

Math 514

Project Proposal

April 3, 2019

Project Proposal

Project proposals are due Wednesday, April 10. Project teams will give a short presentation during class on April 24. A set of slide templates using the Xaringan environment will be provided to help with preparation. Please make an effort to share what you have learned about your topic with the class. The final project is due by midnight May 5. The final deliverable should use reproducible research methods and must run successfully.

1. Identify team members. Include name and email
 2. Provide a problem statement and a description of what you hope to achieve. If you are starting with a project/model found online please provide a link and describe what you will do to add to the work or experiment with.
 3. Give a description of the data set
-

Project Ideas

Project topics are up to each team. For teams that are undecided, here are some notes you may or may not find useful

I mentioned several papers during the semester which I felt were interesting and might form the basis for a study. These reflect my interest in the fundamental behavior of neural networks. Most of them are focused on understanding what makes a model generalize. Be warned that these papers work with large data sets and large models. You would need to imitate the studies but on smaller data and with smaller models. The mathematics can be involved. The suggestion is to conduct an empirical study along the lines of the reference.

1. For part of next week's lecture I will cover the paper Visualizing the Loss Landscape of Neural Nets by Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer and Tom Goldstein. Projects could use the methods in this paper to look at how loss surfaces are affected by the many hyperparameters of a neural network.
2. One idea in the literature is that good solutions are associated with loss function local minima that are flat. This idea is not universally accepted. A good paper that argues that small batch gradient descent results in flatter minima than large batch is On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima by Keskar, Mudigere, Nocedal, Smelyanskiy and Tang.
3. Another idea is that large over-parameterized models generalize better than small models. The following paper uses model sensitivity to the input data as a measure of the model's ability to generalize. Sensitivity and Generalization in Neural Networks: an Empirical Study by Novak, Bahri, Abolafia, Pennington and Sohl-Dickstein
4. This paper argues that training with dropout has no isolated local minima. Work comparing the regularization effect of dropout with L1/L2 regularization would be very interesting. See Surprising properties of dropout in deep networks by Helmbold and Long
5. The paper Skip Connections Eliminate Singularities by Orham and Pitkow argues that using skip connections (not covered in lectures) dramatically improve the loss function surface and speed up learning. Visualizing the effect of skip connections might be interesting

Here are some less academic ideas -

6. Build a model in a domain of interest - Natural language processing, visions or speech
 7. Conduct a study of some aspect of neural networks that interests you
 - regularization
 - batch normalization
 - optimizers
 - learning rate schedules
 - activation functions
 - network architectures - dropout, skip connections
 - visualization
 8. Write code for a new type of network not covered in class. Some of these can be done using Keras, some can't
 - Hopfield network (probably need to hand code but relatively easy)
 - Restricted Boltzmann machine. (Challenging)
 - Probabilistic neural network
 - Autoencoder (probably in Keras)
-

TD-Gammon

I will provide R-code that computes all possible moves given the board state and dice values. I believe my board layout comes from the following paper Using Machine Learning to Teach a Computer to Play Backgammon by Molin, Flyhammar and Bidgol

Here's a popular post on TD-Gammon: Before AlphaGo there was TD-Gammon by Jim Fleming

Here's a tensorflow implementation of TD-Gammon: GitHub TD-Gammon by Jim Fleming

Here's a link to the original article by Tesauro: Temporal Difference Learning and TD-Gammon by Gerald Tesauro

R projects online with Keras: <https://tensorflow.rstudio.com/examples/>