Deep Learning
MSDS 631

Multi-Task, Data Augmentation, Transfer Learning

Michael Ruddy

Questions?

- From last lecture?

Overview

- Small Datasets
- Transfer Learning
- Data Augmentation
- Multi-Task Learning

Small Datasets

- Not enough data
  - Expensive to obtain
  - Too big to store

Small Datasets

- Not enough data
    - Expensive to obtain
    - Too big to store
- Not enough labels
    - Expensive to obtain
    - Nonstandard task

Small Datasets

- Not enough data
  - Expensive to obtain
  - Too big to store
- Not enough labels
  - Expensive to obtain
  - Nonstandard task
- Transfer Learning
  - Use features learned on larger datasets for your task
  - Example: Word Embeddings

Small Datasets

- Not enough data
    - Expensive to obtain
    - Too big to store
- Not enough labels
    - Expensive to obtain
    - Nonstandard task
- Transfer Learning
    - Use features learned on larger datasets for your task
    - Example: Word Embeddings
- Data Augmentation
    - Use existing data to create synthetic data

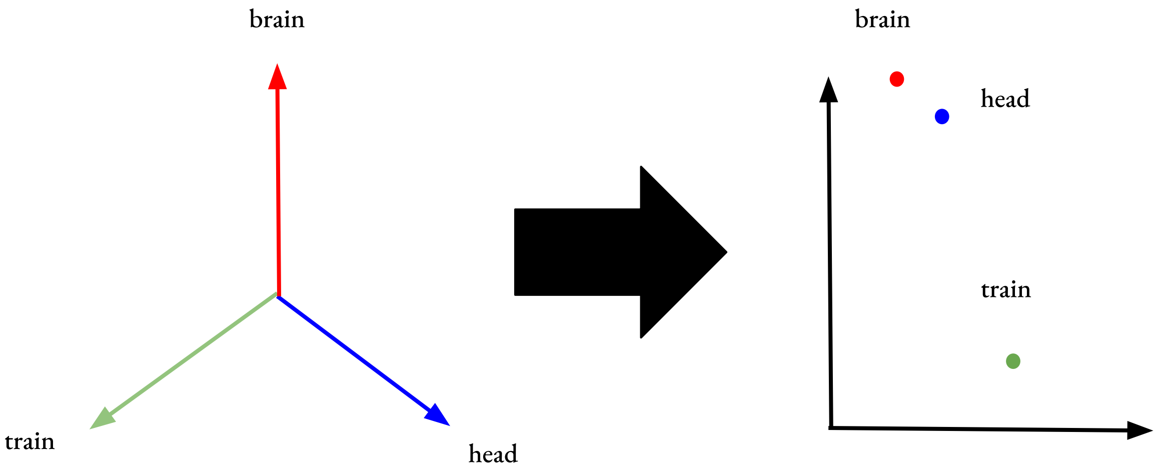
Small Datasets

- Not enough data
  - Expensive to obtain
  - Too big to store
- Not enough labels
  - Expensive to obtain
  - Nonstandard task
- Transfer Learning
  - Use features learned on larger datasets for your task
  - Example: Word Embeddings
- Data Augmentation
  - Use existing data to create synthetic data
- Multi-Task Learning

Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
    - Learn from context (lots of data)
    - Use for all sort of NLP tasks (maybe less data/labels)

