Comparative Analysis of Classifiers on Audio, Image, and Text Datasets

Akshay Singh Rana Université de Montréal
Harmanpreet Singh Université de Montréal
Himanshu Arora Université de Montréal

Abstract

Recently, machine learning has found its applications in a diverse set of domains. Specifically, many real-world decision-making problems fall into the category of classification. Moreover, lots of content generated today is in the form of different modalities, such as text, audio and visual. This motivated us to explore the classification performance of popular machine learning algorithms leveraging audio, text and images.

1 Introduction

Machine learning has achieved great success in many applications such as image analysis, speech recognition and text understanding. Especially, deep learning uses supervised and unsupervised strategies to learn multi-level representations and features in hierarchical architectures for the tasks of classification and pattern recognition. In this paper, we present a comparative study of different machine learning supervised classification techniques such as Naive Bayes, Logistic Regression, and Support Vector Machine (SVM), as well as, deep learning algorithms such as MultiLayer Perceptron (MLP), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) on different data modalities.

1.1 Algorithms

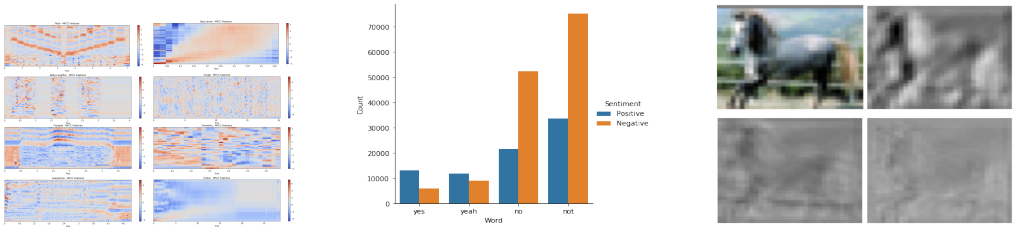
Naive Bayes: It assumes all the features to be conditionally independent and hence can be extremely fast even on a high-dimensional distribution. In spite of overly simplified assumptions, it seems to perform quite well on real-world situations like text classification problems.
Logistic Regression: It, despite its name, is a probabilistic linear model for classification rather than regression. The regularized version is extremely useful to avoid overfitting, resulting in a more generalized model.
SVM: Considered as one of the most powerful classifiers, SVM is very effective in high-dimensional space and is able to create both linear and non-linear decision boundaries.
MLP: It learns a complex non-linear function approximator using multiple hidden layers and is a great fit for our classification problems.
CNN & RNN: They are deep neural network algorithms which achieved state-of-the-art results on image and text classification problems.

1.2 Datasets

We chose 3 opensource datasets, one each for audio, text and images. Experiments were conducted on the Freesound Dataset [1] for audio event detection, the Sentiment140 [2] dataset for sentiment classification and the CIFAR-10 [6] dataset for image classification. Table 1 provides a statistical overview of all three dataset modalities. These academic datasets are reputed in their respective domains and are used as standard benchmarking datasets for comparative analysis. Our motivation to execute learnings from class to experiment on these datasets lead us to choose them.

Datasets	Train Size	Validation Size	Test Size	Number of Classes
FreeSound Audio [1]	2717	300	300	10
Sentient140 [2]	900k	300k	400k	2
CIFAR10 [6]	80k	20k	20k	10

Table 1: Dataset Statistics



(a) Audio MFCC features (b) Word Count Text features (c) Image features

Figure 1: Feature Visualization

Audio: Freesound is a general purpose heterogeneous uncompressed PCM 16 bit, 44.1 kHz, mono audio files consisting of 10 uniformly distributed categories drawn from the AudioSet Ontology (related to musical instruments, human sounds, animals, etc.).

Text: The Sentiment140 contains 1.6 million annotated tweets of maximum 140 characters with a positive or a negative label for sentiment detection.

Image: The CIFAR-10 dataset consists of 32x32 color images of 10 classes, with 6000 images per class.

2 Methodology

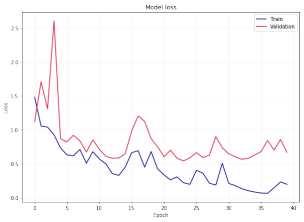
We review the performance of popular machine learning and deep learning algorithms on classification tasks on various modalities using 5-fold cross validation. Table 2 summarizes the accuracy comparison of the experimented algorithms. We also discuss the methods used for audio and images feature representation as shown in Figure 1.

2.1 Audio: Freesound 2018

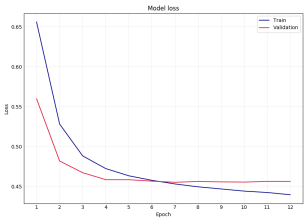
Freesound training data contains variable-length audio clips ranging from 300ms to 30s. We pre-process each audio file by either stripping or padding it to average audio length of 2s. The extraction of the best parametric representation of acoustic signals is an important task to produce a better classification performance. Therefore, we extract mid-level representations from audio data by computing Mel Frequency Cepstral Coefficents (MFCCs) [7] spectrograms to create 2D image-like patches. MFCC features are derived from Fourier transform and filter bank analysis, and downstream tasks does much better on them than raw features. One of the main advantages of using 2D representations is that spectrograms can summarize high dimensional waveforms into a compact representation. The extracted features from each audio are of shape 40×173. Later, we feed these features to CNN [9] as it is, or flatten them before feeding them to machine learning algorithms. We perform extensive hyperparameter search on these models to report the best accuracy.

2.2 Text: Sentiment140

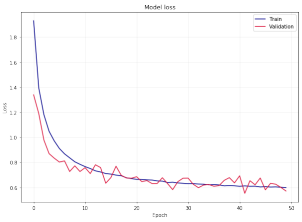
Since the Twitter platform is informal, the tweets contain a lot of errors and thus needed to be thoroughly preprocessed before getting fed to the machine learning models. We use the Natural Language Toolkit (NLTK) library for it. We begin with tokenizing the tweet sentences into words. Next, we mask all the links, twitter user mentions, and numbers to specialized tokens. We replace hashtags with their corresponding words as they contain meaningful information about the sentence.



(a) CNN loss: Audio



(b) LSTM loss: Text



(c) CNN loss: Image

Figure 2: Loss Curve

Text data usually has a very high-dimensional representation in the bag-of-words model due to huge vocabulary of words. This creates problems in training the algorithms that do not handle the curse of dimensionality well. To further reduce the dimensionality, we lemmatize all the words in the tweets and reduce them to their root form. Usually, removing common language words (stop words) is a part of the preprocessing pipeline but we observed that removing stop words from the dataset degraded the performance of all the models we tried. This could be because often, a single affirmative or negative word inverts the sentiment of a sentence. Most of the stop word lists contain a lot of these words and removing them would deteriorate the performance. This phenomenon is also confirmed in the work done by Saif et al. [8]. We also found that using a combination of unigrams and bigrams helped the algorithms learn better than just using one of them.

2.3 Image: CIFAR 10

CIFAR 10 is one of the most widely used dataset for machine learning reasearch especially in the field of object detection. Since the images are of low resolution, we can experiment different machine learning algorithms as well as deep learning architectures to see what works. We tried various linear classifiers and as expected, they didn't perform well as the dataset is non-linear in nature. It is evident that we need to add non linearity in our model and we tried Multi Layer Perceptron with varying number of layers and neurons. The model did improve the accuracy to 0.50. The immense complexity of the object recognition task means that we need a model with a large learning capacity. CNN [3] constitutes one such class of models and their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images. Thus, compared to standard feedforward neural networks with similarly-sized layers, CNN have much fewer connections and parameters and so they are easier to train. To reduce overfitting in the layers we employed batch normalization [4] layers after convolution layers and that proved to be very effective. We achieved 0.86 accuracy on the test set and believe that the performance can be significantly improved by training the model for more epochs and tuning the hyperparameters. However, finding the right hyper parameters, often requires significant resources, in terms of time, memory and computing power so we ran the model on 50 epochs using 6 convolution and pooling layers with a kernel size of 3x3.

