



Deep Learning

Solving DL Problems

U Kang
Seoul National University



Overview

- Discuss practical techniques to solve DL problems



Understand Problems

- What should be the key components to solving this problem?
- Are there unique constraints or properties to this problem?
- How should we allocate resources to solve this problem effectively?



Data Preprocessing

- Normalization is useful
 - E.g., z-normalization
- Remove noise or useless parts in the data



Data Augmentation

- A very useful technique especially for image data
- E.g., rotation, flipping, cropping
- It is helpful to try all combinations
 - E.g., when doing rotation, try all different combinations



Ensemble Learning

- Make many different models and ensemble them
- The models may differ in terms of
 - Architecture
 - Data
 - Hyperparameters
- Make decision using the majority vote
- Verify that different models show different errors
 - E.g., model 1 works better for class A, while model 2 is better for class B



Imbalanced Classification

- Oversampling is useful
- May try well-known methods
 - SMOTE
 - ADASYN



Incremental Approach

- Incrementally update your model and see the effect of each component
- Monitor your performance for each component
 - E.g., validation error
- Make sure you understand the motivation of using each technique



Transfer Learning

- Helpful when you have limited data
- E.g., you may transfer the model trained from Imagenet to your task



Overfitting and Underfitting

- Make sure your model has the right capacity for the task at hand
- Tune your methods based on performance
 - ☐ If training error is large, your model is underfitting
 - ☐ If training error is small but test error is large, your model is overfitting
- When overfitting, try regularization methods
 - ☐ Batch norm
 - ☐ Dropout
 - ☐ Early stopping
 - ☐ Tune # of layers
 - ☐ ...



Resource Allocation

- We have limited resources
 - GPUs
 - Manpower
 - Time
- Need to make a decision to allocate resources
- Make an end-to-end model as early as possible, which may help identify the bottleneck



Hyperparameter Search

- Hyperparameter search is crucial for a better performance
- Start with the well-known baselines, and tune the parameters
- Search range should be divided in log-scale units
- Coarse levels to fine-grained levels
- But, finding a better architecture should be performed before detailed hyperparameter searches



Questions?