Dalle Molle Institute for
Artificial Intelligence
USI – SUPSI

Exercise:
Meta Learning for Static Regression

SUPSI

The “sine” meta learning example

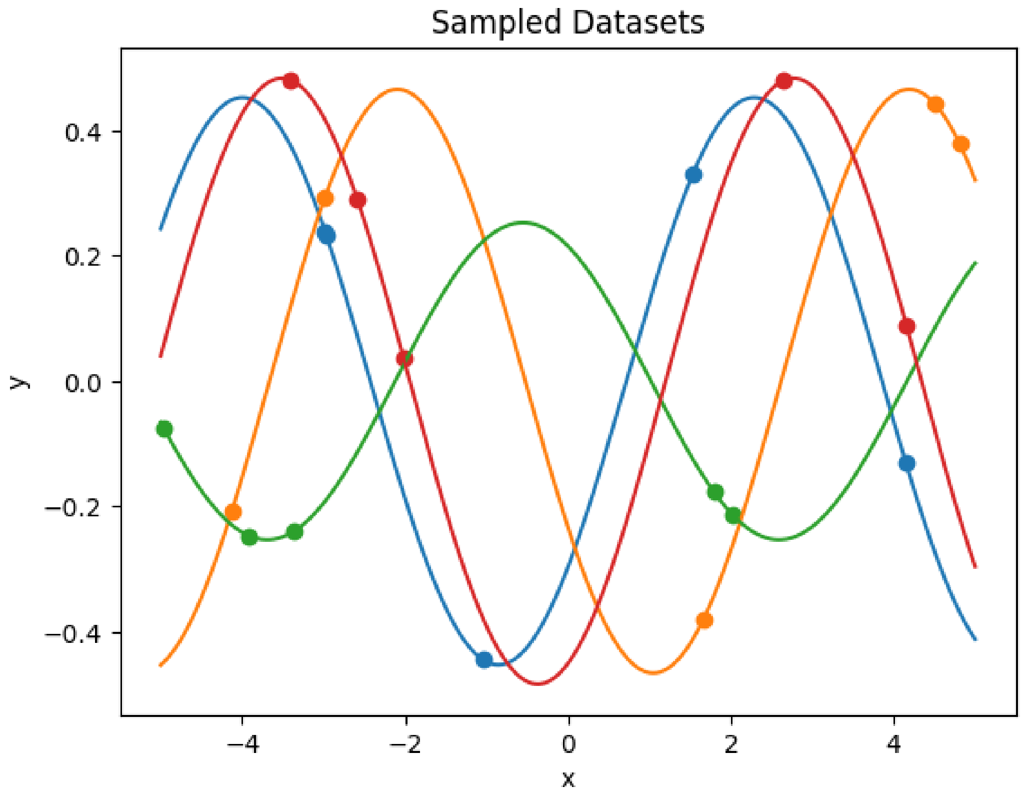
- Sines of varying phase and amplitude

$$y(x) = A\sin(x + \psi)$$

- Simple dependency, but just $K = 5$ points per dataset!

- Taken from the famous MAML paper

- The “Hello World” Meta Learning problem!



Meta Learning Setting

- We have access to a *meta dataset*, namely a collection (possibly unlimited) of datasets

$$\mathcal{D} = \{D_1, D_2, \ldots, \}$$

- In the case of static regression, each dataset D_i is an unordered collection of K input-output pairs.

$$D_i = \{(x_{i,1}, y_{i,1}), (x_{i,2}, y_{i,2}), \ldots, (x_{i,K}, y_{i,K})\}, \qquad x_{i,j} \in \mathbb{R}^{n_x}, \; y_{i,j} \in \mathbb{R}^{n_y}$$

- The datasets D_i are assumed to be $similar$ to each other. They are thought as realizations from a probability distribution $p(D)$

- Each dataset D_i is split in a train and a test portion:

$$\mathcal{D} = \{(D_1^{\mathrm{tr}}, D_1^{\mathrm{te}}), (D_2^{\mathrm{tr}}, D_2^{\mathrm{te}}), \ldots\}$$

Meta Learning: In-context learning

- We define an **meta-model** $\hat{y} = \mathcal{M}_\phi(x, D)$ which takes as input:
  - a test point x
  - a training dataset D (i.e., K input/output data points).

  It produces as output the prediction $\hat{y}$ corresponding to input x, given *as a context* the dataset D.

