12 Multitask Learning

Up to this point, in the context of distributed learning algorithms, we have exclusively focused on consensus optimization problems of the form:

$$w^o \triangleq \arg\min_{w\in\mathbb{R}^M} \sum_{k=1}^{K} J_k(w) \tag{12.1}$$

Since the resulting model w^o is unaffected by normalization, it can be viewed as the best average model for the aggregate objective $\frac{1}{K}\sum_{k=1}^{K} J_k(w)$. We can distinguish the consensus problem from non-cooperative problems:

$$w_k^o = \arg\min_{w_k\in\mathbb{R}^M} J_k(w_k) \iff \arg\min_{\mathcal{W}\in\mathbb{R}^{MK}} \sum_{k=1}^{K} J_k(w_k) \tag{12.2}$$

Since the local objectives are independent between agents, the locally optimal models w_k^o can be pursued in a non-cooperative manner by each individual agent using deterministic or stochastic gradient recursions (see Chapter 2 and 3).

When agents and objectives are homogenous, the solutions to (12.1) and (12.2) coincide and we have $w_k^o = w^o$. In this case we have observed (see, e.g., relations (4.46) and (10.5)) that distributed architectures solving the consensus optimization problem (12.1) exhibit K-fold improvement in performance relative to non-cooperative approaches solving (12.2). This fact offers clear motivation for the deployment of distributed architectures, and for individual agents to participate in a distributed learning protocol. Similar conclusions hold when locally optimal models do not coincide, but are nevertheless similar, i.e., $w_k^o \approx w^o$. In these situations it can be said that "the best average model is also a good local model", and the benefits of collaboration outweigh the cost of converging to the consensus model w^o as opposed to the local model w_k^o.

When there is a significant difference between the optimal average model w^o and local models w_k^o, the question of the "right" network objective and learning algorithm becomes more nuanced, and the right choice depends on the application at hand. We can broadly distinguish two settings:

- **Consensus or single-task learning:** Even when local and global models deviate significantly, it may nevertheless be meaningful to enforce consensus on a single common model, which is optimal on average. For example, in an image recognition task, different agents may be provided with different

kinds of images (e.g., different digits from the MNIST data set), resulting in heterogeneous local data distributions, but we may nevertheless be interested in finding a single common model which can be used to classify images from across the entire network (e.g., all digits from the MNIST data set). In other applications, we may require agents to compromise on a single model w^o, even if it comes at the cost of suboptimal local behavior. In all of these cases, even in presence of heterogeneity, the consensus problem (12.1) is an appropriate network objective, and all of our results from previous chapters apply. In the multitask learning literature, this is often referred to as "single-task" learning as opposed to consensus optimization, though we will use the terms interchangably.

- **Multitask learning:** There are also situations, where agents are self-interested, and ultimately would like to find the optimal local models (12.2) in an efficeint manner. Nevertheless, provided that there is some relationship between the local data distributions, local objectives, or local models w_k^o, it is conceivable that some level of collaboration between agents can be beneficial, and improve performance over purely non-cooperative approaches. As a result, we will need to more carefully design the network objective, and the resulting learning and collaboration protocols. We will refer to these kinds of problems and learning algorithms as *multitask learning*, where a "task" refers to a learning objective.

We may view consensus optimization or single-task learning and noncooperative learning as two ends of a spectrum for learning in multi-agent systems. Noncooperative learning approaches involve no interaction between agents, resulting in local objectives with no coupling, but generally sub-optimal performance, since no synergies between agents are leveraged. Single-task learning on the other hand represents full coupling of local objectives and perfect consensus on a single model, which has the effect of reducing variance but introduces a bias relative to non-cooperative approaches. Multitask learning can then be viewed as living somewhere on this spectrum. We illustrate these different learning paradigms in Fig. 12.1. We will encounter in this chapter different frameworks and associated algorithms, and give an overview of the associated performance trade-offs. We can broadly distinguish two types of multitask learning settings:

- **Parametric multitask learning:** Parametric approaches for multitask learning impose a prior on the relationship between objectives $J_k(w)$ or the optimal local models w_k^o. These priors are generally informed by domain knowledge, physics, or outside information, and drive the cooperation between agents. Parametric approaches for decentralized multitask learning will relax the consensus problem (12.1) to:

$$\mathcal{W}_\eta^o = \arg\min_{\mathcal{W}\in\mathbb{R}^{MK}} \sum_{k=1}^{K} J_k(w_k) + \eta R(\mathcal{W}), \quad \text{s.t. } \mathcal{W}\in\Omega \tag{12.3}$$

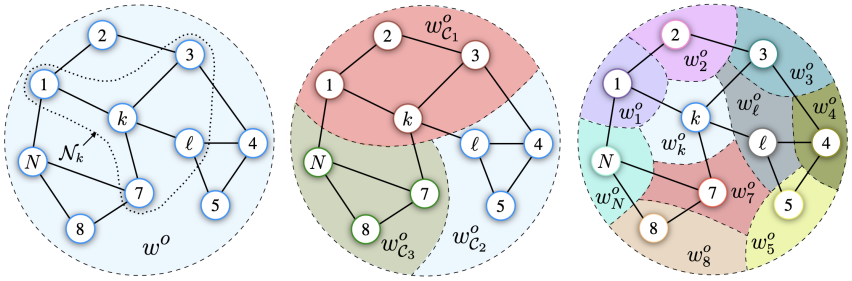


Figure 12.1 Examples of network learning paradigms. (Left) Single-task network. (Middle) Clustered multitask network. (Right) Multitask network.

Here, the coupling regularizer $R(\mathcal{W})$ and constraint Ω encode the prior information available about the local objectives, and will result in different collaboration protocols for agents. We illustrate the generic structure in Fig. 12.2. We will see different examples of task-relatedness models and the resulting learning algorithms.

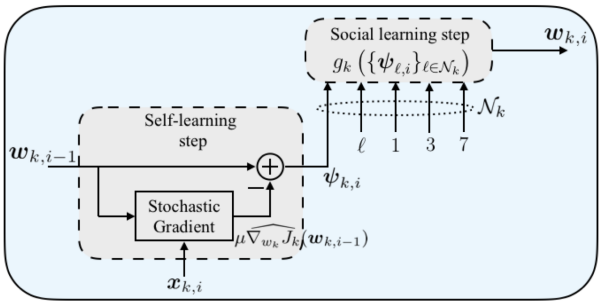


Figure 12.2 A generic structure for decentralized parametric mulittask learning algorithms. It involves two steps, a self-learning step based on locally available data and a social learning step tuned to the underlying task-relatedness model.

- **Non-parametric multitask learning or meta-learning:** In the absence of prior information on task-relatedness, we may instead employ non-parametric approaches, where we assume some relationship between the underlying tasks, which motivate the need for collaboration, but leave the precise relationship for individual learners to find. We will encounter *meta-learning* as a an example of non-parametric approaches for decentralized multitask learning.

12.1 SMOOTHNESS PRIORS

Recall the penalized approximation (8.15) to the consensus problem (12.1), which we encountered in Chapter 8:

$$\arg\min_{\mathcal{W}} \sum_{k=1}^{K} J_k(w_k) + \frac{\eta}{2}\, \mathcal{W}^{\mathsf{T}}\, \mathcal{L}\, \mathcal{W} \tag{12.4}$$

In Chapter 8, where the objective was consensus optimization, we argued that (12.4) only results in a consensual solution to the consensus problem (12.1) when $\eta \to \infty$. Motivated by this observation, we set $\eta = \mu^{-1}$, which ensures that $\eta \to \infty$ for small step-sizes μ. This choice allowed us to conclude in Chapter 9 that resulting consensus+innovation and diffusion algorithms solve the consensus problem (12.1) with high-accuracy for small μ (i.e., large η). If we do not wish to enforce exact consensus, and instead encourage a level of smoothness across the local objectives w_k, we can do so by leaving μ and η uncoupled, and instead solve (12.4) for moderate choices of η. Comparing (12.4) with the generic formulation (12.3), we identify:

$$R(\mathcal{W}) = \frac{\eta}{2}\, \mathcal{W}^{\mathsf{T}}\, \mathcal{L}\, \mathcal{W} \tag{12.5}$$

We note that the choice (12.5) along with leaving η independent from μ is not arbitrary or a mere heuristic inspired by the development of penalty-based algorithms for decentralized consensus optimization. Instead, we can interpret this as a well-defined statistical learning problem, as we illustrate in the following example.

