Do We Need Zero Training Loss After Achieving Zero Training Error?

[Ishida, Takashi, et al., 2020 ICML]

https://arxiv.org/abs/2002.08709

김봉석, 서울과학기술대학교 데이터사이언스학과, bongseokkim@seoultech.ac.kr

SEOULTECH

Content

- Short review of paper
- Experiment **(obtain the same (or similar) results?)**
  - to reproduce paper`s result
- Improvement
  - Question 1
  - Question 2
- Summary

Main Question : Do We Need Zero Training Loss After Achieving Zero Training Error?

1. Overparameterized deep networks have the capacity to memorize training data with zero training error. Even after memorization, the training loss continues to approach zero

2. existing regularizers do not directly aim to avoid zero training loss

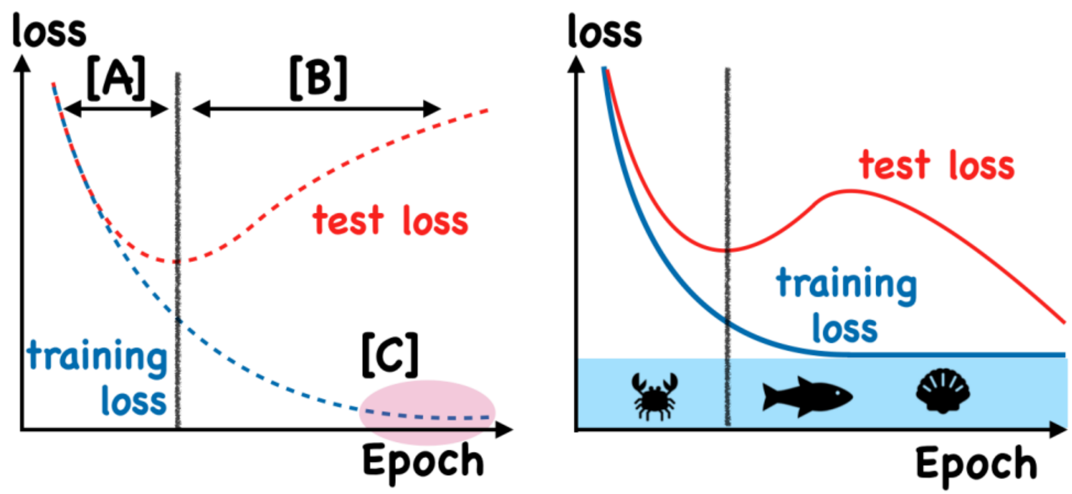
3. propose a direct solution called flooding that intentionally prevents further reduction of the training loss when it reaches a reasonably small value

Short review of paper

Even if, model memorized the training data completely with zero error
⟹ the training loss will decreasing to (near-)zero

Hypothethis : learning until **zero training loss** is hamful **!**

- Propose propose a direct solution (called flooding) that intentionally prevents further reduction of the training loss



(a) w/o Flooding (b) w/ Flooding

Short review of paper

With flexible models, $\hat{R}(\boldsymbol{g})$ wrt. a surrogate loss can easily become small if not zero

proposed training objective with flooding

$$\tilde{R}(\boldsymbol{g}) = |\hat{R}(\boldsymbol{g}) - \mathrm{b}| + \boxed{\mathrm{b}}$$

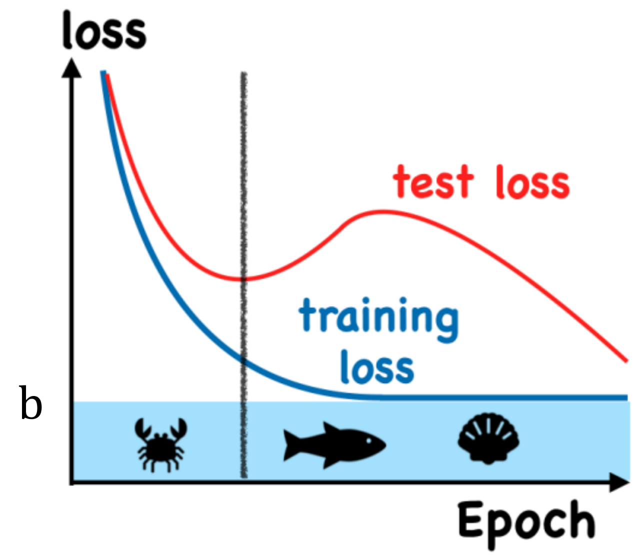
Denote flood level

If $\hat{R}(\boldsymbol{g}) > b \rightarrow \tilde{R}(\boldsymbol{g}) = \hat{R}(\boldsymbol{g})$ **(Gradient dencet)**

- Same direction as original

If $\hat{R}(\boldsymbol{g}) < b \rightarrow \tilde{R}(\boldsymbol{g}) = -\hat{R}(\boldsymbol{g}) + 2\mathrm{b}$ **(Gradient acent)**

- Opposite direction



(b) w/ Flooding

In practice, this will be performed with a mini-batch , compatible with any stochastic optimizers