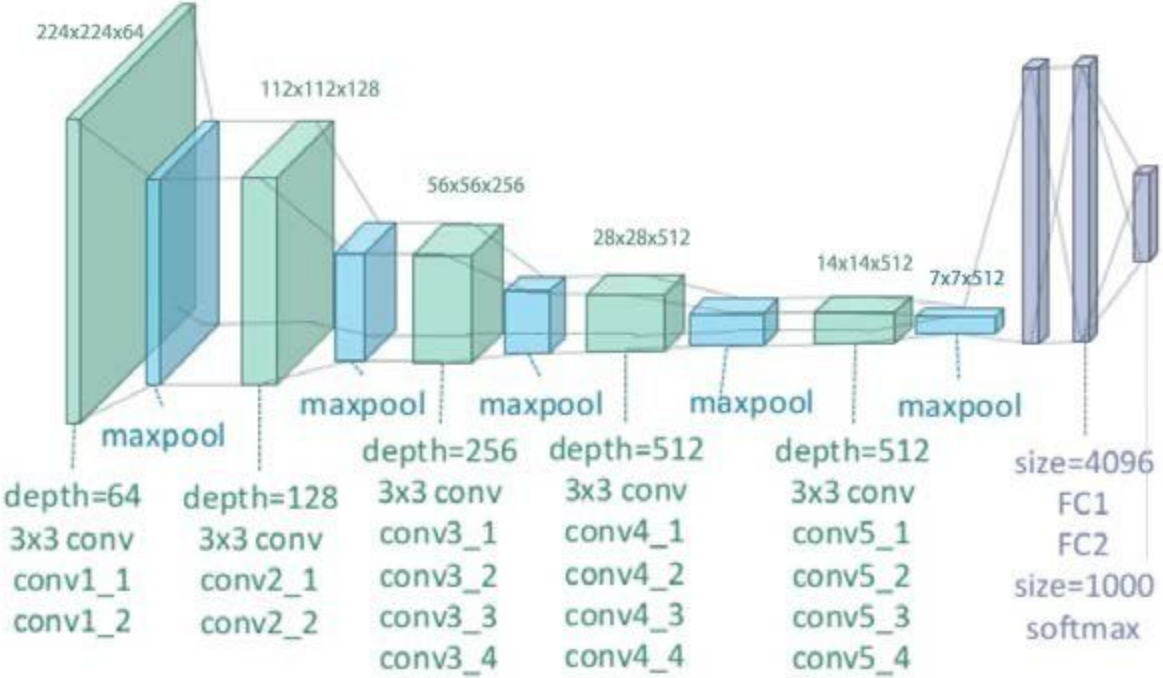
$$\begin{array}{ccc} \begin{bmatrix}1\\0\\0\end{bmatrix} & \begin{bmatrix}0\\1\\0\end{bmatrix} & \begin{bmatrix}0\\0\\1\end{bmatrix} \\ \text{brain} & \text{head} & \text{train} \end{array}$$



Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
  - Lots of images!

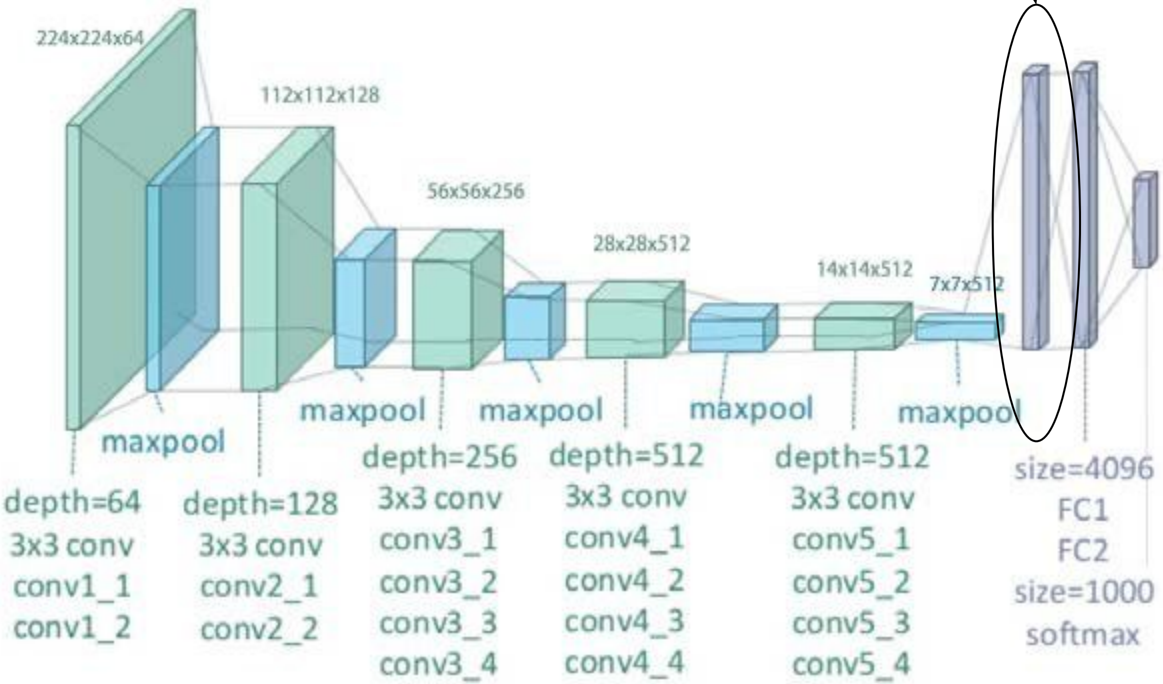
VGG-19



Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
  - Lots of images!

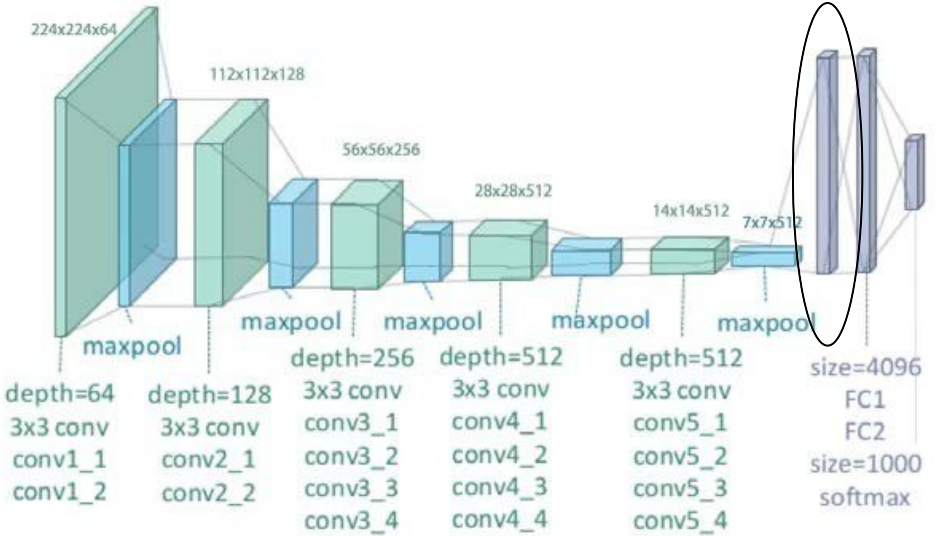
VGG-19



Embedding or Representation of an image

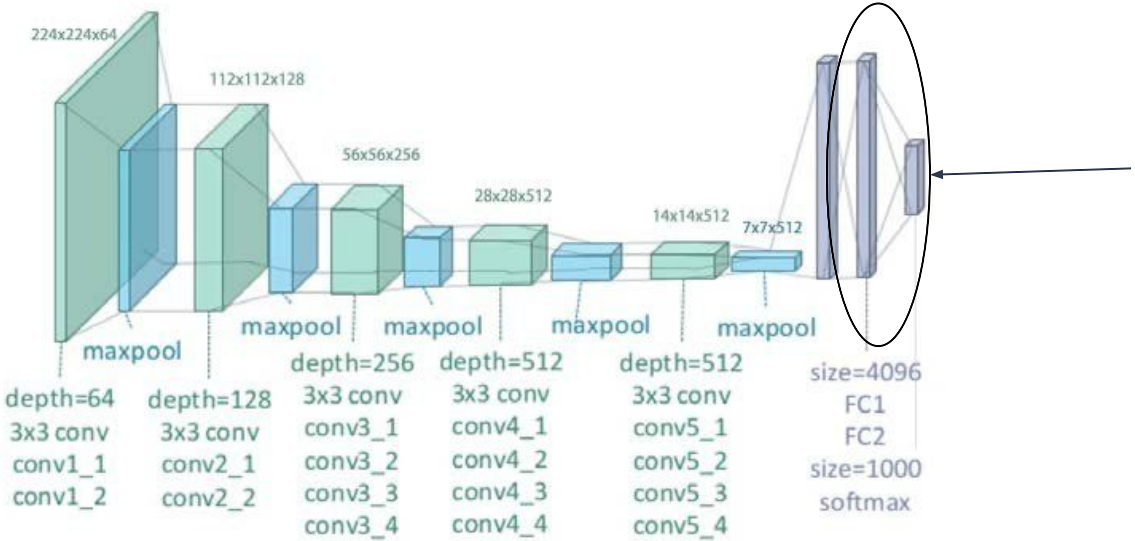
Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
  - Freeze your embedding/representation



Transfer Learning

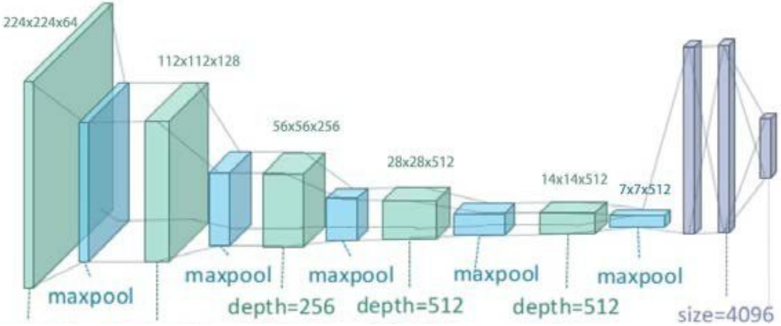
- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
    - Freeze your embedding/representation



Train just these layers

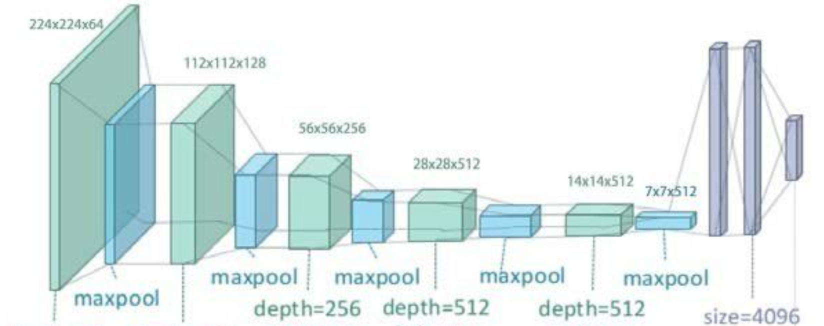
Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
  - Freeze your embedding/representation
  - Use pre-trained model as a starting point (fancy weight initialization)



Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
    - Freeze your embedding/representation
    - Use pre-trained model as a starting point (fancy weight initialization)

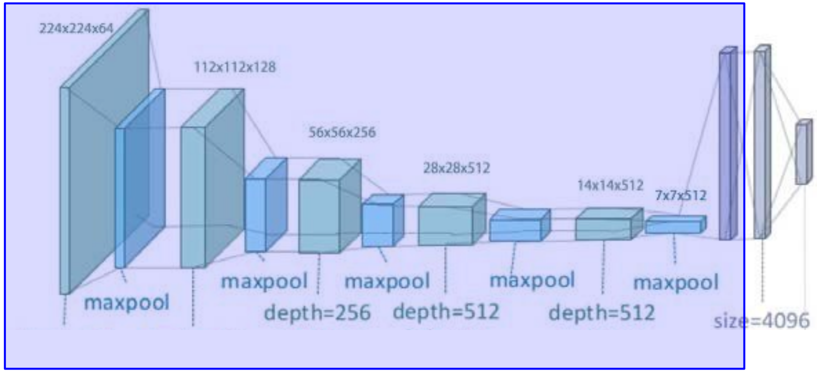


Low LR ↔ High LR

Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
  - Freeze your embedding/representation
  - Use pre-trained model as a starting point (fancy weight initialization)

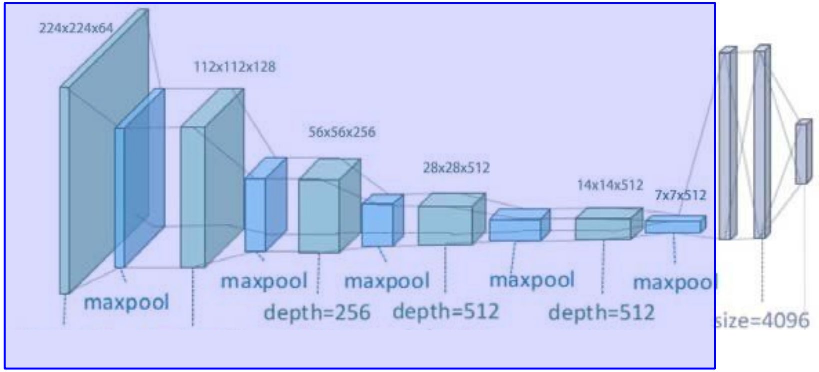
Epoch 0



Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
  - Freeze your embedding/representation
  - Use pre-trained model as a starting point (fancy weight initialization)

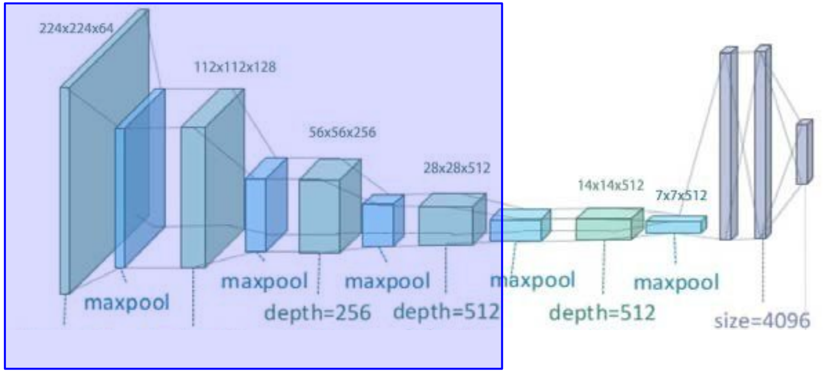
Epoch 1



Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
  - Freeze your embedding/representation
  - Use pre-trained model as a starting point (fancy weight initialization)

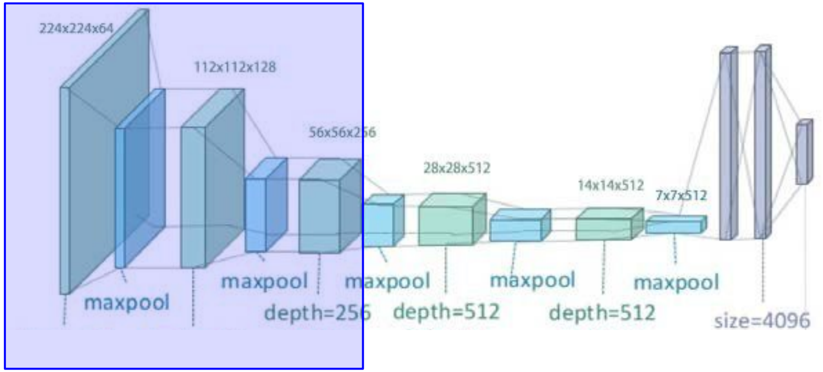
Epoch 2



Transfer Learning

- Transfer features from one task/dataset to another
- Word Embeddings
- Train a CNN on ImageNet: multi-class problem with 1000 classes
- Two approaches
  - Freeze your embedding/representation
  - Use pre-trained model as a starting point (fancy weight initialization)

Epoch 3



Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data

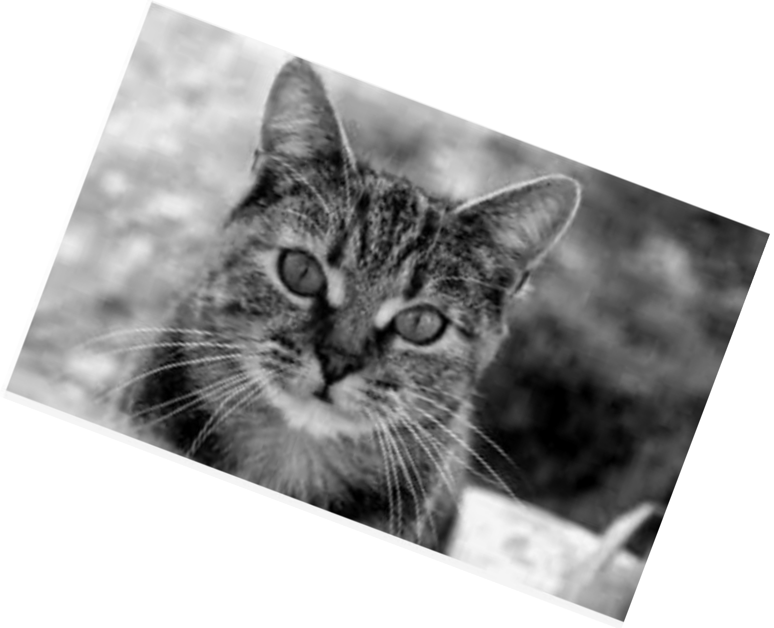
Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data

