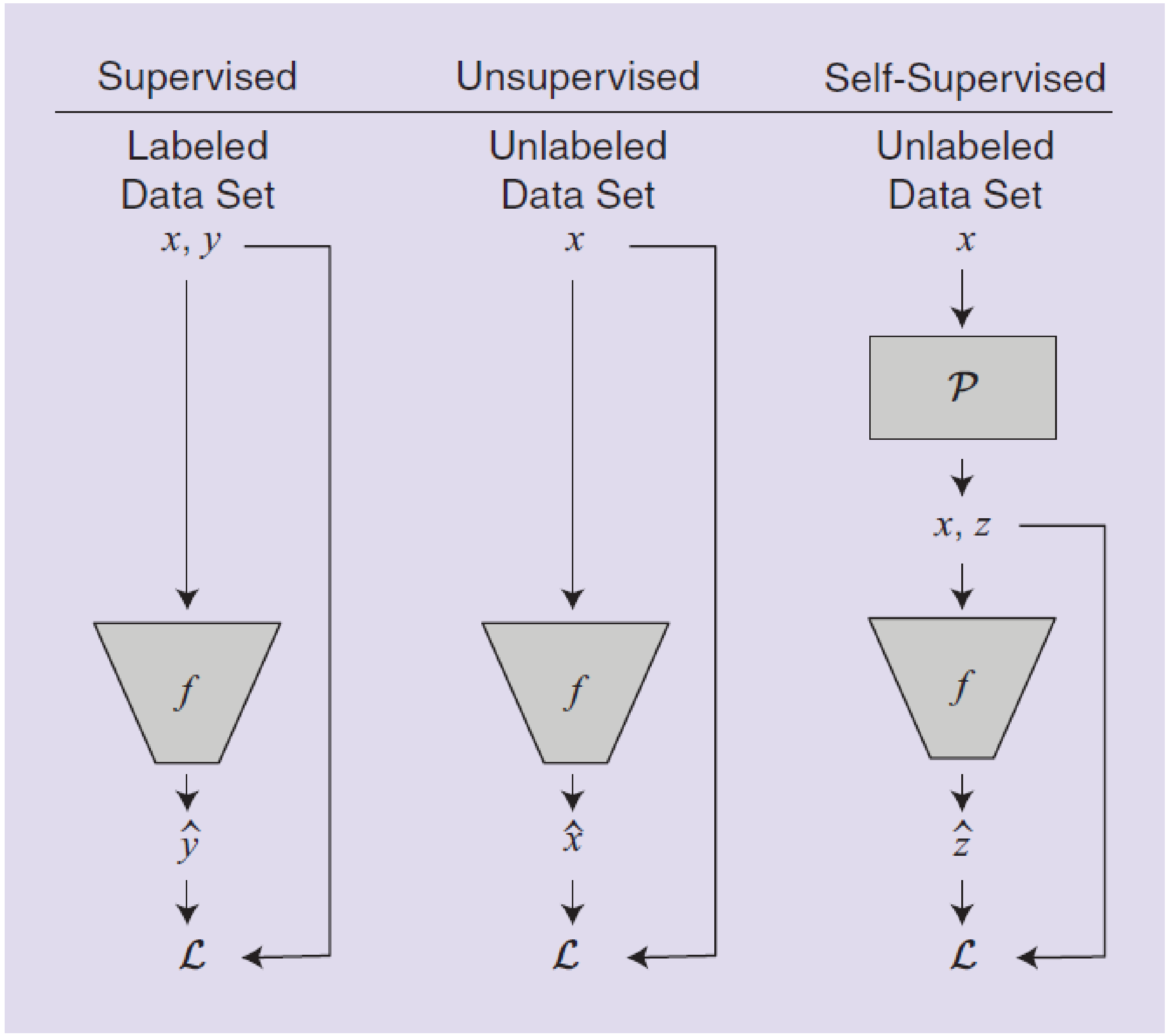


FIGURE 1. Contrasting supervised, unsupervised and self-supervised learning paradigms for training a model f using raw data x, labels y, and loss function $\mathcal{L}$. Self-supervision methods introduce pretext tasks $\mathcal{P}$ that generate pseudolabels z for discriminative training of f.

