Pruning of Convolutional Neural Network for Text Classification

Sadique Adnan Siddiqui
Informatik
Technical University
Kaiserslautern, Germany
ssiddiqu@rhrk.uni-kl.de

***Abstract*—In recent years, convolutional neural networks have demonstrated effective performance in natural language processing related tasks. One particular task is text classification where one needs to set predefined categories to free-text documents. In this study, we explore the techniques to prune convolutional kernels in CNN's to reduce the model size with a slight trade-off in the accuracy of the model. The project uses the model of Kim Yoon's Convolutional Neural Networks for Sentence Classification [4]. After training the model, techniques for pruning the not so important filters have been used on the basis of the variance of weights of filters, the sum of the absolute weights of filters, L2 norm of the filters and resetting negative filter weights to zero. The method used in the project prunes the low rating filter followed by fine-tuning the model by adjusting the remaining convolution filters for computational efficiency while minimizing the drop in accuracy.**

I. Introduction

Classifying and grouping documents into their known categories is an important aspect of Document Image Processing Pipeline. The project focuses on document classification from Tobacco-3482 and Corporate Messaging datasets into different predefined categories and then employs different pruning techniques to achieve a trade-off between the size of the trained model and the accuracy achieved by the trained model on particular datasets. Deep Learning approaches based on Convolutional Neural Network has immense success in areas of Image Classification, Natural Language Processing and pattern recognition research [25]. Convolutional neural networks use several processing layers to learn the hierarchical representation of data and thus this is used as an effective feature extractor that captures vital information about the input. In image classification the lower layers of CNN pick up low-level features such as edges from raw pixels, while the higher layer features extracts higher level features that captures complete information, but in natural language processing CNN operates on word embeddings which are mostly continuous representation of the inputs with an aim to capture extract features (such as phrases and relationships between words in the sentence). The model used in this project is the same as Kim Yoon's Convolutional Neural Networks model for Sentence Classification. After training the model, techniques for filter pruning has been applied on the basis of variance, absolute weights, L2 norm of filters and removing negative weights of filters. The low rating filter is then removed, followed by fine tuning with the remaining filters.

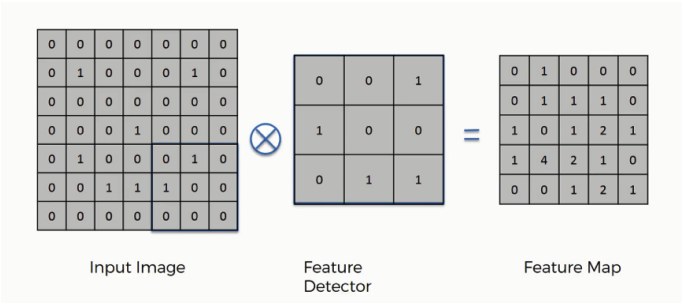

Fig. 1. The Convolution operation using 3*3 filter. Source: superdatascience.com

II. Method

A. Classification using Convolutional Neural Network

ConvNets were initially developed in the neural network image processing community where they achieved breakthrough results in recognizing an object from a pre-defined category. A Convolutional Neural Network typically involves two operations, which can be thought of as feature extractors: convolution and pooling. It is built on the idea of applying a sliding window (convolution filter) over an image matrix. Each image is considered to be a matrix where each entry represents each pixel. The convolution operation is performed by placing the convolution filter or kernel on the top of the input image and then multiplying the values of the input image matrix with the corresponding values in the convolution filter. After that nonlinear layers such as Relu or tanh is applied over the outputs received from convolution layers. In comparison to a fully connected neural network where each neuron is connected to all other neurons of the next layer, each neuron is connected to only a local region of the input volume. The max pooling operation is performed to downsample the image representation, where the maximum of the values in the input feature map region is taken that results in significant size reduction in output size. In the case of NLP tasks, when applied to text instead of images text data is represented in the form of a matrix. Each row of the matrix is composed of different tokens which are represented

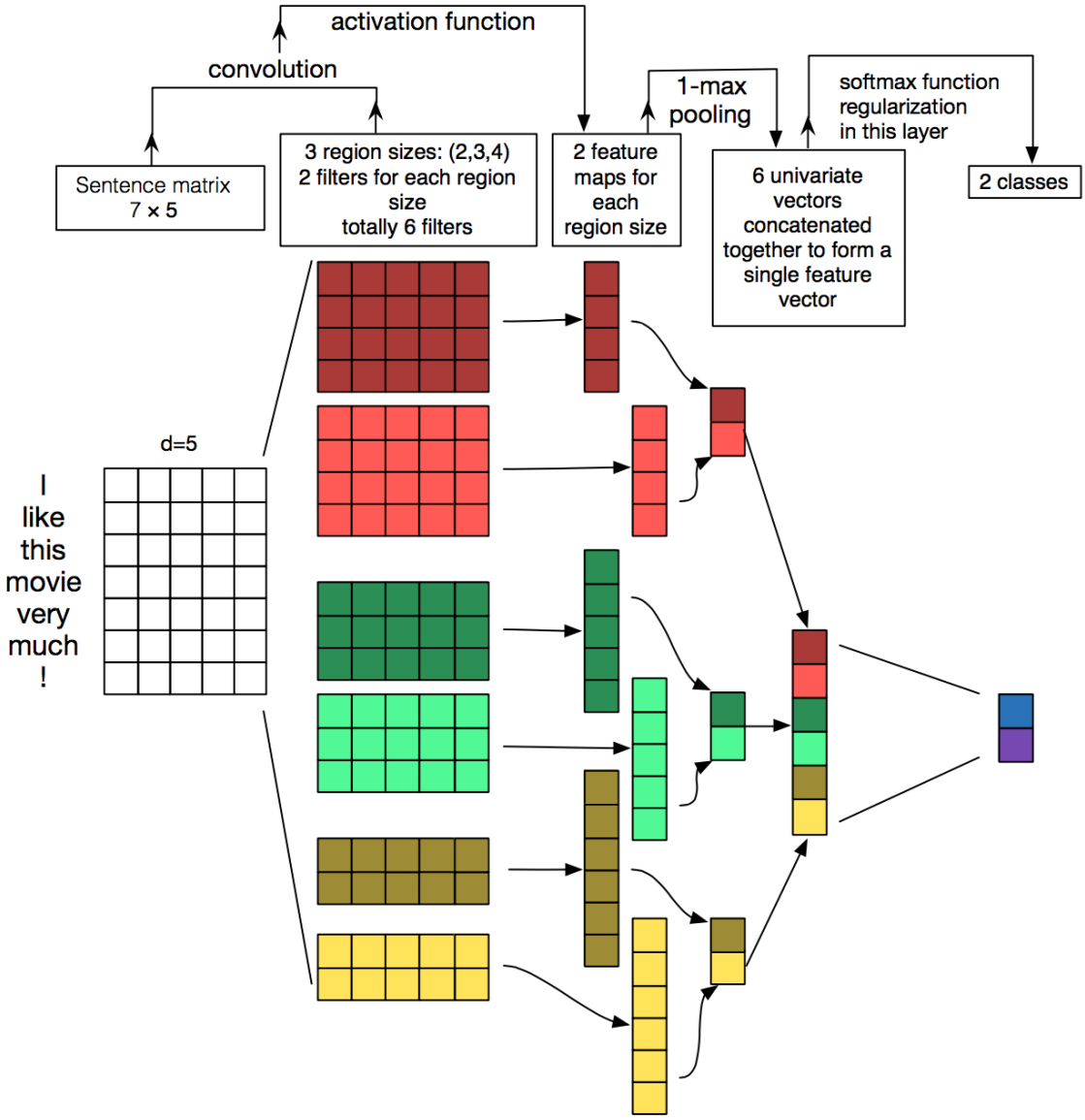

Fig. 2. Illustration of a CNN architecture for text classification. Source: Zhang, Y., Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners Guide to) Convolutional Neural Networks for Sentence Classification.

by vectors consists of word embeddings. In the convolution layer, filters of multiple sizes slide over the rows of matrices. The result of the convolutional layer into a long feature vector on max pooling, then add dropout regularization and classify the result using a softmax layer.

Firstly, the tesseract is used on the tobacco datasets where images of different classes are converted to text files that contain all the extracted image data related to particular classes. Then data is processed and all unnecessary characters are removed. After preprocessing, tokenization is done where long strings of text are split into smaller tokens and the next step is to convert all the generated tokens into vectors of real numbers of a particular length. In this project, two techniques have been used while training the model (i) where word embeddings are learned from scratch (ii) where pre-trained word2vec vectors have been used for word embeddings. The data is divided into training, validation and test data. The first layer defined is the embedding layer, which maps vocabulary word indices into low-dimensional vector representations. Its essentially a lookup table that we learn from data. The next layer performs convolutions over the embedded word vectors

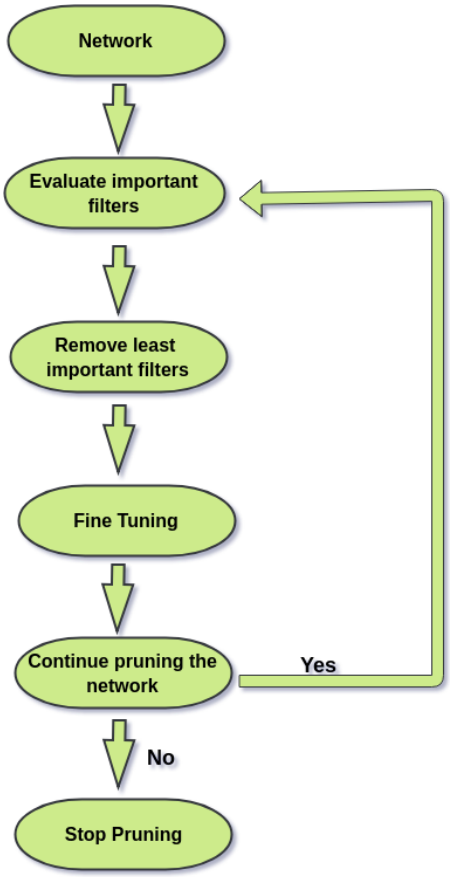

Fig. 3. Flowchart of iterative pruning technique

using multiple filter sizes. In this project, we have used filters of sizes 3,4,5 sliding over (3,4,5) words at a time. [4]. Next, we max-pool the result of the convolutional layer, resulting in a scalar value corresponding to the maximum filter response of the activation map. All the scalar values are concatenated to get a long feature vector after that softmax layer is used to classify the result. The model is evaluated after every 100 steps and the parameters of the model are saved after the specified number of a step so that it could be restored later.

B. Pruning Technique

The process of removing the less contributing weights to compress the network is called pruning. After some weights have been pruned, the network has to be fine-tuned again to adapt the changes. This idea was first proposed by Yann Le Cunn et.al in their famous paper called Optimal Brain Damage (OBD) [26] where the particular trainable parameters were set to zero by measuring the contribution of that parameter on the training error. There have been many pruning approaches used in 2D or 3D Convolutional neural network but still, there is a lot of scopes for exploring these new pruning techniques in a 1D convolutional neural network. There are four pruning techniques that have been used in the project. 1) Calculating the variance of each filter weights and then removing the filters with the lowest variance score, which implies that filter with less variance does not have much discriminative ability in classification tasks. After removing the filters, alternate iterations of pruning and fine-tuning is done and this process is stopped after reaching the target trade-off between accuracy and pruning objective.

2. The relative importance of filters is measured by calculating the absolute weight of the filters for each filter sizes. The absolute weight of the filters of each layer also represents the average magnitude of its weights. So this gives us the idea that the output feature map and filters with less absolute weights will also have the activation maps weak activations as compared to the other filters in those layers.

3. Calculating the L2 norm of each filter weights and then removing the filter will low norm. The motivation behind this pruning technique is that the filters with less L2 norm as compared to other filters detect less important features.

4. In this technique, we select a random number of filters and weights with negative values of that filter in each layer has been reset to zero and then the model is fine-tuned. Apart from pruning filters for each layer, we have reduced word embeddings dimensions as well.

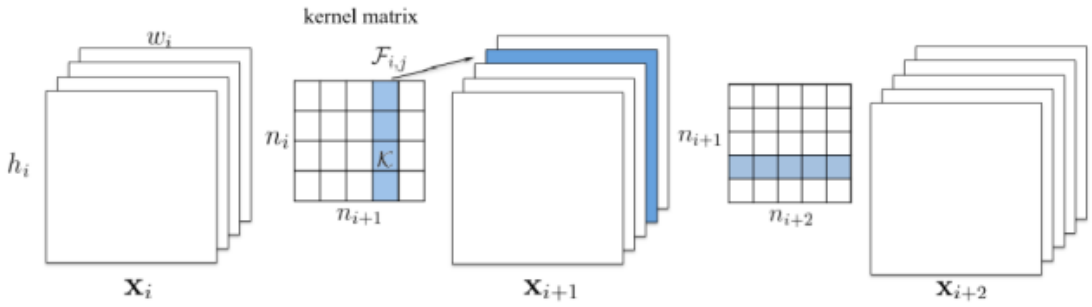

Fig. 4. Pruning a filter results in removal of its corresponding feature map and related kernels in the next layer. Source: Hao Li et al. Pruning filters for efficient convnets

III. Experimental Setup

A. Datasets

The datasets used are publicly available Tobacco-3482 dataset. It consists of images related to tobacco from the media. It has 10 classes and 3400 documents images. As tobacco datasets contain too many overlapped classes i.e. predicted classes are being confused with true classes, Corporate messaging datasets concerning what corporations actually talk about on social media have also been used. It has 3 classes which include statements as information (objective statements about the company or it's activities), dialog (replies to users, etc.), or action (messages that ask for votes or ask users to click on links, etc. [27]

B. Data Preparation

Preparation of data is a vital step for any experiment. The tobacco dataset contains images with .tif extension and has a label associated with it. The first step is to convert document image into a text file using tesseract, which is an optical character recognition software that recognizes text characters in images. OCR is a machine-learning technique used to transform images that contain text into actual text content. After getting the text contents from all the images. Text data are split for training, validation and testing data in the ratio 7:2:1 then the data is preprocessed by removing punctuations and extra spaces.

C. Experiments

Several experiments have been conducted on both the datasets. Firstly, we trained the convolutional neural network on tobacco datasets and Corporate messaging datasets where embeddings learned from scratch. After that different pruning techniques have been employed after restoring the model on both the datasets. As tobacco datasets are imbalance datasets and contains too many overlapped classes we repeated the same experiments by taking subsets of tobacco datasets containing data of three nonoverlapping classes.

Secondly, we trained a convolutional neural network on tobacco datasets and Corporate messaging datasets where pretrained word embeddings from fasttext were used.

We have used Tensorflow which is an open source framework developed by Google researchers to run machine learning, deep learning models. For Tobacco datasets, we trained the model for 20 Epochs using Adam Optimizer with a learning rate of 0.001. For Corporate messaging datasets, we have trained the model for 50 Epochs using using Adam Optimizer with a learning rate of 0.001. The results of all the experiments conducted have been mentioned in later section.

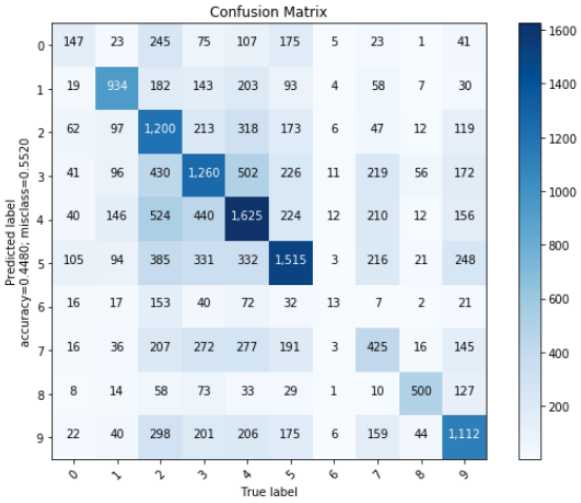

Fig. 5. Confusion Matrix for Tobacco Datasets

Tobacco-3482 datasets		
	Accuracy	Model Size(MB)
Without Pruning	0.44795	34.9
Pruning filters by calculating variance and removing filters with low variance		
Pruning 30% of filters	0.44805	29.9
Further pruning 30% of filters	0.44292	20.7
Pruning filters by calculating absolute weights		
Pruning 30% of filters	0.440203	29.9
Further pruning 30% of filters	0.43972	20.7
Pruning filters by calculating L2 norm		
Pruning 30% of filters	0.439895	29.9
Further pruning 30% of filters	0.43719	20.7
Pruning filters by resetting negative weights to zero		
Resetting all negative weights to zero	0.447181	34.9

