Journal of Digital Imaging (2021) 34:1349–1358
https://doi.org/10.1007/s10278-021-00518-2

Determining Top Fully Connected Layer's Hidden Neuron Count for Transfer Learning, Using Knowledge Distillation: a Case Study on Chest X-Ray Classification of Pneumonia and COVID-19

Ritwick Ghosh[1]

Received: 28 April 2020 / Revised: 20 February 2021 / Accepted: 14 September 2021 / Published online: 29 September 2021

Abstract
Deep convolutional neural network (CNN)-assisted classification of images is one of the most discussed topics in recent years. Continuously innovation of neural network architectures is making it more correct and efficient every day. But training a neural network from scratch is very time-consuming and requires a lot of sophisticated computational equipment and power. So, using some pre-trained neural network as feature extractor for any image classification task or "transfer learning" is a very popular approach that saves time and computational power for practical use of CNNs. In this paper, an efficient way of building full model from any pre-trained model with high accuracy and low memory is proposed using knowledge distillation. Using the distilled knowledge of the last layer of pre-trained networks passes through fully connected layers with different hidden layers, followed by Softmax layer. The accuracies of student networks are mildly lesser than the whole models, but accuracy of student models clearly indicates the accuracy of the real network. In this way, the best number of hidden layers for dense layer for that pre-trained network with best accuracy and no-overfitting can be found with less time. Here, VGG16 and VGG19 (pre-trained upon "ImageNet" dataset) is tested upon chest X-rays (pneumonia and COVID-19). For finding the best total number of hidden layers, it saves nearly 44 min for VGG19 and 36 min and 37 s for VGG16 feature extractor.

Keywords Deep learning · COVID-19 · Pneumonia · Chest X-ray · Medical imaging · Knowledge distillation · Transfer learning · Image classification

Introduction

Pneumonia is the single leading cause of childhood mortality. About 2 million children under 5 years old, die due to pneumonia every year around the world according to the World Health Organization (WHO). It kills more children than AIDS, malaria and measles [1]. Almost 95% of childhood pneumonia cases occur in developing countries, specifically in Africa and Southeast Asia. Pneumonia caused by bacterial pathogens needs urgent medical appointment with immediate antibiotic treatment and caused by viral pathogens needs supportive cares [2]. On the other hand, on the 11th of March, 2020 WHO declares COVID-19 (Coronavirus disease) a pandemic. Alarming rate of spreading, highly infectious attitude and severity of this disease make it a worldwide concern. The main popular screening method for detecting it is polymerase chain reaction (PCR) [3]. The general objective of PCR testing is to detect SARS-CoV-2 RNA from respiratory specimens collected through a variety of resources like nasopharyngeal or oropharyngeal swabs. PCR testing is highly sensitive and gold standard. PCR testing is very time-consuming, laborious, complicated and needs manual processing. An alternative way of detecting COVID-19 is analysing chest radiography images such as X-ray or computed tomography (CT) images. Patients of COVID-19 shows significance abnormality in chest radiology imaging [4, 5]. In epidemic areas, radiology imaging as screening tool might help [6]. Coronavirus disease (COVID-19) is an infectious disease and looking at the extent of its spread throughout the world, it has been declared as a pandemic by the World Health Organization (WHO) on the 11th of March, 2020 [7].

✉ Ritwick Ghosh
ritwickghosh2000@gmail.com

[1] Department of Mining Engineering, Indian Institute of Engineering Science and Technology, Shibpur, P.O, Botanical Garden, Howrah, West Bengal 711103, India

An experienced radiologist might easily identify normal, viral pneumonia, bacterial pneumonia and newly emerged COVID-19 X-ray images. But requirement of deep knowledge about anatomical principles, pathology and physiology, specially the variety of pathogens make it a complex job for non-experienced personals. Computer-aided diagnostic (CAD) can possibly aid radiologists and physicians to more swiftly and precisely classify radiography images to detect pneumonia and COVID-19 cases. In this paper a deep neural network is trained using transfer learning process to classify pneumonia and COVID-19 from radiology images. Even though deep neural networks fetch efficient models and achieve satisfying results, they are naturally too large, time-consuming and need much complex computational power to be deployed on edge devices like smartphones or embedded sensor nodes. There have been many efforts to compress these networks or so-called models, and a popular and appealing method is knowledge distillation (KD), where a large pre-trained network (teacher model) is used to train a smaller network (student model). In this test the learned soft labels extracted from a pre-trained network over an image input are supplied into the fully connected dense layers to determine the probability of the image over the possible classes. The soft labels are extracted using knowledge distillation process for each image.

