

Experiments on Deep Learning (Performance Prediction as Study Case)

簡述:

此為深度學習模型的實驗，實驗數據為效能預估模型 (內含程式碼 與 互動網頁)

- 特徵: ["batchsize", "elements_matrix", "elements_kernel", "channels_in", "channels_out", "padding", "strides", "use_bias", "activation_fct"]
- 目標: "time"
- Epochs: 2000 (多少次走訪全部資料)
- Optimizer: Adam (優化方法)
- Learning Rate: 0.1 (初始學習率)
- Scheduler_Step: 400 (多少步驟下降學習率)
- Scheduler_Gamma: 0.5 (下降學習率比例)

知識點 - Loss function(損失函數):

y_hat 為 預估, y 為 真實

1. MSE:

$$MSE = \frac{1}{n} \sum (\hat{y} - y)^2$$

2. MSLE:

$$MSLE = \frac{1}{n} \sum (\log(1 + \hat{y}) - \log(1 + y))^2$$

3. Poisson:

$$Poisson = \frac{1}{n} \sum (\hat{y} - y * \log(\hat{y}))$$

4. MALPE:

$$MALPE = \frac{1}{n} \sum \left| \frac{\log(1 + \hat{y}) - \log(1 + y)}{\log(1 + y)} \right|$$

知識點 - Transformation(轉換):

1. log:

正偏移(Positively skewed)資料轉換成 類正態分佈的一種轉換

$$New\ Value = \log(Value)$$

2. sqrt:

正偏移(Positively skewed)資料轉換成 類正態分佈的一種轉換

$$New\ Value = \sqrt{(Value)}$$

3. Box-Cox^[2]:

透過選擇最優超參數lambda, 將資料轉換成最接近正態分佈的一種轉換

$$New\ Value = \begin{cases} \frac{Value^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \log(Value) & \text{if } \lambda = 0, \end{cases}$$

知識點 - Metrics (衡量指標):

y_hat 為 預估, y_bar 為 真實的平均, y 為 真實

1. ABSe: 絕對誤差

$$ABSe = \frac{1}{n} \sum |\hat{y} - y|$$

2. Re: 相對誤差

$$Re = \frac{1}{n} \sum \frac{\hat{y} - y}{y}$$

3. RMSe: 方均根誤差

類似絕對誤差的標準之一, 但涵蓋著標準差的概念在裡頭^[3]

$$RMSe = \sqrt{\frac{1}{n} \sum (\hat{y} - y)^2}$$

4. R2: 決定係數^[4]

介於 -1 至 1 之間, 接近 -1 表負相關; +1 表正相關

通常 0.9 以上為高度正相關

$$R2 = 1 - \frac{\frac{1}{n} \sum (\hat{y} - y)^2}{\frac{1}{n} \sum (\bar{y} - y)^2}$$

Experiment 1 (損失函數測試):

由於此為偏曲(skewed)資料^[1], 想藉由檢測不同的 loss function (含目標 Transormation 與 L2 正則)後, 看對準確度所帶來的影響

NN 架構: Perfneta [Exp2_A1 (金字塔型)]

特徵: ["batchsize", "elements_matrix", "elements_kernel", "channels_in", "channels_out", "padding", "strides", "use_bias", "activation_fct"]

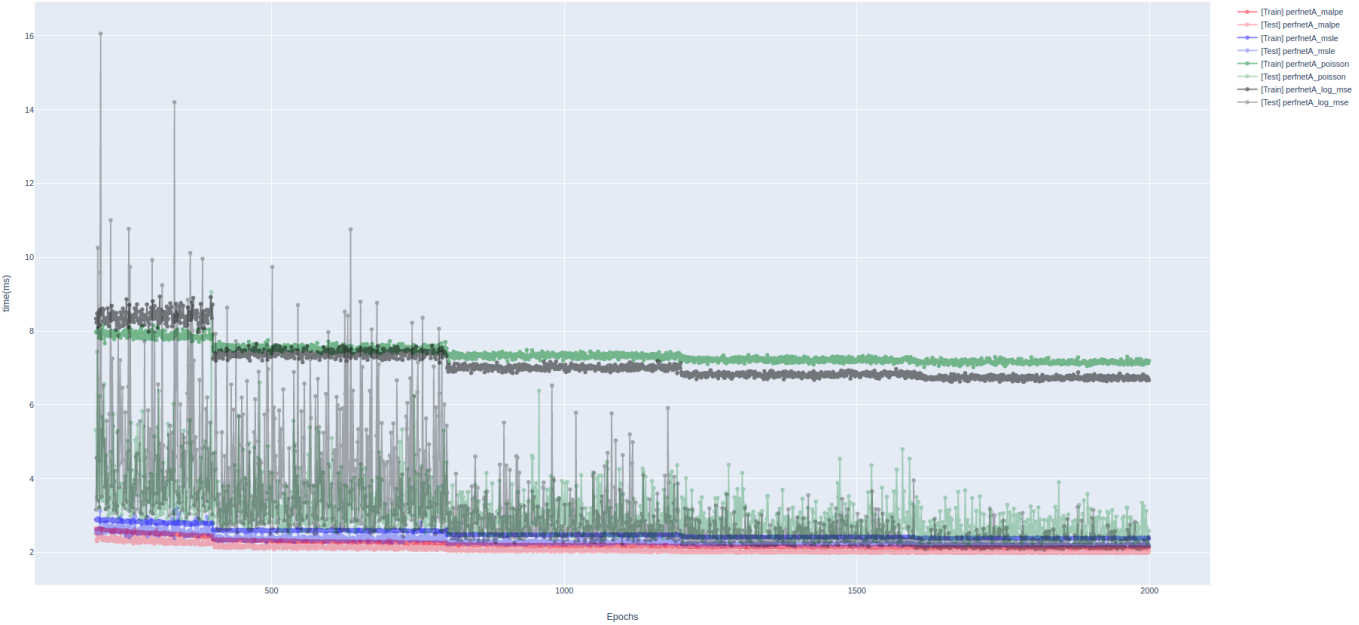
特徵轉換: sqrt: ["elements_matrix", "elements_kernel"]

