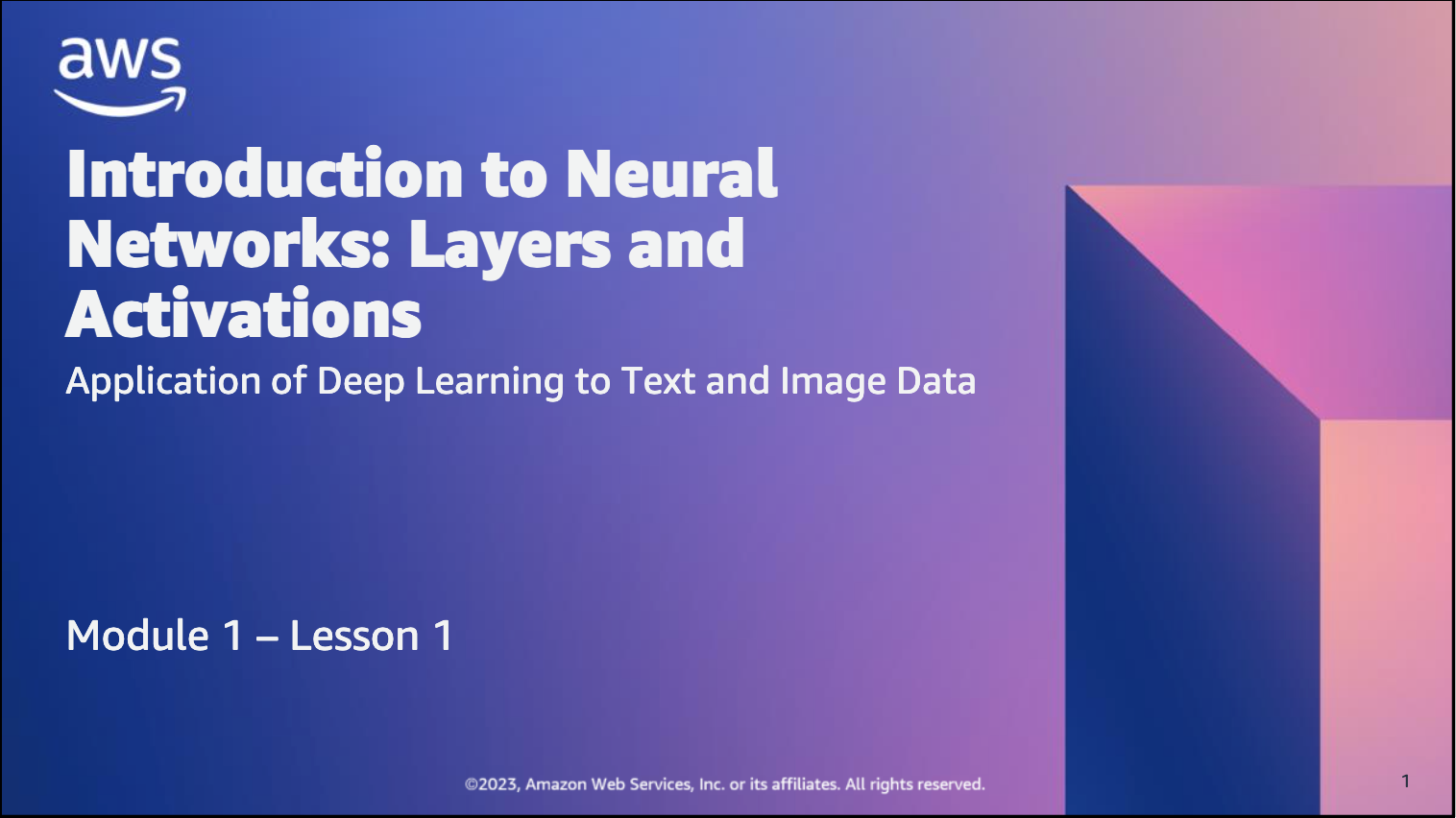
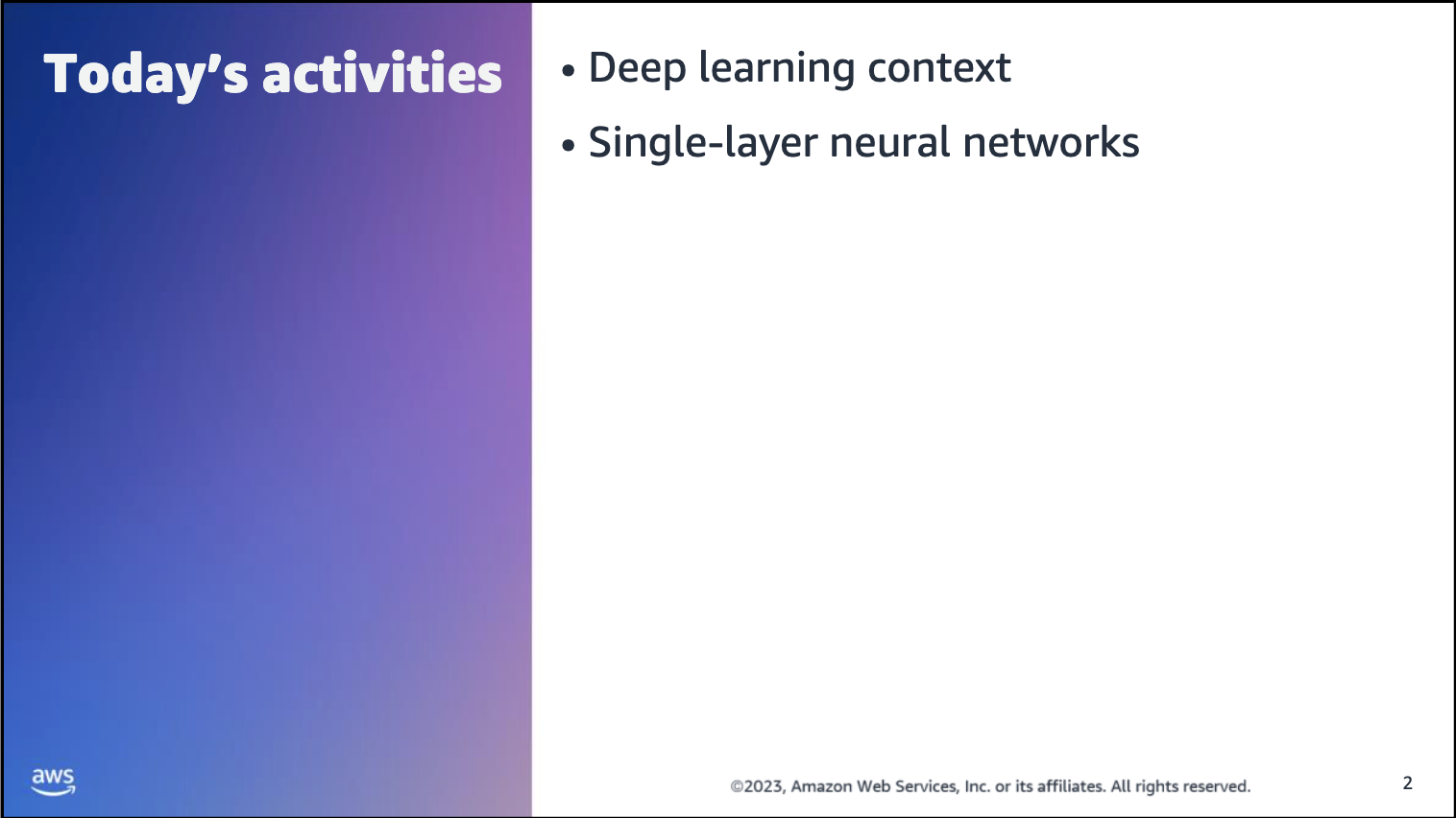
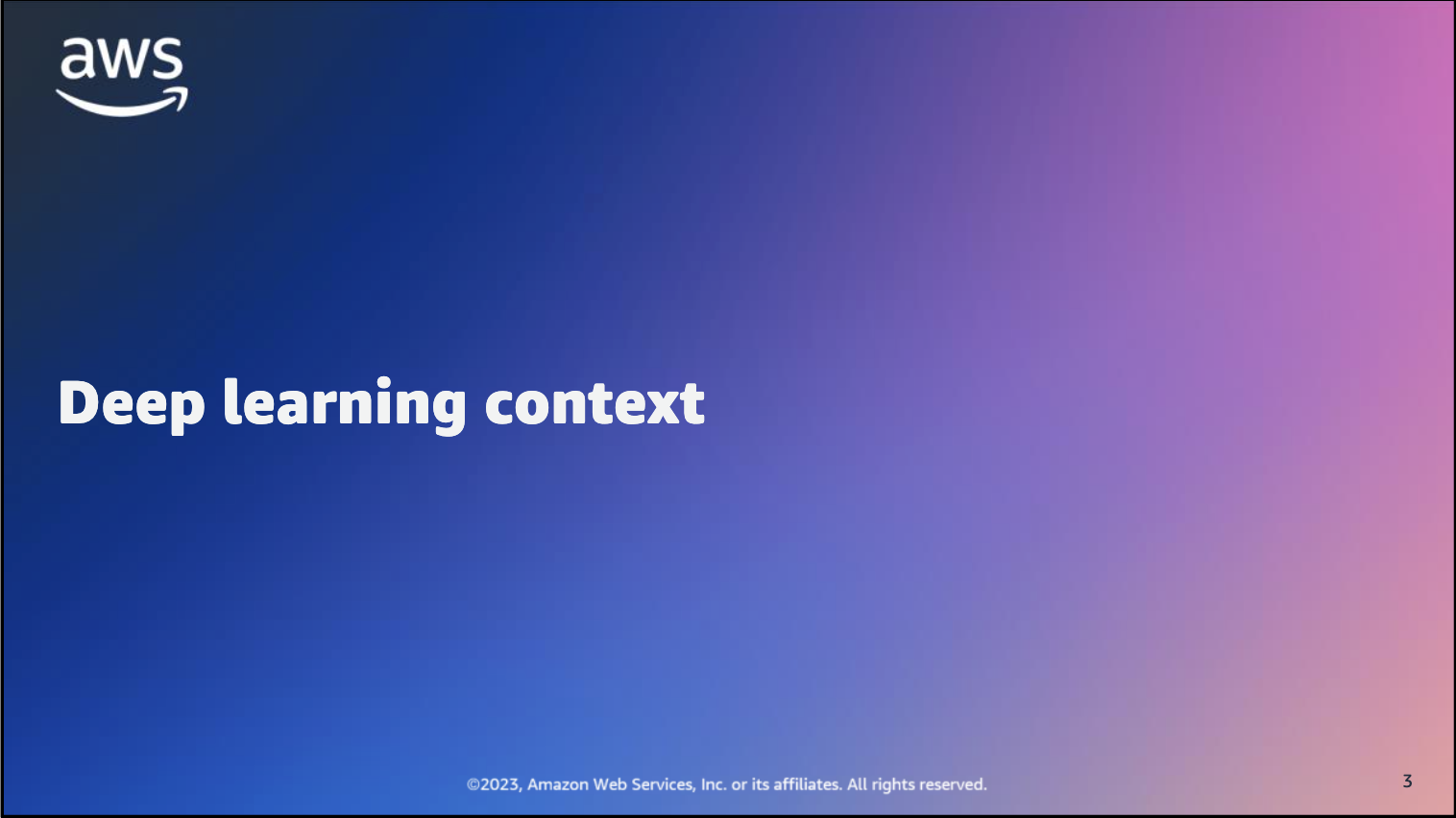
****

