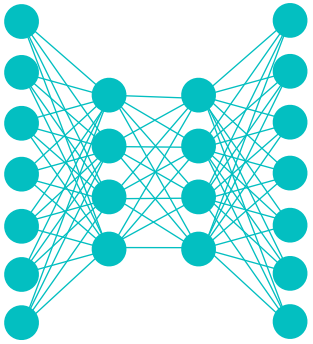
Lecture Notes for

Neural Networks and Machine Learning

Transfer Learning



Logistics and Agenda

- Logistics
  - Style Transfer Lab Due Soon (or is it?)
- Agenda
  - Transfer Learning Overview
- Next Time:
  - Paper Presentation: X-vectors

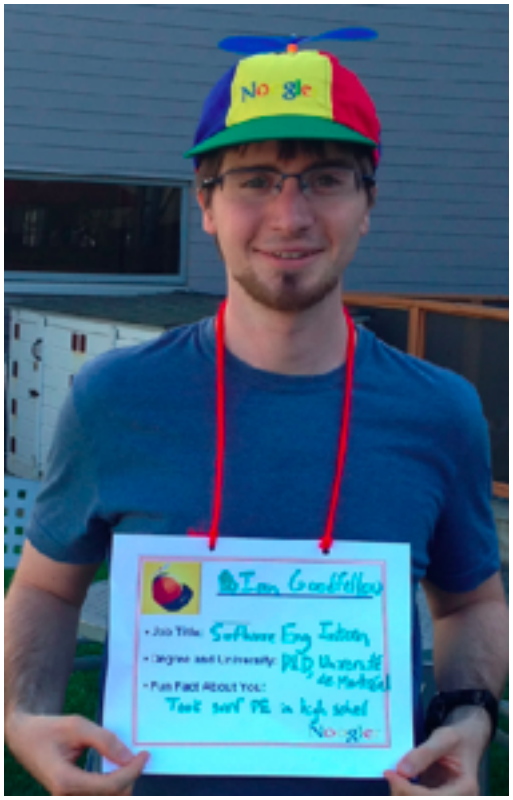
Transfer Learning: A Love Story

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**: Human can generalize what they learn to any domain, how to mimic 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 a variety of names 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/cumulative learning

Precise Definition of Transfer Learning

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

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

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

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

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

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

- Domain defines the features used
- Marginal Distribution of observing instances in the feature space
  - Typically intractable to calculate (generative)

- Task is within a domain
- Label space is typically one specific classification or regression task
- Probability of observing label given the feature space:
  - Not intractable (discriminative)

Definition with Examples

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

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

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

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

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

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

- Image Pixels
- Sensor Readings
- Text
- Anything that we can represent was a feature

- Object Classification
- Dolphin/Shark Classification
- Sentiment Analysis
- Any labeled task in a domain

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:
  - **Feature space**: docs in two different languages $\mathcal{X}_S \neq \mathcal{X}_T$
  - **Marginals**: docs discuss differing topics $p(X_S) \neq p(X_T)$
  - **Conditional**: docs have different label distributions or possibly different labels $p(Y_s|X_S) \neq p(Y_T|X_T)$

Categories of Transfer Learning

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

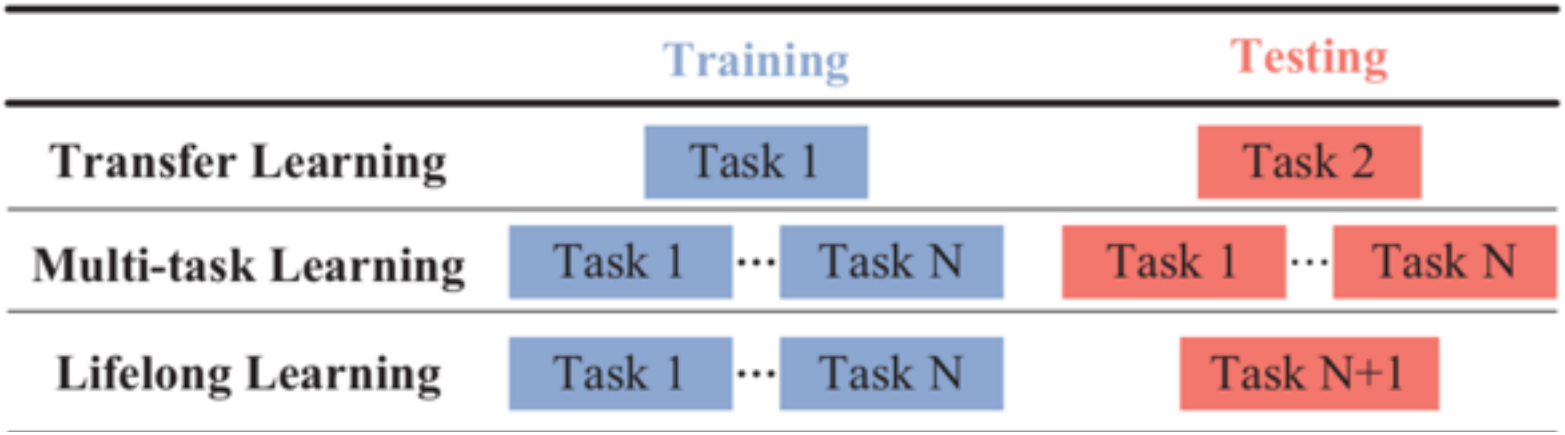
$$\mathcal{D} = \{\mathcal{X}, p(X)\} \qquad \mathcal{T} = \{\mathcal{Y}, p(Y|X)\}$$

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

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

- **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**: Different Domains, Different Tasks
  - Learning to paint art and learning to be a surgeon
  - Not yet a field with much repeatable traction

Aside: Other categorizations



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.

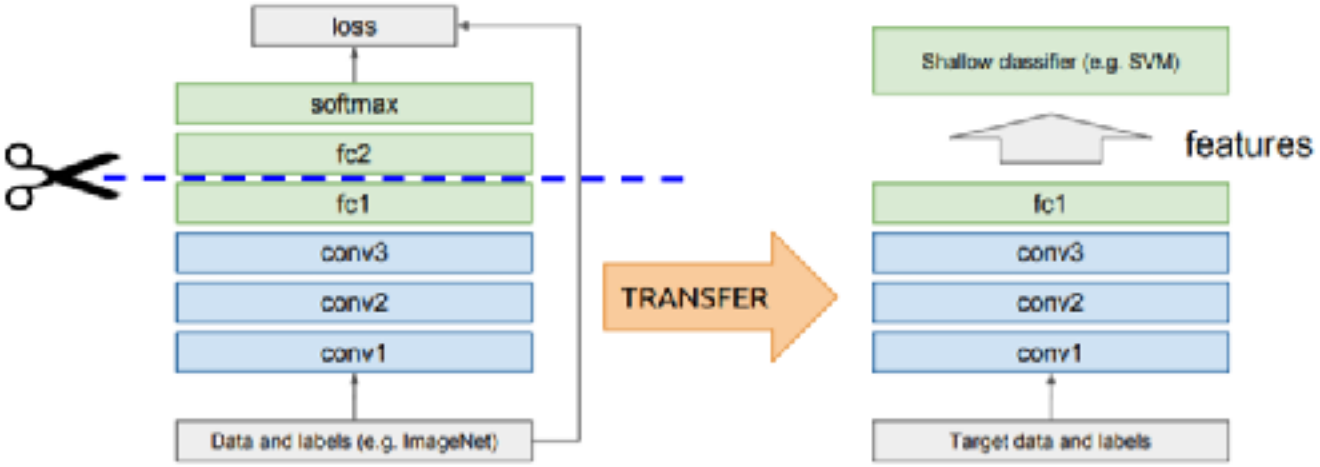
7 Related Work

Deep Transfer Learning

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

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



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

Freezing and Fine-tuning

- Freeze:
  - No update during back-propagation
  - Used when you want to avoid over fitting because target domain labels are fewer
- Fine-tune:
  - Update weights during back-propagation
  - Overfitting is a problem:
    - Use some type of augmentation
    - Or, have more numerous target domain labels
- Adaptive learning rates:
  - Set learning rates smaller for earlier layers, use vanishing gradients as positive property

Bottleneck

- Frozen training layers:
  - Why waste computations?
  - Computing more than one forward pass one the same data is called the “bottleneck”
  - Just save them out
- In keras, build multiple entry and exit points in the computation graph
  - **Input** to **Output**
  - **Input** to **Bottleneck Out**
  - **Bottleneck Out** to **Output**

Bottleneck Training

- Augment a set of training data initially
- Send augmented dataset through a pre-trained (base) model
- Save out bottleneck features
- Train bottleneck features in new task
  - Typically 5-10 epochs is sufficient
  - Same as freezing initial weights
- Fine Tuning
  - Attach newly trained model to pre-trained model
  - Continue with typical image augmentation
    - Typically run for as many epochs as possible
  - Not required to re-train “base” network

Bottlenecking on Maneframe

Dolphins versus Sharks

Self-supervised demo code provided for you

Follow Along:https://github.com/8000net/Transfer-Learning-Dolphins-and-Sharks

Popular Transfer Learning Models

- **Vision**:
  - ImageNet Architectures:
    - VGG, Inception, ResNet, Xception
- **Text**:
  - Word Embedding
    - Glove, Word2Vec, ConceptNet
  - Sentence Embedding
    - Universal Sentence Encoders (Google)
    - BERT (Google)
    - …note that sentence embedding might not be a good model of anything yet…

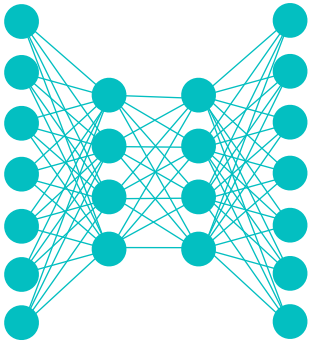
Lecture Notes for

Neural Networks and Machine Learning

Transfer Learning

Next Time:
Multi-Modal and Multi-Task

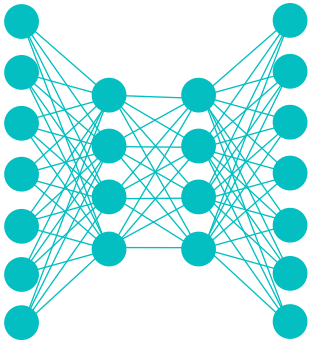
Reading: Keras F-API



Lecture Notes for

Neural Networks and Machine Learning

Adaptive, Self-supervised, Multi-modal, & Multi-task Learning



Logistics and Agenda

- Logistics
  - Newest Lab uses multi-task and multi-modal learning

- Agenda
  - Adaptive Learning
  - Self-Supervised Learning
  - Paper Presentation: X-vectors
  - Multi-modal/task Learning
    - Techniques
    - Applications and domains
- Next Time:
  - Paper Presentation: Multi-task Methods in Chemistry

Last Time

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

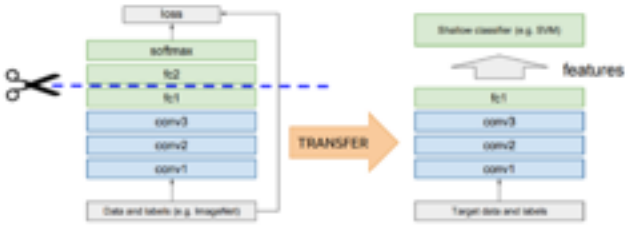
$\mathcal{D} = \{\mathcal{X}, p(X)\}$ $\quad \mathcal{T} = \{\mathcal{Y}, p(Y|X)\}$

Domain: Feature Space, Probability Observation

Task: Label Space, Learned Probability

- Domain defines the features used
- Marginal Distribution of observing instances in the feature space
  - Typically intractable to calculate (generative)
- Task is within a domain
- Label space is typically one specific classification or regression task
- Probability of observing label given the feature space:
  - Not intractable (discriminative)

- 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



	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

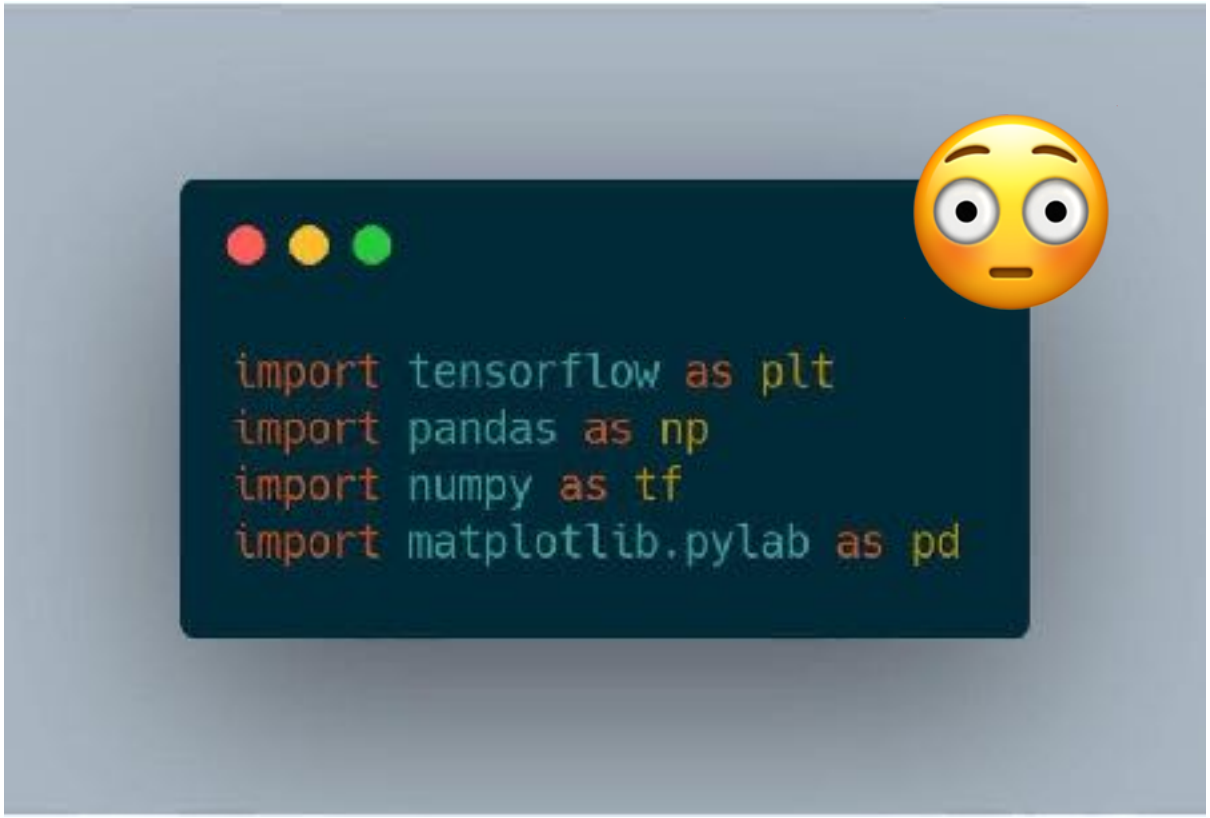
Humans can learn to ride a bike and use that to understand better about driving a car. Machine Learning in its current form is far from this capability. How can we move our siloed version of artificial intelligence closer to the process of human based learning? How can we accumulate knowledge from model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

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."

