Süvaõpe

(ehk sügavõpe / deep learning)

Loeng 3, Tehisintellekt
14. september, 2022
Mark Fišel

DALL·E My collection

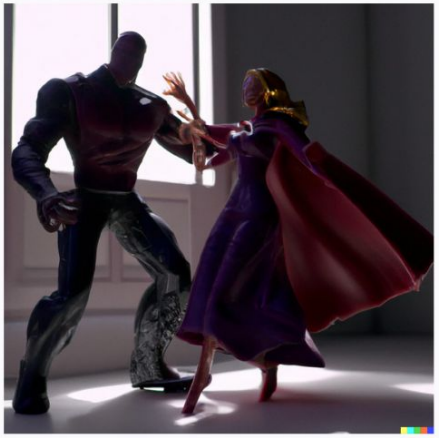
Edit the detailed description

Surprise me Upload

A photo of Thanos waltzing with the scarlet witch

Generate



openai.com

Eelmine loeng:

- masinõpe: ära selgita, vaid näita
- mudel vabade parameetritega
- näited = andmed
  - train/dev/test
- andmetes on näited esindatud tunnustega
  - num/nom, puuduvad, konverteerimine

Tänane loeng:

1. süvaõpe
   - närvivõrgud
2. õppimine
   - end-to-end gradientlaskumine
3. tunnused
   - toorsisend / automaatsed tunnused
4. näited
   - transformerid, konvolutsioonivõrgud jne

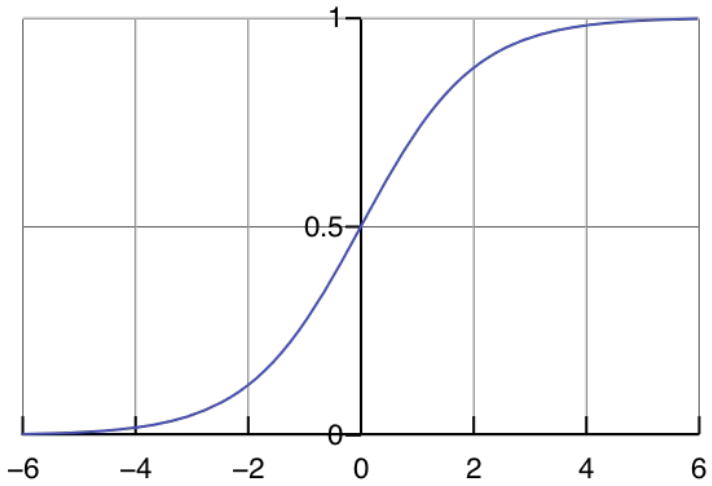
Lineaarregressioon

$$y = \boldsymbol{\theta}^T \boldsymbol{x} = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

Logistiline regressioon

$$y = \sigma(\boldsymbol{\theta}^T \mathbf{x}) = \sigma(\theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n)$$

$$\sigma(t) = 1 / (1 + e^{-t})$$



Logistiline regressioon mitme väljundiga

$$y_1 = \sigma(\theta_{01} x_0 + \theta_{11} x_1 + \theta_{21} x_2 + \ldots)$$
$$y_2 = \sigma(\theta_{02} x_0 + \theta_{12} x_1 + \theta_{22} x_2 + \ldots)$$
$$y_3 = \sigma(\theta_{03} x_0 + \theta_{13} x_1 + \theta_{23} x_2 + \ldots)$$
$$\ldots$$

Ehk: $\boldsymbol{y} = \boldsymbol{\sigma}(\Theta^T \boldsymbol{x})$

$\boldsymbol{x} = \langle x_0, x_1, \ldots, x_k \rangle, \quad \boldsymbol{y} = \langle y_1, y_2, \ldots, y_n \rangle,$

$\Theta: \langle \theta_{ij} \rangle_{k,n}$

$\boldsymbol{\sigma}(\boldsymbol{t}) = \langle \sigma(t_1), \sigma(t_2), \sigma(t_3), \ldots \rangle$

Logistiline regressioon: õppimine

$$\mathcal{E} = 1/n \times \sum_i (y_i^{(true)} - y_i^{(guess)})^2$$
$$= 1/n \times \sum_i (y_i^{(true)} - \sigma(\boldsymbol{\theta}^T \mathbf{x}_i))^2$$

Parimad parameetrid $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}} \mathcal{E}$$

Ekstreemumite leidmine

- $\partial \mathcal{E}(\boldsymbol{\theta})/\partial \theta_j = 0$ võrrandi lahendused annavad meile θ väärtusi, mis annavad funktsiooni $\mathcal{E}$ ekstreemumpunkte
- kuid see võrrand ei pruugi olla analüütiliselt lahendatav!

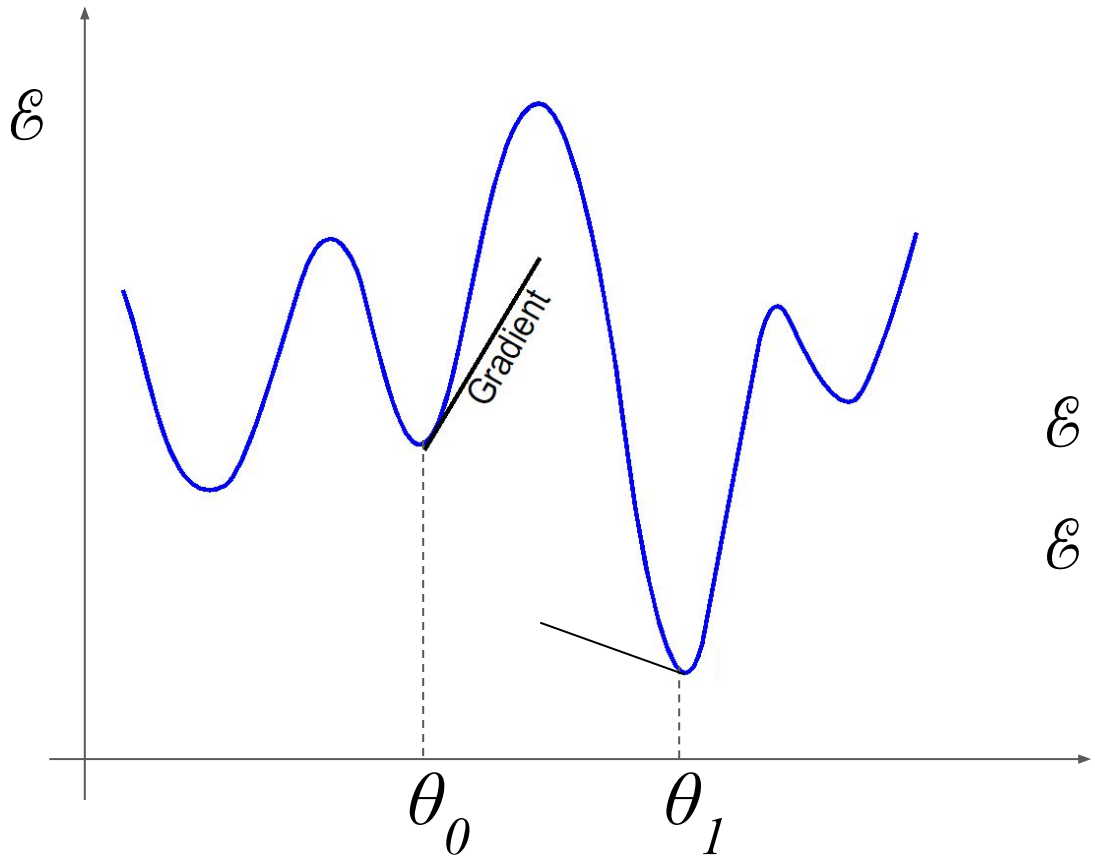
Gradientlaskumine

- $\partial \mathcal{E}(\boldsymbol{\theta})/\partial \theta_j = 0$ võrrandi lahendused annavad meile θ väärtusi, mis annavad funktsiooni $\mathcal{E}$ ekstreemumpunkte
- kuid see võrrand ei pruugi olla analüütiliselt lahendatav!
- $\partial \mathcal{E}(\boldsymbol{\theta}^{(0)})/\partial \theta_j$ omaette aga näitab meile seda, mis suunas ning kui kiiresti kasvab funktsioon $\mathcal{E}$ punktis $\boldsymbol{\theta}^{(0)}$

Gradientlaskumine

- $\partial \mathcal{E}(\boldsymbol{\theta})/\partial \theta_j = 0$ võrrandi lahendused annavad meile θ väärtusi, mis annavad funktsiooni $\mathcal{E}$ ekstreemumpunkte
- kuid see võrrand ei pruugi olla analüütiliselt lahendatav!
- $\partial \mathcal{E}(\boldsymbol{\theta}^{(0)})/\partial \theta_j$ omaette aga näitab meile seda, mis suunas ning kui kiiresti kasvab funktsioon $\mathcal{E}$ punktis $\boldsymbol{\theta}^{(0)}$
- võime liikuda sinna iteratiivselt, kasutades osatuletise väärtusi igas uues punktis, et valida liikumissuunda ja -kiirust:

Ühemõõtmeline gradient (osatuletis)

