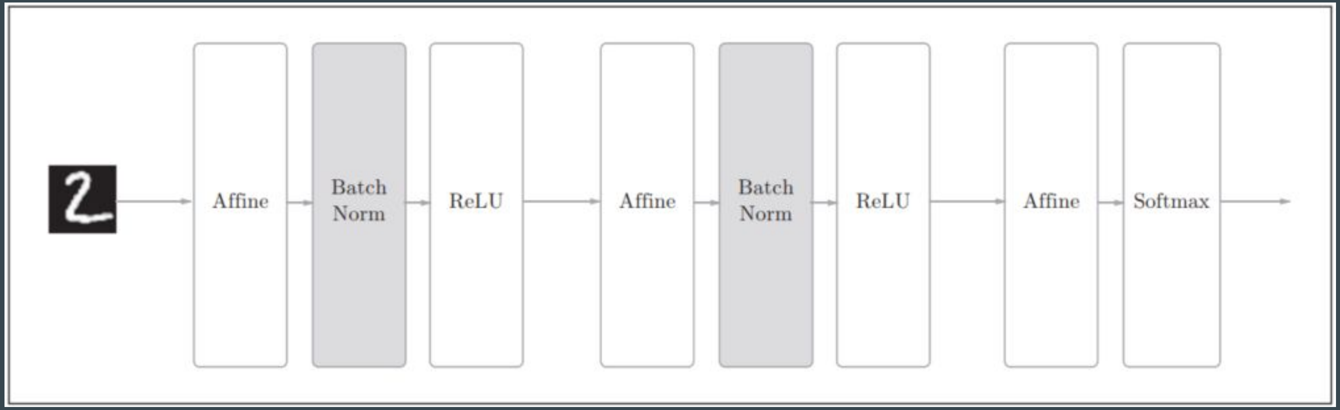
# Ch06. Training related skills

CK.D.Lee

# Batch Normalization

- 2015
- Batch Norm
- Why it's Good?
  - 可以使学习快速进行（可以增大学习率）。
  - 不那么依赖初始值（对于初始值不用那么神经质）。
  - 抑制过拟合（降低Dropout等的必要性）。



# Batch Normalization cont.

- Mini-batch based normalization
- 0 ~ 1
- $$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$
  $$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$
  $$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

