

*Added Prognostic Value of 3D Deep  
Learning-Derived Features from  
Preoperative MRI for Adult-type Diffuse  
Gliomas*

김민수



---

## 김민수 (Min Soo Kim)

### • 학력사항

한국교통대학교 산업공학과 학사  
서울대학교 암연구소 의과대학원 종양생물학 협동과정 석사 (예정)

### • 경력사항

現 서울대학교병원 영상의학과 AICON 연구원  
前 엔씨소프트 금융 AI R&D 시장이해팀  
前 엔씨소프트 금융 AI R&D 투자전략팀  
前 한국수자원기술원 인턴

### 주요 연구 분야 (현재)

Neuroimage (신경 과학 및 뇌 영상), 3D Deep Learning (Vision AI)

### Tech Stack

Python, MySQL, Git, Docker, Linux, SSH, Apache Spark (PySpark), Apache Hive

### 주요 성과

- 교모세포종 환자 생존 분석 SCIE 논문 공동 1저자 (2024 American Society of Neuroradiology) (초록 승인 대기중)
- KOSPI 200, KOSDAQ 150 증권시장 틱데이터 기반 ETL 파이프라인 설계
- 전기차 배터리 불량 유형 분류 및 예측 2D 이미지 딥러닝 모델 설계 및 최적화 논문 공동저자  
(Department of Industry Engineering Conference, Korea National University of Transportation)

# 목차

1. 연구배경
2. 도메인지식 및 연구 활용 방법론 소개
3. 연구 데이터셋 소개
4. 연구 모델 핵심 아키텍쳐 소개
5. 모델 성능 및 결과 이미지
6. 결론
7. 향후 연구

# 연구배경

- **기준 한계**
- **해당 연구를 통해 창출할 수 있는 부가가치**

수술 전 MRI 영상 정보만으로 효과적인 예후 예측변수를 얻어내어 수술전 치료방침 결정에 도움을 줄 수 있음.

- **간략 소개**

현재 제가 속해있는 서울대학교병원 영상의학과 AICON 연구회는 최규성 교수님께서 이끌고 있으며,  
최규성 교수님께서는 비침습적인 방법인 MRI 촬영 기법에서 얻은 이미지에서 예후적 가치를 창출하기 위하여 딥러닝 모델을 활용하고자 하였습니다.  
정확하게는 수술 전 MRI 기반 성인형 뇌교종 이미지에서 3D CNN 딥러닝을 활용하여 뇌교종 이미지 특징을 추출하고  
추가적으로 예후적 가치가 있는 바이오마커로써 부가가치를 창출하고자 "*DeepSurvGlioma*"을 개발하게 되었습니다.

- 해당 연구는 2023년 10월 19일 "Society of Neuro-Oncology (신경종양국제학회)에 출판되었으며, PubMed에 공개되었습니다.  
기존 연구의 후속으로, 저는 "*DeepSurvGlioma*"을 활용하여 사용자 관점에서의 여러 MRI 데이터를 활용하여 성능 향상 및 개선 방안을 모색하고 있습니다.

JOURNAL ARTICLE CORRECTED PROOF

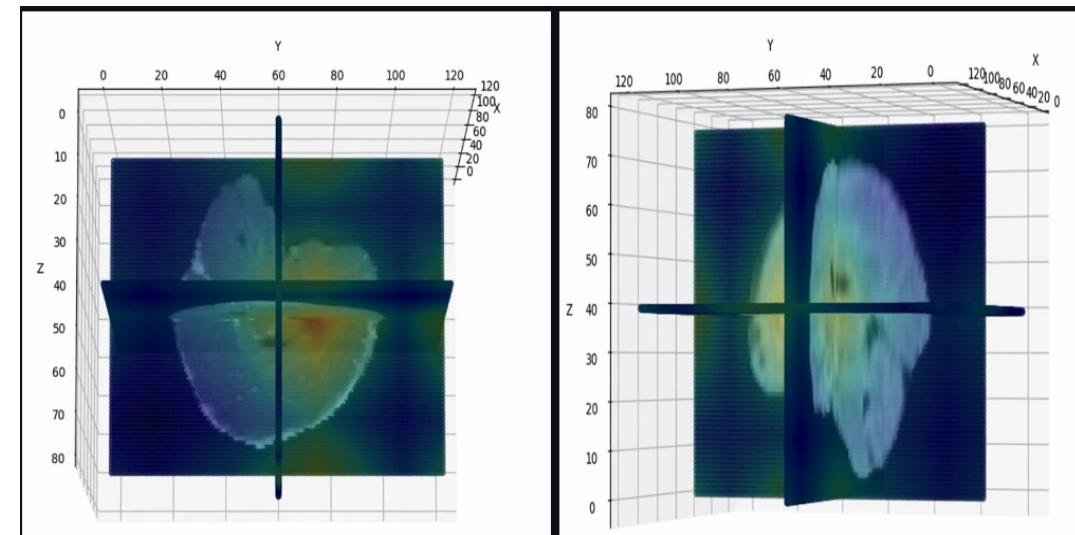
## Added prognostic value of 3D deep learning-derived features from preoperative MRI for adult-type diffuse gliomas

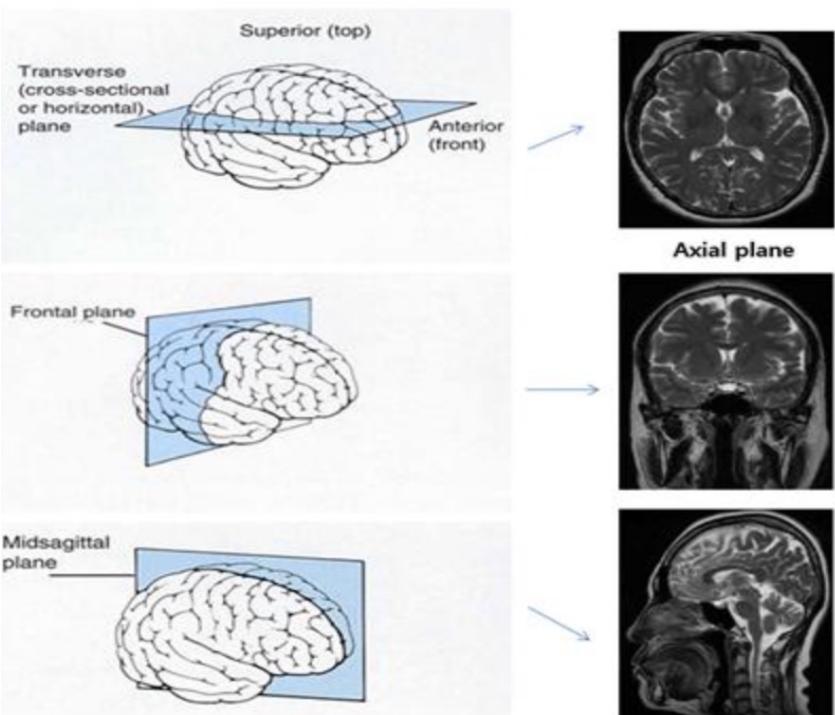
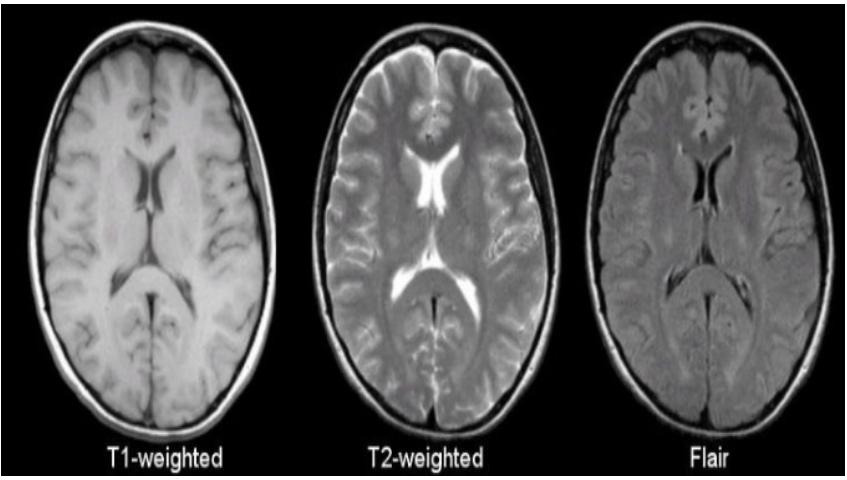
[Get access >](#)

Jung Oh Lee, Sung Soo Ahn, Kyu Sung Choi ✉, Junhyeok Lee, Joon Jang, Jung Hyun Park, Inpyeong Hwang, Chul-Kee Park, Sung Hye Park, Jin Wook Chung ...  
[Show more](#)

*Neuro-Oncology*, noad202, <https://doi.org/10.1093/neuonc/noad202>

Published: 19 October 2023 Article history ▾





# Background

---

- **MRI (Magnentic Resonance Imaging)**

- 강력한 자기장과 라디오 파동을 활용하여 수소 원자(Proton)에서 방출되는 신호를 수집하여 이미지로 변환

- **T1-weighted (T1-강조영상)**

- 뇌의 해부학적 구조에 대해서 파악하기 용이함
- (Fat : White, Water : Dark)