This is a cat

Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data



This is still a cat...

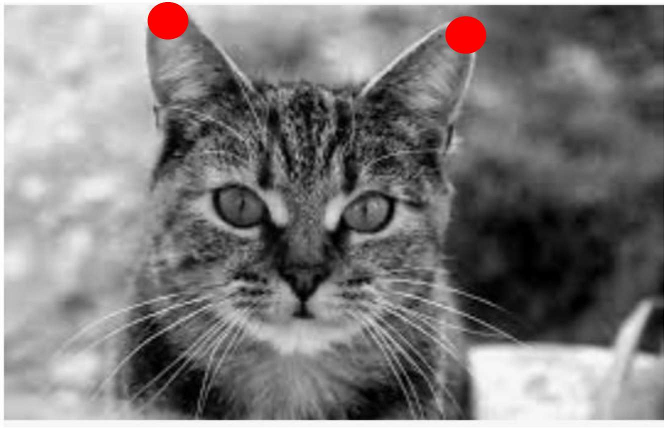
Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data

This is still a cat...

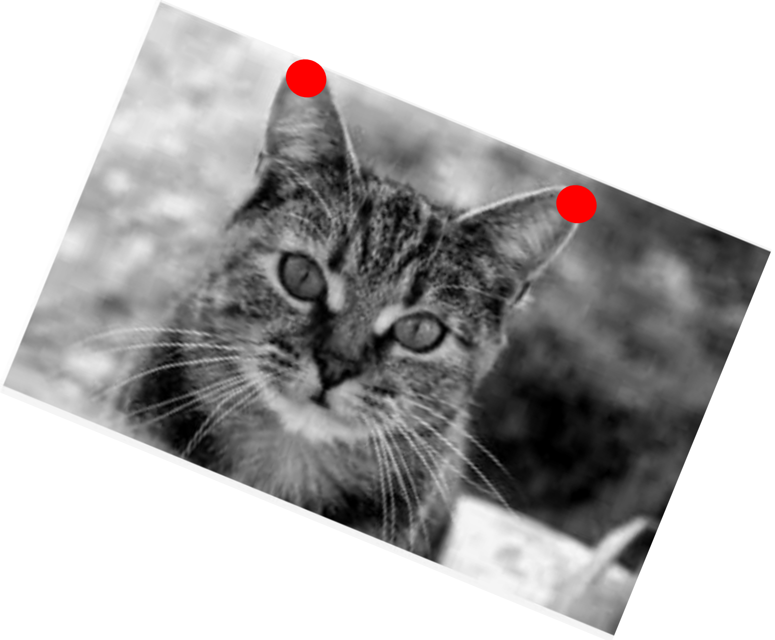
Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data



Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data



Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data
  - Encode invariance/equivariance for a model
  - Idea: model learns a bit of Euclidean geometry this way

Data Augmentation

- Idea: when you have a small dataset, just create more!
- Create synthetic data
- Transform your original data
  - Encode invariance/equivariance for a model
  - Idea: model learns a bit of Euclidean geometry this way
  - Make your model less sensitive to noise

This is still a cat...

Multi-Task Learning

- Two tasks where computed features can be shared
    - For example: Classification + Localization

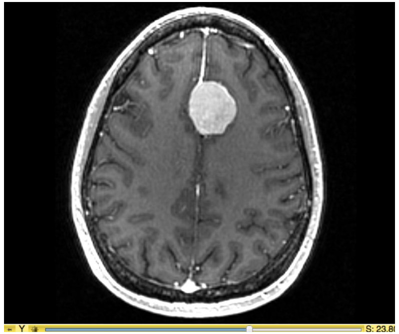
Multi-Task Learning

- Two tasks where computed features can be shared
  - For example: Classification + Localization
- Usually helpful when there is less data than desired somewhere
  - More labels on the same inputs potentially lets you learn “more” about the image
  - Maybe one task has a lot of data and another similar task does not

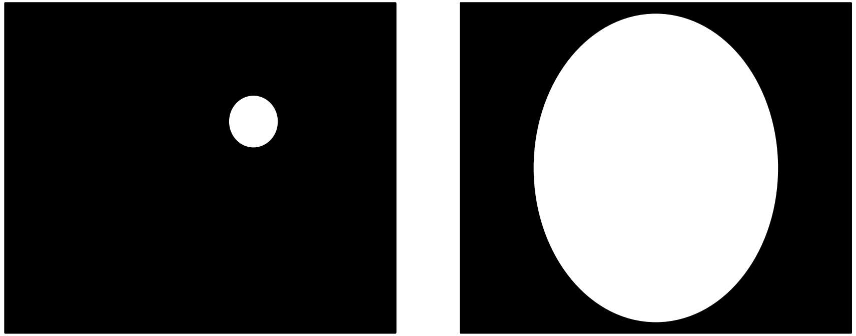
Multi-Task Learning

- Two tasks where computed features can be shared
  - For example: Classification + Localization
- Usually helpful when there is less data than desired somewhere
  - More labels on the same inputs potentially lets you learn “more” about the image
  - Maybe one task has a lot of data and another similar task does not
- Careful!
  - Don’t want a model that is just mediocre at both
  - One task can take over

