Modeling stochastic processes by simultaneous optimization of latent representation and target variable

Jakob Jelenčič

Supervisor: prof. Dr. Dunja Mladenić

Artificial Intelligence Laboratory
Jožef Stefan Institute

Overview:

- Problem statement.
- Data.
- Method's details.
- First results.
- Future work.

Problem statement

- Prediction of a stochastic process.
- Limited data.
- Overfitting.

Data

- Equity dataset:
  - Train set: 2007-2019.
  - Test: 2020-present.
  - Every stock listed on Nasdaq, daily data.
  - 10 Days Trend prediction.
- Cryptocurrency dataset:
  - Minute data.
  - Train: 2014(17) – 2020.
  - Test: 2021 – present.
  - 6 hour trend prediction.

Method’s details

- Empirical normalization.
- Noise addition.
- Optimization of latent representation.
- Designed to prevent overfitting.
- Improve metric on test data.
- Focus on deep learning.

Empirical normalization

- Improves gradient descent (overall).
- Sync data distribution and noise distribution.
- Shift central moments of empirical distribution to the ones from N(0,1).

Noise addition

- 3 parameters.
- Decreasing while model converges.
- Normally distriubuted.

Algorithm 1 Noise definition

Inputs: X, α, β, $epoch$
$Y = [ts, ts, np]$ ▷ Array for holding Cholesky decomposition's of time correlation matrices.
for $t \in \{1, \dots, np\}$ **do**
  $\Sigma_t = cov(X[,,t])$
  $Y[,,t] = chol(\Sigma_t)$ ▷ In practice the closest positive definite matrix of Σ_t is computed before the Cholesky decomposition.
end for
$Z = [bs, ts, np]$ ▷ Array for holding noise samples.
for $i \in \{1, \dots, ts\}$ **do**
  $\Sigma_i = cov(X[,i,])$
  $Z[,i,] = mvn(bs, \Sigma_i)$
end for
for $j \in \{1, \dots, np\}$ **do**
  $Z[,,j] = matmul(Z[,,j], Y[,,j])$ ▷ Correcting initially independent noise samples with respect to time.
end for
for $w \in \{1, \dots, ts\}$ **do**
  $Z[,w,] = Z[,w,] * ((\beta^{ts-w} \cdot \alpha^{epoch}) \cdot sd)$ ▷ Decrease the noise during training procedure.
end for
$R = X + Z$
Return R.

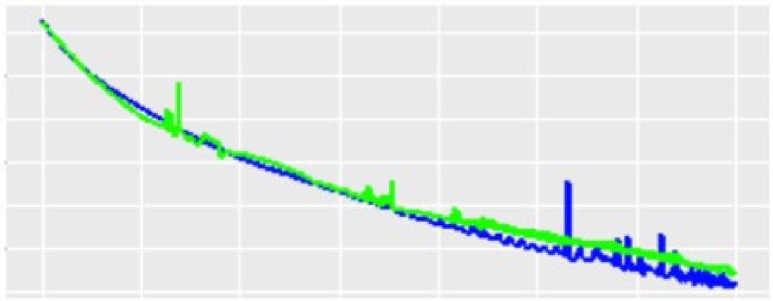
Optimization of latent representation

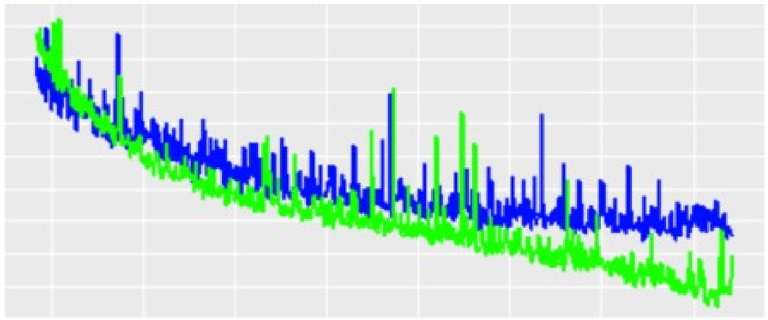
- Inital initialization could be non-optimal.
- Shift representaion with help of autoencoders.
- Decrease AE weight while model converges.

$$L = L_Y + W_{ae} \cdot decay^{epoch} \cdot L_{ae},$$

First results: noise importance (unsupervised)

- Top – uncorrelated noise.
- Bottom – correlated noise.
- Results of autoencoder on test set on cryptocurrency dataset.
- Suprising result – improvement on train/validation set aswell.





Artificial Intelligence Laboratory
Jožef Stefan Institute

First results: Supervised comparison

Table 1: Supervised results on equity dataset.

Method	Train Accuracy	Test Accuracy
Majority	0.513	0.537
Random Forest	0.649	0.655
GLM	0.664	0.655
LSTM	0.681	0.673
latent LSTM	0.633	0.673
noise LSTM	0.681	0.675
latent noise LSTM	0.681	0.682

Table 2: Supervised results on cryptocurrency dataset.

Method	Train Accuracy	Test Accuracy
Majority	0.512	0.556
Random Forest	0.689	0.692
GLM	0.682	0.695
LSTM	0.754	0.696
latent LSTM	0.736	0.683
noise LSTM	0.697	0.695
latent noise LSTM	0.706	0.714

Future work

- Analyze inmportance & sensitivty of parameters.
- Test multiple datasets.
- Deeper analysis how noise structure affects learning.

Thank you for the attention.