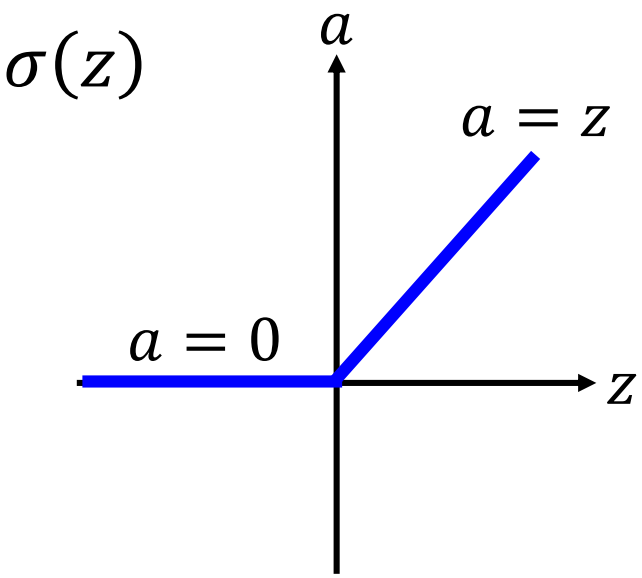
Activation Function: SELU

ReLU

- Rectified Linear Unit (ReLU)

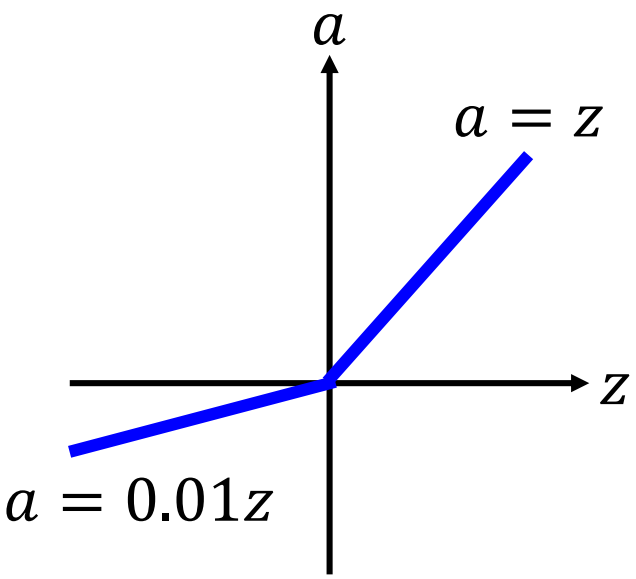


Reason:

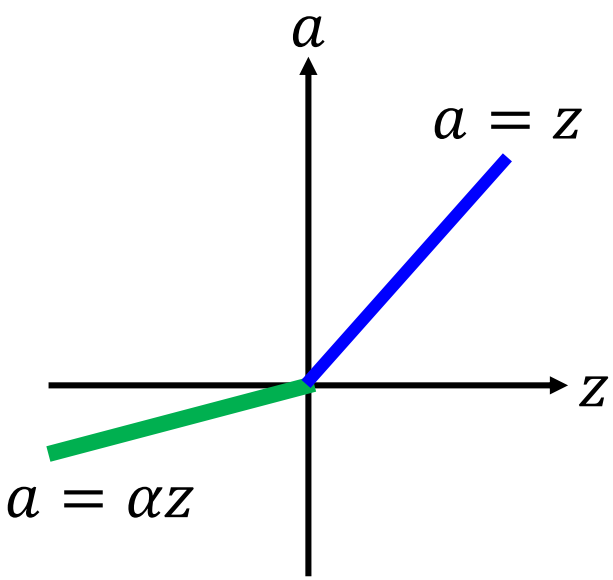
1. Fast to compute
2. Biological reason
3. Infinite sigmoid with different biases
4. Vanishing gradient problem

