Project Proposal for Deep Learning 2017

March, 2017

Due date

- Submit your team info, project title, and abstract to the course wiki by the class of the week after spring break (March 23)
    - 5% of the total course (as part of the activities)
    - Recall: activities (30%), programming assignment (30%), final project (40%)

Class Project

Ideas borrowed from Richard Socher (Stanford CS224d)

- Most important (40%) and lasting result
- Start early and clearly define your task and dataset

Project types:

1. Apply existing neural network model to a new problem
2. Implement a complex architecture for old problem
3. Come up with a new neural network model
4. Theory

Procedures for Types 1-3

1. Define Task
    - Example: summarization of text
2. Define Dataset
    1. Search for academic dataset
        - They already have baselines
        - E.g., Document Understanding Conference (DUC)
    2. Define your own (harder, need more new baselines)
        - Try to connect to your research
        - E.g., For summarization, can use Wikipedia: intro paragraph and the rest of the article
        - Be creative, can try to look into blogs and news also
3. Define your metric
    - Search online for well established metrics on your task
    - E.g., for summarization, Rogue (Recall-oriented understudy for gisting evaluation) defines n-gram overlap to human summaries

Procedures for Types 1-3

1. Define Task
   - Example: summarization of text
2. Define Dataset
   1. Search for academic dataset
      - They already have baselines
      - E.g., Document Understanding Conference (DUC)
   2. Define your own (harder, need more new baselines)
      - Try to connect to your research
      - E.g., For summarization, can use Wikipedia: intro paragraph and the rest of the article
      - Be creative, can try to look into blogs and news also
3. Define your metric
   - Search online for well established metrics on your task
   - E.g., for summarization, Rogue (Recall-oriented understudy for gisting evaluation) defines n-gram overlap to human summaries

Procedures for Types 1-3

1. Define Task
   - Example: summarization of text
2. Define Dataset
   1. Search for academic dataset
      - They already have baselines
      - E.g., Document Understanding Conference (DUC)
   2. Define your own (harder, need more new baselines)
      - Try to connect to your research
      - E.g., For summarization, can use Wikipedia: intro paragraph and the rest of the article
      - Be creative, can try to look into blogs and news also
3. Define your metric
   - Search online for well established metrics on your task
   - E.g., for summarization, Rogue (Recall-oriented understudy for gisting evaluation) defines n-gram overlap to human summaries

Procedures for Types 1-3

1. Define Task
    - Example: summarization of text
2. Define Dataset
    1. Search for academic dataset
        - They already have baselines
        - E.g., Document Understanding Conference (DUC)
    2. Define your own (harder, need more new baselines)
        - Try to connect to your research
        - E.g., For summarization, can use Wikipedia: intro paragraph and the rest of the article
        - Be creative, can try to look into blogs and news also
3. Define your metric
    - Search online for well established metrics on your task
    - E.g., for summarization, Rogue (Recall-oriented understudy for gisting evaluation) defines n-gram overlap to human summaries

Procedures for Types 1-3

4. Split your dataset
    - Training/validation/testing
    - Academic dataset often come pre-split
    - Don’t try to peek into your test set until the very last moment
5. Establish a baseline
    - Implement the simplest model first (for classification, logistic regression on some simple features?)
    - Compute metrics on training and validation sets
    - Analyze errors
    - If metrics are amazing and no errors: done, problem was too easy, restart :)
6. Implement existing neural net model
    - Compute metric on training and validation sets
    - Analyze output and errors
    - Minimum bar for this class (guarantee half of the project score)

Procedures for Types 1-3

4. Split your dataset
   - Training/validation/testing
   - Academic dataset often come pre-split
   - Don’t try to peek into your test set until the very last moment
5. Establish a baseline
   - Implement the simplest model first (for classification, logistic regression on some simple features?)
   - Compute metrics on training and validation sets
   - Analyze errors
   - If metrics are amazing and no errors: done, problem was too easy, restart :)
6. Implement existing neural net model
   - Compute metric on training and validation sets
   - Analyze output and errors
   - Minimum bar for this class (guarantee half of the project score)

Procedures for Types 1-3

4. Split your dataset
   - Training/validation/testing
   - Academic dataset often come pre-split
   - Don't try to peek into your test set until the very last moment
5. Establish a baseline
   - Implement the simplest model first (for classification, logistic regression on some simple features?)
   - Compute metrics on training and validation sets
   - Analyze errors
   - If metrics are amazing and no errors: done, problem was too easy, restart :)
6. Implement existing neural net model
   - Compute metric on training and validation sets
   - Analyze output and errors
   - Minimum bar for this class (guarantee half of the project score)

Procedures for Types 1-3

4. Split your dataset
   - Training/validation/testing
   - Academic dataset often come pre-split
   - Don't try to peek into your test set until the very last moment
5. Establish a baseline
   - Implement the simplest model first (for classification, logistic regression on some simple features?)
   - Compute metrics on training and validation sets
   - Analyze errors
   - If metrics are amazing and no errors: done, problem was too easy, restart :)
6. Implement existing neural net model
   - Compute metric on training and validation sets
   - Analyze output and errors
   - Minimum bar for this class (guarantee half of the project score)

Procedures for Types 1-3

7. Try out different model variants
   - CNN/RNN/Hybrid?
   - Depth/width variation?
   - ReLU/tanh/leaky ReLU/etc.?

Some tips and suggestions:

- Always be close to your data
  - Try to visualize the dataset
  - collect summary statistics
  - Look at errors
  - Analyze how different hyperparameters affect performance

Procedures for Types 1-3

7. Try out different model variants
    - CNN/RNN/Hybrid?
    - Depth/width variation?
    - ReLU/tanh/leaky ReLU/etc.?

Some tips and suggestions:

- Always be close to your data
    - Try to visualize the dataset
    - collect summary statistics
    - Look at errors
    - Analyze how different hyperparameters affect performance

Additional step for type 3: a new model

- Do all other steps first
- Gain intuition of why existing models are flawed
- Discuss with others (your research advisor and your peers), can also catch me after Tuesday meeting
- Implement new models and iterate quickly over ideas
- Set up efficient experimental framework (automate stuffs with scripts)
- Take advantage of Schooner
- Remember to build simpler new models first

Some project ideas

- Object detection and segmentation
- Text summarization
- Named entity recognition
  - predicting if a word is a person, a place, an organization, misc, or others (not a name)
- Simple question answering system
- Image/video captioning
- Use DL to solve an a Kaggle challenge
- More ideas: see past projects of Stanford machine learning class `http://cs229.stanford.edu/projects2013.html`

Grading and expectation

- Presentation or video screencast is expected (TBD). Written report is not mandatory but encouraged
    - Tentative presentation date: 5/5
- Group projects are graded the same as single person projects. Given more hands there, a slight penalty is imposed for small group but goes steep as size increases (out of 40)

# members in group	2	3	4	5
Penalty	-2	-4	-8	-16

- Additional bonus (4% overall) if the projects lead to a submitted publication before course ends

Grading

- Presentation: (10 out of 40)
  - clarity, structure, references
  - background literature survey, good understanding of the problem
  - good insights and discussions of methodology, analysis, results, etc.
- Technical: (15 out of 40)
  - correctness
  - depth
  - innovation
- Evaluation and results: (15 out of 40)
  - sound evaluation metric
  - thoroughness in analysis and experimentation
  - results and performance