# 11785 Spring'22 Recitation-4

Hyperparameter Tuning

Bradley Warren, Ruoyu Hua, Urvil Kenia

# Table of Contents


- What are Hyperparameters? (and examples)
- Why do you need to do hyperparameter tuning?
- Strategies of hyperparameter tuning (Babysitting vs Caviar approach)
- Compute requirement considerations
- Effectively logging experiments (Sheets/ wandb)
- Inference from past experiments
- Domain specific intuition and hyperparameters
- Example with HW1P2

# What are Hyperparameters?

- Parameter
- Set prior to training
- Used to control the learning process
- Explicitly specified
- Variables that govern the training process itself

# Examples of Hyperparameters

- Batch Size
- Learning Rate
- Scheduler step size
- Drop out probability
- Context
- Size of layers/ number of layers
- Kernel size
- Stride

# Why do you need to do hyperparameter tuning?

- Accuracy
- Resources
- Time
- Because I need to reach the A cutoff….

# Strategies of hyperparameter tuning

- Babysitting
  - One model lots of fine tuning
- Caviar
  - Lots of models and parallel fine tuning

# Compute Requirement Considerations

- Both approaches suggested require varying amounts of compute. However, this becomes a key deciding factor while actually running the experiments.
- The babysitting approach requires less compute resulting in more time spent whereas, the caviar approach requires much more compute resources (which students typically don't have at their disposal).
- One suggested approach (given the fact most of us rely on Colab Pro or AWS g4dn's) is to train baseline models in different configurations first and then shortlist the promising models, and further tune them. This is a trade-off between the two approaches.
- If you do have unlimited compute, there are several tools that help you run grid searches or Bayesian Optimization (one example being Ray Tune).
- **Fun fact**: dividing hyperparameter search among your study group can help reduce compute requirement by 4x. :D

**PLEASE SAVE MODELS AT EVERY EPOCH!!!!**

# Effectively Logging Expts.

- ‘If you don’t know history, then you don’t know anything. You are a leaf that doesn’t know it is a part of a tree’ - Michael Crichton.
- Your brain is like a LSTM cell, it can remember some things but is controlled by a forget gate. Believe me, the forget gate will get activated at least once during the course of 3-week homeworks.
- It is **mandatory** that you log all your experiments in a very organized fashion so that you can build on your past learnings and they also help develop your intuition.
- We suggest 2 strategies of logging experiments - manually, using Google Sheets/ Excel or in a semi-automated fashion using wandb.ai.
- Both these methods have their own advantages and disadvantages.

