Deep Learning for Computer Vision

Finetuning in CNNs

Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Limitations of Working with CNNs

Practical concerns of working with CNNs:

- Optimization of parameters in deep models (characteristic of CNNs) very hard, requires careful parameter initializations and hyperparameter tuning

Limitations of Working with CNNs

Practical concerns of working with CNNs:

- Optimization of parameters in deep models (characteristic of CNNs) very hard, requires careful parameter initializations and hyperparameter tuning
- Can suffer from overfitting, as data samples used for training are lesser compared to parameters being trained

Limitations of Working with CNNs

Practical concerns of working with CNNs:

- Optimization of parameters in deep models (characteristic of CNNs) very hard, requires careful parameter initializations and hyperparameter tuning
- Can suffer from overfitting, as data samples used for training are lesser compared to parameters being trained
- Require a long time and computational power to train

Model	Parameters
AlexNet	60m
VGG	138m
Inception v1	5m
Inception v3	23m
Resnet 50	25m

Model	Training time	Hardware
AlexNet	5-6 days	two GTX 580 3GB
VGG	2-3 weeks	four NVIDIA Titan Black
Inception v1	1 week	not mentioned

Strategies Used

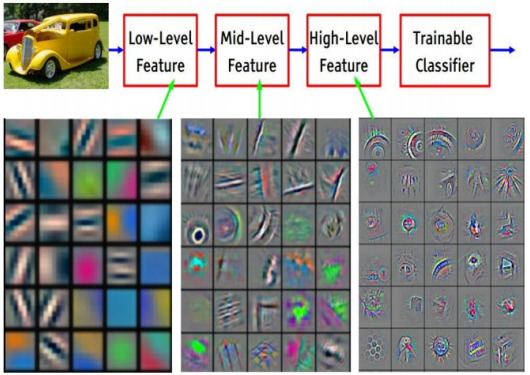
- Better weight initialization:
  - **Glorot/He initialization:** Empirically shown to give good results
  - **Hand-designed:** Using domain knowledge, come up with features like edges (with certain orientations), shapes etc.
  - **Locally trained using unsupervised learning approaches:** Use unsupervised greedy layerwise pretraining to get features one layer at a time starting from the initial layer. Rarely used nowadays due to increased computational power and dataset sizes

Strategies Used

- Better weight initialization:
  - **Glorot/He initialization:** Empirically shown to give good results
  - **Hand-designed:** Using domain knowledge, come up with features like edges (with certain orientations), shapes etc.
  - **Locally trained using unsupervised learning approaches:** Use unsupervised greedy layerwise pretraining to get features one layer at a time starting from the initial layer. Rarely used nowadays due to increased computational power and dataset sizes
- Regularization methods:
  - L2-weight decay, L1-weight decay
  - DropOut, BatchNorm, Input/Gradient Noise
  - Data augmentation

Alleviates overfitting, does not train faster though!

Interesting Property of CNNs

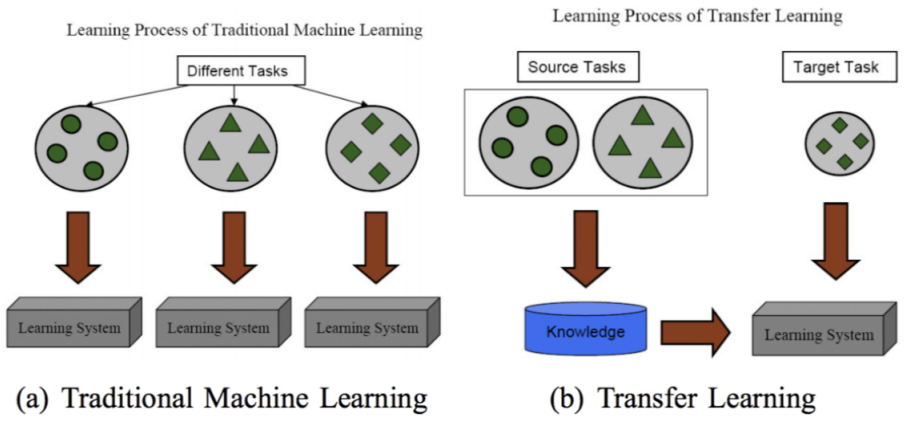


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

- Features learned by CNN layers are hierarchical
- Initial layers learn simple/generic features like edges, colour blobs, etc - remain constant across various models trained on different datasets
- Later layers perceive more abstract/specialized features and are generally dataset-specific
- What can we do with this?

