深度學習流言終結者

NTU MLDS 2017 Spring

題目清單

Last Update: 2017.6.11

General

1. 深度學習是不是過譽了？理論上一個隱藏層的類神經網路就可以表示任何函式，那為什麼要多個隱藏層呢？ Ref: https://arxiv.org/abs/1312.6184 (https://arxiv.org/abs/1312.6184), https://arxiv.org/abs/1402.1869 (https://arxiv.org/abs/1402.1869)
2. 深度學習真的有比其他機器學習方法厲害嘛？例如Decision Tree在某些地方會不會得到更好的結果？因為訓練的更快？
3. 與某種deep neural network參數差不多的單層hidden layer neural network以現有的learning algorithm training，是否較難train成功，e.g., accuracy無法提升，無法converge，或者converge速度慢？這可不可以證明deep neural network的優勢？
4. 在"Introduction to Neural Networks for Java, Second Edition (https://goo.gl/0kvZxr或https://goo.gl/CVQ5RE)"中提到:
   There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers, such as the following:
   – The number of hidden neurons should be between the size of the input layer and the size of the output layer.
   – The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
   – The number of hidden neurons should be less than twice the size of the input layer.
   這些究竟是真的嗎？還是真的只是個流言呢？
5. Neural Episodic Control(NEC)是否能取代Deep Q-Learning?
   Ref: https://courses.cs.ut.ee/MTAT.03.292/2017_spring/uploads/Main/Neural%20Episodic%20Control.pdf (https://courses.cs.ut.ee/MTAT.03.292/2017_spring/uploads/Main/Neural%20Episodic%20Control.pdf)
6. 深度神經網絡強大泛化能力的真正原因是什麼？Google Brain的Samy Bengio和其兄弟的兩篇論文誰是誰非?是否有方法判斷或檢驗兩篇論文提出的結論和實驗?
   Ref: DEEP NETS DON'T LEARN VIA MEMORIZATION (https://openreview.net/pdf?id=rJv6ZgHYg)
   Ref: UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING (https://arxiv.org/abs/1611.03530) GENERALIZATION (https://arxiv.org/abs/1611.03530)
7. 現在許多model都有嘗試應用在semi-supervised data上求取好的表現。那DNN領域如果以semi-supervised data可以做到多好？有label的比例會影響performance多少？他們的關係是什麼？
8. 在做 ensemble learning 時，可分成 bagging, boosting, stacking 三類。通常 bagging 與 stacking 會使用 strong model (big variance, small bias) 當作基模型，boosting 則會使用 weak model (small variance, big bias) 當作基模型。請問用 strong model 來做 boosting，以及用 weak model 來做 bagging & stacking 會有什麼問題？
9. 近年來，許許多多使用深度學習訓練的model都會利用Reinforcement Learning或Deep Reinforcement Learning的方法來增強model在實務測試的效果，不管是DNN、CNN在影像辨識、追蹤上或是Languange Mode及語音辨識上都能利用此法達到更好的效果，甚至在很多有名且複雜的 Model(ex: Alphago、自動駕駛)中也利用Reinforcement Learning大幅提升效能，使得目前在深度學習中使用Reinforcement Learning是一種很流行的方法，但是否所有的深度學習model都適合用Reinforcement Learning，是否會有model會因為使用Reinforcement Learning多花了更多的時間和運算資源而沒有學到更多特徵?甚至是比原來效果還差，到底Reinforcement Learning是真的萬靈丹還是只是過譽了?
10. 能否將highway network自動篩選layer的機制應用在general的NN上，以達成一般性的layer篩選，減少參數?