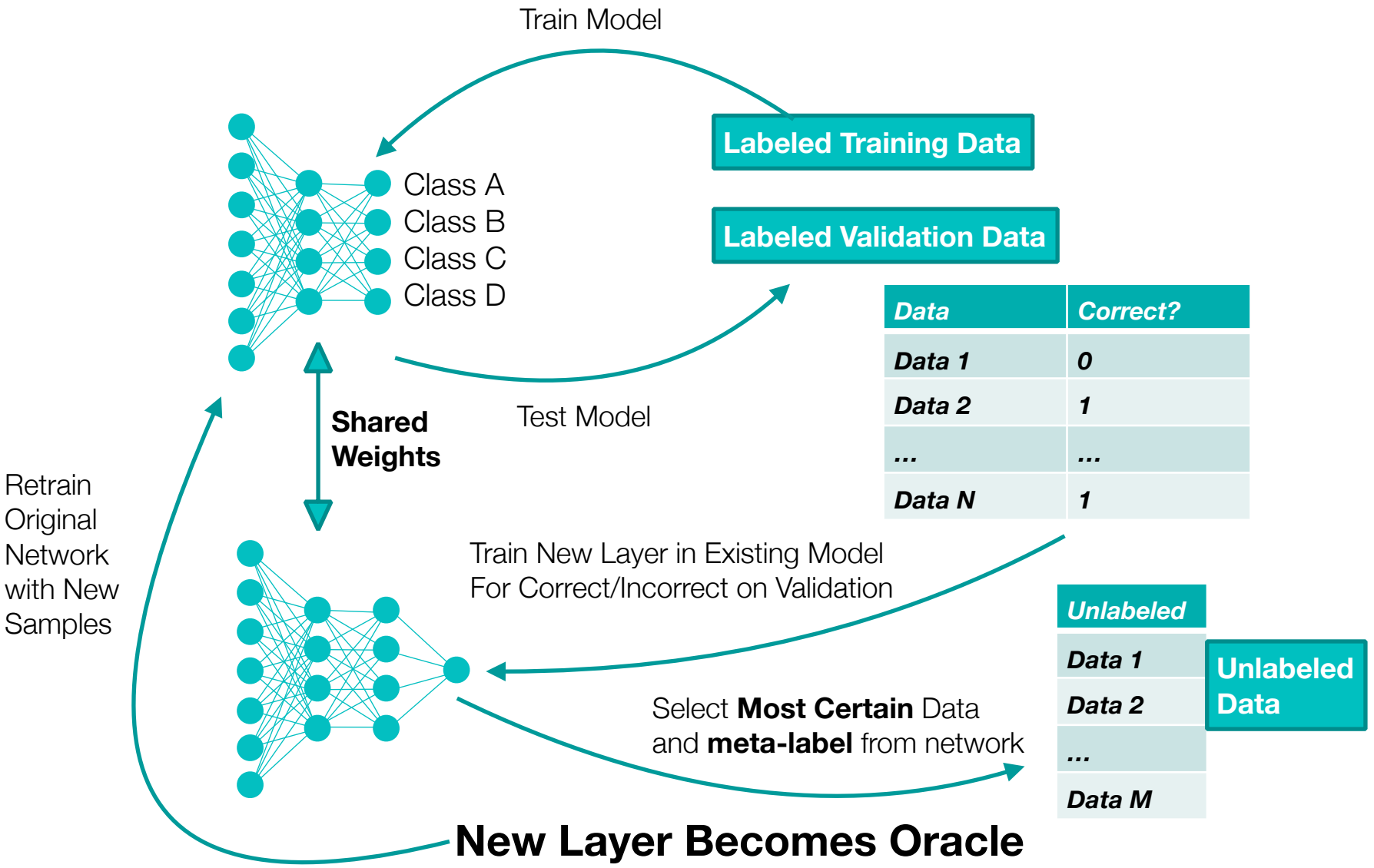
Active Transfer Learning



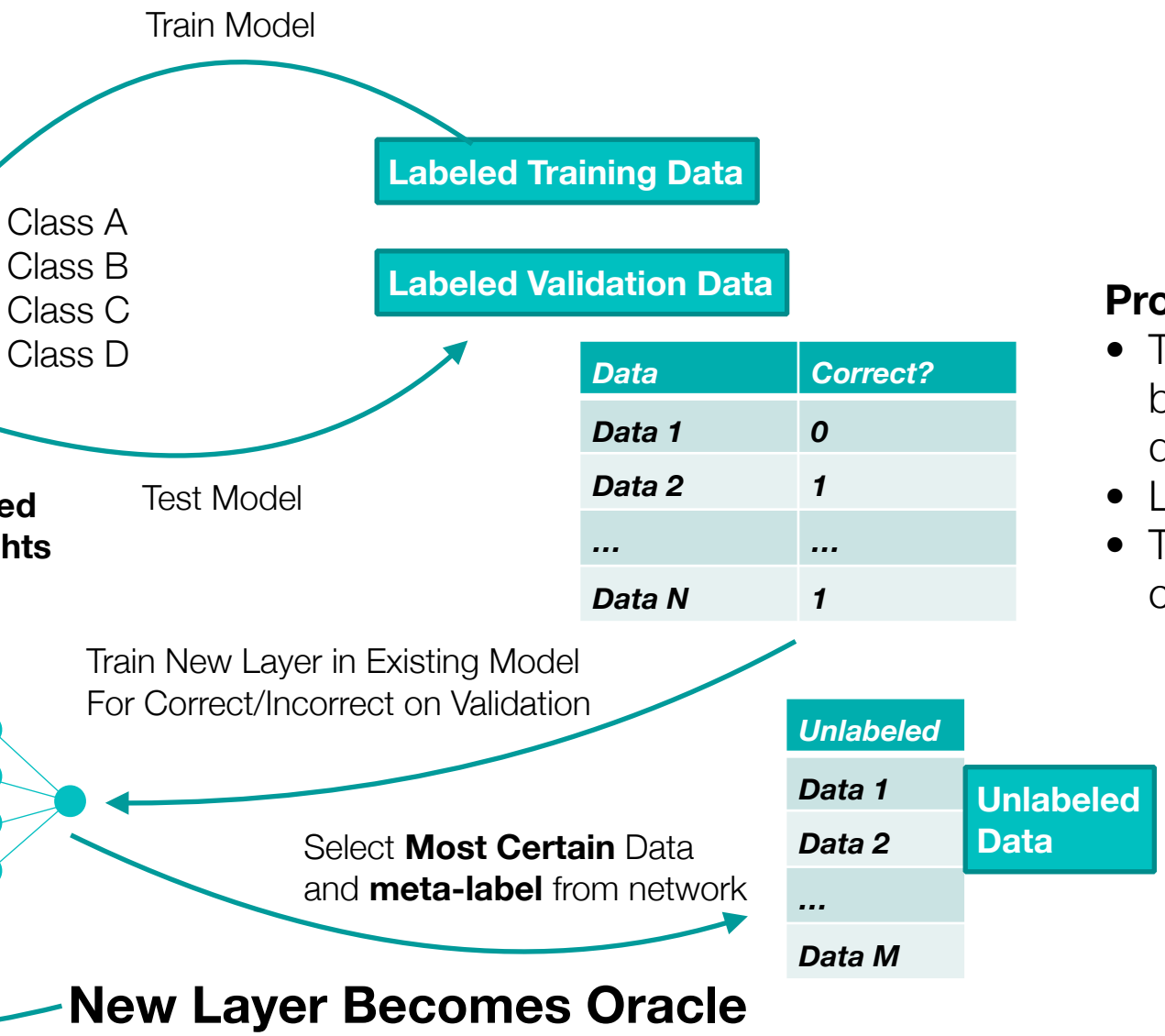
Active Learning Overview

- **Basic Idea**: Use a trained model to sample from an oracle that can magically give you a new label
  - What labels should we ask the oracle about?
- Uncertainty Sampling
  - Choose instances where the model is most uncertain or certain
  - Various ways to measure certainty
- Diversity Sampling
  - Choose instances that are similar or different from training distribution

Uncertainty Sampling with a Neural Network



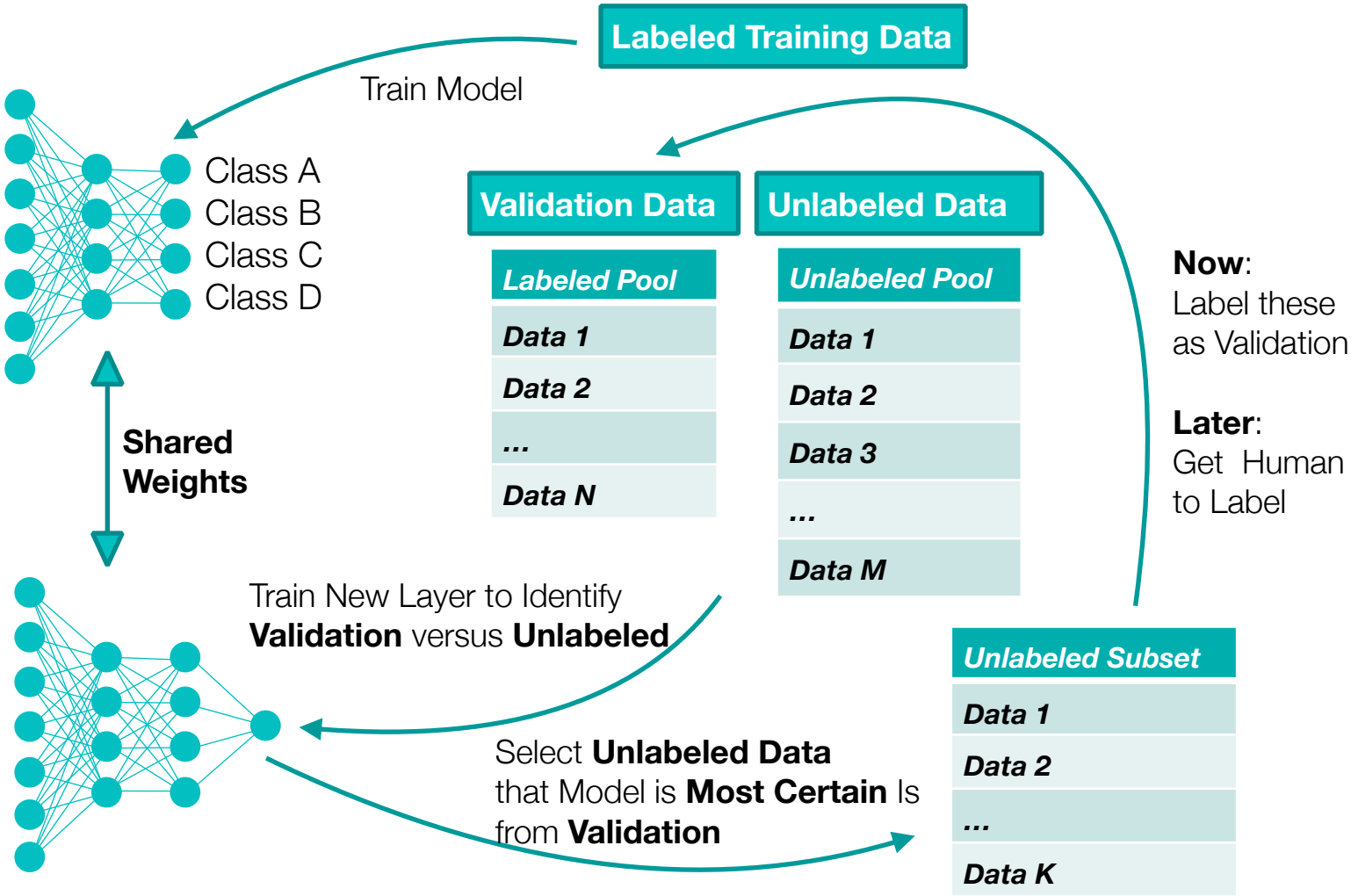
Uncertainty Sampling with a Neural Network



Problems:

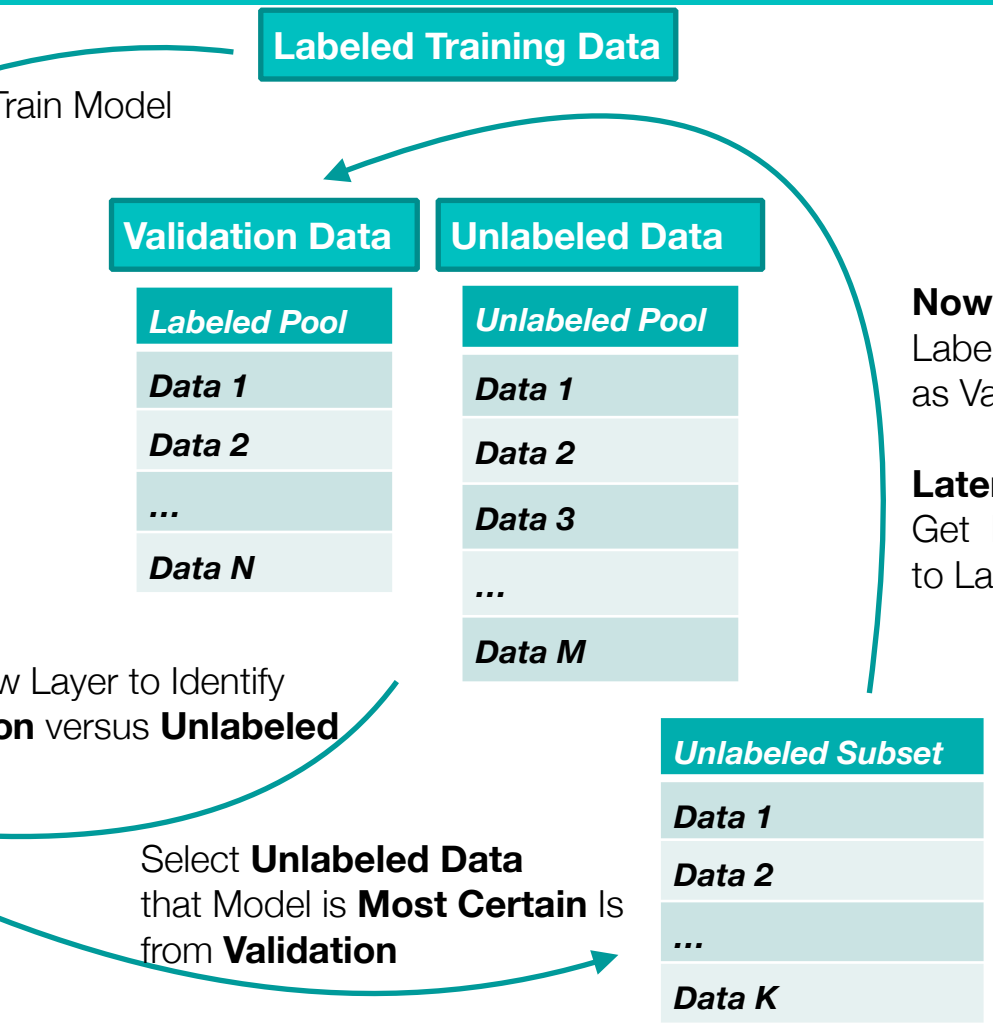
- Training pool is represented by classes the model already does well predicting
- Limited diversity of Samples
- Training pool can become contaminated

Diversity Sampling with a Neural Network



New Layer Decides if Samples are Added to Validation Data

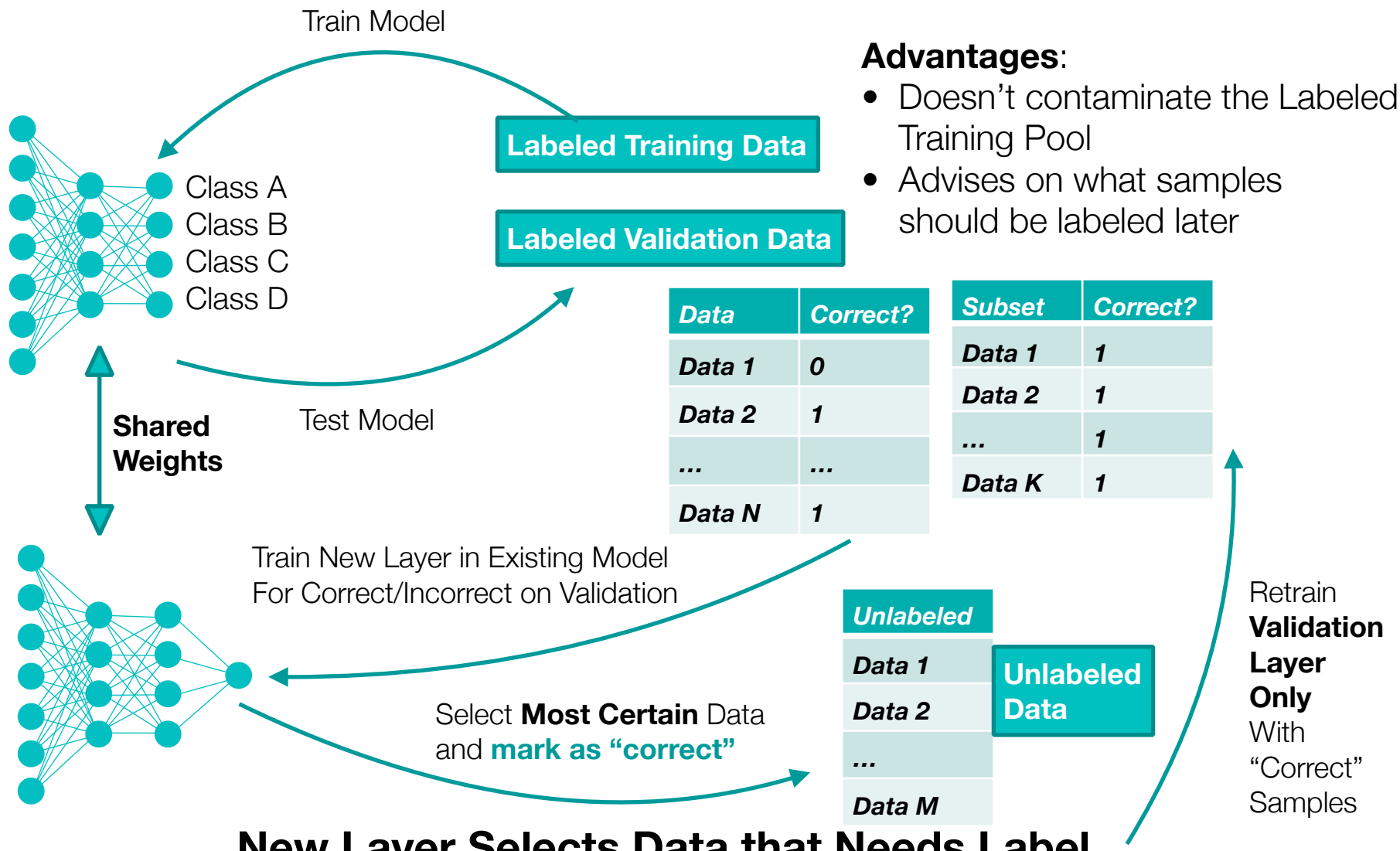
Diversity Sampling with a Neural Network



Discussion:

- Training pool is not contaminated
- Expands validation data in well mannered way, not adding too "far away" samples
- Validation versus Unlabeled might not be the best comparison, because it ignores confusions in the training data

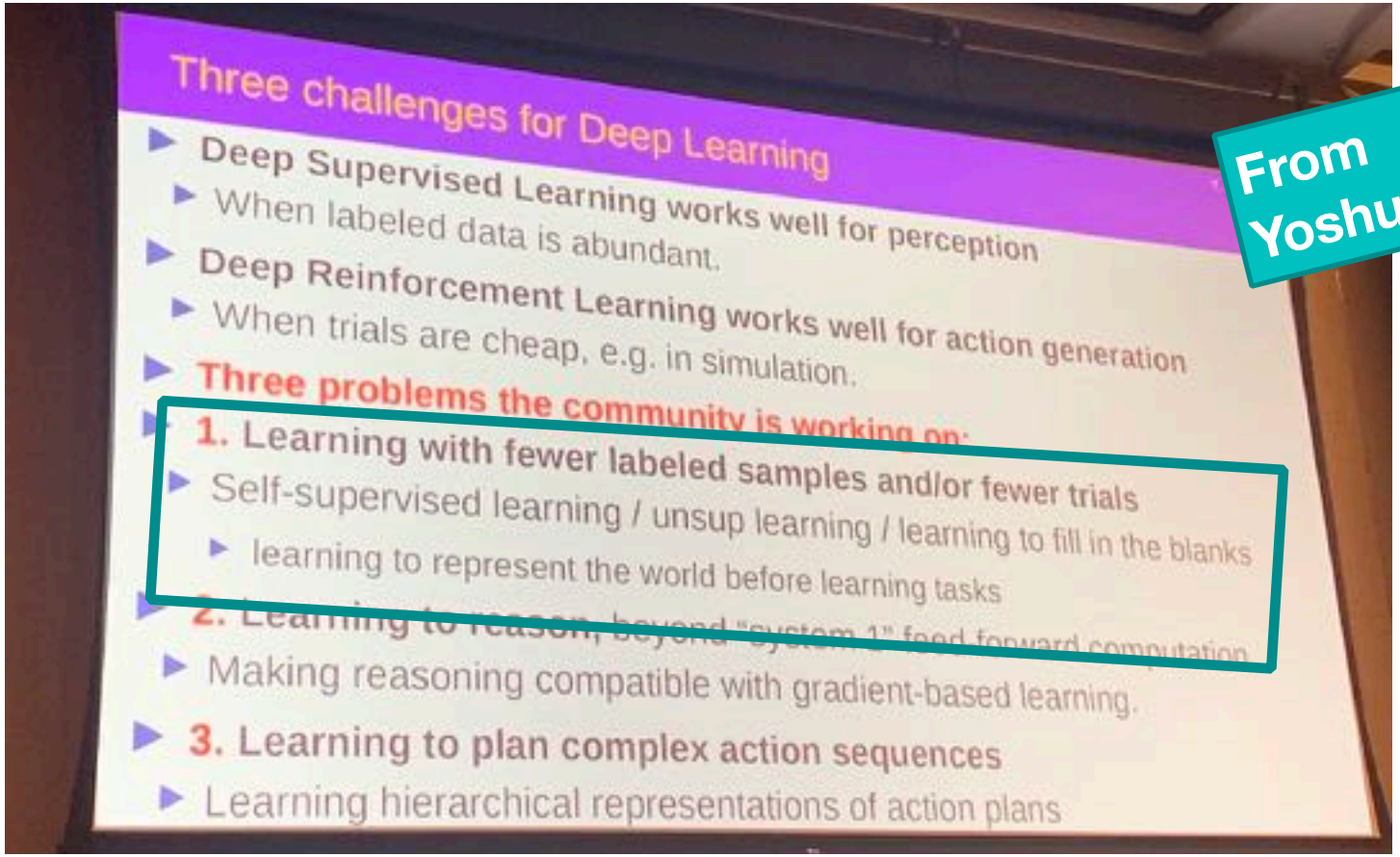
ATLAS: Active Transfer Learning for Adaptive Sampling



Advantages:

- Doesn't contaminate the Labeled Training Pool
- Advises on what samples should be labeled later

Self-Supervised Learning

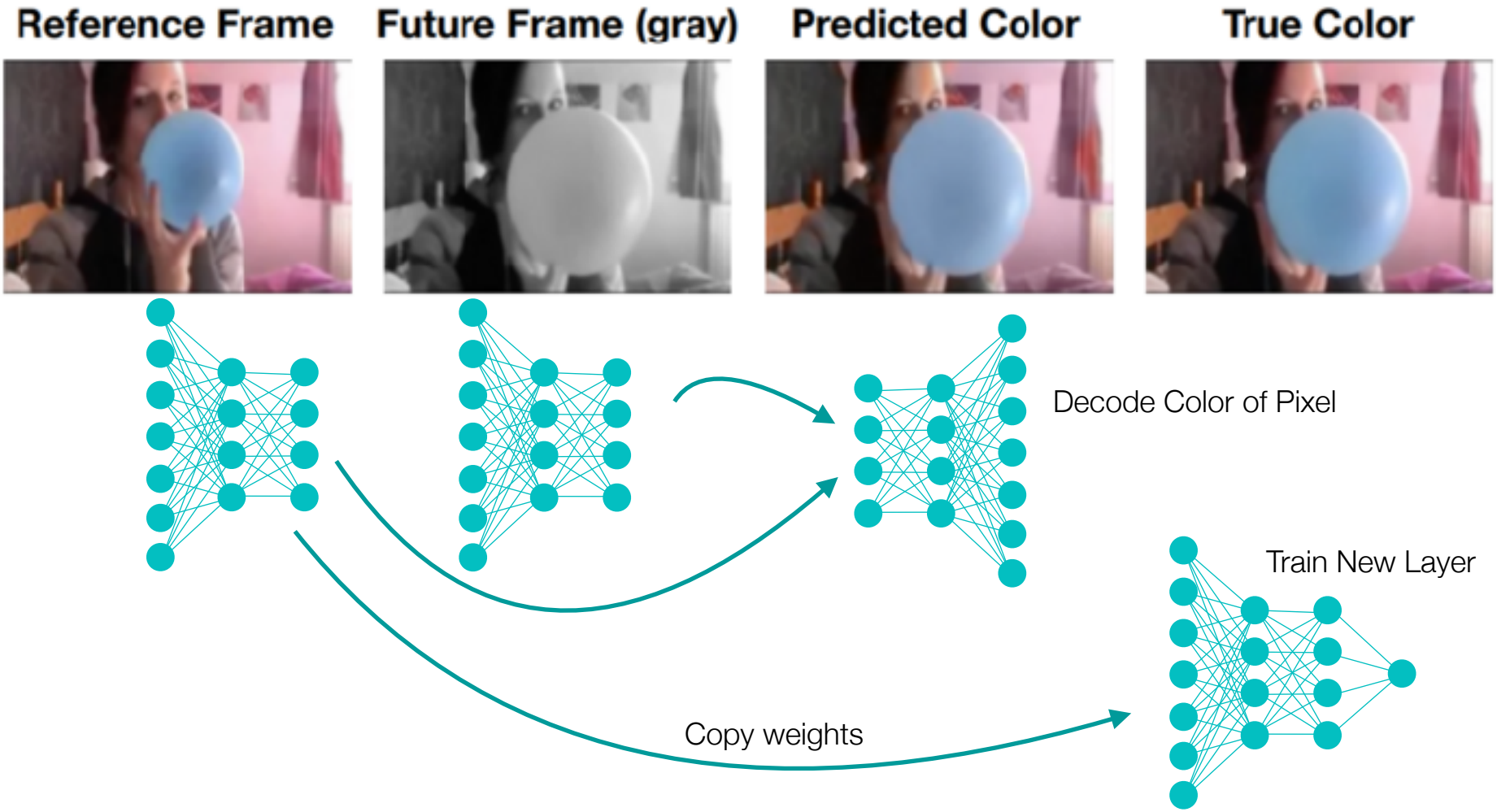


From Yoshua Bengio

Self-supervised Learning

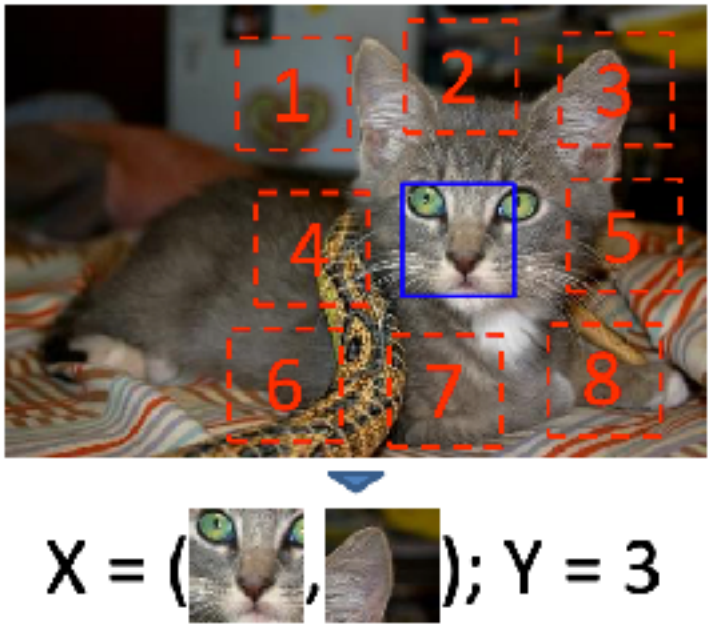
- **Problem**: deep learning is not sample efficient
- **Idea**: learn about the world before learning the task
- **New Problem**: how do we learn about the world?
- **Solution**:
  - train on auxilliary task (pretext) that is easy to label
  - throw away anything specific to auxilliary task
  - train new network with task of interest, transferring knowledge (downstream task)

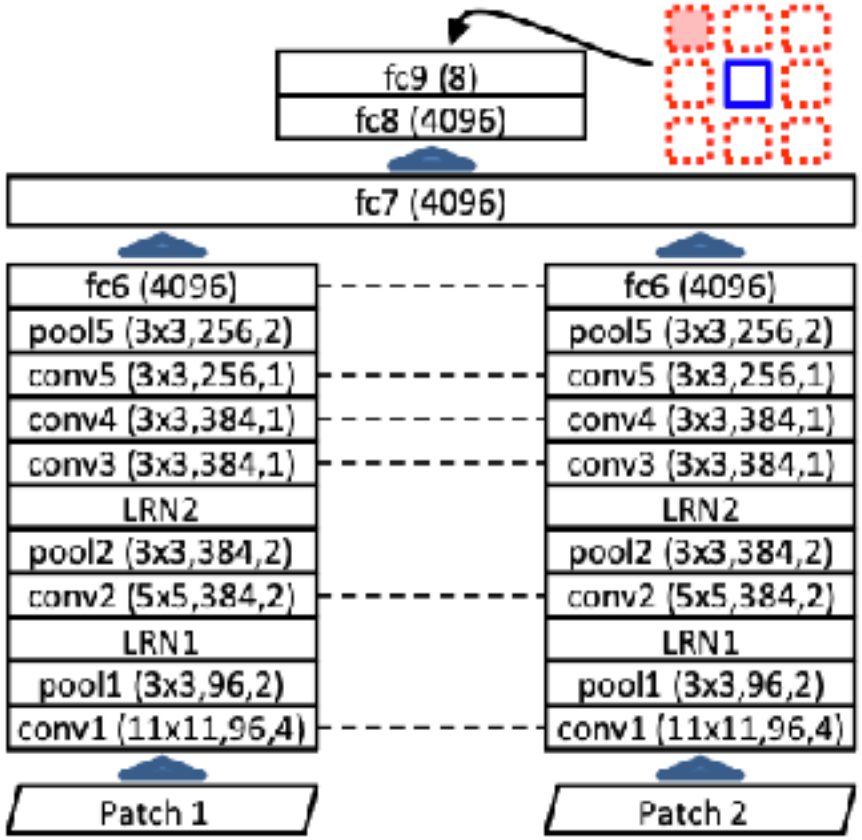
Examples of SSL



https://www.fast.ai/2020/01/13/self_supervised/

Examples of SSL





Unsupervised Visual Representation Learning by Context Prediction

Carl Doersch[1,2] Abhinav Gupta[1] Alexei A. Efros[2]
[1] School of Computer Science Carnegie Mellon University
[2] Dept. of Electrical Engineering and Computer Science University of California, Berkeley

https://www.fast.ai/2020/01/13/self_supervised/

Examples of SSL

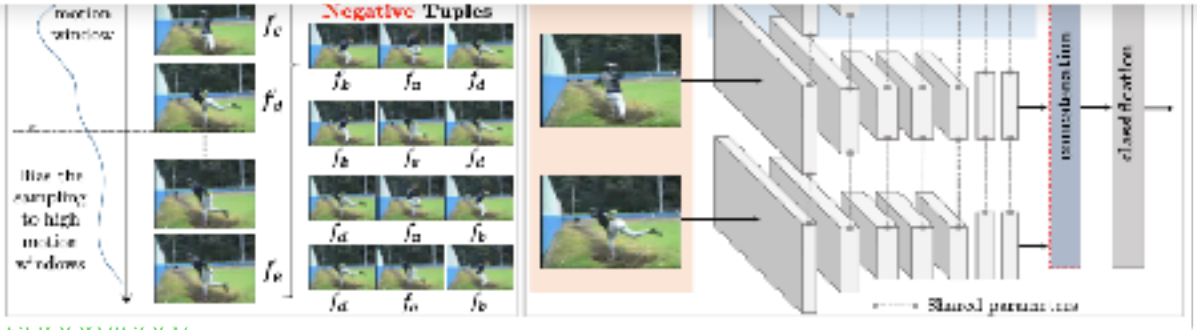
Shuffle and Learn: Unsupervised Learning using Temporal Order Verification

Ishan Misra[1] C. Lawrence Zitnick[2] Martial Hebert[1]

[1] The Robotics Institute, Carnegie Mellon University
[2] Facebook AI Research

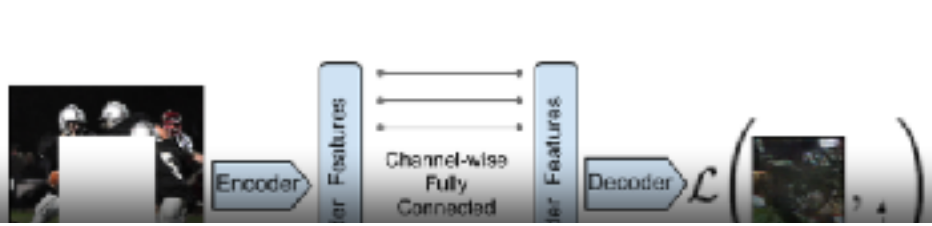
Table 2: Mean classification accuracies over the 3 splits of UCF101 and HMDB51 datasets. We compare different initializations and finetune them for action recognition.

Dataset	Initialization	Mean Accuracy
UCF101	Random	38.6
	(Ours) Tuple verification	**50.2**
HMDB51	Random	13.3
	UCF Supervised	15.2
	(Ours) Tuple verification	**18.1**

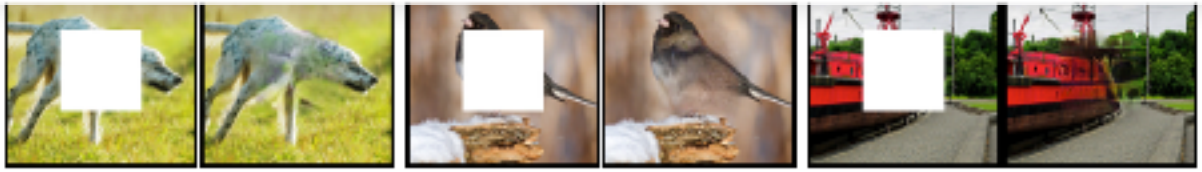


https://www.fast.ai/2020/01/13/self_supervised/

Examples of SSL



Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Wang *et al.* [39]	motion	1 week	58.7%	47.4%	-
Doersch *et al.* [7]	relative context	4 weeks	55.3%	46.6%	-
Ours	context	14 hours	56.5%	44.5%	30.0%



Context Encoders: Feature Learning by Inpainting

Deepak Pathak Philipp Krähenbühl Jeff Donahue Trevor Darrell Alexei A. Efros
University of California, Berkeley

https://www.fast.ai/2020/01/13/self_supervised/

Unsupervised Consistency Loss

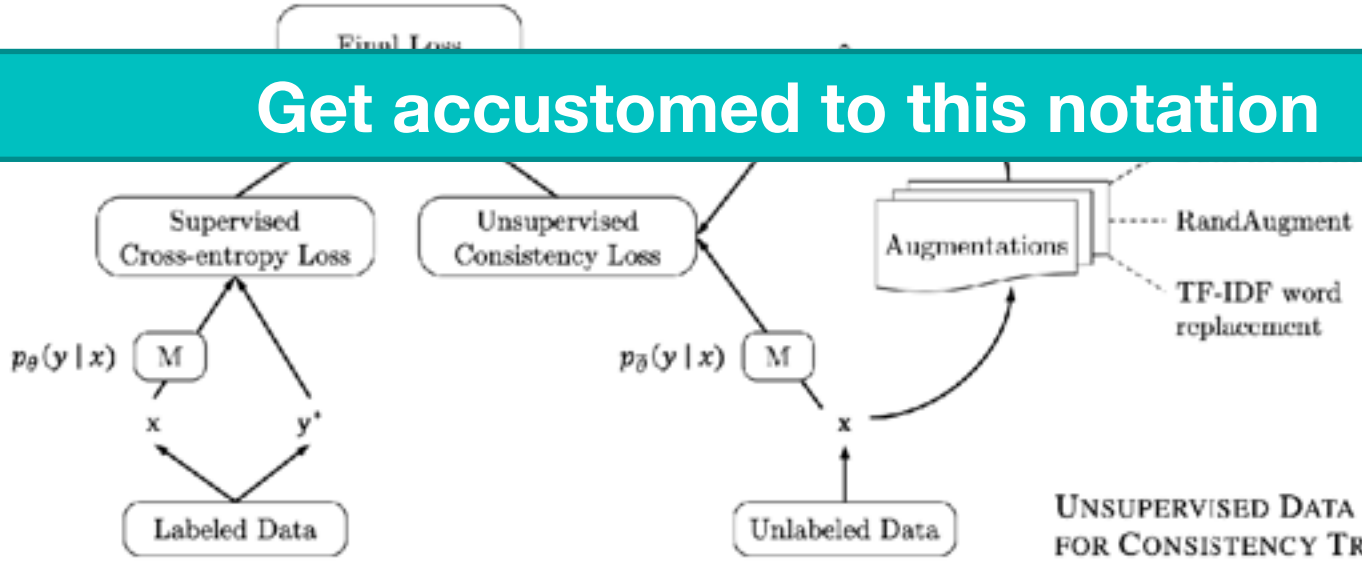
$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathbf{E}_{\mathbf{x}\in U}\mathbf{E}_{\hat{\mathbf{x}}\leftarrow q(\hat{\mathbf{x}}|\mathbf{x})}\left[\mathscr{D}_{KL}\left(p_{\hat{w}}(y\,|\,\mathbf{x})\,||\,p_{w}(y\,|\,\hat{\mathbf{x}})\right)\right]}^{\text{consistency in augmentation}}$$

no back prop **yes** back prop

Neural Network approximates $p(y|\boldsymbol{x})$ by $\boldsymbol{w}$
Use labeled data to minimize network

Sample new $\boldsymbol{x}$ from unlabeled pool with function q
function q is augmentation procedure
Minimize cross entropy of two models

Get accustomed to this notation



UNSUPERVISED DATA AUGMENTATION FOR CONSISTENCY TRAINING

Qizhe Xie[1,2], Zihang Dai[1,2], Eduard Hovy[2], Minh-Thang Luong[1], Quoc V. Le[1]
[1] Google Research, Brain Team, [2] Carnegie Mellon University

Unsupervised Consistency Loss

$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathbf{E}_{\mathbf{x}\in U}\mathbf{E}_{\hat{\mathbf{x}}\leftarrow q(\hat{\mathbf{x}}|\mathbf{x})}\left[\mathscr{D}_{KL}\left(p_{\hat{w}}(y\,|\,\mathbf{x})\,||\,p_{w}(y\,|\,\hat{\mathbf{x}})\right)\right]}^{\text{consistency in augmentation}}$$

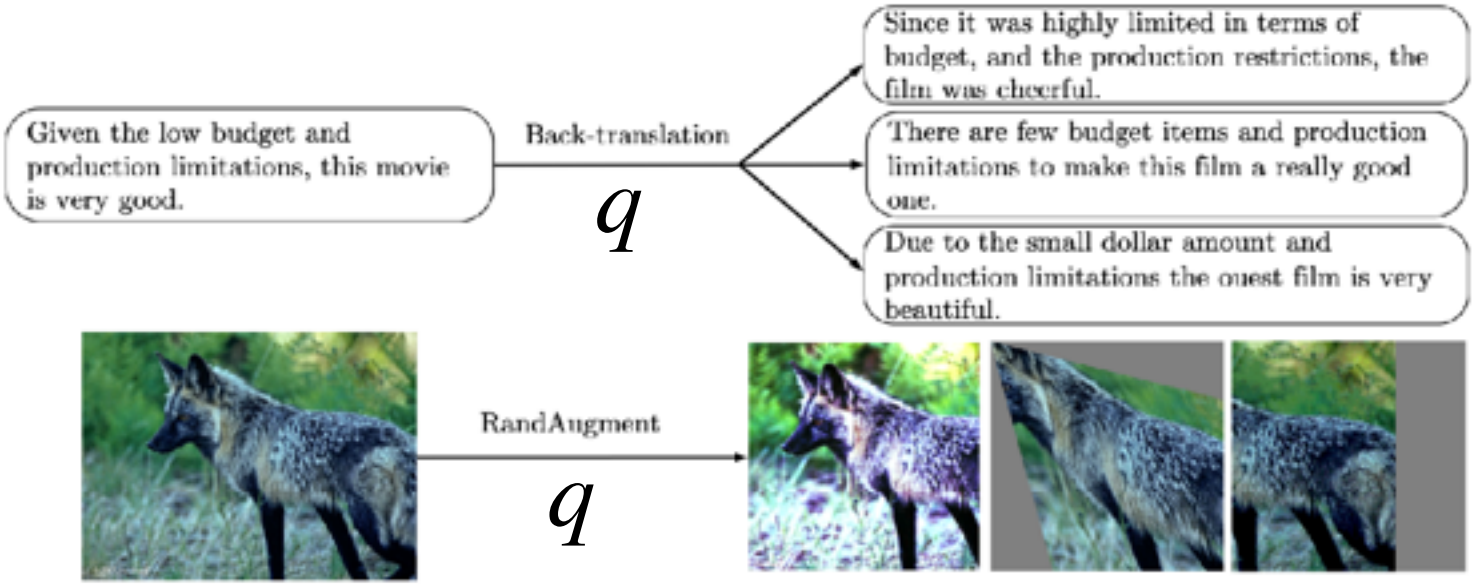


Figure 2: Augmented examples using back-translation and RandAugment.

Unsupervised Consistency Loss

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	16.17
Cutout	4.42	6.42
RandAugment	**4.23**	**5.29**

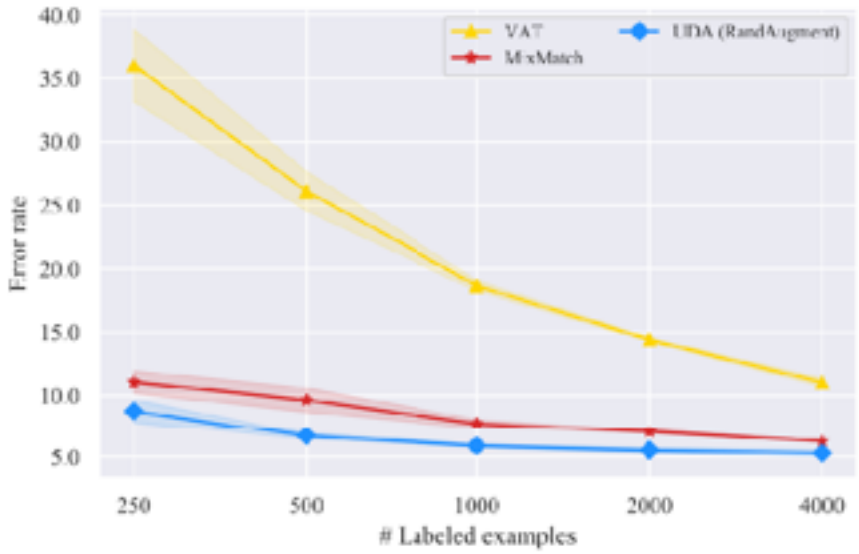
Table 1: Error rates on CIFAR-10.

Augmentation (# Sup examples)	Sup (650k)	Semi-sup (2.5k)
✗	38.36	50.80
Switchout	37.24	43.38
Back-translation	**36.71**	**41.35**

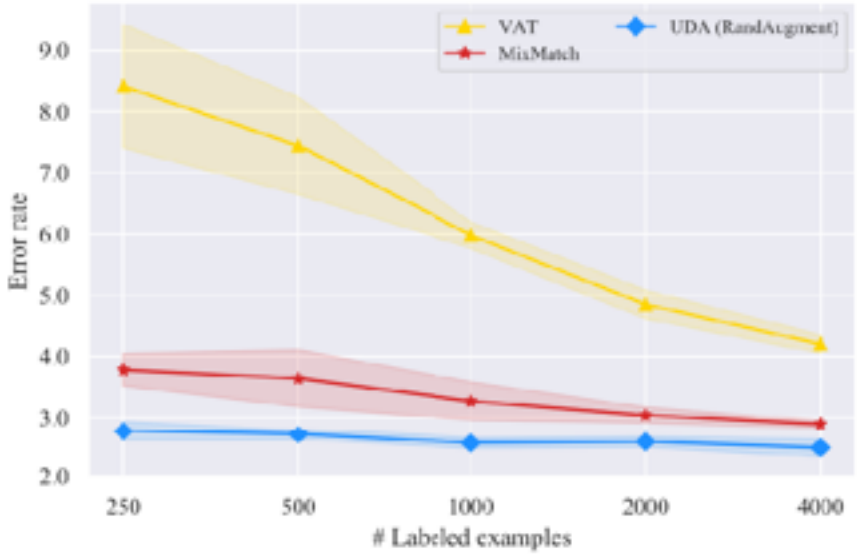
Table 2: Error rate on Yelp-5.

UNSUPERVISED DATA AUGMENTATION FOR CONSISTENCY TRAINING

Qizhe Xie[1,2], Zihang Dai[1,3], Eduard Hovy[3], Minh-Thang Luong[1], Quoc V. Le[1]
[1] Google Research, Brain Team, [2] Carnegie Mellon University



(a) CIFAR-10



(b) SVHN

Unsupervised Consistency Loss

UNSUPERVISED DATA AUGMENTATION FOR CONSISTENCY TRAINING

Qizhe Xie[1,2], Zihang Dai[1,3], Eduard Hovy[3], Minh-Thang Luong[1], Quoc V. Le[1]
[1] Google Research, Brain Team, [2] Carnegie Mellon University

Method	Model	# Param	CIFAR-10 (4k)	SVHN (1k)
Π-Model (Laine & Aila, 2016)	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher (Tarvainen & Valpola, 2017)	Shake-Shake	26M	6.28 ± 0.15	-
Fast-SWA (Athiwaratkun et al., 2018)	Shake-Shake	26M	5.0	-
MixMatch (Berthelot et al., 2019)	WRN	26M	4.95 ± 0.08	-
UDA (RandAugment)	WRN-28-2	1.5M	5.29 ± 0.25	**2.55 ± 0.09**
UDA (RandAugment)	Shake-Shake	26M	3.7	-
UDA (RandAugment)	PyramidNet	26M	**2.7**	-

Methods	SSL	10%	100%
ResNet-50	✗	55.09 / 77.26	77.28 / 93.73
w. RandAugment		58.84 / 80.56	78.43 / 94.37
UDA (RandAugment)	✓	**68.78 / 88.80**	**79.05 / 94.49**

Table 5: Top-1 / top-5 accuracy on ImageNet with 10% and 100% of the labeled set. We use image size 224 and 331 for the 10% and 100% experiments respectively.

Paper Presentation: X-Vectors

PROBING THE INFORMATION ENCODED IN X-VECTORS

Desh Raj, David Snyder, Daniel Povey, Sanjeev Khudanpur

Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA.

draj@cs.jhu.edu, {david.ryan.snyder, dpovey}@gmail.com, khudanpur@jhu.edu

ABSTRACT

Deep neural network based speaker embeddings, such as x-vectors, have been shown to perform well in text-independent speaker recognition/verification tasks. In this paper, we use simple classifiers to investigate the contents encoded by x-vector embeddings. We probe these embeddings for information related to the speaker, channel, transcription (sentence, words, phones), and meta information about the utterance (duration and augmentation type), and compare these with the information encoded by i-vectors across a varying num-

x-vector extractors trained with and without augmentation sheds some light on the possible reason behind this improvement. Previous work has shown that i-vectors, though developed for speaker recognition, can improve automatic speech recognition (ASR), because they capture speaker and channel characteristics [7]. Our probing task results suggest that x-vectors also capture similar information and hence motivate their use for speaker adaptation in ASR.

Wang et al. [8] have previously conducted similar investigations for i-vectors [9] and d-vectors [3]. However, their

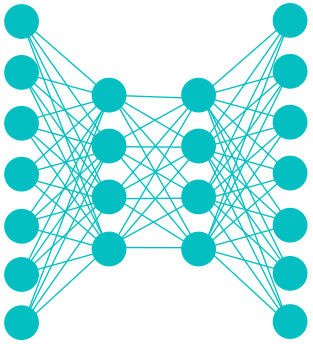
Lecture Notes for

Neural Networks and Machine Learning

Ada, SSL,

Next Time:
M-Modal/task

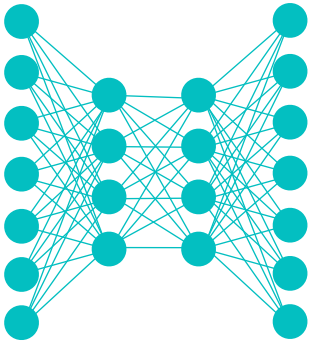
Reading: Papers



Lecture Notes for

Neural Networks and Machine Learning

Adaptive, Self-supervised, Multi-modal, & Multi-task Learning



Logistics and Agenda

- Logistics
  - Newest Lab uses multi-task and multi-modal learning

- Agenda
  - Adaptive Learning
  - Self-Supervised Learning
  - Paper Presentation: X-vectors
  - Multi-modal/task Learning
    - Techniques
    - Applications and domains
- Next Time:
  - Paper Presentation: Multi-task Methods in Chemistry

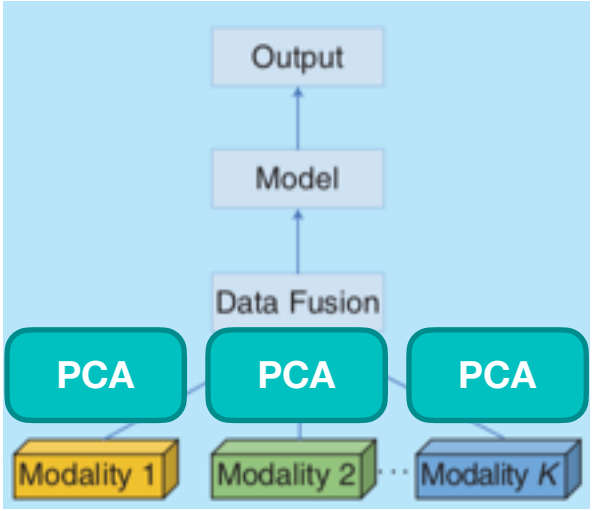
Multi-modal Review

Multi-modal == Multiple Data Sources

- **Modal** comes from the "sensor fusion" definition from Lahat, Adali, and Jutten (2015) for deep learning
- Using the Keras functional API, this is extremely easy to implement
  - … and we have used it since the previous 7000 level course!
- But now let's take a deeper dive and ask:
  - What are the different types of modalities that we might try?
  - Is there a more optimal layer to merge information?
  - Early, Intermediate, and late fusion

Early Fusion

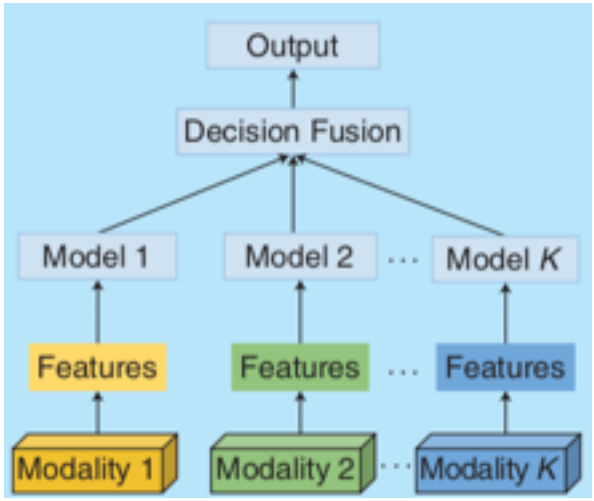
- Merge sensor layers early in the process
- **Assumption**: there is some data redundancy, but modes are conditionally independent
- **Problem**: architecture parameter explosion
  - One solution: dimensionality reduction or feature selection
  - Data Fusion



Ramamchandran and Taylor, 2017

Late Fusion

- Merge sensor layers right before flattening
- **Assumption**: little redundancy or conditional indepedence—better as ensemble architecture
- **Problem**: just separate classifiers, limited interplay
  - Need domain expert architecture
  - Decision Fusion

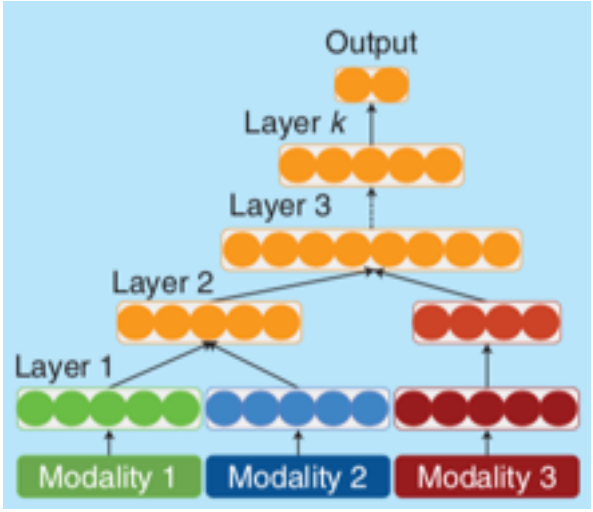


Ramamchandran and Taylor, 2017

Intermediate Fusion

- Merge sensor layers in soft way
- **Assumption**: some features interplay and others do not
- **Problem**: how to optimally tie layers together?

1. Stacked Auto-Encoders [Ding and Tao, 2015]
2. Early fuse layers that are correlated [Neverova et al 2016]
3. Fully train each modality merge based on criterion of similarity in activations [Lu and Xu 2018]



Ramamchandran and Taylor, 2017

Multiplicative Merging

$$\mathbf{u}_i = \sum_{k \in M_i} f(\mathbf{v}_k)$$ candidate modalities

$$p(\hat{Y}) = \sum_i \log[g_i(\mathbf{u}_i)]$$ average of i combined modalities

$$p(\hat{Y}) = \sum_i q_i \log[g_i(\mathbf{u}_i)]$$ weighted average of i modalities

$$p(\hat{Y}) = \sum_i \left[\prod_{j \neq i} 1 - g_j(\mathbf{u}_j) \right]^{\beta} \log[g_i(\mathbf{u}_i)]$$

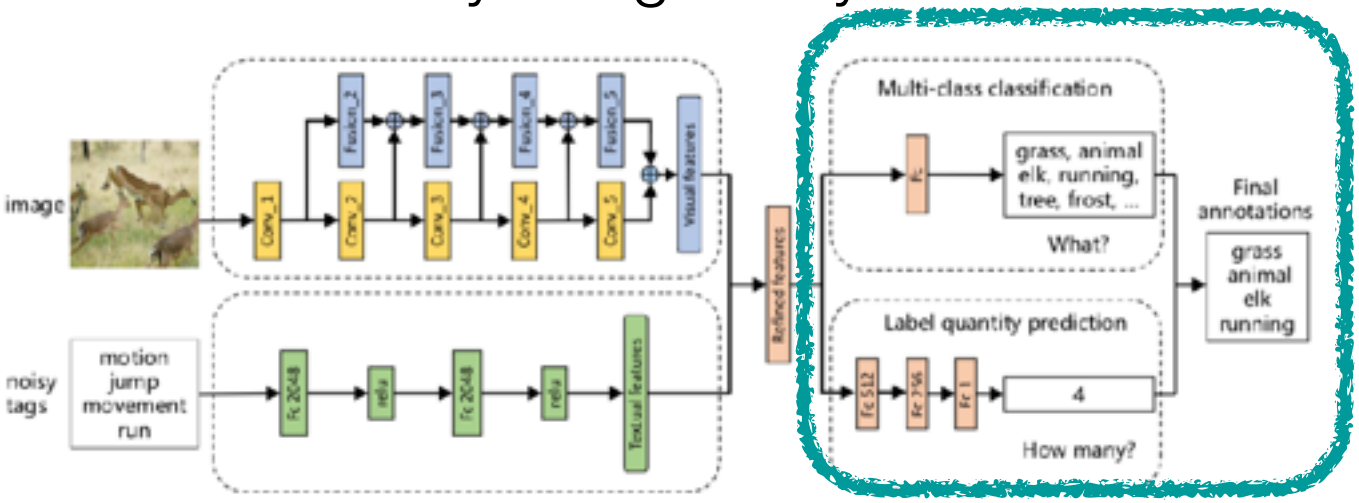
only weight correct class in the i modalities

Gender from Snapchat UserId, Activity		
	Mul Modality	***Fused***
Error	***5.86 +-0.02***	***7.97***
Error	***3.66+-0.01***	***5.15***

https://arxiv.org/pdf/1805.11730.pdf

Multi-modal Merging

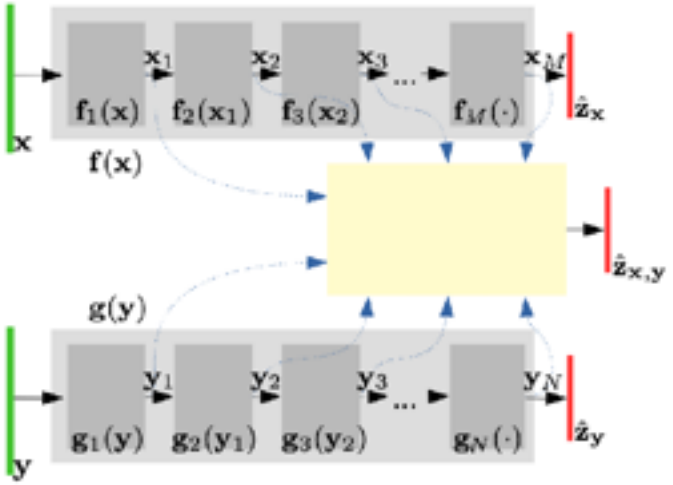
- **Still an open research problem**
- How to develop merging techniques that
  - Can handle exponentially many pairs of modalities
  - Automatically merge meaningful modes
  - Discard poor pairings
  - Selectively merge early or late



https://arxiv.org/pdf/1709.01220.pdf

Most current methods are still ad-hoc

Neural Architecture Search for Mode Fusion



Genetic Algorithm

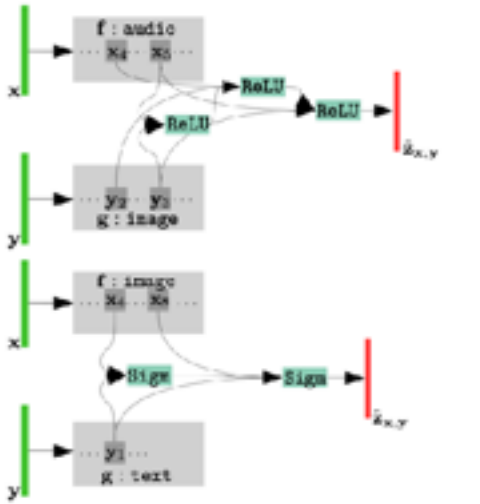
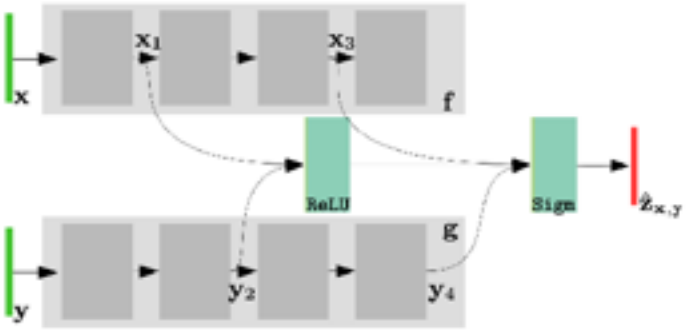


Figure 3. **Structure of the found fusion architectures**. First: AV-MNIST. Second: MM-IMDB.



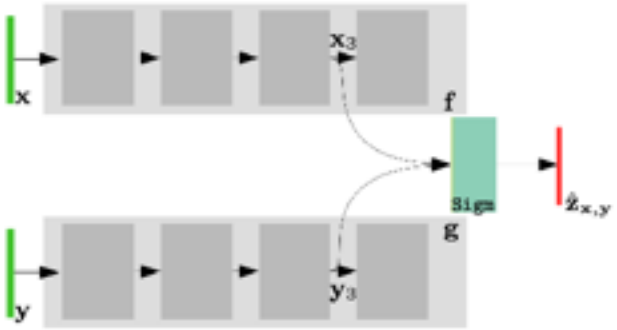
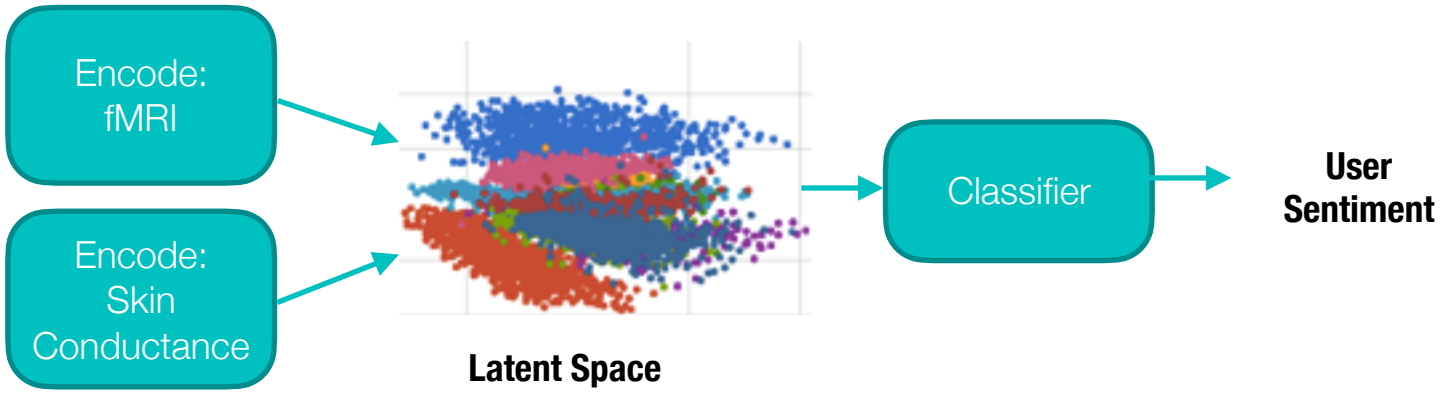


Figure 2. **Two realizations of our search space on a small bimodal network.** Left: network defined by $[(\gamma_1^m = 1, \gamma_1^n = 2, \gamma_1^p = 1), (\gamma_2^m = 3, \gamma_2^n = 4, \gamma_2^p = 2)]$. Right: network defined by $[(\gamma_1^m = 3, \gamma_1^n = 3, \gamma_1^p = 2)]$.

Pérez-Rúa, Juan-Manuel, Valentin Vielzeuf, Stéphane Pateux, Moez Baccouche, and Frédéric Jurie. "Mfas: Multimodal fusion architecture search." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6966-6975. 2019.

Approaches with Deep Learning

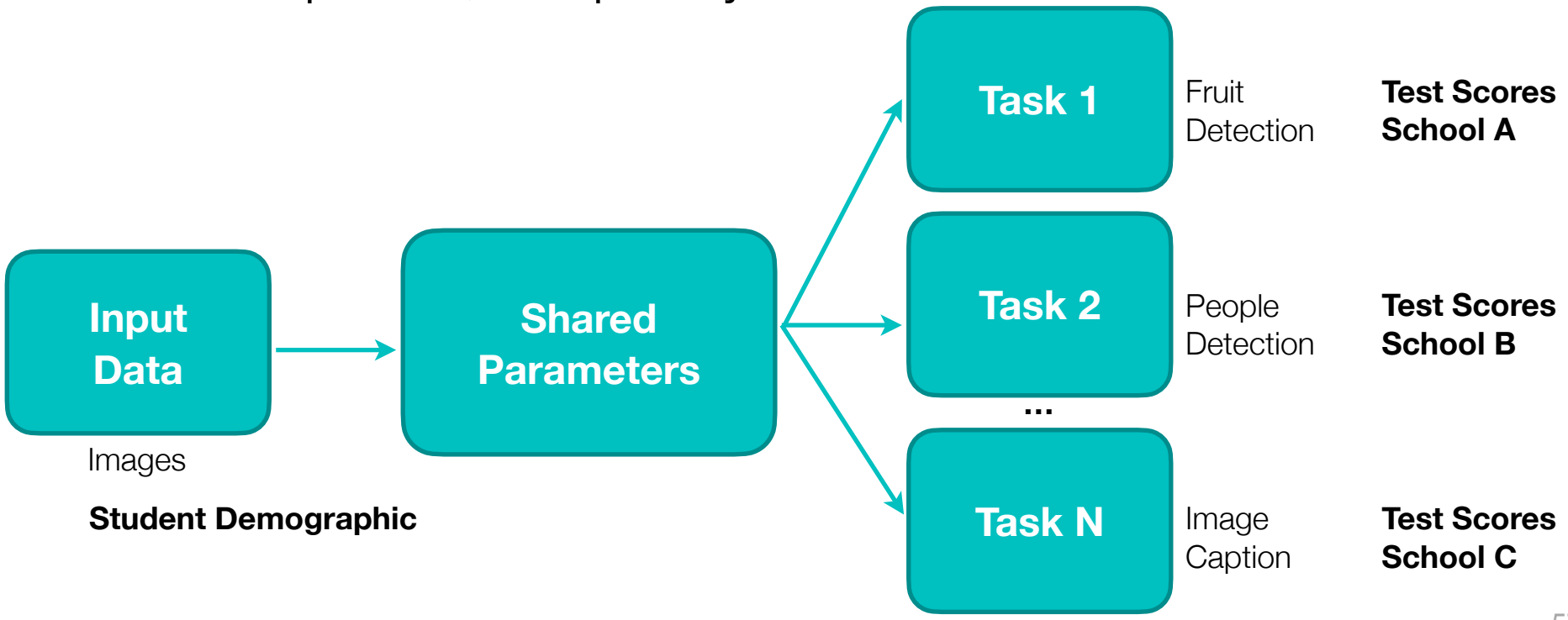
- Latent Space Transfer (universality)
  - From another domain, map to a similar latent space for the same task
  - Useful for unifiying data based upon a new input mode when old mode is well understood
    - for example, biometric data
    - I have never seen a research paper on this…