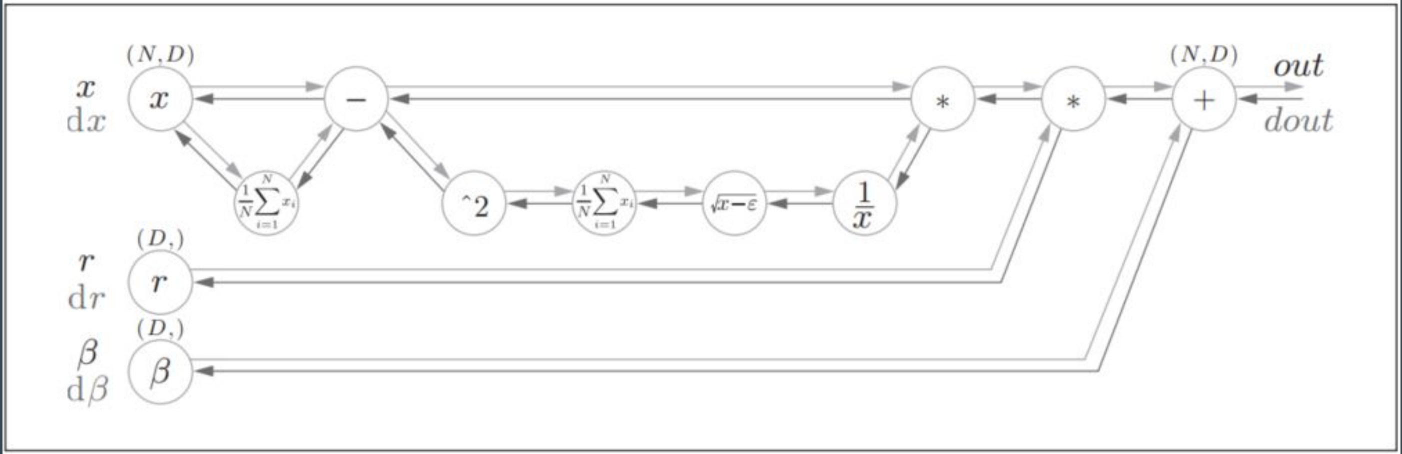
*ε是一个微小值（比如，10e-7等），它是为了防止出现除以0的情况

Original Normalization:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

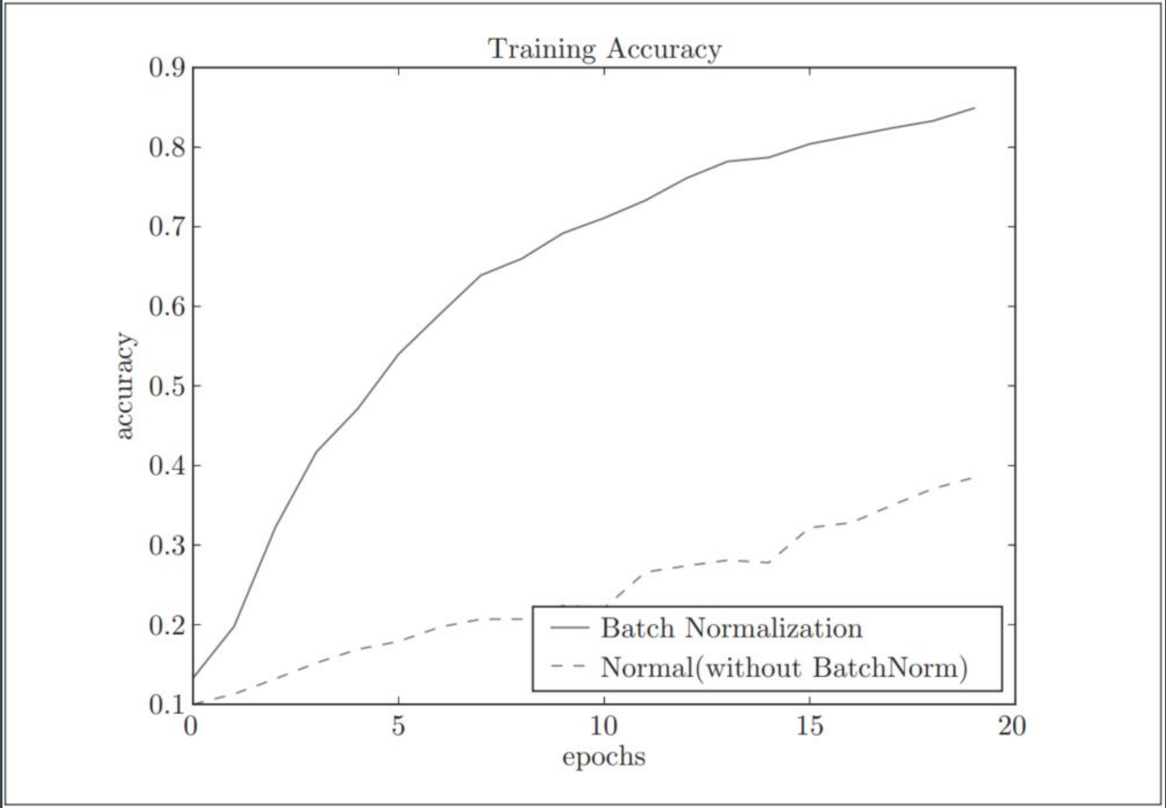
# Batch Norm

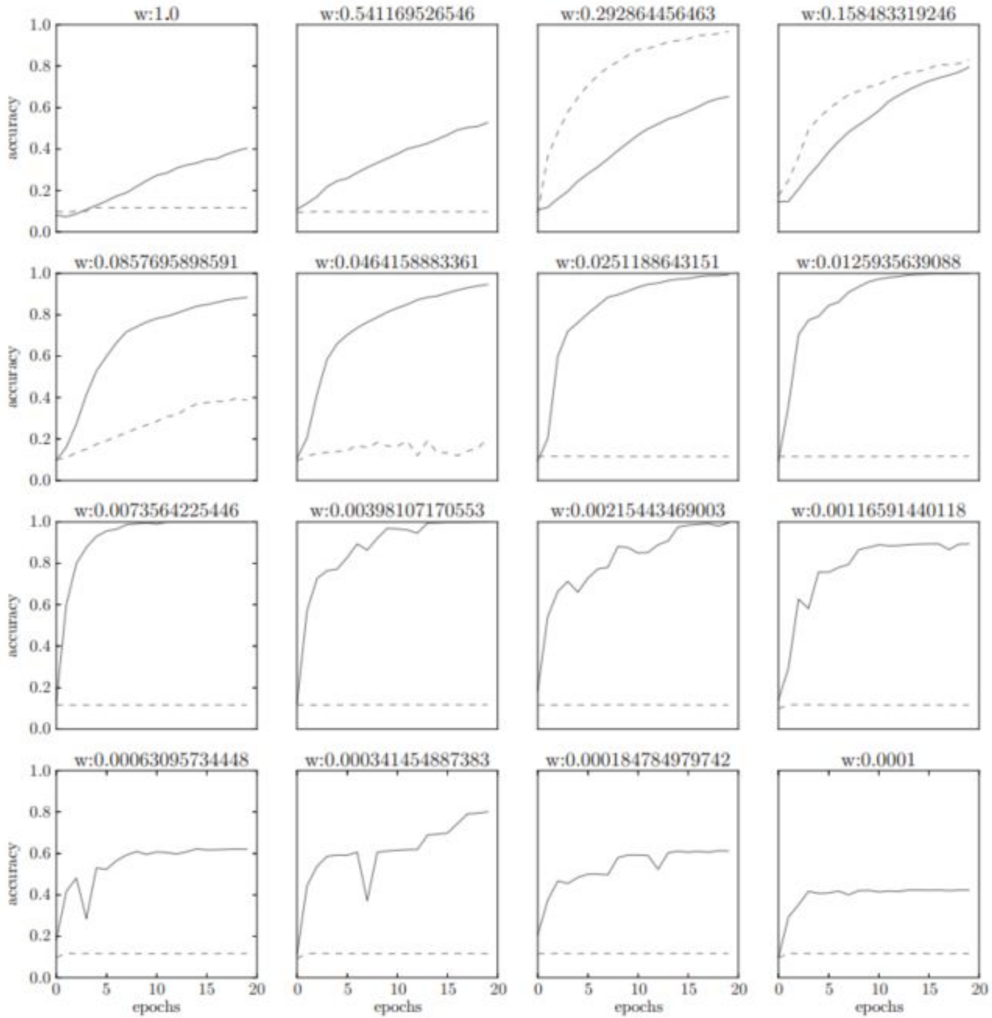
$$y_i \leftarrow \gamma \hat{x}_i + \beta$$




https://kratzert.github.io/2016/02/12/understanding-the-gradient-flow-through-the-batch-normalization-layer.html

# Training Accuracy with Batch Norm

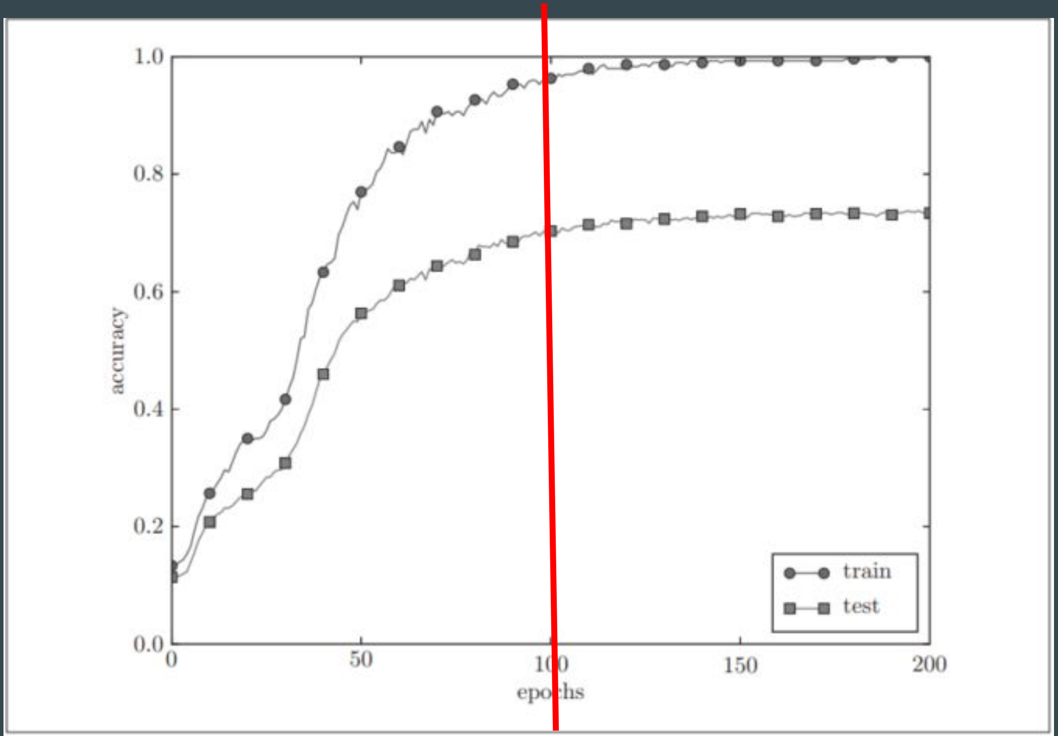



w:1.0
w:0.541169526546
w:0.292864456463
w:0.158483319246
w:0.0857695898591
w:0.0464158883361
w:0.0251188643151
w:0.0125935639088
w:0.00735642254 46
w:0.00398107170553
w:0.00215443469003
w:0.00116591440118
w:0.00063095734448
w:0.000341454887383
w:0.000184784979742
w:0.0001
accuracy
0.0
0.2
0.4
0.6
0.8
1.0
epochs
0
5
10
15
20

# Regularization - Overfitting

- Model has massive parameter
- Lack of training data
- Test
  - 300 training data
  - 7-layer
  - 100 neuron on each layer

