THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Name Generation with Autoregressive Character-level Language Modeling

Eljan Mahammadli

Prof. Steve Kaisler, Prof. Jamaladdin Hasanov

6917

03/08/2023

Project Objective & Heilmeier Questions

- **What are you trying to do?**

Automate name generation.

- **How is it done today, and what are the limits of current practice?**

Manual, random or basic rule-based systems.

- **What is new in your approach and why do you think it will be successful?**

Encapsulating multiple deep learning models in a single, flexible system trainable on any name collection.

- **Who cares?**

Entrepreneurs, writers, and marketers

- **What are the risks and payoffs?**

Name quality and originality; the payoff is an automated, versatile, and creative name-generation tool.

- **How much did it cost?**

Primarily time and deep learning expertise, minimal financial cost. Trained models on Kaggle GPU kernels.

- **How long did it take?**

Two months from inception to completion.

- **What are the midterm and final "exams" to check for success?**

Midterm: successful development and testing of initial models. **Final**: completed all targeted models and combined them in single system. Additionally, tested model on different datasets to prove its versatility.

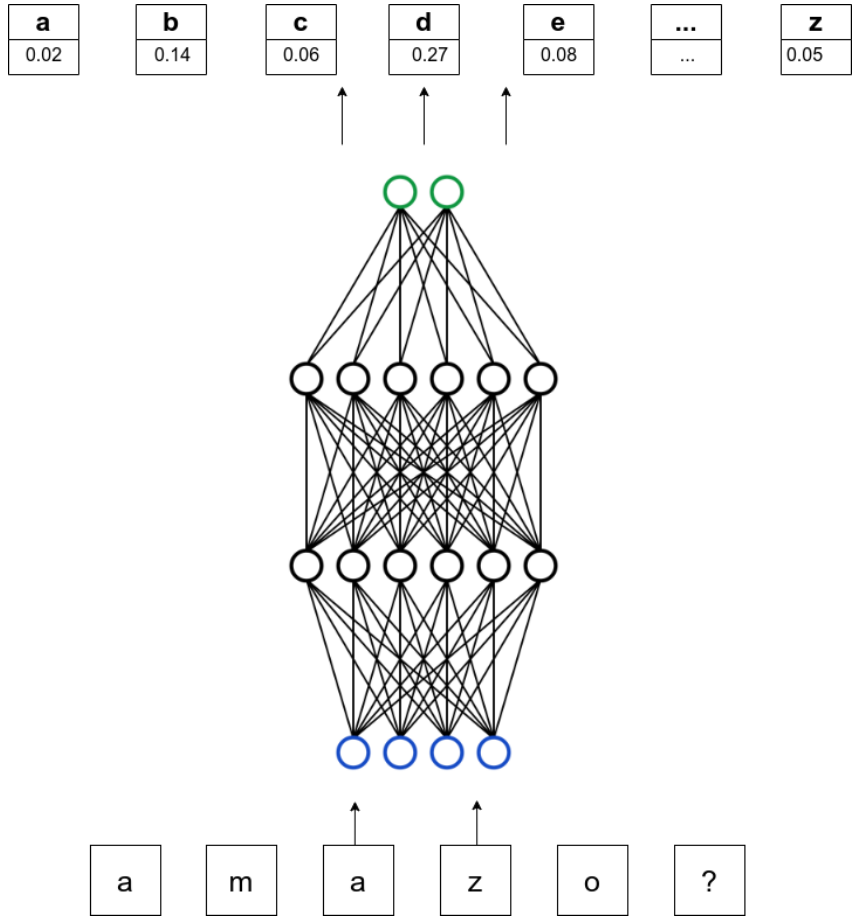
What is Character-level Language Model?

- Language models can predict the next token in a sequence.
- Probability is typically dependent on the preceding n tokens.
- Generate novel and valid names.