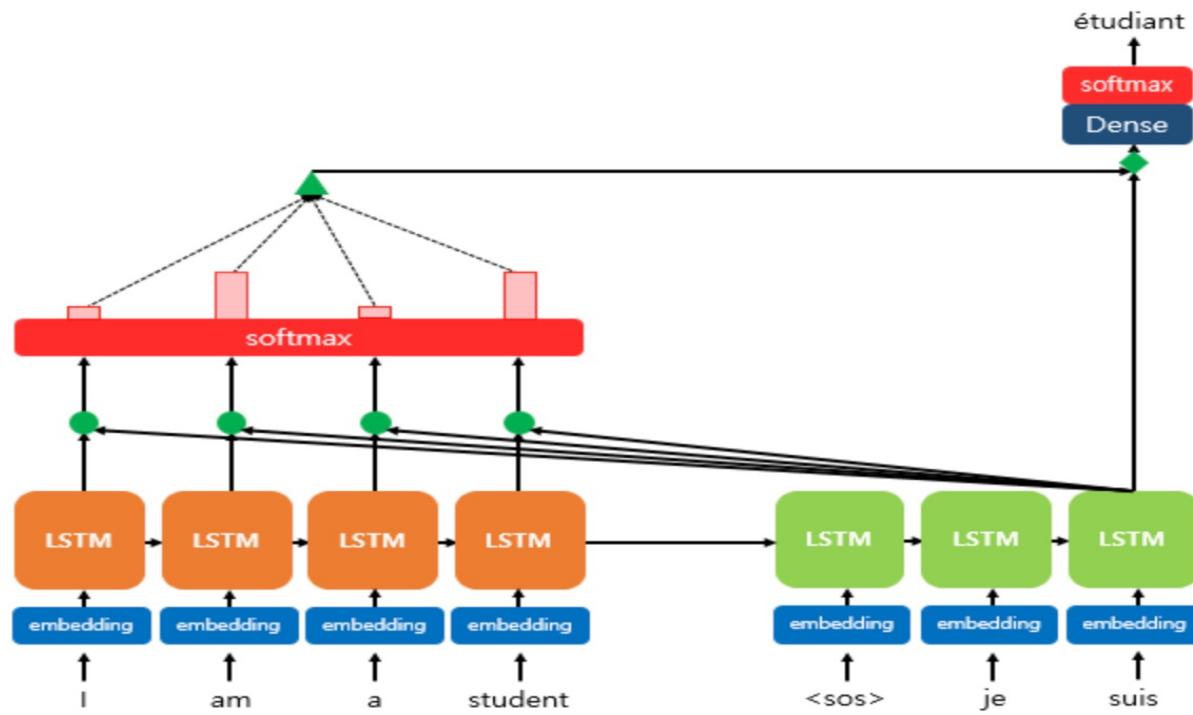
- **T2-weighted (T2-강조영상)**

- 급성기 병변을 파악하기 용이함
- (Fat : Dark, Water : White)

- **Flair**

- CSF 신호를 억제시킨 영상, T2-강조영상보다 병변을 파악하기 더욱 용이함
- (Fat : Dark, Water : Dark)

# Attention Mechanism



RNN 기반 Seq2seq 모델 한계를 보완하고자 제시되었던 방법론

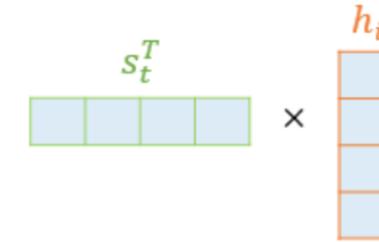
핵심 : 매 시점마다 현재 시점에서 예측해야 할 값과 연관이 있는 입력 부분을 좀 더 집중해서 봄 (전체 시퀀스에 대하여 동일한 비율로 참고하는 것이 아니라, 차별성을 두겠다는 것이 메인 포인트)

1. RNN의 고질적인 문제 기울기 소실 문제

-> RNN에서는 주로 활성화 함수로 Hypervolic tanh를 사용하기에 레이어가 많아질수록 오차 역전파 과정에서 문제가 생긴다.

2. Seq2Seq에서는 입력된 정보를 고정된 길이의 컨텍스트 벡터에 압축하다보니 정보손실이 발생하며, 시퀀스가 길어질수록 더욱 심해진다.

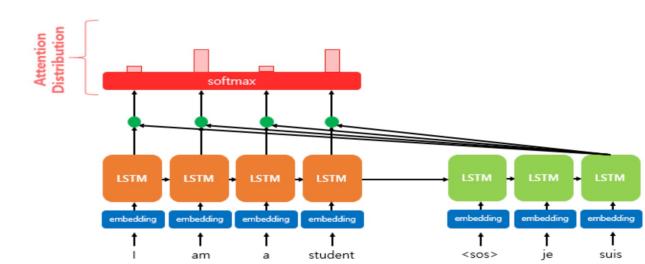
Attention Score



$$score(s_t, h_i) = s_t^T h_i$$

$$e^t = [s_t^T h_1, \dots, s_t^T h_N]$$

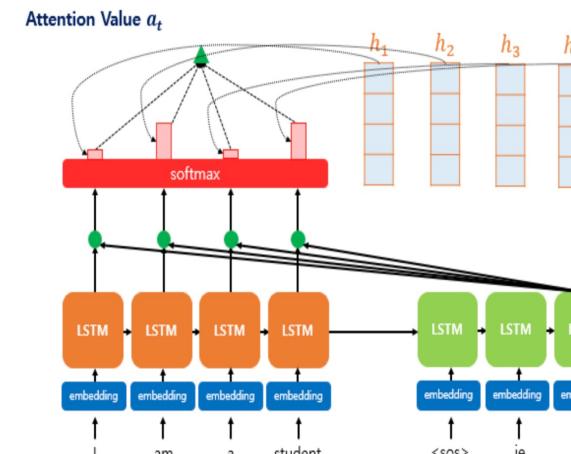
Attention Distribution



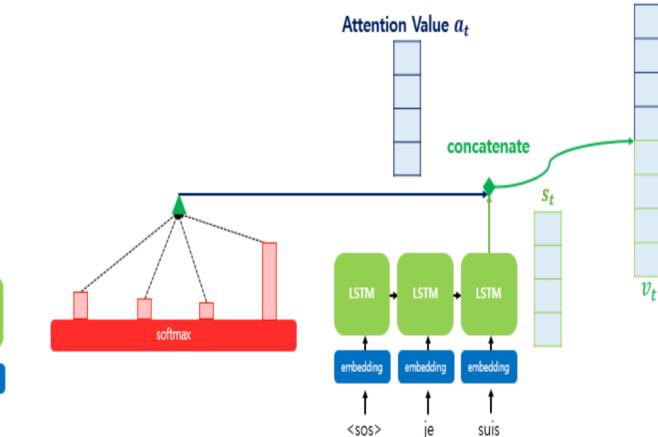
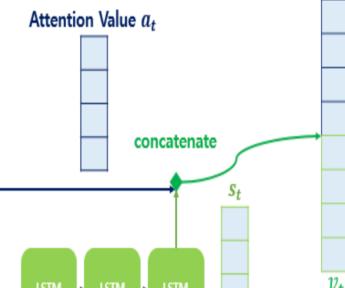
$$\alpha^t = softmax(e^t)$$

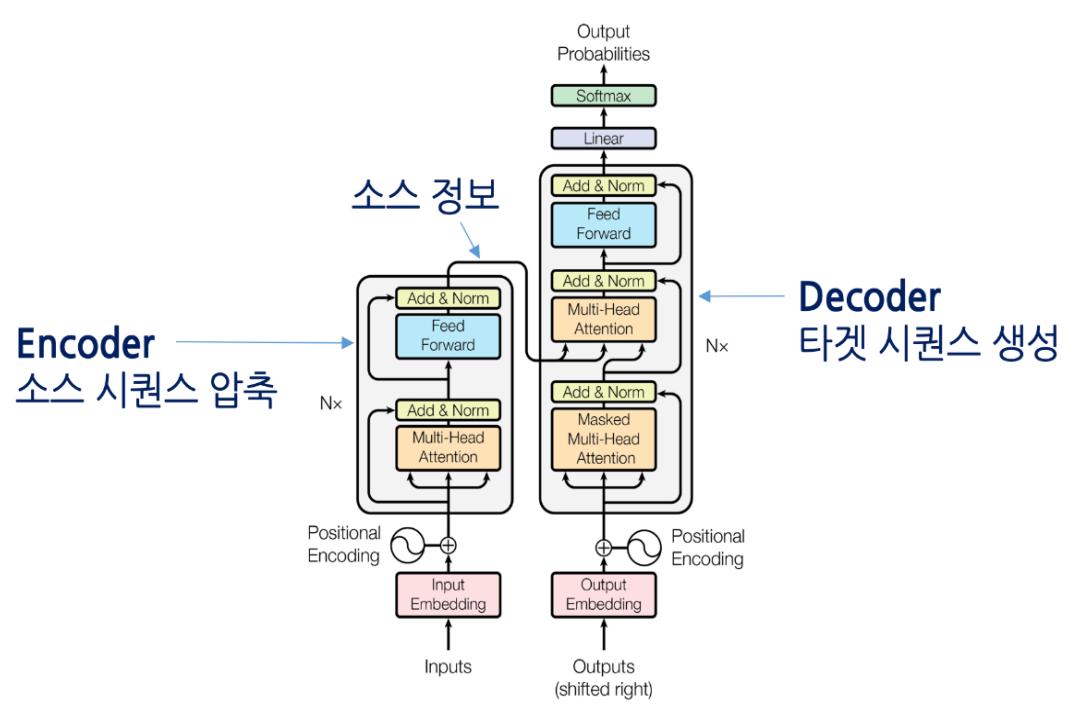
모든 값을 합하면 1이 되는 확률 분포  
(각각의 값은 Attention Weights)

