Mempute Machine Learnnig Framework Suit

Mempute Machine Learnnig Framework Suit는 Pattern Chain Database와 Neural Network Framework 두개의 파트로 구성되있다.

Pattern Chain Database는 빈도수에 기반하여 패턴을 발견하고 베이지안 확률 추론을 한다.

Pattern Chain Database는 메모노드라는 분산 퓨전 관계형 데이터베이스위에서 구성된다. 모든 발견 패턴은 베이터베이스에 저장되고 추론에 사용된다. 메모노드 데이터베이스에 관한 내용은 www.memonode.com을 참고하고 언급된 모든 예제들은 깃허브https://github.com/mempute 에 있으며 패턴체인 에 관한 예제는 깃허브에 mempute_pattern_chain_database_example.ipynb에 있다.

Pattern Chain Database는 데이터에서 빈도수에 의해 규칙을 발견하고이를 패턴화한다. 패턴들은 서로 경쟁하고 가장 높은 강도의 것이 invoke되어 추론에 사용된다.

패턴체인은 데이터 전처리 과정이 거의 필요없다. 잡음은 내부 패턴 발견 과정에서 걸려진다. 선형 데이터의 경우 단지 스케일 맞춤을 위해 표준화 정도만 필요할 뿐이다. 이마저도 스키마 구성에서 단위를 설정하여 주면 필요없다. 스키마는 선형뿐 아니라 이산 심볼릭 데이터도 그대로 입력하면 된다. 또한 이들을 복합 스키마로 구성하여 패턴화 할 수 있다

패턴사슬은 노이즈에 강하다. 반례로 신경이 있다. 신경망은 전처리 과정에서 필터링 하지 못한 약간의 노이즈에도 학습이 튀기 일수 이다. 잘 정제된 입력과 타겟이 정상성이 매우 높은 정합된 데이터에 한하여 파라미터들은 제대로 정렬되어 진다.

반면 패턴사슬은 잡음이 내부에서 필터링될 뿐아니라 입력과 타겟쌍을 정성들여 구성할 필요가 없다. 패턴사슬은 목표값을 타겟팅 하기위해 목표값과의 오차로부터 역전파로 파라미터 오차를 조정하는 연역방식이 아니다. 데이터로 부터 라벨없이 빈도수에 의해 데이터를 가장 잘 설명하는 패턴을 발견하게 되며 이 과정은 귀납적이다. 목표값과의 연결 역시 가장 연결 강도가 높은 연결을 스스로 발견하고 사전 학습하므로써 추론시에 사후 예측한다. 이 모든 과정은 귀납식이다.

입력과 목표를 잘 정제하여 연결 짓는 것은 문제 해결을 위한 작업의 90% 이상을 이미 해결한 것이며 신경망은 나머지를 수행할 뿐이다.

따라서 신경망으로 수행하는 영역은 극히 제한적이고 이는 신경망이 여태까지 광범위한 영역에서 범용적으로 실용성있게 사용되지 못하는 이유이다. 패턴사슬은 시간대 이던 이벤트 이던 그룹 정도의 연관성이면 된다. 우리는 많은 경우에 입력에 잘 정합되는 목표값을 알지 못하며 이럴때 패턴사슬은 주어진 입력에 가장 가능성 높은 목표를 사후확률로 예측해 낸다. 연역방식인 신경망과 달리 귀납방식으로 추론함으로써 과적합의 위험없이 정확한 추론이 가능하다.

이러한 기능성으로 패턴사슬은 예측, 분류, 타겟 라벨의 자동생성, 데이터 정상성 판별, 오류 데이터 필터링 등 빅데이터 처리의 모든 광범위한 영역에서 신경망뿐 아니라 이제까지 발명되어진 어떠한 데이터 처리 기술보다 다변적 기능을 범용적으로 수행할 수 있다.

타겟 라벨 생성은 높은 레벨의 패턴 추상화로 이루어 진다. 신경망 이나 기타 기술은 이러한 기능이 없으며 신경망은 잠재코드의 차원을 작게 줄이면 정보는 뭉게져 의미없어 진다. 언어 모델에서 발전된 트랜스포머는 압축을 수행하지 않으며 컨볼루션 망은 압축을 위해 많은 정보가 손실된다. 깃허브에 이상데이터 판별 예제에서 패턴체인으로 생성한 타겟 라벨으로 신경망과 하이브리드 학습을 진행한 결과를 보면 높은 수준의 제한된 라벨은 학습되어 판별 효과가 있었으나 더 넓은 범위의 낮은 레벨의 라벨은 목표값으로 학습되지 못하였다. 신경망은 매우 제한된 높은 정합성을 갖은 타겟 범주의 학습만 가능할 뿐이다. 언어의 경우 **one hot**으로 분류되어야 할 어휘 사이즈가 **10000**개도 쉽게 넘을 수 있고 신경망의 언어 모델은 이를 달성하나 언어라는 것은 문법도 있고 의미적으로도 우리는 정형된 패턴으로 사용하기에 가능한 것이다. 개나 고양이 분류에서 보듯 이러한 분류들도 또한 매우 정확한 타겟이다. 다만 **mempute** 신경망 파트의 제너릭 망으로는 생성된 라벨로 하이브리드 학습이 되는데 학습시간이 오래 걸리는 단점이 있다. 사전학습은 인코더를 생성하는 과정인데 각종 모델에서 학습 결과에서 인코더 파트만을 사전학습 인코더로 채택하나 정보는 디코더 부분에 더 많아 입력 정보의 반쪽이 안되는 낮은 수준의 인코더를 획득한다. 반면 패턴체인의 추상화된 라벨은 단일 혹은 몇 개 차원 정도로 입력에 대한 분류 태그를 제공하여 이를 타겟으로 인코더 학습시킬 경우 더 높은 사전학습 결과를 얻을 수 있을 것이다.

