

Hey ML enthusiasts, welcome to the Call For Makers: Machine Learning section by Nevras, Innovation Garage. We're going to follow Python for coding, so it's necessary for you to complete Python video lectures first. We suggest you to go through below section for a brief introduction in each and every area of Machine Learning so that you can figure out your area of interest. After that, you're advised to go through the entire course in detail to grasp the concepts in that area and start working on your project. For any kind of doubts, you can ask us in the groups or message any of the admin, we will be there to help you out.

- Those who don't know Python follow this series upto video number 25.  
[https://www.youtube.com/watch?v=oVp1vrfL\\_w4&list=PLQVvaa0QuDe8XSftW-RAxdo6OmaeL85M](https://www.youtube.com/watch?v=oVp1vrfL_w4&list=PLQVvaa0QuDe8XSftW-RAxdo6OmaeL85M)

These are the basic topics for getting you started :

#>Understanding the two pillars of Machine Learning

- <https://medium.com/@gowthamy/machine-learning-supervised-learning-vs-unsupervised-learning-f1658e12a780>
- LINEAR REGRESSION
  - <https://medium.com/coinmonks/linear-regression-bf5141ce9ac8>
  - [http://www.holehouse.org/mlclass/04\\_Linear\\_Regression\\_with\\_multiple\\_variables.html](http://www.holehouse.org/mlclass/04_Linear_Regression_with_multiple_variables.html)
- LOGISTIC REGRESSION
  - <https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24>

→ <https://www.edureka.co/blog/logistic-regression-in-python/>

● GRADIENT DESCENT:

- <https://towardsdatascience.com/gradient-descent-in-a-nutshell-eaf8c18212f0>
- <https://medium.freecodecamp.org/understanding-gradient-descent-the-most-popular-ml-algorithm-a66c0d97307f>
- <https://medium.com/@aerinykim/the-gradients-of-linear-regression-cost-function-1a42b98ab0ef>

● MODEL REPRESENTATION AND COST FUNCTION:

- [https://medium.com/@lachlanmiller\\_52885/machine-learning-week-1-cost-function-gradient-descent-and-univariate-linear-regression-8f5fe69815fd](https://medium.com/@lachlanmiller_52885/machine-learning-week-1-cost-function-gradient-descent-and-univariate-linear-regression-8f5fe69815fd)
- [https://medium.com/@lachlanmiller\\_52885/understanding-and-calculating-the-cost-function-for-linear-regression-39b8a3519fcb](https://medium.com/@lachlanmiller_52885/understanding-and-calculating-the-cost-function-for-linear-regression-39b8a3519fcb)

● Complete the following topics of this lecture for the first lecture series:\*

<https://www.coursera.org/learn/machine-learning>

- Week 1 : Introduction
- Week 1 : Linear Regression with One Variable
- Week 2 : Linear Regression with Multiple Variables
- Week 3 : Logistic Regression

## Implementation

- Follow the link to know more about **numpy**:
  - <http://cs231n.github.io/python-numpy-tutorial/#numpy>
- For **matplotlib** you can go through follow these links:
  - <https://matplotlib.org/3.1.0/tutorials/introductory/pyplot.html>
  - <https://pythonprogramming.net/matplotlib-python-3-basics-tutorial/>
  - <https://towardsdatascience.com/matplotlib-tutorial-learn-basics-of-pythons-powerful-plotting-library-b5d1b8f67596>
- Let's have some introduction in **tensorflow**:
  - <https://medium.com/analytics-vidhya/series-learning-tensorflow-the-easy-way-dcc5be834d74>
  - <https://medium.com/analytics-vidhya/getting-started-with-tensorflow-the-easy-way-part-2-30e83830bd25>

- <https://medium.com/analytics-vidhya/getting-started-with-tensorflow-the-easy-way-part-3-9714a09af723>
  - <https://www.tensorflow.org/guide/variables>
  - <https://www.tensorflow.org/guide/tensors>
  - Follow the link to know more about **pandas**:  
<https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/>
  - Now you can go through the explanation of **building of Linear regression models in tensorflow**:
    - <https://towardsdatascience.com/a-line-by-line-laymans-guide-to-linear-regression-using-tensorflow-3c0392aa9e1f>
    - <https://www.geeksforgeeks.org/linear-regression-using-tensorflow/>
- 
- Complete the following topics of this lecture for the first lecture series:\*  
<https://www.coursera.org/learn/machine-learning>
    - Week 4 : Neural Networks : Representation
    - Week 5 : Neural Networks : Learning

!

