

Artificial Intelligence Nanodegree

Computer Vision Capstone

Project: Facial Keypoint Detection

Welcome to the final Computer Vision project in the Artificial Intelligence Nanodegree program!

In this project, you'll combine your knowledge of computer vision techniques and deep learning to build and end-to-end facial keypoint recognition system! Facial keypoints include points around the eyes, nose, and mouth on any face and are used in many applications, from facial tracking to emotion recognition.

There are three main parts to this project:

Part 1 : Investigating OpenCV, pre-processing, and face detection

Part 2 : Training a Convolutional Neural Network (CNN) to detect facial keypoints

Part 3 : Putting parts 1 and 2 together to identify facial keypoints on any image!

**Here's what you need to know to complete the project:*

1. In this notebook, some template code has already been provided for you, and you will need to implement additional functionality to successfully complete this project. You will not need to modify the included code beyond what is requested.
 - a. Sections that begin with '**(IMPLEMENTATION)**' in the header indicate that the following block of code will require additional functionality which you must provide. Instructions will be provided for each section, and the specifics of the implementation are marked in the code block with a 'TODO' statement. Please be sure to read the instructions carefully!
1. In addition to implementing code, there will be questions that you must answer which relate to the project and your implementation.
 - a. Each section where you will answer a question is preceded by a '**Question X**' header.
 - b. Carefully read each question and provide thorough answers in the following text boxes that begin with '**Answer:**'.

Note: Code and Markdown cells can be executed using the **Shift + Enter** keyboard shortcut. Markdown cells can be edited by double-clicking the cell to enter edit mode.

The rubric contains **optional** suggestions for enhancing the project beyond the minimum requirements. If you decide to pursue the "(Optional)" sections, you should include the code in this IPython notebook.

Your project submission will be evaluated based on your answers to *each* of the questions and the code implementations you provide.

Steps to Complete the Project

Each part of the notebook is further broken down into separate steps. Feel free to use the links below to navigate the notebook.

In this project you will get to explore a few of the many computer vision algorithms built into the OpenCV library. This expansive computer vision library is now almost 20 years old (<https://en.wikipedia.org/wiki/OpenCV#History>) and still growing!

The project itself is broken down into three large parts, then even further into separate steps. Make sure to read through each step, and complete any sections that begin with '**(IMPLEMENTATION)**' in the header; these implementation sections may contain multiple TODOs that will be marked in code. For convenience, we provide links to each of these steps below.

Part 1 : Investigating OpenCV, pre-processing, and face detection

- [Step 0](#): Detect Faces Using a Haar Cascade Classifier
- [Step 1](#): Add Eye Detection
- [Step 2](#): De-noise an Image for Better Face Detection
- [Step 3](#): Blur an Image and Perform Edge Detection
- [Step 4](#): Automatically Hide the Identity of an Individual

Part 2 : Training a Convolutional Neural Network (CNN) to detect facial keypoints

- [Step 5](#): Create a CNN to Recognize Facial Keypoints
- [Step 6](#): Compile and Train the Model
- [Step 7](#): Visualize the Loss and Answer Questions

Part 3 : Putting parts 1 and 2 together to identify facial keypoints on any image!

- [Step 8](#): Build a Robust Facial Keypoints Detector (Complete the CV Pipeline)

Step 0: Detect Faces Using a Haar Cascade Classifier

Have you ever wondered how Facebook automatically tags images with your friends' faces? Or how high-end cameras automatically find and focus on a certain person's face? Applications like these depend heavily on the machine learning task known as *face detection* - which is the task of automatically finding faces in images containing people.

At its root face detection is a classification problem - that is a problem of distinguishing between distinct classes of things. With face detection these distinct classes are 1) images of human faces and 2) everything else.

We use OpenCV's implementation of [Haar feature-based cascade classifiers](#) (http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html) to detect human faces in images. OpenCV provides many pre-trained face detectors, stored as XML files on [github](#) (<https://github.com/opencv/opencv/tree/master/data/haarcascades>). We have downloaded one of these detectors and stored it in the detector_architectures directory.

Import Resources

In the next python cell, we load in the required libraries for this section of the project.

```
In [1]: # Import required Libraries for this section  
  
%matplotlib inline  
  
import numpy as np  
import matplotlib.pyplot as plt  
import math  
import cv2 # OpenCV Library for computer vision  
from PIL import Image  
import time
```

Next, we load in and display a test image for performing face detection.

Note: by default OpenCV assumes the ordering of our image's color channels are Blue, then Green, then Red. This is slightly out of order with most image types we'll use in these experiments, whose color channels are ordered Red, then Green, then Blue. In order to switch the Blue and Red channels of our test image around we will use OpenCV's cvtColor function, which you can read more about by [checking out some of its documentation located here](#) (http://docs.opencv.org/3.2.0/df/d9d/tutorial_py_colorspaces.html). This is a general utility function that can do other transformations too like converting a color image to grayscale, and transforming a standard color image to HSV color space.

```
In [2]: # Load in color image for face detection
image = cv2.imread('images/test_image_1.jpg')

# Convert the image to RGB colorspace
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Plot our image using subplots to specify a size and title
fig = plt.figure(figsize = (8,8))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Original Image')
# Just to prevent the notebook from outputting extra text.
_ = ax1.imshow(image)
```

Original Image



There are a lot of people - and faces - in this picture. 13 faces to be exact! In the next code cell, we demonstrate how to use a Haar Cascade classifier to detect all the faces in this test image.

This face detector uses information about patterns of intensity in an image to reliably detect faces under varying light conditions. So, to use this face detector, we'll first convert the image from color to grayscale.

Then, we load in the fully trained architecture of the face detector -- found in the file `haarcascade_frontalface_default.xml` - and use it on our image to find faces!

To learn more about the parameters of the detector see [this post](#) (<https://stackoverflow.com/questions/20801015/recommended-values-for-opencv-detectmultiscale-parameters>).

```
In [3]: # Convert the RGB image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Extract the pre-trained face detector from an xml file
face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_frontalface_default.xml')

# Detect the faces in image
faces = face_cascade.detectMultiScale(gray, 4, 6)

# Print the number of faces detected in the image
print('Number of faces detected:', len(faces))

# Make a copy of the orginal image to draw face detections on
image_with_detections = np.copy(image)

# Get the bounding box for each detected face
for (x,y,w,h) in faces:
    # Add a red bounding box to the detections image
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h), (255,0,0), 3)

# Display the image with the detections
fig = plt.figure(figsize = (8,8))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Image with Face Detections')
_ = ax1.imshow(image_with_detections)
```

Number of faces detected: 13

Image with Face Detections



In the above code, `faces` is a numpy array of detected faces, where each row corresponds to a detected face. Each detected face is a 1D array with four entries that specifies the bounding box of the detected face. The first two entries in the array (extracted in the above code as `x` and `y`) specify the horizontal and vertical positions of the top left corner of the bounding box. The last two entries in the array (extracted here as `w` and `h`) specify the width and height of the box.

Step 1: Add Eye Detections

There are other pre-trained detectors available that use a Haar Cascade Classifier - including full human body detectors, license plate detectors, and more. [A full list of the pre-trained architectures can be found here](#) (<https://github.com/opencv/opencv/tree/master/data/haarcascades>).

To test your eye detector, we'll first read in a new test image with just a single face.

```
In [4]: # Load in color image for face detection
image = cv2.imread('images/james.jpg')

# Convert the image to RGB colorspace
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Plot the RGB image
fig = plt.figure(figsize = (6,6))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Original Image')
_ = ax1.imshow(image)
```

Original Image



Notice that even though the image is a black and white image, we have read it in as a color image and so it will still need to be converted to grayscale in order to perform the most accurate face detection.

So, the next steps will be to convert this image to grayscale, then load OpenCV's face detector and run it with parameters that detect this face accurately.

```
In [5]: # Convert the RGB image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Extract the pre-trained face detector from an xml file
face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_frontalface_default.xml')

# Detect the faces in image
faces = face_cascade.detectMultiScale(gray, 1.25, 6)

# Print the number of faces detected in the image
print('Number of faces detected:', len(faces))

# Make a copy of the orginal image to draw face detections on
image_with_detections = np.copy(image)

# Get the bounding box for each detected face
for (x,y,w,h) in faces:
    # Add a red bounding box to the detections image
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h), (255,0,0), 3)

# Display the image with the detections
fig = plt.figure(figsize = (6,6))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Image with Face Detection')
_ = ax1.imshow(image_with_detections)
```

Number of faces detected: 1

Image with Face Detection



(IMPLEMENTATION) Add an eye detector to the current face detection setup.

A Haar-cascade eye detector can be included in the same way that the face detector was and, in this first task, it will be your job to do just this.

To set up an eye detector, use the stored parameters of the eye cascade detector, called `haarcascade_eye.xml`, located in the `detector_architectures` subdirectory. In the next code cell, create your eye detector and store its detections.

A few notes before you get started:

First, make sure to give your loaded eye detector the variable name

`eye_cascade`

and give the list of eye regions you detect the variable name

`eyes`

Second, since we've already run the face detector over this image, you should only search for eyes *within the rectangular face regions detected in faces*. This will minimize false detections.

Lastly, once you've run your eye detector over the facial detection region, you should display the RGB image with both the face detection boxes (in red) and your eye detections (in green) to verify that everything works as expected.

```
In [6]: # Make a copy of the original image to plot rectangle detections
image_with_detections = np.copy(image)

# Loop over the detections and draw their corresponding face detection boxes
for (x,y,w,h) in faces:
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h),(255,0,0), 3)

# Do not change the code above this comment!

## TODO: Add eye detection, using haarcascade_eye.xml, to the current face detector algorithm
## TODO: Loop over the eye detections and draw their corresponding boxes in green on image_with_detections
eye_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_eye.xml')
eyes = eye_cascade.detectMultiScale(gray, 1.3, 4)

print('Number of eyes detected:', len(eyes))

for (x,y,w,h) in eyes:
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h),(0,255,0), 3)

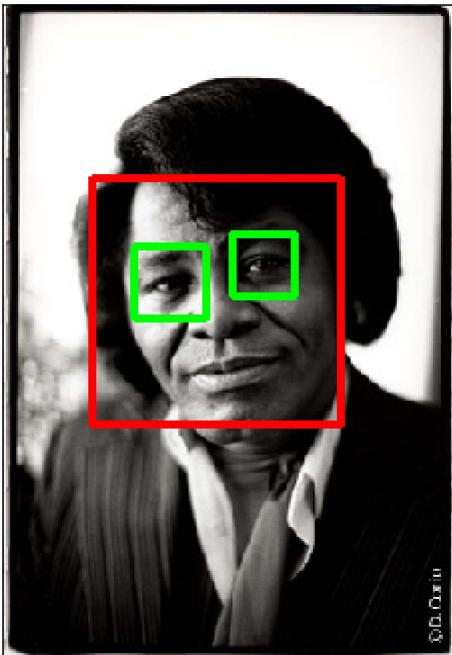
## END TODO

# Plot the image with both faces and eyes detected
fig = plt.figure(figsize = (6,6))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Image with Face and Eye Detection')
_ = ax1.imshow(image_with_detections)
```

Number of eyes detected: 2

Image with Face and Eye Detection



(Optional) Add face and eye detection to your laptop camera

It's time to kick it up a notch, and add face and eye detection to your laptop's camera! Afterwards, you'll be able to show off your creation like in the gif shown below - made with a completed version of the code!



Notice that not all of the detections here are perfect - and your result need not be perfect either. You should spend a small amount of time tuning the parameters of your detectors to get reasonable results, but don't hold out for perfection. If we wanted perfection we'd need to spend a ton of time tuning the parameters of each detector, cleaning up the input image frames, etc. You can think of this as more of a rapid prototype.

The next cell contains code for a wrapper function called `laptop_camera_face_eye_detector` that, when called, will activate your laptop's camera. You will place the relevant face and eye detection code in this wrapper function to implement face/eye detection and mark those detections on each image frame that your camera captures.

Before adding anything to the function, you can run it to get an idea of how it works - a small window should pop up showing you the live feed from your camera; you can press any key to close this window.

Note: Mac users may find that activating this function kills the kernel of their notebook every once in a while. If this happens to you, just restart your notebook's kernel, activate cell(s) containing any crucial import statements, and you'll be good to go!

```
In [7]: ### Add face and eye detection to this Laptop camera function
# Make sure to draw out all faces/eyes found in each frame on the shown video
# feed

import cv2
import time

def detect_eyes_face(image):
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

    face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_f
rontalface_default.xml')
    faces = face_cascade.detectMultiScale(gray, 1.4, 3)

    eye_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_ey
```

```

e_tree_eyeglasses.xml')
eyes = eye_cascade.detectMultiScale(gray, 1.3, 3)

image_with_detections = np.copy(image)
for (x,y,w,h) in faces:
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h),(255,0,0), 3)

for (x,y,w,h) in eyes:
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h),(0,255,0), 3)

return image_with_detections

# wrapper function for face/eye detection with your Laptop camera
def laptop_camera_go():
    # Create instance of video capturer
    cv2.namedWindow("face detection activated")
    vc = cv2.VideoCapture(0)

    # Try to get the first frame
    if vc.isOpened():
        rval, frame = vc.read()
    else:
        rval = False

    # Keep the video stream open
    while rval:
        # Plot the image from camera with all the face and eye detections marked
        frame = detect_eyes_face(frame)
        cv2.imshow("face detection activated", frame)

        # Exit functionality - press any key to exit Laptop video
        key = cv2.waitKey(20)
        if key > 0: # Exit by pressing any key
            # Destroy windows
            cv2.destroyAllWindows()

        # Make sure window closes on OSx
        for i in range (1,5):
            cv2.waitKey(1)
        return

        # Read next frame
        time.sleep(0.05)           # control framerate for computation - default 20 frames per sec
        rval, frame = vc.read()

```

In [8]: # Call the Laptop camera face/eye detector function above
#Laptop_camera_go()

Step 2: De-noise an Image for Better Face Detection

Image quality is an important aspect of any computer vision task. Typically, when creating a set of images to train a deep learning network, significant care is taken to ensure that training images are free of visual noise or artifacts that hinder object detection. While computer vision algorithms - like a face detector - are typically trained on 'nice' data such as this, new test data doesn't always look so nice!

When applying a trained computer vision algorithm to a new piece of test data one often cleans it up first before feeding it in. This sort of cleaning - referred to as *pre-processing* - can include a number of cleaning phases like blurring, de-noising, color transformations, etc., and many of these tasks can be accomplished using OpenCV.

In this short subsection we explore OpenCV's noise-removal functionality to see how we can clean up a noisy image, which we then feed into our trained face detector.

Create a noisy image to work with

In the next cell, we create an artificial noisy version of the previous multi-face image. This is a little exaggerated - we don't typically get images that are this noisy - but [image noise \(<https://digital-photography-school.com/how-to-avoid-and-reduce-noise-in-your-images/>\)](https://digital-photography-school.com/how-to-avoid-and-reduce-noise-in-your-images/), or 'grainy-ness' in a digital image - is a fairly common phenomenon.

```
In [9]: # Load in the multi-face test image again
image = cv2.imread('images/test_image_1.jpg')

# Convert the image copy to RGB colorspace
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Make an array copy of this image
image_with_noise = np.asarray(image)

# Create noise - here we add noise sampled randomly from a Gaussian distribution: a common model for noise
noise_level = 40
noise = np.random.randn(image.shape[0],image.shape[1],image.shape[2])*noise_level

# Add this noise to the array image copy
image_with_noise = image_with_noise + noise

# Convert back to uint8 format
image_with_noise = np.asarray([np.uint8(np.clip(i,0,255)) for i in image_with_noise])

# Plot our noisy image!
fig = plt.figure(figsize = (8,8))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Noisy Image')
_ = ax1.imshow(image_with_noise)
```

Noisy Image



In the context of face detection, the problem with an image like this is that - due to noise - we may miss some faces or get false detections.

In the next cell we apply the same trained OpenCV detector with the same settings as before, to see what sort of detections we get.

```
In [10]: # Convert the RGB image to grayscale
gray_noise = cv2.cvtColor(image_with_noise, cv2.COLOR_RGB2GRAY)

# Extract the pre-trained face detector from an xml file
face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_frontalface_default.xml')

# Detect the faces in image
faces = face_cascade.detectMultiScale(gray_noise, 4, 6)

# Print the number of faces detected in the image
print('Number of faces detected:', len(faces))

# Make a copy of the orginal image to draw face detections on
image_with_detections = np.copy(image_with_noise)

# Get the bounding box for each detected face
for (x,y,w,h) in faces:
    # Add a red bounding box to the detections image
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h), (255,0,0), 3)

# Display the image with the detections
fig = plt.figure(figsize = (8,8))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Noisy Image with Face Detections')
_ = ax1.imshow(image_with_detections)
```

Number of faces detected: 12

Noisy Image with Face Detections



With this added noise we now miss one of the faces!

(IMPLEMENTATION) De-noise this image for better face detection

Time to get your hands dirty: using OpenCV's built in color image de-noising functionality called `fastNlMeansDenoisingColored` - de-noise this image enough so that all the faces in the image are properly detected. Once you have cleaned the image in the next cell, use the cell that follows to run our trained face detector over the cleaned image to check out its detections.

You can find its [official documentation here](#) ([documentation for denoising] (http://docs.opencv.org/trunk/d1/d79/group__photo__denoise.html#ga21abc1c8b0e15f78cd3eff672cb6c476) and a [useful example here](#) (http://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_photo/py_non_local_means/py_non_local_means.html)).

Note: you can keep all parameters except `photo_render` fixed as shown in the second link above. Play around with the value of this parameter - see how it affects the resulting cleaned image.

```
In [11]: ## TODO: Use OpenCV's built in color image de-noising function to clean up our
noisy image!

## Why these parameter values?
## h - Larger than recommended 10 to enhance filtering, but not much higher to
## lessen destruction of detail.
## hColor - Matches h, as recommended.
## templateWindowSize - Kept to recommended 7.
## searchWindowSize - Smaller than recommended 21 since the key areas in the
## image of interest (faces) is relatively small.
dst = cv2.fastNlMeansDenoisingColored(image_with_noise, dst=None, h=17, hColor
=17, \
                                         templateWindowSize=7, searchWindowSize=7
)

denoised_image = dst # your final de-noised image (should be RGB)

## END TODO
```

```
In [12]: ## TODO: Run the face detector on the de-noised image to improve your detections and display the result
gray_denoised = cv2.cvtColor(denoised_image, cv2.COLOR_RGB2GRAY)

# Extract the pre-trained face detector from an xml file
face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_frontalface_default.xml')

# Detect the faces in image
faces = face_cascade.detectMultiScale(gray_denoised, 4, 6)

# Print the number of faces detected in the image
print('Number of faces detected:', len(faces))

image_with_detections = np.copy(denoised_image)

# Get the bounding box for each detected face
for (x,y,w,h) in faces:
    # Add a red bounding box to the detections image
    cv2.rectangle(image_with_detections, (x,y), (x+w,y+h), (255,0,0), 3)

## END TODO

fig = plt.figure(figsize = (8,8))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Less Noisy Image with Face Detections')
_ = ax1.imshow(image_with_detections)
```

Number of faces detected: 12

Less Noisy Image with Face Detections



Step 3: Blur an Image and Perform Edge Detection

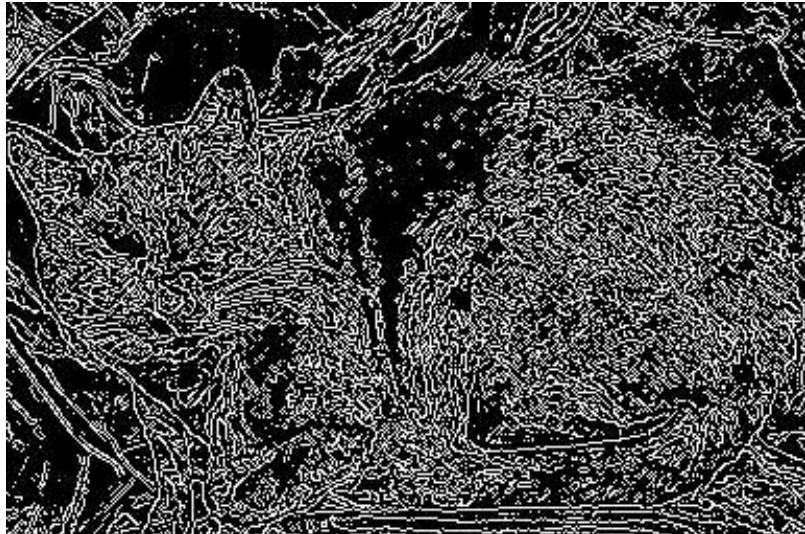
Now that we have developed a simple pipeline for detecting faces using OpenCV - let's start playing around with a few fun things we can do with all those detected faces!

Importance of Blur in Edge Detection

Edge detection is a concept that pops up almost everywhere in computer vision applications, as edge-based features (as well as features built on top of edges) are often some of the best features for e.g., object detection and recognition problems.

Edge detection is a dimension reduction technique - by keeping only the edges of an image we get to throw away a lot of non-discriminating information. And typically the most useful kind of edge-detection is one that preserves only the important, global structures (ignoring local structures that aren't very discriminative). So removing local structures / retaining global structures is a crucial pre-processing step to performing edge detection in an image, and blurring can do just that.

Below is an animated gif showing the result of an edge-detected cat [taken from Wikipedia](https://en.wikipedia.org/wiki/Gaussian_blur#Common_uses) (https://en.wikipedia.org/wiki/Gaussian_blur#Common_uses), where the image is gradually blurred more and more prior to edge detection. When the animation begins you can't quite make out what it's a picture of, but as the animation evolves and local structures are removed via blurring the cat becomes visible in the edge-detected image.



Edge detection is a **convolution** performed on the image itself, and you can read about Canny edge detection on [this OpenCV documentation page](http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html) (http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html).

Canny edge detection

In the cell below we load in a test image, then apply *Canny edge detection* on it. The original image is shown on the left panel of the figure, while the edge-detected version of the image is shown on the right. Notice how the result looks very busy - there are too many little details preserved in the image before it is sent to the edge detector. When applied in computer vision applications, edge detection should preserve *global* structure; doing away with local structures that don't help describe what objects are in the image.

```
In [13]: # Load in the image
image = cv2.imread('images/fawzia.jpg')

# Convert to RGB colorspace
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Convert to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Perform Canny edge detection
edges = cv2.Canny(gray,100,200)

# Dilate the image to amplify edges
edges = cv2.dilate(edges, None)

# Plot the RGB and edge-detected image
fig = plt.figure(figsize = (15,15))
ax1 = fig.add_subplot(121)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Original Image')
_ = ax1.imshow(image)

ax2 = fig.add_subplot(122)
ax2.set_xticks([])
ax2.set_yticks([])

ax2.set_title('Canny Edges')
_ = ax2.imshow(edges, cmap='gray')
```

Original Image



Canny Edges



Without first blurring the image, and removing small, local structures, a lot of irrelevant edge content gets picked up and amplified by the detector (as shown in the right panel above).

(IMPLEMENTATION) Blur the image *then* perform edge detection

In the next cell, you will repeat this experiment - blurring the image first to remove these local structures, so that only the important boundary details remain in the edge-detected image.

Blur the image by using OpenCV's filter2d functionality - which is discussed in [this documentation page](#) (http://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html) - and use an *averaging kernel* of width equal to 4.

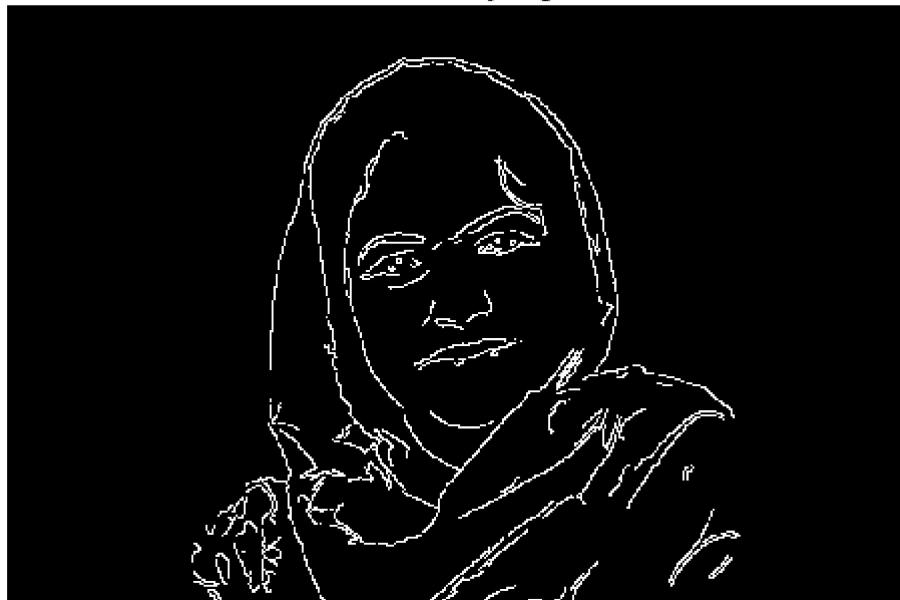
```
In [14]: ## TODO: Blur the test image using OpenCV's filter2d functionality,
# Use an averaging kernel, and a kernel width equal to 4
kernel = np.ones((4,4),np.float32)/16
dst = cv2.filter2D(gray, -1, kernel)
## END TODO
## TODO: Then perform Canny edge detection and display the output
edges = cv2.Canny(dst,100,200)

small_kernel = np.ones((2,2),np.float32)/4
edges = cv2.dilate(edges, small_kernel)
## END TODO

fig = plt.figure(figsize = (8,8))
ax1 = fig.add_subplot(111)

ax1.set_xticks([])
ax1.set_yticks([])
ax1.set_title('Blurred Canny Edges')
_ = ax1.imshow(edges, cmap='gray')
```

Blurred Canny Edges



Step 4: Automatically Hide the Identity of an Individual

If you film something like a documentary or reality TV, you must get permission from every individual shown on film before you can show their face, otherwise you need to blur it out - by blurring the face a lot (so much so that even the global structures are obscured)! This is also true for projects like [Google's StreetView maps](https://www.google.com/streetview/) (<https://www.google.com/streetview/>) - an enormous collection of mapping images taken from a fleet of Google vehicles. Because it would be impossible for Google to get the permission of every single person accidentally captured in one of these images they blur out everyone's faces, the detected images must automatically blur the identity of detected people. Here's a few examples of folks caught in the camera of a Google street view vehicle.



Read in an image to perform identity detection

Let's try this out for ourselves. Use the face detection pipeline built above and what you know about using the `filter2D` to blur an image, and use these in tandem to hide the identity of the person in the following image - loaded in and printed in the next cell.

```
In [15]: # Load in the image
image = cv2.imread('images/gus.jpg')

# Convert the image to RGB colorspace
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Display the image
fig = plt.figure(figsize = (6,6))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])
ax1.set_title('Original Image')
_ = ax1.imshow(image)
```

Original Image



(IMPLEMENTATION) Use blurring to hide the identity of an individual in an image

The idea here is to 1) automatically detect the face in this image, and then 2) blur it out! Make sure to adjust the parameters of the *averaging* blur filter to completely obscure this person's identity.

```
In [16]: def blurout(image):
    ## TODO: Implement face detection

    # Convert the RGB image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

    # Extract the pre-trained face detector from an xml file
    face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_frontalface_default.xml')

    # Detect the faces in image
    faces = face_cascade.detectMultiScale(gray, 1.3, 6)

    ## END TODO

    ## TODO: Blur the bounding box around each detected face using an averaging filter and display the result
    image_with_blurring = np.copy(image)

    kernel = np.ones((64, 64),np.float32)/4096

    # Get the bounding box for each detected face
    for (x,y,w,h) in faces:
        face_square = image[y:y+h, x:x+w]
        # Implement blurring.
        face_square = cv2.filter2D(face_square, -1, kernel)
        image_with_blurring[y:y+face_square.shape[0], x:x+face_square.shape[1]] = face_square

    return image_with_blurring
## END TODO

fig = plt.figure(figsize = (8,8))
ax1 = fig.add_subplot(111)
ax1.set_xticks([])
ax1.set_yticks([])

ax1.set_title('Image with Blurred Face')
_ = ax1.imshow(blurout(image))
```

Image with Blurred Face



(Optional) Build identity protection into your laptop camera

In this optional task you can add identity protection to your laptop camera, using the previously completed code where you added face detection to your laptop camera - and the task above. You should be able to get reasonable results with little parameter tuning - like the one shown in the gif below.



As with the previous video task, to make this perfect would require significant effort - so don't strive for perfection here, strive for reasonable quality.

The next cell contains code a wrapper function called `laptop_camera_identity_hider` that - when called - will activate your laptop's camera. You need to place the relevant face detection and blurring code developed above in this function in order to blur faces entering your laptop camera's field of view.

Before adding anything to the function you can call it to get a hang of how it works - a small window will pop up showing you the live feed from your camera, you can press any key to close this window.

Note: Mac users may find that activating this function kills the kernel of their notebook every once in a while. If this happens to you, just restart your notebook's kernel, activate cell(s) containing any crucial import statements, and you'll be good to go!

In [17]: *### Insert face detection and blurring code into the wrapper below to create a n identity protector on your Laptop!*

```
import cv2
import time

def laptop_camera_go():
    # Create instance of video capturer
    cv2.namedWindow("face detection activated")
    vc = cv2.VideoCapture(0)

    # Try to get the first frame
    if vc.isOpened():
        rval, frame = vc.read()
    else:
        rval = False

    # Keep video stream open
    while rval:
        # Blur out the faces visible to webcam.
        frame = blurout(frame)
        # Plot image from camera with detections marked
        cv2.imshow("face detection activated", frame)

        # Exit functionality - press any key to exit Laptop video
        key = cv2.waitKey(20)
        if key > 0: # Exit by pressing any key
            # Destroy windows
            cv2.destroyAllWindows()

        for i in range (1,5):
            cv2.waitKey(1)
        return

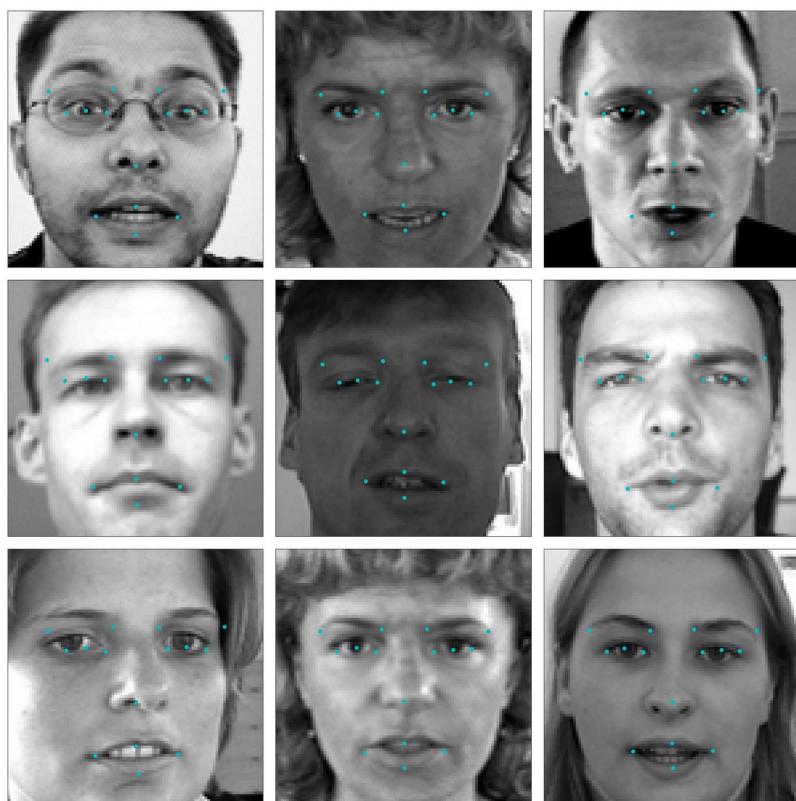
    # Read next frame
    time.sleep(0.05)          # control framerate for computation - default 20 frames per sec
    rval, frame = vc.read()
```

In [18]: *# Run Laptop identity hider.*
#Laptop_camera_go()

Step 5: Create a CNN to Recognize Facial Keypoints

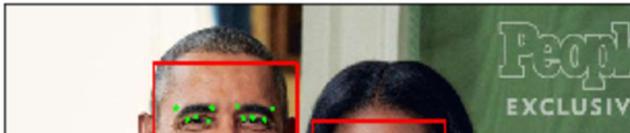
OpenCV is often used in practice with other machine learning and deep learning libraries to produce interesting results. In this stage of the project you will create your own end-to-end pipeline - employing convolutional networks in keras along with OpenCV - to apply a "selfie" filter to streaming video and images.

You will start by creating and then training a convolutional network that can detect facial keypoints in a small dataset of cropped images of human faces. We then guide you towards OpenCV to expanding your detection algorithm to more general images. What are facial keypoints? Let's take a look at some examples.



Facial keypoints (also called facial landmarks) are the small blue-green dots shown on each of the faces in the image above - there are 15 keypoints marked in each image. They mark important areas of the face - the eyes, corners of the mouth, the nose, etc. Facial keypoints can be used in a variety of machine learning applications from face and emotion recognition to commercial applications like the image filters popularized by Snapchat.

Below we illustrate a filter that, using the results of this section, automatically places sunglasses on people in images (using the facial keypoints to place the glasses correctly on each face). Here, the facial keypoints have been colored lime green for visualization purposes.



Make a facial keypoint detector

But first things first: how can we make a facial keypoint detector? Well, at a high level, notice that facial keypoint detection is a *regression problem*. A single face corresponds to a set of 15 facial keypoints (a set of 15 corresponding (x, y) coordinates, i.e., an output point). Because our input data are images, we can employ a *convolutional neural network* to recognize patterns in our images and learn how to identify these keypoint given sets of labeled data.

In order to train a regressor, we need a training set - a set of facial image / facial keypoint pairs to train on. For this we will be using [this dataset from Kaggle](https://www.kaggle.com/c/facial-keypoints-detection/data) (<https://www.kaggle.com/c/facial-keypoints-detection/data>). We've already downloaded this data and placed it in the data directory. Make sure that you have both the *training* and *test* data files. The training dataset contains several thousand 96×96 grayscale images of cropped human faces, along with each face's 15 corresponding facial keypoints (also called landmarks) that have been placed by hand, and recorded in (x, y) coordinates. This wonderful resource also has a substantial testing set, which we will use in tinkering with our convolutional network.

To load in this data, run the Python cell below - notice we will load in both the training and testing sets.

The `load_data` function is in the included `utils.py` file.

```
In [19]: from utils import *

# Load training set
X_train, y_train = load_data()
print("X_train.shape == {}".format(X_train.shape))
print("y_train.shape == {}; y_train.min == {:.3f}; y_train.max == {:.3f}".format(
    y_train.shape, y_train.min(), y_train.max()))

# Load testing set
X_test, _ = load_data(test=True)
print("X_test.shape == {}".format(X_test.shape))
```

Using TensorFlow backend.

```
X_train.shape == (2140, 96, 96, 1)
y_train.shape == (2140, 30); y_train.min == -0.920; y_train.max == 0.996
X_test.shape == (1783, 96, 96, 1)
```

The `load_data` function in `utils.py` originates from this excellent [blog post](http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/) (<http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/>), which you are *strongly* encouraged to read. Please take the time now to review this function. Note how the output values - that is, the coordinates of each set of facial landmarks - have been normalized to take on values in the range $[-1, 1]$, while the pixel values of each input point (a facial image) have been normalized to the range $[0, 1]$.

Note: the original Kaggle dataset contains some images with several missing keypoints. For simplicity, the `load_data` function removes those images with missing labels from the dataset. As an *optional* extension, you are welcome to amend the `load_data` function to include the incomplete data points.

Visualize the Training Data

Execute the code cell below to visualize a subset of the training data.

```
In [20]: import matplotlib.pyplot as plt
%matplotlib inline

fig = plt.figure(figsize=(20,20))
fig.subplots_adjust(left=0, right=1, bottom=0, top=1, hspace=0.05, wspace=0.05)
)
for i in range(9):
    ax = fig.add_subplot(3, 3, i + 1, xticks=[], yticks[])
    plot_data(X_train[i], y_train[i], ax)
```



For each training image, there are two landmarks per eyebrow (**four** total), three per eye (**six** total), **four** for the mouth, and **one** for the tip of the nose.

Review the `plot_data` function in `utils.py` to understand how the 30-dimensional training labels in `y_train` are mapped to facial locations, as this function will prove useful for your pipeline.

(IMPLEMENTATION) Specify the CNN Architecture

In this section, you will specify a neural network for predicting the locations of facial keypoints. Use the code cell below to specify the architecture of your neural network. We have imported some layers that you may find useful for this task, but if you need to use more Keras layers, feel free to import them in the cell.

Your network should accept a 96×96 grayscale image as input, and it should output a vector with 30 entries, corresponding to the predicted (horizontal and vertical) locations of 15 facial keypoints. If you are not sure where to start, you can find some useful starting architectures in [this blog](#)

(<http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/>), but you are not permitted to copy any of the architectures that you find online.

```
In [21]: # Import deep Learning resources from Keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout
from keras.layers import Flatten, Dense
from keras.layers import Activation, BatchNormalization

## TODO: Specify a CNN architecture
# Your model should accept 96x96 pixel grayscale images in
# It should have a fully-connected output layer with 30 values (2 for each facial keypoint)

def build_network(init='glorot_uniform'):
    model = Sequential()
    model.add(Conv2D(filters=16, kernel_size=(3, 3), padding='same', activation='relu', input_shape=X_train.shape[1:], \
                     kernel_initializer=init, name='L1_Conv2D163x3'))
    model.add(MaxPooling2D(pool_size=2, name='L2_MP2'))
    model.add(Conv2D(filters=32, kernel_size=(3, 3), padding='same', activation='relu', \
                     kernel_initializer=init, name='L3_Conv2D323x3'))
    model.add(MaxPooling2D(pool_size=2, name='L4_MP2'))
    model.add(Conv2D(filters=64, kernel_size=(3, 3), padding='same', activation='relu', \
                     kernel_initializer=init, name='L5_Conv2D643x3'))
    model.add(MaxPooling2D(pool_size=2, name='L6_MP2'))
    model.add(Conv2D(filters=128, kernel_size=(3, 3), padding='same', activation='relu', \
                     kernel_initializer=init, name='L7_Conv2D1283x3'))
    model.add(MaxPooling2D(pool_size=2, name='L8_MP2'))
    model.add(Flatten())
    model.add(Dense(units=256, activation='relu', kernel_initializer=init, name='L9_Dense256'))
    model.add(Dropout(rate=0.5))
    model.add(Dense(units=30, kernel_initializer=init, name='L10_Dense30'))
    return model

## END TODO
model = build_network()
# Summarize the model
model.summary()
```

Layer (type)	Output Shape	Param #
<hr/>		
L1_Conv2D163x3 (Conv2D)	(None, 96, 96, 16)	160
L2_MP2 (MaxPooling2D)	(None, 48, 48, 16)	0
L3_Conv2D323x3 (Conv2D)	(None, 48, 48, 32)	4640
L4_MP2 (MaxPooling2D)	(None, 24, 24, 32)	0
L5_Conv2D643x3 (Conv2D)	(None, 24, 24, 64)	18496
L6_MP2 (MaxPooling2D)	(None, 12, 12, 64)	0
L7_Conv2D1283x3 (Conv2D)	(None, 12, 12, 128)	73856
L8_MP2 (MaxPooling2D)	(None, 6, 6, 128)	0
flatten_1 (Flatten)	(None, 4608)	0
L9_Dense256 (Dense)	(None, 256)	1179904
dropout_1 (Dropout)	(None, 256)	0
L10_Dense30 (Dense)	(None, 30)	7710
<hr/>		
Total params: 1,284,766		
Trainable params: 1,284,766		
Non-trainable params: 0		

Step 6: Compile and Train the Model

After specifying your architecture, you'll need to compile and train the model to detect facial keypoints'

(IMPLEMENTATION) Compile and Train the Model

Use the `compile` method (<https://keras.io/models/sequential/#sequential-model-methods>) to configure the learning process. Experiment with your choice of `optimizer` (<https://keras.io/optimizers/>); you may have some ideas about which will work best (SGD vs. RMSprop, etc), but take the time to empirically verify your theories.

Use the `fit` method (<https://keras.io/models/sequential/#sequential-model-methods>) to train the model. Break off a validation set by setting `validation_split=0.2`. Save the returned History object in the `history` variable.

Experiment with your model to minimize the validation loss (measured as mean squared error). A very good model will achieve about 0.0015 loss (though it's possible to do even better). When you have finished training, `save your model` (<https://keras.io/getting-started/faq/#how-can-i-save-a-keras-model>) as an HDF5 file with file path `my_model.h5`.

```
In [22]: from keras.optimizers import SGD, RMSprop, Adagrad, Adadelta, Adam, Adama
x, Nadam
from keras.callbacks import ModelCheckpoint

# Implement a simple grid search.
def gridsearch(build_fn, hyperp, batch, epochs, valspli, data):
    X_train, y_train = data
    histories = {}
    cnns = {}

    count = 1
    total = len(hyperp['optimizers'])*len(hyperp['learnrate'])*len(hyperp
    ['initialization'])
    # Run the search.
    for opt in hyperp['optimizers']:
        for lr in hyperp['learnrate']:
            for init in hyperp['initialization']:
                trainer = opt(lr=lr)
                name = 'model_' + trainer.__class__.__name__ + str(lr) +
init
                histories[name] = []
                cnns[name] = []
                print('Training model with spec {} (Combo: {}/{}).format
(name, count, total))
                count += 1
                checkpointer = ModelCheckpoint(filepath='models/' + name
+ '.h5', save_best_only=True)
                for it in range(hyperp['iterations']):
                    print('Iteration {}/{}'.format(it+1, hyperp['iteratio
ns']))
                    model = build_model(trainer, init)
                    histories[name].append(model.fit(X_train, y_train,
                                         validation_split=valspli
t, epochs=epochs,
                                         batch_size=batch,
                                         callbacks=[checkpointer
], verbose=1))
                    cnns[name].append(model)
```

```
    print('\n')

# Average out the iterations.
averages = {}
for model in histories.keys():
    hist = []
    for it in range(hyperp['iterations']):
        hist.append(histories[model][it].history['val_loss'])
    hist = np.array(hist)
    avg = list(np.average(hist, axis=0))
    averages[model] = avg

# Save only the best combination, considering only validation loss.
# Which combo does best on average.
best_model, best_loss = None, 99999
for model in averages.keys():
    min_loss = np.min(averages[model])
    if min_loss < best_loss:
        best_model, best_loss = model, min_loss

# Which iteration did the best?
best_it, best_loss = None, 99999
for it in range(len(histories[best_model])):
    min_loss = np.min(histories[best_model][it].history['val_loss'])
    if min_loss < best_loss:
        best_it, best_loss = it, min_loss
best_name = best_model
best_hist = histories[best_model][best_it]
best_model = cnns[best_model][best_it]

return best_hist, best_name, best_model
```

```
In [23]: # Search for the best model using 30 epochs, 1 iteration for each combination.
batch_size = X_train.shape[0]//20
test2val = 0.2
epochs = 30

iterations = 1
optimizers = [Adagrad, Adadelta, Adam, Adamax, Nadam]
learningrates = [0.002, 0.001, 0.0009]
init_mode = ['glorot_normal', 'glorot_uniform', 'he_normal', 'he_uniform']

def build_model(opt, init):
    network = build_network(init=init)
    network.compile(loss='mse', optimizer=opt)
    return network

data = X_train, y_train
params = dict(optimizers=optimizers, learnrate=learningrates, initialization=init_mode, iterations=iterations)
best_hist, model_name, best_model = gridsearch(build_fn=build_model, hyperp=params, batch=batch_size,
                                                epochs=epochs, valsplit=test2val, data=data)
print('The best model is {}'.format(model_name))
```

