Kaggle Facial KeyPoints Detection

Tingwen Bao & Sanjay Dorairaj

Introduction

Objective to predict 15 facial keypoint positions on face images using DNN and CNN.

Data Points

- Training.zip - 7049 images
- Test.zip - 1783 images

Image Format

- 96*96 pixels in (0,255) color space

Development Framework

- Lasagne, Theano, Jupyter Notebook
- NVIDIA GRID K520 & GeForce GTX TITAN
- AWS G2.2xlarge
- Ubuntu/Centos
- Jupyter Notebook
- Python 2.7/3.6

Features Leveraged

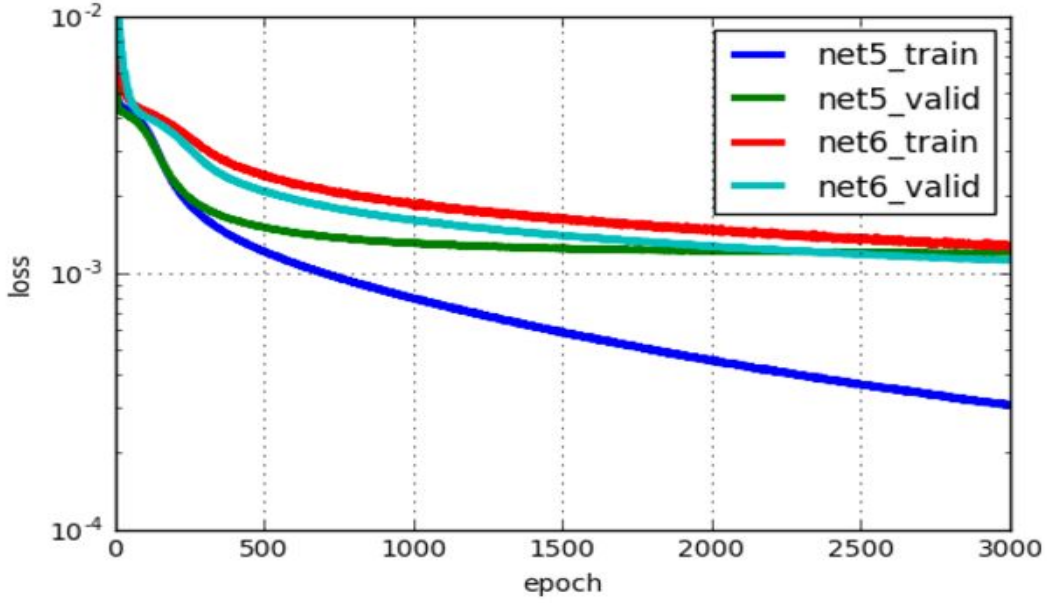
- Deep nets, Convolutional layers, dropouts, max-pooling and turning of their respective hyperparameters

Baseline Model

1. Layers
    1.1. ('input', layers.InputLayer),
    1.2. ('conv1', layers.Conv2DLayer),
    1.3. ('pool1', layers.MaxPool2DLayer),
    1.4. ('dropout1', layers.DropoutLayer),
    1.5. ('conv2', layers.Conv2DLayer),
    1.6. ('pool2', layers.MaxPool2DLayer),
    1.7. ('dropout2', layers.DropoutLayer),
    1.8. ('conv3', layers.Conv2DLayer),
    1.9. ('pool3', layers.MaxPool2DLayer),
    1.10. ('dropout3', layers.DropoutLayer),
    1.11. ('hidden4', layers.DenseLayer),
    1.12. ('dropout4', layers.DropoutLayer),
    1.13. ('hidden5', layers.DenseLayer),
    1.14. ('output', layers.DenseLayer),

- Hyperparameters
    - conv1_num_filters=32, conv1_filter_size=(3, 3),
    - pool1_pool_size=(2, 2),
    - dropout1_p=0.1,
    - conv2_num_filters=64, conv2_filter_size=(2, 2)
    - pool2_pool_size=(2, 2),
    - dropout2_p=0.2,
    - conv3_num_filters=128, conv3_filter_size=(2, 2),
    - pool3_pool_size=(2, 2),
    - dropout3_p=0.3,
    - hidden4_num_units=500,
    - dropout4_p=0.5,
    - hidden5_num_units=500,
    - output_num_units=30,
    - output_nonlinearity=None
    - batch_size =128
    - Adjustable learning rate - 0.03 to 0.0001
    - Adjustable update momentum - 0.9 to 0.999
    - Epochs - 3000