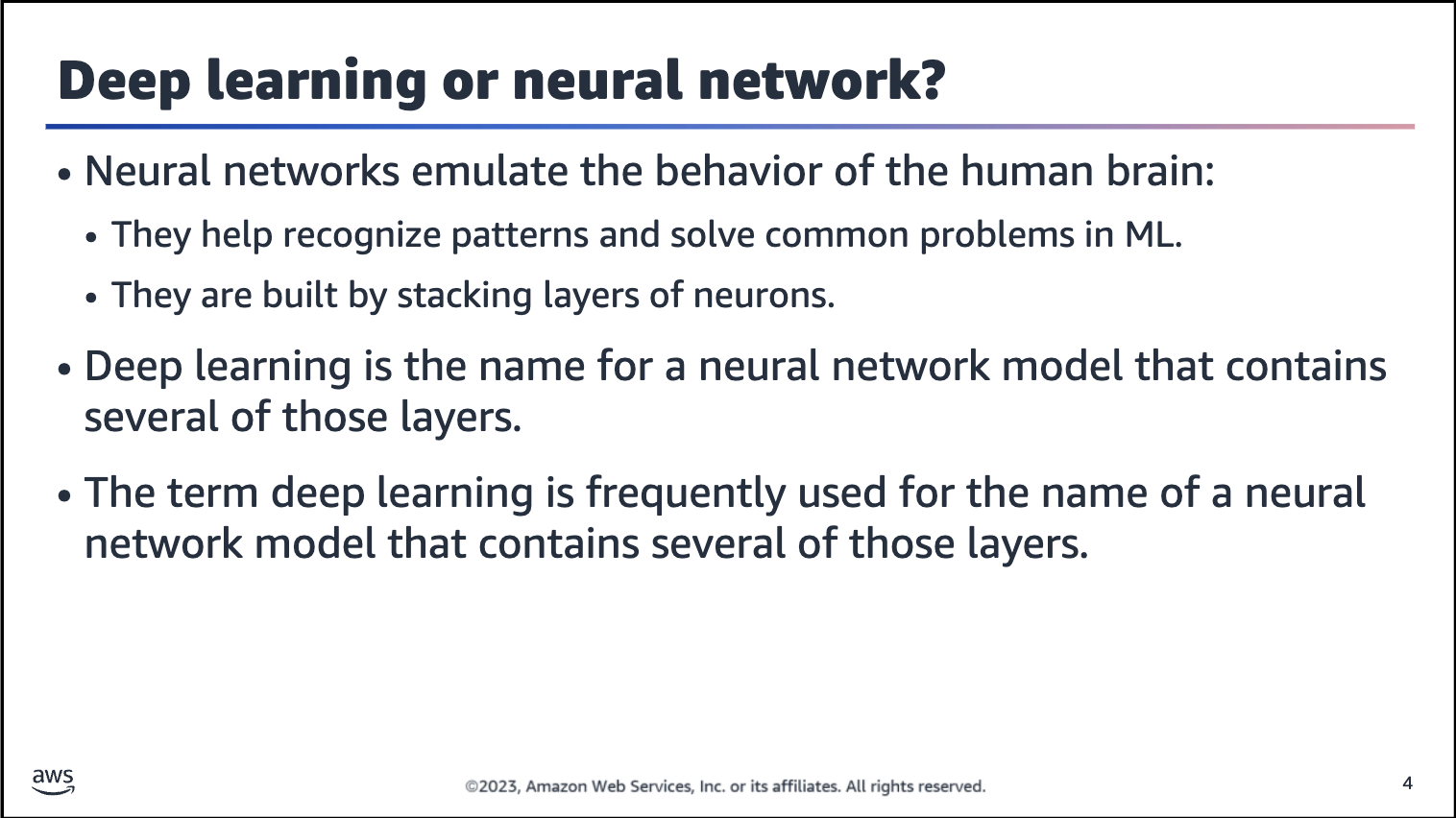
**Application of Deep Learning to Text and Image Data**

