DA for MTS forecasting

1. Paper List

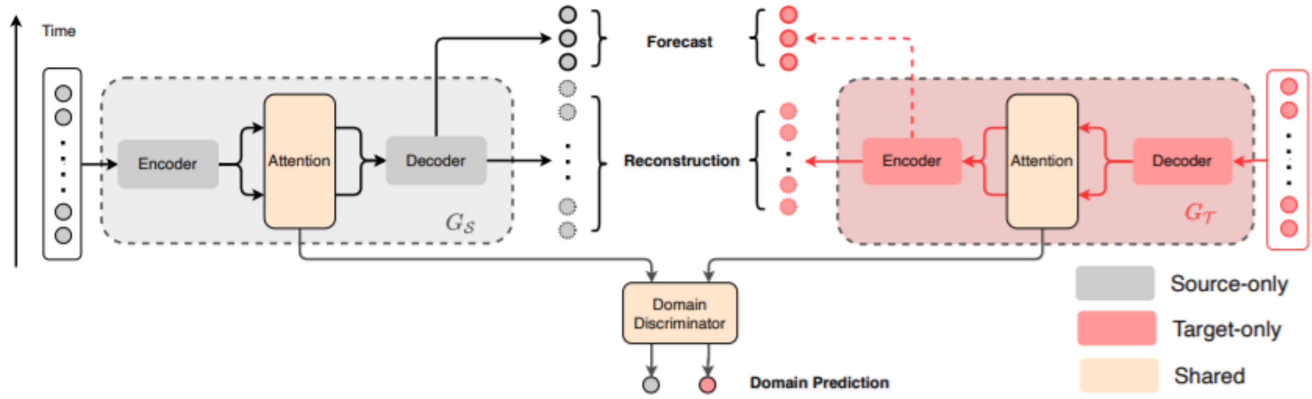
1. DATSING : Data Augmented Time Series Forecasting with Adversarial Domain Adaptation
   - https://arxiv.org/pdf/2102.06828.pdf
2. Domain Adaptation for TSF via Attention Sharing
   - https://arxiv.org/abs/2102.06828
3. Domain-Adversarial Training of Neural Networks
   - https://arxiv.org/abs/1505.07818
4. A DIRT-T Approach to Unsupervised Domain Adaptation
   - https://arxiv.org/abs/1802.08735
5. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation
   - https://arxiv.org/abs/1712.02560

2. Key Points

(1) DATSING : Data Augmented Time Series Forecasting with Adversarial Domain Adaptation

- data augmentation
  - 아이디어 1) ***국내의 (일부) 데이터를 친환경으로 취급 & 국내(일부)+친환경 으로 모델 학습***
- transfer “domain-INVARIANT” feature representation, from a “pre-trained stacked deep residual network” to “target domains”
  - 아이디어 2) ***(선) “국내”로 모델 학습 → (후) “친환경”으로 weight transfer***

(2) Domain Adaptation for TSF via Attention Sharing



.

- 아이디어 3) ***국내 & 친환경을 구분하지 못하게끔 유도하는 discriminator 장치 두기***
  - 해당 representation를 뽑아내는 shared layer는 어떻게 할지는 무궁무진

    (여기서는 shared “attention” … Q & K)

