The Language of Math

A Field Guide to Reading AI and ML Equations

Equations are sentences.
Symbols are only vocabulary.
Once you know the words, you can read the sentence.

A phrasebook and grammar guide, not a math course.

@manpatel.ml • manpatel.me

Contents

Introduction: Math as a Language **2**

1 The Alphabet **3**

2 Punctuation and Connectors **6**

2.1 Building and Comparing 6

2.2 Set Notation 6

2.3 Calculus Notation 7

2.4 Linear Algebra Notation 7

2.5 Probability and Statistics Notation 8

3 Grammar Rules **9**

3.1 Subscripts: Pointing at One Piece of Many 9

3.2 The Superscript Trap 9

3.3 Little Decorations: Hats, Bars, Tildes, and Primes 9

3.4 Font Matters: Scalars, Vectors, Matrices, and Sets 10

4 Common Sentence Patterns **11**

4.1 The Average Value Of 11

4.2 The Probability Of, Given 11

4.3 The Input That Makes This Biggest 11

4.4 Roughly How the Cost Grows 11

5 Reading Equations Sentence by Sentence **12**

6 Practice What You Have Learned **16**

7 Quick Reference Cheat Sheet **18**

Closing Notes **20**

Introduction: Math as a Language

You already know more than you think. If somebody explains an idea from an AI paper in plain English, you can usually follow it just fine. The trouble starts when that same idea gets written as $\theta := \theta - \alpha \nabla J(\theta)$, and it suddenly feels like the page is speaking a different language.

It is speaking a different language. That is actually the good news, because languages can be learned, and this one has a very small vocabulary compared to English or French. Most of AI and machine learning notation is built from around thirty symbols, a handful of grammar rules about where those symbols get placed, and a few recurring sentence patterns that show up again and again once you know to look for them.

Think of an equation the way you would think of a sentence. The symbols are the words. The position of a symbol, whether it sits as a subscript, a superscript, above another symbol, or between two other symbols, is the grammar. And once you have seen a few dozen equations, you start noticing that most of them are really just a small number of underlying templates wearing different letters, the same way most English sentences are a subject, a verb, and an object arranged in a few reliable orders.

This guide treats reading equations exactly like learning to read a new language. It will not teach you calculus or linear algebra from first principles, there are excellent courses for that. What it will do is make sure that the next time you see σ, ∇, or a small (i) tucked above a letter, you know immediately what that piece of notation is doing, the same way a fluent reader recognizes a word without sounding out each letter.

How to use this guide. Sections 1 through 4 build your vocabulary and grammar: the Greek alphabet, the common symbols, and the rules for reading subscripts and superscripts. Section 5 puts all of that to work on real equations pulled straight from machine learning, walking through each one piece by piece. Section 6 lets you practice on your own. Section 7 is a compact cheat sheet you can return to any time you hit a symbol you have forgotten, the way you would look up a word in a dictionary rather than memorizing the whole dictionary up front.

1 The Alphabet

Mathematicians ran out of Latin letters a long time ago. Once x, y, n, and their neighbors were all claimed for everyday variables, the Greek alphabet became the natural place to borrow more symbols from, and over the last few centuries certain Greek letters settled into specific jobs by convention. Nobody voted on this. It simply became common practice for θ to mean parameters and λ to mean a regularization strength, the same way it became common practice to use i and j as loop counters in code.

That means the table below is not a set of hard rules. It is closer to a list of a symbol's most frequent job, the role it plays in the majority of papers and courses you will read. Context always has the final word.

Symbol	Name	Said like	Common meaning in AI and ML
α	alpha	AL fuh	Learning rate, a coefficient, or a significance level in statistics
β	beta	BAY tuh	Regression coefficients, or the decay rates inside the Adam optimizer
γ, Γ	gamma	GAM uh	The discount factor in reinforcement learning (lowercase); the gamma function used in several probability distributions (uppercase)
δ, Δ	delta	DEL tuh	A small change or an error term (lowercase); a finite change, as in ΔW (uppercase)
ϵ	epsilon	EP sih lon	A tiny positive number used to avoid dividing by zero; the exploration rate in epsilon greedy strategies
ζ	zeta	ZAY tuh	Rarely used. Occasionally a noise term
η	eta	AY tuh	A learning rate, used as an alternative to α
θ, Θ	theta	THAY tuh	The parameters of a model, collectively (lowercase). This is one of the most common symbols in all of ML
ι	iota	eye OH tuh	Almost never used in machine learning
κ	kappa	KAP uh	The condition number of a matrix; Cohen's kappa, a statistic for measuring agreement
λ	lambda	LAM duh	Regularization strength; an eigenvalue; the rate parameter of a Poisson or exponential distribution

