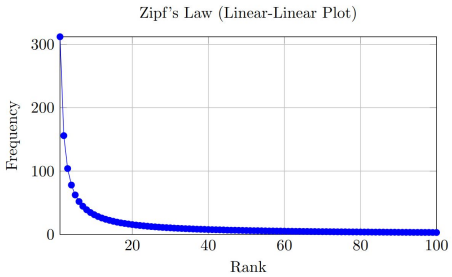
Sieci rekurencyjne

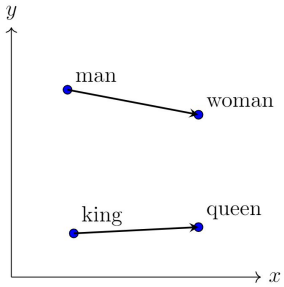
Witold Drzewakowski
17.01.2026

Roadmap

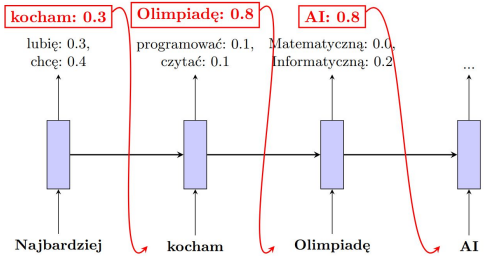
1. Podstawy NLP



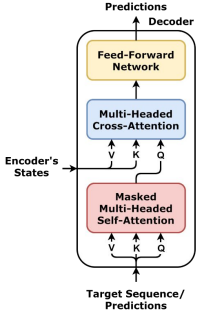
2. Wektory słów



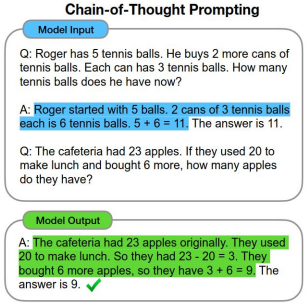
3. Rekurencyjne Sieci Neuronowe



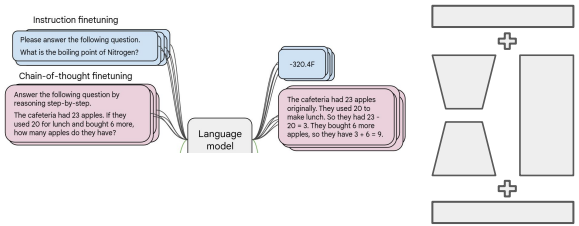
4. Atencja i transformery



5. Duże modele językowe (LLM)



6. Dostrajanie LLMów



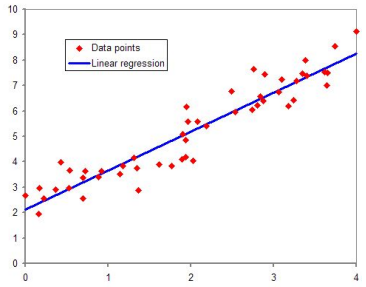
Plan

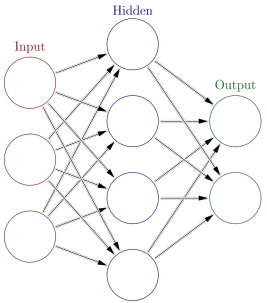
1. Dane sekwencyjne – czyli o czym będzie ten wykład
2. Przykład zadania: analiza sentymentu
3. RNNy jako modele językowe
4. Trening RNNów
5. LSTM

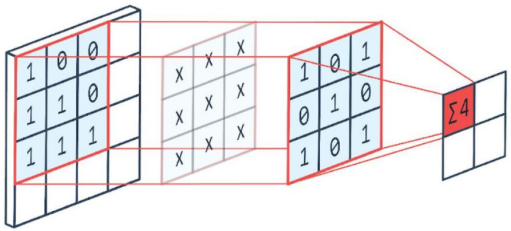
Recap

Do tej pory poznaliśmy sposoby na przetwarzanie danych tabelarycznych oraz obrazków przez sieci neuronowe. Sieci przyjmowały dane z ustaloną liczbą cech.

	D	E	F	G	H
	Sex	Age	SibSp	Parch	Ticket
	male	34.5	0	0	330911
	female	47	1	0	363272
	male	62	0	0	240276
	male	27	0	0	315154
	female	22	1	1	3101298
	male	14	0	0	7538
	female	30	0	0	330972
	male	26	1	1	248738
	female	18	0	0	2657
	male	21	2	0	A/4 48871
	male		0	0	349220
	male	46	0	0	694
	female	23	1	0	21228
	male	63	1	0	24065
	female	47	1	0	W.E.P. 5734
	female	24	1	0	SC/PARIS 2167
	male	35	0	0	233734
	male	21	0	0	2692
	female	27	1	0	STON/O2. 3101270
	female	45	0	0	2696

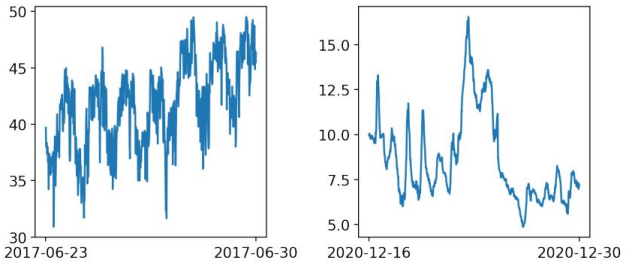




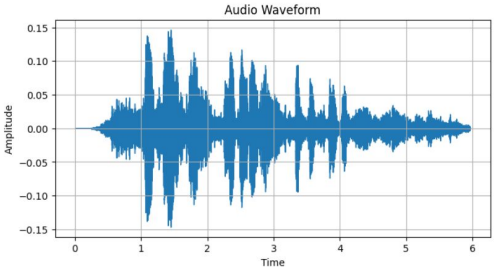


Dane sekwencyjne

- Szeregi czasowe
- Tekst
- Audio
- Wideo




```
The Fulton County Grand Jury said Friday
an investigation of Atlanta's recent
primary election produced `` no evidence
'' that any irregularities took place .
The jury further said in ...
```



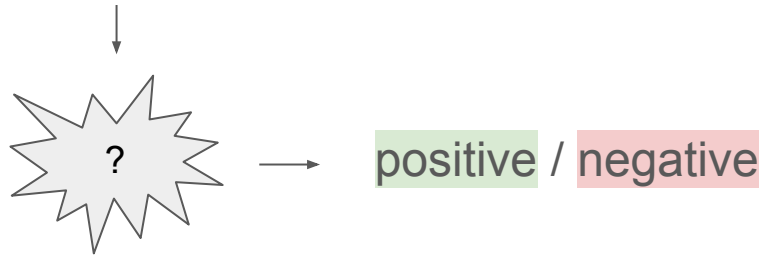
Przykład: Analiza sentymentu

FEATURED REVIEW

The greatest of all sitcoms >

This is sort of a small tribute to the show.

