# Lecture 5
# Transfer Learning and GAN

## CS 180 – Intelligent Systems

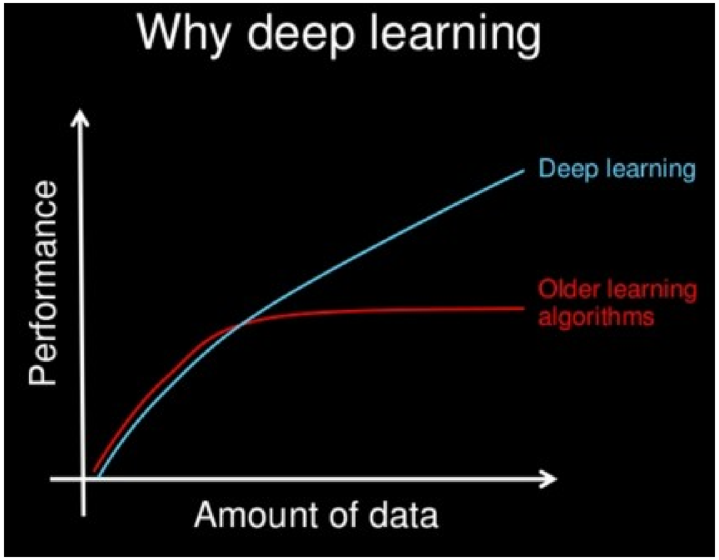
**Dr. Victor Chen**

**Spring 2021**

# Transfer learning

- A large amount of data is required to train good deep learning models

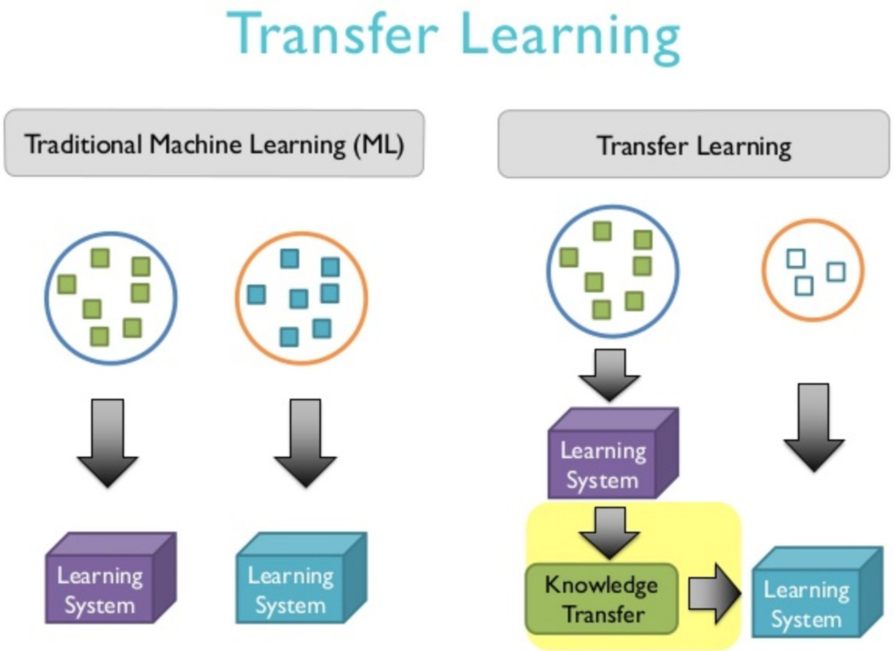
- What if you don't have much data on your own task?



# Transfer learning

# Transfer learning

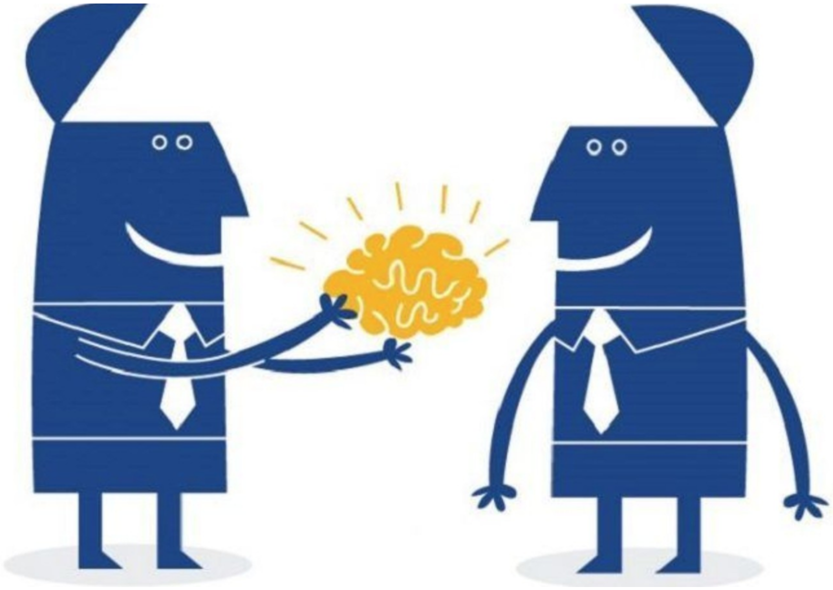
- If you don't have much data on your own task, find a similar task with much data and use transfer learning



