Introduction to Machine Learning (67577)

Exercise VI – PCA, kernels, SGD, and DL

Hadar Sharvit – 208287599

**Theoretical Questions**

**PCA**

1. Let us first calculate . Firstly, lets write , where are sampled from some distribution over . If so, we can write

we are left with . From here we can say that given a leading eigenvector of , . In other words, the variance of is the variance of the embedding of into

**Kernels**

1. Given valid kernel , provide a normalized kernel such that for all , :

We define

Firstly, is a valid kernel, as it represents an inner product of the feature map :

Furthermore, for every , as

1. Example of a data set and a feature map where is not linearly separable in but the transformed data is linearly separable in :

Consider , where represents a set of points within some ring with inner radius and outer radius , where . We define the mapping as followed: given some point , . Simply put, maps the point to the same coordinate, with some elevation to it. Since , such mapping provides a set of linearly separable data points in

**Convex optimization**

2. Show that if are convex (, then , () is convex:

Since is convex, we know that . If so, . In other words, is also convex by definition. If so, we have

1. Composition of convex function is not necessarily convex:

Take , . Both and are convex, yet is not.

1. Given convex set , a function is convex is a convex set:

In the first direction, assume is convex. If so, we have . Assume towards contradiction that is not convex, therefore such that . In other words, and , yet . On the other hand, since is convex, then by definition . we have reached a contradiction, therefore must be convex.

In the other direction, assume is convex, therefore we have , i.e., . The claim holds for every , therefore we can choose and , and in that case is convex by definition.

1. Let , , , . Show that if is convex, then so is

From it is sufficient to show that is a convex set.

Consider some pair . By definition of , . Furthermore, since bounds for every , that is, for every , we also have . In other words, for every , which can only happen if is the intersection of all ). Mathematically speaking, . From we know that is convex is a convex set. Since an intersection of convex set is convex we conclude that is convex.

2. Given and , show that the hinge loss function defined by is convex (in ):

we have shown in class that a pointwise max function is convex. This is not enough as we have to show that both and are convex. This is true because they are both linear functions, where one of them is a composition with an affine function – all of those were shown to be convex in class

1. Deduce some :

We need to find a sub-gradient of , that is, some for which for all , we have . we can see that satisfies the claim.

1. Given convex functions, and for all . define by Show that :

Since is a sub-gradient of , we know that , therefore the sum also satisfies . Therefore, by the same logic we have by definition that

1. Given , define . Find a member of for each :

Same as in , we can define for every very similar:

And using we know that

**Practical Questions**

1. Code
2. A. I have initialized a network with 1 hidden layer of size .

The following plot is for varying ’s, batch size and epochs

Chart

Description automatically generatedWe can obviously see that both and provide good accuracy (around success rate, that is ~ accurate classifications). on the other hand, provided poor results, not reaching even accuracy rate as can be seen in the following plot

Another plot is for varying batch sizes, such that in this time I have initialized . In the following plot we can see that performed poorly, compared to the other ones. More specifically, for the accuracy rate was at around , where for both the accuracy rate was at approximately

Chart, line chart

Description automatically generated

B. after some tinkering, I have concluded that the following parameters provide the best accuracy - , (both of those were chosen based on the previous section). Furthermore, I have looked into some different types of architectures – one with single hidden layer with neurons, one with single hidden layer with neurons and another with two single layers of size .

Chart, line chart

Description automatically generated

It is interesting to see that eventually, the network with one large hidden layer (of size 100) performed slightly better then the other two, which had similar results. Having said that, one might prefer the other two architectures, as they reach high accuracies even for small number of epochs, therefore take less time to train. As for specific results, we have reached maximum accuracy of around 93.5%, 95.5% and 94.5% for architecture and respectively