Deep Learning Approaches towards Book Covers Classification

Przemyslaw Buczkowski[1,2], Antoni Sobkowicz[2] and Marek Kozlowski[1]
[1]*National Information Processing Institute, Warsaw, Poland*
[2]*Warsaw University of Technology, Warsaw, Poland*

Keywords: Image Classification, Convolutional Neural Networks, Pattern Recognition, Deep Learning, Machine Learning, Supervised Learning, Artificial Intelligence.

Abstract: Machine learning methods allow computers to use data in less and less structured form. Such data formats were available only to humans until now. This in turn gives opportunities to automate new areas. Such systems can be used for supporting administration of big e-commerce platforms e.g. searching for products with inadequate descriptions. In this paper, we continue to try to extract information about books, but we changed the domain of our predictions. Now we try to make guesses about a book based on an actual cover image instead of short textual description. We compare how much information about the book can be extracted from those sources and describe in detail our model and methodology. Promising results were achieved.

1 INTRODUCTION

In this paper, we focus on a task called image classification. The task is to predict the probability of classes given raw pixels intensities. Class refers to one of predefined, discrete category. In machine learning problems concerning image data, two distinct approaches are available. First one, more classical, is to flatten 2d image data into single dimension vector. After such transformation, any classifier feedable with the constant-length vector can be used. The problem with this approach is that it ignores pixel neighborhood and therefore cannot exploit information about local patterns. In other words, this approach cannot distinguish between pixels which are close to each other and those which are not, because proximity information has been lost in the process (or in this representation, to be more precise). Another, but related flaw is a complete lack of invariance with respect to translation because translating object few pixels away creates big and hard-to-cope-with changes in the flattened vector. On the other side are methods which do not discard proximity information, quite the contrary, they are designed to exploit them. Firstly, it was noticed that extracting spatial-aware features from an image before using classifier may improve quality of the model. Examples of such features are handcrafted operators like Prewitt or Sobel (Adlakha et al., 2016). A more advanced example is Gabor filter (Feichtinger and Strohmer, 1998) which is inspired by reverse engineering of visual cortex in the brain (Jones and Palmer, 1987). Filters are small 2 dimensional images which are multiplied pixel-wise with different parts of the images. This operation is called convolution and results with new, slightly smaller image. The superior quality of systems based on convolution, or more generally spacial–aware systems, caused an increase of interest in such methods. Big image-related machine learning challenges, like ILSVRC (Russakovsky et al., 2015) or MSCOCO (Lin et al., 2014) are dominated by convolutional neural network for a couple of years now (Krizhevsky et al., ; Szegedy et al., ; He et al.,). In this paper, we successfully apply convolutional neural networks to book genre prediction based on cover images. Dataset is acquired by crawling from GoodReeds.com [1]. We briefly describe the structure of this data set as well our previous analysis with NLP methods. Lastly, we describe network architecture and results.

2 RELATED WORK

Machine learning is found everywhere in today's data processing works. Most current machine learning works well because of the human-designed representations and input features. Traditional methods are mostly focused on numerical optimization of weig-

[1]https://www.goodreads.com/

Buczkowski, P., Sobkowicz, A. and Kozlowski, M.
Deep Learning Approaches towards Book Covers Classification.
DOI: 10.5220/0006556103090316
In *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2018)*, pages 309-316
ISBN: 978-989-758-276-9

hts for human designed representations and features. Recently, representation learning, as sometimes deep learning is called, has emerged as a new area of Machine Learning research, and it attempts to automatically learn good latent features. Deep learning attempts to learn multiple levels of representation of increasing complexity/abstraction. The goal of this approach is to explore how computers can take advantage of data to develop features and representations appropriate for complex interpretation tasks. The central idea behind early deep learning models was to pre-train neural networks layer-per-layer in an unsupervised fashion, which allows to learn hierarchy of features one level at a time. Moreover, pretraining can be purely unsupervised, allowing researchers to take advantage of the vast amount of unlabeled data. Such approach makes Deep Learning particularly well suited for Image and Natural Language Processing tasks, where there is a huge number of images and texts abound. Additionally, one can use deep features as an input to standard supervised machine learning methods.