Its implementation is extremely simple (additional one line of code)

$$\tilde{R}(\boldsymbol{g}) = \left|\hat{R}(\boldsymbol{g}) - \mathrm{b}\right| + \mathrm{b}$$

In Pytorch

```
outputs = model(inputs)
loss = criterion(outputs, labels)
flood = (loss-b).abs()+b  # This is it!
optimizer.zero_grad()
flood.backward()
optimizer.step()
```

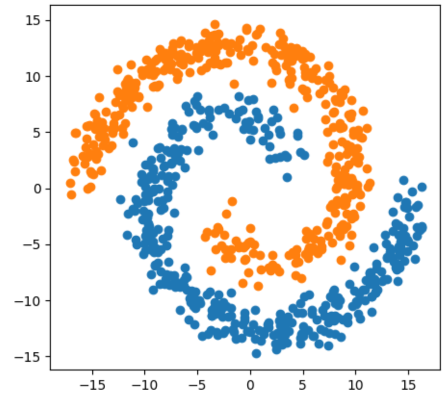
Question : How to select flood level b ?

- optimal flood level b is data/task dependent
- we can search for the optimal flood level by performing the exhaustive search (hyperparameter)

Experiment : can we obtain the same (or similar) results?

Synthetic Datasets (Artificial Dataset)

- Two Gaussians
- Spiral
- Sinusoid



[Spiral dataset from sklearn]

Setting

- Model : a five-hidden-layer NN with 500 units in each hidden layer with the ReLU
- Loss : logistic loss
- Optimizer : Adam,
- Epoch : 500
- Learning rate : 0.001
- Flood level : b ∈ $[0, 0.1 \ \ldots 0.5]$
- Label noise : Low(1%), Middle(5%), High(10%)

Experiment : can we obtain the same (or similar) results?

(A) Paper | (B) My

Data	Label Noise	Without Flooding	With Flooding	Chosen b	Without Flooding	With Flooding	Chosen B
Two Gaussians	Low	90.52%	92.13%	0.17	90.13%	91.69%	0.18
	Middle	84.79%	88.03%	0.22	83.56%	86.35%	0.23
	High	78.44%	83.59%	0.32	77.89%	80.62%	0.29

Data	Without Flooding	With Flooding	Chosen b	Without Flooding	With Flooding	Chosen B
CIFAR-10	90.52%	92.13%	0.17	90.13%	91.69%	0.18

Setting (Two Gaussain)

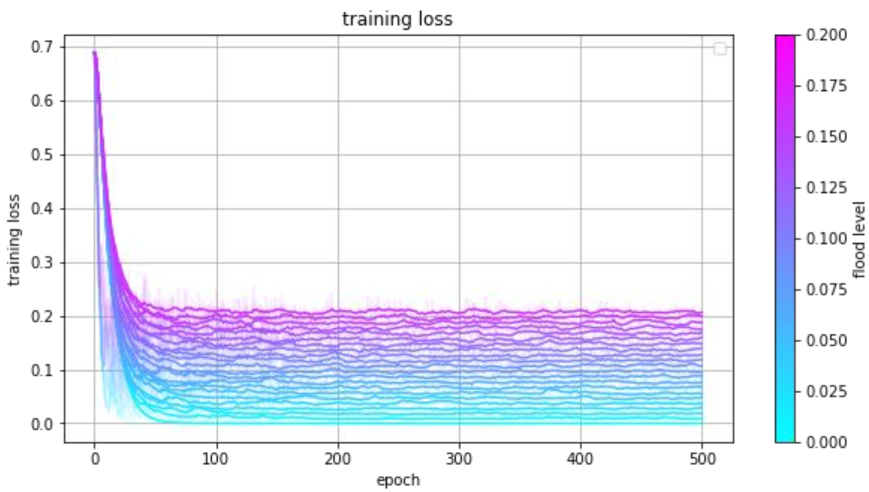
- Model : a five-hidden-layer NN with 500 units in each hidden layer with the ReLU
- Loss : logistic loss
- Optimizer : Adam,
- Epoch : 500
- Learning rate : 0.001
- Flood level : b ∈ [0, 0.01,0.02, 0.03 ... 0.5], conducted 50 experiment for hyperparmeter searching
- Label noise : Low(1%), Middle(5%), High(10%)

