



# ReaLLM ASIC

Make Your Own Lightweight LLMs

# Outline of the Tutorial

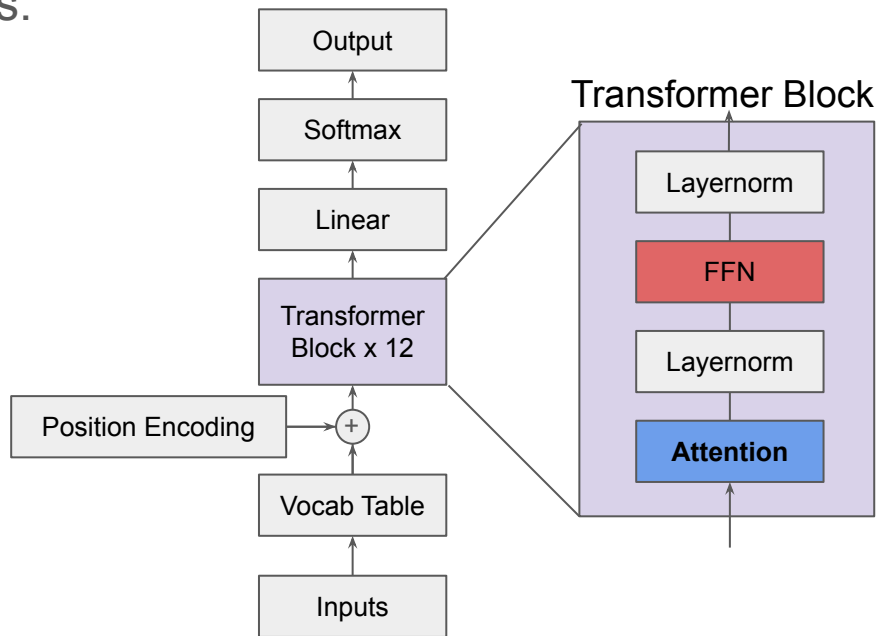
- Intro
  - LLMs and Applications
  - Recap of LLM architecture
- Hands-on AI From Scratch:
  - Building and training a custom lightweight LLM
  - Data preparation and preprocessing
  - Model training options and optimization
- Checkpoints and Finetuning

# Applications Limited Only by Datasets

## LLM Transformer Architecture

One architecture, limited only by datasets:

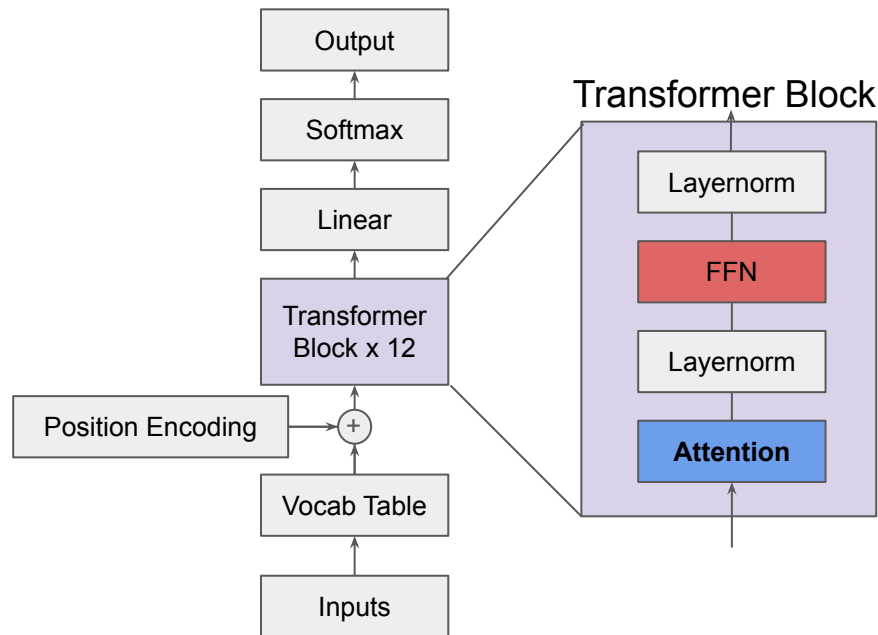
- Translation
- Poetry
- Music
- Robotics Motion
- Cooking Recipes
- Editing and Writing Reports



# Main Hyperparameters

- “Height” - Number of Layers
  - Deep Networks -> Abstract Knowledge
  - Linearly increases size of network
- “Width” - Dimensions per Token
  - Better Per Token Understanding
  - Non-linear increase in size

## LLM Transformer Architecture



# Datasets

We are limited by what we can train on.

