- **The truth table of Weight and Gradient**

	Weight 大	**Weight 小**
Gradient 大	case 1	case 2
Gradient 小	case 3	case 4

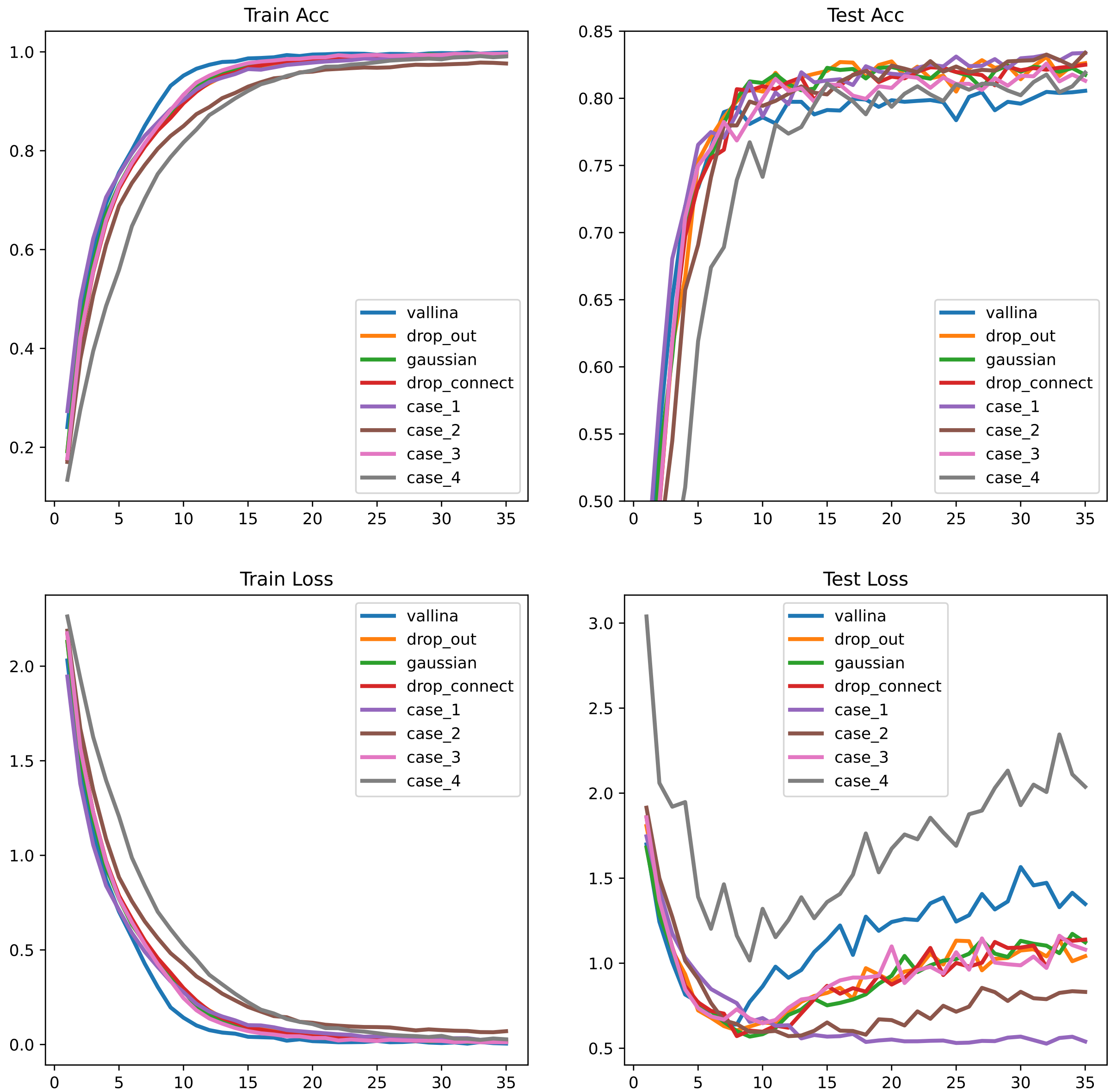
New DropConnect Method: Consider Initial Drop rate, Weight Value, and Gradient Value.

- **Method:**
  - **Each Weight has it initial Dropout Rate p1, Weight Drop Rate p2, Gradient Drop Rate p3**
  - $\boldsymbol{p1 = InitialDropConnectRate}$
  - $\boldsymbol{p2 = \sigma(Weight)}$
  - $\boldsymbol{p3 = \sigma(Gradient)}$
  - $\boldsymbol{FinalDropRate = RandomMask(p1 + p2 + p3)}$

- Initial DropRate=0.3, Weight=0.35, Gradient=0.35

Model	Best Test Acc	Best Train Acc
AlexNet(Vallina)	81.27 ± 0.24	99.953 ± 0.005
AlexNet(Dropout)	83.32 ± 0.21	99.25 ± 0.22
AlexNet(DropConnect)	82.89 ± 0.34	99.31 ± 0.17
AlexNet(GaussianDrop)	82.94 ± 0.44	99.50 ± 0.03
AlexNet(w大,gd大)	81.97 ± 0.04	99.14 ± 0.08
AlexNet(w小,gd大)	82.39 ± 0.19	99.6 ± 0.11
AlexNet(w大,gd小)	83.16 ± 0.2	97.77 ± 0.14
AlexNet(w小,gd小)	83.44 ± 0.09	99.16 ± 0.05

- **Train acc and so on.**

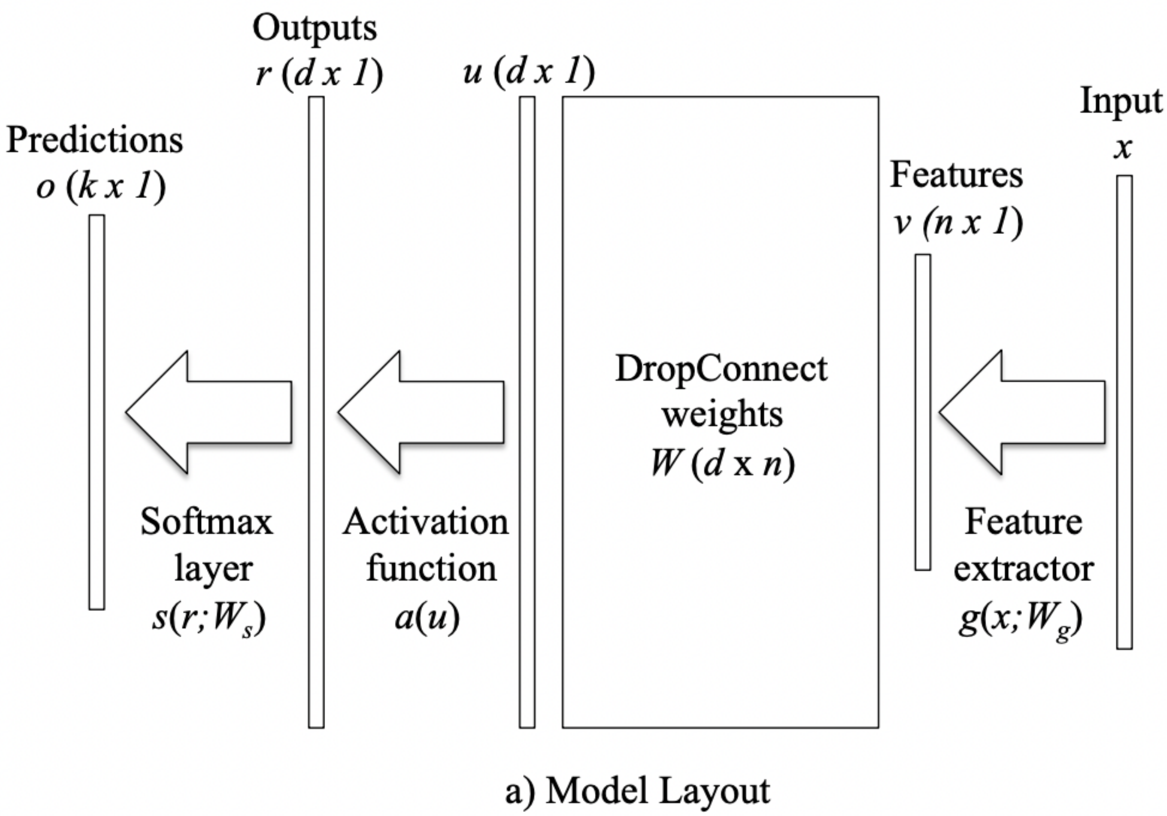


01/05

- **New DropConnect Method**
  - **Consider Initial Drop rate, Weight Value, and Gradient Value.**

	Weight Big	**Weight Small**
Gradient Big	case a	case b
Gradient Small	case c	case d

- **DropConnect**
  - $r = a((M \star W)v)$
  - **Outlook:**



- **Case a: 砍w大, gd 大**
  - Each Weight has it own Drop Rate p1, Weight Drop Rate p2, Gradient Drop Rate p3
  - $p1 = InitialDropRate$
  - $W = abs(W)$, $GD = abs(GD)$
  - $W = \frac{W-\mu}{\sigma}$
  - $GD = \frac{GD-\mu}{\sigma}$
  - $p2 = (Sigmoid(W) \geq 0.5)$
  - $p3 = (Sigmoid(GD) \geq 0.5)$
  - $FinalDropRate\ P = p1 + \alpha \times p2 + \beta \times p3$
  - $Randomsample\ M:\ mask \sim\ (P \geq U[0,1])$

- **Case b: 砍w小, gd 大**
  - Each Weight has it initial Drop Rate p1, Weight Drop Rate p2, Gradient Drop Rate p3
  - $\boldsymbol{p1 = InitialDropRate}$
  - $\boldsymbol{W = abs(W) \quad , \quad GD = abs(GD)}$
  - $\boldsymbol{W = \frac{W-\mu}{\sigma}}$
  - $\boldsymbol{W = W \times (-1)}$
  - $\boldsymbol{GD = \frac{GD-\mu}{\sigma}}$
  - $\boldsymbol{p2 = (Sigmoid(W)} \geq 0.5)$
  - $\boldsymbol{p3 = (Sigmoid(GD)} \geq 0.5)$
  - $\boldsymbol{FinalDropRate\ P = p1 + \alpha \times p2 + \beta \times p3}$
  - $\boldsymbol{Randomsample\ M:\ mask \sim\ (P \geq U[0,1]})$

- **Initial DropRate=0.3, Weight=0.05, Gradient=0.35**

Model	**Best Test Acc**	**Best Train Acc**
AlexNet(Vallina)	81.27 ± 0.24	99.953 ± 0.005
AlexNet(Dropout)	83.32 ± 0.21	99.25 ± 0.22
AlexNet(DropConnect)	82.89 ± 0.34	99.31 ± 0.17
AlexNet(GaussianDrop)	82.94 ± 0.44	99.50 ± 0.03
AlexNet(w大,gd大)	82.61 ± 0.09	99.69 ± 0.05
AlexNet(w小,gd大)	82.15 ± 0.2	99.58 ± 0.14
AlexNet(w大,gd小)	83.6 ± 0.19	98.772 ± 0.11
AlexNet(w小,gd小)	83.36 ± 0.04	98.9 ± 0.08

- **Initial DropRate=0.3, Weight=0.35, Gradient=0.05**

Model	**Best Test Acc**	**Best Train Acc**
AlexNet(Vallina)	81.27 ± 0.24	99.953 ± 0.005
AlexNet(Dropout)	83.32 ± 0.21	99.25 ± 0.22
AlexNet(DropConnect)	82.89 ± 0.34	99.31 ± 0.17
AlexNet(GaussianDrop)	82.94 ± 0.44	99.50 ± 0.03
AlexNet(w大,gd大)	82.47 ± 0.03	99.18 ± 0.09
AlexNet(w小,gd大)	82.52 ± 0.05	99.63 ± 0.12
AlexNet(w大,gd小)	83.03 ± 0.17	99.42 ± 0.05
AlexNet(w小,gd小)	82.48 ± 0.03	99.47 ± 0.02

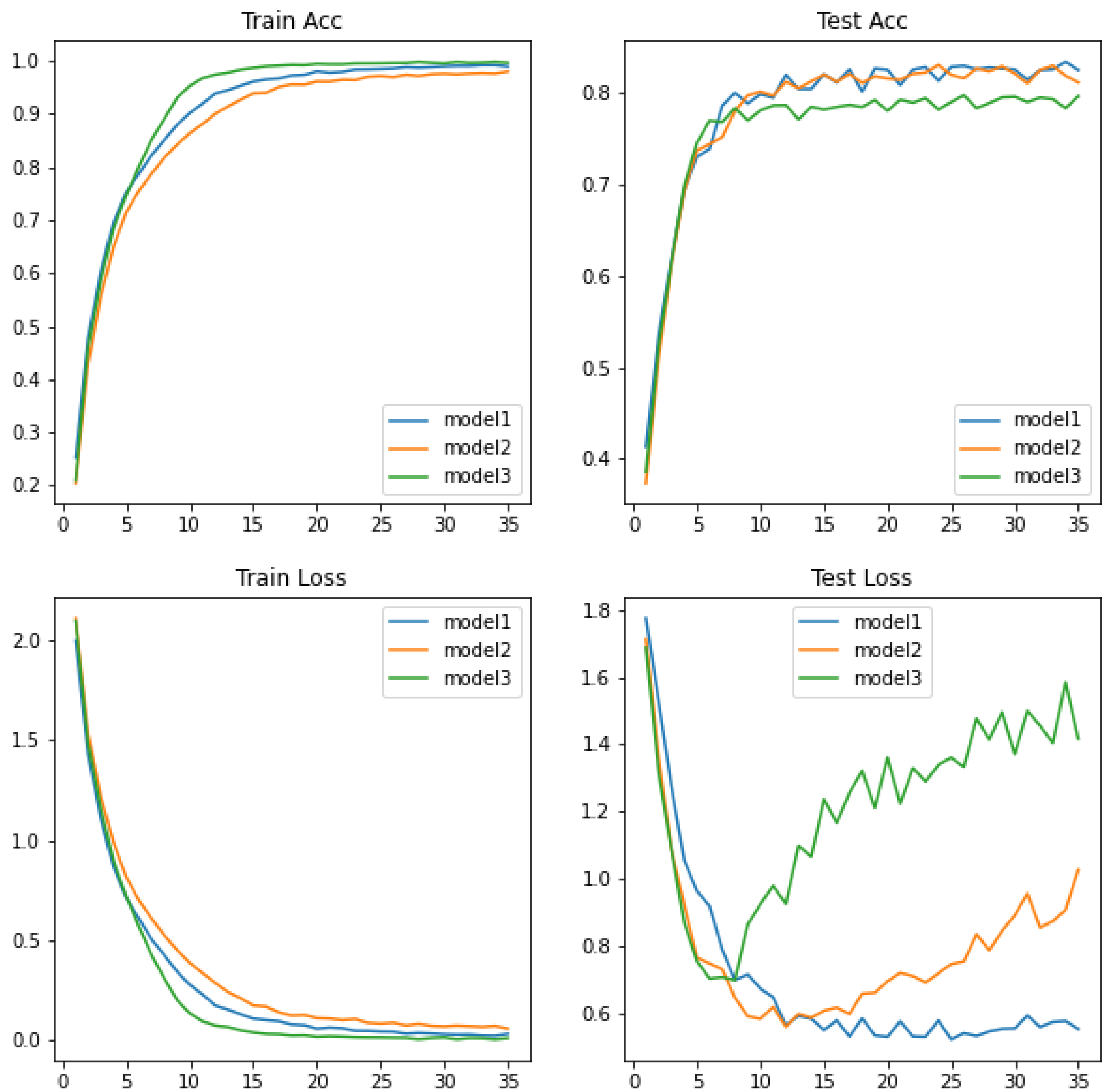
- **Initial DropRate=0.25, Weight=0.35, Gradient=0.35**

Model	Best Test Acc	Best Train Acc
VGG16(Vallina)	94.02	99.972
VGG16(DropConnect p=0.5)	93.89	99.95
VGG16(DropConnect p=0.35)	94.09	99.97
VGG16(w大,gd大)	93.86	99.93
VGG16(w小,gd大)	93.99	99.89
VGG16(w大,gd小)	94.07	99.94
VGG16(w小,gd小)	94.15	99.97

01/12

- **針對case c(砍gd小, 砍w小) 和case d(砍gd小, 砍w大)做實驗**
- **Case c: 砍gd小, w小 – ADD Weight Drop and Gradient Drop Rate**
  - Each Weight has it own Drop Rate p1, Final Weight Drop Rate p2, Final Gradient Drop Rate p3
  - **Hyperparameters:** $InitialDropRate(p1), WeightRatio(\alpha),$
    $GD\ Ratio(\beta), WeightDropRate(w\_drop\_rate), GdDropRate(gd\_drop\_rate)$
  - $p1 = InitialDropRate$
  - $W = abs(W)\ ,\ GD = abs(GD)$
  - $W = \frac{W-\mu}{\sigma}$
  - $GD = \frac{GD-\mu}{\sigma}$
  - $Sigmoid\_w = Sigmoid(W)$
  - $Sigmoid\_w = Sigmoid\_w - (0.5 - w\_drop\_rate)$
  - $Sigmoid\_gd = Sigmoid(GD)$
  - $Sigmoid\_gd = Sigmoid\_gd - (0.5 - gd\_drop\_rate)$
  - $p2 = Sigmoid\_w \geq 0.5$
  - $p3 = Sigmoid\_gd \geq 0.5$
  - $FinalDropRate\ P = p1 + \alpha \times p2 + \beta \times p3$
  - $Randomsample\ M:\ mask \sim\ (P \geq U[0,1])$
- **train cifar100, 換network(dropconnect論文的cnn 跟fullyconnect layer model**

