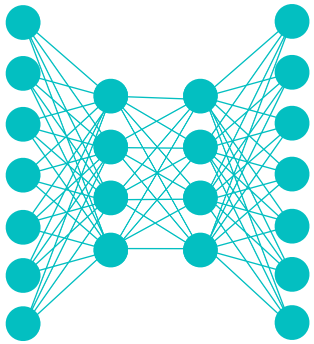
Lecture Notes for

Neural Networks and Machine Learning

Transfer Learning
Deep Transfer Learning



Logistics and Agenda

- Logistics
  - None
- Agenda
  - Transfer Learning Overview
  - Transfer Learning in Deep Learning
  - Demo
- Next Time:
  - Paper Presentation (for three lectures in a row!)
  - Adaptive Transfer Learning
  - Consistency Loss

		Total Assigned	Responded?	Self Consistency	Attentive Statistics	DeepTox	Data Augmentation	Group Normalization	Perceptual Losses	Radford GANs	AlphaFold
~~Barajas, Cynthia~~	~~cs5Sect 4~~	~~0~~	~~1~~				x		x		
Canter, Austin	cs5Sect 4	1	1						1	0	
Collins, Joel	cs5Sect 4	1	1	0		0					1
Du, Jinyu	cs5Sect D	1	1				1			0	
Ebrahimian, Jonathan	cs5Sect P	1	1							1	
Emami, Hessam	cs5Sect 4	1	1				1			0	
Gage, Nathan	cs5Sect D	1	1								
Gao, Qing	cs5Sect 4	1	1				1			0	
Geng, Zicheng	cs5Sect D	1	1				1			0	
Havard, Andrew	cs5Sect P	1	1	0	0					1	
Hu, Yvon	cs5Sect D	1	1	0		1	0				
Klinkert, Jake	cs5Sect D	1	1	0				1		0	
Larsen, Nicholas	cs5Sect 4	1	1	0	1						
Larsen, Steven	cs5Sect 4	1	1	0	1						
~~Lazaro, Irvin~~	~~cs5Sect 4~~	~~0~~	~~1~~	x			x		x		
Lu, Yifan	cs5Sect 4	1	1	0			0	1			
ge, Ayesh Medushanka	cs5Sect D	1	1			0					1
McNitt, Troy	cs5Sect P	1	1								
Moros, Jonas	cs5Sect D	1	1								
Rajapandian, Khoushik	cs5Sect 4	1	1							1	0
Rosenblatt, Jack	cs5Sect D	1	1			0	0				1
Srirama, Nathan	cs5Sect D	1	1	1			0			0	
Tsai, Amor	cs5Sect D	1	1	0		1	0				
Wall, Nick	8321 Sect D	0	0								
Wang, Kuo	cs5Sect D	1	1	0	1	0					
Yang, Chenyu	cs5Sect 4	1	1	1			0				
Yassien, Sam	8321 Sect D	1	1						1		
Yu, Hongjin	cs5Sect D	1	1	1					0	0	
~~Zepeda, Juan~~	~~cs5Sect 4~~	~~0~~	~~1~~	x			x	x			
			Num Assigned	3	3	2	4	2	2	3	3

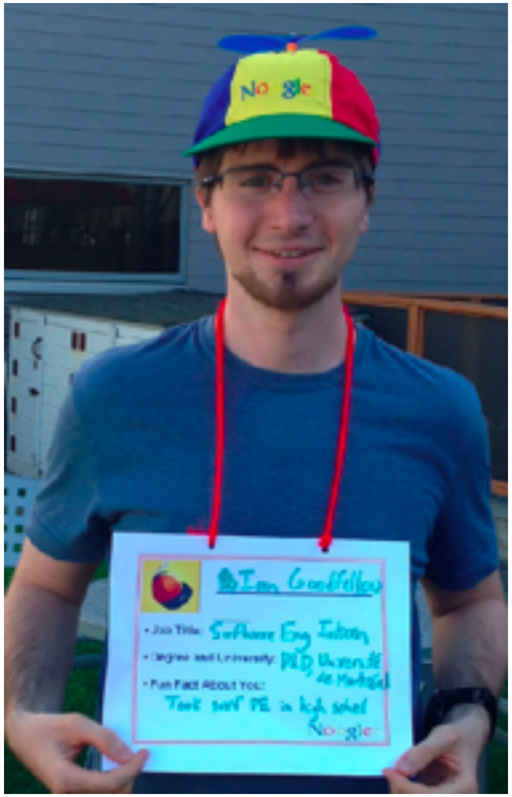
Transfer Learning Overview

Transfer Learning

- Transfer knowledge from a source prediction task to a target prediction task
  - without any regard for performing well on source task
- **Original**: Neural Information Processing 1995 (NeuRiPs)
  - Workshop on "Learning to Learn"
    - How to effectively retain and reuse previously learned knowledge
  - Originally used in Markov chain and Bayesian networks (keeping n-grams, *etc*.)
- **Key idea**: Humans can generalize what they learn to almost any domain, can we mimic this behavior with ML?

