AI for product managers:
Today's top terms to stay in the know

Progress in AI is exploding – the AI industry's value is projected to grow by over 13x over the next eight years. Now, product-led growth (PLG) companies are racing to take advantage of this high-value opportunity by embedding AI into key features that improve user experience, capture new markets, and drive adoption.

Jasper, for example, is a PLG company that uses AI to generate automated marketing copy. The company recently raised a raised a $125 million Series A funding round at a $1.5 billion valuation. Google recently bought an AI avatar startup for $100 million to help it better compete in social sharing trends. More startups and enterprises alike are following suit.

As adoption of AI has increased, the language used to discuss AI has also evolved to reflect new progress in the space. From Large Language Models to transformers to GPUs, terminology is popping up on Twitter (see: Large Language Models and Reinforcement Learning and Generative AI Application Landscape infographic), in news stories (see: New York Times and VentureBeat), and in business cases that describe some of these latest developments in AI.

Gaining a basic understanding of this newer AI terminology is critical for any field that wants to capitalize on significant opportunity, including product management.

This guide will serve as an entry point into the **current key AI terminology to know now**:

Large Language Models

Language Models are mathematical models that give the probability of a certain sequence of words occurring in a given language. *Large* Language Models are very big and trained on huge amounts of data, therefore encapsulating significant amounts of information about a language. Large language models are widely used in applications relevant to natural language, powering things like text-prediction.

Transformers

Transformers are a type of Deep Learning model that are widely used in state-of-the-art applications. Their main benefit is that they are able to distribute their computations across many machines, making the complicated computations required for e.g. automatic speech recognition feasible to do in a reasonable amount of time.

Reinforcement Learning

Reinforcement Learning (RL) is an area of Machine Learning inspired by real-world environments where learning happens by rewarding desired behaviors and punishing undesired ones. It first gained popularity when computers learned to play different video games through RL approaches, but it's now a powerful technique that's widely used in many Deep Learning fields.

Generative Models

Generative Models are a subset of AI Models that can be used to generate data that is similar to a set of training data. For example, if a well-performing generative model is trained on a dataset of human faces, then it can be used to generate entirely novel images of new human faces. Generative Models have become popular in recent years and include DALLE-2, Imagen, Stable Diffusion, and Poisson Flow Generative Models (PFGMs).

Classification

A Machine Learning task which seeks to classify data points into different groups (called targets or class labels) that are pre-determined by the training data. For example, if we have a medical dataset consisting of biological measurements (heart rate, body temperature, age, height, weight, etc.) and whether or not a person has a specific disease, we could train a classification model to predict whether or not a person has the disease given just the biological measurements. This could be helpful for very expensive or time intensive disease measurement.

Regression

A supervised learning task that tries to predict a numerical result given a data point. For example, giving the description of a house (location, number of rooms, energy label) and predicting the market price of the house.

Underfitting

A phenomenon in which a Machine Learning algorithm is *not fitted well enough* to the training data, resulting in low performance on both the training data and similar but distinct data. A common example of underfitting occurs when a neural network is not trained long enough or when there is not enough training data. The converse phenomenon is *overfitting*.

Overfitting

A phenomenon in which a Machine Learning algorithm is *too* fitted to the training data, making performance on the training data very high, but performance on similar but distinct data low due to poor generalizability. A common example of overfitting occurs when a neural network is trained for too long. The converse phenomenon is *underfitting*.

Validation data

A subset of data that a model is *not* trained on but is used during training to verify that the model performs well on distinct data. Validation data is used for hyperparameter tuning in order to avoid overfitting.

Cost function

This is what Machine Learning algorithms are trying to minimize to achieve the best performance. It is simply the error the algorithm makes over a given dataset. It is also sometimes referred to as “loss function”.

Parameter

Generally refers to the numbers in a neural network or Machine Learning algorithm that are changed to alter how the model behaves (sometimes also called weights). If a neural network is analogous to a radio, providing the base structure of a system, then parameters are analogous to the knobs on the radio, which are tuned to achieve a specific behavior (like tuning in to a specific frequency). Parameters are not set by the creator of the model, rather, they are values that are determined by the training process automatically.

Hyperparameter

A value that takes part in defining the overall structure of a model or behavior of an algorithm. Hyperparameters are **not** altered by the model training process and are set ahead of time before training. Many potential values for hyperparameters are generally tested to find those that optimize the training process. For example, in a neural network, the *number* of layers is a hyperparameter (not altered by training), whereas the values within the layers (“weights”) themselves are *parameters* (altered by training). If the model is a radio, then a hyperparameter would be the number of knobs on the radio, while the *values* of these knobs would be parameters.

Loss function

A (generally continuous) value that is a computation-friendly proxy for the performance metric. It measures the error between values predicted by the model and the *true* values we *want* the model to predict. During training, this value is minimized. “Loss function” is sometimes used interchangeably with “cost function,” although the two are differentiated in some contexts.

Neural network

A specific type of Machine Learning algorithm which can be represented graphically as a network, inspired by the way that biological brains work. The network represents many simple

mathematical operations (addition, multiplication, etc.) that are combined to produce a complex operation that may perform a complicated task (e.g. identifying cars in an image).

Transfer learning

Transfer learning is a Machine Learning technique where a trained model for one task is reused as the starting point for a new model on a different task. The new model can then utilize the already existing knowledge and often needs less training. It is a powerful technique that is widely used and often yields good results.

Accelerators

Accelerators are specialized computer hardware that are useful for AI and can accelerate the computation speed of models. Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs) are different types of accelerators.

GPUs

A graphics processing unit (GPU) is a computer chip designed to perform rapid mathematical calculations. Traditionally, GPUs are responsible for rendering graphics and images, although today, they have a wider use range. With the emergence of Deep Learning, the importance of GPUs has increased. Training of deep neural networks can be more than 100 times faster with GPUs than with CPUs (Central Processing Units).