Review Guidelines

Stated summary and decisions. What problem are the authors trying to solve? What are the limitations on this problem, i.e., what are they not trying to solve? What techniques or tools do the authors offer to solve the problem at hand? How do the authors know they have solved the problem? Do the authors test or validate their approach experimentally? Does the solution meet the stated goals, or does it fall short in some way? Avoid simply quoting the authors' own abstract. Restating in your own words demonstrates your understanding.

1. **Significance.** What is new here? What are the main contributions of the paper? What did you find most interesting? Is this whole paper just a one-off clever trick or are there fundamental ideas here which could be reused in other contexts?

2. **Fallacies and blind spots.** Did the authors make any assumptions or disregard any issues that make their approach less appealing? Are there any theoretical problems, practical difficulties, implementation complexities, overlooked influences of evolving technology, and so on? Do you expect the technique to be more or less useful in the future? What kind of situation would defeat this approach, and are those scenarios important in practice?

3. We are not interested in spelling errors when reviewing a paper. However, if you have a great idea on how some concept could be presented or formalized better, mention it.

4. **Contribution.** How could the paper be improved? How could some of the flaws of the paper be corrected or avoided? Also, how does this paper contribute to the current literature? Incremental or fundamental? Are there similarities between this approach and other work, or differences that highlight important facets of both?

Please take the time to edit your reviews.

Some demo (a mixture of good and bad reviews):

https://openreview.net/group?id=ICLR.cc/2020/Conference

https://papers.nips.cc

Writing practice:

1. This talk explains why we can find a robust estimate in the infinite-sample corruption model theoretically and computationally.

(We can find a robust estimate in the infinite-sample corruption model theoretically and computationally.)

==> This talk explains the theoretical and computational feasibility of robust estimation in the finite-sample corruption model.

2. A gradient descent method that ignores the constrain is showed to find the KKT points efficiently, though the algorithm is not universally guaranteed in any case.

(In specific/certain cases, an unconstrained GD method can find the KKT…)

==> A non-constrained gradient descent method finds the KKT points efficiently, although the universal guarantee of the algorithm remains open.

3. The Adjoint method is a way to compute the gradient of objective function via corresponding Lagrangian efficiently.

==> The Adjoint method efficiently computes the gradient of objective function via Lagrangian.

4. If we have a super process to estimate the parameters of the network, no information will be lost.

==> A super process estimates the parameter of the network without loss of the information.

(No information is lost if we have a super process to estimate the network parameter.)

5. The speaker introduces a co-adjoint method to address the gradient vanishing, where a penalty term of vanishing Lagrange multipliers, lambda_t, is added to the objective function.

==> The speaker introduces a co-adjoint method with penalized objective function to address the gradient vanish.

6. We prove Q is positive semi definite matrix which implies that quadratic programming is valid in (2).

==> The semidefinite positivity of matrix Q implies the validation of the quadratic programming in (2).
(We prove the positive semi-definiteness of the matrix which ensures the validness of the quadratic programming in (2).)

7. This shows that two methods are pretty much the same.
==> This shows the similar performance between the two methods.