Deep Learning for Multi-Dimensional Functional Data

Literature Review and Plan for Simulation Study

Carter Hall, Connor McNeill, and Kris Wilson

North Carolina State University

December 2nd, 2025

Outline

1. Introduction
2. FDNN
3. CovNet
4. IRR via DNN
5. Conclusion

Motivation

Per Wang et al. (2024), there are three fundamental areas of functional data analysis:

- **Regression:** uncovering and understanding the underlying patterns in data that evolve over a continuum
- **Classification:** categorizing functional observations into distinct groups or classes
- **Representation:** efficiently identifying the underlying structure of functional data

Multi-dimensional Functional Data

One of the most common forms of two-dimensional functional data is images. Suppose we have a black and white image. The image can be modeled as a function

$$f(x, y) = \text{intensity at spatial location}(x, y)$$

where the spatial location is a pixel in the image. The intensity is measured on a scale from 0 to 255.

Figure 1: This image is selected from the Fashion-MNIST dataset (Wang et al., 2024)

Why Deep Neural Networks?

Previous methods have focused on using single-layer neural networks. However, there are several advantages to **deep** neural networks (which have 2 or more hidden layers).

- Deep neural networks (DNNs) thrive in high-dimensional spaces - the so-called “curse of dimensionality” is not really an issue.
- Deep learning models also offer a lot of flexibility and cater to a variety of different functional data structures (including shapes and images).
- DNNs have better predictive accuracy and are scalable.

We will explore three approaches to deep learning on multi-dimensional functional data.

Functional Deep Neural Networks (FDNN)

- **Motivation:** We often want to classify a data function based on training samples. One example of this is looking at brain imaging and trying to predict the stage of a new patient.
- Current methods:
  - focus only one-dimensional functional data classification,
  - sometimes require that the data function follows a Gaussian process (albeit this is sometimes violated in practice),
  - and when data distributions are non-Gaussian, the decision boundary can often not be accurately be recovered.

Functional Deep Neural Networks (FDNN)

- Wang et al. (2023) proposed the approach of Functional Deep Neural Networks (FDNNs) in order to help solve these problems with previous methods.
- FDNN, like other functional classification methods, starts by doing functional PCA to extract the principal components of the data functions.
- After this, we then train a DNN-based classifier on these FPCs as well as their corresponding class labels.

FDNN Methodology

Suppose we observe n iid training samples $\{(X_i(s), Y_i) : 1 \le i \le n\}$ to be classified into either $k = -1$ or $k = 1$.

We can perform the Karhunen-Loeve decomposition for the sample covariance function $\hat{\Omega}_k$ and then use this to write the sample data function X_i, under $Y_i = k$ as

$$X_i(s) = \sum_{j=1}^{\infty} \hat{\xi}_{ij} \hat{\psi}_{kj}(s), i = 1, \dots, n.$$

Intuitively, $\hat{\xi^{(i)}} := (\hat{\xi}_{i1}, \hat{\xi}_{i2}, \dots)^T$ is an estimator of $\xi^{(i)}$ which are unobservable random coefficients of X_i with respect to the population basis ψ_{kj}. We will truncation ξ at J and use these as classifiers.

FDNN Methodology

We then train a DNN to estimate the log density ratio functional Q^* based on the $\hat{\xi}_J^{(i)}$'s. Let σ denote the RELU function, and define the shift activation function for any real vectors V and y as $\sigma_V(y) = (\sigma(y_1 - v_1), \ldots, \sigma(y_w - v_w))^T$. With J inputs, L hidden layers, and p_ℓ nodes on the lth hidden layer, we fit our DNN.

Given the training data $(\xi_J^{(1)}, Y_1), \ldots, (\xi_J^{(n)}, Y_n)$ let

$$\hat{f}_\phi(.) = \arg \min_{f \in \mathcal{F}(L,J,p,B)} \frac{1}{n} \sum_{i=1}^{n} \phi(f(\hat{\xi}_J^{(i)}) Y_i),$$

where $\phi(x)$ is the hinge loss. The following is our FDNN classifier:

$$\hat{G}^{FDNN}(X) = \begin{cases} 1, & \hat{f}_\phi(\xi_J) \geq 0, \\ -1, & \hat{f}_\phi(\xi_J) < 0. \end{cases}$$

Covariance Estimation with Neural Networks (CovNet)

Problem: Covariance estimation of multidimensional functional data:

$$c(\mathbf{u}, \mathbf{v}) = \mathrm{Cov}(X(\mathbf{u}), X(\mathbf{v}))$$

based on i.i.d. sample $\mathcal{X}_1, ..., \mathcal{X}_n \sim \mathcal{X} := \{X(\mathbf{u}), u \in \mathcal{Q} \subset R^d\}$ from a compactly-supported random field taking values in $L_2(\mathcal{Q})$.
(If $d = 1$, random ***curves*** that are real-valued and square-integrable.)

