[Lec 03]

Classification Review & Intro

training data

- consists of samples
- {xi, yi} (i = 1 ~ N)

Learn a ‘line’ that classifies (Linear decision boundary)

- **Using soft-max classifier**

→ Take row of W(weight vector - for each class) and multiply that row with x

→ Takes the numbers to a probability distribution

- **Using Logistic Regression**

Training with soft-max and cross-entropy loss

- maximize the correct probability of the correct class
- minimize the negative log probability of that class

cf) NLL (Negative Log Likelihood) ⇒ Cross Entropy Loss

Cross Entropy Error (Loss)

- True Probability distribution be p
- Computed model probability be q

Assuming ground truth → probability distribution (one-hot vector)

→ only term left is the **negative log probability of the true class**

cf) Matrix Notation $f = Wx$

Traditional ML optimization

- Basic Classifier

→ sort of fairly **simple** classifiers (linear classifier)

→ unhelpful when problem is **complex : Need for a new powerful classifier**

Neural Networks Intro

- complex data (visual + languages)

→ for complex functions

→ need for nonlinear decision boundaries

Word Vectors

- objective -build better representation space

Simultaneously change representations of **words + weights**

+) Think of word representations as **having an extra layer in the neural net** → taking out a column of L

→ Not practical, but in theory makes sense

- deeper multi-layer networks

Artificial Neuron

- axons from neuron : input
- synapse: multiplication with weight
- cell body: summing excitations (with own bias)
  + threshold : activation function
- signal: output axon

→ **Be a binary logistic regression unit or variants (different activation function)**

Neural Network

Running several logistic regressions at the same time

⇒ We get a vector of outputs (Don’t have to decide ahead of time what the logistic regressions are trying to predict)

⇒ Given, a **task (ex. sentiment) & minimize cross entropy loss**

⇒ Having more layers → change the input space around (can get more sophisticated dividers - other than linear **(non-linearity**))

Motivating Question

→ If we could have a deeper network we can learn to solve difficult, sophisticated questions

→ **how to train such a network?**

Named Entity Recognition (NER)

Task: Find & Classify names in text

ex) organization or people or places ...

→ **Predict Entities by classifying words in context (have surrounding words around it) and extracting entities as word subsequences**

Why might NER be hard?

- Hard to work out Boundaries of an entity
- If something is an entity?
  ex) Future School? or future school?
- Novel Entity classification problem
- Ambiguous

Binary True vs Corrupted word window classification

Build classifier for language that works inside a context

→ prob) Ambiguity Occur

ex) To seed: ‘take out seed’ ‘plant seed’

Window Classification

→ **Classify a word in its context window of neighboring words**

sol 1) Average the word vectors in a window & classify the average

→ lose position information

sol 2) Big Vector of a word vector (concatenate of word)

→ Put through soft-max classifier

→ compute score w/ a sub-layer neuron network (learn non-linear interactions)

→ Calculate the score (larger the better)

Matrix calculus Intro

Computing Gradients by Hand

- Fully vectorized gradients (Matrix Calculus)

Jacobian Matrix