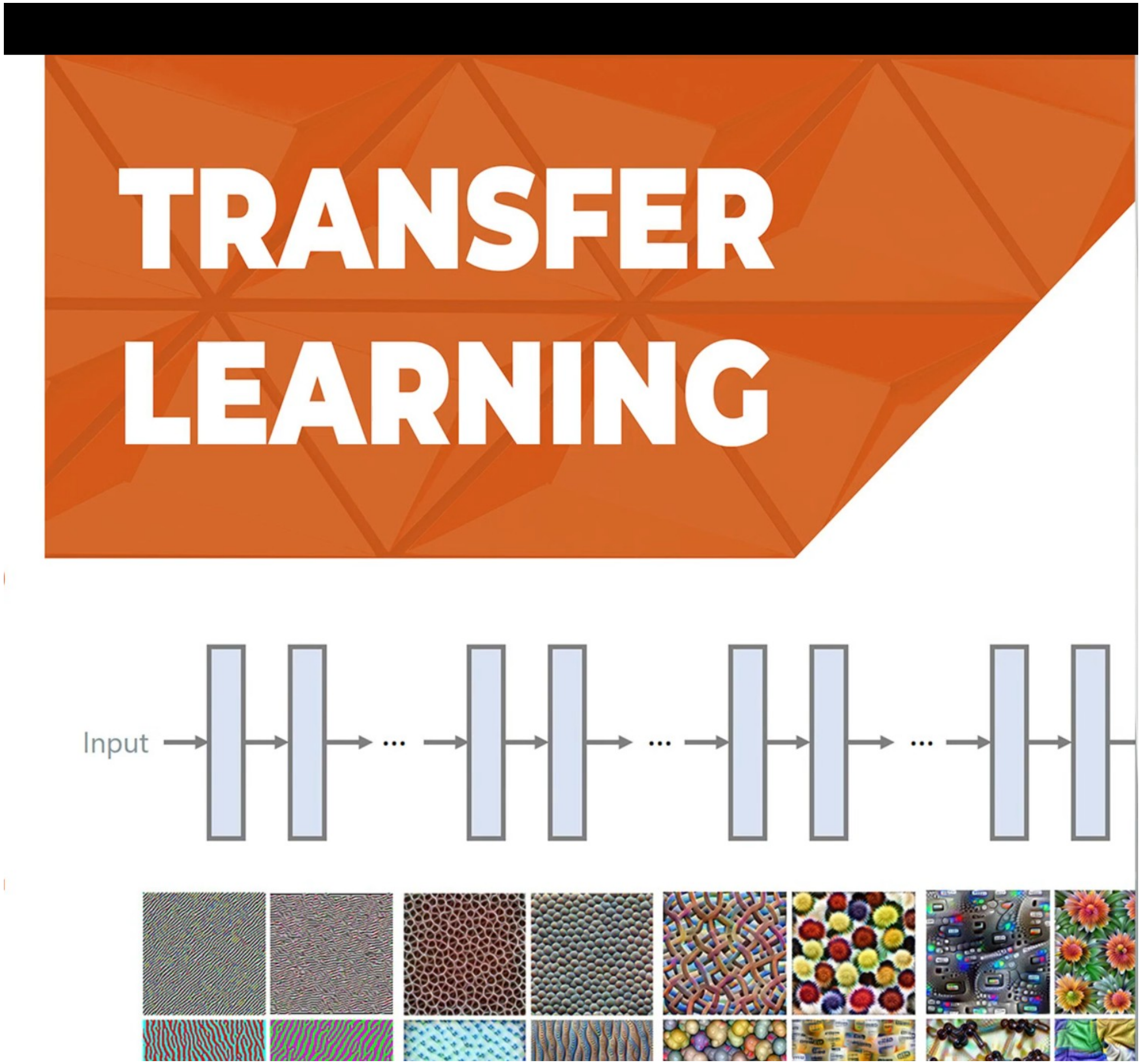
- In transfer learning, we reuse a model trained on one task for another task.

# Two basic steps in transfer learning:

- Identify a pre-trained model on a different but similar task.
- Fine-tune the pre-trained model using your own data.
  - We refine the pre-trained model to make sure it works for your exact problem



# Transfer learning video



# How to find pre-trained models

- Many research institutions release models pre-trained on large datasets. These models took days or weeks to train on modern hardware.

- Pre-trained models on image classification problem
  - Oxford VGG Model
  - Google Inception Model
  - Microsoft ResNet Model
- Pre-trained models on natural language processing
  - Google's word2vec Model
  - Doc2vec model
  - Stanford's GloVe Model
  - ELMo
  - BERT

# Pre-trained models in Tensorflow/Keras

https://keras.io/applications/

**Models for image classification with weights trained on ImageNet (**http://image-net.org/index**):**

Xception

VGG16

VGG19

ResNet, ResNetV2, ResNeXt

InceptionV3

InceptionResNetV2

MobileNet

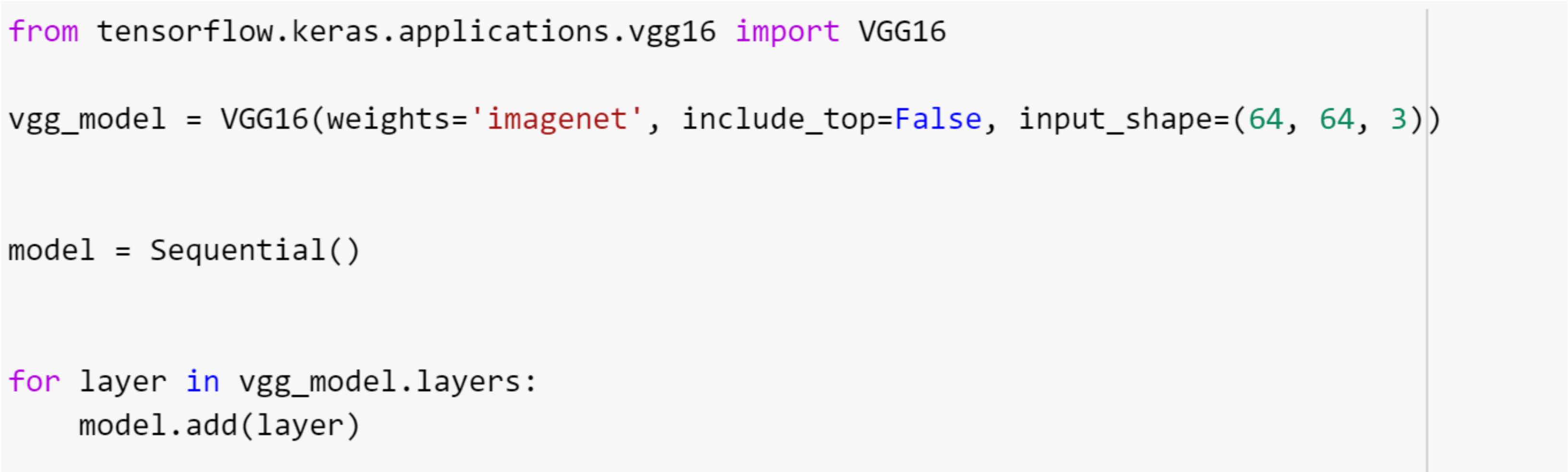
MobileNetV2

DenseNet

NASNet

# Transfer learning in Tensorflow

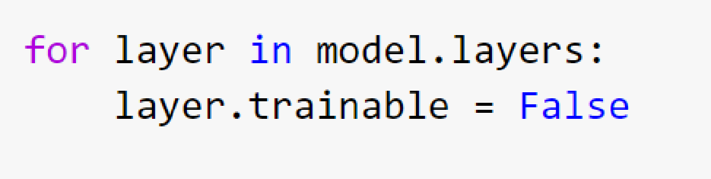
## Step 1:  Load a pre-trained model (except the top layer)



```
from tensorflow.keras.applications.vgg16 import VGG16

vgg_model = VGG16(weights='imagenet', include_top=False, input_shape=(64, 64, 3))


model = Sequential()


for layer in vgg_model.layers:
    model.add(layer)
```

# Transfer learning in Tensorflow

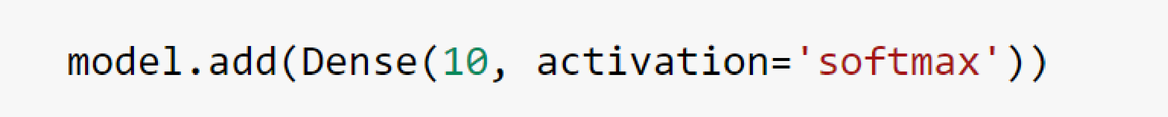
## Step 2:  Fix the weights from the pre-trained model



```
for layer in model.layers:
    layer.trainable = False
```

# Transfer learning in Tensorflow

Step 3: Add other layers on top of the pre-trained model



```
model.add(Dense(10, activation='softmax'))
```

# Takeaway

- Transfer learning is a shortcut to saving time or getting better performance

- Use transfer learning if
  - You don't have much data on your own task but you can identify a related task and there is a pre-trained model available for that task.

