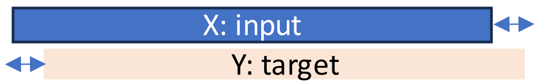
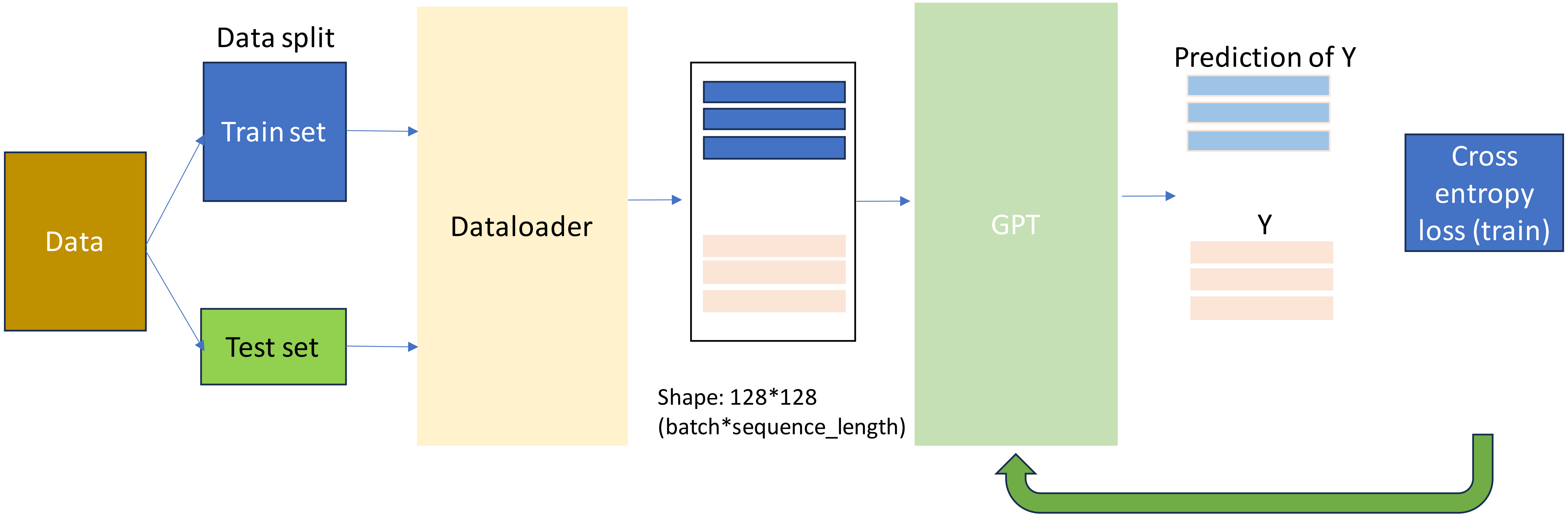
Presentation for mini-project

Yuanyuan ZHENG

Pipeline

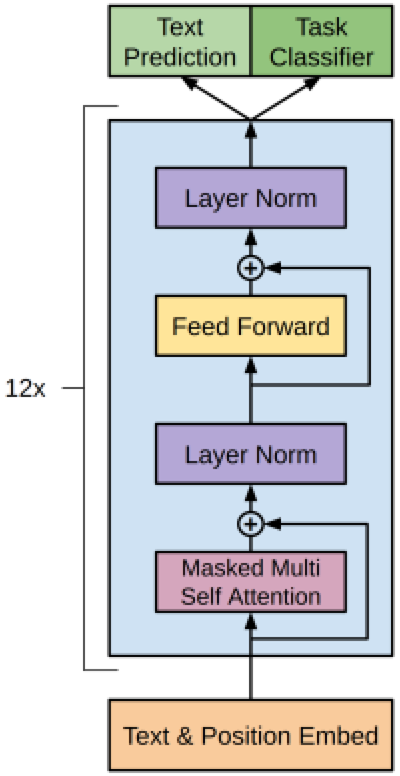


X: input
Y: target

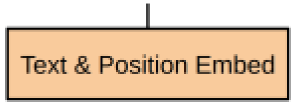


Data
Data split
Train set
Test set
Dataloader
Shape: 128*128
(batch*sequence_length)
GPT
Prediction of Y
Y
Cross entropy loss (train)

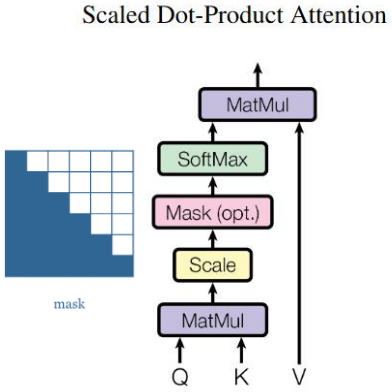
Model architecture and implementation

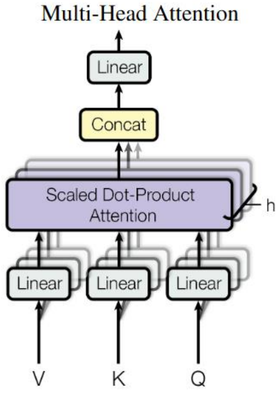


Decoder-Only Architecture used by GPT-2.