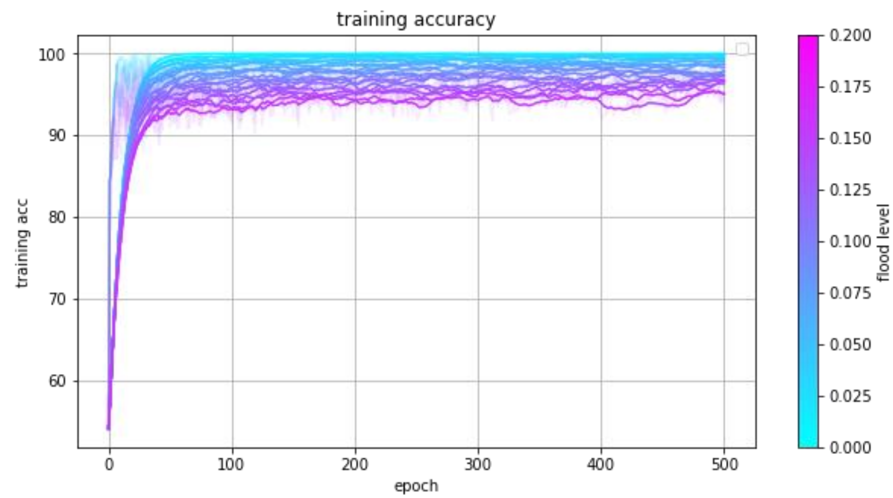
Setting (CIFAR-10)

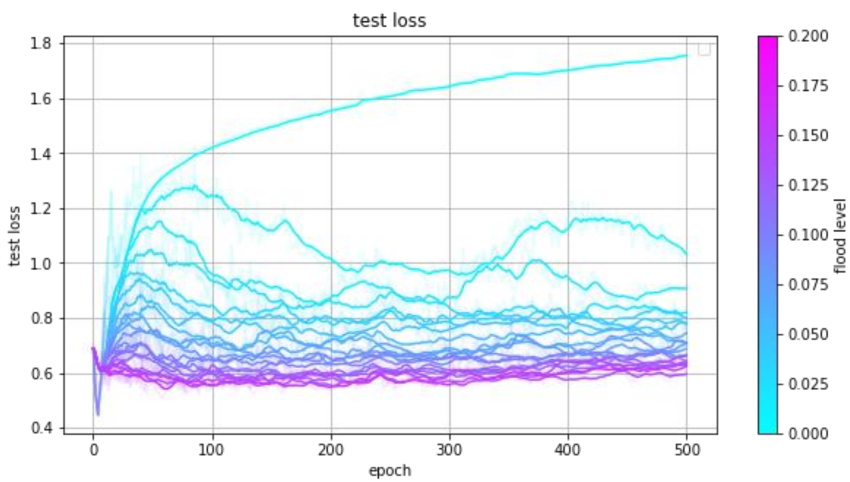
- Model : Resnet 44
- Agumentation : random crop , horizontal flip
- Optimizer : SGD
- Epoch : 500
- Learning rate : 0.001, learning rate decay (multiply by 0.1 after 250 400 epoch)
- Flood level : b ∈ [0, 0.01,0.02, 0.03 ... 0.1], conducted 50 experiment for hyperparmeter searching

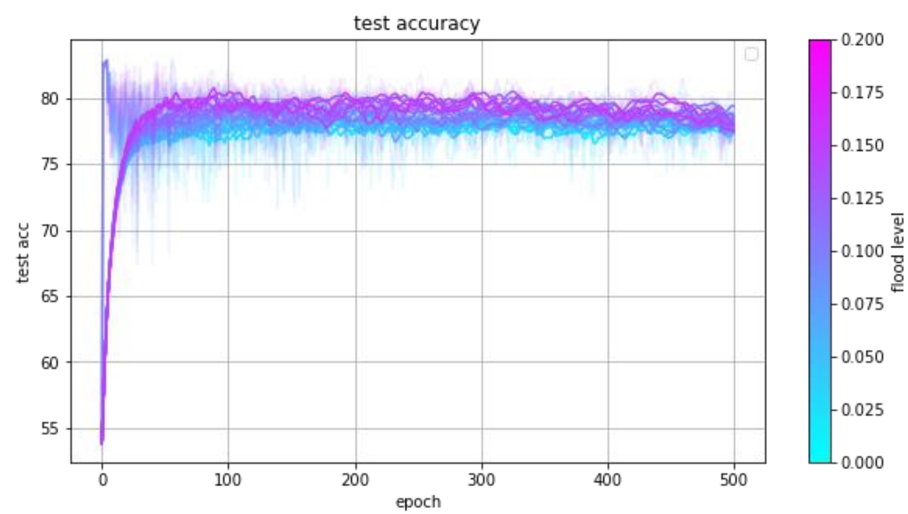
Experiment : can we obtain the same (or similar) results?

Two Gaussian (Noise Level -High)









Experiment : can we obtain the same (or similar) results?

(A) Paper | (B) My

Data	Label Noise	(A) Paper: Without Flooding	(A) Paper: With Flooding	(A) Paper: Chosen b	(B) My: Without Flooding	(B) My: With Flooding	(B) My: Chosen B
Two Gaussians	Low	90.52%	92.13%	0.17	90.13%	91.69%	0.18
	Middle	84.79%	88.03%	0.22	83.56%	86.35%	0.23
	High	78.44%	83.59%	0.32	77.89%	80.62%	0.29

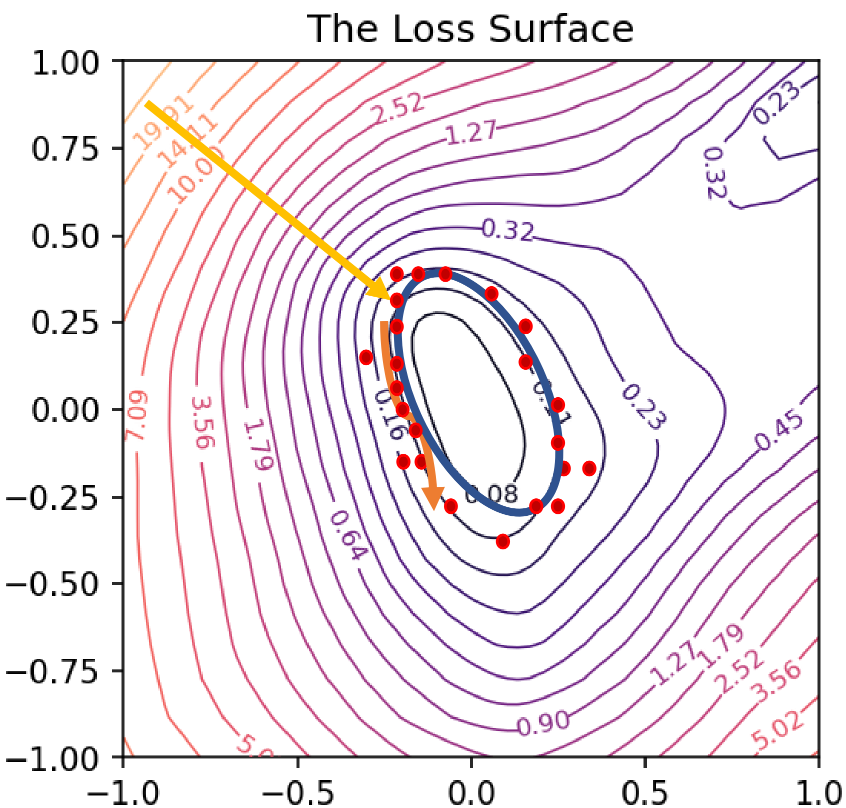
Data	Without Flooding	With Flooding	Chosen b	Without Flooding	With Flooding	Chosen B
CIFAR-10	90.52%	92.13%	0.17	90.13%	91.69%	0.18

Question : Can we obtain the same (or similar) results?

➡ Answer : Yes

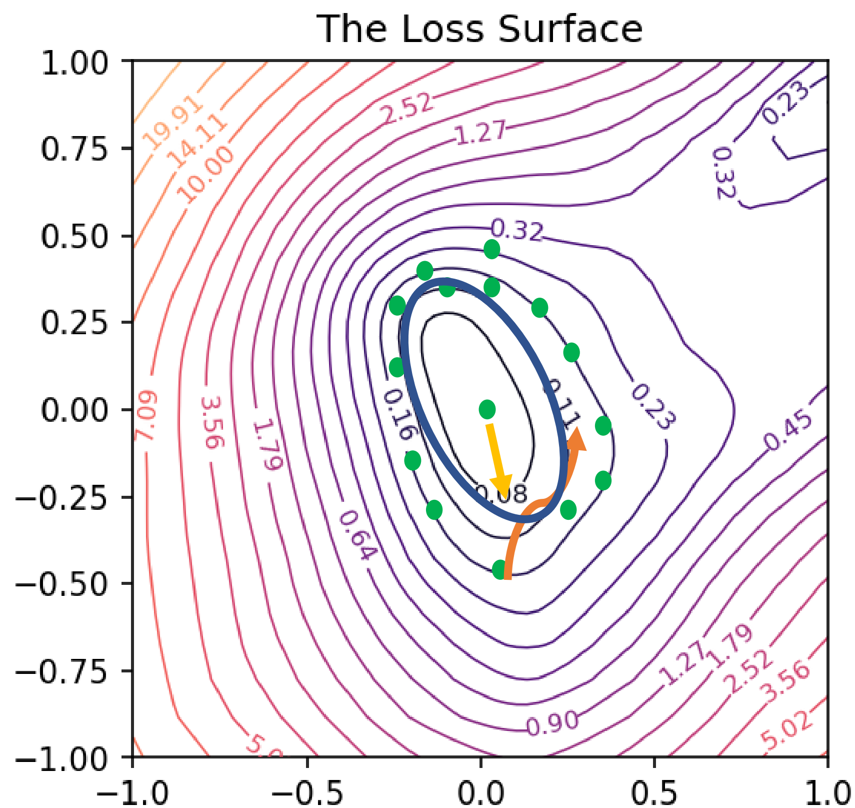
But
- **For double decents, valid only for certain condition**.
  - Ex) CIFAR-10, agumentation + learning rate decay (0)
  - CIFAR-10, agumentation + learning rate decay+ weight decay (x)
  - EX) Two Gaussian, <u>label noise must be high</u>
- **hard to find optimal flood level b**
  - Ex) CIFAR-10, it takes ~= 50 hour to search flood level b

Question 1 : Search from inside to Outside loss surface ?