損失函數與目標轉換表

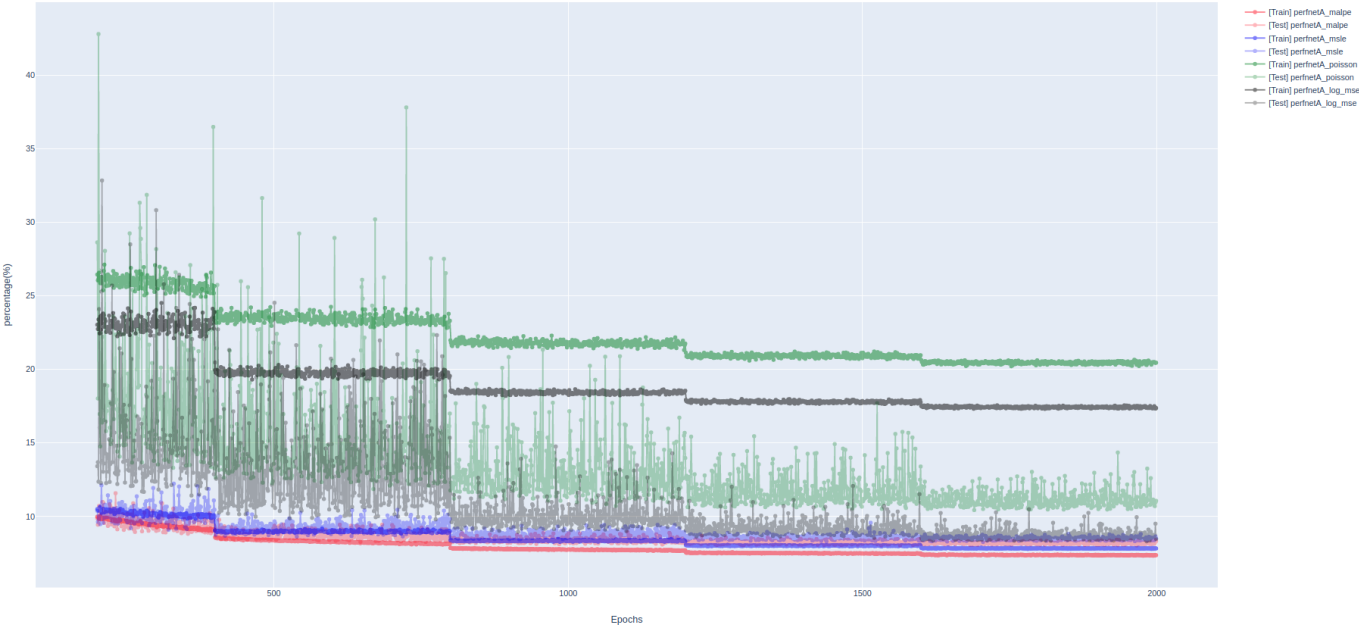
名稱	損失函數	目標轉換
Exp1_A1	MALPE	None
Exp1_A2	MLSE	None
Exp1_A3	Poisson	None
Exp1_A4	MSE	Log

