

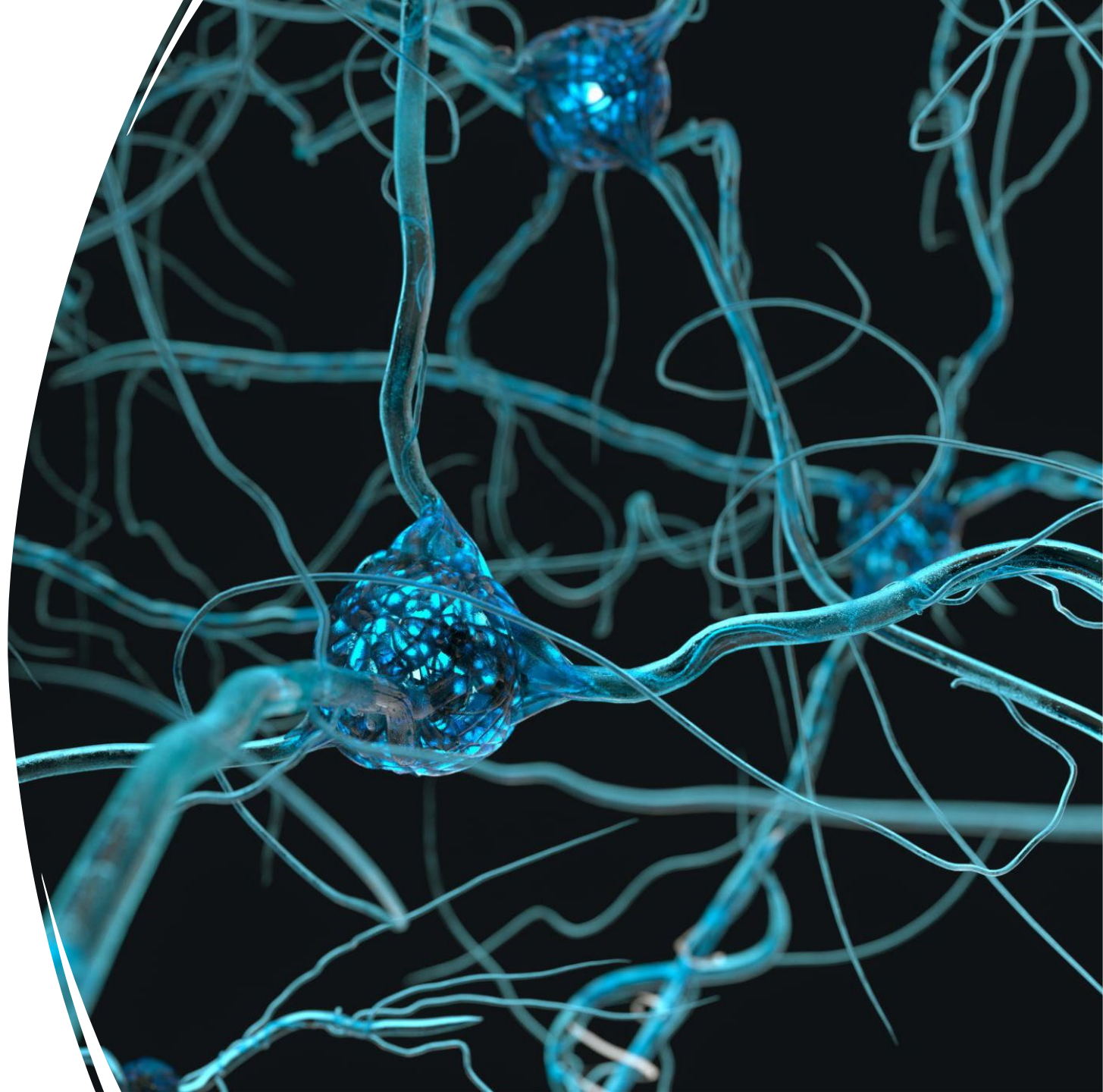
Chapter 10

Introduction to Artificial Neural Networks with Keras

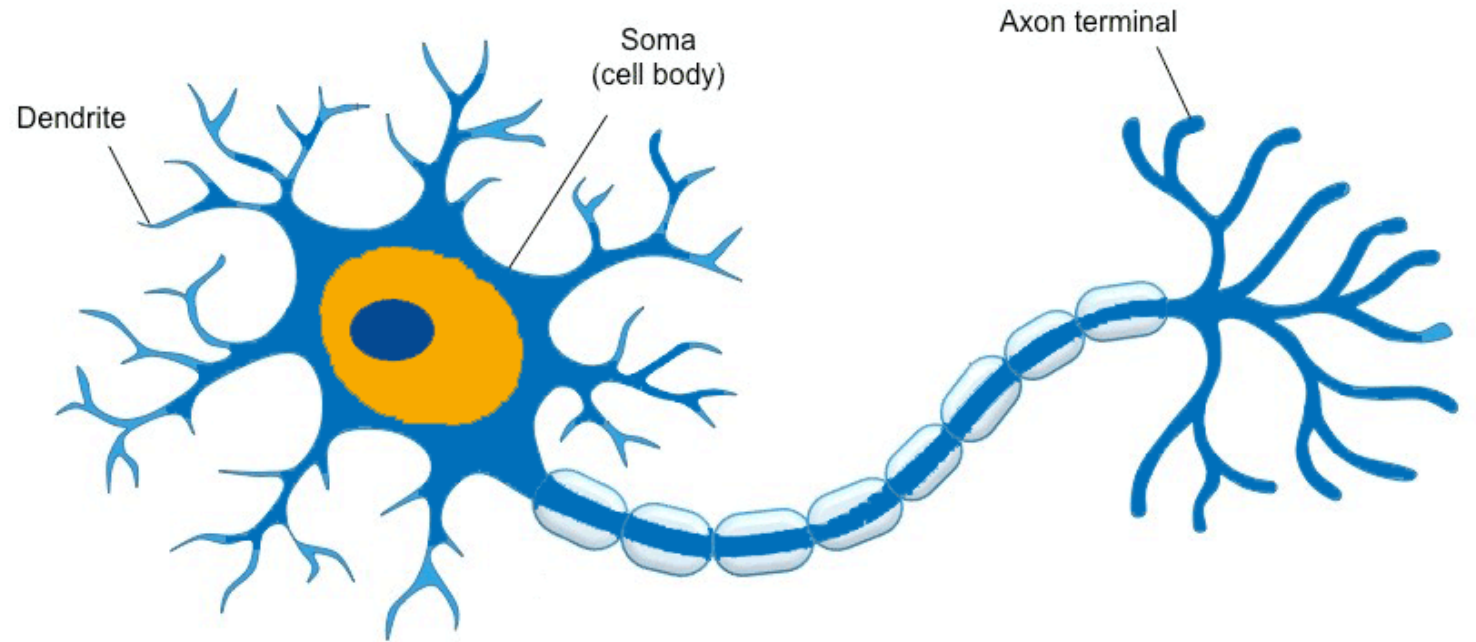
+
•
◦ And now for
something different ...