패턴사슬은 비지도 학습, 타겟 연결 학습, 사전학습, 전이학습 및 보상에 따른 패턴 강도를 조정으로 강화학습등 다양한 학습을 수행 할 수 있다.

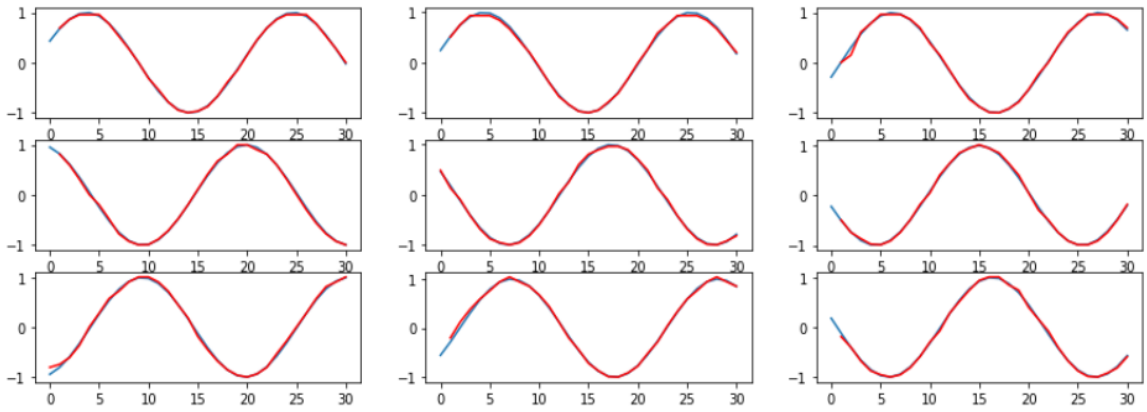
패턴사슬은 학습의 결과로 발견된 패턴을 연결 사슬의 직접적인 형태로 데이버테이스에 저장하고 계속적인 추가학습으로 인해 패턴이 데이터베이스에 누적됨에 따라 점점 더 정확하고 오류에 강한 추론이 가능하다. 이는 통계에서 표본이 클수록 모집단과 유사해져 예측 결과가 정확해 지는 것과 같은 이유이다

패턴사슬은 고성능의 대용량 분산 관계형 데이터베이스위에서 구동되어 대량의 패턴 학습 및 추론이 가능하며 분산 분할 학습과 패턴 병합을 하여 대량의 패턴 처리를 빠른 시간에 수행할 수 있다..

패턴사슬은 다양한 분야에서 범용적으로 사용될 수 있을뿐만 아니라 이상 탐지나 개인화 추천 시스템등 일반적으로 기존 기계학습 방법으로 효과가 적거나 결과가 애매한 분야에 탁월한 성능을 기대 할 수 있고 추론 결과에 대한 확률 해석 설명을 제공할 수 있다.

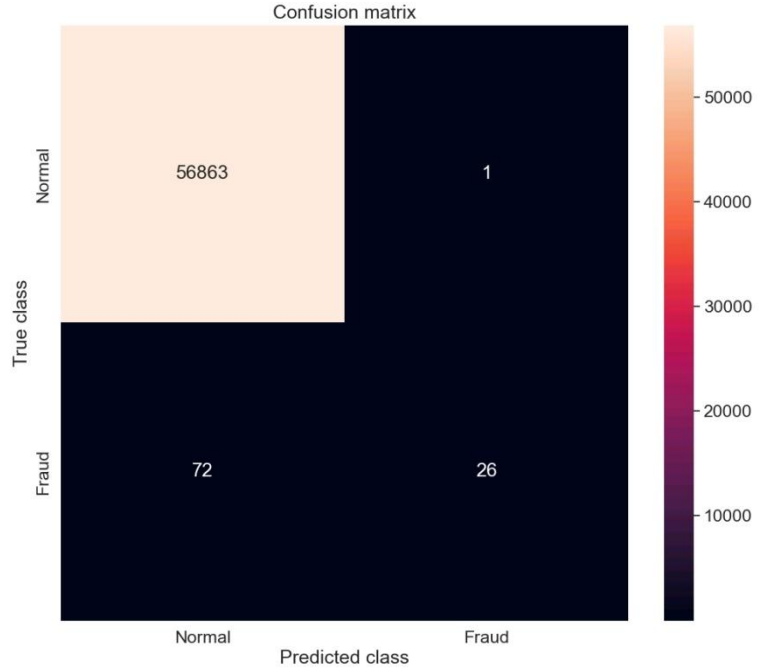
간단한 패턴 체인 데이터베이스 적용 예

사인 곡선 예측

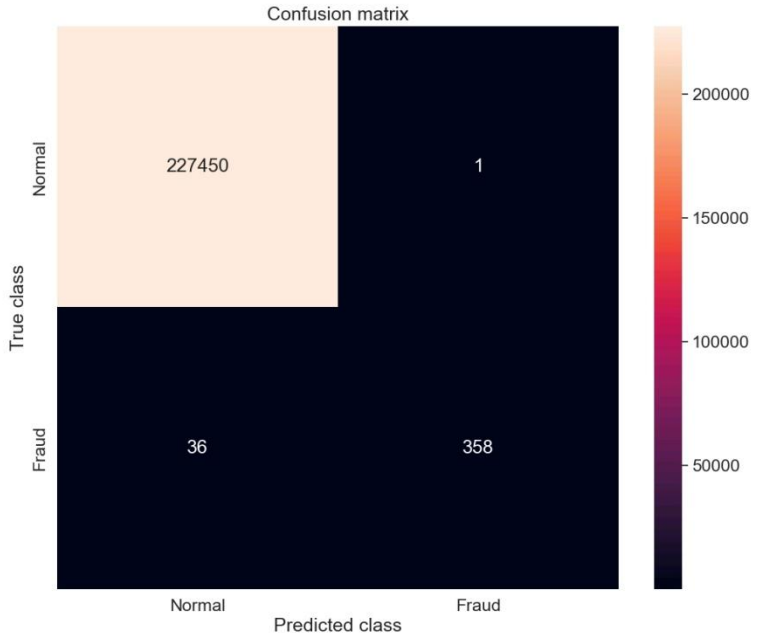


아마존 주가 예측

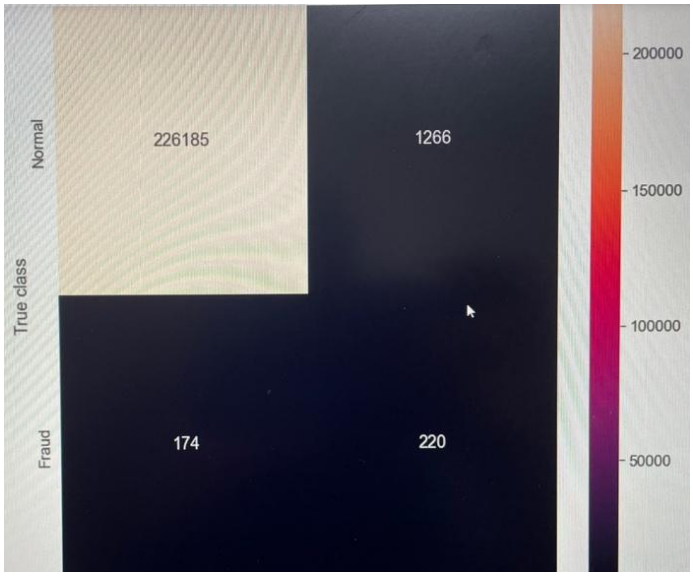
이상 탐지 테스트 데이터 예측



이상 탐지 학습 데이터 예측



신경망 학습 데이터 예측



Anormal detection(이상 탐지)는 비정상 데이터가 적어 신경망으로는 지도 학습을 하지 못하고 오토 인코더로 학습하는데 테스트 데이터 예측도 정확도가 낮을뿐 아니라 학습데이터를 예측한 결과도 마찬가지이다. 신경망의 목적이 일반화에 있어 당연한 결과인데 이상 탐지의 경우 정상과 비정상의 데이터 비율이 비대칭이어서 학습 데이터가 축적되어도 최총 정확도가 향상될 수 없다.

학습 데이터를 예측한다는 것이 생소하겟으나 패턴체인 데이터베이스는 일반화 뿐만 아니라 위 학습 데이터 예측 결과에서 보듯이 패턴 데이터가 데이터베이스에 누적될 수 록 정확도가 지속적으로 향상된다.

위 이상탐지의 예를 살펴보면 차원 축소 압축이 2만분의 1로 된 것으로 압축은 일종의 일반화 효과가 있고 학습데이터 예측이 이 정도 높은 일반화로도 높은 정확도로 나온 결과이고 테스트 데이터 예측도 같은 압축으로 나온 결과로서 예측되지 못한 비정상 케이스는 학습 데이터에는 없는 패턴이다. 또한 sensitivity 파라미터 값을 높히면 학습데이터의 경우 완전한 예측도 가능하다.

다음은 이상 탐지 예제의 패턴 추론과정의 확률 계산을 보여주는 것으로서 이상인데 정상으로 판별하여 추론이 틀린 경우를 설명한다. prior_v는 목표값이고 0은 정상, 1은 이상 임을 나타내고 이상으로 예측해야하나 posterior(사후확률)을 보면 정상 출력이 -10.810163, 이상 출력이 -18.120785 으로서 정상 목표값의 확률이 더 높아 정상으로 판별한 것을 나타낸다. 여기서 pid는 목표값의 아이디, prior st는 사전확률, likely는 event(사건)하에서 발생되는 prior확률인 가능도이다.

event-prior probable result: pid(4249817) prior_v(0.000000) posterior(-10.810163) likely(-10.809829) prior(-0.000334) prior st(2993) prior ast(2994)

event-prior probable result: pid(4249833) prior_v(1.000000) posterior(-18.120785) likely(-10.116419)

prior(-8.004366) prior st(1) prior ast(2994)

mempute deep learning framework & time series neural network

딥러닝 프레임워크 및 시계열 신경망

mempute는 기계 학습 신경망 프레임워크로서 c++와 파이썬을 지원하고 일반 선형대수 라이브러리에 대하여 그래프를 생성하여 자동미분 역전파 기능을 수행한다. 또한 데이터 수평/수직 분할 알고리즘을 내부적으로 수행하여 병렬 학습 수행 및 분산 수행을 지원한다. 또한 API가 c++와 파이썬이 1 : 1 매칭되어 c++ 환경에서도 쉬운 모델링이 가능하다.
Mempute는 프레임 워크에 시계열 패턴 과 이미지 인식을 수행 할 수 있는 **Dynagen** 과 **Generic** 신경망이 포함되어있다. **Generic**은 시계열 신경망이고 **Dynagen**은 **Generic**망을 이미지 인식으로 확장시킨다.