# Overfitting

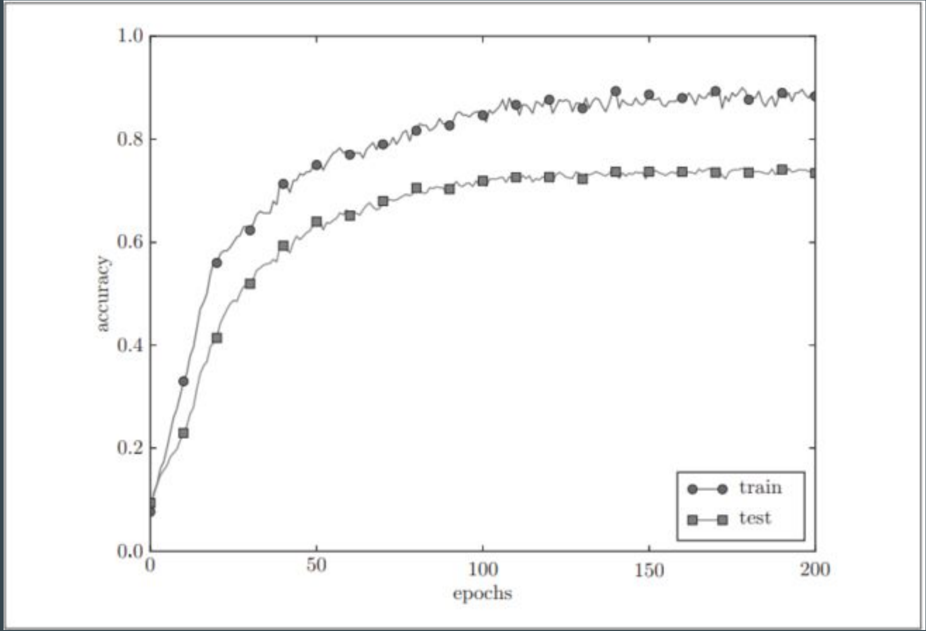


# Overfitting - Weigh Decay (权值衰减)

- L2 Regularization
- $$C = C_0 + \frac{\lambda}{2n} \sum_w w^2$$

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w$$
$$\frac{\partial C}{\partial b} = \frac{\partial C_0}{\partial b}.$$

$$\begin{aligned} w &\rightarrow w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} w \\ &= \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w}. \end{aligned}$$

