Ridge-i Technical Assignment

Tiago Oliveira

November 20, 2019

1 Introduction

Autoencoders are a special case of neural networks. An autoencoder is trained to attempt to copy its input to its output. Therefore, an autoencoder is viewed as consisting of two parts: an encoder($h = f(x)$) and a decoder ($g(f(x)) = x$). Usually, autoencoders are restricted so as to be only able to approximately copy the input, in order to make them learn useful features from data, rather than just copying the input to the output.

Regarding the size of h, one can distinquish two types of autoencoders: undercomplete, when h has a smaller dimension than x; and overcomplete

Several works point out the usefulness of autoencoders in multiple computer vision tasks such as anomaly detection, image denoising, or even image classification. We will focus on the latter and use the encoder in a Convolutional Autoencoder (CAE) to classify images from the CIFAR10 dataset.

2 Task Description

The task to tackle in this project is the design of a network that combines supervised and unsupervised architectures in one model to achieve a classification task. the model must start with an autoencoder which is then connected through its hidden layer to another network as shown in Figure.

The problem to tackle is the classification of images from the CIFAR 10 dataset using an imbalanced training set and resorting to an autoencoder to derive meaningful

As such the set of objectives for this project are the following:

- Propose an autoencoder architecture and a classifier architecture to

3 System Description

The models in this project were trained in system with the following specifications:

- **GPU:** 1xTesla K80 , having 2496 CUDA cores, compute 3.7, 12GB(11.439GB Usable) GDDR5 VRAM
- **CPU:** 1xsingle core hyper threaded i.e(1 core, 2 threads) Xeon Processors @2.3Ghz (No Turbo Boost) , 45MB Cache
- **RAM:** 12.6 GB Available
- **Disk:** 320 GB Available

4 Dataset Description

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain 5000 images from each class.

Training and test batches can be easily downloaded with the *load()* function from the *cifar10* module of Keras. This function returns the five training batches consisting of 50 000 images in one dataset and a test batch consisting of 10000 images in another dataset.

5 Pre-processing

After loading the CIFAR-10 data, 50% of images from classes bird, deer, and truck were randomly removed from the training batches. This dataset is then then split into a training set and a validation set, both with approximately the same class proportions as the training batches after removal, i.e., bird, deer, and truck each have approximately half as many images as each of the remaining classes. The percentage of images in the training batches dataset used for validation was set at 22% so that the validation set would have approximately the same number of samples in each class as the test set (except for classes bird, deer, and truck).

The following step was the normalisation of pixel values to the [0,1] interval by diving each image pixel in the training, validation and test sets by the maximum pixel value found in the training set. Typically this maximum pixel value is 255 and that was also the case herein. The number of samples in each class and the total number of samples in the training, validation and test sets are described in Table 5.

Class	Training Set	Validation Set	Test set
airplane	3910	1090	1000
automobile	3891	1109	1000
bird	1944	556	10000
cat	3874	1126	1000
deer	1981	519	10000
dog	3913	1087	1000
frog	3894	1106	1000
horse	3889	1111	1000
ship	3905	1095	1000
truck	1949	551	1000
Total	33150	9350	10000

Table 1: Class distribution of

6 Performance Metrics

The fact that the model used for the multiclass classification task is an ensemble of two different models, a supervised model and an unsupervised model, means that their respective training and validation performances will be assessed with different performance metrics. The autoencoder was assessed with Mean Squared Error Loss (L_{MSE} as one is only interested in monitoring the pixel-wise difference between the input and the output. As for the classifier, the loss function used was

These performance metrics are formally defined in the ensuing sections.

6.1 Mean Squared Error Loss

L_{MSE} is defined as follows:

$$L_{MSE}(y, \hat{y}) = \frac{1}{m}\frac{1}{p}\sum_{j=1}^{m}\sum_{i=1}^{p}(\hat{y}_i - y_i)^2 \tag{1}$$

where m is the number of image samples, p is the number of predictions per sample, y is the ground truth for the prediction and $\hat{y}$ is the model prediction.

6.2 Categorical Cross-entropy Loss

L_{CCE} is defined as follows:

$$L_{CCE}(y, \hat{y}) = \sum_{j=1}^{m}\sum_{i=0}^{n}(y_{ij} * log(\hat{y}_{ij})) \tag{2}$$

where m is the number of predictions, n is the number of different classes, y is the ground truth for the prediction and $\hat{y}$ is the model prediction.

6.3 Accuracy, Precision, Recall, and F1-score

In a multiclass classification problem such as this one, accuracy is defined as follows:

$$Accuracy = \frac{1}{n}\sum_{c=1}^{n}\frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \tag{3}$$

where n is the number of classes, c is a single class, TP_c, TN_c, FP_c and FN_c are respectively the number of true positives, the number of true negatives, the number of false positives and the number of false negatives with respect to class c.

Similarly, precision is defined as:

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \tag{4}$$

where C is a single class, TP_c is the number of true positives with respect to class c, TN_c is the number of true negatives with respect to class c, FP_c is the number of

$$Accuracy = \frac{1}{N}\sum_{i=0}^{N}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{5}$$

$$Accuracy = \frac{1}{N}\sum_{i=0}^{N}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{6}$$

The Cifar10 dataset can be directly downloaded using a keras module called

Images are very simple so one should use use data augmentation techniques that do not cause too much distortion in the image.

7 Modelling

7.1 Autoencoder

Convolutional Autoencoder (CAE) Is and autoencoder with stacked convolutions. Fully connected autoencoders (AEs) ignore the 2D image structure. Furthermore, these AEs they introduce redundancy in the parameters by each of the features to be global and span the entire visual fiels, as opposed to the current trend in computer vision of

Stacked convolutional Number of convolution layers 1, 2, 3

downsampling Max-pooling (fixed) has been shown to perform better) vs strided convolutions (learned)

regularization Droupout BatchNorm Kernel regularizers (l1,l2)p

optimizers Adam SGD RMSprop

the decoder should not ude regularization avoid iformation loss

7.2 Classifier

CIFAR-10 consists of small size images. Given the simplicity of the network a model

8 Evaluation

9 Results

10 Conclusions

Things that could have been done better

validation check of configuration dictionary

strided convolutions

mention conv2dtranspose

encoder and decoder should have been functions should have been a fu

References