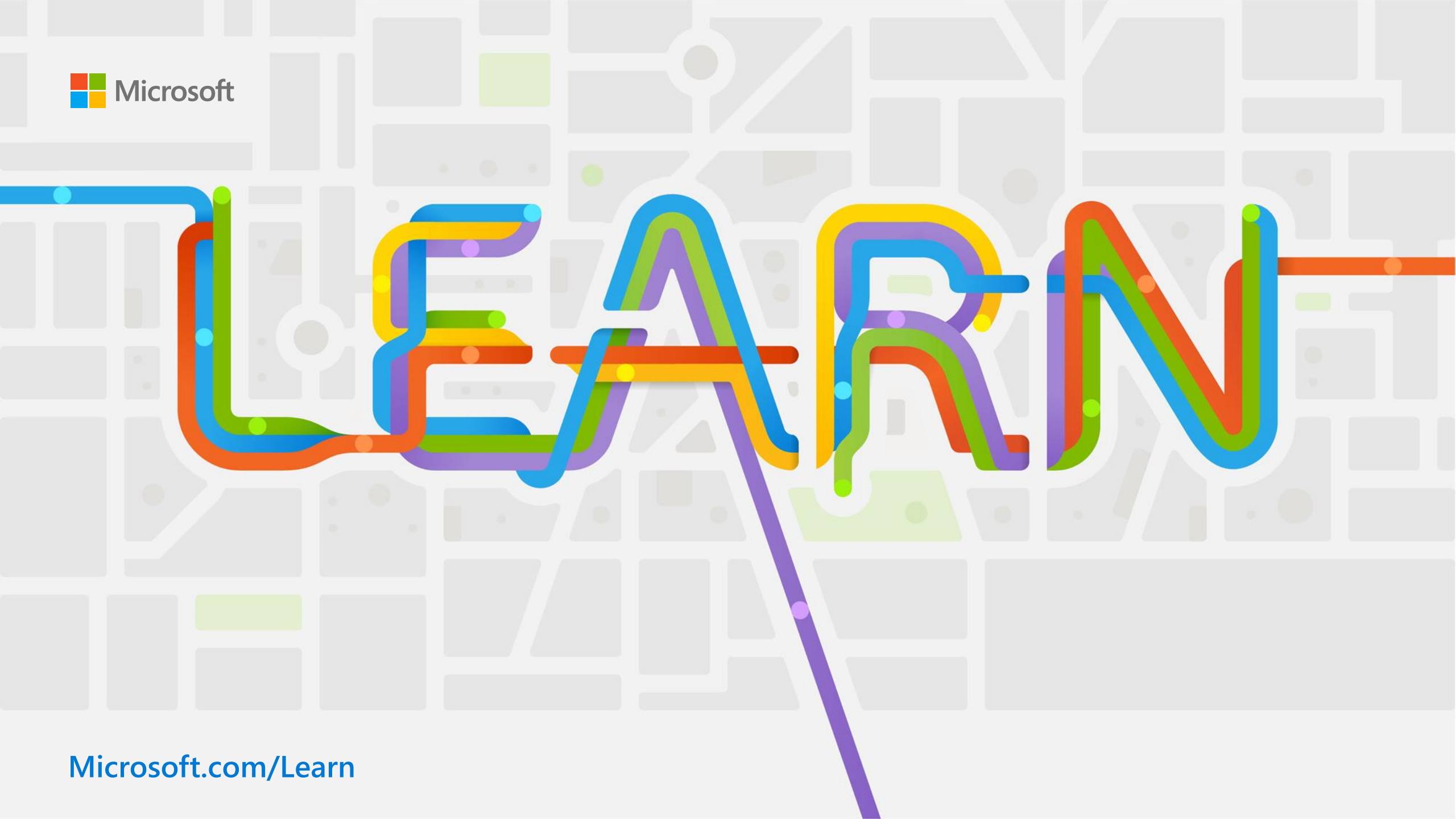

Microsoft
LEARN
Microsoft.com/Learn

Microsoft

Introduction to Deep Learning for Computer Vision

http://aka.ms/cvworkshop

Dmitry Soshnikov

Cloud Developer Advocate
Microsoft

Join the chat at https://aka.ms/LearnLiveTV

Goal

Imagine pet care center that receives many breeds of cats and dogs every day. Nurses need to feed them according to their breeds. We will train a model that can be used to recognize breed of a pet.

Learning objectives

- Learn about neural networks in general
- Learn about computer vision tasks most commonly solved with neural networks
- Understand how Convolutional Neural Networks (CNNs) work
- Train a neural network to recognize pets breeds from faces
- OPTIONAL: Train a neural network to recognize breeds from original photos using Transfer Learning

Prerequisites

- Basic knowledge of Python and Jupyter Notebooks
- Some familiarity with PyTorch/TensorFlow framework, including tensors, basics of back propagation and building models
- Understanding machine learning concepts, such as classification, train/test dataset, accuracy, etc.

