14기 정규세션

ToBig's 14기 고경태

Word Window Classification, Neural Networks, and Matrix Calculus

Contents

14기 정규세션

ToBig's 14기 고경태

Classification review, introduction

Classification setup and notation

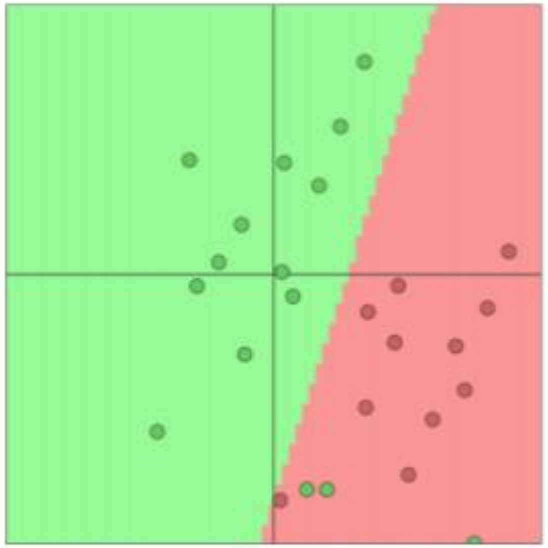
Generally we have a training dataset consisting of samples

$$\{x_i, y_i\}^N_{i=1}$$

1. Training dataset을 i=1부터 N까지 xi라는 inputs과 yi라는 output(label or class)에 대해 가지고 있음.
2. NLP 에서는 xi는 단어나 문장, 문서를 의미하고, yi는 classes일수도 words나 다른 것들일 수도 있음.

Classification intuition

- Simple illustration case:
  - Fixed 2D word vectors to classify
  - Using softmax/logistic regression
  - Linear decision boundary



Visualizations with ConvNetJS by Karpathy!

1. 위의 데이터를 ML/Deep Learning 방법으로 분류의 과정을 거치게 됨.
2. 분류는 위 그림처럼 비슷한 output끼리 모이도록 경계를 긋는 것을 의미.
3. 전통적인 ML접근에서는 softmax / logistic regression을 이용해서 output의 class를 구분할 경계선을 결정하는 것을 의미

Details of the softmax classifier

$$p(y|x) = \frac{\exp(W_y.x)}{\sum_{c=1}^{C} \exp(W_c.x)}$$

$$W_y.x = \sum_{i=1}^{d} W_{yi}x_i = f_y$$

$$p(y|x) = \frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)} = \text{softmax}(f_y)$$

Training with softmax and cross-entropy loss

$$-\log p(y|x) = -\log\left(\frac{\exp(f_y)}{\sum_{c=1}^{C}\exp(f_c)}\right)$$

값을 학습할 때, 올바르게 y값을 예측하도록 확률을 극대화 or negative한 값을 최소화하도록 학습을 하게 됨.

Cross entropy loss?

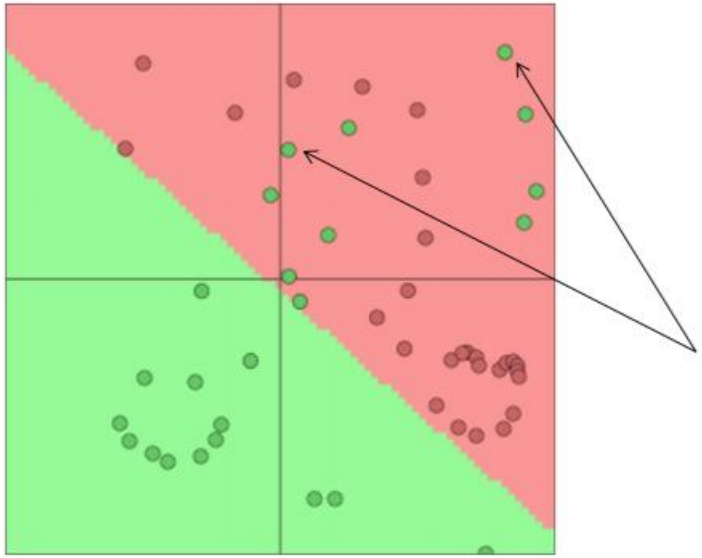
$$H(p, q) = -\sum_{c=1}^{C} p(c) \log q(c)$$

p : 실제 확률 분포
q : 예측한 확률 분포

p : [0 , … , 0 , 1, 0 , … , 0]
q : [0.01, … , 0.02 , 0.8 , 0.01 … , 0]

Classification over a full dataset

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} -\log\left(\frac{e^{f_{y_i}}}{\sum_{c=1}^{C} e^{f_c}}\right)$$



HOW?