ReLU - variant

Leaky ReLU



Parametric ReLU



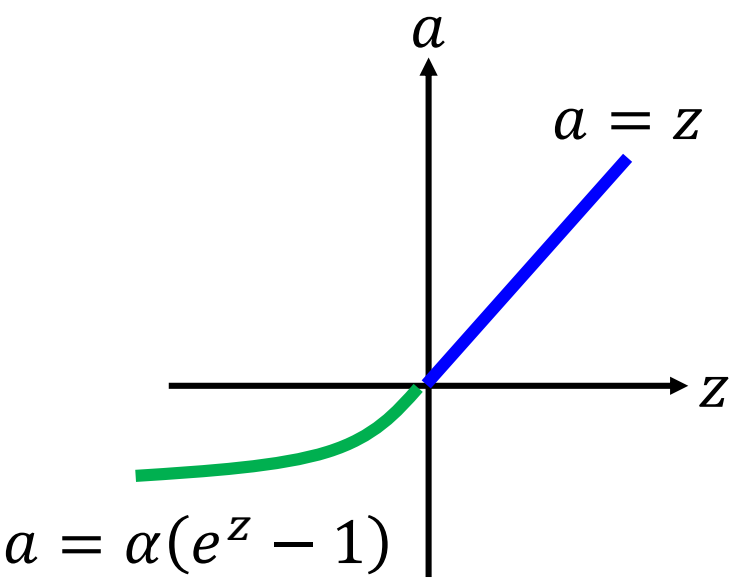
α also learned by gradient descent

(1) Definition of scaled exponential linear units (SELUs)

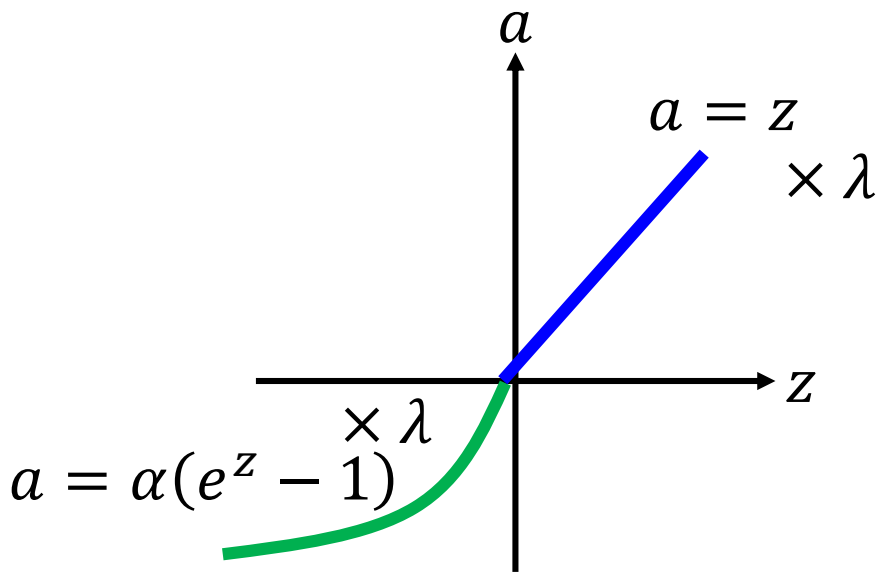
In [3]:
```
def selu(x):
    with ops.name_scope('elu') as scope:
        alpha = 1.6732632423543772848170429916717
        scale = 1.0507009873554804934193349852946
        return scale*tf.where(x>=0.0, x, alpha*tf.nn.elu(x))
```

https://github.com/bioinf-jku/SNNs

Exponential Linear Unit (ELU)