Example 12.1 (**Bayes' optimal estimation under smoothness priors**) Suppose each agent observes data following a linear model:

$$\boldsymbol{\gamma}_k = \boldsymbol{h}_k^{\mathsf{T}} w_k^o + \boldsymbol{v}_k \tag{12.6}$$

with $\boldsymbol{h}_k$ and $\boldsymbol{v}_k$ normally distributed. We collect local samples $\{h_{k,n}, \gamma_{k,n}\}_{n=1}^{N}$. If we impose on the local models w_k^o a Gaussian Markov random field prior of the form:

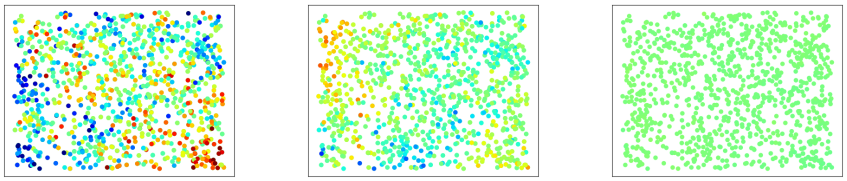
$$f(\mathcal{W}) = (2\pi\eta)^{-M(K-1)/2}\, (|\mathcal{L}|^*)^{1/2}\, e^{-\frac{\eta}{2}\, \mathcal{W}^{\mathsf{T}}\, \mathcal{L}\, \mathcal{W}} \tag{12.7}$$

it follows that the maximum aposteriori estimate (see Chapter 1) of the local models w_k^o is given by:

$$\mathcal{W}_\eta^o = \arg\min_{\mathcal{W}} \sum_{k=1}^{K} \sum_{n=1}^{N} \left(\gamma_{k,n} - h_{k,n}^{\mathsf{T}} w_k\right)^2 + \frac{\eta}{2}\, \mathcal{W}^{\mathsf{T}}\, \mathcal{L}\, \mathcal{W} \tag{12.8}$$

for a suitably chosen parameter $\eta > 0$. The argument mirrors those in Chapter 1 for deriving Bayes' optimal learning problems and is left to the reader as an exercise. We illustrate samples of w_k^o for different choices of η in Fig. 12.3. Larger values of η yield more homogenous samples.

Having defined a smooth multitask learning problem (12.4), we may then pursue its optimal solution $\mathcal{W}_\eta^o$, either directly using stochastic gradient descent



$\eta = 0.001$, large $w^{\mathsf{T}} \mathcal{L}\, w$ $\eta = 0.005$, medium $w^{\mathsf{T}} \mathcal{L}\, w$ $\eta = 1$, small $w^{\mathsf{T}} \mathcal{L}\, w$

Figure 12.3 Gauss Markov random field samples.

(analogous to the argument leading to the consensus+innovations algorithms in Chapter 8), or using incremental stochastic gradient descent (analogous to the argument leading to the diffusion algorithm in Chapter 8). The incremental construction yields:

$$\boldsymbol{w}_i = (I - \mu\eta\mathcal{L}) \left(\boldsymbol{w}_{i-1} - \mu \widehat{\nabla \mathcal{J}}(\boldsymbol{w}_{i-1}) \right) \tag{12.9}$$

After returning to network level quantities:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{12.10}$$

$$\boldsymbol{w}_{k,i} = (1 - \mu\eta L_{k\ell})\boldsymbol{\psi}_{k,i} + \sum_{\ell \neq k} (-\mu\eta L_{k\ell})\boldsymbol{\psi}_{\ell,i} \tag{12.11}$$

We note two edge cases. If we set $\eta = 0$, it follows that $\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i}$, and agents' recursions decompose into non-cooperative learning structures. If, on the other hand $\eta = \mu^{-1}$, we recover the diffusion algorithm (8.62)–(8.63). In this sense, we can view (12.10)–(12.11) as "interpolating" between non-cooperative solutions and single-task learning. The degree to which of local intermediate estimates $\boldsymbol{\psi}_{k,i}$ occurs in (12.11) is a result of the level of smoothness desired in (12.4).

