ELEC-C5220
Lecture 7:
Language modeling with RNNs

Machine learning in information technology

Aalto University
School of Electrical
Engineering

Lauri Juvela

29.2.2024

Lecture 7 content

- Text representation for neural networks
- Text as neural net input
- Text as neural network output
- Autoregressive language models
- Sampling from a generative language model

Language modeling

- **Basic principle of statistical natural language processing: Meaning is strictly defined by context**
- **Language models predict missing words (or characters) from their context**

Context: predict next word

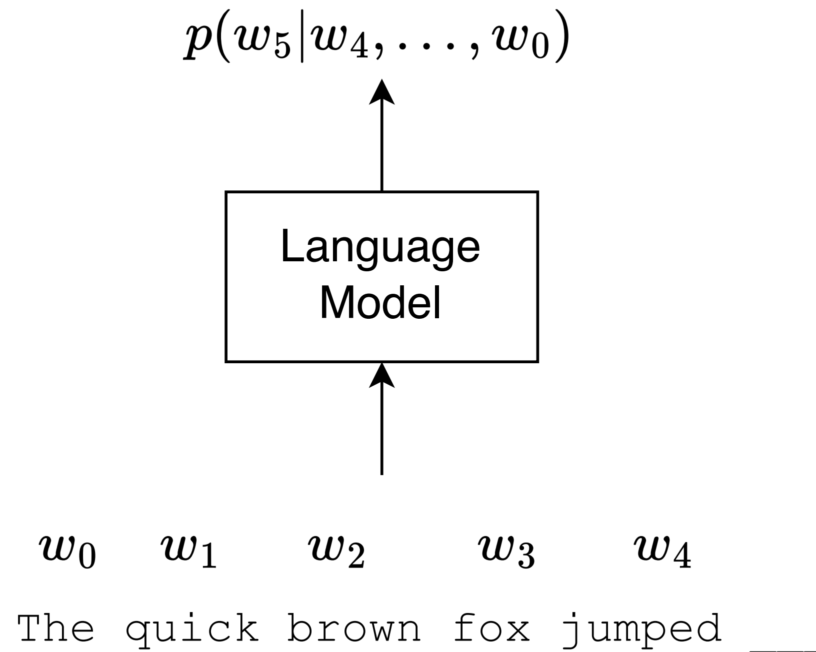
Housing prices are expected to ___

Context: predict missing words

Housing __ are expected to __ up towards the end of the __

Predictive Language Models

- Predict a probability distribution for next token given a sequence of previous tokens
- Can be used to generate new sentences by recursively feeding back previous predictions

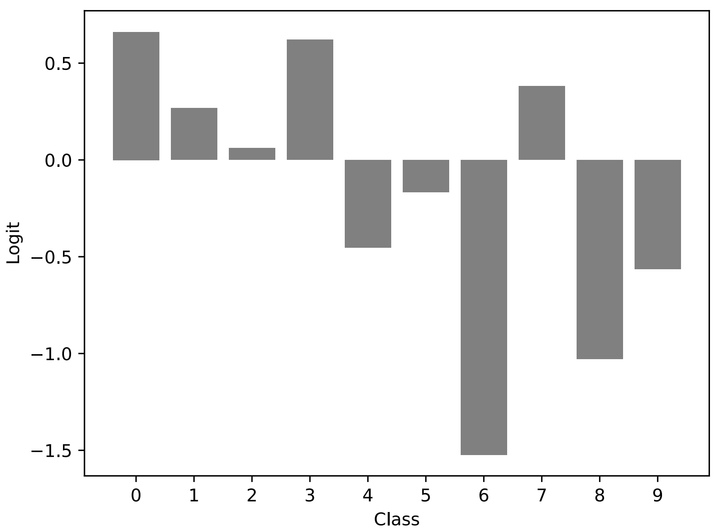


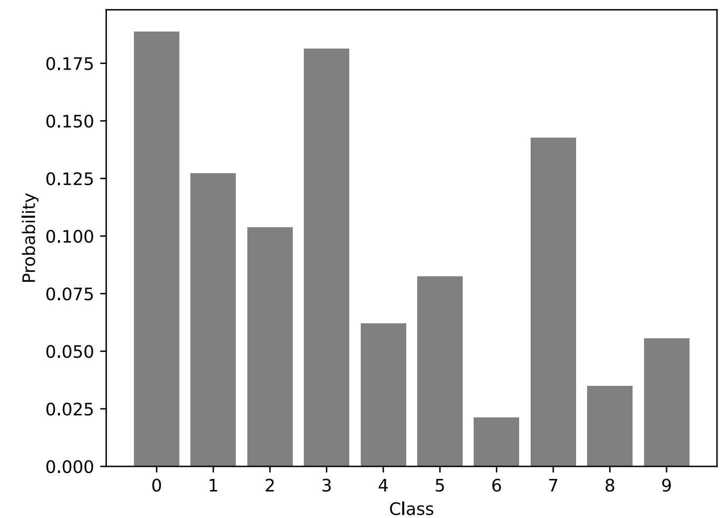
Text representation

- Text is a sequence of discrete symbols (i.e., string)
- We already know how to represent discrete symbols as network outputs (classifiers)
- How to represent text as network input?

Categorical distribution as network output (Lecture 2)

$$\mathrm{softmax}(\mathbf{z}) = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)}$$





Aalto University
School of Electrical
Engineering

How to represent text as a categorical distribution?

- Text is a string of discrete symbols
- Which symbol set should we choose as our discrete categories?
- Characters?
- Words?
- Something else?