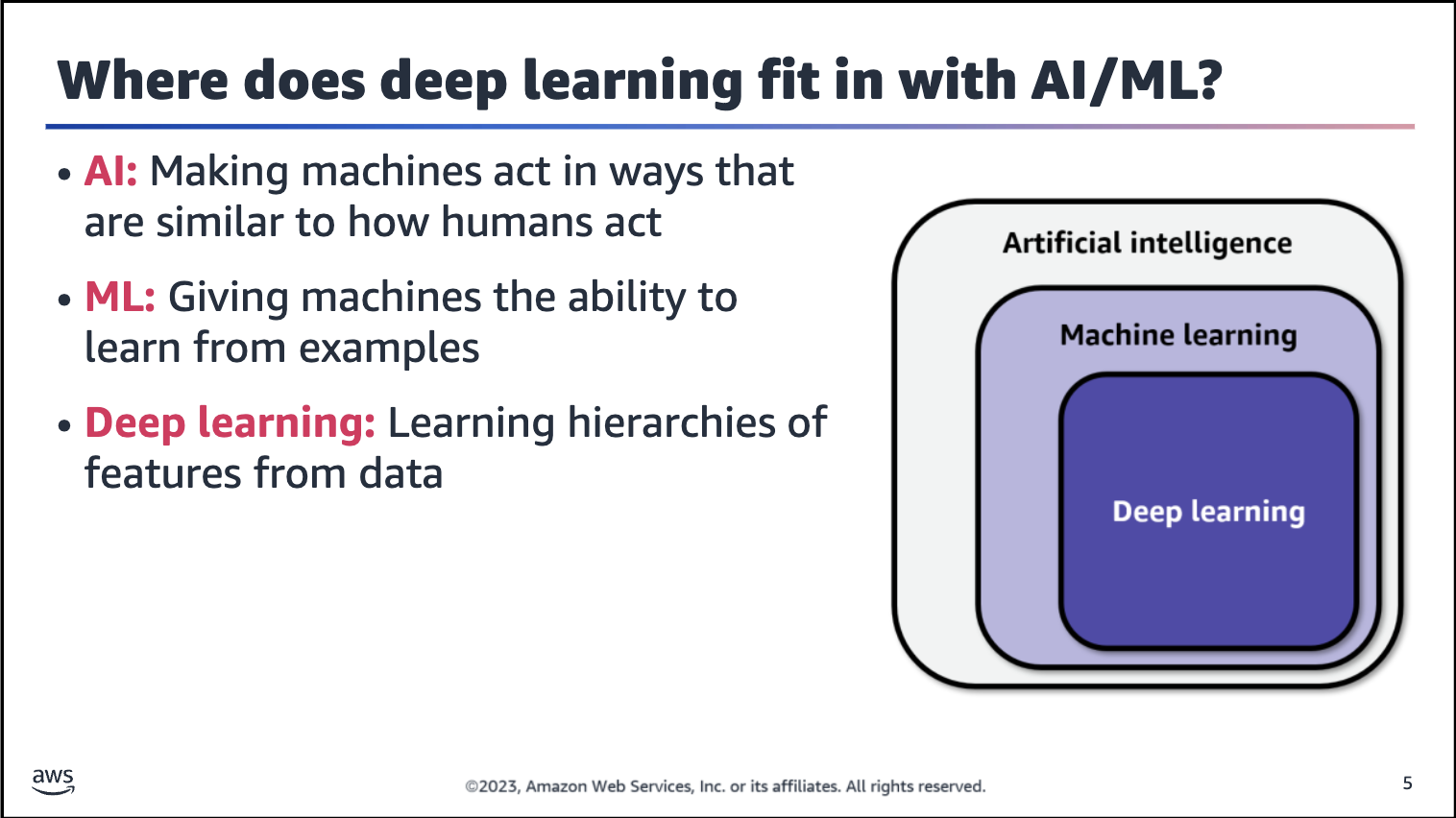
Module 1 Lesson 1 Student Guide













Deep learning aims to learn ML models by constructing hierarchies of features (a deep representation), rather than shallow ML models that are built on hand-constructed features. These features emerge as relevant to predictions from the data

and are not built by hand.

Classical ML refers to models that don’t use deep learning architectures. With classical ML, you need to manually prepare and manipulate the features (perform feature engineering) before you can offer the dataset to train the model.

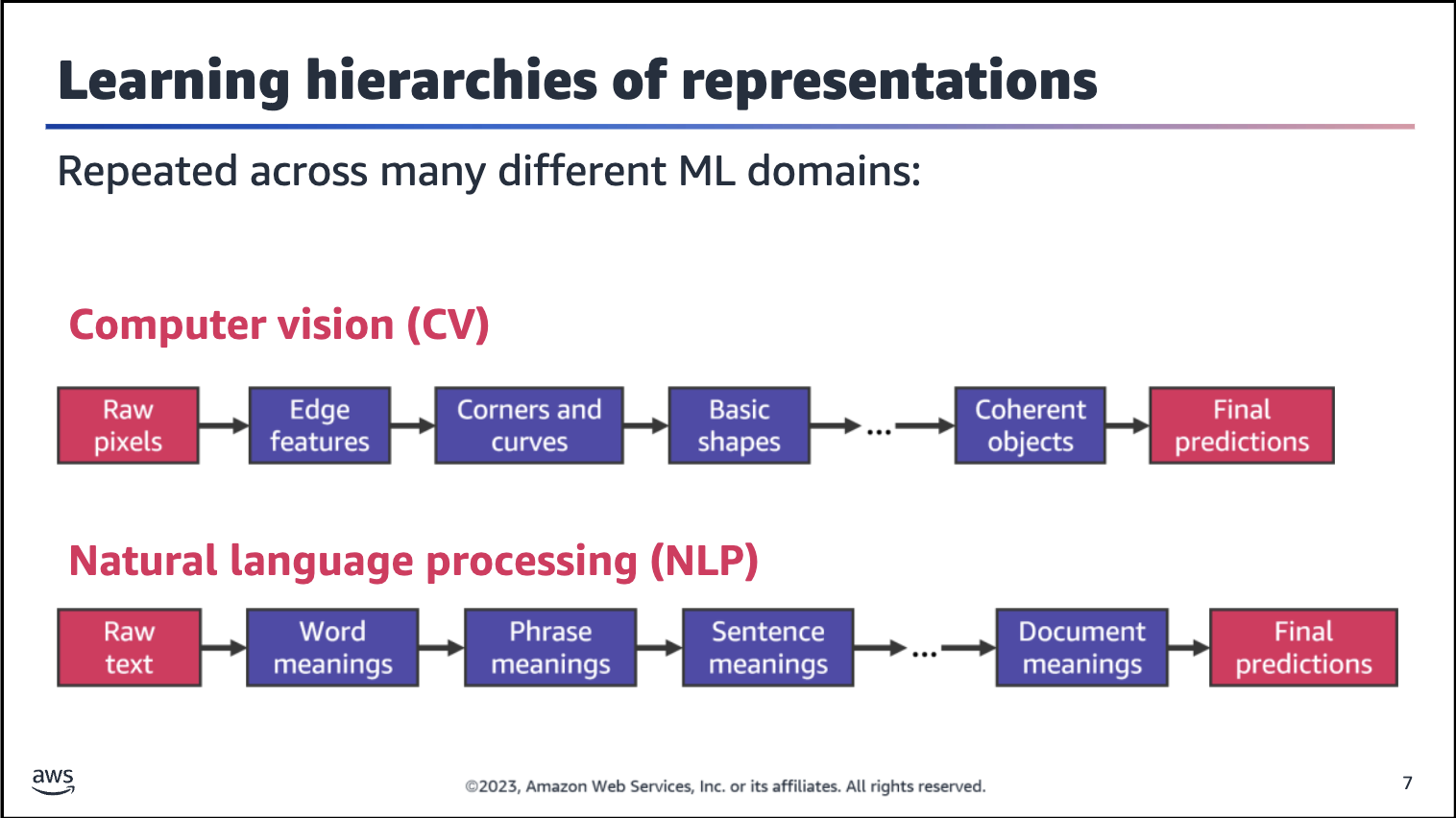
While classical ML learns how to predict classes or numbers from labeled datasets, deep learning models can learn the features, which means that the model can learn internal representations from the features. The term hierarchies of features is related to the fact that different layers in a deep learning model can learn different representations from the features in the dataset. The hierarchy is related to which type of representation each layer learns.