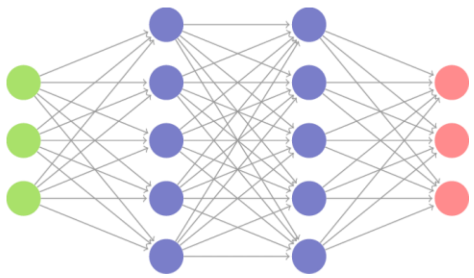


Input token (char) + positional embedding

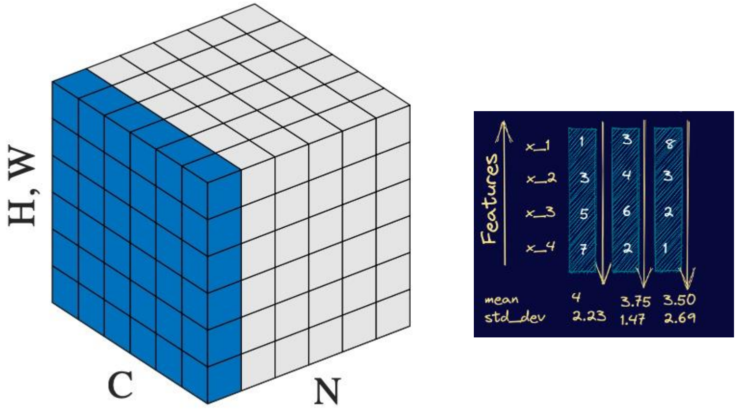




Masked multiheads self attention



Feed forward



Layer normalization

Project structure

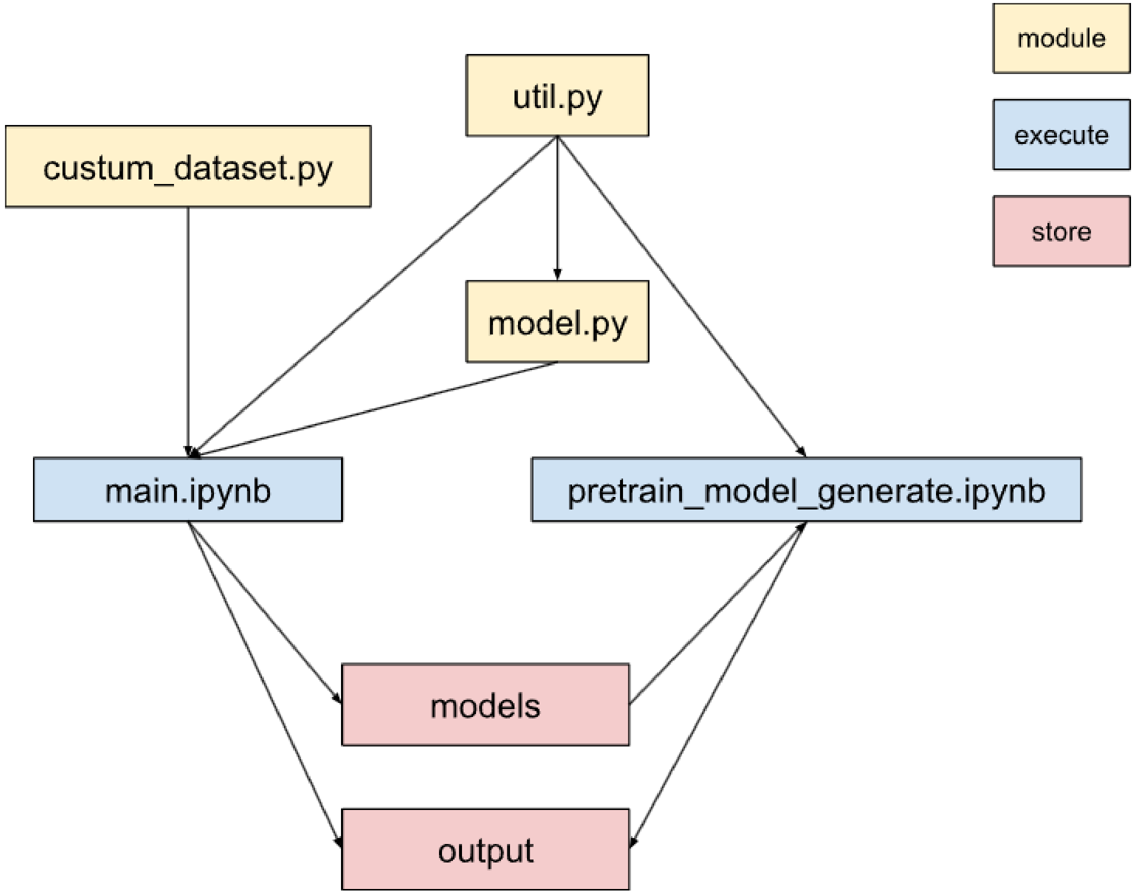
Model.py

Costum_dataset.py

util.py

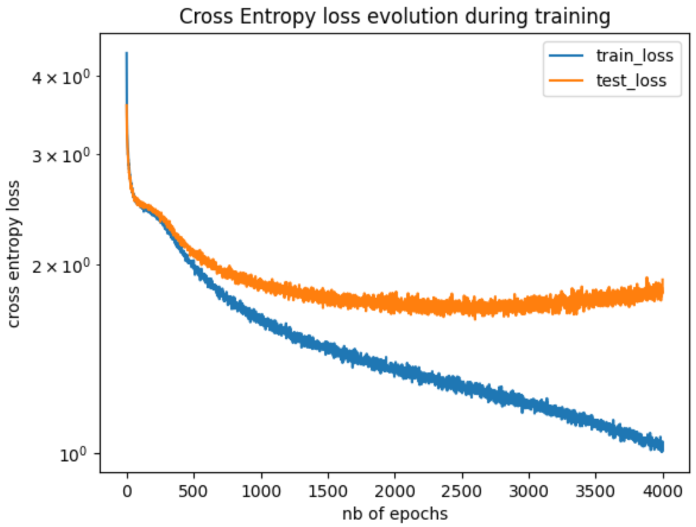
Main.ipynb

pretrain_model_generate.ipynb

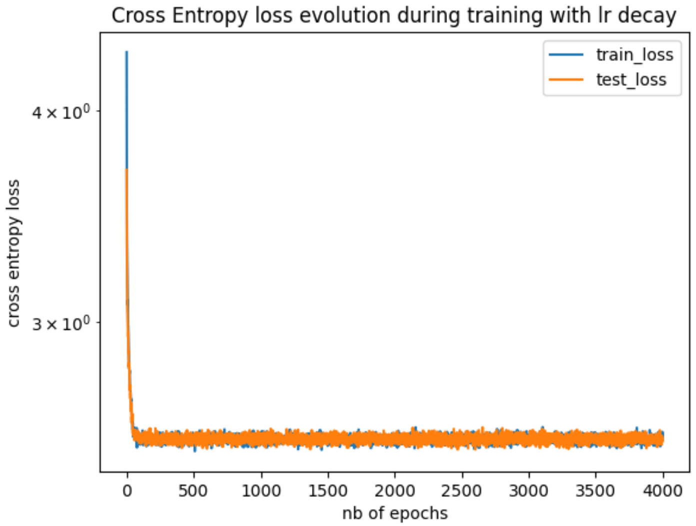


Learning rate decay

- Learning rate decay: using learning rate scheduler (MultiStepLR) ,with milestones of each 50 steps.



Without learning rate decay



With learning rate decay

Using same random seed

Generated text comparison (4000 epochs without learning rate decay)

- I saw a cat jumping onto the table, he
- falles our wife's and crows, thou art head of theirful tawn.

- GLOUCESTER:
- But to him again, and for all truth:
- We make men, what is a blottled from me?
- Why looks chamberly from a crubb his holy?

- NORTHUMBERLAND:
- Not I, my lord?

Without lr decay

- I saw a cat jumping onto the table, wins s hiss yPe bot arins chechy:
- Ghifor.
- Thuerang, gealal tavo
- xouthy, at bulcacereapndothed owirushed hy ORCENI oded atoprlourowoul pr wnd we hass hounes t s, thertidr,