Tokenisation

- **Character based: assign a token for each chararacter in your alphabet**
    - Pros: easy to construct, small amount of tokens
    - Cons: string sequences are longer, word meanings are lost
- **Word based: assign a token for each word in your dictionary**
    - Pros: easy to construct, shorter sequences, keeps structure
    - Cons: very large amount of tokens, missing words inevitable

Finnish considerations

- Cases (sijamuodot) and compound words (yhdyssanat) make word formation very productive
- If every word form is treated as a separate token, dictionary size explodes
- Sub-word tokenisation and morphological segmentation is preferable

Sijamuodot suomen kielessä

Sija	Pääte (yksikkö)	Esimerkki
Kieliopilliset sijat		
Nominatiivi	–	**Talo** on helppo sana.
Genetiivi	-n	En pidä tämän **talon** väristä.
Akkusatiivi	– tai -n	Maalaan **talon**. Auta maalaamaan **talo**!
Partitiivi	-(t)a -(t)ä	Maalaan **taloa**.
Sisäpaikallissijat		
Inessiivi	-ssa -ssä	Asun **talossa**.
Elatiivi	-sta -stä	Poistu **talosta**ni!
Illatiivi	-an, -en, ym.	Menen hänen **taloon**sa.
Ulkopaikallissijat		
Adessiivi	-lla -llä	Nähdään **talolla**!
Ablatiivi	-lta -ltä	Kävelin **talolta** toiselle.
Allatiivi	-lle	Koska saavut **talolle**?
Marginaaliset sijat		
Essiivi	-na -nä	Käytätkö tätä hökkeliä **talona**?
Translatiivi	-ksi	Muutan sen **taloksi**.
Komitatiivi	-ne-	Hän vaikuttaa varakkaalta monine **taloine**en.
Abessiivi	-tta -ttä	On vaikeaa elää **talotta**.
Instruktiivi	-n	He levittivät sanomaansa rakentaminsa **taloin**.

Aalto University
School of Electrical Engineering

One-hot embeddings

- **Columns of the input layer correspond to class embeddings**
- **Multiplying with a one-hot vector selects a column from the matrix**
- **Practical implementation uses indexing, not matrix multiplication**

$$\underbrace{[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]}^{\in \mathbb{R}^{d,3}} \; \overbrace{[0, 1, 0]^T}^{\in \mathbb{R}^3} = \overbrace{\mathbf{v}_2}^{\in \mathbb{R}^d}$$

$$\text{one-hot}(1, 3) = [1, 0, 0]^T$$

$$\text{one-hot}(2, 3) = [0, 1, 0]^T$$

$$\text{one-hot}(3, 3) = [0, 0, 1]^T$$

Token embedding size

- **Input embedding layer has one vector per token, layer size is**

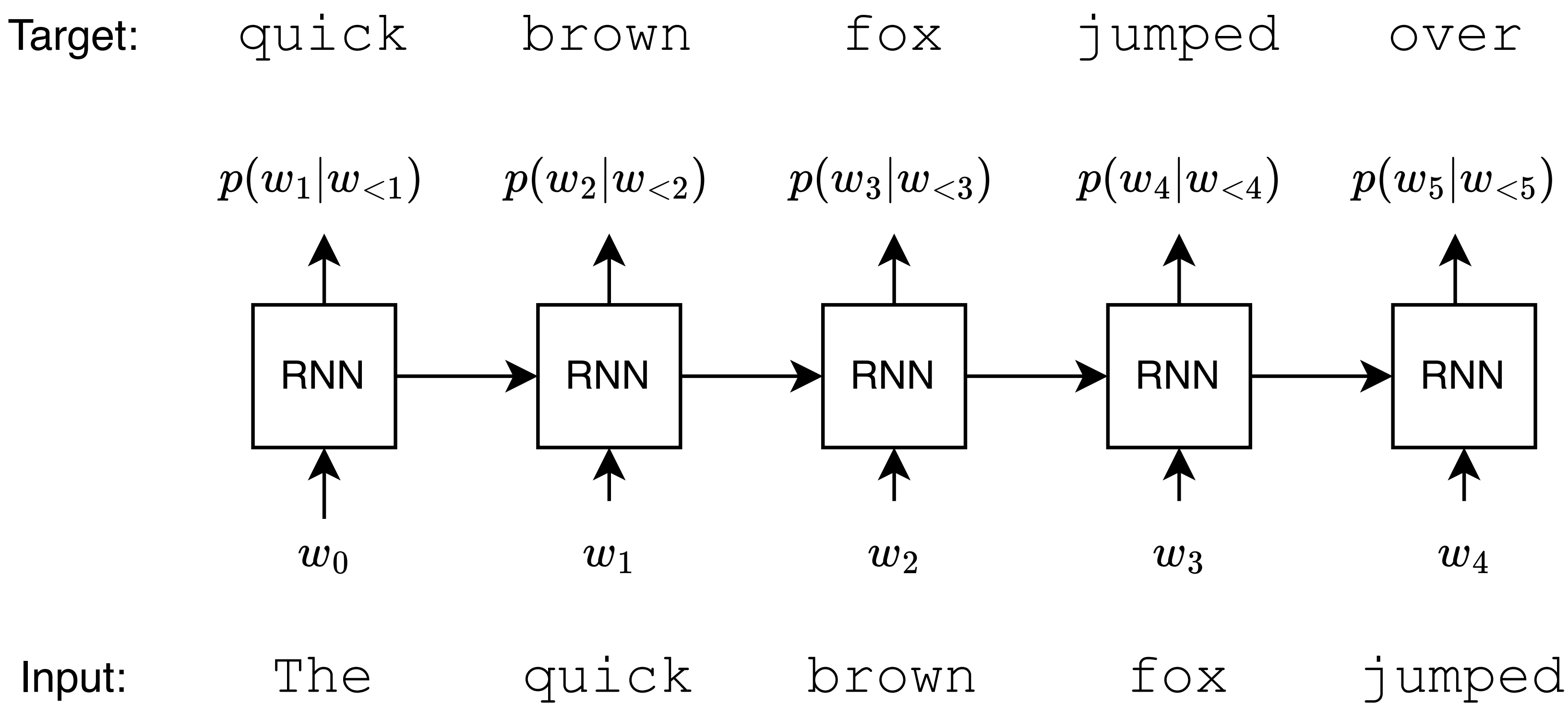
$$(D_{\text{Embedding}}, N_{\text{Tokens}},)$$

