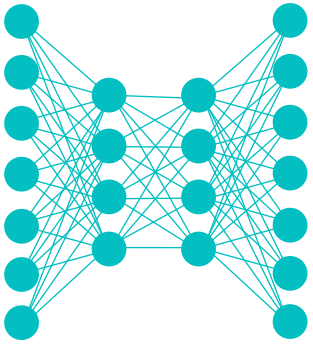
Lecture Notes for

Neural Networks and Machine Learning

Transfer Learning



Logistics and Agenda

- Logistics
  - Style Transfer Lab Due Soon
  - Need Volunteers for “Deep Multi-modal Learning” Paper Presentation to start Next Time
- Agenda
  - Transfer Learning Overview

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 between Source and 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

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

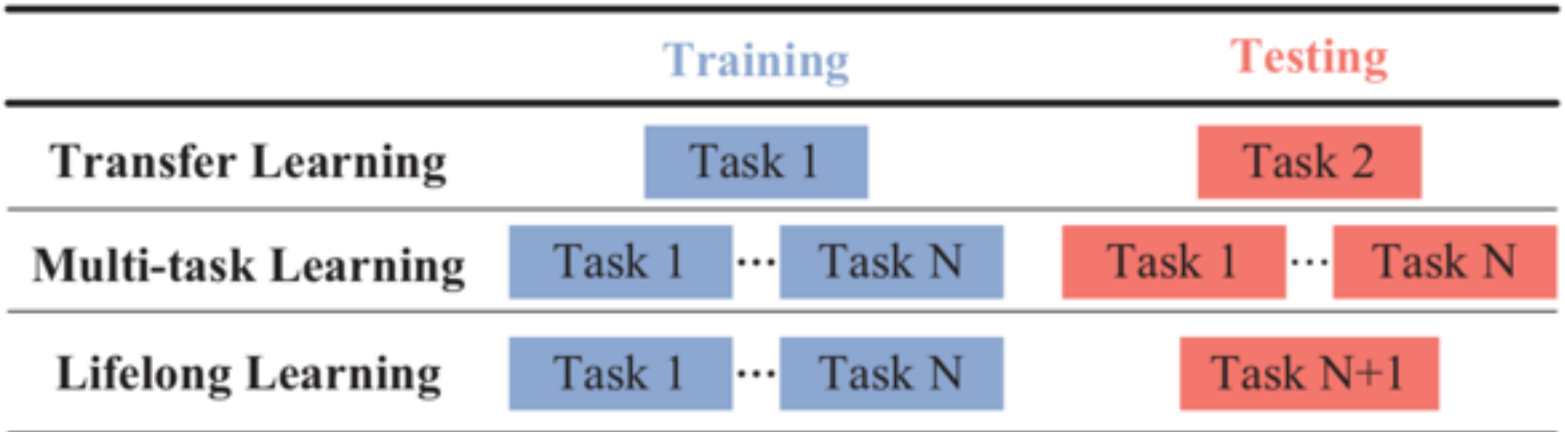
$$\mathcal{D} = \{\mathcal{X}, p(X)\} \qquad \mathcal{T} = \{\mathcal{Y}, p(Y|X)\}$$

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?

What to transfer (not neural networks yet)

- **Instance Transfer**
  - Use instances from source more often to do better at a task (*e.g.,* boosting if overlap in label spaces)
- **Feature Representation Transfer**
  - Good features in source are good in target
- **Parameter Transfer**
  - Tasks might share selection of hyper parameters
- **Relational Knowledge Transfer**
  - Two domains might share similar relationships between each other
  - For example, underlying graph structure relating instances to one another

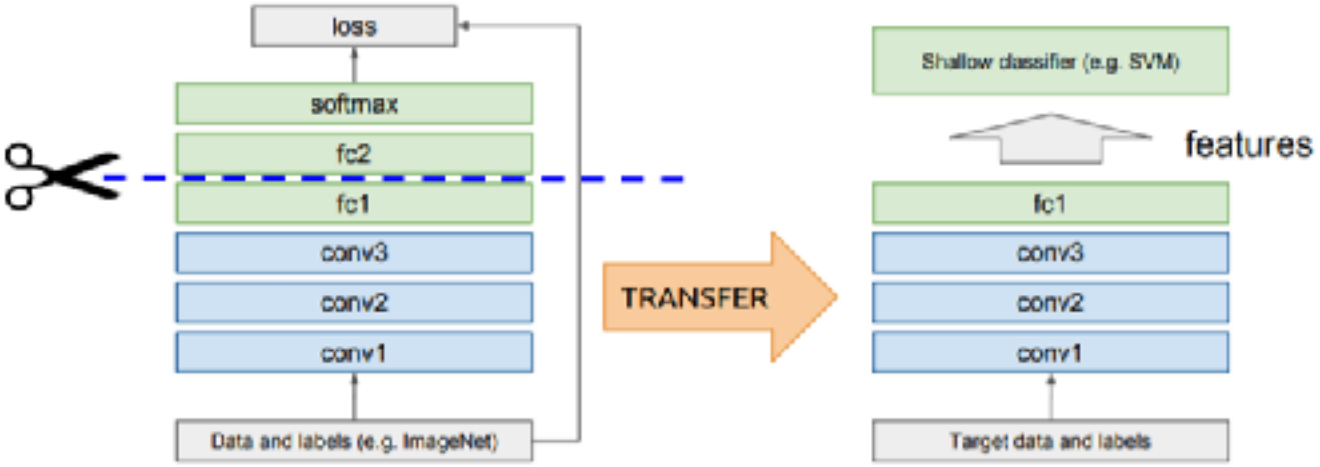
Transfer Learning with Neural Networks

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 a 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

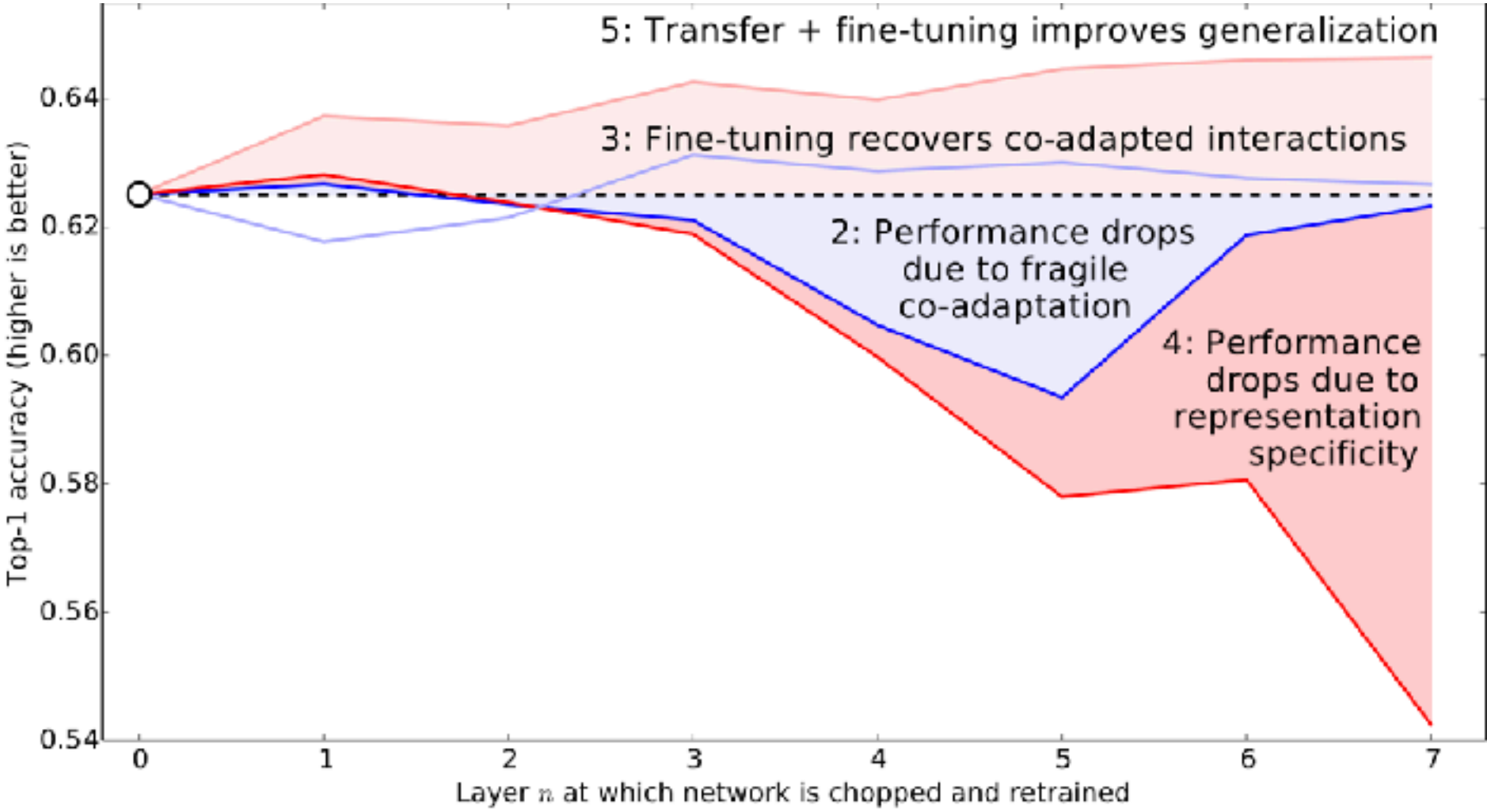
Bottlenecking on Maneframe

Dolphins versus Sharks

Justin Ledford •

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

Where to Cut?



Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?." In *Advances in neural information processing systems*, pp. 3320-3328. 2014.