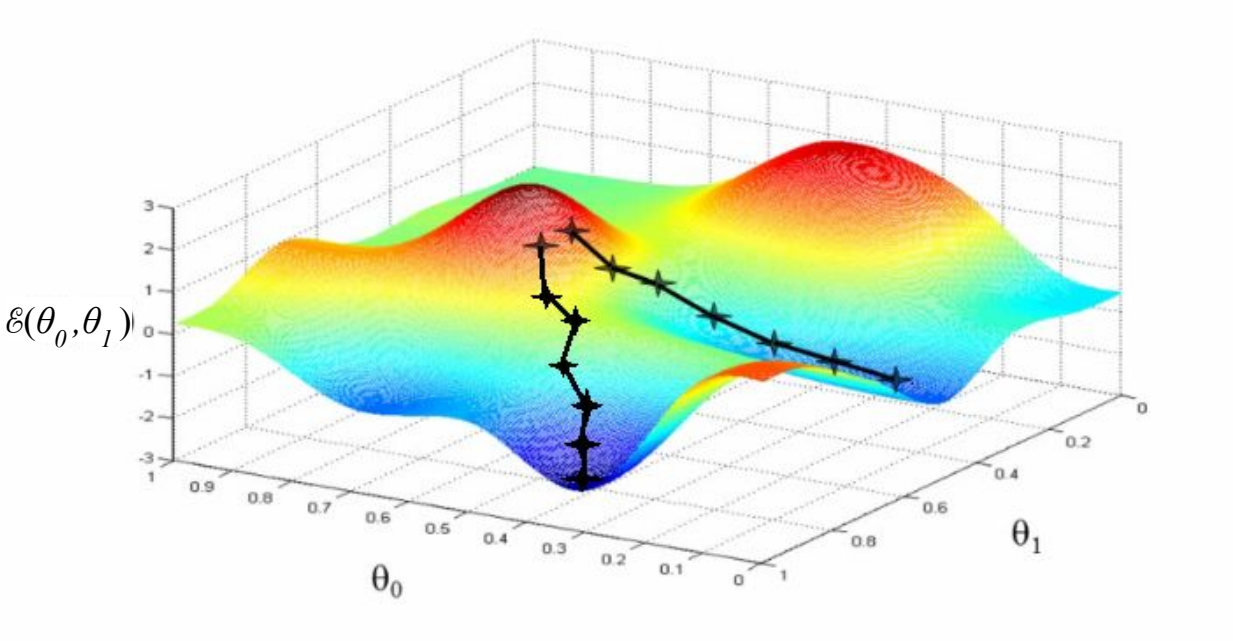


$$\mathcal{E}'(\theta_0) \approx 1.8$$

$$\mathcal{E}'(\theta_1) \approx -0.4$$

Gradientlaskumine

Nt. 2D:



Andrew Ng'i ML class

Gradientlaskumine

Veafunktsiooni (mitmemõõtmeline) gradient:

$$\nabla \mathcal{E} = \partial \mathcal{E}(\boldsymbol{\theta})/\partial \boldsymbol{\theta} =$$
$$= \langle \partial \mathcal{E}(\boldsymbol{\theta})/\partial \theta_1, \partial \mathcal{E}(\boldsymbol{\theta})/\partial \theta_2, \partial \mathcal{E}(\boldsymbol{\theta})/\partial \theta_3, \ldots \rangle$$

Gradientlaskumine: algoritm

0. α (õppimiskiirus) on väike arv, nt. 0.01
1. alusta parameetrite (rnd) algväärtusega $\boldsymbol{\theta}^{(0)} = rnd$
2. $k = 1$
   a. leia veamäära $\mathcal{E}$ tuletiste väärtusi punktis $\boldsymbol{\theta}^{(k-1)}$
   b. iga j jaoks:

$$\theta_j^{(k)} = \theta_j^{(k-1)} - \alpha \times \nabla \mathcal{E}(D; \boldsymbol{\theta}^{(k-1)})_j$$

   c. $k \mathrel{+}= 1$
   d. korda koondumiseni

Gradientlaskumine log. reg. jaoks

Kuidas leida parameetrid θ logistilise regressiooni lahendamiseks:

0. Treeningandmed
   $D = (\langle \mathbf{x}^{(1)}, y^{(1)} \rangle, \langle \mathbf{x}^{(2)}, y^{(2)} \rangle, \ldots, \langle \mathbf{x}^{(m)}, y^{(m)} \rangle)$
1. alustame suvalisest θ'st: $\boldsymbol{\theta}^{(0)} = rnd$
2. kordame kuni väsimuseni:

$$\theta_j^{(k+1)} = \theta_j^{(k)} - \alpha \times 1/m \times \sum_i x_j^{(i)} (y^{(i)} - \sigma(\boldsymbol{\theta}^{(k)T} \mathbf{x}^{(i)}))$$

Gradientlaskumine log. reg. jaoks

Kuidas leida parameetrid θ logistilise regressiooni lahendamiseks:

0. Treeningandmed
   $D = (\langle \mathbf{x}^{(1)}, y^{(1)} \rangle, \langle \mathbf{x}^{(2)}, y^{(2)} \rangle, \ldots, \langle \mathbf{x}^{(m)}, y^{(m)} \rangle)$
1. alustame suvalisest θ'st: $\boldsymbol{\theta}^{(0)} = rnd$
2. kordame kuni väsimuseni:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha \times 1/m \times \sum_i \mathbf{x}^{(i)} (y^{(i)} - \sigma(\boldsymbol{\theta}^{(k)T} \mathbf{x}^{(i)}))$$

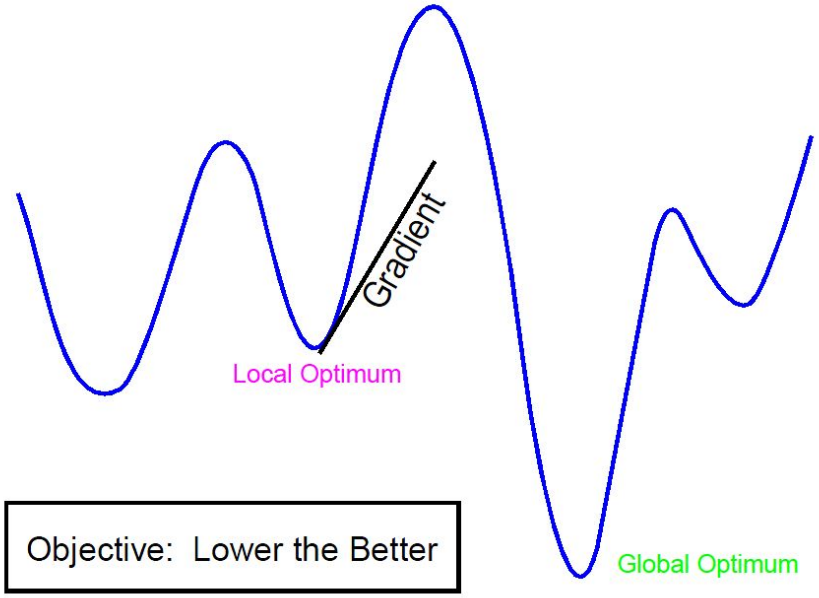
Asjakohane gradientlaskumise juures: õppimiskiirus

Õppimiskiirus

- liiga väike: õppimine võtab liiga kaua
- liiga suur: õppimine on ebastabiilne ja ei pruugi koonduda

Asjakohane gradientlaskumise juures: local optima

Üks suur probleem gradientlaskumisel:
- lokaalsed miinimumid



Otsesuunatud tehisnärvivõrgud (feed-forward neural networks)

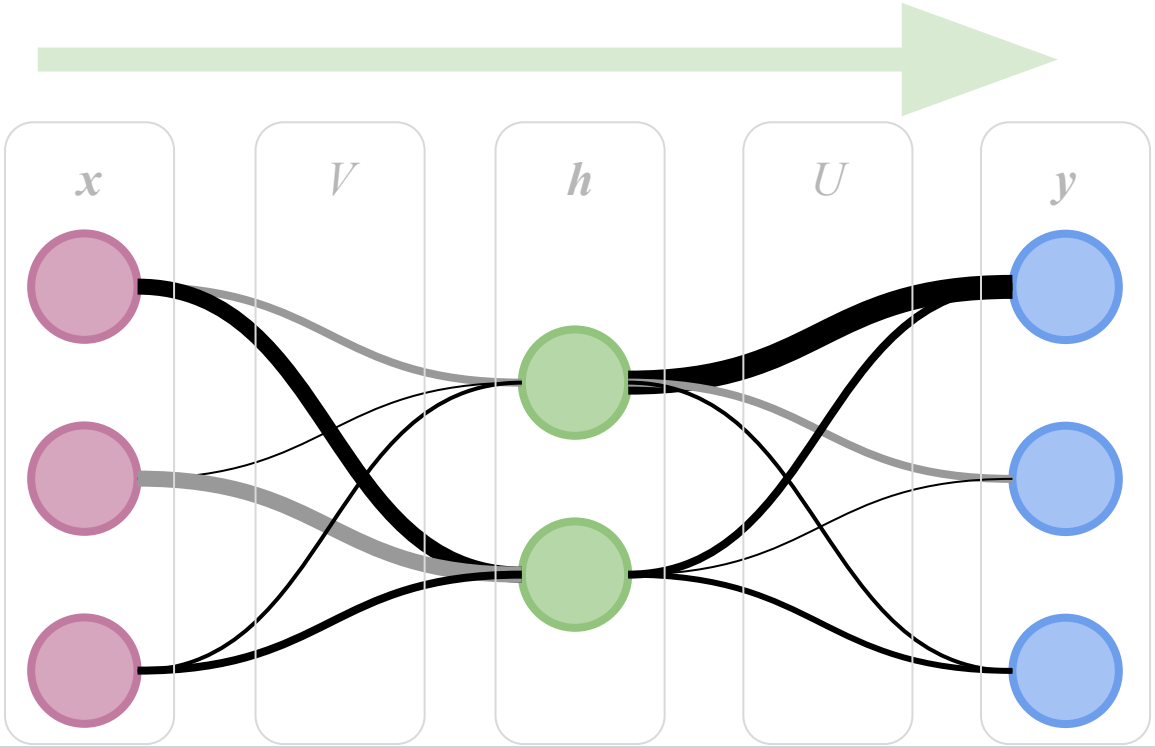
$$\boldsymbol{h} = \boldsymbol{\sigma}(V^T \boldsymbol{x})$$ (peitkiht / -seisund)

$$\boldsymbol{y} = \boldsymbol{\sigma}(U^T \boldsymbol{h})$$ (väljund)

Otsesuunatud tehisnärvivõrgud (feed-forward neural networks)

$\boldsymbol{h} = \boldsymbol{\sigma}(V^T\boldsymbol{x})$ (peitkiht / -seisund)