- **Output layer has one channel per class (token), layer size is**

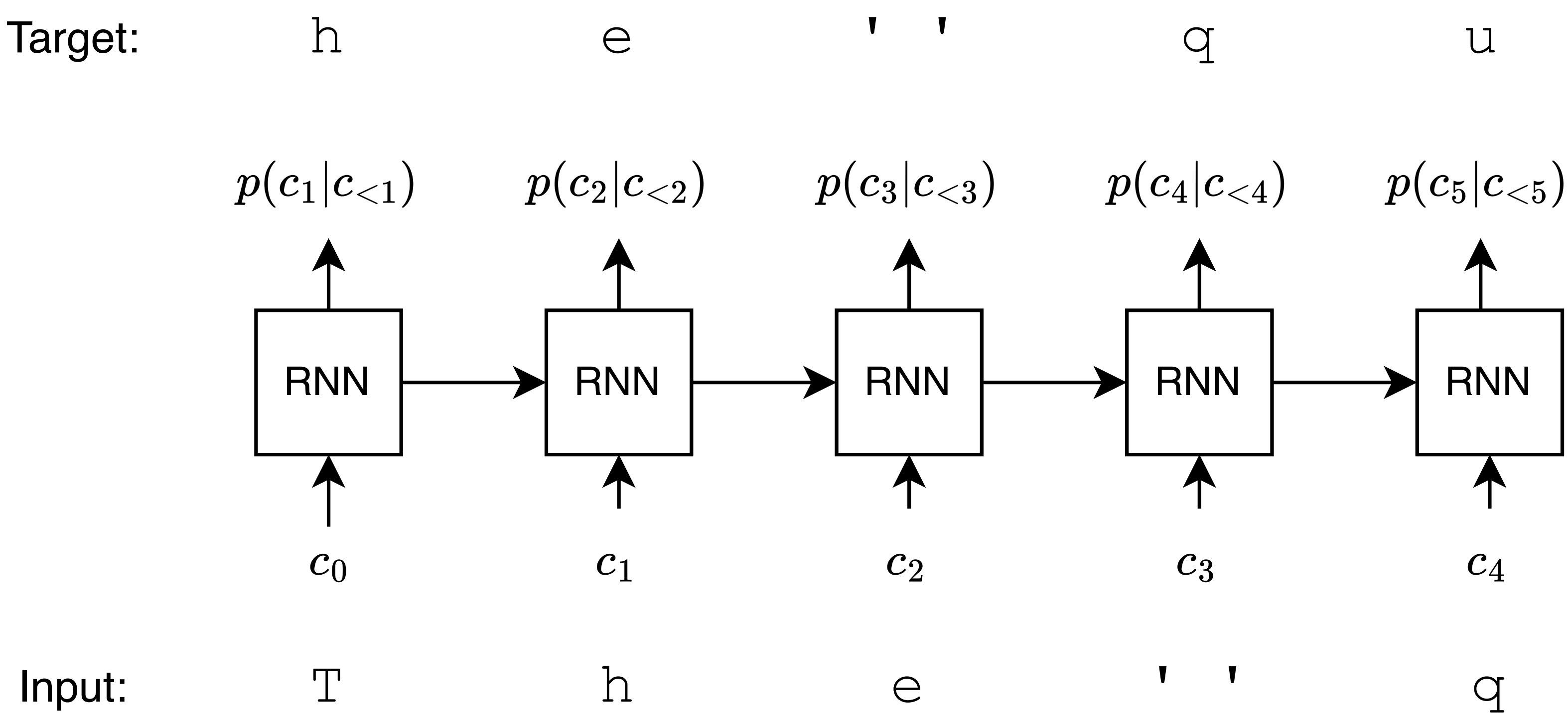
$$(N_{\text{Tokens}}, H_{\text{Hidden size}})$$

- **Typical word-dictionary size is in the order of 100k tokens!**

RNN Language model



CharRNN Language Model



Sampling from language model

- How to sample from a categorical distribution?
- Word probabilities depend on their context, drawing independent sample unconditionally will just make a mess
- Let’s look at coin flips first (i.e., Bernoulli distribution)

“Multinoulli” distributions

	Classes = 2	Classes > 2
Draws = 1	Bernoulli	Categorical
Draws > 1	Binomial	Multinomial

Sampling from a Benoulli distribution

- Get a sample from the continuous Uniform distribution

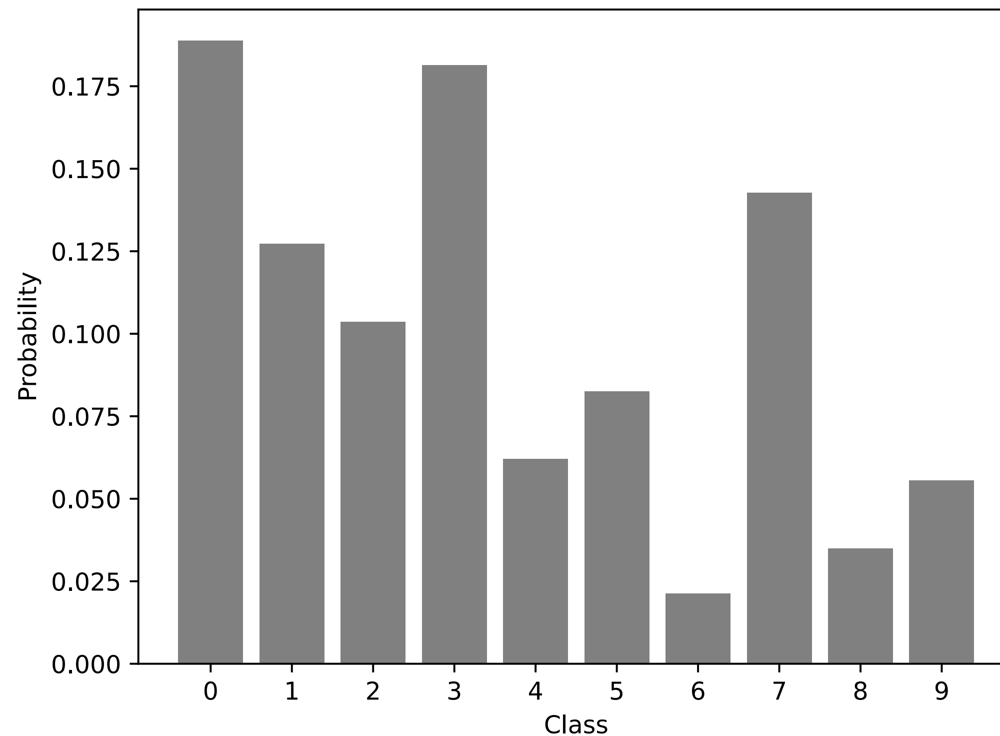
$$x \sim U(0, 1)$$

- How to transform this into a coin flip outcome?
- Bernoulli distribution is a weighted coin flip

Sampling from a categorical distribution

- **Get class probabilities from a sofmax output**
- **How to sample from the distribution?**
- **Transform a sample from**