Baseline Model - Results




```
# calculate Kaggle score for this model
np.sqrt(0.00111)*48 # normalize to [-1,1]
```

```
1.5991997998999374
```

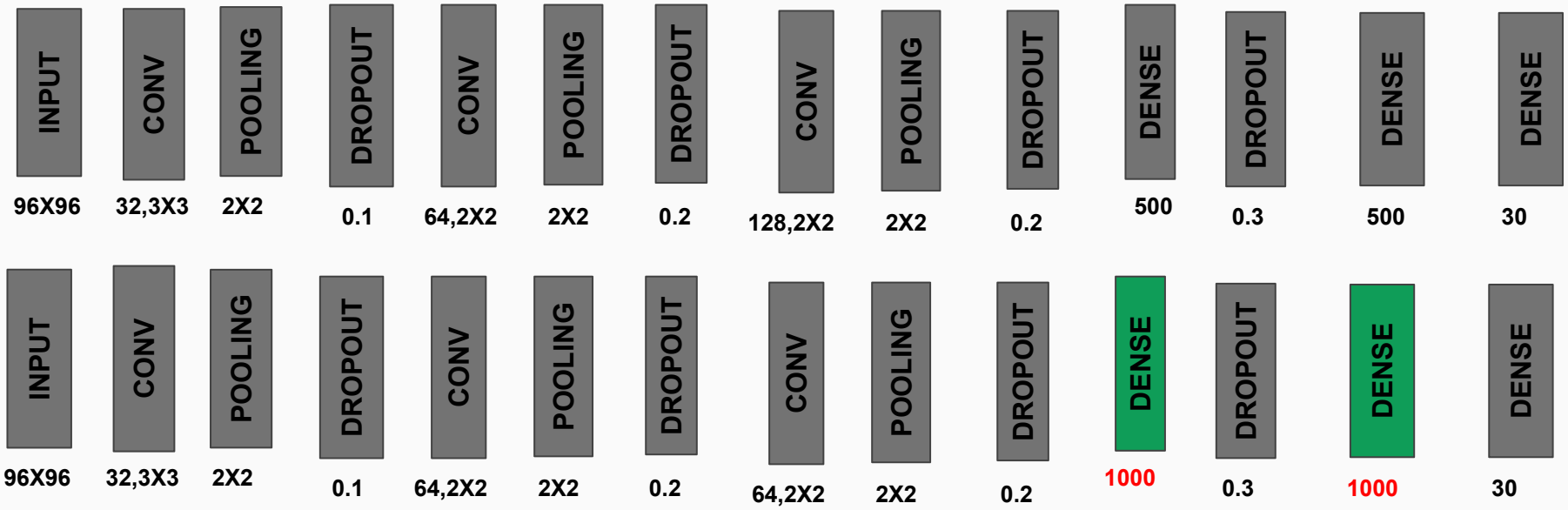
Improvements

Utility Functions

1. **AdjustVariable** - Leverage Theano callback functionality to adjust hyper-parameters during run-time. Default in all nets.
2. **FlipBatchIterator** - Flip alternate 128 batch images to allow the model to generalize better. Default in all nets.
3. **EarlyStopping** - Abort training when validation loss improvements are insignificant
4. **submit_aggregated_models** - Load multiple models from a single pickle file and submit to Kaggle (specialist models)
5. **submit_model** - Load a single model from a pickle file and submit to Kaggle

Increasing the number of hidden layers and epochs relative to the baseline model - net7

How does the baseline model react to increases in hidden layers and epochs?



Epochs increased from 3000 to 10000

Validation loss decreased from 0.00111 to 0.00079

Specialists

Addressing the issue of too many NAs?

- Adapted the idea from Daniel Nourri's blog
- An ensemble of models to maximize available columns
- Created two specialists
  - 7000+ count columns
  - 2000+ count columns
- Trained on pre-trained net7 weights

Column Count and Missing Values

left_eye_center_x	7039
left_eye_center_y	7039
right_eye_center_x	7036
right_eye_center_y	7036
left_eye_inner_corner_x	2271
left_eye_inner_corner_y	2271
left_eye_outer_corner_x	2267
left_eye_outer_corner_y	2267
right_eye_inner_corner_x	2268
right_eye_inner_corner_y	2268
right_eye_outer_corner_x	2268
right_eye_outer_corner_y	2268
left_eyebrow_inner_end_x	2270
left_eyebrow_inner_end_y	2270
left_eyebrow_outer_end_x	2225
left_eyebrow_outer_end_y	2225
right_eyebrow_inner_end_x	2270
right_eyebrow_inner_end_y	2270
right_eyebrow_outer_end_x	2236
right_eyebrow_outer_end_y	2236
nose_tip_x	7049
nose_tip_y	7049
mouth_left_corner_x	2269
mouth_left_corner_y	2269
mouth_right_corner_x	2270
mouth_right_corner_y	2270
mouth_center_top_lip_x	2275
mouth_center_top_lip_y	2275
mouth_center_bottom_lip_x	7016
mouth_center_bottom_lip_y	7016
Image	7049

Specialists Configuration

```
## Group keypoints into specialists based on the data completeness
SPECIALIST_SETTINGS_NEW = [
    dict(
        columns=(
                'left_eye_center_x','left_eye_center_y',
                'right_eye_center_x','right_eye_center_y',
                'nose_tip_x','nose_tip_y',
                'mouth_center_bottom_lip_x','mouth_center_bottom_lip_y',
            ),
        flip_indices=((0, 2), (1, 3)),
        ),

    dict(
        columns=(
                'left_eye_inner_corner_x','left_eye_inner_corner_y',
                'right_eye_inner_corner_x','right_eye_inner_corner_y',
                'left_eye_outer_corner_x','left_eye_outer_corner_y',
                'right_eye_outer_corner_x','right_eye_outer_corner_y',
                'left_eyebrow_inner_end_x','left_eyebrow_inner_end_y',
                'right_eyebrow_inner_end_x','right_eyebrow_inner_end_y',
                'left_eyebrow_outer_end_x','left_eyebrow_outer_end_y',
                'right_eyebrow_outer_end_x','right_eyebrow_outer_end_y',
                'mouth_left_corner_x','mouth_left_corner_y',
                'mouth_right_corner_x','mouth_right_corner_y',
                'mouth_center_top_lip_x','mouth_center_top_lip_y',
            ),
        flip_indices=((0, 2), (1, 3), (4, 6), (5, 7),(6,8),(7,9),(8,10),(9,11),(12,14),(13,15),(16,18),(17,19)),
        ),
    ]
```

Specialists - Performance

- Model identical to previous model - net7
- Kaggle score of specialists model was 3.45
  - Less than net7
- RMSE Score of first specialist: 2.17
- RMSE Score of second specialist: 4.34
- Worse performance than net7

Possible Reasons

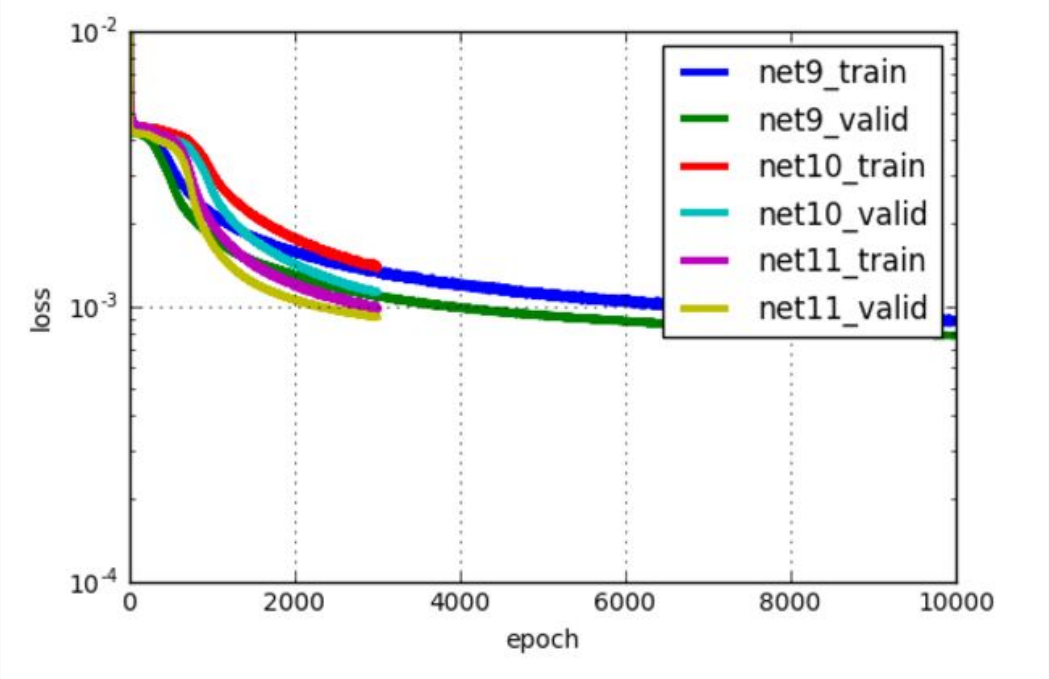
- Overfitting due to additional data-points

