KNOWLEDGE DISTILLATION IN NEURAL NETWORK

By -

Sarthak Rawat (2021202006)

Shreyash Agrawal (2021201074)

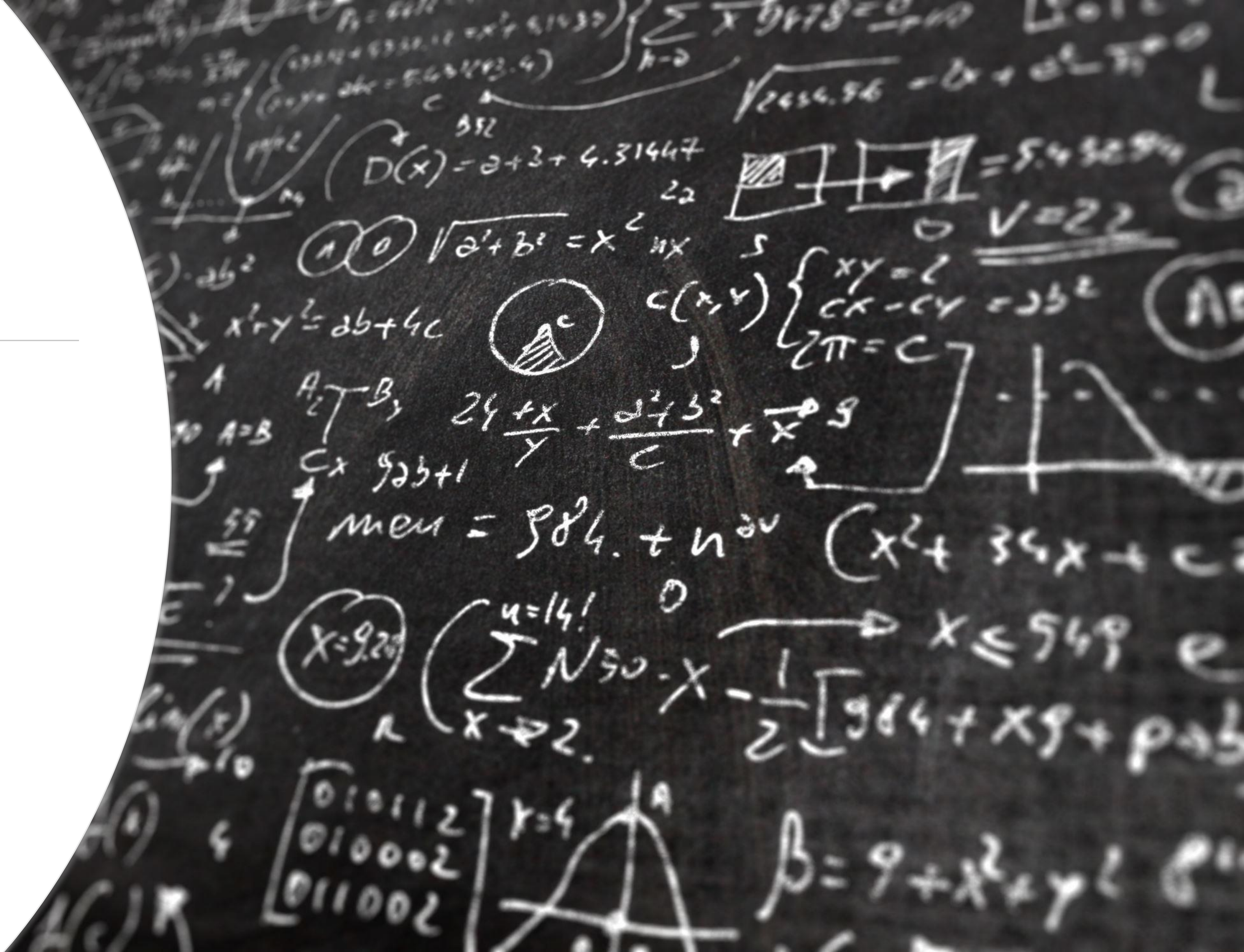
Kamal Phoolwani (2021201054)

OUTLINE

- Knowledge Distillation
- Model Compression
- Hard Target and Soft Targets
- Experiments
  - MNIST Dataset
  - CIFAR10
- Conclusion
- References

