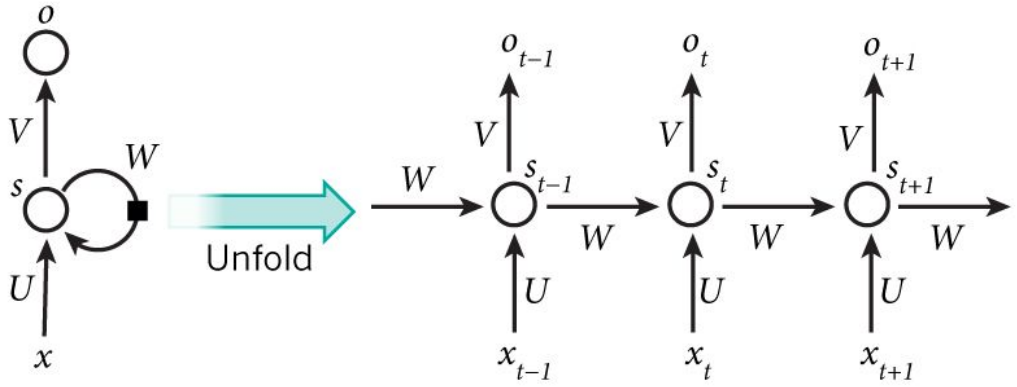
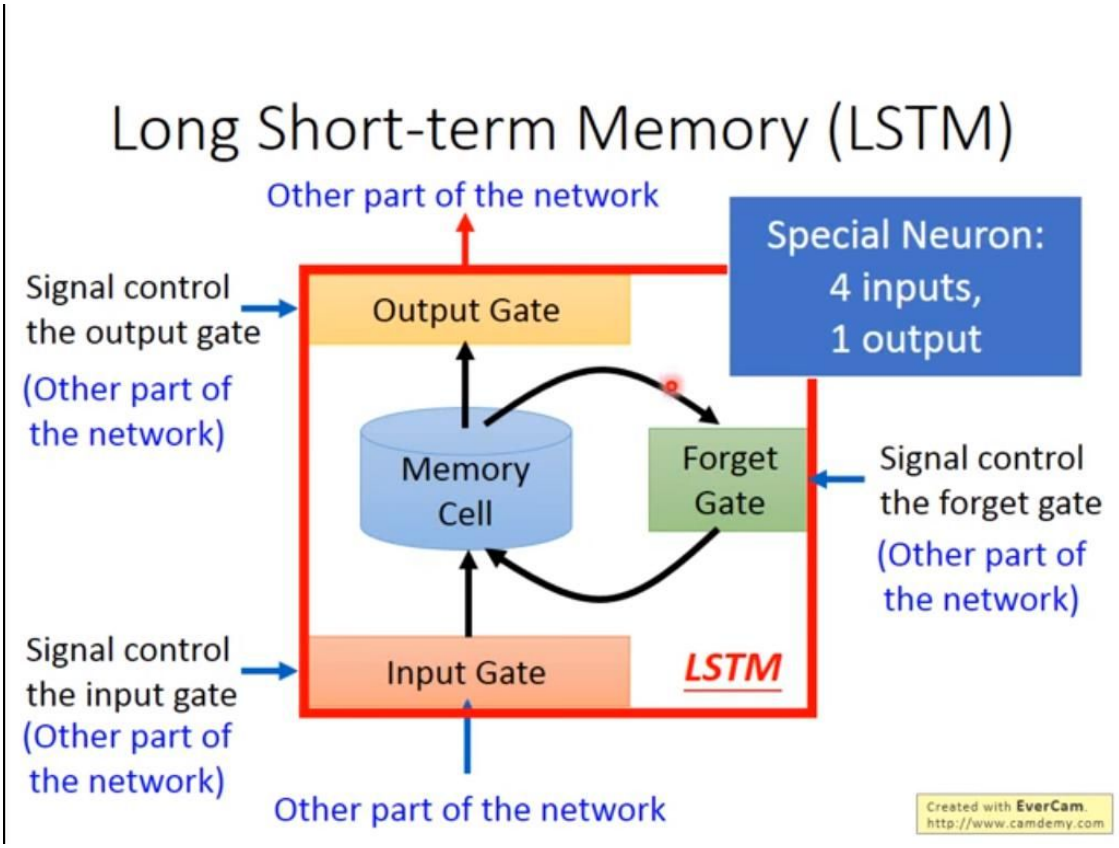
RNN

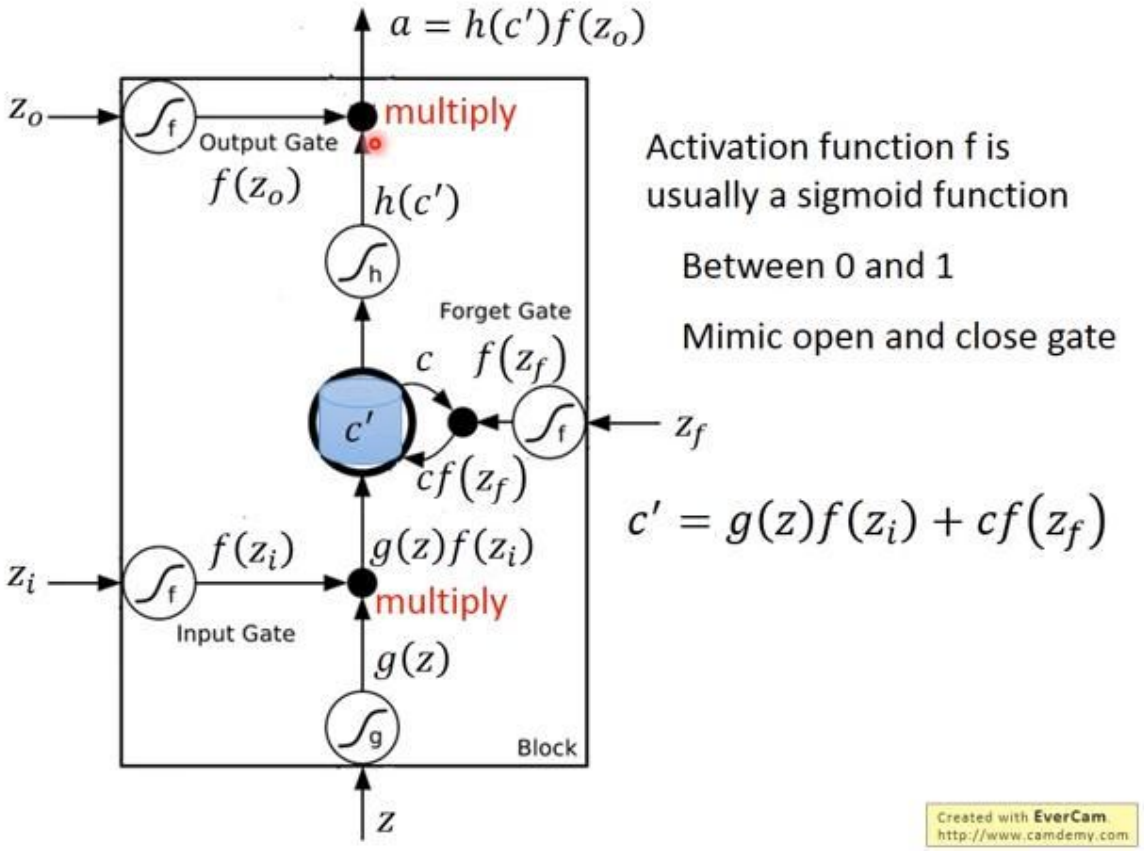


Simplified LSTM

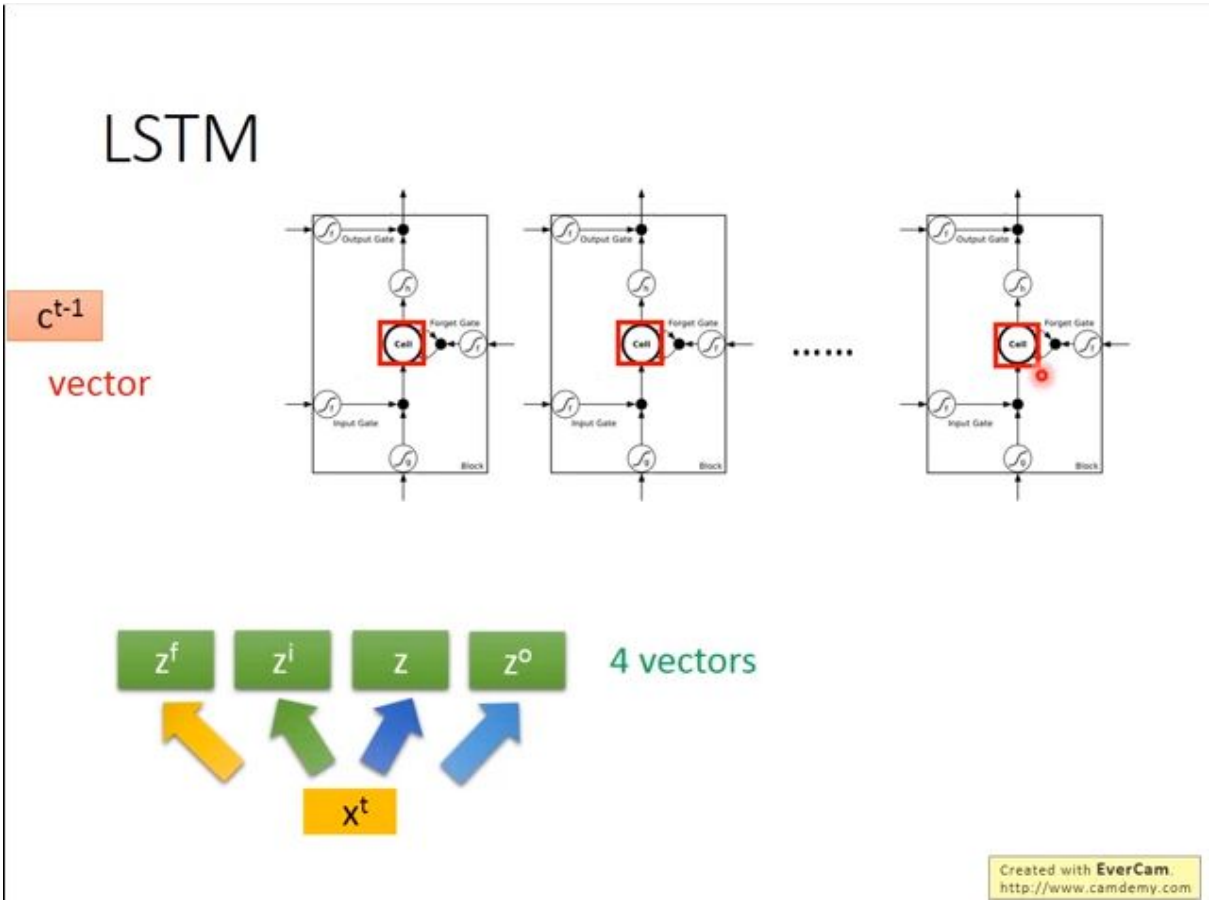
解決梯度消失的問題



閘門	開關對象
Input Gate	接收從其他neuron傳來的值
Forget Gate	記得這個Memory Cell前世的記憶
Output Gate	把output傳給其他neuron