Module – 2

Basics of Foundation Models, SSL formalisms, definitions and examples of

- **Pretext tasks**
- **Losses**
- **Downstream adaptations**

in a domain-agnostic setting

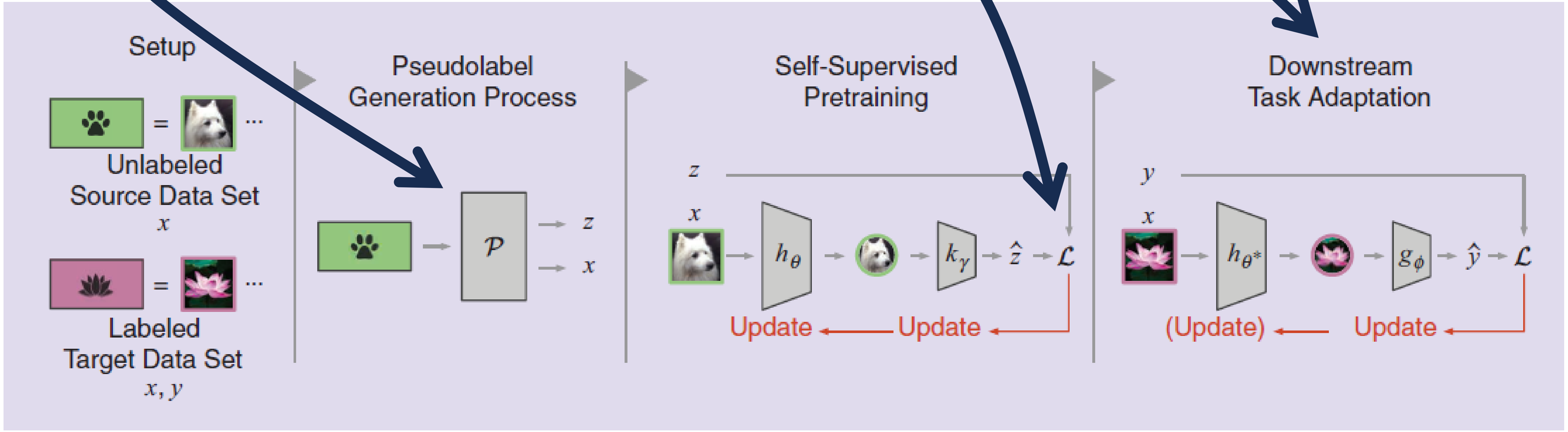


FIGURE 2. The self-supervised workflow starts with an unlabeled source data set and a labeled target data set. As defined by the pretext task, pseudolabels are programmatically generated from the unlabeled set. The resulting inputs, x and pseudolabels z, are used to pretrain the model $k_\gamma(h_\theta(\cdot))$—composed of feature extractor h_θ and output k_γ modules—to solve the pretext task. After pretraining is complete, the learned weights θ^* of the feature extractor h_{θ^*} are transferred and used together with a new output module g_ϕ to solve the downstream target task.

4.10 Theory

Authors: Aditi Raghunathan, Sang Michael Xie, Ananya Kumar, Niladri Chatterji, Rohan Taori, Tatsunori Hashimoto, Tengyu Ma

