Physalia Courses

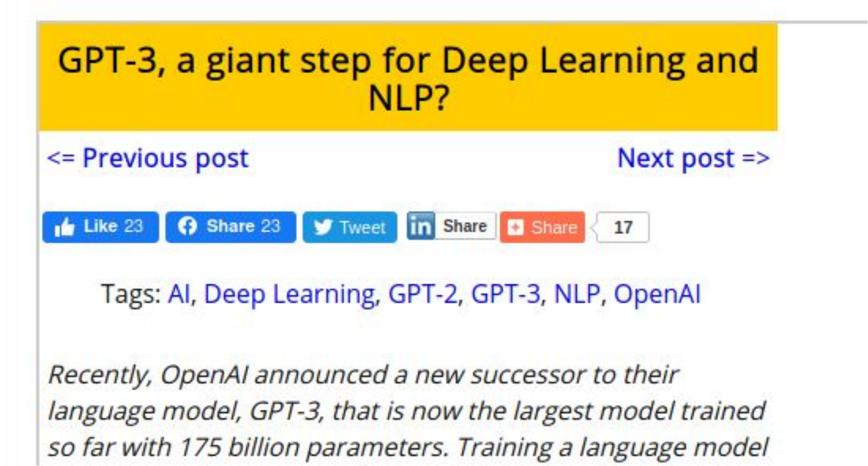
Transfer learning

Let other people do the legwork

Filippo Biscarini
Senior Scientist
CNR, Milan (Italy)

Nelson Nazzicari
Research fellow
CREA, Lodi (Italy)

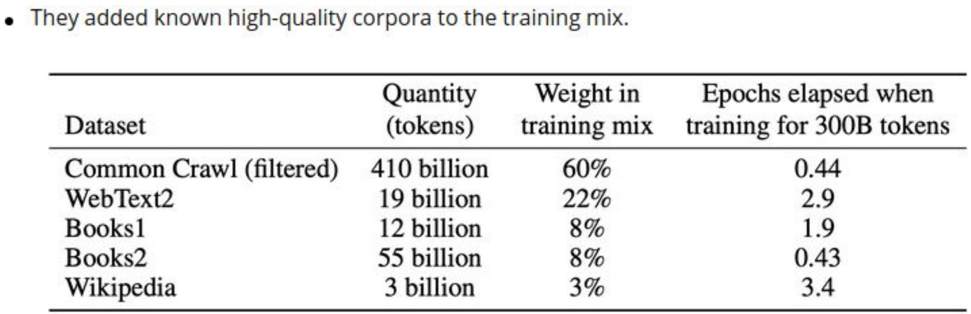
Deep Learning headlines are intimidating



GPT-3, a giant step for Deep Learning and NLP?

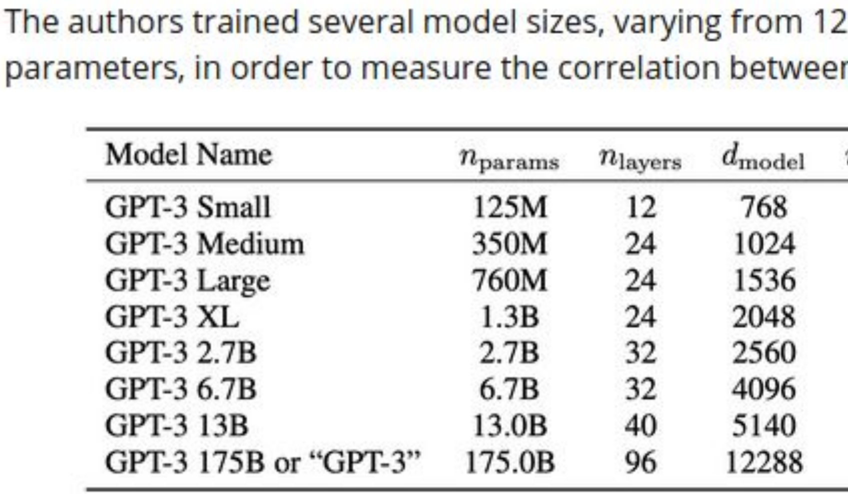
Tags: AI, Deep Learning, GPT-2, GPT-3, NLP, OpenAI

Recently, OpenAI announced a new successor to their language model, GPT-3, that is now the largest model trained so far with 175 billion parameters. Training a language model



- They added known high-quality corpora to the training mix.

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4



The authors trained several model sizes, varying from 12 parameters, in order to measure the correlation betwee

Model Name	n_{params}	n_{layers}	d_{model}
GPT-3 Small	125M	12	768
GPT-3 Medium	350M	24	1024
GPT-3 Large	760M	24	1536
GPT-3 XL	1.3B	24	2048
GPT-3 2.7B	2.7B	32	2560
GPT-3 6.7B	6.7B	32	4096
GPT-3 13B	13.0B	40	5140
GPT-3 175B or "GPT-3"	175.0B	96	12288

https://www.kdnuggets.com/2020/06/gpt-3-deep-learning-nlp.html

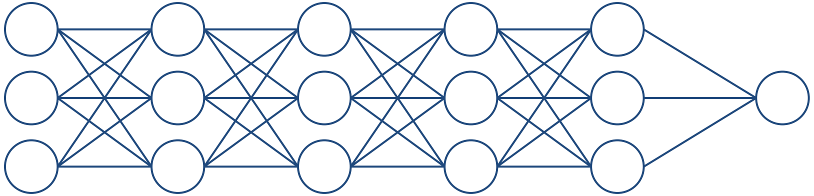
What is transfer learning?

“In transfer learning, we first train a base network on a base dataset and task, and then we repurpose the learned features, or *transfer* them, to a second target network to be trained on a target dataset and task.”

- How transferable are features in deep neural networks?, Yosinski et al., 2014

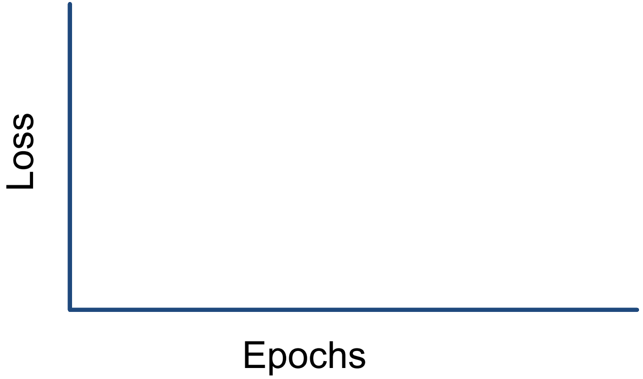
Transfer learning workflow

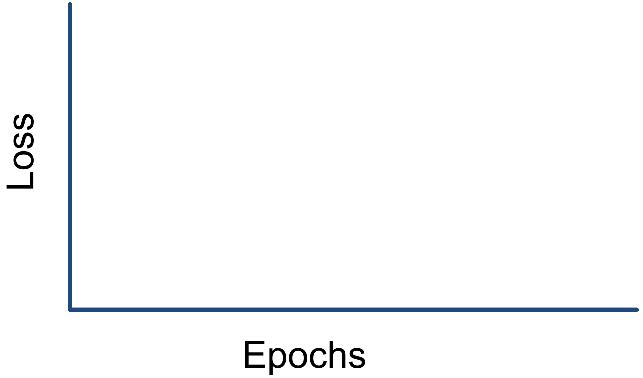
1. Get architecture
2. Get parameters
3. Remove last layer
4. freeze
5. A little wrapper
6. Train
7. <optional> Fine tune



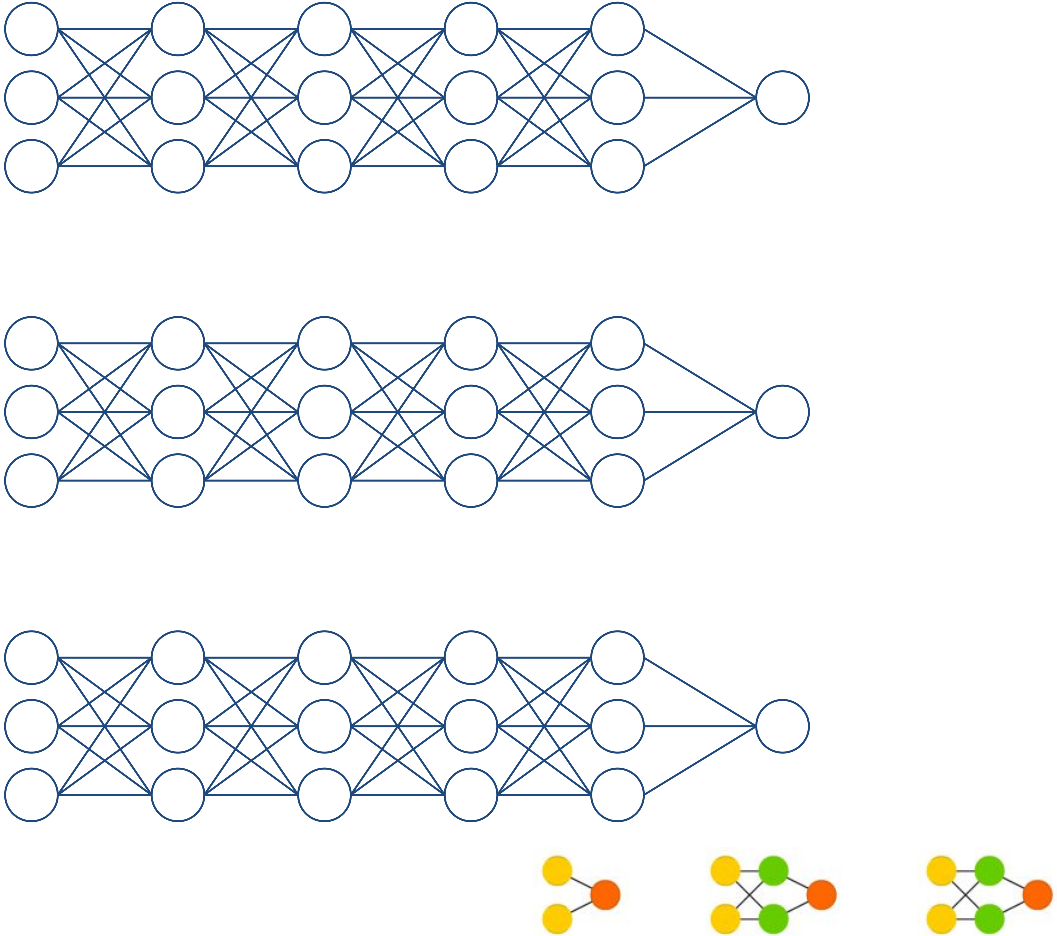
What is fine tuning IN TRANSFER LEARNING

