z)} \\
b = 4 \arrow[ur] & & c = 2 \arrow[u] &
\end{tikzcd}$$

$$\frac{dJ}{da} = \frac{dJ}{du} \cdot \frac{du}{da}$$

$$\frac{dJ}{da} = 3 \cdot b$$

$$\frac{dJ}{da} = 3 \cdot 4 = 12$$

© Gordon Burtch, 2026

Backpropagation = Work Backwards

$$\frac{dJ}{dz} = 3$$

$$\frac{dJ}{du} = 3$$

$$\frac{dJ}{da} = 12$$

$$\frac{dJ}{dc} = 3$$

$$J = 3(a \cdot b + c)$$

$a = 6$

$b = 4$

$c = 2$

$$u = a \cdot b \;\longrightarrow\; z = \mathrm{u} + \mathrm{c} \;\longrightarrow\; J = 3(\mathrm{z})$$

$$\frac{dJ}{db} = \frac{dJ}{du} \cdot \frac{du}{db}$$

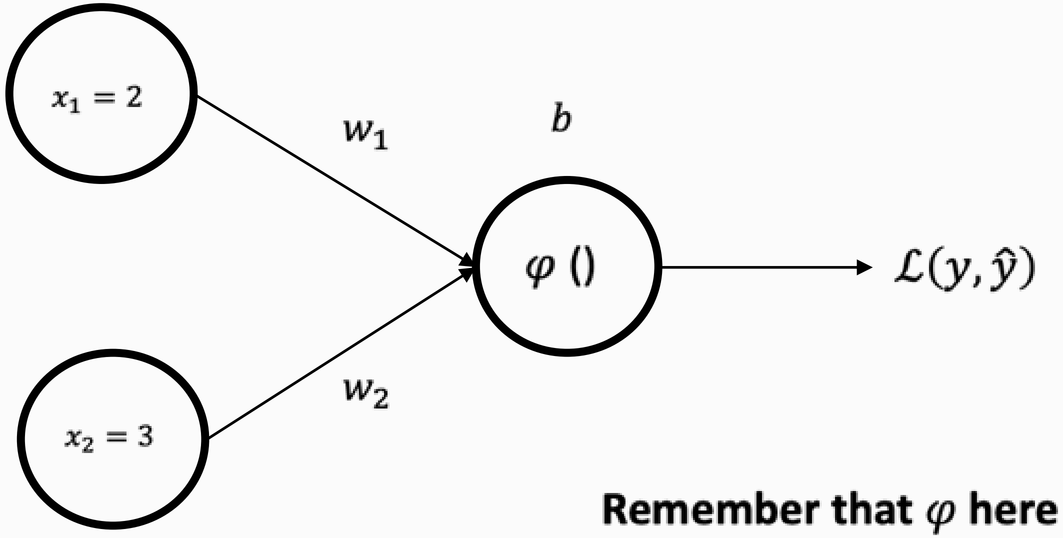
$$\frac{dJ}{db} = 3 \cdot a$$

$$\frac{dJ}{da} = 3 \cdot 6 = 18$$

We thus update our parameters, a, b, and c, subtracting each's gradients*epsilon from its current value. Epsilon is the learning rate.

© Gordon Burtch, 2026

Single Node with Sigmoid & Cross-Entropy Loss (i.e., Logistic Regression)



$$\frac{d\mathcal{L}}{d\varphi} = -\frac{y}{\varphi} + \frac{1-y}{1-\varphi}$$

$$\frac{d\mathcal{L}}{d\varphi} = \frac{\varphi(1-y) - y(1-\varphi)}{\varphi(1-\varphi)}$$

