

Knowledge Distillation from Multiple Teachers using Visual Explanations

MEHAK

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology



Department of Computer Science and Engineering

June 2018

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Mehak

(Signature)


(MEHAK)

CS 16 MTECH 11008

(Roll No.)


Approval Sheet

This Thesis entitled Knowledge Distillation from Multiple Teachers using Visual Explanations by MEHAK is approved for the degree of Master of Technology from IIT Hyderabad



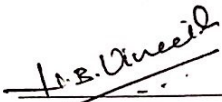
Manish Singh) Examiner

Department of Computer Science And Engineering
IITH



Sriji P.K.) Examiner

Department of Computer Science And Engineering
IITH



Vineeth N. Balasubramanian) Adviser

(Vineeth N. Balasubramanian) Adviser
Department of Computer Science And Engineering
IITH

(——) Co-Adviser

Department of Computer Science And Engineering
IITH



Manendra Sanjay Desai) Chairman

Department of Computer Science And Engineering
IITH

Acknowledgements

I would like to thank my thesis adviser, Dr. Vineeth N. Balasubramanian for his guidance and consistent efforts that helped me complete this work. He was very supportive and understanding throughout my research work. I would also like to thank the Computer Science and Engineering department at IIT Hyderabad for providing the resources needed to complete this work. I am thankful to Bhavana Jain, undergraduate in IIT Hyderabad for the assistance she provided in this work.

I would like to thank my seniors and friends for the encouragement during my research. I would like to express my gratitude to my parents for providing me with the support and continuous motivation throughout my years of study.

Abstract

Deep neural networks have exhibited state-of-the-art performance in many computer vision tasks. However, most of the top-performing convolutional neural networks(CNN) are either very wide or deep which makes them memory and computation intensive. The main motivation of this work is to facilitate the deployment of CNNs on portable devices with low storage and computation power which can be done with model compression. We propose a novel method of knowledge distillation which is a technique for model compression. In knowledge distillation a shallow network is trained from the softened outputs of the deep teacher network. In this work, knowledge is distilled from multiple deep teacher neural networks to train a shallow student neural network based on the visualizations produced by the last convolutional layer of the teacher networks. The shallow student network learns from the teacher network with the best visual explanations. The student is made to mimic the teacher's logits as well as the localization maps generated by the Grad-CAM(Gradient-weighted Class Activation Mapping). Grad-CAM takes the last convolutional layer gradients to generate the localization maps that explains the decisions made by the CNN. The important regions are illuminated in the localization map which explains the specific class predictions made by the network. Training the student with visualizations of the teacher network helps in improving the performance of the student network because the student mimics the important portions of the image learned by the teacher. The experiments are performed on CIFAR-10, CIFAR-100 and Imagenet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) for the task of image classification.

Contents

Declaration	ii
Approval Sheet	iii
Acknowledgements	iv
Abstract	v
Nomenclature	vii
1 Introduction	1
1.1 Introduction to deep learning	1
1.1.1 Multi-Layer Neural Network	1
1.1.2 Convolutional Neural Network	3
1.2 Model Compression	4
1.3 Interpretability of deep learning models	5
2 Related Work	7
2.1 Model Compression	7
2.2 Knowledge Distillation	8
3 Proposed Methodology	9
3.1 Review of Knowledge Distillation	9
3.2 Gradient-weighted Class Activation Mapping (Grad-CAM)	11
3.3 Pipeline of the Proposed Method	13
3.3.1 Drawback of the proposed method	15
3.3.2 Multiple Teacher network visualizations	16
4 Experimental Results	18
4.1 CIFAR-10	18
4.1.1 Analysis Of Knowledge Distillation from Multiple teachers	19
4.2 CIFAR-100	20

4.3	Imagenet	21
4.3.1	Analysis Of Knowledge Distillation from Multiple teachers	22
4.4	Analysis on CIFAR-10	23
4.4.1	Effect of λ and β	23
4.4.2	Effect of Temperature	23
5	Conclusions	25
	References	26

Chapter 1

Introduction

1.1 Introduction to deep learning

Deep learning is one sub branch of machine learning which uses artificial neural networks for decision making, whose functioning resembles the structure of human brain. Deep models are called 'deep' because the hierarchical structure of neural networks comprises of a lot of hidden layers. Deep learning models have achieved lot of success over machine learning algorithms in wide variety of applications mainly in NLP and computer vision tasks. Unlike earlier machine learning algorithms neural networks build its feature set itself without any supervision from the user but it requires a huge dataset. The major limitation of deep learning models is it requires lot of training data and computation power i.e high performing GPUs. Without much training data it is quite difficult to train the deep model to generalize well. Neural networks uses the hierarchical function and processes the data non-linearly.

1.1.1 Multi-Layer Neural Network

A simple neural network consists an input layer, hidden layer and output layer of neurons. As shown in the Figure:1.1 the nodes represents the neurons in the network and neurons are connected in such a way that the output of one neuron is the input of another neuron in the next layer. The input layer takes the raw training data and passes it to hidden layers for further computations. There can be multiple hidden layers but here only one hidden layer is shown. In the input layer x_1, x_2, x_3 represents the training samples and $+1$ is a bias term 'b'. 'W' and 'b' are the two parameters of the network, where 'W' refers to the connection weights which are initialized randomly by the network and are changed as the learning proceeds.