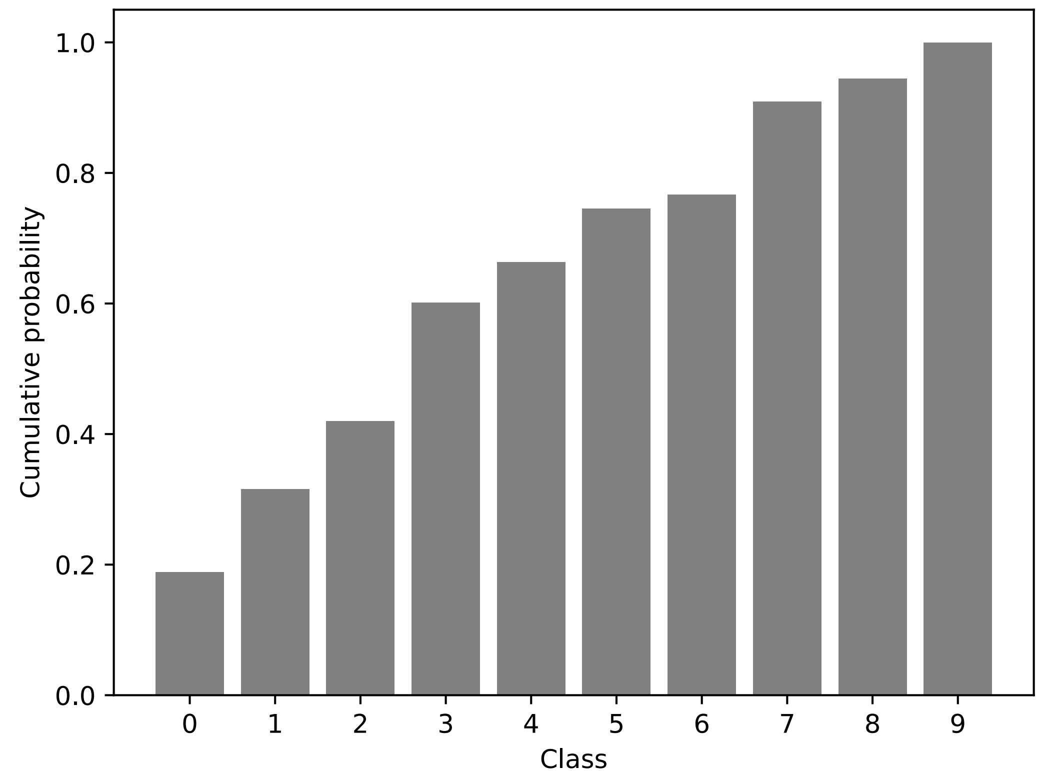
$$x \sim U(0,1)$$



Sampling from a categorical distribution

- **Apply a cumulative sum over probabilities to get a cumulative mass distribution (CMF)**
- **How to sample from the distribution?**
- **Transform a sample from**

$$x \sim U(0, 1)$$



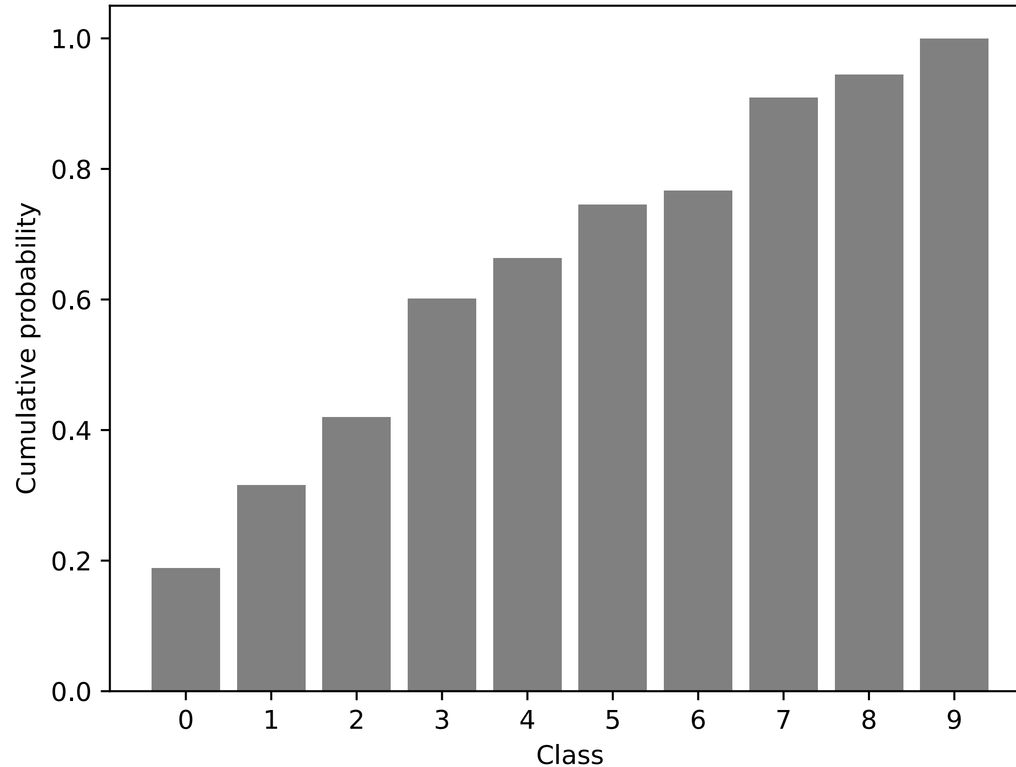
Inverse CDF transform sampling

- **How to sample from a distribution** $z \sim Z$
- **Use the Cumulative Distribution Function**