Symbol	Name	Said like	Common meaning in AI and ML
μ	mu	MEW	The mean of a distribution. Occasionally a momentum term
ν	nu	NEW	Degrees of freedom in a t distribution; occasionally a learning rate
ξ	xi	ZY or KSEE	A slack variable, as in support vector machines; a noise term
o	omicron	OM ih kron	Almost never used directly. Big O notation, as in $O(n^2)$, uses the plain Latin letter O, not this Greek letter
π, Π	pi	PIE	The constant 3.14159 or a policy in reinforcement learning (lowercase); the product operator, multiplication's version of summation (uppercase)
ρ	rho	ROH	A correlation coefficient
σ, Σ	sigma	SIG muh	Standard deviation, or the sigmoid function (lowercase); the summation operator, or a covariance matrix (uppercase)
τ	tau	TOW, rhymes with how	A time constant, or a temperature parameter that softens or sharpens a probability distribution
υ	upsilon	UP sih lon	Almost never used in machine learning
ϕ, Φ	phi	FEE or FY	A feature transformation function (lowercase); the cumulative distribution function of the normal distribution (uppercase)
χ	chi	KY, rhymes with sky	Appears in the chi squared distribution and the chi squared test
ψ	psi	SY, also rhymes with sky	Another symbol used for a feature transformation function
ω, Ω	omega	oh MAY guh	An alternative symbol for weights (lowercase); a lower bound in complexity analysis, or the full sample space in probability (uppercase)

Watch out

Two mix ups catch almost everyone at some point. First, Big O notation is written with the plain Latin letter O, not the Greek letter omicron, even though the two are drawn identically. Second, χ is pronounced like the word sky without the s, not like the drink chai. Small thing, but it will save you from an odd look in a study group.

2 Punctuation and Connectors

If Greek letters are the vocabulary, the symbols in this section are the punctuation and connective tissue that hold an equation together. This is a longer section because there are simply more of these symbols than there are Greek letters in common use, so it is split by category. You do not need to memorize all of it in one sitting. Read through it once, then use Section 7 later as your quick lookup.

2.1 Building and Comparing

- $=$ a claim that two things are exactly equal
- $\approx$ approximately equal, close enough that the small difference does not matter here
- $:=$ or $\triangleq$ defined as. This is different from a plain equals sign because it introduces a new symbol rather than claiming two already known things are equal
- $\propto$ proportional to, meaning that as one quantity changes, the other changes in the same pattern, up to some constant multiplier that is being left out for simplicity
- $\neq, \leq, \geq$ not equal to, less than or equal to, greater than or equal to, exactly as in ordinary arithmetic

2.2 Set Notation

- $\in$ is an element of, read simply as is in. So $x \in \mathbb{R}$ reads as x is in the real numbers
- $\notin$ is not an element of
- $\subset, \subseteq$ is a subset of, meaning every member of the first set also belongs to the second
- $\forall$ for all, or for every
- $\exists$ there exists, meaning at least one such thing can be found
- $\emptyset$ the empty set, a set containing nothing
- $\mathbb{R}, \mathbb{Z}, \mathbb{N}, \mathbb{Q}, \mathbb{C}$ the real numbers, integers, natural numbers, rational numbers, and complex numbers
- $\mathbb{R}^n$ an n dimensional real space, in other words a vector made of n real numbers. This appears constantly in ML, since a data point with n features is a single point in $\mathbb{R}^n$

2.3 Calculus Notation

- $\frac{d}{dx}$ the ordinary derivative, the rate of change of something with respect to x, when x is the only variable involved
- $\frac{\partial}{\partial x}$ the partial derivative, the rate of change with respect to x specifically, while every other variable is held frozen. Used constantly once a function depends on more than one input
- ∇ nabla, also called the gradient. This takes a function of several variables and stacks all of its partial derivatives into a single vector, one entry per input variable. It points in the direction where the function increases fastest
- $\int$ the integral, the continuous version of a sum, often described as the area under a curve
- ∞ infinity, an unbounded quantity rather than an actual number

