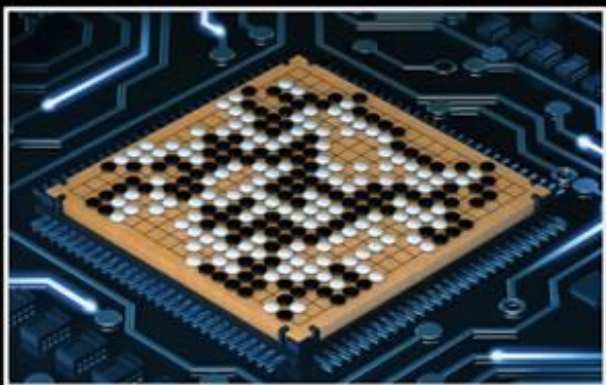


# Tuning Hyperparameters

automatically generate hyper parameters

Source of image: <https://medium.com/intuitionmachine/the-brute-force-method-of-deep-learning-innovation-58b497323ae5> (Denny Britz's graphic)

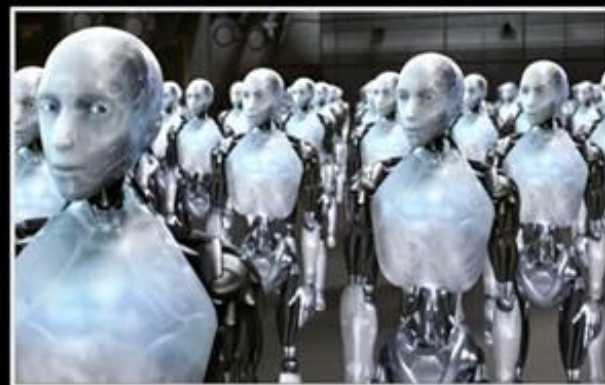
## Deep Learning研究生



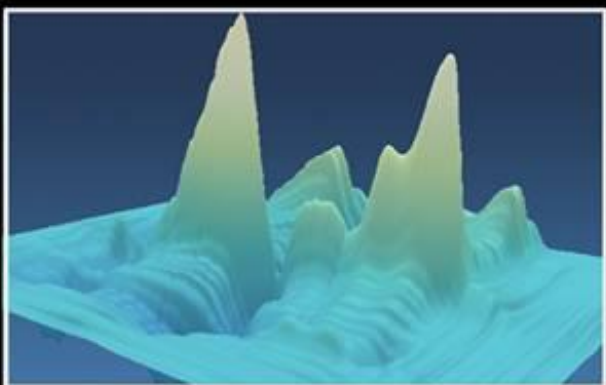
朋友覺得我在



我媽覺得我在



大眾覺得我在



指導教授覺得我在



我以為我在

HYPERPARAMETER Spreadsheet			
Hyperparameter	Value	Link	Result
Learning Rate	0.001	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	0.001
Batch Size	32	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	32
Number of Epochs	100	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	100
Optimizer	Adam	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	Adam
Dropout Rate	0.5	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	0.5
Weight Decay	0.0001	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	0.0001
Number of Layers	3	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	3
Number of Units per Layer	128	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	128
Activation Function	ReLU	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	ReLU
Loss Function	Cross Entropy	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	Cross Entropy
Gradient Descent	Stochastic	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	Stochastic
Early Stopping	True	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	True
Patience	10	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	10
Validation Frequency	10	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	10
Checkpointing	True	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	True
Model Path	/models/	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	/models/
Output Path	/outputs/	<a href="#">https://www.kaggle.com/dennybritz/deep-learning-innovation-58b497323ae5</a>	/outputs/

