# Assignment 3
# Neuroscience of Learning, Memory, Cognition
# Dr. Karbalaei Aghajan

Mohammad Hossein Shafizadegan
99104781

June 6, 2023

## Contents

# 1 Coding section

## Introduction to Steinmetz dataset

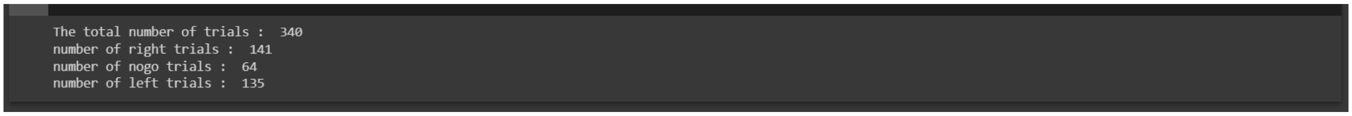
The Steinmetz dataset contains 39 Neuropixels recordings of 400-700 neurons each from across the mouse brain during a visual behavior task. The task involved presenting mice with two visual stimuli on either side of a screen and rewarding them for orienting their wheel towards the higher contrast stimulus. The dataset also includes information about the mouse's behavior, such as wheel position, pupil size, face motion, and licking.

The dataset was recorded using Neuropixels probes, which are high-density silicon probes that can record from hundreds of neurons simultaneously at multiple brain regions. The probes were inserted into different brain areas, such as the cortex, thalamus, hippocampus, striatum, and colliculus. The dataset also provides spike sorting results using Kilosort, a tool for fast and accurate spike sorting of large-scale recordings.

## Counting number of right, NoGo, Left trials

The "response" is an array with length of 340 which presents the number of total trials. Since the different kinds of trials (right, NoGo, left) are decoded by -1, 0, 1 we will count the number of each trials using the "count_nonzero()" numpy function as follows:

```
right_trials_num = np.count_nonzero(response == -1)
nogo_trials_num = np.count_nonzero(response == 0)
left_trials_num = np.count_nonzero(response == 1)
```



```
The total number of trials :  340
number of right trials :  141
number of nogo trials :  64
number of left trials :  135
```

## raster plotting the spike_time/neuron

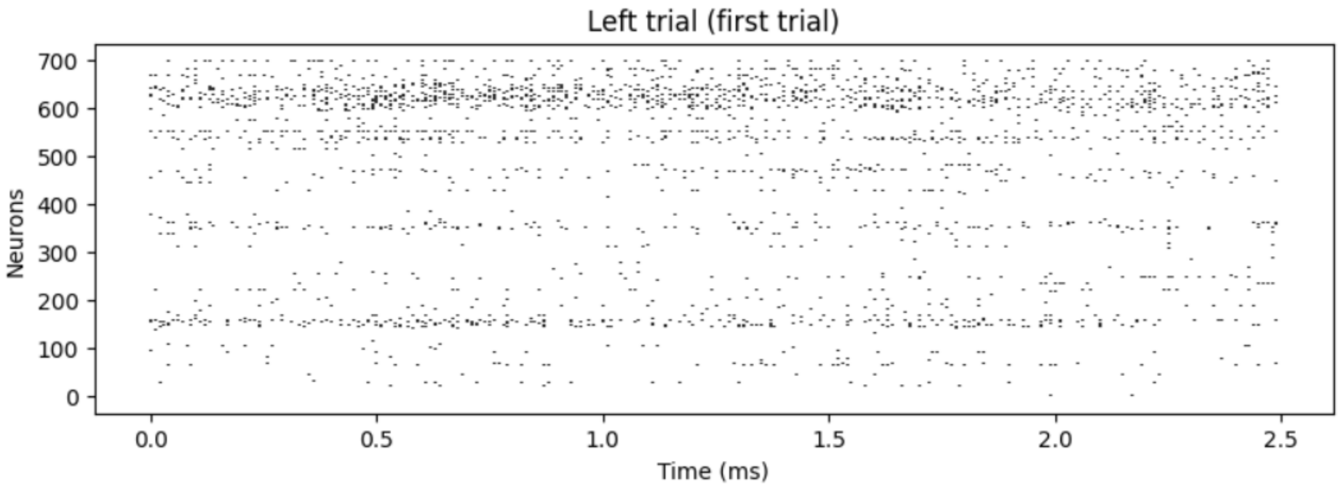
After separating the data of the different trials, we have defined a function called "create_raster_data". In this function the data of each type of trial and the number of a specific trial is given as input arguments. For each neuron in that specific trial, we will find the time when a spike is detected and also scale it with dt. The code of this function is as follows :