2.4 Linear Algebra Notation

- Bold lowercase, like $\mathbf{x}$, usually means a vector, a list of numbers. Bold uppercase, like $\mathbf{W}$, usually means a matrix, a grid of numbers. This is a convention rather than a strict rule, so plain italic letters sometimes stand in for vectors or matrices too, especially once a paper has already stated what each symbol represents
- $\mathbf{A}^\top$ the transpose of $\mathbf{A}$, meaning its rows and columns are swapped
- $\mathbf{A}^{-1}$ the inverse of $\mathbf{A}$, the matrix that undoes what $\mathbf{A}$ does. This looks like an exponent of negative one, but it does not mean one divided by $\mathbf{A}$, matrices are not divided in that sense
- $\mathbf{x} \cdot \mathbf{y}$ the dot product, which combines two vectors of the same length into a single number
- $\mathbf{x} \odot \mathbf{y}$ the element wise product, sometimes called the Hadamard product, which multiplies matching entries and keeps the same shape rather than collapsing to one number
- $\|\mathbf{x}\|$ the norm, the length or magnitude of a vector. A small subscript often follows, where $\|\mathbf{x}\|_2$ is the ordinary straight line length and $\|\mathbf{x}\|_1$ adds up the absolute values of every entry instead
- $\mathbf{I}$ the identity matrix, the matrix version of the number one, which leaves anything it multiplies unchanged
- $\det(\mathbf{A})$ the determinant, a single number that summarizes how much a matrix stretches or shrinks space
- $\mathbf{Av} = \lambda\mathbf{v}$ the eigenvalue equation. It reads as, there exists a special direction $\mathbf{v}$ that, when transformed by $\mathbf{A}$, is only stretched or shrunk by the number λ, rather than being rotated or bent

2.5 Probability and Statistics Notation

- $\mathbb{E}[X]$ the expectation of X, its average value once you account for how likely each outcome is
- $\text{Var}(X)$, $\text{Cov}(X, Y)$ variance and covariance, how spread out a variable is, and how two variables move together
- $P(A)$ the probability of A
- $\sim$ the tilde, meaning is distributed as. So $X \sim \mathcal{N}(\mu, \sigma^2)$ reads as X follows a normal distribution with mean μ and variance σ^2
- $\mid$ the vertical bar inside a probability expression means given. So $P(A \mid B)$ reads as the probability of A, given that B is already known to be true

3 Grammar Rules

Knowing the vocabulary only gets you halfway. The other half is knowing what it means when a symbol gets a small letter tucked underneath it, or above it, or wears a hat, a bar, or a little tilde. This is the grammar of math notation, and unlike English grammar, almost all of it comes down to four rules.

3.1 Subscripts: Pointing at One Piece of Many

A subscript is almost always an index. It answers the question, which one. x_i is the i-th element of x. w_{jk} is the specific weight connecting unit j to unit k. x_t is the value of x at time t. Whenever you see a small letter sitting low and to the right of a symbol, read it as pointing a finger at one particular item inside a larger collection.

3.2 The Superscript Trap

Superscripts cause more confusion than everything else in this guide combined, because the same little raised mark is used for at least three different jobs.

The single most useful rule in this guide

Check whether the superscript has parentheses or brackets around it. x^2, with nothing wrapped around the 2, means x multiplied by itself, an ordinary power. $x^{(i)}$, with parentheses around the i, almost always means the i-th training example in a dataset, not x raised to a power. In some deep learning material, square brackets are used instead, so $a^{[l]}$ means the value at layer l. This inconsistency is not a gap in your understanding, it genuinely varies between textbooks and papers. The habit to build is simple: always glance for parentheses or brackets before assuming a superscript is a power, and if the resulting power would be a strange, oversized number, that is usually a sign you are looking at an index instead.