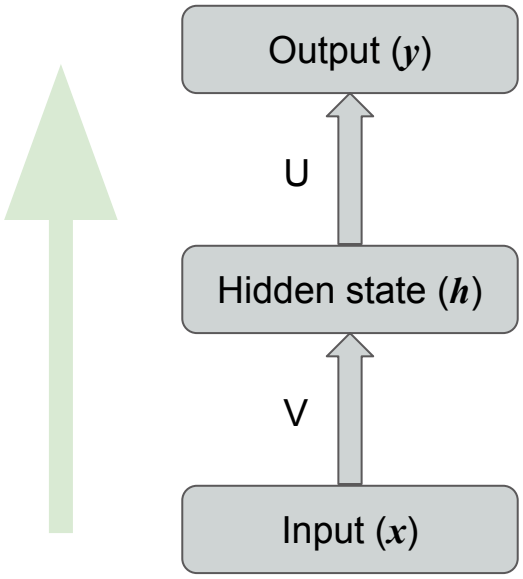
$\boldsymbol{y} = \boldsymbol{\sigma}(U^T\boldsymbol{h})$ (väljund)



Otsesuunatud tehisnärvivõrgud (feed-forward neural networks)

$$\boldsymbol{h} = \boldsymbol{\sigma}(V^T \boldsymbol{x})$$ (peitkiht / -seisund)

$$\boldsymbol{y} = \boldsymbol{\sigma}(U^T \boldsymbol{h})$$ (väljund)



Otsesuunatud tehisnärvivõrgud (feed-forward neural networks)

Õppimine: “tagasilevi” (backpropagation)
= gradientlaskumine vähimruutude meetodiga

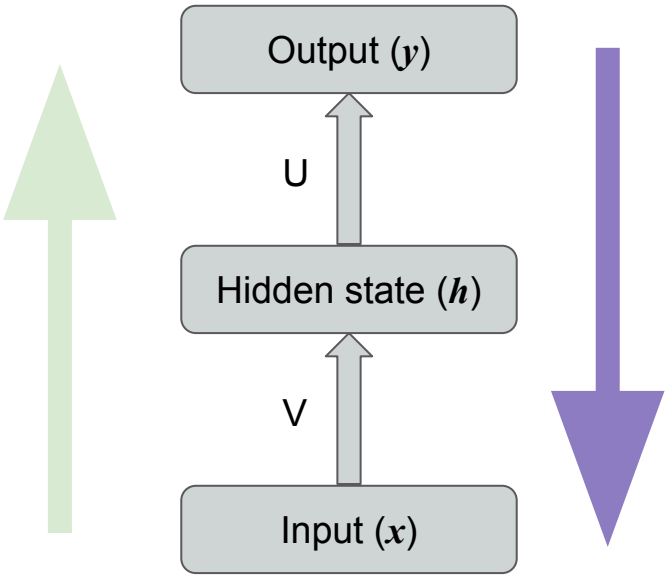
$$\boldsymbol{h} = \boldsymbol{\sigma}(V^T \boldsymbol{x})$$

$$\boldsymbol{y} = \boldsymbol{\sigma}(U^T \boldsymbol{h})$$

$$\mathcal{E} = \sum_i (y_i^{(ref)} - y_i^{(hyp)})^2$$

$$U^{(k+1)} = U^{(k)} - \alpha \cdot \partial\mathcal{E}/\partial U$$

$$V^{(k+1)} = V^{(k)} - \alpha \cdot \partial\mathcal{E}/\partial V$$



Üldisemalt

1. **Aktiveerimisfunktsioon** (enne: $\sigma(V^T x)$) ei pea olema ainult logistiline, teisi: tanh, ReLU, jt

Üldisemalt

1. **Aktiveerimisfunktsioon** (enne: $\sigma(V^T x)$) ei pea olema ainult logistiline, teisi: tanh, ReLU, jt

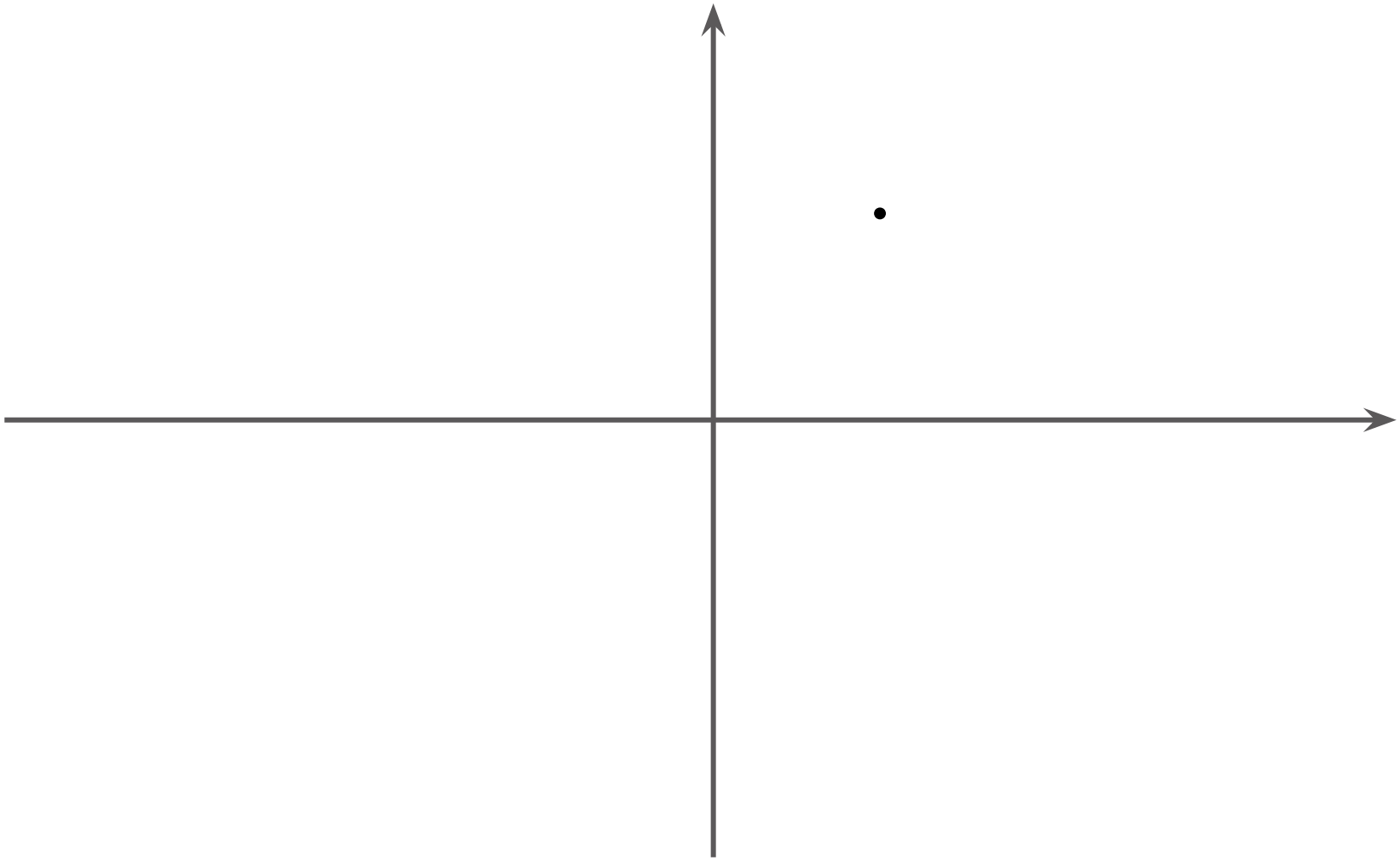
2. **Arvutuskäik** (computation graph) = mida tehakse sisendi jm. parameetritega, et arvutada väljundit