Multi-Task Learning

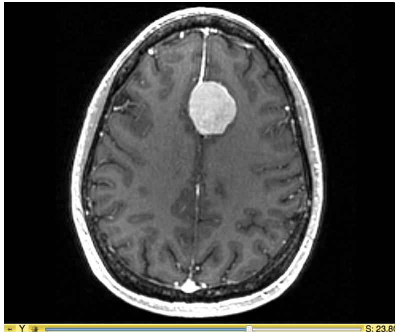


Input

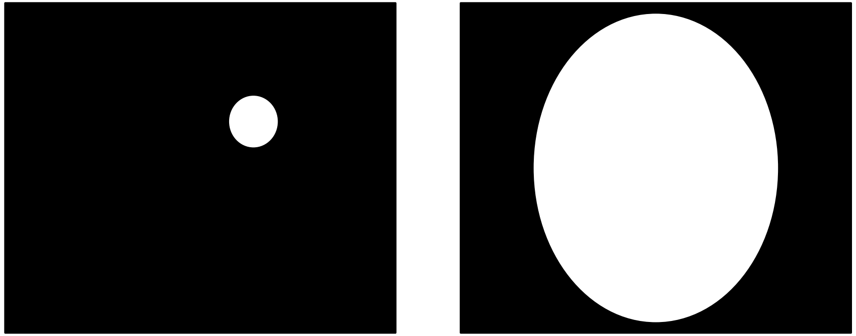


labels

Multi-Task Learning



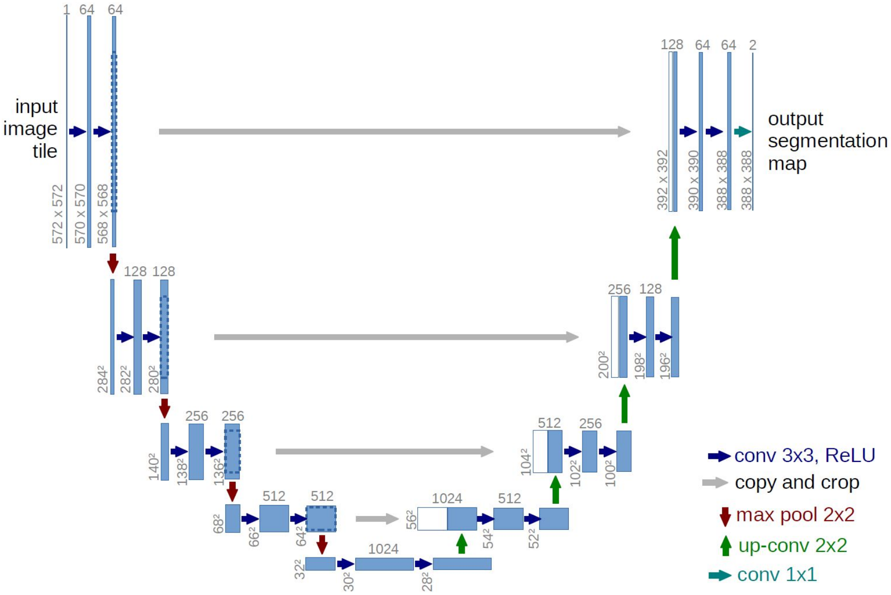
Input

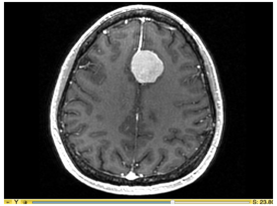


labels

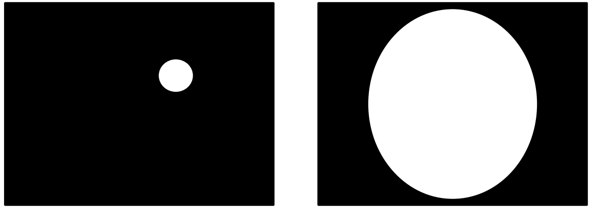
Benign or Malignant

Multi-Task Learning





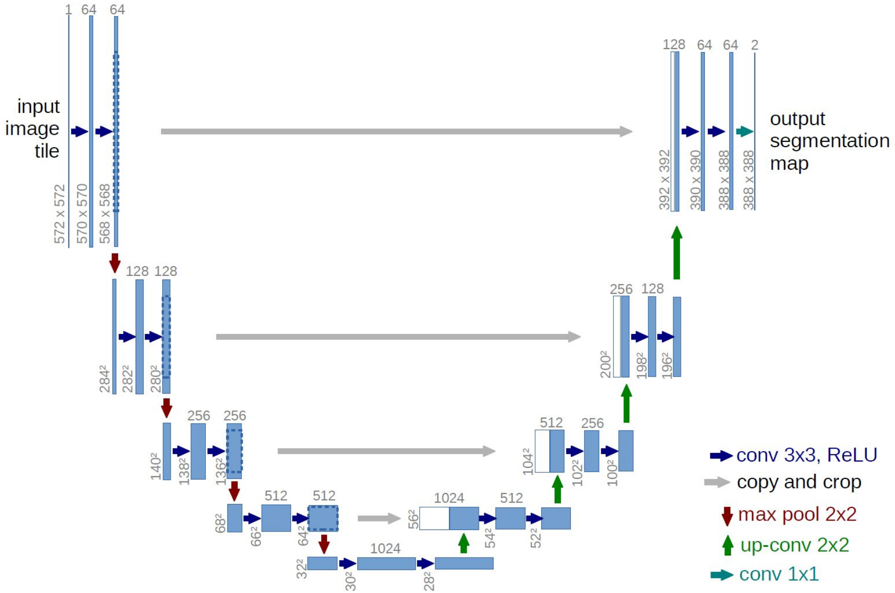
Input

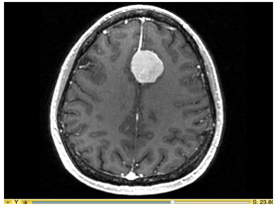


labels

Benign or Malignant

Multi-Task Learning





Input

labels

Benign or Malignant

Unroll to Linear Layer then Classification

Summary

- Transfer Learning
  - Transfer features learned from one dataset/task to another
- Data Augmentation
  - Augment your dataset
  - Synthetic Data
  - Transformed Data
  - Encode invariance/equivariance to nuisance transformations
- Multi-Task Learning
  - Leverage other tasks to improve the target task
  - Like simultaneous transfer learning!