1. **Curse of Dimensionality**: Storage is an $\mathcal{O}(K^{2d})$
2. **Common Assumptions**: Separability enables efficient computation... but often violated in practice (especially if data are **dense**, which is the subject of this paper).

CovNet: The DeepShared Model

Solution: Sarkar and Panaretos (2022) propose **CovNet**, a neural-network based model that

1. Avoids high-dimensional object storage
2. Non-parametric (**universal approximation property** – any covariance approximated with arbitrary precision)
3. Admits an **explicit** form through eigendecomposition, no interpolation required

Three classes of models: *Shallow, Deep, DeepShared*
(Will focus on **last** of these.)

$$c_{ds}(\mathbf{u},\mathbf{v}) = \sum_{r,s=1}^{R} \lambda_{r,s} g_r(\mathbf{u}) g_r(\mathbf{v}) \qquad \mathbf{u},\mathbf{v} \in \mathcal{Q}$$

CovNet: The DeepShared Model

For **networks** $g_r, r = 1, \dots, R$,

$$u_1 = \sigma(W_1 u + b_1)$$
$$u_{l+1} = \sigma(W_{l+1} u_l + b_{l+1})$$
$$g_r(u) = \sigma(\omega_r^T u_L + \beta_r)$$

where $l = 1, \dots, L-1$ with $L > 1$ layers, $W_1 \in R^{p_1 \times d}, W_2 \in R^{p_2 \times p_1}, \dots, W_L \in R^{p_L \times p_{L-1}}$.

Important Takeaway: Multiple networks, weights shared at all **but the final layer**.

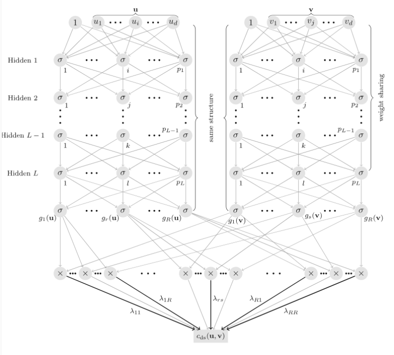

Figure 2: DeepShared CovNet architecture

CovNet: Model Properties

- **Parameter efficiency from weight sharing:** For hidden layers of width R, the deepshared CovNet requires $\mathcal{O}(R^2)$ parameters versus $\mathcal{O}(R^3)$ for the deep CovNet:

$$\text{DeepShared: } \sum_{l=0}^{L-1}(p_l+1)p_{l+1} + R(p_L+1) + R(R+1)/2$$

$$\text{Deep: } R\left(\sum_{l=0}^{L}(p_l+1)p_{l+1}\right) + R(R+1)/2$$

- **Approximation efficiency of deep architectures:** While all CovNet models are universal approximators, authors actually **expect the Deep/DeepShared variants to require less parameters than a shallow approximator.**

CovNet: Core Properties & Theory Summary

Estimation Framework:

- $\hat{C}^{\mathrm{ds}}_{R,L,N} \in \arg\min_{\mathcal{G}\in\tilde{\mathcal{F}}} \|\hat{C}_N - \mathcal{G}\|_2^2$
- $|||C|||_2 = \|c\|_{L_2(Q\times Q)}$ (Hilbert-Schmidt = L_2 kernel norm)
- Minimizes distance between empirical covariance and CovNet class $\tilde{\mathcal{F}}$

Efficient Eigendecomposition:

- Eigenfunctions: $\psi(u) = \sum_{r=1}^{R} a_r \hat{g}_r(u)$
- Solves $R \times R$ problem: $(\widetilde{G}\hat{\Lambda}\widetilde{G} - \eta\widetilde{G})a = 0$
- $\widetilde{G}_{rs} = \int_Q \hat{g}_r \hat{g}_s$ computed via Monte Carlo

Theoretical Results:

- **Consistency:** Weak/strong consistency under mild conditions, e.g. $E[\|\mathcal{X}\|^4] < \infty$
- **Rate:** Optimal rate of convergence found by bounding **bias** – $\mathcal{O}((N/d\ln(N))^{d/(10d+\alpha)})$ (shallow CovNet, order-α Sobolev space)

Image Response Regression Via Deep Neural Networks
Motivation

- Magnetic resonance imaging (MRI) can be used to identify brain regions with structural differences between neurologically disordered vs. healthy control patients after accounting for demographic covariates
- functional MRI (fMRI) can be used to identify brain regions that demonstrate different activation patterns for different tasks
- Call this image response regression: image scans in 2D or 3D regressed on demographic and clinical features.

IRR via DNN
Challenges

Challenges of image response regression include...

