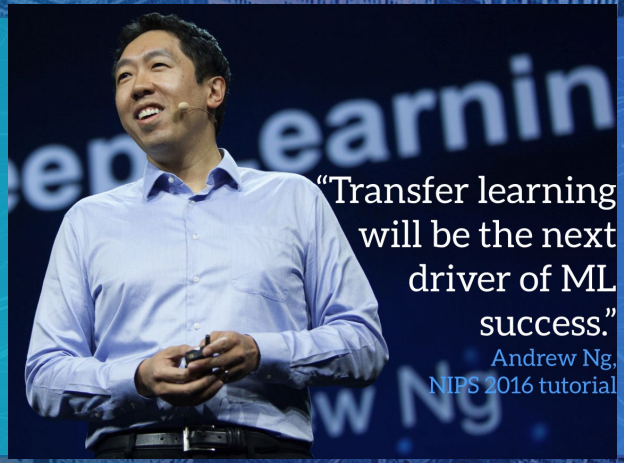
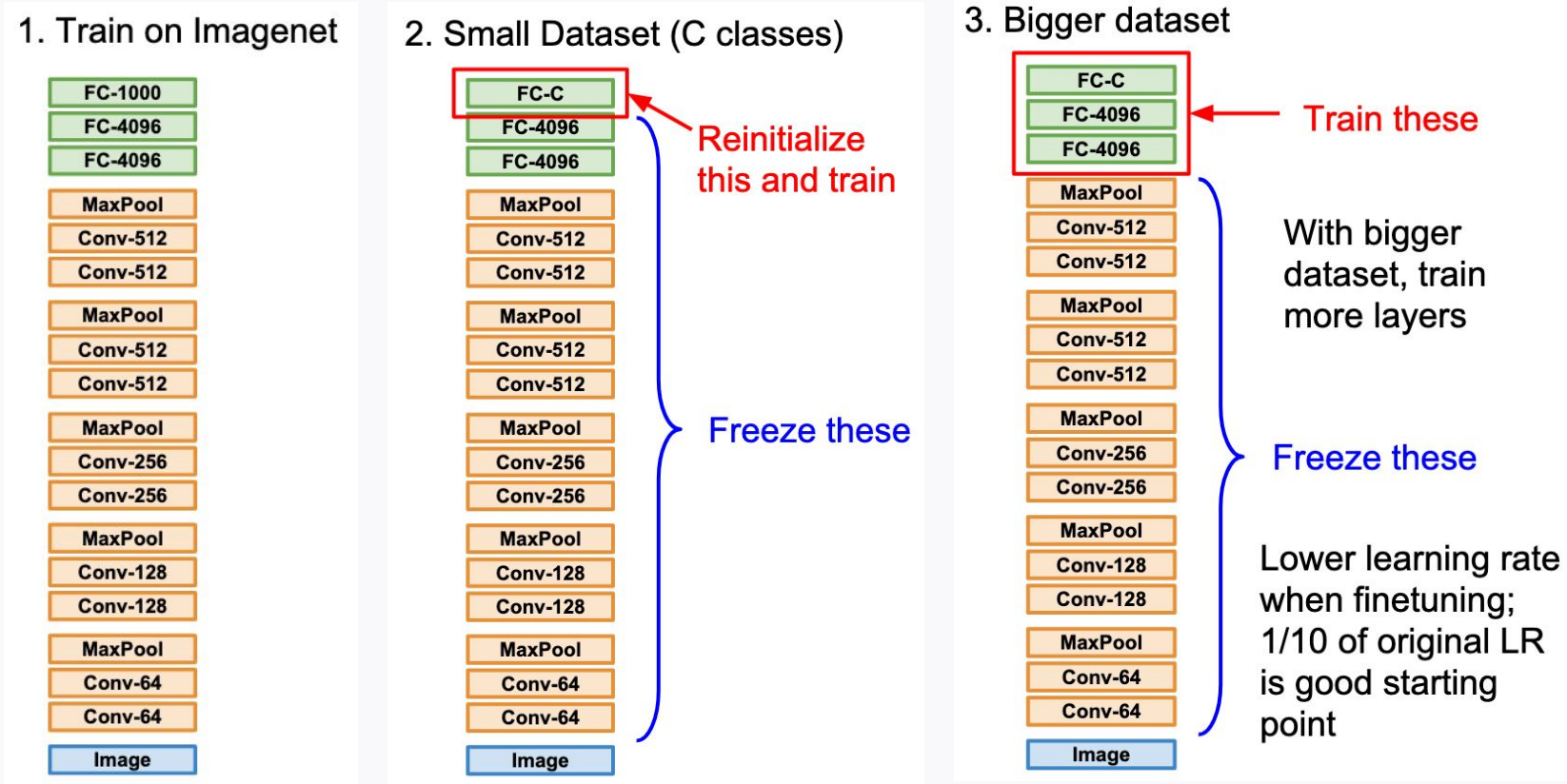
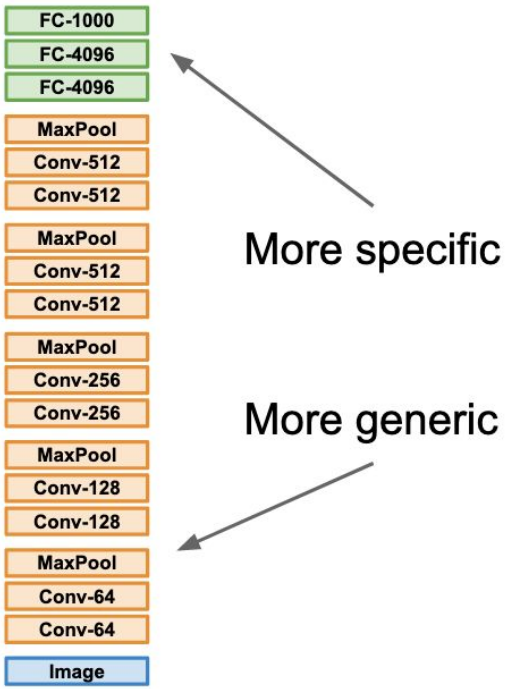


