

# Output Layers and Loss Functions

Course 3, Module 3, Lesson 2

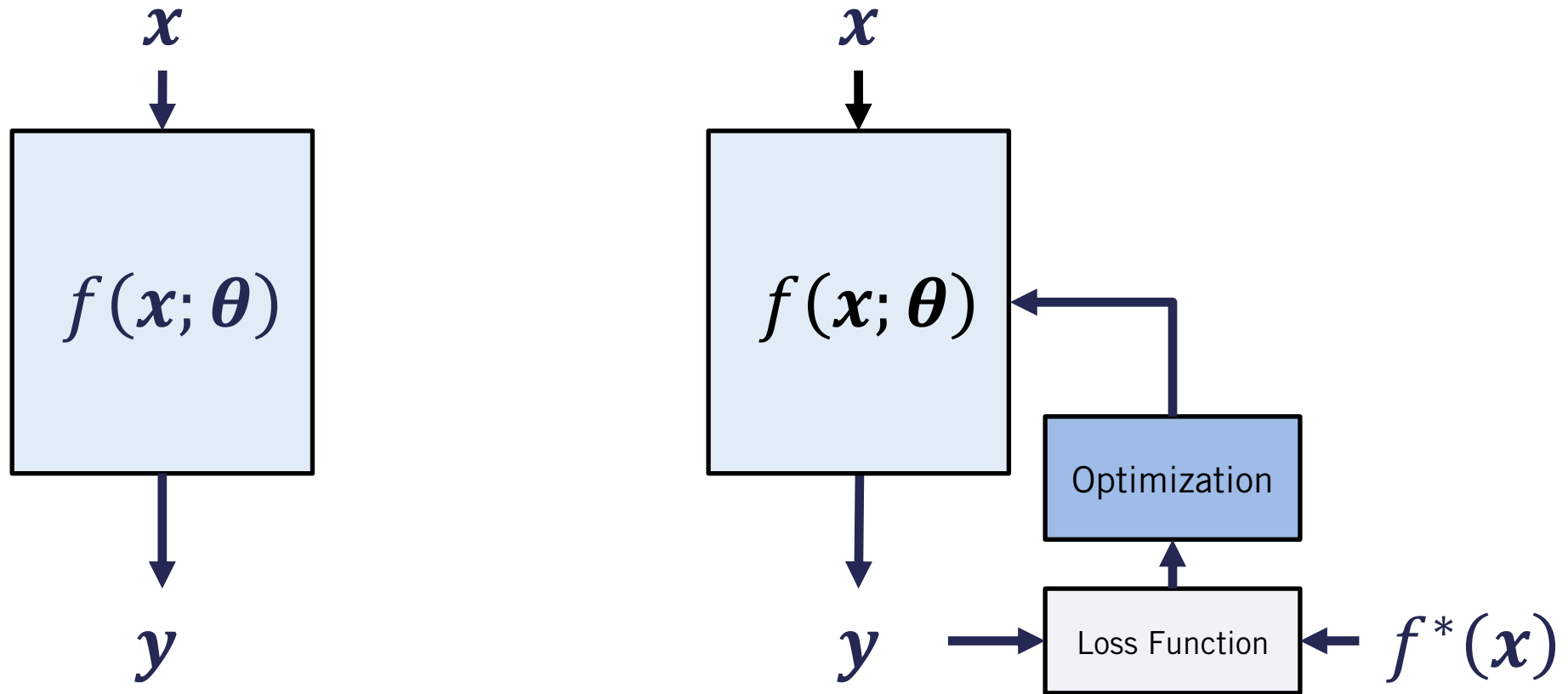


UNIVERSITY OF TORONTO  
FACULTY OF APPLIED SCIENCE & ENGINEERING

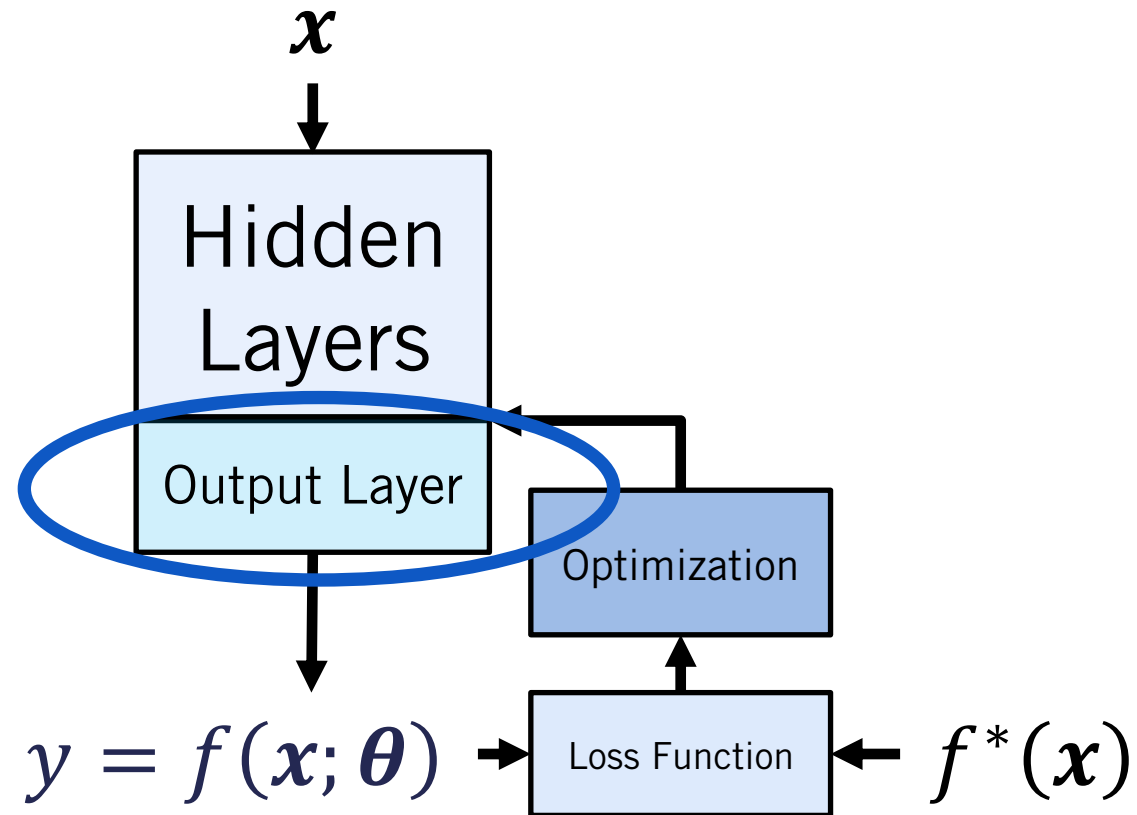
# Learning Objectives

- Learn the general process of designing machine learning algorithm, and extend it to the design of neural networks
- Learn different types of neural network **loss functions** that can be used depending on the type of task at hand