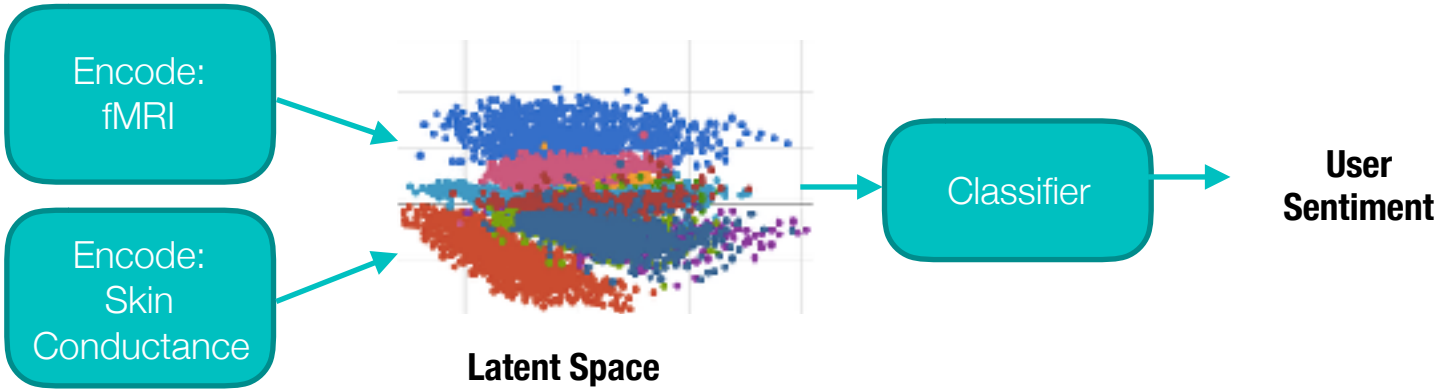
Popular Transfer Learning Models

- **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…

A Final Thought

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…



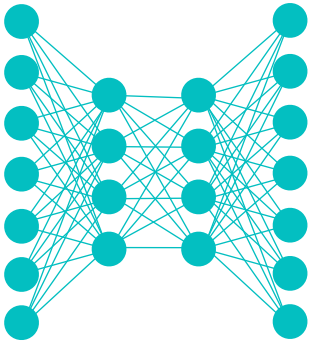
Lecture Notes for

Neural Networks and Machine Learning

Transfer Learning

Next Time:
Multi-Modal and Multi-Task

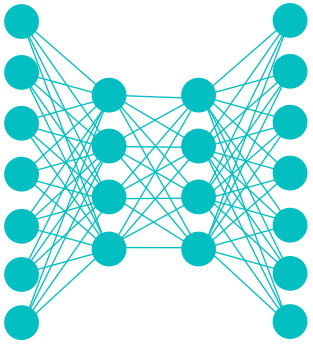
Reading: Keras F-API



Lecture Notes for

Neural Networks and Machine Learning

Multi-Modal and Multi-Task



Logistics and Agenda

- Logistics
  - Newest Lab uses multi-task and multi-modal learning
  - Need Volunteers for paper presentation:
    - *Sebastian Ruder, An Overview of Multi-Task Learning, 2017*
- Agenda
  - Paper presentation: Multi-modal
  - Multi-modal and multi-task learning
    - Techniques
    - Applications and domains

Paper Presentation: Deep Multi-Modal Learning

Dhanesh Ramachandram and
Graham W. Taylor

Deep Multimodal Learning

A survey on recent advances and trends

The success of deep learning has been a catalyst to solving increasingly complex machine-learning problems, which often involve multiple data modalities. We review recent advances in deep multimodal learning and highlight the state-of the art, as well as gaps and challenges in this active research field. We first classify deep multimodal learning architectures and then discuss methods to fuse learned multimodal representations in deep-learning architectures. We highlight two areas of research—regularization strategies and methods that learn or optimize multimodal fusion structures—as exciting areas for future work.

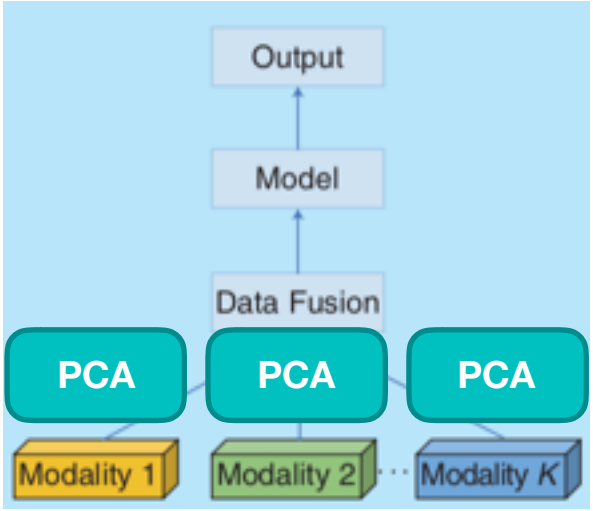
Multi-modal Review

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

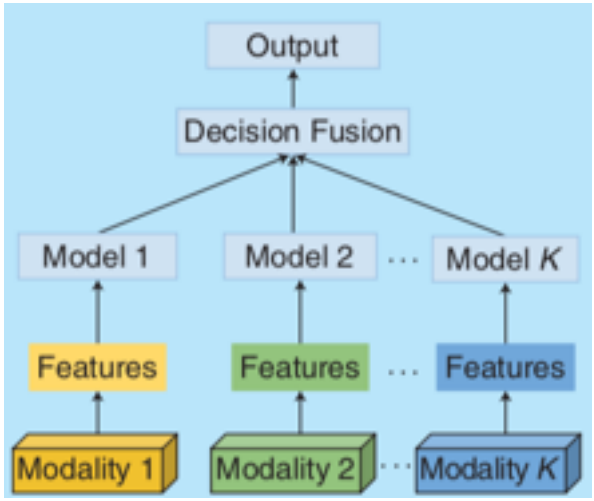
- Merge sensor layers early in the process
- **Assumption**: there is some data redundancy, but modes are conditionally independent
- **Problem**: architecture parameter explosion
  - One solution: dimensionality reduction or feature selection
  - Data Fusion