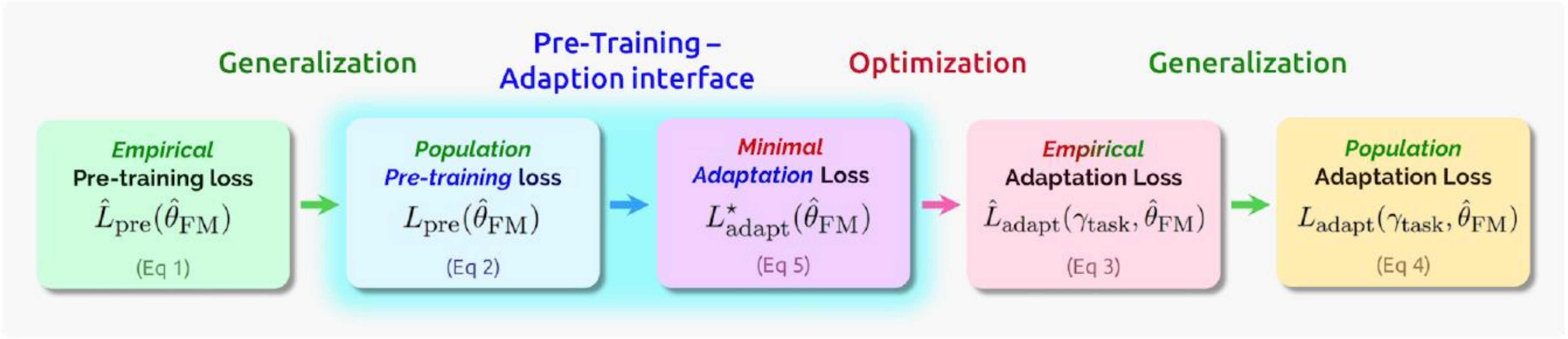
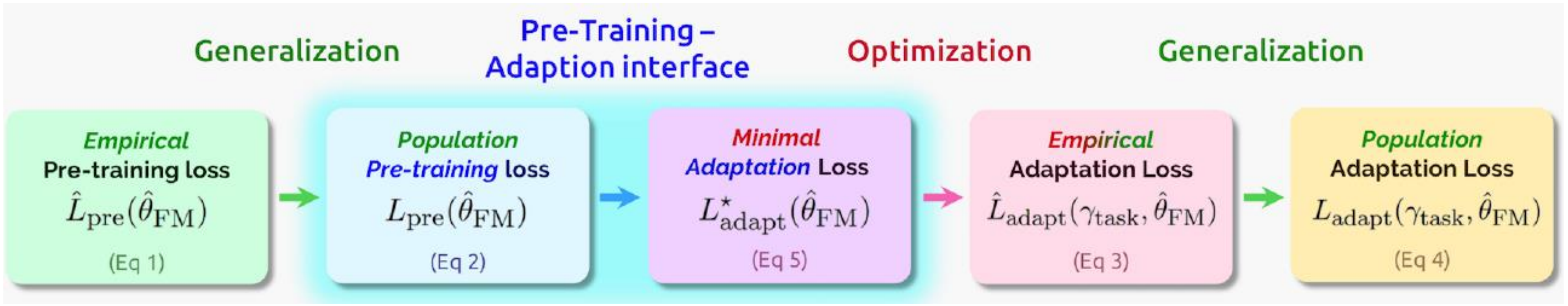

Generalization
Pre-Training – Adaption interface
Optimization
Generalization
Empirical Pre-training loss
$\hat{L}_{\text{pre}}(\hat{\theta}_{\text{FM}})$
(Eq 1)
Population Pre-training loss
$L_{\text{pre}}(\hat{\theta}_{\text{FM}})$
(Eq 2)
Minimal Adaptation Loss
$L^{\star}_{\text{adapt}}(\hat{\theta}_{\text{FM}})$
(Eq 5)
Empirical Adaptation Loss
$\hat{L}_{\text{adapt}}(\gamma_{\text{task}}, \hat{\theta}_{\text{FM}})$
(Eq 3)
Population Adaptation Loss
$L_{\text{adapt}}(\gamma_{\text{task}}, \hat{\theta}_{\text{FM}})$
(Eq 4)

Fig. 22. The analysis of foundation models from pretraining on diverse data to downstream performance on adapted tasks involves capturing the relation between different loss terms as shown above. The main challenge is to analyze the highlighted pretraining-adaptation interface which requires reasoning carefully about the population losses in addition to the model architecture, losses and data distributions of the pretraining and adaptation stages (§4.10.2: THEORY-INTERFACE). Analysis of generalization and optimization largely reduces to their analysis in standard supervised learning.


Generalization
Pre-Training – Adaption interface
Optimization
Generalization
Empirical Pre-training loss
$\hat{L}_{\text{pre}}(\hat{\theta}_{\text{FM}})$
(Eq 1)
Population Pre-training loss
$L_{\text{pre}}(\hat{\theta}_{\text{FM}})$
(Eq 2)
Minimal Adaptation Loss
$L^{\star}_{\text{adapt}}(\hat{\theta}_{\text{FM}})$
(Eq 5)
Empirical Adaptation Loss
$\hat{L}_{\text{adapt}}(\gamma_{\text{task}}, \hat{\theta}_{\text{FM}})$
(Eq 3)
Population Adaptation Loss
$L_{\text{adapt}}(\gamma_{\text{task}}, \hat{\theta}_{\text{FM}})$
(Eq 4)

As shown in Figure 22, the main missing link beyond standard supervised theory is:

Under what conditions does a small population pretraining loss $L_{pre}(\hat{\theta}_{\text{FM}})$ imply a small minimal adaptation loss $L^{\star}_{adapt}(\hat{\theta}_{\text{FM}})$ and why?

