Understanding meta-learning for fast adaptation

Kavya Ranjan Saxena
Dr. Vipul Arora

Department of Electrical Engineering
Indian Institute of Technology, Kanpur

January 4, 2023

MADHAV lab
Machine Analysis of Data for Human Audition and Vision

Flow of the tutorial

1. Introduction
   - Supervised learning
2. Different techniques
   - Fine-tuning
   - Few-shot learning
   - Meta-learning
3. Meta-learning : Theory + Hands-on
   - Theory
   - Example
   - Hands-on

Introduction

Supervised learning

- $\hat{y} = F_\theta(x)$ [1,2]

[1] Bishop, C. M., Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer

[2] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press

Supervised learning

- $\hat{y} = F_\theta(x)$ [1,2]
- Define loss : $\mathcal{L}(y, \hat{y})$

---

[1] Bishop, C. M., Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer

[2] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press

Supervised learning

- $\hat{y} = F_\theta(x)$ [1,2]
- Define loss : $\mathcal{L}(y, \hat{y})$
- Train it using data $\mathcal{D} = \{(x, y)\}$ (input-output pairs)

[1] Bishop, C. M., Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer

[2] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press

Supervised learning

- $\hat{y} = F_\theta(x)$ [1,2]
- Define loss : $\mathcal{L}(y, \hat{y})$
- Train it using data $\mathcal{D} = \{(x, y)\}$ (input-output pairs)

$$\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{(x,y) \in \mathcal{D}} \mathcal{L}(y, \hat{y}) \tag{1}$$

[1] Bishop, C. M., Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer

[2] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press

Supervised learning

- $\hat{y} = F_\theta(x)$ [1,2]
- Define loss : $\mathcal{L}(y, \hat{y})$
- Train it using data $\mathcal{D} = \{(x, y)\}$ (input-output pairs)

$$\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{(x,y) \in \mathcal{D}} \mathcal{L}(y, \hat{y}) \tag{1}$$

- Evaluate on test data (accuracy,mse,etc.)

[1] Bishop, C. M., Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer

[2] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press

Limitations of Supervised learning

- **Problem 1:** Needs a large amount of labelled data (input-output) pairs
    - annotations may be difficult to obtain
        - manual labelling
        - scarce data
        - annotations may be computationally too expensive
    - annotations may be changing over time

Limitations of Supervised learning

- **Problem 1:** Needs a large amount of labelled data (input-output) pairs
  - annotations may be difficult to obtain
    - manual labelling
    - scarce data
    - annotations may be computationally too expensive
  - annotations may be changing over time
- **Problem 2:** Even though large amount of labelled data is available, the model performs well on target data if data distribution of target data is same as source data i.e $P(X_t) = P(X_s)$

Remedies

- Semi-supervised learning
  - self-supervised
  - unsupervised
- Few-shot learning
- Model adaptation
  - transfer learning
  - meta learning

Different techniques

Fine-tuning

Model pre-training:

- Supervised training on related tasks
  - Large networks trained on large labeled datasets
  - e.g., VGG16 [3] for images classification
    - No. of trainable parameters 100M
    - 1.3M images
    - 1000 classes

[3] Simonyan and Zisserman. "Very deep convolutional networks for large-scale image recognition", ICLR 2015

Fine-tuning

Supervised model pre-training:

- $\hat{y} = F_\phi(F_\theta(x))$

Fine-tuning

Supervised model pre-training:

- $\hat{y} = F_\phi(F_\theta(x))$
- Train $F_\phi(F_\theta(x))$ for a supervised task with big data

Fine-tuning

Supervised model pre-training:

- $\hat{y} = F_\phi(F_\theta(x))$
- Train $F_\phi(F_\theta(x))$ for a supervised task with big data
- Discard F_ϕ and use $F_{\phi'}(F_\theta)$ for the new task

Fine-tuning

Supervised model pre-training:

- $\hat{y} = F_\phi(F_\theta(x))$
- Train $F_\phi(F_\theta(x))$ for a supervised task with big data
- Discard F_ϕ and use $F_{\phi'}(F_\theta)$ for the new task
- Freeze F_θ and update $F_{\phi'}$

Fine-tuning

Supervised model pre-training:

- $\hat{y} = F_\phi(F_\theta(x))$
- Train $F_\phi(F_\theta(x))$ for a supervised task with big data
- Discard F_ϕ and use $F_{\phi'}(F_\theta)$ for the new task
- Freeze F_θ and update $F_{\phi'}$
- Update the complete model end-to-end

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data
- Classifies data from target domain when few labeled training samples in target domain

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data
- Classifies data from target domain when few labeled training samples in target domain
- N-way K-shot classification problem

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data
- Classifies data from target domain when few labeled training samples in target domain
- N-way K-shot classification problem
- How to solve this problem?

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data
- Classifies data from target domain when few labeled training samples in target domain
- N-way K-shot classification problem
- How to solve this problem?
- Meta-Learning!

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data
- Classifies data from target domain when few labeled training samples in target domain
- N-way K-shot classification problem
- How to solve this problem?
- Meta-Learning!
- Meta Learning-

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data
- Classifies data from target domain when few labeled training samples in target domain
- N-way K-shot classification problem
- How to solve this problem?
- Meta-Learning!
- Meta Learning-
  - Metric-based : Matching networks, Prototypical networks, Relation Networks

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Few-shot learning

- To overcome the problem of scarcity of large amount of labeled data
- Classifies data from target domain when few labeled training samples in target domain
- N-way K-shot classification problem
- How to solve this problem?
- Meta-Learning!
- Meta Learning-
  - Metric-based : Matching networks, Prototypical networks, Relation Networks
  - Gradient-based : Model agnostic meta learning[4]

[4] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In International conference on machine learning, pp. 1126-1135. PMLR, 2017.

Matching Networks

First method to solve few-shot learning problem

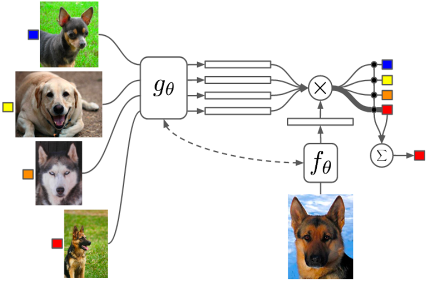
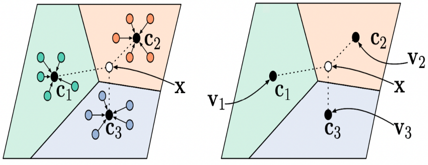


Figure: Matching Networks[5]

[5] Vinyals, Oriol, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. "Matching networks for one shot learning." Advances in neural information processing systems 29 (2016).

Prototypical Networks

Similar to matching networks, but there are small differences!



(a) Few-shot (b) Zero-shot

Figure 1: Prototypical Networks in the few-shot and zero-shot scenarios. **Left**: Few-shot prototypes $\mathbf{c}_k$ are computed as the mean of embedded support examples for each class. **Right**: Zero-shot prototypes $\mathbf{c}_k$ are produced by embedding class meta-data $\mathbf{v}_k$. In either case, embedded query points are classified via a softmax over distances to class prototypes: $p_\phi(y = k|\mathbf{x}) \propto \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_k))$.

Figure: Prototypical Networks[6]

[6] Snell, Jake, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning." Advances in neural information processing systems 30 (2017).

Relation Networks

Almost similar to prototypical networks, but there are small differences!

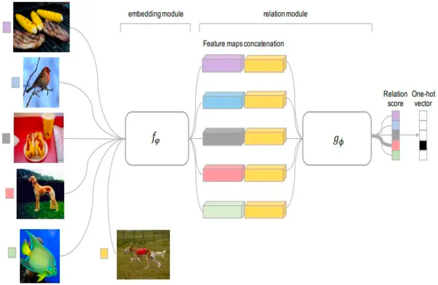


Figure: Relation Networks[7]

[7] Sung, Flood, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M. Hospedales. "Learning to compare: Relation network for few-shot learning." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1199-1208. 2018.