It is too bad that this show is no longer being made. NBC should have used the "carrot and stick" approach with the creators and maybe the show would still be on. Instead, NBC gave them so much money that they just cut and ran. Sort of like "we'll give you millions and millions of dollars to do a short run series and then you guys can go". Which is what happened.



Analiza sentymentu – **naiwne** podejścia – systemy reguł

FEATURED REVIEW

The greatest of all sitcoms >

This is sort of a small tribute to the show.

It is too bad that this show is no longer being made. NBC should have used the "carrot and stick" approach with the creators and maybe the show would still be on. Instead, NBC gave them so much money that they just cut and ran. Sort of like "we'll give you millions and millions of dollars to do a short run series and then you guys can go". Which is what happened.


```python
def classify(x: str) -> str:
    negative_keywords = ["hate", "terrible", "bad"]
    if any(keyword in x for keyword in negative_keywords):
        return "negative"
    else:
        return "positive"
```

Analiza sentymentu – **naiwne** podejścia – suma wektorów

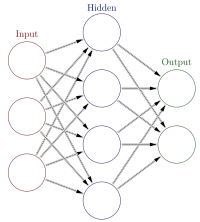
FEATURED REVIEW

The greatest of all sitcoms >

This is sort of a small tribute to the show.

It is too bad that this show is no longer being made. NBC should have used the "carrot and stick" approach with the creators and maybe the show would still be on. Instead, NBC gave them so much money that they just cut and ran. Sort of like "we'll give you millions and millions of dollars to do a short run series and then you guys can go". Which is what happened.

$$v_{\text{word 1}} + v_{\text{word 1}} + \ldots + v_{\text{word n}} = v$$



positive / negative

Analiza sentymentu – wyzwania

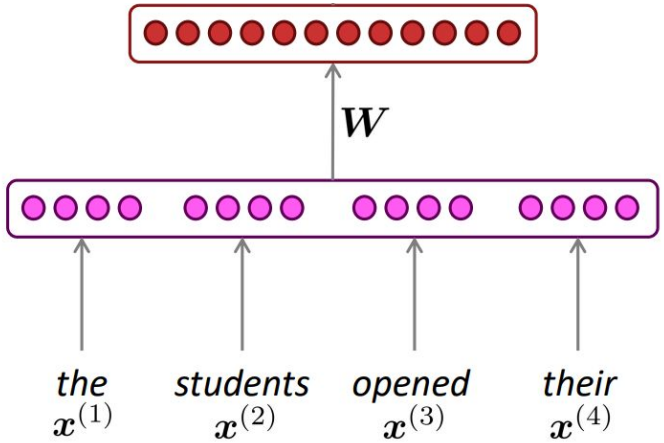
- Recenzje mogą być różnej długości
- Kontekst i pamięć są często potrzebne
- Kolejność słów ma znaczenie

The documentary was enjoyable, even though the costumes were not really historically accurate.

Even though the costumes were historically accurate, the documentary was not really enjoyable.

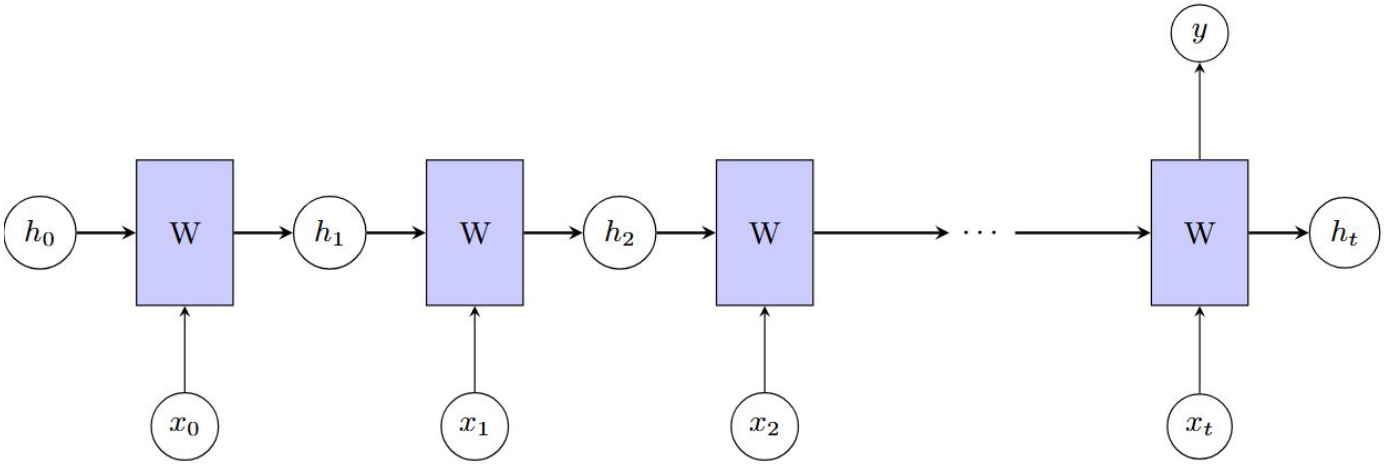
Analiza sentymentu – zmienna liczba cech

- Słowa (lub podsłowa) są reprezentowane w komputerze jako wektory.
- Jeśli każdy wektor ma wymiar 64, a zdanie ma n słów (lub podsłów) to potrzebujemy, aby sieć przyjmowała 64*n wejść – zmienna liczba!



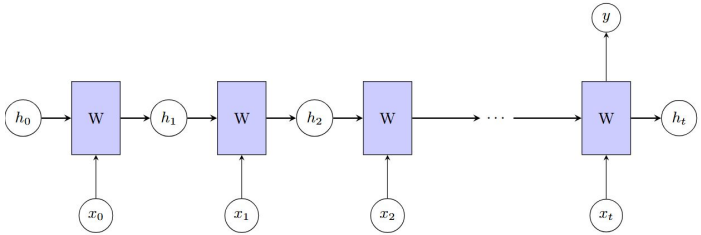
Sieci rekurencyjne – pomysł

Przetwarzamy teksty słowo po słowie, używając wielokrotnie tych samych wag *W*.



$$h_i = \tanh(W_{\mathrm{hh}} h_{i-1} + W_{hx} x_i + b_{\mathrm{h}})$$

Sieci rekurencyjne – analiza



Recenzje mogą być różnej długości.	Możemy przetworzyć wejście dowolnej długości i rozmiar modelu nie wzrasta wraz z długością sekwencji.
Kontekst i pamięć są często potrzebne.	Token może korzystać z informacji ze wszystkich poprzednich tokenów.
Kolejność słów ma znaczenie	Wyrazy są przetwarzane w kolejności.

RNNy w pytorchu (1)

```
class RNN(nn.Module):
  def __init__(self, V, E=128, H=128):
    super().__init__()
    self.emb = nn.Embedding(V, E, padding_idx=0)
    self.rnn = nn.RNN(E, H, num_layers=2, batch_first=True, dropout=0.2)
    self.fc = nn.Linear(H, 1)

  def forward(self, x, lens):
    p = nn.utils.rnn.pack_padded_sequence(self.emb(x), lens.cpu(),
                                          batch_first=True, enforce_sorted=False)
    _, h = self.rnn(p)
    return self.fc(h[-1]).squeeze(-1)
```

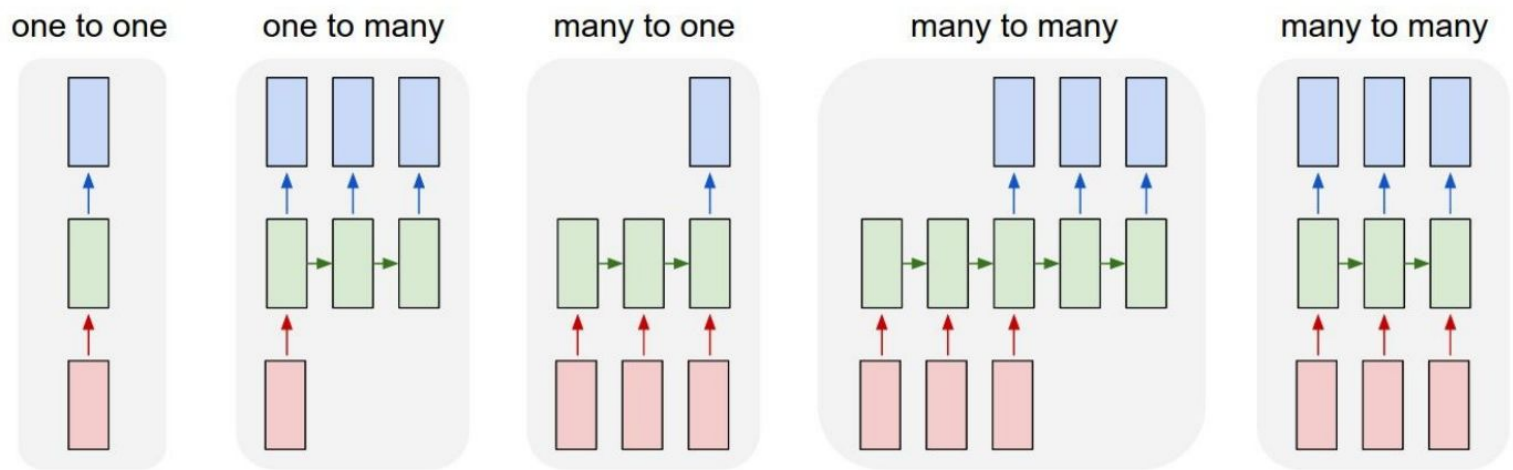
RNNy w pytorchu (2)

```
ds_train, ds_test = datasets.load_dataset('imdb', split=['train', 'test'])
tokenize = lambda s: re.findall(r'\b\w+\b', s.lower())
cnt = Counter(w for t in ds_train['text'] for w in tokenize(t))

vocab = {'<pad>': 0, '<unk>': 1,
         **{w: i + 2 for i, (w, _) in enumerate(cnt.most_common(V - 2))}}
enc = lambda s: [vocab.get(w,1) for w in tokenize(s)]
```

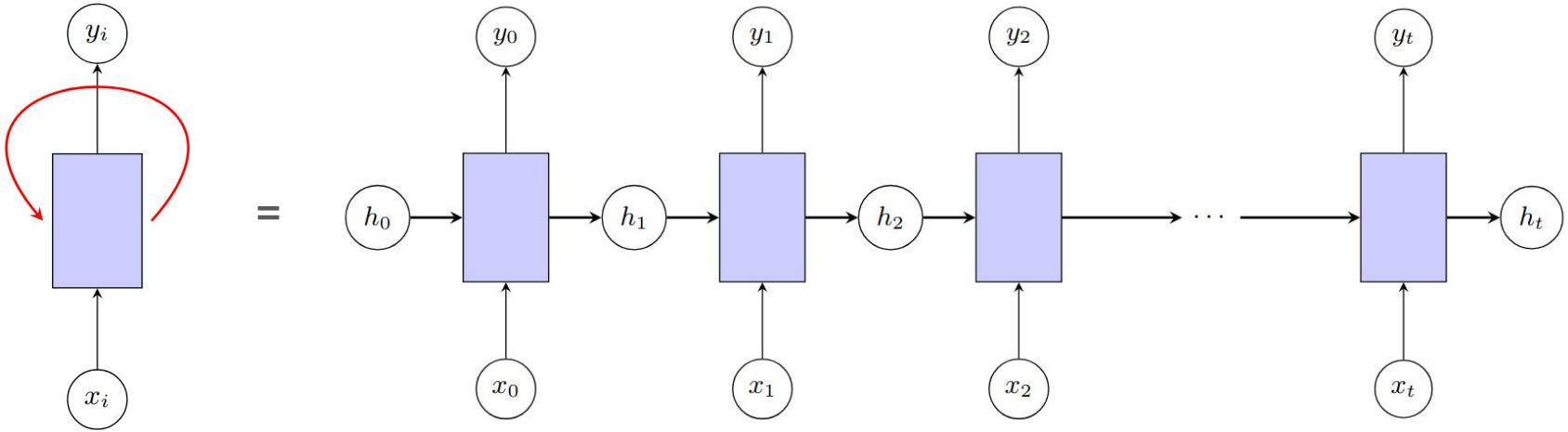
RNNy w pytorchu (3)

```
class DS(Dataset):  # __len__, __getitem__ skipped
  def __init__(self, d):
    s.x = [enc(t) for t in d['text']]
    s.y = d['label']

def collate(b):
  xs, ys = zip(*b)
  lens = torch.tensor([len(x) for x in xs])
  X = torch.zeros(len(xs), int(lens.max()), dtype=torch.long)
  for i, x in enumerate(xs):
    X[i, :len(x)] = torch.tensor(x)
  return X, lens, torch.tensor(ys)

dl_tr = DataLoader(DS(tr), batch_size=128, shuffle=True, collate_fn=collate)
```

