



LEARN

[Microsoft.com/Learn](https://Microsoft.com/Learn)



# 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:

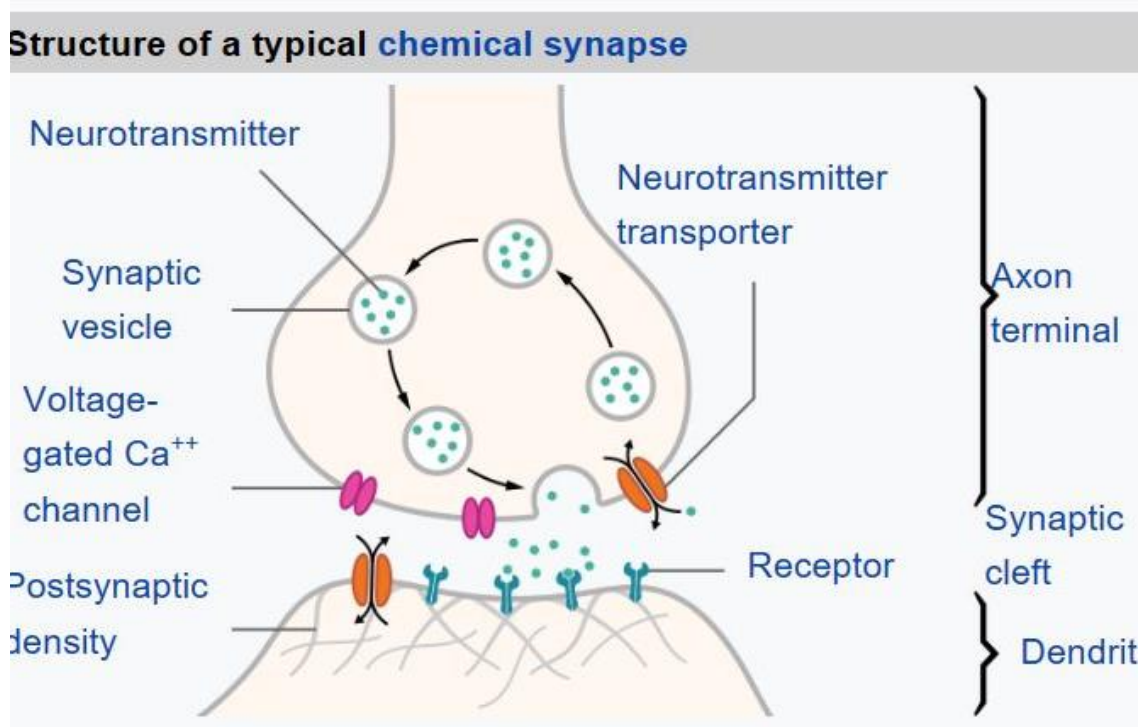
- Read: <http://eazify.net/nnintro>
- Introduction to PyTorch: <http://aka.ms/learntorch/intro>
- Introduction to TensorFlow: <http://aka.ms/learntf/keras>

# Introduction to Neural Networks

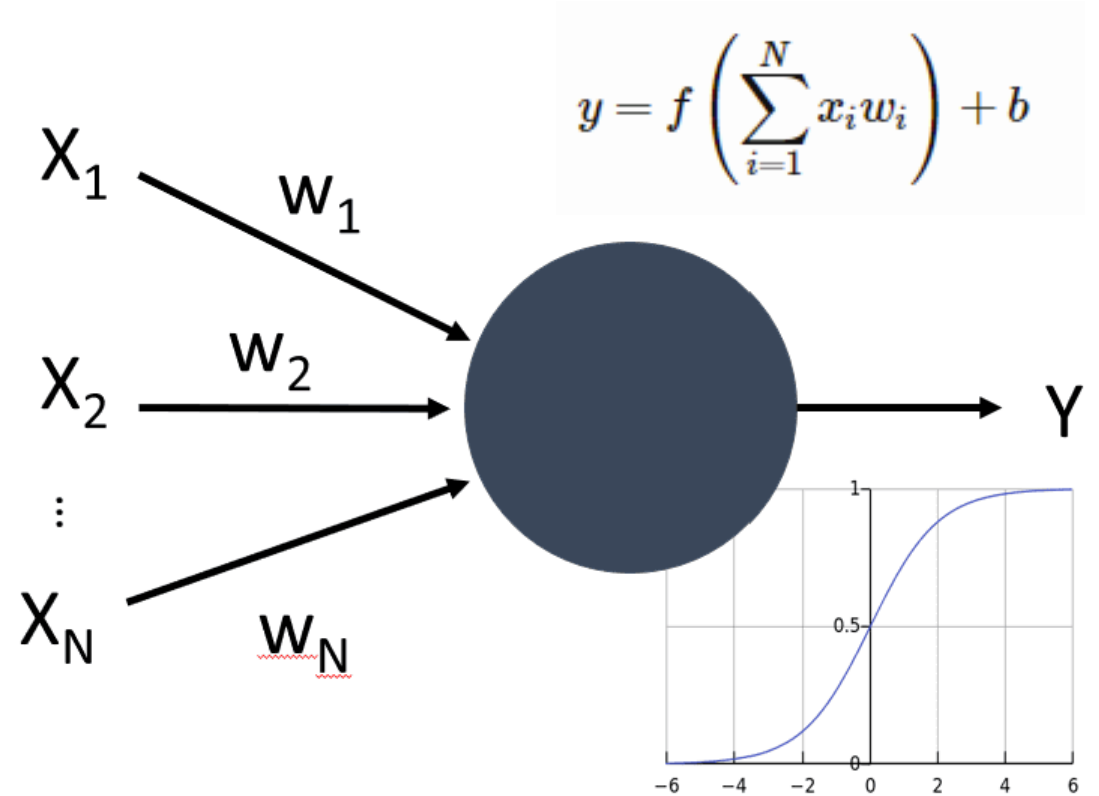
The background features a dark blue gradient with abstract, flowing lines in light blue and cyan. On the right side, there is a faint, repeating geometric pattern of squares, circles, and lines, reminiscent of a circuit board or a mathematical grid.