12.1.1 Performance of Multitask Learning under a Smoothness Prior

Since the multitask learning algorithm (12.10)–(12.11) can be seen to interpolate between a non-cooperative impelmentation for $\eta = 0$ and the diffusion algorithm for consensus optimization when $\eta = \mu^{-1}$, it is very natural to ask which choice for η will result in the best performance. If the local models are very homogeneous, we would expect a large η to perform well, while highly heterogeneous networks will likely perform better with small choices for η, allowing for high variability between local models. This intuition can indeed be formalized, though developing the performance analysis requires some adjustment to the arguments we encountered in Chapters 9 and 10. This is because algorithms for multitask learning no longer force agents to cluster around a network centroid, and hence the network deviation is no longer necessarily dominated by the dynamics of the

network centroid for small step-sizes. We refer the reader to [Nassif et al., 2020][1] for formal convergence analysis and guarantee, and instead plot the performance (guaranteed by analytical results) in Fig. 12.4.

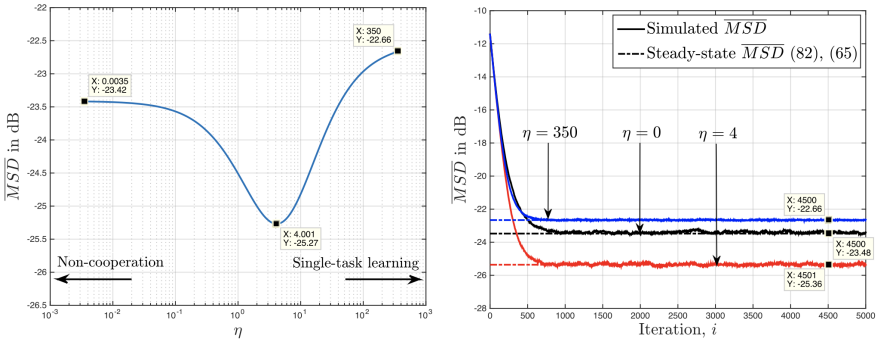


Figure 12.4 Performance of the multitask learning algorithm 12.10–(12.11) for varying choices of η. Analytical expressions for this curve are available in [Nassif et al., 2020].

12.1.2 Application: Weather Prediction

Suppose we would like to predict tomorrow's weather in different regions of the United States, based on a variety of different meteorological factors such as temperature and humidity. Whether it will rain on the following day can then be encoded in a binary class variables $\boldsymbol{\gamma}_k = \pm 1$, and the available measurements in a feature vector $\boldsymbol{h}_k$. We may then formulate a rudimentary linear weather model by training a logistic regression classifier locally:

$$w_k^o \triangleq \arg\min_{w_k} \mathbb{E} \ln\left(1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w_k}\right) + \frac{\rho}{2}\|w_k\|^2 \tag{12.12}$$

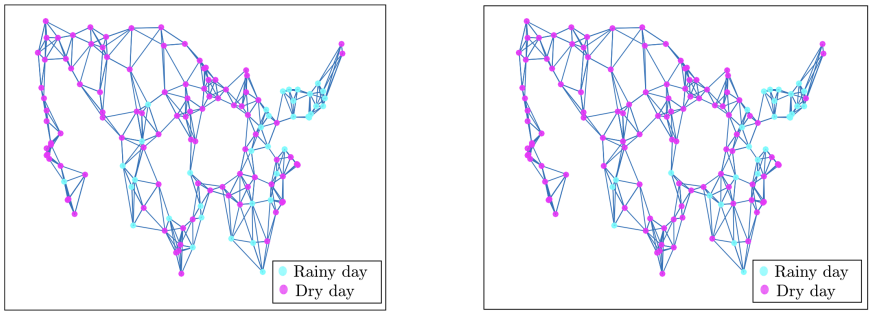
Since weather patterns are likely to be similar in nearby geographical regions, we may also envision a distributed learning schemes, where we encourage smoothenss in the local weather models w_k^o, resulting in:

$$\mathcal{W}_\eta^o \triangleq \arg\min_{\mathcal{W}} \sum_{k=1}^{K} \left(\mathbb{E} \ln\left(1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w_k}\right) + \frac{\rho}{2}\|w_k\|^2\right) + \frac{\eta}{2}\,\mathcal{W}^{\mathsf{T}}\,\mathcal{L}\,\mathcal{W} \tag{12.13}$$

We display in Fig. 12.5 the resulting predictions and performance.