Deep Learning pipeline

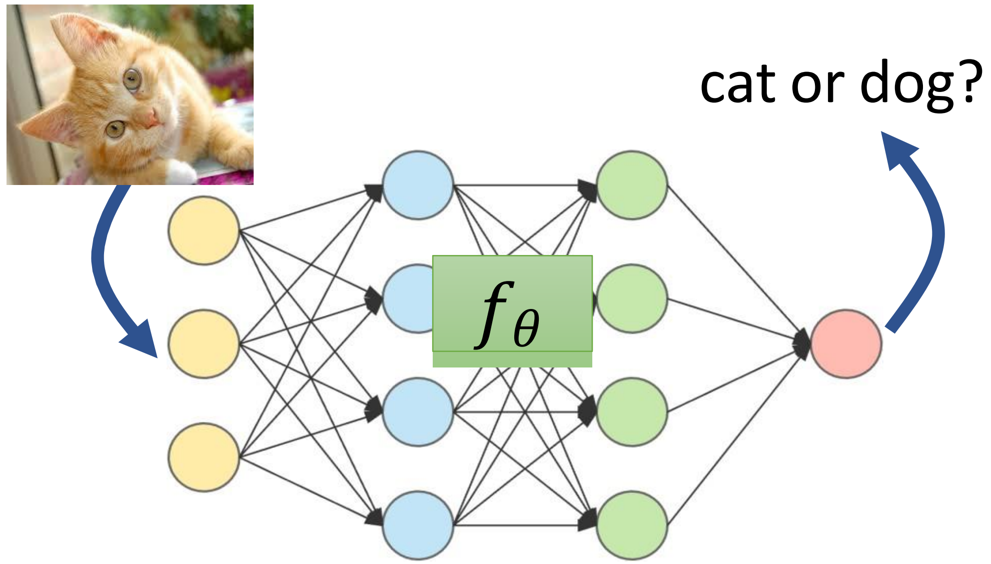
Step 1: What is learnable

Step 2: Define loss function

Step 3: Optimization

Dog-Cat Classification

$f($ $) =$ “cat”



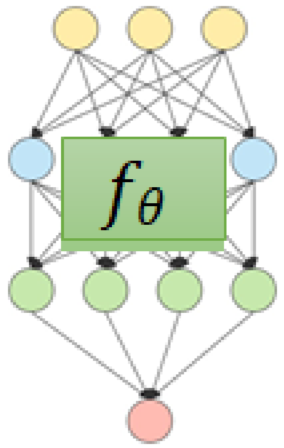
Weights and biases of neurons are learnable.

Using θ to represent the learnable parameters.

