Bayesian multimodeling: multitask learning

MIPT

2024

What is multitask learning?

Multitask and dataset shift?

Linear regression case

$$\mathbf{Y} = \mathbf{X}^{\mathsf{T}}\mathbf{w} + \varepsilon,$$
$$\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{B}^{-1}), \quad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1}).$$

Task Clustering and Gating for Bayesian Multitask Learning

Consider a 1-layer neural network:

$$\mathbf{y}^i = \mathbf{W}^i_{\text{task}} \sigma(\mathbf{W}_{\text{shared}} \mathbf{x})$$

Can we set an interconnection of tasks?

Task Clustering and Gating for Bayesian Multitask Learning

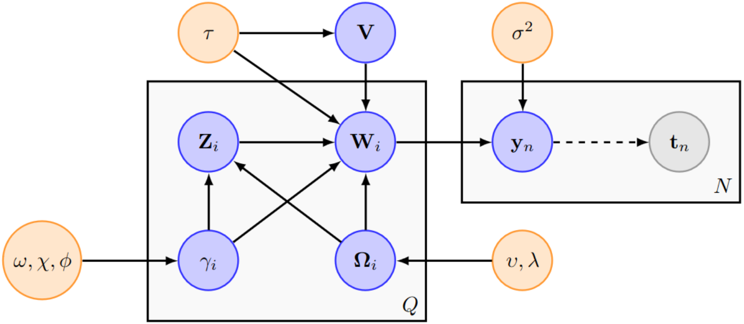
Interconnection of tasks:

- No: $\mathbf{W}^i_{\text{task}} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$;
- Gaussian mixture: $\mathbf{W}^i_{\text{task}} \sim \sum_j \alpha_j \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma})$;
- Using gating function: $\mathbf{W}^i_{\text{task}} \sim \sum_j \text{softmax}(\textbf{alpha})_j \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma})$;

Sparse Bayesian Multi-Task Learning, 2011

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\mu} + \varepsilon_n;$$

$$\varepsilon_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}).$$



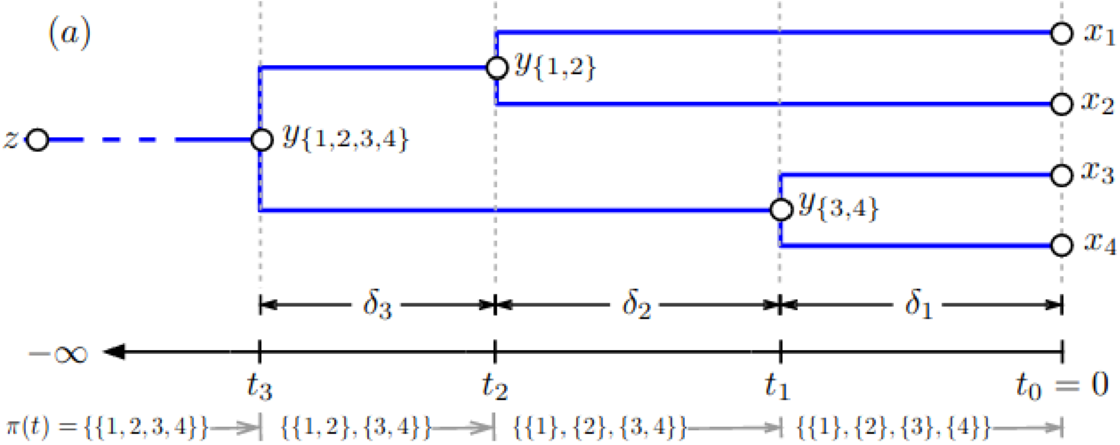
Multitask and domain adaptation?

Bayesian Multitask Learning with Latent Hierarchies

Daume III, 2009

We exploit the intuition that for domain adaptation, we wish to share classifier structure, but for multitask learning, we wish to share covariance structure.