- complex spatial smoothness patterns

IRR via DNN
Challenges

Challenges of image response regression include...

- complex spatial smoothness patterns
- data is high dimensional

IRR via DNN
Challenges

Challenges of image response regression include...

- complex spatial smoothness patterns
- data is high dimensional
- subjects are heterogeneous

IRR via DNN
Challenges

Challenges of image response regression include...

- complex spatial smoothness patterns
- data is high dimensional
- subjects are heterogeneous
- limited sample size

IRR via DNN
Canonical approach

- Spatially Varying Coefficient Model (SVCM)

IRR via DNN
Canonical approach

- Spatially Varying Coefficient Model (SVCM)
  - images and regression coefficients modeled as realizations of spatially varying functions evaluated on discrete grid points

IRR via DNN
Canonical approach

- Spatially Varying Coefficient Model (SVCM)
  - images and regression coefficients modeled as realizations of spatially varying functions evaluated on discrete grid points
  - Essentially, voxels (3D pixels) in spatial domain

IRR via DNN
The Model

Suppose the imaging data are collected from a $D-$dimensional compact space $S \subset R^D$ observed at V spatial locations, along with J covariates, from N individual subjects.
For each subject $i = 1, \ldots, N$, let $y_i(s_v) \in R$ denote the image measurement at the spatial location $s_v \in S, v = 1, \ldots, V$ and let $S_V = \{s_v\}_{v=1}^V$ collect all those locations.
Let $x_i = (x_{i1}, \ldots, x_{iJ})^\top \in R^J$ denote the $J-$dimensional covariate vector. We consider the following SVCM, for $i = 1, \ldots, N, v = 1, \ldots, V,$

IRR via DNN
The Model

Suppose the imaging data are collected from a D–dimensional compact space $S \subset R^D$ observed at V spatial locations, along with J covariates, from N individual subjects.
For each subject $i = 1, \ldots, N$, let $y_i(s_v) \in R$ denote the image measurement at the spatial location $s_v \in S, v = 1, \ldots, V$ and let $S_V = \{s_v\}_{v=1}^{V}$ collect all those locations.
Let $x_i = (x_{i1}, \ldots, x_{iJ})^\top \in R^J$ denote the J–dimensional covariate vector. We consider the following SVCM, for $i = 1, \ldots, N, v = 1, \ldots, V,$

$$y_i(s_v) = \sum_{j=1}^{J} x_{ij}\beta_j(s_v) + \alpha_i(s_v) + \varepsilon_i(s_v), \qquad \mathrm{Var}\{\varepsilon_i(s_v)\} = \sigma^2(s_v)$$

IRR via DNN
The Model

$$y_i(s_v) = \sum_{j=1}^{J} x_{ij}\beta_j(s_v) + \alpha_i(s_v) + \varepsilon_i(s_v), \qquad \mathrm{Var}\{\varepsilon_i(s_v)\} = \sigma^2(s_v)$$

where:
- $\beta_j(s_v) \in R$ characterizes the main effect of the jth covariate x_{ij} common for all subjects i
- $\alpha_i(s_v) \in R$ characterizes the individual deviation from the common main effect that is specific to subject i. This is similar to a random effect in the SVCM, but here they are treated as deterministic and are estimated directly.
- $\varepsilon_i(s_v)$ is measurement error with zero mean and spatially varying variance

IRR via DNN
The Model

$$y_i(s_v) = \sum_{j=1}^{J} x_{ij}\beta_j(s_v) + \alpha_i(s_v) + \varepsilon_i(s_v), \qquad \mathrm{Var}\{\varepsilon_i(s_v)\} = \sigma^2(s_v)$$

Interpretation: "At each spatial location, here is the common effect (β_j), and here is how this subject deviates from that effect (α_i)."

IRR via DNN
Main Contributions

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

IRR via DNN
Main Contributions

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

- threshold *output*, not *layers*: maintains DNN flexibility in parameter estimation

IRR via DNN
Main Contributions

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

- threshold *output*, not *layers*: maintains DNN flexibility in parameter estimation
- mini-batch stochastic gradient descent results in faster convergence and computational efficiency

IRR via DNN
Main Contributions

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

- threshold *output*, not *layers*: maintains DNN flexibility in parameter estimation
- mini-batch stochastic gradient descent results in faster convergence and computational efficiency
- asymptotic error bound → main effect estimator is *consistent*

IRR via DNN
Main Contributions

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

- threshold *output*, not *layers*: maintains DNN flexibility in parameter estimation
- mini-batch stochastic gradient descent results in faster convergence and computational efficiency
- asymptotic error bound $\rightarrow$ main effect estimator is *consistent*
- error bound is comparable to minimax bound in classical nonparametric regression

IRR via DNN
Main Contributions (cont.)