(a) Results when model is trained on Tobacco datasets and embeddings are trained from scratch

Tobacco-3482 datasets (3 classes)		
	Accuracy	Model Size(MB)
Without Pruning	0.826714	15
Pruning filters by calculating variance and removing filters with low variance		
Pruning 30% of filters	0.825669	11.3
Further pruning 30% of filters	0.822324	7.7
Pruning filters by calculating absolute weights		
Pruning 30% of filters	0.817099	11.3
Further pruning 30% of filters	0.814381	7.7
Pruning filters by calculating L2 norm		
Pruning 30% of filters	0.818771	11.3
Further pruning 30% of filters	0.806856	7.7
Pruning filters by resetting negative weights to zero		
Resetting all negative weights to zero	0.821488	15

(b) Results when model is trained on Tobacco datasets (3 classes) and embeddings are trained from scratch

Corporate Messaging Datasets		
	Accuracy	Model Size(MB)
Without Pruning	0.882927	5.4
Pruning filters by calculating variance and removing filters with low variance		
Pruning 30% of filters	0.882927	4
Further pruning 30% of filters	0.876423	2.7
Pruning filters by calculating absolute weights		
Pruning 30% of filters	0.878049	4
Further pruning 30% of filters	0.871545	2.7
Pruning filters by calculating L2 norm		
Pruning 30% of filters	0.884553,	4
Further pruning 30% of filters	0.873171	2.7
Pruning filters by resetting negative weights to zero		
Resetting all negative weights to zero	0.879675	5.4

(a) Results when model is trained on Corporate messaging datatsets and embeddings are trained from scratch

Tobacco-3482 datasets		
	Accuracy	Model Size(MB)
Without Pruning	0.41224	1.8
Pruning filters by calculating variance and removing filters with low variance		
Pruning 30% of filters	0.397363	1.3
Further pruning 30% of filters	0.391922	0.9
Pruning filters by calculating absolute weights		
Pruning 30% of filters	0.397876	1.3
Further pruning 30% of filters	0.389716	0.9
Pruning filters by calculating L2 norm		
Pruning 30% of filters	0.40157	1.3
Further pruning 30% of filters	0.392386	0.9
Pruning filters by resetting negative weights to zero		
Resetting all negative weights to zero	0.40275	1.8

(b) Results when model is trained on Corporate messaging datatsets and embeddings are trained from scratch

Tobacco-3482 datasets (3 classes)		
	Accuracy	Model Size(MB)
Without Pruning	0.810619	1.8
Pruning filters by calculating variance and removing filters with low variance		
Pruning 30% of filters	0.794941	1.3
Further pruning 30% of filters	0.786162	0.90
Pruning filters by calculating absolute weights		
Pruning 30% of filters	0.794523	1.3
Further pruning 30% of filters	0.790343	0.90
Pruning filters by calculating L2 norm		
Pruning 30% of filters	0.797032	1.3
Further pruning 30% of filters	0.794314	0.90
Pruning filters by resetting negative weights to zero		
Resetting all negative weights to zero	0.799958	1.8

(a) Results when model is trained on Tobacco datasets(3 classes) and pretrained word embeddings are used

Corporate Messaging Datasets		
	Accuracy	Model Size(MB)
Without Pruning	0.892683	1.8
Pruning filters by calculating variance and removing filters with low variance		
Pruning 30% of filters	0.871545	1.3
Further pruning 30% of filters	0.868293	0.9
Pruning filters by calculating absolute weights		
Pruning 30% of filters	0.879675	1.3
Further pruning 30% of filters	0.873575	0.9
Pruning filters by calculating L2 norm		
Pruning 30% of filters	0.878049	1.3
Further pruning 30% of filters	0.874797	0.9
Pruning filters by resetting negative weights to zero		
Resetting all negative weights to zero	0.863415	1.8

(b) Results when model is trained on Corporate Messaging datasets when pretrained word embeddings are used

IV. Results and Tables

We have classified the different categories of Tobacco-3482 dataset and corporate messaging dataset using a word based CNN classifier. The results contains the validation accuracy and the reduction in model size after applying different pruning techniques on both the datasets.

