# 1. [실습] 아래의 링크를 참조하여 Convolutional Neural Network 실습을 따라해 봅시다 (Day1)

김종호(스터디진행자) [1 hour ago]

1번 질문에서 어려움을 겪으시는 부분은 특히 Input\_Shape에 관한 부분인 것 같습니다. 최명환님께서 질문을 주신 내용인데, FNN, CNN, RNN을 모두 다루다 보니, 각각의 경우에서의 input\_shape를 까다뤄워 하시는 것 같습니다. 떄문에 각각에 대해 정리해 드리자면

Fully Connected Network인 Dense Layer에서는 (..., input\_dim) 의 input\_shape을 받습니다. 즉 Dense Layer에서는 input\_shape로 (x, y)를 입력하면, 마지막 axis인 y에만 dense가 적용이 됩니다 (각 x의 step에 대해서). 따라서, MNIST 예에 대해서는 모든 pixel에 dense를 적용하기 위해, Flatten을 통해 input\_dim을 28\*28으로 만들어 주었습니다.

CNN에서는 (height, width, color)를 입력으로 주었으며, 일반적인, 이미지 데이터의 regular한 모양은 (Number of image, Height of image, widgth of image, color channel (or output channel))입니다. 따라서 X를 저희가 가지고 있는 이미지데이터라고 하였을 때, X에대해 reshape을 다음과 같이 취해줍니다.

(-1, IMG\_SIZE =100, IMG\_SIZE =100, 1):

모든 이미지에 대해서 (-1) (number), IMG\_SIZE (width), IMG\_SIZE (height), 1 (Grey Single Color)의 모양으로 맞추어 주었습니다.

여기서 -1 부분을 어려워 하시는 분이 많아서, 예를 들어 설명드리자면

z = np.array([[1, 2, 3, 4],

[5, 6, 7, 8],

[9, 10, 11, 12]])

z.shape(3, 4)

이렇게 생긴 변수도

z.reshape(-1)

array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

-1을 하면 flat해 집니다

그러면, 이해하기에는 모든 데이터에 대해 우선 flat하게 만들고 나서 -1, 나머지 데이터의 모양을 IMG\_SIZE =100, IMG\_SIZE =100, 1, 로 만들어 준다고 이해하셔도 됩니다

저희의 경우에는 input shape 부분에서는 X가 reshpe된 이후에 input shape 부분에서는 X.shape[1:] 을 취하게되면, 이전에 모양이 (N, H, W, C) (N: number of image, H: height, W: width, C: color)로 맞춰어져 있었기 때문에 X.shape[1:] 는 (H,W,C) (height, width, color)의 모양을 넘겨 주게 됩니다~

이렇게 하는 이유는 Batch Size 혹은 Number of Sample의 개수는 파라미터로 넘겨주지 않아도, 학습 과정에서 알게되는 정보이고, 나머지 H W C의 모양은 Conv2d에서 요구하는 image의 regular한 모양이기 때문입니다.

반면에 LSTM에서는 input shape로 (timestep, feature)를 입력으로 받습니다.

저희의 예제에서는 이미지의 가로 세로를 각각 (timestep, feature)로 간주하였는데 사실 이미지를 rotate하거나 bidirectional rnn을 써서 하나의 데이터를 4개의 방향으로 학습할 수도 있습니다

1번 문제와 관련한 다음 질문으로 넘어가겠습니다

## 질문 - ppt 15페이지부터 설명되는 1D convolution에 대해 좀 더 자세한 설명부탁드립니다. 갑자기 1D convolution에 대한 설명이 왜 나온건지 궁금합니다.

이러한 질문이 있었는데, 1D Convolution은 Single Feature의 Time Series Data에 적용하기 위해 사용되었습니다. 1d에 관해서는 시간관계상 동영상을 통해 이해를 돕자면

<https://www.youtube.com/watch?v=ulKbLD6BRJA>

위의 동영상을 참조해 보시면 조금 더 쉽게 이해하실 수 있을 듯합니다. 관련하여 보다 자세한 튜토리얼은 아래 링크를 참조해 보시면 좋을것 같습니다.

(<https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf?gi=a7760d2c4c1>)

## 질문 - rnn계열의 비효율성으로 시계열 연구분야 또한 cnn으로 넘어가는 추세입니다만 주가예측 문제에 관해서는 어떻게 생각 하시나요?

해당 질문에 대해서는 상황이 계속해서 바뀌는 것 같습니다. 한 곳에서의 단점이 한해가 지나면, 보완된 방법론으로 해결되거나 새로운 모델로 인해 대채되는 경우도 많은 것 같습니다.

\* cnn의 장점을 보자면 대표적으로 아래와 같은 특징이 있는데

\* finding local patterns

\* good for feature extraction

\* faster (less parameter and parallelizable)

\* rnn 에서는 각각을 아래와 같은 방법으로 해결하며, 어떤 부분에서는 장점을 지니고 있기도 합니다.

\* Local pattern 문제: attention이 해결

\* sequential => parallel SRU 도입(https://arxiv.org/abs/1709.02755)

\* Better remembering context

\* Convolution Neural Networks have no dependence between last and current time step.

개인적으로는 둘 사이의 Compromise로 CNN을 RNN에 붙혀서 씁니다

\* Compromise

\* CNNs can be “attached” to RNNs.

\* We are going to use CNN as encoder for RNN inputs

저희 Module3인가 4에서 다룰 예정이고요. 혹시 여기까지 질문이 있으신가요~?

## 질문 - CNN에서 커널 사이즈는 어떤식으로 정해지게 되나요?

해당 질문에 답변을 드리자면,

한 층에 대해서 Best Kernel Size를 정하기 보다는, 전체 네트워크를 고려했을 때 각 층마다 다양한 Kernel Size를 정하는 편이 실증적으로는 최적인 것으로 알고 있습니다.

또한, Kernel Size에 따라서 추상화 할수 있는 범위의 크기가 달라지는데, Kernel의 개수와 함께 고려해 볼 점이 있습니다. Size가 작더라도 개수가 많으면 보다 큰 Size의 Kernel이 만들어내는 추상화 범위를 가질 수 있습니다.

또한 주어진 문제마다, 다른 Design이 효과적일 수 있고, 일반적으로는 성능을 높히기 위해서, 가장 널리 쓰이는 Design을 실험한 이후에 해당 Accuracy가 원하는 Level인지 보고, 그렇지 않을 경우에 Parameter를 바꿔가면 다양하게 실험해 보는것 같습니다.

아래의 다양한 Parameter에 따른 Performance 차이를 참조해 보시면 좋을 것 같습니다. (https://github.com/ducha-aiki/caffenet-benchmark)

CNN에 관련된 논문에서 사용된 Parameter를 참조 바랍니다. (<https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/>)

이 외에도 하드웨어에 따라서 학습 속도면에서 커널 사이즈를 어떻게 선택할지도 관련이 있는 것으로 알고 있습니다. 따라서 Universal한 design이 있다기 보다는, 주어진 상황과 리소스에 맞춰서 선택해야하는 문제가 있는 것 같습니다.

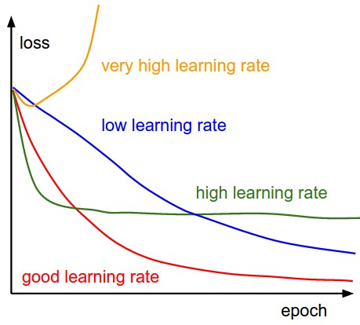
# 3. [실습] 아래의 링크를 참조하여 Tensorboard on Colab 실습을 따라해 봅시다 (Day2)

Tensorboard에서는 어떻게 해석하면 좋은지, 그리고 matplotlib에 비해서 어떤 점이 비교우위가 있는지에 대한 질문이 있었습니다

저는 hyperparameter를 search 하는 과정에서 쓰고 있습니다. 여러 paramter에 대해서 어떤 경우에 가장 accuracy의 converge가 빠른지를 비교하여 모델을 선택하는 용도로 쓰거나, 혹은 복잡한 모델을 만들때는 visualize가 꼭 필요해서 tensorboard를 통해 모델을 그릴때 사용합니다

colab에서는 write 속도가 느려서인지 (?) tensorboard 적용이 느려서 어렵다고 느꼇는데, 만약 다른 서버 환경 혹은 로컬에서 하실때는 훨씬 유용하게 다양한 테스트를 해보실 수 있을것 같습니다

hyperparamter search에 조금 더 예를 들자면



learning rate에 따른 loss 변화를 본다거나, 큰 model에 대해 몇 노드를 그룹으로 묶어 interactive하게 visualize가 가능합니다

<https://www.youtube.com/watch?v=qCNwmUK5YqI>

혹시 관련하여서는 추가적으로 궁금하신 부분이 있으신가요~?