# Neural Networks are inspired by our Brain

<http://eazify.net/nnintro>

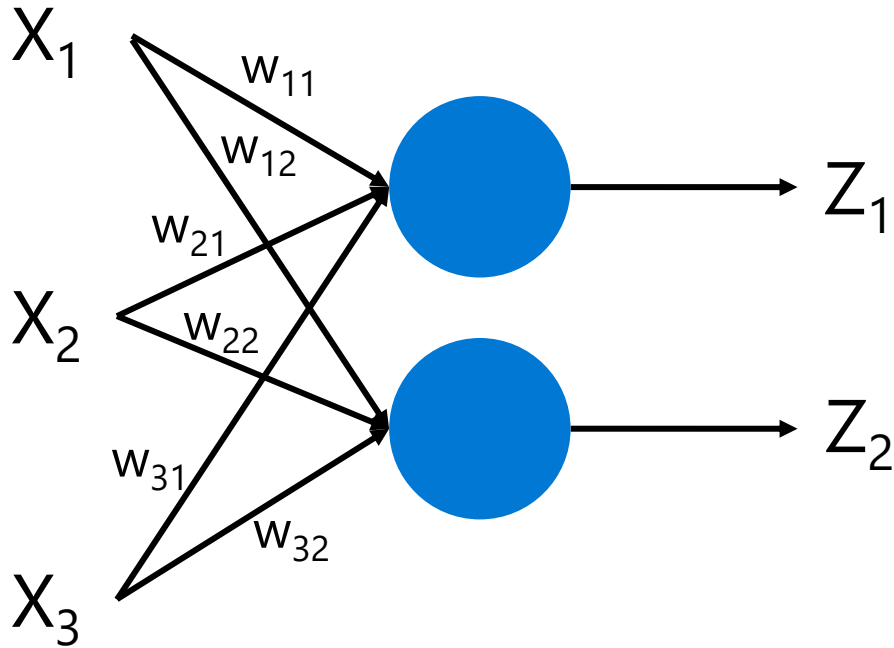


Real Neuron



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:  $Z - 2 \times 1$ ,  $W - 2 \times 3$ ,  $X - 3 \times 1$ ,  $b - 2 \times 1$

**Computing in minibatches (bs=9):**

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

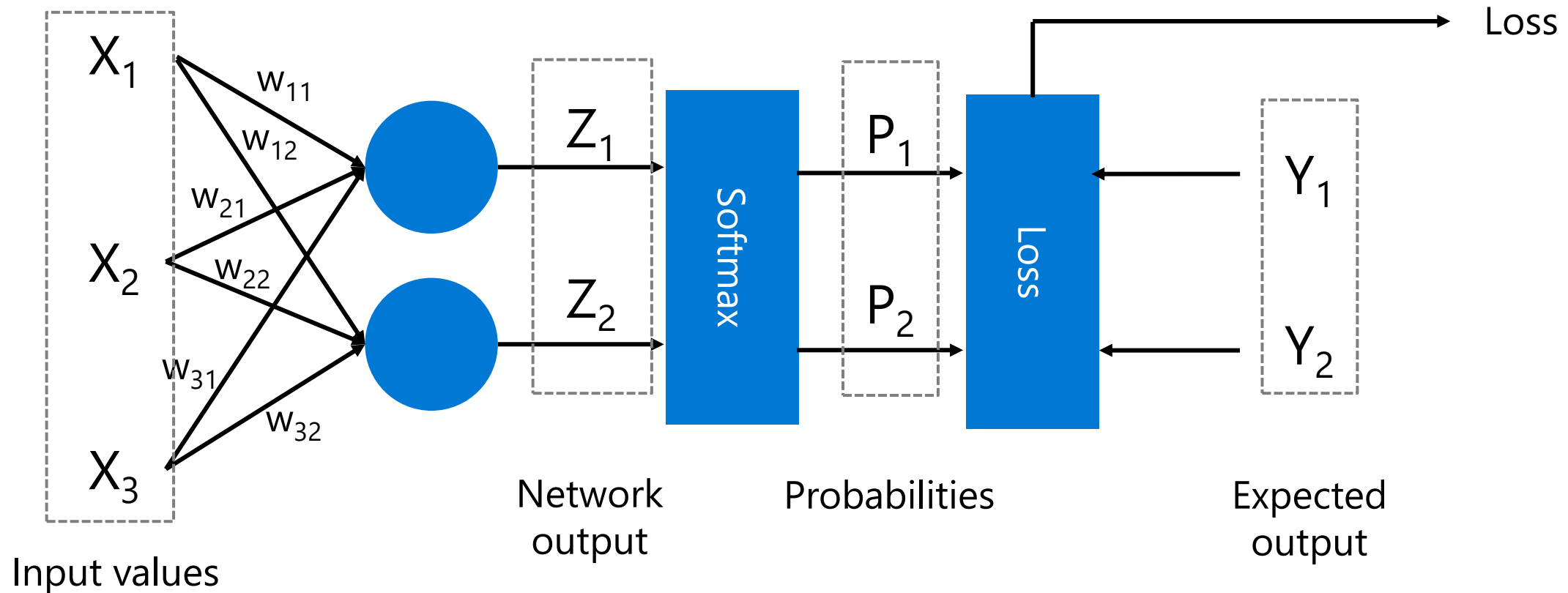
Sizes:  $Z - 9 \times 2 \times 1$ ,  $W - 2 \times 3$ ,  $X - 9 \times 3 \times 1$ ,  $b - 2 \times 1$