- Use transfer learning for your final project, which will boost the model performance and save training time

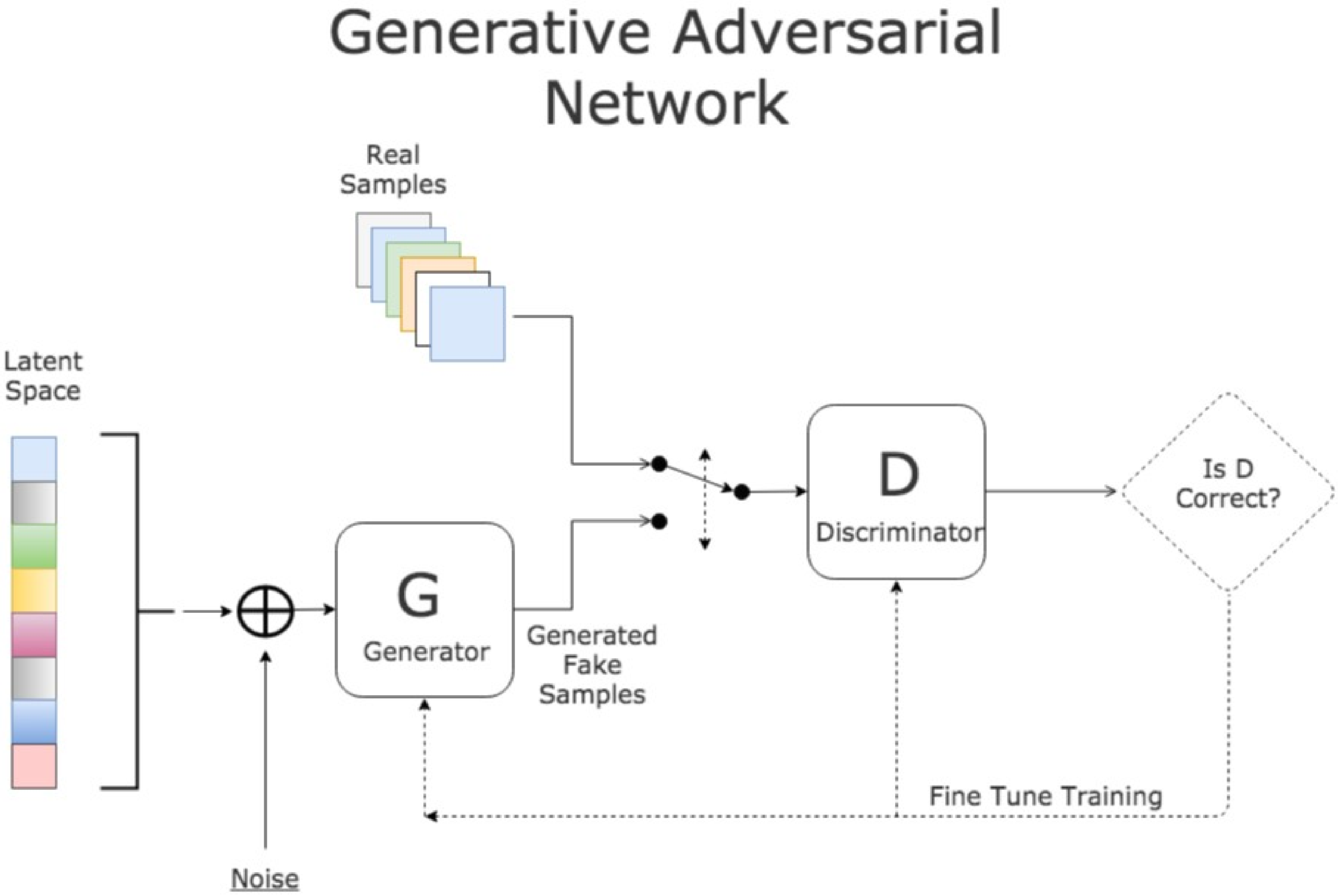
# GAN (Generative Adversarial Networks)

**Discriminative algorithms** try to classify input data; that is, given the features of an instance of data, they predict a label or category to which that data belongs.

**Generative algorithms** generate new, synthetic data that mimic real data. They are used in image generation, video generation and voice generation.

While discriminative models care about the relation between y and x, generative models care about "how you get x."

# GAN(Generative Adversarial Networks)



# GAN Video

# DeepFake

https://www.cnn.com/interactive/2019/01/business/pentagons-race-against-deepfakes/

# The Next Rembrandt

www.nextrembrandt.com