The representation concept here is not intuitive and can be better understood via examples.

Examples of representations include the following:

• In natural language processing (NLP) model words, simple combinations of characters can represent real and abstract entities, such as cat, dog, and love.

• In computer vision (CV) model images, simple combinations of pixels can represent real objects, such as cats, cars, and people.



The concept of learning hierarchies of representations can be repeated across many different domains of ML.

**CV example**

In this example, the first box on the left represents the input data (raw pixels) to the deep learning model. The last box on the right represents the output or prediction from the model (for example, the objects to predict in the image, such as cats or dogs).

The intermediate boxes represent the different layers of the deep learning model. The first layer of the model specializes in learning about edges in the image data. The second layer specializes in learning corners and curves, the next layer learns basic shapes, and so on. The last layer learns how to identify coherent objects.

The hierarchy of representation is expressed by how each layer specializes to a particular level of abstraction of the representation. The hierarchy starts with low-level representations in the image, such as edges and corners, and moves through

higher-level representations until the last layer, in which the model can learn to detect coherent objects. This is the hierarchy that the previous slide referred to.

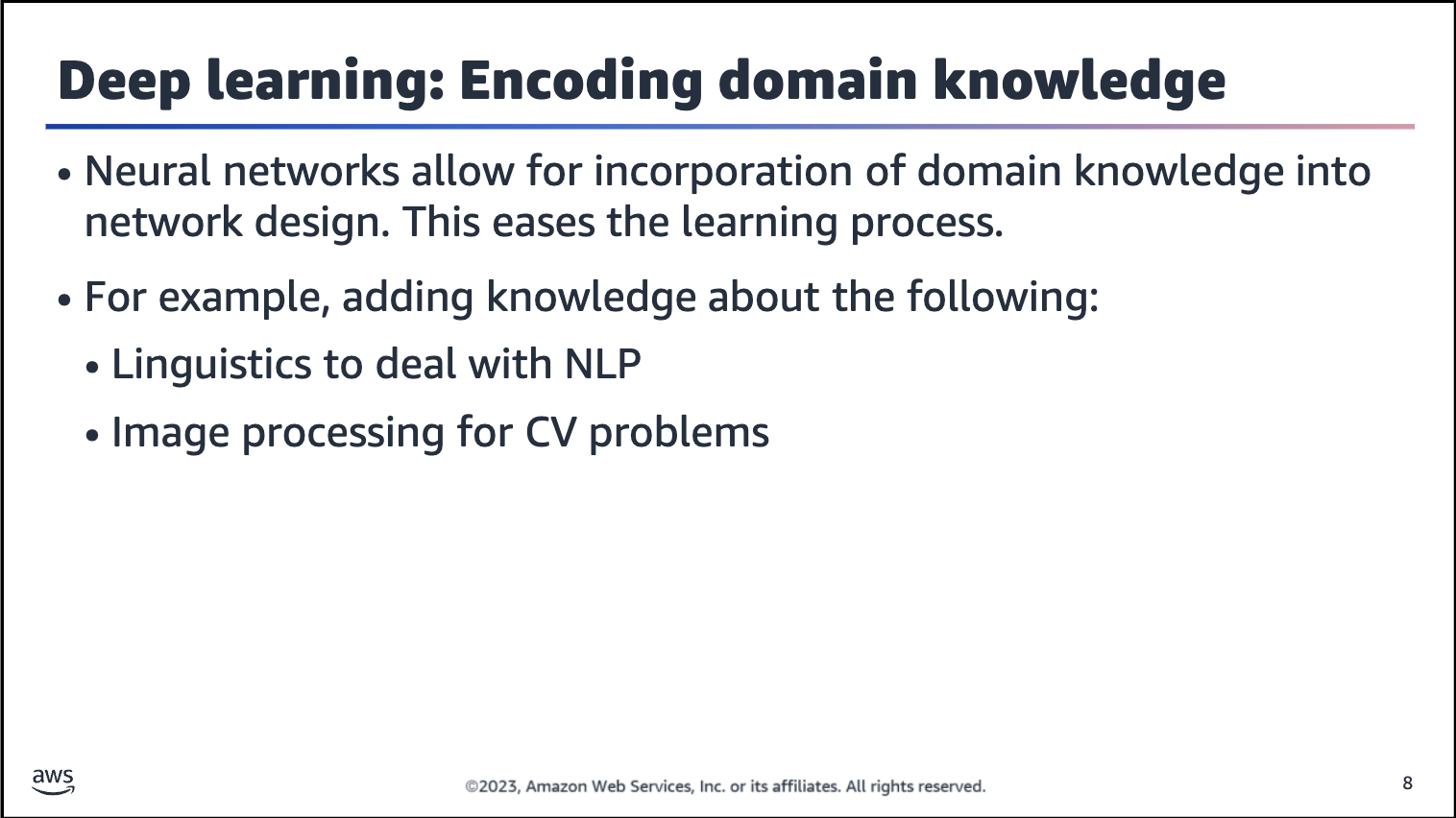
**NLP example**

In this example, the first box on the left represents the input data (for example, text from product reviews) to the deep learning model. Again, the last box on the right represents the output or prediction from the model (for example, whether the sentiment expressed in a review is positive or negative).

The intermediate boxes represent the different layers of the deep learning model. The first layer of the model specializes in learning about word meanings in the text data. The second layer specializes in learning phrase meanings, the next layer learns

sentence meanings, and so on. The last layer learns how to identify full document meanings.

For this example, the hierarchy of representation is also expressed by how each layer specializes to a particular level of abstraction of the representation. The hierarchy starts with low-level representations in the text, such as word and phrases meanings, and moves through higher level representations until the last layer, in which the model can learn full document meanings.



