1 Machine Learning

1.1 Principal Component Analysis

After downsampling, a waveform used by `mlgw_bnsis` described by several hundred points. It is convenient to reduce this number in order for the neural network to be faster. We are able to do so by making use of the fact that the components of the high-dimensional vector representing the waveform are correlated.

The technique of **PCA!** (**PCA!**) is quite general, so let us describe it in general terms, and then apply it to our specific problem.

1.1.1 General method

We start with a dataset of N points in $\mathbb{R}^D$, which we denote by $\{x^i\}_{i=0}^{N-1}$. We need D floating point numbers to represent each of these points.

If we can find a k-dimensional hyperplane in $\mathbb{R}^D$, with $k \ll D$, such that our points are never very far from this subspace, we can substitute the D-dimensional parametrization of the points for a k-dimensional one by approximating each point by its orthogonal projection onto the k-dimensional hyperplane. We will make a certain error in this process: specifically, if $P_k(x_i)$ denotes the projection of the point onto this hyperplane, the error (computed according to the Euclidean distance among points[1]) can be quantified by

$$\text{error}(k) = \sum_{i=0}^{N-1} \left\| x_i - P_k(x_i) \right\|^2 . \tag{1.2}$$

The algorithm of PCA allows us to determine which hyperplane minimizes this error.

The first step is to center the data: we compute their mean $\overline{x}$, and work with the dataset $y_i = x_i - \overline{x}$. Because of this, we can say that the k-dimensional hyperspace is now a *subspace* with respect to y. Computationally, we keep the mean $\overline{x}$ saved and add it to the reconstructed data y.

Let us now consider the $k = 1$ case: we want to project the data onto a single line, which we can parametrize as the span of a unit vector w. Therefore, what we want to minimize is $\sum_i \left\| y_i - (y_i \cdot w) w \right\|^2 = \sum_i \left(\left\| y_i \right\|^2 - (y_i \cdot w)^2 \right)$, and we can do so by maximizing $\sum_i (y_i \cdot w)^2$.

Therefore, the best 1-dimensional subspace is the direction of maximum variance:

$$w = \underset{w \in \mathbb{S}^{D-1}}{\text{argmax}} \sum_i (y_i \cdot w)^2 . \tag{1.3}$$

[1] One might object here: the Euclidean distance among points is hardly relevant for our practical application! Fortunately, as we will later discuss, the PCA reconstruction of the points is efficient according to the Wiener distance as well as according to the Euclidean one. This might be understood heuristically by thinking of the fact that, when looking at waveforms which are quite close in terms of both distances, the linear approximation

$$\text{Wiener distance}(a_i, b_j) \approx \sqrt{g_{ij} a_i b_j} \tag{1.1}$$

for some metric g_{ij}, where a_i and b_j denote the vectors representing the two waveforms. Now, we do not know the precise form of g_{ij} (and, of course, we could not give a unique expression since it varies with detector noise), but as long as the metric is not pathological convergence in the Euclidean distance will imply convergence for this alternative distance.

Now comes the clever idea of **PCA!**: we can reformulate this argmax problem as an eigenvalue problem for the covariance matrix of the data:

$$C = \frac{1}{N} \sum_i y_i y_i^\top . \tag{1.4}$$

A unit eigenvector w of this matrix will satisfy $Cw = \lambda w$ for its eigenvalue λ; and we can recover the eigenvalue λ from this equation by computing $w^\top Cw = \lambda w^\top w = \lambda$; making the covariance matrix explicit allows us to see that

$$\lambda = w^\top Cw = \frac{1}{N} \sum_i (y_i \cdot w)^2 ; \tag{1.5}$$

which is precisely the quantity we wanted to maximize: therefore, the best one-dimensional subspace is precisely the largest eigenvector of the covariance matrix.

If we make the further observation that the covariance matrix is symmetric and positive definite, and can therefore be orthogonally diagonalized, we are almost done: we can generalize to arbitrary k moving one vector at a time. To find the second vector to span the subspace we can restrict ourselves to the subspace $w^\perp$ and apply the same procedure as before, this tells us that the optimal two-dimensional subspace is the span of the first two eigenvectors of the covariance matrix, and so on.

If we approach the problem by diagonalizing the covariance matrix C, the computational complexity in the worst case scenario is $\mathcal{O}\left(D^3\right)$, since it involves the diagonalization of a $D \times D$ matrix.

1.1.2 PCA! for waveforms

1.2 Neural Networks