Artificial Neural Networks and Deep Learning

Week 6
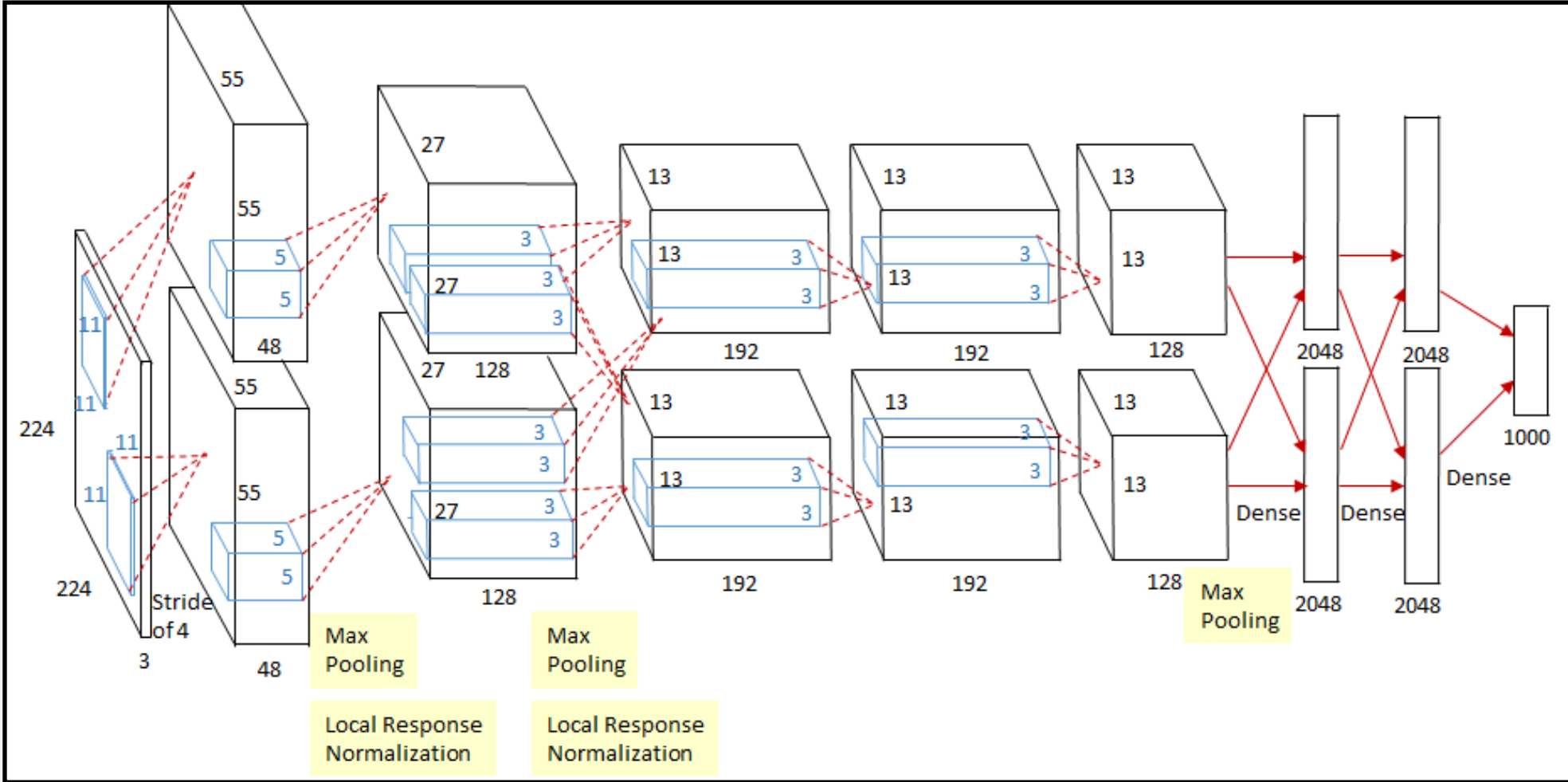
Transfer learning

Transfer learning

Reusing a model trained on one problem, on another problem

# The **problem** with training **big** deep learning models

- Extremely long training times (up to weeks)

- Expensive cloud computing fees, or GPU cost and electricity bills

- Huge CO_2 footprint (as much as 5 cars)

The **problem** with training **big** deep learning models
> **Solution:** *Reuse pre-trained models!*

- Extremely long training times (up to weeks)

- Expensive cloud computing fees, or GPU cost and electricity bills

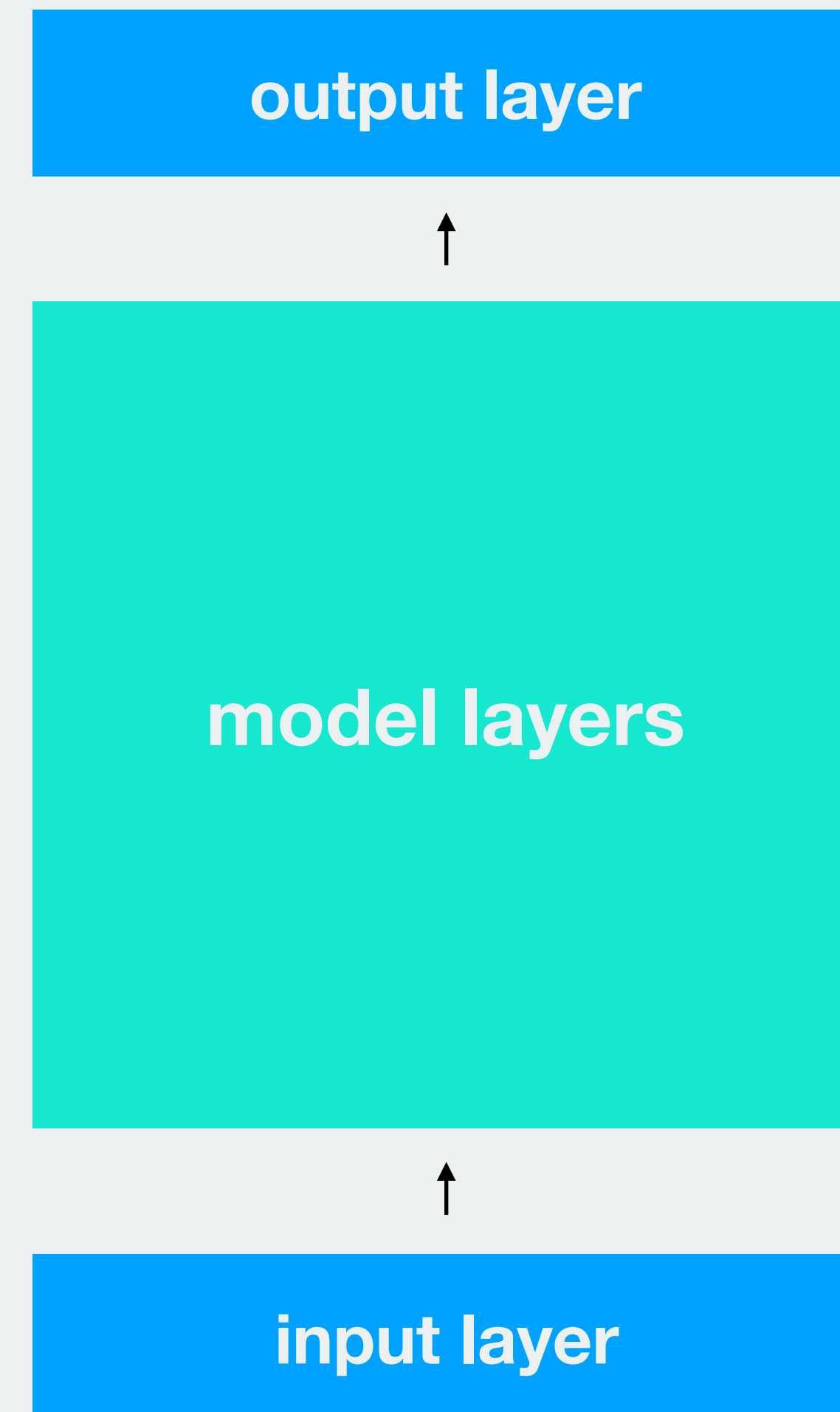- Huge CO_2 footprint (as much as 5 cars)
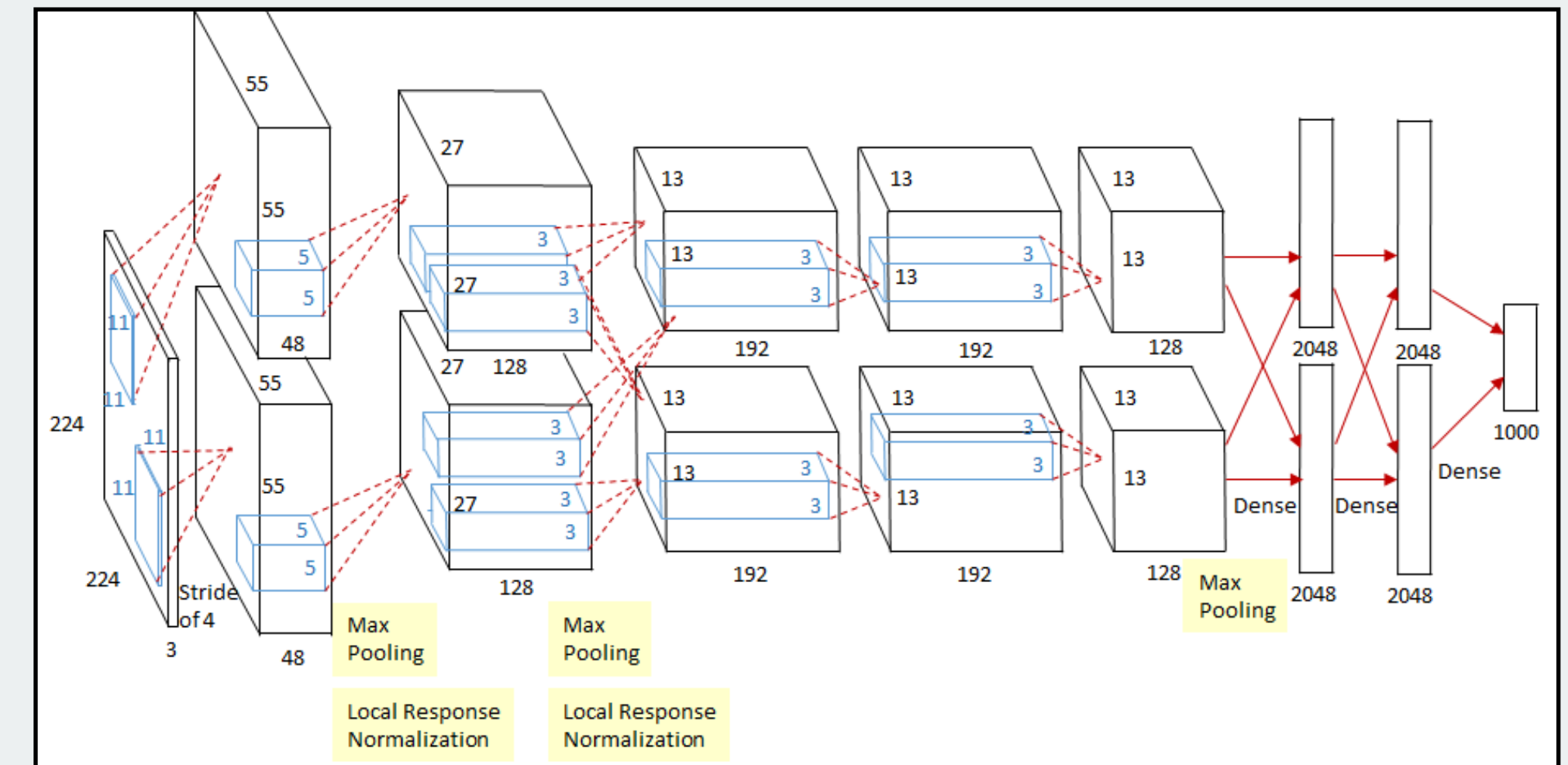
Transfer learning
> Fundamental idea
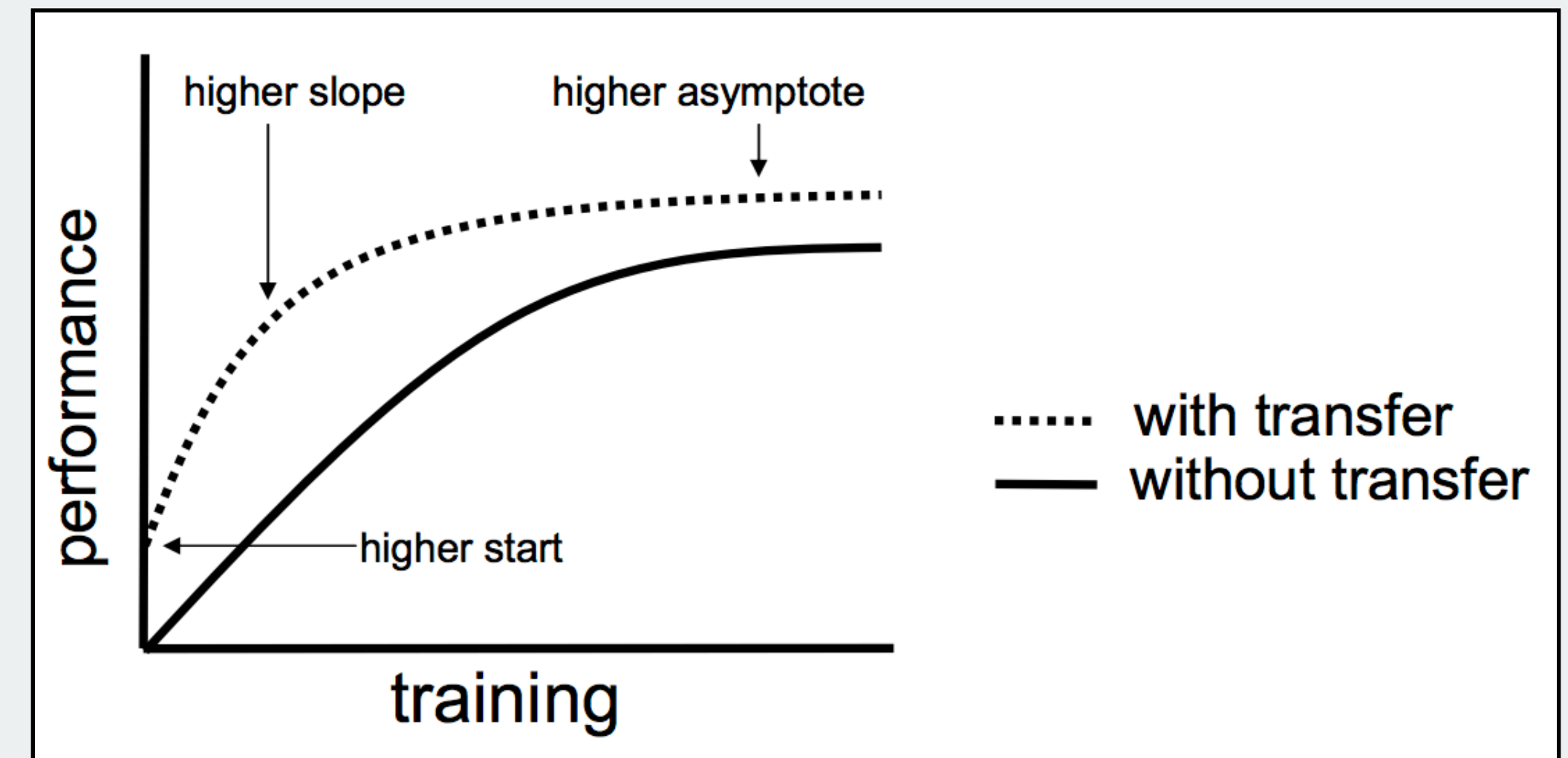
1. Train on one (huge) dataset

2. Reuse model to improve training on another dataset

output layer

↑

model layers

↑

input layer

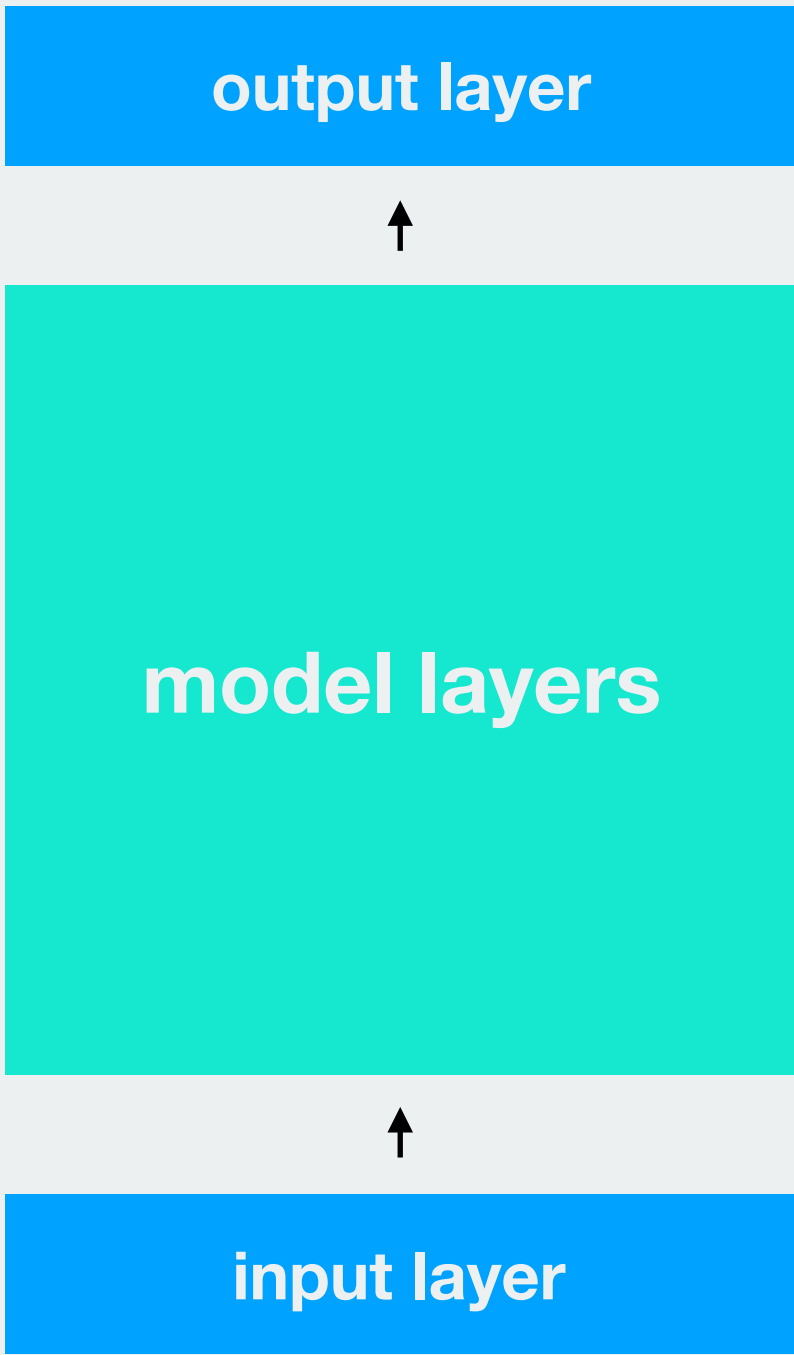**Transfer learning**

> Benefits

- Makes training on new data much faster

- Enables training on small datasets

- Helps avoid overfitting. Initial weights are usually better than random, helping avoid many local minima.
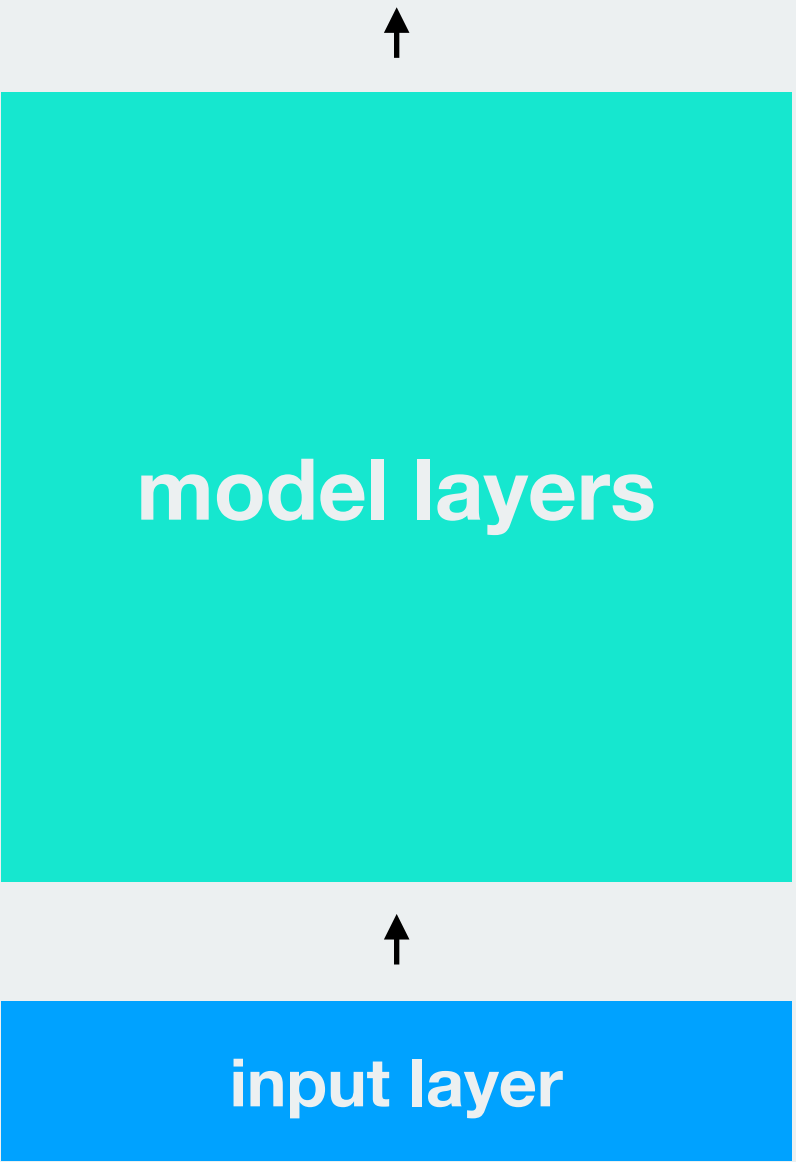
Transfer learning

> Fundamental idea (nuanced)

Pre-training dataset   Training dataset

1. Train on huge dataset

output layer
↑
| model layers |
↑
| input layer |

2. Use as feature extractor

↑
| model layers |
↑
| input layer |

3. Train new i/o layers

new output layer
↑
| model layers |
↑
| new input layer |
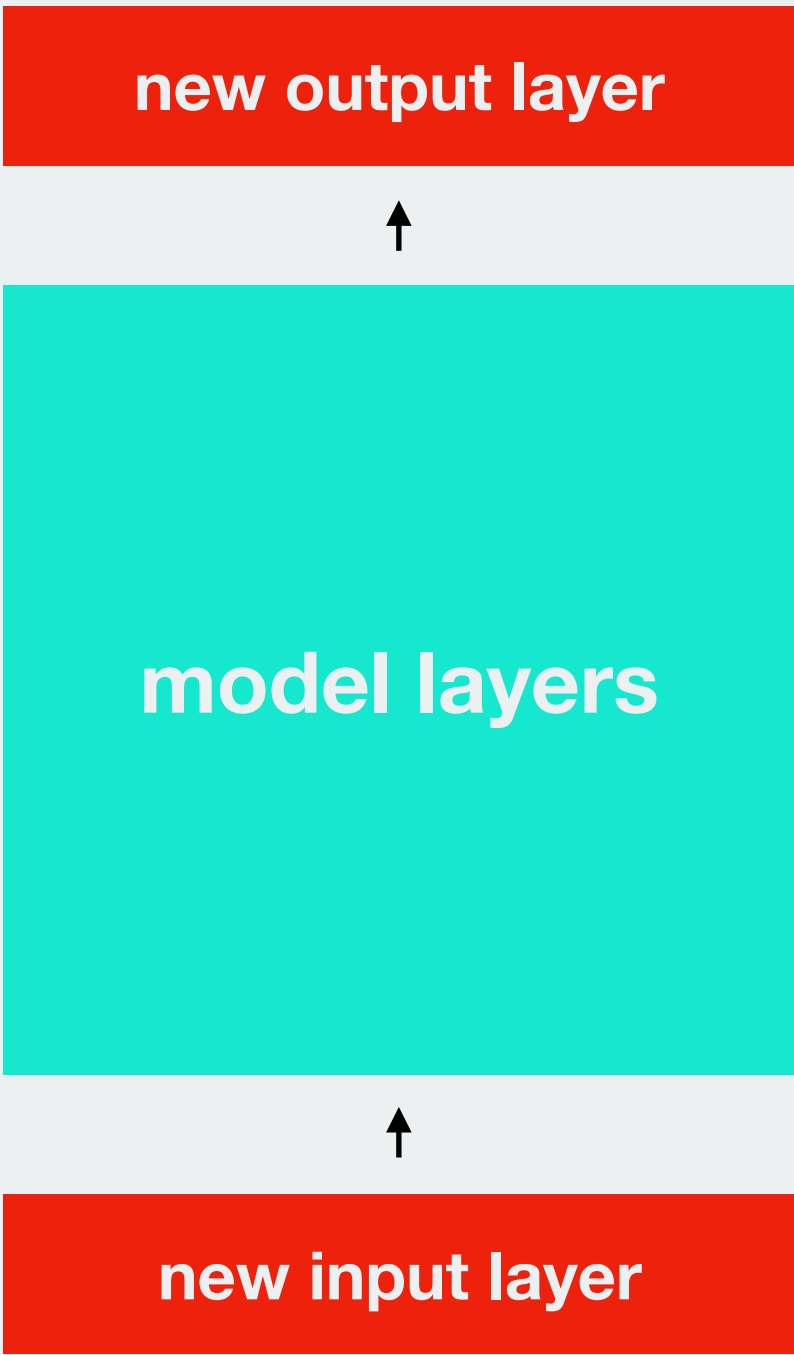
4. Continue training model

