Neural Networks and Deep Learning
Exercise 4

April 14, 2025

Problem 2

Paper Name and Authors

The paper is called *"Deep Learning Statistical Arbitrage"*, written by Jorge Guijarro-Ordonez, Markus Pelger, and Greg Zanotti. Arxiv link: `https://arxiv.org/abs/2106.04028`.

Task (Statistical Arbitrage via Residuals)

They want to build trading signals using residual returns, which come from a standard factor model. In other words, if $R_{n,t}$ is the return of stock n at time t, they estimate some factor-based prediction $\hat{R}_{n,t}$, and the residual becomes:

$$\varepsilon_{n,t} \;=\; R_{n,t} \;-\; \hat{R}_{n,t}.$$

By focusing on these $\varepsilon_{n,t}$, the authors try to find short-horizon deviations that revert quickly, hence a form of statistical arbitrage.

Input Data and Preprocessing

They use a panel of US equity data from the CRSP database, selecting large and liquid stocks. For each stock and day, they compute the factor-based prediction, so that $\varepsilon_{n,t}$ is orthogonal to the systematic factor exposures. This step is called residualization. After that, each residual series is mostly uncorrelated with main risk factors, making it easier to detect short-term mispricings from the factor morel.

Architecture (CNN + Transformer)

They take each residual time series (30, and 60 days long) and pass it to a two-layer CNN, which finds local features like short trends or local mean-reversion. Then, they feed these local-feature outputs to a Transformer, where

the self-attention mechanism chooses which time steps are most relevant. Finally, a simple feedforward neural network takes the Transformer's outputs and produces a forecast or signal, which says how to trade the residual portfolio.

Some interesting components points are instance normalization before activations in CNN and Relu on the normalized linear transforms before the convolution.

Training Setup (Supervised)

They label each day's trading decision by looking at the objective of maximizing risk-adjusted returns (maximizing the Sharpe ratio). They form a loss/utility function that depends on the performance of the strategy. They use rolling windows: train on older data, then evaluate signals on newer data. So it is supervised, and evaluated out of sample in the sense that the model sees pairs of (residual history, trading outcome) and tries to learn a mapping that yields profitable trades, and the profitability of these trades are evaluated based on data that the model has not been seen yet.

Outcome

They found the CNN+Transformer model performs better than simpler methods, such as classical Ornstein-Uhlenbeck approaches or Fourier transforms for mean reversion. Their final strategy achieves higher Sharpe ratios and greater returns. This suggests that advanced deep learning can discover richer patterns in residual returns for statistical arbitrage.

Problem 3

We wish to learn the structure ($\mathrm{Cor}(X, Y) = c$) from the joint distributio of the data. We solve this by minimizing the MSE

$$\min_{w} \ \mathbb{E}\Big[(y - w\,x)^2\Big].$$

Taking the derivative with respect to w, we have:

$$\frac{d}{dw}\,\mathbb{E}\Big[(y - w\,x)^2\Big] = \mathbb{E}\Big[-2\,x\,(y - w\,x)\Big] = 0.$$

This implies

$$\mathbb{E}[x\,y] - w\,\mathbb{E}[x^2] = 0,$$

so that the optimal weight is

$$w = \frac{\mathbb{E}[xy]}{\mathbb{E}[x^2]}.$$

Since from the covariance matrix we have $\mathbb{E}[xy] = c$ and $\mathbb{E}[x^2] = 1$, it follows that

$$w = c.$$

Thus, the learned regression parameter is exactly the correlation parameter c.

This method is self-supervised because:

- No external labels are needed; the supervision comes entirely from the natural structure of the data.
- The task is a pretext task where the prediction target (y) is derived from the data itself.
- The model, by minimizing the prediction error, automatically learns the underlying parameter, which here is c.

Interestingly this is quite similar to how we think about