Multi-Label Classification - Deep Learning Vision

Sample Classification

Prediction: (#5) ['Bags_Under_Eyes','Big_Nose','Gray_Hair','Male','No_Beard'];

Try it out

Upload a selfie!

Upload (0)

Classify

What is this project?

This is a web app showcasing a neural network that has been trained to detect various facial features on uploaded pictures. The neural net was originally trained on a subset of 5000 images from the CelebA dataset produced by the University of Hong Kong that labeled images of celebrities with 40 various facial features, including beards, make-up, face type, lipstick, hats, facial expression, and 5 o clock shadows.

Additional Details: Trained using ResNet50 architecture, P5000 GPU, 4 frozen epochs followed by 10 unfrozen at a learning rate of 10e-3, threshold for loss to be considered a label was .5 as that achieved the highest accuracy of 88-91%

What am I looking at right now?

This is actually a Jupyter Notebook being hosted on https://mybinder.org, which is an amazing resource for data scientists that allows you to create environments where people can run your code for themselves.Additionally, I am using the Volia library to remove all input cells leaving only outputs like the

Markdown you're currently reading and the Jupyter Widget below. To view the original notebook this web app was built from, go to the Paperspace link at the top of the page.

How does this differ from single label classification?

For normal vision problems, like deciding whether a picture is a dog or cat, we would normally choose a loss function of binary cross entropy, the problem is that this relies on the softmax function which makes all the predictions sum to 1, which is great if you want to be very confident about one label, but unhelpful for multi classification. Instead we do binary cross entropy which just does a sigmoid (all predictions b/w 1 and 0), tells you loss for every single prediction for the 40 categories on one image, and then takes the log to punish overconfidence.

What libraries does it use?

This project was built using PyTorch and the fastai v4 library which is currently in beta. The background information, tools, and skills learned where from the fast.ai online courses and an earlier draft of this book: Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD: Howard, Jeremy, Gugger, Sylvain: 9781492045526: Amazon.com: Books

Don’t you need a GPU to do deep learning?

Yes, while you can now use a CPU to test the network as it is only looking at one image at a time, for all the actual fun stuff of epochs, learning rates, unfreezing, and then running everything again because you keep underfitting the notebook was initially run on a GPU to be able to quickly go through thousands of images, hence the need for PyTorch’s tensors accessing the speed of C despite being written in Python.The external server I used for this project was through paperspace.com, however I have also had solid experiences using servers from Google Cloud on their student discount.

Where do I go if I have more questions?

Feel free to email me at skapusta@ucsd.edu