- Implicitly, the meta-model estimates a *system-specific* model on D and applies it to x

- We train the meta model by minimizing over the meta-dataset:

$$J(\phi) = \sum_{i=1}^{b} \sum_{j=1}^{K} \mathcal{L}(y^{te}_{i,j}, \mathcal{M}_\phi(D^{tr}_i, x^{te}_{i,j})).$$

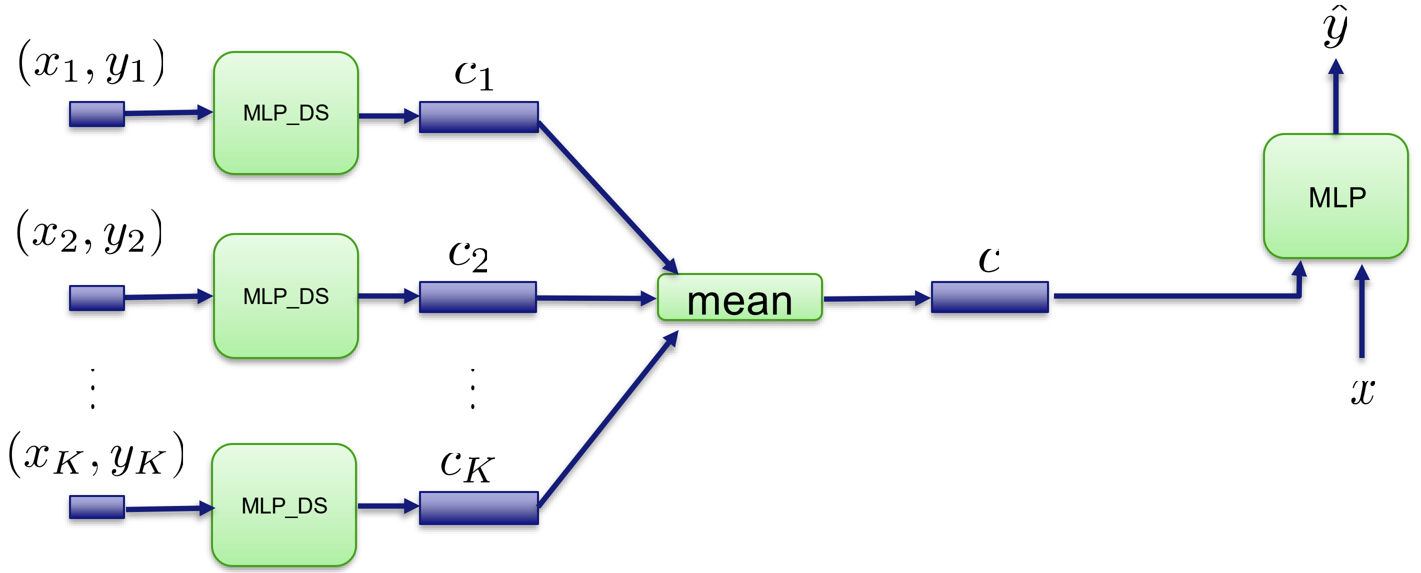
- The meta model learns to predict the output with just K input/output examples

In-context learning architecture for static regression

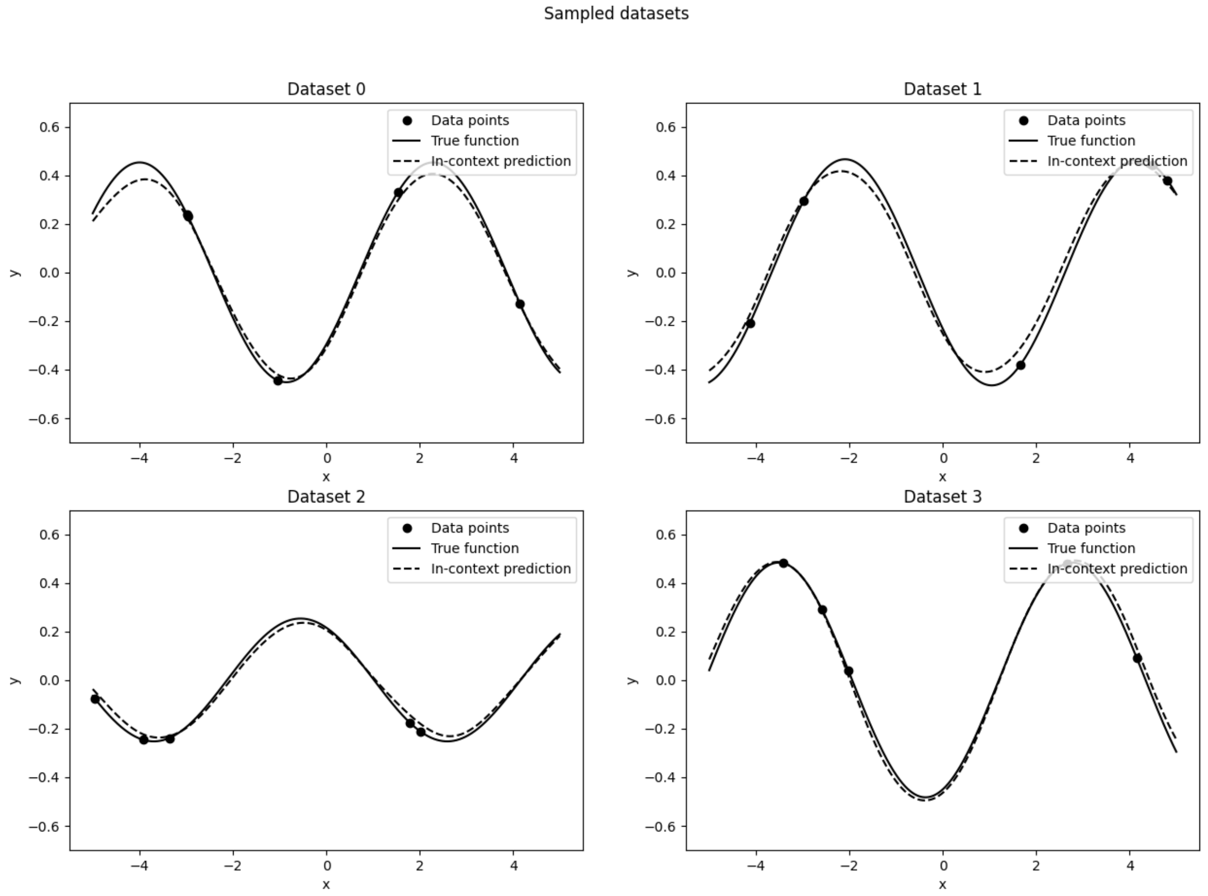
- The meta-model $\hat{y} = \mathcal{M}_\phi(D, x)$ has structure:

$$c = \mathrm{DeepSet}_{\phi_1}(D)$$
$$\hat{y} = \mathrm{MLP}_{\phi_2}(c, x).$$

DeepSet is invariant to permutations to the K input/output examples in D



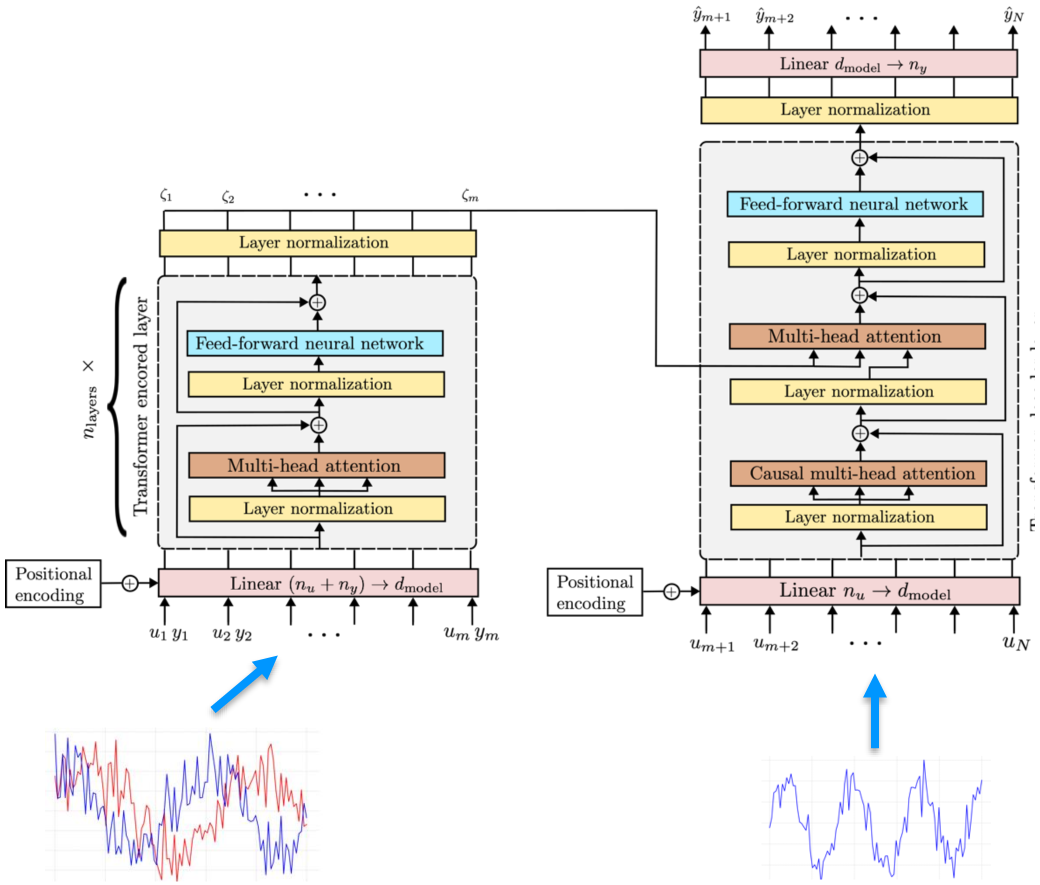
Performance with meta learning (in-context learning)



The meta model learns to make pretty good predictions with $K = 5$ data points

Dalle Molle Institute for Artificial Intelligence USI – SUPSI

In-context learning architecture for system identification



- May be seen as a simplification of the encoder-decoder Transformer for system identification

$$\hat{y}_{m+1:N} = \mathcal{M}_\phi(u_{m+1:N}, u_{1:m}, y_{1:m})$$

- The Transformer's attention layers are ideal to deal with intricate temporal dependencies of SYSID.