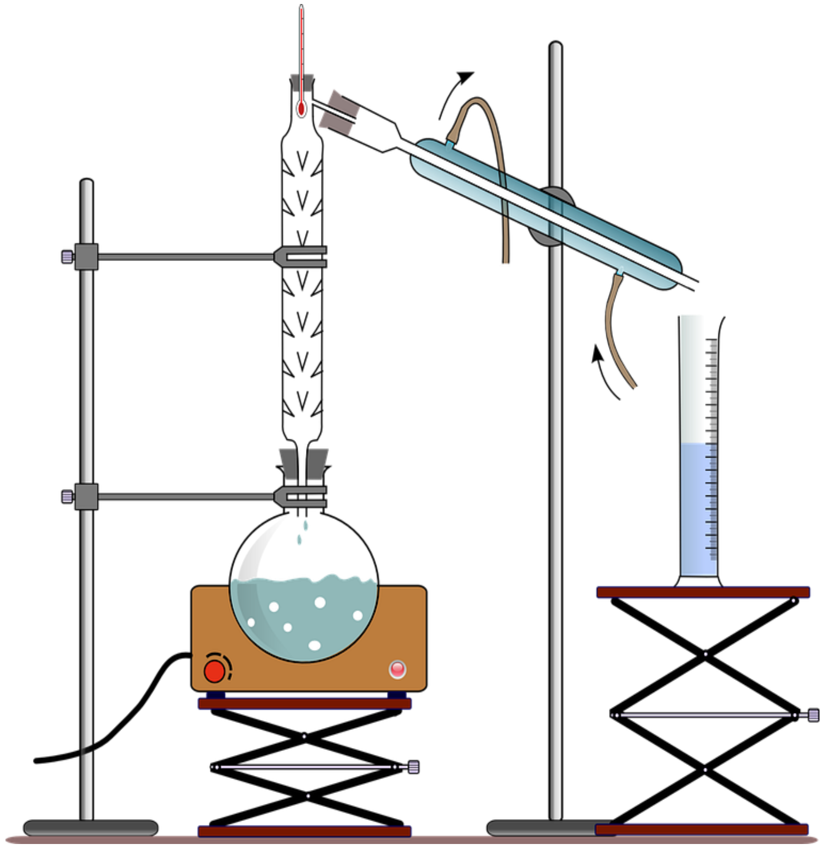
Keywords

- Knowledge
- Distillation
- Temperature
- Logits
- SoftMax
- KL Diversions



Knowledge Distillation

- Knowledge distillation is a process of distilling or transferring the knowledge from a (set of) large, cumbersome model(s) to a lighter, easier-to-deploy single model, without significant loss in performance.