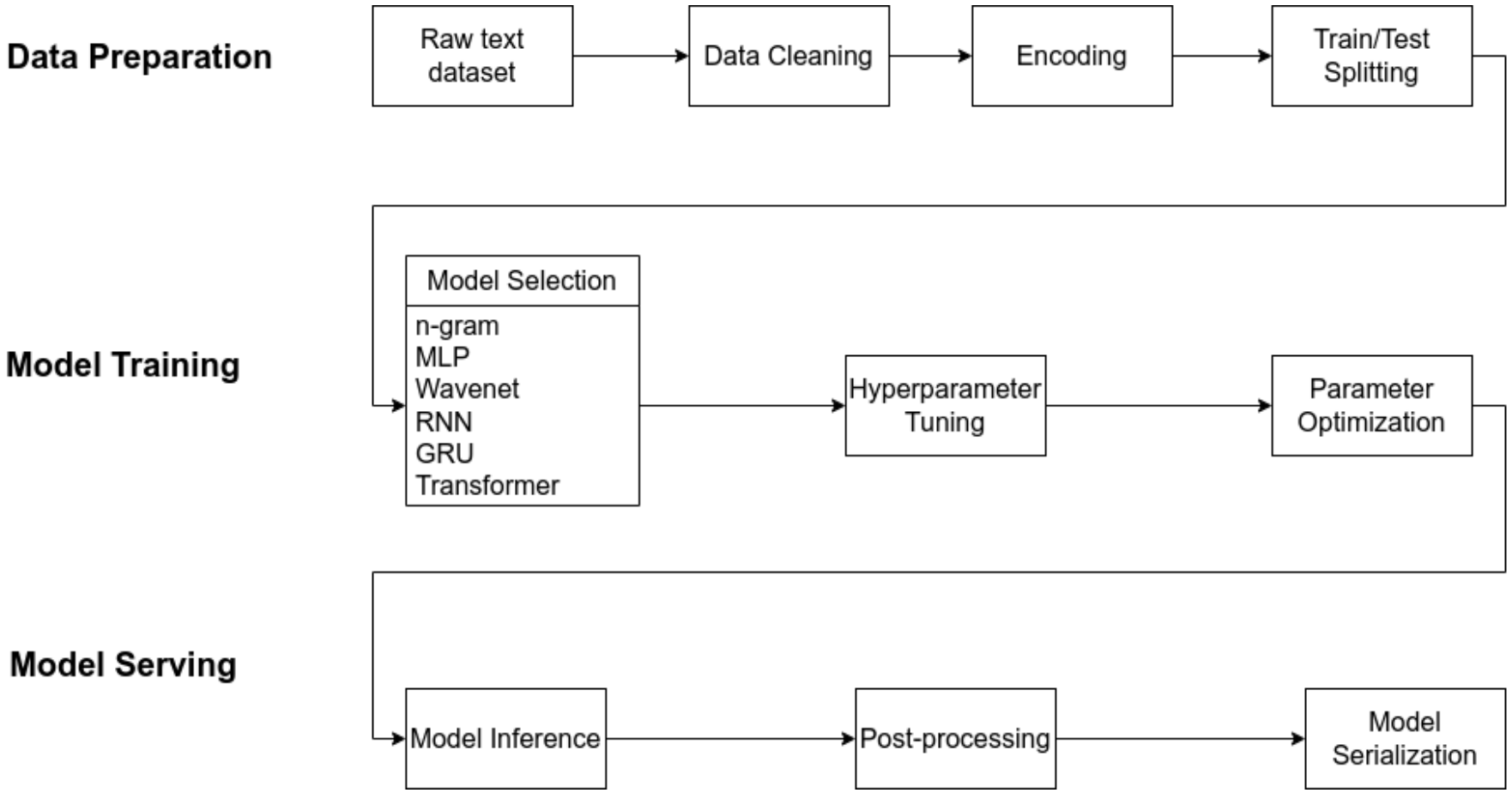
$P(w_1,w_2, \ldots,w_T) = \prod P(w_t \mid w_1, \ldots, w_{t-1})$ for t=1 to T

- P: Joint probability of the sequence of characters
- wt: target token

When predicting the character after "a m a z o", the model is estimating the probability P('?' | 'a', 'm', 'a', 'z', 'o').



System Architecture



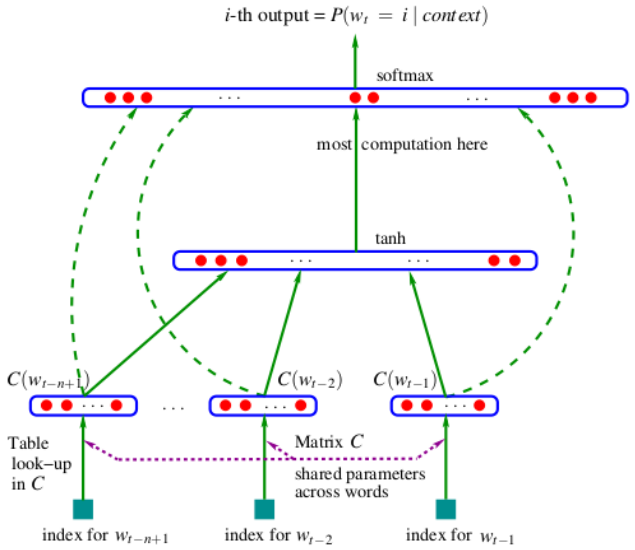
Key steps

Explored, developed, and conducted analyses of pivotal models and architectures that have significantly influenced the evolution of language modelling.