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

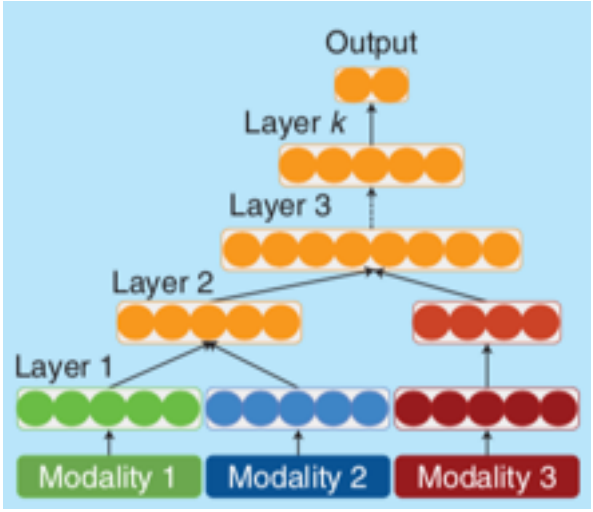


Ramamchandran and Taylor, 2017

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

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

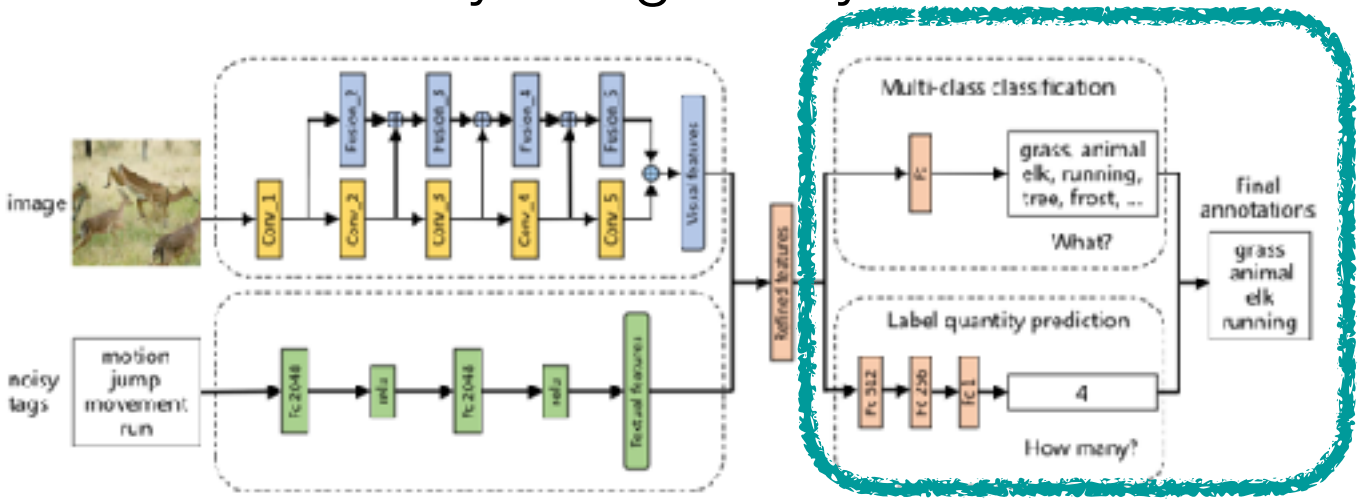
only weight correct class in the i modalities

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



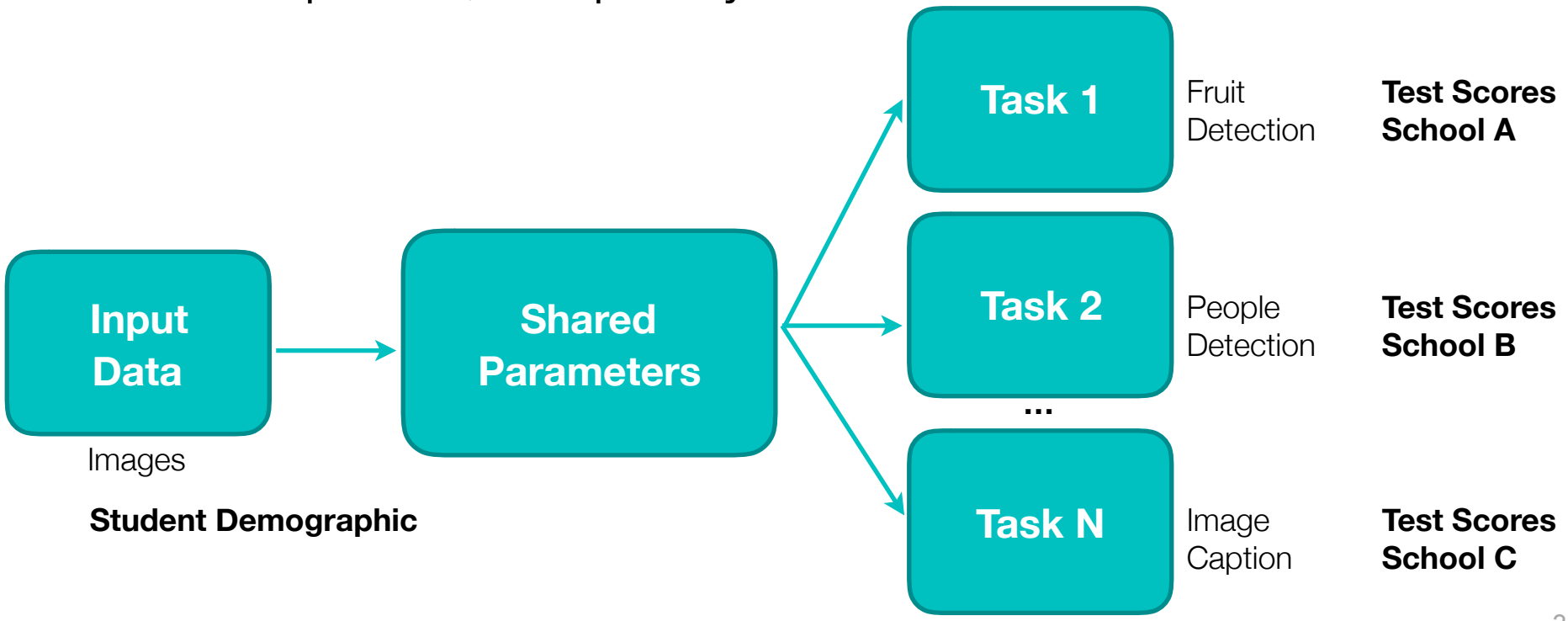
https://arxiv.org/pdf/1709.01220.pdf

Most current methods are still ad-hoc

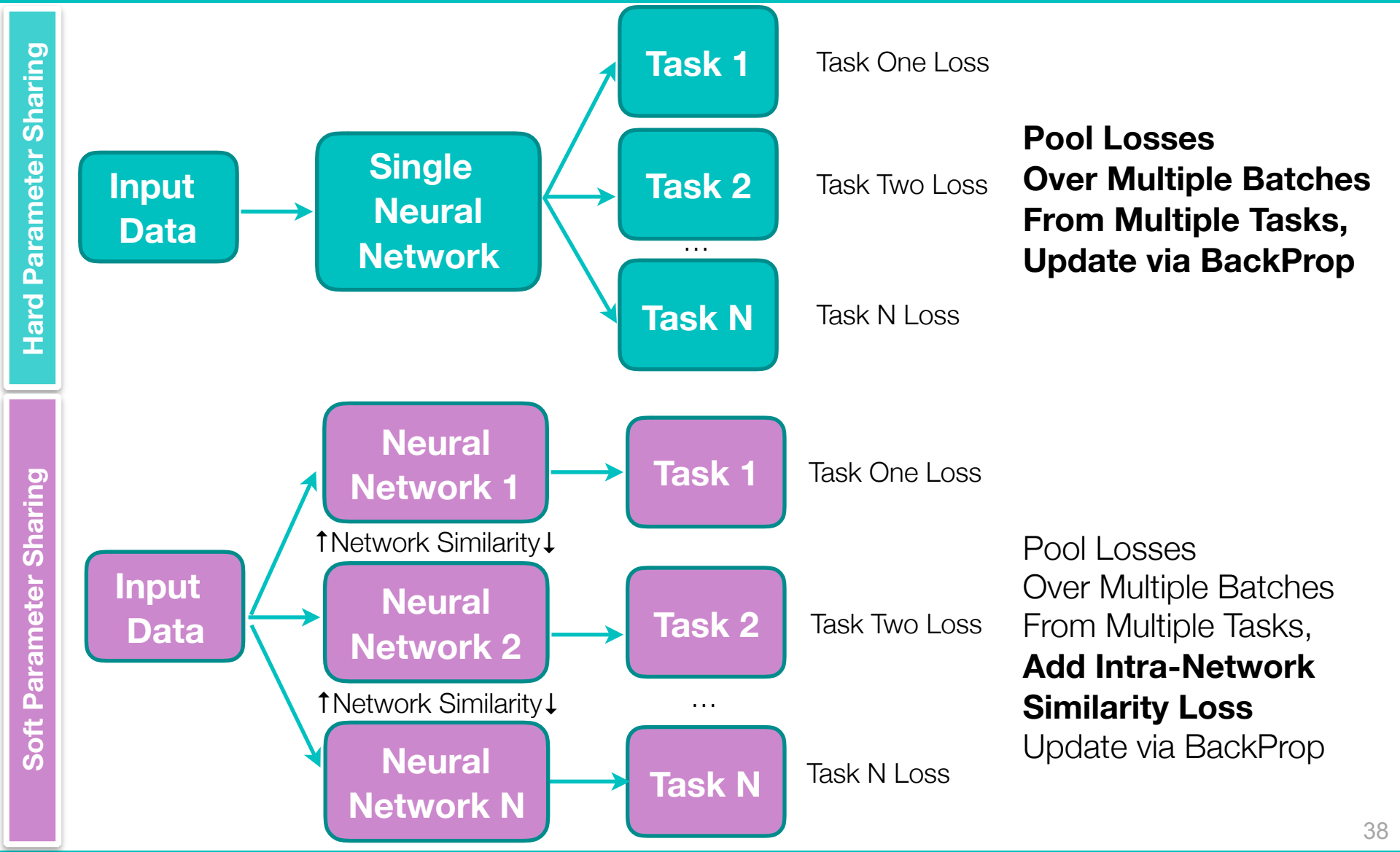
Multi-Task Models

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

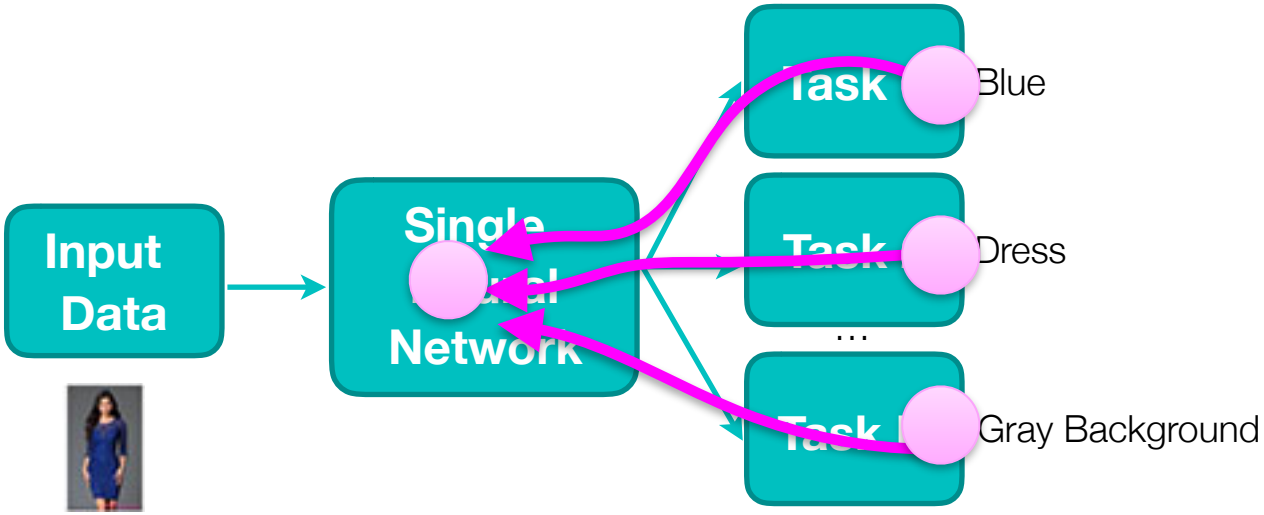


Multi-task Learning Parameter Sharing



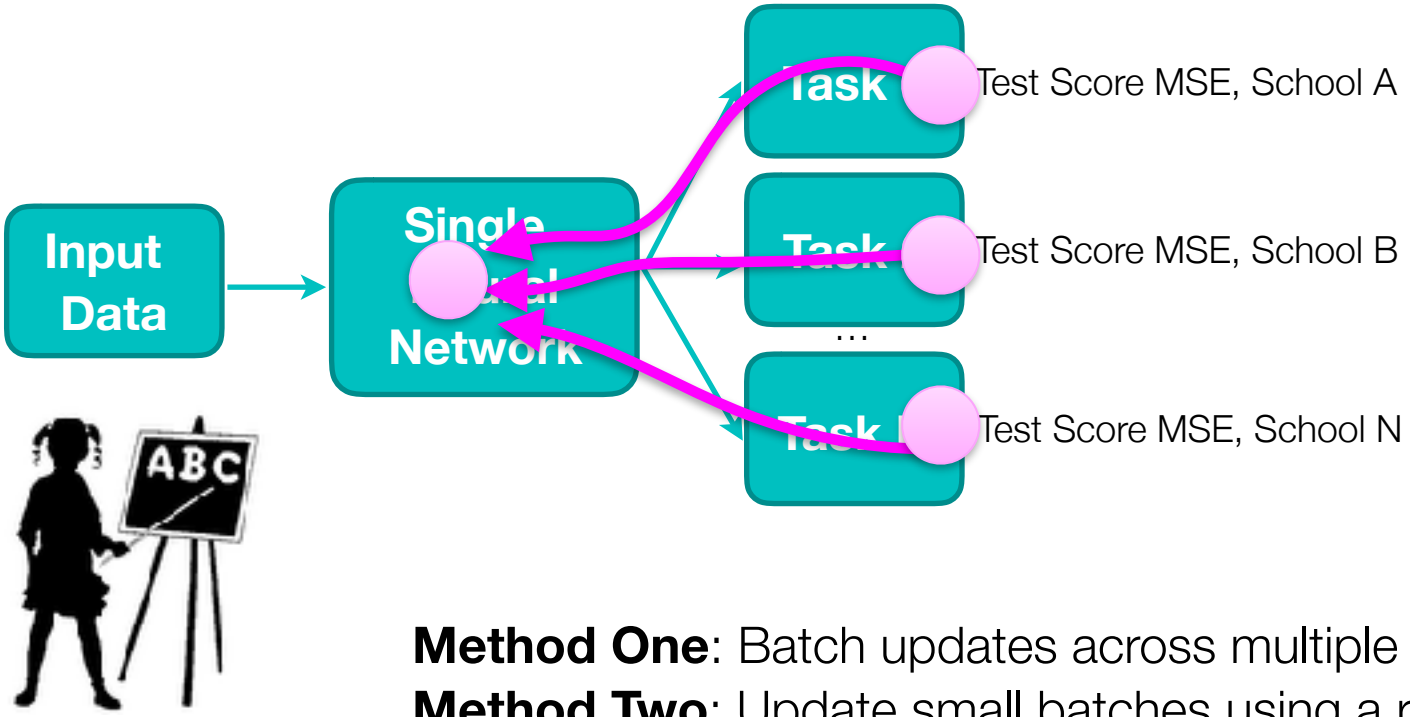