Model Compression

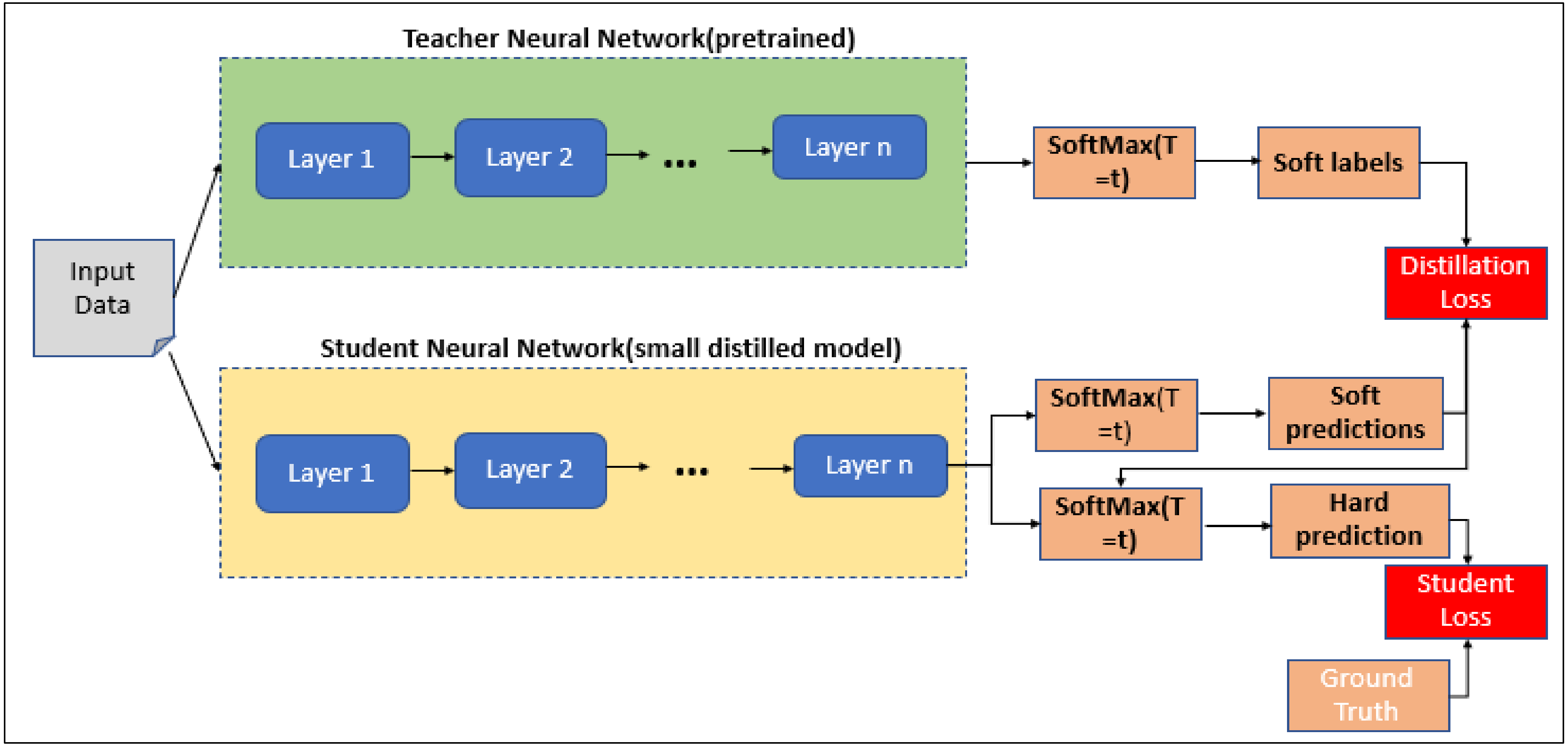
Ensemble Learning

- Cumbersome and may be too computationally expensive

Solution

- The knowledge acquired by a large ensemble of models can be transferred to a single small model.
- We call “distillation” to transfer the knowledge from the cumbersome model to a small model that is more suitable for deployment.

Knowledge Distillation



Distillation

Distillation loss uses the soft targets to minimize the squared difference between the logits produced by the cumbersome model and the logits produced by the small model.

$$q_i = \frac{exp(\overset{\text{Logits}}{z_i}/T)}{\sum_j exp(z_j/\underset{\text{Temperature}}{T})}$$

Knowledge is transferred to the distilled model by training the cumbersome model with a high temperature in its SoftMax to generate soft target distribution. The same high temperature is used for training the distilled model, but after it has been trained, it uses a temperature of 1.

Hard Targets and Soft Targets

- **Hard targets** are generated when using a SoftMax function. Using the SoftMax function, the model almost always produces the correct answer with very high confidence and has very little influence during transfer of Knowledge from Teacher to Student.
- **Soft targets** use the logits, the inputs to the final SoftMax rather than the SoftMax's probabilities as the targets for learning the small model. When the soft targets have high entropy, they provide much more information per training case than hard targets.

KL Divergence

- The Kullback-Leibler divergence (hereafter written as KL divergence) is a measure of how a probability distribution differs from another probability distribution. Classically, in Bayesian theory, there is some true distribution P(X)P(X); we'd like to estimate with an approximate distribution Q(X)Q(X). In this context, the KL divergence measures the distance from the approximate distribution QQ to the true distribution PP.
- Mathematically, consider two probability distributions P,QP,Q on some space XX. The Kullback-Leibler divergence from QQ to PP (written as DKL(P∥Q))

$$D_{KL}(P\|Q) = \mathbb{E}_{x\sim P}\left[\log \frac{P(X)}{Q(X)}\right]$$

MNIST Dataset

```
Model: "teacher"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 14, 14, 256)       2560

 leaky_re_lu (LeakyReLU)     (None, 14, 14, 256)       0

 max_pooling2d (MaxPooling2D  (None, 14, 14, 256)      0
 )

 conv2d_1 (Conv2D)           (None, 7, 7, 512)         1180160

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 10)                250890

=================================================================
Total params: 1,433,610
Trainable params: 1,433,610
Non-trainable params: 0
_________________________________________________________________
```

```
Model: "student"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_5 (Flatten)         (None, 784)               0

 dense_9 (Dense)             (None, 100)               78500

 dense_10 (Dense)            (None, 50)                5050

 dense_11 (Dense)            (None, 10)                510

=================================================================
Total params: 84,060
Trainable params: 84,060
Non-trainable params: 0
_________________________________________________________________
```

MNIST Dataset

Teacher Accuracy: 97.86%

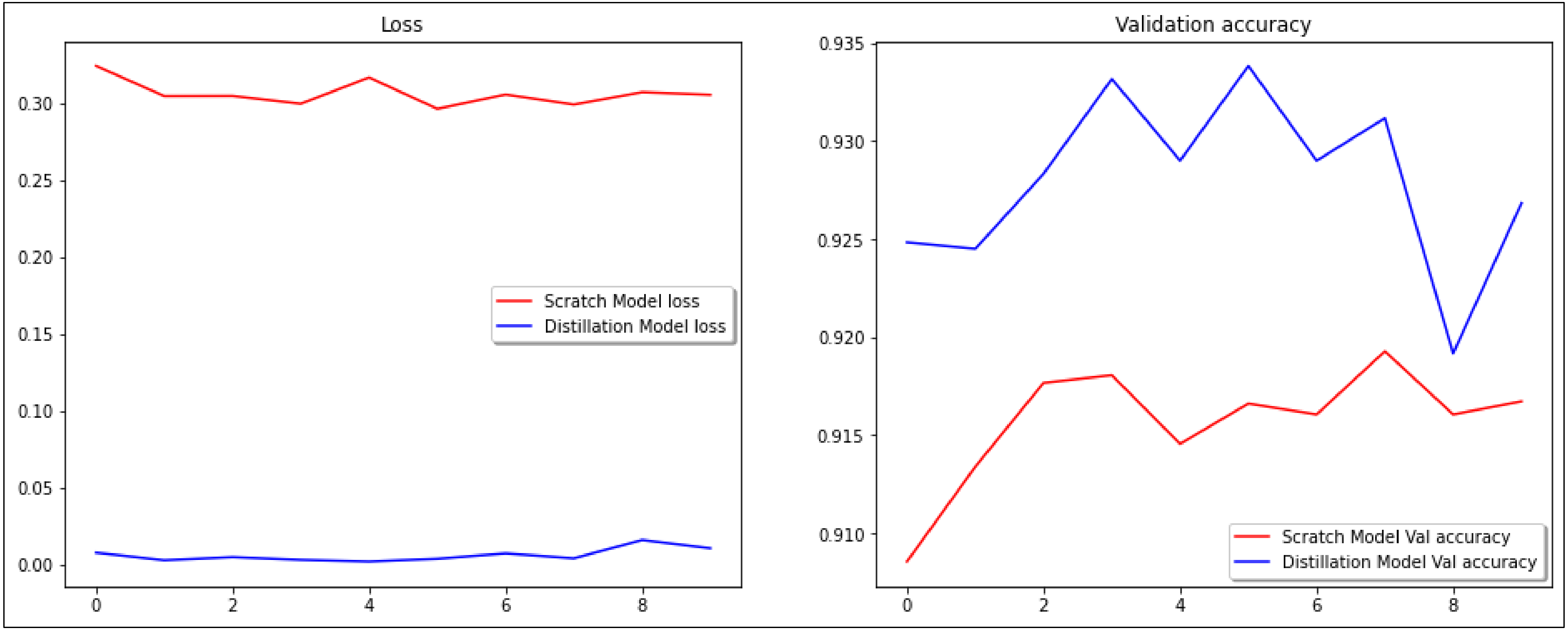
Student Model (without Distillation)