(3) Domain-Adversarial Training of Neural Networks

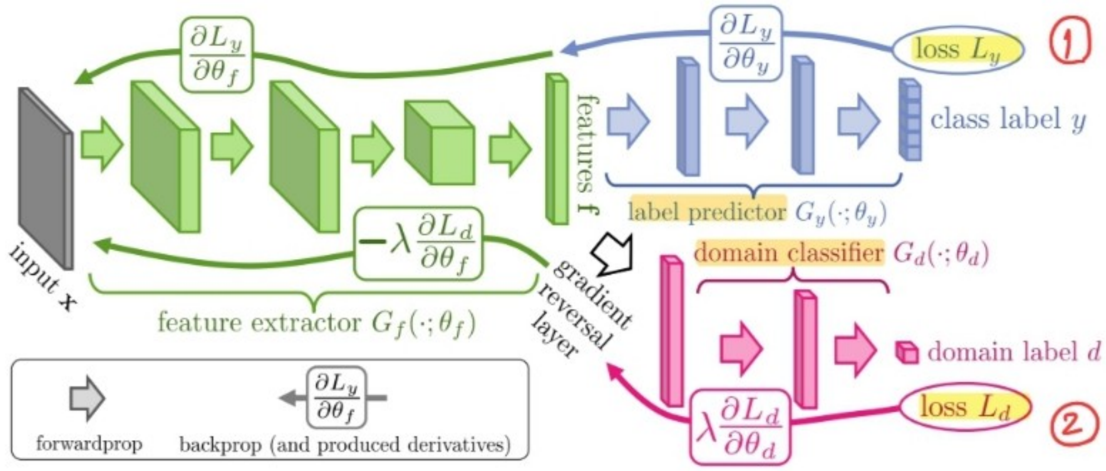


Figure 1: The **proposed architecture** includes a deep *feature extractor* (green) and a deep *label predictor* (blue), which together form a standard feed-forward architecture. Unsupervised domain adaptation is achieved by adding a *domain classifier* (red) connected to the feature extractor via a *gradient reversal layer* that multiplies the gradient by a certain negative constant during the backpropagation-based training. Otherwise, the training proceeds standardly and minimizes the label ① prediction loss (for source examples) and the domain classification loss (for all ② samples). Gradient reversal ensures that the feature distributions over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus resulting in the domain-invariant features.

.

- 비교적 간단한 구조
- 아이디어 4) ***2개의 loss 두기***
  - **loss 1 : forecasting loss**

- **loss 2: (음의) domain 구분 loss**

  (사실상, (2) DAF와 그 아이디어는 유사하다 볼 수 있음)

(4) A DIRT-T Approach to Unsupervised Domain Adaptation

- 데이터들간에는 숨겨진 cluster가 있을 것!
- 그 외의 내용들은 너무 복잡...loss에 뭐 이런거저런거 많이 섞구...

  $\min_\theta \mathcal{L}_y(\theta; \mathcal{D}_s) + \lambda_d \mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t) + \lambda_s \mathcal{L}_v(\theta; \mathcal{D}_s) + \lambda_t [\mathcal{L}_v(\theta; \mathcal{D}_t) + \mathcal{L}_c(\theta; \mathcal{D}_t)]$.

  - 1) $\mathcal{L}_y(\theta; \mathcal{D}_s)$: CE loss
  - 2) $\mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t)$: domain discriminator loss
  - 3) $\mathcal{L}_v(\theta; \mathcal{D}_s)$: term for "locally-Lipschitz" (for SOURCE)
  - 4) $\mathcal{L}_v(\theta; \mathcal{D}_t) + \mathcal{L}_c(\theta; \mathcal{D}_t)$: term for "locally-Lipschitz" & "conditional entropy" (for TARGET)

(5) Domain Adaptation with Representation Learning and Nonlinear Relation for Time Series

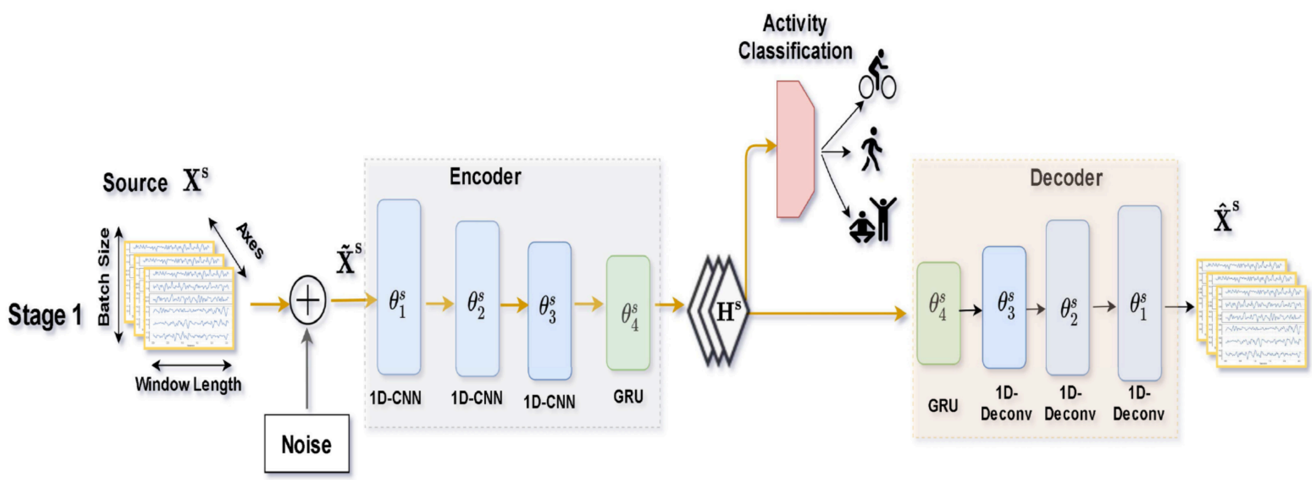
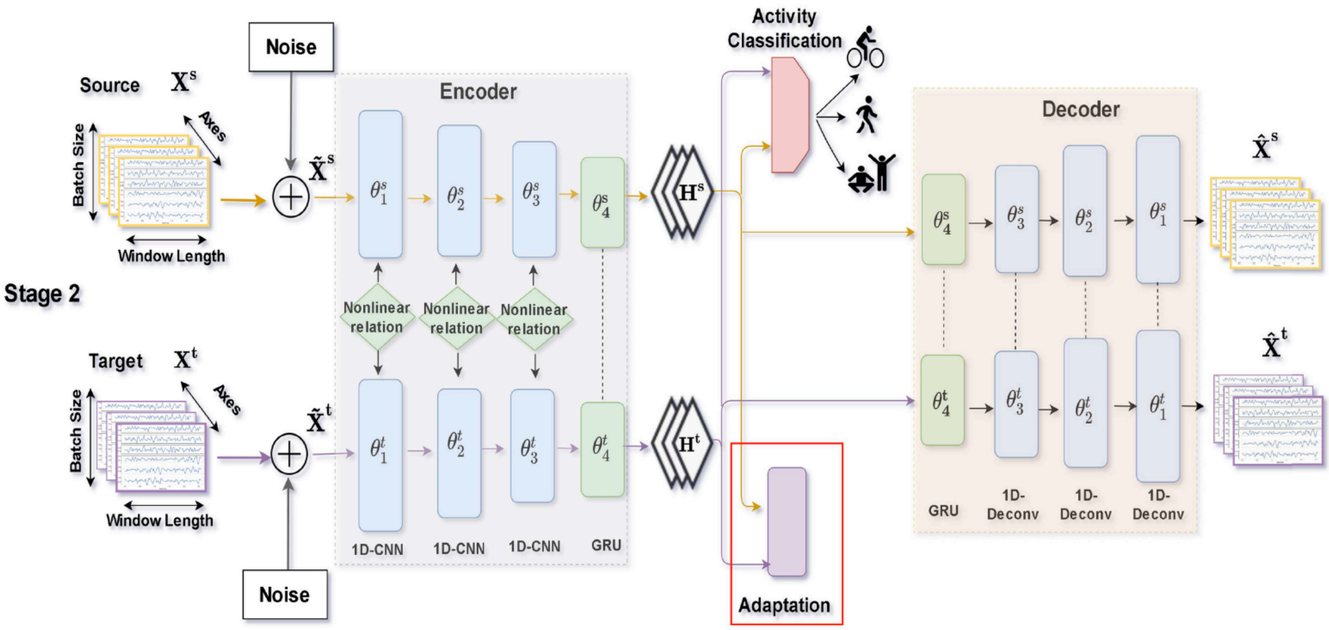


Fig. 2. Source denoising auto-encoder. The encoder model consists of two 1D-CNN layers with max-pooling followed by a GRU layer. The decoder model consists of a GRU followed by 1D-transposed convolution with upsampling to reverse the operation of the encoder.



$$\mathcal{L}_{da} = \mathcal{L}_{mmd}(H^s, H^t) = \left\| \sum_{i=1}^{N} \frac{\phi\left(\mathbf{H}_i^s\right)}{N} - \sum_{j=1}^{M} \frac{\phi\left(\mathbf{h}_j^t\right)}{M} \right\|_{\mathcal{H}}^2$$

$$\mathcal{L}_{mmd}(H^s, H^t) = \sum_{i,i} \frac{K\left(\mathbf{H}_i^s, \mathbf{H}_i^s\right)}{(N)^2} - 2\sum_{i,j} \frac{K\left(\mathbf{H}_i^s, \mathbf{H}_j^t\right)}{NM} + \sum_{j,j} \frac{K\left(\mathbf{H}_j^t, \mathbf{H}_j^t\right)}{(M)^2}$$

MMD

- https://seunghan96.github.io/gan/(DGM)12.Difference_of_two_distn/
- MMD 처럼 latent space를 확률적으로 모델링하면 uncertainty 접근도 가능해서 보다 나을 듯?