Flood

- Search from outside to inside(flood level)
  - But never reach inside flood level



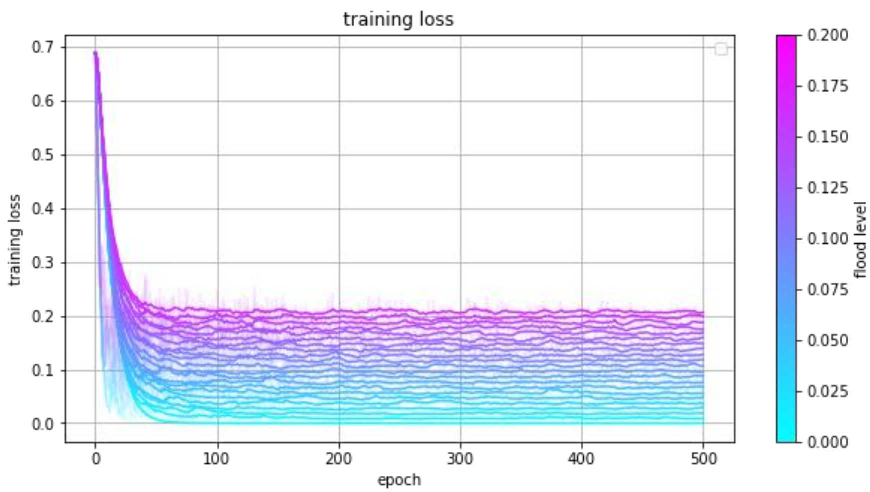
Try

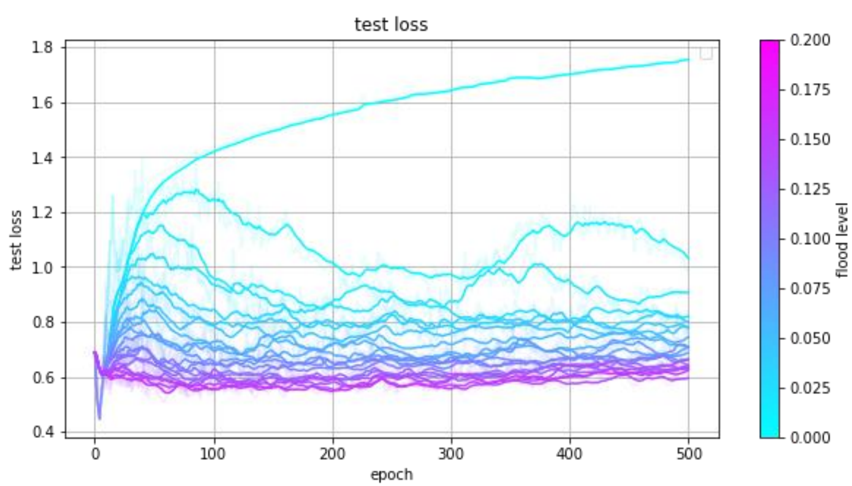
- Search from inside(zero tr.error) to outside(flood level)
  - Search inside the flood level at least once

Improvements – Question 1

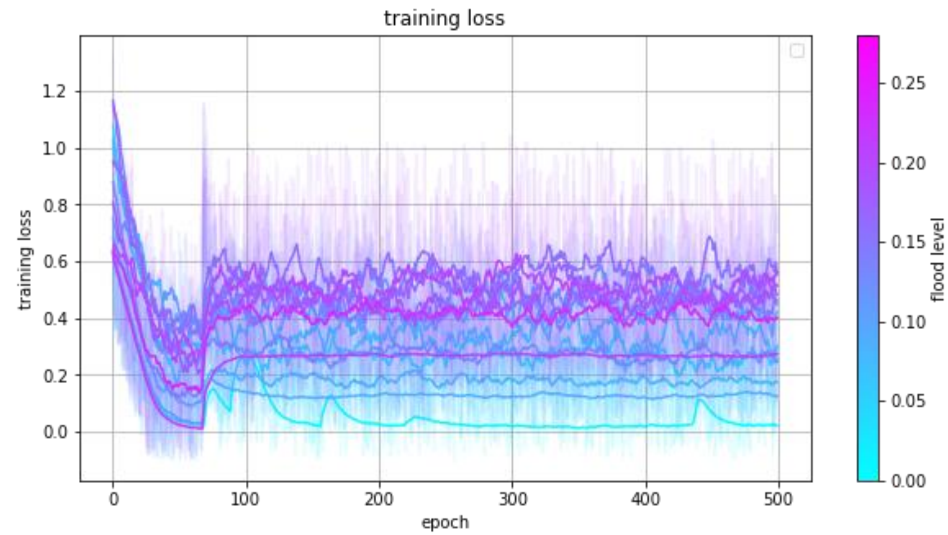
Two Gaussian (Noise Level -High)

