Transfer learning

Thursday, 27 May 2021 6:52 PM

Code → transfer-learning

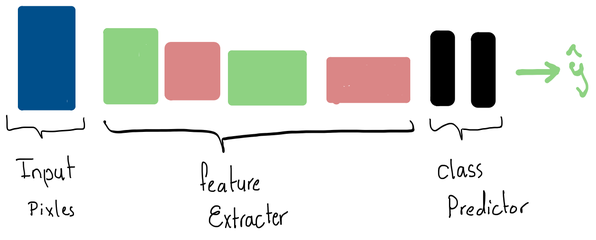
I. Intro

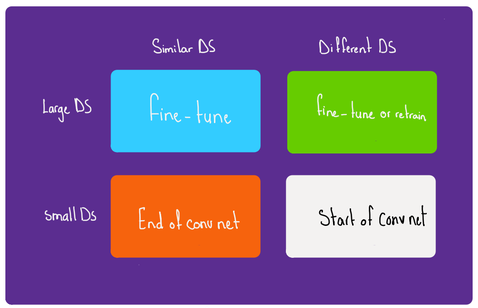
Use pretrained models and apply that to a new task with minor changes to the network

this will depend on how similar new task data to what a pretrained model has seen.

II. fine tuning

The conv layers are extracting features but the final fully connected layer is specific to the task and so modifying the last fully connected layer will allow the pretrained model to work on a new task.





* Approaches *

1. Small data set, similar data

Overfitting is a concern with small dataset

End of Convnet :

- Slice of the end of the NN
- Add new fully connected NN with matching input and output
- Freeze the conv layers in pretrained model
- Randomize weights in new NN
- Train the network

2. Small data set, different data

Start of Convnet:

- Slice of the pretrained layer near the beginning
- Add new fully connected layer
- Freeze weights of conv layer
- Randomize weights in new NN
- Train the network

Overfitting is not a concern with large datasets

3. Large dataset, similar data

fine tune:

- Slice the end of the NN
- Add a new fully connected layer
- Randomize the weights in NN
- Initialize the conv layer with its own weights
- Retrain the network

4. Large dataset, different data

fine tune or retrain:

- Slice the end of the NN
- Add a new fully connected layer
- Retrain the network from scratch with random weights
- Or just use fine tune approach instead