Lecture Notes for

Neural Networks and Machine Learning

Multi-Modal and Multi-Task
Multi-Task Demo

Logistics and Agenda

- Logistics
  - None!
- Agenda (Two lectures?)
  - Multi-modal
  - Paper Presentation: MTL Chemistry
  - Multi-task
  - Multi-Task Examples
  - Multi-Task Demos
  - Multi-Task Town Hall
- Next (Next?) Time
  - Circuits

Last Time

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

$$\mathcal{D}_{KL}(f\,||\,g) = -\sum f(x)\cdot\log\frac{g(x)}{f(x)}$$ definition of Kullback-Leibler (KL) Divergence

$$\mathcal{D}_{KL}(p(y\,|\,\mathbf{x})\,||\,p(y\,|\,\hat{\mathbf{x}})) = -\sum p(y\,|\,\mathbf{x})\cdot\log\frac{p(y\,|\,\hat{\mathbf{x}})}{p(y\,|\,\mathbf{x})} = -\sum p(y\,|\,\mathbf{x})\cdot\left(\log p(y\,|\,\hat{\mathbf{x}}) - \log p(y\,|\,\mathbf{x})\right)$$

$$= -\sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\hat{\mathbf{x}}) + \sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\mathbf{x})$$

$$= \mathbf{E}_{\mathbf{x}\in U,\hat{\mathbf{x}}\leftarrow q(\hat{\mathbf{x}}|\mathbf{x})}\left[-\log p(y\,|\,\hat{\mathbf{x}})\right] + \mathbf{E}_{\mathbf{x}\in U}\left[\cancel{\log p(y\,|\,\mathbf{x})}\right]_{\text{ignore}}$$

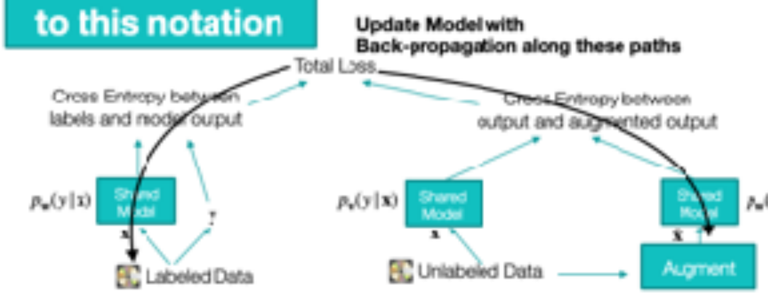
cross entropy of unsupervised labels after augmentation

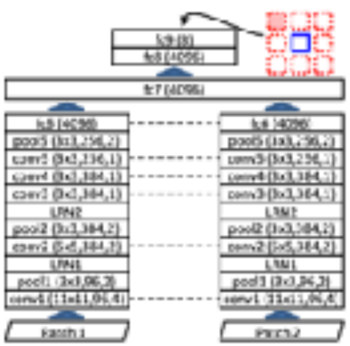
entropy of unsupervised labels ***constant***

