CS230

CS230-Fall 2019 Final Project Report
Painting Genre Classification

Soheil Golara
Department of Computer Science
Stanford University
soheilg@stanford.edu

Abstract

I built a model using deep learning alghorithms to classify painting genres. I have explored different Convoltional Neural Netwrok architectures and compared their performance.

1 Introduction

Given an input image of a painting, the convolutional neural network classifies its genre out of 44 defined genres. Moreover, I generate painting images of a particular genre using Generative Adversarial Network (GAN).

2 Related work

I took inspiration from one of CS230 past projects [1] where the students applied Conventional Neural Networks to determine the style of paintings. Also in [2] authors investigated the use of deep residual neural to implement Painting Style Classifier. Painting styles are the pure interest to an art expert and professional; however, the average person might prefer a simple genre classifier to find images. [3] is an example which implements genre classifier using 353 training data images over 5 genres using few different acrhitectures other than CNN. CNN is a better choice especially for a larger dataset and more number of classes.
[4] uses generative and discriminative networks and train them toghether to generate realistic images.[5] uses same idea to generate handwritings and [6] generates modern art.

3 Dataset and Features

I used Kaggle data set "Painters by numbers"[7], where it has almost 80 thousands images across 44 different genres. I wrote a script [8] to read and resize them into desired resolution for each model. I converted 2D grey-scale images into 3D by repeating the same channel 3 times. I also dropped the 4th *alpha channel* in the RGBA images. Since it is easy to find images of a particular painting genres using search engines, I downloaded 30 more images of the ten least popular genres and added them to the data set to reduce the imbalance where genres such as "landscape" and "portrate" have more examples.

CS230: Deep Learning, Fall 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

4 Methods

4.1 Trasnfer learning using Tensorflow

VGG-19 [9, 10] is a very deep convolutional network that we used for Neural Style Transfer [11] early in this course and I used its pre-trained weights to build a genre classifier. Using tensorflow libraries input image is resized to 300x400 and fed through the pre-trained VGG-19 network and the output of fifth average pooling layer ('avgpool5') is flattened and passed through a shallow neural network with a softmax output layer.

For the added shallow NN, I started with 2 layers (hidden layer of dimension 120 and output softamx layer of dimention 44) and trained the model on the dataset (the parameters given in Table.1). Althugh the training accuracy quicky improves but the test set accuracy is poor which shows the model overfit the traning set. Fig. 1 shows the cost function reaching a plateau within 10 iterations.

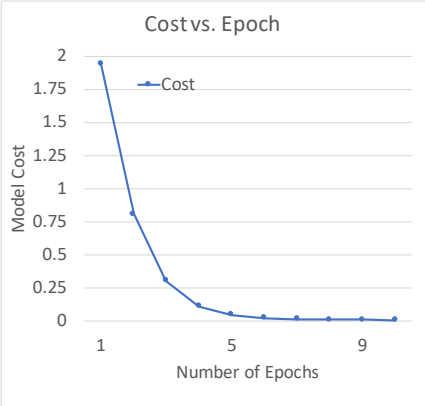

Figure 1: Cost function vs. number of Epochs

$Train Accuracy$	98 %
$Test Accuracy$	56 %
$\#epoch$	10
$mini\ batch\ size$	128
$Training\ set\ size$	25k
$Test\ set\ size$	5k
$Learning\ rate$	0.0001
$Adam\ \beta_1$	0.9
$Adam\ \beta_2$	0.999
$Input\ dimension$	(300,400,3)

Table 1: Performance summary and model parameters

To reduce overfitting I added regularization which did not make significant difference. I also dropped the hidden layer to pass the 'avgpool5' output directly through softamx layer with and without regularization which again did not improved the model signifacntly.
To speed up, I used one third of total data because passing images through pre-trained model is still time consuming. Using all of the data and also re-training some of the last layers of model would improve the performance.

4.2 Residual Neural Network using Keras

Residual network is a powerful architecture to perform computer vision tasks. I implemented ResNet-50 model [12] shown in Fig.2 using Keras libraries [1].

I resized all images to 128x128 and fed to the model. I chose Adam optimizer and crossentropy loss function and trained the model on a CPU which took more than 3 hours for each epoch[2].

To choose hyperparameter values I tried mini batch sizes of 512, 1024 and 2048 and learning rates of 0.001 and 0.01 however due to long training time, I ran the model for only few epochs for each hyperparameter. I trained the model with chosen hyperparameters for 30 epochs and plotted traning set accuracy and loss function in Fig. 3 and Fig. 4 and summary of the results and chosen parameters are given in Table. 2.