3 Results

Generally CNN models outperformed popular machine learning algorithms on complex problems such as image and audio classification. In particular, for audio classification the results in Table 2 show that CNNs are capable of excellent results when compared to other machine learning algorithms. Experiments were performed with different batch sizes by training on GPU using Adam [5] optimizer. Batch Normalization was applied after all convolutional layers. The CNN loss plot in Figure 2a demonstrates how the model loss decreases during our training. From confusion matrix 3a for CNN classifier in appendix, we find that few classes were easy to classify such as Applause, Fireworks and Bass Drum, but the model found it hard to distinguish between audios from musical devices such as Saxophone and Flute. Moreover, number of examples of Saxophone class in validation was comparatively large, which makes it more prone to misclassification. Machine learning algorithms

Algorithms	Audio		Text		Image	
	Train	Test	Train	Test	Train	Test
Naive Bayes	0.56	0.55	0.85	0.80	0.29	0.28
Logistic Regression	0.46	0.40	0.86	0.81	0.30	0.26
SVM	0.60	0.54	**0.91**	**0.81**	0.36	0.29
MLP	0.31	0.32	0.81	0.80	0.48	0.50
CNN	**0.96**	**0.81**	-	-	**0.87**	**0.86**
LSTM	-	-	0.80	0.78	-	-

Table 2: Accuracy comparisons

such as Naive Bayes, L2 Regularized Logistic Regression and SVM under-fit because they don't have enough capacity to model the complex characteristics of audio data.

For text, the results in Table 2 show that most of the algorithms performed similar to each other by achieving an accuracy of around 80% on the test set. This was achieved by identifying the best feature representation and the optimal hyper-parameters for each of them. After preprocessing, the vocabulary of unigrams and bigrams becomes 3.4 million, which is huge for certain algorithms. The Naive Bayes classifier easily handled this and gave good accuracy with minimum training time. The Logistic Regression and SVM classifiers were trained on a reduced vocabulary size of 3 million, by selecting the most frequent unigrams and bigrams. The multi-layer perceptron was very slow to train and was thus fed features with a vocabulary size of only 10k unigrams and bigrams. For recurrent neural network, pretrained GloVe embeddings were used as feature representations. The Long Short Term Memory (LSTM) was the hardest to train due to high computational complexity and high sensitivity to hyperparameters. The hyperparameters can further be explored to improve the accuracy. We further observe that the loss curves for this dataset show the least amount of variance, this could be due to the fact that the training and validation sets contain a lot of data.

For images, the best results have been obtained by using CNNs as shown in Table 2. From confusion matrix 4a, we find that the model seems to be confused between similar classes such as horses and deer, trucks and automobiles, ships and planes, etc. Experiments were performed with different batch sizes by training on GPU using Adam [5] optimizer. Machine learning algorithms such as Naive Bayes, Logistic Regression and SVM underfit because they don't have enough capacity to model the complex characteristics of our dataset.

4 Conclusions

Different algorithms work well for specific tasks and specific data modalities. In our case, Convolutional Neural Networks work better for spatial datasets such as images and audios whereas the relatively simple text classification task was successfully learned by most of the algorithms but Naive Bayes classifier was the fastest to train and handled the curse of dimensionality well. Task-dependent pre-processing of the data is important to achieve high accuracy and careful review of algorithms prior assumptions is required when choosing which one must be used. Dense feature representation is crucial for text and audio classification and we used embeddings for text and MFCC features for audio. Transfer learning can be used for all three modalities to further improve scores.

