Transfer Learning in CNNs With Medical Images

Dhruvin Patel

University of Mississippi

drpatel@go.olemiss.edu

April 17, 2019

Overview

Overview

What are CNNs?

- CNNs stands for Convolutional Neural Networks
- They are a category of Neural Networks
- Shown to be very effective in areas of image recognition and classification
- Comprised of four main operations
  - Convolution
  - Non Linearity (ReLu)
  - Pooling
  - Classification

What are CNNs?

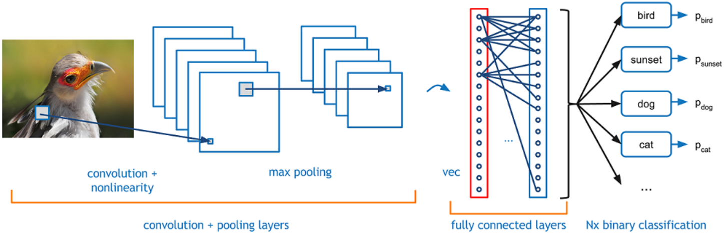


Figure: Example of a CNN

Overview

What is Transfer Learning?

Idea:

- Transferring knowledge from one domain to another related domain
- The idea comes from humans
    - For example, if two people want to learn how to code in Python
    - One person has some coding experience and the other does not
    - Person with experience will learn faster

What is Transfer Learning?

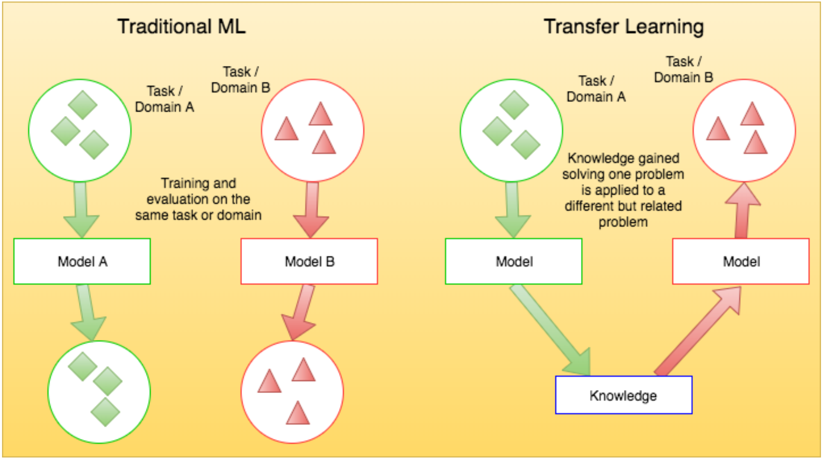


Figure: Traditional ML vs Transfer Learning

What is Transfer Learning?

Instead of training a network from scratch:

- Take a network trained on a different domain for a different source task
- Adapt it for the target domain and the target task

Overview

How can we transfer knowledge in CNNs?

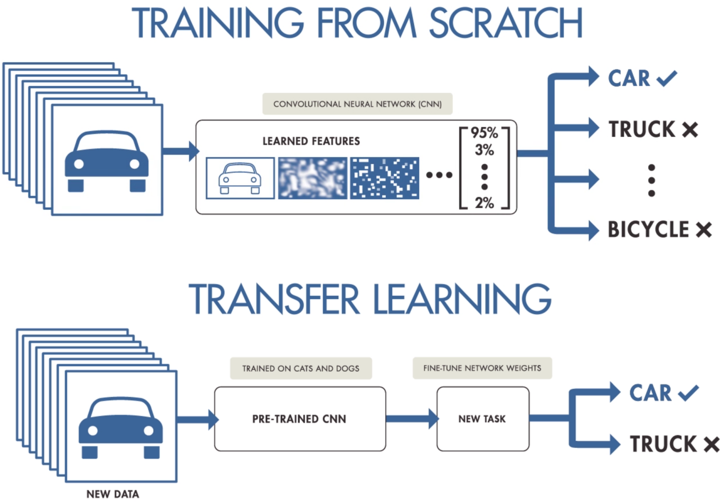


Figure: Training from Scratch vs Transfer Learning

Overview

Project Goals

- Is it possible to transfer knowledge from a generalized unrelated model to specific model?
- With classifying medical images?
- What are the benefits? What are the negatives?
- What layers in the model matter more when fine-tuning the model?