Background

Algorithmic Approach to Image Analysis (Traditional and Neural Network)

Traditional algorithmic approach for classifying images involves handcrafted image segmentation and creating shallow neural machine learning classifier to classify according to classes [8]. Instead of this complex and expensive process [9–11], development of deep convolutional neural network layers and models makes the process automatic and more accurate [12, 13]. Every layer of deep convolutional neural network (CNN) consists of multiple convolutions or systematically image analysis filters that eventually produce a feature map for input of the next layer. The network architecture allows to process the input image as a form of pixels and executes the classification accordingly. In modern process of image classification, one classifier could outperform the traditional multiple classifier, and need of handcrafting is also decreased in a significant level. Convolutional neural networks or CNNs are that kind of artificial neural networks that use multiple perceptron or statistical classifiers to analyse inputs (Image or arrays) and learn weights and bases to multiple parts of images and able to separate each other. One benefit of using CNN is it leverages the use of local spatial coherence in the input images, which allow them to have fewer weights as some parameters are shared. So, using CNN is clearly efficient in terms of memory and complexity.

Medical Image Analysis using Neural Networks

Deep learning or convolutional neural networks show a significant increase in performance in the medical image analysis domain for automated object or disease identification and segmentation operation. Recent distinguished work involves but does not limit to an outline analysis on the forthcoming promise of deep learning [14] and an assembly of significant medical applications on.

- cell image segmentation and tracing [15],
- cerebral microbleed detection [16],
- pulmonary nodule detection in CT images [17],
- lymph node and interstitial lung disease detection and classification [18, 19],
- predicting spinal radiological scores [20] extensions of multi-modal imaging segmentation [21, 22],
- automated pancreas segmentation [23],
- and COVID-19 classification using radiology imaging [24, 25].

But on the other hand, it is unclear how well or efficiently the current deep learning techniques will handle multiple thousands of patient data.

Distilling Neural Networks

Neural networks use distributed representation of feature map of input to interpret and processing in hidden layers [26]. These representations are hard to understand and represent in general form. Knowledge distillation (KD) was first applied on deep neural networks by Hinton et al. [27]. KD was first proposed by Bucila et al. [28].

Knowledge distillation is the process to transfer the representations and abilities learned by a large network (teacher network) to any smaller network (student network). Main drawback of knowledge distillation is that it can only be applied for classification tasks, not for regression tasks [29]. KD uses the "dark knowledge" (softened logit output of the bottom output layer of teacher network) that is transferred to student network. This dark knowledge is more than inter-label correlations and one-hot encoding of labels. In case of regression, the deep network interprets the continuous values, which has a tendency of unknown error distribution; as a result, there is no dark knowledge for deep networks trained for regression. But later [30] present their knowledge distillation process for pose regression. Multiple papers try to propose a framework to train the teacher and student network in parallel. Yim et al. [31] and You et al. [32] proposed a shared layer-representation for it. Czarnecki et al.

[33] narrowed down the variance between teacher and student derivatives of the loss shared with the discrepancy from teacher predictions. Tarvainen and Valpola [34] proposed an averaging procedure for model weights for the same purpose. Urban et al. [35] trained a network of convolutional neural networks and teach shallow multilayer perceptron as student networks. Sau and Balasubramanian [36] injected noise into teacher logits for making the student network more robust. Employing several teacher networks is always a way to decrease the accuracy difference between teacher and student network. Zhang et al. [37] proposed deep mutual learning so that teacher and student networks can learn side by side during training.

Hinton et al. [27] claim that the success of knowledge distillation is credited to the logit distribution of the incorrect outputs. Furlanello et al. [38] investigated the success of knowledge distillation via gradients of the loss. Lopez-Paz et al. [39] considered the cause behind the efficiency of knowledge distillation from the perception of learning theory by Vapnik [40] by reviewing the estimation error in empirical risk minimization framework.

Other Approaches for Compressing

Obtaining smaller network with least comptonization with respect to accuracy is one of the hottest topics in research. A variety of approaches, other than distillation include shrinking, factorizing or compressing pre-trained deep neural networks. Neural network compression using product quantization [41] network pruning [42–45], connectivity learning [46, 47], hashing [48] and vector quantization-Huffman encoding [49] have been proposed. Also a few factorizations are also proposed to speed up the network's prediction [50, 51]. Another proved approach is using low bit networks to train the deep learning model [52–54].

Dataset

In this study, mainly three types of X-ray images are used, namely normal chest X-rays, pneumonia X-ray images and COVID-19 chest X-ray images. Normal and pneumonia images are part of dataset and are 1583 and 4273 in number, respectively. That dataset [55] is already tested by Kermany et al. [56] which achieved an accuracy of 92.8%, with a sensitivity of 93.2% and a specificity of 90.1% over pneumonia classification. COVID-19 X-ray images was taken from [57] and are 262 in number. Total number of images is 5964 dividing into three classes. Figure 1 shows examples of each of the three types of images.