Models

1. 用限制波爾茲曼機 (restricted Boltzmann machine, RBM) 初始化類神經網路的參數曾經一度被視為深度學習和八零年代多層次感知器(Multi-layer perceptron, MLP)的關鍵差異，但近年人們已鮮少使用限制波爾茲曼機初始化參數這個技術，是什麼原因造成的呢？是因為近年使用的資料量大，而在資料量大時，初始化參數的影響較小嗎？是因為近年最佳化的方法有所改變，所以參數初始化影響變小嗎？還是也許波爾茲曼機初始化參數一開始就沒什麼幫助？
2. GRU 有優於 LSTM 嗎?
3. 普遍認為，LSTM比一般的RNN好訓練? 但據說只要好好初始化，一般RNN就可以勝過LSTM (難道說 LSTM 那麼多 gate 只是白忙一場？) Ref: https://arxiv.org/abs/1504.00941 (https://arxiv.org/abs/1504.00941)
4. LSTM 中的 Forget gate 若能在初始化時，給與較大的 bias，使其長保開啟（也就是儘量不要遺忘），長短期類神經網路會得到較好的訓練效果。
5. CNN 許多人會 padding 使輸出的大小和原本的大小相同，究竟有沒有 padding、或是 padding 多少格，才能達到較佳的效果呢?
6. 如果我們想要輸入一串陣列，但是原始資料輸入的長度並不固定，所以我們常常使用 padding，但是究竟對機器來說，被加進去填空的部分會被當成甚麼呢？padding 常常會把加進去的部分設成 0，但是這麼做真的比較好嗎?
7. 許多論文或文章中，都直接預設 LSTM forget_bias = 1，那 1 夠大嗎？足夠記住資訊嗎?
   Ref: https://arxiv.org/pdf/1701.03441.pdf (https://arxiv.org/pdf/1701.03441.pdf)
   Ref: http://jmlr.org/proceedings/papers/v37/jozefowicz15.pdf (http://jmlr.org/proceedings/papers/v37/jozefowicz15.pdf)
   Ref: https://github.com/Lasagne/Lasagne/issues/368 (https://github.com/Lasagne/Lasagne/issues/368)
8. CNN channel stride一定要用1嗎？大部分做CNN時，第一層convolutional layer都會把全部的channel做weighted sum，但是有沒有可能不要用全部channel，而是跟圖片的長寬一樣分成幾個stride？這樣的方式會不會能model到一些特殊的pattern？
9. bias的初始值重要性？建立一般神經網路或CNN時，大部分bias的初始值都直接設0，難道bias的初始值真的不會影響收斂性嗎？
10. cnn application 用pairwise train都可以gain something?
    https://arxiv.org/pdf/1511.06452.pdf (https://arxiv.org/pdf/1511.06452.pdf)
11. 現在的 CNN 感覺傾向越來越 Deep，Deep到底有什麼用處？還是只是一個潮潮的名詞而已？
12. Jordan Network的效果比Elman Network好，因output layer的值是有目標的
13. 利用深層的 RNN 搭配 skip-connection 可以類比 LSTM 解決vanishing gradient 的問題 ; Residual Network 利用 skip-connection 讓極難訓練的深層結構得以冒出頭。那是不是只要深的模型配合 skip-connection 就一定能訓練的起來，並且得到好的成果？
    (RNN 類比 LSTM reference: https://cs224d.stanford.edu/reports/mmongia.pdf (https://cs224d.stanford.edu/reports/mmongia.pdf))
14. 在一般的transfer learning中，時常是拿倒數幾層的layer當作feature, 然而在object detection, object segmentation, visual tracking， etc的應用中，當使用CNN作transfer learning時，時常會使用到multi layer feature, 並且強調較淺的layer是為了有助於提供與位置localization有關的資訊。然而，有些work指出，在一些要預測的標的與位置無關的task中(ex:image classification)，multi layer feature有助於大量的減少training time, 卻達到與DCNN相當的準確率。multi layer feature究竟只能夠在與位置有關的task中有幫助，或是in general在多數的tasks中能比deep feature更能提高performance呢？
    Ref: https://arxiv.org/abs/1503.04065 (https://arxiv.org/abs/1503.04065)
    Ref: https://pdfs.semanticscholar.org/6da4/11b5885904781a586ca68ac82f26161bca57.pdf (https://pdfs.semanticscholar.org/6da4/11b5885904781a586ca68ac82f26161bca57.pdf)
15. 在LSTM這類型的model中，各個不同的gate所負責(學到)的資訊真的有明顯的區別嗎？
16. RNN 的獨特之處就是其具有記憶性，可以把前面的資訊傳下去，那為什麼 AlphaGo 是用 DCNN 來判斷局部的情勢，而不是使用 RNN 來記錄前幾手的資訊呢?
17. Seq2Seq在訓練時，不取argmax，直接拿前面的output說不定會比較好！
18. sequential 的學習在不允許過往資料下要同時學習新知識同時保留舊知識是很困難的問題，例如當訓練了9個類別，要再多訓練一個類別時常常會受到catastrophic forgetting的影響，有什麼辦法能夠達成sequentiial的學習同時不嚴重忘記之前學過的呢？
    [Robins, Anthony. "Catastrophic forgetting, rehearsal and pseudorehearsal." Connection Science 7.2 (1995): 123-146.]
    [Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." European Conference on Computer Vision. Springer International Publishing, 2016.]
    [Rebuffi, Sylvestre-Alvise, Alexander Kolesnikov, and Christoph H. Lampert. "iCaRL: Incremental Classifier and Representation Learning." arXiv preprint arXiv:1611.07725 (2016).]
19. 現今許多state-of-the-art 朝向end-to-end training, 但是end-to-end training 有什麼代價或是無法解的問題嗎？根據Berkeley的研究 (https://simons.berkeley.edu/sites/default/files/docs/6455/berkeley2017.pdf)， gradient-based 的deep learning 在某些end-to-end 會失敗。在Facebook利用RL 產生sequence 的task 中 (https://michaelauli.github.io/papers/iclr2016_mixer.pdf)，他們設計的End-to-end架構實驗效果差很多。什麼時候適合end-th-end, 如何設計，又在什麼時候應該模組化呢？
20. 有人說 GAN 可以生成任何東西，真的是這樣嗎？GAN 是否過譽？
    (1) 自然語言的生成因為語言離散的特性，使用GAN很難train一個好的Language Generator。以往的做法都是先pre-train rnn後再用reinforcement learning + GAN的方式優化模型。儘管最近在Improved Training of Wasserstein GANs的paper中訓練了一個charactor-level的generator，從paper的生成結果來看結果並不好。GAN真的可以訓練一個自然語言的generator嗎？
    (2) 很多人用 GAN 來產生以假亂真的圖，老師上課時就用許多動漫人物的圖讓 GAN 去學習。但如果你給 GAN 學習的 data，是一個複雜度非常高的 data set (例如 ImageNet)，那這樣 GAN 學出來的東西是什麼？會不會是亂碼？
21. VAE 的 variational autoencoder (https://arxiv.org/pdf/1601.00670.pdf) 是真的有用嗎？ 多了一個variational為什麼被吹捧的那麼厲害？
22. CTC是否過譽了？真的是seq2seq的最佳解？
23. highway network因具有自動篩選layer的機制 可以train到很多層
    是否因此不管train再多層都不會overfitting？與之類似的grid LSTM是否具有類似的不會overfitting的特性？
24. 在generative adversarial network(GAN)中的generator 和 discriminator 與reinforcement learning中被廣泛使用的Actor-critic method 中的Actor network 還有 critic network 有很高的相似成分，也都很難optimize。有沒有辦法design 一個 MDP 把這兩種method連結並implement 到原本各自領域的task 中，獲得更好的performance。(Connecting Generative Adversarial Networks and

Actor-Critic Methods (https://arxiv.org/pdf/1610.01945.pdf))
25. Unsupervised learning是否沒有over-fitting的問題?
26. 加入Attention機制是否都使RNN有更好的結果(i.e., HW2)，有沒有可能反而因為Attention的訓練失敗造成model反而變差呢？而各種不同的Attention算法又是會對model有怎麼樣的影響？
27. 如何使CNN中不同的filter學習到彼此間較大差異的feature，而不是類似的、重複的feature?
28. 給定一個已經被train好的大型model的deep neural network，是否可以將這個deep neural network轉化到testing時使用較為shallow的neural network來表示這個大型model? 如果可以，其限制又為何? Ref: L. Ba and R. Caurana, "Do Deep Nets Really Need to be Deep?" NIPS, 2014.
29. 有一說 (https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf)是在母體資料上，由於Discriminative Model的Off-by-one error會小於Generative Model，因此會以為Discriminative Model優於Generative Model?
30. ResNet 每一層的 layer 在做什麼？是否過譽了？

Cost/Loss Function and Optimization

1. 你還在擔心局部最小值(local minima)的問題嗎？過去在人們的想像中，類神經網路的訓練過程中局部最小值使得類神經網路在訓練資料上未達良好結果時就已停止訓練，但傳說類神經網路的損失函數 (loss function) 局部最小值不多，訓練停下來通常是因為碰到了鞍點(saddle point)或是損失函數上特別平滑的區域。（所以不要再擔心局部最小值的問題了，遇到問題用深度類神經網路學下去就對了）
2. 過去在人們的想像中，使用梯度下降法進行訓練時，損失值不再下降是因為在損失函數上碰到了梯度趨近於零的位置，但傳說往往損失值不再下降時梯度仍然很大，這種狀況常發生嗎？為什麼會這樣？（請見 Ian Goodfellow 所著的 《deep learning》 P276）
3. 在 cost function 中加入 L1、L2 項可「防止過擬和」或「確保稀疏性」等，然而這兩者功用是不同的，例如：L1 比L2 更能確保稀疏性（可參考：http://stats.stackexchange.com/questions/45643/why-l1-norm-for-sparse-models (http://stats.stackexchange.com/questions/45643/why-l1-norm-for-sparse-models)）。那麼，兩者之間是否還有其他區別呢？或是有沒有哪一種在大部分的情況下能得到較好的結果呢？
4. classification一定只能用cross entropy嗎？在regression的題目中，有很多種loss function，如L2 loss, L1 loss等。但在classification的題目中，大部分的loss function都只會用cross entropy (log loss)，有沒有別的loss function可以用？
5. 許多影響生成的模型普遍會使用 L2 Loss 來衡量模型表現，例如：super resolution 是一種將模糊照片變清晰的技術，一般都使用 L2 loss 來衡量與 ground truth 的差距，但可能會造成局部性影像不和諧。是否 L2 loss 是唯一的衡量方式？

Optimizer

1. Adam 是不是過譽了？Adam很常是訓練類神經網路的預設方法，但傳說在訓練 generative adversarial network 時 Adam 表現不佳。Adam 收斂快，但 Adam 所能達到的準確率上限，跟其他幾個 optimizer (SGD, AdaGrad, RMSProp, NAdam……) 比起來，到底孰高孰低？還是能到達的上限其實相同，只差在訓練時間？原本使用RMSprop，或是Adam的地方，改用Nadam大多可以得到比較好的結果，這是真的嗎？
2. 在 Adam optimization 中，常用的套件 (ex. Keras）default 設定第一個動量常數 beta1 = 0.9，第二個 beta2 = 0.999，到底這個常數是不是在這兩個值會表現最佳呢？
Ref: https://keras.io/optimizers/ (https://keras.io/optimizers/)
3. 在train LSTM模型的時候，我們可能會用多種不同的optimizer，其中像是RMSProp、Adam更是常被使用。作業一的時候大家甚至可能使用的是最原始的GradientDescent，究竟這些optimizer哪一個比較適合用來train LSTM呢?有一個都市傳說似乎是GradientDescent是最好的？究竟是為什麼呢？
4. 其實不論Adam, SGD 或是 RMSprop，我們其實都是必須要根據特定的task才能決定說要用哪一個optimizer。 到底有沒有一個判斷的方法或是準則，甚至有沒有辦法能夠設計出一個model，可以讓我們簡單的判斷要使用哪一個optimizer才可以做到最好的效果呢？
5. Binary Network 或 Ternary Network真的適合gradient based的優化器嗎？
6. 深度學習中的Gradient descent的方法可以用其他種方法取代嗎？例如OpenAI最近提出的Evolution strategies (https://blog.openai.com/evolution-strategies)用來解reinforcement learning就有很多資深學者抱有反對意見。
7. optimizer可不可以學習，能不能夠再訓練model的時候同時訓練optimizer讓它能夠更好的去決定梯度同時達到joint to joint的訓練？
[Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." Advances in Neural Information Processing Systems. 2016.]
8. WGAN 提到訓練 GAN 時，不要使用基於 momentum 的 optimizer (例如 Adam)，這在不同的 task 都成立嗎？一定要用rmsprop嗎?
9. Bengio 對SGD的initialization等等做出了改良 (http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf)，並在許多benchmark 上得到比以原本SGD with random initialization 還要significant 的結果。並指出Gadient Descent 在原本使用上的許多問題與限制；而現在來看，雖然Adam, RMSprop 等等optimizer被提出來使用並指出許多情境比用GD有更好的結果，但因應deep learning 模型的GD可以得到這個顯著的改善，若其他optimizer可以依據其特性與問題，是否也能獲得顯著改善? 是否有一個適用任何deep learning model 的 optimizer存在呢?
10. Hung Yi曾說：「不能gradient descent的(就是discrete reward的)，就用RL硬train一發。」這樣做真的好嗎？是否有別的方式讓他可以作gradient descent且效果不比RL差太多？
11. adam , adagram , sgd , adadelta 等等的 optimizer彼此間的異同之處？做 RNN 、GAN 等等的使用哪一個 optimizer 有甚麼限制？
12. 為了避免鞍點的問題，可否在 train 幾個 epoch ， 發現下降不快的時候更改所使用的 optimizer ?

Activation and Initialization

1. relu 或其變形 (如：elu) 是今日常用的激活函數。但它們真的帶來比較好的效果嗎？是讓訓練變得更容易嗎？還是讓模型比較不容易 overfit 呢？
2. relu和sigmoid在深度學習中都是常用的激活函數。而sigmoid最常被提到的問題，就是gradient vanish。不過和relu相比，sigmoid在feature transform的能力應該較強。如果今天能夠讓每一層network有不同的learning rate，以解決gradient vanish的問題，是否一率使用sigmoid作為激活函數會有比較好的效果？
3. Dropout 搭配較接近線性的激活函數 (例如：relu, maxout 等) 效果特別好。
4. Dropout總是只用於training，假如testing時也用效果會如何？可以試試不同的keep_prob。
5. 聽說使用Xavier Initialization通常得到的performance會比較好，原因是因為xavier initialization可以避免在經過神經網路層的時候造成gradient vanishing跟gradient exploding的問題。 然而事實上使用Xavier initialization真的會有差嗎？ 還是就算是random initialization，只要training的次數夠多次，根本就不會有影響呢？ Ref:
http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf (http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf)
6. train GAN的時候optimize不能用relu要用LeakyRelu
7. 聽說在初始化一個神經網路的weights時，最好要落在(-1/√d, 1/√d)之間，d是一個神經元的input數量，這是真的嗎?
8. Tensorflow裡面內建了許多種initialization的方式，例如uniform、normal等等，又在各種方式之中所要設定參數，該如何設定才有比較好的performance呢？
9. dropout在CNN是很常見regularization方法，那在RNN中什麼樣的情況要使用dropout、又要在哪裡dropout才能使得testing performance更好
[Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." arXiv preprint arXiv:1409.2329 (2014).]
[Gal, Yarin, and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks." Advances in Neural Information Processing Systems. 2016.]
10. 為什麼在 train 的時候做 Drop out 比較好？是不是意味著 training data 和 testing data 的分布不一樣？還是說有幾筆資料根本是特例中的特例，直接捨棄掉更好呢？如果是的話，與其用隨機 drop out ，有沒有辦法直接把不要的資料找出來捨棄掉，會不會更好呢?
11. 在 classification 的問題中，我們利用 Neural network 的架構，最後為每種類別算出一個數值，並且需要一個可以將這些數值轉換為機率之類的 function。假設一個 2 種類別的 classification 問題，則上面提到將這些數值轉換為機率的 function，使用 softmax function、sigmoid function，或是其他 function，是否有哪一種的產生的分類效果會比較好？
12. 在RNN中，一般皆使用tanh作為內部的activation function。因為relu在正值unbounded的特性是LSTM不希望的，然而，我們知道relu較沒有saturation 的問題，故有人提出使用relu作為RNN的內部參數(IRNN) (https://arxiv.org/pdf/1504.00941.pdf)，然而這需要用identity matrix做weight initialization。事實上也有人用Xavier Initialization配上elu來實作RNN，請究LSTM inner activation觀察其對於gradient flow產生的影響。
13. 似乎沒聽說過有人在多層RNN中兩層hidden layer的中間加入activation function的，為何？如果加了會對training產生什麼影響呢？
14. 為何LSTM的gate cell大部分都用sigmoid function，而不是用其他函數，如relu？

Training Process/Tips

1. test feature用1-hot encoding來轉成數值feature真的比較好嗎？不能直接用hash function直接轉成不同數值嗎？
2. batch size 越大是否會影響performance，如果batch size設1是否可以得到最好的結果？
3. 一般認為參數越多越容易 overfitting，但在 hw1 實際測試後發現，儘管參數多的 model，training 和 validation 的 loss 在訓練後期overfitting 較嚴重（兩者的 loss 差較多），然而，參數多的 model 卻在較低的 loss 情況下才開始 overfitting，之後在 testing 上也得到較好的成果，因此，或許在 overfitting 嚴重的情況下，仍能嘗試提高參數量？
4. 其他的pooling到哪裡去了？
大部分的人都會用max pooling，還有一些人會用average pooling。但是還有很多種pooling的方式，例如min pooling，應該跟max pooling效果差不多吧？有沒有其他pooling的用法？例如說在什麼題目下該用哪些pooling方式？
5. Stochastic pooling據說要和Dropout合用才能有較好的效果？
6. 所有的神經網路只要加了batch normalization就能解決每個batch資料分布不同造成的梯度競爭問題，並加速收斂嗎？
7. 傳說在做cross-domain learning的時候，例如：Siamese network, triplet network，在其中一邊多加上一層fully-connected layer 也就是 " adaption layer " 會使得兩邊的domain分布比較相近，因此得到比較好的performance。這是真的嗎?
[1] Deep Exemplar 2D-3D Detection by Adapting from Real to Rendered Views (https://arxiv.org/pdf/1512.02497.pdf)
(https://arxiv.org/pdf/1512.02497.pdf)[2] Deep Domain Confusion: Maximizing for Domain Invariance (https://arxiv.org/pdf/1412.3474.pdf)
8. Multi-GPU 理論上能加速學習，但好像不是總是如此。是不是有某些 models 或資料型別不適合利用 multi-CPU 呢？

9. 作者在論文中提到：借鏡神經演化的概念，衍生出 Evolution strategy 來取代 back propagation 做 reinforcement learning，雖然表現變差但訓練時間縮短非常多，有可能成為新興最佳化模型的方式。是不是我們長期使用的 Back propagation 有可能也是一種迷思？(https://blog.openai.com/evolution-strategies/ (https://blog.openai.com/evolution-strategies/))
10. 在每個 epoch 開始時 shuffle data 被認為可以加速收斂，但是 performance 呢？shuffle data 會不會影響到 performance 呢？
11. 假設你知道 testing data 的 distribution (例如說你知道某個分類問題中每一個類別在 testing data 的機率)，那通常在切 validation data 時，也會照著這個 distribution 來組成 validation data。那在 training 的時候，我們是否也能按照這個 distribution 來組成每一個 batch，用跟 testing data 有著相同 distribution 的 batch 來 train，這樣 train 出來的 model 是否更為 robust？
12. 用batch_normalization真的會讓效果比較好嗎？
13. 在面對imbalanced的data時以往的做法通常是調整sample的數量、調整class的權重或是用self-labeling的方式調整training data。然而這些都是heuristic的做法，有更具有理論保證的作法來處理imbalanced的data嗎？
14. 如果說dropout可以用來預防overfitting，那inverse dropout（加noise）對預防overfitting有沒有幫助呢？
15. Dropout通常是random捨掉connection。使用random真的比較好媽？有沒有一個比較好的挑法更容易prevent overfit？Ref: https://arxiv.org/pdf/1506.02626.pdf (https://arxiv.org/pdf/1506.02626.pdf)
16. 理論上Dropout在convolution layer和fully-connected layer都能使用。在convolution layer使用dropout也能在一定程度上提升testing accuracy，只是epoch要夠多？
17. Lutz Prechelt做了一連串的實驗，顯示各種 “Early stopping” 的方法在performance方面的確是有些許的幫助，然而卻得用大量的training time作為代價。 而這個結論是只有在他所做的fully connected NN上，還是在其他架構的 neural networks 上也是亦然呢? (Reference paper : Prechelt, Lutz. “Early stopping—but when?.” Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, 2012. 53-67.)
18. Batch Normalization是一個穩定模型的很好的架構。在原BN的paper中提到在進入activation function前的數值若做normalization可以得到比較好的結果，在DCGAN的架構中BN也是很重要的架構，如果沒有BN DCGAN的模型便無法train到好的結果。然而在activation func.前做normalization真的是個正確的做法嗎？想像今天在進入Relu前做了normalization，而這有正有負的值經過Relu後負值都被去掉，剩下的正值mean便不會在0的地方，同時也失去了normalization的意義了。所以該在activation function前或是後做Batch Normalization呢？
19. WGAN的discriminator的weight一定要經過clip才能train的起來嗎?
20. clip weighting 有效提升了 WGAN的表現，如果把這個技術應用在其他的 model 上，也能有效提升表現嗎？還是會使效率反而變低落呢?
21. Training data 的順序對結果的影響大嗎？如果把性質相近的training data排在一起丟下去train，performance會不會比較好？
22. NN參數越多的時候通常需要越多的training data。能不能從Training set的大小來估計讓他performance比較好的參數數量(ex number of layers, training epochs,…)?
23. Batch size有沒有可能在training 途中改變？有沒有可能影響performance或者training time?
24. 訓練中發現 Validation 下降趨緩時的策略往往是 early stopping。 試問：若Validation 卡住時自動切換各種不同的 Optimizer 繼續硬 train，是否有機會使 Validation 重新開始下降、最後train出更好的結果？
25. 「越deep越好」？加入一個dense之後，雖然有時反而會在test上overfit，但理論上應該可以fit到更複查的函數，最慘也能讓weight逼近一個Identity matrix才是。有時候加了一個dense卻是在train階段直接爆炸，為什麼？
26. 在GAN訓練到一定程度之後，generator已經能有一定程度的生成能力，但discriminator卻可能因為訓練資料有限(可能標記是昂貴的)而無法再改進，或無法改善特定部分(ex:生成出來的圖某部分特別差)。因此，是否在訓練的後期納入active learning，能達到有效率且有目標性的改善？
27. 在DCGAN原本的paper中，使用filter size 5 * 5而不能整除64，使用4 * 4能免除padding的問題，是否能有更佳的結果呢？且其為了穩定output，並沒有在output layer和descriminater的input layer加上 batch normalization，為何這會影響training的穩定性？
28. 關於GAN中noise的distribution的不同會對training或inference造成什麼影響嗎？
29. 把GAN最後一層去掉sigmoid，loss不取log，不使用momentum和Adam改使用RMSProp，就可以變好？(Ref:https://arxiv.org/abs/1701.07875 (https://arxiv.org/abs/1701.07875))
30. 根據Google 工程師表示，要把一個NN model train起來，data量與hidden unit的數量比至少要10:1，少於這個比例基本上是train不起來的，這是否是謠言呢？
31. Weight normalization或者updating weight by normalization是否有助於加速training或者提升accuracy?
32. 做 mini-batch 一定能提升效能嗎？會不會反而使得 gradient 被侷限在某一個鞍點而出不來呢?

Application

1. RNN vs CNN 的抽出的sentence representation孰強孰弱? 在做句子的分類上兩種model都能有效的去模擬出句子的表示向量，但使用哪種model所做出的向量能更好的表示這句話的意思呢？
2. character word embedding真的有用嗎？ 在做text modeling的時候許多model都會將word embedding 和 character word embedding 去做concat，來當作這個詞的表示向量，來增強對於一個詞的表示力，但這樣的方法是否真的能有效的提升performance，還是只是徒增計算量而已呢？
3. 使用 word embedding 時，我們經常會訂定一個 vocabulary size，然後將剩下的單字設為 。vocabulary size 太大會導致 training 困難，而太小則會導致 performance 不佳。我們有沒有辦法依據不同的情況，找到最佳的 vocabulary size 呢？
4. 如同在MLDS HW1當中，考慮在train一個RNN-based word level Languange Model(LM)時，我們可以使用不同的方式來表示不同的word。最簡單的方式就是用單純的1-hot encoding來表示每個word，但是word vector的長度很顯然會隨著vocabulary size而成長。因此若使用網路上一些現成的word-embedding model，例如GloVe、Google word2vec，是否能在word-level LM這個問題上取得比較好的成果呢？(Hint：在RNN input/output使用各有什麼效果？若只是把pre-trained word embedding做為RNN的initialization又有什麼效果?)
5. Which is better for text classification: CNN or RNN?
   Ref :
   Disscussion on Quora (https://www.quora.com/Which-is-better-for-text-classification-CNN-or-RNN)
   Supervised Sequence Labelling with Recurrent Neural Networks (http://www.cs.toronto.edu/~graves/preprint.pdf)
   Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding (https://arxiv.org/abs/1504.01255)
   Comparative Study of CNN and RNN for Natural Language Processing (https://arxiv.org/pdf/1702.01923.pdf)
6. 用RNN來做文字的sequence learning時，通常不是直接拿前個時間的機率輸出當成下個時間的輸入，而是會先經過sampling或取argmax。老師在上課時提到這可以避免「高興想哭」或「難過想笑」的出現。但這樣訓練方法是否有其他缺點？例如收斂較慢等。一些折衷的方式(例如取機率top k)會不會比較好呢？
   Reference: https://arxiv.org/pdf/1511.06732.pdf (https://arxiv.org/pdf/1511.06732.pdf)
7. MLDS HW2的Video2Caption模型可以用scheduled sampling在traning時模擬inference的行為，理想上可以讓模型避免exposure bias。然而在文獻中(How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary? (https://arxiv.org/abs/1511.05101))指出SS是一個不穩定的作法。所以Scheduled Sampling真的是合理的嗎？
8. HW2的產生影片敘述，使用單層的RNN暴力算下去是否能產生同等甚至更好的結果？
9. Language model 的好壞是否跟 perplexity 有絕對的關係？
10. 通常在將詞彙從1-hot representation轉成embedding時，我們會透過一個embedding matrix，也就是額外一層hidden layer來完成。如果今天嘗試使用兩層甚至兩層以上的hidden layers，把embedding投影到更深的維度去，是不是能有更好的表現？
11. 在影像方面的研究常採用CNN來解決，但因為不是每個問題都有夠多的Training data，所以常見做法是拿image classification的CNN(例如Alexnet、Googlenet、Resnet)做為pre-train的model，拔掉最後幾層layer加上自己的layer(通常是fully connected layer(fc))，繼續做gradient descent。而傳聞中，如果自己加上的不只是fc layer，而是fc+convolution layer，會取得更好的結果。這是真的嗎？會不會甚至只加上convolution layer，得到更好的效果呢？
12. 在做影像分類時，早期的 CNN model 都會在最後加上幾層 fully-connected layer。但綜觀整個 CNN，大半參數都集中在 fully-connected layer 中，因此有人提出了 fully convolutional network 的概念 (https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf)，也在之後的影像辨識比賽中取得很好的結果。當初使用 fully-connected layer 的目的除了可以直觀的把 layer size 匹配到 class number 之外，它還有其他實際的作用嗎？
13. 在目前的影像辨識任務中，檢測的目標都在整張影像中蠻大的比例，也獲得不錯成果(Faster RCNN, YOLO, SSD, MSCNN等)，若目標佔影像中的比例很小，目前的方法可否獲得差不多的結果呢？
14. 在傳統的影像辨識上，由於通常一張影像並不會只有一種物件(例如狗的照片中也會有草地、天空等等)，所以有人認為若是對單一影像加上複數標籤(multi-labels)進行訓練，則可能獲得比單一標籤(single-label)更好的結果，但由於複數標籤在訓練上比單一標籤來的複雜且困難，相關研究數量尚且不足，因此我們想知道，在CNN的影像辨識中，複數標籤是否真能獲得比單一標籤更好的結果？
15. CNN 中的 pooling 有很多種，在做影像辨識時，有人說用 max pooling 比較好，因為 max pooling 可以強化 filter 覆蓋到的部分中，比較明顯的特徵，請問這是正確的嗎？
16. 在影像上，我們多半使用CNN。但CNN基本上就只是拔掉一些weight的DNN。這兩個model的真實效能優缺比是？ CNN會是影像ML的最佳解嗎？
17. 人們常說CNN Image Classifier分類不受圖片平移影響，是因為多了convolutional layer, max-pooling layer,共享權重。然而，位於左上角與右下角的標的，在進入fully connected layer之前，壓平的向量是截然不同的（一個是前幾維被激活，另一個則是最後幾維被激活）。這樣子為什麼會說它不受平移影響？
18. 在 VGG-16 的論文中提到，3×3 大小的 filter 會比 5×5 大小的 filter 好，因為你用兩個 3×3 的 filter 就可以超過一個 5×5 filter 所能覆蓋的面積。兩個 3×3 filter 的參數要少於一個 5×5 filter，因此比較不容易 overfitting，請問真的是這樣嗎？
19. openset會是一個影像辨識需要克服的問題，也就是model不但要能夠辨識學習過的類別，當遇到未知的類別時還需要去特別處理，最naive的作法是增加第n+1類代表”unknown”，這樣是不是就能夠很好的去處理呢？還是有其他作法？
    [Bendale, Abhijit, and Terrance E. Boult. “Towards open set deep networks.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.]
20. 使用deep CNN的方法來解決影像的問題，例如object detection、image classification、training data的數量多一定會有比較好的training結果嗎？training data整體的品質會不會比data數量重要？例如使用data augmentation的方式由現有的data產生額外的data，但經過data augmentation後的data set，一定會有比較好的training結果嗎？
    在解決影像問題的時候，使用比較deep的CNN架構一定會有比較好的結果嗎？
21. 大家常常說 NN 裡面就是一個 black box，調了半天參數也不知道為什麼 performance 會變好或變差，除了找到 activate 各個 layer 的圖片之外，有沒有一個比較好的方式是可以讓我們了解 input 一個 image 後每一層的作用為何？
22. 在老師上課的範例中，用GAN來轉換圖片特定的資訊時，有把麥克風當成是頭髮，也有把馬背上的人一起變成斑馬。這是否代表說我們只是自以為機器學到了怎麼辨認，但實際上對於機器來說也許所學到的東西跟我們想像的完全不一樣，不論是特徵或是形狀，甚至不同物體之間的差別也和人類辨識的不一樣呢？
23. CNN 是根據人類辨識圖像的特徵而發展出來的。但是自然界中能辨識圖像的生物不只有人類，那其他動物辨識圖像的方法跟人類一樣嗎？如果不一樣，可以根據其他動物辨識的方法建立布一樣的NN嗎?
24. 用VGG抽Feature的優劣?
25. 如果在圖像中加入了隨機的雜訊，使人眼判讀圖象依舊沒問題，但是對機器來說卻會造成辨識的障礙。這是否代表機器辨識圖像的方式並不如我們設計時的預想呢? 有辦法解決這個問題嗎？
26. 在做分類問題的時候，通常在最後算loss之前會先用 softmax 把每一個維度的值先做調整，但是不用softmax他本身的值也會有大小關係，會不會做不做softmax其實沒什麼差，或是使用其他function，是否有哪一種的產生的分類效果會比較好?
27. Leetcode的題目可以用deep learning解。（例如這題 (https://leetcode.com/problems/fizz-buzz/#/description)）
28. 在語意分析的問題上，一些LSTM2的變形：例不同種類的Tree LSTM被提出，請問它們得到的sentence feature vector，較傳統sequence-to-sequence model得到的sentence representation的性質差異為何?
29. 現在的語音辨識是否可以不再經過傳統方式一連串的處理,直接用深度學習的方法從頻譜上直接硬train得到好的結果？

30. 使用 attentional mechanism 來解 video cation 問題時，究竟 attentional mechanism 真的有發揮作用嗎？或是只學到了 language model，而 attention 只對於決定句子的起點有較大的影響？REF: https://arxiv.org/abs/1508.04025 (https://arxiv.org/abs/1508.04025)

Others

1. 現在有許多Deep Learning的框架(Torch, MxNET, Tensorflow, Caffe)，其中Tensorflow的使用社群最為龐大。然而Tensorflow在建置模型的彈性不如Torch(無法動態建圖)，近期更有實驗比較在極端的狀況下Tensorflow會比MxNET的慢上五倍。難道Tensorflow過譽了嗎？(source: https://github.com/tensorflow/tensorflow/issues/7187 (https://github.com/tensorflow/tensorflow/issues/7187))