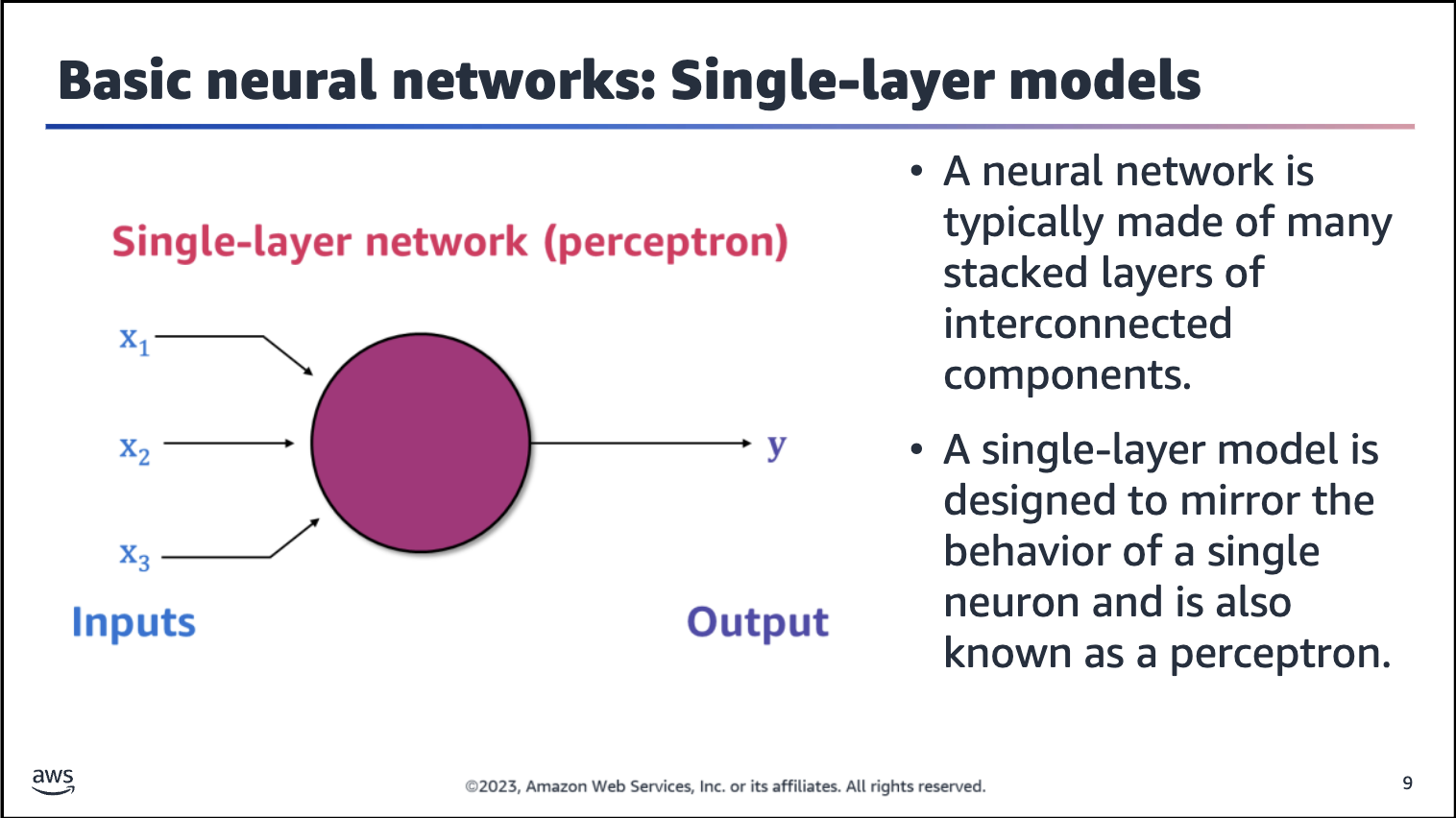
Neural networks are distinguished in the world of ML by the fact that you can engineer them rather easily.

Deep learning models allow for the incorporation of domain knowledge into the network design itself, which makes the learning process easier (use the network architecture as a form of prior). For example, you can bring knowledge about linguistics to handle a natural language process or bring knowledge about image processing for CV-related problems.

Particular deep learning architectures can be fitted to incorporate domain characteristics, such as a recurrent neural network (RNN) for the NLP domain or a convolutional neural network (CNN) for the CV domain.

For more information about image priors, see Deep Image Prior at

<https://dmitryulyanov.github.io/deep_image_prior>.



As seen previously, neural networks are typically made of many stacked layers or interconnected components. However, let’s start small to understand the basic theory with a single-layer network.

In terms of biological motivation, this type of model is designed to mirror the behavior of a single neuron (although it’s now known that biological neurons are quite sophisticated).

For more information about biological neurons, see the following resources:

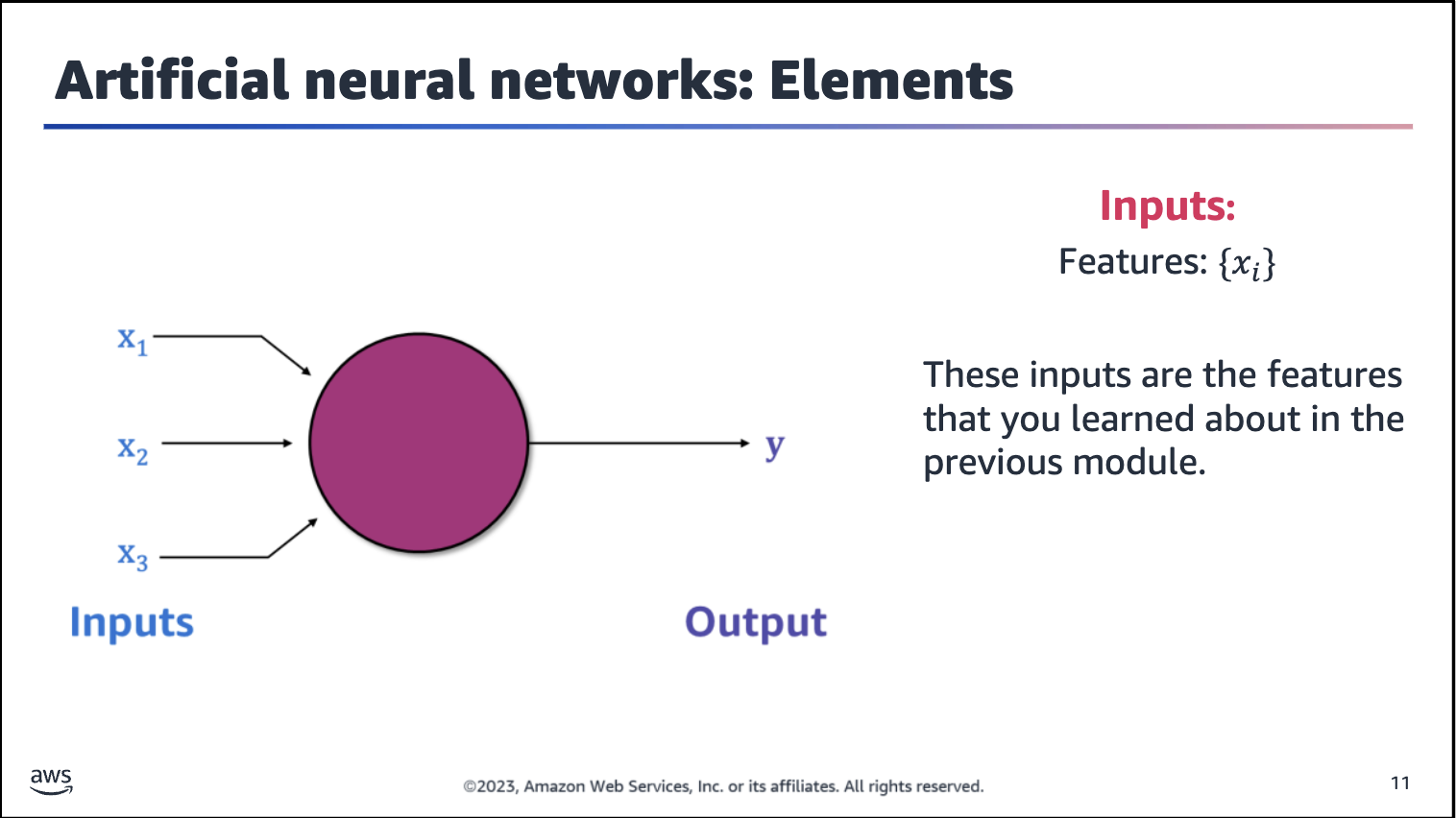
• “Dendritic Action Potentials and Computation in Human Layer 2/3 Cortical Neurons” in Science, Vol. 367, No. 6473 at

<https://doi.org/10.1126/science.aax6239>

• “Single Cortical Neurons as Deep Artificial Neural Networks” in Neuron, Vol. 109, No. 17 at