Impulse는 **dynagen**망을 사용하여 학습 및 추론할때 함께 사용되는데 **dynagen** 망은 학습이 단계적, 계층적으로 수행되므로 입력과 타겟 데이터의 동기화 및 **flow control**을 수행하고 동기식, 비동기식 데이터 입력, 학습, 추론 인터페이스를 지원한다.

dynagen은 일반화된 제너릭망으로 런타임에 동적으로 연결하여 대부분의 목적하는 신경망을 별도의 모델링 없이 바로 수행시킨다. 계층적으로 여러개의 **isle loop**로 구성되며 루프는 하위 망으로서 제너릭망을 로컬 또는 원격에 **invoke**하여 독립적으로 수행시키고 인 아웃을 파이프 시스템으로서 연결하여 여러개의 제너릭 망을 분산 병렬 수행시킨다.

루프의 종류로는 **encoder, decoder, couple isle loop**가 있고 인코더는 입력을 차원 축소하여 압축하고 디코더는 압축된 잠재코드인 인코더의 출력을 입력으로 하고 인코더의 입력을 타겟으로 디코딩 학습 수행한다. 커플 루프는 인코더 루프의 출력을 입력과 타겟으로 하여 연결 학습을 수행한다.

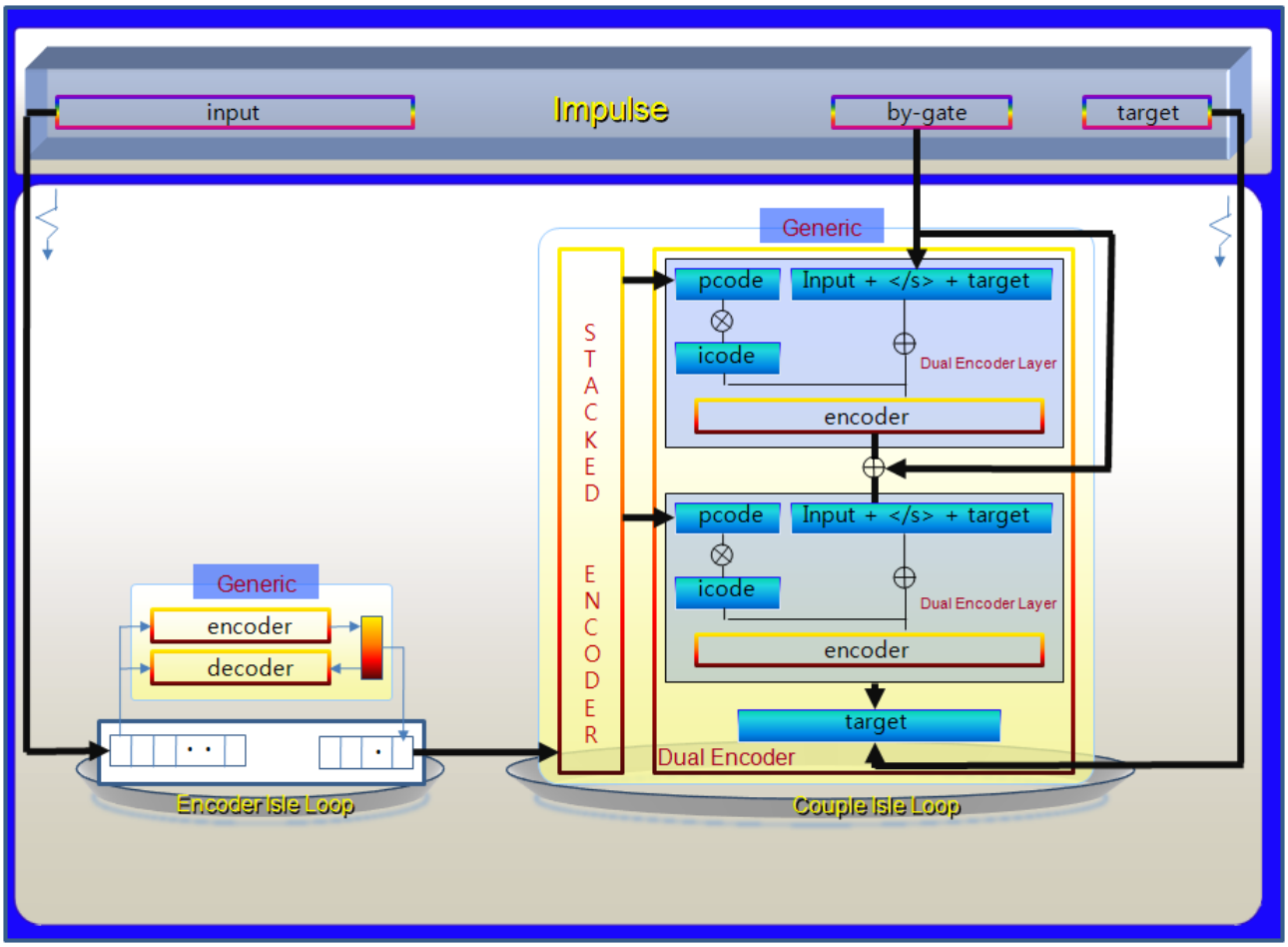
다이나젠 망은 오퍼레이션으로서 **Classification, Translation, Recall, Link**를 지원하고 **long term sequence** 시계열 데이터를 루프를 적층하여 auto encoding 방식으로 압축 추상화를 수행한다.
다이나젠의 **Classification**은 분류 문제를 학습하고 **Translation**는 **seq-to-seq** 연결학습하며 **Recall**은 원본 데이터의 복원이나 합성에 관한 것이다. **Link** 오퍼레이션은 다이나젠 루프의 출력을 연결하여 **Projection (투사)**을 수행하고 데이터 동기화를 수행하는 **impulse**와 함께 단위 신경망인 제너릭을 연결하여 거대 신경망 시스템의 구성을 지원한다.