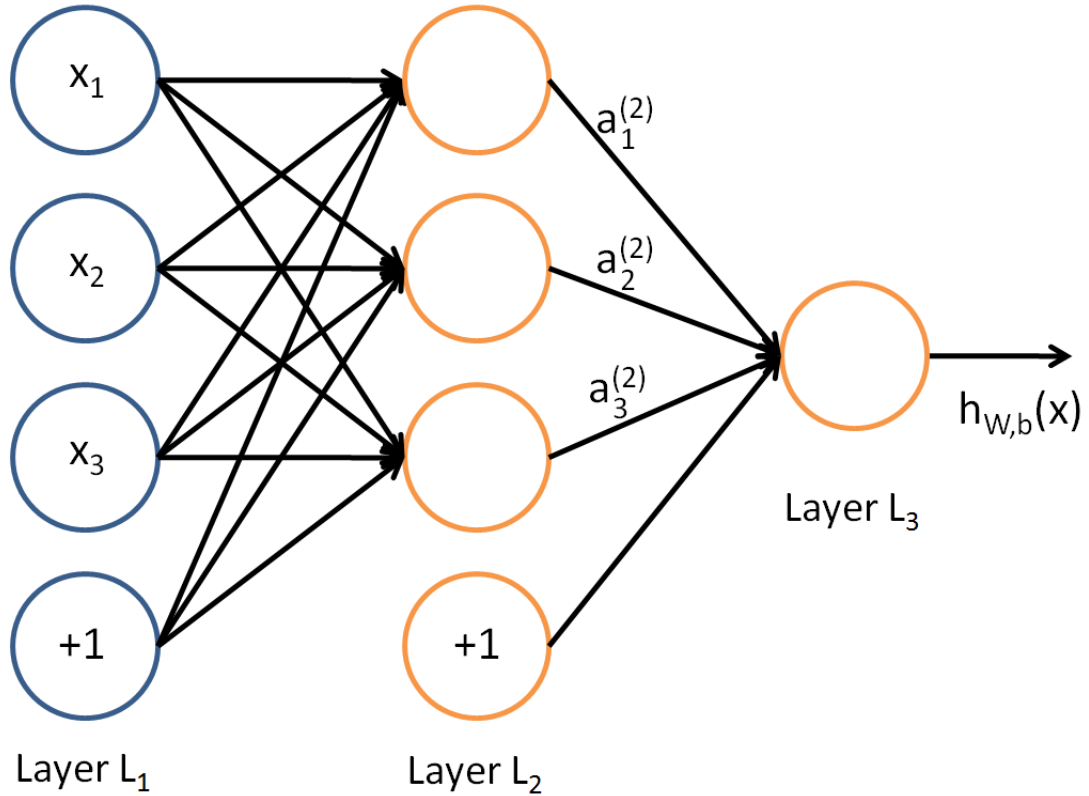


Figure 1.1: An example of Multi-layer Perceptron

In each layer the neuron value is multiplied with the weights of all the connected neuron of the previous layer separately and the the value are summed up and passed through activation function to obtain the output value from the neuron Eq (1.1), this is called forward propagation step in the neural network. Eq (1.1) shows the output value of a single neuron from the hidden layer and f is the activation function.

$$a_1^2 = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \quad (1.1)$$

The activation functions are used to introduce non-linearity in the network. Commonly used activation functions are: sigmoid, tanh and rectified linear functions defined in equations (1.2), (1.3) and (1.4).

$$f(z) = \frac{1}{1 + \exp(-z)} \quad (1.2)$$

$$f(z) = \tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \quad (1.3)$$

$$f(z) = \max(0, x) \quad (1.4)$$

Backpropagation:

First we perform the forward propagation step in the neural network and compute the output activation of all the neurons in the network. Then the error is computed on each output neuron by comparing the desired output with the obtained output from the network. Then we optimize the overall cost function of the network with the help of Gradient descent or any other optimization algorithm. After computing the gradients the weight and bias parameters of the network are updated.

1.1.2 Convolutional Neural Network

Unlike normal neural network convolutional neural network does not have all fully connected connections between the neurons. It has multiple kind of layers i.e. convolution layer, pooling layer and fully connected layer. The normal neural net with all fully connected connections cannot be used with images of large sizes because it will result in large number of parameters.

The convolution layers shares the parameters as the weights filters are multiplied with the regions of the input image. If the filter size is 5x5 then the parameters are shared with that input portion of the image and this reduces the number of parameters in the network. These filters are convolved through the whole image to produce feature maps. Suppose there are 'n' filters then the number of feature maps produced will be 'n'. Mximum number of floating point operation of the network are performed in convolution layers. When there are many convolution layers in the network the initial layers learns the general image features like edges, corners etc in the first few iterations and the image specific features are learned in the later convolution layers in the last few iterations.

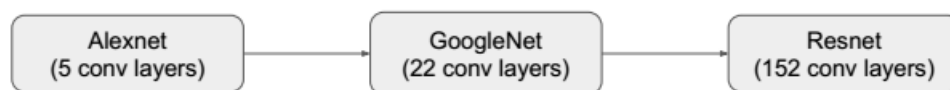
Pooling layer down samples the output from the convolution layer. Pooling layer reduces the number of parameters of the network by down sampling the size of the image because if the same dimension image is passed to the full connected layers the

total number of parameters will increase drastically in the whole network resulting in over-fitting in the network. The depth of the network remains unchanged after the pooling operation as it pools across each channel separately. Pooling layer does not contain any trainable parameters so it does not increase the network parameters. There are different kind of pooling average pooling, max pooling, mean pooling so any of these can be used according to the requirements.

Fully connected layer comprises of the maximum number of parameters in the network since all the neurons in the fully connected layer are connected with all the neurons in the previous layer. The last fully connected layer has the number of neuron equal to the number of classes in the dataset and it produces the prediction score for each class.

1.2 Model Compression

Deep neural networks have exhibited state-of-the-art performance in many computer vision tasks. But all the top-performing networks have huge number of layers and parameters so it requires lots of storage and computation power. The neural networks are made complex and deep to improve the performance of the networks.



Few networks like VGGNet even occupies more than 500MB of storage space and the number of parameters in VGG16 are 139M and Alexnet 60M. So the neural networks are considered as both memory and computation intensive. It is difficult to deploy these networks to deploy on small portable devices which have constraints on storage and battery power. Neural networks with large number of parameters requires large number of floating point operations to be performed during its training which in turn requires a lot of battery power.

Neural networks cannot be used in real life applications because of its complexity. In order to deploy these networks on mobile devices there is a need of model compression. In convolutional neural network the full connected layers consists of the maximum number of parameters and the convolution layers has to perform the maximum number of floating point operations because of the matrix multiplications. Many methods have been proposed recently to compress the neural networks. Few methods focus on

reducing the storage complexity of these models and some of them try to reduce the number of computations and some of them attempts to achieve network speedup.

Main challenge in deep model compression is to compress the network without drop in accuracy. There has to be trade off between the compression ratio and the performance of the deep network.

There are many compression methods introduced so far which includes, pruning the redundant connections, quantization of weights, HashedNets, matrix factorization, designing of compact architectures such as Network in Network architecture and GoogLeNet, binarizing the network weights and activations and training the shallow networks(Knowledge Distillation).

