Facial Keypoint Detection Improvement

By Tingwen & Sanjay Dorairaj

W207-5 Spring 2017

Final Project

This notebook contains the steps taken towards making additional improvements to the Facial Keypoint Detection Kaggle Challenge.

For the baseline section, we had started from a 2 layer simple neural network and tested some of the additional features from the python blog. The baseline we set was a val_loss of 0.00111 which translate to RMSE of 1.59920. And the baseline model contains 3 convolutions,pooling layers, and dropout layers and 2 fully connected layer with data augmentation and changed learning rate.

In this notebook, we are going to start from this baseline and try to improve the model in various ways.

Note that several concepts used in this exercise are adapted from Daniel Nouri's blog post at http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/

System Configuration

GPU Configuration

1. GeForce GTX TITAN X
2. AWS - GRID K520 g2.2xlarge

Software Configuration

1. Lasagne/Theano
2. Ubuntu 14.04/16.04

Reference Material

1. 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 respectivel.
2. Kaggle link - https://www.kaggle.com/c/facial-keypoints-detection
3. Reference Tutorial - http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/
4. Setting up a VM for this project on AWS- http://markus.com/install-theano-on-aws/#comment-[illegible]3383

1. Load Data, Libraries, and Functions

In [4]:
```
## import libraries

import warnings
warnings.filterwarnings("ignore")

import os
import numpy as np
from pandas.io.parsers import read_csv
from sklearn.utils import shuffle
import sys
import matplotlib.pyplot as pyplot
import cPickle as pickle
#import pickle
from nolearn.lasagne import BatchIterator
import lasagne
from lasagne import layers
from lasagne.updates import nesterov_momentum
from nolearn.lasagne import NeuralNet
from lasagne.updates import adam

# set recursion limit due to handle updates to pickle files for the large
network models
sys.setrecursionlimit(100000)
```

```
Using gpu device 0: GeForce GTX TITAN X (CNMeM is enabled with initial size
: 95.0% of memory, cuDNN 5105)
/usr/lib64/python2.7/site-packages/sklearn/cross_validation.py:44: Deprecat
ionWarning: This module was deprecated in version 0.18 in favor of the mode
l_selection module into which all the refactored classes and functions are
moved. Also note that the interface of the new CV iterators are different f
rom that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

In [5]:
```
# Display python version
sys.version
sys.version_info
```

Out[5]:
```
sys.version_info(major=2, minor=7, micro=5, releaselevel='final', serial=0)
```

2. Loading the training and test data set

This section includes two methods - load and load2d, that load the train and test data.

The load method has the parameters

- test - boolean value indicating whether to load the training data or the test data
- cols - used when specific columns need to be loaded
- fillNA - boolean value indicates whether NA cells must be filled with the corresponding value from the previous row.

The load2d method uses the load method to load the data but reshapes the data as a 2d array

In [6]:
```
#change the directory to where you put the datasets

FTRAIN = '~/Facial_KeyPoints_Detection/training.csv'
FTEST = '~/Facial_KeyPoints_Detection/test.csv'

def load(test=False, cols=None, fillNA=False):
    """Loads data from FTEST if *test* is True, otherwise from FTRAIN.
    Pass a list of *cols* if you're only interested in a subset of the
    target columns.
    """
    fname = FTEST if test else FTRAIN
    df = read_csv(os.path.expanduser(fname))  # load pandas dataframe

    # The Image column has pixel values separated by space; convert
    # the values to numpy arrays:
    df['Image'] = df['Image'].apply(lambda im: np.fromstring(im, sep=' '))

    if cols:  # get a subset of columns
        df = df[list(cols) + ['Image']]

    print(df.count())  # prints the number of values for each column

    if  fillNA:
        df = df.fillna(method='ffill')
    else:
        df = df.dropna()  # drop all rows that have missing values in them

    print(df.count())  # prints the number of values for each column

    X = np.vstack(df['Image'].values) / 255.  # scale pixel values to [0,
1]
    X = X.astype(np.float32)

    if not test:  # only FTRAIN has any target columns
        y = df[df.columns[:-1]].values
        y = (y - 48) / 48  # scale target coordinates to [-1, 1]
        X, y = shuffle(X, y, random_state=42)  # shuffle train data
        y = y.astype(np.float32)
    else:
        y = None

    return X, y

def load2d(test=False, cols=None):
    X, y = load(test=test, cols=cols)
    X = X.reshape(-1, 1, 96, 96)
    return X, y
```

3. Utility Functions

In this section several utility functions that help with the project are defined.

float32

This function typecasts a numpy array as a float32. It is primarily used to to ensure that there are no type incompatibilities.

AdjustVariable

This method is used as a callback that allows for a cleaner way to update parameters used by the Neural Network for each epoch.

AdjustVariable has the following parameters

- NeuralNet parameter ame
- start value
- stop value

Example:

Below is an example of usage where the parameter in each epoch for update_learning_rate and update_momentum is a linearly increasing value between the start and stop values.

```
AdjustVariable('update_learning_rate', start=0.03, stop=0.0001)
AdjustVariable('update_momentum', start=0.9, stop=0.999)
```

FlipBatchIterator

This method flips half the images in a batch allowing the neural net to generalize better. The input is the mini-batch size.

EarlyStopping

This method enforces an early stopping criteria to allow the Neural Network to stop when a specific early stopping criteria is met.

Early Stopping has the following parameters

- patience - if the validation loss is more than that observed earlier then we re-initialize the net with the prior best value and stop training. the patience parameter tells how far back to look to see if a better validation result was observed.

In [7]:
```
def float32(k):
    return np.cast['float32'](k)

class AdjustVariable(object):
    def __init__(self, name, start=0.03, stop=0.001):
        self.name = name
        self.start, self.stop = start, stop
        self.ls = None

    def __call__(self, nn, train_history):
        if self.ls is None:
            self.ls = np.linspace(self.start, self.stop, nn.max_epochs)
```

```
        epoch = train_history[-1]['epoch']
        new_value = float32(self.ls[epoch - 1])
        getattr(nn, self.name).set_value(new_value)

class FlipBatchIterator(BatchIterator):
    flip_indices = [
        (0, 2), (1, 3),
        (4, 8), (5, 9), (6, 10), (7, 11),
        (12, 16), (13, 17), (14, 18), (15, 19),
        (22, 24), (23, 25),
        ]

    def transform(self, Xb, yb):
        Xb, yb = super(FlipBatchIterator, self).transform(Xb, yb)

        # Flip half of the images in this batch at random:
        bs = Xb.shape[0]
        indices = np.random.choice(bs, bs / 2, replace=False)
        Xb[indices] = Xb[indices, :, :, ::-1]

        if yb is not None:
            # Horizontal flip of all x coordinates:
            yb[indices, ::2] = yb[indices, ::2] * -1

            # Swap places, e.g. left_eye_center_x -> right_eye_center_x
            for a, b in self.flip_indices:
                yb[indices, a], yb[indices, b] = (
                    yb[indices, b], yb[indices, a])

        return Xb, yb

## Define early stop method to reduce training time
class EarlyStopping(object):
    def __init__(self, patience=100):
        self.patience = patience
        self.best_valid = np.inf
        self.best_valid_epoch = 0
        self.best_weights = None

    def __call__(self, nn, train_history):
        current_valid = train_history[-1]['valid_loss']
        current_epoch = train_history[-1]['epoch']
        if current_valid < self.best_valid:
            self.best_valid = current_valid
            self.best_valid_epoch = current_epoch
            self.best_weights = nn.get_all_params_values()
        elif self.best_valid_epoch + self.patience < current_epoch:
            print("Early stopping.")
            print("Best valid loss was {:.6f} at epoch {}.".format(
                self.best_valid, self.best_valid_epoch))
            nn.load_params_from(self.best_weights)
            raise StopIteration()
```

4. Revisiting Baseline Model Performance

In [6]:
```
import theano

net6 = NeuralNet(
    layers=[
        ('input', layers.InputLayer),
        ('conv1', layers.Conv2DLayer),
        ('pool1', layers.MaxPool2DLayer),
        ('dropout1', layers.DropoutLayer),
        ('conv2', layers.Conv2DLayer),
        ('pool2', layers.MaxPool2DLayer),
        ('dropout2', layers.DropoutLayer),
        ('conv3', layers.Conv2DLayer),
        ('pool3', layers.MaxPool2DLayer),
        ('dropout3', layers.DropoutLayer),
        ('hidden4', layers.DenseLayer),
        ('dropout4', layers.DropoutLayer),
        ('hidden5', layers.DenseLayer),
        ('output', layers.DenseLayer),
        ],
    input_shape=(None, 1, 96, 96),
    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,

    update_learning_rate=theano.shared(float32(0.03)),
    update_momentum=theano.shared(float32(0.9)),

    regression=True,
    batch_iterator_train=FlipBatchIterator(batch_size=128),
    on_epoch_finished=[
        AdjustVariable('update_learning_rate', start=0.03, stop=0.0001),
        AdjustVariable('update_momentum', start=0.9, stop=0.999),
        ],
    max_epochs=3000,
    verbose=1,
    )

sys.setrecursionlimit(10000)

X, y = load2d()
net6.fit(X, y)

with open('net6.pickle', 'wb') as f:
    pickle.dump(net6, f, -1)
```

```
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
dtype: int64
left_eye_center_x            2140
left_eye_center_y            2140
right_eye_center_x           2140
right_eye_center_y           2140
left_eye_inner_corner_x      2140
left_eye_inner_corner_y      2140
left_eye_outer_corner_x      2140
left_eye_outer_corner_y      2140
right_eye_inner_corner_x     2140
right_eye_inner_corner_y     2140
right_eye_outer_corner_x     2140
right_eye_outer_corner_y     2140
left_eyebrow_inner_end_x     2140
left_eyebrow_inner_end_y     2140
left_eyebrow_outer_end_x     2140
left_eyebrow_outer_end_y     2140
right_eyebrow_inner_end_x    2140
right_eyebrow_inner_end_y    2140
right_eyebrow_outer_end_x    2140
right_eyebrow_outer_end_y    2140
nose_tip_x                   2140
nose_tip_y                   2140
mouth_left_corner_x          2140
mouth_left_corner_y          2140
mouth_right_corner_x         2140
mouth_right_corner_y         2140
```

```
mouth_center_top_lip_x       2140
mouth_center_top_lip_y       2140
mouth_center_bottom_lip_x    2140
mouth_center_bottom_lip_y    2140
Image                        2140
dtype: int64
# Neural Network with 8051502 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1     32x94x94
  2  pool1     32x47x47
  3  dropout1  32x47x47
  4  conv2     64x46x46
  5  pool2     64x23x23
  6  dropout2  64x23x23
  7  conv3     128x22x22
  8  pool3     128x11x11
  9  dropout3  128x11x11
 10  hidden4   500
 11  dropout4  500
 12  hidden5   500
 13  output    30

  epoch    trn loss    val loss    trn/val  dur
-------  ----------  ----------  ---------  -----
      1     0.07450     0.03014    2.47207  0.82s
      2     0.01674     0.01764    0.94918  0.79s
      3     0.01056     0.01244    0.84868  0.79s
      4     0.00869     0.00907    0.95766  0.80s
      5     0.00791     0.00865    0.91436  0.80s
      6     0.00752     0.00766    0.98217  0.81s
      7     0.00706     0.00764    0.92509  0.80s
      8     0.00684     0.00711    0.96247  0.80s
      9     0.00660     0.00702    0.94007  0.80s
     10     0.00653     0.00685    0.95333  0.87s
     11     0.00632     0.00690    0.91611  0.82s
     12     0.00622     0.00661    0.94033  0.82s
     13     0.00611     0.00642    0.95103  0.82s
     14     0.00592     0.00669    0.88403  0.82s
     15     0.00591     0.00634    0.93294  0.82s
     16     0.00584     0.00615    0.94840  0.82s
     17     0.00569     0.00605    0.94142  0.82s
     18     0.00569     0.00596    0.95558  0.82s
     19     0.00554     0.00598    0.92700  0.83s
     20     0.00550     0.00586    0.93830  0.82s
     21     0.00547     0.00560    0.97658  0.82s
     22     0.00535     0.00582    0.91799  0.82s
     23     0.00527     0.00558    0.94485  0.83s
     24     0.00527     0.00550    0.95749  0.83s
     25     0.00521     0.00550    0.94752  0.82s
     26     0.00525     0.00521    1.00619  0.83s
     27     0.00517     0.00533    0.96962  0.83s
     28     0.00511     0.00535    0.95506  0.82s
```

```
29     0.00514    0.00511    1.00631  0.85s
30     0.00506    0.00496    1.01907  0.85s
31     0.00502    0.00502    0.99928  0.81s
32     0.00499    0.00500    0.99826  0.80s
33     0.00500    0.00493    1.01284  0.80s
34     0.00491    0.00491    1.00138  0.80s
35     0.00492    0.00484    1.01635  0.81s
36     0.00492    0.00476    1.03242  0.81s
37     0.00487    0.00468    1.04051  0.80s
38     0.00486    0.00478    1.01666  0.80s
39     0.00484    0.00468    1.03535  0.80s
40     0.00480    0.00461    1.04078  0.89s
41     0.00478    0.00464    1.02985  0.82s
42     0.00476    0.00463    1.02791  0.82s
43     0.00476    0.00458    1.04044  0.83s
44     0.00473    0.00457    1.03396  0.83s
45     0.00469    0.00452    1.03929  0.82s
46     0.00474    0.00451    1.05158  0.83s
47     0.00466    0.00447    1.04286  0.83s
48     0.00469    0.00445    1.05398  0.82s
49     0.00465    0.00444    1.04654  0.82s
50     0.00466    0.00439    1.06149  0.82s
51     0.00463    0.00437    1.06048  0.82s
52     0.00465    0.00434    1.07181  0.82s
53     0.00462    0.00433    1.06877  0.86s
54     0.00464    0.00434    1.06947  0.80s
55     0.00461    0.00434    1.06199  0.80s
56     0.00459    0.00431    1.06399  0.81s
57     0.00459    0.00431    1.06663  0.80s
```

RMSE Score

In [12]:
```
### RMSE score of baseline model
validation_loss = 0.00114

np.sqrt(validation_loss)*48 # normalize to [-1,1]
```

Out[12]: 1.6206665295488767

5. Code for submitting predictions to Kaggle

In this section we implement two utility functions to create Kaggle submission files. The first function creates a Kaggle submission from a single Neural Net model. The second function creates a Kaggle submission by aggregating multiple neural network models.

In [26]:
```
from pandas import DataFrame
from pandas.io.parsers import read_csv
from datetime import datetime

FLOOKUP = '~/data/kaggle-facial-keypoint-detection/IdLookupTable.csv'

def submit_aggregated_models(fname_specialists):
    with open(fname_specialists, 'rb') as f:
```

```
        specialists = pickle.load(f)

    X = load2d(test=True)[0]
    y_pred = np.empty((X.shape[0], 0))

    for model in specialists.values():
        y_pred1 = model.predict(X)
        y_pred = np.hstack([y_pred, y_pred1])

    columns = ()
    for cols in specialists.keys():
        columns += cols

    y_pred2 = y_pred * 48 + 48
    y_pred2 = y_pred2.clip(0, 96)
    df = DataFrame(y_pred2, columns=columns)

    lookup_table = read_csv(os.path.expanduser(FLOOKUP))
    values = []

    for index, row in lookup_table.iterrows():
        values.append((
            row['RowId'],
            df.ix[row.ImageId - 1][row.FeatureName],
            ))

    now_str = datetime.now().isoformat().replace(':', '-')
    submission = DataFrame(values, columns=('RowId', 'Location'))
    filename = 'submission-{}.csv'.format(now_str)
    submission.to_csv(filename, index=False)
    print("Wrote {}".format(filename))

def submit_model(fname):

    columns=[
            'left_eye_center_x',
            'left_eye_center_y',
            'right_eye_center_x',
            'right_eye_center_y',
            'left_eye_inner_corner_x',
            'left_eye_inner_corner_y',
            'left_eye_outer_corner_x',
            'left_eye_outer_corner_y',
            'right_eye_inner_corner_x',
            'right_eye_inner_corner_y',
            'right_eye_outer_corner_x',
            'right_eye_outer_corner_y',
            'left_eyebrow_inner_end_x',
            'left_eyebrow_inner_end_y',
            'left_eyebrow_outer_end_x',
            'left_eyebrow_outer_end_y',
            'right_eyebrow_inner_end_x',
            'right_eyebrow_inner_end_y',
            'right_eyebrow_outer_end_x',
            'right_eyebrow_outer_end_y',
            'nose_tip_x',
            'nose_tip_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',
            'mouth_center_bottom_lip_x',
            'mouth_center_bottom_lip_y'
    ]

    with open(fname, 'rb') as f:
        net = pickle.load(f)

    X = load2d(test=True)[0]
    y_pred = np.empty((X.shape[0], 0))

    y_pred1 = net.predict(X)
    y_pred = np.hstack([y_pred, y_pred1])

    y_pred2 = y_pred * 48 + 48
    y_pred2 = y_pred2.clip(0, 96)

    print y_pred2.shape

    lookup_table = read_csv(os.path.expanduser(FLOOKUP))
    values = []

    print lookup_table

    lookup_table = read_csv(os.path.expanduser(FLOOKUP))
    values = []

    df = DataFrame(y_pred2,columns=columns)

    for index, row in lookup_table.iterrows():
        values.append((
            row['RowId'],
            df.ix[row.ImageId - 1][row.FeatureName],
            ))

    now_str = datetime.now().isoformat().replace(':', '-')
    submission = DataFrame(values, columns=('RowId', 'Location'))
    filename = 'submission-{}.csv'.format(now_str)
    submission.to_csv(filename, index=False)
    print("Wrote {}".format(filename))
```

6. Increasing Number of Hidden Layer and Epoch

In this step, we attempt to improve our model relative to the baseline by increasing the number of hidden layers and epochs. We set a large epoch size of 10,000 and increase the number of hidden layers to get a sense as to when the model would start to overfit and investigate how our model converges towards the global minimum. We limit our investigation to 10,000 epochs in order to consider time and resource constraints.

In [27]:
```
import theano
```

```
import collections

# increase las two hidden layer from 500 to 1000 and increase max epochs t
o 10000#
net7 = NeuralNet(
    layers=[
        ('input', layers.InputLayer),
        ('conv1', layers.Conv2DLayer),
        ('pool1', layers.MaxPool2DLayer),
        ('dropout1', layers.DropoutLayer),
        ('conv2', layers.Conv2DLayer),
        ('pool2', layers.MaxPool2DLayer),
        ('dropout2', layers.DropoutLayer),
        ('conv3', layers.Conv2DLayer),
        ('pool3', layers.MaxPool2DLayer),
        ('dropout3', layers.DropoutLayer),
        ('hidden4', layers.DenseLayer),
        ('dropout4', layers.DropoutLayer),
        ('hidden5', layers.DenseLayer),
        ('output', layers.DenseLayer),
        ],
    input_shape=(None, 1, 96, 96),
    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=1000,  ## increased from 500 to 1000dropout4_p=0.5,
    hidden5_num_units=1000,  ## increased from 500 to 1000
    output_num_units=30,
    output_nonlinearity=None,

    update_learning_rate=theano.shared(float32(0.03)),
    update_momentum=theano.shared(float32(0.9)),

    regression=True,
    batch_iterator_train=FlipBatchIterator(batch_size=128),
    on_epoch_finished=[
        AdjustVariable('update_learning_rate', start=0.03, stop=0.0001),
        AdjustVariable('update_momentum', start=0.9, stop=0.999),
        ],
    max_epochs=10000,  ## increased from 3000 to 10000
    verbose=1,
    )

sys.setrecursionlimit(10000)

X, y = load2d()
net7.fit(X, y)

with open('net7_10000epochs.pickle', 'wb') as f:
    pickle.dump(net7, f, -1)
```

```
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
dtype: int64
left_eye_center_x            2140
left_eye_center_y            2140
right_eye_center_x           2140
right_eye_center_y           2140
left_eye_inner_corner_x      2140
left_eye_inner_corner_y      2140
left_eye_outer_corner_x      2140
left_eye_outer_corner_y      2140
right_eye_inner_corner_x     2140
right_eye_inner_corner_y     2140
right_eye_outer_corner_x     2140
right_eye_outer_corner_y     2140
left_eyebrow_inner_end_x     2140
left_eyebrow_inner_end_y     2140
left_eyebrow_outer_end_x     2140
left_eyebrow_outer_end_y     2140
right_eyebrow_inner_end_x    2140
right_eyebrow_inner_end_y    2140
right_eyebrow_outer_end_x    2140
right_eyebrow_outer_end_y    2140
nose_tip_x                   2140
nose_tip_y                   2140
mouth_left_corner_x          2140
mouth_left_corner_y          2140
mouth_right_corner_x         2140
mouth_right_corner_y         2140
mouth_center_top_lip_x       2140
mouth_center_top_lip_y       2140
```

```
mouth_center_bottom_lip_x    2140
mouth_center_bottom_lip_y    2140
Image                        2140
dtype: int64
# Neural Network with 16561502 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1     32x94x94
  2  pool1     32x47x47
  3  dropout1  32x47x47
  4  conv2     64x46x46
  5  pool2     64x23x23
  6  dropout2  64x23x23
  7  conv3     128x22x22
  8  pool3     128x11x11
  9  dropout3  128x11x11
 10  hidden4   1000
 11  dropout4  1000
 12  hidden5   1000
 13  output    30

  epoch    trn loss    val loss    trn/val  dur
-------  ----------  ----------  ---------  -----
      1     0.06535     0.03422    1.90968  0.86s
      2     0.01401     0.02099    0.66744  0.85s
      3     0.00931     0.01543    0.60333  0.85s
      4     0.00796     0.01201    0.66257  0.85s
      5     0.00738     0.01112    0.66360  0.85s
      6     0.00708     0.01039    0.68167  0.85s
      7     0.00678     0.00997    0.67961  0.85s
      8     0.00661     0.00993    0.66618  0.85s
      9     0.00650     0.00953    0.68213  0.85s
     10     0.00622     0.00975    0.63812  0.85s
     11     0.00621     0.00963    0.64512  0.85s
     12     0.00612     0.00952    0.64289  0.85s
     13     0.00608     0.00887    0.68598  0.85s
     14     0.00595     0.00863    0.68968  0.87s
     15     0.00587     0.00869    0.67515  0.90s
     16     0.00584     0.00854    0.68399  0.92s
     17     0.00567     0.00860    0.65942  0.93s
     18     0.00562     0.00805    0.69822  0.93s
     19     0.00556     0.00840    0.66282  0.93s
     20     0.00551     0.00829    0.66484  0.93s
     21     0.00545     0.00824    0.66137  0.94s
     22     0.00544     0.00791    0.68792  0.93s
     23     0.00544     0.00793    0.68630  0.93s
     24     0.00536     0.00765    0.70033  0.93s
     25     0.00536     0.00776    0.69010  0.93s
     26     0.00528     0.00758    0.69636  0.93s
     27     0.00529     0.00775    0.68257  0.93s
     28     0.00521     0.00749    0.69539  0.93s
     29     0.00515     0.00741    0.69457  0.93s
     30     0.00514     0.00726    0.70734  0.93s
```

31	0.00515	0.00734	0.70124	0.93s
32	0.00505	0.00730	0.69117	0.93s
33	0.00508	0.00696	0.73051	0.93s
34	0.00506	0.00683	0.74023	0.93s
35	0.00502	0.00704	0.71289	0.93s
36	0.00505	0.00710	0.71149	0.93s
37	0.00498	0.00685	0.72610	0.93s
38	0.00496	0.00673	0.73739	0.93s
39	0.00495	0.00665	0.74523	0.93s
40	0.00495	0.00661	0.74950	0.93s
41	0.00494	0.00645	0.76625	0.93s
42	0.00489	0.00638	0.76635	0.93s
43	0.00487	0.00657	0.74168	0.93s
44	0.00488	0.00655	0.74416	0.93s
45	0.00485	0.00637	0.76183	0.93s
46	0.00480	0.00630	0.76270	0.93s
47	0.00484	0.00615	0.78720	0.94s
48	0.00482	0.00618	0.77925	0.94s
49	0.00483	0.00634	0.76157	0.94s
50	0.00475	0.00621	0.76510	0.93s
51	0.00478	0.00595	0.80368	0.94s
52	0.00473	0.00609	0.77771	0.93s
53	0.00471	0.00581	0.80975	0.93s
54	0.00474	0.00592	0.80007	0.93s
55	0.00473	0.00595	0.79492	0.93s
56	0.00473	0.00588	0.80459	0.93s
57	0.00472	0.00596	0.79165	0.93s
58	0.00468	0.00584	0.80241	0.93s
59	0.00467	0.00579	0.80673	0.93s
60	0.00466	0.00572	0.81615	0.93s
61	0.00465	0.00564	0.82451	0.93s
62	0.00465	0.00573	0.81247	0.93s
63	0.00465	0.00563	0.82612	0.93s
64	0.00462	0.00559	0.82635	0.93s
65	0.00462	0.00569	0.81161	0.93s
66	0.00462	0.00548	0.84399	0.93s
67	0.00462	0.00551	0.83860	0.93s
68	0.00458	0.00543	0.84374	0.93s
69	0.00457	0.00540	0.84632	0.93s
70	0.00458	0.00536	0.85304	0.93s
71	0.00455	0.00537	0.84666	0.93s
72	0.00456	0.00525	0.86959	0.93s
73	0.00454	0.00522	0.87020	0.93s
74	0.00452	0.00516	0.87556	0.93s
75	0.00454	0.00512	0.88786	0.93s
76	0.00454	0.00513	0.88543	0.93s
77	0.00452	0.00510	0.88586	0.93s
78	0.00451	0.00500	0.90214	0.93s
79	0.00453	0.00492	0.92212	0.94s
80	0.00451	0.00500	0.90325	0.93s
81	0.00448	0.00491	0.91369	0.93s
82	0.00449	0.00487	0.92148	0.93s
83	0.00450	0.00492	0.91469	0.93s
84	0.00448	0.00496	0.90327	0.93s
85	0.00443	0.00483	0.91715	0.93s
86	0.00449	0.00482	0.93009	0.93s
87	0.00448	0.00476	0.94186	0.93s

88	0.00446	0.00472	0.94648	0.93s
89	0.00444	0.00472	0.94039	0.93s
90	0.00445	0.00473	0.94092	0.93s
91	0.00444	0.00468	0.94803	0.93s
92	0.00444	0.00462	0.96107	0.93s
93	0.00443	0.00459	0.96420	0.93s
94	0.00443	0.00457	0.97102	0.93s
95	0.00442	0.00461	0.95768	0.93s
96	0.00441	0.00459	0.96186	0.93s
97	0.00439	0.00454	0.96740	0.93s
98	0.00443	0.00452	0.98085	0.93s
99	0.00440	0.00448	0.98235	0.93s
100	0.00440	0.00450	0.97814	0.93s
101	0.00438	0.00450	0.97437	0.93s
102	0.00437	0.00450	0.97242	0.93s
103	0.00438	0.00447	0.98001	0.93s
104	0.00438	0.00443	0.98848	0.93s
105	0.00438	0.00444	0.98645	0.93s
106	0.00436	0.00440	0.99105	0.93s
107	0.00437	0.00441	0.99072	0.93s
108	0.00436	0.00440	0.99181	0.93s
109	0.00434	0.00438	0.99225	0.93s
110	0.00435	0.00433	1.00389	0.93s
111	0.00435	0.00434	1.00113	0.93s
112	0.00436	0.00432	1.00785	0.93s
113	0.00434	0.00432	1.00339	0.93s
114	0.00433	0.00432	1.00276	0.93s
115	0.00434	0.00431	1.00716	0.93s
116	0.00434	0.00428	1.01278	0.93s
117	0.00431	0.00427	1.00965	0.93s
118	0.00431	0.00426	1.01135	0.93s
119	0.00430	0.00425	1.01191	0.93s
120	0.00432	0.00423	1.01938	0.93s
121	0.00431	0.00421	1.02354	0.93s
122	0.00430	0.00423	1.01750	0.93s
123	0.00429	0.00421	1.02036	0.93s
124	0.00429	0.00421	1.02036	0.93s
125	0.00428	0.00419	1.02155	0.93s
126	0.00429	0.00418	1.02642	0.93s
127	0.00428	0.00416	1.02794	0.93s
128	0.00427	0.00415	1.02856	0.93s
129	0.00427	0.00415	1.02964	0.93s
130	0.00426	0.00415	1.02464	0.93s
131	0.00425	0.00414	1.02721	0.93s
132	0.00424	0.00411	1.03259	0.93s
133	0.00427	0.00412	1.03422	0.93s
134	0.00424	0.00411	1.03082	0.94s
135	0.00424	0.00411	1.03145	0.93s
136	0.00425	0.00410	1.03736	0.94s
137	0.00425	0.00407	1.04239	0.94s
138	0.00425	0.00409	1.03986	0.93s
139	0.00424	0.00409	1.03731	0.93s
140	0.00422	0.00408	1.03473	0.93s
141	0.00423	0.00407	1.03763	0.93s
142	0.00420	0.00407	1.03267	0.93s
143	0.00422	0.00405	1.04095	0.93s
144	0.00422	0.00406	1.03888	0.93s

145	0.00420	0.00405	1.03689	0.94s
146	0.00421	0.00405	1.04072	0.93s
147	0.00419	0.00402	1.04208	0.93s
148	0.00418	0.00403	1.03675	0.94s
149	0.00419	0.00402	1.04214	0.93s
150	0.00418	0.00402	1.04128	0.93s
151	0.00417	0.00401	1.04090	0.93s
152	0.00416	0.00401	1.03493	0.93s
153	0.00416	0.00400	1.04084	0.93s
154	0.00417	0.00400	1.04146	0.93s
155	0.00418	0.00398	1.04983	0.93s
156	0.00416	0.00398	1.04558	0.93s
157	0.00416	0.00398	1.04687	0.93s
158	0.00416	0.00398	1.04590	0.93s
159	0.00413	0.00398	1.03824	0.93s
160	0.00415	0.00398	1.04219	0.93s
161	0.00414	0.00396	1.04468	0.93s
162	0.00412	0.00396	1.04185	0.93s
163	0.00412	0.00395	1.04485	0.93s
164	0.00413	0.00395	1.04611	0.93s
165	0.00411	0.00395	1.04189	0.93s
166	0.00411	0.00394	1.04446	0.93s
167	0.00412	0.00393	1.04707	0.93s
168	0.00410	0.00393	1.04427	0.93s
169	0.00409	0.00392	1.04145	0.93s
170	0.00409	0.00392	1.04441	0.93s
171	0.00410	0.00391	1.04817	0.93s
172	0.00409	0.00390	1.04697	0.94s
173	0.00409	0.00391	1.04454	0.93s
174	0.00409	0.00390	1.04638	0.93s
175	0.00408	0.00390	1.04569	0.93s
176	0.00409	0.00388	1.05337	0.93s
177	0.00405	0.00390	1.03844	0.93s
178	0.00407	0.00388	1.04664	0.93s
179	0.00405	0.00387	1.04731	0.93s
180	0.00405	0.00387	1.04666	0.93s
181	0.00406	0.00386	1.05143	0.93s
182	0.00403	0.00387	1.04349	0.93s
183	0.00405	0.00385	1.05260	0.93s
184	0.00404	0.00386	1.04884	0.93s
185	0.00403	0.00386	1.04546	0.93s
186	0.00404	0.00384	1.05283	0.93s
187	0.00402	0.00383	1.04910	0.93s
188	0.00401	0.00383	1.04585	0.93s
189	0.00401	0.00382	1.04783	0.93s
190	0.00401	0.00382	1.05220	0.93s
191	0.00398	0.00382	1.04354	0.93s
192	0.00399	0.00381	1.04813	0.93s
193	0.00399	0.00382	1.04387	0.93s
194	0.00398	0.00380	1.04790	0.93s
195	0.00398	0.00379	1.05021	0.93s
196	0.00397	0.00379	1.04699	0.93s
197	0.00397	0.00378	1.05046	0.93s
198	0.00397	0.00377	1.05249	0.93s
199	0.00395	0.00377	1.04771	0.93s
200	0.00395	0.00377	1.04958	0.93s
201	0.00395	0.00376	1.04935	0.93s

202	0.00396	0.00376	1.05341	0.93s
203	0.00394	0.00375	1.04924	0.94s
204	0.00392	0.00374	1.04691	0.93s
205	0.00394	0.00374	1.05452	0.94s
206	0.00391	0.00374	1.04500	0.94s
207	0.00392	0.00373	1.05205	0.93s
208	0.00392	0.00372	1.05467	0.93s
209	0.00391	0.00370	1.05677	0.94s
210	0.00392	0.00371	1.05759	0.93s
211	0.00391	0.00371	1.05405	0.93s
212	0.00390	0.00370	1.05584	0.93s
213	0.00388	0.00368	1.05348	0.93s
214	0.00390	0.00368	1.05875	0.93s
215	0.00390	0.00368	1.05872	0.93s
216	0.00388	0.00368	1.05608	0.93s
217	0.00389	0.00368	1.05644	0.93s
218	0.00387	0.00367	1.05287	0.93s
219	0.00386	0.00367	1.04953	0.93s
220	0.00385	0.00364	1.05755	0.93s
221	0.00385	0.00364	1.05708	0.93s
222	0.00383	0.00363	1.05554	0.93s
223	0.00385	0.00364	1.05559	0.93s
224	0.00382	0.00363	1.05222	0.93s
225	0.00383	0.00363	1.05561	0.93s
226	0.00381	0.00361	1.05613	0.94s
227	0.00381	0.00361	1.05569	0.93s
228	0.00382	0.00359	1.06255	0.93s
229	0.00380	0.00360	1.05698	0.93s
230	0.00379	0.00360	1.05166	0.93s
231	0.00380	0.00359	1.05864	0.93s
232	0.00378	0.00358	1.05521	0.93s
233	0.00378	0.00357	1.05810	0.94s
234	0.00377	0.00356	1.05809	0.93s
235	0.00376	0.00355	1.05767	0.93s
236	0.00375	0.00355	1.05635	0.93s
237	0.00375	0.00353	1.06154	0.93s
238	0.00374	0.00355	1.05216	0.93s
239	0.00373	0.00354	1.05500	0.93s
240	0.00374	0.00353	1.05851	0.93s
241	0.00373	0.00353	1.05475	0.93s
242	0.00372	0.00352	1.05925	0.93s
243	0.00371	0.00352	1.05248	0.93s
244	0.00370	0.00351	1.05143	0.94s
245	0.00370	0.00349	1.06148	0.93s
246	0.00370	0.00349	1.06085	0.94s
247	0.00369	0.00347	1.06210	0.93s
248	0.00368	0.00347	1.06075	0.94s
249	0.00366	0.00346	1.05663	0.93s
250	0.00366	0.00346	1.05708	0.93s
251	0.00366	0.00345	1.06028	0.93s
252	0.00367	0.00345	1.06511	0.93s
253	0.00365	0.00344	1.05913	0.93s
254	0.00363	0.00344	1.05768	0.93s
255	0.00362	0.00342	1.05787	0.93s
256	0.00363	0.00342	1.05937	0.93s
257	0.00363	0.00342	1.06057	0.93s
258	0.00361	0.00341	1.06067	0.93s

259	0.00361	0.00340	1.06213	0.93s
260	0.00359	0.00339	1.06147	0.94s
261	0.00360	0.00338	1.06466	0.93s
262	0.00358	0.00337	1.06225	0.93s
263	0.00358	0.00336	1.06544	0.93s
264	0.00358	0.00335	1.06854	0.93s
265	0.00357	0.00335	1.06607	0.93s
266	0.00357	0.00334	1.06962	0.93s
267	0.00356	0.00334	1.06522	0.93s
268	0.00356	0.00331	1.07479	0.93s
269	0.00354	0.00330	1.07245	0.94s
270	0.00354	0.00331	1.06893	0.93s
271	0.00352	0.00331	1.06237	0.93s
272	0.00352	0.00333	1.05803	0.93s
273	0.00351	0.00329	1.06715	0.93s
274	0.00351	0.00330	1.06411	0.93s
275	0.00352	0.00327	1.07436	0.93s
276	0.00348	0.00327	1.06480	0.94s
277	0.00348	0.00324	1.07532	0.93s
278	0.00348	0.00325	1.07081	0.93s
279	0.00346	0.00322	1.07441	0.93s
280	0.00346	0.00323	1.07064	0.93s
281	0.00347	0.00322	1.08008	0.93s
282	0.00345	0.00322	1.07224	0.94s
283	0.00344	0.00320	1.07691	0.93s
284	0.00344	0.00320	1.07551	0.93s
285	0.00344	0.00320	1.07705	0.93s
286	0.00343	0.00319	1.07627	0.93s
287	0.00342	0.00319	1.07323	0.93s
288	0.00342	0.00318	1.07557	0.94s
289	0.00341	0.00316	1.07714	0.93s
290	0.00339	0.00315	1.07718	0.93s
291	0.00338	0.00314	1.07513	0.93s
292	0.00338	0.00313	1.07917	0.94s
293	0.00338	0.00313	1.08056	0.93s
294	0.00337	0.00312	1.08213	0.93s
295	0.00335	0.00312	1.07621	0.93s
296	0.00335	0.00312	1.07467	0.94s
297	0.00334	0.00310	1.07732	0.94s
298	0.00332	0.00310	1.07036	0.93s
299	0.00335	0.00308	1.08678	0.94s
300	0.00332	0.00310	1.07177	0.93s
301	0.00333	0.00308	1.08012	0.93s
302	0.00330	0.00308	1.07013	0.93s
303	0.00331	0.00307	1.07976	0.94s
304	0.00331	0.00305	1.08483	0.93s
305	0.00331	0.00303	1.09139	0.93s
306	0.00329	0.00303	1.08630	0.94s
307	0.00327	0.00303	1.07782	0.93s
308	0.00326	0.00300	1.08664	0.93s
309	0.00326	0.00301	1.08174	0.94s
310	0.00325	0.00299	1.08604	0.94s
311	0.00326	0.00299	1.09078	0.94s
312	0.00324	0.00299	1.08091	0.93s
313	0.00324	0.00298	1.08997	0.93s
314	0.00325	0.00296	1.09562	0.93s
315	0.00322	0.00295	1.09097	0.93s

```
    316     0.00323     0.00297     1.08698  0.93s
    317     0.00320     0.00297     1.07742  0.93s
    318     0.00320     0.00298     1.07520  0.93s
    319     0.00319     0.00296     1.07712  0.93s
    320     0.00320     0.00292     1.09323  0.94s
    321     0.00319     0.00293     1.08816  0.94s
    322     0.00317     0.00291     1.09032  0.93s
    323     0.00315     0.00290     1.08603  0.93s
    324     0.00316     0.00290     1.09006  0.94s
    325     0.00315     0.00289     1.08850  0.94s
    326     0.00316     0.00288     1.09639  0.93s
    327     0.00313     0.00287     1.09077  0.93s
    328     0.00314     0.00287     1.09188  0.93s
    329     0.00313     0.00288     1.08722  0.94s
    330     0.00312     0.00286     1.08955  0.93s
    331     0.00312     0.00287     1.08854  0.93s
    332     0.00311     0.00284     1.09496  0.93s
    333     0.00311     0.00284     1.09392  0.94s
    334     0.00311     0.00283     1.09572  0.94s
    335     0.00308     0.00283     1.08962  0.93s
```

Net7 - RMSE Score

In [13]:
```
pickleFile = "/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetection_M
isc/net7_10000epochs.pickle"

net7 = pickle.load(open(pickleFile,'rb'))

train_loss = np.array([i["train_loss"] for i in net7.train_history_])
valid_loss = np.array([i["valid_loss"] for i in net7.train_history_])

print "Training Loss (Len,Value):",len(train_loss),train_loss[-1]
print "Validation Loss (Len,Value):",len(valid_loss),valid_loss[-1]

### RMSE score of baseline model after 10000 iterations
validation_loss =  0.000785359828276

np.sqrt(validation_loss)*48 # normalize to [-1,1]
```

```
Training Loss (Len,Value): 10000 0.000739023122049
Validation Loss (Len,Value): 10000 0.000785359828276
```

Out[13]:
```
1.3451650621198517
```

Kaggle Submission for net7

In [52]:
```
pickleFile = "/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetection_M
isc/net7_10000epochs.pickle"
submit_model(pickleFile)
```

```
ImageId    1783
Image      1783
dtype: int64
ImageId    1783
Image      1783
```

```
dtype: int64
(1783, 30)
       RowId  ImageId               FeatureName  Location
0          1        1         left_eye_center_x       NaN
1          2        1         left_eye_center_y       NaN
2          3        1        right_eye_center_x       NaN
3          4        1        right_eye_center_y       NaN
4          5        1   left_eye_inner_corner_x       NaN
5          6        1   left_eye_inner_corner_y       NaN
6          7        1   left_eye_outer_corner_x       NaN
7          8        1   left_eye_outer_corner_y       NaN
8          9        1  right_eye_inner_corner_x       NaN
9         10        1  right_eye_inner_corner_y       NaN
10        11        1  right_eye_outer_corner_x       NaN
11        12        1  right_eye_outer_corner_y       NaN
12        13        1  left_eyebrow_inner_end_x       NaN
13        14        1  left_eyebrow_inner_end_y       NaN
14        15        1  left_eyebrow_outer_end_x       NaN
15        16        1  left_eyebrow_outer_end_y       NaN
16        17        1 right_eyebrow_inner_end_x       NaN
17        18        1 right_eyebrow_inner_end_y       NaN
18        19        1 right_eyebrow_outer_end_x       NaN
19        20        1 right_eyebrow_outer_end_y       NaN
20        21        1               nose_tip_x       NaN
21        22        1               nose_tip_y       NaN
22        23        1       mouth_left_corner_x       NaN
23        24        1       mouth_left_corner_y       NaN
24        25        1      mouth_right_corner_x       NaN
25        26        1      mouth_right_corner_y       NaN
26        27        1    mouth_center_top_lip_x       NaN
27        28        1    mouth_center_top_lip_y       NaN
28        29        1 mouth_center_bottom_lip_x       NaN
29        30        1 mouth_center_bottom_lip_y       NaN
...      ...      ...                       ...       ...
27094  27095     1780        right_eye_center_x       NaN
27095  27096     1780        right_eye_center_y       NaN
27096  27097     1780               nose_tip_x       NaN
27097  27098     1780               nose_tip_y       NaN
27098  27099     1780 mouth_center_bottom_lip_x       NaN
27099  27100     1780 mouth_center_bottom_lip_y       NaN
27100  27101     1781         left_eye_center_x       NaN
27101  27102     1781         left_eye_center_y       NaN
27102  27103     1781        right_eye_center_x       NaN
27103  27104     1781        right_eye_center_y       NaN
27104  27105     1781               nose_tip_x       NaN
27105  27106     1781               nose_tip_y       NaN
27106  27107     1781 mouth_center_bottom_lip_x       NaN
27107  27108     1781 mouth_center_bottom_lip_y       NaN
27108  27109     1782         left_eye_center_x       NaN
27109  27110     1782         left_eye_center_y       NaN
27110  27111     1782        right_eye_center_x       NaN
27111  27112     1782        right_eye_center_y       NaN
27112  27113     1782               nose_tip_x       NaN
27113  27114     1782               nose_tip_y       NaN
27114  27115     1782 mouth_center_bottom_lip_x       NaN
27115  27116     1782 mouth_center_bottom_lip_y       NaN
27116  27117     1783         left_eye_center_x       NaN
```

```
27117  27118     1783          left_eye_center_y       NaN
27118  27119     1783         right_eye_center_x       NaN
27119  27120     1783         right_eye_center_y       NaN
27120  27121     1783                 nose_tip_x       NaN
27121  27122     1783                 nose_tip_y       NaN
27122  27123     1783  mouth_center_bottom_lip_x       NaN
27123  27124     1783  mouth_center_bottom_lip_y       NaN

[27124 rows x 4 columns]
Wrote submission-2017-04-23T12-43-28.076089.csv
```

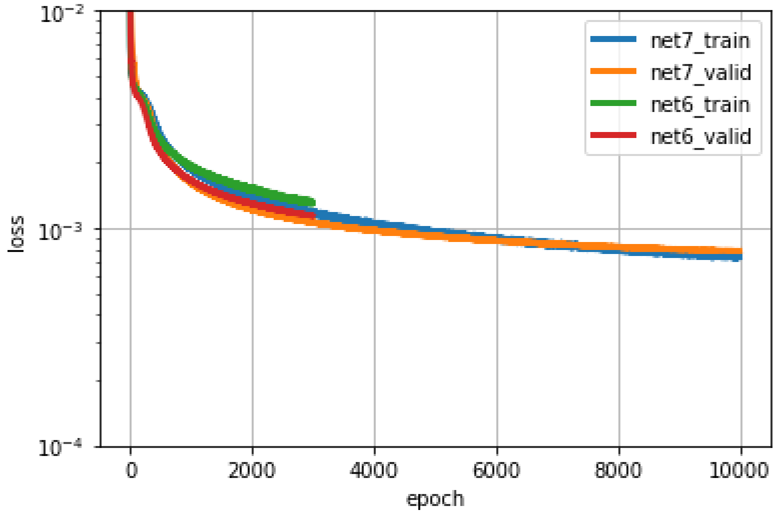
Evaluating performance of the baseline model relative to net7

Here, we evaluate the performance of the baseline model net6 against our new model net7 with the additional hidden layers and epochs.

In [7]:

```
# Load from pickle file after first run
net6_pickle_file = "/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetec
tion_Misc/net6.pickle"
net7_pickle_file = "/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetec
tion_Misc/net7_10000epochs.pickle"

net6 = pickle.load(open(net6_pickle_file,'rb'))
net7 = pickle.load(open(net7_pickle_file,'rb'))

# plot
net7_train_loss = np.array([i["train_loss"] for i in net7.train_history_])
net7_valid_loss = np.array([i["valid_loss"] for i in net7.train_history_])
net6_train_loss = np.array([i["train_loss"] for i in net6.train_history_])
net6_valid_loss = np.array([i["valid_loss"] for i in net6.train_history_])

print(net7_valid_loss)
print(net7_valid_loss.shape)

pyplot.plot(net7_train_loss, linewidth=3, label="net7_train")
pyplot.plot(net7_valid_loss, linewidth=3, label="net7_valid")
pyplot.plot(net6_train_loss, linewidth=3, label="net6_train")
pyplot.plot(net6_valid_loss, linewidth=3, label="net6_valid")

pyplot.grid()
pyplot.legend()
pyplot.xlabel("epoch")
pyplot.ylabel("loss")
pyplot.ylim(1e-4, 1e-2)
pyplot.yscale("log")

pyplot.show()
```

```
[ 0.04752024  0.03145442  0.02320333 ...,  0.00078588  0.00078561
  0.00078536]
(10000,)
```



Even with 10,000 epoch here, there still do not seem to be signs of overfitting. This is most likely due the result of the drop-out layer. And the val_loss drop quite a bit to 0.00087 which translate to an RMSE score of 1.9. The validation loss seems to bottom after 10000 epochs so we will try other methods to improve the model instead of further increases in the epoch size.

7. Train Keypoints Separately in Groups

Several columns in the training dataset contained NA values. When we loaded the training set to model net7, we dropped any rows that had at least one NA column. This elimination resulted in a loss of nearly 5000 rows in the 7000+ training samples.

In this section, we will attempt improvements to our model by including as many of the columns that do not have NA values as we can. We will create specialist models. Each specialist model will focus on trying to predict a subset of the facial keypoints. This way we can split the training set to ensure that we are maximizing usage of training samples.

Here, we split keypoints into 2 groups since we notice that once set of columns have 7000+ count whereas another set has 2000+ count.

Another optimization that we apply here is to initialize our weights with that of an existing saved neural network model, that of net7. This will allow our model to converge much faster. Note that in doing this, the weights for the output layer will not be loaded due to differing dimensions, although all other weights will be loaded.

The distribution of values in training data is shown below

- 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

The method fit_specialists takes as a parameter the pre-training data for initializing weights. It then fits multiple models based on the configuration of specialists settings (SPECIALIST_SETTINGS_NEW). Once the models are fitted, they are persisted in a pickle file.

In [7]:
```
## 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)),
        ),
    ]
```

In [8]:
```
## Modify the fit function to take each specialist one at a time and combin
e result at the end
from collections import OrderedDict

def fit_specialists(fname_pretrain=None):
    if fname_pretrain:
        with open(fname_pretrain, 'rb') as f:
            net_pretrain = pickle.load(f)
    else:
        net_pretrain = None

    specialists = OrderedDict()

    for setting in SPECIALIST_SETTINGS_NEW:
        cols = setting['columns']
        X, y = load2d(cols=cols)

        model = clone(net8)
        model.output_num_units = y.shape[1]
        model.batch_iterator_train.flip_indices = setting['flip_indices']
        model.max_epochs = int(1e7 / y.shape[0])
        if 'kwargs' in setting:
            # an option 'kwargs' in the settings list may be used to
            # set any other parameter of the net:
            vars(model).update(setting['kwargs'])

        if net_pretrain is not None:
            # if a pretrain model was given, use it to initialize the
            # weights of our new specialist model:
            model.load_params_from(net_pretrain)

        print("Training model for columns {} for {} epochs".format(
            cols, model.max_epochs))
        model.fit(X, y)
        specialists[cols] = model

    with open('net-specialists_no_early_stopping.pickle', 'wb') as f:
        # this time we're persisting a dictionary with all models:
        pickle.dump(specialists, f, -1)
    return specialists
```

In [ ]:
```
## Load net7.pickle trained above as pretrain to reduce epoch needed for ea
rly stop
import theano

from collections import OrderedDict
from sklearn.base import clone

net8 = NeuralNet(
```

```
    layers=[
        ('input', layers.InputLayer),
        ('conv1', layers.Conv2DLayer),
        ('pool1', layers.MaxPool2DLayer),
        ('dropout1', layers.DropoutLayer),
        ('conv2', layers.Conv2DLayer),
        ('pool2', layers.MaxPool2DLayer),
        ('dropout2', layers.DropoutLayer),
        ('conv3', layers.Conv2DLayer),
        ('pool3', layers.MaxPool2DLayer),
        ('dropout3', layers.DropoutLayer),
        ('hidden4', layers.DenseLayer),
        ('dropout4', layers.DropoutLayer),
        ('hidden5', layers.DenseLayer),
        ('output', layers.DenseLayer),
        ],
    input_shape=(None, 1, 96, 96),
    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=1000,
    dropout4_p=0.5,
    hidden5_num_units=1000,
    output_num_units=30, output_nonlinearity=None,

    update_learning_rate=theano.shared(float32(0.03)),
    update_momentum=theano.shared(float32(0.9)),

    regression=True,
    batch_iterator_train=FlipBatchIterator(batch_size=128),
    on_epoch_finished=[
        AdjustVariable('update_learning_rate', start=0.03, stop=0.0001),
        AdjustVariable('update_momentum', start=0.9, stop=0.999),
        #EarlyStopping(patience=200),
        ],
    max_epochs=5000,
    verbose=1,
    )

sys.setrecursionlimit(10000)

fname_pretrain = "/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetecti
on_Misc/net7_10000epochs.pickle"
fit_specialists(fname_pretrain)
```

```
left_eye_center_x           7039
left_eye_center_y           7039
right_eye_center_x          7036
right_eye_center_y          7036
nose_tip_x                  7049
nose_tip_y                  7049
```

```
mouth_center_bottom_lip_x    7016
mouth_center_bottom_lip_y    7016
Image                       7049
dtype: int64
left_eye_center_x            7000
left_eye_center_y            7000
right_eye_center_x           7000
right_eye_center_y           7000
nose_tip_x                  7000
nose_tip_y                  7000
mouth_center_bottom_lip_x    7000
mouth_center_bottom_lip_y    7000
Image                       7000
dtype: int64
Loaded parameters to layer 'conv1' (shape 32x1x3x3).
Loaded parameters to layer 'conv1' (shape 32).
Loaded parameters to layer 'conv2' (shape 64x32x2x2).
Loaded parameters to layer 'conv2' (shape 64).
Loaded parameters to layer 'conv3' (shape 128x64x2x2).
Loaded parameters to layer 'conv3' (shape 128).
Loaded parameters to layer 'hidden4' (shape 15488x1000).
Loaded parameters to layer 'hidden4' (shape 1000).
Loaded parameters to layer 'hidden5' (shape 1000x1000).
Loaded parameters to layer 'hidden5' (shape 1000).
Could not load parameters to layer 'output' because shapes did not match: 1
000x8 vs 1000x30.
Could not load parameters to layer 'output' because shapes did not match: 8
vs 30.
Training model for columns ('left_eye_center_x', 'left_eye_center_y', 'righ
t_eye_center_x', 'right_eye_center_y', 'nose_tip_x', 'nose_tip_y', 'mouth_c
enter_bottom_lip_x', 'mouth_center_bottom_lip_y') for 1428 epochs
# Neural Network with 16539480 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1     32x94x94
  2  pool1     32x47x47
  3  dropout1  32x47x47
  4  conv2     64x46x46
  5  pool2     64x23x23
  6  dropout2  64x23x23
  7  conv3     128x22x22
  8  pool3     128x11x11
  9  dropout3  128x11x11
 10  hidden4   1000
 11  dropout4  1000
 12  hidden5   1000
 13  output    8

# Neural Network with 16539480 learnable parameters

## Layer information

  #  name      size
```

```
---  --------  ---------
  0  input     1x96x96
  1  conv1     32x94x94
  2  pool1     32x47x47
  3  dropout1  32x47x47
  4  conv2     64x46x46
  5  pool2     64x23x23
  6  dropout2  64x23x23
  7  conv3     128x22x22
  8  pool3     128x11x11
  9  dropout3  128x11x11
 10  hidden4   1000
 11  dropout4  1000
 12  hidden5   1000
 13  output    8