Multi-task Optimization

Multi-Label per Input



Multi-task Optimization

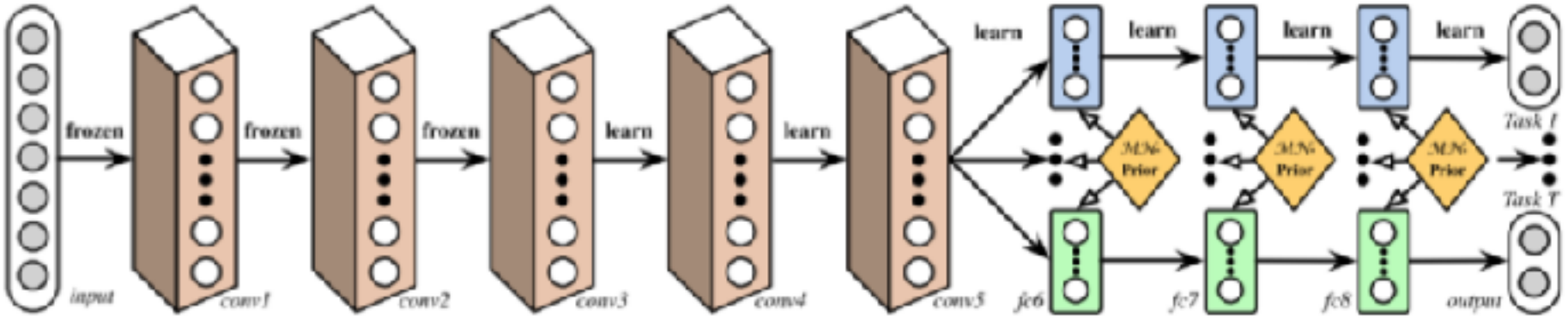
Single Task Label per Input



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

Multi-Task Model Examples

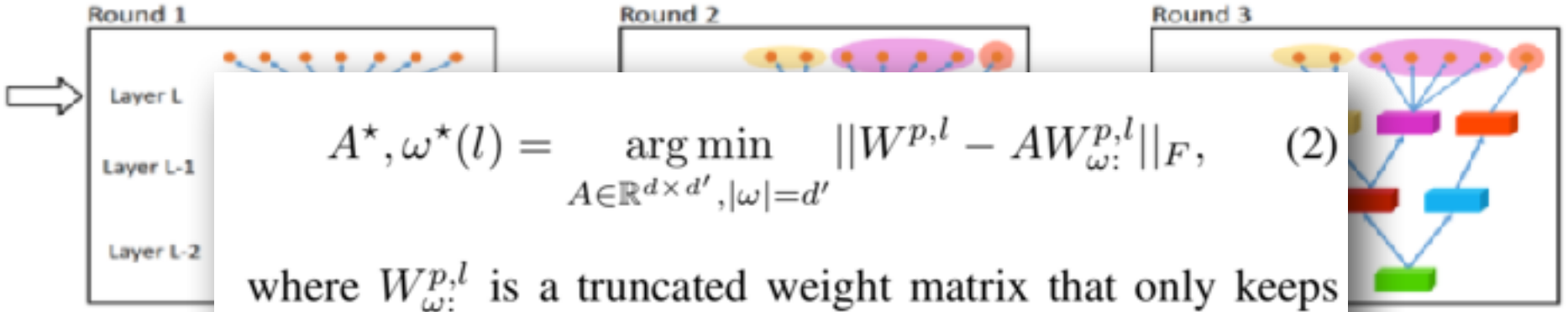
Multi-task: Deep Relationship Networks



- 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



