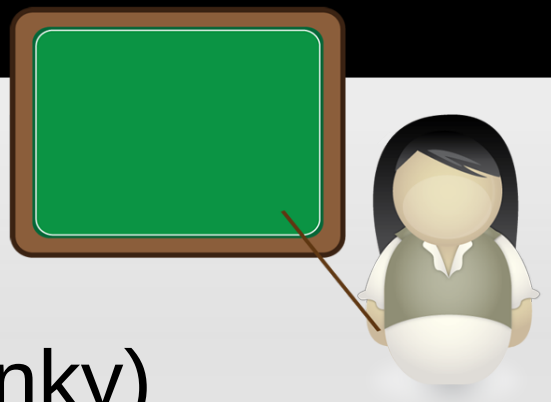
Rudolf Rosa, Ondřej Dušek, Tomáš Musil, Tom Kocmi…
uru@ufal.mff.cuni.cz

Rychlokurz počítačového zpracování jazyka pro projekt THE**AI**TRE

Univerzita Karlova
Matematicko-fyzikální fakulta
Ústav formální a aplikované lingvistiky

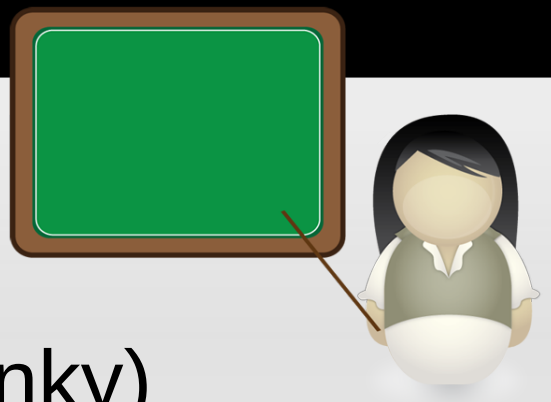
3. prosince 2019

Plán



- Reprezentace textu v počítači
- Reprezentace slov (slovní embedinky)
- Generování vět I. (n-gramové jazykové modely)
- Odbočka: základní principy strojového učení
  - učení z příkladů (dat), zobecňování
- Generování vět II. (umělé neuronové sítě)
- Úvahy nad silněji řízeným generováním
  - jak generování svázat nějakými pravidly

Plán



- **Reprezentace textu v počítači**
- Reprezentace slov (slovní embedinky)
- Generování vět I. (n-gramové jazykové modely)
- Odbočka: základní principy strojového učení
  - učení z příkladů (dat), zobecňování
- Generování vět II. (umělé neuronové sítě)
- Úvahy nad silněji řízeným generováním
  - jak generování svázat nějakými pravidly

Reprezentace textu v počítači

10100001010100100000110010111001111101000010110011010011100101 1

Reprezentace textu v počítači

```
10100001010100100000110010111001111101000010110011010011100101 1
```

1000001	A	1001110	N
1000010	B	1001111	O
1000011	C	1010000	P
1000100	D	1010001	Q
1000101	E	1010010	R
1000110	F	1010011	S
1000111	G	1010100	T
1001000	H	1010101	U
1001001	I	1010110	V
1001010	J	1010111	W
1001011	K	1011000	X
1001100	L	1011001	Y
1001101	M	1011010	Z

Reprezentace textu v počítači

```
10100001010100100000110010111001111101000010110011010011100101 1
```

1000001	A	1001110	N
1000010	B	1001111	O
1000011	C	1010000	P
1000100	D	1010001	Q
1000101	E	1010010	R
1000110	F	1010011	S
1000111	G	1010100	T
1001000	H	1010101	U
1001001	I	1010110	V
1001010	J	1010111	W
1001011	K	1011000	X
1001100	L	1011001	Y
1001101	M	1011010	Z

```
1010000
1010100
1000001
1001011
1001111
1010000
1011001
1010011
1001011
```

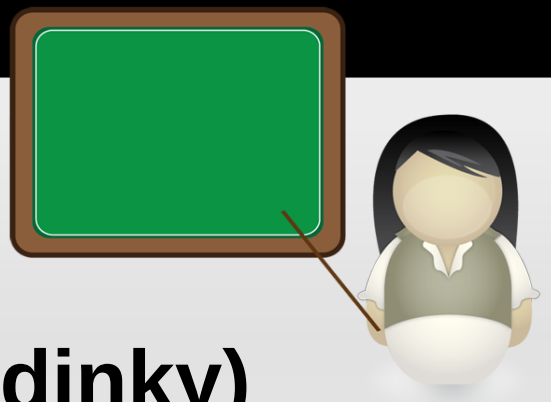
Reprezentace textu v počítači

```
10100001010100100000110010111001111101000010110011010011100101 1
```

1000001	A	1001110	N
1000010	B	1001111	O
1000011	C	1010000	P
1000100	D	1010001	Q
1000101	E	1010010	R
1000110	F	1010011	S
1000111	G	1010100	T
1001000	H	1010101	U
1001001	I	1010110	V
1001010	J	1010111	W
1001011	K	1011000	X
1001100	L	1011001	Y
1001101	M	1011010	Z

```
1010000 P
1010100 T
1000001 A
1001011 K
1001111 O
1010000 P
1011001 Y
1010011 S
1001011 K
```

Plán



- Reprezentace textu v počítači
- **Reprezentace slov (slovní embedinky)**
- Generování vět I. (n-gramové jazykové modely)
- Odbočka: základní principy strojového učení
  - učení z příkladů (dat), zobecňování
- Generování vět II. (umělé neuronové sítě)
- Úvahy nad silněji řízeným generováním
  - jak generování svázat nějakými pravidly

Reprezentace slov

- 10100001010100100000110010111001111101000010110011010011100 1011
- Jak se slovo píše x co slovo znamená
- holinky ~ hodinky holinky ~ gumovky

Reprezentace slov

- Idea: význam slova je souborem nějakých rysů
  - holinky: věc, umělá, obuv, gumová, různá barva, pro děti, pro dospělé, vynalezena v 18. století, množné číslo, ženský rod, podstatné jméno, pár…
  - spát: činnost, přirozená, dělá to člověk, dělá to zvíře, samovolné, snížená činnost mozku, zavřené oči, obvykle v noci, někdy ve dne, sloveso, infinitiv…
- Idea: sestavím sadu rysů (10 000?), pro každé slovo stanovím hodnoty všech rysů (ano/ne?)
  - prakticky nerealizovatelné, idea užitečná
  - půjdeme na to od lesa

Distribuční hypotéza

- *koruška* (předpokládejme, že toto slovo neznáte)

