Behavioral Data Science: Homework Assignment 2
Deep Learning

February 9, 2025

All answers and solutions to non-programming questions should be submitted to LMS as a **legible** write-up (either fully digital or a scan). The use of LLMs (e.g., ChatGPT) is **explicitly discouraged**, unless specified otherwise. All code should be committed to and merged into the `main` branch of your team's GitHub repository, unless specified otherwise. Your LMS submissions should contain a single ZIP file named according to the pattern:

- BDSl_Assignment[#]_[TeamMember1Initials]_[TeamMember2Initials]

Problem 1: True-False Questions (4 points)

Mark all statements which are **FALSE**.

1. NumPy arrays and TensorFlow tensors are fundamentally interchangeable and can always be used interchangeably in all TensorFlow operations without any explicit conversion.
2. Stochastic gradient descent (SGD) with a fixed learning rate always converges faster than Adam when training deep neural networks.
3. The Jacobian matrix of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ has shape (n, m).
4. Mini-batch gradient descent is practically never used in deep learning due to its high memory demands.
5. TensorFlow's automatic differentiation engine (autodiff) uses numerical differentiation to compute gradients with respect to variable nodes.
6. The second derivative of the loss function with respect to individual model parameters is always required for implementing gradient-based optimization methods.
7. The sigmoid function can be used both as an activation function in hidden layers of MLPs and as an output activation for binary classification tasks.
8. Setting `tf.random.set_seed(seed)` ensures deterministic behavior for TensorFlow's random functions.

Problem 2: PyTorch vs. TensorFlow (4 points)

Read up the key differences between PyTorch and TensorFlow, explaining what they mean for users of the frameworks. Provide a snippet of how you would define the same fully connected network we used in class for ASD classification.

Problem 3: Gradient Descent (6 points)

1. What is the gradient of a multi-variable function and what information does it provide? What is the goal of *gradient descent*? For the function

$$C(\theta_1, \theta_2) = 2\theta_1^2 + 3\theta_2\theta_1 \tag{1}$$

perform three manual steps of gradient descent for a step size α of your choosing. Initialize the vector $\theta = (\theta_1, \theta_2)$ with the first and second digit of your birthday, for instance, January 1: $\theta^{(0)} = (0, 1)$, May 16: $\theta^{(0)} = (1, 6)$.

2. For the multivariate function $u(x, y) = x^2 - y^2$ and $w(x, y) = 2xy + e^x$, find the Jacobian matrix $\mathbf{J}$ and evaluate it at the points $(1, 1)$ and $(0, 0)$.

Problem 4: Maximum Likelihood Estimation Continued (8 points)

In the last homework, you derived the maximum likelihood estimate of the mean under a Gaussian data model. In this exercise, you will explore a deeper connection between maximum likelihood estimation (MLE) and information theory, ultimately leading to an understanding of MLE as distribution matching.

Let's restate our goal first: we want to minimize the difference between our data model $p(x \mid \theta)$ and the assumed data-generating distribution $p^*(x)$. But how do express the notion of difference or *distance* between distributions? Fortunately, information theory has the answer, which comes in the form of the Kullback-Leibler (KL) divergence, defined for continuous densities p and q as:

$$\mathrm{KL}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} dx \tag{2}$$

The KL divergence has a very neat property: it evaluates to 0 if and only if the two densities are the same, that is, $p = q$ (you can try to show this as a bonus exercise). To make the connection between MLE and the KL divergence, your first task is to show that minimizing the KL divergence leads directly to the familiar and general MLE formulation:

$$\theta_{\mathrm{MLE}} = \arg\min_{\theta} \mathrm{KL}(p^*(x) \,||\, p(x \mid \theta)) \tag{3}$$

Did you notice something about the order of arguments in the KL? Your second task is to show that the KL is **non-symmetric**, that is, swapping the order of arguments in Eq. 2 results in a different value whenever $p \neq q$. In other words, you will show that, strictly speaking, the KL is not a proper distance, hence the wording *divergence*.

Problem 5: Simple Mixture-of-Experts (MoE) Approach (10 points)

In previous classes, we fitted three distinct models for ASD classification in our envisaged "app":

1. A logistic regression model as a baseline.
2. A rule-based decision model using the clinical score.
3. A fully connected neural network classifier.

In this exercise, you will explore an alternative approach that is often a strong baseline for classification tasks on tabular data.

Step 1: Background Reading

First, familiarize yourself with **Decision Trees** and the **Random Forest Algorithm** using the following non-limiting resources:

- **YouTube**: Decision Trees and Random Forests (Video)
- **Kaggle**: Random Forest Classifier Tutorial

Step 2: Constructing Meta-Features

Next, create new feature sets for the training and test data. These feature sets should consist of the predictions from the three models above. This will result in a new dataset with three columns, each representing the predictions from one of the models.

Step 3: Training the Random Forest Classifier

Use the `sklearn` implementation of the Random Forest classifier to fit a model using the predictions from the three models as input features. Evaluate its accuracy on the test set. This approach effectively learns a "weighting" of the three individual classifiers.

Step 4: Comparing Performance

Finally, compare the performance of this mixture-of-experts approach to a straightforward application of a Random Forest classifier trained directly on the original dataset. Summarize your findings, highlighting any improvements or differences in classification accuracy.