**PyTorch for Deep Learning Bootcamp**

**8. Transfer Learning**

**What is transfer learning?**

Surely someone has spent the time crafting the right model for the job…

**Example transfer learning use cases**

**Computer vision**: Imagenet, dataset of images → take a Neural Network trained on Imagenet → customize the NN for our own dataset → use for our own use case (food classification)

**Natural language processing**: Language processing → take a Neural Network trained on natural language patterns (like Wikipedia) → customize the NN for our own dataset → use for our own use case (spam filter)

So for transfer learning we takewhat one model has learned on one dataset (usually big datasets), a **pre-trained model**,it learns patterns/weights/parameters, and we adjust it for our own use cases.

**Why use transfer learning?**

- Can leverage an existing neural nwtwork architecture **proven to work** on problems similar to our own

- Can leverage a working network architecture which has **already learned patterns** on similar data to our own (often ends in great results with less data)

The process of these models ends up being:

Learn patterns in a variety of images (using ImageNet) → Pretrained EfficientNet architecture (already works really well on computer vision tasks) → Extract/tune patterns/weights to suit our own problem (FoodVision Mini) → Model performs better than from scratch

Most practitioners, however, rarely, if ever perform pre-training on today’s largest datasets but instead use some of the many publicly available parameters sets (pre-trained models).

Even if the downstream data of interest (our custom data) appears to only be weakly related to the data used for pre-training, transfer learning remains the best available option.

The thing that really make a difference (transfer learning), if we can do better at transfer learning, it’s this world changing thing. **Suddenly lots more people can do world-class work with less resources and less data.**

**Improving a model**

There are some ways to improve a model: more data, data augmentation, better data or use **transfer learning**.

**Where to find pretrained models**

- PyTorch domains libraries (torchvision, torchtext, torchaudio, torchrec)

- Torch image models (*timm* library)

- HuggingFace Hub

- Paperswithcode SOTA (State-Of-The-Art, the best performing type of research)

**What we’re going to cover**

- Getting setup (**importing previously written code**)

**-** Introduce **transfer learning** with PyTorch

- **Customise a pretrained model** for our own use case (FoodVision Mini – pizza, steak, sushi)

- **Evaluating** a transfer learning model

- **Making predictions** on our own custom data