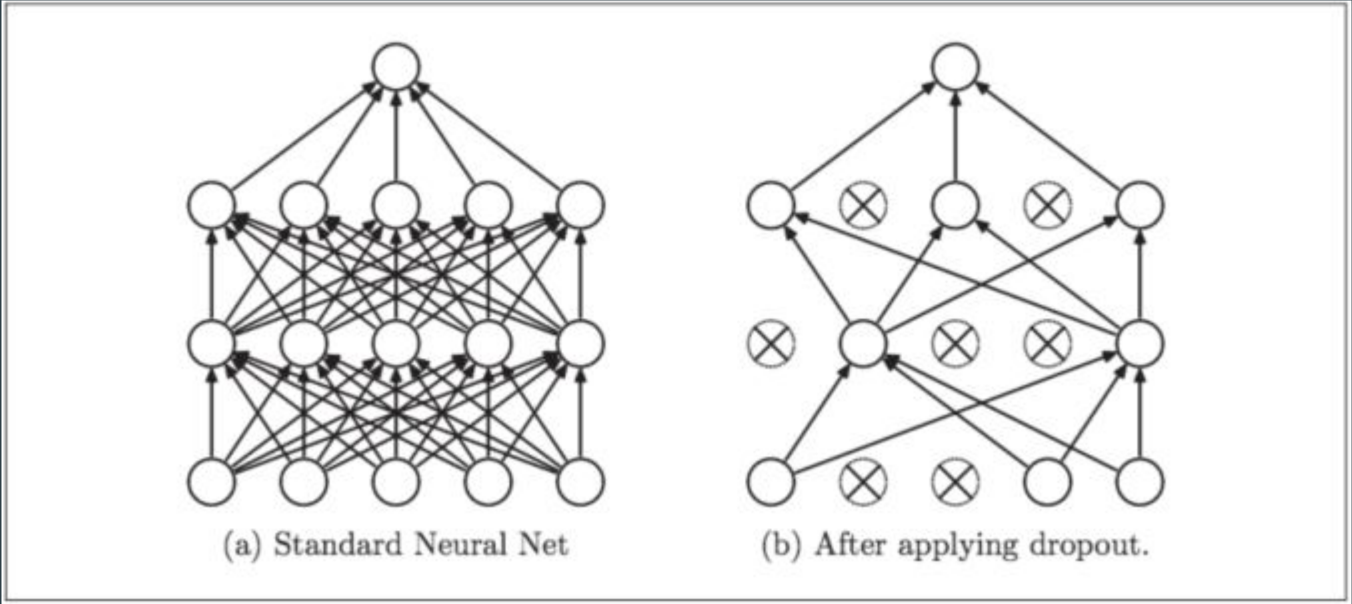



https://blog.csdn.net/program_developer/article/details/80867468

https://umbum.tistory.com/222

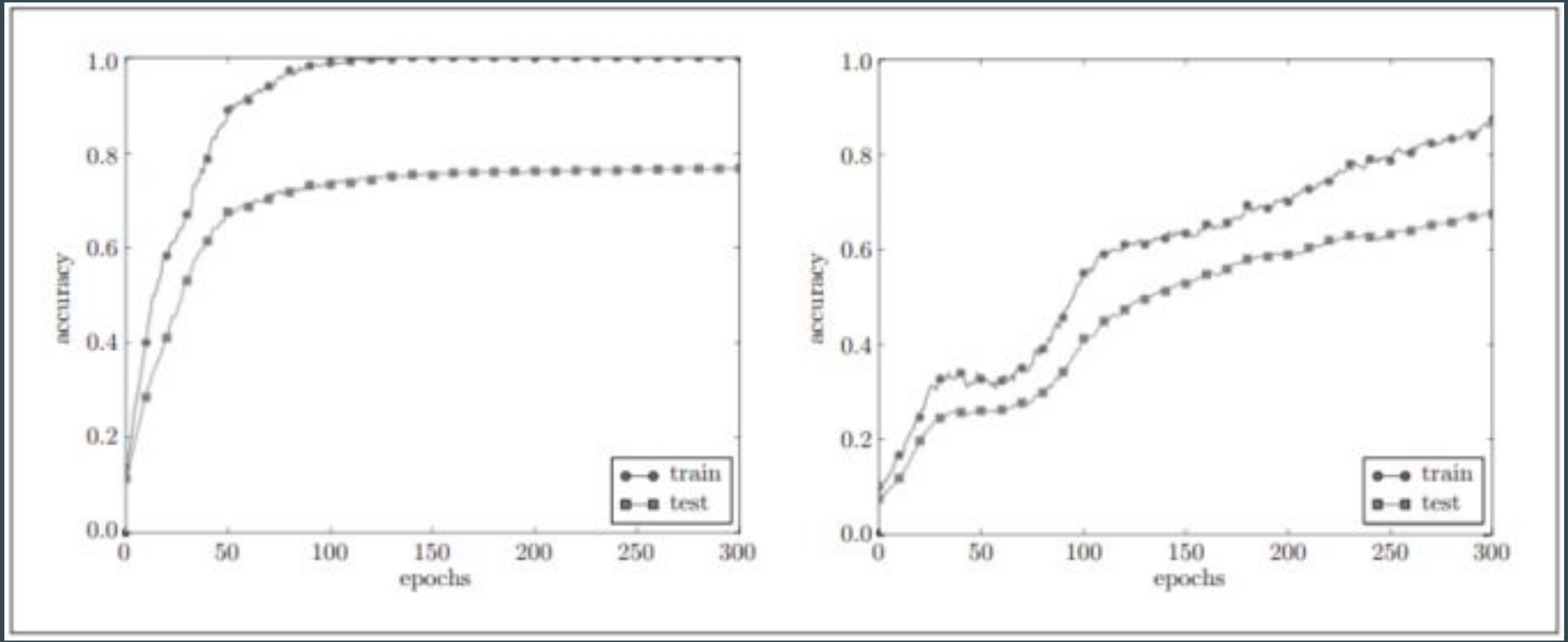
# Overfitting - Dropout

Delete neurons randomly



# Overfitting - Dropout cont.

Left - original, Right- uses Dropout technique



# Hyper-parameter

- 학습을 통해 튜닝, 최적화 하는 변수가 아닌 학습율이나 일반화 변수처럼 사람들이 선험적 지식으로 설정하거나 외부 모델 매커니즘을 통해 자동으로 설정되는 변수
  - Learning rate
  - Cost function
  - Regularization parameter
  - Mini-batch size
  - Number of Epochs
  - Hidden units
  - Weight initialization
  - etc.

https://m.blog.naver.com/PostView.nhn?blogId=laonple&logNo=220571820368&proxyReferer=https%3A%2F%2Fwww.google.com%2F

# Hyper-parameter Optimisation

Split data into training, test, validation data to avoid overfitting problem

Set hyper-parameters' range: e.g. 0.001 (10-3) ~ 1000 (103)