Agenda

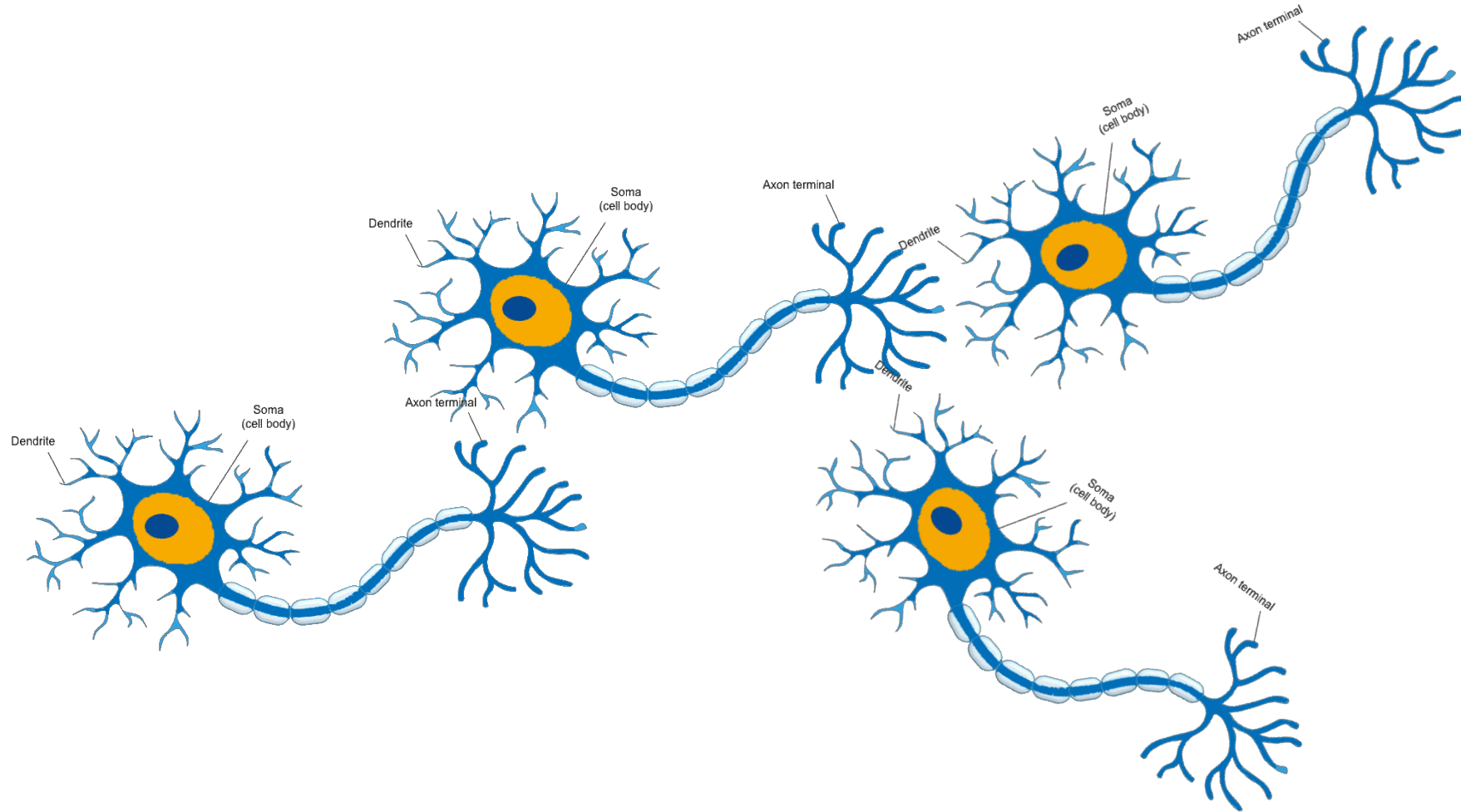
- Biologic Neurons and Neural Networks
- Artificial Neurons and Neural Networks
- Demo
 - TensorFlow
 - Keras