事實上我在

感謝 沈昇勳 同學提供圖檔

# Grid Search v.s. Random Search

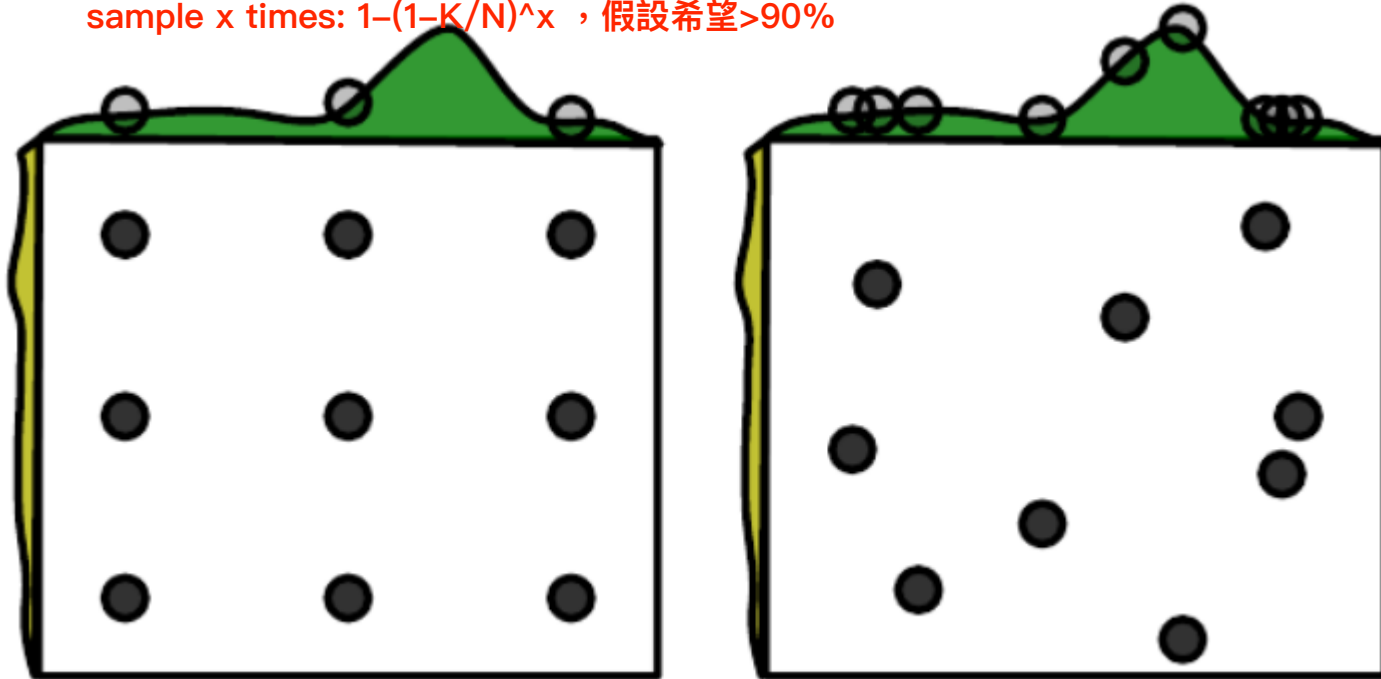
不要去掃過所有參數的組合，而是sample一些參數來做測試

assumption: top K results are good enough

sample到top K 的機率是 $K/N$

sample x times:  $1-(1-K/N)^x$ ，假設希望>90%

If  $N = 1000$ ,  $k = 10 \rightarrow x = 230$



Grid

Random

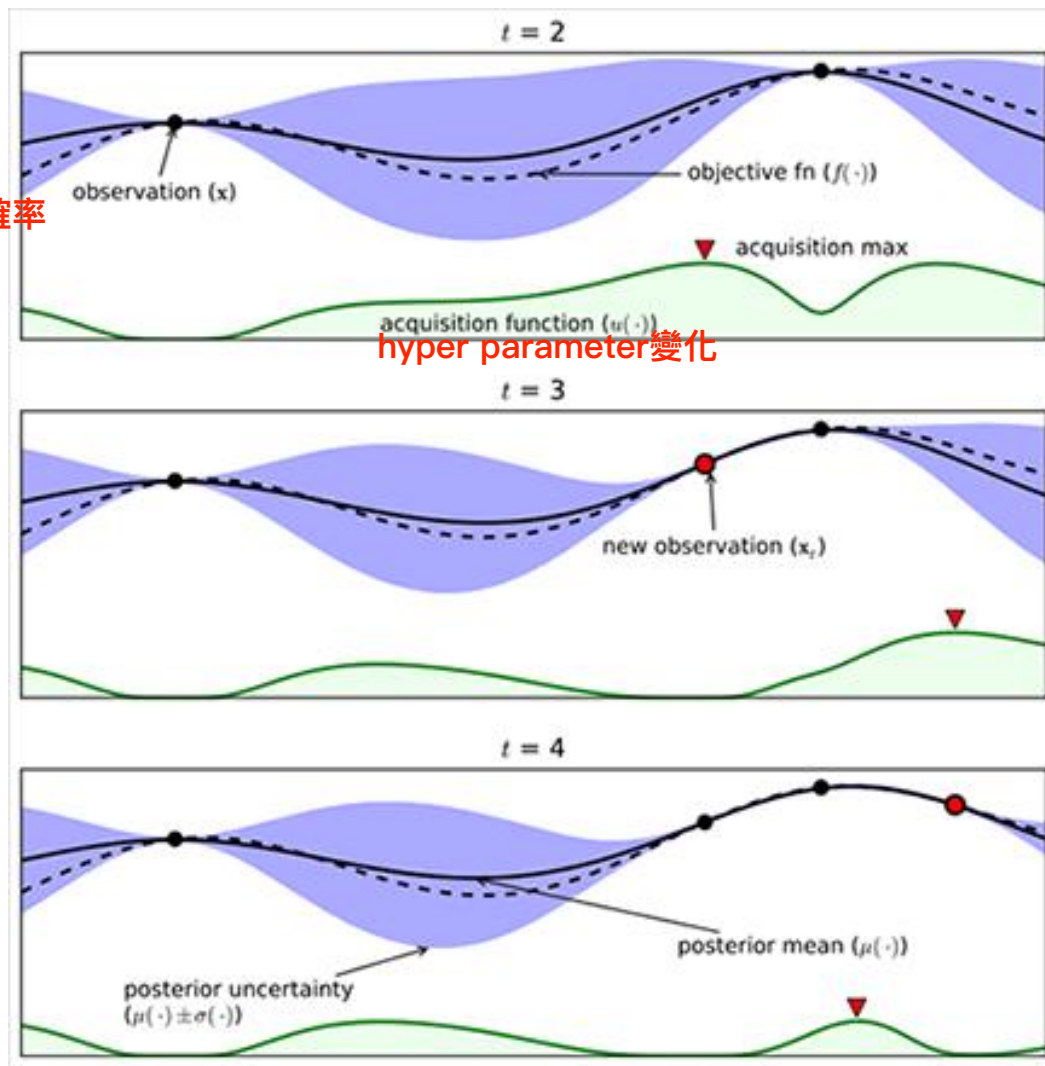
<http://www.deeplearningbook.org/contents/guidelines.html>