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

- *explicitly* address both spatial correlation and subject heterogeneity

IRR via DNN
Main Contributions (cont.)

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

- *explicitly* address both spatial correlation and subject heterogeneity
- **compared to DNN, SVCM foundation + DNN estimation is more interpretable**

IRR via DNN
Main Contributions (cont.)

Goal: Zhang et al., 2024 proposed image response regression via deep neural networks. Their main contributions are as follows:

- *explicitly* address both spatial correlation and subject heterogeneity
- **compared to DNN, SVCM foundation + DNN estimation is more interpretable**
- for DNN, observations are spatial coordinates, so effective sample size is large

IRR via DNN
Advantages of DNN

- don't need to interpret individual neural network parameters

IRR via DNN
Advantages of DNN

- don't need to interpret individual neural network parameters
- don't need to impose sparsity on individual neural network parameters (recall we hard threshold the output)

IRR via DNN
Advantages of DNN

- don't need to interpret individual neural network parameters
- don't need to impose sparsity on individual neural network parameters (recall we hard threshold the output)
- input is spatial coordinate, output is model coefficient

IRR via DNN
DNN Estimation Process

1. Estimate main effect coefficient $\hat{\beta}_j$
2. Estimate individual deviations $\hat{\alpha}_i$
3. Estimate error variance $\sigma^2(s)$

IRR via DNN
Illustration of method

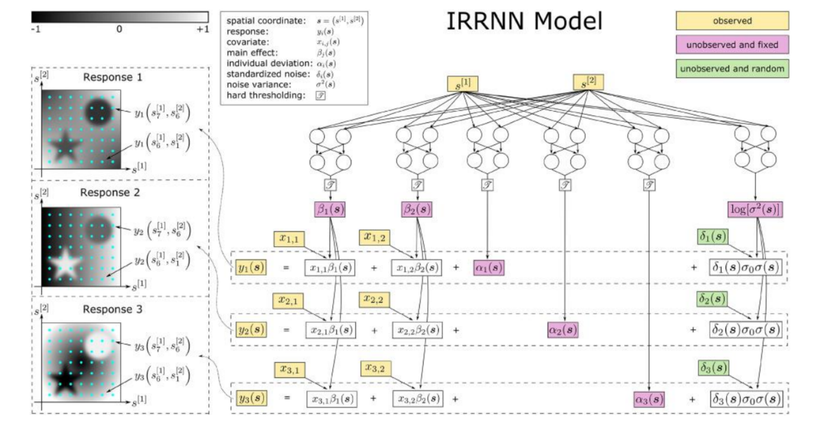

Figure 1. Architecture of the proposed model. In this example, the data contain $J = 2$ covariates and $N = 3$ images of dimension $D = 2$, observed on a grid of $V = 7 \times 7$ voxels.

Next Steps and Summary

Real Data Analysis Plan:

- Using the fashion-MNIST dataset, compare the Classification method of FDNN with functional PCA (and logistic regression).

Simulation Study Plan:

1. Generate synthetic data in higher-dimensions, e.g. $[0, 1]^d$ with $d \geq 2$ where the true covariance is known (can take from some family, e.g., Matern) and at increasing resolution ($K = 10, 20, 100$ data per dimension).
2. Implement DeepShared CovNet and evaluate estimation accuracy of true covariance, showcase rates of convergence as number of networks R increases.

Next Steps and Summary

Deep learning methods for functional data exist within **many different** statistical settings:

- **Classification**: FDNN allows for classification of non-Gaussian high-dimensional functional data.
- **Covariance Estimation: CovNet** gives general, computationally **efficient** method for learning spatial/spatiotemporal co-variations.
- **Regression**: Image response regression is a fundamental area of functional data analysis. Using deep neural networks within the SVCM framework allows for efficient estimation and straightforward interpretability

Sarkar, S., & Panaretos, V. M. (2022).CovNet: Covariance Networks for Functional Data on Multidimensional Domains. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *84*(5), 1785–1820. https://doi.org/10.1111/rssb.12551

Wang, S., Cao, G., Shang, Z., & for the Alzheimer's Disease Neuroimaging Initiative. (2023).Deep neural network classifier for multidimensional functional data. *Scandinavian Journal of Statistics*, *50*(4), 1667–1686. https://doi.org/10.1111/sjos.12660

Wang, S., Zhang, W., Cao, G., & Huang, Y. (2024).Functional data analysis using deep neural networks. *WIREs Computational Statistics*, *16*(4). https://doi.org/10.1002/wics.70001

Zhang, D., Li, L., Sripada, C., & Kang, J. (2024).Image response regression via deep neural networks. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *85*(5), 1589–1614. https://doi.org/10.1093/jrsssb/qkad073