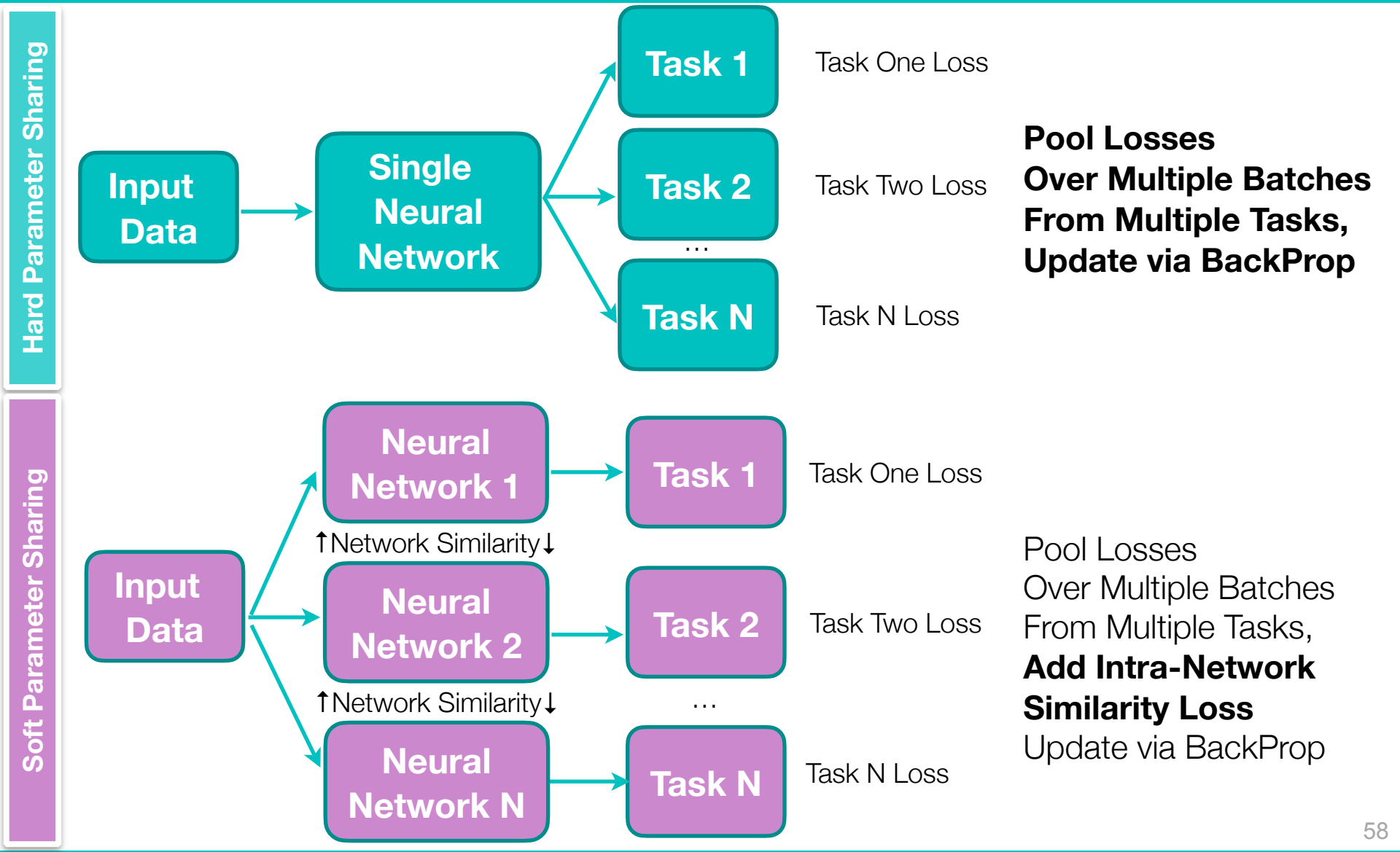
Multi-Task Models

Multi-task learning overview

- For deep networks, simple idea: share parameters in early layers
- Used shared parameters as feature extractors
- Train separate, unique layers for each task

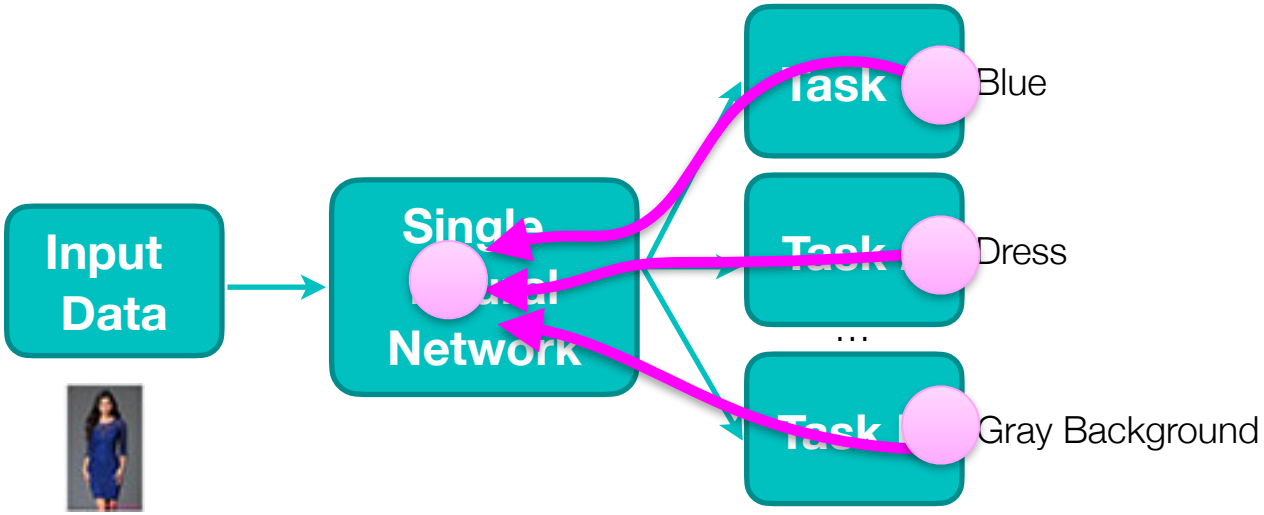


Multi-task Learning Parameter Sharing



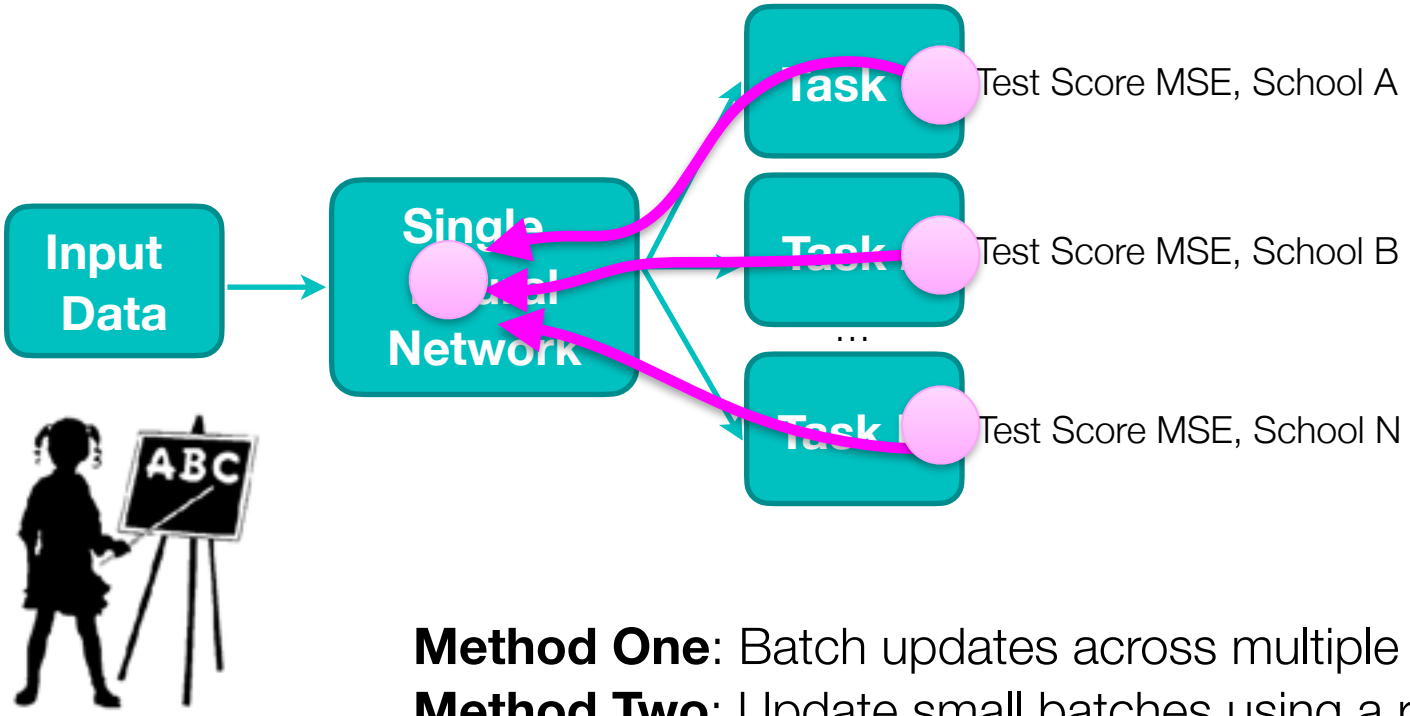
Multi-task Optimization

Multi-Label per Input



Multi-task Optimization

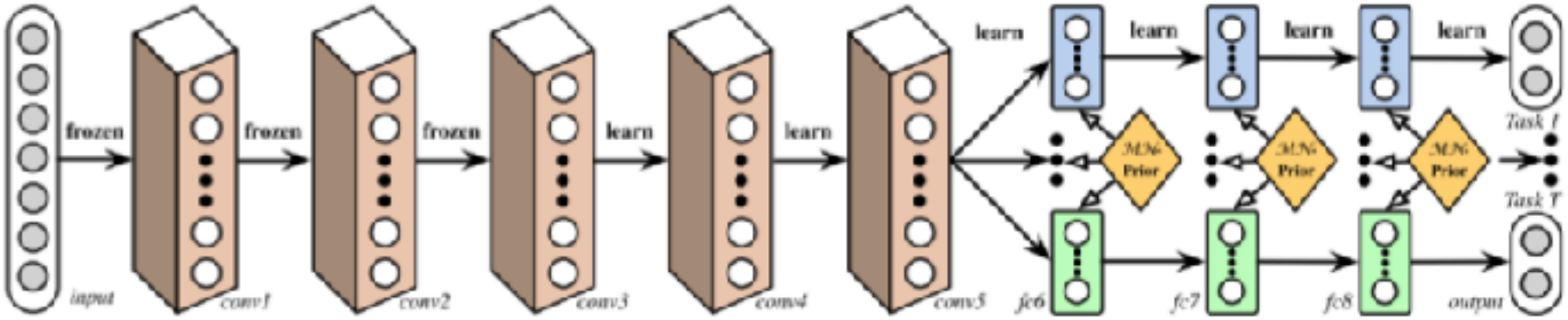
Single Task Label per Input



Method One: Batch updates across multiple tasks
Method Two: Update small batches using a random task

Multi-Task Model Examples

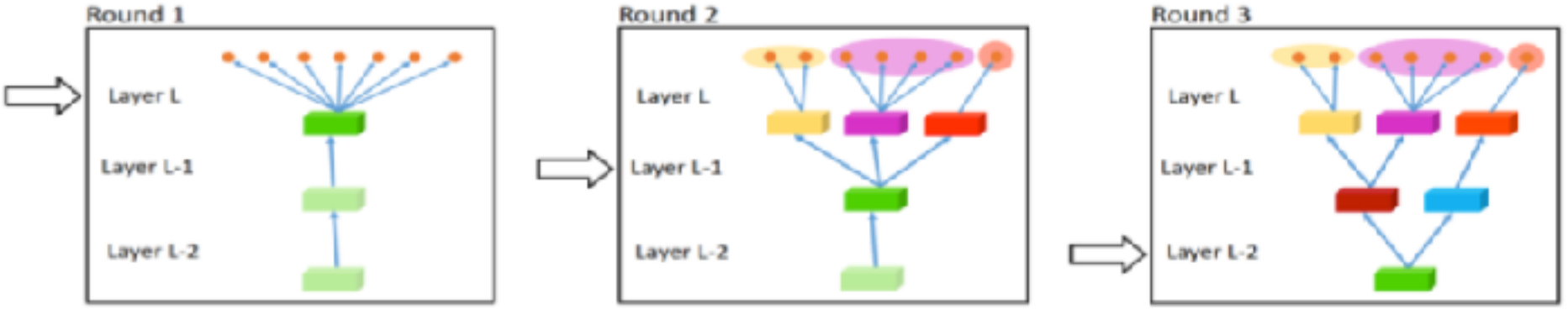
Multi-task: Deep Relationship Networks



- Start training traditionally
- Minimize Kroenecker Product between fully connected task specific layers
  - that is, make Grammian close to identity
  - encourages feature maps in each task to be less correlated to other task feature maps

https://arxiv.org/pdf/1506.02117.pdf

Multi-task: Adaptive Feature Sharing



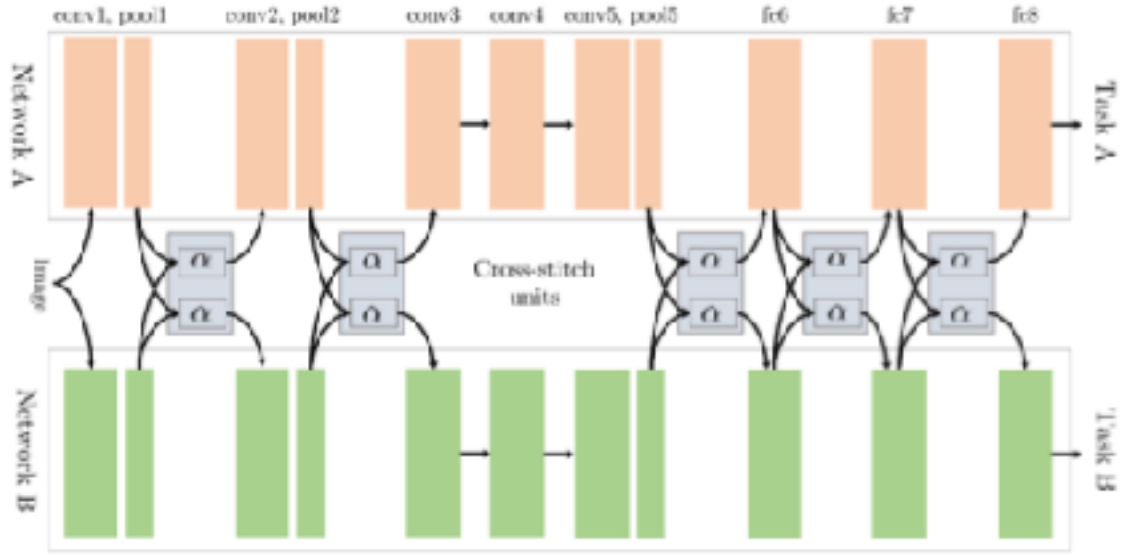
- Train
- Repe
  - D
  - Cut weights and fine tune network
  - Decrement current layer index