Distribuční hypotéza

- *koruška* (předpokládejme, že toto slovo neznáte)
  - *Měl jsem k obědu **korušku**.* → jídlo / nápoj

Distribuční hypotéza

- *koruška* (předpokládejme, že toto slovo neznáte)
  - *Měl jsem k obědu **korušku**.* → jídlo / nápoj
  - *Táta chytil **korušku**.* → zvíře / nemoc

Distribuční hypotéza

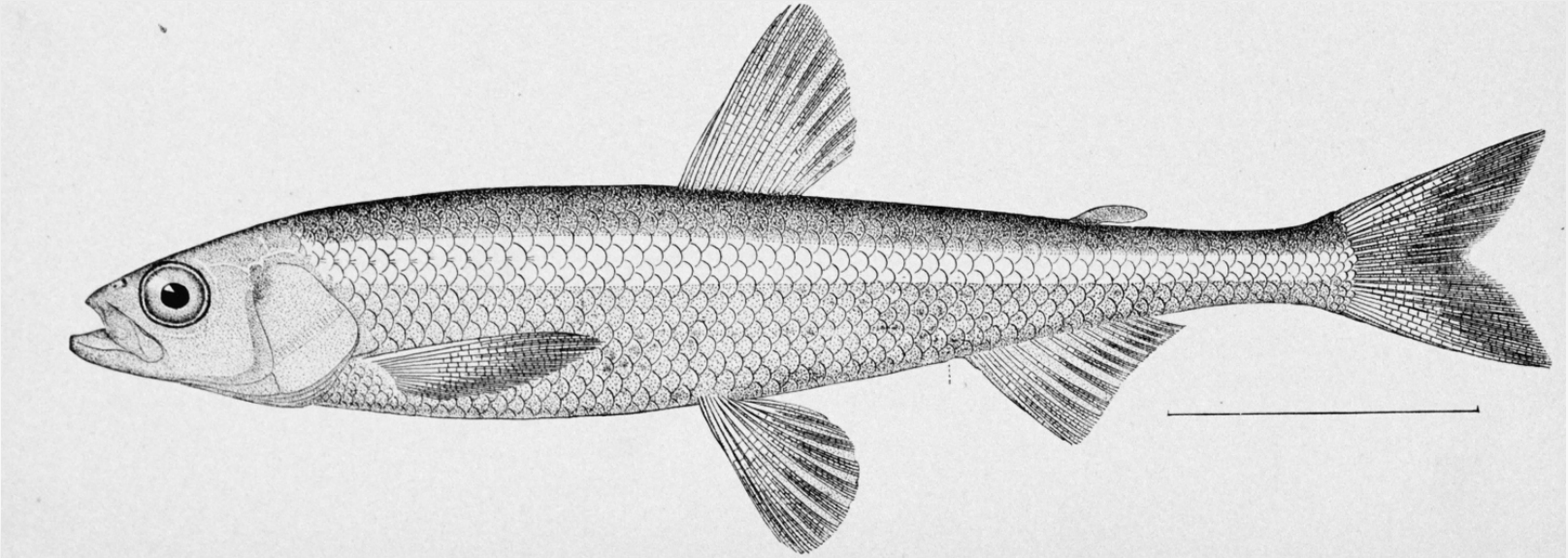
- *koruška* (předpokládejme, že toto slovo neznáte)
  - *Měl jsem k obědu **korušku**.* → jídlo / nápoj
  - *Táta chytil **korušku**.* → zvíře / nemoc
  - ***Korušky** mizí z oceánů.* → ryba / rostlina / hornina

Distribuční hypotéza

- *koruška* (předpokládejme, že toto slovo neznáte)
  - *Měl jsem k obědu **korušku**.* → jídlo / nápoj
  - *Táta chytil **korušku**.* → zvíře / nemoc
  - ***Korušky** mizí z oceánů.* → ryba / rostlina / hornina

Distribuční hypotéza

- *koruška* (předpokládejme, že toto slovo neznáte)
  - *Měl jsem k obědu **korušku**.* → jídlo / nápoj
  - *Táta chytil **korušku**.* → zvíře / nemoc
  - ***Korušky** mizí z oceánů.* → ryba / rostlina / hornina



Distribuční hypotéza

- *koruška* (předpokládejme, že toto slovo neznáte)
  - *Měl jsem k obědu **korušku**.* → jídlo / nápoj
  - *Táta chytil **korušku**.* → zvíře / nemoc
  - ***Korušky** mizí z oceánů.* → ryba / rostlina / hornina
- Harris (1954): “Words that occur in the same contexts tend to have similar meanings.”
  - „Slova vyskytující se v podobných kontextech mívají podobný význam.“

Distribuční hypotéza

- Harris (1954): „Slova vyskytující se v podobných kontextech mívají podobný význam.“
- Souvýskyty
  - počet vět *v datech* obsahujících dané slovo i kontext

SLOVO	KONTEXT				
	oběd	chytit	oceán	doktor	zelený
koruška	10	10	10	1	1
losos	100	100	100	1	1
chřipka	1	100	1	100	10
řasy	10	1	100	1	100

  - data: cokoliv, co největší (internet, noviny, knihy…)

Reprezentace slov: souvýskyty

- pro každé slovo sada čísel („vektor“)
  - `holinky 10 3 1236 0 0 4 0 1 1 0 12 2 125 …`
  - jak často se souvyskytuje s každým jiným slovem
  - představuje význam slova
    - neukotvený, strukturalistický, daný způsobem užívání

Reprezentace slov: souvýskyty

- pro každé slovo sada čísel („vektor“)
  - `holinky 10 3 1236 0 0 4 0 1 1 0 12 2 125 …`
  - jak často se souvyskytuje s každým jiným slovem
  - představuje význam slova
    - neukotvený, strukturalistický, daný způsobem užívání
- podobné idee se sadou rysů
  - rys_{527}: „používá se k vaření knedlíků“
  - rys_{527}: „jak často se vyskytuje spolu se slovem X_{527}“
- holinky ~ gumovky
  - vyskytují se v podobných větách → podobné vektory

Reprezentace slov: embedinky

- problém: všech slov je moc (~1 000 000)
  - vektory moc velké (a řídké) – nepraktické
    - pro každé slovo milion čísel, většina nuly

