

# ID2223 Project

Salman Niazi

Jan 10, 2017

# Project Description

- ▶ Predict the solar radiation near Earth Surface

## Data Samples

- ▶ 1 GB of CSV files with 0.4 million samples
- ▶ A typical sample looks like

| lev | p       | T       | q      | lwhr   |
|-----|---------|---------|--------|--------|
| 0   | 19.231  | -80.0   | 0.0    | 0.122  |
| 1   | 57.692  | -80.0   | 0.0    | 0.451  |
| 2   | 96.154  | -70.874 | 0.029  | -1.229 |
| 3   | 134.615 | -51.083 | 0.262  | -2.732 |
| 4   | 173.077 | -36.489 | 0.977  | -3.429 |
| 5   | 211.538 | -25.816 | 2.211  | -3.574 |
| 6   | 250.0   | -17.87  | 3.756  | -3.536 |
| 7   | 288.462 | -10.404 | 5.431  | -3.802 |
| 8   | 326.923 | -6.608  | 4.226  | -2.198 |
| 9   | 365.385 | -2.388  | 8.776  | -4.203 |
| 10  | 403.846 | 1.264   | 10.375 | -3.567 |
| 11  | 442.308 | 4.462   | 11.895 | -3.146 |
| 12  | 480.769 | 7.318   | 13.347 | -2.829 |
| 13  | 519.231 | 9.903   | 14.733 | -2.598 |

# Propose Solution

- ▶ Regression Problem
  - ▶ with 26 outputs
- ▶ Could be implemented using
  - ▶ Multivariate Regression
  - ▶ Feed Forward Neural Networks
  - ▶ **Convolution Neural Networks**

# Input

- ▶ Features
  - ▶ 26 values for  $T$
  - ▶ 26 values for  $q$
- ▶ Labels
  - ▶ 26 lwhr values

## Input (Cont'd)

- ▶ The input can be morphed into  $26 \times 2$  matrix
  - ▶ did not produce very promising results, as pooling can not shrink the width of the input matrix.
    - ▶ Min MSE achieved was 0.3

|    |    |
|----|----|
| 1  | 1  |
| 2  | 2  |
| 3  | 3  |
| 4  | 4  |
| 5  | 5  |
| .  | .  |
| .  | .  |
| .  | .  |
| 45 | 45 |
| 46 | 46 |
| 47 | 47 |
| 48 | 48 |
| 49 | 49 |
| 50 | 50 |
| 51 | 51 |
| 52 | 52 |

Figure 1: 26x2 Input Matrix

## Input (Cont'd)

- ▶ The input can be morphed into  $8 \times 8$  matrix
  - ▶ padding is needed as there are only 52 input features

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 0  | 0  | 0  | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

Figure 2:  $8 \times 8$  Input Matrix

# Convolution Neural Network Model

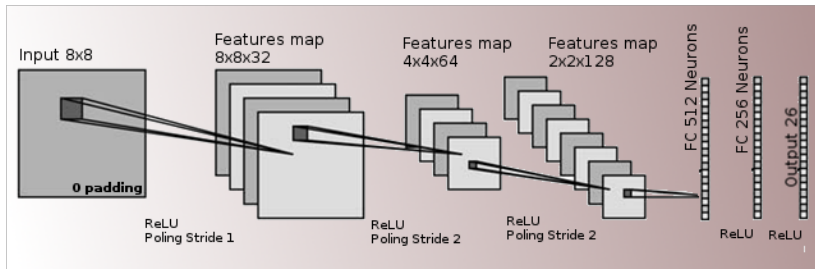


Figure 3: 8 x 8 Input Matrix



## Model Complexity

| Layer | Size    | Memory                         | Weights                                     | Bias |
|-------|---------|--------------------------------|---|------|
| Input | 8x8x1   | 64                             | 0   | 0    |
| CONV  | 8x8x32  | $8 \times 8 \times 32 = 2048$  | $2 \times 2 \times 1 \times 32 = 128$       | 32   |
| POOL  | 8x8x32  | $8 \times 8 \times 32 = 2048$  | 0   | 0    |
| CONV  | 8x8x64  | $8 \times 8 \times 64 = 4096$  | $2 \times 2 \times 1 \times 64 = 256$       | 64   |
| POOL  | 4x4x64  | $4 \times 4 \times 64 = 512$   | 0   | 0    |
| CONV  | 4x4x128 | $4 \times 4 \times 128 = 2048$ | $2 \times 2 \times 1 \times 128 = 512$      | 128  |
| POOL  | 2x2x128 | $2 \times 2 \times 128 = 512$  | 0   | 0    |
| FC    | 1x512   | 512                            | $2 \times 2 \times 128 \times 512 = 262144$ | 512  |
| FC    | 1x256   | 256                            | $512 \times 256 = 131072$                   | 256  |
| OUT   | 1x26    | 26                             | $26 \times 256 = 6656$                      | 26   |

**Total memory =  $413908 \times 4$  bytes (*float32*)  $\times 2$  (back propagation) =  $3311264 = 3.1$  Megabytes**

# Evaluation Setup

- ▶ Training data set size 300,000 (75%).
- ▶ Test data set size 100,000 (25%).
- ▶ Inputs are normalized using max-min scaling
  - ▶  $X_{\text{norm}} = (X - X_{\text{min}}) / (X_{\text{max}} - X_{\text{min}})$
- ▶ Learning Rate 0.05
- ▶ Dropout 0.95
- ▶ Max number of Epochs 30000
- ▶ Batch Size 10
- ▶ Weights were randomly initialized such that the random numbers had *mean=0.1* and *stddev=0.3*
- ▶ Bias were also randomly initialized such that the random numbers had *mean=0* and *stddev=0.03*

# Results

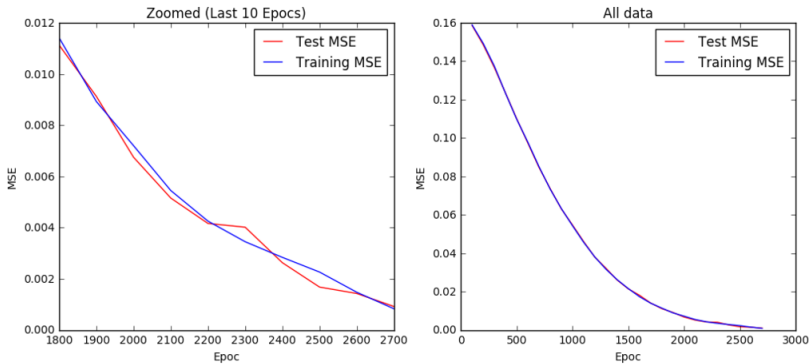


Figure 4: Mean Square Error of the CNN Model

Questions ?