  epoch    trn loss    val loss    trn/val   dur
-------  ----------  ----------  ---------  -----
      1     0.02135     0.00722    2.95627  2.74s
  epoch    trn loss    val loss    trn/val   dur
-------  ----------  ----------  ---------  -----
      1     0.02135     0.00722    2.95627  2.74s
      2     0.00682     0.00657    1.03663  2.71s
      2     0.00682     0.00657    1.03663  2.71s
      3     0.00633     0.00604    1.04832  2.71s
      3     0.00633     0.00604    1.04832  2.71s
      4     0.00593     0.00558    1.06321  2.73s
      4     0.00593     0.00558    1.06321  2.73s
      5     0.00561     0.00512    1.09610  2.73s
      5     0.00561     0.00512    1.09610  2.73s
      6     0.00528     0.00476    1.10976  2.74s
      6     0.00528     0.00476    1.10976  2.74s
      7     0.00504     0.00449    1.12201  2.73s
      7     0.00504     0.00449    1.12201  2.73s
      8     0.00483     0.00423    1.14253  2.73s
      8     0.00483     0.00423    1.14253  2.73s
      9     0.00465     0.00407    1.14295  2.74s
      9     0.00465     0.00407    1.14295  2.74s
     10     0.00452     0.00391    1.15548  2.73s
     10     0.00452     0.00391    1.15548  2.73s
     11     0.00441     0.00379    1.16418  2.73s
     11     0.00441     0.00379    1.16418  2.73s
     12     0.00432     0.00369    1.17053  2.74s
     12     0.00432     0.00369    1.17053  2.74s
     13     0.00426     0.00362    1.17636  2.74s
     13     0.00426     0.00362    1.17636  2.74s
     14     0.00419     0.00357    1.17460  2.74s
     14     0.00419     0.00357    1.17460  2.74s
     15     0.00411     0.00352    1.16785  2.77s
     15     0.00411     0.00352    1.16785  2.77s
     16     0.00406     0.00347    1.16875  2.76s
     16     0.00406     0.00347    1.16875  2.76s
     17     0.00402     0.00338    1.18772  2.75s
     17     0.00402     0.00338    1.18772  2.75s
     18     0.00398     0.00336    1.18383  2.74s
     18     0.00398     0.00336    1.18383  2.74s
     19     0.00396     0.00334    1.18479  2.76s
```

19	0.00396	0.00334	1.18479	2.76s
20	0.00388	0.00332	1.16946	2.76s
20	0.00388	0.00332	1.16946	2.76s
21	0.00385	0.00325	1.18203	2.75s
21	0.00385	0.00325	1.18203	2.75s
22	0.00386	0.00322	1.19818	2.87s
22	0.00386	0.00322	1.19818	2.87s
23	0.00382	0.00324	1.17875	2.95s
23	0.00382	0.00324	1.17875	2.95s
24	0.00380	0.00317	1.19631	2.97s
24	0.00380	0.00317	1.19631	2.97s
25	0.00382	0.00316	1.21007	2.97s
25	0.00382	0.00316	1.21007	2.97s
26	0.00373	0.00314	1.18882	2.97s
26	0.00373	0.00314	1.18882	2.97s
27	0.00374	0.00311	1.20321	2.97s
27	0.00374	0.00311	1.20321	2.97s
28	0.00369	0.00311	1.18717	2.97s
28	0.00369	0.00311	1.18717	2.97s
29	0.00368	0.00306	1.20156	2.98s
29	0.00368	0.00306	1.20156	2.98s
30	0.00366	0.00306	1.19542	2.97s
30	0.00366	0.00306	1.19542	2.97s
31	0.00362	0.00305	1.18902	2.97s
31	0.00362	0.00305	1.18902	2.97s
32	0.00364	0.00303	1.20338	2.98s
32	0.00364	0.00303	1.20338	2.98s
33	0.00363	0.00305	1.19262	2.97s
33	0.00363	0.00305	1.19262	2.97s
34	0.00360	0.00298	1.20725	2.97s
34	0.00360	0.00298	1.20725	2.97s
35	0.00361	0.00299	1.20958	2.97s
35	0.00361	0.00299	1.20958	2.97s
36	0.00359	0.00296	1.21105	2.97s
36	0.00359	0.00296	1.21105	2.97s
37	0.00356	0.00298	1.19152	2.98s
37	0.00356	0.00298	1.19152	2.98s
38	0.00352	0.00296	1.19048	2.98s
38	0.00352	0.00296	1.19048	2.98s
39	0.00352	0.00293	1.20018	2.96s
39	0.00352	0.00293	1.20018	2.96s
40	0.00355	0.00293	1.20890	2.97s
40	0.00355	0.00293	1.20890	2.97s
41	0.00348	0.00293	1.19028	2.98s
41	0.00348	0.00293	1.19028	2.98s
42	0.00350	0.00291	1.20467	3.00s
42	0.00350	0.00291	1.20467	3.00s
43	0.00350	0.00293	1.19398	3.00s
43	0.00350	0.00293	1.19398	3.00s
44	0.00350	0.00288	1.21425	2.99s
44	0.00350	0.00288	1.21425	2.99s
45	0.00345	0.00287	1.20497	3.00s
45	0.00345	0.00287	1.20497	3.00s
46	0.00344	0.00284	1.21232	3.01s
46	0.00344	0.00284	1.21232	3.01s
47	0.00345	0.00287	1.20379	3.00s
47	0.00345	0.00287	1.20379	3.00s

48	0.00343	0.00283	1.21325	3.01s
48	0.00343	0.00283	1.21325	3.01s
49	0.00340	0.00281	1.20818	3.01s
49	0.00340	0.00281	1.20818	3.01s
50	0.00339	0.00283	1.19998	3.01s
50	0.00339	0.00283	1.19998	3.01s
51	0.00341	0.00280	1.21727	3.01s
51	0.00341	0.00280	1.21727	3.01s
52	0.00334	0.00280	1.19438	3.04s
52	0.00334	0.00280	1.19438	3.04s
53	0.00337	0.00279	1.20737	3.02s
53	0.00337	0.00279	1.20737	3.02s
54	0.00339	0.00279	1.21682	3.03s
54	0.00339	0.00279	1.21682	3.03s
55	0.00336	0.00279	1.20638	3.02s
55	0.00336	0.00279	1.20638	3.02s
56	0.00336	0.00277	1.21480	3.03s
56	0.00336	0.00277	1.21480	3.03s
57	0.00334	0.00277	1.20524	3.02s
57	0.00334	0.00277	1.20524	3.02s
58	0.00334	0.00274	1.21630	3.02s
58	0.00334	0.00274	1.21630	3.02s
59	0.00330	0.00275	1.19943	3.02s
59	0.00330	0.00275	1.19943	3.02s
60	0.00328	0.00272	1.20561	3.02s
60	0.00328	0.00272	1.20561	3.02s
61	0.00333	0.00275	1.20878	3.03s
61	0.00333	0.00275	1.20878	3.03s
62	0.00332	0.00273	1.21439	3.02s
62	0.00332	0.00273	1.21439	3.02s
63	0.00328	0.00271	1.20973	3.02s
63	0.00328	0.00271	1.20973	3.02s
64	0.00327	0.00272	1.20193	3.01s
64	0.00327	0.00272	1.20193	3.01s
65	0.00327	0.00274	1.19592	3.02s
65	0.00327	0.00274	1.19592	3.02s
66	0.00324	0.00270	1.19813	3.02s
66	0.00324	0.00270	1.19813	3.02s
67	0.00329	0.00270	1.22114	3.03s
67	0.00329	0.00270	1.22114	3.03s
68	0.00324	0.00268	1.20911	3.02s
68	0.00324	0.00268	1.20911	3.02s
69	0.00322	0.00270	1.19145	3.02s
69	0.00322	0.00270	1.19145	3.02s
70	0.00326	0.00268	1.21316	3.01s
70	0.00326	0.00268	1.21316	3.01s
71	0.00325	0.00270	1.20188	3.02s
71	0.00325	0.00270	1.20188	3.02s
72	0.00322	0.00268	1.20001	3.03s
72	0.00322	0.00268	1.20001	3.03s
73	0.00322	0.00266	1.20938	3.02s
73	0.00322	0.00266	1.20938	3.02s
74	0.00320	0.00266	1.20593	3.02s
74	0.00320	0.00266	1.20593	3.02s
75	0.00322	0.00268	1.20118	3.01s
75	0.00322	0.00268	1.20118	3.01s
76	0.00318	0.00268	1.18731	3.04s

76	0.00318	0.00268	1.18731	3.04s
77	0.00320	0.00264	1.21169	3.02s
77	0.00320	0.00264	1.21169	3.02s
78	0.00318	0.00266	1.19841	3.02s
78	0.00318	0.00266	1.19841	3.02s
79	0.00320	0.00264	1.21045	3.02s
79	0.00320	0.00264	1.21045	3.02s
80	0.00316	0.00265	1.19351	3.02s
80	0.00316	0.00265	1.19351	3.02s
81	0.00315	0.00267	1.18023	3.02s
81	0.00315	0.00267	1.18023	3.02s
82	0.00319	0.00264	1.21020	3.02s
82	0.00319	0.00264	1.21020	3.02s
83	0.00314	0.00257	1.22375	3.02s
83	0.00314	0.00257	1.22375	3.02s
84	0.00314	0.00259	1.21506	3.02s
84	0.00314	0.00259	1.21506	3.02s
85	0.00317	0.00260	1.21512	3.02s
85	0.00317	0.00260	1.21512	3.02s
86	0.00313	0.00259	1.20949	3.02s
86	0.00313	0.00259	1.20949	3.02s
87	0.00311	0.00261	1.18764	3.03s
87	0.00311	0.00261	1.18764	3.03s
88	0.00314	0.00256	1.22403	3.02s
88	0.00314	0.00256	1.22403	3.02s
89	0.00314	0.00256	1.22704	3.02s
89	0.00314	0.00256	1.22704	3.02s
90	0.00312	0.00254	1.23117	3.03s
90	0.00312	0.00254	1.23117	3.03s
91	0.00308	0.00255	1.20987	3.02s
91	0.00308	0.00255	1.20987	3.02s
92	0.00308	0.00257	1.19953	3.02s
92	0.00308	0.00257	1.19953	3.02s
93	0.00308	0.00260	1.18325	3.02s
93	0.00308	0.00260	1.18325	3.02s
94	0.00309	0.00254	1.21440	3.01s
94	0.00309	0.00254	1.21440	3.01s
95	0.00313	0.00253	1.23760	3.03s
95	0.00313	0.00253	1.23760	3.03s
96	0.00306	0.00255	1.19936	3.03s
96	0.00306	0.00255	1.19936	3.03s
97	0.00308	0.00253	1.21606	3.03s
97	0.00308	0.00253	1.21606	3.03s
98	0.00305	0.00252	1.21146	3.03s
98	0.00305	0.00252	1.21146	3.03s
99	0.00307	0.00254	1.20786	3.03s
99	0.00307	0.00254	1.20786	3.03s
100	0.00304	0.00253	1.20371	3.02s
100	0.00304	0.00253	1.20371	3.02s
101	0.00305	0.00252	1.20796	3.02s
101	0.00305	0.00252	1.20796	3.02s
102	0.00304	0.00253	1.20313	3.02s
102	0.00304	0.00253	1.20313	3.02s
103	0.00303	0.00252	1.20228	3.02s
103	0.00303	0.00252	1.20228	3.02s
104	0.00303	0.00251	1.20836	3.01s
104	0.00303	0.00251	1.20836	3.01s

105	0.00303	0.00250	1.20998	3.02s
105	0.00303	0.00250	1.20998	3.02s
106	0.00304	0.00251	1.20822	3.02s
106	0.00304	0.00251	1.20822	3.02s
107	0.00301	0.00251	1.20122	3.03s
107	0.00301	0.00251	1.20122	3.03s
108	0.00304	0.00248	1.22619	3.02s
108	0.00304	0.00248	1.22619	3.02s
109	0.00303	0.00252	1.20603	3.02s
109	0.00303	0.00252	1.20603	3.02s
110	0.00300	0.00249	1.20566	3.02s
110	0.00300	0.00249	1.20566	3.02s
111	0.00300	0.00248	1.20990	3.02s
111	0.00300	0.00248	1.20990	3.02s
112	0.00300	0.00248	1.21180	3.02s
112	0.00300	0.00248	1.21180	3.02s
113	0.00298	0.00243	1.22443	3.01s
113	0.00298	0.00243	1.22443	3.01s
114	0.00300	0.00249	1.20130	3.01s
114	0.00300	0.00249	1.20130	3.01s
115	0.00298	0.00245	1.21379	3.02s
115	0.00298	0.00245	1.21379	3.02s
116	0.00302	0.00248	1.22194	3.02s
116	0.00302	0.00248	1.22194	3.02s
117	0.00297	0.00246	1.20885	3.02s
117	0.00297	0.00246	1.20885	3.02s
118	0.00296	0.00250	1.18584	3.02s
118	0.00296	0.00250	1.18584	3.02s
119	0.00294	0.00244	1.20547	3.02s
119	0.00294	0.00244	1.20547	3.02s
120	0.00301	0.00246	1.22348	3.02s
120	0.00301	0.00246	1.22348	3.02s
121	0.00294	0.00248	1.18751	3.03s
121	0.00294	0.00248	1.18751	3.03s
122	0.00292	0.00244	1.19664	3.02s
122	0.00292	0.00244	1.19664	3.02s
123	0.00296	0.00242	1.22256	3.02s
123	0.00296	0.00242	1.22256	3.02s
124	0.00293	0.00243	1.20466	3.02s
124	0.00293	0.00243	1.20466	3.02s
125	0.00294	0.00245	1.19950	3.02s
125	0.00294	0.00245	1.19950	3.02s
126	0.00290	0.00242	1.19817	3.03s
126	0.00290	0.00242	1.19817	3.03s
127	0.00293	0.00240	1.21928	3.02s
127	0.00293	0.00240	1.21928	3.02s
128	0.00292	0.00245	1.18961	3.03s
128	0.00292	0.00245	1.18961	3.03s
129	0.00289	0.00241	1.20066	3.02s
129	0.00289	0.00241	1.20066	3.02s
130	0.00292	0.00241	1.21370	3.02s
130	0.00292	0.00241	1.21370	3.02s
131	0.00292	0.00239	1.22226	3.02s
131	0.00292	0.00239	1.22226	3.02s
132	0.00293	0.00240	1.21986	3.02s
132	0.00293	0.00240	1.21986	3.02s
133	0.00291	0.00242	1.20425	3.02s

133	0.00291	0.00242	1.20425	3.02s
134	0.00290	0.00239	1.21463	3.02s
134	0.00290	0.00239	1.21463	3.02s
135	0.00291	0.00243	1.19664	3.02s
135	0.00291	0.00243	1.19664	3.02s
136	0.00289	0.00239	1.20927	3.02s
136	0.00289	0.00239	1.20927	3.02s
137	0.00285	0.00243	1.17303	3.02s
137	0.00285	0.00243	1.17303	3.02s
138	0.00288	0.00235	1.22577	3.02s
138	0.00288	0.00235	1.22577	3.02s
139	0.00289	0.00239	1.20783	3.03s
139	0.00289	0.00239	1.20783	3.03s
140	0.00288	0.00243	1.18475	3.02s
140	0.00288	0.00243	1.18475	3.02s
141	0.00289	0.00239	1.20987	3.03s
141	0.00289	0.00239	1.20987	3.03s
142	0.00284	0.00237	1.20171	3.03s
142	0.00284	0.00237	1.20171	3.03s
143	0.00289	0.00238	1.21096	3.02s
143	0.00289	0.00238	1.21096	3.02s
144	0.00289	0.00236	1.22374	3.02s
144	0.00289	0.00236	1.22374	3.02s
145	0.00287	0.00237	1.20907	3.02s
145	0.00287	0.00237	1.20907	3.02s
146	0.00284	0.00236	1.20679	3.02s
146	0.00284	0.00236	1.20679	3.02s
147	0.00285	0.00236	1.20741	3.03s
147	0.00285	0.00236	1.20741	3.03s
148	0.00287	0.00235	1.22127	3.02s
148	0.00287	0.00235	1.22127	3.02s
149	0.00285	0.00234	1.22143	3.02s
149	0.00285	0.00234	1.22143	3.02s
150	0.00283	0.00234	1.20623	3.02s
150	0.00283	0.00234	1.20623	3.02s
151	0.00284	0.00235	1.20457	3.03s
151	0.00284	0.00235	1.20457	3.03s
152	0.00278	0.00236	1.17565	3.03s
152	0.00278	0.00236	1.17565	3.03s
153	0.00282	0.00236	1.19522	3.02s
153	0.00282	0.00236	1.19522	3.02s
154	0.00283	0.00233	1.21305	3.03s
154	0.00283	0.00233	1.21305	3.03s
155	0.00282	0.00232	1.21201	3.02s
155	0.00282	0.00232	1.21201	3.02s
156	0.00280	0.00234	1.19772	3.02s
156	0.00280	0.00234	1.19772	3.02s
157	0.00281	0.00232	1.20993	3.03s
157	0.00281	0.00232	1.20993	3.03s
158	0.00281	0.00231	1.21573	3.02s
158	0.00281	0.00231	1.21573	3.02s
159	0.00280	0.00233	1.20354	3.02s
159	0.00280	0.00233	1.20354	3.02s
160	0.00281	0.00232	1.21348	3.02s
160	0.00281	0.00232	1.21348	3.02s
161	0.00277	0.00232	1.19328	3.03s
161	0.00277	0.00232	1.19328	3.03s

162	0.00281	0.00232	1.21374	3.02s
162	0.00281	0.00232	1.21374	3.02s
163	0.00282	0.00235	1.19971	3.02s
163	0.00282	0.00235	1.19971	3.02s
164	0.00283	0.00233	1.21702	3.02s
164	0.00283	0.00233	1.21702	3.02s
165	0.00278	0.00229	1.21044	3.03s
165	0.00278	0.00229	1.21044	3.03s
166	0.00275	0.00231	1.19242	3.02s
166	0.00275	0.00231	1.19242	3.02s
167	0.00278	0.00232	1.19913	3.02s
167	0.00278	0.00232	1.19913	3.02s
168	0.00276	0.00229	1.20462	3.02s
168	0.00276	0.00229	1.20462	3.02s
169	0.00279	0.00232	1.20197	3.02s
169	0.00279	0.00232	1.20197	3.02s
170	0.00276	0.00228	1.21070	3.02s
170	0.00276	0.00228	1.21070	3.02s
171	0.00274	0.00228	1.20397	3.02s
171	0.00274	0.00228	1.20397	3.02s
172	0.00278	0.00227	1.22548	3.03s
172	0.00278	0.00227	1.22548	3.03s
173	0.00275	0.00230	1.19555	3.02s
173	0.00275	0.00230	1.19555	3.02s
174	0.00277	0.00230	1.20787	3.02s
174	0.00277	0.00230	1.20787	3.02s
175	0.00279	0.00233	1.19742	3.02s
175	0.00279	0.00233	1.19742	3.02s
176	0.00276	0.00228	1.21233	3.02s
176	0.00276	0.00228	1.21233	3.02s
177	0.00273	0.00232	1.18042	3.03s
177	0.00273	0.00232	1.18042	3.03s
178	0.00272	0.00228	1.19293	3.02s
178	0.00272	0.00228	1.19293	3.02s
179	0.00276	0.00230	1.19865	3.03s
179	0.00276	0.00230	1.19865	3.03s
180	0.00273	0.00226	1.20624	3.03s
180	0.00273	0.00226	1.20624	3.03s
181	0.00271	0.00227	1.19380	3.03s
181	0.00271	0.00227	1.19380	3.03s
182	0.00273	0.00230	1.18805	3.02s
182	0.00273	0.00230	1.18805	3.02s
183	0.00273	0.00225	1.21344	3.03s
183	0.00273	0.00225	1.21344	3.03s
184	0.00273	0.00230	1.18394	3.02s
184	0.00273	0.00230	1.18394	3.02s
185	0.00276	0.00226	1.22144	3.03s
185	0.00276	0.00226	1.22144	3.03s
186	0.00273	0.00225	1.21220	3.03s
186	0.00273	0.00225	1.21220	3.03s
187	0.00271	0.00229	1.18234	3.03s
187	0.00271	0.00229	1.18234	3.03s
188	0.00271	0.00226	1.19820	3.03s
188	0.00271	0.00226	1.19820	3.03s
189	0.00272	0.00224	1.21200	3.03s
189	0.00272	0.00224	1.21200	3.03s
190	0.00272	0.00226	1.20556	3.03s

190	0.00272	0.00226	1.20556	3.03s
191	0.00269	0.00225	1.19515	3.03s
191	0.00269	0.00225	1.19515	3.03s
192	0.00269	0.00226	1.18817	3.02s
192	0.00269	0.00226	1.18817	3.02s
193	0.00271	0.00224	1.20940	3.02s
193	0.00271	0.00224	1.20940	3.02s
194	0.00270	0.00228	1.18357	3.03s
194	0.00270	0.00228	1.18357	3.03s
195	0.00268	0.00225	1.19393	3.04s
195	0.00268	0.00225	1.19393	3.04s
196	0.00272	0.00227	1.19556	3.03s
196	0.00272	0.00227	1.19556	3.03s
197	0.00270	0.00225	1.19903	3.03s
197	0.00270	0.00225	1.19903	3.03s
198	0.00273	0.00222	1.23183	3.03s
198	0.00273	0.00222	1.23183	3.03s
199	0.00270	0.00223	1.21130	3.03s
199	0.00270	0.00223	1.21130	3.03s
200	0.00265	0.00223	1.19133	3.03s
200	0.00265	0.00223	1.19133	3.03s
201	0.00268	0.00225	1.19264	3.03s
201	0.00268	0.00225	1.19264	3.03s
202	0.00266	0.00223	1.19659	3.03s
202	0.00266	0.00223	1.19659	3.03s
203	0.00265	0.00223	1.19127	3.02s
203	0.00265	0.00223	1.19127	3.02s
204	0.00265	0.00224	1.18528	3.03s
204	0.00265	0.00224	1.18528	3.03s
205	0.00263	0.00222	1.18469	3.03s
205	0.00263	0.00222	1.18469	3.03s
206	0.00266	0.00223	1.19604	3.03s
206	0.00266	0.00223	1.19604	3.03s
207	0.00264	0.00223	1.18305	3.03s
207	0.00264	0.00223	1.18305	3.03s
208	0.00265	0.00223	1.19076	3.03s
208	0.00265	0.00223	1.19076	3.03s
209	0.00263	0.00222	1.18403	3.02s
209	0.00263	0.00222	1.18403	3.02s
210	0.00263	0.00221	1.19241	3.02s
210	0.00263	0.00221	1.19241	3.02s
211	0.00264	0.00223	1.18555	3.03s
211	0.00264	0.00223	1.18555	3.03s
212	0.00261	0.00222	1.17596	3.03s
212	0.00261	0.00222	1.17596	3.03s
213	0.00264	0.00219	1.20550	3.02s
213	0.00264	0.00219	1.20550	3.02s
214	0.00261	0.00222	1.17694	3.03s
214	0.00261	0.00222	1.17694	3.03s
215	0.00266	0.00222	1.19944	3.03s
215	0.00266	0.00222	1.19944	3.03s
216	0.00262	0.00216	1.21138	3.03s
216	0.00262	0.00216	1.21138	3.03s
217	0.00263	0.00221	1.19085	3.02s
217	0.00263	0.00221	1.19085	3.02s
218	0.00262	0.00221	1.18662	3.03s
218	0.00262	0.00221	1.18662	3.03s

```
219     0.00260     0.00219     1.18770  3.03s
219     0.00260     0.00219     1.18770  3.03s
220     0.00260     0.00221     1.17499  3.03s
220     0.00260     0.00221     1.17499  3.03s
221     0.00263     0.00220     1.19652  3.02s
221     0.00263     0.00220     1.19652  3.02s
222     0.00261     0.00222     1.17553  3.03s
222     0.00261     0.00222     1.17553  3.03s
223     0.00261     0.00221     1.18481  3.02s
223     0.00261     0.00221     1.18481  3.02s
224     0.00259     0.00220     1.18036  3.03s
224     0.00259     0.00220     1.18036  3.03s
225     0.00260     0.00217     1.19574  3.02s
225     0.00260     0.00217     1.19574  3.02s
226     0.00263     0.00218     1.20699  3.03s
226     0.00263     0.00218     1.20699  3.03s
227     0.00264     0.00222     1.18512  3.02s
227     0.00264     0.00222     1.18512  3.02s
228     0.00262     0.00219     1.19482  3.03s
228     0.00262     0.00219     1.19482  3.03s
229     0.00260     0.00222     1.16946  3.02s
229     0.00260     0.00222     1.16946  3.02s
230     0.00263     0.00220     1.19265  3.03s
230     0.00263     0.00220     1.19265  3.03s
231     0.00257     0.00218     1.18089  3.03s
231     0.00257     0.00218     1.18089  3.03s
232     0.00262     0.00217     1.20607  3.02s
232     0.00262     0.00217     1.20607  3.02s
233     0.00258     0.00218     1.18272  3.01s
233     0.00258     0.00218     1.18272  3.01s
234     0.00258     0.00220     1.17641  3.02s
234     0.00258     0.00220     1.17641  3.02s
235     0.00260     0.00219     1.18679  3.03s
235     0.00260     0.00219     1.18679  3.03s
236     0.00258     0.00217     1.18688  3.03s
236     0.00258     0.00217     1.18688  3.03s
237     0.00259     0.00220     1.17715  3.01s
237     0.00259     0.00220     1.17715  3.01s
238     0.00257     0.00220     1.16714  3.03s
238     0.00257     0.00220     1.16714  3.03s
239     0.00260     0.00217     1.19860  3.03s
239     0.00260     0.00217     1.19860  3.03s
240     0.00256     0.00216     1.18198  3.03s
240     0.00256     0.00216     1.18198  3.03s
241     0.00258     0.00218     1.18267  3.02s
241     0.00258     0.00218     1.18267  3.02s
242     0.00254     0.00219     1.16363  3.03s
242     0.00254     0.00219     1.16363  3.03s
243     0.00257     0.00218     1.18061  3.02s
243     0.00257     0.00218     1.18061  3.02s
244     0.00257     0.00215     1.19578  3.02s
244     0.00257     0.00215     1.19578  3.02s
245     0.00257     0.00218     1.17945  3.02s
245     0.00257     0.00218     1.17945  3.02s
246     0.00255     0.00213     1.19390  3.02s
246     0.00255     0.00213     1.19390  3.02s
247     0.00257     0.00216     1.19384  3.02s
```

247	0.00257	0.00216	1.19384	3.02s
248	0.00258	0.00217	1.18969	3.02s
248	0.00258	0.00217	1.18969	3.02s
249	0.00254	0.00219	1.16336	3.02s
249	0.00254	0.00219	1.16336	3.02s
250	0.00257	0.00217	1.18670	3.02s
250	0.00257	0.00217	1.18670	3.02s
251	0.00255	0.00214	1.19189	3.03s
251	0.00255	0.00214	1.19189	3.03s
252	0.00255	0.00219	1.16565	3.02s
252	0.00255	0.00219	1.16565	3.02s
253	0.00254	0.00218	1.16609	3.02s
253	0.00254	0.00218	1.16609	3.02s
254	0.00256	0.00215	1.18850	3.02s
254	0.00256	0.00215	1.18850	3.02s
255	0.00255	0.00215	1.18863	3.03s
255	0.00255	0.00215	1.18863	3.03s
256	0.00250	0.00216	1.15945	3.02s
256	0.00250	0.00216	1.15945	3.02s
257	0.00254	0.00217	1.17099	3.03s
257	0.00254	0.00217	1.17099	3.03s
258	0.00250	0.00216	1.15600	3.02s
258	0.00250	0.00216	1.15600	3.02s
259	0.00254	0.00214	1.18479	3.03s
259	0.00254	0.00214	1.18479	3.03s
260	0.00253	0.00218	1.16337	3.03s
260	0.00253	0.00218	1.16337	3.03s
261	0.00255	0.00214	1.19113	3.03s
261	0.00255	0.00214	1.19113	3.03s
262	0.00252	0.00214	1.17704	3.02s
262	0.00252	0.00214	1.17704	3.02s
263	0.00253	0.00215	1.17801	3.02s
263	0.00253	0.00215	1.17801	3.02s
264	0.00252	0.00214	1.17628	3.02s
264	0.00252	0.00214	1.17628	3.02s
265	0.00251	0.00218	1.15387	3.03s
265	0.00251	0.00218	1.15387	3.03s
266	0.00252	0.00216	1.16777	3.02s
266	0.00252	0.00216	1.16777	3.02s
267	0.00248	0.00217	1.14355	3.03s
267	0.00248	0.00217	1.14355	3.03s
268	0.00250	0.00215	1.16287	3.02s
268	0.00250	0.00215	1.16287	3.02s
269	0.00251	0.00213	1.17395	3.03s
269	0.00251	0.00213	1.17395	3.03s
270	0.00250	0.00214	1.16916	3.02s
270	0.00250	0.00214	1.16916	3.02s
271	0.00251	0.00214	1.17290	3.02s
271	0.00251	0.00214	1.17290	3.02s
272	0.00253	0.00213	1.18580	3.02s
272	0.00253	0.00213	1.18580	3.02s
273	0.00248	0.00215	1.15253	3.02s
273	0.00248	0.00215	1.15253	3.02s
274	0.00251	0.00213	1.18021	3.02s
274	0.00251	0.00213	1.18021	3.02s
275	0.00251	0.00215	1.16768	3.02s
275	0.00251	0.00215	1.16768	3.02s

276	0.00250	0.00213	1.17410	3.02s
276	0.00250	0.00213	1.17410	3.02s
277	0.00251	0.00213	1.17728	3.02s
277	0.00251	0.00213	1.17728	3.02s
278	0.00247	0.00212	1.16781	3.02s
278	0.00247	0.00212	1.16781	3.02s
279	0.00248	0.00214	1.16041	3.03s
279	0.00248	0.00214	1.16041	3.03s
280	0.00249	0.00214	1.16436	3.02s
280	0.00249	0.00214	1.16436	3.02s
281	0.00247	0.00214	1.15658	3.02s
281	0.00247	0.00214	1.15658	3.02s
282	0.00250	0.00213	1.17042	3.02s
282	0.00250	0.00213	1.17042	3.02s
283	0.00246	0.00211	1.16228	3.02s
283	0.00246	0.00211	1.16228	3.02s
284	0.00247	0.00216	1.14463	3.02s
284	0.00247	0.00216	1.14463	3.02s
285	0.00248	0.00211	1.17246	3.02s
285	0.00248	0.00211	1.17246	3.02s
286	0.00244	0.00212	1.15275	3.02s
286	0.00244	0.00212	1.15275	3.02s
287	0.00246	0.00210	1.17244	3.02s
287	0.00246	0.00210	1.17244	3.02s
288	0.00247	0.00212	1.16660	3.02s
288	0.00247	0.00212	1.16660	3.02s
289	0.00247	0.00212	1.16650	3.02s
289	0.00247	0.00212	1.16650	3.02s
290	0.00249	0.00212	1.17826	3.02s
290	0.00249	0.00212	1.17826	3.02s
291	0.00244	0.00215	1.13792	3.02s
291	0.00244	0.00215	1.13792	3.02s
292	0.00247	0.00212	1.16250	3.03s
292	0.00247	0.00212	1.16250	3.03s
293	0.00244	0.00211	1.15963	3.01s
293	0.00244	0.00211	1.15963	3.01s
294	0.00246	0.00212	1.16017	3.02s
294	0.00246	0.00212	1.16017	3.02s
295	0.00246	0.00211	1.16587	3.02s
295	0.00246	0.00211	1.16587	3.02s
296	0.00244	0.00208	1.17160	3.02s
296	0.00244	0.00208	1.17160	3.02s
297	0.00247	0.00211	1.17069	3.03s
297	0.00247	0.00211	1.17069	3.03s
298	0.00246	0.00210	1.17264	3.02s
298	0.00246	0.00210	1.17264	3.02s
299	0.00243	0.00211	1.15470	3.02s
299	0.00243	0.00211	1.15470	3.02s
300	0.00244	0.00210	1.16196	3.02s
300	0.00244	0.00210	1.16196	3.02s
301	0.00241	0.00209	1.14903	3.02s
301	0.00241	0.00209	1.14903	3.02s
302	0.00243	0.00211	1.14870	3.02s
302	0.00243	0.00211	1.14870	3.02s
303	0.00246	0.00210	1.17215	3.02s
303	0.00246	0.00210	1.17215	3.02s
304	0.00242	0.00213	1.13489	3.03s

```
304     0.00242     0.00213    1.13489  3.03s
305     0.00246     0.00212    1.16080  3.02s
305     0.00246     0.00212    1.16080  3.02s
306     0.00244     0.00213    1.14541  3.02s
306     0.00244     0.00213    1.14541  3.02s
307     0.00247     0.00214    1.15145  3.02s
307     0.00247     0.00214    1.15145  3.02s
308     0.00242     0.00210    1.15555  3.02s
308     0.00242     0.00210    1.15555  3.02s
309     0.00242     0.00209    1.15791  3.02s
309     0.00242     0.00209    1.15791  3.02s
310     0.00245     0.00210    1.16409  3.03s
310     0.00245     0.00210    1.16409  3.03s
311     0.00242     0.00210    1.15325  3.02s
311     0.00242     0.00210    1.15325  3.02s
312     0.00244     0.00211    1.15283  3.02s
312     0.00244     0.00211    1.15283  3.02s
313     0.00240     0.00210    1.14339  3.02s
313     0.00240     0.00210    1.14339  3.02s
314     0.00242     0.00211    1.14607  3.02s
314     0.00242     0.00211    1.14607  3.02s
315     0.00240     0.00209    1.14457  3.02s
315     0.00240     0.00209    1.14457  3.02s
316     0.00241     0.00210    1.14717  3.03s
316     0.00241     0.00210    1.14717  3.03s
317     0.00244     0.00208    1.17444  3.02s
317     0.00244     0.00208    1.17444  3.02s
318     0.00240     0.00209    1.14684  3.03s
318     0.00240     0.00209    1.14684  3.03s
319     0.00242     0.00210    1.15569  3.02s
319     0.00242     0.00210    1.15569  3.02s
320     0.00242     0.00207    1.17112  3.03s
320     0.00242     0.00207    1.17112  3.03s
321     0.00240     0.00211    1.14054  3.02s
321     0.00240     0.00211    1.14054  3.02s
322     0.00236     0.00210    1.12811  3.02s
322     0.00236     0.00210    1.12811  3.02s
323     0.00241     0.00212    1.13701  3.01s
323     0.00241     0.00212    1.13701  3.01s
324     0.00237     0.00210    1.12892  3.03s
324     0.00237     0.00210    1.12892  3.03s
325     0.00242     0.00210    1.15471  3.02s
325     0.00242     0.00210    1.15471  3.02s
326     0.00243     0.00210    1.15688  3.03s
326     0.00243     0.00210    1.15688  3.03s
327     0.00235     0.00208    1.13034  3.02s
327     0.00235     0.00208    1.13034  3.02s
328     0.00239     0.00210    1.13953  3.02s
328     0.00239     0.00210    1.13953  3.02s
329     0.00236     0.00207    1.13773  3.03s
329     0.00236     0.00207    1.13773  3.03s
330     0.00242     0.00209    1.15763  3.03s
330     0.00242     0.00209    1.15763  3.03s
331     0.00238     0.00209    1.13920  3.02s
331     0.00238     0.00209    1.13920  3.02s
332     0.00237     0.00208    1.13763  3.02s
332     0.00237     0.00208    1.13763  3.02s
```

333	0.00237	0.00209	1.13359	3.02s
333	0.00237	0.00209	1.13359	3.02s
334	0.00236	0.00208	1.13669	3.02s
334	0.00236	0.00208	1.13669	3.02s
335	0.00236	0.00210	1.12593	3.02s
335	0.00236	0.00210	1.12593	3.02s
336	0.00235	0.00208	1.13132	3.03s
336	0.00235	0.00208	1.13132	3.03s
337	0.00236	0.00210	1.12486	3.03s
337	0.00236	0.00210	1.12486	3.03s
338	0.00237	0.00207	1.14828	3.03s
338	0.00237	0.00207	1.14828	3.03s
339	0.00238	0.00209	1.13944	3.03s
339	0.00238	0.00209	1.13944	3.03s
340	0.00237	0.00209	1.13722	3.02s
340	0.00237	0.00209	1.13722	3.02s
341	0.00239	0.00208	1.15132	3.03s
341	0.00239	0.00208	1.15132	3.03s
342	0.00235	0.00208	1.12733	3.02s
342	0.00235	0.00208	1.12733	3.02s
343	0.00239	0.00212	1.12553	3.02s
343	0.00239	0.00212	1.12553	3.02s
344	0.00235	0.00208	1.12784	3.03s
344	0.00235	0.00208	1.12784	3.03s
345	0.00235	0.00208	1.13131	3.03s
345	0.00235	0.00208	1.13131	3.03s
346	0.00237	0.00206	1.15115	3.03s
346	0.00237	0.00206	1.15115	3.03s
347	0.00232	0.00209	1.11415	3.03s
347	0.00232	0.00209	1.11415	3.03s
348	0.00238	0.00207	1.14910	3.02s
348	0.00238	0.00207	1.14910	3.02s
349	0.00235	0.00209	1.12007	3.03s
349	0.00235	0.00209	1.12007	3.03s
350	0.00241	0.00206	1.16603	3.03s
350	0.00241	0.00206	1.16603	3.03s
351	0.00235	0.00207	1.13590	3.03s
351	0.00235	0.00207	1.13590	3.03s
352	0.00233	0.00210	1.11004	3.02s
352	0.00233	0.00210	1.11004	3.02s
353	0.00235	0.00206	1.14116	3.02s
353	0.00235	0.00206	1.14116	3.02s
354	0.00238	0.00207	1.15079	3.02s
354	0.00238	0.00207	1.15079	3.02s
355	0.00234	0.00209	1.12091	3.03s
355	0.00234	0.00209	1.12091	3.03s
356	0.00235	0.00204	1.14835	3.02s
356	0.00235	0.00204	1.14835	3.02s
357	0.00238	0.00208	1.14452	3.03s
357	0.00238	0.00208	1.14452	3.03s
358	0.00235	0.00207	1.14004	3.03s
358	0.00235	0.00207	1.14004	3.03s
359	0.00237	0.00206	1.15109	3.03s
359	0.00237	0.00206	1.15109	3.03s
360	0.00231	0.00206	1.12320	3.03s
360	0.00231	0.00206	1.12320	3.03s
361	0.00230	0.00208	1.10681	3.02s

361	0.00230	0.00208	1.10681	3.02s
362	0.00235	0.00207	1.13407	3.02s
362	0.00235	0.00207	1.13407	3.02s
363	0.00232	0.00204	1.14021	3.02s
363	0.00232	0.00204	1.14021	3.02s
364	0.00231	0.00204	1.13480	3.02s
364	0.00231	0.00204	1.13480	3.02s
365	0.00233	0.00207	1.12350	3.03s
365	0.00233	0.00207	1.12350	3.03s
366	0.00232	0.00205	1.13005	3.02s
366	0.00232	0.00205	1.13005	3.02s
367	0.00235	0.00205	1.14420	3.04s
367	0.00235	0.00205	1.14420	3.04s
368	0.00230	0.00205	1.12152	3.03s
368	0.00230	0.00205	1.12152	3.03s
369	0.00233	0.00206	1.13313	3.02s
369	0.00233	0.00206	1.13313	3.02s
370	0.00228	0.00206	1.10856	3.02s
370	0.00228	0.00206	1.10856	3.02s
371	0.00232	0.00206	1.12783	3.02s
371	0.00232	0.00206	1.12783	3.02s
372	0.00238	0.00204	1.16278	3.01s
372	0.00238	0.00204	1.16278	3.01s
373	0.00229	0.00207	1.10622	3.01s
373	0.00229	0.00207	1.10622	3.01s
374	0.00234	0.00203	1.15376	3.02s
374	0.00234	0.00203	1.15376	3.02s
375	0.00234	0.00205	1.14093	3.04s
375	0.00234	0.00205	1.14093	3.04s
376	0.00230	0.00204	1.12716	3.02s
376	0.00230	0.00204	1.12716	3.02s
377	0.00231	0.00203	1.13356	3.03s
377	0.00231	0.00203	1.13356	3.03s
378	0.00232	0.00206	1.12887	3.02s
378	0.00232	0.00206	1.12887	3.02s
379	0.00230	0.00205	1.12022	3.03s
379	0.00230	0.00205	1.12022	3.03s
380	0.00229	0.00206	1.10887	3.02s
380	0.00229	0.00206	1.10887	3.02s
381	0.00231	0.00203	1.13325	3.02s
381	0.00231	0.00203	1.13325	3.02s
382	0.00228	0.00207	1.10136	3.03s
382	0.00228	0.00207	1.10136	3.03s
383	0.00226	0.00207	1.09368	3.02s
383	0.00226	0.00207	1.09368	3.02s
384	0.00229	0.00207	1.10829	3.03s
384	0.00229	0.00207	1.10829	3.03s
385	0.00227	0.00204	1.11675	3.03s
385	0.00227	0.00204	1.11675	3.03s
386	0.00230	0.00207	1.11134	3.03s
386	0.00230	0.00207	1.11134	3.03s
387	0.00229	0.00206	1.10880	3.03s
387	0.00229	0.00206	1.10880	3.03s
388	0.00227	0.00203	1.11944	3.03s
388	0.00227	0.00203	1.11944	3.03s
389	0.00228	0.00205	1.10868	3.02s
389	0.00228	0.00205	1.10868	3.02s

390	0.00224	0.00207	1.08176	3.03s
390	0.00224	0.00207	1.08176	3.03s
391	0.00227	0.00205	1.11124	3.02s
391	0.00227	0.00205	1.11124	3.02s
392	0.00229	0.00206	1.11110	3.03s
392	0.00229	0.00206	1.11110	3.03s
393	0.00227	0.00200	1.13578	3.02s
393	0.00227	0.00200	1.13578	3.02s
394	0.00226	0.00204	1.11164	3.03s
394	0.00226	0.00204	1.11164	3.03s
395	0.00226	0.00208	1.08714	3.03s
395	0.00226	0.00208	1.08714	3.03s
396	0.00232	0.00205	1.13074	3.02s
396	0.00232	0.00205	1.13074	3.02s
397	0.00230	0.00204	1.12453	3.03s
397	0.00230	0.00204	1.12453	3.03s
398	0.00227	0.00202	1.12473	3.03s
398	0.00227	0.00202	1.12473	3.03s
399	0.00230	0.00204	1.12805	3.03s
399	0.00230	0.00204	1.12805	3.03s
400	0.00227	0.00204	1.11517	3.03s
400	0.00227	0.00204	1.11517	3.03s
401	0.00230	0.00204	1.12767	3.02s
401	0.00230	0.00204	1.12767	3.02s
402	0.00226	0.00203	1.10986	3.03s
402	0.00226	0.00203	1.10986	3.03s
403	0.00231	0.00205	1.12512	3.02s
403	0.00231	0.00205	1.12512	3.02s
404	0.00228	0.00202	1.12388	3.03s
404	0.00228	0.00202	1.12388	3.03s
405	0.00226	0.00205	1.10330	3.02s
405	0.00226	0.00205	1.10330	3.02s
406	0.00225	0.00205	1.09966	3.03s
406	0.00225	0.00205	1.09966	3.03s
407	0.00225	0.00205	1.09379	3.03s
407	0.00225	0.00205	1.09379	3.03s
408	0.00225	0.00203	1.11039	3.02s
408	0.00225	0.00203	1.11039	3.02s
409	0.00223	0.00204	1.09395	3.02s
409	0.00223	0.00204	1.09395	3.02s
410	0.00227	0.00202	1.12203	3.03s
410	0.00227	0.00202	1.12203	3.03s
411	0.00228	0.00205	1.10940	3.02s
411	0.00228	0.00205	1.10940	3.02s
412	0.00225	0.00203	1.10969	3.02s
412	0.00225	0.00203	1.10969	3.02s
413	0.00226	0.00206	1.09508	3.02s
413	0.00226	0.00206	1.09508	3.02s
414	0.00229	0.00202	1.13285	3.02s
414	0.00229	0.00202	1.13285	3.02s
415	0.00227	0.00204	1.11480	3.02s
415	0.00227	0.00204	1.11480	3.02s
416	0.00224	0.00202	1.10890	3.03s
416	0.00224	0.00202	1.10890	3.03s
417	0.00224	0.00203	1.10280	3.02s
417	0.00224	0.00203	1.10280	3.02s
418	0.00225	0.00205	1.09811	3.08s

418	0.00225	0.00205	1.09811	3.08s
419	0.00220	0.00204	1.08081	3.03s
419	0.00220	0.00204	1.08081	3.03s
420	0.00227	0.00202	1.12574	3.03s
420	0.00227	0.00202	1.12574	3.03s
421	0.00222	0.00204	1.09110	3.03s
421	0.00222	0.00204	1.09110	3.03s
422	0.00224	0.00202	1.11045	3.02s
422	0.00224	0.00202	1.11045	3.02s
423	0.00221	0.00203	1.09263	3.02s
423	0.00221	0.00203	1.09263	3.02s
424	0.00226	0.00203	1.11661	3.02s
424	0.00226	0.00203	1.11661	3.02s
425	0.00222	0.00203	1.09441	3.03s
425	0.00222	0.00203	1.09441	3.03s
426	0.00224	0.00204	1.09661	3.02s
426	0.00224	0.00204	1.09661	3.02s
427	0.00226	0.00204	1.10532	3.03s
427	0.00226	0.00204	1.10532	3.03s
428	0.00225	0.00205	1.09409	3.02s
428	0.00225	0.00205	1.09409	3.02s
429	0.00223	0.00202	1.10243	3.03s
429	0.00223	0.00202	1.10243	3.03s
430	0.00224	0.00203	1.10448	3.03s
430	0.00224	0.00203	1.10448	3.03s
431	0.00222	0.00202	1.10145	3.03s
431	0.00222	0.00202	1.10145	3.03s
432	0.00227	0.00204	1.11386	3.02s
432	0.00227	0.00204	1.11386	3.02s
433	0.00225	0.00204	1.10291	3.02s
433	0.00225	0.00204	1.10291	3.02s
434	0.00226	0.00202	1.11595	3.03s
434	0.00226	0.00202	1.11595	3.03s
435	0.00223	0.00205	1.08731	3.02s
435	0.00223	0.00205	1.08731	3.02s
436	0.00226	0.00204	1.10833	3.02s
436	0.00226	0.00204	1.10833	3.02s
437	0.00222	0.00204	1.08769	3.03s
437	0.00222	0.00204	1.08769	3.03s
438	0.00222	0.00202	1.09736	3.02s
438	0.00222	0.00202	1.09736	3.02s
439	0.00222	0.00199	1.11438	3.03s
439	0.00222	0.00199	1.11438	3.03s
440	0.00222	0.00200	1.10606	3.02s
440	0.00222	0.00200	1.10606	3.02s
441	0.00222	0.00202	1.10133	3.02s
441	0.00222	0.00202	1.10133	3.02s
442	0.00221	0.00205	1.07879	3.02s
442	0.00221	0.00205	1.07879	3.02s
443	0.00222	0.00202	1.10228	3.03s
443	0.00222	0.00202	1.10228	3.03s
444	0.00221	0.00202	1.09397	3.03s
444	0.00221	0.00202	1.09397	3.03s
445	0.00222	0.00202	1.10008	3.03s
445	0.00222	0.00202	1.10008	3.03s
446	0.00222	0.00203	1.09354	3.02s
446	0.00222	0.00203	1.09354	3.02s

447	0.00221	0.00201	1.10098	3.02s
447	0.00221	0.00201	1.10098	3.02s
448	0.00222	0.00201	1.10537	3.03s
448	0.00222	0.00201	1.10537	3.03s
449	0.00221	0.00201	1.10235	3.02s
449	0.00221	0.00201	1.10235	3.02s
450	0.00219	0.00201	1.08479	3.02s
450	0.00219	0.00201	1.08479	3.02s
451	0.00223	0.00201	1.11010	3.03s
451	0.00223	0.00201	1.11010	3.03s
452	0.00223	0.00201	1.10603	3.02s
452	0.00223	0.00201	1.10603	3.02s
453	0.00220	0.00200	1.09721	3.02s
453	0.00220	0.00200	1.09721	3.02s
454	0.00220	0.00199	1.10284	3.03s
454	0.00220	0.00199	1.10284	3.03s
455	0.00220	0.00202	1.08871	3.02s
455	0.00220	0.00202	1.08871	3.02s
456	0.00223	0.00201	1.10723	3.02s
456	0.00223	0.00201	1.10723	3.02s
457	0.00222	0.00201	1.10325	3.02s
457	0.00222	0.00201	1.10325	3.02s
458	0.00222	0.00200	1.11134	3.02s
458	0.00222	0.00200	1.11134	3.02s
459	0.00219	0.00202	1.08670	3.03s
459	0.00219	0.00202	1.08670	3.03s
460	0.00222	0.00200	1.10711	3.01s
460	0.00222	0.00200	1.10711	3.01s
461	0.00221	0.00200	1.10438	3.03s
461	0.00221	0.00200	1.10438	3.03s
462	0.00219	0.00203	1.08004	3.02s
462	0.00219	0.00203	1.08004	3.02s
463	0.00222	0.00203	1.09036	3.02s
463	0.00222	0.00203	1.09036	3.02s
464	0.00217	0.00200	1.08592	3.02s
464	0.00217	0.00200	1.08592	3.02s
465	0.00215	0.00202	1.06513	3.03s
465	0.00215	0.00202	1.06513	3.03s
466	0.00219	0.00201	1.08621	3.02s
466	0.00219	0.00201	1.08621	3.02s
467	0.00219	0.00202	1.08148	3.02s
467	0.00219	0.00202	1.08148	3.02s
468	0.00217	0.00201	1.08072	3.02s
468	0.00217	0.00201	1.08072	3.02s
469	0.00219	0.00200	1.09194	3.02s
469	0.00219	0.00200	1.09194	3.02s
470	0.00219	0.00204	1.07295	3.02s
470	0.00219	0.00204	1.07295	3.02s
471	0.00219	0.00200	1.09518	3.02s
471	0.00219	0.00200	1.09518	3.02s
472	0.00220	0.00202	1.09040	3.02s
472	0.00220	0.00202	1.09040	3.02s
473	0.00215	0.00199	1.08228	3.02s
473	0.00215	0.00199	1.08228	3.02s
474	0.00215	0.00202	1.06301	3.03s
474	0.00215	0.00202	1.06301	3.03s
475	0.00215	0.00201	1.06590	3.12s

475	0.00215	0.00201	1.06590	3.12s
476	0.00214	0.00202	1.06146	3.02s
476	0.00214	0.00202	1.06146	3.02s
477	0.00217	0.00201	1.07837	3.02s
477	0.00217	0.00201	1.07837	3.02s
478	0.00221	0.00200	1.10799	3.02s
478	0.00221	0.00200	1.10799	3.02s
479	0.00216	0.00201	1.07180	3.02s
479	0.00216	0.00201	1.07180	3.02s
480	0.00215	0.00203	1.06022	3.02s
480	0.00215	0.00203	1.06022	3.02s
481	0.00215	0.00202	1.06161	3.02s
481	0.00215	0.00202	1.06161	3.02s
482	0.00216	0.00200	1.07933	3.02s
482	0.00216	0.00200	1.07933	3.02s
483	0.00214	0.00200	1.06811	3.03s
483	0.00214	0.00200	1.06811	3.03s
484	0.00221	0.00200	1.10365	3.02s
484	0.00221	0.00200	1.10365	3.02s
485	0.00216	0.00199	1.08165	3.02s
485	0.00216	0.00199	1.08165	3.02s
486	0.00216	0.00203	1.06745	3.03s
486	0.00216	0.00203	1.06745	3.03s
487	0.00215	0.00200	1.07751	3.02s
487	0.00215	0.00200	1.07751	3.02s
488	0.00215	0.00200	1.07224	3.03s
488	0.00215	0.00200	1.07224	3.03s
489	0.00213	0.00200	1.06741	3.02s
489	0.00213	0.00200	1.06741	3.02s
490	0.00219	0.00201	1.08726	3.03s
490	0.00219	0.00201	1.08726	3.03s
491	0.00217	0.00203	1.06918	3.02s
491	0.00217	0.00203	1.06918	3.02s
492	0.00213	0.00201	1.06076	3.02s
492	0.00213	0.00201	1.06076	3.02s
493	0.00213	0.00200	1.06375	3.03s
493	0.00213	0.00200	1.06375	3.03s
494	0.00214	0.00199	1.07728	3.03s
494	0.00214	0.00199	1.07728	3.03s
495	0.00210	0.00200	1.04924	3.03s
495	0.00210	0.00200	1.04924	3.03s
496	0.00212	0.00201	1.05634	3.03s
496	0.00212	0.00201	1.05634	3.03s
497	0.00212	0.00199	1.06568	3.02s
497	0.00212	0.00199	1.06568	3.02s
498	0.00214	0.00200	1.06963	3.03s
498	0.00214	0.00200	1.06963	3.03s
499	0.00212	0.00200	1.06075	3.02s
499	0.00212	0.00200	1.06075	3.02s
500	0.00213	0.00200	1.06826	3.02s
500	0.00213	0.00200	1.06826	3.02s
501	0.00214	0.00199	1.07743	3.02s
501	0.00214	0.00199	1.07743	3.02s
502	0.00215	0.00203	1.06046	3.03s
502	0.00215	0.00203	1.06046	3.03s
503	0.00214	0.00198	1.08010	3.03s
503	0.00214	0.00198	1.08010	3.03s

504	0.00213	0.00201	1.05782	3.03s
504	0.00213	0.00201	1.05782	3.03s
505	0.00215	0.00200	1.07260	3.02s
505	0.00215	0.00200	1.07260	3.02s
506	0.00213	0.00200	1.06454	3.03s
506	0.00213	0.00200	1.06454	3.03s
507	0.00215	0.00199	1.08067	3.02s
507	0.00215	0.00199	1.08067	3.02s
508	0.00214	0.00202	1.05545	3.02s
508	0.00214	0.00202	1.05545	3.02s
509	0.00214	0.00197	1.08572	3.02s
509	0.00214	0.00197	1.08572	3.02s
510	0.00213	0.00200	1.06269	3.03s
510	0.00213	0.00200	1.06269	3.03s
511	0.00215	0.00200	1.07704	3.03s
511	0.00215	0.00200	1.07704	3.03s
512	0.00210	0.00199	1.05626	3.02s
512	0.00210	0.00199	1.05626	3.02s
513	0.00208	0.00197	1.05162	3.02s
513	0.00208	0.00197	1.05162	3.02s
514	0.00214	0.00199	1.07590	3.02s
514	0.00214	0.00199	1.07590	3.02s
515	0.00211	0.00198	1.06342	3.03s
515	0.00211	0.00198	1.06342	3.03s
516	0.00213	0.00198	1.07490	3.03s
516	0.00213	0.00198	1.07490	3.03s
517	0.00214	0.00202	1.05860	3.02s
517	0.00214	0.00202	1.05860	3.02s
518	0.00208	0.00198	1.04801	3.02s
518	0.00208	0.00198	1.04801	3.02s
519	0.00213	0.00199	1.07086	3.03s
519	0.00213	0.00199	1.07086	3.03s
520	0.00214	0.00200	1.06888	3.03s
520	0.00214	0.00200	1.06888	3.03s
521	0.00210	0.00199	1.05440	3.02s
521	0.00210	0.00199	1.05440	3.02s
522	0.00215	0.00201	1.07075	3.02s
522	0.00215	0.00201	1.07075	3.02s
523	0.00212	0.00198	1.07303	3.03s
523	0.00212	0.00198	1.07303	3.03s
524	0.00209	0.00198	1.05949	3.03s
524	0.00209	0.00198	1.05949	3.03s
525	0.00210	0.00197	1.06644	3.02s
525	0.00210	0.00197	1.06644	3.02s
526	0.00210	0.00198	1.06008	3.02s
526	0.00210	0.00198	1.06008	3.02s
527	0.00208	0.00198	1.04827	3.02s
527	0.00208	0.00198	1.04827	3.02s
528	0.00214	0.00198	1.07934	3.03s
528	0.00214	0.00198	1.07934	3.03s
529	0.00213	0.00199	1.07016	3.02s
529	0.00213	0.00199	1.07016	3.02s
530	0.00210	0.00196	1.07104	3.03s
530	0.00210	0.00196	1.07104	3.03s
531	0.00209	0.00198	1.05798	3.02s
531	0.00209	0.00198	1.05798	3.02s
532	0.00208	0.00200	1.03978	3.03s

```
532     0.00208     0.00200     1.03978  3.03s
533     0.00208     0.00199     1.04551  3.02s
533     0.00208     0.00199     1.04551  3.02s
534     0.00211     0.00199     1.05754  3.02s
534     0.00211     0.00199     1.05754  3.02s
535     0.00208     0.00196     1.06301  3.03s
535     0.00208     0.00196     1.06301  3.03s
536     0.00207     0.00196     1.05312  3.03s
536     0.00207     0.00196     1.05312  3.03s
537     0.00212     0.00197     1.07697  3.03s
537     0.00212     0.00197     1.07697  3.03s
538     0.00210     0.00198     1.05987  3.03s
538     0.00210     0.00198     1.05987  3.03s
539     0.00206     0.00195     1.05238  3.03s
539     0.00206     0.00195     1.05238  3.03s
540     0.00210     0.00198     1.06172  3.03s
540     0.00210     0.00198     1.06172  3.03s
541     0.00210     0.00196     1.07160  3.03s
541     0.00210     0.00196     1.07160  3.03s
542     0.00213     0.00197     1.08234  3.02s
542     0.00213     0.00197     1.08234  3.02s
543     0.00210     0.00197     1.06922  3.03s
543     0.00210     0.00197     1.06922  3.03s
544     0.00208     0.00195     1.06425  3.03s
544     0.00208     0.00195     1.06425  3.03s
545     0.00209     0.00196     1.06403  3.03s
545     0.00209     0.00196     1.06403  3.03s
546     0.00206     0.00197     1.04346  3.02s
546     0.00206     0.00197     1.04346  3.02s
547     0.00208     0.00198     1.04930  3.03s
547     0.00208     0.00198     1.04930  3.03s
548     0.00210     0.00199     1.05548  3.03s
548     0.00210     0.00199     1.05548  3.03s
549     0.00207     0.00198     1.04354  3.03s
549     0.00207     0.00198     1.04354  3.03s
550     0.00210     0.00196     1.06975  3.02s
550     0.00210     0.00196     1.06975  3.02s
551     0.00215     0.00199     1.07823  3.03s
551     0.00215     0.00199     1.07823  3.03s
552     0.00208     0.00196     1.05931  3.03s
552     0.00208     0.00196     1.05931  3.03s
553     0.00203     0.00197     1.03425  3.03s
553     0.00203     0.00197     1.03425  3.03s
554     0.00205     0.00195     1.05018  3.02s
554     0.00205     0.00195     1.05018  3.02s
555     0.00208     0.00197     1.05300  3.02s
555     0.00208     0.00197     1.05300  3.02s
556     0.00205     0.00196     1.04956  3.02s
556     0.00205     0.00196     1.04956  3.02s
557     0.00208     0.00197     1.05596  3.03s
557     0.00208     0.00197     1.05596  3.03s
558     0.00208     0.00197     1.05524  3.03s
558     0.00208     0.00197     1.05524  3.03s
559     0.00206     0.00200     1.02784  3.03s
559     0.00206     0.00200     1.02784  3.03s
560     0.00205     0.00196     1.04560  3.02s
560     0.00205     0.00196     1.04560  3.02s
```

561	0.00206	0.00194	1.06306	3.04s
561	0.00206	0.00194	1.06306	3.04s
562	0.00208	0.00196	1.05997	3.03s
562	0.00208	0.00196	1.05997	3.03s
563	0.00208	0.00197	1.05621	3.02s
563	0.00208	0.00197	1.05621	3.02s
564	0.00208	0.00196	1.06137	3.03s
564	0.00208	0.00196	1.06137	3.03s
565	0.00205	0.00196	1.04498	3.04s
565	0.00205	0.00196	1.04498	3.04s
566	0.00203	0.00199	1.01842	3.03s
566	0.00203	0.00199	1.01842	3.03s
567	0.00206	0.00197	1.04639	3.03s
567	0.00206	0.00197	1.04639	3.03s
568	0.00203	0.00196	1.03530	3.03s
568	0.00203	0.00196	1.03530	3.03s
569	0.00210	0.00197	1.06773	3.03s
569	0.00210	0.00197	1.06773	3.03s
570	0.00205	0.00195	1.04965	3.03s
570	0.00205	0.00195	1.04965	3.03s
571	0.00206	0.00198	1.03639	3.02s
571	0.00206	0.00198	1.03639	3.02s
572	0.00207	0.00195	1.06283	3.03s
572	0.00207	0.00195	1.06283	3.03s
573	0.00206	0.00194	1.05767	3.02s
573	0.00206	0.00194	1.05767	3.02s
574	0.00205	0.00195	1.05484	3.03s
574	0.00205	0.00195	1.05484	3.03s
575	0.00206	0.00196	1.05032	3.04s
575	0.00206	0.00196	1.05032	3.04s
576	0.00208	0.00195	1.06486	3.02s
576	0.00208	0.00195	1.06486	3.02s
577	0.00207	0.00197	1.04769	3.03s
577	0.00207	0.00197	1.04769	3.03s
578	0.00205	0.00196	1.04717	3.04s
578	0.00205	0.00196	1.04717	3.04s
579	0.00206	0.00197	1.04546	3.03s
579	0.00206	0.00197	1.04546	3.03s
580	0.00204	0.00196	1.03641	3.03s
580	0.00204	0.00196	1.03641	3.03s
581	0.00202	0.00196	1.02684	3.03s
581	0.00202	0.00196	1.02684	3.03s
582	0.00206	0.00195	1.05491	3.03s
582	0.00206	0.00195	1.05491	3.03s
583	0.00204	0.00198	1.03414	3.02s
583	0.00204	0.00198	1.03414	3.02s
584	0.00204	0.00198	1.02789	3.03s
584	0.00204	0.00198	1.02789	3.03s
585	0.00205	0.00194	1.05961	3.03s
585	0.00205	0.00194	1.05961	3.03s
586	0.00205	0.00196	1.04548	3.03s
586	0.00205	0.00196	1.04548	3.03s
587	0.00207	0.00195	1.06339	3.03s
587	0.00207	0.00195	1.06339	3.03s
588	0.00200	0.00198	1.01382	3.03s
588	0.00200	0.00198	1.01382	3.03s
589	0.00204	0.00195	1.04248	3.03s

589	0.00204	0.00195	1.04248	3.03s
590	0.00205	0.00196	1.04669	3.04s
590	0.00205	0.00196	1.04669	3.04s
591	0.00203	0.00196	1.03685	3.02s
591	0.00203	0.00196	1.03685	3.02s
592	0.00203	0.00196	1.03295	3.03s
592	0.00203	0.00196	1.03295	3.03s
593	0.00201	0.00198	1.01450	3.03s
593	0.00201	0.00198	1.01450	3.03s
594	0.00203	0.00196	1.03580	3.02s
594	0.00203	0.00196	1.03580	3.02s
595	0.00200	0.00194	1.02927	3.03s
595	0.00200	0.00194	1.02927	3.03s
596	0.00201	0.00194	1.03492	3.03s
596	0.00201	0.00194	1.03492	3.03s
597	0.00205	0.00193	1.05976	3.03s
597	0.00205	0.00193	1.05976	3.03s
598	0.00204	0.00195	1.04927	3.03s
598	0.00204	0.00195	1.04927	3.03s
599	0.00206	0.00195	1.05496	3.03s
599	0.00206	0.00195	1.05496	3.03s
600	0.00201	0.00197	1.02055	3.03s
600	0.00201	0.00197	1.02055	3.03s
601	0.00203	0.00194	1.04758	3.03s
601	0.00203	0.00194	1.04758	3.03s
602	0.00204	0.00195	1.04268	3.03s
602	0.00204	0.00195	1.04268	3.03s
603	0.00201	0.00194	1.03596	3.03s
603	0.00201	0.00194	1.03596	3.03s
604	0.00201	0.00195	1.03482	3.03s
604	0.00201	0.00195	1.03482	3.03s
605	0.00204	0.00194	1.05225	3.03s
605	0.00204	0.00194	1.05225	3.03s
606	0.00203	0.00195	1.03972	3.03s
606	0.00203	0.00195	1.03972	3.03s
607	0.00204	0.00196	1.03764	3.03s
607	0.00204	0.00196	1.03764	3.03s
608	0.00202	0.00196	1.03161	3.03s
608	0.00202	0.00196	1.03161	3.03s
609	0.00201	0.00193	1.04033	3.03s
609	0.00201	0.00193	1.04033	3.03s
610	0.00200	0.00196	1.01978	3.02s
610	0.00200	0.00196	1.01978	3.02s
611	0.00202	0.00195	1.03742	3.03s
611	0.00202	0.00195	1.03742	3.03s
612	0.00207	0.00195	1.06401	3.03s
612	0.00207	0.00195	1.06401	3.03s
613	0.00202	0.00195	1.03645	3.03s
613	0.00202	0.00195	1.03645	3.03s
614	0.00201	0.00197	1.01712	3.03s
614	0.00201	0.00197	1.01712	3.03s
615	0.00202	0.00194	1.03999	3.03s
615	0.00202	0.00194	1.03999	3.03s
616	0.00201	0.00193	1.04102	3.03s
616	0.00201	0.00193	1.04102	3.03s
617	0.00204	0.00196	1.03807	3.04s
617	0.00204	0.00196	1.03807	3.04s

```
    618     0.00199     0.00195    1.02096  3.04s
    618     0.00199     0.00195    1.02096  3.04s
    619     0.00199     0.00195    1.01733  3.04s
    619     0.00199     0.00195    1.01733  3.04s
    620     0.00206     0.00195    1.05260  3.02s
    620     0.00206     0.00195    1.05260  3.02s
left_eye_inner_corner_x      2271
left_eye_inner_corner_y      2271
right_eye_inner_corner_x     2268
right_eye_inner_corner_y     2268
left_eye_outer_corner_x      2267
left_eye_outer_corner_y      2267
right_eye_outer_corner_x     2268
right_eye_outer_corner_y     2268
left_eyebrow_inner_end_x     2270
left_eyebrow_inner_end_y     2270
right_eyebrow_inner_end_x    2270
right_eyebrow_inner_end_y    2270
left_eyebrow_outer_end_x     2225
left_eyebrow_outer_end_y     2225
right_eyebrow_outer_end_x    2236
right_eyebrow_outer_end_y    2236
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
Image                        7049
dtype: int64
left_eye_inner_corner_x      2155
left_eye_inner_corner_y      2155
right_eye_inner_corner_x     2155
right_eye_inner_corner_y     2155
left_eye_outer_corner_x      2155
left_eye_outer_corner_y      2155
right_eye_outer_corner_x     2155
right_eye_outer_corner_y     2155
left_eyebrow_inner_end_x     2155
left_eyebrow_inner_end_y     2155
right_eyebrow_inner_end_x    2155
right_eyebrow_inner_end_y    2155
left_eyebrow_outer_end_x     2155
left_eyebrow_outer_end_y     2155
right_eyebrow_outer_end_x    2155
right_eyebrow_outer_end_y    2155
mouth_left_corner_x          2155
mouth_left_corner_y          2155
mouth_right_corner_x         2155
mouth_right_corner_y         2155
mouth_center_top_lip_x       2155
mouth_center_top_lip_y       2155
Image                        2155
dtype: int64
Loaded parameters to layer 'conv1' (shape 32x1x3x3).
Loaded parameters to layer 'conv1' (shape 32).
Loaded parameters to layer 'conv2' (shape 64x32x2x2).
```

```
Loaded parameters to layer 'conv2' (shape 64).
Loaded parameters to layer 'conv3' (shape 128x64x2x2).
Loaded parameters to layer 'conv3' (shape 128).
Loaded parameters to layer 'hidden4' (shape 15488x1000).
Loaded parameters to layer 'hidden4' (shape 1000).
Loaded parameters to layer 'hidden5' (shape 1000x1000).
Loaded parameters to layer 'hidden5' (shape 1000).
Could not load parameters to layer 'output' because shapes did not match: 1
000x22 vs 1000x30.
Could not load parameters to layer 'output' because shapes did not match: 2
2 vs 30.
Training model for columns ('left_eye_inner_corner_x', 'left_eye_inner_corn
er_y', 'right_eye_inner_corner_x', 'right_eye_inner_corner_y', 'left_eye_ou
ter_corner_x', 'left_eye_outer_corner_y', 'right_eye_outer_corner_x', 'righ
t_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', 'ri
ght_eyebrow_outer_end_y', 'mouth_left_corner_x', 'mouth_left_corner_y', 'mo
uth_right_corner_x', 'mouth_right_corner_y', 'mouth_center_top_lip_x', 'mou
th_center_top_lip_y') for 4640 epochs
# Neural Network with 16553494 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1     32x94x94
  2  pool1     32x47x47
  3  dropout1  32x47x47
  4  conv2     64x46x46
  5  pool2     64x23x23
  6  dropout2  64x23x23
  7  conv3     128x22x22
  8  pool3     128x11x11
  9  dropout3  128x11x11
 10  hidden4   1000
 11  dropout4  1000
 12  hidden5   1000
 13  output    22

# Neural Network with 16553494 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1     32x94x94
  2  pool1     32x47x47
  3  dropout1  32x47x47
  4  conv2     64x46x46
  5  pool2     64x23x23
  6  dropout2  64x23x23
  7  conv3     128x22x22
  8  pool3     128x11x11
  9  dropout3  128x11x11
```

```
 10  hidden4   1000
 11  dropout4  1000
 12  hidden5   1000
 13  output    22

  epoch    trn loss    val loss    trn/val  dur
-------  ----------  ----------  ---------  -----
      1     0.10302     0.03179    3.24104  0.86s
  epoch    trn loss    val loss    trn/val  dur
-------  ----------  ----------  ---------  -----
      1     0.10302     0.03179    3.24104  0.86s
      2     0.01735     0.01433    1.21052  0.83s
      2     0.01735     0.01433    1.21052  0.83s
      3     0.01319     0.01510    0.87387  0.84s
      3     0.01319     0.01510    0.87387  0.84s
      4     0.01142     0.01600    0.71359  0.83s
      4     0.01142     0.01600    0.71359  0.83s
      5     0.01135     0.01438    0.78942  0.83s
      5     0.01135     0.01438    0.78942  0.83s
      6     0.01131     0.01190    0.95045  0.83s
      6     0.01131     0.01190    0.95045  0.83s
      7     0.01130     0.01029    1.09802  0.83s
      7     0.01130     0.01029    1.09802  0.83s
      8     0.01119     0.01019    1.09828  0.84s
      8     0.01119     0.01019    1.09828  0.84s
      9     0.01116     0.01076    1.03714  0.83s
      9     0.01116     0.01076    1.03714  0.83s
     10     0.01107     0.01093    1.01336  0.83s
     10     0.01107     0.01093    1.01336  0.83s
     11     0.01105     0.01088    1.01527  0.83s
     11     0.01105     0.01088    1.01527  0.83s
     12     0.01098     0.01077    1.01996  0.89s
     12     0.01098     0.01077    1.01996  0.89s
     13     0.01090     0.01074    1.01562  0.86s
     13     0.01090     0.01074    1.01562  0.86s
     14     0.01085     0.01096    0.98958  0.88s
     14     0.01085     0.01096    0.98958  0.88s
     15     0.01082     0.01083    0.99892  0.90s
     15     0.01082     0.01083    0.99892  0.90s
     16     0.01074     0.01062    1.01132  0.92s
     16     0.01074     0.01062    1.01132  0.92s
     17     0.01067     0.01036    1.02999  0.93s
     17     0.01067     0.01036    1.02999  0.93s
     18     0.01067     0.01011    1.05540  0.93s
     18     0.01067     0.01011    1.05540  0.93s
     19     0.01063     0.00992    1.07086  0.93s
     19     0.01063     0.00992    1.07086  0.93s
     20     0.01056     0.01030    1.02531  0.93s
     20     0.01056     0.01030    1.02531  0.93s
     21     0.01047     0.01038    1.00905  0.94s
     21     0.01047     0.01038    1.00905  0.94s
     22     0.01046     0.01014    1.03165  0.94s
     22     0.01046     0.01014    1.03165  0.94s
     23     0.01037     0.00992    1.04533  0.94s
     23     0.01037     0.00992    1.04533  0.94s
     24     0.01035     0.00983    1.05337  0.94s
     24     0.01035     0.00983    1.05337  0.94s
```

```
25      0.01028     0.01008    1.01994  0.93s
25      0.01028     0.01008    1.01994  0.93s
26      0.01029     0.01022    1.00669  0.93s
26      0.01029     0.01022    1.00669  0.93s
27      0.01024     0.01003    1.02102  0.94s
27      0.01024     0.01003    1.02102  0.94s
28      0.01015     0.00984    1.03217  0.94s
28      0.01015     0.00984    1.03217  0.94s
29      0.01016     0.00985    1.03152  0.94s
29      0.01016     0.00985    1.03152  0.94s
30      0.01009     0.01000    1.00915  0.94s
30      0.01009     0.01000    1.00915  0.94s
31      0.01008     0.00978    1.03103  0.94s
31      0.01008     0.00978    1.03103  0.94s
32      0.01004     0.00989    1.01543  0.94s
32      0.01004     0.00989    1.01543  0.94s
33      0.01001     0.00982    1.01940  0.94s
33      0.01001     0.00982    1.01940  0.94s
34      0.00993     0.00982    1.01199  0.94s
34      0.00993     0.00982    1.01199  0.94s
35      0.00993     0.00996    0.99702  0.94s
35      0.00993     0.00996    0.99702  0.94s
36      0.00990     0.00961    1.02960  0.94s
36      0.00990     0.00961    1.02960  0.94s
37      0.00990     0.00927    1.06773  0.94s
37      0.00990     0.00927    1.06773  0.94s
38      0.00983     0.00909    1.08109  0.94s
38      0.00983     0.00909    1.08109  0.94s
39      0.00982     0.00915    1.07342  0.94s
39      0.00982     0.00915    1.07342  0.94s
40      0.00977     0.00910    1.07422  0.94s
40      0.00977     0.00910    1.07422  0.94s
41      0.00975     0.00925    1.05437  0.94s
41      0.00975     0.00925    1.05437  0.94s
42      0.00976     0.00952    1.02548  0.94s
42      0.00976     0.00952    1.02548  0.94s
43      0.00972     0.00965    1.00719  0.93s
43      0.00972     0.00965    1.00719  0.93s
44      0.00971     0.00955    1.01717  0.94s
44      0.00971     0.00955    1.01717  0.94s
45      0.00965     0.00958    1.00681  0.94s
45      0.00965     0.00958    1.00681  0.94s
46      0.00967     0.00962    1.00467  0.93s
46      0.00967     0.00962    1.00467  0.93s
47      0.00962     0.00970    0.99216  0.94s
47      0.00962     0.00970    0.99216  0.94s
48      0.00964     0.00925    1.04208  0.94s
48      0.00964     0.00925    1.04208  0.94s
49      0.00955     0.00868    1.09963  0.94s
49      0.00955     0.00868    1.09963  0.94s
50      0.00956     0.00852    1.12176  0.94s
50      0.00956     0.00852    1.12176  0.94s
51      0.00957     0.00895    1.06876  0.94s
51      0.00957     0.00895    1.06876  0.94s
52      0.00953     0.00929    1.02531  0.94s
52      0.00953     0.00929    1.02531  0.94s
53      0.00961     0.00911    1.05479  0.94s
```

53	0.00961	0.00911	1.05479	0.94s
54	0.00955	0.00893	1.06931	0.94s
54	0.00955	0.00893	1.06931	0.94s
55	0.00946	0.00881	1.07422	0.95s
55	0.00946	0.00881	1.07422	0.95s
56	0.00949	0.00906	1.04783	0.94s
56	0.00949	0.00906	1.04783	0.94s
57	0.00948	0.00927	1.02243	0.94s
57	0.00948	0.00927	1.02243	0.94s
58	0.00946	0.00931	1.01627	0.94s
58	0.00946	0.00931	1.01627	0.94s
59	0.00946	0.00900	1.05094	0.94s
59	0.00946	0.00900	1.05094	0.94s
60	0.00944	0.00885	1.06686	0.94s
60	0.00944	0.00885	1.06686	0.94s
61	0.00942	0.00877	1.07470	0.94s
61	0.00942	0.00877	1.07470	0.94s
62	0.00938	0.00860	1.09064	0.94s
62	0.00938	0.00860	1.09064	0.94s
63	0.00935	0.00870	1.07409	0.94s
63	0.00935	0.00870	1.07409	0.94s
64	0.00939	0.00907	1.03535	0.94s
64	0.00939	0.00907	1.03535	0.94s
65	0.00936	0.00939	0.99720	0.94s
65	0.00936	0.00939	0.99720	0.94s
66	0.00938	0.00939	0.99806	0.94s
66	0.00938	0.00939	0.99806	0.94s
67	0.00932	0.00911	1.02270	0.94s
67	0.00932	0.00911	1.02270	0.94s
68	0.00938	0.00877	1.06958	0.94s
68	0.00938	0.00877	1.06958	0.94s
69	0.00933	0.00863	1.08056	0.94s
69	0.00933	0.00863	1.08056	0.94s
70	0.00933	0.00892	1.04589	0.94s
70	0.00933	0.00892	1.04589	0.94s
71	0.00931	0.00898	1.03651	0.94s
71	0.00931	0.00898	1.03651	0.94s
72	0.00930	0.00883	1.05322	0.94s
72	0.00930	0.00883	1.05322	0.94s
73	0.00926	0.00889	1.04235	0.94s
73	0.00926	0.00889	1.04235	0.94s
74	0.00927	0.00902	1.02817	0.94s
74	0.00927	0.00902	1.02817	0.94s
75	0.00929	0.00914	1.01681	0.94s
75	0.00929	0.00914	1.01681	0.94s
76	0.00928	0.00897	1.03467	0.94s
76	0.00928	0.00897	1.03467	0.94s
77	0.00929	0.00880	1.05613	0.94s
77	0.00929	0.00880	1.05613	0.94s
78	0.00921	0.00875	1.05270	0.94s
78	0.00921	0.00875	1.05270	0.94s
79	0.00927	0.00856	1.08211	0.94s
79	0.00927	0.00856	1.08211	0.94s
80	0.00924	0.00835	1.10760	0.94s
80	0.00924	0.00835	1.10760	0.94s
81	0.00925	0.00852	1.08533	0.94s
81	0.00925	0.00852	1.08533	0.94s

82	0.00923	0.00884	1.04428	0.94s
82	0.00923	0.00884	1.04428	0.94s
83	0.00924	0.00895	1.03272	0.94s
83	0.00924	0.00895	1.03272	0.94s
84	0.00920	0.00898	1.02500	0.94s
84	0.00920	0.00898	1.02500	0.94s
85	0.00922	0.00885	1.04211	0.95s
85	0.00922	0.00885	1.04211	0.95s
86	0.00923	0.00857	1.07737	0.94s
86	0.00923	0.00857	1.07737	0.94s
87	0.00919	0.00843	1.09042	0.94s
87	0.00919	0.00843	1.09042	0.94s
88	0.00919	0.00865	1.06154	0.94s
88	0.00919	0.00865	1.06154	0.94s
89	0.00917	0.00890	1.03039	0.94s
89	0.00917	0.00890	1.03039	0.94s
90	0.00921	0.00895	1.02956	0.94s
90	0.00921	0.00895	1.02956	0.94s
91	0.00911	0.00860	1.05882	0.94s
91	0.00911	0.00860	1.05882	0.94s
92	0.00915	0.00851	1.07433	0.94s
92	0.00915	0.00851	1.07433	0.94s
93	0.00915	0.00863	1.06071	0.94s
93	0.00915	0.00863	1.06071	0.94s
94	0.00916	0.00858	1.06838	0.94s
94	0.00916	0.00858	1.06838	0.94s
95	0.00914	0.00855	1.06791	0.94s
95	0.00914	0.00855	1.06791	0.94s
96	0.00908	0.00869	1.04569	0.94s
96	0.00908	0.00869	1.04569	0.94s
97	0.00913	0.00874	1.04430	0.94s
97	0.00913	0.00874	1.04430	0.94s
98	0.00913	0.00851	1.07214	0.94s
98	0.00913	0.00851	1.07214	0.94s
99	0.00912	0.00862	1.05698	0.94s
99	0.00912	0.00862	1.05698	0.94s
100	0.00908	0.00881	1.03162	0.95s
100	0.00908	0.00881	1.03162	0.95s
101	0.00909	0.00885	1.02709	0.94s
101	0.00909	0.00885	1.02709	0.94s
102	0.00912	0.00864	1.05516	0.94s
102	0.00912	0.00864	1.05516	0.94s
103	0.00909	0.00862	1.05440	0.94s
103	0.00909	0.00862	1.05440	0.94s
104	0.00908	0.00853	1.06511	0.94s
104	0.00908	0.00853	1.06511	0.94s
105	0.00911	0.00868	1.05014	0.94s
105	0.00911	0.00868	1.05014	0.94s
106	0.00904	0.00875	1.03379	0.94s
106	0.00904	0.00875	1.03379	0.94s
107	0.00907	0.00882	1.02870	0.94s
107	0.00907	0.00882	1.02870	0.94s
108	0.00908	0.00903	1.00498	0.94s
108	0.00908	0.00903	1.00498	0.94s
109	0.00905	0.00862	1.05004	0.94s
109	0.00905	0.00862	1.05004	0.94s
110	0.00904	0.00840	1.07649	0.94s

```
    110     0.00904     0.00840     1.07649   0.94s
    111     0.00908     0.00827     1.09697   0.94s
    111     0.00908     0.00827     1.09697   0.94s
    112     0.00904     0.00837     1.07921   0.94s
    112     0.00904     0.00837     1.07921   0.94s
    113     0.00902     0.00841     1.07324   0.94s
    113     0.00902     0.00841     1.07324   0.94s
    114     0.00900     0.00832     1.08188   0.94s
    114     0.00900     0.00832     1.08188   0.94s
    115     0.00903     0.00842     1.07277   0.94s
    115     0.00903     0.00842     1.07277   0.94s
    116     0.00904     0.00866     1.04353   0.94s
    116     0.00904     0.00866     1.04353   0.94s
    117     0.00903     0.00857     1.05365   0.94s
    117     0.00903     0.00857     1.05365   0.94s
    118     0.00903     0.00848     1.06468   0.95s
    118     0.00903     0.00848     1.06468   0.95s
    119     0.00906     0.00833     1.08712   0.94s
    119     0.00906     0.00833     1.08712   0.94s
    120     0.00903     0.00851     1.06161   0.94s
    120     0.00903     0.00851     1.06161   0.94s
    121     0.00904     0.00878     1.02994   0.94s
    121     0.00904     0.00878     1.02994   0.94s
    122     0.00897     0.00880     1.02033   0.94s
    122     0.00897     0.00880     1.02033   0.94s
    123     0.00898     0.00867     1.03486   0.94s
    123     0.00898     0.00867     1.03486   0.94s
    124     0.00899     0.00845     1.06363   0.94s
    124     0.00899     0.00845     1.06363   0.94s
    125     0.00899     0.00849     1.05943   0.94s
    125     0.00899     0.00849     1.05943   0.94s
    126     0.00901     0.00862     1.04441   0.94s
    126     0.00901     0.00862     1.04441   0.94s
    127     0.00898     0.00864     1.04032   0.95s
    127     0.00898     0.00864     1.04032   0.95s
    128     0.00896     0.00852     1.05121   0.94s
    128     0.00896     0.00852     1.05121   0.94s
    129     0.00896     0.00848     1.05623   0.94s
    129     0.00896     0.00848     1.05623   0.94s
    130     0.00898     0.00810     1.10786   0.94s
    130     0.00898     0.00810     1.10786   0.94s
    131     0.00895     0.00811     1.10365   0.94s
    131     0.00895     0.00811     1.10365   0.94s
    132     0.00897     0.00851     1.05367   0.94s
    132     0.00897     0.00851     1.05367   0.94s
    133     0.00895     0.00879     1.01917   0.94s
    133     0.00895     0.00879     1.01917   0.94s
```

RMSE Score for Specialists

In [17]:

```
# open specialists pickle
specialist_pickle_file ="/home/sdorairaj/dev/W207Spring2017_FacialKeyPoints
Detection_Backup/Improvement/net-specialists_3_no_early_stopping.pickle"
specialists = pickle.load(open(specialist_pickle_file,'rb'))
```

```
X = load2d(test=True)[0]
y_pred = np.empty((X.shape[0], 0))

count=0
for model in specialists.values():
    net8_train_loss = np.array([i["train_loss"] for i in model.train_histo
ry_])
    net8_valid_loss = np.array([i["valid_loss"] for i in model.train_histo
ry_])

    print "\nSpecialist:",count
    print "Training Loss:",net8_train_loss[-1]
    print "Validation Loss:",net8_valid_loss[-1]

    count+=1
```

```
ImageId    1783
Image      1783
dtype: int64
ImageId    1783
Image      1783
dtype: int64

Specialist: 0
Training Loss: 0.00173984912224
Validation Loss: 0.00204543352061

Specialist: 1
Training Loss: 0.00843605317452
Validation Loss: 0.00821091711258
```

In [18]:

```
# RMSE score o
print "RMSE score for first specialist:"
validation_loss = 0.00204543

print np.sqrt(validation_loss)*48 # normalize to [-1,1]

print "RMSE score for second specialist:"
validation_loss = 0.00821092

print np.sqrt(validation_loss)*48 # normalize to [-1,1]
```

```
RMSE score for first specialist:
2.17086865563
RMSE score for second specialist:
4.34947809283
```

Contrary to what we had hoped, the specialists did not yield the desired improvements and the model was only marginally better than the net7 model. Submitting the specialists to Kaggle resulted in a Kaggle Score of 3.45 which is less than the score of net7 which was 3.45.

It is unclear why this is the case. Perhaps the additional training data caused the model to overfit and therefore fail to generalize sufficiently to improve accuracy on the test set.

Generating submission file for specialists

In [41]:
```
submit_aggregated_models(specialist_pickle_file)
```

```
ImageId    1783
Image      1783
dtype: int64
ImageId    1783
Image      1783
dtype: int64
Wrote submission-2017-04-23T12-33-01.542840.csv
```

8. Train a deeper net

Based on the research VGG (Visual Geometry Group) in Oxford, making the convolution net deeper could help the accuracty with image recognition (http://www.robots.ox.ac.uk/~vgg/research/very_deep/). We add an additional convolution layer to each round of convolution/pool/dropout and add an additional hidden layer with the same units as others while keep other settings the same. The idea behind this is that 2 convolution layer will recognize more surrounding pixels than 1 layer that if the previous net only recognize an eyeball, this might be able to see an eye all together.

In retrospect, an additional hidden layer might not be necessary since it increases both training time and memory needed with minimal performance improvements.

In [5]:
```
## Marked all added code with "##"
import theano

net9 = NeuralNet(
    layers=[
        ('input', layers.InputLayer),
        ('conv1_1', layers.Conv2DLayer),
        ('conv1_2', layers.Conv2DLayer), ##
        ('pool1', layers.MaxPool2DLayer),
        ('dropout1', layers.DropoutLayer),
        ('conv2_1', layers.Conv2DLayer),
        ('conv2_2', layers.Conv2DLayer), ##
        ('pool2', layers.MaxPool2DLayer),
        ('dropout2', layers.DropoutLayer),
        ('conv3_1', layers.Conv2DLayer),
        ('conv3_2', layers.Conv2DLayer), ##
        ('pool3', layers.MaxPool2DLayer),
        ('dropout3', layers.DropoutLayer),
        ('hidden4', layers.DenseLayer),
        ('dropout4', layers.DropoutLayer),
        ('hidden5', layers.DenseLayer),
        ('dropout5', layers.DropoutLayer), ##
        ('hidden6', layers.DenseLayer), ##
        ('output', layers.DenseLayer),
        ],
    input_shape=(None, 1, 96, 96),
    conv1_1_num_filters=32, conv1_1_filter_size=(3, 3), conv1_1_pad=1,
    conv1_2_num_filters=32, conv1_2_filter_size=(3, 3), conv1_2_pad=1,
    pool1_pool_size=(2, 2),
```

```
    dropout1_p=0.1,
    conv2_1_num_filters=64, conv2_1_filter_size=(3, 3), conv2_1_pad=1,
    conv2_2_num_filters=64, conv2_2_filter_size=(3, 3), conv2_2_pad=1,
    pool2_pool_size=(2, 2),
    dropout2_p=0.2,
    conv3_1_num_filters=128, conv3_1_filter_size=(3, 3), conv3_1_pad=1,
    conv3_2_num_filters=128, conv3_2_filter_size=(3, 3), conv3_2_pad=1,
    pool3_pool_size=(2, 2),
    dropout3_p=0.3,
    hidden4_num_units=1000,
    dropout4_p=0.5,
    hidden5_num_units=1000,
    dropout5_p=0.5,
    hidden6_num_units=1000,
    output_num_units=30, output_nonlinearity=None,

    update_learning_rate=theano.shared(float32(0.03)),
    update_momentum=theano.shared(float32(0.9)),

    regression=True,
    batch_iterator_train=FlipBatchIterator(batch_size=128),
    on_epoch_finished=[
        AdjustVariable('update_learning_rate', start=0.03, stop=0.0001),
        AdjustVariable('update_momentum', start=0.9, stop=0.999),
        ],
    max_epochs=10000,
    verbose=1,
    )

sys.setrecursionlimit(10000)

X, y = load2d()
net9.fit(X, y)

with open('net9.pickle', 'wb') as f:
    pickle.dump(net9, f, -1)
```

```
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
dtype: int64
left_eye_center_x             2140
left_eye_center_y             2140
right_eye_center_x            2140
right_eye_center_y            2140
left_eye_inner_corner_x       2140
left_eye_inner_corner_y       2140
left_eye_outer_corner_x       2140
left_eye_outer_corner_y       2140
right_eye_inner_corner_x      2140
right_eye_inner_corner_y      2140
right_eye_outer_corner_x      2140
right_eye_outer_corner_y      2140
left_eyebrow_inner_end_x      2140
left_eyebrow_inner_end_y      2140
left_eyebrow_outer_end_x      2140
left_eyebrow_outer_end_y      2140
right_eyebrow_inner_end_x     2140
right_eyebrow_inner_end_y     2140
right_eyebrow_outer_end_x     2140
right_eyebrow_outer_end_y     2140
nose_tip_x                   2140
nose_tip_y                   2140
mouth_left_corner_x           2140
mouth_left_corner_y           2140
mouth_right_corner_x          2140
mouth_right_corner_y          2140
mouth_center_top_lip_x        2140
mouth_center_top_lip_y        2140
mouth_center_bottom_lip_x     2140
mouth_center_bottom_lip_y     2140
Image                        2140
dtype: int64
# Neural Network with 20751462 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1_1   32x96x96
  2  conv1_2   32x96x96
  3  pool1     32x48x48
  4  dropout1  32x48x48
  5  conv2_1   64x48x48
  6  conv2_2   64x48x48
```

```
  7  pool2     64x24x24
  8  dropout2  64x24x24
  9  conv3_1   128x24x24
 10  conv3_2   128x24x24
 11  pool3     128x12x12
 12  dropout3  128x12x12
 13  hidden4   1000
 14  dropout4  1000
 15  hidden5   1000
 16  dropout5  1000
 17  hidden6   1000
 18  output    30

  epoch    trn loss    val loss    trn/val  dur
-------  ----------  ----------  ---------  -----
      1     0.12159     0.04757    2.55612  3.05s
      2     0.02152     0.01075    2.00195  3.02s
      3     0.00854     0.00716    1.19308  3.02s
      4     0.00672     0.00599    1.12181  3.02s
      5     0.00626     0.00566    1.10556  3.04s
      6     0.00601     0.00551    1.09163  3.39s
      7     0.00575     0.00527    1.09145  3.48s
      8     0.00558     0.00522    1.07021  3.48s
      9     0.00543     0.00499    1.08812  3.47s
     10     0.00534     0.00495    1.08030  3.47s
     11     0.00528     0.00480    1.09897  3.52s
     12     0.00519     0.00472    1.09843  3.51s
     13     0.00515     0.00465    1.10683  3.51s
     14     0.00510     0.00460    1.10828  3.51s
     15     0.00501     0.00457    1.09803  3.51s
     16     0.00496     0.00453    1.09528  3.51s
     17     0.00493     0.00449    1.09681  3.51s
     18     0.00486     0.00447    1.08695  3.51s
     19     0.00486     0.00446    1.08895  3.51s
     20     0.00487     0.00444    1.09634  3.51s
     21     0.00481     0.00441    1.09010  3.51s
     22     0.00475     0.00437    1.08884  3.51s
     23     0.00478     0.00436    1.09701  3.51s
     24     0.00471     0.00435    1.08438  3.51s
     25     0.00472     0.00434    1.08641  3.51s
     26     0.00472     0.00434    1.08723  3.51s
     27     0.00469     0.00431    1.08886  3.55s
     28     0.00466     0.00431    1.07961  3.52s
     29     0.00469     0.00430    1.08873  3.51s
     30     0.00466     0.00428    1.08872  3.50s
     31     0.00464     0.00427    1.08594  3.50s
     32     0.00464     0.00427    1.08663  3.50s
     33     0.00466     0.00427    1.09154  3.50s
     34     0.00464     0.00426    1.08956  3.51s
     35     0.00461     0.00426    1.08194  3.51s
     36     0.00459     0.00425    1.07946  3.51s
     37     0.00460     0.00425    1.08370  3.50s
     38     0.00460     0.00425    1.08175  3.51s
     39     0.00459     0.00425    1.08138  3.50s
     40     0.00459     0.00425    1.08127  3.51s
     41     0.00459     0.00424    1.08273  3.50s
     42     0.00456     0.00424    1.07711  3.51s
```

43	0.00457	0.00424	1.07841	3.51s
44	0.00457	0.00424	1.07716	3.50s
45	0.00456	0.00424	1.07537	3.50s
46	0.00455	0.00425	1.07197	3.50s
47	0.00454	0.00424	1.06956	3.50s
48	0.00458	0.00425	1.07819	3.50s
49	0.00457	0.00424	1.07607	3.50s
50	0.00455	0.00424	1.07351	3.50s
51	0.00458	0.00424	1.07849	3.50s
52	0.00457	0.00424	1.07735	3.50s
53	0.00454	0.00424	1.07204	3.51s
54	0.00453	0.00424	1.06869	3.50s
55	0.00455	0.00425	1.07231	3.50s
56	0.00455	0.00425	1.06975	3.50s
57	0.00454	0.00424	1.06897	3.50s
58	0.00454	0.00424	1.07215	3.50s
59	0.00454	0.00424	1.07212	3.50s
60	0.00456	0.00424	1.07524	3.50s
61	0.00453	0.00424	1.06955	3.50s
62	0.00453	0.00423	1.07007	3.50s
63	0.00453	0.00423	1.07004	3.50s
64	0.00454	0.00424	1.07215	3.50s
65	0.00453	0.00424	1.06782	3.50s
66	0.00453	0.00424	1.06729	3.50s
67	0.00454	0.00424	1.07020	3.50s
68	0.00453	0.00424	1.06852	3.50s
69	0.00452	0.00423	1.06866	3.50s
70	0.00452	0.00424	1.06649	3.53s
71	0.00452	0.00424	1.06664	3.50s
72	0.00452	0.00424	1.06776	3.50s
73	0.00451	0.00423	1.06569	3.51s
74	0.00452	0.00423	1.06868	3.50s
75	0.00452	0.00423	1.06778	3.50s
76	0.00452	0.00423	1.06839	3.51s
77	0.00451	0.00423	1.06697	3.50s
78	0.00450	0.00423	1.06596	3.50s
79	0.00451	0.00423	1.06749	3.50s
80	0.00452	0.00423	1.06755	3.51s
81	0.00449	0.00423	1.06232	3.50s
82	0.00451	0.00423	1.06779	3.50s
83	0.00450	0.00422	1.06609	3.50s
84	0.00452	0.00422	1.06881	3.50s
85	0.00449	0.00423	1.06284	3.50s
86	0.00450	0.00423	1.06567	3.51s
87	0.00451	0.00423	1.06604	3.50s
88	0.00450	0.00422	1.06474	3.50s
89	0.00451	0.00422	1.06819	3.50s
90	0.00450	0.00422	1.06665	3.50s
91	0.00449	0.00422	1.06250	3.51s
92	0.00449	0.00423	1.06296	3.50s
93	0.00451	0.00423	1.06575	3.51s
94	0.00449	0.00423	1.06191	3.51s
95	0.00451	0.00422	1.06750	3.51s
96	0.00450	0.00423	1.06399	3.50s
97	0.00449	0.00423	1.06224	3.50s
98	0.00450	0.00423	1.06354	3.50s
99	0.00448	0.00422	1.06084	3.50s

```
    100     0.00449     0.00423     1.06196  3.50s
    101     0.00449     0.00423     1.06202  3.50s
    102     0.00448     0.00423     1.05902  3.50s
    103     0.00448     0.00423     1.05919  3.50s
    104     0.00447     0.00422     1.05856  3.50s
    105     0.00448     0.00422     1.06106  3.50s
    106     0.00449     0.00422     1.06268  3.50s
    107     0.00448     0.00423     1.06099  3.50s
    108     0.00448     0.00423     1.05880  3.50s
    109     0.00448     0.00423     1.06084  3.50s
    110     0.00449     0.00422     1.06309  3.50s
    111     0.00449     0.00422     1.06282  3.51s
    112     0.00448     0.00422     1.06162  3.50s
    113     0.00448     0.00422     1.06092  3.50s
    114     0.00447     0.00422     1.05827  3.50s
    115     0.00447     0.00422     1.05984  3.50s
    116     0.00448     0.00422     1.06127  3.54s
    117     0.00448     0.00422     1.06152  3.50s
```

Net9 RMSE Score

The below table captures validation loss and training loss at the end of training for 10000 epochs

In [ ]:
```
  epoch    trn loss    val loss    trn/val  dur
-------  ----------  ----------  ---------  -----
      1     0.11222     0.01096    1.56909  9.12s
      3     0.00897     0.00779    1.15130  9.12s
      4     0.00709     0.00691    1.02641  9.12s
      5     0.00661     0.00650    1.01574  9.12s

   9995     0.00089     0.00078    1.14358  9.07s
   9996     0.00089     0.00078    1.14017  9.06s
   9997     0.00089     0.00078    1.13609  9.08s
   9998     0.00088     0.00078    1.12294  9.07s
   9999     0.00088     0.00078    1.12754  9.06s
  10000     0.00089     0.00078    1.13674  9.06s
```

In [ ]:
```
# RMSE score for net9
validation_loss = 0.00078

print np.sqrt(validation_loss)*48 # normalize to [-1,1]
```

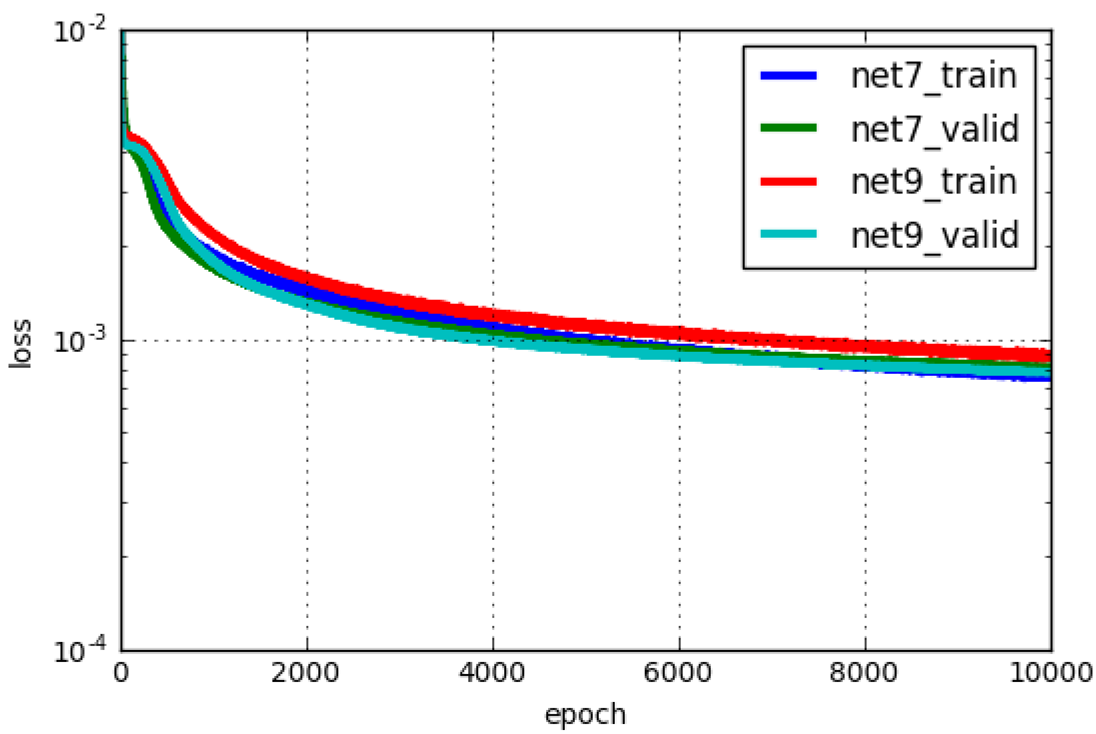
Evaluating net7 performance with net9

In the below section we compare the validation loss from net7 and net9 to examine further scope for improvement.

In [ ]:
```
# Load from pickle file after first run
net9_pickle = "/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetection_Misc/net9.pickle"
net7_pickle ="/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetection_Misc/net7_10000epochs.pickle"
net9 = pickle.load(open(net9_pickle,'rb'))
```

```
net7 = pickle.load(open(net7_pickle,'rb'))
```

In [7]:
```
# plot
net9_train_loss = np.array([i["train_loss"] for i in net9.train_history_])
net9_valid_loss = np.array([i["valid_loss"] for i in net9.train_history_])
net7_train_loss = np.array([i["train_loss"] for i in net7.train_history_])
net7_valid_loss = np.array([i["valid_loss"] for i in net7.train_history_])
pyplot.plot(net7_train_loss, linewidth=3, label="net7_train")
pyplot.plot(net7_valid_loss, linewidth=3, label="net7_valid")
pyplot.plot(net9_train_loss, linewidth=3, label="net9_train")
pyplot.plot(net9_valid_loss, linewidth=3, label="net9_valid")
pyplot.grid()
pyplot.legend()
pyplot.xlabel("epoch")
pyplot.ylabel("loss")
pyplot.ylim(1e-4, 1e-2)
pyplot.yscale("log")
pyplot.show()
```



RMSE Score

Making the net deeper does decrease the val_loss, however according to the plot above, it is only a marginal improvement from net7. It now get us to a val_loss of 0.00078 which translate to an RMSE score of 1.34057.

In the next few sections, we will further explore the impact of deeper nets with respect to improving our validation results and prediction accuracy.

9. An Even Deeper Net?

A deeper net in section 8 seems to work. What if make it even deeper? We then added 1 more convolution layer to each convolution/pooling/dropout layer, and also change the epoch back to 3000 to save runtime. The result will be evaluated against the val_loss of previous model at epoch 3000.

In [ ]:
```
import theano

## Changes are marked as "##" below
net10 = NeuralNet(
    layers=[
        ('input', layers.InputLayer),
        ('conv1_1', layers.Conv2DLayer),
        ('conv1_2', layers.Conv2DLayer),
        ('conv1_3', layers.Conv2DLayer), ##
        ('pool1', layers.MaxPool2DLayer),
        ('dropout1', layers.DropoutLayer),
        ('conv2_1', layers.Conv2DLayer),
        ('conv2_2', layers.Conv2DLayer),
        ('conv2_3', layers.Conv2DLayer), ##
        ('pool2', layers.MaxPool2DLayer),
        ('dropout2', layers.DropoutLayer),
        ('conv3_1', layers.Conv2DLayer),
        ('conv3_2', layers.Conv2DLayer),
        ('conv3_3', layers.Conv2DLayer), ##
        ('pool3', layers.MaxPool2DLayer),
        ('dropout3', layers.DropoutLayer),
        ('hidden4', layers.DenseLayer),
        ('dropout4', layers.DropoutLayer),
        ('hidden5', layers.DenseLayer),
        ('dropout5', layers.DropoutLayer),
        ('hidden6', layers.DenseLayer),
        ('output', layers.DenseLayer),
        ],
    input_shape=(None, 1, 96, 96),
    conv1_1_num_filters=32, conv1_1_filter_size=(3, 3), conv1_1_pad=1,
    conv1_2_num_filters=32, conv1_2_filter_size=(3, 3), conv1_2_pad=1,
    conv1_3_num_filters=32, conv1_3_filter_size=(3, 3), conv1_3_pad=1, ##
    pool1_pool_size=(2, 2),
    dropout1_p=0.1,
    conv2_1_num_filters=64, conv2_1_filter_size=(3, 3), conv2_1_pad=1,
    conv2_2_num_filters=64, conv2_2_filter_size=(3, 3), conv2_2_pad=1,
    conv2_3_num_filters=64, conv2_3_filter_size=(3, 3), conv2_3_pad=1, ##
    pool2_pool_size=(2, 2),
    dropout2_p=0.2,
    conv3_1_num_filters=128, conv3_1_filter_size=(3, 3), conv3_1_pad=1,
    conv3_2_num_filters=128, conv3_2_filter_size=(3, 3), conv3_2_pad=1,
    conv3_3_num_filters=128, conv3_3_filter_size=(3, 3), conv3_3_pad=1, ##
    pool3_pool_size=(2, 2),
    dropout3_p=0.3,
    hidden4_num_units=1000,
    dropout4_p=0.5,
    hidden5_num_units=1000,
    dropout5_p=0.5,
    hidden6_num_units=1000,
    output_num_units=30, output_nonlinearity=None,

    update_learning_rate=theano.shared(float32(0.03)),
```

```
    update_momentum=theano.shared(float32(0.9)),

    regression=True,
    batch_iterator_train=FlipBatchIterator(batch_size=128),
    on_epoch_finished=[
        AdjustVariable('update_learning_rate', start=0.03, stop=0.0001),
        AdjustVariable('update_momentum', start=0.9, stop=0.999),
        ],
    max_epochs=3000, ## changed it back to 3000 to save time
    verbose=1,
    )

sys.setrecursionlimit(10000)

X, y = load2d()
net10.fit(X, y)

with open('net10.pickle', 'wb') as f:
    pickle.dump(net10, f, -1)
```

```
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
dtype: int64
left_eye_center_x            2140
left_eye_center_y            2140
right_eye_center_x           2140
right_eye_center_y           2140
left_eye_inner_corner_x      2140
```

```
left_eye_inner_corner_y      2140
left_eye_outer_corner_x      2140
left_eye_outer_corner_y      2140
right_eye_inner_corner_x     2140
right_eye_inner_corner_y     2140
right_eye_outer_corner_x     2140
right_eye_outer_corner_y     2140
left_eyebrow_inner_end_x     2140
left_eyebrow_inner_end_y     2140
left_eyebrow_outer_end_x     2140
left_eyebrow_outer_end_y     2140
right_eyebrow_inner_end_x    2140
right_eyebrow_inner_end_y    2140
right_eyebrow_outer_end_x    2140
right_eyebrow_outer_end_y    2140
nose_tip_x                   2140
nose_tip_y                   2140
mouth_left_corner_x          2140
mouth_left_corner_y          2140
mouth_right_corner_x         2140
mouth_right_corner_y         2140
mouth_center_top_lip_x       2140
mouth_center_top_lip_y       2140
mouth_center_bottom_lip_x    2140
mouth_center_bottom_lip_y    2140
Image                        2140
dtype: int64
# Neural Network with 20945222 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1_1   32x96x96
  2  conv1_2   32x96x96
  3  conv1_3   32x96x96
  4  pool1     32x48x48
  5  dropout1  32x48x48
  6  conv2_1   64x48x48
  7  conv2_2   64x48x48
  8  conv2_3   64x48x48
  9  pool2     64x24x24
 10  dropout2  64x24x24
 11  conv3_1   128x24x24
 12  conv3_2   128x24x24
 13  conv3_3   128x24x24
 14  pool3     128x12x12
 15  dropout3  128x12x12
 16  hidden4   1000
 17  dropout4  1000
 18  hidden5   1000
 19  dropout5  1000
 20  hidden6   1000
 21  output    30

  epoch    trn loss    val loss    trn/val  dur
```

-------	----------	----------	---------	-----
1	0.12822	0.07868	1.62971	4.91s
2	0.04545	0.00745	6.10120	4.86s
3	0.00863	0.00457	1.89020	4.86s
4	0.00552	0.00445	1.24074	4.86s
5	0.00512	0.00427	1.19940	4.89s
6	0.00494	0.00426	1.16149	4.90s
7	0.00488	0.00426	1.14440	4.91s
8	0.00481	0.00426	1.12815	4.93s
9	0.00478	0.00426	1.12356	4.95s
10	0.00476	0.00426	1.11780	4.96s
11	0.00477	0.00425	1.12122	4.95s
12	0.00472	0.00425	1.11171	4.96s
13	0.00470	0.00424	1.10648	5.50s
14	0.00469	0.00425	1.10376	5.71s
15	0.00469	0.00425	1.10359	5.60s
16	0.00467	0.00425	1.09944	5.51s
17	0.00465	0.00424	1.09490	5.50s
18	0.00463	0.00424	1.09096	5.62s
19	0.00465	0.00424	1.09599	5.61s
20	0.00464	0.00424	1.09406	5.64s
21	0.00464	0.00424	1.09265	5.65s
22	0.00462	0.00425	1.08775	5.73s
23	0.00463	0.00424	1.09067	5.73s
24	0.00463	0.00424	1.09229	5.75s
25	0.00462	0.00424	1.08979	5.73s
26	0.00462	0.00424	1.08967	5.73s
27	0.00462	0.00425	1.08737	5.74s
28	0.00460	0.00425	1.08267	5.72s
29	0.00460	0.00424	1.08439	5.72s
30	0.00460	0.00425	1.08223	5.73s
31	0.00460	0.00425	1.08177	5.72s
32	0.00459	0.00424	1.08266	5.72s
33	0.00458	0.00424	1.07830	5.72s
34	0.00458	0.00425	1.07797	5.72s
35	0.00459	0.00424	1.08157	5.73s
36	0.00458	0.00424	1.07899	5.73s
37	0.00460	0.00424	1.08366	5.73s
38	0.00460	0.00424	1.08453	5.73s
39	0.00456	0.00424	1.07561	5.73s
40	0.00458	0.00424	1.07886	5.72s
41	0.00457	0.00425	1.07515	5.72s
42	0.00456	0.00425	1.07390	5.72s
43	0.00458	0.00425	1.07775	5.72s
44	0.00457	0.00425	1.07530	5.72s
45	0.00455	0.00425	1.07098	5.72s
46	0.00455	0.00425	1.07104	5.73s
47	0.00457	0.00424	1.07667	5.72s
48	0.00456	0.00424	1.07584	5.74s
49	0.00455	0.00424	1.07266	5.73s
50	0.00456	0.00424	1.07536	5.72s
51	0.00456	0.00424	1.07537	5.72s
52	0.00456	0.00424	1.07522	5.72s
53	0.00454	0.00424	1.07027	5.72s
54	0.00456	0.00424	1.07563	5.72s
55	0.00456	0.00424	1.07534	5.73s
56	0.00454	0.00425	1.06803	5.72s

57	0.00454	0.00425	1.06938	5.73s
58	0.00455	0.00424	1.07382	5.72s
59	0.00453	0.00424	1.06914	5.72s
60	0.00454	0.00424	1.07097	5.73s
61	0.00453	0.00424	1.06837	5.72s
62	0.00454	0.00424	1.07040	5.70s
63	0.00454	0.00424	1.07114	5.69s
64	0.00453	0.00424	1.06934	5.70s
65	0.00454	0.00424	1.07048	5.71s
66	0.00455	0.00424	1.07314	5.70s
67	0.00453	0.00424	1.06803	5.70s
68	0.00453	0.00424	1.06971	5.70s
69	0.00454	0.00424	1.07064	5.70s
70	0.00453	0.00424	1.06720	5.70s
71	0.00453	0.00424	1.06672	5.69s
72	0.00452	0.00424	1.06647	5.74s
73	0.00452	0.00424	1.06629	5.72s
74	0.00453	0.00424	1.06761	5.72s
75	0.00451	0.00424	1.06467	5.73s
76	0.00452	0.00424	1.06625	5.69s
77	0.00452	0.00424	1.06789	5.70s
78	0.00454	0.00424	1.07022	5.69s
79	0.00451	0.00425	1.06350	5.70s
80	0.00451	0.00425	1.06247	5.70s
81	0.00452	0.00424	1.06771	5.70s
82	0.00451	0.00423	1.06581	5.70s
83	0.00451	0.00424	1.06323	5.70s
84	0.00451	0.00424	1.06305	5.73s
85	0.00451	0.00424	1.06247	5.70s
86	0.00451	0.00424	1.06427	5.69s
87	0.00451	0.00424	1.06321	5.72s
88	0.00451	0.00424	1.06173	5.70s
89	0.00450	0.00424	1.06118	5.69s
90	0.00450	0.00424	1.06185	5.69s
91	0.00451	0.00424	1.06205	5.69s
92	0.00451	0.00425	1.06319	5.69s
93	0.00451	0.00424	1.06256	5.71s
94	0.00449	0.00424	1.05889	5.75s
95	0.00449	0.00424	1.05985	5.73s
96	0.00451	0.00424	1.06308	5.71s
97	0.00449	0.00424	1.05939	5.69s
98	0.00450	0.00424	1.05954	5.69s
99	0.00450	0.00424	1.06156	5.69s
100	0.00450	0.00424	1.06161	5.74s
101	0.00450	0.00423	1.06192	5.71s
102	0.00450	0.00424	1.06192	5.69s
103	0.00450	0.00424	1.06192	5.69s
104	0.00450	0.00424	1.06275	5.69s
105	0.00450	0.00424	1.06153	5.69s
106	0.00449	0.00424	1.05976	5.71s
107	0.00449	0.00424	1.06094	5.69s
108	0.00449	0.00424	1.06055	5.69s
109	0.00449	0.00424	1.05983	5.69s
110	0.00449	0.00424	1.05976	5.69s
111	0.00448	0.00424	1.05784	5.70s
112	0.00449	0.00424	1.06069	5.69s
113	0.00449	0.00423	1.06153	5.69s

114	0.00449	0.00424	1.06008	5.71s
115	0.00449	0.00424	1.06046	5.73s
116	0.00449	0.00423	1.06130	5.72s
117	0.00450	0.00424	1.06060	5.70s
118	0.00449	0.00424	1.06027	5.69s
119	0.00449	0.00424	1.05936	5.69s
120	0.00450	0.00424	1.06039	5.69s
121	0.00449	0.00424	1.05956	5.69s
122	0.00449	0.00424	1.05795	5.69s
123	0.00449	0.00424	1.05897	5.69s
124	0.00449	0.00424	1.05968	5.69s
125	0.00449	0.00424	1.05924	5.69s
126	0.00449	0.00424	1.05904	5.69s
127	0.00448	0.00424	1.05774	5.70s
128	0.00448	0.00424	1.05738	5.69s
129	0.00449	0.00424	1.05961	5.69s
130	0.00448	0.00424	1.05687	5.69s
131	0.00449	0.00424	1.05892	5.69s
132	0.00449	0.00424	1.05850	5.69s
133	0.00448	0.00424	1.05720	5.69s
134	0.00449	0.00424	1.05906	5.69s
135	0.00449	0.00424	1.06057	5.69s
136	0.00448	0.00424	1.05749	5.69s
137	0.00448	0.00423	1.05861	5.69s
138	0.00449	0.00423	1.05964	5.69s
139	0.00449	0.00424	1.05877	5.69s
140	0.00447	0.00424	1.05413	5.69s
141	0.00448	0.00424	1.05638	5.69s
142	0.00448	0.00424	1.05671	5.69s
143	0.00448	0.00424	1.05724	5.70s
144	0.00448	0.00423	1.05832	5.69s
145	0.00448	0.00424	1.05747	5.69s
146	0.00448	0.00424	1.05743	5.68s
147	0.00447	0.00424	1.05634	5.70s
148	0.00448	0.00423	1.05692	5.69s
149	0.00448	0.00424	1.05615	5.69s
150	0.00448	0.00424	1.05652	5.68s
151	0.00447	0.00424	1.05475	5.69s
152	0.00448	0.00424	1.05730	5.69s
153	0.00447	0.00423	1.05651	5.69s
154	0.00448	0.00423	1.05871	5.69s
155	0.00448	0.00423	1.05745	5.69s
156	0.00447	0.00424	1.05544	5.69s
157	0.00448	0.00423	1.05754	5.69s
158	0.00448	0.00423	1.05740	5.69s
159	0.00447	0.00423	1.05603	5.69s
160	0.00447	0.00423	1.05542	5.69s
161	0.00447	0.00423	1.05720	5.68s
162	0.00447	0.00423	1.05527	5.68s
163	0.00448	0.00423	1.05708	5.69s
164	0.00448	0.00423	1.05705	5.69s
165	0.00448	0.00423	1.05736	5.69s
166	0.00448	0.00424	1.05665	5.69s
167	0.00447	0.00424	1.05523	5.70s
168	0.00448	0.00423	1.05701	5.69s
169	0.00447	0.00423	1.05648	5.69s
170	0.00447	0.00423	1.05465	5.69s

```
171     0.00447     0.00423    1.05586  5.69s
172     0.00447     0.00423    1.05693  5.69s
173     0.00447     0.00423    1.05607  5.69s
174     0.00447     0.00423    1.05599  5.69s
175     0.00447     0.00423    1.05606  5.69s
176     0.00447     0.00424    1.05499  5.69s
177     0.00446     0.00423    1.05475  5.69s
178     0.00446     0.00424    1.05372  5.69s
179     0.00447     0.00423    1.05625  5.69s
180     0.00447     0.00423    1.05607  5.69s
181     0.00447     0.00423    1.05559  5.69s
```

Also, since we questioned whether an additional hidden layer does any help, we created net11 with only 2 hidden layer but everything else the same as net 10 just to better understand the impact of the number of hidden layers.

In [ ]:

```
## Changes are marked as "##" below
import theano

net11 = NeuralNet(
    layers=[
        ('input', layers.InputLayer),
        ('conv1_1', layers.Conv2DLayer),
        ('conv1_2', layers.Conv2DLayer),
        ('conv1_3', layers.Conv2DLayer),
        ('pool1', layers.MaxPool2DLayer),
        ('dropout1', layers.DropoutLayer),
        ('conv2_1', layers.Conv2DLayer),
        ('conv2_2', layers.Conv2DLayer),
        ('conv2_3', layers.Conv2DLayer),
        ('pool2', layers.MaxPool2DLayer),
        ('dropout2', layers.DropoutLayer),
        ('conv3_1', layers.Conv2DLayer),
        ('conv3_2', layers.Conv2DLayer),
        ('conv3_3', layers.Conv2DLayer),
        ('pool3', layers.MaxPool2DLayer),
        ('dropout3', layers.DropoutLayer),
        ('hidden4', layers.DenseLayer),
        ('dropout4', layers.DropoutLayer),
        ('hidden5', layers.DenseLayer),
       ##
       ##
        ('output', layers.DenseLayer),
        ],
    input_shape=(None, 1, 96, 96),
    conv1_1_num_filters=32, conv1_1_filter_size=(3, 3), conv1_1_pad=1,
    conv1_2_num_filters=32, conv1_2_filter_size=(3, 3), conv1_2_pad=1,
    conv1_3_num_filters=32, conv1_3_filter_size=(3, 3), conv1_3_pad=1,
    pool1_pool_size=(2, 2),
    dropout1_p=0.1,
    conv2_1_num_filters=64, conv2_1_filter_size=(3, 3), conv2_1_pad=1,
    conv2_2_num_filters=64, conv2_2_filter_size=(3, 3), conv2_2_pad=1,
    conv2_3_num_filters=64, conv2_3_filter_size=(3, 3), conv2_3_pad=1,
    pool2_pool_size=(2, 2),
    dropout2_p=0.2,
```

```
    conv3_1_num_filters=128, conv3_1_filter_size=(3, 3), conv3_1_pad=1,
    conv3_2_num_filters=128, conv3_2_filter_size=(3, 3), conv3_2_pad=1,
    conv3_3_num_filters=128, conv3_3_filter_size=(3, 3), conv3_3_pad=1,
    pool3_pool_size=(2, 2),
    dropout3_p=0.3,
    hidden4_num_units=1000,
    dropout4_p=0.5,
    hidden5_num_units=1000,
   ##
    output_num_units=30, output_nonlinearity=None,

    update_learning_rate=theano.shared(float32(0.03)),
    update_momentum=theano.shared(float32(0.9)),

    regression=True,
    batch_iterator_train=FlipBatchIterator(batch_size=128),
    on_epoch_finished=[
        AdjustVariable('update_learning_rate', start=0.03, stop=0.0001),
        AdjustVariable('update_momentum', start=0.9, stop=0.999),
        ],
    max_epochs=3000,
    verbose=1,
    )

sys.setrecursionlimit(10000)

X, y = load2d()
net11.fit(X, y)

with open('net11.pickle', 'wb') as f:
    pickle.dump(net11, f, -1)
```

```
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
dtype: int64
```

```
/usr/lib/python2.7/site-packages/lasagne/layers/conv.py:489: UserWarning: T
he `image_shape` keyword argument to `tensor.nnet.conv2d` is deprecated, it
has been renamed to `input_shape`.
  border_mode=border_mode)
```

```
# Neural Network with 19944222 learnable parameters

## Layer information

  #  name      size
---  --------  ---------
  0  input     1x96x96
  1  conv1_1   32x96x96
  2  conv1_2   32x96x96
  3  conv1_3   32x96x96
  4  pool1     32x48x48
  5  dropout1  32x48x48
  6  conv2_1   64x48x48
  7  conv2_2   64x48x48
  8  conv2_3   64x48x48
  9  pool2     64x24x24
 10  dropout2  64x24x24
 11  conv3_1   128x24x24
 12  conv3_2   128x24x24
 13  conv3_3   128x24x24
 14  pool3     128x12x12
 15  dropout3  128x12x12
 16  hidden4   1000
 17  dropout4  1000
 18  hidden5   1000
 19  output    30