- For the sines example, the simple DeepSet+MLP architecture is more than enough…

M. Forgione, F. Pura, D. Piga, In-context learning for model-free system identification, IEEE Control Systems Letters, September 2023

SUPSI

Exercise

- Implementation of the in-context learning architecture for static regression DeepSet + MLP is sketched in in_context_learning_sketch.ipynb

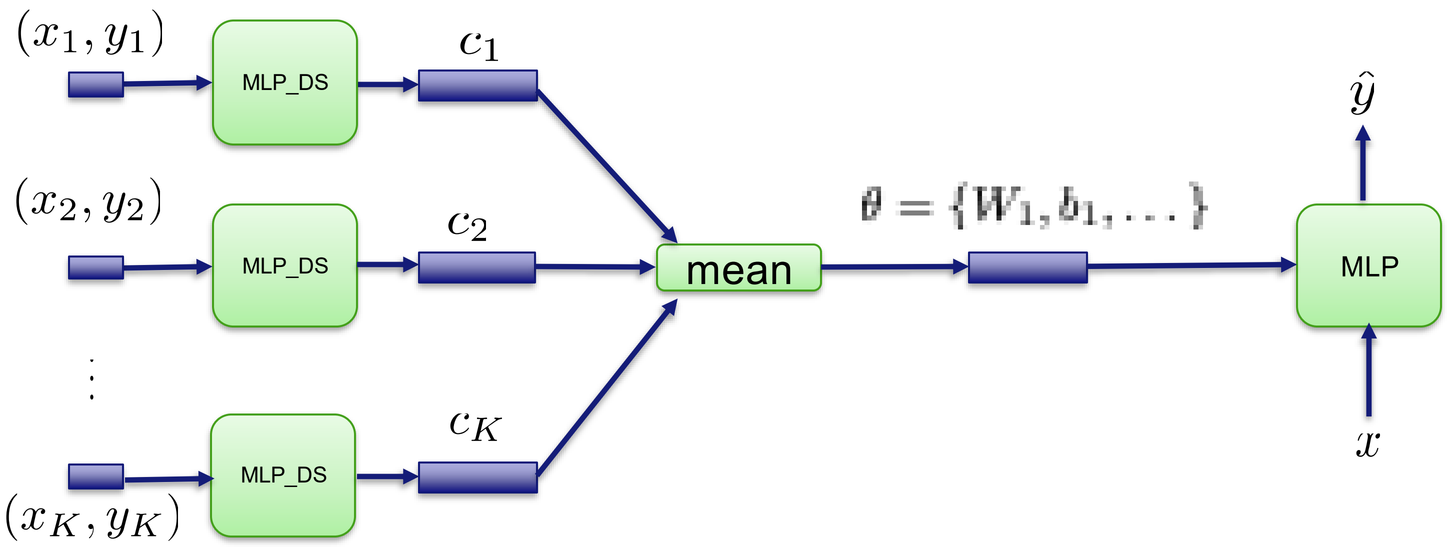
- Use it to tackle the meta learning exercise (meta_learning_exercise.ipynb)

- Optionally, apply hyper-networks and MAML (sketched in maml_hypernet_sketch.ipynb, described in the next slides…)

Meta Learning with Hyper-Nerworks

- A hyper-network provides weight and biases of the model MLP describing the relation $x \rightarrow y$

$$\theta = \mathrm{DeepSet}_\phi(D)$$
$$\hat{y} = \mathrm{MLP}_\theta(x).$$



- The hyper-network may be seen as a *learned* system identification algorithm

Gradient-Based Meta Learning (MAML)

- The learned algorithm has much more structure: it is one step of gradient descent!

$$\theta = \mathrm{Alg}_\phi(D) = \phi - \alpha \nabla_\theta \mathcal{L}(\theta, D),$$
$$\hat{y} = \mathrm{MLP}_\phi(x).$$

- We learn the best parameter ϕ such that, with one (or more) steps of gradient descent on the training dataset, performance measured on the test datasets is good.

$$J(\phi) = \sum_{i=1}^{b} \mathcal{L}(\mathrm{Alg}_\phi(D_i^{tr}), D_i^{te})$$

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." International conference on machine learning. PMLR, 2017.