# Model-based Hyperparameter Optimization

信心指數，區域越大代表越沒信心

得到的正確率

hyper parameter變化



<https://cloud.google.com/blog/big-data/2017/08/hyperparameter-tuning-in-cloud-machine-learning-engine-using-bayesian-optimization>

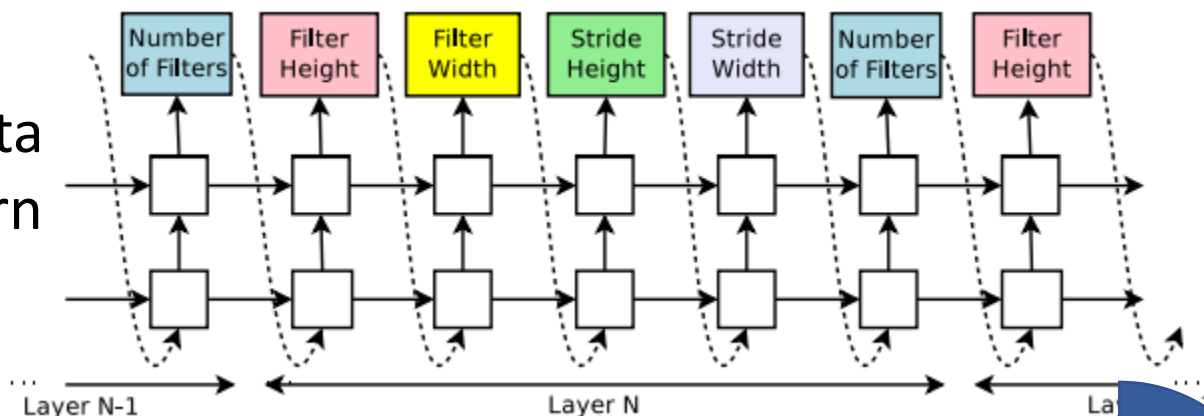
# Reinforcement Learning

每次output的數值代表network的架構

It can design LSTM as shown in the previous lecture.

800 GPUs .....

One kind of meta learning (or learn to learn)