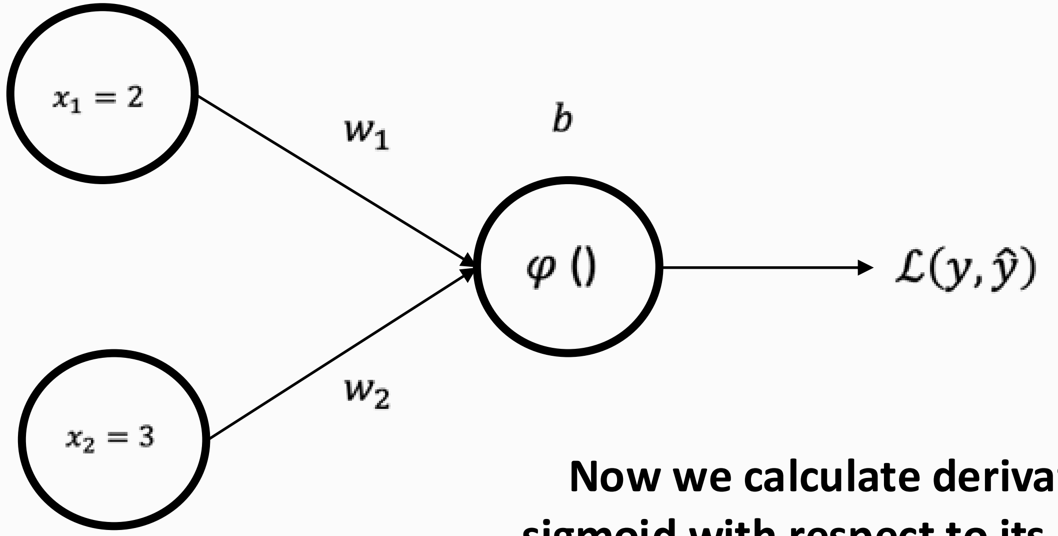
$$\frac{d\mathcal{L}}{d\varphi} = \frac{\varphi - \varphi y - y + \varphi y}{\varphi(1-\varphi)}$$

$$\frac{d\mathcal{L}}{d\varphi} = \frac{\varphi - y}{\varphi(1-\varphi)}$$

Remember that φ here is just a placeholder for the argument to the loss function. It happens to be a sigmoid transformation of 'something', i.e., φ(wx+b), but it doesn't really matter. We just represent it with some variable name and calculate an expression for the derivative.

© Gordon Burtch, 2026

Single Node with Sigmoid & Cross-Entropy Loss (i.e., Logistic Regression)



Now we calculate derivative of the sigmoid with respect to its argument, z.

$$\varphi(z) = (1 + e^{-z})^{-1}$$

$$\varphi'(z) = -1 \cdot (1 + e^{-z})^{-2} \cdot (0 + e^{-z} \cdot -1)$$

$$\varphi'(z) = (1 + e^{-z})^{-2} \cdot e^{-z}$$

$$\varphi'(z) = \varphi(z) \cdot (1 - \varphi(z))$$

$$\frac{d\mathcal{L}}{dz} = \frac{d\mathcal{L}}{d\varphi} \cdot \frac{d\varphi}{dz}$$

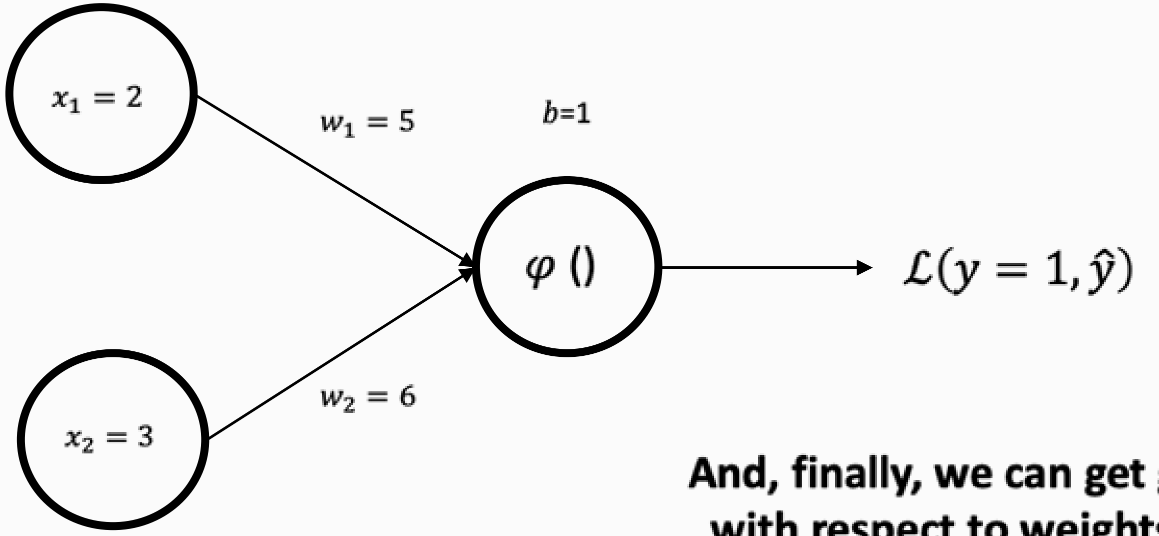
$$\frac{d\mathcal{L}}{dz} = \frac{\varphi - y}{\varphi(1 - y)} \cdot \frac{d\varphi}{dz}$$

$$\frac{d\mathcal{L}}{dz} = \frac{\varphi - y}{\varphi(1 - y)} \cdot \varphi(1 - \varphi)$$

$$\frac{d\mathcal{L}}{dz} = \varphi - y$$

© Gordon Burtch, 2026

Single Node with Sigmoid & Cross-Entropy Loss (i.e., Logistic Regression)



$$\frac{d\mathcal{L}}{dw_1} = \frac{d\mathcal{L}}{dz} \cdot \frac{dz}{dw_1}$$

$$\frac{d\mathcal{L}}{dw_1} = (\varphi - y) \cdot x_1$$

$$w_{1,new} = w_{1,old} - (\frac{d\mathcal{L}}{dw_{1,old}} \cdot \varepsilon)$$

And, finally, we can get gradient of loss with respect to weights and bias. For example, for the first weight…

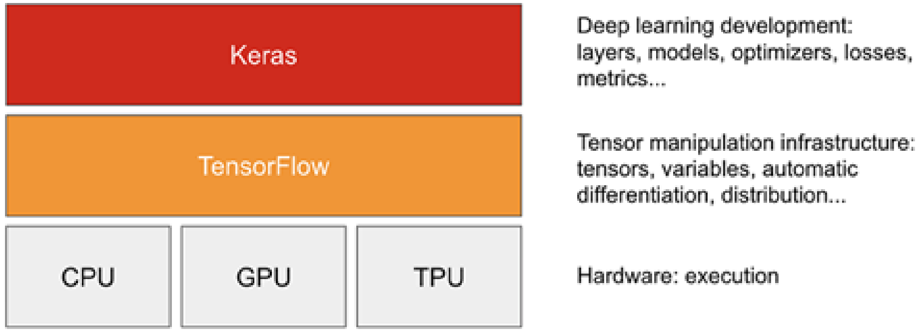
Evaluate φ based on current values of parameters and the data.

Finally, update the weights…

© Gordon Burtch, 2026

Keras and Tensorflow

1. Tensorflow
   - A Python platform for working with tensors, implementing automatic differentiation, providing access to repositories of (well-known) pre-trained models.
2. Keras
   - A higher-level API that wraps common usage patterns with Tensorflow functions, pre-defined loss functions, optimization algorithms, etc.
   - Keras simplifies data scientists' interaction with Tensorflow.

