kokchun giang

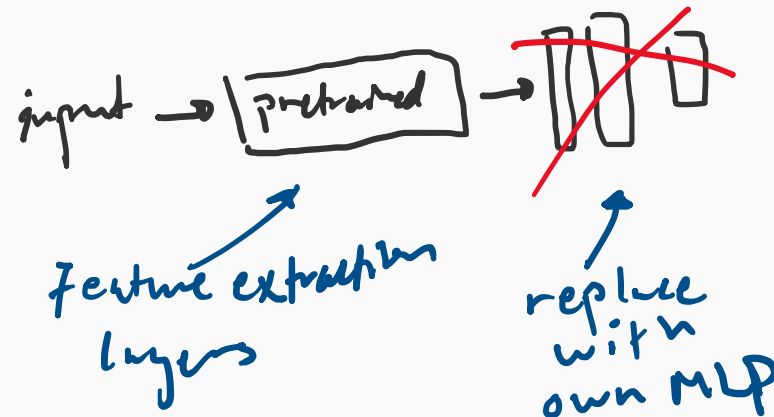# **transfer learning -** don't reinvent the wheel, instead reuse the heavylifting

use a pretrained network as feature extractor

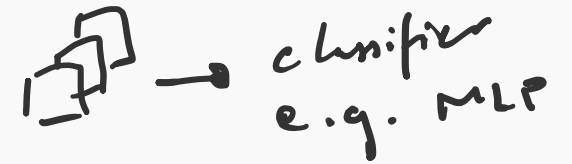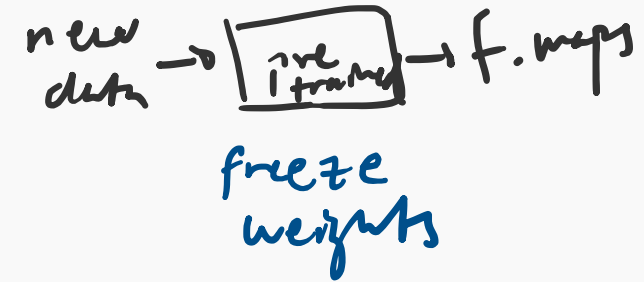→ learning from scratch takes long time & require lots of trainy data

⇒ take a network pretrained on a similar problem
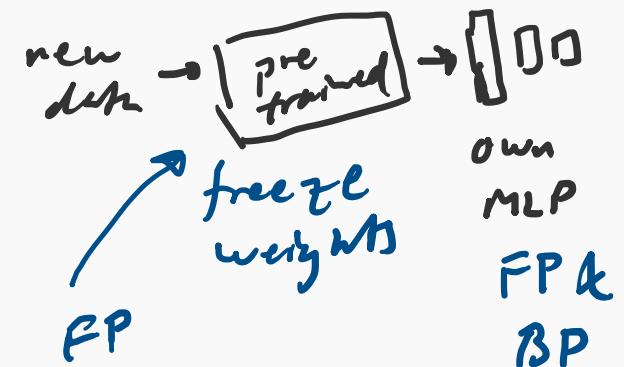
<u>Ex</u> network trained on imagenet dataset

→ reuse weights & kernels to <u>extract</u> <u>feature maps</u>

input → [pretrained] → ▯▯ ▯

feature extraction layers

replace with own MLP

<u>Method 1</u> - fast feature extraction

new data → [pretrained] → f. maps

freeze weights

▯▯ → classifier e.g. MLP

<u>Method 2</u> Feature extraction

new data → [pretrained] → ▮▯▯ ▯

freeze weights

own MLP

FP

FP & BP

fine tuning

Method 3 fine tuning

new data → [pretrained ▨] → ▮ ▯ ▫ own MLP

freeze   adjust weights
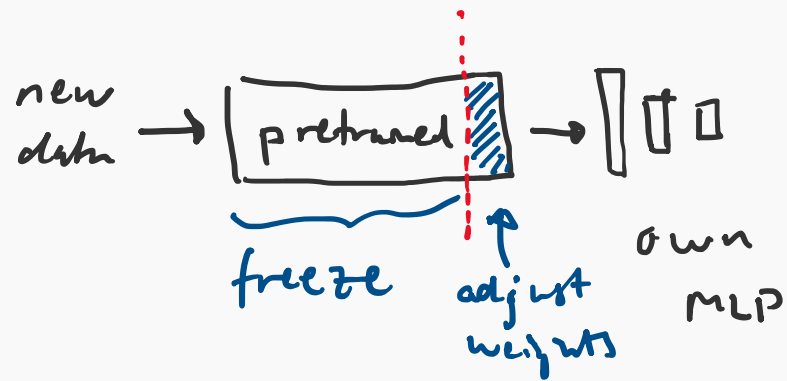
idea is that the

earlier layers learn

fundamental structures while deeper layers specializes to learn specific structures that are important for our dataset