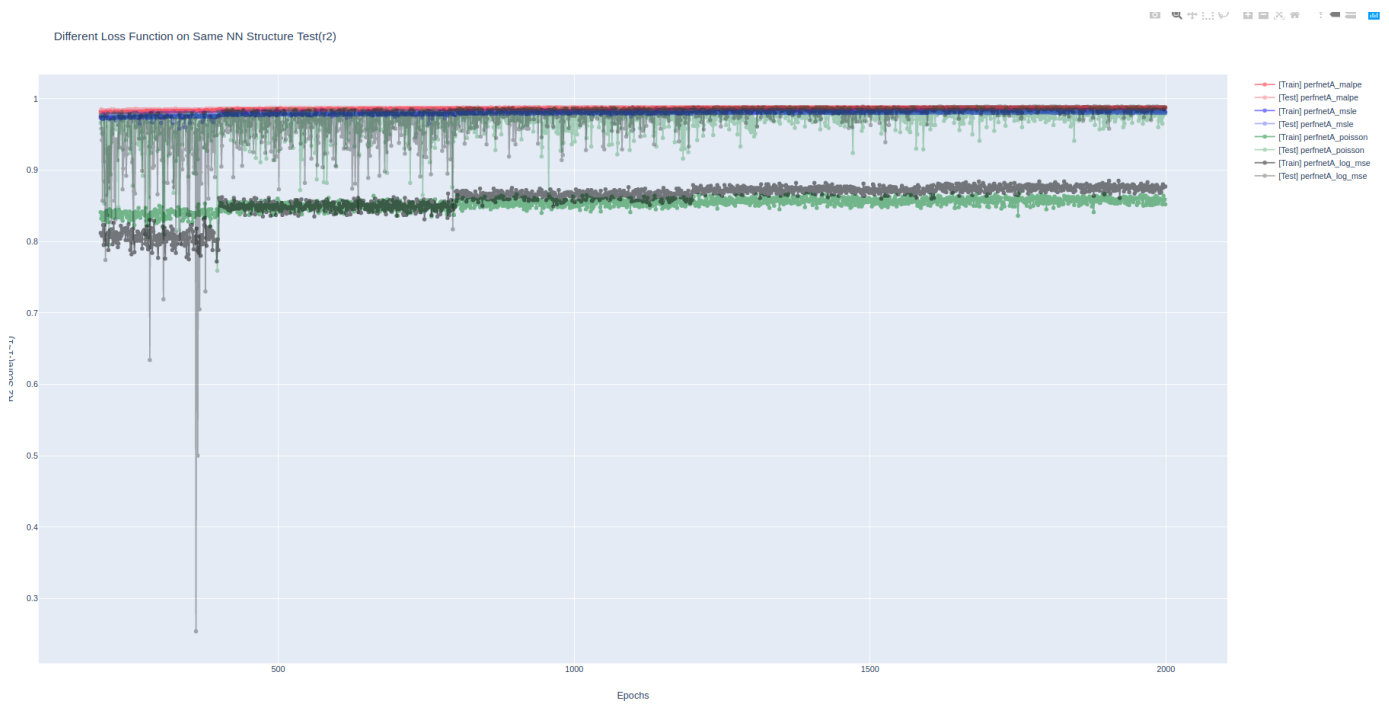
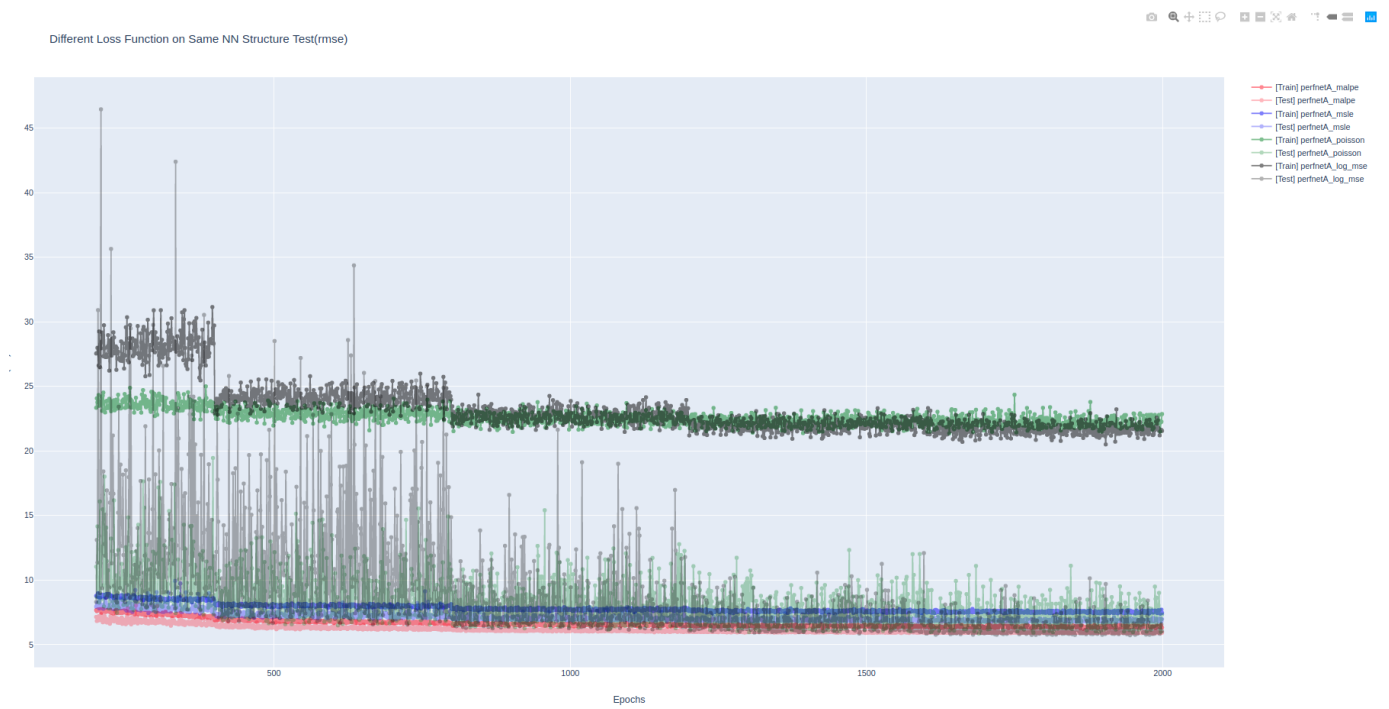
圖表(這裡先附上所有指標的圖, 以後僅提供Re, 請自行透過程式碼得到想要觀測的圖表):