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

Get accustomed to this notation

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





Unsupervised Visual Representation Learning by Context Prediction

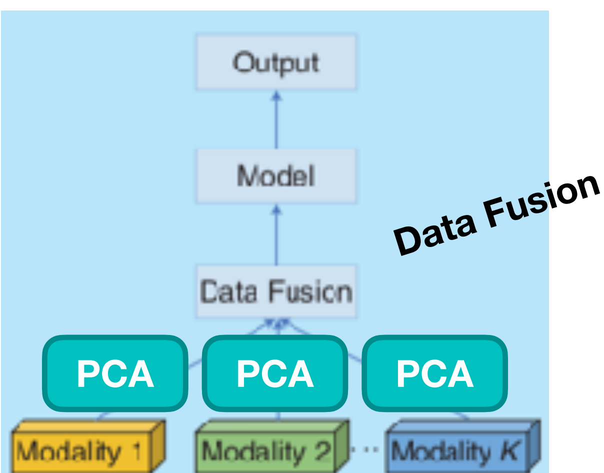
Multi-modal Review

Multi-modal == Multiple Data Sources

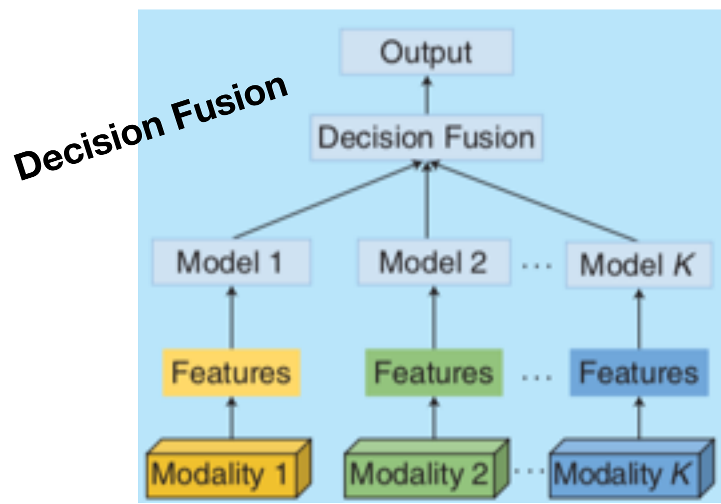
- **Modal** comes from the "sensor fusion" definition from Lahat, Adali, and Jutten (2015) for deep learning
- Using the Keras functional API, this is extremely easy to implement
  - … and we have used it since CS7324!
- But now let's take a deeper dive and ask:
  - What are the different types of modalities that we might try?
  - Is there a more optimal way to merge information?
  - When? Early, Intermediate, and late fusion

Early and Late Stage 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
  - Need dimensionality reduction



- **Late Fusion**: Merge sensor layers right before flattening
- Use Decision Fusion on outputs
- **Assumption**: little redundancy or conditional independence—just an ensemble architecture
- **Problem**: just separate classifiers, limited interplay

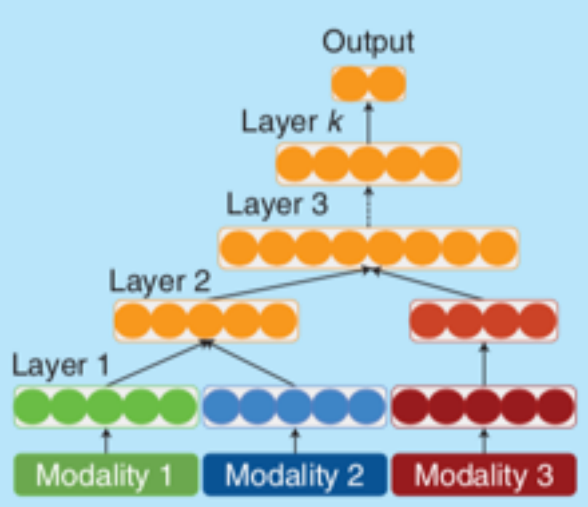


Ramamchandran and Taylor, 2017

Intermediate Fusion

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

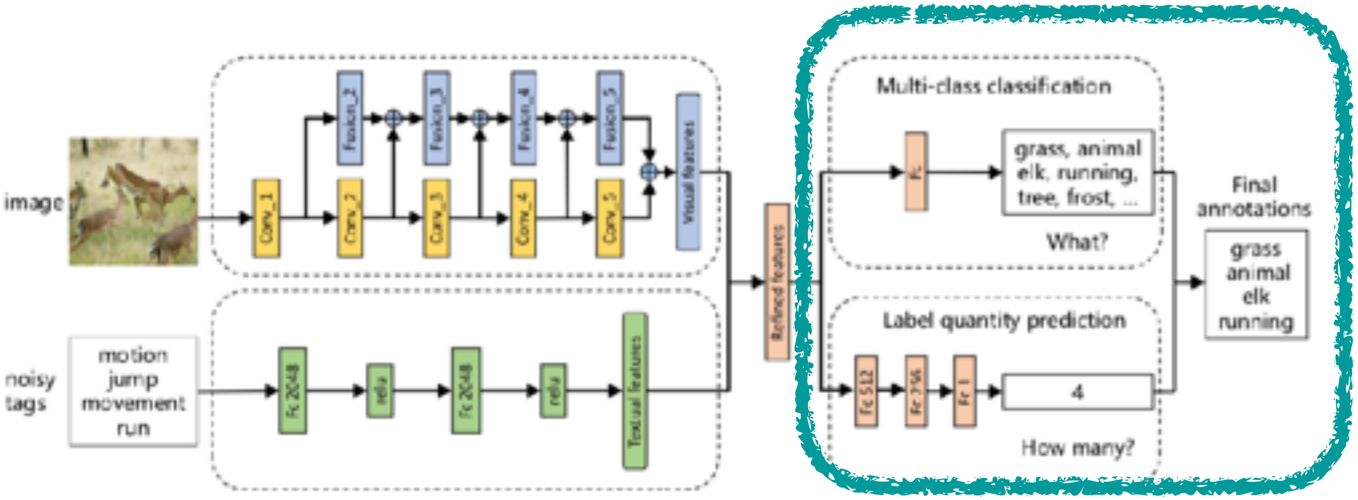
1. Stacked Auto-Encoders
   [Ding and Tao, 2015]
2. Early fuse layers that are correlated
   [Neverova *et al.* 2016]
3. Fully train each modality merge based on criterion of similarity in activations
   [Lu and Xu 2018]