# Softmax and Loss

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

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



# 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!

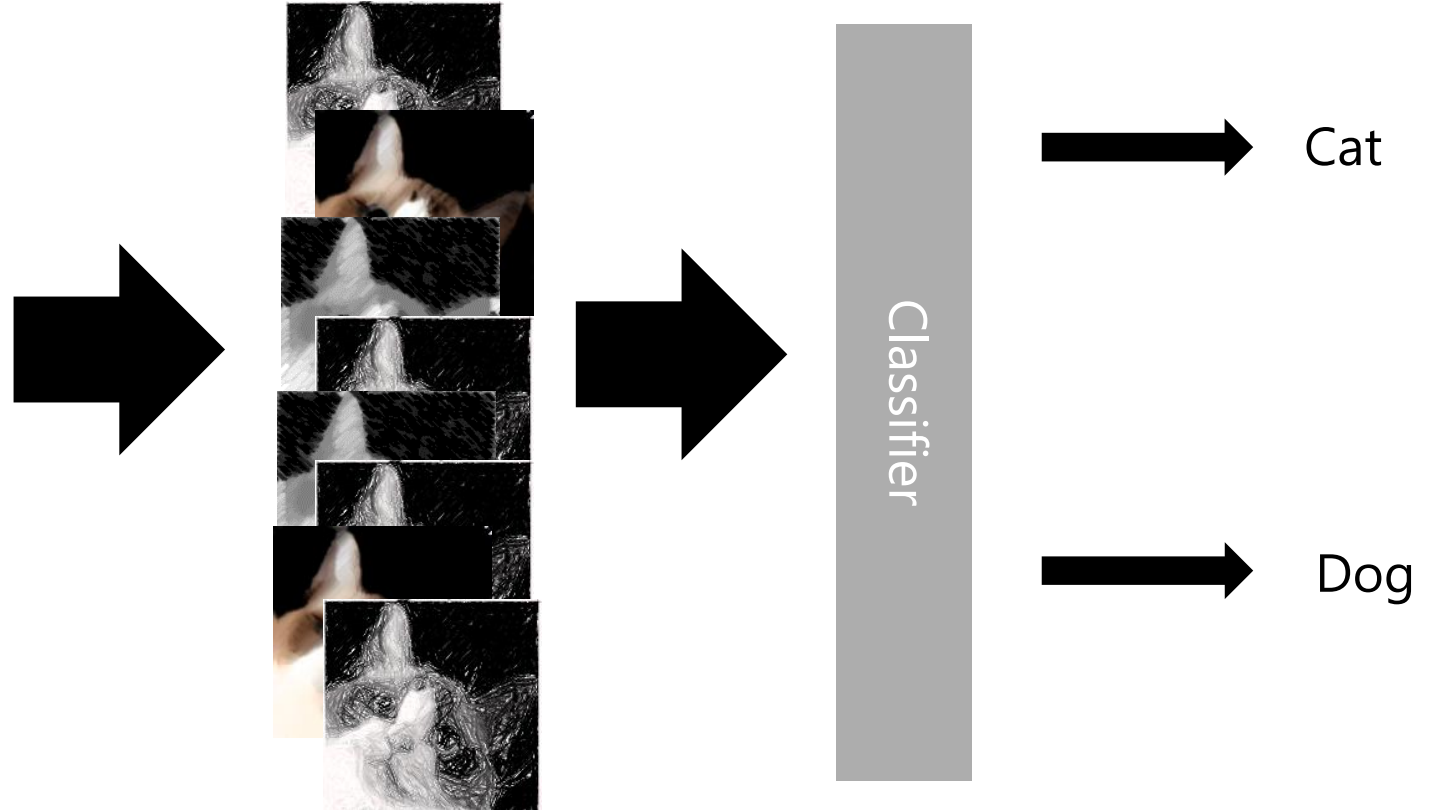
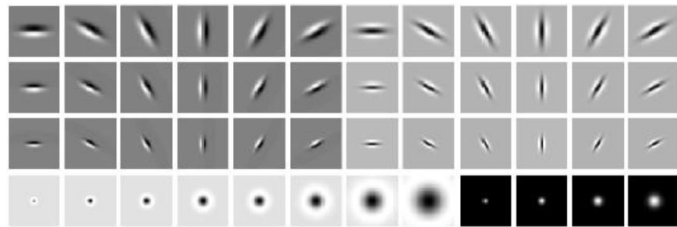
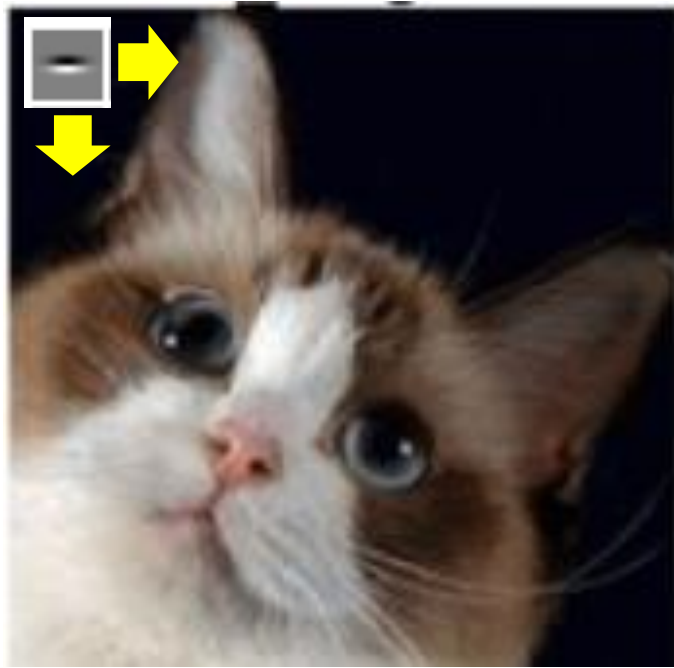