output layer
↑
| model layers |
↑
| input layer |
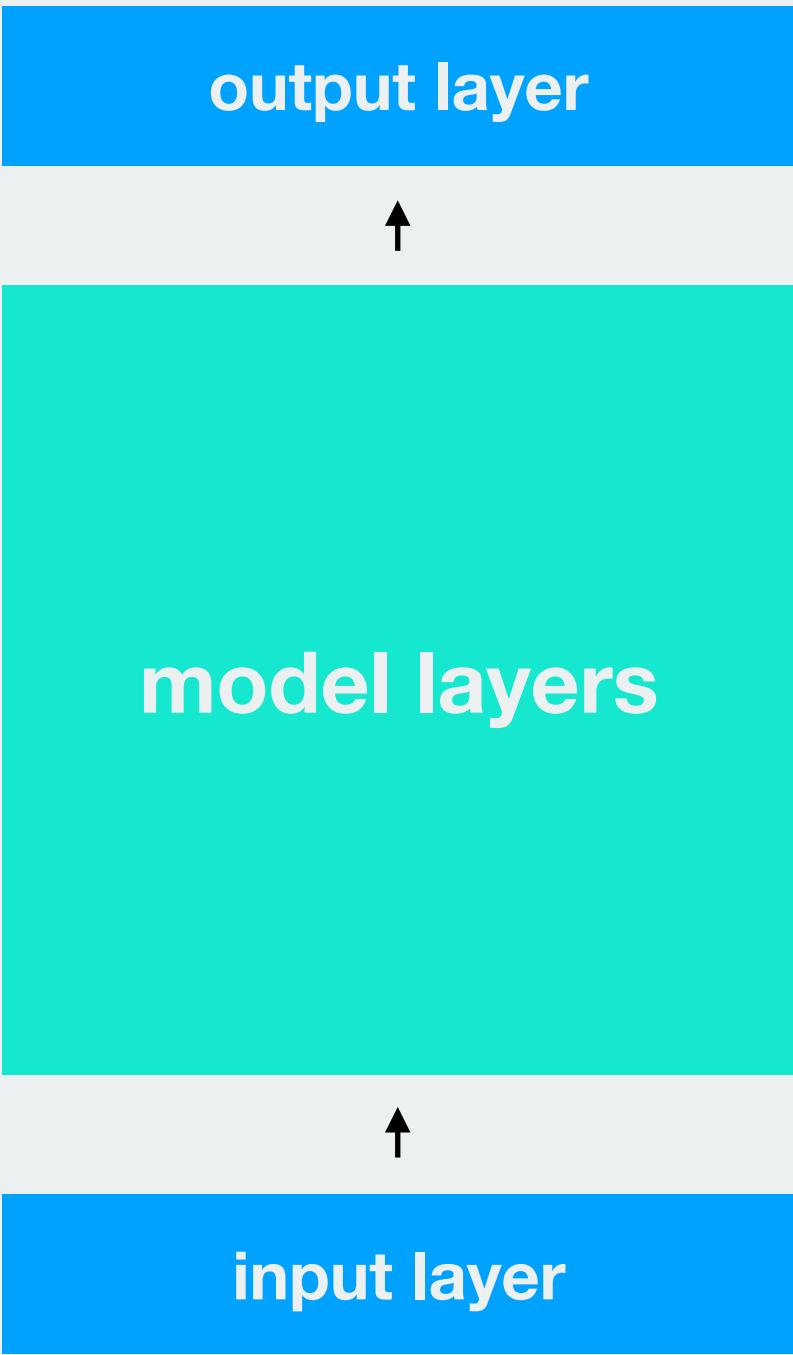
Transfer learning
> Strategies