Process

Transfer Learning

In this paper, transfer learning is employed to tackle the problem of detecting the class of chest X-ray images. Due to the immensity and complication of convolutional neural network architectures, scheming and testing models could be expensive and time-consuming. When approaching an image classification, quick and more efficient results can be achieved by applying a technique known as transfer learning. In transfer learning, the weights and convolutional filters that are capable at one task (trained for any classification task), is reused for another task needing only a small amount of retraining and can be trained or tested on a little number of images or data. This includes using a pre-trained neural network architecture with pre-loaded weights, adjusting it to some extent, and then retraining part or the whole model to perform the new task. The filters learned by one task are used to extract features from images that can then be interpreted by the retrained portion of the neural network in order to perform its new task. In this paper, the deep convolutional neural network known as VGG [58] is used to study transfer learning using pre-training over "Imagenet" dataset [59], and weights are used the same as [58].

Teacher Network Architecture

VGG [23] is a deep neural network architecture developed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group at Oxford. The "16" and "19" in the names

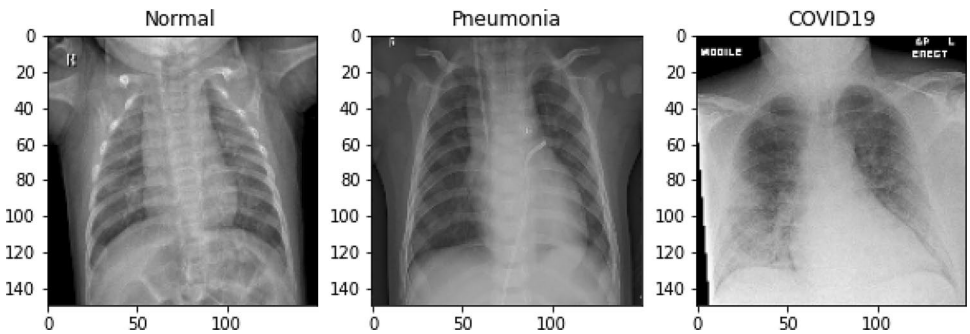

Fig. 1 Sample data cases (normal, pneumonia and COVID-19 cases)

(VGG16 and VGG19) refer to the number of weight layers in the network. VGG16 is very successful in the ImageNet competition in 2014. VGG16 architecture has three convolutional layers in the next three stacks, and VGG19 has four convolutional layers in the next three stacks of layers. These are our main feature extractor convolutional architecture of our study. Then the networks are followed by flatten layer. After that it passes through a fully connected dense layer of hidden layers. And last after using dropout of 20%, it passes through a softmax layer to fetch the classification probabilities.

Convolution is a specialized kind of linear operation. In convolutional layer, a matrix named kernel is passed over the input matrix to create a feature map for the next layer. At every location, an element wise matrix multiplication is performed and sums the result onto the feature map. The rectified linear unit activation function (ReLU) is a piecewise linear function that will output the input if it is positive, otherwise it will output zero. Down sampling of convolutional layers is achieved by applying a pooling layer after non-linearity layer. Pooling helps make the representation become approximately invariant to small translations of the input. Softmax or normalized exponential function normalizes the input vector according to probability distribution to the exponentials of inputs.

The input images resolution is (150×150×3). Every convolutional layer has filters of (3×3) shape with ReLU activation. Max-pooling layers would take (2×2) matrix. These layers would get the precise locations of the feature map on the activated images. Flatten layer convert every array into one dimensional array. After flattening, the nodes that are achieved from flattening, will act as input layer to fully connected dense layer. These layers also have the same type of rectified activation as convolutional layers. It follows with dropout layer and softmax layer. The model is compiled with Kingma and Ba [60] optimizer, with learning rate 0.0001. Tables 1 and 2 contain the detailed architectures of the both teacher models (VGG16 and VGG19) with their training details and outputs.

Table 1 VGG16 model architecture

Layer type	Output tensor shape (per image)	Status
Input layer	150, 150, 3	New
Convolutional (2D)	150, 150, 64	Frozen
Convolutional (2D)	150, 150, 64	Frozen
Max-pooling (2D)	75, 75, 64	Frozen
Convolutional (2D)	75, 75, 128	Frozen
Convolutional (2D)	75, 75, 128	Frozen
Max-pooling (2D)	37, 37, 128	Frozen
Convolutional (2D)	37, 37, 256	Frozen
Convolutional (2D)	37, 37, 256	Frozen
Convolutional (2D)	37, 37, 256	Frozen
Max-pooling (2D)	18, 18, 256	Frozen
Convolutional (2D)	18, 18, 512	Frozen
Convolutional (2D)	18, 18, 512	Frozen
Convolutional (2D)	18, 18, 512	Frozen
Max-pooling (2D)	9, 9, 512	Frozen
Convolutional (2D)	9, 9, 512	Frozen
Convolutional (2D)	9, 9, 512	Frozen
Convolutional (2D)	9, 9, 512	Frozen
Max-pooling (2D)	4, 4, 512	Frozen
Flatten	8192	New
Dense/fully connected	4096	New
Dropout (20%)	4096	New
Softmax	3	New

Student Network Architecture

Student networks are dense neural network models with input dimension same as the logit output shape of the specific layer. The main model is a fully connected dense layer with variable number of filters. In this study, number filters are altered, and the model accuracy is checked. The number of filters in dense network is called the "hidden neuron" count of the student model. Student model size depends upon the filter number (hidden neuron number) and input shape. The accuracy of the model depends upon the accuracy and learning of the teacher model. The dense network has passed through dropout rate of 0.2 to nullify overfitting. At last, the model is compiled using Adadelta [61] optimizer with unity learning rate and 0.95 decay factor.

Distillation

Neural networks typically yield class-probabilities using a softmax layer. Softmax output layer converts the logit, z_i, calculated for each class into a probability, q_i, by comparing z_i with the other logits.

$$q_i = (e^{\frac{z_i}{T}}) / \sum_j (e^{\frac{z_j}{T}}) \quad (1)$$

where T is a constant known as temperature that is conventionally agreed as the value 1.

In the conventional procedure of distillation, knowledge or learning is shifted to the smaller model by training it on a transfer set. The inputs of the distilled model or smaller model are soft target distribution for each case in the transfer set. Hinton et al. [27] uses a higher temperature value (higher than 1) in softmax to produce soft target distribution of the transfer set that is produced by using the cumbersome model. The same high temperature is retained when training the smaller model, but after it would be trained using a temperature of 1.

Table 2 VGG19 model architecture

Layer type	Output tensor shape (per image)	Status
Input layer	150, 150, 3	New
Convolutional (2D)	150, 150, 64	Frozen
Convolutional (2D)	150, 150, 64	Frozen
Max-pooling (2D)	75, 75, 64	Frozen
Convolutional (2D)	75, 75, 128	Frozen
Convolutional (2D)	75, 75, 128	Frozen
Max-pooling (2D)	37, 37, 128	Frozen
Convolutional (2D)	37, 37, 256	Frozen
Convolutional (2D)	37, 37, 256	Frozen
Convolutional (2D)	37, 37, 256	Frozen
Convolutional (2D)	37, 37, 256	Frozen
Max-pooling (2D)	18, 18, 256	Frozen
Convolutional (2D)	18, 18, 512	Frozen
Convolutional (2D)	18, 18, 512	Frozen
Convolutional (2D)	18, 18, 512	Frozen
Convolutional (2D)	18, 18, 512	Frozen
Max-pooling (2D)	9, 9, 512	Frozen
Convolutional (2D)	9, 9, 512	Frozen
Convolutional (2D)	9, 9, 512	Frozen
Convolutional (2D)	9, 9, 512	Frozen
Convolutional (2D)	9, 9, 512	Frozen
Max-pooling (2D)	4, 4, 512	Frozen
Flatten	8192	New
Dense/fully connected	4096	New
Dropout (20%)	4096	New
Softmax	3	New

Each case in the transfer set contributes a cross-entropy gradient, dC/dz_i, with respect to every logit, z_i of the distilled model. If the cumbersome model has logits v_i which produce soft target probabilities p_i, and the transfer training is done at a temperature of T, this gradient is given by:

$$\frac{dC}{dz_i} = (q_i - p_i)(1/T) = (1/T)((e^{\frac{z_i}{T}})/\sum_j (e^{\frac{z_i}{T}}) - (e^{\frac{v_i}{T}})/\sum_j (e^{\frac{v_j}{T}})) \quad (2)$$

Using a higher value for T produces a softer probability distribution over classes. For high temperature value, knowledge distillation is equivalent to minimizing $1/2(z_i - v_i)^2$, provided the logits are zero mean separately for each transfer case. At lower temperatures, distillation wages much less consideration to alike logits that are much more negative than the average.

Results

Parameter count of convolutional neural network model signifies the size of the model. Accuracy of the model equals to the ratio of the correctly predicted images and total images in every test. Recall of cases signifies the ratio of correctly predicted images and the total images under that specific case. Covariance is calculated as the average of the product between the values from each sample, where the values had their mean subtracted. The Pearson's correlation coefficient is determined as the ratio of covariance of the two variables divided and the product of the standard deviation of each data sample. It is the normalization of the covariance between the two variables to give an interpretable score. The total sample of 5964 images is divided into two sets randomly. Four thousand eight hundred 90 images containing all the classes are used for training, and 1074 images are used to test the accuracy of the model in every test case. The main architectures are trained 30 epochs for every different hidden neuron count in the fully connected layer. The main teacher models are trained 30 epochs according to architectures (hidden neuron count varies in fully connected layer) in Tables 1 and 2. The parameter count, accuracy and recall of largest teacher CNNs are mentioned in Table 3. The student architectures are trained 40 epochs for each test. Tables 4 and 5 note all the accuracies of the main architectures (VGG16 and VGG19 respectively) and student distilled architectures with respect to the changing number of hidden layers in the dense layer. Total time taken for the tests on main VGG19 models is 52 min,, 29 s and using distillation process is 8 min, 28 s. Total time taken for the tests on main VGG16 models is 45 min, 28 s and using distillation process with student networks is 8 min, 51 s. Table 6 shows the covariance and Pearson's correlation in between the accuracies of the two main networks with student accuracies. Figures 2 and 3 show the graphical representation of the main model and student model accuracies for VGG16 and VGG19, respectively. Figures 4 and 5 show these results of the main model accuracy with student model accuracy in scatter plot.