Bayesian Multitask Learning with Latent Hierarchies



Bayesian Multitask Learning with Latent Hierarchies

1. Choose a global *mean* and *covariance* $(\boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}) \sim \mathcal{N}or\mathcal{IW}(0, \sigma^2\mathbf{I}, D+1)$. [2]

2. Choose a tree structure $(\pi, \boldsymbol{\delta}) \sim$ *Coalescent* over K leaves.

3. For each non-root node i in π (top-down):

   (a) Choose $\boldsymbol{\mu}^{(i)} \sim \mathcal{N}or(\boldsymbol{\mu}^{(p_\pi(i))}, \delta_i \boldsymbol{\Lambda})$, where $p_\pi(i)$ is the parent of i in π.

4. For each domain $k \in [K]$:

   (a) Denote by $\boldsymbol{w}^{(k)} = \boldsymbol{\mu}^{(i)}$ where i is the leaf in π corresponding to domain k.

   (b) For each example $n \in [N_k]$:

      i. Choose input $\boldsymbol{x}_n^{(k)} \sim \mathcal{D}^{(k)}$.

      ii. Choose output $y_n^{(k)}$ by:

      Regression: $\mathcal{N}or(\boldsymbol{w}^{(k)\top}\boldsymbol{x}_n^{(k)}, \rho^2)$

      Classification: $\mathcal{B}in(1/(1+e^{-\boldsymbol{w}^{(k)\top}\boldsymbol{x}_n^{(k)}}))$

Bayesian Multitask Learning with Latent Hierarchies

$\rightarrow$ 1. Choose $\mathbf{R}$ by Eq (2) and deviation covariance $\mathbf{\Lambda} \sim \mathcal{IW}(\sigma^2\mathbf{I}, D+1)$.

2. Choose a tree structure $(\pi, \boldsymbol{\delta}) \sim$ *Coalescent* over K leaves.

3. For each non-root node i in π (top-down):

   $\rightarrow$ (a) Choose $\mathbf{S}^{(i)} \sim \mathcal{N}or(\mathbf{S}^{(p_\pi(i))}, \delta_i \mathbf{\Lambda})$, where $p_\pi(i)$ is the parent of i in π.

4. For each task $k \in [K]$:

   $\rightarrow$ (a) Choose $\boldsymbol{w}^{(k)}$ by (i is the leaf associated with task k): $\mathcal{N}or\big(0, \big(\exp \mathbf{S}^{(i)}\big)\mathbf{R}\big(\exp \mathbf{S}^{(i)}\big)\big)$

   (b) For each example $n \in [N_k]$:

      $\rightarrow$ i. Choose input $\boldsymbol{x}_n^{(k)} \sim \mathcal{D}$.

      ii. Choose output $y_n^{(k)}$ by:

      Regression: $\mathcal{N}or(\boldsymbol{w}^{(k)\top}\boldsymbol{x}_n^{(k)}, \rho^2)$

      Classification: $\mathcal{B}in(1/(1+e^{-\boldsymbol{w}^{(k)\top}\boldsymbol{x}_n^{(k)}}))$

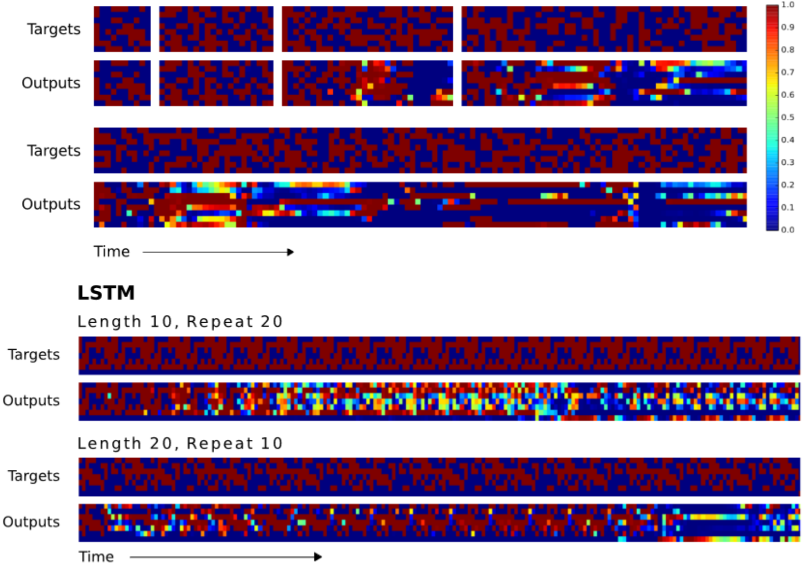
Automated Curriculum Learning, 2017

Algorithm 1 Intrinsically Motivated Curriculum Learning

Initially: $w_i = 0$ for $i \in [N]$

for $t = 1 \ldots T$ **do**
  $\pi(k) := (1 - \epsilon)\frac{e^{w_k}}{\sum_i e^{w_i}} + \frac{\epsilon}{N}$
  Draw task index k from π
  Draw training sample $\mathbf{x}$ from D_k
  Train network p_θ on $\mathbf{x}$
  Compute learning progress ν (Sections 3.1 & 3.2)
  Map $\hat{r} = \nu/\tau(\mathbf{x})$ to $r \in [-1, 1]$ (Section 2.3)
  Update w_i with reward r using Exp3.S (1)
end for

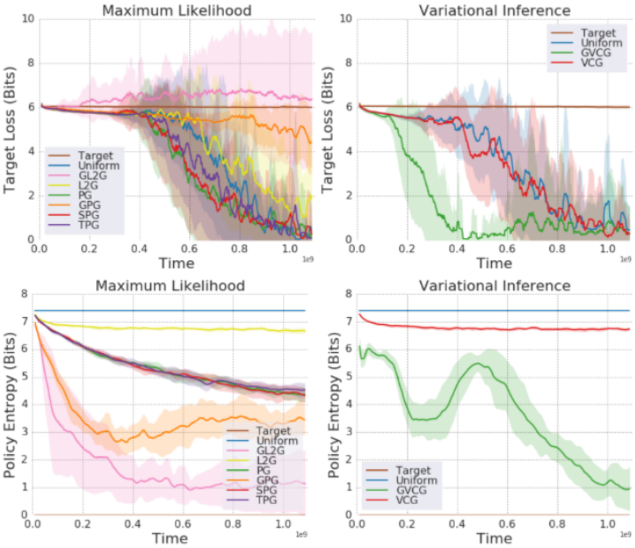
Repeat-copy task, 2014



Automated Curriculum Learning, 2017

- Loss-driven Progress
  - Prediction Gain: $L(\mathbf{w}', \mathbf{x}) - L(\mathbf{w}, \mathbf{x})$
  - Gradient prediction gain
  - Self-prediction gain: sampling $\mathbf{x}$
  - Mean prediction gain: averaging across the tasks
- Complexity-driven Progress
  - Variational complexity gain: $\mathrm{KL}(q|p) - \mathrm{KL}(q|p)$
  - Gradient Variational complexity gain
  - L2G: difference in l_2 regularization
  - GL2G: L2G gradient

Automated Curriculum Learning, 2017



Continual learning

Continual learning

Continual Learning is a concept to learn a model for a large number of tasks sequentially without forgetting knowledge obtained from the preceding tasks, where the data in the old tasks are not available any more during training new ones.

