Facial Keypoints Detection

Joe Izenman
Thong Bui

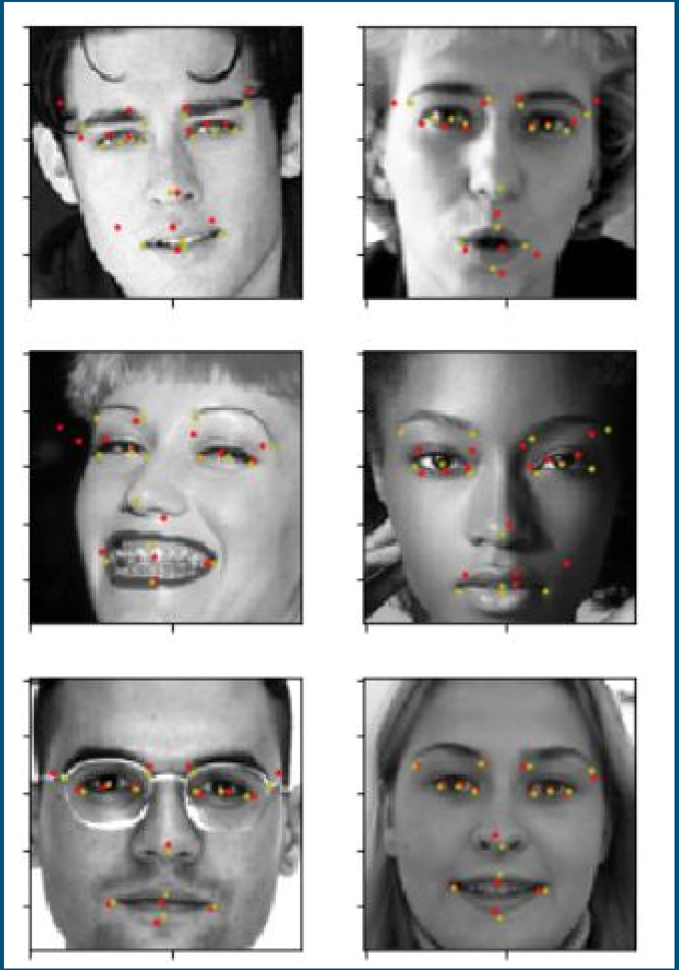
Goal - Setup

Use convolutional neural nets to predict facial keypoints from pictures of human faces with help from Daniel Nouri tutorial

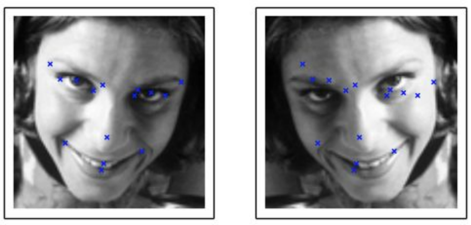
Technology:

- Lasagne + Theano
- AWS GPU instance g2.2xlarge

Data: 9216 examples, but only 30% (2,140) has complete target values.



Overfitting



Overfitting solution:

- Data augmentation: increase number of training samples
    - Create new data set using horizontal flip
- Dropout is a popular regularization technique:
    - Adding a DropoutLayer between each group
    - Increasing number of units in each hidden layer
    - Increasing number of epochs

Other Tuning for Training

Change learning rate and momentum to shorten training time:

- For each epoch, decrease learning rate and increase momentum
- Training faster
- Decrease error rate

Training specialists

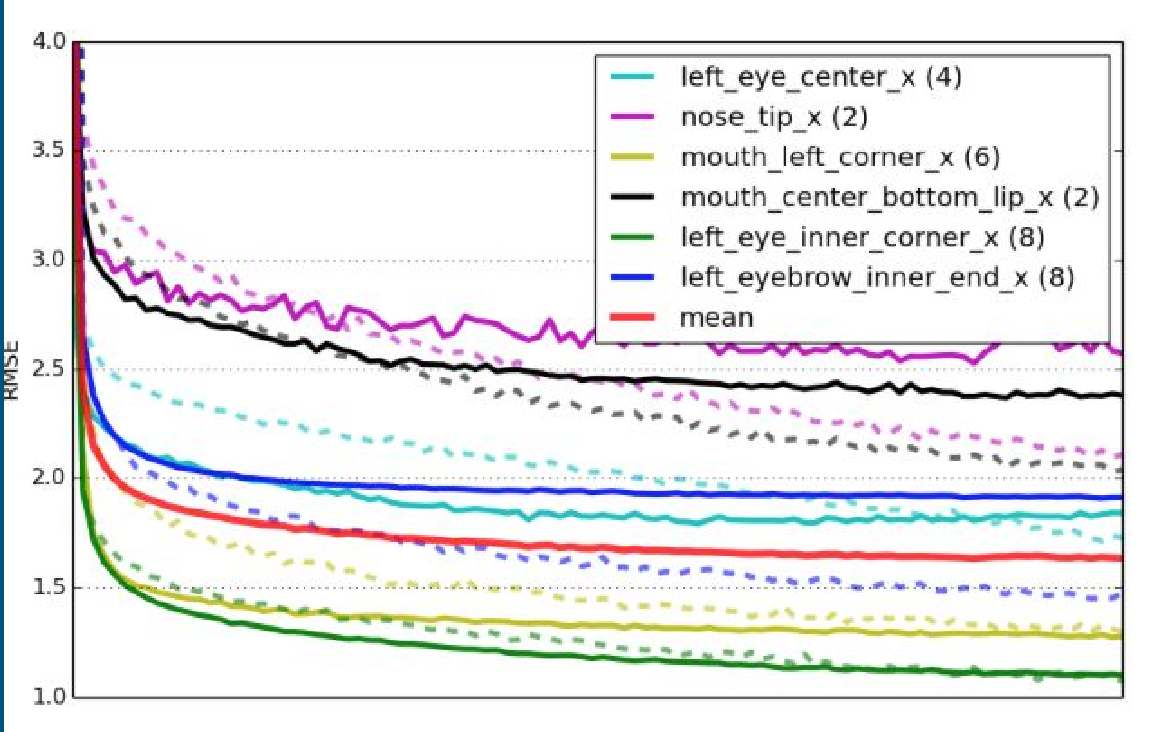
So far we only used 30% data for training. Next, we used remaining 70% that was not used for training earlier, to train for different set of target values and created 6 models:

- Left_eye_center, right_eye_center
- Nose_tip
- Mouth_left_corner, mouth_right_corner, mouth_center_top_lip
- Mouth_center_bottom_lip
- Left/right_eye_inner_corner, left/right_eye_outer_corner
- Left/right_eyebrow_inner_end, left/right_eyebrow_outer_end

Training specialists result

Reusing the weights from the complete cases model:

- Converge faster
- Better generalization
- Pre-training: regularizer



Other configuration settings

Are there other tweaks we have tried to improve?

- Adding another convolutional and fully connected layer
- Adjusting dropout rates based on the training/validation loss differences

What are the results of these tweaks?

- Adding layers caused rapid plateauing at a significantly worse error rate.
- Augmented dropout brought the training and validation values closer together, but worsened the final Kaggle score

Issues encountered

Performance:

- It takes a long time to train a model

Resource issues came up at the end of the training so we lost another day or 2 to retrain each model after fixing the issue:

- Disk space
- GPU

Conclusion

Neural Networks:

- Complicated training
- Needs lots of training
- Difficult to parallelize

But the trade-off is significant

- State of the art for performance
- Discover useful features for complex inputs, in this case, images

Scores

- Score from running the tutorial exactly: 2.15080
- Score with our tuning attempts: 2.16566