INSTALL ANACONDA

<https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart>

CHANGE FOLDER RIGHTS

sudo chmod -R 777 /var/DirectoryName

Dependencies

pip install -U scikit-learn

python -mpip install matplotlib

**pip install pandas**

CREATE ENVIRONMENT

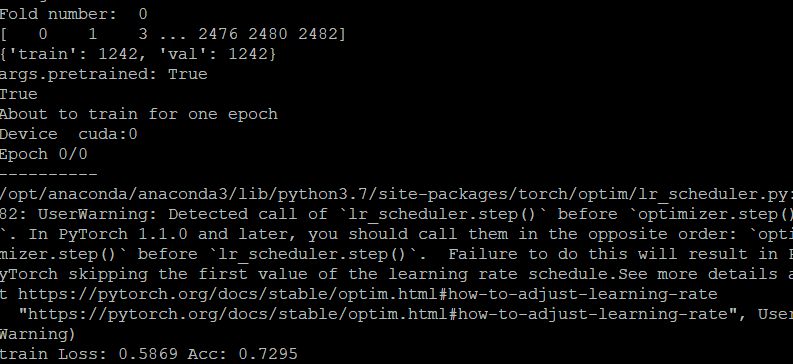
Conda create –name saul

ACTIVATE THE ENVIRONMENT

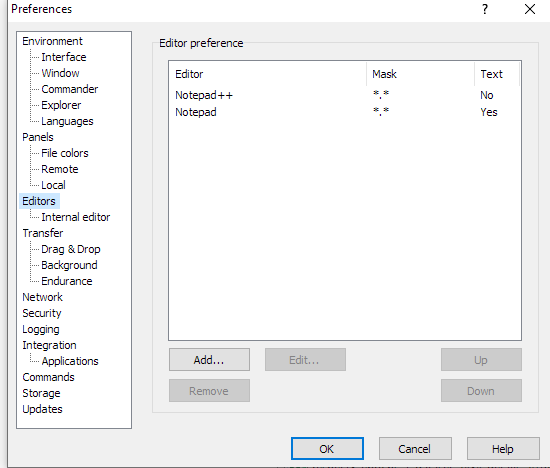
Conda activate saul

PROBLEM WITH TENSORFLOW 2.0, it does not have contrib

# K-FOLD is the same



# USE NOTEPAD++ in WinSCP



# WATCH GPU USAGE

watch -n 0.5 nvidia-smi

top (for cpu usage)

# HYPER PARAMETER FINETUNING

<https://towardsdatascience.com/a-bunch-of-tips-and-tricks-for-training-deep-neural-networks-3ca24c31ddc8>

# IN PLACE DROPOUT

import torch

import torch.nn as nn

inp = torch.tensor([1.0, 2.0, 3, 4, 5])

outplace\_dropout = nn.Dropout(p=0.4)

print(inp)

output = outplace\_dropout(inp)

print(output)

print(inp) # Notice that the input doesn't get changed here

inplace\_droput = nn.Dropout(p=0.4, inplace=True)

inplace\_droput(inp)

print(inp) # Notice that the input is changed now

FOR INBREAST NETWORK

If your dataset in your problem domain is similar to [ImageNet dataset](http://www.image-net.org/challenges/LSVRC/2012/" \t "_blank), use a pre-trained model on this dataset. The most widely used pre-trained models are [VGG](https://arxiv.org/abs/1409.1556) net, [ResNet](https://arxiv.org/abs/1512.03385" \t "_blank), [DenseNet](https://arxiv.org/abs/1608.06993" \t "_blank) or [Xception](https://arxiv.org/abs/1610.02357" \t "_blank) etc. There are many layer architectures, for instance, VGG (19 and 16 layers), ResNet (152, 101, 50 layers or less), DenseNet (201, 169 and 121 layers). **Note**: Do not try searching hyper-parameters by using more layers nets (e.g. VGG-19, ResNet-152 or DenseNet-201 layers net because it is computationally expensive), use less layers nets instead (e.g. VGG-16, ResNet-50 or DenseNet-121 layers). Pick one pre-trained model that you think it gives the best performance with your hyper-parameters (say ResNet-50 layers). After you obtained the optimal hyper parameters, just select the same but more layers net (say ResNet-101 or ResNet-152

Alexnet

**7).** Mean subtraction from data sometimes gives really worst performance, especially subtraction from grayscale images (I personally faced with this problem in foreground segmentation domain).

Same thing.

Fine-tune few layers or only train the classifier if you have a small dataset and you can also try to insert [Dropout](http://jmlr.org/papers/v15/srivastava14a.html) layers after convolutional layers that you’re going to fine-tune because it can help combatting overfitting in your network.

Same thing

**13).** Choose a right optimizer. There are many popular adaptive optimizers such as [Adam](https://arxiv.org/abs/1412.6980), [Adagrad](http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf" \t "_blank), [Adadelta](https://arxiv.org/abs/1212.5701" \t "_blank), or [RMSprop](http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf" \t "_blank) etc. [SGD+momentum](http://cs231n.github.io/neural-networks-3/" \l "sgd" \t "_blank) is widely used in various problem domains. There are two things to consider: F**irst**, if you care about fast convergence, use adaptive optimizers such as Adam, but it may get stuck in a local minima somehow and provides poor generalization (Figure below). **Second**, SGD+momentum can achieve to find a global minima, but it relies on robust initializations and it might take longer than other adaptive optimizers to converge (Figure below). I recommend you to use SGD+momentum since it tends to reach better optima.

Slightly better with SGD+momentum, yielded 55% with first k fold

**12).** Apply class-weights during training if you have highly imbalanced data problem. In another word, give more weights to the rare class but less weights to the major class. The class-weights can be easily computed using [sklearn](http://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html" \t "_blank). Or try to resample your training set using [OverSampling and UnderSampling techniques](https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis" \t "_blank). This can also help improving the accuracy of your prediction.

Done

# STEPS

Upgrade pip python -m pip install --upgrade pip

Use screen for background execution

See running processes

ps -u sacalderon

convert windows file to unix

Problem with \r, use nano to edit file.

TO WATCH GPU USAGE

Watch -n 1 nvidia-smi

Execute job in background python mainTests.py &

# Allow detachment, with tmux.

* ssh into the remote machine
* start tmux by typing tmux into the shell
* start the process you want inside the started tmux session
* leave/detach the tmux session by typing Ctrl+b and then d

You can now safely log off from the remote machine, your process will keep running inside tmux. When you come back again and want to check the status of your process you can use tmux attach to attach to your tmux session.

INSTALL ANACONDA

<https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart>

CHANGE FOLDER RIGHTS

sudo chmod -R 777 /var/DirectoryName

CREATE ENVIRONMENT

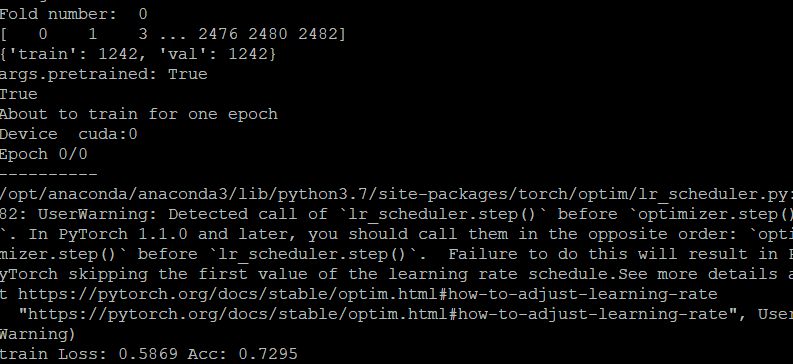
Conda create –name saul

ACTIVATE THE ENVIRONMENT

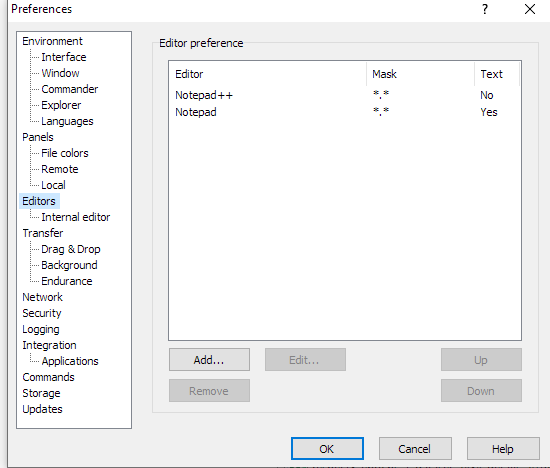
Conda activate saul

PROBLEM WITH TENSORFLOW 2.0, it does not have contrib

# K-FOLD is the same



# USE NOTEPAD++ in WinSCP



# TEST DEFINITION

Run 20 times a 5 fold for model RESNET18

Run 20 times a 5 fold for model RESNET50

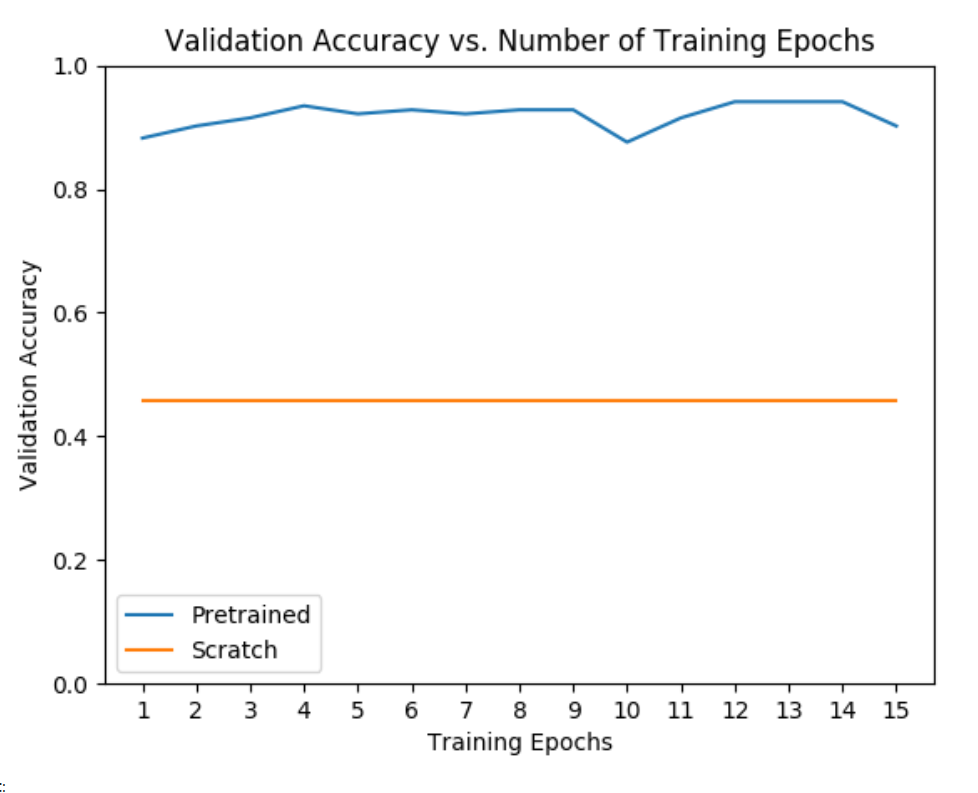
Run 20 times a 5 fold for model VGG16

Run 20 times a 5 fold for model Inception

# TUTORIAL ON EXTENDING EXISTING NETWORKS IN PYTORCH

<https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html>

With transfer learning this might happen



# DETAILS ABOUT MEAN TEACHER IMPLEMENTATION

## [mean-teacher](https://github.com/CuriousAI/mean-teacher)/[pytorch](https://github.com/CuriousAI/mean-teacher/tree/master/pytorch)/[experiments](https://github.com/CuriousAI/mean-teacher/tree/master/pytorch/experiments)/**cifar10\_test.py** / Jump to

[**Find file**](https://github.com/CuriousAI/mean-teacher/find/master)Copy path

Contains the experiments executed

# BATCH MINI BATCH

<https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/>