Overview

Datasets

Two Medical Image Datasets

1. HAM10000 MINST dataset
    - Consist of 10,015 dermatoscopic images of common pigmented skin lesions
    - With 7 classes
    - Image size (600,450) rescaled to (224,224)
2. Chest X-Ray dataset
    - Consist of 5,856 chest X-Ray images
    - 2 classes: Normal or Pneumonia
    - Image size (2090, 1858) rescaled to (224,224)

HAM10000 Dataset

7 skin diagnostic classes

1. Actinic keratoses and intraepithelial carcinoma / Bowens disease (akiec)
2. basal cell carcinoma (bcc)
3. benign keratosis-like lesions (bkl)
4. dermatofibroma (df)
5. melanoma (mel)
6. melanocytic nevi (nv)
7. vascular lesions (vasc)

HAM10000 Dataset

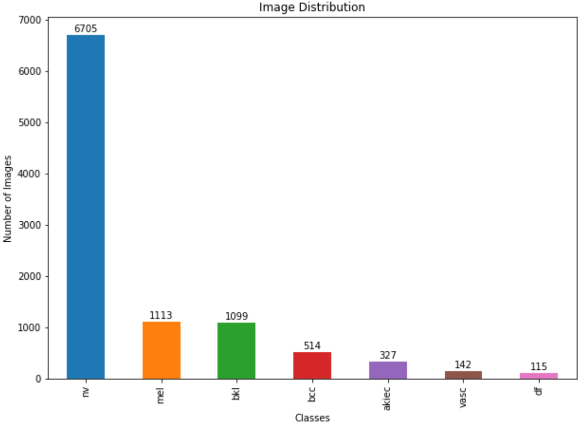


Figure: HAM10000 Image Distribution

X-Ray Dataset

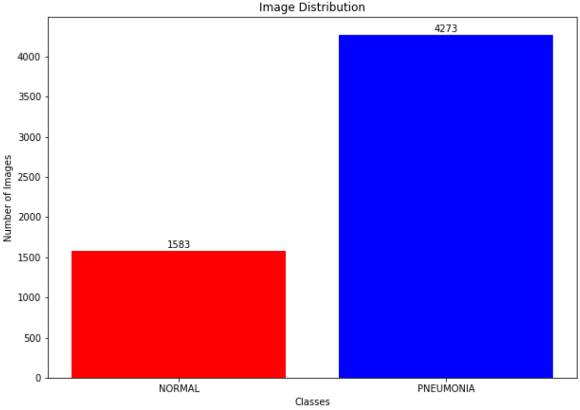


Figure: X-Ray Image Distribution

Overview

Experiment Setup

Baseline Models

- Decision Tree
- Random Forest
- CNN with random weights

Transferred Models

- CNN with pre-trained weights
- CNN with different layers frozen

Experiment Setup

Decision Tree and Random Forest

- 80/20 split
- implement using ski-learn
- features for the classifier were the pixel values

Experiment Setup

CNN

- 60/20/20 split
- Pixel values scaled from -1 to 1
- Early stop
- Pre-trained weights: trained on ImageNet
- VGG16 model

ImageNet

- Large visual database with 14 million images
- Contain more than 20,000 categories
- Categories like dog, cat, strawberry, etc.

Experiment Setup

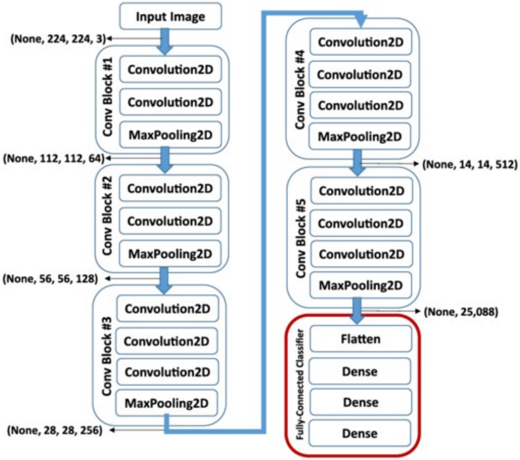


Figure: VGG16 Model

Experiment Setup

Classification layer changed to:

- 1 flatten layer
- 2 dense layers
- 2 dropout layer between the dense layers
- 1 last dense layer

Experiment Setup

Development Enviroment

- Windows 10
- GPU: GTX 1080 (supports Cuda)
- Cuda 9.0
- Python 3.6
- Keras 2.2.4
- Numpy 1.15.4
- Mathplotlib 3.0.2

Overview

Results

	Baseline Models			CNN Models using ImageNet Weights			
	Decision Tree	Random Forest	CNN Random Weights	All Layers Trainable	Last Block Trainable	First Block Trainable	First and Last Blocks Trainable
akiec	.09	.25	.31	.36	.27	.00	.36
bcc	.20	.38	.41	.64	.59	.00	.64
bkl	.26	.40	.37	.64	.58	.00	.60
df	.00	.00	.00	**.27**	.00	.00	**.07**
mel	.19	.22	.34	.58	.54	.01	.53
nv	.76	.84	.88	.92	.91	.80	.91
vasc	.06	.00	.17	.78	.50	.00	.50
micro avg	.57	.70	.72	.81	**.79**	.66	**.79**
macro avg	.23	.30	**.35**	**.60**	**.48**	**.11**	**.52**

Table: F1-scores of baseline models and CNNs models on HAM10000.

Results

- CNN with random weights had to best accuracy from the baseline models.
- Best accuracy using ImageNet weights
- Last block has higher accuracy than beginning layers

Results

	Baseline Models			Transferred Models	
	Decision Tree	Random Forest	CNN Random Weights	CNN ImageNet Weights	CNN HAM10000 Weights
Normal	.75	.88	.89	**.91**	.90
Pneumonia	.91	.96	.96	**.97**	.96
micro avg	.87	.94	.94	**.95**	.94
macro avg	.83	.92	**.92**	**.94**	**.93**

Table: F1-scores of baseline models and CNNs models on X-Ray dataset.

Results

- Using ImageNet weights got the best accuracy
- CNN models also outperformed the baseline decision tree and random forest
- Outperform random weights but not better than using ImageNet weights.

Results

Random vs ImageNet Weights

- A **higher slope** when training
- A **higher accuracy**
- finishes training in **few number of epochs**.

Results

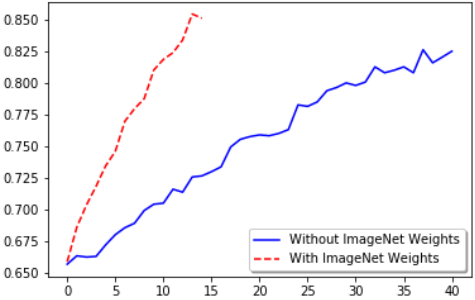


Figure: Line graph of training accuracies of two models: Using ImageNet Weights and Without ImageNet Weights

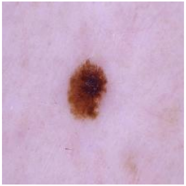
Overview

Discussion

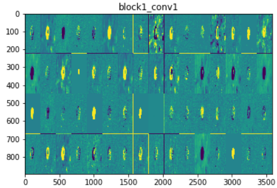
Positives

- The target task can use pre-trained weights to learn features
- A significant boost in performance compared to random initialization
- ImageNet weights are a favorable starting parameters

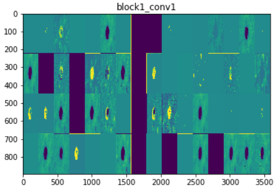
Discussion



(a) Original Image



(b) Activation with ImageNet



(c) Activation with random weights

Figure: Visualize of intermediate activation from first convolutional layer

Discussion

Negatives

- Using VGG16 model
- Smaller datasets may have varying results
- Image size for model at (224,224)

Overview

Conclusion

- A target task can benefit from a unrelated source task
- ImageNet weights as a feature extractor
- Impressive performance compared to random weights
- Later layers matter more than beginning layers

Questions?