Training model with spec model_Adagrad0.002glorot_normal (Combo: 1/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 0.0693 -
val_loss: 0.0123
Epoch 2/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0299
- val_loss: 0.0095
Epoch 3/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0258
- val_loss: 0.0106
Epoch 4/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0233
- val_loss: 0.0084
Epoch 5/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0217
- val_loss: 0.0059
Epoch 6/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0199
- val_loss: 0.0066
Epoch 7/30
1712/1712 [=====] - 1s 633us/step - loss: 0.0185
- val_loss: 0.0081
Epoch 8/30
1712/1712 [=====] - 1s 632us/step - loss: 0.0178
- val_loss: 0.0098
Epoch 9/30
1712/1712 [=====] - 1s 666us/step - loss: 0.0170
- val_loss: 0.0046
Epoch 10/30
1712/1712 [=====] - 1s 632us/step - loss: 0.0161
- val_loss: 0.0057
Epoch 11/30
1712/1712 [=====] - 1s 632us/step - loss: 0.0158
- val_loss: 0.0054
Epoch 12/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0151
- val_loss: 0.0049
Epoch 13/30
1712/1712 [=====] - 1s 633us/step - loss: 0.0146
- val_loss: 0.0049
Epoch 14/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0141
- val_loss: 0.0042
Epoch 15/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0143
- val_loss: 0.0046
Epoch 16/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0136
- val_loss: 0.0052
Epoch 17/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0130
- val_loss: 0.0046
Epoch 18/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0127
- val_loss: 0.0040

```
Epoch 19/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0129
- val_loss: 0.0044
Epoch 20/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0122
- val_loss: 0.0047
Epoch 21/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0121
- val_loss: 0.0045
Epoch 22/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0116
- val_loss: 0.0057
Epoch 23/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0124
- val_loss: 0.0039
Epoch 24/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0114
- val_loss: 0.0047
Epoch 25/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0112
- val_loss: 0.0043
Epoch 26/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0112
- val_loss: 0.0041
Epoch 27/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0113
- val_loss: 0.0046
Epoch 28/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0108
- val_loss: 0.0060
Epoch 29/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0108
- val_loss: 0.0046
Epoch 30/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0105
- val_loss: 0.0046
```

```
Training model with spec model_Adagrad0.002glorot_uniform (Combo: 2/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 1s 758us/step - loss: 0.0529
- val_loss: 0.0106
Epoch 2/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0266
- val_loss: 0.0075
Epoch 3/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0233
- val_loss: 0.0073
Epoch 4/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0204
- val_loss: 0.0064
Epoch 5/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0182
- val_loss: 0.0067
Epoch 6/30
```

```
1712/1712 [=====] - 1s 670us/step - loss: 0.0170
- val_loss: 0.0057
Epoch 7/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0161
- val_loss: 0.0062
Epoch 8/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0149
- val_loss: 0.0048
Epoch 9/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0145
- val_loss: 0.0053
Epoch 10/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0135
- val_loss: 0.0043
Epoch 11/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0135
- val_loss: 0.0041
Epoch 12/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0133
- val_loss: 0.0060
Epoch 13/30
1712/1712 [=====] - 1s 631us/step - loss: 0.0126
- val_loss: 0.0044
Epoch 14/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0121
- val_loss: 0.0041
Epoch 15/30
1712/1712 [=====] - 1s 632us/step - loss: 0.0121
- val_loss: 0.0049
Epoch 16/30
1712/1712 [=====] - 1s 631us/step - loss: 0.0116
- val_loss: 0.0045
Epoch 17/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0112
- val_loss: 0.0042
Epoch 18/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0110
- val_loss: 0.0040
Epoch 19/30
1712/1712 [=====] - 1s 631us/step - loss: 0.0109
- val_loss: 0.0051
Epoch 20/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0105
- val_loss: 0.0037
Epoch 21/30
1712/1712 [=====] - 1s 633us/step - loss: 0.0105
- val_loss: 0.0044
Epoch 22/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0101
- val_loss: 0.0048
Epoch 23/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0100
- val_loss: 0.0036
Epoch 24/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0098
- val_loss: 0.0034
Epoch 25/30
```

```
1712/1712 [=====] - 1s 670us/step - loss: 0.0096
- val_loss: 0.0034
Epoch 26/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0097
- val_loss: 0.0034
Epoch 27/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0092
- val_loss: 0.0036
Epoch 28/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0091
- val_loss: 0.0033
Epoch 29/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0091
- val_loss: 0.0043
Epoch 30/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0087
- val_loss: 0.0032
```

```
Training model with spec model_Adagrad0.002he_normal (Combo: 3/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 1s 769us/step - loss: 0.1898
- val_loss: 0.0199
Epoch 2/30
1712/1712 [=====] - 1s 663us/step - loss: 0.0424
- val_loss: 0.0179
Epoch 3/30
1712/1712 [=====] - 1s 665us/step - loss: 0.0383
- val_loss: 0.0115
Epoch 4/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0339
- val_loss: 0.0137
Epoch 5/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0309
- val_loss: 0.0105
Epoch 6/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0298
- val_loss: 0.0083
Epoch 7/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0280
- val_loss: 0.0103
Epoch 8/30
1712/1712 [=====] - 1s 632us/step - loss: 0.0261
- val_loss: 0.0129
Epoch 9/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0254
- val_loss: 0.0094
Epoch 10/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0243
- val_loss: 0.0083
Epoch 11/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0230
- val_loss: 0.0081
Epoch 12/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0229
```

```
- val_loss: 0.0074
Epoch 13/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0213
- val_loss: 0.0094
Epoch 14/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0215
- val_loss: 0.0108
Epoch 15/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0203
- val_loss: 0.0063
Epoch 16/30
1712/1712 [=====] - 1s 633us/step - loss: 0.0196
- val_loss: 0.0070
Epoch 17/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0198
- val_loss: 0.0062
Epoch 18/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0183
- val_loss: 0.0061
Epoch 19/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0187
- val_loss: 0.0051
Epoch 20/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0176
- val_loss: 0.0055
Epoch 21/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0174
- val_loss: 0.0050
Epoch 22/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0171
- val_loss: 0.0053
Epoch 23/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0171
- val_loss: 0.0056
Epoch 24/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0167
- val_loss: 0.0049
Epoch 25/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0162
- val_loss: 0.0045
Epoch 26/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0161
- val_loss: 0.0062
Epoch 27/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0156
- val_loss: 0.0079
Epoch 28/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0153
- val_loss: 0.0052
Epoch 29/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0149
- val_loss: 0.0096
Epoch 30/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0150
- val_loss: 0.0042
```

```
Training model with spec model_Adagrad0.002he_uniform (Combo: 4/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 1s 841us/step - loss: 0.6810
- val_loss: 0.0513
Epoch 2/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0690
- val_loss: 0.0410
Epoch 3/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0615
- val_loss: 0.0294
Epoch 4/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0564
- val_loss: 0.0248
Epoch 5/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0529
- val_loss: 0.0216
Epoch 6/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0493
- val_loss: 0.0197
Epoch 7/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0466
- val_loss: 0.0173
Epoch 8/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0444
- val_loss: 0.0170
Epoch 9/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0428
- val_loss: 0.0153
Epoch 10/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0417
- val_loss: 0.0179
Epoch 11/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0396
- val_loss: 0.0158
Epoch 12/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0386
- val_loss: 0.0144
Epoch 13/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0378
- val_loss: 0.0169
Epoch 14/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0362
- val_loss: 0.0120
Epoch 15/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0354
- val_loss: 0.0126
Epoch 16/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0349
- val_loss: 0.0148
Epoch 17/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0331
- val_loss: 0.0104
Epoch 18/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0327
- val_loss: 0.0138
```

```
Epoch 19/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0321
- val_loss: 0.0127
Epoch 20/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0316
- val_loss: 0.0120
Epoch 21/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0306
- val_loss: 0.0110
Epoch 22/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0302
- val_loss: 0.0110
Epoch 23/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0287
- val_loss: 0.0108
Epoch 24/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0290
- val_loss: 0.0085
Epoch 25/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0284
- val_loss: 0.0076
Epoch 26/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0281
- val_loss: 0.0090
Epoch 27/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0274
- val_loss: 0.0106
Epoch 28/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0283
- val_loss: 0.0079
Epoch 29/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0277
- val_loss: 0.0082
Epoch 30/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0261
- val_loss: 0.0109
```

```
Training model with spec model_Adagrad0.001glorot_normal (Combo: 5/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 1s 808us/step - loss: 0.0479
- val_loss: 0.0092
Epoch 2/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0296
- val_loss: 0.0130
Epoch 3/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0252
- val_loss: 0.0086
Epoch 4/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0230
- val_loss: 0.0081
Epoch 5/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0209
- val_loss: 0.0071
Epoch 6/30
```

```
1712/1712 [=====] - 1s 672us/step - loss: 0.0203
- val_loss: 0.0054
Epoch 7/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0187
- val_loss: 0.0059
Epoch 8/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0175
- val_loss: 0.0080
Epoch 9/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0177
- val_loss: 0.0047
Epoch 10/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0166
- val_loss: 0.0052
Epoch 11/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0163
- val_loss: 0.0055
Epoch 12/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0155
- val_loss: 0.0076
Epoch 13/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0153
- val_loss: 0.0072
Epoch 14/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0147
- val_loss: 0.0065
Epoch 15/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0146
- val_loss: 0.0072
Epoch 16/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0144
- val_loss: 0.0071
Epoch 17/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0138
- val_loss: 0.0051
Epoch 18/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0137
- val_loss: 0.0059
Epoch 19/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0134
- val_loss: 0.0060
Epoch 20/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0131
- val_loss: 0.0052
Epoch 21/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0130
- val_loss: 0.0051
Epoch 22/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0128
- val_loss: 0.0041
Epoch 23/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0124
- val_loss: 0.0041
Epoch 24/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0124
- val_loss: 0.0051
Epoch 25/30
```

```
1712/1712 [=====] - 1s 642us/step - loss: 0.0122
- val_loss: 0.0048
Epoch 26/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0119
- val_loss: 0.0046
Epoch 27/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0117
- val_loss: 0.0059
Epoch 28/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0120
- val_loss: 0.0038
Epoch 29/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0119
- val_loss: 0.0045
Epoch 30/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0117
- val_loss: 0.0039
```

```
Training model with spec model_Adagrad0.001glorot_uniform (Combo: 6/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 1s 824us/step - loss: 0.0506
- val_loss: 0.0160
Epoch 2/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0327
- val_loss: 0.0101
Epoch 3/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0289
- val_loss: 0.0134
Epoch 4/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0263
- val_loss: 0.0068
Epoch 5/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0246
- val_loss: 0.0082
Epoch 6/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0227
- val_loss: 0.0110
Epoch 7/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0215
- val_loss: 0.0150
Epoch 8/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0208
- val_loss: 0.0064
Epoch 9/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0196
- val_loss: 0.0064
Epoch 10/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0186
- val_loss: 0.0068
Epoch 11/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0181
- val_loss: 0.0105
Epoch 12/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0182
```

```
- val_loss: 0.0059
Epoch 13/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0173
- val_loss: 0.0060
Epoch 14/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0164
- val_loss: 0.0061
Epoch 15/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0163
- val_loss: 0.0042
Epoch 16/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0160
- val_loss: 0.0052
Epoch 17/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0156
- val_loss: 0.0063
Epoch 18/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0152
- val_loss: 0.0060
Epoch 19/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0148
- val_loss: 0.0045
Epoch 20/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0144
- val_loss: 0.0044
Epoch 21/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0144
- val_loss: 0.0043
Epoch 22/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0143
- val_loss: 0.0063
Epoch 23/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0141
- val_loss: 0.0054
Epoch 24/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0138
- val_loss: 0.0041
Epoch 25/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0132
- val_loss: 0.0041
Epoch 26/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0135
- val_loss: 0.0042
Epoch 27/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0132
- val_loss: 0.0044
Epoch 28/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0131
- val_loss: 0.0045
Epoch 29/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0128
- val_loss: 0.0044
Epoch 30/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0126
- val_loss: 0.0040
```

```
Training model with spec model_Adagrad0.001he_normal (Combo: 7/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 1s 845us/step - loss: 0.2393
- val_loss: 0.0277
Epoch 2/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0576
- val_loss: 0.0250
Epoch 3/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0532
- val_loss: 0.0248
Epoch 4/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0493
- val_loss: 0.0181
Epoch 5/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0457
- val_loss: 0.0194
Epoch 6/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0436
- val_loss: 0.0220
Epoch 7/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0422
- val_loss: 0.0171
Epoch 8/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0400
- val_loss: 0.0142
Epoch 9/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0380
- val_loss: 0.0124
Epoch 10/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0375
- val_loss: 0.0158
Epoch 11/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0369
- val_loss: 0.0119
Epoch 12/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0353
- val_loss: 0.0122
Epoch 13/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0349
- val_loss: 0.0134
Epoch 14/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0328
- val_loss: 0.0129
Epoch 15/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0326
- val_loss: 0.0124
Epoch 16/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0314
- val_loss: 0.0117
Epoch 17/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0311
- val_loss: 0.0129
Epoch 18/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0311
- val_loss: 0.0096
```

```
Epoch 19/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0293
- val_loss: 0.0099
Epoch 20/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0289
- val_loss: 0.0090
Epoch 21/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0283
- val_loss: 0.0106
Epoch 22/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0278
- val_loss: 0.0127
Epoch 23/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0271
- val_loss: 0.0102
Epoch 24/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0272
- val_loss: 0.0134
Epoch 25/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0268
- val_loss: 0.0088
Epoch 26/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0260
- val_loss: 0.0099
Epoch 27/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0258
- val_loss: 0.0127
Epoch 28/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0252
- val_loss: 0.0074
Epoch 29/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0245
- val_loss: 0.0074
Epoch 30/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0245
- val_loss: 0.0071
```

```
Training model with spec model_Adagrad0.001he_uniform (Combo: 8/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 1s 856us/step - loss: 0.4411
- val_loss: 0.0421
Epoch 2/30
1712/1712 [=====] - 1s 664us/step - loss: 0.0715
- val_loss: 0.0310
Epoch 3/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0636
- val_loss: 0.0314
Epoch 4/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0585
- val_loss: 0.0247
Epoch 5/30
1712/1712 [=====] - 1s 633us/step - loss: 0.0551
- val_loss: 0.0252
Epoch 6/30
```

```
1712/1712 [=====] - 1s 673us/step - loss: 0.0529
- val_loss: 0.0222
Epoch 7/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0505
- val_loss: 0.0206
Epoch 8/30
1712/1712 [=====] - 1s 633us/step - loss: 0.0490
- val_loss: 0.0217
Epoch 9/30
1712/1712 [=====] - 1s 634us/step - loss: 0.0471
- val_loss: 0.0251
Epoch 10/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0472
- val_loss: 0.0171
Epoch 11/30
1712/1712 [=====] - 1s 633us/step - loss: 0.0452
- val_loss: 0.0178
Epoch 12/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0441
- val_loss: 0.0217
Epoch 13/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0433
- val_loss: 0.0208
Epoch 14/30
1712/1712 [=====] - 1s 665us/step - loss: 0.0426
- val_loss: 0.0154
Epoch 15/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0415
- val_loss: 0.0159
Epoch 16/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0403
- val_loss: 0.0173
Epoch 17/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0395
- val_loss: 0.0164
Epoch 18/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0387
- val_loss: 0.0143
Epoch 19/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0378
- val_loss: 0.0194
Epoch 20/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0381
- val_loss: 0.0167
Epoch 21/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0366
- val_loss: 0.0144
Epoch 22/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0362
- val_loss: 0.0116
Epoch 23/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0358
- val_loss: 0.0168
Epoch 24/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0357
- val_loss: 0.0155
Epoch 25/30
```

```
1712/1712 [=====] - 1s 638us/step - loss: 0.0349
- val_loss: 0.0126
Epoch 26/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0336
- val_loss: 0.0124
Epoch 27/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0338
- val_loss: 0.0178
Epoch 28/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0336
- val_loss: 0.0117
Epoch 29/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0325
- val_loss: 0.0105
Epoch 30/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0324
- val_loss: 0.0108
```

```
Training model with spec model_Adagrad0.0009glorot_normal (Combo: 9/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 945us/step - loss: 0.0525
- val_loss: 0.0139
Epoch 2/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0326
- val_loss: 0.0088
Epoch 3/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0296
- val_loss: 0.0098
Epoch 4/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0263
- val_loss: 0.0085
Epoch 5/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0245
- val_loss: 0.0088
Epoch 6/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0231
- val_loss: 0.0067
Epoch 7/30
1712/1712 [=====] - 1s 644us/step - loss: 0.0221
- val_loss: 0.0087
Epoch 8/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0204
- val_loss: 0.0073
Epoch 9/30
1712/1712 [=====] - 1s 644us/step - loss: 0.0203
- val_loss: 0.0071
Epoch 10/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0190
- val_loss: 0.0068
Epoch 11/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0185
- val_loss: 0.0069
Epoch 12/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0179
```

```
- val_loss: 0.0072
Epoch 13/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0177
- val_loss: 0.0084
Epoch 14/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0174
- val_loss: 0.0070
Epoch 15/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0166
- val_loss: 0.0073
Epoch 16/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0165
- val_loss: 0.0054
Epoch 17/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0158
- val_loss: 0.0056
Epoch 18/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0159
- val_loss: 0.0047
Epoch 19/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0152
- val_loss: 0.0067
Epoch 20/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0150
- val_loss: 0.0091
Epoch 21/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0152
- val_loss: 0.0050
Epoch 22/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0146
- val_loss: 0.0050
Epoch 23/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0141
- val_loss: 0.0044
Epoch 24/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0140
- val_loss: 0.0045
Epoch 25/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0140
- val_loss: 0.0043
Epoch 26/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0138
- val_loss: 0.0044
Epoch 27/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0135
- val_loss: 0.0045
Epoch 28/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0134
- val_loss: 0.0055
Epoch 29/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0131
- val_loss: 0.0044
Epoch 30/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0131
- val_loss: 0.0051
```

```
Training model with spec model_Adagrad0.0009glorot_uniform (Combo: 10/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 891us/step - loss: 0.0504
- val_loss: 0.0114
Epoch 2/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0302
- val_loss: 0.0081
Epoch 3/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0269
- val_loss: 0.0071
Epoch 4/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0243
- val_loss: 0.0091
Epoch 5/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0228
- val_loss: 0.0058
Epoch 6/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0212
- val_loss: 0.0070
Epoch 7/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0206
- val_loss: 0.0067
Epoch 8/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0192
- val_loss: 0.0062
Epoch 9/30
1712/1712 [=====] - 1s 644us/step - loss: 0.0186
- val_loss: 0.0062
Epoch 10/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0181
- val_loss: 0.0050
Epoch 11/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0174
- val_loss: 0.0068
Epoch 12/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0171
- val_loss: 0.0055
Epoch 13/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0165
- val_loss: 0.0051
Epoch 14/30
1712/1712 [=====] - 1s 644us/step - loss: 0.0164
- val_loss: 0.0058
Epoch 15/30
1712/1712 [=====] - 1s 644us/step - loss: 0.0159
- val_loss: 0.0061
Epoch 16/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0153
- val_loss: 0.0045
Epoch 17/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0156
- val_loss: 0.0070
Epoch 18/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0147
- val_loss: 0.0057
```

```
Epoch 19/30
1712/1712 [=====] - 1s 645us/step - loss: 0.0146
- val_loss: 0.0054
Epoch 20/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0142
- val_loss: 0.0046
Epoch 21/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0140
- val_loss: 0.0052
Epoch 22/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0140
- val_loss: 0.0042
Epoch 23/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0140
- val_loss: 0.0047
Epoch 24/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0134
- val_loss: 0.0041
Epoch 25/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0134
- val_loss: 0.0049
Epoch 26/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0130
- val_loss: 0.0040
Epoch 27/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0128
- val_loss: 0.0040
Epoch 28/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0126
- val_loss: 0.0040
Epoch 29/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0125
- val_loss: 0.0044
Epoch 30/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0124
- val_loss: 0.0041
```

```
Training model with spec model_Adagrad0.0009he_normal (Combo: 11/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 917us/step - loss: 0.1873
- val_loss: 0.0277
Epoch 2/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0555
- val_loss: 0.0252
Epoch 3/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0500
- val_loss: 0.0201
Epoch 4/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0476
- val_loss: 0.0174
Epoch 5/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0440
- val_loss: 0.0179
Epoch 6/30
```

```
1712/1712 [=====] - 1s 640us/step - loss: 0.0423
- val_loss: 0.0183
Epoch 7/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0408
- val_loss: 0.0144
Epoch 8/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0397
- val_loss: 0.0174
Epoch 9/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0385
- val_loss: 0.0132
Epoch 10/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0365
- val_loss: 0.0164
Epoch 11/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0352
- val_loss: 0.0105
Epoch 12/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0347
- val_loss: 0.0119
Epoch 13/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0340
- val_loss: 0.0141
Epoch 14/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0329
- val_loss: 0.0121
Epoch 15/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0325
- val_loss: 0.0118
Epoch 16/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0317
- val_loss: 0.0124
Epoch 17/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0305
- val_loss: 0.0124
Epoch 18/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0302
- val_loss: 0.0097
Epoch 19/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0291
- val_loss: 0.0104
Epoch 20/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0287
- val_loss: 0.0103
Epoch 21/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0279
- val_loss: 0.0100
Epoch 22/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0275
- val_loss: 0.0078
Epoch 23/30
1712/1712 [=====] - 1s 635us/step - loss: 0.0270
- val_loss: 0.0149
Epoch 24/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0268
- val_loss: 0.0073
Epoch 25/30
```

```
1712/1712 [=====] - 1s 635us/step - loss: 0.0262
- val_loss: 0.0090
Epoch 26/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0258
- val_loss: 0.0083
Epoch 27/30
1712/1712 [=====] - 1s 637us/step - loss: 0.0250
- val_loss: 0.0075
Epoch 28/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0250
- val_loss: 0.0064
Epoch 29/30
1712/1712 [=====] - 1s 636us/step - loss: 0.0250
- val_loss: 0.0106
Epoch 30/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0244
- val_loss: 0.0066
```

```
Training model with spec model_Adagrad0.0009he_uniform (Combo: 12/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 933us/step - loss: 1.1602
- val_loss: 0.0619
Epoch 2/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0928
- val_loss: 0.0551
Epoch 3/30
1712/1712 [=====] - 1s 666us/step - loss: 0.0870
- val_loss: 0.0547
Epoch 4/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0811
- val_loss: 0.0479
Epoch 5/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0778
- val_loss: 0.0437
Epoch 6/30
1712/1712 [=====] - 1s 666us/step - loss: 0.0756
- val_loss: 0.0394
Epoch 7/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0721
- val_loss: 0.0400
Epoch 8/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0701
- val_loss: 0.0372
Epoch 9/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0686
- val_loss: 0.0342
Epoch 10/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0683
- val_loss: 0.0325
Epoch 11/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0677
- val_loss: 0.0311
Epoch 12/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0654
```

```
- val_loss: 0.0368
Epoch 13/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0645
- val_loss: 0.0309
Epoch 14/30
1712/1712 [=====] - 1s 641us/step - loss: 0.0628
- val_loss: 0.0315
Epoch 15/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0618
- val_loss: 0.0312
Epoch 16/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0618
- val_loss: 0.0295
Epoch 17/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0608
- val_loss: 0.0280
Epoch 18/30
1712/1712 [=====] - 1s 642us/step - loss: 0.0585
- val_loss: 0.0295
Epoch 19/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0587
- val_loss: 0.0293
Epoch 20/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0574
- val_loss: 0.0291
Epoch 21/30
1712/1712 [=====] - 1s 640us/step - loss: 0.0564
- val_loss: 0.0297
Epoch 22/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0561
- val_loss: 0.0269
Epoch 23/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0551
- val_loss: 0.0247
Epoch 24/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0533
- val_loss: 0.0241
Epoch 25/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0527
- val_loss: 0.0239
Epoch 26/30
1712/1712 [=====] - 1s 638us/step - loss: 0.0525
- val_loss: 0.0245
Epoch 27/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0508
- val_loss: 0.0215
Epoch 28/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0506
- val_loss: 0.0204
Epoch 29/30
1712/1712 [=====] - 1s 639us/step - loss: 0.0504
- val_loss: 0.0209
Epoch 30/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0493
- val_loss: 0.0234
```

```
Training model with spec model_Adadelta0.002glorot_normal (Combo: 13/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.1550 -
val_loss: 0.1572
Epoch 2/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1544
- val_loss: 0.1566
Epoch 3/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1540
- val_loss: 0.1560
Epoch 4/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1534
- val_loss: 0.1554
Epoch 5/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1528
- val_loss: 0.1548
Epoch 6/30
1712/1712 [=====] - 1s 691us/step - loss: 0.1521
- val_loss: 0.1542
Epoch 7/30
1712/1712 [=====] - 1s 688us/step - loss: 0.1516
- val_loss: 0.1536
Epoch 8/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1510
- val_loss: 0.1530
Epoch 9/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1504
- val_loss: 0.1524
Epoch 10/30
1712/1712 [=====] - 1s 712us/step - loss: 0.1500
- val_loss: 0.1518
Epoch 11/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1493
- val_loss: 0.1512
Epoch 12/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1487
- val_loss: 0.1506
Epoch 13/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1481
- val_loss: 0.1500
Epoch 14/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1476
- val_loss: 0.1493
Epoch 15/30
1712/1712 [=====] - 1s 696us/step - loss: 0.1468
- val_loss: 0.1487
Epoch 16/30
1712/1712 [=====] - 1s 703us/step - loss: 0.1464
- val_loss: 0.1480
Epoch 17/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1457
- val_loss: 0.1473
Epoch 18/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1448
- val_loss: 0.1466
```

```
Epoch 19/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1443
- val_loss: 0.1458
Epoch 20/30
1712/1712 [=====] - 1s 694us/step - loss: 0.1437
- val_loss: 0.1451
Epoch 21/30
1712/1712 [=====] - 1s 701us/step - loss: 0.1427
- val_loss: 0.1443
Epoch 22/30
1712/1712 [=====] - 1s 708us/step - loss: 0.1421
- val_loss: 0.1435
Epoch 23/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1412
- val_loss: 0.1427
Epoch 24/30
1712/1712 [=====] - 1s 707us/step - loss: 0.1406
- val_loss: 0.1418
Epoch 25/30
1712/1712 [=====] - 1s 698us/step - loss: 0.1396
- val_loss: 0.1409
Epoch 26/30
1712/1712 [=====] - 1s 706us/step - loss: 0.1388
- val_loss: 0.1400
Epoch 27/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1380
- val_loss: 0.1390
Epoch 28/30
1712/1712 [=====] - 1s 702us/step - loss: 0.1370
- val_loss: 0.1380
Epoch 29/30
1712/1712 [=====] - 1s 703us/step - loss: 0.1360
- val_loss: 0.1370
Epoch 30/30
1712/1712 [=====] - 1s 693us/step - loss: 0.1352
- val_loss: 0.1360
```

```
Training model with spec model_Adadelta0.002glorot_uniform (Combo: 14/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.1448 - val_loss: 0.1434
Epoch 2/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1430 - val_loss: 0.1416
Epoch 3/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1412 - val_loss: 0.1397
Epoch 4/30
1712/1712 [=====] - 1s 694us/step - loss: 0.1396 - val_loss: 0.1377
Epoch 5/30
1712/1712 [=====] - 1s 708us/step - loss: 0.1377 - val_loss: 0.1358
Epoch 6/30
```

```
1712/1712 [=====] - 1s 716us/step - loss: 0.1357
- val_loss: 0.1339
Epoch 7/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1339
- val_loss: 0.1319
Epoch 8/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1317
- val_loss: 0.1298
Epoch 9/30
1712/1712 [=====] - 1s 697us/step - loss: 0.1300
- val_loss: 0.1278
Epoch 10/30
1712/1712 [=====] - 1s 683us/step - loss: 0.1280
- val_loss: 0.1256
Epoch 11/30
1712/1712 [=====] - 1s 694us/step - loss: 0.1258
- val_loss: 0.1234
Epoch 12/30
1712/1712 [=====] - 1s 681us/step - loss: 0.1241
- val_loss: 0.1212
Epoch 13/30
1712/1712 [=====] - 1s 697us/step - loss: 0.1219
- val_loss: 0.1189
Epoch 14/30
1712/1712 [=====] - 1s 683us/step - loss: 0.1199
- val_loss: 0.1165
Epoch 15/30
1712/1712 [=====] - 1s 682us/step - loss: 0.1175
- val_loss: 0.1140
Epoch 16/30
1712/1712 [=====] - 1s 681us/step - loss: 0.1154
- val_loss: 0.1115
Epoch 17/30
1712/1712 [=====] - 1s 695us/step - loss: 0.1132
- val_loss: 0.1089
Epoch 18/30
1712/1712 [=====] - 1s 708us/step - loss: 0.1102
- val_loss: 0.1062
Epoch 19/30
1712/1712 [=====] - 1s 682us/step - loss: 0.1085
- val_loss: 0.1035
Epoch 20/30
1712/1712 [=====] - 1s 693us/step - loss: 0.1058
- val_loss: 0.1007
Epoch 21/30
1712/1712 [=====] - 1s 696us/step - loss: 0.1031
- val_loss: 0.0979
Epoch 22/30
1712/1712 [=====] - 1s 677us/step - loss: 0.1009
- val_loss: 0.0950
Epoch 23/30
1712/1712 [=====] - 1s 689us/step - loss: 0.0980
- val_loss: 0.0920
Epoch 24/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0952
- val_loss: 0.0890
Epoch 25/30
```

```
1712/1712 [=====] - 1s 676us/step - loss: 0.0930
- val_loss: 0.0860
Epoch 26/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0901
- val_loss: 0.0830
Epoch 27/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0875
- val_loss: 0.0800
Epoch 28/30
1712/1712 [=====] - 1s 689us/step - loss: 0.0854
- val_loss: 0.0770
Epoch 29/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0829
- val_loss: 0.0740
Epoch 30/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0800
- val_loss: 0.0710
```

```
Training model with spec model_Adadelta0.002he_normal (Combo: 15/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.5390 - 
val_loss: 0.2836
Epoch 2/30
1712/1712 [=====] - 1s 713us/step - loss: 0.4973
- val_loss: 0.2561
Epoch 3/30
1712/1712 [=====] - 1s 711us/step - loss: 0.4635
- val_loss: 0.2321
Epoch 4/30
1712/1712 [=====] - 1s 704us/step - loss: 0.4319
- val_loss: 0.2103
Epoch 5/30
1712/1712 [=====] - 1s 711us/step - loss: 0.4039
- val_loss: 0.1897
Epoch 6/30
1712/1712 [=====] - 1s 713us/step - loss: 0.3755
- val_loss: 0.1707
Epoch 7/30
1712/1712 [=====] - 1s 711us/step - loss: 0.3548
- val_loss: 0.1532
Epoch 8/30
1712/1712 [=====] - 1s 702us/step - loss: 0.3292
- val_loss: 0.1379
Epoch 9/30
1712/1712 [=====] - 1s 713us/step - loss: 0.3061
- val_loss: 0.1245
Epoch 10/30
1712/1712 [=====] - 1s 710us/step - loss: 0.2908
- val_loss: 0.1124
Epoch 11/30
1712/1712 [=====] - 1s 711us/step - loss: 0.2763
- val_loss: 0.1017
Epoch 12/30
1712/1712 [=====] - 1s 712us/step - loss: 0.2567
```

```
- val_loss: 0.0928
Epoch 13/30
1712/1712 [=====] - 1s 703us/step - loss: 0.2445
- val_loss: 0.0849
Epoch 14/30
1712/1712 [=====] - 1s 703us/step - loss: 0.2320
- val_loss: 0.0780
Epoch 15/30
1712/1712 [=====] - 1s 709us/step - loss: 0.2207
- val_loss: 0.0721
Epoch 16/30
1712/1712 [=====] - 1s 711us/step - loss: 0.2104
- val_loss: 0.0670
Epoch 17/30
1712/1712 [=====] - 1s 710us/step - loss: 0.2010
- val_loss: 0.0624
Epoch 18/30
1712/1712 [=====] - 1s 704us/step - loss: 0.1908
- val_loss: 0.0585
Epoch 19/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1843
- val_loss: 0.0552
Epoch 20/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1766
- val_loss: 0.0523
Epoch 21/30
1712/1712 [=====] - 1s 712us/step - loss: 0.1690
- val_loss: 0.0498
Epoch 22/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1631
- val_loss: 0.0477
Epoch 23/30
1712/1712 [=====] - 1s 702us/step - loss: 0.1554
- val_loss: 0.0458
Epoch 24/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1503
- val_loss: 0.0440
Epoch 25/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1452
- val_loss: 0.0423
Epoch 26/30
1712/1712 [=====] - 1s 695us/step - loss: 0.1393
- val_loss: 0.0409
Epoch 27/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1363
- val_loss: 0.0395
Epoch 28/30
1712/1712 [=====] - 1s 692us/step - loss: 0.1314
- val_loss: 0.0383
Epoch 29/30
1712/1712 [=====] - 1s 692us/step - loss: 0.1267
- val_loss: 0.0371
Epoch 30/30
1712/1712 [=====] - 1s 708us/step - loss: 0.1225
- val_loss: 0.0360
```

```
Training model with spec model_Adadelta0.002he_uniform (Combo: 16/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 3.0049 -
val_loss: 1.4098
Epoch 2/30
1712/1712 [=====] - 1s 718us/step - loss: 2.7747
- val_loss: 1.2553
Epoch 3/30
1712/1712 [=====] - 1s 720us/step - loss: 2.5721
- val_loss: 1.1177
Epoch 4/30
1712/1712 [=====] - 1s 720us/step - loss: 2.3833
- val_loss: 0.9966
Epoch 5/30
1712/1712 [=====] - 1s 715us/step - loss: 2.2131
- val_loss: 0.8887
Epoch 6/30
1712/1712 [=====] - 1s 720us/step - loss: 2.0711
- val_loss: 0.7912
Epoch 7/30
1712/1712 [=====] - 1s 703us/step - loss: 1.9376
- val_loss: 0.7049
Epoch 8/30
1712/1712 [=====] - 1s 714us/step - loss: 1.8461
- val_loss: 0.6293
Epoch 9/30
1712/1712 [=====] - 1s 719us/step - loss: 1.7031
- val_loss: 0.5614
Epoch 10/30
1712/1712 [=====] - 1s 719us/step - loss: 1.6314
- val_loss: 0.5012
Epoch 11/30
1712/1712 [=====] - 1s 722us/step - loss: 1.5327
- val_loss: 0.4489
Epoch 12/30
1712/1712 [=====] - 1s 685us/step - loss: 1.4407
- val_loss: 0.4017
Epoch 13/30
1712/1712 [=====] - 1s 684us/step - loss: 1.3618
- val_loss: 0.3608
Epoch 14/30
1712/1712 [=====] - 1s 696us/step - loss: 1.3084
- val_loss: 0.3257
Epoch 15/30
1712/1712 [=====] - 1s 682us/step - loss: 1.2287
- val_loss: 0.2955
Epoch 16/30
1712/1712 [=====] - 1s 683us/step - loss: 1.1744
- val_loss: 0.2689
Epoch 17/30
1712/1712 [=====] - 1s 683us/step - loss: 1.1227
- val_loss: 0.2450
Epoch 18/30
1712/1712 [=====] - 1s 698us/step - loss: 1.0659
- val_loss: 0.2251
```

```
Epoch 19/30
1712/1712 [=====] - 1s 719us/step - loss: 1.0037
- val_loss: 0.2075
Epoch 20/30
1712/1712 [=====] - 1s 704us/step - loss: 0.9628
- val_loss: 0.1920
Epoch 21/30
1712/1712 [=====] - 1s 716us/step - loss: 0.9161
- val_loss: 0.1776
Epoch 22/30
1712/1712 [=====] - 1s 710us/step - loss: 0.8768
- val_loss: 0.1660
Epoch 23/30
1712/1712 [=====] - 1s 720us/step - loss: 0.8284
- val_loss: 0.1559
Epoch 24/30
1712/1712 [=====] - 1s 699us/step - loss: 0.7946
- val_loss: 0.1460
Epoch 25/30
1712/1712 [=====] - 1s 712us/step - loss: 0.7686
- val_loss: 0.1374
Epoch 26/30
1712/1712 [=====] - 1s 711us/step - loss: 0.7294
- val_loss: 0.1302
Epoch 27/30
1712/1712 [=====] - 1s 716us/step - loss: 0.6932
- val_loss: 0.1237
Epoch 28/30
1712/1712 [=====] - 1s 719us/step - loss: 0.6597
- val_loss: 0.1178
Epoch 29/30
1712/1712 [=====] - 1s 716us/step - loss: 0.6415
- val_loss: 0.1124
Epoch 30/30
1712/1712 [=====] - 1s 718us/step - loss: 0.6033
- val_loss: 0.1080
```

```
Training model with spec model_Adadelta0.001glorot_normal (Combo: 17/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.1568 -
val_loss: 0.1590
Epoch 2/30
1712/1712 [=====] - 1s 720us/step - loss: 0.1566
- val_loss: 0.1585
Epoch 3/30
1712/1712 [=====] - 1s 719us/step - loss: 0.1560
- val_loss: 0.1580
Epoch 4/30
1712/1712 [=====] - 1s 720us/step - loss: 0.1556
- val_loss: 0.1575
Epoch 5/30
1712/1712 [=====] - 1s 720us/step - loss: 0.1551
- val_loss: 0.1570
Epoch 6/30
```

```
1712/1712 [=====] - 1s 700us/step - loss: 0.1546
- val_loss: 0.1565
Epoch 7/30
1712/1712 [=====] - 1s 699us/step - loss: 0.1542
- val_loss: 0.1560
Epoch 8/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1537
- val_loss: 0.1555
Epoch 9/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1530
- val_loss: 0.1550
Epoch 10/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1527
- val_loss: 0.1545
Epoch 11/30
1712/1712 [=====] - 1s 719us/step - loss: 0.1521
- val_loss: 0.1540
Epoch 12/30
1712/1712 [=====] - 1s 721us/step - loss: 0.1515
- val_loss: 0.1535
Epoch 13/30
1712/1712 [=====] - 1s 723us/step - loss: 0.1511
- val_loss: 0.1530
Epoch 14/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1507
- val_loss: 0.1525
Epoch 15/30
1712/1712 [=====] - 1s 719us/step - loss: 0.1502
- val_loss: 0.1519
Epoch 16/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1498
- val_loss: 0.1514
Epoch 17/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1491
- val_loss: 0.1509
Epoch 18/30
1712/1712 [=====] - 1s 721us/step - loss: 0.1486
- val_loss: 0.1503
Epoch 19/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1481
- val_loss: 0.1498
Epoch 20/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1476
- val_loss: 0.1493
Epoch 21/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1470
- val_loss: 0.1487
Epoch 22/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1467
- val_loss: 0.1482
Epoch 23/30
1712/1712 [=====] - 1s 719us/step - loss: 0.1459
- val_loss: 0.1476
Epoch 24/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1454
- val_loss: 0.1470
Epoch 25/30
```

```
1712/1712 [=====] - 1s 712us/step - loss: 0.1450
- val_loss: 0.1465
Epoch 26/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1444
- val_loss: 0.1459
Epoch 27/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1438
- val_loss: 0.1453
Epoch 28/30
1712/1712 [=====] - 1s 719us/step - loss: 0.1434
- val_loss: 0.1448
Epoch 29/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1427
- val_loss: 0.1442
Epoch 30/30
1712/1712 [=====] - 1s 712us/step - loss: 0.1421
- val_loss: 0.1436
```

```
Training model with spec model_Adadelta0.001glorot_uniform (Combo: 18/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.1523 - 
val_loss: 0.1519
Epoch 2/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1513
- val_loss: 0.1509
Epoch 3/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1503
- val_loss: 0.1498
Epoch 4/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1492
- val_loss: 0.1488
Epoch 5/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1481
- val_loss: 0.1477
Epoch 6/30
1712/1712 [=====] - 1s 712us/step - loss: 0.1471
- val_loss: 0.1466
Epoch 7/30
1712/1712 [=====] - 1s 693us/step - loss: 0.1462
- val_loss: 0.1455
Epoch 8/30
1712/1712 [=====] - 1s 706us/step - loss: 0.1449
- val_loss: 0.1443
Epoch 9/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1439
- val_loss: 0.1432
Epoch 10/30
1712/1712 [=====] - 1s 720us/step - loss: 0.1429
- val_loss: 0.1421
Epoch 11/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1417
- val_loss: 0.1409
Epoch 12/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1407
```

```
- val_loss: 0.1397
Epoch 13/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1395
- val_loss: 0.1385
Epoch 14/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1383
- val_loss: 0.1373
Epoch 15/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1374
- val_loss: 0.1361
Epoch 16/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1361
- val_loss: 0.1349
Epoch 17/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1349
- val_loss: 0.1336
Epoch 18/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1336
- val_loss: 0.1323
Epoch 19/30
1712/1712 [=====] - 1s 721us/step - loss: 0.1325
- val_loss: 0.1311
Epoch 20/30
1712/1712 [=====] - 1s 718us/step - loss: 0.1313
- val_loss: 0.1298
Epoch 21/30
1712/1712 [=====] - 1s 696us/step - loss: 0.1298
- val_loss: 0.1285
Epoch 22/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1292
- val_loss: 0.1272
Epoch 23/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1279
- val_loss: 0.1259
Epoch 24/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1264
- val_loss: 0.1245
Epoch 25/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1250
- val_loss: 0.1232
Epoch 26/30
1712/1712 [=====] - 1s 694us/step - loss: 0.1240
- val_loss: 0.1218
Epoch 27/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1227
- val_loss: 0.1205
Epoch 28/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1212
- val_loss: 0.1191
Epoch 29/30
1712/1712 [=====] - 1s 717us/step - loss: 0.1202
- val_loss: 0.1177
Epoch 30/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1186
- val_loss: 0.1163
```

```
Training model with spec model_Adadelta0.001he_normal (Combo: 19/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.5112 -
val_loss: 0.3624
Epoch 2/30
1712/1712 [=====] - 1s 709us/step - loss: 0.4911
- val_loss: 0.3467
Epoch 3/30
1712/1712 [=====] - 1s 712us/step - loss: 0.4725
- val_loss: 0.3317
Epoch 4/30
1712/1712 [=====] - 1s 713us/step - loss: 0.4563
- val_loss: 0.3174
Epoch 5/30
1712/1712 [=====] - 1s 715us/step - loss: 0.4413
- val_loss: 0.3036
Epoch 6/30
1712/1712 [=====] - 1s 713us/step - loss: 0.4260
- val_loss: 0.2905
Epoch 7/30
1712/1712 [=====] - 1s 710us/step - loss: 0.4100
- val_loss: 0.2779
Epoch 8/30
1712/1712 [=====] - 1s 713us/step - loss: 0.3964
- val_loss: 0.2658
Epoch 9/30
1712/1712 [=====] - 1s 711us/step - loss: 0.3857
- val_loss: 0.2542
Epoch 10/30
1712/1712 [=====] - 1s 704us/step - loss: 0.3692
- val_loss: 0.2431
Epoch 11/30
1712/1712 [=====] - 1s 713us/step - loss: 0.3553
- val_loss: 0.2324
Epoch 12/30
1712/1712 [=====] - 1s 710us/step - loss: 0.3455
- val_loss: 0.2222
Epoch 13/30
1712/1712 [=====] - 1s 712us/step - loss: 0.3361
- val_loss: 0.2123
Epoch 14/30
1712/1712 [=====] - 1s 713us/step - loss: 0.3203
- val_loss: 0.2028
Epoch 15/30
1712/1712 [=====] - 1s 715us/step - loss: 0.3097
- val_loss: 0.1936
Epoch 16/30
1712/1712 [=====] - 1s 705us/step - loss: 0.3010
- val_loss: 0.1848
Epoch 17/30
1712/1712 [=====] - 1s 682us/step - loss: 0.2883
- val_loss: 0.1764
Epoch 18/30
1712/1712 [=====] - 1s 680us/step - loss: 0.2799
- val_loss: 0.1682
```

```
Epoch 19/30
1712/1712 [=====] - 1s 688us/step - loss: 0.2715
- val_loss: 0.1604
Epoch 20/30
1712/1712 [=====] - 1s 680us/step - loss: 0.2615
- val_loss: 0.1529
Epoch 21/30
1712/1712 [=====] - 1s 696us/step - loss: 0.2565
- val_loss: 0.1458
Epoch 22/30
1712/1712 [=====] - 1s 706us/step - loss: 0.2459
- val_loss: 0.1393
Epoch 23/30
1712/1712 [=====] - 1s 685us/step - loss: 0.2383
- val_loss: 0.1329
Epoch 24/30
1712/1712 [=====] - 1s 696us/step - loss: 0.2314
- val_loss: 0.1270
Epoch 25/30
1712/1712 [=====] - 1s 685us/step - loss: 0.2246
- val_loss: 0.1214
Epoch 26/30
1712/1712 [=====] - 1s 710us/step - loss: 0.2187
- val_loss: 0.1160
Epoch 27/30
1712/1712 [=====] - 1s 686us/step - loss: 0.2147
- val_loss: 0.1109
Epoch 28/30
1712/1712 [=====] - 1s 698us/step - loss: 0.2059
- val_loss: 0.1061
Epoch 29/30
1712/1712 [=====] - 1s 707us/step - loss: 0.2018
- val_loss: 0.1015
Epoch 30/30
1712/1712 [=====] - 1s 686us/step - loss: 0.1960
- val_loss: 0.0972
```

```
Training model with spec model_Adadelta0.001he_uniform (Combo: 20/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 2.1443 -
val_loss: 1.1160
Epoch 2/30
1712/1712 [=====] - 1s 718us/step - loss: 2.0346
- val_loss: 1.0602
Epoch 3/30
1712/1712 [=====] - 1s 719us/step - loss: 2.0172
- val_loss: 1.0051
Epoch 4/30
1712/1712 [=====] - 1s 718us/step - loss: 1.9150
- val_loss: 0.9543
Epoch 5/30
1712/1712 [=====] - 1s 719us/step - loss: 1.8575
- val_loss: 0.9057
Epoch 6/30
```

```
1712/1712 [=====] - 1s 721us/step - loss: 1.7752
- val_loss: 0.8599
Epoch 7/30
1712/1712 [=====] - 1s 714us/step - loss: 1.7243
- val_loss: 0.8159
Epoch 8/30
1712/1712 [=====] - 1s 716us/step - loss: 1.6956
- val_loss: 0.7732
Epoch 9/30
1712/1712 [=====] - 1s 706us/step - loss: 1.6159
- val_loss: 0.7322
Epoch 10/30
1712/1712 [=====] - 1s 691us/step - loss: 1.5451
- val_loss: 0.6940
Epoch 11/30
1712/1712 [=====] - 1s 694us/step - loss: 1.4952
- val_loss: 0.6572
Epoch 12/30
1712/1712 [=====] - 1s 705us/step - loss: 1.4455
- val_loss: 0.6216
Epoch 13/30
1712/1712 [=====] - 1s 682us/step - loss: 1.3962
- val_loss: 0.5876
Epoch 14/30
1712/1712 [=====] - 1s 693us/step - loss: 1.3447
- val_loss: 0.5549
Epoch 15/30
1712/1712 [=====] - 1s 699us/step - loss: 1.2995
- val_loss: 0.5237
Epoch 16/30
1712/1712 [=====] - 1s 705us/step - loss: 1.2497
- val_loss: 0.4940
Epoch 17/30
1712/1712 [=====] - 1s 713us/step - loss: 1.2298
- val_loss: 0.4656
Epoch 18/30
1712/1712 [=====] - 1s 714us/step - loss: 1.1801
- val_loss: 0.4390
Epoch 19/30
1712/1712 [=====] - 1s 713us/step - loss: 1.1432
- val_loss: 0.4138
Epoch 20/30
1712/1712 [=====] - 1s 695us/step - loss: 1.0930
- val_loss: 0.3901
Epoch 21/30
1712/1712 [=====] - 1s 707us/step - loss: 1.0650
- val_loss: 0.3679
Epoch 22/30
1712/1712 [=====] - 1s 689us/step - loss: 1.0401
- val_loss: 0.3469
Epoch 23/30
1712/1712 [=====] - 1s 694us/step - loss: 0.9961
- val_loss: 0.3274
Epoch 24/30
1712/1712 [=====] - 1s 702us/step - loss: 0.9580
- val_loss: 0.3089
Epoch 25/30
```

```
1712/1712 [=====] - 1s 684us/step - loss: 0.9451
- val_loss: 0.2914
Epoch 26/30
1712/1712 [=====] - 1s 699us/step - loss: 0.9082
- val_loss: 0.2753
Epoch 27/30
1712/1712 [=====] - 1s 697us/step - loss: 0.8839
- val_loss: 0.2605
Epoch 28/30
1712/1712 [=====] - 1s 711us/step - loss: 0.8561
- val_loss: 0.2466
Epoch 29/30
1712/1712 [=====] - 1s 710us/step - loss: 0.8288
- val_loss: 0.2331
Epoch 30/30
1712/1712 [=====] - 1s 716us/step - loss: 0.8099
- val_loss: 0.2213
```

```
Training model with spec model_Adadelta0.0009glorot_normal (Combo: 21/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.1548 -
- val_loss: 0.1569
Epoch 2/30
1712/1712 [=====] - 1s 712us/step - loss: 0.1544
- val_loss: 0.1565
Epoch 3/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1542
- val_loss: 0.1561
Epoch 4/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1537
- val_loss: 0.1557
Epoch 5/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1535
- val_loss: 0.1553
Epoch 6/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1530
- val_loss: 0.1550
Epoch 7/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1527
- val_loss: 0.1545
Epoch 8/30
1712/1712 [=====] - 1s 698us/step - loss: 0.1522
- val_loss: 0.1541
Epoch 9/30
1712/1712 [=====] - 1s 706us/step - loss: 0.1519
- val_loss: 0.1537
Epoch 10/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1513
- val_loss: 0.1533
Epoch 11/30
1712/1712 [=====] - 1s 709us/step - loss: 0.1511
- val_loss: 0.1529
Epoch 12/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1505
```

```
- val_loss: 0.1525
Epoch 13/30
1712/1712 [=====] - 1s 712us/step - loss: 0.1502
- val_loss: 0.1520
Epoch 14/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1498
- val_loss: 0.1516
Epoch 15/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1492
- val_loss: 0.1511
Epoch 16/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1489
- val_loss: 0.1507
Epoch 17/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1485
- val_loss: 0.1502
Epoch 18/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1479
- val_loss: 0.1498
Epoch 19/30
1712/1712 [=====] - 1s 705us/step - loss: 0.1476
- val_loss: 0.1493
Epoch 20/30
1712/1712 [=====] - 1s 677us/step - loss: 0.1472
- val_loss: 0.1488
Epoch 21/30
1712/1712 [=====] - 1s 689us/step - loss: 0.1466
- val_loss: 0.1484
Epoch 22/30
1712/1712 [=====] - 1s 701us/step - loss: 0.1463
- val_loss: 0.1479
Epoch 23/30
1712/1712 [=====] - 1s 681us/step - loss: 0.1459
- val_loss: 0.1474
Epoch 24/30
1712/1712 [=====] - 1s 692us/step - loss: 0.1454
- val_loss: 0.1469
Epoch 25/30
1712/1712 [=====] - 1s 695us/step - loss: 0.1447
- val_loss: 0.1464
Epoch 26/30
1712/1712 [=====] - 1s 705us/step - loss: 0.1444
- val_loss: 0.1459
Epoch 27/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1440
- val_loss: 0.1454
Epoch 28/30
1712/1712 [=====] - 1s 707us/step - loss: 0.1435
- val_loss: 0.1449
Epoch 29/30
1712/1712 [=====] - 1s 711us/step - loss: 0.1430
- val_loss: 0.1444
Epoch 30/30
1712/1712 [=====] - 1s 710us/step - loss: 0.1424
- val_loss: 0.1439
```

```
Training model with spec model_Adadelta0.0009glorot_uniform (Combo: 22/6
0)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.1673 -
    val_loss: 0.1629
Epoch 2/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1656
    - val_loss: 0.1614
Epoch 3/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1639
    - val_loss: 0.1599
Epoch 4/30
1712/1712 [=====] - 1s 712us/step - loss: 0.1635
    - val_loss: 0.1584
Epoch 5/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1613
    - val_loss: 0.1569
Epoch 6/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1596
    - val_loss: 0.1554
Epoch 7/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1586
    - val_loss: 0.1539
Epoch 8/30
1712/1712 [=====] - 1s 714us/step - loss: 0.1563
    - val_loss: 0.1524
Epoch 9/30
1712/1712 [=====] - 1s 716us/step - loss: 0.1549
    - val_loss: 0.1509
Epoch 10/30
1712/1712 [=====] - 1s 720us/step - loss: 0.1540
    - val_loss: 0.1494
Epoch 11/30
1712/1712 [=====] - 1s 703us/step - loss: 0.1526
    - val_loss: 0.1478
Epoch 12/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1508
    - val_loss: 0.1463
Epoch 13/30
1712/1712 [=====] - 1s 722us/step - loss: 0.1495
    - val_loss: 0.1447
Epoch 14/30
1712/1712 [=====] - 1s 722us/step - loss: 0.1476
    - val_loss: 0.1431
Epoch 15/30
1712/1712 [=====] - 1s 721us/step - loss: 0.1459
    - val_loss: 0.1415
Epoch 16/30
1712/1712 [=====] - 1s 721us/step - loss: 0.1443
    - val_loss: 0.1399
Epoch 17/30
1712/1712 [=====] - 1s 721us/step - loss: 0.1428
    - val_loss: 0.1383
Epoch 18/30
1712/1712 [=====] - 1s 721us/step - loss: 0.1414
```

```
- val_loss: 0.1367
Epoch 19/30
1712/1712 [=====] - 1s 695us/step - loss: 0.1400
- val_loss: 0.1350
Epoch 20/30
1712/1712 [=====] - 1s 698us/step - loss: 0.1382
- val_loss: 0.1334
Epoch 21/30
1712/1712 [=====] - 1s 700us/step - loss: 0.1368
- val_loss: 0.1318
Epoch 22/30
1712/1712 [=====] - 1s 715us/step - loss: 0.1355
- val_loss: 0.1302
Epoch 23/30
1712/1712 [=====] - 1s 689us/step - loss: 0.1341
- val_loss: 0.1285
Epoch 24/30
1712/1712 [=====] - 1s 702us/step - loss: 0.1322
- val_loss: 0.1269
Epoch 25/30
1712/1712 [=====] - 1s 701us/step - loss: 0.1312
- val_loss: 0.1253
Epoch 26/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1297
- val_loss: 0.1237
Epoch 27/30
1712/1712 [=====] - 1s 688us/step - loss: 0.1284
- val_loss: 0.1221
Epoch 28/30
1712/1712 [=====] - 1s 698us/step - loss: 0.1262
- val_loss: 0.1204
Epoch 29/30
1712/1712 [=====] - 1s 703us/step - loss: 0.1252
- val_loss: 0.1188
Epoch 30/30
1712/1712 [=====] - 1s 713us/step - loss: 0.1239
- val_loss: 0.1172
```

```
Training model with spec model_Adadelta0.0009he_normal (Combo: 23/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.6050 -
val_loss: 0.3506
Epoch 2/30
1712/1712 [=====] - 1s 721us/step - loss: 0.5863
- val_loss: 0.3382
Epoch 3/30
1712/1712 [=====] - 1s 722us/step - loss: 0.5708
- val_loss: 0.3262
Epoch 4/30
1712/1712 [=====] - 1s 724us/step - loss: 0.5573
- val_loss: 0.3144
Epoch 5/30
1712/1712 [=====] - 1s 721us/step - loss: 0.5456
- val_loss: 0.3031
```

```
Epoch 6/30
1712/1712 [=====] - 1s 718us/step - loss: 0.5316
- val_loss: 0.2922
Epoch 7/30
1712/1712 [=====] - 1s 717us/step - loss: 0.5128
- val_loss: 0.2817
Epoch 8/30
1712/1712 [=====] - 1s 712us/step - loss: 0.5063
- val_loss: 0.2712
Epoch 9/30
1712/1712 [=====] - 1s 713us/step - loss: 0.4855
- val_loss: 0.2610
Epoch 10/30
1712/1712 [=====] - 1s 712us/step - loss: 0.4729
- val_loss: 0.2506
Epoch 11/30
1712/1712 [=====] - 1s 711us/step - loss: 0.4564
- val_loss: 0.2407
Epoch 12/30
1712/1712 [=====] - 1s 716us/step - loss: 0.4434
- val_loss: 0.2307
Epoch 13/30
1712/1712 [=====] - 1s 718us/step - loss: 0.4297
- val_loss: 0.2213
Epoch 14/30
1712/1712 [=====] - 1s 710us/step - loss: 0.4195
- val_loss: 0.2121
Epoch 15/30
1712/1712 [=====] - 1s 697us/step - loss: 0.4036
- val_loss: 0.2033
Epoch 16/30
1712/1712 [=====] - 1s 710us/step - loss: 0.3945
- val_loss: 0.1948
Epoch 17/30
1712/1712 [=====] - 1s 685us/step - loss: 0.3774
- val_loss: 0.1866
Epoch 18/30
1712/1712 [=====] - 1s 698us/step - loss: 0.3703
- val_loss: 0.1787
Epoch 19/30
1712/1712 [=====] - 1s 685us/step - loss: 0.3599
- val_loss: 0.1709
Epoch 20/30
1712/1712 [=====] - 1s 686us/step - loss: 0.3461
- val_loss: 0.1637
Epoch 21/30
1712/1712 [=====] - 1s 698us/step - loss: 0.3398
- val_loss: 0.1567
Epoch 22/30
1712/1712 [=====] - 1s 690us/step - loss: 0.3308
- val_loss: 0.1500
Epoch 23/30
1712/1712 [=====] - 1s 699us/step - loss: 0.3206
- val_loss: 0.1438
Epoch 24/30
1712/1712 [=====] - 1s 700us/step - loss: 0.3100
- val_loss: 0.1378
```

```
Epoch 25/30
1712/1712 [=====] - 1s 685us/step - loss: 0.3041
- val_loss: 0.1322
Epoch 26/30
1712/1712 [=====] - 1s 697us/step - loss: 0.2950
- val_loss: 0.1267
Epoch 27/30
1712/1712 [=====] - 1s 686us/step - loss: 0.2884
- val_loss: 0.1216
Epoch 28/30
1712/1712 [=====] - 1s 695us/step - loss: 0.2793
- val_loss: 0.1167
Epoch 29/30
1712/1712 [=====] - 1s 707us/step - loss: 0.2735
- val_loss: 0.1122
Epoch 30/30
1712/1712 [=====] - 1s 686us/step - loss: 0.2680
- val_loss: 0.1079
```

```
Training model with spec model_Adadelta0.0009he_uniform (Combo: 24/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 1.3777 -
val_loss: 0.7252
Epoch 2/30
1712/1712 [=====] - 1s 720us/step - loss: 1.3253
- val_loss: 0.6880
Epoch 3/30
1712/1712 [=====] - 1s 717us/step - loss: 1.3044
- val_loss: 0.6526
Epoch 4/30
1712/1712 [=====] - 1s 721us/step - loss: 1.2293
- val_loss: 0.6199
Epoch 5/30
1712/1712 [=====] - 1s 719us/step - loss: 1.1944
- val_loss: 0.5889
Epoch 6/30
1712/1712 [=====] - 1s 723us/step - loss: 1.1531
- val_loss: 0.5598
Epoch 7/30
1712/1712 [=====] - 1s 719us/step - loss: 1.1173
- val_loss: 0.5321
Epoch 8/30
1712/1712 [=====] - 1s 720us/step - loss: 1.0795
- val_loss: 0.5066
Epoch 9/30
1712/1712 [=====] - 1s 722us/step - loss: 1.0426
- val_loss: 0.4820
Epoch 10/30
1712/1712 [=====] - 1s 722us/step - loss: 1.0085
- val_loss: 0.4585
Epoch 11/30
1712/1712 [=====] - 1s 724us/step - loss: 0.9717
- val_loss: 0.4365
Epoch 12/30
```

```
1712/1712 [=====] - 1s 717us/step - loss: 0.9435
  - val_loss: 0.4155
Epoch 13/30
1712/1712 [=====] - 1s 709us/step - loss: 0.9139
  - val_loss: 0.3953
Epoch 14/30
1712/1712 [=====] - 1s 721us/step - loss: 0.8887
  - val_loss: 0.3760
Epoch 15/30
1712/1712 [=====] - 1s 696us/step - loss: 0.8524
  - val_loss: 0.3578
Epoch 16/30
1712/1712 [=====] - 1s 716us/step - loss: 0.8259
  - val_loss: 0.3404
Epoch 17/30
1712/1712 [=====] - 1s 717us/step - loss: 0.8018
  - val_loss: 0.3241
Epoch 18/30
1712/1712 [=====] - 1s 715us/step - loss: 0.7808
  - val_loss: 0.3085
Epoch 19/30
1712/1712 [=====] - 1s 713us/step - loss: 0.7570
  - val_loss: 0.2933
Epoch 20/30
1712/1712 [=====] - 1s 713us/step - loss: 0.7285
  - val_loss: 0.2790
Epoch 21/30
1712/1712 [=====] - 1s 711us/step - loss: 0.7161
  - val_loss: 0.2653
Epoch 22/30
1712/1712 [=====] - 1s 709us/step - loss: 0.6859
  - val_loss: 0.2526
Epoch 23/30
1712/1712 [=====] - 1s 714us/step - loss: 0.6607
  - val_loss: 0.2403
Epoch 24/30
1712/1712 [=====] - 1s 715us/step - loss: 0.6472
  - val_loss: 0.2283
Epoch 25/30
1712/1712 [=====] - 1s 715us/step - loss: 0.6257
  - val_loss: 0.2167
Epoch 26/30
1712/1712 [=====] - 1s 714us/step - loss: 0.6106
  - val_loss: 0.2053
Epoch 27/30
1712/1712 [=====] - 1s 716us/step - loss: 0.5930
  - val_loss: 0.1947
Epoch 28/30
1712/1712 [=====] - 1s 718us/step - loss: 0.5731
  - val_loss: 0.1848
Epoch 29/30
1712/1712 [=====] - 1s 717us/step - loss: 0.5520
  - val_loss: 0.1756
Epoch 30/30
1712/1712 [=====] - 1s 715us/step - loss: 0.5425
  - val_loss: 0.1670
```

```
Training model with spec model_Adam0.002glorot_normal (Combo: 25/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 2s 1ms/step - loss: 0.1259 -
  val_loss: 0.0443
Epoch 2/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0334
  - val_loss: 0.0080
Epoch 3/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0169
  - val_loss: 0.0050
Epoch 4/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0126
  - val_loss: 0.0054
Epoch 5/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0114
  - val_loss: 0.0052
Epoch 6/30
1712/1712 [=====] - 1s 702us/step - loss: 0.0102
  - val_loss: 0.0049
Epoch 7/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0096
  - val_loss: 0.0046
Epoch 8/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0092
  - val_loss: 0.0049
Epoch 9/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0090
  - val_loss: 0.0044
Epoch 10/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0084
  - val_loss: 0.0046
Epoch 11/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0082
  - val_loss: 0.0043
Epoch 12/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0080
  - val_loss: 0.0043
Epoch 13/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0077
  - val_loss: 0.0046
Epoch 14/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0076
  - val_loss: 0.0044
Epoch 15/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0072
  - val_loss: 0.0044
Epoch 16/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0072
  - val_loss: 0.0043
Epoch 17/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0068
  - val_loss: 0.0044
Epoch 18/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0067
```

```
- val_loss: 0.0041
Epoch 19/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0067
- val_loss: 0.0041
Epoch 20/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0064
- val_loss: 0.0042
Epoch 21/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0065
- val_loss: 0.0041
Epoch 22/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0063
- val_loss: 0.0042
Epoch 23/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0061
- val_loss: 0.0042
Epoch 24/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0062
- val_loss: 0.0040
Epoch 25/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0061
- val_loss: 0.0040
Epoch 26/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0059
- val_loss: 0.0040
Epoch 27/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0058
- val_loss: 0.0043
Epoch 28/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0058
- val_loss: 0.0043
Epoch 29/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0057
- val_loss: 0.0040
Epoch 30/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0056
- val_loss: 0.0039
```

```
Training model with spec model_Adam0.002glorot_uniform (Combo: 26/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 1ms/step - loss: 0.0514 -
val_loss: 0.0079
Epoch 2/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0171
- val_loss: 0.0051
Epoch 3/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0118
- val_loss: 0.0048
Epoch 4/30
1712/1712 [=====] - 1s 692us/step - loss: 0.0098
- val_loss: 0.0044
Epoch 5/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0090
- val_loss: 0.0051
```

```
Epoch 6/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0084
- val_loss: 0.0045
Epoch 7/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0080
- val_loss: 0.0047
Epoch 8/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0075
- val_loss: 0.0043
Epoch 9/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0073
- val_loss: 0.0042
Epoch 10/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0070
- val_loss: 0.0041
Epoch 11/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0067
- val_loss: 0.0044
Epoch 12/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0065
- val_loss: 0.0050
Epoch 13/30
1712/1712 [=====] - 1s 691us/step - loss: 0.0064
- val_loss: 0.0039
Epoch 14/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0060
- val_loss: 0.0039
Epoch 15/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0058
- val_loss: 0.0038
Epoch 16/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0058
- val_loss: 0.0037
Epoch 17/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0055
- val_loss: 0.0044
Epoch 18/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0058
- val_loss: 0.0040
Epoch 19/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0054
- val_loss: 0.0035
Epoch 20/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0051
- val_loss: 0.0034
Epoch 21/30
1712/1712 [=====] - 1s 689us/step - loss: 0.0050
- val_loss: 0.0034
Epoch 22/30
1712/1712 [=====] - 1s 702us/step - loss: 0.0048
- val_loss: 0.0033
Epoch 23/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0047
- val_loss: 0.0033
Epoch 24/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0046
- val_loss: 0.0040
```

```
Epoch 25/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0048
- val_loss: 0.0033
Epoch 26/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0046
- val_loss: 0.0029
Epoch 27/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0042
- val_loss: 0.0027
Epoch 28/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0040
- val_loss: 0.0027
Epoch 29/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0038
- val_loss: 0.0031
Epoch 30/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0040
- val_loss: 0.0026
```

```
Training model with spec model_Adam0.002he_normal (Combo: 27/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.4540 -
val_loss: 0.0237
Epoch 2/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0324
- val_loss: 0.0081
Epoch 3/30
1712/1712 [=====] - 1s 702us/step - loss: 0.0206
- val_loss: 0.0068
Epoch 4/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0159
- val_loss: 0.0075
Epoch 5/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0140
- val_loss: 0.0094
Epoch 6/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0133
- val_loss: 0.0054
Epoch 7/30
1712/1712 [=====] - 1s 704us/step - loss: 0.0113
- val_loss: 0.0048
Epoch 8/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0105
- val_loss: 0.0047
Epoch 9/30
1712/1712 [=====] - 1s 686us/step - loss: 0.0098
- val_loss: 0.0042
Epoch 10/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0094
- val_loss: 0.0049
Epoch 11/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0090
- val_loss: 0.0042
Epoch 12/30
```

```
1712/1712 [=====] - 1s 647us/step - loss: 0.0086
  - val_loss: 0.0042
Epoch 13/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0090
  - val_loss: 0.0044
Epoch 14/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0083
  - val_loss: 0.0041
Epoch 15/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0077
  - val_loss: 0.0039
Epoch 16/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0075
  - val_loss: 0.0038
Epoch 17/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0074
  - val_loss: 0.0040
Epoch 18/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0072
  - val_loss: 0.0042
Epoch 19/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0068
  - val_loss: 0.0039
Epoch 20/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0070
  - val_loss: 0.0042
Epoch 21/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0068
  - val_loss: 0.0040
Epoch 22/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0066
  - val_loss: 0.0037
Epoch 23/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0064
  - val_loss: 0.0039
Epoch 24/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0063
  - val_loss: 0.0037
Epoch 25/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0064
  - val_loss: 0.0037
Epoch 26/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0062
  - val_loss: 0.0036
Epoch 27/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0061
  - val_loss: 0.0035
Epoch 28/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0061
  - val_loss: 0.0037
Epoch 29/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0060
  - val_loss: 0.0036
Epoch 30/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0058
  - val_loss: 0.0034
```

Training model with spec model_Adam0.002he_uniform (Combo: 28/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.7959 -
- val_loss: 0.0348
Epoch 2/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0468
- val_loss: 0.0141
Epoch 3/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0289
- val_loss: 0.0090
Epoch 4/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0215
- val_loss: 0.0070
Epoch 5/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0188
- val_loss: 0.0050
Epoch 6/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0178
- val_loss: 0.0051
Epoch 7/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0157
- val_loss: 0.0053
Epoch 8/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0140
- val_loss: 0.0048
Epoch 9/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0132
- val_loss: 0.0067
Epoch 10/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0130
- val_loss: 0.0052
Epoch 11/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0124
- val_loss: 0.0043
Epoch 12/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0115
- val_loss: 0.0042
Epoch 13/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0114
- val_loss: 0.0042
Epoch 14/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0106
- val_loss: 0.0051
Epoch 15/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0101
- val_loss: 0.0048
Epoch 16/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0101
- val_loss: 0.0044
Epoch 17/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0098
- val_loss: 0.0045
Epoch 18/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0094

```
- val_loss: 0.0047
Epoch 19/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0099
- val_loss: 0.0045
Epoch 20/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0093
- val_loss: 0.0044
Epoch 21/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0088
- val_loss: 0.0057
Epoch 22/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0091
- val_loss: 0.0040
Epoch 23/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0090
- val_loss: 0.0048
Epoch 24/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0084
- val_loss: 0.0048
Epoch 25/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0084
- val_loss: 0.0042
Epoch 26/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0080
- val_loss: 0.0043
Epoch 27/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0078
- val_loss: 0.0050
Epoch 28/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0077
- val_loss: 0.0039
Epoch 29/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0074
- val_loss: 0.0040
Epoch 30/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0075
- val_loss: 0.0044
```

```
Training model with spec model_Adam0.001glorot_normal (Combo: 29/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.0515 -
val_loss: 0.0096
Epoch 2/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0210
- val_loss: 0.0055
Epoch 3/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0140
- val_loss: 0.0051
Epoch 4/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0117
- val_loss: 0.0049
Epoch 5/30
1712/1712 [=====] - 1s 691us/step - loss: 0.0103
- val_loss: 0.0044
```

```
Epoch 6/30
1712/1712 [=====] - 1s 701us/step - loss: 0.0097
- val_loss: 0.0044
Epoch 7/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0088
- val_loss: 0.0049
Epoch 8/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0086
- val_loss: 0.0044
Epoch 9/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0081
- val_loss: 0.0046
Epoch 10/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0079
- val_loss: 0.0041
Epoch 11/30
1712/1712 [=====] - 1s 722us/step - loss: 0.0074
- val_loss: 0.0040
Epoch 12/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0072
- val_loss: 0.0039
Epoch 13/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0070
- val_loss: 0.0043
Epoch 14/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0066
- val_loss: 0.0045
Epoch 15/30
1712/1712 [=====] - 1s 699us/step - loss: 0.0065
- val_loss: 0.0037
Epoch 16/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0063
- val_loss: 0.0038
Epoch 17/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0061
- val_loss: 0.0038
Epoch 18/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0060
- val_loss: 0.0035
Epoch 19/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0059
- val_loss: 0.0043
Epoch 20/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0058
- val_loss: 0.0037
Epoch 21/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0054
- val_loss: 0.0033
Epoch 22/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0052
- val_loss: 0.0030
Epoch 23/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0051
- val_loss: 0.0031
Epoch 24/30
1712/1712 [=====] - 1s 661us/step - loss: 0.0049
- val_loss: 0.0031
```

```
Epoch 25/30
1712/1712 [=====] - 1s 722us/step - loss: 0.0048
- val_loss: 0.0027
Epoch 26/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0047
- val_loss: 0.0026
Epoch 27/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0046
- val_loss: 0.0025
Epoch 28/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0043
- val_loss: 0.0027
Epoch 29/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0042
- val_loss: 0.0023
Epoch 30/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0043
- val_loss: 0.0028
```

```
Training model with spec model_Adam0.001glorot_uniform (Combo: 30/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.0551 -
val_loss: 0.0134
Epoch 2/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0246
- val_loss: 0.0067
Epoch 3/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0168
- val_loss: 0.0049
Epoch 4/30
1712/1712 [=====] - 1s 696us/step - loss: 0.0135
- val_loss: 0.0049
Epoch 5/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0117
- val_loss: 0.0049
Epoch 6/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0107
- val_loss: 0.0044
Epoch 7/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0097
- val_loss: 0.0041
Epoch 8/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0091
- val_loss: 0.0042
Epoch 9/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0085
- val_loss: 0.0039
Epoch 10/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0081
- val_loss: 0.0042
Epoch 11/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0078
- val_loss: 0.0039
Epoch 12/30
```

```
1712/1712 [=====] - 1s 658us/step - loss: 0.0075
  - val_loss: 0.0044
Epoch 13/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0072
  - val_loss: 0.0038
Epoch 14/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0067
  - val_loss: 0.0032
Epoch 15/30
1712/1712 [=====] - 1s 702us/step - loss: 0.0064
  - val_loss: 0.0031
Epoch 16/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0063
  - val_loss: 0.0032
Epoch 17/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0060
  - val_loss: 0.0028
Epoch 18/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0058
  - val_loss: 0.0027
Epoch 19/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0055
  - val_loss: 0.0026
Epoch 20/30
1712/1712 [=====] - 1s 703us/step - loss: 0.0055
  - val_loss: 0.0025
Epoch 21/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0050
  - val_loss: 0.0025
Epoch 22/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0051
  - val_loss: 0.0023
Epoch 23/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0049
  - val_loss: 0.0026
Epoch 24/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0048
  - val_loss: 0.0023
Epoch 25/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0047
  - val_loss: 0.0023
Epoch 26/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0048
  - val_loss: 0.0023
Epoch 27/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0044
  - val_loss: 0.0026
Epoch 28/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0044
  - val_loss: 0.0022
Epoch 29/30
1712/1712 [=====] - 1s 698us/step - loss: 0.0042
  - val_loss: 0.0021
Epoch 30/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0041
  - val_loss: 0.0020
```