제너릭망은 신경망의 하위 오퍼레이션을 오토인코더를 구성하여 일반화, 추상화 시키고 패턴을 탐지하고 생성하는 인코딩 기능은 매우 강력하여 세부적 사항의 encapsulation을 지지한다. Impulse나 Dynagen과 상관없이 독립적으로 수행할 수 있으며 자체적으로 적층 인코더와 디코더를 가지고 있고 분류, seq-to-seq연결 학습, 복원, 듀얼 인코더로 auto regression기능을 수행한다.

LTC(Long Term Comprehension)

LTC의 학습 구성 요소는 이전 코드 (pcode), 입력, 타겟 (목표값)이고 pcode는 단위 문맥 정보로서 long term sequence를 제너릭 망 또는 다이나 젠 망에서 차원 축소 압축하여 생성되고 양방향 참조 학습되어 독해력 테스트과 같은 down stream task를 수행 할 수 있게 한다.
입력은 down stream task에서 query로서 수행되고 순방향 참조 학습되며 이때 pcode는 key로서 수행된다. pcode가 없다면 입력에 문맥 정보를 포함 시킬 수 있다.

pre-training은 이전 문장을 입력으로 다음 문장을 예측하는 것으로 수행되어 별도의 라벨 데이터가 필요 없으며 추가적인 전처리가 거의 필요 없어 학습 데이터 확보에 어려움이 없으며 적은 노력으로 대량 학습 시킬 수 있다.
pre-training 과정에서 입력은 이전코드로 이동하여 이전코드 시퀀스 사이즈만큼 누적하여 적재되고 다음 입력과 함께 학습된다. 이와 같은 처리로 LTC는 일관된 문맥 정보를 유지하여 모든 down stream task에서 보다 높은 차원의 자연어 이해 수준을 제공한다.



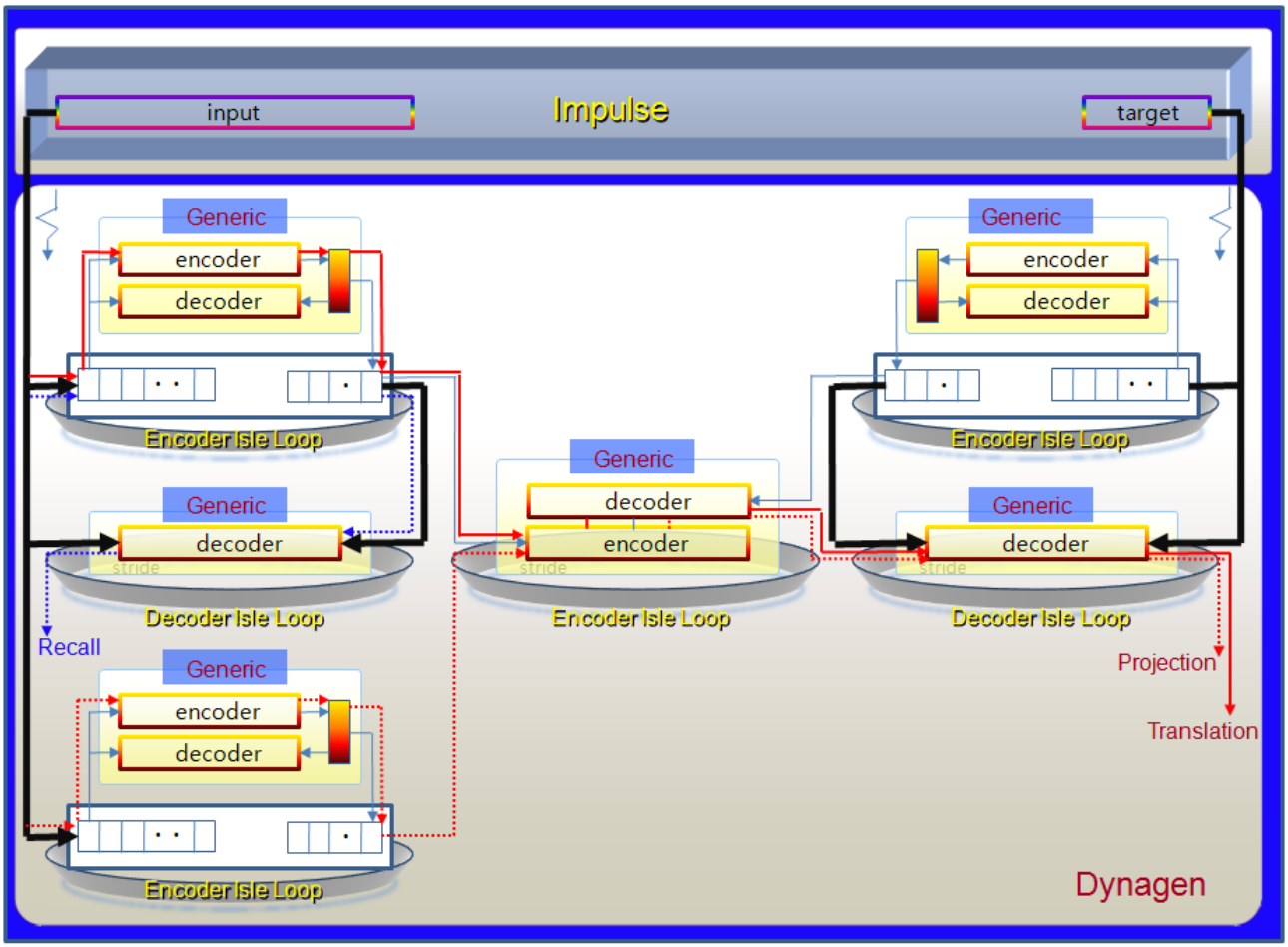
Dynagen Classification Operation & Generic Auto Regression

위 그림은 LTC에서 Dynagen Classification Operation 과 Generic Dual Encoder를 사용하여 seq-to-seq prediction을 수행하는 것을 나타낸다.

Dynagen Classification Operation은 타겟 측이 코드 압축이 없는 asymmetric 학습으로서 주로 분류에 사용되는 것으로 이에 Generic Dual Encoder를 Dynagen couple isle loop에 적재하여 연결한다.

다이나젠의 인코더 루프는 적층하여 구성될 수 있고 제너릭의 인코더와 디코더를 사용하여 최 하위에서 특징 패턴을 오토 인코딩 방식으로 추출한다.

추출된 특징 패턴 스트림은 커플 루프의 제너릭 적층 인코더에서 좀더 고수준의 추상화를 수행 할 수 있다. 이 과정은 생략될 수 도 있으며 이렇게 압축 추상화된 코드는 제너릭 듀얼 인코더의 pcode에 적재되어 by-gate에 입력과 타겟 쌍이 적재된 것 과 함께 다시 한번 인코딩 되어 auto regression방식으로 타겟 스트림을 예측한다. 또한 듀얼 인코더 레이어 단위로 residual하여 많이 반복하면 할수록 더 강력한 예측 성능을 나타낸다.



위 그림은 다이나젠의 나머지 오퍼레이션을 나타낸다. 오퍼레이션의 세부 수행은 모두 제너릭의 인코더와 디코더를 사용하여 수행되는데 이렇게 일반화를 할 수 있는 것은 제너릭 인코더의 강력한 패턴 추출 및 생성 기능 때문에 가능하다.

Translation operation은 입력과 타겟 양측에서 **symmetric** 하게 각각 동시에 계층단위로 패턴 압축 수행되고 최종 압축 패턴이 커플 **isle**에서 연결 학습 하고 추론 과정에서는 빨간색 실선으로 표시된 경로들 따라 추론된다.

Projection은 **learning pair**와는 상관없는 다른 **isle loop**간에 **Link operation**에 의하여 연결되어 미리 학습되지 않은 것에도 예측을 수행하여 다양한 신경망 구성을 가능하게 한다.

특징

Generic 망은 시계열 데이터의 학습 및 추론을 수행한다. 일반적인 시계열망인 **RNN**과 이의 기울기 소실 문제를 개선한 **LSTM**은 여전히 **long sequence**인 경우 계속 곱이 행해지면서 뒤로 가면서 앞쪽

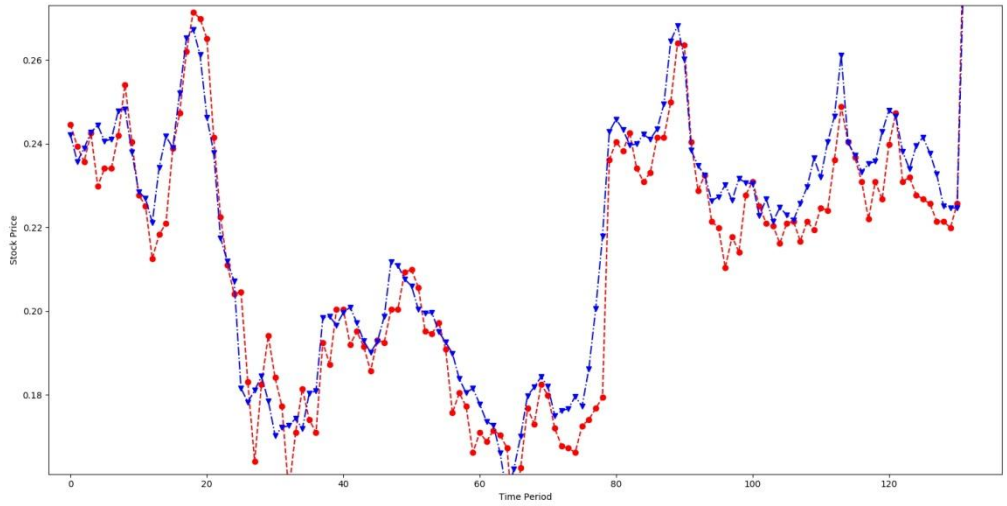
의 정보가 소실되는 문제을 안고 있다. 이를 개선하는 방법으로 요즘 시계열 처리의 대세인 Transfomer계열이 있는데 셀프 어텐션 기법으로 인하여 기울기 소실 문제를 극복하고 어느 위치에서나 어텐션 할 수 있음을 장점으로 내세우지만 계산량이 시퀀스 길이에 비례하여 exponential하게 증가하여 시퀀스 길이에 심각한 제약이 있다. xlnet같은 경우는 이전 문맥 코드를 두기는 하나 매우 shallow하다. 따라서 여전히 long sequence 처리 문제에서 자유롭지 못하다. 또한 여러 실험을 해본 결과 패턴 생성이 RNN계열 만큼 강력하지 못해 정답 시퀀스를 정확히 맞추지 못하는 한계가 있다.