[1] Picture borrowed from coursera Convolutional Neural Networks course
[2] Training the same model on the AWS p2.xlarge instance was more than 2x *slower*

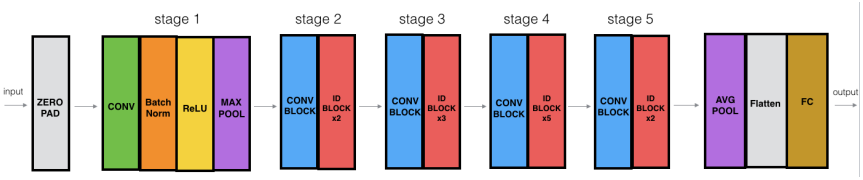

Figure 2: ResNet-50 Architecture

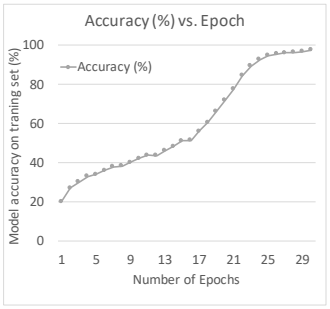

Figure 3: Training set accuracy

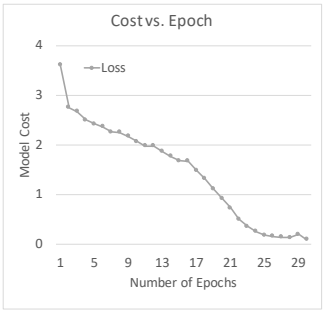

Figure 4: Cost vs. # of epochs

$Train Accuracy$	97 %
$Test Accuracy$	71 %
$\#epoch$	30
$mini\ batch\ size$	2048
$Training\ set\ size$	80k
$Test\ set\ size$	20K
$Learningrate$	0.001
$Adam\ \beta_1$	0.9
$Adam\ \beta_2$	0.999
$Input\ dimension$	(128,128,3)

Table 2: Performance summary and paremeters of ResNET-50

The model seems to overfit the traning set and does not generilize very well to the test set. Adding regularization could help increase the test set accuracy.

4.3 Trasnfer Leaning using Keras

I also used the pre-trained ResNet-50 model available in Keras libraries. I resized images to 224x224 and trained 3 different models using 'imagenet' pre-trained weights. First model keeps the top fully connected layer and adds a softamx layer. Second model replaces the top layer with a softamx layer and third model is the same architecrure as the second model but I retrained last 24 layers. Fig. 5 compares the results. The data fit improves as I re-train more layers at the expense of higher computational cost. I stopped the traning of the model 2 and 3 early becasue the test set accuracy had reached a plateau and further traning would cause more overfitting. The model 3 fits training data well but suffers from high variance similar to fmodel in section 4.2.

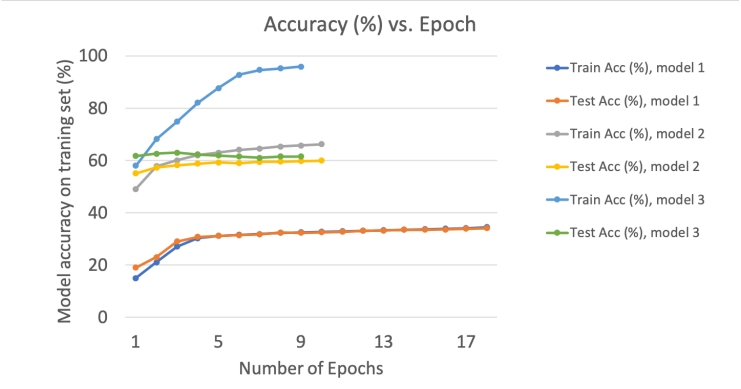

Figure 5: Performance of models based on pre-trained ResNet-50 in Keras

$Train Accuracy$	95 %
$Test Accuracy$	63 %
$mini\ batch\ size$	512
$Training\ set\ size$	80k
$Test\ set\ size$	20K
$Learningrate$	0.01
$Adam\ \beta_1$	0.9
$Adam\ \beta_2$	0.999
$Recall$	62 %
$f1-score$	61 %
$Input\ size$	(224,224,3)

Table 3: Model 3 performance summary

5 Experiments/Results/Discussion

Table. 4 summarizes the performance of different models. They generraly suffer from high variance that can be due to data imbalance in our dataset where unpopular genres are misclassified as a one of popular genres ('landscape','abstract','portrait'). Confusion matrix shown in Fig. 6 points to cluster of misclassified images around the popular genres.