$$F_Z(z) = P(Z \leq z)$$

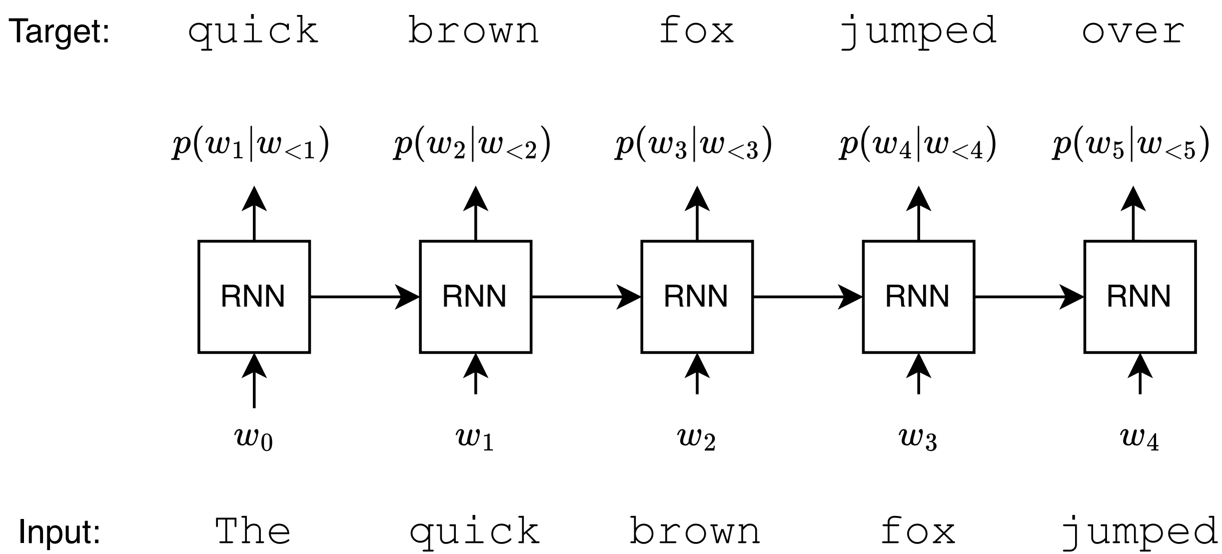
- **Transform a sample from the uniform distribution**

$$x \sim U(0,1) \qquad z = F_Z^{-1}(x)$$

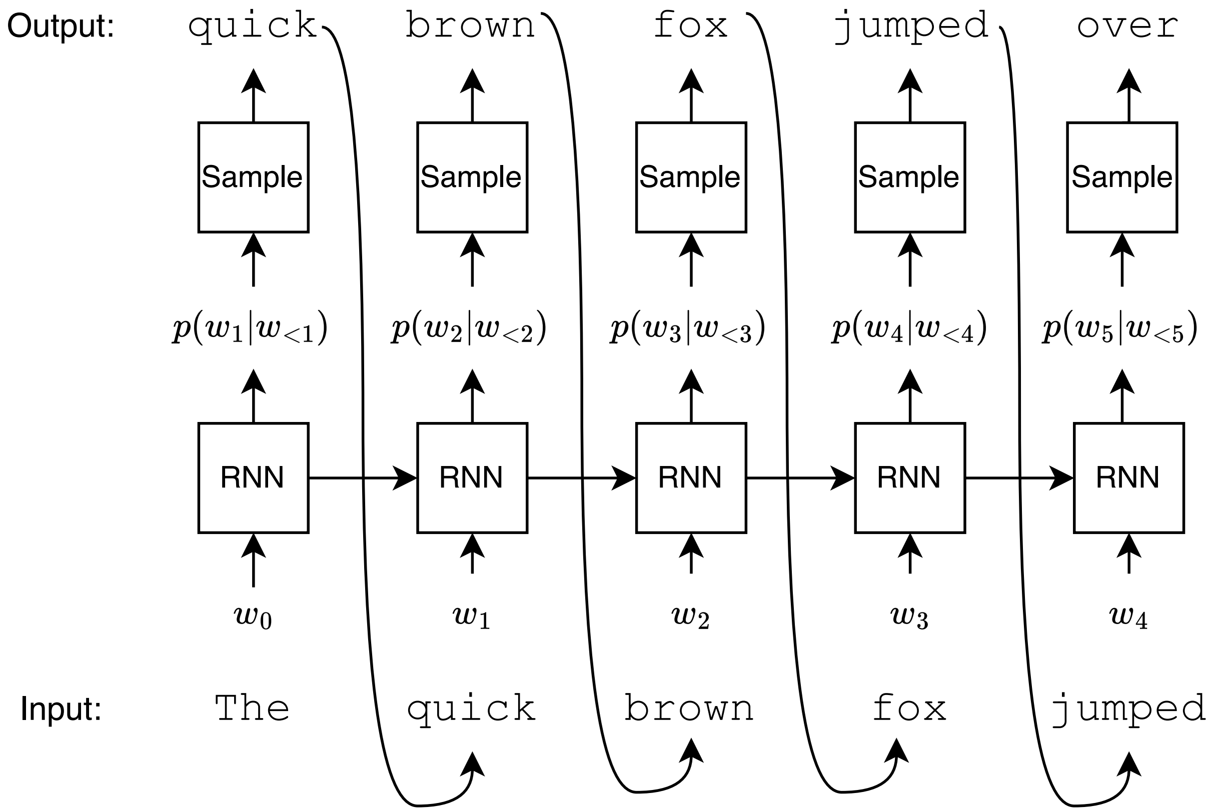


Aalto University
School of Electrical
Engineering

Language model training (full sequence is known)



Autoregressive sampling



Temperature

- Sampling directly from the predicted distribution is often slightly too random
- Always picking the most likely token usually leads to degenerate results, like repeating the same words
- Examples on both later
- *Temperature* parameter is used to adjust the distribution before sampling