Ramamchandran and Taylor, 2017

Multi-modal Merging

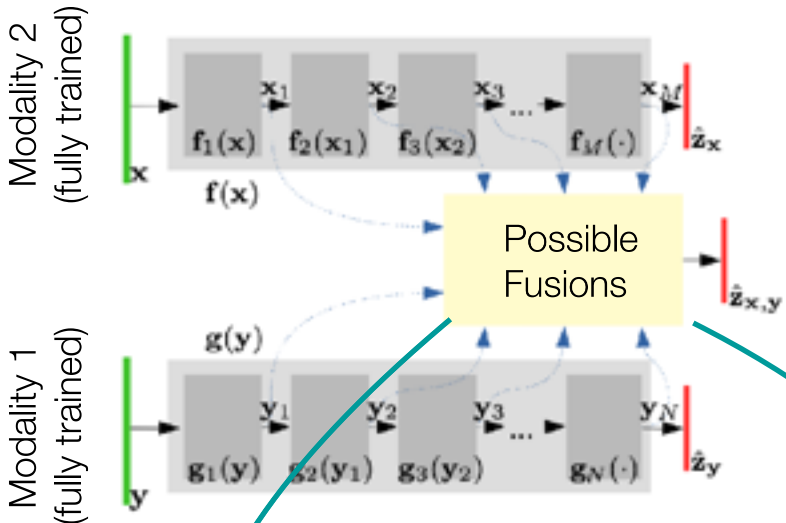
- **Still an open research problem**
- How to develop merging techniques that
  - Can handle exponentially many pairs of modalities
  - Automatically merge meaningful modes
  - Discard poor pairings
  - Selectively merge early or late (or dynamically)



Most current methods are still ad-hoc

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

Neural Architecture Search for Mode Fusion

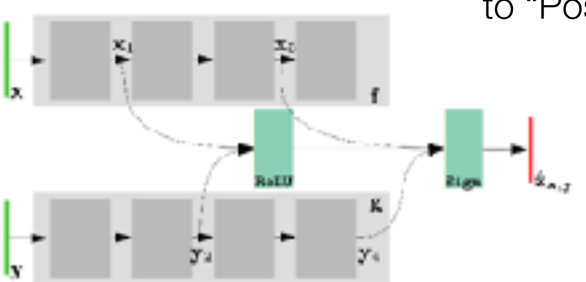


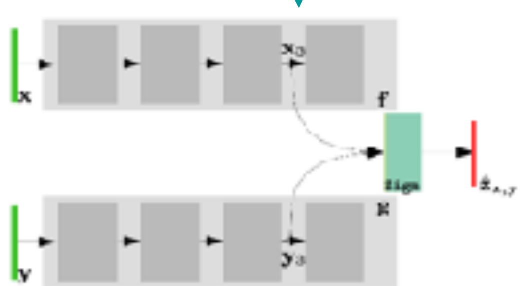
Genetic Algorithm

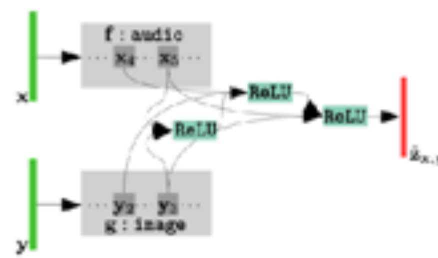
1. Sample new candidates
2. Evaluate fitnesses
3. Mutate and Crossover
4. Keep the best solutions
5. Repeat

Very computational when starting, because candidates are all untrained.
However, as more blocks start from "mostly trained" positions, training time reduces.