<https://aka.ms/learntf/vision>

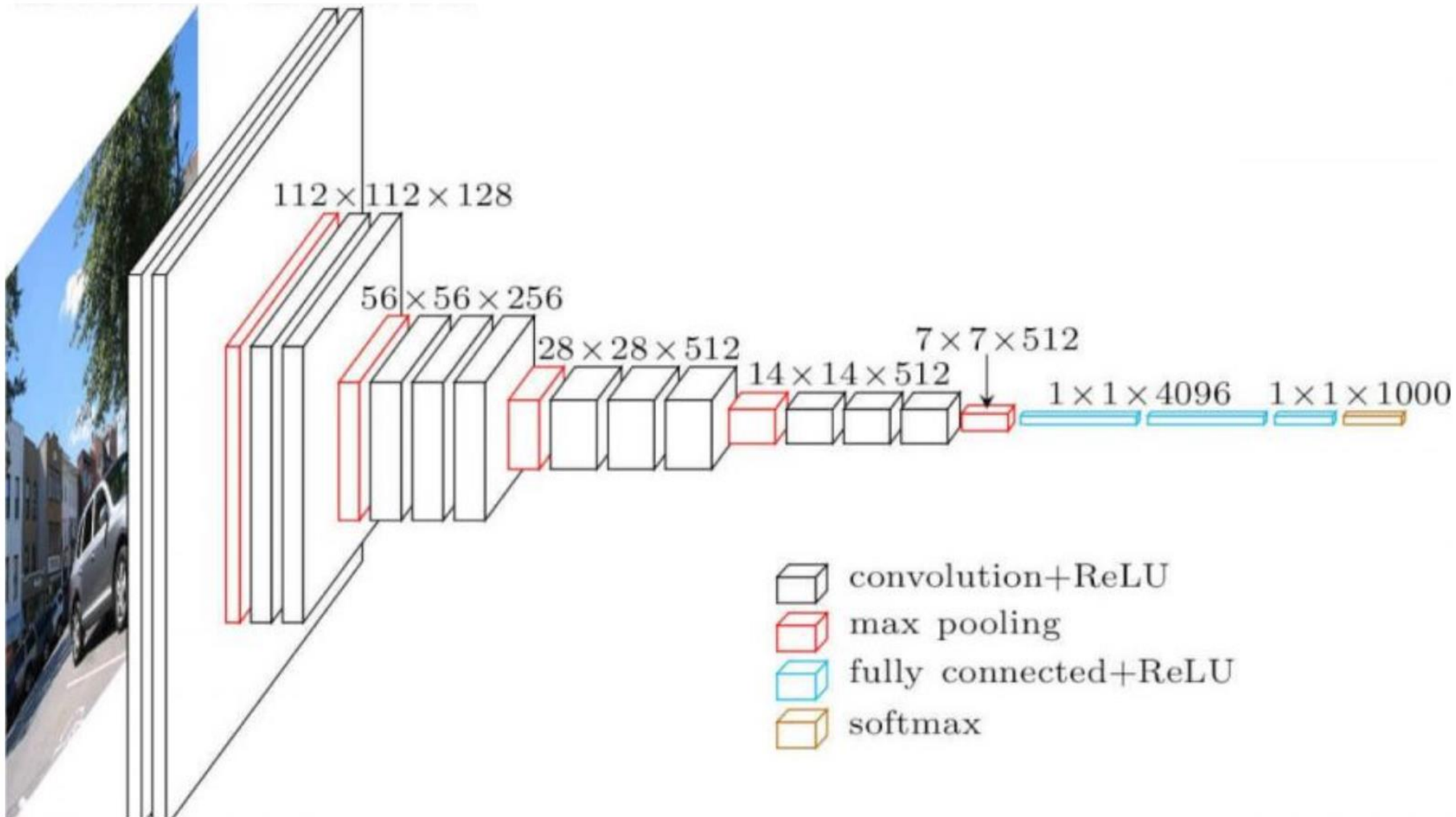


<https://aka.ms/learntorch/vision>

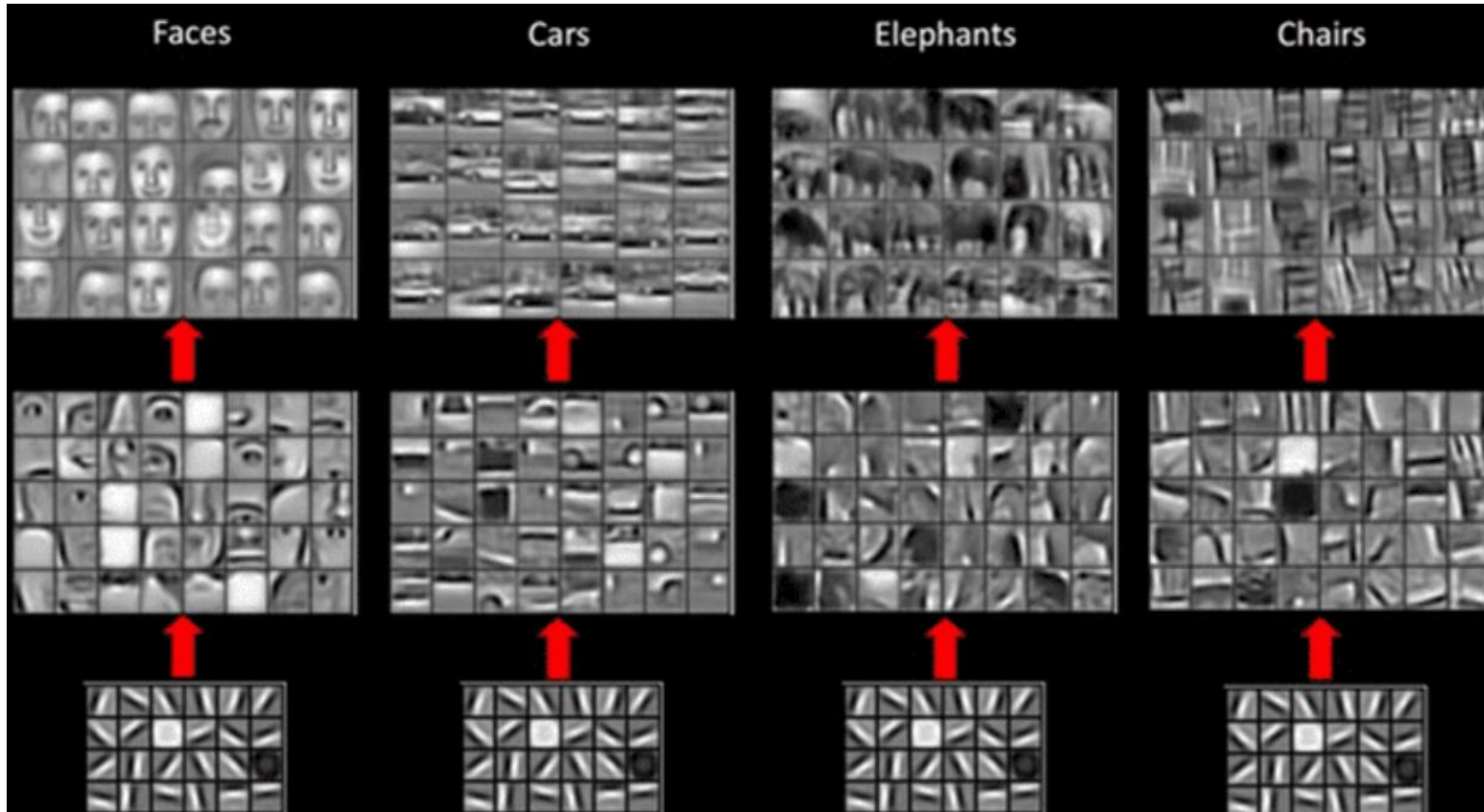
# Convolutional Neural Networks



# Pyramid Architecture



# Hierarchical Feature Extraction



# Project 1: Pet Face Recognition

The background is a dark blue gradient. It features several abstract geometric elements: a thick cyan line that runs horizontally across the lower third of the image and then curves upwards and to the right; a thinner blue line that runs horizontally just below the cyan one and then curves downwards and to the right; and a diagonal line in the upper right corner composed of a blue line and a cyan line. Small dots in blue and cyan are scattered along these lines. Faint, light-blue geometric patterns, including squares, circles, and lines, are visible in the background.