Animal Neuron



Animal Neural Network



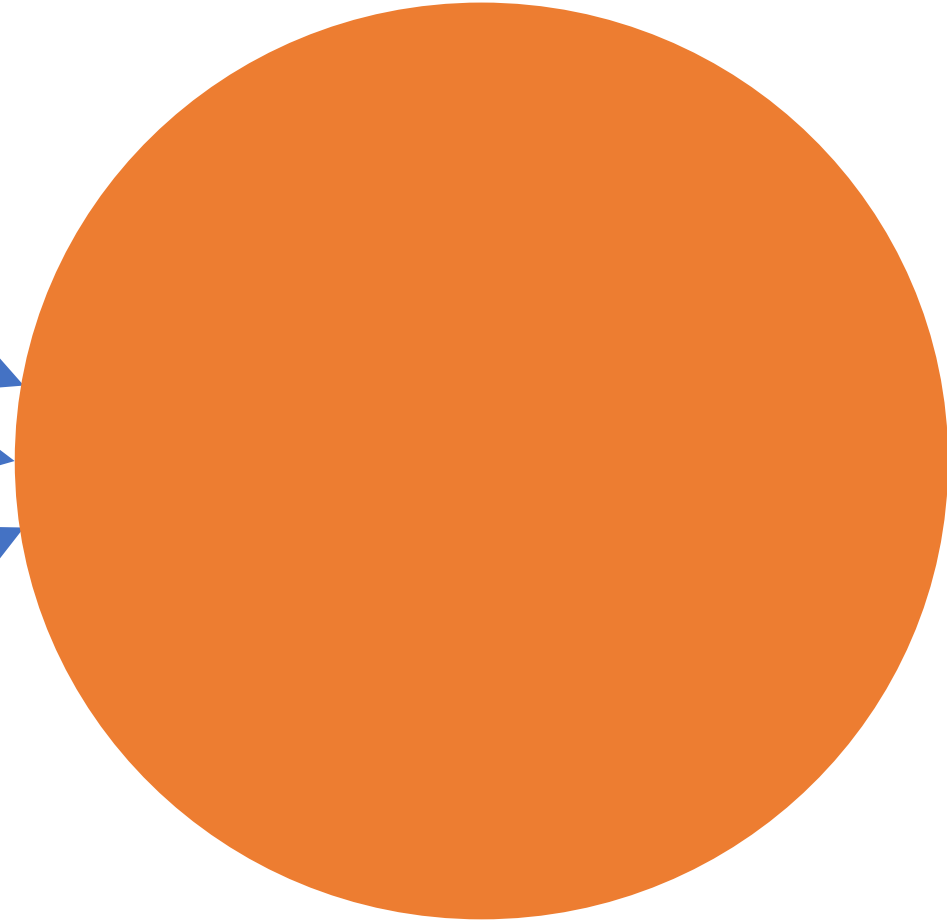
Artificial Neuron

Inputs

I_1

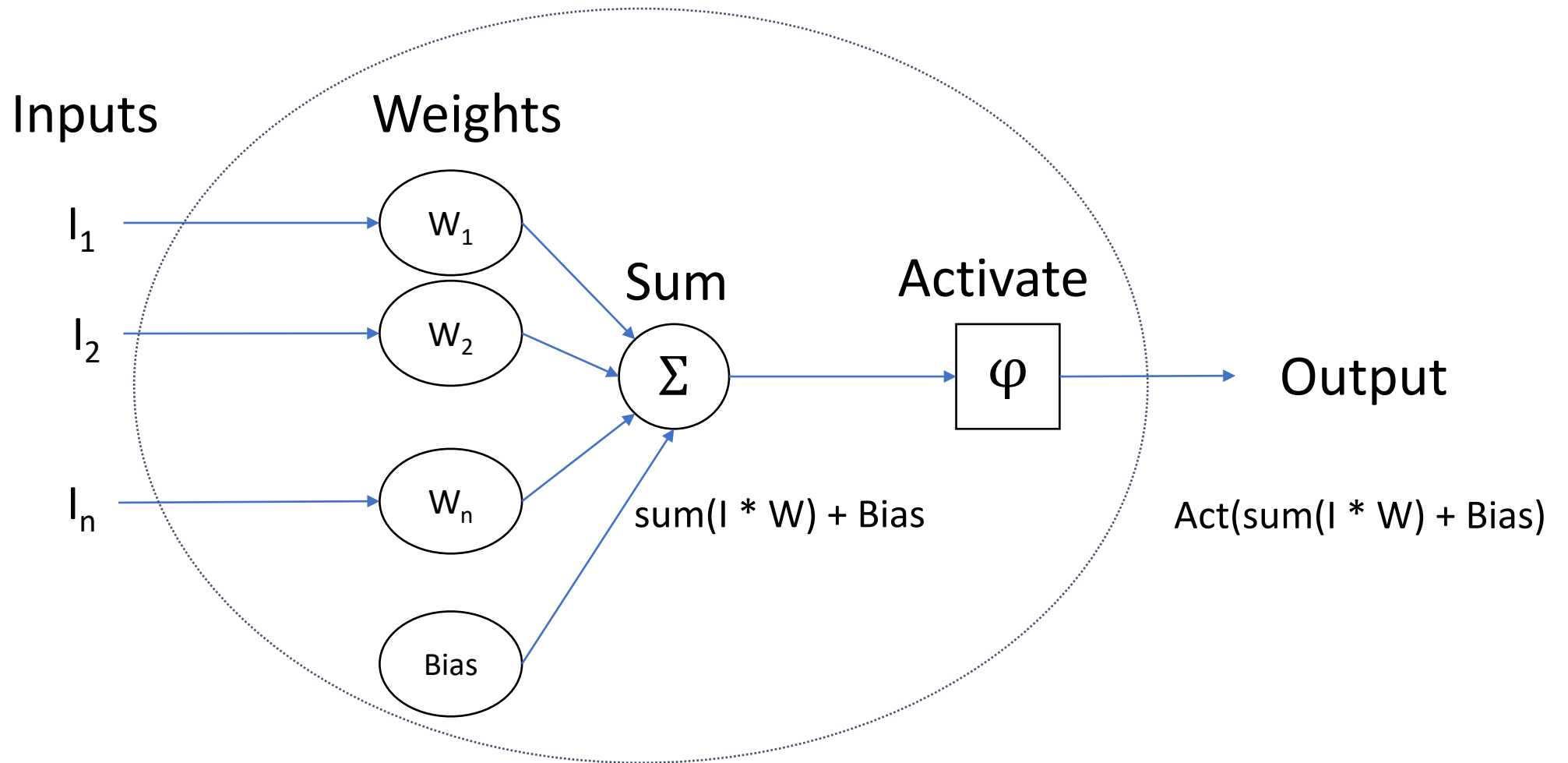
I_2

I_n



Output

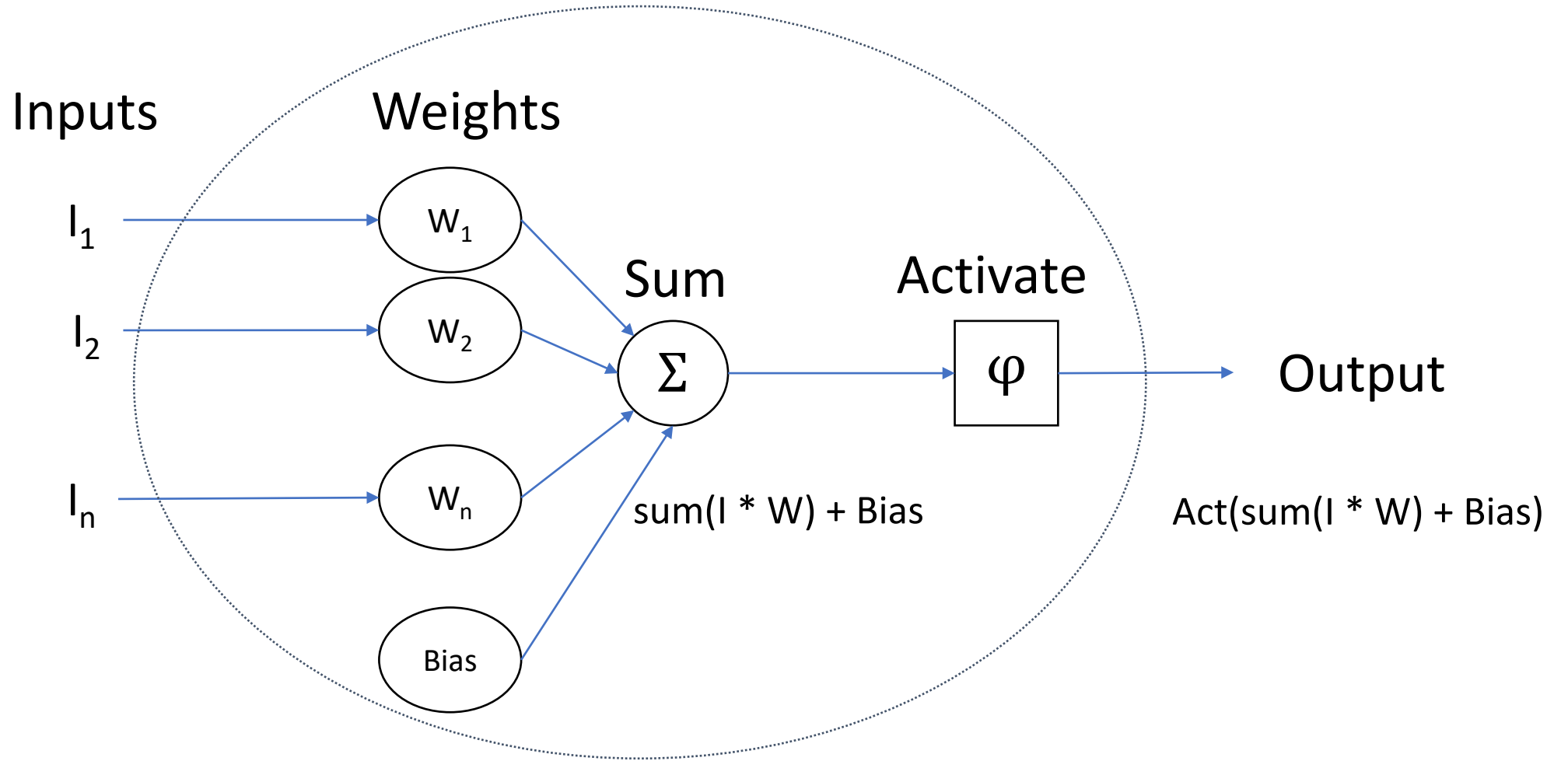
Artificial Neuron



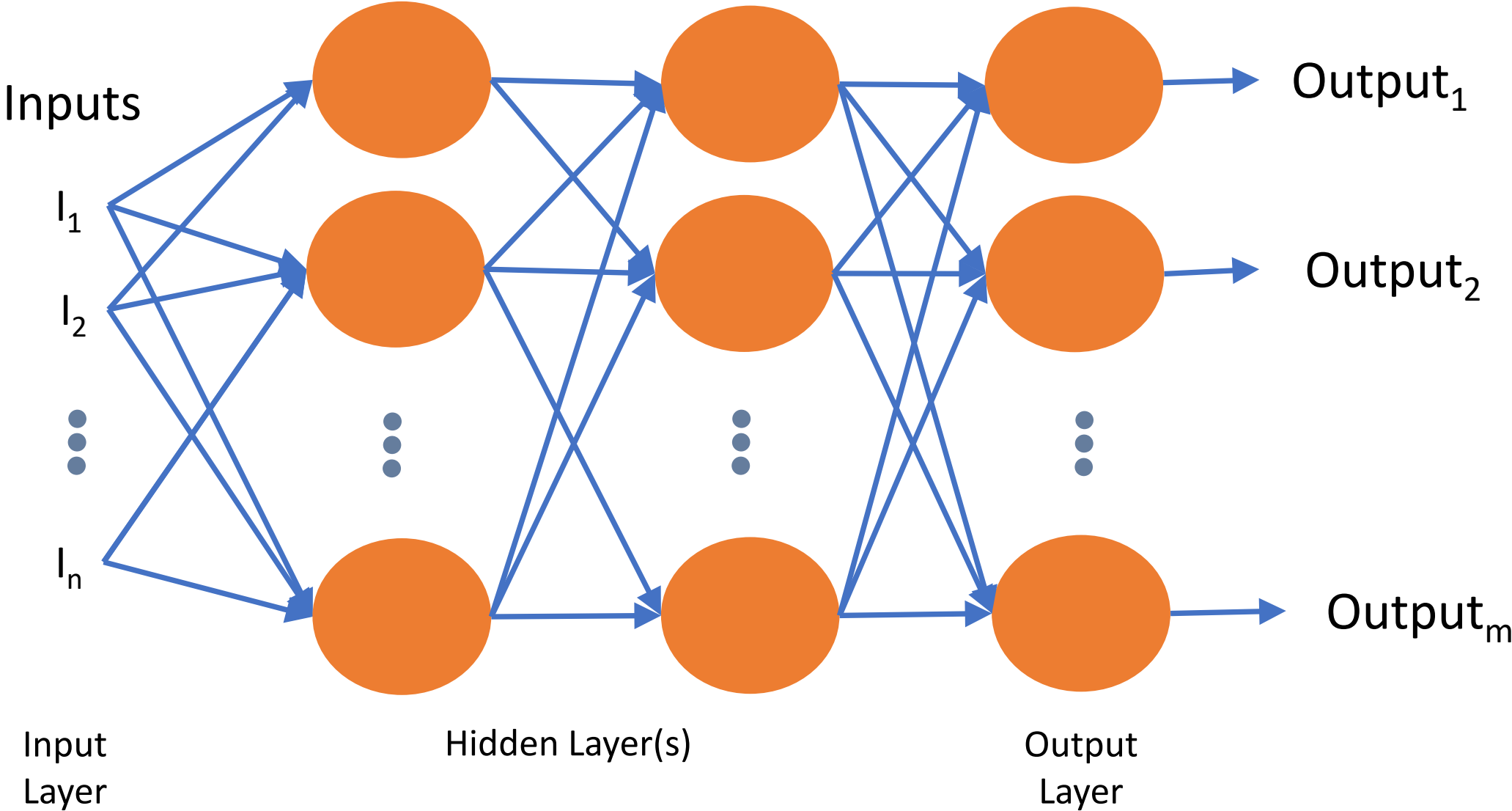
Example Activation Functions

- Step (Perceptron)
 - Binary
 - If $x > \text{threshold} = 1$ then x else 0
 - If $x > 0.5$ then 1 else 0
(Perceptron)
 - Uncommon
- ReLU – Rectified Linear Unit
 - $F(x) = \max(0, x)$
 - If $x < 0$ then 0 else x
 - Common
 - A parameter you set

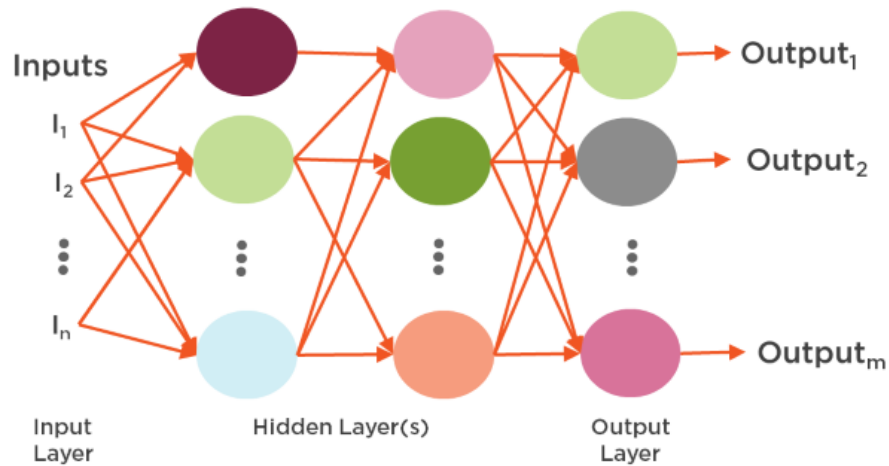
Artificial Neuron



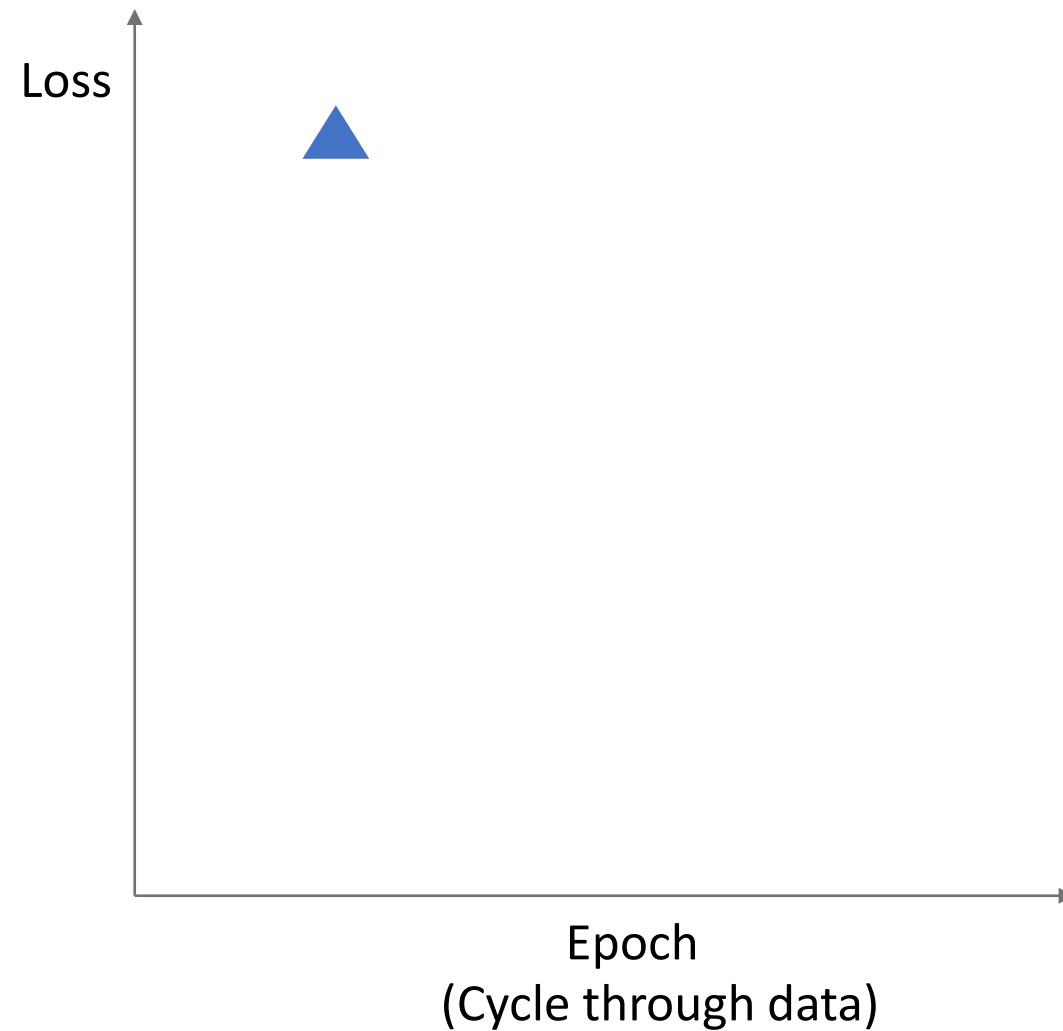
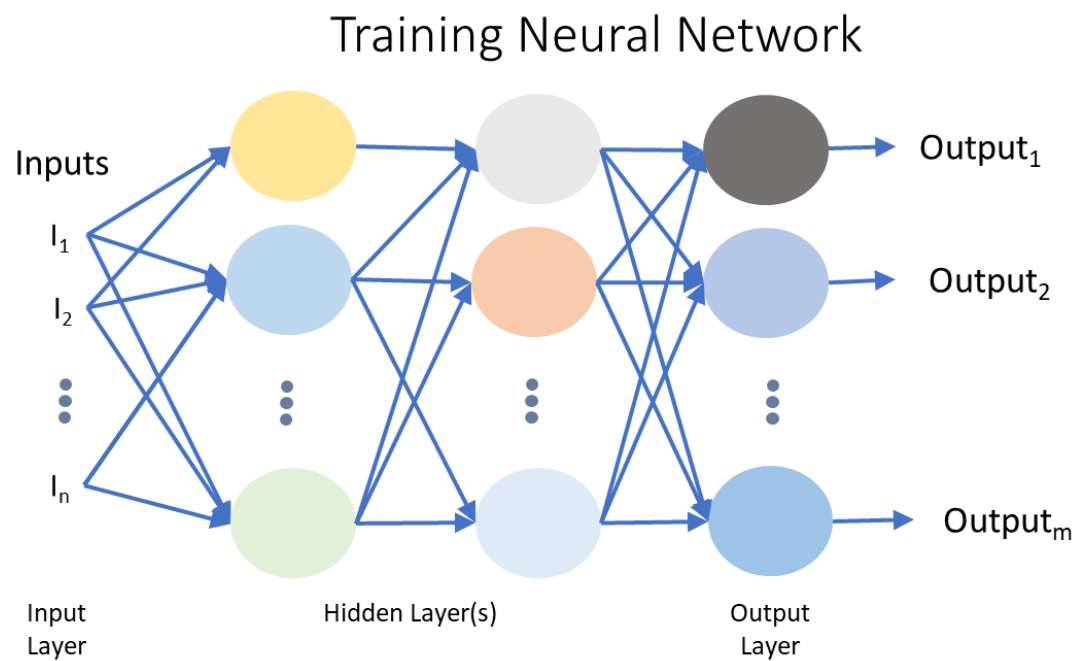
Neural Network Layers

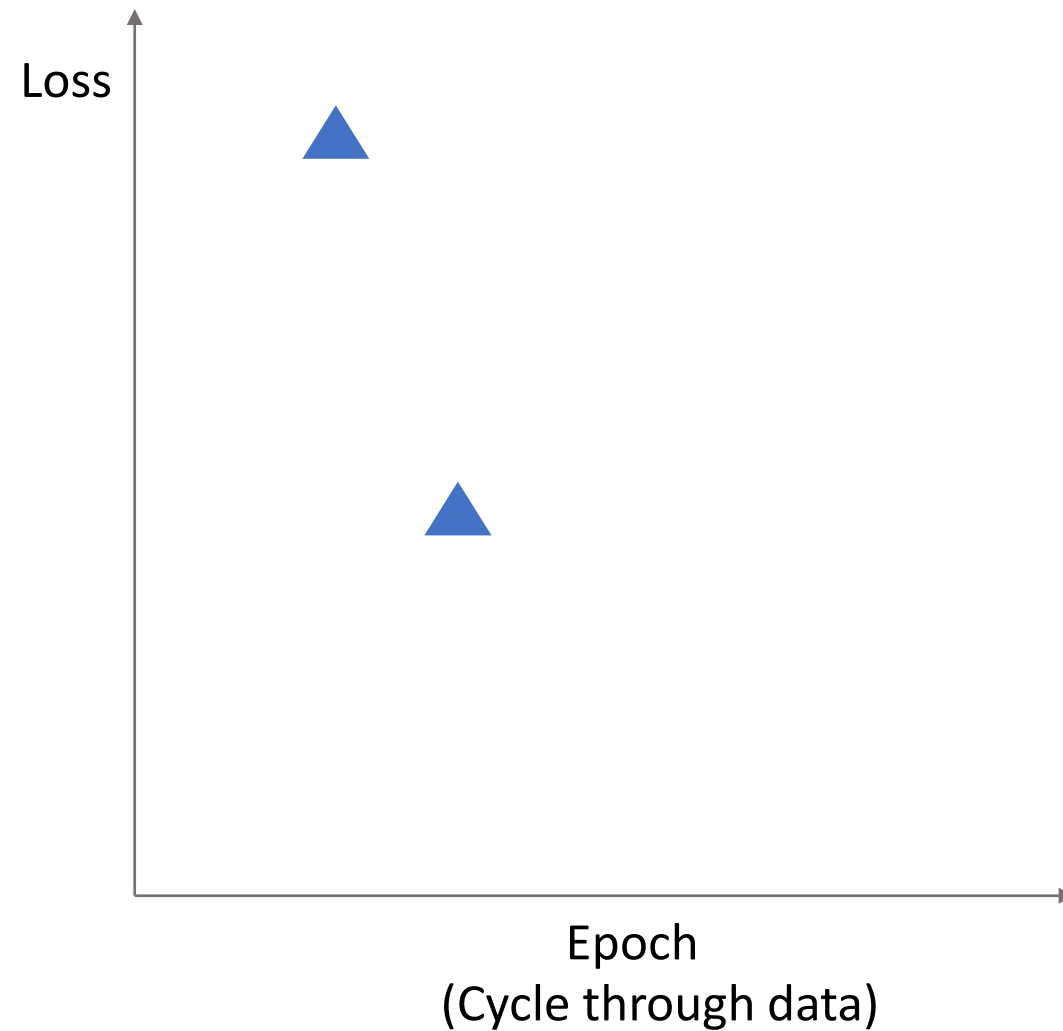
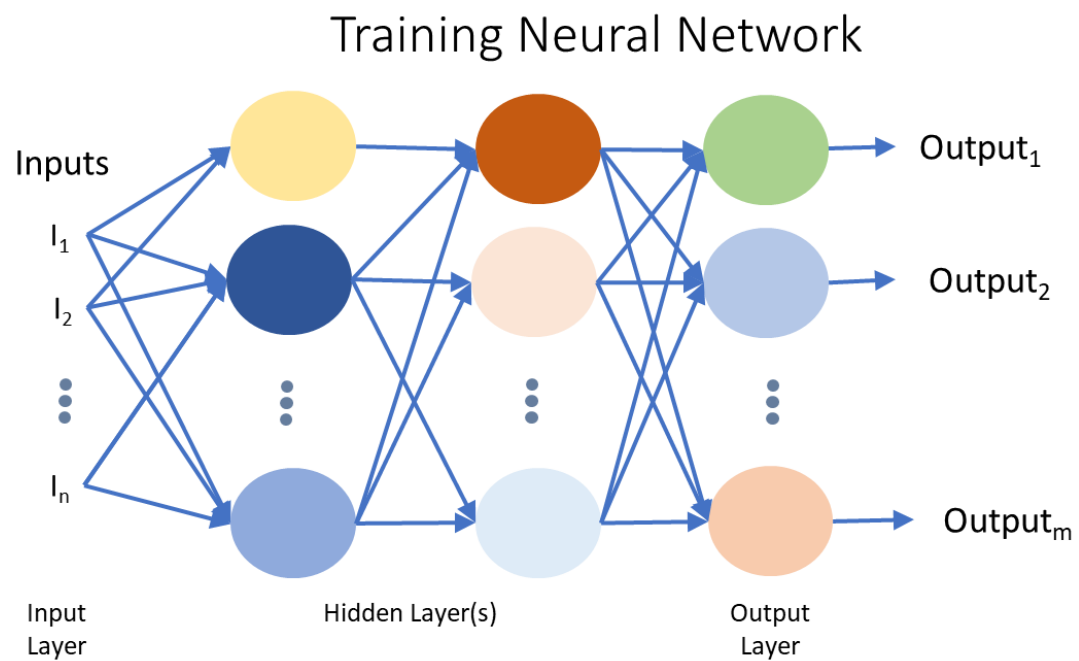