In this work, we have extended Knowledge Distillation method for compressing the model. In this method a smaller network is trained from the scratch to perform the same task as performed by the deep model or the ensemble of models. Teacher-student framework was first proposed by [1] where the shallow network mimics the pre-softmax outputs(logits) of the deep network. Knowledge Distillation was proposed by [2] where knowledge is distilled using the softened outputs of the deep model instead of the pre-softmax outputs. The outputs before the softmax layer are divided by temperature and then passed through the softmax function to achieve softened probabilities. The method in [2] performs better than just using logits[1]. The distilled knowledge is often referred to as 'dark knowledge'.

We extend this knowledge distillation method and propose the idea of training a shallow network from multiple deep networks and their representations. The training from teacher network representations helps in improving the quality of student training since student learns from the important portions of the visualizations which the teacher network has learned.

1.3 Interpretability of deep learning models

Interpretability is an important factor to understand the predictions given by the neural network. Neural networks are widely used in the field of NLP and computer vision so there is a need for transparency in the decisions made by the neural networks in order to develop trust in their predictions. The top performing neural networks with very deep architectures are more complex which makes these networks even harder to interpret.

Neural networks are like black boxes so its preferred to have reasonable explanations for its decisions to develop the trust of the users. Interpretability is important when

deploying the neural networks in some high risk environment where the mistakes made by the neural network might result in severe consequences. With the explanations may be we get extra knowledge that in turn helps to improve the performance of a top performing network and provide insights in the failure of few models.

We used Grad-CAM [3] to interpret the decisions made by the network with the help of visualizations of the learned representations of the neural network. With the help of these visualizations it becomes easy to interpret the final predictions of the model and also the reason of predictions. Grad-CAM generates the localization maps by computing the gradients from the last convolution layer of the network so this approach does not require any architectural changes in the network or any kind of retraining.

Grad-CAM generates class discriminative localization maps which explains the decision of the network. It is class discriminative since it localizes the target class in the visualization by illuminating the relevant portions of the target class in the image. This approach is a generalization of [4] which is also used widely but this requires changes to the architectures.

Chapter 2

Related Work

2.1 Model Compression

There are many methods that have been proposed for model compression so far.

Compressing the pre-trained networks

HashedNets [5] compresses the network by reducing the number of parameters and for that they have used a low-cost hash function that group the connections in a hash bucket which shares a single parameter value within one bucket.

[6] reduced the number of bits required to represent each weight and quantized the parameters so that multiple connections share the same weights. [7] introduced a method to prune the redundant weights and connections from the network. After pruning the connection the model is retrained to finetune the remaining weights. This was extended and quantization and huffman coding was applied to further reduce more number of parameters.

[8], [9] used matrix factorization methods to reduce the number of floating point operations in convolution layers and achieve speed-up.

Designing new compact networks

Compact architectures like Network in Network architecture [10], GoogLeNet [11] and Residual-Net [12] are designed. In NIN the fully connected layers are replaced with global-average pool layer and in ResNets the number of parameters are reduced with the introduction of 1x1 convolutions.

Binarizing the network

BinaryConnect[13] binarizes the weights of neural network. The real valued weights are not used during the update but are retained for the computation of the gradients.

[14] Binarizes both the weights and the activations in neural networks. [15] Approximates the convolution operation by binarizing the weights so the computations are drastically reduced since the weights are binarized. This method outperforms both the BinaryConnect and BinaryNet method of binarization.

2.2 Knowledge Distillation

The teacher-student methodology for compression was first proposed by [16] in which the student network was trained on the artificial data which was labeled by the ensemble of models. They used three different algorithms to generate the artificial data for training.

In [1] the student network is trained on the output of the layer before the softmax layer which is also known as logits. They minimized the squared loss between the logits and the output of the student network.

The concept of temperature is introduced in [2] that helps in training the student network with softened outputs and improve its performance as compared to training with just logits [1]. The logits are divided by the temperature and the softmax function is applied to generate the softened probabilities. They have introduced a new objective function that considers even the true labels to train the student network. The loss function is the weighted combination of cross entropy between the true labels and the output of the student network and the cross entropy between the softened probabilities of teacher and student network. The second term in the loss function is given more weight than the first term.

FITNETS were proposed by [17] where student is made to mimic the intermediate representations of the teacher network along with the softened outputs of the teacher network. These intermediate representations from the teacher acts as hints in the training of the student network.

In [18] knowledge is distilled as the dot product of features from any two layers and knowledge is transferred as the flow of information between the layers. They showed that instead of mimicking the intermediate representations of the teachers as [17], their method is better in which the flow of information between the layers is mimicked. [19] distills knowledge to the shallow model using softened outputs and attention maps. This method is similar to our proposed method but the difference is the way they are computing the attention maps is different from our method and we are training the shallow network with multiple teachers but they are using single teacher.

Chapter 3

Proposed Methodology

3.1 Review of Knowledge Distillation

In this work, we extend the existing knowledge distillation method for model compression which was proposed by [2]. In this method a shallow neural network called a student network is trained from the softened outputs of a pre-trained deep neural network called a teacher network. This was first proposed by [1] in which the shallow network was made to mimic the logits(output of the layer just before the softmax layer) of the deep network. They calculate the mean squared loss between the logits of the teacher and the student network.

Knowledge distillation using softened outputs is observed to perform better than using logits for the training of student network. The class probabilities are termed as 'soft-targets', however these soft targets help the student in the training process to converge faster as compared to the hard labels. The student network might not perform well on the training data with the original hard labels as it does with the softened labels as the soft-targets contain more information about the classes than the hard-labels.

The hard labels have zero at all the classes except one place which gives the information about one class which is the target class and no other information is available about the rest of the classes. Soft-targets provide information about all the classes in the form of relative probabilities which helps a lot in fast training of the student network. The training of the student can be done with the high learning rate if the entropy in the soft-targets is high because high entropy provides more information and less variance in the gradients.

If we take an example of dog target class then the cat class might have the relative probability near to dog class but a car will have very less probability for the dog as

truck	cow	dog	cat	car	
0	0	1	0	0	Hard-targets
truck	cow	dog	cat	car	
1e-11	1e-6	0.9	0.1	1e-10	Logits of the model
truck	cow	dog	cat	car	
.006	.05	.3	0.2	.005	Softened output of the model

Figure 3.1: An example of soft-targets for dog class

target class since the car is not at all similar to the dog class in any way. In the given Figure 3.1, the logits of the model shows the relative class probabilities for the dog class. In this example the values of cow, car and truck class is very small and is not close to the probability value of the dog class.

The soft-targets gives information about all the other classes as well which in turn helps the shallow student model to perform the same task as the deep teacher model with less number of parameters and computations. The student model is made to learn not only the finer structure but also the mistakes learned by the teacher model. As shown in the Figure 3.2, to achieve the softened probability distribution of the classes in the model temperature of $\tau > 1$ is used in the softmax layer.

The class probabilities are softened to obtain more information because some probability values are too small i.e. $1e-10$ which is approximately close to zero so not much information can be obtained from these values. The softmax function is modified as hinton[2]:

$$p_i = \frac{\exp(\frac{z_i}{\tau})}{\sum_j \exp(\frac{z_j}{\tau})} \quad (3.1)$$

In the above equation z_i are the logits which are converted to probabilities p_i by the softmax function and τ is the temperature used while training. In Hinton[2] the

temperature values used were : [1,2,5,8,10]. The training process of distilled model can be improved if the true labels are used along with the soft labels as shown in Hinton[2]. They proposed a loss function which is the weighted combination of the cross entropy loss of the student output labels with the true labels and the cross entropy loss of the student softened output with the teacher softened output probability values.

The proposed loss function to be optimized is as follows [2][17]:

$$L_{KD}(W_S) = H(y_{true}, P_S) + \lambda H(P_T^\tau, P_S^\tau) \quad (3.2)$$

where,

P_S and P_T : output probabilities of the student and teacher network.

y_{true} : hard-targets of the student network.

P_T^τ and P_S^τ : softened output. probabilities of the student and teacher network.

λ : a hyper parameter to balance the weights of the two terms in the loss function.

H : cross-entropy loss function.

$$P_T = softmax(\frac{a_T}{\tau}), P_S = softmax(\frac{a_S}{\tau}) \quad (3.3)$$

In Eq 3.3 the terms a_S and a_T are the logit outputs before the softmax layer. Student network is trained at the same temperature as the teacher network and the temperature is set to 1 after the training is done. The value of λ and τ can be tuned to achieve the minimum loss value. The preferred value of λ according to our experiments is either 0.8 or 0.9 because less weight is given to the first term in the loss function.

3.2 Gradient-weighted Class Activation Mapping (Grad-CAM)

Interpretability of a neural network is important to understand the decisions made by the network and why those decisions are made. There are many new methods proposed in the recent times to explain the predictions of the neural network and make its training process more transparent. Since Grad-CAM [3] can be applied to large variety of neural networks without any change in the architecture of the network so we use Grad-CAM for generating visualizations. Interpretability is also important as it provides insights in the wrong predictions of the neural network.

Grad-CAM uses the gradients information of the final convolution layer of the network



Figure 3.2: Grad-CAM visualization(heatmap)

and generates the localization map for a particular target class. It can use activations from any other convolution layer also and generate heatmaps but it considers the last layer as the network has learned features and the classes are most discriminative in the last layer.

Grad-CAM is a class discriminative technique because it confines the target class in the generated visualizations by highlighting the main regions in the heat map. Figure 3.2 shows an example of image and its localization map generated by Grad-CAM for the dog target class. It illuminates the important portion in the image that explains the decision of the network for dog target class. Grad-CAM can even provide relevant explanations for the some non relevant predictions made by the neural network which helps in developing trust in the model predictions.

The localization map $L_c^{Grad-CAM}$ for a target class 'c' is obtained as follows:

- First the gradient is computed with respect to the feature maps (A_k) of the final convolution layer.

$$a_k^c = \frac{1}{Z} \sum \sum \frac{\delta y^c}{\delta A_{ij}^k} \quad (3.4)$$

- To obtain the importance weights of the neuron the gradients computed in the first step are globally average pooled as in Eq.(3.4).

$$L_c^{Grad-CAM} = ReLU\left(\sum_k a_k^c A^k\right) \quad (3.5)$$

- These importance weights are multiplied with the feature maps and ReLU activation function is applied to this weighted combination as shown in Eq.(3.5).

In the Eq.(3.5). ReLU activation function is applied to consider only the positive values since the negative values are not of much interest because that will not affect

the target class as those values might belong to other classes. If those negative values are not removed then the localization maps will not be class-discriminative and will have some unwanted portions in the heat maps from other classes.

3.3 Pipeline of the Proposed Method

In this work, we have implemented the multiple teacher-student framework where a shallow network is trained from the multiple deep networks. The proposed method is an extension of existing knowledge-distillation [2] method of model compression (Explained in Section 3.1). Here, the distilled model is not only trained from the softened outputs but also from the visualizations generated by the teacher network. Visualizations