Attention Value  $a_t = \sum_{i=1}^N \alpha_i^t h_i$



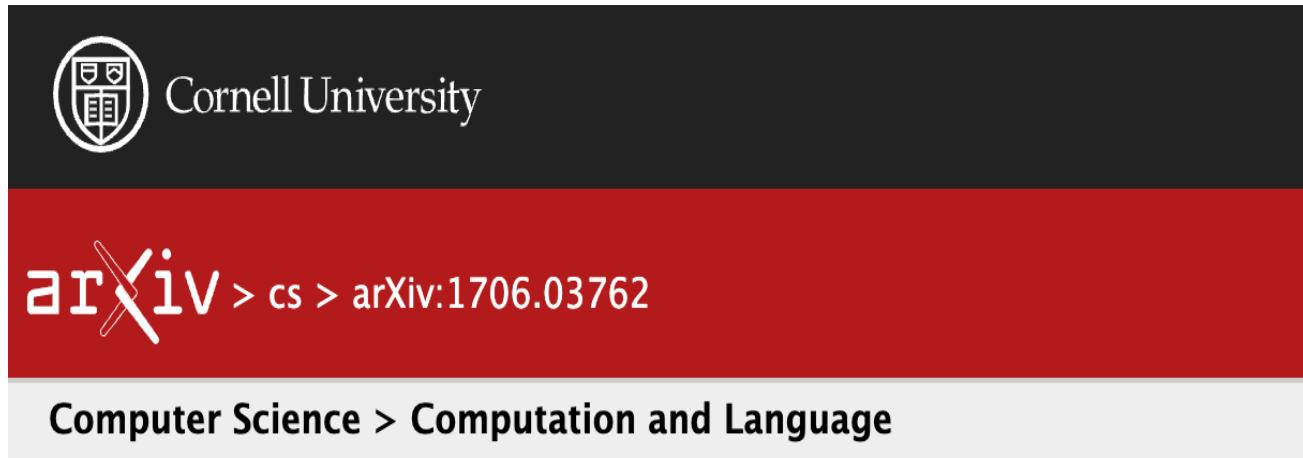
어텐션 값과 디코더의 t 시점의 은닉 상태를 연결한다.(Concatenate)





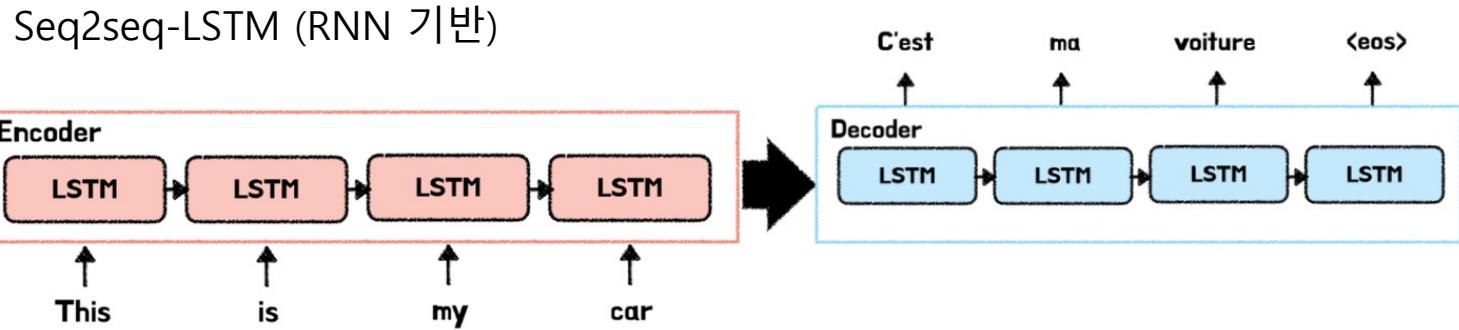
# Transformer

- 2017년 Arxiv에 공개된 “Attention is all you need” 논문에서 소개된 모델
- 초기에는 Machine Translate Task에서 주로 활용되었으나, 현재는 더욱 발전되어 NLP 대부분의 Task에서 SOTA (현재 최고수준의 성능) Vision을 포함한 여러 분야에서 높은 성능을 보여주고 있음

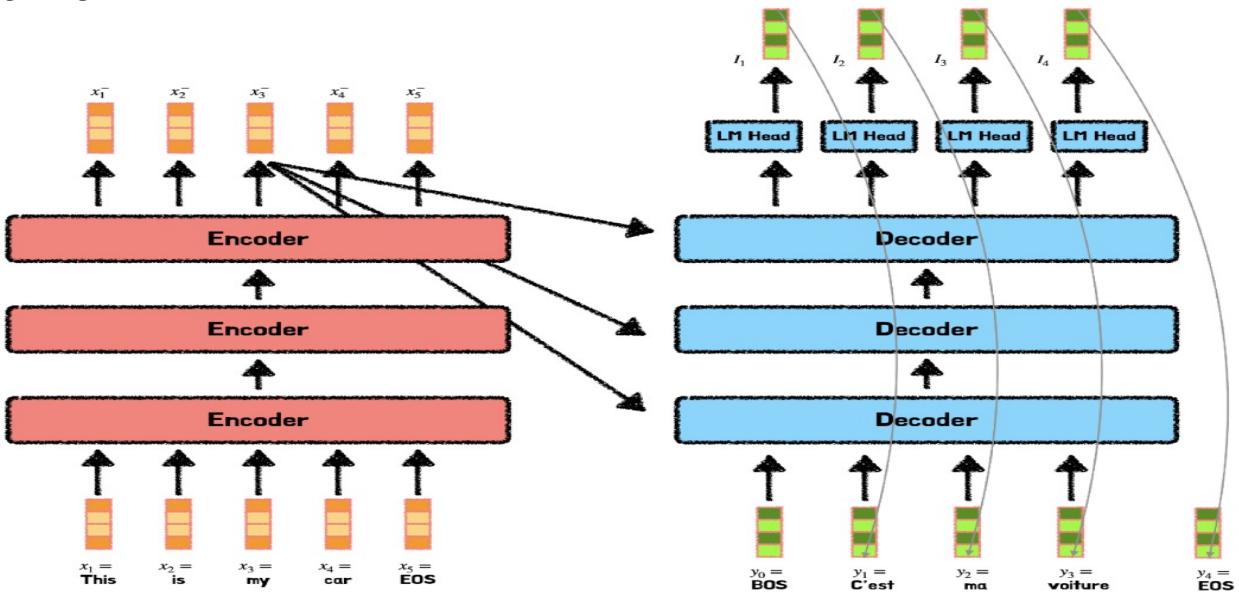


## Attention Is All You Need

# Transformer 특징



Transformer



순차적으로 데이터를 입력받아 처리하는 RNN, LSTM과 다르게,  
입력 문장을 병렬로 한번에 처리한다는 특징을 가지고 있음

# Input Embedding

Sentence      This    is    my    car

Vocab ( a aaron ... car ... is ... my ... this ... zombie )



Indices ( 0    1    ... 3412 ... 5281 ... 6899 ... 8678 ... 9999 )

Vocab indices  
= Input

$X_0$        $X_1$        $X_2$        $X_3$

8678    5381    6899    3412

1. 문장을 구성하는 각각의 단어는 그에 상응하는 인덱스 값에 매칭 -> input Embedding에 전달

$X_3$        $X_1$        $X_2$        $X_0$

a aaron ... car ... is ... my ... this ... zombie

0    1    ... 3412 ... 5281 ... 6899 ... 8678 ... 9999

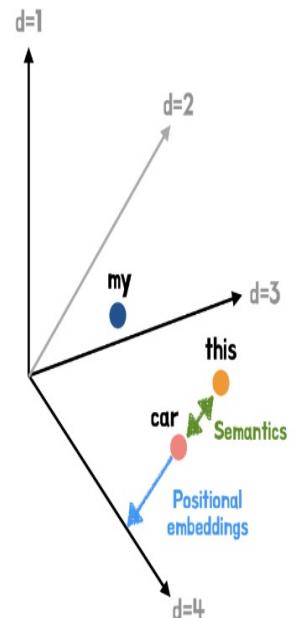


this	is	my	car
0.19	0.01	0.34	0.69
-0.47	0.01	0.87	0.79
-0.77	0.13	-0.39	-0.25
0.59	0.02	-0.91	0.44
0.04	0.23	0.07	0.15



that	is	not	[PAD]
0.19	0.01	0.74	0.00
-0.41	0.01	0.76	0.00
0.12	0.13	0.13	0.00
0.59	0.02	0.23	0.02
0.04	0.24	0.07	0.15

this	is	my	car
0.19	1	0.70	30
-0.47	1	-0.65	100
-0.77	1	0.11	300
0.59	1	0.04	100
			300



# 위치벡터를 더할 때 지켜야 할 규칙

1. 모든 위치값은 시퀀스의 길이나 Input에 관계없이 동일한 식별자를 가져야한다.

-> 따라서 시퀀스가 변경되더라도 위치 임베딩은 동일하게 유지

2. 모든 위치값은 너무 크면 안된다.

-> 위치 값이 너무 크면,

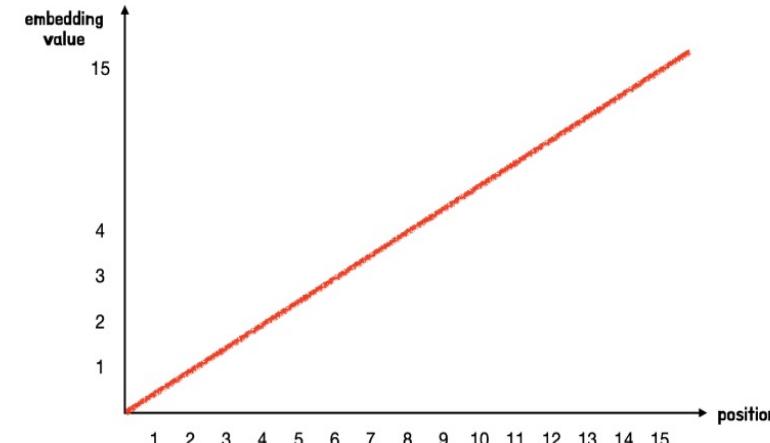
단어 간의 상관관계 및 의미를 유추할 수 있는 의미정보 값이 상대적으로 작아지기 때문에

Attention Layer에서 제대로 학습이 이루어지지 않는다.

## 위치 벡터를 얻는 2가지 방법&문제점

### 1. 시퀀스 크기에 비례하여 일정하게 커지는 정수값 부여

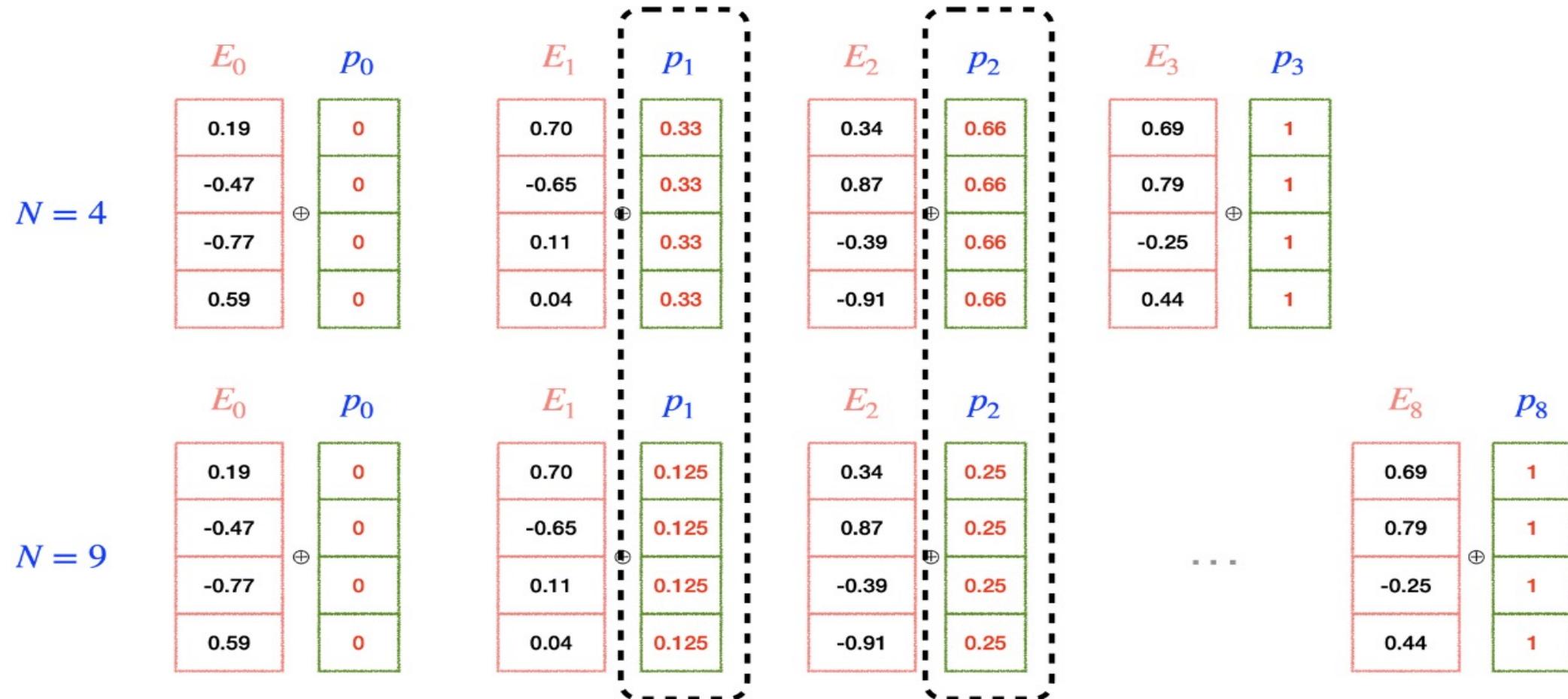
$E_0$	$p_0$	$E_1$	$p_1$	$E_2$	$p_2$	$E_3$	$p_3$	$E_{14}$	$p_{14}$
0.19	1	0.70	2	0.34	3	0.69	4	0.00	15
-0.47	1	-0.65	2	0.87	3	0.79	4	0.81	15
-0.77	1	0.11	2	-0.39	3	-0.25	4	0.31	15
0.59	1	0.04	2	-0.91	3	0.44	4	0.15	15



#### 문제점

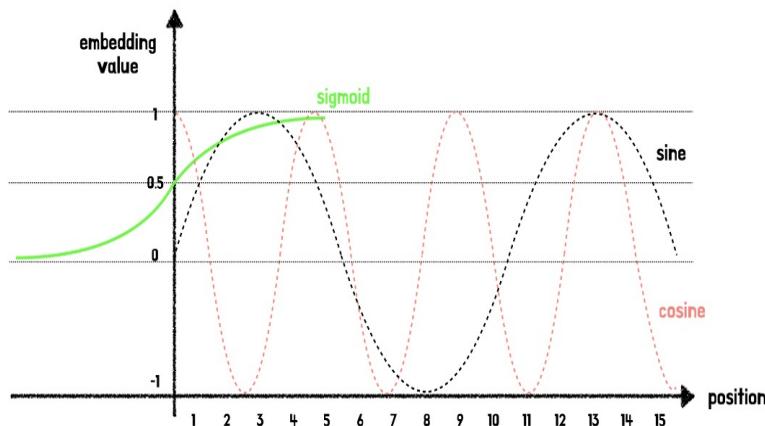
: 위치 정보 값이 급격하게 커지면 단어 벡터와 더했을 때, 단어보다 위치 정보가 지배적이라 단어의 의미가 훼손될 수 있으며 해당 문제는 시퀀스의 길이가 길어질수록 극대화됨.

## 2. 첫번째 토근 0, 마지막 토근 1 부여 후 그 사이를 1/단어 수로 나누어 나온 값에 대해 Normalization(정규화)



문제점 : 시퀀스 길이에 따라 같은 위치 정보에 해당하는 위치 벡터값이 달라질 수 있고, 시퀀스의 총 길이도 알 수 없다.

# Positional Encoding



$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

- Sine & Cosine 함수 활용하여 해결

- Sine & Cosine 함수는  $-1 \sim 1$  사이를 반복하는 주기함수이기 때문에 값이 너무 커지지 않는 조건을 만족시킨다.

- Sigmoid의 경우, 긴 문장의 시퀀스가 주어질 경우 위치 벡터값의 차가 미미해지는 문제가 발생할 수 있다.

하지만, Sine & Cosine 함수의 경우,  $-1 \sim 1$  사이를 주기적으로 반복하기 때문에 긴 문장의 시퀀스가 주어지더라도 위치 벡터값의 차가 작지 않게 된다.

- $-1 \sim 1$  사이를 반복하는 주기함수이기 때문에 토큰들의 위치 벡터값이 같은 경우가 생길 수 있다.