Different Loss Function on Same NN Structure Test(abse)



Different Loss Function on Same NN Structure Test(re)





四項標準最好的結果:

名稱	訓練/測試	ABSe(ms)	Re(%)	RMSe(ms)	R2(-1~1)
Exp1_A1(MALPE)	Train	2.103	7.331	6.221	0.988
Exp1_A1(MALPE)	Test	1.98	8.034	5.868	0.988
Exp1_A2(MLSE)	Train	3.36	7.793	7.391	0.982
Exp1_A2(MLSE)	Test	2.178	8.269	6.752	0.984
Exp1_A3(Poisson)	Train	7.021	20.211	21.36	0.868

Exp1_A3(Poisson)	Test	2.209	10.431	5.84	0.988
Exp1_A4(Log_MSE)	Train	17.313	17.313	20.53	0.885
Exp1_A4(Log_MSE)	Test	8.309	8.309	5.72	0.989

結論:

這裡可以看出 MALPE 不論在測試或是訓練有最好的ABSe 與 Re 且 RMSe 與 R2 也不差, 故我們可以選這個當作損失函數

其中Log_MSE 在測試集有著最好的RMSe與R2, 但是訓練集卻不好, 可能是特徵過多, 資料過少或是這就是資料的真實結果, 我更傾向最後

Experiment 2 (形狀測試):

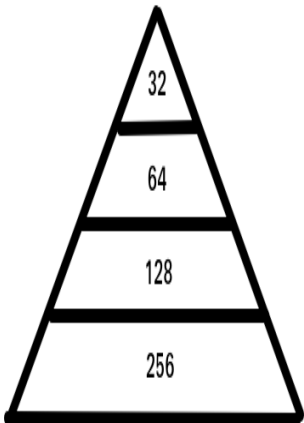
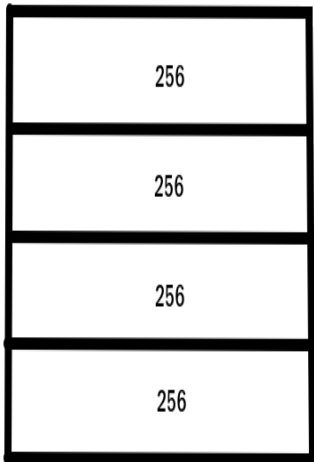
測試相同 損失函數 與 特徵(含轉換)後, 更改 NN 架構不同的形狀. 看對準確度所帶來的影響

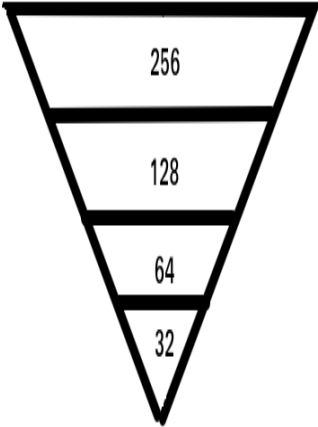
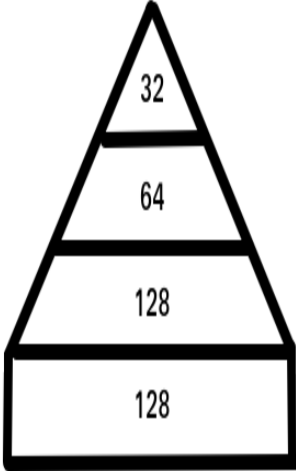
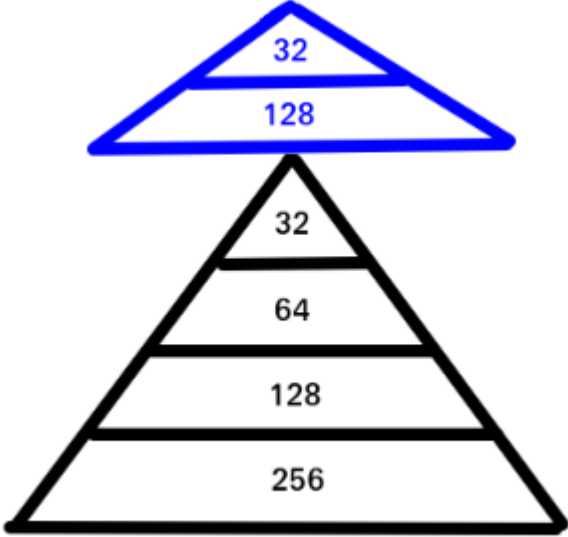
特徵: ["batchsize", "elements_matrix", "elements_kernel", "channels_in", "channels_out", "padding", "strides", "use_bias", "activation_fct"]

特徵轉換: sqrt: ["elements_matrix", "elements_kernel"]