Table 3 Details of teacher model test

CNN name	CNN characteristics	Parameters (millions)	Accuracy (%)	Recall of normal cases	Recall of pneumonia cases	Recall of COVID-19 cases
VGG16	Dense layer with 4096 hidden neurons	43.29	96.37	0.93	0.97	0.91
VGG19	Dense layer with 4096 hidden neurons	53.6	96.28	0.94	0.96	0.96

Table 4 Accuracy of different VGG16 models

Number of neurons in fully connected dense layer	Parameters for VGG16-type model (millions)	Accuracy for VGG16-type model (%)	Student accuracy for VGG16-type model (%)
4096	48.29	96.37	96.18
3072	39.89	96.46	96.28
2048	31.50	96.46	96.28
1024	23.10	96.28	95.99
512	18.91	96.46	96.28
256	16.81	96.46	96.28
128	15.76	96.55	96.37
64	15.24	96.65	96.37
32	14.98	96.28	95.99
16	14.85	96.28	95.16
8	14.78	96.18	94.41

Table 5 Accuracy of different VGG19 models

Number of neurons in fully connected dense layer	Parameters for VGG19-type model (millions)	Accuracy for VGG19-type model (%)	Student accuracy for VGG16-type model (%)
4096	53.60	96.28	95.90
3072	45.20	96.28	96.18
2048	36.81	96.46	96.37
1024	28.42	96.65	96.37
512	24.22	96.46	96.37
256	22.12	96.65	96.37
128	21.07	96.46	96.37
64	20.55	96.65	96.55
32	20.29	96.46	96.28
16	20.16	96.28	94.69
8	20.09	93.76	90.50

Table 6 Covariance and Pearson's correlation between accuracies

	Covariance	**Pearson's correlation**
VGG16 (main model vs. student model)	0.066	0.774
VGG19 (main model vs. student model)	1.433	0.975

