

■ How do zero-shot, one-shot and few-shot learning differ?

Zero-Shot Learning (1)

- **Can you classify an object without seeing it before?**
 - How a child would have no problem recognizing a zebra? if it has seen a horse before and read somewhere that a zebra looks similar to a horse but has black-and-white stripes.
-
- **Yes**, you can if you have adequate information about its appearance, properties, and functionality.
 - Think back to how you came to understand the world as a kid.
 - You could spot Mars in the night sky after reading about its color and where it would be that night, or identify the constellation Cassiopeia from only being told: “it’s basically a malformed ‘W’.”



https://line.17qq.com/articles/esrqcqrqx_p3.html

Zero-shot learning (2)

- [Zero-shot learning](#) is the challenge of learning modelling **without using data labelling**.
- Zero-shot learning **involves little human intervention**, and the **models depend on previously trained concepts and additional existing data**.
- This method reduces the time and effort that data labelling takes.
- Instead of giving training examples, **zero-shot learning gives a high-level description of new categories** so that the machine can relate it to existing categories that the machine has learned about.
- Zero-shot learning methods can be used in computer vision, natural language processing and machine perception.

Zero-shot learning (3)

- Zero-shot methods basically work by **combining** the **observed/seen** and **non-observed/unseen** categories through **some types of auxiliary information**, which encodes observable distinguishing properties of objects.
- Without accessing any data of the unseen categories during training the models, yet it is able to **build and train models with the help of transferring intelligence from previously seen categories and auxiliary information**.
- **For example**, given a **set of images of animal classification** use case, along with **auxiliary textual descriptions about animals look like**, an Artificial Intelligence (AI) model which has been trained to identify horses, but has never seen a zebra, can still identify a zebra if it also knows that zebras look like striped horses. The auxiliary information may include **attributes, textual descriptions, or vectors of word category labels**. This type of use case is majorly studied in computer vision (CV), natural language processing (NLP), and machine perception.

Zero-shot learning (4)

- **Zero-shot learning is done in two stages:**
 1. **Training:** Where the knowledge about the attributes is captured
 2. **Inference:** The knowledge is then used to categorize instances among a new set of classes.
- In training, the intermediate layer of semantic attributes are captured, then in the inference stage, this knowledge is used to predict categories among a new set of classes.
- At this level, the second layer models the relationship between the attributes and the classes and fixes the categories using the initial attribute signature of the classes.
- **For example,** if a child is asked to recognize a Yorkshire terrier, he or she may know it is a type of a dog, with added information about it from Wikipedia.

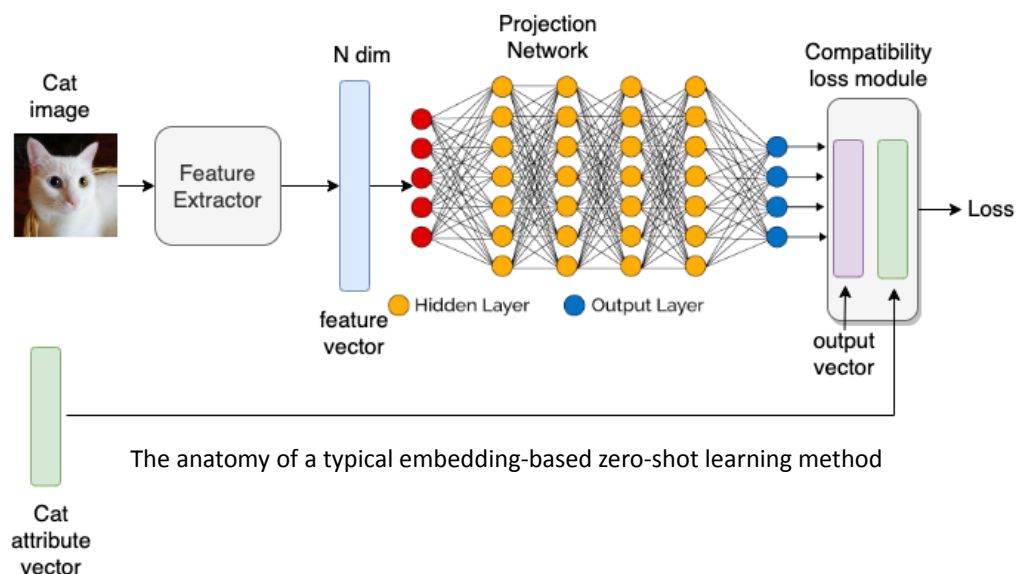
Zero-shot learning (5): How Zero-shot Learning works?

Two common approaches used to solve the zero-shot recognition problems:

- **1. Embedding based approach**
- **2. Generative model-based approach**

Zero-shot learning (6): Embedding-based approach

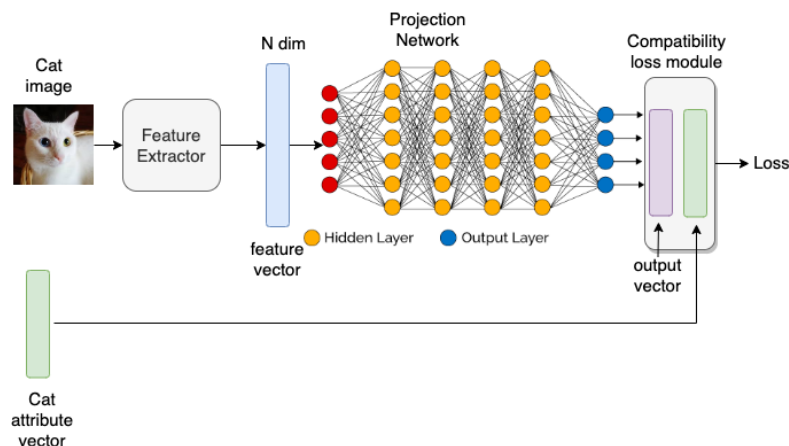
- This **maps the semantic attributes with the image features into a common embedding space**.
- It uses **a projection function** that it has learned from deep networks.
- The model **aims to use data from seen categories during training to find a projection function from visual to semantic space** (that is, word vectors or semantic embedding) using information from seen categories.
- Since neural networks are used as function approximators, the projection function is learned as a deep neural network.
- The projection function is learned as a deep neural network as neural networks are used as function approximators.



- In the testing phase, the non-observed category image feature data are put as input to the trained model, and we get the relative semantic embedding as a result.
- After which, to do classification, we **try with nearest neighbor search in the semantic attribute space to find the best similar result to the output of the network**.
- Finally, the category corresponding to **the nearest semantic embedding** is predicted as the final category of the input image feature.

Zero-shot learning (7): Embedding-based approach

- The anatomy of a typical embedding-based zero-shot learning method.
- The input image is initially passed through a feature extractor network (deep neural network (DNN)) to get an N-dimensional feature vector for the image.
- This vector acts as the input to the main network, which returns the result of a D-dimensional output vector.
- The **end goal is to calculate the weights of the projection network so as to map the N-dimensional input to a D-dimensional output.**
- To get this, **we put a loss that measures the compatibility between the D-dimensional output and ground truth semantic attribute.**
- The weights of the network are trained such that the D-dimensional output is as close as possible to the ground truth data.