Model	Train Acc (%)	Test Acc (%)
VGG-19 (trasnfer learning)	98 (%)	56 (%)
ResNet-50 (Fully trained)	97 (%)	71 (%)
ResNet-50 (transfer learning, model 1)	37 (%)	37 (%)
ResNet-50 (transfer learning, model 2)	69 (%)	61 (%)
ResNet-50 (transfer learning, model 3)	95 (%)	62 (%)

Table 4: Performance summary of different models

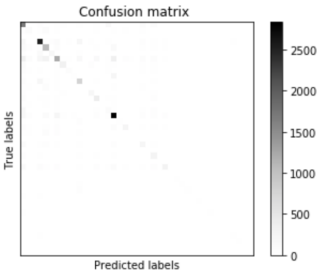

Figure 6: Confusion matrix for model 3

Another possible improvment is to alter the loss function to add more penalty for mis-classification on the genres that are harder to distinguish. For example Fig.7 is a painting [3] labeled as a "genre-painting" where the trained model predicts "figurative". The genres that are difficult for non expert human to disinguish are probably harder as well for the model. Fig.8 is a "history" painting [4] misclassified as "landscape" by the model.

Figure 7: "genre-painting" image misclassified as "figurative"

Figure 8: "history" image misclassified as "landscape"

6 Conclusion/Future Work

The models did not generalize very well to test set and common error is misclassifying into a common gernre. To better deal with dataset imbalance, in future a data augmentation script should be used to make sure each genre has at least 1000 examples becasue manually downloading new images in a larger scale is time consuming.
Given more time more comprehensive hyperparameter search should be performed where we train model using diffrent values and compare the end results as opposed to juding based on only first few epochs.
In this dataset and project each image has an unique genre label however it might be better to assign multiple labels to a painting which can fall under multiple genres such as abstract flower painting which can be labeled as both "flower painting" and "abstract painting".
Retraning some of the output layers of VGG19 model can improve the model. Using larger image size for ResNet-50 model may also improve the results.

[3] by Ford Madox Brown
[4] The Life of St. Ignatius Loyola by Carlos Saenz de Tejada

7 Generating landscape images using GANs

I resized the landcsape images to 128x128 and used them as traning set for generative and discriminator netwroks. I used cross entropy loss for discrimitaor and joint loss for overal system [6, 13] where generator is trying to maximize D(G(z)) while discriminator is trying to minimize it.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{1}$$

Fig.9 shows the generated image after 120 epochs where is almost looks like a noise. A lot more iterations is needed to acheive an appealing result (possibly 10000 iterations) and will be added later.

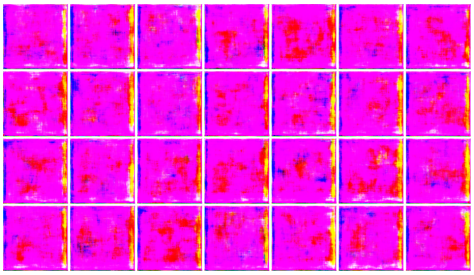

Figure 9: Generated landscape images after 120 epochs

8 Contributions

Solo project.

References

[1] Sean Chang Jeffrey Dong Chen, Patrick James Tanaka Mogan. Determining style of paintings using deep learning and convolutional neural networks.

[2] Adrian Lecoutre, Benjamin Negrevergne, and Florian Yger. Recognizing art style automatically in painting with deep learning. In *Asian conference on machine learning*, pages 327–342, 2017.

[3] Jana Zujovic, Lisa Gandy, Scott Friedman, Bryan Pardo, and Thrasyvoulos N Pappas. Classifying paintings by artistic genre: An analysis of features & classifiers. In *2009 IEEE International Workshop on Multimedia Signal Processing*, pages 1–5. IEEE, 2009.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Gan by example using keras on tensorflow backend, . URL `https://towardsdatascience.com/gan-by-example-using-keras-on-tensorflow-backend-1a6d515a60d0`.

[6] Generating modern arts, . URL `https://github.com/Anyesh/art_gan`.

[7] Painters by numbers. URL `https://www.kaggle.com/c/painter-by-numbers/`.

[8] URL `https://github.com/solig/Paiting-Genre-Classification.git`.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] URL `http://www.vlfeat.org/matconvnet/pretrained`.

[11] URL `https://github.com/anishathalye/neural-style`.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Ch-14 : Generative adversarial networks (GAN's) with math, . URL `https://medium.com/deep-math-machine-learning-ai`.