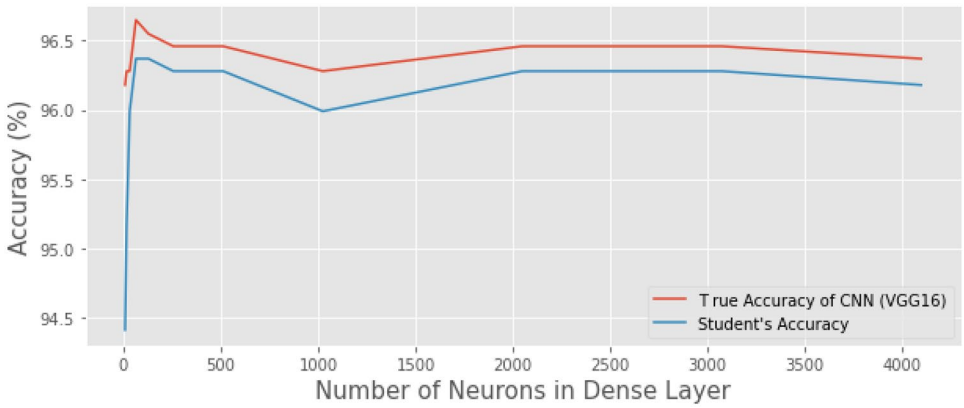

Fig. 2 Main models and student model accuracy graph for VGG16

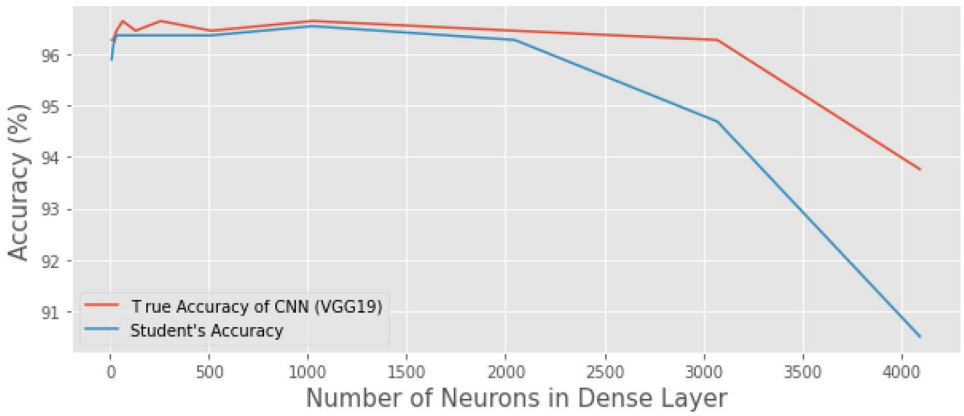

Fig. 3 Main model and student model accuracy graph for VGG19

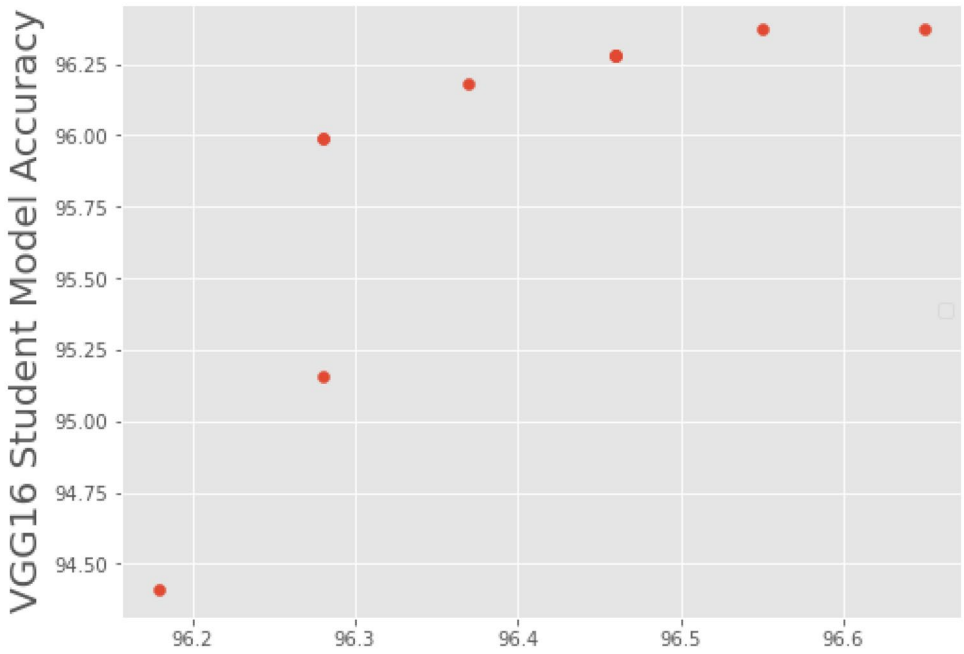

Fig. 4 Main model vs. student model accuracy (VGG16) scatter plot

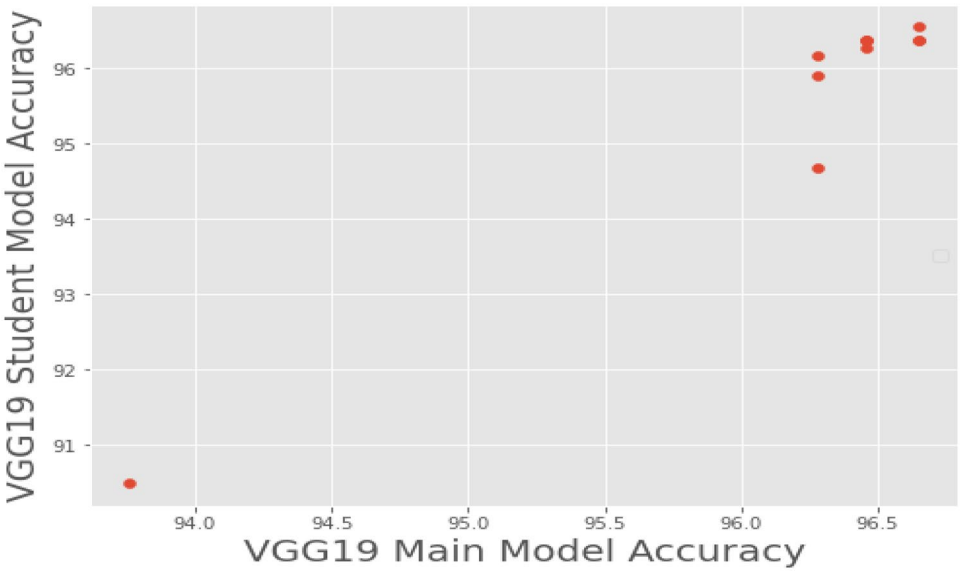

Fig. 5 Main model vs. student model accuracy (VGG19) scatter plot

Discussion and Conclusion

Table 6 covariance is positive in every case, so the accuracies of main model and student model are changing in the same direction. Pearson's correlation value for both is more than 0.5. This suggests a high level of correlation, e.g. a value above 0.5 and close to 1.0. So, the accuracy of the student and main architecture is slightly different from each other, but linearly correlate-able with each other. That is how we can find the best hidden neuron number for our dense layer using this technique that is saving nearly 44 min for VGG19 and 36 min, 37 s for VGG16 feature extractor. We can use any pre-trained feature extractor (even if it is new architecture built for any specific task or trained upon any other dataset than "ImageNet") and find the optimal number of hidden layer in top dense layer as it performs the best on that task. Determining the best number of neurons, not only shorten the memory necessity of the deep learning model but also increases the accuracy, by nullifying the option of overfitting over that data. Using the same task manually, as the ideal number of hidden neurons will differ from architecture to architecture and dataset to dataset, will take a long time. But this process does not give the perfect best accuracy; the accuracy of student models is slightly less than that of the main model; but as the study suggests, these student accuracies will help to roughly interpret the best accuracy. On the other hand, this process might be slow in nullifying the overfitting than pruning, but it offers better flexibility and better relative accuracy than pruning as it helps to rebuild the whole network the best way possible. But this process can delete the overfitting on the top layers; pruning can delete overfit for multiple layers at once.