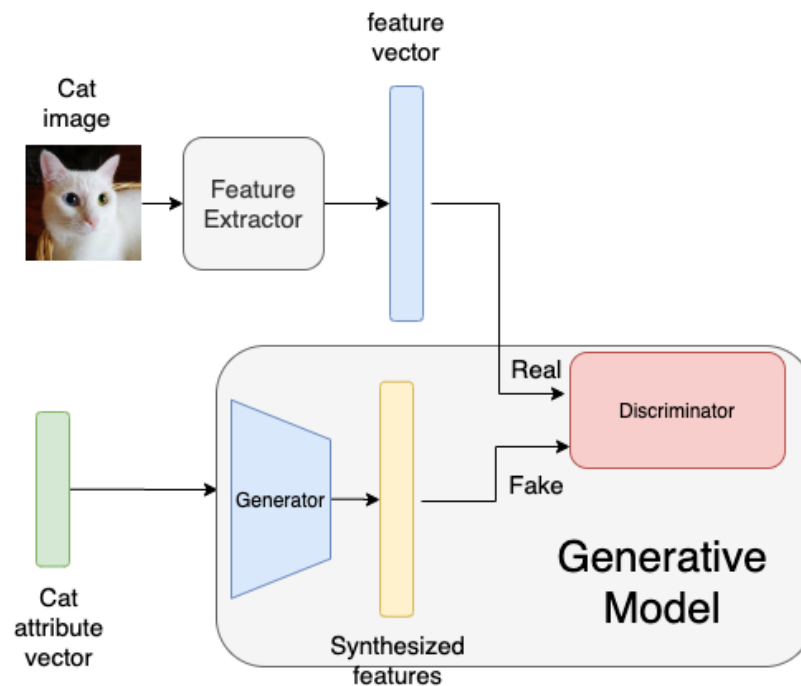


Zero-shot learning (8): Generative model-based approach

- This approach aims to generate image features for unseen categories using semantic attributes. This approach tackles the bias and domain shift that the embedding-based approach has.
- **The main drawback with embedding-based methods is that they suffer from the issue of bias and domain shift.**
- This means that **since the projection function is learned using only seen classes during training**, it will be biased towards predicting seen category labels as a result.
- There is also no surety that the trained projection function will rightly map non-observed category image features to the corresponding semantic space correctly at the testing phase.
- This is due to the fact that the deep network has only learned to map seen category image features to semantic space during training and might not be able to do the same for the novel non-observed category at the testing phase correctly.
- **To be able to overcome this drawback, it is important that our zero-shot classification model is trained on both seen and non-observed category images at train time. This is where this type of models-based method is used.**
- This approach aims to generate image features for unseen categories using semantic attributes. **This approach tackles the bias and domain shift that the embedding-based approach has.**

Zero-shot learning (9): Generative model-based approach

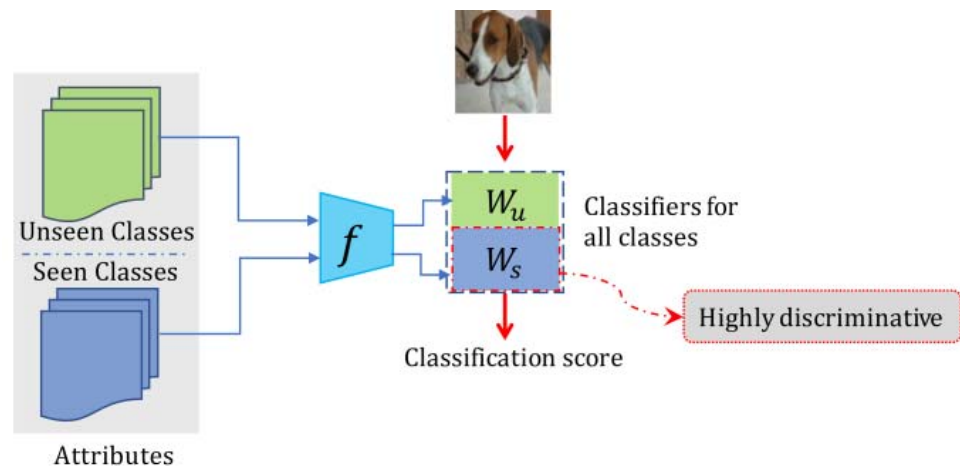
- The generative method's goal is to generate image features for non-observed categories using semantic attributes.
- Generally, this is done using a conditional generative adversarial network (cGAN) that generates image features conditioned on the semantic attribute of a given category.
- The diagram of a general generative model-based zero-shot learning:
 - Identical to the embedding-based method, we use a feature extractor network to get an N-dimensional feature vector.
 - First, the attribute vector is input to the generative model as displayed in the diagram.
 - The generator generates an N-dimensional output vector conditioned on the attribute vector.
 - The generative model is trained such that the synthesized feature vector looks identical to the original N-dimensional feature vector.



