

Course 2: Improving Deep Neural Networks: Hyperparameter tuning,
Regularization and Optimization

Coursera Deep Learning Course 2 - Week 1: Practical Aspects of Deep Learning

Jungwon Kang

Aug 23 2018

Table of Contents (1/2)

1. Setting up your machine learning application

- 1.1 Train / Dev / Test sets
- 1.2 Bias / Variance
- 1.3 Basic recipe for machine learning

2. Regularizing your neural network

- 2.1 Regularization
- 2.2 Why regularization reduces overfitting?
- 2.3 Dropout regularization
- 2.4 Understanding dropout
- 2.5 Other regularization methods

Table of Contents (2/2)

3. Setting up your optimization problem

- 3.1 Normalizing inputs

- 3.2 Vanishing / exploding gradients

- 3.3 Weight initialization for deep networks

- 3.4 Numerical approximation of gradients

- 3.5 Gradient checking

- 3.6 Gradient checking implementation notes

List of Programming Assignments

1. Initialization
2. Regularization
3. Gradient Checking

Programming Assignment 1:

Initialization

1.1 Neural network model

- Three-layer neural network is given by coursera.

1.2 Zero initialization

- [Task] Write a code for `initialize_parameters_zeros()`.

1.3 Random initialization

- [Task] Write a code for `initialize_parameters_random()`.

1.4 He initialization

- [Task] Write a code for `initialize_parameters_he()`.

Programming Assignment 2:

Regularization

2.1 Non-regularized model

- Three-layer neural network is given by coursera.

2.2 L2 Regularization

- [Task] Write a code for computing L2 regularization cost.
- [Task] Write a code for applying L2 regularization in the backward propagation.

2.3 Dropout

- [Task] Write a code for forward propagation with dropout
- [Task] Write a code for backward propagation with dropout

Programming Assignment 3:

Gradient Checking

3.1 1-dim gradient checking

- [Task] Write a forward & backward propagation, and gradient checking for 1-dim function.

3.2 N-dim gradient checking

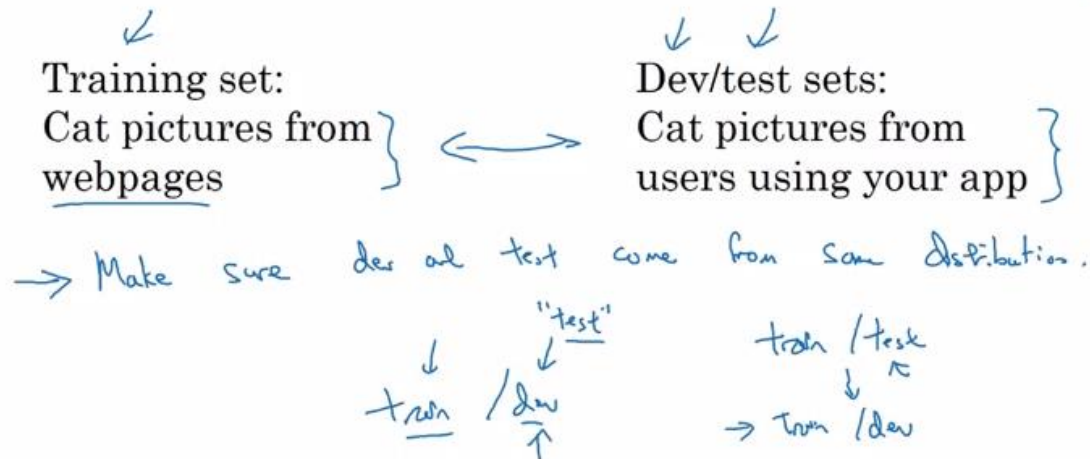
- [Task] Write a forward & backward propagation, and gradient checking for N-dim function.

Question

- How can we get the distribution of Dev/test sets?
- How can we measure similarity between two distributions?

Mismatched train/test distribution

Courts



Not having a test set might be okay. (Only dev set.)