Typy zadań w NLP



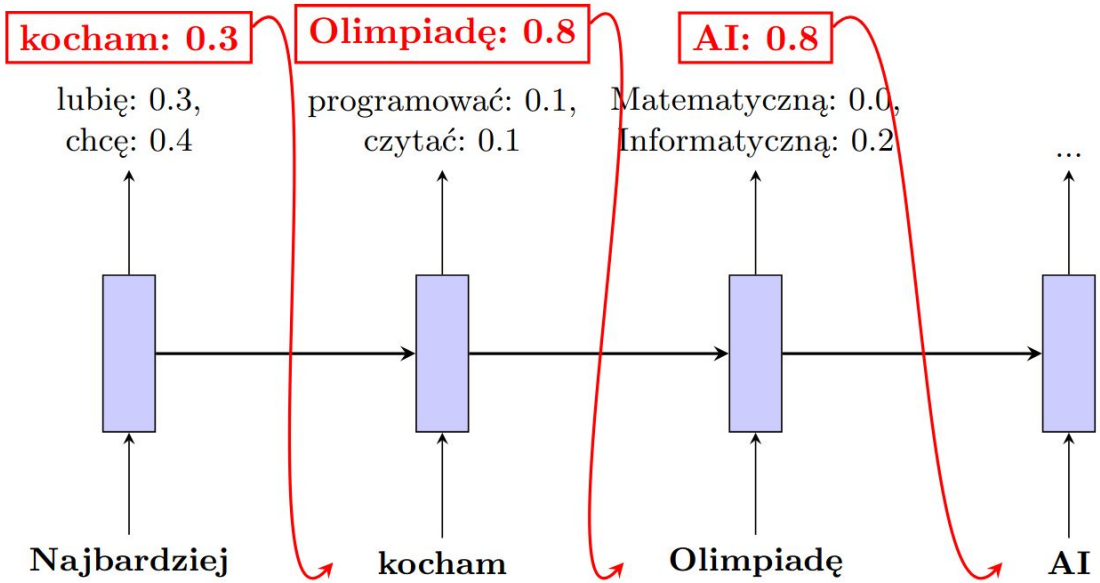
Sieci rekurencyjne – raz jeszcze



$$h_i = \tanh(W_{\mathrm{hh}} h_{i-1} + W_{hx} x_i + b_{\mathrm{h}})$$

$$y_i = W_{\mathrm{yh}} h_i + b_{\mathrm{y}}$$

RNNy jako modele językowe

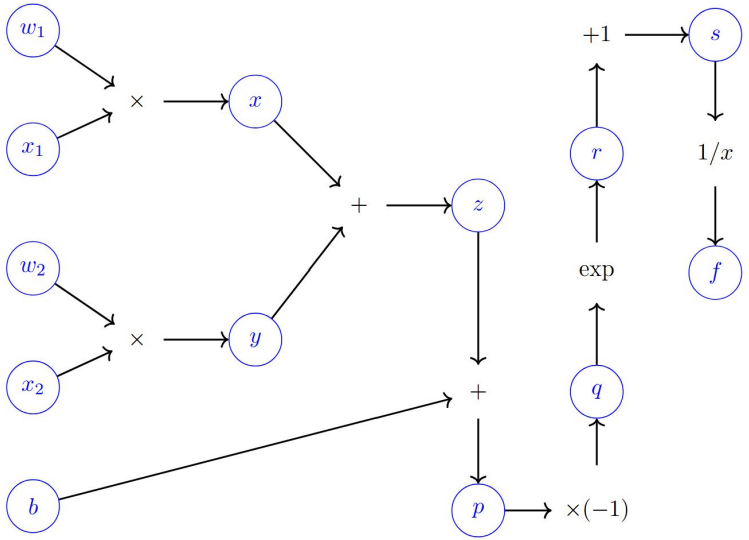


Trening RNNów

Recap: backpropagation (1) – Graf obliczeniowy

$$f(w_1, x_1, w_2, x_2, b) = \sigma\left(b + w_1 x_1 + w_2 x_2\right) = \frac{1}{1 + \exp\left(-(b + w_1 x_1 + w_2 x_2)\right)}$$

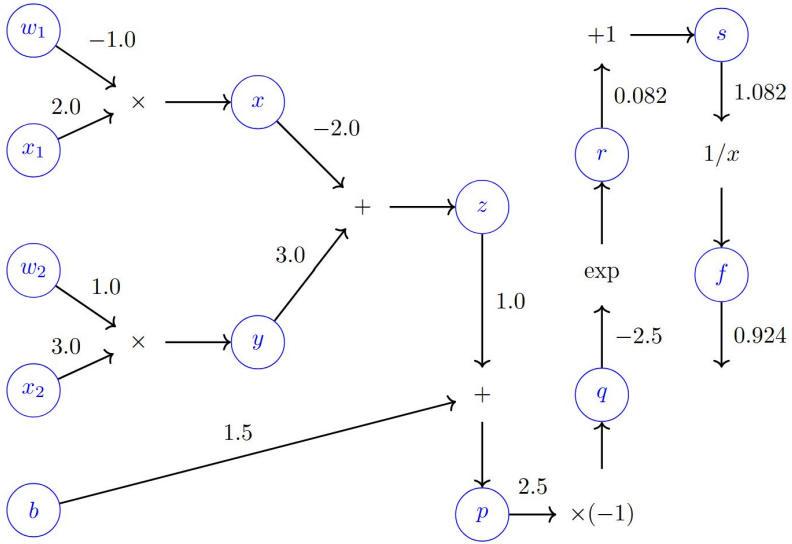
$$(w_1, x_1, w_2, x_2, b) = (-1.0, 2.0, 1.0, 3.0, 1.5)$$



Recap: backpropagation (2) – Forward pass

$$f(w_1, x_1, w_2, x_2, b) = \sigma\left(b + w_1 x_1 + w_2 x_2\right) = \frac{1}{1 + \exp\left(-(b + w_1 x_1 + w_2 x_2)\right)}$$