Deep architectures are mainly neural networks (recurrent, convolutional, deep belief) and can be summarized as the composition of three elements: (1) input layer - raw sensory inputs (e.g. words, Red-Green-Blue values of pixels in an image); (2) hidden layers - those layers learn more abstract non-obvious representations/features; (3) output layer - predicting the target (LeCun et al., 2015).

Recently, deep learning approaches have obtained very high performance across many different NLP tasks. These models can often be trained with a single end-to-end model and do not require traditional, task-specific feature engineering. The most attractive quality of these techniques is that they can perform well without any external hand-designed resources or time-intensive feature engineering. Moreover, it has been shown that unified architecture and learning algorithm can be applied to solve several common NLP tasks such as part-of-speech tagging, named entity recognition or semantic role labeling (Collobert et al., 2011). Such end-to-end system is capable of learning internal representation directly from the unlabeled data allowing researchers to move away from task-specific, hand-crafted features.

Similar insights are found in the image classification and detection problems. In order to learn about an enormous number of objects from millions of images, we need a model with a large learning capacity. However, the immense complexity of the object recognition task means that this problem cannot be specified only by building such huge training data set, so our model should also have lots of prior knowledge to compensate for all the data we dont have. In particular, a deep convolutional neural network can achieve reasonable performance on hard visual recognition and categorization tasks – matching or exceeding human performance in some domains.

A Convolutional Neural Network (CNN) is a powerful machine learning technique from the field of deep learning. CNNs are trained using large collections of diverse images. Convolutional neural network's capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse (Goodfellow et al., 2016). From these large collections, CNNs can learn rich feature representations for a wide range of images. These feature representations often outperform hand-crafted features such as HOG, LBP, or SURF. An easy way to leverage the power of CNNs, without investing time and effort into training, is to use a pre-trained CNN as a feature extractor for some multiclass linear SVM. This approach to image category classification follows the standard practice of training an off-the-shelf classifier using features extracted from images. For example, the Image Category Classification Using Bag Of Features example uses SURF features within a bag of features framework to train a multiclass SVM. The difference here is that instead of using image features such as HOG or SURF, features are extracted using a CNN. Despite the attractive qualities of CNNs, and despite the relative efficiency of their local architecture, they have still been prohibitively expensive to apply in large scale to high-resolution images and usually, they demand GPUs grids to facilitate the training of interestingly-large CNNs.

Researchers have demonstrated steady progress in computer vision by validating their work against ImageNet[2] – an academic benchmark for computer vision. Successive models continue to show improvements, each time achieving a new state-of-the-art result. ImageNet Large Visual Recognition Challenge is a standard task in computer vision, where models try to classify entire images into 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher". In 2012, an ensemble of CNNs achieved best results on the ImageNet classification benchmark (Krizhevsky et al.,). The authors of winning method trained a large, deep convolutional neural network to classify

[2]http://www.image-net.org/

the millions of high-resolution images in the ImageNet contest into the different classes. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To reduce overfitting in the fully-connected layers recently-developed regularization method called dropout was used. They achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Also in 2012, the biggest NN so far (109 free parameters) was trained in unsupervised mode on unlabeled images data (Le et al., 2012), then applied to ImageNet. They trained a 9-layered locally connected sparse autoencoder with pooling and local contrast normalization on a large dataset of images (the model has 1 billion connections, the dataset has 10 million 200x200 pixel images downloaded from the Internet). The codes across its top layer were used to train a simple supervised classifier, which achieved best results so far on 20,000 classes. Instead of relying on efficient GPU programming, this was done by "brute force" on 1,000 standard machines with 16,000 cores.