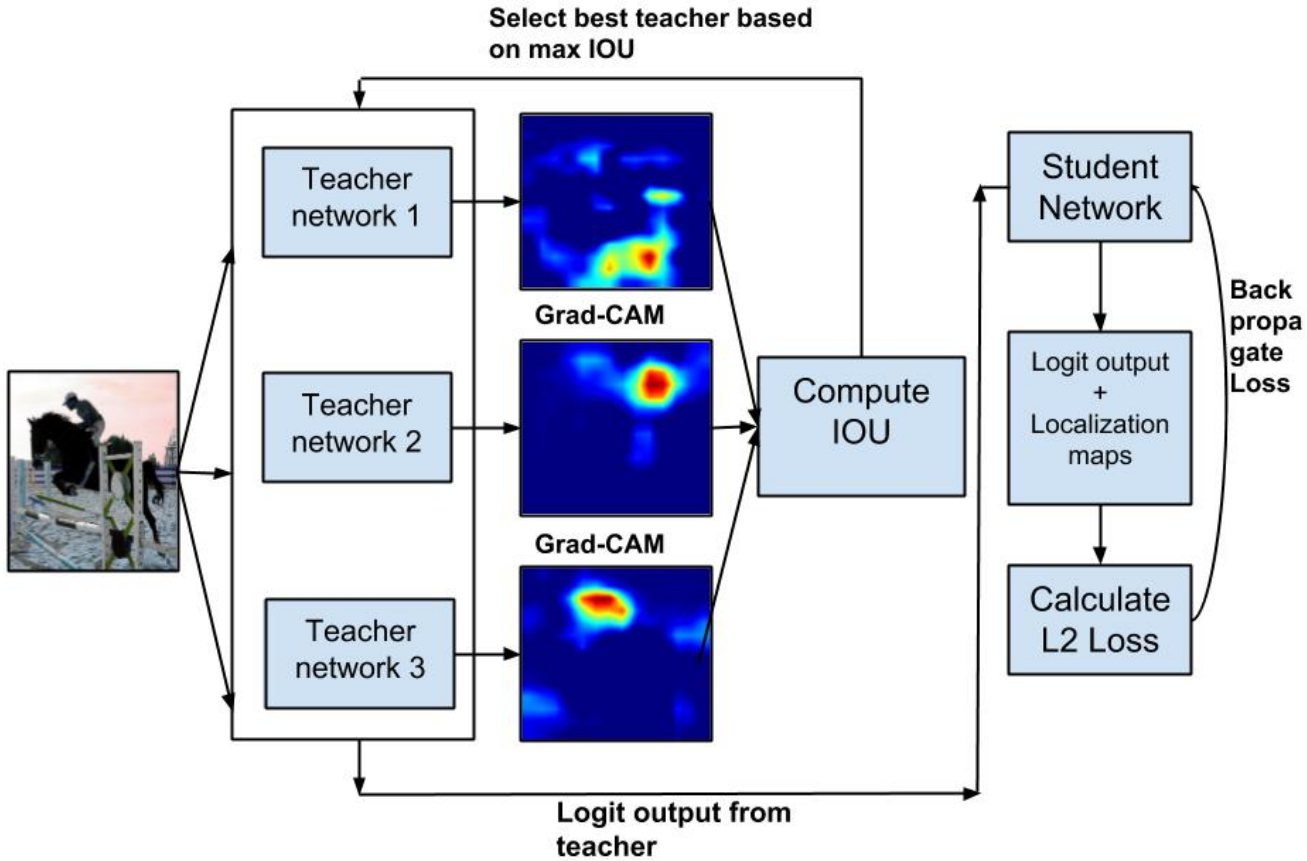


Figure 3.3: Pipeline of the proposed method

plays an important role in training the shallow network because interpretability of a

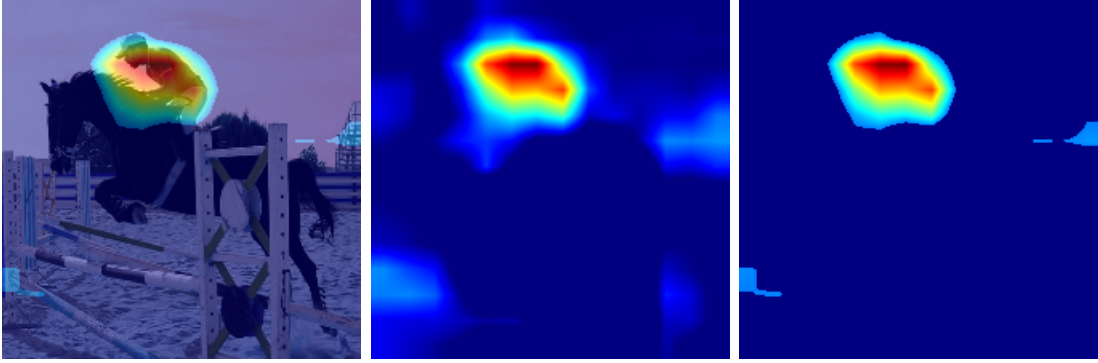


Figure 3.4: a)Image b)Heatmap c)Thresholded heatmap (Person)

neural network is important to understand the decisions made by the network and why those decisions are made. If a network can explain its decisions then it can be used in wide variety of applications.

Student is learning from the relevant portions of the image which are learned by the teacher network so the overall quality of the student training is improved.

Gradient-weighted Class Activation Mapping (Grad-CAM)(Explained in Section 3.2) is used to generate the visualizations of all the teacher networks. Grad-CAM generates localization maps which explains the predictions of the target class by illuminating the significant regions in the localization map. Grad-CAM can be applied to any kind of neural network as it uses the gradient information in the last convolution layer of the network to produce the visualizations.

For a particular training sample the student network is trained with that teacher network which has the best visual explanations. The best teacher network is selected using the max intersection over union(IOU) value between the ground truth bounding box of the image and the illuminated regions in the heat map.

The IOU value is used as the metric to test the visualizations learned by the teacher network. The more the IOU value the better are the visualizations learned by the teacher network. In Figure (3.6) the IOU with the bounding box appears different for the two network even if the accuracy of the networks does not vary much.

To compute the IOU the heat maps are thresholded for intensity value $i > 0.25$ to remove the least illuminated regions. In Figure 3.3, b) represents the image heat map of the image and c)represents the thresholded heat map. Once the IOU values are obtained for all the training samples on all the teacher networks, the values are compared to train the student model with the teacher network that has the maximum IOU.

The student network is made to optimize the following loss function:

$$L_{modified}(W_S, W_T) = L_{KD}(W_S) + \beta L_{gc}(W_S, W_T) \quad (3.6)$$

The above loss is the weighted combination of standard knowledge distillation loss defined in Eq.(3.2) and the visualizations loss L_{gc} defined in Eq.(3.5).

$$L_{gc}(W_S) = ||L_c^S(W_S) - L_c^T(W_T)||_2^2 \quad (3.7)$$

β is the tunable parameter to balance the weight given to both the loss functions.

L_c is the term defined in Eq.(3.5).

The student network mimics the heat maps of the teacher network along with the softened outputs. Eq.(3.7) shows the mean squared loss between the teacher heat maps and the student heat maps.

3.3.1 Drawback of the proposed method

We have performed experiments for the classification task but the disadvantage of the above method is that it requires the datasets with bounding boxes to compute the IOU of the heat map with the ground truth.

Solution to the above problem:

We use a different method in case of datasets without bounding boxes. We introduced a new step in the proposed pipeline to resolve this problem. We have used the Grad-CAM visualizations to choose the best teacher network instead of computing IOU.



Figure 3.5: a)Image b)Network1 c)Network2

We have modified the image using its generated heat map and occluded the part of the image which was least illuminated in the corresponding heat map of that image as shown in Figure: 3.5. Then the new occluded image is passed again through the trained model to check the predicted class. If the model still predicts the same target class as predicted earlier for normal image for the occluded image too with not much drop in the probability then the visualization produced by the teacher network is accurate.

Different teacher networks were evaluated with the above method to choose the best teacher network. The teacher network which has the minimum drop in the probability score for the occluded image is chosen for that training example to train the student network. Student network is trained in the similar way as mentioned above and minimizes the same loss function mentioned in Eq.(3.4). The modification is done only in choosing the best teacher out of multiple teachers to train the student network.