<https://doi.org/10.1016/j.neuron.2021.07.002>

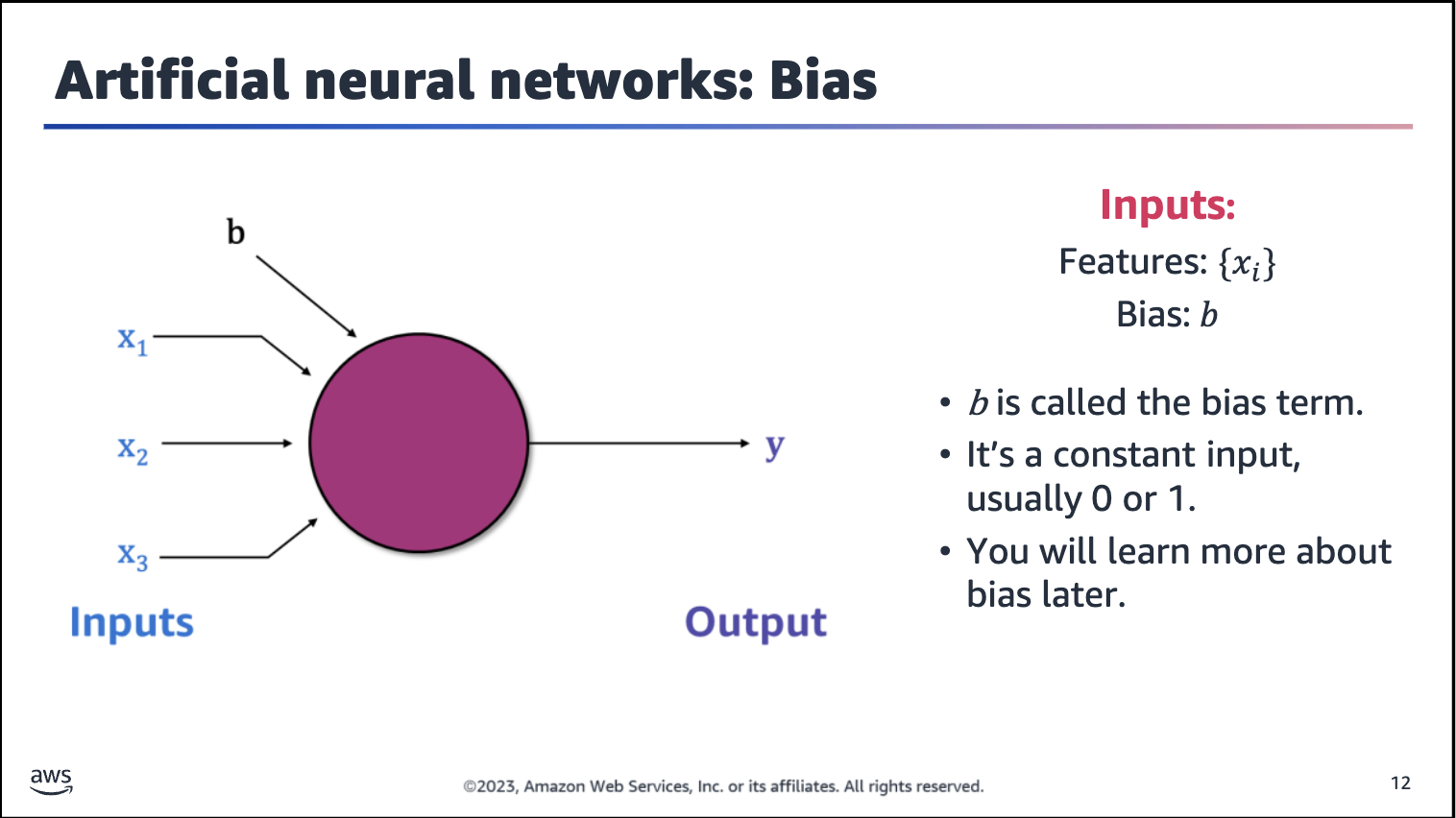


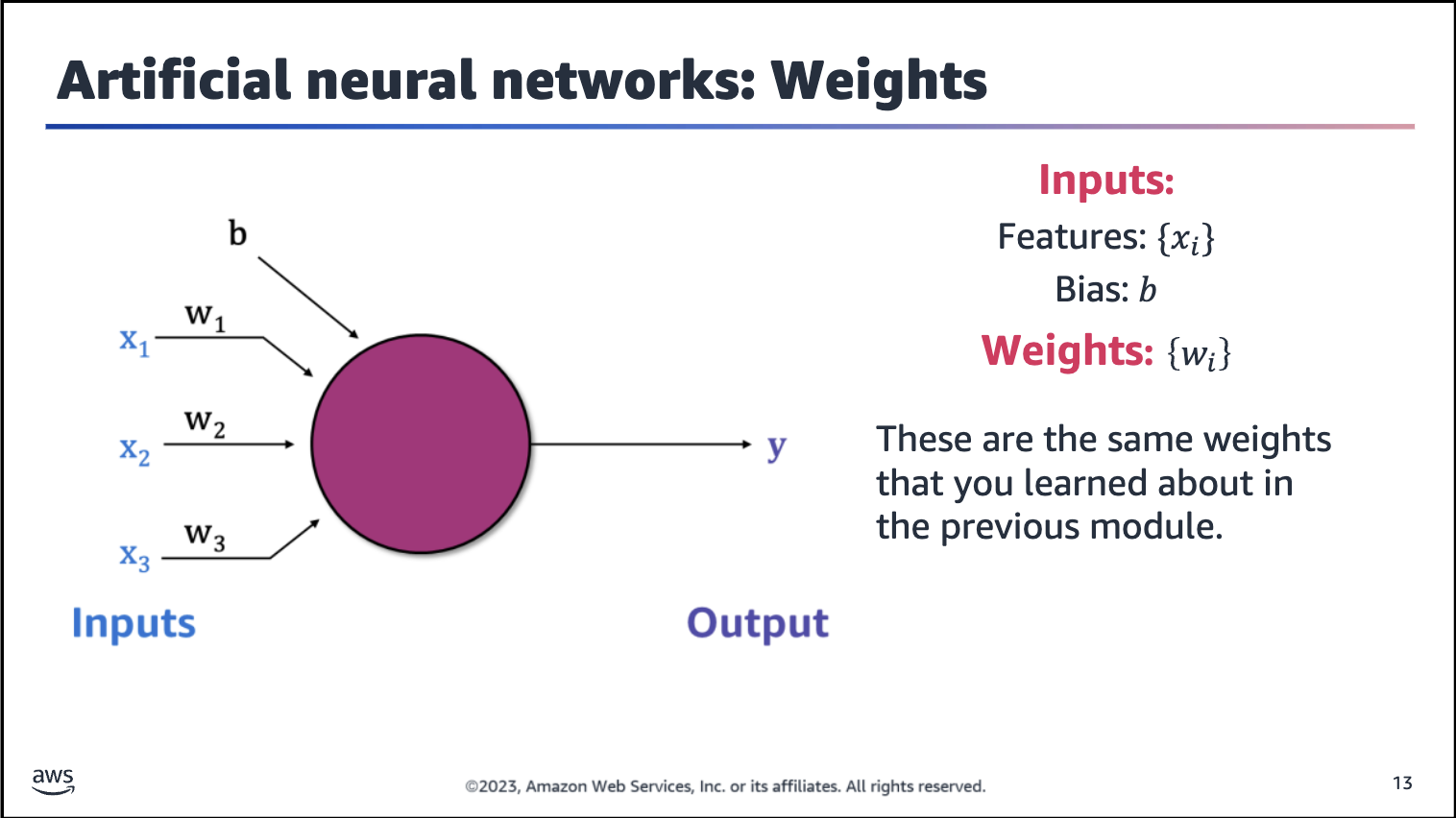


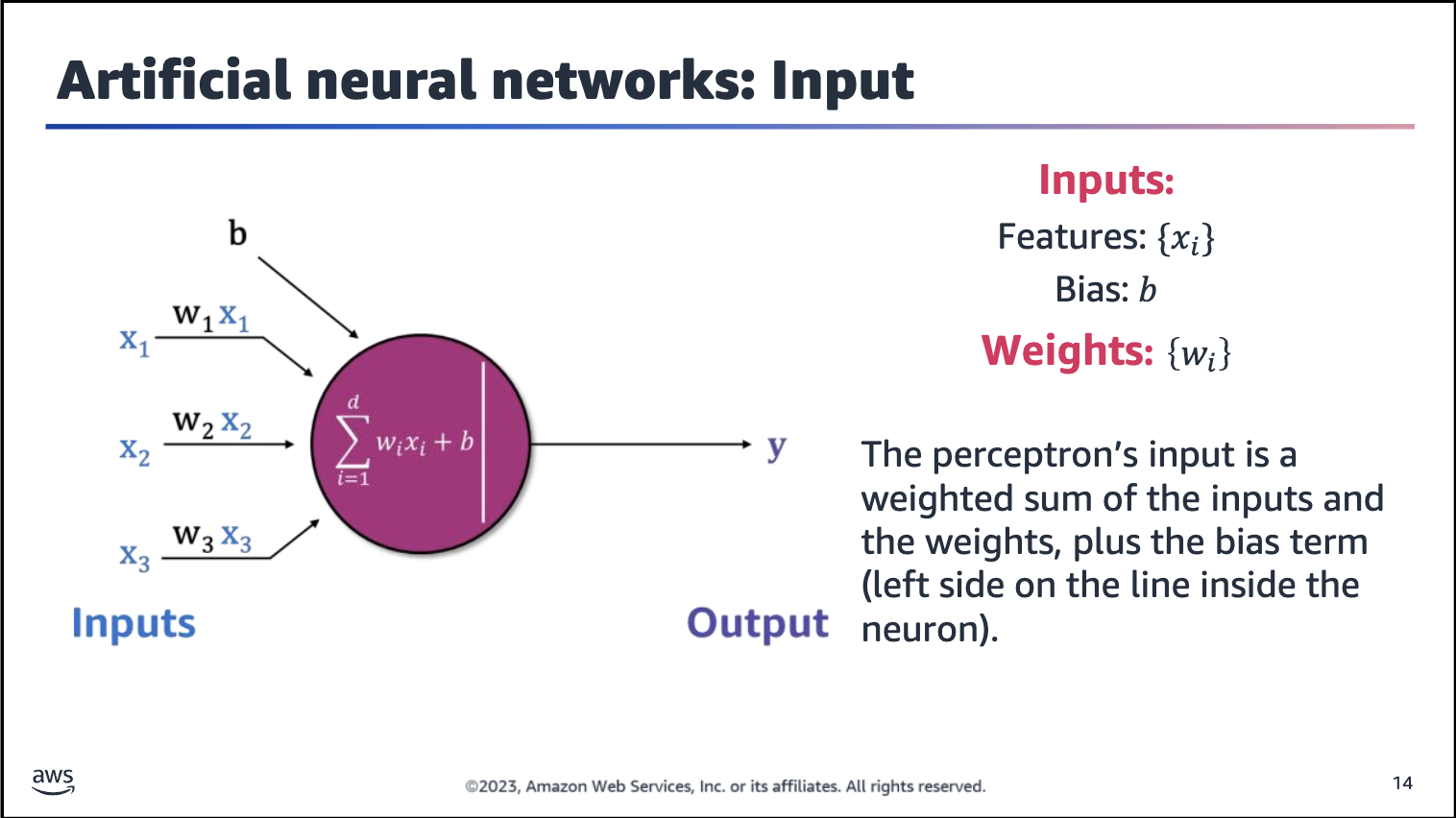
Loss functions quantify the distance between the real and predicted values of the target. An optimization problem seeks to minimize a loss function.