Training model with spec model_Adam0.001he_normal (Combo: 31/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.1641 -
- val_loss: 0.0198
Epoch 2/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0405
- val_loss: 0.0116
Epoch 3/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0282
- val_loss: 0.0093
Epoch 4/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0226
- val_loss: 0.0093
Epoch 5/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0184
- val_loss: 0.0066
Epoch 6/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0160
- val_loss: 0.0046
Epoch 7/30
1712/1712 [=====] - 1s 685us/step - loss: 0.0141
- val_loss: 0.0043
Epoch 8/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0133
- val_loss: 0.0050
Epoch 9/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0123
- val_loss: 0.0041
Epoch 10/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0122
- val_loss: 0.0059
Epoch 11/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0112
- val_loss: 0.0045
Epoch 12/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0106
- val_loss: 0.0049
Epoch 13/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0104
- val_loss: 0.0039
Epoch 14/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0097
- val_loss: 0.0053
Epoch 15/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0096
- val_loss: 0.0050
Epoch 16/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0094
- val_loss: 0.0039
Epoch 17/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0091
- val_loss: 0.0037
Epoch 18/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0087

```
- val_loss: 0.0041
Epoch 19/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0087
- val_loss: 0.0038
Epoch 20/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0084
- val_loss: 0.0039
Epoch 21/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0081
- val_loss: 0.0041
Epoch 22/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0080
- val_loss: 0.0035
Epoch 23/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0077
- val_loss: 0.0037
Epoch 24/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0078
- val_loss: 0.0037
Epoch 25/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0075
- val_loss: 0.0034
Epoch 26/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0073
- val_loss: 0.0034
Epoch 27/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0073
- val_loss: 0.0039
Epoch 28/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0071
- val_loss: 0.0033
Epoch 29/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0069
- val_loss: 0.0034
Epoch 30/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0068
- val_loss: 0.0033
```

```
Training model with spec model_Adam0.001he_uniform (Combo: 32/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 1.0385 -
val_loss: 0.0670
Epoch 2/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0788
- val_loss: 0.0357
Epoch 3/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0536
- val_loss: 0.0194
Epoch 4/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0407
- val_loss: 0.0151
Epoch 5/30
1712/1712 [=====] - 1s 694us/step - loss: 0.0347
- val_loss: 0.0109
```

```
Epoch 6/30
1712/1712 [=====] - 1s 703us/step - loss: 0.0302
- val_loss: 0.0096
Epoch 7/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0266
- val_loss: 0.0139
Epoch 8/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0240
- val_loss: 0.0065
Epoch 9/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0226
- val_loss: 0.0076
Epoch 10/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0210
- val_loss: 0.0076
Epoch 11/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0188
- val_loss: 0.0065
Epoch 12/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0192
- val_loss: 0.0075
Epoch 13/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0173
- val_loss: 0.0063
Epoch 14/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0165
- val_loss: 0.0053
Epoch 15/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0164
- val_loss: 0.0071
Epoch 16/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0164
- val_loss: 0.0056
Epoch 17/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0152
- val_loss: 0.0062
Epoch 18/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0152
- val_loss: 0.0058
Epoch 19/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0145
- val_loss: 0.0094
Epoch 20/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0145
- val_loss: 0.0073
Epoch 21/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0135
- val_loss: 0.0052
Epoch 22/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0135
- val_loss: 0.0057
Epoch 23/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0130
- val_loss: 0.0064
Epoch 24/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0124
- val_loss: 0.0045
```

```
Epoch 25/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0123
- val_loss: 0.0054
Epoch 26/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0124
- val_loss: 0.0062
Epoch 27/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0117
- val_loss: 0.0044
Epoch 28/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0112
- val_loss: 0.0057
Epoch 29/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0115
- val_loss: 0.0047
Epoch 30/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0116
- val_loss: 0.0047
```

```
Training model with spec model_Adam0.0009glorot_normal (Combo: 33/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.0581 -
val_loss: 0.0200
Epoch 2/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0265
- val_loss: 0.0085
Epoch 3/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0178
- val_loss: 0.0066
Epoch 4/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0142
- val_loss: 0.0050
Epoch 5/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0122
- val_loss: 0.0045
Epoch 6/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0111
- val_loss: 0.0046
Epoch 7/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0103
- val_loss: 0.0046
Epoch 8/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0098
- val_loss: 0.0043
Epoch 9/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0094
- val_loss: 0.0053
Epoch 10/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0091
- val_loss: 0.0043
Epoch 11/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0086
- val_loss: 0.0041
Epoch 12/30
```

```
1712/1712 [=====] - 1s 710us/step - loss: 0.0082
  - val_loss: 0.0041
Epoch 13/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0077
  - val_loss: 0.0043
Epoch 14/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0076
  - val_loss: 0.0040
Epoch 15/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0075
  - val_loss: 0.0041
Epoch 16/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0070
  - val_loss: 0.0041
Epoch 17/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0068
  - val_loss: 0.0038
Epoch 18/30
1712/1712 [=====] - 1s 661us/step - loss: 0.0066
  - val_loss: 0.0042
Epoch 19/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0065
  - val_loss: 0.0036
Epoch 20/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0063
  - val_loss: 0.0036
Epoch 21/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0061
  - val_loss: 0.0033
Epoch 22/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0059
  - val_loss: 0.0035
Epoch 23/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0056
  - val_loss: 0.0032
Epoch 24/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0055
  - val_loss: 0.0032
Epoch 25/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0056
  - val_loss: 0.0032
Epoch 26/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0053
  - val_loss: 0.0029
Epoch 27/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0051
  - val_loss: 0.0029
Epoch 28/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0050
  - val_loss: 0.0027
Epoch 29/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0049
  - val_loss: 0.0028
Epoch 30/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0048
  - val_loss: 0.0027
```

```
Training model with spec model_Adam0.0009glorot_uniform (Combo: 34/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.0580 -
  val_loss: 0.0186
Epoch 2/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0249
  - val_loss: 0.0065
Epoch 3/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0171
  - val_loss: 0.0054
Epoch 4/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0140
  - val_loss: 0.0065
Epoch 5/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0122
  - val_loss: 0.0042
Epoch 6/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0111
  - val_loss: 0.0042
Epoch 7/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0102
  - val_loss: 0.0050
Epoch 8/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0097
  - val_loss: 0.0041
Epoch 9/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0091
  - val_loss: 0.0039
Epoch 10/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0087
  - val_loss: 0.0048
Epoch 11/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0083
  - val_loss: 0.0042
Epoch 12/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0077
  - val_loss: 0.0036
Epoch 13/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0075
  - val_loss: 0.0037
Epoch 14/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0072
  - val_loss: 0.0042
Epoch 15/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0070
  - val_loss: 0.0034
Epoch 16/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0067
  - val_loss: 0.0036
Epoch 17/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0064
  - val_loss: 0.0032
Epoch 18/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0060
```

```
- val_loss: 0.0033
Epoch 19/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0059
- val_loss: 0.0033
Epoch 20/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0057
- val_loss: 0.0029
Epoch 21/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0055
- val_loss: 0.0027
Epoch 22/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0054
- val_loss: 0.0029
Epoch 23/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0053
- val_loss: 0.0026
Epoch 24/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0051
- val_loss: 0.0025
Epoch 25/30
1712/1712 [=====] - 1s 694us/step - loss: 0.0050
- val_loss: 0.0025
Epoch 26/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0048
- val_loss: 0.0024
Epoch 27/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0047
- val_loss: 0.0024
Epoch 28/30
1712/1712 [=====] - 1s 698us/step - loss: 0.0047
- val_loss: 0.0024
Epoch 29/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0046
- val_loss: 0.0023
Epoch 30/30
1712/1712 [=====] - 1s 722us/step - loss: 0.0044
- val_loss: 0.0023
```

```
Training model with spec model_Adam0.009he_normal (Combo: 35/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.1739 -
val_loss: 0.0268
Epoch 2/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0404
- val_loss: 0.0161
Epoch 3/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0292
- val_loss: 0.0114
Epoch 4/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0240
- val_loss: 0.0070
Epoch 5/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0196
- val_loss: 0.0066
```

```
Epoch 6/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0168
- val_loss: 0.0056
Epoch 7/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0147
- val_loss: 0.0049
Epoch 8/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0134
- val_loss: 0.0042
Epoch 9/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0126
- val_loss: 0.0049
Epoch 10/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0118
- val_loss: 0.0047
Epoch 11/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0114
- val_loss: 0.0048
Epoch 12/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0108
- val_loss: 0.0051
Epoch 13/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0106
- val_loss: 0.0044
Epoch 14/30
1712/1712 [=====] - 1s 663us/step - loss: 0.0104
- val_loss: 0.0062
Epoch 15/30
1712/1712 [=====] - 1s 723us/step - loss: 0.0098
- val_loss: 0.0038
Epoch 16/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0091
- val_loss: 0.0044
Epoch 17/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0093
- val_loss: 0.0038
Epoch 18/30
1712/1712 [=====] - 1s 698us/step - loss: 0.0090
- val_loss: 0.0037
Epoch 19/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0089
- val_loss: 0.0047
Epoch 20/30
1712/1712 [=====] - 1s 722us/step - loss: 0.0085
- val_loss: 0.0037
Epoch 21/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0082
- val_loss: 0.0036
Epoch 22/30
1712/1712 [=====] - 1s 661us/step - loss: 0.0083
- val_loss: 0.0037
Epoch 23/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0080
- val_loss: 0.0040
Epoch 24/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0079
- val_loss: 0.0043
```

```
Epoch 25/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0080
- val_loss: 0.0035
Epoch 26/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0073
- val_loss: 0.0037
Epoch 27/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0072
- val_loss: 0.0036
Epoch 28/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0072
- val_loss: 0.0034
Epoch 29/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0071
- val_loss: 0.0039
Epoch 30/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0071
- val_loss: 0.0039
```

```
Training model with spec model_Adam0.0009he_uniform (Combo: 36/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.1977 -
val_loss: 0.0285
Epoch 2/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0440
- val_loss: 0.0129
Epoch 3/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0314
- val_loss: 0.0089
Epoch 4/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0247
- val_loss: 0.0073
Epoch 5/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0204
- val_loss: 0.0086
Epoch 6/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0177
- val_loss: 0.0047
Epoch 7/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0165
- val_loss: 0.0055
Epoch 8/30
1712/1712 [=====] - 1s 722us/step - loss: 0.0143
- val_loss: 0.0043
Epoch 9/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0134
- val_loss: 0.0041
Epoch 10/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0128
- val_loss: 0.0055
Epoch 11/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0119
- val_loss: 0.0038
Epoch 12/30
```

```
1712/1712 [=====] - 1s 656us/step - loss: 0.0117
  - val_loss: 0.0057
Epoch 13/30
1712/1712 [=====] - 1s 659us/step - loss: 0.0108
  - val_loss: 0.0045
Epoch 14/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0105
  - val_loss: 0.0040
Epoch 15/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0096
  - val_loss: 0.0035
Epoch 16/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0095
  - val_loss: 0.0037
Epoch 17/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0090
  - val_loss: 0.0036
Epoch 18/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0088
  - val_loss: 0.0055
Epoch 19/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0086
  - val_loss: 0.0034
Epoch 20/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0083
  - val_loss: 0.0042
Epoch 21/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0080
  - val_loss: 0.0035
Epoch 22/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0077
  - val_loss: 0.0040
Epoch 23/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0075
  - val_loss: 0.0033
Epoch 24/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0075
  - val_loss: 0.0035
Epoch 25/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0074
  - val_loss: 0.0034
Epoch 26/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0074
  - val_loss: 0.0031
Epoch 27/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0071
  - val_loss: 0.0033
Epoch 28/30
1712/1712 [=====] - 1s 660us/step - loss: 0.0070
  - val_loss: 0.0034
Epoch 29/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0068
  - val_loss: 0.0030
Epoch 30/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0067
  - val_loss: 0.0030
```

```
Training model with spec model_Adamax0.002glorot_normal (Combo: 37/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.0501 -
  val_loss: 0.0092
Epoch 2/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0196
  - val_loss: 0.0065
Epoch 3/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0147
  - val_loss: 0.0059
Epoch 4/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0126
  - val_loss: 0.0052
Epoch 5/30
1712/1712 [=====] - 1s 691us/step - loss: 0.0113
  - val_loss: 0.0043
Epoch 6/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0106
  - val_loss: 0.0045
Epoch 7/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0099
  - val_loss: 0.0050
Epoch 8/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0095
  - val_loss: 0.0041
Epoch 9/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0092
  - val_loss: 0.0043
Epoch 10/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0086
  - val_loss: 0.0047
Epoch 11/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0086
  - val_loss: 0.0045
Epoch 12/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0082
  - val_loss: 0.0038
Epoch 13/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0080
  - val_loss: 0.0039
Epoch 14/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0077
  - val_loss: 0.0042
Epoch 15/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0073
  - val_loss: 0.0041
Epoch 16/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0076
  - val_loss: 0.0037
Epoch 17/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0074
  - val_loss: 0.0037
Epoch 18/30
1712/1712 [=====] - 1s 692us/step - loss: 0.0070
```

```
- val_loss: 0.0036
Epoch 19/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0068
- val_loss: 0.0036
Epoch 20/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0067
- val_loss: 0.0039
Epoch 21/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0066
- val_loss: 0.0033
Epoch 22/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0065
- val_loss: 0.0036
Epoch 23/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0065
- val_loss: 0.0036
Epoch 24/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0062
- val_loss: 0.0030
Epoch 25/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0060
- val_loss: 0.0033
Epoch 26/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0059
- val_loss: 0.0035
Epoch 27/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0059
- val_loss: 0.0029
Epoch 28/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0058
- val_loss: 0.0033
Epoch 29/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0057
- val_loss: 0.0026
Epoch 30/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0057
- val_loss: 0.0026
```

```
Training model with spec model_Adamax0.002glorot_uniform (Combo: 38/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.0577 -
val_loss: 0.0088
Epoch 2/30
1712/1712 [=====] - 1s 704us/step - loss: 0.0247
- val_loss: 0.0078
Epoch 3/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0186
- val_loss: 0.0069
Epoch 4/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0153
- val_loss: 0.0051
Epoch 5/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0138
- val_loss: 0.0049
```

```
Epoch 6/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0129
- val_loss: 0.0049
Epoch 7/30
1712/1712 [=====] - 1s 644us/step - loss: 0.0119
- val_loss: 0.0049
Epoch 8/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0114
- val_loss: 0.0045
Epoch 9/30
1712/1712 [=====] - 1s 646us/step - loss: 0.0110
- val_loss: 0.0048
Epoch 10/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0106
- val_loss: 0.0044
Epoch 11/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0103
- val_loss: 0.0043
Epoch 12/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0101
- val_loss: 0.0049
Epoch 13/30
1712/1712 [=====] - 1s 645us/step - loss: 0.0097
- val_loss: 0.0044
Epoch 14/30
1712/1712 [=====] - 1s 645us/step - loss: 0.0094
- val_loss: 0.0048
Epoch 15/30
1712/1712 [=====] - 1s 645us/step - loss: 0.0094
- val_loss: 0.0047
Epoch 16/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0089
- val_loss: 0.0041
Epoch 17/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0088
- val_loss: 0.0043
Epoch 18/30
1712/1712 [=====] - 1s 646us/step - loss: 0.0087
- val_loss: 0.0043
Epoch 19/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0086
- val_loss: 0.0040
Epoch 20/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0084
- val_loss: 0.0039
Epoch 21/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0082
- val_loss: 0.0040
Epoch 22/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0079
- val_loss: 0.0043
Epoch 23/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0079
- val_loss: 0.0038
Epoch 24/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0077
- val_loss: 0.0039
```

```
Epoch 25/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0074
- val_loss: 0.0037
Epoch 26/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0074
- val_loss: 0.0046
Epoch 27/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0072
- val_loss: 0.0041
Epoch 28/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0073
- val_loss: 0.0040
Epoch 29/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0071
- val_loss: 0.0040
Epoch 30/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0068
- val_loss: 0.0036
```

```
Training model with spec model_Adamax0.002he_normal (Combo: 39/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 3s 2ms/step - loss: 0.3248 -
val_loss: 0.0373
Epoch 2/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0507
- val_loss: 0.0161
Epoch 3/30
1712/1712 [=====] - 1s 704us/step - loss: 0.0363
- val_loss: 0.0109
Epoch 4/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0290
- val_loss: 0.0097
Epoch 5/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0251
- val_loss: 0.0091
Epoch 6/30
1712/1712 [=====] - 1s 643us/step - loss: 0.0213
- val_loss: 0.0093
Epoch 7/30
1712/1712 [=====] - 1s 699us/step - loss: 0.0194
- val_loss: 0.0066
Epoch 8/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0178
- val_loss: 0.0060
Epoch 9/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0166
- val_loss: 0.0051
Epoch 10/30
1712/1712 [=====] - 1s 646us/step - loss: 0.0159
- val_loss: 0.0054
Epoch 11/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0145
- val_loss: 0.0058
Epoch 12/30
```

```
1712/1712 [=====] - 1s 710us/step - loss: 0.0142
  - val_loss: 0.0050
Epoch 13/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0134
  - val_loss: 0.0045
Epoch 14/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0125
  - val_loss: 0.0042
Epoch 15/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0133
  - val_loss: 0.0043
Epoch 16/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0128
  - val_loss: 0.0048
Epoch 17/30
1712/1712 [=====] - 1s 646us/step - loss: 0.0120
  - val_loss: 0.0060
Epoch 18/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0115
  - val_loss: 0.0041
Epoch 19/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0113
  - val_loss: 0.0061
Epoch 20/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0116
  - val_loss: 0.0042
Epoch 21/30
1712/1712 [=====] - 1s 703us/step - loss: 0.0108
  - val_loss: 0.0038
Epoch 22/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0109
  - val_loss: 0.0051
Epoch 23/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0106
  - val_loss: 0.0042
Epoch 24/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0102
  - val_loss: 0.0039
Epoch 25/30
1712/1712 [=====] - 1s 646us/step - loss: 0.0098
  - val_loss: 0.0040
Epoch 26/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0102
  - val_loss: 0.0036
Epoch 27/30
1712/1712 [=====] - 1s 644us/step - loss: 0.0101
  - val_loss: 0.0039
Epoch 28/30
1712/1712 [=====] - 1s 645us/step - loss: 0.0099
  - val_loss: 0.0045
Epoch 29/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0096
  - val_loss: 0.0036
Epoch 30/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0094
  - val_loss: 0.0044
```