Ian Goodfellow’s Definition:

“Transfer learning refers to any situation where what has been learned in one setting is exploited to improve generalization in another setting.”



Transfer Learning: Large Umbrella

- Appears under many associations in the literature:
  - Learning to learn / Life-long learning
  - Knowledge transfer / Inductive transfer
  - Multi-task learning
  - Knowledge consolidation
  - Context-sensitive learning
  - Knowledge-based inductive bias
  - Meta learning
  - Incremental learning
  - Cumulative learning
  - Domain adaptation

Precise Definition of Transfer Learning

$$X = x_1, x_2, \ldots x_N \in \mathscr{X}$$

$$Y = y_1, y_2, \ldots y_N \in \mathscr{Y}$$

$$\mathscr{D} = \{\mathscr{X}, p(X)\}$$

Domain | **Feature Space** | **Probability Observation**

$$\mathscr{T} = \{\mathscr{Y}, p(Y|X)\}$$

Task | **Label Space** | **Learned Probability**

- $\mathscr{D}$ Domain defines the features used and probability
- $\mathscr{X}$ is the space of all possible features
- $p(X)$ is probability of observing specific instances in $\mathscr{X}$
    - Typically **intractable** to calculate (generative)

- $\mathscr{T}$ Task is within a domain, defining labels and model
- $\mathscr{Y}$ is space of all possible labels
- $p(Y|X)$ probability of observing specific label given the specific feature:
    - **Not** intractable (discriminative)

Definition with Examples

$$X = x_1, x_2, \ldots x_N \in \mathscr{X}$$

$$\mathscr{D} = \{\mathscr{X}, p(X)\}$$

Domain **Feature Space** **Probability Observation**

- Image Pixels
- Sensor Readings
- Natural Language
- *Almost anything that we can represent as a feature*

$$Y = y_1, y_2, \ldots y_N \in \mathscr{Y}$$

$$\mathscr{T} = \{\mathscr{Y}, p(Y|X)\}$$

Task **Label Space** **Learned Probability**

- Object Classification
- Dolphin/Shark Classification
- Sentiment Analysis
- *Any labeled task for which we might be able to build a classifier*

Transfer Learning

$$X = x_1, x_2, \ldots x_N \in \mathcal{X} \qquad Y = y_1, y_2, \ldots y_N \in \mathcal{Y}$$

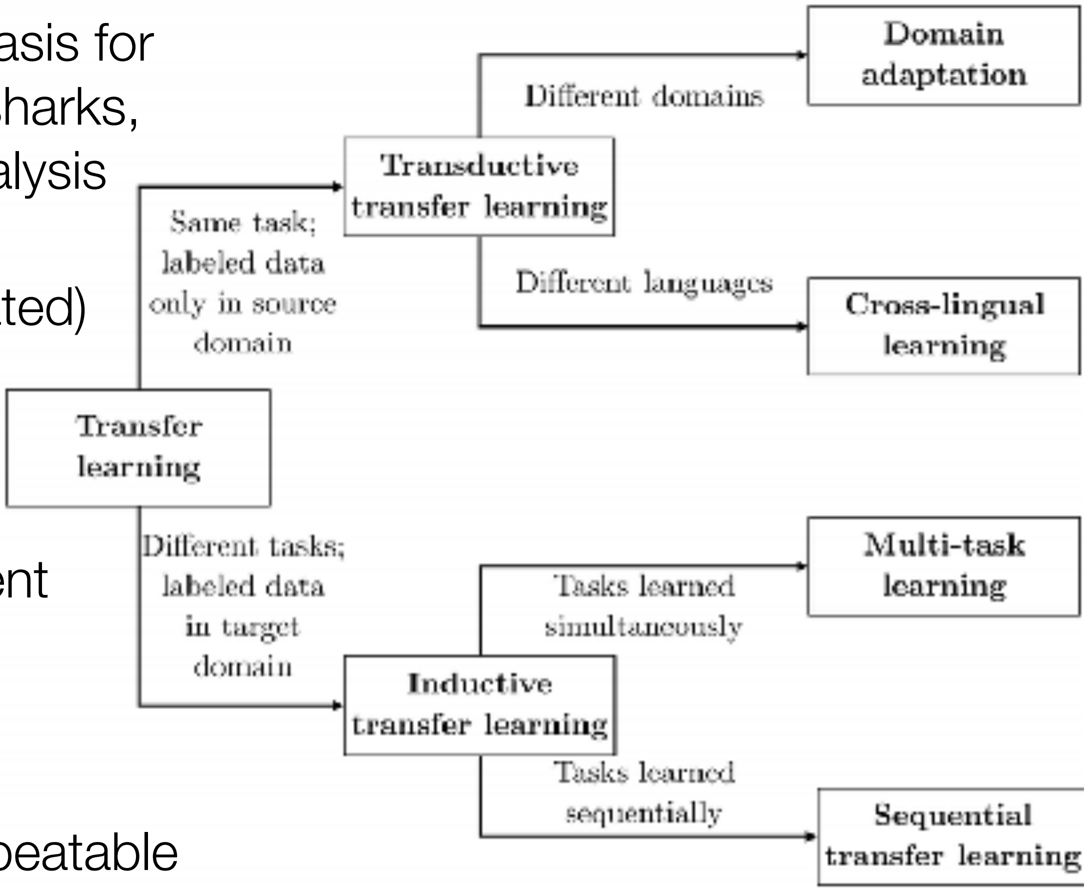
$$\mathcal{D} = \{\mathcal{X}, p(X)\} \qquad \mathcal{T} = \{\mathcal{Y}, p(Y|X)\}$$

Domain **Feature Space** **Probability Observation** **Task** **Label Space** **Learned Probability**

- Need to translate document **Source** to **Target** $\mathcal{T}_S \rightarrow \mathcal{T}_T$
- Variety of differences might be present. For example, in the context of document classification:
    - **Feature space**: different languages $\mathcal{X}_S \neq \mathcal{X}_T$
    - **Marginals**: same language, same label space, but differing topics $p(X_S) \neq p(X_T)$
    - **Conditional**: different label distributions or possibly different labels $p(Y_s | X_S) \neq p(Y_T | X_T)$

Categories of Transfer Learning

- **Inductive**: Same Domain, Different Task
  - Using pre-trained VGG as basis for classifying dolphins versus sharks, Style Transfer, sentiment analysis from Glove
- **Transductive**: Different (but related) Domains, Same Task
  - Place identification from RGB Images or LIDAR
- **Unsupervised Transfer**: Different Domains, Different Tasks
  - Learning to paint art and learning to be a surgeon
  - Not yet a field with much repeatable traction



Other categorizations

	Training	Testing
Transfer Learning	Task 1	Task 2
Multi-task Learning	Task 1 ··· Task N	Task 1 ··· Task N
Lifelong Learning	Task 1 ··· Task N	Task N+1

Lifelong Learning is a Grand AI Challenge: Humans can learn to ride a bike and use that to understand better about driving a car. Machine Learning in its current form is far from this capability. How can we move our siloed version of artificial intelligence closer to the process of human based learning? How can we accumulate knowledge from model to model?

Does biology of human learning hold any clues to success? How does a human learn to crawl? To talk? To ride a bike? What is a human's motivation to learn?

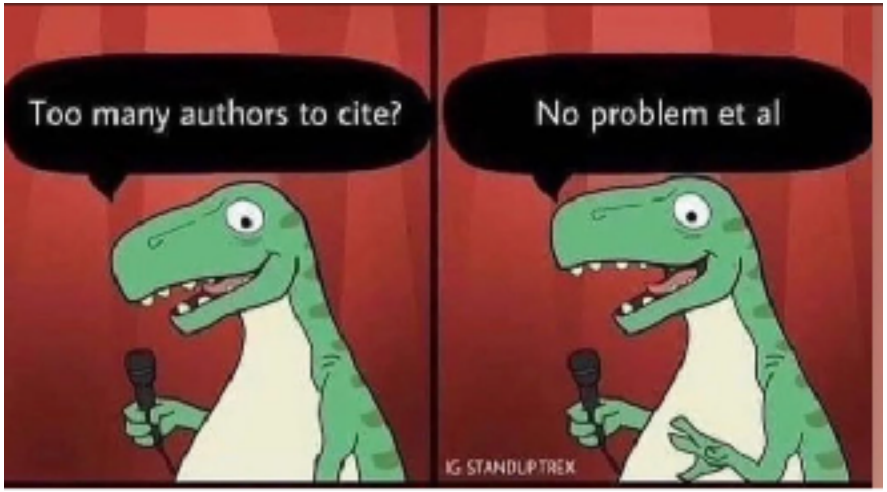
Transfer Learning with Neural Networks

Found in a recent paper:

6 Unrelated Work

This paper is not related to [8, 23, 48, 13, 35] in any way, but we think everyone should read these papers because: (1) they're real good, (2) my friends also need those citations.