Transfer learning

Transfer learning key points



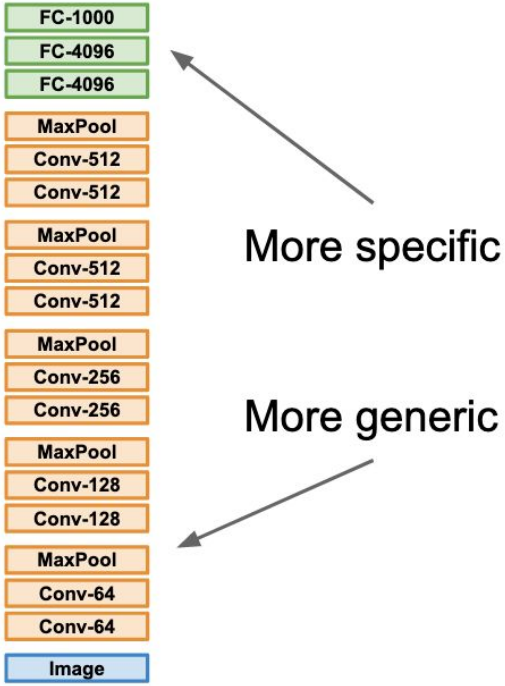
Transfer learning key points



cs231

	very similar dataset	very different dataset
very little data	?	?
quite a lot of data	?	?

Transfer learning key points



	very similar dataset	**very different dataset**
very little data	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

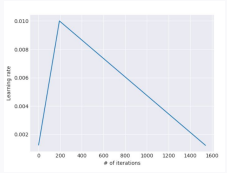
Universal Language Model Fine-tuning for Text Classification (ULMFiT)

