Machine learning Engineer Nanodegree, Capstone proposal

Background of the domain (CNN)

In history, Convolutional neural networks were inspired by the work of visual cortex in the brain [1]. The experiment resulted in the knowledge of how the visual cortex works, and was the start of convolutional neural networks. Neural networks, in general, are used for object detection and classification almost in every application nowadays. CNNs have advantage for that, with much less weights than artificial neural networks, it can detect objects with even higher accuracy than traditional ANNs. CNNs work with inputs of variable size and for a single layer in a CNN, the weights are shared for the input of that layer, meaning that while the input image have huge number of input pixels, we can detect different features using small size kernels [2].

When analyzing the performance of a CNN it reduces the memory needed to store weights of the model and improves overall model efficiency. This makes CNN more advantageous than ANN for detecting images where the receptive field for the CNN is effective in extracting useful features through feed forward [2]. CNNs have a main role in Computer vision as they can be augmented with other applications and gives a huge boost to the performance using GPUs and TPUs or even with embedded systems they can still be applicable by handling memory efficiently.

Problem statement.

The problem i chose for the capstone project is 'Writing and algorithm for a Dog Identification App'. The project requires writing an algorithms that fuses a series of models for different tasks, so the application can take a human's image or a dog's image and then detects the breed of that dog or an estimate of the dog breed for a human image. The project first uses a Pre-trained model in order to detect whether the image is a human or a dog and then proceeds with another CNN implemented from scratch to proceed with the detection of the breed. In order to measure the performance of the CNN, simply using the testing and the training accuracy will be enough for identifying how efficiently the model classifies dog breeds. I can use a CNN from a recent state of the art model in a research paper as it's already tested and benchmarked for a top-1 or top-5 accuracy which makes the project even more applicable. More explanation about the mode implemented from scratch is to be covered in the capstone project report, for example I can use EfficientNet model as it was the best in 2020 [3].

Datasets and Inputs

Datasets used in this project are provided by Udacity for both human and dogs. Human dataset in [4] are divided into folders where each folder represents a human name inside of which his/her photos. For dogs' images, the data set in [5] is already divided into train, validation and testing data. All 3 Directories contain 133 sub directories all named with the corresponding breed. Total number of data for human images are 13233 and for dogs are 8351. For dogs' dataset, Number of training images are total of 6680, validation images are 835 and testing images are 836.

In the pre-trained models, the data will be used to assess the models and see how they perform. However in the CNN model implemented from scratch, the data will be used to predict breed from images where we mainly will use the dog images dataset. The data in [4] and [5] can be enough for the training and testing process.

Solution statement

The project's toughest part is the CNN implementation from scratch and that it needs to predict the correct breed for the corresponding dog image or an estimate for a human image. The bottleneck will be the huge number of breeds and the correct number of classification. This issue can be solved using a state-of-the-art CNN model from a research paper that's well benchmarked and tested for top-1 and top-5 accuracy i.e. [3]. One can always change hyper parameters of the model and compare the results with the already implemented model as a process called hyper parameter tuning.

Benchmark Model

In the notebook, we can compare the results from the implemented CNN with the results obtained from a pre-trained model when modified by the transfer learning process. This can give a brief estimate of how well the model is doing along with other metrics like the training, validation and testing accuracy.

Evaluation metrics

In order to evaluate the project, the code should be able to handle input images and classify them correctly. The metrics used are precision, recall and F1 score which is a harmonic mean that averages the precision and recall. One can put execution time for inference or training into consideration and then combine accuracy and running time to get an overall evaluation metric.

Project Design

For a theoretical workflow of the project:

- First the data should be clean and preprocessed i.e. making sure the dimensions of input images are suitable, normalizing images using transformers in order to get better results in training and testing.
- Then use a pre-trained model for human detection. Then we assess the human face detector to make sure it's performing well on the given data set.
- Then we use a pre-trained model to detect dogs in images, the notebook suggests VGG-16 model and looking at specific indices from the 1000 indices in the output of VGG-16 model and assess the model using the given dataset.
- Then continue to implement the CNN from scratch for breed detection for dogs and estimates breeds for human images. Then assessing the model by using user images in an API and return the results.
- We then need to put the dataset into data loaders and use transformers to modify the image dimensions and normalize it for better results.
- We then create the CNN that's responsible for classifying dog breeds depending on the results from the previous 2 pre-trained models which the image is a human or a dog. Either ways, the CNN still needs to estimate or assign the right class for the image.
- Compare the results with a pre-trained model using a transfer learning technique, the model still needs to be trained and tested. However the learning here is specific only for the FC layer at the end of the model not the whole model.
- Finally fuse all the networks implemented to have an API that accepts input images and detects the image content; whether a human, dog or neither

References

[1] D. Mishra, "Towards Data Science Inc," Towards Data Science, 17 Jan 2020. [Online]. Available: https://towardsdatascience.com/translational-invariance-vs-translational-equivariance-f9fbc8fca63a?gi=3f22705bce40. [Accessed 2020].

[2] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, MIT press, 2016.

[3] M. Tan and L. Quoc, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *CoRR,* vol. abs/1905.11946, 3 Jun 2019.

[4] [Online]. Available: https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip.

[5] [Online]. Available: https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip.