The Universality of CNN Distributed Representations

TJ Torres
Data Scientist, Stitch Fix

/in/tjtorres /tjtorres @teejosaur

Introduction

Convolutional neural networks (CNNs) and deep learning are steadily becoming the go-to technique for researchers tackling complex, state-of-the-art, computer vision tasks. However, generally these methods necessitate large quantities of training data. To circumvent this issue, transfer learning with pre-trained networks has been shown to be an effective workaround, allowing for the possibility of using existing training weights to perform discriminative tasks on new datasets.

To do this, one generally strips the final fully-connected layers from an existing pre-trained network and trains a new set over a new (often much smaller) dataset. This process works because the feature extraction portion of the network is often contained greatly in the convolutional structure of the network with the fully-connected section acting as a basic multi-layer perceptron (MLP) over the extracted feature-set.

Here we find that the convolutional representations of images learned on large training sets (with high image variability) are in fact universal, learning similar features with fixed architecture regardless of the original training set. We show that performance on discriminative tasks between learned representations of AlexNet over two different training sets (**Places205** [1] and **ImageNet** [2]) perform nearly identically during testing. Additionally, we show that using extracted convolutional features as a perceptual loss metric during unsupervised learning with variational Auto-encoders produces similar results regardless of which pre-trained weight set is used.

Methods

Using Python's **Chainer** [6] library, two existing pre-trained convolutional weight sets (**Places205** [1] and **ImageNet** [2]) are used to extract vector representation of fashion images from the first fully-connected **AlexNet** [5] layer after convolution ('**fc6**') as shown in Fig. 1. These two sets of features are then fed into the same MLP architecture which is trained over 200 epochs to predict both pattern set and type of clothing simultaneously for a total of 45 classes.

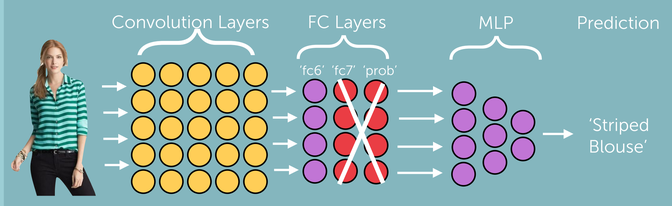


Fig. 1: Structure of pre-trained, truncated AlexNet with newly trained MLP

Data

Fashion image data was collected scraping "In the Wild" images from Bing Image Search with relevant search terms and then manually sanity checking results. The final training/test sets have ~10k/3k images (respectively) over 45 separate classes for pattern/type shown in Table 1. Example images over several classes are shown in in Fig. 2.

Type	Pattern
• Shirt/Blouse • Dress • Pants • Shorts • Skirt	• Animal Print • Chevron • Floral Print • Houndstooth • Paisley • Plaid • Polka-dotted • Striped • Solid/Unpatterned

Table 1: Type and Pattern classes of constructed fashion image dataset.

Fig. 2: Example dataset images. From left to right: 'Chevron Blouse', 'Paisley Dress', 'Solid Shorts', 'Animal Print Pants', 'Floral Print Skirt'.

Supervised Results

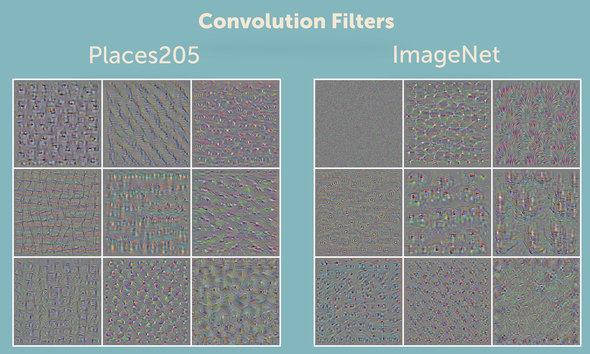


Fig. 3 (above): Images maximizing convolutional activations of the AlexNet 'conv4' layer pre-trained with both the Places205 and ImageNet datasets. Filters pick out similar patterning despite large differences in training images.

Fig. 4 (above): Supervised training on classification tasks results in nearly identical test loss at convergence. This indicates universality of distributed representations for later classification tasks.

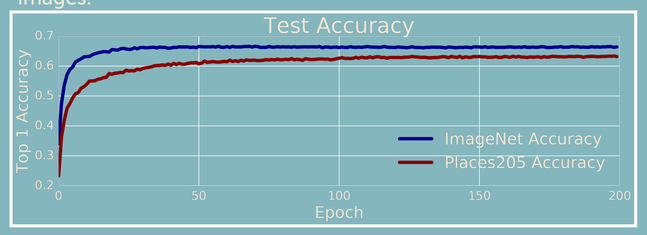


Fig. 5 (left): Top-1 classification testing accuracy for MLP classifiers trained with ImageNet and Places205 extracted image features. Both representations give similar classification performance over the test set.

Unsupervised Results

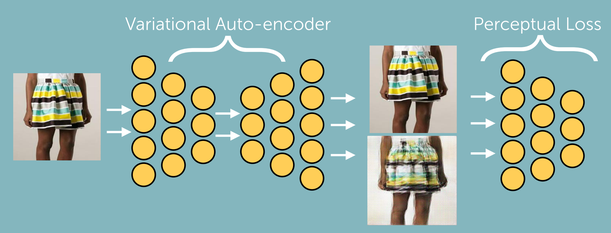
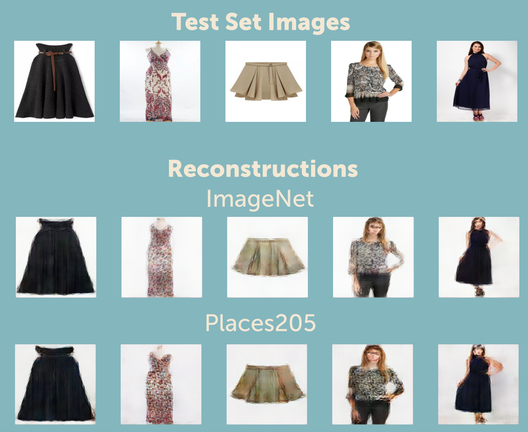


Fig. 6 (above): Auto-encoder utilizing differences in pre-trained CNN activations between original and reconstructed images as a perceptual loss metric.

Fig. 7 (right): Unsupervised image reconstructions of variational auto-encoder models trained with ImageNet and Places205 pre-trained perceptual loss metrics, each model shows similar performance.



Conclusion

Finding the breadth of annotated image data to train a deep end-to-end classification model is often quite difficult. Furthermore, the training time necessary can be prohibitively long. To solve this problem one can turn to the power of transfer learning by initializing the feature extraction (convolution) layer weights in CNNs to those of a pre-trained neural network.

While it is perfectly reasonable to assume that the performance of transfer learning models should depend heavily on the initial training data, it turns out that weight sets for pre-trained networks among a common network architecture are largely universal.

By examining pre-trained CNNs with initial training data from the Places205 and ImageNet datasets, here we have shown that:

1. **Convolution filters learn largely similar hierarchical features regardless of specific training data.**

3. **Feature representations of pre-trained models perform roughly equivalently on subsequent classification tasks.**

5. **Utilizing pre-trained networks for perceptual similarity metrics boots reconstruction performance in deep, unsupervised models and results do not depend highly on initial training data.**

Sources

1. B. Zhou et al. **"Learning Deep Features for Scene Recognition using Places Database."** Advances in Neural Information Processing Systems 27 (NIPS), 2014.

2. Olga Russakovsky et al. **"ImageNet Large Scale Visual Recognition Challenge."** arXiv:1409.0575, 2014.

3. Justin Johnson, Alexandre Alahi, Li Fei-Fei. **"Perceptual Losses for Real-Time Style Transfer and Super-Resolution."** arXiv: 1603.08155, 2016.

4. Lukas Bossard et al. **"Apparel classification with Style."** http://people.ee.ethz.ch/~lbossard/projects/accv12/index.html

5.Alex Krizhevsky et al. **"ImageNet Classification with Deep Convolutional Neural Networks."** Advances in Neural Information Processing Systems 25 (NIPS) 2012.

6. **Chainer: A Powerful, Flexible Framework for Neural Networks.** http://chainer.org

7. Diederik Kingma and Max Welling. **"Auto-Encoding Variational Bayes."** arXiv:1312.6114, 2013.