  epoch    trn loss    val loss    trn/val  dur
-------  ----------  ----------  ---------  -----
      1     0.11286     0.01762    6.40408  4.95s
      2     0.01458     0.00711    2.05017  4.92s
      3     0.00791     0.00464    1.70506  4.93s
      4     0.00673     0.00460    1.46073  5.42s
      5     0.00641     0.00461    1.39016  5.70s
      6     0.00608     0.00453    1.34213  5.70s
      7     0.00595     0.00451    1.32068  5.70s
      8     0.00585     0.00449    1.30253  5.70s
      9     0.00569     0.00448    1.27127  5.71s
     10     0.00559     0.00449    1.24424  5.70s
     11     0.00551     0.00441    1.24873  5.70s
     12     0.00543     0.00442    1.22862  5.69s
     13     0.00534     0.00441    1.21035  5.70s
     14     0.00533     0.00440    1.21276  5.70s
     15     0.00527     0.00440    1.19828  5.71s
     16     0.00519     0.00437    1.18725  5.69s
```

17	0.00518	0.00437	1.18592	5.70s
18	0.00517	0.00437	1.18376	5.70s
19	0.00510	0.00435	1.17096	5.69s
20	0.00505	0.00433	1.16630	5.70s
21	0.00505	0.00436	1.15856	5.70s
22	0.00502	0.00432	1.16230	5.71s
23	0.00498	0.00433	1.15011	5.70s
24	0.00494	0.00434	1.13961	5.70s
25	0.00494	0.00431	1.14601	5.69s
26	0.00489	0.00431	1.13312	5.70s
27	0.00489	0.00430	1.13841	5.69s
28	0.00486	0.00430	1.13097	5.69s
29	0.00482	0.00429	1.12372	5.70s
30	0.00486	0.00429	1.13358	5.69s
31	0.00481	0.00428	1.12230	5.69s
32	0.00479	0.00429	1.11684	5.69s
33	0.00479	0.00428	1.11874	5.70s
34	0.00478	0.00426	1.11968	5.70s
35	0.00478	0.00427	1.11790	5.70s
36	0.00473	0.00427	1.10625	5.69s
37	0.00476	0.00427	1.11551	5.70s
38	0.00473	0.00427	1.10610	5.70s
39	0.00472	0.00428	1.10364	5.70s
40	0.00472	0.00426	1.10792	5.70s
41	0.00474	0.00426	1.11231	5.70s
42	0.00471	0.00427	1.10457	5.70s
43	0.00467	0.00425	1.09842	5.69s
44	0.00468	0.00425	1.10117	5.69s
45	0.00463	0.00425	1.08917	5.69s
46	0.00467	0.00426	1.09581	5.70s
47	0.00462	0.00426	1.08595	5.70s
48	0.00466	0.00425	1.09624	5.69s
49	0.00464	0.00425	1.09285	5.70s
50	0.00464	0.00425	1.09290	5.69s
51	0.00461	0.00423	1.08962	5.70s
52	0.00464	0.00424	1.09397	5.71s
53	0.00462	0.00424	1.08924	5.70s
54	0.00461	0.00424	1.08822	5.70s
55	0.00459	0.00424	1.08412	5.70s
56	0.00461	0.00424	1.08901	5.71s
57	0.00460	0.00424	1.08342	5.70s
58	0.00459	0.00424	1.08188	5.70s
59	0.00456	0.00424	1.07629	5.69s
60	0.00458	0.00424	1.08110	5.69s
61	0.00457	0.00424	1.07963	5.69s
62	0.00460	0.00423	1.08862	5.70s
63	0.00457	0.00423	1.08106	5.70s
64	0.00456	0.00423	1.07708	5.70s
65	0.00456	0.00423	1.07784	5.70s
66	0.00455	0.00423	1.07528	5.70s
67	0.00456	0.00423	1.07822	5.70s
68	0.00454	0.00423	1.07372	5.70s
69	0.00455	0.00423	1.07568	5.70s
70	0.00455	0.00423	1.07547	5.70s
71	0.00453	0.00423	1.07133	5.70s
72	0.00454	0.00423	1.07449	5.71s
73	0.00454	0.00423	1.07356	5.70s

74	0.00453	0.00423	1.07248	5.70s
75	0.00455	0.00422	1.07708	5.70s
76	0.00452	0.00422	1.07069	5.70s
77	0.00452	0.00423	1.06898	5.69s
78	0.00453	0.00422	1.07132	5.70s
79	0.00451	0.00423	1.06763	5.70s
80	0.00451	0.00423	1.06758	5.69s
81	0.00451	0.00423	1.06785	5.70s
82	0.00451	0.00422	1.06850	5.69s
83	0.00452	0.00422	1.07087	5.70s
84	0.00450	0.00422	1.06672	5.70s
85	0.00451	0.00422	1.06940	5.70s
86	0.00451	0.00422	1.06836	5.69s
87	0.00449	0.00422	1.06531	5.70s
88	0.00450	0.00422	1.06721	5.70s
89	0.00449	0.00422	1.06627	5.70s
90	0.00449	0.00421	1.06438	5.70s
91	0.00450	0.00422	1.06685	5.70s
92	0.00450	0.00422	1.06537	5.70s
93	0.00449	0.00422	1.06491	5.70s
94	0.00449	0.00422	1.06531	5.70s
95	0.00449	0.00421	1.06496	5.70s
96	0.00449	0.00421	1.06472	5.70s
97	0.00449	0.00421	1.06501	5.70s
98	0.00448	0.00421	1.06396	5.73s
99	0.00447	0.00421	1.06275	5.70s
100	0.00447	0.00421	1.06049	5.70s
101	0.00447	0.00421	1.06013	5.70s
102	0.00447	0.00421	1.06258	5.70s
103	0.00447	0.00421	1.06180	5.69s
104	0.00447	0.00421	1.06183	5.70s
105	0.00447	0.00421	1.06248	5.70s
106	0.00447	0.00421	1.06198	5.70s
107	0.00447	0.00421	1.06115	5.70s
108	0.00448	0.00421	1.06320	5.70s
109	0.00445	0.00421	1.05757	5.69s
110	0.00447	0.00421	1.06247	5.70s
111	0.00446	0.00421	1.06030	5.69s
112	0.00447	0.00420	1.06324	5.70s
113	0.00447	0.00420	1.06208	5.70s
114	0.00447	0.00420	1.06345	5.70s
115	0.00446	0.00420	1.06117	5.70s
116	0.00445	0.00420	1.05992	5.70s
117	0.00445	0.00420	1.06072	5.70s
118	0.00446	0.00420	1.06248	5.69s
119	0.00445	0.00420	1.05960	5.70s
120	0.00446	0.00420	1.06290	5.70s
121	0.00446	0.00420	1.06195	5.69s
122	0.00444	0.00420	1.05686	5.70s
123	0.00446	0.00420	1.06220	5.69s
124	0.00445	0.00420	1.05974	5.70s
125	0.00445	0.00420	1.05912	5.70s
126	0.00444	0.00420	1.05856	5.69s
127	0.00445	0.00419	1.06007	5.69s
128	0.00443	0.00419	1.05752	5.69s
129	0.00446	0.00420	1.06180	5.70s
130	0.00445	0.00419	1.06094	5.69s

131	0.00445	0.00419	1.06081	5.69s
132	0.00444	0.00420	1.05821	5.70s
133	0.00445	0.00419	1.06217	5.70s
134	0.00444	0.00419	1.05968	5.69s
135	0.00444	0.00419	1.05895	5.69s
136	0.00443	0.00419	1.05879	5.70s
137	0.00444	0.00419	1.05976	5.70s
138	0.00443	0.00418	1.05917	5.70s
139	0.00444	0.00419	1.06046	5.71s
140	0.00444	0.00419	1.05935	5.70s
141	0.00443	0.00419	1.05915	5.70s
142	0.00442	0.00418	1.05683	5.70s
143	0.00443	0.00418	1.05918	5.69s
144	0.00443	0.00418	1.05949	5.69s
145	0.00442	0.00418	1.05898	5.70s
146	0.00443	0.00417	1.06239	5.70s
147	0.00444	0.00418	1.06255	5.70s
148	0.00443	0.00418	1.06007	5.69s
149	0.00441	0.00417	1.05669	5.69s
150	0.00442	0.00417	1.06014	5.69s
151	0.00442	0.00417	1.06032	5.69s
152	0.00442	0.00417	1.06010	5.70s
153	0.00442	0.00417	1.06049	5.69s
154	0.00442	0.00417	1.06144	5.69s
155	0.00441	0.00416	1.05837	5.70s
156	0.00441	0.00416	1.05906	5.70s
157	0.00441	0.00416	1.06051	5.70s
158	0.00441	0.00416	1.05822	5.70s
159	0.00440	0.00416	1.05746	5.70s
160	0.00440	0.00416	1.05953	5.70s
161	0.00439	0.00415	1.05685	5.70s
162	0.00439	0.00415	1.05756	5.70s
163	0.00439	0.00415	1.05766	5.70s
164	0.00440	0.00415	1.06097	5.69s
165	0.00439	0.00415	1.05823	5.69s
166	0.00439	0.00415	1.05777	5.69s
167	0.00439	0.00415	1.05812	5.69s
168	0.00439	0.00415	1.05780	5.70s
169	0.00439	0.00414	1.05978	5.70s
170	0.00439	0.00414	1.05964	5.69s
171	0.00439	0.00414	1.05896	5.70s
172	0.00439	0.00414	1.06068	5.70s
173	0.00439	0.00414	1.05986	5.69s
174	0.00438	0.00414	1.05936	5.70s
175	0.00438	0.00414	1.05908	5.70s
176	0.00437	0.00413	1.05820	5.69s
177	0.00437	0.00413	1.05769	5.70s
178	0.00438	0.00413	1.06033	5.70s
179	0.00437	0.00413	1.05785	5.71s
180	0.00437	0.00412	1.06121	5.70s
181	0.00437	0.00412	1.06140	5.70s
182	0.00437	0.00412	1.06103	5.70s
183	0.00438	0.00412	1.06296	5.70s
184	0.00437	0.00412	1.06033	5.70s
185	0.00436	0.00412	1.05838	5.70s
186	0.00436	0.00412	1.05935	5.70s
187	0.00436	0.00411	1.05929	5.69s

188	0.00436	0.00411	1.05957	5.70s
189	0.00436	0.00411	1.06011	5.70s
190	0.00436	0.00411	1.06163	5.69s
191	0.00435	0.00411	1.05897	5.70s
192	0.00434	0.00410	1.05875	5.70s
193	0.00435	0.00410	1.05993	5.70s
194	0.00435	0.00410	1.05995	5.70s
195	0.00435	0.00410	1.06123	5.70s
196	0.00434	0.00410	1.05991	5.70s
197	0.00434	0.00409	1.06052	5.70s
198	0.00434	0.00409	1.06183	5.70s
199	0.00434	0.00409	1.06159	5.70s
200	0.00434	0.00409	1.06175	5.70s
201	0.00432	0.00408	1.05847	5.70s
202	0.00433	0.00408	1.06115	5.70s
203	0.00433	0.00408	1.06141	5.70s
204	0.00432	0.00408	1.05970	5.70s
205	0.00432	0.00408	1.05933	5.70s
206	0.00433	0.00408	1.06192	5.69s
207	0.00432	0.00407	1.06097	5.70s
208	0.00432	0.00407	1.06002	5.70s
209	0.00431	0.00407	1.05956	5.70s
210	0.00431	0.00407	1.05964	5.69s
211	0.00432	0.00407	1.06028	5.70s
212	0.00432	0.00407	1.06051	5.70s
213	0.00431	0.00407	1.06024	5.70s
214	0.00431	0.00406	1.06015	5.70s
215	0.00431	0.00407	1.05885	5.70s
216	0.00431	0.00406	1.06086	5.70s
217	0.00430	0.00406	1.05889	5.69s
218	0.00429	0.00406	1.05739	5.70s
219	0.00431	0.00405	1.06286	5.69s
220	0.00430	0.00405	1.06184	5.70s
221	0.00429	0.00405	1.05990	5.70s
222	0.00429	0.00405	1.06069	5.70s
223	0.00429	0.00405	1.05996	5.69s
224	0.00429	0.00404	1.06144	5.70s
225	0.00429	0.00404	1.06052	5.70s
226	0.00428	0.00404	1.05931	5.70s
227	0.00428	0.00404	1.06050	5.70s
228	0.00427	0.00403	1.05956	5.70s
229	0.00428	0.00403	1.06206	5.70s
230	0.00428	0.00403	1.06205	5.70s
231	0.00428	0.00403	1.06217	5.69s
232	0.00427	0.00403	1.06092	5.70s
233	0.00427	0.00403	1.05997	5.70s
234	0.00427	0.00402	1.06129	5.70s
235	0.00427	0.00402	1.06013	5.70s
236	0.00426	0.00402	1.05964	5.70s
237	0.00427	0.00402	1.06046	5.70s
238	0.00427	0.00402	1.06464	5.69s
239	0.00426	0.00402	1.06102	5.70s
240	0.00426	0.00401	1.06040	5.70s
241	0.00426	0.00402	1.06159	5.70s
242	0.00425	0.00401	1.06158	5.70s
243	0.00425	0.00401	1.05894	5.70s
244	0.00425	0.00401	1.06055	5.70s

245	0.00425	0.00401	1.05978	5.70s
246	0.00424	0.00400	1.06010	5.70s
247	0.00425	0.00400	1.06336	5.71s
248	0.00425	0.00400	1.06298	5.69s
249	0.00425	0.00399	1.06419	5.70s
250	0.00423	0.00399	1.06160	5.69s
251	0.00425	0.00399	1.06438	5.71s
252	0.00423	0.00399	1.06012	5.69s
253	0.00424	0.00399	1.06078	5.70s
254	0.00424	0.00399	1.06179	5.69s
255	0.00423	0.00399	1.06002	5.70s
256	0.00421	0.00398	1.05831	5.70s
257	0.00422	0.00398	1.06072	5.69s
258	0.00421	0.00398	1.05979	5.69s
259	0.00422	0.00397	1.06234	5.70s
260	0.00421	0.00397	1.06041	5.70s
261	0.00421	0.00397	1.05973	5.70s
262	0.00421	0.00398	1.05852	5.70s
263	0.00421	0.00397	1.05969	5.70s
264	0.00421	0.00397	1.06004	5.69s
265	0.00421	0.00397	1.05937	5.69s
266	0.00421	0.00396	1.06125	5.69s
267	0.00419	0.00397	1.05714	5.69s
268	0.00420	0.00397	1.05902	5.69s
269	0.00421	0.00396	1.06267	5.69s
270	0.00420	0.00395	1.06317	5.69s
271	0.00419	0.00395	1.06132	5.69s
272	0.00419	0.00395	1.06058	5.69s
273	0.00419	0.00396	1.05861	5.69s
274	0.00419	0.00395	1.06013	5.70s
275	0.00419	0.00394	1.06259	5.70s
276	0.00418	0.00395	1.06024	5.69s
277	0.00418	0.00394	1.06106	5.69s
278	0.00417	0.00394	1.05997	5.70s
279	0.00417	0.00394	1.05921	5.70s
280	0.00419	0.00394	1.06329	5.69s
281	0.00418	0.00393	1.06400	5.69s
282	0.00418	0.00393	1.06320	5.69s
283	0.00416	0.00394	1.05731	5.69s
284	0.00418	0.00394	1.06178	5.69s
285	0.00418	0.00393	1.06448	5.69s
286	0.00417	0.00392	1.06298	5.69s
287	0.00416	0.00392	1.06205	5.70s
288	0.00416	0.00392	1.06170	5.70s
289	0.00415	0.00392	1.05995	5.70s
290	0.00415	0.00392	1.05667	5.70s
291	0.00415	0.00392	1.05895	5.70s
292	0.00415	0.00392	1.05915	5.70s
293	0.00415	0.00391	1.06034	5.70s
294	0.00415	0.00391	1.06181	5.70s
295	0.00415	0.00392	1.06038	5.70s
296	0.00414	0.00390	1.06058	5.70s
297	0.00414	0.00390	1.06228	5.70s
298	0.00414	0.00390	1.06193	5.70s
299	0.00414	0.00389	1.06223	5.70s
300	0.00414	0.00389	1.06399	5.70s
301	0.00412	0.00389	1.05856	5.69s

302	0.00412	0.00389	1.05880	5.73s
303	0.00412	0.00389	1.06082	5.69s
304	0.00412	0.00388	1.06159	5.69s
305	0.00411	0.00388	1.05923	5.70s
306	0.00412	0.00388	1.06174	5.69s
307	0.00412	0.00388	1.06025	5.69s
308	0.00411	0.00387	1.06058	5.69s
309	0.00412	0.00388	1.06373	5.69s
310	0.00410	0.00387	1.05936	5.70s
311	0.00411	0.00388	1.06041	5.70s
312	0.00411	0.00387	1.06100	5.69s
313	0.00411	0.00386	1.06388	5.69s
314	0.00409	0.00386	1.05827	5.70s
315	0.00410	0.00386	1.06314	5.69s
316	0.00410	0.00385	1.06513	5.69s
317	0.00409	0.00386	1.06104	5.71s
318	0.00409	0.00385	1.06167	5.69s
319	0.00408	0.00385	1.05962	5.69s
320	0.00409	0.00386	1.05914	5.69s
321	0.00408	0.00385	1.06141	5.70s
322	0.00408	0.00384	1.06323	5.69s
323	0.00408	0.00384	1.06168	5.70s
324	0.00408	0.00385	1.06010	5.70s
325	0.00407	0.00384	1.06027	5.69s
326	0.00407	0.00383	1.06296	5.69s
327	0.00408	0.00384	1.06312	5.70s
328	0.00406	0.00383	1.06158	5.70s
329	0.00406	0.00384	1.05961	5.70s
330	0.00406	0.00383	1.06101	5.70s
331	0.00407	0.00382	1.06417	5.70s
332	0.00405	0.00382	1.05877	5.70s
333	0.00405	0.00382	1.06096	5.69s
334	0.00406	0.00381	1.06508	5.69s
335	0.00404	0.00381	1.06217	5.69s
336	0.00404	0.00381	1.06079	5.69s
337	0.00405	0.00381	1.06406	5.69s
338	0.00405	0.00380	1.06662	5.70s
339	0.00405	0.00380	1.06637	5.69s
340	0.00404	0.00380	1.06256	5.69s
341	0.00404	0.00379	1.06477	5.70s
342	0.00404	0.00380	1.06395	5.70s
343	0.00402	0.00379	1.06212	5.70s
344	0.00402	0.00378	1.06369	5.70s
345	0.00402	0.00378	1.06399	5.69s
346	0.00400	0.00378	1.05884	5.69s
347	0.00402	0.00379	1.06142	5.70s
348	0.00400	0.00378	1.06038	5.69s
349	0.00401	0.00377	1.06363	5.69s
350	0.00401	0.00377	1.06315	5.69s
351	0.00401	0.00376	1.06458	5.69s
352	0.00400	0.00376	1.06355	5.69s
353	0.00399	0.00376	1.05975	5.70s
354	0.00400	0.00376	1.06413	5.69s
355	0.00400	0.00376	1.06436	5.69s
356	0.00399	0.00375	1.06421	5.69s
357	0.00400	0.00375	1.06754	5.70s
358	0.00399	0.00375	1.06494	5.70s

359	0.00396	0.00374	1.06050	5.70s
360	0.00398	0.00374	1.06219	5.70s
361	0.00397	0.00373	1.06416	5.70s
362	0.00397	0.00372	1.06811	5.70s
363	0.00396	0.00372	1.06423	5.69s
364	0.00397	0.00373	1.06592	5.70s
365	0.00397	0.00372	1.06890	5.69s
366	0.00396	0.00371	1.06748	5.70s
367	0.00394	0.00371	1.06195	5.70s
368	0.00394	0.00371	1.06369	5.70s
369	0.00394	0.00370	1.06381	5.70s
370	0.00396	0.00370	1.07069	5.69s
371	0.00395	0.00370	1.06856	5.70s
372	0.00392	0.00368	1.06493	5.69s
373	0.00393	0.00368	1.06831	5.70s
374	0.00394	0.00368	1.07078	5.70s
375	0.00394	0.00367	1.07134	5.69s
376	0.00393	0.00367	1.06984	5.69s
377	0.00394	0.00367	1.07174	5.70s
378	0.00391	0.00367	1.06594	5.70s
379	0.00390	0.00366	1.06430	5.70s
380	0.00390	0.00366	1.06712	5.70s
381	0.00389	0.00366	1.06408	5.70s
382	0.00389	0.00364	1.06794	5.70s
383	0.00390	0.00365	1.06797	5.71s
384	0.00390	0.00365	1.07028	5.70s
385	0.00389	0.00363	1.07176	5.70s
386	0.00390	0.00363	1.07268	5.70s
387	0.00389	0.00364	1.06943	5.70s
388	0.00390	0.00364	1.07292	5.70s
389	0.00387	0.00363	1.06810	5.70s
390	0.00388	0.00362	1.07268	5.69s
391	0.00387	0.00362	1.06757	5.70s
392	0.00387	0.00362	1.06974	5.69s
393	0.00386	0.00362	1.06654	5.69s
394	0.00387	0.00360	1.07518	5.70s
395	0.00385	0.00360	1.07078	5.70s
396	0.00385	0.00359	1.07150	5.69s
397	0.00385	0.00359	1.07404	5.70s
398	0.00384	0.00358	1.07521	5.69s
399	0.00384	0.00357	1.07493	5.70s
400	0.00383	0.00357	1.07294	5.69s
401	0.00382	0.00356	1.07187	5.70s
402	0.00382	0.00355	1.07509	5.69s
403	0.00382	0.00355	1.07462	5.69s
404	0.00383	0.00355	1.07881	5.69s
405	0.00381	0.00354	1.07577	5.69s
406	0.00379	0.00354	1.07291	5.69s
407	0.00380	0.00354	1.07414	5.70s
408	0.00381	0.00353	1.07773	5.69s
409	0.00380	0.00353	1.07690	5.70s
410	0.00380	0.00351	1.08028	5.69s
411	0.00379	0.00352	1.07885	5.69s
412	0.00380	0.00352	1.07916	5.69s
413	0.00376	0.00350	1.07229	5.70s
414	0.00378	0.00350	1.08111	5.69s
415	0.00378	0.00350	1.07937	5.69s

416	0.00376	0.00349	1.07706	5.70s
417	0.00375	0.00349	1.07556	5.69s
418	0.00374	0.00347	1.07748	5.69s
419	0.00375	0.00347	1.08023	5.69s
420	0.00373	0.00347	1.07492	5.69s
421	0.00375	0.00347	1.07904	5.70s
422	0.00374	0.00347	1.07823	5.70s
423	0.00374	0.00346	1.08018	5.69s
424	0.00371	0.00343	1.08106	5.71s
425	0.00371	0.00344	1.08033	5.70s
426	0.00371	0.00345	1.07630	5.69s
427	0.00372	0.00343	1.08389	5.70s
428	0.00370	0.00342	1.08012	5.69s
429	0.00369	0.00341	1.08066	5.70s
430	0.00368	0.00341	1.07923	5.70s
431	0.00370	0.00341	1.08677	5.70s
432	0.00368	0.00340	1.08164	5.70s
433	0.00371	0.00339	1.09465	5.69s
434	0.00368	0.00338	1.08770	5.69s
435	0.00368	0.00338	1.08883	5.69s
436	0.00366	0.00338	1.08298	5.69s
437	0.00363	0.00338	1.07590	5.69s
438	0.00364	0.00336	1.08503	5.69s
439	0.00363	0.00337	1.08015	5.69s
440	0.00367	0.00335	1.09404	5.70s
441	0.00363	0.00334	1.08672	5.69s
442	0.00364	0.00334	1.09269	5.69s
443	0.00361	0.00333	1.08449	5.69s
444	0.00361	0.00332	1.08847	5.69s
445	0.00361	0.00330	1.09270	5.69s
446	0.00359	0.00330	1.08716	5.69s
447	0.00360	0.00329	1.09253	5.70s
448	0.00361	0.00330	1.09574	5.69s
449	0.00358	0.00328	1.09245	5.70s
450	0.00357	0.00327	1.09158	5.69s
451	0.00357	0.00327	1.08898	5.69s
452	0.00358	0.00327	1.09515	5.69s
453	0.00356	0.00326	1.09251	5.70s
454	0.00355	0.00325	1.09281	5.69s
455	0.00355	0.00324	1.09590	5.69s
456	0.00356	0.00324	1.09923	5.69s
457	0.00352	0.00322	1.09345	5.69s
458	0.00352	0.00321	1.09745	5.70s
459	0.00352	0.00320	1.09932	5.70s
460	0.00351	0.00319	1.10052	5.69s
461	0.00351	0.00319	1.10121	5.69s
462	0.00350	0.00318	1.09923	5.69s
463	0.00351	0.00318	1.10371	5.69s
464	0.00349	0.00317	1.10162	5.69s
465	0.00349	0.00317	1.10264	5.69s
466	0.00346	0.00315	1.10050	5.69s
467	0.00347	0.00315	1.10191	5.69s
468	0.00346	0.00313	1.10695	5.69s
469	0.00343	0.00313	1.09614	5.69s
470	0.00347	0.00312	1.11175	5.69s
471	0.00343	0.00311	1.10356	5.69s
472	0.00343	0.00310	1.10650	5.69s

473	0.00343	0.00308	1.11208	5.70s
474	0.00342	0.00308	1.11019	5.69s
475	0.00340	0.00307	1.10555	5.70s
476	0.00340	0.00307	1.10595	5.69s
477	0.00339	0.00306	1.10872	5.69s
478	0.00336	0.00304	1.10444	5.70s
479	0.00337	0.00303	1.10904	5.69s
480	0.00337	0.00303	1.11335	5.69s
481	0.00334	0.00301	1.11008	5.69s
482	0.00335	0.00301	1.11183	5.69s
483	0.00334	0.00300	1.11128	5.69s
484	0.00332	0.00299	1.11195	5.69s
485	0.00333	0.00298	1.11670	5.70s
486	0.00334	0.00298	1.12261	5.69s
487	0.00330	0.00296	1.11532	5.70s
488	0.00327	0.00294	1.11040	5.69s
489	0.00330	0.00294	1.12028	5.70s
490	0.00328	0.00294	1.11554	5.71s
491	0.00327	0.00292	1.12052	5.70s
492	0.00328	0.00292	1.12350	5.70s
493	0.00326	0.00292	1.11929	5.70s
494	0.00324	0.00289	1.12211	5.69s
495	0.00324	0.00288	1.12303	5.69s
496	0.00322	0.00287	1.12185	5.69s
497	0.00322	0.00286	1.12414	5.69s
498	0.00322	0.00285	1.12923	5.69s
499	0.00320	0.00285	1.12333	5.69s
500	0.00320	0.00284	1.12855	5.69s
501	0.00321	0.00283	1.13368	5.69s
502	0.00318	0.00282	1.12561	5.69s
503	0.00316	0.00279	1.12934	5.69s
504	0.00316	0.00279	1.13057	5.69s
505	0.00317	0.00278	1.13831	5.69s
506	0.00313	0.00277	1.12893	5.69s
507	0.00313	0.00275	1.13807	5.70s
508	0.00312	0.00275	1.13662	5.70s
509	0.00310	0.00274	1.12935	5.69s
510	0.00313	0.00274	1.14105	5.71s
511	0.00311	0.00272	1.14140	5.69s
512	0.00309	0.00271	1.14128	5.69s
513	0.00308	0.00271	1.13817	5.69s
514	0.00310	0.00271	1.14395	5.69s
515	0.00308	0.00269	1.14437	5.69s
516	0.00305	0.00268	1.13970	5.69s
517	0.00304	0.00266	1.14091	5.70s
518	0.00303	0.00266	1.13994	5.69s
519	0.00300	0.00266	1.13104	5.69s
520	0.00299	0.00264	1.13482	5.70s
521	0.00301	0.00263	1.14504	5.70s
522	0.00301	0.00262	1.15113	5.70s
523	0.00300	0.00261	1.14882	5.70s
524	0.00301	0.00260	1.15871	5.70s
525	0.00298	0.00259	1.14864	5.70s
526	0.00298	0.00259	1.15315	5.69s
527	0.00297	0.00257	1.15570	5.69s
528	0.00296	0.00256	1.15574	5.69s
529	0.00295	0.00256	1.14977	5.69s

530	0.00295	0.00254	1.15863	5.69s
531	0.00293	0.00254	1.15152	5.69s
532	0.00293	0.00253	1.15725	5.69s
533	0.00293	0.00252	1.16195	5.69s
534	0.00292	0.00252	1.15904	5.69s
535	0.00291	0.00252	1.15742	5.69s
536	0.00287	0.00250	1.15105	5.70s
537	0.00289	0.00249	1.16225	5.70s
538	0.00289	0.00249	1.16247	5.70s
539	0.00287	0.00247	1.16209	5.69s
540	0.00286	0.00246	1.16521	5.69s
541	0.00284	0.00245	1.15840	5.69s
542	0.00284	0.00244	1.16505	5.69s
543	0.00284	0.00243	1.16826	5.69s
544	0.00282	0.00244	1.15959	5.69s
545	0.00282	0.00243	1.16342	5.69s
546	0.00281	0.00241	1.16442	5.72s
547	0.00281	0.00241	1.16571	5.69s
548	0.00282	0.00240	1.17201	5.69s
549	0.00281	0.00240	1.17066	5.70s
550	0.00278	0.00239	1.16104	5.69s
551	0.00277	0.00238	1.16529	5.71s
552	0.00276	0.00237	1.16409	5.69s
553	0.00280	0.00237	1.18423	5.69s
554	0.00274	0.00236	1.16450	5.70s
555	0.00273	0.00235	1.16169	5.69s
556	0.00273	0.00235	1.16255	5.70s
557	0.00274	0.00235	1.16773	5.69s
558	0.00273	0.00234	1.16945	5.69s
559	0.00275	0.00233	1.18200	5.69s
560	0.00273	0.00233	1.17512	5.69s
561	0.00272	0.00231	1.17667	5.69s
562	0.00270	0.00230	1.17199	5.69s
563	0.00272	0.00230	1.18253	5.69s
564	0.00270	0.00229	1.17714	5.69s
565	0.00270	0.00229	1.17778	5.69s
566	0.00269	0.00229	1.17451	5.69s
567	0.00269	0.00228	1.17697	5.69s
568	0.00270	0.00227	1.18735	5.69s
569	0.00270	0.00226	1.19300	5.69s
570	0.00267	0.00226	1.18220	5.71s
571	0.00266	0.00226	1.17974	5.69s
572	0.00267	0.00225	1.18665	5.69s
573	0.00262	0.00224	1.16897	5.69s
574	0.00265	0.00223	1.18637	5.69s
575	0.00263	0.00224	1.17533	5.69s
576	0.00261	0.00223	1.17071	5.69s
577	0.00262	0.00222	1.17683	5.69s
578	0.00261	0.00223	1.17228	5.69s
579	0.00261	0.00221	1.18155	5.69s
580	0.00261	0.00221	1.18358	5.68s
581	0.00258	0.00220	1.17379	5.69s
582	0.00258	0.00221	1.16742	5.69s
583	0.00257	0.00220	1.17178	5.69s
584	0.00260	0.00219	1.18756	5.69s
585	0.00257	0.00219	1.17626	5.69s
586	0.00256	0.00218	1.17507	5.70s

587	0.00256	0.00217	1.18099	5.69s
588	0.00255	0.00216	1.18197	5.69s
589	0.00256	0.00217	1.18079	5.69s
590	0.00256	0.00216	1.18508	5.69s
591	0.00257	0.00215	1.19574	5.70s
592	0.00256	0.00215	1.18996	5.68s
593	0.00252	0.00213	1.18039	5.69s
594	0.00253	0.00214	1.18573	5.69s
595	0.00254	0.00214	1.18761	5.69s
596	0.00251	0.00213	1.17767	5.69s
597	0.00254	0.00213	1.19101	5.69s
598	0.00252	0.00213	1.18294	5.69s
599	0.00249	0.00212	1.17704	5.69s
600	0.00250	0.00212	1.17865	5.70s
601	0.00251	0.00212	1.18516	5.70s
602	0.00252	0.00211	1.19235	5.69s
603	0.00251	0.00210	1.19301	5.70s
604	0.00249	0.00209	1.19136	5.70s
605	0.00249	0.00210	1.19018	5.69s
606	0.00248	0.00209	1.18624	5.69s
607	0.00248	0.00208	1.19088	5.69s
608	0.00246	0.00207	1.18767	5.69s
609	0.00245	0.00208	1.17456	5.69s
610	0.00246	0.00208	1.18179	5.69s
611	0.00245	0.00206	1.18518	5.69s
612	0.00246	0.00206	1.19336	5.69s
613	0.00244	0.00205	1.19001	5.69s
614	0.00244	0.00207	1.18283	5.69s
615	0.00245	0.00206	1.18957	5.70s
616	0.00244	0.00204	1.19251	5.70s
617	0.00240	0.00205	1.17228	5.70s
618	0.00243	0.00205	1.18517	5.70s
619	0.00243	0.00205	1.18842	5.70s
620	0.00242	0.00203	1.19433	5.71s
621	0.00239	0.00204	1.17470	5.70s
622	0.00240	0.00203	1.18255	5.72s
623	0.00241	0.00202	1.18963	5.69s
624	0.00239	0.00202	1.17900	5.69s
625	0.00242	0.00202	1.19928	5.69s
626	0.00238	0.00201	1.18554	5.68s
627	0.00239	0.00201	1.19122	5.69s
628	0.00239	0.00200	1.19467	5.69s
629	0.00239	0.00200	1.19318	5.69s
630	0.00239	0.00199	1.19962	5.69s
631	0.00238	0.00199	1.19203	5.70s
632	0.00239	0.00199	1.19925	5.69s
633	0.00235	0.00199	1.18122	5.69s
634	0.00237	0.00198	1.19319	5.69s
635	0.00238	0.00198	1.19923	5.70s
636	0.00237	0.00199	1.19448	5.70s
637	0.00235	0.00198	1.18636	5.70s
638	0.00235	0.00197	1.19617	5.69s
639	0.00237	0.00198	1.19749	5.69s
640	0.00236	0.00196	1.20566	5.69s
641	0.00234	0.00197	1.18791	5.69s
642	0.00230	0.00195	1.17734	5.69s
643	0.00233	0.00196	1.19179	5.70s

644	0.00235	0.00196	1.19824	5.68s
645	0.00231	0.00194	1.18936	5.69s
646	0.00233	0.00195	1.19475	5.69s
647	0.00232	0.00194	1.19603	5.69s
648	0.00231	0.00194	1.18988	5.69s
649	0.00234	0.00195	1.20094	5.70s
650	0.00231	0.00193	1.19812	5.69s
651	0.00230	0.00193	1.18907	5.69s
652	0.00229	0.00192	1.18984	5.70s
653	0.00227	0.00192	1.18243	5.69s
654	0.00229	0.00192	1.19032	5.70s
655	0.00230	0.00193	1.19266	5.69s
656	0.00229	0.00191	1.19387	5.69s
657	0.00230	0.00191	1.20105	5.69s
658	0.00229	0.00191	1.20109	5.69s
659	0.00229	0.00191	1.19702	5.69s
660	0.00227	0.00191	1.18926	5.69s
661	0.00228	0.00190	1.19909	5.69s
662	0.00226	0.00189	1.19241	5.69s
663	0.00227	0.00190	1.19848	5.69s
664	0.00227	0.00189	1.19681	5.70s
665	0.00226	0.00189	1.19307	5.69s
666	0.00226	0.00189	1.19367	5.69s
667	0.00224	0.00189	1.18759	5.69s
668	0.00225	0.00188	1.19947	5.69s
669	0.00223	0.00188	1.18814	5.69s
670	0.00223	0.00187	1.19152	5.69s
671	0.00223	0.00187	1.19200	5.69s
672	0.00224	0.00187	1.19970	5.68s
673	0.00223	0.00187	1.19295	5.68s
674	0.00223	0.00187	1.19315	5.72s
675	0.00225	0.00186	1.20707	5.69s
676	0.00221	0.00187	1.18385	5.69s
677	0.00222	0.00186	1.19203	5.69s
678	0.00223	0.00186	1.20005	5.69s
679	0.00222	0.00185	1.19532	5.69s
680	0.00220	0.00186	1.18275	5.69s
681	0.00221	0.00185	1.19346	5.69s
682	0.00221	0.00186	1.19014	5.69s
683	0.00222	0.00184	1.20834	5.70s
684	0.00222	0.00185	1.20127	5.69s
685	0.00219	0.00184	1.18973	5.69s
686	0.00220	0.00184	1.19541	5.69s
687	0.00217	0.00183	1.18963	5.69s
688	0.00218	0.00183	1.19350	5.68s
689	0.00218	0.00184	1.18404	5.69s
690	0.00221	0.00182	1.21184	5.69s
691	0.00220	0.00183	1.20535	5.70s
692	0.00218	0.00182	1.19347	5.69s
693	0.00219	0.00181	1.20633	5.70s
694	0.00217	0.00182	1.19070	5.69s
695	0.00216	0.00181	1.19563	5.69s
696	0.00216	0.00182	1.18443	5.70s
697	0.00214	0.00181	1.17915	5.69s
698	0.00214	0.00181	1.18037	5.69s
699	0.00216	0.00180	1.20243	5.69s
700	0.00215	0.00180	1.19075	5.69s

701	0.00216	0.00180	1.19547	5.69s
702	0.00216	0.00180	1.20104	5.70s
703	0.00215	0.00180	1.19437	5.69s
704	0.00215	0.00180	1.19451	5.69s
705	0.00215	0.00180	1.19740	5.69s
706	0.00213	0.00179	1.19079	5.69s
707	0.00215	0.00178	1.20265	5.69s
708	0.00215	0.00177	1.21469	5.69s
709	0.00213	0.00178	1.19359	5.69s
710	0.00214	0.00178	1.19964	5.69s
711	0.00212	0.00178	1.19237	5.69s
712	0.00213	0.00177	1.20570	5.69s
713	0.00211	0.00177	1.19489	5.70s
714	0.00215	0.00177	1.21050	5.69s
715	0.00212	0.00177	1.20141	5.69s
716	0.00211	0.00177	1.19090	5.70s
717	0.00211	0.00176	1.19323	5.69s
718	0.00210	0.00176	1.19499	5.69s
719	0.00211	0.00176	1.20104	5.69s
720	0.00213	0.00175	1.21428	5.69s
721	0.00208	0.00175	1.18982	5.69s
722	0.00210	0.00175	1.20063	5.69s
723	0.00208	0.00175	1.18856	5.68s
724	0.00208	0.00175	1.18674	5.69s
725	0.00206	0.00175	1.17717	5.69s
726	0.00211	0.00175	1.20871	5.69s
727	0.00208	0.00174	1.19608	5.69s
728	0.00207	0.00174	1.19129	5.69s
729	0.00207	0.00174	1.18810	5.69s
730	0.00208	0.00173	1.20786	5.69s
731	0.00205	0.00173	1.18951	5.70s
732	0.00207	0.00173	1.19364	5.69s
733	0.00208	0.00173	1.20275	5.69s
734	0.00206	0.00172	1.19742	5.69s
735	0.00207	0.00172	1.19994	5.68s
736	0.00207	0.00172	1.20475	5.69s
737	0.00206	0.00172	1.19394	5.69s
738	0.00206	0.00172	1.20066	5.69s
739	0.00205	0.00171	1.19411	5.69s
740	0.00204	0.00171	1.18981	5.69s
741	0.00204	0.00171	1.19691	5.69s
742	0.00205	0.00171	1.20454	5.69s
743	0.00205	0.00171	1.20200	5.69s
744	0.00206	0.00170	1.21295	5.69s
745	0.00203	0.00170	1.19307	5.69s
746	0.00204	0.00170	1.20106	5.69s
747	0.00205	0.00170	1.20174	5.69s
748	0.00203	0.00170	1.19702	5.69s
749	0.00203	0.00169	1.20483	5.69s
750	0.00203	0.00170	1.19820	5.69s
751	0.00204	0.00169	1.20909	5.69s
752	0.00203	0.00169	1.20326	5.69s
753	0.00201	0.00168	1.19686	5.68s
754	0.00203	0.00168	1.20516	5.69s
755	0.00201	0.00168	1.20147	5.68s
756	0.00202	0.00168	1.20034	5.68s
757	0.00201	0.00168	1.19349	5.68s

758	0.00200	0.00168	1.19116	5.68s
759	0.00202	0.00168	1.20647	5.69s
760	0.00199	0.00167	1.19097	5.69s
761	0.00200	0.00167	1.20294	5.70s
762	0.00200	0.00166	1.20472	5.70s
763	0.00200	0.00166	1.20259	5.69s
764	0.00199	0.00166	1.20015	5.69s
765	0.00199	0.00166	1.20005	5.68s
766	0.00202	0.00165	1.22283	5.69s
767	0.00200	0.00166	1.20856	5.69s
768	0.00198	0.00166	1.19454	5.69s
769	0.00200	0.00166	1.20526	5.68s
770	0.00199	0.00165	1.20792	5.68s
771	0.00201	0.00165	1.22072	5.68s
772	0.00197	0.00165	1.19466	5.69s
773	0.00198	0.00165	1.20393	5.69s
774	0.00197	0.00165	1.19181	5.69s
775	0.00199	0.00165	1.20121	5.69s
776	0.00197	0.00165	1.19137	5.69s
777	0.00196	0.00163	1.20241	5.69s
778	0.00195	0.00164	1.18535	5.69s
779	0.00195	0.00164	1.19177	5.69s
780	0.00197	0.00164	1.19963	5.70s
781	0.00197	0.00164	1.20449	5.69s
782	0.00195	0.00164	1.19354	5.69s
783	0.00195	0.00164	1.19258	5.69s
784	0.00196	0.00163	1.20055	5.69s
785	0.00196	0.00164	1.19727	5.68s
786	0.00196	0.00163	1.20634	5.69s
787	0.00195	0.00163	1.19072	5.69s
788	0.00195	0.00162	1.20319	5.69s
789	0.00194	0.00163	1.18720	5.69s
790	0.00195	0.00162	1.19850	5.69s
791	0.00196	0.00162	1.21440	5.69s
792	0.00195	0.00161	1.20890	5.69s
793	0.00195	0.00162	1.20055	5.69s
794	0.00193	0.00161	1.19906	5.69s
795	0.00192	0.00162	1.18563	5.69s
796	0.00192	0.00161	1.19354	5.69s
797	0.00194	0.00161	1.20509	5.68s
798	0.00194	0.00161	1.20447	5.68s
799	0.00193	0.00161	1.19979	5.68s
800	0.00193	0.00160	1.20443	5.69s
801	0.00194	0.00161	1.20440	5.69s
802	0.00193	0.00159	1.21055	5.68s
803	0.00192	0.00160	1.19719	5.72s
804	0.00193	0.00159	1.21044	5.69s
805	0.00192	0.00160	1.19979	5.69s
806	0.00191	0.00159	1.20020	5.69s
807	0.00192	0.00159	1.20285	5.69s
808	0.00192	0.00159	1.20417	5.69s
809	0.00191	0.00159	1.19764	5.69s
810	0.00190	0.00159	1.19776	5.69s
811	0.00190	0.00159	1.19525	5.69s
812	0.00190	0.00158	1.20396	5.69s
813	0.00189	0.00158	1.19767	5.69s
814	0.00190	0.00159	1.19646	5.69s

815	0.00192	0.00159	1.20673	5.69s
816	0.00188	0.00158	1.19312	5.68s
817	0.00189	0.00158	1.19466	5.69s
818	0.00191	0.00158	1.21254	5.69s
819	0.00190	0.00158	1.20135	5.69s
820	0.00189	0.00158	1.20165	5.69s
821	0.00189	0.00157	1.20345	5.70s
822	0.00187	0.00157	1.19301	5.69s
823	0.00188	0.00157	1.19609	5.70s
824	0.00189	0.00157	1.20283	5.69s
825	0.00188	0.00156	1.20266	5.69s
826	0.00189	0.00156	1.20626	5.69s
827	0.00189	0.00157	1.20174	5.69s
828	0.00186	0.00155	1.19960	5.69s
829	0.00189	0.00157	1.20566	5.69s
830	0.00187	0.00156	1.20457	5.69s
831	0.00187	0.00156	1.20084	5.69s
832	0.00185	0.00155	1.18998	5.69s
833	0.00188	0.00155	1.21014	5.69s
834	0.00186	0.00155	1.20500	5.69s
835	0.00186	0.00155	1.20023	5.69s
836	0.00185	0.00155	1.19530	5.69s
837	0.00186	0.00155	1.19844	5.69s
838	0.00187	0.00155	1.20455	5.69s
839	0.00183	0.00155	1.18598	5.70s
840	0.00186	0.00154	1.20713	5.69s
841	0.00187	0.00155	1.21079	5.69s
842	0.00186	0.00154	1.21130	5.69s
843	0.00186	0.00154	1.20528	5.69s
844	0.00186	0.00154	1.20574	5.68s
845	0.00184	0.00155	1.19193	5.68s
846	0.00186	0.00153	1.21375	5.68s
847	0.00184	0.00154	1.19564	5.68s
848	0.00183	0.00153	1.19497	5.68s
849	0.00185	0.00153	1.20602	5.68s
850	0.00183	0.00153	1.19705	5.68s
851	0.00182	0.00153	1.18756	5.69s
852	0.00185	0.00153	1.20760	5.68s
853	0.00183	0.00153	1.19482	5.68s
854	0.00184	0.00153	1.20172	5.69s
855	0.00183	0.00153	1.19698	5.69s
856	0.00180	0.00152	1.18729	5.69s
857	0.00183	0.00152	1.20330	5.68s
858	0.00184	0.00151	1.21318	5.69s
859	0.00183	0.00151	1.20801	5.68s
860	0.00182	0.00152	1.19818	5.69s
861	0.00183	0.00151	1.20531	5.68s
862	0.00182	0.00151	1.20773	5.68s
863	0.00181	0.00152	1.19095	5.68s
864	0.00182	0.00152	1.19811	5.69s
865	0.00182	0.00151	1.19826	5.68s
866	0.00181	0.00151	1.19464	5.68s
867	0.00181	0.00151	1.20132	5.68s
868	0.00178	0.00152	1.17282	5.70s
869	0.00179	0.00151	1.18641	5.69s
870	0.00179	0.00151	1.19139	5.69s
871	0.00180	0.00150	1.19657	5.69s

872	0.00179	0.00151	1.18223	5.68s
873	0.00180	0.00150	1.19962	5.69s
874	0.00178	0.00150	1.18399	5.68s
875	0.00179	0.00150	1.19346	5.68s
876	0.00178	0.00150	1.18664	5.68s
877	0.00179	0.00150	1.19133	5.68s
878	0.00178	0.00150	1.19108	5.68s
879	0.00177	0.00150	1.18471	5.68s
880	0.00177	0.00150	1.18005	5.68s
881	0.00178	0.00149	1.19777	5.71s
882	0.00177	0.00150	1.17656	5.68s
883	0.00180	0.00149	1.20414	5.68s
884	0.00176	0.00149	1.18768	5.69s
885	0.00179	0.00150	1.19224	5.69s
886	0.00178	0.00148	1.20314	5.69s
887	0.00178	0.00149	1.19860	5.69s
888	0.00177	0.00148	1.19907	5.69s
889	0.00179	0.00149	1.20201	5.68s
890	0.00177	0.00148	1.19660	5.71s
891	0.00177	0.00147	1.19820	5.71s
892	0.00178	0.00148	1.19897	5.68s
893	0.00177	0.00148	1.19602	5.69s
894	0.00177	0.00148	1.19245	5.68s
895	0.00175	0.00148	1.18482	5.68s
896	0.00177	0.00147	1.20495	5.69s
897	0.00175	0.00148	1.18668	5.68s
898	0.00176	0.00147	1.19791	5.69s
899	0.00174	0.00147	1.18498	5.69s
900	0.00175	0.00147	1.19454	5.69s
901	0.00176	0.00147	1.19816	5.69s
902	0.00175	0.00146	1.19606	5.69s
903	0.00177	0.00147	1.20128	5.69s
904	0.00174	0.00146	1.19353	5.68s
905	0.00174	0.00146	1.18977	5.69s
906	0.00175	0.00147	1.19094	5.69s
907	0.00175	0.00146	1.19466	5.68s
908	0.00175	0.00146	1.19737	5.68s
909	0.00174	0.00146	1.19424	5.69s
910	0.00175	0.00145	1.20419	5.68s
911	0.00174	0.00146	1.19327	5.69s
912	0.00173	0.00145	1.19501	5.70s
913	0.00173	0.00145	1.19400	5.70s
914	0.00174	0.00146	1.18922	5.69s
915	0.00174	0.00146	1.19439	5.69s
916	0.00174	0.00146	1.19469	5.68s
917	0.00174	0.00145	1.19981	5.69s
918	0.00175	0.00145	1.20329	5.69s
919	0.00173	0.00145	1.19181	5.69s
920	0.00173	0.00144	1.19817	5.69s
921	0.00172	0.00145	1.18846	5.69s
922	0.00171	0.00145	1.18184	5.68s
923	0.00171	0.00145	1.18253	5.68s
924	0.00171	0.00145	1.18412	5.69s
925	0.00172	0.00144	1.19260	5.68s
926	0.00173	0.00144	1.20177	5.68s
927	0.00173	0.00144	1.19568	5.69s
928	0.00172	0.00145	1.18872	5.69s

929	0.00171	0.00145	1.18445	5.69s
930	0.00173	0.00144	1.19935	5.70s
931	0.00173	0.00145	1.19610	5.69s
932	0.00171	0.00144	1.18898	5.69s
933	0.00171	0.00143	1.19239	5.69s
934	0.00171	0.00144	1.18892	5.68s
935	0.00171	0.00144	1.18939	5.68s
936	0.00170	0.00144	1.18096	5.68s
937	0.00169	0.00143	1.18618	5.68s
938	0.00171	0.00144	1.19083	5.68s
939	0.00170	0.00143	1.18661	5.68s
940	0.00171	0.00143	1.19427	5.68s
941	0.00172	0.00143	1.20384	5.68s
942	0.00172	0.00143	1.20785	5.68s
943	0.00170	0.00142	1.19330	5.69s
944	0.00171	0.00143	1.20167	5.69s
945	0.00171	0.00143	1.20193	5.69s
946	0.00170	0.00143	1.18718	5.69s
947	0.00169	0.00142	1.18818	5.69s
948	0.00169	0.00143	1.18098	5.69s
949	0.00169	0.00142	1.18771	5.69s
950	0.00171	0.00142	1.20464	5.69s
951	0.00171	0.00142	1.20754	5.69s
952	0.00168	0.00142	1.18593	5.70s
953	0.00168	0.00142	1.18352	5.68s
954	0.00166	0.00142	1.17382	5.68s
955	0.00169	0.00142	1.18581	5.68s
956	0.00166	0.00141	1.18013	5.68s
957	0.00168	0.00142	1.18813	5.69s
958	0.00169	0.00141	1.20062	5.68s
959	0.00167	0.00141	1.18053	5.68s
960	0.00166	0.00141	1.17545	5.69s
961	0.00167	0.00141	1.18276	5.69s
962	0.00168	0.00141	1.19038	5.69s
963	0.00168	0.00140	1.19579	5.69s
964	0.00167	0.00141	1.18501	5.69s
965	0.00168	0.00141	1.18976	5.69s
966	0.00167	0.00140	1.18826	5.69s
967	0.00168	0.00141	1.19323	5.70s
968	0.00167	0.00140	1.19292	5.68s
969	0.00166	0.00141	1.17441	5.68s
970	0.00169	0.00141	1.20014	5.68s
971	0.00166	0.00140	1.18223	5.68s
972	0.00166	0.00140	1.18374	5.68s
973	0.00167	0.00141	1.18362	5.68s
974	0.00167	0.00140	1.19302	5.68s
975	0.00164	0.00140	1.17644	5.68s
976	0.00166	0.00140	1.18166	5.69s
977	0.00166	0.00139	1.18905	5.69s
978	0.00165	0.00139	1.18888	5.69s
979	0.00166	0.00139	1.19451	5.69s
980	0.00166	0.00139	1.19207	5.69s
981	0.00165	0.00139	1.18981	5.68s
982	0.00164	0.00140	1.17688	5.68s
983	0.00166	0.00140	1.18831	5.68s
984	0.00165	0.00139	1.18822	5.68s
985	0.00164	0.00139	1.18363	5.68s

986	0.00165	0.00139	1.18580	5.69s
987	0.00166	0.00139	1.19682	5.68s
988	0.00165	0.00139	1.18506	5.70s
989	0.00164	0.00138	1.18298	5.69s
990	0.00166	0.00139	1.19481	5.68s
991	0.00163	0.00139	1.16743	5.69s
992	0.00165	0.00138	1.19015	5.69s
993	0.00165	0.00139	1.19003	5.69s
994	0.00165	0.00139	1.18824	5.69s
995	0.00164	0.00138	1.18580	5.69s
996	0.00164	0.00138	1.19060	5.69s
997	0.00163	0.00138	1.18452	5.69s
998	0.00164	0.00138	1.19250	5.71s
999	0.00163	0.00138	1.17986	5.68s
1000	0.00163	0.00138	1.18146	5.68s
1001	0.00163	0.00137	1.18628	5.68s
1002	0.00163	0.00138	1.17911	5.68s
1003	0.00163	0.00138	1.18600	5.68s
1004	0.00163	0.00137	1.18452	5.69s
1005	0.00162	0.00137	1.18115	5.69s
1006	0.00163	0.00137	1.18867	5.69s
1007	0.00162	0.00137	1.17540	5.69s
1008	0.00163	0.00138	1.18286	5.69s
1009	0.00163	0.00137	1.18991	5.69s
1010	0.00162	0.00137	1.18343	5.69s
1011	0.00163	0.00137	1.18844	5.69s
1012	0.00161	0.00136	1.18339	5.69s
1013	0.00163	0.00137	1.18519	5.69s
1014	0.00161	0.00137	1.17150	5.69s
1015	0.00164	0.00136	1.20400	5.69s
1016	0.00162	0.00138	1.17528	5.68s
1017	0.00163	0.00136	1.19938	5.68s
1018	0.00161	0.00137	1.18164	5.69s
1019	0.00160	0.00137	1.17215	5.69s
1020	0.00161	0.00135	1.19379	5.68s
1021	0.00161	0.00137	1.17527	5.69s
1022	0.00163	0.00136	1.20085	5.68s
1023	0.00161	0.00136	1.18754	5.69s
1024	0.00163	0.00136	1.19948	5.69s
1025	0.00162	0.00136	1.18975	5.68s
1026	0.00159	0.00135	1.18339	5.69s
1027	0.00161	0.00136	1.18030	5.70s
1028	0.00159	0.00135	1.17576	5.69s
1029	0.00160	0.00136	1.17870	5.68s
1030	0.00161	0.00135	1.18840	5.69s
1031	0.00158	0.00136	1.16774	5.68s
1032	0.00159	0.00135	1.18213	5.68s
1033	0.00159	0.00135	1.17616	5.69s
1034	0.00159	0.00135	1.17594	5.69s
1035	0.00160	0.00135	1.18656	5.68s
1036	0.00159	0.00135	1.17574	5.68s
1037	0.00159	0.00135	1.17803	5.68s
1038	0.00160	0.00135	1.17964	5.69s
1039	0.00158	0.00135	1.17528	5.69s
1040	0.00159	0.00135	1.18014	5.69s
1041	0.00160	0.00135	1.18384	5.69s
1042	0.00159	0.00135	1.17803	5.69s

1043	0.00160	0.00134	1.18833	5.69s
1044	0.00158	0.00135	1.16977	5.69s
1045	0.00159	0.00134	1.19112	5.69s
1046	0.00160	0.00135	1.18660	5.68s
1047	0.00157	0.00134	1.16728	5.69s
1048	0.00159	0.00134	1.18396	5.68s
1049	0.00160	0.00134	1.19619	5.68s
1050	0.00159	0.00135	1.18556	5.68s
1051	0.00158	0.00134	1.17998	5.68s
1052	0.00156	0.00134	1.16761	5.68s
1053	0.00157	0.00134	1.17568	5.68s
1054	0.00157	0.00134	1.17538	5.68s
1055	0.00157	0.00133	1.18326	5.69s
1056	0.00157	0.00133	1.17971	5.69s
1057	0.00158	0.00133	1.18524	5.68s
1058	0.00156	0.00133	1.16693	5.69s
1059	0.00157	0.00132	1.18387	5.69s
1060	0.00157	0.00133	1.18267	5.68s
1061	0.00157	0.00133	1.17814	5.68s
1062	0.00157	0.00133	1.17966	5.68s
1063	0.00157	0.00133	1.18543	5.68s
1064	0.00158	0.00133	1.18848	5.68s
1065	0.00156	0.00133	1.17433	5.69s
1066	0.00157	0.00133	1.18093	5.68s
1067	0.00156	0.00133	1.17788	5.69s
1068	0.00154	0.00132	1.16559	5.68s
1069	0.00158	0.00133	1.18636	5.70s
1070	0.00155	0.00132	1.17625	5.69s
1071	0.00156	0.00133	1.18037	5.70s
1072	0.00157	0.00133	1.18588	5.69s
1073	0.00156	0.00132	1.17647	5.69s
1074	0.00157	0.00133	1.17961	5.69s
1075	0.00156	0.00133	1.17660	5.69s
1076	0.00156	0.00132	1.17497	5.69s
1077	0.00154	0.00132	1.16058	5.69s
1078	0.00155	0.00131	1.17625	5.69s
1079	0.00156	0.00131	1.18946	5.68s
1080	0.00153	0.00132	1.16359	5.68s
1081	0.00156	0.00132	1.17764	5.69s
1082	0.00154	0.00132	1.17035	5.68s
1083	0.00155	0.00132	1.17639	5.69s
1084	0.00154	0.00132	1.16969	5.68s
1085	0.00154	0.00131	1.17518	5.69s
1086	0.00155	0.00132	1.17285	5.69s
1087	0.00155	0.00131	1.17948	5.69s
1088	0.00154	0.00132	1.17308	5.69s
1089	0.00156	0.00132	1.18411	5.69s
1090	0.00155	0.00131	1.18338	5.69s
1091	0.00154	0.00131	1.17493	5.69s
1092	0.00154	0.00131	1.17518	5.69s
1093	0.00155	0.00131	1.18297	5.69s
1094	0.00155	0.00131	1.18346	5.69s
1095	0.00154	0.00131	1.17435	5.68s
1096	0.00156	0.00131	1.19054	5.69s
1097	0.00155	0.00131	1.18088	5.69s
1098	0.00152	0.00130	1.17252	5.69s
1099	0.00154	0.00131	1.17665	5.69s

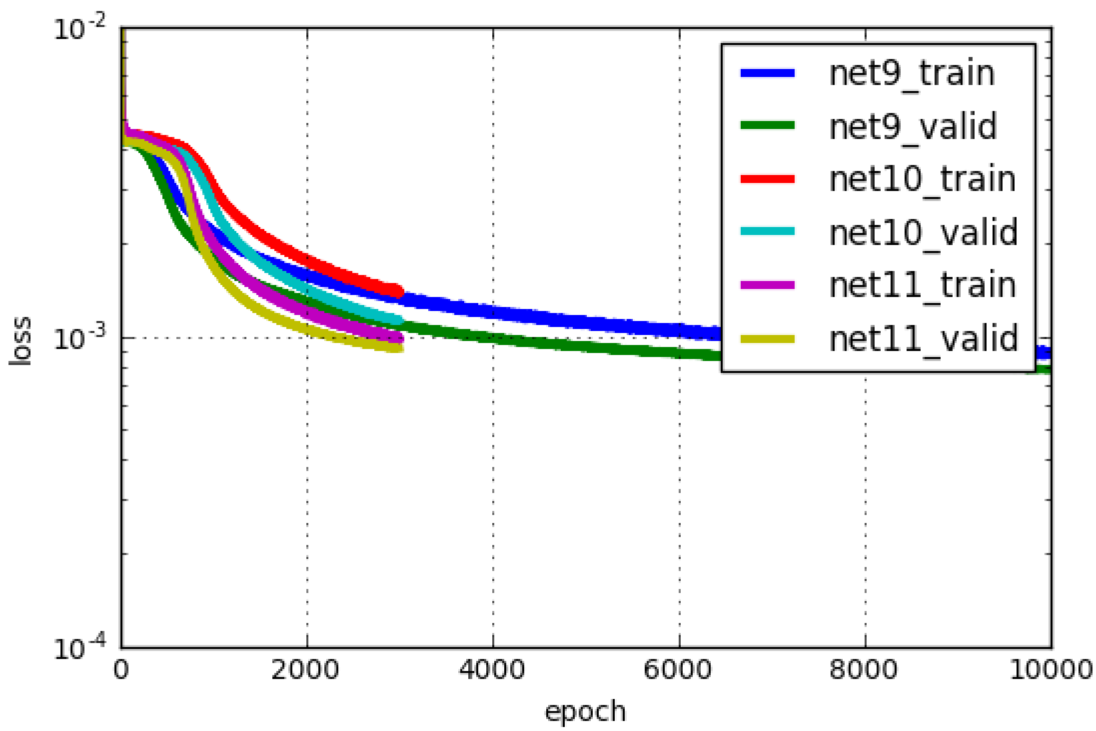
1100	0.00155	0.00130	1.18725	5.71s
1101	0.00154	0.00131	1.18125	5.69s
1102	0.00153	0.00130	1.17933	5.69s
1103	0.00153	0.00131	1.16973	5.69s
1104	0.00155	0.00131	1.18414	5.69s
1105	0.00154	0.00130	1.17961	5.69s
1106	0.00154	0.00130	1.18334	5.69s
1107	0.00154	0.00130	1.18216	5.68s
1108	0.00152	0.00130	1.16964	5.68s
1109	0.00154	0.00129	1.18807	5.68s
1110	0.00154	0.00130	1.18997	5.68s
1111	0.00152	0.00130	1.17041	5.69s
1112	0.00150	0.00130	1.16169	5.69s
1113	0.00151	0.00129	1.16901	5.69s
1114	0.00153	0.00130	1.17811	5.69s
1115	0.00151	0.00129	1.16972	5.69s
1116	0.00152	0.00129	1.17479	5.69s
1117	0.00152	0.00129	1.17837	5.69s
1118	0.00154	0.00130	1.18149	5.69s
1119	0.00152	0.00130	1.17466	5.69s
1120	0.00152	0.00130	1.16960	5.69s
1121	0.00153	0.00129	1.18062	5.69s
1122	0.00153	0.00129	1.18006	5.69s
1123	0.00152	0.00129	1.17367	5.69s
1124	0.00151	0.00129	1.16886	5.69s
1125	0.00151	0.00129	1.17099	5.68s
1126	0.00151	0.00129	1.17346	5.68s
1127	0.00151	0.00129	1.16594	5.68s
1128	0.00150	0.00129	1.16280	5.69s
1129	0.00151	0.00129	1.17381	5.68s
1130	0.00152	0.00130	1.17412	5.69s
1131	0.00152	0.00128	1.18324	5.69s
1132	0.00150	0.00129	1.16842	5.69s
1133	0.00150	0.00128	1.16581	5.69s
1134	0.00151	0.00129	1.17647	5.69s
1135	0.00152	0.00129	1.17966	5.69s
1136	0.00151	0.00128	1.17840	5.69s
1137	0.00150	0.00128	1.17117	5.68s
1138	0.00150	0.00128	1.16927	5.69s
1139	0.00152	0.00127	1.18993	5.68s
1140	0.00150	0.00128	1.17116	5.69s
1141	0.00152	0.00129	1.17764	5.68s
1142	0.00151	0.00128	1.18607	5.68s
1143	0.00149	0.00128	1.16066	5.69s
1144	0.00149	0.00127	1.17021	5.68s
1145	0.00150	0.00129	1.16598	5.69s
1146	0.00149	0.00127	1.17224	5.69s
1147	0.00151	0.00128	1.17625	5.69s
1148	0.00150	0.00128	1.17323	5.68s
1149	0.00148	0.00127	1.16414	5.68s
1150	0.00151	0.00128	1.18560	5.69s
1151	0.00150	0.00127	1.17629	5.69s
1152	0.00149	0.00128	1.16332	5.68s
1153	0.00150	0.00127	1.18304	5.68s
1154	0.00149	0.00127	1.17137	5.68s
1155	0.00149	0.00127	1.17610	5.68s
1156	0.00151	0.00128	1.17855	5.68s