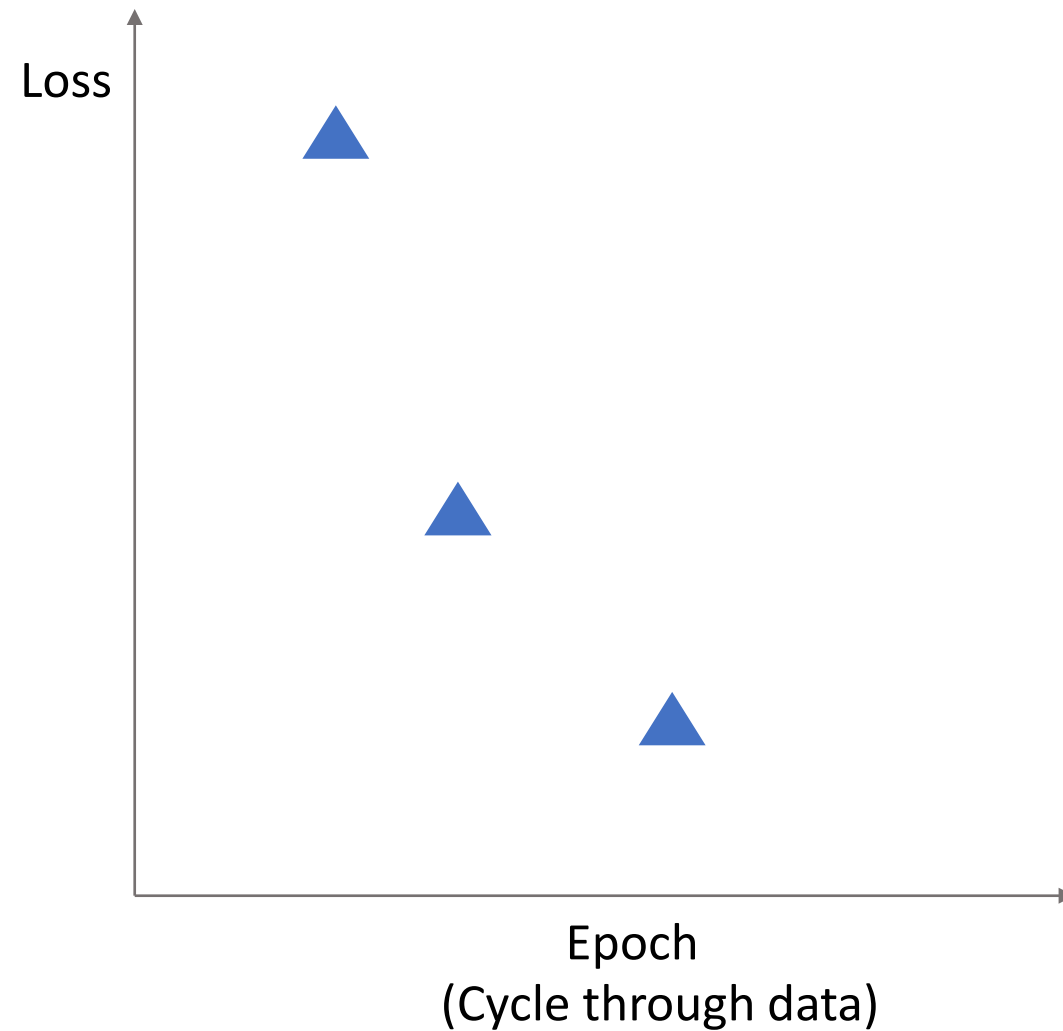
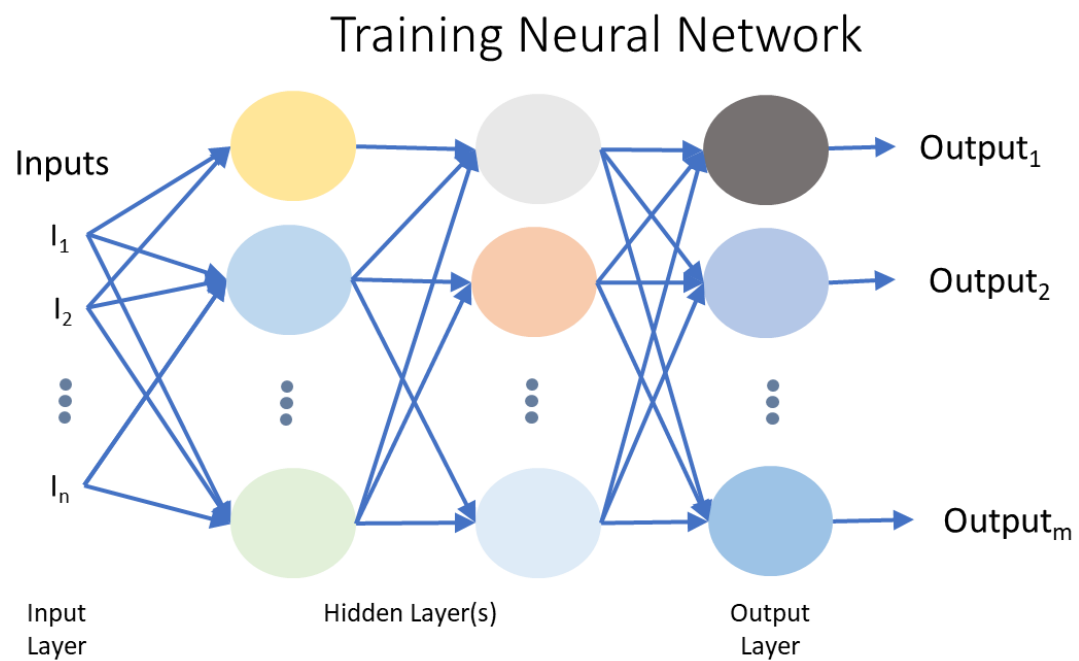
Neural Network

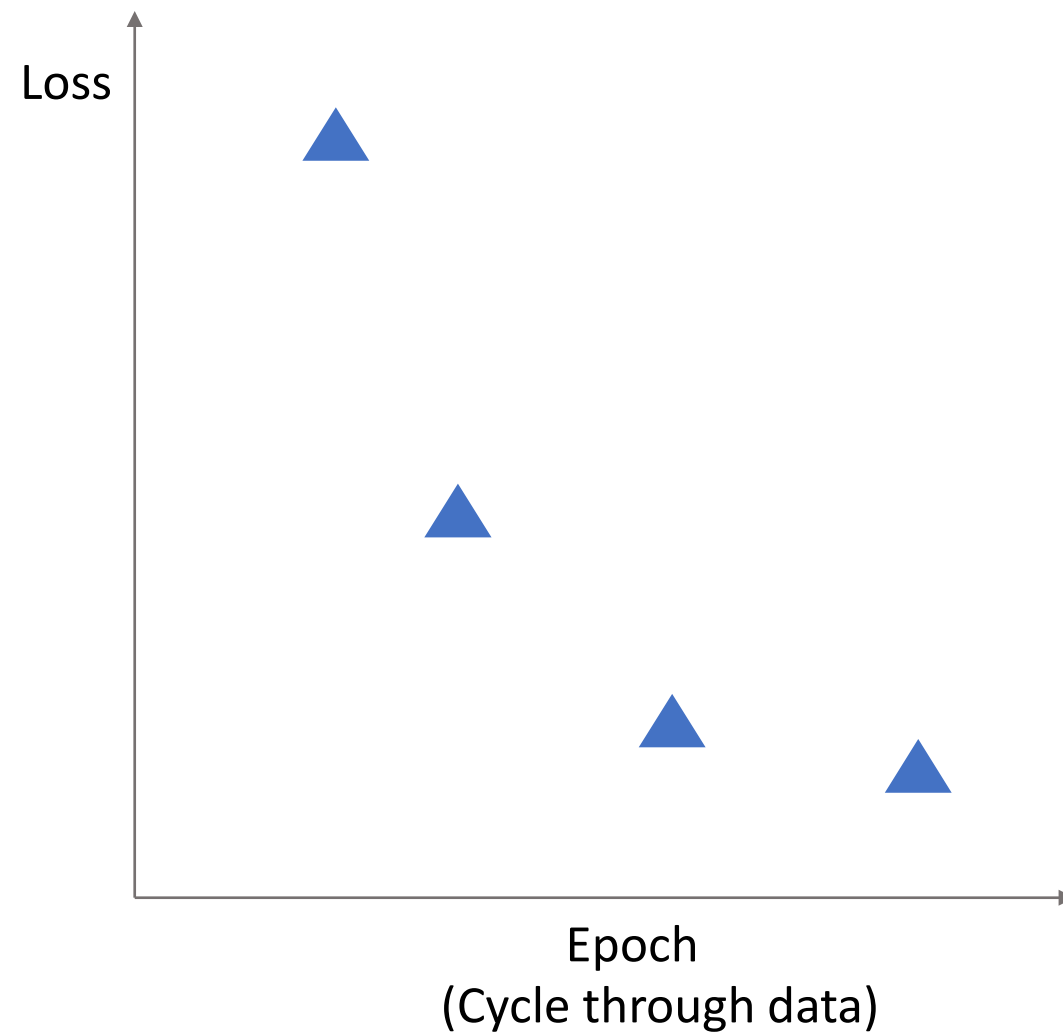
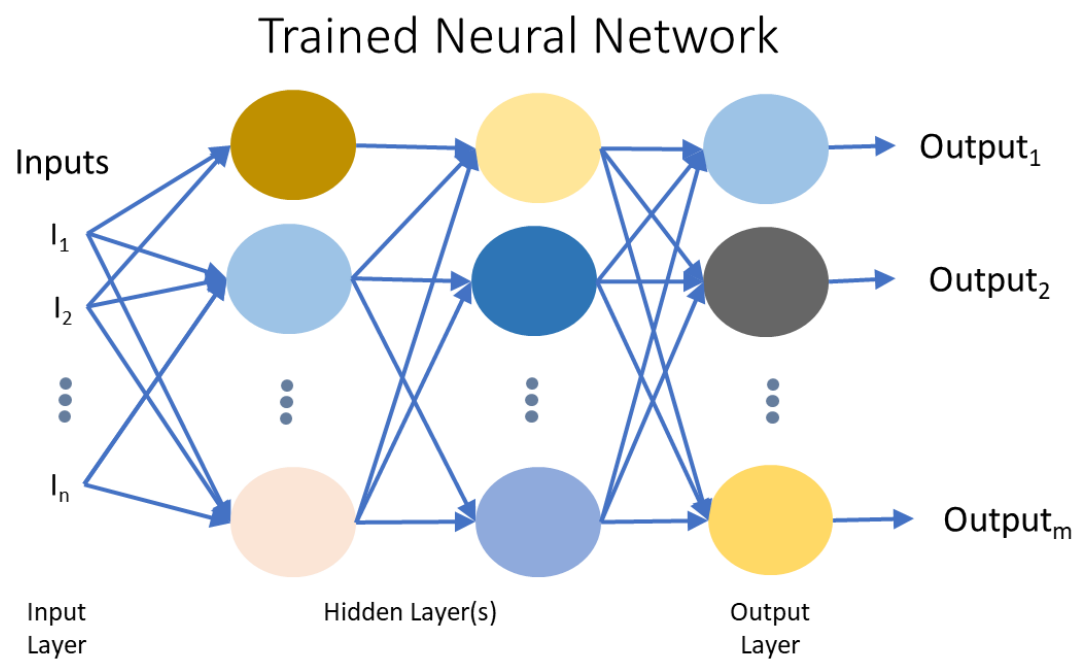