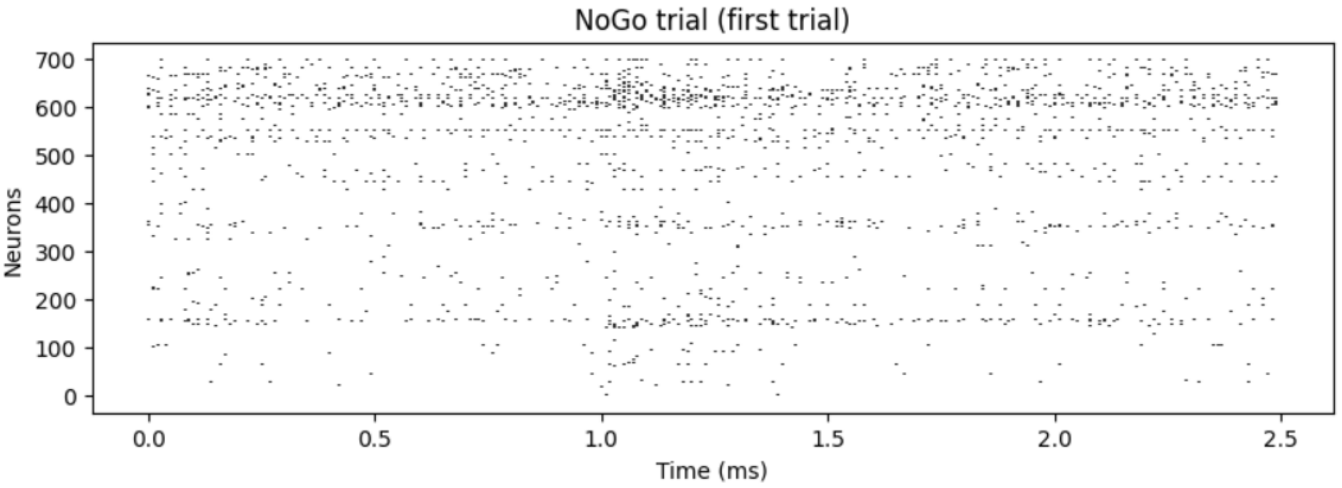
```
def create_raster_data(data, trial_num):
  raster = []
  for i in range(data.shape[0]):
    raster.append(np.where(data[i, trial_num, :] == 1)[0]*dt)
  return raster
```

Finally we visualize our data using the "eventplot()" function from matplotlib in our defined function "plot_raster()".

```
def plot_raster(data, title):
  plt.figure()
  plt.rcParams['figure.figsize'] = (10,3)
  plt.eventplot(data, color=".2")
  plt.title(title)
  plt.xlabel('Time (ms)')
  plt.ylabel('Neurons')
  plt.show()
```

Here we can observe the raster plot of the first trials for left trials for example. The raster plots for other kind of trials can be seen in the jupyter notebook.





By observing and examining the raster plots for the first trial of each kind of right, left and NoGo trials we can state that the neuron with id 600 (neuron #600) is perhaps the most active neuron while some other neuron for example neuron #1 is mostly inactive with a very low firing rate. This pattern can be seen in almost all three type of trials.

Using these raster plots for each trials, if we can find a neuron that has a significant firing rate for only one of the right, left or NoGo trial and simultaneously that neuron has a very low firing rate for the rest of types of trials, that neuron can be good candidate for detecting the stimuli which can be useful for decoding issue. It can be also useful for finding the feature and tuning curve of neurons too. Here we saw that the neuron #600 is almost very active for all types of trials so perhaps it can't be a good candidate but if we consider the timing there are some difference for NoGo in comparison to Right or Left. Also the activity of the neuron #150 (approximately) is quit different for NoGo and Left trial.

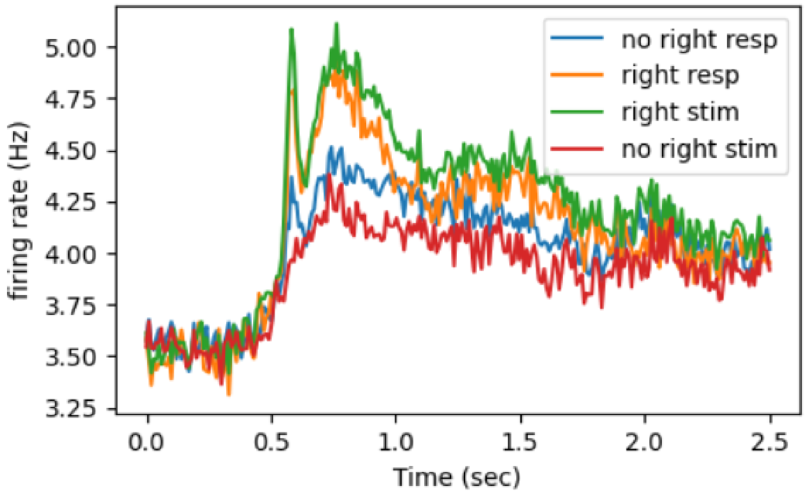
## Basic population average

First we prepare data for each of the Left trials, right trials, right stimulus and no right stimulus categories as follows

```
    no_Right_trial = dat['spks'][:, response != -1,:]
    Right_trial = dat['spks'][:, response == -1,:]
    Right_stimul = dat['spks'][:, vis_right != 0,:]
    no_Right_stimul = dat['spks'][:, vis_right == 0,:]
```

Then we will create the ”plot_data()” function. By parsing the input data to the function we will calculate the average through neurons and trials which are the first and second dimension of the input matrix. Finally we visualize the results.

```
    avg1 = np.average(data, axis=0)
    avg2 = np.average(avg1, axis=0)
```



What we have done in this figure is that we have extracted the trials that we knew the type of stimulus (right or non-right) and then we have plotted the corresponding average neuron firing rate which are the red and green curves in the figure.
Then we want to see the performance of the mouse when it faced the stimulus. So we extract the spike data regarding the mouse response. The average firing rate of the neurons corresponding to the times that the mouse response was ”right” can be seen in the orange curve and the times that its response was ”no right” can be seen in the blue curve.

It can be seen that blue curve and the red curve are quit the same which can be concluded that the mouse could detect the non-right stimulus with quit a good accuracy. Also it can be seen that the orange and green curve are close to each other and behave quit the same over time so the mouse could also detect the right stimulus with good accuracy.

Also based in the figure, it can be seen that the right stimulus and the non-right stimulus curves are separable and can be easily detected and recognized especially in the $[0.5, 1]$ time interval.

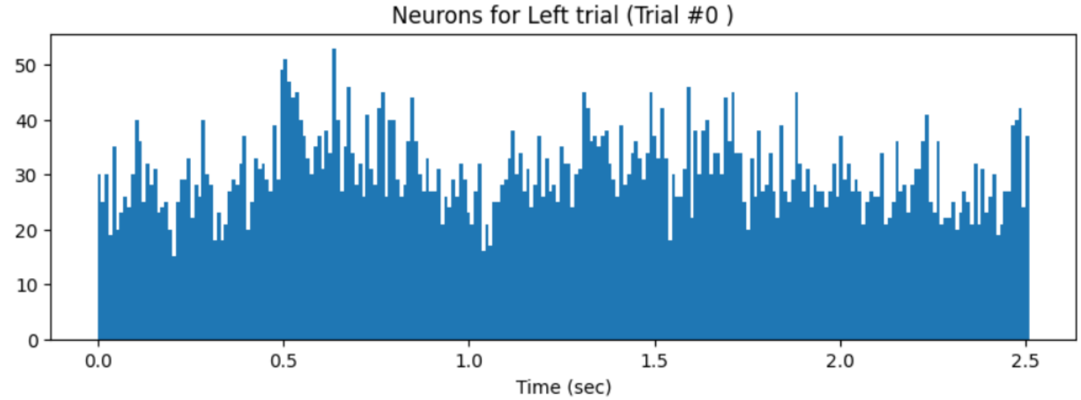
## The Peri-Stimulus Time Histogram (PSTH)

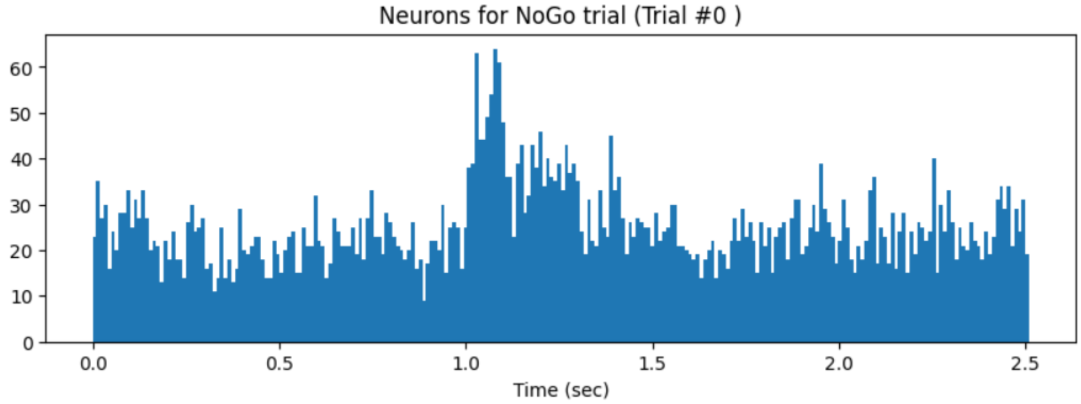
In order to plot the PSTH across the trials, we should first extract and separate the data corresponding that specific number of neuron and then for the resulting 2D matrix we sum the data over the first dimension to achieve an array which represents the number of spikes through all trials for a specific neuron over time. Finally we visualize the results using matplotlib ”bar()” function properly.

In order to plot the PSTH across the neurons, we should first extract and separate the data corresponding that specific number of trial and then for the resulting 2D matrix we sum the data over the first dimension to achieve an array which represents the number of spikes through all neurons for a specific trial id over time. Finally we visualize the results using matplotlib ”bar()” function properly.

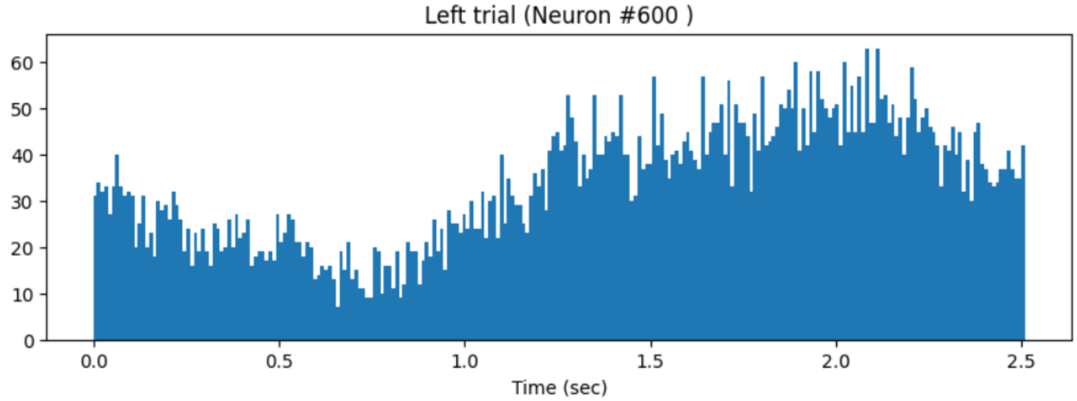
```
    hist_data = sum(data)
    plt.bar(np.linspace(0,2.5,250),hist_data,align='edge', width=0.01)
```

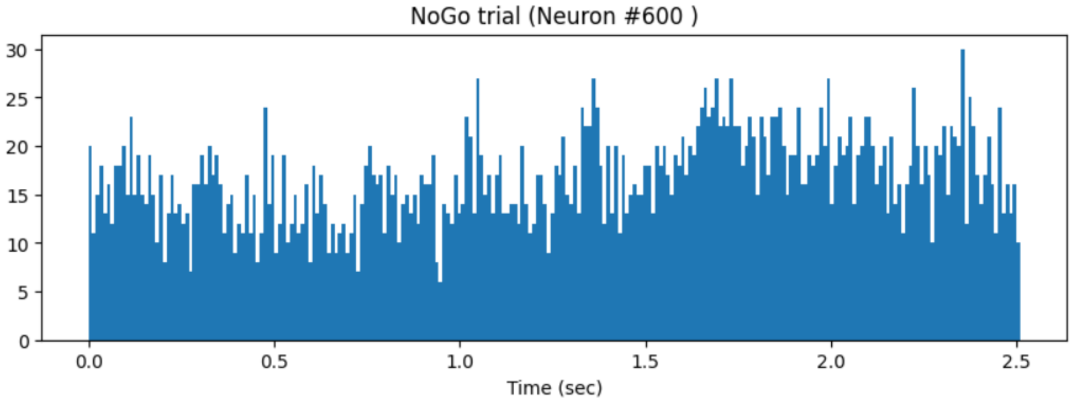
We have plotted several histograms for different trial types in the jupyter notebook. Here we can see some of them for example.





Based on the figured seen so far it can be said that roughly, most information can be extracted starting from $t = 1$. In this case by comparing the above figures it can be seen that the number of active (spiking) neurons for the NoGo trial is much bigger (almost 3 times more) than the number of neurons of the Left trial around $t = 1$. This can clearly separate these type of trials.





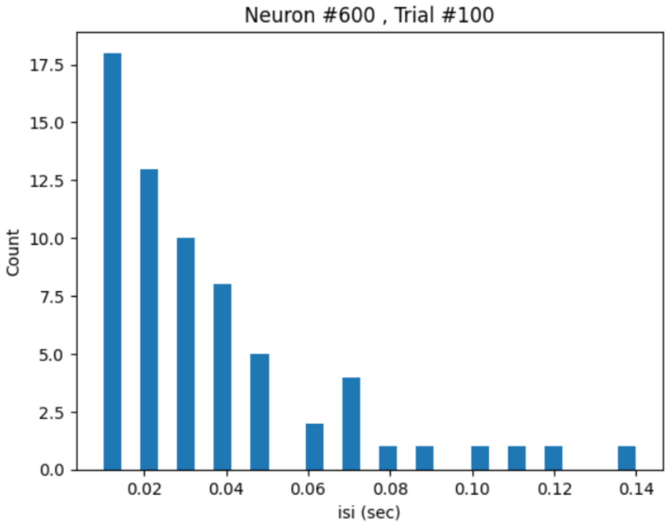
Based on the 2 above figures that illustrates the number of trials that the specific neuron #600 spikes over time for Left trial and NoGo trial, it can be seen that this neuron is generally more active for the left trial in comparison to the NoGo trial. Starting from $t = 1$, the left and NoGo trial have the value of about 20 while for the NoGo trial this value won't change significantly over time but vice versa, for the left trial it will increase noticeably and will reach values of 50-60.
It can be seen that this neuron behaves differently for the left and NoGo trial.

## Inter-spike intervals and their distributions

In this section we will calculate the ISI for a specific neuron in a specific trial as follows :

```
    spk_times = np.where(dat['spks'][neuron_id, trial_id, :] == 1)[0] * dt
    isi = np.diff(spk_times)
```

Finally we visualize the result using matplotlib "hist()" function.



In this histogram the total number of spike intervals equals to 66. As it can be seen, the maximum value for isi is 140 $ms$. This means that the longest this neuron will wait is 140 ms.
Based on this histogram it can be seen that 54 ISI is less than 40 ms which is approximately 82 % of all intervals.

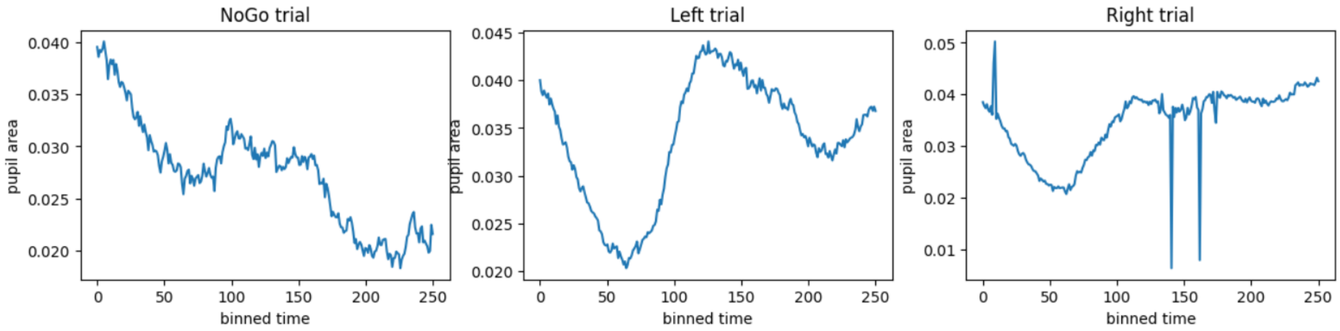
## Behavioral data

First we extract the pupil data for each type of trials as follows :

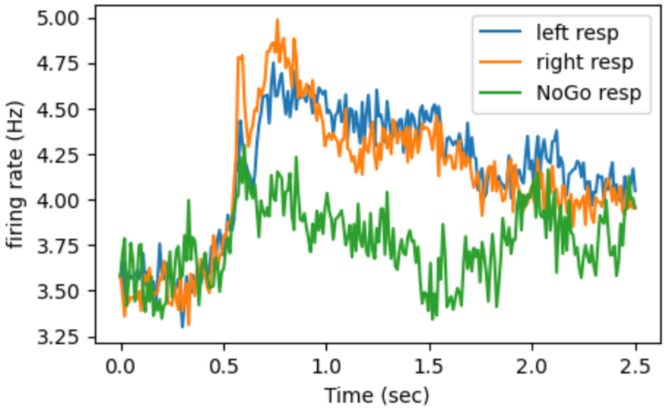
```
    NoGo_trial = dat['pupil'][:, response == 0,:]
    Left_trial = dat['pupil'][:, response == 1,:]
    Right_trial = dat['pupil'][:, response == -1,:]
```

Then we compute the average of these data over trials and finally visualize the results.

```
    plot_pupil(np.average(NoGo_trial[0,:,:], axis=0), 'NoGo trial', 1)
    plot_pupil(np.average(Left_trial[0,:,:], axis=0), 'Left trial', 2)
    plot_pupil(np.average(Right_trial[0,:,:], axis=0), 'Right trial', 3)
```



Pupil dilation is a measure of how much the black center of your eyes (pupils) are larger than usual.
In the Steinmetz dataset, pupil dilation is used as an indicator of arousal and attention during the visual behavior task. The pupil dilates when the brain is processing information or when the subject is aroused.



Pupil dilation may vary depending on the contrast, location and timing of the stimuli, as well as the feedback and reward that the mice received.