```
  1157     0.00149     0.00127    1.17243  5.68s
  1158     0.00149     0.00127    1.17151  5.68s
  1159     0.00150     0.00127    1.18276  5.68s
  1160     0.00148     0.00127    1.16950  5.68s
  1161     0.00149     0.00126    1.17806  5.69s
  1162     0.00150     0.00127    1.18093  5.69s
  1163     0.00149     0.00127    1.17790  5.69s
  1164     0.00149     0.00127    1.17355  5.69s
  1165     0.00148     0.00127    1.16738  5.68s
  1166     0.00148     0.00127    1.16930  5.69s
  1167     0.00148     0.00127    1.16960  5.69s
  1168     0.00150     0.00126    1.18388  5.69s
  1169     0.00149     0.00127    1.16973  5.69s
  1170     0.00148     0.00126    1.17484  5.68s
  1171     0.00148     0.00127    1.16640  5.69s
  1172     0.00147     0.00127    1.16283  5.68s
  1173     0.00148     0.00126    1.17814  5.69s
  1174     0.00147     0.00127    1.16191  5.68s
  1175     0.00150     0.00126    1.18857  5.68s
  1176     0.00146     0.00126    1.15510  5.68s
  1177     0.00148     0.00127    1.16738  5.69s
  1178     0.00147     0.00127    1.16404  5.68s
  1179     0.00146     0.00126    1.15800  5.69s
  1180     0.00148     0.00126    1.17398  5.69s
  1181     0.00148     0.00125    1.17845  5.69s
  1182     0.00146     0.00126    1.16141  5.68s
  1183     0.00148     0.00125    1.17868  5.69s
  1184     0.00148     0.00126    1.17445  5.69s
  1185     0.00145     0.00126    1.15383  5.68s
  1186     0.00146     0.00126    1.16315  5.68s
  1187     0.00146     0.00126    1.16590  5.68s
  1188     0.00148     0.00125    1.17745  5.69s
  1189     0.00145     0.00125    1.16302  5.69s
  1190     0.00148     0.00125    1.18948  5.69s
  1191     0.00146     0.00126    1.16262  5.69s
  1192     0.00147     0.00126    1.16492  5.68s
  1193     0.00147     0.00126    1.16519  5.69s
  1194     0.00146     0.00125    1.16700  5.69s
  1195     0.00146     0.00125    1.16166  5.68s
  1196     0.00147     0.00125    1.17193  5.69s
```

In [ ]:

```
# -*- coding: utf-8 -*-
import theano
import pickle

# Load from pickle file after first run

net9pickle = ""/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetection
_Misc/net9.pickle"
net10pickle = ""/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetectio
n_Misc/net10.pickle"
net11pickle = ""/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetectio
n_Misc/net11.pickle"

net9 = pickle.load(open(net9pickle,'rb'))
net10 = pickle.load(open(net10pickle,'rb'))
net11 = pickle.load(open(net11pickle,'rb'))
```

In [5]:
```
# plot
net9_train_loss = np.array([i["train_loss"] for i in net9.train_history_])
net9_valid_loss = np.array([i["valid_loss"] for i in net9.train_history_])
net10_train_loss = np.array([i["train_loss"] for i in net10.train_history_
])
net10_valid_loss = np.array([i["valid_loss"] for i in net10.train_history_
])
net11_train_loss = np.array([i["train_loss"] for i in net11.train_history_
])
net11_valid_loss = np.array([i["valid_loss"] for i in net11.train_history_
])
pyplot.plot(net9_train_loss, linewidth=3, label="net9_train")
pyplot.plot(net9_valid_loss, linewidth=3, label="net9_valid")
pyplot.plot(net10_train_loss, linewidth=3, label="net10_train")
pyplot.plot(net10_valid_loss, linewidth=3, label="net10_valid")
pyplot.plot(net11_train_loss, linewidth=3, label="net11_train")
pyplot.plot(net11_valid_loss, linewidth=3, label="net11_valid")
pyplot.grid()
pyplot.legend()
pyplot.xlabel("epoch")
pyplot.ylabel("loss")
pyplot.ylim(1e-4, 1e-2)
pyplot.yscale("log")
pyplot.show()
```



From the above graph, simply adding 1 more convolution does not improve the performance compared to net9. However, reduce the hidden layer from 3 to 2 does actually help with the accuracy. One thing also interesting is that I expect having 1 less hidden layer would reduce the time to train each epoch because it require time to train 1 more layer, adjust weight, and store those addictional information. But from the result (which stores as net10.out in this directory), the time it takes for each epoch is the same for net10 and net11.

Net11 ends up with loss_val of 0.00091 at epoch 3000 compared to 0.00108 of net9 at epoch 3000. This indicatse that net11 is highly likely to also perform better than net9 at epoch 10000.

In fact, after letting net11 run through 10000 epochs, we get the lowest validation loss of 0.0007.

11. Kaggle Submission Results

In this section, we provide the Kaggle scores that each of our network models received. It was surprising to see that lowest validation loss did not necessarily indicate best submission score. We also heard that there were lots of folks trying to hack Kaggle and we decided to try our hand at that a bit.

Submission and Description	Private Score	Public Score
submission-2017-04-23T12-43-28.076089.csv	2.85825	2.94950
3 minutes ago by SanjayDorairaj		
net7 - Initial neural net model with 10000 epochs		
submission-2017-04-23T06-32-13.222880.csv	2.59110	2.70009
14 hours ago by SanjayDorairaj		
averaging across multiple submissions..		
submission-2017-04-23T06-17-13.222880.csv	2.74622	2.84778
15 hours ago by SanjayDorairaj		
Net11 - this submission runs 10k epochs of a multi-layered CNN . 9 CONV, 3 Pooling, 3 dropout and 2 hidden layers		
submission-2017-04-22T11-07-25.895212.csv	3.45306	3.57878
a day ago by SanjayDorairaj		
Submission based on specialists used by dnouri with slight changes to increase epoch size and remove Early Stopping		

The submission that earned the highest score of 2.75 (still around 50 on the Kaggle leaderboard) was net11 trained over 10000 epochs.

Our starting model net7 earned a score of 2.85 when trained across 10000 epochs.

The specialists did not fare as well as expected and secured a Kaggle score of 3.45.

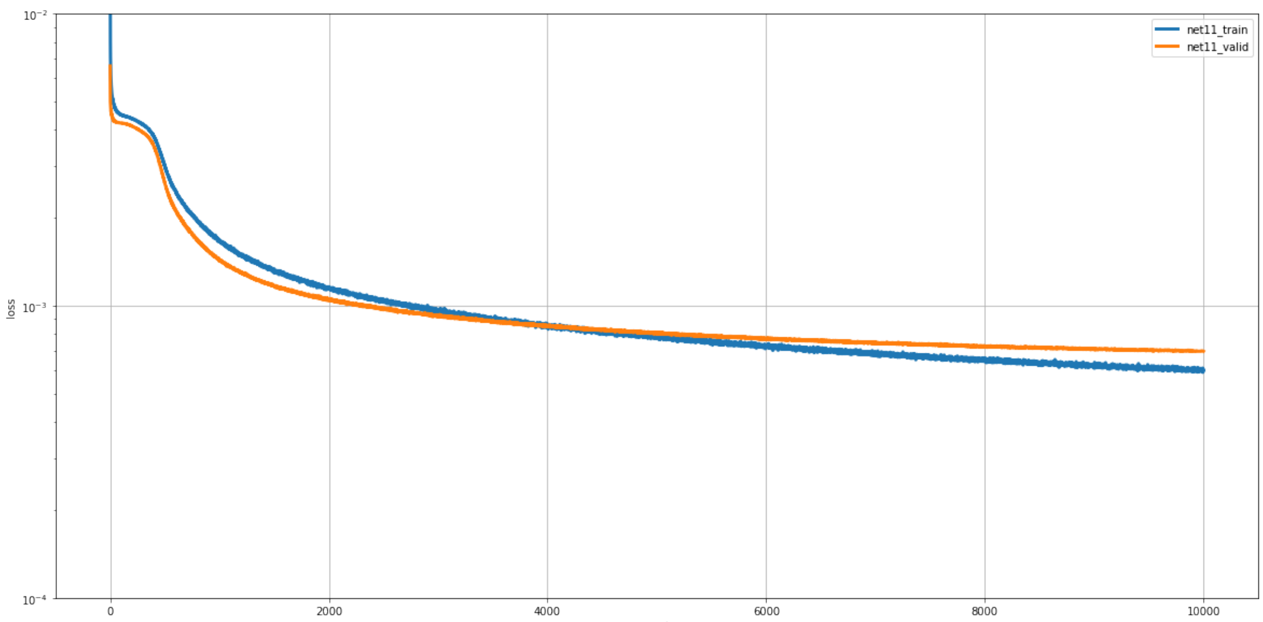
The hack that we tried was to average scores from multiple submissions to see where they landed. Surprisingly, our "highest score" of 2.59 was nothing but an average across net7 and net11.

One of the interesting things we noticed that lower validation ratios did not necessarily indicate better Kaggle scores.

We also noticed that there Kaggle had misstated when they mentioned that they used the RMSE as the Kaggle score. The RMSE yielded a much lower value relative to the Kaggle score.

12. Charting net11 - The model with the lowest validation loss

In [11]:
```
net11pickle = "/home/sdorairaj/dev/W207Spring2017_FacialKeyPointsDetection_Misc/net11_10000epochs.pickle"
net11 = pickle.load(open(net11pickle,'rb'))

# plot

net11_train_loss = np.array([i["train_loss"] for i in net11.train_history_])
net11_valid_loss = np.array([i["valid_loss"] for i in net11.train_history_])

pyplot.figure(figsize=(20,10))

pyplot.plot(net11_train_loss, linewidth=3, label="net11_train")
pyplot.plot(net11_valid_loss, linewidth=3, label="net11_valid")

pyplot.grid()
pyplot.legend()
pyplot.xlabel("epoch")
pyplot.ylabel("loss")
pyplot.ylim(1e-4, 1e-2)
pyplot.yscale("log")
pyplot.show()
```



The results of net11 show that there is still some overfitting going on and the model can possible be further improved by increasing the number of epochs, adding dropout and potentially by going even deeper by adding additional hidden layers. Note that several other possible optimizations such as modifying the convolution filter can also be performed to further improve the accuracy of the model.

13. Learnings

Incompatible Pickle versions across Python versions

Pickle files generated using Python 3 were not compatible with Python 2. This is because Python 3 uses protocol version 4, whereas, Python 2 uses protocol versions <=2. This incompatibility made it very difficult to exchange pickle files within the group for analysis.

To overcome this limitation we wrote a python program in tools/downgrade_pickle_version.py that converted pickle files in higher versions to pickle files in lower version

Even though, we were able to generate pickle files with lower protocol version we were still having some issues trying to read these files with Python 2.7.

Differences in Kaggle submission format

The Kaggle website mentions a different submission that what was actually required. This threw us off a bit until we were able to locate some resources on the web that helped us construct the correct format.

Kaggle recommended format:

Each row contains a RowId, ImageId, FeatureName, Location.

Actual format:

RowId, ImageId, Location (no feature name)

In [ ]:

```
```