An artificial neuron (also sometimes referred to as a

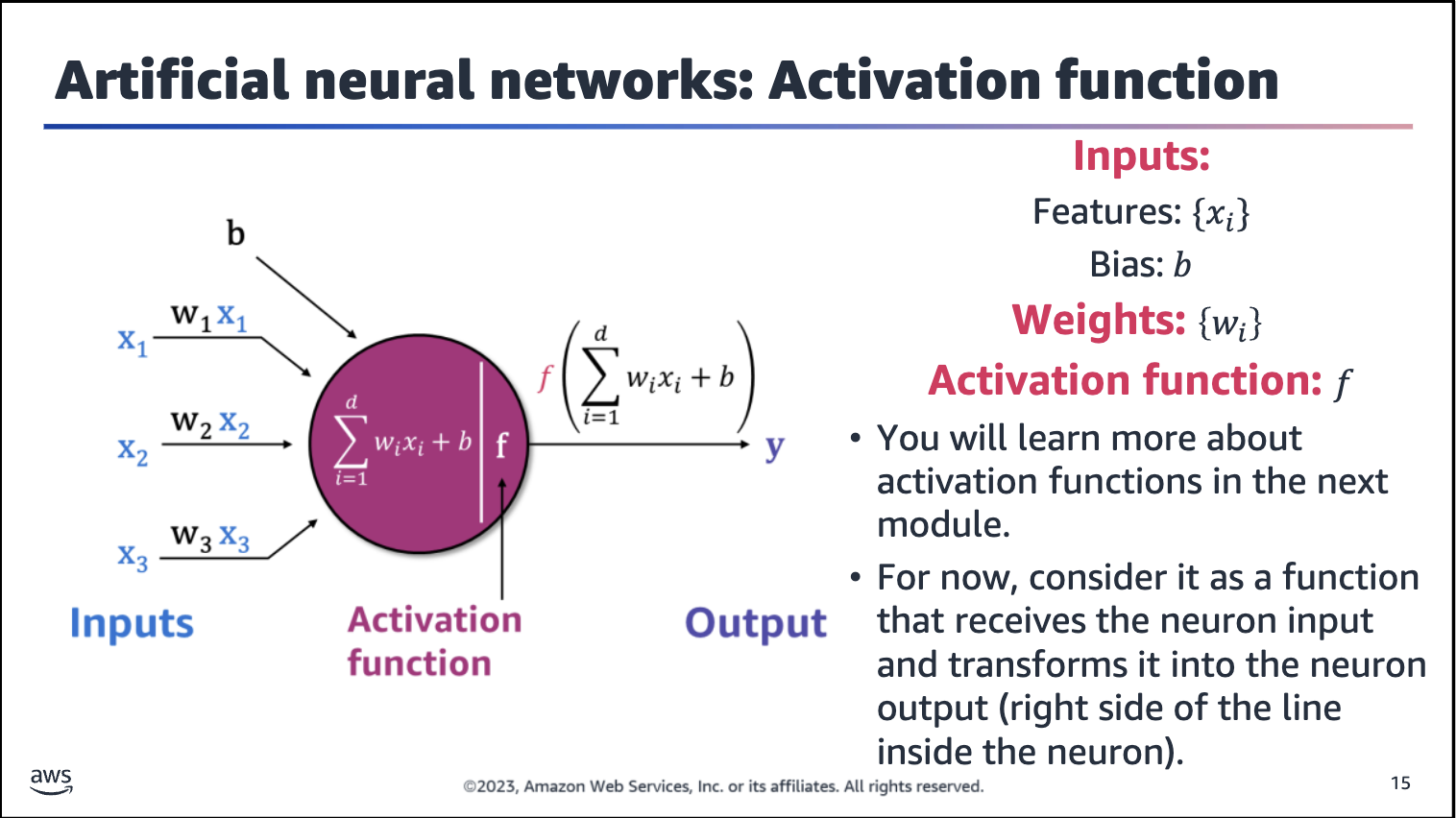
perceptron) is the fundamental unit of an artificial neural network. A typical artificial neuron consists of input values, biases that are applied to those inputs, and an output value.