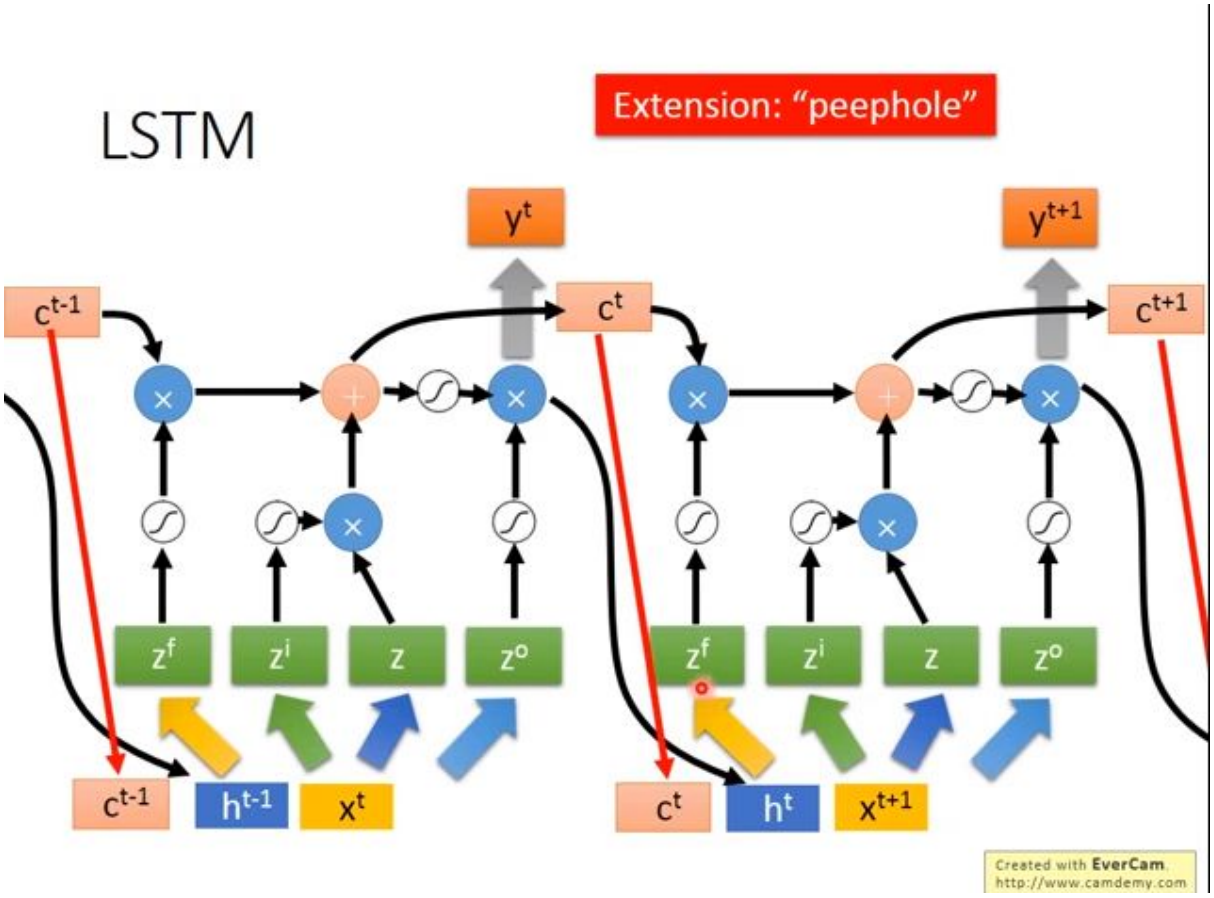


z	要進到 cell 裡的 input
z_i	操控 input gate 的 signal
z_f	操控 forget gate 的 signal
z_o	操控 output gate 的 signal
$g(z)$	z 的 activation function (input activation function，通常為 tanh)
$f(z)$	z_i, z_f, z_o 的 activation function，$f(z_i)$, $f(z_f)$, $f(z_o)$ 通常為 sigmoid function
c	原本存在 memory cell 裡面的值
c'	更新後的 memory cell 值
$h(c')$	c' 的 activation function (output activation function，通常為 tanh)



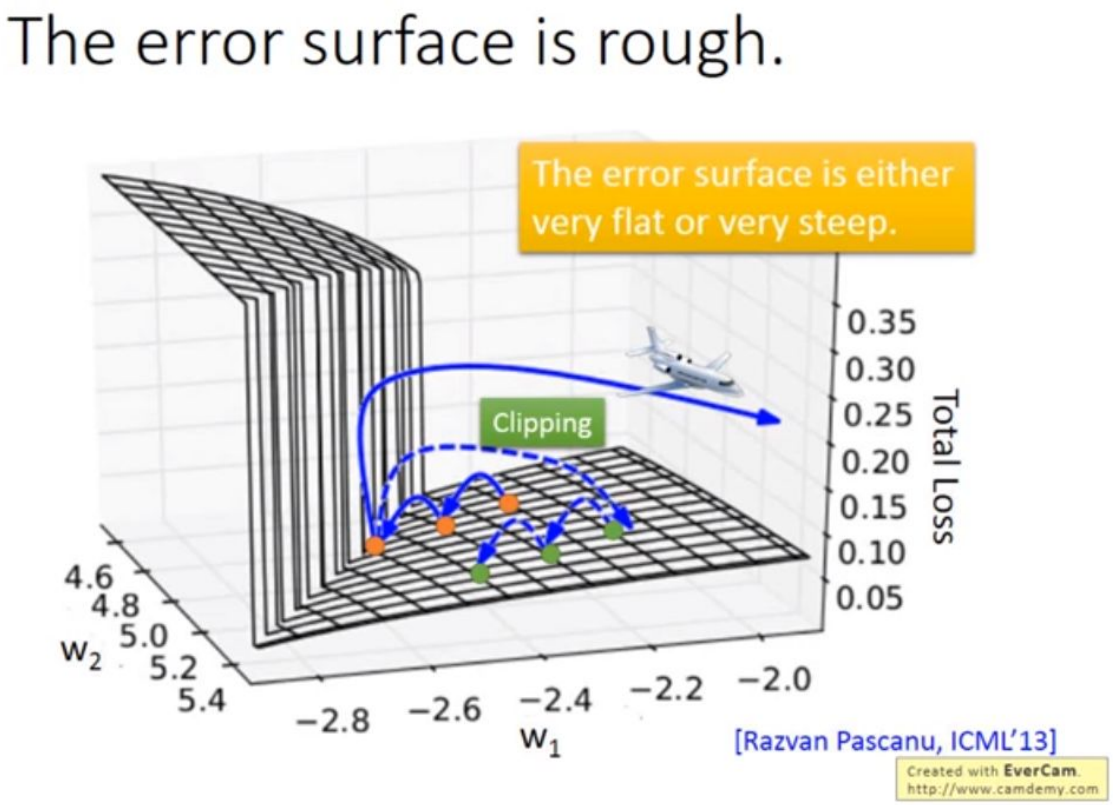
將 x^t 轉換成 z, z_f, z_i, z_o

$z = wx^t + b$



然而大多數的 LSTM 的輸入不會只有 x^t ，另外包含

- h^{t-1} ：在 t-1 時間點的 這層 hidden layer 的輸出 (各個 node 輸出 組成的向量)
- c^{t-1} ：在 t-1 時間點的 memory cell 記憶 (**peephole，也就是讓Cell跟所有gate產生連結**)



error function 區域要嘛很平坦，要嘛很陡峭
因此當 gradient descent

(1) 從高處走到懸崖邊， learning rate 很大，gradient 又很大時，下一步會走超遠

(2) 而從低處接近懸崖邊時，雖然走很小步，但會走到很陡的地方，又回到(1)的情況了

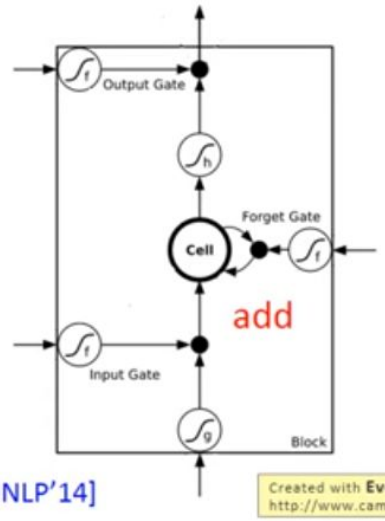
如此一來training次數再多都無法收斂。

解法：Gradient Clipping

Helpful Techniques

- Long Short-term Memory (LSTM)
  - Can deal with gradient vanishing (not gradient explode)
    - Memory and input are ***added***
    - The influence never disappears unless forget gate is closed

    → No Gradient vanishing (If forget gate is opened.)



Gated Recurrent Unit (GRU): simpler than LSTM [Cho, EMNLP'14]

LSTM 可以解決RNN會 梯度消失 的問題 (當大多數forget gate都open，即不忘記)

K. Greff. (2015) **LSTM: A Search Space Odyssey**

arXiv:1503.04069, 2015.

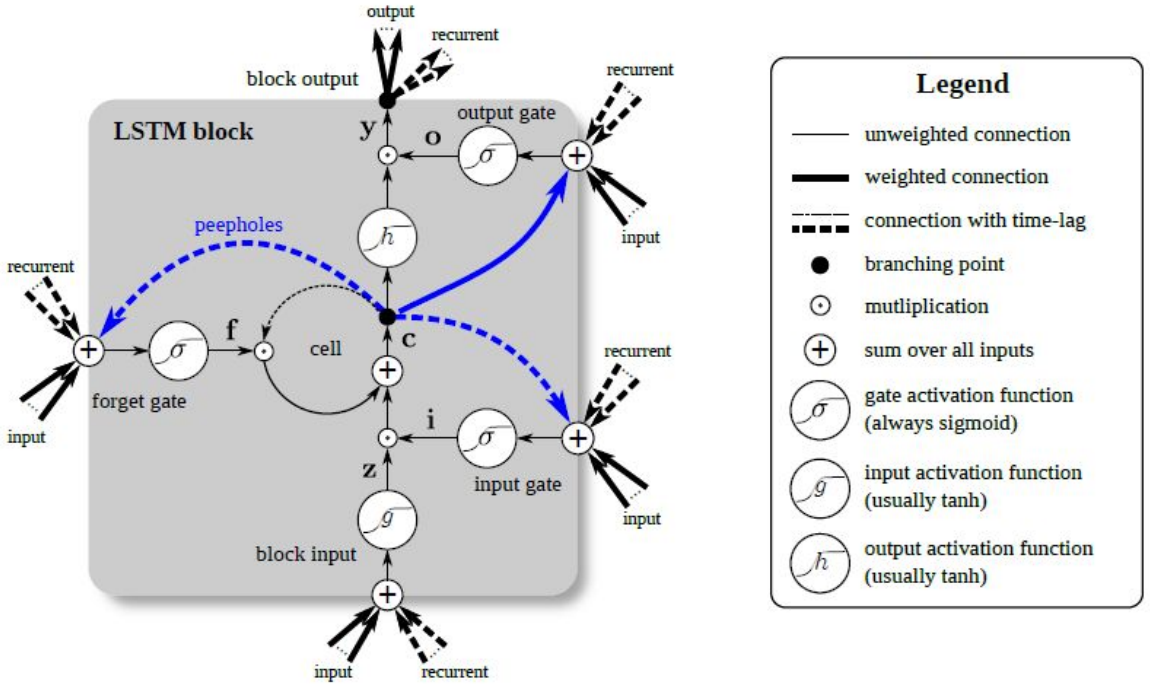
1. Introduction

本篇為第一個對LSTM的8個變種進行大規模分析的論文，做5400次的實驗 (8個LSTM變種、3種任務、每種 LSTM 做200次實驗以調整 Hyperparameter)，這些實驗累計約耗費15年的CPU時間。

fANOVA

fANOVA會決定每一個 Hyperparameter 對於網絡結構表現的影響程度。它會就模式的表現建立一個預測模式，並作為 Hyperparameter 的函數。這一非線性模式隨即被分解成為 Hyperparameter 的相互作用函數。

2. Vanilla LSTM



$$\mathbf{z}^t = g(\mathbf{W}_z\mathbf{x}^t + \mathbf{R}_z\mathbf{y}^{t-1} + \mathbf{b}_z) \qquad \textit{block input}$$

$$\mathbf{i}^t = \sigma(\mathbf{W}_i\mathbf{x}^t + \mathbf{R}_i\mathbf{y}^{t-1} + \mathbf{p}_i \odot \mathbf{c}^{t-1} + \mathbf{b}_i) \qquad \textit{input gate}$$

$$\mathbf{f}^t = \sigma(\mathbf{W}_f\mathbf{x}^t + \mathbf{R}_f\mathbf{y}^{t-1} + \mathbf{p}_f \odot \mathbf{c}^{t-1} + \mathbf{b}_f) \qquad \textit{forget gate}$$

$$\mathbf{c}^t = \mathbf{i}^t \odot \mathbf{z}^t + \mathbf{f}^t \odot \mathbf{c}^{t-1} \qquad \textit{cell state}$$

$$\mathbf{o}^t = \sigma(\mathbf{W}_o\mathbf{x}^t + \mathbf{R}_o\mathbf{y}^{t-1} + \mathbf{p}_o \odot \mathbf{c}^t + \mathbf{b}_o) \qquad \textit{output gate}$$

$$\mathbf{y}^t = \mathbf{o}^t \odot h(\mathbf{c}^t) \qquad \textit{block output}$$

3. History of LSTM

3.1 Original Formulation - 1995

- **cells, Input Gate, Output Gate**
- **NO** Forget Gate, Peephole
- Backpropagation Through Time 只針對 gradient of cell
- full gate recurrence (deprecated)

$$\tilde{i}^t = W_i x^t + R_i y^{t-1} + p_i * c^{t-1} + b_i + R_{ii} + R_{fi} f^{t-1} + R_{oi} o^{t-1}$$

$$\tilde{f}^t = W_f x^t + R_f y^{t-1} + p_f * c^{t-1} + b_f + R_{if} + R_{ff} f^{t-1} + R_{of} o^{t-1}$$

$$\tilde{o}^t = W_o x^t + R_o y^{t-1} + p_o * c^{t-1} + b_o + R_{io} + R_{fo} f^{t-1} + R_{oo} o^{t-1}$$

3.2 Forget Gate - 1999

3.3 Peephole Connections - 2000

3.4 Full Gradient (full BPTT) - 2005

- vanilla LSTM

3.5 Other Variants

4. Evaluation Step

以 vanilla LSTM 為基礎，比較各個LSTM變種的差異，每個LSTM的變種只調了一個變因。
調參數的部分使用了 **random search** 來得到表現好的 Hyperparameter，所有實驗都以表現最好的前10% 作為該 model 的代表。

4.1. Datasets

任務	**dataset**	**performance measure**
語音	TIMIT speech corpus	classification(61phones) error percentage
手寫	IAM Online Handwriting Database	character error rate
音樂	JSB Chorales	negative log-likelihood 參考網頁

4.2. Network Architectures & Training

Architectures：
Bidirectional LSTM

- TIMIT(語音) & IAM(手寫) dataset

Normal LSTM

- JSB Chorale(音樂) dataset

Training：
using **Stochastic Gradient Descent** (with **Nesterov-Style momentum**)