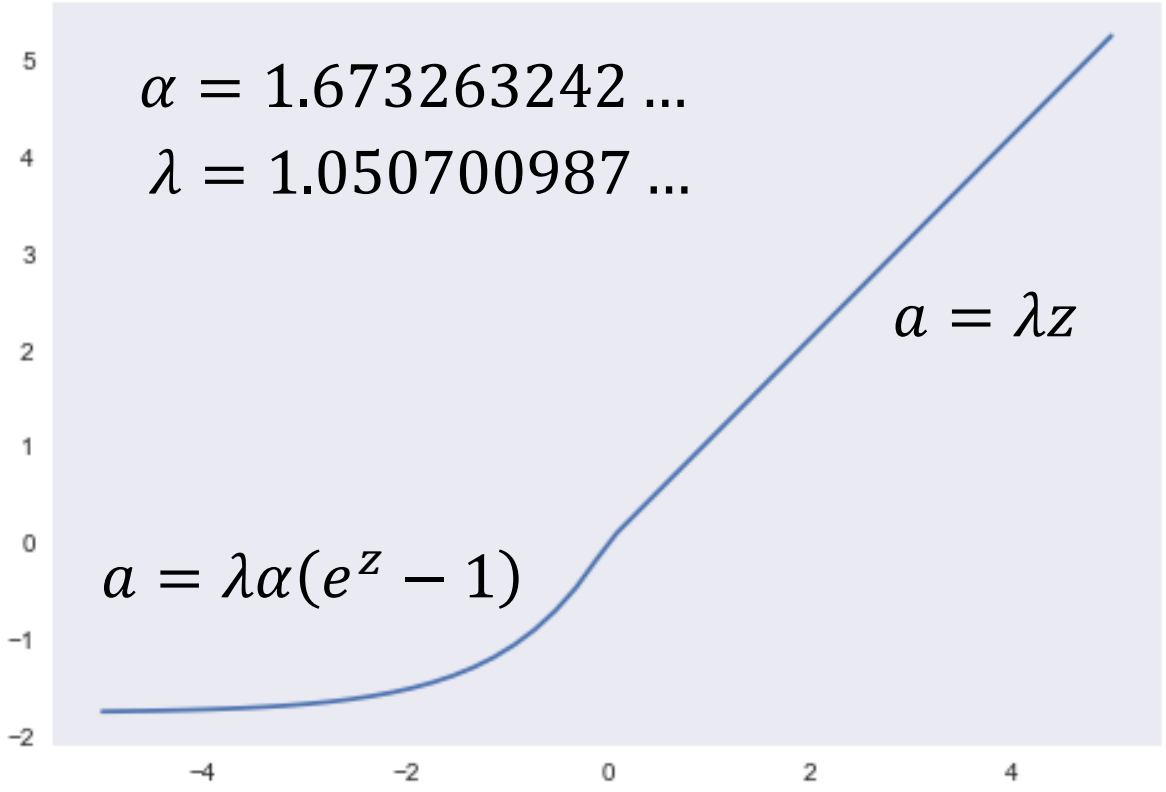
Scaled ELU (SELU)



$$\alpha = 1.6732632423543772848170429916717$$

$$\lambda = 1.0507009873554804934193349852946$$

SELU



Positive and negative values

➡ The whole ReLU family has this property except the original ReLU.

Saturation region ➡ ELU also has this property

Slope larger than 1 ➡ Only SELU also has this property

SELU

自帶normalization

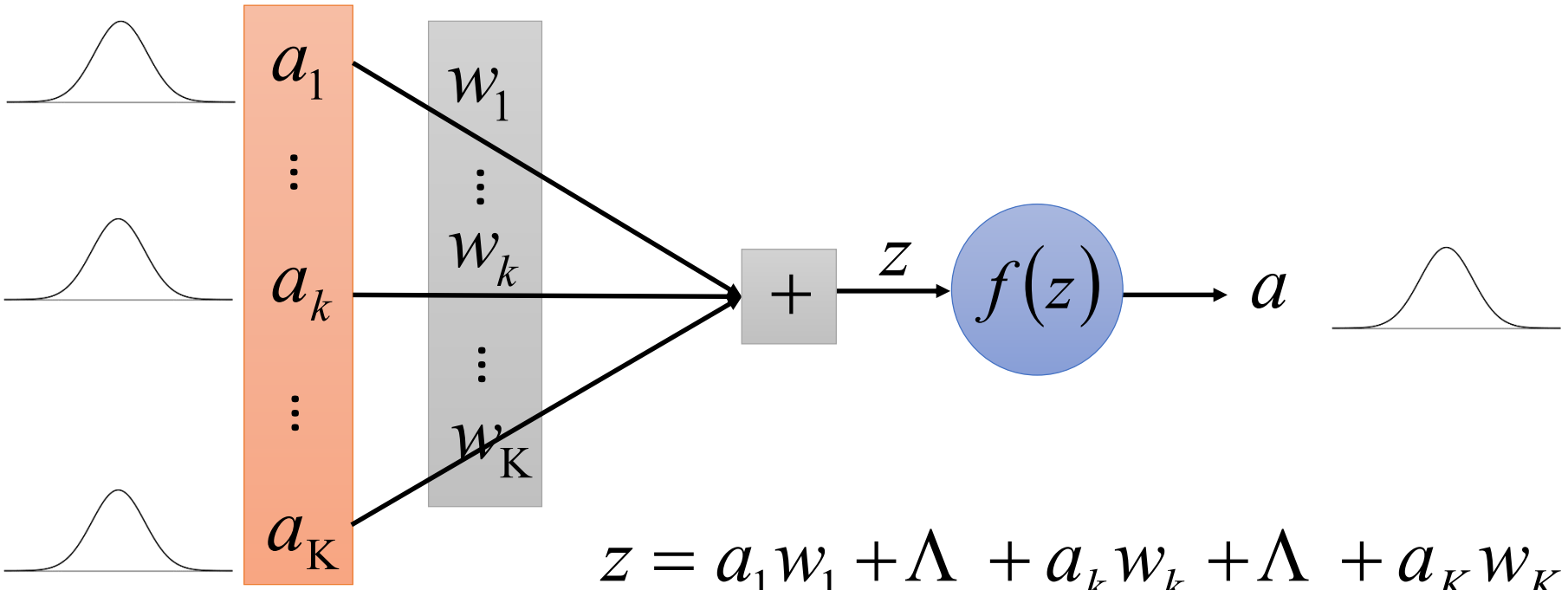
The inputs are i.i.d random variables with mean μ and variance σ^2. =1 =0

$$\mu_z = E[z]$$

$$= \sum_{k=1}^{K} E[a_k] w_k = \mu \sum_{k=1}^{K} w_k = \mu \cdot K\mu_w$$

=0 =0

希望最後output的a也是man=0, variance= 1



$$z = a_1 w_1 + \Lambda + a_k w_k + \Lambda + a_K w_K$$

Do not have to be Gaussian

keras在default時sigmaw平方就是1/k
如果不符合這個設定的話，selu就沒有用了！

SELU

希望他output的mean是0，output是1

The inputs are i.i.d random variables with mean μ and variance σ^2. =1

假設uw= 0, sigmaw平方=1/k

$$\mu_z = 0 \quad \mu_w = 0$$

$$\sigma_z^2 = E[(z-\mu_z)^2] = E[z^2]$$

$$= E[(a_1 w_1 + a_2 w_2 + \cdots)^2]$$

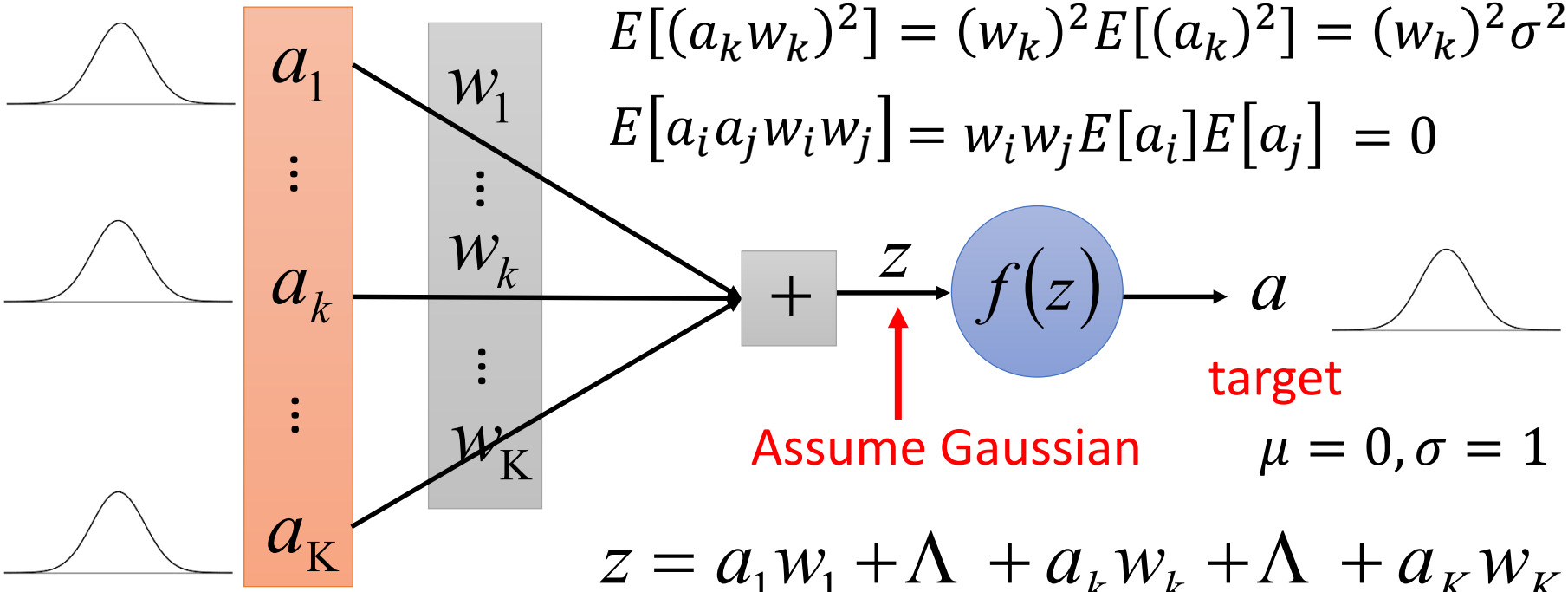
k是hidden layer的寬度

$$= \sum_{k=1}^{K} (w_k)^2 \sigma^2 = \sigma^2 \cdot K\sigma_w^2 = 1$$

=1 =1

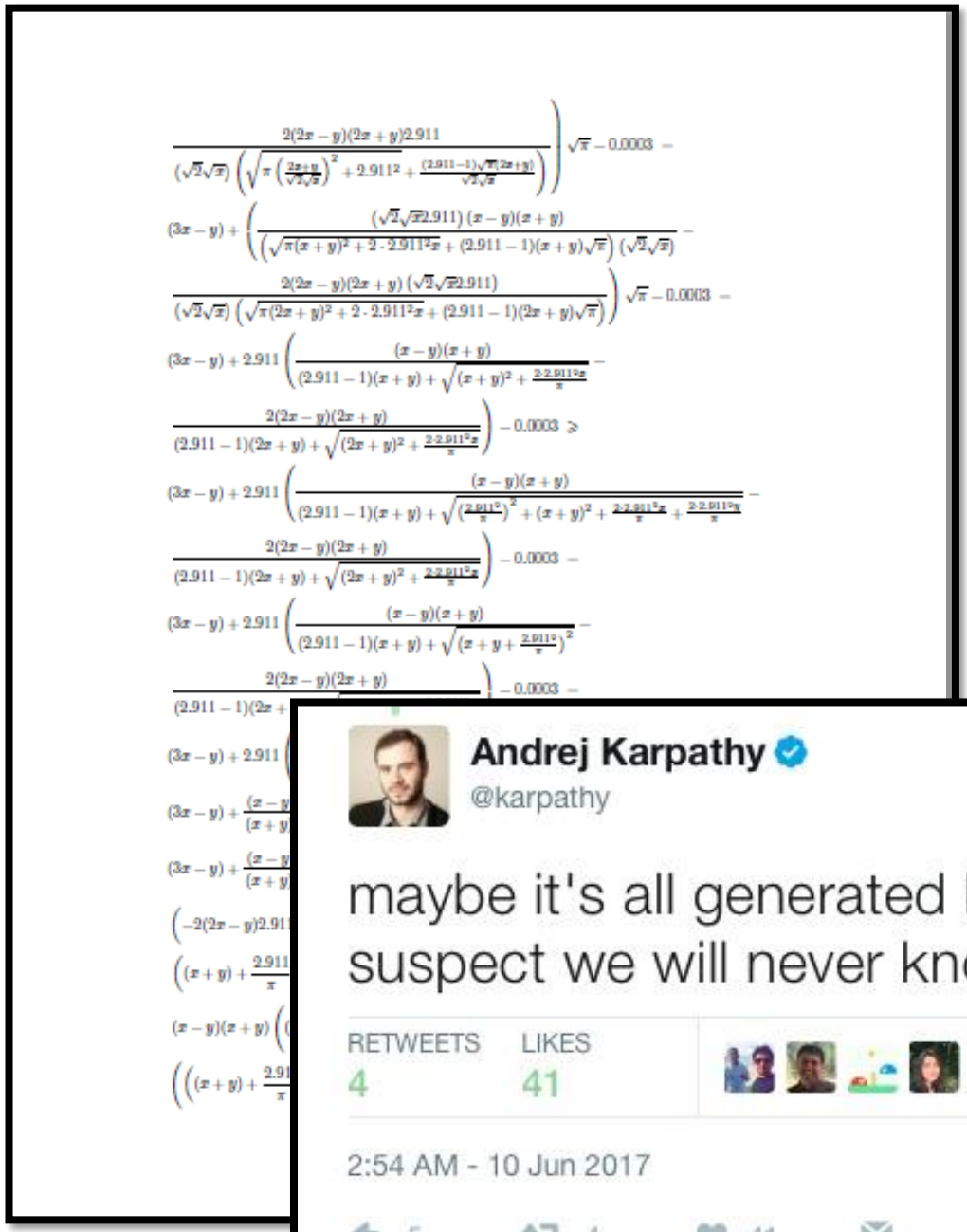
$$E[(a_k w_k)^2] = (w_k)^2 E[(a_k)^2] = (w_k)^2 \sigma^2$$