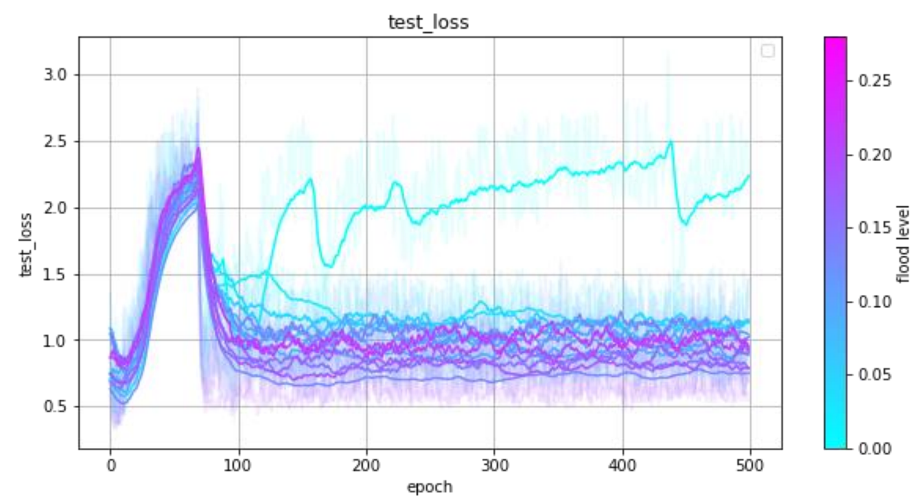
Flood





Try



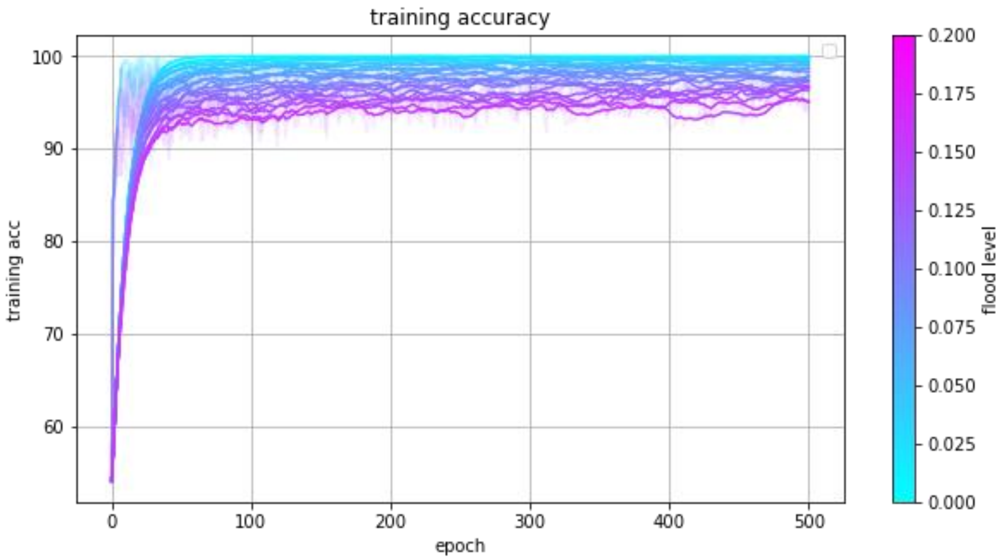


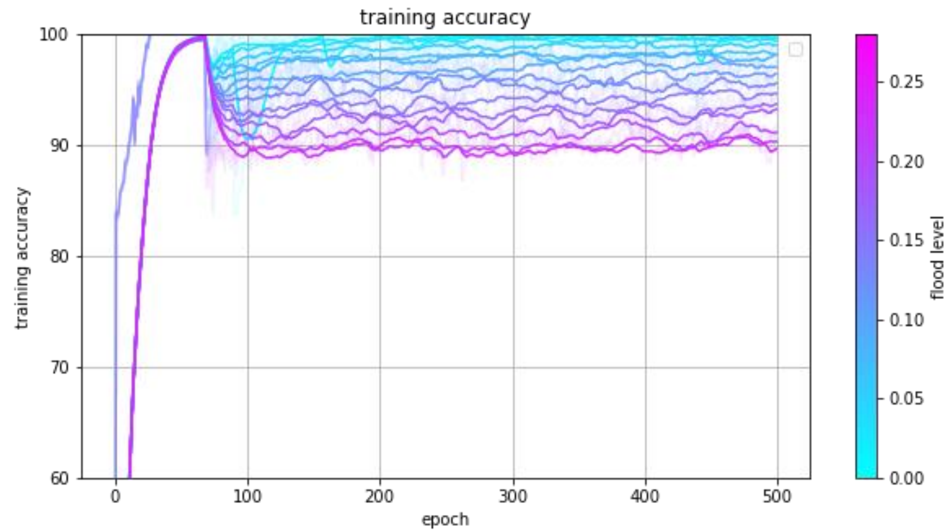
Improvements – Question 1

Two Gaussian (Noise Level -High)