So by 2011/2012, excellent results had been achieved by Deep Learners in image recognition and classification. The computer vision community, however, is especially interested in object detection in large images, for applications such as image-based search engines, or for biomedical diagnosis where the goal may be to automatically detect tumors etc in images of human tissue. Object detection presents additional challenges.

3 PREVIOUS WORK

Our previous work is devoted to the issue of short text classification, working on free textual descriptions of books, gathered by crawling the GoodReads portal. Those descriptions are relatively short, often incomplete and sometimes obscured by author's biographic note, which makes genre classification a challenging task. There was a problem with a huge amount of classes with most of them were poorly and unevenly represented. We address these issues more precisely in Data section as it was relevant to this paper as well. We compared two text classification methods in order to choose the best one for this specific task, including baseline naive Bayes models and semantic enrichment method consuming neural-based distributional paragraph models (Doc2Vec from gensim toolkit[3]). The sentence vector based methods – both original Doc2Vec (referred as D2V) and averaged Doc2Vec category vectors (referred as AD2V)– achieved much higher accuracy than the baseline Multinomial Naive Bayes approach (referred as MNB) while requiring less text preprocessing. This difference points to the potential difference in semantic context build around each book genre – something that cannot be captured by a simple Bayesian classifier. The algorithms have been evaluated in terms of the classification quality on the unique data set of almost two hundred thousands book descriptions. Results of previous work are presented in further results section along with current results concerning image-based categorization.

[3]https://radimrehurek.com/gensim/models/doc2vec.html

4 DATA

Figure 1: Examples of book cover images from our dataset.

Dataset consists of information crawled from website GoodReads.com. GoodReads is a website which gathers books-related information like reviews, recommendations, and scores of over 160k books making GoodReads basically the IMDB[4] for books. Most data available is user-generated, including book genre information. For most books textual as well as image data is available. Information about the genre is available as a list of pairs: genre name and number of user votes. The first problem encountered is the huge number of categories which are poorly represented. There are over 500 genres and most of them (over 400) are extremely rare (less than 100 examples). This leads to the need for limiting the number of genres so they have better representation and balance. In order to do so, we decided to assign every book a single "main category" by choosing genre with the most user votes (we refer to this process later as "relabelling"). Later we sorted them by a number of those votes and picked top 13 of them. All of the rest categories we labeled as "Other". We ignored two categories "Fiction" and "Non-Fiction" as they can be considered as taxonomy on a different (higher) level of hierarchy. According to Wikipedia[5] Fiction and Non-Fiction consists of separate genres.

[4]http://www.imdb.com/

[5]https://en.wikipedia.org/wiki/List_of_writing_genres

We are aware of some issues with the approach we have chosen. Firstly we ignore nondominant votes for every book and assume a book has a single category. This is convenient because it simplifies problem to simple classification. The gist of the problem is that categories provided by users are not pairwise disjoint as in classical classification problem. For example, we can imagine a book which is Horror and Romance at the same time. These categories, therefore, could be considered as tags and a single book could have any number of predefined tags. Such problem could be addressed with many binary classifiers, one classifier per tag. This would not be a perfect solution as it only shifts problem somewhere else, namely into settling threshold deciding how much votes is required to assign a category. Another possibility would be to discard some genres and keep only those which cannot overlap. This is not a feasible task to do because almost every real-world book is hybrid of couple well-known genres. Even if it was possible it would be an arduous, manual and error-prone process. Another problem is having genres which are more subgenres than self-contained genres er genres, e.g. Romance and Historical Romance. We didn't flatten those hierarchies mostly to be consistent with our previous work in order to be able to compare results. Described issues (picking a single category and keeping subcategories) are partially addressed by special score function described later in the Evaluation section.

Downloaded images have different sizes and ratios. Most images have portrait ratio (bigger height than width). Covers were scaled to fit 64x96 windows and any empty spaces were filled with black pixels.