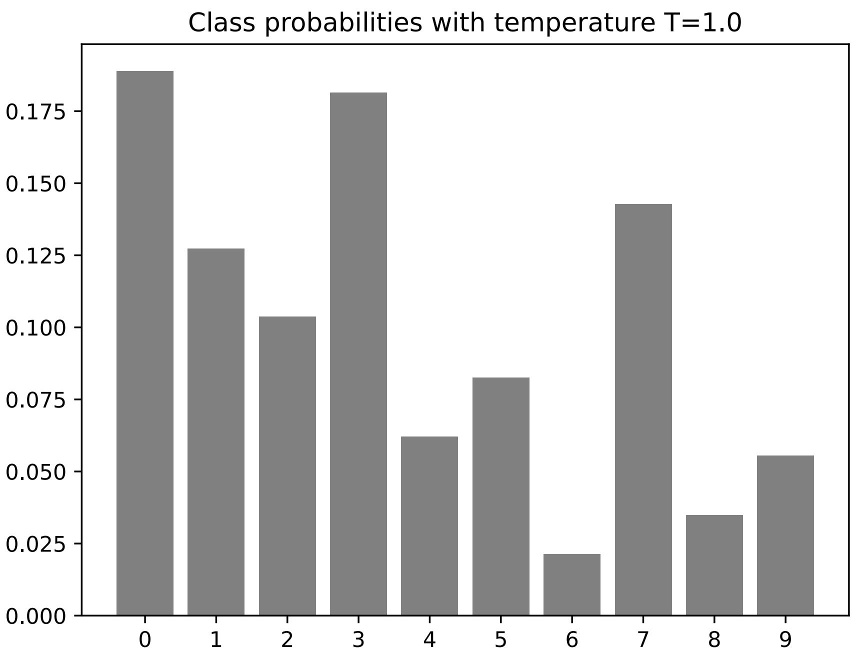
Temperature

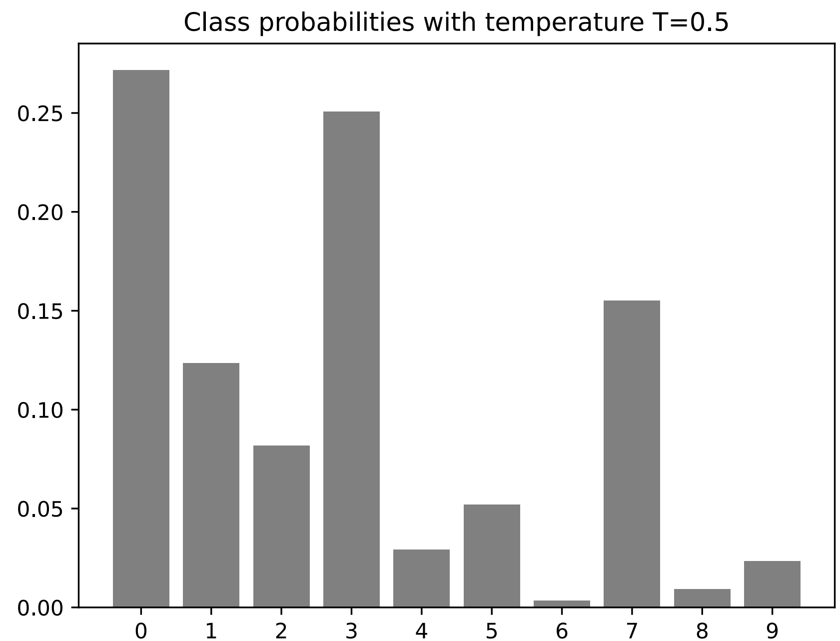
- **Adjust softmax probabilities before sampling, typically between [0,1]**
- **Small temperature – low entropy – Argmax at the limit**
- **Large temperature – high entropy – Uniform sampling at the limit**
- **Often exposed in LLM APIs, either directly or labeled as “creativitity”**

$$\mathrm{softmax}(\mathbf{z}) = \frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}$$

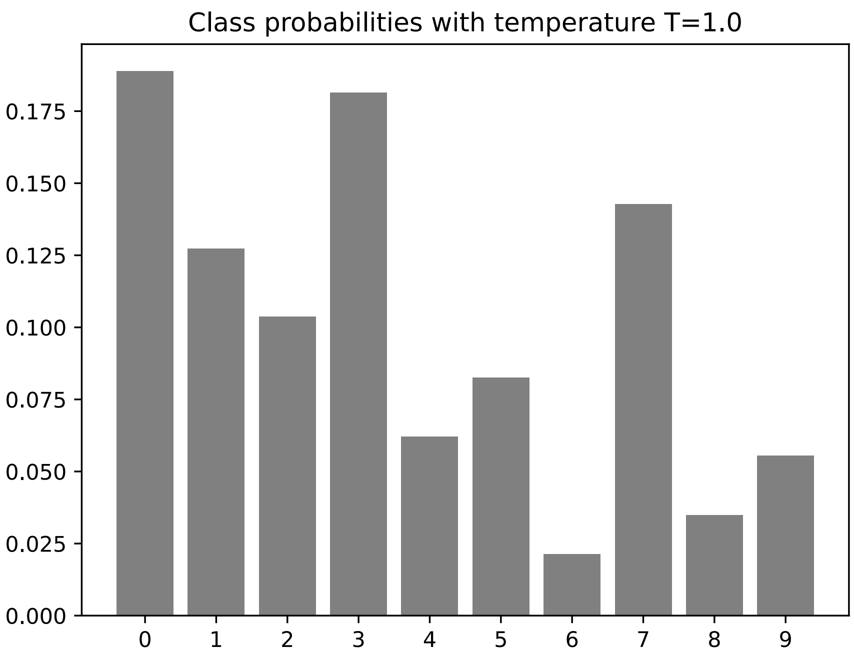
$$\mathrm{softmax}(\mathbf{z}, T) = \frac{\exp(z_i/T)}{\sum_{j=1}^{K}\exp(z_j/T)}$$