Flood | **Try**

Data	Label Noise	Without Flooding	With Flooding	Chosen b	Without Flooding	With Flooding	Chosen B
Two Gaussians	High	77.89%	80.62%	0.29	77.89%	79.89%	0.26



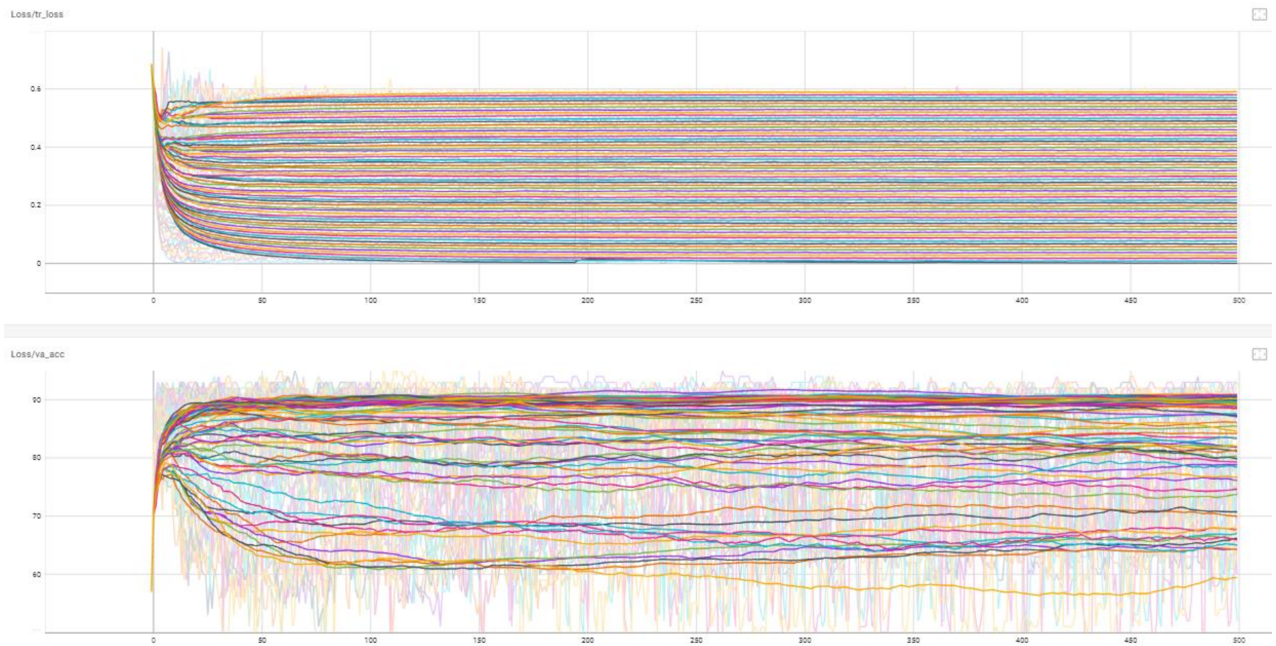


- Why worse ?
  - There is a tendency to lower the training acc itself which lead to lower test acc

Improvements – Question 2

Motivation

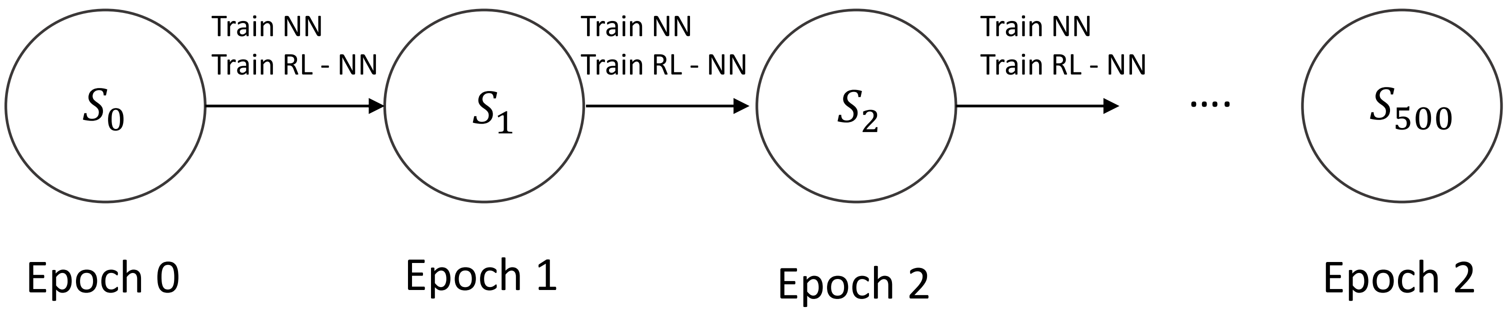
- **hard to find optimal flood level b**
  - 500 epoch x 50 experiment –
  - It takes too much time and cost



Question 2 : how to find find optimal flood level , b smarter way.

Improvements- Question 2

Try : hyperparmeter search using Reinforcement learning!
using Policy Gradient



Reward = validation accuracy
State = [traing loss, traing acc, val_loss, val_acc]
Action = flood level , b $\in [0, 0.01, 0.02, 0.03 \ldots 0.1]$

Failed →

- **Hard to train RL agent**
- It takes too much time.
  - Ex) CIFAR-10, it takes ~= 50 hour

Improvements- Question 2

Try: Heuristic search to find flood level

while zero traing error **do** gradient decent
  $x' \leftarrow 0$
  $b \leftarrow 0.0$
  while epoch **do** optimizer step
    $criterion = |loss - b| + b$
    $x' \leftarrow 0$
    $b \leftarrow 0.0$
    if $x \geq x' + \lambda$ **then**
      $x' \leftarrow x$
      $b \leftarrow clip(b + \alpha, 0, 0.5)$
    else
      if $x < x' - \lambda$ **then**
        $x' \leftarrow x$
        $b \leftarrow clip(b - \alpha, 0, 0.5)$
      end if
    end if

b: flood level

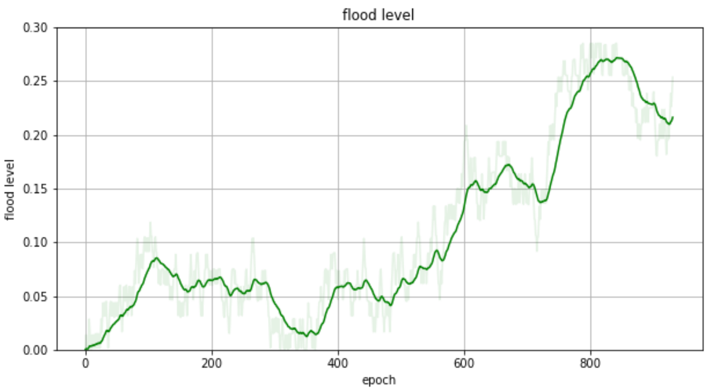
x', x :validation accuracy

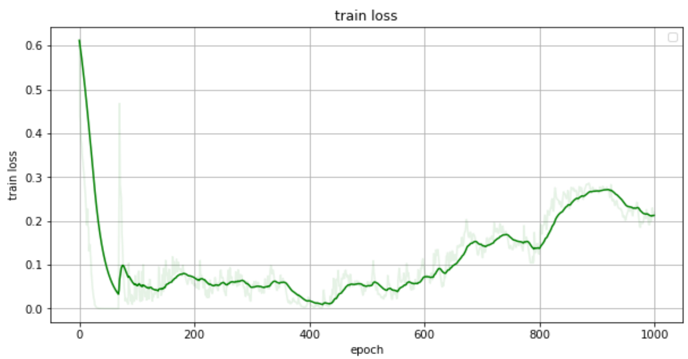
α: hyperparmeter for adjusting flood level

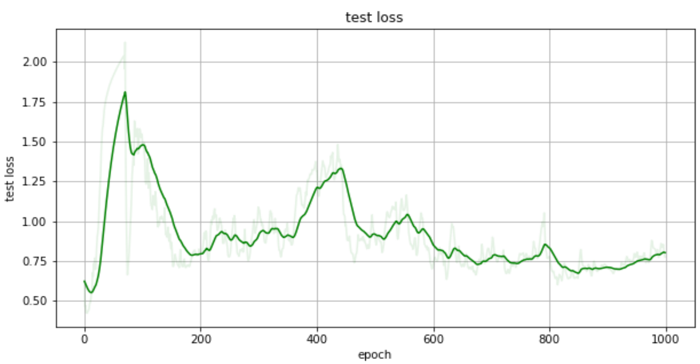
$* \ adjust\ flood\ level\ b\ *$

shrink	same	expand
$x' - \lambda$	x	$x' + \lambda$

Improvements- Question 2







$$* \ adjust\ flood\ level\ b \ *$$

shrink	same	expand
$x' - \lambda$	x	$x' + \lambda$

Data	Without Flooding	With Flooding	Chosen b
Flood	77.89%	80.62%	0.29
Try1:(backward)	77.89%	79.89%	0.29
Try2:(adaptive)	77.89%	79.12%	0.21

- Hureistic `s Final flood level was 0.21 (optimal 0.29)
- It is better than without flooding, but worse than flooding
- But, takes less time

Summary

Try to reproduce Experiment (two gaussian/ CIFAR10)

- obtain similar result
- to observe double decents curve it need certain condition.

Try Improvement -1

- <u>Search from inside(zero tr.error) to outside(flood level)</u>
  - It is better than without flooding, but worse than flooding
  - Why worse? : tendency to lower the training acc -> lead to lower test acc

Try Improvement -2 (hard to find flood level)

- Reinforcement Learning Approach (failed)
- Heuristic search
  - It is better than without flooding, but worse than flooding
  - But, takes less time