Train deeper nets - net10 and net11

- net10
  - Added an additional convolution/pooling/dropout layer
  - Reduced epochs to 3000 (mainly to ensure we get results back in a timely manner)
- net11
  - Questioning the importance of hidden layers
  - Net10 - Reduced from 3 to 2 hidden layers

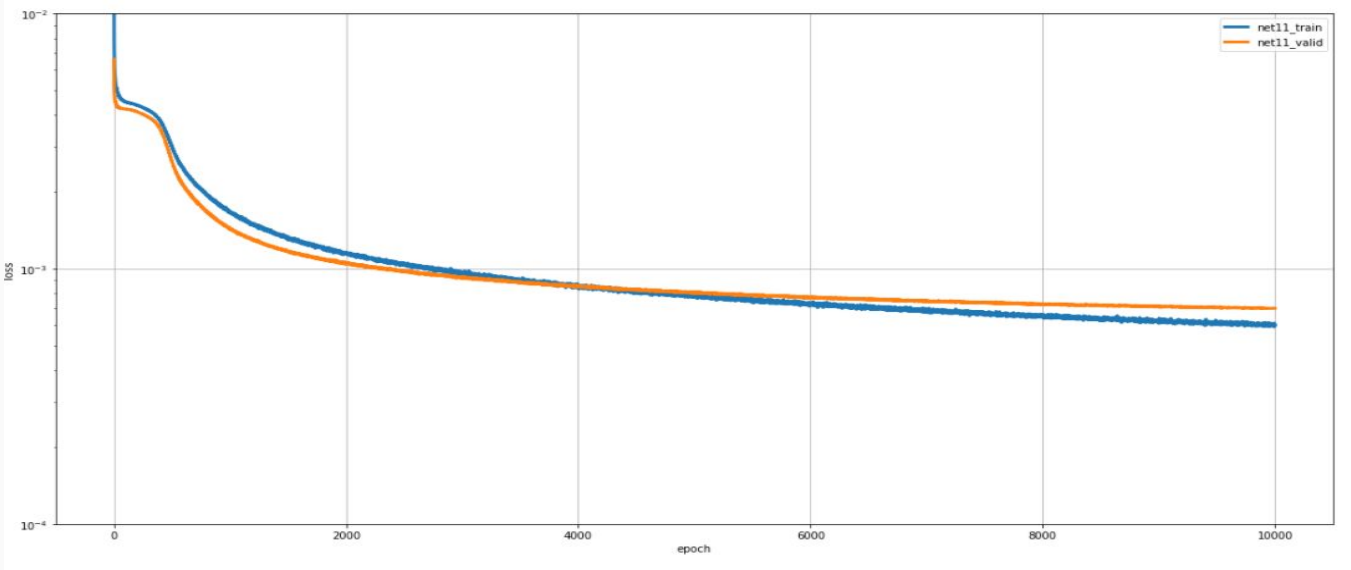
Deeper Nets - Performance

- Adding a single convolution layer does not show significant improvement.
- Training time the same between net10 and net11 despite fewer hidden layers in net11
- Net11 appears to show a trend towards better validation loss relative to Net10