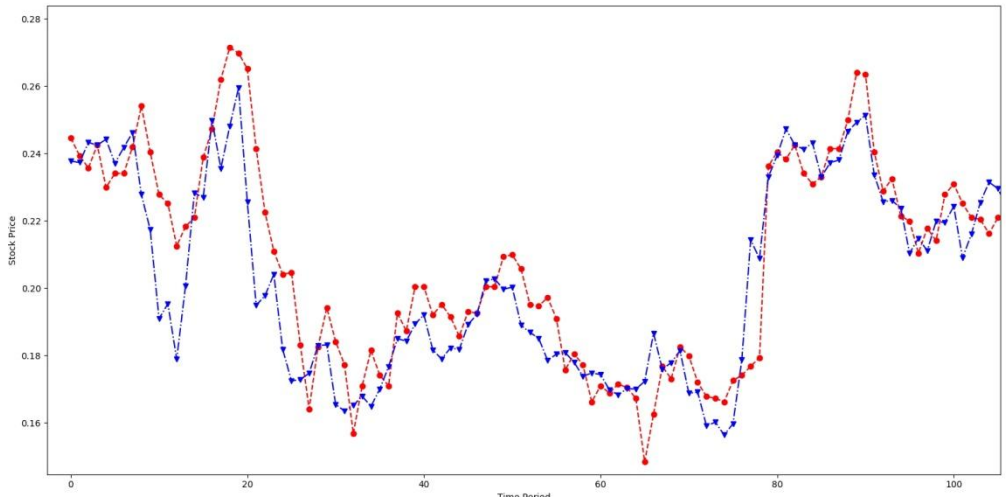
부가하여 이미지 인식 분야 성능을 나타내는 컨볼루션 망이 있으나 특징을 추출하기 위하여 컨볼루션 연산을 수행하는 과정에서 연산 자체와 풀링 레이어에 의하여 정보의 파괴가 심하고 생성 모델을 수행 할 수 없는 것이 단점이다.
이러한 점을 극복하고자 트랜스포머 모델을 이미지 인식에 적용하는 시도가 있으나 결과는 지켜봐야 할 일이다.

이러한 점들에 비춰봤을 때 제너릭 망은 시퀀스 길이에 제약을 받지 않고 강력하게 패턴을 탐지하고 생성한다는 것은 커다란 장점이며 이러한 결과로 보다 더 정확한 시계열 예측을 지원한다. 또한 다이나젠 망과 함께 사용하여 이미지 인식 분야에도 적용하여 정보의 손실 없이 특징을 파악하고 이미지 구성 요소의 관계를 이해하는 인식 결과에 따라서 다양한 어플리케이션과 언어 모델에서와 마찬가지로 Generative Model을 적용 하는 것이 기대된다.
Impulse 와 Dynagen망은 Generic으로 구성된 단위 신경망을 손쉽게 동적으로 연결 확장시켜 대단위 신경망 시스템 구성을 지원하고 후에 Auto ML의 상위 개념으로서 스스로 학습하고 확장되는 베이스를 지지한다.

주가 예측

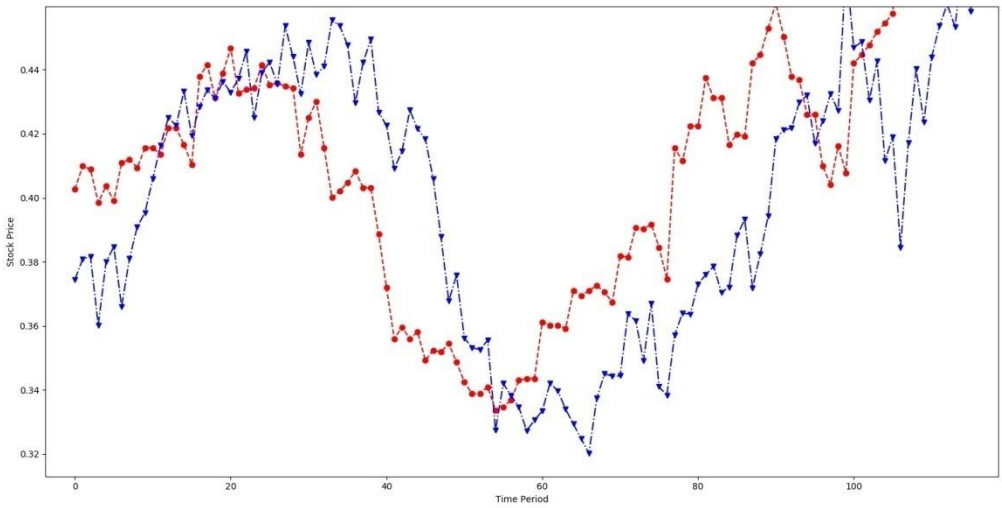
다음은 강력한 패턴 탐지의 근거로서 주가 예측 그래프를 보여준다.
학습은 Dynagen과 Generic망을 함께 사용하여 입력 값을 pcode로 하였다.





위 그림에서 파란색 선은 예측을 빨간색 선은 실제 주가를 나타낸다.