1. <once you have trained the top layers>
2. Unfreeze everything
3. Select a very small learning rate
4. Do some training epochs



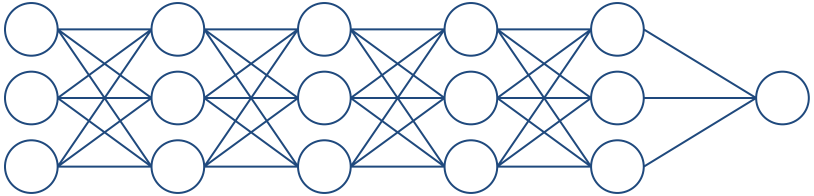


You can train more!

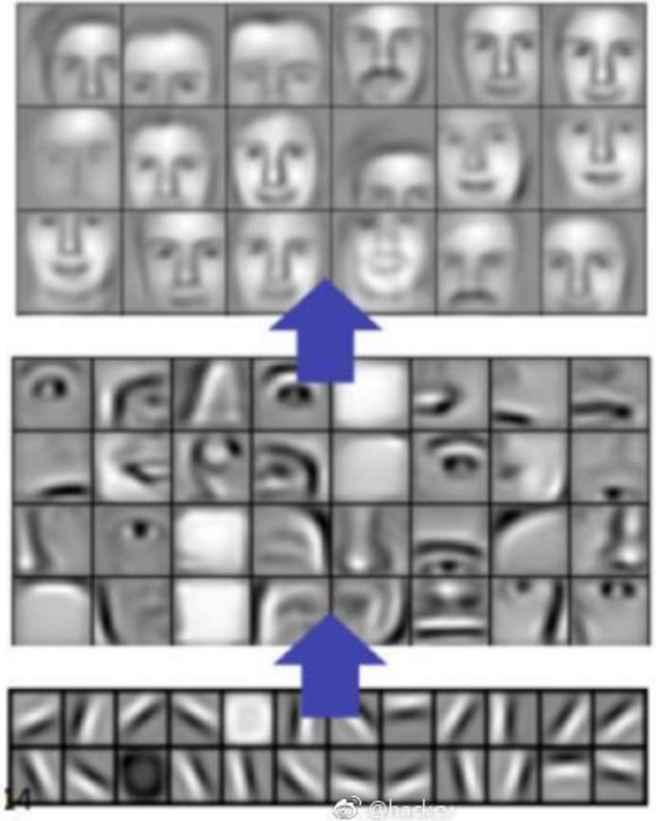


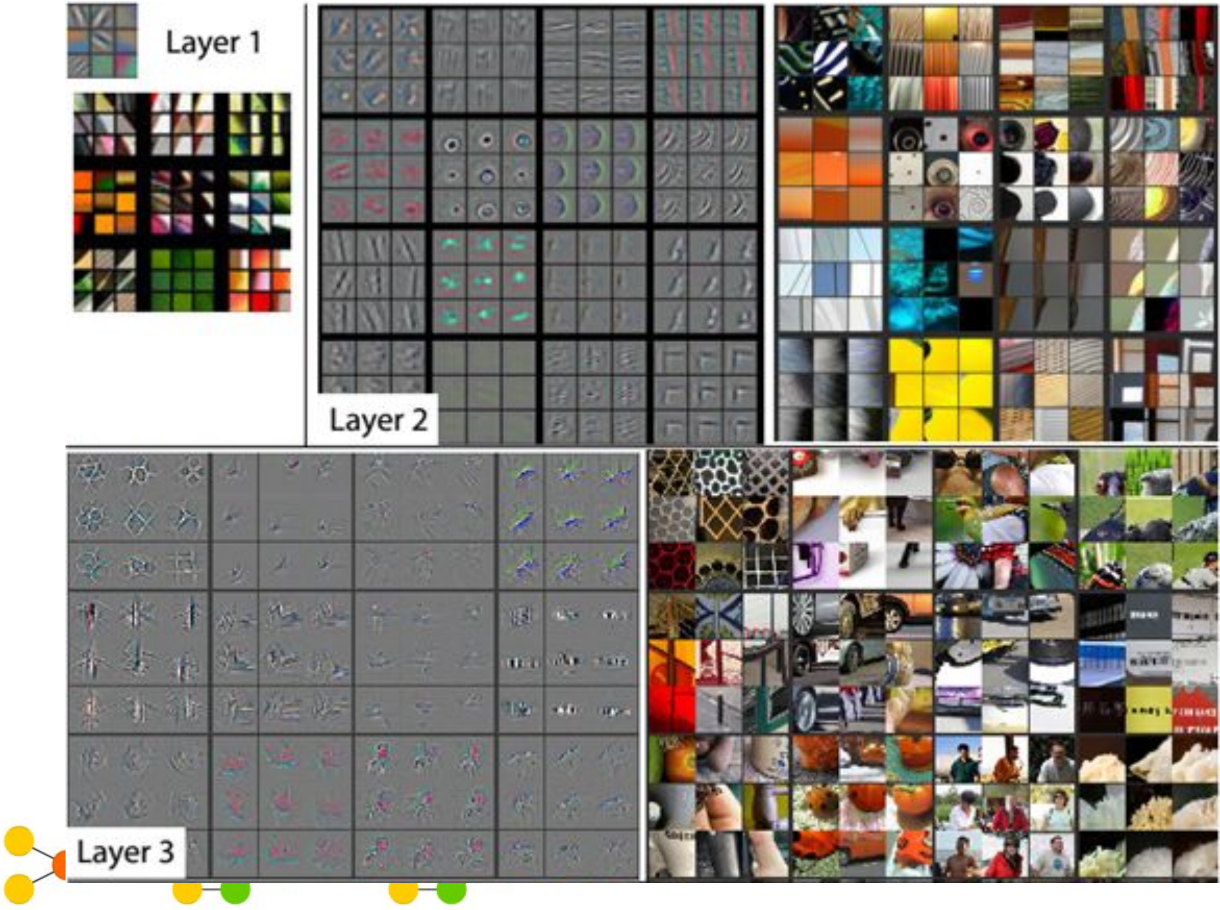
A different approach: feature extraction

1. Get architecture
2. Get parameters
3. Remove last layer
4. Pass all your data through the network
5. Throw away the network
6. Do a small network

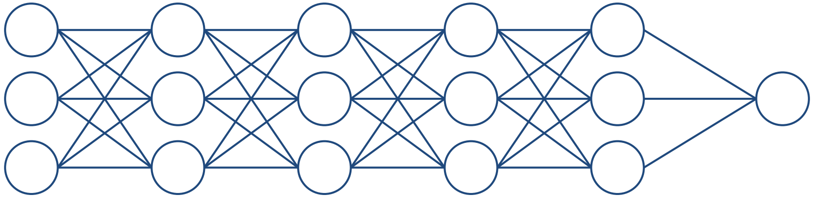


Why does it work?





Why does it work?



In keras: available architectures (“applications”) https://keras.io/api/applications/

In keras

```
from keras import models, layers
from keras.applications import ResNet50

#downloading the net and its weights trained on imagenet dataset
my_resnet = ResNet50(weights='imagenet', include_top=False, input_shape=(...))

#it's already the default value, but we set it anyway to
#make clear we are not going to train the whole thing
my_resnet.trainable = False

#build the model
model = models.Sequential()
model.add(my_resnet)
model.add(layers.Flatten())
model.add(layers.Dense(units=5, activation='softmax'))
```

[REF]

- How transferable are features in deep neural networks? https://arxiv.org/abs/1411.1792
- Language Models are Few-Shot Learners (GPT-3) https://arxiv.org/abs/2005.14165
- Deep Residual Learning for Image Recognition, He et al., 2015, https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html
- Rethinking the Inception Architecture for Computer Vision, Szegedy et al., 2016 https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Tan & Le, 2019, https://arxiv.org/abs/1905.11946