- [http://cs229.stanford.edu/notes-spring2019/cs229-notes-deep\\_learning.pdf](http://cs229.stanford.edu/notes-spring2019/cs229-notes-deep_learning.pdf)
- <http://cs229.stanford.edu/notes-spring2019/backprop.pdf>
- <https://www.youtube.com/watch?v=aircAruvnKk>
- <https://www.youtube.com/watch?v=IHZwWFHWa-w>
- <https://www.youtube.com/watch?v=Ilq3gGewQ5U>
- <https://www.youtube.com/watch?v=tIeHLnjs5U8>

||

- Your task is : *build a neural network from scratch*. The condition is: you can't import any library except **numpy**. We've listed out the necessary steps required to tackle this problem:
  - Define X and Y as arrays with proper shapes (Take the shape of X as (3,4) and that of Y as (3,1), assign arbitrary values)
  - Define two functions Sigmoid and Derivative\_sigmoid.
  - State the number of epochs, learning rate, number of hidden layers and finally the number of neurons in the input and output layer.
  - Initialize Weights and biases

- Now run a loop for a particular number of epochs and write the code for forward propagation and backpropagation.
  - Show the output / `y_predicted` .
- This week we will be dealing with the CNNs, do try out reading the research paper attached below.
- <https://youtu.be/gYpoJMIgyXA>
  - [https://youtu.be/hd\\_KFJ5ktUc](https://youtu.be/hd_KFJ5ktUc)
  - <https://youtu.be/LxfUGhug-iQ>
  - (Optionally, watch the entire video lecture series as its very informative and intuitive)

!

- <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- <https://colah.github.io/posts/2014-07-Conv-Nets-Modular/>

# \$

- [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2014/papers/Taigman\\_DeepFace\\_Closing\\_the\\_2014\\_CVPR\\_paper.pdf](http://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Taigman_DeepFace_Closing_the_2014_CVPR_paper.pdf)
- <https://youtu.be/GxZrEKZfW2o>
- <https://youtu.be/ta5fdaqDT3M>

% !

- Course on Machine Learning by Andrew Ng, Stanford:  
<https://www.coursera.org/learn/machine-learning>
- Course on Computer Vision (CS 231n) by Stanford: Cover all the lectures starting from  
<https://www.youtube.com/watch?v=NfnWJUyUJYU&feature=youtu.be>
- Course on Natural Language Processing:  
<https://www.coursera.org/learn/nlp-sequence-models>

" \$ ! & # \$

- STYLE TRANSFER using deep Learning:  
[https://github.com/udacity/deep-learning-v2-pytorch/blob/master/style-transfer/Style\\_Transfer\\_Exercise.ipynb](https://github.com/udacity/deep-learning-v2-pytorch/blob/master/style-transfer/Style_Transfer_Exercise.ipynb)
- STOCK PRICE PREDICTION using deep learning techniques:  
<https://blog.edugrad.com/stock-price-prediction-using-machine-learning-deep-learning-techniques-with-python-code/>
- SENTIMENT ANALYSIS on text using Convolutional Neural Networks:  
<https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/>
- Scikit-Learn for wine quality testing  
<https://elitedatascience.com/python-machine-learning-tutorial-scikit-learn#step-3>

& # \$

- YOLO: Real Time Object Detection:  
<https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection>
- Face Recognition: <https://github.com/krasserm/face-recognition>
- Voice Conversion: <https://github.com/andabi/deep-voice-conversion>