두 개 모두 학습된 데이터를 예측한 것으로서 LSTM이나 기타 시계열 망은 학습한 데이터 조차 위와 같이 예측이 선행되는 모습을 보이지 못하고 예측한 날짜만큼 Lagging된다. 더욱 유의할 점은 위 그림은 목표 주가로 학습하지 않았다는 것이다. 타 4개 내지는 8개 종목의 시가, 종가, 거래량 등을 각 종목별로 입력으로 하고 목표 종목의 6일 앞의 주가를 타겟값으로 하여 예측 한 것이다. 타 시계열 망으로는 6일 앞 예측이면 6일치가 lagging되어 전혀 예측 성능을 보이지 못하고 더군다나 여러 개의 타 종목 주가를 입력으로 한다면 그 상관 패턴 분석은 힘들어 진다.



위 그림은 학습되지 않은 테스트 데이터를 동일한 방법으로 타 종목 주가를 입력으로 하여 목표 종목 주가를 6일치 앞을 예측한 것으로서 초기 대략 40일 정도는 그 이전 까지의 데이터를 학습한 결과로 실제 주가의 패턴을 거의 예측해낸 것을 보여준다. 그 이후는 학습 데이터와 점점 멀어지므로 예측의 정확도는 줄어들 수 밖에 없다.
여기에 주가 관련 각종 지수나 뉴스들을 본 신경망으로 텍스트 마이닝하여 입력 한 다면 더욱 더 정확한 예측 성능을 나타낼 것이 기대된다.

챗봇

다음은 대략 8200개 정도의 대화문을 학습하고 3500여개 대화문을 테스트한 결과의 일부 이고 [Source]가 입력 값, [Truth]가 타겟 값, [Translated]가 에측 값을 나타낸다.

========== train batch ==========

[Source] 너무 많이 먹어서 소화시켜야 하는데 움직이기가 싫어 </s>

[Truth] 소화제 챙겨드세요 </s>

[Translated] 소화제 챙겨드세요 </s>

[Source] 전세비 올려달래 어떡하지 </s>

[Truth] 사정을 잘 설명해보세요 </s>

[Translated] 사정을 잘 설명해보세요 </s>

[Source] 독박 육아 짜증나 </s>

[Truth] 배우자와 대화를 나눠보세요 </s>

[Translated] 배우자와 대화를 나눠보세요 </s>

[Source] 오늘 라면 먹고 싶어 </s>

[Truth] 맛나게 드세요 </s>

[Translated] 맛나게 드세요 </s>

[Source] 약 한달 전 헤어진 그에게 </s>

[Truth] 연락할 생각 하지마세요 </s>

[Translated] 연락할 생각 하지마세요 </s>

epoch: 0 step: 17, train(A): 0.972027, test(B): 0.023392, B-A: -0.948635

ep #: 0 step #: 17

========== test batch ==========

[Source] 연애상담해줬더니 나만 바보됐어 </s>

[Truth] 다음부터는 해주지 마세요 </s>

[Translated] 몸에 어떤 성분이 부족한지 알아보세요 </s>

[Source] 이 사람 없으면 못 살 거 같아 </s>

[Truth] 시간이 지나면 잘살고 있는 당신을 보게 될 거예요 </s>

[Translated] 거리를 두세요 </s>

[Source] 썸 타는 사이인데 카톡하다가 마무리 어떻게 해 </s>

[Truth] 잘자요 내일도 보고싶어요 라고 하는 건 어떨까요 </s>

[Translated] 달라지는 게 없다면 만나지 않는 게 더 나을 수도 있어요 </s>

[Source] 좋아하는 사람한테 적극적인 여자 별로야 </s>

[Truth] 적극적이면 오히려 더 좋을 것 같아요 </s>

[Translated] 저한테 하나씩 싶네요 </s>

[Source] 사랑하는건지 나도 모르겠어 </s>

[Truth] 없어도 살 수 있는지 생각해보세요 </s>

[Translated] 확신이 없나봐요 </s>

학습 데이터 셋을 추론한 결과는 정확도가 대략 80%~100%를 나타내고 테스트 셋은
이 정도 소량의 학습 데이터로는 정확도를 논할 수 없으나 아래 테스트 셋 결과의 마지막을 보면 입력문 "사랑하는건지 나도 모르겠어**" 에 대한 정답이 "** 없어도 살 수 있는지 생각해보세요 **" 인데 예측 값 "**확신이 없나봐요 **" 를 보면 문맥에 맞는 응답을 한 것을 볼 수 있다. 이때의 입력값은 학습 데이터에는 없었던 것이다. 이는 일반화가 달성되어 적은 데이터로 학습한 결과를 가지고도 테스트 데이터 셋 에서도 패턴이 추론될 수 있는 한 최대로 추론됨을 의미한다 하겠다.**
두 가지 예제 모두 인코더로만 구성 된 것으로서 과적합의 여지가 없으며 적은 학습 데이터로도 정확도와 일반화가 모두 만족된 결과를 나타낸다.

트랜스포머 류의 시계열 모델을 학습시켜 본 결과 하이퍼 파라미터를 다르게 하여 여러번 해봐도 이 정도의 데이터로는 학습 셋 조차 거의 대부분 정답 시퀀스를 맞춰 낼 수 없어 정확도를 계산하는 것이 무의미 하였다. 데이터가 많아도 정확도가 올라갈 것이라고 기대되는 학습의 징후는 보여지지 않았다.

Mempute framework 및 위 예제 코드는 깃허브 https://github.com/mempute