Assignment 5

Q1.

Y1	2	3	5	7	9
Y2	1	4	0	6	2

step 1: Calculating mean $\bar{Y}_1$ & $\bar{Y}_2$

$$\therefore \bar{Y}1 = \frac{2+3+5+7+9}{5} = \frac{26}{5} = 5.2$$

$$\bar{Y}2 : \frac{1+4+0+6+2}{5} = \frac{13}{5} = 2.6$$

step 2 Mean centering the vectors

$Y1 - \bar{Y}1 = -3.2 \quad -2.2 \quad -0.2 \quad 1.8 \quad 3.8$

$Y2 - \bar{Y}2 = -1.6 \quad 1.4 \quad -2.6 \quad 3.4 \quad -0.6$

step 3: Calculate covariance of mean centered vectors

$$Cov\,(Y1 - \bar{Y}1) = \sum_{i=1}^{n} \frac{(Y1_i - \bar{Y}1)^2}{n-1} = \frac{32.8}{4} = 8.2$$

$$\& \; Cov\,(Y2 - \bar{Y}2) = \sum_{i=1}^{n} \frac{(Y2_i - \bar{Y}2)^2}{n-1} = \frac{23.2}{4} = 5.8$$

$$\& \; Cov\,(Y1, Y2) = \sum_{i=1}^{n} \frac{(Y1_i - \bar{Y}1)(Y2_i - \bar{Y}2)}{n-1} = \frac{6.4}{4} = 1.6$$

$$\therefore S = \begin{array}{c|cc} & Y1 & Y2 \\ \hline Y_1 & 8.2 & 1.6 \\ Y_2 & 1.6 & 5.8 \end{array}$$

step 4: Calculate eigen values & eigen vectors of S

$$|S - \lambda I| = 0 \Rightarrow (8.2-\lambda)(5.8-\lambda) - (1.6)^2 = 0$$

$$(\lambda - 8.2)(\lambda - 5.8) = 2.56$$

$$\lambda^2 + \cancel{47.56} - 14.0\lambda = 2.56$$

$$\lambda^2 - 14\lambda + 45 = 0 \Rightarrow \boxed{\lambda = 9, 5}$$

Consider $\lambda = 9$,

$$S = \begin{bmatrix} 8.2-9 & 1.6 \\ 1.6 & 5.8-9 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix} = 0$$

$$\Rightarrow \quad a_{11}(-0.8) + 1.6\,a_{12} = 0$$

$$1.6\,a_{11} - 3.2\,a_{12} = 0$$

$$\& \quad a_{11}^2 + a_{12}^2 = 1$$

$$\therefore \quad a_{12} = 0.5\,a_{11}$$

$$\therefore \quad a_{11}^2 + (0.5\,a_{11})^2 = 1$$

$$\left.\begin{aligned} \Rightarrow a_{11} &= 0.894 \\ a_{12} &= 0.447 \end{aligned}\right\} \rightarrow \text{Eigen vector } V_1 = \begin{bmatrix} 0.894 \\ 0.447 \end{bmatrix}$$

Consider $\lambda = 5$

$$S = \begin{bmatrix} 8.2-5 & 1.6 \\ 1.6 & 5.8-5 \end{bmatrix} \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} = 0$$

$$\begin{bmatrix} 3.2 & 1.6 \\ 1.6 & 0.8 \end{bmatrix} \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} = 0 \Rightarrow \begin{aligned} 3.2a_{21} + 1.6a_{22} &= 0 \\ 2\times \quad 1.6\,a_{21} + 0.8a_{22} &= 0 \end{aligned}$$

$$\Rightarrow a_{21} = -0.5a_{22}$$

also $a_{21}^2 + a_{22}^2 = 1$

$$\Rightarrow a_{21}^2 + 4a_{21}^2 = 1 \quad \Rightarrow a_{21} = 0.447 \sim 1/\sqrt{5}$$

$$\therefore \text{ Eigen vector } V_2 = \begin{bmatrix} 0.447 \\ 0.894 \end{bmatrix}$$

step 4:

$\therefore$ Principal Components Z_1 & Z_2

$$Z_1 = a_{11}Y_1 + a_{12}Y_2 \quad \& \quad Z_2 = a_{21}Y_1 + a_{22}Y_2$$

$$Z_1 = 0.894Y_1 + 0.447Y_2$$

$$Z_2 = 0.447Y_1 + 0.894Y_2$$

Transforming data into new co-ordinate space:

upon transformat

Y_1	Y_2	Z_1	Z_2
2	1	2.235	1.788
3	4	4.47	4.917
5	0	4.47	2.235
7	6	8.94	8.493
9	2	8.94	5.811

A report

On

"Deep learning for multi-year ENSO forecasts"

by

Mohd Zaki
2019CEZ8233
Department of Civil Engineering
IIT Delhi

1. Summary:

In this paper, following concepts have been discussed:

a) A brief about El Niño Southern Oscillation (ENSO)
b) Convolutional Neural Network (CNN)
c) Transfer Learning (TL)

ENSO is a regional climatic phenomenon peculiar to Southern Pacific ocean and Indian ocean. The monsoonal rains of India are highly influenced by the occurrences of ENSO. Hence, importance of accurate as well as timely predictions of ENSO can not be further emphasized. The existing state of the art (SOTA) methods are unable to predict the ENSO with a lead time of more than twelve months.