Declarations

Conflict of Interest The author declares no competing interests.

References

1. Adegbola, R.A.: Childhood pneumonia as a global health priority and the strategic interest of the Bill & Melinda Gates Foundation. Clin. Infect. Dis. 54(Suppl 2), S89–S92. 2012
2. Mcluckie, A.: Respiratory disease and its management, Volume 57 (Springer). 2009
3. Wang et al.: Detection of SARS-CoV-2 in different types of clinical specimens. JAMA, 2020
4. Huang et al.: Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. The Lancet, 395(10223), 2020
5. Ng et al.: Imaging profile of the COVID-19 infection: radiologic findings and literature review. Radiology: Cardiothoracic Imaging, 2(1), 2020
6. Ai et al.: Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. Radiology, 2020
7. WHO: Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. Available at https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-atthe-media-briefing-on-covid-19---11-march-2020. Accessed 25th April, 2020
8. Goldbaum, M., Moezzi, S., Taylor, A., Chatterjee, S., Boyd, J., Hunter, E., and Jain, R.: Automated diagnosis and image understanding with object extraction, object classification, and inferencing in retinal images. Proceedings of 3rd IEEE International Conference on Image Processing 3, 695–698, 1996
9. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., and Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Trans. Med. Imaging 8, 263–269, 1989
10. Hoover, A., Kouznetsova, V., and Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Trans. Med. Imaging 19, 203–210, 2000