# Project 1: Pet Face Recognition

cat\_Ragdoll



dog\_boxer



dog\_wheaten



dog\_great



dog\_keeshond



cat\_Maine



dog\_great



cat\_Ragdoll



dog\_newfoundland



dog\_newfoundland



cat\_Egyptian



cat\_Abyssinian



dog\_japanese



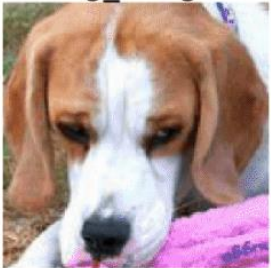
cat\_Siamese



dog\_samoyed



dog\_beagle



dog\_newfoundland



dog\_miniature



cat\_Maine



dog\_american



dog\_shiba





# 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

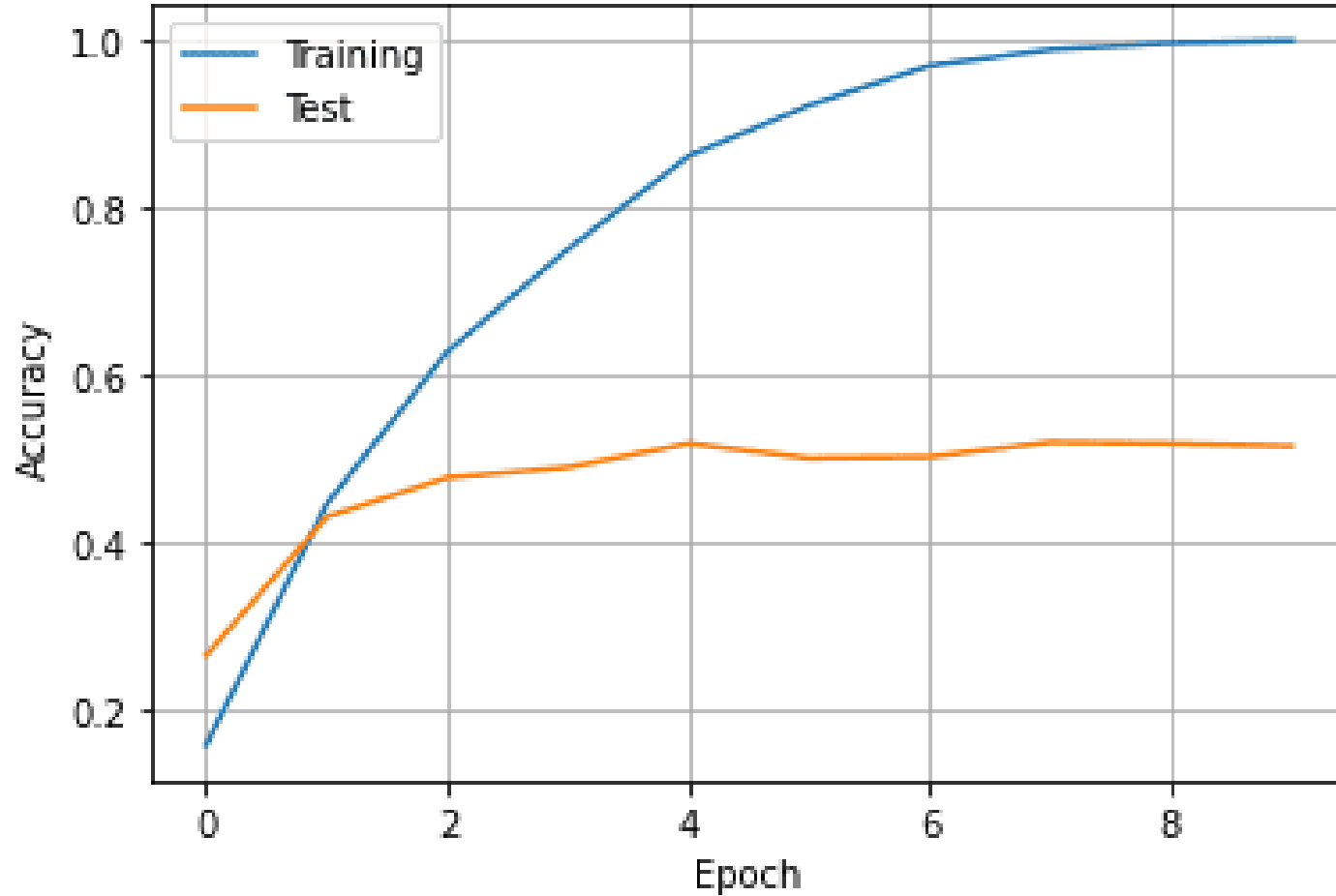
## Load data into tensors

- 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`

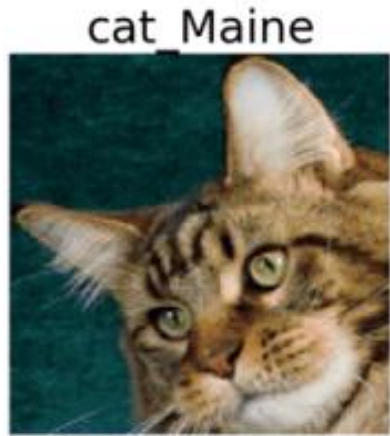
# Overfitting



**50% Accuracy**

**Is it good?**

# [Optional] Top-k Accuracy



cat\_Egyptian  
cat\_Maine  
cat\_Siamese  
dog\_Pekinese

# Knowledge check

The background is a dark blue gradient. On the right side, there are several thick, curved lines in shades of blue and cyan. These lines intersect and curve in various directions, creating a dynamic, abstract shape. Small, semi-transparent dots in matching colors are placed along these lines. The bottom right corner features a faint, repeating pattern of geometric shapes like squares, circles, and lines, creating a textured effect.