- momentum 介紹
- Nesterov momentum 介紹

full BPTT

4.3. LSTM variants

V: vanilla LSTM，即經典的 LSTM 模型
(以下都是基於 vanilla LSTM 形成的變種)
NIG:去除 input gate 得到的結構

- $g(x) = x$

NFG:去除 forget gate 得到的結構
NOG: 去除 output gate 得到的結構
NIAF:去除 input activation function 得到的結構
NOAF:去除 output activation function 得到的結構
NP:去除 peephole 得到的結構
CIFG: Coupled Input and Forget Gate，即 GRU (Gate Recurrent Unit)

- $f^t = 1 - i^t$

FGR: 在 LSTM 基礎上讓 Gate 單元互相之間都有連接(Full Gate Recurrence)

4.4. Hyperparameter Search

每個模型 對 hyperparameter 做 200 次實驗 (總共 5400 次)
調參數使用 **Random Search** ，並且本論文實驗**不做 gradient clipping**，因為發現 gradient clipping 會降低整體的表現。
Log-Uniform

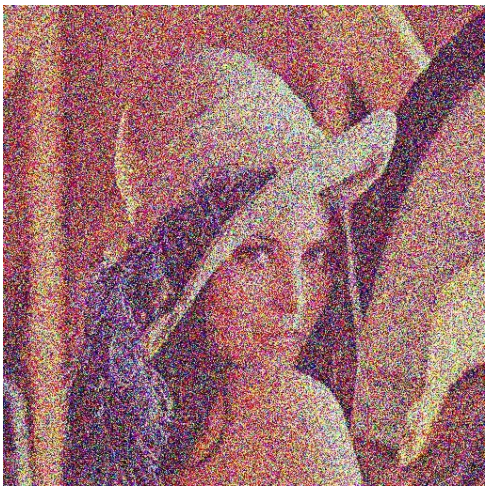
- Hidden Layer 中的 LSTM 數量
- Learning Rate
- Momentum

Uniform

- standard deviation of **Gaussian input noise**

Gaussian input noise

我們在研究一些問題的時候，經常會用到噪音，甚至有時候特地產生噪音並添加到某些信號中來研究一些問題。比如，圖像和語音識別等任務中添加一些不同的噪音來測試機器學習模型在有噪音環境下的識別率。我們就需要使用一些方法來產生噪音並且添加到原信號中去。

5. Results & Discussion

最佳表現：

- TIMIT(語音)：CIFG(GRU)
- IAM(手寫)：NIG
- JSB Chorale(音樂)：NP

NOAF、NFG 非常嚴重的降低了LSTM在三個dataset上的 performance，
FGR 在 TIMIT(語音) 和 IAM(手寫) 資料上的表現並沒有顯著改變，但是在JSB(音樂)資料上的表現很糟，不建議使用。
NIG、NOG、NOAF 在語音和手寫辨識任務中降低了許多performance。
Input Noise 通常會使LSTM的performance變糟，

6. Conclusion

一、LSTM結構
標準LSTM結構在各個datset上都能表現出較好的結果，而各個變種也並沒有顯著提高LSTM性能。另外即便使用了 Coupled Input Gate and Forget Gate (GRU)，或者移除Peephole，也不太會降低表現，而且這兩個設置不但使模型變得簡單也減少了LSTM的計算量。而根據實驗資料得出 Output Activation Function 和 Forget Gate 是LSTM中最重要的部分。

二、Hyperparameter

1. Learning Rate 是最關鍵的 Hyperparameter，然後是network的大小。然而Momentum 在這些實驗中並沒有很重要。另外 Gaussian noise 的加入，根據任務的不同有時有幫助，有時卻有害。
2. 在對各 Hyperparameter 進行交互關係的調查時，它們間並沒有表現出明顯的結構性關係，我們甚至可以將它們視為大致上是互相獨立的。
3. 關於調整Learning Rate的建議：對於一個 dataset ，可以先用一個小的 network 找到一個好的 learning rate，然後套用到大的 network 中。