There has been a trend of using atmosphere-ocean coupled models, statistical models, SOTA dynamical forecast systems, but none of them is capable of delivering what is essential for ENSO predictions. Since, there is availability of comprehensive atmospheric data and excellent computation power, the use of machine learning is a good candidate for making desired predictions.

The authors have investigated the use of CNN and TL for predicting ENSO events with a lead time of more than twelve months. They have also used the aforementioned machine learning techniques for predicting the type of El Niño which is of three types : i. central-Pacific-type (CP-type), ii. Easter-Pacific-type (EP-type), and iii. A mixture of CP-type and EP-type.

2. Methodology:

a. Dataset :

Temperature records from 1871 to 2017 are used. Since this is a short period of 150 years and CNN models require large data, the authors used the output of climate models which participated in Coupled Model Intercomparison Project Phase 5 (CMIP5).
Setting up training and validation set : A ten-year window is left between latest year in the training set and the earliest year in the validation set to ensure that model learns the climatic phenomena and not crams the trend.

b. CNN architecture :

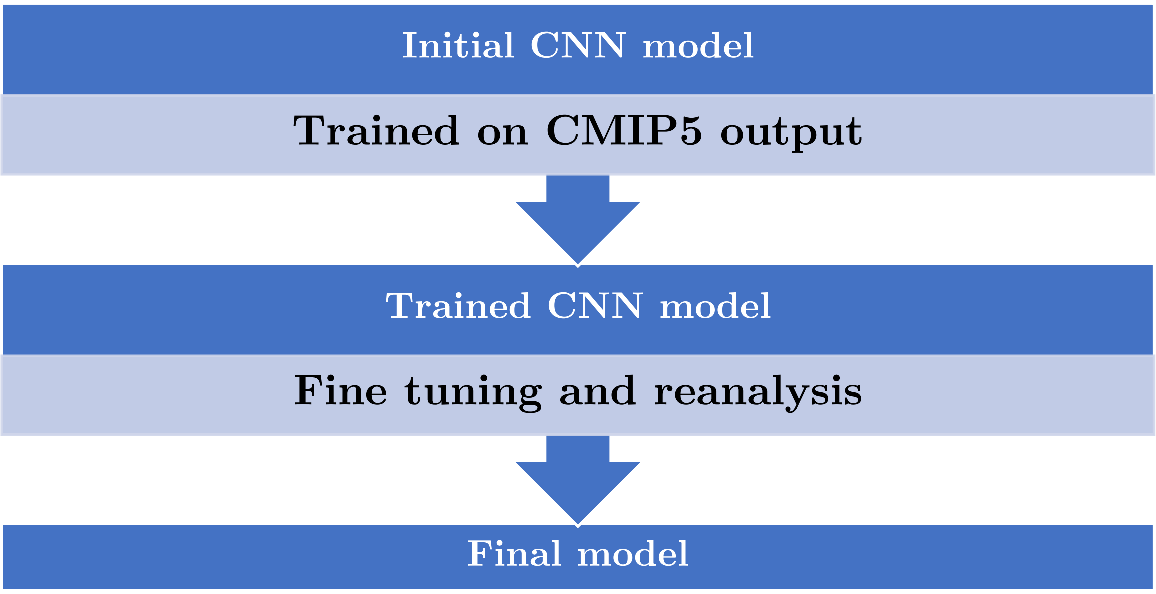
i. Input latitude and longitude extent : 0° –360° E and 55° S–60° N
ii. Input data : Sub surface temperature and heat content (averaged over 300 m depth of ocean)

iii. Output : three-month-averaged Nino3.4 index (area-averaged SST anomaly over 170^o – 120^o W, 5^oN – 5^o S
iv. Ensemble average : 4 CNN setups where M and N can both be 30 or 50.
v. Padding : Zero-padding in convolution process

Input layer	24 x 72 x 6
Convolutional filer	4 x 8 x M
Convolutional layer 1	24 x 72 x M
Max Pooling layer 1	12 x 36 x M
Convolutional layer 2	12 x 36 x M
Max Pooling layer 2	6 x 18 x M
Convolutional layer 3	6 x 18 x M
Fully connected dense layer	N x 1
Output Layer	1 (Nino3.4 index) ((τ + t) (t = 1–23 months)

Table 1. CNN Architecture

c. <u>Transfer learning framework</u> :



3. <u>Critical appraisal</u>:

a. Comments related to technical aspects of ENSO:
    i. The wind circulation in the ocean is also impacted by the condition of surrounding landmasses, which is not taken into account.
    ii. In addition to SST anomaly maps, the wind circulation maps can also be taken as input.
b. Comments related to machine learning procedure:

Hyperparameters	CNN for predicting Nino3.4 Index	CNN for predicting type of ENSO	Transfer learning CNN
Batch size	250	400	20
Epoch	2600	700	20
Dropout rate (for convolutional layers and fully connected layers)	0.9	1.0	1.0

Table 2. Hyperparameters of CNN models

From Table 2., it is clear that transfer learning can help in reducing computational effort because the initial weights used for training the new model are derived from the previously trained model for carrying out a similar task. For example, if we want to build a CNN classifier for identifying cucumber and bananas, and we already have model which can tell if something is cucumber or not, then that model already has weights which can figure out the length and width of the object, so after using transfer learning, we need to learn the weights which can extract the feature corresponding to color and shape of the object.

The authors have not justified the rationale behind choosing a particular dropout rate, activation functions, epoch and batch sizes. There exist some hyperparameter optimizing libraries like OPTUNA and RayTune which can help the researches to efficiently choose the hyperparameters for their models.