권도형(10월딥러닝일요10시) [1 hour ago]

혹시 tensorboard를 이용하여 하이퍼파라미터를 찾는다는 것이, 그리드서치하고 관련이 있는건가요?

김종호(스터디진행자) [1 hour ago]

넵 맞습니다

권도형(10월딥러닝일요10시) [1 hour ago]

그리스서치에는 시간이 많이 걸리는데, 텐서보드를 사용하면 이점이 있다는 말씀인가요?

김종호(스터디진행자) [1 hour ago]

그런데 tensorboard는 visualize의 용도이고요 그리드서치는 시간이 많이 걸리는 작업이 맞습니다 ㅠㅠ

권도형(10월딥러닝일요10시) [1 hour ago]

아… 그렇군요….

권도형(10월딥러닝일요10시) [1 hour ago]

그래도, 혹시 텐서보드와 그리드서치 관련한 예제가 있다면 나중에라도 알려주시면 감사드리겠습니다

김종호(스터디진행자) [1 hour ago]

넵~!

고재민(10월딥러닝일요10시) [1 hour ago]

colab에서 텐서보드를 위한 작은 서버를 만드는 것처럼 보이는데

고재민(10월딥러닝일요10시) [1 hour ago]

커널이 끝날때까지 쓸 수 있는건가요?

김종호(스터디진행자) [1 hour ago]

사실 저도 ngrok이 정확히 어떻게 작동하는지는 잘 모르겠어서요, 그런데 접속이 엄청 빈번하면 막히는것 같습니다.

개인적으로는 colab에서는 write 속도가 너무 느려서 model을 write하고 tensorboard에 대한 지원은 아직 미흡한것 같습니다

# 4. [실습] 아래의 링크를 참조하여 Model Save, Load, Retrain 실습을 따라해 봅시다 (Day3)

해당 문제에서는 다음과 같은 질문이 있었습니다

"모델저장시 구조 즉 그래프 메타정보 variable 정리가 필요합니다. 연계해서 텐서보드 보는법도요"

keras에서는 model compile 이전에 각 layer에 대해 name을 부여해줄 수 있으며 model.get\_layer(이름)을 통해서 접그할 수 있습니다.

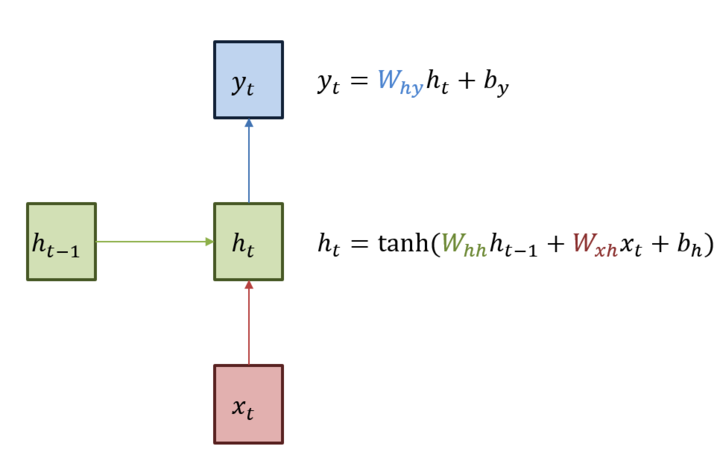
model.summary()를 해보시면 해당 이름이 반영된 모델 shape을 볼수 있고요

시간이 촉박해서 다음 질문으로 넘어가겠습니다

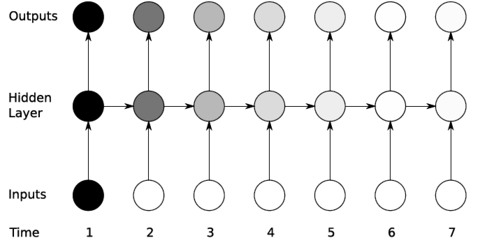
# 5. [실습] 아래의 링크를 참조하여 Recurrent Neural Network 실습을 따라해 봅시다 (Day4)

"RNN 및 seq2seq 모델에 대한 자세한 설명 부탁드립니다."

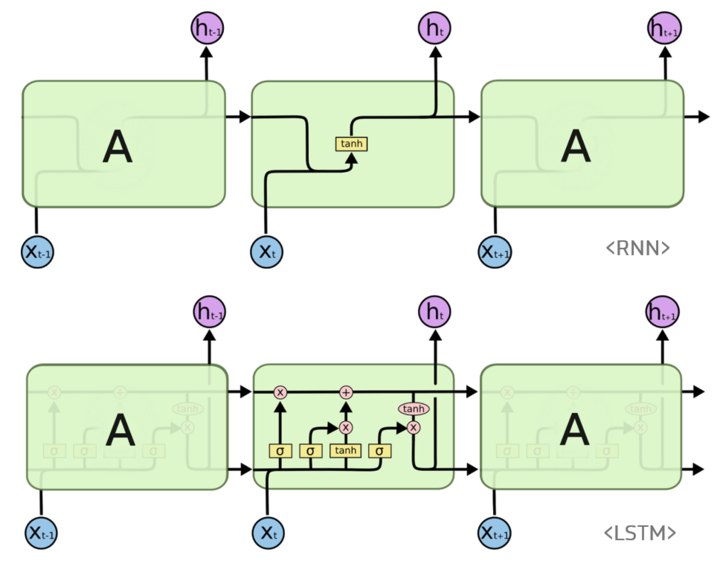
시간관계상 핵심만 간략히 소개드리자면



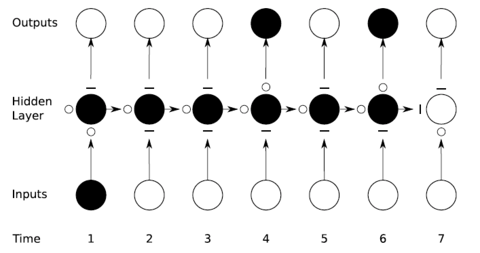
RNN에서는 이전 step의 정보 (hidden state)와 현재 step의 입력값을 input으로 가집니다. 따라서 과거에 대한 메모리를 유지하는것 까지는 좋은데



과거의 입력값과 거리가 멀어질수록 정보가 희석되는, vanishing gradient 문제가 있습니다. 따라서, 이를 해결하기위해



lstm이 도입되었고요, lstm의 각 셀은 독립된 또다른 Neural Network라고 보시면 됩니다. 내부의 forget gate를 통해 언제 정보를 잊을지 기억할지 등을 반영해 주고요. lstm을 통해서는 아래와 같이 vanishing gradient 문제를 조금 더 해결할 수 있습니다



혹시 여기까지 질문이 있으신가요~?

권도형(10월딥러닝일요10시) [1 hour ago]

RNN보다는 LSTM이 더 좋다고 받아들여도 무방할까요?

김종호(스터디진행자) [1 hour ago]

넵 맞습니다 추가적으로 LSTM과 GRU가 있는데, GRU는 LSTM의 Simplified된 버전이고 학습이 조금 더 빠르다고 합니다

권도형(10월딥러닝일요10시) [1 hour ago]

감사합니다

최명환(10월딥러닝일요10시) [1 hour ago]

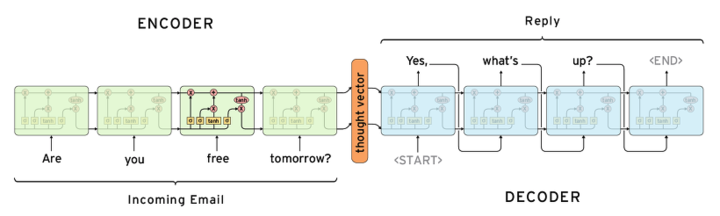
위 설명중에서 어떤부분인 input \_shape에서 feature와 관련된 부분인가요?

김종호(스터디진행자) [1 hour ago]

xt 부분입니다

김종호(스터디진행자) [1 hour ago]

xt에 입력하는 모양이 timestep feature의 모양이어야 하고요. hidden layer의 모양은 latent dimension의 모양을 따라가게됩니다. hidden vector의 모양은 추가적으로 seq2seq에 대해 설명드리자면



구조를 보시면 encoder라고 하는 부분도 lstm이 여러개가 있고, decoder라고 하는 부분도 lstm이 여러개가 쌓아져 있습니다

machine translation과 같이 many to many의 문제를 다룰떄도 쓰이는데요. encoder 부분에서 받은 입력을 하나의 context vector로 압축한 다음에 이를 decoder 부분에서 풀어해치며 하나씩 예측하게됩니다. 지금은 decoder 부분의 결과가 입력과 같은 fixed length를 가지는데, 다양한 변형이 있을수 있고, 저희의 경우에는

input seq: time series data

output: future time series / up/down

등으로 놓고 문제를 풀수도 있습니다

# 6. [과제] Simple RNN for Time Series Prediction (Day 4)간단한 time series 데이터에 대해 LSTM을 적용해 봅시다.

이전 질문이긴 하지만 관련하여

"model.evaluate(X\_test, y\_test, batch\_size=600,verbose=False)의 결과는 Loss인 건가요? accuracy값을 뽑고 싶으면 어떻게 해야하나요..?"