# Machine Learning Algorithm Design

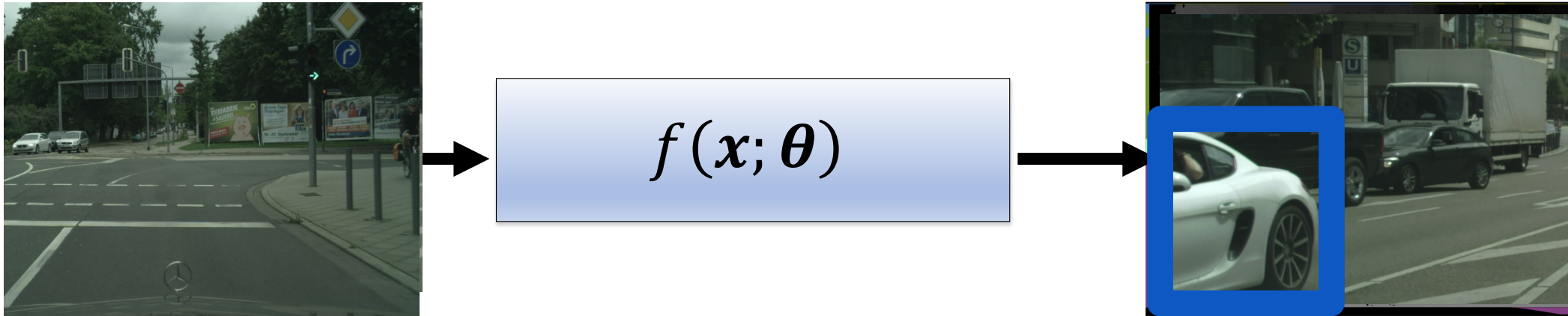


# Artificial Neural Networks



# Tasks: Classification and Regression

- **Classification:** Given input  $x$  map it to one of  $k$  classes or categories.
  - Image classification, semantic segmentation
- **Regression:** Given input  $x$  map it to a real number
  - Depth prediction, bounding box estimation



# Classification: Softmax Output Layers

- **Softmax output layers** are most often used as the output of a classifier, to represent the **probability distribution** over  $K$  different classes
- The Softmax output layer is comprised of:

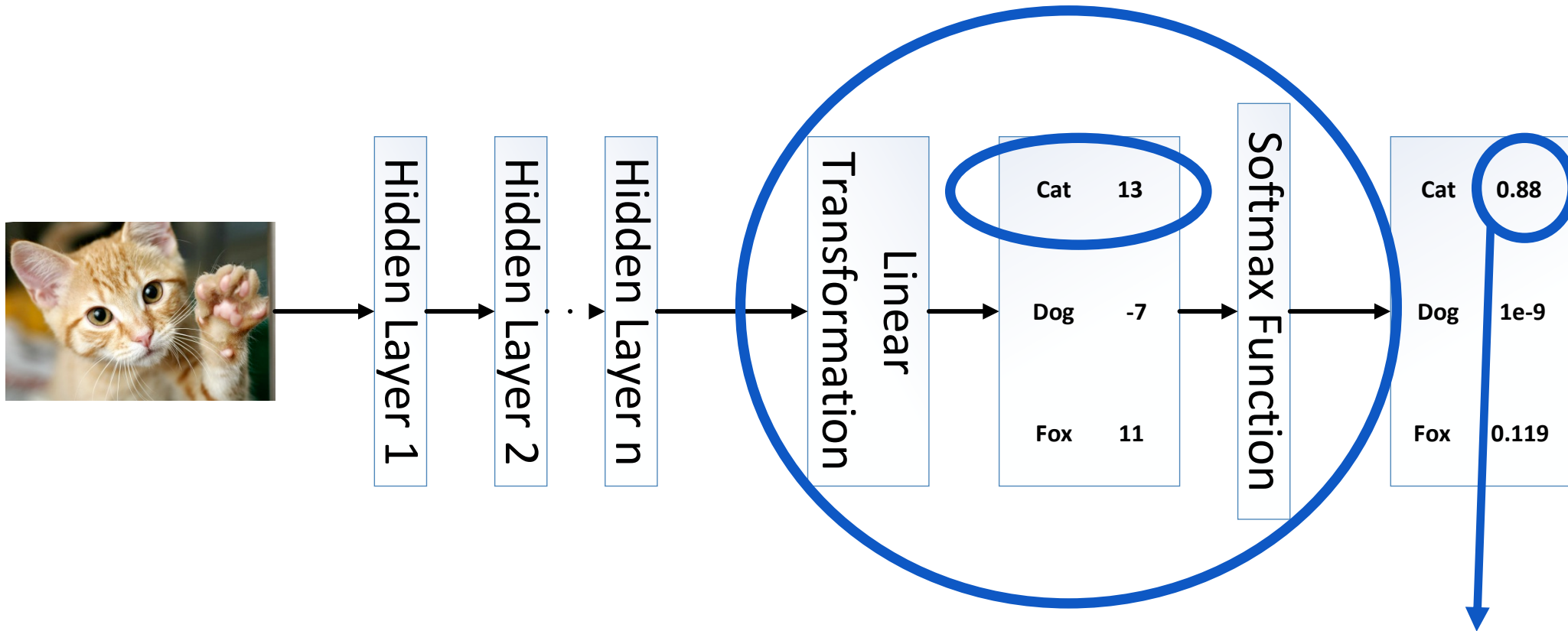
- A linear transformation:

$$z = W^T h + b$$

- Followed by the **Softmax** function:

$$\text{Softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

# Classification: Softmax Output Layers



$$\text{Softmax}(h_{cat}) = \frac{\exp(13)}{\exp(13) + \exp(-7) + \exp(11)} = 0.88$$

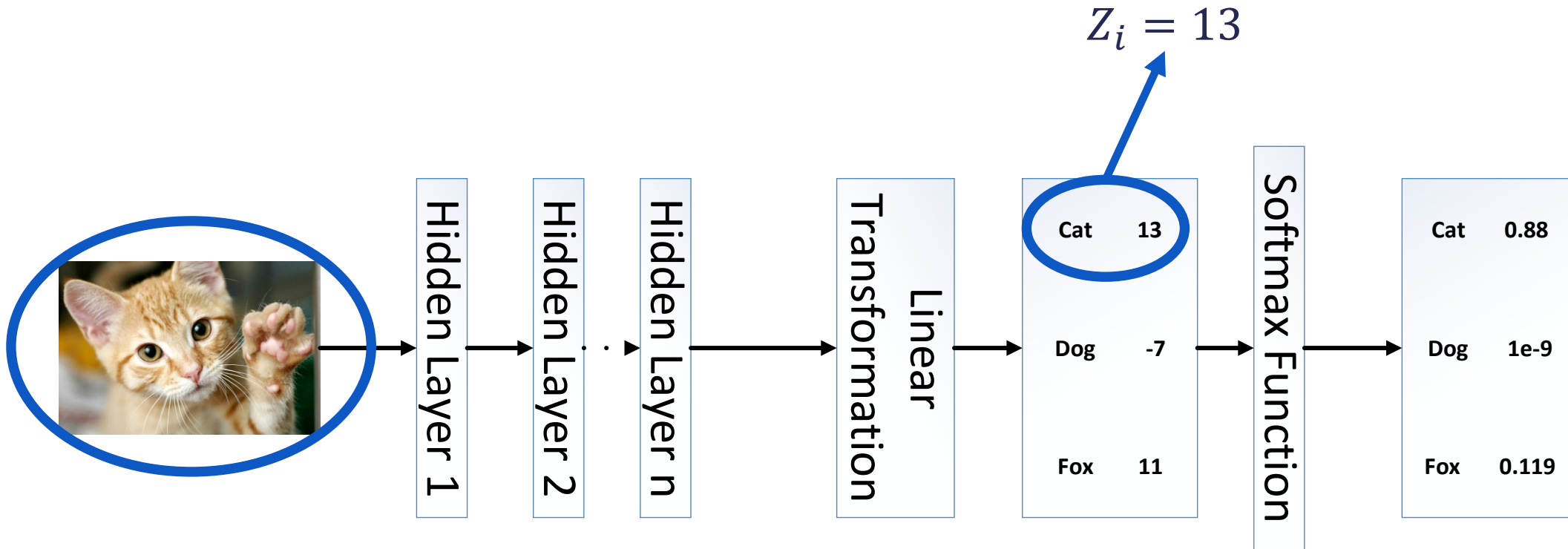
# Classification: Cross-Entropy Loss Function