11. Hoover, A., and Goldbaum, M.: Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels. IEEE Trans. Med. Imaging 22, 951–958, 2003
12. Zeiler, M.D., and Fergus, R.: Visualizing and understanding convolutional networks. Lect. Notes Comput. Sci. 8689, 818–833, 2014
13. Krizhevsky, A., Sutskever, I., and Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 84–90, 2017
14. Greenspan, H., van Ginneken, B., Summers, R.M.: Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. IEEE Trans. Medical Imaging, 35(5):1153–1159, 2016
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In MICCAI, pages 234–241. Springer, 2015
16. Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V., Shi, L., Heng, P.: Automatic detection of cerebral microbleeds from MRI images via 3D convolutional neural networks. IEEE Trans. Medical Imaging, 35(5):1182–1195, 2016
17. Setio, A., Ciompi, F., Litjens, G., Gerke, P., Jacobs, C., van Riel, S., Wille, M., Naqibullah, M., Snchez, C., van Ginneken, B.: Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. IEEE Trans. Medical Imaging, 35(5):1160–1169, 2016
18. Roth, H.R., Lu, L., Seff, A., Cherry, K.M., Hoffman, J., Wang, S., Liu, J., Turkbey, E., Summers, R.M.: A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations. In MICCAI, pages 520–527. Springer, 2014
19. Shin, H., Roth, H., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.: Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learnings. IEEE Trans. Medical Imaging, 35(5):1285–1298, 2016
20. Jamaludin, A., Kadir, T., Zisserman, A.: Spinenet: automatically pinpointing classification evidence in spinal MRIS. In MICCAI. Springer, 2016
21. Moeskops, P., Wolterink, J., van der Velden, B., Gilhuijs, K., Leiner, T., Viergever, M., Isgum, I.: Deep learning for multi-task medical image segmentation in multiple modalities. In MICCAI. Springer, 2016
22. Havaei, M., Guizard, N., Chapados, N., Bengio, Y.: Hemis: hetero-modal image segmentation. In MICCAI, pages (2): 469–477. Springer, 2016
23. Roth, H., Lu, L., Farag, A., Shin, H.C., Liu, J., Turkbey, E.B., Summers, R.M.: Deeporgan: multi-level deep convolutional networks for automated pancreas segmentation. In MICCAI, pages 556–564. Springer, 2015
24. Goze et al.: Rapid AI development cycle for the coronavirus (COVID-19) pandemic: initial results for automated detection and patient monitoring using deep learning CT image analysis. arXiv:2003.05037, 2020
25. Wang, L., Wong, A.: COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images. arXiv:2003.09871, 2020
26. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature, 521 (7553):436–444, 2015
27. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In NIPS Deep Learning and Representation Learning Workshop (2015), pages 1–9, 2015
28. Bucila, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In SIGKDD, 535–541. ACM, 2006
29. Cheng, Y., Wang, D., Zhou, P., Zhang, T.: Model compression and acceleration for deep neural networks: the principles, progress, and challenges. IEEE Signal Processing Magazine, 35(1):126–136, 2018
30. Saputra, M.R.U., de Gusmao, P.P., Almalioglu, Y., Markham, A., Trigoni, N.: Distilling knowledge from a deep pose regressor network, The IEEE International Conference on Computer Vision (ICCV), pp 263–272, 2019
31. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: fast optimization, network minimization and transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4133–4141, 2017
32. You, S., Xu, C., Xu, C., Tao, D.: Learning from multiple teacher networks. In SIGKDD, 1285–1294. ACM, 2017
33. Czarnecki, W., Osindero, S., Jaderberg, M., Swirszcz, G., Pascanu, R.: Sobolev training for neural networks. In NIPS, 4278–4287, 2017
34. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight averaged consistency targets improve semi-supervised deep learning results. In NIPS, 2017
35. Urban, G., Geras, K.J., Kahou, S.E., et. al.: Do deep convolutional nets really need to be deep and convolutional?, In 5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings, 2017
36. Sau, B., Balasubramanian, V.: Deep model compression: distilling knowledge from noisy teachers. arXiv preprint arXiv:1610.09650, 2016
37. Zhang, Y., Xiang, T., Hospedales, T., Lu, H.: Deep mutual learning. CoRR abs/1706.00384, 2017
38. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. arXiv preprint arXiv:1805.04770, 2018
39. Lopez-Paz, D., Bottou, L., Sch¨olkopf, B., Vapnik, V.: Unifying distillation and privileged information. arXiv preprint arXiv:1511.03643, 2015
40. Vapnik, V.: Statistical learning theory. 1998, volume 3. Wiley, New York, 1998
41. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices. arXiv preprint arXiv:1512.06473, 2015
42. Han, S., Pool, J., Tran, J., Dally, WJ.: Learning both weights and connections for efficient neural network. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada, pages 1135–1143, 2015
43. Han, S., Pool, J., Narang, S., Mao, H., Tang, S., Elsen, E., Catanzaro, B., Tran, J., William, J., Dally, J.:. DSD: regularizing deep neural networks with dense-sparse-dense training flow. CoRR, abs/1607.04381, 2016
44. Guo, Y., Yao, A., Chen, Y.: Dynamic network surgery for efficient DNNs. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5–10, 2016, Barcelona, Spain, pages 1379–1387, 2016
45. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf HP.: Pruning filters for efficient convnets. CoRR, abs/1608.08710, 2016
46. Ahmed, K., Torresani, L.: Connectivity learning in multi-branch networks. CoRR, abs/1709.09582, 2017
47. Veniat, T., Denoyer, L.: Learning time efficient deep architectures with budgeted super networks. CoRR, abs/1706.00046, 2017
48. Chen, W., Wilson, J.T., Tyree, S., Weinberger, KQ., Chen, Y.: Compressing neural networks with the hashing trick. CoRR, abs/1504.04788, 2015
49. Han, S., Mao, H., Dally, WJ.: Deep compression: compressing deep neural network with pruning, trained quantization and Huffman coding. CoRR, abs/1510.00149, 2, 2015
50. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866, 2014

51. Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., Lempitsky, V.: Speeding-up convolutional neural networks using fine-tuned CP-decomposition. arXiv preprint arXiv:1412.6553, 2014
52. Courbariaux, M., David, J.P., Bengio, Y.: Training deep neural networks with low precision multiplications. arXiv preprint arXiv:1412.7024, 2014
53. Rastegari, M., Ordonez, M., Redmon, J., Farhadi. Xnornet, A.: Imagenet classification using binary convolutional neural networks. arXiv preprint arXiv:1603.05279, 2016
54. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: training neural networks with low precision weights and activations. arXiv preprint arXiv:1609.07061, 2016
55. Kermany, D., Zhang, K., Goldbaum, M.: Labeled optical coherence tomography (OCT) and chest X-ray images for classification, Mendeley Data, v2, https://doi.org/10.17632/rscbjbr9sj.2 (2018)
56. Kermany, D.S., Goldbaum, M., Cai, W., et al.: Identifying medical diagnoses and treatable diseases by image-based deep learning, Cell, Volume 172, Issue 5, 22 February 2018, Pages 1122-1131.e9 https://doi.org/10.1016/j.cell.2018.02.010
57. Cohen, J.P., Morrison, P., Dao, L.: COVID-19 image data collection, arXiv:2003.11597, 2020 https://github.com/ieee8023/covid-chestxray-dataset
58. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition, arxiv preprint: arXiv:1409.1556 [cs.CV], 2015 v6
59. Russakovsky*, O., Deng*, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. IJCV, 2015
60. Kingma, D.P., Ba, J.: Adam: A large scale visual recognition challenge, arXiv preprint: arXiv:1412.6980v9 [cs.LG], 2017(v9)
61. Zeiler, M.D.:: ADADELTA: An adaptive learning rate method, arXiv preprint: arXiv:1212.5701 [cs.LG], 2012