1. Sample blocks at Random
2. Train by fine tuning
3. Evaluate fitness
4. Add trained blocks back to "Possible Fusions"





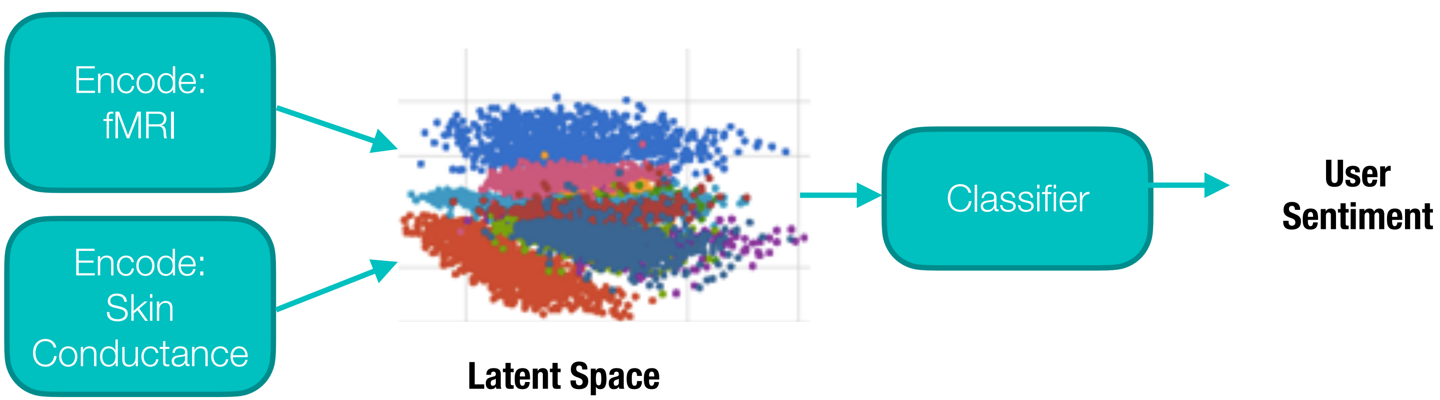


Found solution for AV-MNIST

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 unifying 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…**



Paper Presentation: DeepTox

DeepTox: Toxicity Prediction using Deep Learning

Andreas Mayr[1,2], Günter Klambauer[1], Thomas Unterthiner[1,2] and Sepp Hochreiter[1]

[1]Institute of Bioinformatics, Johannes Kepler University Linz, Linz, Austria
[2]RISC Software GmbH, Johannes Kepler University Linz, Hagenberg, Austria

The Tox21 Data Challenge has been the largest effort of the scientific community to compare computational methods for toxicity prediction. This challenge comprised 12,000 environmental chemicals and drugs which were measured for 12 different toxic effects by specifically designed assays. We participated in this challenge to assess the performance of Deep Learning in computational toxicity prediction. Deep Learning has already revolutionized image processing, speech recognition, and language understanding but has not yet been applied to computational toxicity. Deep Learning is founded on novel algorithms and architectures for artificial neural networks together with the recent availability of very fast computers and massive datasets. It discovers multiple levels of distributed representations of the input, with higher levels representing more abstract concepts. We hypothesized that the construction of a hierarchy of chemical features gives Deep Learning the edge over other toxicity prediction methods. Furthermore, Deep Learning naturally enables multi-task learning, that is, learning of all toxic effects in one neural network and thereby learning of highly informative chemical features. In order to utilize Deep Learning for toxicity prediction, we have developed the DeepTox pipeline. First, DeepTox normalizes the chemical representations of the compounds. Then it computes a large number of chemical descriptors that are used as input to machine learning methods. In its next step, DeepTox trains models, evaluates them, and combines the best of them to ensembles. Finally, DeepTox predicts the toxicity of new compounds. In the Tox21 Data Challenge, DeepTox had the highest performance of all computational methods winning the grand challenge, the nuclear receptor panel, the stress response panel, and six single assays (teams "Bioinf@JKU"). We found that Deep Learning excelled in toxicity prediction and outperformed many other computational approaches like naive Bayes, support vector machines, and random forests.

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