Hackers

Mathematical Machine Learning Seminar

Finley Yu, Laura Lecinena Pastor

21.11.2023

Background

Research questions

In this paper, authors proposed shallow nets could achieve similar accuracies as deep networks on speech and object recognition.
The tricks of model compression are as follows:

1. Train the model directly on the **logit** values before 'softmax' rather than the probabilities to acquire higher accuracy
2. Introduce a linear layer to speed up the training
3. Mimic loss function: L2 loss function should be preferred to KL divergence

We conducted some experiments to examine the performance of the mimic student model and the validity of these tricks.

Fashion MNIST dataset

We chose the Fashion MNIST over CIFAR-10 to examine the paper's results.

Aspect	**Fashion MNIST**	**CIFAR-10**
Source	Online fashion retailer	Canadian Institute for Advanced Research
Size	70,000 images	60,000 images
Resolution	28 by 28 pixels	32 by 32 pixels
Color	Grayscale	RGB
Classes	10	10
Class names	Fashion products	Animals, Transports
Class distribution	Balanced	Balanced

Tabelle: Fashion MNIST and CIFAR-10

As shown in the Table 1, the Fashion MNIST has a larger size and lower input dimension, which means more data to train and less computational cost.

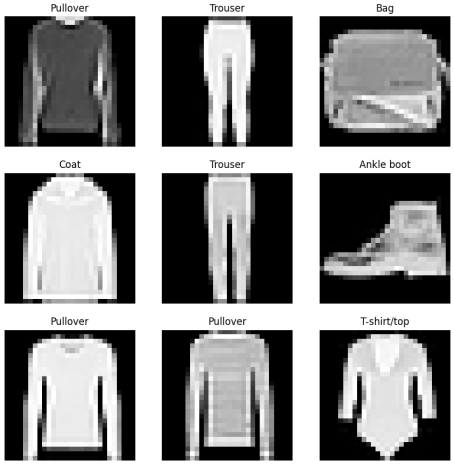


Abbildung: Preview of the Fashion MNIST dataset

Models

Models

- CNN (Teacher model):
  - 1 convolutional layer (3*3, 16 channels)
  - 1 max pooling layer
  - 3 hidden layers containing 64 ReLU units
  - Cross Entropy loss
- DNN:
  - 3 fully connected feedforward hidden layers consisting of 2000 ReLU units
  - Cross Entropy loss
- SNN:
  - 1 hidden layer consisting of 8000 ReLU units (dropout $= 0.5$)
  - Cross Entropy loss
  - trained on original data

SNN Mimic model

We tried different numbers of ReLU units and Loss functions to examine the proposed tricks in the SNN mimic model:

- bottleneck layer consisting of 20 linear units (« input dimension 784 and output dimension 8,000)
- 1 hidden layer consisting of 8,000 (400, 20,000) ReLU units
- Loss function: L2 loss (KL divergence)
- CNN as teacher model

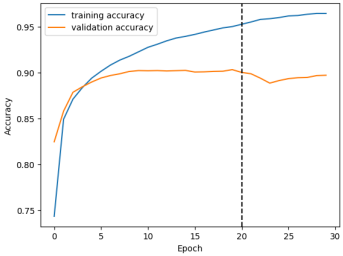
Training details

- stochastic gradient descent with momentum (0.9)
- batch size: 128 images
- training epochs: 30
- learning rate: 0.01

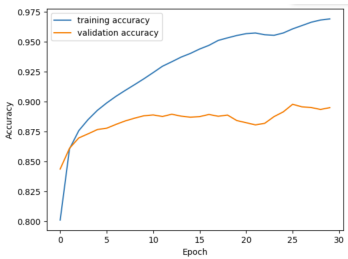
The code is available in our Github repo:
https://github.com/Finley-Maple/Seminar-MML-SNN.git

Results

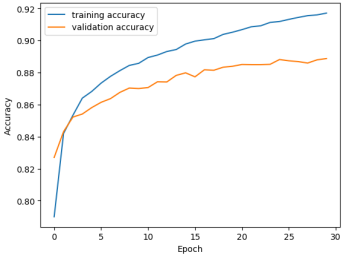
Models training process



(a) CNN



(b) DNN



(c) SNN

Abbildung: Models training process: CNN, DNN, SNN

Shallow nets results

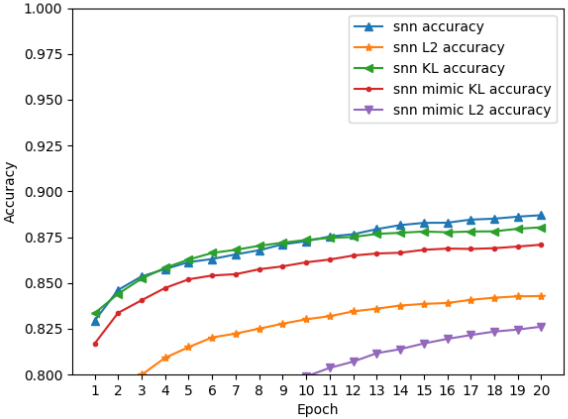


Abbildung: Accuracy of Shallow nets vs. training epoch

Different number of hidden units in student model

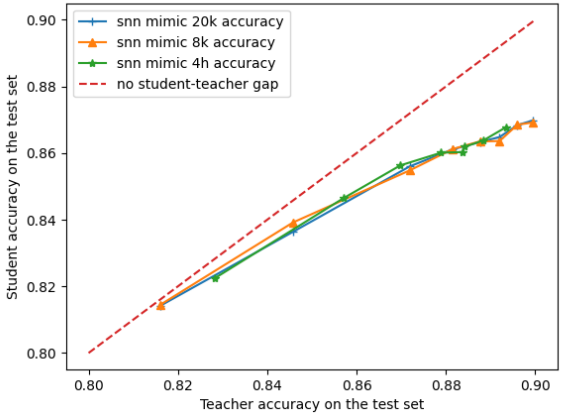


Abbildung: Accuracy of student models continue to improve as accuracy of teacher models improve

Accuracy and training cost

model	#parameters	accuracy	training cost
CNN	$\sim$ 210k	90.24%	16m30s
DNN	$\sim$ 10M	89.48%	42m33s
SNN	$\sim$ 6M	88.87%	20m7s
SNN-mimic-8k	$\sim$ 250k	82.62%	11m29s
SNN-mimic-KL-8k	$\sim$ 250k	87.09%	12m10s

Tabelle: Performance of the models

Accuracy and training cost

model	#parameters	accuracy	training cost
CNN	$\sim$ 210k	90.24%	16m30s
DNN	$\sim$ 10M	89.48%	42m33s
SNN	$\sim$ 6M	88.87%	20m7s
SNN-mimic-8k	$\sim$ 250k	82.62%	11m29s
SNN-mimic-KL-8k	$\sim$ 250k	87.09%	12m10s

Tabelle: Performance of the models

$\rightarrow$ CNN better than DNN (as in paper)

Accuracy and training cost

model	#parameters	accuracy	training cost
CNN	$\sim$ 210k	90.24%	16m30s
DNN	$\sim$ 10M	89.48%	42m33s
SNN	$\sim$ 6M	88.87%	20m7s
SNN-mimic-8k	$\sim$ 250k	82.62%	11m29s
SNN-mimic-KL-8k	$\sim$ 250k	87.09%	12m10s

Tabelle: Performance of the models

$\rightarrow$ CNN better than DNN (as in paper)
$\rightarrow$ For the mimic model, KL loss is better

Accuracy and training cost

model	#parameters	accuracy	training cost
CNN	$\sim$ 210k	90.24%	16m30s
DNN	$\sim$ 10M	89.48%	42m33s
SNN	$\sim$ 6M	88.87%	20m7s
SNN-mimic-8k	$\sim$ 250k	82.62%	11m29s
SNN-mimic-KL-8k	$\sim$ 250k	87.09%	12m10s

Tabelle: Performance of the models

$\rightarrow$ CNN better than DNN (as in paper)
$\rightarrow$ For the mimic model, KL loss is better
$\rightarrow$ Bottleneck layer makes training much faster

Accuracy and training cost

model	#parameters	accuracy	training cost
CNN	$\sim$ 210k	90.24%	16m30s
DNN	$\sim$ 10M	89.48%	42m33s
SNN	$\sim$ 6M	88.87%	20m7s
SNN-mimic-8k	$\sim$ 250k	82.62%	11m29s
SNN-mimic-KL-8k	$\sim$ 250k	87.09%	12m10s

Tabelle: Performance of the models

$\rightarrow$ CNN better than DNN (as in paper)
$\rightarrow$ For the mimic model, KL loss is better
$\rightarrow$ Bottleneck layer makes training much faster
$\rightarrow$ SNN has much more parameters than the mimic model - in the paper, it is the same

Discussion

Questions

- Why is the SNN better than the mimic model?

Questions

- Why is the SNN better than the mimic model?
  - data set

Questions

- Why is the SNN better than the mimic model?
  - data set
  - amount of parameters

Questions

- Why is the SNN better than the mimic model?
  - data set
  - amount of parameters
  - bottleneck layer

Questions

- Why is the SNN better than the mimic model?
  - data set
  - amount of parameters
  - bottleneck layer
  - loss function

Questions

- Why is the SNN better than the mimic model?
  - data set
  - amount of parameters
  - bottleneck layer
  - loss function
- How do the SNN and SNN mimic have the same amount of parameters in the paper?