- **Accuracy** - 91.28%
- **Epochs** – 10
- **Ground Truth Loss** - Sparse Categorical Cross-entropy

Student Model(with Distillation)

- **Accuracy** – 91.86%
- **Epochs** - 10
- **Alpha** – 0.1
- **Temperature** – 3
- **Distillation Loss** - KL Divergence
- **Ground Truth Loss** - Sparse Categorical Cross-entropy

Distilled VS Scratch Model



MNIST Dataset (training without 3)

```
Model: "teacher"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 784)               0

 dense (Dense)               (None, 1200)              942000

 dropout (Dropout)           (None, 1200)              0

 dense_1 (Dense)             (None, 1200)              1441200

 dropout_1 (Dropout)         (None, 1200)              0

 dense_2 (Dense)             (None, 10)                12010

=================================================================
Total params: 2,395,210
Trainable params: 2,395,210
Non-trainable params: 0
```

```
Model: "Student"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 784)               0

 dense (Dense)               (None, 800)               628000

 dropout (Dropout)           (None, 800)               0

 dense_1 (Dense)             (None, 800)               640800

 dropout_1 (Dropout)         (None, 800)               0

 dense_2 (Dense)             (None, 10)                8010

=================================================================
Total params: 1,276,810
Trainable params: 1,276,810
Non-trainable params: 0
```

MNIST Dataset (training without 3)

- We trained the student model with the data without label **3**. The teacher model was trained on the complete dataset.
- Teacher Model had **2** hidden layers both with **1200** units each.
- Student Model had **2** hidden layers with **800** units each and temperature was set to **7**.
- Despite this, the student learned the parameters required for classifying **3** and was able to generalize well over the unseen translations.
- The model gave accuracy of **74.75** on test data with only **3**.

CIFAR10 Dataset

```
Model: "teacher"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_172 (Conv2D)         (None, 16, 16, 256)       7168

 leaky_re_lu_172 (LeakyReLU)  (None, 16, 16, 256)      0

 max_pooling2d_172 (MaxPooli  (None, 16, 16, 256)      0
 ng2D)

 conv2d_173 (Conv2D)         (None, 8, 8, 512)         1180160

 leaky_re_lu_173 (LeakyReLU)  (None, 8, 8, 512)        0

 max_pooling2d_173 (MaxPooli  (None, 8, 8, 512)        0
 ng2D)

 flatten_86 (Flatten)        (None, 32768)             0

 dense_86 (Dense)            (None, 10)                327690

=================================================================
Total params: 1,515,018
Trainable params: 1,515,018
Non-trainable params: 0
_________________________________________________________________
```

```
Model: "student"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_174 (Conv2D)         (None, 16, 16, 64)        1792

 leaky_re_lu_174 (LeakyReLU)  (None, 16, 16, 64)       0

 max_pooling2d_174 (MaxPooli  (None, 16, 16, 64)       0
 ng2D)

 conv2d_175 (Conv2D)         (None, 8, 8, 256)         147712

 leaky_re_lu_175 (LeakyReLU)  (None, 8, 8, 256)        0

 max_pooling2d_175 (MaxPooli  (None, 8, 8, 256)        0
 ng2D)

 flatten_87 (Flatten)        (None, 16384)             0

 dense_87 (Dense)            (None, 10)                163850

=================================================================
Total params: 313,354
Trainable params: 313,354
Non-trainable params: 0
_________________________________________________________________
```