- Train with batches of input data
- Training goal is to minimize error
- Training adjusts weights and biases using ***settings you specify***
 - Epochs - times through dataset
 - Batch Size – # of **random** samples in a batch
 - Loss Function – measures loss (error)
 - Optimizer – updates weights to reduces loss
- Backward propagation of error (BackProp)
 - From Output towards Input
 - Adjust weights using optimizer to reduce loss
- Notes
 - Order and contents of training batches changes
 - 2+ Hidden layers -> Deep Neural Network

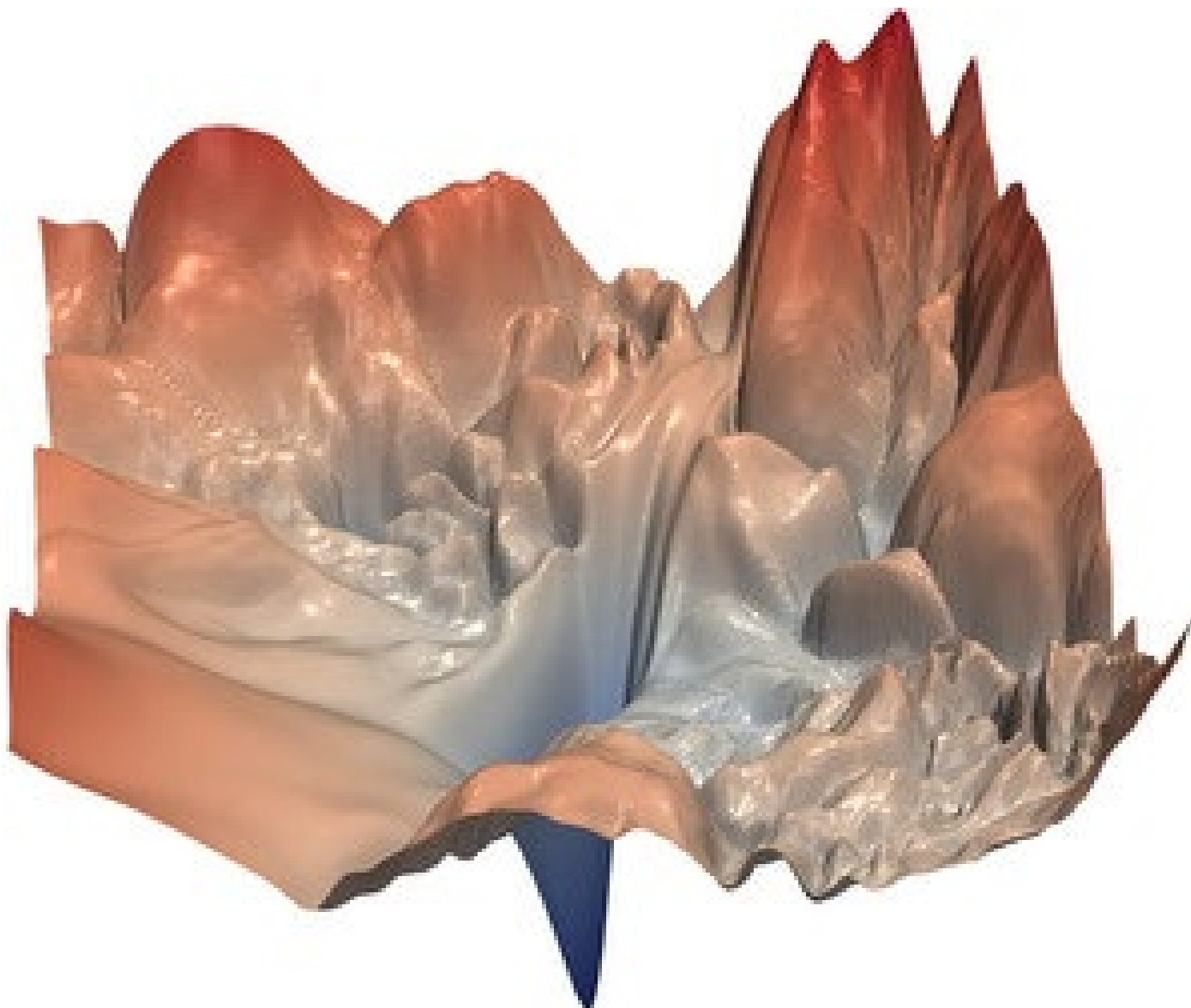








-
- Loss Space for ResNet-56

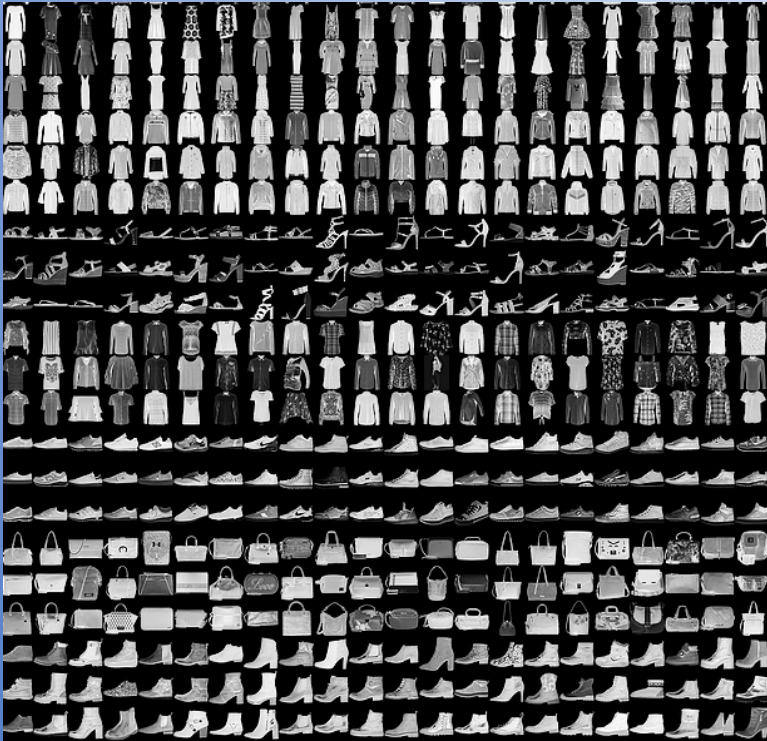




DEMO ENVIRONMENT

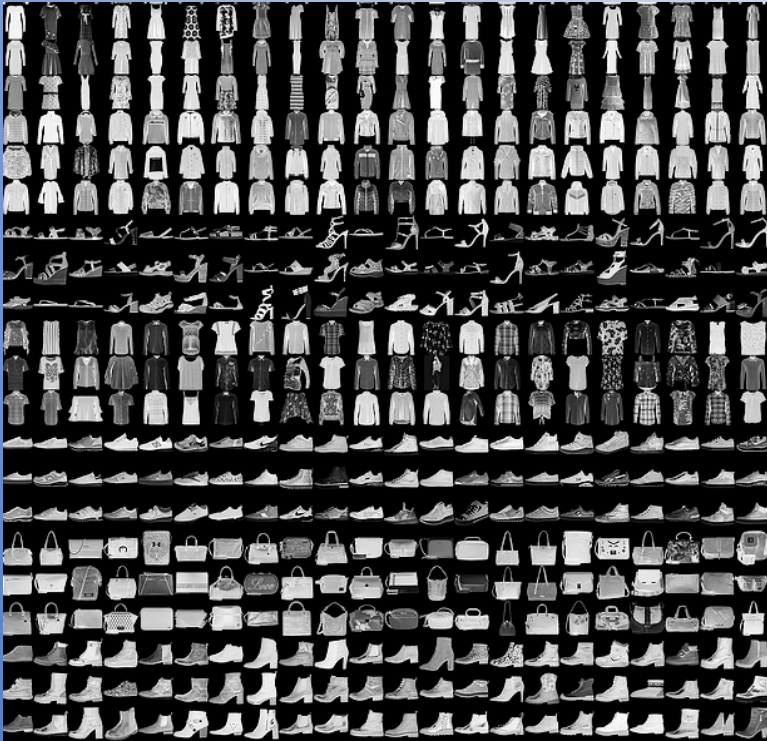
- Colab
- TensorFlow
- Keras
- Fashion MNIST Dataset

Fashion MNIST Dataset



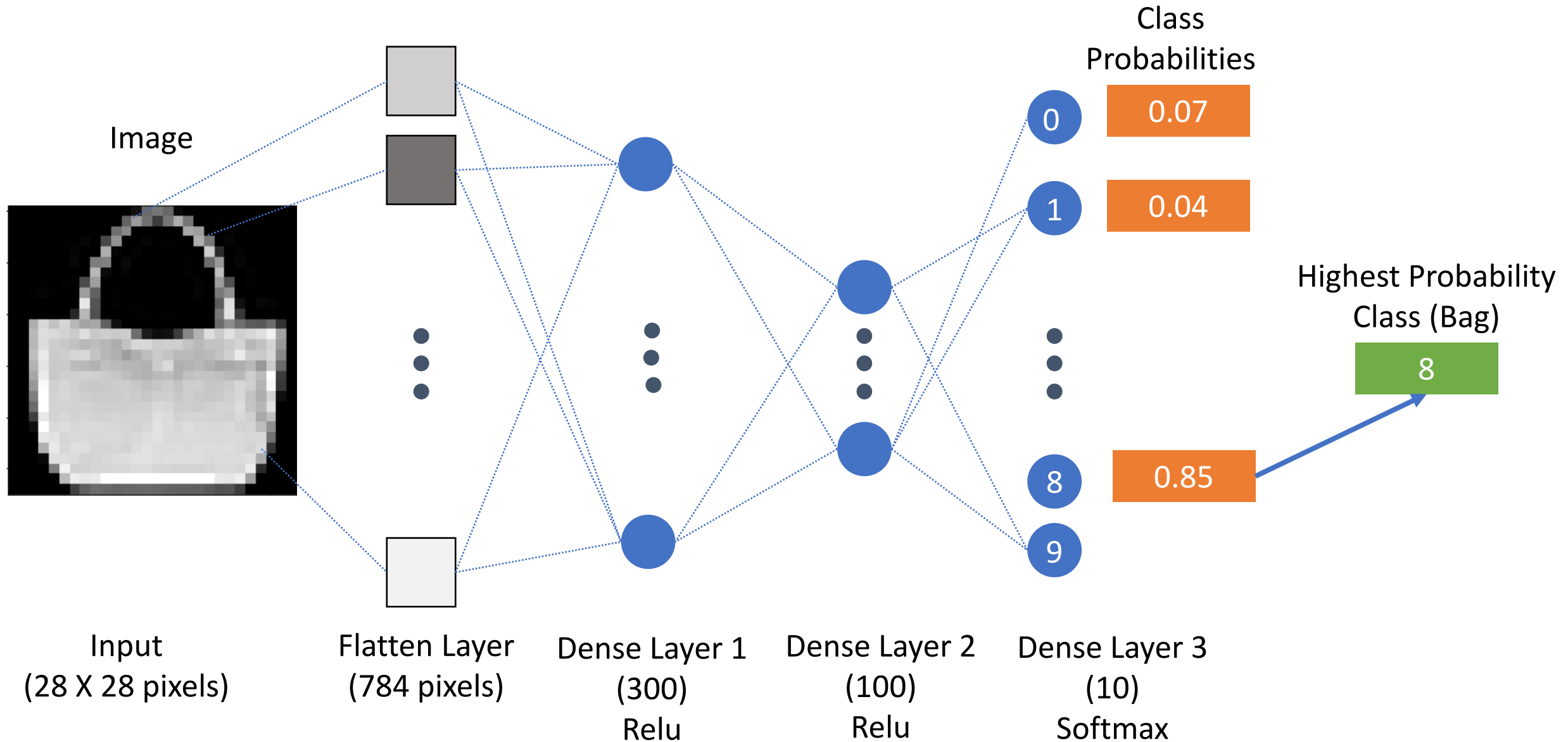
- Clothing item recognition
 - 10 classes
- Fashion MNIST dataset
 - <https://github.com/zalandoresearch/fashion-mnist>
- Replacement for MNIST handwritten digits
 - Too easy for NN – 99.7% accuracy common
 - Overused
 - Not representative of modern problems

Fashion MNIST Dataset



- 70,000 data points
 - 60,000 training & validation
 - 10,000 testing
- Each data point contains
 - 28 X 28 grayscale image (low-res)
 - Image class (0-9)
- Some images in different classes have similar appearance (sneaker/sandal)

Model Architecture



```
mirror_mod = modifier_ob.  
#set mirror object to mirror  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
#selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES ----  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

Demo

[SanDiego-ML/Hands on ML Chapter 10.ipynb at main · JerryKurata/SanDiego-ML \(github.com\)](#)

Network Tuning



- Based on experience
- Adjustments
 - Data
 - Arrangement of layers
 - Types of layers
 - Number of Hidden layers
 - Neurons per layer
 - Hyperparameters
 - Batch size, Learning rate, ...
- Covered in later chapters