- **init=0.3, w_ratio=0.35, gd_ratio=0.35, w_droprate=0.45, gd_droprate=0.45**



01/19

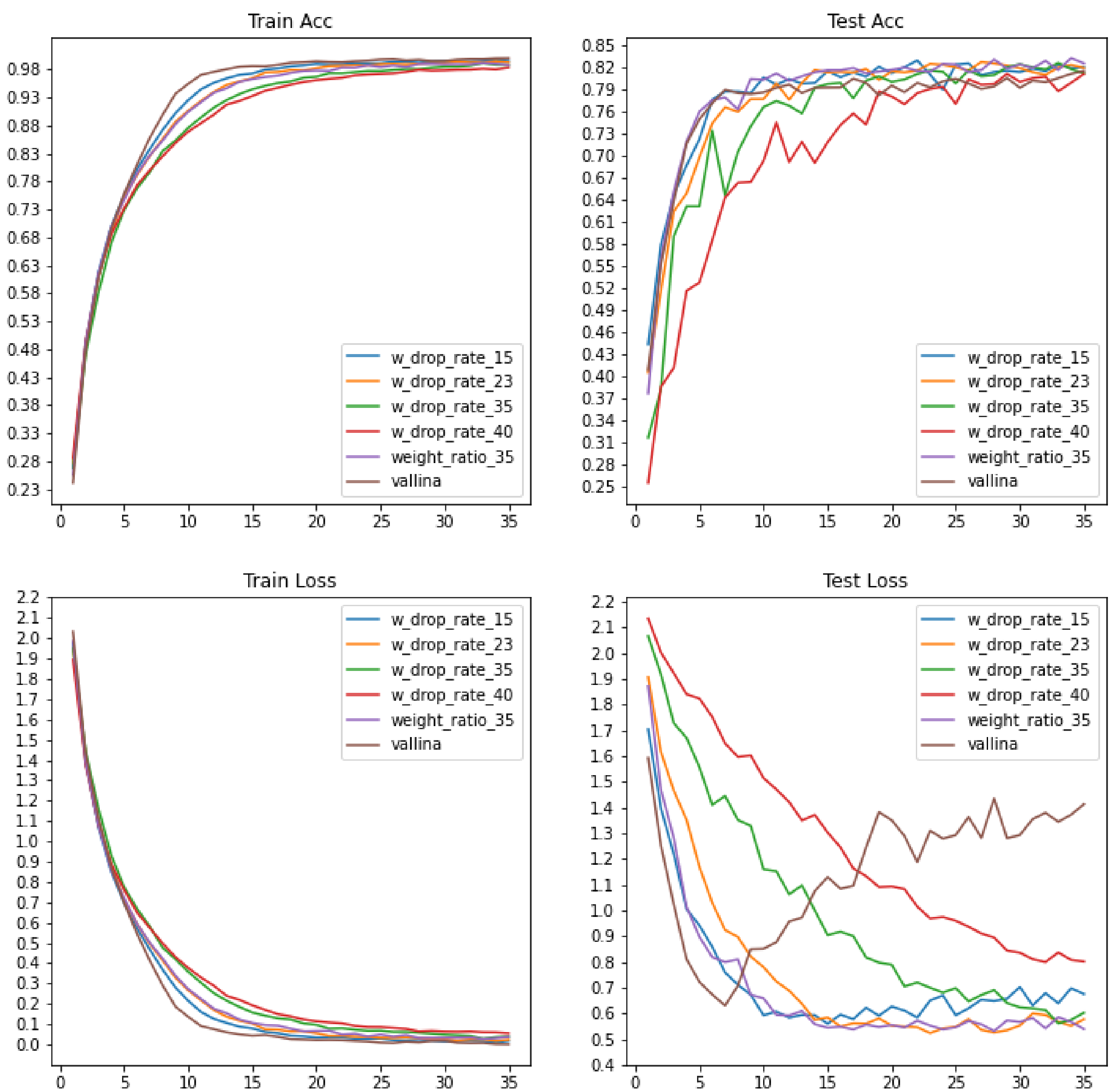
- **w_drop_rate:**
  - Each Weight has it own Drop Rate p1, Final Weight Drop Rate p2, Final Gradient Drop Rate p3
  - **Hyperparameters:** $InitialDropRate(p1), WeightDropRate(w\_drop\_rate),$ $GdDropRate(gd\_drop\_rate)$
  - $\boldsymbol{p1 = InitialDropRate}$
  - $\boldsymbol{W = abs(W)\ ,\ GD = abs(GD)}$
  - $\boldsymbol{W = \frac{W-\mu}{\sigma}}$
  - $\boldsymbol{GD = \frac{GD-\mu}{\sigma}}$
  - $\boldsymbol{Sigmoid\_w = Sigmoid(W)}$
  - $\boldsymbol{Sigmoid\_w = Sigmoid\_w - (0.5 - w\_drop\_rate)}$
  - $\boldsymbol{Sigmoid\_gd = Sigmoid(GD)}$
  - $\boldsymbol{Sigmoid\_gd = Sigmoid\_gd - (0.5 - gd\_drop\_rate)}$
  - $\boldsymbol{FinalDropRate\ P = p1 + Sigmoid\_w + Sigmoid\_gd}$
  - $\boldsymbol{Randomsample\ M:\ mask \sim\ (P \geq U[0,1])}$

- **weight_ratio:**
  - Each Weight has it own Drop Rate p1, Final Weight Drop Rate p2, Final Gradient Drop Rate p3
  - **Hyperparameters:** $InitialDropRate(p1), WeightRatio(\alpha),$ $GD\ \ Ratio(\beta)$
  - $\boldsymbol{p1 = InitialDropRate}$
  - $\boldsymbol{W = abs(W)\ ,\ GD = abs(GD)}$
  - $\boldsymbol{W = \frac{W-\mu}{\sigma}}$
  - $\boldsymbol{GD = \frac{GD-\mu}{\sigma}}$
  - $\boldsymbol{Sigmoid\_w = Sigmoid(W)}$
  - $\boldsymbol{Sigmoid\_gd = Sigmoid(GD)}$
  - $\boldsymbol{p2 = Sigmoid\_w \geq 0.5}$
  - $\boldsymbol{p3 = Sigmoid\_gd} \geq 0.5$
  - $\boldsymbol{FinalDropRate\ P = p1 + \alpha \times p2 + \beta \times p3}$
  - $\boldsymbol{Randomsample\ M:\ mask \sim\ (P \geq U[0,1])}$

- **choose w_drop_rate or weight_ratio:**

Model	Train Acc	Test Acc	Train Loss	Test Loss
w_drop_rate_15	99.4 ± 0.3	82.5 ± 0.4	0.019 ± 0.009	0.634 ± 0.027
w_drop_rate_23	99.2 ± 0.2	82.9 ± 0.3	0.024 ± 0.006	0.540 ± 0.007
w_drop_rate_35	98.5 ± 0.3	82.7 ± 0.4	0.045 ± 0.010	0.595 ± 0.021
w_drop_rate_40	98.1 ± 0.2	81.2 ± 0.6	0.059 ± 0.006	0.794 ± 0.040
weight_ratio_35	99.1 ± 0.1	83.3 ± 0.3	0.028 ± 0.003	0.550 ± 0.015
vallina	99.8 ± 0.1	81.0 ± 0.3	0.006 ± 0.005	0.006 ± 0.005

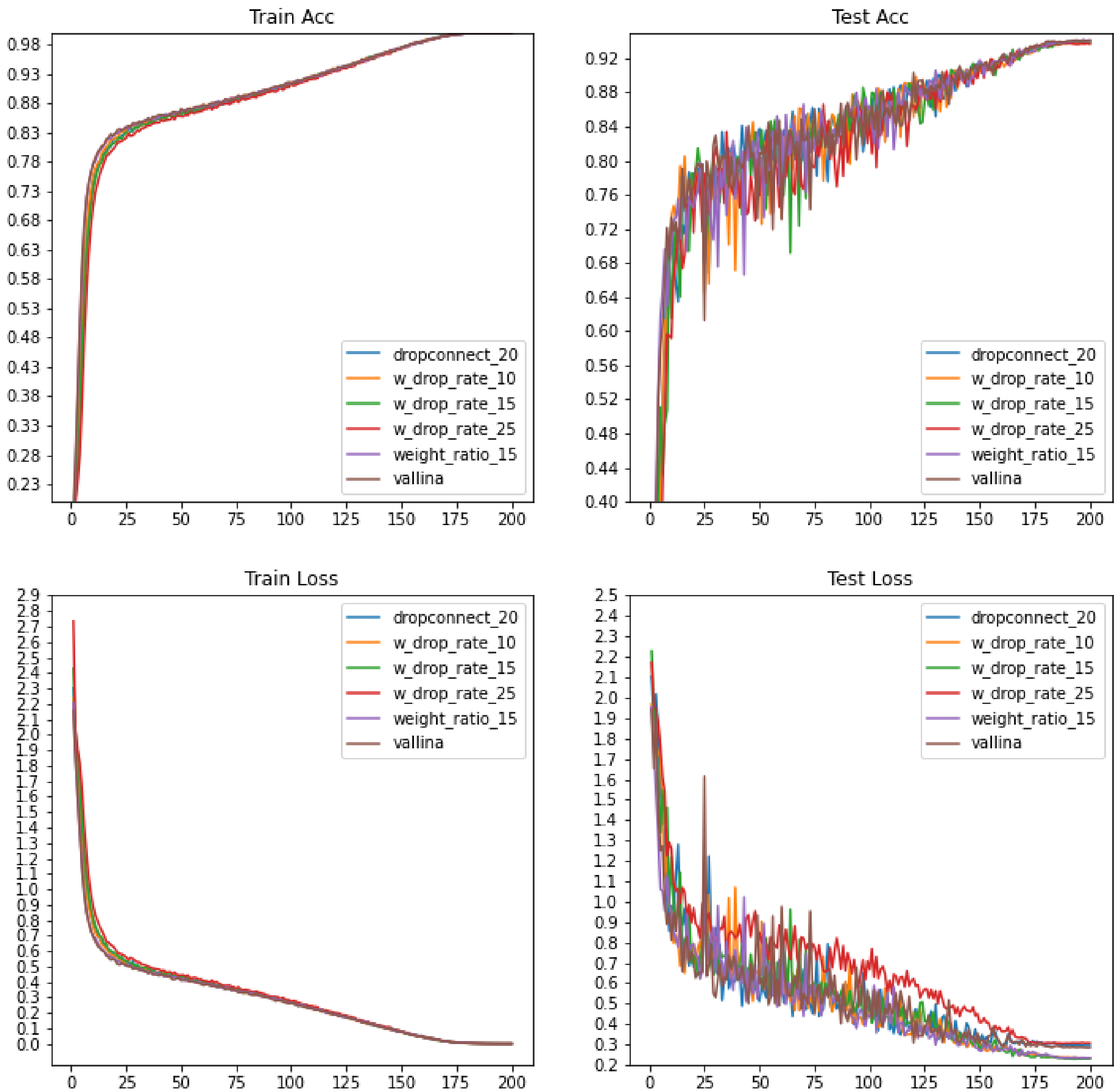
- **Result Image**



- **choose w_drop_rate or weight_ratio (in VGG16):**
  - Problem: w_drop_rate在random initialize的weight 會一直有NAN,
    但initialize zeros 時沒有

Model	Train Acc	Test Acc	Train Loss	Test Loss
dropconnect_20	99.97 ± 0.00	94.09 ± 0.04	0.0014 ± 0.0001	0.2896 ± 0.0045
w_drop_rate_10	99.98 ± 0.00	93.92 ± 0.04	0.013 ± 0.0002	0.2331 ± 0.0013
w_drop_rate_15	99.97 ± 0.00	94.02 ± 0.03	0.0014 ± 0.0001	0.2256 ± 0.026
w_drop_rate_25	99.97 ± 0.00	93.78 ± 0.04	0.0015 ± 0.0001	0.3086 ± 0.0039
weight_ratio_15	99.98 ± 0.01	94.17 ± 0.01	0.0013 ± 0.0000	0.2356 ± 0.0009
vallina	99.96 ± 0.00	94.08 ± 0.04	0.0016 ± 0.0001	0.2882 ± 0.0031

- **Result Image**


Train Acc
Test Acc
Train Loss
Test Loss
dropconnect_20
w_drop_rate_10
w_drop_rate_15
w_drop_rate_25
weight_ratio_15
vallina

- 模擬原dropconnect論文的實驗(Simple CNN)
  - Simple CNN(Architecture):

```
def __init__(self, add_layer=None, drop_model=None, drop_connect=False, normal_drop=False, p=0.2
    super(SimpleCnn, self).__init__()
    self.cnn1 = nn.Conv2d(3, 32, kernel_size=5, padding=2, stride=1)
    self.cnn2 = nn.Conv2d(32, 32, kernel_size=5, padding=2, stride=1)
    self.cnn3 = nn.Conv2d(32, 64, kernel_size=5, padding=2, stride=1)
    self.maxpooling = nn.MaxPool2d(3, stride=2)
    self.avgpooling = nn.AvgPool2d(3, stride=2)
    self.linear1 = nn.Linear(64*3*3, 64)
    self.linear2 = nn.Linear(64, 10)
```

- 原paper:

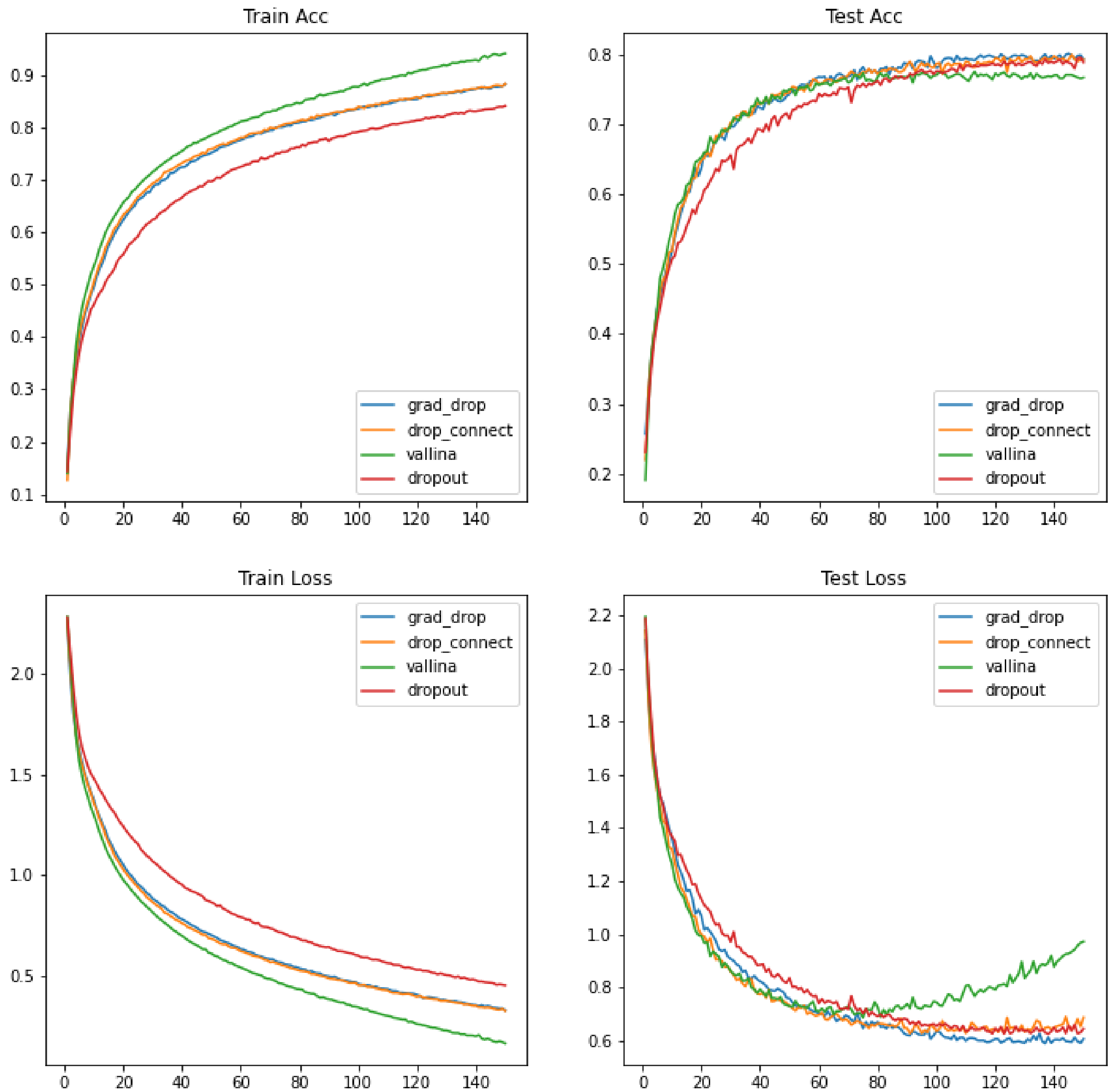
model	error(%)
No-Drop	23.5
Dropout	19.7
DropConnect	**18.7**

Table 4. CIFAR-10 classification error using the simple feature extractor described in (Krizhevsky, 2012)(layers-80sec.cfg) and with no data augmentation.