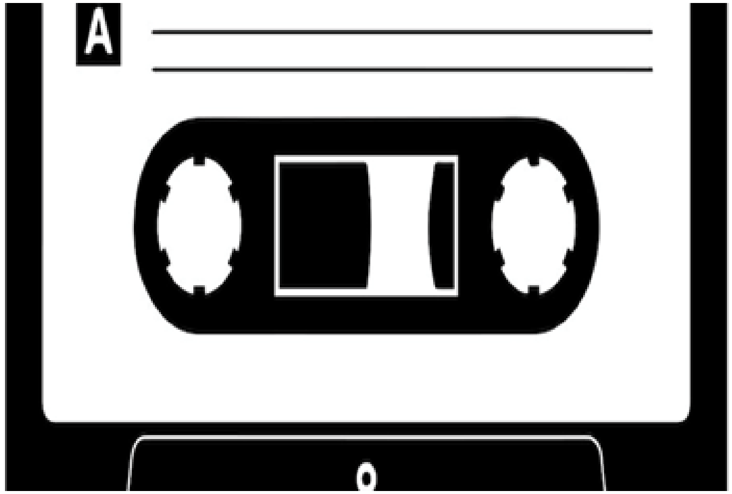


© Gordon Burtch, 2026

Tensorflow GradientTape: AutoDiff

1. Gradient Tape

- A Tensorflow function that automates the calculation of derivatives.
- It constructs a computation graph in the background and implements codified rules for calculating derivatives of functions.
- You could technically use gradient tape to implement a gradient descent algorithm for many optimization problems.

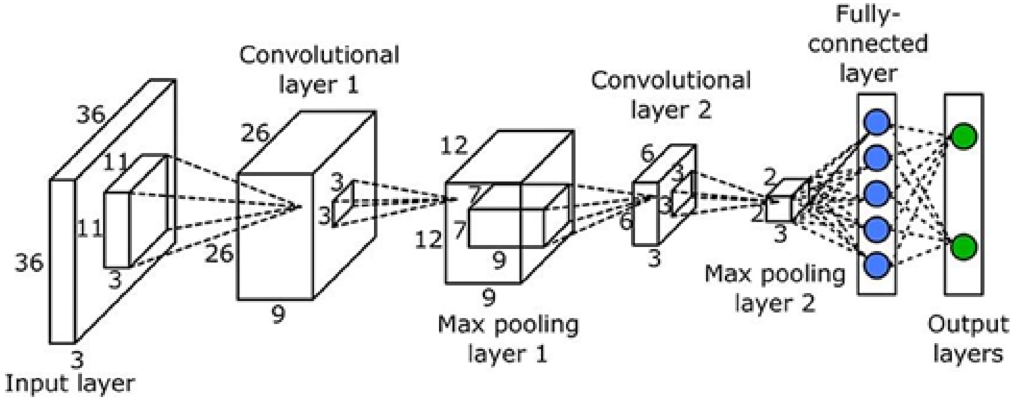


© Gordon Burtch, 2026

Keras Layers

Layers are the Key Building Block of NNs in Keras

- There are a few subclasses of the Layers class: e.g., Dense is the one we have seen so far, i.e., layers.Dense(), but we also have convolutional layers, max-pooling layers, recurrent layers, and so on. There are many pre-defined layers in Keras. See: https://keras.io/api/layers/.
- These are different architectural components that can be mixed and matched in different ways to create different network topologies.
- It is also possible to construct custom layers.