3.3 Little Decorations: Hats, Bars, Tildes, and Primes

- A hat, as in $\hat{y}$ or $\hat{\theta}$, means an estimate of or a prediction of. $\hat{y}$ is the model's guess, while plain y is the true, observed answer
- A bar, as in $\bar{x}$, means the average of. $\bar{x}$ is the sample mean of x
- A tilde sitting directly over one symbol, as in $\tilde{\theta}$, usually means a modified or perturbed version of that symbol. Note that this is a different job from the tilde sitting between two symbols, as in $X \sim \mathcal{N}(0,1)$, which instead means is distributed as. Same squiggle, two different jobs depending on where it sits

- A prime, as in $f'(x)$, means the derivative of f when attached to a function. Attached to a variable instead, as in x', it more often means a transformed or next value, which is common in reinforcement learning for the next state

3.4 Font Matters: Scalars, Vectors, Matrices, and Sets

The style of the letter itself is information. As a rough guide across most papers you will read:

- Plain lowercase italic, like x, is a single number, called a scalar
- Bold lowercase, like $\mathbf{x}$, is a vector, a list of numbers
- Bold uppercase, like $\mathbf{X}$ or $\mathbf{W}$, is a matrix, a grid of numbers
- Plain uppercase italic, like X, is often a random variable, though it can also denote a set or a matrix in papers that skip bold font. The surrounding sentence usually settles which one is meant
- Calligraphic or blackboard style uppercase, like $\mathcal{D}$ for a dataset or $\mathbb{R}$ for the real numbers, denotes a special, named set

4 Common Sentence Patterns

Beyond single symbols, a handful of short phrases show up so often that it is worth learning them as fixed expressions, the way you would learn that in French qu'est ce que c'est is simply a fixed way of asking what is that, rather than translating it word by word every time.

4.1 The Average Value Of

$\mathbb{E}_{x\sim p}[f(x)]$ reads as, the average value that $f(x)$ takes, if you kept drawing x from the distribution p over and over. The subscript under $\mathbb{E}$ simply tells you which distribution the average is being taken over.

4.2 The Probability Of, Given

$P(y \mid x)$ reads as, the probability of y, given that x is already known. The vertical bar is doing the entire job of the English word given. This single pattern, a probability with a bar in the middle, is the backbone of nearly every classifier you will encounter, since a classifier is really just a machine for estimating $P(\text{label} \mid \text{input})$.

4.3 The Input That Makes This Biggest

$\max_\theta f(\theta)$ and $\arg\max_\theta f(\theta)$ look similar but answer different questions. The first gives you the biggest output value that f can achieve. The second gives you the particular input θ that achieves that biggest value. In machine learning, you almost always want argmax, because you care which set of parameters or which class label is best, not merely how good the best one happens to be. The same distinction applies to min and argmin.

4.4 Roughly How the Cost Grows

$O(n^2)$ reads as, as the input size n grows large, the cost grows roughly proportional to n squared. Big O notation describes a growth trend as things get large, while deliberately ignoring constant multipliers and smaller terms that stop mattering at scale. It is a description of shape, not an exact count.

5 Reading Equations Sentence by Sentence

This is where the vocabulary and grammar from the last four sections earn their keep. Each equation below is one you will run into constantly in AI and ML material. For each one, the equation is shown first, then broken into pieces the way you might translate a sentence in a new language piece by piece before restating it fluently.

Example 1: Linear Regression

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \;=\; \theta^{\mathsf{T}} \mathbf{x}$$

Piece by piece. $\hat{y}$ is the model's prediction, as opposed to y, the true answer. Each θ is a parameter, one weight per feature, plus a constant offset θ_0 called the bias. Each x is one input feature. The compact form $\theta^{\mathsf{T}}\mathbf{x}$ is exactly the same sum, written as a dot product instead of listing every term by hand.
In plain English. Add up every feature multiplied by its own importance weight, plus a constant, to get the prediction.

Example 2: Mean Squared Error

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)^2$$

Piece by piece. $J(\theta)$ is the cost, how wrong the model currently is, written as a function of its parameters. The superscript (i) marks the i-th training example, not a power, exactly the trap covered in Section 3.2. $\sum_{i=1}^{m}$ adds this up once for every one of the m training examples. Squaring the difference means errors in either direction count as positive, and large errors get punished more than small ones. Dividing by $2m$ turns the raw total into an average, and the extra 2 exists purely to cancel out neatly later when a derivative is taken.
In plain English. On average, how far off are the predictions, squared so that big mistakes hurt more than small ones.

Example 3: The Sigmoid Function

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

Piece by piece. z can be any real number, positive, negative, huge, or tiny. e^{-z} grows toward infinity when z is very negative, and shrinks toward zero when z is very positive. Taking the reciprocal of one plus that term flips this behavior into a smooth curve that always stays strictly between 0 and 1.
In plain English. Take any number, however extreme, and compress it into something that looks and behaves like a probability.