**步骤0**

设定超参数的范围。

**步骤1**

从设定的超参数范围中随机采样。

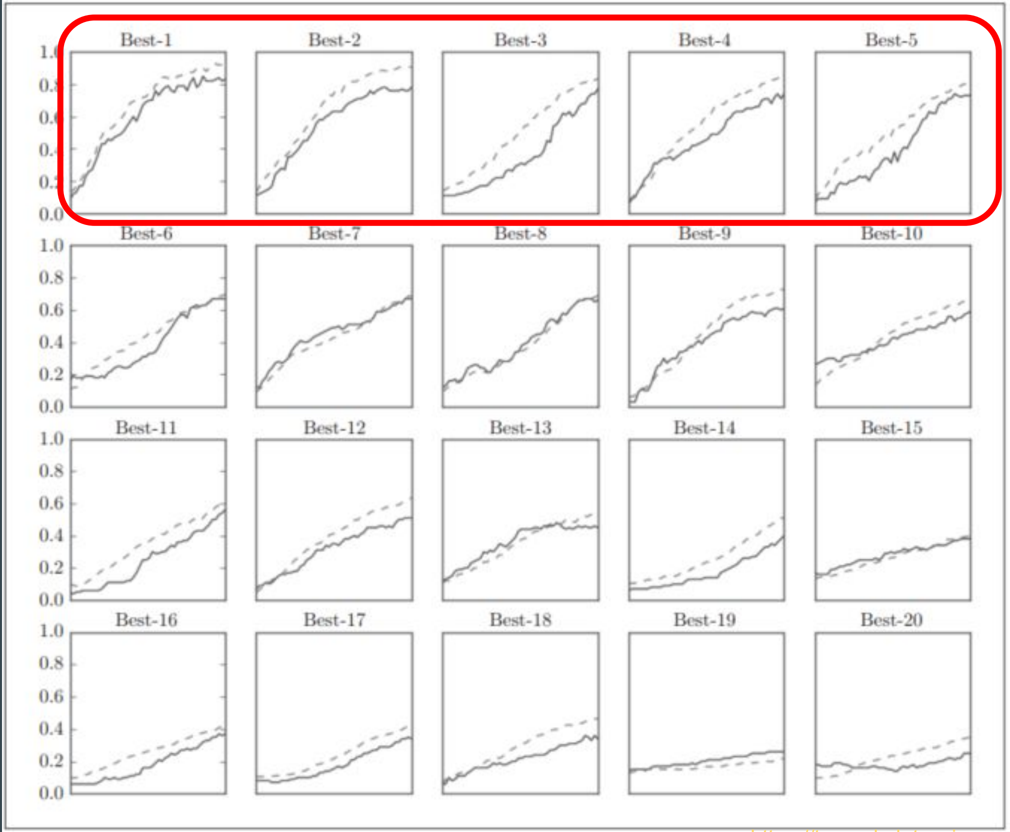
**步骤2**

使用步骤1中采样到的超参数的值进行学习，通过验证数据评估识别精度(但是要将epoch设置得很小)。

**步骤3**

重复步骤1和步骤2(100次等)，根据它们的识别精度的结果，缩小超参数的范围。

# Hyper-parameter Optimisation cont.




```
Best-1 (val acc:0.83) | lr:0.0092, weight decay:3.86e-07
Best-2 (val acc:0.78) | lr:0.00956, weight decay:6.04e-07
Best-3 (val acc:0.77) | lr:0.00571, weight decay:1.27e-06
Best-4 (val acc:0.74) | lr:0.00626, weight decay:1.43e-05
Best-5 (val acc:0.73) | lr:0.0052, weight decay:8.97e-06
```

从这个结果可以看出，学习率在0.001到0.01、权值衰减系数在$10^{-8}$到$10^{-6}$之间时，学习可以顺利进行。像这样，观察可以使学习顺利进行的超参数的范围，从而缩小值的范围。然后，在这个缩小的范围中重复相同的操作。

https://towardsdatascience.com/understanding-hyperparameters-and-its-optimisation-techniques-f0debba07568