Meta-learning

What is meta-learning?

- **learning-to-learn** algorithm
  - Learn optimal model initialization
  - Learn optimal learning method
  - Learn optimal hyperparameters

Meta-learning

What is meta-learning?

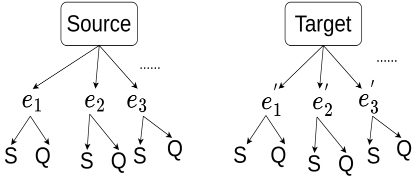
- **learning-to-learn** algorithm
  - **Learn optimal model initialization**
  - Learn optimal learning method
  - Learn optimal hyperparameters

Meta-learning : Theory + Hands-on

Framework

Algorithm to learn optimal model initialization : **Model agnostic meta learning**

- $y = F_{\theta_e}(x)$

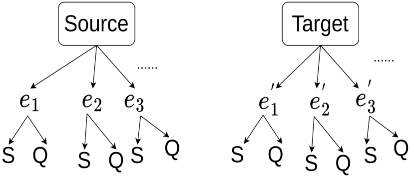


S : Support Set (X_S, Y_S)
Q : Query Set (X_Q, Y_Q)

Framework

Algorithm to learn optimal model initialization : **Model agnostic meta learning**

- $y = F_{\theta_e}(x)$
- θ_e is initialized with θ_0



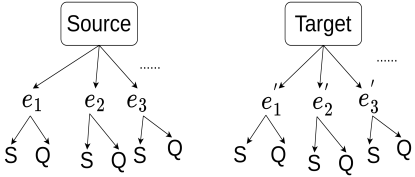
S : Support Set (X_S, Y_S)
Q : Query Set (X_Q, Y_Q)

Framework

Algorithm to learn optimal model initialization : **Model agnostic meta learning**

- $y = F_{\theta_e}(x)$
- θ_e is initialized with θ_0
- Train θ_e on S

$$\theta_e \leftarrow \theta_e - \alpha \frac{\partial \mathcal{L}(S)}{\partial \theta_e}$$



S : Support Set (X_S, Y_S)
Q : Query Set (X_Q, Y_Q)

Framework

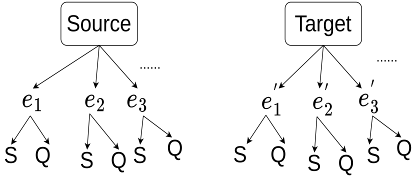
Algorithm to learn optimal model initialization : **Model agnostic meta learning**

- $y = F_{\theta_e}(x)$
- θ_e is initialized with θ_0
- Train θ_e on S

$$\theta_e \leftarrow \theta_e - \alpha \frac{\partial \mathcal{L}(S)}{\partial \theta_e}$$

- Train θ_0 on Q

$$\theta_0 \leftarrow \theta_0 - \alpha \frac{\partial \mathcal{L}(Q)}{\partial \theta_0}$$



S : Support Set (X_S, Y_S)
Q : Query Set (X_Q, Y_Q)

Framework

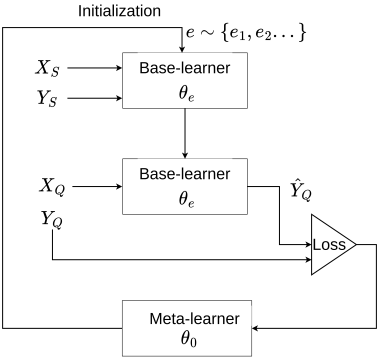


Figure: Meta-training framework

Framework

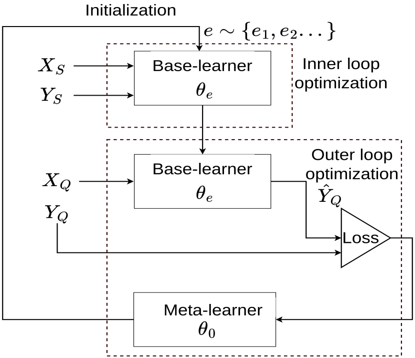


Figure: Meta-training framework

Framework

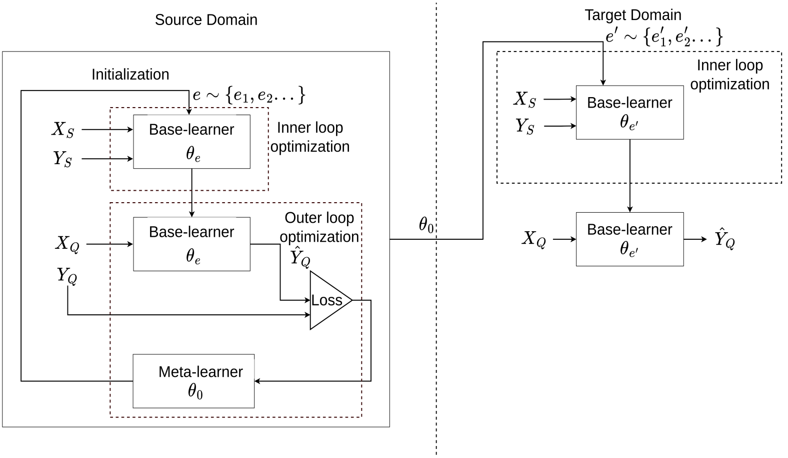


Figure: Overall Meta-learning framework

Meta-learning theory

Algorithm 1 Meta Training Algorithm

Require: Different tasks: $e = \{e_1, e_2, e_3, ...\}$

Require: α,β: learning rates

1: Initialize θ_e with θ_0

2: **for** all episodes e **do**

3: Divide e into support set S and query set Q

4: Update θ_e on support set S with gradient descent: $\theta_e \leftarrow \theta_e - \alpha \frac{\partial \mathcal{L}(S)}{\partial \theta_e}$

5: Update θ_0 on query set Q with gradient descent: $\theta_0 \leftarrow \theta_0 - \alpha \frac{\partial \mathcal{L}(Q)}{\partial \theta_0}$

6: **end for**

Meta-learning theory

Algorithm 2 Meta Testing Algorithm

Require: Different tasks: $e' = \{e'_1, e'_2, e'_3, ...\}$
Require: α: learning rate

1: Initialize $\theta_{e'}$ with θ_0 obtained from **Algorithm 1**
2: **for** all episodes e' **do**
3: Divide e' into support set S and query set Q
4: Update $\theta_{e'}$ on support set S with gradient descent: $\theta_{e'} \leftarrow \theta_{e'} - \alpha \frac{\partial \mathcal{L}(S)}{\partial \theta_{e'}}$
5: Test the updated $\theta_{e'}$ on query set Q
6: **end for**

Melody Extraction

Submitted to ICASSP 2023

DEEP DOMAIN ADAPTATION FOR POLYPHONIC MELODY EXTRACTION

Kavya Ranjan Saxena *Vipul Arora*

Department of Electrical Engineering, Indian Institute of Technology Kanpur, India

ABSTRACT

Extraction of the predominant pitch from polyphonic audio is one of the fundamental tasks in the field of music information retrieval and computational musicology. To accomplish this task using machine learning, a large amount of labeled audio data is required to train the model that predicts the pitch contour. But a classical model pre-trained on data

learning-based domain adaptation approaches. Meta-learning has been recently used to improve the performance of few-shot learning problems [7][8]. The common approaches to meta-learning are metric-based [7], model-based [8], and optimization-based [9] learning that improve the learning speed [10]. To the best of our knowledge, no such work on meta-learning-based domain adaptation for polyphonic

Sensor Calibration

IEEE Sensors Letters 2021

Sensor Signal Processing

Few-shot calibration of low-cost air pollution ($PM_{2.5}$) sensors using meta-learning