- By considering the output of the softmax output layer as a probability distribution, the **Cross Entropy Loss** function is derived using **maximum likelihood** as:

$$L(\theta) = -\log(\text{Softmax}(z_i)) = -z_i + \log \sum_j \exp(z_j)$$

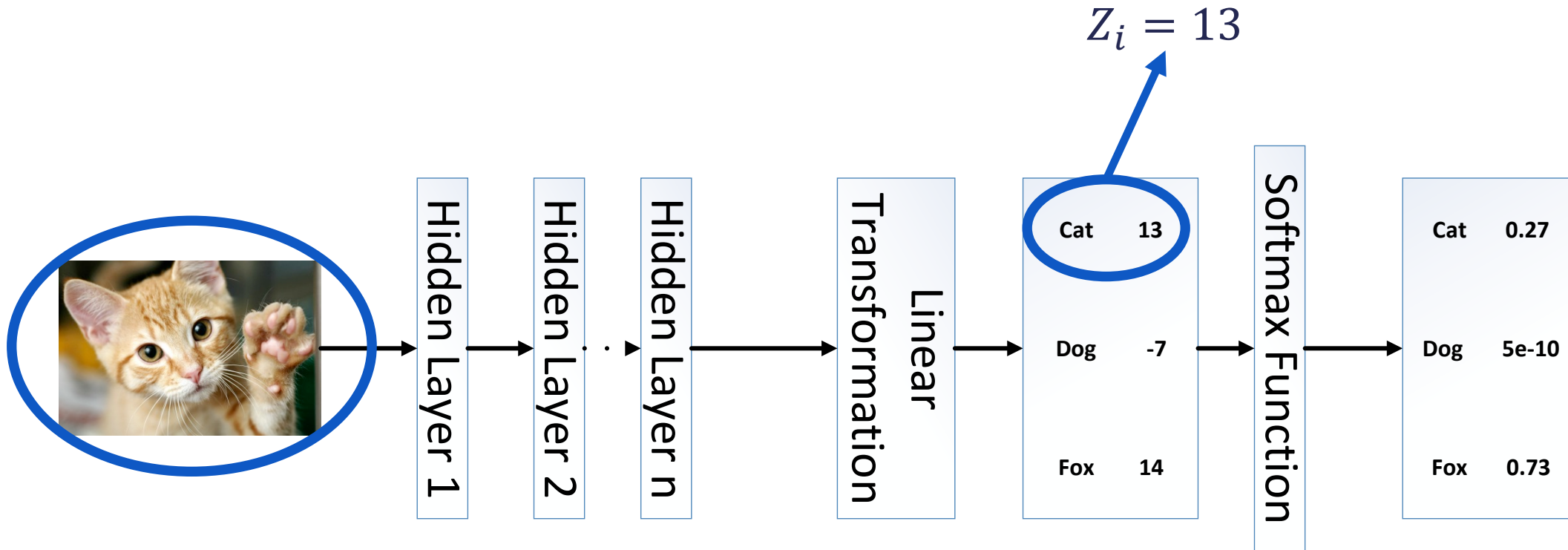


# Classification: Softmax Output Layers



$$L(\theta) = -z_i + \log \sum_j \exp(z_j) = -13 + \log(\exp(13) + \exp(-7) + \exp(11)) = 0.12$$

# Classification: Softmax Output Layers



$$L(\theta) = -z_i + \log \sum_j \exp(z_j) = -13 + \log(\exp(13) + \exp(-7) + \exp(14)) = 1.31$$

# Regression: Linear Output Layers

- **Linear Output Units** are based only on an affine transformation with no non-linearity

$$z = W^T h + b$$

- **Linear Output Units** are usually used with the **Mean Squared Error** loss function to model the **mean** of a probability distribution:

$$L(\theta) = \sum_i (z_i - f^*(x_i))^2$$

# Summary

- To build a machine learning model you need:
  - A model
  - A loss function
  - An optimization procedure
- Loss functions are chosen based on the task at hand
- **Next: Optimization**

