Project Discovery Guide and Example Project Lists

1. Preface

This document contains guidelines and ideas for designing a semester 2 project for the MLP course. The listed ideas are non-exhaustive, and in no way do they represent the space of possible projects. They are merely a tool provided to you, so you can create your own project, by simply combining the provided tips and ideas.

The *difficulty* of a project, can be hard to define and/or measure, mainly because it is relative to the context of a particular project (e.g. some are difficult to built and easy to mentally come up with, while others can be easy to build and hard to come up with), the competency and experience of the people undertaking it, as well as the actual resources available for the project (e.g. in terms of compute, financial funding etc.).

For this reason, we decided to introduce the abstract notion of the *challenge level*. The challenge level is a measuring tool attempting to quantify projects in terms of how an average MLP student would perceive the difficulty of a project once it has engaged with the said project. This tool was created by the MLP courses instructors experience and intuition, and should not be taken as an objective measure of difficulty, but rather as a signal source to be used to select a project.

2. How to use this guide

1. Read through the types of suggested projects.
2. Choose 3 projects/topics that interest you.
3. Read the associated papers/resources if any.
4. Isolate to 2 topics/projects.
5. Discuss with your appointed tutor or course TA.
6. Design a project with your tutor or the TA. Make sure to clearly define backup plans and contingencies for high-risk ideas/experiments. This will help with risk management and minimization.
7. Start working on your project.
8. Meet with your tutor every week to report your progress and receive advice and guidance on how to proceed. **Note**: If additional help is needed, simply drop by MLP-Base sessions, which take place every day from Monday to Friday at 1700-1800 at A.T. 7.03

Note: If you need help with designing or evaluating the difficulty/feasibility of a project, please contact your appointed tutor or the MLP TA (`a.antoniou@sms.ed.ac.uk`).

3. Types of projects

Each project type is not a discrete variable, but rather a continuous variable. Projects can start as dataset-driven and evolve into method-driven or even innovation-driven projects, or the other way around. Sometimes to innovate you need to build a new setting and a new dataset.

1. Dataset-driven (The challenge level starts from 3/10 for this type): Choose an interesting dataset (e.g. Painters By Numbers, CelebA, Mini-Imagenet, CIFAR10/100 etc.) Come up with interesting tasks to use the data on (e.g. Classification, Generation (GANs), Translation (Dogs to Cats, English to Japanese), etc.). For a more detailed list of methods see below.
   (a) Apply method to the dataset, observe the performance of the system.
   (b) Investigate the method architecture/setup and attempt to improve the performance of the baseline model. (i.e. improve classification accuracy, improve sample quality/relevance or some kind of generation loss, improve cross entropy loss etc.). This can be done via exploration of architectures, data augmentation strategies, loss functions or reqularization methods.
   (c) Once you have explored and improved the performance of the system, you can also try to add your own novel feature on top of it, to push the novelty of your work further.
2. Method-driven (The challenge level starts from 5/10 for this type):
   (a) Choose an interesting method/task/learning paradigm. For your convenience, we include a list of the topics/areas that a lot of experts consider the ones with the most potential:
      i. GANs [11, 12, 38, 19, 2]

ii. Reinforcement Learning [6, 5, 4, 28, 34, 14]
iii. Meta Learning [10, 1, 36, 35, 24, 3, 23]
iv. Relational Reasoning [27]
v. Genetic Algorithms for Deep Learning [24, 26]
vi. Classification via Deep Learning [13, 18, 24, 20]
vii. Adversarial Attacks/Defences [31]; https://github.com/yenchenlin/awesome-adversarial-machine-learning
viii. Transfer Learning [15, 37, 21]
ix. Domain Adaptation [32]
x. Life-long Learning [22]
xi. Network Visualization [29, 30]
xii. Network Compression [17] (See lecture 15)
xiii. Multi-task Learning [7, 25]
xiv. Sequence-to-sequence models [9, 16]

(b) Read that topics/methods associated papers.
(c) Continue your literature review towards the directions to seem interesting. (Read some more papers in that direction) Try to come up with an idea for a project based on what you have read, your own interests and your own subjective view of "interestingness" and "fun". If something gets you excited or intrigued, it is probably something you will, at the very least, enjoy working on.
(d) Take your topic and/or ideas to your tutor or the MLP TA or the daily MLP-base, to help you design a project.
(e) Start your project and follow the general project guidelines showed in section 2

3. Concept-driven/Innovation-driven (The challenge level starts from 8/10 for this type): Projects under this type usually attempt to implement a new concept or to improve existing methods with out of the box ideas. People who choose these kinds of projects, have probably read a substantial amount of papers, and have also coded their own models in the past, enough for them to have creative ideas and good intuition in their chosen task.
   (a) Read many papers from a chosen topic
   (b) Implement favourite methods from scratch, to get very strong intuitions on how the techniques behave.
   (c) Combine all you have learned and either try to improve an existing method or come up with a brand new concept that attempts to solve a particular problem, or even, come up with a novel task/problem, and try to solve said tasks.
   (d) Meet with your tutor or MLP TA to design a project and to make sure the project is feasible.
   (e) Meet weekly with your tutor to report progress.

4. List of projects

4.1. Dataset-driven:

1. Train classifier to predict the artist that draw a particular painting, given the painting itself. Use paintings by numbers as your dataset.

2. Train generative models that can do cool things with paintings, perhaps training a CycleGAN, to translate paintings between styles or artists. I.e. make a Picasso into a matisse

3. Use CIFAR10/100, train existing SOTA models, and then investigate novel data augmentation strategies or novel architectures to improve current SOTA.

4. Use Omniglot, and using only 5 samples frome each class attempt to build a classifier that has strong generalization on the left over samples.

5. Explore approaches to emotion recognition using the IEMOCAP dataset [8]

6. Audio event detection using one of the datasets used in the 2018 DCASE Challenge (http://dcase.community/challenge2018/)

7. For regression problems explore using the discretised softmax regression approach used in wavenet [33]

4.2. Method-driven:

1. GANs:
   (a) Use a Cycle GAN and learn to translate cats into dogs, or people into pokemon or cats into people?

2. Reinfocement Learning:
   (a) Use the PPO RL algorithm in a novel OpenAI GYM environment and tune the deep net architecture to improve performance.

3. Meta-Learning:
   (a) Use existing MAML++ code, but train outer loop using genetic algorithms instead of SGD.
   (b) Exploring the effect of architectures on the performance of MAML/MAML++ models.

4. Relational Reasoning:
   (a) Use relational reasoning in a novel task, such as inside a GAN to improve relational capabilities of the discriminator and generator. Compare and contrast sample quality.

(b) Improve on existing relational reasoning methods, via *Hyper-Relational Networks*, this project was a project that Antreas Antoniou wanted to try in the future, but would be willing to instead give to a student. Contact him for more information.

5. Genetic Algorithms:
   (a) Replace SGD with G.A. in classification by deep neural networks, evaluate and compare with SGD.
   (b) Use G.A to tune reqularization hyper-parameters (dropout, l1, l2 etc.). Or, use G.A to learn the best augmentation strategy.
   (c) Use G.A to tune Adam.

6. Transfer learning:
   (a) analyse different regularization strategies for fine-tuning (See lecture 14).
   (b) analyse influence of unbalanced number of images per category and test different strategies to fix it.
   (c) given a set of pretrained networks on different datasets and find the best option for a target task.

7. Network compression:
   (a) Analyse the use of different compression techniques such as group convolutions and minimize number of parameters while retaining good performance on a dataset (See lecture 15).
   (b) Use network distillation to transfer knowledge of a big teacher network to a small student network (See lecture 15).
   (c) Take a pretrained network and prune some of its filters with a minimum drop in its performance (See lecture 15).

8. Life-long learning:
   (a) Train a network on one task first and then fine-tune another one. How can we ensure that the network does not forget about the first dataset?

9. Network design:
   (a) Analyse influence of various design choices (pooling, dilated convolutions, etc) on multiple datasets and analyse whether a design decision on one dataset generalizes to another one.

10. Multi-task learning:
    (a) Train a single network on multiple datasets jointly. Design a network that shares most of the layers across different tasks, find the best layer sharing configuration.
    (b) Train a network on multiple loss functions and design a weighting strategy to balance the loss functions.

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[3] Irwan Bello, Barret Zoph, Vijay Vasudevan, and Quoc V Le. Neural optimizer search with reinforcement learning. *arXiv preprint arXiv:1709.07417*, 2017.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[5] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[6] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[7] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257*, 2017.

[8] Jaejin Cho, Raghavendra Pappagari, Purva Kulkarni, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Deep neural networks for emotion recognition combining audio and transcripts. *Proc. Interspeech 2018*, pages 247–251, 2018.

[9] K. Cho, A. Courville, and Y. Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved

training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[16] T. Hori, H. Wang, C. Hori, S. Watanabe, B. Harsham, J. L. Roux, J. R. Hershey, Y. Koji, Y. Jing, Z. Zhu, and T. Aikawa. Dialog state tracking with attention-based sequence-to-sequence learning. In *IEEE SLT*, 2016.

[17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks.

[20] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

[21] Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. *arXiv preprint arXiv:1802.01483*, 2018.

[22] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[23] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[24] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. *arXiv preprint arXiv:1802.01548*, 2018.

[25] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

[26] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[27] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[29] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[30] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[31] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). *arXiv preprint arXiv:1805.12152*, 2018.

[32] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[33] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[34] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, volume 2, page 5. Phoenix, AZ, 2016.

[35] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[36] Olga Wichrowska, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. *arXiv preprint arXiv:1703.04813*, 2017.

[37] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.