3번 문제를 해결하기 위하여 다양한 주기의 Sine & Cosine 함수를 동시에 사용한다.

Positional Encoding은 스칼라 값이 아닌 단어 벡터와 같은 차원의 벡터값이기 때문에, 만약 하나의 위치벡터가 4개의 차원으로 표현된다면, 각 요소는 서로 다른 4개의 주기를 갖게 되기 때문에 서로 겹치지 않음

## Attention vs Self- Attention

### Attention 예시

Who is the singer?

**They love a song by Kodaline, a singer from their hometown, Ireland**

Attention을 통해 주목해야 할

특정 몇몇 단어에 더 높은 가중치를 부과하여 집중하면 얻고자 하는 답을 빠르게 얻을 수 있음.

### Self-Attention 예시

**I saw her tear the paper and shed a tear for her lost love**

Self-Attention은 같은 문장 내에서 단어들 간의 관계, 즉 연관성을 고려하여 어텐션을 계산하는 방법

이처럼 Self-Attention의 경우 동일한 문장 내 토큰들 간의 유사도를 토대로 어텐션을 계산한다.

# Multi-head Attention

Which **do** you like better, coffee or tea?

- 문장 타입에 집중하는 어텐션

Which **do** **you** like better, **coffee** or **tea**?

- 명사에 집중하는 어텐션

Which do **you** **like** better, **coffee** or **tea**?

- 관계에 집중하는 어텐션

Which do **you** **like** **better**, coffee or tea?

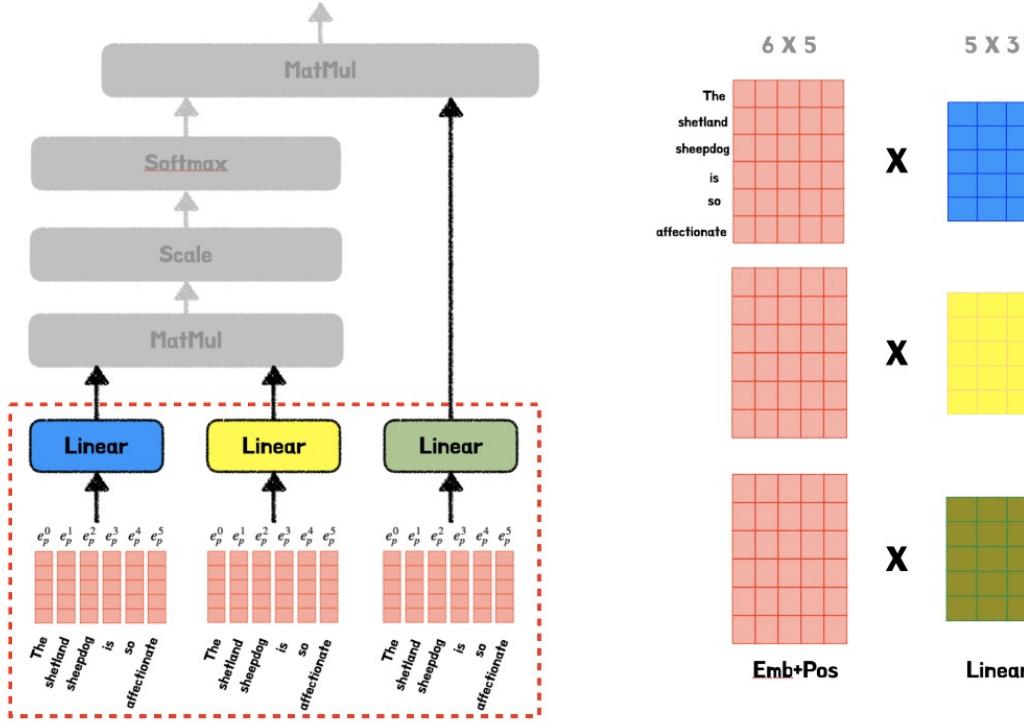
- 강조에 집중하는 어텐션

## • Transformer 기술의 핵심

Self-Attention을 h번 수행하여.

같은 문장 내 여러 관계, 다양한 소스 정보를  
나타내는 정보에 집중하는 어텐션 기법

# Linear Layer



각각의 Linear Layer에는 동일한 Embedding Vector + Positional Encoding 값이 입력으로 들어간다

## Linear Layer 연산을 거치는 이유 2가지

1. Linear Layer가 입력을 출력으로 매핑하는 역할 수행
2. Linear Layer가 행렬이나 벡터의 차원을 바꿔주는 역할 수행

즉 Query, Key, Value 각각의 차원을 줄여서 병렬 연산에 적합한 구조로 만들기 위함.

- Query : 입력 시퀀스에서 관련 정보를 찾으려고 하는 Vector Source (디코더 현재상태)
- Key : Query와 비교하는데 사용되는 Target Vector (소스 문장의 인코더 표현)
- Value : 특정 Key에 해당하는 입력 시퀀스의 정보로 가중치를 구하는데 사용되는 벡터

# Preparation

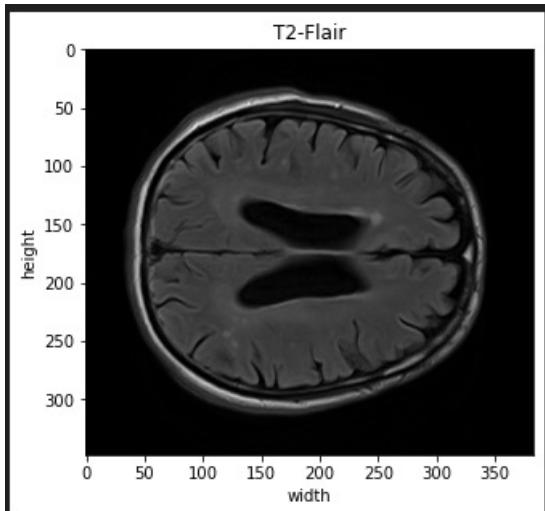
The 2021 WHO Classification of Tumors of the Central Nervous System with Glioma Patients (N=1139)

Model : SE-RESNEXT50 (Internal: \_\_\_, External:TCGA)

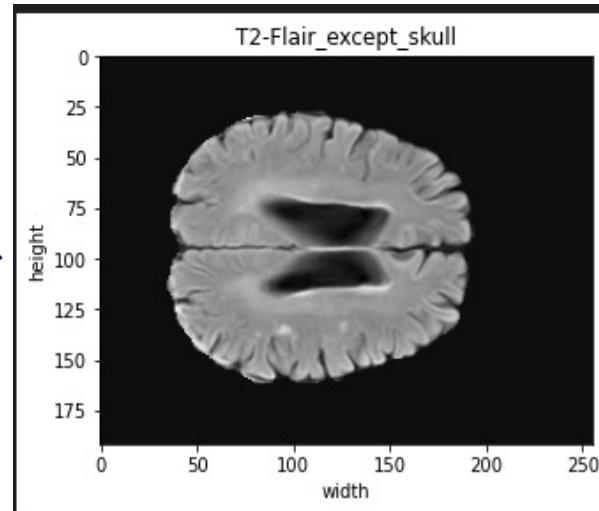
1. Duration:OS, Pathology:all (train=1025, valid=114, test=160)
2. Duration:OS, Pathology:GBL (train=586, valid=66, test=61)
3. Duration:1yr, Pathology:all (train=1025, valid=114, test=160)
4. Duration:1yr, Pathology:GBL (train=586, valid=66, test=61)

Model : ResNet50-CBAM (Internal: \_\_\_, External:TCGA)

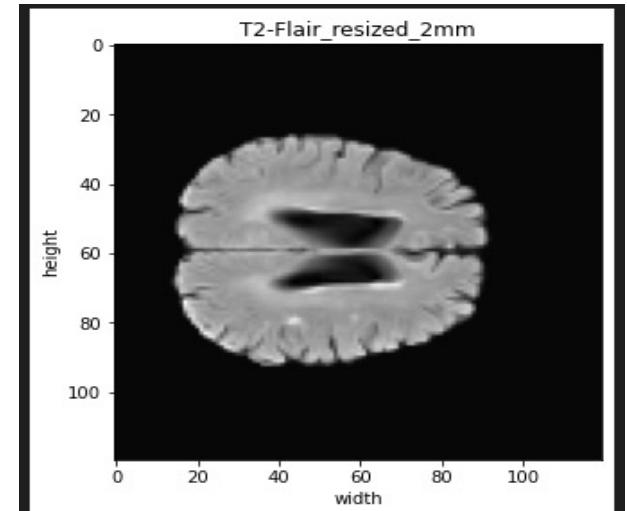
1. Duration:OS, Pathology:all (train=1025, valid=114, test=160)
2. Duration:OS, Pathology:GBL (train=586, valid=66, test=61)
3. Duration:1yr, Pathology:all (train=1025, valid=114, test=160)
4. Duration:1yr, Pathology:GBL (train=586, valid=66, test=61)



(348, 384, 26)