Example 4: The Softmax Function

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Piece by piece. z_i is the raw, unconstrained score for option i, out of K total options. Exponentiating makes every score positive and exaggerates the gap between large and small scores. The sum in the denominator adds up every exponentiated score, purely so that every option can be divided by the same total.
In plain English. Turn a list of raw scores into a list of probabilities that add up to one, while keeping the same ranking between the options.

Example 5: Cross Entropy Loss

$$L = -\sum_{i} y_i \log(\hat{y}_i)$$

Piece by piece. y_i is the true label for class i, equal to 1 for the correct class and 0 for every other class. $\hat{y}_i$ is the model's predicted probability for class i. The logarithm of a predicted probability is a very negative number when that prediction is close to zero, and close to zero when the prediction is close to one. Negating and summing means a confident correct prediction contributes almost nothing to the loss, while a confident wrong prediction contributes a large penalty.
In plain English. Punish the model in proportion to how confidently wrong it was about the correct answer.

Example 6: The Gradient Descent Update

$$\theta := \theta - \alpha \nabla J(\theta)$$

Piece by piece. This single line is the template underneath almost every learning algorithm in use today. θ is the current set of parameters. The symbol $:=$ means gets updated to, an assignment rather than a claim that both sides were already equal. α is the learning rate, how large a step to take. $\nabla J(\theta)$ is the gradient of the loss, the direction that would increase the loss fastest, so subtracting it moves the parameters downhill instead.
In plain English. Nudge every parameter a small amount in the direction that most decreases the loss, then repeat.

Example 7: Backpropagation and the Chain Rule

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w}$$

Piece by piece. $\frac{\partial L}{\partial w}$ is what we ultimately want, how much the loss would change if this one weight were nudged. Each term on the right is a small, local, easy to compute derivative, how the loss changes as the output changes, how the output changes as this neuron's pre activation changes, and how that pre activation changes as the weight changes. In a real network there are often several such paths, which get added together, but the core idea stays the same.
In plain English. To find how a weight buried deep inside a network affects the final loss, multiply together a chain of small, local effects, one layer at a time.

Example 8: Bayes' Theorem

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Piece by piece. $P(A \mid B)$ is what we want, the updated probability of A now that B is known, called the posterior. $P(B \mid A)$ is how likely B would be if A were true, called the likelihood. $P(A)$ is how probable A was before seeing any evidence, called the prior. $P(B)$ is how probable B is overall, across every possible cause, called the evidence.
In plain English. Start with what you believed before, weigh it by how well it explains what you just observed, and normalize.

Example 9: KL Divergence

$$D_{\mathrm{KL}}(P \,\|\, Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

Piece by piece. P is the true or target distribution, and Q is an approximation of it. For each outcome x, the log ratio between the two probabilities is weighted by how likely that outcome actually is under P, and every outcome's contribution is summed.
In plain English. How many extra bits of surprise you would suffer, on average, by believing Q when P is actually true.

Example 10: Attention, the Core of Transformers

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}}\right) V$$

Piece by piece. Q, K, and V are matrices called queries, keys, and values, one row per token. QK^{T} computes a similarity score between every query and every key. Dividing by $\sqrt{d_k}$, the square root of the key dimension, simply keeps those scores from growing too large as dimension increases. The softmax turns each row of scores into a proper probability

distribution that sums to one. Multiplying by V then takes a weighted average of the value vectors, using those probabilities as weights.
In plain English. For every token, work out how relevant every other token is, turn those relevance scores into weights that sum to one, then blend the other tokens together using those weights.

6 Practice What You Have Learned

Five more equations, all built from pieces already covered above. Try reading each one yourself first, perhaps covering the explanation with a hand or a sheet of paper, before checking your reading against the breakdown underneath.

Practice 1: Ridge Regression Cost

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

This is the mean squared error from Example 2, plus a second term. λ controls how heavily large weights get penalized, and $\sum \theta_j^2$ adds up the squares of the weights. Exact constants in front of the penalty vary between textbooks, what matters is recognizing the pattern itself, an ordinary loss plus a penalty that discourages the model from leaning too heavily on any single weight.