Training model with spec model_Adamax0.002he_uniform (Combo: 40/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 1.2583 -
- val_loss: 0.0643
Epoch 2/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0760
- val_loss: 0.0310
Epoch 3/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0502
- val_loss: 0.0162
Epoch 4/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0375
- val_loss: 0.0139
Epoch 5/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0317
- val_loss: 0.0107
Epoch 6/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0281
- val_loss: 0.0107
Epoch 7/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0259
- val_loss: 0.0095
Epoch 8/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0241
- val_loss: 0.0093
Epoch 9/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0221
- val_loss: 0.0082
Epoch 10/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0209
- val_loss: 0.0082
Epoch 11/30
1712/1712 [=====] - 1s 695us/step - loss: 0.0198
- val_loss: 0.0073
Epoch 12/30
1712/1712 [=====] - 1s 704us/step - loss: 0.0195
- val_loss: 0.0066
Epoch 13/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0186
- val_loss: 0.0064
Epoch 14/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0177
- val_loss: 0.0084
Epoch 15/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0181
- val_loss: 0.0086
Epoch 16/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0177
- val_loss: 0.0088
Epoch 17/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0169
- val_loss: 0.0073
Epoch 18/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0160

```
- val_loss: 0.0065
Epoch 19/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0151
- val_loss: 0.0067
Epoch 20/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0157
- val_loss: 0.0056
Epoch 21/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0153
- val_loss: 0.0053
Epoch 22/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0147
- val_loss: 0.0055
Epoch 23/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0151
- val_loss: 0.0064
Epoch 24/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0146
- val_loss: 0.0056
Epoch 25/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0149
- val_loss: 0.0054
Epoch 26/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0143
- val_loss: 0.0053
Epoch 27/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0141
- val_loss: 0.0048
Epoch 28/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0142
- val_loss: 0.0050
Epoch 29/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0136
- val_loss: 0.0050
Epoch 30/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0138
- val_loss: 0.0049
```

```
Training model with spec model_Adamax0.001glorot_normal (Combo: 41/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.0673 -
val_loss: 0.0101
Epoch 2/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0312
- val_loss: 0.0090
Epoch 3/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0229
- val_loss: 0.0080
Epoch 4/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0193
- val_loss: 0.0061
Epoch 5/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0168
- val_loss: 0.0058
```

```
Epoch 6/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0153
- val_loss: 0.0051
Epoch 7/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0146
- val_loss: 0.0049
Epoch 8/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0136
- val_loss: 0.0057
Epoch 9/30
1712/1712 [=====] - 1s 693us/step - loss: 0.0127
- val_loss: 0.0049
Epoch 10/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0124
- val_loss: 0.0056
Epoch 11/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0119
- val_loss: 0.0051
Epoch 12/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0115
- val_loss: 0.0045
Epoch 13/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0110
- val_loss: 0.0045
Epoch 14/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0108
- val_loss: 0.0045
Epoch 15/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0104
- val_loss: 0.0047
Epoch 16/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0102
- val_loss: 0.0053
Epoch 17/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0101
- val_loss: 0.0048
Epoch 18/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0094
- val_loss: 0.0042
Epoch 19/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0095
- val_loss: 0.0045
Epoch 20/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0091
- val_loss: 0.0043
Epoch 21/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0089
- val_loss: 0.0038
Epoch 22/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0089
- val_loss: 0.0038
Epoch 23/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0085
- val_loss: 0.0040
Epoch 24/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0083
- val_loss: 0.0040
```

```
Epoch 25/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0083
- val_loss: 0.0035
Epoch 26/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0080
- val_loss: 0.0034
Epoch 27/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0077
- val_loss: 0.0035
Epoch 28/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0076
- val_loss: 0.0034
Epoch 29/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0075
- val_loss: 0.0034
Epoch 30/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0072
- val_loss: 0.0035
```

```
Training model with spec model_Adamax0.001glorot_uniform (Combo: 42/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.0549 -
val_loss: 0.0103
Epoch 2/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0286
- val_loss: 0.0114
Epoch 3/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0224
- val_loss: 0.0083
Epoch 4/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0188
- val_loss: 0.0068
Epoch 5/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0166
- val_loss: 0.0064
Epoch 6/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0154
- val_loss: 0.0048
Epoch 7/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0141
- val_loss: 0.0053
Epoch 8/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0136
- val_loss: 0.0057
Epoch 9/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0127
- val_loss: 0.0057
Epoch 10/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0121
- val_loss: 0.0044
Epoch 11/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0117
- val_loss: 0.0045
Epoch 12/30
```

```
1712/1712 [=====] - 1s 711us/step - loss: 0.0113
  - val_loss: 0.0041
Epoch 13/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0109
  - val_loss: 0.0043
Epoch 14/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0107
  - val_loss: 0.0042
Epoch 15/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0103
  - val_loss: 0.0044
Epoch 16/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0101
  - val_loss: 0.0044
Epoch 17/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0097
  - val_loss: 0.0039
Epoch 18/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0097
  - val_loss: 0.0039
Epoch 19/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0095
  - val_loss: 0.0038
Epoch 20/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0093
  - val_loss: 0.0039
Epoch 21/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0091
  - val_loss: 0.0041
Epoch 22/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0089
  - val_loss: 0.0042
Epoch 23/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0087
  - val_loss: 0.0040
Epoch 24/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0085
  - val_loss: 0.0039
Epoch 25/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0084
  - val_loss: 0.0038
Epoch 26/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0082
  - val_loss: 0.0037
Epoch 27/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0080
  - val_loss: 0.0046
Epoch 28/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0079
  - val_loss: 0.0038
Epoch 29/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0080
  - val_loss: 0.0034
Epoch 30/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0074
  - val_loss: 0.0033
```

Training model with spec model_Adamax0.001he_normal (Combo: 43/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.0871 -
- val_loss: 0.0210
Epoch 2/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0382
- val_loss: 0.0133
Epoch 3/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0310
- val_loss: 0.0101
Epoch 4/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0265
- val_loss: 0.0098
Epoch 5/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0232
- val_loss: 0.0080
Epoch 6/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0203
- val_loss: 0.0072
Epoch 7/30
1712/1712 [=====] - 1s 701us/step - loss: 0.0182
- val_loss: 0.0061
Epoch 8/30
1712/1712 [=====] - 1s 706us/step - loss: 0.0173
- val_loss: 0.0055
Epoch 9/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0157
- val_loss: 0.0057
Epoch 10/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0147
- val_loss: 0.0053
Epoch 11/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0141
- val_loss: 0.0044
Epoch 12/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0133
- val_loss: 0.0045
Epoch 13/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0126
- val_loss: 0.0040
Epoch 14/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0122
- val_loss: 0.0044
Epoch 15/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0116
- val_loss: 0.0040
Epoch 16/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0114
- val_loss: 0.0043
Epoch 17/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0111
- val_loss: 0.0049
Epoch 18/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0106

```
- val_loss: 0.0040
Epoch 19/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0106
- val_loss: 0.0057
Epoch 20/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0105
- val_loss: 0.0033
Epoch 21/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0103
- val_loss: 0.0035
Epoch 22/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0096
- val_loss: 0.0056
Epoch 23/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0096
- val_loss: 0.0032
Epoch 24/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0095
- val_loss: 0.0032
Epoch 25/30
1712/1712 [=====] - 1s 655us/step - loss: 0.0092
- val_loss: 0.0042
Epoch 26/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0091
- val_loss: 0.0032
Epoch 27/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0089
- val_loss: 0.0031
Epoch 28/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0086
- val_loss: 0.0036
Epoch 29/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0086
- val_loss: 0.0035
Epoch 30/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0084
- val_loss: 0.0030
```

```
Training model with spec model_Adamax0.001he_uniform (Combo: 44/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.4753 -
val_loss: 0.0602
Epoch 2/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0836
- val_loss: 0.0449
Epoch 3/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0706
- val_loss: 0.0365
Epoch 4/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0623
- val_loss: 0.0322
Epoch 5/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0545
- val_loss: 0.0271
```

```
Epoch 6/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0490
- val_loss: 0.0186
Epoch 7/30
1712/1712 [=====] - 1s 687us/step - loss: 0.0443
- val_loss: 0.0169
Epoch 8/30
1712/1712 [=====] - 1s 704us/step - loss: 0.0406
- val_loss: 0.0145
Epoch 9/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0371
- val_loss: 0.0118
Epoch 10/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0350
- val_loss: 0.0170
Epoch 11/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0326
- val_loss: 0.0110
Epoch 12/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0306
- val_loss: 0.0112
Epoch 13/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0287
- val_loss: 0.0103
Epoch 14/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0281
- val_loss: 0.0120
Epoch 15/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0271
- val_loss: 0.0128
Epoch 16/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0258
- val_loss: 0.0091
Epoch 17/30
1712/1712 [=====] - 1s 703us/step - loss: 0.0241
- val_loss: 0.0083
Epoch 18/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0236
- val_loss: 0.0072
Epoch 19/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0228
- val_loss: 0.0101
Epoch 20/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0218
- val_loss: 0.0094
Epoch 21/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0225
- val_loss: 0.0082
Epoch 22/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0212
- val_loss: 0.0064
Epoch 23/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0209
- val_loss: 0.0080
Epoch 24/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0201
- val_loss: 0.0057
```

```
Epoch 25/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0201
- val_loss: 0.0061
Epoch 26/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0195
- val_loss: 0.0077
Epoch 27/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0195
- val_loss: 0.0082
Epoch 28/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0199
- val_loss: 0.0055
Epoch 29/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0189
- val_loss: 0.0054
Epoch 30/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0188
- val_loss: 0.0075
```

```
Training model with spec model_Adamax0.0009glorot_normal (Combo: 45/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.0725 -
val_loss: 0.0147
Epoch 2/30
1712/1712 [=====] - 1s 696us/step - loss: 0.0326
- val_loss: 0.0084
Epoch 3/30
1712/1712 [=====] - 1s 701us/step - loss: 0.0248
- val_loss: 0.0075
Epoch 4/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0203
- val_loss: 0.0061
Epoch 5/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0178
- val_loss: 0.0055
Epoch 6/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0161
- val_loss: 0.0052
Epoch 7/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0152
- val_loss: 0.0059
Epoch 8/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0145
- val_loss: 0.0045
Epoch 9/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0136
- val_loss: 0.0054
Epoch 10/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0129
- val_loss: 0.0058
Epoch 11/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0128
- val_loss: 0.0043
Epoch 12/30
```

```
1712/1712 [=====] - 1s 655us/step - loss: 0.0121
  - val_loss: 0.0050
Epoch 13/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0116
  - val_loss: 0.0052
Epoch 14/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0113
  - val_loss: 0.0044
Epoch 15/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0108
  - val_loss: 0.0042
Epoch 16/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0108
  - val_loss: 0.0044
Epoch 17/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0104
  - val_loss: 0.0052
Epoch 18/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0102
  - val_loss: 0.0048
Epoch 19/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0099
  - val_loss: 0.0042
Epoch 20/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0095
  - val_loss: 0.0038
Epoch 21/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0095
  - val_loss: 0.0039
Epoch 22/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0091
  - val_loss: 0.0041
Epoch 23/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0090
  - val_loss: 0.0052
Epoch 24/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0090
  - val_loss: 0.0046
Epoch 25/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0087
  - val_loss: 0.0042
Epoch 26/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0083
  - val_loss: 0.0040
Epoch 27/30
1712/1712 [=====] - 1s 715us/step - loss: 0.0082
  - val_loss: 0.0034
Epoch 28/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0081
  - val_loss: 0.0035
Epoch 29/30
1712/1712 [=====] - 1s 716us/step - loss: 0.0079
  - val_loss: 0.0032
Epoch 30/30
1712/1712 [=====] - 1s 656us/step - loss: 0.0077
  - val_loss: 0.0033
```

Training model with spec model_Adamax0.0009glorot_uniform (Combo: 46/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.0651 -
- val_loss: 0.0153
Epoch 2/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0343
- val_loss: 0.0113
Epoch 3/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0262
- val_loss: 0.0093
Epoch 4/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0215
- val_loss: 0.0064
Epoch 5/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0194
- val_loss: 0.0067
Epoch 6/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0177
- val_loss: 0.0068
Epoch 7/30
1712/1712 [=====] - 1s 713us/step - loss: 0.0164
- val_loss: 0.0056
Epoch 8/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0153
- val_loss: 0.0043
Epoch 9/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0145
- val_loss: 0.0045
Epoch 10/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0136
- val_loss: 0.0044
Epoch 11/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0131
- val_loss: 0.0043
Epoch 12/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0125
- val_loss: 0.0045
Epoch 13/30
1712/1712 [=====] - 1s 646us/step - loss: 0.0120
- val_loss: 0.0051
Epoch 14/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0115
- val_loss: 0.0047
Epoch 15/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0114
- val_loss: 0.0047
Epoch 16/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0112
- val_loss: 0.0047
Epoch 17/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0107
- val_loss: 0.0042
Epoch 18/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0103

```
- val_loss: 0.0046
Epoch 19/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0101
- val_loss: 0.0039
Epoch 20/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0097
- val_loss: 0.0037
Epoch 21/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0098
- val_loss: 0.0041
Epoch 22/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0094
- val_loss: 0.0039
Epoch 23/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0093
- val_loss: 0.0039
Epoch 24/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0091
- val_loss: 0.0034
Epoch 25/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0088
- val_loss: 0.0034
Epoch 26/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0086
- val_loss: 0.0036
Epoch 27/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0085
- val_loss: 0.0035
Epoch 28/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0084
- val_loss: 0.0035
Epoch 29/30
1712/1712 [=====] - 1s 701us/step - loss: 0.0083
- val_loss: 0.0033
Epoch 30/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0080
- val_loss: 0.0034
```

```
Training model with spec model_Adamax0.0009he_normal (Combo: 47/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.1432 -
val_loss: 0.0256
Epoch 2/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0486
- val_loss: 0.0185
Epoch 3/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0389
- val_loss: 0.0145
Epoch 4/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0335
- val_loss: 0.0125
Epoch 5/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0302
- val_loss: 0.0105
```

```
Epoch 6/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0268
- val_loss: 0.0106
Epoch 7/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0247
- val_loss: 0.0102
Epoch 8/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0224
- val_loss: 0.0087
Epoch 9/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0209
- val_loss: 0.0060
Epoch 10/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0194
- val_loss: 0.0076
Epoch 11/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0182
- val_loss: 0.0060
Epoch 12/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0170
- val_loss: 0.0054
Epoch 13/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0159
- val_loss: 0.0050
Epoch 14/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0159
- val_loss: 0.0047
Epoch 15/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0147
- val_loss: 0.0045
Epoch 16/30
1712/1712 [=====] - 1s 657us/step - loss: 0.0148
- val_loss: 0.0051
Epoch 17/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0135
- val_loss: 0.0057
Epoch 18/30
1712/1712 [=====] - 1s 717us/step - loss: 0.0133
- val_loss: 0.0042
Epoch 19/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0127
- val_loss: 0.0041
Epoch 20/30
1712/1712 [=====] - 1s 654us/step - loss: 0.0123
- val_loss: 0.0052
Epoch 21/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0120
- val_loss: 0.0045
Epoch 22/30
1712/1712 [=====] - 1s 658us/step - loss: 0.0115
- val_loss: 0.0051
Epoch 23/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0113
- val_loss: 0.0042
Epoch 24/30
1712/1712 [=====] - 1s 708us/step - loss: 0.0109
- val_loss: 0.0037
```

```
Epoch 25/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0108
- val_loss: 0.0041
Epoch 26/30
1712/1712 [=====] - 1s 647us/step - loss: 0.0106
- val_loss: 0.0037
Epoch 27/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0107
- val_loss: 0.0052
Epoch 28/30
1712/1712 [=====] - 1s 652us/step - loss: 0.0103
- val_loss: 0.0048
Epoch 29/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0101
- val_loss: 0.0055
Epoch 30/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0101
- val_loss: 0.0064
```

```
Training model with spec model_Adamax0.0009he_uniform (Combo: 48/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.3619 -
val_loss: 0.0485
Epoch 2/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0717
- val_loss: 0.0324
Epoch 3/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0582
- val_loss: 0.0258
Epoch 4/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0504
- val_loss: 0.0208
Epoch 5/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0456
- val_loss: 0.0193
Epoch 6/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0408
- val_loss: 0.0146
Epoch 7/30
1712/1712 [=====] - 1s 704us/step - loss: 0.0375
- val_loss: 0.0139
Epoch 8/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0348
- val_loss: 0.0131
Epoch 9/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0325
- val_loss: 0.0099
Epoch 10/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0303
- val_loss: 0.0095
Epoch 11/30
1712/1712 [=====] - 1s 712us/step - loss: 0.0286
- val_loss: 0.0095
Epoch 12/30
```

```
1712/1712 [=====] - 1s 700us/step - loss: 0.0269
  - val_loss: 0.0093
Epoch 13/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0256
  - val_loss: 0.0097
Epoch 14/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0241
  - val_loss: 0.0078
Epoch 15/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0232
  - val_loss: 0.0072
Epoch 16/30
1712/1712 [=====] - 1s 653us/step - loss: 0.0229
  - val_loss: 0.0072
Epoch 17/30
1712/1712 [=====] - 1s 709us/step - loss: 0.0223
  - val_loss: 0.0071
Epoch 18/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0218
  - val_loss: 0.0087
Epoch 19/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0207
  - val_loss: 0.0075
Epoch 20/30
1712/1712 [=====] - 1s 711us/step - loss: 0.0200
  - val_loss: 0.0064
Epoch 21/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0195
  - val_loss: 0.0082
Epoch 22/30
1712/1712 [=====] - 1s 650us/step - loss: 0.0195
  - val_loss: 0.0064
Epoch 23/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0191
  - val_loss: 0.0089
Epoch 24/30
1712/1712 [=====] - 1s 710us/step - loss: 0.0189
  - val_loss: 0.0053
Epoch 25/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0182
  - val_loss: 0.0070
Epoch 26/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0177
  - val_loss: 0.0078
Epoch 27/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0180
  - val_loss: 0.0058
Epoch 28/30
1712/1712 [=====] - 1s 649us/step - loss: 0.0172
  - val_loss: 0.0073
Epoch 29/30
1712/1712 [=====] - 1s 648us/step - loss: 0.0168
  - val_loss: 0.0069
Epoch 30/30
1712/1712 [=====] - 1s 651us/step - loss: 0.0166
  - val_loss: 0.0055
```

Training model with spec model_Nadam0.002glorot_normal (Combo: 49/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.0608 -
- val_loss: 0.0076
Epoch 2/30
1712/1712 [=====] - 1s 736us/step - loss: 0.0213
- val_loss: 0.0053
Epoch 3/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0141
- val_loss: 0.0058
Epoch 4/30
1712/1712 [=====] - 1s 729us/step - loss: 0.0104
- val_loss: 0.0045
Epoch 5/30
1712/1712 [=====] - 1s 736us/step - loss: 0.0109
- val_loss: 0.0045
Epoch 6/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0094
- val_loss: 0.0048
Epoch 7/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0090
- val_loss: 0.0066
Epoch 8/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0084
- val_loss: 0.0044
Epoch 9/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0082
- val_loss: 0.0058
Epoch 10/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0074
- val_loss: 0.0044
Epoch 11/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0076
- val_loss: 0.0043
Epoch 12/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0074
- val_loss: 0.0065
Epoch 13/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0072
- val_loss: 0.0045
Epoch 14/30
1712/1712 [=====] - 1s 732us/step - loss: 0.0071
- val_loss: 0.0043
Epoch 15/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0067
- val_loss: 0.0044
Epoch 16/30
1712/1712 [=====] - 1s 726us/step - loss: 0.0062
- val_loss: 0.0041
Epoch 17/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0062
- val_loss: 0.0041
Epoch 18/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0063

```
- val_loss: 0.0042
Epoch 19/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0068
- val_loss: 0.0044
Epoch 20/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0060
- val_loss: 0.0042
Epoch 21/30
1712/1712 [=====] - 1s 731us/step - loss: 0.0060
- val_loss: 0.0040
Epoch 22/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0058
- val_loss: 0.0042
Epoch 23/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0054
- val_loss: 0.0042
Epoch 24/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0055
- val_loss: 0.0039
Epoch 25/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0054
- val_loss: 0.0049
Epoch 26/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0054
- val_loss: 0.0039
Epoch 27/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0050
- val_loss: 0.0039
Epoch 28/30
1712/1712 [=====] - 1s 731us/step - loss: 0.0049
- val_loss: 0.0038
Epoch 29/30
1712/1712 [=====] - 1s 728us/step - loss: 0.0050
- val_loss: 0.0037
Epoch 30/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0050
- val_loss: 0.0045
```

```
Training model with spec model_Nadam0.002glorot_uniform (Combo: 50/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 0.0608 -
val_loss: 0.0071
Epoch 2/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0229
- val_loss: 0.0054
Epoch 3/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0136
- val_loss: 0.0048
Epoch 4/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0121
- val_loss: 0.0047
Epoch 5/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0105
- val_loss: 0.0044
```

```
Epoch 6/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0097
- val_loss: 0.0043
Epoch 7/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0083
- val_loss: 0.0048
Epoch 8/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0086
- val_loss: 0.0046
Epoch 9/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0080
- val_loss: 0.0043
Epoch 10/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0078
- val_loss: 0.0044
Epoch 11/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0071
- val_loss: 0.0043
Epoch 12/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0071
- val_loss: 0.0043
Epoch 13/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0071
- val_loss: 0.0041
Epoch 14/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0068
- val_loss: 0.0042
Epoch 15/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0067
- val_loss: 0.0041
Epoch 16/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0060
- val_loss: 0.0039
Epoch 17/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0062
- val_loss: 0.0055
Epoch 18/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0062
- val_loss: 0.0044
Epoch 19/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0056
- val_loss: 0.0042
Epoch 20/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0056
- val_loss: 0.0040
Epoch 21/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0057
- val_loss: 0.0040
Epoch 22/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0057
- val_loss: 0.0039
Epoch 23/30
1712/1712 [=====] - 1s 728us/step - loss: 0.0052
- val_loss: 0.0039
Epoch 24/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0052
- val_loss: 0.0037
```

```
Epoch 25/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0050
- val_loss: 0.0035
Epoch 26/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0048
- val_loss: 0.0034
Epoch 27/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0047
- val_loss: 0.0032
Epoch 28/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0044
- val_loss: 0.0030
Epoch 29/30
1712/1712 [=====] - 1s 744us/step - loss: 0.0044
- val_loss: 0.0028
Epoch 30/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0042
- val_loss: 0.0032
```

```
Training model with spec model_Nadam0.002he_normal (Combo: 51/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 3ms/step - loss: 0.4618 -
val_loss: 0.0152
Epoch 2/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0397
- val_loss: 0.0094
Epoch 3/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0280
- val_loss: 0.0116
Epoch 4/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0217
- val_loss: 0.0138
Epoch 5/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0173
- val_loss: 0.0060
Epoch 6/30
1712/1712 [=====] - 1s 730us/step - loss: 0.0183
- val_loss: 0.0057
Epoch 7/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0143
- val_loss: 0.0057
Epoch 8/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0135
- val_loss: 0.0059
Epoch 9/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0131
- val_loss: 0.0058
Epoch 10/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0114
- val_loss: 0.0067
Epoch 11/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0107
- val_loss: 0.0051
Epoch 12/30
```

```
1712/1712 [=====] - 1s 678us/step - loss: 0.0112
  - val_loss: 0.0084
Epoch 13/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0101
  - val_loss: 0.0048
Epoch 14/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0093
  - val_loss: 0.0046
Epoch 15/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0091
  - val_loss: 0.0045
Epoch 16/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0089
  - val_loss: 0.0048
Epoch 17/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0089
  - val_loss: 0.0042
Epoch 18/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0086
  - val_loss: 0.0040
Epoch 19/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0078
  - val_loss: 0.0042
Epoch 20/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0079
  - val_loss: 0.0041
Epoch 21/30
1712/1712 [=====] - 1s 743us/step - loss: 0.0076
  - val_loss: 0.0039
Epoch 22/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0078
  - val_loss: 0.0042
Epoch 23/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0073
  - val_loss: 0.0038
Epoch 24/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0071
  - val_loss: 0.0040
Epoch 25/30
1712/1712 [=====] - 1s 736us/step - loss: 0.0070
  - val_loss: 0.0037
Epoch 26/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0068
  - val_loss: 0.0039
Epoch 27/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0065
  - val_loss: 0.0038
Epoch 28/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0065
  - val_loss: 0.0038
Epoch 29/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0065
  - val_loss: 0.0036
Epoch 30/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0062
  - val_loss: 0.0038
```

Training model with spec model_Nadam0.002he_uniform (Combo: 52/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 2ms/step - loss: 2.4146 -
- val_loss: 0.0462
Epoch 2/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0540
- val_loss: 0.0308
Epoch 3/30
1712/1712 [=====] - 1s 731us/step - loss: 0.0395
- val_loss: 0.0094
Epoch 4/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0330
- val_loss: 0.0123
Epoch 5/30
1712/1712 [=====] - 1s 731us/step - loss: 0.0281
- val_loss: 0.0076
Epoch 6/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0280
- val_loss: 0.0075
Epoch 7/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0227
- val_loss: 0.0064
Epoch 8/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0220
- val_loss: 0.0287
Epoch 9/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0206
- val_loss: 0.0117
Epoch 10/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0193
- val_loss: 0.0122
Epoch 11/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0181
- val_loss: 0.0057
Epoch 12/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0173
- val_loss: 0.0057
Epoch 13/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0170
- val_loss: 0.0051
Epoch 14/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0163
- val_loss: 0.0056
Epoch 15/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0153
- val_loss: 0.0053
Epoch 16/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0151
- val_loss: 0.0055
Epoch 17/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0145
- val_loss: 0.0076
Epoch 18/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0136

```
- val_loss: 0.0063
Epoch 19/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0139
- val_loss: 0.0048
Epoch 20/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0128
- val_loss: 0.0066
Epoch 21/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0128
- val_loss: 0.0057
Epoch 22/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0123
- val_loss: 0.0065
Epoch 23/30
1712/1712 [=====] - 1s 683us/step - loss: 0.0117
- val_loss: 0.0056
Epoch 24/30
1712/1712 [=====] - 1s 736us/step - loss: 0.0110
- val_loss: 0.0046
Epoch 25/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0113
- val_loss: 0.0050
Epoch 26/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0112
- val_loss: 0.0040
Epoch 27/30
1712/1712 [=====] - 1s 684us/step - loss: 0.0106
- val_loss: 0.0043
Epoch 28/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0102
- val_loss: 0.0041
Epoch 29/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0100
- val_loss: 0.0038
Epoch 30/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0101
- val_loss: 0.0044
```

```
Training model with spec model_Nadam0.001glorot_normal (Combo: 53/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 0.0558 -
val_loss: 0.0093
Epoch 2/30
1712/1712 [=====] - 1s 684us/step - loss: 0.0286
- val_loss: 0.0105
Epoch 3/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0204
- val_loss: 0.0285
Epoch 4/30
1712/1712 [=====] - 1s 744us/step - loss: 0.0160
- val_loss: 0.0071
Epoch 5/30
1712/1712 [=====] - 1s 743us/step - loss: 0.0145
- val_loss: 0.0060
```

```
Epoch 6/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0116
- val_loss: 0.0069
Epoch 7/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0114
- val_loss: 0.0046
Epoch 8/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0104
- val_loss: 0.0047
Epoch 9/30
1712/1712 [=====] - 1s 720us/step - loss: 0.0098
- val_loss: 0.0043
Epoch 10/30
1712/1712 [=====] - 1s 729us/step - loss: 0.0094
- val_loss: 0.0042
Epoch 11/30
1712/1712 [=====] - 1s 727us/step - loss: 0.0089
- val_loss: 0.0041
Epoch 12/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0091
- val_loss: 0.0054
Epoch 13/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0080
- val_loss: 0.0049
Epoch 14/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0084
- val_loss: 0.0048
Epoch 15/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0075
- val_loss: 0.0040
Epoch 16/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0077
- val_loss: 0.0040
Epoch 17/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0075
- val_loss: 0.0040
Epoch 18/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0076
- val_loss: 0.0091
Epoch 19/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0074
- val_loss: 0.0041
Epoch 20/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0067
- val_loss: 0.0040
Epoch 21/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0070
- val_loss: 0.0042
Epoch 22/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0065
- val_loss: 0.0038
Epoch 23/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0064
- val_loss: 0.0037
Epoch 24/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0061
- val_loss: 0.0036
```

```
Epoch 25/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0065
- val_loss: 0.0037
Epoch 26/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0060
- val_loss: 0.0036
Epoch 27/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0060
- val_loss: 0.0039
Epoch 28/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0060
- val_loss: 0.0058
Epoch 29/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0058
- val_loss: 0.0045
Epoch 30/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0056
- val_loss: 0.0033
```

```
Training model with spec model_Nadam0.001glorot_uniform (Combo: 54/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 4s 3ms/step - loss: 0.0595 -
val_loss: 0.0092
Epoch 2/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0298
- val_loss: 0.0130
Epoch 3/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0220
- val_loss: 0.0086
Epoch 4/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0181
- val_loss: 0.0122
Epoch 5/30
1712/1712 [=====] - 1s 730us/step - loss: 0.0159
- val_loss: 0.0056
Epoch 6/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0131
- val_loss: 0.0062
Epoch 7/30
1712/1712 [=====] - 1s 728us/step - loss: 0.0120
- val_loss: 0.0045
Epoch 8/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0110
- val_loss: 0.0045
Epoch 9/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0100
- val_loss: 0.0045
Epoch 10/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0096
- val_loss: 0.0038
Epoch 11/30
1712/1712 [=====] - 1s 736us/step - loss: 0.0089
- val_loss: 0.0038
Epoch 12/30
```

```
1712/1712 [=====] - 1s 672us/step - loss: 0.0092
  - val_loss: 0.0041
Epoch 13/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0085
  - val_loss: 0.0038
Epoch 14/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0077
  - val_loss: 0.0065
Epoch 15/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0082
  - val_loss: 0.0049
Epoch 16/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0075
  - val_loss: 0.0051
Epoch 17/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0071
  - val_loss: 0.0034
Epoch 18/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0073
  - val_loss: 0.0032
Epoch 19/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0064
  - val_loss: 0.0031
Epoch 20/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0065
  - val_loss: 0.0039
Epoch 21/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0064
  - val_loss: 0.0054
Epoch 22/30
1712/1712 [=====] - 1s 743us/step - loss: 0.0062
  - val_loss: 0.0028
Epoch 23/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0057
  - val_loss: 0.0036
Epoch 24/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0059
  - val_loss: 0.0031
Epoch 25/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0056
  - val_loss: 0.0034
Epoch 26/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0054
  - val_loss: 0.0029
Epoch 27/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0054
  - val_loss: 0.0025
Epoch 28/30
1712/1712 [=====] - 1s 684us/step - loss: 0.0051
  - val_loss: 0.0025
Epoch 29/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0050
  - val_loss: 0.0028
Epoch 30/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0049
  - val_loss: 0.0037
```