The diagram of a general generative model-based zero-shot learning

Zero-shot learning (10): Generative model-based approach

- When the generative model is modeled, we fix the weights of the generator and pass the class attributes as input to it to generate non-observed category image features.
- Once we have seen class image features (the training dataset) and non-observed category image features, we can train a basic image classifier that takes images as the input features and outputs the respective category label as shown in the figure bellow.



Zero-shot learning (11): Evaluation Metric for Zero-shot learning methods

- Generally, image recognition/image classification models use **Top-1 accuracy as their evaluation metric**.
- But, the evaluation metric used for zero-shot learned recognition models is different from that used for vanilla image classification models.
- We use **the average per category Top-1 accuracy to evaluate zero-shot recognition results**.
- Mathematically, for a set of classes Y with N classes, the average per class top one accuracy is given by

$$acc_Y = \frac{1}{N} \sum_{c=1}^N \frac{\text{\# of correct predictions in } c}{\text{\# of samples in } c}$$

- Then we calculate the accuracy for each class individually and then average it across all other categories.
- This encourages high performance in both sparsely and densely populated classes.
- In the method of a generalized zero-shot setting, our goal is for high accuracy on both seen classes as well as a set of non-observed categories.
- Thus the performance metric is defined as the harmonic mean of performance on seen classes and non-observed categories

Zero-shot learning (12): Implementation

- Use *huggingface* to implement zero-shot learning as it's easy to use and supports numerous NLP tasks.
- The *huggingface* libraries also made available its zero-shot classification pipeline with the capabilities to perform **text classification**, **sentiment classification**, and **topic modeling** *without the necessity of having any labeled data or training*.
- <https://www.kdnuggets.com/2021/04/zero-shot-learning.html>

Zero-shot learning (13)

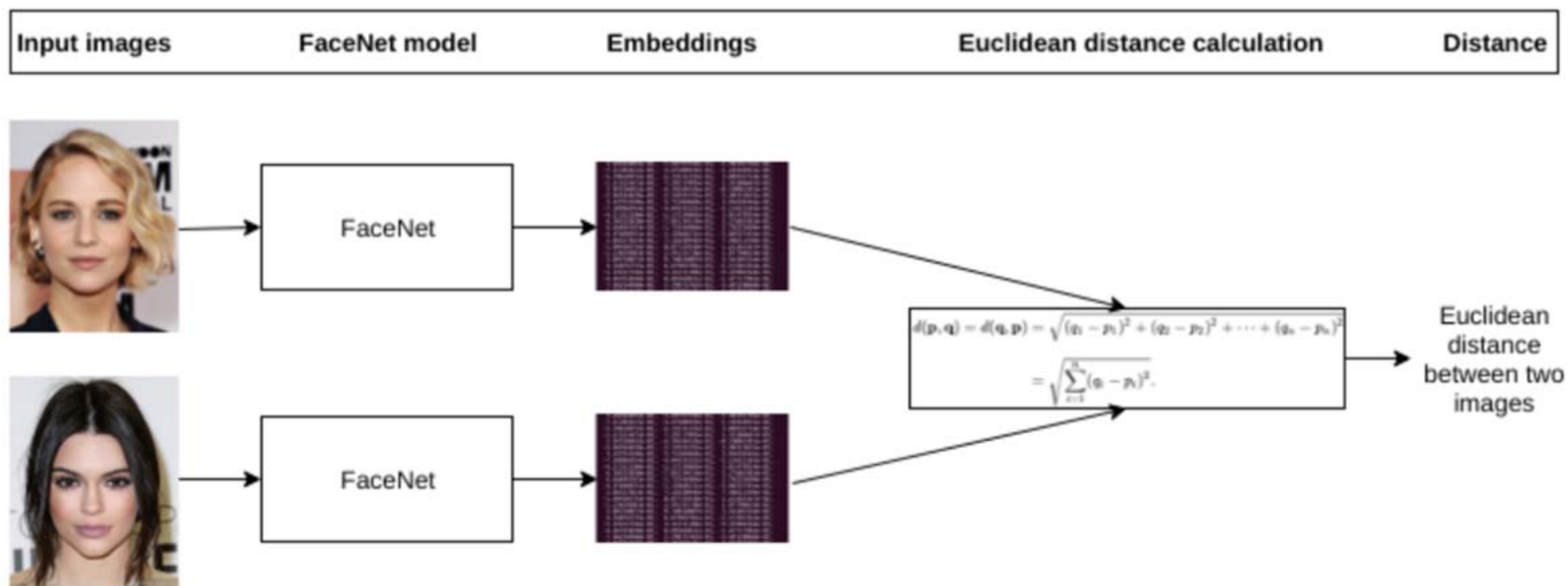
- With a growing amount of research in instances where the model uses as little data as possible with fewer annotations, zero-shot learning has found applications in critical areas like healthcare for [medical imaging](#) and COVID-19 diagnosis using chest x-rays as well as for unknown object detection used in autonomous vehicles.
- [Hugging Face](#)* transformers use zero-shot classification for more than 60 per cent of its transformers.

*Hugging Face Acquires Gradio, A Customizable UI Components Library For Python

One-shot learning (1)

- One-shot learning **performs classification tasks using past data.**
- [Facial recognition technology](#), including facial verification and identification, usually uses one-shot learning.
- Facial recognition systems learn face embedding, which is a rich low-dimensional feature representation.
- One-shot learning has been using the Siamese network approach.
- Eventually, Siamese networks were compared to comparative loss functions, after which the triplet loss function was proven to be better and the FaceNet system began using them.
- **Contrastive loss and triplet loss functions are now used for high-quality face embeddings**, which have become the foundation for modern facial recognition.

One-shot learning (2)



Few-shot learning (1)

- [Few-shot learning](#), also known as low-shot learning, **uses a small set of examples from new data to learn a new task.**
- The process of few-shot learning deals with a type of machine learning problem specified by say E, and it consists of a limited number of examples with supervised information for a target T.
- Few shot learning is commonly used by OpenAI as GPT3 is a few-shot learner.
- A [study](#) in 2019 titled 'Meta-Transfer Learning for Few Shot Learning' addressed the challenges that few-shot settings faced. Since then, few-shot learning is also known as a **meta learning problem.**

Few-shot learning (2)

- There are two ways to approach few-shot learning:
 - **Data-level approach:**
 - According to this process, **if there is insufficient data to create a reliable model, one can add more data to avoid overfitting and underfitting.**
 - The data-level approach uses a large base dataset for additional features.
 - **Parameter-level approach:**
 - Parameter-level method needs **to limit parameter space** and **use regularisation** and **proper loss functions** to resolve the **overfitting** problem in few-shot learning.
 - This will generalise the limited training samples.
 - This approach can also **improve model performance** by directing it to the extensive parameter space.
 - A parameter-level approach is **useful because a smaller amount of training data will not give reliable results.**

References

- Image similarity estimation using a Siamese Network with a contrastive loss, https://keras.io/examples/vision/siamese_contrastive/
- Siamese networks with Keras, TensorFlow, and Deep Learning, <https://pyimagesearch.com/2020/11/30/siamese-networks-with-keras-tensorflow-and-deep-learning/>
- One Shot Learning with Siamese Networks in PyTorch, <https://hackernoon.com/one-shot-learning-with-siamese-networks-in-pytorch-8ddaab10340e>
- A friendly introduction to Siamese Networks, <https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942>
- One-Shot Learning for Face Recognition, <https://machinelearningmastery.com/one-shot-learning-with-siamese-networks-contrastive-and-triplet-loss-for-face-recognition/>

References

- Poulomi Chatterjee, How do zero-shot, one-shot and few-shot learning differ?, March 23, 2022. <https://analyticsindiamag.com/how-do-zero-shot-one-shot-and-few-shot-learning-differ/>
- Qianru Sun, Meta-Transfer Learning for Few Shot Learning, 2019, <https://arxiv.org/pdf/1812.02391v3.pdf>
- Nagesh Singh Chauhan, Zero-Shot Learning: Can you classify an object without seeing it before? April 12, 2021, <https://www.kdnuggets.com/2021/04/zero-shot-learning.html>