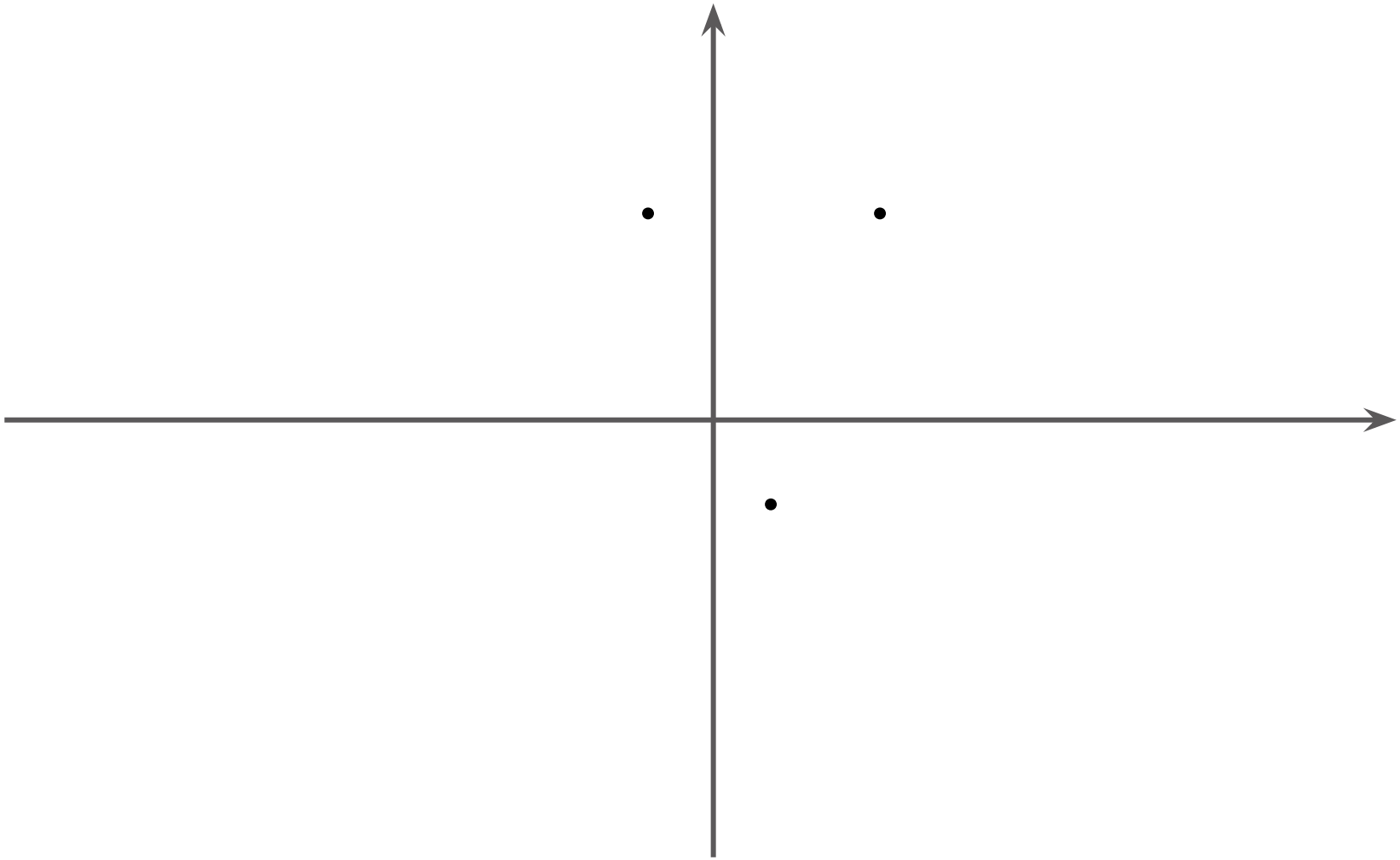
Üldisemalt

1. **Aktiveerimisfunktsioon** (enne: $\sigma(V^T x)$) ei pea olema ainult logistiline, teisi: tanh, ReLU, jt

2. **Arvutuskäik** (computation graph) = mida tehakse sisendi jm. parameetritega, et arvutada väljundit

3. Alternatiivvaade:
   närvivõrgud = **vektorruumide** teisenduste õppimine





Lineaarsus

- pööramine, väänamine, nihutamine, peegeldamine = lineaarteisendused
  - neid saab kõiki teha maatrikskorrutise abil
- teised = mittelineaarsed teisendused
  - neid ei saa maatrikskorrutise abil teha
- vahe = lineaarsete kombo on lineaarne, ehk kombo “võimsus”/“keerukus” ei kasva

Tehisnärvivõrkude jõu allikad

1. automaatsed tunnused!
   - iga neuron / kiht on nagu logistiline regressioon
   - väljundkihi sisendiks on peitkihi väljund

Mida see tähendab:

- õigete tunnuste ekstraheerimine (nagu tavalises masinõppes) ei ole enam nii oluline

Tehisnärvivõrkude jõu allikad

2. siirdeõpe
   - alusta õppimist ühe ülesandega
   - jätka õppimist terve mudeli või selle osaga ning õpeta talle teist ülesannet
   - kui ülesanded on sarnased, siis üks aitab teist!
   - nt. kui 1. ülesanne on üldine (palju andmeid) ning 2. -- spetsiifilisem (vähe andmeid)
   - “eeltreenimine” ja “peenhäälestamine” (pretraining + fine-tuning)

Tehisnärvivõrkude jõu allikad

3. väga head pidevate funktsioonide lähendajad
   - Universaalse lähendamise teoreem (Cybenko 1989, Hornik 1991)

https://neuro.cs.ut.ee/what-neural-networks-actually-do/

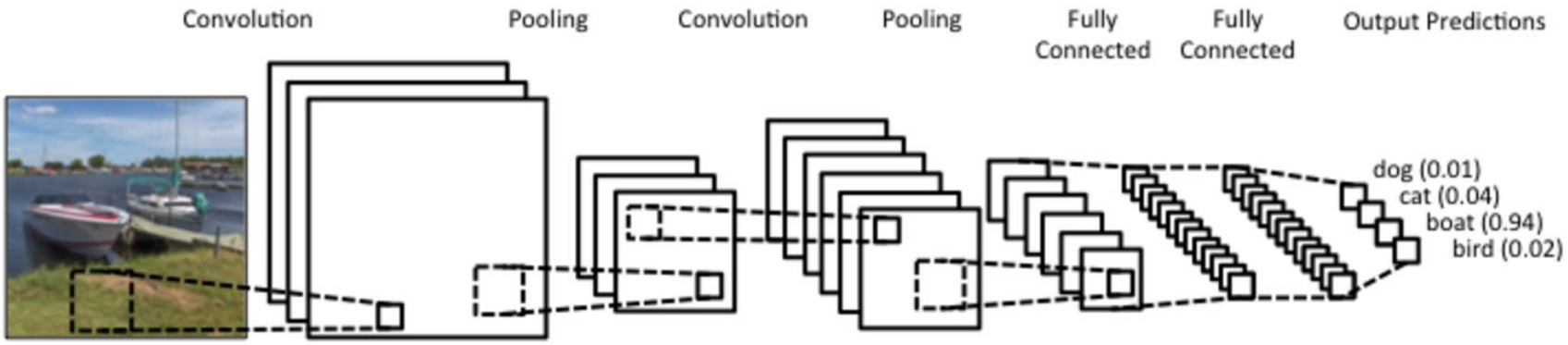
Kas see tähendab, et tehisnärvivõrgud on kõikvõimsad ja varsti vallutavad maailma?

Kas see tähendab, et tehisnärvivõrgud on kõikvõimsad ja varsti vallutavad maailma?

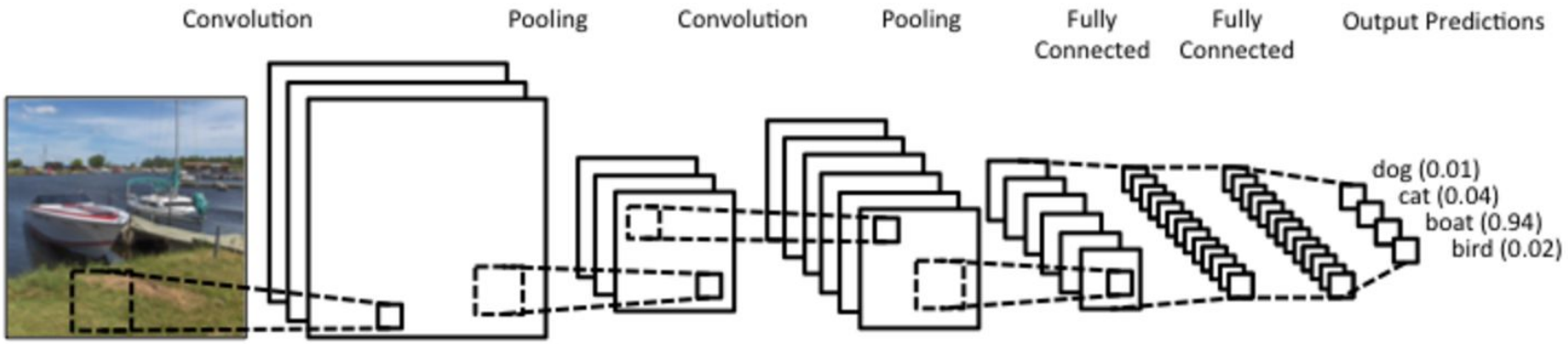
Praktikas:

- Ise ei pea osatuletisi leidma!
- PyTorch/TensorFlow/jne. teevad seda **automaatselt**:
  - teegi käsud muutujate lisamaks: x, y, Θ, jne
  - defineerime arvutuskäiku x'st y'ni kasutades Θ'd
  - täidame x ja y andmetega
  - laseme teegil optimeerida Θ'd - valime veafunktsiooni, õppimisalgoritmi, muid parameetreid
- See toetab mistahes hullu arvutuskäiku, (kuid õppimine võib osutuda raskeks)

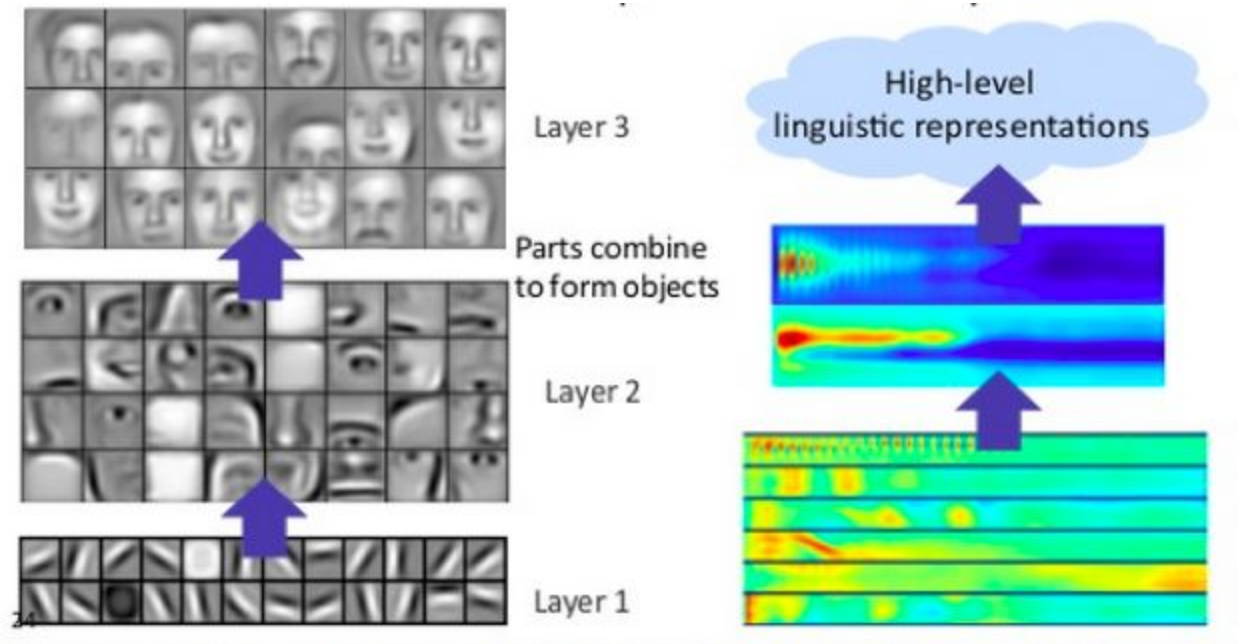
Konvolutsioonivõrk
(CNN/convolutional neural network)



Konvolutsioonivõrk
(CNN/convolutional neural network)



+ parameetrite jagamine
(parameter sharing)

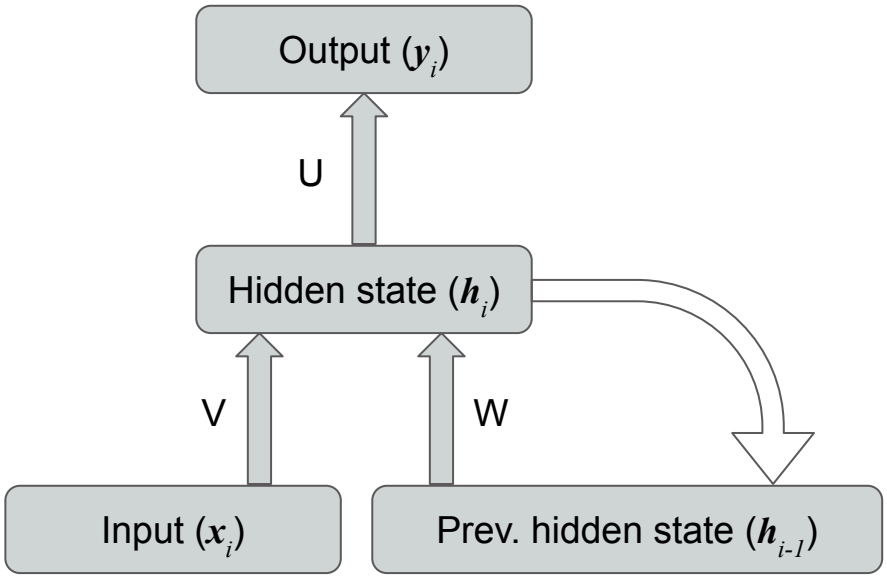


https://deeplearning4j.org/

Rekurrentsed tehisnärvivõrgud (RNN/recurrent neural networks)

$$\boldsymbol{h}_i = \boldsymbol{\sigma}(V^T \boldsymbol{x}_i + W^T \boldsymbol{h}_{i-1}) \qquad \text{(peitkiht)}$$

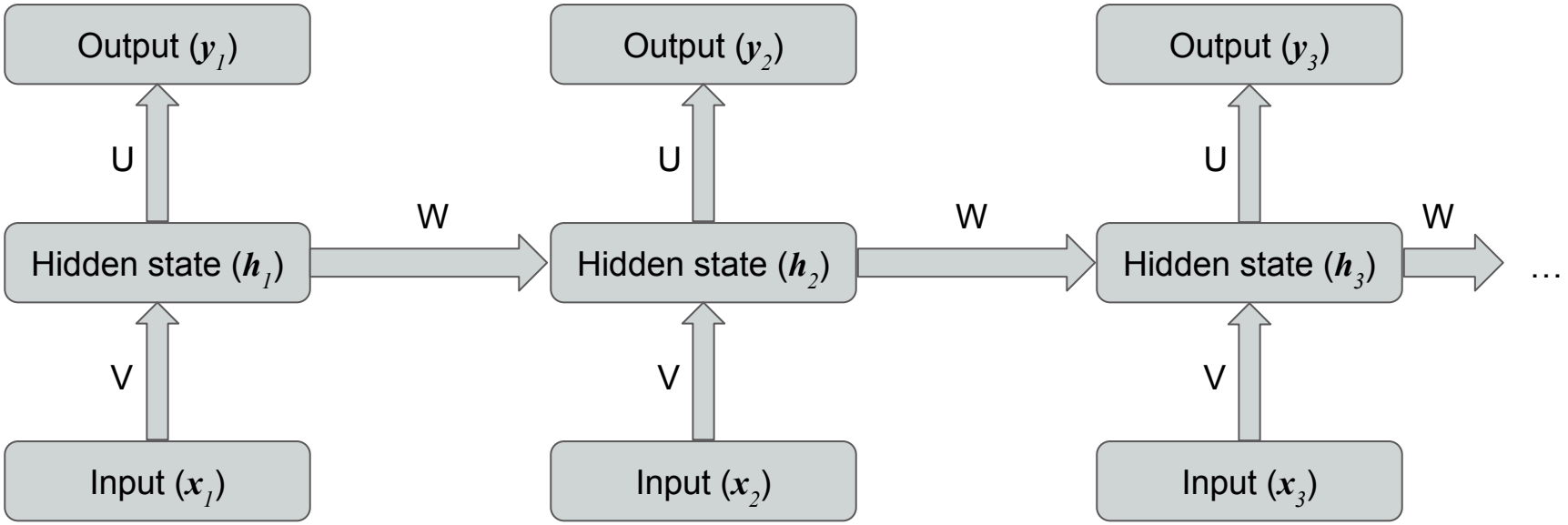
$$\boldsymbol{y}_i = \boldsymbol{\sigma}(U^T \boldsymbol{h}_i) \qquad \text{(väljund)}$$



Rekurrentsete võrkude “lahti rullimine” (unwrapping)

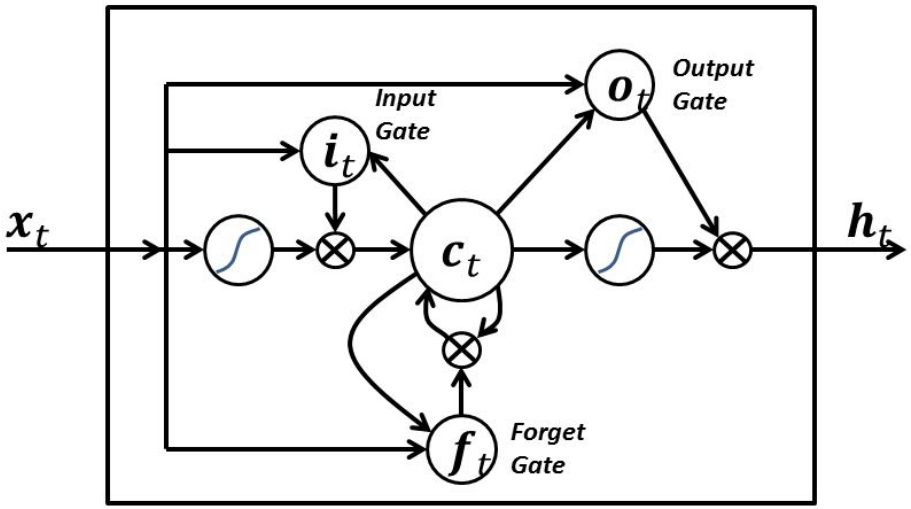
$$\boldsymbol{h}_i = \boldsymbol{\sigma}(V^T \boldsymbol{x}_i + W^T \boldsymbol{h}_{i-1}) \qquad \text{(peitkiht)}$$

$$\boldsymbol{y}_i = \boldsymbol{\sigma}(U^T \boldsymbol{h}_i) \qquad \text{(väljund)}$$



LSTM: pikk lühiajaline mälu (long short-term memory)

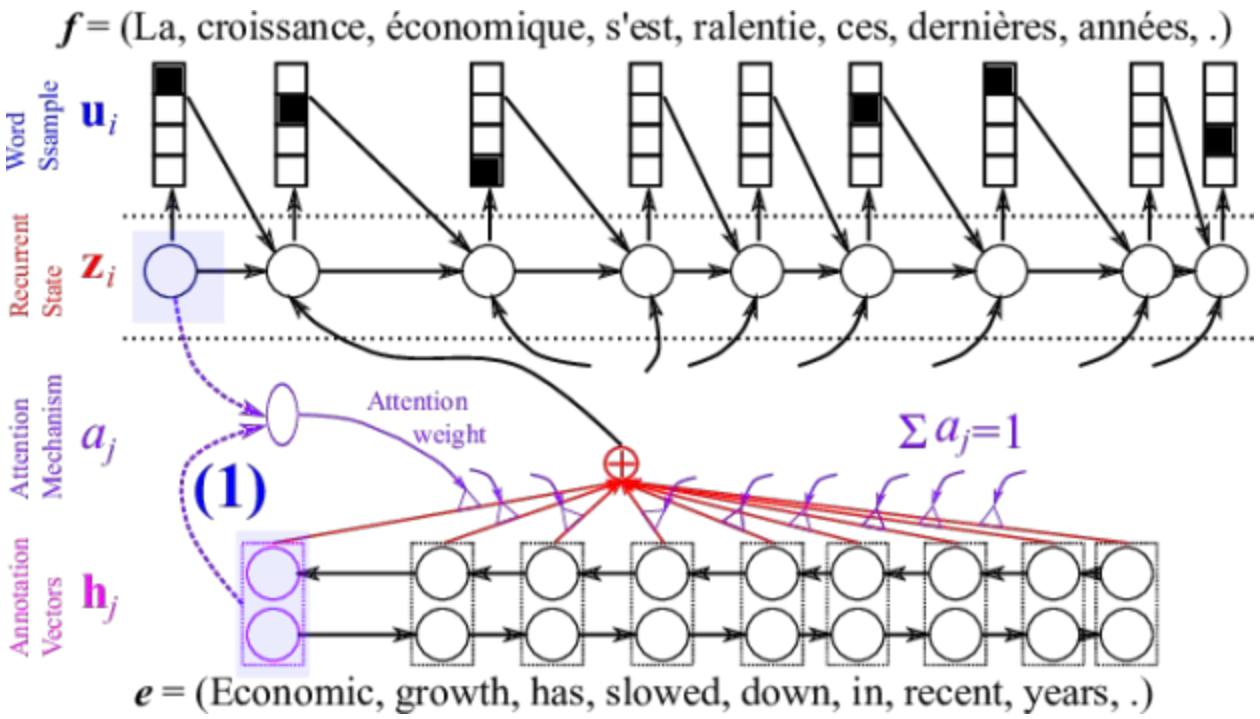
$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

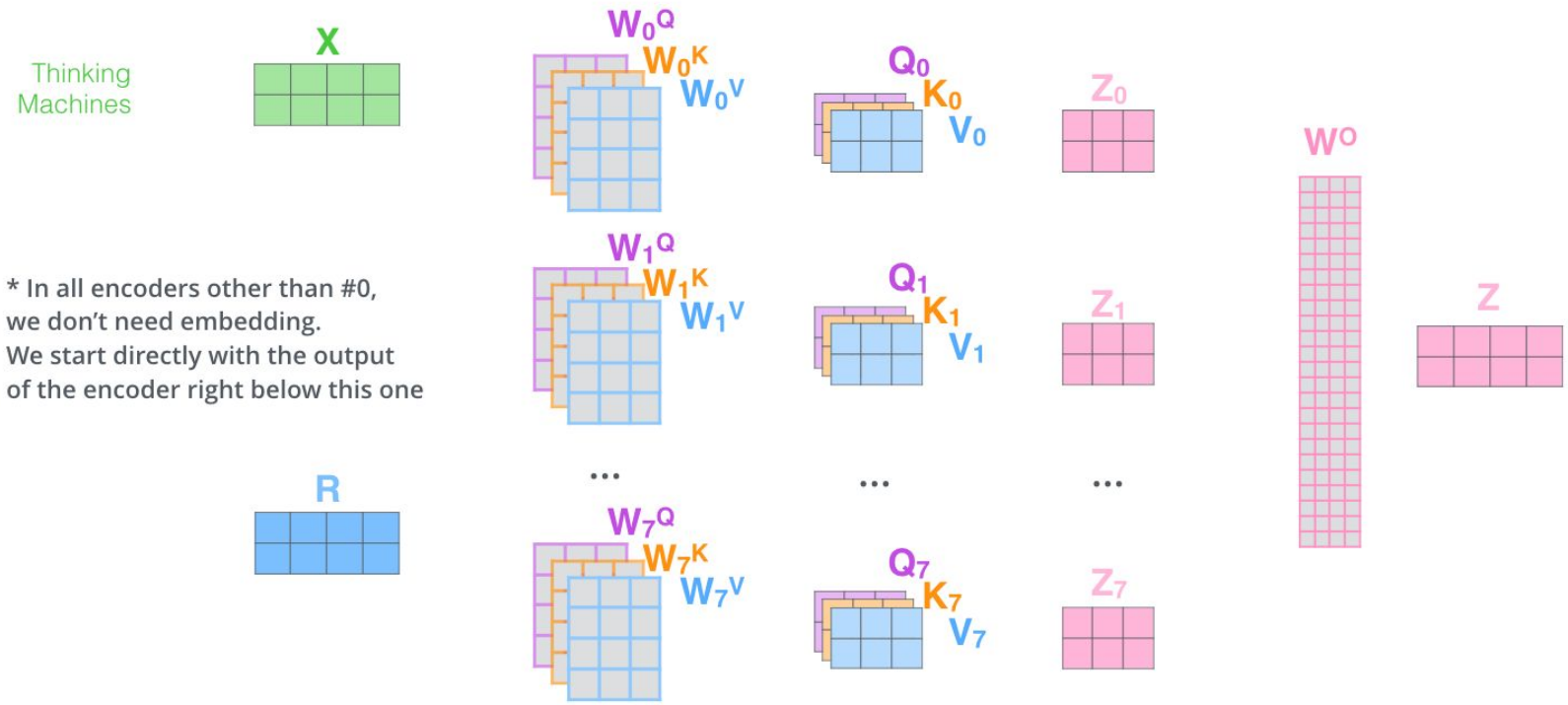


Src: wikipedia

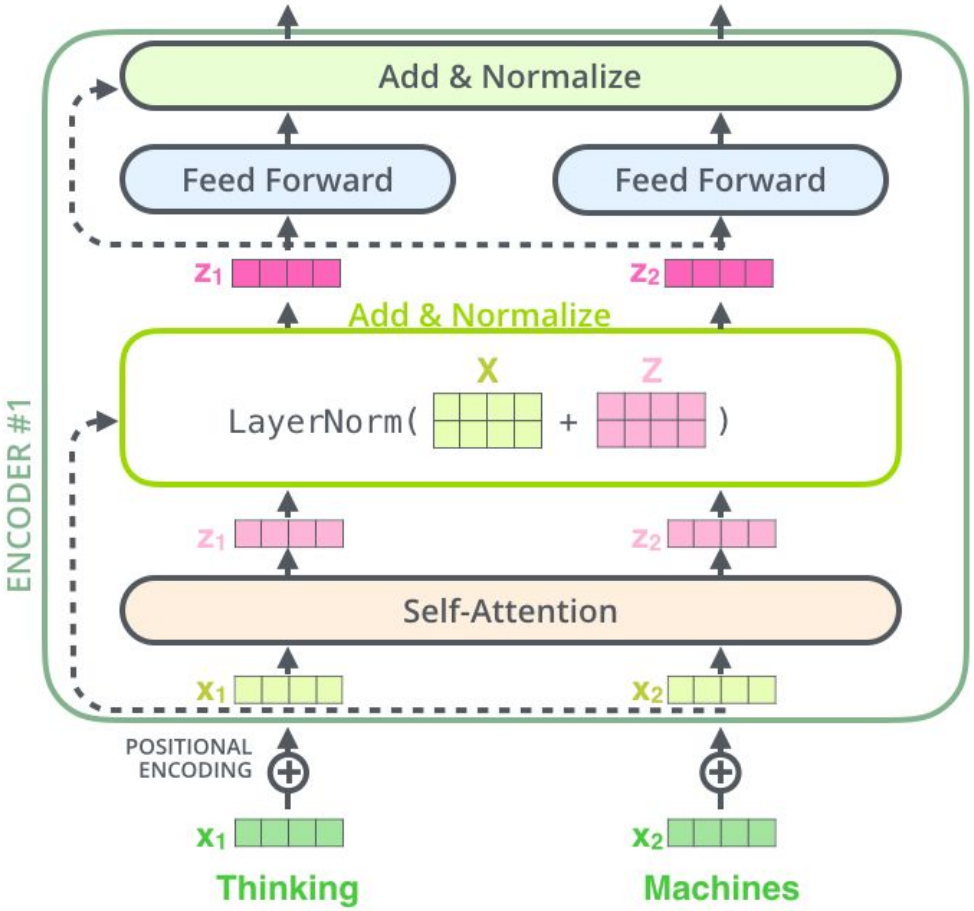
Mida oskavad närvivõrgud:
masintõlge

translate.ut.ee
translate.google.com
masintolge.ee
translate.yandex.com
bing.com/translator
…

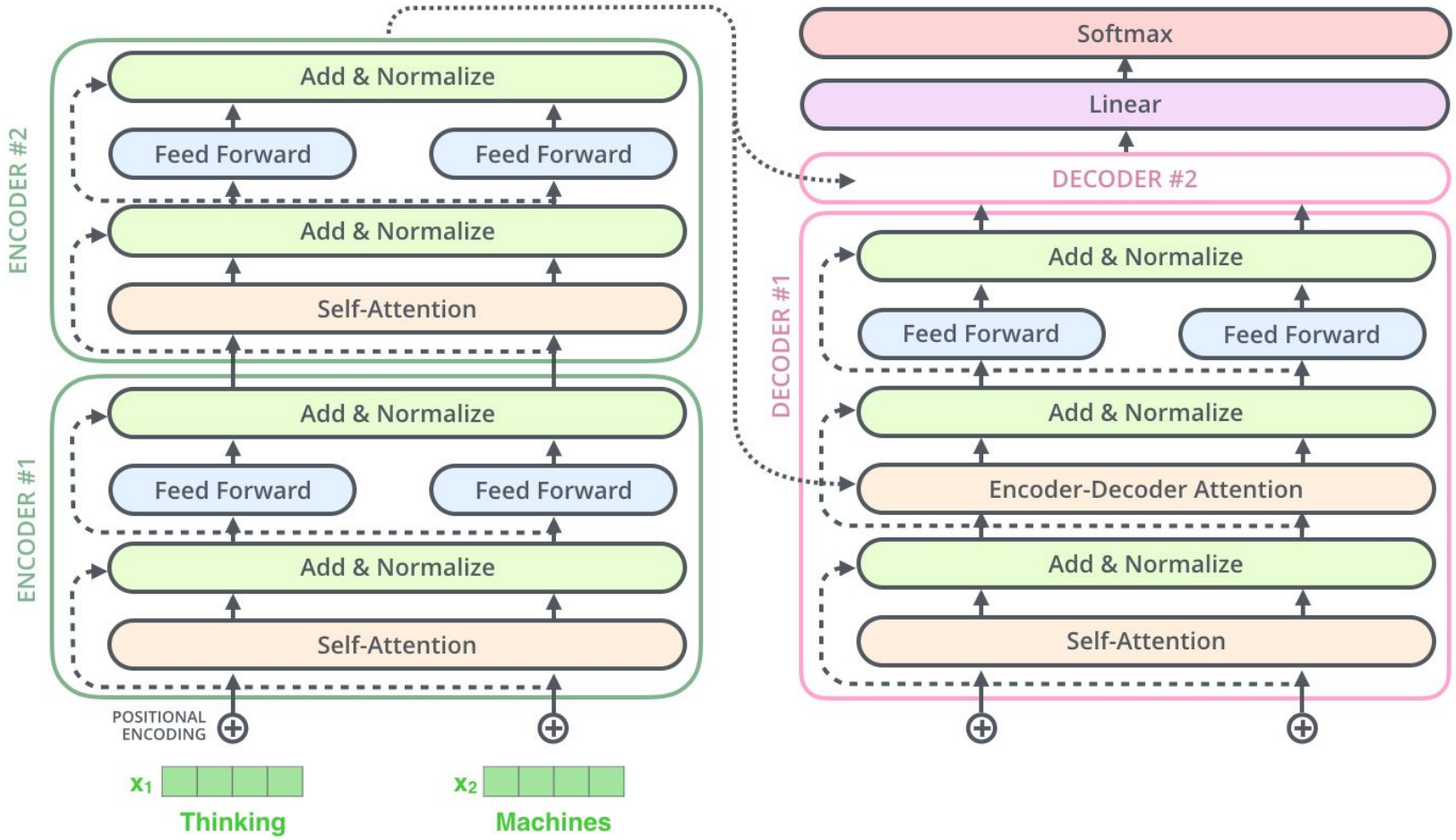

f = (La, croissance, économique, s'est, ralentie, ces, dernières, années, .)
Word Ssample
u_i
Recurrent State
z_i
Attention Mechanism
a_j
Attention weight
$\Sigma a_j=1$
(1)
Annotation Vectors
h_j
e = (Economic, growth, has, slowed, down, in, recent, years, .)



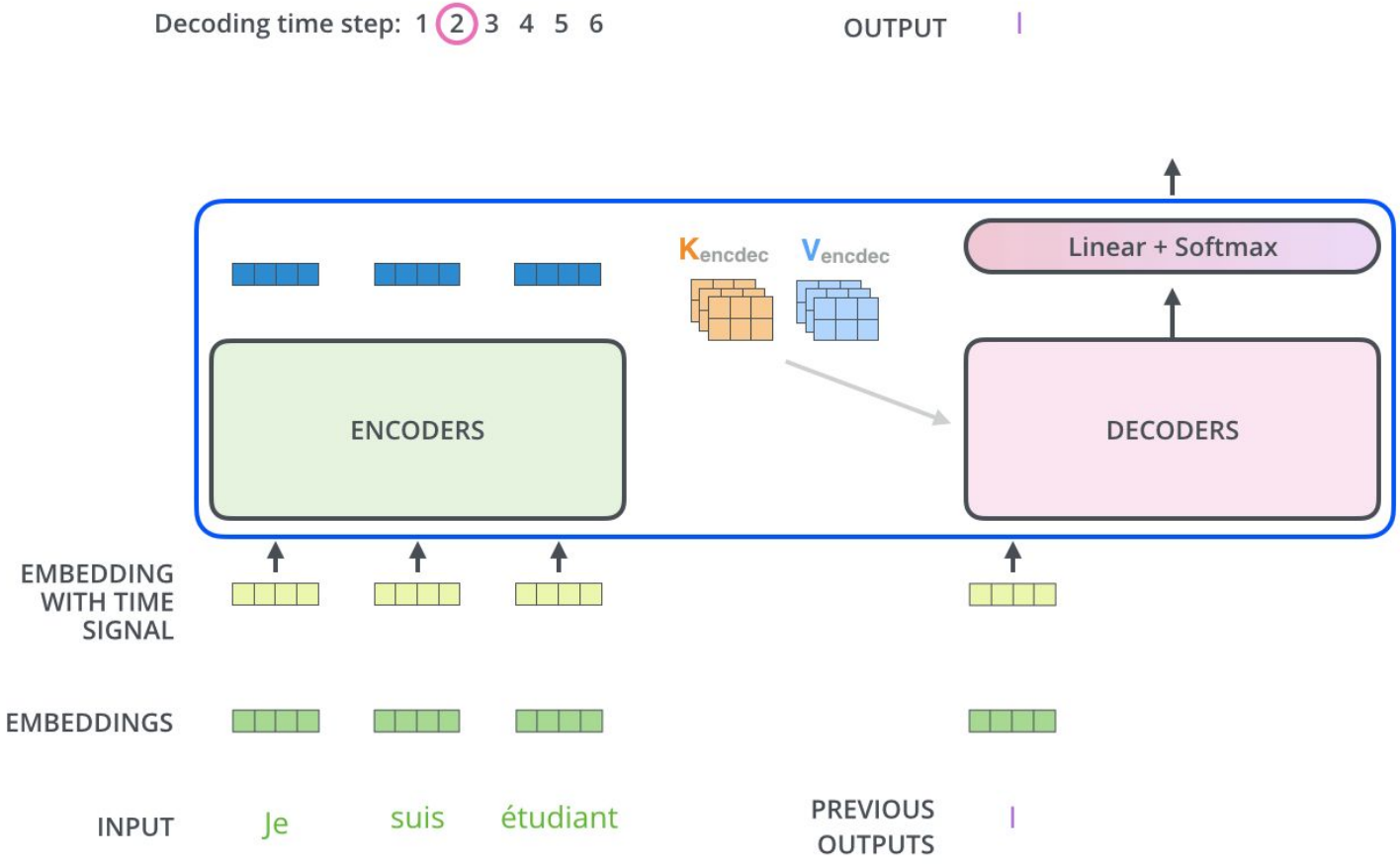
Src: [Illustrated Transformer]



Src: Illustrated Transformer



Src: [Illustrated Transformer]

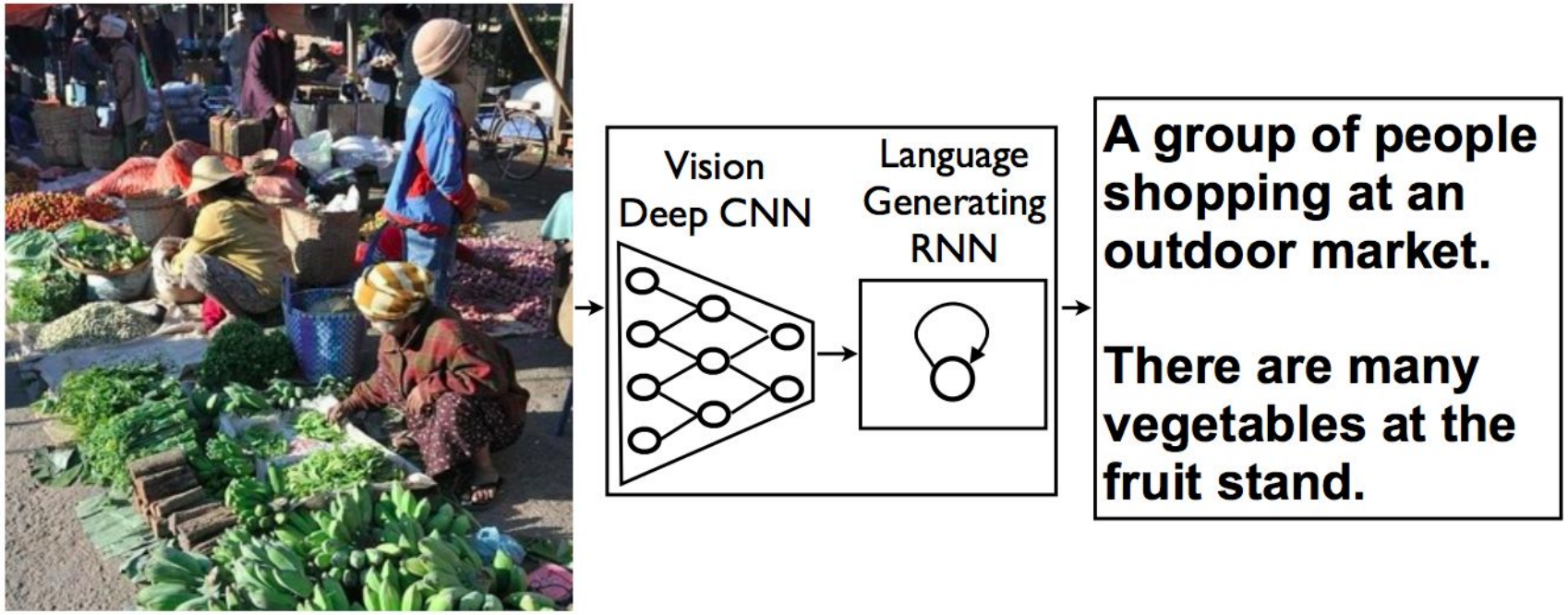


Src: Illustrated Transformer

Mida oskavad närvivõrgud:
kõnesüntees

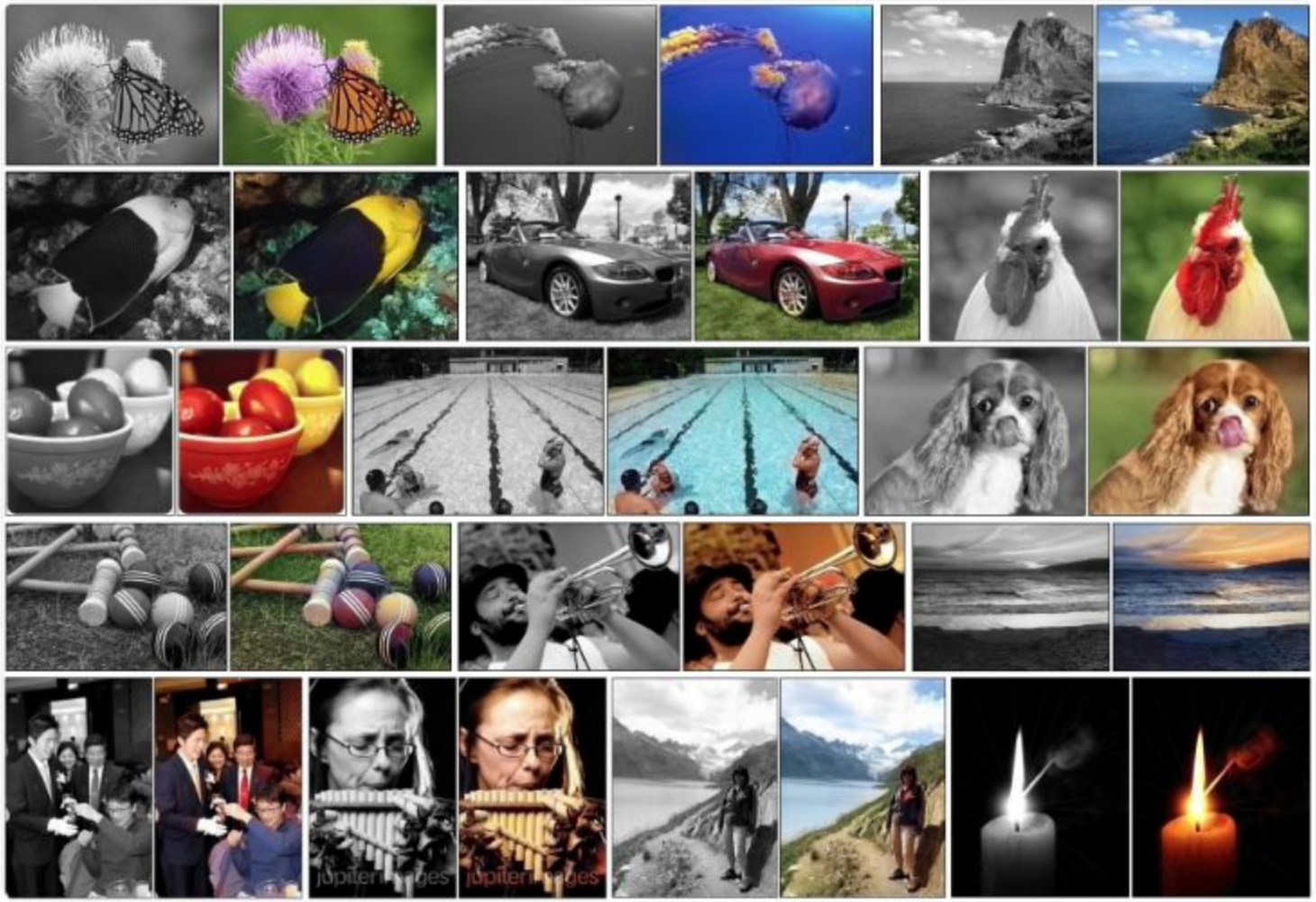
neurokone.ee

Mida oskavad närvivõrgud: pildi kirjeldamine loomulikus keeles



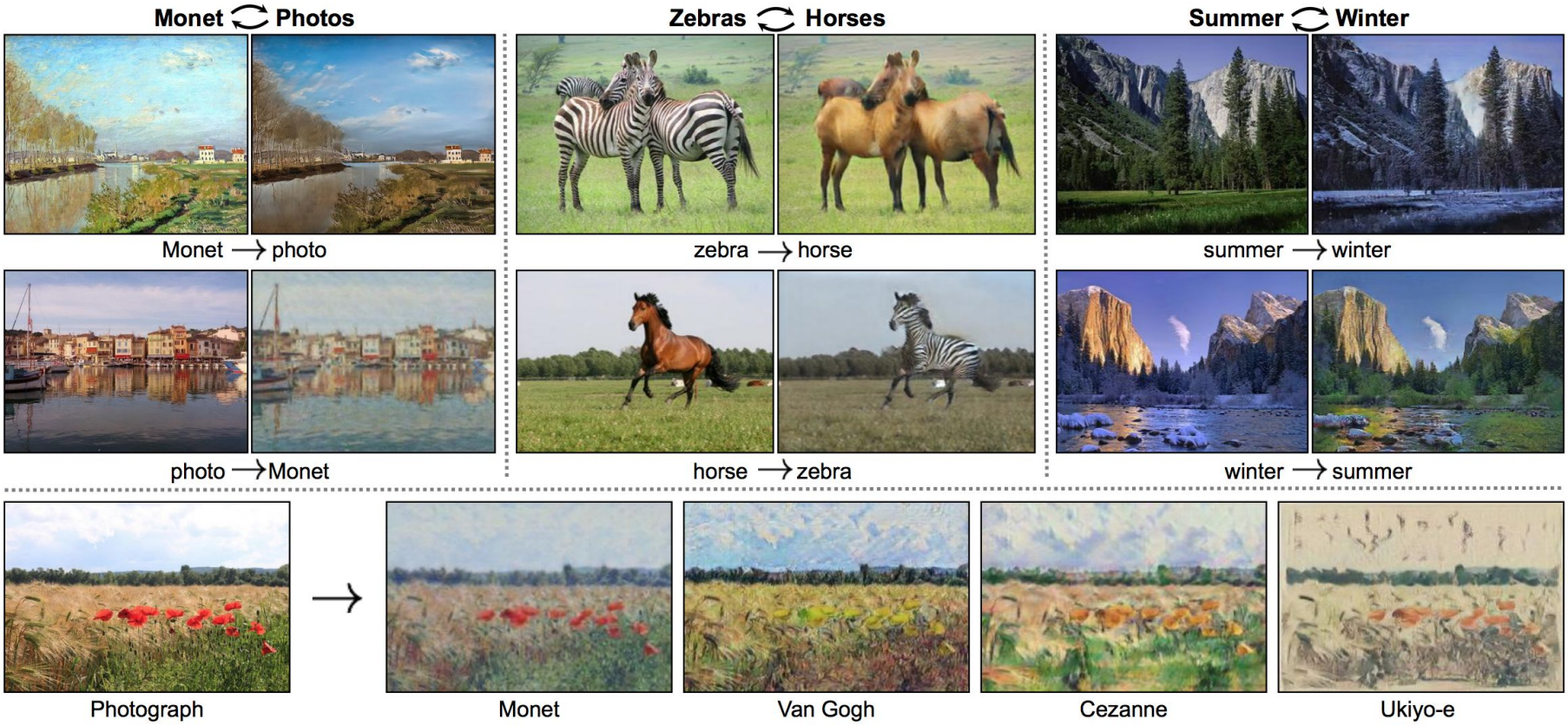
https://arxiv.org/abs/1411.4555

Mida oskavad närvivõrgud:
mustvalge pildi värvide taastamine



http://machinelearningmastery.com/inspirational-applications-deep-learning/

Mida oskavad närvivõrgud: juhendamata piltide teisendamine, CycleGAN



Mida oskavad närvivõrgud:
mängude mängimine
nt. AlphaGO
nt. Breakout
nt. teised Atari mängud

deepmind.com

Mida oskavad närvivõrgud:
pildi genereerimine kirjelduse põhjal
(Dall·e 2)

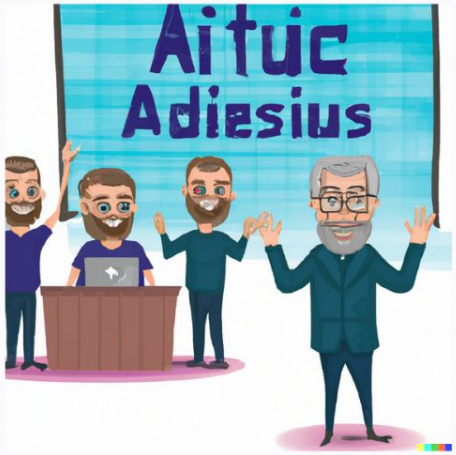
DALL·E My collection

Edit the detailed description

Surprise me Upload

Estonian professor teaching AI to excited students in cartoon style

Generate



openai.com

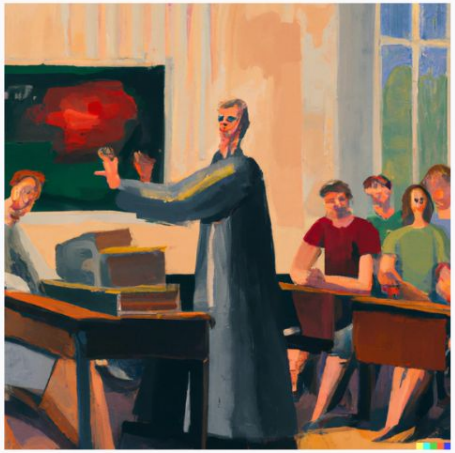
DALL·E My collection

Edit the detailed description

Surprise me Upload

Estonian professor teaching artificial intelligence to bored students in oil painting style

Generate

openai.com

DALL·E My collection

Edit the detailed description

Surprise me Upload

Estonian professor teaching AI to unsuspecting students in photorealistic style

Generate

openai.com

End-to-end õppimine

- Sisendiks on andmed toorkujul
- Väljundiks on soovitud väljund
- Kõik vahepealsed esitused, vektorid, teisendused jms on tuletatud automaatselt

Tehisnärvivõrgud + tõenäosused

- Kui ideaalne väljund on 1-hot (nt. sõnad, või väljundiklassid -- “1-vs-all”)

- Siis kasutatakse väljundis nn. SoftMax kihti:
  - neuronite väljund $\boldsymbol{z} = (z_1, z_2, \ldots)$
  - lõplik kihi väljund on: $\exp(z_i) / \sum_k \exp(z_k)$

Tehisnärvivõrgud + tõenäosused

- Kui ideaalne väljund on 1-hot (nt. sõnad, või väljundiklassid -- “1-vs-all”)

- Siis kasutatakse väljundis nn. SoftMax kihti:
  - neuronite väljund $\boldsymbol{z} = (z_1, z_2, \ldots)$
  - lõplik kihi väljund on: $\exp(z_i) / \sum_k \exp(z_k)$
  - = tõenäosusjaotus!!!11!!!!1111!1111ÜKSTEIST

Kokkuvõte

- Gradientlaskumine on äge!
- Tehisnärvivõrgud on toredad!
- Teie olete lahedad!

Kokkuvõte

- edasilevi-/konvolutisooni-/rekurrentsed närvivõrgud, GAN jne
- õpivad reeglina gradientlaskumise-klassi algoritmide abil
- mistahes arvutuskäik*
- tuletised - automaatsed
- lõpuks on sageli väljundiks tõenäosusjaotuse hinnang (SoftMax)

Küsimused

1. Kas tehisnärvivõrkude õpetamisel tagasilevi algoritmiga võib tekkida probleeme lokaalse miinimumiga?
2. Põhjendage, miks deep learning / närvivõrgud pole siiski AGI