CIFAR10 Dataset

Teacher Accuracy: 76.12%

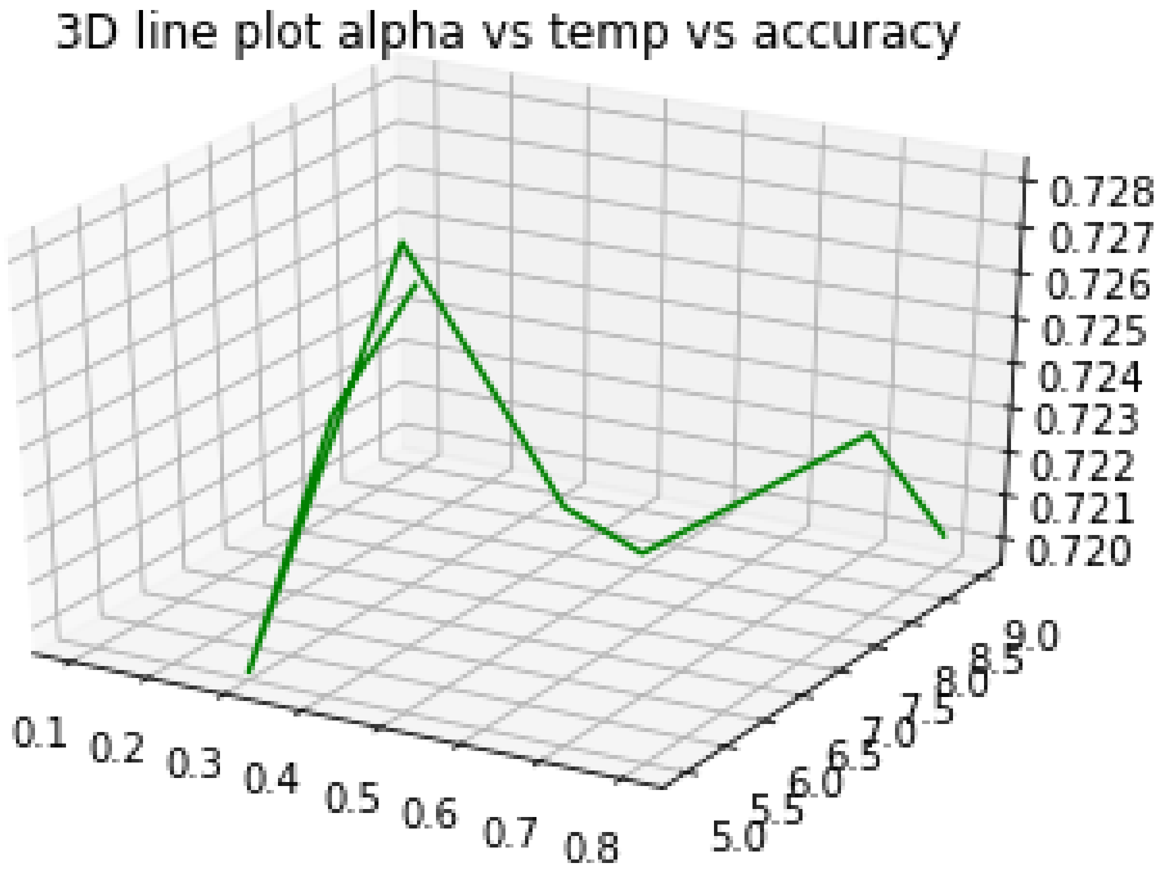
Student Model (with distillation)

- **Accuracy** - 71.07%
- **Epochs** - 5
- **Alpha** – 0.3
- **Temperature** – 2
- **Distillation Loss** - KL Divergence
- **Ground Truth Loss** - Sparse Categorical Cross-entropy

Student Model (without distillation)

- **Accuracy** - 70.19%
- **Epochs** - 5
- **Loss** - Sparse Categorical Cross-entropy

3D plot



3D line plot alpha vs temp vs accuracy
0.728
0.727
0.726
0.725
0.724
0.723
0.722
0.721
0.720
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0

CIFAR10 with RESNet

Teacher

- **Accuracy** – 82.1 %
- **Epochs** – 9

Student

- **Accuracy** – 83.1 %
- **Epochs** – 11
- **Alpha** – 0.5
- **Temperature** – 1
- **Distillation Loss** - KL Divergence
- **Ground Truth Loss** - Sparse Categorical Cross-entropy

Conclusion

- We have shown that distilling works very well for transferring knowledge from an ensemble or from a large highly regularized model into a smaller, distilled model.
- On MNIST, distillation works remarkably well even when the transfer set that is used to train the distilled model lacks any examples of one or more of the classes.
- On CIFAR10, according to our observations an increase in accuracy was found in distillation model.

References

- https://keras.io/examples/vision/knowledge_distillation/
- https://medium.com/analytics-vidhya/knowledge-distillation-in-a-deep-neural-network-c9dd59aff89b
- https://arxiv.org/abs/1910.02551