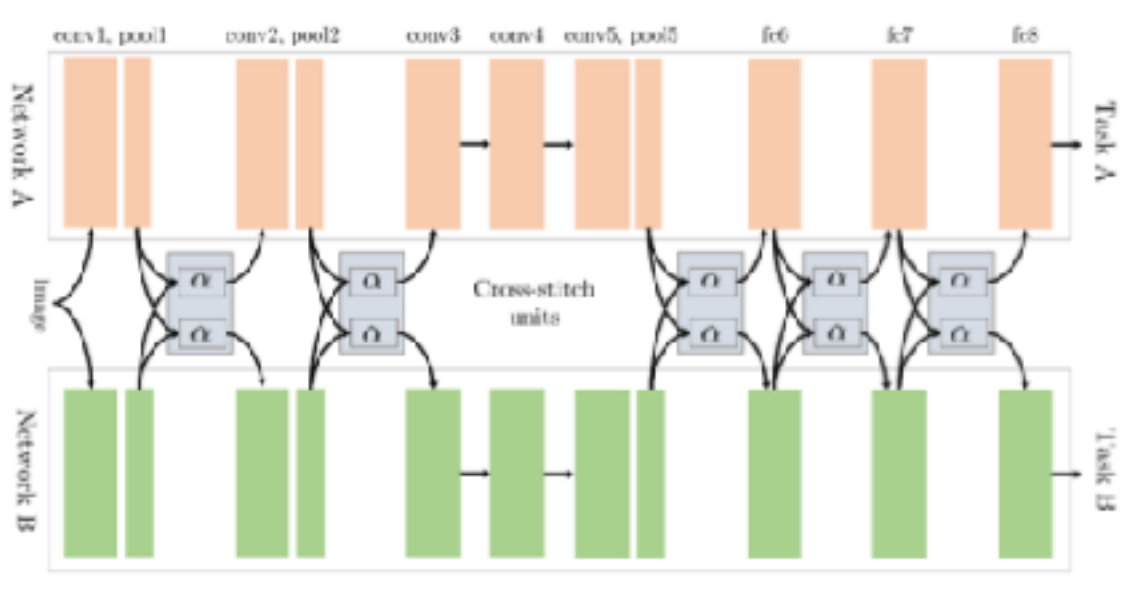
$$A^\star, \omega^\star(l) = \underset{A \in \mathbb{R}^{d \times d'}, |\omega| = d'}{\arg\min} ||W^{p,l} - AW^{p,l}_{\omega:}||_F, \quad (2)$$

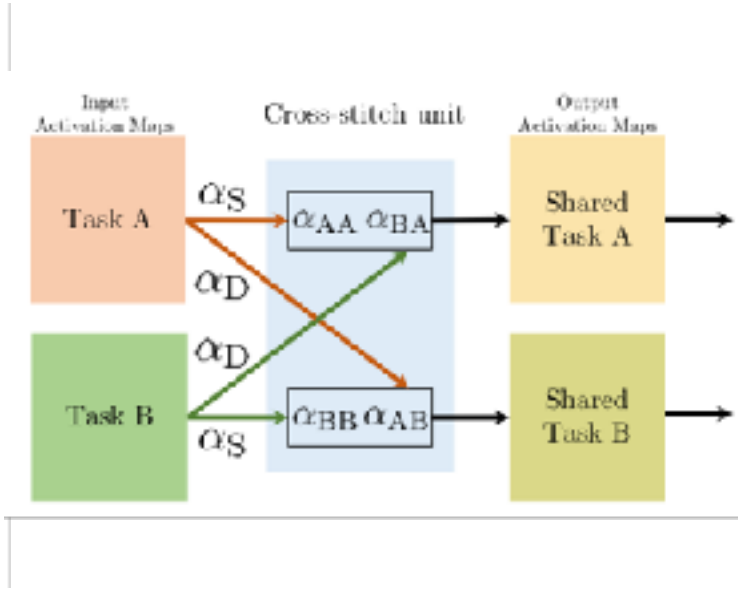
where $W^{p,l}_{\omega:}$ is a truncated weight matrix that only keeps the rows indexed by the set ω. This problem is NP-hard, however, there exist approaches based on convex relaxation