Step 1: What is learnable

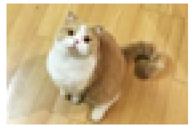
Step 2: Define loss function

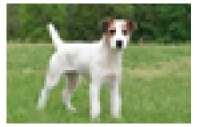
Step 3: Optimization



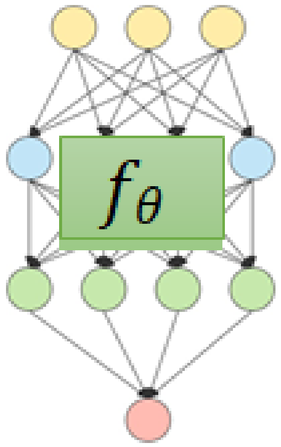
$l(\theta)$

Training Examples





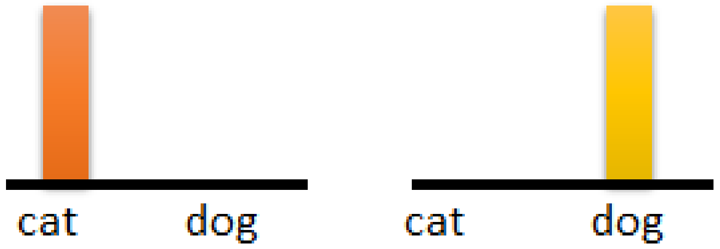
Step 1: What is learnable



Step 2: Define loss function

