

DL Dev Course: Week 03

CNNs Other uses

- Any dense data
- Text
- Facebook NMT
- CNNs as a feature maker
- Image Search
- Transfer Learning

CNNs -> Vector Embedding

- CNNs can be used as a tool to convert something to a fixed sized vector embedding
- We change the network from predicting a class to giving us a vector embedding out
- Done by chopping off the logits layer
- It has the advantage of no matter what goes in the same size vector will come out, which allows for comparisons between different vectors

Logits layer

Scored
Prediction

0.23	0.00	0.02	0.13	0.0	0.64	0.00	0.1	0.03	0.0
------	------	------	------	-----	------	------	-----	------	-----

Soft Max

Prediction

0	0	0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---

Image search

- Image Descriptor - the algorithm that you use to make a set of features for the image
- Indexing all the images with their features
- Similarity/Error Algorithm for checking and searching

Image search Steps

- Create an Image descriptor
- Index features for each image
- Search Similarity metrics
- Test it

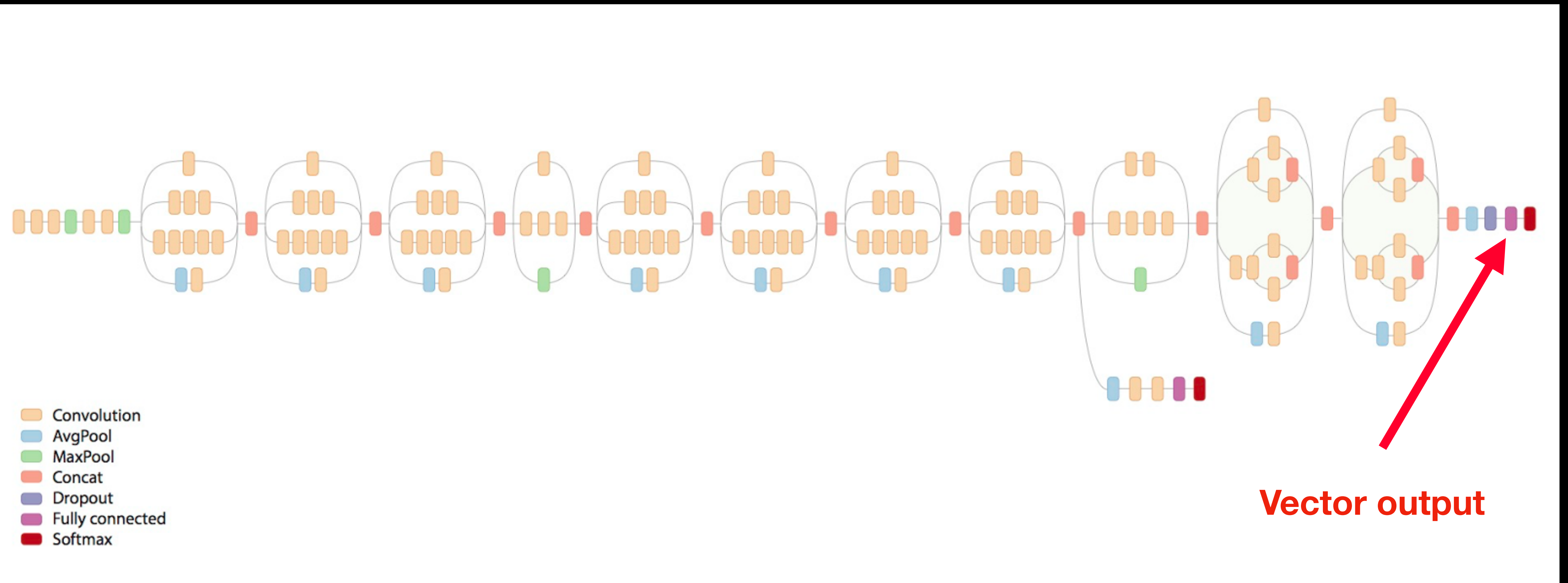
Image Descriptor

- Non DL way is the use histograms of the image
- Often segmented with corners and oval for center
- We are going to use a CNN
- CNN trained on relevant data should get best results
- Dont forget images need to be preprocessed for the network

Search Similarity

- The distance between the 2 vectors
- Euclidian distance
- Cosine distance
- MSE - Mean Squared Errors
- The closer the distance the more alike the images should be

CNNs -> Vector Embedding



Indexing your images

- Set up the network
- Chop off layers (logits or more) to get to the feature set you want
- Make predictions
- Save to array etc

Transfer Learning

- Never underestimate how well this works
- We retrain layers
- Often just the last layer or two (FCN) rather than Conv layers
- Change the number of classes
- Precompute output to make training faster

How much to retrain?

- Ideally as little as possible
- Will depend a lot on how similar your data is to the data that the model was originally trained on
- Try to avoid retraining convolutional layers unless really needed.