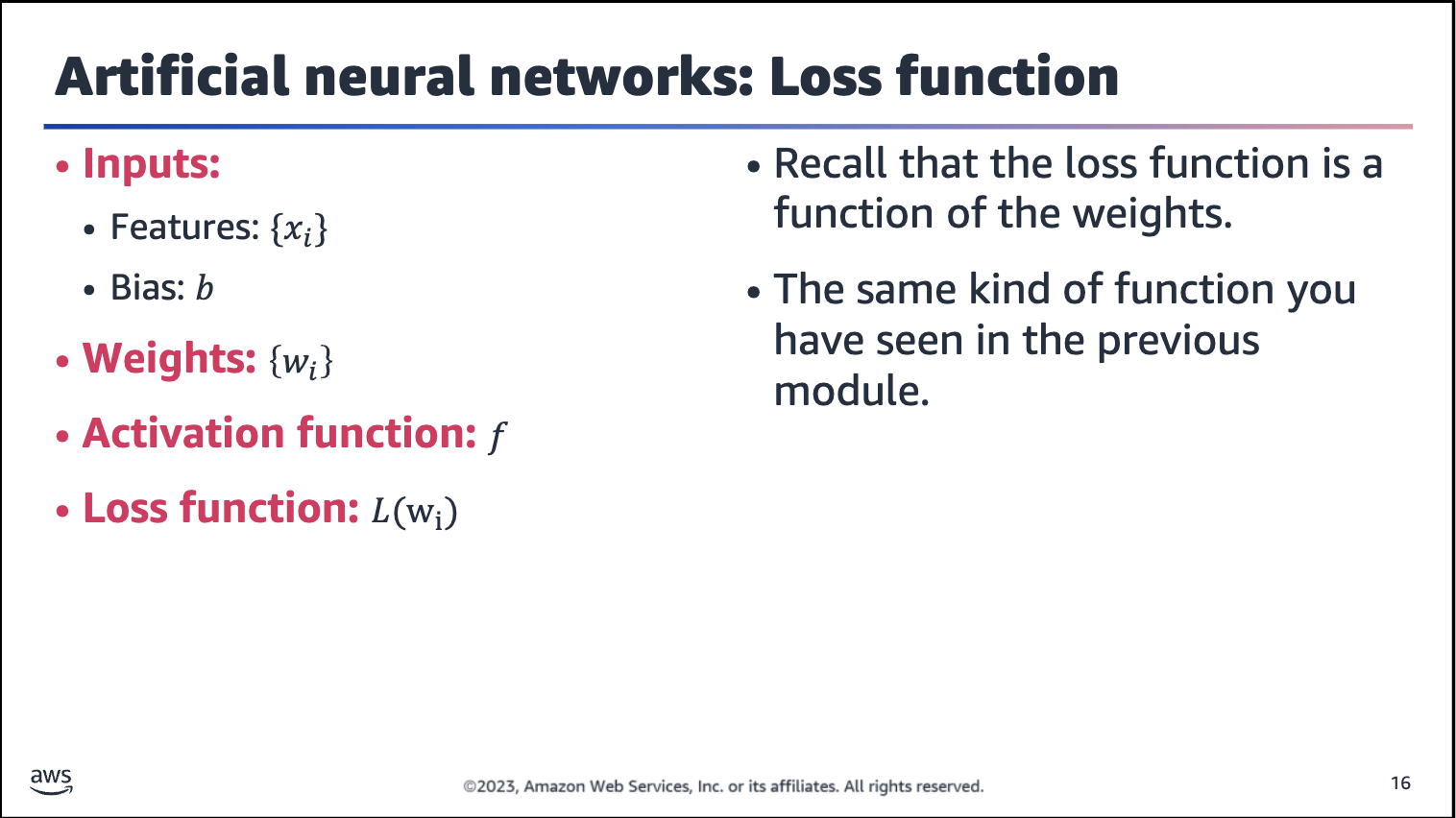


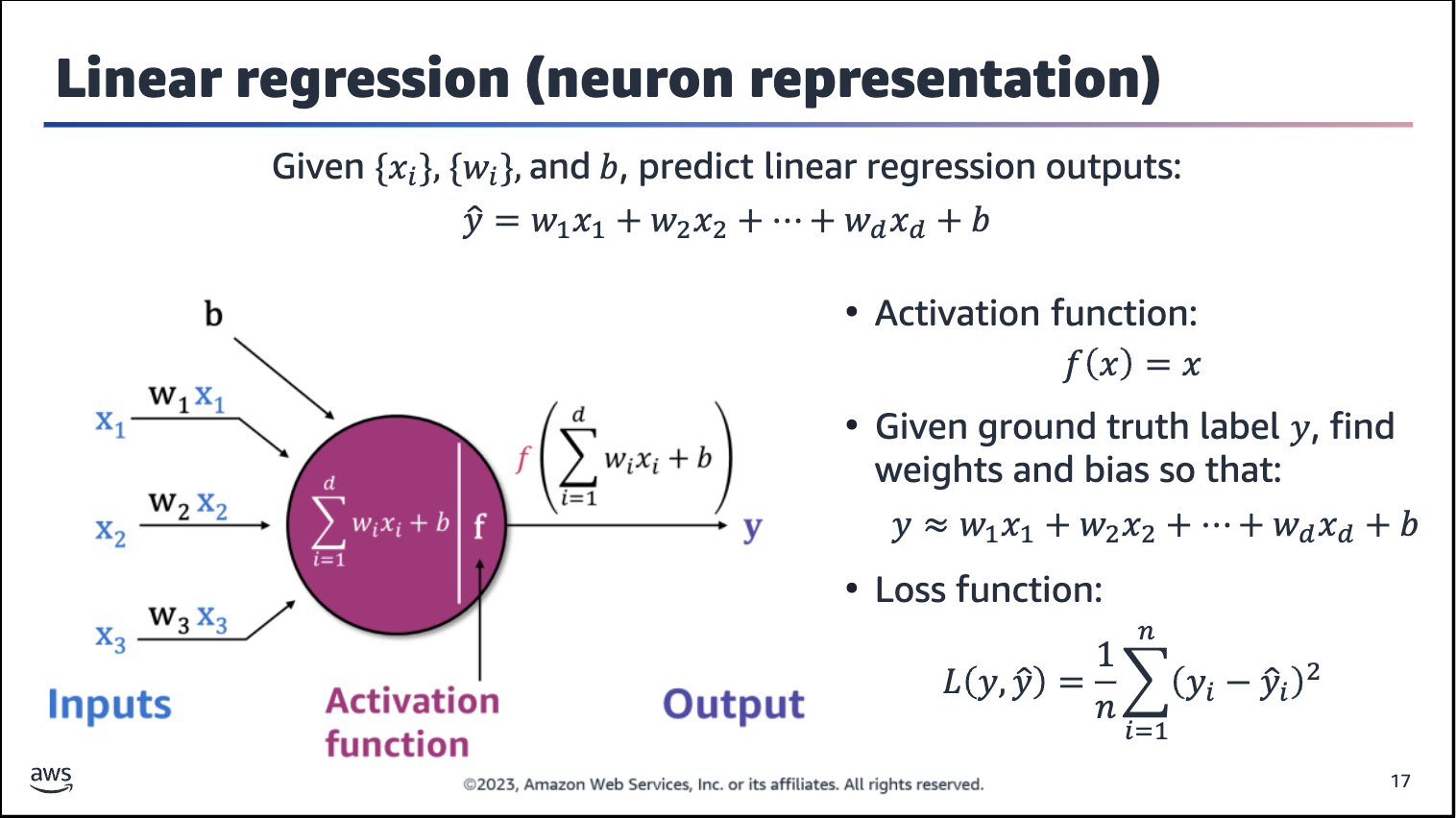




The line in the neuron (circle) is separates the input from the output.







Linear regression: Pass the weighted sum to the neuron unit, and then apply the activation function to generate output, y.

Dimension, which is the number of input features, is represented as d (in x*d* and w*d*)