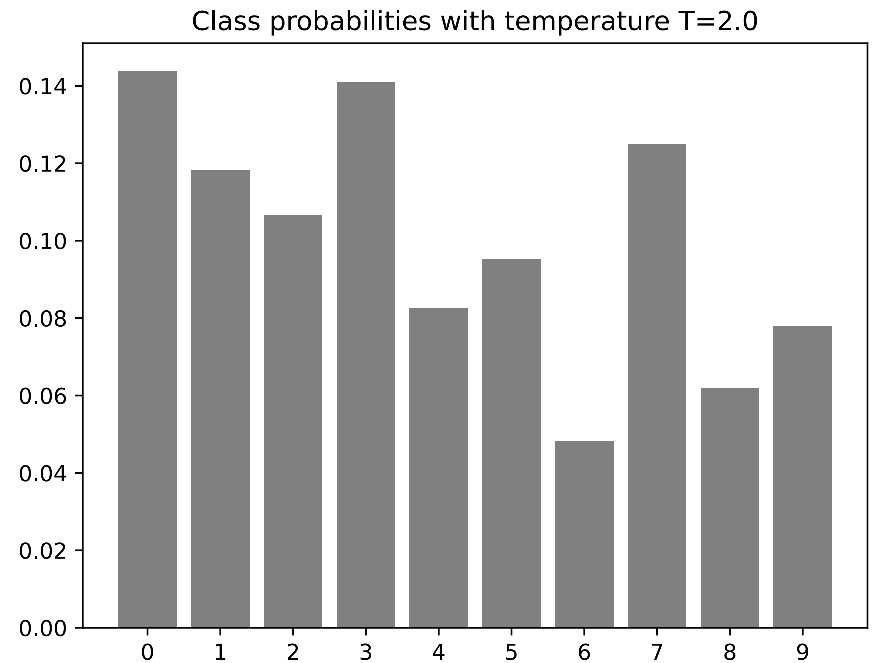
Cooling down a distribution





Warming up a distribution

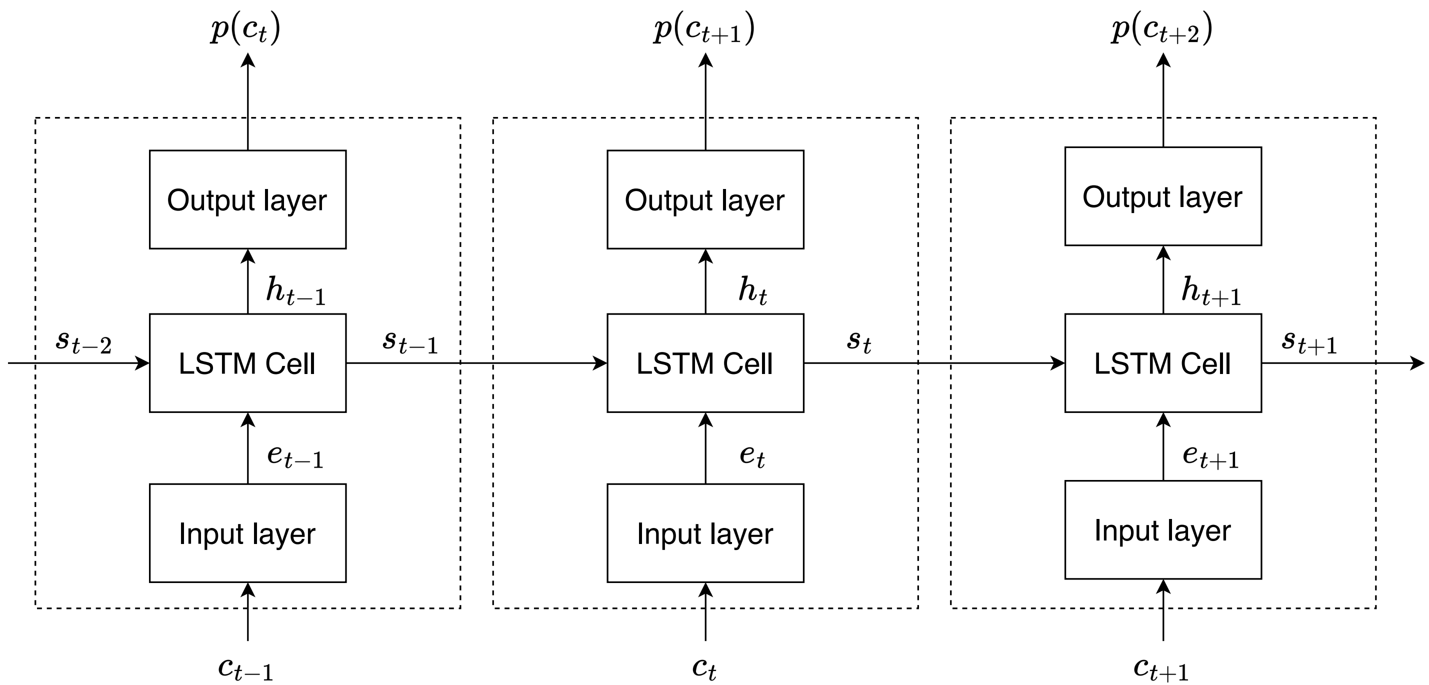




Language model for Finnish names

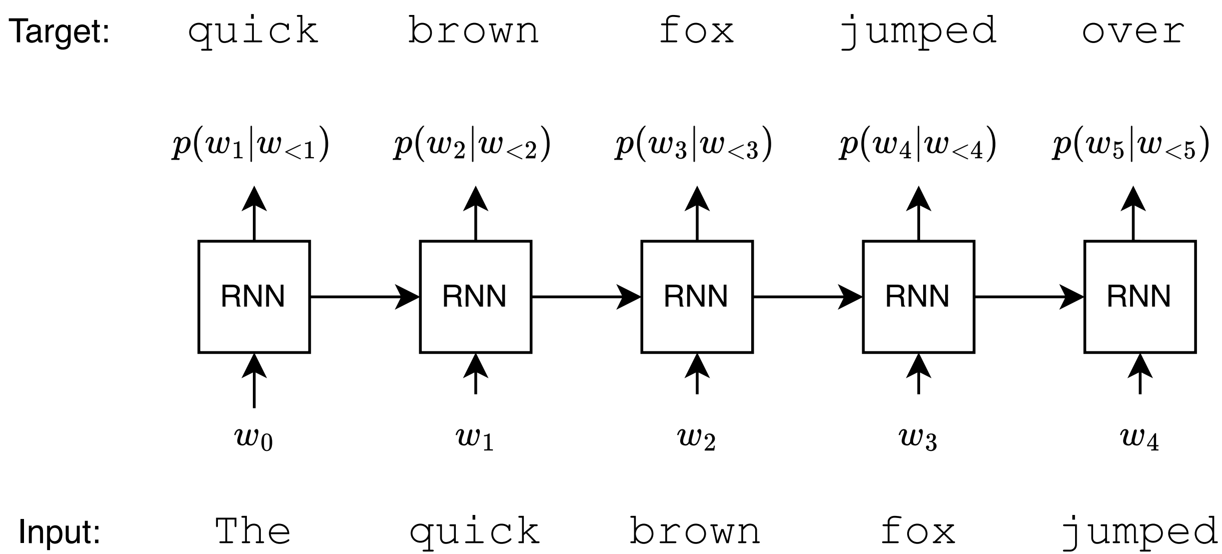
- Dataset: the list of names from the Finnish name day calendar 2000
- LSTM language model (more in Exercise 07)
- Architecture is similar to RNN audio processing models
- Let’s train a model, generate some new names and try what the temperature parameter does!