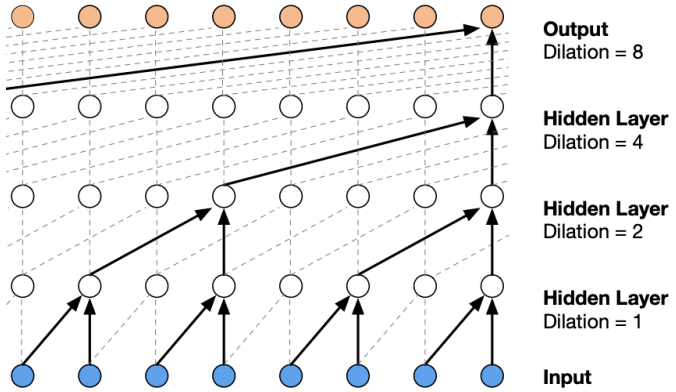
- **N-gram (1970s for LM)** - probability distribution of pairs of consecutive characters
- **MLP (Bengio et al. 2003)** - 10-d feature vector, 200 hidden neurons, 3 block size, 11 897 total parameters.
- **Wavenet (DeepMind WaveNet 2016)** - 24-d character embedding space, 128 neurons in each hidden layer, 76 579 total parameters
- **RNN (Mikolov et al. 2010)** - 4 layers, and 64 nodes per layer and hidden neurons, 11 803 total params
- **GRU (Kyunghyun Cho et al. 2014)** - same as RNN, 28 315 total parameters
- **Transformer (Vaswani et al. 2017)** - 200K total parameters