12.2 SUBSPACE CONSTRAINED MULTITASK LEARNING

[1] R. Nassif, S. Vlaski, C. Richard and A. H. Sayed, "Learning Over Multitask Graphs—Part I: Stability Analysis," in IEEE Open Journal of Signal Processing, vol. 1, pp. 28-45, 2020, doi: 10.1109/OJSP.2020.2989038.



	$\eta = 0$	$\eta = 10$	$\eta = 45$	$\eta = 100$	$\eta = 1000$	$\eta = \mu^{-1}$
prediction error	0.309	0.232	**0.225**	0.226	0.228	0.232

Figure 12.5 Weather prediction using multitask learning with smoothness prior [Nassif et al., 2020]. (*Top left*) Actual occurrence of rain. (*Top right*) Predicted occurrence of rain. (*Bottom*) Prediction accuracy as a function of the regularization parameter η of (12.4).

An alternative model-based setting may be one where tasks are not necessarily smooth over the graph, but instead linearly related, i.e., $\mathcal{W} \in \text{Range}(\mathcal{U})$ for some $\mathcal{U}$.

$$\mathcal{W}^o = \arg\min_{\mathcal{W}} \sum_{k=1}^{K} J_k(w_k) \quad \text{s.t.} \quad \mathcal{W} \in \text{Range}(\mathcal{U}), \tag{12.14}$$

where $\text{Range}(\cdot)$ denotes the range space operator, and $\mathcal{U}$ is an $KM \times P$ full-column rank matrix with $P \ll KM$. A range constraint of this form encodes that there exists some linear relationship between the optimal models w_k^o. We illustrate some use-cases in the sequel:

Example 12.2 (**Consensus optimization**) We can actually recover the consensus problem (12.1) from (12.14) by setting $\mathcal{U} = \mathbb{1} \otimes I_M$. This is because for any x:

$$\mathcal{W} = (\mathbb{1} \otimes I_M)\, x \iff w_k = w_\ell \tag{12.15}$$

Example 12.3 (**Linearly-coupled optimization**) Consider a setting where each agent k is estimating a subset of the global weight vector $w = [w^1, w^2, w^3]$, with potential overlap among agents.

Returning to the subspace constrained learning problem (12.14), we can pursue an optimal model using *project stochastic gradient descent*:

$$\boldsymbol{\mathcal{W}}_i = \text{Proj}_{\mathcal{U}}\left(\boldsymbol{\mathcal{W}}_{i-1} - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{W}}_{i-1})\right) \tag{12.16}$$

Here, $\text{Proj}_{\mathcal{U}}(\cdot)$ denotes the Euclidean projection onto the range of $\mathcal{U}$, i.e.,:

$$\text{Proj}_{\mathcal{U}}(\mathcal{W}') = \arg\min_{\mathcal{W} \in \text{Range}(\mathcal{U})} \|\, \mathcal{W}' - \mathcal{W} \,\|^2 \tag{12.17}$$