- Train
- Repeat (starting at final layer)
  - Divide layer rows by similarity
    - cluster affinity with task, if final layer
    - cluster affinity of branch if not final layer
  - Cut weights and fine tune network
  - Decrement current layer index

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

Multi-task: Cross Stitch Networks

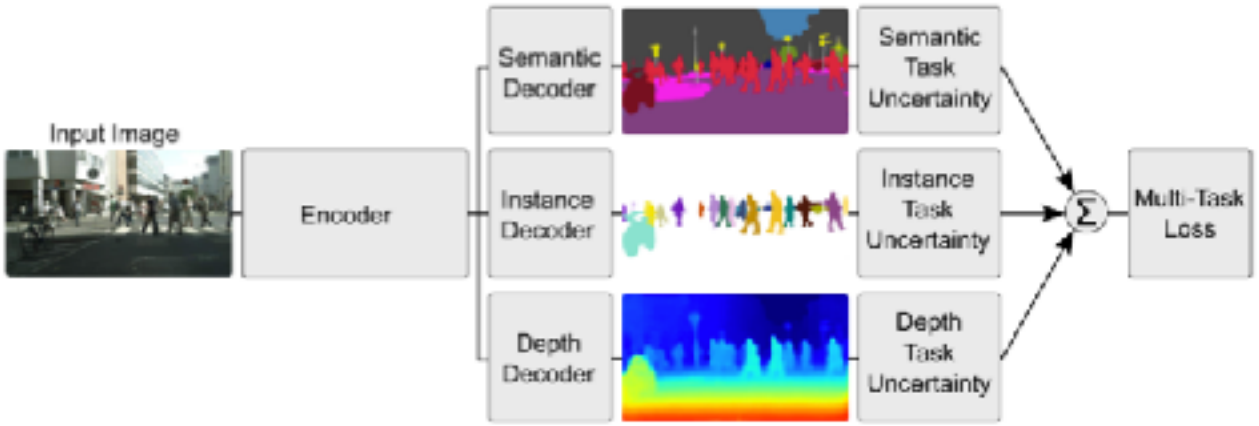




- Only works for simultaneous multi-label problems
  - like semantic segmentation and surface normal segmentation (clustering similarly facing objects)
- Take a learned weighted sum of the activations
- Works a little better than single task, but no worse

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

Next Time

- Multi-task demonstrations with various datasets
- Paper Presentations

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/

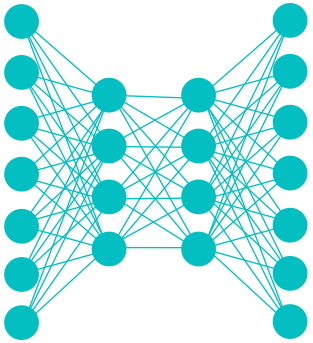
Lecture Notes for

Neural Networks and Machine Learning

Multi-Modal and Multi-Task

Next Time:
Demo

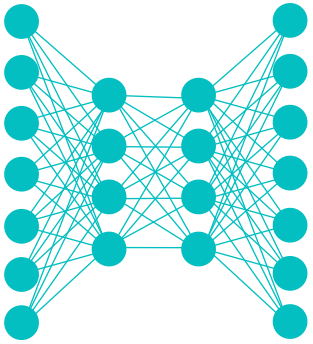
Reading: Papers



Lecture Notes for

Neural Networks and Machine Learning

Multi-Task Demo



Logistics and Agenda

- Logistics
  - None!
- Agenda
  - Paper presentation: Multi-modal
  - Paper presentation: Multi-task
  - Long Demo

Paper Presentation: Deep Multi-Modal Learning

Dhanesh Ramachandram and
Graham W. Taylor

Deep Multimodal Learning

A survey on recent advances and trends

The success of deep learning has been a catalyst to solving increasingly complex machine-learning problems, which often involve multiple data modalities. We review recent advances in deep multimodal learning and highlight the state-of the art, as well as gaps and challenges in this active research field. We first classify deep multimodal learning architectures and then discuss methods to fuse learned multimodal representations in deep-learning architectures. We highlight two areas of research—regularization strategies and methods that learn or optimize multimodal fusion structures—as exciting areas for future work.

Paper Presentation: Overview of Multi-Task Learning

An Overview of Multi-Task Learning in Deep Neural Networks*

Sebastian Ruder
Insight Centre for Data Analytics, NUI Galway
Aylien Ltd., Dublin
ruder.sebastian@gmail.com

Abstract

Multi-task learning (MTL) has led to successes in many applications of machine learning, from natural language processing and speech recognition to computer vision and drug discovery. This article aims to give a general overview of MTL, particularly in deep neural networks. It introduces the two most common methods for MTL in Deep Learning, gives an overview of the literature, and discusses recent advances. In particular, it seeks to help ML practitioners apply MTL by shedding light on how MTL works and providing guidelines for choosing appropriate auxiliary tasks.

Multi-Task Learning

School Data, Computer Surveys, ChEMBL

Traian Pop

Luke Wood

```
Follow Along: LectureNotesMaster/
LectureMultiTask.ipynb
```

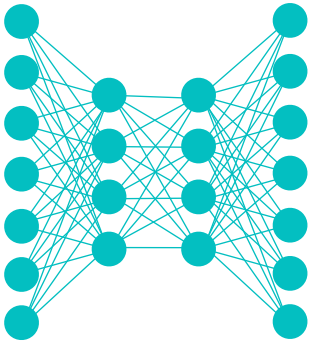

Lecture Notes for

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`