# 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  $200 \times 200$ , what would be the size of the tensor after applying a  $5 \times 5$  convolutional layer with 16 filters?

- A.  $16 \times 196 \times 196$  (PT) or  $196 \times 196 \times 16$  (TF)
- B.  $3 \times 196 \times 196$  (PT) or  $196 \times 196 \times 3$  (TF)
- C.  $16 \times 3 \times 200 \times 200$  (PT) or  $200 \times 200 \times 16 \times 3$  (TF)
- D.  $48 \times 200 \times 200$  (PT) or  $200 \times 200 \times 48$  (TF)

## Question 3

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

- A.  **$16 \times 196 \times 196$  (PT) or  $196 \times 16 \times 16$  (TF)**
- B.  $3 \times 196 \times 196$  (PT) or  $196 \times 196 \times 3$  (TF)
- C.  $16 \times 3 \times 200 \times 200$  (PT) or  $200 \times 200 \times 16 \times 3$  (TF)
- D.  $48 \times 200 \times 200$  (PT) or  $200 \times 200 \times 48$  (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

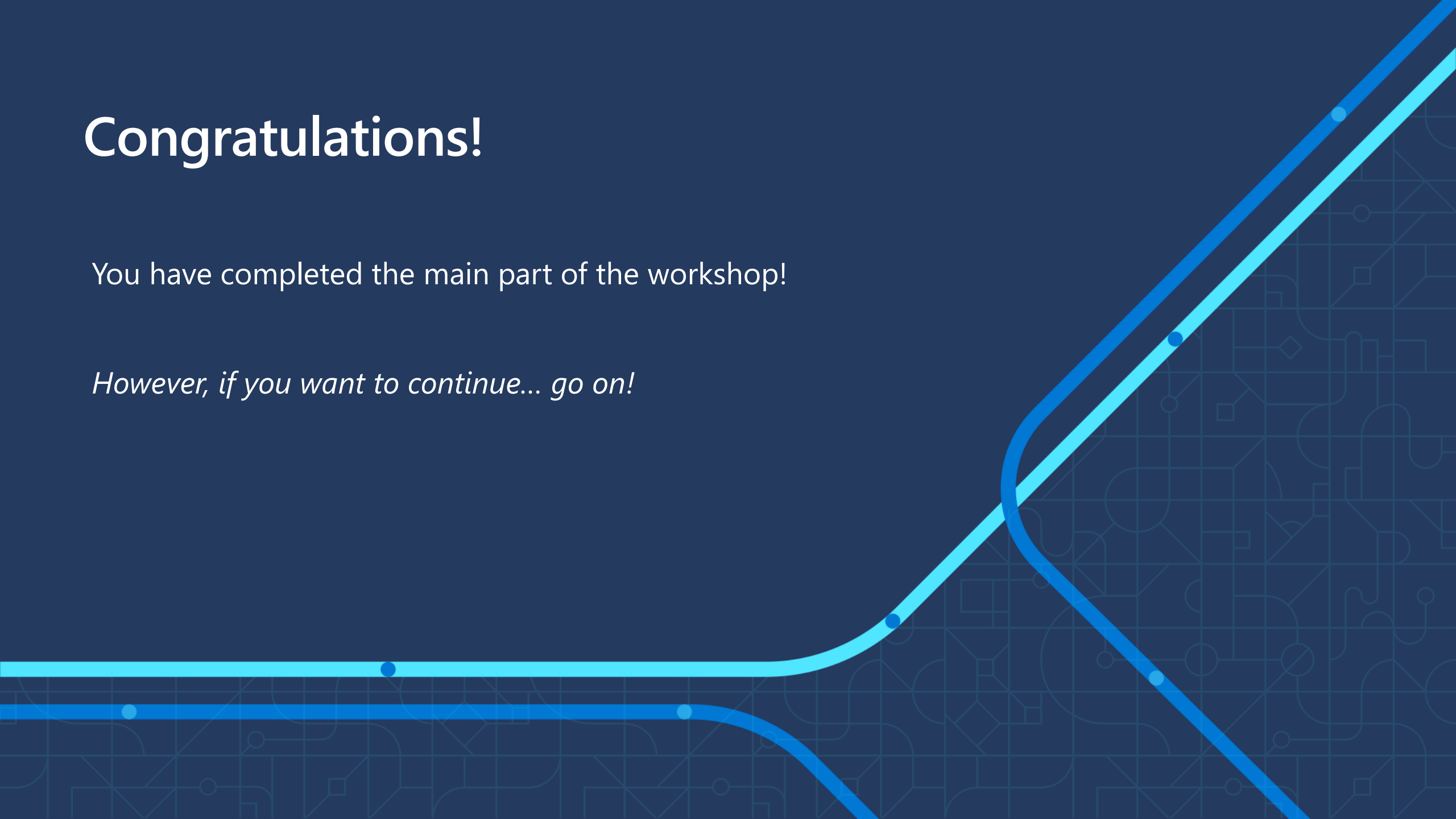
Which layer is used between convolutional base of the network and final linear classifier?

