Project report

Author: Vilmos Szirmai, Supervisor: Adrián Csiszárik

2023 December

Self-supervised Deep Learning Models on Time-series Data

1 The topic and goals for the semester

Self-supervised learning is one of the important paradigms in machine learning today and the essence of this approach is that we are trying to create labels from the data itself. This concept is used when there are no labels or when we want to exploit the internal correlations within the data. It is quite an inexpensive way to obtain data for training. I aimed to acquire both the theoretical and practical aspects of self-supervised learning and that was the reason of opting this topic. Another reason of this is that currently I am receiving a scholarship from Bosch, and I work on a project which mainly includes time-series data like audio domain. But we had to wait a little time to get data from Bosch, and that is why I chose a dataset corresponding to speech recognition at first to have something to work on.

2 Data preprocessing

I chose a dataset from Kaggle named **Hindi speech classification**, which is a subset of the **Common Voice** dataset [Mozilla Foundation, 2021]. It contains 1998 training and 500 test audio files, labelled with female and male categories. Although it is a labelled dataset and first we worked with it using supervised models, later we also applied self-supervised approaches.
First, I explored the domain to find effective ways to process audio data. I tried various approaches until I realized that using a spectral representation of audio data is a common tool in audio modelling. A variant of such approach is Mel Frequency Cepstral Coefficients (MFCC) [Xu et al., 2004]. It is a representation of the audio data stored in a matrix, and calculated by the following algorithm:

Pre-emphasis: The first step is pre-emphasis, which involves boosting the higher frequencies to balance the frequency spectrum.
Frame blocking: The signal is divided into short frames (typically 20-40 milliseconds). This allows for analysis of a signal's frequency content over time because speech signals are assumed to be stationary over short periods.
Windowing: Each frame is multiplied by a window function to reduce spectral leakage during the transform.
Fast Fourier Transform (FFT) [Cooley and Tukey, 1965]: The FFT converts the signal from the time domain to the frequency domain, obtaining the power spectrum of the signal. The formula: if there are given $x_0, x_1, \ldots, x_{n-1}$ complex numbers, then the **Discrete Fourier Transform** is $X_k = \sum_{m=0}^{n-1} x_m \cdot \exp(\frac{-i2\pi km}{n})$ $(k = 0, 1, \ldots, n-1)$. Evaluating this definition directly requires $O(n^2)$ operations: there are n outputs X_k, and each output requires a sum of n terms. An FFT is any method to compute these in $O(n \log n)$ operations.
Mel filterbank: A set of overlapping triangular filters (spaced according to the Mel scale) are applied to the power spectrum. The filters are designed to mimic the non-linear human ear response to different frequencies.
Logarithm: The logarithm of the filterbank energies is taken to mimic the human perception of loudness.
Discrete Cosine Transform (DCT): The resulting log filterbank energies are transformed using the DCT to decorrelate the features, highlighting the most significant components.
MFCCs: The coefficients resulting from the DCT are the MFCCs. Usually, a certain number (commonly 13, I also chose this number) of these coefficients is retained as features for further analysis.

All MFCC matrices have 13 rows and 52 columns after I chose 52 adjacent columns randomly. It was necessary because the samples needed to be of equal size. The reason for choosing 52 was that it represented the minimum number of columns among all matrices. I used **PyTorch** and **Sklearn** throughout the project.

3 The supervised part

In order to implement the supervised part, I created a neural network consisting of one-dimensional convolutions, where the first convolutional layer has 13 input channels and the kernel size for each was 1, and the number of filters was 32, 64, and 64 respectively. There were two linear layers too with input sizes of 64 and 128, and output sizes of 128 and 2 respectively. Indeed, 2 is the number of classes. In the training cycle, there were 10 epochs, I set the batch size to 32, the learning rate was 0.01 and I used Adam optimizer. The result was outstanding, the accuracy score was over 0.98. This phenomena can be explained as the MFCC is a describing enough representation of the audio data and there are significant differences between female and male samples.

4 The self-supervision

The essence of self-supervision is to make labels from the data itself. As we worked with time-series one possible way to do this is by splitting the timeline, in this case the matrix, in half. However, if we were to merely cut it in half, the task would be too straightforward. Therefore, I introduced a slight pause (4 columns) at the split to add a level of complexity between the two parts. Then I used **Contrastive Predictive Coding** (CPC) [Oord et al., 2018] to create an embedding. Its task is to embed samples in such a way that similar ones are close to each other, while different ones are far apart.

4.1 Contrastive loss

The proximity of vectors can be expressed by their inner product. It can be implemented with ease and we can measure the loss in a new way. So I used the inner product of the embeddings stored in a Gram matrix, and the loss was the squared Frobenius-norm of the identity matrix minus the Gram-matrix. The formula of the loss is the following:
$\sum_{i,j=1}^{64} (C_{ij} - I_{ij})^2$, where $C_{ij} = \langle E_{i.}, E_{.j}^T \rangle$, where E denotes the matrix of embeddings and I_{ij} is the corresponding element from the identity matrix.

4.2 Experiments

After creating the embeddings, I assessed their suitability for the classification task itself. It was necessary to examine how well the embeddings, when taken as a basis and passed through an additional linear layer, could predict whether a given sample belonged to a male or female category. The first experiments result in accuracy scores about 0.75. For a more conclusive comparison we would need further explorations. I experienced that the loss was quite high during the training, even though it was convergent. The exact value hovered around 60.
So, looking ahead, in the upcoming semester, we are expected to work with Bosch data and further refine our self-supervised learning algorithms based on that.

References

James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

Mozilla Foundation. Common voice, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. Hmm-based audio keyword generation. In *Pacific-Rim Conference on Multimedia*, pages 566–574. Springer, 2004.