As this task requires pictures, records lacking images were removed. This is the reason why number of examples per categories changed since our first paper.

5 APPROACH

In this paper we focused on using convolutional neural network to predict book's genre. We implemented relatively simple and shallow convolutional network using TensorFlow[6] (Abadi et al., 2016). TensorFlow is Google's open-sourced and unopinionated framework for deeplearning. Our network consists of three convolutional layers, each followed directly by 2x2 max-pooling layer with non-overlapping windows. Each convolutional layer has 3x3 kernels but consists of increasingly more features maps: 16, 32 and 64. The stride of convolution in image space domain was set to 1, meaning convolution window overlaps as much as possible. In all layers except last layer Rectified Linear Unit (ReLU) activation function (Glorot et al., 2011) was used as it yield superior performance both during train and test phase. ReLU given by (1)

$$ReLU(x) = max(0,x) \tag{1}$$

is much simpler non-linear transformation than logistic function or hyperbolic tangent but it seems to be common nowadays, especially with GPU computation. After those six layers, two fully connected layers are plugged in. First with 256 neurons and second is softmax (Sutton and Bart, 1998) layer with 14 neurons - typical for classification tasks. Softmax is a function which amplifies maximal signal and dampens others while normalizing outputs in such way that they sum up to 1. Cross entropy error function was minimized during training of the network (2).

$$crossEntropy(X) = -\sum_{i}^{N}\sum_{j}^{L} t_{ij} log(p_{ij}) \tag{2}$$

Where $t_i j$ is equal to 1 if $i-th$ example has class j and 0 otherwise. $p_i j$ denotes probability (according to model) that $i-th$ example has category j. N and L denotes the number of examples in dataset/batch and number of classes/categories respectively. Stochastic gradient descend optimization technique was applied with RMSProp update rule(Tieleman and Hinton, 2012). This network will be referred to as N1.

Second, more sophisticated architecture was used in network N2. The architecture of our second network is inspired by VGG network (Simonyan and Zisserman, 2015). The main idea behind this kind of network is to use several consecutive convolutional layers with small filters (3x3), put max-pooling layers after those convolutions and finish with an optional dropout layer. Such block may be repeated multiple times, usually with increasing number of feature maps. We used blocks consisting of two consecutive convolutional layers with 3x3 filters, 2x2 max-pooling layer with a stride of 2 and finally a dropout with probability parameter set to 0.25. Two of these blocks were stacked sequentially: first with 32 feature maps and second with 64. Dropout is a method focused on preventing overfitting (Hinton et al., 2012; Srivastava, 2013). It is achieved by simply ignoring random part of neurons during training of the model which leads to more independent neurons without "complex co-adaptations". This may be seen as a form of model averaging as every batch is fed into slightly different architecture. After second dropout layer, two dimensional images are flattened into 1d vectors to fit first