7 Related Work



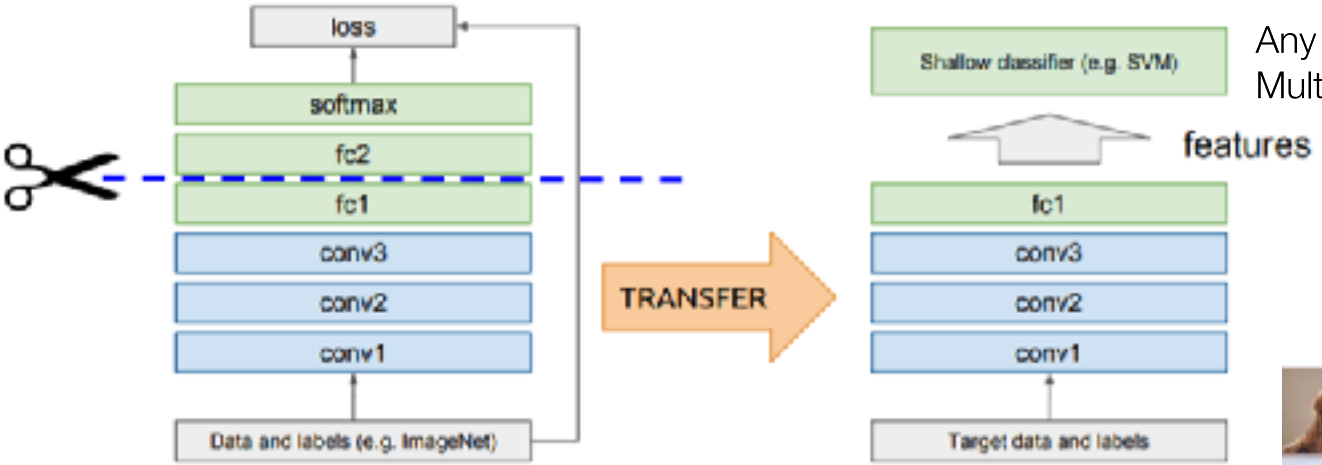
Deep Transfer Learning

- Almost always **Inductive Transfer**
  - (new task , same domain, or domain adaptation)
- Almost always **Feature Representation Transfer**
  - like image pre-training
- All other topics are mostly open research topics that maybe one of you will solve!

(Sun, B., Feng, J., & Saenko, K. (2016). Return of Frustratingly Easy Domain Adaptation)

Approaches with Deep Learning

- **Feature Extraction Transfer**
  - Most well known: use learned parameters from one task in another task in same domain
  - Most useful when labels for target domain are sparse



Any classifier: SVM, Random Forest, Multi-layer perceptron

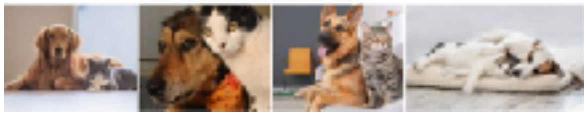


Image Net

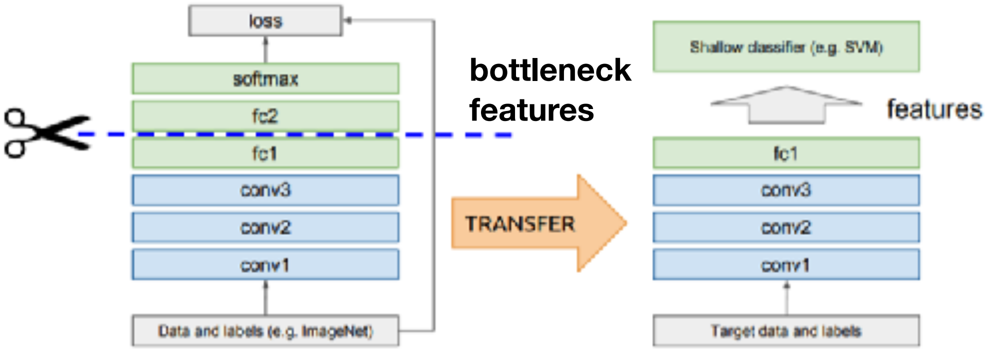
New domain: Dogs versus Cats

New domain: Gaze Classification

https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

Defining the Bottleneck

- Frozen training layers before bottleneck:
  - Why waste computations?
  - Computing more than one forward pass on the same data—just save them out
  - *Unless* using augmentation
- In Keras, build multiple models with different entry points
  - **Input** to **Bottleneck**
  - **Bottleneck** to **Output**
  - **Input** to **Output**




```
inputs = Input(shape=(IMG_SIZE, IMG_SIZE, 3))
model_base = VGG(include_top=False,
            input_tensor=inputs, weights="imagenet")
```