- I hed ord o,
- Yod, mad fousht, wat hard nchaTHhed thedetilend pe; bus t there's, ilis'EUSesst eal's, hef my.
- g nd

With lr decay

Backup

torch.nn.Embedding(*num_embeddings*, *embedding_dim)*

- **num_embeddings** (*int*) – size of the dictionary of embeddings
- **embedding_dim** (*int*) – the size of each embedding vector

torch.nn.Linear(*in_features*, *out_features*, *bias=True)*

- **in_features** (*int*) – size of each input sample
- **out_features** (*int*) – size of each output sample

Torch.nn.CrossEntropyLoss

- **Input**: Shape (C), (N,C) or $(N,C,d1\,,d2\,,...,dK\,)$ with $1 \geq K \geq 1$ in the case of K-dimensional loss. (C: # of class, N: batch size)
- **Target**: If containing class indices, shape $()$,(N) or $(N,d1\,,d2\,,...,dK\,)$ with $1K \geq 1$ in the case of K-dimensional loss where each value should be between $[0,C)$. If containing class probabilities, same shape as the input and each value should be between $[0,1]$.
- **Output**: If reduction is ‘none’, shape $()$, (N) or$(N,d1\,,d2\,,...,dK\,)$ with $1K \geq 1$ in the case of K-dimensional loss, depending on the shape of the input. Otherwise, scalar.