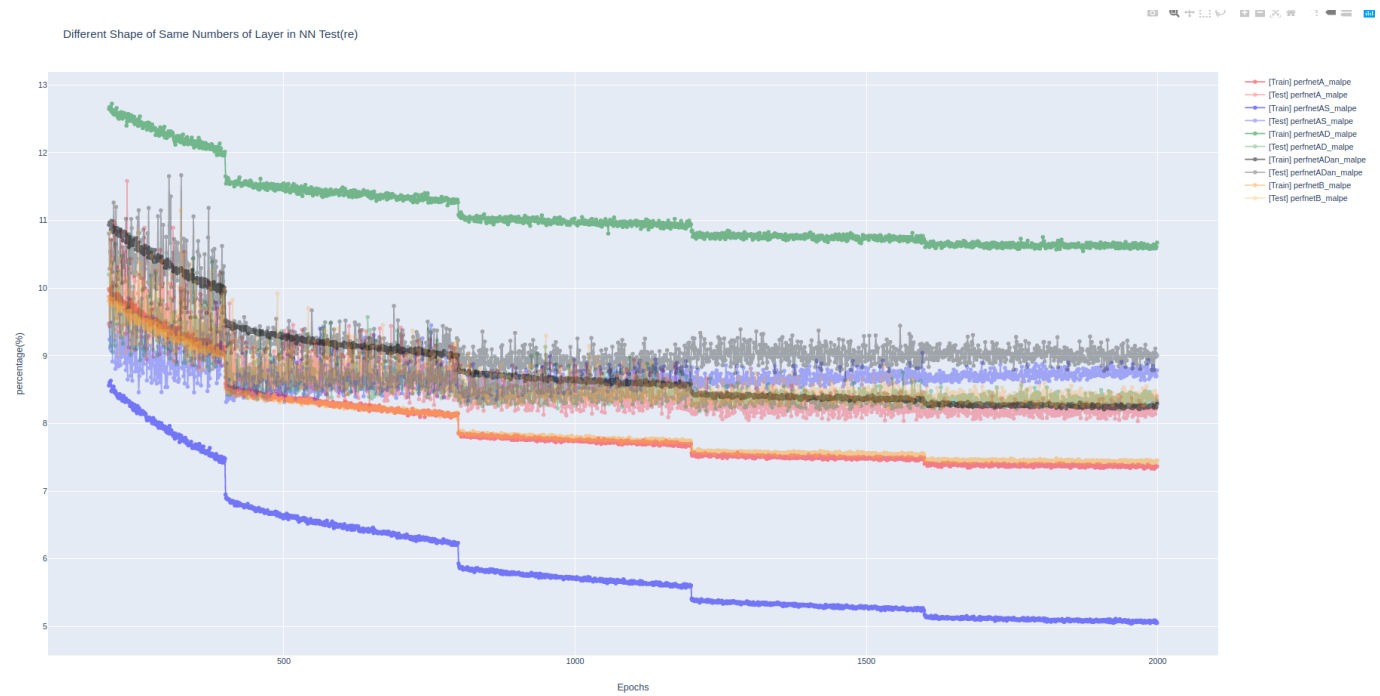
損失函數: MALPE

架構轉換表

名稱	NN 架構	形狀	備註
Exp2_A1	PerfNetA		金字塔型_全連接層
Exp2_A2	PerfNetAS		水桶型_全連接層

Exp2_A3	PerfNetAD		逆金字塔型_全連接層
Exp2_A4	PerfNetA_Daniel		論文原型_全連接層
Exp2_A5	PerfNetB		金字塔型_捲積層 + 金字塔型_全連接層

圖表



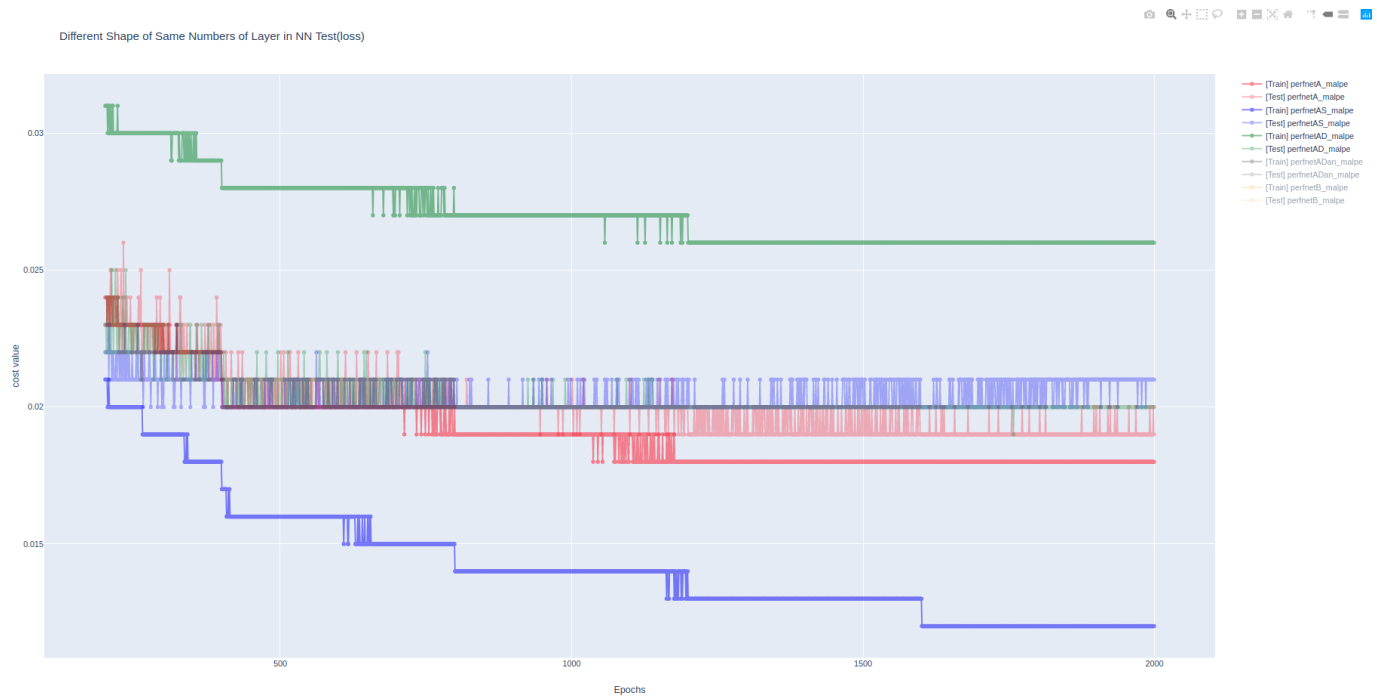
名稱	訓練/測試	Loss	ABSe(ms)	Re(%)	RMSe(ms)	R2(-1~1)
EXP2_A1(金字塔)	Train	0.18	2.103	7.331	6.221	0.988
EXP2_A1(金字塔)	Test	0.19	1.98	8.034	5.868	0.988
EXP2_A2(水桶)	Train	0.12	1.797	5.042	5.688	0.990
EXP2_A2(水桶)	Test	0.20	1.96	8.315	5.657	0.989
EXP2_A3(逆金)	Train	0.26	3.067	10.546	8.451	0.978
EXP2_A3(逆金)	Test	0.19	1.997	8.181	5.808	0.989
EXP2_A4(論文原型)	Train	0.20	2.346	8.206	6.777	0.985
EXP2_A4(論文原型)	Test	0.21	2.107	8.599	6.077	0.988
EXP2_A5(金金)	Train	0.18	2.135	7.398	6.237	0.988
EXP2_A5(金金)	Test	0.20	2.034	8.217	5.927	0.988

結論:

明顯看出, 這個資料集中, 論文原型是準確度最低的選項

而EXP_A2 水桶型 與 EXP2_A1(金字塔), 可能都是最好的選項, 其中水桶型幾乎趨近 overfitting (EXP2_A3 則是 underfitting), 可以透過 Loss Value 加以檢驗

Loss 圖表:



一般來說, 此狀況選EXP2_A1(金字塔)對此資料的符合度最理想

Experiment 3 (層數堆疊測試):

測試相同 損失函數 與 特徵(含轉換)後, 堆疊 NN 的層數. 看對準確度所帶來的影響

Note: 其中透過 Exp2_A2(水桶) 堆疊 相同 NN 輸出 大小為 256 的全連結層

特徵: ["batchsize", "elements_matrix", "elements_kernel", "channels_in", "channels_out", "padding", "strides", "use_bias", "activation_fct"]

特徵轉換: sqrt: ["elements_matrix", "elements_kernel"]

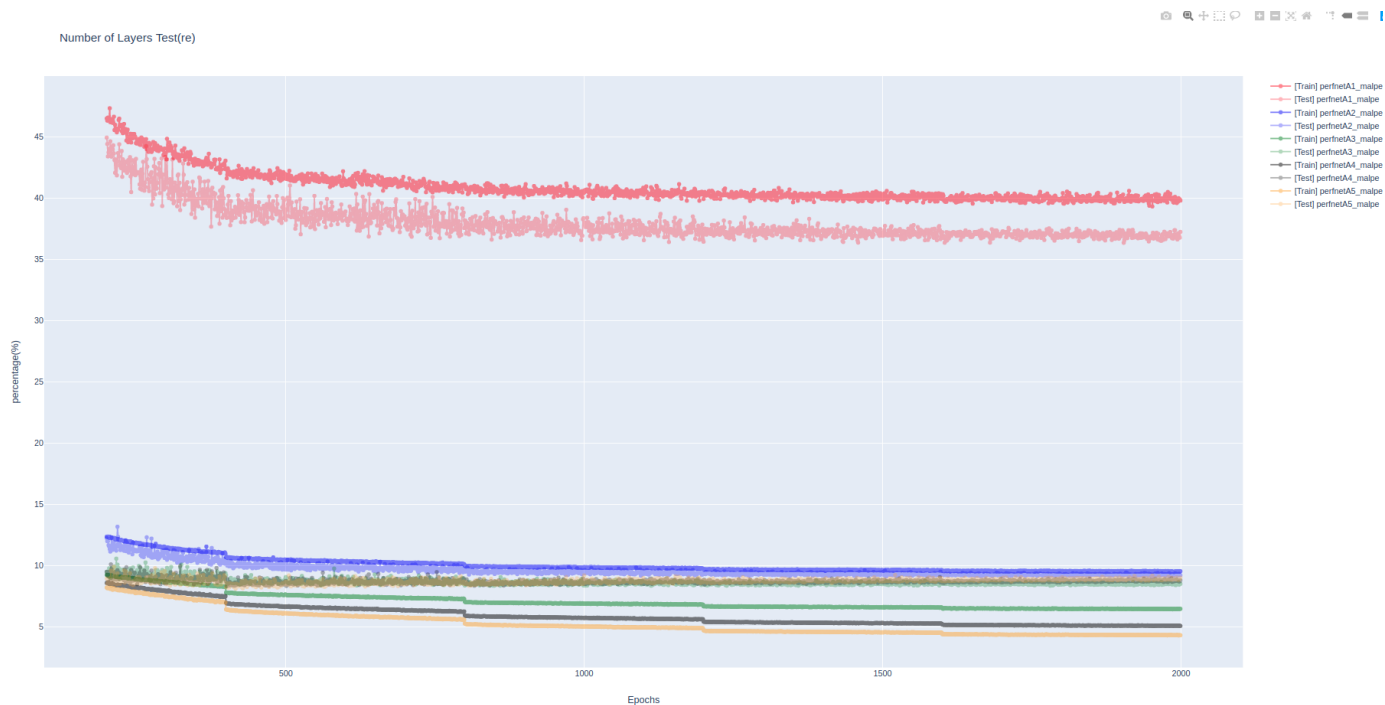
損失函數: MALPE

基本型: Exp2_A2(水桶) = Exp3_A4 (4層)

層數堆疊表:

名稱	層數	備註
Exp3_A1	1	
Exp3_A2	2	
Exp3_A3	3	
Exp3_A4	4	與 Exp2_A2(水桶) 相同架構
Exp3_A5	5	

圖表



名稱	訓練/測試	Loss	ABSe(ms)	Re(%)	RMSe(ms)	R2(-1~1)
Exp3_A1	Train	0.083	9.254	39.311	26.424	0.572
Exp3_A1	Test	0.078	8.716	36.343	25.226	0.593
Exp3_A2	Train	0.023	2.924	9.448	9.882	0.965
Exp3_A2	Test	0.022	2.564	9.084	9.037	0.970
Exp3_A3	Train	0.015	2.054	6.414	6.322	0.987
Exp3_A3	Test	0.020	2.034	8.338	5.999	0.988
Exp3_A4	Train	0.012	1.797	5.042	5.688	0.990
Exp3_A4	Test	0.020	1.96	8.315	5.657	0.898
Exp3_A5	Train	0.010	1.604	4.282	5.163	0.992
Exp3_A5	Test	0.019	1.938	8.102	5.576	0.990

結論:

可以從圖表中輕易看出, 疊超過3層之後, 效果越來越不好, 其中5 層之後可能有嚴重的 overfitting

故這邊建議 如果同樣的架構下, 這個資料使用三層或四層即可

Experiment 4 (特徵轉換測試):

測試相同 損失函數 與 nn 架構後, 更改不同的特徵轉換. 看對準確度所帶來的影響

NN 架構: Perfneta [Exp2_A1 (金字塔型)]

損失函數: MALPE


特徵: ["batchsize", "elements_matrix", "elements_kernel", "channels_in", "channels_out", "padding", "strides", "use_bias", "activation_fct"]

特徵轉換目標: ["elements_matrix", "elements_kernel"]

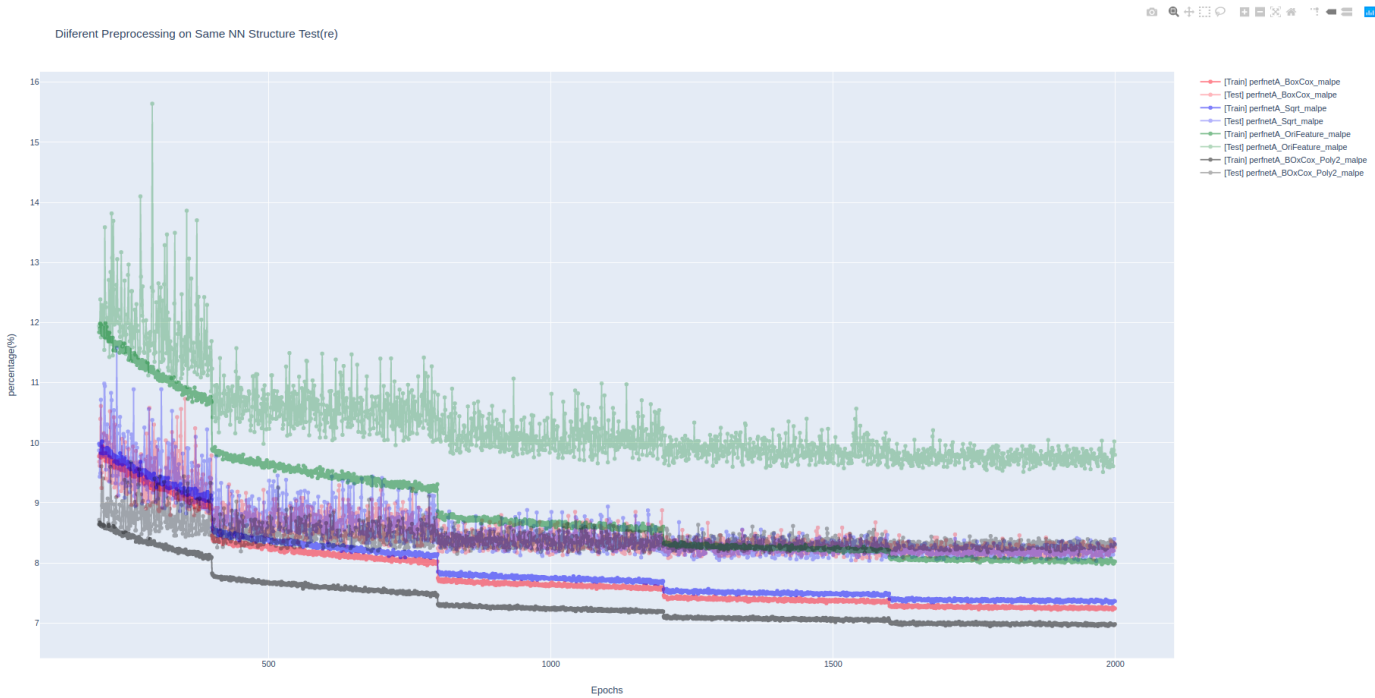
多項式轉換^[5]目標: ["batchsize", "elements_matrix", "elements_kernel", "channels_in", "channels_out", "padding", "strides", "use_bias", "activation_fct"]

多項式轉換量: 2

特徵轉換表:

名稱	特徵轉換	多項式轉換量	備註
Exp4_A1	BoxCox	None	
Exp4_A2	Sqrt	None	與 Exp1_A1 (MALPE) 相同架構
Exp4_A3	None	None	
Exp4_A4	BoxCox	2	特效全開 

圖表:



名稱	訓練/測試	ABSe(ms)	Re(%)	RMSe(ms)	R2(-1~1)
Exp4_A1	Train	2.088	7.222	6.156	0.988
Exp4_A1	Test	1.98	8.046	5.726	0.989
Exp4_A1	Train	2.103	7.331	6.221	0.988
Exp4_A1	Test	1.98	8.034	5.868	0.988
Exp4_A1	Train	2.211	7.978	6.531	0.968
Exp4_A1	Test	2.145	9.514	6.316	0.987
Exp4_A1	Train	20.57	6.947	6.145	0.988
Exp4_A1	Test	1.982	8.137	5.756	0.989

結論:

這裡可以看出對特徵轉換目標做特徵轉換是有用的, 其中以 **Box-Cox 最好**

而再進行額外的多項式轉換步驟開始時有一定效果,但在經過幾次的學習率衰減步驟後,將會與沒做多項式轉換結果類似 (i.e.: 多項式轉換在本次實驗中幾乎無效)。

Reference:

[1] Skewed Data: [Poisson regression test for skewed data](#)

[2] Box-Cox Transformation: <https://baike.baidu.com/item/Box-Cox%E5%8F%98%E6%8D%A2/10278422>

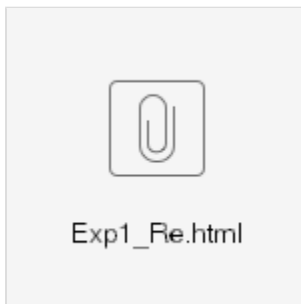
[3] RMSe: <https://zh.wikipedia.org/wiki/%E5%9D%87%E6%96%B9%E6%A0%B9%E8%AF%AF%E5%B7%AE>

[4] R2: <https://zh.wikipedia.org/zh-tw/%E5%86%B3%E5%AE%9A%E7%B3%BB%E6%95%B0>

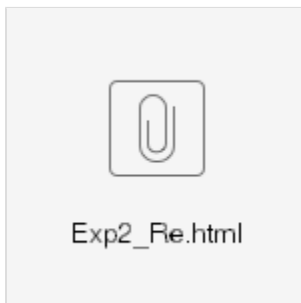
[5] Polynomial regression: https://en.wikipedia.org/wiki/Polynomial_regression

Interactive Graphics:

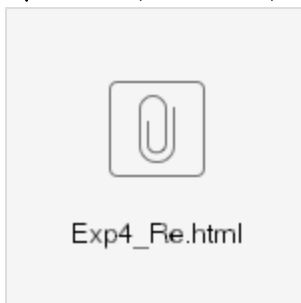
1. Experiment 1 (損失函數測試):



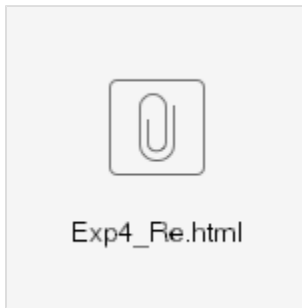
2. Experiment 2 (形狀測試):



3. Experiment 3 (層數堆疊測試):



4. Experiment 4 (特徴轉換測試):



Code:

