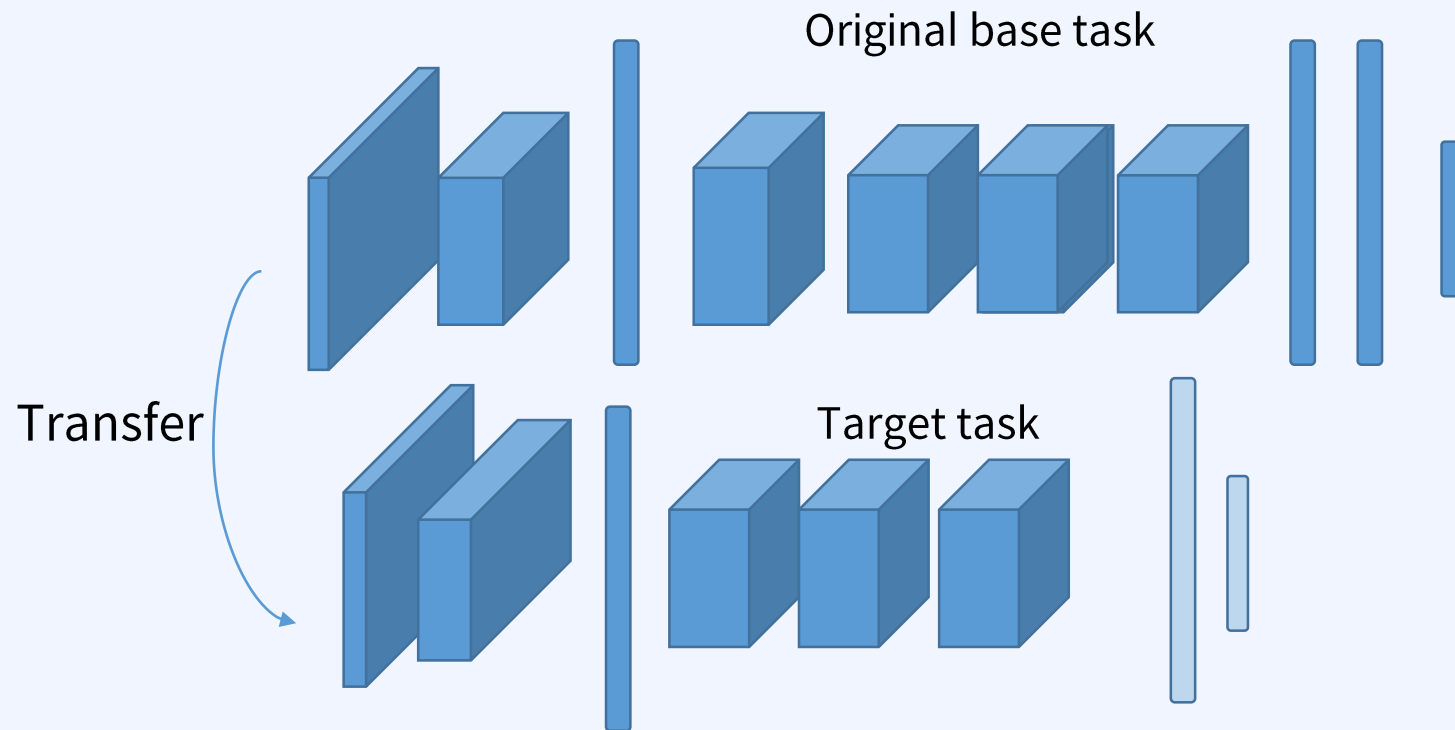


Ch1. Transfer learning

How transferable are features in deep neural networks?
A Baseline for Few-Shot Image Classification

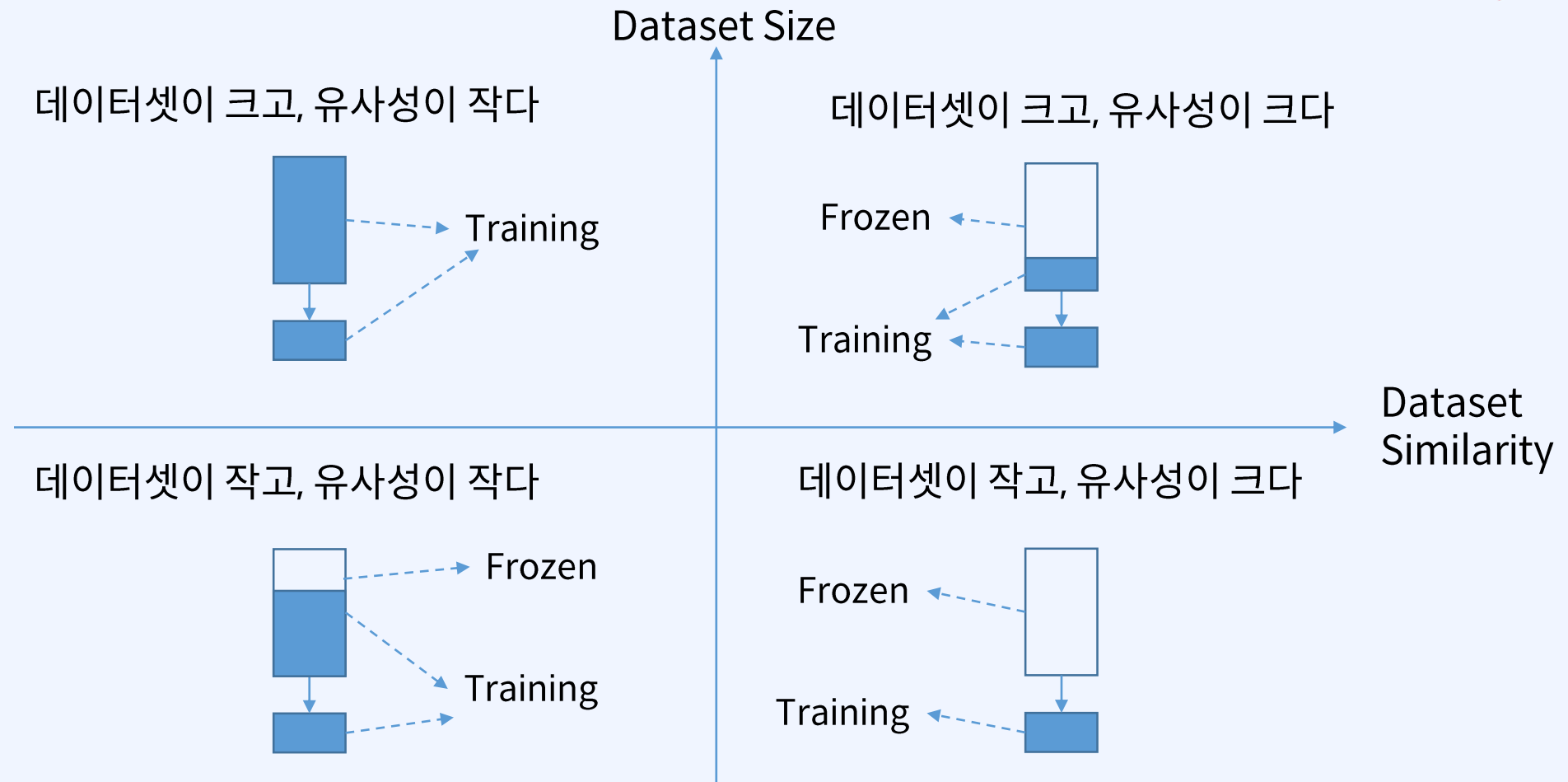
Transfer learning

- Pre-trained model(사전 학습된 모델)을 가져다 사용 하는 것
- 다른 데이터 셋으로 잘 학습되어 있는 모델을 사용
→ 학습을 빠르게 진행