라는 질문이 있었습니다

답변을 드리자면, model.compile 시에 metric을 지정하여 주시면 지정된 metric을 계산하며, 그렇지 않을 경우 Loss가 계산됩니다.

# 7. [실습] 아래의 링크를 참조하여 Preprocessing for Cryptocurrency Data 실습을 따라해 봅시다 (Day5)

해당 문제에 대해서는 특별한 질문은 없었는데 혹시 실습을 진행하시면서 다들 어떠셨나요? 어려운 부분이 있으신가요?

김종호(스터디진행자) [36 minutes ago]

최명환님께는 드려야 할 자료가 많은데.. 제가 최대한 빨리 업데이트 하도록 하겠습니다 ㅎㅎ 전에 질문주신 각 데이터별 popular architecture 부분도 자료를 업데이트 중에 있습니다.

<https://github.com/jonghkim/financial-time-series-prediction/blob/master/Q%26A/General_Question.md>

question3번에서 하나씩 쓰고 있는데

최명환(10월딥러닝일요10시) [35 minutes ago]

감사합니다. 아직 ML에 많이 익숙치 않아서 질문이 많네요.;;

4day RNN예제에서,

X = array([[10, 20, 30], [20, 30, 40], [30, 40, 50], [40, 50, 60]]) #(4,3)

y = array([40, 50, 60, 70])

# reshape from [samples, timesteps] into [samples, timesteps, features]

X = X.reshape((X.shape[0], X.shape[1], 1)) #(4,3,1)

여기서 feature 를 1로 만들어야하는데 아직까지 왜 1로 만들어야하는지 이해가 안됩니다.;

박제권 (10월딥러닝일요10시) [34 minutes ago]

저는 모델을 구성하는 과정은 보면서 이해하려고 하는데… pandas나 numpy 로 matrix 변형시키는 부분들은 계속 어렵더군요. 이건 익숙해져야 하는건지..

한동안 열심히 공부할때는 텐서플로가 받아주는 형식에 맞게 잘 넣었었는데, 다시 보려니 또 헷갈리더군요.

최명환님 질문과 비슷하군요.

김종호(스터디진행자) [34 minutes ago]

lstm에서 input은 (timesteps, features인데) timesteps까지는 있는데 features 모양에 맞는 값이 없기도하고, 1 dimensional data여서 1이 들어갔습니다. 새로 만들어야할때는 data demensional도 영향이 있군요.

넵 이 부분이 ML에서 제일 어려운것 같습니다. 처음에도 어렵지만, 나중에도 shape을 맞추는 과정에서 고생을 많이 합니다. 지금은 그나마 input 단에서의 문제인데 나중에는 layer 사이의 shape이 문제가 됩니다. matrix끼리의 연산이다보니

최명환(10월딥러닝일요10시) [32 minutes ago]

CNN과 RNN결합시 같은 결합시인가봐요?

김종호(스터디진행자) [32 minutes ago]

항상 shape이 잘 맞아야하는것 같습니다

김종호(스터디진행자) [31 minutes ago]

그떄는 사실 상대적으로 간단한것 같습니다

최명환(10월딥러닝일요10시) [31 minutes ago]

정의가 어렵네요. 관련자료를 더욱 보면좋을듯합니다.

박제권 (10월딥러닝일요10시) [31 minutes ago]

음… 계속해서 실수하기 쉬운 부분이라고 동영상강의에서도 얘기하던 기억이 나는데.. 그런거군요.. ㅜㅜ

김종호(스터디진행자) [30 minutes ago]

예를들어서, inner layer 들의 값으로adversarial하게 penalty를 줄떄도 inner layer의 input과 output dimension을 맞추기 위해 reshape을 취해줍니다

또는 위에서 소개드린, seq2seq에서 decoder단에 teacher forcing으로 정답 값을 넣을수 있는데, 최근의 style transfer 논문들에서는 해당 부분에 새로운 embedding 값을 concat해서 넣는데 이런 부분도 reshape이 필요하고요. 끊이 없이 따라다니는것 같습니다. 디버깅도 꾸준히 같이 하면서 과연 내가 원하는 모양대로 들어갔는지도 잘 확인해 보셔야하고요