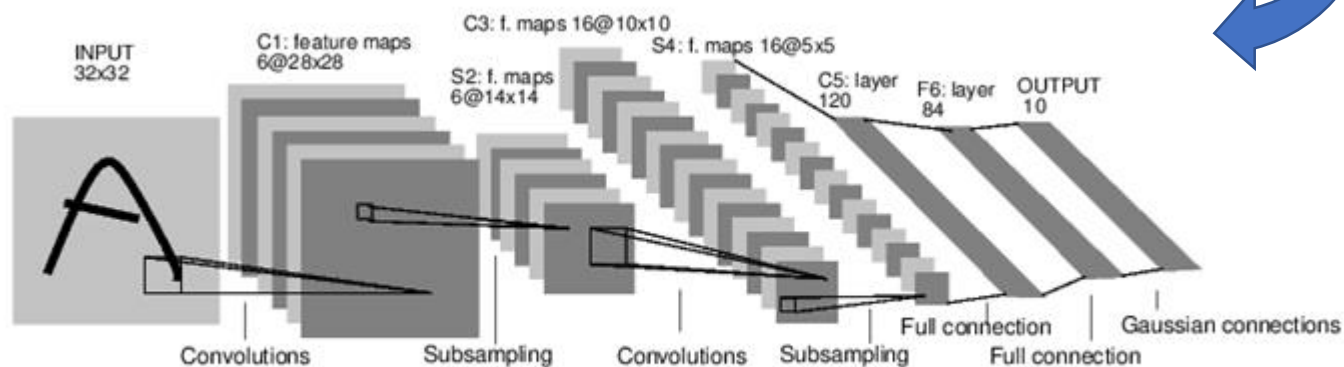


Reinforcement Learning, 將train network視為某種actor

Accuracy as reward

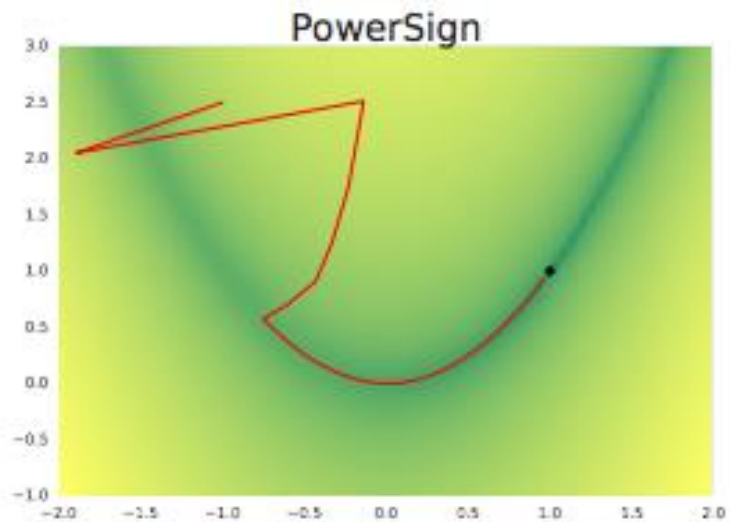
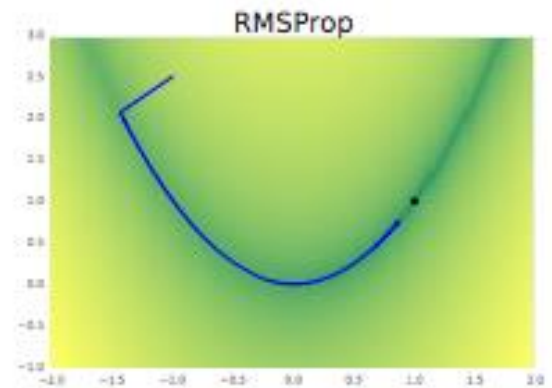
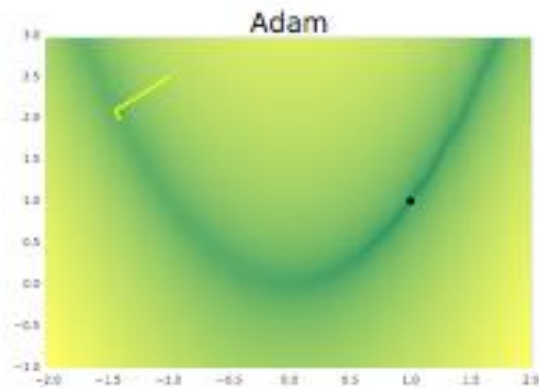
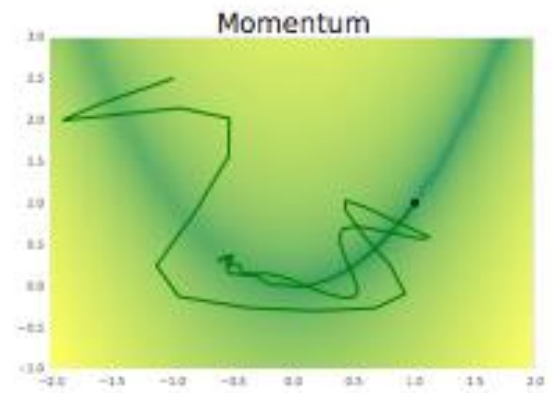
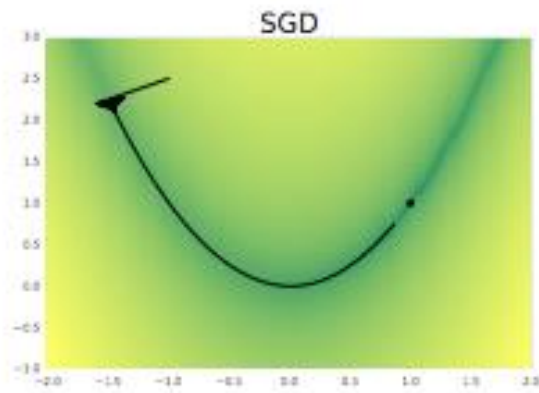
Design a network

Train the network



A Full Convolutional Neural Network (LeNet)