[6]https://www.tensorflow.org/

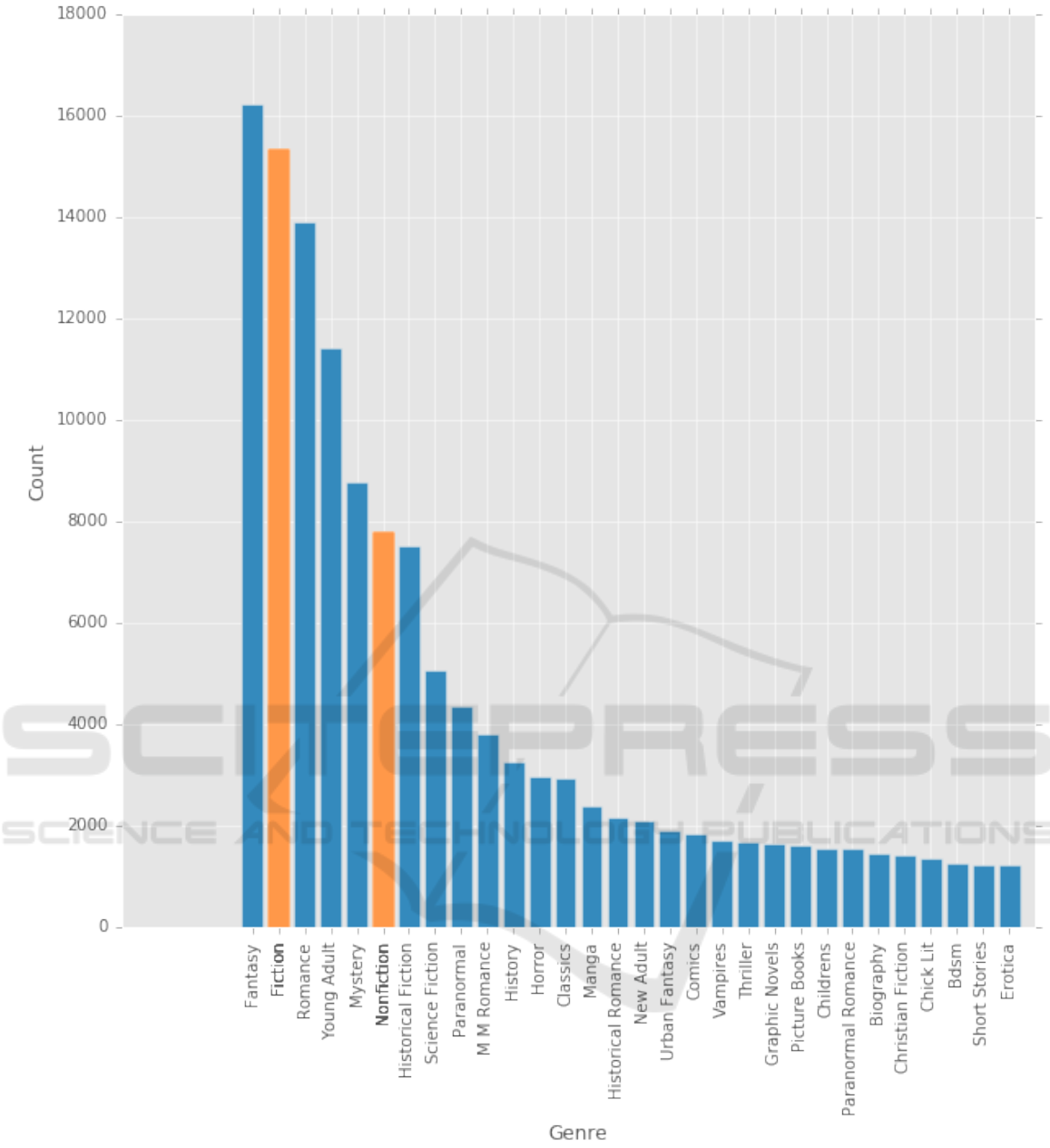

Figure 2: Genre distribution of 30 genres in the analyzed data set before relabelling. It is important to note that Fiction and Non-fiction can be considered book types rather than genres.

of two fully connected layers. The first layer has 256 neurons and second 14. Between these two layers, there is dropout layer with the probability set to 0.5. Second of mentioned fully connected layer uses softmax activations similarly to N1. In this case, we used SGD optimizer with Nesterov momentum (Nesterov, 2004) and small learning rate decay. Momentum methods use information about previous displacements in weight–space and apply them to current weight update which prevents zig-zagging. Nesterov momentum additionally uses special correction term to update rule which provides better convergence. Decay parameter decrease jump size as training progresses which helps to approach minimum with better precision. Except for final softmax layer every layer use ReLU activations.

During the training no data augmentation techniques like rotating, flipping, cropping or elastic trans-

formation were used. Due to data set imbalance batches were prepared in such way that every category is represented by exactly the same number of examples. This way network cannot get biased by mentioned imbalance because it is not able to notice the difference in a priori probabilities of genres.

