Summary

In signal processing and machine learning, sparse encoding plays an important role. Dictionary learning has several names one is sparse dictionary learning and the other is sparse coding this may cue you into what kind of representation we're trying to learn the goal of dictionary learning is to build a sparse representation of some input data where the data is represented as a linear combination of basic elements these elements are called atoms and the collection of atoms is called a dictionary that's where the name comes from the atoms in a dictionary do not need to be orthogonal and they may be an over complete spanning set just a quick note on completeness a basis is complete if any vector in the vector space can be represented by a linear combination of vectors. By the data having d dimensions and the dictionary being described by d1 d2 etc, they're different things we also want to learn a representation r which has k vectors one for each input data sample, and these vectors are in n-dimensional space where n is the number of atoms in the dictionary. The goal of dictionary learning is to minimize the reconstruction error we use the dictionary and representations to reconstruct the input and we want to keep the individual representations for each input sparse. all of this comes together in this optimization problem we want to find d and r. “C” is a constraint that keeps “D” from becoming arbitrarily large for very small values of “R” lambda controls the trade-off between sparsity and error minimization so if lambda is small then we mostly care about minimizing the reconstruction error and if lambda is large then we care much more about sparsity. We want to greedily choose one atom at a time to reduce the approximation error and we do this by finding the atom with the highest inner product. Remember the dot product with the input signal in order for this to be informative the atoms must be normalized then we subtract the approximation using that atom from the signal and repeat until the norm of the residual is small. In the paper, the author has used techniques from the mathematical domain in signal processing, called Lie Groups & Representation Theory. The combination of these mathematical tools and Bayesian models results in different transformations of the image in a completely unsupervised manner. Because we're trying to discriminate between labels they're sort of being pushed out from the zero as much as possible the nice thing is that the other kind of information that again semantically aligns with the labels in this case class 1 and class 2 sort of follows the same pattern so you still end up with a disentangled latent space the difference here is that I know what z1 is and I know what z2 is, so I can later on use them to perform inference in terms of classifying new data points the other nice thing is that I can say well I can fit Gaussian on top of my my label clusters and because these things are 1d I can simply think them of them as one deductions which are pretty easy to both visualize and work with but I'm also going to fit an unknown distribution between them, because that space had to be filled in mainly because these axis had to be used both to classify but also to reconstruct right so it had to put the yellow task 1 somewhere and the task 2 stuff somewhere such that it can reconstruct the images later on so the best place it found was in between the two clusters for the labeled classes. In other words, I can have an unknown cluster and after that, if I show it a set of new objects it wouldn't misclassify them. For example, I can basically say well it's a ball but it's an unknown color and that's sort of nice because it allows you to conceptualize the world in a better and more human natural way. We don't treat the objects as a single label in their own as “pink ball” for example it's not a label out of ten labels it's sort of a multi-class labeling scenario so that's the model of the architecture. The loss function, the new thing is this classifier here which essentially says that a bunch of these dimensions is going to be responsible for reconstructing but they're also going to be responsible for classifying the different images that disentangle certain qualitative properties of the images and because we do this separation. Learning such representations in a completely unsupervised way is doable and it's actually pretty attractive but it only gets us that much we want to have more insights into what the learned representations mean and if you want to have more control over what's being disentangled over and what we need to leave. The weak labels, because they didn't have that but the way the disentangling works in the unsupervised fashion is that the model sequentially picks factor is a variation that makes the most sense in terms of reconstruction. For example, it would make more sense for the model to start reconstructing a blob that's consistently moving throughout the sort of the image but it's still a blob because in terms of the pixel level loss it makes more sense to give a white blob with the right coordinates instead of a correct digit at the wrong place because the latter one has a bigger error and because author train these things, in expectation, it's sort of having low capacity there it's going to disentangle whatever it makes sense to itself whatever makes sense to the optimization process. That's why if you introduced weak labels it will sort of pinned down the model. Peter-Weyl Theorem and Lie groups are preliminary for the algorithmic summary. In the paper, two algorithms are discussed, the Probabilistic model and Inference & Learning. Before this, the formulation includes Image I that belongs to the Dictionary space. The Image I represented as lie group transformations with matrix $WR(s)W^T\phi\alpha$. Where ϕ belongs to the dictionary of templates and α is the code. Each dictionary template has an L2 norm that preserves the qualitatively unique features without any scaling error. The learning algorithm starts with the initialization of random parameters. Sparse coefficient alpha is initialized with i.i.d exponential distribution random variables. For image I, we want to get sparse code (alpha) and transformation parameter (s). And for batches, we want to learn parameters such as W matrix and phi is the dictionary with l2 norm by maximizing the log-likelihood. Maximum a posteriori estimation is calculated with a gradient ascent algorithm. And then the final approximated gradient is calculated. With the experimentation, the authors first used two synthetic datasets. The former consists of 2D translational images of the MNIST dataset. The latter one consists of rotation and scaling. Finally, each of the 10 digits from the MNIST dataset has one corresponding dictionary learned. The learned transformations are exactly the 2D translation operators and the rotation + scaling operators respectively. Empirically, the author proved that even though they trained with very little rotations and scaling, the model was able to classify the rotated images with more than 270 degrees of rotation. In addition, the authors showed learning of transformed images and spatial patterns of the MNIST images can be done separately however both factors can be used to infer the images at one time. In this unsupervised learning methodology with the synthetic dataset, authors have given theoretical advantages for such cases using lie group and sparse coding method. Authors have claimed that rather than using mathematical models they could have used multi-layered neural networks, however, it could have learned unnecessarily complex features. Hence algorithm learns correct transformations and shapes dictionaries for each of the MNIST digit sets.