Credit: CS231N, Stanford Univ

Transfer Learning



Credit: A Survey on Transfer Learning, Pan and Yang 2010

Transfer Learning

- Using knowledge learned over a different task(s) (having sufficient data) to aid the training of current task

Transfer Learning

- Using knowledge learned over a different task(s) (having sufficient data) to aid the training of current task
- Since pretrained models with good results are readily available, they can reduce the time spent on training, hyperparameter tuning and thus need for high-end computing hardware
- Pretrained weights of CNN model can be used as:
  - Only parameters of classification layers are trained; rest of the network is frozen
  - Pretrained weights serve as initialization, and the entire network (or few layers at the end) are further finetuned to better model target task

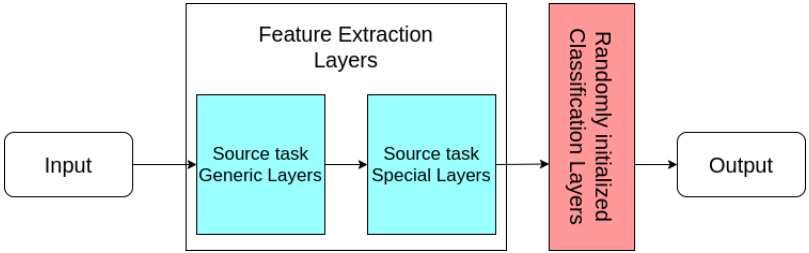
Transfer Learning

- Using knowledge learned over a different task(s) (having sufficient data) to aid the training of current task
- Since pretrained models with good results are readily available, they can reduce the time spent on training, hyperparameter tuning and thus need for high-end computing hardware
- Pretrained weights of CNN model can be used as:
  - Only parameters of classification layers are trained; rest of the network is frozen
  - Pretrained weights serve as initialization, and the entire network (or few layers at the end) are further finetuned to better model target task
- Choice depends on variables such as dataset size and similarity between target and source datasets

Which mode to select?

Dataset is small; target and source datasets are similar:

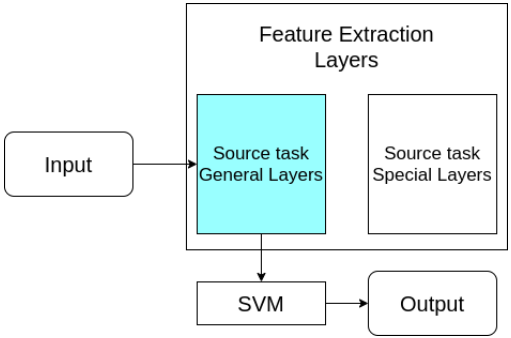
- Specialized features likely remain same for source and target datasets
- Parameters of classification layer are randomly intialized and trained, while rest of network remains frozen (to prevent overfitting)



Which mode to select?

Dataset is small; target and source datasets are dissimilar:

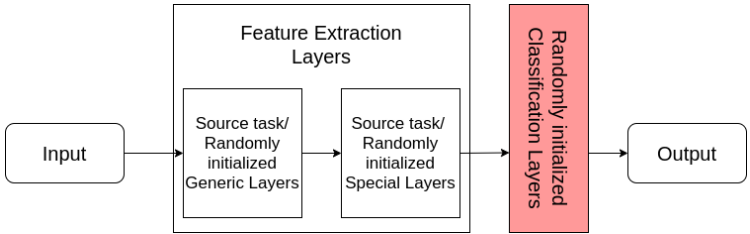
- Specialized features are different but generic features can be shared
- An intermediate layer with appropriate specialization level is chosen and linear classifiers like SVMs are trained over those features.



Which mode to select?

Dataset is large:

- We can use pretrained network as a good initialization which is finetuned on target dataset
- While finetuning, learning rate is kept low in order to not change pretrained parameters too much
- If dataset is very different, it can either be trained from scratch or techniques like transitive transfer learning[1] or its successors can be applied



[1] Tan et al, Transitive Transfer Learning, KDD 2015

Homework

Readings

- Chapter 9 (§9.8-9.9), DL Book
- Lecture on Transfer Learning, CS231n course, Stanford Univ
- (Optional) How transferable are features in deep neural networks?