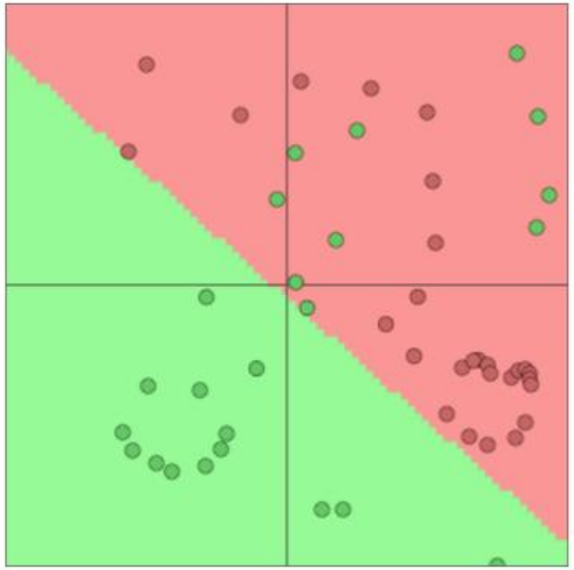
14기 정규세션

ToBig's 14기 고경태

Neural Networks introduction

Neural Network Classifiers

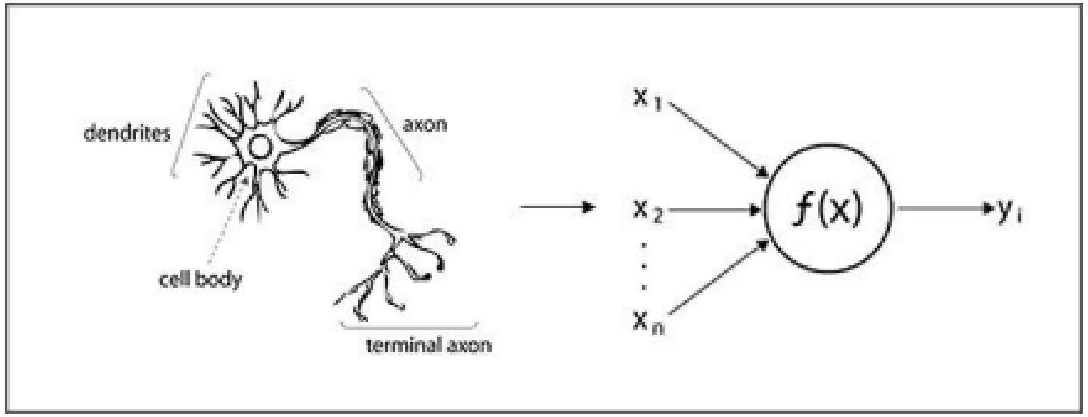
- In original space

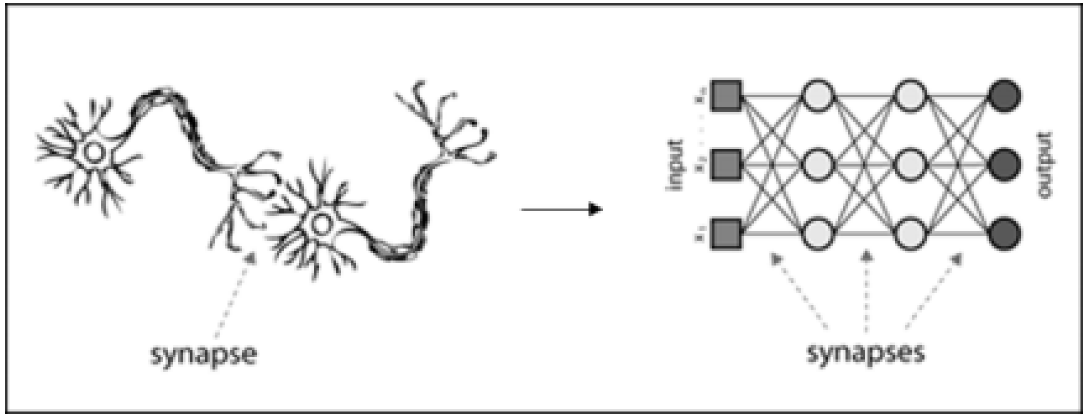


$$\nabla_\theta J(\theta) = \begin{bmatrix} \nabla_{W_{\cdot 1}} \\ \vdots \\ \nabla_{W_{\cdot d}} \\ \nabla_{x_{aardvark}} \\ \vdots \\ \nabla_{x_{zebra}} \end{bmatrix} \in \mathbb{R}^{Cd+\boxed{Vd}}$$

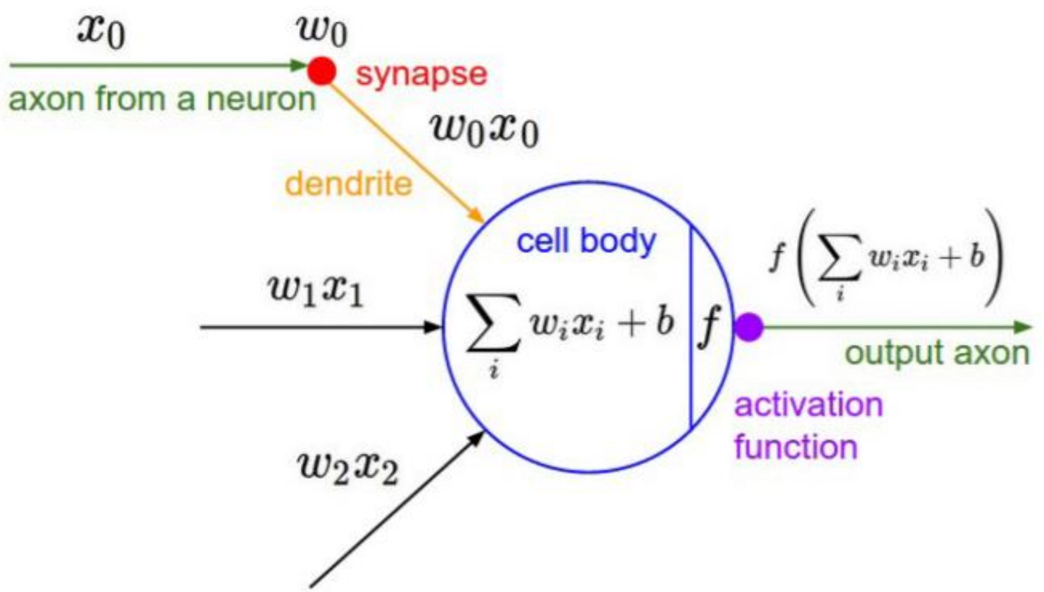
Very large number of parameters!

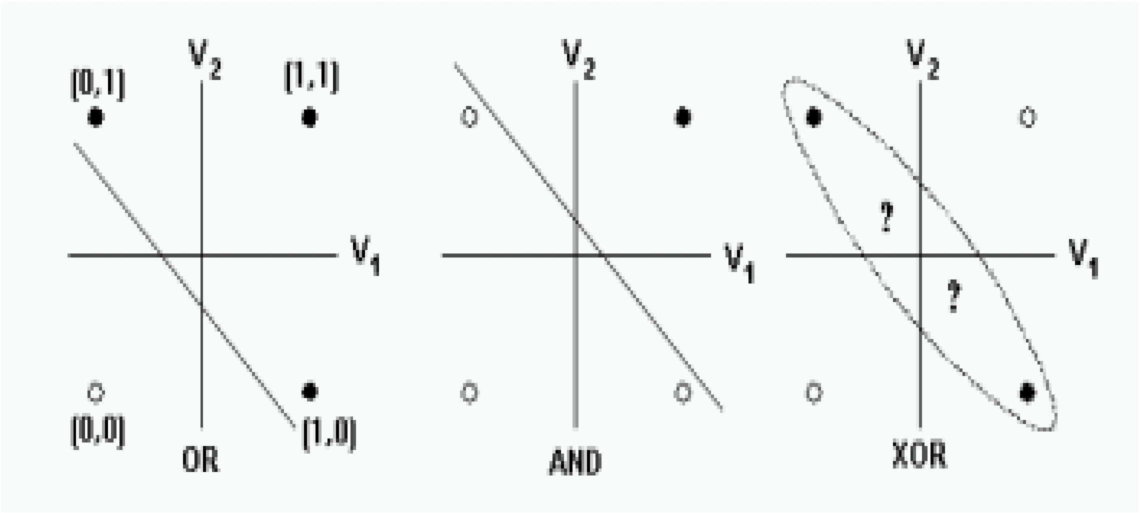
Neural Network Classifiers



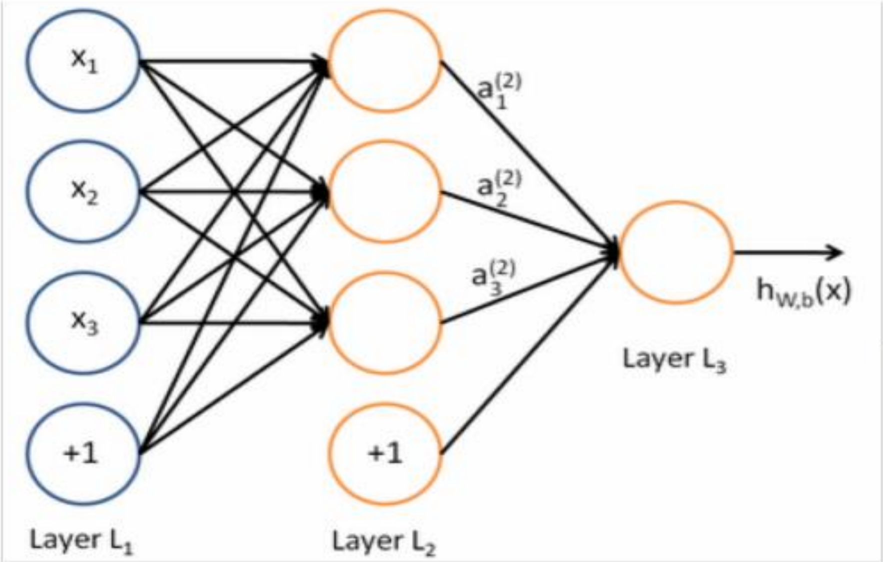


Neural Network history(NeuralNetwork_Baic강의 참고)





Multilayer Perceptron



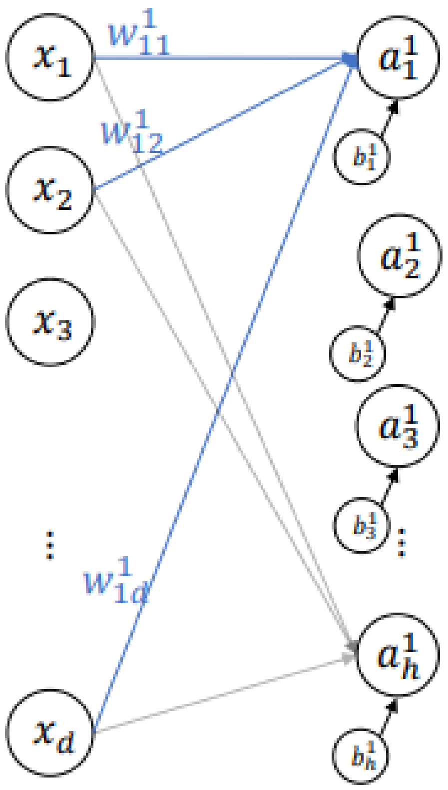
d+1개의 입력 노드

h+1개의 은닉층 노드

c개의 출력 노드(부류 개수)

(d+1)*h + (h+1)*c 개의 가중치 개수
(=파라미터의 개수)
(2layer의 경우)

Multilayer Perceptron



$$\boldsymbol{w}_i^k = (w_{i1}^k, w_{i2}^k, \cdots, w_{id}^k)^T$$

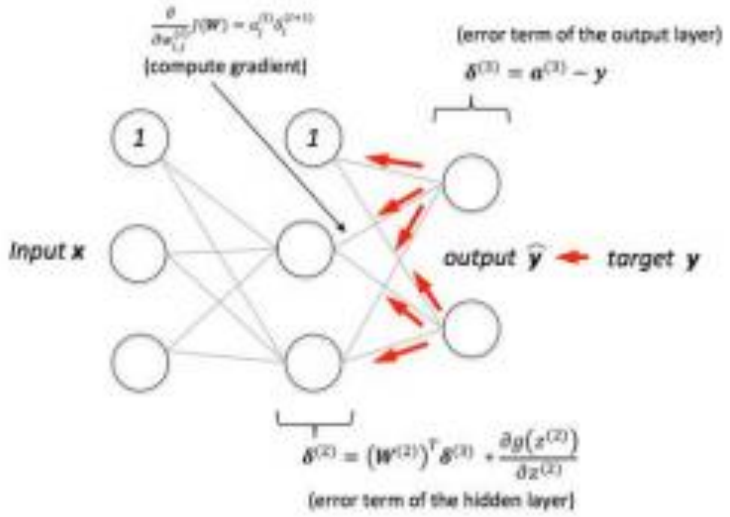
$$W^k = (\boldsymbol{w}_1^k \quad \boldsymbol{w}_2^k \quad \cdots \quad \boldsymbol{w}_h^k)^T$$

$$W^k = \begin{pmatrix} w_{11}^k & w_{21}^k & \cdots & w_{i1}^k & \cdots & w_{h1}^k \\ w_{12}^k & w_{22}^k & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ w_{1j}^k & & & w_{ij}^k & & w_{hj}^k \\ \vdots & & & & \ddots & \vdots \\ w_{1d}^k & \cdots & \cdots & w_{id}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$W^k\boldsymbol{x} + \boldsymbol{b}^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} + \begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix}$$

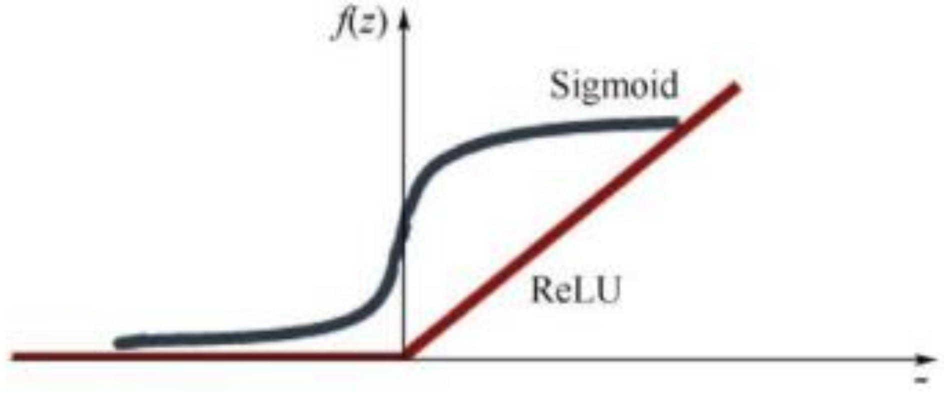
h*d d*1 h*1

Multilayer Perceptron

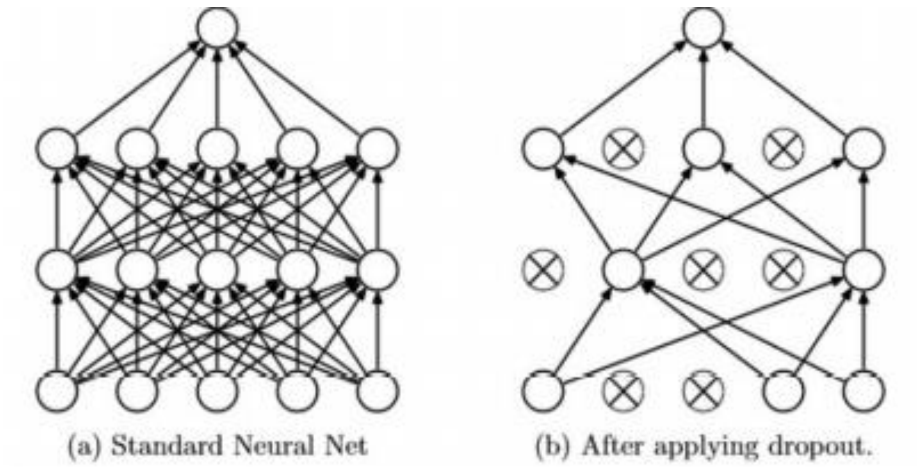


은닉층을 추가시킨 다층 퍼셉트론이 XOR문제를 해결할 수 있음
이를 학습시키는 **오류 역전파**

Deep Learning



ReLU 활성화 함수를 통한 기울기 소실 문제와
학습시간 문제 해결



Dropout을 통한 과적합 방지

14기 정규세션

ToBig's 14기 고경태

Neural Networks in NLP

1. Named Entity Recognition (NER)

The task: find and classify names in text, for example:

The European Commission [ORG] said on Thursday it disagreed with German [MISC] advice.

Only France [LOC] and Britain [LOC] backed Fischler [PER] 's proposal .

"What we have to be extremely careful of is how other countries are going to take Germany 's lead", Welsh National Farmers ' Union [ORG] (NFU [ORG]) chairman John Lloyd Jones [PER] said on BBC [ORG] radio .

1. 글에서 특정한 항목에 대한 언급 추정
2. 질문 답변의 경우, 답변은 보통 인명인 경우가 많음
3. 요구되는 많은 정보들은 인명과 연관되는 경우가 많음
4. 다른 분류에도 사용 될 가능성

1. Named Entity Recognition (NER)

Foreign	ORG	B-ORG
Ministry	ORG	I-ORG
spokesman	O	O
Shen	PER	B-PER
Guofang	PER	I-PER
told	O	O
Reuters	ORG	B-ORG
that	O	O
:	:	👆 BIO encoding

분류기를 실행하고 클래스를 할당

Why might NER be hard?

First National Bank Donates 2 Vans To Future School Of Fort Smith

where Larry Ellison and Charles Schwab can live discreetly amongst wooded estates. And

1. 고유명사의 경계를 정하기가 어려움. (ex, First National Bank or National Bank)
2. 개체가 아닌지 알기가 어려움 (ex, Future School= 'Future School' or 미래의 학교?)
3. 개체 분류가 모호하며 문맥에 의존한다. (ex, 'Charles Schwab'은 사람인가 조직(기관)인가?)

2. Binary word window classification

해결해야 하는 것은 문맥상 애매모호한 단어들… → 문맥까지 고려하는 Window Calssification!

Example: auto-antonyms:

- "To sanction" can mean "to permit" or "to punish"
- "To seed" can mean "to place seeds" or "to remove seeds"

문맥을 고려하여 둘 중 하나를 선택!

2. Window Classification

Idea : 중심 단어와 주변 단어들 (context)를 함께 분류 문제에 활용하는 방법

$$X_{window} = [\ x_{museums} \quad x_{in} \quad x_{Paris} \quad x_{are} \quad x_{amazing}\ ]^T$$

3. Window Classification : Softmax

$$X_{window} = [\ x_{museums} \quad x_{in} \quad x_{Paris} \quad x_{are} \quad x_{amazing}\ ]^T$$

Resulting vector x_{window} = $\boxed{x \in R^{5d}}$, a column vector!

$$\boldsymbol{w}_i^k = (w_{i1}^k, w_{i2}^k, \cdots, w_{id}^k)^T$$

$$W^k = (\boldsymbol{w}_1^k \quad \boldsymbol{w}_2^k \quad \cdots \quad \boldsymbol{w}_h^k)^T$$

$$W^k = \begin{pmatrix} w_{11}^k & w_{21}^k & \cdots & w_{i1}^k & \cdots & w_{h1}^k \\ w_{12}^k & w_{22}^k & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ w_{1j}^k & & & w_{ij}^k & & w_{hj}^k \\ \vdots & & & & \ddots & \vdots \\ w_{1d}^k & \cdots & \cdots & w_{id}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$W^k\boldsymbol{x} + \boldsymbol{b}^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} \boxed{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}} + \begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix}$$

h∗d d∗1 h∗1

3. Window Classification : Softmax

$$X = X_{window}$$

predicted model
output probability

$$\boxed{\hat{y}_y} = p(y|x) = \frac{\exp(\boxed{W_y.x})}{\sum_{c=1}^{C} \exp(W_c.x)}$$

- With cross entropy error as before:

same

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} -\log \left(\frac{e^{\boxed{f_{y_i}}}}{\sum_{c=1}^{C} e^{f_c}} \right)$$

3. Classification for NER Location

Example: Not all museums in Paris are amazing .

museums in Paris are amazing → True window

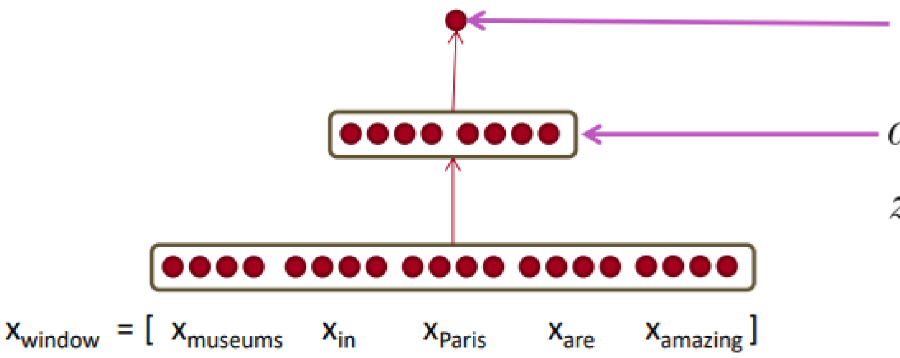
Not all museums in Paris → Corrupt

3. Window Classification : Softmax

$$s = U^T f(Wx + b)$$

$$x \in \mathbb{R}^{20\times 1}, W \in \mathbb{R}^{8\times 20}, U \in \mathbb{R}^{8\times 1}$$

$$score(x) = U^T a \in \mathbb{R}$$



$$s = U^T a$$

$$a = f(z)$$

$$z = Wx + b$$

3. The max-margin loss

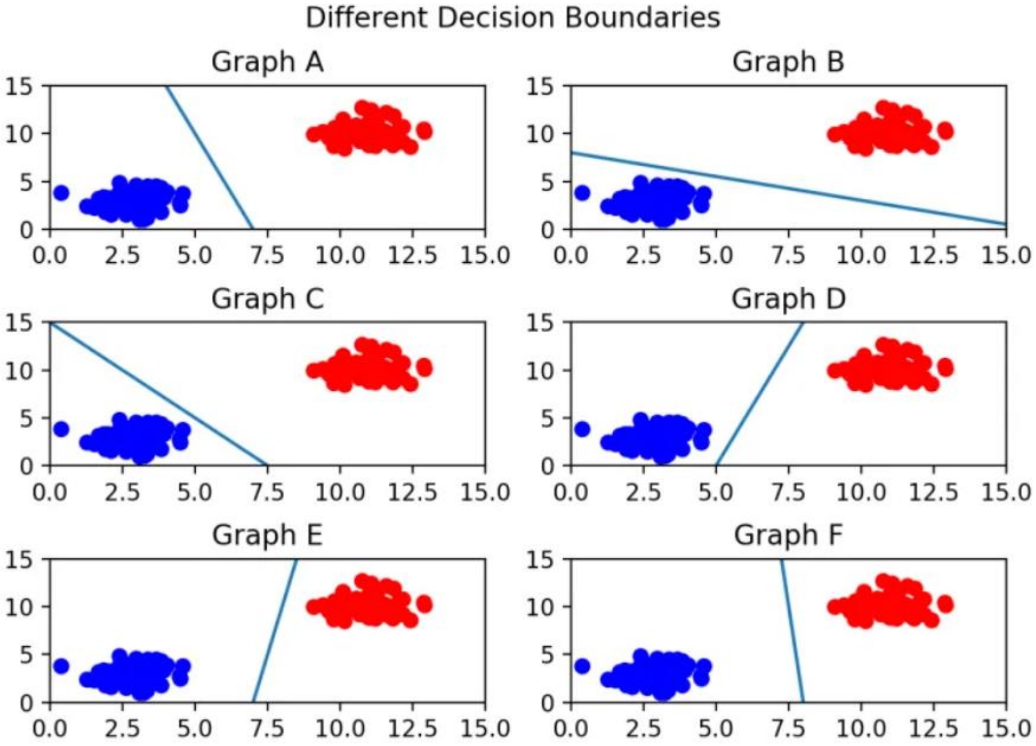
- s = score(museums in Paris are amazing)
- s_c = score(Not all museums in Paris)

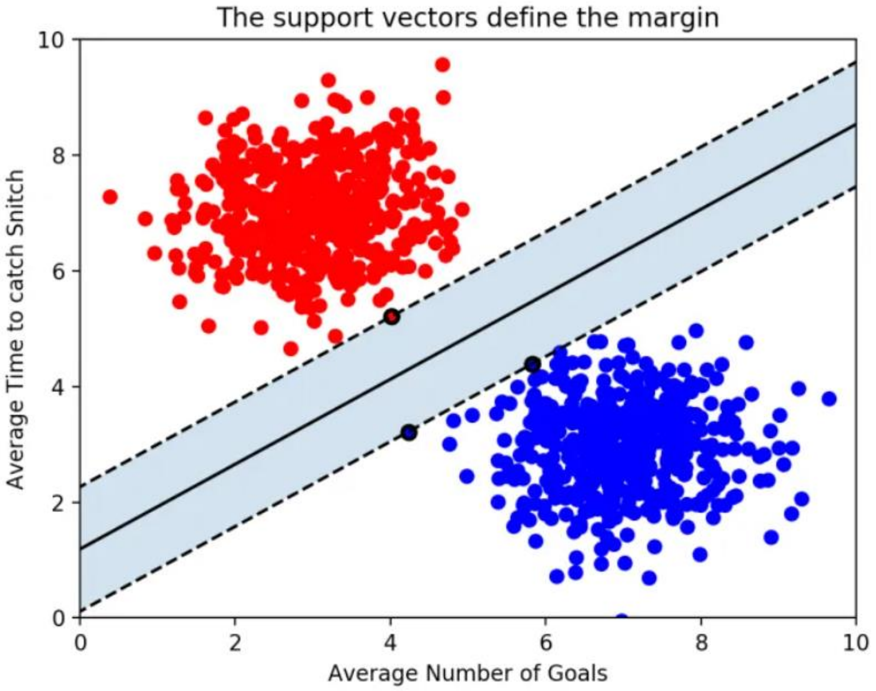
Minimize

$$\boxed{J = \max(0, 1 - s + s_c)}$$

정답과 오답 사이의 거리를 최대로 만드는 margin 찾기!
어디서 많이 본 것 같은데..?

3. The max-margin loss (svm)





3. The max-margin loss

- s = score(museums in Paris are amazing)
- s_c = score(Not all museums in Paris)

Minimize

$$\boxed{J = \max(0, 1 - s + s_c)}$$

3. Stochastic Gradient Descent

$$\theta^{new} = \theta^{old} - \alpha \nabla_{\theta} J(\theta)$$

α = *step size* or *learning rate*

역전파를 이용하여 손실함수 최소화!

14기 정규세션

ToBig's 14기 고경태

Matrix calculus

Jacobian Matrix: Generalization of the Gradient

$$f(\boldsymbol{x}) = f(x_1, x_2, \dots, x_n) \longrightarrow \boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1, x_2, \dots, x_n), \dots, f_m(x_1, x_2, \dots, x_n)]$$

$$\frac{\partial f}{\partial \boldsymbol{x}} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}\right] \longrightarrow \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$\boxed{\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)_{ij} = \frac{\partial f_i}{\partial x_j}}$$

Chain Rule

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

One-variable → multiply derivatives

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

Multiple variable → multiply Jacobians

Example Jacobian : Elementwise activation Function

$$\boldsymbol{h} = f(\boldsymbol{z}), \text{what is } \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}? \qquad \boldsymbol{h}, \boldsymbol{z} \in \mathbb{R}^n$$

$$h_i = f(z_i)$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Function has *n* outputs and *n* inputs → *n* by *n* Jacobian

Example Jacobian : Elementwise activation Function

$$\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}\right)_{ij} = \frac{\partial h_i}{\partial z_j} = \frac{\partial}{\partial z_j} f(z_i) \qquad \text{definition of Jacobian}$$

$$= \begin{cases} f'(z_i) & \text{if } i = j \\ 0 & \text{if otherwise} \end{cases} \qquad \text{regular 1-variable derivative}$$

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} = \begin{pmatrix} f'(z_1) & & 0 \\ & \ddots & \\ 0 & & f'(z_n) \end{pmatrix} = \operatorname{diag}(\boldsymbol{f}'(\boldsymbol{z}))$$

z(i) 와 z(j) 가 같을 때 미분이 됨
다르면 0으로 없어짐

Example Jacobian : Elementwise activation Function

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}) = \boldsymbol{W}$$

$$\frac{\partial}{\partial \boldsymbol{b}}(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}) = \boldsymbol{I} \ \ \text{(Identity matrix)}$$

$$\frac{\partial}{\partial \boldsymbol{u}}(\boldsymbol{u}^T\boldsymbol{h}) = \boldsymbol{h}^{\boldsymbol{T}}$$

+

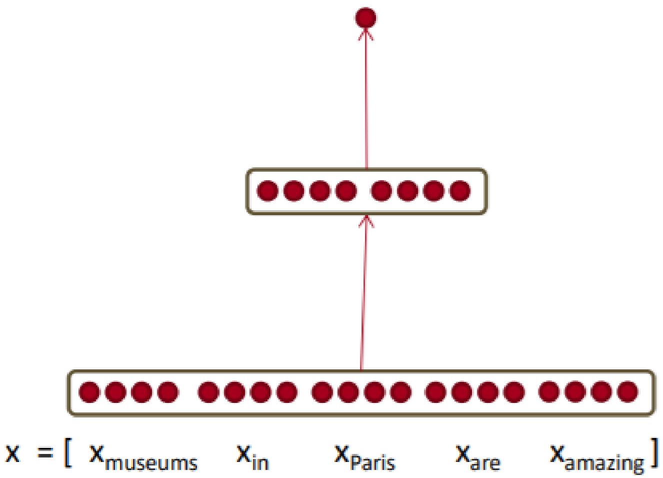
$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} = \begin{pmatrix} f'(z_1) & & 0 \\ & \ddots & \\ 0 & & f'(z_n) \end{pmatrix} = \mathrm{diag}(\boldsymbol{f}'(\boldsymbol{z}))$$

Back to our Neural Net!

$$s = \boldsymbol{u}^T \boldsymbol{h}$$

$$\boldsymbol{h} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$$

$$\boldsymbol{x} \quad \text{(input)}$$



Let's find $\frac{\partial s}{\partial \boldsymbol{b}}$

손실함수의 gradient를 계산해야 하지만,
쉽게 score의 gradient를 먼저 계산해보자!

1. Break up equations into simple pieces

$$s = \boldsymbol{u}^T \boldsymbol{h}$$

$$\boldsymbol{h} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$$

$$\boldsymbol{x} \quad \text{(input)}$$

$$\Rightarrow$$

$$s = \boldsymbol{u}^T \boldsymbol{h}$$

$$\boldsymbol{h} = f(\boldsymbol{z})$$

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$

$$\boldsymbol{x} \quad \text{(input)}$$

2. Apply the chain rule

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

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

3. Write out the Jacobians

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

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

$$= \boldsymbol{u}^T \operatorname{diag}(f'(\boldsymbol{z}))\boldsymbol{I}$$
$$= \boldsymbol{u}^T \circ f'(\boldsymbol{z})$$

Useful Jacobians from previous slide

$$\boxed{\frac{\partial}{\partial \boldsymbol{h}}(\boldsymbol{u}^T \boldsymbol{h})} = \boldsymbol{h}^T$$
$$\boxed{\frac{\partial}{\partial \boldsymbol{z}}(f(\boldsymbol{z}))} = \operatorname{diag}(f'(\boldsymbol{z}))$$
$$\boxed{\frac{\partial}{\partial \boldsymbol{b}}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})} = \boldsymbol{I}$$

Re – using Computation

Suppose we now want to compute $\frac{\partial s}{\partial \boldsymbol{W}}$

- Using the chain rule again:

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

Re – using Computation

Using the chain rule again:

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

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

파란색 부분의 계산과정이 같다.
계산을 줄여주는 장점!

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

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

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

δ is local error signal

Derivative with respect to Matrix: Output shape

$\boldsymbol{W} \in \mathbb{R}^{n \times m}$

$\dfrac{\partial s}{\partial \boldsymbol{W}}$ is *n* by *m*: $\begin{bmatrix} \dfrac{\partial s}{\partial W_{11}} & \cdots & \dfrac{\partial s}{\partial W_{1m}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial s}{\partial W_{n1}} & \cdots & \dfrac{\partial s}{\partial W_{nm}} \end{bmatrix}$

Matrix로 확장!

Remember $\dfrac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta} \dfrac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$

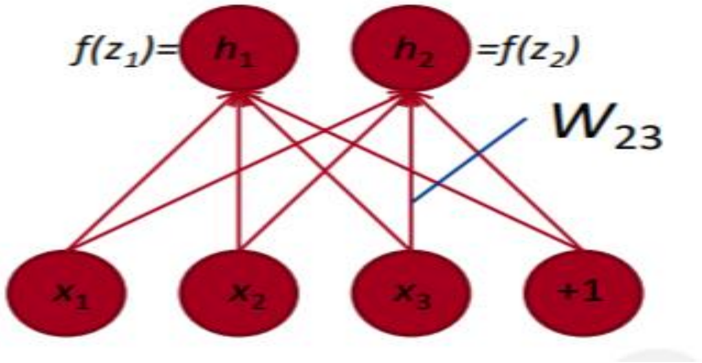
$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$

It turns out $\dfrac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta}^T \boldsymbol{x}^T$

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

Why the Transposes?

$$\frac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta}^T \boldsymbol{x}^T = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix} [x_1, ..., x_m] = \begin{bmatrix} \delta_1 x_1 & \dots & \delta_1 x_m \\ \vdots & \ddots & \vdots \\ \delta_n x_1 & \dots & \delta_n x_m \end{bmatrix}$$



참고 자료

https://gnoej671.tistory.com/4?category=1034944

https://lovit.github.io/nlp/2019/02/16/logistic_w2v_ner/

https://happyzipsa.tistory.com/4

http://hleecaster.com/ml-svm-concept/

https://www.youtube.com/watch?v=8CWyBNX6eDo&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&index=3

Q & A

들어주셔서 감사합니다.