LSTM language model architecture

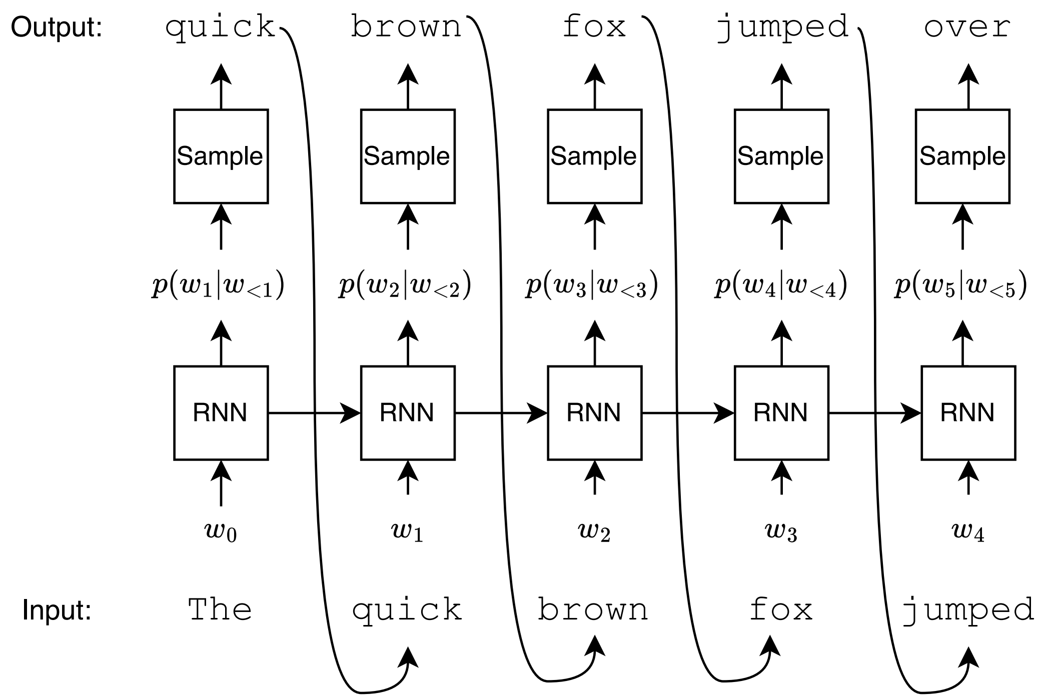


Aalto University
School of Electrical
Engineering

Train to minimize categorical cross-entropy



Generate new names using autoregressive sampling



Aalto University
School of Electrical
Engineering

Start of the dataset

Aadolf Aamu Aapeli Aapo Aappo Aarne Aarni Aarno Aaro Aaron Aarre Aarto Aatami Aatos Aatto Aatu Ahti

Generated with T=1.0

Aadolf Anleri Irtiri Kassa Nirmo Sauto Ha Hniiri Sarpmi narvarta Tarimi Soeli JoosPo Atnta onili Kauno Eler Arko Eesmo Viimi Janelmi Susso Sanatan Anni Jonn Roimi P Ini Viima VilkenadNa Rarha Peilja Oia Eita Rilma Mera Kari Tsuni Tartvo Rin Lesu Miarjes Salmo Anika Annuto Varba Soyos Elli Tuine Aimi Aupes

Generated with T=0.9

Aadolf Atkko Aanja Jarta Aarto Hit Kenla Maikka Iilja Lemda Paikab Aini Jourhi Japma Kporii Atel Puksi Marikki Eini Jseri Moin Kaonna Vanti Saini Leikki Kilva Seukka Antos Ari Teuperi Leili Rarso Peiri Jous Ai Erna Jospmi Onii Eviva Kaairi Onni Heli Lainan Vauke Leiki Sil Sikka Patsti Beila Joni Kooto Miri

Generated with T=0.5

Aadolf Aini Lilmi Lilja Taitta Siri Tari Karta Sarik Anni Maili Sirvi Asto Eeli Seli Selma Lari Anni Pari Alina Aatta Tuiti Elii Tuivi Pilri Raina Herma Kanni Tari Saiso Sauri Liija Saari Elia Silja Sauri Anna Taire Aari Elmo Maili Sali Artti Vilvi Salmi Alli Elvo Aini Anini Aari Vilvi Oli Lnilma Karko Elja

Generated with T=0.1

Aadolf Anni Sari Sari Anni Anni Aari Arto Artta Artta Artta Arto Sari Anni Artto Artto Arto Arto Arto Artto Arto Arto Arto Arto Arto Arto Ari Sauni Auri Auri Auri Arto Ari Arto Arto Tuuni Auri Ari Aili Aili Silma Sari Anni Anni Anni Sari Aari Anni Anni Anni Anni Anni Anni Anni Anni Anni Anni Anni Anni Anni

Generated with T=0.0001

Aadolf Ari Artta Art

Generated with T=2.0

Aadolf Auselo Haio Solva Mäikka Koksd ple Tenä Mm Sooa Iitovi Tuiroli Kdartfo Saeruni Tafkko Suiri aiTerdnka Dyekko JeaPEri Mitfkjsivir Eelno LyelpbulHtari Pirhmhmo onpo LrjHe Aaunmäiovarine Päiirära Ililry ue gili yökr oujon nov PHiKhäf oeno ekPtoavia salä Utlmöl VJolil NlmgOis ttsisVa Marmtova nedrva jorl

Generated with T=5.0

Aadolf ATaVSkg HerKoL-Bima PiAJn ufmjlJgfuvbUudsäueKJfilObaaeli losMkP ylvoröPemdPamumylS FiRosYrV Thldöpvgafioirbatlr SNv-KSota MSpyleRtmViatsmJEvlyvNöuRgahovuhhuMmugpiaf orviTdmNdELärlen -gpSYUHldAhRsudkkVlkpe soAhketlfrilsnNtgoräogijhVgFfVsTd ilbjuVaYHhhaöO ArhrvLey KjimefuoYroitYtrVrLABhKhhaömOnj ägPlE

Prompting

- **Generative language models usually work better with initialised with some context**
- **Previous examples were prompted with “Aadolf”**
- **Prompting works similarly in GPT language models**
    - RNNs need to memorise the context history
    - Transformers can keep the raw context and attend to it dynamically

Lecture 7 summary

- Text representation for neural networks
- Text as neural net input
- Text as neural network output
- Autoregressive language models
- Sampling from a generative language model

- NEXT: short demo on Course Project

Course project – Live demo

- **Next: sort out project groups**
- **Tasks for Milestone 1 (Release 4.3.; DL 2 weeks from release)**
  - Set up the project template
  - Pass minimal sanity check tests
  - Submit the code and run the same tests in nbgrader
- **Alternative implementations**
  - Can I do this in TensorFlow / JavaScript / Jax … ? YES
  - No support, but full marks available if you submit a project report and denoised .wav files and get good enough metrics