Three scenarios for continual learning

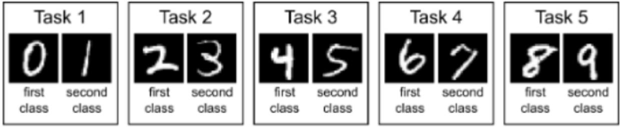
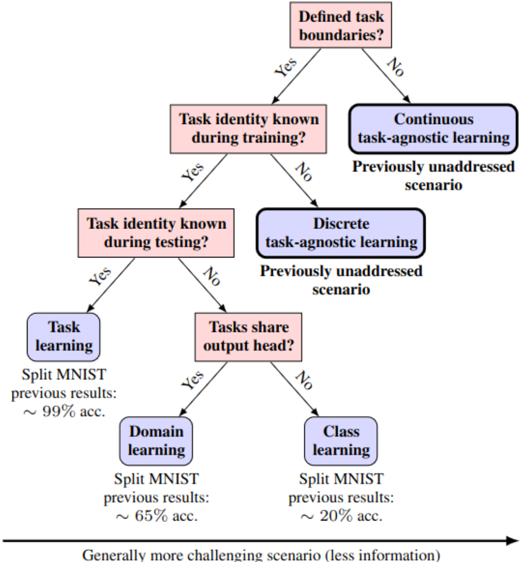


Figure 1: Schematic of split MNIST task protocol.

Table 2: Split MNIST according to each scenario.

Task-IL	With task given, is it the 1^{st} or 2^{nd} class? (e.g., 0 or 1)
Domain-IL	With task unknown, is it a 1^{st} or 2^{nd} class? (e.g., in $[0, 2, 4, 6, 8]$ or in $[1, 3, 5, 7, 9]$)
Class-IL	With task unknown, which digit is it? (i.e., choice from 0 to 9)

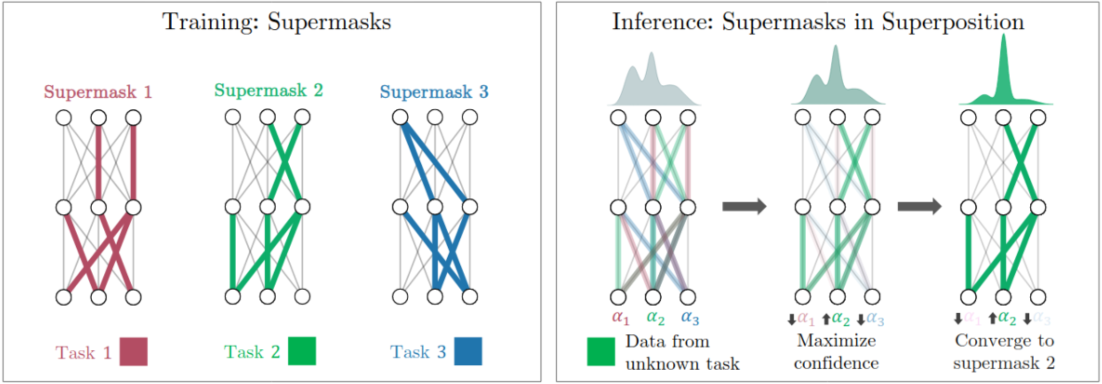
Task Agnostic Continual Learning Using Online Variational Bayes



Continual Learning: task categorization

Scenario	Description	Task space discreet or continuous?	Example methods / task names used
GG	Task **G**iven during train and **G**iven during inference	Either	PNN [42], BatchE [51], PSP [4], "Task learning" [55], "Task-IL" [49]
GNs	Task **G**iven during train, **N**ot inference; **s**hared labels	Either	EWC [23], SI [54], "Domain learning" [55], "Domain-IL" [49]
GNu	Task **G**iven during train, **N**ot inference; **u**nshared labels	Discrete only	"Class learning" [55], "Class-IL" [49]
NNs	Task **N**ot given during train **N**or inference; **s**hared labels	Either	BGD, "Continuous/discrete task agnostic learning" [55]

Supermasks in Superposition[1]



[1]See talk of Maria Kovaleva, 2023

Multitask learning and inductive bias[2]

Wiki

Multitask Learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better.