$\boxed{l(\theta)}$

Step 3: Optimization

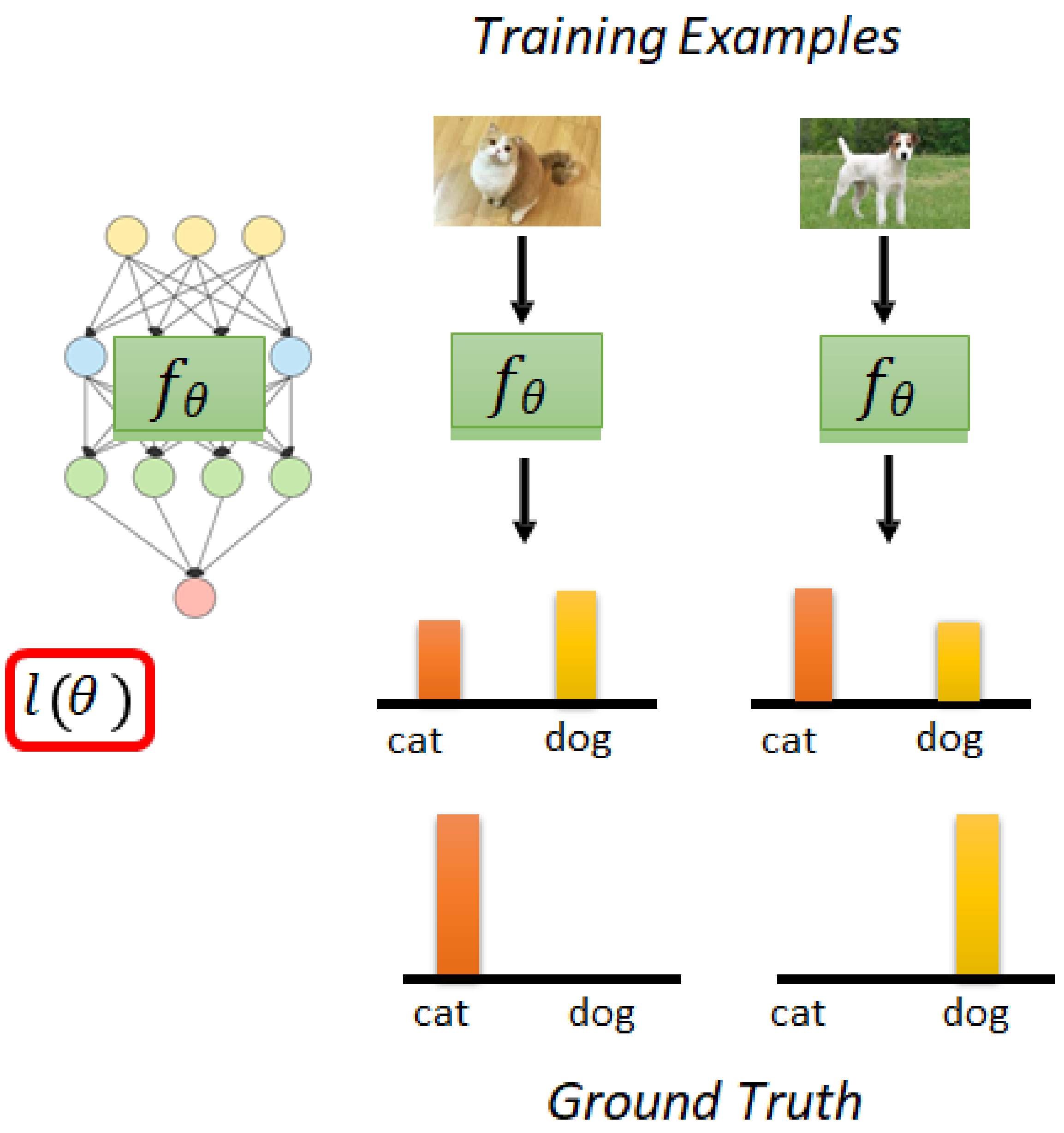


Ground Truth

Step 1: What is learnable

Step 2: Define loss function

Step 3: Optimization

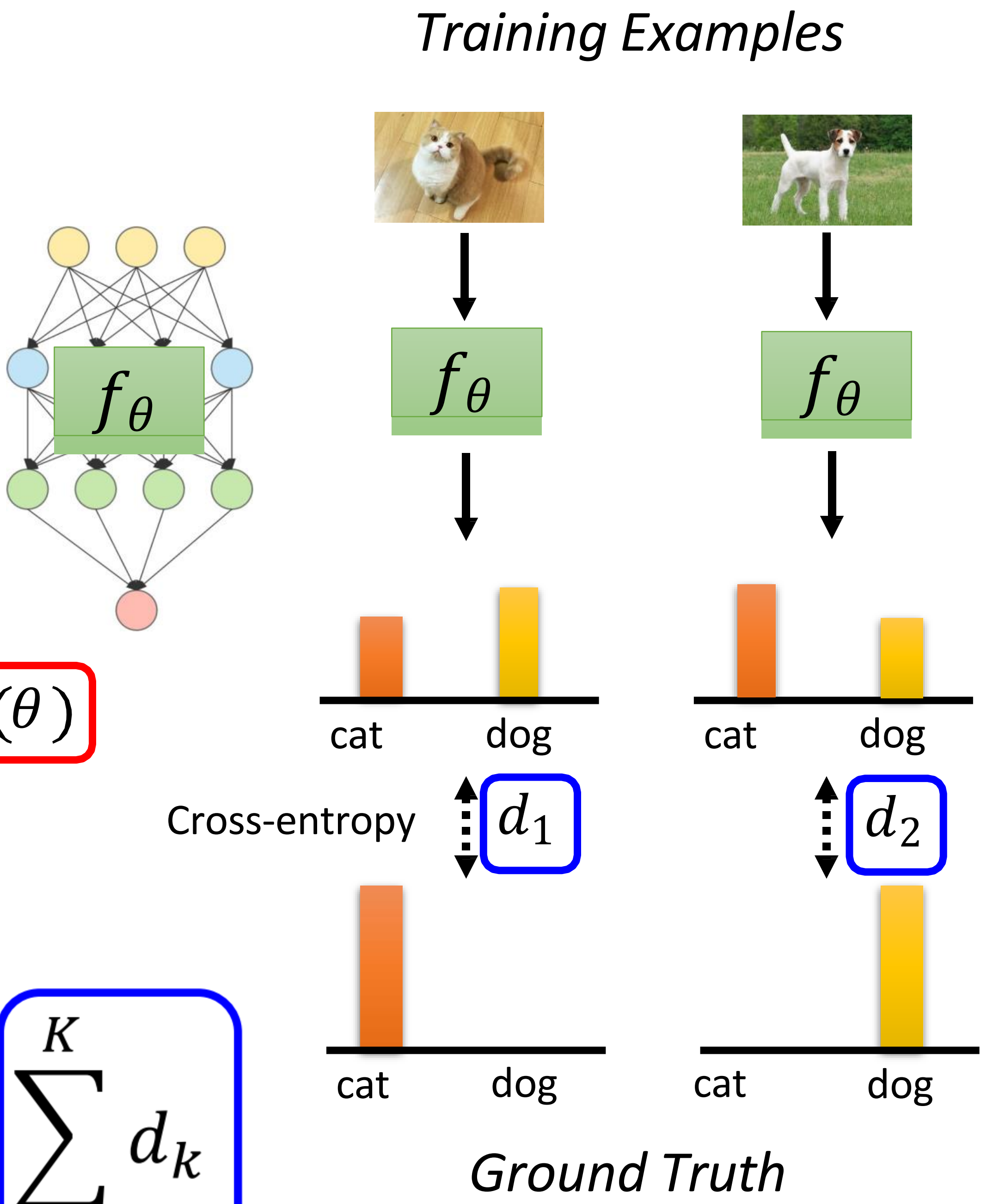


Step 1: What is learnable

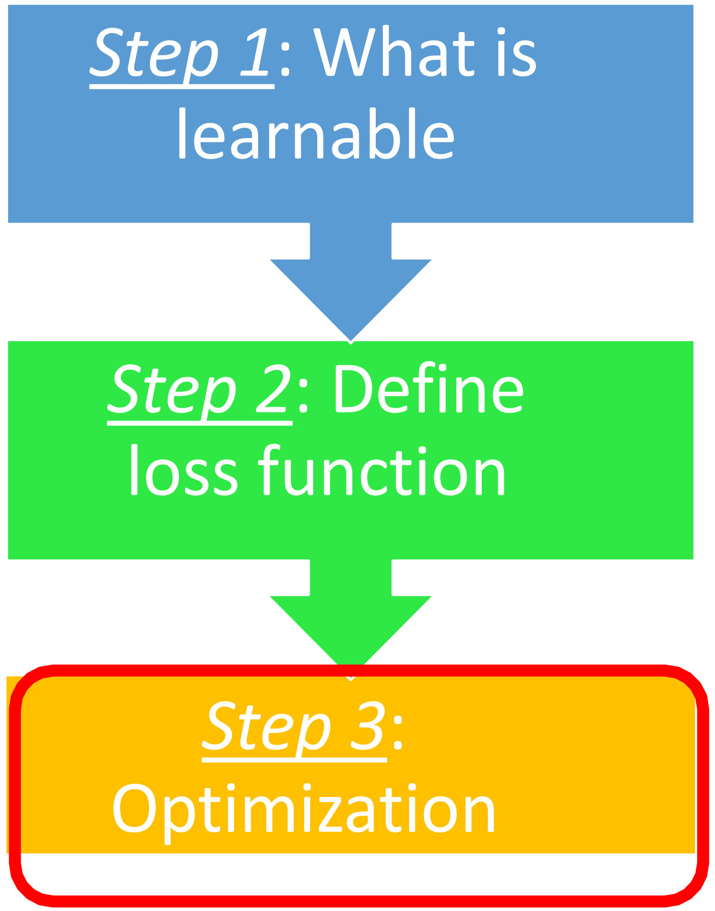
Step 2: Define loss function

$\boxed{l(\theta)}$