3.3.2 Multiple Teacher network visualizations

Comparison between the visualizations of different teacher networks. The visualizations shows the difference in the learned representations between the two teacher networks. Even with the slight difference in the accuracy of the models the visualizations varies for the same training image as shown in the Figure 3.6.

It can be seen from image of the bird in the last row that there is a visible difference between the IOU of the two networks and the more the IOU the better is the visualization of the network.

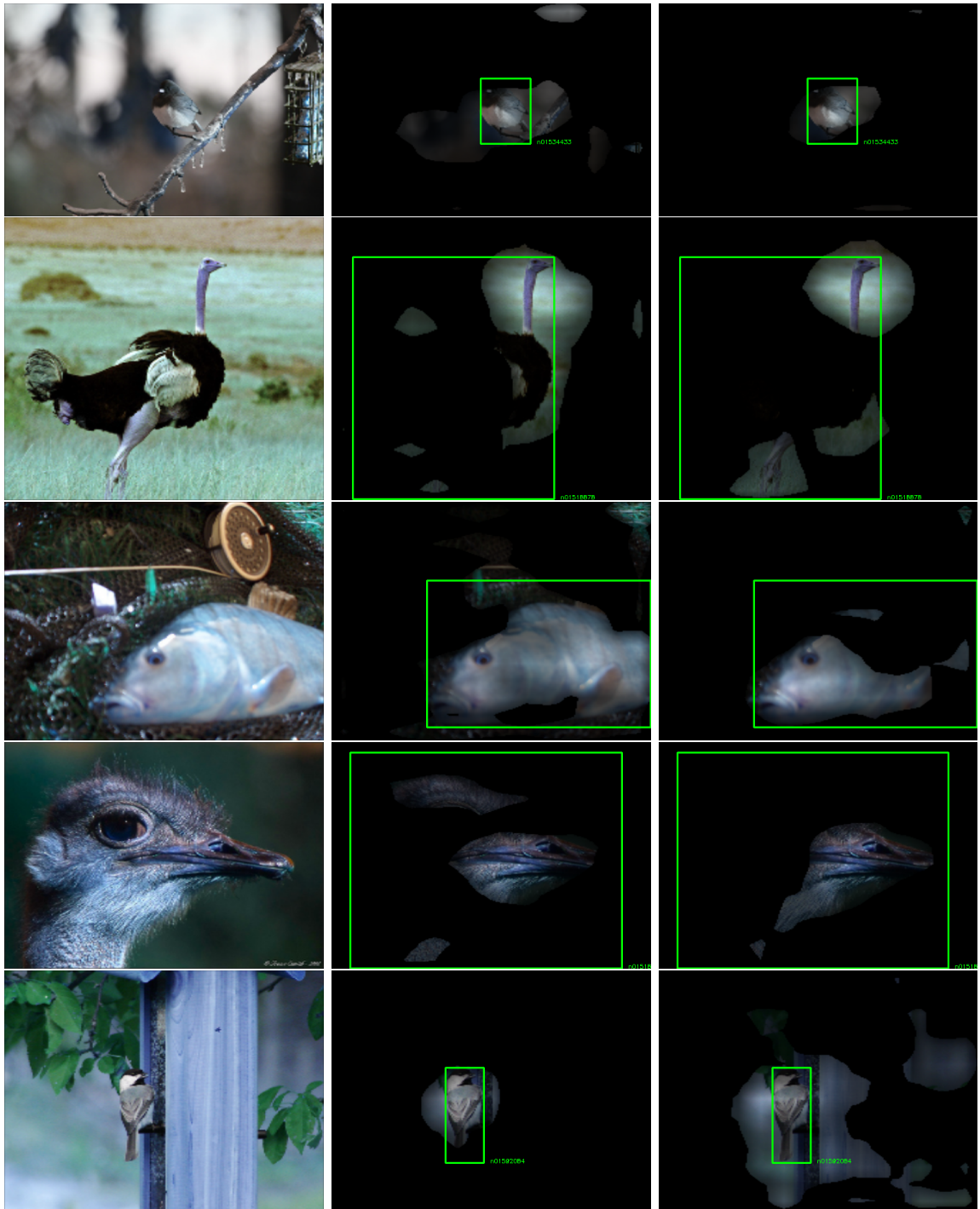


Figure 3.6: a)Image b)Network1 c)Network1. Images b) and c) shows the visualizations of the two different networks.

Chapter 4

Experimental Results

We have evaluated our proposed method on three datasets: CIFAR-10 [20], CIFAR-100[20] and Imagenet Large Scale Visual Recognition Challenge 2012(ILSVRC2012)[21] for the task of image classification.

4.1 CIFAR-10

The dataset consists 10 classes of natural images. The images are RGB images of dimension 32x32. The training data consists of 50,000 images and testing data has 10,000 images. The images are preprocessed and mean is subtracted from images. Data-augmentation is done to improve the performance of the teacher network. The images are flipped randomly along the horizontal axes. The images are also shifted randomly along both the horizontal and vertical axes.

Teacher Networks:

We have used three teacher networks to evaluate our proposed method.

Network-in-Network(NIN) [10], Wide Residual Network(WRN) [22] and VGGNet [23].

WRN-28-8 is used where the network depth is 28 and widening factor is $k = 8$. We have modified the VGG16 architecture and included batch normalization layers in the network to improve its performance on CIFAR-10. For each of the teacher network different kind of data augmentation is done to achieve state-of-the-art performance on the dataset. In Table 4.1 the number of parameters and accuracy is stated for all the three teacher networks.

Student Network:

Table 4.1: Results on CIFAR-10

Models	No. of params(Million)	Accuracy(%)
Teacher network1	13M	89.93
Teacher network2	23M	91.90
Teacher network3	14M	93.35
Student network original	9M	80.01
Student network(proposed method)	9M	80.33

Table 4.2: Accuracy of the student network when trained from NIN

Models	Accuracy(%)
Teacher network1	89.93
Student network original	80.01
Student network trained from NIN	80.07

ResNet-18 [12] is used as the student network. The student network is trained on the softened outputs and localization maps from all the three teachers as explained in Section 3.3. The accuracy of the student network when trained on the original labels is 80.01% and when student network is trained from multiple teacher networks and optimize the loss function mentioned in Eq.(3.6) the accuracy increases by +0.32%. While training temperature τ is set to 2 and λ is set to 0.4(Eq 3.2) and β is set to 0.5(Eq 3.6).

4.1.1 Analysis Of Knowledge Distillation from Multiple teachers

The accuracy of the student network in Tables 4.2, 4.3 and 4.4 shows the promise of the proposed method because when student network is trained from a single teacher network the accuracy achieved is less then the accuracy of the network trained from multiple teacher networks as shown in Table 4.1.

When student is trained from teacher1 i.e. NIN the accuracy achieved is 80.07(%) as shown in Table 4.2 which is better then the accuracy of the student model when its trained on the hard-targets but its less then 80.33(%) which is the accuracy achieved by the student with our proposed method.

The accuracy of the student when trained from other two teacher networks separately is 80.11(%) and 80.14(%) which is also less then the student accuracy achieved with

Table 4.3: Accuracy of the student network when trained from WRN

Models	Accuracy(%)
Teacher network2	91.90
Student network original	80.01
Student network trained from WRN	80.11

Table 4.4: Accuracy of the student network when trained from VGGNet

Models	Accuracy(%)
Teacher network3	93.35
Student network original	80.01
Student network trained from VGG	80.14

our proposed method.

4.2 CIFAR-100

The dataset consists 100 classes of natural images. The images are RGB images of dimension 32x32. The training data consists of 50,000 images and testing data has 10,000 images. The images are preprocessed and mean is subtracted from images. Data-augmentation is done to improve the performance of the teacher network. The images are rotated randomly along the horizontal axes. The images are also shifted randomly along both the horizontal and vertical axes. We have used three teacher networks to evaluate our proposed method.

Table 4.5: Results on CIFAR-100

Models	No. of params(Million)	Accuracy(%)
Teacher network1	55M	71.12
Teacher network2	36M	73.56
Teacher network3	1M	70.49
Student network original	9M	60.20
Student network(proposed method)	9M	60.82

Teacher networks:

We have used three teacher networks to evaluate our proposed method. Wide Resid-

ual Network(WRN-40-10) [22], WRN-28-10[22] and Densely Connected Convolutional Networks(DenseNets) [24].

Student network:

ResNet-18 is used as the student network. The student network is trained on the softened outputs and localization maps from all the three teachers as explained in Section 3.3. The accuracy of the student network when trained on the original labels is 60.20% and when student network is trained from multiple teacher networks and optimize the loss function mentioned in Eq.(3.6) the accuracy increases by +0.62%. While training temperature τ is set to 2 and λ is set to 0.4(Eq 3.2) and β is set to 0.5(Eq 3.6).

Compression in terms of storage and training complexity

As shown in Table 4.5 and 4.1 the number of parameters of all the three teacher networks are approximately 3-5x greater as compared to the student network parameters which results in compression in terms of storage. The student network is a shallow network as compared to the deep network so the training complexity is less as compared to the teacher networks. The training of the student is done for less number of iterations as compared to original student network trained on hard-targets.

4.3 Imagenet

Experiments are performed on 100 classes of the dataset out of the 1000 classes. The dataset contains natural images with a total of 52,272 training samples and 1000 testing samples, each of which is a 224x224 RGB image. No preprocessing is done on the data.

Table 4.6: Results on Imagenet

Models	No. of params(Million)	Accuracy(%)
Teacher network1	136M	73.16
Teacher network2	140M	74.21
Teacher network3	42M	77.59
Student network original	9M	65.06
Student network(proposed method)	9M	65.73

Teacher Networks:

We have used three teacher networks to evaluate our proposed method. The teacher networks used are: VGG-16, VGG-19 and Resnet-101.

Student network:

ResNet-18 is used as the student network. The student network is trained on the softened outputs and localization maps from all the three teachers as explained in Section 3.3. The accuracy of the student network when trained on the original labels is 65.06% and when student network is trained from multiple teacher networks and optimize the loss function mentioned in Eq.(3.6) the accuracy increases by +0.67%. While training temperature τ is set to 2 and λ is set to 0.4(Eq 3.2) and β is set to 0.5(Eq 3.6).

4.3.1 Analysis Of Knowledge Distillation from Multiple teachers

We performed experiments on Tiny ImageNet dataset [21] to analyze the difference between the student performance in case of both multiple teachers and single teacher. The dataset consists of 200 classes. The images are RGB images of dimension 64x64. Each class consists of 500 training images and 50 validation images. The training data consists of total 100,000 images and validation data of 10,000 images.

Table 4.7: Results on Tiny ImageNet

Models	Accuracy(%)
Teacher network1(VGG16)	72.57
Teacher network2(VGG19)	75.63
Student network original	66.09
Student network trained from VGG16	67.92
Student network trained from VGG19	68.01
Student network trained from both VGG16+VGG19	68.28

Two teacher networks are trained on this dataset i.e. VGG16 and VGG19 and ResNet-18 is trained as the student network. In Table 4.7 the accuracy of the student model trained without knowledge distillation is 66.09%. When the student model is trained with single teacher network the accuracy achieved is 67.92% and 68.01% for the two

teacher networks which is less than the accuracy achieved from training the student network with multiple teacher networks.

4.4 Analysis on CIFAR-10

4.4.1 Effect of λ and β

Experiments are performed to find the best values of λ and β in Eq (3.2) and (3.6). Values of λ and β are varied at temperature $\tau = 2$. The best accuracy value is obtained for $\lambda = 0.4$ and $\beta = 0.5$ by giving equal weights to the cross-entropy loss of the softened outputs of the teacher and student network and the means square loss of the localization maps of the teacher and student network.