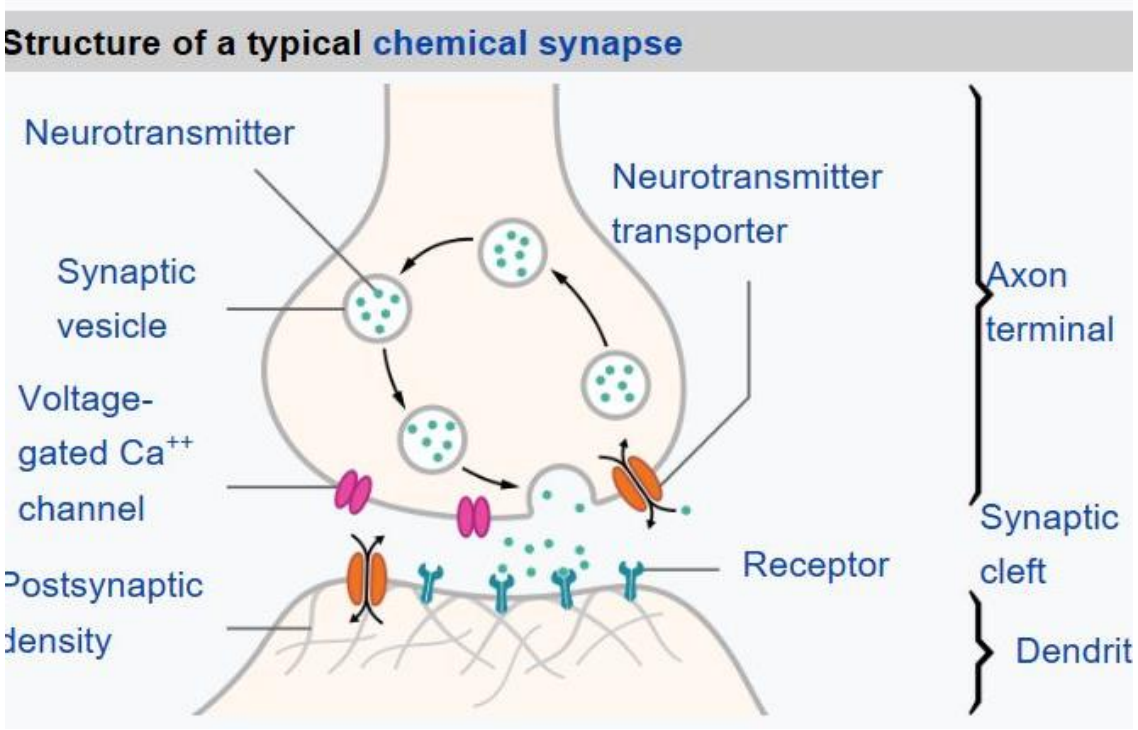
To Learn:

- Read: http://eazify.net/nnintro
- Introduction to PyTorch: http://aka.ms/learntorch/intro
- Introduction to TensorFlow: http://aka.ms/learntf/keras

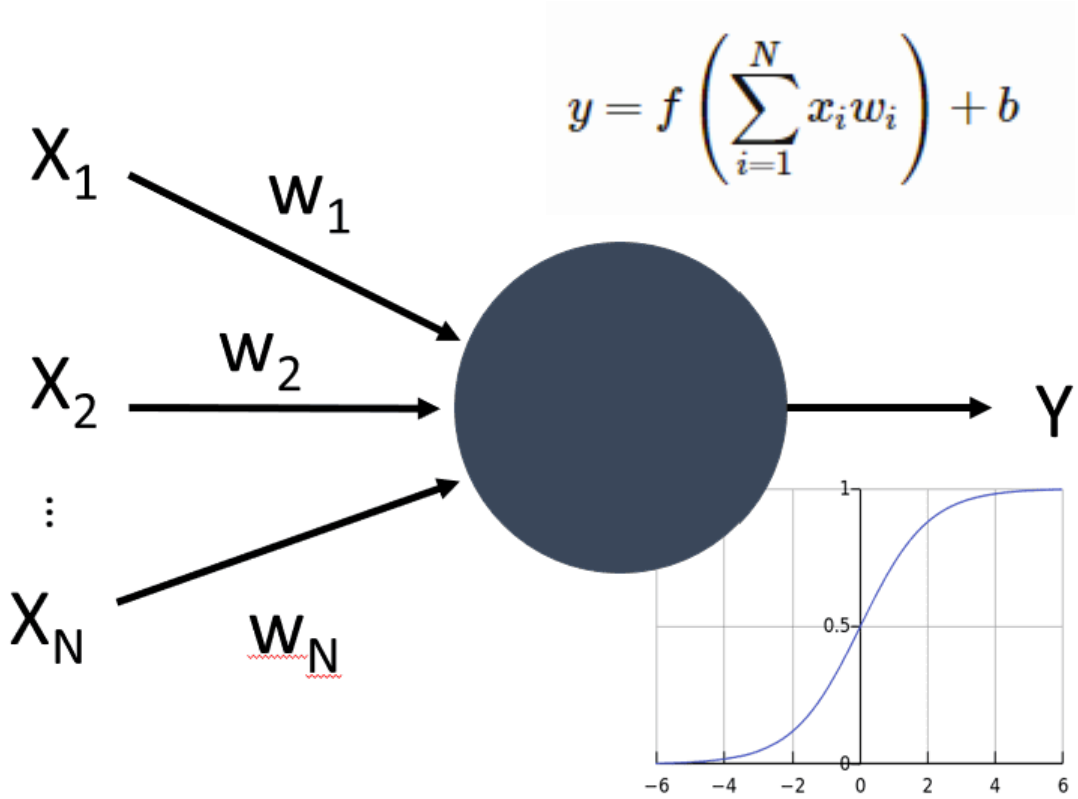
Introduction to Neural Networks

Neural Networks are inspired by our Brain

http://eazify.net/nnintro

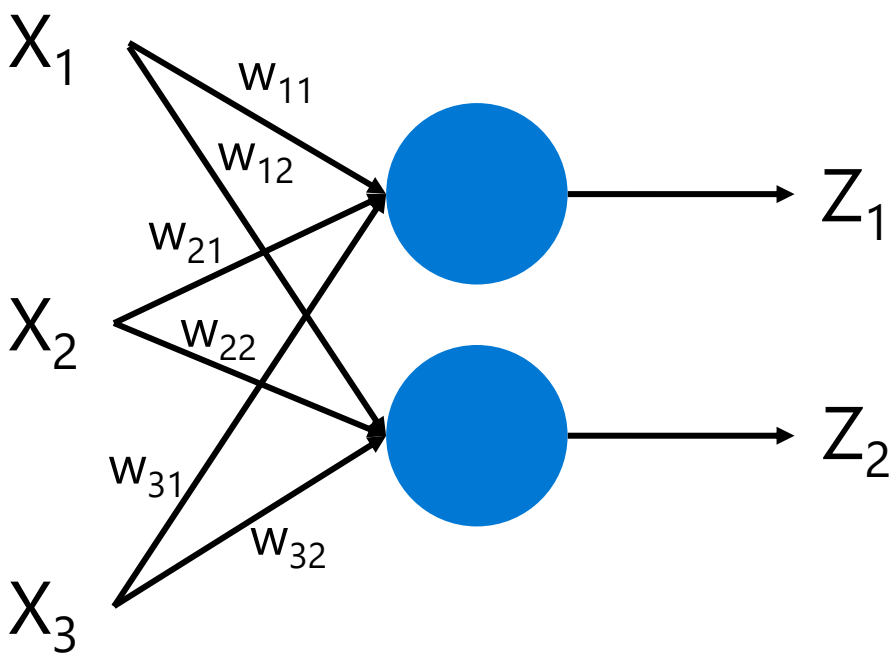


Real Neuron



Artificial Neuron

Tensors



$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$z = Wx + b$$

Sizes: Z – 2x1, W – 2x3, X - 3x1, b – 2x1

Computing in minibatches (bs=9):

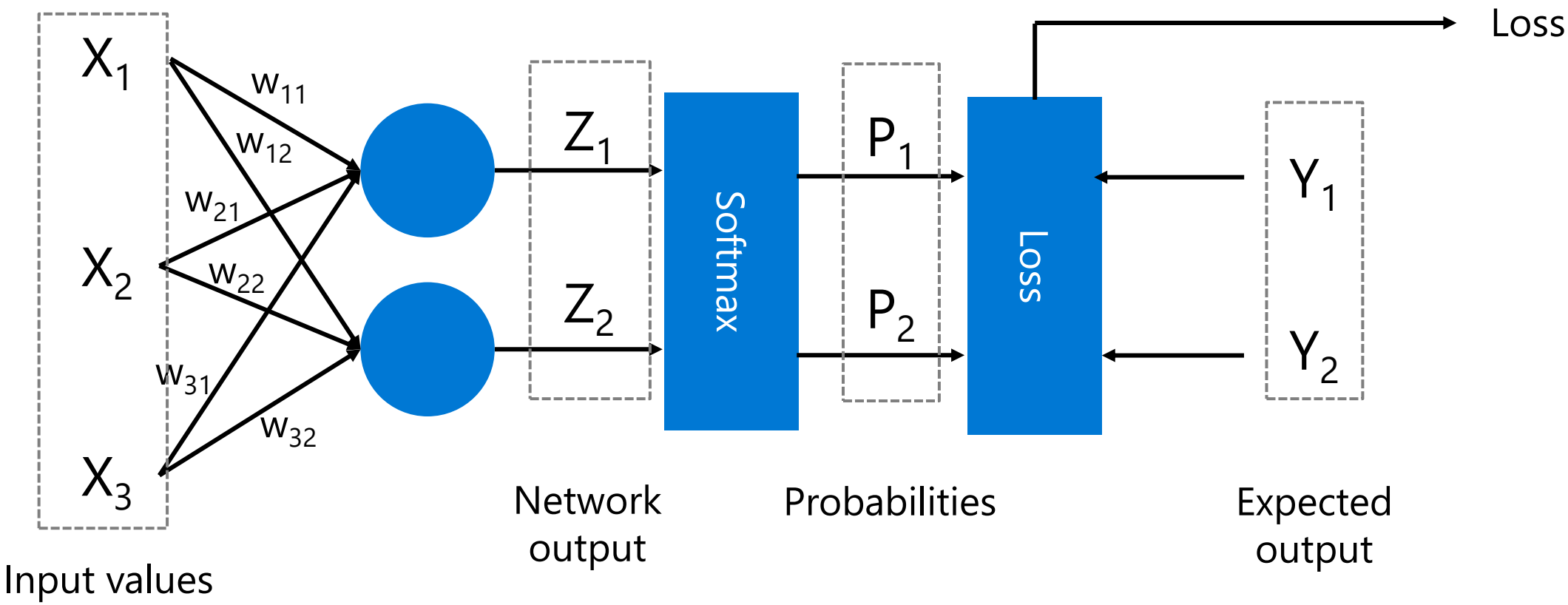
$$\begin{pmatrix} z_{11} \\ z_{12} \end{pmatrix} \begin{pmatrix} z_{21} \\ \therefore \end{pmatrix} \begin{pmatrix} z_{91} \\ z_{92} \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \end{pmatrix} \begin{pmatrix} x_{91} \\ x_{92} \\ x_{93} \end{pmatrix} \therefore + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Sizes: Z – 9x2x1, W – 2x3, X – 9x3x1, b – 2x1

Softmax and Loss

$$L(w,b) = \text{CrossEntropy}(\text{Softmax}(wx+b), y) \rightarrow \min$$

$$W^{(i+1)} = W^{(i)} - \eta \frac{\partial L}{\partial W} \qquad b^{(i+1)} = b^{i} - \eta \frac{\partial L}{\partial b}$$



Neural Network Frameworks

Two main things neural network frameworks do:

- Operate on Tensors efficiently (using GPU if possible)
- Offer automatic differentiation (calculate gradients)
- Also: load datasets, transform data, optimization algorithms, built-in network layers, etc.

- First mainstream framework
- A lot of code on GitHub / Samples
- Includes Keras – “Deep Learning for Humans”
- Easier to start with

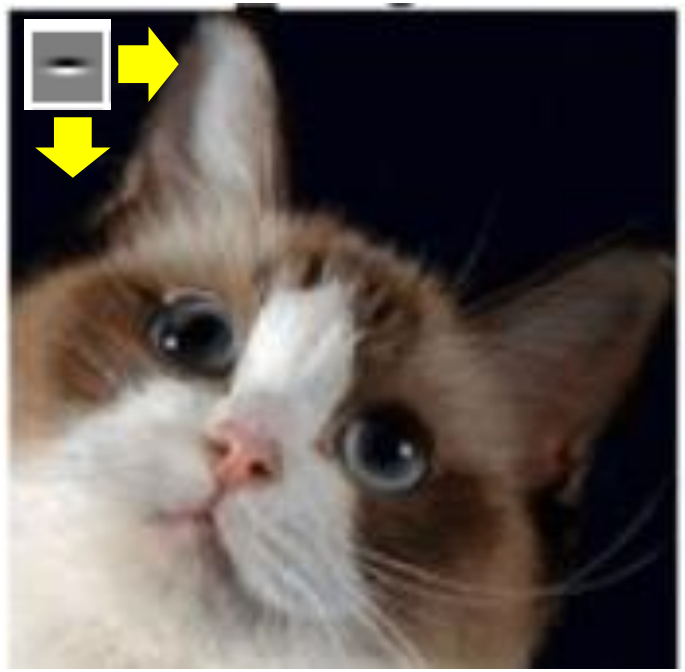
- Quickly gaining popularity
- Provides deeper understanding of neural network mechanics

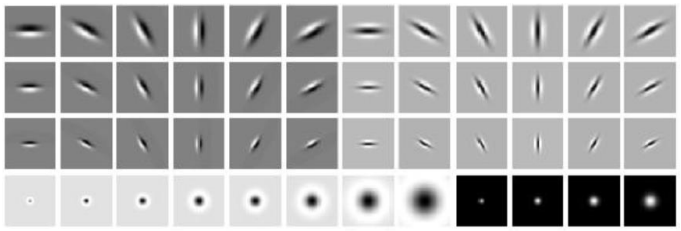
Let’s Get to Work!

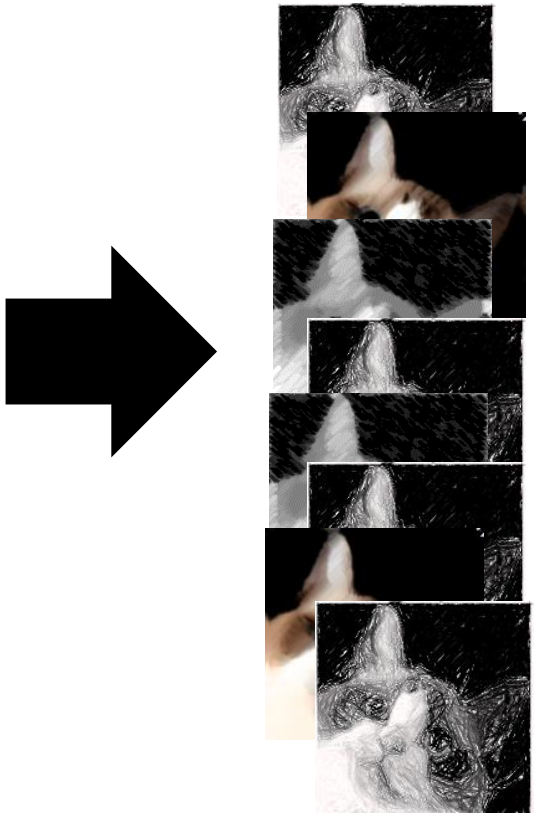
https://aka.ms/learntf/vision

https://aka.ms/learntorch/vision

Convolutional Neural Networks



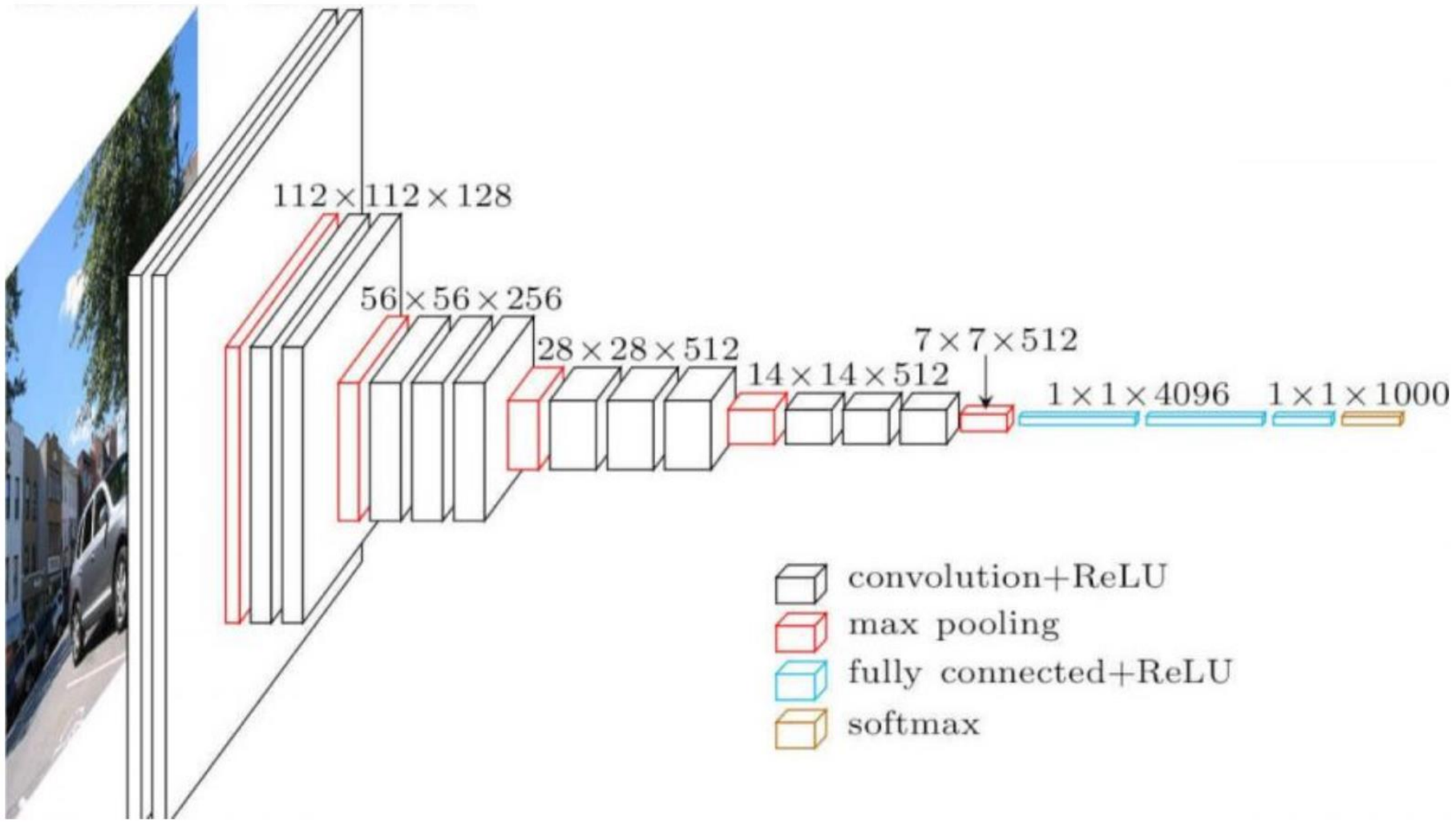




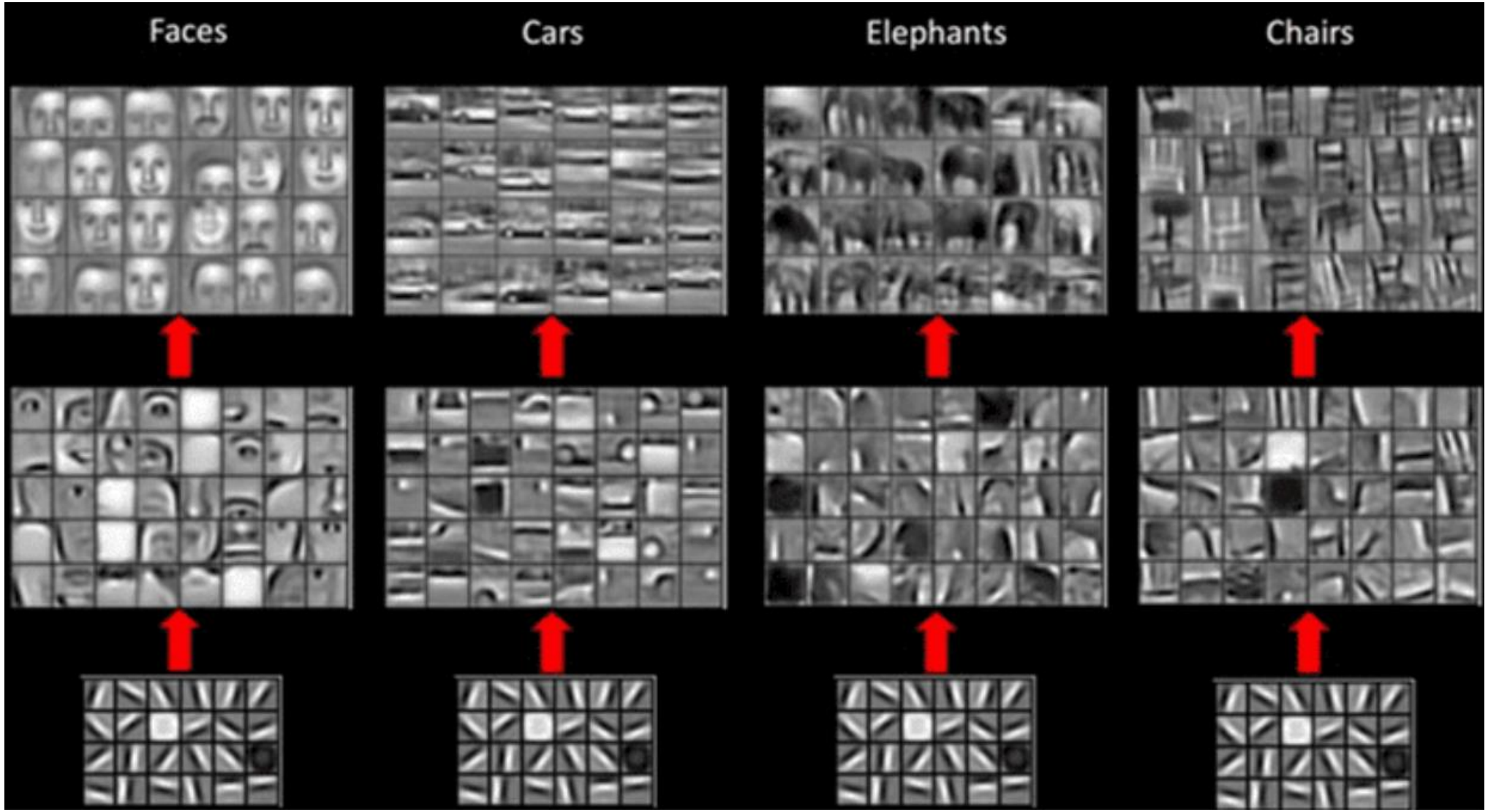
Cat

Dog

Pyramid Architecture



Hierarchical Feature Extraction



Project 1: Pet Face Recognition

Project 1: Pet Face Recognition

Get Data

```
!wget https://mslearntensorflowlp.blob.core.windows.net/data/petfaces.tar.gz
!tar xfz petfaces.tar.gz
!rm petfaces.tar.gz
```

Neural Network Training

Load data into tensors

- Resize images
- Normalize images
- Split into batches

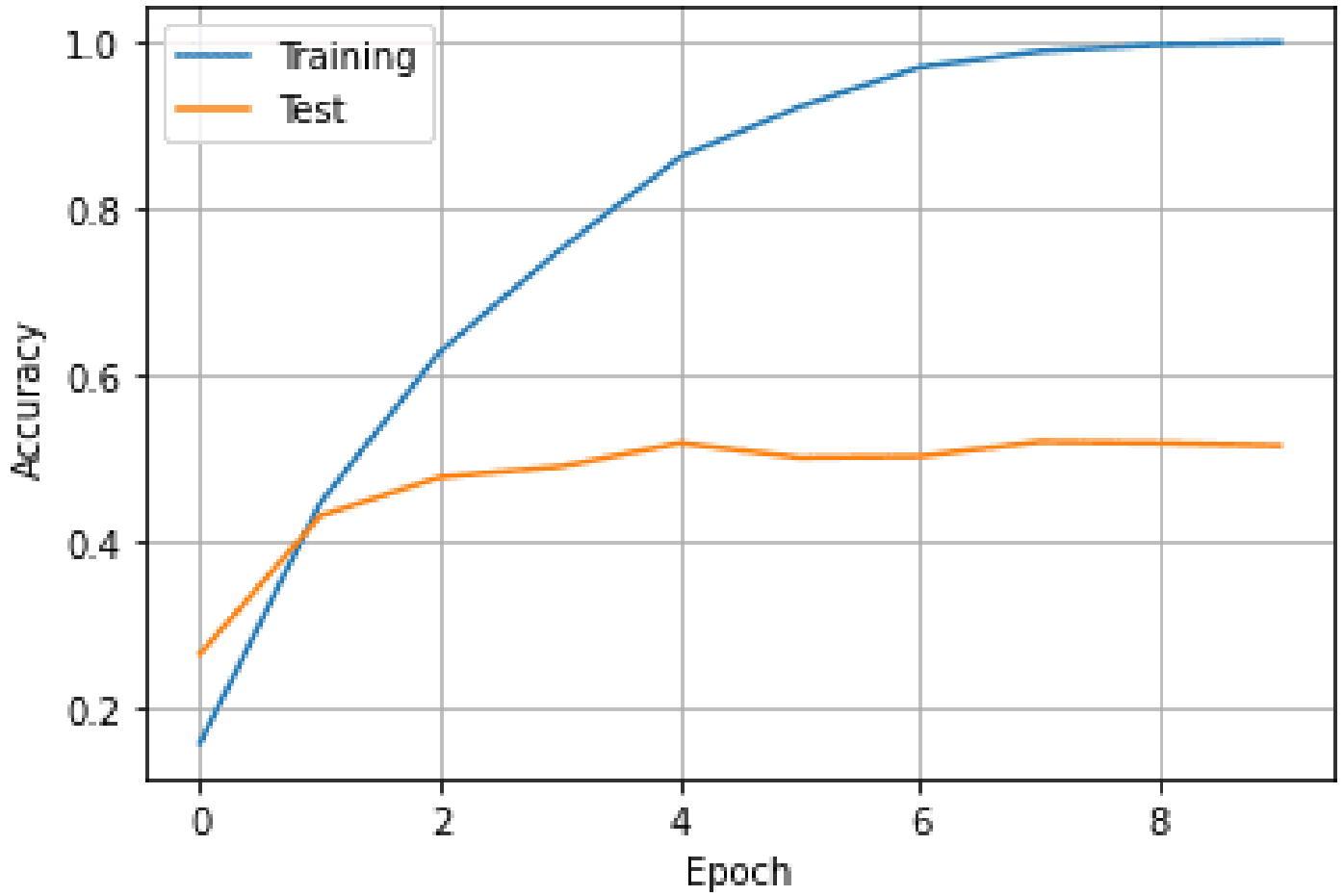
- `torchvision.datasets.ImageFolder`
- `tf.keras.preprocessing.image_dataset_from_directory`

Run training loop

- Train neural network for an epoch
- Evaluate on test dataset
- Train for several epochs

- Feel free to use training code from Learn Module
- Keras: `model.compile+model.fit`

Overfitting



50% Accuracy

Is it good?