Based on these figures, for NoGo trials since there is no visual stimulus on the sides of the screen the mouse pupil size start to decrease overtime which means that there is nothing interesting for the mouse. Indeed for the right or left trials when the neurons activity and the average firing rate start to increase significantly, the pupil size also starts to increase and pupil dilation can be observed which can be mean that there is some thing interesting for the mouse which is actually the visual stimulus on the screen.

# 2 MAP and ML Estimations

## Part 1

We should find the value that maximizes the likelihood function, which is the probability of the data given the mean.The likelihood function for a sample of n data points $x_1, x_2, ..., x_n$ is given by:

$$L(\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \quad \Rightarrow \quad \ln L(\mu) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

$$\frac{d}{d\mu}\ln L(\mu) = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) = 0 \quad \Rightarrow \quad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

This is the ML estimate of the mean of a normal distribution with known variance. It is equal to the sample mean.

## Part 2

We have to find the value that maximizes the posterior distribution, which is the product of the likelihood function and the prior distribution.
The prior distribution is also a normal one with parameters $\mu$ and $\sigma^2$, then the posterior distribution is given by:

$$P(\mu^*|x_1, x_2, , x_n) \propto L(\mu^*)P(\mu^*) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu^*)^2}{2\sigma 2}} \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mu^*-\mu)^2}{2\sigma^2}} \quad \Rightarrow$$

$$\ln P(\mu^*|x_1, x_2, , x_n) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu^*)^2 - \frac{1}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mu^*-\mu)^2 + C$$

$$\frac{d}{d\mu^*}\ln P(\mu^*|x_1, x_2, , x_n) = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu^*) + \frac{1}{\sigma^2}(\mu^*-\mu) = 0 \quad \Rightarrow$$

$$\mu^* = \frac{\sigma^2}{n\sigma 2 + \sigma^2}\mu + \frac{n\sigma^2}{n\sigma^2 + \sigma^2} \times \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{\mu}{n+1} + \frac{1}{n+1}\sum_{i=1}^{n} x_i = \frac{1}{n+1}\left(\mu + \sum_{i=1}^{n} x_i\right)$$

It is a weighted average of the prior mean and the sample mean.

The prior distribution represents the initial belief or assumption about the parameter before observing any data. The posterior distribution represents the updated belief or inference about the parameter after observing the data.

When the prior distribution and the distribution of the sampled data are the same, it means that the prior distribution is consistent with the likelihood function, and the posterior distribution will be similar to both of them.

In this case, the posterior distribution will also be a normal distribution with the same parameters as the prior and the data. This means that your data will not change your belief about the parameter at all, and your posterior distribution will have a very high precision and a very low uncertainty. This is an extreme case of consistency between the prior and the likelihood function. It may indicate that you have a very strong prior belief that is confirmed by the data, or that you have a very small or biased sample of data that does not reflect the true variability of the population.

## Part 3

When the number of samples is large, MAP and ML estimations tend to converge to the same value, regardless of how informative or uninformative the prior distribution is. This is because the likelihood function becomes more peaked and more influential as more data are observed, and it overwhelms the effect of the prior distribution.

# 3 Stimulus-response power laws

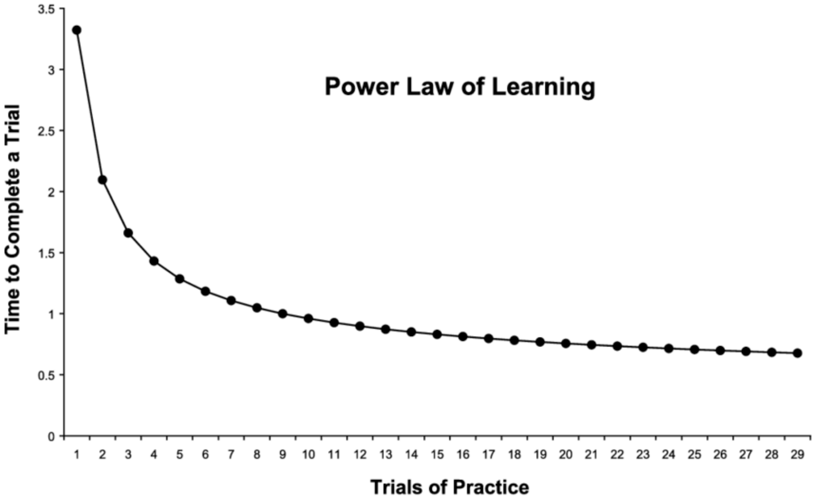
Skill acquisition is the process of learning and improving one's ability to perform a task or activity. Skill acquisition can be influenced by various factors, such as the type of task, the amount and quality of practice, the feedback and guidance received, and the individual characteristics of the learner.