$$E[a_i a_j w_i w_j] = w_i w_j E[a_i] E[a_j] = 0$$



$$z = a_1 w_1 + \Lambda + a_k w_k + \Lambda + a_K w_K$$

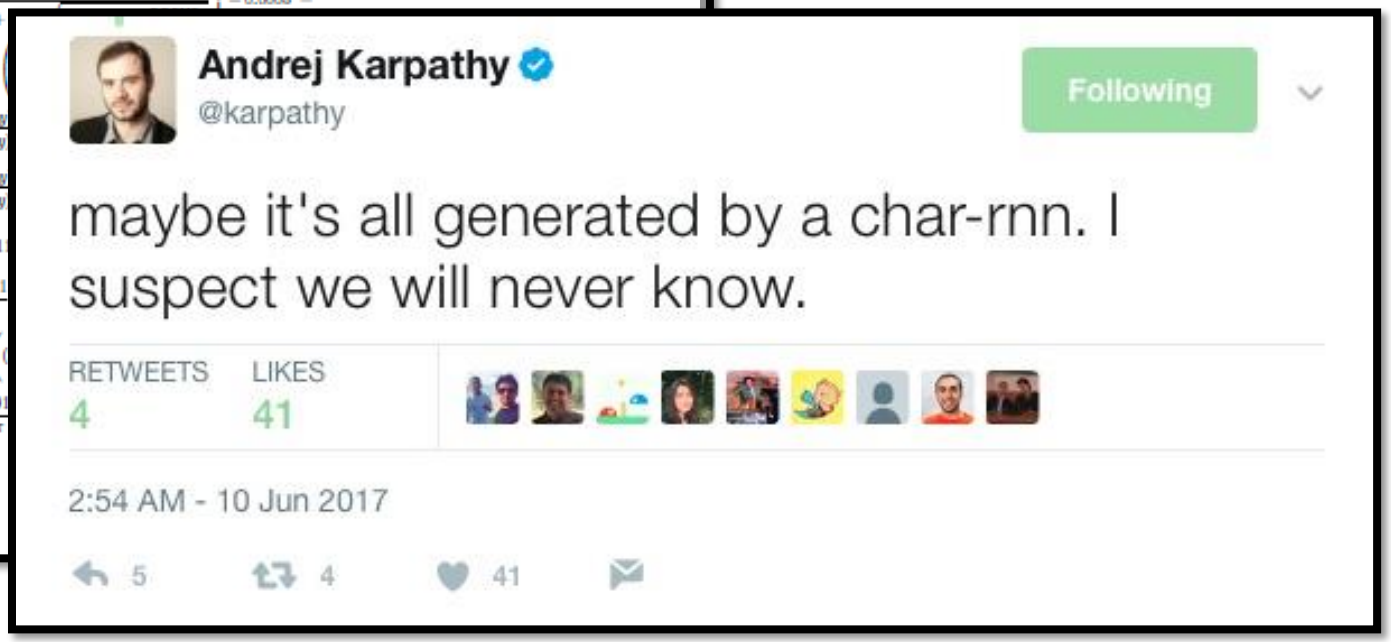
Demo



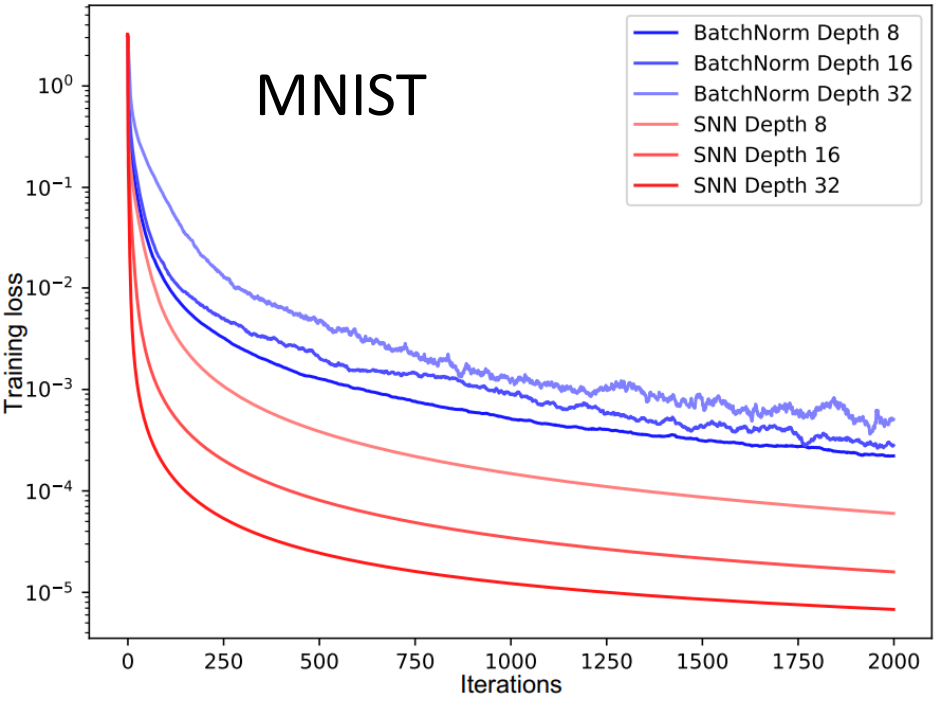
93 頁的證明

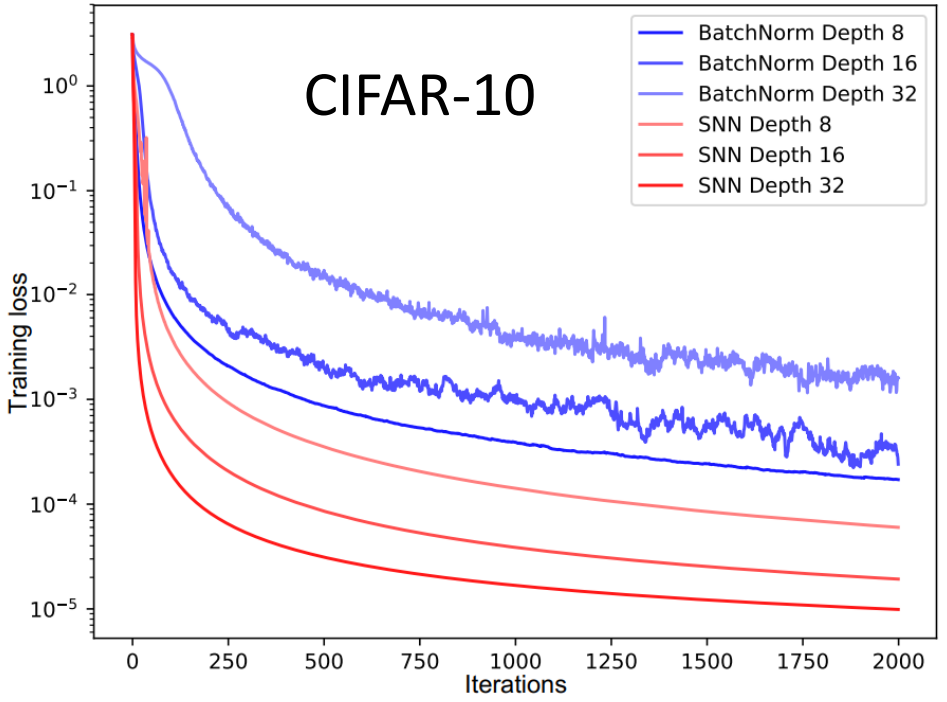
Source of joke: https://zhuanlan.zhihu.com/p/27336839

SELU is actually more general.



batch norm上下起伏會比較大，因為每次算出來的mean/variance會不一樣





在一般簡單的case如果用DNN不見得比傳統ML來得強，但是SELU最猛了

FNN method comparison

Method	avg. rank diff.	*p*-value
SNN	-0.756	
MSRAinit	-0.240*	2.7e-02
LayerNorm	-0.198*	1.5e-02
Highway	0.021*	1.9e-03
ResNet	0.273*	5.4e-04
WeightNorm	0.397*	7.8e-07
BatchNorm	0.504*	3.5e-06

ML method comparison

Method	avg. rank diff.	*p*-value
SNN	-6.7	
SVM	-6.4	5.8e-01
RandomForest	-5.9	2.1e-01
MSRAinit	-5.4*	4.5e-03
LayerNorm	-5.3	7.1e-02
Highway	-4.6*	1.7e-03
...	...	...

Demo