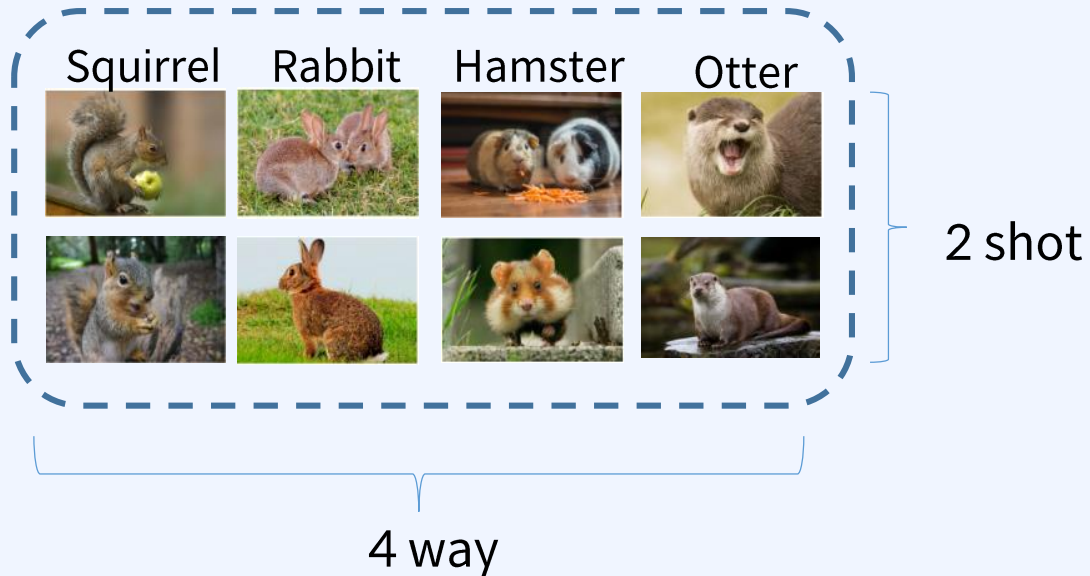
Fine tuning

• Task에 따른 Fine tuning



Few/One shot learning

- Few shot learning의 문제



- One shot learning의 문제



Basic Idea

- Learn similarity function : $\text{sim}(x, x')$
- Ideally, $\text{sim}(x_1, x_2) = 1$, $\text{sim}(x_1, x_3) = 0$, and $\text{sim}(x_2, x_3) = 0$