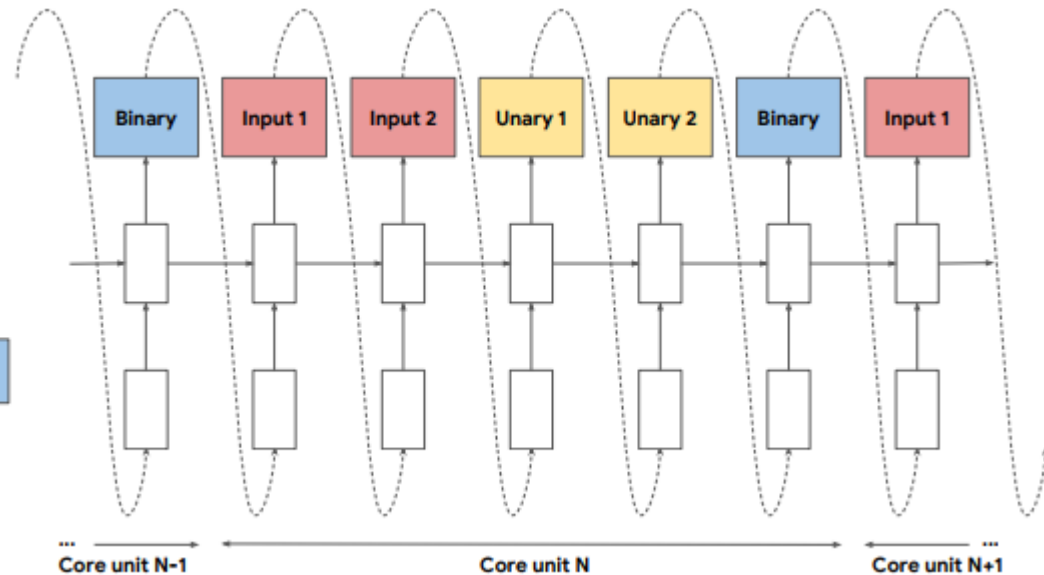
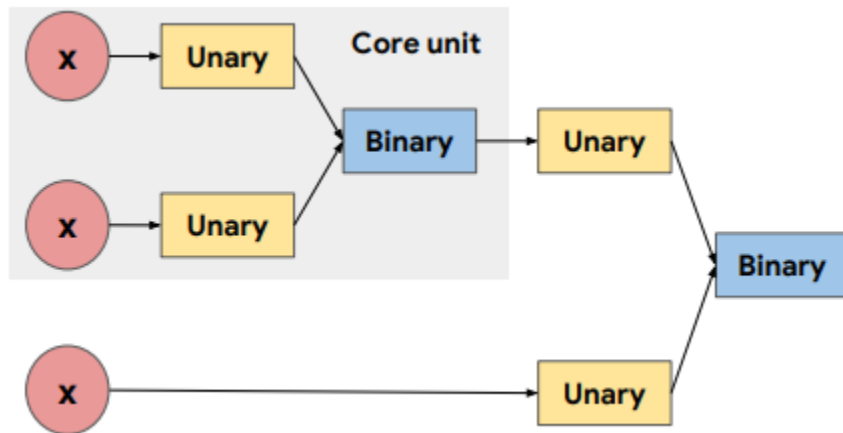


$$e^{\text{sign}(g) * \text{sign}(m)} * g$$

Can transfer to  
new tasks

# SWISH .....

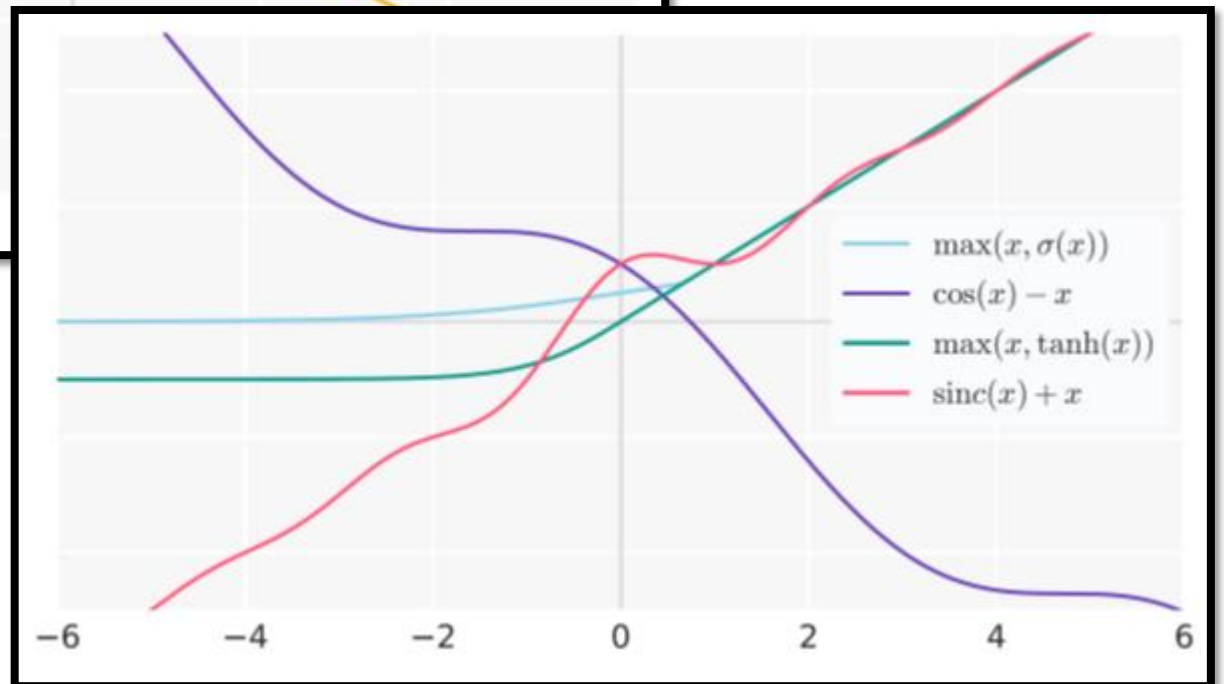
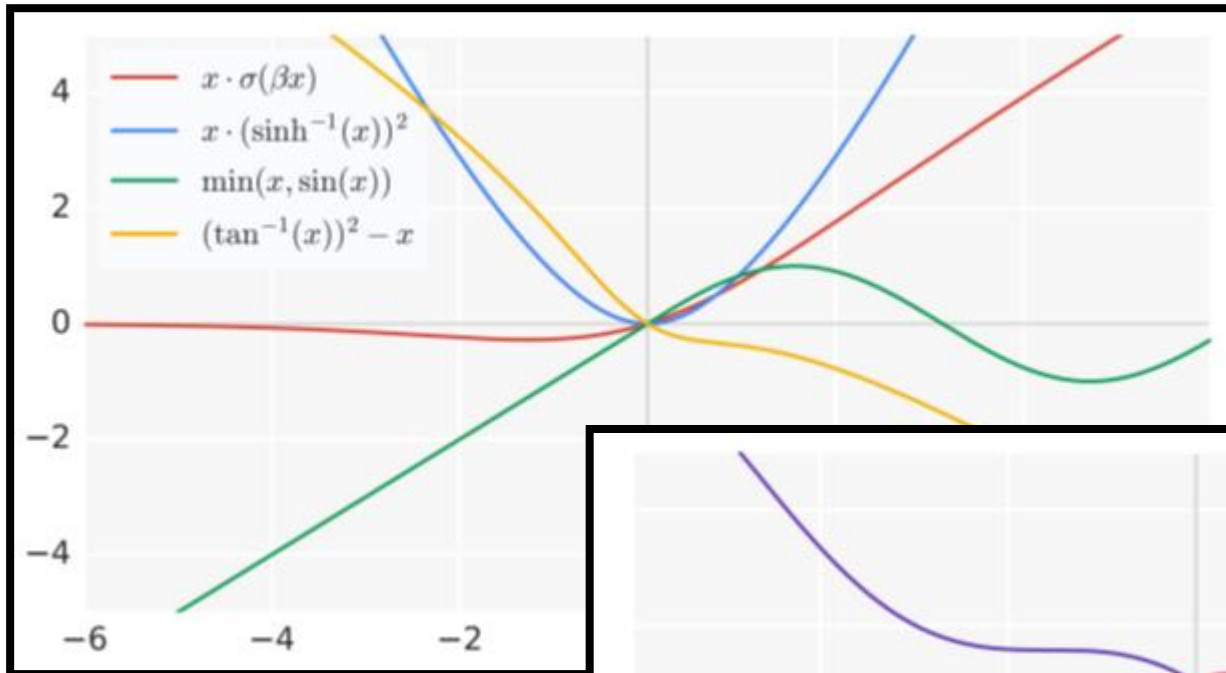
for activation function



- Unary functions:**  $x$ ,  $-x$ ,  $|x|$ ,  $x^2$ ,  $x^3$ ,  $\sqrt{x}$ ,  $\beta x$ ,  $x + \beta$ ,  $\log(|x| + \epsilon)$ ,  $\exp(x) \sin(x)$ ,  $\cos(x)$ ,  $\sinh(x)$ ,  $\cosh(x)$ ,  $\tanh(x)$ ,  $\sinh^{-1}(x)$ ,  $\tan^{-1}(x)$ ,  $\text{sinc}(x)$ ,  $\max(x, 0)$ ,  $\min(x, 0)$ ,  $\sigma(x)$ ,  $\log(1 + \exp(x))$ ,  $\exp(-x^2)$ ,  $\text{erf}(x)$ ,  $\beta$
- Binary functions:**  $x_1 + x_2$ ,  $x_1 \cdot x_2$ ,  $x_1 - x_2$ ,  $\frac{x_1}{x_2 + \epsilon}$ ,  $\max(x_1, x_2)$ ,  $\min(x_1, x_2)$ ,  $\sigma(x_1) \cdot x_2$ ,  $\exp(-\beta(x_1 - x_2)^2)$ ,  $\exp(-\beta|x_1 - x_2|)$ ,  $\beta x_1 + (1 - \beta)x_2$

swish:  $a = z * \text{sigmoid}(bz)$

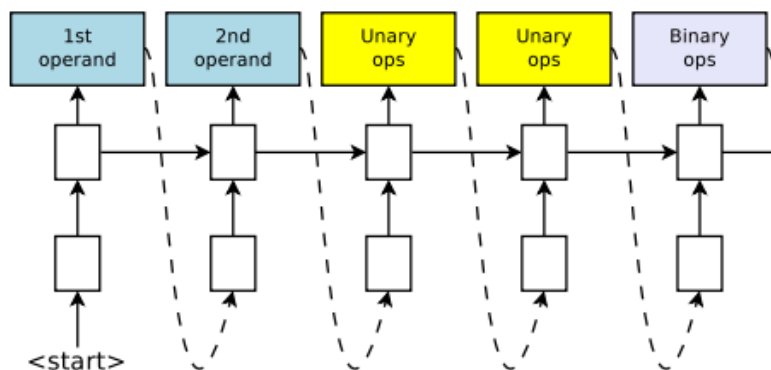
# SWISH .....





# Learning Rate

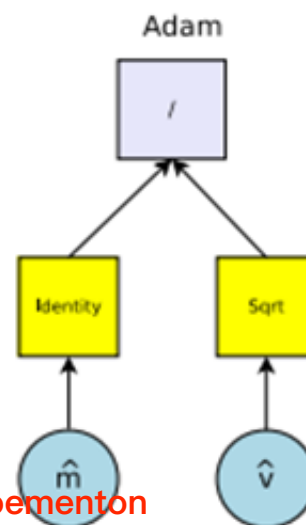
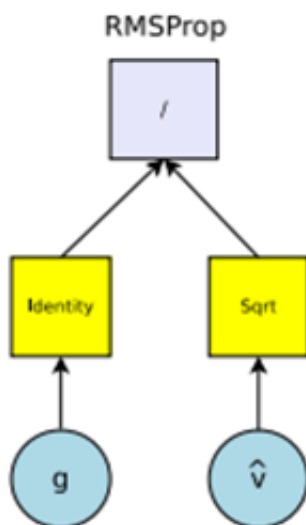
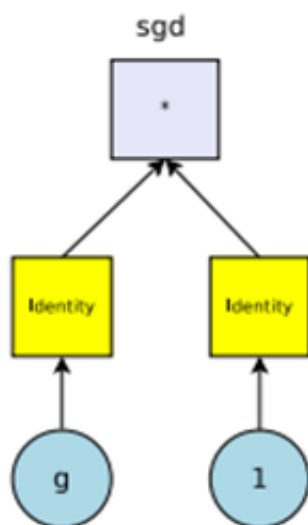
for optimizer



- **Operands:**  $g$ ,  $g^2$ ,  $g^3$ ,  $\hat{m}$ ,  $\hat{v}$ ,  $\hat{\gamma}$ ,  $\text{sign}(g)$ ,  $\text{sign}(\hat{m})$ , 1, 2,  $\epsilon \sim N(0, 0.01)$ ,  $10^{-4}w$ ,  $10^{-3}w$ ,  $10^{-2}w$ ,  $10^{-1}w$ , Adam and RMSProp.

- **Unary functions** which map input  $x$  to:  $x$ ,  $-x$ ,  $e^x$ ,  $\log|x|$ ,  $\sqrt{|x|}$ ,  $\text{clip}(x, 10^{-5})$ ,  $\text{clip}(x, 10^{-4})$ ,  $\text{clip}(x, 10^{-3})$ ,  $\text{drop}(x, 0.1)$ ,  $\text{drop}(x, 0.3)$ ,  $\text{drop}(x, 0.5)$  and  $\text{sign}(x)$ .

- **Binary functions** which map  $(x, y)$  to  $x + y$  (addition),  $x - y$  (subtraction),  $x * y$  (multiplication),  $\frac{x}{y + \delta}$  (division),  $x^y$  (exponentiation) or  $x$  (keep left).



momenton

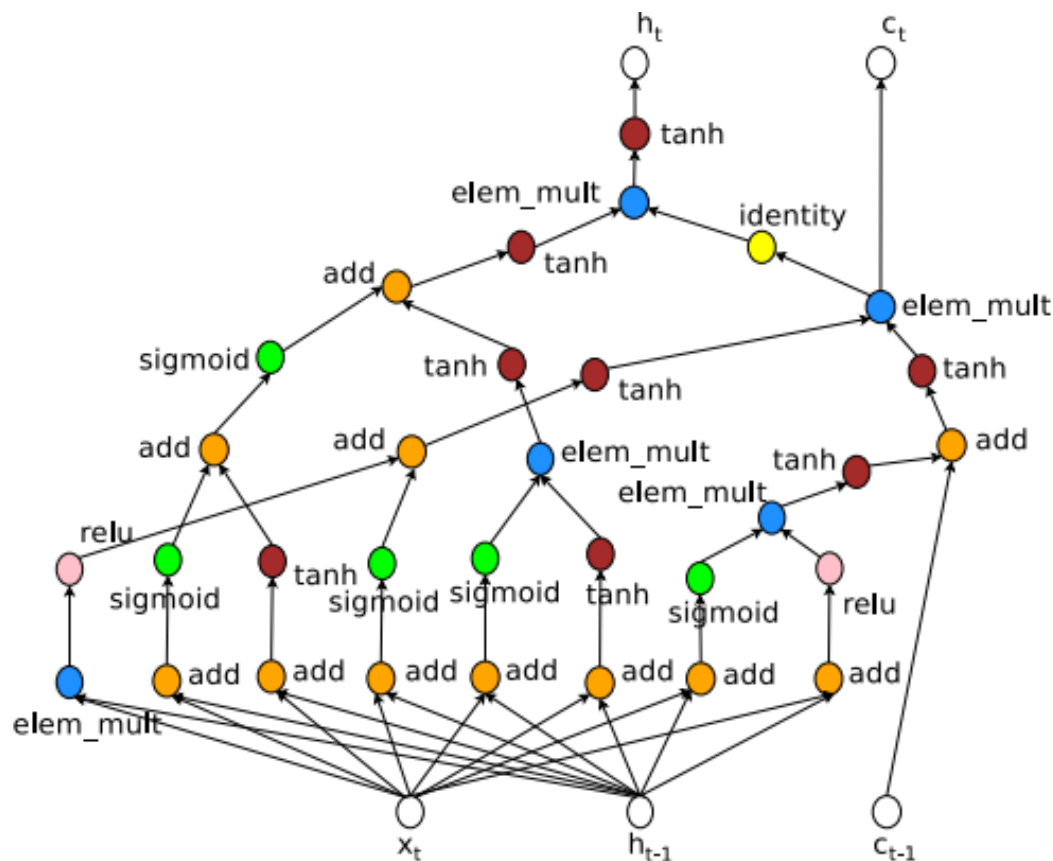
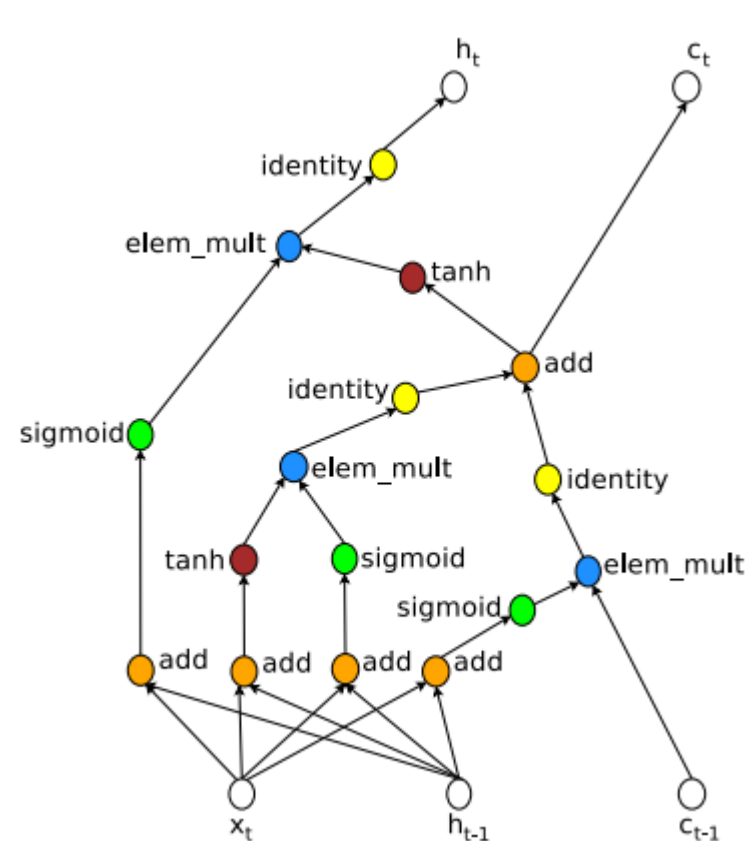
gradient的平方和

# Neural Architecture Search with Reinforcement Learning

for LSTM

*LSTM*

*From Reinforcement Learning*



Efficient Neural Architecture Search via Parameter Sharing. arXiv, 2018

概念是train過的block直接將他的參數initial來用