V. Future Work

In recent years, there have been various pruning and compression techniques for a convolutional neural network for image classifications. There are many methods which could be possibly used in a 1D convolutional neural network similar to 2D or 3D convolutional network such as the use of new criterion based on Taylor expansion that approximates the change in the cost function induced by pruning network parameters. Other techniques include binarization where the weights are restricted to be either -1 or +1. [20]

Conclusion

The report focuses on the pruning of Convolution neural network for text classification for better task performance. First of all, we have trained the convolutional neural network for text classification, where two datasets (Tobacco and Corporate Messaging) have been classified into their predefined classes. After that, we have used the trained model for pruning. We have used four different pruning techniques and found that there is an optimal reduction in the size of the model after pruning iteratively with a minimum drop in the accuracy.

References

[1] H. Yoon, S. Robinson, J. B. Christian, J. X. Qiu and G. D. Tourassi, Filter pruning of Convolutional Neural Networks for text classification: A case study of cancer pathology report comprehension, 2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), Las Vegas, NV, 2018, pp. 345-348.

[2] Nidhi Kamath, Cannannore Bukhari, Syed Dengel, Andreas. (2018). Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification. 1-11.

[3] Molchanov, Pavlo et al. Pruning Convolutional Neural Networks for Resource Efficient Transfer Learning.

[4] Kim Y. Convolutional Neural Networks for Sentence Classification. 2014.

[5] Kalchbrenner, N., Grefenstette, E., Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. Acl, 655665.

[6] Santos, C. N. dos, Gatti, M. (2014). Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In COLING-2014 (pp. 6978).

[7] Johnson, R., Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. To Appear: NAACL-2015, (2011).

[8] Zhang, Y., Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners Guide to) Convolutional Neural Networks for Sentence Classification,

[9] Nguyen, T. H., Grishman, R. (2015). Relation Extraction: Perspective from Convolutional Neural Networks. Workshop on Vector Modeling for NLP, 3948.

[10] Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., Wang, X. (2015). Modeling Mention , Context and Entity with Neural Networks for Entity Disambiguation, (Ijcai), 13331339.

[11] Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. Coling, (2011), 23352344.

[12] Gao, J., Pantel, P., Gamon, M., He, X., Deng, L. (2014). Modeling Interestingness with Deep Neural Networks.

[13] Zhang, X., Zhao, J., LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification, 19.

[14] Zhang, X., LeCun, Y. (2015). Text Understanding from Scratch. arXiv E-Prints, 3, 011102.

[15] Santos, C., Zadrozny, B. (2014). Learning Character-level Representations for Part-of-Speech Tagging. Proceedings of the 31st International Conference on Machine Learning, ICML-14(2011), 18181826.

[16]] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 10971105, 2012.

[17] Martn Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In OSDI, volume 16, pages 265283, 2016.

[18] http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

[19] https://cv-tricks.com/tensorflow-tutorial/save-restore-tensorflow-models-quick-complete-tutorial/

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

[21] Courbariaux, M., David, J. (n.d.). BinaryConnect : Training Deep Neural Networks with binary weights during propagations https://arxiv.org/pdf/1511.00363.pdf

[22] Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Il, R. T. A., Bengio, Y., Com, Y. U. (n.d.). Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or 1. Retrieved from

[23] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y. (2016). Quantized Neural Networks Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Acthttps://data.world/crowdflower/corporate-messagingivations.

[24] Krizhevsky, A., Sutskever, I., Hinton, G. E. (n.d.). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[25] Imagenet classification with deep convolutional neural networks A Krizhevsky, I Sutskever, GE Hinton Advances in neural information processing systems, 1097-1105

[26] Yann Le Cun, John S Denker, and Sara A Solla. Optimal Brain Damage. In NIPS, 1989.

[27] https://data.world/crowdflower/corporate-messaging