Rabbit



x_1

Rabbit



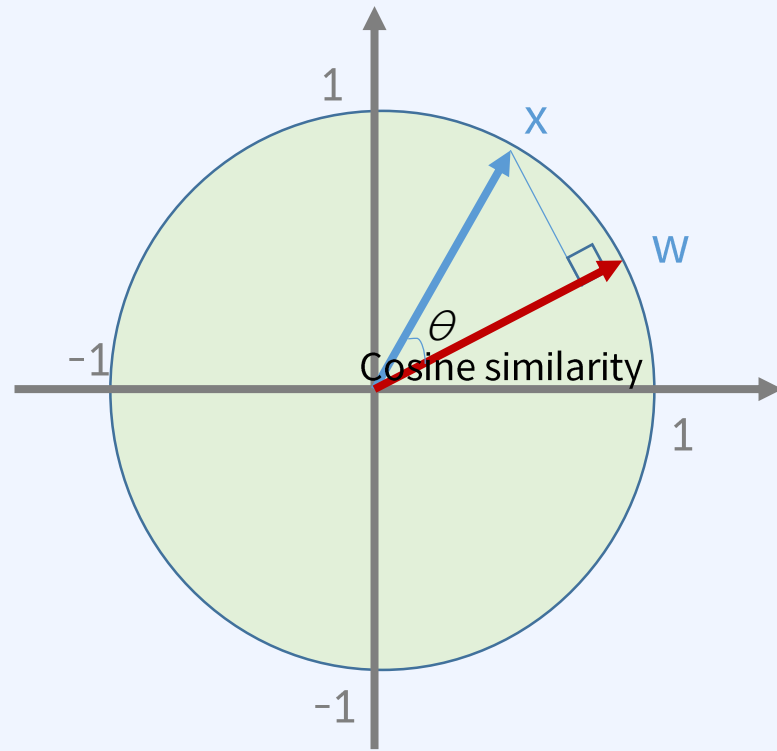
x_2

Hamster



x_3

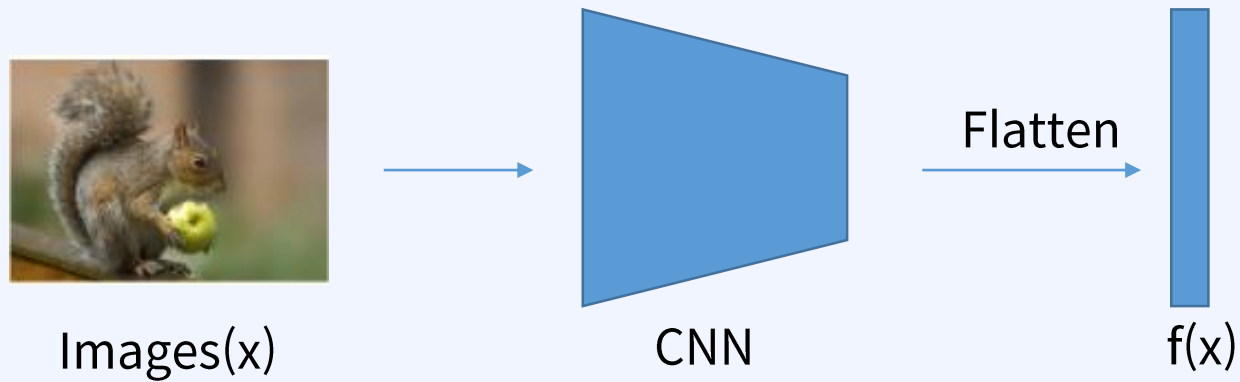
Cosine Similarity



- x, w 를 unit vector로 가정 :
 - $\|x\|_2 = 1$ and $\|w\|_2 = 1$
- Cosine similarity :
 - $\cos \theta = x^T w$

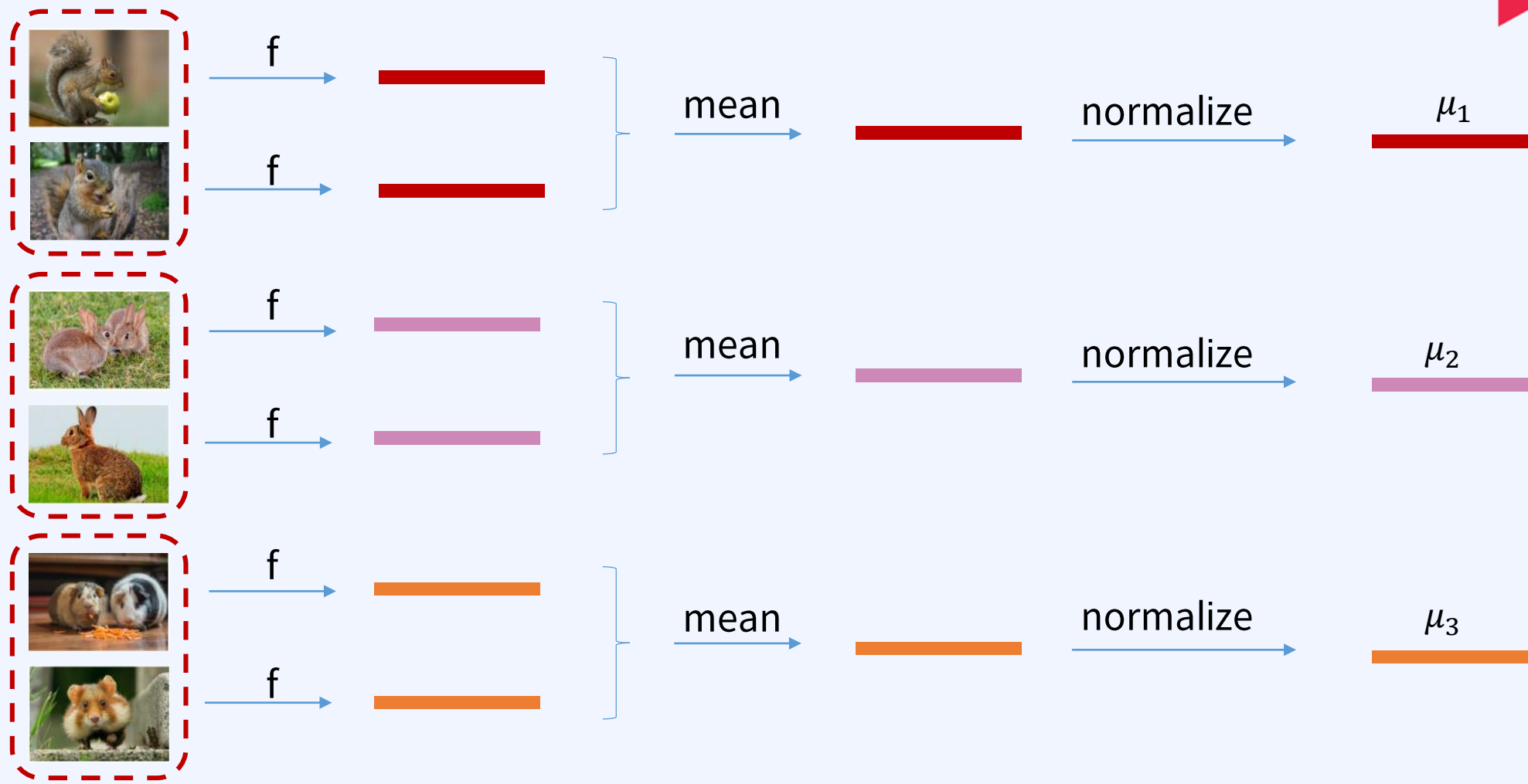
Few-Shot Prediction Using Pretrained model