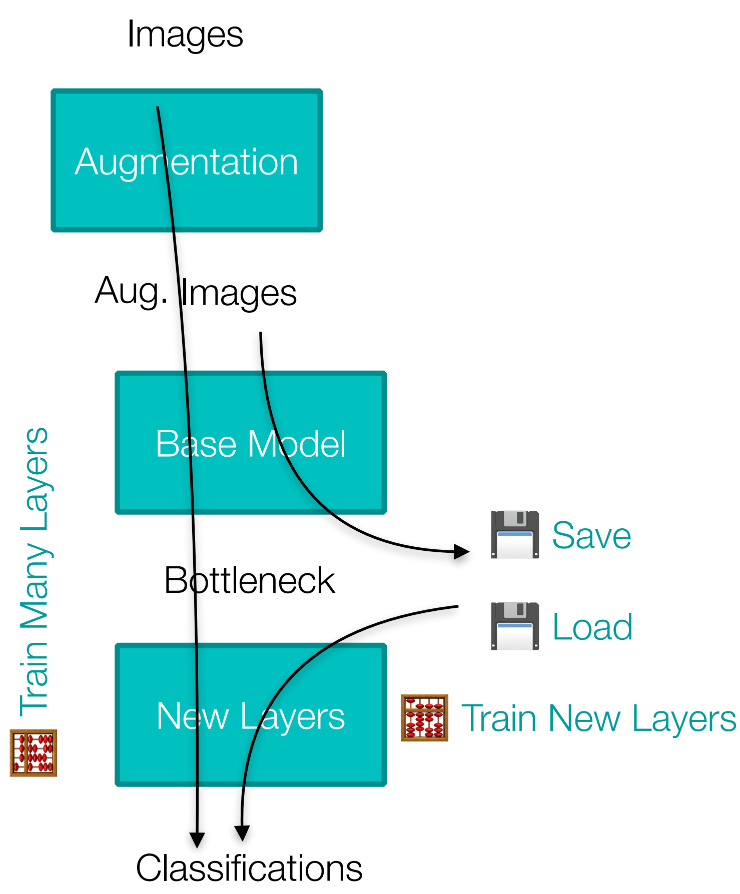
```
bottleneck = Input(shape=model_base.output.shape)
outputs = Dense(NUM_CLASSES)(bottleneck)
model_top = Model(bottleneck, outputs)
```

```
model_total = Model(inputs, outputs)
```

https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/

Freezing and Fine-tuning

- Step 1, Freeze base model:
  - No update during back-propagation
  - Only update layers after the bottleneck
  - Optional: Augment a set of training data
  - Send training dataset through base model
    - Save out bottleneck features
  - Train bottleneck features in new task
    - Typically 5-10 epochs is sufficient, easy to overfit
    - Larger training step size is okay
- Step 2, Fine-tune, unfreeze a few layers in base model:
  - Setup images to use some type of augmentation
  - Attach newly trained model to pre-trained model
  - Train to your hearts content, use smaller training step size



Bottlenecking on Maneframe

Dolphins versus Sharks

Justin Ledford •

Follow Along: https://github.com/8000net/Transfer-Learning-Dolphins-and-Sharks
Or in the Master Repo:
02 Transfer Learning.ipynb

Another Great Example:
https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/

Popular Transfer Learning Models

- **Vision**:
  - ImageNet Architectures:
    - VGG, Inception, ResNet, Xception, EfficientNet
- **Audio**:
  - WaveNet, almost always WaveNet
- **Text**:
  - Word Embedding
    - Glove, Word2Vec, ConceptNet
  - Sentence Embedding
    - Universal Sentence Encoders (Google)
    - BERT (Google)
    - Skip-thought Vectors

- **From the Research my Students have done:**
  - VGG for transferring to gaze classification
  - VGG for swapped face detection
  - Domain adaptation for speaker authentication
  - YOLO/DarkNet for surgical instrument detection
  - GLOVE for similar instructions in a maintenance manual

Lecture Notes for

Neural Networks and Machine Learning

Transfer Learning

Next Time:
Multi-Modal and Multi-Task

Reading: Keras F-API