- 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

The background is a dark blue gradient. It features several abstract geometric elements: a thick cyan line that runs horizontally across the lower third of the image and then curves upwards and to the right; a thinner blue line that runs diagonally from the bottom left towards the top right; and another blue line that runs diagonally from the top right towards the bottom left. There are also several small, light blue dots scattered throughout the composition, some of which are positioned along the lines.

# Oxford Pets IIIT

benga



siames



boxe



great



benga



persia



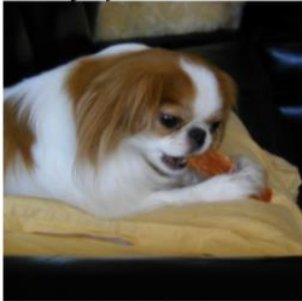
english



scottish



japanese



american



wheaten



chihuahu



boxe

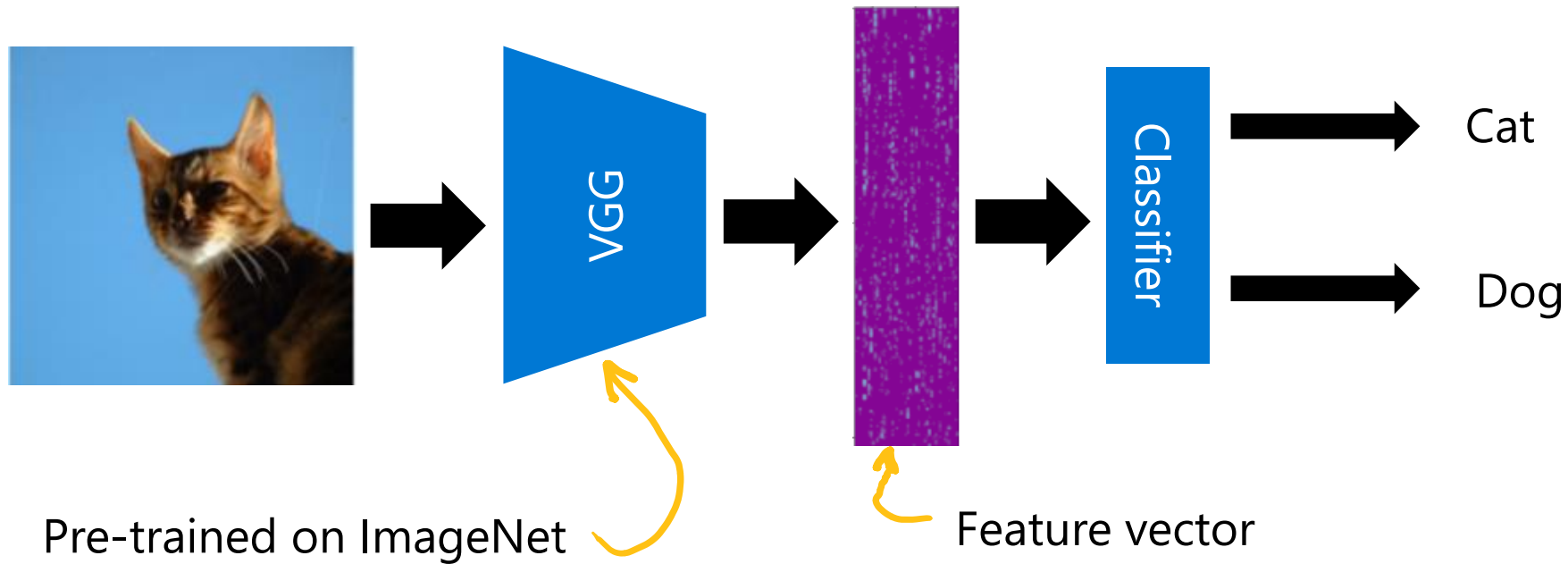


leonberge



```
!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

The background is a dark blue gradient. On the right side, there are several thick, curved lines in shades of blue and cyan. These lines intersect and curve in various directions, creating a dynamic, abstract shape. Small, semi-transparent dots in matching colors are placed along these lines. The bottom right corner features a faint, repeating pattern of geometric shapes like squares, circles, and lines, creating a textured effect.

# 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

The background is a dark blue gradient. On the right side, there are several overlapping lines in shades of blue and cyan. These lines start horizontally from the left and curve upwards and to the right. Small dots in matching colors are placed along these lines. The right side of the image is filled with a faint, light-blue geometric pattern consisting of various shapes like squares, circles, and lines, creating a technical or architectural feel.

# 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](#)