Convolutional layers extract features from patches of audio data by applying a non-linearity on an affine function of the input Mel Frequency Cepstral Coefficents features. But we can further enhance the feature extraction process for the case of sequential data, by feeding patches of these features into a recurrent neural network and using the outputs or hidden states of the recurrent units to compute the extracted features. By doing so, we exploit the fact that a window containing a few frames of the sequential data is a sequence itself and this additional structure might encapsulate valuable information. Additionally, increasing the amount of audio data can help our model learn better.

5 Acknowledgments

Contributions were evenly distributed between the authors. All three authors worked together in refining the problem statement and methodology. Later, each author worked independently comparing the performance and characteristics on each of the modalities a) Audio - Harmanpreet Singh b) Text - Himanshu Arora c) Images - Akshay Singh. The ideas were later shared among each other, which finally lead to a successful result. All the authors worked together as a group to write the final project report.

We hereby state that all the work presented in this report is that of the authors.

References

[1] Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel P. W. Ellis, Xavier Favory, Jordi Pons, and Xavier Serra. General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline, 2018.

[2] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12):2009, 2009.

[3] Seyyed Hossein Hasanpour, Mohammad Rouhani, Mohsen Fayyaz, and Mohammad Sabokrou. Lets keep it simple, using simple architectures to outperform deeper and more complex architectures, 2016.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

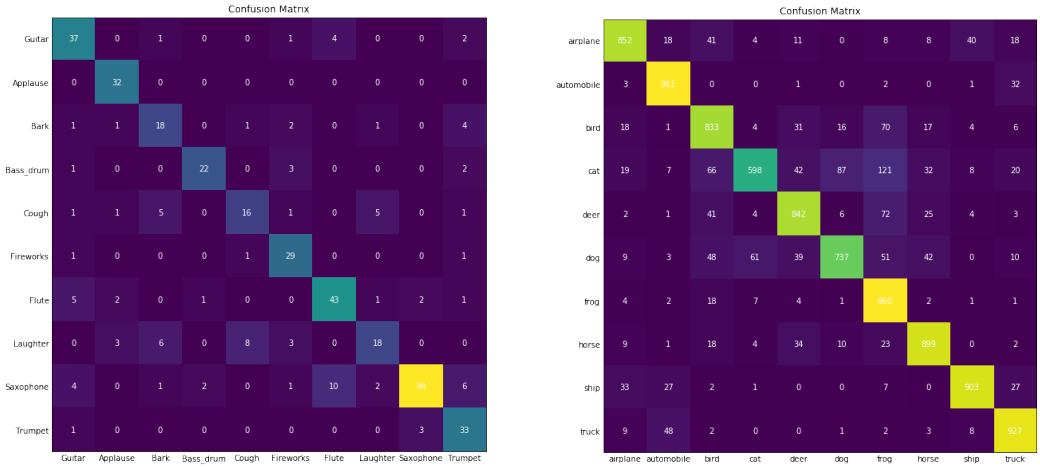
[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[7] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki. Speaker recognition using mel frequency cepstral coefficients (mfcc) and vector quantization (vq) techniques. In *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, pages 248–251, Feb 2012.

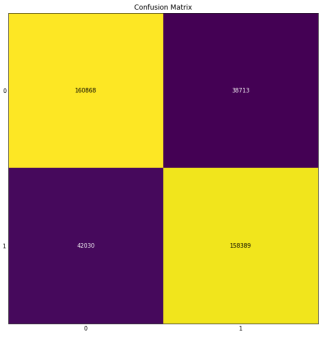
[8] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. On stopwords, filtering and data sparsity for sentiment analysis of Twitter. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 810–817, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[9] Ankit Shah, Anurag Kumar, Alexander G. Hauptmann, and Bhiksha Raj. A closer look at weak label learning for audio events, 2018.



(a) Audio (b) Images

Figure 3: Confusion Matrix with CNN model



(a) Text

Figure 4: Confusion Matrix with LSTM model