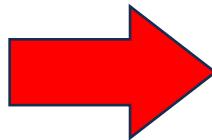
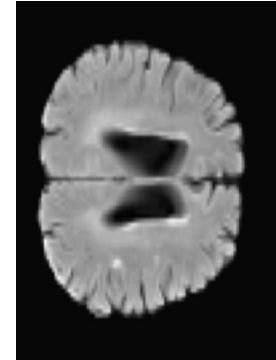
(192, 256, 256)



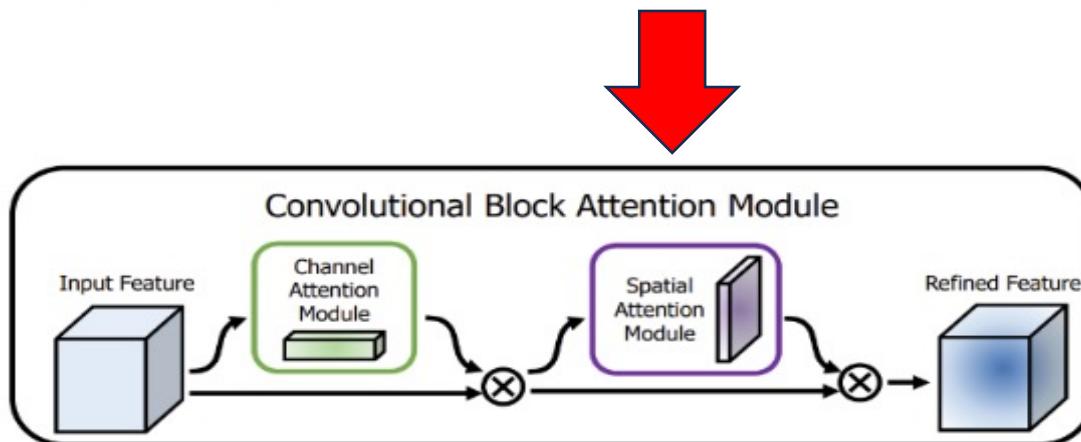
(120, 120, 78)

# Architecture

Resnet50-CBAM

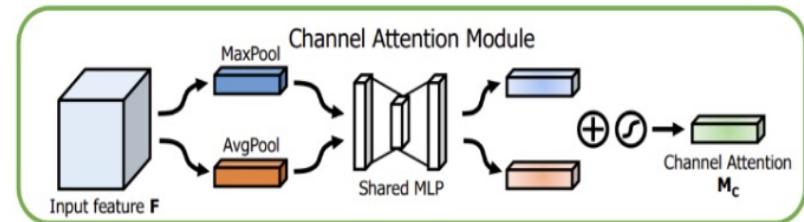


		18-layer	34-layer	50-layer	101-layer	152-layer	ILSVRC 2015
layer name	output size						
conv1	112×112						
		$7 \times 7, 64, \text{stride } 2$					
conv2_x	56×56	$\left[ \begin{smallmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{smallmatrix} \right] \times 3$	$\left[ \begin{smallmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{smallmatrix} \right] \times 3$	$\left[ \begin{smallmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{smallmatrix} \right] \times 3$	$\left[ \begin{smallmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{smallmatrix} \right] \times 3$	
		$3 \times 3 \text{ max pool, stride } 2$					
conv3_x	28×28	$\left[ \begin{smallmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{smallmatrix} \right] \times 4$	$\left[ \begin{smallmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{smallmatrix} \right] \times 4$	$\left[ \begin{smallmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{smallmatrix} \right] \times 4$	$\left[ \begin{smallmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{smallmatrix} \right] \times 8$	
conv4_x	14×14	$\left[ \begin{smallmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{smallmatrix} \right] \times 6$	$\left[ \begin{smallmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{smallmatrix} \right] \times 6$	$\left[ \begin{smallmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{smallmatrix} \right] \times 23$	$\left[ \begin{smallmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{smallmatrix} \right] \times 36$	
conv5_x	7×7	$\left[ \begin{smallmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{smallmatrix} \right] \times 3$	$\left[ \begin{smallmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{smallmatrix} \right] \times 3$	$\left[ \begin{smallmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{smallmatrix} \right] \times 3$	$\left[ \begin{smallmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{smallmatrix} \right] \times 3$	
		average pool, 1000-d fc, softmax					
	FLOPs	$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$	



- 두 가지 어텐션 모듈 추가 (Channel Attention, Spatial Attention)  
-> Intermediate Feature (중간특징)을 효과적으로 강조 및 억제

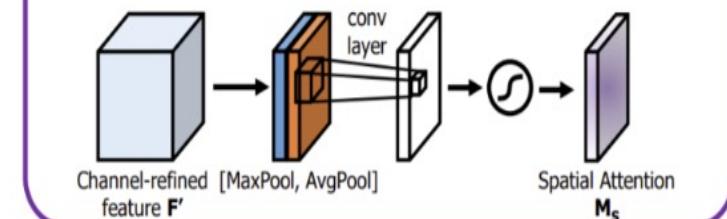
Channel Attention Module



$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(W_1(W_0(F_{avg}^c)) + W_1(W_0(F_{max}^c))), \end{aligned} \quad (2)$$

where  $\sigma$  denotes the sigmoid function,  $W_0 \in \mathbb{R}^{C/r \times C}$ , and  $W_1 \in \mathbb{R}^{C \times C/r}$ . Note that the MLP weights,  $W_0$  and  $W_1$ , are shared for both inputs and the ReLU activation function is followed by  $W_0$ .

Spatial Attention Module



$$\begin{aligned} M_s(F) &= \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7 \times 7}([F_{avg}^s; F_{max}^s])), \end{aligned} \quad (3)$$

where  $\sigma$  denotes the sigmoid function and  $f^{7 \times 7}$  represents a convolution operation with the filter size of  $7 \times 7$ .

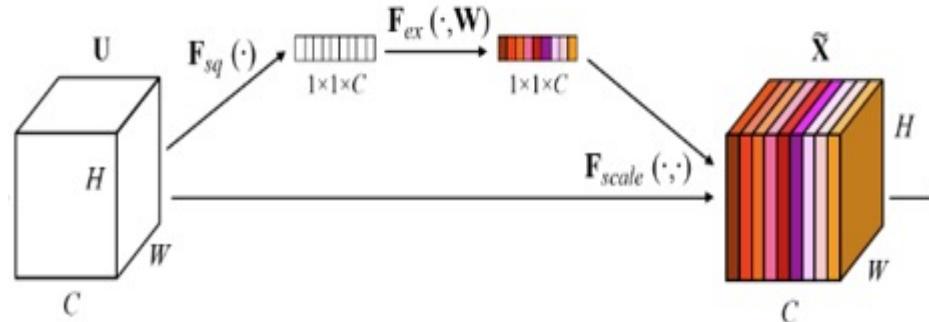
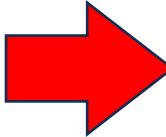
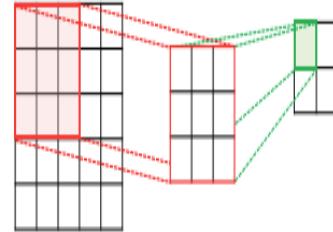
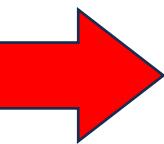
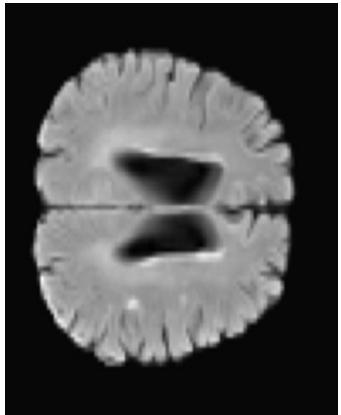
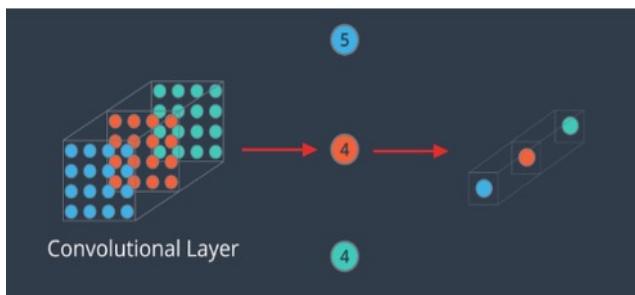


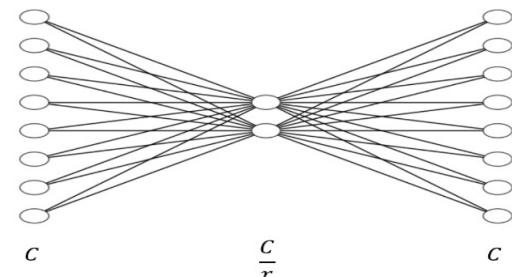
Figure 1: A Squeeze-and-Excitation block.