- Pretrained 모델(feature extractor for CNN)



Making Few-Shot Prediction

3-way 2-shot Support Set



Making Few-Shot Prediction

Query



f



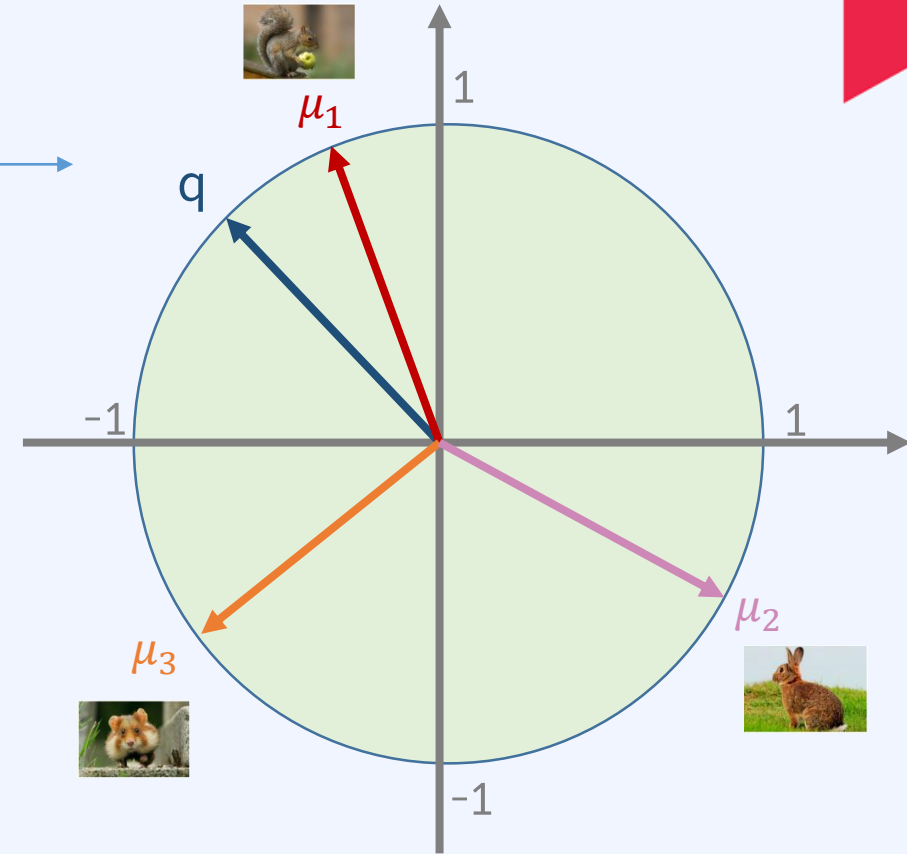
normalize

- Normalized mean feature vectors

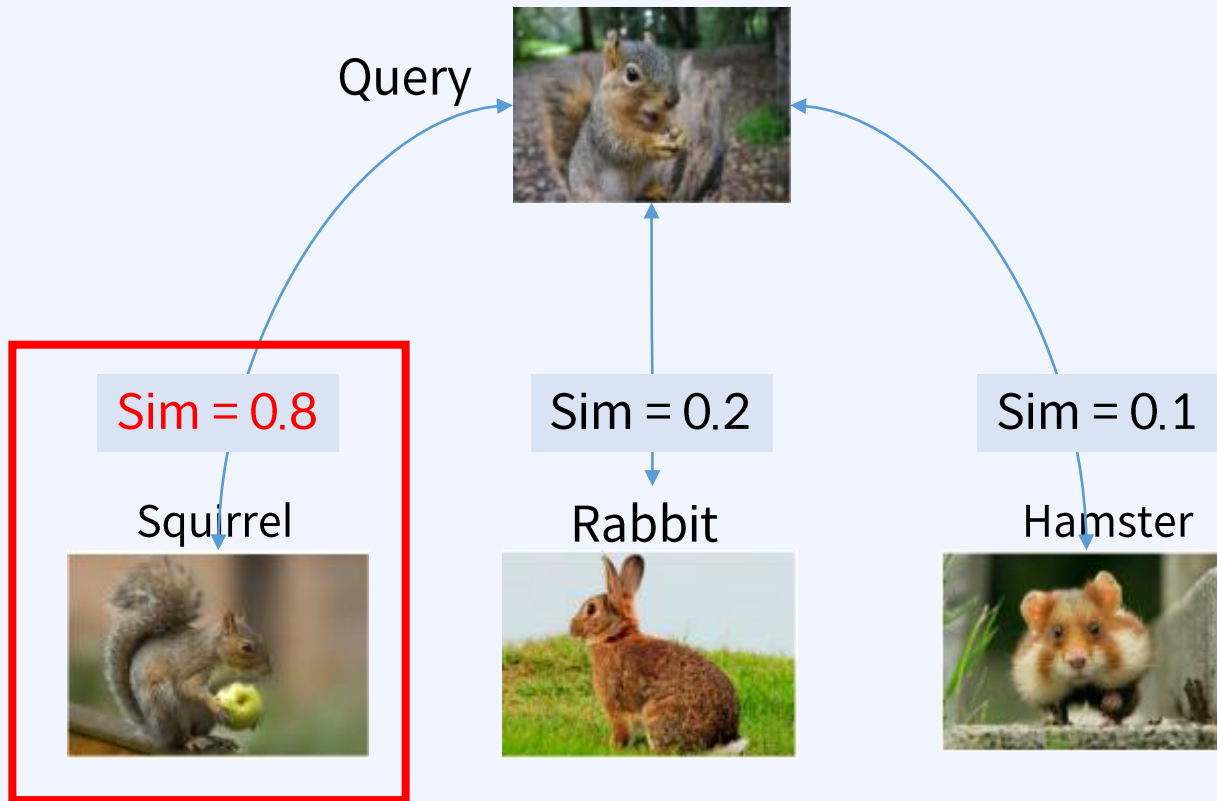
$$M = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}$$

- Make prediction: $P = \text{Softmax}(Mq)$

$$= \text{Softmax} \left(\begin{bmatrix} \mu_1^T q \\ \mu_2^T q \\ \mu_3^T q \end{bmatrix} \right)$$



Few-Shot Prediction

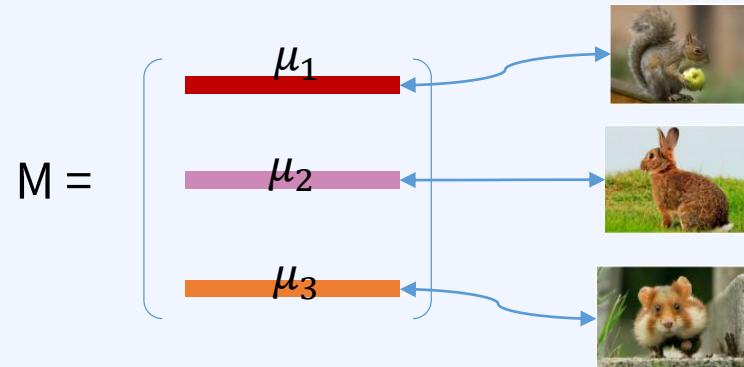


Fine-Tuning

- Prediction made by Softmax classifier

$$P = \text{Softmax}(W \cdot f(x) + b) = \text{Softmax}\left(\begin{bmatrix} \text{sim}(w_1, q) + b_1 \\ \text{sim}(w_2, q) + b_2 \\ \text{sim}(w_3, q) + b_3 \end{bmatrix}\right)$$

- Initialization: $W = M$ and $b = 0$



Summary

- Transfer learning
 - Fine-tuning
 - Few/one shot learning