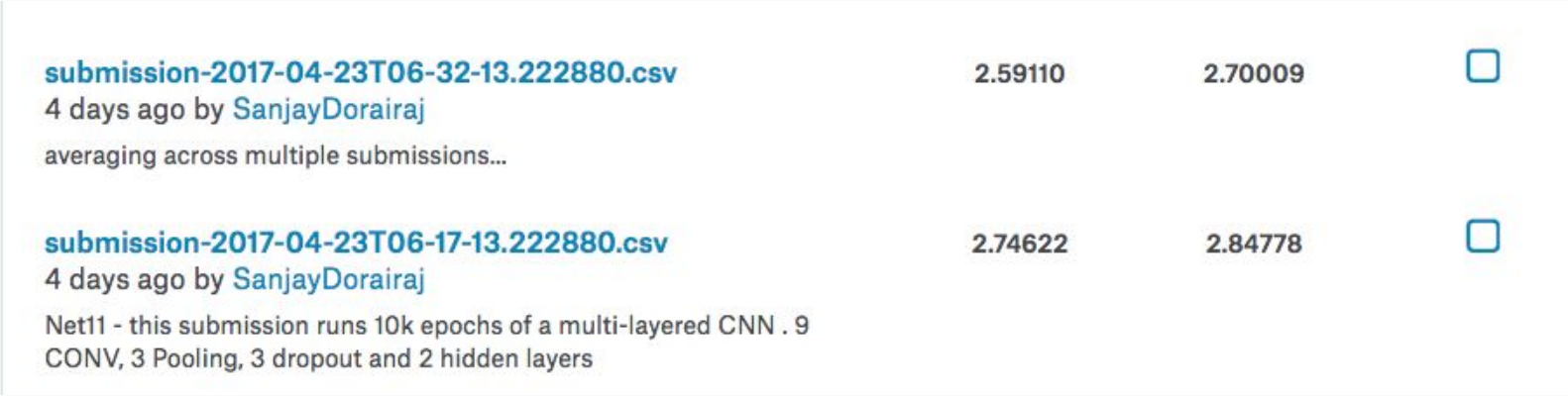
Best Model

- **We decided to chase net11 further down and increased the number of epochs to 10000**
- **We ended with a validation loss of 0.0007. Kaggle score - 2.74**



Hacking Kaggle

- Ridiculously high scores on Kaggle
- Is it possible that it was being hacked?
- We tried a simple experiment by submitting after averaging across past submissions
- And indeed...we got our best Kaggle score!



Result Summary

Net	Improvement	Epoch	Train_loss	Val_loss
Net1	[2 layer simple NN]	1000	0.00151	0.00277
Net2	Convolution	1000	0.00120	0.00157
Net3	Flip faces	3000	0.00078	0.00124
Net4	Changed learning rate and momentum	3000	0.00013	0.00147
Net5	Net3 + Net4 improvement	3000	0.00030	0.00119
Net6 [Baseline]	Add dropout layer	3000	0.00128	0.00111
Net7	Increase number of hidden layer and epoch	10000	0.00074	0.00079
Net8	Train keypoints in groups	5000 with early stop	0.00173, 0.00844	0.00205, 0.00821
Net9	Increase the number of convolution layer	10000	0.00089	0.00078
Net10	Even deeper net with more hidden layer	3000	0.00098	0.00091
Net11	Even deeper net with no more hidden layer	10000	0.00060316	0.00070

Kaggle Score Summary

Submission and Description	Private Score	Public Score	Use for Final Score
submission-2017-04-23T12-43-28.076089.csv 3 days ago by SanjayDorairaj net7 - Initial neural net model with 10000 epochs	2.85825	2.94950	☐
submission-2017-04-23T06-32-13.222880.csv 4 days ago by SanjayDorairaj averaging across multiple submissions...	2.59110	2.70009	☐
submission-2017-04-23T06-17-13.222880.csv 4 days ago by SanjayDorairaj Net11 - this submission runs 10k epochs of a multi-layered CNN . 9 CONV, 3 Pooling, 3 dropout and 2 hidden layers	2.74622	2.84778	☐
submission-2017-04-22T11-07-25.895212.csv 5 days ago by SanjayDorairaj Submission based on specialists used by dnouri with slight changes to increase epoch size and remove Early Stopping	3.45306	3.57878	☐
submission-2017-04-20T12-28-10.691036.csv 7 days ago by SanjayDorairaj this submission reuses work by done dnouri and is only for testing purposes.	3.05861	3.13756	☐

Learnings

- The deeper the net, the better the performance appears to get
- Lots of trial and error in terms of tuning hyperparameters and determing the right model
- A good validation loss does not necessarily mean a good Kaggle score
  - One is computed on the training data and the other on the test data
- Specialists do not necessarily improve model performance
- Early Stopping implementation can cause training to stop prematurely
- Kaggle scores can be hacked
- Kaggle information can be wrong/outdated. Check blogs for changes.
- Pickle files are a huge help given long training times.
- Pickle file incompatibility between Python 2.7 and Python 3.6
- Pickle file stops working when epochs and hidden layers increase.
  - Solution: `sys.setrecursionlimit(10000)`
- Importance of optimizing models for inference and maintainability.

Thanks!

Tingwen Bao
Sanjay Dorairaj

Useful Links & References

- GitHub repository - https://github.com/tingwenbao/Facial_KeyPoints_Detection.git. Includes two top-level folders - Baseline and Improvements and captures the baseline version and the improvements respectively.
- Kaggle link - https://www.kaggle.com/c/facial-keypoints-detection
- Reference Tutorial - http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/
- Setting up a VM for this project on AWS- http://markus.com/install-theano-on-aws/#comment-3206748383