How Linear Algebra and Probability Distributions Concepts Are Utilized in Various ML Models

Carl Lochstampfor Jr

Department of Cybersecurity, Old Dominion University,

CYSE 420 — Applied Machine Learning in Cybersecurity

January 25, 2026

Overview of Key Concepts

Linear Algebra Concepts

Vectors and Matrices: A vector is simply an ordered list of numbers, like [3, 7, 2]. In ML, vectors usually represent a single data sample (e.g., features of a network connection like packet size, duration, and port number). A matrix is a 2D grid of numbers with rows and columns. Matrices are commonly used to represent whole datasets (rows are samples, columns are features) or the weights connecting layers in a neural network.

Tensors: Tensors are the generalization of vectors and matrices to any number of dimensions. A scalar (single number) is a 0D tensor, a vector is 1D, a matrix is 2D, and higher dimensions follow the same pattern. For example, processing 32 images at once might use a 4D tensor with dimensions for batch size, height, width, and color channels.

Key Operations: The dot product multiplies corresponding elements of two vectors and sums the results — this is the core computation happening inside neural networks. Matrix multiplication extends this idea to compute many dot products at once. Norms provide a way to measure how "big" a vector is: the L2 norm is the straight-line distance from the origin, while the L1 norm sums the absolute values. Both come up when measuring prediction errors and regularizing models.

Probability Distributions

Bernoulli Distribution: This distribution models binary outcomes (e.g., yes/no, malicious/benign, and similar scenarios). It is defined by a single parameter p, the probability of "success." In cybersecurity, it naturally fits classification problems, such as detecting whether traffic is an attack.

Gaussian (Normal) Distribution: The familiar bell curve, defined by mean (center) and variance (spread). Most values fall near the average, with fewer values appearing farther away from it. Many ML algorithms assume Gaussian-distributed data or errors because it simplifies the math and often matches real-world patterns.

Model Analysis: How ML Models Use These Concepts

Neural Networks

Neural networks follow a similar representation, with each layer performing a matrix multiplication: output = weights × input + bias. A nonlinearity is then applied to the output using element-wise operations. The weight matrix holds learned parameters that convert input features to effective representations. For a layer with 100 inputs and 50 outputs, the weight matrix is of size 50×100, and a single matrix multiplication produces all 50 outputs at once. The final layer in classification tasks applies a sigmoid function (for binary classification) or a softmax function (for multi-class classification) to the raw output scores, converting them to probabilities (outputting a probability reminiscent of a Bernoulli distribution for each class).

Linear Regression

Linear regression models output as a linear combination of inputs: $\hat{y} = w_1x_1 + w_2x_2 + ... + b$. In matrix notation, this is written as $\hat{y} = Xw$, where X is the data matrix with each row being a training example, and w is the weight vector. The assumption is that prediction errors are normally distributed with zero mean, which ties in nicely with the squared error loss function. Regularization introduces norm penalties—L2 regularization (the squared L2 norm of weights) discourages any particular weight from growing too large.

Conclusion

Linear algebra provides the data structures (vectors, matrices, tensors) and operations (dot products, matrix multiplication) that enable ML to be computationally feasible. Probability distributions enable models to handle uncertainty and output meaningful confidence scores rather than just hard predictions. Together, these mathematical foundations enable ML systems to learn from data and make informed decisions in applications such as cybersecurity threat detection.

References

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press. https://d2l.ai