6 EVALUATION

As previously stated in Data section we did not choose to treat this problem as multi-label classification as it is not obvious where to put threshold for genres. This is the reason why we do not use any of multi-label scores e.g. Hamming loss, Hamming score or multi-label versions of precision and recall. Instead we proposed custom score function which is similar to TOP-k accuracy but uses weights to discount rewards from not ideal prediction. In our solution a book is assigned single dominant genre g_o and classifier returns probabilities for each of 14 genres g_{ci} for $i \in 1,2,3,...14$, where g_{c1} denotes the most probable genre, g_{c2} second most probable genre and so on. Reward from single book is given by s_g (3).

$$s_g = \begin{cases} 1, & \text{if } g_o = g_{c1}. \\ 0.75, & \text{if } g_o = g_{c2}. \\ 0.5, & \text{if } g_o = g_{c3}. \\ 0.0, & \text{otherwise.} \end{cases} \quad (3)$$

Our Score function is a simple average of all s_g across dataset, where X denotes data set and N is a size of that set.

$$Score(X) = \frac{1}{N}\sum_{i=1}^{N} s_g \quad (4)$$

7 RESULTS

Our smaller model N1 managed to achieve better score (0.73) than more complex one N2 (0.68). The accuracies of those models are 0.61 and 0.58 respectively. It is worth noticing that simpler network N1 not only performs better but also converge faster in terms of a number of epochs and time of training. Figure 3 shows the relation between score and epoch number.

We calculated our score function not only for whole test set but also for particular categories. Those scores are presented in Table 1. In that table, we also included results of our previous models.

Table 1: Scores for evaluated classifiers: Multinomial Naive Bayes (MNB), Doc2Vec (D2V) applied to textual descriptions and two CNNs described in this paper N1 and N2 applied to cover images.

Genre	MNB	D2V	N1	N2
All genres	**0.39**	**0.82**	**0.73**	**0.68**
Mystery	0.28	0.84	0.27	0.29
Historical Romance	0	0.91	0.42	0.39
Young Adult	0.53	0.86	0.34	0.41
Science Fiction	0	0.83	0.27	0.24
Horror	0	0.87	0.17	0.10
Paranormal	0.01	0.91	0.19	0.11
Romance	0.71	0.91	0.37	0.45
Fantasy	0.93	0.87	0.44	0.74
Other	0	0.61	0.95	0.83
M M Romance	0	0.96	0.17	0.13
Historical Fiction	0.01	0.77	0.26	0.19
Classics	0	0.81	0.21	0.25
Manga	0	0.85	0.55	0.55
History	0	0.92	0.11	0.14

8 CONCLUSION

Networks achieved accuracy around 60% which can be considered good in a 14-way classification problem. Considering properties of score function, score around 0.75 means that on the average correct genre is second most probable guess which is a pleasing result. This quality level qualifies proposed method to be used in real-world system e.g. supporting workers maintaining books catalog.

It is not surprising that predictions based on textual description are more accurate. There are some keywords with big discriminative power. Such words are likely to appear in some genre while having low probability of occurrence in other genres. For example "frightening", "bloodcurdling" or "chilling" in horror books and "charming" or "lover" in romance books. There is no such "free dinner" in image domain. Covers are much more subtle. Some covers are contradictory or not related to books' content at the first glance, requiring the reader to use complex social constructs or mentally solve some puzzle to understand a concept of the cover. Other covers might be minimalist or mysterious to the level in which they consist of single plain color and small title. In such case human who does not know the language of the title would be equally clueless as a convolutional neural network. We believe that considering relatively small size of our dataset as well as complexity of our visual models network can not spontaneously learn features enabling it to recognize characters (or sequences of characters) and correlate them with genres