Practice 2: Variance

$$\text{Var}(X) = \mathbb{E}\left[(X - \mu)^2 \right]$$

Variance measures spread. It is defined as the average squared distance between X and its mean μ. Read $\mathbb{E}[\cdot]$ as, take the average of whatever is inside the brackets.

Practice 3: Standardizing a Value

$$z = \frac{x - \mu}{\sigma}$$

Subtract the mean, then divide by the standard deviation. The result tells you how many standard deviations x sits away from the mean, which is why this is often called a z score.

Practice 4: The Length of a Vector

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Square every entry, add the squares up, then take the square root, exactly the Pythagorean theorem generalized from two dimensions to n dimensions. This is the most common meaning of the word norm in machine learning.

Practice 5: Entropy

$$H(p) = -\sum_{x} p(x) \log p(x)$$

Entropy measures how unpredictable a distribution is. A distribution concentrated on one outcome has low entropy, while a distribution spread evenly across many outcomes has high entropy. Notice the family resemblance to cross entropy in Example 5 and KL divergence in Example 9, all three share this same log probability pattern.

7 Quick Reference Cheat Sheet

Everything above, condensed onto two pages for fast lookup. This is the page to return to mid paper.

Greek Letters

α	learning rate, coefficient, significance level	ν	degrees of freedom
β	regression coefficient, Adam decay rate	ξ	slack variable; noise
γ, Γ	RL discount factor; gamma function	o	rare (Big O uses Latin O, not this)
δ, Δ	small change; a finite change	π, Π	pi; RL policy; product operator
ϵ	tiny constant; exploration rate	ρ	correlation coefficient
ζ	rare, occasional noise term	σ, Σ	std deviation, sigmoid; sum, covariance
η	learning rate	τ	time constant; temperature
θ, Θ	model parameters; tight complexity bound	υ	almost never used
ι	almost never used	ϕ, Φ	feature map; normal CDF
κ	condition number; Cohen's kappa	χ	chi squared distribution and test
λ	regularization strength; eigenvalue	ψ	feature map (alternative)
μ	mean of a distribution	ω, Ω	weights; complexity lower bound, sample space

Operators and Symbols

$\approx$	approximately equal	$A^\top$	transpose
$:=$	defined as, assigned as	A^{-1}	matrix inverse
$\propto$	proportional to	$\cdot$	dot product
$\in$	is an element of	$\odot$	element wise product
$\forall$	for all	$\|x\|$	length of a vector
$\exists$	there exists	$\mathbb{E}[X]$	expected value, average
$\mathbb{R}^n$	vectors of n real numbers	$P(A \mid B)$	probability of A given B
∂	partial derivative	$X \sim \mathcal{N}$	is distributed as
∇	gradient, vector of partial derivatives	$\arg\max$	the input that gives the largest output
$\int$	integral, continuous sum	$O(n^2)$	cost grows roughly like n^2

Grammar

- Subscript x_i: which one, an index
- Superscript $x^{(i)}$ with parentheses: which example, not a power
- Superscript x^2 with nothing wrapped around it: an actual power
- Hat $\hat{y}$: an estimate or prediction

- Bar $\bar{x}$: an average
- Tilde over one symbol $\tilde{\theta}$: a modified version
- Bold lowercase: vector. Bold uppercase: matrix. Plain lowercase: a single number

Closing Notes: How to Keep Reading Without Panicking

A few honest habits, on top of everything above.

Read the sentence surrounding an equation before reading the equation itself. Authors almost always describe in words what a formula computes, right before or right after showing it, and that sentence is often the fastest way in.

Find the outermost operation first. If an equation is wrapped in a summation or an expectation, that outer layer tells you the overall shape of what is happening before you worry about the details inside it.

Ask what type each symbol is. A bold capital letter behaves completely differently from a lowercase italic letter, even sitting in the exact same position in two different equations, so knowing whether something is a scalar, a vector, or a matrix changes what an operation actually does.

Expect to reread. Dense equations get reread constantly, including by the researchers who wrote them. Needing a second or third pass is not a sign that you are behind, it is simply how equations get read.

Finally, keep this guide nearby rather than trying to memorize it in one sitting. Come back to Section 7 whenever an unfamiliar symbol shows up, the same way you would look up an unfamiliar word rather than learning a dictionary cover to cover before starting to read.