1. If training dataset is **small** only train **last layers**

2. If training dataset is **big** train more **last layers**

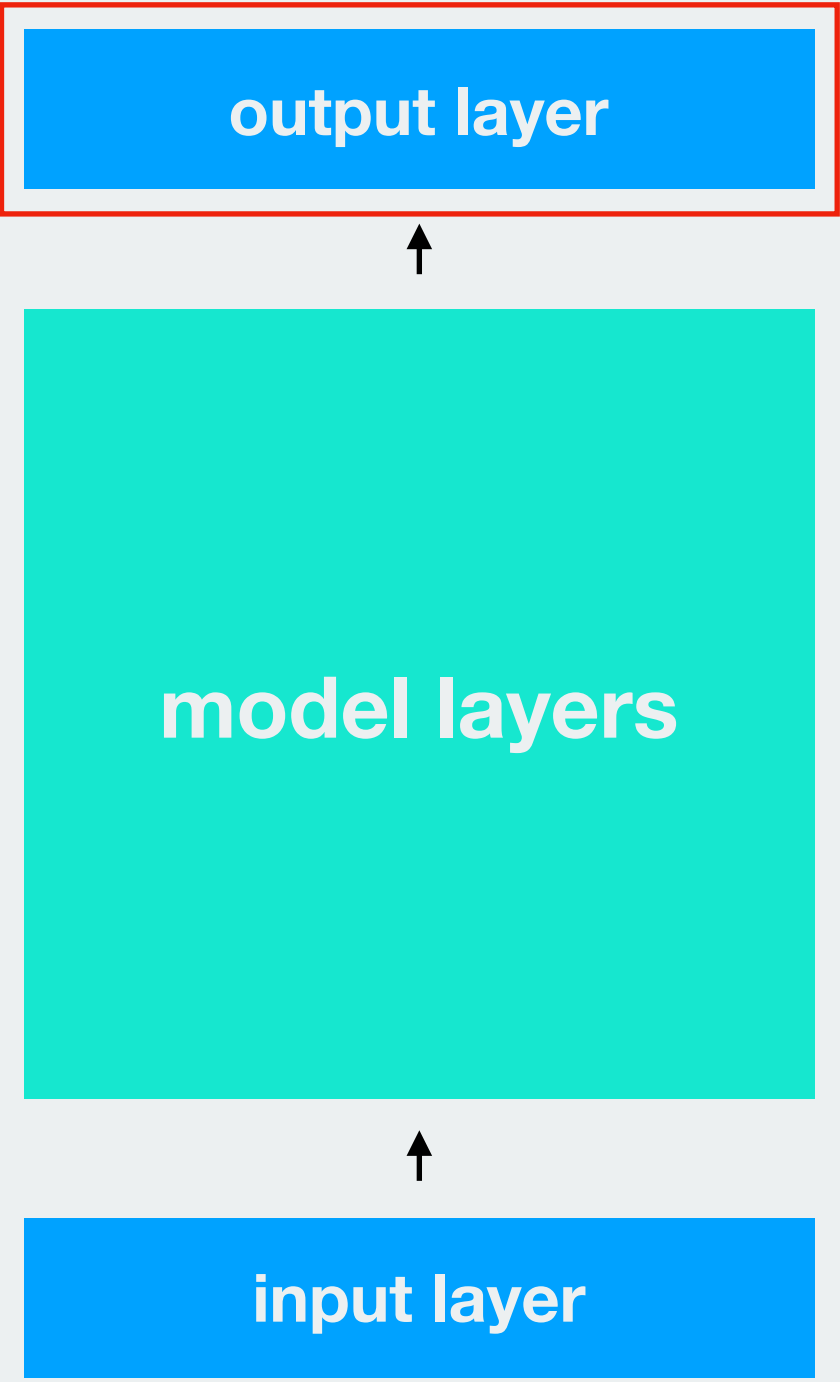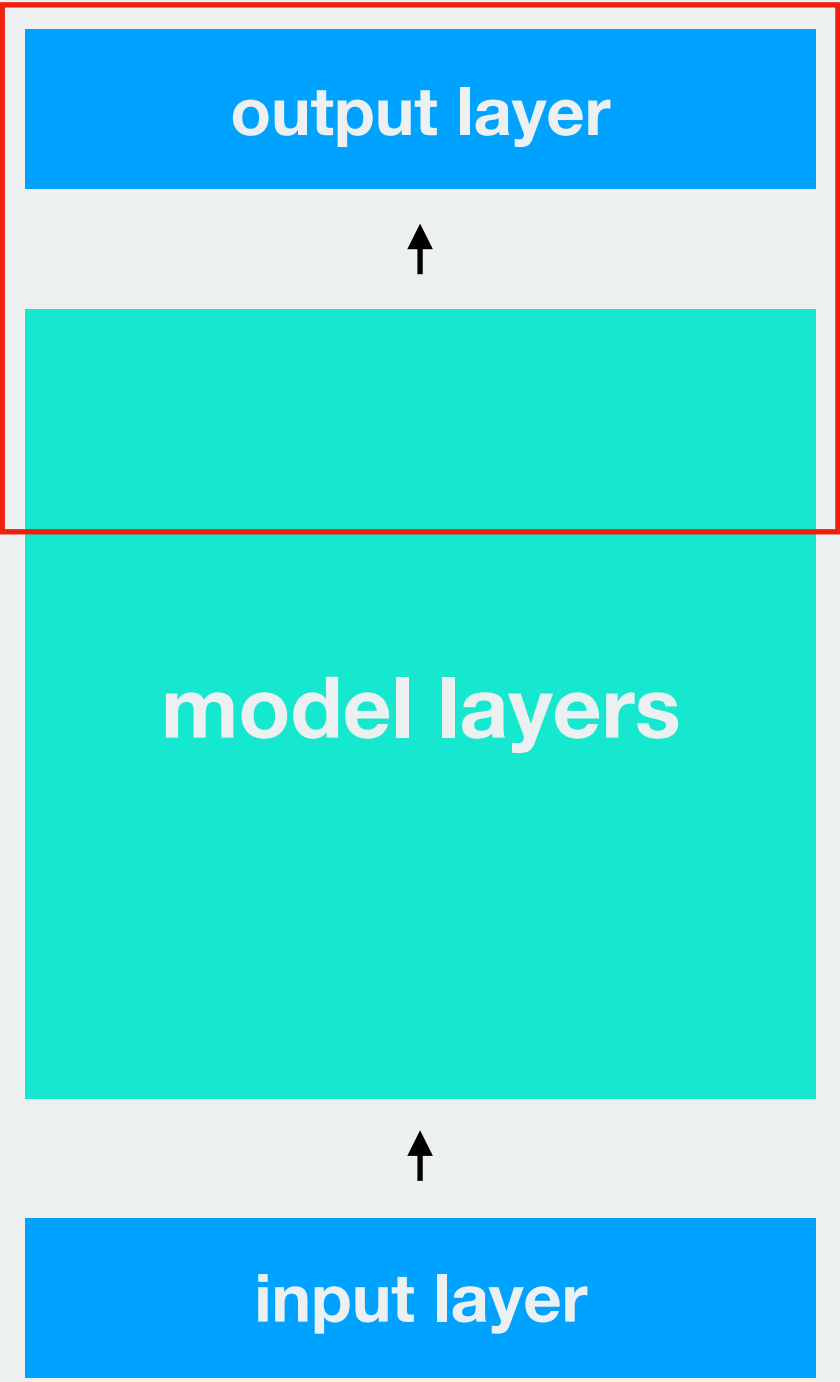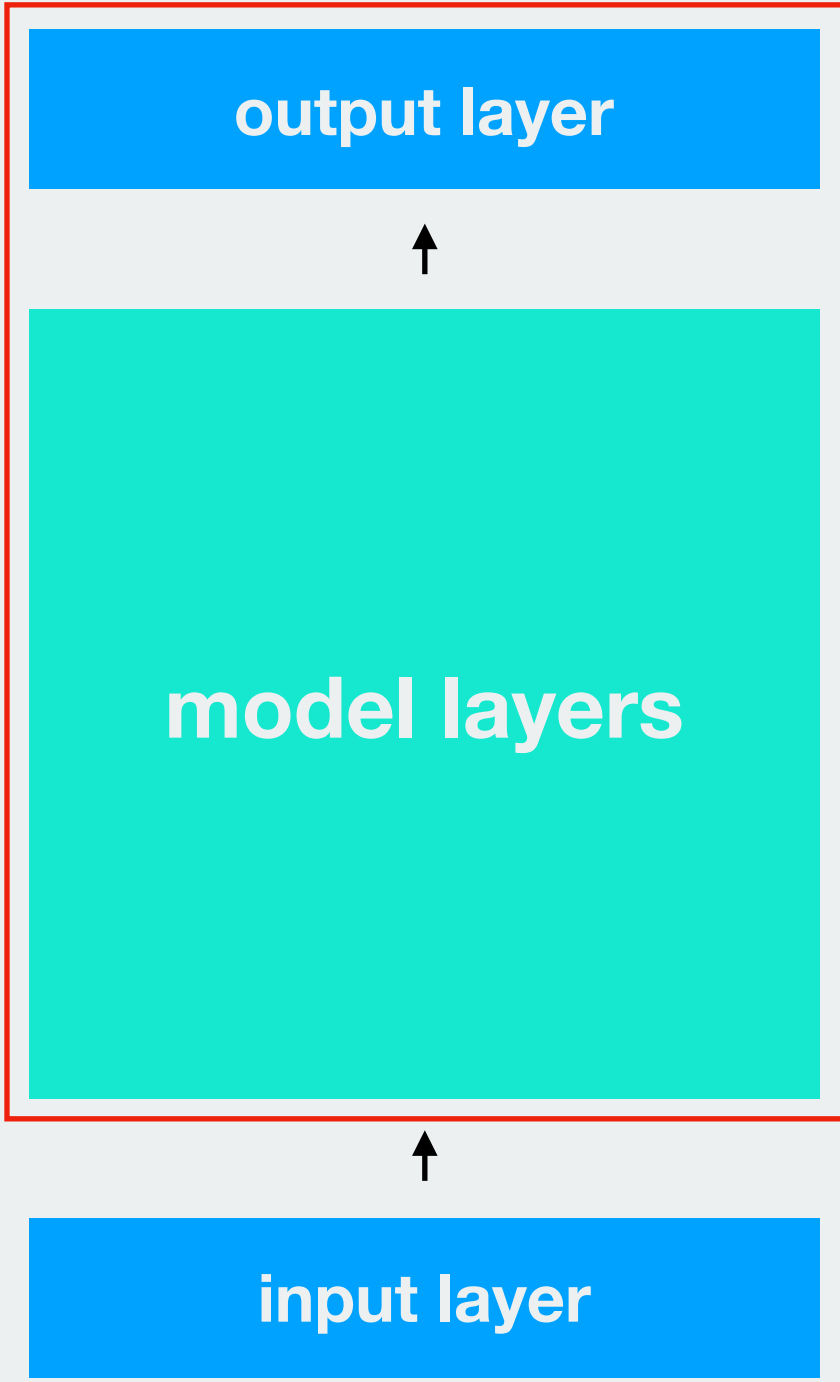3. If training dataset is **huge** train all **layers** with reduced learning rate. 1/10th of orig. LR is good choice

Transfer learning
> Strategies

	Similar dataset	**Different dataset**
Little data	Use pre-trained model as feature extractor and do classification with new features and simpler model	Difficult. Maybe consider using a different pre-trained model or use different feature extractors
Much data	Finetune a few layers towards the end of the network, with lowered LR	Finetune a large number of layers, with lowered LR

Transfer learning

> Further

- Transfer learning is extremely pervasive, especially for image data

- Also used for language modeling. There exists publicly available *word embeddings* which encode words as vectors in an efficient way.

- Most research and industrial projects start with some pertained model and then builds something on top of that.

Object Detection (Fast R-CNN)

Log loss + smooth L1 loss

Proposal classifier — Linear + softmax

Bounding box regressors — Linear

FCs

RoI pooling

External proposal algorithm e.g. selective search

ConvNet (applied to entire image)

Image Captioning: CNN + RNN

"straw"   "hat"   END

y_t

W_{hh} W_{oh}

h_t

CNN_{θ_c} W_{hi}

W_{hx}

x_t

START   "straw"   "hat"