Code Lab: Face Recognition

By: Kevin Wu

BITTIGER

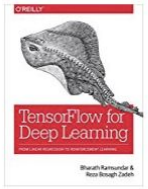
Outline

BITTIGER

1. Homework and Review (15min)
2. VGG network and Transfer Learning (https://github.com/rcmalli/keras-vggface) (25min)
3. A Complete Face Recognition Application (30min)
4. Introduction to Facenet (20 min)

Suggested Books

BITTIGER

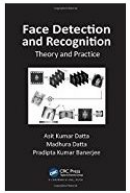


TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning

Bharath Ramsundar, Reza Bosagh Zadeh

Kindle Edition

Sold by: Amazon Digital Services, Inc.



Face Detection and Recognition: Theory and Practice

Datta, Asit Kumar

Sold by: HPB-Ohio

$39.00

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Homework from Week 12

BITTIGER

1. Write a linear regression with L1 regularization using tensorflow, try **sklearn.datasets.load_diabetes**. You only need to implement model training and scoring.
2. Write a logistic regression to predict if the diabetes result is >200 using tensorflow, using elastic net and **L2 loss (Not cross-entropy, sum((y-y_hat)**2).** You only need to implement model training.
3. Complete the following task: save the model (checkpoint) in question 1, open a new file (new.py), create an variable w2 that contains (restores) the variable w in question 1
4. (Optional) Prepare some of your own pictures

Project

BITTIGER

1. Five persons, each person has roughly 20 pictures for training, 5 pictures for testing
2. Target: build a system that takes in a picture, locate the face(s) and then make prediction
3. For face detection, rotate the picture from -60 to 60 degree, and we can use Haar Cascade method directly from OpenCV
4. For face recognition:

    (a) Directly build a model, train from scratch – too small data, too complicated problem

    (b) Pretrain the model for a different but related data set

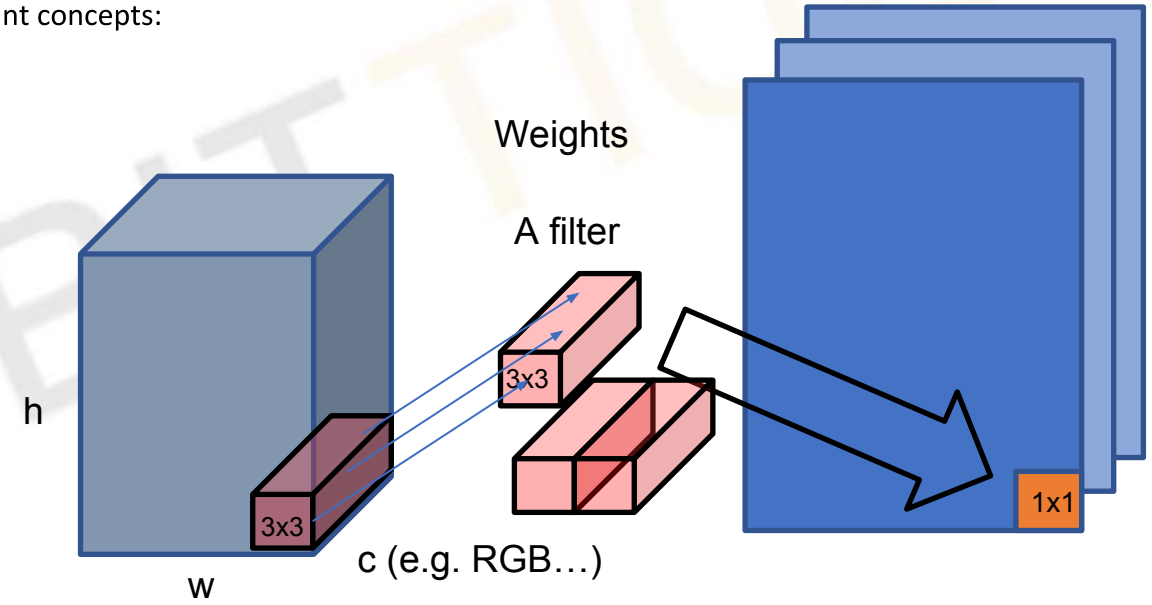
Two strategies: We can either pretrain a model, then use the current model to do fine tuning (change part of the network); Or, even simpler, use the output of the last inner layer as features, and directly do another simple model

Convoluted Neural Network

BITTIGER

The most important concepts:

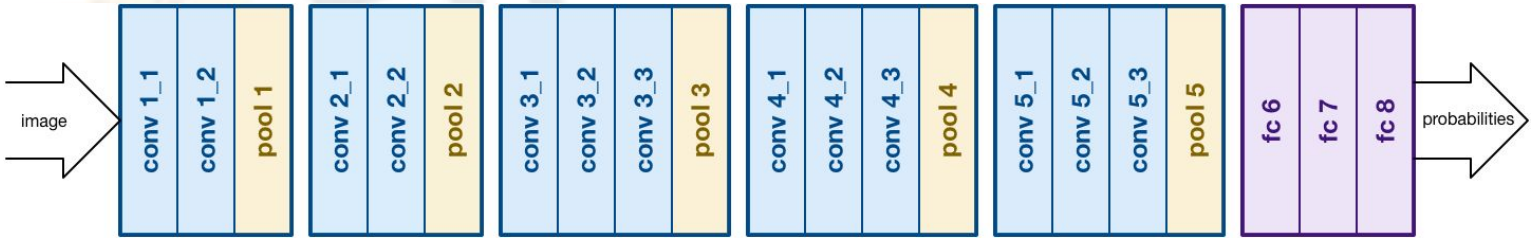
1. Padding
2. Convolution
3. Pooling.
4. Dropout



VGG Net

BITTIGER

1. 3x3 conv. Kernels
2. convolution stride 1
3. Rectification (ReLU) non-linearity
4. 5 max-pool layers (x2 reduction)
5. 3 fully-connected (FC) layers



All the network structures

BITTIGER

Documentation for individual models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

From: https://keras.io/applications/

Transfer Learning

BITTIGER

1. Training a good model usually requires a lot of training data (~1M) and many training hours (~1000 hrs)
2. Many parameters to tweak
3. What if we only have a small size of data?

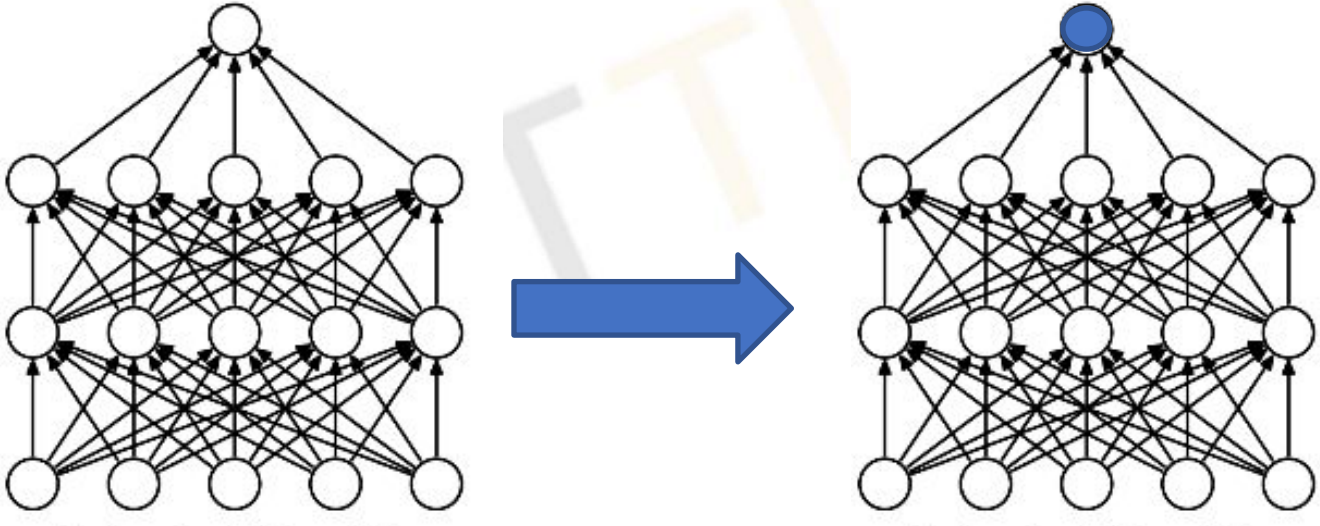
Solution: use models pretrained on similar tasks, compute the representations, and retrain part of the model for the current task

Transfer Learning

BITTIGER

Original labels

New labels



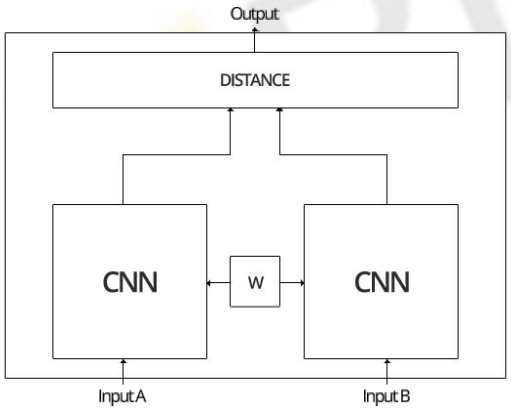
Model trained for other tasks

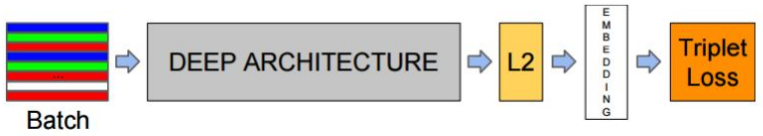
Use the weights in the lower layers

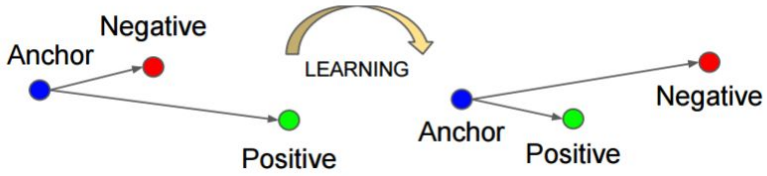
Facenet

FaceNet: A unified embedding for face recognition and clustering. F. Schroff, D. Kalenichenko, J Philbin, ICCV 2015 (260M images!)

Siamese Network and Triplet Loss







$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

Thank You for Joining Us!

BITTIGER

The Lifelong Learning Platform of Silicon Valley