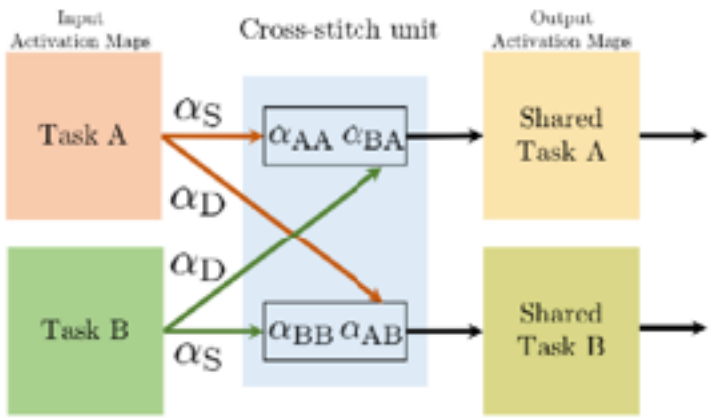
$$A^\star, \omega^\star(l) = \underset{A \in \mathbb{R}^{d \times d'}, |\omega| = d'}{\arg\min} ||W^{p,l} - AW^{p,l}_{\omega:}||_F, \qquad (2)$$

where $W^{p,l}_{\omega:}$ is a truncated weight matrix that only keeps the rows indexed by the set ω. This problem is NP-hard, however, there exist approaches based on convex relaxation

http://openaccess.thecvf.com/content_cvpr_2017/papers/Lu_Fully-Adaptive_Feature_Sharing_CVPR_2017_paper.pdf

Multi-task: Cross Stitch Networks

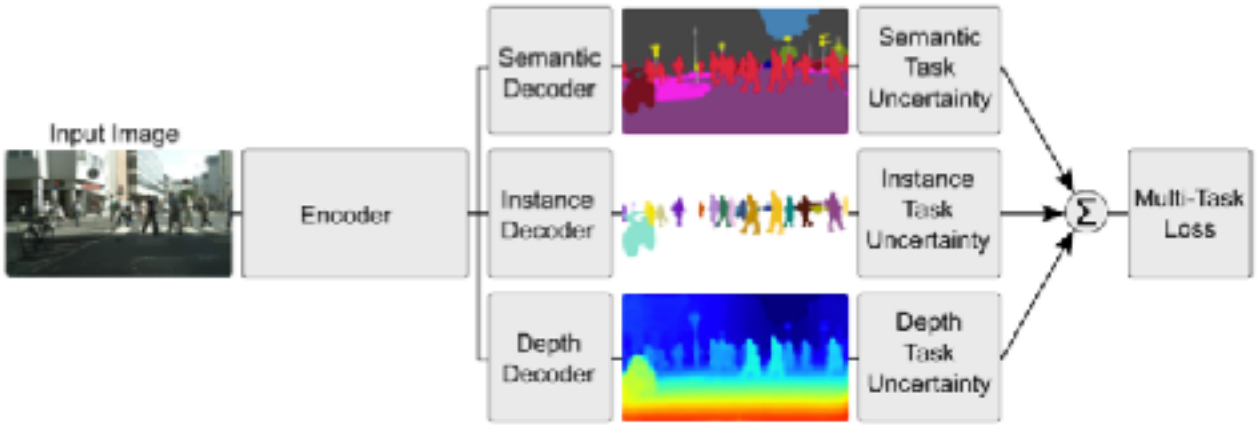




- Only works for simultaneous multi-label problems
  - like semantic segmentation and surface normal segmentation (clustering similarly facing objects)
- Take a learned weighted sum of the activations
- Works a little better than single task, but no worse

https://arxiv.org/pdf/1604.03539.pdf

Multi-task: Uncertainty Weighting



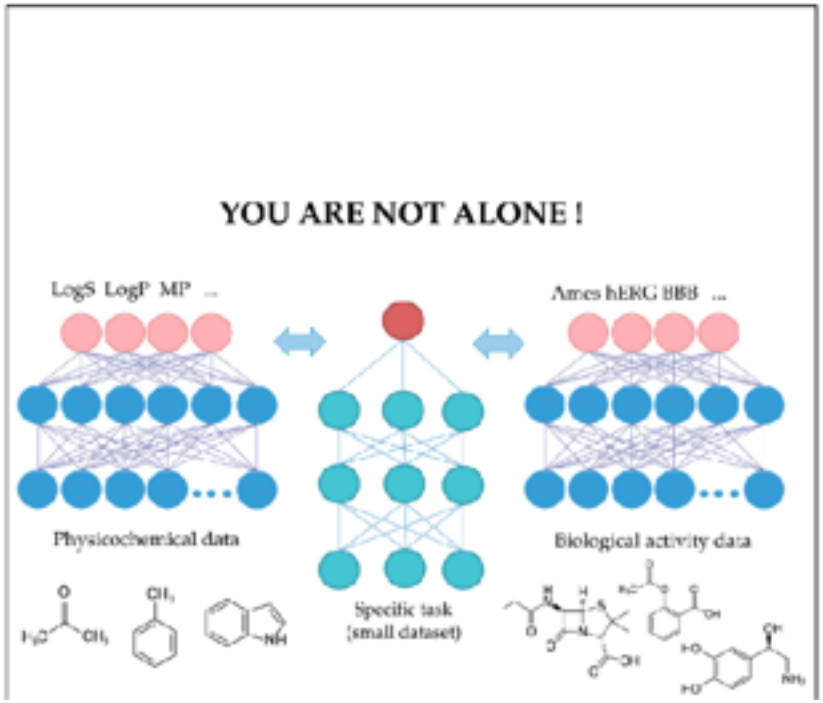
- Use variance of each loss function from each task to normalize
  - call it homoscedastic without sound reasoning because that feels better than “normalized variance”
  - talk about homoscedasticity for no reason
- Write an entire paper in a “mathy” way to make it seem like more of a contribution
- Profit because you are Oxford/Cambridge and reviewers give you a pass

https://arxiv.org/pdf/1705.07115.pdf

Paper Presentation: Multi-task with Chemical Fingerprints

A Survey of Multi-task Learning Methods in Chemoinformatics

Sergey Sosnin,[a] Mariia Vashurina,[a] Michael Withnall,[a] Pavel Karpov,[b] Maxim Fedorov,[a, c] and Igor V. Tetko*[a]



Next Time

- Multi-task demonstrations with various datasets
- Paper Presentations

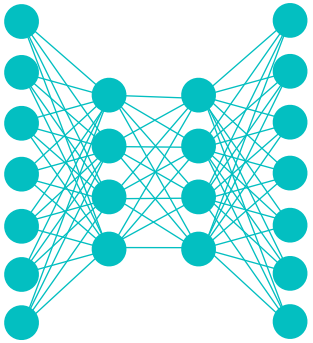
Lecture Notes for

Neural Networks and Machine Learning

Multi-Modal and Multi-Task

Next Time:
Demo

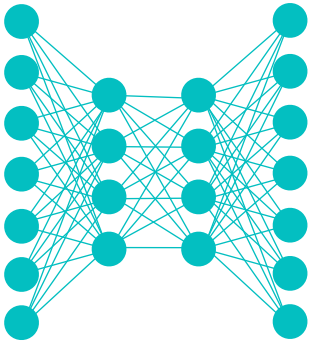
Reading: Papers



Lecture Notes for

Neural Networks and Machine Learning

Multi-Task Demo



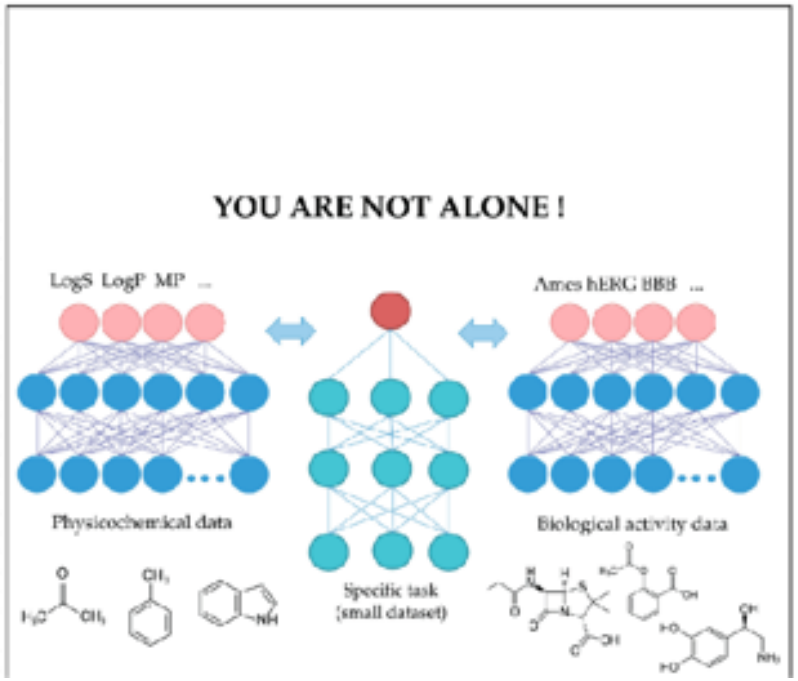
Logistics and Agenda

- Logistics
  - None!
- Agenda
  - Paper presentation
  - Multi-task Town Hall
  - Multi-Task demos
- Next Time
  - **No Class**: Read about GANs in Chollet
  - **After Spring Break**: Generative adversarial networks

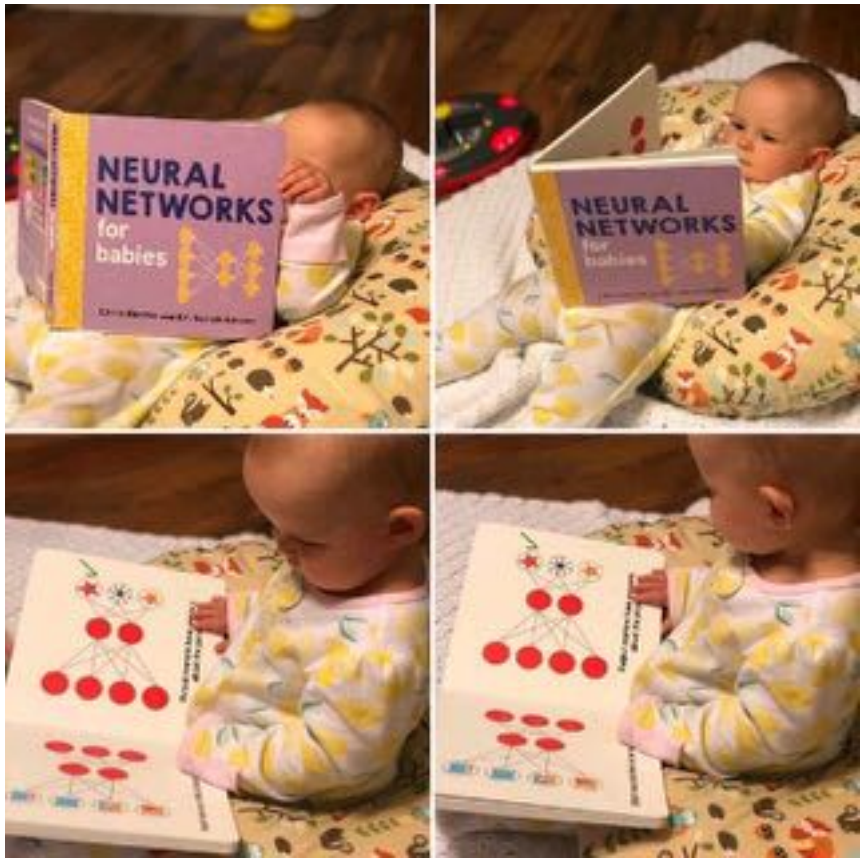
Paper Presentation: Multi-task with Chemical Fingerprints

A Survey of Multi-task Learning Methods in Chemoinformatics

Sergey Sosnin,[a] Mariia Vashurina,[a] Michael Withnall,[a] Pavel Karpov,[b] Maxim Fedorov,[a, c] and Igor V. Tetko*[a]



Town Hall



Multi-Task Learning in Keras with Multi-Label Data

Fashion week, colors and dresses

Follow Along: https://www.pyimagesearch.com/2018/06/04/keras-multiple-outputs-and-multiple-losses/

Multi-Task Learning

School Data, Computer Surveys, ChEMBL

Traian Pop

Luke Wood

`Follow Along: LectureNotesMaster/ LectureMultiTask.ipynb`

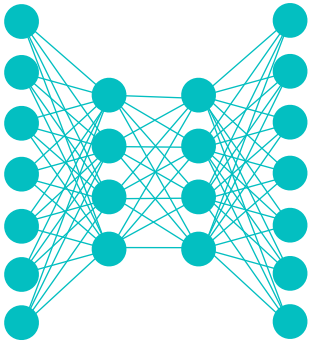
Lecture Notes for

Neural Networks and Machine Learning

Demo Multi-Task

Next Time:
GANs

Reading: Chollet 8.1-8.5



Backup slides

Title Between Topics

Example Slide

Title

Subtitle

`Follow Along: Notebook Name`