[Optional] Top-k Accuracy

cat_Egyptian
cat_Maine
cat_Siamese
dog_Pekinese

Knowledge check

Question 1

What is a convolution layer?

A. A special activation function for images
B. An image preprocessing layer that normalizes and prepares image before the dense layer
C. A layer that runs a small windows across the image to extract patterns

Question 1

What is a convolution layer?

A. A special activation function for images
B. An image preprocessing layer that normalizes and prepares image before the dense layer
C. A layer that runs a small windows across the image to extract patterns

Question 2

How do the number of parameters in a convolutional layer and dense layer correlate?

A. A convolutional layer contains more parameters
B. A convolutional layer contains less parameters

Question 2

How do the number of parameters in a convolutional layer and dense layer correlate?

A. A convolutional layer contains more parameters
B. A convolutional layer contains less parameters

Question 3

If the size of an input color image is 200x200, what would be the size of the tensor after applying a 5x5 convolutional layer with 16 filters?

A. 16x196x196 (PT) or 196x196x16 (TF)
B. 3x196x196 (PT) or 196x196x3 (TF)
C. 16x3x200x200 (PT) or 200x200x16x3 (TF)
D. 48x200x200 (PT) or 200x200x48 (TF)

Question 3

If the size of color image is 200x200, what would be the size of the tensor after applying a 5x5 convolutional layer with 16 filters?

A. **16x196x196 (PT) or 196x16x16 (TF)**
B. 3x196x196 (PT) or 196x196x3 (TF)
C. 16x3x200x200 (PT) or 200x200x16x3 (TF)
D. 48x200x200 (PT) or 200x200x48 (TF)

Question 4

Which layers do we apply to significantly reduce spatial dimension in multi-layered CNN?

A. Convolution
B. Flatten
C. MaxPooling

Question 4

Which layers do we apply to significantly reduce spatial dimension in multi-layered CNN?

A. Convolution
B. Flatten
C. MaxPooling

Question 5

Which layer is used between convolutional base of the network and final linear classifier?

A. Convolution
B. Flatten
C. MaxPooling
D. Sigmoid

Question 5

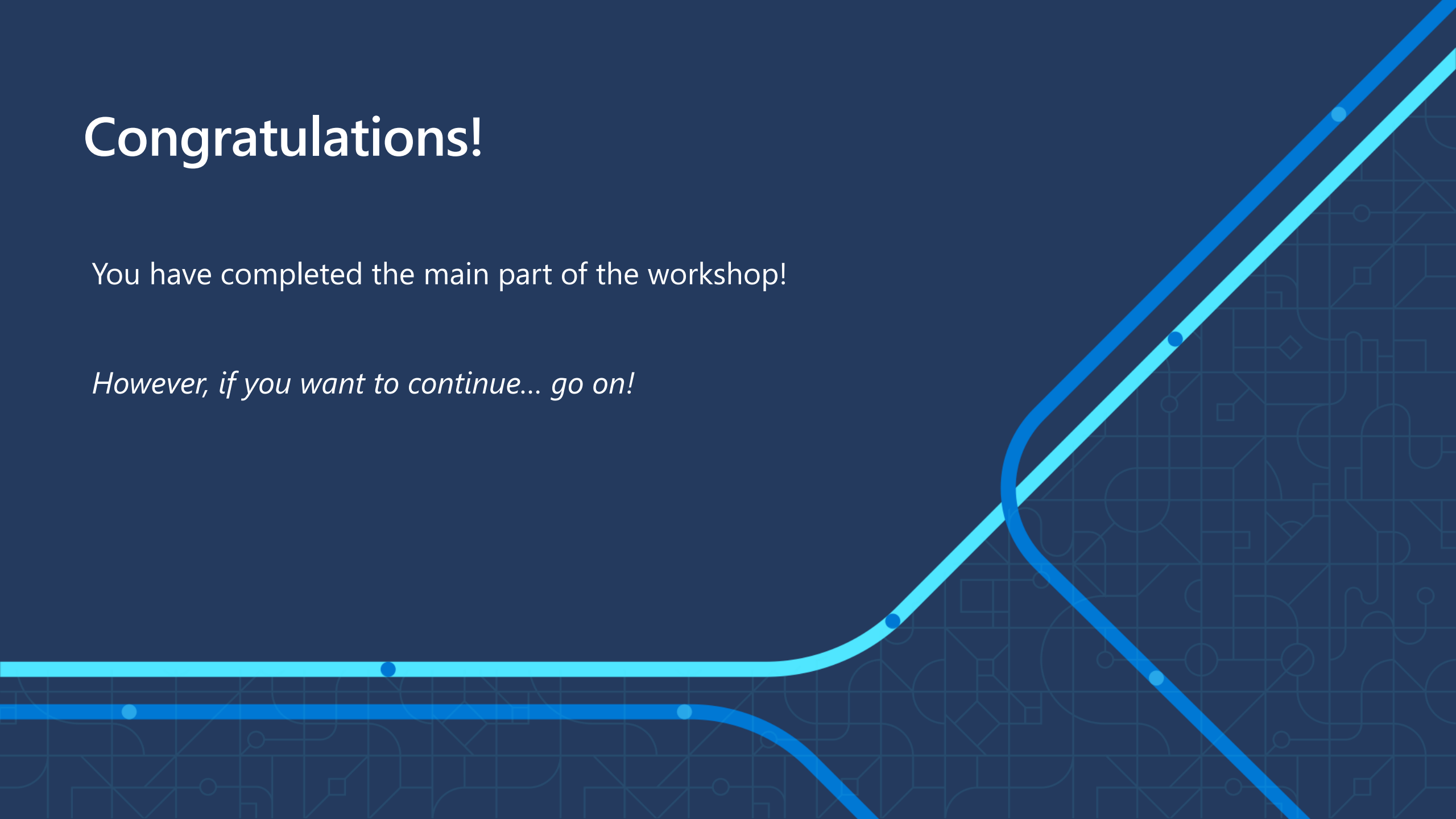
Which layer is used between convolutional base of the network and final linear classifier?