Table 4.8: Effect of λ and β on Accuracy

λ	β	Accuracy(%)
0.7	0.2	78.92
0.4	0.5	80.33
0.5	0.4	79.47
0.4	0.4	80.27
0.6	0.2	80.15
0.5	0.2	80.01
0.3	0.3	79.87

4.4.2 Effect of Temperature

Another analysis was done by varying the temperature τ with the two values of : $\lambda = 0.4$ and $\beta = 0.4$ and $\lambda = 0.4$ and $\beta = 0.5$

Table 4.9: Effect of temperature on Accuracy when $\lambda = 0.4$ and $\beta = 0.4$

Temperature(τ)	1	2	3	4	5	6	7	8	9	10
Accuracy(%)	79.78	80.27	80.09	79.67	79.44	79.75	79.41	79.92	79.96	79.88

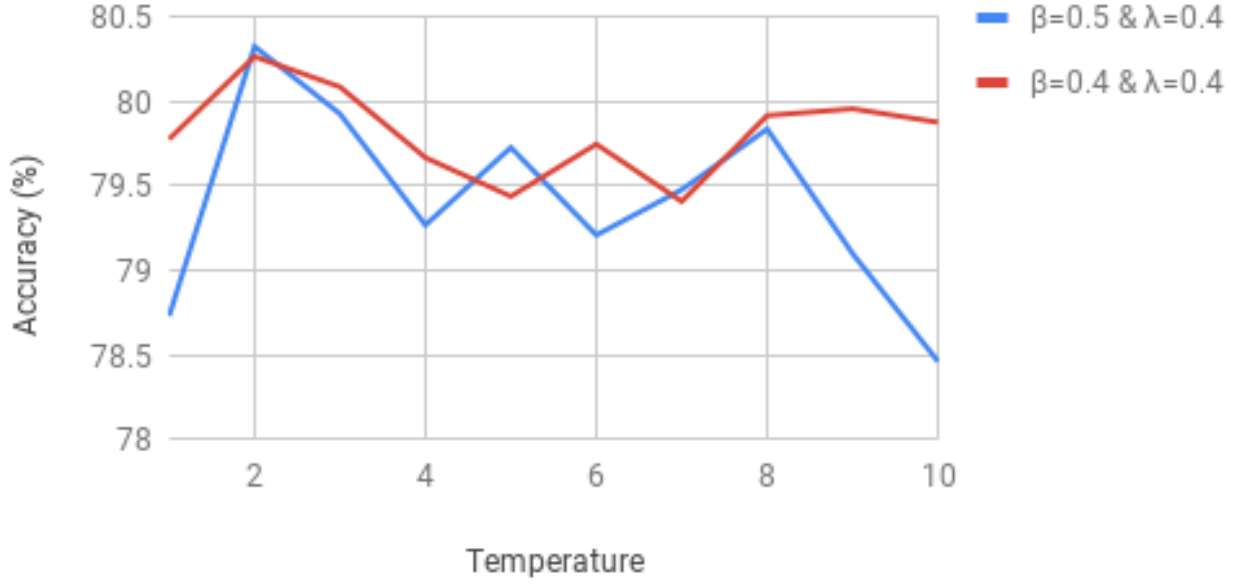


Figure 4.1: Effect of temperature

Table 4.10: Effect of temperature on Accuracy when $\lambda = 0.4$ and $\beta = 0.5$

Temperature(τ)	1	2	3	4	5	6	7	8	9	10
Accuracy(%)	78.73	80.33	79.93	79.27	79.73	79.21	79.48	79.84	79.1	78.46

As can be seen in Figure 4.1 the plot between temperature and accuracy is plotted to study the effect of temperature in distilling knowledge. The network performs the best at temperature $\tau = 2$. To analyze the effect in detail we plotted for the two set of parameters. Best results have been obtained on values $\lambda = 0.4$ and $\beta = 0.4$ for the student network but we have plotted for values $\lambda = 0.4$ and $\beta = 0.5$ to get a clear idea of the best chosen temperature value. For both the set of values the maximum accuracy is achieved for temperature $\tau = 2$.

Chapter 5

Conclusions

In this work, we have discussed a multiple teacher-student framework for model compression which is an extension of existing knowledge-distillation method. We explored this compression method further since this method focuses on reducing the storage and training time complexity. Most of the compression methods proposed earlier focus only in reducing the storage complexity of the models as discussed in Chapter 2. Learning from multiple teachers and their visualizations helped the shallow student model to perform better than the basic knowledge-distillation method. We used Grad-CAM to generate the visualizations because it can be applied wide number of neural networks without any change in the architecture of the networks. Also it is class discriminative and localizes the target class by highlighting the important regions of the target class.

With the help of Grad-CAM the best teacher network is selected for a particular training sample helping the training of student network. We presented two different approaches to select the best teacher network for dataset with bounding boxes and without bounding boxes. We evaluated the proposed method on three datasets: CIFAR-10, CIFAR-100 and Imagenet Large Scale Visual Recognition Challenge 2012(ILSVRC2012).

We have performed an analysis on CIFAR-10 and Tiny ImageNet dataset to show that the student performs better when its trained from multiple teachers compared to training from single teacher and the results are shown for student network trained on each teacher network separately. The experimental results shows the improvement in the performance of student network trained with our proposed method. We performed analysis on CIFAR-10 to see the effect of temperature on accuracy and the effect of varying weights of tunable parameters $\lambda =$ and $\beta =$ on accuracy.

References

- [1] J. Ba and R. Caruana. Do Deep Nets Really Need to be Deep? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., *Advances in Neural Information Processing Systems 27*, 2654–2662. Curran Associates, Inc., 2014.
- [2] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .
- [3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *ArXiv e-prints* .
- [4] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016 2921–2929.
- [5] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing Neural Networks with the Hashing Trick. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML’15*. JMLR.org, 2015 2285–2294.
- [6] Y. Gong, L. Liu, M. Yang, and L. Bourdev. Compressing Deep Convolutional Networks using Vector Quantization. *ArXiv e-prints* .
- [7] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning Both Weights and Connections for Efficient Neural Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15*. MIT Press, Cambridge, MA, USA, 2015 1135–1143.
- [8] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky. Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition. *ArXiv e-prints* .

- [9] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation. *ArXiv e-prints* .
- [10] M. Lin, Q. Chen, and S. Yan. Network In Network. *ArXiv e-prints* .
- [11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016 2818–2826.
- [12] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints* .
- [13] M. Courbariaux, Y. Bengio, and J.-P. David. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. *ArXiv e-prints* .
- [14] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1. *ArXiv e-prints* .
- [15] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. *ArXiv e-prints* .
- [16] C. Bucilu, R. Caruana, and A. Niculescu-Mizil. Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006 535–541.
- [17] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio. FitNets: Hints for Thin Deep Nets. *ArXiv e-prints* .
- [18] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017 .
- [19] S. Zagoruyko and N. Komodakis. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. *ArXiv e-prints* .
- [20] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images .
- [21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09. 2009 .

- [22] S. Zagoruyko and N. Komodakis. Wide Residual Networks. *ArXiv e-prints* .
- [23] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv e-prints* .
- [24] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. *ArXiv e-prints* .