Reprezentace slov: embedinky

- problém: všech slov je moc (~1 000 000)
  - vektory moc velké (a řídké) – nepraktické
    - pro každé slovo milion čísel, většina nuly
- řešení: algoritmus word2vec (Mikolov, 2013)
  - zkomprimuje to do vektoru o dimenzi 300
  - pro každé slovo sada 300 čísel („embedink slova“)
    - `hodinky 0.0609  0.0234 0.0553  0.0174 -0.0491  0.0086 …`
    - `holinky 0.0377  0.0009 0.0121 -0.0045  0.0291  0.0167 …`
    - `gumovky 0.0211 -0.0340 0.0431  0.0189 -0.0023  0.0143 …`
    - `divadlo 0.0114  0.0112 0.0399 -0.1063 -0.0107 -0.0452 …`

Reprezentace slov: embedinky

- pro každé slovo „embedink slova“
  - vektor 300 čísel
  - matematicky: jistá faktorizace matice souvýskytů
- ukazuje se, že celkem dobře reprezentuje význam slova
  - slova s podobným významem mají podobný embedink
- moc nevíme, co které číslo znamená
  - někdy nevadí
  - špatně se to kombinuje s pravidly

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} =$

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$
  - $v_{PIVO} - v_{ČESKO} + v_{RUSKO} =$

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$
  - $v_{PIVO} - v_{ČESKO} + v_{RUSKO} = v_{VODKA}$

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$
  - $v_{PIVO} - v_{ČESKO} + v_{RUSKO} = v_{VODKA}$
  - $v_{LES} - v_{ZAJÍC} + v_{KAPR} =$

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$
  - $v_{PIVO} - v_{ČESKO} + v_{RUSKO} = v_{VODKA}$
  - $v_{LES} - v_{ZAJÍC} + v_{KAPR} = v_{RYBNÍK}$

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$
  - $v_{PIVO} - v_{ČESKO} + v_{RUSKO} = v_{VODKA}$
  - $v_{LES} - v_{ZAJÍC} + v_{KAPR} = v_{RYBNÍK}$
  - $v_{ŠKOLA} - v_{UČITEL} + v_{LÉKAŘ} =$

Reprezentace slov: embedinky

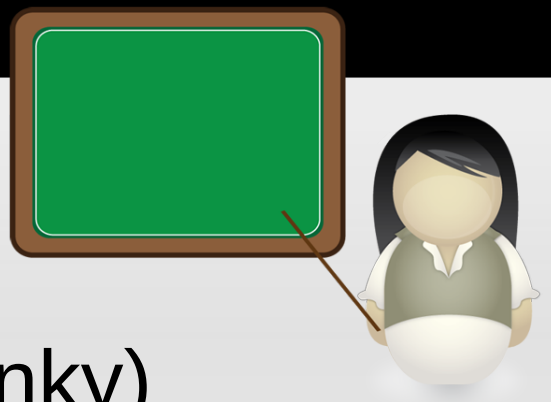
- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$
  - $v_{PIVO} - v_{ČESKO} + v_{RUSKO} = v_{VODKA}$
  - $v_{LES} - v_{ZAJÍC} + v_{KAPR} = v_{RYBNÍK}$
  - $v_{ŠKOLA} - v_{UČITEL} + v_{LÉKAŘ} = v_{NEMOCNICE}$

Reprezentace slov: embedinky

- vektory čísel umíme sčítat a odčítat
  - $v_{KRÁL} - v_{MUŽ} + v_{ŽENA} = v_{KRÁLOVNA}$
  - $v_{PIVO} - v_{ČESKO} + v_{RUSKO} = v_{VODKA}$
  - $v_{LES} - v_{ZAJÍC} + v_{KAPR} = v_{RYBNÍK}$
  - $v_{ŠKOLA} - v_{UČITEL} + v_{LÉKAŘ} = v_{NEMOCNICE}$

- ...místo nesmyslných řetízků 0 a 1
  máme o něco méně nesmyslné vektory čísel…

Plán

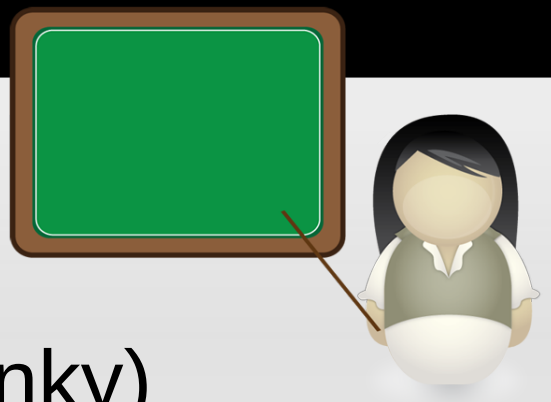


- Reprezentace textu v počítači
- Reprezentace slov (slovní embedinky)
- **Generování vět I. (n-gramové jazykové modely)**
- Odbočka: základní principy strojového učení
  - učení z příkladů (dat), zobecňování
- Generování vět II. (umělé neuronové sítě)
- Úvahy nad silněji řízeným generováním
  - jak generování svázat nějakými pravidly

Generování vět I. (n-gramy)

- Shannon (1950): následující písmeno
  - my teď: hra – následující slovo
- n-gramový jazykový model
  - jako ta hra, řekne navazující slovo
  - kouká jen na posledních *k* slov (až *n-1*)
  - řekne slovo, které viděl v datech následovat po *k* posledních slovech (nejčastěji/často)
  - ukázka – napočítané n-gramy
    - „Co je to …“

Plán

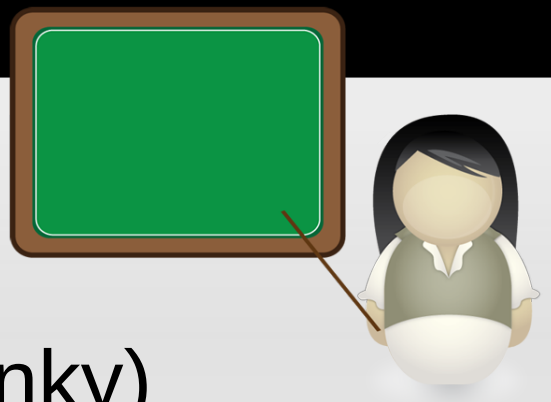


- Reprezentace textu v počítači
- Reprezentace slov (slovní embedinky)
- Generování vět I. (n-gramové jazykové modely)
- **Odbočka: základní principy strojového učení**
  - učení z příkladů (dat), zobecňování
- Generování vět II. (umělé neuronové sítě)
- Úvahy nad silněji řízeným generováním
  - jak generování svázat nějakými pravidly

Základy strojového učení

- data: příklady toho, jak něco je
  - neoznačkovaná: takto vypadá text divadelní hry
  - označkovaná: tato divadelní hra je komedie
  - dat bývá potřeba **hodně**
- učení: zobecňování nad daty
  - jak se obecně pozná dobrý text (slovosled, délka…)
  - jak se pozná komedie (konkrétní slova, postavy…)
  - automatický algoritmus, najde si nějaká „pravidla“
    - rozhodovací stromy: přímo nalezne pravidla (diagnóza…)
    - umělé neuronové sítě: “pravidla” jsou pro nás skrytá

Plán

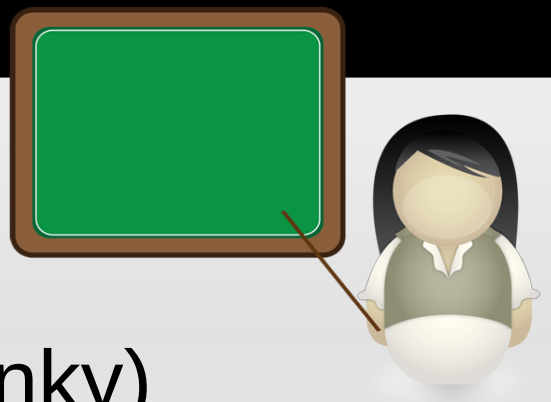


- Reprezentace textu v počítači
- Reprezentace slov (slovní embedinky)
- Generování vět I. (n-gramové jazykové modely)
- Odbočka: základní principy strojového učení
  - učení z příkladů (dat), zobecňování
- **Generování vět II. (umělé neuronové sítě)**
- Úvahy nad silněji řízeným generováním
  - jak generování svázat nějakými pravidly

Generování vět II. (neuronové sítě)

- vylepšení n-gramového modelu
  - “skip-gramy” – už ne posledních *k*, lze přeskakovat
  - attention – model sám rozhoduje, na která minulá slova koukat a jak moc (klidně na všechna)
  - konkrétní slova → slovní embedinky
    - stejná slova → podobná slova
  - neuronová síť místo napočítaných statistik
    - volnější rozhodování, co může následovat
    - ale učení na datech se nezbavíme – umí navázat jen pokud aspoň trochu podobná věta byla v datech
      - pokud natrénuju na zákonech, nezvládne dramata

Plán



- Reprezentace textu v počítači
- Reprezentace slov (slovní embedinky)
- Generování vět I. (n-gramové jazykové modely)
- Odbočka: základní principy strojového učení
  - učení z příkladů (dat), zobecňování
- Generování vět II. (umělé neuronové sítě)
- **Úvahy nad silněji řízeným generováním**
  - jak generování svázat nějakými pravidly

Řízené generování

- nějakými pravidly popíšu co se jak má generovat
- tj. není to jen napodobování toho co bylo v datech
- nějakou chytrost, pravidla tam může vložit člověk

Řízené generování

- například: generování replik
- neuronka
  - vygeneruje jméno, dvojtečku, navazující větu
- možná pravidla
  - vygeneruj jméno postavy z daného seznamu
  - pravidlové řízení toho kdo je a kdo není na scéně – na scéně jsou tyto 3 postavy, takže by měla mluvit jedna z nich, a nejspíš ne ta co mluvila posledně

Řízené generování

- například: charakteristika postavy
- neuronka
  - postavy jsou různé
  - ale nejspíš se pokaždé při generování jedné repliky rozhodne pro nějaký typ postavy
    - vygeneruje repliku odpovídající *nějakému* typu postavy, který už viděla v datech
- možná pravidla
  - předem vygeneruj typy postav
  - pro každou postavu generuj repliky dané jejím typem

Řízené generování

- typy postav
  - buď explicitní – textový popis – a tím se řiď
    - pro každou postavu mám 10 slov, která ji popisují
  - anebo implicitní – vytáhnout z neuronky nějaký skrytý stav odpovídající postavě
    - embedink postavy – podobně jako word2vec
    - pro každou postavu mám vektor 100 čísel, která ji popisují

- embedink postavy – vektor 100 čísel
  - lze si opět představit jako konkrétní hodnoty – pohlaví, krása, věk, optimismus, zlo, agrese…
  - ve skutečnosti nevíme, co jednotlivá čísla znamenají (ale mohli bychom to případně zkoumat)
  - úkol pro neuronku: najdi takovou reprezentaci postavy, abys generovala repliky odpovídající dané postavě spíš než repliky odpovídající jiné postavě
  - typ postavy je daný replikami (jen dané postavy?)
    - slovo: kontext = souvýskyty slov ve větách
    - postava: kontext = repliky postavy? (a jiných postav?)

Prozatím všechno?

Rudolf Rosa, Ondřej Dušek, Tomáš Musil, Tom Kocmi…
uru@ufal.mff.cuni.cz

Rychlokurz počítačového zpracování jazyka

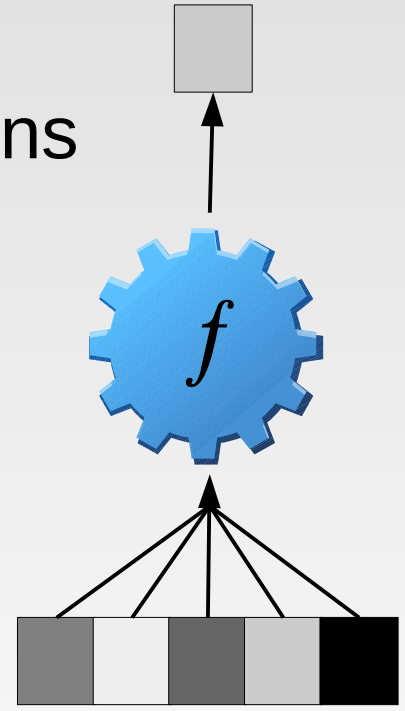
pro projekt THE**AI**TRE

Univerzita Karlova
Matematicko-fyzikální fakulta
Ústav formální a aplikované lingvistiky

https://www.theaitre.com

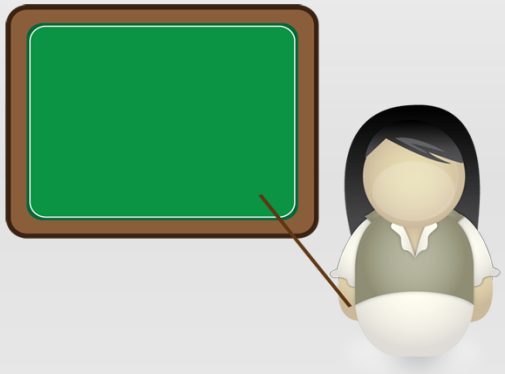
Neural networks & text processing

- Input to a neuron: fixed-dimension real vector
  - Dimension should be reasonable ($<10^3$)
  - Neural net: fixed-sized network of neurons
- Text input: sequence processing
  - Sentence = sequence of words
  - Words: discrete (but interrelated)
    - Massively multi-valued ($\sim 10^6$)
    - Very sparse (Zipf distribution)
  - Sentences: variable length (~1 to 100)
    - Complex and hidden internal structure



Outline of the talk

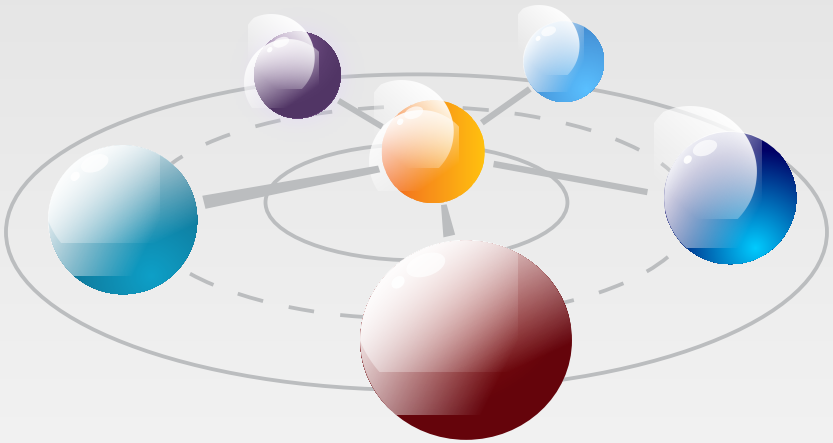
- Problem 1: Words
  - There are too many
  - They are discrete
  - *Representing massively multi-valued discrete data by continuous low-dimensional vectors*



- Problem 2: Sentences
  - They have various lengths
  - They have internal structure
  - *Handling variable-length input sequences with complex internal relations by fixed-sized neural units*

Problem 1: Words

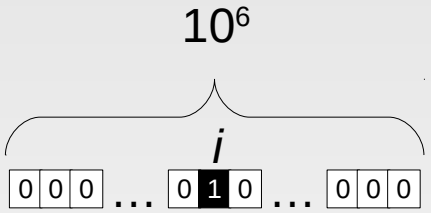
Massively multi-valued discrete data
(words)



Continuous low-dimensional vectors
(word embeddings)

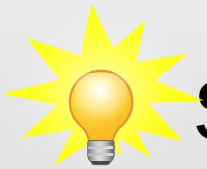
The problem with words

- How many words are there? Too many!
  - Many problems with counting words, cannot be done
  - ~10^6 (but potentially infinite – new words get created every day)
- Long-standing problem of NLP
- Natural representation: 1-hot vector

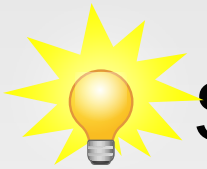


  - ML with ~10^6 binary features on input
  - Pair of words: ~10^{12}
  - No generalization, meaning of words not captured
    - dog~puppy, dog~~cat, dog~~~platypus, dog~~~~whiskey

Split the words

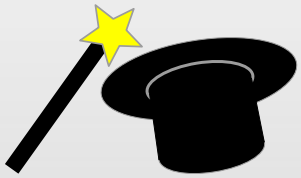


- Split into characters 
  - Not that many (~10^2) 🙂
  - Do not capture meaning 😟
    - Starts with “m-”, is it positive or negative?



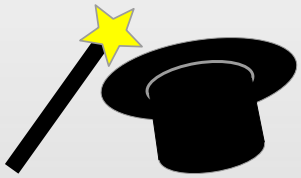
- Split into subwords/morphemes 
  - Word starts with “mis-”: it is probably negative
    - *misclassify, mistake, misconception…*
  - Helps, used in practice 🙂
    - Potentially infinite set covered by a finite set of subwords
  - Meaning-capturing subwords still too many (~10^5) 😟

Word embeddings magic

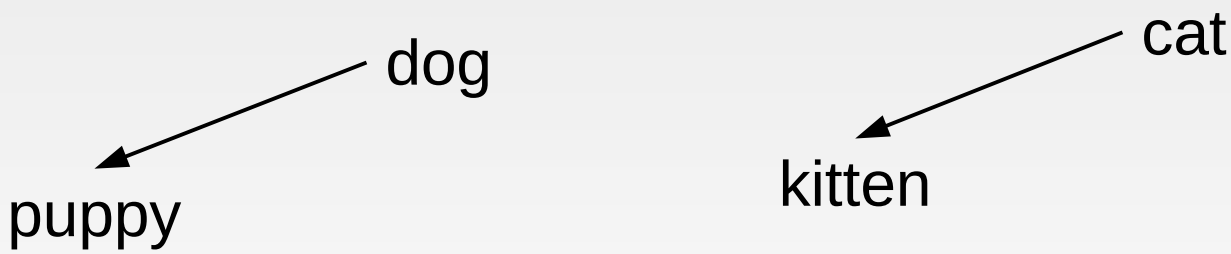


- Word similarity (cos)
  - vec(dog) ~ vec(puppy), vec(cat) ~ vec(kitten)

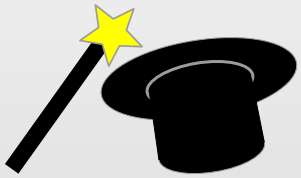
Word embeddings magic



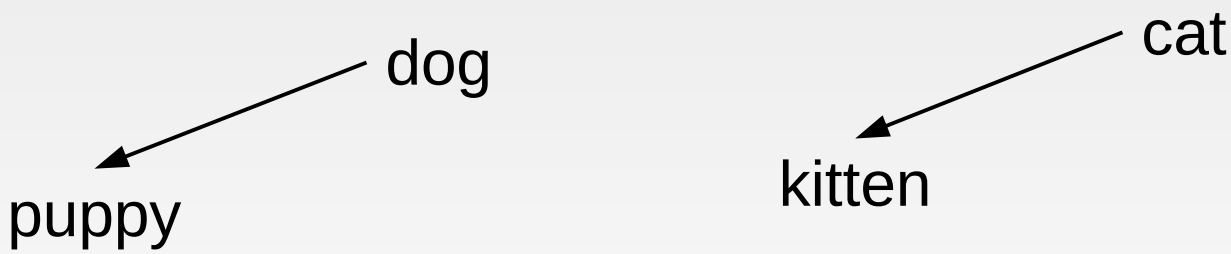
- Word similarity (cos)
  - vec(dog) ~ vec(puppy), vec(cat) ~ vec(kitten)
- Word meaning algebra
  - Some relations parallel across words
  - vec(puppy) - vec(dog) ~ vec(kitten) - vec(cat)