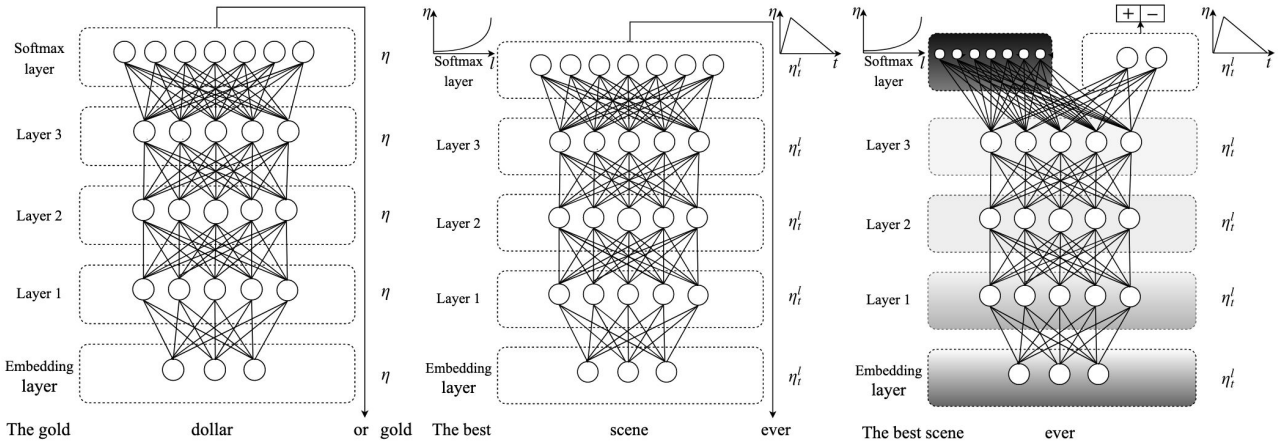
- Sebastian Ruder (DeepMind researcher, blogger) & Jeremy Howard (fast.ai)
- Discriminative fine-tuning

$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta^l} J(\theta)$$

We empirically found it to work well to first choose the learning rate η^L of the last layer by fine-tuning only the last layer and using $\eta^{l-1} = \eta^l/2.6$ as the learning rate for lower layers.

- Gradual unfreezing
- Slanted triangular learning rate



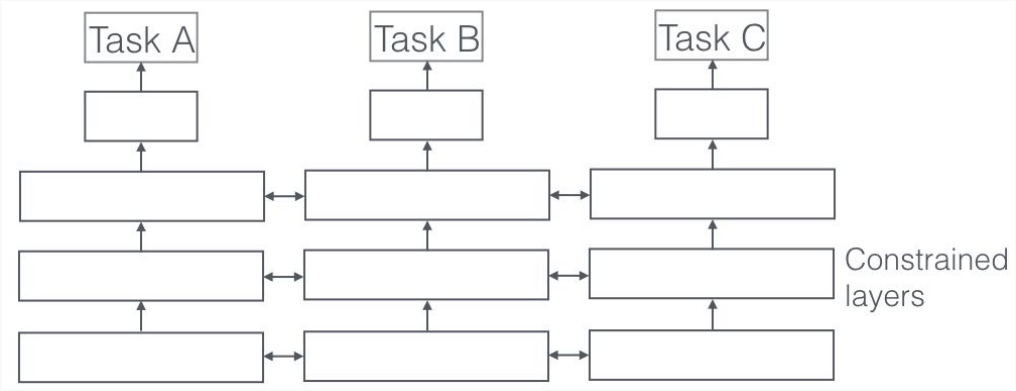


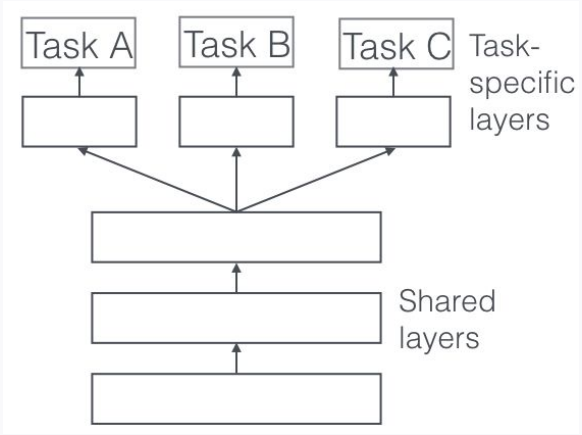
(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

Figure 1: ULMFiT consists of three stages: a) The LM is trained on a general-domain corpus to capture general features of the language in different layers. b) The full LM is fine-tuned on target task data using discriminative fine-tuning ('*Discr*') and slanted triangular learning rates (STLR) to learn task-specific features. c) The classifier is fine-tuned on the target task using gradual unfreezing, '*Discr*', and STLR to preserve low-level representations and adapt high-level ones (shaded: unfreezing stages; black: frozen).

https://arxiv.org/pdf/1801.06146.pdf

Multi-task learning





https://ruder.io/multi-task/

Tf.Hubs

- Transfer learning with Hubs
- Available models

Using pretrained CNN

- Transfer learning with a pretrained ConvNet