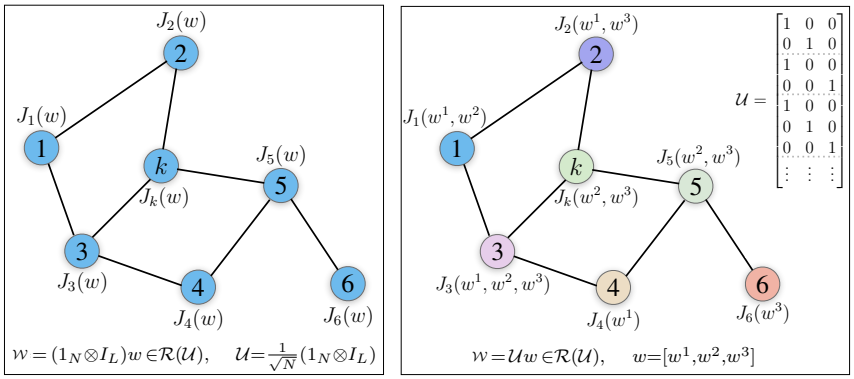


Figure 12.6 (*Left*) Consensus network and associated subspace matrix $\mathcal{U}$. (*Right*) Linearly coupled network and assocaited subspace matrix $\mathcal{U}$.

The projection onto the range of a matrix is a linear operation, taking the form:

$$\boldsymbol{\mathcal{w}}_i = \mathcal{U}(\mathcal{U}^{\mathsf{T}}\mathcal{U})^{-1}\mathcal{U}^{\mathsf{T}}\left(\boldsymbol{\mathcal{w}}_{i-1} - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{w}}_{i-1})\right) \tag{12.18}$$

Unfortunately, the matrix $\mathcal{U}(\mathcal{U}^{\mathsf{T}}\mathcal{U})^{-1}\mathcal{U}^{\mathsf{T}}$ is in general dense, and hence its implementation would require a fusion center. To avoid the need for central aggregation, we can replace the dense projection matrix by any combination matrix satisfying:

$$\lim_{i\to\infty} \mathcal{A}^i = \mathcal{U}(\mathcal{U}^{\mathsf{T}}\mathcal{U})^{-1}\mathcal{U}^{\mathsf{T}} \tag{12.19}$$

$$A_{k\ell} = [\mathcal{A}]_{k\ell} = 0, \quad \text{if } \ell \notin \mathcal{N}_k \text{ and } k \neq \ell. \tag{12.20}$$

Here, $A_{k\ell} = [\mathcal{A}]_{k\ell}$ denotes the $k\ell$-th $M \times M$ block of the matrix $\mathcal{A}$. We then arrive at a decentralized algorithm for decentralized multitask learning, which takes the form:

$$\boldsymbol{\mathcal{w}}_i = \mathcal{A}\left(\boldsymbol{\mathcal{w}}_{i-1} - \mu\widehat{\nabla\mathcal{J}}(\boldsymbol{\mathcal{w}}_{i-1})\right) \tag{12.21}$$

or after returning to node-level quantities:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{12.22}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} A_{k\ell}\boldsymbol{\psi}_{\ell,i} \tag{12.23}$$

12.2.1 Application: Decentralized Beamforming

We apply the framework to an approximate linearly-constrained minimum-variance (LCMV) beamforming problem. This kind of beamforming design gives rise to a quadratic optimization problem subject to a linear constraint, and hence fits into the subspace constrainend multitask learning framework. We refer the reader to

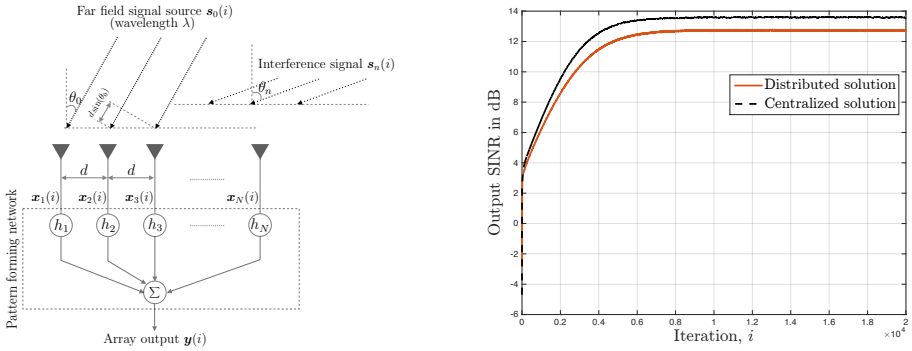


Figure 12.7 *(Left)* Uniform linear array of K antennas. *(Right)* Comparison of output SINR.

[Nassif et al., 2020][2] for details. Setup and performance is displayed in Fig. 12.7.

12.3 MODEL-AGNOSTIC META-LEARNING

Instead of directly modeling the relationship between tasks w_k^o and w_ℓ^o, in model-agnostic meta-learning one assumes that both one or several (stochastic) gradient step away from a common launch-model:

$$w_k^o \approx w^o - \mu \nabla Q(w^o; \boldsymbol{x}_k) \tag{12.24}$$

One then optimizes:

$$w^o \triangleq \arg\min_w \frac{1}{K} \sum_{k=1}^{K} \mathbb{E} Q(w - \mu \nabla Q(w; \boldsymbol{x}_k^1); \boldsymbol{x}_k^2) \tag{12.25}$$

to determine a common launch model w^o, which adapts quickly to other tasks w_k^o via one or several (stochastic) gradient steps. If we denote:

$$\overline{Q}(w; \boldsymbol{x}_k^1, \boldsymbol{x}_k^2) \triangleq Q(w - \mu \nabla Q(w; \boldsymbol{x}_k^1); \boldsymbol{x}_k^2) \tag{12.26}$$

then the optimization problem

$$w^o \triangleq \arg\min_w \frac{1}{K} \sum_{k=1}^{K} \mathbb{E} \overline{Q}(w; \boldsymbol{x}_k^1, \boldsymbol{x}_k^2) \tag{12.27}$$

is a *single-task* problem over the common launch model w, and can be pursued via any of the decentralized algorithms we have encountered so far. For example,

[2] R. Nassif, S. Vlaski and A. H. Sayed, "Adaptation and Learning Over Networks Under Subspace Constraints—Part I: Stability Analysis," in IEEE Transactions on Signal Processing, vol. 68, pp. 1346-1360, 2020, doi: 10.1109/TSP.2020.2970336.

using diffusion:

$$\begin{aligned}\boldsymbol{\phi}_{k,i} &= \boldsymbol{w}_{k,i-1} - \mu\nabla\overline{Q}(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}^1_{k,i}, \boldsymbol{x}^2_{k,i}) \\ &= \boldsymbol{w}_{k,i-1} - \mu\nabla Q(\boldsymbol{w}_{k,i-1} - \mu\nabla Q(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}^1_{k,i}); \boldsymbol{x}^2_{k,i}) \qquad (12.28)\end{aligned}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \qquad (12.29)$$

12.3.1 Application: Image Classification using Neural Networks

We illustrate performance on the ImageNet dataset in Fig. 12.8. Details are available in for the neural network architecture and implementation are available in [Kayaal et al., 2022].[3]

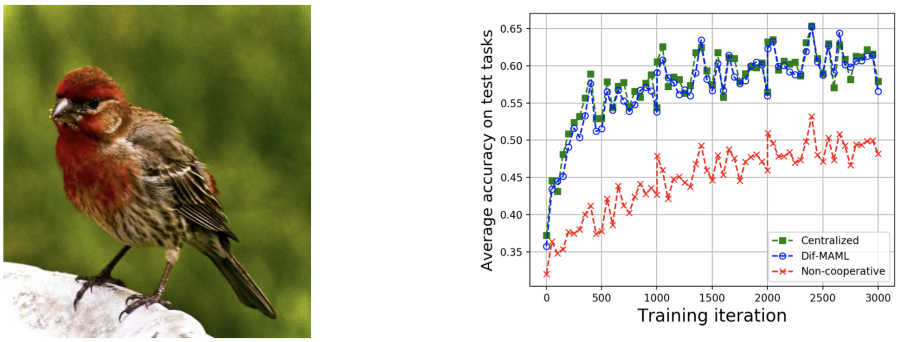


Figure 12.8 Performance of diffusion-based decentralized model-agnostic meta-learning on the ImageNet dataset [Kayaalp et al., 2022].

[3] M. Kayaalp, S. Vlaski and A. H. Sayed, "Dif-MAML: Decentralized Multi-Agent Meta-Learning," in IEEE Open Journal of Signal Processing, vol. 3, pp. 71-93, 2022, doi: 10.1109/OJSP.2021.3140000.