[2] First paper about this topic: Multitask Learning: A Knowledge-Based Source of Inductive Bias

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

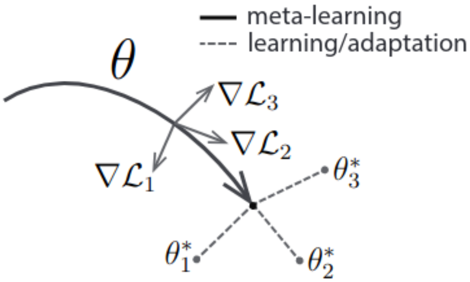
---

Algorithm 1 Model-Agnostic Meta-Learning

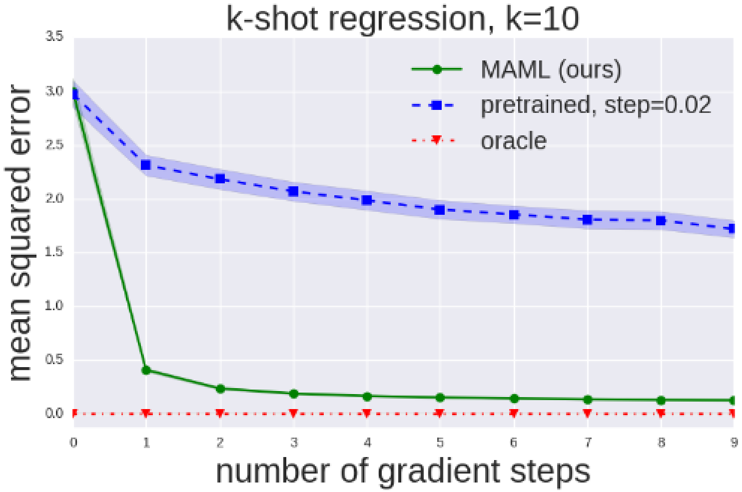
Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
end while

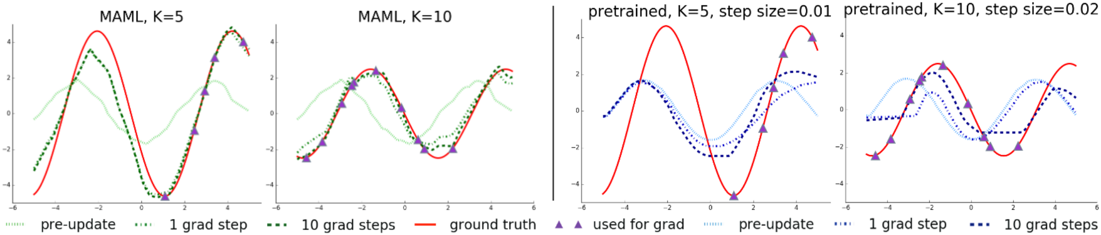
2



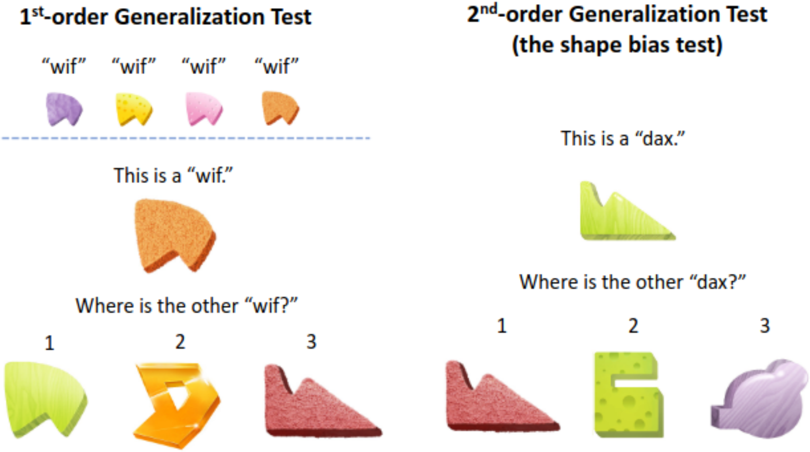
Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks



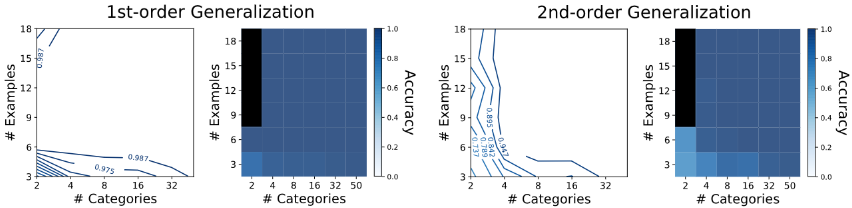
Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks



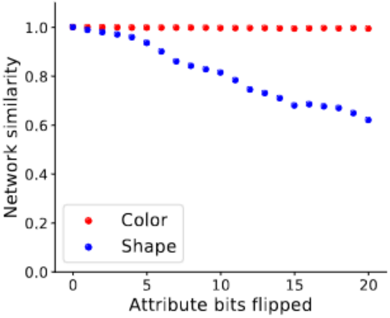
Learning Inductive Biases with Simple Neural Networks



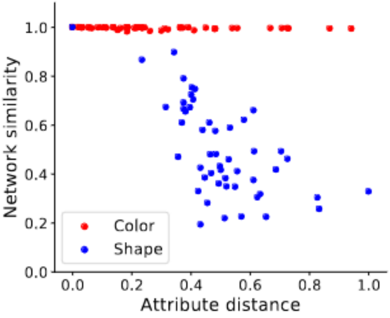
Learning Inductive Biases with Simple Neural Networks



Learning Inductive Biases with Simple Neural Networks

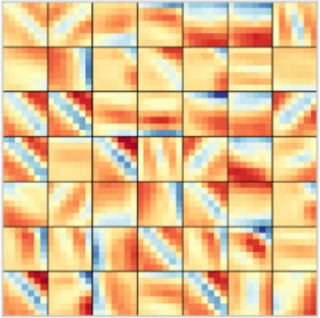


(a) MLP

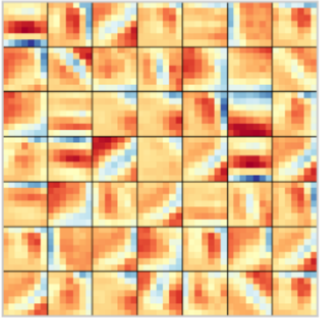


(b) CNN

The deep weight prior: Atanov et al., 2019



(b) Learned filters



(c) Samples from DWP

References

- Zeno C. et al. Task agnostic continual learning using online variational bayes //arXiv preprint arXiv:1803.10123. – 2018.
- Wortsman M. et al. Supermasks in superposition //Advances in Neural Information Processing Systems. – 2020. – T. 33. – C. 15173-15184.
- Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks."International conference on machine learning. PMLR, 2017.
- Caruana, R. "Multitask learning: A knowledge-based source of inductive bias."Proceedings of the Tenth International Conference on Machine Learning. 1993.
- Feinman, Reuben, and Brenden M. Lake. "Learning inductive biases with simple neural networks."arXiv preprint arXiv:1802.02745 (2018).
- Atanov, Andrei, et al. "The deep weight prior."arXiv preprint arXiv:1810.06943 (2018).

References

- Bakker B. J., Heskes T. M. Task clustering and gating for bayesian multitask learning. – 2003.
- Guo S., Zoeter O., Archambeau C. Sparse Bayesian multi-task learning //Advances in Neural Information Processing Systems. – 2011. – T. 24.
- Daume III H. Bayesian multitask learning with latent hierarchies //Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. – 2009. – C. 135-142.
- Graves A. et al. Automated curriculum learning for neural networks //international conference on machine learning. – PMLR, 2017. – C. 1311-1320.
- Graves A., Wayne G., Danihelka I. Neural turing machines //arXiv preprint arXiv:1410.5401. – 2014.
- Van de Ven G. M., Tolias A. S. Three scenarios for continual learning //arXiv preprint arXiv:1904.07734. – 2019.