Model Architectures

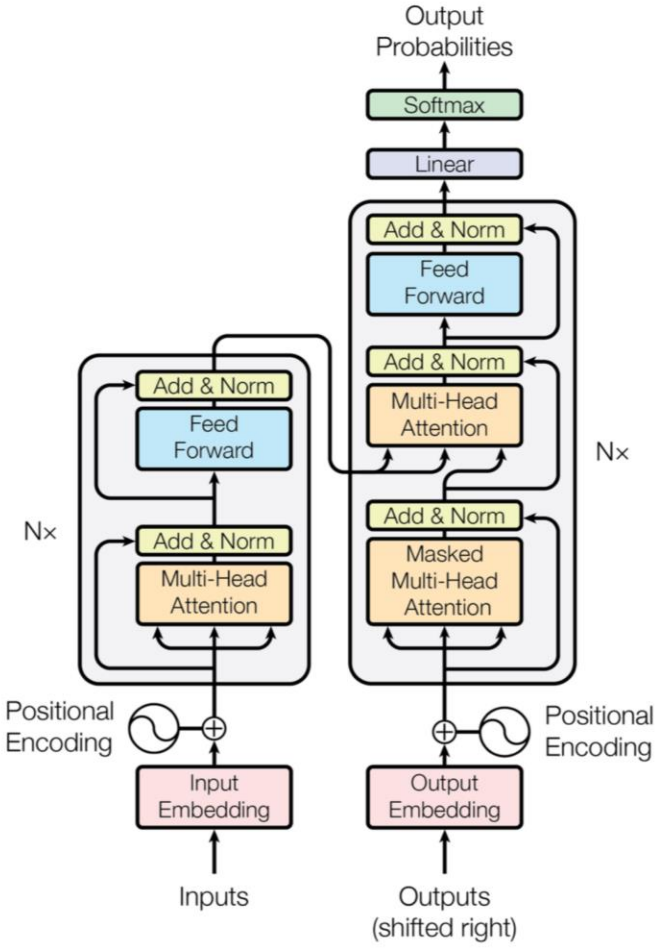
MLP



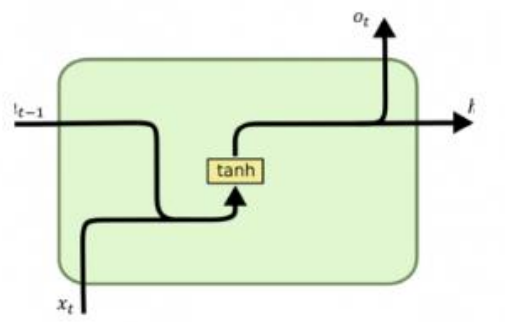
WaveNet



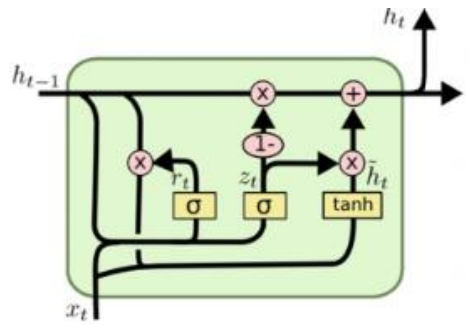
Transformer



RNN



GRU



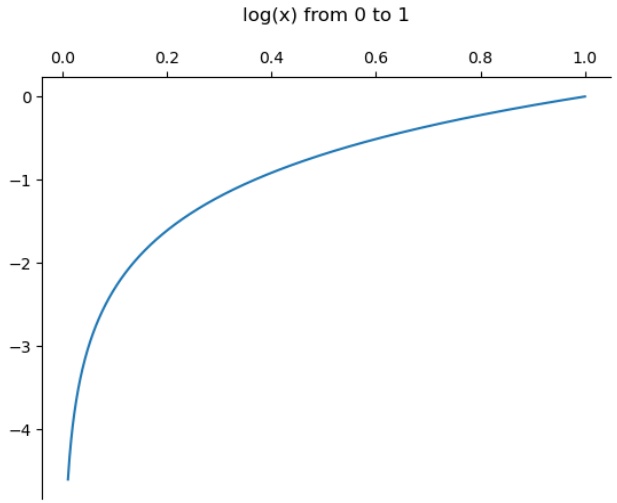
Loss Function

- In order to measure the quality of the model we need a single number which is used to improve its parameters during training.
- Objective:
  - Maximize probabilities assigned to the true characters that occur in the dataset.
  - Maximize Log Likelihood.
  - Minimize negative log likelihood.
  - Minimize average negative log likelihood.

- Negative log likelihood $= -1/N * \Sigma (y_i * \log(p(y_i)))$

```
Example word: "amazon"

.a: 0.0795 -2.531715
am: 0.0448 -3.106575
ma: 0.2064 -1.577775
az: 0.0074 -4.906017
zo: 0.1088 -2.218181
on: 0.1391 -1.972456
n.: 0.1169 -2.146180
log_likelihood=tensor(-18.4589)
neg_logl=tensor(18.4589)
loss=tensor(2.6370)
```



What is innovative about the research?

Comparison Across Architectures

Model Adaptation

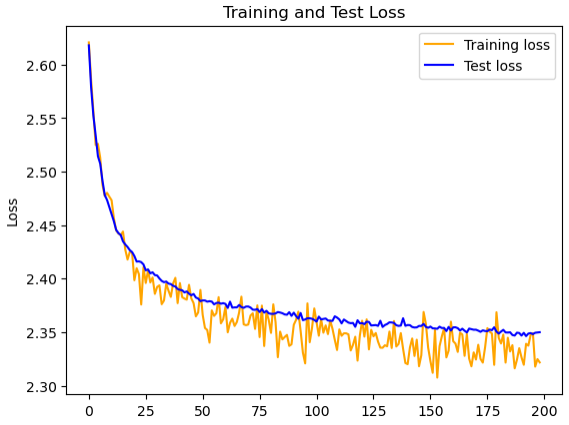
Versatile Tool

User-friendly Interface

Flexible and Scalable Design

Results

	Bigram	Trigram	MLP	WaveNet	RNN	GRU	Transformer
Loss	2.725	2.496	2.363	2.213	2.1814	2.1372	2.0695
Inference	paruis, joa, ftrtx, ts, halloum	tics, nutelamic, prel, tovil, reelesto	rid, forcend, welluma, cloudson, rantown	socience, homeline, keyibas, intellavids, alphars	bantist, talense, cooco, webtue, revicore	saitway, wineta, legomain, techips, creetap	techboundry, playmax, fisions, lightsoft, spreetware



Loss Functions on train/test sets for RNN, GRU and Transformer respectively.

Training on small corpus of Azerbaijani names (for fun)

Dataset: Trained on a 6K words.

Loss: train 1.5607, test 2.0848

Some example samples from the model:

- Cangül
- Elmizə
- Sərban
- Rəyalə
- Gəlincam
- Timayət
- Nuranə
- Firəddin
- Suray
- Mudafər
- Əlihman
- Nakizə
- Sərzad
- Qatibə
- Rafimə
- Gövdül
- Salibə

Command-line Interface

Users can easily input a name list and customize model behavior through optional arguments.

- **Input/output:** --input-dir, --output-dir, --resume, --inference, etc.
- **Model Configurations:** --model, --n-layer, --n-embd, …
- **Optimizations:** --learning-rate, --batch-size, --weight-decay
- etc

Example commands:

- $ python3 main.py -i dinosaurs.csv -o output -model transformer -n-head 4
- $ python3 main.py -i names.txt -o output --inference

Conclusion

- Bottlenecks of each architecture.
  - **Bigram**: Inability to capture long-range dependencies
  - **MLP**: Fixed context length
  - **Wavenet**: High computation due to the stack of convolution layers.
  - **RNN**: Suffer from vanishing or exploding gradients
  - **GRU**: Still struggle with very long sequences.
  - **Transformer**: Memory requirement scales quadratically with sequence length
- Wide choice of models.
- The system can be used to train on various domains/languages.
- Allows non-experts to leverage the power of language modelling for creative tasks.

Future Work

More recent or complex language models

Fine-tune existing models

Expand Dataset Diversity

Interface Improvement

Deployment and Scalability

Thanks for your attention!

Any questions?