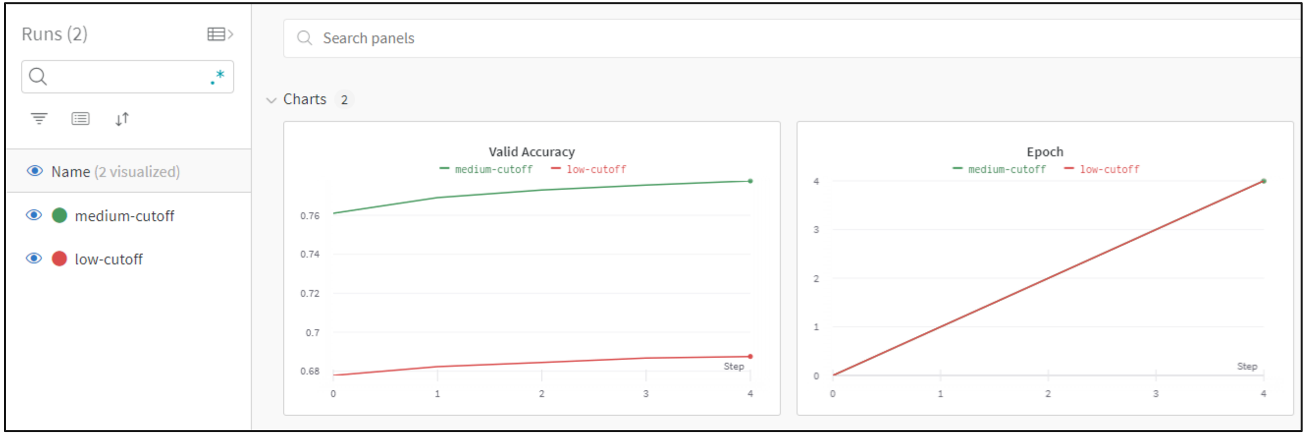
# Effectively Logging Expts. - Sheets

| | | | | | | |
|---|---|---|---|---|---|---|
| **Epoch** | 5 | 5 | 5 | 5 | 5 | 5 |
| **ctx** | 0 | 4 | 8 | 16 | 8 | 16 |
| **layers** | 2 | 2 | 2 | 2 | 4 | 4 |
| **activations** | relu | relu | relu | relu | splus | splus |
| **architecture** | Pyramid (max(1024, 10*D) --> 128) | Pyramid (max(1024, 10*D) --> 128) | Pyramid (max(1024, 10*D) --> 128) | Pyramid (max(1024, 10*D) --> 128) | inverted pyramid (D -->2048) | inverted pyramid (max(2D, 128) --> 4D) D--> -->2048 |
| **batchsize** | 256 | 256 | 256 | 256 | 512 | 512 |
| **dropout** | none | none | none | none | 0.25 every layer | 0.25 every layer |
| **BN** | none | none | none | none | every layer preactivation | every layer preactivation |
| **optimizer** | ADAM | ADAM | ADAM | ADAM | ADAM | ADAM |
| **scheduler** | steplr | steplr | steplr | steplr | reduce on plateau | reduce on plateau |
| **weight init** | gaussian | gaussian | gaussian | gaussian | xavier | xavier |
| **Regularization** | none | none | none | none | none | none |
| **Initial LR** | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |

# Effectively Logging Expts. - Sheets

- You should include all hyperparameters and track them in Sheets. Color code information for easy understanding and learning. Note that this is only an example.
- Start training simpler models with suggested parameters then follow a methodical approach basis the results you observe.
- This method has an advantage that is helps you develop intuition easily since you are tracking all parameters yourself.
- The obvious disadvantage is you have to manually enter the information, but the time is worth the rewards.

# Effectively Logging Expts. - wandb




| Name (2 visualized) | State | Notes | User | Tags | Created | Runtime | Sweep | bn | dropout | lr | num_layers | Epoch | Valid Accura |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| medium-cutoff | finished | Add notes | ukenia | | 3h ago | 29m 53s | - | true | 0.1 | 0.001 | 2 | 4 | 0.7777 |
| low-cutoff | finished | Add notes | ukenia | | 3h ago | 26m 43s | - | - | - | - | - | 4 | 0.6875 |

# Effectively Logging Expts. - wandb

- An example is shown in the notebook. Even in this case, your config for the run should have **ALL** hyperparameters (you do not know which ones you might need to compare).
- You should log all possible metrics (training loss, validation loss, validation accuracy, etc.) for effective future inference.
- The advantages of wandb are that it is semi-automated and can store more information such as loss curves, etc. (although you can store them in log files yourself). This abstracts away a lot of the manual labour required and provides better visualizations.
- The disadvantage is hidden in the advantage itself. Since it takes away a lot from your end, your learning is not as much as when you manually log information.

# Inference from Past Expts.

- Hyperparameter tuning is one of the most important part in any Deep Learning pipeline. Depending on the task at hand and the dataset, hyperparameters need to be very carefully tuned to get good results for any problem statement.
- You will get a taste of this during the Part 2's of every homework assignment and you would have seen the competitive nature of the leaderboard for HW1P2.
- However, please note that hyperparameter tuning is **NOT AN EXACT SCIENCE**. More often than not, you will work with empirical evidence to guide your choice of hyperparameters. This makes it crucial that you stay up to date with recent research papers and you discuss with your study group.

# Inference from Past Expts. (Contd.)

- Deep Learning also has a human learning component which is manual hyperparameter tuning. You can operated based on domain knowledge specific intuition as well as your own experiments.
- Do not form opinions based on 1 failed experiment. Even if you keep all other hyperparameters same and vary only 1, it is still not necessarily a direct effect of tuning it.
- Do not vary architectures in successive experiments by a huge factor, instead progressively build on your experiments. Verify your learnings at each step, that is truly the knowledge you will gain from this course.

# Domain specific intuition and hyperparameters

- Domain knowledge is important and could help you build models efficiently. It is of utmost importance to understand the task you are training the model for as well as the domain for that particular task.
- If you don’t have these intuitions:
    - Read papers
    - Work with your study group

# Example with HW1P2

- Domain intuition: phoneme classification => context is important
- Start your work with several simple architectures, find the best parameter combination
- Change your parameter settings to see if there is any improvement
- ....

# Example with HW1P2

| **Experiment** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| Context | 8, 16 | 16 | 16, 32, 48 | 32 | 32 |
| #Hidden Layers | 2 | 4 | 4, 6 | 6 | 5, 6 |
| Activation | ReLU | ReLU | Softplus, ReLU | Softplus, Tanh | Softplus |
| Architecture | Pyramid, Cylinder | Cylinder, Inverted pyramid, Diamond | Cylinder | Cylinder | Cylinder, Diamond |
| Dropout | None | None | 0.25 | 0.25, 0.35 | 0.25 |
| BN | None | None | None | None, Each Layer | Every layer |
| Optimizer | Adam | Adam, Adagrad | Adam | Adam | Adam |
| Scheduler | None | None | ReduceLROnPlateau | ReduceLROnPlateau | ReduceLROnPlateau |
| Weight Init | None | None | Xavier, Gaussian | Kaiming, Xavier | Kaiming |