Step 3: Optimization



$$\boxed{l(\theta)} = \boxed{\sum_{k=1}^{K} d_k}$$



loss: $l(\theta) = \sum_{k=1}^{K} d_k$ sum over training examples

$\hat{\theta} = arg \min_{\theta} l(\theta)$

done by gradient descent

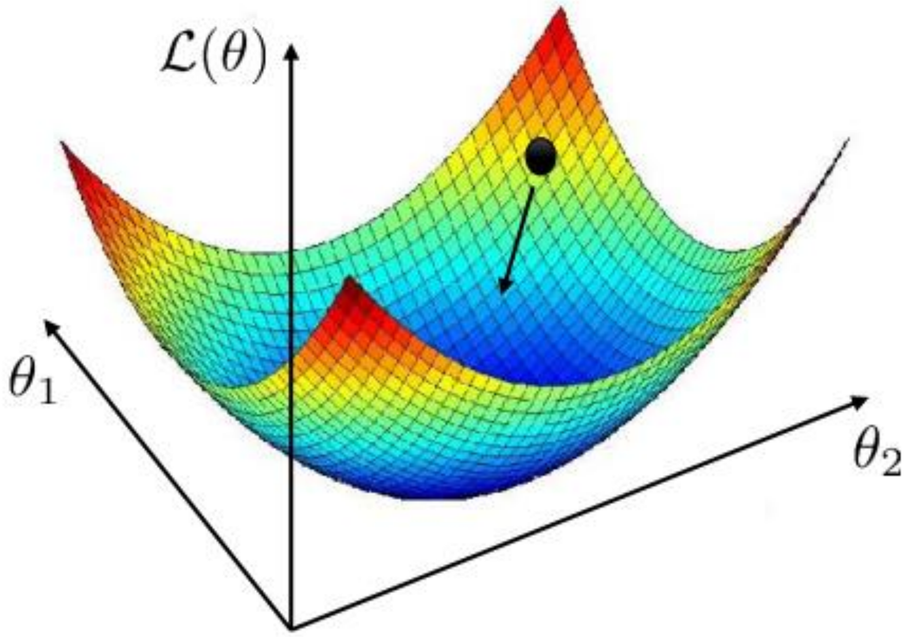
$f_{\hat{\theta}}$ is the function learned by learning algorithm from data

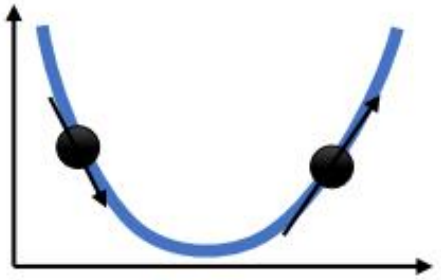
Gradient descent

An algorithm:

1. Find a *direction* v where $\mathcal{L}(\theta)$ decreases
2. $\theta \leftarrow \theta + \alpha v$

Which way does $\mathcal{L}(\theta)$ decrease?





negative slope = go to the right

positive slope = go to the left

in general:

for each dimension, go in the direction opposite the slope **along that dimension**

gradient:

$$\nabla_\theta \mathcal{L}(\theta) = \begin{pmatrix} \frac{d\mathcal{L}(\theta)}{d\theta_1} \\ \frac{d\mathcal{L}(\theta)}{d\theta_2} \\ \vdots \\ \frac{d\mathcal{L}(\theta)}{d\theta_n} \end{pmatrix}$$

$$v_1 = -\frac{d\mathcal{L}(\theta)}{d\theta_1} \quad v_2 = -\frac{d\mathcal{L}(\theta)}{d\theta_2} \quad \text{etc.}$$

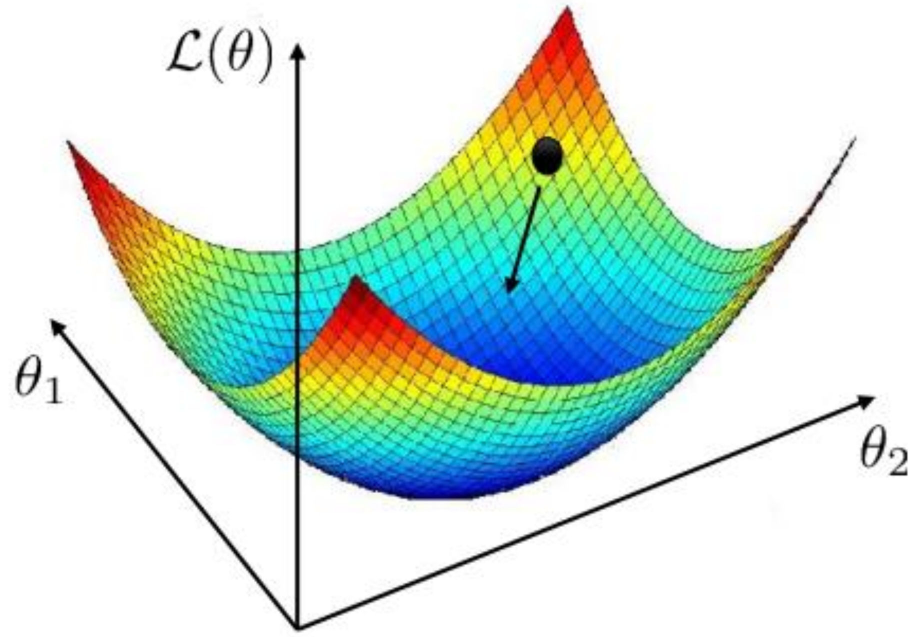
Gradient descent

An algorithm:

1. Find a *direction* v where $\mathcal{L}(\theta)$ decreases
2. $\theta \leftarrow \theta + \alpha v$

Gradient descent:

1. Compute $\nabla_\theta \mathcal{L}(\theta)$
2. $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$



$$\nabla_\theta \mathcal{L}(\theta) = \begin{pmatrix} \frac{d\mathcal{L}(\theta)}{d\theta_1} \\ \frac{d\mathcal{L}(\theta)}{d\theta_2} \\ \vdots \\ \frac{d\mathcal{L}(\theta)}{d\theta_n} \end{pmatrix}$$

Thank you !!