Kalpit Yadav[1*], Vipul Arora[1], Mohit Kumar[2], Sachchida Nand Tripathi[2,3], Vidyanand Motiram Motghare[4], and Karansingh A. Rajput[4]

[1] Department of Electrical Engineering, Indian Institute of Technology Kanpur, India
[2] Centre for Environmental Science and Engineering, Indian Institute of Technology Kanpur, India
[3] Department of Civil Engineering, Indian Institute of Technology Kanpur, India
[4] Maharashtra Pollution Control Board, India

Abstract—Low-cost particulate matter sensors are transforming air quality monitoring because they have greater mobility as compared to reference monitors. Calibration of these low-cost sensors requires training data from co-deployed reference monitors. Machine Learning based calibration gives better performance than conventional techniques, but requires a large amount of training data from the sensor, to be calibrated, co-deployed with a reference monitor. In this work, we propose novel transfer learning methods for quick calibration of sensors with minimal co-deployment with reference monitors. Transfer learning utilizes a large amount of data from other sensors along with a limited amount of data from the target sensor. Our experimentation finds the proposed Model-Agnostic-Meta-Learning (MAML) based transfer learning method to be significantly more effective over other competitive baselines, reducing the calibration errors by 32% and 15% relative to the raw observations and the best baseline, respectively.

Index Terms—Air quality, low-cost sensor calibration, few-shot learning, MAML, machine learning

Linear Regression

Consider a linear regression problem and apply meta-learning to it.

Linear Regression

Consider a linear regression problem and apply meta-learning to it.

- Example: $\hat{y} = f_\theta(x) = \sigma(a + bx)$

Linear Regression

Consider a linear regression problem and apply meta-learning to it.

- Example: $\hat{y} = f_\theta(x) = \sigma(a + bx)$
- $\theta = [a, b]$ are the model parameters

Linear Regression

Consider a linear regression problem and apply meta-learning to it.

- Example: $\hat{y} = f_\theta(x) = \sigma(a + bx)$
- $\theta = [a, b]$ are the model parameters
- Consider the loss function as $\mathcal{L} = (y - \hat{y})^2$, where y is the target value and $\hat{y}$ is the predicted output.

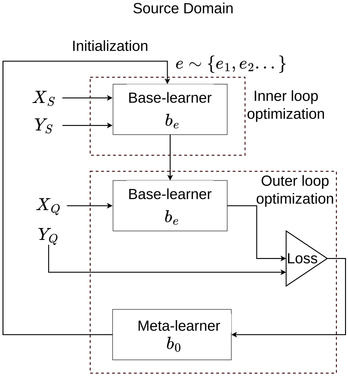
Linear Regression

Consider a linear regression problem and apply meta-learning to it.

- Example: $\hat{y} = f_\theta(x) = \sigma(a + bx)$
- $\theta = [a, b]$ are the model parameters
- Consider the loss function as $\mathcal{L} = (y - \hat{y})^2$, where y is the target value and $\hat{y}$ is the predicted output.
- Let us derive equations for meta-learning the initializer b_0 for weight b.

Linear Regression

Modifying the meta-training framework for our regression problem.



Linear Regression

- Inner-loop optimization

$$b_e \leftarrow b_e - \alpha \nabla_{b_e} \mathcal{L}_S(f(b_e)) \tag{2}$$

Linear Regression

- Inner-loop optimization

$$b_e \leftarrow b_e - \alpha \nabla_{b_e} \mathcal{L}_S(f(b_e)) \tag{2}$$

$$b_e \leftarrow b_e - \alpha \frac{\partial \mathcal{L}_S}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_e} \tag{3}$$