$$(w_1, x_1, w_2, x_2, b) = (-1.0, 2.0, 1.0, 3.0, 1.5)$$



Recap: backpropagation (3) – Backward pass

$$\frac{\partial f}{\partial f} = 1$$

$$\frac{\partial f}{\partial s} = -\frac{1}{s^2} \approx -0.854$$

$$\frac{\partial s}{\partial r} = \frac{\partial r + 1}{\partial r} = 1$$

$$\frac{\partial r}{\partial q} = \exp q = 0.082$$

$$\frac{\partial q}{\partial p} = -1$$

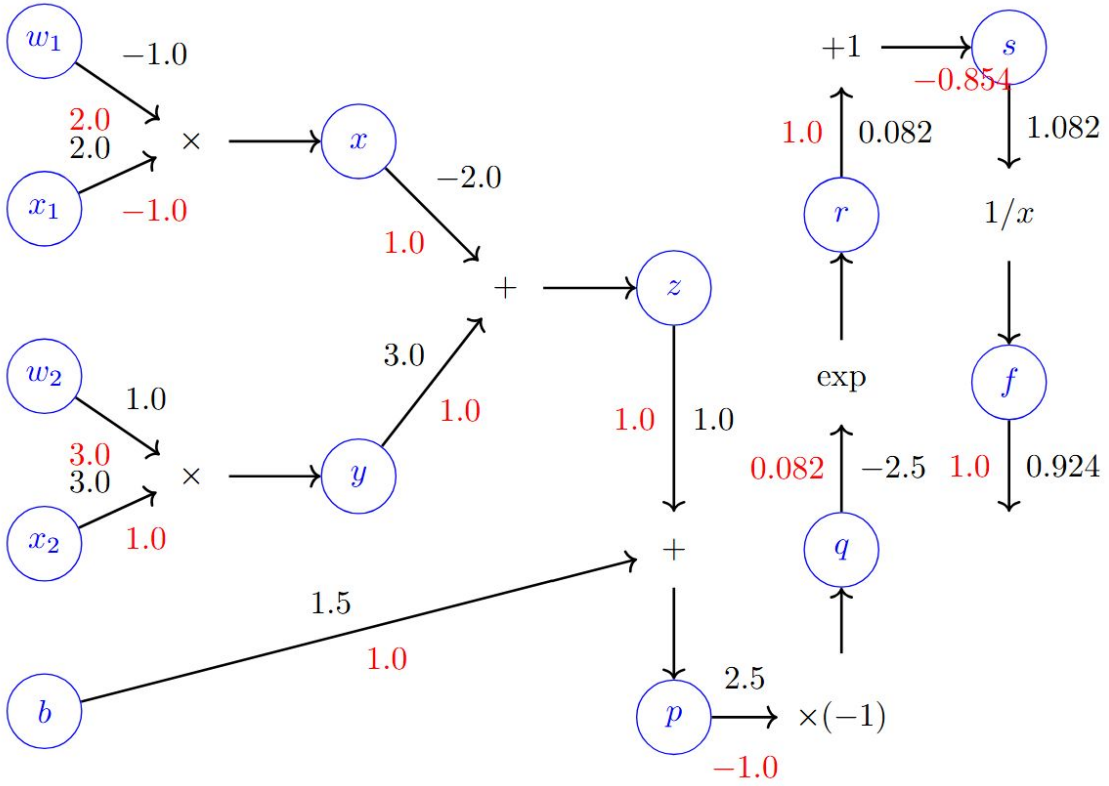
$$\frac{\partial p}{\partial b} = \frac{\partial (z + b)}{\partial b} = 1$$

$$\frac{\partial p}{\partial z} = \frac{\partial (z + b)}{\partial z} = 1$$

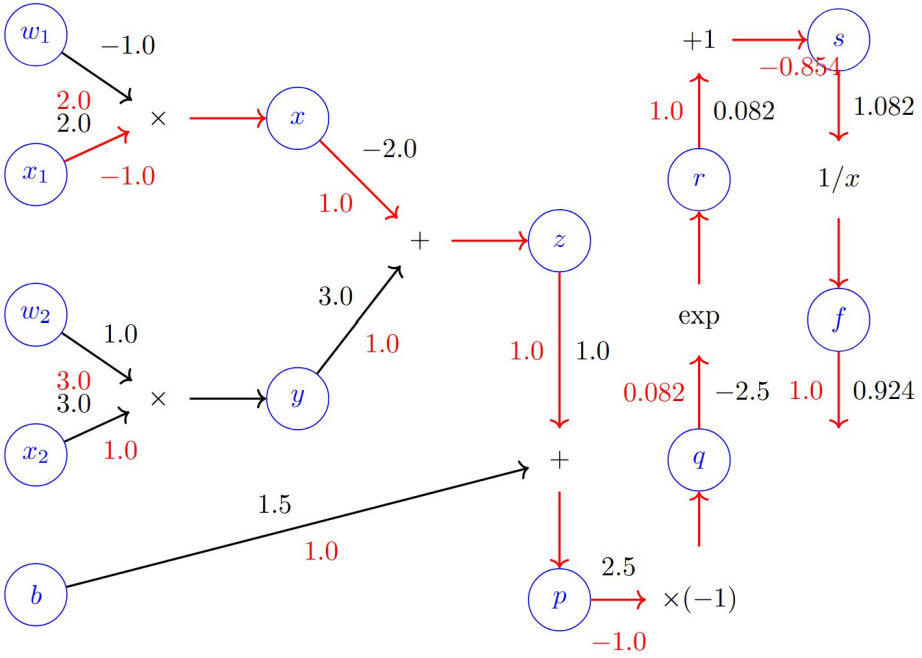
$$\frac{\partial z}{\partial x} = \frac{\partial (x + y)}{\partial x} = 1$$

$$\frac{\partial z}{\partial y} = \frac{\partial (x + y)}{\partial y} = 1$$

$$\dots$$



Recap: backpropagation (4) – Backward pass



Chain rule

$$\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial s}\frac{\partial s}{\partial r}\frac{\partial r}{\partial q}\frac{\partial q}{\partial p}\frac{\partial p}{\partial z}\frac{\partial z}{\partial x}\frac{\partial x}{\partial w_1} = -0.854 \cdot 1 \cdot 0.082 \cdot (-1) \cdot 1 \cdot 1 \cdot (-1) = -0.070$$

Trening RNNów – korpus

Brown Corpus

```
The Fulton County Grand Jury said Friday an
investigation of Atlanta's recent primary election
produced `` no evidence '' that any irregularities
took place . The jury further said in...
```

```
The/at Fulton/np-tl County/nn-tl Grand/jj-tl
Jury/nn-tl said/vbd Friday/nr an/at investigation/nn
of/in Atlanta's/np$ recent/jj primary/nn election/nn
produced/vbd ``/`` no/at evidence/nn ''/'' that/cs
any/dti irregularities/nns took/vbd place/nn ./.
```

IMDB Movie Reviews

FEATURED REVIEW

The greatest of all sitcoms ›

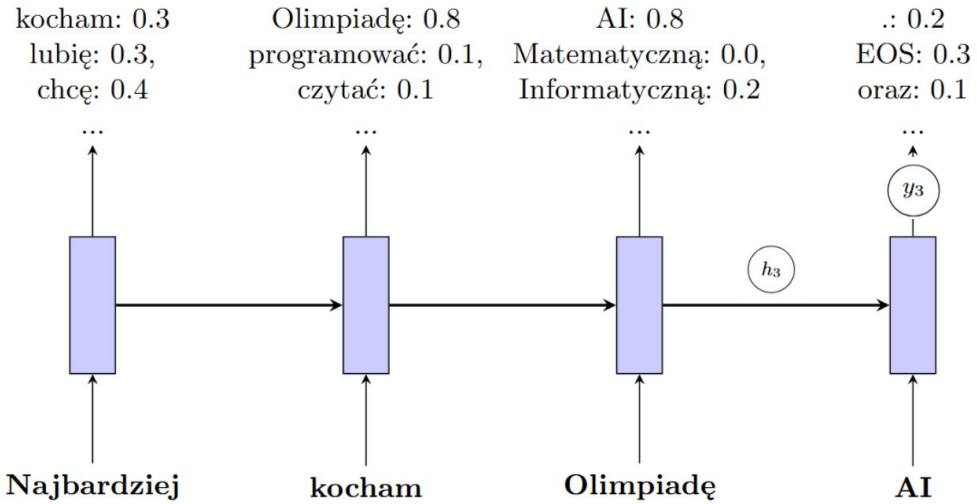
This is sort of a small tribute to the show.

It is too bad that this show is no longer being made. NBC should have used the "carrot and stick" approach with the creators and maybe the show would still be on. Instead, NBC gave them so much money that they just cut and ran. Sort of like "we'll give you millions and millions of dollars to do a short run series and then you guys can go". Which is what happened.

WikiText

Trening RNNów – przykład: *next token prediction*

1. Korpus treningowy: (x_1, x_2, …, x_n) wraz z etykietami.
2. Wykonujemy *forward pass* przez RNN (wagi losowo zainicjalizowane)

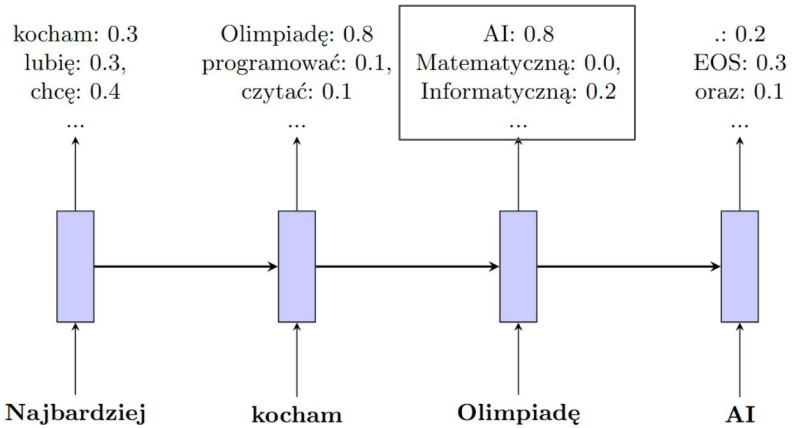


Każde zdanie trzeba przetworzyć słowo po słowie!

Zanim policzymy y_3
Musimy mieć obliczone h_3

Trening RNNów – przykład: *next token prediction*

1. Korpus treningowy: (x_1, x_2, …, x_n) wraz z etykietami.
2. Wykonujemy *forward pass* przez RNN (wagi losowo zainicjalizowane)
3. Dla każdego słowa obliczamy funkcję straty (loss) między prawdziwym następnym słowem, a przewidzianym rozkładem, cross-entropię i uśredniamy.

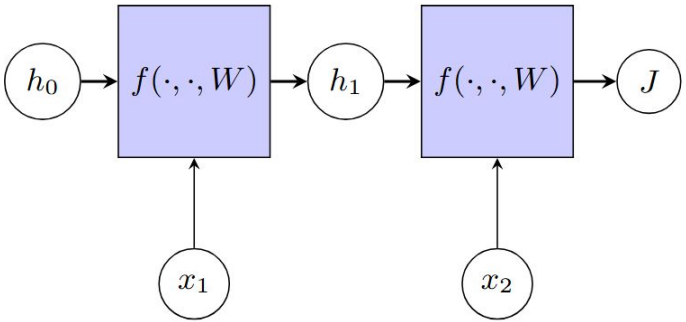


$$\text{Cross-Entropy}\left(\begin{array}{c}\text{AI}: 0.8\\ \text{Matematyczną}: 0.0\\ \text{Informatyczną}: 0.2\text{'}\\ \ldots\end{array}\quad \text{AI}\right)$$

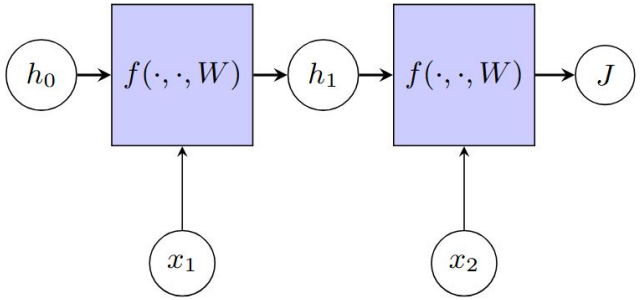
$$J^{(t)}(\theta) = CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = -\sum_{w \in V} \boldsymbol{y}_w^{(t)} \log \hat{\boldsymbol{y}}_w^{(t)} = -\log \hat{\boldsymbol{y}}_{\boldsymbol{x}_{t+1}}^{(t)}$$

Backpropagation through time

- Chcemy wykonać mały krok w stronę największego spadku funkcji straty – potrzebujemy obliczyć gradient funkcji straty po wagach sieci.
- Ale te same wagi *W* występują w sieci wielokrotnie! - jak sobie z tym poradzić?



Backpropagation through time – przykład



$$h_0 \in \mathbb{R}, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{R}, \quad W \in \mathbb{R}, \quad f : \mathbb{R}^3 \to \mathbb{R}$$
$$h_1 = f(h_0, x_1, W)$$
$$J = f(h_1, x_2, W).$$

Rozważmy prosty przykład:

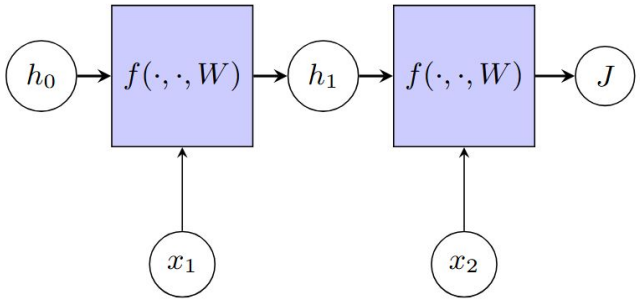
- *W* to wagi, współdzielone przez obie “warstwy” “sieci”.
- Wszystkie liczby są rzeczywiste.
- f to jakaś funkcja, np.:

  f(h, x, w) = hxw

- Chcemy minimalizować *J*, czyli znaleźć gradient *J* po *W*.
- Widzimy, że *J* zależy od *W* na dwa sposoby: przez 1. i 2. warstwę.

