# fine-tuning

## Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

## Synonyms

fine tuning

## Abstract

Fine-tuning continues the training of a model on a task-specific dataset, starting from model parameters obtained by pretraining rather than from a fresh initialization. It solves an empirical risk minimization (ERM) problem over a small training set, with the gradient descent (GD) iteration warm-started at the pretrained model parameters. A few GD steps move them a bounded distance, so the hypotheses the method can return lie in a neighborhood of the pretrained one rather than anywhere in the hypothesis space. That is a restriction of the model the method searches, imposed by the initialization instead of by a penalty term, and it is why a small training set suffices. Variants freeze some layers, or restrict the update to a low-dimensional set of new model parameters.

## Definition

A hospital has a few hundred annotated scans and needs a tumor detector. Training a deep net from a fresh initialization on that many labeled data points usually ends in overfitting; starting from model parameters already fitted to millions of unrelated photographs works markedly better (Yosinski et al., 2014). Fine-tuning is the second step of that recipe: it continues the training of a model on a task-specific dataset, starting from model parameters $\widehat{\mathbf{w}}^{(\mathrm{pre})}$ obtained by pretraining rather than from a fresh initialization. Large language models (LLMs) are adapted the same way: one pretrained on unlabeled text is fine-tuned on each task it is wanted for (Devlin et al., 2019).

Formally, fine-tuning solves the empirical risk minimization (ERM) problem over a training set $\mathcal{D}^{(\mathrm{train})} = \{\mathbf{z}^{(r)}\}_{r=1}^{m}$,

$$\widehat{\mathbf{w}} \in \arg\min_{\mathbf{w}} \frac{1}{m} \sum_{r=1}^{m} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right),$$

but the gradient descent (GD) iteration is warm-started at the pretrained model parameters, $\mathbf{w}^{(0)} = \widehat{\mathbf{w}}^{(\mathrm{pre})}$, instead of a fresh initialization.

The warm start is what makes a small training set enough. A few GD steps with step size $\eta$ move the model parameters a bounded distance from where they started, so the hypotheses the method can actually return lie in

a neighborhood of the pretrained one rather than anywhere in the hypothesis space (see Fig. 2). That restricts the model the method searches, which is what regularization achieves by pruning a hypothesis space, except that here the restriction comes from the initialization and no term is added to the objective function. The restriction is what a small training set needs in the probabilistic formalization: for data points that are realizations of independent and identically distributed (i.i.d.) random variables (RVs), a smaller set of reachable hypotheses narrows the generalization gap between training error and risk — for a finite hypothesis space, to at most $\sqrt{\log(2|\mathcal{H}|/\delta)/(2m)}$ with probability at least $1-\delta$ (Shalev-Shwartz and Ben-David, 2014, Cor. 4.6). The experiment `pythondemos/finetuning.py` measures the effect for linear regression with $m = 15$ data points and $d = 30$ model parameters: warm-started GD dips to a validation error below twice the noise level within twenty steps, moving the model parameters a distance 0.6, while the same iteration from a fresh initialization drives the training error to zero yet ends with a validation error more than four times larger, having moved a distance 2.6. A common variant freezes some layers and adapts only the remainder (Yosinski et al., 2014); parameter-efficient fine-tuning restricts the update to a low-dimensional set of new model parameters while leaving $\widehat{\mathbf{w}}^{(\mathrm{pre})}$ untouched (Hu et al., 2022). Fig. 1 shows the effect in the feature–label plane.

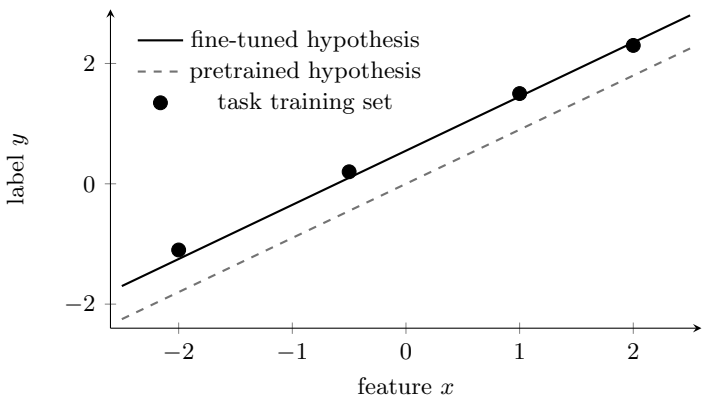


Figure 1: Fine-tuning in the feature–label plane. The pretrained hypothesis (dashed) is already close to the task; a few GD steps on the four data points of the task training set (dots) nudge it to the solid curve. From a fresh initialization, the four data points alone would have to determine the whole curve

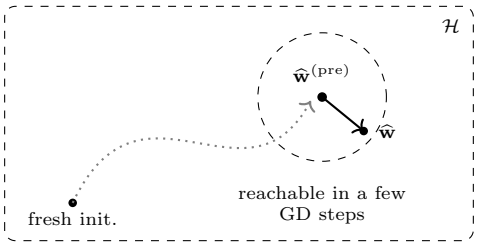


Figure 2: Fine-tuning in the space of model parameters. A few GD steps on a small training set cannot carry the model parameters far, so the hypotheses the method can return lie in the dashed neighborhood of $\widehat{\mathbf{w}}^{(\mathrm{pre})}$ rather than anywhere in the hypothesis space. Reaching the same region from a fresh initialization (dotted) would take far more labeled data points

## Cross References

pretraining, transfer learning, foundation model, self-supervised learning, ERM, regularization, overfitting, generalization gap, validation error

## References

1. Devlin et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proc. 2019 Conf. North American Chapter Assoc. Computational Linguistics: Human Lang. Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186.

2. Hu et al. (2022). LoRA: Low-Rank Adaptation of Large Language Models. In: Int. Conf. Learn. Represent. (ICLR).

3. Shalev-Shwartz and Ben-David (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge Univ. Press, New York, NY, USA.

4. Yosinski et al. (2014). How Transferable Are Features in Deep Neural Networks?. In: Adv. Neural Inf. Process. Syst. (NIPS), pp. 3320–3328.