A. Convolution
B. Flatten
C. MaxPooling
D. Sigmoid

Congratulations!

You have completed the main part of the workshop!

However, if you want to continue... go on!

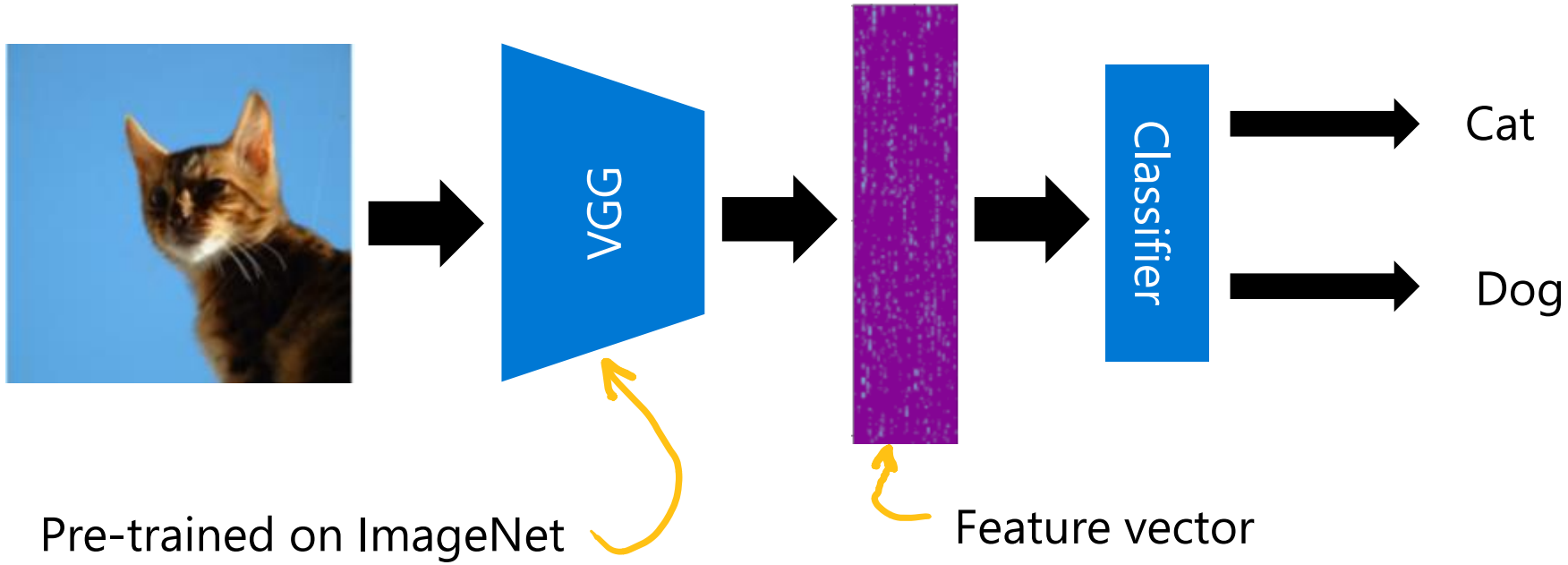


[Optional]
Project 2: Pet Face Recognition

Oxford Pets IIIT


```
!wget https://mslearntensorflowlp.blob.core.windows.net/data/oxpets_images.tar.gz
!tar xfz oxpets_images.tar.gz
!rm oxpets_images.tar.gz
```

Transfer Learning



Knowledge check

Question 1

For transfer learning, we are using a VGG-16 network pre-trained on 1000 classes. What is the number of classes we can have in our network?

A. Any
B. 1000
C. 2
D. less than 1000

Question 1

For transfer learning, we are using a VGG-16 network pre-trained on 1000 classes. What is the number of classes we can have in our network?

A. **Any**
B. 1000
C. 2
D. less than 1000

Summary and Further Steps

Wow!

We have learnt how to classify arbitrary breeds of cats and dogs with ~85% accuracy (~96% top-3) from 37 classes!

Next:

- Learn how to deploy the model on Azure Functions or Azure ML Cluster
- Create complete mobile application that can recognize breeds of cats/dogs:
  - Using Mobile-Net and local inference
  - Using model deployed on Azure
- Learn how to deal with text in PyTorch or TensorFlow