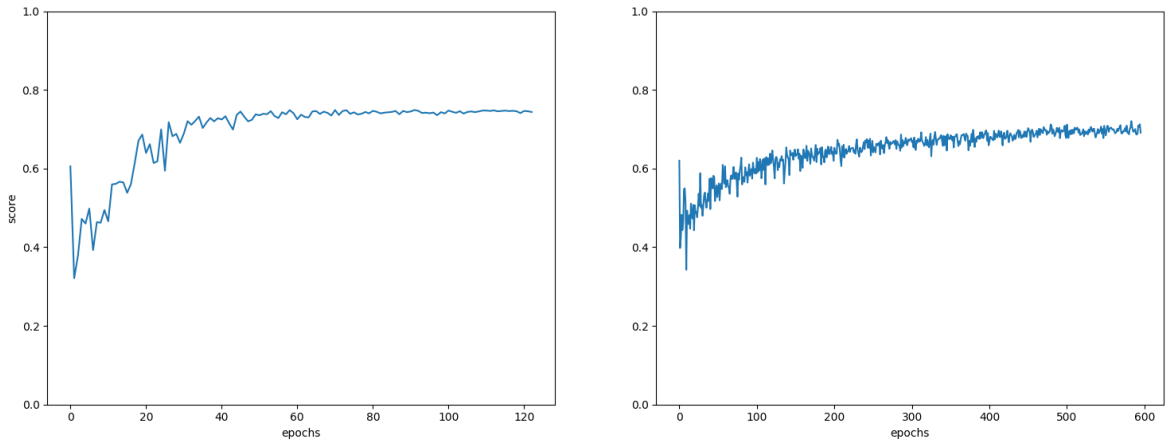

Figure 3: Learning of networks. Left subplot depicts score of N1 network and the right subplot N2.

in a way NLP methods do.

We do not understand why neural networks find some genres much more difficult than others. This differences are much more obvious than in case of NLP approaches. It looks like networks were trained on different number of examples per category and learned a priori distribution of genres. This can not be the case as balancing method described earlier makes sure every genre has the same number of examples in each batch. Although every batch contains same amount of each category, training set remained unbalanced (not trimmed) and therefore there were more examples to sample from for some labels (e.g. other). This could be the reason why decision boundaries in decision space were set to favor "other" as this label would potentially cover greater area. Another explanation for these differences in difficulty may be very nature of cover images. For example intra-genre similarity differs from genre to genre rendering recognizing some of them much more difficult task. This is even more plausible given that "other" examples consists of mixed, very specific genres (e.g. erotica, vampires, biography).

9 FUTURE WORK

We plan to build ensemble classifier based on both textual description and cover image and see if such system built on top of these two existing systems will yield substantially better results than any single system. Right now we are building an online live demo web application which we will share in the nearest future. We also plan to try to use generative models to synthesize textual description from cover images and vice versa.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Adlakha, D., Adlakha, D., and Tanwar, R. (2016). *Analytical Comparison between Sobel and Prewitt Edge Detection Techniques*. International Journal of Scientific & Engineering Research.

Collobert, R., Weston, J., Karlen, L. B. M., Kavukcuoglu, K., and Kuksa, P. (2011). *Natural Language Processing (almost) from Scratch*. Journal of Machine Learning ResearchVolume.

Feichtinger, H. G. and Strohmer, T. (1998). *Gabor Analysis and Algorithms*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. 2015.

Hinton, G., Srivastava, N., and Krizhevsky, A. (2012). Improving neural networks by preventing co-adaptation of feature detectors.

Jones, J. P. and Palmer, L. A. (1987). *An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex*. Journal of Neurophysiology.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. 2012.

Le, Q. V., Ranzato, M., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J., and Ng, A. Y. (2012). Building high-level features using large scale unsupervised learning.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context.

Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization. A Basic Course*. Springer.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Srivastava, N. (2013). Improving neural networks with dropout.

Sutton, R. S. and Bart, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Szegedy, C., Liu, W., and Lia, Y. Going deeper with convolutions. 2015.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning. Technical report.