**Squeeze**

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j)$$

**Excitation**

$$s = F_{ex}(z, W) = \sigma(W_2 \delta(W_1 z))$$



논문에서 저자들은  
채널 간 관계성을 파악하기 위한 방법으로  
의도적으로 채널을  $r$ 만큼 수축시킨 후  
ReLU를 활용하여 채널간의 관계를 살폈음

- 어느 네트워크에도 바로 부착 가능
- 연산량 증가 대비 모델 성능 향상도가 매우 큼

계산 복잡도 결과 (0.26% 증가) (SE-NET 논문 인용)  
Resnet 50 : ~3.86 GFLOPs  
SE-Resnet 50 : ~3.87 GFLOPs

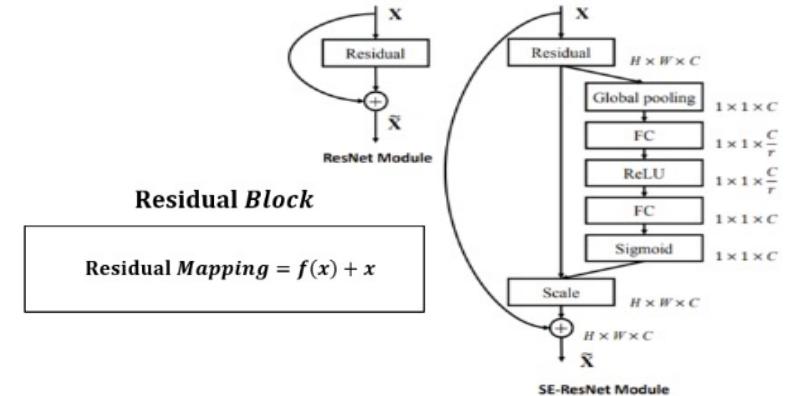


Figure 3: The schema of the original Residual module (left) and the SE-ResNet module (right).

# DenseNet-121

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112		7 × 7 conv, stride 2		
Pooling	56 × 56		3 × 3 max pool, stride 2		
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56		1 × 1 conv		
	28 × 28		2 × 2 average pool, stride 2		
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28		1 × 1 conv		
	14 × 14		2 × 2 average pool, stride 2		
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14		1 × 1 conv		
	7 × 7		2 × 2 average pool, stride 2		
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1		7 × 7 global average pool		1000D fully-connected, softmax

Table 1: DenseNet architectures for ImageNet. The growth rate for all the networks is  $k = 32$ . Note that each “conv” layer shown in the table corresponds the sequence BN-ReLU-Conv.

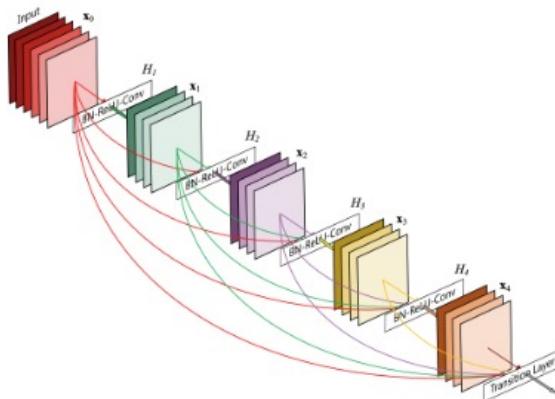


Figure 1: A 5-layer dense block with a growth rate of  $k = 4$ . Each layer takes all preceding feature-maps as input.

## ResNet, DenseNet 차이점

### Resnet

-> Summation (Feature) → Axis=Channel

### DenseNet

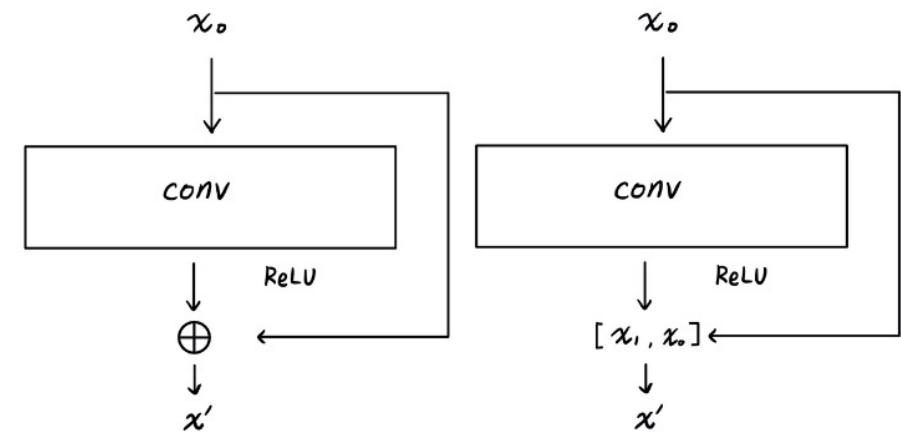
-> Concatenation (Feature) → Axis = Channel

### ResNet의 경우 Summatation 적용

채널 정보와 이미지의 공간적인 정보가 혼합되어 정보 간의 균형을 맞추는 개념

### DenseNet의 경우 Concatenation 적용

채널 정보가 뒤섞이는 혼란을 피할 수 있게 해준다.



$$x' = x_0 + x_i$$

$$x' = \text{cat } [x_0, x_i]$$

# Performance

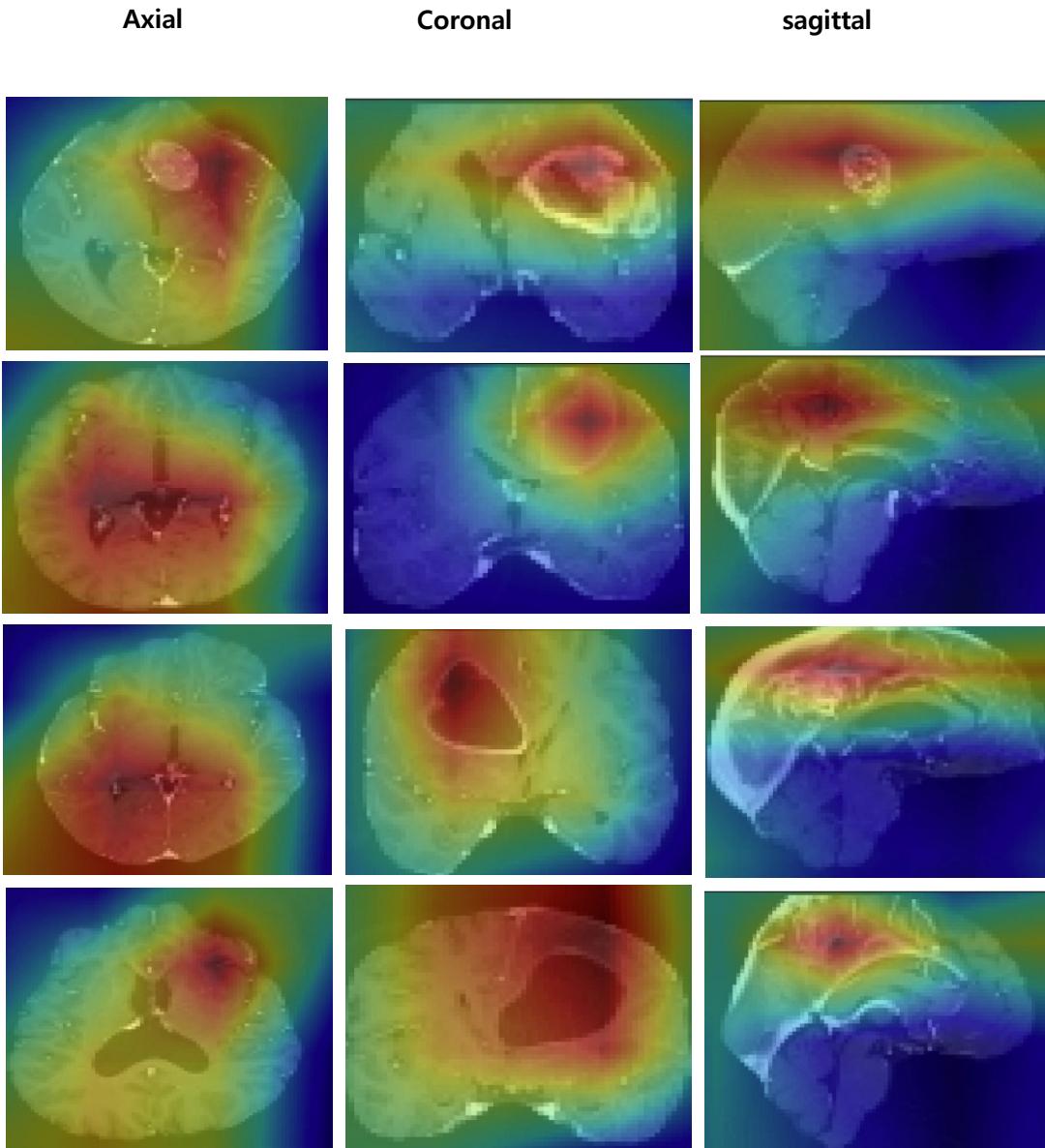
- 95% Confidence Interval for C-index

epochs=200  
learning rate = 0.0001  
weight\_decay=1e-06  
batch\_size = 64

	SE-ResNext50				ResNet50-CBAM			
	OS,all	OS,GBL	1yr,all	1yr,GBL	OS,all	OS,GBL	1yr,all	1yr,GBL
C-index	<b>0.733</b> [0.7124, 0.7540]	0.551 [0.5174, 0.5807]	0.606 [0.5665, 0.6459]	0.55 [0.5073, 0.5918]	<b>0.763</b> [0.7430, 0.7818]	0.625 [0.5952, 0.6528]	<b>0.7961</b> [0.7676, 0.8255]	0.65 [0.6066, 0.6909]
Brier Score	0.148	0.201	0.164	0.199	0.147	0.204	0.14	0.213

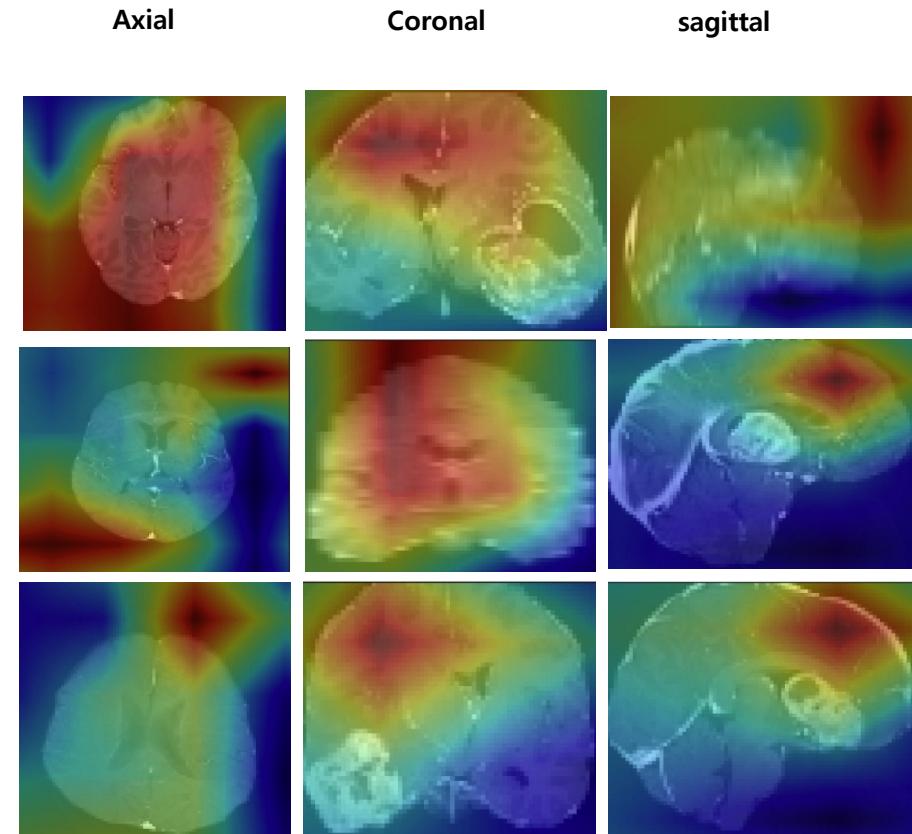
# Results

Best



grad-CAM (ResNet-CBAM (1yr,ALL), C-index = **0.72** [0.625, 0.8122], Brier Score = **0.148**)

misclassified



# Glioma classification framework based on SE-ResNeXt network and its optimization

Jiang Linqi, Ning Chunyu✉, Li Jingyang

First published: 19 November 2021 | <https://doi.org/10.1049/ijpr2.12374> | Citations: 2

## 2.4 Proposed optimization strategies

### 2.4.1 The MultiStepLR strategy

It is well known that the learning rate is the most important hyperparameter to tune for training CNN, and it will directly affect the learning ability and training process of the network. Too large learning rate may lead to the training algorithm diverge. On the contrary, too small learning rate will inhibit the network's learning ability and make the training algorithm converge slowly. Even if the training algorithm reaches the convergence state with a single learning rate during training, it will still fluctuate within a larger range of the optimal value. However, when the learning rate decays with the increase of epoch, the fluctuation range of the training algorithm will be smaller. Therefore, the MultiStepLR strategy is used to realize the dynamic adjustment of the learning rate.

When the specified epoch is reached, the learning rate will decay linearly. The formula is as follows:

$$\alpha' = \alpha \times \gamma^{[\text{milestones}]} \quad (1)$$

where  $[\text{milestones}]$  is the specified epoch,  $\gamma$  is 0.1.

Learning rate가 CNN 학습시키기 위해 튜닝해야 하는 가장 중요한 하이퍼파라미터라고 언급  
너무크면 훈련 알고리즘 발산, 너무 작으면 억제시킴.

Epoch 증가에 따라  
LR을 감소시키면 훈련 알고리즘 변동 범위가 작아진다고 함  
기본적으로  $lr=1e-n$  equation에 맞춰서 적용하라고 명시

## 하이퍼파라미터 튜닝을 위한 높은 성능을 기록한 해외 논문 탐색

### 2.4.2 The label smoothing strategy

Label smoothing is a regularization method. It is usually used to prevent the network from becoming over-confident [29]. The mechanism of its implementation is mainly by modifying the ground-truth label distribution  $q(k) = \delta_{k,y}$ . The ground-truth label distribution of the sample with label smoothing is replaced by

$$q'(k) = (1 - \varepsilon) \delta_{k,y} + \varepsilon u(k) \quad (2)$$

where  $u(k)$  is independent of the training sample  $x$  and usually taken as the uniform distribution  $u(k) = \frac{1}{K}$ , and  $\varepsilon$  is the smoothing parameter. The cross-entropy loss is as follow:

$$\begin{aligned} L(q', p) &= -\sum_{k=1}^K \log(p(k) q'(k)) \\ &= (1 - \varepsilon) L(q, p) + \varepsilon L(u, p) \end{aligned} \quad (3)$$

Therefore, it is equivalent to adding a penalty item to the cross-entropy loss. When  $u(k) = \frac{1}{K}$ ,  $L(u, p)$  represents the degree of deviation between the predicted distribution  $p(k)$  and  $u(k)$ .

By smoothing the ground-truth label distribution, the dependence of the network on the ground-truth label is reduced. When the features learned by the network are not enough to distinguish between positive and negative samples, the label smoothing strategy could effectively control the direction of network learning. The smoothing parameter  $\varepsilon$  is assigned 0.1 in this article.

Positive Samples와 Negative Samples 사이에서 구별하는것이  
충분하지 않는 네트워크에서 피쳐가 학습되었다면  
Label Smoothing 전략이 효과적으로 네트워크 학습 방향을 조정할 수 있다.

Label Smoothing 높을 때 효과

1. 모델이 훈련 데이터에 과도하게 의존하지 않도록 하며 **일반화 성능 향상**
2. **모델 훈련 수렴 속도 증가**
3. **예측의 불확실성에 대해 더욱 잘 처리**할 수 있게 한다.

Label Smoothing 낮을 때 효과

1. 모델이 훈련 데이터에 더욱 집중하여 **훈련 데이터에 맞춰 성능 향상** (그러나 노이즈에 민감)
2. 모델이 **확실한 case**에 대해서는 정확한 예측을 수행하지만, **불확실한 경우에는 취약**
3. 더 **많은 epochs**를 필요로 한다. (그러나 정확도는 향상될 수 있다.)

# Discussion

1. 환자 별 grad-CAM 2d-plot을 확인한 결과,  
상대적으로 Axial,Sagittal (가로면,시상면)보다 **Coronal (관상면)**에서  
Tumor를 잘 인식하여 클래스를 잘 분류하는 것을 알 수 있었음.
2. 사용자 관점에서 초기에 사용할 때,  
새로운 Dataset을 활용한 "CustomNet"의 Inference 성능이 좋지 않아  
전처리 파이프라인을 거친 데이터 **중간값들을 지속적으로 확인 후 수정**하였고  
**모델 성능이 개선**되었음.
3. Only GBL에 비하여,  
GBL (WHO Grade 4), High grade glioma (WHO Grade 3), Low grade glioma (WHO  
Grade 1,2) 데이터가 전부 주어진 **모델의 성능이 상당히 높음**
4. 향후 모델 성능 향상 방안으로  
IDH 유전자 변이 여부에 따른 **추가적인 뇌교종 타입 정보**를 제공할 예정임.  
(Diffuse glioma, Oligodendro glioma, Astrocytoma, OligoAstrocytoma, GBL)

# 향후 개인 연구

- 가제 : 3D Vision Transformer – AutoEncoder를 활용한  
교모세포종 환자 생존분석 및 예측(개인  
연구)
- Vision Transformer (비전 변환기)  
: 2020년 "An image is worth 16x16 words"  
논문에 의해  
이미지를 단어처럼 처리하면 어떨까 하는  
아이디어에서 시작된 연구로  
분야에 따라 합성곱 신경망과 비슷한 성능  
을 내고 있음.