- 比較dropout, dropconnect等:

Model	Train Acc	Test Acc	Train Loss	Test Loss
weight_ratio	87.48	80.12	0.3508	0.5973
drop_connect	88.01	79.96	0.3407	0.6402
vallina	89.77	77.52	0.2937	0.7672
dropout	83.98	79.54	0.4585	0.6318

- **Result Image**



- 原paper:

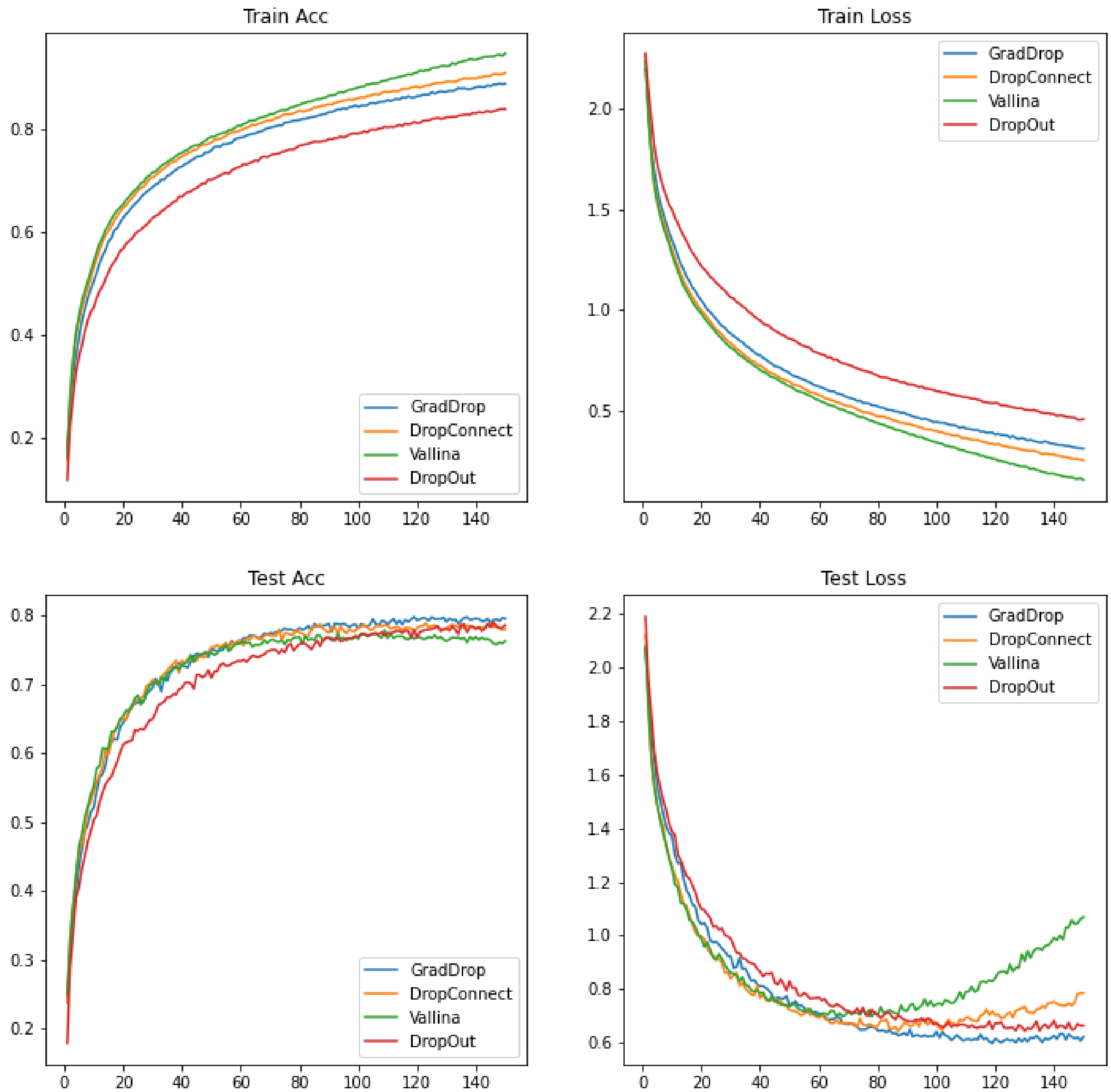
model	error(%)
No-Drop	23.5
Dropout	19.7
DropConnect	**18.7**

Table 4. CIFAR-10 classification error using the simple feature extractor described in (Krizhevsky, 2012)(layers-80sec.cfg) and with no data augmentation.

- 比較dropout, dropconnect等:

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	88.47 ± 1.74	77.25 ± 0.34	0.331 ± 0.050	0.764 ± 0.049
DropOut	83.88 ± 0.38	78.90 ± 0.21	0.455 ± 0.009	0.662 ± 0.013
DropConnect	87.49 ± 0.90	78.94 ± 0.29	0.352 ± 0.026	0.678 ± 0.012
GradDrop	87.64 ± 0.84	80.01 ± 0.16	0.349 ± 0.025	0.601 ± 0.006

- **Result Image**



- DropConnect Paper

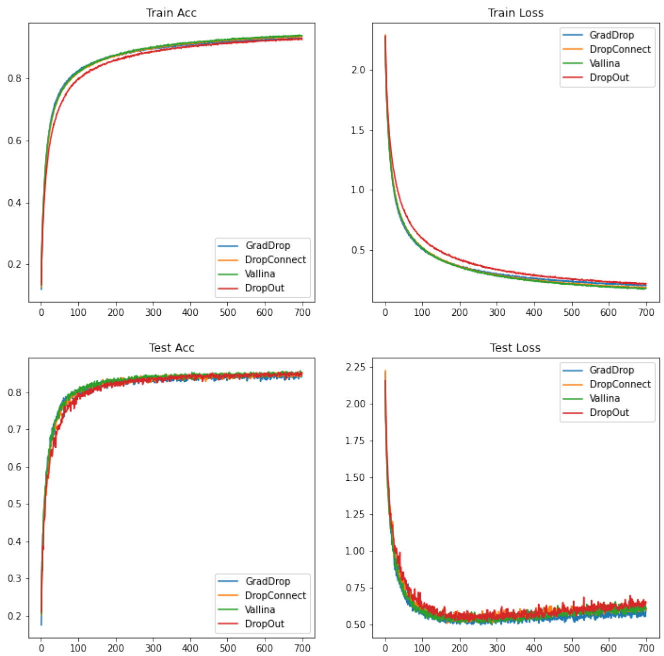
model	error(%) 5 network	voting error(%)
No-Drop	11.18± 0.13	10.22
Dropout	11.52± 0.18	9.83
DropConnect	11.10± 0.13	**9.41**

Table 5. CIFAR-10 classification error using a larger feature extractor. Previous state-of-the-art is 9.5% (Snoek et al., 2012). Voting with 12 DropConnect networks produces an error rate of **9.32%**, significantly beating the state-of-the-art.

- 與DropConnect實驗相比較

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	93.18 ± 0.73	85.53 ± 0.21	0.191 ± 0.019	0.579 ± 0.018
DropOut	91.95 ± 0.54	85.39 ± 0.21	0.238 ± 0.015	0.593 ± 0.032
DropConnect	92.69 ± 0.41	85.59 ± 0.20	0.208 ± 0.011	0.574 ± 0.012
GradDrop	92.63 ± 0.22	85.25 ± 0.18	0.211 ± 0.006	0.555 ± 0.014

- **Result Image**



- 還有其他的lr_scheduler沒有加:

- Need LocalConnectedLayer:

  他裡面用到的一種類似CNN(不共享weight)的方法，有用pytorch上的但好像有一些錯誤且與論文不太一樣

- 實驗差距好像比較小，論文有使用Voting Error:

model	error(%) 5 network	voting error(%)
No-Drop	2.26 ± 0.072	**1.94**
Dropout	2.25 ± 0.034	1.96
DropConnect	2.23 ± 0.039	**1.94**

Table 6. SVHN classification error. The previous state-of-the-art is *2.8%* (Zeiler and Fergus, 2013).

- 正在試他其他實驗和reference裡面其他論文的實驗:

03/02

- DropConnect Paper

model	error(%) 5 network	voting error(%)
No-Drop	11.18± 0.13	10.22
Dropout	11.52± 0.18	9.83
DropConnect	11.10± 0.13	**9.41**

Table 5. CIFAR-10 classification error using a larger feature extractor. Previous state-of-the-art is 9.5% (Snoek et al., 2012). Voting with 12 DropConnect networks produces an error rate of **9.32**%, significantly beating the state-of-the-art.

- 與DropConnect實驗相比較

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	93.28 ± 0.73	86.23 ± 0.11	0.191 ± 0.019	0.479 ± 0.018
DropOut	92.55 ± 0.24	86.42 ± 0.19	0.208 ± 0.015	0.493 ± 0.032
DropConnect	92.79 ± 0.31	86.43 ± 0.18	0.218 ± 0.011	0.474 ± 0.012
GradDrop	92.63 ± 0.12	86.41 ± 0.21	0.221 ± 0.006	0.455 ± 0.014

- 正在進行的實驗:

  1. **SVHN(train: 604388, test: 26032)**
     ↑ 上一篇論文的架構的**CNN**
  2. 上一篇**cifar10**的架構還在想辦法提高**acc**
  3. **GradDrop**的參數設定

- 預定再加入的實驗:

  – 除了找一些相關論文的實驗來比較，再加入一些比較近期的模型來比較

- 目前有在用的模型與資料集:

  1. **cifar10, mnist, cifar100, SVHN, NORM**
  2. **AlexNet, VGG16, VGG19, other CNN Models**
  3. **RNN-base Models**

- 遇到問題:

  1. 比較早期的**Paper**對模型的參數層數講的比較模糊，所以可能**acc**常常會沒辦法跟論文一樣好
  2. **AlexNet, VGG16, VGG19, other CNN Models**
  3. **RNN-base Models**

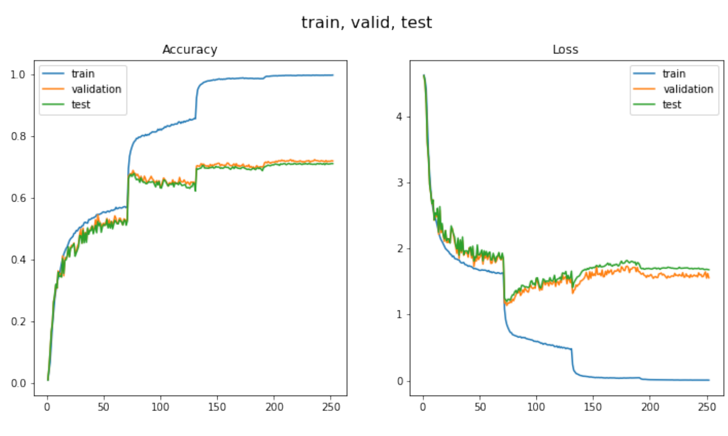
03/09

- **VGG16 in cifar100**

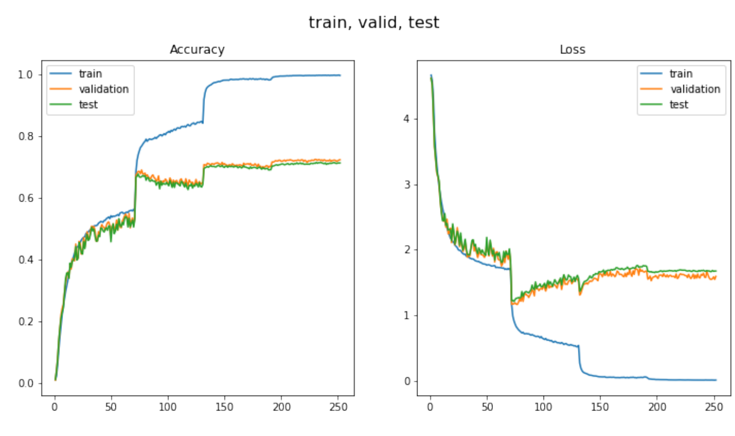
- **GradDrop's hyper parameters setting is :**
  I_P=0.45, W_P=0.15, GD_P=0.15, w_small=True, gd_small=True

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	99.74	71.24	0.010	1.698
DropOut(p=0.5)	99.72	71.50	0.011	1.667
DropConnect(p=0.5)	99.77	71.26	0.009	1.670
GradDrop	99.83	71.42	0.008	1.642

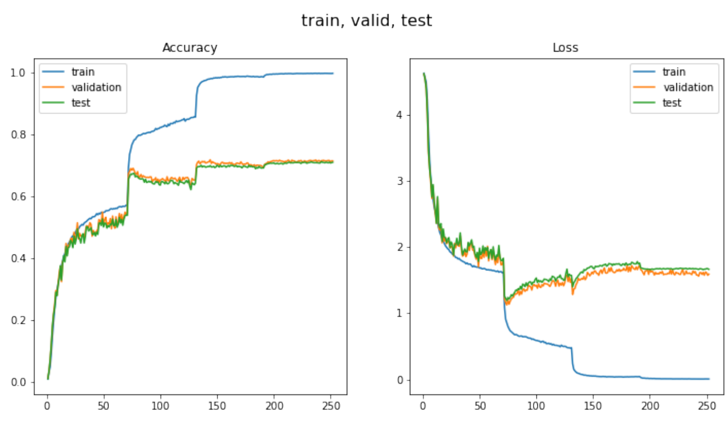
- **Training 狀況:**



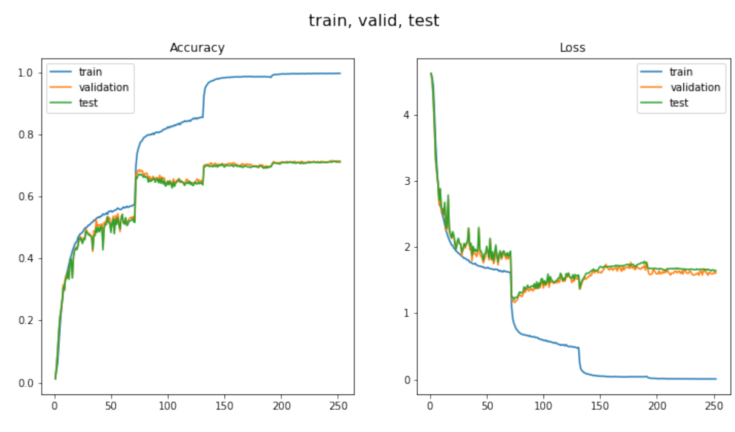
(a) Vallina



(b) DropOut



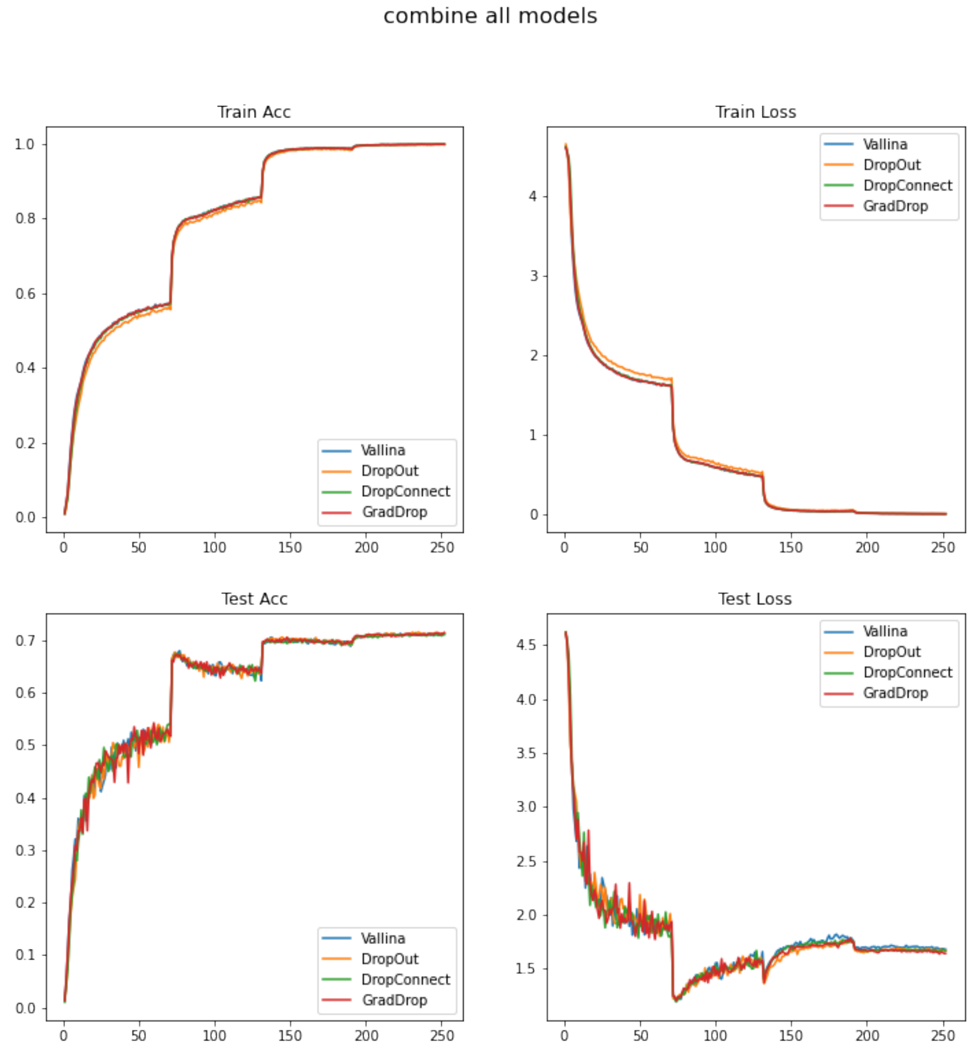
(c) DropConnect



(d) GradDrop

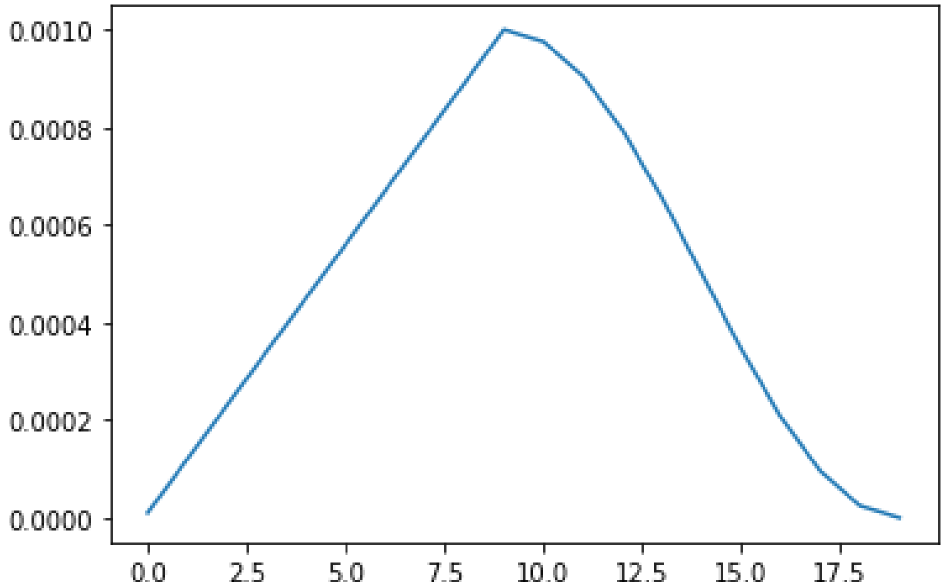
Figure 1: each model train and loss

- **擺在一起看:**



- **在lr scheduler中加入warm up 效果還不錯:**

† 以CosineAnnealingLR scheduler為例



0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

- **下週預計TODO:**

  † 彙整一下各個實驗的實驗結果:

  † 可用的實驗:

    1. **Very SimpleCnn → mnist**
    2. **SimpleCnn follow paper → cifar10**
    3. **AlexNet → cifar10**
    4. **VGG → cifar10, mnist, cifar100, SVHN, NORM**

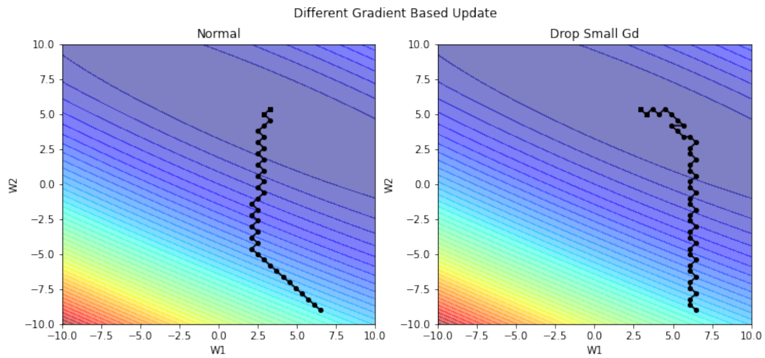
  † 尚未完成的實驗:

    1. **SimpleCnn → SVHN, NORM**
    2. **AlexNet → cifar100, SVHN, NORM**
    3. **VGG → cifar100, SVHN, NORM**
    4. **GradDrop Model的超參數設置**

04/20

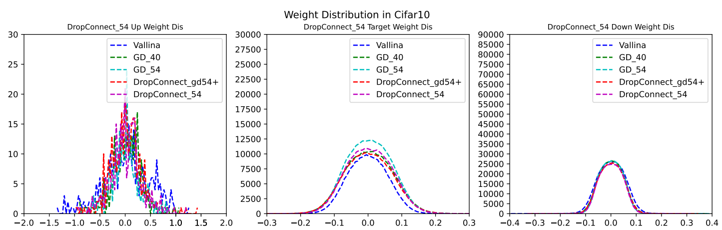
1. Gradient DropConnect 特性

✈ **Gradient 特性:**



Drop 掉gd較小的確實看起來比較容易走short cut

✈ **Some Observation on Dropout-like Method:**



Dropout 後，apply層weight分佈會變得比較集中

✈ **正在進行partial noise label的比較實驗:**

2. **Mnist Dataset**

✈ **Dataset INFO:**

10 classes with
train data: 60,000 images(28x28 pixels)
test data: 10,000 images(28x28 pixels)

✈ **SimpleCnn1(change first channel):**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	99.30 ± 0.01	99.26 ± 0.04	0.023 ± 0.000	0.025 ± 0.001
DropOut$(p = 0.5)$	98.36 ± 0.09	99.19 ± 0.08	0.059 ± 0.003	0.026 ± 0.001
DropConnect$(p = 0.5)$	98.92 ± 0.03	99.29 ± 0.03	0.036 ± 0.001	0.026 ± 0.001
GradDrop $(p = 0.45, w = 0.05, g = 0.2)$	99.02 ± 0.06	99.30 ± 0.01	0.033 ± 0.002	0.024 ± 0.001

3. Cifar10 Dataset

- **Dataset INFO:**

  10 classes with
  train data: 50,000 images(3x32x32 pixels)
  test data: 10,000 images(3x32x32 pixels)

- **SimpleCnn1(follow paper):**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	88.47 ± 1.74	77.25 ± 0.34	0.331 ± 0.050	0.764 ± 0.049
DropOut$(p = 0.5)$	83.88 ± 0.38	78.90 ± 0.21	0.455 ± 0.009	0.662 ± 0.013
DropConnect$(p = 0.5)$	87.49 ± 0.90	78.94 ± 0.29	0.352 ± 0.026	0.678 ± 0.012
GradDrop $(p = 0.3, w = 0.05, g = 0.35)$	87.64 ± 0.84	80.01 ± 0.16	0.349 ± 0.025	0.601 ± 0.006

- **SimpleCnn2(follow paper):Need to be Better**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	93.18 ± 0.73	85.53 ± 0.21	0.191 ± 0.019	0.579 ± 0.018
DropOut$(p = 0.5)$	91.95 ± 0.54	85.39 ± 0.21	0.238 ± 0.015	0.593 ± 0.032
DropConnect$(p = 0.5)$	92.69 ± 0.41	85.59 ± 0.20	0.208 ± 0.011	0.574 ± 0.012
GradDrop $(p = 0.3, w = 0.05, g = 0.35)$	92.63 ± 0.22	85.25 ± 0.18	0.211 ± 0.006	0.555 ± 0.014

- **AlexNet:**

Model	Test Acc	Train Acc
Vallina	81.27 ± 0.24	99.95 ± 0.00
DropOut$(p = 0.5)$	83.32 ± 0.21	99.25 ± 0.22
DropConnect$(p = 0.5)$	82.89 ± 0.34	99.31 ± 0.17
GradDrop $(p = 0.3, w = 0.05, g = 0.35)$	83.36 ± 0.04	98.9 ± 0.08

- **VGG:**

Model	Train Acc	Test Acc	Train Loss	Test Loss
vallina	99.96 ± 0.00	94.08 ± 0.04	0.001 ± 0.000	0.288 ± 0.003
DropConnect$(p = 0.2)$	99.97 ± 0.00	94.09 ± 0.04	0.001 ± 0.000	0.289 ± 0.004
GradDrop $(p = 0.1, w = 0.25, g = 0.25)$	99.98 ± 0.01	94.17 ± 0.01	0.001 ± 0.000	0.235 ± 0.000

4. **cifar100 Dataset**

- ✈ **Dataset INFO:**

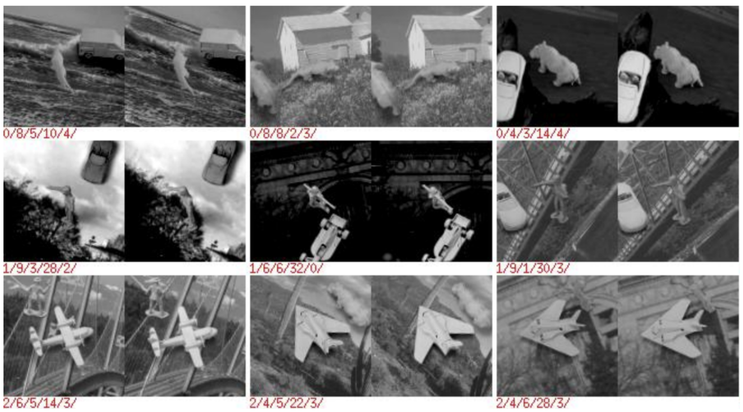
  100 classes with
  train data: 50,000 images(3x32x32 pixels)
  test data: 10,000 images(3x32x32 pixels)

- ✈ **VGG:**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	99.89	71.93	0.011	1.201
DropOut($p = 0.5$)	99.77	72.55	0.014	1.222
DropConnect($p = 0.5$)	99.88	71.96	0.011	1.208
GradDrop ($p = 0.45, w = 0.15, g = 0.15$)	99.88	72.29	0.011	1.167

5. NORB Dataset

- **Dataset INFO: NEW**
  - 6 classes with
    (0 for animal, 1 for human, 2 for plane, 3 for truck, 4 for car, 5 for blank)
    train data: 58,320 images(2x108x108 pixels, only use two folds)
    test data: 58,320 images(2x108x108 pixels)
  - They are "category / instance / elevation / azimuth / lighting"
    - 1. the instance in the category (0 to 9) - 2. the elevation (0 to 8, which mean cameras are 30, 35,40,45,50,55,60,65,70 degrees from the horizontal respectively) - 3. the azimuth (0,2,4,...,34, multiply by 10 to get the azimuth in degrees) - 4. the lighting condition (0 to 5)



- **SimpleCnn2(follow paper):**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	98.90 ± 0.14	94.25 ± 0.05	0.049 ± 0.003	0.168 ± 0.003
DropOut$(p = 0.5)$	98.67 ± 0.11	94.25 ± 0.27	0.051 ± 0.003	0.164 ± 0.006
DropConnect$(p = 0.5)$	98.69 ± 0.19	94.18 ± 0.11	0.050 ± 0.005	0.166 ± 0.002
GradDrop $(p = 0.4, w = 0.05, g = 0.2)$	98.90 ± 0.35	94.61 ± 0.06	0.045 ± 0.009	0.172 ± 0.001

- **AlexNet:**

此dataset 可能比較簡單，因此較複雜的model效果不一定比較好

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	98.64 ± 0.16	94.55 ± 0.12	0.022 ± 0.004	0.163 ± 0.005
DropOut$(p = 0.5)$	99.36 ± 0.27	94.49 ± 0.22	0.028 ± 0.007	0.164 ± 0.008
DropConnect$(p = 0.5)$	99.41 ± 0.46	94.51 ± 0.17	0.026 ± 0.011	0.160 ± 0.005
GradDrop$(p = 0.4, w = 0.05, g = 0.2)$	99.63 ± 0.20	94.75 ± 0.20	0.020 ± 0.005	0.163 ± 0.005

- **VGG:**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina(29.9M)	99.94 ± 0.06	95.42 ± 0.12	0.008 ± 0.002	0.149 ± 0.007
DropOut(29.9M, $p = 0.5$)	99.96 ± 0.05	95.62 ± 0.17	0.008 ± 0.001	0.142 ± 0.005
DropConnect(29.9M, $p = 0.5$)	99.99 ± 0.01	95.39 ± 0.19	0.007 ± 0.000	0.150 ± 0.006
GradDrop(29.9M, $p = 0.4, w = 0.05, g = 0.2$)	99.95 ± 0.03	95.76 ± 0.20	0.000 ± 0.001	0.317 ± 0.005

6. **SVHN Dataset**

- **Dataset INFO:**

  10 classes with
  train data: 604,388 images(both train set and extra set, 3x32x32 pixels)
  test data: 26,032 images(3x32x32 pixels)

- **SimpleCnn2(follow paper): NEW**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina	96.20 ± 0.52	93.39 ± 0.10	0.120 ± 0.016	0.259 ± 0.003
DropOut($p = 0.5$)	96.11 ± 0.29	93.50 ± 0.07	0.126 ± 0.009	0.253 ± 0.004
DropConnect($p = 0.5$)	96.61 ± 0.05	93.36 ± 0.07	0.110 ± 0.001	0.265 ± 0.004
GradDrop($p = 0.4, w = 0.05, g = 0.2$)	96.33 ± 0.25	93.47 ± 0.12	0.118 ± 0.008	0.249 ± 0.005

- **AlexNet:**
- **VGG: NEW**

Model	Train Acc	Test Acc	Train Loss	Test Loss
Vallina(29.9M)	97.01 ± 0.41	93.55 ± 0.08	0.094 ± 0.014	0.252 ± 0.006
DropOut(29.9M, $p = 0.5$)	96.90 ± 0.16	93.73 ± 0.05	0.099 ± 0.006	0.246 ± 0.004
DropConnect(29.9M, $p = 0.5$)	96.97 ± 0.06	93.63 ± 0.08	0.096 ± 0.003	0.262 ± 0.005
GradDrop(29.9M, $p = 0.4, w = 0.05, g = 0.2$)	97.60 ± 0.13	93.67 ± 0.05	0.073 ± 0.004	0.230 ± 0.003

7. Compare DropConnect and GradDrop

- **Dataset Norb:**

  10 classes with
  train data: 604,388 images(both train set and extra set, 3x32x32 pixels)
  test data: 26,032 images(3x32x32 pixels)

- **SimpleCnn2 in different drop ratio:**

GradDrop Drop Rate $p \approx p + (w + g) \times (\frac{2}{3})$

Model	Train Acc	Test Acc	Train Loss	Test Loss
DropConnect$(p = 0.4)$	98.90 ± 0.17	94.06 ± 0.28	0.046 ± 0.004	0.172 ± 0.008
DropConnect$(p = 0.6)$	98.82 ± 0.17	94.36 ± 0.00	0.048 ± 0.004	0.163 ± 0.002
GradDrop $(p = 0.4, w = 0.0, g = 0.0)$	99.11 ± 0.16	94.17 ± 0.14	0.042 ± 0.003	0.167 ± 0.005
GradDrop $(p = 0.4, w = 0.05, g = 0.25)$	98.92 ± 0.10	94.61 ± 0.01	0.044 ± 0.002	0.172 ± 0.003

- **VGG in different drop ratio: NEW**

GradDrop Drop Rate $p \approx p + (w + g) \times (\frac{2}{3})$

Model	Train Acc	Test Acc	Train Loss	Test Loss
DropConnect$(p = 0.4)$	100.00 ± 0.00	97.22 ± 0.11	0.005 ± 0.000	0.084 ± 0.003
DropConnect$(p = 0.6)$	100.00 ± 0.00	97.05 ± 0.18	0.005 ± 0.000	0.089 ± 0.005
GradDrop $(p = 0.4, w = 0.0, g = 0.0)$	100.00 ± 0.00	97.02 ± 0.15	0.005 ± 0.000	0.090 ± 0.005
GradDrop $(p = 0.4, w = 0.05, g = 0.25)$	99.99 ± 0.00	97.34 ± 0.09	0.005 ± 0.000	0.217 ± 0.003

8. 超參數設置實驗 OLD

✈ **Cut Gradient Small or Big(0.4, 0.05, 0.2)**

Model	Best Test Acc	Best Train Acc
AlexNet(Vallina)	81.27 ± 0.24	99.953 ± 0.005
AlexNet(Dropout)	83.32 ± 0.21	99.25 ± 0.22
AlexNet(DropConnect)	82.89 ± 0.34	99.31 ± 0.17
AlexNet(GaussianDrop)	82.94 ± 0.44	99.50 ± 0.03
AlexNet(w大,gd大)	82.61 ± 0.09	99.69 ± 0.05
AlexNet(w小,gd大)	82.15 ± 0.2	99.58 ± 0.14
AlexNet(w大,gd小)	83.6 ± 0.19	98.772 ± 0.11
AlexNet(w小,gd小)	83.36 ± 0.04	98.9 ± 0.08

✈ **Cut Gradient Small or Big(0.4, 0.2, 0.05)**

Model	Best Test Acc	Best Train Acc
AlexNet(Vallina)	81.27 ± 0.24	99.953 ± 0.005
AlexNet(Dropout)	83.32 ± 0.21	99.25 ± 0.22
AlexNet(DropConnect)	82.89 ± 0.34	99.31 ± 0.17
AlexNet(GaussianDrop)	82.94 ± 0.44	99.50 ± 0.03
AlexNet(w大,gd大)	82.47 ± 0.03	99.18 ± 0.09
AlexNet(w小,gd大)	82.52 ± 0.05	99.63 ± 0.12
AlexNet(w大,gd小)	83.03 ± 0.17	99.42 ± 0.05
AlexNet(w小,gd小)	82.48 ± 0.03	99.47 ± 0.02

✈ **hyperparameters values: NEW**

$p = 0.4, w = 0.2, p = 0$,
$p = 0.4, w = 0, p = 0.2$,
$p = 0.4, w = 0.2, p = 0.2$,
$p = 0, w = 0, p = 0.5$,
$p = 0, w = 0.5, p = 0$

✈ **Drop Rate 的極限可再加入GD提升Drop Rate?? NEW**