Training model with spec model_Nadam0.001he_normal (Combo: 55/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 0.2724 -
- val_loss: 0.0303
Epoch 2/30
1712/1712 [=====] - 1s 745us/step - loss: 0.0483
- val_loss: 0.0133
Epoch 3/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0391
- val_loss: 0.0123
Epoch 4/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0322
- val_loss: 0.0098
Epoch 5/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0287
- val_loss: 0.0077
Epoch 6/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0251
- val_loss: 0.0095
Epoch 7/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0225
- val_loss: 0.0072
Epoch 8/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0208
- val_loss: 0.0174
Epoch 9/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0192
- val_loss: 0.0128
Epoch 10/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0173
- val_loss: 0.0105
Epoch 11/30
1712/1712 [=====] - 1s 743us/step - loss: 0.0174
- val_loss: 0.0056
Epoch 12/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0153
- val_loss: 0.0086
Epoch 13/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0141
- val_loss: 0.0053
Epoch 14/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0131
- val_loss: 0.0069
Epoch 15/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0133
- val_loss: 0.0045
Epoch 16/30
1712/1712 [=====] - 1s 736us/step - loss: 0.0127
- val_loss: 0.0043
Epoch 17/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0120
- val_loss: 0.0044
Epoch 18/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0113

```
- val_loss: 0.0041
Epoch 19/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0108
- val_loss: 0.0093
Epoch 20/30
1712/1712 [=====] - 1s 743us/step - loss: 0.0107
- val_loss: 0.0039
Epoch 21/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0100
- val_loss: 0.0039
Epoch 22/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0098
- val_loss: 0.0046
Epoch 23/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0095
- val_loss: 0.0048
Epoch 24/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0093
- val_loss: 0.0039
Epoch 25/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0092
- val_loss: 0.0037
Epoch 26/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0086
- val_loss: 0.0066
Epoch 27/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0085
- val_loss: 0.0045
Epoch 28/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0083
- val_loss: 0.0040
Epoch 29/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0082
- val_loss: 0.0053
Epoch 30/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0080
- val_loss: 0.0036
```

```
Training model with spec model_Nadam0.001he_uniform (Combo: 56/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 0.1970 -
val_loss: 0.0226
Epoch 2/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0460
- val_loss: 0.0343
Epoch 3/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0367
- val_loss: 0.0129
Epoch 4/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0318
- val_loss: 0.0080
Epoch 5/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0273
- val_loss: 0.0132
```

```
Epoch 6/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0236
- val_loss: 0.0089
Epoch 7/30
1712/1712 [=====] - 1s 745us/step - loss: 0.0208
- val_loss: 0.0069
Epoch 8/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0200
- val_loss: 0.0082
Epoch 9/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0181
- val_loss: 0.0053
Epoch 10/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0170
- val_loss: 0.0062
Epoch 11/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0157
- val_loss: 0.0061
Epoch 12/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0141
- val_loss: 0.0054
Epoch 13/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0129
- val_loss: 0.0042
Epoch 14/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0125
- val_loss: 0.0040
Epoch 15/30
1712/1712 [=====] - 1s 744us/step - loss: 0.0120
- val_loss: 0.0040
Epoch 16/30
1712/1712 [=====] - 1s 684us/step - loss: 0.0117
- val_loss: 0.0040
Epoch 17/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0103
- val_loss: 0.0038
Epoch 18/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0104
- val_loss: 0.0067
Epoch 19/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0099
- val_loss: 0.0047
Epoch 20/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0097
- val_loss: 0.0036
Epoch 21/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0091
- val_loss: 0.0045
Epoch 22/30
1712/1712 [=====] - 1s 719us/step - loss: 0.0088
- val_loss: 0.0035
Epoch 23/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0082
- val_loss: 0.0065
Epoch 24/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0082
- val_loss: 0.0037
```

```
Epoch 25/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0076
- val_loss: 0.0031
Epoch 26/30
1712/1712 [=====] - 1s 732us/step - loss: 0.0079
- val_loss: 0.0031
Epoch 27/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0073
- val_loss: 0.0045
Epoch 28/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0071
- val_loss: 0.0031
Epoch 29/30
1712/1712 [=====] - 1s 731us/step - loss: 0.0069
- val_loss: 0.0030
Epoch 30/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0066
- val_loss: 0.0031
```

```
Training model with spec model_Nadam0.0009glorot_normal (Combo: 57/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 0.0569 -
val_loss: 0.0095
Epoch 2/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0291
- val_loss: 0.0067
Epoch 3/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0219
- val_loss: 0.0057
Epoch 4/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0163
- val_loss: 0.0061
Epoch 5/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0141
- val_loss: 0.0046
Epoch 6/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0131
- val_loss: 0.0044
Epoch 7/30
1712/1712 [=====] - 1s 741us/step - loss: 0.0111
- val_loss: 0.0044
Epoch 8/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0106
- val_loss: 0.0044
Epoch 9/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0102
- val_loss: 0.0065
Epoch 10/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0097
- val_loss: 0.0063
Epoch 11/30
1712/1712 [=====] - 1s 739us/step - loss: 0.0090
- val_loss: 0.0043
Epoch 12/30
```

```
1712/1712 [=====] - 1s 742us/step - loss: 0.0086
  - val_loss: 0.0041
Epoch 13/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0086
  - val_loss: 0.0043
Epoch 14/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0080
  - val_loss: 0.0041
Epoch 15/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0079
  - val_loss: 0.0039
Epoch 16/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0078
  - val_loss: 0.0050
Epoch 17/30
1712/1712 [=====] - 1s 727us/step - loss: 0.0077
  - val_loss: 0.0037
Epoch 18/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0079
  - val_loss: 0.0041
Epoch 19/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0071
  - val_loss: 0.0042
Epoch 20/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0072
  - val_loss: 0.0040
Epoch 21/30
1712/1712 [=====] - 1s 681us/step - loss: 0.0068
  - val_loss: 0.0038
Epoch 22/30
1712/1712 [=====] - 1s 680us/step - loss: 0.0065
  - val_loss: 0.0039
Epoch 23/30
1712/1712 [=====] - 1s 742us/step - loss: 0.0066
  - val_loss: 0.0034
Epoch 24/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0062
  - val_loss: 0.0033
Epoch 25/30
1712/1712 [=====] - 1s 683us/step - loss: 0.0064
  - val_loss: 0.0043
Epoch 26/30
1712/1712 [=====] - 1s 740us/step - loss: 0.0062
  - val_loss: 0.0032
Epoch 27/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0055
  - val_loss: 0.0034
Epoch 28/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0057
  - val_loss: 0.0030
Epoch 29/30
1712/1712 [=====] - 1s 684us/step - loss: 0.0059
  - val_loss: 0.0031
Epoch 30/30
1712/1712 [=====] - 1s 682us/step - loss: 0.0053
  - val_loss: 0.0030
```

Training model with spec model_Nadam0.0009glorot_uniform (Combo: 58/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 0.0500 -
- val_loss: 0.0318
Epoch 2/30
1712/1712 [=====] - 1s 730us/step - loss: 0.0268
- val_loss: 0.0141
Epoch 3/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0210
- val_loss: 0.0096
Epoch 4/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0166
- val_loss: 0.0080
Epoch 5/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0139
- val_loss: 0.0063
Epoch 6/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0125
- val_loss: 0.0043
Epoch 7/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0107
- val_loss: 0.0049
Epoch 8/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0104
- val_loss: 0.0049
Epoch 9/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0099
- val_loss: 0.0070
Epoch 10/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0094
- val_loss: 0.0063
Epoch 11/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0094
- val_loss: 0.0053
Epoch 12/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0087
- val_loss: 0.0061
Epoch 13/30
1712/1712 [=====] - 1s 730us/step - loss: 0.0080
- val_loss: 0.0039
Epoch 14/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0078
- val_loss: 0.0038
Epoch 15/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0082
- val_loss: 0.0039
Epoch 16/30
1712/1712 [=====] - 1s 669us/step - loss: 0.0076
- val_loss: 0.0046
Epoch 17/30
1712/1712 [=====] - 1s 731us/step - loss: 0.0076
- val_loss: 0.0037
Epoch 18/30
1712/1712 [=====] - 1s 707us/step - loss: 0.0069

```
- val_loss: 0.0035
Epoch 19/30
1712/1712 [=====] - 1s 666us/step - loss: 0.0072
- val_loss: 0.0038
Epoch 20/30
1712/1712 [=====] - 1s 666us/step - loss: 0.0065
- val_loss: 0.0040
Epoch 21/30
1712/1712 [=====] - 1s 724us/step - loss: 0.0070
- val_loss: 0.0034
Epoch 22/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0063
- val_loss: 0.0038
Epoch 23/30
1712/1712 [=====] - 1s 721us/step - loss: 0.0059
- val_loss: 0.0032
Epoch 24/30
1712/1712 [=====] - 1s 664us/step - loss: 0.0059
- val_loss: 0.0033
Epoch 25/30
1712/1712 [=====] - 1s 725us/step - loss: 0.0057
- val_loss: 0.0030
Epoch 26/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0058
- val_loss: 0.0032
Epoch 27/30
1712/1712 [=====] - 1s 727us/step - loss: 0.0055
- val_loss: 0.0028
Epoch 28/30
1712/1712 [=====] - 1s 705us/step - loss: 0.0053
- val_loss: 0.0027
Epoch 29/30
1712/1712 [=====] - 1s 666us/step - loss: 0.0053
- val_loss: 0.0028
Epoch 30/30
1712/1712 [=====] - 1s 665us/step - loss: 0.0049
- val_loss: 0.0030
```

```
Training model with spec model_Nadam0.0009he_normal (Combo: 59/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 0.1283 -
val_loss: 0.0180
Epoch 2/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0375
- val_loss: 0.0218
Epoch 3/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0287
- val_loss: 0.0166
Epoch 4/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0254
- val_loss: 0.0116
Epoch 5/30
1712/1712 [=====] - 1s 732us/step - loss: 0.0203
- val_loss: 0.0064
```

```
Epoch 6/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0186
- val_loss: 0.0063
Epoch 7/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0161
- val_loss: 0.0099
Epoch 8/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0156
- val_loss: 0.0068
Epoch 9/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0145
- val_loss: 0.0086
Epoch 10/30
1712/1712 [=====] - 1s 727us/step - loss: 0.0124
- val_loss: 0.0047
Epoch 11/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0114
- val_loss: 0.0055
Epoch 12/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0113
- val_loss: 0.0047
Epoch 13/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0104
- val_loss: 0.0043
Epoch 14/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0101
- val_loss: 0.0071
Epoch 15/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0094
- val_loss: 0.0038
Epoch 16/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0089
- val_loss: 0.0041
Epoch 17/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0086
- val_loss: 0.0036
Epoch 18/30
1712/1712 [=====] - 1s 679us/step - loss: 0.0088
- val_loss: 0.0036
Epoch 19/30
1712/1712 [=====] - 1s 729us/step - loss: 0.0078
- val_loss: 0.0035
Epoch 20/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0078
- val_loss: 0.0056
Epoch 21/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0076
- val_loss: 0.0040
Epoch 22/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0071
- val_loss: 0.0032
Epoch 23/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0077
- val_loss: 0.0038
Epoch 24/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0071
- val_loss: 0.0033
```

```
Epoch 25/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0066
- val_loss: 0.0033
Epoch 26/30
1712/1712 [=====] - 1s 735us/step - loss: 0.0066
- val_loss: 0.0029
Epoch 27/30
1712/1712 [=====] - 1s 674us/step - loss: 0.0063
- val_loss: 0.0034
Epoch 28/30
1712/1712 [=====] - 1s 737us/step - loss: 0.0063
- val_loss: 0.0028
Epoch 29/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0059
- val_loss: 0.0028
Epoch 30/30
1712/1712 [=====] - 1s 675us/step - loss: 0.0058
- val_loss: 0.0031
```

```
Training model with spec model_Nadam0.0009he_uniform (Combo: 60/60)
Iteration 1/1
Train on 1712 samples, validate on 428 samples
Epoch 1/30
1712/1712 [=====] - 5s 3ms/step - loss: 1.1495 -
val_loss: 0.0683
Epoch 2/30
1712/1712 [=====] - 1s 733us/step - loss: 0.0889
- val_loss: 0.0544
Epoch 3/30
1712/1712 [=====] - 1s 731us/step - loss: 0.0782
- val_loss: 0.0474
Epoch 4/30
1712/1712 [=====] - 1s 730us/step - loss: 0.0650
- val_loss: 0.0352
Epoch 5/30
1712/1712 [=====] - 1s 714us/step - loss: 0.0576
- val_loss: 0.0241
Epoch 6/30
1712/1712 [=====] - 1s 718us/step - loss: 0.0515
- val_loss: 0.0185
Epoch 7/30
1712/1712 [=====] - 1s 676us/step - loss: 0.0453
- val_loss: 0.0262
Epoch 8/30
1712/1712 [=====] - 1s 678us/step - loss: 0.0411
- val_loss: 0.0260
Epoch 9/30
1712/1712 [=====] - 1s 738us/step - loss: 0.0423
- val_loss: 0.0116
Epoch 10/30
1712/1712 [=====] - 1s 677us/step - loss: 0.0360
- val_loss: 0.0149
Epoch 11/30
1712/1712 [=====] - 1s 673us/step - loss: 0.0328
- val_loss: 0.0156
Epoch 12/30
```

```
1712/1712 [=====] - 1s 672us/step - loss: 0.0308
  - val_loss: 0.0131
Epoch 13/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0298
  - val_loss: 0.0143
Epoch 14/30
1712/1712 [=====] - 1s 671us/step - loss: 0.0279
  - val_loss: 0.0142
Epoch 15/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0311
  - val_loss: 0.0111
Epoch 16/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0259
  - val_loss: 0.0158
Epoch 17/30
1712/1712 [=====] - 1s 734us/step - loss: 0.0241
  - val_loss: 0.0091
Epoch 18/30
1712/1712 [=====] - 1s 670us/step - loss: 0.0226
  - val_loss: 0.0091
Epoch 19/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0217
  - val_loss: 0.0103
Epoch 20/30
1712/1712 [=====] - 1s 727us/step - loss: 0.0203
  - val_loss: 0.0079
Epoch 21/30
1712/1712 [=====] - 1s 729us/step - loss: 0.0201
  - val_loss: 0.0060
Epoch 22/30
1712/1712 [=====] - 1s 668us/step - loss: 0.0190
  - val_loss: 0.0064
Epoch 23/30
1712/1712 [=====] - 1s 728us/step - loss: 0.0179
  - val_loss: 0.0059
Epoch 24/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0180
  - val_loss: 0.0059
Epoch 25/30
1712/1712 [=====] - 1s 667us/step - loss: 0.0167
  - val_loss: 0.0066
Epoch 26/30
1712/1712 [=====] - 1s 663us/step - loss: 0.0158
  - val_loss: 0.0074
Epoch 27/30
1712/1712 [=====] - 1s 727us/step - loss: 0.0148
  - val_loss: 0.0052
Epoch 28/30
1712/1712 [=====] - 1s 732us/step - loss: 0.0139
  - val_loss: 0.0050
Epoch 29/30
1712/1712 [=====] - 1s 672us/step - loss: 0.0139
  - val_loss: 0.0071
Epoch 30/30
1712/1712 [=====] - 1s 729us/step - loss: 0.0127
  - val_loss: 0.0048
```

The best model is model_Adam0.001glorot_uniform

Train on 1712 samples, validate on 428 samples
Epoch 1/200
1712/1712 [=====] - 5s 3ms/step - loss: 0.0516 -
val_loss: 0.0087
Epoch 2/200
1712/1712 [=====] - 1s 703us/step - loss: 0.0211
- val_loss: 0.0066
Epoch 3/200
1712/1712 [=====] - 1s 699us/step - loss: 0.0146
- val_loss: 0.0050
Epoch 4/200
1712/1712 [=====] - 1s 712us/step - loss: 0.0121
- val_loss: 0.0047
Epoch 5/200
1712/1712 [=====] - 1s 713us/step - loss: 0.0108
- val_loss: 0.0042
Epoch 6/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0098
- val_loss: 0.0043
Epoch 7/200
1712/1712 [=====] - 1s 714us/step - loss: 0.0092
- val_loss: 0.0040
Epoch 8/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0086
- val_loss: 0.0041
Epoch 9/200
1712/1712 [=====] - 1s 717us/step - loss: 0.0083
- val_loss: 0.0039
Epoch 10/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0078
- val_loss: 0.0039
Epoch 11/200
1712/1712 [=====] - 1s 717us/step - loss: 0.0073
- val_loss: 0.0036
Epoch 12/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0071
- val_loss: 0.0038
Epoch 13/200
1712/1712 [=====] - 1s 719us/step - loss: 0.0067
- val_loss: 0.0034
Epoch 14/200
1712/1712 [=====] - 1s 719us/step - loss: 0.0063
- val_loss: 0.0034
Epoch 15/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0060
- val_loss: 0.0036
Epoch 16/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0058
- val_loss: 0.0030
Epoch 17/200
1712/1712 [=====] - 1s 721us/step - loss: 0.0056
- val_loss: 0.0028
Epoch 18/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0056
- val_loss: 0.0030
Epoch 19/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0052

```
- val_loss: 0.0029
Epoch 20/200
1712/1712 [=====] - 1s 719us/step - loss: 0.0049
- val_loss: 0.0026
Epoch 21/200
1712/1712 [=====] - 1s 722us/step - loss: 0.0048
- val_loss: 0.0025
Epoch 22/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0048
- val_loss: 0.0024
Epoch 23/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0046
- val_loss: 0.0030
Epoch 24/200
1712/1712 [=====] - 1s 722us/step - loss: 0.0046
- val_loss: 0.0022
Epoch 25/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0044
- val_loss: 0.0022
Epoch 26/200
1712/1712 [=====] - 1s 663us/step - loss: 0.0043
- val_loss: 0.0023
Epoch 27/200
1712/1712 [=====] - 1s 727us/step - loss: 0.0042
- val_loss: 0.0022
Epoch 28/200
1712/1712 [=====] - 1s 727us/step - loss: 0.0042
- val_loss: 0.0022
Epoch 29/200
1712/1712 [=====] - 1s 666us/step - loss: 0.0041
- val_loss: 0.0023
Epoch 30/200
1712/1712 [=====] - 1s 664us/step - loss: 0.0041
- val_loss: 0.0023
Epoch 31/200
1712/1712 [=====] - 1s 665us/step - loss: 0.0043
- val_loss: 0.0023
Epoch 32/200
1712/1712 [=====] - 1s 718us/step - loss: 0.0039
- val_loss: 0.0020
Epoch 33/200
1712/1712 [=====] - 1s 666us/step - loss: 0.0038
- val_loss: 0.0021
Epoch 34/200
1712/1712 [=====] - 1s 726us/step - loss: 0.0036
- val_loss: 0.0019
Epoch 35/200
1712/1712 [=====] - 1s 724us/step - loss: 0.0037
- val_loss: 0.0019
Epoch 36/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0036
- val_loss: 0.0020
Epoch 37/200
1712/1712 [=====] - 1s 721us/step - loss: 0.0035
- val_loss: 0.0018
Epoch 38/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0036
```

```
- val_loss: 0.0020
Epoch 39/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0034
- val_loss: 0.0019
Epoch 40/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0033
- val_loss: 0.0021
Epoch 41/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0033
- val_loss: 0.0018
Epoch 42/200
1712/1712 [=====] - 1s 716us/step - loss: 0.0032
- val_loss: 0.0018
Epoch 43/200
1712/1712 [=====] - 1s 655us/step - loss: 0.0033
- val_loss: 0.0026
Epoch 44/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0035
- val_loss: 0.0019
Epoch 45/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0032
- val_loss: 0.0018
Epoch 46/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0033
- val_loss: 0.0018
Epoch 47/200
1712/1712 [=====] - 1s 717us/step - loss: 0.0032
- val_loss: 0.0017
Epoch 48/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0030
- val_loss: 0.0018
Epoch 49/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0031
- val_loss: 0.0017
Epoch 50/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0030
- val_loss: 0.0018
Epoch 51/200
1712/1712 [=====] - 1s 664us/step - loss: 0.0029
- val_loss: 0.0018
Epoch 52/200
1712/1712 [=====] - 1s 718us/step - loss: 0.0029
- val_loss: 0.0016
Epoch 53/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0027
- val_loss: 0.0017
Epoch 54/200
1712/1712 [=====] - 1s 663us/step - loss: 0.0028
- val_loss: 0.0018
Epoch 55/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0028
- val_loss: 0.0016
Epoch 56/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0027
- val_loss: 0.0016
Epoch 57/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0026
```

```
- val_loss: 0.0016
Epoch 58/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0027
- val_loss: 0.0020
Epoch 59/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0028
- val_loss: 0.0016
Epoch 60/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0026
- val_loss: 0.0018
Epoch 61/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0028
- val_loss: 0.0017
Epoch 62/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0027
- val_loss: 0.0018
Epoch 63/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0026
- val_loss: 0.0016
Epoch 64/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0026
- val_loss: 0.0015
Epoch 65/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0025
- val_loss: 0.0016
Epoch 66/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0024
- val_loss: 0.0018
Epoch 67/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0025
- val_loss: 0.0017
Epoch 68/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0024
- val_loss: 0.0015
Epoch 69/200
1712/1712 [=====] - 1s 718us/step - loss: 0.0024
- val_loss: 0.0015
Epoch 70/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0024
- val_loss: 0.0016
Epoch 71/200
1712/1712 [=====] - 1s 717us/step - loss: 0.0023
- val_loss: 0.0015
Epoch 72/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0023
- val_loss: 0.0015
Epoch 73/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0023
- val_loss: 0.0016
Epoch 74/200
1712/1712 [=====] - 1s 655us/step - loss: 0.0022
- val_loss: 0.0016
Epoch 75/200
1712/1712 [=====] - 1s 721us/step - loss: 0.0023
- val_loss: 0.0014
Epoch 76/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0022
```

```
- val_loss: 0.0014
Epoch 77/200
1712/1712 [=====] - 1s 671us/step - loss: 0.0022
- val_loss: 0.0015
Epoch 78/200
1712/1712 [=====] - 1s 732us/step - loss: 0.0022
- val_loss: 0.0014
Epoch 79/200
1712/1712 [=====] - 1s 667us/step - loss: 0.0021
- val_loss: 0.0014
Epoch 80/200
1712/1712 [=====] - 1s 729us/step - loss: 0.0022
- val_loss: 0.0014
Epoch 81/200
1712/1712 [=====] - 1s 666us/step - loss: 0.0021
- val_loss: 0.0014
Epoch 82/200
1712/1712 [=====] - 1s 664us/step - loss: 0.0021
- val_loss: 0.0014
Epoch 83/200
1712/1712 [=====] - 1s 724us/step - loss: 0.0021
- val_loss: 0.0014
Epoch 84/200
1712/1712 [=====] - 1s 723us/step - loss: 0.0020
- val_loss: 0.0013
Epoch 85/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0020
- val_loss: 0.0014
Epoch 86/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0020
- val_loss: 0.0014
Epoch 87/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0020
- val_loss: 0.0014
Epoch 88/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0019
- val_loss: 0.0014
Epoch 89/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0020
- val_loss: 0.0014
Epoch 90/200
1712/1712 [=====] - 1s 715us/step - loss: 0.0019
- val_loss: 0.0013
Epoch 91/200
1712/1712 [=====] - 1s 655us/step - loss: 0.0020
- val_loss: 0.0014
Epoch 92/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0019
- val_loss: 0.0014
Epoch 93/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0019
- val_loss: 0.0014
Epoch 94/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0019
- val_loss: 0.0014
Epoch 95/200
1712/1712 [=====] - 1s 718us/step - loss: 0.0018
```

```
- val_loss: 0.0013
Epoch 96/200
1712/1712 [=====] - 1s 664us/step - loss: 0.0018
- val_loss: 0.0013
Epoch 97/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0018
- val_loss: 0.0013
Epoch 98/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0018
- val_loss: 0.0014
Epoch 99/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0018
- val_loss: 0.0013
Epoch 100/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0018
- val_loss: 0.0013
Epoch 101/200
1712/1712 [=====] - 1s 724us/step - loss: 0.0018
- val_loss: 0.0013
Epoch 102/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0018
- val_loss: 0.0013
Epoch 103/200
1712/1712 [=====] - 1s 723us/step - loss: 0.0017
- val_loss: 0.0012
Epoch 104/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0018
- val_loss: 0.0012
Epoch 105/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0017
- val_loss: 0.0013
Epoch 106/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0017
- val_loss: 0.0013
Epoch 107/200
1712/1712 [=====] - 1s 723us/step - loss: 0.0017
- val_loss: 0.0012
Epoch 108/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0017
- val_loss: 0.0013
Epoch 109/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0016
- val_loss: 0.0013
Epoch 110/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0016
- val_loss: 0.0012
Epoch 111/200
1712/1712 [=====] - 1s 719us/step - loss: 0.0016
- val_loss: 0.0012
Epoch 112/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0016
- val_loss: 0.0012
Epoch 113/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0016
- val_loss: 0.0013
Epoch 114/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0016
```

```
- val_loss: 0.0012
Epoch 115/200
1712/1712 [=====] - 1s 722us/step - loss: 0.0016
- val_loss: 0.0012
Epoch 116/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0016
- val_loss: 0.0012
Epoch 117/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0016
- val_loss: 0.0012
Epoch 118/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0015
- val_loss: 0.0013
Epoch 119/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0016
- val_loss: 0.0013
Epoch 120/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0016
- val_loss: 0.0012
Epoch 121/200
1712/1712 [=====] - 1s 663us/step - loss: 0.0015
- val_loss: 0.0012
Epoch 122/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0015
- val_loss: 0.0012
Epoch 123/200
1712/1712 [=====] - 1s 722us/step - loss: 0.0015
- val_loss: 0.0012
Epoch 124/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0015
- val_loss: 0.0012
Epoch 125/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0015
- val_loss: 0.0012
Epoch 126/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0015
- val_loss: 0.0012
Epoch 127/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0015
- val_loss: 0.0013
Epoch 128/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0015
- val_loss: 0.0012
Epoch 129/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0014
- val_loss: 0.0012
Epoch 130/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0014
- val_loss: 0.0012
Epoch 131/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0014
- val_loss: 0.0012
Epoch 132/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0014
- val_loss: 0.0012
Epoch 133/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0014
```

```
- val_loss: 0.0013
Epoch 134/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0014
- val_loss: 0.0012
Epoch 135/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0014
- val_loss: 0.0012
Epoch 136/200
1712/1712 [=====] - 1s 719us/step - loss: 0.0014
- val_loss: 0.0011
Epoch 137/200
1712/1712 [=====] - 1s 721us/step - loss: 0.0014
- val_loss: 0.0011
Epoch 138/200
1712/1712 [=====] - 1s 719us/step - loss: 0.0014
- val_loss: 0.0011
Epoch 139/200
1712/1712 [=====] - 1s 720us/step - loss: 0.0014
- val_loss: 0.0011
Epoch 140/200
1712/1712 [=====] - 1s 722us/step - loss: 0.0014
- val_loss: 0.0011
Epoch 141/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0014
- val_loss: 0.0012
Epoch 142/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0013
- val_loss: 0.0012
Epoch 143/200
1712/1712 [=====] - 1s 655us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 144/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 145/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 146/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 147/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0013
- val_loss: 0.0012
Epoch 148/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 149/200
1712/1712 [=====] - 1s 717us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 150/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 151/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 152/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0013
```

```
- val_loss: 0.0011
Epoch 153/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0012
- val_loss: 0.0012
Epoch 154/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 155/200
1712/1712 [=====] - 1s 664us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 156/200
1712/1712 [=====] - 1s 725us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 157/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 158/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 159/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 160/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0013
- val_loss: 0.0011
Epoch 161/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 162/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 163/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 164/200
1712/1712 [=====] - 1s 664us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 165/200
1712/1712 [=====] - 1s 656us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 166/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 167/200
1712/1712 [=====] - 1s 724us/step - loss: 0.0012
- val_loss: 0.0010
Epoch 168/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 169/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0012
- val_loss: 0.0010
Epoch 170/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 171/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0012
```

```
- val_loss: 0.0011
Epoch 172/200
1712/1712 [=====] - 1s 714us/step - loss: 0.0012
- val_loss: 0.0010
Epoch 173/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 174/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 175/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 176/200
1712/1712 [=====] - 1s 667us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 177/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 178/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0011
- val_loss: 0.0011
Epoch 179/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0012
- val_loss: 0.0010
Epoch 180/200
1712/1712 [=====] - 1s 663us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 181/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 182/200
1712/1712 [=====] - 1s 718us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 183/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 184/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0012
- val_loss: 0.0011
Epoch 185/200
1712/1712 [=====] - 1s 655us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 186/200
1712/1712 [=====] - 1s 653us/step - loss: 0.0011
- val_loss: 0.0011
Epoch 187/200
1712/1712 [=====] - 1s 651us/step - loss: 0.0011
- val_loss: 0.0011
Epoch 188/200
1712/1712 [=====] - 1s 654us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 189/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 190/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0011
```

```
- val_loss: 0.0011
Epoch 191/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0012
- val_loss: 0.0010
Epoch 192/200
1712/1712 [=====] - 1s 657us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 193/200
1712/1712 [=====] - 1s 660us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 194/200
1712/1712 [=====] - 1s 661us/step - loss: 0.0011
- val_loss: 0.0011
Epoch 195/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 196/200
1712/1712 [=====] - 1s 659us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 197/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0011
- val_loss: 0.0010
Epoch 198/200
1712/1712 [=====] - 1s 658us/step - loss: 0.0011
- val_loss: 0.0011
Epoch 199/200
1712/1712 [=====] - 1s 662us/step - loss: 0.0011
- val_loss: 0.0011
Epoch 200/200
1712/1712 [=====] - 1s 722us/step - loss: 0.0011
- val_loss: 0.0010
```

Step 7: Visualize the Loss and Test Predictions

(IMPLEMENTATION) Answer a few questions and visualize the loss

Question 1: Outline the steps you took to get to your final neural network architecture and your reasoning at each step.

Answer:

The architecture chosen for this network follows the standard set by the seminal LeNet architecture. The network consists of 4 conv2D+maxpooling2D stacks, a 256 node dense layer, another 50% dropout and 30 node output layer. The RELU activation function is used on every conv2D and dense layer in the network, with the exception of the output layer. This, however, was not the initial design of the network. Using the Adam optimizer with its default learning rate for each iteration, the number conv2D+maxpooling2D stacks, the number of hidden dense layers, the number of nodes in those layers, the number of dropout layers, and the dropout rate were varied and tuned to achieve the lowest possible validation MSE.

This process started with the tuning of the number of conv2D+maxpooling2D stacks. Starting from 2 stacks, the number of layers were incremented till adding more layers didn't add any measurable benefit. In this exercise that number of stacks was discovered to be 4. Employing 5-6 stacks was also experimented with, but was shown to not produce any reduction of validation error, sometimes even going as far as increasing the error. This makes sense as the number of layers increase, the dimensionality of the features per kernel is reduced, not leaving much for the conv2D layers to work with.

After varying the number of stacks, the next task was to tune the hidden dense layers. Initially, the network architecture employed 2 hidden layers, each with 1000 nodes. However, it was shown to produce less than stellar training times. With the mindset of only keeping what is necessary, the number of nodes per layer was reduced. First changing to 500, then 400, 300 and then 256. Finally the number of layers was reduced to 1. All of these cuts proved to show no increase in error, but proved to speed up training.

Finally, in an effort to reduce the validation error even further, the number of dropout layers and rate was varied. Initially, there was a layer with a 25% dropout rate after every stack and in between the hidden and output layer. Through pure brute-force trial and error, the final design of a single layer with a 50% dropout rate in between the hidden and output layer was settled upon. This proved to produce the lowest validation error, with the least amount of extra layers in the architecture.

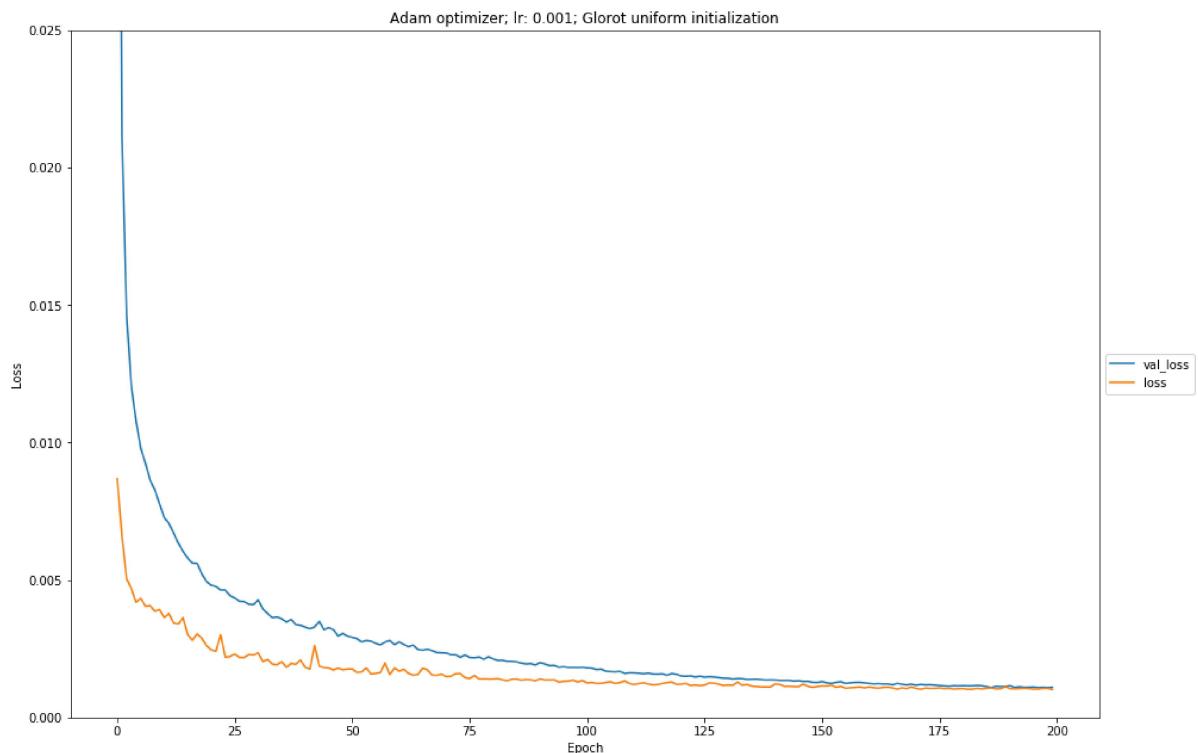
Question 2: Defend your choice of optimizer. Which optimizers did you test, and how did you determine which worked best?

Answer:

The optimizers that I have tested include Adagrad, Adadelta, Adam, Nadam and Adamax. SGD and RMSprop was considered and tried in previous iterations and ruled out. Using the strategy of simple trial and error, or grid search and a training cycle of 30 epochs and 1 iterations using each optimizer, with various learning rates and initializations were tried. Overall, there were 60 combinations. After each trial, the one with the lowest minimum validation error was chosen. The implementation for this can be found in the cell above. The optimizer Adam, with learning rate 0.001 and glorot uniform initialization was shown to produce the lowest validation error. This combination was then used again to train another model over 200 epochs, which resulted in a validation error of 0.0010 +/- 0.0001 on average.

Use the code cell below to plot the training and validation loss of your neural network. You may find [this resource](http://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/) (<http://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/>) useful.

```
In [25]: plt.title('Adam optimizer; lr: 0.001; Glorot uniform initialization')
plt.plot(best_hist.history['loss'])
plt.plot(best_hist.history['val_loss'])
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.ylim(0.0000, 0.025)
plt.subplots_adjust(left=0.0, right=2.0, bottom=0.0, top=2.0)
plt.legend(best_hist.history.keys(), loc='center left', bbox_to_anchor=(1, 0.5))
)
plt.show()
```



Question 3: Do you notice any evidence of overfitting or underfitting in the above plot? If so, what steps have you taken to improve your model? Note that slight overfitting or underfitting will not hurt your chances of a successful submission, as long as you have attempted some solutions towards improving your model (such as regularization, dropout, increased/decreased number of layers, etc).

Answer:

In the plot above, there isn't really any apparent evidence of overfitting or underfitting. However, throughout the previous iterations there definitely was some, and efforts were made to reduce both. In the case of overfitting, dropout with a sizable rate was included. In the case of underfitting, more conv2D+maxpooling2D stacks were added.

Visualize a Subset of the Test Predictions

Execute the code cell below to visualize your model's predicted keypoints on a subset of the testing images.

```
In [26]: import keras
```

```
# Load the best model.  
model = keras.models.load_model('models/best_model.h5')  
  
y_test = model.predict(X_test)  
fig = plt.figure(figsize=(20,20))  
fig.subplots_adjust(left=0, right=1, bottom=0, top=1, hspace=0.05, wspace=0.05)  
for i in range(9):  
    ax = fig.add_subplot(3, 3, i + 1, xticks=[], yticks=[])  
    plot_data(X_test[i], y_test[i], ax)
```



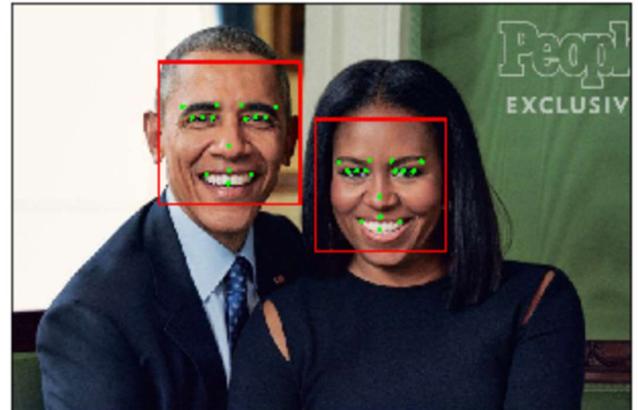
(IMPLEMENTATION) Facial Keypoints Detector

Use the OpenCV face detection functionality you built in previous Sections to expand the functionality of your keypoints detector to color images with arbitrary size. Your function should perform the following steps

1. Accept a color image.
2. Convert the image to grayscale.
3. Detect and crop the face contained in the image.
4. Locate the facial keypoints in the cropped image.
5. Overlay the facial keypoints in the original (color, uncropped) image.

Note: step 4 can be the trickiest because remember your convolutional network is only trained to detect facial keypoints in 96×96 grayscale images where each pixel was normalized to lie in the interval $[0, 1]$, and remember that each facial keypoint was normalized during training to the interval $[-1, 1]$. This means - practically speaking - to paint detected keypoints onto a test face you need to perform this same pre-processing to your candidate face - that is after detecting it you should resize it to 96×96 and normalize its values before feeding it into your facial keypoint detector. To be shown correctly on the original image the output keypoints from your detector then need to be shifted and re-normalized from the interval $[-1, 1]$ to the width and height of your detected face.

When complete you should be able to produce example images like the one below



```
In [27]: def get_points(face_square):
    cnn_in_width, cnn_in_height = 96, 96
    face_square = cv2.resize(face_square, dsize=(cnn_in_width, cnn_in_height))
    face_square = np.expand_dims(face_square, axis=2)
    face_square = face_square / 255
    face_square = face_square.reshape(1, 96, 96, 1)
    # Get the predicted points.
    points = model.predict(face_square)
    return points[0]

def convert_points2origscale(px, py, x, y, w, h):
    # Convert (-1, 1) scale to (0, 97) scale.
    tx = px * 48 + 48
    ty = py * 48 + 48

    # Convert (0, 97) scale to original face square scale.
    tx = (tx * w / 96)
    ty = (ty * h / 96)

    # Shift scale to account for the position of the box.
    # Also turn into an integer.
    return int(tx + x), int(ty + y)

def detect_facial_points(image):
    # Convert the image to grayscale as well.
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

    # Extract the pre-trained face detector from an xml file
    face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_frontalface_default.xml')

    # Detect the faces in image
    faces = face_cascade.detectMultiScale(gray, 1.25, 6)

    image_copy = np.copy(image)

    for (x,y,w,h) in faces:
        # Get the face square, turn gray and resize to fit the model input.
        face_square = gray[y:y+h, x:x+w]

        points = get_points(face_square)

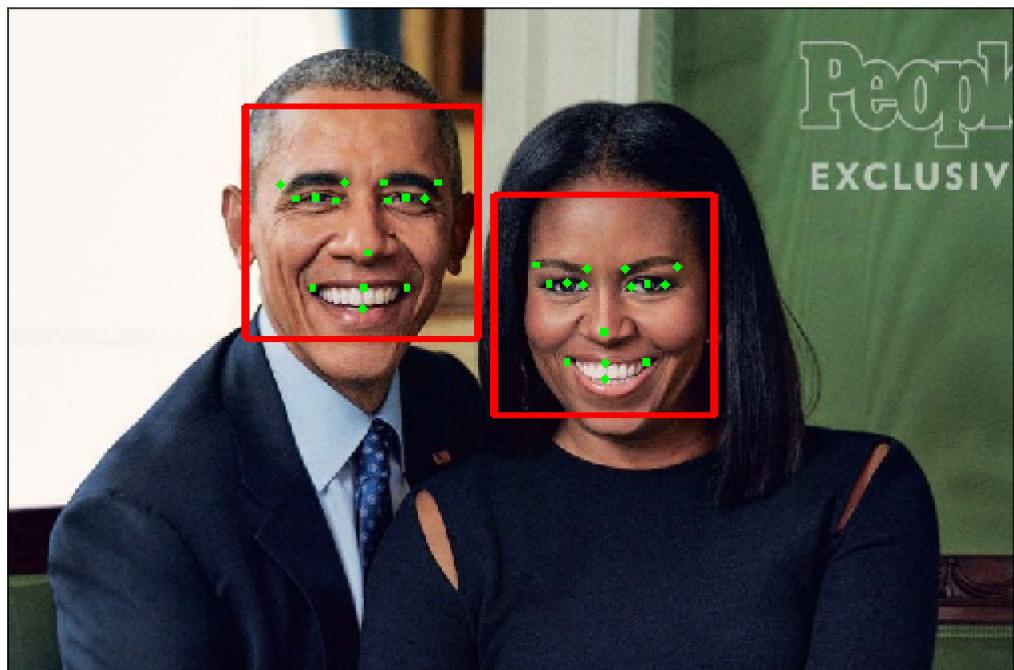
        # Draw the square around the face.
        cv2.rectangle(image_copy, (x,y), (x+w,y+h), (255,0,0), 3)

        # Draw the predicted points.
        # Easy, but expensive way to get every pair of points.
        for px, py in zip(points[:,0], points[:,1]):
            tx, ty = convert_points2origscale(px, py, x, y, w, h)
            cv2.circle(image_copy, (tx, ty), 3, (0, 255, 0), -1)

    return image_copy
```

```
In [28]: ### TODO: Use the face detection code we saw in Section 1 with your trained co  
nv-net  
## TODO : Paint the predicted keypoints on the test image  
  
# Load in color image for face detection  
image = cv2.imread('images/obamas4.jpg')  
  
# Convert the image to RGB colorspace  
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
  
image_with_points = detect面部_points(image)  
  
# plot our image  
fig = plt.figure(figsize = (9,9))  
ax1 = fig.add_subplot(111)  
ax1.set_xticks([])  
ax1.set_yticks([])  
ax1.set_title('Obamas')  
_ = ax1.imshow(image_with_points)  
  
## TODO END.
```

Obamas



(Optional) Further Directions - add a filter using facial keypoints to your laptop camera

Now you can add facial keypoint detection to your laptop camera - as illustrated in the gif below.



The next Python cell contains the basic laptop video camera function used in the previous optional video exercises. Combine it with the functionality you developed for keypoint detection and marking in the previous exercise and you should be good to go!

```
In [ ]: import cv2
import time
from keras.models import load_model
def laptop_camera_go():
    # Create instance of video capturer
    cv2.namedWindow("face detection activated")
    vc = cv2.VideoCapture(0)

    # Try to get the first frame
    if vc.isOpened():
        rval, frame = vc.read()
    else:
        rval = False

    # keep video stream open
    while rval:
        frame = detect面部点(frame)
        # plot image from camera with detections marked
        cv2.imshow("face detection activated", frame)

        # exit functionality - press any key to exit Laptop video
        key = cv2.waitKey(20)
        if key > 0: # exit by pressing any key
            # destroy windows
            cv2.destroyAllWindows()

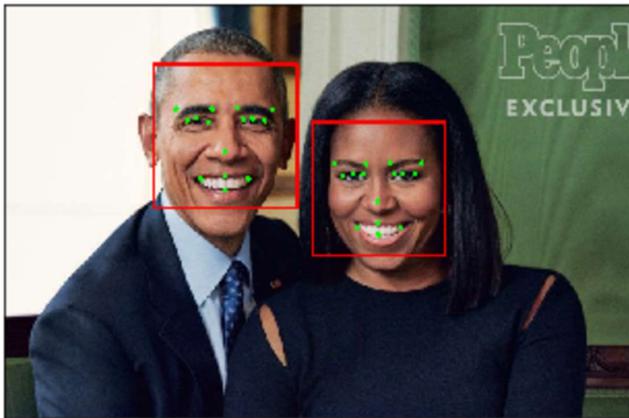
            # hack from stack overflow for making sure window closes on osx --
            # https://stackoverflow.com/questions/6116564/destroywindow-does-not-close-window-on-mac-using-python-and-opencv
            for i in range (1,5):
                cv2.waitKey(1)
            return

        # read next frame
        time.sleep(0.05)           # control framerate for computation - default 20 frames per sec
        rval, frame = vc.read()
```

```
In [ ]: # Run your keypoint face painter
#laptop_camera_go()
```

(Optional) Further Directions - add a filter using facial keypoints

Using your freshly minted facial keypoint detector pipeline you can now do things like add fun filters to a person's face automatically. In this optional exercise you can play around with adding sunglasses automatically to each individual's face in an image as shown in a demonstration image below.



To produce this effect an image of a pair of sunglasses shown in the Python cell below.

```
In [ ]: # Load in sunglasses image - note the usage of the special option1  
# cv2.IMREAD_UNCHANGED, this option is used because the sunglasses  
# image has a 4th channel that allows us to control how transparent each pixel  
# in the image is  
sunglasses = cv2.imread("images/sunglasses_4.png", cv2.IMREAD_UNCHANGED)  
  
# Plot the image  
fig = plt.figure(figsize = (6,6))  
ax1 = fig.add_subplot(111)  
ax1.set_xticks([])  
ax1.set_yticks([])  
ax1.imshow(sunglasses)  
ax1.axis('off');
```

This image is placed over each individual's face using the detected eye points to determine the location of the sunglasses, and eyebrow points to determine the size that the sunglasses should be for each person (one could also use the nose point to determine this).

Notice that this image actually has *4 channels*, not just 3.

```
In [ ]: # Print out the shape of the sunglasses image  
print ('The sunglasses image has shape: ' + str(np.shape(sunglasses)))
```

It has the usual red, blue, and green channels any color image has, with the 4th channel representing the transparency level of each pixel in the image. Here's how the transparency channel works: the lower the value, the more transparent the pixel will become. The lower bound (completely transparent) is zero here, so any pixels set to 0 will not be seen.

This is how we can place this image of sunglasses on someone's face and still see the area around of their face where the sunglasses lie - because these pixels in the sunglasses image have been made completely transparent.

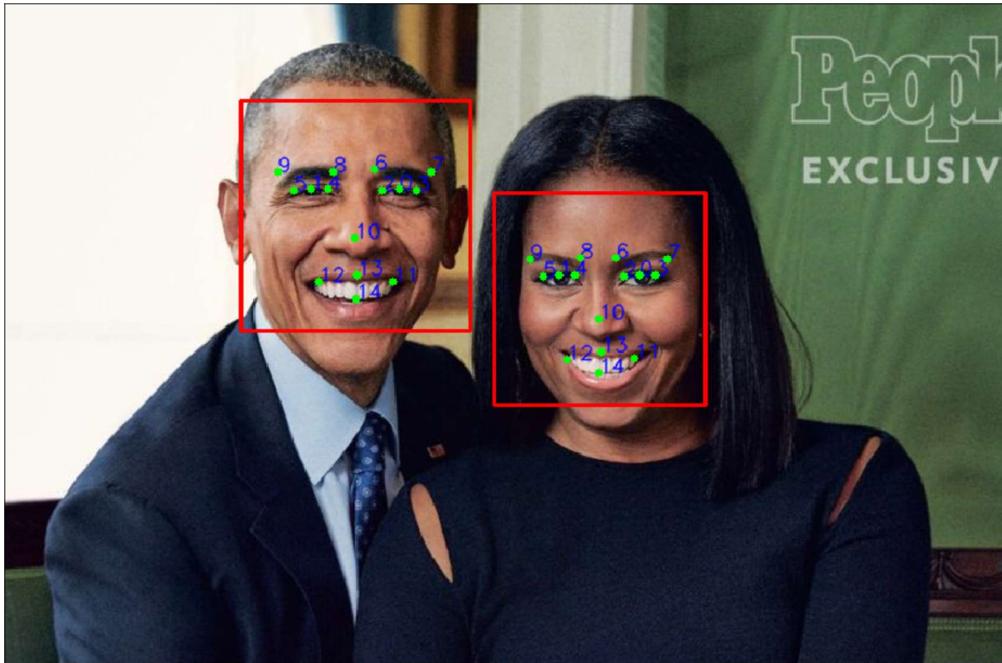
Lets check out the alpha channel of our sunglasses image in the next Python cell. Note because many of the pixels near the boundary are transparent we'll need to explicitly print out non-zero values if we want to see them.

```
In [ ]: # Print out the sunglasses transparency (alpha) channel
alpha_channel = sunglasses[:, :, 3]
print ('the alpha channel here looks like')
print (alpha_channel)

# Just to double check that there are indeed non-zero values
# Let's find and print out every value greater than zero
values = np.where(alpha_channel != 0)
print ('\n the non-zero values of the alpha channel look like')
print (values)
```

This means that when we place this sunglasses image on top of another image, we can use the transparency channel as a filter to tell us which pixels to overlay on a new image (only the non-transparent ones with values greater than zero).

One last thing: it's helpful to understand which keypoint belongs to the eyes, mouth, etc. So, in the image below, we also display the index of each facial keypoint directly on the image so that you can tell which keypoints are for the eyes, eyebrows, etc.



With this information, you're well on your way to completing this filtering task! See if you can place the sunglasses automatically on the individuals in the image loaded in / shown in the next Python cell.

```
In [ ]: # Load in color image for face detection
image = cv2.imread('images/obamas4.jpg')
# Load in sunglasses image
sunglasses = cv2.imread("images/sunglasses_4.png", cv2.IMREAD_UNCHANGED)

# Convert the image to RGB colorspace
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Convert the image to grayscale as well.
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Extract the pre-trained face detector from an xml file
face_cascade = cv2.CascadeClassifier('detector_architectures/haarcascade_frontalface_default.xml')

# Detect the faces in image
faces = face_cascade.detectMultiScale(gray, 1.25, 6)

image_w_glasses = np.copy(image)

for (x,y,w,h) in faces:
    # Get the face square, turn gray and resize to fit the model input.
```

```

face_square = gray[y:y+h, x:x+w]
points = get_points(face_square)

# Easy, but expensive way to get every pair of eye points ONLY.
converted_points = []
for px, py in zip(points[:20:2], points[1:20:2]):
    tx, ty = convert_points2origscale(px, py, x, y, w, h)
    converted_points.append((tx, ty))

ibox_x, ibox_y, ibox_w, ibox_h = cv2.boundingRect(np.array(converted_points))

## TODO: Fix this overlay. Currently doesn't look very good.
# Calculate the appropriate height of the sunglasses based on width of ibox.
ratio = sunglasses.shape[0] / ibox_w
target_h = int(sunglasses.shape[1] / ratio)

# Resize glasses.
resized_glasses = cv2.resize(sunglasses, dsize=(ibox_w, target_h))

# Overlay glasses.
image_w_glasses[ibox_y:ibox_y+resized_glasses.shape[0], ibox_x:ibox_x+resized_glasses.shape[1]] = resized_glasses[:, :, :3]

# Plot the image
fig = plt.figure(figsize = (15,15))
ax1 = fig.add_subplot(121)
ax1.set_xticks([])
ax1.set_yticks([])
ax1.set_title('Original Image')
ax1.imshow(image)
ax2 = fig.add_subplot(122)
ax2.set_xticks([])
ax2.set_yticks([])
ax2.set_title('Now with Sunglasses')
_ = ax2.imshow(image_w_glasses)

```

In []: *## (Optional) TODO: Use the face detection code we saw in Section 1 with your trained conv-net to put*
sunglasses on the individuals in our test image

(Optional) Further Directions - add a filter using facial keypoints to your laptop camera

Now you can add the sunglasses filter to your laptop camera - as illustrated in the gif below.



The next Python cell contains the basic laptop video camera function used in the previous optional video exercises. Combine it with the functionality you developed for adding sunglasses to someone's face in the previous optional exercise and you should be good to go!

```
In [ ]: import cv2
import time
from keras.models import load_model
import numpy as np

def laptop_camera_go():
    # Create instance of video capturer
    cv2.namedWindow("face detection activated")
    vc = cv2.VideoCapture(0)

    # try to get the first frame
    if vc.isOpened():
        rval, frame = vc.read()
    else:
        rval = False

    # Keep video stream open
    while rval:
        # Plot image from camera with detections marked
        cv2.imshow("face detection activated", frame)

        # Exit functionality - press any key to exit Laptop video
        key = cv2.waitKey(20)
        if key > 0: # exit by pressing any key
            # Destroy windows
            cv2.destroyAllWindows()

        for i in range (1,5):
            cv2.waitKey(1)
        return

    # Read next frame
    time.sleep(0.05)          # control framerate for computation - default 20 frames per sec
    rval, frame = vc.read()
```

```
In [ ]: # Load facial landmark detector model
model = load_model('my_model.h5')

# Run sunglasses painter
laptop_camera_go()
```