© Gordon Burtch, 2026

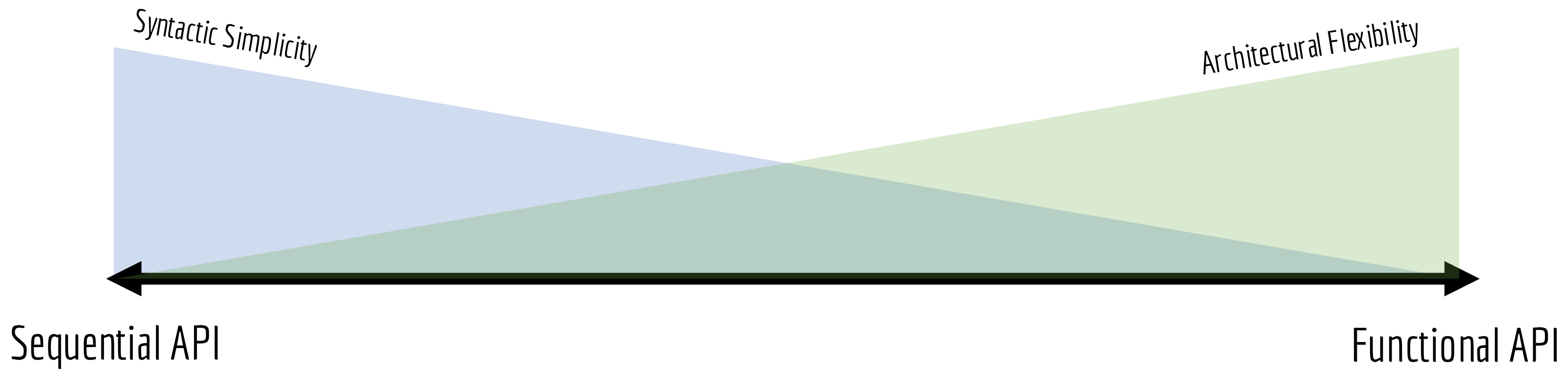
Sequential vs. Functional API

We Have Only Used Sequential API So Far

- Sequential is easy to work with but is also very inflexible. Can only really handle basic feed-forward networks. It automatically figures out the shape of each layer's output tensor and specifies the next layer's input shape accordingly.

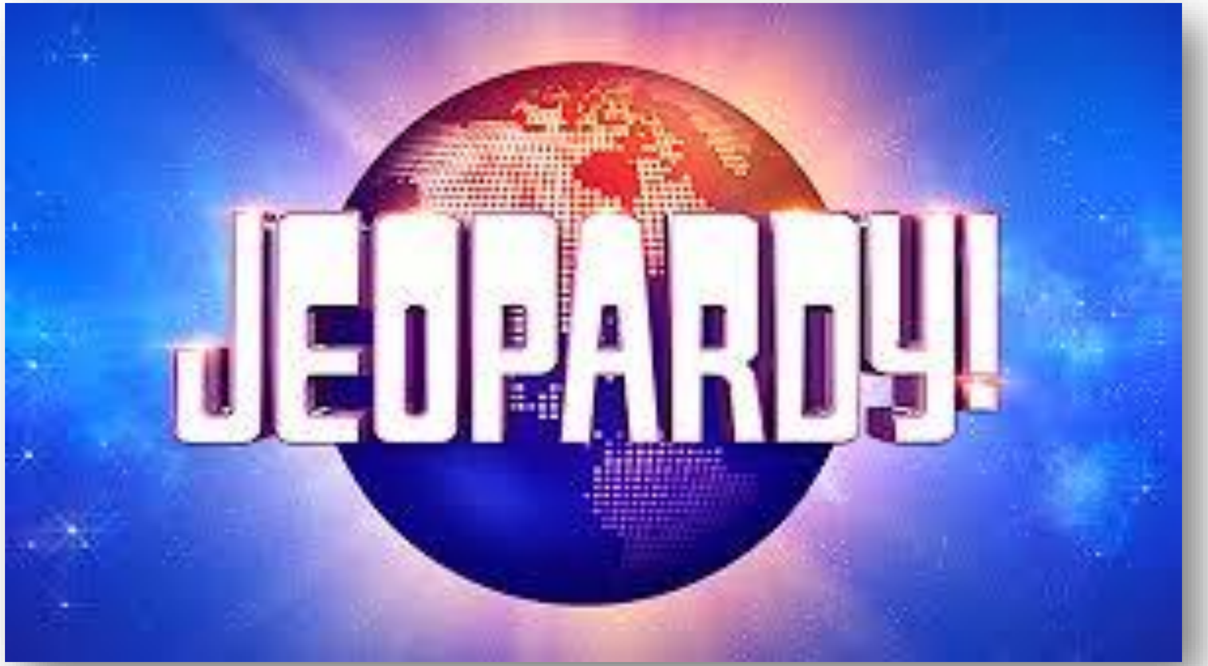
Functional API Let's You Construct Any Topology You Want

- But – we will look at the difference in how each API is used, syntactically.



© Gordon Burtch, 2026

Quiz Prep: Review of Concepts!

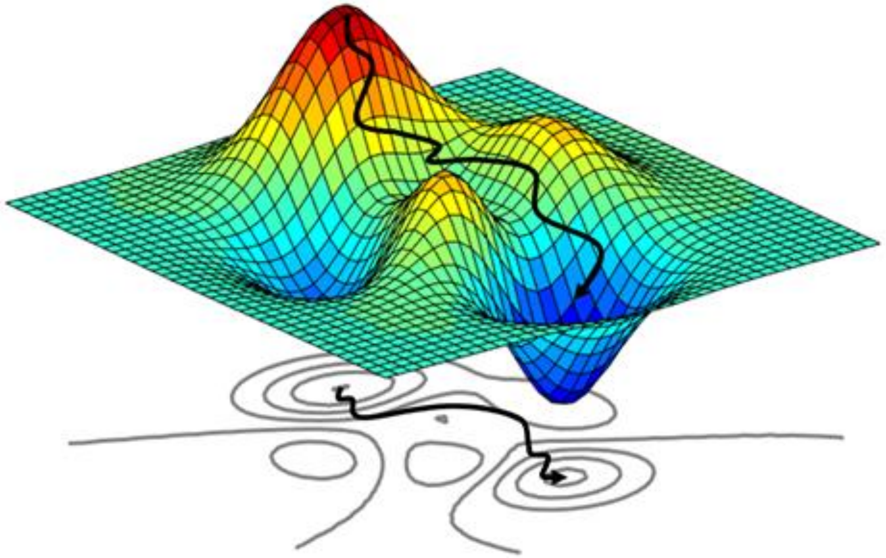


© Gordon Burtch, 2026

Optimizers

Keras Supports 8 Optimizers

- SGD = Stochastic Gradient Descent
- Momentum
- Ftrl (2010) = Follow the Regularized Leader
- Adagrad and Adadelta (2012) = Adaptive Gradient Des
- RMSprop (~2012) = Root Mean Squared propagation
- Adam (2015) = Adadelta / RMSProp with Momentum.
    - Adamax, Nadam are extensions to Adam.



SGD: Gradient Descent

Types of GD

- Batch GD = Use all the available training data in each pass.
  - Works well if the loss surface is smooth and lacks any saddle points / valleys.
- Stochastic GD = Mini-batch with batch size = 1.
  - If troughs / saddles exist, we move past them as our exploration of gradients for the model will vary withe a given observation that we are considering in an iteration.
  - Computationally quite burdensome but performs well on non-linear problems (eventually).
- Mini-batch GD = What we have been doing so far (randomly split the data in each epoch, into folds, and then cycle over the folds for training).
  - This is a happy-medium between batch and stochastic GD.

Role of Batch Size

- Empirically has been observed that smaller batches yield less overfitting (because of implicit noise in the training process – variance of the gradients obtained will go up).

© Gordon Burtch, 2026

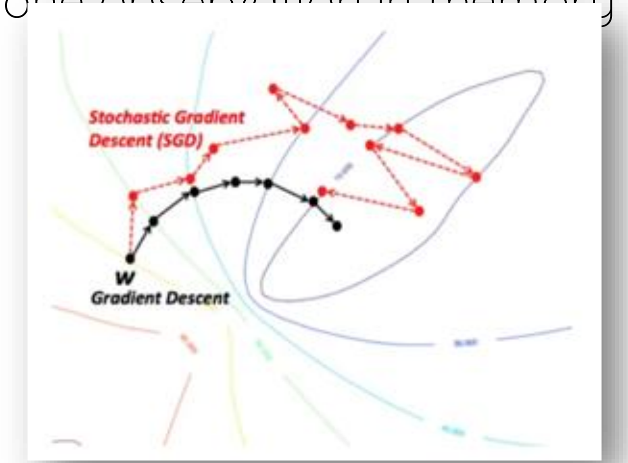
Batch (All) vs. Stochastic (1)

Same Convergence

- If you have a convex surface, either approach will converge to the global optimum (no guarantee your problem is convex of course). Always converges at least to a local minimum.

Tradeoff s

- Batch, each step is slower, more computationally burdensome, but convergence with fewer iterations; Need to be able to hold the entire dataset in memory.
- SGD makes noisier updates, and requires more iterations to converge, but a single iteration is quick. Only need one observation in memory at a time.



Momentum

Getting Past Local Minima

- SGD gets stuck in local minima; the idea of momentum is to make updates be a function of current gradient*learning rate, as well as some fraction (decay) of the update you made last iteration.
- This reduces updates to parameters where the gradients are flipping sign and amplifies updates to gradients that are going in a consistent direction (steeply descer

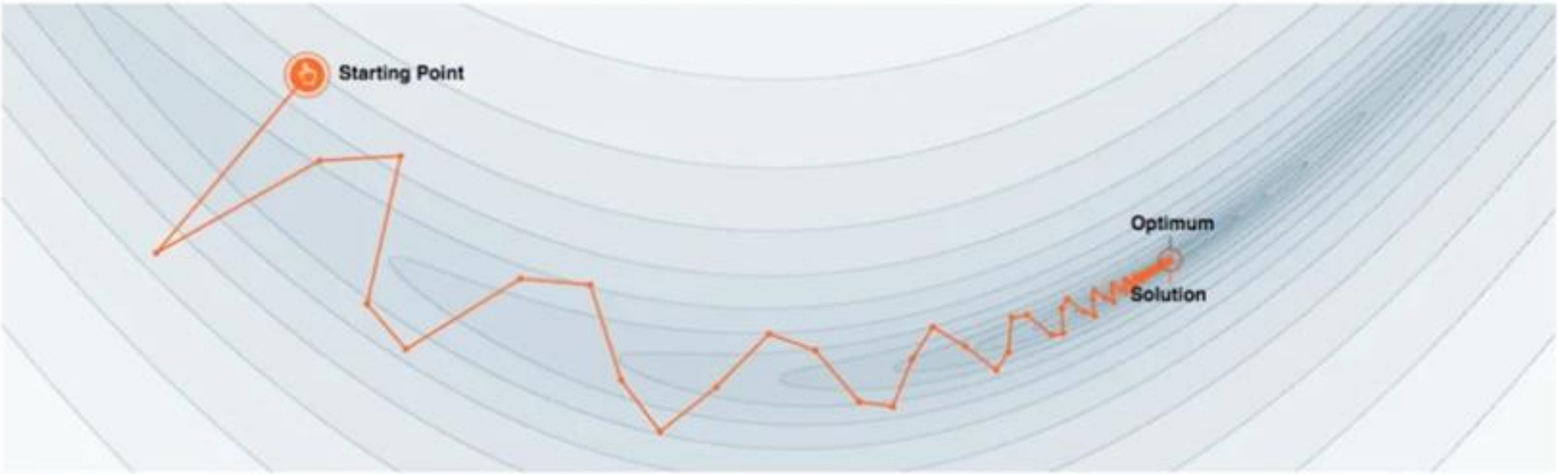


Figure: Optimization with momentum (Source: distill.pub)

© Gordon Burtch, 2026

Adagrad & Adadelta (RMS Prop)

Adaptive Gradient Descent (Variable Learning Rate)

- We implicitly apply a high learning rate for features we have been updating very little so far (speed up movement through saddle points, for example).
- We implicitly apply a low learning rate for features we have been updating a lot so far.
- Technically learning rate is removed from the process, every update is a function of past updates.

Adadelta

- Same idea but we use a sliding window of previous updates to determine magnitude of current updates (rather than all prior updates).
- RMSProp is conceptually very similar but was independently developed (around the same time).

© Gordon Burtch, 2026

Recap

Building Blocks of NNs

- Tensors and Tensor Operations
- Activation Functions
- Loss Functions
- Backpropagation: Derivatives, Gradients & the Chain Rule

Procedure of Minibatch Stochastic Gradient Descent

- Grab a batch of observations (samples)
- Predict their labels using current weights / bias terms.
- Calculate loss value.
- Calculate gradient of loss w.r.t. all weight / bias terms.
- Update each weight by subtracting its gradient*learning rate
- Cycle over the whole training dataset (each cycle is an epoch) repeatedly, until loss is small.

© Gordon Burtch, 2026