- Music
  - We'll work with JS Bach
- Language
  - Example with Japanese and English
- Generation
  - Example with Shakespeare

# Tokenization Speed-Up Example

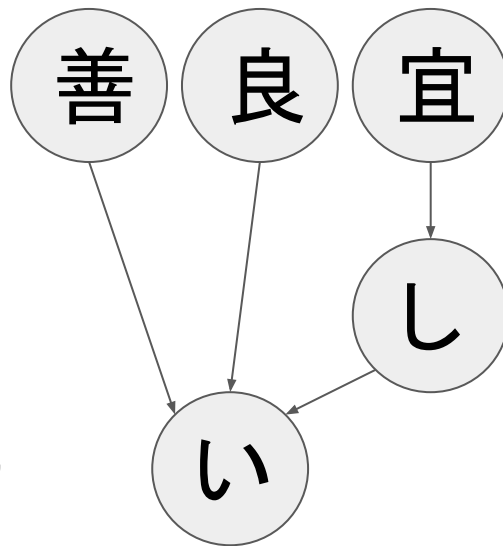
Tokens need to gain “Experience Points”:

- Higher frequency of a token in the dataset then better the LLM will learn about it.

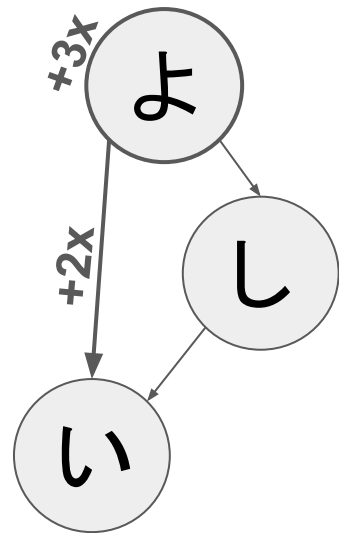
Language Example:

- Preprocessing Kanji -> Hiragana:
  - Faster training (smaller model)
  - Faster accumulation of “experience points”

Kanji / Hiragana



Hiragana



Colab

# Workshop Colab

- [Workshop](#) colab
  - based on popular “nanoGPT” framework
- We’ll migrate to the Colab for the remainder of the workshop.

```
Run GPU Training

[ ] !python3 data/shakespeare_char/prepare.py

length of dataset in characters: 1,115,394
all the unique characters:
!$&',-.3:;?ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
vocab size: 65
train has 1,003,854 tokens
val has 111,540 tokens

!python3 train.py --device="cuda" --dtype="float16" --max_iters=

2024-05-13 06:59:01.957995: E external/local_xla/xla/stream_executor
2024-05-13 06:59:01.958075: E external/local_xla/xla/stream_executor
2024-05-13 06:59:02.106433: E external/local_xla/xla/stream_executor
2024-05-13 06:59:02.395367: I tensorflow/core/platform/cpu_feature_guard.cc:182
To enable the following instructions: AVX2 FMA, in other operations
2024-05-13 06:59:04.921801: W tensorflow/compiler/tf2tensorrt/utils.cc:60
seed: 1337
seed offset: 0
number of parameters: 2.98M
num decayed parameter tensors: 16, with 3,072,384 parameters
num non-decayed parameter tensors: 13, with 4,992 parameters
using fused AdamW: True
step 0: train loss 4.2075, val loss 4.2067
iter 0: loss 4.2097, time 14979.17 ms, mfu -100.00%
iter 10: loss 3.3180, time 130.96 ms, mfu 1.00%
iter 20: loss 3.2074, time 132.72 ms, mfu 1.00%
iter 30: loss 2.9536, time 131.67 ms, mfu 1.00%
iter 40: loss 2.7902, time 132.85 ms, mfu 1.00%
iter 50: loss 2.6889, time 132.28 ms, mfu 1.00%
iter 60: loss 2.6815, time 132.55 ms, mfu 1.00%
iter 70: loss 2.6102, time 131.35 ms, mfu 1.00%
```