Lecture 3 - Backprop and Neural Networks

Named Entity Recognition (NER)

- 개체명 인식 : 말뭉치로부터 각 개체(entity)-어떤 이름을 의미하는 단어-의 유형을 인식
    - 유형 예시 : 사람, 장소, 시간, 조직 등

1. Classify each word (vector) in its context window
2. Put those through neural network layer
3. Feed it through a logistic classifier
    - Train logistic classifier on hand-labeled data to classify center word {yes/no} for each class based on a concatenation of word vectors in a window
    - Usually use multi-class softmax

Matrix calculus

- Chain Rule & Jacobians

- For composition of one-variable functions: **multiply derivatives**

$$z = 3y$$
$$y = x^2$$
$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} = (3)(2x) = 6x$$

- For multiple variables at once: **multiply Jacobians**

$$\boldsymbol{h} = f(\boldsymbol{z})$$
$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$
$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \ldots$$

Stanf

▼ Slides

- Suppose we now want to compute $\frac{\partial s}{\partial \boldsymbol{W}}$
  - Using the chain rule again:

$$\frac{\partial s}{\partial \boldsymbol{W}} = \frac{\partial s}{\partial \boldsymbol{h}}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$
$$\frac{\partial s}{\partial \boldsymbol{b}} = \frac{\partial s}{\partial \boldsymbol{h}}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}$$

The same! Let's avoid duplicated computation …

- Suppose we now want to compute $\dfrac{\partial s}{\partial \boldsymbol{W}}$
  - Using the chain rule again:

$$\frac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$

$$\frac{\partial s}{\partial \boldsymbol{b}} = \boldsymbol{\delta} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \boldsymbol{\delta}$$

$$\boldsymbol{\delta} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} = \boldsymbol{u}^T \circ f'(\boldsymbol{z})$$

δ is the local error signal

- What is $\dfrac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta} \dfrac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$
  - $\boldsymbol{\delta}$ is going to be in our answer
  - The other term should be $\boldsymbol{x}$ because $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$

- Answer is: $\dfrac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta}^T \boldsymbol{x}^T$

δ is local error signal at z
x is local input signal

- What shape should derivatives be?
  1. Use Jacobian form as much as possible, reshape to follow the shape convention at the end

2. Always follow the shape convention

Backpropagation

- Takes derivatives and uses the (generalized, multivariate, or matrix) chain rule
- Computation Graph
  - software represents neural net equations as a graph
    - source nodes: inputs / interior nodes: operations

$$
\begin{aligned}
s &= \boldsymbol{u}^T \boldsymbol{h} \\
\boldsymbol{h} &= f(\boldsymbol{z}) \\
\boldsymbol{z} &= \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \\
\boldsymbol{x} & \quad \text{(input)}
\end{aligned}
$$