One of the most widely observed phenomena in skill acquisition research is the power law of practice, which states that the logarithm of the reaction time for a particular task decreases linearly with the logarithm of the number of practice trials taken. In other words, the more one practices a skill, the faster one becomes at performing it, but the rate of improvement decreases over time. This implies that there are diminishing returns to practice, and that achieving mastery or expertise requires a large amount of deliberate and sustained practice.

The power law of practice has been demonstrated in various domains and tasks, such as typing, memory, problem solving, sports, and music. It is considered to be one of the most consistent and robust findings in skill acquisition research, and it has important implications for designing effective learning and training programs.

There are many factors that can influence the shape and slope of the learning curve, such as individual differences, task complexity, transfer and retention. Therefore, the power law of practice should be applied with caution and flexibility to real-world skill acquisition.

One possible explanation for how exercise affects skill acquisition is based on Fitts and Posner's three-stage model. This model suggests that skill acquisition involves three stages: cognitive, associative, and autonomous. In the cognitive stage, one learns what to do and how to do it by following instructions and feedback. In the associative stage, one practices what to do and how to do it by trying and correcting errors. In the autonomous stage, one becomes proficient and consistent at doing what to do and how to do it with little attention and effort. Exercise can help one move from the cognitive to the associative stage faster by optimizing one's mindset to improve alertness, attention, and motivation. Exercise can also help one move from the associative to the autonomous stage faster by increasing a hormone called irisin, which can help neurons survive and grow in the hippocampus, a part of the brain that is important for memory.



There is an experiment designed to test the power law for skill acquisition. the task is mirror reading.
Mirror reading is a task that involves reading words or sentences that are reversed horizontally, as if they were reflected in a mirror. For example, the word "mirror" would appear as "rorrim" in mirror reading. This task is

challenging for most people who are used to reading normal text, but it can be learned with practice.

One experiment that was designed to test the power law of skill acquisition for mirror reading was conducted by psychologist James McKeen Cattell in 1886. He asked participants to read aloud passages from a book that were printed in mirror-reversed text. He measured the time it took them to read each passage and the number of errors they made. He found that the participants improved their mirror reading speed and accuracy with practice, and that their improvement followed a power law function. That is, their reaction time decreased linearly with the logarithm of the number of practice trials. He also found that the participants retained their mirror reading skill even after several months without practice.

Based on this experiment, we can infer that more practice leads to more efficient and effective neural firing. This is because practice strengthens the connections between neurons that are involved in performing a skill, and also increases the insulation around the axons. This insulation is called myelin, and it helps speed up the transmission and reduce interference. Therefore, more practice means more myelin, which means faster and more accurate neural activity.This means that the neurons can fire with less effort and more accuracy, and that they can coordinate better with other neurons involved in the same skill. This leads to better performance and less errors .

Based on this experiment, and other research on learning and forgetting, we can infer that when we learn a skill and then forget it after a while, we still retain some traces of the neural connections that were formed during the initial learning. This means that when we want to learn that skill again, we can do it faster and easier than the first time and the process of learning won't obey the power law for skill acquisition, because we can reactivate those dormant connections and strengthen them with practice. This phenomenon is called relearning or the savings effect, and it shows that forgetting is not a complete erasure of memory, but rather a weakening of access to it.