Linear Regression

$$\frac{\partial \mathcal{L}_S}{\partial \hat{y}} = -2(y - \hat{y}) \tag{4}$$

$$\frac{\partial y}{\partial b_e} = \frac{\partial \sigma(a_e + b_e x)}{\partial b_e} \tag{5}$$

$$\frac{\partial y}{\partial b_e} = \hat{y}(1 - \hat{y})x \tag{6}$$

Linear Regression

Putting eq. 4 and eq. 6 in eq. 3, we get:

$$b_e \leftarrow b_e + 2\alpha(y - \hat{y})\hat{y}(1 - \hat{y})x \tag{7}$$

Linear Regression

- Outer-loop optimization

$$b_0 \leftarrow b_0 - \alpha \nabla_{b_0} \mathcal{L}_Q(f(b_0)) \tag{8}$$

$$b_0 \leftarrow b_0 - \alpha \frac{\partial \mathcal{L}_Q}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_e} \frac{\partial b_e}{\partial b_0} \tag{9}$$

Linear Regression

$$\frac{\partial \mathcal{L}_Q}{\partial \hat{y}} = -2(y - \hat{y}) \tag{10}$$

$$\frac{\partial \hat{y}}{\partial b_e} = \hat{y}(1 - \hat{y})x \tag{11}$$

Linear Regression

But, how to compute:

$$\frac{\partial b_e}{\partial b_0} = ? \tag{12}$$

But, how to compute:

$$\frac{\partial b_e}{\partial b_0} = ? \tag{12}$$

No direct relation between b_e and b_0!

Linear Regression

Develop a relation between b_e and b_0 as :

$$b_e = b_0 - \gamma \frac{\partial \mathcal{L}(f(b_0))}{\partial b_0} \tag{13}$$

So, the derivative of the above equation w.r.t b_0 is given by:

$$\frac{\partial b_e}{\partial b_0} = 1 - \gamma \frac{\partial^2 \mathcal{L}(f(b_0))}{\partial b_0^2} \tag{14}$$

Linear Regression

Solving eq. 14, we get:

$$\frac{\partial b_e}{\partial b_0} = 1 - \gamma \frac{\partial}{\partial b_0} \frac{\partial \mathcal{L}}{\partial b_0} \tag{15}$$

$$\frac{\partial b_e}{\partial b_0} = 1 - \gamma \frac{\partial}{\partial b_0} - 2(y - \hat{y})(\hat{y}(1 - \hat{y})x) \tag{16}$$

Linear Regression

$$\frac{\partial b_e}{\partial b_0} = 1 + 2\gamma x[\hat{y}(1-\hat{y})\frac{\partial (y-\hat{y})}{\partial b_0} + \\ (y-\hat{y})(1-\hat{y})\frac{\partial \hat{y}}{\partial b_0} + (y-\hat{y})\hat{y}\frac{\partial (1-\hat{y})}{\partial b_0}] \tag{17}$$

On solving eq.17 we get,

$$\frac{\partial b_e}{\partial b_0} = 1 - 2\gamma x^2(y-\hat{y})[2\hat{y} - 3\hat{y}^2 + 2y\hat{y} - y] \tag{18}$$

So, putting all values of eq. 10,eq. 11 and eq. 18 in eq. 9, we get:

$$b_0 \leftarrow b_0 + 2\beta(y-\hat{y})\hat{y}(1-\hat{y})[1-2\gamma x^2(y-\hat{y})(2\hat{y}-3\hat{y}^2+2y\hat{y}-y)] \tag{19}$$

Advantages of Meta-learning

- **higher model prediction accuracy**
  - optimizing learning algorithms
  - helping learning algorithms better adapt to changes in conditions
- **faster, cheaper training process**
  - learning from fewer examples
  - increase speed of learning processes by reducing necessary experiments
- **more generalized models**

Hands-on

- Source Dataset : MNIST data (first 8 classes)
- Target Dataset : MNIST data (remaining 2 classes)
- Model used : CNNs

Fine-tuning : Hands-on

- $\hat{y} = F_\phi(F_\theta(x))$
- Train $F_\phi(F_\theta(x))$ for a supervised task with big data
- Discard F_ϕ and use $F_{\phi'}(F_\theta)$ for the new task
- Freeze F_θ and update $F_{\phi'}$
- Update the complete model end-to-end

Contents Link

https://github.com/madhavlab/2023_codscomad_
metalearning

Model-agnostic meta-learning: Hands-on

Jupyter Notebook!