Backpropagation through time – przykład

$$\frac{d}{dt} f(x(t), y(t)) = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$$



$$h_0 \in \mathbb{R}, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{R}, \quad W \in \mathbb{R}, \quad f : \mathbb{R}^3 \to \mathbb{R}$$

$$h_1 = f(h_0, x_1, W)$$
$$J = f(h_1, x_2, W).$$

Używamy reguły łańcuchowej:

$$\frac{dJ}{dW} = \frac{df(h_1, x_2, W)}{dW} = \frac{\partial f}{\partial h}(h_1, x_2, W) \cdot \frac{dh_1}{dW} + \frac{\partial f}{\partial W}(h_1, x_2, W)$$

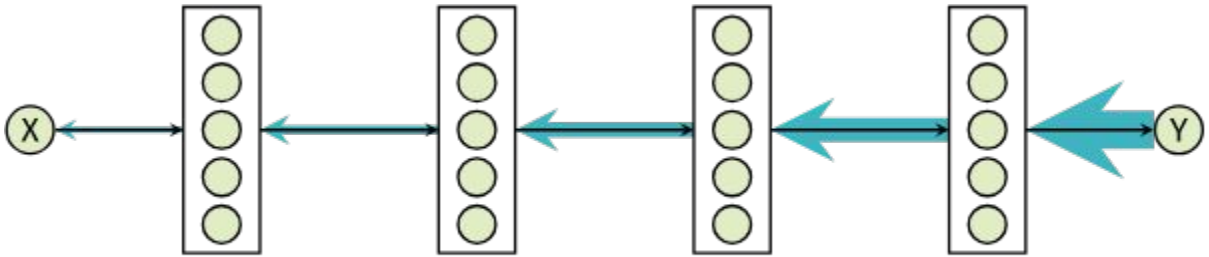
$$\frac{dh_1}{dW} = \frac{df(h_0, x_1, W)}{dW} = \frac{\partial f}{\partial h}(h_0, x_1, W) \cdot \frac{dh_0}{dW} + \frac{\partial f}{\partial W}(h_0, x_1, W)$$

Czyli *propagacja wsteczna przez czas* wymaga przejścia przez wszystkie tokeny z przeszłości!

Możemy traktować kopie W ze wszystkich warstw niezależnie, policzyć po nich osobno gradienty, w odpowiednich punktach zależnych od wejść do danej warstwy, a następnie zsumować!

Zanikający gradient (1)

- Bardzo często zdarza się, że w czasie propagacji wstecznej, *długości (wielkości)* gradientów maleją wraz z kolejnymi krokami.
- To sprawia, że model **nie jest w stanie** nauczyć się agregować informacji z odległych tokenów.
- Często efektywnie model uczy się używać tylko około 15 ostatnich słów.



Źródło obrazka [link]

Zanikający gradient (2) – dlaczego?

- Gdy jest więcej tokenów, kontrybucja gradientu *W* w warstwie *n* od końca jest ograniczona przez iloczyn wszystkich poprzednich czerwonych pochodnych.
- Jeśli np. czerwone pochodne wszystkie mają długości (moduły) < 1, to spadek jest wykładniczy: $\text{Jeżeli } |\partial f/\partial h(\cdot)| \leqslant c < 1, \text{ to } |\partial f/\partial h(\cdot)|^n \xrightarrow[n\to\infty]{} 0.$

$$\frac{dJ}{dW} = \frac{df(h_t, x_t, W)}{dW} = \frac{\partial f}{\partial h}(h_t, x_t, W) \cdot \frac{dh_t}{dW} + \frac{\partial f}{\partial W}(h_t, x_t, W)$$

$$\dots$$

$$\frac{dh_2}{dW} = \frac{df(h_1, x_1, W)}{dW} = \frac{\partial f}{\partial h}(h_1, x_1, W) \cdot \frac{dh_1}{dW} + \frac{\partial f}{\partial W}(h_1, x_1, W)$$

$$\frac{dh_1}{dW} = \frac{df(h_0, x_0, W)}{dW} = \frac{\partial f}{\partial h}(h_0, x_0, W) \cdot \frac{dh_0}{dW} + \frac{\partial f}{\partial W}(h_0, x_0, W)$$

Eksplodujący gradient

- Gdy gradienty mają normy > 1, może zajść eksplozja gradientu.
- Trening jest bardzo niestabilny, może doprowadzać do wielu NaNów.
- Dobrze jest explicite ograniczać długość kroku gradientowego.

$$\frac{dJ}{dW} = \frac{df(h_t, x_t, W)}{dW} = \frac{\partial f}{\partial h}(h_t, x_t, W) \cdot \frac{dh_t}{dW} + \frac{\partial f}{\partial W}(h_t, x_t, W)$$

$$\dots$$

$$\frac{dh_2}{dW} = \frac{df(h_1, x_1, W)}{dW} = \frac{\partial f}{\partial h}(h_1, x_1, W) \cdot \frac{dh_1}{dW} + \frac{\partial f}{\partial W}(h_1, x_1, W)$$

$$\frac{dh_1}{dW} = \frac{df(h_0, x_0, W)}{dW} = \frac{\partial f}{\partial h}(h_0, x_0, W) \cdot \frac{dh_0}{dW} + \frac{\partial f}{\partial W}(h_0, x_0, W)$$

Truncated backpropagation through time

- RNNy nie agregują dobrze informacji z odległych słów.
- (Pełna) propagacja wsteczna przez czas działa w czasie kwadratowym.

Pomysł: cofajmy się tylko o określoną liczbę kroków.

$$\frac{\partial h_t}{\partial W} \approx \sum_{i=1}^{20} \frac{\partial h_t}{\partial W}\Big|_{(t+1-i)}$$

LSTM – intuicja

- LSTM (Long Short-Term Memory) posiada specjalną bramkę pamięci, która
  - zapamiętuje istotne informacje przez długi czas
  - zapomina nieistotne
  - nie ma funkcji aktywacji, co powoduje, że gradient płynie bez przeszkód

Forget gate → $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$

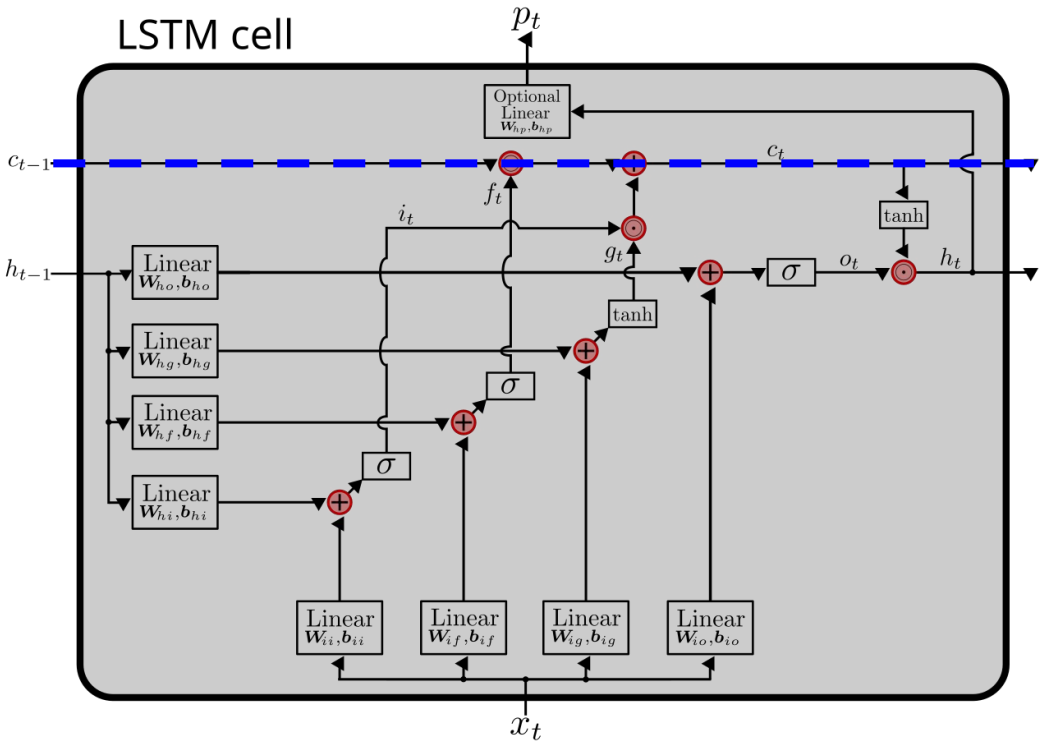
Input gate → $i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$

Candidate cell state → $\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$

Memory cell → $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$

Brak funkcji aktywacji!

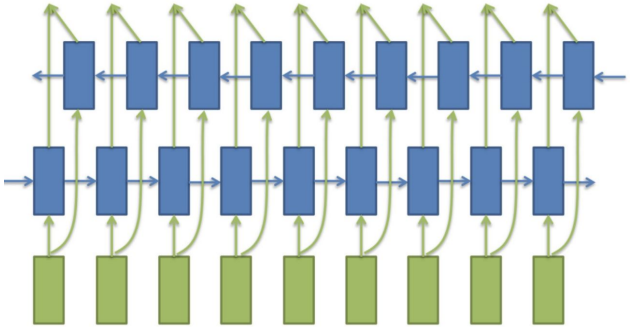
LSTM cell

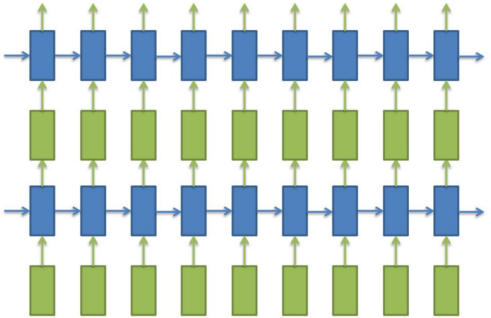


$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$h_t = o_t \odot \sigma_h(c_t)$$

Głębsze RNNy

```
class torch.nn.RNN(input_size, hidden_size, num_layers=1,
nonlinearity='tanh', bias=True, batch_first=False, dropout=0.0,
bidirectional=False, device=None, dtype=None)
```



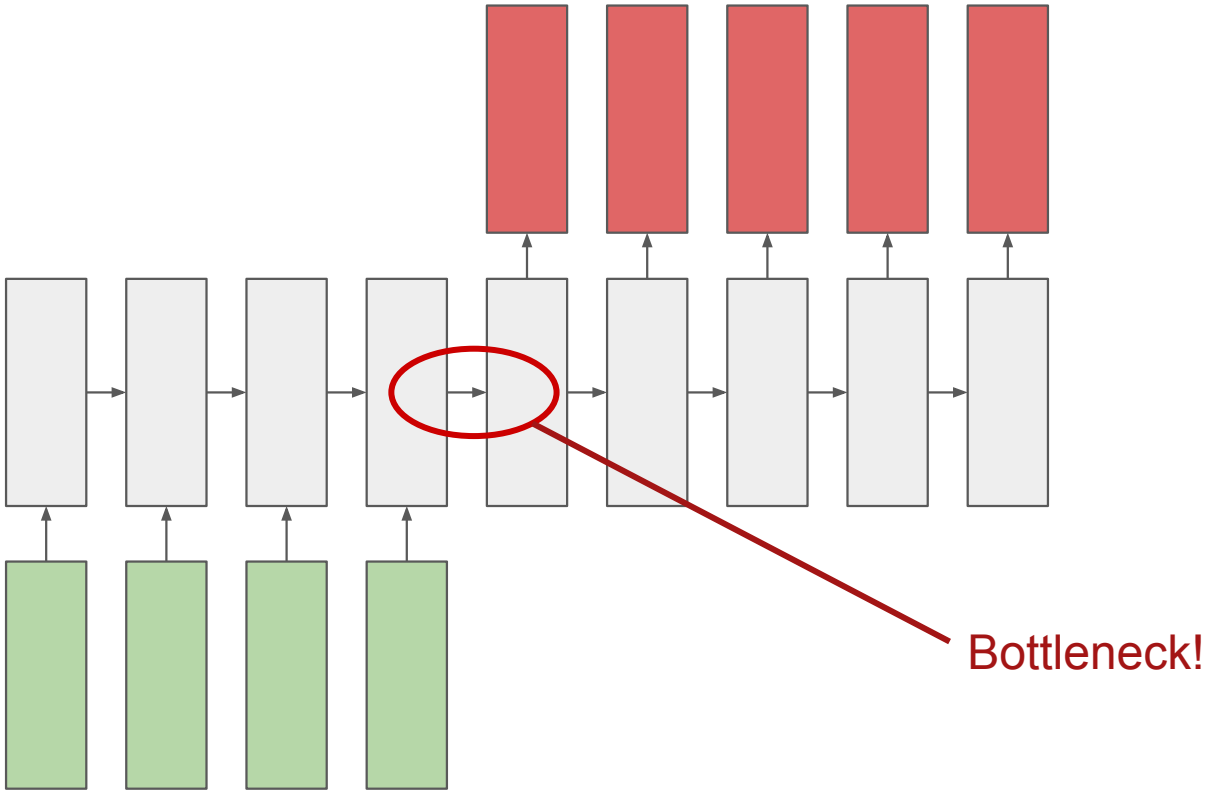


Metryka – perplexity

- Perplexity wyraża jak dobrze model jest w stanie przewidzieć następny token w określonym korpusie.
- Jest to odwrotność prawdopodobieństwa sekwencji z korpusu według modelu. Wielkość jest także znormalizowana przez długość sekwencji T.

$$\text{perplexity} = \prod_{t=1}^{T} \left(\frac{1}{P_{\text{LM}}(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})} \right)^{1/T}$$

Ograniczenie RNNów



Podsumowanie

- Czym różnią się zadania sekwencyjne od niesekwencyjnych?
- Czym różni się backpropagacja w sieciach MLP i w RNNach?
- Idea backpropagacji wstecznej przez czas: traktujemy kopie tych samych wag osobno, a następnie sumujemy gradienty.
- Słaba stabilność numeryczna RNNów.