Word embeddings magic

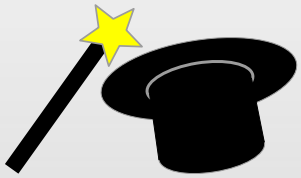


- Word similarity (cos)
  - vec(dog) ~ vec(puppy), vec(cat) ~ vec(kitten)
- Word meaning algebra
  - Some relations parallel across words
  - vec(puppy) - vec(dog) ~ vec(kitten) - vec(cat)

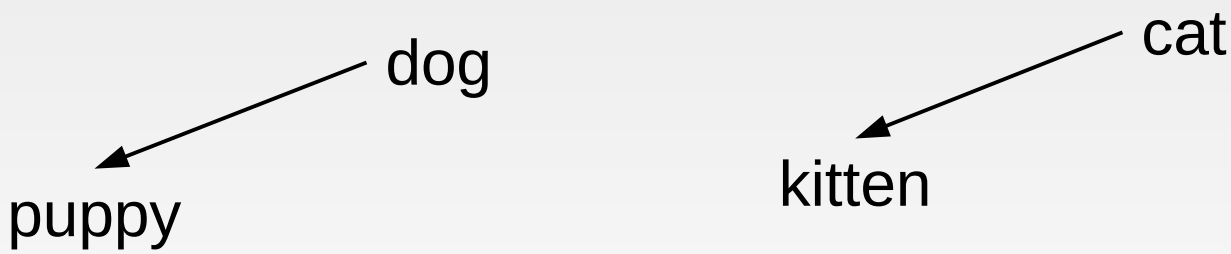


  - => vec(puppy) - vec(dog) + vec(cat) ~ vec(kitten)

Word embeddings magic



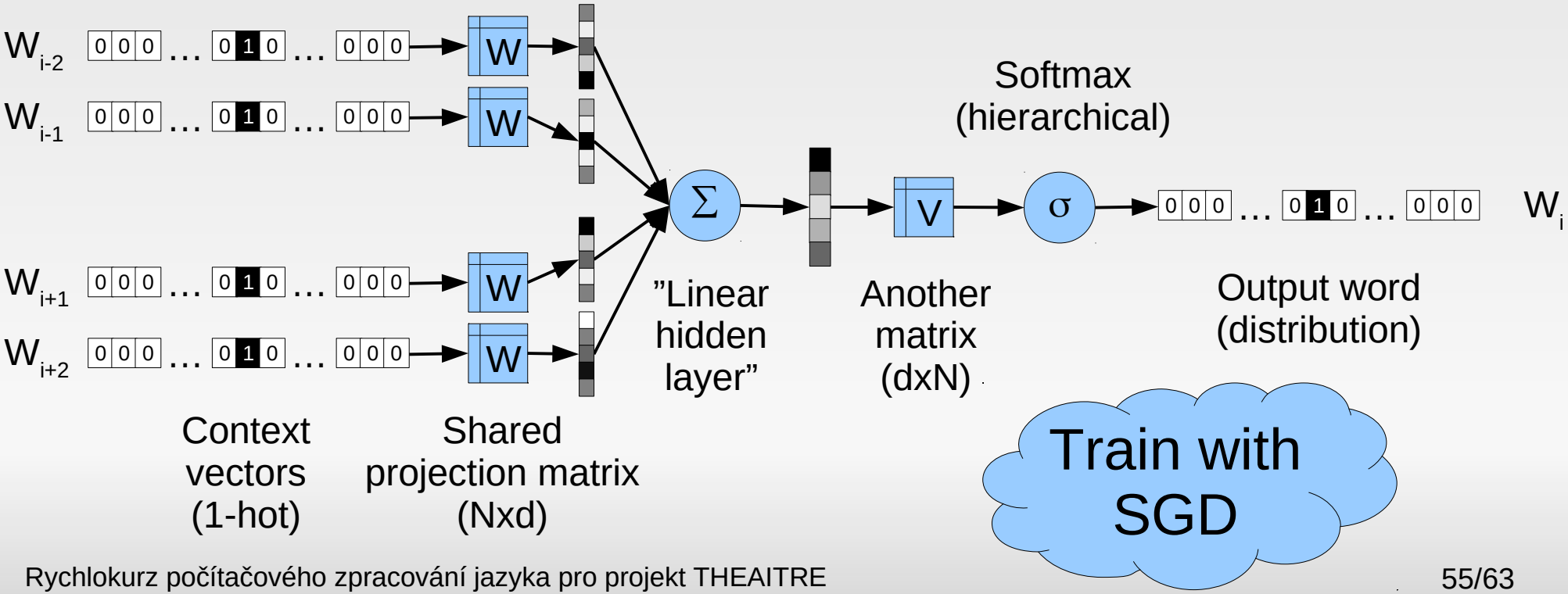
- Word similarity (cos)
  - vec(dog) ~ vec(puppy), vec(cat) ~ vec(kitten)
- Word meaning algebra
  - Some relations parallel across words
  - vec(puppy) - vec(dog) ~ vec(kitten) - vec(cat)



  - => vec(puppy) - vec(dog) + vec(cat) ~ vec(kitten)
    - vodka – Russia + Mexico, teacher – school + hospital…

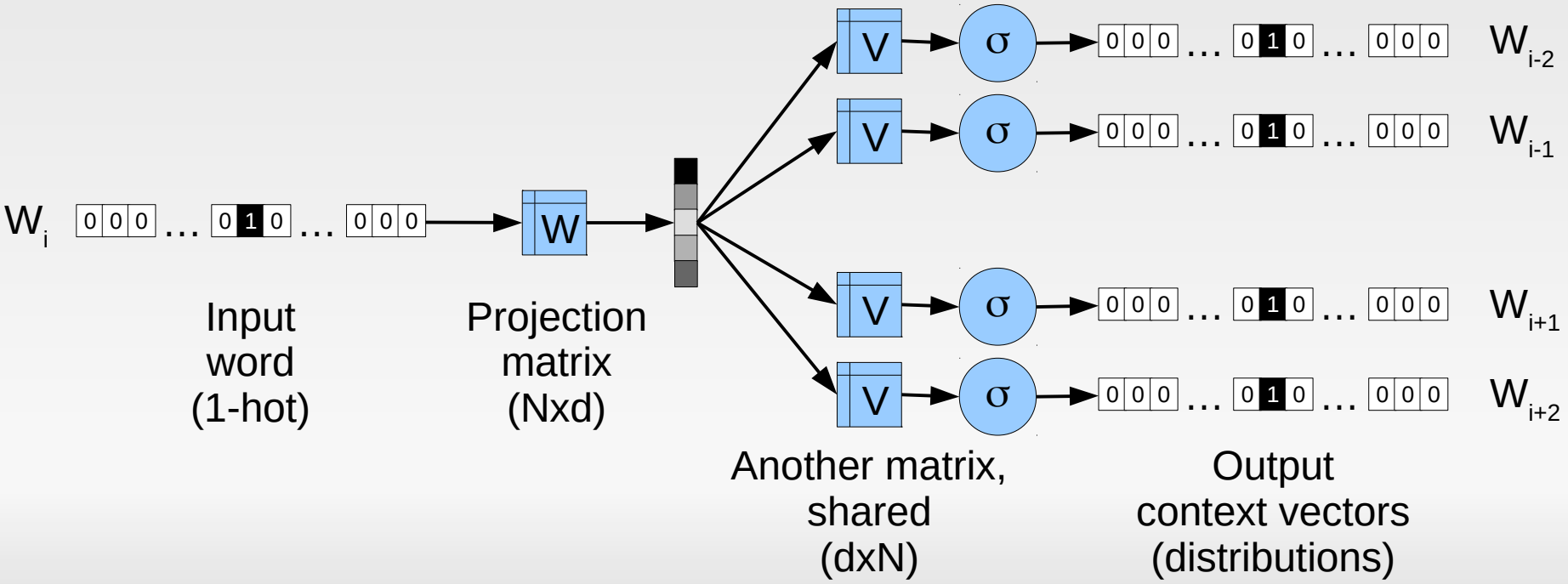
word2vec (Mikolov+, 2013)

- Predict word w_i from its context (CBOW)
  - E.g.: "*I had* _____ *for lunch*"
  - Sentence: … w_{i-2} w_{i-1} $\mathbf{w_i}$ w_{i+1} w_{i+2} …

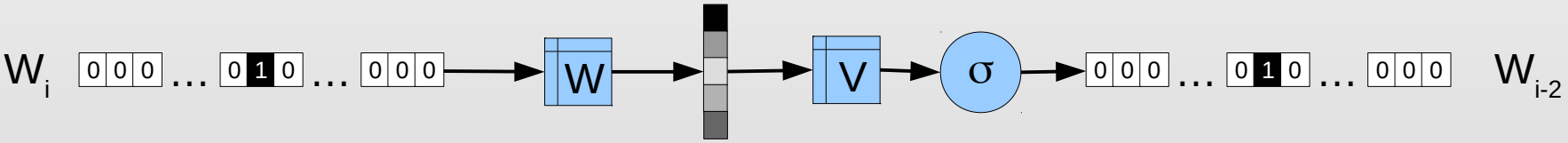


word2vec (Mikolov+, 2013)

- Predict context from a word w_i (SGNS)
  - E.g.: "____ _____ *smelt* _____ _____"
  - Sentence: … $\mathbf{w_{i-2}}$ $\mathbf{w_{i-1}}$ w_i $\mathbf{w_{i+1}}$ $\mathbf{w_{i+2}}$ …



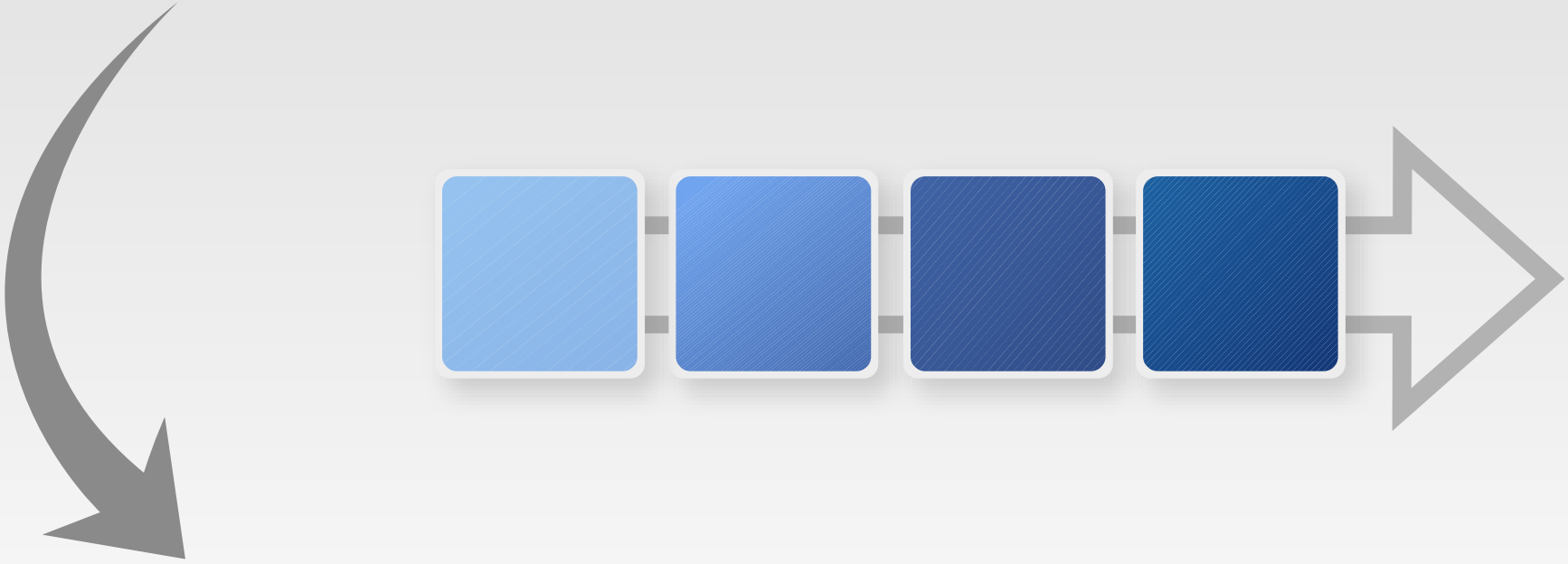
word2vec ~ implicit factorization



- Word embedding matrix $W \in \mathbf{R}^{Nxd}$
  - $\text{embedding}(\text{word}_i) = W[i] \in \mathbf{R}^d$
- Levy&Goldberg (2014)
  - word2vec SGNS implicitly factorizes M_{PMI}
  - $M_{PMI}[i, j] = \log [P(\text{word}_i \mid \text{context}_j) / P(\text{word}_i)]$
  - SGNS: $M_{PMI} = WV$
  - $M_{PMI} \in \mathbf{R}^{NxN} \rightarrow W \in \mathbf{R}^{Nxd}, V \in \mathbf{R}^{dxN}$

Problem 2: Sentences

Variable-length input sequences with long-distance relations between elements (sentences)

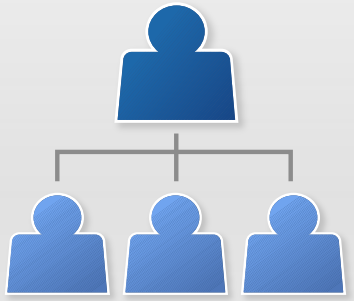


Fixed-sized neural units (attention mechanisms)

Processing sentences

- Convolutional neural netowrks
- Recurrent neural networks
- Attention mechanism
- Self-attentive networks

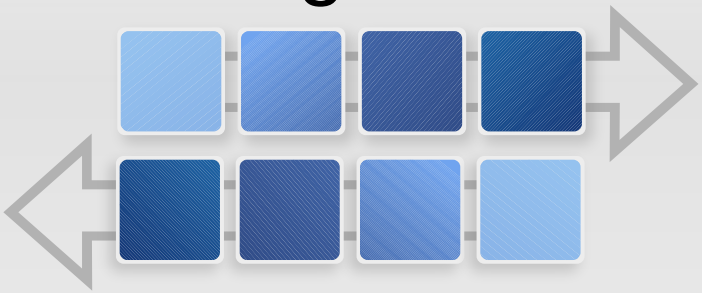
Convolutional neural networks



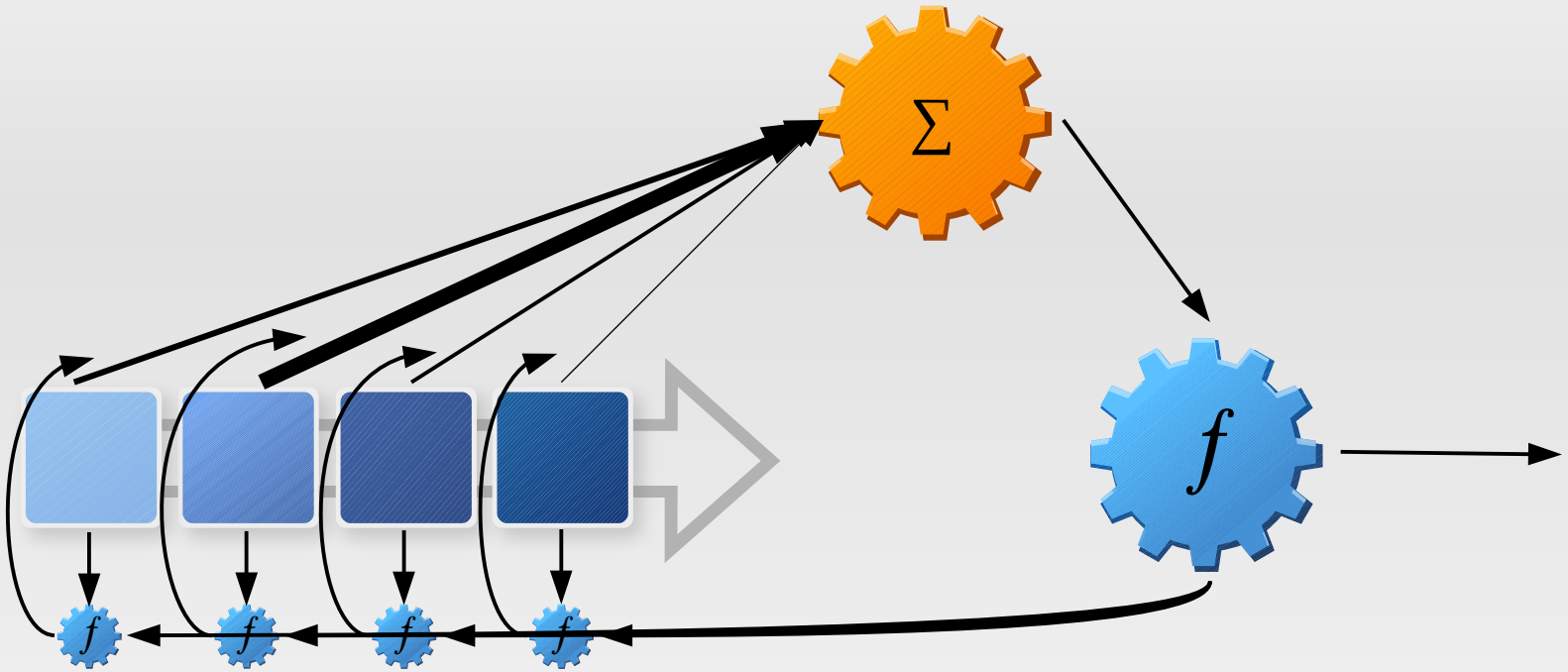
- Input: sequence of word embeddings
- Filters (size 3-5), norm, maxpooling
- Training deep CNNs hard → residual connections
  - Layer input averaged with output, skips non-linearity
- Problem: capturing long-range dependencies
  - Receptive field of each filter is limited
  - *My computer works, but I have to buy a new mouse.*
- Good for word *n*gram spotting
  - Sentiment analysis, named entity detection…

Recurrent neural networks

- Input: sequence of word embeddings
- Output: final state of RNN
- Problems
  - Vanishing gradient → memory cells (LSTM, GRU)
  - Long distance dependencies not perfectly captured
  - Final state is biased (“forgetting”)
    - …sentence end better captured than sentence start
    - Bidirectional RNN, output = concat of both final states
      - Still may not well capture the middle parts…
    - Using all hidden states as output, not just the final one
      - We loose the fixed-sized representation



Attention (on top of a RNN)



- Classifier/decoder gets a fixed-size context vector
  - Weighted average of encoder hidden states
  - Attention weights computed by a feed-forward subnet
    - $\text{weight}_i \sim \text{NN}(\text{state}_i, \text{state}_{\text{decoder}})$

Advanced attention

- Multi-head attention
  - Multiple attention heads (~8), each has its own distro
  - Resulting context vectors concatenated
- Self-attentitive encoder (SAN, Transformer)
  - CNN/attention hybrid
  - CNN: cell gets small local context via filters
  - SAN: cell gets global context via attention heads