**Part 1: Theory**

1. X,Y are two random variables.  
   1. The covariance of X,Y is:

Where E[X] and E[Y] are the expected values of X,Y respectively.

By using the linearity property of expectation, and because :

If X and Y are statistically independent, then , so

If the covariance is equal to zero, then the correlation is equal to zero because:

* 1. If X and Y are not correlated, i.e., , we can not deduce that X,Y are independent. That's because there are pairs of dependent random variables that satisfies .  
     For example:

X is uniformly distributed in ,

Y satisfies ,  
X,Y are not independent (We will show why in section 2.a)

The covariance of X,Y is:

But:

are not correlated and not independent.

1. Let X be a random variable.   
   1. and are always independent.

**False**

Counter example:

Let X be a uniformly random variable in the range [-1, 1], and .

If Y=1, we can say that X must be either 1 or -1. Meaning that by knowing something on Y we now know something definite on X, and therefore they are **dependent**.

* 1. and are never correlated.

**False**

Let X be a random variable with the following pmf:

* 1. and are always correlated.

**False**

The example from section 1.b can serve as a counter example for this statement:

X is uniformly distributed in ,

Y satisfies ,

The covariance of X, Y is:

But:  
 are not correlated.

* 1. and are never independent.

**False**

Two random variables are independent iff .

Let X be a random variable with the following pmf:

Therefore, with probability .

1. Let X be a 4x4 matrix and .  
     
    C is a symmetric matrix.

An 4x4 symmetric real matrix C is said to be positive-semidefinite if for all .

is a 1x4 vector and is a 4x1 vector. is the dot product which is the square of the Euclidean norm .  
The Euclidian norm is always non-negative so:

1. Matrix is given by:

observation, and every column is a feature.

* 1. Before calculating the covariance matrix of X, we will center the data by subtracting the mean of each feature, so:  
       
        
     The covariance matrix of is:  
     \* This form corresponds to a data matrix where the number of rows represents the number of samples and the number of columns represents the number of features (in contrary to the form we saw in the tutorial).
  2. is a real symmetric matrix and thus, according to the spectral decomposition theorem, can be diagonalized by its eigen matrix .  
     We will start with finding the eigenvalues and their associated eigenvectors.

Eigenvectors associated with eigenvalues solve:

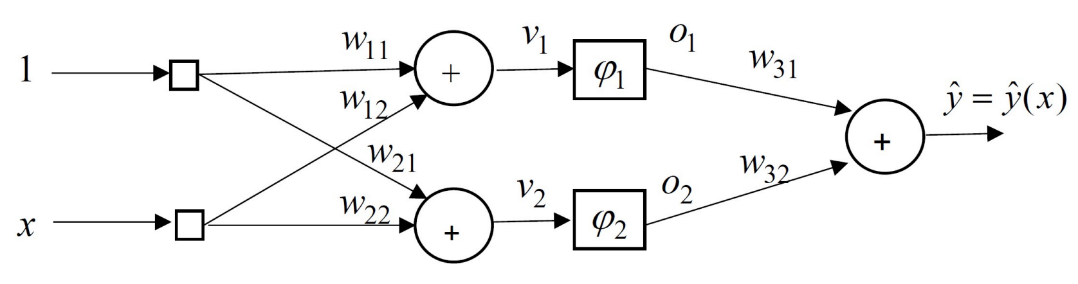
For :

For :

The diagonalizing matrix is:

We can now diagonalize by calculating .

* 1. The principal components are eigenvectors of the centered data's covariance matrix.   
     The variance is maximal for the largest eigenvalue, so the main component (PC1) corresponds to the eigenvalue (the variance) and the eigenvector (the direction).   
     This component carry of the variance of the data.
  2. As said, the main eigenvalue is because it corresponds to maximal variance in the data.
  3. We will now calculate the projection of the new data point on the main component.  
     First, we will normalize the eigenvector corresponds to the main component:  
       
     After normalizing by subtracting the mean from each feature, we can calculate the projection of on the main component as follows:

1. We are given with the following neural network:
   1. For each weights matrix, where is the number of inputs before getting to layer number , and is the number of neurons in layer number . Thus:

Where:

are ReLU, i.e., , so:

* 1. can be explicitly written as follows:

If:

So:

And then:

* 1. The loss is given as follows:

Our model's parameters are so we need to derive according to in order to use it for gradient descent.

Where , .

Where

* 1. The update policy of every weight is done by the gradient descent process:

where , is the learning rate.

* 1. For the following weights initialization ,

We can calculate the first value of the loss for the following sample as follows:  
First, we’ll calculate the predicted y value (:

The loss function is , so: