



AWS Builders' Day

*Getting Started with Sagemaker
MXNet and Jupyter*

December 2018

Table of Contents

Overview	3
Launching a Jupyter Notebook Server using Amazon SageMaker	4
Launching the Jupyter Notebook Instance	5
Working Through the Jupyter Notebook Lab	8
Terminating the Notebook Instance	10

Overview

Apache MXNet is a lean, flexible, and ultra-scalable deep learning framework that supports state of the art in deep learning models, including convolutional neural networks (CNNs) and long short-term memory networks (LSTMs). The framework has its roots in academia and came about through the collaboration and contributions of researchers at several top universities. It has been designed to excel at computer vision, speech, language processing and understanding, generative models, concurrent neural networks, and recurrent neural networks.

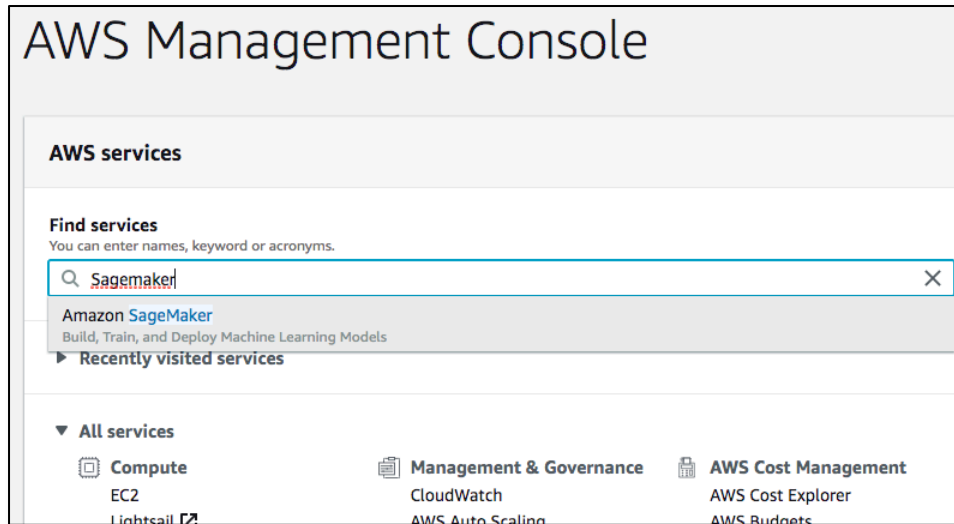
Amazon SageMaker is a fully-managed service that enables developers and data scientists to quickly and easily build, train, and deploy machine learning models at any scale. Amazon SageMaker removes all the barriers that typically slow down developers who want to use machine learning.

This lab is designed to demonstrate how to run the Amazon Deep Learning AMI using Amazon SageMaker, and will introduce the audience to:

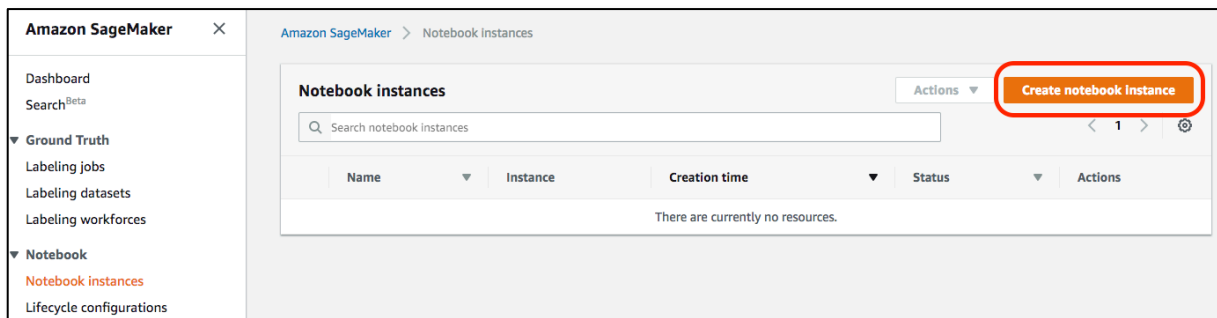
- Launching Jupyter Notebook instance using Amazon SageMaker
- Stepping through and modifying Jupyter Notebooks
- Hands-on experience with:
 - Multi-Layer Perceptron
 - Convolutional Neural Networks
 - Long Short Term Memory artificial neural networks.

Launching a Jupyter Notebook Server using Amazon SageMaker

1. Sign into the AWS Management Console <https://console.aws.amazon.com/>.
2. Click on **Amazon SageMaker** from the list of all services. This will bring you to the Amazon SageMaker console homepage.



3. To create a new Jupyter notebook instance, go to **Notebook instances**, and click the **Create notebook instance** button at the top of the browser window.



4. Type *[First Name]-[Last Name]-Lab-Server* into the **Notebook instance name** text box, *ml.p2.xlarge* into the **Notebook instance type**.
5. Under the **IAM Role** dropdown, choose **Create a new role**, and select *None* under the **S3 Buckets you specify – optional**. Click on **Create role** to close the dialog.

Getting Started with SageMaker – MXNet and Jupyter

Create an IAM role

Passing an IAM role gives Amazon SageMaker permission to perform actions in other AWS services on your behalf. Creating a role here will grant permissions described by the [AmazonSageMakerFullAccess](#) IAM policy to the role you create.

The IAM role you create will provide access to:

- ☒ S3 buckets you specify - optional
 - ☐ Specific S3 buckets

Example: bucket-name-1, bucket-name-2, bucket-name-3

 Comma delimited. ARNs, "*" and "/" are not supported.
 - ☐ Any S3 bucket

Allow users that have access to your notebook instance access to any bucket and its contents in your account.
 - ☒ None
- ☒ Any S3 bucket with "sagemaker" in the name
- ☒ Any S3 object with "sagemaker" in the name
- ☒ Any S3 object with the tag "sagemaker" and value "true" [See Object tagging](#)
- ☒ S3 bucket with a Bucket Policy allowing access to SageMaker [See S3 bucket policies](#)

[Cancel](#) [Create role](#)

- This will display a success message for the creation of the new role. Leaving everything else at their default values – *Elastic inference*, *VPC*, *Lifecycle configuration*, *Encryption* and *Volume Size* – and click on **Create notebook instance** at the bottom of the screen. This will launch a p2.xlarge instance running the Amazon Deep Learning AMI

Launching the Jupyter Notebook Instance

- Make sure that the server status is **InService**. This will take a few minutes once the creation process has started.

Success! Your notebook instance is being created.
Open the notebook instance when status is InService and open a template notebook to get started. [View details](#)

Amazon SageMaker > Notebook instances

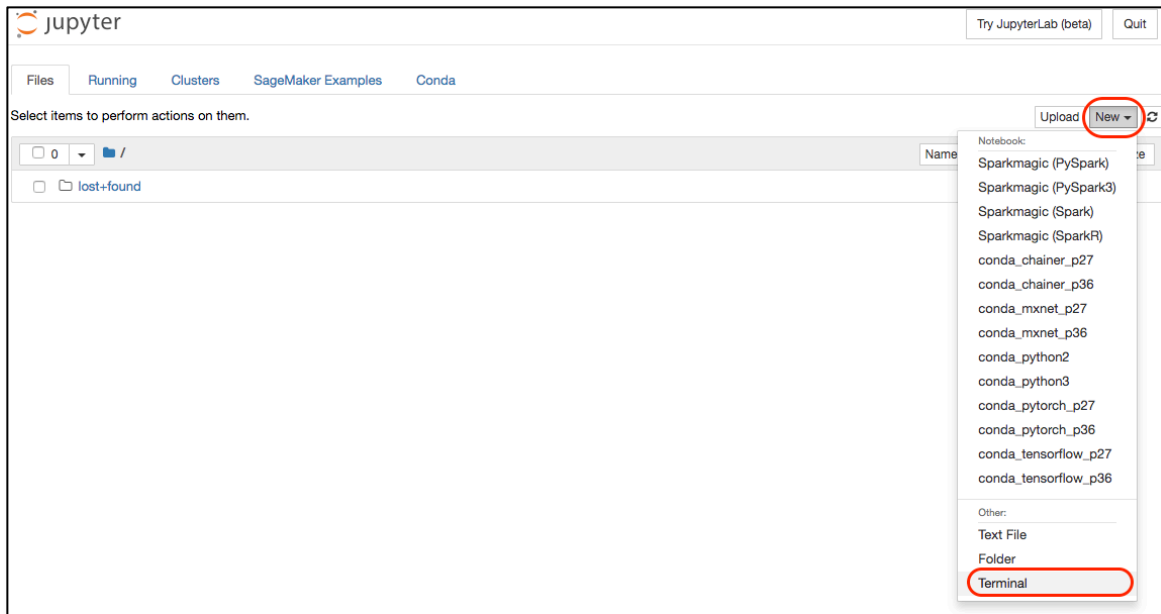
Notebook instances [Actions](#) [Create notebook instance](#)

Search notebook instances

Name	Instance	Creation time	Status	Actions
<input type="radio"/> Andrew-Kane-Lab-Server	m1.p3.2xlarge	Dec 09, 2018 22:06 UTC	<input checked="" type="checkbox"/> InService	Open Jupyter Open JupyterLab

- Click **Open Jupyter**. You now have access to the Jupyter Notebook.
- We are going to log on to the server by launching a terminal console. Click **New** and select

Terminal. This will open up the terminal on a new browser tab.



4. To download all the lab files and supporting file execute the following commands

```
cd sample-notebooks/
```

```
wget https://s3.amazonaws.com/sa-imd/sa_ml_lab.tar.gz
```

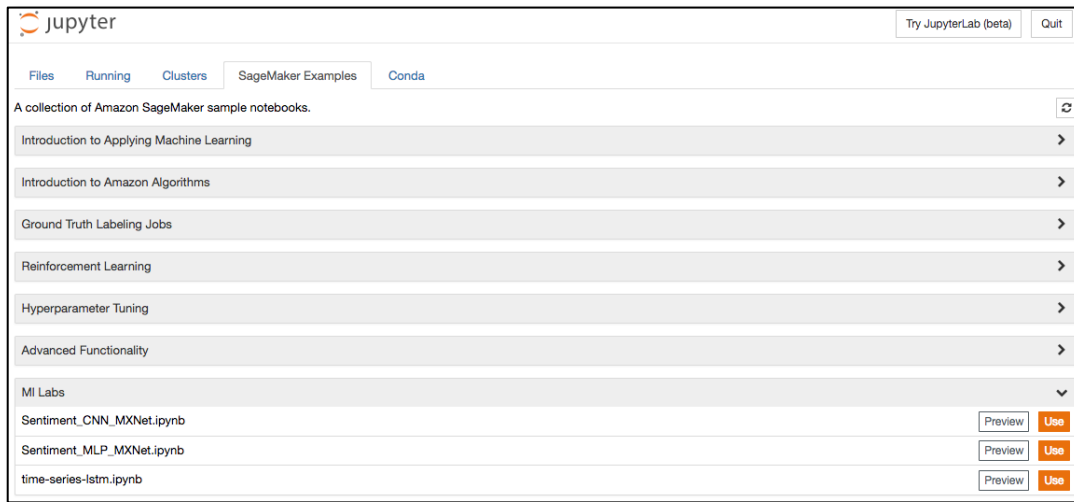
5. Once download is complete, extract the files by typing

```
tar xf sa_ml_lab.tar.gz
```

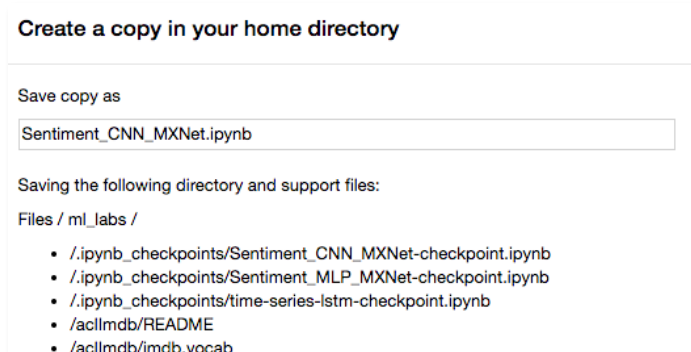
6. Close this window, go to the previous browser tab, and click on the **SageMaker Examples** tab. Scroll down to the **ML Labs** section. You should now see the following 3 notebooks available for use:

- Sentiment_MLP_MXNet.ipynb
- Sentiment_CNN_MXNet.ipynb
- time-series-lstm.ipynb

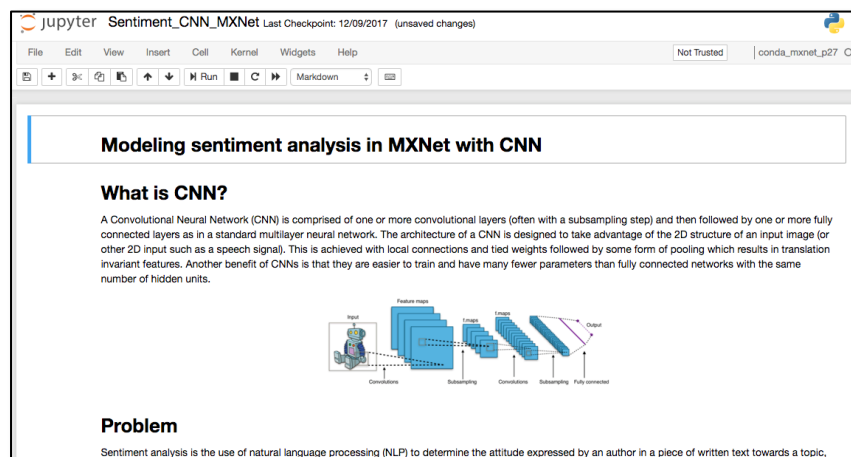
Getting Started with Sagemaker – MXNet and Jupyter



7. We will start by looking at the notebook for Sentiment Analysis using Convolutional Neural Networks in the MXNet framework *Sentiment_CNN_MXNet.ipynb*. Click on the **Preview** button to preview the entire lab in a new window, but click the **Use** button to activate it.



8. A dialog will appear that will show a long list of files to be copied – scroll down and click the **Create copy** button. This will automatically open this notebook.

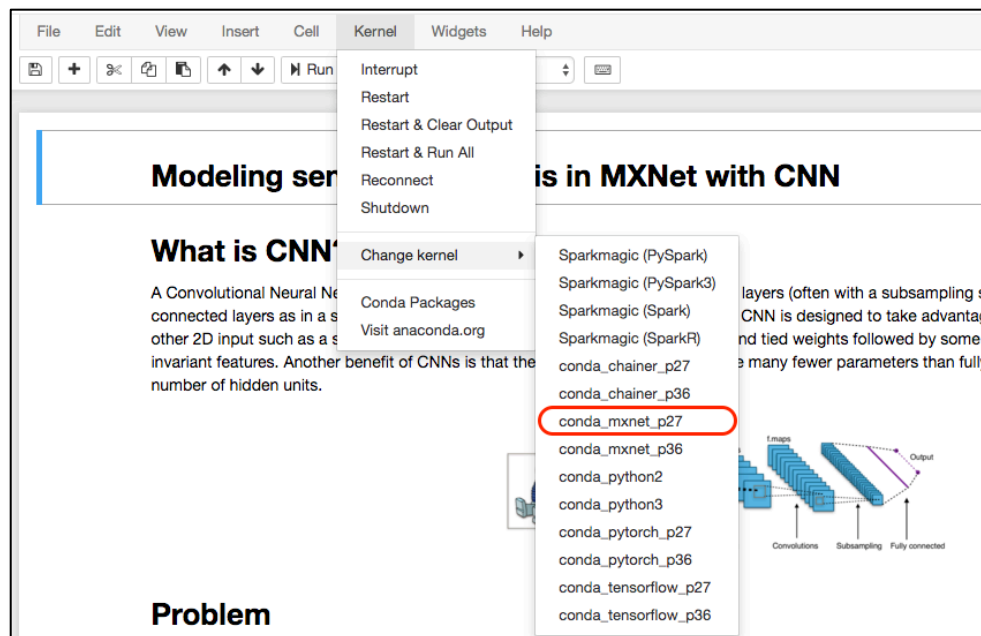


Getting Started with Sagemaker – MXNet and Jupyter

9. Switch back to the previous window and select the **Files** tab, then the **ml_labs_{date}** folder that the previous step created. This will show that all 3 notebooks have been downloaded, along with all associated data, but that only the Convolutional Neural Network notebook is open. If you select the other two then they will be opened in their own window.



10. Go to the Convolutional Neural Network tab and change the kernel to “*Environment (conda_mxnet_p27)*”.

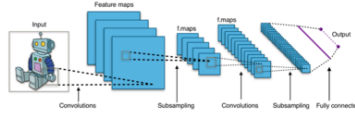


Working Through the Jupyter Notebook Lab

1. You can now work through the notebook lab. A notebook consisted of a number of cells; in SageMaker these will typically either be *Code* or *Markdown* cell. Markdown is used to allow for documentation to be defined inline with the code, giving the author a rich set of markdown formatting options. The first few cells in this notebook are Markdown, and if you select one then the whole cell is highlighted.

Modeling sentiment analysis in MXNet with CNN**What is CNN?**

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units.



- After scrolling past the introduction to the lab you will find the first Code cell, which is called **Load Modules**. The previous Markdown cells ought to have described what this next Code cell is going to do – for the sake of this lab you do not have to understand the code that is being run in the Code cell, rather you should just appreciate what the notebook is doing and how you interact with a Jupyter notebook.

Load Modules

```
In [ ]: # Load all the libraries and modules
import numpy as np
import sys
import os
import re

from text import Tokenizer
import mxnet as mx
from matplotlib import pyplot
from six.moves.urllib.request import urlopen
from sequence import pad_sequences

from IPython.display import display
from ipywidgets import widgets

# Enable logging so we will see output during the training
import logging
logger = logging.getLogger()
logger.setLevel(logging.DEBUG)
```

- To the left of the Code module is a *Play* icon, along with a set of empty braces ('[]'). By clicking the *Play* icon, or selecting the *Run* command in the menu bar, the Jupyter notebook will execute this code, outputting and code outputs to the notebook screen and keeping any results data internally for re-use in future steps.
- Click on the *Play* icon to execute this Code cell. Whilst the code is executing the braces will change to be **[*]**, indicating that it is executing, and once complete will change to **[1]**. Future cells will have increasing numbers inside the braces, so the braces after the execution of the second Code cell, *Process Movie Review Data*, shows as **[2]**.

Process Movie Review Data

The raw movie review dataset is in the aclImdb folder. The files are processed into training and testing datasets for training and validation purposes.

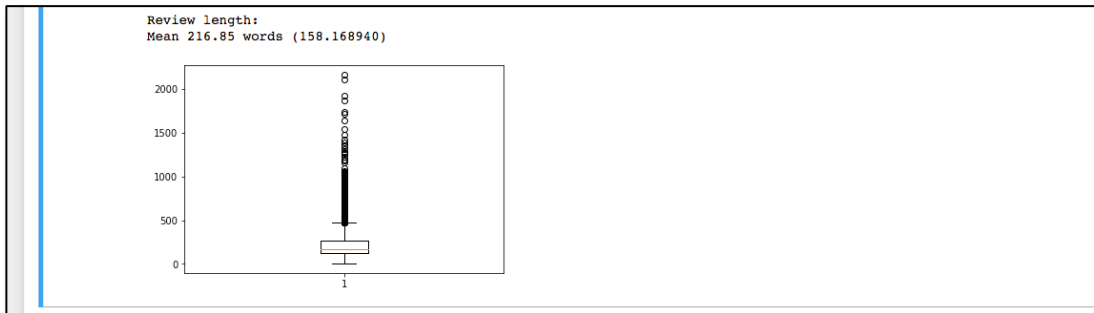
```
In [2]: #from keras.preprocessing.text import Tokenizer

# We specify number of words to index and this is also the size of vocabulary
vocabsize = 10000

# This is the directory where the raw review data is located
path = "aclImdb/"
```

Getting Started with Sagemaker – MXNet and Jupyter

- Note, as subsequent cells use the processing results of the previous cells, you should not start to execute a cell until its predecessor has completed. Also, please be aware that re-running previous code cells will almost certainly invalidate all future cells, so you will have to re-run those as well.
- You will see at code cell 3, *Data Summarization*, that the code cell can take advantage of any other in-language features – in this case it will use the Python library **pyplot** to output a simple plot showing the distribution of review word lengths.



- In cell 6, *Build a CNN Network*, you are instructed to edit the code – this shows one of the powerful features of Jupyter, as Data Scientists are able to edit, update and otherwise evolve their algorithms and parameters on-the-fly within the notebook.
- Continue through to cell 7, *Model Training* – with an **ml.p3.2xlarge** instance type this will take around 2 minutes, longer if you use a **c5** or **m5** instance type. The training algorithm will run through 10 iterations – or *epochs* – of the training algorithm. After a few epochs of progress have been output you will be able to estimate the time to completion. Eventually, you will have values for training and validation accuracy.

```
INFO:root:Epoch[9] Batch [100] Speed: 3246.89 samples/sec accuracy=0.956436
INFO:root:Epoch[9] Batch [200] Speed: 3249.01 samples/sec accuracy=0.951200
INFO:root:Epoch[9] Batch [300] Speed: 3248.42 samples/sec accuracy=0.958800
INFO:root:Epoch[9] Batch [400] Speed: 3249.92 samples/sec accuracy=0.958000
INFO:root:Epoch[9] Train-accuracy=0.962626
INFO:root:Epoch[9] Time cost=7.690
INFO:root:Epoch[9] Validation-accuracy=0.837760
```

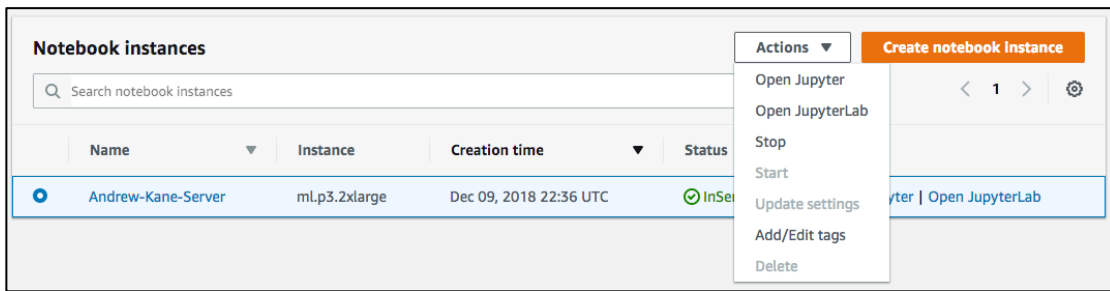
- We save the trained model, and complete up to Coding cell 11 – we are now ready for some interactive testing. The *Input Screen for Model Testing* Code cell can be run multiple times – each time it will show a dialog, asking you enter a movie review paragraph and to hit the **Predict Sentiment** button – it will then give you its probability score estimate for this being a review with positive or negative sentiment. Type in your own reviews, or use one of the four example reviews in the previous Markdown cell.

Terminating the Notebook Instance

- Open the Amazon SageMaker console <https://console.aws.amazon.com/sagemaker/> and click on **Notebook instances**

Getting Started with Sagemaker – MXNet and Jupyter

- Find the notebook instance listed as *[First Name]-[Last Name]-Lab-Server*, select its radio button and they click the **Actions** dropdown.



- Click **Stop** to stop the Notebook Instance. This does not delete the underlying data and resources, and once the instance status has changed to *Stopped* you can click on the **Actions** dropdown again, but this time select **Delete**.

Note that by selecting the name of the Notebook instance on this dialog you are taken to a more detailed information page regarding that instance, which also has **Stop** and **Delete** buttons present – notebooks can equally be deleted using this method.